**Crime Prediction in Chicago 2001 to Present**

1. **Setup**

* Created a new project by adding Spark and Watson-Machine Learning services and then loaded the dataset Crimes*-*2001*to*Present(1).csv which is of 1.6GB and launched the notebook with Python 3.5 and Spark 2.3 environments selected and cloud storage created to store the project.

1. **Load and explore data**

* Installed wget to get all the python files and then installed tensorflow and pyspark for accessing the data.
* A SparkSession dataframe is created to read the datafile into the notebook.
* **.printSchema()** s used to see the columns with their datatypes.
* Here, I individually took some columns into consideration to predict the crime rate in certain areas. So, I checked the distinct values present in certain columns and I am providing the count of only the columns which I considered to predict the data.

**Primary Type - 35**

**Year - 19**

**District - 25**

**Ward - 51**

**Community Area - 79**

**FBI Code – 26**

1. **Creating an Apache Spark Machine Learning model**

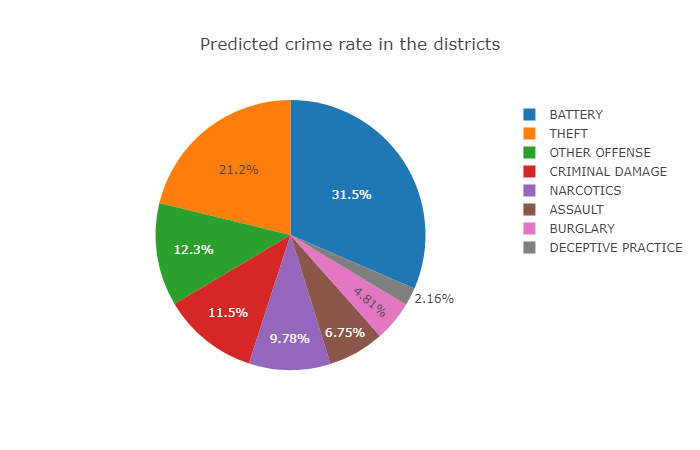
* Taking my dataset into consideration I randomly splitted my dataset into training, testing and predicting data. As it goes with the mapping method I made sure that the data is not skipped and want to predict atmost accurate data. Hence, I assumed training data to be 80% and testing data to 18% and then the data I want to predict is about 2%.
* I made sure that when performing data cleaning there shouldn't be any **NULL** values in my dataset. Hence, I placed **.dropna(how="any")** in my code so that it drops the rows which has null columns.
* I installed some pyspark.ml models to perform transformation, extraction, classification and evaluation on data. Firstly, from the columns I am taking into consideration Primary Type (main label) and FBI Code are object and rest are int32 and float64. So, I converted String to numerics using StringIndexer and then in the vectorAssembler I made sure that I placed the other columns which I am using and now using RandomForestClassifier I classified the label data. Now, again the strings are converted back from numeric to object and then placed in pipeline for training.
* **pipeline*rf.fit(train*data.dropna(how="any"))** this makes sure that the data is trained according to the percentage we mentioned and we can test the accuracy of data where in my project I got it as **Accuracy = 0.826085**.

1. **Persist model**

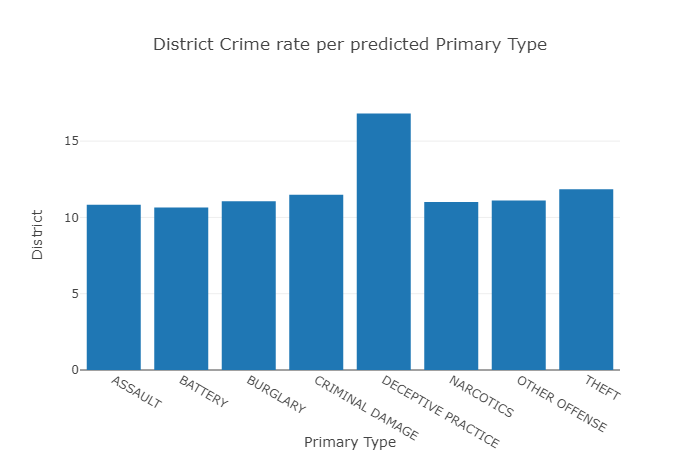
* I placed my service credentials of IBM Watson to store the ML repositories. I named my training data as Crime Line Prediction which stores only the trained dataset.
* In this part we are mostly concerned about saving the pipeline and model and then loading the model.

1. **Predict locally and visualize**

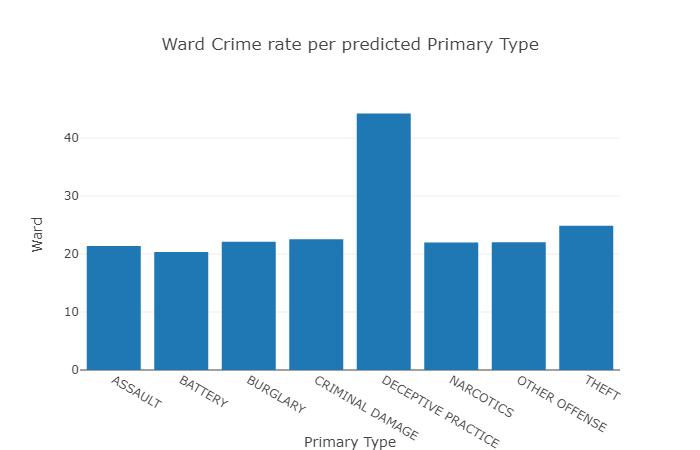
* Here, we are making a local prediction by using previously loaded model and test data and installing plotly for data visualization. I converted the Spark dataframe to Pandas dataframe to visualize data using **.toPandas**.
* **Predicted crime rate in the districts**

****

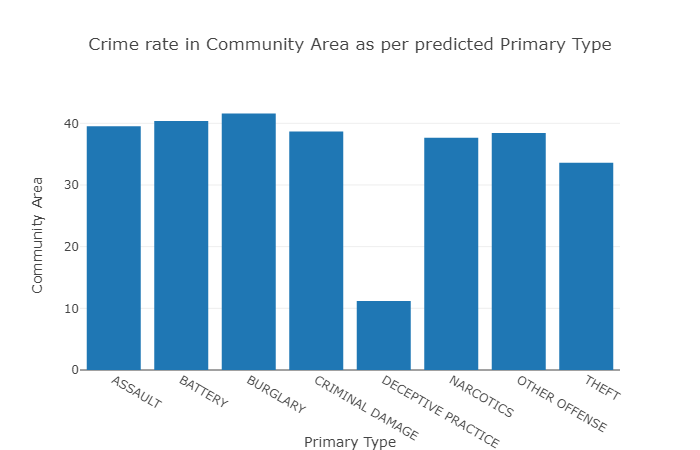
* **District Crime rate per predicted Primary Type**

****

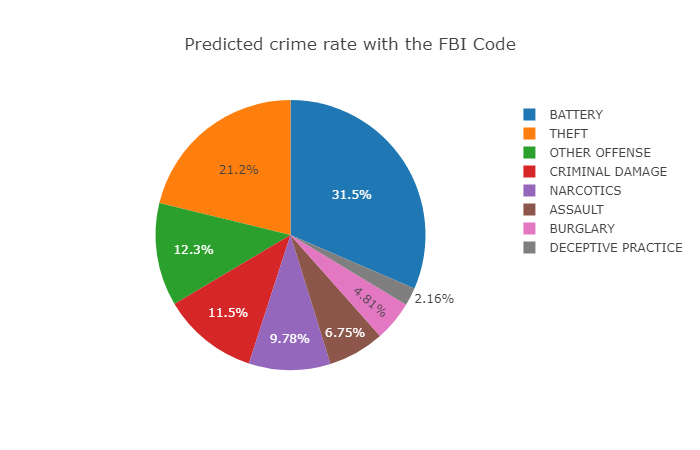
* **Ward Crime rate per predicted Primary Type**

****

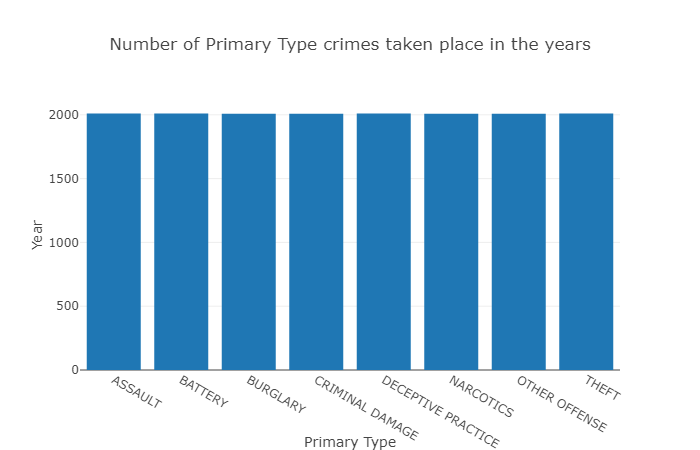
* **Crime rate in community Area as per predicted Primary Type**

****

* **Predicted Crime rate with the FBI Code**

****

* **Number of Primary Type crimes taken place in the years.**

****