Final Project Report

Prediction of Accident Severity

By Shraddha Nimbalkar

IBM Data Science Professional Certificate

Section 1: Introduction

Motivation

Road accidents are a serious concern for the majority of nations around the world because accidents can cause severe injuries and fatalities. According to the World Health Organization’s Global Status Report, approximately 1.25 million people deaths happened per year are because of road accident injuries, and most fatality rates were in lower income countries [1]. Our motivation is to predict the accident severity of any road, which will play a crucial factor for traffic control authorities to take proactive precautionary measures. In addition, the dataset we chose was rarely solved in prediction point of view, so we took this opportunity to predict the severity of the accident.

Objective

The purpose of this project is to predict the severity of an accident by training an efficient machine learning model with the help of existing accidents.

This machine learning pipeline will be exploring whether weather, light, and road conditions – that increase the amount of information available about the situation – can increase the accuracy of predicting the severity of an accident.

Section 2: System Design & Implementation details

Algorithms, technologies, and tools & classification algorithms were used and evaluated to predict the accident severity. Algorithms considered for classification were K-nearest neighbours, Naive Bayes classifier, Random Forest classifier, Logistic Regression, Gradient Boosting classifier, SVM. The first thing that came to our mind is that severity is based on different decisions like road, weather conditions. So, we tried different decision tree classifiers. We also tried ensemble methods as the data is very imbalanced.

9 models are used for prediction, each with and without weather variable.

K-Nearest Neighbours Classifier:

This model is a non-parametric method used for classification. We tried using different values of neighbours and got the best result for considering three neighbours for each point and weights parameter was set to ‘distance’ which weights the points by an inverse of their distances.

Naive Bayes Classifier:

This model is a probabilistic framework for solving classification problems. As the features selected are discrete, we used multinomial Naive Bayes classifier instead of Gaussian. We tried this algorithm after data cleaning on the entire dataset which gave an F1-score

Random Forest classifier:

As our data is very imbalanced, we tried ensemble methods, of which random forest classifier is one. A random forest is a meta estimator that fits number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting.

Logistic Regression:

This model is the basic and popular for solving classification problems. Unlike Linear Regression, Logistic regression model uses a sigmoid function to deal with outliers. Class weights parameter sets the weights for imbalanced classes by adjusting weights inversely proportional to class frequency.

Support Vector Machine:

SVMs are based on the idea of finding a hyperplane that best divides a data set into two classes. As the features in this dataset were very sparse and non-separable by any hyperplane.

Below is the Score generated for the following conditions

Machine Learning algorithm scores without weather related conditions		Machine Learning algorithm scores without weather related conditions	
Model	**Score**	**Model**	**Score**
Logistic Regression	92.47	Random Forest	92.59
Random Forest	92.26	Logistic Regression	92.47
Support Vector Machines	92.06	Support Vector Machines	92.12
Stochastic Gradient Decent	92.06	Perceptron	92.06
Linear SVC	92.06	Linear SVC	92.06
Perceptron	92	Stochastic Gradient Decent	92.04
KNN	90.4	KNN	90.2
Naive Bayes	90	Naive Bayes	87.94
Decision Tree	86.4	Decision Tree	86.18

Technologies & Tools used

For developing this project, below tools and technologies have been used.

Python: Python is easy to understand language and has a rich set of libraries to use for data pre-processing, modelling, and evaluating the algorithms. Moreover, python has very good community support which is very useful for debugging the code.

Jupyter Notebook: Jupyter notebook is a simple and interactive tool for running python code. Also, it has many different sets of features such as downloadable to .py, .ipynb, and .html files.

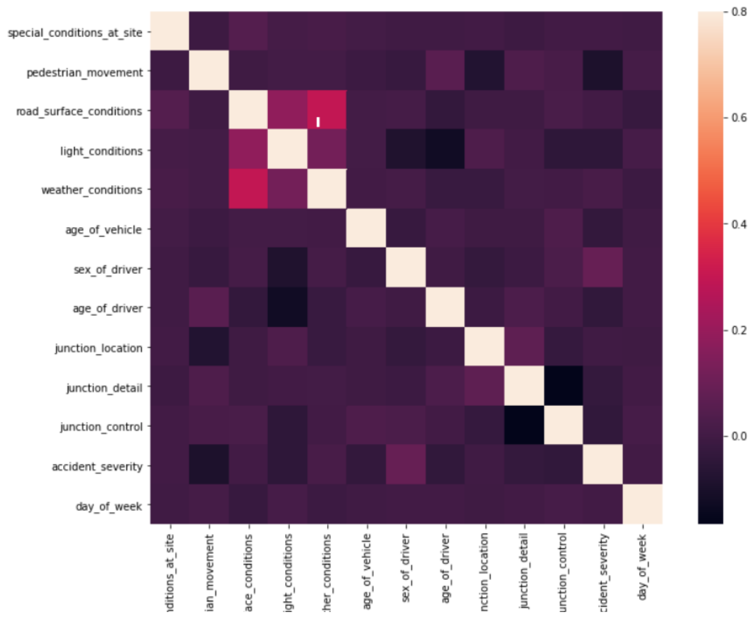
Section 3: Experiments / Proof of concept evaluation

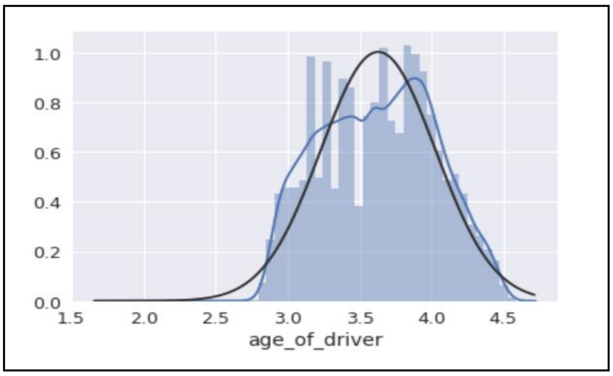
Dataset used:

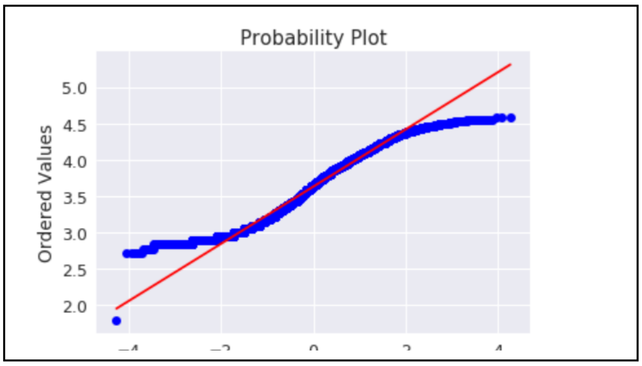
Data has been fetched from Open Data platform UK. It Captures Road Accidents in UK between 1979 and 2015 and has 70 features/columns and more than 250K rows. The Data set contains column including weather conditions, Road class, road type, junction details, road surface conditions, light conditions, etc.

Data Visualization

Heatmap : Plotting importance of each feature







After taking the logarithm of both the age of the driver and age of the vehicle improved the fit by altering the scale and making the variables more "normally" distributed. After taking the log, one can notice that the values range from approximately 2.5 to 4.5. This increases the performance of machine learning algorithms, as the numerical values do not have disproportionate amounts of computing value compared to all the other categorical variables.

Section 4: Discussion & Conclusions

Decisions, difficulties, and discussions: Our main aim was to predict the severity of the accident when it is “serious” and “fatal”. It was very difficult to handle this large-sized data. Data is highly imbalanced so even though most of our algorithms were giving > 89% accuracies, it was of no use. It was predicting all the accidents as slight accidents. After checking on all these algorithms, the team even tried dimensionality reduction techniques and but the results were not improved. Then the team decided to use the under sampled dataset as it was giving better results in predicting the severe/fatal accidents. This decision was made on trying out oversampling, under sampling, test and train data with an equal ratio of classification classes.

Conclusion:

The results indicated that adding weather-related features to a machine learning algorithm in predicting severity of an accident did not change the accuracy of the model. When adding three features of light condition, weather condition, and the condition of the road surface, the measures of recall, precision, and f1-score remained unchanged.

When looking at the overall performance of all the algorithms, there was an increase in accuracy between the data with weather conditions when compared to data without weather related conditions. Namely, random forest algorithm increased performance by 0.59%. The previous top performer when no weather-related conditions were introduced, Logistic Regression, sustained the same level of accuracy. Hence, it was concluded to further scrutinize the recall, precision, and f1-score of random forest algorithm to see whether there was an actual change in prediction power.

One reason why we did not see much difference by the weather might be that: when the weather is bad, perhaps people are discouraged from driving at all. So the traffic volume reduces and traffic accidents are rarer and smaller.