Predicting the Severity of Car Accident

Sumit Kumar
October 20, ‘20

1. Introduction

1.1 Preface

Road Accidents are becoming a very common phenomenon now-a-days, given the circumstances like high traffic, weather, condition of driver while driving, condition of vehicle. Taking into consideration all these aspects, the stakeholders, i.e. the common people and the roadways authorities, need a system that can predict the fatalities, damage and precautions to road accidents.

1.2 Problem Statement

The Business Problem Statement is to predict the Severity of a road accident, given different attributes of the situation, like coordinates, number of vehicles, road condition, light condition, drug usage, alcohol usage during driving, not paying attention.

1.3 Interest

The Traffic Control Board will be the primary target area, who’d be interested in this project. Even Individuals who are aware and want to be updated and take an extra step of precaution can use this.

2. Data

The data source is provided by Coursera for the same purpose. The dataset contains 37 different columns, a few redundant and a few not very impactful. So the first step after acquiring data was to process the data.

2.1 Feature Selection

Following columns were selected for the project:

['SEVERITYCODE', 'ADDRTYPE', 'INTKEY', 'COLLISIONTYPE', 'PERSONCOUNT', 'PEDCOUNT', 'PEDCYLCOUNT', 'VEHCOUNT', 'JUNCTIONTYPE', 'INATTENTIONIND', 'UNDERINFL', 'WEATHER', 'ROADCOND', 'LIGHTCOND', 'SPEEDING', 'HITPARKEDCAR']

2.2 Data Cleansing and Preprocessing

All these selected columns were then converted to usable format by one-hot coding or converting categorical data into processable integer values to ease out the execution of the problem.

The Nan values in feature columns were removed as follows:

'SEVERITYCODE': 0,
'ADDRTYPE': '',
'INTKEY': 0,
'COLLISIONTYPE': '',
'PERSONCOUNT': 0,
'PEDCOUNT': 0,
'PEDCYLCOUNT': 0,
'VEHCOUNT': 0,
'JUNCTIONTYPE': '',
'INATTENTIONIND': 'N',
'UNDERINFL': '',
'WEATHER': '',
'ROADCOND': '',
'LIGHTCOND': '',
'SPEEDING': 'N',
'HITPARKEDCAR': ''

2.2.1 One Hot Coding

The following columns are one-hot coded:

['ADDRTYPE', 'INATTENTIONIND', 'UNDERINFL', 'SPEEDING', 'HITPARKEDCAR']

2.2.2 Categorical Splitting

The following columns are categorically splitted:

['COLLISIONTYPE', 'JUNCTIONTYPE', 'WEATHER', 'ROADCOND', 'LIGHTCOND']

2.3 Data Interpretation and Data Visualization

The data was analyzed for a lot of insights. The impactees are recorded for each scenario, and are showcased as below.

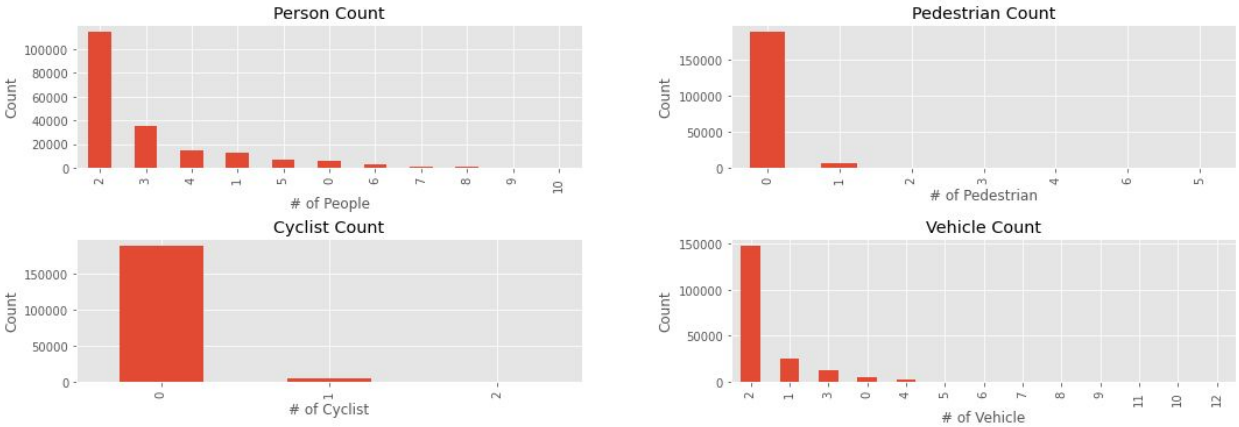
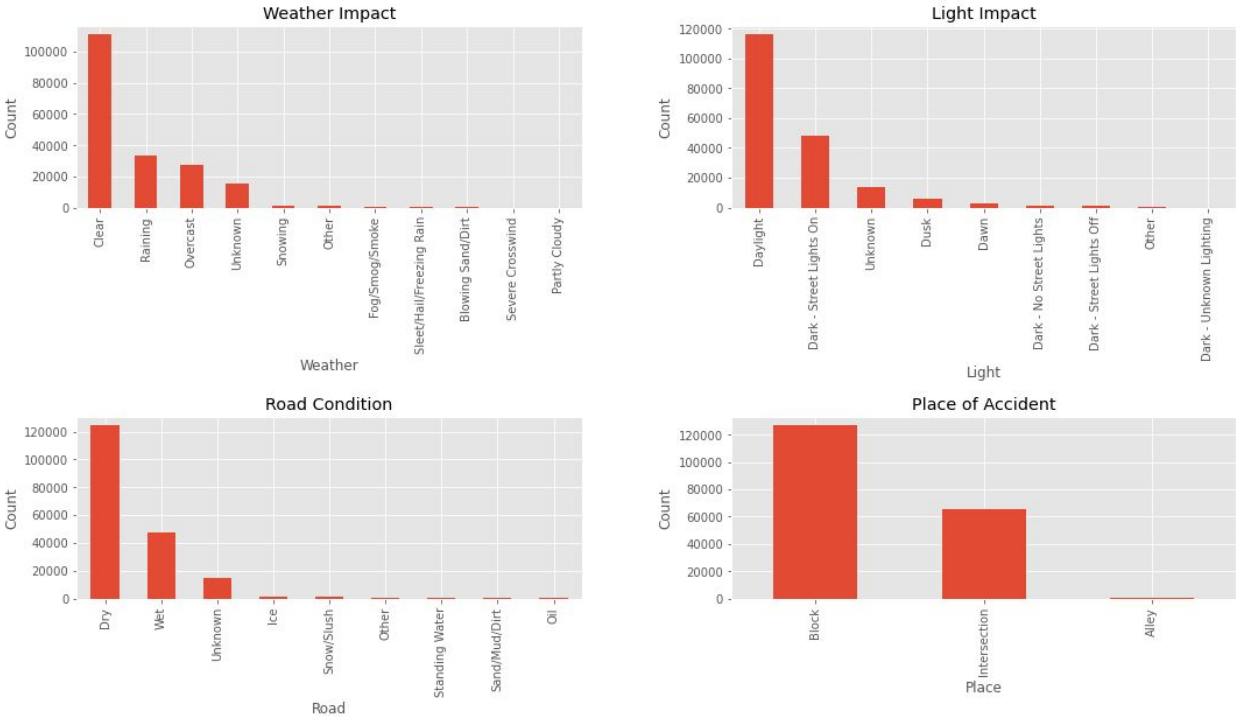


Fig 2.1: Number of impactees (a) person (b) pedestrian (c) cyclist (d) vehicles

The impact of Weather, light, road conditions are showcased below and along with it, the place of Accident.



In [ ]:

Fig 2.2: Number of Accidents v/s (a) Weather (b) Light (c) Road (d) Place of Accident

3. Methodology

3.1 Exploratory Data Analysis

As part of Exploratory Data Analysis correlation between features were calculated. The heatmap of correlation is shown below

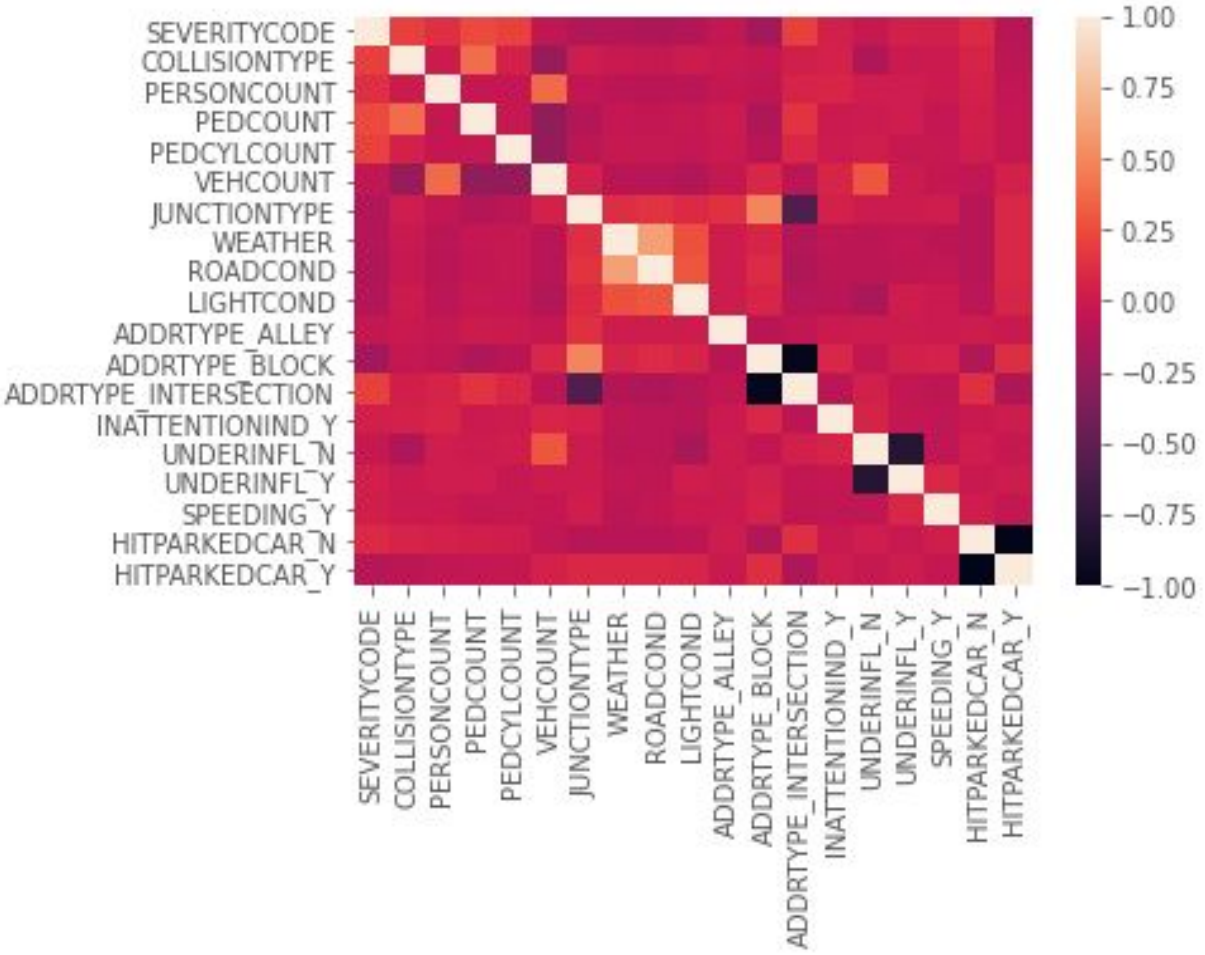


Fig 3.1: Correlation of selected features

3.2 Machine Learning Models

Following Machine Learning Models are implemented in the Project.

1. Decision Tree
2. Gaussian Naive Bayes
3. Nearest Neighbors
4. Neural Networks

3.2.1 Decision Tree

Decision Trees (DTs) are a non-parametric supervised learning method used for classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features.

Some advantages of decision trees are: Simple to understand and to interpret. Trees can be visualised. Requires little data preparation. Other techniques often require data normalisation, dummy variables need to be created and blank values to be removed. Note however that this module does not support missing values. The cost of using the tree (i.e., predicting data) is logarithmic in the number of data points used to train the tree. Able to handle both numerical and categorical data. Other techniques are usually specialised in analysing datasets that have only one type of variable. See algorithms for more information. Able to handle multi-output problems. Uses a white box model. If a given situation is observable in a model, the explanation for the condition is easily explained by boolean logic. By contrast, in a black box model (e.g., in an artificial neural network), results may be more difficult to interpret. Possible to validate a model using statistical tests. That makes it possible to account for the reliability of the model. Performs well even if its assumptions are somewhat violated by the true model from which the data were generated.

The disadvantages of decision trees include: Decision-tree learners can create over-complex trees that do not generalise the data well. This is called overfitting. Mechanisms such as pruning, setting the minimum number of samples required at a leaf node or setting the maximum depth of the tree are necessary to avoid this problem. Decision trees can be unstable because small variations in the data might result in a completely different tree being generated. This problem is mitigated by using decision trees within an ensemble. The problem of learning an optimal decision tree is known to be NP-complete under several aspects of optimality and even for simple concepts. Consequently, practical decision-tree learning algorithms are based on heuristic algorithms such as the greedy algorithm where locally optimal decisions are made at each node. Such algorithms cannot guarantee to return the globally optimal decision tree. This can be mitigated by training multiple trees in an ensemble learner, where the features and samples are randomly sampled with replacement. There are concepts that are hard to learn because decision trees do not express them easily, such as XOR, parity or multiplexer problems. Decision tree learners create biased trees if some classes dominate. It is therefore recommended to balance the dataset prior to fitting with the decision tree.

3.2.2 Gaussian Naive Bayes

In spite of their apparently over-simplified assumptions, naive Bayes classifiers have worked quite well in many real-world situations, famously document classification and spam filtering. They require a small amount of training data to estimate the necessary parameters. Naive Bayes learners and classifiers can be extremely fast compared to more sophisticated methods. The decoupling of the class conditional feature distributions means that each distribution can be independently estimated as a one dimensional distribution. This in turn helps to alleviate problems stemming from the curse of dimensionality. On the flip side, although naive Bayes is known as a decent classifier, it is known to be a bad estimator, so the probability outputs from predict_proba are not to be taken too seriously.

3.2.3 Nearest Neighbors

The principle behind nearest neighbor methods is to find a predefined number of training samples closest in distance to the new point, and predict the label from these. The number of samples can be a user-defined

constant (k-nearest neighbor learning), or vary based on the local density of points (radius-based neighbor learning). The distance can, in general, be any metric measure: standard Euclidean distance is the most common choice. Neighbors-based methods are known as non-generalizing machine learning methods, since they simply “remember” all of its training data (possibly transformed into a fast indexing structure such as a Ball Tree or KD Tree).

3.2.4 Neural Networks

A multilayer perceptron (MLP) is a class of feedforward artificial neural network (ANN). The term MLP is used ambiguously, sometimes loosely to any feedforward ANN, sometimes strictly to refer to networks composed of multiple layers of perceptrons (with threshold activation); see § Terminology. Multilayer perceptrons are sometimes colloquially referred to as "vanilla" neural networks, especially when they have a single hidden layer. An MLP consists of at least three layers of nodes: an input layer, a hidden layer and an output layer. Except for the input nodes, each node is a neuron that uses a nonlinear activation function. MLP utilizes a supervised learning technique called backpropagation for training. Its multiple layers and non-linear activation distinguish MLP from a linear perceptron. It can distinguish data that is not linearly separable.

The advantages of Multi-layer Perceptron (MLP) are: Capability to learn non-linear models. Capability to learn models in real-time (on-line learning) using partial_fit.

The disadvantages of Multi-layer Perceptron (MLP) include: MLP with hidden layers have a non-convex loss function where there exists more than one local minimum. Therefore different random weight initializations can lead to different validation accuracy. MLP requires tuning a number of hyperparameters such as the number of hidden neurons, layers, and iterations. MLP is sensitive to feature scaling.

4. Results

The accuracy of all 4 predictive models is as given below.

- Accuracy for **Decision Tree**: 0.7472004383411527
- Accuracy for **Gaussian Naive Bayes**: 0.7047532618745933
- Accuracy for **Nearest Neighbors**: 0.6791205780623951
- Accuracy for **Neural Network**: 0.7035889181877333

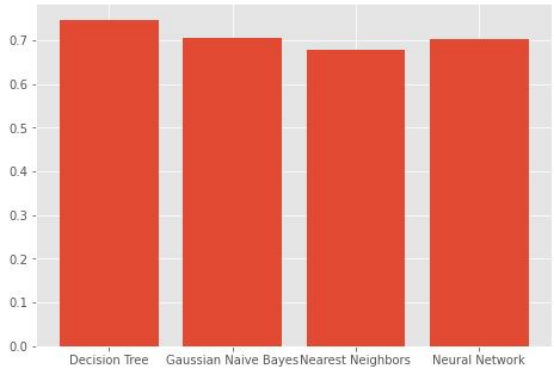


Fig 4.1: Accuracy Plot for implemented Models

5. Conclusions

The conclusion from the Studies are as follows.

The accuracy of the classifiers is adequately high, highest being 74.7%. This usually means that the model is under fitted i.e. it needs to be trained on more data. Though the dataset has a lot of variety in terms of scenarios, more volume of the data for such scenarios has to be collected. Certain features with missing values were removed, this reduced the dimensionality of the dataset, these features could have been correlated to other important features but they had to be removed. A better effort has to be made to collect data to reduce the number of missing values.

6. Follow Ups

As mentioned above, the amount of data available to train the above mentioned models is not sufficient and it does not seem to have enough data of all varieties. Hence, integrating Cross Validation methods with hyper parameter models would help in training and possibly increase the accuracy of every classification model.