Data Mining

CSCI-6674

Analytics and Insights Team

Project Report

Phase-6

Master’s in Computer science

University of New Haven

TAGLIATELA COLLEGE OF ENGINEERING, West Haven,

CT

Submitted to:

Prof. Dr. Reza Sadeghi

Table of Contents

Project Report : Phase – 6

Team Details

Team Name	:	**Analytics and Insights Team**

Team Members

1. Vyshampa Maringanti	:	vmari2@unh.newhaven.edu (Project Lead)
2. Akhila Gade	:	agade1@unh.newhaven.edu (Team Member)
3. Bagyasree Chitikela	:	bchit1@unh.newhaven.edu (Team Member)

Repository Address

https://github.com/vmari2/Crime-Prediction-of-NYPD.git

Research Question:

Can we predict the occurrence of offenses for a given precinct within the next three months, based on historical crime data of the New York Police Department (NYPD) by using various classification techniques?

Research Objective:

The main objective of this project is to create a predictive model using Negative Binomial Regression1 to predict the number of crime occurrences/offences for a given precinct of the NYPD2.This prediction will aid the police department in determining the required patrolling manpower per precinct.

Dataset Description:

The dataset covers the data of all crimes reported to the New York Police Department from 2006 to the end of last year i.e.,2019. This is a large dataset consisting of 6.98 million records with

35 attributes. Given, the large size, we could filter data to a more manageable size and per our project requirement.

Data set Source:

https://www1.nyc.gov/site/nypd/stats/crime-statistics/citywide-crime-stats.page.

Modeling Techniques

Modeling is of various kinds. Predictive Modeling is used to analyze the past data and predict the future outcome. In our project we implemented modeling on NYPD Dataset. NYPD maintains monthly crime data based on police incident reports across New York. The data is saved on the NYPD website with each month corresponding to an Excel file. Different modeling algorithms in Python have been used for the prediction.

There are various modeling techniques which can be used, some of the modeling techniques we used are as follows:

Modeling Technique	Accuracy	Precision	Recall
Logistic Regression	99.96	1.0	1.0
Random Forest	100	1.0	1.0
XG Boost	100	1.0	1.0
K-Nearest Neighbor(KNN)	99.97	1.0	1.0
Decision Trees	100	1.0	1.0

List of Parameters and Hyper Parameters

Modeling Technique	Parameters/Hyper parameters
Logistic Regression	‘C’, Penalty
Random Forest	n_estimators, min_samples_split, min_samples_leaf
XG Boost	max_depth, learning_rate, n_estimators, reg_lambda
K-Nearest Neighbor(KNN)	n_neighbours
Decision Trees	min_samples_split, min_samples_leaf,max_depth

Model Parameters and Hyperparameters

Logistic Regression:

Logistic Regression has different parameters, but we used specific parameters such as ‘C’ and ‘penalty’ . Penalty is used to specify the regularization. Usually ‘l2’ is the default penalty, which is used by most of the solvers, if we are not specifying any penalty then no regularization is applied. C parameter is used to control the penalty strength.

Random Forest:

The parameters used in Random Forest are ‘n_estimators’, ‘min_samples_split’, ‘min_samples_leaf’ . The number of random features to sample at each split point is the most significant parameter. A log scale of 10 to 1,000 might be good value. Until no further change is seen in the model, the number of trees should be increased.

XG Boost:

In XG Boost, the parameters used for the model are ‘n_estimators’, ‘max_depth’, ‘reg_lambda’ and ‘learning_rate’. The **n_estimators**: Number of gradient boosted trees. Equivalent to number of boosting rounds, **max_depth**: Maximum tree depth for base learners, **learning_rate**: Boosting learning rate and **reg_lambda** : L2 regularization term on weights.

KNN:

The number of neighbors (n neighbors) is the most significant KNN hyperparameter. It measures values between 1 and 21. The optimal number of neighbors for our project are 19. We also have some parameters in KNN such as weights and metric.

Decision Trees:

The parameters used in Decision Tree are like Random Forest, but we have an extra parameter namely ‘max_depth’ . If no max_depth values are specified, then the nodes are expanded until all leaves are either pure or till they are less than the min_samples_split values.

Optimization Techniques

Grid Search CV :

Grid Search is an efficient method for modifying supervised learning parameters and improving a model's generalization efficiency. It tries all possible combinations of the parameters of interest with Grid Search and finds the best ones. We used grid search cv in Logistic Regression model. In the Logistic Regression, the parameter that determines the strength of the regularization is called C. For a high C, we will set a less regularization and that means we try to fit the training set as fit as possible. We used scikit learn class library in order to perform Grid search CV and then specified parameters to search then GridSearchCV will perform all the necessary model fits.

parameters = {‘C’: [0.001, 0.01, 0.1, 1, 10, 100]

Randomize Search CV:

Randomizedsearchcv is similar to GridSearchCV but it can be used when we have more parameters to use. Random search is where we can use different combinations of parameters to find best solution for the model. We implemented Random forest, XG Boost and Decision tree classifier using RandomizedsearchCV. While using the RandomizedSearchCv we used the "tuned parameters" and also the parameters like 'n_iter', 'scoring', 'n_jobs' . The "tuned parameters" were different for different models based on the classifier we choose.

Performance Metrics

Accuracy: We got an accuracy of 99% for all our models for both training and testing data by using all features of the dataset. Below table represents the training and testing data accuracy of different models.

Modeling Technique	Train data accuracy	Test data accuracy
Logistic Regression	99.96	100
Random Forest	100	99.99
XG Boost	99.97	100
K-Nearest Neighbor(KNN)	99.74	99.99
Decision Trees	100	100

Precision and Recall : Precision and Recall are the evaluation metrics of a model. Precision helps us to know what proportion of the positive identifications are correct and Recall helps us to know what proportion of actual positives are correctly identified.

For the dataset and features we used, all the models had a precision and recall of 1.

Feature selection and Feature importance graphs for each model :

1. **Logistic Regression :** The below are the F-score for each feature used in the model. We can see that the highest score is recorded for Feature 2.

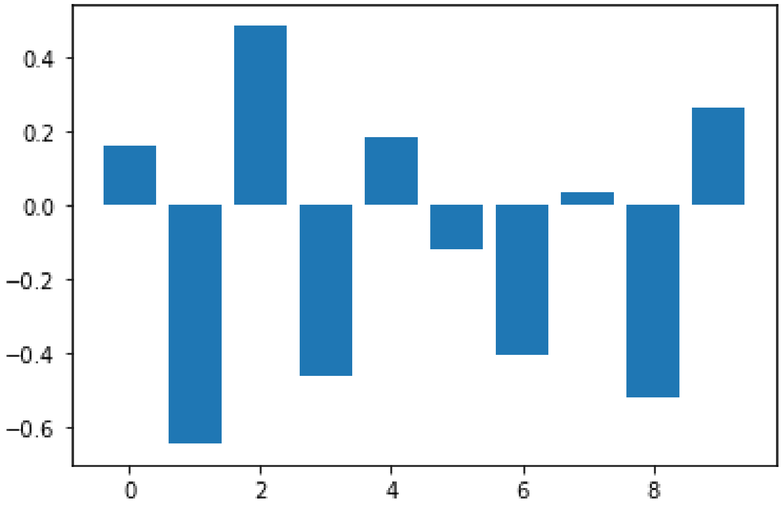
   Feature: 0, Score: 0.16320
   Feature: 1, Score: -0.64301
   Feature: 2, Score: 0.48497
   Feature: 3, Score: -0.46190
   Feature: 4, Score: 0.18432
   Feature: 5, Score: -0.11978

Feature: 6, Score: -0.40602

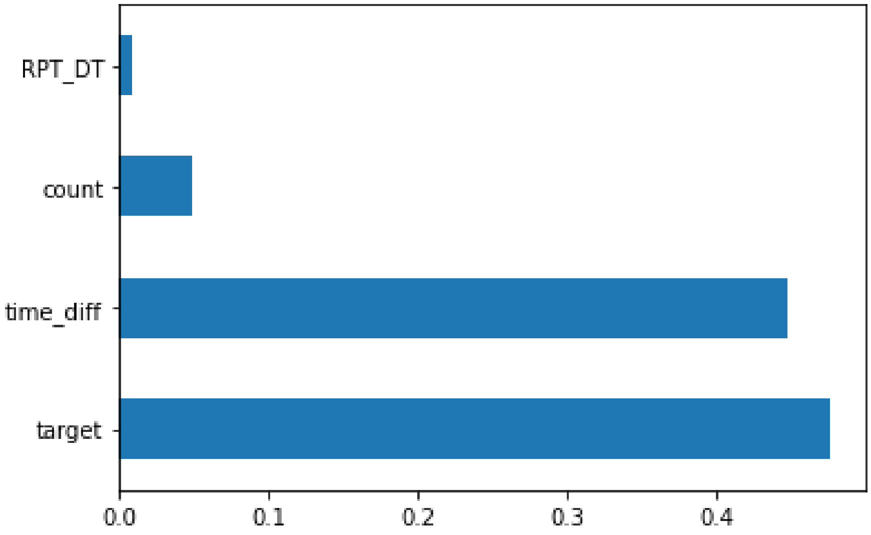
Feature: 7, Score: 0.03772

Feature: 8, Score: -0.51785

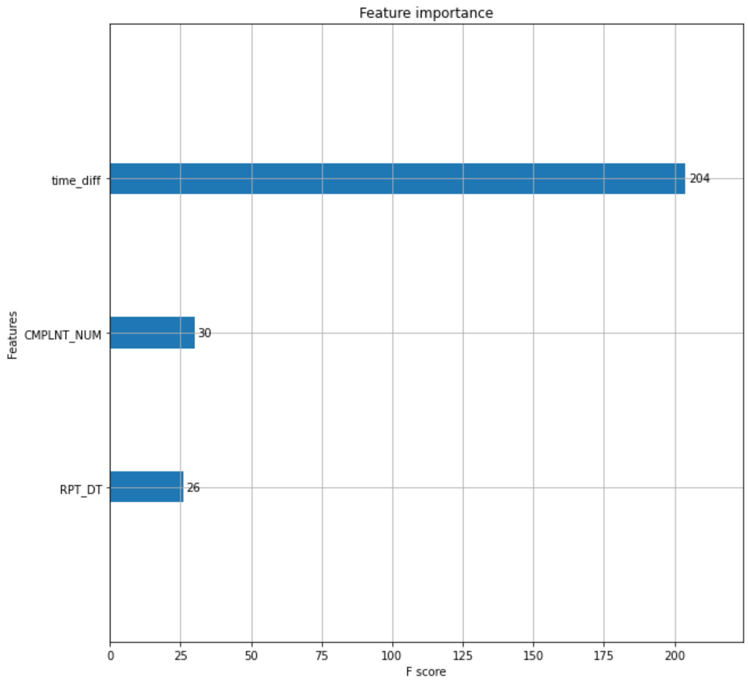
Feature: 9, Score: 0.26540



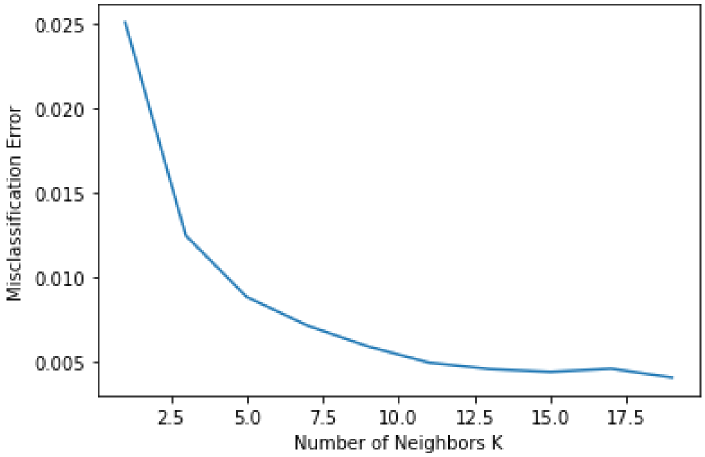
2. **Random Forest :** In the below graph, we can see that the target has the highest score of more than 0.4



3. **XG Boost:** From the below graph it can be noted that the time_diff has more importance than the other features.

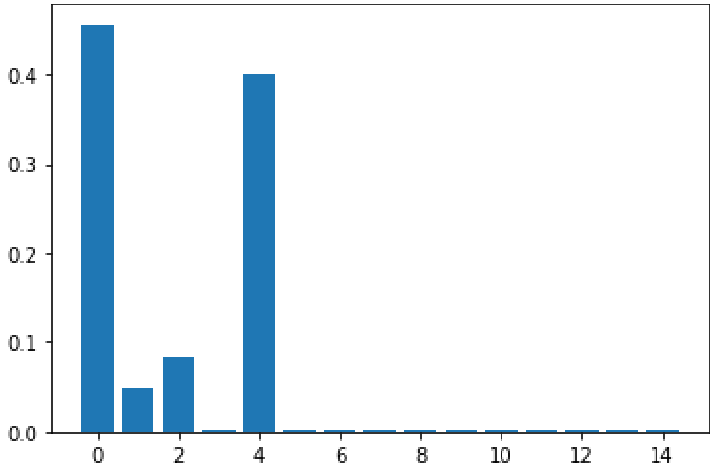


4. **KNN :** In the below graph, we can observe that the misclassification error decreases as the number of neighbors k value increases.



5. **Decision Trees :** From the below values we can see that the highest score is recorded for the Feature 0.

```
Feature: 0, Score: 0.45619
Feature: 1, Score: 0.04764
Feature: 2, Score: 0.08236
Feature: 3, Score: 0.00109
Feature: 4, Score: 0.39992
Feature: 5, Score: 0.00118
Feature: 6, Score: 0.00121
Feature: 7, Score: 0.00145
Feature: 8, Score: 0.00155
Feature: 9, Score: 0.00112
Feature: 10, Score: 0.00133
Feature: 11, Score: 0.00161
Feature: 12, Score: 0.00110
Feature: 13, Score: 0.00125
Feature: 14, Score: 0.00101
```



Conclusion

Initially, we considered the dataset of 1,200,000 records with 24 features but while modelling the data we observed that, the target is biased towards specific outcome i.e., the dataset is imbalanced. Then, we performed rigorous cleaning and we tried modelling data and observed that execution of models was time consuming and hence we had to reduce the data to 60,000 records with 24 features. By training our models with the training dataset we observed that all our models provide the accuracy of 99%. Since we observed a very high accuracy, we considered few more performance metrics such as Precession and Recall which were also high as accuracy i.e., 1. From this we can conclude that there is a higher possibility of crime in New York City.

References

1. https://machinelearningmastery.com/hyperparameters-for-classification-machine-learning-algorithms/
2. https://scikit-learn.org/
3. https://xgboost.readthedocs.io/en/latest/python/python_api.html#module-xgboost.sklearn
4. https://towardsdatascience.com/machine-learning-gridsearchcv-randomizedsearchcv-d36b89231b10