

“Minería de Texto aplicada para el análisis de denuncias penales de delitos contra la propiedad en la provincia de Salta”

Rivero, Agustín Saúl

Ingeniería en Informática

Facultad de Ingeniería

Proyecto de Grado

Año 2022



**“Minería de Texto aplicada para el análisis de denuncias penales de delitos
contra la propiedad en la provincia de Salta”**

Rivero, Agustín Saúl

Ingeniería en Informática

Facultad de Ingeniería

Proyecto de Grado

Año 2022

Título

Ingeniero en Informática

Profesor guía

Nombre: Lic. Patricia Aballay

Firma: _____

Tribunal Evaluador

Nombre: _____ **Firma:** _____

Nombre: _____ **Firma:** _____

Nombre: _____ **Firma:** _____

Alumno

Nombre: Rivero, Agustín Saúl

Firma: _____

Fecha de exposición: ____/____/____

Dedicatorias

Este trabajo está dedicado a mis padres Saúl y Susana, y a mi hermana Ailén, que me apoyaron de manera incondicional desde siempre.

Agradecimientos

Principalmente a mis padres, que fueron no solo los que me apoyaron económicamente, sino que me brindaron sus consejos, apoyo incondicional y amor constante.

A mi abuela Elba que me recibió y acogió innumerables veces durante todos mis estudios.

A mi familia, amigos, compañeros y profesores que fueron indispensables para poder llegar a este punto, haciendo todo el proceso más ameno y llevadero.

A mi profesora guía, la Lic. Patricia Aballay por su compromiso total con la realización de este proyecto, brindando consejo, guía y nuevas opciones permanentemente.

A la Lic. María Lorena Talamé, sobre todo por su consejo cuando apenas estaba comenzando con este proyecto.

Índice

Abstract	9
Capítulo 1 - Introducción	10
1.1 Descripción del Problema	12
1.2 Motivación	13
1.3 Organización del Trabajo	13
Capítulo 2 - Estado de la cuestión	15
2.1 Minería de Datos	15
2.2 Metodología de minería de datos: CRISP-DM	15
2.3 Minería de Texto	17
2.3.1 Fases de la Minería de Textos	18
2.3.2 Las Aplicaciones de la Minería de Texto	18
2.3.3 Tareas en la Minería de Texto	19
2.4 Agrupamiento	20
2.4.1 Agrupamiento de Documentos	20
2.4.2 Tipos de Clustering	20
2.4.3 Algoritmos de Clustering	21
2.5 Proyectos Similares	24
Capítulo 3 - Comprensión del Negocio	27
3.1 Definición del Problema	27
3.2 Solución Propuesta	28
3.3 Objetivos del Proyecto	28
3.3.1 Objetivo General	28
3.3.2 Objetivos Específicos	28
3.4 Criterios de Éxito del Proyecto	28
3.5 Evaluación de la Situación	29
3.5.1 Fuentes de Datos y Conocimientos	29
3.5.2 Requerimientos, Presunciones y Restricciones	29
3.6 Plan de Proyecto	30
3.6.1 Alcance del Proyecto	30
3.6.2 Estructura de Desglose del Trabajo (EDT)	31
3.6.3 Diccionario de la EDT	32
3.6.4 Lista de Actividades	34
3.6.5 Requisitos de Recursos de la Actividad	35
3.6.6 Cronograma del Proyecto	37
3.6.7 Gestión de Costos y Adquisiciones	40
3.6.8 Descripciones de los Cargos	42
3.6.9 Gestión de Riesgos	42

3.6.10 Alternativas Tecnológicas	46
3.6.11 Elección de las Alternativas Tecnológicas	50
3.6.12 Análisis Costo-Beneficio del Proyecto	51
Capítulo 4 - Comprensión de los Datos	53
4.1 Datos Iniciales	53
4.2 Descripción de los Datos	53
4.3 Exploración de los Datos	53
4.3.1 Lectura de Datos	54
4.3.2 Transformación de Datos	55
4.3.3 Bolsa de Palabras	56
4.4 Calidad de los Datos	58
Capítulo 5 - Preparación de los Datos	59
5.1 Limpieza de Datos	59
5.1.1 Datos Faltantes	59
5.1.2 Datos Repetidos	61
5.1.3 Datos No Correspondientes	65
5.1.4 Limpieza de Atributos	66
5.2 Selección de Atributos	75
5.2.2 FchConfirmación	76
5.2.3 Dependencia	76
5.2.4 ActaNro	76
5.2.5 Anio	77
5.2.6 FchHrHecho	77
5.2.7 LugarHecho	79
5.2.8 Delito	79
5.2.9 Fiscalia	80
5.2.10 Relato	80
5.2.11 Selección final	80
5.3 Construcción de la Bolsa de Palabras	81
5.4 Construcción de los Datos	82
5.4.1 Importancia de las Palabras	82
Capítulo 6 - Modelado	88
6.1 Métrica de Evaluación	88
6.2 Construcción de los Modelos	89
6.2.1 K-Means	89
6.2.2 DBSCAN	93
6.2.3 Agrupamiento Jerárquico	97
6.3 Evaluación de los modelos	102
6.4 Modelo final	102
6.4.1 Cluster 0	104

6.4.2 Cluster 1	111
6.4.3 Cluster 2	118
6.4.4 Cluster 3	125
6.4.5 Cluster 4	132
Capítulo 7 - Evaluación	140
7.1 Evaluación de Resultados	140
7.1.1 Criterios de Éxito	140
7.1.2 Restricciones del Proyecto	141
7.2 Revisión del proyecto	141
7.2.1 Metodología	141
7.2.2 Fuentes de Datos y Conocimiento	141
7.2.3 Software para Minería de Texto	142
7.2.4 Gestión del Proyecto	142
Capítulo 8 - Conclusiones y Futuras Líneas de Investigación	143
8.1 Conclusiones	143
8.2 Futuras Líneas de Investigación	143
Bibliografía	145
Anexos	149
Anexo I: Vinculación de Google Drive con Google Colaboratory	149
Anexo II: Codificación	150

Índice de Figuras

Figura 1: Encuesta Nacional de Victimización en base a INDEC	10
Figura 2: Tasa de robos nacional y provincial entre los años 2001 y 2020	11
Figura 3: Tasa de hurtos por jurisdicción y por año entre 2014-2020	12
Figura 4: Fases de CRISP-DM	17
Figura 5: Iteraciones del algoritmo K-means	22
Figura 6: EDT del proyecto	31
Figura 7: Recursos necesarios para realizar las actividades del proyecto (primera parte)	36
Figura 8: Recursos necesarios para realizar las actividades del proyecto (segunda parte)	37
Figura 9: Diagrama de Gantt. Primera parte	38
Figura 10: Diagrama de Gantt. Segunda parte	39
Figura 11: Categorización de un riesgo de acuerdo a su valoración	45
Figura 12: Código de la lectura y visualización del dataset	54
Figura 13: Columnas obtenidas de la visualización del dataset	54
Figura 14: Función para tokenizar los datos	55
Figura 15: Uso de la función creada para tokenizar nuestros datos sobre la columna “Relato”	56
Figura 16: Función para la obtención de la frecuencia de cada término en una bolsa de palabras	57
Figura 17: Los 20 términos más frecuentes de la bolsa de palabras	57
Figura 18: Nueva lectura de dataset original	59
Figura 19: Valores faltantes detectados en el dataframe	60
Figura 20: Eliminación de denuncias con atributos nulos	60
Figura 21: Eliminación correcta de registros	61
Figura 22: Denuncias N° 35 y N°36	62
Figura 23: Relatos de las denuncias N° 35 y N°36	62
Figura 24: Función para clasificar los delitos en “Robo” y “Hurto”	63
Figura 25: Corroboración de correcta ejecución de la función de la Figura 29	63
Figura 26: Obtención de registros repetidos	63
Figura 27: Eliminación de denuncias duplicadas	64
Figura 28: Corroboración de una correcta eliminación de denuncias repetidas	64
Figura 29: Denuncias n°15 y n°16	64
Figura 30: Relatos de las denuncias n°15 y n°16	64
Figura 31: Función para eliminar denuncias repetidas	65
Figura 32: Denuncias correspondientes a otros años	66
Figura 33: Eliminación de denuncias de otros años	66
Figura 34: Creación de dataframe con los nombres a utilizar	67
Figura 35: Nombres y cantidad de nombres registrados por año	67
Figura 36: Obtención de nombres individuales	68
Figura 37: Verificación de valores nulos	69
Figura 38: Nombres repetidos en el dataset	69
Figura 39: Eliminación de nombres repetidos	70
Figura 40: Tokenización del dataframe que contiene a los nombres a utilizar	70
Figura 41: Obtención de stop words y muestra de algunas de ellas	71
Figura 42: Función para tokenizar, anonimizar y eliminar stop words	72
Figura 43: Ejecución de la función de limpieza de la Figura 42	72
Figura 44: Función utilizada para lematizar las denuncias	73
Figura 45: Denuncias luego de ser lematizadas	73
Figura 46: Reemplazo de datos del atributo “Relato”	74
Figura 47: Creación de función para tokenizar y eliminar las stop words	74
Figura 48: Eliminación de stop words de la columna “lugarHecho”	75
Figura 49: Reemplazo de datos del atributo “lugarHecho”	75
Figura 50: Nombres de los atributos del dataframe	76
Figura 51: Eliminación del atributo “fchConfirmación”	76
Figura 52: Eliminación del atributo “DEPENDENCIA”	76
Figura 53: Eliminación del atributo “actaNro”	76
Figura 54: Eliminación del atributo “Anio”	77
Figura 55: Creación de los atributos “MesHecho” y “DíaHecho”	77
Figura 56: Función para detectar día y mes de la columna “fchHrHecho” y agregarlos a las nuevas columnas	78

Figura 57: Resultado de la ejecución de la función de la Figura 56	79
Figura 58: Eliminación del atributo “fchHrHecho”	79
Figura 59: Renombramiento del atributo “lugarHecho”	79
Figura 60: Renombramiento del atributo “DELITO”	80
Figura 61: Eliminación del atributo “FISCALIA”	80
Figura 62: Renombramiento del atributo “RELATO”	80
Figura 63: Atributos seleccionados para efectuar el proceso de clustering	81
Figura 64: Función para obtener cada término de una bolsa de palabras y su frecuencia	81
Figura 65: Los 20 términos más frecuentes en la bolsa de palabras	82
Figura 66: Fórmula de “TF”	83
Figura 67: Fórmula de “IDF”	83
Figura 68: Creación de lista con los datos seleccionados de cada denuncia	84
Figura 69: Función para agregar los 100 bigramas más frecuentes a las denuncias	85
Figura 70: Función para agregar los 50 trigramas más frecuentes a las denuncias	86
Figura 71: Creación del vector de palabras	86
Figura 72: Coeficiente de silueta para una muestra	88
Figura 73: Función para utilizar el método del codo	90
Figura 74: Ejecución de la función de la Figura 73	91
Figura 75: Función para obtener el valor de “k” mediante su coeficiente de silueta	92
Figura 76: Ejecución de la función de la Figura 75	92
Figura 77: Coeficiente de silueta con “k=5” utilizando K-means	93
Figura 78: Cálculo de los vecinos más cercanos	94
Figura 79: Ordenamiento y graficado de los valores obtenidos del cálculo de la Figura 102	95
Figura 80: Función para encontrar el mejor modelo utilizando DBSCAN	96
Figura 81: Ejecución de la función de la Figura 80	97
Figura 82: Conversión del vector de palabras a formato array	98
Figura 83: Código para la creación del dendrograma a partir de nuestras denuncias	98
Figura 84: Dendrograma obtenido a partir del código de la Figura 109	99
Figura 85: Trazado de líneas horizontales para definir la cantidad de clusters	100
Figura 86: Función para encontrar el mejor modelo utilizando agrupamiento jerárquico	101
Figura 87: Ejecución de la función de la Figura 86	101
Figura 88: Ejecución del modelo K-means	103
Figura 89: Asignación de clusters para cada denuncia	103
Figura 90: Obtención de los términos más representativos de cada cluster	103
Figura 91: Creación de dataframe con denuncias del cluster 0	104
Figura 92: Obtención de los 20 términos más representativos del cluster 0	105
Figura 93: Cantidad y porcentaje de denuncias del cluster 0	105
Figura 94: Cantidades y porcentajes de denuncias del cluster 0 según el tipo de delito	106
Figura 95: Cantidades y porcentajes de denuncias del cluster 0 según el mes de ocurrencia del delito (primer cuatrimestre)	107
Figura 96: Cantidades y porcentajes de denuncias del cluster 0 según el mes de ocurrencia del delito (segundo cuatrimestre)	108
Figura 97: Cantidades y porcentajes de denuncias del cluster 0 según el mes de ocurrencia del delito (tercer cuatrimestre)	109
Figura 98: Cantidades y porcentajes de denuncias del cluster 0 según la quincena del mes en la que ocurrieron	110
Figura 99: Creación de dataframe con denuncias del cluster 1	111
Figura 100: Obtención de los 20 términos más representativos del cluster 1	112
Figura 101: Cantidad y porcentaje de denuncias del cluster 1	113
Figura 102: Cantidades y porcentajes de denuncias del cluster 1 según el tipo de delito	113
Figura 103: Cantidades y porcentajes de denuncias del cluster 1 según el mes de ocurrencia del delito (primer cuatrimestre)	114
Figura 104: Cantidades y porcentajes de denuncias del cluster 1 según el mes de ocurrencia del delito (segundo cuatrimestre)	115
Figura 105: Cantidades y porcentajes de denuncias del cluster 1 según el mes de ocurrencia del delito (tercer cuatrimestre)	116
Figura 106: Cantidades y porcentajes de denuncias del cluster 1 según la quincena del mes en la que ocurrieron	117
Figura 107: Creación de dataframe con denuncias del cluster 2	118
Figura 108: Obtención de los 20 términos más representativos del cluster 2	119

Figura 109: Cantidad y porcentaje de denuncias del cluster 2	120
Figura 110: Cantidades y porcentajes de denuncias del cluster 2 según el tipo de delito	120
Figura 111: Cantidades y porcentajes de denuncias del cluster 2 según el mes de ocurrencia del delito (primer cuatrimestre)	121
Figura 112: Cantidades y porcentajes de denuncias del cluster 2 según el mes de ocurrencia del delito (segundo cuatrimestre)	122
Figura 113: Cantidades y porcentajes de denuncias del cluster 2 según el mes de ocurrencia del delito (tercer cuatrimestre)	123
Figura 114: Cantidades y porcentajes de denuncias del cluster 2 según la quincena del mes en la que ocurrieron	1234
Figura 115: Creación de dataframe con denuncias del cluster 3	125
Figura 116: Obtención de los 20 términos más representativos del cluster 3	126
Figura 117: Cantidad y porcentaje de denuncias del cluster 3	127
Figura 118: Cantidades y porcentajes de denuncias del cluster 3 según el tipo de delito	127
Figura 119: Cantidades y porcentajes de denuncias del cluster 3 según el mes de ocurrencia del delito (primer cuatrimestre)	128
Figura 120: Cantidades y porcentajes de denuncias del cluster 3 según el mes de ocurrencia del delito (segundo cuatrimestre)	129
Figura 121: Cantidades y porcentajes de denuncias del cluster 3 según el mes de ocurrencia del delito (tercer cuatrimestre)	130
Figura 122: Cantidades y porcentajes de denuncias del cluster 3 según la quincena del mes en la que ocurrieron	131
Figura 123: Creación de dataframe con denuncias del cluster 4	132
Figura 124: Obtención de los 20 términos más representativos del cluster 4	133
Figura 125: Cantidad y porcentaje de denuncias del cluster 4	134
Figura 126: Cantidades y porcentajes de denuncias del cluster 4 según el tipo de delito	134
Figura 127: Cantidades y porcentajes de denuncias del cluster 4 según el mes de ocurrencia del delito (primer cuatrimestre)	135
Figura 128: Cantidades y porcentajes de denuncias del cluster 4 según el mes de ocurrencia del delito (segundo cuatrimestre)	136
Figura 129: Cantidades y porcentajes de denuncias del cluster 4 según el mes de ocurrencia del delito (tercer cuatrimestre)	137
Figura 130: Cantidades y porcentajes de denuncias del cluster 4 según la quincena del mes en la que ocurrieron	138

Índice de Tablas

Tabla 1: Recursos necesarios para realizar las actividades del proyecto	40
Tabla 2: Costos de los recursos humanos del proyecto	41
Tabla 3: Costos de gastos varios del proyecto	42
Tabla 4: Línea base de costos del proyecto	42
Tabla 5: Matriz de probabilidad e impacto del proyecto	45
Tabla 6: Comparación entre sistemas operativos	46
Tabla 7: Comparación de rendimiento de los distintos algoritmos	102
Tabla 8: Resumen del análisis cuantitativo del cluster 0	111
Tabla 9: Resumen del análisis cuantitativo del cluster 1	118
Tabla 10: Resumen del análisis cuantitativo del cluster 2	125
Tabla 11: Resumen del análisis cuantitativo del cluster 3	132
Tabla 12: Resumen del análisis cuantitativo del cluster 4	139

Abstract

La delincuencia y la inseguridad son unos de los principales problemas insertos en la sociedad argentina. El presente proyecto aplica el proceso de minería de texto para encontrar características comunes entre hechos delictivos, específicamente robos y hurtos, que permitan detectar los elementos presentes en estos delitos. Para ello fue necesario obtener denuncias penales de robos y hurtos, comprenderlos, prepararlos, elaborar el modelo de datos y analizar los resultados obtenidos. Con el objetivo de encontrar información certera sobre estos delitos y poder, de alguna manera, contribuir al combate y erradicación de estos delitos en la provincia de Salta. Como resultado del proyecto, se obtienen conclusiones a partir del análisis de los grupos de características asociadas entre elementos presentes en las denuncias obtenidas.

Capítulo 1 - Introducción

La inseguridad en Argentina es un problema que está muy inserto en la sociedad y es una de las mayores preocupaciones para la población, ya que se convive con ella permanentemente.

En la Figura 1, observamos un estudio realizado en 2016 por INDEC (Instituto Nacional de Estadística y Censos), al analizar los cinco delitos más frecuentes a nivel nacional, entre los que están el robo o hurto en la vivienda, el hurto personal, el robo con violencia a las personas, robo o hurto de autopartes y las amenazas.

Población de 18 años y más. Por provincias. Año 2016

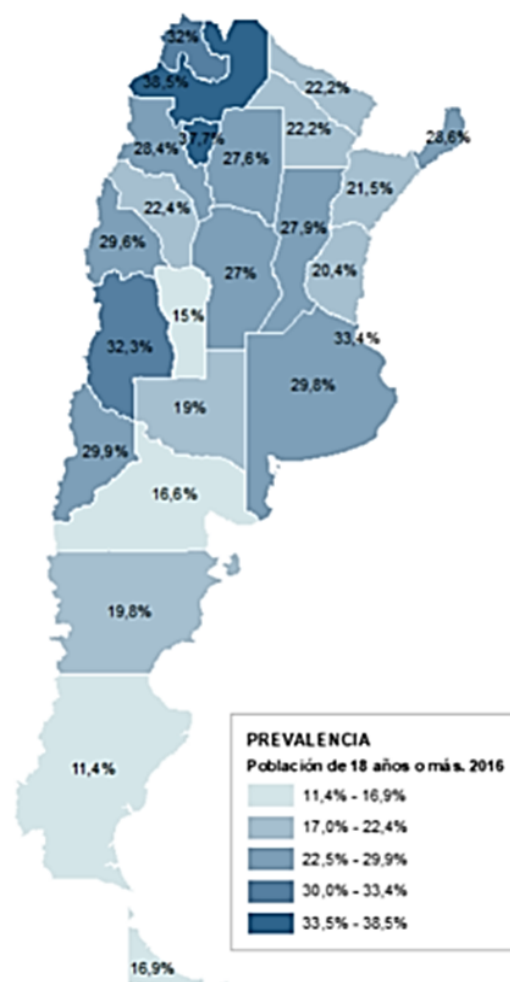


Figura 1: Encuesta Nacional de Victimización en base a INDEC (Observatorio de Seguridad Ciudadana, s.f.).

En cuanto a Salta, se aprecia que ese año fue la provincia con mayor índice delictivo en todo el país (38,5%).

Según el INDEC, el robo o hurto en la vivienda es el delito de mayor prevalencia (8.5%). Le siguen, en orden decreciente, el hurto personal (7.5%), el robo con violencia a las personas (6.2%), el robo o hurto de autopartes (4.3%), y las amenazas (4.0%) (Observatorio de Seguridad Ciudadana, s.f.). Como se observa, los robos y hurtos son los delitos más comunes en Argentina, por lo que combatir con este tipo de delitos reduciría en gran parte el índice delictivo en el país, y por consiguiente, la inseguridad.

En las siguientes Figuras 2 y 3, se muestra cómo se dio el constante crecimiento de la ocurrencia de robos y hurtos en la provincia de Salta, y del promedio en todo el país, a través de los últimos 20 años, apreciando claramente como la pandemia producida por el Covid-19 redujo la cantidad de delitos en 2020 por una cuestión del aislamiento provocado por esta enfermedad viral.



Figura 2: Tasa de robos nacional y provincial entre los años 2001 y 2020 (Observatorio de Seguridad Ciudadana, s.f.)



Figura 3: Tasa de hurtos por jurisdicción y por año entre 2014-2020 (Observatorio de Seguridad Ciudadana, s.f.).

La falta de información certera sobre los delitos es una de las tantas causas por las cuales no se logra contrarrestar esta gran problemática, por lo que este proyecto busca hacer un aporte valioso para esta lucha contra la inseguridad proporcionando esta información certera que permita tomar decisiones al respecto con la menor incertidumbre posible.

1.1 Descripción del Problema

La delincuencia y la seguridad se ven reflejadas en el nivel de denuncias efectuadas en la provincia de Salta (Argentina), en este proyecto se van a evaluar específicamente denuncias penales de robos y hurtos, que se conocen en el contexto de la justicia como delitos contra la propiedad, así que se va a proceder a definir y diferenciar estos conceptos:

- **Denuncia penal:** Es el acto mediante el cual, de buena fe y con motivos razonables, se pone en conocimiento de la autoridad judicial hechos que podrían constituir delitos (Argentina.gob.ar, s.f.).
- **Diferencia entre robo y hurto:** Los delitos de robo y hurto suceden, ambos, cuando alguien se apodera ilegítimamente de una cosa ajena. Sin embargo, el robo implica la utilización de la fuerza o la violencia en la apropiación ilegal de bienes ajenos, mientras que en el hurto no existe ningún tipo de violencia ni intimidación (Periódico Judicial, s.f.).

A fin de tratar de combatir y contrarrestar estos delitos y, por ende, la inseguridad y la delincuencia en la provincia, es necesario obtener información certera acerca de este tipo de

actos ilegales para poder tomar decisiones y precauciones correspondientes con la menor incertidumbre posible.

1.2 Motivación

La temática elegida para la realización de este proyecto, se debe al intento de contribuir con la lucha contra algunas de las problemáticas sociales presentes en la provincia de Salta, como son la inseguridad y la delincuencia.

La elección del uso de minería de texto como herramienta para obtener información útil y certera, corresponde a un interés personal acerca de esta herramienta, sobre todo, debido a la posibilidad que esta presenta para plasmar conclusiones obtenidas a partir de los resultados de la aplicación de un algoritmo a un conjunto de datos, en este caso, de denuncias penales contra la propiedad.

Se destaca la posibilidad de esta herramienta para encontrar relaciones y resultados a partir de datos, que en un principio quizás parecieran no ser influyentes o estar relacionados entre sí, pero que terminan contribuyendo a la obtención de información que permita extraer conclusiones ciertas y eficaces, y que estas puedan ser brindadas a los organismos correspondientes para el uso efectivo de las mismas.

En conclusión, resulta desafiante e interesante utilizar la tecnología de minería de texto para contribuir contra la lucha de esta problemática social presente en la sociedad salteña.

1.3 Organización del Trabajo

Este trabajo estará estructurado de la siguiente manera:

- En el Capítulo 2, se presenta el estado de la cuestión o marco teórico ligado a la solución del problema planteado.

Los siguientes capítulos corresponden a la aplicación de la metodología CRISP-DM al proyecto, excepto la fase de “Implantación”.

- El Capítulo 3 contiene la fase de “Comprensión del Negocio”. Se desarrolla un análisis del contexto y la problemática que dan origen al desarrollo del proyecto. También se establece la planificación para alcanzar los objetivos planteados.
- En el Capítulo 4, se desarrollan las actividades vinculadas a la fase de “Comprensión de los Datos”. En esta parte del proyecto se ejecutan las tareas de recolección de datos

y análisis de los mismos para realizar la selección de los atributos relevantes para el proyecto.

- En el capítulo 5, se realizan las actividades correspondientes a la fase 3 de “Preparación de los Datos”. El objetivo de estas tareas, es obtener el conjunto de datos que serán utilizados en la etapa de modelado. Para esto es necesario acondicionar los datos, es decir, realizar las operaciones de limpieza y generar la representación adecuada de los mismos para poder ser utilizados en la siguiente etapa.
- En el capítulo 6, se lleva a cabo el desarrollo de la fase de “Modelado”. Aquí, se preparan y ejecutan los modelos de clustering para agrupar el conjunto de datos.
- En el capítulo 7, se lleva a cabo la fase de “Evaluación”. Aquí se evalúan tanto los resultados obtenidos en el proyecto, así como, los distintos factores que afectaron al desarrollo del mismo.
- Finalmente, en el capítulo 9, se presentan las “Conclusiones y Futuras Líneas de Investigación” obtenidas a partir del desarrollo del proyecto.

Capítulo 2 - Estado de la cuestión

2.1 Minería de Datos

Si bien el desarrollo de este proyecto se basará en el uso de minería de texto, es importante primero entender el concepto de minería de datos o *datamining*, una definición bastante acertada de esta es que se define como el proceso de extraer conocimiento útil y comprensible, previamente desconocido, desde grandes cantidades de datos almacenados en distintos formatos. Es decir, la tarea fundamental de la minería de datos es encontrar modelos inteligibles a partir de los datos.

Para que este proceso sea efectivo, debería ser automático o semi-automático (asistido) y el uso de los patrones descubiertos debería ayudar a tomar decisiones más seguras que reporten, por tanto, algún beneficio a la organización (Hernández Orallo, Ferri Ramírez, & Ramírez Quintana, 2004). Este proceso siempre se realiza teniendo en cuenta que la minería de datos trata de resolver problemas (Witten et al., 2016).

El *datamining* surge para intentar ayudar a comprender el contenido de un repositorio de datos. Con este fin, hace uso de prácticas estadísticas y, en algunos casos, de algoritmos de búsqueda próximos a la Inteligencia Artificial y a las redes neuronales.

De forma general, los datos son la materia prima bruta. En el momento que el usuario les atribuye algún significado especial pasan a convertirse en información. Cuando los especialistas elaboran o encuentran un modelo, haciendo que la interpretación que surge entre la información y ese modelo represente un valor agregado, entonces nos referimos al conocimiento (Sinnexus, 2007).

2.2 Metodología de minería de datos: CRISP-DM

CRISP-DM de Cross Industry Standard Process for Data Mining es un modelo de proceso de minería de datos. Está compuesto por seis fases (Chapman, et. al., 2000):

1. **Comprensión del negocio:** Esta fase inicial se enfoca en la comprensión de los objetivos de proyecto y exigencias desde una perspectiva de negocio, luego

convirtiendo este conocimiento de los datos en la definición de un problema de minería de datos y en un plan preliminar diseñado para alcanzar los objetivos.

2. **Comprensión de los datos:** La fase de entendimiento de datos comienza con la colección de datos inicial y continua con las actividades que le permiten familiarizarse primero con los datos, identificar los problemas de calidad de datos, descubrir los primeros conocimientos en los datos, y/o descubrir subconjuntos interesantes para formular hipótesis en cuanto a la información oculta.
3. **Preparación de los datos:** La fase de preparación de los datos cubre todas las actividades necesarias para construir el conjunto de datos final (los datos que serán provistos en las herramientas de modelado) de los datos en brutos iniciales. Las tareas de preparación de datos probablemente van a ser realizadas muchas veces y no en cualquier orden prescripto. Las tareas incluyen la selección de tablas, registros, y atributos, así como la transformación y la limpieza de datos para las herramientas que modelan.
4. **Modelado:** En esta fase, varias técnicas de modelado son seleccionadas y aplicadas, y sus parámetros son calibrados a valores óptimos. Típicamente hay varias técnicas para el mismo tipo de problema de minería de datos. Algunas técnicas tienen requerimientos específicos sobre la forma de datos. Por lo tanto, volver a la fase de preparación de datos es a menudo necesario.
5. **Evaluación:** Antes de proceder al despliegue final del modelo, es importante evaluar a fondo ello y la revisión de los pasos ejecutados para crearlo, para comparar el modelo correctamente obtenido con los objetivos de negocio. Un objetivo clave es determinar si hay alguna cuestión importante de negocio que no ha sido suficientemente considerada. En el final de esta fase, una decisión en el uso de los resultados de minería de datos debería ser obtenida.
6. **Implantación:** Dependiendo de los requerimientos, la fase de implantación puede ser tan simple como la generación de un informe o tan compleja como la realización repetida de un proceso cruzado de minería de datos a través de la empresa. En muchos casos, es el cliente, no el analista de datos, quien lleva el paso del desarrollo. Sin embargo, incluso si el analista realizara el esfuerzo de despliegue, esto es importante para el cliente para entender de frente qué acciones necesita para ser ejecutadas en orden para hacer uso de los modelos creados actualmente.

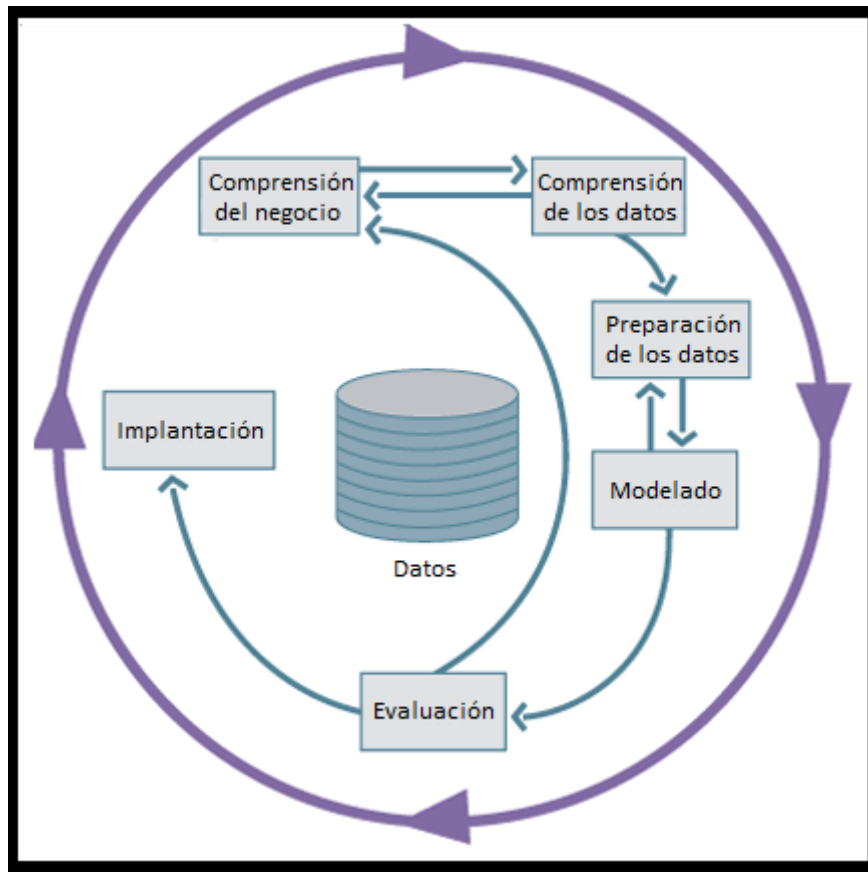


Figura 4: Fases de CRISP-DM (Chapman et al, 2000)

2.3 Minería de Texto

Una vez aclarado qué es la minería de datos, se va a proceder a definir lo que es la minería de texto o *text mining* en sí. La minería de textos se la puede entender como el análisis de textos con el fin de extraer información a partir de los mismos. Es decir, una tecnología cuyo objetivo es la búsqueda de conocimiento en grandes colecciones de documentos. A simple vista, el proceso de text mining no se diferencia de la definición de datamining. No obstante, la minería de textos es una variante de la minería de datos, adoptando sus técnicas de aprendizaje automático para reconocer patrones y comprender nueva información (Master Data Scientist, s.f.).

En la minería de texto se obtiene información nueva a partir de grandes cantidades de texto. Pero, en estos datos, la información no suele estar estructurada. Por su parte, en la minería de datos, el conocimiento se obtiene a partir de las bases de datos, donde la información está estructurada.

Por lo tanto, en la minería de textos los datos a tratar serán los documentos y textos de las organizaciones, en lugar de los datos de las bases de datos. Llevando a cabo así un análisis de los datos compartidos por todos los textos de la colección (Master Data Scientist, s.f.).

2.3.1 Fases de la Minería de Textos

Se puede decir que la minería de textos se encarga de descubrir información que no existía previamente de forma explícita, pero que surge de relacionar el contenido de varios de ellos. Para realizar este proceso, el text mining comprende tres actividades:

- 1. Recuperación:** Selección de los textos oportunos.
- 2. Extracción:** Extracción de la información incluida en esos textos mediante el procesamiento del lenguaje natural: acontecimientos, datos clave, hechos concretos, relaciones entre ellos, etc.
- 3. Minería de datos:** Encontrar asociaciones entre los datos clave previamente extraídos de entre los textos.

Estas actividades se dividen en tres fases:

- **Fase de pre-procesamiento:** Los textos se transforman en algún tipo de representación estructurada o semiestructurada que facilite su análisis posterior. El primer paso dentro de la minería de texto sería definir el conjunto de documentos; además se debe evitar la duplicación de documentos dentro del corpus.

Con el conjunto de documentos seleccionado y estructurado, se deben reconocer los tokens (unidades gramaticales esenciales), lo que implica representar el texto como una lista de palabras mediante una representación vectorial.

- **Etapas de descubrimiento:** Las representaciones internas se analizan para descubrir patrones interesantes o nueva información.
- **Etapas de visualización:** Los usuarios pueden observar y explorar los resultados de una forma amena y sencilla (Big Data Advanced Analytics, s.f.).

2.3.2 Las Aplicaciones de la Minería de Texto

La minería de textos puede ser encontrada en campos diferentes, como serían el caso de la medicina, el análisis de opiniones, la filología, la biología, etcétera. No obstante, para

que estos textos fueran publicados han tenido que pasar por un análisis de minería. Las cinco principales aplicaciones del text mining que podemos encontrar son las siguientes:

- **La búsqueda de información:** Son las búsquedas que se hacen a partir de una pregunta, palabra concreta o contenido interesante. Por ejemplo, se puede encontrar esta aplicación en Google y su motor de búsqueda del contenido que se puede hallar en Internet.
- **El reconocimiento de entidades mencionadas y referencias:** Se utilizan estadísticas, nombres de lugares, personas, entidades, etcétera. Con este análisis se pretende agrupar una cantidad de textos que tengan la misma palabra o información, como sería el caso de los *trending topic* (temas de tendencia) de Twitter.
- **Clustering:** Se agrupan los textos que reúnen criterios similares que no pueden ser vistos directamente. Sería el caso de reunir los libros por estilo de redacción o vocabulario utilizado.
- **Clasificación:** Se trata de etiquetar los textos para dividirlos en categorías de forma más eficaz. Por poner un ejemplo se pueden ver los documentos de texto de Facebook que están agrupados por ofensivos o no ofensivos, contenido violento, etcétera.
- **Análisis de sentimientos:** En base a los resultados estadísticos se puede saber si un determinado texto es positivo, negativo o incluso neutral. En este caso se podría hablar de Tripadvisor, una app donde se puede acceder a los comentarios sobre restaurantes, hoteles etcétera. A través del análisis de texto se pueden ver cuántos comentarios son positivos y cuántos negativos (Master Data Scientist, s.f.).

2.3.3 Tareas en la Minería de Texto

Este proceso nos permite realizar las siguientes tareas definidas a grandes rasgos:

- **Agrupamiento de documentos:** Para organizar los documentos en torno a una jerarquía basándose en alguna medida de similitud.
- **Identificación de categorías:** Extracción de términos significativos.
- **Categorización:** Asignar una o más categorías a un documento.
- **Clasificación:** Asignar una (y sólo una) clase a un documento.
- **Asociaciones:** Generalmente entre conceptos más que entre palabras (Hernandez Orallo, Ferri Ramírez, & Ramírez Quintana, 2004).

Para este proyecto en particular se buscará realizar un agrupamiento de documentos (Clustering) para demarcar las características comunes de los mismos.

2.4 Agrupamiento

El proceso de agrupamiento o *clustering* consiste en la división de los datos en grupos de objetos similares, para medir la similitud entre objetos suelen utilizar diferentes formas de distancia como Euclídea, Manhattan, Maximum, Minkowski, etc. El clustering juega un papel muy importante en aplicaciones de minería de datos, tales como exploración de datos científicos, recuperación de la información y minería de texto, aplicaciones sobre bases de datos en aplicaciones web (Garre, Cuadrado, Sicilia, Rodríguez,& Rejas, 2007).

El análisis de agrupamiento encuentra clusters de objetos de datos que son similares en un sentido a otro. Los miembros de un conglomerado se parecen más unos a otros de lo que son como miembros de otros grupos. El objetivo del análisis de agrupamiento es encontrar agrupaciones de alta calidad de tal manera que la semejanza entre los distintos clústeres sea baja y la similitud entre los datos del mismo grupo sea alta (Sanabria Ruiz, 2017)

2.4.1 Agrupamiento de Documentos

El agrupamiento de documentos, también conocido como *clustering de texto* descubre automáticamente la estructura implícita en una colección de documentos, identificando los temas más frecuentes dentro de la colección y distribuyendo los documentos en varios grupos o clusters. Esta distribución se caracteriza por maximizar la similitud entre los elementos de un mismo grupo y a la vez maximizar las diferencias entre los grupos (Jing, 2008).

2.4.2 Tipos de Clustering

Aquí se va a realizar una distinción de los distintos tipos de clustering.

2.4.2.1 Jerárquico vs. Particional

Un “clustering particional” es simplemente una división de un conjunto de datos en *clusters* (grupos o conglomerados) no superpuestos de modo que cada objeto de datos esté en exactamente un subconjunto.

Mientras que, si se permiten a los clusters tener sub-clusters se obtendrá un “clustering jerárquico”, el cuál es un conjunto de clusters anidados que están organizados como un árbol. Cada nodo (cluster) en el árbol es la unión de sus hijos (sub-clusters), y la raíz del árbol es el cluster que contiene todos los objetos (Tan, Steinbach, & Kumar, 2005).

2.4.2.2 Exclusivo vs. Superpuestos vs. Difuso

Los “clustering exclusivos” asignan cada objeto a un solo cluster. Hay muchas situaciones en las que podría ser razonable ubicar un punto en más de un cluster, y en estas situaciones es mejor utilizar “clusterings no exclusivos”. En un sentido más general, se usan los clustering superpuestos o no exclusivos para poder reflejar el hecho de que objeto puede pertenecer simultáneamente a más de un grupo.

En un “clustering difuso” cada objeto pertenece a todos los cluster con un peso de membresía que se encuentre entre 0 (absolutamente no pertenece) y 1 (absolutamente pertenece). En otras palabras, los clusters son tratados como conjuntos difusos (Tan, Steinbach, & Kumar, 2005).

2.4.2.3 Completo vs. Parcial

Un “clustering completo” asigna cada objeto a un cluster, mientras que un “clustering parcial” no. La motivación para un clustering parcial es que algunos objetos en el conjunto de datos puede no pertenecer a grupos bien definidos (Tan, Steinbach, & Kumar, 2005).

2.4.3 Algoritmos de Clustering

A continuación se explicarán algunos de los algoritmos de clustering más utilizados.

2.4.3.1 K-Means

Este algoritmo particiona los “n” objetos en “k” particiones (“k” siendo un valor arbitrario) en donde un objeto irá al cluster con la media más cercana. El algoritmo asigna “k” centros aleatoriamente, luego asigna los objetos al centro más cercano. El centro se re-calcula como la media de los puntos que tiene asignado, una vez actualizado se vuelven a reasignar los objetos al más cercano y así hasta tener convergencia (Blanco,& Sanz, 2016).

Este algoritmo depende mucho de la asignación inicial de los centros, nos puede dar un resultado u otro por lo que es mejor hacer varias pruebas con diferentes valores. Una

variante llamada “K-means++” intenta resolver este problema al escoger mejores centros. En la Figura 5, se puede observar cómo se efectúan las iteraciones en el algoritmo.

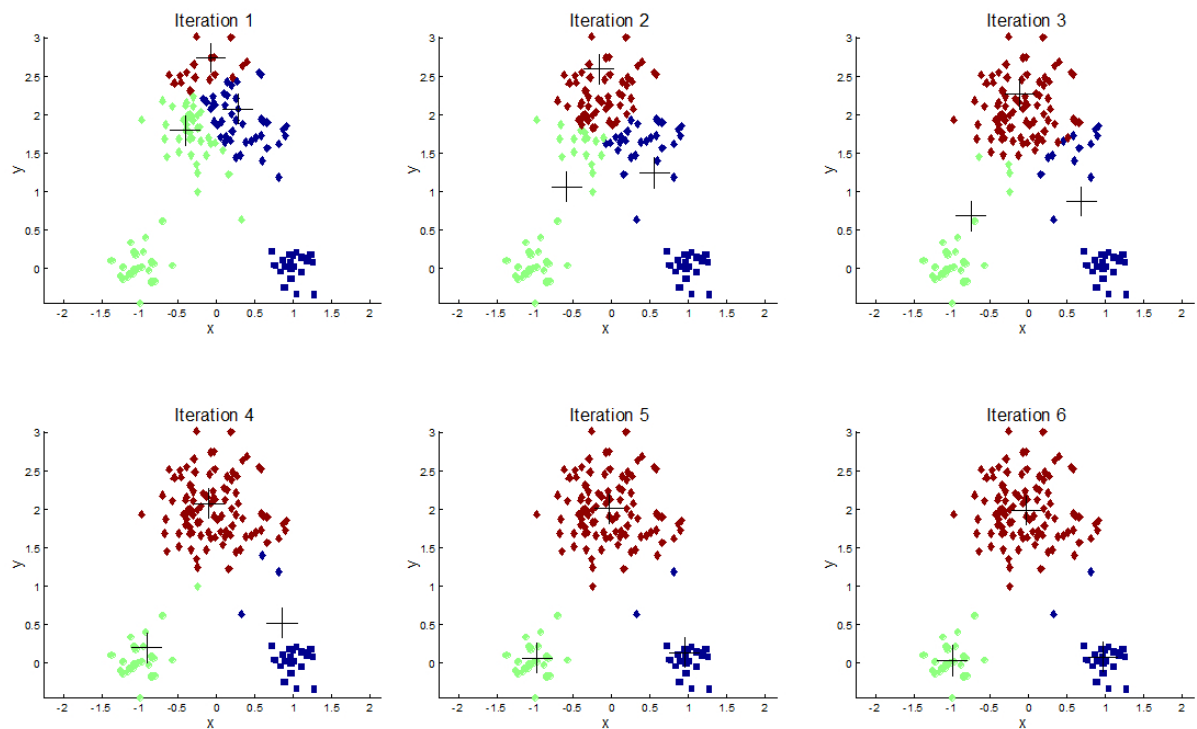


Figura 5: Iteraciones del algoritmo K-means (Blanco & Sanz, 2016).

En la primera iteración se ven tres cruces correspondientes a los tres centros iniciales. Se han asignado los puntos al centro más cercano, es visible gracias a los colores. Tras la asignación, se calcula la media de los puntos asignados al cluster y se actualiza el centroide con este valor. Se puede ver en la iteración dos como se han desplazados los centroides y se han reasignado los puntos al centro más cercano. Se siguen realizando estos pasos hasta que las asignaciones de los puntos no cambien.

La principal desventaja de este algoritmo es que hay que especificar el número de clusters que queremos que forme, y dependiendo de este número los resultados varían (Blanco,& Sanz, 2016).

2.4.3.2 Agglomerative Hierarchical Clustering

El agrupamiento jerárquico es un tipo de algoritmo de aprendizaje automático no supervisado, este tipo de aprendizaje es un método donde un modelo es ajustado a las observaciones. Se distingue del aprendizaje supervisado por el hecho de que no hay un

conocimiento a priori, sino que trabaja con datos no identificados y no clasificados, a partir de estos datos, el algoritmo calcula los patrones y similitudes y crea varios grupos (Espino, Mur, & de Miguel, 2004).

Al igual que el agrupamiento de “K-Means”, el agrupamiento jerárquico también agrupa los puntos de datos con características similares.

Este enfoque de agrupación, se refiere a una colección de técnicas de agrupación estrechamente relacionadas, que producen una agrupación jerárquica al comenzar con cada punto como un grupo único, y luego fusionar repetidamente los dos grupos más cercanos hasta que queda un grupo único que lo abarca todo. (Tan, Steinbach, & Kumar, 2005).

Hay dos tipos de agrupamiento jerárquico, el “Aglomerativo” y el “Divisivo”.

En el primero, los puntos de datos se agrupan utilizando un enfoque de abajo hacia arriba que comienza con puntos de datos individuales.

En el segundo, se sigue un enfoque de arriba hacia abajo donde todos los puntos de datos se tratan como un gran grupo y el proceso de agrupamiento implica dividir al único grupo en varios grupos pequeños.

Los pasos involucrados en la agrupación aglomerativa son los siguientes:

1. Al principio, se trata cada punto de datos como un grupo o cluster. Por lo tanto, el número de grupos al principio será “k”, mientras que “k” es un número entero que representa el número de puntos de datos.
2. Se forma un grupo uniendo los dos puntos de datos más cercanos que dan como resultado grupos “k-1”.
3. Se forman más grupos uniendo los dos grupos más cercanos que dan como resultado grupos “k-2”.
4. Se repiten los tres pasos anteriores hasta que se forme un gran cluster.
5. Una vez que se forma un solo cluster, se hace uso de los dendrogramas para poder dividir en varios clusters según el problema (Malik, s.f.).

2.4.3.3 DBSCAN

El propio nombre “DBSCAN” indica cómo funciona el algoritmo. DBSCAN viene de las palabras *Density Based Spatial Clustering of Applications*, es decir, que se trata de un algoritmo de clusterización basado en la densidad.

En este sentido, la densidad se refiere al número de observaciones que se disponen de una misma zona. Sin embargo, el concepto de densidad es un poco subjetivo. Porque, ¿cuántos puntos son necesarios para considerar a una zona como densa? Y, ¿qué tan lejos pueden estar estos puntos entre sí?

Precisamente tanto el número de puntos para considerar una zona como densa, así como la distancia a la que se encuentran esos puntos son los dos parámetros del modelo.

El funcionamiento del algoritmo DBSCAN se basa en clasificar las observaciones en tres tipos:

- **Puntos core:** Son aquellos puntos que cumplen con las condiciones de densidad que se haya fijado.
- **Puntos alcanzables:** Son aquellos puntos que, aún no cumplen con las condiciones de densidad, pero tienen cerca otros puntos core.
- **Ruido:** son los puntos que no cumplen con las condiciones de densidad y, además, en su radio no tienen otros puntos.

Así, DBSCAN se basa en calcular la matriz de distancias entre los distintos puntos. Generalmente se utiliza la distancia Euclídea, aunque se pueden usar otras.

Teniendo en cuenta los parámetros del modelo, clasifica a cada punto entre punto core, punto frontera y ruido. En este sentido, puede que salgan diferentes puntos core ya que puede haber varias zonas de densidad. Cada uno de esos puntos core pertenece a un cluster. Asigna los núcleos alcanzables de cada cluster al cluster correspondiente (Fernández Jauregui A., s.f.).

2.5 Proyectos Similares

Se han encontrado muchos trabajos referidos al uso de minería de texto y/o de datos aplicada para el análisis de hechos criminales o delictivos, aunque no se encuentran entre ellos proyectos referidos específicamente al análisis de denuncias penales de delitos contra la propiedad. A continuación, se muestran algunos de los proyectos que se consideraron más destacados y que mayor aporte pueden proveer a este proyecto de grado:

Título: “*APLICACIÓN DE MINERÍA DE DATOS PARA LA EXPLORACIÓN Y DETECCIÓN DE PATRONES DELICTIVOS EN ARGENTINA*”

Autor/es: Perversi, Ignacio

Año de publicación: 2007

Resumen: Esta tesis se centra en la identificación y detección de patrones de homicidios dolosos cometidos en Argentina en 2005 aplicando Minería de Datos, específicamente una tarea de clustering.

Título: *“Data Mining Approaches to Criminal Career Analysis”*

Autor/es: Jeroen S. de Bruin, Tim K. Cocx, Walter A. Kusters, Jeroen F. J. Laros and Joost N. Kok

Año de publicación: 2006

Resumen: Este paper mediante el uso de minería de datos aplicados sobre bases de datos policiales en los Países Bajos, utilizando clustering busca determinar y caracterizar las carreras criminales de las personas que efectúan dichos delitos.

Título: *“Using Formal Concepts Analysis Techniques in Mining Data from Criminal Databases and Profiling Events Based on Factors to Understand Criminal Environments”*

Autor/es: Quist-Aphetsi Kester

Año de publicación: 2016

Resumen: En este proyecto se hace uso de la minería de datos para detectar los factores directos e indirectos que crean un entorno propicio para las actividades delictivas en Ghana.

Título: *“Data Mining in Crime Analysis”*

Autor/es: Nahid Jabeen and Parul Agarwal

Año de publicación: 2021

Resumen: Este proyecto se encarga de realizar el proceso de minería de texto para el análisis de informes de delitos e identificación de los datos tales como patrones, series, estadísticas, mapas, etc. Busca relacionar estos datos para

correlacionar delitos y buscar patrones que puedan identificarse para hacer la predicción de los mismos en India.

Capítulo 3 - Comprensión del Negocio

Como ya se definió en el capítulo anterior (ver sección 2.2), esta fase inicial se enfoca en la comprensión de los objetivos de proyecto y exigencias desde una perspectiva de negocio, luego convirtiendo este conocimiento de los datos en la definición de un problema de minería de datos y en un plan preliminar diseñado para alcanzar los objetivos.

3.1 Definición del Problema

Los conceptos de delincuencia e inseguridad son términos que están muy ligados entre sí, ya que, a medida que aumenta la delincuencia aumenta la inseguridad. Como ya vimos en el capítulo de introducción, la delincuencia posee un crecimiento constante en los últimos años, de acuerdo a los datos brindados por INDEC, haciendo una salvedad del año 2020 que afectó considerablemente los datos estadísticos de delincuencia debido a la pandemia provocada por el covid-19. Por este motivo los datos del año 2020 no van a ser utilizados en este proyecto, ya que, se considera que no representan la realidad en condiciones normales.

También, según datos del INDEC, que se pudieron observar en el capítulo de introducción, los cinco delitos más cometidos corresponden a robos y hurtos de distinta índole, pero robos y hurtos al fin, por lo que disminuir el número de estos hechos delictivos en específico reduciría en gran medida tanto el índice delictivo como la inseguridad plasmada en la sociedad argentina.

Poder tomar medidas para contrarrestar este mal tan inserto de la sociedad argentina, se hace muy difícil por falta de información de calidad disponible para las personas encargadas de tomar decisiones, ya que, la incertidumbre es el principal enemigo a la hora de decidir.

Particularmente, la provincia de Salta, que es de donde se extraerán los datos para llevar a cabo el proyecto, en 2019 fue la segunda provincia con más hurtos en el país y la sexta con más robos según datos brindados por INDEC. Se puede observar claramente que la sociedad salteña es una de las principales víctimas de este tipo de delitos.

3.2 Solución Propuesta

El presente proyecto busca dar solución a la problemática planteada haciendo uso de los conceptos, conocimientos y aplicaciones de minería de texto, específicamente utilizando la técnica descriptiva llamada “Agrupamiento” o “Clustering”.

Para aplicar esta solución, el proyecto se llevará a cabo de acuerdo a la metodología CRISP-DM, donde las fases y tareas se adaptarán a las necesidades del proyecto.

Los datos de entrada del proyecto serán el contenido de las denuncias penales de robo y hurto, pertenecientes a los años 2018 y 2019, brindadas por el Ministerio Público Fiscal, y por otro lado, las salidas serán las características comunes encontradas en las denuncias mediante el proceso de minería de texto.

3.3 Objetivos del Proyecto

3.3.1 Objetivo General

Desarrollar un proyecto de minería de texto para obtener información certera sobre los delitos contra la propiedad en la provincia de Salta, haciendo uso de las denuncias penales de delitos contra la propiedad (robos y hurtos) proporcionados por el Ministerio Público Fiscal de la Provincia de Salta.

3.3.2 Objetivos Específicos

- Descubrir características comunes entre los hurtos y robos analizados.
- Aplicar un proceso de minería de texto a un caso concreto.
- Presentar conclusiones útiles en el contexto elegido.

3.4 Criterios de Éxito del Proyecto

Los criterios de éxito de este proyecto son:

- Obtener características comunes de los grupos obtenidos a partir de los delitos analizados.
- Comparar el rendimiento de los distintos modelos de agrupamiento (clustering) empleados, a partir de la métrica de evaluación elegida (coeficiente de silueta).
- Obtener un coeficiente de silueta (métrica de evaluación utilizada para evaluar el rendimiento de los modelos planteados) superior a 0 en el modelo elegido como final.

- Obtener información certera a partir del análisis de los resultados obtenidos del proceso de minería de texto.

3.5 Evaluación de la Situación

La evaluación de la situación se encarga de determinar los factores que van a afectar en el desarrollo del proyecto. Tanto en este proyecto, como en cualquier proyecto de Minería de Texto, es fundamental contar con un conjunto de datos inicial, así como los conocimientos técnicos necesarios para llevar a cabo todas las actividades correspondientes de manera eficaz. Estos y otros factores serán detallados en este inciso.

3.5.1 Fuentes de Datos y Conocimientos

La fuente de datos para este proyecto serán las denuncias penales de robo y hurto, brindadas por el Ministerio Público Fiscal de la Provincia de Salta, que se encuentran comprendidas entres los años 2018 y 2019.

En cuanto a conocimientos, las fuentes serán por un lado el juicio de expertos que se consideren de utilidad a los fines de este proyecto y, por otro lado, la bibliografía colocada en el inciso del proyecto de grado del mismo nombre. Cabe aclarar que con juicio de expertos se hace referencia a los conocimientos y/o experiencias aportados por una persona o grupo de personas dúctiles en los temas que competen a este proyecto de grado.

3.5.2 Requerimientos, Presunciones y Restricciones

3.5.2.1 Requerimientos

- **Gestionar el alcance del proyecto:** Realizar las acciones requeridas para garantizar que el proyecto incluya todas las actividades necesarias para llevarse a cabo correctamente.
- **Gestionar el tiempo del proyecto:** Realizar la correcta planificación y ejecución adecuada de las tareas para garantizar que el proyecto concluya en el tiempo estimado.
- **Gestionar los costos del proyecto:** Estimar, presupuestar y controlar los costos necesarios para concretar el proyecto.
- **Gestionar los recursos humanos del proyecto:** Adquirir, desarrollar y dirigir el equipo empleado en el proyecto.

- **Gestionar los riesgos del proyecto:** Identificar y dimensionar los riesgos, para luego determinar las acciones que se llevarán a cabo para atenuar o mitigar estos riesgos.
- **Gestionar las adquisiciones del proyecto:** Efectuar y administrar las adquisiciones de productos o servicios.
- **Gestionar las comunicaciones del proyecto:** Generar una comunicación eficiente entre los diferentes interesados del proyecto.

3.5.2.2 Presunciones

- Se presume que el conjunto de datos brindados por la fuente de datos (Ministerio Público Fiscal de la Provincia de Salta) no cuenta con denuncias falsas, es decir, que no contengan datos falsos.

3.5.2.3 Restricciones

- El proyecto será realizado con el equipamiento de hardware disponible, el cual cuenta con las siguientes características:
 - Procesador Intel(R) Core(TM) i7-5500U CPU 2.40 GHz.
 - Memoria RAM 16 GB.
 - Disco Rígido Sata 500 GB.
 - Sistema Operativo Windows Home 10 64bits.
 - Placa de Video NVIDIA GeForce 920M.
 - Unidad de Disco Samsung SSD 850 EVO 1TB

3.6 Plan de Proyecto

3.6.1 Alcance del Proyecto

- Transformación de las denuncias penales de hurtos y robos de la provincia de Salta comprendidos entre los años 2018-2019, a un formato correcto para llevar a cabo la minería de texto.
- Preparación del software requerido para poder llevar a cabo las actividades del proyecto, así como el aprendizaje necesario para poder hacer uso del mismo eficientemente.
- Desarrollo del proyecto de minería de texto con las respectivas etapas de la metodología CRISP-DM, excepto de la etapa de implantación, debido a que este proyecto únicamente se encargará de obtener información certera de las denuncias analizadas pero no se encargará de hacer uso de la misma. Por ende, las etapas que se realizarán son las siguientes:

- Comprensión del negocio.
- Comprensión de los datos.
- Preparación de los datos.
- Modelado.
- Evaluación.
- Presentación de los resultados obtenidos mediante una tabla informativa por cada cluster obtenido.
- Análisis de los resultados obtenidos.
- Elaboración de conclusiones basadas en el análisis realizado.

3.6.2 Estructura de Desglose del Trabajo (EDT)

La Estructura de Desglose del Trabajo (EDT) es una estructura jerárquica que contiene todos los paquetes de trabajo que deben realizarse para completar el proyecto. Esta estructura se determinó a partir de las fases correspondientes a la metodología CRISP-DM (ver sección 2.2).

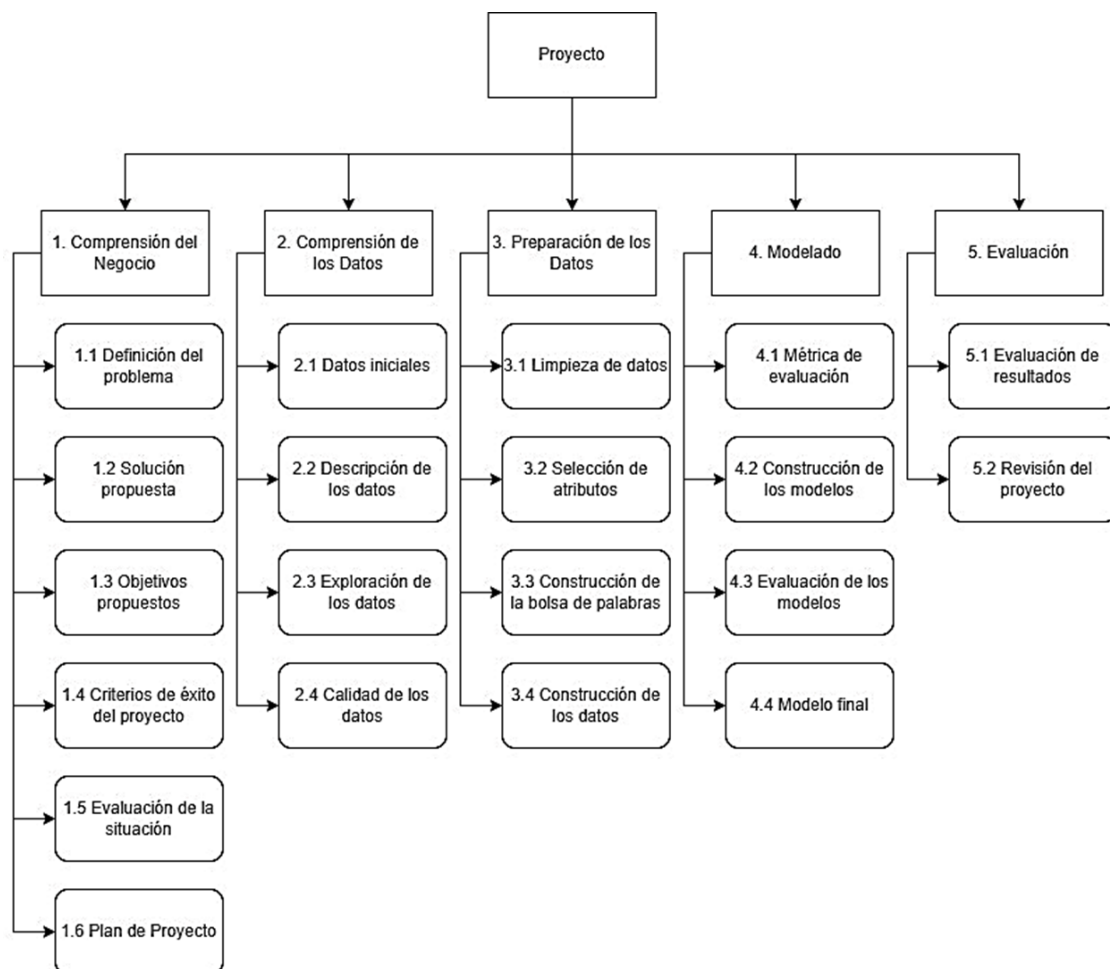


Figura 6: EDT del proyecto.

3.6.3 Diccionario de la EDT

En esta sección se presenta el diccionario de la EDT, su objetivo es proporcionar una descripción de los paquetes de trabajo definidos en la Estructura de Desglose de Trabajo (EDT).

1 Comprensión de los datos: Esta fase inicial se enfoca en la comprensión de los objetivos de proyecto y exigencias desde una perspectiva de negocio, luego convirtiendo este conocimiento de los datos en la definición de un problema de minería de texto y en un plan preliminar diseñado para alcanzar los objetivos.

1.1 Definición del problema: Determinar el contexto y definir el problema que se desea resolver.

1.2 Solución propuesta: Proponer una solución viable al problema planteado.

1.3 Objetivos del proyecto: Determinar el objetivo general del proyecto, y sus objetivos específicos.

1.4 Criterios de éxito del proyecto: Plasmar cuáles serán aquellos criterios de los que dependerá la evaluación de éxito de la realización del proyecto en su totalidad.

1.5 Evaluación de la situación: Investigación más detallada sobre recursos, requerimientos, restricciones, presunciones y otros factores a ser considerados en el desarrollo del proyecto.

1.6 Plan de proyecto: Elaboración de un plan de acción para alcanzar objetivos de negocio, gestionando el alcance, tiempo, riesgos, costos, adquisición y analizar las factibilidades del proyecto.

2 Comprensión de los datos: Esta fase comienza con la colección de datos inicial y continúa con las actividades que permiten la familiarización de los mismos, identificando problemas de calidad y descubriendo los primeros conocimientos en los datos.

2.1 Datos iniciales: Recolección de datos iniciales.

2.2 Descripción de los datos: Examinar propiedades de los datos.

2.3 Exploración de los datos: Conocer los datos para tomar decisiones adecuadas al crear los modelos.

2.4 Calidad de los datos: Determinar si los datos requieren de una limpieza para ser procesados por los modelos de clasificación.

3 Preparación de los datos: La fase de preparación de datos cubre todas las actividades necesarias para construir el conjunto de datos final, el cual será provisto para el entrenamiento y prueba de los modelos.

3.1 Limpieza de datos: Realizar tareas de limpieza a los datos, para elevar la calidad de los mismos al nivel requerido para la aplicación del modelado.

3.2 Selección de atributos: Seleccionar los datos que se utilizarán para llevar a cabo el modelo de agrupamiento elegido.

3.3 Construcción de la bolsa de palabras: Creación de la bolsa de palabras utilizando los datos seleccionados y después de haber pasado por el respectivo proceso de limpieza

3.4 Construcción de los datos: Operaciones para representar cada texto del conjunto de datos, de una manera particular para la aplicación del modelado.

4 Modelado: En esta fase se realiza la construcción y evaluación de los modelos de agrupamiento, así como también se debe producir una vista adecuada de los resultados.

4.1 Métrica de evaluación: Se debe determinar cuáles son las medidas de rendimiento que se tendrán en cuenta para la evaluación de los modelos.

4.2 Construcción de los modelos: Ejecutar los algoritmos de modelado sobre el conjunto de datos preparados para crear los modelos de agrupamiento.

4.3 Evaluación de los modelos: Examinar las métricas de evaluación de rendimientos obtenidos de los modelos y determinar cuál es el óptimo.

4.4 Modelo final: Analizar los resultados obtenidos a partir del modelo elegido como óptimo.

5 Evaluación: Consiste en realizar una evaluación sobre los resultados obtenidos del proyecto, y revisar los aspectos generales del mismo.

5.1 Evaluación de resultados: Evaluar los resultados obtenidos por el modelo de agrupamiento, y el nivel de respuesta a los objetivos de negocio planteados anteriormente.

5.2 Revisión del proyecto: Evaluar aspectos del proyecto como planificaciones y estimaciones, para generar una crítica sobre lo que se realizó correctamente y lo que se debería mejorar.

3.6.4 Lista de Actividades

Las siguientes actividades definen las acciones a realizar para llevar a cabo los paquetes de trabajo definidos en la EDT.

- Fase 1: Comprensión del negocio
 - Definir el problema.
 - Proponer una solución.
 - Determinar los objetivos del proyecto.
 - Determinar los criterios de éxito del proyecto.
 - Evaluar la situación.
 - Producir el plan del proyecto.
- Fase 2: Comprensión de los datos
 - Recolectar datos iniciales.
 - Describir los datos.
 - Realizar la exploración de los datos.
 - Verificar la calidad de los datos.
- Fase 3 Preparación de los datos
 - Limpiar datos.
 - Selección de datos.
 - Construir la bolsa de palabras.
 - Construir datos.
- Fase 4: Modelado
 - Determinar métricas de evaluación.
 - Construir los modelos.
 - Evaluar los modelos.
 - Analizar el modelo elegido.
- Fase 5: Evaluación

- Realizar evaluación de resultados.
- Realizar revisión del proyecto.

3.6.5 Requisitos de Recursos de la Actividad

Una vez definidas las actividades que deben ser llevadas a cabo para concretar el proyecto, se deben determinar los tipos de recursos humanos, materiales o equipos y la cantidad necesaria de estos.

A continuación, en las Figuras 7 y 8, se identifican el tipo y la cantidad de los recursos necesarios para las diferentes actividades.

Actividad	Recursos	Cantidad
Definir el problema	Director del proyecto	1
	Computadora	1
Proponer una solución	Director del proyecto	1
	Computadora	1
Determinar los objetivos del proyecto	Director del proyecto	1
	Computadora	1
Determinar los criterios de éxito del proyecto	Director del proyecto	1
	Computadora	1
Evaluar la situación	Director del proyecto	1
	Computadora	1
Producir el plan del proyecto	Director del proyecto	1
	Computadora	1
Recolectar datos iniciales	Analista de datos	1
	Computadora	1
Describir los datos	Analista de datos	1
	Computadora	1
Realizar la exploración de los datos	Analista de datos	1
	Director del proyecto	1
	Computadora	1
Verificar la calidad de los datos	Analista de datos	1
	Director del proyecto	1
	Computadora	1

Figura 7: Recursos necesarios para realizar las actividades del proyecto (primera parte).

Actividad	Recursos	Cantidad
Limpiar datos	Analista de datos	1
	Computadora	1
Construir datos	Analista de datos	1
	Computadora	1
Selección de datos	Analista de datos	1
	Computadora	1
Determinar técnicas de modelado	Analista de datos	1
	Computadora	1
Diseñar pruebas del modelo	Analista de datos	1
	Computadora	1
Construir el modelo	Analista de datos	1
	Computadora	1
Evaluar el modelo	Analista de datos	1
	Director del proyecto	1
	Computadora	1
Realizar evaluación de resultados	Analista de datos	1
	Director del proyecto	1
	Computadora	1
Realizar revisión del proyecto	Director del proyecto	1
	Computadora	1
Producir el informe final del proyecto	Director del proyecto	1
	Computadora	1

Figura 8: Recursos necesarios para realizar las actividades del proyecto (segunda parte).

3.6.6 Cronograma del Proyecto

El cronograma del proyecto incluye todos los procesos requeridos para gestionar la finalización del proyecto en tiempo y forma.

A través del juicio de expertos y la información sobre la distribución del esfuerzo en las diferentes etapas de la metodología CRISP-DM, se procedió a estimar la duración de las actividades. Esta estimación se observa en el siguiente diagrama de Gantt, donde se representan las actividades con sus fechas de inicio y finalización esperada.

Tarea	Fecha de inicio	Duración (Días)	Fecha finalización
Proyecto Minería de Texto	1/10/2021	243	1/6/2022
1. Comprensión del negocio	1/10/2021	60	30/11/2021
1.1 Definir el problema	1/10/2021	4	5/10/2021
1.2 Proponer una solución	5/10/2021	6	11/10/2021
1.3 Determinar los objetivos del proyecto	11/10/2021	4	15/10/2021
1.4 Determinar los criterios de éxito del proyecto	15/10/2021	3	18/10/2021
1.5 Evaluar la situación	18/10/2021	10	28/10/2021
1.6 Producir el plan del proyecto	28/10/2021	33	30/11/2021
2. Comprensión de los datos	30/11/2021	45	14/1/2022
2.1 Recolectar datos iniciales	30/11/2021	5	5/12/2021
2.2 Describir los datos	5/12/2021	20	25/12/2021
2.3 Realizar la exploración de los datos	25/12/2021	10	4/1/2022
2.4 Verificar la calidad de los datos	4/1/2022	10	14/1/2022
3. Preparación de los datos	14/1/2022	75	30/3/2022
3.1 Limpiar datos	14/1/2022	50	5/3/2022
3.2 Seleccionar los atributos de los datos	5/3/2022	10	15/3/2022
3.3 Construir bolsa de palabras	15/3/2022	5	20/3/2022
3.4 Construir los datos	20/3/2022	10	30/3/2022
4. Modelado	30/3/2022	50	19/5/2022
4.1 Determinar las métricas de evaluación	30/3/2022	5	4/4/2022
4.2 Construir los modelos	4/4/2022	20	24/4/2022
4.3 Evaluar los modelos	24/4/2022	10	4/5/2022
4.4 Analizar el modelo final	4/5/2022	15	19/5/2022
5. Evaluación	19/5/2022	13	1/6/2022
5.1 Realizar evaluación de resultados	19/5/2022	5	24/5/2022
5.2 Realizar revisión del proyecto	24/5/2022	8	1/6/2022

Figura 9: Diagrama de Gantt (primera parte).

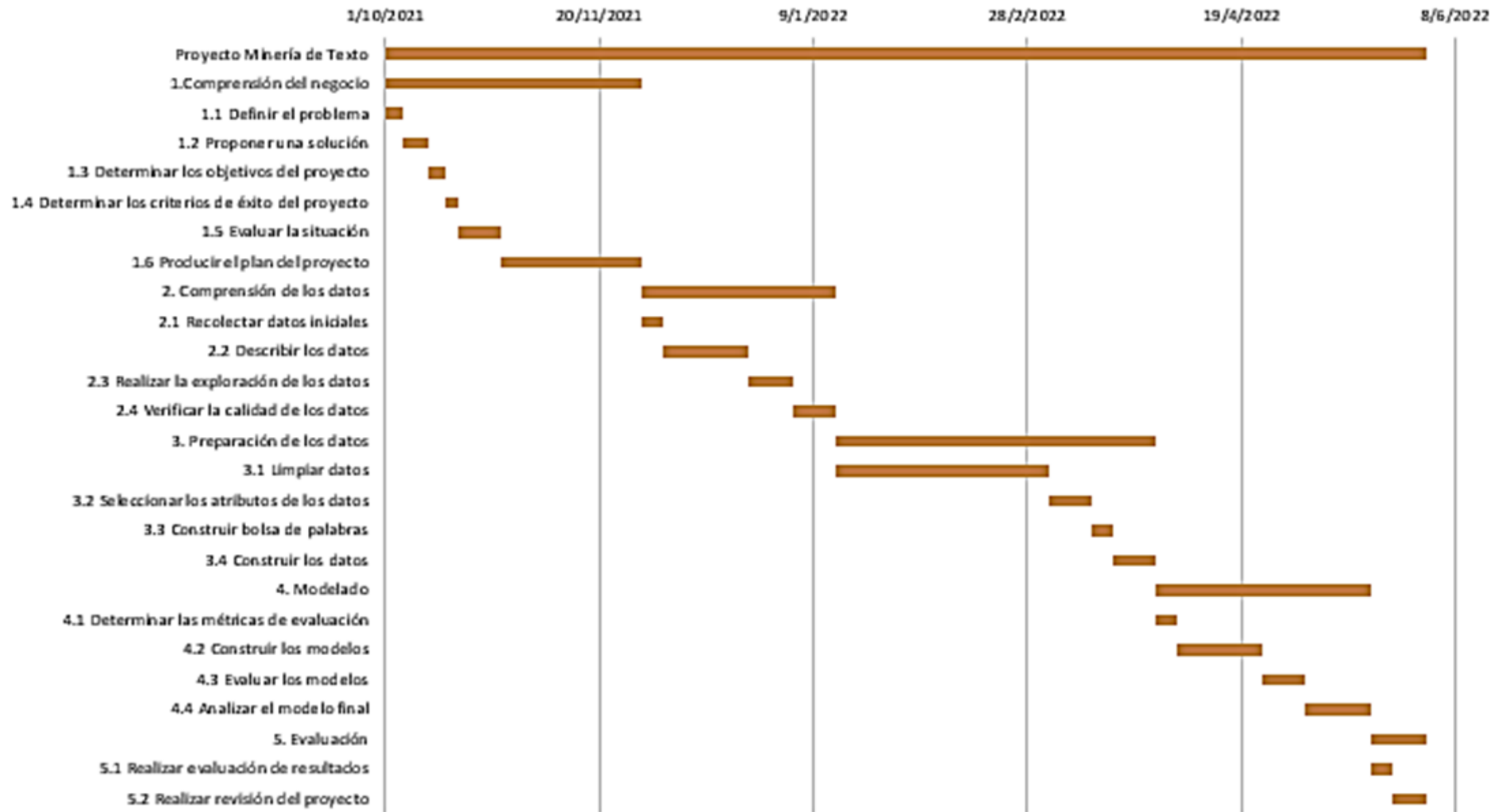


Figura 10: Diagrama de Gantt (segunda parte).

3.6.7 Gestión de Costos y Adquisiciones

Para llevar a cabo la gestión de costos y adquisiciones del proyecto, se aplicó la técnica de estimación paramétrica. *“La estimación paramétrica utiliza una relación entre los datos históricos y otras variables para calcular una estimación de parámetros de una actividad tales como el costo, presupuesto y duración”* (Project Management Institute, 2017).

A partir de los valores de los honorarios por horas obtenidos y la utilización de unidades de tiempo se logró obtener estimaciones con buena precisión.

La gestión de costos y adquisiciones comprende los costos vinculados a recursos humanos, los costos de hardware y otros gastos varios del proyecto que se irán detallando más adelante en este mismo inciso.

3.6.7.1 Línea Base de Costos de Recursos Humanos

La estimación de costos de recursos humanos se presenta en la siguiente Tabla 1. En ella se identifica el tipo de recurso humano, los honorarios por horas hombre, los días laborales a lo largo del proyecto, la cantidad de horas laborales por día y el costo asociado al recurso. Todos los valores referidos al dinero estarán en pesos argentinos.

Recurso	\$/ Hora H.	Días	Horas H. / Día	Total Horas H.	Costo
Director del proyecto	\$ 4.000,00	243	4	972	\$3.888.000,00
Analista de datos	\$3.800,00	165	4	660	\$2.258.000,00
Total					\$6.396.000,00

Tabla 1: Costos de los recursos humanos del proyecto.

La información correspondiente a los honorarios de los profesionales incluidos en el proyecto fue extraída de la página oficial del CPCIPC a la fecha 8/11/2021 (Consejo Profesional de Ciencias Informáticas de la Provincia de Córdoba, 2021)

3.6.7.2 Costos de Hardware

Los costos de hardware hacen referencia a las adquisiciones necesarias a realizar para desarrollar el proyecto.

En este caso, no hace falta adquirir ningún tipo de hardware, debido a que, la computadora disponible (ver sección 3.5.2.3) posee las características necesarias para llevar a cabo todas las actividades sin problemas.

3.6.7.3 Costos de Software

No se presentan costos de software, ya que únicamente se usará Google Colaboratory como herramienta de desarrollo del proyecto, el cuál es un software gratuito.

El sistema operativo utilizado ya se encuentra instalado en el equipamiento anteriormente (ver sección 3.5.2.3), por lo que tampoco se considera gasto.

Y el resto de herramientas software accesorias que fueron elegidas también son gratuitas.

3.6.7.4 Gastos Varios

En la Tabla 2, se detallan los gastos varios del proyecto.

Concepto	Cantidad de meses	Costo mensual	Costo total
Servicio de internet	8	\$1.300,00	\$10.400,00
Servicio de electricidad	8	\$300,00	\$2.400,00
TOTAL			\$12.800,00

Tabla 2: Costos de gastos varios del proyecto.

Cabe destacar que el proveedor del servicio de internet es Claro (50Mb) y el proveedor del servicio de electricidad es EDESA, y los valores corresponden al mes de noviembre del año 2021.

3.6.7.5 Línea Base de Costos

En la Tabla 3, se muestran los costos del proyecto.

Concepto	Costo
Costos de recursos humanos	\$6.396.000,00
Costos de hardware	-
Costos de software	-
Gastos varios	\$12.800,00
TOTAL	\$6.408.800,00

Tabla 3: Línea base de costos del proyecto.

3.6.8 Descripciones de los Cargos

En la Tabla 4, se realiza una descripción de los cargos que se encuentran dentro del proyecto, así como su responsabilidad dentro del proyecto.

Cargo	Responsabilidad
Director del proyecto	Realizar la planificación, ejecución y control de las actividades del proyecto.
Analista de datos	Responsable de desarrollar las tareas de minería de texto, ya sea recolectar los datos, comprenderlos, repararlos, generar el modelo de minería de texto y evaluar el modelo.

Tabla 4: Descripción de los cargos del proyecto.

3.6.9 Gestión de Riesgos

Los riesgos son todos aquellos posibles problemas futuros asociados al proyecto, los cuales tienen cierta probabilidad de ocurrencia. La gestión de riesgos tiene como resultado el registro de riesgos, el cual permite identificar y analizar los riesgos, para planificar las respuestas a cada uno de ellos.

Para identificar los riesgos se utilizó el juicio de expertos y se realizó una revisión de la documentación del proyecto, particularmente los requerimientos, presunciones y restricciones. Mientras que para planificar las respuestas a los riesgos identificados se optó por mitigar o reducir los riesgos, es decir, reducir la probabilidad o el impacto del riesgo en el proyecto.

3.6.9.1 Registro de Riesgos

A continuación, se determinan los riesgos identificados, así como las medidas llevadas a cabo para mitigar o reducir los mismos.

- **Finalización del proyecto en un periodo mayor a dos años:** Debido al reglamento impuesto para este proyecto de grado, el proyecto debe ser concretado antes de los dos años a partir de diciembre del año 2021. El plazo de finalización del proyecto es una restricción que debe cumplirse obligatoriamente.

Para evitar retrasos se realizará un periódico control acerca del estado de avance del mismo, por otro lado, se revisarán los criterios utilizados a la hora de elaborar el cronograma de tiempo.

- **Fallos en el equipamiento informático disponible:** El presente proyecto se desarrollará sobre el equipamiento informático antes nombrado, una falla en el software o en el hardware puede perjudicar el proyecto perdiendo datos críticos y haciendo que las actividades no se realicen en el plazo definido.

Para evitar que esto suceda, se realizarán copias periódicas de seguridad de los diferentes documentos del proyecto para no perder el progreso en caso de algún fallo. Estas copias se almacenarán en dispositivos externos y también en la nube para mayor seguridad.

- **Uso de datos sesgados:** existe la posibilidad que el dataset inicial se encuentre sesgado, es decir que los datos dirijan el proceso de agrupamiento hacia cierto cluster o hacia ciertas características. Esto se traduciría como un resultado no deseado.

Para evitar este problema, se hará uso de los procesos y herramientas necesarias para normalizar los datos.

- **Exposición de datos sensibles:** Este proyecto tratará con datos personales de gente involucrada en el relato de las denuncias, sería muy desafortunado que alguna de las partes con acceso al contenido del conjunto de datos inicial filtrara cualquier información sensible de alguna de las personas involucradas, ya que podría traducirse en cometer un delito y, potencialmente, enfrentar un conflicto judicial.

Para evitar este riesgo, se realizará una adecuada anonimización del relato de las denuncias utilizadas y, de esta manera se eliminará toda información personal incluida en las mismas.

- **Potencial privatización del software:** Si bien los sistemas software que se van a emplear en el proyecto como herramientas son todas de uso gratuito, siempre está la posibilidad de que esto en el transcurso del desarrollo del proyecto cambie y pasen a ser herramientas en las que haya que pagar una licencia de uso para emplearlas en su totalidad.

Ante esta situación, se analizan dos posibles cursos de acción, por un lado en caso de no ser muy onerosa la cantidad a pagar para hacer uso del software, se abonará el coste necesario. La otra opción es la de cambiar de alternativa tecnológica a fines de concretar el proyecto de la forma menos costosa posible.

3.6.9.2 Matriz de Probabilidad e Impacto

Luego de identificar los riesgos presentes en el proyecto, así como las respuestas a emplear para mitigar dichos riesgos, se procede a realizar el análisis de dichos riesgos mediante el uso de una matriz de probabilidad e impacto, mostrada en la Tabla 5.

En esta tabla se analiza la probabilidad de ocurrencia de los riesgos, así como el impacto de los mismos sobre los objetivos del proyecto.

Entre las columnas de la matriz de probabilidades e impacto se tiene:

- **Riesgo:** Nombre del riesgo a analizar.
- **Probabilidad:** La probabilidad de los riesgos se define en el intervalo $[0;1]$.
- **Impacto:** El impacto definido para cada riesgo corresponde a un valor que se encuentra en el siguiente intervalo $[0;1]$.
- **Valoración:** La valoración representa el producto entre la probabilidad y el impacto de cada riesgo.
- **Categorización:** La categorización se realiza en función del resultado de la valoración del riesgo. Esta puede tomar cualquiera de los siguientes valores [Baja (color verde), Media (color amarillo), Alta (color rojo)], de acuerdo a la Figura 9.

		Riesgo = Probabilidad x Impacto				
Probabilidad	0,9	0,05	0,09	0,18	0,36	0,72
	0,7	0,04	0,07	0,14	0,28	0,56
	0,5	0,03	0,05	0,10	0,20	0,40
	0,3	0,02	0,03	0,06	0,12	0,24
	0,1	0,01	0,01	0,02	0,04	0,08
		0,05	0,10	0,20	0,40	0,80
		Impacto				

Figura 11: Categorización de un riesgo de acuerdo a su valoración. (Escuela de Organización Industrial, 2016).

Riesgo	Probabilidad	Impacto	Valoración	Categorización
Finalización del proyecto en un periodo mayor a dos años	0,1	0,8	0,8	Media
Fallos en el equipamiento informático disponible	0,1	0,2	0,2	Baja
Uso de datos sesgados	0,1	0,8	0,8	Media
Exposición de datos sensibles	0,1	0,8	0,8	Media
Potencial privatización del software	0,1	0,4	0,04	Baja

Tabla 5: Matriz de probabilidad e impacto del proyecto.

3.6.10 Alternativas Tecnológicas

En este inciso, se muestran las distintas alternativas que se consideraron para el desarrollo del proyecto, así como también el análisis y la selección de las herramientas y/o software.

3.6.10.1 Alternativas de Sistemas Operativos

En la Tabla 6, se expondrán las ventajas y desventajas de los distintos sistemas operativos que se tuvieron en cuenta a la hora de hacer la elección del mismo (Mantenimiento Informático Económico, s.f.).

SO	Ventajas	Desventajas
Windows	Windows es fácil de usar y configurar, sin necesidad de poseer conocimientos informáticos avanzados. El sistema operativo Windows se basa en el lenguaje, mientras que Mac está basado en Unix. Es capaz de ejecutar múltiples aplicaciones.	Es el sistema operativo que usa el 90% de los ordenadores personales de todo el mundo. Existen multitud de virus que aprovechan los fallos del sistema para infectar el PC. A la hora de instalar Windows prácticamente no puedes configurar nada. Aunque la instalación es mucho más fácil y cómoda.
Mac	Mac Os es considerado por muchos expertos como uno de los sistemas operativos más sencillos de utilizar, más innovador y de estética más cuidada. Al igual que Windows tiene ventanas y escritorio, ambos tienen papelera de reciclaje.	Mac ha ganado estabilidad, seguridad y ausencia de virus. Su mayor costo no está en la propia licencia de su uso, sino que únicamente funciona en los ordenadores que fabrica la misma compañía. Falta total de una línea de comandos; es un

		sistema operativo completamente de extensiones.
Linux	Es un poco más seguro como Mac ya que no muchos lo manejan lo que hace menos frecuente los ataques a este sistema operativo. Es capaz de manejar archivos multimedia, de mantener una interfaz de usuario en forma gráfica y tiene papelera de reciclaje al igual que los otros dos.	Ofrece potencia, estabilidad, seguridad contra los virus y sirve para realizar cualquier trabajo. Linux es un sistema operativo libre y gratuito, pero es algo más difícil de utilizar. Para instalar Linux hay que realizar relativos esfuerzos, aunque cada vez es una tarea más sencilla, pero puedes personalizar totalmente la instalación.

Tabla 6: Ventajas y desventajas entre sistemas operativos.

3.6.10.2 Alternativas de Software para Minería de Texto

A continuación, se procede a establecer las distintas alternativas analizadas a la hora de seleccionar el software utilizado como herramienta para el desarrollo del proyecto de minería de texto.

- **Google Colaboratory:** Colaboratory, o "*Colab*" para abreviar, es un software de Google Research. Permite a cualquier usuario escribir y ejecutar código arbitrario de Python en el navegador.

Es especialmente adecuado para tareas de aprendizaje automático, análisis de datos y educación, como es el caso de este proyecto. Colab es un servicio de cuaderno alojado de Jupyter que no requiere configuración y que ofrece acceso sin coste adicional a recursos informáticos, como GPUs.

Colab puede usarse sin ningún coste. Aunque tiene algunas limitaciones, los recursos del software no están garantizados ni son ilimitados, y los límites de uso a veces varían.

Los recursos se asignan dando prioridad a los casos prácticos interactivos. Están prohibidas las acciones asociadas a operaciones informáticas en bloque, las

acciones que afecten negativamente a otras y las acciones orientadas a eludir las políticas de Google (Google, s.f.).

- **Rapidminer:** Es un sistema que apoya el diseño y la documentación de un conjunto de procesos de minería de datos. Es de uso gratuito.

Ofrece no sólo un conjunto casi completo de operadores, sino también estructuras que expresan el flujo de control del proceso. Para utilizar estas estructuras no es necesario realizar ningún tipo de programación, basta con solo arrastrar, soltar y hacer clic.

Visualmente, las estructuras se muestran mediante recuadros que están vinculados o anidados. Esta presentación es fácil de entender.

Dadas las largas cadenas de operadores y los procesos anidados en la minería de datos, el aspecto de la documentación se vuelve indispensable. Los parámetros elegidos de, por ejemplo, la discretización, las transformaciones de características particulares y los criterios de selección de características se almacenan con el proceso RapidMiner.

Los metadatos de los datos están disponibles en toda etapa del proceso, por lo tanto, el resultado es explicable y reproducible (Hofmann & Klinkenberg, 2013).

- **Weka:** El *workbench* de Weka es una plataforma para el aprendizaje automático aplicado. Weka es un acrónimo que significa Waikato Environment for Knowledge Analysis. Cinco de las principales características para usar Weka son:
 - Código abierto: Weka se publica como software de código abierto bajo la licencia GPL de GNU.
 - Interfaz Gráfica: Posee una Interfaz Gráfica de Usuario (GUI). Esto permite completar proyectos de aprendizaje automático sin programar.
 - Interfaz de líneas de comando: todas las funciones del software se pueden usar desde líneas de comando. Esto puede ser muy útil si está buscando llevar su uso de la plataforma al siguiente nivel y comenzar a automatizar tareas comunes.
 - API de Java: está escrito en Java y proporciona una API que está bien documentada y promueve integración en sus propias aplicaciones.
 - Comunidad: la interfaz y los algoritmos están bien documentados y la gran comunidad que posee la plataforma se traduce en mucho soporte.

La razón principal por la que Weka es tan bueno para los principiantes es que se puede trabajar en el proceso de aprendizaje automático utilizando la interfaz gráfica sin tener que escribir una sola línea de código. Un principiante debería aprender a controlar los procesos, manejar datos y experimentar con algoritmos, sería un problema si además de esto debería tener que aprender un lenguaje de programación para poder llevarlos a cabo. Aquí hay algunas razones adicionales:

- Proporciona una interfaz gráfica de usuario simple que encapsula el proceso de aplicación aprendizaje automático.
 - Facilita la exploración de algoritmos y conjuntos de datos, así como el diseño riguroso de experimentos y análisis.
 - Es gratuito y de código abierto, lo que significa que puede descargarlo y comenzar a usarlo.
 - Es multiplataforma y se ejecuta en Windows, Mac OS X y Linux, lo que significa que se ejecutará en sea cual sea el entorno que estés usando (Brownlee, 2016).
- **R:** Es un entorno software gratuito para llevar a cabo técnicas de manipulación de datos, estadísticas y gráficas.

R puede extenderse fácilmente a través de paquetes, hay una gran variedad de paquetes disponibles en el repositorio de paquetes CRAN (Zhao, 2012).

Entre sus ventajas posee:

- Un eficaz manejo y almacenamiento de datos.
- Un conjunto de operadores para cálculos en arreglos, en particular matrices.
- Una colección grande, coherente e integrada de herramientas intermedias para el análisis de datos.
- Herramientas gráficas para el análisis y la visualización de datos.
- Un lenguaje de programación bien desarrollado, simple y efectivo (llamado 'S') que incluye condicionales, bucles, funciones recursivas definidas por el usuario y facilidades de entrada y salida. (Por cierto la mayoría de las funciones proporcionadas por el sistema están escritas en lenguaje S).

El término “entorno” pretende caracterizarlo como un sistema totalmente planificado y coherente, en lugar de una acumulación incremental de herramientas muy específicas e inflexibles, como suele ser el caso con otros software de análisis de datos (Venables, 2009).

3.6.11 Elección de las Alternativas Tecnológicas

A continuación, se procederá a elegir cuáles serán las alternativas tecnológicas que van a utilizarse en el desarrollo del proyecto justificando el motivo de su elección.

3.6.11.1 Sistema Operativo

El sistema operativo elegido para desempeñar todas las actividades requeridas del proyecto será Windows, específicamente Windows 10.

El motivo de la elección de este por sobre el resto, es en primer lugar, que no se posee el hardware necesario para utilizar el sistema operativo MAC, por lo que queda descartado automáticamente.

Con respecto a la elección de Windows por sobre Linux, el principal motivo se debe a una cuestión de comodidad propia, debido a que ya se encuentra instalado en el equipo informático que va a usarse (por lo que no debería realizarse la compra del mismo).

También se destaca que se tiene muchísima más experiencia en el uso del mismo de manera que las tareas se realizarán de manera más fluida.

Otro aspecto a destacar de Windows es que, en general, presenta una mayor compatibilidad con las distintas herramientas complementarias que puedan llegar a necesitarse para el desarrollo del proyecto.

3.6.11.2 Software para Minería de Texto

El software que se va a utilizar para llevar a cabo el proyecto de minería de texto como herramienta principal es Google Colaboratory.

Se eligió este software por sobre el resto debido al drástico crecimiento que tuvo en los últimos años siendo, hoy en día, una de las herramientas más utilizadas para el desarrollo de proyectos tanto de minería de datos como de texto. Esto se traduce en un mayor soporte en línea, lo que facilita de sobremanera la resolución de cualquier inconveniente que pueda producirse.

Siempre es importante destacar que es un software gratuito.

Otro de los aspectos de Google Colaboratory por los que se lo eligió, es el que usa Python como lenguaje de programación, el cual posee muchas herramientas y librerías para facilitar el desarrollo de un proyecto de minería de texto.

Otro de los principales motivos de su elección fue la portabilidad que ofrece colab. Este puede ejecutarse en el navegador sin importar el hardware que lo esté ejecutando, gracias a que Google ofrece 12 GB de memoria RAM y 107 GB de disco por cuaderno. De manera que, si en algún momento se sufriera algún percance con el hardware utilizado, no

imposibilitaría continuar con el desarrollo del proyecto, y aún más importante, no se perdería todo el trabajo realizado en el cuaderno.

3.6.12 Análisis Costo-Beneficio del Proyecto

El costo final estimado del proyecto es de \$6.408.800,00 (seis millones cuatrocientos ocho mil ochocientos pesos) según los valores presupuestados.

Por tratarse de un proyecto de investigación se dificulta el hecho de plasmar los beneficios de forma monetaria. Por ende se van a nombrar aquellos aspectos positivos que promueve el proyecto.

- Proporcionar información certera sobre delitos penales de robo y hurto en la provincia de Salta, con el fin de que las autoridades pertinentes, en este caso el Ministerio Público Fiscal de manera conjunta y coordinada con las fuerzas policiales, tomen esta información para poder combatir con este mal inserto en la sociedad argentina mediante acciones que contribuyan a su prevención.
- Brindar información certera a la población salteña, para que sepan de los potenciales peligros a los que se pueden enfrentar y sepan tomar los recaudos necesarios para no ser víctimas de este tipo de delitos.
- Ahorrar tiempo y dinero a la hora de realizar un estudio/análisis de denuncias contra la propiedad. Para dimensionar el tiempo y dinero empleados en esta actividad, se va a hacer un pequeño cálculo.

Particularmente en el Ministerio Público Fiscal de la Provincia de Salta, las personas encargadas de realizar un análisis sobre las denuncias ingresadas son el personal administrativo, en este caso con una asignación de funciones de auxiliares de investigación, los cuales cobran un sueldo básico de \$88.884,63 por mes (Poder Judicial de Salta, 2022).

En este caso se trabajarán con 3600 denuncias aproximadamente. En promedio, un sumariante demora 4 minutos en realizar una lectura comprensiva de cada denuncia, por ende para realizar únicamente una lectura de los datos actuales se demoraría un total de 30 días hábiles si trabaja 8 horas por día, lo que sería un mes y medio de trabajo si es que se trabaja 5 días a la semana. Traducido en dinero serían unos \$133.326.00 aproximadamente. Y cabe resaltar que únicamente se trata de realizar una lectura, ni siquiera se habla de realizar algún tipo de registro o análisis.

El software utilizado solo demora aproximadamente 1 hora en procesar todas las denuncias y devolver información certera basada en los datos.

- Ser reproducible para otras denuncias contra la propiedad. Una vez terminado el proyecto, se podrá realizar el proceso de minería de texto sobre cualquier dataset que contenga denuncias de robo y hurto con el mismo formato sin tener que modificar nada o muy poco.

En este caso al haber utilizado como conjunto de datos a denuncias provistas por el Ministerio Público Fiscal de la provincia de Salta, se podría ingresar cualquier dataset provisto por esta entidad (o cualquier entidad que almacene las denuncias con el mismo formato) y obtener información presuntamente útil.

Capítulo 4 - Comprensión de los Datos

La segunda etapa en la metodología CRISP-DM consiste en el estudio y comprensión de los datos. En esta etapa del proyecto, se procede a recolectar los datos de las respectivas fuentes, y a realizar un primer análisis que luego será detallado en la siguiente etapa.

El objetivo de esta etapa es familiarizarse con el conjunto de datos y encontrar en ellos problemas de calidad para luego tomar decisiones al respecto.

4.1 Datos Iniciales

Los datos contenidos en el dataset inicial con el que va a llevarse a cabo el proyecto de minería, fueron provistos por el Ministerio Público Fiscal de la provincia de Salta en un archivo con extensión .csv. Estos datos contienen denuncias policiales de delitos contra la propiedad (robo y hurto), dichas denuncias están comprendidas entre los años 2018 y 2019, por lo que son bastante recientes y, por ende, consideradas útiles.

4.2 Descripción de los Datos

El conjunto de datos está compuesto por 3597 denuncias policiales, de las cuales 3589 son del año 2018, solo 7 denuncias del año 2019, 1 denuncia que corresponde al año 2014 y 2 correspondientes al año 2017, en la siguiente etapa se decidirá que hacer al respecto con estas denuncias aisladas.

Todas estas denuncias se encuentran en el archivo llamado “*Denuncias2018-2019.csv*”.

4.3 Exploración de los Datos

Para poder hacer un análisis más detallado, se efectuó la creación de un archivo en Google Drive del tipo “*Google Colaboratory*”.

A su vez, para poder trabajar con el dataset elegido, este fue colocado en una carpeta de Google Drive para poder acceder al mismo desde Google Colaboratory.

4.3.1 Lectura de Datos

Lo primero que se debe realizar es importar las librerías que se van a utilizar para realizar la correspondiente lectura de los datos, en este caso “*Pandas*” y “*DataTable*” (ver Anexo II).

Luego, se debe vincular el archivo “*Denuncias2018-2019.csv*” desde Google Drive con Google Colaboratory (ver Anexo I) para luego poder acceder al dataset, colocar los datos en un *dataframe* y poder visualizarlos.

```
#Lectura del Dataset
d_denuncias = pd.read_csv("/content/drive/MyDrive/Proyecto de Grado/Denuncias2018-2019.csv", sep = ";")
DataTable(d_denuncias)
```

Figura 12: Código de la lectura y visualización del dataset.

En la primera línea de código, se crea el dataframe llamado “*d_denuncias*” con todos los datos del archivo, cabe destacar que como se observa en la Figura 12, se agrega el campo “*sep = “;”*” debido a que las denuncias dentro del archivo “.csv” se encuentran separadas por “;”, por lo que es la forma indicada de leer los datos.

Una vez obtenidas las denuncias en formato de tabla, se observa en la Figura 13 que tiene distintas columnas, estas se analizan a continuación.

index	fchConfirmacion	DEPENDENCIA	actaNro	Anio	fchHrHecho	lugarHecho	DELITO	FISCALIA	RELATO
-------	-----------------	-------------	---------	------	------------	------------	--------	----------	--------

Figura 13: Columnas obtenidas de la visualización del dataset.

Descripción de los campos obtenidos del dataset:

- **index:** Corresponde al índice propio de la tabla obtenida con la librería “*Pandas*”. Contiene números enteros.
- **fchConfirmación:** Es un campo que refleja la fecha y hora a la cuál se realizó la denuncia. Con el formato “*aaaa/mm/dd/hh/MinMin/ss*”.
- **DEPENDENCIA:** Este campo indica cuál fue el lugar físico donde se realizó la respectiva denuncia.
- **actaNro:** Refleja el número de acta al cuál pertenece la denuncia realizada.
- **Anio:** Indica el año en el que se realizó la denuncia del hecho.
- **fchHrHecho:** Refleja la fecha y hora la cuál sucedió el hecho denunciado.
- **lugarHecho:** Indica el lugar donde sucedió el hecho denunciado.

- **DELITO:** Indica que tipo de delito se está denunciando. En este caso solo hay delitos contra la propiedad, es decir, robos y hurtos.
- **FISCALIA:** Indica bajo la jurisdicción de qué fiscalía corresponde asignar a cada denuncia.
- **RELATO:** Consiste en la denuncia del hecho en sí, donde el denunciante explica el delito sucedido.

4.3.2 Transformación de Datos

En esta etapa, las únicas transformaciones que se van a realizar sobre el dataset son las necesarias para poder tokenizar los datos como se explica a continuación.

4.3.2.1 Tokenización

Tokenizar denuncias consiste en dividir el texto en las unidades que lo conforman, entendiendo por unidad el elemento más sencillo con significado propio, en este caso, las palabras.

Lo primero a realizar, es importar las librerías necesarias para llevar a cabo el proceso de tokenización, en este caso “*nltk*” y “*re*” (ver Anexo II).

Luego, se define una función que se encarga de hacer una primera limpieza básica y separar el texto en palabras individuales (tokenizar).

```
#Función para tokenizar
def tokenizar(texto):

    # Se convierte todo a string
    nuevo_texto = str(texto)
    # Se convierte todo el texto a minúsculas
    nuevo_texto = nuevo_texto.lower()
    # Eliminación de signos de puntuación
    regex = '[\!|\\"|\#\|\$\%\|\&\|\'\|\(\|\)\|\*\|\+|\,|\-|\.\|\/|\:|\;|\<|\=|\>|\?|\@|\[\|\]\|\^|\_|\`|\{|\}|\||~]'
    nuevo_texto = re.sub(regex, ' ', nuevo_texto)
    # Eliminación de espacios en blanco múltiples
    nuevo_texto = re.sub("\s+", ' ', nuevo_texto)
    # Tokenización por palabras individuales
    nuevo_texto = nuevo_texto.split(sep = ' ')

    return(nuevo_texto)
```

Figura 14: Función para tokenizar los datos.

Como se observa en la Figura 14, la función primero se convierten todos los datos a tipo “*string*” para poder operarlos, luego se procede a llevar todos las palabras a minúsculas para que no se confundan palabras iguales pero que en algunas ocasiones están escritas con

mayúsculas y en otras con minúsculas. Luego, se procede a eliminar los signos de puntuación y los espacios en blanco para después generar los tokens.

A continuación, se procede a realizar una copia del dataframe obtenido de la lectura para trabajar sobre esta copia y no sobre el original, a partir de aquí cada vez que se realice un proceso diferente se realizará una copia de los datos obtenidos del paso anterior por cuestiones de seguridad de los datos y prolijidad (ver Anexo II).

Ahora, se hará uso de la función creada para tokenizar sobre el atributo “*Relato*”, que es el que contiene la descripción de los hechos, y el principal foco de análisis a la hora de realizar este proceso de minería.

```
##Ejecución de la función
d_denuncias_token['RELATO'] = d_denuncias_token['RELATO'].apply(lambda x: tokenizar(x))

##Corroboración
d_denuncias_token["RELATO"]

0      [primeramente, aclara, que, radica, la, denunci...
1      [primeramente, manifiesta, que, la, acusada, e...
2      [en, el, día, de, la, fecha, hora, de, epigraf...
3      [primeramente, quiere, dejar, aclarado, que, e...
4      [manifiesta, que, en, fecha, hs, aprox, el, ma...
...
3592   [deja, aclarado, que, esta, de, pareja, con, e...
3593   [radica, la, denuncia, en, contra, de, su, ex,...
3594   [radica, la, denuncia, en, contra, de, su, ex,...
3595   [que, se, encuentra, hace, cuatro, años, en, p...
3596   [radica, la, presente, en, contra, de, su, ex,...
```

Figura 15: Uso de la función creada para tokenizar nuestros datos sobre la columna “*Relato*”.

4.3.3 Bolsa de Palabras

La bolsa de palabras consiste en una forma de representar un documento o conjunto de documentos, en base a las palabras que éste o éstos contienen.

Esta primera bolsa de palabras, se hará principalmente para tener una primera visualización de los datos más frecuentes, corroborar la calidad actual de los datos y, más adelante, poder compararlo con una bolsa de palabras obtenida a partir de los datos que ya hayan pasado por un proceso de limpieza.

Lo siguiente consistirá en crear una función que permita crear la bolsa de palabras y que devuelva cada término con su respectiva frecuencia como se observa en la Figura 16.

```
#Función para mostrar la frecuencia de los términos en una bolsa de palabras
#creada a partir de una columna de un dataframe previamente tokenizada
def bow(dataframe, columna):
    long_bow = len(dataframe[columna])
    wordfreq = {}
    ind_col = dataframe.columns.get_loc(columna)
    for i in range(long_bow):
        for token in dataframe.iloc[i, ind_col]:
            if token not in wordfreq.keys():
                wordfreq[token] = 1
            else:
                wordfreq[token] += 1
    sorted_bow = sorted(wordfreq.items(), key=lambda kv: kv[1], reverse=True)
    return sorted_bow
```

Figura 16: Función para la obtención de la frecuencia de cada término en una bolsa de palabras.

Luego se procede a ejecutar la respectiva función (ver Anexo II). Una vez que se obtiene la bolsa de palabras, se procede a mostrar los 20 términos que se repiten con mayor frecuencia en el relato de las denuncias.

```
#Obtención de los 20 términos más frecuentes
bow[:20]

[('de', 38484),
 ('que', 30348),
 ('la', 27209),
 ('en', 21366),
 ('el', 18384),
 ('se', 13616),
 ('su', 10987),
 ('del', 10051),
 ('con', 8834),
 ('al', 7985),
 ('por', 7448),
 ('no', 6807),
 ('color', 5873),
 ('le', 5871),
 ('un', 5769),
 ('una', 4838),
 ('los', 4487),
 ('lo', 3957),
 ('fecha', 3559),
 ('las', 3547)]
```

Figura 17: Los 20 términos más frecuentes de la bolsa de palabras.

4.4 Calidad de los Datos

Como se observa en la Figura 17, casi todas las palabras más utilizadas son preposiciones, pronombres o artículos, por lo que estas palabras no aportan información útil al análisis que se quiere realizar en este proyecto.

En las siguientes etapas del proyecto, se procederá a realizar procesos para que los datos con los que se trabajen sean de calidad y aporten información útil para el proceso de agrupamiento.

Capítulo 5 - Preparación de los Datos

Durante esta etapa de preparación de datos, se realizan las transformaciones necesarias para poder utilizar el contenido de las denuncias obtenidas en el proceso de agrupamiento (clustering). Es necesario representar los datos de manera adecuada para poder ser utilizados por los algoritmos que serán empleados en la siguiente etapa de modelado.

5.1 Limpieza de Datos

La limpieza de datos, consiste en un procedimiento mediante el cuál se busca eliminar palabras o datos que no aporten información útil al proceso de agrupamiento que se llevará a cabo más adelante.

5.1.1 Datos Faltantes

La primera limpieza que se va a realizar, es la de detectar los datos faltantes para luego decidir qué hacer con ellos.

Para esta actividad, se instaló el conjunto de librerías “*Funpymodeling*” (ver Anexo II) que facilitará la detección de valores nulos en el dataframe.

Particularmente, se va a utilizar la librería “*status*”, por lo que es la librería que se va a importar (ver Anexo II).

Una vez obtenidas las herramientas necesarias, se va a proceder a realizar una nueva lectura del dataset inicial, es decir, sin tokenizar, debido a que resulta más sencillo realizar este tipo de procedimientos en ese formato.

```
#Lectura del dataset inicial  
d_denuncias_prep = pd.read_csv("/content/drive/MyDrive/Proyecto de Grado/Denuncias2018-2019.csv", sep = ";")
```

Figura 18: Nueva lectura de dataset original.

Se hará uso de la librería “*status*” para poder visualizar la cantidad de valores nulos o faltantes en el dataframe.

```
#Registros faltantes
status(d_denuncias_prep)
```

	variable	q_nan	p_nan	q_zeros	p_zeros	unique	type
0	fchConfirmacion	0	0.000000	0	0.0	3451	object
1	DEPENDENCIA	0	0.000000	0	0.0	100	object
2	actaNro	0	0.000000	0	0.0	1927	int64
3	Anio	0	0.000000	0	0.0	4	int64
4	fchHrHecho	1	0.000278	0	0.0	921	object
5	lugarHecho	127	0.035307	0	0.0	3264	object
6	DELITO	0	0.000000	0	0.0	4	object
7	FISCALIA	0	0.000000	0	0.0	29	object
8	RELATO	0	0.000000	0	0.0	3451	object

Figura 19: Valores faltantes detectados en el dataframe.

Como se observa en la Figura 19, hay 1 registro que carece de la fecha en la que sucedió el delito y 127 que carecen de la ubicación en la que sucedió el hecho.

Una vez detectados los datos faltantes, se tomó la decisión de eliminar las denuncias que carecen de estos registros, debido a que, gracias a la considerable cantidad de denuncias disponibles, eliminar estos registros incompletos no representaría un cambio sustancial al resultado del procedimiento que se quiere realizar.

```
#Eliminación de denuncias
d_denuncias_prep = d_denuncias_prep.dropna()
```

Figura 20: Eliminación de denuncias con atributos nulos.

Luego se corrobora que los datos fueron eliminados correctamente.

```
#Confirmación de la eliminación de denuncias
status(d_denuncias_prep)
```

	variable	q_nan	p_nan	q_zeros	p_zeros	unique	type
0	fchConfirmacion	0	0.0	0	0.0	3326	object
1	DEPENDENCIA	0	0.0	0	0.0	98	object
2	actaNro	0	0.0	0	0.0	1872	int64
3	Anio	0	0.0	0	0.0	4	int64
4	fchHrHecho	0	0.0	0	0.0	896	object
5	lugarHecho	0	0.0	0	0.0	3263	object
6	DELITO	0	0.0	0	0.0	4	object
7	FISCALIA	0	0.0	0	0.0	29	object
8	RELATO	0	0.0	0	0.0	3326	object

Figura 21: Eliminación correcta de registros.

5.1.2 Datos Repetidos

La siguiente operación de limpieza, consiste en verificar si hay denuncias repetidas para luego eliminarlas en caso de que sí existan.

Lo primero será importar la librería “*numpy*” (ver Anexo II) que se utilizará en este paso. Específicamente, se utilizará el método “*unique*” de esta librería, este devuelve los elementos únicos (sin repetir) ordenados de un array, lo cuál será requerido en una función más adelante.

Una vez que se tiene la librería a disposición, se procederá a simplificar la columna “*DELITO*”, esto se debe a que si bien los delitos son clasificados en “Robo” y “Hurto”, en algunas denuncias además de este dato se aclara si es calificado y en otras especifican de acuerdo a qué artículo del código penal se define de esta forma.

Por el motivo descrito, se procederá a modificar esta columna para que únicamente muestre si se trata de un robo o un hurto, sin ninguna otra especificación. Una de las principales razones para realizar esta modificación consiste en que se pudo observar que hay

algunas denuncias iguales a otras pero que únicamente varían en la especificación de su clasificación de delito, un ejemplo de esto son las denuncias observadas en las Figuras 22 y 23.

35	2018-01-04 02:00:15.490	COM. N°04	15	2018	2018-01-02 00:00:00.000	calle Uriburu y Obispo Romero Villa Soledad	ROBO (ART. 164 DEL C.P.)	EX FISCALIA PENAL N° 6 (15/10/2018)
36	2018-01-04 02:00:15.490	COM. N°04	15	2018	2018-01-02 00:00:00.000	calle Uriburu y Obispo Romero Villa Soledad	ROBO CALIFICADO	EX FISCALIA PENAL N° 6 (15/10/2018)

Figura 22: Denuncias n° 35 y n°36.

Que en fecha 03/01/17 a hs. 09:00 aprox. recepciono llamado telefónico del número [pereneciente al [my] dño. 6° 20 de Junio, quien le dijo "YO ME LLEVE EL CAMION, PAGAME", por lo que al tomar conocimiento inmediatamente fue a ver si el camión de su propiedad, el cual se encontraba estacionado en calle Uriburu y Obispo Romero Villa Soledad efectivamente se encontraba en el lugar, haciendo notar que lo dejó en ese sitio porque queda a una cuadra de la casa de su ex pareja: (45) quien vive en Pje de	
Por lo que al llegar al lugar se dio con la novedad de que el mismo ya no se encontraba, procediendo a entrevistarse con un vecino que trabaja en un lavadero (único dato), quien le expuso que en fecha 02/01/17 a hs. 13:00 aprox., un masculino se llevó el rodado, previo abrir la puerta, para luego ser tirado por una camioneta mca Toyota de color gris, haciendo notar que este se encontraba con otro masculino no recordando características del mismo. Aportando que el vehículo sería un camión mca FIAT MECO dominio de color rojo Cuadro N° Motor n°	
Manifestando que el acusado es prestamista ilegal, y que en el año 2014 le solicitó un préstamo de \$ 50.000 (cincuenta mil pesos), el cual fue abonado en su totalidad más intereses en el mes de Octubre de año del 2015, pagando un total de \$62.000 (sesenta y dos mil pesos), pero que el encartado quiso obtener más dinero, por lo que el damnificado se negó a pagarle ya que había salido su deuda, pero nuevamente el: le recordó que el docente le había firmado la transferencia del rodado, como garantía, recordando el damnificado que el día que se realizó el préstamo el Acusado lo llevo ante una escribana la con día laboral en	
donde el mismo le firmo el "GS" como garantía para que le preste el dinero. Expresando que jamás imagino que el acusado podría hacerle pasar por esta situación, ya que era una persona de confianza y que esta situación le trae aparejado varios problemas económicos por cuanto es el único medio que utiliza para sostener a su familia. Es por ello que solicita de manera urgente se tomen las medidas correspondientes al caso. Es todo cuanto denuncia al respecto.	
Que en fecha 03/01/17 a hs. 09:00 aprox. recepciono llamado telefónico del número [pereneciente al [my] dño. 6° 20 de Junio, quien le dijo "YO ME LLEVE EL CAMION, PAGAME", por lo que al tomar conocimiento inmediatamente fue a ver si el camión de su propiedad, el cual se encontraba estacionado en calle Uriburu y Obispo Romero Villa Soledad efectivamente se encontraba en el lugar, haciendo notar que lo dejó en ese sitio porque queda a una cuadra de la casa de su ex pareja: (45) quien vive en Pje de	
Por lo que al llegar al lugar se dio con la novedad de que el mismo ya no se encontraba, procediendo a entrevistarse con un vecino que trabaja en un lavadero (único dato), quien le expuso que en fecha 02/01/17 a hs. 13:00 aprox., un masculino se llevó el rodado, previo abrir la puerta, para luego ser tirado por una camioneta mca Toyota de color gris, haciendo notar que este se encontraba con otro masculino no recordando características del mismo. Aportando que el vehículo sería un camión mca FIAT MECO dominio de color rojo Cuadro N° Motor n°	
Manifestando que el acusado es prestamista ilegal, y que en el año 2014 le solicitó un préstamo de \$ 50.000 (cincuenta mil pesos), el cual fue abonado en su totalidad más intereses en el mes de Octubre de año del 2015, pagando un total de \$62.000 (sesenta y dos mil pesos), pero que el encartado quiso obtener más dinero, por lo que el damnificado se negó a pagarle ya que había salido su deuda, pero nuevamente el: le recordó que el docente le había firmado la transferencia del rodado, como garantía, recordando el damnificado que el día que se realizó el préstamo el Acusado lo llevo ante una escribana la con día laboral en	
donde el mismo le firmo el "GS" como garantía para que le preste el dinero. Expresando que jamás imagino que el acusado podría hacerle pasar por esta situación, ya que era una persona de confianza y que esta situación le trae aparejado varios problemas económicos por cuanto es el único medio que utiliza para sostener a su familia. Es por ello que solicita de manera urgente se tomen las medidas correspondientes al caso. Es todo cuanto denuncia al respecto.	

Figura 23: Relatos de las denuncias n° 35 y n°36.

Una vez aclaradas las modificaciones a realizar y los motivos de las mismas, se procede a crear una función para realizarlas.

```
#Función para detectar tipo de delito
def tipo_delito(dataframe,col_delito):
    long = len(dataframe[col_delito])
    ind_col_delito = dataframe.columns.get_loc(col_delito)
    for i in range(long):
        if "ROBO" in dataframe.iloc[i,ind_col_delito]:
            dataframe.iloc[i,ind_col_delito] = "Robo"
        elif "HURTO" in dataframe.iloc[i,ind_col_delito]:
            dataframe.iloc[i,ind_col_delito] = "Hurto"
```

Figura 24: Función para clasificar los delitos en "Robo" y "Hurto".

Una vez definida la función, se procede a ejecutarla (ver Anexo II). Por último, se debe corroborar que se haya ejecutado correctamente.

```
#Corroboración
d_denuncias_selec["Delito"]
```

0	Hurto
1	Hurto
2	Hurto
3	Hurto
4	Hurto
...	
3590	Robo
3591	Hurto
3592	Robo
3595	Hurto
3596	Hurto
...	

Figura 25: Corroboración de correcta ejecución de la función de la Figura 24.

Una vez realizadas las modificaciones correspondientes, se procederá a ejecutar el código observado en la Figura 26 para corroborar si hay denuncias repetidas.

```
#Obtener registros duplicados
len(d_denuncias_prep[d_denuncias_prep.duplicated()])

43
```

Figura 26: Obtención de registros repetidos.

Como se observa en la Figura 26, el dataset posee 43 denuncias repetidas, por lo que se procederá a eliminarlas del dataset y corroborarlo.

```
#Eliminación de registros duplicados
d_denuncias_prep = d_denuncias_prep.drop_duplicates()
```

Figura 27: Eliminación de denuncias duplicadas.

```
#Corroboración
len(d_denuncias_prep[d_denuncias_prep.duplicated()])
0
```

Figura 28: Corroboración de una correcta eliminación de denuncias repetidas.

A pesar de haber eliminado algunas denuncias repetidas, mediante una visualización un poco más minuciosa de los datos, se pudo observar que hay denuncias que no son exactamente iguales, por lo que el código antes mostrado no las detecta como repetidas, pero que coinciden en absolutamente todos los valores de sus respectivos campos excepto en el campo “DELITO”, en las que una dice que es un robo y otra dice que es un hurto, pero poseen al resto de campos idénticos entre ellas como se observa en las Figuras 29 y 30.

15	2018-01-02 18:35:43.533	COM. N°100 SAN LORENZO	3	2018	2017-09-29 00:00:00.000	RIO SECO INTERCEPCION RIO SAN LORENZO FINCA LAS COSTAS	HURTO	FISCALIA PENAL N° 1
16	2018-01-02 18:35:43.533	COM. N°100 SAN LORENZO	3	2018	2017-09-29 00:00:00.000	RIO SECO INTERCEPCION RIO SAN LORENZO FINCA LAS COSTAS	ROBO (ART. 184 DEL C.P.)	FISCALIA PENAL N° 1

Figura 29: Denuncias n°15 y n°16.

SE HACE PRESENTE A FINES DE REALIZAR DENUNCIA DEBIDO QUE EN FECHA Y LUGAR DE EPIGRAFE AL MISMO LE HABRIAN SUSTRAI DO LUEGO DE DAÑAR UN ALAMBRADO DE UN POTRERO DEL LUGAR EN MENCIÓN UN CABALLO POTRO MACHO ZAINO OSCURO EDAD 3 AÑOS MCA (), A LO CUAL TRAS REALIZAR AVERIGUACIONES EN EL MES DE DICIEMBRE DEL AÑO 2017 , SE DIO QUE EL ANIMAL SE ENCONTRABA EN LA QUEBRADA DEL SAUZAL , EN LA ÚLTIMA CASA Y AL SEGUIR CON LA AVERIGUACIONES TOMO CONOCIMIENTO QUE EL ANIMAL SE ENCONTRABA EN EL PODER DE , CON QUIEN SE ENTREVISTÓ Y QUIEN LE MANIFESTÓ QUE AL ANIMAL LO HABRÍA COMPRADO EN LA SUMA DE \$ 2.000 DOS MIL PESOS A LA , CON QUIEN SE ENTREVISTÓ Y EXHIBIENDO UN PAPEL FIRMADO POR LA ACUSADA, HABIENDOLE EXPRESADO NO TENER PROBLEMAS EN DEVOLVER EL ANIMAL REQUIRIENDO QUE LA PERSONA QUE LE VENDIÓ DE MALA FE LE DEVUELVA EL DINERO, YA TENÍA QUE TENIR MCA PERO COMO ESTABA PELUNCHANDO NO SE NOTABA, POR LO QUE A TRAVÉS DE ESTA DENUNCIA SOLICITA QUE SE TOMEN LAS MEDIDAS AL CASO EN CONTRA DE LA , YA QUE ESTA LE HABRÍA SUSTRAI DO EL CABALLO Y VENDIDO AL , QUIEN LE DIO PASTAJE Y VACUNO AL ANIMAL. ES TODO CUANTO DENUNCIA.

SE HACE PRESENTE A FINES DE REALIZAR DENUNCIA DEBIDO QUE EN FECHA Y LUGAR DE EPIGRAFE AL MISMO LE HABRIAN SUSTRAI DO LUEGO DE DAÑAR UN ALAMBRADO DE UN POTRERO DEL LUGAR EN MENCIÓN UN CABALLO POTRO MACHO ZAINO OSCURO EDAD 3 AÑOS MCA (), A LO CUAL TRAS REALIZAR AVERIGUACIONES EN EL MES DE DICIEMBRE DEL AÑO 2017 , SE DIO QUE EL ANIMAL SE ENCONTRABA EN LA QUEBRADA DEL SAUZAL , EN LA ÚLTIMA CASA Y AL SEGUIR CON LA AVERIGUACIONES TOMO CONOCIMIENTO QUE EL ANIMAL SE ENCONTRABA EN EL PODER DE , CON QUIEN SE ENTREVISTÓ Y QUIEN LE MANIFESTÓ QUE AL ANIMAL LO HABRÍA COMPRADO EN LA SUMA DE \$ 2.000 DOS MIL PESOS A LA , CON QUIEN SE ENTREVISTÓ Y EXHIBIENDO UN PAPEL FIRMADO POR LA ACUSADA, HABIENDOLE EXPRESADO NO TENER PROBLEMAS EN DEVOLVER EL ANIMAL REQUIRIENDO QUE LA PERSONA QUE LE VENDIÓ DE MALA FE LE DEVUELVA EL DINERO, YA TENÍA QUE TENIR MCA PERO COMO ESTABA PELUNCHANDO NO SE NOTABA, POR LO QUE A TRAVÉS DE ESTA DENUNCIA SOLICITA QUE SE TOMEN LAS MEDIDAS AL CASO EN CONTRA DE LA , YA QUE ESTA LE HABRÍA SUSTRAI DO EL CABALLO Y VENDIDO AL , QUIEN LE DIO PASTAJE Y VACUNO AL ANIMAL. ES TODO CUANTO DENUNCIA.

Figura 30: Relatos de las denuncias n°15 y n°16.

Para tratar este conflicto, se creó la función observada en la Figura 31 para luego ejecutarse (ver Anexo II).

```
#Función para eliminar las filas en las que se repite el
#contenido de una columna determinada
def repetidos(dataframe,columna):
    long = len(dataframe)
    ind_col = dataframe.columns.get_loc(columna)
    repetidos = []
    for i in range(long):
        for j in range(long):
            if i != j:
                if dataframe.iloc[i,ind_col] == dataframe.iloc[j,ind_col]:
                    repetidos.append(dataframe.loc[dataframe[columna] == dataframe.iloc[i,ind_col],:].index.tolist())
    repetidos = np.unique(repetidos).tolist()
    dataframe = dataframe.drop(repetidos)

    return dataframe
```

Figura 31: Función para eliminar denuncias repetidas.

La función de la Figura 31, se encarga de comparar el contenido del campo “RELATO” de cada denuncia, con el contenido del resto de las denuncias en el dataset. En caso de que el contenido de dos denuncias diferentes sea el mismo, entonces se agrega el número de índice de estas denuncias a una lista.

Luego, se obtienen los valores únicos dentro de esta lista de repetidos (con el método “unique” antes nombrado), es decir, que no se repita ningún valor dentro de la lista, para luego eliminar las denuncias correspondientes a los índices incluidos en la lista de repetidos.

5.1.3 Datos No Correspondientes

Como se pudo observar en una etapa anterior (ver sección 4.2), en el dataset se encuentran denuncias correspondientes a otros años además de las de los años 2018 y 2019 como se puede observar en la Figura 32.

```
#Denuncias de otros años
d_denuncias_er = d_denuncias_prep[d_denuncias_prep["Anio"]!= 2018]
d_denuncias_er = d_denuncias_er[d_denuncias_prep["Anio"]!= 2019]
d_denuncias_er
```

	fchConfirmacion	DEPENDENCIA	actaNro	Anio	fchHrHecho
23	2018-01-03 09:56:18.093	DEST. SAN ANTONIO	171	2017	2017-12-30 00:00:00.000
54	2018-01-05 14:52:41.990	COM. N°102 LA LONJA	1607	2017	2018-01-29 00:00:00.000
3589	2019-03-21 22:06:56.497	COM. N°09	1468	2014	2014-06-04 00:00:00.000

Figura 32: Denuncias correspondientes a otros años.

Como se puede apreciar, hay 2 denuncias del año 2017 y 1 del año 2014, así que se va a proceder a eliminar dichas denuncias como se observa en la Figura 33.

```
#Eliminación de denuncias de otros años
d_denuncias_prep = d_denuncias_prep.drop(d_denuncias_prep[d_denuncias_prep["Anio"] == 2014].index)
d_denuncias_prep = d_denuncias_prep.drop(d_denuncias_prep[d_denuncias_prep["Anio"] == 2017].index)
```

Figura 33: Eliminación de denuncias de otros años.

5.1.4 Limpieza de Atributos

Se llevará a cabo la limpieza de dos atributos específicamente, estos son: “*Relato*” y “*LugarHecho*”.

A continuación, se especificará cuál será la limpieza efectuada sobre cada atributo así como su justificación.

5.1.4.1 Atributo: “*Relato*”

La limpieza que se realizará en este atributo, consiste en anonimizar los datos, eliminar las stop words (serán definidas más adelante) y lematizar (proceso que será definido más adelante) los datos obtenidos para poder obtener información de calidad.

5.1.4.1.1 Preparación de Nombres

Como es sabido, las denuncias fueron realizadas por personas reales, por lo que corresponde eliminar tanto sus identidades, sus domicilios específicos, números de DNI, entre otros datos específicos para proteger sus datos personales.

Para este proceso, se llevará a cabo la eliminación de nombres de personas existentes en cada denuncia, para que no se sepa la identidad de las personas participantes, y además, se eliminarán los números encontrados (en formato número, no letra) para eliminar las posibles

direcciones específicas de los domicilios o números de DNI o cualquier otro dato personal de carácter numérico.

Los nombres que se van a utilizar, se obtuvieron de un archivo con extensión “.csv”, descargado de la página web “*datos.gob.ar*”, donde se encuentran todos los nombres registrados en Argentina desde el año 1922 hasta 2015.

Antes de llevar a cabo los procesos, es necesario importar y descargar las librerías “*punkt*” y “*stopwords*” (ver Anexo II).

El primer paso que se va a realizar, es el de preparar los nombres descargados para poder eliminarlos de las denuncias utilizadas. Primeramente, hay que crear un dataframe donde podamos trabajar los datos, como se observa en las Figuras 34 y 35.

```
#Creación dataframe con nombres
d_nombres = pd.read_csv("/content/drive/MyDrive/Proyecto de Grado/historico-nombres.csv")
```

Figura 34: Creación de dataframe con los nombres a utilizar.

d_nombres			
	nombre	cantidad	año
0	Maria	314	1922
1	Rosa	203	1922
2	Jose	163	1922
3	Maria Luisa	127	1922
4	Carmen	117	1922
...	...	...	...
9761604	Antonella Paloma	1	2015
9761605	Briana Princesa Abigail	1	2015
9761606	Neylan Dilara	1	2015
9761607	Laya Jazmín	1	2015
9761608	Fiorella Luz Mailén	1	2015

Figura 35: Nombres y cantidad de nombres registrados por año.

Una vez obtenido el dataframe, se observa que contiene tanto nombres individuales como combinación de estos, por ejemplo observamos el nombre “María” y también “María Luisa” como dos nombres distintos, por lo que el siguiente paso será obtener únicamente los nombres individuales para reducir significativamente la cantidad de datos, y de esta manera, agilizar el procesamiento de los mismos, sobre todo teniendo en cuenta que el proceso consistirá en comparar cada palabra incluida en el relato de cada denuncia con cada uno de los nombres para su posterior eliminación y podría llevar una cantidad de tiempo de procesamiento bastante interesante.

Con el código utilizado, como muestra la Figura 36, se observamos que se obtienen únicamente los nombres individuales, disminuyendo la cantidad de 9.761.609 de nombres a sólo 406.633.

```
#Obtención de nombres individuales
mask=d_nombres['nombre'].str.strip().str.split(' ').str.len().eq(1)
d_nombres2 = d_nombres[mask]
d_nombres2
```

	nombre	cantidad	anio
0	Maria	314	1922
1	Rosa	203	1922
2	Jose	163	1922
4	Carmen	117	1922
5	Ana	113	1922
...	...	...	...
9761435	Kailúa	1	2015
9761488	Ahsley	1	2015
9761489	Georgiy	1	2015
9761544	Dillan	1	2015
9761595	Amaralina	1	2015

Figura 36: Obtención de nombres individuales.

Lo siguiente a realizar, es corroborar si el dataframe posee valores nulos y eliminarlos.

```
#Verificación de nulos
status(d_nombres2)
```

	variable	q_nan	p_nan	q_zeros	p_zeros	unique	type
0	nombre	0	0.0	0	0.0	96734	object
1	cantidad	0	0.0	0	0.0	1567	int64
2	anio	0	0.0	0	0.0	94	int64

Figura 37: Verificación de valores nulos.

Como se observa en la Figura 37, el dataframe no posee valores nulos, así que lo siguiente es corroborar y eliminar aquellos valores repetidos.

```
#Verificación de repetidos
d_nombres2[d_nombres2["nombre"].duplicated()]
```

	nombre	cantidad	anio
10333	Maria	351	1923
10334	Rosa	255	1923
10335	Jose	236	1923
10338	Juana	153	1923
10339	Juan	152	1923
...	...	...	...
9761195	Oriel	1	2015
9761267	Mileva	1	2015
9761299	Octavia	1	2015
9761489	Georgiy	1	2015
9761544	Dillan	1	2015

Figura 38: Nombres repetidos en el dataset.

Ejecutando el código de la Figura 38, se observan los nombres repetidos, siendo estos un total de 309.899. Por lo que, se procederá a eliminarlos como se muestra en la Figura 39.

```
#Eliminación de repetidos  
d_nombres3 = d_nombres2["nombre"].drop_duplicates()
```

Figura 39: Eliminación de nombres repetidos.

Finalmente, se obtuvo un total de 96.734 nombres distintos, una cantidad muy inferior a la inicial, por lo que nos va a agilizar mucho el procesamiento de los datos.

Lo siguiente consiste en darle a los nombres el formato deseado, para poder llevar a cabo el proceso de eliminación de los mismos en las denuncias, para esto se van a tokenizar los nombres obtenidos y se guardarán en una variable como se observa en la Figura 40.

```
#Tokenización de dataframe de nombres  
nombres_p = np.vectorize(tokenizar)  
nombres = nombres_p(list(d_nombres3))  
nombres  
  
array(['maria', 'rosa', 'jose', ..., 'kailúa', 'ahsley', 'amaralina'],  
      dtype='<U27')
```

Figura 40: Tokenización del dataframe que contiene a los nombres a utilizar.

5.1.4.1.2 Stop Words

Las “*Stop Words*” son términos considerados irrelevantes para la minería de texto, ya que no presentan un contenido específico que aporte valor real, además, pueden aparecer con mucha frecuencia en los textos como se puede ver en la bolsa de palabras obtenida anteriormente (ver sección 4.3.3).

Como ejemplo de este tipo de palabras, se pueden contemplar: preposiciones, conjunciones y artículos. Estas palabras deben ser eliminadas de las denuncias para eliminar datos que no aportan valor al análisis que se quiere realizar.

Para utilizarlas, primero se debe colocarlas en una variable como una cadena de string separados por comas (tokens). Por fortuna, la librería “*nltk*” (importada anteriormente) ya cuenta con stop words en distintos idiomas entre los que se encuentra el español, así que únicamente se creará una variable y se colocará en ella las stop words deseadas como se observa en la Figura 41.

```
#Obtención de stopwords
stop_words = nltk.corpus.stopwords.words('spanish')
stop_words

['de',
 'la',
 'que',
 'el',
 'en',
 'y',
 'a',
 'los',
 'del',
 'se',
 'las',
 'por',
 'un',
 'para',
 'con',
 'no',
 'una',
 '.']
```

Figura 41: Obtención de stop words y muestra de algunas de ellas.

5.1.4.1.3 Ejecución de la Limpieza

Una vez que se obtuvieron tanto los nombres, como las stop words que se quieren quitar de las denuncias, lo único que resta es compararlos con cada palabra incluida en las denuncias y eliminar las que coincidan.

Para cumplir este cometido, se creó la función mostrada en la Figura 42, donde no solo elimina los nombres, sino también los números encontrados (para eliminar datos personales numéricos) y las stop words.

Figura 42: Función para tokenizar, anonimizar y eliminar stop words.

```
#Ejecución de la limpieza
prep = np.vectorize(limpiar)
prep_f = prep(list(d_denuncias_prep["RELATO"]))
prep_f

array(['primeramente aclara radica denuncia representacion sobrino',
      'primeramente manifiesta acusada madre ex pareja sra separo l',
      'dia fecha hora epigrafe circunstancia enc ntraba hospital in',
      ...,
      'deja aclarado pareja sr lapso quince años común cuatro hijo:',
      'encuentra hace cuatro años pareja sr reside asentamiento bº',
      'radica ex pareja sr dllo vº velata psje palmas casa matuvo',
      dtype='<U8889')]
```

Figura 43: Ejecución de la función de limpieza de la Figura 42.

La lematización consiste en convertir una palabra a su forma base o lema. Por ejemplo las palabras “soy”, “son” y “es” son conjugaciones del verbo “ser”, por lo que el lema de esas palabras sería “ser”.

Para llevar a cabo este procedimiento sobre las denuncias trabajadas, lo primero a realizar es importar/descargar la librería “*spacy*” (ver Anexo II).

```
#Función para lematizar
def lematizar(vector):
    #Longitud del vector a trabajar
    long = len(vector)
    #Creación del vector final
    vec_final = vector
    #Ciclo para lematizar palabra por palabra de cada elemento
    for i in range(long):
        #Formateo del vector de entrada
        doc = nlp(str(vector[i]))
        #Obtención de las palabras lematizadas
        lemmas = [tok.lemma_.lower() for tok in doc]
        #Asignación de las palabras lematizadas al vector final
        vec_final[i] = str(lemmas)
        # Eliminación de signos de puntuación
        regex = '[!@#$%^&*(){}~|:;'\",.-_+=\[\]`~\s\p{Punct}]'
        vec_final[i] = re.sub(regex, '', vec_final[i])
        # Eliminación de espacios en blanco múltiples
        vec_final[i] = re.sub("\s+", ' ', vec_final[i])

    return vec_final
```

Una vez que creada la función que se va a utilizar para lematizar las denuncias, lo siguiente sería ejecutarla sobre las denuncias (ver Anexo II).

Figura 45: Denuncias luego de ser lematizadas.

Una vez que se llevaron a cabo los distintos procesos de limpieza, corresponde reemplazar los datos obtenidos por los datos que se encuentran en el dataframe sin “limpiar” como se observa en la Figura 46.

Figura 46: Reemplazo de datos del atributo “Relato”.

Si bien ya se poseen denuncias anónimas y sin stop words, hay información útil que ha sido eliminada ya que, tanto el nombre de los barrios como de las calles contiene nombres y/o números y por ende, se perdió información útil sobre la ubicación de los delitos.

Para llevar a cabo el proceso de eliminación, antes se deben tokenizar las palabras contenidas en la columna “*lugarHecho*” y luego sí realizar el descarte de stop words, así como se muestra en las Figuras 47 y 48.

Figura 47: Creación de función para tokenizar y eliminar las stop words.

```
#Eliminación de stopwords
lugares = np.vectorize(sw)
lugares_f = lugares(list(d_denuncias_prep["lugarHecho"]))
lugares_f

array(['salon eventos terrazas bº almudena',
      'bº 2º junio calle david zambrano 94',
      'hospital materno infantil milagro', ...,
      'paraje fuerte alto banda sur cachi',
      'asentamiento bº 13 abril s nº', 'vº velata joaquin corbalan s n'],
      dtype='<U88')
```

Figura 48: Eliminación de stop words de la columna “lugarHecho”.

5.1.4.2.2 Reemplazo de Datos

Una vez obtenidos los datos (lugares de hecho de las denuncias) sin stop words (Figura 48), corresponde reemplazarlos por los datos que se encuentran en el dataframe sin “limpiar” como se observa en la figura 49.

```
#Reemplazo de columna "lugarHecho"
long = len(lugares_f)
ind_col_lugar = d_denuncias_prep.columns.get_loc("lugarHecho")
for i in range(long):
    d_denuncias_prep.iloc[i,ind_col_lugar] = lugares_f[i]
```

Figura 49: Reemplazo de datos del atributo “lugarHecho”.

De esta manera, quedaría el dataframe listo para la siguiente etapa.

5.2 Selección de Atributos

En esta etapa del proyecto, se hará una selección de los atributos del dataframe que se utilizarán en el proceso de clustering/agrupamiento. Se van a determinar los motivos de la exclusión o inclusión de cada atributo, así como también las modificaciones realizadas en caso de ser necesario.

Cabe destacar que no se volverá a definir el significado de cada atributo debido a que ya fue definido anteriormente (ver sección 4.3.1).

Una vez realizada la copia pertinente del dataframe (ver Anexo II), se procede a mostrar nuevamente los nombres de los atributos actuales.

```
#Atributos del dataframe  
d_denuncias_selec.columns.values  
  
array(['fchConfirmacion', 'DEPENDENCIA', 'actaNro', 'Anio', 'fchHrHecho',  
      'lugarHecho', 'DELITO', 'FISCALIA', 'RELATO'], dtype=object)
```

Figura 50: Nombres de los atributos del dataframe.

5.2.2 FchConfirmación

Este atributo no será tenido en cuenta para el proceso de clustering, debido a que no es considerada útil la fecha en la que se realizó la denuncia.

```
#Eliminación de atributo  
d_denuncias_selec = d_denuncias_selec.drop(columns='fchConfirmacion')
```

Figura 51: Eliminación del atributo “fchConfirmación”.

5.2.3 Dependencia

Este atributo no será tenido en cuenta para el proceso de clustering, debido a que no es considerada útil la ubicación en la que se llevó a cabo la denuncia del hecho.

```
#Eliminación de atributo  
d_denuncias_selec = d_denuncias_selec.drop(columns='DEPENDENCIA')
```

Figura 52: Eliminación del atributo “DEPENDENCIA”.

5.2.4 ActaNro

Este atributo no será tenido en cuenta para el proceso de clustering, debido a que no es considerado útil el número de acta de cada denuncia.

```
#Eliminación de atributo  
d_denuncias_selec = d_denuncias_selec.drop(columns='actaNro')
```

Figura 53: Eliminación del atributo “actaNro”.

5.2.5 Anio

Este atributo no será tenido en cuenta para el proceso de clustering, debido a que no es considerado útil el año en el que se realizó la denuncia.

```
#Eliminación de atributo
d_denuncias_selec = d_denuncias_selec.drop(columns='Anio')
```

Figura 54: Eliminación del atributo “Anio”.

5.2.6 FchHrHecho

Este atributo sí será tenido en cuenta para el proceso de clustering, pero no en su totalidad.

Recordemos que este campo posee el formato año/mes/día/hora, de todos estos datos solo se utilizarán el mes y el día en el que se llevó a cabo el delito.

Aunque, no se considerará directamente el día en el que suceden los delitos, sino que, se considerará si ocurrieron en la primera quincena del mes o en la segunda, por lo que se crearán dos atributos nuevos y se eliminará este.

El año no se tendrá en cuenta porque no se considera relevante y si bien, la hora si se considera relevante, a simple vista se observa que la gran mayoría de las denuncias carece de este dato, por lo que finalmente no se tendrá en cuenta.

Lo primero, consiste en crear los nuevos atributos “MesHecho” y “DíaHecho” como se observa en la Figura 55.

```
#Creación de nuevos atributos
d_denuncias_selec = d_denuncias_selec.assign(MesHecho=" ", DíaHecho=" ")
d_denuncias_selec.head()
```

fchHrHecho	lugarHecho	DELITO	FISCALIA	RELATO	MesHecho	DíaHecho
------------	------------	--------	----------	--------	----------	----------

Figura 55: Creación de los atributos “MesHecho” y “DíaHecho”.

El siguiente paso a realizar, consiste en crear una función que de acuerdo al mes y el día detectado en los datos del atributo “fchHrHecho”, se agreguen dichos datos a las columnas recién creadas. Para ello, lo primero a realizar es importar la librería “datetime” (ver Anexo II).

Ahora que se posee la librería necesaria, se hará uso de ella para construir la función antes descrita como se observa en la Figura 56. Cabe aclarar que los días se dividirán en

“Primera_Quincena” y “Segunda_Quincena”, mientras que los meses serán reemplazados por sus nombres.

```
#Función para detectar día y mes
def fecha(dataframe,col_fecha,col_mes,col_dia):
    long = len(dataframe[col_fecha])
    ind_col_fecha = dataframe.columns.get_loc(col_fecha)
    ind_col_dia = dataframe.columns.get_loc(col_dia)
    ind_col_mes = dataframe.columns.get_loc(col_mes)
    for i in range(long):
        fecha_dt = datetime.strptime(dataframe.iloc[i,ind_col_fecha], '%Y-%m-%d %H:%M:%S.%f')
        if (fecha_dt.day >= 1) & (fecha_dt.day <= 15):
            dataframe.iloc[i,ind_col_dia] = "Primera_Quincena"
        elif (fecha_dt.day >= 16) & (fecha_dt.day <= 31):
            dataframe.iloc[i,ind_col_dia] = "Segunda_Quincena"
        if (fecha_dt.month == 1):
            dataframe.iloc[i,ind_col_mes] = "Enero"
        elif (fecha_dt.month == 2):
            dataframe.iloc[i,ind_col_mes] = "Febrero"
        elif (fecha_dt.month == 3):
            dataframe.iloc[i,ind_col_mes] = "Marzo"
        elif (fecha_dt.month == 4):
            dataframe.iloc[i,ind_col_mes] = "Abril"
        elif (fecha_dt.month == 5):
            dataframe.iloc[i,ind_col_mes] = "Mayo"
        elif (fecha_dt.month == 6):
            dataframe.iloc[i,ind_col_mes] = "Junio"
        elif (fecha_dt.month == 7):
            dataframe.iloc[i,ind_col_mes] = "Julio"
        elif (fecha_dt.month == 8):
            dataframe.iloc[i,ind_col_mes] = "Agosto"
        elif (fecha_dt.month == 9):
            dataframe.iloc[i,ind_col_mes] = "Septiembre"
        elif (fecha_dt.month == 10):
            dataframe.iloc[i,ind_col_mes] = "Octubre"
        elif (fecha_dt.month == 11):
            dataframe.iloc[i,ind_col_mes] = "Noviembre"
        elif (fecha_dt.month == 12):
            dataframe.iloc[i,ind_col_mes] = "Diciembre"
```

Figura 56: Función para detectar día y mes de la columna “fchHrHecho” y agregarlos a las nuevas columnas.

Una vez ejecutada la función (ver Anexo II), se procede a ver el resultado.

MesHecho	DiaHecho
Diciembre	Segunda_Quincena
Diciembre	Segunda_Quincena
Enero	Primera_Quincena
Enero	Primera_Quincena
Diciembre	Segunda_Quincena
...	...
Septiembre	Primera_Quincena
Octubre	Primera_Quincena
Noviembre	Primera_Quincena
Noviembre	Segunda_Quincena
Noviembre	Segunda_Quincena

Figura 57: Resultado de la ejecución de la función de la Figura 56.

Finalmente, se procederá a eliminar el atributo “fchHrHecho”.

```
#Eliminación de atributo
d_denuncias_selec = d_denuncias_selec.drop(columns='fchHrHecho')
```

Figura 58: Eliminación del atributo “fchHrHecho”.

5.2.7 LugarHecho

Este atributo sí será tenido en cuenta para el proceso de clustering, debido a que sí se considera como información fundamental el lugar donde se lleva a cabo el delito.

Luego del proceso de limpieza realizado anteriormente (ver sección 5.1.4.4), solamente se procederá a renombrar la columna, cambiando su nombre de “lugarHecho” a “LugarHecho”.

```
#Renombramiento de atributo
d_denuncias_selec.rename(columns={"lugarHecho": "LugarHecho"})
```

Figura 59: Renombramiento del atributo “lugarHecho”.

5.2.8 Delito

Este atributo sí será tenido en cuenta para el proceso de clustering, debido a que sí se considera como información fundamental el tipo de delito manifestado en cada denuncia.

Se realizará una modificación sobre esta columna, consiste en renombrar la columna, pasando de llamarse “*DELITO*” a “*Delito*”.

```
#Renombramiento de atributo  
d_denuncias_selec = d_denuncias_selec.rename(columns={"DELITO":"Delito"})
```

Figura 60: Renombramiento del atributo “*DELITO*”.

5.2.9 Fiscalia

Este atributo no será tenido en cuenta para el proceso de clustering, debido a que no es considerado útil saber a la jurisdicción de qué fiscalía corresponde cada denuncia.

```
#Eliminación de atributo  
d_denuncias_selec = d_denuncias_selec.drop(columns='FISCALIA')
```

Figura 61: Eliminación del atributo “*FISCALIA*”.

5.2.10 Relato

Este atributo sí será tenido en cuenta para el proceso de clustering, debido a que sí se considera como información fundamental al relato sobre los hechos incluido en cada denuncia.

Solo se realizará un cambio de nombre de este atributo, pasará de llamarse “*RELATO*” a “*Relato*”.

```
#Renombramiento de atributo  
d_denuncias_selec = d_denuncias_selec.rename(columns={"RELATO":"Relato"})
```

Figura 62: Renombramiento del atributo “*RELATO*”.

5.2.11 Selección final

En resumen, luego de ser justificada la elección de cada uno, estos son los atributos seleccionados para llevar a cabo el proceso de clustering.

```
d_denuncias_selec.columns  
Index(['LugarHecho', 'Delito', 'Relato', 'MesHecho', 'DiaHecho'], dtype='object')
```

Figura 63: Atributos seleccionados para efectuar el proceso de clustering.

5.3 Construcción de la Bolsa de Palabras

Luego de realizar los procesos de limpieza sobre los datos, se creará nuevamente una bolsa de palabras con estos datos limpios, para poder observar las diferencias con la bolsa de palabras obtenida a partir del conjunto de datos inicial (ver sección 4.3.3).

Como siempre, antes de trabajar con los datos se realizó una copia del dataframe (Anexo II)

Se utilizará la función mostrada en la Figura 64, que se diferencia de la usada anteriormente (ver sección 4.3.3) porque esta función también se encarga de tokenizar la columna a trabajar a diferencia de la anterior función donde tomaba datos ya tokenizados.

```
#Función para mostrar los términos con su respectiva frecuencia de una  
#bolsa de palabras creada a partir de una columna  
def bow_f(dataframe, columna):  
    long_bow = len(dataframe[columna])  
    wordfreq = {}  
    ind_col = dataframe.columns.get_loc(columna)  
    for i in range(long_bow):  
        tokens = nltk.word_tokenize(str(dataframe.iloc[i,ind_col]))  
        for token in tokens:  
            if token not in wordfreq.keys():  
                wordfreq[token] = 1  
            else:  
                wordfreq[token] += 1  
    sorted_bow = sorted(wordfreq.items(), key=lambda kv: kv[1], reverse=True)  
    return sorted_bow
```

Figura 64: Función para obtener cada término de una bolsa de palabras y su frecuencia.

Luego de ejecutar la función (ver Anexo II), se observan en la Figura 65 los 20 términos más repetidos.

```
#Los 20 términos más frecuentes
#luego de la limpieza
bow_f[:20]

[('color', 5692),
 ('hacer', 4251),
 ('encontrar', 3832),
 ('fecho', 3424),
 ('personar', 3158),
 ('mismo', 2822),
 ('celular', 2721),
 ('denunciar', 2707),
 ('hora', 2639),
 ('lugar', 2638),
 ('dicente', 2623),
 ('sustraer', 2311),
 ('ser', 2289),
 ('masculino', 2140),
 ('solicitar', 2106),
 ('cuanto', 2078),
 ('mca', 2004),
 ('o', 1918),
 ('dejar', 1888),
 ('casar', 1797)]
```

Figura 65: Los 20 términos más frecuentes en la bolsa de palabras.

Esta vez, sí que se observan palabras significativas para el análisis, y no solo stop words como obtuvimos en la primera bolsa de palabras construida a partir de los datos iniciales sin pasar por el proceso de limpieza.

5.4 Construcción de los Datos

Para llevar a cabo el clustering de textos, hace falta representar los datos de una forma específica para que pueda aplicarse el algoritmo de agrupamiento sobre ellos.

La forma usada en este caso será la de representar los datos mediante el modelo vectorial, es decir, crear un vector de palabras para cada denuncia.

A su vez, a cada palabra o token de cada denuncia se le asociará un valor numérico dependiendo de la frecuencia del mismo en el documento, para asignar este valor numérico se utilizará “*TF-IDF*” como se explica en el siguiente paso.

5.4.1 Importancia de las Palabras

La importancia de cada palabra/token será calculada con “*TF-IDF*”.

Tf-Idf son las siglas de Term Frequency(TF) - Inverse Document Frequency(IDF), (Frecuencia de término - Frecuencia inversa de documento), es una medida numérica que expresa cuán relevante es una palabra para un documento en una colección. El valor de Tf-idf, aumenta proporcionalmente a la frecuencia de una palabra, pero es compensada por la

frecuencia de esa palabra en la colección de documentos, lo que permite manejar el hecho de que algunas palabras son más comunes en la colección.

Como Tf-idf es el producto de dos medidas, vamos a mostrar por separado su formulación.

La frecuencia de término (TF) viene dado por la frecuencia de un término “ t ”, en un documento “ d ”, dividido por la frecuencia del término más frecuente (máximo) en el documento:

$$tf(t, d) = \frac{f(t, d)}{\max\{f(w, d) : w \in d\}}$$

Figura 66: Fórmula de “TF”.

La frecuencia inversa de documento, es la importancia de una palabra en el conjunto de documentos y se calcula como:

$$idf(t, D) = \log \frac{|D|}{|\{d \in D : t \in d\}|}$$

Figura 67: Fórmula de “IDF”.

Siendo “ $|D|$ ” la cardinalidad de “ D ”, o número de documentos en la colección y siendo “ $|\{d \in D : t \in d\}|$ ” el número de documentos donde aparece el término “ t ”.

Una vez explicada conceptualmente la forma de calcular la importancia de las palabras, se creará el vector de palabras y se obtendrá el valor de cada token/palabra.

5.4.1.1 N-Gramas

Antes de la creación del vector de palabras, se procederá a incorporar los n-gramas (un n-grama es una subsecuencia de n elementos, en este caso cada elemento es un token/palabra, de una secuencia dada) de longitud 2 (bigramas) y de longitud 3 (trigramas) a las denuncias que fueron tratadas mediante el proceso de limpieza anteriormente ejecutado.

Si bien la librería “*TfidfVectorizer*”, librería que se va a utilizar en el siguiente paso para la creación del vector de palabras, cuenta con un parámetro que nos permite tener en cuenta los n-gramas deseados, este procedimiento aumenta de manera exponencial el número

de elementos a ingresar al algoritmo de modelado y excede la capacidad que nos brinda Google Colaboratory, excediendo específicamente la memoria RAM disponible (12Gb).

Aún así, se considera importante la posibilidad de agregar dichos n-gramas así que se van a agregar “manualmente” a los datos que vayan a ingresar al algoritmo de agrupamiento, pero no todos los bigramas y trigramas, sino los 100 bigramas y 50 trigramas más frecuentes. A continuación se explica este proceso.

Lo primero a realizar será importar las librerías “*ngrams*” y “*heapq*” necesarias para llevar a cabo el proceso (ver Anexo II).

Como en todo el proceso, primero se hará una copia del dataframe obtenido del proceso anterior (ver Anexo II).

A continuación, se creará una lista con todas las denuncias seleccionadas luego del proceso de limpieza, unificando los valores de los atributos seleccionados anteriormente (ver sección 5.2) debido a que este es el formato en el que hay que introducir los datos en el siguiente paso.

```
#Combinación de columnas
denuncias = []
ind_col_lugar = d_denuncias_final.columns.get_loc("LugarHecho")
ind_col_delito = d_denuncias_final.columns.get_loc("Delito")
ind_col_relato = d_denuncias_final.columns.get_loc("Relato")
ind_col_mes = d_denuncias_final.columns.get_loc("MesHecho")
ind_col_dia = d_denuncias_final.columns.get_loc("DiaHecho")

for i in range(len(d_denuncias_final['Relato'])):
    denuncias.append(str(d_denuncias_final.iloc[i,ind_col_lugar])+ ' '+str(d_denuncias_final.iloc[i,ind_col_delito))+\
        ' '+str(d_denuncias_final.iloc[i,ind_col_relato])+ ' '+str(d_denuncias_final.iloc[i,ind_col_mes))+\
        ' '+str(d_denuncias_final.iloc[i,ind_col_dia]))
```

Figura 68: Creación de lista con los datos seleccionados de cada denuncia.

Lo siguiente a realizar, consiste en definir una función que permita agregar los bigramas deseados a nuestras denuncias.

Figura 69: Función para agregar los 100 bigramas más frecuentes a las denuncias.

Por último, debemos definir una función que permita agregar los trigramas deseados a nuestras denuncias.

Como se observa en la Figura 71, se utilizó la función “*TfidfVectorizer*” que se encarga de convertir en tokens a los documentos, aprender el vocabulario y asignar las ponderaciones correspondientes formando la matriz tf-idf deseada.

En la Figura 71, también se observa que se utilizó la función “*fit_transform*” que se encarga de calcular la media y la varianza de los datos, y basado en esos valores estadísticos, los lleva los a una distribución normal estándar (media igual a 0 y desviación estándar igual a 1) para que el modelo de agrupamiento no tienda hacia una característica particular del conjunto de datos.

Capítulo 6 - Modelado

En esta etapa se desarrollan los conceptos utilizados para la construcción y evaluación del modelo de agrupamiento, así como el análisis y obtención de conclusiones a partir de los resultados obtenidos.

6.1 Métrica de Evaluación

Cuando los datos del dataset no se encuentran agrupados previamente, como es este caso, de hecho eso es lo que se quiere lograr, la evaluación debe realizarse utilizando el propio modelo, por ello, para evaluar el modelo de agrupamiento/clustering de este proyecto se va a hacer uso del “Coeficiente de Silueta” (*Silhouette Coefficient* en inglés) que es un ejemplo de este tipo de evaluación.

En el coeficiente de silueta, la obtención de una valoración mayor corresponde a un modelo con clusters mejor definidos, esta valoración se basa en dos valores:

- **a**: La distancia media entre una muestra y todos los demás puntos de la misma clase.
- **b**: La distancia media entre una muestra y todos los demás puntos en el siguiente grupo más cercano.

El coeficiente de silueta “*s*” para una sola muestra se da como:

$$s = \frac{b - a}{\max(a, b)}$$

Figura 72: Coeficiente de silueta para una muestra (Scikit Learn, s.f.).

El coeficiente de silueta para un conjunto de muestras, se proporciona como la media del coeficiente de silueta para cada muestra.

La valoración de esta evaluación está limitada entre -1 y 1. La valoración es más alta cuando los clusters son densos y están bien separados de los otros clusters, una valoración cercana a -1 hablaría de una agrupación errónea y una valoración alrededor de 0 estaría indicando que hay clusters superpuestos (Scikit Learn, s.f.).

6.2 Construcción de los Modelos

Los modelos que van a construirse en este proyecto se crearán utilizando los siguientes algoritmos de agrupamiento/clustering:

- **K-means**
- **DBSCAN**
- **Agrupamiento jerárquico**

Se elegirá el mejor modelo utilizando cada algoritmo, para luego elegir aquel que obtenga una valoración mayor entre ellos de su coeficiente de silueta (tomando cuatro decimales sin redondear).

6.2.1 K-Means

Una vez que se posee el vector de palabras (obtenido en la sección 5.4.1.1), lo siguiente es aplicar sobre este el algoritmo deseado, en este caso K-means, que fue definido anteriormente en el “Estado de la Cuestión” (ver sección 2.4.4.1).

Para efectuar este procedimiento, lo primero a realizar es importar la librería “*Kmeans*” (ver Anexo II).

Lo siguiente a realizar, es elegir el valor de “*k*” para efectuar el algoritmo K-means, esto se realizará en la siguiente sección.

6.2.1.1 Elección del Valor de “*k*”

Para llevar a cabo el algoritmo K-means, es necesario seleccionar un valor “*k*” del número de grupos en los que se agrupan los datos.

En general, no hay una forma exacta de determinar el número de grupos, pero se pueden usar ciertas reglas o valores estadísticos que ayudan a estimar el número de grupos.

En este caso, para determinar el valor de “*k*” se usará el método del codo para intentar aproximar el valor adecuado para luego elegir el valor de “*k*” que maximice la valoración del coeficiente de silueta.

6.2.1.1.1 Método del Codo

El método del codo utiliza la distancia media de las observaciones a su centroide (centro de cada cluster). Es decir, se fija en las distancias intra-cluster o inercia.

Cuanto más grande es el número de clusters “*k*”, la varianza intra-cluster tiende a disminuir. Cuanto menor sea la distancia intra-cluster mejor, ya que significa que los clústers son más compactos.

El método del codo busca el valor “ k ” que satisfaga que un incremento de “ k ”, no mejore sustancialmente la distancia media intra-cluster, gráficamente esto se ve representado mediante un “codo”, es decir, el punto de inflexión de la curva.

Para poder realizar el gráfico y poder visualizar cuál podría ser el valor adecuado de “ k ” debemos importar la librería “*matplotlib.pyplot*” necesaria (ver Anexo II) para poder graficar.

Lo siguiente consiste en crear y ejecutar una función que permita ejecutar el método del codo.

```
#Función para ejecutar el Método del Codo
def codo(lim_inf,lim_sup):
    k = range(lim_inf,lim_sup)
    inertias = []
    for i in k:

        #Ejecución de kmeans
        km = KMeans(n_clusters=i, max_iter=1000, random_state=26)
        km.fit(features)

        #Obtención de inercia
        inertias.append(km.inertia_)

    #Graficado
    plt.plot(k, inertias, 'o-')
    plt.xlabel("Valor de k")
    plt.ylabel("Inercia")
    plt.title('Método del Codo')
    plt.show()
```

Figura 73: Función para utilizar el método del codo.

Lo que se realiza en la Figura 73, consiste en crear un rango de posibles valores de “ k ”, en este caso se eligieron los valores 2 y 35 para que sean el límite inferior y el límite superior correspondientes, se eligieron estos valores para abarcar una gran cantidad de posibles valores de “ k ”.

Luego, se efectúa el algoritmo k-means con cada valor entero de “ k ” que se encuentre dentro del rango definido, de esta ejecución del algoritmo se extrae el valor de la inercia correspondiente al resultado obtenido con ese parámetro y se agrega a una lista que contiene las inercias obtenidas de cada valor de “ k ”.

Una vez que se obtuvieron todos los valores de la inercia, se procede a graficar todos los valores dentro de la lista.

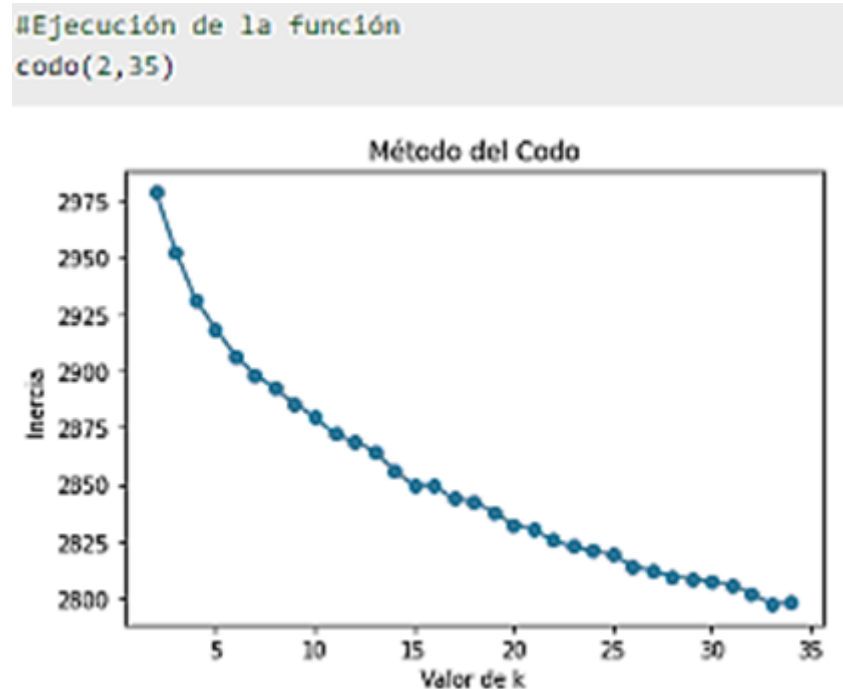


Figura 74: Ejecución de la función de la Figura 73.

Observando la Figura 74, no se podría decir que se visualice un “codo” que indique un posible valor ideal de “ k ”, por lo que se deberá determinar el valor de “ k ” de otra manera.

6.2.1.1.2 Coeficiente de Silueta

Al no obtener un posible valor de “ k ” en el método del codo, se procederá a comparar el coeficiente de silueta obtenido de cada modelo k-means ejecutado, variando el valor de “ k ” en un determinado rango, para poder obtener el valor de “ k ” deseado.

Lo primero a realizar será importar la librería “silhouette_score”.

Luego, se debe crear y ejecutar la función que realice el procedimiento antes planteado.

```
#Función para determinar el valor de k que obtenga
#el mayor valor de coeficiente de silueta
def coef_silueta(lim_inf,lim_sup):
    k = range(lim_inf,lim_sup)
    silhouette_avg = []
    for num_clusters in k:

        #Ejecución de kmeans
        kmeans = KMeans(n_clusters=num_clusters, random_state=26)
        kmeans.fit(features)
        cluster_labels = kmeans.labels_

        #Coeficiente de silueta
        silhouette_avg.append(silhouette_score(features, cluster_labels, random_state=1))

    #Graficado
    plt.plot(k,silhouette_avg,"bx-")
    plt.xlabel("Valores de K")
    plt.ylabel("Coeficiente de la Silueta")
    plt.title("Análisis del Coeficiente de Silueta")
    plt.show()
```

Figura 75: Función para obtener el valor de “k” mediante su coeficiente de silueta.

Como se observa en la Figura 75, la función nos pide únicamente un rango de valores para incrementar el valor de “k”, luego se encarga de utilizar el algoritmo k-means con cada valor de “k” dentro del rango definido, obtener el coeficiente de silueta de cada ejecución del algoritmo y agregarlo a una lista y, por último, se realiza el graficado de la lista obtenida.

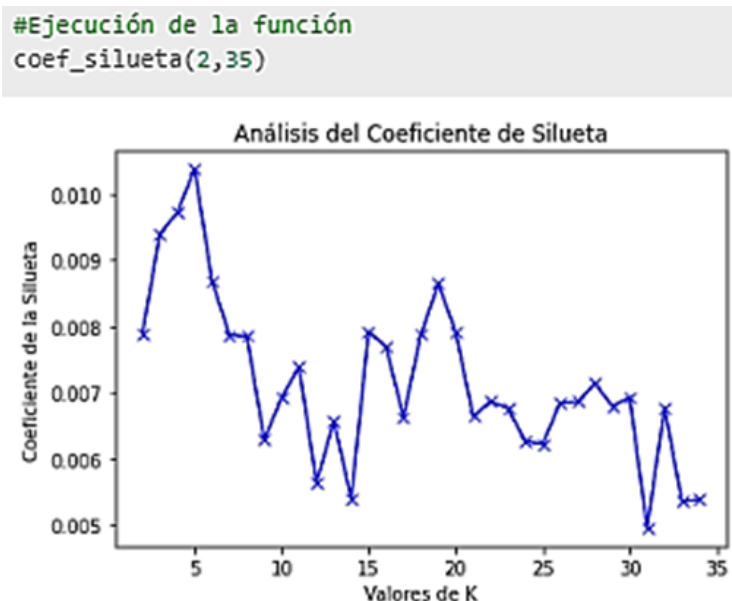


Figura 76: Ejecución de la función de la Figura 75.

Como se observa en la Figura 76, se ejecutó la función utilizando los valores 2 y 35 para definir el rango de valor posibles para “ k ”, se eligió este rango para abarcar una buena cantidad de valores.

Observando la gráfica, se puede apreciar claramente que el valor que obtiene el coeficiente de silueta más alto es “ $k = 5$ ”.

También se puede observar que los valores obtenidos con valores mayores de “ k ” son bajos comparados con el observado como ideal, por lo que no pareciera ser necesario utilizar más valores que los planteados.

Por último, solo resta determinar el valor exacto del coeficiente de silueta obtenido a partir de la ejecución del algoritmo k-means con “ $k = 5$ ” (mejor valor obtenido) para poder compararlo más adelante.

```
#Obtención de coeficiente de silueta
kmeans = KMeans(n_clusters=5, random_state=26)
kmeans.fit(features)
cluster_labels = kmeans.labels_

sc = silhouette_score(features, cluster_labels, random_state=1)
sc

0.010375681487775904
```

Figura 77: Coeficiente de silueta con “ $k=5$ ” utilizando K-means.

Como se observa en la Figura 77, el coeficiente de silueta obtenido es igual a 0,0103.

6.2.2 DBSCAN

Este algoritmo ya fue definido anteriormente en el “Estado de la Cuestión” del presente proyecto (ver sección 2.4.4.3).

Por lo que el siguiente paso a realizar, es elegir el valor de los parámetros necesarios para ejecutar el algoritmo.

6.2.2.1 Elección de Parámetros

Como se nombró anteriormente, el algoritmo DBSCAN requiere de la definición obligatoria de dos parámetros por parte del usuario para poder ejecutarse:

- **Muestras mínimas (“ $min_samples$ ”):** El menor número de puntos necesarios para formar un cluster.

- **Épsilon (“*eps*”):** La distancia máxima a la que dos puntos pueden estar entre sí mientras aún pertenecen al mismo grupo (Mullin T., 2020).

A continuación, se van a definir los mejores valores para estos parámetros.

6.2.2.1.1 *Nearest Neighbors*

En esta sección, se encontrará un valor adecuado para épsilon calculando la distancia a los “*n*” puntos más cercanos para cada punto, clasificando y graficando los resultados.

Luego, similar al método del codo usado en k-means, se visualiza dónde el cambio es más pronunciado en la gráfica y se selecciona como valor épsilon.

Para calcular la distancia de cada punto a su vecino más cercano, se va a usar “*NearestNeighbors*” (vecinos más cercanos por su traducción al español), un algoritmo que se encarga justamente de ese objetivo. El punto en sí, está incluido en los “*n*” vecinos.

El método “*kneighbors*” devuelve dos matrices, una que contiene la distancia a los puntos “*n_neighbors*” más cercanos y la otra que contiene el índice de cada uno de esos puntos.

Primeramente se va a importar la librería “*NearestNeighbors*” (ver Anexo II).

Luego se ejecuta el algoritmo “*NearestNeighbors*” antes mencionado.

```
#Cálculo de los vecinos más cercanos
neighbors = NearestNeighbors(n_neighbors=2)
neighbors_fit = neighbors.fit(features)
distances, indices = neighbors_fit.kneighbors(features)
```

Figura 78: Cálculo de los vecinos más cercanos.

A continuación, se ordenan y grafican los resultados obtenidos para poder detectar el posible valor de épsilon.

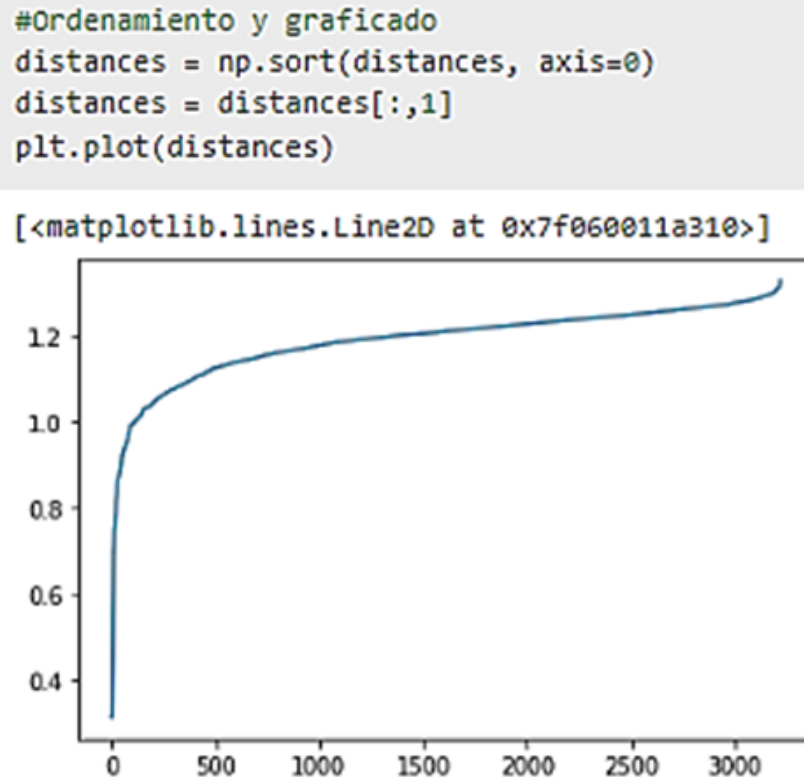


Figura 79: Ordenamiento y graficado de los valores obtenidos del cálculo de la Figura 78.

El valor óptimo de ϵ debería encontrarse en la curvatura máxima, en este caso tomaremos un rango de posibles valores, de acuerdo a la Figura 79 el valor óptimo de ϵ debería estar entre 0.8 y 1.2 incluidos.

6.2.2.1.2 Coeficiente de Silueta

Como se dijo anteriormente, el modelo a elegir será aquel que maximice el valor del coeficiente de silueta, por lo que se calcularán los valores de silueta variando los parámetros en el algoritmo DBSCAN.

Antes que nada, es necesario importar la librería “*DBSCAN*” (ver Anexo II).

Para determinar el mejor modelo, se creó la siguiente función.

```
#Función para determinar los mejores valores de epsilon y min_samples
def coef_silueta_dbscan(eps_inf,eps_sup,ms_inf,ms_sup):

    #Definición del rango de hiperparámetros
    eps_list=np.arange(start=eps_inf, stop=eps_sup, step=0.1)
    min_sample_list=np.arange(start=ms_inf, stop=ms_sup, step=1)

    #Creación de un dataframe vacío para almacenar los resultados
    silhouette_scores_data=pd.DataFrame()

    for eps_trial in eps_list:
        for min_sample_trial in min_sample_list:

            #Generación de clusters con DBSCAN
            db = DBSCAN(eps=eps_trial, min_samples=min_sample_trial)

            #Cálculo del coeficiente de silueta de cada prueba
            if(len(np.unique(db.fit_predict(features)))>2):
                sil_score=silhouette_score(features, db.fit_predict(features), random_state=1)
            else:
                continue
            trial_parameters="eps:" + str(eps_trial.round(1)) + " min_samples :" + str(min_sample_trial)

            #Agregado del valor del coeficiente de silueta y del valor de los parámetros utilizados
            silhouette_scores_data=silhouette_scores_data.append(pd.DataFrame(data=[[sil_score,trial_parameters]],\
                                                                                columns=["score", "parameters"]))

    #Obtención del valor más alto del coeficiente de silueta
    #con sus respectivos parámetros
    value = silhouette_scores_data.sort_values(by='score', ascending=False).head(1)

    return value
```

Figura 80: Función para encontrar el mejor modelo utilizando DBSCAN.

La función de la Figura 80, solicita como parámetros un rango para variar los valores de epsilon y otro rango para variar el valor de las muestras mínimas.

Debido la Figura 79, obtenida anteriormente con “*NearestNeighbors*”, el rango elegido para los valores de epsilon está entre 0.7 y 1.3.

El rango elegido para las muestras mínimas está entre 2 y 15, fueron elegidos simplemente para abarcar una buena cantidad de valores.

Una vez definidos los rangos de los valores, se crea un dataframe vacío para almacenar tanto el valor de coeficiente de silueta así como los parámetros utilizados para obtener dicho valor.

Entonces se procede a iterar variando los parámetros dentro del rango elegido y guardando los valores antes mencionados dentro del dataframe. Una vez que termina de iterar se muestra el valor de silueta más alto así como los parámetros correspondientes a la obtención de ese valor.

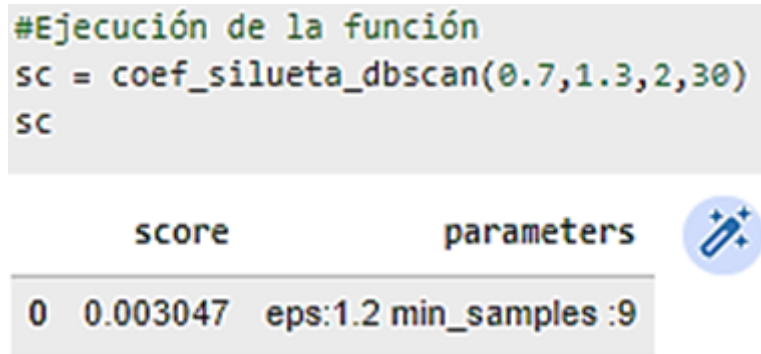


Figura 81: Ejecución de la función de la Figura 80.

Como se observa en la Figura 81, los valores de los parámetros que obtienen un mayor coeficiente de silueta son 1,2 para épsilon y 9 para muestras mínimas, obteniendo un coeficiente de silueta de 0,0030.

6.2.3 Agrupamiento Jerárquico

Este algoritmo ya fue definido anteriormente en el “Estado de la Cuestión” del presente proyecto (ver sección 2.4.4.3).

Por lo que el siguiente paso a realizar, es elegir el valor de “ k ” necesario para ejecutar el algoritmo.

6.2.3.1 Elección del Valor de “ k ”

Como ya se nombró anteriormente, para ejecutar el algoritmo de agrupamiento jerárquico se debe definir un valor de “ k ” que designe la cantidad de clusters en la que se van a agrupar los datos.

Para elegir este valor de “ k ”, primero se hará una aproximación del posible valor con el uso de dendrogramas, para luego, determinar el mejor valor de acuerdo al coeficiente de silueta obtenido más alto a partir de estos posibles valores de “ k ”.

6.2.3.1.1 Dendrogramas

El dendrograma es un diagrama de árbol que muestra los grupos que se forman al crear clusters de observaciones en cada paso y sus niveles de distancia. El nivel de distancia se mide en el eje vertical y las diferentes observaciones se especifican en el eje horizontal (Minitab, s.f.).

Se utiliza el dendrograma para ver cómo se forman los clusters en cada paso y para evaluar los niveles de distancia de los clusters que se forman.

La decisión del número de clústeres que pueden representar mejor a los diferentes grupos se puede elegir observando el dendrograma. La mejor elección del número de

clústeres es el número de líneas verticales en el dendrograma cortadas por una línea horizontal que puede cruzar la distancia máxima verticalmente sin intersectar un clúster. Sin embargo, no es necesario hacer un corte solo en esta línea que cruce la distancia máxima sin intersectar un cluster, sino que, se puede elegir cualquier punto como punto de corte (González, 2020).

Ahora se creará un dendrograma para este modelo, para ello primero se van a importar las librerías “*scipy.cluster.hierarchy*” y “*dendrogram*” (ver Anexo II).

Una vez que ya se dispone de las librerías necesarias, se procederá a crear el dendrograma con las denuncias, en este caso se utilizará la distancia euclídea.

Tanto el algoritmo de dendrograma como el de agrupación jerárquica, solicitan que las denuncias estén en formato array, por lo que se las llevarán a dicho formato.

```
#Conversión del vector de palabras  
features_arr = features.toarray()
```

Figura 82: Conversión del vector de palabras a formato array.

```
#Dendrograma  
plt.figure(figsize=(17,10))  
plt.title('Dendrograma')  
plt.xlabel('Denuncias')  
plt.ylabel('Distancia Euclídea')  
dendrogram = sch.dendrogram(sch.linkage(features_arr, method = 'ward'))  
plt.show()
```

Figura 83: Código para la creación del dendrograma a partir de nuestras denuncias.

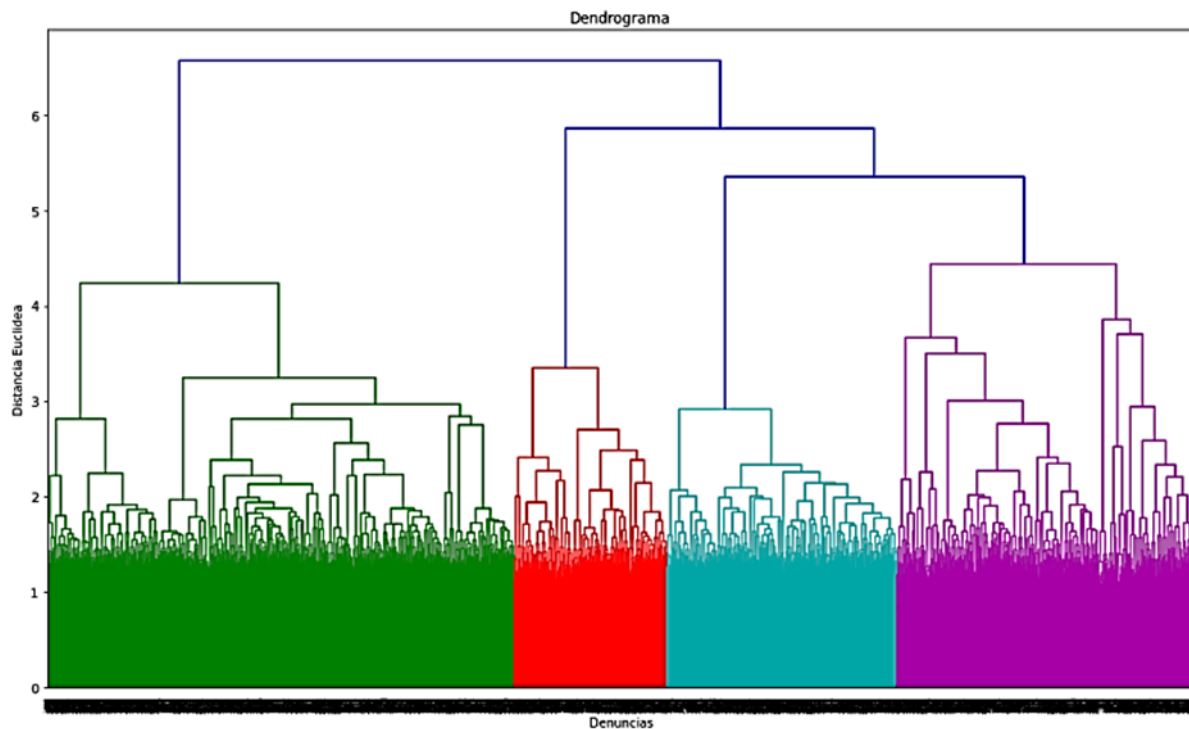


Figura 84: Dendrograma obtenido a partir del código de la Figura 83.

Ahora que se posee el dendrograma, se agregarán las líneas horizontales para definir el número de clusters a utilizar.

En principio el mejor valor para trazar la línea es donde la distancia euclídea es un poco superior a 6, porque es la distancia máxima sin interceptar un cluster (en este caso representado por las líneas verticales) a partir de otro cluster.

En este caso se trazará más de una línea horizontal para poder contemplar una mayor cantidad de valores de “ k ”.

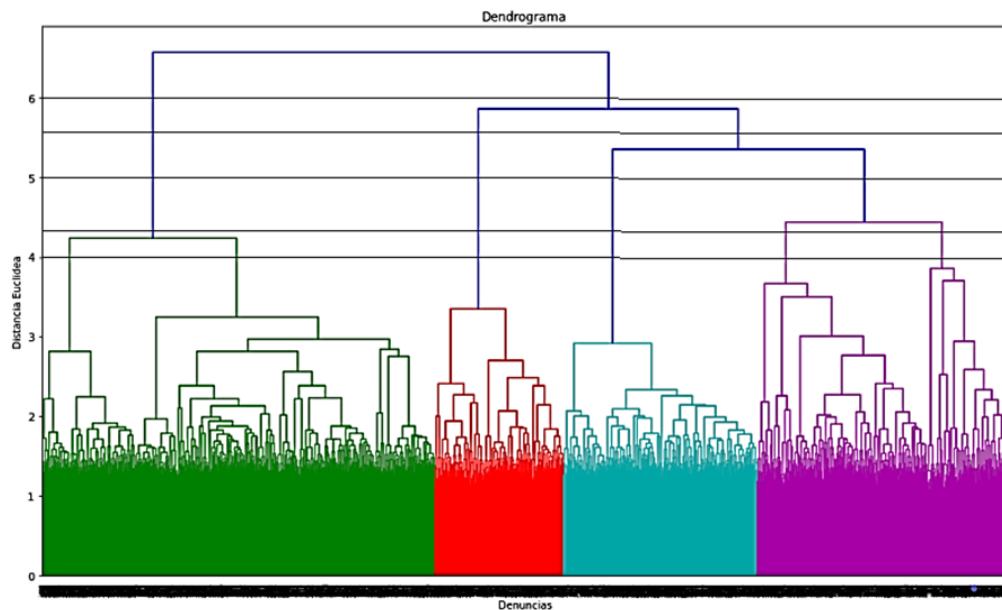


Figura 85: Trazado de líneas horizontales para definir la cantidad de clusters.

Como se observa en la Figura 85, la primera línea situada en el valor de distancia euclídea igual a 6 nos indica que el valor de “ k ” debería ser 2 (en teoría el mejor valor por lo antes explicado), la siguiente nos marca que debería ser 3, la que le sigue nos marca que debería ser 4, la siguiente que deberíamos tomar el valor 5 y la última nos indica que el valor de “ k ” debería ser 6.

No se trazaron más líneas debido a que luego la distancia entre clusters es considerablemente menor, por lo que, no serían buenos valores.

6.2.3.1.2 Coeficiente de Silueta

Como se estableció previamente, el modelo elegido como el mejor será el que obtenga el valor más alto de su coeficiente de silueta, por lo que se calcularán los valores de silueta variando los valores de “ k ” en el algoritmo de agrupamiento jerárquico.

Primero se importará la librería “*AgglomerativeClustering*” (ver Anexo II).

Luego, se realizó la creación de la función que permitirá encontrar el mejor coeficiente.

```
#Función para obtener el modelo con coeficiente de silueta más alto
def coef_silueta_hierarchica(lim_inf,lim_sup):

    #Creación de un dataframe vacío para almacenar los resultados
    silhouette_scores_data=pd.DataFrame()

    k = range(lim_inf,lim_sup)
    for num_cluster in k:

        hc = AgglomerativeClustering(n_clusters = num_cluster, affinity = 'euclidean', linkage = 'ward')
        y_hc = hc.fit_predict(features_arr)
        cluster_labels = hc.labels_
        sc = silhouette_score(features_arr, cluster_labels, random_state=1)
        trial_k = "K=" + str(num_cluster)

        #Cálculo del coeficiente de silueta de cada prueba
        silhouette_scores_data = silhouette_scores_data.append(pd.DataFrame(data=[[sc,num_cluster]],\
                                                                            columns=["Coeficiente de Silueta", "Valor de K"]))

    #Obtención del valor más alto del coeficiente de silueta
    #con sus respectivos parámetros
    value = silhouette_scores_data.sort_values(by='Coeficiente de Silueta', ascending=False).head(1)

    return value
```

Figura 86: Función para encontrar el mejor modelo utilizando agrupamiento jerárquico.

La función de la Figura 86, solicita como parámetros un límite inferior y superior para definir un rango para variar los valores de “k”.

Debido a lo observado en la Figura 85 obtenida anteriormente cuando se graficó el dendrograma del modelo, el rango elegido para los valores de “k” variará entre 2 y 6.

Una vez definido el rango de los valores de “k”, se crea un dataframe vacío para almacenar tanto el valor de coeficiente de silueta, así como el número de clusters utilizado para obtener dicho valor.

Entonces, se procede a iterar variando el parámetro dentro del rango elegido y guardando los valores antes mencionados dentro del dataframe. Una vez que termina de iterar, se muestra el valor de silueta más alto así como el número de clusters indicado por el valor de “k” perteneciente al modelo que obtiene un mejor coeficiente de silueta.

```
#Ejecución de la función
sc = coef_silueta_hierarchica(2,6)
sc
```

	Coeficiente de Silueta	Valor de K
0	0.008274	2

Figura 87: Ejecución de la función de la Figura 86.

Como se ve en la Figura 87, el valor de “ k ” resultó ser de 2, como se preveía según el dendrograma, y el coeficiente de silueta es igual a 0,0082.

6.3 Evaluación de los modelos

Como se definió anteriormente, la métrica de evaluación para determinar el mejor modelo será el coeficiente de silueta del modelo, por lo que, el modelo con un mayor valor en esta métrica será considerado el modelo a analizar.

En la Tabla 7, se observan los mejores modelos obtenidos de cada algoritmo utilizado, junto con sus respectivos coeficientes de silueta y valores de parámetros. Cabe destacar que los coeficientes de silueta fueron recortados a cuatro dígitos después de la coma decimal sin redondear.

Algoritmo	Parámetros	Coeficiente de silueta
K-means	$k = 5$ random_state (semilla) = 26	0,0103
DBSCAN	$\text{eps} = 1,2$ min_samples = 9	0,0030
Agrupamiento Jerárquico	$k = 2$	0,0082

Tabla 7: Comparación de rendimiento de los distintos algoritmos.

Como se observa en la Tabla 7, el mejor modelo es el realizado con el algoritmo k-means, ya que tiene le mayor coeficiente de silueta.

Si se analiza el valor del coeficiente es muy cercano a cero, lo que nos habla de un modelo que contiene clusters superpuestos o muy cercanos entre sí, por lo que no podría considerarse como un modelo ideal sino, como uno solo aceptable.

6.4 Modelo final

Como ya se estableció anteriormente, el modelo elegido como mejor es el llevado a cabo con el algoritmo k-means con el valor de “ k ” igual a 5, así que se ejecutará y analizará dicho modelo.

Primeramente, se va a realizar la copia del dataframe como en todas las etapas del proyecto (ver Anexo II).

Luego, se procede a ejecutar el modelo k-means con el valor de “*k*” y el valor de “*random_state*” correspondientes.

```
#Modelo K-means
k = 5
model = KMeans(n_clusters=k, init='k-means++', max_iter=1000, n_init=1, random_state=26)
model.fit(features)
```

Figura 88: Ejecución del modelo K-means.

Lo siguiente, será asignar el número de cluster correspondiente a cada denuncia dentro del dataframe para poder realizar un análisis más profundo de cada cluster.

```
#Agrupamiento de las denuncias en clusters
d_denuncias_cluster["Cluster"] = model.labels_
d_denuncias_cluster
```

	LugarHecho	Delito	Relato	MesHecho	DiaHecho	Cluster
0	salon eventos terrazas b° almodena	Hurto	primeramente aclarar radicar denunciar repres...	Diciembre	Segunda_Quincena	4
1	b° 20 junio calle david zambrano 94	Hurto	primeramente manifesto acusar madre ex parej...	Diciembre	Segunda_Quincena	4
2	hospital materno infantil milagro	Hurto	dia fecho hora epigrafe circuntancia enc ntra...	Enero	Primera_Quincena	4
3	domicilio sra yohana cerca comercio esmeralda ...	Hurto	primeramente querer dejar aclarar fecho hora ...	Enero	Primera_Quincena	1
4	avda san martin 2 100 aprox cerca mercado arte...	Hurto	manifesto fecho hs aprox maridar denunciante...	Diciembre	Segunda_Quincena	3
...	...	...	...	...	...	...
3218	belgrano n° 159 ciudad	Robo	primeramente querer dejar aclarar radicar den...	Septiembre	Primera_Quincena	4
3219	b° joaquin rueda calle pinos n° 246	Hurto	primeramente querer dejar aclarar relación se...	Octubre	Primera_Quincena	4
3220	paraje fuerte alto banda sur cachi	Robo	dejar aclarar parejo sr lapso quince año comú...	Noviembre	Primera_Quincena	4
3221	asentamiento b° 13 abril s n°	Hurto	encontrar hacer cuatro año parejo sr residir ...	Noviembre	Segunda_Quincena	4
3222	v° velata joaquin corbalan s n	Hurto	radicar ex parejo sr dlio v ° velata psje pal...	Noviembre	Segunda_Quincena	4

3223 rows x 6 columns

Figura 89: Asignación de clusters para cada denuncia.

Como se observa en la Figura 89, se posee en total 3223 filas, es decir, 3223 denuncias, este dato será útil más adelante para calcular datos porcentuales.

Ahora se obtendrán los términos más representativos de cada cluster.

```
#Obtener palabras características de cada cluster
clusters = d_denuncias_cluster.groupby('Cluster')

order_centroids = model.cluster_centers_.argsort()[:, :-1]
terms = vectorizador.get_feature_names()
```

Figura 90: Obtención de los términos más representativos de cada cluster.

Como ya se obtuvieron los términos más representativos de cada cluster, se va a proceder a realizar un análisis más profundo de cada cluster.

Cabe aclarar que los términos obtenidos se encuentran lematizados, por lo que estarán redactados de acuerdo a lo que el algoritmo de lematización detecte como su palabra raíz (ver sección 5.1.4.1.4).

Otra aclaración pertinente, es que lamentablemente los datos contenidos en el dataset no cuentan con una descripción estandarizada del lugar donde ocurren los delitos, por lo que, no puede hacerse una descripción cuantitativa adecuada como sí se realizará con los demás atributos.

6.4.1 Cluster 0

En esta sección, se hará un análisis del cluster 0 obtenido a partir del modelo de agrupamiento realizado.

Se creará un dataframe únicamente con las denuncias pertenecientes a este cluster.

```
#Creación de dataframe con denuncias pertenecientes al cluster 0  
d_denuncias_cluster0 = d_denuncias_cluster[d_denuncias_cluster["Cluster"]== 0]
```

Figura 91: Creación de dataframe con denuncias del cluster 0.

Ahora que ya se posee el dataframe, a continuación se realizará una caracterización a partir de los términos que componen al cluster, calcular las cantidades y porcentajes de acuerdo a los distintos valores de los atributos del dataframe y poder brindar información certera a partir de estos cálculos y análisis.

6.4.1.1 Caracterización

Se mostrarán los 20 términos más significativos de este cluster para poder caracterizarlo.

```
#Mostrar los 20 términos más representativos
#del cluster
print("Cluster %d:" % 0)
for j in order_centroids[0, :20]:
    print (' %s' % terms[j])

Cluster 0:
tarjeta
billetero
banco
nombrar
dni
interior
colectivo
macro
peso
carnet
personar
hurto
documentar
contener
cartero
identidad
banco_macro
sustraer
dicente
saeta
```

Figura 92: Obtención de los 20 términos más representativos del cluster 0.

Observando los términos representativos de este primer cluster, se puede ver que hacen referencia a robos y hurtos de objetos tales como billeteras, bolsos, documentos de identidad, tarjetas y dinero.

6.4.1.2 Análisis General

Primero, se contarán cuántas denuncias posee el dataframe y qué porcentaje representa con respecto a la totalidad de denuncias.

```
#Cantidad y porcentaje general
cant_cluster = len(d_denuncias_cluster0)
print("Cantidad:",cant_cluster)
cant_total = len(d_denuncias_cluster)
porcentaje = round(cant_cluster*100/cant_total,2)
print("Porcentaje:",porcentaje,"%")

Cantidad: 448
Porcentaje: 13.9 %
```

Figura 93: Cantidad y porcentaje de denuncias del cluster 0.

Como se observa en la Figura 93, este cluster contiene 448 denuncias que representan el 13,9% de las denuncias totales.

6.4.1.3 Análisis por Tipo de Delito

Ahora, se calculará la cantidad de denuncias de este cluster según el tipo de delito, y el porcentaje que representa dicha cantidad con respecto a las denuncias pertenecientes a este cluster.

```
#Cantidad y porcentaje de denuncias por robo
cant = len(d_denuncias_cluster0[d_denuncias_cluster0["Delito"]=="Robo"])
print("Cantidad:",cant)
porcentaje = round(cant*100/cant_cluster,2)
print("Porcentaje:",porcentaje,"%")

Cantidad: 58
Porcentaje: 12.95 %

#Cantidad y porcentaje de denuncias por hurto
cant = len(d_denuncias_cluster0[d_denuncias_cluster0["Delito"]=="Hurto"])
print("Cantidad:",cant)
porcentaje = round(cant*100/cant_cluster,2)
print("Porcentaje:",porcentaje,"%")

Cantidad: 390
Porcentaje: 87.05 %
```

Figura 94: Cantidades y porcentajes de denuncias del cluster 0 según el tipo de delito.

Como observamos en la Figura 94, un 87,05% de las denuncias de este tipo son hurtos, por lo que la mayoría de estos delitos se efectúan sin violencia de por medio.

6.4.1.4 Análisis por Mes

Ahora, se calculará la cantidad de denuncias de este cluster según el mes en el que ocurrieron y el porcentaje que representa dicha cantidad con respecto a las denuncias pertenecientes a este cluster.

```
#Cantidad y porcentaje de denuncias en enero
cant = len(d_denuncias_cluster0[d_denuncias_cluster0["MesHecho"]== "Enero"])
print("Cantidad:",cant)
porcentaje = round(cant*100/cant_cluster,2)
print("Porcentaje:",porcentaje,"%")
```

Cantidad: 40
Porcentaje: 8.93 %

```
#Cantidad y porcentaje de denuncias en febrero
cant = len(d_denuncias_cluster0[d_denuncias_cluster0["MesHecho"]== "Febrero"])
print("Cantidad:",cant)
porcentaje = round(cant*100/cant_cluster,2)
print("Porcentaje:",porcentaje,"%")
```

Cantidad: 55
Porcentaje: 12.28 %

```
#Cantidad y porcentaje de denuncias en marzo
cant = len(d_denuncias_cluster0[d_denuncias_cluster0["MesHecho"]== "Marzo"])
print("Cantidad:",cant)
porcentaje = round(cant*100/cant_cluster,2)
print("Porcentaje:",porcentaje,"%")
```

Cantidad: 57
Porcentaje: 12.72 %

```
#Cantidad y porcentaje de denuncias en abril
cant = len(d_denuncias_cluster0[d_denuncias_cluster0["MesHecho"]== "Abril"])
print("Cantidad:",cant)
porcentaje = round(cant*100/cant_cluster,2)
print("Porcentaje:",porcentaje,"%")
```

Cantidad: 42
Porcentaje: 9.38 %

Figura 95: Cantidades y porcentajes de denuncias del cluster 0 según el mes de ocurrencia del delito (primer cuatrimestre).

```
#Cantidad y porcentaje de denuncias en mayo
cant = len(d_denuncias_cluster0[d_denuncias_cluster0["MesHecho"]=="Mayo"])
print("Cantidad:",cant)
porcentaje = round(cant*100/cant_cluster,2)
print("Porcentaje:",porcentaje,"%")
```

Cantidad: 39
Porcentaje: 8.71 %

```
#Cantidad y porcentaje de denuncias en junio
cant = len(d_denuncias_cluster0[d_denuncias_cluster0["MesHecho"]=="Junio"])
print("Cantidad:",cant)
porcentaje = round(cant*100/cant_cluster,2)
print("Porcentaje:",porcentaje,"%")
```

Cantidad: 43
Porcentaje: 9.6 %

```
#Cantidad y porcentaje de denuncias en julio
cant = len(d_denuncias_cluster0[d_denuncias_cluster0["MesHecho"]=="Julio"])
print("Cantidad:",cant)
porcentaje = round(cant*100/cant_cluster,2)
print("Porcentaje:",porcentaje,"%")
```

Cantidad: 38
Porcentaje: 8.48 %

```
#Cantidad y porcentaje de denuncias en agosto
cant = len(d_denuncias_cluster0[d_denuncias_cluster0["MesHecho"]=="Agosto"])
print("Cantidad:",cant)
porcentaje = round(cant*100/cant_cluster,2)
print("Porcentaje:",porcentaje,"%")
```

Cantidad: 39
Porcentaje: 8.71 %

Figura 96: Cantidades y porcentajes de denuncias del cluster 0 según el mes de ocurrencia del delito (segundo cuatrimestre).

```
#Cantidad y porcentaje de denuncias en septiembre
cant = len(d_denuncias_cluster0[d_denuncias_cluster0["MesHecho"]== "Septiembre"])
print("Cantidad:",cant)
porcentaje = round(cant*100/cant_cluster,2)
print("Porcentaje:",porcentaje,"%")
```

Cantidad: 37
Porcentaje: 8.26 %

```
#Cantidad y porcentaje de denuncias en octubre
cant = len(d_denuncias_cluster0[d_denuncias_cluster0["MesHecho"]== "Octubre"])
print("Cantidad:",cant)
porcentaje = round(cant*100/cant_cluster,2)
print("Porcentaje:",porcentaje,"%")
```

Cantidad: 35
Porcentaje: 7.81 %

```
#Cantidad y porcentaje de denuncias en noviembre
cant = len(d_denuncias_cluster0[d_denuncias_cluster0["MesHecho"]== "Noviembre"])
print("Cantidad:",cant)
porcentaje = round(cant*100/cant_cluster,2)
print("Porcentaje:",porcentaje,"%")
```

Cantidad: 21
Porcentaje: 4.69 %

```
#Cantidad y porcentaje de denuncias en diciembre
cant = len(d_denuncias_cluster0[d_denuncias_cluster0["MesHecho"]== "Diciembre"])
print("Cantidad:",cant)
porcentaje = round(cant*100/cant_cluster,2)
print("Porcentaje:",porcentaje,"%")
```

Cantidad: 2
Porcentaje: 0.45 %

Figura 97: Cantidades y porcentajes de denuncias del cluster 0 según el mes de ocurrencia del delito (tercer cuatrimestre).

Como se observa en las anteriores Figuras 95, 96 y 97, el mayor porcentaje de estos delitos ocurre en los meses de febrero (12,28%) y marzo (12,72%), mientras que en los meses de noviembre (4,69%) y, principalmente, diciembre (0,45%) ocurre la menor cantidad.

6.4.1.5 Análisis por Quincena del Mes

Ahora, se calculará la cantidad de denuncias de este cluster según la quincena del mes en la que ocurre el delito y el porcentaje que representa dicha cantidad con respecto a las denuncias pertenecientes a este cluster.

```
#Cantidad y porcentaje de denuncias en la primera quincena del mes
cant = len(d_denuncias_cluster0[d_denuncias_cluster0["DiaHecho"]== "Primera_Quincena"])
print("Cantidad:",cant)
porcentaje = round(cant*100/cant_cluster,2)
print("Porcentaje:",porcentaje,"%")
```

```
Cantidad: 224
Porcentaje: 50.0 %
```

```
#Cantidad y porcentaje de denuncias en la segunda quincena del mes
cant = len(d_denuncias_cluster0[d_denuncias_cluster0["DiaHecho"]== "Segunda_Quincena"])
print("Cantidad:",cant)
porcentaje = round(cant*100/cant_cluster,2)
print("Porcentaje:",porcentaje,"%")
```

```
Cantidad: 224
Porcentaje: 50.0 %
```

Figura 98: Cantidades y porcentajes de denuncias del cluster 0 según la quincena del mes en la que ocurrieron.

Observando la Figura 98, se ve que no hay una preferencia dentro de la quincena del mes para efectuar este tipo de delitos ya que son porcentajes iguales.

6.4.1.6 Análisis por Lugar del Hecho

Solo se pudo realizar una descripción analizando “manualmente” las denuncias pertenecientes a este cluster debido a las dificultades antes descritas (ver sección 6.4), se pudo observar que la mayoría de estos delitos ocurren en la zona céntrica de la ciudad de Salta (principalmente avenida San Martín) y, en menor medida pero siendo una buena cantidad, en corredores de colectivo.

6.4.1.7 Resumen del Análisis

A continuación, se presenta la Tabla 8 que contiene todos las cantidades y porcentajes calculados de este cluster a modo de resumen del análisis realizado.

Cantidad total (porcentaje sobre el total de denuncias)	Cantidad por tipo de delito (porcentaje sobre denuncias de este cluster)	Cantidad por mes (porcentaje sobre denuncias de este cluster)	Cantidad por quincena del mes (porcentaje sobre denuncias de este cluster)
Total: 448 (13,90%)	Robo: 58 (12,95%) Hurto: 390 (87,05%)	Enero: 40 (8,93 %) Febrero: 55 (12,28 %) Marzo: 57 (12,72%) Abril: 42 (9,38%) Mayo: 39 (8,71%) Junio: 43 (9,60%) Julio: 38 (8,48%) Agosto: 39 (8,71%) Septiembre: 37 (8,26%) Octubre: 35 (7,81%) Noviembre: 21 (4,69%) Diciembre: 2 (0,45%)	Primera: 224 (50,00%) Segunda: 224 (50,00%)

Tabla 8: Resumen del análisis cuantitativo del cluster 0.

6.4.2 Cluster 1

En esta sección, se hará un análisis del cluster 1 obtenido a partir del modelo de agrupamiento realizado.

Se creará un dataframe únicamente con las denuncias pertenecientes a este cluster.

```
#Creación de dataframe con denuncias pertenecientes al cluster 1
d_denuncias_cluster1 = d_denuncias_cluster[d_denuncias_cluster["Cluster"]== 1]
```

Figura 99: Creación de dataframe con denuncias del cluster 1.

Ahora que ya se posee el dataframe, a continuación se va a realizar una caracterización a partir de los términos que componen al cluster, calcular las cantidades y porcentajes de acuerdo a los distintos valores de los atributos del dataframe y poder brindar información certera a partir de estos cálculos y análisis.

6.4.2.1 Caracterización

Se van a mostrar los 20 términos más significativos de este cluster para poder caracterizarlo.

```
#Mostrar los 20 términos más representativos  
#del cluster  
print("Cluster %d:" % 1)  
for j in order_centroids[1, :20]:  
    print (' %s' % terms[j])
```

```
Cluster 1:  
celular  
colectivo  
imei  
personar  
n_  
color  
teléfono  
hurto  
encontrar  
empresa  
sustraer  
personal  
celular_mca  
bolsillo  
desconocido  
personar_desconocido  
samsung  
gmail  
corredor  
mca
```

Figura 100: Obtención de los 20 términos más representativos del cluster 1.

Observando los términos representativos de este segundo cluster, se ve que hacen referencia a robos y hurtos de objetos de telefonía móvil.

6.4.2.2 Análisis General

Primero, se contarán cuántas denuncias posee el dataframe y qué porcentaje representa con respecto a la totalidad de denuncias.

```
#Cantidad y porcentaje general
cant_cluster = len(d_denuncias_cluster1)
print("Cantidad:",cant_cluster)
cant_total = len(d_denuncias_cluster)
porcentaje = round(cant_cluster*100/cant_total,2)
print("Porcentaje:",porcentaje,"%")

Cantidad: 618
Porcentaje: 19.17 %
```

Figura 101: Cantidad y porcentaje de denuncias del cluster 1.

Como se ve en la Figura 101, este cluster contiene 618 denuncias que representan el 19,17% de las denuncias totales.

6.4.2.3 Análisis por Tipo de Delito

Ahora, se calculará la cantidad de denuncias de este cluster según el tipo de delito y el porcentaje que representa dicha cantidad con respecto a las denuncias pertenecientes a este cluster.

```
#Cantidad y porcentaje de denuncias por robo
cant = len(d_denuncias_cluster1[d_denuncias_cluster1["Delito"]=="Robo"])
print("Cantidad:",cant)
porcentaje = round(cant*100/cant_cluster,2)
print("Porcentaje:",porcentaje,"%")
```

```
Cantidad: 53
Porcentaje: 8.58 %
```

```
#Cantidad y porcentaje de denuncias por hurto
cant = len(d_denuncias_cluster1[d_denuncias_cluster1["Delito"]=="Hurto"])
print("Cantidad:",cant)
porcentaje = round(cant*100/cant_cluster,2)
print("Porcentaje:",porcentaje,"%")
```

```
Cantidad: 565
Porcentaje: 91.42 %
```

Figura 102: Cantidades y porcentajes de denuncias del cluster 1 según el tipo de delito.

Como se observa en la Figura 102, un 91,42% de las denuncias de este tipo son hurtos, por lo que la gran mayoría de estos delitos se efectúan sin violencia de por medio.

6.4.2.4 Análisis por mes

Ahora, se calculará la cantidad de denuncias de este cluster según el mes en el que ocurrieron y el porcentaje que representa dicha cantidad con respecto a las denuncias pertenecientes a este cluster.

```
#Cantidad y porcentaje de denuncias en enero
cant = len(d_denuncias_cluster1[d_denuncias_cluster1["MesHecho"]== "Enero"])
print("Cantidad:",cant)
porcentaje = round(cant*100/cant_cluster,2)
print("Porcentaje:",porcentaje,"%")
```

Cantidad: 61
Porcentaje: 9.87 %

```
#Cantidad y porcentaje de denuncias en febrero
cant = len(d_denuncias_cluster1[d_denuncias_cluster1["MesHecho"]== "Febrero"])
print("Cantidad:",cant)
porcentaje = round(cant*100/cant_cluster,2)
print("Porcentaje:",porcentaje,"%")
```

Cantidad: 58
Porcentaje: 9.39 %

```
#Cantidad y porcentaje de denuncias en marzo
cant = len(d_denuncias_cluster1[d_denuncias_cluster1["MesHecho"]== "Marzo"])
print("Cantidad:",cant)
porcentaje = round(cant*100/cant_cluster,2)
print("Porcentaje:",porcentaje,"%")
```

Cantidad: 65
Porcentaje: 10.52 %

```
#Cantidad y porcentaje de denuncias en abril
cant = len(d_denuncias_cluster1[d_denuncias_cluster1["MesHecho"]== "Abril"])
print("Cantidad:",cant)
porcentaje = round(cant*100/cant_cluster,2)
print("Porcentaje:",porcentaje,"%")
```

Cantidad: 47
Porcentaje: 7.61 %

Figura 103: Cantidades y porcentajes de denuncias del cluster 1 según el mes de ocurrencia del delito del (primer cuatrimestre).

```
#Cantidad y porcentaje de denuncias en mayo
cant = len(d_denuncias_cluster1[d_denuncias_cluster1["MesHecho"]== "Mayo"])
print("Cantidad:",cant)
porcentaje = round(cant*100/cant_cluster,2)
print("Porcentaje:",porcentaje,"%")
```

Cantidad: 53
Porcentaje: 8.58 %

```
#Cantidad y porcentaje de denuncias en junio
cant = len(d_denuncias_cluster1[d_denuncias_cluster1["MesHecho"]== "Junio"])
print("Cantidad:",cant)
porcentaje = round(cant*100/cant_cluster,2)
print("Porcentaje:",porcentaje,"%")
```

Cantidad: 45
Porcentaje: 7.28 %

```
#Cantidad y porcentaje de denuncias en julio
cant = len(d_denuncias_cluster1[d_denuncias_cluster1["MesHecho"]== "Julio"])
print("Cantidad:",cant)
porcentaje = round(cant*100/cant_cluster,2)
print("Porcentaje:",porcentaje,"%")
```

Cantidad: 87
Porcentaje: 14.08 %

```
#Cantidad y porcentaje de denuncias en agosto
cant = len(d_denuncias_cluster1[d_denuncias_cluster1["MesHecho"]== "Agosto"])
print("Cantidad:",cant)
porcentaje = round(cant*100/cant_cluster,2)
print("Porcentaje:",porcentaje,"%")
```

Cantidad: 61
Porcentaje: 9.87 %

Figura 104: Cantidades y porcentajes de denuncias del cluster 1 según el mes de ocurrencia del delito (segundo cuatrimestre).

```
#Cantidad y porcentaje de denuncias en septiembre
cant = len(d_denuncias_cluster1[d_denuncias_cluster1["MesHecho"]=="Septiembre"])
print("Cantidad:",cant)
porcentaje = round(cant*100/cant_cluster,2)
print("Porcentaje:",porcentaje,"%")
```

Cantidad: 53
Porcentaje: 8.58 %

```
#Cantidad y porcentaje de denuncias en octubre
cant = len(d_denuncias_cluster1[d_denuncias_cluster1["MesHecho"]=="Octubre"])
print("Cantidad:",cant)
porcentaje = round(cant*100/cant_cluster,2)
print("Porcentaje:",porcentaje,"%")
```

Cantidad: 59
Porcentaje: 9.55 %

```
#Cantidad y porcentaje de denuncias en noviembre
cant = len(d_denuncias_cluster1[d_denuncias_cluster1["MesHecho"]=="Noviembre"])
print("Cantidad:",cant)
porcentaje = round(cant*100/cant_cluster,2)
print("Porcentaje:",porcentaje,"%")
```

Cantidad: 27
Porcentaje: 4.37 %

```
#Cantidad y porcentaje de denuncias en diciembre
cant = len(d_denuncias_cluster1[d_denuncias_cluster1["MesHecho"]=="Diciembre"])
print("Cantidad:",cant)
porcentaje = round(cant*100/cant_cluster,2)
print("Porcentaje:",porcentaje,"%")
```

Cantidad: 2
Porcentaje: 0.32 %

Figura 105: Cantidades y porcentajes de denuncias del cluster 1 según el mes de ocurrencia del delito (tercer cuatrimestre).

Como se observa en las anteriores Figuras 103, 104 y 105, el mayor porcentaje de estos delitos ocurre en los meses de julio (14,08%) y marzo (10,52%), mientras que en los meses de noviembre (4,37%) y, principalmente, diciembre (0,32%) ocurre la menor cantidad.

6.4.2.5 Análisis por Quincena del Mes

Ahora, se calculará la cantidad de denuncias de este cluster según la quincena del mes en la que ocurre el delito y el porcentaje que representa dicha cantidad con respecto a las denuncias pertenecientes a este cluster.

```
#Cantidad y porcentaje de denuncias en la primera quincena del mes
cant = len(d_denuncias_cluster1[d_denuncias_cluster1["DiaHecho"]== "Primera_Quincena"])
print("Cantidad:",cant)
porcentaje = round(cant*100/cant_cluster,2)
print("Porcentaje:",porcentaje,"%")
```

```
Cantidad: 298
Porcentaje: 48.22 %
```

```
#Cantidad y porcentaje de denuncias en la segunda quincena del mes
cant = len(d_denuncias_cluster1[d_denuncias_cluster1["DiaHecho"]== "Segunda_Quincena"])
print("Cantidad:",cant)
porcentaje = round(cant*100/cant_cluster,2)
print("Porcentaje:",porcentaje,"%")
```

```
Cantidad: 320
Porcentaje: 51.78 %
```

Figura 106: Cantidades y porcentajes de denuncias del cluster 1 según la quincena del mes en la que ocurrieron.

Observando la Figura 106, se ve que no hay una preferencia dentro de la quincena del mes para efectuar este tipo de delitos ya que son porcentajes muy parejos.

6.4.2.6 Análisis por Lugar del Hecho

Se pudo realizar una descripción, analizando “manualmente” las denuncias pertenecientes a este cluster debido a las dificultades antes descritas (ver sección 6.4), se pudo observar que la mayoría de estos delitos ocurren en zona céntrica (principalmente avenida San Martín) y también en boliches bailables, donde los más nombrados son “La Roka”, “Castillo Mao Mao” y “Salón VIP”.

6.4.2.7 Resumen del Análisis

A continuación, se presenta la Tabla 9 que contiene todos las cantidades y porcentajes calculados de este cluster a modo de resumen del análisis realizado.

Cantidad total (porcentaje sobre el total de denuncias)	Cantidad por tipo de delito (porcentaje sobre denuncias de este cluster)	Cantidad por mes (porcentaje sobre denuncias de este cluster)	Cantidad por quincena del mes (porcentaje sobre denuncias de este cluster)
Total: 618 (19,17%)	Robo: 53 (8,58%) Hurto: 565 (91,42%)	Enero: 61 (9,87 %) Febrero: 58 (9,39 %) Marzo: 65 (10,52%) Abril: 47 (7,61%) Mayo: 53 (8,58%) Junio: 45 (7,28%) Julio: 87 (14,08%) Agosto: 61 (9,87%) Septiembre: 53 (8,58%) Octubre: 59 (9,55%) Noviembre: 27 (4,37%) Diciembre: 2 (0,32%)	Primera: 298 (48,22%) Segunda: 320 (51,78%)

Tabla 9: Resumen del análisis cuantitativo del cluster 1.

6.4.3 Cluster 2

En esta sección, se hará un análisis del cluster 2 obtenido a partir del modelo de agrupamiento realizado.

Se creará un dataframe únicamente con las denuncias pertenecientes a este cluster.

```
#Creación de dataframe con denuncias pertenecientes al cluster 2
d_denuncias_cluster2 = d_denuncias_cluster[d_denuncias_cluster["Cluster"]== 2]
```

Figura 107: Creación de dataframe con denuncias del cluster 2.

Ahora que ya se posee el dataframe, a continuación se va a realizar una caracterización a partir de los términos que componen al cluster, calcular las cantidades y porcentajes de acuerdo a los distintos valores de los atributos del dataframe y poder brindar información certera a partir de estos cálculos y análisis.

6.4.3.1 Caracterización

Se van a mostrar los 20 términos más significativos de este cluster para poder caracterizarlo.

```
#Mostrar los 20 términos más representativos  
#del cluster  
print("Cluster %d:" % 2)  
for j in order_centroids[2, :20]:  
    print (' %s' % terms[j])
```

```
Cluster 2:  
masculino  
color  
celular  
hacer  
fugar  
recordar  
dicente  
encontrar  
robo  
vestir  
causante  
n_  
mismo  
motocicleta  
rostro  
contextura  
arrebatar  
lugar  
ser  
do
```

Figura 108: Obtención de los 20 términos más representativos del cluster 2.

Observando los términos representativos del segundo cluster, se ve que hacen referencia a robos y hurtos de objetos efectuados por hombres. También se aprecia el término motocicleta como principal medio para este tipo de robos.

6.4.3.2 Análisis General

Primero, se contarán cuántas denuncias posee el dataframe y qué porcentaje representan con respecto a la totalidad de denuncias.

```
#Cantidad y porcentaje general
cant_cluster = len(d_denuncias_cluster2)
print("Cantidad total:",cant_cluster)
cant_total = len(d_denuncias_cluster)
porcentaje = round(cant_cluster*100/cant_total,2)
print("Porcentaje total:",porcentaje,"%")

Cantidad total: 797
Porcentaje total: 24.73 %
```

Figura 109: Cantidad y porcentaje de denuncias del cluster 2.

Como se observa en la Figura 109, este cluster contiene 797 denuncias que representan el 24,73% de las denuncias totales.

6.4.3.3 Análisis por Tipo de Delito

Ahora, se calculará la cantidad de denuncias de este cluster según el tipo de delito y el porcentaje que representa dicha cantidad con respecto a las denuncias pertenecientes a este cluster.

```
#Cantidad y porcentaje de denuncias por robo
cant = len(d_denuncias_cluster2[d_denuncias_cluster2["Delito"]=="Robo"])
print("Cantidad:",cant)
porcentaje = round(cant*100/cant_cluster,2)
print("Porcentaje:",porcentaje,"%")

Cantidad: 662
Porcentaje: 83.06 %

#Cantidad y porcentaje de denuncias por hurto
cant = len(d_denuncias_cluster2[d_denuncias_cluster2["Delito"]=="Hurto"])
print("Cantidad:",cant)
porcentaje = round(cant*100/cant_cluster,2)
print("Porcentaje:",porcentaje,"%")

Cantidad: 135
Porcentaje: 16.94 %
```

Figura 110: Cantidades y porcentajes de denuncias del cluster 2 según el tipo de delito.

Como se observa en la Figura 110, la gran mayoría de las denuncias son delitos por robo, lo que implica que las denuncias de este cluster se caracterizan por ser violentas.

6.4.3.4 Análisis por Mes

Ahora, se calculará la cantidad de denuncias de este cluster según el mes en el que ocurrieron y el porcentaje que representa dicha cantidad con respecto a las denuncias pertenecientes a este cluster.

```
#Cantidad y porcentaje de denuncias en enero
cant = len(d_denuncias_cluster2[d_denuncias_cluster2["MesHecho"]== "Enero"])
print("Cantidad:",cant)
porcentaje = round(cant*100/cant_cluster,2)
print("Porcentaje:",porcentaje,"%")
```

```
Cantidad: 93
Porcentaje: 11.67 %
```

```
#Cantidad y porcentaje de denuncias en febrero
cant = len(d_denuncias_cluster2[d_denuncias_cluster2["MesHecho"]== "Febrero"])
print("Cantidad:",cant)
porcentaje = round(cant*100/cant_cluster,2)
print("Porcentaje:",porcentaje,"%")
```

```
Cantidad: 87
Porcentaje: 10.92 %
```

```
#Cantidad y porcentaje de denuncias en marzo
cant = len(d_denuncias_cluster2[d_denuncias_cluster2["MesHecho"]== "Marzo"])
print("Cantidad:",cant)
porcentaje = round(cant*100/cant_cluster,2)
print("Porcentaje:",porcentaje,"%")
```

```
Cantidad: 84
Porcentaje: 10.54 %
```

```
#Cantidad y porcentaje de denuncias en abril
cant = len(d_denuncias_cluster2[d_denuncias_cluster2["MesHecho"]== "Abril"])
print("Cantidad:",cant)
porcentaje = round(cant*100/cant_cluster,2)
print("Porcentaje:",porcentaje,"%")
```

```
Cantidad: 63
Porcentaje: 7.9 %
```

Figura 111: Cantidades y porcentajes de denuncias del cluster 2 según el mes de ocurrencia del delito (primer cuatrimestre).

```
#Cantidad y porcentaje de denuncias en mayo
cant = len(d_denuncias_cluster2[d_denuncias_cluster2["MesHecho"]=="Mayo"])
print("Cantidad:",cant)
porcentaje = round(cant*100/cant_cluster,2)
print("Porcentaje:",porcentaje,"%")
```

Cantidad: 79
Porcentaje: 9.91 %

```
#Cantidad y porcentaje de denuncias en junio
cant = len(d_denuncias_cluster2[d_denuncias_cluster2["MesHecho"]=="Junio"])
print("Cantidad:",cant)
porcentaje = round(cant*100/cant_cluster,2)
print("Porcentaje:",porcentaje,"%")
```

Cantidad: 73
Porcentaje: 9.16 %

```
#Cantidad y porcentaje de denuncias en julio
cant = len(d_denuncias_cluster2[d_denuncias_cluster2["MesHecho"]=="Julio"])
print("Cantidad:",cant)
porcentaje = round(cant*100/cant_cluster,2)
print("Porcentaje:",porcentaje,"%")
```

Cantidad: 60
Porcentaje: 7.53 %

```
#Cantidad y porcentaje de denuncias en agosto
cant = len(d_denuncias_cluster2[d_denuncias_cluster2["MesHecho"]=="Agosto"])
print("Cantidad:",cant)
porcentaje = round(cant*100/cant_cluster,2)
print("Porcentaje:",porcentaje,"%")
```

Cantidad: 63
Porcentaje: 7.9 %

Figura 112: Cantidades y porcentajes de denuncias del cluster 2 según el mes de ocurrencia del delito (segundo cuatrimestre).

```
#Cantidad y porcentaje de denuncias en septiembre
cant = len(d_denuncias_cluster2[d_denuncias_cluster2["MesHecho"]== "Septiembre"])
print("Cantidad:",cant)
porcentaje = round(cant*100/cant_cluster,2)
print("Porcentaje:",porcentaje,"%")
```

Cantidad: 70
Porcentaje: 8.78 %

```
#Cantidad y porcentaje de denuncias en octubre
cant = len(d_denuncias_cluster2[d_denuncias_cluster2["MesHecho"]== "Octubre"])
print("Cantidad:",cant)
porcentaje = round(cant*100/cant_cluster,2)
print("Porcentaje:",porcentaje,"%")
```

Cantidad: 70
Porcentaje: 8.78 %

```
#Cantidad y porcentaje de denuncias en noviembre
cant = len(d_denuncias_cluster2[d_denuncias_cluster2["MesHecho"]== "Noviembre"])
print("Cantidad:",cant)
porcentaje = round(cant*100/cant_cluster,2)
print("Porcentaje:",porcentaje,"%")
```

Cantidad: 51
Porcentaje: 6.4 %

```
#Cantidad y porcentaje de denuncias en diciembre
cant = len(d_denuncias_cluster2[d_denuncias_cluster2["MesHecho"]== "Diciembre"])
print("Cantidad:",cant)
porcentaje = round(cant*100/cant_cluster,2)
print("Porcentaje:",porcentaje,"%")
```

Cantidad: 4
Porcentaje: 0.5 %

Figura 113: Cantidades y porcentajes de denuncias del cluster 2 según el mes de ocurrencia del delito (tercer cuatrimestre).

Como se observa en las anteriores Figuras 111, 112 y 113, los porcentajes varían entre 7% y 9%, excepto los meses de noviembre (6,40%) y, principalmente, diciembre (0,50%) que son en los que ocurre la menor cantidad de este tipo de delitos, y destacándose levemente los meses de enero (11,67%), febrero(10,92%) y marzo (10,54%) como los meses con mayor cantidad.

6.4.3.5 Análisis por Quincena del Mes

Ahora, se calculará la cantidad de denuncias de este cluster según la quincena del mes en la que ocurre el delito y el porcentaje que representa dicha cantidad con respecto a las denuncias pertenecientes a este cluster.

```
#Cantidad y porcentaje de denuncias en la primera quincena del mes
cant = len(d_denuncias_cluster2[d_denuncias_cluster2["DiaHecho"]=="Primera_Quincena"])
print("Cantidad:",cant)
porcentaje = round(cant*100/cant_cluster,2)
print("Porcentaje:",porcentaje,"%")
```

```
Cantidad: 408
Porcentaje: 51.19 %
```

```
#Cantidad y porcentaje de denuncias en la segunda quincena del mes
cant = len(d_denuncias_cluster2[d_denuncias_cluster2["DiaHecho"]=="Segunda_Quincena"])
print("Cantidad:",cant)
porcentaje = round(cant*100/cant_cluster,2)
print("Porcentaje:",porcentaje,"%")
```

```
Cantidad: 389
Porcentaje: 48.81 %
```

Figura 114: Cantidades y porcentajes de denuncias del cluster 2 según la quincena del mes en la que ocurrieron.

Observando la Figura 114, se ve que no hay una preferencia marcada dentro de la quincena del mes para efectuar este tipo de delitos ya que son porcentajes muy parejos.

6.4.3.6 Análisis por Lugar del Hecho

Se pudo realizar una descripción, analizando “manualmente” las denuncias pertenecientes a este cluster debido a las dificultades antes descritas (ver sección 6.4), no se pudo observar que estos delitos se efectúan en sitios que presenten un patrón, cubriendo alguna zona específica o alguna otra particularidad.

6.4.3.7 Resumen del Análisis

A continuación, se presenta la Tabla 10 que contiene todos las cantidades y porcentajes calculados de este cluster a modo de resumen del análisis realizado.

Cantidad total (porcentaje sobre el total de denuncias)	Cantidad por tipo de delito (porcentaje sobre denuncias de este cluster)	Cantidad por mes (porcentaje sobre denuncias de este cluster)	Cantidad por quincena del mes (porcentaje sobre denuncias de este cluster)
Total: 797 (24,73%)	Robo: 662 (83,06%) Hurto: 135 (16,94%)	Enero: 93 (11,17 %) Febrero: 87 (10,92 %) Marzo: 84 (10,54%) Abril: 63 (7,90%) Mayo: 79 (9,91%) Junio: 73 (9,16%) Julio: 60 (7,53%) Agosto: 63 (7,90%) Septiembre: 70 (8,78%) Octubre: 70 (8,78%) Noviembre: 51 (6,40%) Diciembre: 4 (0,50%)	Primera: 408 (51,19%) Segunda: 389 (48,81%)

Tabla 10: Resumen del análisis cuantitativo del cluster 2.

6.4.4 Cluster 3

En esta sección, se hará un análisis del cluster 3 obtenido a partir del modelo de agrupamiento realizado.

Se creará un dataframe únicamente con las denuncias pertenecientes a este cluster.

```
#Creación de dataframe con denuncias pertenecientes al cluster 3
d_denuncias_cluster3 = d_denuncias_cluster[d_denuncias_cluster["cluster"]== 3]
```

Figura 115: Creación de dataframe con denuncias del cluster 3.

Ahora que ya se posee el dataframe, a continuación se va a realizar una caracterización a partir de los términos que componen al cluster, calcular las cantidades y

porcentajes de acuerdo a los distintos valores de los atributos del dataframe y poder brindar información certera a partir de estos cálculos y análisis.

6.4.4.1 Caracterización

Se van a mostrar los 20 términos más significativos de este cluster para poder caracterizarlo.

```
print("Cluster %d:" % 3)
for j in order_centroids[3, :20]:
    print (' %s' % terms[j])
```

Cluster 3:
rodar
estacionar
dejar
dominio
dejar_estacionar
color
novedad
mca
hacer
sustraer
personar
personar_desconocido
desconocido
novedad_personar
vehículo
novedad_personar_desconocido
camioneta
hora
vehicular
fecho

Figura 116: Obtención de los 20 términos más representativos del cluster 3.

Observando los términos representativos de este cuarto cluster, se ve que hacen referencia a robos y hurtos relacionados con vehículos pertenecientes a las víctimas.

6.4.4.2 Análisis General

Primero, se contarán cuántas denuncias posee el dataframe y qué porcentaje representa con respecto a la totalidad de denuncias.

```
#Cantidad y porcentaje general
cant_cluster = len(d_denuncias_cluster3)
print("Cantidad:",cant_cluster)
cant_total = len(d_denuncias_cluster)
porcentaje = round(cant_cluster*100/cant_total,2)
print("Porcentaje:",porcentaje,"%")

Cantidad: 494
Porcentaje: 15.33 %
```

Figura 117: Cantidad y porcentaje de denuncias del cluster 3.

Como se observa en la Figura 117, este cluster contiene 494 denuncias que representan el 15,33% de las denuncias totales.

6.4.4.3 Análisis por Tipo de Delito

Ahora, se calculará la cantidad de denuncias de este cluster según el tipo de delito y el porcentaje que representa dicha cantidad con respecto a las denuncias pertenecientes a este cluster.

```
#Cantidad y porcentaje de denuncias por robo
cant = len(d_denuncias_cluster3[d_denuncias_cluster3["Delito"]=="Robo"])
print("Cantidad:",cant)
porcentaje = round(cant*100/cant_cluster,2)
print("Porcentaje:",porcentaje,"%")
```

Cantidad: 276
Porcentaje: 55.87 %

```
#Cantidad y porcentaje de denuncias por hurto
cant = len(d_denuncias_cluster3[d_denuncias_cluster3["Delito"]=="Hurto"])
print("Cantidad:",cant)
porcentaje = round(cant*100/cant_cluster,2)
print("Porcentaje:",porcentaje,"%")
```

Cantidad: 218
Porcentaje: 44.13 %

Figura 118: Cantidades y porcentajes de denuncias del cluster 3 según el tipo de delito.

Como se observa en la Figura 118, en este tipo de delitos predominan levemente los robos por sobre los hurtos, por lo que es más común que se den de manera violenta.

6.4.4.4 Análisis por Mes

Ahora, se calculará la cantidad de denuncias de este cluster según el mes en el que ocurrieron y el porcentaje que representa dicha cantidad con respecto a las denuncias pertenecientes a este cluster.

```
#Cantidad y porcentaje de denuncias en enero
cant = len(d_denuncias_cluster3[d_denuncias_cluster3["MesHecho"]=="Enero"])
print("Cantidad:",cant)
porcentaje = round(cant*100/cant_cluster,2)
print("Porcentaje:",porcentaje,"%")
```

Cantidad: 53
Porcentaje: 10.73 %

```
#Cantidad y porcentaje de denuncias en febrero
cant = len(d_denuncias_cluster3[d_denuncias_cluster3["MesHecho"]=="Febrero"])
print("Cantidad:",cant)
porcentaje = round(cant*100/cant_cluster,2)
print("Porcentaje:",porcentaje,"%")
```

Cantidad: 62
Porcentaje: 12.55 %

```
#Cantidad y porcentaje de denuncias en marzo
cant = len(d_denuncias_cluster3[d_denuncias_cluster3["MesHecho"]=="Marzo"])
print("Cantidad:",cant)
porcentaje = round(cant*100/cant_cluster,2)
print("Porcentaje:",porcentaje,"%")
```

Cantidad: 57
Porcentaje: 11.54 %

```
#Cantidad y porcentaje de denuncias en abril
cant = len(d_denuncias_cluster3[d_denuncias_cluster3["MesHecho"]=="Abril"])
print("Cantidad:",cant)
porcentaje = round(cant*100/cant_cluster,2)
print("Porcentaje:",porcentaje,"%")
```

Cantidad: 41
Porcentaje: 8.3 %

Figura 119: Cantidades y porcentajes de denuncias del cluster 3 según el mes de ocurrencia del delito (primer cuatrimestre).

```
#Cantidad y porcentaje de denuncias en mayo
cant = len(d_denuncias_cluster3[d_denuncias_cluster3["MesHecho"]== "Mayo"])
print("Cantidad:",cant)
porcentaje = round(cant*100/cant_cluster,2)
print("Porcentaje:",porcentaje,"%")
```

Cantidad: 37
Porcentaje: 7.49 %

```
#Cantidad y porcentaje de denuncias en junio
cant = len(d_denuncias_cluster3[d_denuncias_cluster3["MesHecho"]== "Junio"])
print("Cantidad:",cant)
porcentaje = round(cant*100/cant_cluster,2)
print("Porcentaje:",porcentaje,"%")
```

Cantidad: 35
Porcentaje: 7.09 %

```
#Cantidad y porcentaje de denuncias en julio
cant = len(d_denuncias_cluster3[d_denuncias_cluster3["MesHecho"]== "Julio"])
print("Cantidad:",cant)
porcentaje = round(cant*100/cant_cluster,2)
print("Porcentaje:",porcentaje,"%")
```

Cantidad: 42
Porcentaje: 8.5 %

```
#Cantidad y porcentaje de denuncias en agosto
cant = len(d_denuncias_cluster3[d_denuncias_cluster3["MesHecho"]== "Agosto"])
print("Cantidad:",cant)
porcentaje = round(cant*100/cant_cluster,2)
print("Porcentaje:",porcentaje,"%")
```

Cantidad: 51
Porcentaje: 10.32 %

Figura 120: Cantidades y porcentajes de denuncias del cluster 3 según el mes de ocurrencia del delito (segundo cuatrimestre).

```
#Cantidad y porcentaje de denuncias en septiembre
cant = len(d_denuncias_cluster3[d_denuncias_cluster3["MesHecho"]== "Septiembre"])
print("Cantidad:",cant)
porcentaje = round(cant*100/cant_cluster,2)
print("Porcentaje:",porcentaje,"%")
```

Cantidad: 39
Porcentaje: 7.89 %

```
#Cantidad y porcentaje de denuncias en octubre
cant = len(d_denuncias_cluster3[d_denuncias_cluster3["MesHecho"]== "Octubre"])
print("Cantidad:",cant)
porcentaje = round(cant*100/cant_cluster,2)
print("Porcentaje:",porcentaje,"%")
```

Cantidad: 53
Porcentaje: 10.73 %

```
#Cantidad y porcentaje de denuncias en noviembre
cant = len(d_denuncias_cluster3[d_denuncias_cluster3["MesHecho"]== "Noviembre"])
print("Cantidad:",cant)
porcentaje = round(cant*100/cant_cluster,2)
print("Porcentaje:",porcentaje,"%")
```

Cantidad: 19
Porcentaje: 3.85 %

```
#Cantidad y porcentaje de denuncias en diciembre
cant = len(d_denuncias_cluster3[d_denuncias_cluster3["MesHecho"]== "Diciembre"])
print("Cantidad:",cant)
porcentaje = round(cant*100/cant_cluster,2)
print("Porcentaje:",porcentaje,"%")
```

Cantidad: 5
Porcentaje: 1.01 %

Figura 121: Cantidades y porcentajes de denuncias del cluster 3 según el mes de ocurrencia del delito (tercer cuatrimestre).

Como se observa en las anteriores Figuras 119, 120 y 121, los porcentajes varían entre 7% y 10% excepto los meses de noviembre (3,85%) y, principalmente, diciembre (1,01%) que son en los que ocurre la menor cantidad de este tipo de delitos. Destacándose levemente febrero (12,55%) y marzo (11,54%) como los meses en los que se observa mayor cantidad.

6.4.4.5 Análisis por Quincena del Mes

Ahora, se calculará la cantidad de denuncias de este cluster según la quincena del mes en la que ocurre el delito y el porcentaje que representa dicha cantidad con respecto a las denuncias pertenecientes a este cluster.

```
#Cantidad y porcentaje de denuncias en la primera quincena del mes
cant = len(d_denuncias_cluster3[d_denuncias_cluster3["DiaHecho"]== "Primera_Quincena"])
print("Cantidad:",cant)
porcentaje = round(cant*100/cant_cluster,2)
print("Porcentaje:",porcentaje,"%")
```

```
Cantidad: 232
Porcentaje: 46.96 %
```

```
#Cantidad y porcentaje de denuncias en la segunda quincena del mes
cant = len(d_denuncias_cluster3[d_denuncias_cluster3["DiaHecho"]== "Segunda_Quincena"])
print("Cantidad:",cant)
porcentaje = round(cant*100/cant_cluster,2)
print("Porcentaje:",porcentaje,"%")
```

```
Cantidad: 262
Porcentaje: 53.04 %
```

Figura 122: Cantidades y porcentajes de denuncias del cluster 3 según la quincena del mes en la que ocurrieron.

Observando la Figura 122, se ve que no hay una preferencia marcada dentro de la quincena del mes para efectuar este tipo de delitos ya que son porcentajes muy parejos.

6.4.4.6 Análisis por Lugar del Hecho

Se pudo realizar una descripción, analizando “manualmente” las denuncias pertenecientes a este cluster debido a las dificultades antes descritas (ver sección 6.4). Se pudo observar que estos delitos se efectúan mayormente en avenidas (Bolivia, San Martín, Paraguay, Bicentenario de la batalla de Salta, Tavella, son algunas de las más mencionadas), barrios o lugares donde abundan edificios (B° Juan Pablo II, B° El Huayco, B° Castañares, B° Limache son algunos de los más mencionados) y se menciona particularmente el estacionamiento del centro de comercio “Hiper Libertad” conocido actualmente como “Paseo Salta”.

6.4.4.7 Resumen del Análisis

A continuación, se presenta la Tabla 11 que contiene todos las cantidades y porcentajes calculados de este cluster a modo de resumen del análisis realizado.

Cantidad total (porcentaje sobre el total de denuncias)	Cantidad por tipo de delito (porcentaje sobre denuncias de este cluster)	Cantidad por mes (porcentaje sobre denuncias de este cluster)	Cantidad por quincena del mes (porcentaje sobre denuncias de este cluster)
Total: 494 (15,33%)	Robo: 276 (55,87%) Hurto: 218 (44,13%)	Enero: 53 (10,73 %) Febrero: 62 (12,55 %) Marzo: 57 (11,54%) Abril: 41 (8,30%) Mayo: 37 (7,49%) Junio: 35 (7,09%) Julio: 42 (8,50%) Agosto: 51 (10,32%) Septiembre: 19 (7,89%) Octubre: 53 (10,73%) Noviembre: 19 (3,85%) Diciembre: 5 (1,01%)	Primera: 290 (47,70%) Segunda: 318 (52,30%)

Tabla 11: Resumen del análisis cuantitativo del cluster 3.

6.4.5 Cluster 4

En esta sección, se hará un análisis del cluster 4 obtenido a partir del modelo de agrupamiento realizado.

Se creará un dataframe únicamente con las denuncias pertenecientes a este cluster.

```
#Creación de dataframe con denuncias pertenecientes al cluster 4
d_denuncias_cluster4 = d_denuncias_cluster[d_denuncias_cluster["Cluster"]== 4]
```

Figura 123: Creación de dataframe con denuncias del cluster 4.

Ahora, que ya se obtuvo el dataframe, a continuación se va a realizar una caracterización a partir de los términos que componen al cluster, calcular las cantidades y

porcentajes de acuerdo a los distintos valores de los atributos del dataframe y poder brindar información certera a partir de estos cálculos y análisis.

6.4.5.1 Caracterización

Se van a mostrar los 20 términos más significativos de este cluster para poder caracterizarlo.

```
print("Cluster %d:" % 4)
for j in order_centroids[4, :20]:
    print (' %s' % terms[j])
```

```
Cluster 4:
hacer
acusar
mismo
encontrar
dicente
domiciliar
ingresar
color
lugar
dejar
casar
denunciar
ser
fecho
hijo
decir
aclarar
sr
hora
solicitar
```

Figura 124: Obtención de los 20 términos más representativos del cluster 4.

Observando los términos representativos de este quinto cluster, puede observarse que este grupo está conformado por delitos ocurridos en los domicilios de las víctimas o en la proximidad de los mismos.

6.4.5.2 Análisis General

Primero, se contarán cuántas denuncias posee el dataframe y qué porcentaje representa con respecto a la totalidad de denuncias.

```
#Cantidad y porcentaje general
cant_cluster = len(d_denuncias_cluster4)
print("Cantidad:",cant_cluster)
cant_total = len(d_denuncias_cluster)
porcentaje = round(cant_cluster*100/cant_total,2)
print("Porcentaje:",porcentaje,"%")

Cantidad: 866
Porcentaje: 26.87 %
```

Figura 125: Cantidad y porcentaje de denuncias del cluster 4.

Como se observa en la Figura 125, este cluster contiene 866 denuncias que representan el 26,87% de las denuncias totales, siendo el grupo que más denuncias contiene.

6.4.5.3 Análisis por Tipo de Delito

Ahora, se calculará la cantidad de denuncias de este cluster según el tipo de delito y el porcentaje que representa dicha cantidad con respecto a las denuncias pertenecientes a este cluster.

```
#Cantidad y porcentaje de denuncias por robo
cant = len(d_denuncias_cluster4[d_denuncias_cluster4["Delito"]== "Robo"])
print("Cantidad:",cant)
porcentaje = round(cant*100/cant_cluster,2)
print("Porcentaje:",porcentaje,"%")
```

```
Cantidad: 332
Porcentaje: 38.34 %
```

```
#Cantidad y porcentaje de denuncias por hurto
cant = len(d_denuncias_cluster4[d_denuncias_cluster4["Delito"]== "Hurto"])
print("Cantidad:",cant)
porcentaje = round(cant*100/cant_cluster,2)
print("Porcentaje:",porcentaje,"%")
```

```
Cantidad: 534
Porcentaje: 61.66 %
```

Figura 126: Cantidades y porcentajes de denuncias del cluster 4 según el tipo de delito.

Como se observa en la Figura 126, hay un mayor porcentaje de denuncias por hurto lo que marca que no suelen ser delitos violentos.

6.4.5.4 Análisis por mes

Ahora, se calculará la cantidad de denuncias de este cluster según el mes en el que ocurrieron y el porcentaje que representa dicha cantidad con respecto a las denuncias pertenecientes a este cluster.

```
#Cantidad y porcentaje de denuncias en enero
cant = len(d_denuncias_cluster4[d_denuncias_cluster4["MesHecho"]=="Enero"])
print("Cantidad:",cant)
porcentaje = round(cant*100/cant_cluster,2)
print("Porcentaje:",porcentaje,"%")
```

Cantidad: 77
Porcentaje: 8.89 %

```
#Cantidad y porcentaje de denuncias en febrero
cant = len(d_denuncias_cluster4[d_denuncias_cluster4["MesHecho"]=="Febrero"])
print("Cantidad:",cant)
porcentaje = round(cant*100/cant_cluster,2)
print("Porcentaje:",porcentaje,"%")
```

Cantidad: 87
Porcentaje: 10.05 %

```
#Cantidad y porcentaje de denuncias en marzo
cant = len(d_denuncias_cluster4[d_denuncias_cluster4["MesHecho"]=="Marzo"])
print("Cantidad:",cant)
porcentaje = round(cant*100/cant_cluster,2)
print("Porcentaje:",porcentaje,"%")
```

Cantidad: 71
Porcentaje: 8.2 %

```
#Cantidad y porcentaje de denuncias en abril
cant = len(d_denuncias_cluster4[d_denuncias_cluster4["MesHecho"]=="Abril"])
print("Cantidad:",cant)
porcentaje = round(cant*100/cant_cluster,2)
print("Porcentaje:",porcentaje,"%")
```

Cantidad: 84
Porcentaje: 9.7 %

Figura 127: Cantidades y porcentajes de denuncias del cluster 4 según el mes de ocurrencia del delito (primer cuatrimestre).

```
#Cantidad y porcentaje de denuncias en mayo
cant = len(d_denuncias_cluster4[d_denuncias_cluster4["MesHecho"]== "Mayo"])
print("Cantidad:",cant)
porcentaje = round(cant*100/cant_cluster,2)
print("Porcentaje:",porcentaje,"%")
```

Cantidad: 76
Porcentaje: 8.78 %

```
#Cantidad y porcentaje de denuncias en junio
cant = len(d_denuncias_cluster4[d_denuncias_cluster4["MesHecho"]== "Junio"])
print("Cantidad:",cant)
porcentaje = round(cant*100/cant_cluster,2)
print("Porcentaje:",porcentaje,"%")
```

Cantidad: 74
Porcentaje: 8.55 %

```
#Cantidad y porcentaje de denuncias en julio
cant = len(d_denuncias_cluster4[d_denuncias_cluster4["MesHecho"]== "Julio"])
print("Cantidad:",cant)
porcentaje = round(cant*100/cant_cluster,2)
print("Porcentaje:",porcentaje,"%")
```

Cantidad: 74
Porcentaje: 8.55 %

```
#Cantidad y porcentaje de denuncias en agosto
cant = len(d_denuncias_cluster4[d_denuncias_cluster4["MesHecho"]== "Agosto"])
print("Cantidad:",cant)
porcentaje = round(cant*100/cant_cluster,2)
print("Porcentaje:",porcentaje,"%")
```

Cantidad: 75
Porcentaje: 8.66 %

Figura 128: Cantidades y porcentajes de denuncias del cluster 4 según el mes de ocurrencia del delito (segundo cuatrimestre).

```
#Cantidad y porcentaje de denuncias en septiembre  
cant = len(d_denuncias_cluster4[d_denuncias_cluster4["MesHecho"]== "Septiembre"])  
print("Cantidad:",cant)  
porcentaje = round(cant*100/cant_cluster,2)  
print("Porcentaje:",porcentaje,"%")
```

Cantidad: 85
Porcentaje: 9.82 %

```
#Cantidad y porcentaje de denuncias en octubre  
cant = len(d_denuncias_cluster4[d_denuncias_cluster4["MesHecho"]== "Octubre"])  
print("Cantidad:",cant)  
porcentaje = round(cant*100/cant_cluster,2)  
print("Porcentaje:",porcentaje,"%")
```

Cantidad: 90
Porcentaje: 10.39 %

```
#Cantidad y porcentaje de denuncias en noviembre  
cant = len(d_denuncias_cluster4[d_denuncias_cluster4["MesHecho"]== "Noviembre"])  
print("Cantidad:",cant)  
porcentaje = round(cant*100/cant_cluster,2)  
print("Porcentaje:",porcentaje,"%")
```

Cantidad: 52
Porcentaje: 6.0 %

```
#Cantidad y porcentaje de denuncias en diciembre  
cant = len(d_denuncias_cluster4[d_denuncias_cluster4["MesHecho"]== "Diciembre"])  
print("Cantidad:",cant)  
porcentaje = round(cant*100/cant_cluster,2)  
print("Porcentaje:",porcentaje,"%")
```

Cantidad: 21
Porcentaje: 2.42 %

Figura 129: Cantidades y porcentajes de denuncias del cluster 4 según el mes de ocurrencia del delito (tercer cuatrimestre).

Como se observa en las anteriores Figuras 127, 128 y 129, los porcentajes varían entre 8% y 10% excepto los meses de noviembre (6%) y, principalmente, diciembre (2,42%) que son en los que ocurre la menor cantidad de este tipo de delitos.

6.4.5.5 Análisis por Quincena del Mes

Ahora, se calculará la cantidad de denuncias de este cluster según la quincena del mes en la que ocurre el delito y el porcentaje que representa dicha cantidad con respecto a las denuncias pertenecientes a este cluster.

```
#Cantidad y porcentaje de denuncias en la primera quincena del mes
cant = len(d_denuncias_cluster4[d_denuncias_cluster4["DiaHecho"]=="Primera_Quincena"])
print("Cantidad:",cant)
porcentaje = round(cant*100/cant_cluster,2)
print("Porcentaje:",porcentaje,"%")
```

```
Cantidad: 457
Porcentaje: 52.77 %
```

```
#Cantidad y porcentaje de denuncias en la segunda quincena del mes
cant = len(d_denuncias_cluster4[d_denuncias_cluster4["DiaHecho"]=="Segunda_Quincena"])
print("Cantidad:",cant)
porcentaje = round(cant*100/cant_cluster,2)
print("Porcentaje:",porcentaje,"%")
```

```
Cantidad: 409
Porcentaje: 47.23 %
```

Figura 130: Cantidades y porcentajes de denuncias del cluster 4 según la quincena del mes en la que ocurrieron.

Observando la Figura 130, se ve que no hay una preferencia marcada dentro de la quincena del mes para efectuar este tipo de delitos ya que son porcentajes muy parejos.

6.4.5.6 Análisis por Lugar del Hecho

Se pudo realizar una descripción, analizando “manualmente” las denuncias pertenecientes a este cluster debido a las dificultades antes descritas (ver sección 6.4), sólo se observó que los delitos ocurrieron en un domicilio o en la proximidad del mismo pero ninguna otra particularidad.

6.4.5.7 Resumen del Análisis

A continuación, se presenta la Tabla 12 que contiene todos las cantidades y porcentajes calculados de este cluster a modo de resumen del análisis realizado.

Cantidad total (porcentaje sobre el total de denuncias)	Cantidad por tipo de delito (porcentaje sobre denuncias de este cluster)	Cantidad por mes (porcentaje sobre denuncias de este cluster)	Cantidad por quincena del mes (porcentaje sobre denuncias de este cluster)
Total: 866 (26,87%)	Robo: 332 (38,34%) Hurto: 534 (61,66%)	Enero: 77 (8,89%) Febrero: 87 (10,05%) Marzo: 71 (8,20%) Abril: 84 (9,70%) Mayo: 76 (8,78%) Junio: 74 (8,55%) Julio: 74 (8,55%) Agosto: 75 (8,66%) Septiembre: 85 (9,82%) Octubre: 90 (10,39%) Noviembre: 52 (6,00%) Diciembre: 21 (2,42%)	Primera: 404 (52,88%) Segunda: 360 (47,12%)

Tabla 12: Resumen del análisis cuantitativo del cluster 4.

Capítulo 7 - Evaluación

En esta etapa del proyecto se realizan las evaluaciones del proyecto realizado. El objetivo es determinar si todas las actividades llevadas a cabo contribuyeron a concretar los objetivos planteados al inicio del proyecto. También se llevará a cabo una evaluación sobre los recursos utilizados, restricciones, presunciones, y otros factores que influenciaron la realización del trabajo.

7.1 Evaluación de Resultados

Los dos factores que serán evaluados para determinar el nivel de éxito del proyecto son los criterios de éxito y las restricciones del proyecto.

7.1.1 Criterios de Éxito

Los criterios de éxito planteados al comienzo de este proyecto fueron:

- **Obtener características comunes de los grupos obtenidos a partir de los delitos analizados.**

A partir del proceso de agrupamiento, pudo obtenerse cinco clusters los cuales fueron caracterizados a partir de sus términos más relevantes, por lo que este criterio fue satisfecho, sobre todo la caracterización realizada sobre el primer, segundo y cuarto clusters, la cual considero como muy efectiva.

- **Comparar el rendimiento de los distintos algoritmos de Agrupamiento (Clustering) empleados a partir de la métrica de evaluación elegida (coeficiente de silueta).**

Este criterio también fue concretado, debido a que se pudo comparar los coeficientes de silueta de los modelos que utilizaban los algoritmos k-means, DBSCAN y agrupamiento jerárquico para este caso particular.

- **Obtener un coeficiente de silueta (métrica de evaluación utilizada para evaluar el rendimiento de los modelos planteados) superior a 0 en el modelo elegido como final.**

Este criterio fue concretado, debido a que el coeficiente de silueta del modelo elegido como final fue de 0,0103 que es superior a 0.

- **Obtener información certera a partir del análisis de los resultados obtenidos del proceso de minería de texto.**

Este criterio pudo ser satisfecho, ya que se obtuvieron muchas conclusiones basadas en un análisis de los resultados obtenidos del modelo empleado.

Así también, cabe aclarar que se esperaba obtener una mayor cantidad de información certera, en parte se debió al uso del dataset utilizado en el proyecto, más adelante cuando se analicen las fuentes de datos se hará una crítica más detallada al respecto.

7.1.2 Restricciones del Proyecto

El proyecto pudo llevarse a cabo sin ningún tipo de problema con el equipo de hardware detallado y se concretó antes de la fecha pactada como límite, así que se puede asegurar que se cumplieron las restricciones que se plantearon inicialmente.

7.2 Revisión del proyecto

7.2.1 Metodología

Para llevar a cabo las tareas de minería de texto a lo largo del proyecto, se efectuaron cinco de las seis fases del modelo CRISP-DM, cabe recordar que la etapa de implantación no fue ni será ejecutada como se explicó anteriormente en el alcance del proyecto (ver sección 3.6.1).

Se considera como una correcta decisión el uso de esta metodología, ya que no se presentaron inconvenientes a la hora de desarrollar el proyecto, al contrario, resultó bastante práctico y útil.

Cabe destacar que si bien las tareas realizadas están planteadas de manera lineal, no fueron concretadas de esta manera, sino que se desarrollaron generalmente de manera iterativa. En muchos casos hubo que volver a tareas previas o etapas previas para volver a plantear la forma en que se llevaron a cabo y poder lograr un resultado más satisfactorio. Las etapas a las que más hubo que rehacer o redefinir son principalmente las etapas de “Comprensión del negocio” y “Preparación de datos”.

7.2.2 Fuentes de Datos y Conocimiento

Tanto la fuente de datos como el conocimiento de expertos permitieron llevar a cabo todas las tareas de manera eficaz y cumplir con los objetivos del proyecto.

Una crítica que debe hacerse sobre la fuente de datos, es que las denuncias incluidas en el dataset inicial carecían de una forma estandarizada a la hora de definir el lugar donde ocurrieron los delitos. Otra crítica a realizar, sería que una cantidad mínima de las denuncias contaban con el detalle de la hora en la que se cometieron los delitos mientras que la mayoría no contaba con esta información.

A pesar de estas deficiencias en el dataset inicial, pudo llevarse a cabo todas las actividades del proyecto de manera eficaz, aunque se considera que podrían haberse encontrado una mayor cantidad de conclusiones certeras a partir del análisis de los resultados obtenidos teniendo estos datos nombrados en condiciones favorables.

7.2.3 Software para Minería de Texto

La elección de Google Colaboratory como software utilizado para llevar a cabo la minería de texto, con el lenguaje de programación Python, significó una decisión correcta.

Cabe destacar la versatilidad que brindó tanto Google Colaboratory como Python para llevar a cabo todas las tareas de manera personalizada, sobre todo las librerías disponibles que facilitaron de sobre manera algunas de las actividades y procedimientos realizados. También muy destacado el soporte brindado tanto por las páginas web oficiales de las herramientas y librerías, así como el soporte online brindado por usuarios y expertos que permitieron tener una mayor comprensión tanto de los conceptos utilizados y de las maneras de realizar ciertos procedimientos.

Una única crítica, quizás sería el aspecto visual que brinda la herramienta para presentar tanto el código utilizado como los resultados obtenidos.

7.2.4 Gestión del Proyecto

Tanto la planificación como la ejecución del proyecto, fueron realizados exitosamente debido a que las tareas definidas y los plazos pactados se definieron de manera adecuada, teniendo que ser modificados mínimamente luego de ser planteados.

Capítulo 8 - Conclusiones y Futuras Líneas de Investigación

8.1 Conclusiones

La motivación correspondiente al desarrollo de este proyecto de grado, fue la de obtener información útil que permita actuar a los organismos provinciales, tales como el Ministerio Público Fiscal en conjunto con las fuerzas policiales, contra la inseguridad y la delincuencia mediante acciones que contribuyan a la prevención de delitos. Se considera que se pudieron brindar datos bastante útiles para contrarrestar esta problemática social tan presente en nuestra provincia y en el país en general.

El uso de minería de texto, particularmente implementada con Python, fue un gran desafío que pudo ser llevado a cabo con mucho esfuerzo y dedicación, con la permanente predisposición de comprender cómo funcionan los procesos y los motivos por lo cuales se realizan, así como el aprendizaje de la correcta ejecución de los mismos.

Hubiera sido ideal que el dataset inicial, contenedor de las denuncias utilizadas, tuviera descritos de manera adecuada los lugares donde ocurren los delitos, así como la hora en que se llevaron a cabo, para poder obtener una cantidad aún mayor de información útil. Por supuesto, esta falta de estandarización debido a la forma en que se toman las denuncias se entiende hasta cierto punto, por un lado debido al estado en que se encuentran las víctimas a la hora de realizarlas y, principalmente, porque el principal objetivo de las mismas no es ser fuente de procedimientos de ciencia de datos, sino, la de ayudar al denunciante y penalizar a los delincuentes.

La evaluación realizada, da la pauta de que se cumplieron con los objetivos planteados, y se espera con mucho anhelo que estos datos puedan ser de utilidad en el futuro para ayudar a la provincia de Salta a ser más segura y seguir creciendo en todos los aspectos posibles.

8.2 Futuras Líneas de Investigación

Este proyecto de grado, se plantea como un procedimiento que podría ser replicado en otras provincias u otras regiones, tanto de Argentina como de otros países, incluso considerando otro tipo de denuncias, no solo de robo y hurto como fue en este proyecto.

También, se contempla la vinculación de los organismos encargados de salvaguardar la seguridad y bienestar social de los habitantes, como lo son el Ministerio Público Fiscal en

conjunto con las fuerzas policiales en la provincia de Salta, con el desarrollo de estas investigaciones, para que de manera conjunta se puedan articular acciones de prevención de delitos haciendo uso de la información obtenida.

Se espera que este trabajo pueda ser usado como base o motivación para futuras investigaciones de esta índole.

Bibliografía

- Argentina.gob.ar. *Guía del Denunciante*. (s. f.). Recuperado 11 de octubre de 2021, de https://www.argentina.gob.ar/sites/default/files/guia_del_denunciante_0.pdf
- Big Data Advanced Analytics. *¿Qué es el Text Mining y cuáles son sus aplicaciones?* – *Máster en Big Data Málaga*. (s. f.). Recuperado 1 de octubre de 2021, de <https://www.bigdata.uma.es/que-es-el-text-mining-y-cuales-son-sus-aplicaciones/>
- Blanco, E. J., & Sanz, H. (2016, enero). *Algoritmos de clustering y aprendizaje automático aplicados a Twitter*. Universitat Politècnica de Catalunya.
- Brownlee, J. (2016). *Machine Learning Mastery With Weka*. Machine Learning Mastery.
- Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., & Wirth, R. (2000). *CRISP-DM 1.0: Step-by-step data mining guide*. SPSS.
- Consejo Profesional de Ciencias Informáticas de la Provincia de Córdoba. *Honorarios Recomendados Agosto 2021*. (2021, agosto). Recuperado 8 de noviembre de 2021, de https://www.cpcipc.org.ar/sites/default/files/honorarios_recomendados_agosto_2021.pdf
- Escuela de Organización Industrial. (2016). *Matriz de Probabilidad de Impacto*. Recuperado 13 de noviembre de 2021, de <https://www.eoi.es/blogs/mcalidadon/2016/02/03/la-matriz-probabilidad-impacto/>
- Espino, A. I. L., Mur, R. A., & de Miguel, M. A. S. (2004). *Aprendizaje automático en conjuntos clasificadores heterogéneos y modelado de agentes*. Universidad Carlos III de Madrid, Departamento de Informática.
- Fernández Jauregui, A. (s.f.). *DBSCAN en Python: aprende cómo funciona*. Ander Fernández Jauregui. Data Scientist & Business Intelligence. Recuperado 30 de octubre de 2021, de <https://anderfernandez.com/blog/dbscan-python/>

-
- Garre, M., Cuadrado, J., Sicilia, M. A., Rodríguez, D., & Rejas, R. (2007). Comparación de diferentes algoritmos de clustering en la estimación de coste en el desarrollo de software. *Revista Española de Innovación, Calidad e Ingeniería del Software*. Published.
- González, L. (2020, 8 septiembre). *Algoritmo Agrupamiento Jerárquico – Teoría*. Aprende IA. Recuperado 30 de abril de 2022, de <https://aprendeia.com/algoritmo-agrupamiento-jerarquico-teoria/>
- Google. (2020). *Colaboratory*. Recuperado 14 de noviembre de 2021, de <https://research.google.com/colaboratory/intl/es/faq.html>
- Hernández Orallo, J., Ramírez Quintana, J., & Ferri Ramírez, C. (2004). *Introducción a la Minería de Datos*. Pearson.
- Hofmann, M., & Klinkenberg, R. (2013). *Rapidminer: Data Mining Use Cases and Business Analytics Applications*. CRC Press.
- Jing, L. (2008). *Survey of text clustering*. Department of Mathematics, the University of Hong Kong.
- Maklin, C. (2019, 30 junio). *DBSCAN Python Example: The Optimal Value For Epsilon (EPS)*. Towards Data Science. Recuperado 25 de abril de 2022, de <https://towardsdatascience.com/machine-learning-clustering-dbscan-determine-the-optimal-value-for-epsilon-eps-python-example-3100091cfbc>
- Malik, U. (s.f.). *Hierarchical Clustering with Python and Scikit-Learn*. Stack Abuse. Recuperado 26 de octubre de 2021, de <https://stackabuse.com/hierarchical-clustering-with-python-and-scikit-learn/>
- Mantenimiento Informático Económico. (s.f.). *Sistemas operativos. Diferencias y similitudes*. Recuperado 14 de noviembre de 2021, de <https://mantenimientoinformaticoeconomico.com/2020/07/21/sistemas-operativos-diferencias-y-similitudes/>
- Master Data Scientist. *Diferencia entre Data Mining y Text Mining*. (s. f.). Recuperado 2 de octubre de 2021, de <https://www.master-data-scientist.com/diferencias-data-mining-text-mining/>
-

- Minitab. (s.f.). *Dendrograma*. Soporte de Minitab. Recuperado 30 de abril de 2022, de <https://support.minitab.com/es-mx/minitab/18/help-and-how-to/modeling-statistics/multivariate/how-to/cluster-observations/interpret-the-results/all-statistics-and-graphs/dendrogram/>
- Mullin, T. (2020, 9 julio). *DBSCAN Parameter Estimation Using Python*. Medium. Recuperado 20 de abril de 2022, de <https://medium.com/@tarammullin/dbscan-parameter-estimation-ff8330e3a3bd>
- Observatorio de Seguridad Ciudadana. *Prevalencias Delictivas en Argentina*. (s. f.). Recuperado 25 de octubre de 2021, de <http://www.seguridadciudadana.org.ar/recursos/articulos/prevalencias-delictivas-en-argentina>
- Observatorio de Seguridad Ciudadana. *Estadísticas Criminales 2020*. (s. f.). Recuperado 25 de octubre de 2021, de <http://www.seguridadciudadana.org.ar/estadisticas/datos-a-nivel-subnacional/estadisticas-criminales-2>
- Periódico Judicial. *Robo y Hurto: Diferencias y Penas*. (s. f.). Recuperado 11 de octubre de 2021, de <https://www.periodicojudicial.gov.ar/?p=418204>
- Project Management Institute (2017). *A Guide to the Project Management Body of Knowledge (PMBOK(R) Guide-Sixth Edition / Agile Practice Guide Bundle*. Project Management Institute.
- Sanabria Ruiz, V. A. (2017). *Aplicación de técnicas de agrupamiento (clustering) para el análisis estadístico de tendencias en twitter basado en el lenguaje de programación R*. Universidad Industrial de Santander.
- Scikit Learn (2020). *Clustering*. Recuperado 15 de abril de 2022, de <https://scikit-learn.org/stable/modules/clustering.html#adjusted-rand-score>
- Tan, P., Steinbach, M., & Kumar, V. (2005). *Introduction to Data Mining*. Addison-Wisley.
- Venables, W. N. (2009). *An Introduction to R 2nd edition by Venables, William N., Smith, David M. (2009) Paperback*. Network Theory.

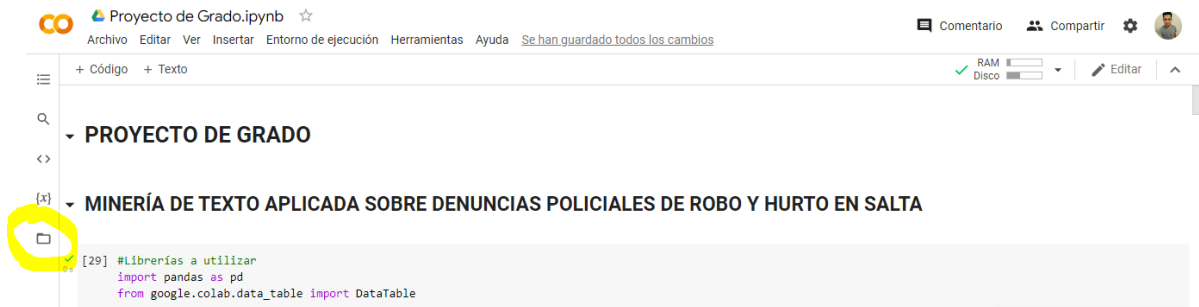
Witten, I. H., Frank, E., Hall, M. A., & Pal, C. (2016). *Data Mining: Practical Machine Learning Tools and Techniques* (4th ed.). Morgan Kaufmann Publishers.

Zhao, Y. (2012). *R and Data Mining: Examples and Case Studies* (1. ed.). Academic Press.

Anexos

Anexo I: Vinculación de Google Drive con Google Colaboratory

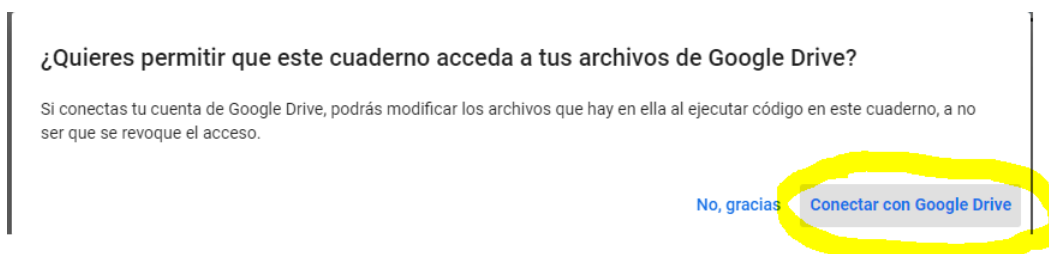
Ubicado en el proyecto de minería de texto en Google Colaboratory, se debe hacer click en el símbolo de “Archivos” ubicado a la izquierda como muestra la imagen:



Una vez desplegada la pestaña de archivos, se debe seleccionar la opción de “Montar Drive” como se muestra en la imagen.



Se debe permitir el acceso a Drive haciendo click en la opción mostrada a continuación:



Una vez colocados el nombre de usuario y la contraseña ya se podrá acceder a los archivos ubicados en la carpeta de Google Drive.

Anexo II: Codificación

A continuación, se van a colocar todas aquellas líneas de código que fueron utilizadas en el desarrollo del proyecto, pero que no aportan demasiado valor al lector, por lo que no fueron incluidas en la redacción de los capítulos de esta tesis.

- **Importación/Instalación de librerías:**

```
#Librerías a utilizar
import pandas as pd
from google.colab.data_table import DataTable
```

```
#Librerías para tokenizar
import nltk
import re
```

```
#Instalación de nuevas librerías: funpymodeling
!pip install funpymodeling
```

```
from funpymodeling.exploratory import status
```

```
#Librerías
import numpy as np
```

```
#Librerías
nltk.download('punkt')
nltk.download('stopwords')
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
True
```

```
#Instalación de librerías
!python -m spacy download es_core_news_sm
!python -m spacy link es_core_news_sm es
import spacy
nlp = spacy.load('es')
```

```
#Librerías a utilizar
from datetime import datetime
```

```
#Librerías
from nltk.util import ngrams
import heapq
```

```
#Librerías
from sklearn.feature_extraction.text import TfidfVectorizer
```

```
#Librerías
from sklearn.cluster import KMeans
```

```
#Librerías
import matplotlib.pyplot as plt
```

```
#Librerías
from sklearn.metrics import silhouette_score
```

```
#Librerías
from sklearn.neighbors import NearestNeighbors
```

```
#Librerías
from sklearn.cluster import DBSCAN
```

```
#Librerías
import scipy.cluster.hierarchy as sch
from scipy.cluster.hierarchy import dendrogram
```

```
#Librerías
from sklearn.cluster import AgglomerativeClustering
```

- Copias de dataframe:

```
#Copia del dataframe
d_denuncias_token = d_denuncias
```

```
#Copia del dataframe
d_denuncias_selec = d_denuncias_prep
```

```
#Copia del dataframe
d_bow = d_denuncias_selec
```

```
#Copia del dataframe  
d_denuncias_final = d_denuncias_selec
```

```
#Copia del dataframe  
d_denuncias_cluster = d_denuncias_final
```

- Ejecución de funciones:

```
#Ejecución de la función  
bow = bow(d_denuncias_token, "RELATO")
```

```
#Ejecución de la función  
tipo_delito(d_denuncias_selec, "Delito")
```

```
#Ejecución de la función  
d_denuncias_prep = repetidos(d_denuncias_prep, "RELATO")
```

```
#Ejecución de la lematización  
denuncias_lem = lematizar(prepare_f)
```

```
#Ejecución de la función  
fecha(d_denuncias_selec, "fchHrHecho", "MesHecho", "DiaHecho")  
d_denuncias_selec
```

```
#Ejecución de la función  
bow_f = bow_f(d_bow, "Relato")
```

```
#Ejecución de la función  
denuncias_fin = bigramas(denuncias)
```

```
#Ejecución de la función  
denuncias_fin = trigramas(denuncias_fin)
```