Road Accident Severity Classification

Karnik Ketan Kalani
Department of Applied Data Science, San Jose State University
DATA 240 : Data Mining
Dr. Seungjoon Lee
May 19, 2023

Introduction

Road accidents are a major public safety concern that results in injuries, fatalities, and disability on a daily basis. According to the World Health Organization, approximately 1.35 million people die each year from road accidents, with an additional 50 million injured. These accidents affect not only the individuals involved but also their families and the broader community. To prevent such tragedies and reduce the economic impact associated with accidents, measures must be taken to improve road safety. Improving road infrastructure, increasing the number of traffic personnel, and raising awareness about the importance of wearing helmets and seatbelts are just a few examples of measures that can be implemented to promote road safety. However, predicting the severity of road accidents can also be crucial in reducing their impact.

In this project, we will develop a machine-learning model to predict the severity of road accidents. Specifically, we will use classification machine learning models, including Decision Trees, Random Forest Classifiers, and Naive Bayes, to predict accident severity and categorize accidents as Fatal, Serious, or Slight. In addition, we will perform feature importance analysis to select the most valuable features and evaluate model performance using metrics such as Accuracy, F1 Score, Roc-Auc Curve, and Confusion Matrix.
While this project has the potential to contribute to the field of road safety significantly, it is essential to acknowledge potential limitations and challenges. For example, data quality issues and biases could impact the model's accuracy. Nonetheless, by addressing these challenges and implementing measures to improve road safety, we can work towards reducing the devastating impact of road accidents.

Dataset Description

This dataset has a total of three different csv files which are named as AccidentBig, CasualitiesBig and VehiclesBig which are of sizes 231.48 MB , 100.94 MB, 194.07 MB respectively. The data is originally from data.gov.uk. It has records of all the accidents from 2005-2015. All the data values are in numerical format however, description of each numerical value is provided in the supporting document. Each of these files has 31,21,14 columns respectively. The final Target Variable which predicts the model if it is prone to accident or not is Accident_Severity.

VehiclesBig.CSV

Name	Description
Vehicle_Type	Types of vehicle for ex: motorcycle, cycle, car etc.
Vehicle_Manoeuvre	Description of what was doing during the accident For ex: Reversing, parked, etc.
Age_of_Vehicle	Age of the vehicle
First_Point_of_Impact	Point of contact of accidents For ex: Front, Back, etc.
Age_of_Driver	Age of Driver

CasualitiesBig.CSV

Name	Description
Age_of_Casualty	Age of the person who got affected by the event
Sex_of_Casualty	Gender of the person affected by the incident
Bus_or_Coach_Passenger	Tells if the passenger travels by car or coach
Casualty_Class	Describes if a person injured is driver or passenger
age_band_casualty	Tells about the age category the person injured falls into. For ex: 0-5,10-25, etc.

AccidentsBig.CSV

Name	Description
Weather_Conditions	It describes about the weather conditions such as Raining, snowing, Fog or Mist
Accident_Year	The occurrence year of the accident
Road_Type	It gives information about the one way street, Dual carriageway

Vehicle_Type	It has cars, motorcycle, Bus etc
Day_of_Week	It gives information about the particular day in a week.

*Note : These csv files have over 60 plus columns in total. Hence, only a few attributes are mentioned.

2. Why use Classification Methods for Road Accidents

The use of classification methods can help in analyzing road accidents to identify patterns and trends in the data. It can also help in preventing accidents from happening. For suppose, through classification we can know which road or route is most prone to accidents. It can help in having more traffic personnel to guard that area, and have more traffic sign boards indicating to slow down or sometimes improving the road condition by reconstructing them.This analysis can help in comprehending the causes and contributing factors of accidents. By categorizing road accidents based on characteristics such as severity, vehicle type, road conditions, and time of day, strategies and policies can be developed to reduce accidents, enhance road safety, and allocate resources effectively. Classification methods can also be utilized to predict the probability of future accidents based on historical data. This can assist in developing early warning systems and identifying high-risk areas. Overall, the use of classification methods in analyzing road accidents can provide valuable insights for improving road safety and decreasing the number of accidents.

3. Literature Review

(Hala et al,2021) discusses the application of machine learning techniques in forecasting the seriousness of traffic accidents. The authors suggest various models that can be trained on previous accident data to predict the severity of accidents based on factors such as weather and road conditions, as well as time of day. The models used in this paper are Gaussian NB,K-neighbors classifier, Decision tree, SVM, and Multi-layer perceptron. The F1 scores are as follows: 0.48, 0.91, 0.85, 0.91, 0.94.The study finds that machine learning techniques such as random forests and support vector machines are more effective than traditional statistical methods in predicting the severity of accidents. Additionally, the authors identify weather conditions and time of day as important predictors of accident severity. Overall, the study highlights the potential of machine learning methods to enhance road safety by predicting accident severity and guiding the development of targeted interventions.

(Ahmed et al,2021) presents a comparative analysis of various machine learning algorithms used for predicting the severity of road accidents. The study aims to find the most efficient algorithm by considering factors such as weather, vehicle type, and road conditions. The research utilized a dataset of road accidents that took place in the UK from 2005 to 2015.Different algorithms, including decision trees, random forests, support vector machines, and neural networks, were evaluated for their accuracy, precision, and recall. The findings revealed that the random forest algorithm performed the best among all algorithms. Additionally, the study identified essential factors that significantly influence accident severity, including the driver's age, vehicle type, and road type. In conclusion, the research provides significant insights

into the effectiveness of machine learning algorithms in predicting road accident severity. It highlights the random forest algorithm's superiority and identifies significant factors that impact accident severity.

(Paul et al, 2020) proposed a multiclass model in which they combined accident prediction and severity to develop a better model for avoiding road collisions using The National Traffic Accident Report 2007, BRTA by implementing various machine learning methods such as Decision Tree, Random Forest, Multilayer Perceptron, and Categorical Naive Bayes all produced acceptable results, but Decision Tree produced the best. Repeatedly evaluating the test dataset yielded the desired prediction model. With an F1 score of 98.68% and 99.80%, this algorithm achieved a high accuracy of 99.77% for accident prediction and 99.80% for severity prediction. Decision performed the best.

According to research by (Thaninthorn & Watchareewan 2022) the ministry of transportation catalog data was chosen which had about 8500 records and about 40 columns.The data was split into 70 percent for train dataset 30 percent for test dataset. After refining through all the attributes Thaninthorn and Watchareewan (2022) found 11 attributes to be most important and used them eventually. After all the preprocessing and data modeling the study evaluated performance models based on accuracy, precision, recall, and f-measure. The comparative results revealed that the accuracy of RF is the best for predicting road deaths on the road network, with an accuracy of 89%, precision of 0.86, recall of 0.89, and f-measure of 0.85.

The study by (Shanthi & Geetha Ramani, 2021) categorizes the accidents based on severity and evaluates the accuracy of various characteristics in predicting the severity of road traffic accidents. The most relevant features were chosen using the Correlation-based Feature Subset (CFS), Fast Correlation Based Filter (FCBF), and Mutual Information Feature Selector (MIFS), while various Decision Trees (DT), Naive Bayes (NB), and Random Forest (RF) algorithms were used to classify the data. The research used a dataset by Fatality Analysis Reporting System (FARS), which has more than 457549 entries from 56 states of the U.S, with 33 attributes each. The random forest Classifier method did better than other algorithms, achieving an accuracy of 99.73% with 0.27% misclassification rate.

According to (Bulbul et al., 2016), in their analysis of road accidents, they have utilized algorithms such as Decision Trees, Naive Bayes, and OneR. The dataset was obtained from Traffic Insurance Information Center (TRAMER).As a benchmark for algorithms, ROC Area value, Precision, Recall, F-Criterion, Kappa Statistics, and Accuracy were used. According to the research, Naive Bayes performed 80.2%, CART performed 81.5%, and Ibk performed 81.3% when it came to forecasting the probability of traffic accidents in Istanbul. These findings imply that these machine learning methods are useful for forecasting traffic accidents.

Data Preprocessing

The merged dataset has 4427649 columns and 66 rows. After an inspection of the data and going through the metadata it found that data had ‘-1’ values filled whenever there was no data available. Thus we replaced all the cells containing ‘-1’ with nulls. The null percentage of the dataset was calculated and the features with more than 30% null values were dropped from the dataset. The columns Pedestrian_Road_Maintenance_Worker, 2nd_Road_Class,

Junction_Control are dropped from the dataset. The Age_Band_of_Driver and Age_Band_of_Casualty column were dropped as Age_of_Driver and Age_of_Casuality column were already present.

Based on domain knowledge and research the Journey_Purpose_of_Driver, and Was_Vehicle_Left_Hand_Drive columns were dropped as it had no significance with the problem statement. Vehicle_Reference_y and Vehicle_Reference_x were dropped as they were a unique value for each vehicle in a singular accident which again did not seem important for the goal of this project. Following that data imputations were performed. The Null values in the categorical features were filled using the mode of their respective columns while numerical columns were filled using median. For dealing with the outliers the first quartile (Q1), third quartile (Q3), and interquartile range (IQR) of the columns are calculated using the quantile() function.Then calculated the upper and lower bounds for the outliers and created a boolean mask “outliers" to to identify the rows where the 'Age_of_Vehicle' column has values below the lower bound or above the upper bound.It counts the number of outliers using the sum() function on the 'outliers' mask. The outliers in the other features are removed by following the same procedure. The Duplicate NaN values are dropped from the dataset.

Figure 1

Before Removing Outliers

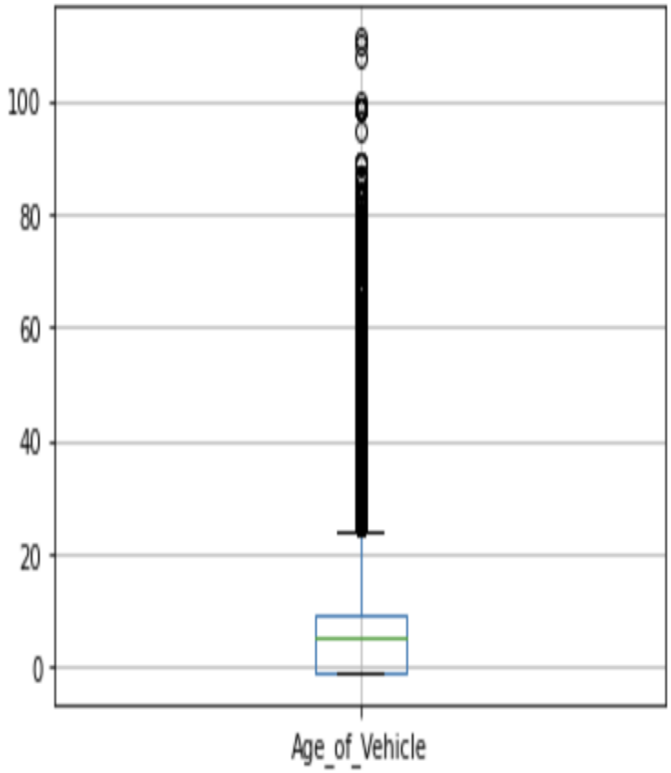
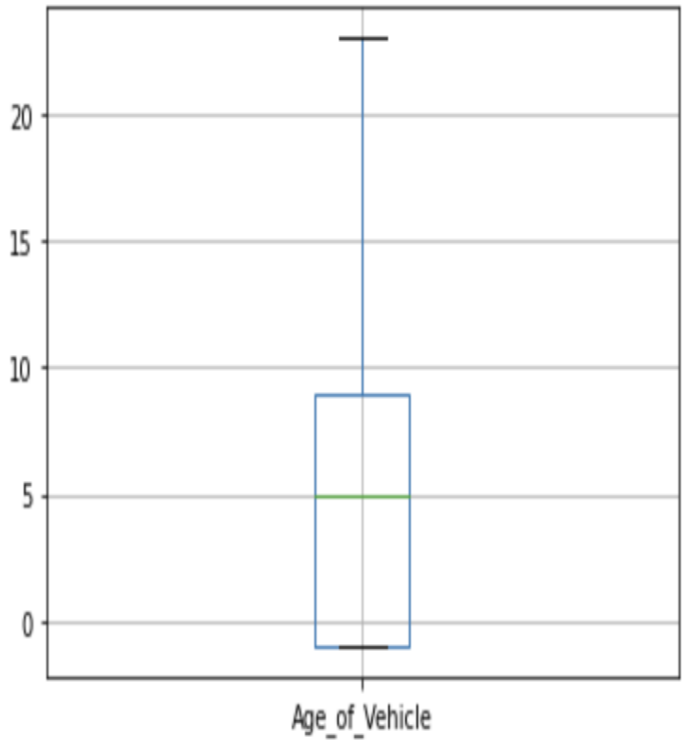


Figure 2

After Removing the Outliers

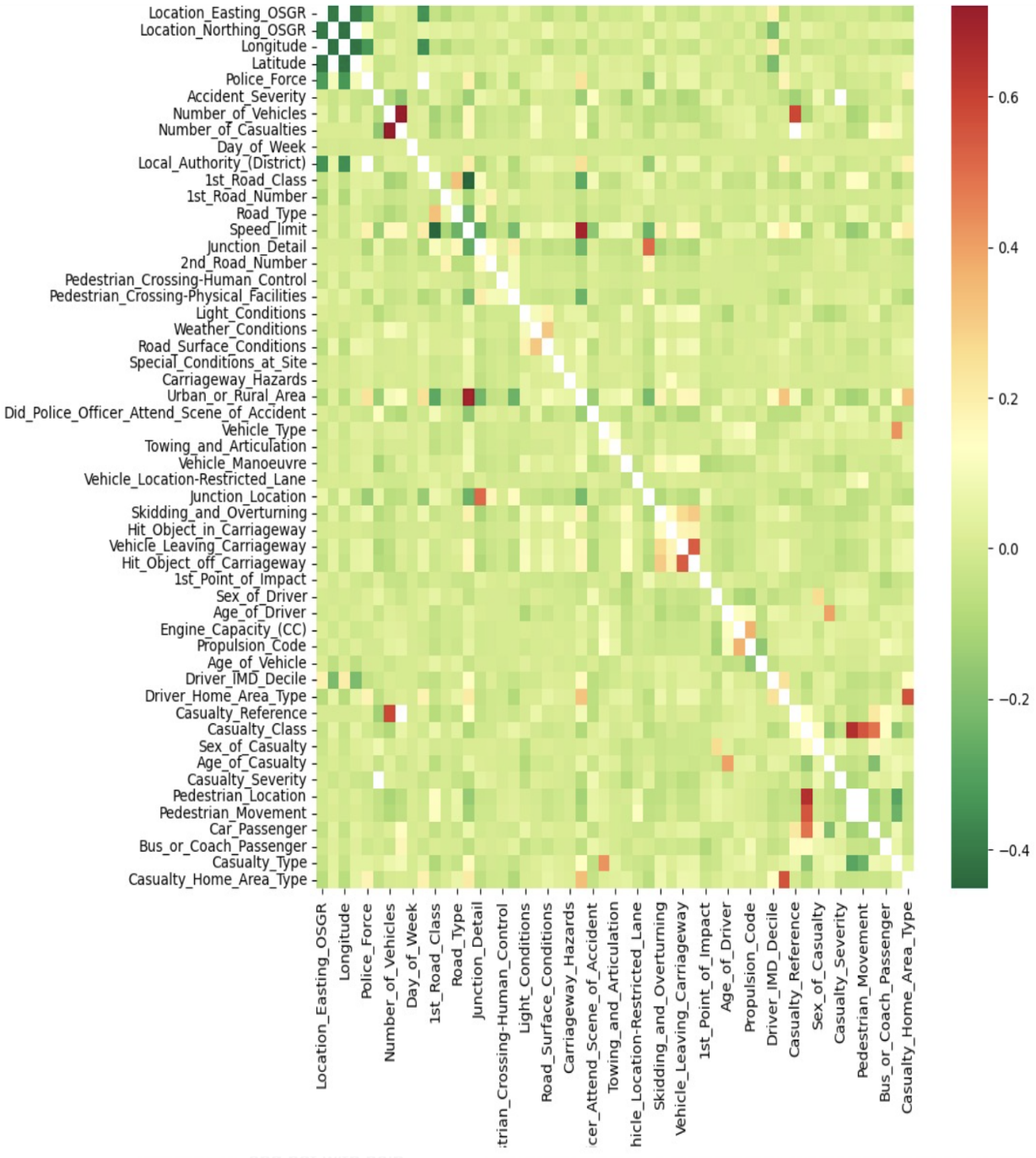


Data Transformation

Although most of our data was already label encoded, we still had to perform some data transformations. Date was converted into datetime objects from mm/dd/yyyy format. Following that month and year were extracted and saved into new columns and the old datetime column was dropped. Similarly hours were extracted in a 24 hour format from the time column. The columns Latitude, Longitude, local_authourity_district were highly correlated as seen in the figure 3. Hence, latitude and longitude were dropped. The values in the target feature Severity_index have many imbalances. These values are filled by using the Random Under Sampling method.

Figure 3

Correlation Matrix of the Data

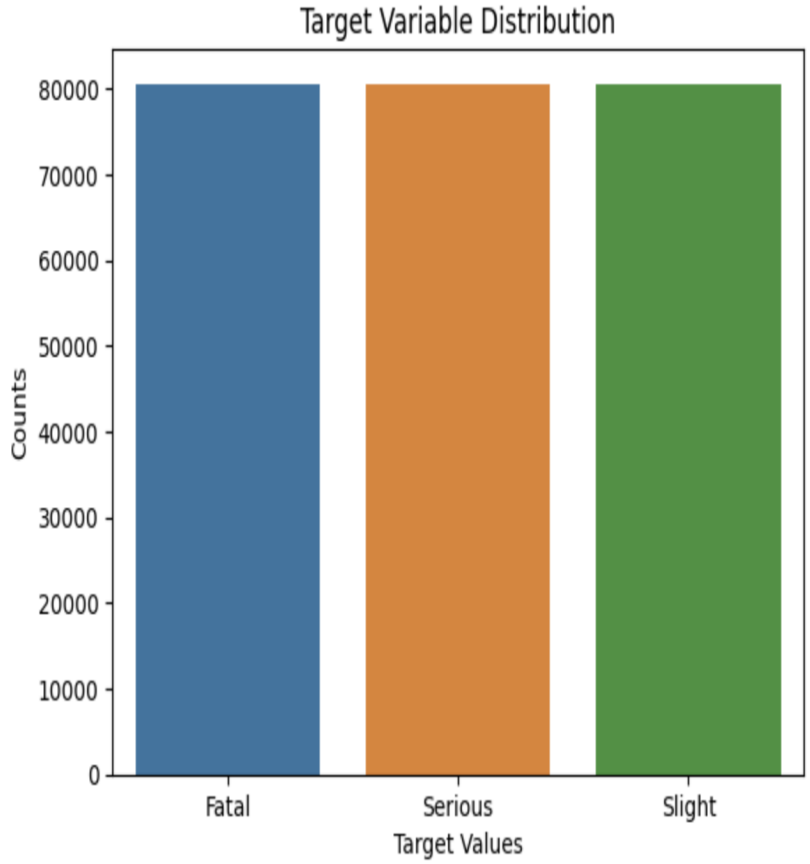


Min-Max scaling technique is used to scale the numerical features to a specific range 0 and 1. The UK dataset contains features with different measurement units, such as hour, year, and some few other categorical features as well. By applying min-max scaling, all features are

transformed to a standardized range, removing the differences in measurement units. After that the target feature of our dataset ‘accidents_severity’ had imbalance. Hence it was handled using random undersampling as we can see in the figure 4.

Figure 4

Target Feature Distribution after Random Undersampling



As mentioned earlier, most of our data was already label encoded and few other columns such as date and time were handled during data transformations. Finally, label encoding was performed to transform the 'LSOA_of_Accident_Location' and 'Local_Authority_(Highway)' columns from their original categorical values to numerical labels in order to make it interpretable for the machine learning model. Finally the dataset was split into train and test in the ratio 80:20 as shown in the figure 5.

Figure 5

Label Encoding And Data Splitting

```
from sklearn.preprocessing import LabelEncoder

# create a LabelEncoder object
le = LabelEncoder()

# fit and transform the object datatype column in the dataframe
cleaned_data['LSOA_of_Accident_Location'] = le.fit_transform(cleaned_data['LSOA_of_Accident_Location'])
cleaned_data['Local_Authority_(Highway)'] = le.fit_transform(cleaned_data['Local_Authority_(Highway)'])
```

```
# Splitting data into train and test sets
X_train, X_test, y_train, y_test = train_test_split(X_res, y_res, test_size=0.2, random_state=42)
```

Feature Selection

One of the main challenges of feature selection is determining which features are truly relevant to the outcome being predicted. It can be time-consuming and computationally expensive, especially for large datasets or complex models. Furthermore, while removing irrelevant features can reduce overfitting and improve accuracy, it is not a guarantee. Overfitting can still occur if the model is not properly tuned or if the feature selection process is not rigorous enough. Despite these challenges, feature selection has many advantages. It can significantly reduce the complexity of the model, making it easier to interpret and understand. It also reduces the computational time required to train the model, especially for large datasets. Moreover, selecting only the most relevant features can improve the accuracy of the model by focusing on the important aspects of the data.

The dataset contains 57 features after transforming the data. In order to find the best features among those, we use four types of feature selection:

1. Anova test
2. Chi-squared
3. Random Forest
4. Logistic regression.

1. Anova Test

To use ANOVA for feature selection in a road accident severity classification problem, we first need to encode the categorical variables as numerical features. Then we need to split the data into training and testing sets. Next, we need to perform the ANOVA test on each feature to determine the F-value and the corresponding p-value. Feature selection using Anova would help to identify which features have a significant impact on the "Accident_Severity " and should be included in the model. ANOVA is best suited for continuous predictor variables and categorical target variables.The F-value is used to determine the significance of the feature, while the p-value is used to determine the level of significance. Top 15 features given by the algorithm can be seen in Figure 6.

Figure 6

Top 15 Features by ANOVA

```
# Apply ANOVA to get F-scores and p-values for each feature
f_scores, p_values = f_classif(X_res, y_res)

# Create a DataFrame with feature names, F-scores, and p-values
anova_results = pd.DataFrame({'feature': X.columns, 'F-score': f_scores, 'p-value': p_values})

# Sort the DataFrame by F-score in descending order
anova_results = anova_results.sort_values('F-score', ascending=False)

# Print the top 15 features with highest F-scores
print(anova_results.head(15))
```

	feature	F-score	p-value
46	Casualty_Severity	89804.533300	0.0
12	Speed_limit	14731.762327	0.0
22	Urban_or_Rural_Area	12753.854820	0.0
23	Did_Police_Officer_Attend_Scene_of_Accident	8095.342602	0.0
27	Vehicle_Manoeuvre	5465.905830	0.0
9	1st_Road_Class	3848.771027	0.0
29	Junction_Location	3698.934785	0.0
13	Junction_Detail	3529.043288	0.0
4	Number_of_Casualties	3269.026282	0.0
32	Vehicle_Leaving_Carriageway	3168.233141	0.0
17	Light_Conditions	2994.139368	0.0
33	Hit_Object_off_Carriageway	2326.492588	0.0
30	Skidding_and_Overturning	2289.120226	0.0
42	Casualty_Reference	2240.736804	0.0
44	Sex_of_Casualty	2073.254726	0.0

2. Chi-Square Test

The chi-square test is a statistical test used to determine the independence of two categorical variables. In the context of feature selection for classification models, the chi-square test can be used to identify which features are most likely to be associated with the target variable. By calculating the chi-square statistic between each feature and the target variable, we can determine the strength of association between the two variables. Features with higher

chi-square statistics and lower p-values are considered more strongly associated with the target variable and are therefore more likely to be useful in predicting it. Using the chi-square test in feature selection helps to identify the most relevant and informative features for the classification model, leading to better performance and more accurate predictions. The Top 15 features based on the chi-square scores as shown in the Figure 7 below.

Figure 7

Top 15 Features by Chi-Square

```
from sklearn.feature_selection import SelectKBest, chi2
# Select the top 15 features based on chi-square test
selector = SelectKBest(chi2, k=15)
selector.fit(X_res, y_res)
# Get the indices of the selected features
feature_indices = selector.get_support(indices=True)
# Get the names of the selected features
selected_features = list(X.columns[feature_indices])
# Get the chi-square scores and p-values for all features
scores = selector.scores_
p_values = selector.pvalues_
# Create a DataFrame with feature names, chi-square scores, and p-values
chi2_results = pd.DataFrame({'feature': X.columns, 'chi2-score': scores, 'p-value': p_values})
# Sort the DataFrame by chi-square scores in descending order
chi2_results = chi2_results.sort_values('chi2-score', ascending=False)
# Print the top 10 features with highest chi-square scores
print(chi2_results.head(15))
```

	feature	chi2-score	p-value
46	Casualty_Severity	21562.112132	0.0
23	Did_Police_Officer_Attend_Scene_of_Accident	7068.618337	0.0
22	Urban_or_Rural_Area	5219.310751	0.0
29	Junction_Location	4046.594809	0.0
32	Vehicle_Leaving_Carriageway	3822.887420	0.0
12	Speed_limit	3120.328119	0.0
33	Hit_Object_off_Carriageway	2982.142238	0.0
17	Light_Conditions	2824.564576	0.0
44	Sex_of_Casualty	2627.342966	0.0
13	Junction_Detail	2486.847687	0.0
30	Skidding_and_Overturning	2111.771641	0.0
41	Driver_Home_Area_Type	2015.826478	0.0
35	Sex_of_Driver	1776.268487	0.0
52	Casualty_Home_Area_Type	1611.983420	0.0
27	Vehicle_Manoeuvre	1524.775854	0.0

3. Random Forest

Feature selection using random forest is a popular method for identifying the most important features in a dataset. Random forests are a type of ensemble learning method that constructs a multitude of decision trees during training and outputs the class that is the mode of the classes of the individual trees. One of the main advantages of random forest is that it can

provide feature importance scores, which can help identify the most important features that contribute to the prediction. To extract feature importance scores from a random forest model, you can use the feature_importances attribute of the trained model. This attribute is an array that contains the importance score for each feature in the input data. The Top 15 features based on importance scores as shown in Figure 8 below.

Figure 8

Top 15 Features by Random Forest

```
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X_res, y_res, test_size=0.2, random_state=42)
# Create the Random Forest model
rf = RandomForestClassifier()
# Train the model on the training data
rf.fit(X_train, y_train)
# Use feature_importances_ to get feature importance scores
importances = rf.feature_importances_
# Create a DataFrame with feature names and importance scores
feature_importances = pd.DataFrame({'feature': X.columns, 'importance': importances})
# Sort the DataFrame by importance score in descending order
feature_importances = feature_importances.sort_values('importance', ascending=False)
# Print the top 15 features with highest importance scores
print(feature_importances.head(15))
```

```
                       feature  importance
46           Casualty_Severity    0.356732
4          Number_of_Casualties    0.054854
12                 Speed_limit    0.025411
42           Casualty_Reference    0.025024
0         Location_Easting_OSGR    0.023502
1        Location_Northing_OSGR    0.023439
45             Age_of_Casualty    0.021884
24   LSOA_of_Accident_Location    0.021749
10             1st_Road_Number    0.021687
6                         Time    0.021093
7   Local_Authority_(District)    0.020309
55                         day    0.020124
8    Local_Authority_(Highway)    0.019207
36               Age_of_Driver    0.018729
51               Casualty_Type    0.018711
```

4. Feature Selection using Logistic Regression

Feature selection using logistic regression is the process of selecting a subset of relevant features from a larger set of features for use in a logistic regression model. The goal of feature selection is to improve the model's performance by reducing the dimensionality of the input data and removing irrelevant or redundant features that can negatively impact the model's accuracy. One approach to feature selection using logistic regression is to use a backward elimination method. This involves starting with a model that includes all of the features, and then iteratively removing the feature with the highest p-value until a stopping criterion is met. The p-value is a statistical measure that indicates the probability that the coefficient for a feature is zero, which means that the feature has no effect on the outcome variable. The Top 15 features are from the results shown in Figure 9.

Figure 9

Top 15 Features by Logistic Regression

```
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X_res, y_res, test_size=0.2, random_state=42)
# Create the Logistic Regression model
lr = LogisticRegression()
# Fit the model on the training data
lr.fit(X_train, y_train)
# Use the SelectFromModel method to select the top 10 features based on the logistic regression coefficients
sfm = SelectFromModel(lr, threshold='median', max_features=15)
sfm.fit(X_train, y_train)
# Get the indices of the selected features
feature_indices = sfm.get_support(indices=True)
# Get the names of the selected features
selected_features = list(X.columns[feature_indices])
# Create a DataFrame with the selected feature names and their coefficients
lr_results = pd.DataFrame({'feature': selected_features, 'coefficient': abs(lr.coef_[0][feature_indices])})
# Sort the DataFrame by coefficient in descending order
lr_results = lr_results.sort_values('coefficient', ascending=False)
# Print the top 15 features with highest coefficients
print(lr_results.head(15))
```

```
                                    feature  coefficient
3                         Number_of_Casualties    15.241869
11                           Casualty_Severity    11.549108
2                           Number_of_Vehicles     3.849388
14                               Casualty_Type     2.731317
7   Did_Police_Officer_Attend_Scene_of_Accident     2.667354
1                                Police_Force     1.464640
4                  Local_Authority_(District)     1.417720
6                                 Speed_limit     1.324304
13                       Pedestrian_Movement     1.221582
10                        Casualty_Reference     1.215714
12                       Pedestrian_Location     1.143325
9                            Propulsion_Code     1.068117
5                                  Road_Type     1.052924
8                              Vehicle_Type     0.990605
0                  Location_Northing_OSGR     0.615067
```

Feature Selection Analysis

After the feature analysis of the four models we can see that all the four models had a few features in common like Number Of Casualties, Casualty_Severity, Did_Polics_Attend_The_Scene_Of_Accident and so on. Chi square and Anova got similar results .One reason could be because even though they focus on categorical and continuous variables respectively they focus mainly on the relationship between each feature and target variable. Random Forest and Logistic regression have only a few results in common ,could be because random forest captures non linearity and interactions that the others do not consider. Logistic regression captures features which are not too complex and are linear.

Modeling

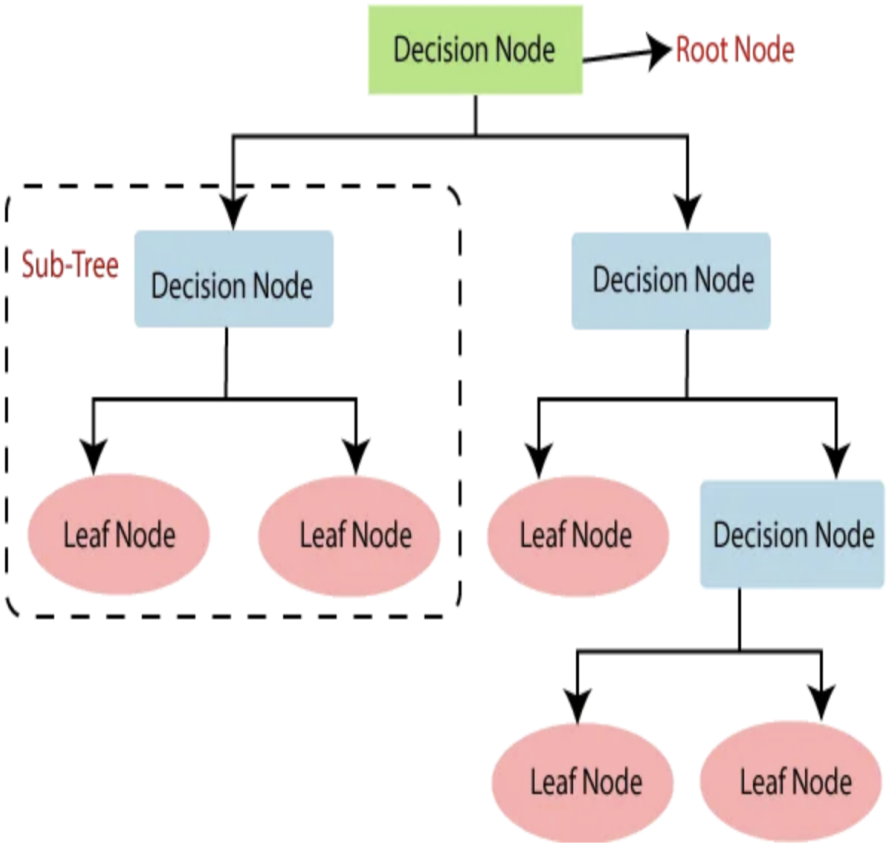
Decision Trees

Decision Trees algorithm is a supervised classification algorithm. The decision trees work in such a way that they construct the tree by placing the feature with highest priority at the root node and then follow next decisions based on the root node. There are specific metrics like Gini and entropy which measure the importance of the node (Severity_index) and place them in descending order of priority. Figure 10 represents the structure of the decision tree. The input

features include factors such as weather conditions, road conditions, vehicle type, and driver behavior, while the target variable is the severity of the accident (e.g.Slight,Serious,fatal).

Figure 10

Decision Tree Structure



The ID3 decision tree method has been implemented for our project. Based on entropy, this method measures the importance of nodes and places them accordingly. Using the max_depth as 10, we are pruning the decision tree after 10 levels of depth and evaluating its performance.
The classification_report() function generates a report that includes precision, recall, and F1-score metrics for each class. The confusion_matrix() function generates a confusion matrix that shows the number of true positives, true negatives, false positives, and false negatives for each class. The accuracy_score() function calculates the accuracy of the model on the test set. The Accuracy score of the Decision tree classifier is 0.8358. Detailed results of the decision tree model before feature selection can be seen in Figure 11.

Figure 11

Results of Decision Tree Model before Feature Selection

```
# Splitting data into train and test sets
X_train, X_test, y_train, y_test = train_test_split(X_res, y_res, test_size=0.2, random_state=42)

# Creating and fitting the decision tree model
dtc = DecisionTreeClassifier()
dtc.fit(X_train, y_train)

# Predicting on the test set
y_pred = dtc.predict(X_test)

# Evaluating the model
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score

print('Confusion Matrix:\n', confusion_matrix(y_test, y_pred))
print('\nClassification Report:\n', classification_report(y_test, y_pred))
print('\nAccuracy:', accuracy_score(y_test, y_pred))
```

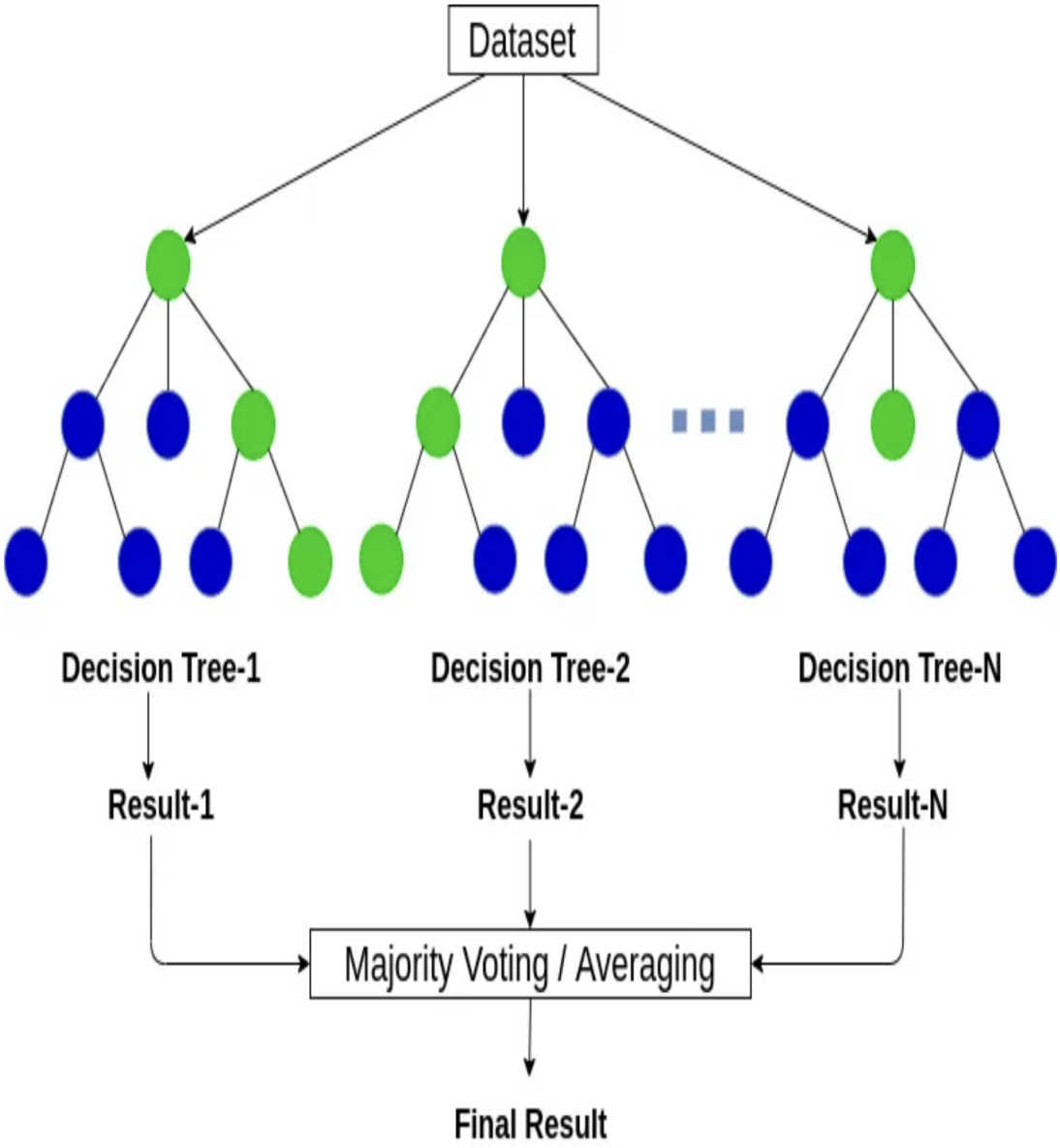
```
Confusion Matrix:
 [[14105  1275   556]
 [ 1777 12601  1581]
 [  857  1894 13712]]

Classification Report:
              precision    recall  f1-score   support

           1       0.84      0.89      0.86     15936
           2       0.80      0.79      0.79     15959
           3       0.87      0.83      0.85     16463

    accuracy                           0.84     48358
   macro avg       0.84      0.84      0.84     48358
weighted avg       0.84      0.84      0.84     48358


Accuracy: 0.835807932503412
```

Random Forest

Random Forest is a supervised classification algorithm which can be used for both classification and regression problems. The Random Forest algorithm works by constructing various numbers of decision trees by using different subsets of the given data and taking the average of the predicted value of all the trees. Instead of relying on one tree, the random forest algorithm considers the output of all the decision trees that are constructed on various subsets of the dataset and then predicts the mean of the majority prediction as the output. The structure of random forest is shown below in Figure 12

Figure 12

Structure of Random Forest Model



During the implementation of our project, we used a random forest with default parameters and changed a few parameters from the default parameter set. When different sets of parameters are considered, there is a difference in the results. As this is a binary classification problem, we used random forest since it outperforms many datasets due to the way it is implemented. Results achieved by the random forest model before feature selection can be seen in Figure 13.

Figure 13

Results of Random Forest Model before Feature Selection

```
from sklearn.ensemble import RandomForestClassifier

# Split the data into train and test sets
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X_res, y_res, test_size=0.3, random_state=42)

# Train the Random Forest classifier
rfc = RandomForestClassifier(n_estimators=100, random_state=42)
rfc.fit(X_train, y_train)

# Predict on the test set
y_pred_rfc = rfc.predict(X_test)

# Evaluate the Random Forest classifier
from sklearn.metrics import classification_report, accuracy_score

print('Random Forest Classification Report:')
print(classification_report(y_test, y_pred_rfc))
print('Accuracy:', accuracy_score(y_test, y_pred_rfc))
```

```
Random Forest Classification Report:
              precision    recall  f1-score   support

           1       0.92      0.92      0.92     24060
           2       0.89      0.82      0.86     24048
           3       0.87      0.92      0.89     24429

    accuracy                           0.89     72537
   macro avg       0.89      0.89      0.89     72537
weighted avg       0.89      0.89      0.89     72537

Accuracy: 0.8911452086521362
```

Gaussian Naive Bayes Classification

The main principle of the Gaussian Naive Bayes classifier is to use Bayes theorem to compute the probability of each class label given the input features, assuming that the input features follow a Gaussian (i.e., normal) distribution and are independent of each other.When handling real-time data with continuous distribution, Naive Bayes classifier considers that the big data is generated through a Gaussian process with normal distribution.

Figure 14

Formula for Gaussian Naive Bayes

$$P(X|Y=c) = \frac{1}{\sqrt{2\pi\sigma_c^2}} e^{\frac{-(x-\mu_c)^2}{2\sigma_c^2}}$$

To evaluate the performance of our algorithms, we have implemented a number of metrics. It is possible to evaluate classification algorithms based on confusion matrix, precision, recall, and accuracy metrics. Following is a comparison of all the metrics for various algorithms.

Model Comparison before Feature Selection

All the models were trained on all 57 columns that were left after cleaning. In case of Decision Tree and Random Forest ML models, Both models have been run on the same

parameters which is ‘Random_state=42’. While no parameters have been changed in case of Gaussian Naive Bayes. All of these models have been run in the same environment to be able to compare their performance accurately.

Table 1

Results before feature selection

Model	Recall	Precision	Accuracy	F1-score
Decision Tree	0.84	0.84	0.835	0.84
Random Forest	0.90	0.90	0.896	0.90
Gaussian Naive Bayes	0.69	0.72	0.686	0.66

The Decision Tree model has an accuracy of 0.835, a recall of 0.84, and a precision of 0.84. The F1-score is also 0.84, which is a harmonic mean of recall and accuracy.With an accuracy of 0.896, the Random Forest model outperformed the others. It had a greater recall and accuracy of 0.90, indicating that it was better at properly detecting positive samples.
The Gaussian Naive Bayes model, on the other hand, performed the worst of the three models. It was 0.686 accurate, with a recall of 0.69 and precision of 0.72. This model has an F1-score of 0.66. In terms of accuracy, recall, precision, and F1-score, the Random Forest model beats the Decision Tree and Gaussian Naive Bayes models.

Model Comparison after Feature Selection

Table 2

Results after feature selection

Model	Recall	Precision	Accuracy	F1-score
Decision Tree	0.85	0.85	0.846	0.85
Random Forest	0.86	0.86	0.862	0.86
Gaussian Naive Bayes	0.68	0.72	0.686	0.66

After feature selection, the Decision Tree model's recall, precision, accuracy, and F1-score all increased marginally. The recall rose from 0.84 to 0.85, the precision from 0.84 to 0.85, the accuracy from 0.835 to 0.846, and the F1-score from 0.84 to 0.85. In contrast, there was a modest decrease in recall, precision, accuracy, and F1-score following feature selection for the Random Forest model. The recall rose from 0.90 to 0.86, the precision rose from 0.90 to 0.86, the accuracy rose from 0.896 to 0.862, and the F1-score rose from 0.90 to 0.86. The performance of the Gaussian Naive Bayes model, on the other hand, did not change much following feature

selection. The recall stayed constant at 0.68, the precision remained constant at 0.72, the accuracy remained constant at 0.686, and the F1-score remained constant at 0.66. However this loss of performance is easily justified considering that these results are obtained only from 15 features instead of 57 for the previous one.The main reason why Gaussian Naive Bayes did not perform well could be because this model assumes that all features are independent of each other. But like we can see in real life it is not always true. A few can have highly positive correlation and a few can have highly negative correlation which proves that assumption is wrong which is why it did not perform well for the road accident severity classification task.

Hyperparameter Tuning

In our project, hyperparameter tuning was a critical step in enhancing our model's performance. As Random forest was the best performer we decided to perform the hyperparameter tuning on random forest in order to increase the accuracy of the model. The Optuna package, a strong tool for hyperparameter tuning, was used. We were able to discover the optimal set of hyperparameters for our Random Forest model by using Optuna. The improved settings, as indicated in the picture, yielded an outstanding 90% accuracy. Optuna uses a dynamic technique to choose the most effective optimization strategy for the given dataset, taking into account both grid search and random search. It extensively traverses the hyperparameter space to find the best combination of parameters. Based on considerable experimentation and research, the following values were considered to be the best for our UK accidents dataset. Results of the hypertuning can be seen in Figure 14. We also pu

Best Parameters can be seen in Figure 14. These parameters have worked for our data for the following reasons.The large number of estimators (827) enabled strong predictions by capturing a wide range of data patterns. With a greater maximum depth (38) the trees were able to understand complex linkages, enhancing accuracy. Random feature selection (sqrt) at each split prevented overfitting and efficiently handled high-dimensional data. The number of data per split (7) and leaf (1) limited model development and reduced noise in predictions. The model produced more consistent predictions by training on the complete dataset (bootstrap: False). Class imbalance concerns were resolved by balancing class weights (balanced_subsample), resulting in better accuracy for both classes. The results achieved can be seen in Figure 15.

Figure 15

Best Parameters

```
best_params
```

```
{'n_estimators': 827,
 'max_depth': 38,
 'max_features': 'sqrt',
 'min_samples_split': 7,
 'min_samples_leaf': 1,
 'bootstrap': False,
 'class_weight': 'balanced_subsample'}
```

Figure 16

Results of Random Forest Model after Hyperparameter Tuning

```
Random Forest Classification Report (with best hyperparameters):
              precision    recall  f1-score   support

           1       0.95      0.93      0.94     15936
           2       0.90      0.85      0.87     15959
           3       0.87      0.93      0.90     16463

    accuracy                           0.90     48358
   macro avg       0.91      0.90      0.90     48358
weighted avg       0.90      0.90      0.90     48358

Accuracy: 0.9036353860788288
```

Discussions

1) Why is one method better than other methods in your data?

When compared to other models, the Random Forest model produced superior outcomes. It improved considerably more after conducting hyper parameter adjustment. The reasons why we believe one model performed better than others are clearly outlined below:

Random Forest

The Random Forest model, as previously stated, predicts the test data by generating numerous decision trees for distinct subsets of data each time and takes the average of the majority projected value. So, even for our dataset, when we implemented the Random Forest with almost default parameters and only one parameter, 'random_state= 42', which was kept the same for the decision tree as well in order to compare both, we saw that the accuracy of the Random Forest model was very high 89.6%.

After modeling with the top 15 features deduced from the feature selection and methods and domain knowledge, the accuracy dropped slightly to 86.2 percent. However this drop is easily justifiable considering that it was just about a 3% percent drop while columns were reduced from 57 to 15. It was the best performer among the different models. The features in our dataset tend to have complicated linkages and nonlinear interactions. Random Forest is capable of successfully capturing and simulating such nonlinearity. It does this through the use of an ensemble of decision trees, which can collectively understand complicated patterns and relationships in data. A huge number of features are also included in the dataset. Random Forest could successfully handle high-dimensional data by picking a subset of characteristics at each decision tree node. This procedure reduces the danger of overfitting and may result in improved generalization performance.

As mentioned earlier in this report, Random Forest was selected for hyper parameter tuning as it was the best performer among the other models. After hyper parameter tuning we were able to achieve 90% accuracy. The ability of these parameters to find a compromise between bias and variance is why they are rated ideal for our dataset. They allow the model to capture the underlying patterns and properties of the accident data without overfitting. Optuna established during the optimization process that these precise values for the hyperparameters optimize the model's accuracy on our dataset, making them the best configuration for our Random Forest model.

Decision Tree

Due to its limited abilities to handle complexity and nonlinear interactions, a Decision Tree may have struggled with the UK accidents dataset. It is prone to overfitting, especially when the depth is not regulated, and may struggle with high-dimensional data, perhaps resulting in an extremely complicated tree or poor generalization. Decision Trees are susceptible to outliers and noise, which can affect their decision-making process and accuracy. Furthermore, unlike Random Forests, Decision Trees lack ensemble averaging, making them less resistant to individual biases or mistakes. Maybe because of this, the decision tree failed to outperform random forest. Even after a parameter hyper tuning we could only achieve the highest accuracy of 86%.

Gaussian Naive Bayes

Gaussian Naive Bayes was the worst performer amongst the other models with accuracy of 69%. Several variables can be blamed for Gaussian Naive Bayes' low performance on the dataset. For starters, the concept of feature independence may not be valid in real-world circumstances, and complicated interactions among the characteristics in the UK accidents dataset may have resulted in incorrect predictions. Second, if the features might not have a Gaussian distribution, as anticipated by Gaussian Naive Bayes, thus may have difficulty effectively capturing the underlying patterns. Furthermore, an uneven class distribution might bias the model in favor of the dominant class, resulting in bad forecasts for the minority class. Finally, the Gaussian Naive Bayes' susceptibility to outliers and the effect of irrelevant characteristics might add noise and reduce its performance.

2) Difference between all the features and the selected features

In this project the accuracies are compared before and after feature selection . Before feature selection there were about 57 features where columns like “Pedestrian_Road_Maintenance_Worker", "2nd_road_class",and "junction control" were removed as they have null values more than 30 percent and a few columns like “Age_Band_Of_Driver” ,”Age_Band_Of_Casualty”,”The_Driver_was_Left_handed” and a few other columns were removed using the domain knowledge gained while working on this project. The Latitude and Longitude columns were removed due to high correlation (> 0.9) with the local_authority_district column.A few columns were removed based on the heat map ,a few features which were positively highly correlated and negatively highly correlated were analyzed and removed.

All features talk about the entire set of features available during the initial stages i.e 57 features and the selected features talk about the ones which were later chosen based on domain knowledge and feature performance. Selected features are important because they have shown strong association with respect to the target variable i.e “Accident_Severity”. By selecting selected features the model performs better and only focuses on the features which are important and potentially improves its predictive accuracy and generalization capabilities.After getting top 15 features from four different models the final 15 features were chosen based on both domain knowledge and tests. Based on the research papers and domain knowledge it is seen that most of the accidents occur when they don't follow the “speed limit”, An accident is severe most of the time when a “Police officer” attends the scene,According to [1]”Light Conditions” also play an important role in road accidents.Feature selection is done by using models such as Logistic Regression,Decision Trees ,Anova Test,Chi Square test .
The fifteen features which are chosen are

Table 3

Top 15 features and their importance

Feature	Importance(Why it is chosen)
Casualty_Severity	Talks about the condition of the person injured.
Speed_Limit	Over speeding is often one of the most commonly seen reasons for a road accident
Number_Of_Casualties	Talks about the number of people injured. It is always important to know how many people have been affected by the accident.
Light_Conditions	Bad lighting results in the risk for pedestrians,no clear visibility and hence causes accidents.
Urban_or_Rural_Area	Factors like traffic density,infrastructure,speed limit, and road designs play an important role in these areas.
Vehicle_Manoeurve	U-turns,abrupt lane changing and other factors result in road accidents.
Did_Police_Officer_Attend_Scene	Represents the severity of the accidents.
Local_Authority_(District)	It helps in analyzing and making better rules for areas which have many road accidents.
Number_of_Vehicles	In most cases when more vehicles are

	involved, higher severity of accidents tend to occur.
Vehicle_Type	Each vehicle has different endurance levels. A car is safer than bikes.A bus has more people so might cause more casualties. It is important to know the vehicle type to help improve the safety regulations.
Road_Type	Improper roads lead to higher accident severity.
Pedestrian_Location	It is important to keep track of the location to identify the high risk areas for pedestrians.
Junction_Location	Describes the location of the accident with respect to junction or intersection
Junction_Detail	Mentions the specifics of that particular junction such as light conditions,bad roads and so on.
Skidding_and_Overturning	Important as it indicates overspeeding suggests a loss of vehicle control.

3) Result (interpretability) based on your domain knowledge and references

Models like Logistic Regression, Decision Trees, Anova Test, and Chi Square Test are used for feature selection. Based on the domain knowledge and research, the top 15 features are listed in the above section but to elaborate even more on that topic while performing feature selection Casualty_Severity is one of the most important features as it categorizes the severity of the injury of the casualties which is to some extent directly proportional to accident severity. Light_conditions result in poor visibility due to darkness, fog, heavy rain, or other adverse weather conditions that can increase risk of accidents.

"Vehicle_Type" and "Road_Type" were also deemed to be important. To illustrate bigger vehicles such as buses would have a higher probability of a high severity accident. Similarly road_type can affect the severity as well. "Did_Police_Officer_Attend_Scene_of_Accident" was found to be equally important as the absence of police officers could result in minor accidents such as fender benders."Vehicle_Manoeuvre", "Junction_detail", "Junction_location", and "Pedestrian_location" were identified as important features since accidents locations such as intersections and maneuvers such as overtaking could result to a severe accident. "Number_of_Vehicles" and "Number_of_Casualties" were highly important features since higher values for both features corresponded to a greater severity of accidents.

After feature selection 57 features were cut down to 15 features where after feature selection decision tree and random forest have shown improvement and gaussian naive bayes did not perform well after feature selection. Coming to Hidden Knowledge, One reason is because given for the specific project the input size is not the issue. There are quite a few correlated features in this project and gaussian naive bayes consider each input as an independent feature and it is not right. Decision trees and random forests performed better after feature selection as , usually when a large number of inputs are given the model is complex and also it tends to overfit the data. So it doesn't perform well before feature selection. After feature selection according to [2] decision trees and random forest performed well with small input samples .

References

[1]Analysis of the Influencing Factors of Road Environment in Road Traffic Accidents. (2020, September 1). IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=& arnumber=9546339

[2]A Structural Sampling Technique for Better Decision Trees. (2009, April 1). IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=& arnumber=5175961

[3] Bulbul, H. I., Kaya, T., & Tulgar, Y. (2016). Analysis for status of the road accident occurrence and determination of the risk of accident by machine learning in Istanbul. 2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA). https://doi.org/10.1109/icmla.2016.0075

[4] H. Hala, C. Anass, B. Rajaa, B. Youssef and J. Garza-Reyes, "Machine learning techniques for forecasting the traffic accident severity," 2021 International Conference on Digital Age & Technological Advances for Sustainable Development (ICDATA), Marrakech, Morocco, 2021, pp. 47-52, doi: 10.1109/ICDATA52997.2021.00018.

[5]J. Paul, Z. Jahan, K. F. Lateef, M. R. Islam and S. C. Bakchy, "Prediction of Road Accident and Severity of Bangladesh Applying Machine Learning Techniques," 2020 IEEE 8th R10 Humanitarian Technology Conference (R10-HTC), Kuching, Malaysia, 2020, pp. 1-6, doi: 10.1109/R10-HTC49770.2020.9356987.

[6]Prediction of Road Accident and Severity of Bangladesh Applying Machine Learning Techniques. (2020b, December 1). IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9356987

[7]Shanthi, S., & Ramani, R. G. (2012, October). Feature relevance analysis and classification of road traffic accident data through data mining techniques. In Proceedings of the World Congress on Engineering and Computer Science (Vol. 1, pp. 24-26). Sn.

[8] S. Ahmed, M. A. Hossain, M. M. I. Bhuiyan and S. K. Ray, "A Comparative Study of Machine Learning Algorithms to Predict Road Accident Severity," 2021 20th International Conference on Ubiquitous Computing and Communications (IUCC/CIT/DSCI/SmartCNS), London, United Kingdom, 2021, pp. 390-397, doi: 10.1109/IUCC-CIT-DSCI-SmartCNS55181.2021.00069.

[9] Whasphutthisit, T., & Jitsakul, W. (2022). *Comparison of Prediction Models for Road Deaths On Road Network*. https://doi.org/10.1109/kst53302.2022.9729086