Final Report

Chicago city severity of crashes Analysis and predictions

Introduction

Chicago, the primary city in the state of Illinois, is the largest urban center in the state and the third-largest in the United States. As a major hub for commerce, industry, transportation, and culture, it boasts a population of over 2.7 million residents. Sadly, the city is also known for its high number of traffic accidents, which have a devastating impact on countless individuals each year. In an effort to improve the quality of life for residents, the city has launched initiatives called Vision Zero, which aims to prevent such tragedies. A key objective of this program is to reduce speed-related fatal and serious injury crashes by 25%, thereby keeping the city's roads safe.

Problem Statement

By using the crash data from city of Chicago, I analyzed crash severity in the city of Chicago and developed a predictive model that can accurately identify fatal and serious injury crashes based on various crash types in different neighborhoods across the city. I used various exploratory techniques such as identifying and handling missing values, data visualization, including creating charts, graphs, and other visual representations of the data to identify patterns, trends, and relationships between different variables in the dataset. I also used descriptive Statistics to understand the mean, median, mode, standard deviation, and variances in the dataset helping to describe the data and identify key features. I also conducted feature dngineering, including creating new features from the existing ones to improve the performance of the machine learning models.

I reduced the dataset from 600,000 to 200,000 as this is a common technique to save computational time and reduce processing requirements for data analysis tasks. This helps me to focus on the most relevant and informative data points. After comparing 5 different classification models, my tuned Random Forest model was able to achieve the highest model accuracy of 0.998.

This model can be implemented to assist the City of Chicago in predicting the severity of a traffic crash, allowing them to allocate their emergency resources more efficiently. By identifying potential severe/fatal crashes in advance, the city can prepare and respond more effectively, potentially reducing the impact of these incidents.

Note: is important to ensure that the reduced dataset still maintains a representative sample of the original data and that any analysis or modeling based on the reduced dataset is reliable and accurate. You may need to carefully select the data points to include in the reduced dataset based on their relevance, diversity, and representativeness, while ensuring that the data integrity is not compromised.

Data Wrangling

The raw dataset from Chicago city crash data had 600,000 rows and 49 features. I reduced the size of the column because some columns were not relevant for the analysis. I also reduced the size of the row to save computation time. The dataset is updated daily to include new dataset describing new crash incidents. After reducing the dataset, I remained with 200,000 rows and after dropping irrelevant features, I remained with 38 columns.

Furthermore, when dropping column, I made sure to include featues that have more than 65% null value. I used the 'fillna' method to replace all missing values in the DataFrame with the mean value of the respective column.

After the data wrangling process, my data frame had 200,000 rows and 38 columns.

Exploratory data Analysis

Through visualizations and statistical analysis, I was ablet to gain insights into the underlying patterns and trends in the data, identify potential factors that contribute to crashes which can help develop strategies to reduce the incidence and severity of crashes in the future.

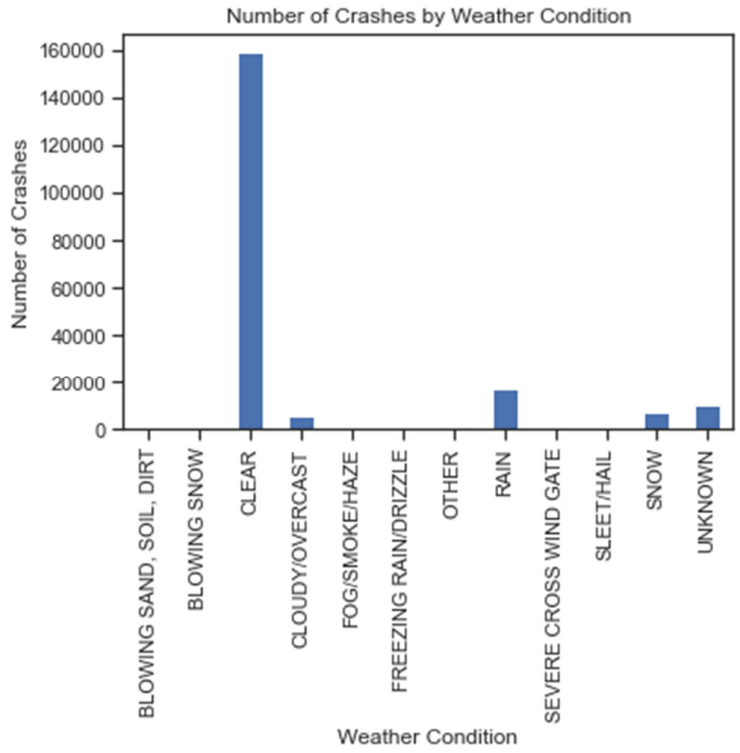
By conducting EDA on this dataset, I explored he following questions:

1. What are the most common causes of traffic crashes in Chicago?

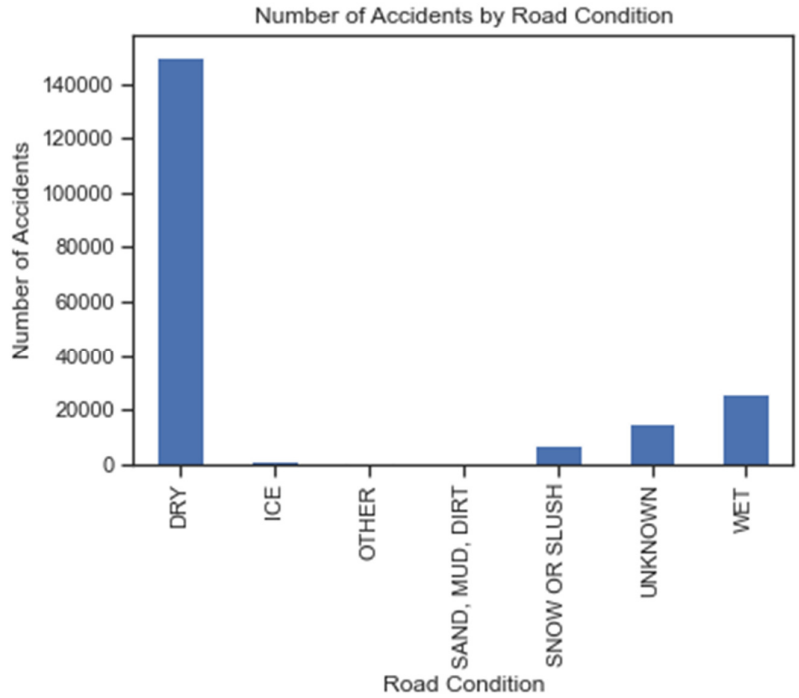
   To analyze the most common causes of traffic crashes, I looked at the 'PRIM_CONTRIBUTORY_CAUSE' variable, which provides information about the primary contributing cause of the crash. After calculating the Chi-squared statistic, I was ablet to determine that the following factors were top 5 causes of traffic crashes in Chicago.
   - Failing to yield right away
   - Following too closely
   - Improper overtaking/passing.
   - Failing to reduce speed limit to avoid crash.
   - Improper backing

2. What is the relationship between weather conditions and the incidence of crashes?

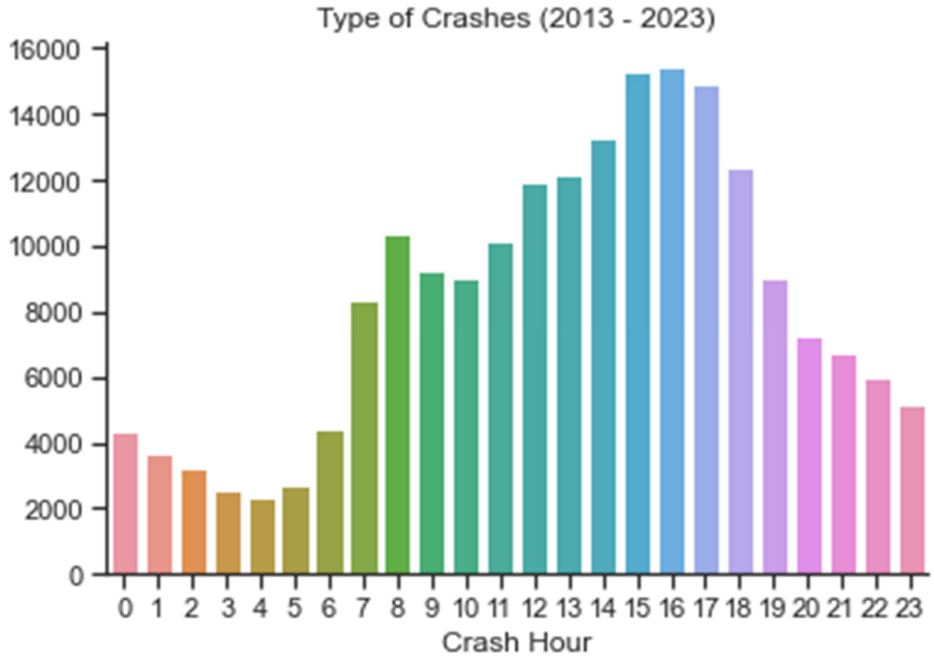
   Most crashes happened in clear weather conditions. However, it is important to note that this analysis only considers the incidence of crashes, and does not account for factors such as driver behavior or road conditions, which may also play a role in the relationship between weather conditions and crash risk.



I also investigated the relationship between road conditions and the number of accidents. As the graph below shows, the majority of accidents happen in dry conditions.

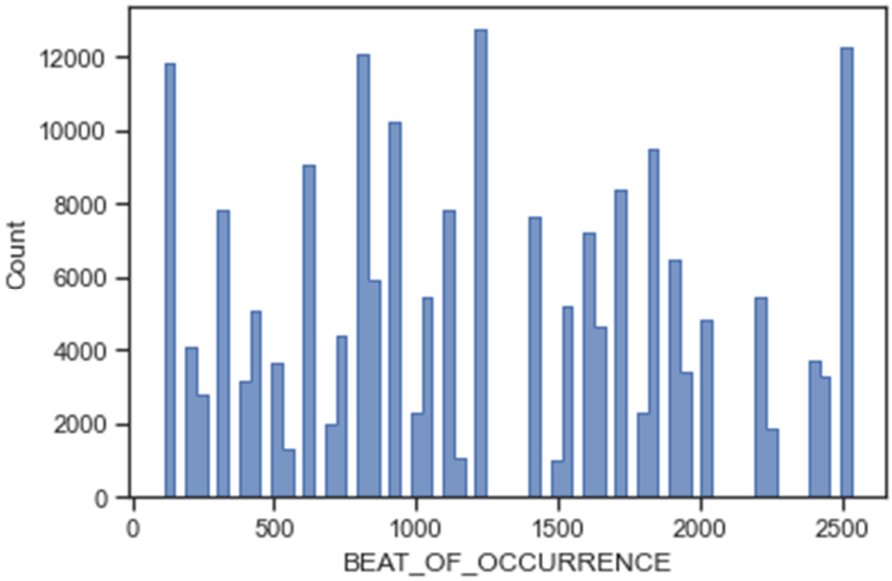


3. Is there a relationship between the time of day and the severity of crashes?
   Yes, as the graph below shows, pick hour for severe crashes is between 2pm-6pm

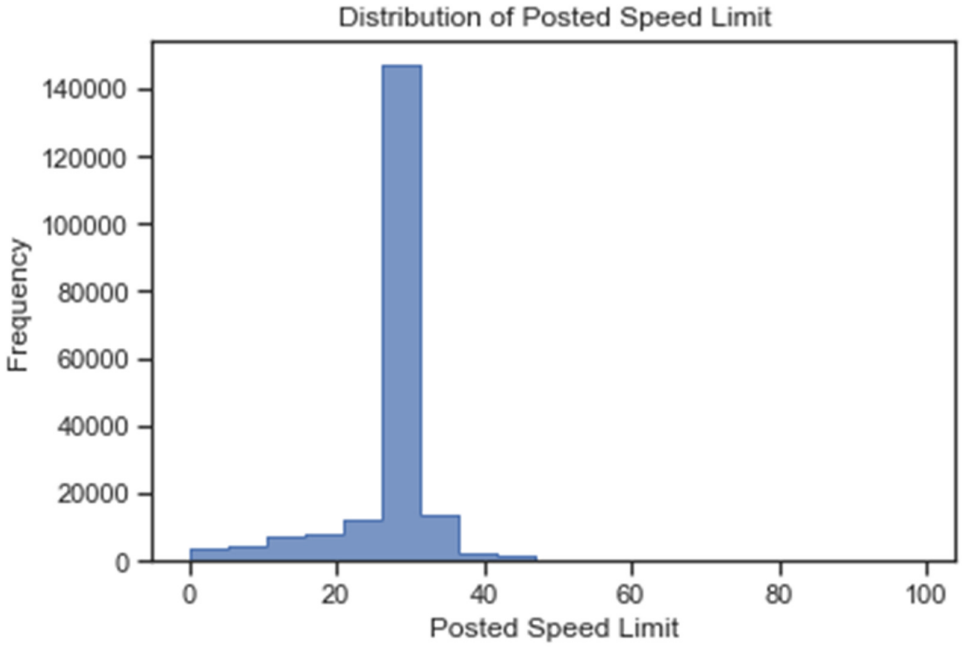


4. Are there specific areas of the city that are more prone to crashes?
   Yes, based on my analysis, the top 5 BEAT_OF_OCCURRENCE with the most accidents are:
   1. Beat 1834.0 with a total of 2446 accidents
   2. Beat 114.0 with a total of 2035 accidents
   3. Beat 813.0 with a total of 1975 accidents
   4. Beat 1831.0 with a total of 1891 accidents
   5. Beat 815.0 with a total of 1834 accidents

   With beat 1834 and beat 1831 located in the southern part of Chicago, one can conclude that the southern part of Chicago has the highest number of accidens.



5. What speed causes the most accidents?



The graph above shows that most accidents happen on roads with a speed limit of 30 mph.

By conducting EDA on this dataset, I was also able to identify correlations between the following features and severe injuries that resulted from crashes:

• There is a strong positive correlation between the "INJURIES_TOTAL" and "INJURIES_NON_INCAPACITATING" columns.

• There is a strong positive correlation between "INJURIES_TOTAL" and "INJURIES_REPORTED_NOT_EVIDENT".

The following features have no correlation or are negatively correlated.

• "CRASH_HOUR" and "CRASH_DAY_OF_WEEK" have a weak correlation, indicating that the time of day and day of the week do not strongly affect the occurrence of traffic crashes.

There is a significant association between contributory causes and crash severity.The combinations of contributory causes "PRIM_CONTRIBUTORY_CAUSE" and "SEC_CONTRIBUTORY_CAUSE" lead to more severe crashes.

The following areas have been identified as high-risk areas:

- Pulaski Rd.,
- Cicero Ave,
- Halsted st, and
- State St..

Model Selection

In this project we are dealing with classificion problem. Thus, to predict the severity of a crash, we have considered the following classification models: K-Nearest Neighbor (KNN) Random Forest Logistic Regression Gradient Boost Naive Bayes

After conducting a feature selection analysis, the featues "INJURIES_TOTAL", and "MOST_SEVERE_INJURY" were dropped as they were leading to bias results.

To avoid overfitting, it is not recommended to evaluate the performance of a model by training and testing on the same dataset. Instead, the dataset should be split into a train set and a validation set for model evaluation. However, the choice of (train, validation) set can affect the performance of the model, which can be overcome by using the Cross-Validation (CV) procedure. Under the k-fold CV approach, the training set is split into k smaller sets, where a model is trained using k-1 of the folds as training data and validated on the remaining part.

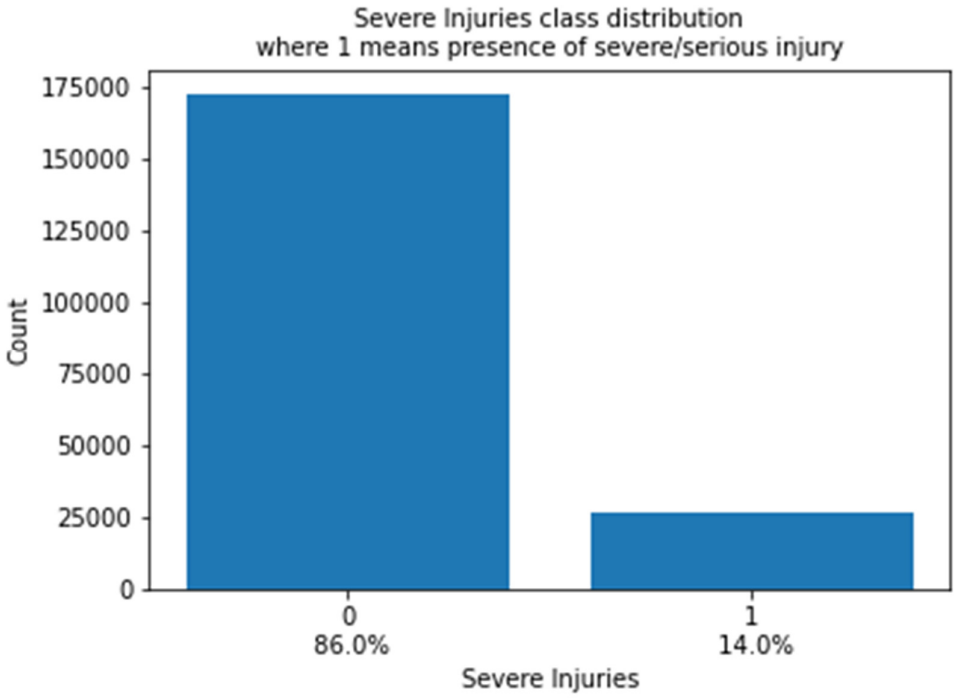
In our study, we evaluated the performance of each model in terms of model accuracy score and ROC-AUC score for both the training and test data. We also plotted the results to visually inspect the outcome. Based on our analysis, the Random Forest and Naive Bayes were the best performing models.

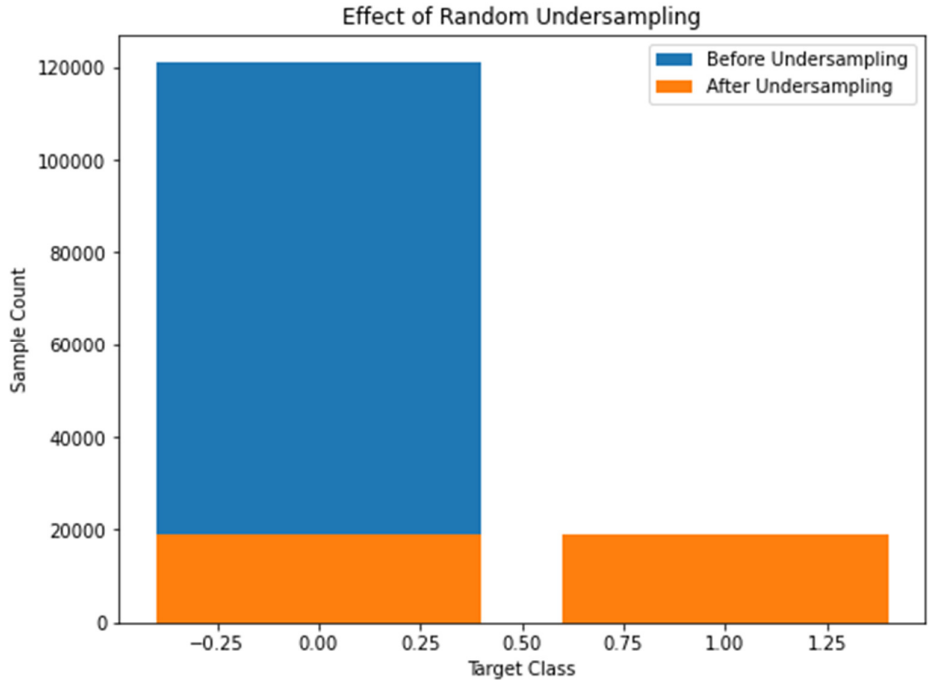
Imbalance data:

Since this particular case study deals with an imbalance among the classes, we will not be able to build useful models with the given dataset--without introducing additional interventions. One approach to deal with ICP is by either generating sythetic data (oversampling), or by generating a set of smaller "majority classes" by taking chunks from the original majority class (undersampling). In general, these approaches are collectively referred to as resampling.

For this case study, we performed the undersampling technique to balance the classes in our dataset. This will be achieved by removing some of the instances from the over-represented

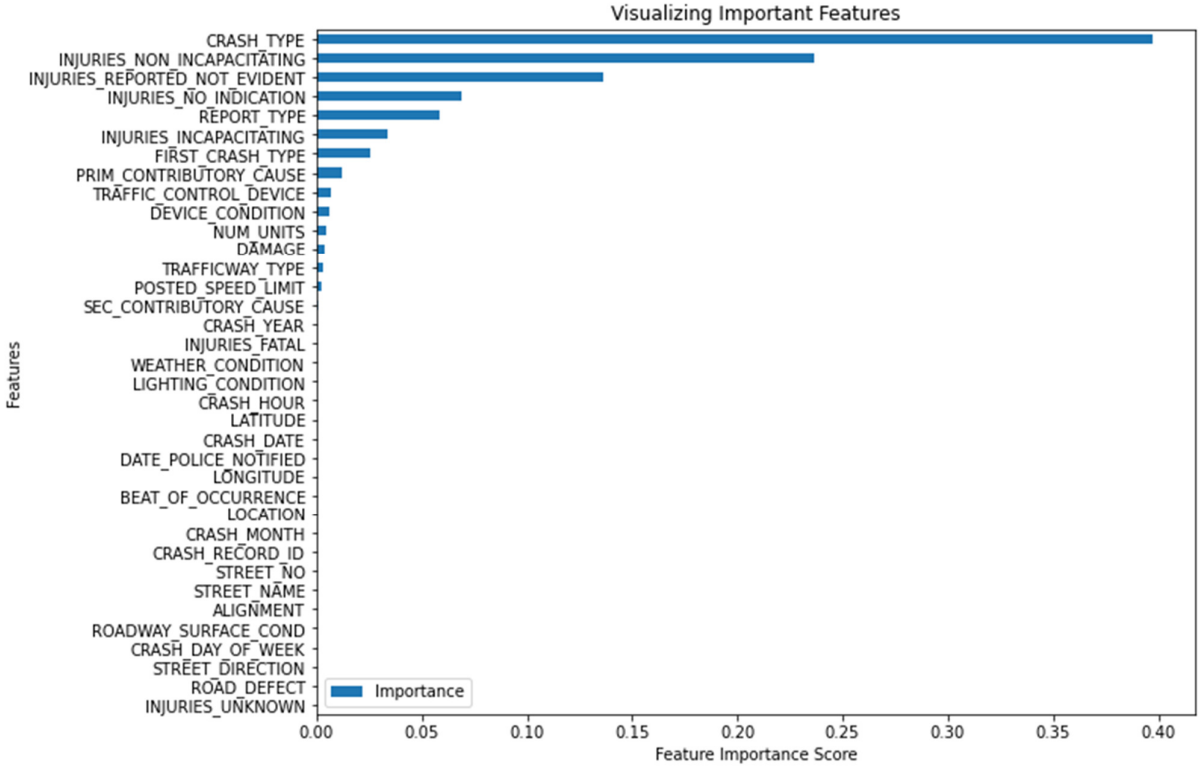
class (0, the absence of severe injuries) to match the number of instances in the under-represented class (1, the presence of severe injuries)





Feature Selection:

After conducting a feature selection analysis, I dropped the features "INJURIES_TOTAL", and "MOST_SEVERE_INJURY" to prevent overfitting and bias in the model training process.



Compare machine learning models:

The table below shows the model accuracy scores for the different algorithms on a our

Algorithm	Model accuracy score
KNN	0.999633
Random Forest	0.949183
Logistic Regression	1.000000
Gradient Boost	0.995367
Naive Bayes	0.999283

dataset. The accuracy score represents the proportion of correctly classified instances out of the total instances in the dataset. A score of 1.0 indicates perfect accuracy, while a score of 0.0 indicates no accuracy.

According to the table, the logistic regression model achieved the highest accuracy score of 1.0, indicating that it correctly classified all instances in the dataset. This suggests that the logistic regression model is a good fit for the data and could be useful for making predictions on new data.

The KNN and Naive Bayes models also achieved high accuracy scores of 0.999633 and 0.999283, respectively, suggesting that they are also good fits for the data.

The Random Forest and Gradient Boost models achieved lower accuracy scores of 0.949183 and 0.995367, respectively, suggesting that they may not be as good of a fit for the data as the other models. However, it's important to note that the performance of these models may vary depending on the specific dataset and problem at hand.

	Algorithm	ROC-AUC train score	ROC-AUC test score
0	KNN	0.987377	0.964990
1	Random Forest	0.999945	1.000000
2	Logistic Regression	1.000000	1.000000
3	Gradient Boost	0.999873	0.999938
4	Naive Bayes	1.000000	0.999971

The table above shows the ROC-AUC (Receiver Operating Characteristic - Area Under Curve) scores for the different algorithms we used. The ROC-AUC score is a measure of the performance of a classification model, with a score of 1.0 indicating perfect classification.

According to the table, the logistic regression model achieved the highest ROC-AUC scores of 1.0 on both the training and test sets. This indicates that the model has perfect discrimination ability in separating the positive and negative classes, and it can effectively distinguish between them. This suggests that the logistic regression model is a good fit for the data and could be useful for making predictions on new data.

The Naive Bayes and Random Forest models also achieved high ROC-AUC scores on both the training and test sets, suggesting that they are also good fits for the data.

The KNN and Gradient Boost models achieved slightly lower ROC-AUC scores on both the training and test sets, but they still performed well overall.

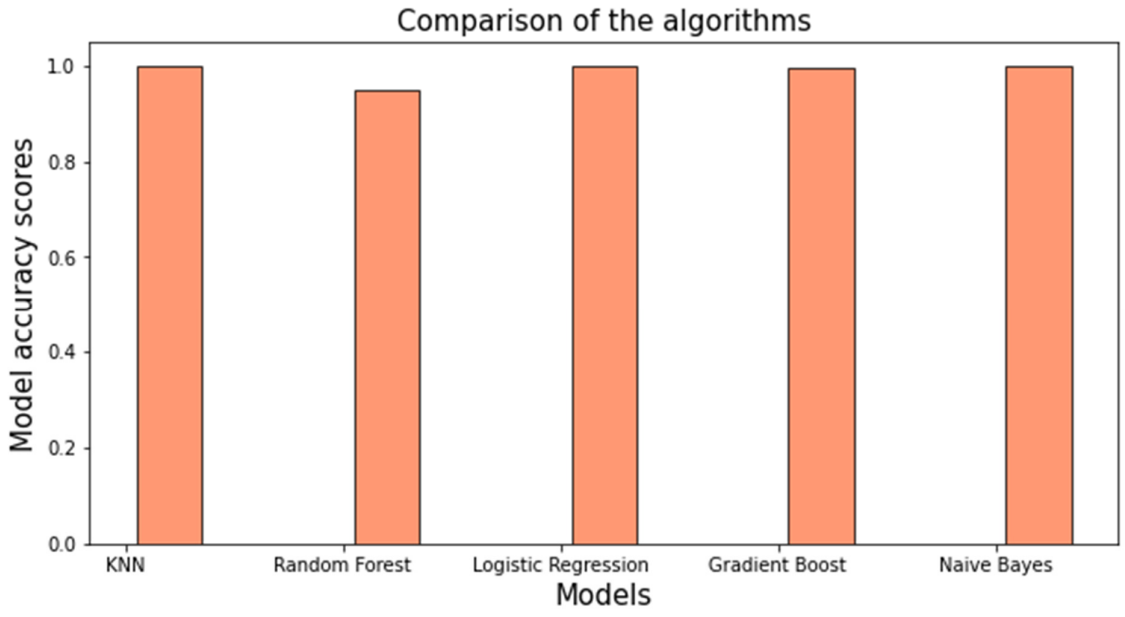
Overall, the table shows that all of the algorithms have good discrimination ability, but the logistic regression model achieved the highest scores, indicating that it may be the best choice for this particular dataset.

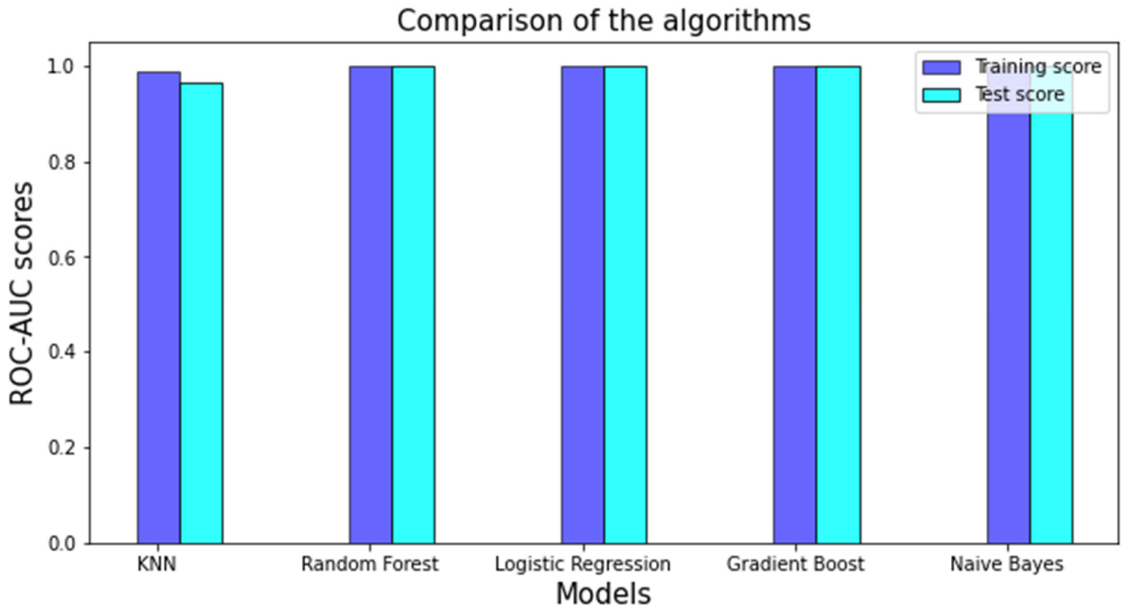
Note:

For the logistic regression model, the ROC-AUC score for both the training and testing sets is 1.0, it means that the logistic regression model is able to perfectly distinguish between the positive and negative classes in both datasets. This indicates that the model is overfitting the training data and may not generalize well to new, unseen data.

It should also be considered to use a separate dataset for model validation: Reserve a portion of the data as a validation set and do not use it for model training. Use this set to evaluate the model's performance and tune its parameters.

In the case of logic regression model, it is possible that the model is too complex or that there is some form of data leakage that is allowing the model to perform so well on both the training and testing sets. It is important to check the data for any errors or inconsistencies and to consider simplifying the model or using regularization techniques to prevent overfitting.





Future project/direction

In this case study, to deal with the imbalanced data, we used the imblearn.under_sampling library and implemented its techniques for under-sampling the majority class.

This helped to address the problem of imbalanced datasets by reducing the number of samples in the majority class, which at the same time helped to improve the performance of our models.

On the other hand, Under-sampling the majority class can lead to loss of information and may result in biased models. This is because random under-sampling can result in loss of important information that may be present in the majority class.

Because of the reasons mentioned above, the next step would be to over-sample the minority class to address the problem of imbalanced datasets.This technique involves generating new synthetic samples for the minority class, which can be used to balance the dataset.

Ultimately, Implementing both techniques will give us a better understanding of which model performance best.

Additionally, it is important to evaluate the model's performance on a separate validation set or with cross-validation to get a better estimate of its generalization performance.