Alma Mater Studiorum - Università di Bologna

BDA project- Presentation

San Francisco Crime Classification

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
Artificial Intelligence

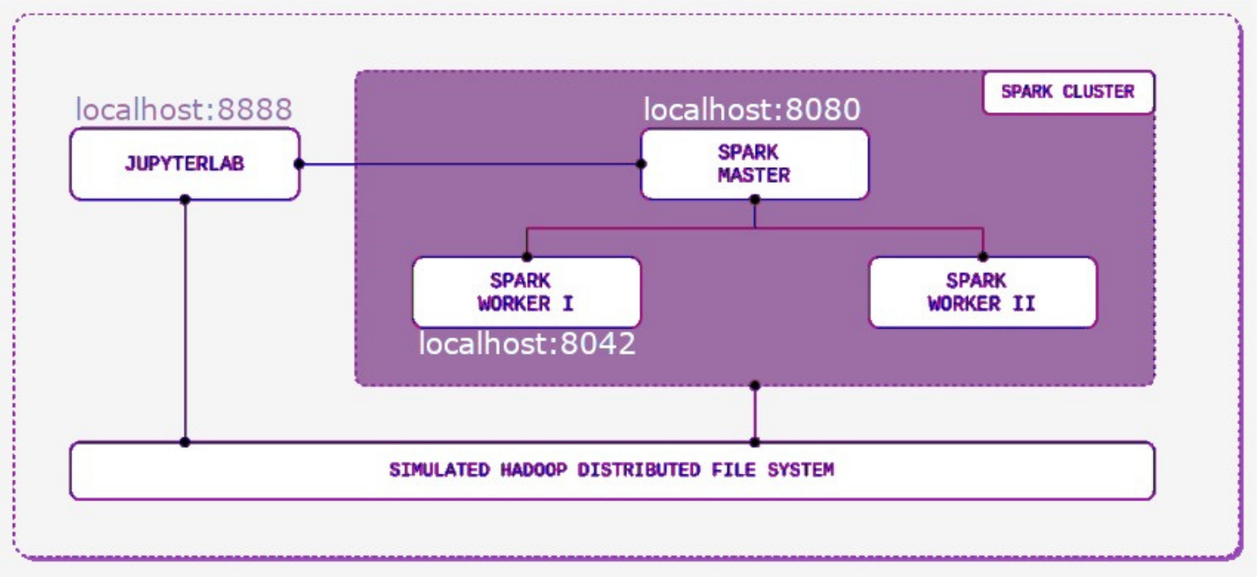
Student
Alessio Bonacchi

Academic year 2022-2023
May

Introduction

Cluster architecture

- Apache Spark cluster is built upon Docker
- We have 2 Spark worker nodes with 2 cores and 5 GB memory by default
- Then the Spark session is created on Jupyter Notebook



Task

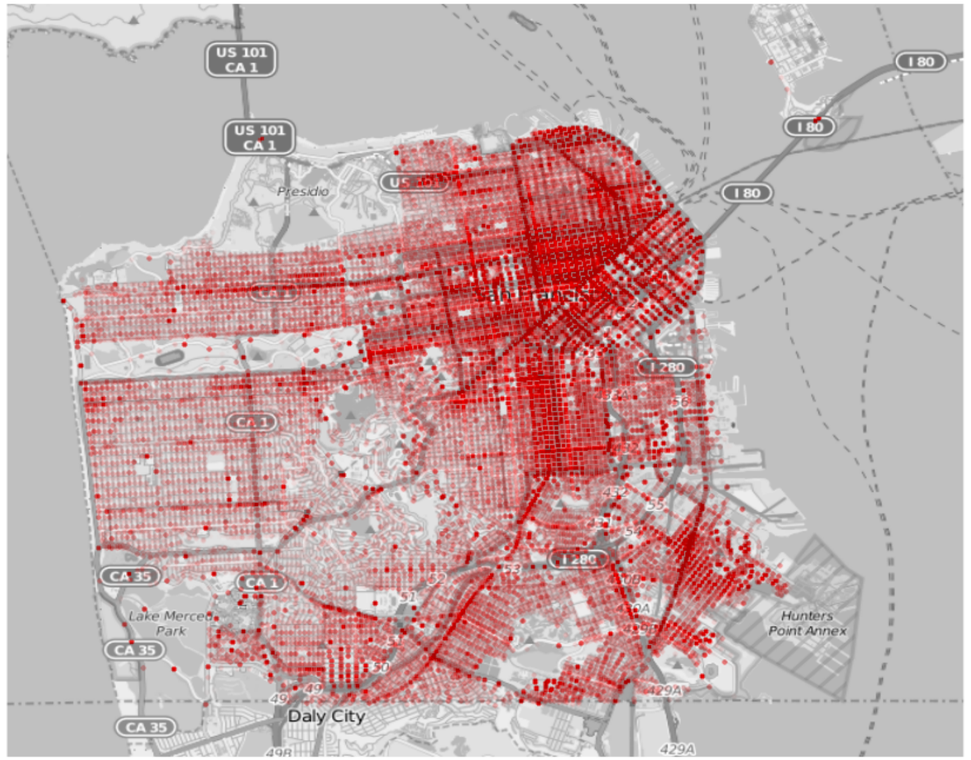
Multi-class classification:

Given a set of features, predict the category of the crime associated.

Categories values are robbery, drunkness, vehicle theft...
39 in total

Features comprehend:

- **dayofweek:** day of crime
- **month:** **month** of crime
- **year:** year of crime
- **description:** description of the crime
- **resolution:** how the crime has been solved
- **pddistrict:** the district in which the crime happened



Dataset

- I am using the dataset available on **Kaggle**
- Originally the dataset contained other features like the date, X and Y (longitude and latitude) coordinates

Then i decided to drop the coordinates since the **pddistrict** features already provided a geographical information and I decided to derive **month** and **year** features from **date**

Dates	Category	Descript	DayOfWeek	PdDistrict	Resolution	Address	X	Y
2015-05-13 23:53:00	WARRANTS	WARRANT ARREST	Wednesday	NORTHERN	ARREST, BOOKED	OAK ST / LAGUNA ST	-122.425891675136	37.7745985956747
2015-05-13 23:53:00	OTHER OFFENSES	TRAFFIC VIOLATION...	Wednesday	NORTHERN	ARREST, BOOKED	OAK ST / LAGUNA ST	-122.425891675136	37.7745985956747
2015-05-13 23:33:00	OTHER OFFENSES	TRAFFIC VIOLATION...	Wednesday	NORTHERN	ARREST, BOOKED	VANNESS AV / GREE...	-122.42436302145	37.8004143219856
2015-05-13 23:30:00	LARCENY/THEFT	GRAND THEFT FROM ...	Wednesday	NORTHERN	NONE	1500 Block of LOM...	-122.42699532676599	37.80087263276921
2015-05-13 23:30:00	LARCENY/THEFT	GRAND THEFT FROM ...	Wednesday	PARK	NONE	100 Block of BROD...	-122.438737622757	37.771541172057795

At the end it looked like this!

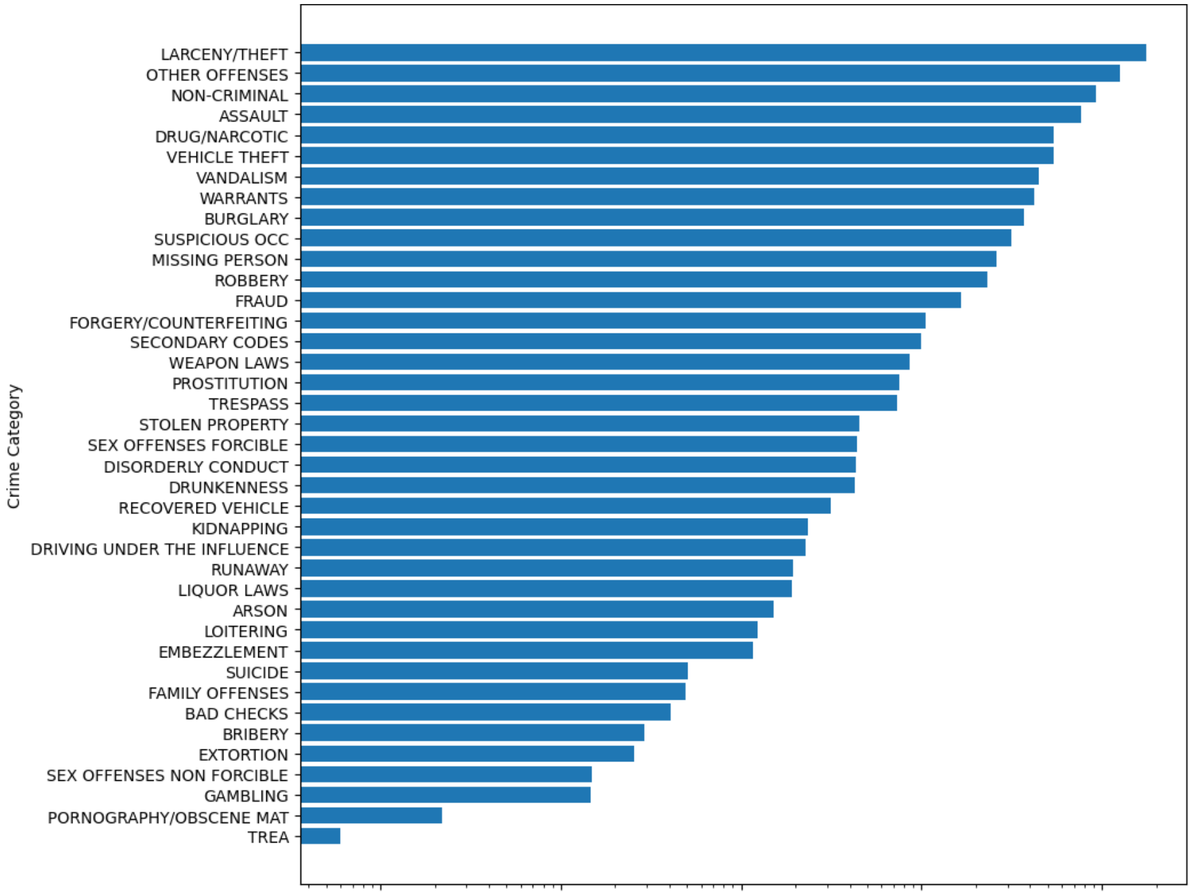
Category	Descript	DayOfWeek	PdDistrict	Resolution	month	year
WARRANTS	WARRANT ARREST	Wednesday	NORTHERN	ARREST, BOOKED	5	2015
OTHER OFFENSES	TRAFFIC VIOLATION...	Wednesday	NORTHERN	ARREST, BOOKED	5	2015
OTHER OFFENSES	TRAFFIC VIOLATION...	Wednesday	NORTHERN	ARREST, BOOKED	5	2015
LARCENY/THEFT	GRAND THEFT FROM ...	Wednesday	NORTHERN	NONE	5	2015
LARCENY/THEFT	GRAND THEFT FROM ...	Wednesday	PARK	NONE	5	2015

Data overview

- **878049** data points in total
- **80%** data points in *train* split
- **20%** data points in *test* split

High imbalanced class ratio

The number of instances for each class is not uniform, resulting in classes having a very high support and classes having a very low one.



Feature Processing

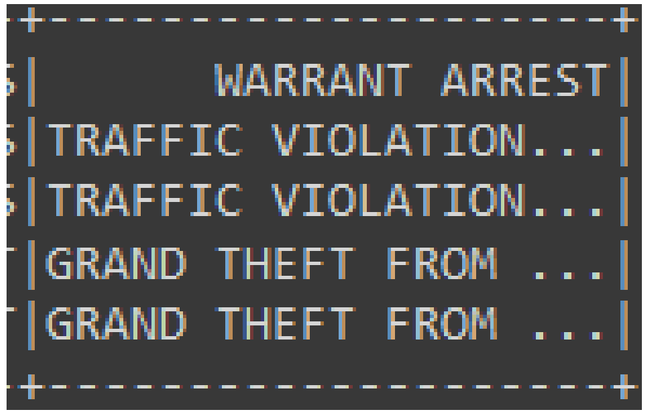
Vectorization of 'Descript' feature

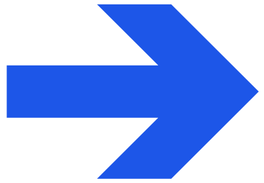
Since 'Descript' values are of type string I transformed them using this pipeline:

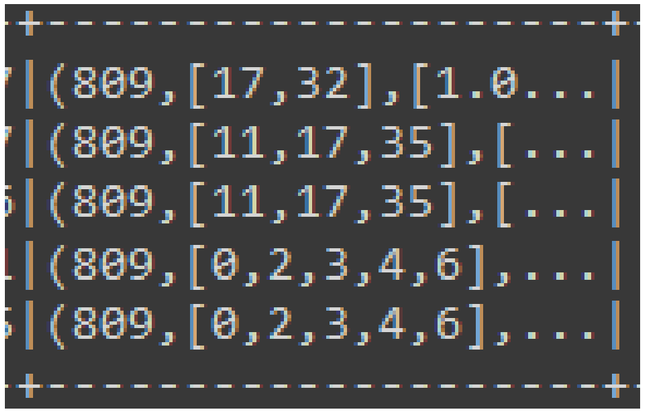
1. **RegexTokenizer:** initial tokenization of words
2. **Stop words removal:** I remove a set of stopwords that appear in the value

3a. **CountVectorization:** transform tokens into counvectors

3b. **Tf-Idf vectorization:** transform tokens into tf-idf vectors





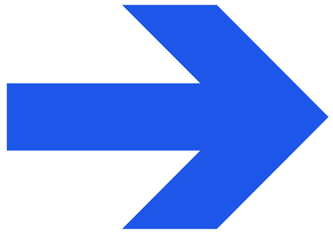


Indexing of categorical features

The remaining features are categorical (dayofweek,month,year,pddistrict,resolution).
Then what I have done is to pass them to an indexer, in this case StringIndexer from MLLib

Here we have an example of the transformation of column pddistrict

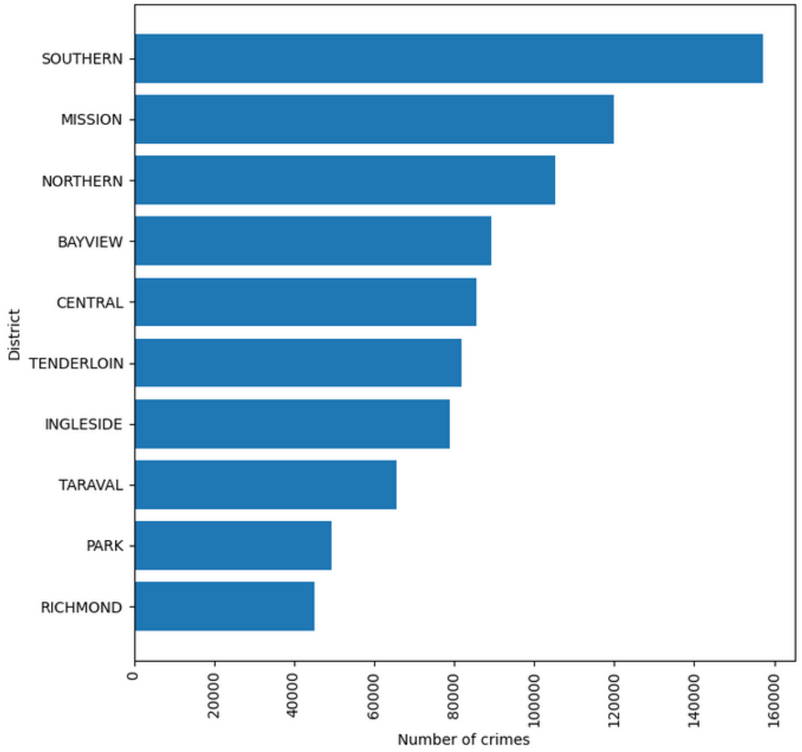
PdDistrict
NORTHERN
NORTHERN
NORTHERN
NORTHERN
PARK

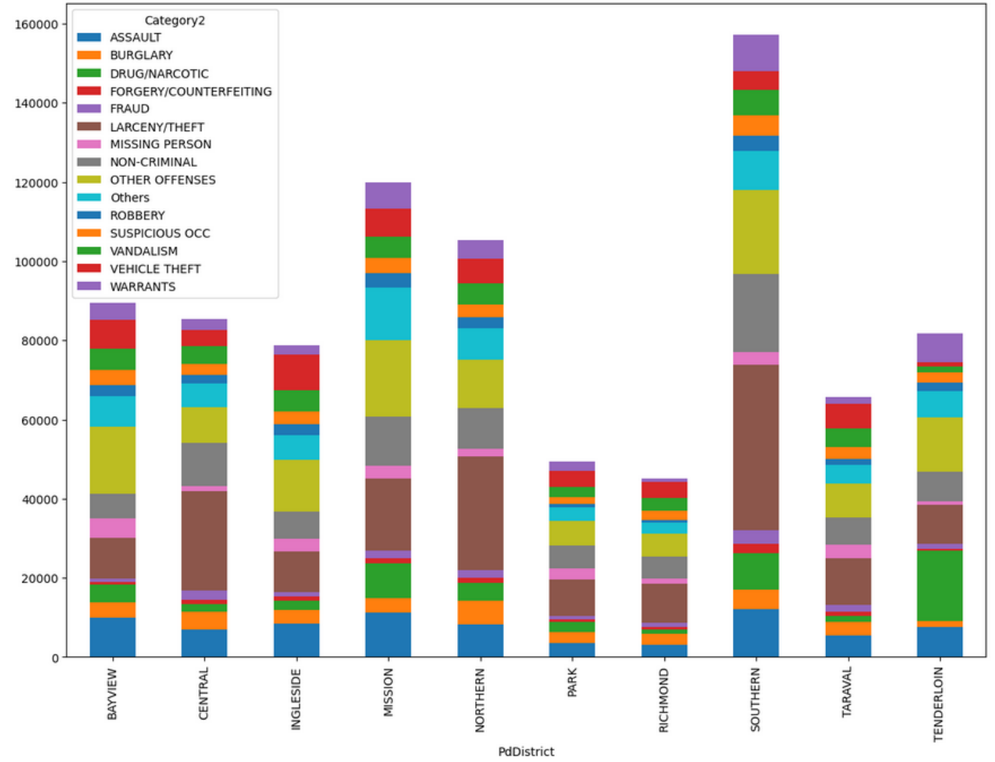


pddistrict
2.0
2.0
2.0
2.0
8.0

Data Analysis

Distribution of crimes among districts



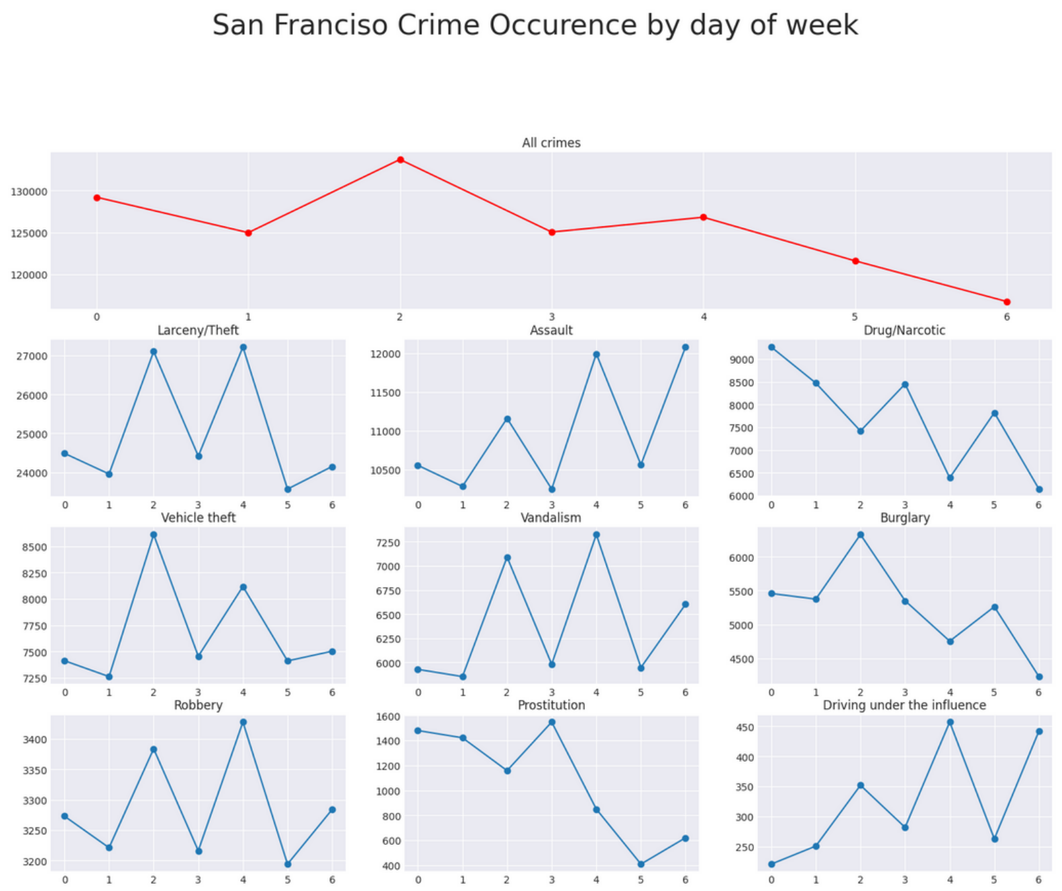


Distribution of crimes among days

In red we have the distribution of all crimes while in the other plots we can see the distribution among days for evey category

Severe drop of crime occurrence in the weekends

The activities for some of these, intuitively, grows in the weekend, for example 'driving under the influence of drugs' and 'assault'.



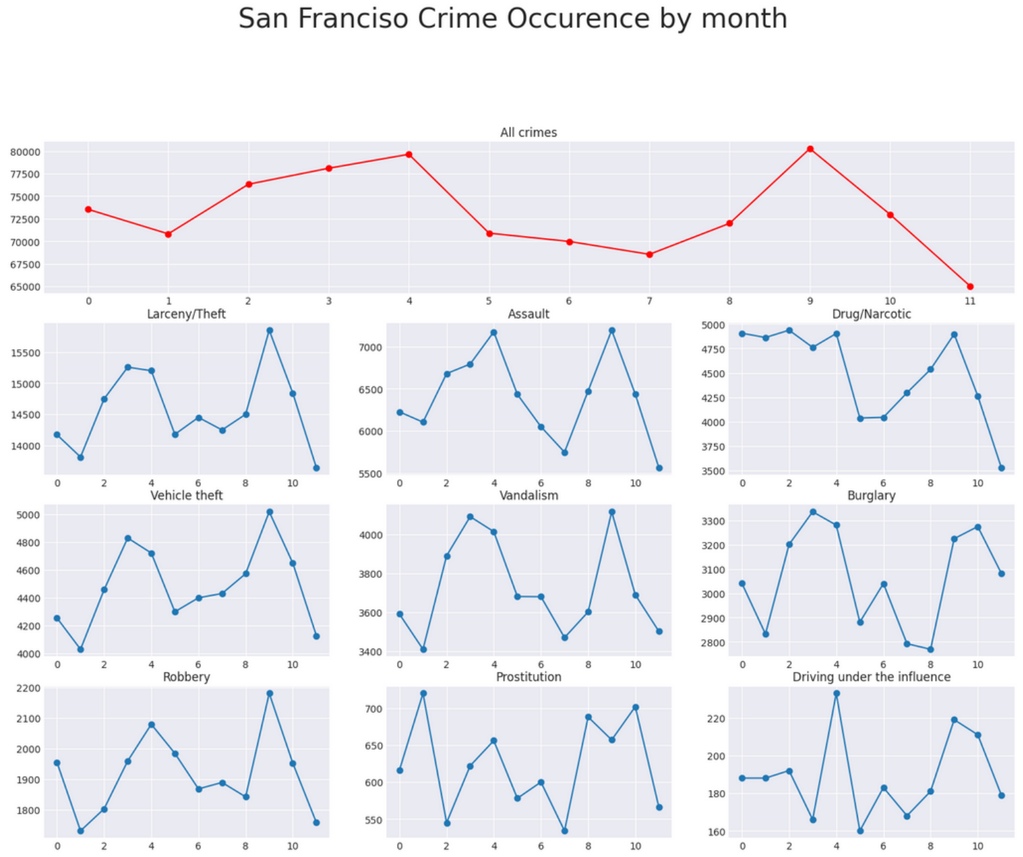
Distribution of crimes among months

In red we have the distribution of all crimes while in the other plots we can see the distribution among days for evey category

We can see that the most active months are May and October

Crimes seem to drop in December....

....We are all better at Christmas, even criminals!

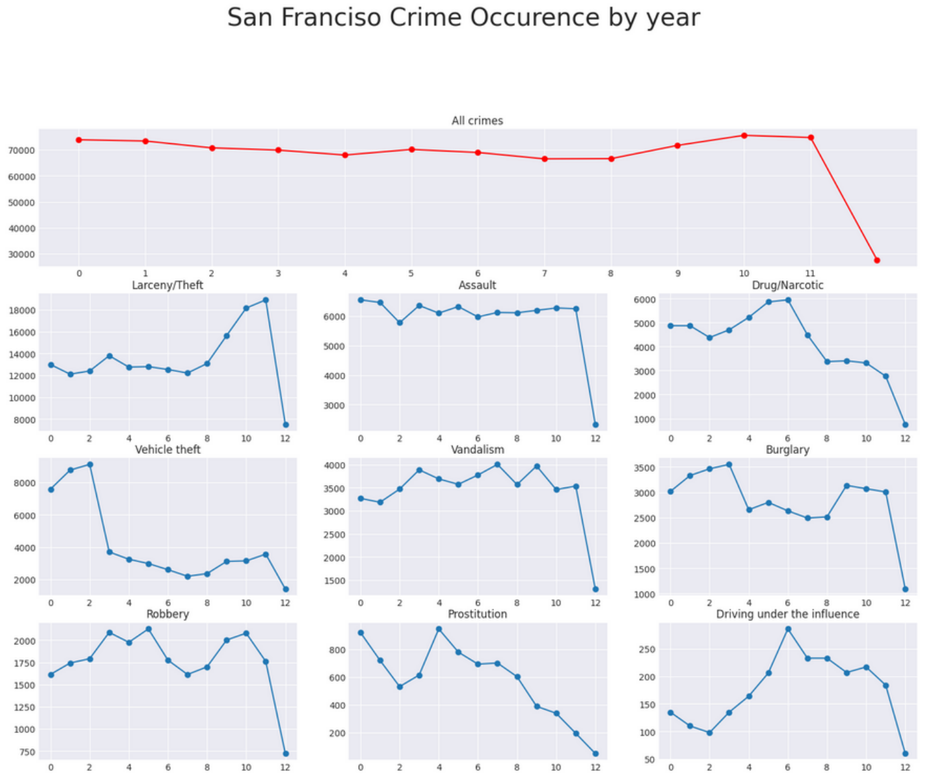


Distribution of crimes among years

In red we have the distribution of all crimes while in the other plots we can see the distribution among days for evey category

The reason for the drop in number of crimes in 2015 is that the collection of data stops in May, so it does not cover the rest of the year

I was undecided whether to remove all instances referring to 2015, but as we can see later the models give strong results even mantaining it



Modeling

Feature preparation

I used models implemented in MLLib, then it is necessary to assemble all the features in a unique 'features' column.
This can be done with VectorAssembler, a function available in MLLib that takes in inpus a list of input_cols and return an output_col properly assembled

Train/Test split

The splits are randomly generated using the function randomSplit with seed = 100, available in MLLib.
At the end we are left with 614485 samples for training set and 263564 for test set

Evaluation of results

I created a custom function evaluation_custom in order to display the results of each classification model. It takes the predictions generated by the model and then display the macro_avg F1 score, which is our reference metric since it a multiclass context.
It also display the precision and the recall

Cross Validation

I used MLLib CrossValidator in order to evaluate model robustness.
In particular I use a 5-fold cross validator which generates 5 sets of training/testing pairs.

Naive Bayes with smoothing as paramGrid

Selected Models

- Logistic Regression with CountVector features
- Logistic Regression with TF-IDF features
- Random Forest
- Naive Bayes
- BERT large
- XLNet base
- DistilBERT
- XLNet large

Logistic Regression

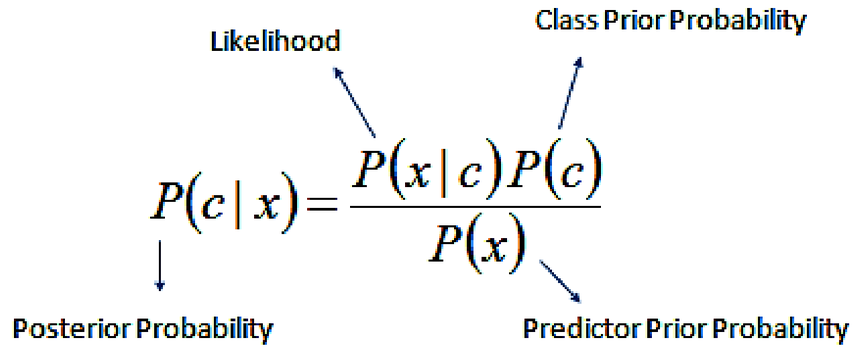
- One of the easiest ML algorithm
- Supervised learning
- Predict the categorical dependent variable using a given set of independent variables
- The outcome must be a categorical or discrete value, but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1

CountVector features: bag-of-words style approach,very easy, simply counts the number of times a word appears in a document

TF-IDF features: it gives importance both to the frequency with which the word appears in the document and the inverse appearency frequency of the word in the whole corpus, more refined technique for string vectorization

Naive Bayes

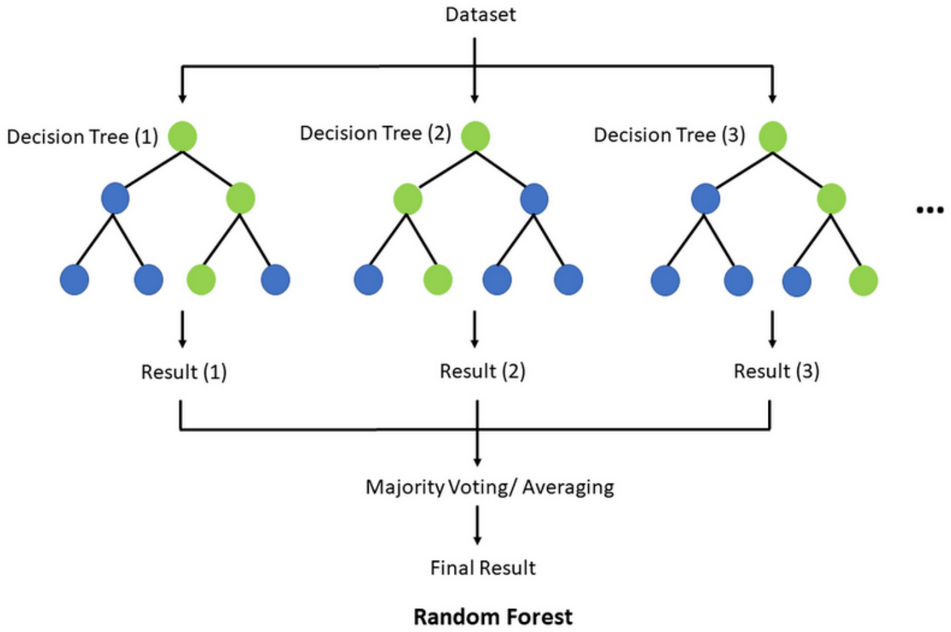
- Utilizes the principles of Bayesian probability to make predictions
- Supervised learning
- "Naive" because it makes a strong assumption of independence among the features in the dataset
- It calculates the probability of a certain outcome given a set of input features by combining prior knowledge with observed evidence
- During prediction, Naive Bayes calculates the posterior probability of each class label given the input features using Bayes' theorem and selects the label with the highest probability.
- Smoothing handles the problem of zero probability in Naïve Bayes



$$P(c \mid \mathbf{X}) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

Random Forest

- Combines the strengths of multiple decision trees to make accurate predictions
- Supervised learning
- It creates an ensemble of decision trees and combines their results to produce a final prediction
- Works by constructing a multitude of decision trees, each trained on a random subset of the training data and using a random subset of the features
- This randomness introduces diversity and usually helps preventing overfitting



numTrees=100 **maxDepth=4** **maxBins=32**

Results

MODEL	Macro F1	Precision	Recall
LR CV	0.983	0.948	0.999
LR TF-IDF	0.982	0.945	0.999
Naive Bayes	0.996	0.986	0.999
Random Forest	0.745	0.378	0.999

Logistic regression and Naive Bayes model performs extremely good

Random Forest seems not to fit the problem

This can be explained since it's no mystery that for high-dimensional sparse data random forest is not the best choice.

Error Analysis

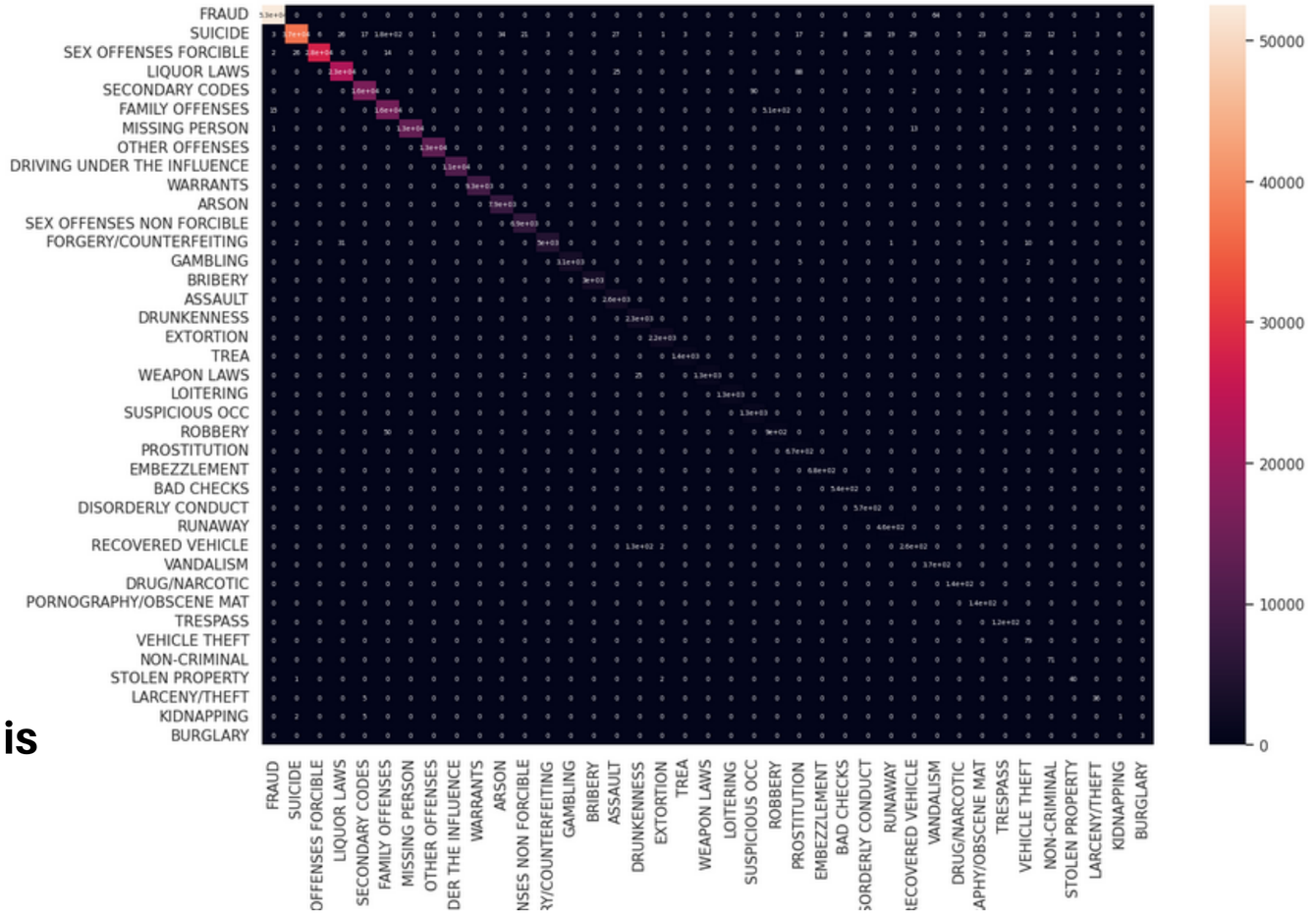
Error analysis (Naive Bayes)

A confusion matrix helps visualizing where the model misclassifies pairs of labels

Results are strong then there are not evident misclassifications

'Suicide' is usually confused with 'family offenses', which is then confused with 'robbery'

Another misclassification which can be verified is the one between 'vehicle recovered' and 'drunkness'.



Thanks for the attention!