


 **Vaibhav3M / Chicago-crime-analysis** Public

BigData analytics on Chicago crime dataset.

☆ 3 stars 🍴 6 forks

 Star Watch ▼**Code**

Issues


Pull requests

Actions

Projects

Security

Insights

 master ▼

...

**Vaibhav3M** ...

on 23 Aug 2020

[View code](#)

Project-SOEN691-BigData

Team 30

Chicago Crime Dataset

Link: <https://data.cityofchicago.org/Public-Safety/Crimes-2001-to-present/ijzp-q8t2>

➤ Abstract

Crime is an inseparable part of our society, either being a victim or an offender everybody has witnessed crimes. In our project, we analysed the crimes data for that we selected the "Chicago Crime dataset report" which has the incidents of crimes for Chicago city that occurred from 2001 - present. We analyzed the trends of crime over the years, locations of the crimes where it happened the most, and hotspot of crimes over the years. In our experiment, we used supervised machine learning techniques including random forest, KNN and ensemble method for the prediction of crimes based on time, location and other parameters.

➤ I. Introduction

Crime is an important social problem in the country, affecting public safety, children development, and adult socioeconomic status. A police officer may know where the dangerous areas are according to his experience, but he may not be able to tell about what kind of crime could happen. Our objective is to help police officers by providing useful information which is hard to get, so that they can take appropriate measures. We have used Apache spark's Random Forest and SciKit-learn's KNN classifier and ensemble method for the same. Our project has below two parts -

Exploratory Analysis:

1. Trends of the crimes over the years (2010 - 2019)
2. Type of locations where crimes happen the most
3. Timelapse of crimes hotspots over the years (2010 - 2019)
4. A brief literal sense about those crimes

Predictive Analysis:

1. Predicting the type of crime(s) and probability of crimes based on location.
2. Predicting the type of crime(s) based on Time and also on other parameters.

Related Work: Many literatures are trying to predict the crimes. Sunil Yadav et al M. Timbadia is using supervised, semi-supervised and unsupervised learning techniques to predict the crime in India, including Apriori Algorithm, Naive Bayes Algorithm and regression algorithm [1]. Romika Yadav and S. Kumari Sheoran came up with auto regression techniques for time series data where she is trying to improve the accuracy of prediction by identifying the relationship among crime attributes [2]. B.Sivanagaleela and S. Rajesh is using a fuzzy C-means algorithm, but he only focuses on where the crime will occur and doesn't care about the type of the crimes [3].

References:

1. S. Yadav, M. Timbadia, A. Yadav, R. Vishwakarma and N. Yadav, "Crime pattern detection, analysis & prediction," 2017 International conference of Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, 2017, pp. 225-230.
2. R. Yadav and S. Kumari Sheoran, "Crime Prediction Using Auto Regression Techniques for Time Series Data," 2018 3rd International Conference and Workshops on Recent Advances and Innovations in Engineering (ICRAIE), Jaipur, India, 2018, pp. 1-5.
3. B. Sivanagaleela and S. Rajesh, "Crime Analysis and Prediction Using Fuzzy C-Means Algorithm," 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India, 2019, pp. 595-599.

➤ II. Materials and Methods

Dataset

The dataset chosen for this project consists of incidents of crime reported in the city of Chicago from 2001 to 2019. Data is extracted from the Chicago Police Department's CLEAR (Citizen Law Enforcement Analysis and Reporting) system. It is one of the richest data sources in the area of crime. The dataset includes enough information about Date, Type, Description, location etc about the crime for our analysis.

Approach and corresponding technologies

The dataset contains 7 million records of the crime. Data of this size needs fast and efficient data processing. We have used Spark framework as its in-memory processing capability makes it easy to deal with data of this volume. We have implemented some techniques such as: k-fold cross validation, KNN, Random Forest, ensemble method and feature selection.

Below is the pipeline we followed:

1. **Data Pre-processing:** In this step we chose data from year 2010-2019 as the accuracy stabilized for this time period.
 - Dropped missing/null values as it accounted for <1% of data.
 - Filtered out irrelevant features from the dataset.
 - Reduced/merged number of crime types from 32 to 16.
 - Used Random Over sampling/Under sampling techniques to balance the data.

Technologies: Apache Spark, pyspark Dataframe.

2. **Exploration Analysis:** In this step, we inferred useful information and analyzed important trends for crime detection and prevention. The analysis will also help identify useful features for building predictive models.

Methods: Bar graph, line graph, pie-chart, heatmaps, querying data.

Technologies: pyspark DataFrame, pyspark SQL, pyspark RDD, Matplotlib, Folium, Tableau.

3. **Predictive Analysis:** Below predictions were tried on both KNN and Random Forest and the results were compared with each other. Below are the steps involved:
 - We used random split, k-Fold Cross-Validation technique while training.
 - We further trained the model with additional features such as Location Description, Arrest etc. to achieve better accuracy.
 - We transformed categorical data to binary vectors using One Hot Vector/ Label Encoding.

- Used ExtraTreesClassifier, Correlation Matrix/HeatMap, Principal Component Analysis (PCA) as feature selection techniques.
- Tuned the hyperparameters such as no of neighbors in KNN and no of trees in Random Forest.
- Used an ensemble of different classification models and used soft voting for output.

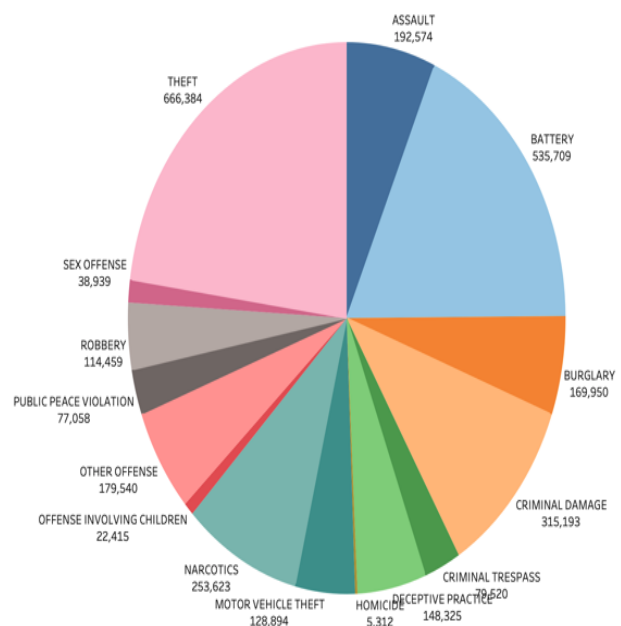
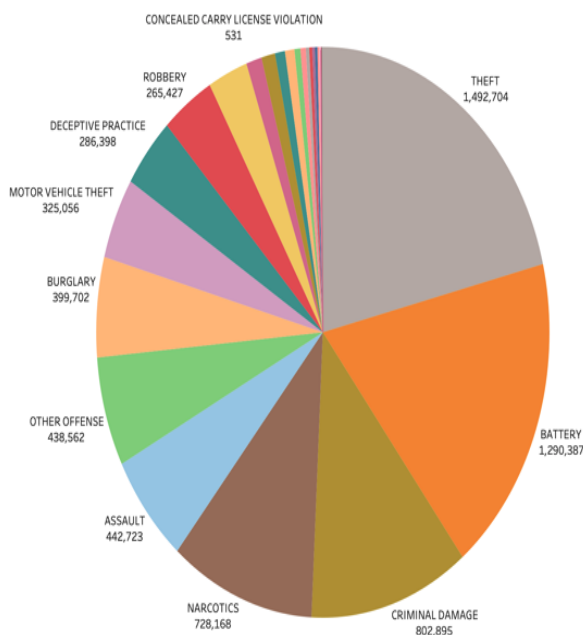
3.1. Predicting the type of crime (probabilities) based on its location: We used latitude, longitude as location to predict the type of crime. We used vector assembler to transform two columns into a vector.

3.2. Predicting the crime based on time(week): We used week as a feature to predict the crime based on time.

➤ III. Results

1. Important Preprocessing Steps

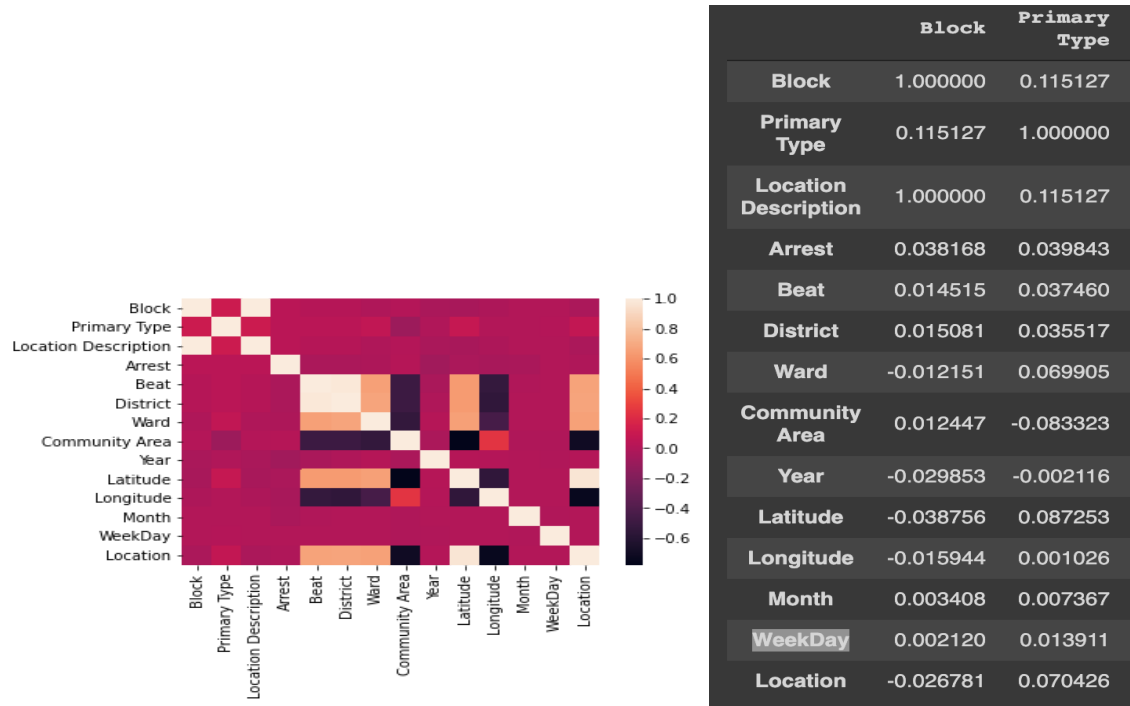
- **Dataset Analysis** - Our dataset was quite imbalanced and had a lot of features. Therefore, we tried making it balanced by merging similar types or dropping insignificant ones.



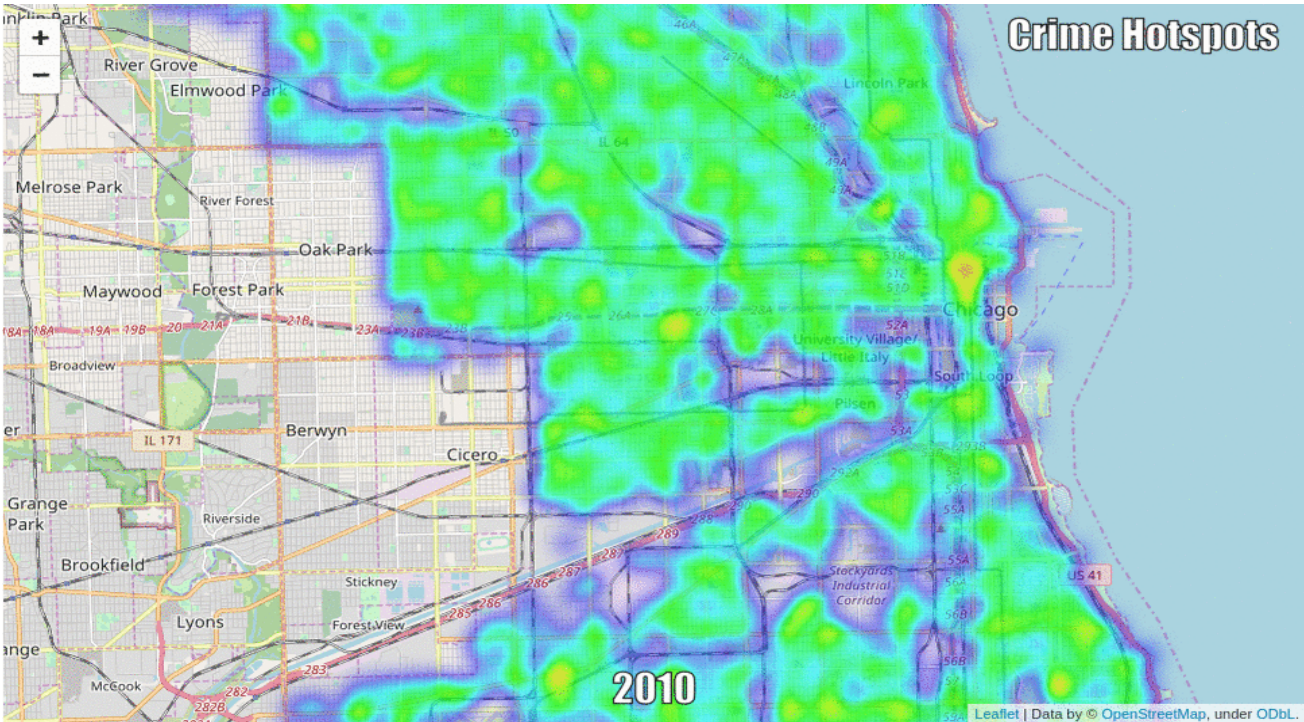
- **Feature Extraction** -
 - Feature importance in Extra Tree Classifier

- ii. Principal Component Analysis
- iii. Correlation Matrix/HeatMap.

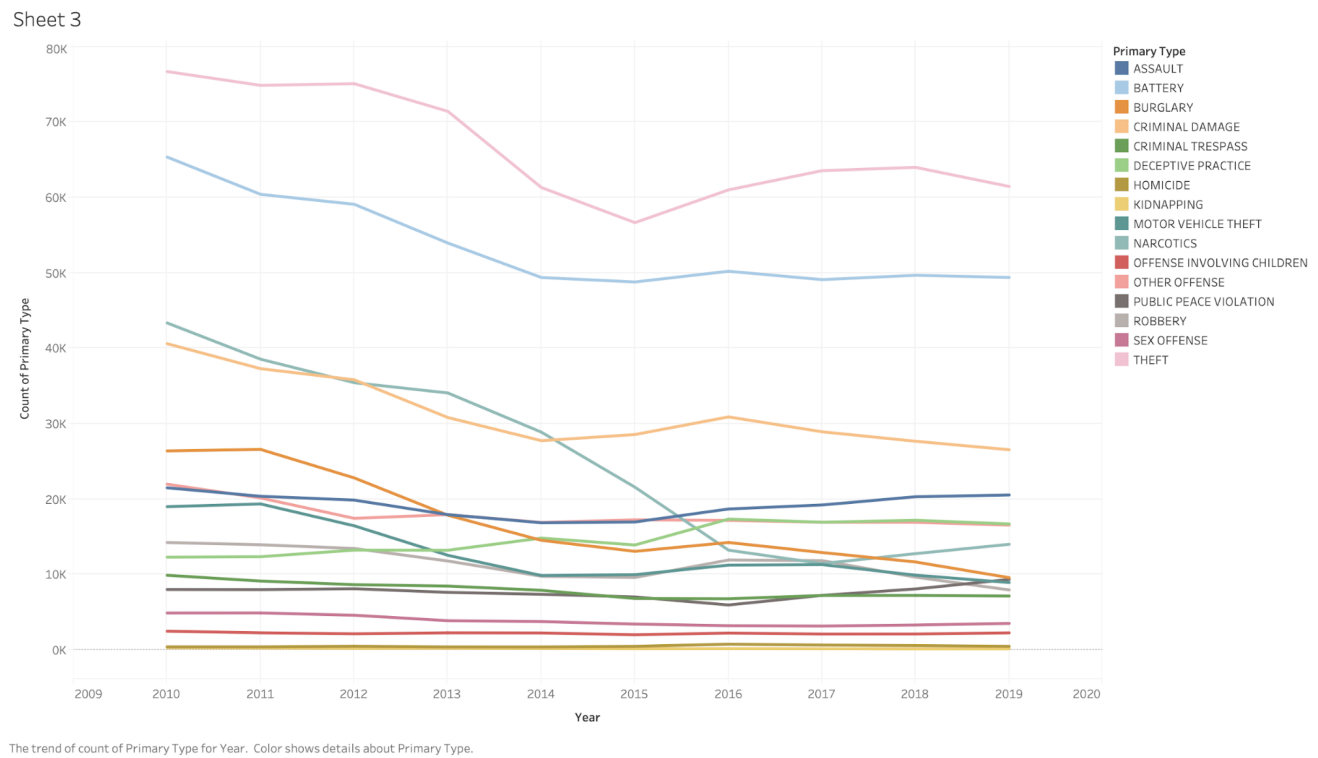
Correlation Matrix/HeatMap - The heatmap and matrix help us decide features which are in high correlation with Primary Type crime.



2. Exploratory Analysis:



Crime hotstops across the past decade



Trend of crime types across the past decade

3. Predictive Analysis:

Predicting the type of crime(s) and probability of crimes based on location and time data:

Prediction Model	Measure	Location Data Based	Time Data Based
Random Forest	Accuracy	26.33%	22.65%
Random Forest	F1 Score	17.58 %	8.37%
KNN	Accuracy	29.62%	27.7%
KNN	F1 Score	25.33%	21.2%

We concluded that location or time data alone donot provide sufficient details.

Predicting the type of crime(s) and probability of crimes based on both location and time data:

Random Forest Classifier

Grid-search and k-fold cross validation provided the best params for RF.


```
ParamGridBuilder()\
    .addGrid(rf.numTrees, [3,10,100])\
    .addGrid(rf.maxBins, [32,64,128])\
    .addGrid(rf.maxDepth, [5,10,15])\
    .addGrid(rf.minInstancesPerNode, [1, 5, 10])\
    .addGrid(rf.impurity, ['gini', 'entropy'])\
    .build()
```

parameter	value
maxBins	32
subsamplingRate	1.0
maxDepth	10
impurity	gini
minInstancesPerNode	1
minInfoGain	0.0
maxMemoryInMB	256
checkpointInterval	10
featureSubsetStrategy	auto
numTrees	10

Results:

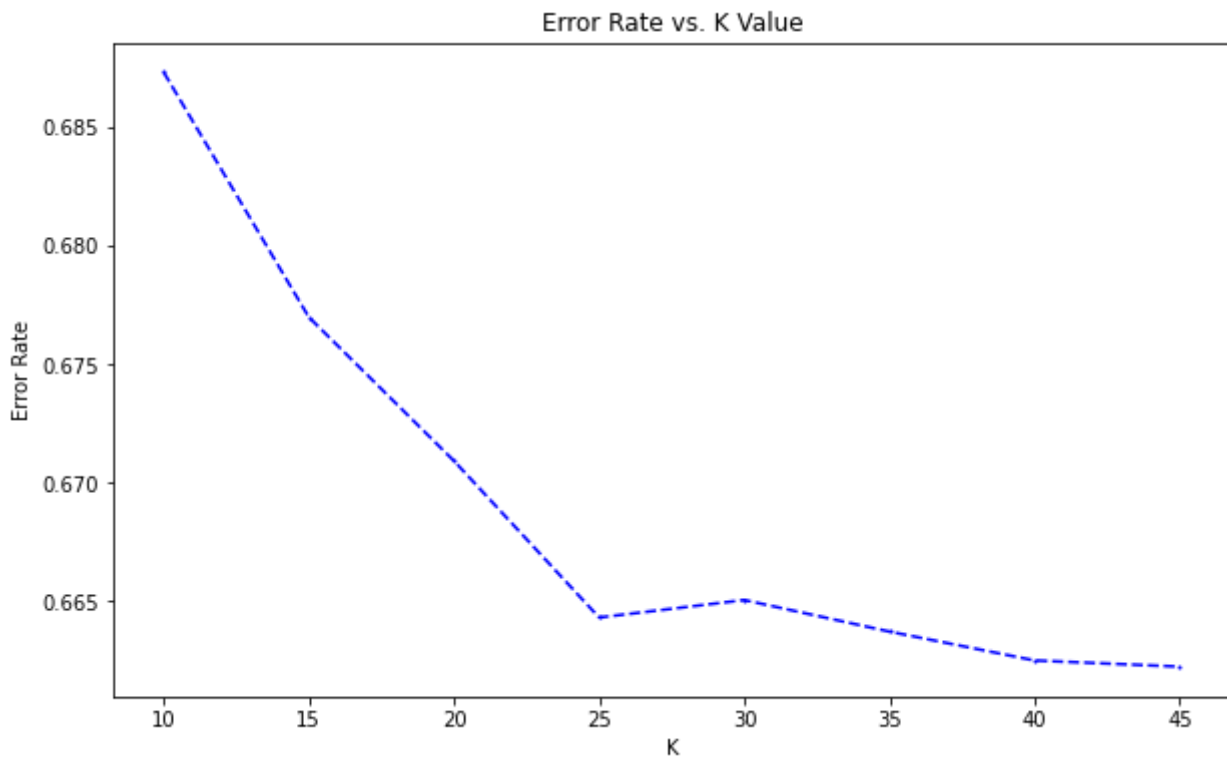
- Accuracy = 36.86%
- F1 score = 25.42%

Additionally providing crime probabilities.

predictedLabel	Primary Type	Latitude	Longitude	probability
0	NARCOTICS	HOMICIDE	41.789497 -87.672963	{BATTERY=0.12389312747887293, HOMICIDE=0.004287664351617304 , ROBBERY=0.024675434994493085, SEX OFFENSE=0.03402628882875742, OFFENSE INVOLVING CHILDREN=0.0031874394937005737, NARCOTICS=0.26812021522576096 , CRIMINAL SEXUAL ASSAULT=6.565644220574121E-6, CRIMINAL TRESPASS=0.0356014328113521, CRIMINAL DAMAGE=0.07445128604299998, DECEPTIVE PRACTICE=0.020746391069255053, OTHER OFFENSE=0.06791794207885075, PUBLIC PEACE VIOLATION=0.09139155510488867, ASSAULT=0.05495839694148831, MOTOR VEHICLE THEFT=0.06105804132567455, BURGLARY=0.007274508463013829, KIDNAPPING=4.4972687407872776E-4, THEFT=0.12795398327097504}

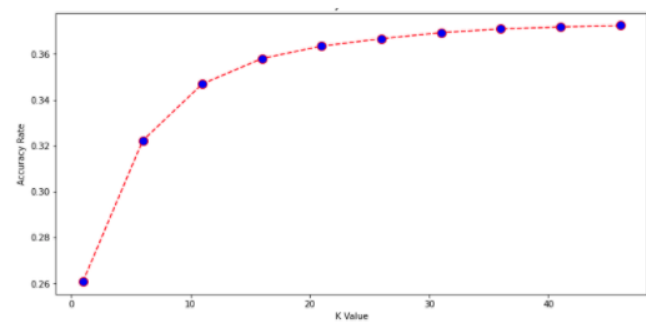
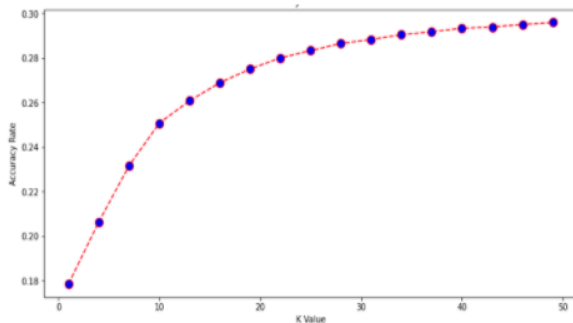
KNN Classifier:

Finding optimal K, using the elbow method.



Optimum K = 25 Parameter tuning using Random Search and K-Fold cross-validation:
 'weights' = 'uniform' 'metric' = 'manhattan' (Haversine - in case of Latitude and Longitude)

Accuracy comparison before and after training model with additional features



Impact of sampling on the KNN model:

Model with no sampling:

- Accuracy - 33.5%
- F1 Score - 29.6%

UnderSampling

Sampling Technique	Accuracy	F1 Score
Cluster Centroids	24.92%	19.55%
Random	24.93%	19.55%

OverSampling

Sampling Technique	Accuracy	F1 Score
SMOTE	32.96%	29.62%
Random	41%	31.60%

Random oversampling of minority classes improved the prediction of the model. This could be as the model now better fits the minority data due to availability of a higher number of instances

Ensemble models - Voting Classifier

An ensemble of KNeighborsClassifier, RandomForestClassifier, and SVC. We have used soft voting for output. Individual accuracy:

☰ README.md

- RF -> 33.65%
- SVC -> 22.81%
- Overall Ensemble - 35.21%

4. Comparison of best models from each category:

Measures	Random Forest	KNN (K = 25)	KNN (OverSampling)	Ensemble (KNN, RF, SVM)
Accuracy	36.8%	33.5%	41%	35%
F1-Score	25.4%	29.6%	31.6%	26.7%
Time(Approx.)	5 mins	25 mins	30 mins	1 hour

KNN (OverSampling) provides the best results.

➤ IV. Discussion

Relevance of solution:

- Machine Learning models are as good or as bad as the data you have. Correlation between features is important for predictions. In our case, we experienced low correlation features with our predicting variable. We experimented with different features in order to get better predictions such as using week/month/year to predict crime type based on time and using additional features such as location description, arrest. The results became better, however, not significant enough.
- The original dataset was highly imbalanced. Even after dropping/merging related some crime types we still had an imbalance of 100:3. Then, we tried sampling techniques for balancing. Random Oversampling gave best results in comparison to other sampling techniques. However, the increase was comparably small. Applying combination of both undersampling and oversampling might result in better overall performance.
- Ensembling various classification models also seemed useful, particularly 'soft voting' technique provided better results
- We used Google Colab as development environment.
 - GPU acceleration for sklearn models
 - 25GB RAM
 - Easy collaboration between team

Limitations:

- Predicting crime patterns have complicated factors, some of them are related to sociology, economics, even history, and geography. The tasks can be further extended to include information about the victims and the offenders are made available.
- Good predictions are based on two factors: Good Model, but more importantly, good data. Even though we have a big dataset, the features it provides are not good to predict where and when a crime may happen.
- Not all crimes had a good correlation with parameters such as latitude and longitude.

Future work:

- Adding data: More data such as economic, demographic and weather data can help make better predictions.
- Using models such as XGboost and Neural Network to identify patterns between data.
- Focus on specific crime types can provide better intuition.
- Using a combination of oversampling and undersampling techniques.

License

licence MIT

Releases

No releases published

Packages

No packages published

Contributors 4



Vaibhav3M Vaibhav Malhotra



SinglaAnkur



PritamKumar24



HanfordWu Hai-feng

Languages

● **Jupyter Notebook** 99.5% ● **HTML** 0.5%