Abstract

It is challenging to predict crime, which has been a big issue, especially in metropolitan areas with large populations. Many federal agencies heavily rely on the Chicago open past criminal data to find various patterns. In order to uncover and analyse crime trends and lower the crime rate, criminology employs a variety of crime analysis and prediction tools. Machine Learning techniques were applied in this project to comprehend the dataset on historical crimes in Chicago and to give a statistical and graphical analysis of the relationships between the dataset's multiple attributes.

I employed a variety of machine learning models and assembling approaches to estimate the likelihood of crime in different Chicago neighbourhoods. My models were trained on historical crime data from the Chicago Police Department, and a range of evaluation metrics were utilized to assess the model's accuracy. My findings demonstrate the model's capability to accurately predict the occurrence of crimes across the city where decision tree algorithms with boosting outperformed other algorithms. This study reveals how machine learning could be used to address the issue of urban crime.

Introduction

Crime is a pervasive problem in urban areas, with many cities struggling to keep their citizens safe. The city of Chicago is no exception, with a significant crime rate that has garnered national attention. However, advances in computer science have enabled researchers to develop sophisticated tools to analyse and predict criminal activity, giving law enforcement agencies the ability to be more proactive in preventing crime.

Due to its high crime rate and reputation as one of the most violent cities in the country, Chicago is a great place for study on crime analysis and prediction. There are some factors which make predicting crime difficult. One of these is data quality, as the information that is currently available is frequently inaccurate, skewed, and incomplete. Another is the influences on crime, which might include social and economic elements that make it challenging to anticipate crime with accuracy as these vary with every city. I created robust machine learning algorithms that can learn to identify patterns in the data and predict the crime level of the city. By identifying high-risk regions and allocating police resources accordingly, these models have the potential to lower crime rates and make cities safer for their inhabitants. To enhance the performance of the models, I applied modern ensemble techniques.

Exploratory data analysis was performed on the crime data, and characteristics such as the crime's location and time were used to uncover underlying trends over time. Under- standing the relationships between crime and other factors was made easier by visualizing the patterns of crime. It was discovered that a yearly pattern of crime was discovered, with the months of March through August consistently having the highest crime rates.

The Chicago crime data set was taken from Kaggle. The year wise records were obtained from the source from 2010-2022. The data contains information on various types of crimes, including homicide, robbery, burglary, theft, and more. The data includes the date, time, location, and type of crime, whether arrested or not. The data set before pre-processing consisted of 23 features.

Methodology

Regressor:

● Decision Trees: The purpose of employing a Decision Tree is to build a training model that can predict the class or value of a target variable by learning basic choice rules from past trained data. In Decision Trees, I begin at the root of the tree to forecast a class label for a record. The values of the root attribute are compared to the values of the record's attribute. Based on the comparison, I follow the branch corresponding to that value and proceed to the next node. This model is also used to predict the crime level of the region. This model was chosen because decision trees themselves aid in lowering the dimensionality of the data by performing splitting based on the most crucial feature that is provided.

Classifier:

● Random Forest: Random Forest is a technique based on decision trees that is used in forecasting predictions and behaviour analysis. It contains numerous decision trees, each reflecting a unique instance of the random forest's classification of data input. The random forest approach examines each incident independently, selecting the forecast with the highest number of votes. Each tree in the classifications is fed samples from the initial dataset. The features are then chosen at random and utilized to build the tree at each node. Until the end of the exercise, when the prediction is achieved conclusively, no tree in the forest should be trimmed. The random forest allows any classifier with weak correlations to produce a strong classifier in this way. This model is used to predict the crime level of the region. I chose this model because as there are multiple features which affect the crime of the city, this multi class classification method will best suit the needs.

Clustering:

● KNN: The data is categorized in K-Nearest Neighbour (KNN) based on the majority vote of the neighbours of that specific new data point, which has yet to be classified. It works by calculating the distance between a new point and all the previous data points, and then predicting the label of the new location by assigning the most commonly occurring label based on the nearest 'k' neighbours. In this study, KNN was used to identify the crime level by looking at prior crimes, recognizing comparable crime patterns, and predicting the crime level value based on the neighbours. Feature Engineering: ● Data transformation helps in improving the quality of data and make it suitable for data analysis. The first step performed was to create a single data frame of records by combining the csv files of each year. Three new features namely hour, date, month and year were created from the timestamp feature. These features were necessary for this project to successfully find the spatial patterns of crime. Alarm feature was also added to the data set with three values. 0 indicating that crime rate of the ward is less than 15, 1 indicating that crime rate is between 15 to 25 and 2 indicating crime rate over 25. This feature was added to predict the crime level of the region at any given day.

Advanced Modelling:

● Bagging is a procedure that creates multiple copies of the original training data set using the bootstrap, fitting a separate decision tree to each copy, and then combining all of the trees in order to create a single predictive model. I employed bagging in the project as it helps in reducing overfitting and improving the stability of the classification model by reducing the variance in the predictions.

● Boosting refers to any Ensemble method that can combine several weak learners into a strong learner. The general idea of most boosting methods is to train models sequentially, each trying to correct its predecessor.

Anomaly Detection:

● Anomaly detection is a technique for discovering data points in a dataset that differ considerably from the norm or anticipated behaviour. These data points are referred to as "anomalies" and can be generated by a variety of variables such as measurement mistakes etc. DBSCAN is a clustering and anomaly detection unsupervised machine learning technique. Clusters in the data space are defined by the algorithm as areas of high density divided by areas of low density. The approach is especially beneficial when dealing with huge datasets with complicated structures and an unknown number of clusters.

● DBSCAN was utilized in my code to detect anomalies. Missing values were deleted and DBSCAN was used to cluster the data, with an epsilon of 0.1 and a sample count of 10 necessary to establish a dense zone.

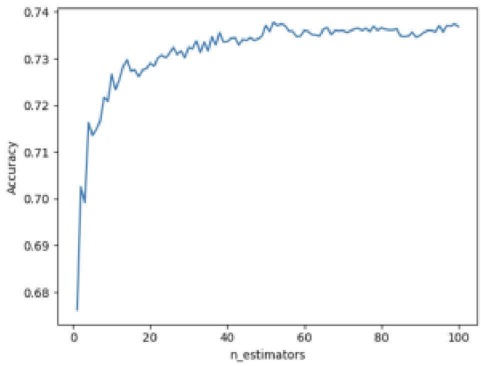
Results

Model	Accuracy	R-Squared	MAE	MSE
Random Forest	**74.05**	**40%**	**0.31**	**0.32**
KNN – K fold	**67.91**	**9%**	**0.42**	**0.48**
DT Regressor	**59.71**	**12%**	**0.55**	**0.46**
Bagging Cl	**62.00**	**19%**	**0.39**	**0.42**
Ada-Boost	**91.03**	**19%**	**0.39**	**0.42**

In my project I utilized different types of models to achieve the best possible accuracy and to know by which method I will get the most appropriate answer. So, with this aim I used 5 different types of models to check which amongst these will give us the highest accuracy. The 5 different models that were used are Random Forest, KNN - K Fold, Decision Tree Regressor, Bagging classifier and also Ada-Boost classifier. After using all these models in my project on my dataset I came to a conclusion that Ada-Boost is the best performing model with an accuracy of 91.03%, followed by Random Forest with 74.05% accuracy. KNN, Decision Tree Regressor, and Bagging Classifier have relatively lower accuracy values compared to the top two models

Classifier:

In my project, I employed a Random Forest Classifier with 100 decision trees and entropy as the criterion to predict the target variable. The scikit-learn library's metrics module was used to evaluate the model's accuracy by computing various metrics such as accuracy score, mean absolute error (MAE), mean squared error (MSE), and R-squared value on the test set. The results showed that the model had an accuracy of 69.58% and an R-squared value of 0.41. In addition, I also used K-Fold cross-validation to assess the model's accuracy. The data was divided into training and testing sets, and the accuracy of the model was computed using the accuracy score method from the metrics module in each cycle. The obtained accuracy scores were averaged to compute the mean accuracy rate, which was found to be 74.06%. I discovered that K-Fold cross-validation delivers a higher mean accuracy score than single test-train split assessment. This indicates that the model's performance is more consistent across data folds. I used grid search to find the optimal parameters for training and evaluating the models. The image shown below provides us with variation in accuracy of the model with different numbers of estimators. After analysing the graph, I selected the number of estimators to be 100 as it provided the maximum accuracy

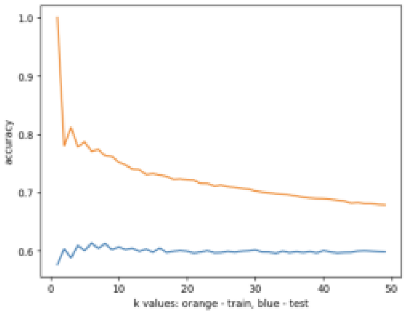


Regressor:

The Decision Tree Regressor approach was used to train a model on the given dataset using various values of the maximum depth parameter. The loop iterates from 1 to 14 times to experiment with different depths for the model. For each model, the mean squared error (MSE), mean absolute error (MAE), root mean squared error (RMSE), accuracy, and R-squared values were calculated. The values for these measures were then updated with greater precision for each model. The final results for the best-performing model's R-squared, MAE, MSE, and accuracy were then printed. The best-performing model had an R-squared value of 0.13, an MAE of 0.56, an MSE of 0.46, and an accuracy of 59.71%.

Clustering:

The K-nearest neighbours (KNN) classifier was trained using 17 neighbours which was found out using grid search approach and then tested on the test set. The values for accuracy, mean squared error (MSE), mean absolute error (MAE), and R-squared were computed and reported. The model's accuracy was determined to be 59.71%. The value of R squared was 0.094. On the same KNN model, K-fold cross-validation was performed with 10 splits, where the data was divided into 10 equal parts. Each fold's accuracy score was determined, and the average accuracy score was computed and printed. The average accuracy rate was discovered to be 67.92%, which is greater than the accuracy score achieved by testing the model alone on the test set. I discovered that K-fold cross-validation provides a more accurate assessment of the model's accuracy than testing it on the test set alone. This is due to the fact that K-fold cross-validation lets us use all of the data for training and testing, whereas testing on a single test set may result in the model being overfitted or under-fitted.

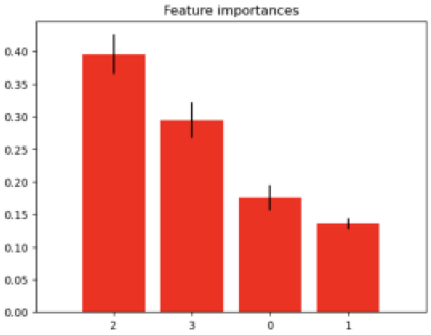


Feature Engineering:

Data transformation helped us in improving the quality of data and making it suitable for data analysis. The first step performed was to create a single data frame of records by combining the csv files of each year. Three new features namely hour, date, month and year were created from the timestamp feature. These features were necessary for this project to successfully find the spatial patterns of crime. Level feature was also added to the data set with three values. 0 indicating that crime rate of the ward is less than 15, 1 indicating that crime rate is between 10 to 20 and 2 indicating crime rate over 20. This feature was added to predict the crime level of the region at any given day.

Advanced Method:

I first employed the random forest bagging classifier to determine the significance of the features in the dataset during the model's training phase. The collected results are displayed below, with feature 2 (hour of the day) being the most critical feature. The Adaboost classifier was then examined with a decision tree as the base classifier, and it achieved the highest accuracy among all classifiers of 90%.



Conclusion

Using the data management and data analysis components of big data analytics, I examined the Chicago crime dataset. With the help of Python’s strong libraries like Pandas, matplotlib, and scikit-learn, many models have been successfully created. Comparisons were made between the evaluation outcomes of models before and after using the ensemble techniques. For estimating the region’s crime level, I was able to reach an accuracy of over 90 percent. Using the ensemble techniques, I was also able to improve the models’ performance.