Accident Severity Prediction Using Machine
Learning Algorithms.

Machine Learning Project Report

ABSTRACT: -

The smart city concept offers chances to address urban challenges while also improving inhabitants' living conditions. Road traffic accidents have become one of the world's most serious public health crises in recent years, and they are the major cause of deaths. In recent years, road accidents have emerged as a global issue, ranking as the world's ninth leading cause of death in the whole world. Allowing citizens to die in vehicle accidents is completely unacceptable and tragic. As a result, a precise analysis is essential to deal with this overburdened situation. This project is conducted to investigate traffic accidents in greater depth and assess the severity of accidents using machine learning methodologies. We also identify the important aspects that have a direct impact on road accidents and make some helpful recommendations on the subject. Dataset we are using is UK Traffic Accidents records from Kaggle. In the end we are implementing our best model on our Test data and checking out Accident severity prediction on Stream Lit API. This prediction model will help traffic management department and authorities to take necessary precautions and implement safety rules where severity is high and will also contribute to making sustainable transportation system.

INTRODUCTION: -

For traffic safety management and control, it is critically important to accurately estimate the severity of traffic accidents. Traffic accidents are a significant concern that has resulted in significant human injury and financial losses. According to the World Health Organization (WHO), road traffic accidents have claimed the lives of nearly 1.2 million people and wounded 50 million others. Individuals, their families, and nations suffer significant economic losses because of road traffic injuries. The cost of treatment as well as lost productivity for individuals killed or

disabled by their injuries, as well as family members who must take time off from work to care for the injured, contribute to these losses.
Most countries' gross domestic product is lost due to road accidents. Road traffic accidents cost approximately 1% of the Gross National Product (GNP) in developing nations, 1.5 percent in developing countries, and roughly 2% in wealthy countries. Traffic accidents cost approximately $518 billion, which is a significant economic expense. The developing countries' contribution is over US$ 68 billion, which is a substantial sum and well exceeds the amount received in terms of alleviating poverty in the country. Furthermore, considering detailed information and measuring methodologies, the annual projected cost of road traffic accidents in Europe alone, which accounts for 5% of the world's total death toll, exceeds €180 billion, may be significantly underestimated. Identification of factors which are increasing road accident severity needs to minimize to decrease severe accidents. Accident severity prediction, as one of the primary concerns in accident management, aids rescuers in determining the severity of traffic accidents, their potential impact, and executing effective accident management methods.

In this project, our objective is to predict Accident severity of accidents happened in United Kingdom during the period of 2012 to 2014. According to recent report of Government of United Kingdom, in 2019, twice as many men (18,600) as women (8,500) were killed or seriously injured on Britain's roads. Our main purpose is to correctly predict severity which will help traffic authorities and different departments such as healthcare to take necessary steps for making our country’s traffic system and road quality better. We are using different features such as Area (Urban or rural), Number of cars, Day of accidents, weather condition, Police involved, Road light condition, Location etc. to predict the severity of accidents. In this project, we are predicting severity for both urban and rural areas. Our problem is multi-Class classification problem, in which we have three different classes of accident severity. Multi Class classification problems are problems in which we have more than 2 classes. We are predicting severity as 1,2,3 where 3 mean light weight accident rarely some injuries or we can minor accident, 2 show the class of intermediate accident with some sort of injury and 1 indicates heavy accidents in which there are casualties, and maybe more vehicles are involved.

DATASET: -

Dataset we are using in this project is basically taken from famous dataset website Kaggle. Name of dataset is 1.6 million UK Accidents, this is a countrywide car accident dataset, which covers both urban and rural areas of the UK. Each accident record is described by a wide range of data attributes, including the accident location, weather, time, Date etc. This dataset is officially collected by UK Police department and contain mostly severe accidents in which police was also involved. Dataset is officially present on Government of UK website.

Whole datasets consist of three CSV files, each CSV file has around 4 lac and 70 thousand records and 33 different features. All 3 CSV files contain same columns and same number of records. First CSV consist of record of 2005 to 2007, whereas second CSV contain records of accidents happened in time of 2009 to

2011 and third contain data of year 2012 to 2014. Raw data need to be pre-processed, including removing variables with too many missing values, filling variables, dropping unique features, and encoding variables.

In this project, we are using 3rd CSV as our dataset, which contain around 4,64,697 records of accidents and 33 different features. Datasets contain both categorical and continuous features. After finalizing our best model, we are testing our best model on 2nd which also has same structure and shape but different records and values. So, we are using it as our validation set.

DATA PRE – PROCESSING: -

Data preprocessing is an important stage for handling the data before using it in the Machine learning algorithms. Data preprocessing is a data mining technique which is used to transform the raw data in a useful and efficient format. For our dataset, we also must perform some data pre-processing. Task we need to perform are Data Cleaning, Dropping unique features and one-hot encoding.
First, we load our dataset using pandas and check shape of our dataset which is 4,64,697 and 33 columns.

1) Data Cleaning: -

Data cleaning is the process of eliminating or changing data that is inaccurate, incomplete, irrelevant, redundant, or incorrectly formatted to prepare it for analysis. When it comes to data analysis, this data is usually not necessary or beneficial because it can slow down the process or produce false results. Faulty data will badly impact our result, so we must remove it. So now after importing our dataset, we now check for null values using python panda's library and find that one of our column junction details is almost null. So, if we drop null values before dropping this column then it will make all our dataset empty. So, we are dropping this column junction detail. Now, we again check for null values and drop all the null values using pandas "dropna" function.
After dropping null values, we have 4,35,236 records remaining before it was 4,64,697.

2) Data Transformation: -

Now we are checking our columns and dropping those columns which have all its values unique such as Date, time etc. Because in next step we must perform One-hot encoding. So, if also encode unique item columns, then it will generate too many columns which we cannot manage. So, we are dropping all the columns which has totally unique values.
First, we are checking uniqueness of column using pandas. unique() function and check out different columns. The columns which we dropped here are Time, Date, Accident Index, LSOA Accident location, LSOA Highway.
So, after dropping all these columns we have final shape of (435236,26) before One – Hot

encoding.

3) One-Hot Encoding: -

One hot encoding is a process of converting categorical data variables so they can be provided to machine learning algorithms to improve predictions. One hot encoding is a crucial part of feature engineering for machine learning. our machine learning algorithms only understand numbers, so we must provide them numbers by converting categorical variables into continuous variables. It Generate different columns and assign them binary values. One hot encoding is essential before running machine learning algorithm on data set.
Some algorithms can understand categorical data directly such as decision tree but most of The supervised learning algorithm cannot operate on categorical data, they require all input variables to be numeric and generate output in numeric value. This technique of transforming columns into binary variables data set is quite famous in Supervised learning algorithm. These binary variables are also known as dummy variables in statistics
So, after performing one-hot encoding on our dataset, we now have around 66 columns. Before it was 26. So now, we can see number of features transformed and increased after encoding.

Train – Test Split: -

The train-test split procedure is used to estimate the performance of machine learning algorithms when they are used to make predictions on data not used to train the model. Here we are splitting our dataset in train dataset and test dataset with the ratio of 70-30% where 70% is our train dataset and 30 is our test dataset.

METHODOLOGY: -

We are using different machine learning algorithms in this project for accident severity prediction and comparing results of each algorithm to check out which model is fitting performing well on training data set. Hence, it's a multi-class classification problem. So, we are using ROC_AUC with OVR (one vs rest) strategy

1) Random Forest Classifier: -

Random forest is one of the famous supervised learning algorithms. It's an ensemble model which combines the findings of numerous decision trees to create a more powerful learner. RF has a high noise resistance and is resistant to overfitting. Bagging and random selection are the two main concepts of Random Forest.

In our project, performance of random forest is quite well. It is giving us ROC score better than other models maybe because it is good with handling large number of features. It is good with high dimension data as we know that it works with making subsets by replacement. So, maybe this is the reason it is giving us good accuracy and roc score on train dataset.

```
ROC Score: 0.6938906595174897
Accuracy of Random Forest: 0.8469597754911132
```

```
[68]
from sklearn.metrics import classification_report
print(classification_report(testy,y_pred, labels=[1,2,3]))
```

```
              precision    recall  f1-score   support

           1       0.00      0.00      0.00      1492
           2       0.00      0.00      0.00     18594
           3       0.85      1.00      0.92    110485

    accuracy                           0.85    130571
   macro avg       0.28      0.33      0.31    130571
weighted avg       0.72      0.85      0.78    130571
```

2) Decision Tree Classifier: -

The second model we are implementing on our train dataset is Decision Tree Classifier which classify result by creating decision tree. It is quite famous supervised machine learning algorithm use for both regression and classification problems. In our project, we implement decision tree on different parameters but the result of ROC score prediction was just fine. Although, Decision tree was giving us very good accuracy of 99%. These are some ROC score prediction which we get after implementing decision of different depths.

```
[72]
print("ROC Score :" , roc_auc_score(testy,y_pred_proba,multi_class="ovr"))
accuracy = DT.score(trainX, trainy)
print("Accuracy of Decision Tree:",accuracy)
print(classification_report(testy,y_pred, labels=[1,2,3]))
```

```
ROC Score : 0.5832909371444646
Accuracy of Decision Tree: 0.9999343541266638
              precision    recall  f1-score   support

           1       0.13      0.15      0.14      1492
           2       0.28      0.30      0.29     18594
           3       0.88      0.86      0.87    110485

    accuracy                           0.77    130571
   macro avg       0.43      0.44      0.43    130571
weighted avg       0.78      0.77      0.78    130571
```

3) Logistics Regression: -

The first model we use is Linear regression model, which is a classification model and commonly use to predict probability that an instance belongs to class or not. It uses sigmoid function to predict probability of a particular class and this function return value between 0 and 1. In our case, Logistic regression is not performing very well on our train data set. It is giving us ROC score of only 0.56. which is quite low. We use different random states to check if score increase but its result was not up to the mark.

```
[75]
    print("ROC Score :" , roc_auc_score(testy,y_pred_proba,multi_class="ovr"))
    accuracy = lr.score(trainX, trainy)
    print("Accuracy of Logistic Regression:",accuracy)
    print(classification_report(testy,y_pred, labels=[1,2,3]))

...  ROC Score : 0.5632775790906478
     Accuracy of Logistic Regression: 0.8469597754911132
                  precision    recall  f1-score   support

               1       0.00      0.00      0.00      1492
               2       0.00      0.00      0.00     18594
               3       0.85      1.00      0.92    110485

        accuracy                           0.85    130571
       macro avg       0.28      0.33      0.31    130571
    weighted avg       0.72      0.85      0.78    130571
```

RESULTS: -

Random Forest is performing best among the current 3 models we applied, with ROC score of 0.69. We can further work on this project to make prediction better, here in this project we use only 5 base models but there is other several models such as SVM, Naïve Bayes, XGBoost etc. which maybe perform better than our models. Furthermore, we can also implement optimization technique on Voting classifier to enhance its prediction.