



UNIVERSIDAD PONTIFICIA DE SALAMANCA  
FACULTAD DE INFORMÁTICA  
Grado en Informática

Trabajo Fin de Grado

**PREDICCIÓN DE LA CATEGORÍA DE CRÍMENES  
OCURRIDOS EN SAN FRANCISCO**

MAITE ECHEVERRY MOLINA

Dr. Manuel Martín Merino Acera  
DIRECTOR

Salamanca, Mayo de 2016



## **Resumen**

El análisis de los registros de crímenes es fundamental en la prevención de delitos. Entre otras cosas, porque permite el diseño de políticas y planes de prevención efectivos, este análisis se ha realizado históricamente mediante herramientas estadísticas descriptivas básicas, considerando fundamentalmente variables y relaciones primarias. Sin embargo, muchas veces la estadística descriptiva clásica no refleja la verdadera interrelación de las variables y por lo tanto, el problema real. Este contexto requiere un tratamiento más complejo que obliga a evolucionar en el análisis de información de este tipo.

En este contexto, el objetivo de este trabajo es realizar una implementación de minería de datos en el análisis de información de los crímenes y comprobar su efectividad y valor añadido. Para ello se trabajará en la identificación y detección de categorías de crímenes cometidos en San Francisco entre los años 2003–2015.

## **Abstract**

Analysis Crime Records is critical in the prevention of crimes . Among other things, it allows the design of Policies and Effective Prevention Plans , this analysis has been done historically Using Basic Tools Descriptive statistics , mainly considering the variables and primary relationships . However, many times without Classical descriptive statistics reflect the true interrelation of variables and therefore what the real problem . This context requires UN complex treatment More Than one obli evolve in the Analysis of such information .

In this context, the m objective of this work is to Implementation of Data Mining Information Analysis Crimes and check v Do Effectiveness and added value. This will work for the identification and detection of categories of crimes in San Francisco between 2003-2015

## **Descriptores**

Exploración, Análisis, minería de datos, crímenes, preparación, predicciones, clasificación, inseguridad, big data



## ÍNDICE

<b>1.</b>	<b>PROCESO DE EXTRACCIÓN DEL CONOCIMIENTO .....</b>	<b>1</b>
1.1	ANÁLISIS EXPLORATORIO .....	2
1.2	LIMPIEZA DE DATOS (ELIMINACIÓN DE RUIDO E INCONSISTENCIAS) .....	9
1.3	INTEGRACIÓN DE DATOS (AGRUPACIÓN, OBTENCIÓN DE NUEVAS VARIABLES) .....	9
1.4	REDUCCIÓN/SELECCIÓN DE DATOS (IDENTIFICACIÓN DE DATOS RELEVANTES PARA EL PROBLEMA)	10
1.5	TRANSFORMACIÓN DE DATOS (PREPARACIÓN DE LOS DATOS PARA SU ANÁLISIS) .....	11
<b>2.</b>	<b>MINERÍA DE DATOS (TÉCNICAS DE EXTRACCIÓN DE PATRONES Y MEDIDAS DE INTERÉS) 13</b>	
2.1	ASPECTOS ALGORÍTMICOS DE LA MINERÍA DE DATOS.....	14
2.2	SELECCIÓN DE LOS ALGORITMOS DE LA MINERÍA DE DATOS .....	14
2.3	EVALUACIÓN DE LOS MODELOS.....	15
<b>3.</b>	<b>CONOCIMIENTO Y VISUALIZACIÓN DE LOS RESULTADOS .....</b>	<b>23</b>
3.1	TÉCNICAS DE VISUALIZACIÓN .....	24
3.2	TÉCNICAS DE REPRESENTACIÓN CON PYTHON .....	28
<b>4.</b>	<b>CONCLUSIONES .....</b>	<b>31</b>
<b>5.</b>	<b>BIBLIOGRAFÍA .....</b>	<b>33</b>

## ÍNDICE DE FIGURAS

FIGURA 1-1. DIAGRAMA DEL PROCESO DE EXTRACCIÓN DEL CONOCIMIENTO.....	1
FIGURA 1-2. VISUALIZACIÓN DE LA CATEGORÍA DE CRÍMENES. ....	4
FIGURA 1-3.A. COEFICIENTE DE VARIACIÓN.....	5
ILUSTRACIÓN 1-4. CANTIDAD POR DÍA DE ALGUNOS DE LOS DELITOS EN ESTUDIO. ....	6
FIGURA 1-5. VISUALIZACIÓN DE CRÍMENES TOTALES POR DISTRITO.....	6
ILUSTRACIÓN 1-6. VISUALIZACIÓN DE ALGUNOS DE LOS CRÍMENES DE MAYOR INCIDENCIA POR DISTRITO.....	7
ILUSTRACIÓN 1-7. DELITOS DE HURTO/ROBO DISTRIBUIDOS POR DÍA DE LA SEMANA. ....	7
ILUSTRACIÓN 1-8. REPRESENTACIÓN GRÁFICA DEL DÍA DE LA SEMANA CON MAYOR NÚMERO DE DELITOS.....	8
ILUSTRACIÓN 1-9. HORA DE LA SEMANA CON MAYOR ÍNDICE DE DELINCUENCIA .....	8
ILUSTRACIÓN 2-1. ESQUEMA DE MINERÍA DE DATOS .....	13
ILUSTRACIÓN 2-2. EJEMPLO DEL MODELO LINEAR SVM .....	15
ILUSTRACIÓN 2-3. EJEMPLO DEL MODELO KMN. .....	17
ILUSTRACIÓN 2-4. EJEMPLO DEL MODELO RANDOM FOREST.....	19

## ÍNDICE DE TABLAS

TABLA 1-1. CONJUNTO DE DATOS UTILIZADOS PARA EL ANÁLISIS EXPLORATORIO.....	2
TABLA 1-2. CONTROL DE LA CANTIDAD DE VALORES ÚNICOS .....	3
TABLA 1-3. CABECERA DE ENTRENAMIENTO .....	3
TABLA 1-4. CABECERA DE TEST .....	3
TABLA 1-5. CABECERA DE TEST ELIMINANDO LA CATEGORÍA.....	4
TABLA 1-6. TOTAL POR CATEGORÍAS. ....	4
TABLA 1-7. CRIMEN CON MAS ALTO COEFICIENTE DE VARIACIÓN POR DÍA .....	5



# 1. Proceso de extracción del conocimiento

Los datos utilizados para este trabajo han seguido el proceso de extracción recogidos en la Figura 1.1.

Partiendo de una amplia base de datos, se procedido a especificar cual es el problema de los mismos, comprendiendo su finalidad. Posteriormente, se ha efectuado una limpieza de los datos para eliminar datos no válidos. A partir de ahí, los pasos a seguir han sido el pre-procesamiento, la minería de datos y la posterior evaluación e interpretación de los mismos.

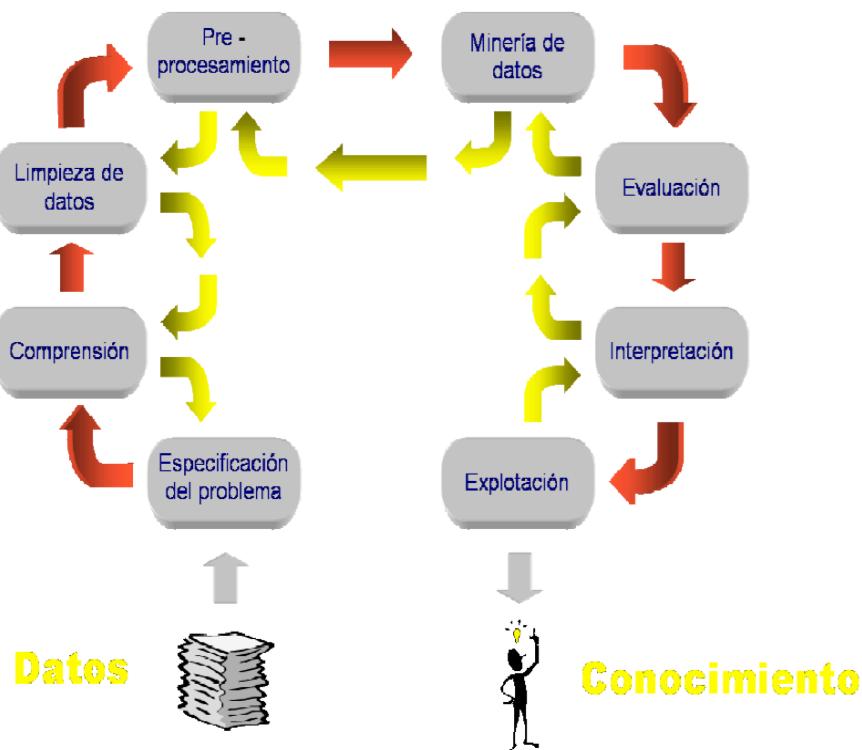


Figura 1-1. Diagrama del Proceso de Extracción del Conocimiento.

## 1.1 Análisis exploratorio

Comenzamos con el análisis exploratorio de la base de datos que recoge los crímenes ocurridos en San Francisco que se recogen a modo de resumen en la siguiente tabla (Tabla 1-1).

**Tabla 1-1. Conjunto de datos utilizados para el análisis exploratorio.**

2003-01-07 07:52:00	WARRANTS	WARRANT ARREST	Tuesday	SOUTHERN	ARREST, BOOKED	5TH ST / SHIPLEY ST	-122.402843	37.779829
2003-01-07 04:49:00	WARRANTS	ENROUTE TO OUTSIDE JURISDICTION	Tuesday	TENDERLOIN	ARREST, BOOKED	CYRIL MAGNIN STORTH ST / EDDY ST	-122.408495	37.784452
2003-01-07 03:52:00	WARRANTS	WARRANT ARREST	Tuesday	NORTHERN	ARREST, BOOKED	OFARRELL ST / LARKIN ST	-122.417904	37.785167
2003-01-07 03:34:00	WARRANTS	WARRANT ARREST	Tuesday	NORTHERN	ARREST, BOOKED	DIVISADERO ST / LOMBARD ST	-122.442650	37.798999
2003-01-07 01:22:00	WARRANTS	WARRANT ARREST	Tuesday	SOUTHERN	ARREST, BOOKED	900 Block of MARKET ST	-122.409537	37.782691
2003-01-06 23:30:00	WARRANTS	ENROUTE TO OUTSIDE JURISDICTION	Monday	BAYVIEW	ARREST, BOOKED	REVERE AV / INGALLS ST	-122.384557	37.728487
2003-01-06 23:14:00	WARRANTS	WARRANT ARREST	Monday	CENTRAL	ARREST, BOOKED	BUSH ST / HYDE ST	-122.417019	37.789110
2003-01-06 22:45:00	WARRANTS	WARRANT ARREST	Monday	SOUTHERN	ARREST, BOOKED	800 Block of BRYANT ST	-122.403405	37.775421
2003-01-06 22:45:00	WARRANTS	ENROUTE TO OUTSIDE JURISDICTION	Monday	SOUTHERN	ARREST, BOOKED	800 Block of BRYANT ST	-122.403405	37.775421
2003-01-06 22:19:00	WARRANTS	ENROUTE TO OUTSIDE JURISDICTION	Monday	NORTHERN	ARREST, BOOKED	GEARY ST / POLK ST	-122.419740	37.785893
2003-01-06 21:54:00	WARRANTS	ENROUTE TO OUTSIDE JURISDICTION	Monday	NORTHERN	ARREST, BOOKED	SUTTER ST / POLK ST	-122.420120	37.787757

Este conjunto de datos contiene incidentes derivados del sistema de policía de San Francisco del crimen de notificación de incidentes.

Los datos oscilan entre 01/01/2003 hasta 05/13/2015. El conjunto de entrenamiento y de prueba rotan cada semana; es decir, la semana 1,3,5,7... pertenecen a ensayo de deformación, la semana 2,4,6,8 pertenecen al conjunto de entrenamiento.

Siendo:

- **DATES**: Hace referencia al día y la hora a la que se dio el crimen
- **CATEGORY**: Categoría que se le asigna al crimen
- **DESCRIPT**: Pequeña descripción del crimen
- **DAYOFWEEK**: Día de la semana en el que se cometió el crimen
- **PdDISTRICT**: Nombre del distrito de San Francisco donde se cometió el crimen
- **RESOLUTION**: Solución que se aplicó al crimen
- **ADDRESS**: Dirección donde sucedió el crimen
- **X**: Coordenada X de la posición donde se realizó el crimen
- **Y**: Coordenada Y de la posición donde se realizó el crimen

En la siguiente tabla (Tabla 1-2) presentamos información general de los datos:

**Tabla 1-2. Control de la cantidad de valores únicos**

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 878049 entries, 0 to 878048
Data columns (total 9 columns):
Dates          878049 non-null object
Category       878049 non-null object
Descript        878049 non-null object
DayOfWeek       878049 non-null object
PdDistrict     878049 non-null object
Resolution      878049 non-null object
Address         878049 non-null object
X              878049 non-null float64
Y              878049 non-null float64
dtypes: float64(2), object(7)
memory usage: 67.0+ MB
```

En cuanto a los datos de interés, existen 39 categorías de delitos. Así que sabemos según el análisis solicitado que será nuestra variable independiente. Además se observan 879 “descripts” únicos, en comparación con 389.257 fechas. Esto significa que hay una gran cantidad de “descripts” duplicados. También se observa que existen 10 distritos diferentes. Un número bastante bajo que podría ayudar a que en la visualización la agrupación sea mejor.

**Tabla 1-3. Cabecera de entrenamiento**

```
Uniques:
Unique in 'Dates': 389257
Unique in 'Category': 39
Unique in 'Descript': 879
Unique in 'DayOfWeek': 7
Unique in 'PdDistrict': 10
Unique in 'Resolution': 17
Unique in 'Address': 23228
Unique in 'X': 34243
Unique in 'Y': 34243
```

Con el fin de tener una idea de qué tipo de datos se almacenan en el conjunto de datos se hará un impresión de las cabeceras.

**Tabla 1-4. Cabecera de test.**

	Dates	Category	Descript	DayOfWeek	PdDistrict	Resolution	Address	X	Y
0	2015-05-13 23:53:00	WARRANTS	WARRANT ARREST	Wednesday	NORTHERN	ARREST, BOOKED	OAK ST / LAGUNA ST	-122.425892	37.774599
1	2015-05-13 23:53:00	OTHER OFFENSES	TRAFFIC VIOLATION ARREST	Wednesday	NORTHERN	ARREST, BOOKED	OAK ST / LAGUNA ST	-122.425892	37.774599
2	2015-05-13 23:33:00	OTHER OFFENSES	TRAFFIC VIOLATION ARREST	Wednesday	NORTHERN	ARREST, BOOKED	VANNES AV / GREENWICH ST	-122.424363	37.800414
3	2015-05-13 23:30:00	LARCENY/THEFT	GRAND THEFT FROM LOCKED AUTO	Wednesday	NORTHERN	NONE	1500 Block of LOMBARD ST	-122.426995	37.800873
4	2015-05-13 23:30:00	LARCENY/THEFT	GRAND THEFT FROM LOCKED AUTO	Wednesday	PARK	NONE	100 Block of BRODERICK ST	-122.438738	37.771541

## PREDICCIÓN DE LA CATEGORÍA DE CRÍMENES OCURRIDOS EN SAN FRANCISCO

---

**Tabla 1-5. Cabecera de test eliminando la categoría.**

	<b>Id</b>	<b>Dates</b>	<b>DayOfWeek</b>	<b>PdDistrict</b>	<b>Address</b>	<b>X</b>	<b>Y</b>
<b>0</b>	0	2015-05-10 23:59:00	Sunday	BAYVIEW	2000 Block of THOMAS AV	-122.399588	37.735051
<b>1</b>	1	2015-05-10 23:51:00	Sunday	BAYVIEW	3RD ST / REVERE AV	-122.391523	37.732432
<b>2</b>	2	2015-05-10 23:50:00	Sunday	NORTHERN	2000 Block of GOUGH ST	-122.426002	37.792212
<b>3</b>	3	2015-05-10 23:45:00	Sunday	INGLESIDE	4700 Block of MISSION ST	-122.437394	37.721412
<b>4</b>	4	2015-05-10 23:45:00	Sunday	INGLESIDE	4700 Block of MISSION ST	-122.437394	37.721412

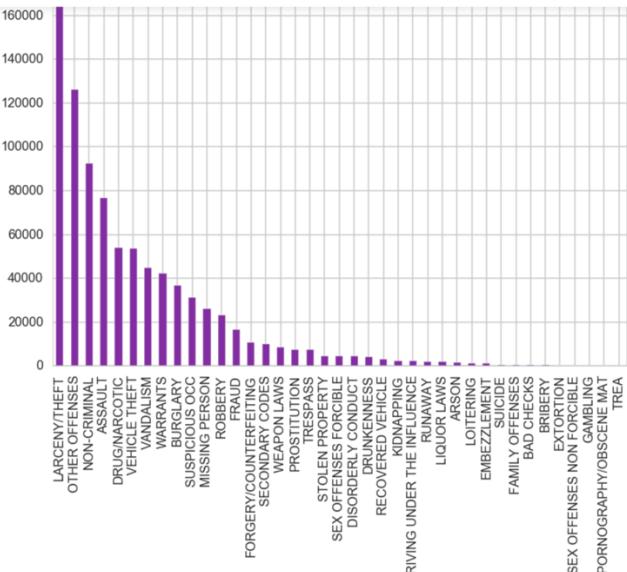
Para tener una idea clara de la diversidad de las categorías imprimimos los nombres únicos que aparecen en "Categoría", y que se muestran a continuación.

**Tabla 1-6. Total por categorías.**

```
[ 'WARRANTS' 'OTHER OFFENSES' 'LARCENY/THEFT' 'VEHICLE THEFT' 'VANDALISM'
'NON-CRIMINAL' 'ROBBERY' 'ASSAULT' 'WEAPON LAWS' 'BURGLARY'
'SUSPICIOUS OCC' 'DRUNKENNESS' 'FORGERY/COUNTERFEITING' 'DRUG/NARCOTIC'
'STOLEN PROPERTY' 'SECONDARY CODES' 'TRESPASS' 'MISSING PERSON' 'FRAUD'
'KIDNAPPING' 'RUNAWAY' 'DRIVING UNDER THE INFLUENCE'
'SEX OFFENSES FORCIBLE' 'PROSTITUTION' 'DISORDERLY CONDUCT' 'ARSON'
'FAMILY OFFENSES' 'LIQUOR LAWS' 'BRIBERY' 'EMBEZZLEMENT' 'SUICIDE'
'LOITERING' 'SEX OFFENSES NON FORCIBLE' 'EXTORTION' 'GAMBLING'
'BAD CHECKS' 'TREA' 'RECOVERED VEHICLE' 'PORNOGRAPHY/OBSCENE MAT' ]
```

Visualizar el recuento de cada categoría (Figura 1-2), nos ayudará a ver qué categorías son más relevantes o cuáles contienen una gran cantidad de muestras.

LARCENY/THEFT	174900
OTHER OFFENSES	126182
NON-CRIMINAL	92304
ASSAULT	76876
DRUG/NARCOTIC	53971
VEHICLE THEFT	53781
VANDALISM	44725
WARRANTS	42214
BURGLARY	36755
SUSPICIOUS OCC	31414
MISSING PERSON	25989
ROBBERY	23000
FRAUD	16679
FORGERY/COUNTERFEITING	10609
SECONDARY CODES	9985
WEAPON LAWS	8555
PROSTITUTION	7484
TRESPASS	7326
STOLEN PROPERTY	4540
SEX OFFENSES FORCIBLE	4388
DISORDERLY CONDUCT	4320
DRUNKENNESS	4280
RECOVERED VEHICLE	3138
KIDNAPPING	2341
DRIVING UNDER THE INFLUENCE	2268
RUNAWAY	1946
LIQUOR LAWS	1903
ARSON	1513
LOITERING	1225
EMBEZZLEMENT	1166
SUICIDE	508
FAMILY OFFENSES	491
BAD CHECKS	406
BRIBERY	289
EXTORTION	256
SEX OFFENSES NON FORCIBLE	148
GAMBLING	146
PORNOGRAPHY/OBSCENE MAT	22
TREA	6



**Figura 1-2. Visualización de la Categoría de crímenes.**

Mediante la visualización de las siguiente figura (1-3), podemos notar una varianza entre medio día, viernes y sábado tienen valores más altos. No por un amplio margen, pero parece que hay una diferencia comparada con los otros días.

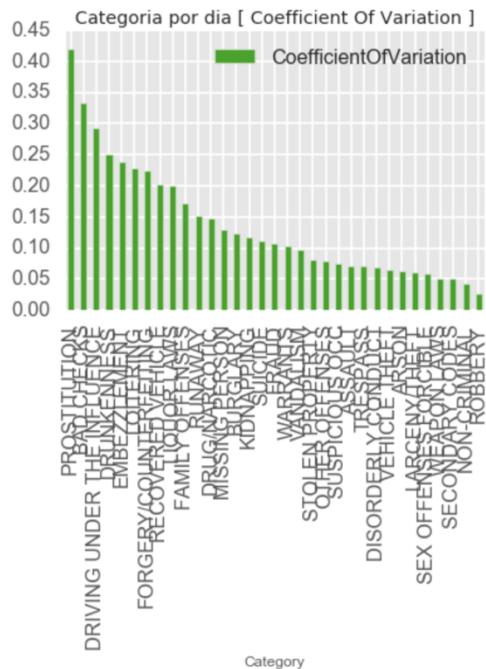


Figura 1-3.A. Coeficiente de variación

Teniendo en cuenta esto, vamos a analizar qué grupo tiene el mayor coeficiente de variación por día.

Tabla 1-7. Crimen con mas alto coeficiente de variación por día

```

Top 5 Coefficient Of Variation by day:
23           PROSTITUTION
31           BAD CHECKS
21   DRIVING UNDER THE INFLUENCE
11           DRUNKNESS
28           EMBEZZLEMENT
Name: Category, dtype: object
Bottom 5 Coefficient Of Variation by day:
22   SEX OFFENSES FORCIBLE
8            WEAPON LAWS
15          SECONDARY CODES
5            NON-CRIMINAL
6             ROBBERY
Name: Category, dtype: object
    
```

Algunas de las categorías parecen tener una mayor variación por día que otras. Para ver las diferencias, mostraremos el Top 5 categorías diferentes por día. Ejemplo de código:

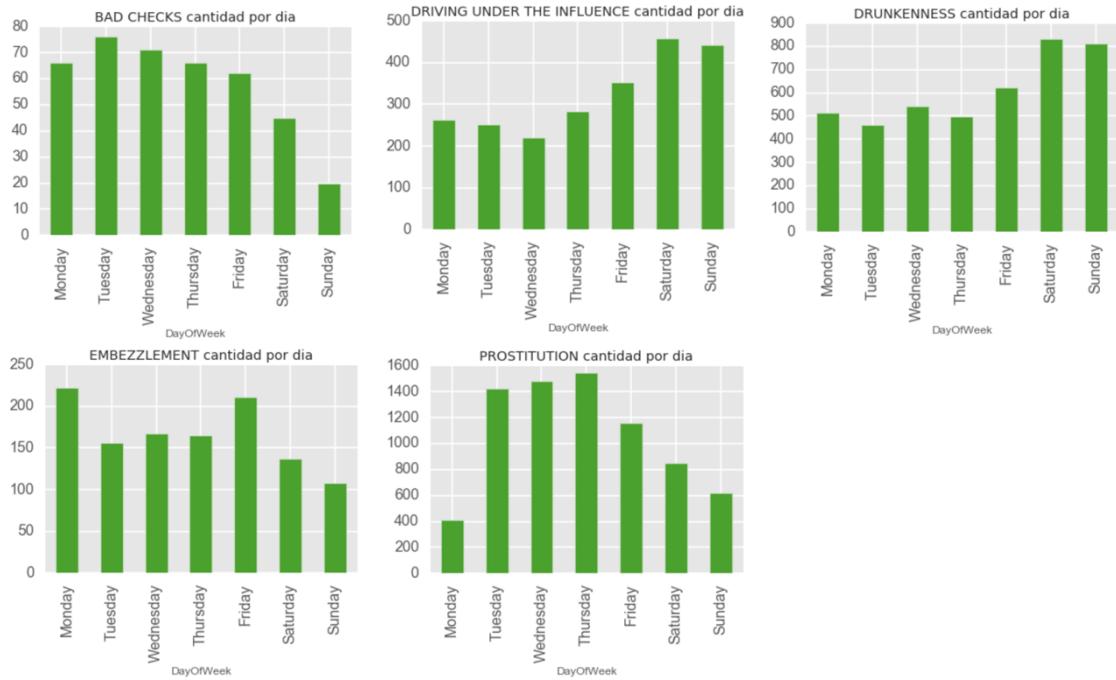
```

for category in categoryDayCV["Category"][:5]:
    dfCategory = dftrain[dftrain["Category"] == category]
    groups = dfCategory.groupby("DayOfWeek")["Category"].count()
    weekdays = ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday']
    groups = groups[weekdays]
    plt.figure()
    groups.plot(kind="bar", color="#4AA02C", title=category + " cantidad por dia")
    
```

## PREDICCIÓN DE LA CATEGORÍA DE CRÍMENES OCURRÍDOS EN SAN FRANCISCO

---

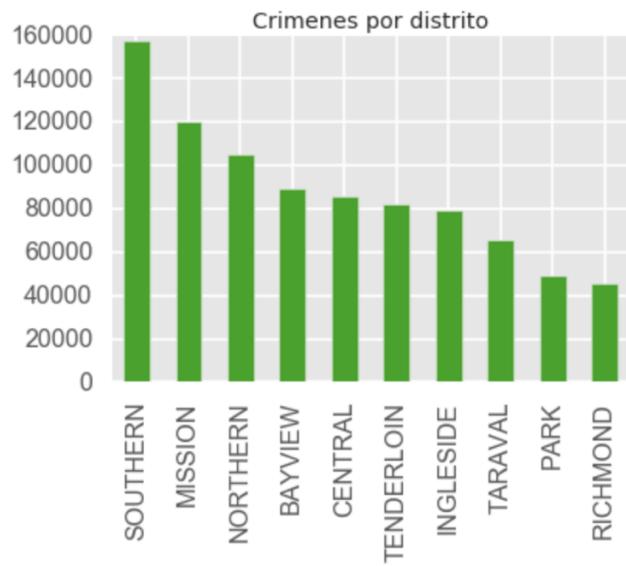
Si representamos algunos de los crímenes por día (Figura 1-4), podemos observar que algunos delitos parecen ocurrir con más frecuencia los fines de semana. Por ejemplo, la embriaguez y la conducción bajo esta influencia.



**Ilustración 1-4. Cantidad por día de algunos de los delitos en estudio.**

En cuanto al total de distritos estudiados, en la siguiente figura se recogen los crímenes agrupados por distritos, siendo Sourthern y Richmond los de mayor y menor incidencia respectivamente.

SOUTHERN	157182
MISSION	119908
NORTHERN	105296
BAYVIEW	89431
CENTRAL	85460
TENDERLOIN	81809
INGLESIDE	78845
TARAVAL	65596
PARK	49313
RICHMOND	45209



**Figura 1-5. Visualización de crímenes totales por distrito.**

A modo de ejemplo se recogen las representaciones gráficas (Figura 1-6) de los delitos de violación, drogas, asaltos, y venta de drogas por distrito.

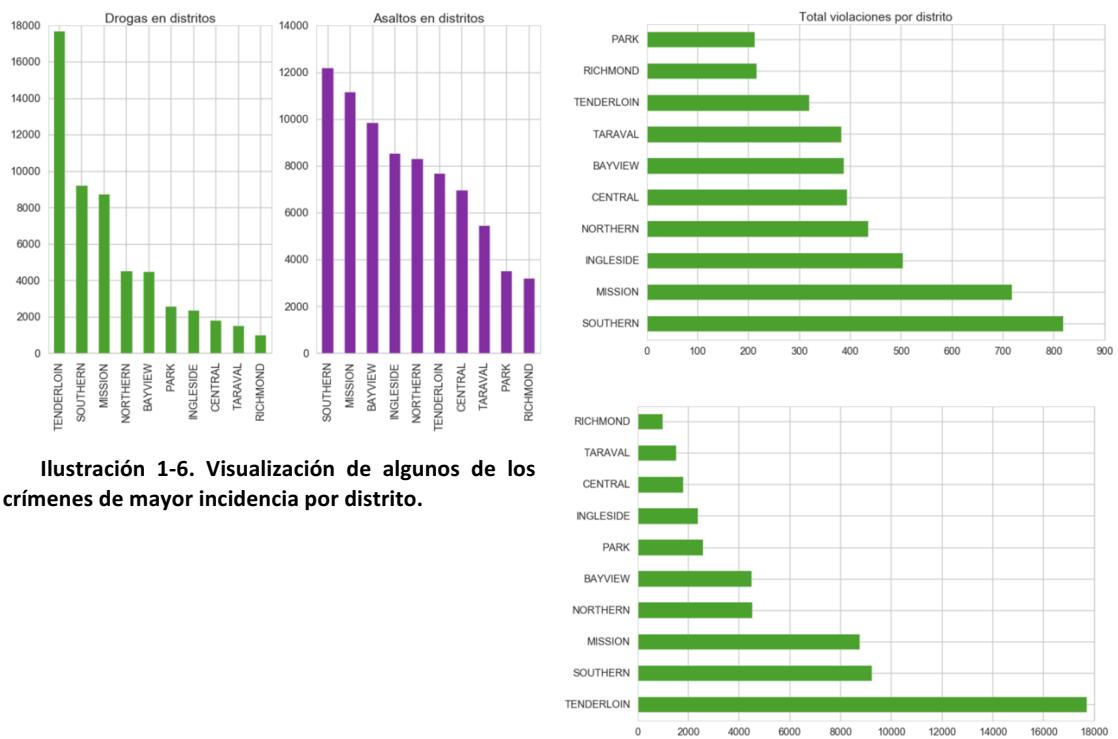


Ilustración 1-6. Visualización de algunos de los crímenes de mayor incidencia por distrito.

Por otro lado, es posible observar que los recuentos de las categorías tienen una gran variedad; hasta un factor de >1000. Si queremos comparar la desviación estándar o la varianza más adelante, debemos utilizar una proporción en relación con el recuento global o la media para cada categoría para obtener resultados comparables. Cabe señalar que la categoría más importante, con mucho, es HURTO/ROBO. Dicha categoría se muestra a continuación distribuida por días de la semana (Figura 1-6), observando que el viernes y sábado son los días de mayor incidencia en este tipo de delitos.

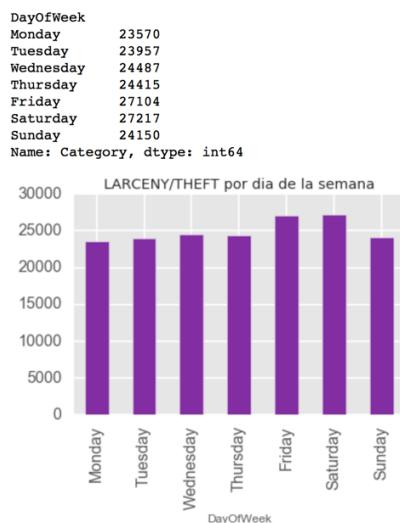


Ilustración 1-7. Delitos de hurto/robo distribuidos por día de la semana.

Teniendo en cuenta lo anterior, vamos a investigar más a fondo por el trazado de los crímenes de ocurrencia para cada día de la semana, siendo el sábado el de mayor cantidad con 27.217.

En la Figura 1-8 se muestra que, con la excepción del distrito CENTRAL que es el sábado, el viernes es el día de la semana con mayor incidencia de crímenes.

DayOfWeek	
Monday	23570
Tuesday	23957
Wednesday	24487
Thursday	24415
Friday	27104
Saturday	27217
Sunday	24150

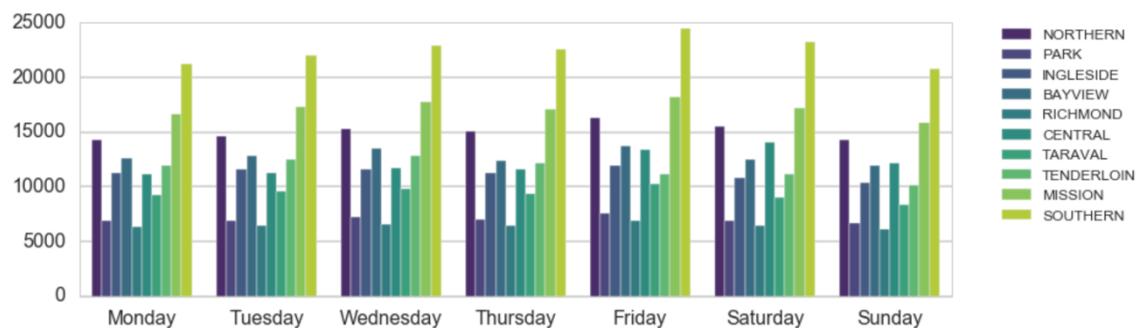


Ilustración 1-8. Representación gráfica del día de la semana con mayor número de delitos

En este sentido, si representamos los delitos distribuidos por horas del día, observamos que de lunes a sábado las horas del crepúsculo (entre las 20:00 a 22:00 horas) son las de mayor número de crímenes. El sábado y el domingo el horario de criminalidad se desplaza ligeramente a horas entre las 22:00 a 01:00 horas.

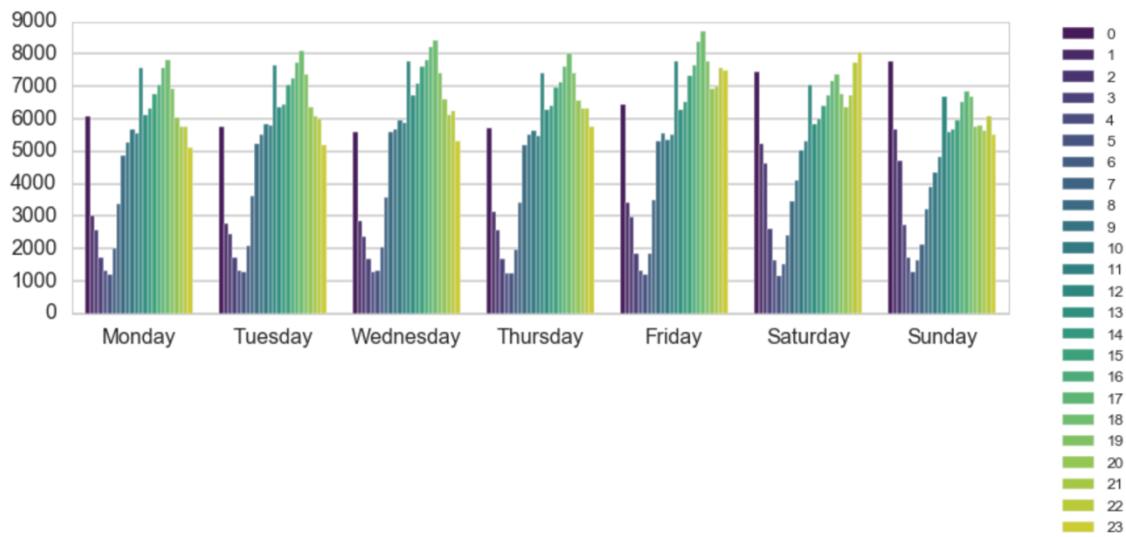


Ilustración 1-9. Hora de la semana con mayor índice de delincuencia

## 1.2 Limpieza de datos (eliminación de ruido e inconsistencias)

Este es un paso preparatorio inicial en donde se prepara el escenario para la comprensión de lo que se debe hacer con las muchas decisiones (acerca de la limpieza y preparación de los datos para la aplicación de los modelos).

- Eliminación en el entrenamiento y test los datos nulos, debido a la gran cantidad de datos ‘none’ que tenemos en la variable ‘Resolution’, hemos decidido eliminarla del estudio
- Eliminamos del entrenamiento las variables ‘Descript’ y ‘Address’ ya que debido a su variación serán insignificantes para nuestro estudio en el modelo
- Eliminamos del test la variable ‘Address’ ya que disponemos de las coordenadas para este análisis y contar con esta variable sería redundante

Ejemplo de código:

```
dftrain = dftrain.drop(['Descript', 'Resolution', 'Address'], axis = 1)
dftest = dftest.drop(['Address'], axis = 1)
```

## 1.3 Integración de datos (agrupación, obtención de nuevas variables)

Necesitamos entender y definir los objetivos del problema y el medio ambiente en el que el proceso de combinación de variables para el descubrimiento de conocimiento se llevará a cabo (incluyendo el conocimiento previo relevante). Habiendo entendido los objetivos mediante el análisis exploratorio, haremos un pre-procesamiento de separación y creación de una nueva variable intentando conseguir mayor precisión en el modelo.

Dates
2015-05-10 23:59:00
2015-05-10 23:51:00
2015-05-10 23:50:00
2015-05-10 23:45:00
2015-05-10 23:45:00

Ejemplo de código:

Separación de la variable ‘Dates’, y conversión de los datos en una estructura categórica, para la implementación del modelo.

```
Hours=pd.get_dummies(train_SIN.Dates.map(lambda x:pd.to_datetime(x).hour), prefix="hour")
months=pd.get_dummies(train_SIN.Dates.map(lambda x:pd.to_datetime(x).month), prefix="month")
years=pd.get_dummies(train_SIN.Dates.map(lambda x: pd.to_datetime(x).year), prefix="year")
```

## 1.4 Reducción/Selección de datos (identificación de datos relevantes para el problema)

Mediante este proceso de selección de variables, selección de atributos o selección de subconjuntos de variables, es el proceso de selección de un subconjunto de características relevantes para su uso en la construcción del modelo. Los datos contienen muchas características redundantes o irrelevantes. Características redundantes son las que no proporcionan más información que las características seleccionadas.

. Si algunos atributos importantes se pierden, entonces todo el estudio puede fallar. Para el éxito del proceso, es bueno tener en cuenta el mayor número posible de atributos en esta etapa.

En este paso los datos significativos son seleccionados o creados. Buscando los atributos apropiados de entrada y la información de salida para representar la tarea. Es decir, lo primero que se tiene que tener en cuenta antes de comenzar con el proceso, es saber qué es lo que se quiere obtener y cuáles son los datos que nos permitirán realizar esta tarea.

Las técnicas de selección de características proporcionan tres ventajas principales en la construcción de modelos de predicción:

1. Mejoran la interpretabilidad del modelo.
2. Permiten tiempos más cortos de entrenamiento.
3. Generalización mejorada mediante la reducción del sobre-ajuste.

En este punto haremos una selección de variables dependientes e independientes para entrenar y comprobar la predicción de nuestro modelo:

**Variables dependientes: [years, months, hours, day\_of\_week]**

**Variable independiente: [ Category ]**

Ejemplo de código:

```
predictor = train_clean[['PdDistrict','DayOfWeek','Year','Week','Hour']]  
target = train_clean.Category  
pred_train, pred_test, tar_train, tar_test = train_test_split(predictor, target, test_size=0.4)
```

## 1.5 Transformación de datos (preparación de los datos para su análisis)

En esta etapa, se persigue preparar y generar datos de mayor calidad para la minería de datos. Los métodos aquí incluyen la reducción de la dimensión (como la selección y extracción de características, y el muestreo de registros), la transformación de atributos (tales como la discretización de atributos numéricos , y la transformación funcional). Este paso es a menudo crucial para el éxito del análisis, pero por lo general es muy específico para el proyecto. La discretización es un procedimiento de tratamiento de datos que transforma los datos cuantitativos en cualitativos. Las transformaciones discretas de los datos mejoran la comprensión de las reglas descubiertas al transformar los datos de bajo nivel en datos de alto nivel y también reducen significativamente el tiempo de ejecución del algoritmo de inducción.

En nuestro caso, crearemos una estructura categórica con las variables 'hour', 'month', 'year'

Ejemplo de código:

```
train['Year'] = train['Dates'].map(lambda x: x.year)
train['Week'] = train['Dates'].map(lambda x: x.week)
train['Hour'] = train['Dates'].map(lambda x: x.hour)

train['Category'] = pd.Categorical(train.Category).codes
train['DayOfWeek'] = pd.Categorical(train.DayOfWeek).codes
train['PdDistrict'] = pd.Categorical(train.PdDistrict).codes
train_clean = train.dropna()
train_clean.dtypes
```



# 2. Minería de datos (técnicas de extracción de patrones y medidas de interés)

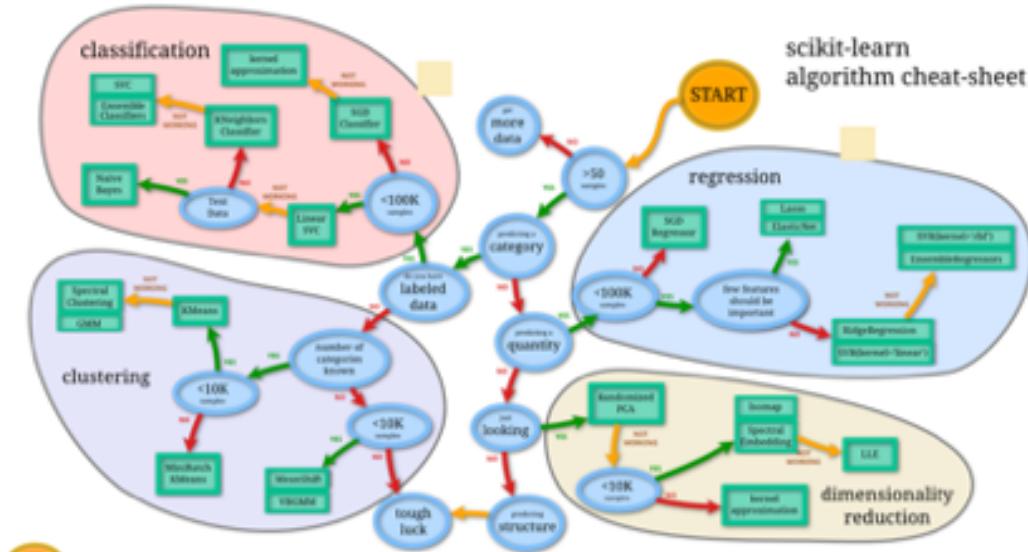


Ilustración 2-1. Esquema de Minería de Datos

Siguiendo la imagen que tenemos aquí (Figura 2.1), con los datos que tenemos, cerca de 900.000 solo en el conjunto de “train”, fuimos siguiendo las instrucciones de los modelos que se iban proponiendo en la imagen.

Debido a la cantidad de datos y la aplicación de cualquier operación tardaba unos cuantos minutos, decidimos trabajar con 100.000 datos del total, y cuando funcionara bien aquí alguno de los modelos lo aplicaríamos al resto.

Utilizamos la función “train\_test\_split” para dividir los 100.000 datos en dos conjuntos “train” y test, partiendo solo del “train”, para poder ver de una mejor forma el índice de acierto de la predicción.

## 2.1 Aspectos algorítmicos de la minería de datos

En función de su propósito general:

- Modelos descriptivos: (describen el comportamiento de los datos de forma que sea interpretable por un usuario experto).
  - Modelos predictivos: (además de describir los datos, se utilizan para predecir el valor de algún atributo desconocido).
1. Arboles de decisión
  2. Métodos Ensemble :
    - 1.1.GradientBoosting
    - 2.1.Bagging
    - 3.1.AdaBoost
    - 4.1.RandomForest
  3. Support Vector Machine SVM
  4. Regresión Logística
  5. Redes Neuronales
  6. Naïve Bayes
  7. K-Vicindad mas Cercana

## 2.2 Selección de los algoritmos de la minería de datos

Se realizarán las estimaciones de los modelos utilizando los pasos previamente descritos en la secciones 1.2, 1.3, 1.4 y 1.5 (etapas del proceso de extracción de conocimiento).

En nuestro caso tenemos un problema de clasificación hemos seleccionado 3 algoritmos para el análisis de nuestro modelo:

- Modelo Linear SVM
- RandomForest
- KNN ( K-Nearest-Neighbour)

## 2.3 Evaluación de los modelos

### A. Modelo Linear SVM

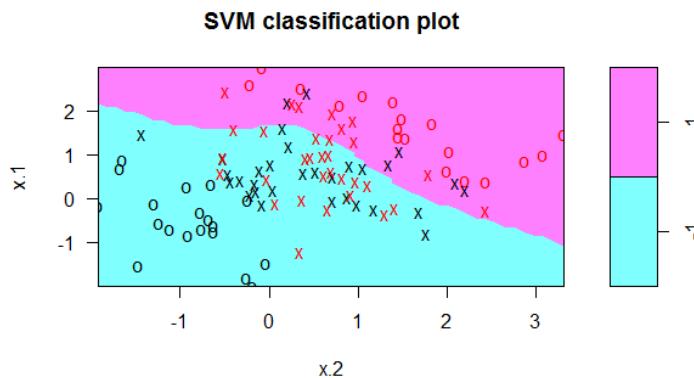


Ilustración 2-2. Ejemplo del modelo linear SVM

Este es el primer modelo que encontramos para aplicar de la imagen. Su funcionamiento se basa principalmente en que teniendo un conjunto de puntos en el que cada uno de estos puede pertenecer a dos posibles categorías, este modelo es capaz de predecir si un punto nuevo pertenece a un grupo u al otro buscando un hiperplano capaz de separar los puntos de una categoría respecto de la otra. Así se puede ver a que grupo, parte del plano, pertenece el nuevo punto.

Para aplicar este modelo importamos el “svm” de la librería “sklearn”. También cargamos el modelo lineal “linearsvc” que vamos a aplicar.

```
from sklearn import svm

clf=svm.LinearSVC()
```

Ahora que ya tenemos cargado el modelo que vamos a utilizar, mediante el método “fit” entrenamos el modelo pasando los parámetro de train que hemos obtenido anteriormente.

```
clf.fit(X_train, Y_train)

LinearSVC(C=1.0, class_weight=None, dual=True, fit_intercept=True,
          intercept_scaling=1, loss='squared_hinge', max_iter=1000,
          multi_class='ovr', penalty='l2', random_state=None, tol=0.0001,
          verbose=0)
```

Con esto ya tenemos hecho el entrenamiento de nuestro modelo, mediante la función “predict” le pasaremos al modelo los datos con los que queremos predecir la categoría de crimen.

Para saber como de correcto son los resultados de predicción que arroja el modelo, por un lado vamos a sacar el “scorer”, la puntuación de acierto que se ha obtenido, y por otro lado vamos a comparar los resultados que nos ha devuelto esta predicción con los datos que son los correctos.

```
prediction = clf.predict(X_test)

print clf.score(X_test, Y_test)
pd.DataFrame({"SVM Prediction": prediction, "Real Prediction": Y_test})
```

---

Scorer: 0.1841

	Real Prediction	SVM Prediction
85117	WARRANTS	LARCENY/THEFT
56806	ASSAULT	MISSING PERSON
45585	WARRANTS	MISSING PERSON
56607	VANDALISM	LARCENY/THEFT
70302	LARCENY/THEFT	LARCENY/THEFT
71658	FRAUD	MISSING PERSON
18517	LARCENY/THEFT	LARCENY/THEFT

Los resultados que hemos obtenido son bastante malos, teniendo en cuenta que nos devuelve una tasa de acierto del 18% y que en un primera vista de comparación entre los datos reales “Real Prediction” y los datos que ha generado “SVM Prediction” apenas se da alguna coincidencia.

Con estas pruebas hemos comprobado que este modelo de predicción no es válido para nuestro conjunto de datos.

## B. KMN (K-Nearest-Neighbour)

Una vez que hemos probado el un modelo SVM y no hemos obtenido buenos resultados, el siguiente en la lista es el algoritmo KNN, o algoritmo de agrupación.

K-Nearest Neighbor Example

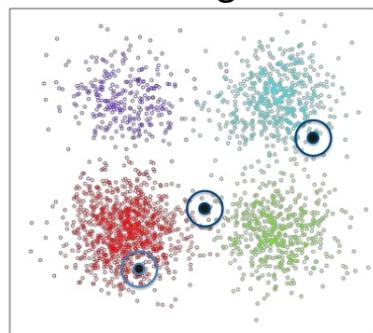


Ilustración 2-3. Ejemplo del modelo KMN.

La idea básica de este modelo se basa en que teniendo un conjunto de puntos, los cuales están agrupados en función de la cercanía con otros puntos, en el momento que llegue un nuevo punto este se va a clasificar en función de la clase mas frecuente a la que pertenecen sus  $K$  vecinos mas cercanos.

Para aplicar este modelo, tenemos que importar “KNeighborsClassifier” de la librería “sklearn”.

Un vez que lo tenemos importado, para entrenar este modelo mediante el método “fit” tenemos que pasarle como parámetro el número de “grupos” que queremos que se formen para agrupar los datos.

Nosotros hemos dejado el modelo corriendo dentro de un bucle for con un rango de 1 a 50 grupos y que vaya saltando de 2 en 2 para ver que indice de acierto se iba generando.

```
from sklearn.neighbors import KNeighborsClassifier
for i in range(1, 50, 5):
    print "Neighbords: " + str(i)
    knn = KNeighborsClassifier(n_neighbors=i)
    knn.fit(X_train, Y_train)
    print knn.score(X_train,Y_train)
```

Neighbords: 1	
0.6237	
Neighbords: 6	
0.40855	
Neighbords: 11	Neighbords: 31
0.364783333333	0.317733333333
Neighbords: 16	Neighbords: 36
0.3432	0.3118
Neighbords: 21	Neighbords: 41
0.332	0.30915
Neighbords: 26	Neighbords: 46
0.323383333333	0.3062

Para las agrupaciones que hemos puesto, vemos que ha medida que se va aumentando el numero de grupos, la precisión del modelo va cayendo.

Teniendo en cuenta que como máximo nos ha una precisión del 36%, descartamos los dos primero resultados puesto que con un numero tan pequeño de grupos la precision sea mayor, y como mínimo una precisión del 30%, nos quedamos con una media de grupos para aplicarlo sobre el conjunto de test.

```
knn = KNeighborsClassifier(n_neighbors=21)
knn.fit(X_train, Y_train)
outputKnn = knn.predict(X_test)

dfPrediction = pd.DataFrame({'Real predicton': Y_test , 'KNN Prediction': outputKnn})
```

<b>97705</b>	LARCENY/THEFT	LARCENY/THEFT
<b>36235</b>	MISSING PERSON	MISSING PERSON
<b>11411</b>	LARCENY/THEFT	LARCENY/THEFT
<b>82341</b>	LARCENY/THEFT	LARCENY/THEFT
<b>66946</b>	OTHER OFFENSES	OTHER OFFENSES
<b>9253</b>	WARRANTS	FRAUD
<b>69161</b>	OTHER OFFENSES	LARCENY/THEFT
<b>76139</b>	OTHER OFFENSES	LARCENY/THEFT
<b>66395</b>	OTHER OFFENSES	ASSAULT

Aunque vemos que con este modelo hemos conseguido subir el índice de acierto casi al 30%, y vemos que los resultados, a simple vista, se ve un mayor acierto de los datos. Seguimos sin tener un modelo válido para nuestro conjunto de datos.

## C. Random Forest Classifier

Ya hemos probado las alternativas de SVM y KNN para procesar nuestro dataset, siguiendo el esquema que marcamos arriba, el último que nos queda probar, es un clasificador de tipo “Ensemble Classifier” del cual hemos tomado el “Random Forest Classifier”.

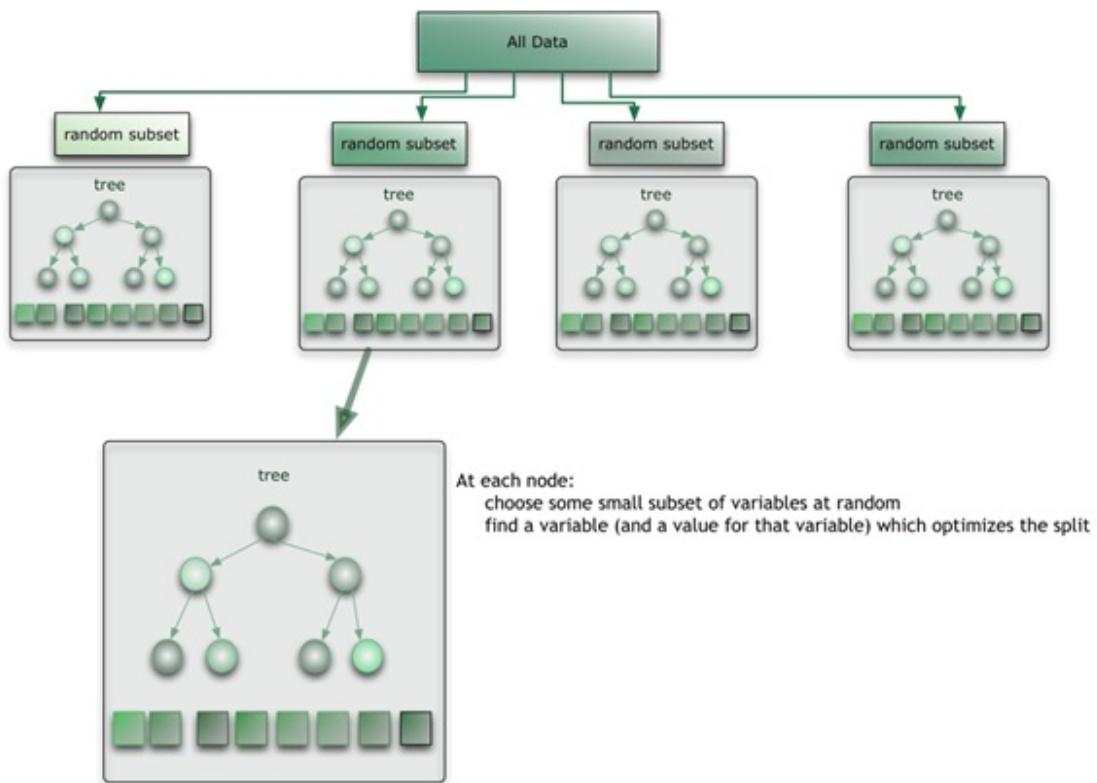


Ilustración 2-4. Ejemplo del modelo Random Forest.

Este modelo se basa principalmente en árboles de decisión, con una única entrada de datos, estos se dividen en subgrupos de manera aleatoria, construyendo así sub-árboles de decisión diferentes a los demás. A media que estos sub-árboles van creciendo en profundidad, se van escogiendo mas variables al azar para cada nodo de árbol que se crea nuevo.

La predicción de este modelo se basa principalmente en la “mayoría común” donde se clasificara como positiva la predicción si la mayoría de árboles coinciden en ella.

Para aplicar este modelo tenemos que importar de la librería “sklearn” el modulo de “RandomForestClassifier”.

Características de Random Forest:

- No es superable en la precisión, de entre los algoritmos actuales.
- Funciona de manera eficiente en grandes bases de datos.
- Puede manejar miles de variables de entrada sin borrado de variables.
- Aporta estimaciones de qué variables son importantes en la clasificación.
- Se genera una estimación objetiva interna de la generalización de error a medida que avanza la construcción del bosque.
- Tiene un método eficaz para la estimación de los datos faltantes y mantiene la precisión cuando una gran parte de los datos que faltan.
- Tiene métodos para error de equilibrio en la población de conjuntos de datos no balanceados de clase.
- Los bosques generados se pueden guardar para uso futuro en otros datos.
- Los prototipos que calcula dan información acerca de la relación entre las variables y la clasificación.
- Se calcula proximidades entre pares de casos que se pueden utilizar en la agrupación, la localización de los valores atípicos, o (por escala) dar interesantes vistas de los datos.
- Las características anteriores se pueden extender a los datos no etiquetados, que conducen a agrupamiento no supervisado, vistas de datos y la detección de valores atípicos.
- Ofrece un método experimental para la detección de interacciones de variables.

Los árboles y los bosques. El Random Forest comienza con una técnica de aprendizaje automático estándar llamada "árbol de decisiones", que, en cuanto al conjunto, corresponde a un aprendizaje. En un árbol de decisión, una entrada se introduce en la parte superior y hacia abajo a medida que atraviesa el árbol de los datos se acumulan en conjuntos más pequeños y más pequeños(sobra).

A este modelo tenemos que pasarle el número de "árboles" que queremos que cree en la estimación de los datos, en nuestro caso para quitarnos los problemas de los mínimos locales que podemos encontrar debido a la disparidad de los datos hemos puesto un máximo de 100 árboles.

Para realizar el entrenamiento también se realiza mediante el método "fit".

```
from sklearn.ensemble import RandomForestClassifier  
rfcl = RandomForestClassifier(n_estimators=100)  
rfcl.fit(X_train, Y_train)  
  
Score: 0.866885714286  
  
prediction = rfcl.predict(X_test)  
  
pdPrecision = pd.DataFrame({"Real Prediction": Y_test, "RF Prediction" : prediction})
```

## 2. CONOCIMIENTO Y VISUALIZACIÓN DE LOS RESULTADOS

---

<b>36403</b>	WARRANTS	WARRANTS
<b>9582</b>	NON-CRIMINAL	NON-CRIMINAL
<b>49640</b>	OTHER OFFENSES	NON-CRIMINAL
<b>15391</b>	LARCENY/THEFT	LARCENY/THEFT
<b>43744</b>	VEHICLE THEFT	VEHICLE THEFT
<b>2005</b>	LARCENY/THEFT	OTHER OFFENSES
<b>2253</b>	PROSTITUTION	PROSTITUTION
<b>97587</b>	LARCENY/THEFT	LARCENY/THEFT

A la vista de estos resultados, y obteniendo un 86% de acierto de los datos, este es el modelo que mejor resultado nos ha arrojado, y es el que utilizamos para subir a la plataforma de la competición de Kaggle.

Ejemplo del resultado Submmisión subido a la competición de Kaggle:

**San Francisco Crime Classification**

5 entries in team [UpsaTeam](#)

Closed - Validating Final Results  
**1772nd/2335**

```

Cargando datos entrenamiento...
Convirtiendo el entrenamiento a una estructura categorica...
train cleaning...
Cargando datos de test...
Convirtiendo el test a una estructura categorica...
Limpieza de test...
running classifier...
Predictor Error...
0.278518624815
Resultados...

      ARSON   ASSAULT  BAD CHECKS  BRIBERY  BURGLARY  DISORDERLY CONDUCT \
Id
0       0  0.103333          0       0  0.010000                  0
1       0  0.073333          0       0  0.000000                  0
2       0  0.096667          0       0  0.013333                  0
3       0  0.030000          0       0  0.000000                  0
4       0  0.030000          0       0  0.000000                  0

      DRIVING UNDER THE INFLUENCE  DRUG/NARCOTIC  DRUNKENNESS  EMBEZZLEMENT \
Id
0                 0.01           0.000         0.00                  0
1                 0.00           0.060         0.00                  0

```

## PREDICCIÓN DE LA CATEGORÍA DE CRÍMENES OCURRIDOS EN SAN FRANCISCO

---

```
2           0.00      0.014      0.00      0
3           0.00      0.000      0.01      0
4           0.00      0.000      0.01      0

...       SEX OFFENSES NON FORCIBLE STOLEN PROPERTY SUICIDE \
Id
0       ...
1       ...
2       ...
3       ...
4       ...

SUSPICIOUS OCC  TREA  TRESPASS  VANDALISM  VEHICLE  THEFT  WARRANTS \
Id
0       0.030000    0     0.00     0.051667    0.145000   0.00
1       0.030000    0     0.00     0.033333    0.080000   0.04
2       0.070000    0     0.03     0.052000    0.030333   0.02
3       0.033333    0     0.00     0.010000    0.300000   0.01
4       0.033333    0     0.00     0.010000    0.300000   0.01

WEAPON LAWS
Id
0       0.20
1       0.26
2       0.00
3       0.01
4       0.01

[ 5 rows x 39 columns ]
```

# 3. Conocimiento y Visualización de los resultados

En la parte de conocimiento y visualización de los datos hemos recogido una tabla comparativa de los modelos utilizados en el estudio.

Algoritmo Aplicado	Sofware	Variables para predicción	Train Partition % acierto	Test Partition % acierto	Número de datos train	Número de datos test
Modelo Linear SVM	Python		0,083	0,084	70.000	30.000
KNN	Python		0,33	0,27	70.000	30.000
RandomForest	Python	X, Y, hours, months, years, district, day_of_week	0,86	0,52	70.000	30.000

A la vista de los diferentes datos que arrojan los modelos que se han aplicado, se puede ver que:

- Error Modelo SVM: 0,917 (Train) - 0,916 (Test)
- Error KNN: 0,67 (Train) - 0,73 (Test)
- Error de RandomForest: 0,14 (Train) - 0,52 (Test)

Como se puede ver el modelo que mas error tiene es el SVM, debido seguramente a que la creación de hiperplanos para separar los diferentes grupos se ha visto dificultada por la gran diversidad de datos que se encuentra en el conjunto. Respecto al KNN podemos decir que la separación por cercanía, al igual que el SVM, se ha visto dificultada porque los datos no estaban

lo suficientemente separados como para establecer grupos claros formando patrones. Y por ultimo el Random Forest ha sido el que mejor resultado hemos obtenido debido, a que a diferencia de los otros dos modelos este ha sido capaz de “desviarse” de los mínimos locales poniendo un número alto de árboles en el modelo para despreciarlos. Así los datos que estaban alejados de los demás han podido ser despreciados con mayor facilidad

### 3.1 Técnicas de visualización

Los datos por sí mismos, que consisten de bits y bytes almacenados en un archivo en el disco rígido de una computadora, son invisibles. Para poder verlos y encontrarles sentido, necesitamos visualizarlos.

Hemos optado para la visualización de los datos Tableau ya que es una herramienta de visualización de datos interactiva, es decir, el usuario tiene la posibilidad de interactuar con los datos: comparar, filtrar, conectar unas variables con otras... Además, la plataforma y los paneles que se pueden crear con la herramienta son muy visuales (facilita la comprensión rápida de los datos). También tiene algunas ventajas interesantes cuando manejamos bases de datos: acepta formatos con Excel, Access y texto; y podemos acceder a muchas bases de datos comunes como Microsoft SQL Server, MySQL, Oracle o Greenplum; y también tienes la posibilidad de usar la API de Tableau para la extracción sistemática de datos.

Nuestra publicación de Dashboard Tableau:

[https://public.tableau.com/profile/publish/train\\_1/Dashboard1 - !/publish-confirm](https://public.tableau.com/profile/publish/train_1/Dashboard1 - !/publish-confirm)

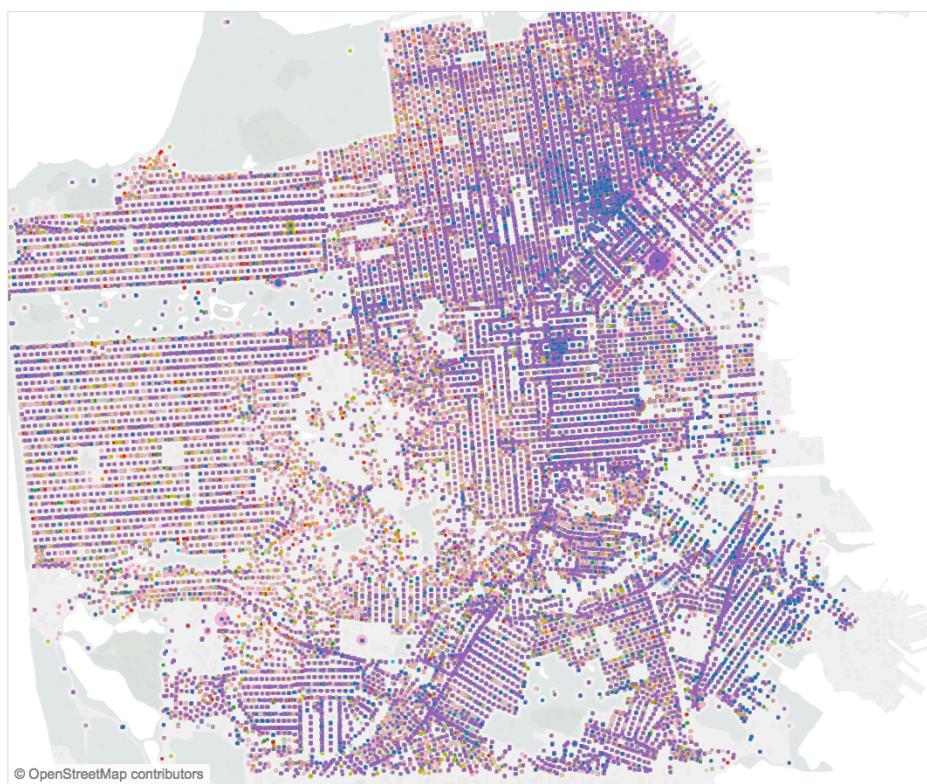
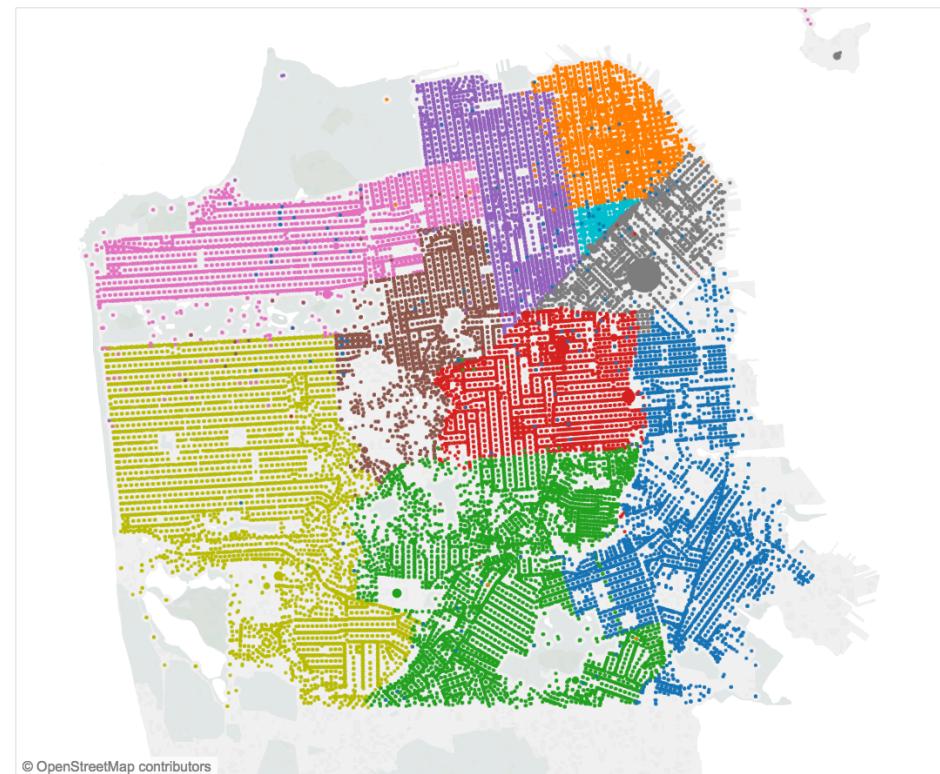
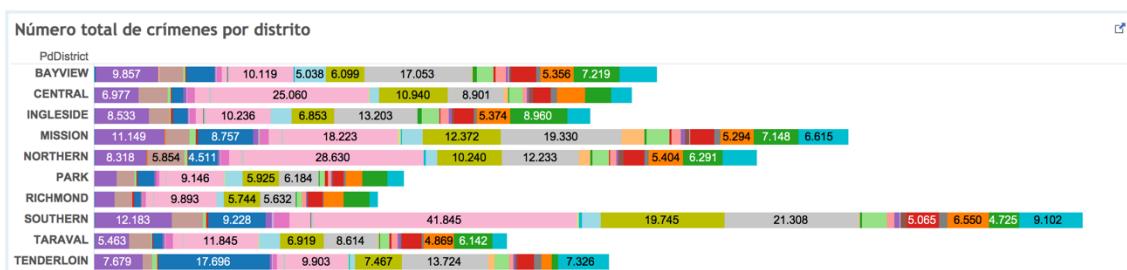


Ilustración 3-1. Distribución de crímenes por categoría.



**Ilustración 3-2. Representación de los distintos distritos diferenciados por colores.**



**Ilustración 3-3. El total de número de crímenes por cada distrito diferenciado.**

## PREDICCIÓN DE LA CATEGORÍA DE CRÍMENES OCURRÍDOS EN SAN FRANCISCO

---

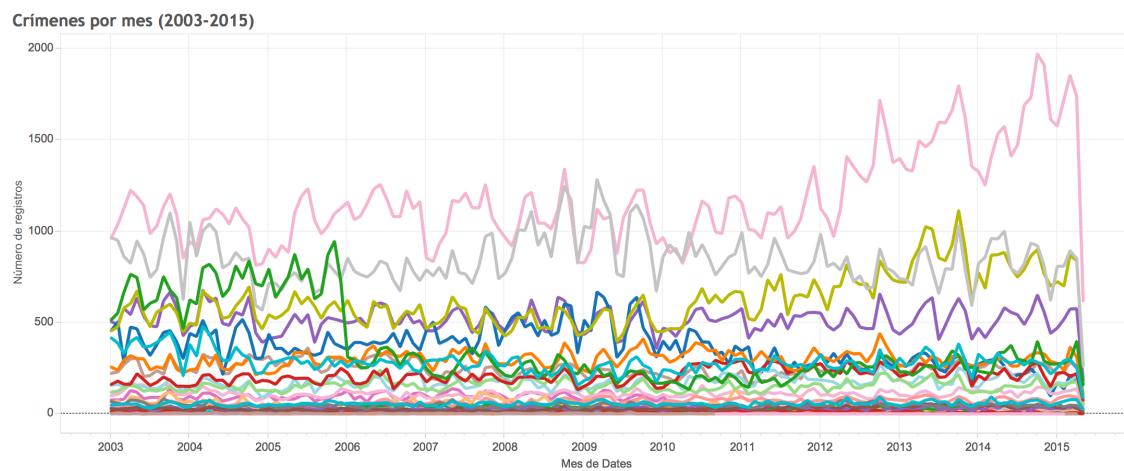


Ilustración 3-4. Representación gráfica de los crímenes sufridos por mes desde 2003 hasta 2015.

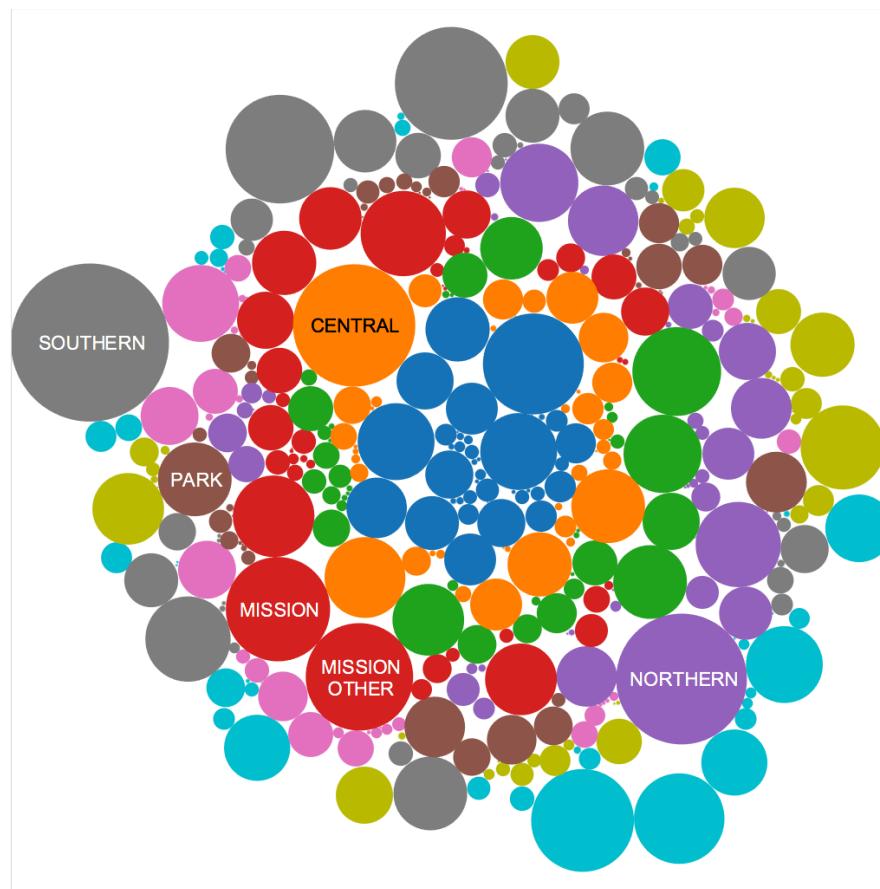


Ilustración 3-5. Incidencia de crímenes por distrito en la ciudad de San Francisco.

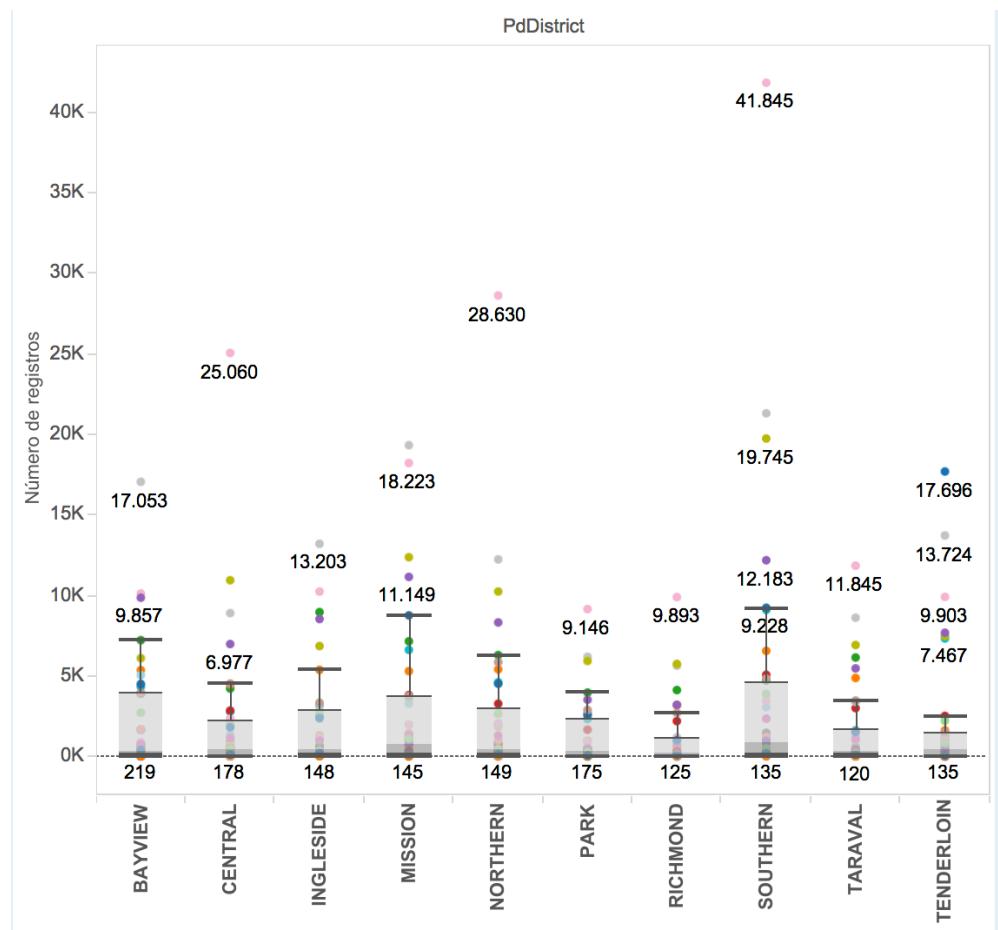


Ilustración 3-6. Incidencia de crímenes por distrito.



Ilustración 3-7. Árbol de categorías para cada tipo de crímenes.

## 3.2 Técnicas de representación con Python

En cuanto a Python, a continuación se muestra las representaciones de los datos mediante esta técnica.

```

          Category           Descript DayOfWeek PdDistrict \
count      878049            878049  878049   878049
unique     39                  879      7       10
top    LARCENY/THEFT  GRAND THEFT FROM LOCKED AUTO  Friday  SOUTHERN
freq     174900             60022  133734  157182

          Resolution          Address
count      878049            878049
unique     17                  23228
top        NONE   800 Block of BRYANT ST
freq     526790             26533

```

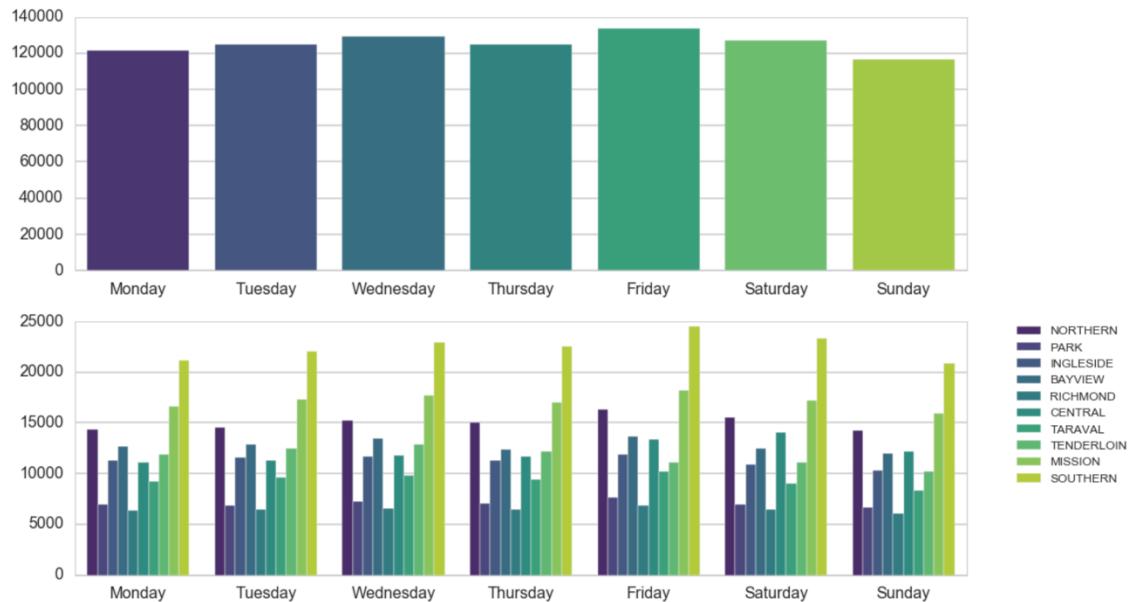
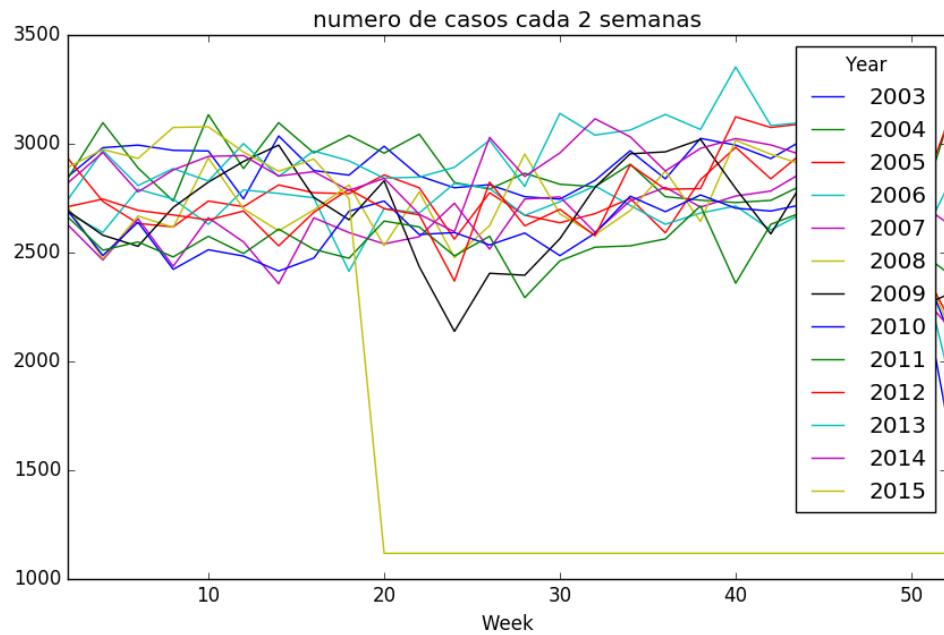
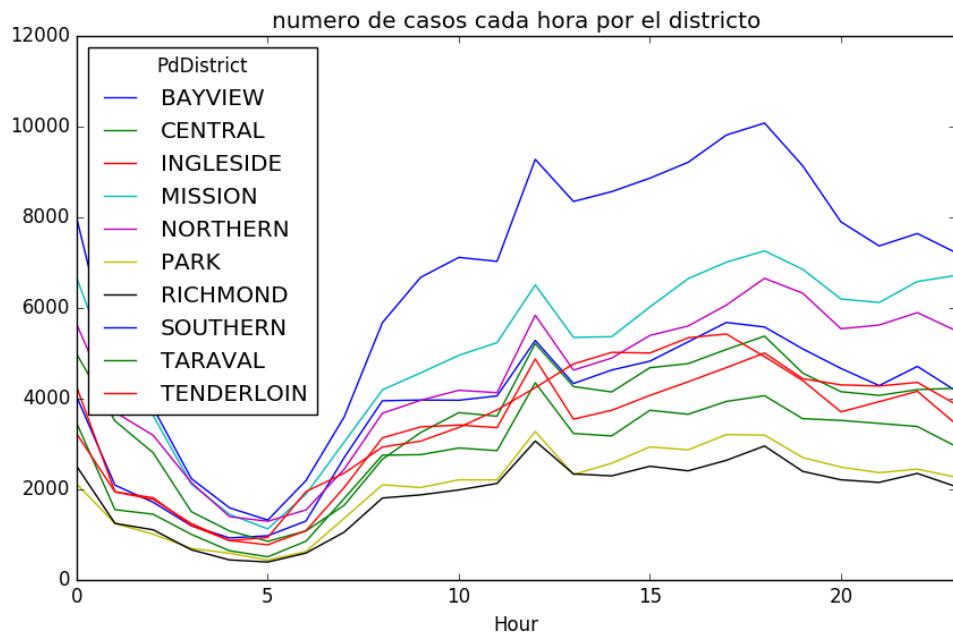


Ilustración 3-8. Cantidad de crímenes por día de la semana y por distrito.



**Ilustración 3-9. Series temporales cada dos semanas y para cada año.**



**Ilustración 3-10. Número de casos de crímenes por hora en cada uno de los distritos estudiados.**

## PREDICCIÓN DE LA CATEGORÍA DE CRÍMENES OCURRIDOS EN SAN FRANCISCO

---



**Ilustración 3-11. Representación gráfica de las zonas con mayor incidencia de crímenes por zona.**

# 4. Conclusiones

---

Tras el trabajo realizado aquí plasmado, podemos decir que hemos superado con creces los objetivos que se plantearon a la hora de empezar este trabajo de fin de experto.

La prueba que se nos planteó con la competición de “kaggle” ha sido para nosotros un último esfuerzo que teníamos que realizar para poner en práctica todos los conocimientos que hemos ido adquiriendo a lo largo de este pequeño curso.

Cuando recibimos los datos de la competición lo primero que realizamos fue un análisis exploratorio, como nos habían indicado, pero en realidad no era tan sencillo como esperábamos, había un gran número de variables a tener en cuenta, como los datos que nos podíamos encontrar con el campo vacío, o como muchos de ellos despuntaban en las gráficas porque se salían de los límites establecidos.

Una vez que ya nos hicimos con los datos con los que estábamos trabajando teníamos que escoger qué tipo de modelo se debía aplicar para realizar la mejor predicción posible, por más que probamos modelos ninguno nos daba más allá de un 30% de acierto, hasta que dimos con el idóneo que fue el que aplicamos para la competición en la que estábamos participando.

Este trabajo no ha sido fácil ni sencillo para nosotros, pues nunca nos habíamos puesto en la situación de tener que aplicar estas herramientas en un campo meramente práctico, se podría decir que cerca de los problemas reales, por eso creemos que con los resultados que hemos obtenido y aquí presentamos hemos superado la prueba planteada.



# 5. Bibliografía

---

1. **Forum Kaggle.** <https://www.kaggle.com/maite828/forum>
2. **RandomForest.** <http://randomforest2013.blogspot.com.es/2013/05/randomforest-definicion-random-forests.html>
3. **Bioinformatics at COMAV.**  
<https://bioinf.comav.upv.es/courses/linux/python/pandas.html>
4. **UAM+CSIC.** [http://www.uam.es/personal\\_pdi/ciencias/jspinill/CFCUAM2014/RF\\_BRT-CFCUAM2014.html](http://www.uam.es/personal_pdi/ciencias/jspinill/CFCUAM2014/RF_BRT-CFCUAM2014.html)
5. **Interactive visualización Tableau.** <http://www.tableau.com/es-es/stories/topic/data-visualization>
6. **Machine Learning en Español.** <http://scikit-learn.org/stable/>