CSE343: Machine Learning
Mid-Project Report : Traffic Severity Analysis

Chaitanya Garg (2021248) Deepanshu Dabas (2021249) Rudra Jyotirmay (2021280) Arpan Kumar (2021020)

Abstract

Road Accidents have a substantial economic impact. However, their effects on lost lives are more significant. In the USA alone,[1] The National Highway Traffic Safety Administration released its latest projections for traffic fatalities in 2022, estimating that 42,795 people died in motor vehicle traffic crashes. Reducing these accidents is challenging; this enlightenment came upon finding multiple articles about deaths in road accidents.

1. Introduction

The issue of road safety is increasingly gaining prominence as a significant societal issue globally.
Recognising the primary causes of road traffic accidents is critical for developing effective solutions to lessen the detrimental impact on human lives and property. Road severity is not random; it follows predictable patterns that can be predicted and minimised.
Accurate traffic severity predictions can assist in reducing response times of emergency services and improving overall road safety. This project aims to predict the severity of traffic accidents based on various features such as weather conditions, distance, and time of day.

2. Literature Survey

We reviewed various research papers pertaining to Traffic Severity Analysis, which used the following models:

2.1. Improved naive Bayes classification algorithm for traffic risk management

[1] The paper introduces the Naive Bayes classification method and how it is advantageous because it only needs to estimate the necessary parameters (mean and variance of variables) based on a small amount of training data. The authors of the paper have then used the Naive Bayes classifier to predict traffic severity based on the standard Bayes Theorem.

However, Naive Bayes faces some obvious shortcomings, so the paper explores ways to arrive at an "Improved Naive Bayes Classifier". This is achieved by first performing feature weighting, which adds an extra "weight" term to the standard Bayes Theorem, which considers the importance of a particular feature in the dataset compared to the other features.

Secondly, the Naive Bayes classifier might be inaccurate when the number of training samples is small and the number of attributes is large. In order to resolve this, the authors use the concept of Laplace calibration, which solves the problem of the category conditional probability being 0 while not changing the classification of the sample.

2.2. Traffic Accidents Severity Prediction using Support Vector Machine Models

[3] The paper discusses about the use of the SVM to predict the fatality rate of an accident and draws a comparison between the SVM based on the radial basis function and the linear kernel function. Then, the methodology aims to use the one with the better confusion matrix as the kernel function.
The paper has worked on the dataset of accidents in Lebanon in the years 2016-2017. In the data preprocessing step, they normalised and removed the outliers. SVM is an algorithm that aims to find the maximum margin of the hyperplane, which in turn provides the maximum distance between separation decision classes. This is calculated using the formula: $Y_i(w^T\phi(x_i) + b) >= 1$
SVM involves the extensive use of mathematical functions called kernels, which transform the input into the needed form. Kernels can be of functions such as linear, RBF, sigmoid, polynomial, etc.
The model used gives the best accuracy of 91% on the testing set for RBF kernel while linear follows with an accuracy of 84.6% on the testing set.

2.3. Modeling Road Accident Severity with Logistic Regression

[2] The paper talks about how Logistic Regression (LR) is widely employed in traffic accident severity analysis as it helps clearly establish the factors that the severity of an accident is correlated to by providing insights into optimum values of variables, standard errors, varying importance of different features, and their effects on the target variable.

In order to train the model, the authors of the paper used IBM Modeler 18.0 software, making use of the logit function to get the probability of a serious accident. The dataset was divided into training and validation sets (in a 70:30 ratio) for model development and validation. LR's output included p-values, determining variable significance. Also, the importance of different features was calculated by using the different probability values obtained.

The study categorised accidents as "serious" (including fatalities and injuries) or "minor" (including only the damage to property) to ensure analytical balance. To address the limited fatality data, fatalities and injuries were combined into "serious accidents," and "minor accidents" were randomly sampled to match their count.

Using the Spearman's rank correlation coefficient method conducted on the highways of Taiwan, it was identified that features like "major cause" and "collision type," and "weather condition" and "surface condition" are strongly correlated, and thus only one of each pair of features was enough to train the model. Thus, the other corresponding feature in each pair was deleted as it was deemed to be redundant in the analysis.

In the final analysis, the authors found that LR made highly accurate predictions and was highly sensitive. It also gauged the correlation between different features and was useful in determining what features had to be handled with more importance to prevent accidents.

2.4. Traffic Accident Analysis Using Decision Trees and Neural Networks

[4] The paper employs Decision Trees and Neural Network techniques to perform traffic accident analysis which is used to identify variables that are correlated to the severity level, standard errors, varying importance of different features, and their effects on the target variable.

The paper uses an accident data set from the Nigeria Road Safety Corps, covering a 24-month period from January 2002 to December 2003. As part of preprocessing, the dataset was divided into both categorical and continuous data and decision tree was trained on the categorical data while neural networks were trained on continuous data.

The first model trained was ANN-based modeling, which used the hyperbolic activation function in the hidden layer and the logistic activation function in the output layer. Such a Radical Basis Function (RBF) neural network achieved training and testing accuracies as 54.73% and 40.56% respectively, with 0.3478 of mean absolute error.

The next model trained was a Decision Tree algorithm, which used an entropy splitting criteria, 100 maximum nodes limit, 10-fold cross-validation, and pruning to prevent overfitting. The results obtained were that the model predicted 115 correctly classified instances and 33 incorrectly classified instances with a Mean absolute error of 0.1835 and Root mean squared error of 0.3029.

Finally, on comparing the results of the models, it was found that the Decision Tree performs better than the Neural Networks based on the error report and number of correctly classified instances. Among the Neural Networks, the RBF Neural Network performed better than the MLP.

3. Dataset

3.1. Dataset Details

We used a countrywide traffic accident dataset available on Kaggle
It comprises of seven years of data, about 7700000 rows, and 46 columns. Since this is raw data, we would need to process and clean this data, and hence Pre-processing is very much needed.
In the dataset, the traffic impacted due to accident, data were collected from February 2016 to March 2023, using multiple APIs that provide streaming traffic incident (or event) data. These APIs broadcast traffic data captured by various entities, including the US and state departments of transportation, law enforcement agencies, traffic cameras, and traffic sensors within the road networks. The dataset currently contains approximately 7.7 million accident records. For more information about this dataset, please visit here .

The filtered dataset consists of only the following columns: Year, Severity, Start_Lat, Start_Lng, Distance(mi), Street, City, County, State, Airport_Code, Temperature(F), Wind_Chill(F), Visibility(mi), Wind_Direction, Weather_Condition, Traffic_Signal, Sunrise_Sunset, TimeDiff

3.2. Data Pre-processing Techniques

We used the following pre-processing techniques to process raw data:

1. **Handling missing values**
   We checked for all NULL value entries in our dataset, which were around 10,000 in total, and deleted all such entries.

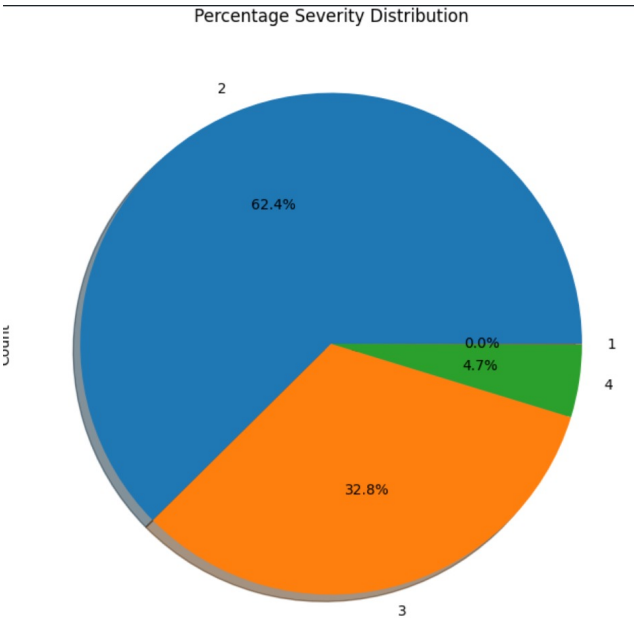

Figure 1. Percentage Severity Distribution

2. **Handling duplicate values**
   Our dataset contained around 5,000 repeated entries. We deleted all duplicates to make all rows unique.

3. **Slicing the dataset**
   Our dataset initially contained about 7 million entries from 2016 to 2023. Training any model on such a large database is not time and resource-feasible. So, we only took entries from 2016 to 2018, bringing down the number of rows to around 3,00,000.

4. **Encoding categorical variables**
   Our cleaned dataset contained 9 numerical columns (type : float64, int64) and 9 categorical columns (type : object, boolean), which had to be encoded before applying any models on it. We used both Label Encoding and One-Hot Encoding in order to do so.

5. **Splitting dataset into Training and Testing set**
   We divided the number of entries into Training and Testing sets in the ratio 80 : 20.

6. **Feature scaling**
   We scaled the features in our dataset to the same range so no feature dominates over the other. We used Standardization using the StandardScaler class of sklearn.preprocessing library in order to do so.

3.3. Data Inferences

The pie-chart of the percentage severity distribution tells us that most of the traffic observed on the roads is of severity level 2 (62.4%) and severity level 3 (32.8%). Traffic severity levels of 1 and 4 are rarely observed.

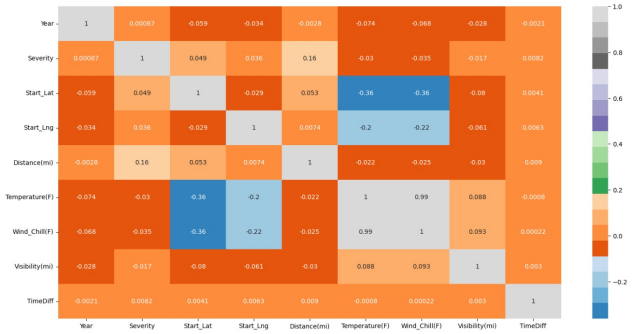

Figure 2. Corelation Heatmap

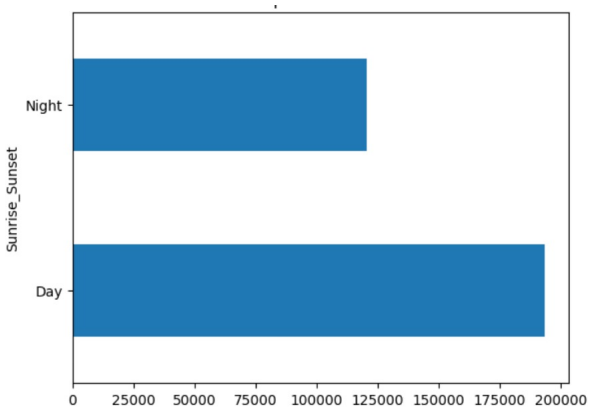

Figure 3. Traffic Severity during night and day

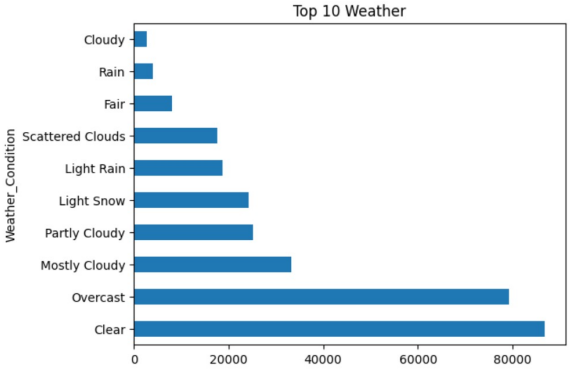

Figure 4. Traffic Severity during different weather conditions

The correlation heatmap of our dataset shows the relationship between different pairs of features. For example, we observe that Wind_Chill(F) and Temperature(F) are strongly positively correlated, TimeDiff and Severity are mildly positively correlated, and Temperature and Start_Lat are moderately negatively correlated.

The bar graph depicting sunrise and sunset times indicates that the majority of accidents occur during daylight hours.

```
Accuracy
0.6698008394370834
confusion_matrix
[[    0    15     1     0]
 [    0 26865  3951     2]
 [    0 11153  5688     1]
 [    0   731   195     2]]
classification_report
              precision    recall  f1-score   support

           0       0.00      0.00      0.00        16
           1       0.69      0.87      0.77     30818
           2       0.58      0.34      0.43     16842
           3       0.40      0.00      0.00       928

    accuracy                           0.67     48604
   macro avg       0.42      0.30      0.30     48604
weighted avg       0.65      0.67      0.64     48604
```

Figure 5. Naive Bayes testing set analysis

The bar graph representing weather conditions reveals that the majority of accidents occur during clear or overcast weather conditions.

4. Methodology, Model Details

We have experimented with the following machine-learning models:

4.1. Mixed Naive Bayes

Naive Bias is a supervised learning classification model. It uses the naive bayes formula with a naive bias assumption that data features are independent of each other

1. **Algorithm** We have made a custom class to handle Naïve Bias Classification (NBC) and used scikit learn for testing purposes such as f1 score from sklearn.metrics.

    - Upon running train on the initiated class for NBC, the model counts the required parameters conditioned on each value of the output and stores them for categorical columns and stores the relevant values of mean and std for Numerical data.
    - Predict function takes in the row for which prediction is to be performed
    - It checks which category does a particular column lie in. If it's a category then naïve bias is applied using naïve formula along with a Laplacian method with alpha =5 . If its numeric category then it applies the relevant Gaussian model.
    - Predict_alpha takes a custom alpha to find the prediction
    - Accuracy score takes in test set and its ground truth and for each row in set runs predict function on it to give the accuracy score. Accuracy_score_alpha helps in finding the best fit alpha
    - Predict_weighted makes an attempt towards weighted naïve bias

4.2. Support Vector Machine(SVM)

SVMs are supervised learning models with associated learning algorithms that analyze data for classification, regression and clustering analysis. We will be using it for the classification of traffic severity levels (In the range between 1-4).

1. **Algorithm**

    - Need to determine kernel that performs best on the dataset.
    - Determine best by hyperparameter tuning for a SVM classifier using grid search.
    - Higher degree models can be likely to be overfit, used regularization,and likely prevent the model from overfitting the data.
    - After performing a grid search, we will get the best-performing SVM model along with regularization parameters and therefore we can train the best-performing model on training data.
    - We will then check its performance on test data to know the performance of model.
    - We use the cross-validation technique to determine the performance of model

4.3. Logistic Regression

1. **Algorithm**

    - The first model runs on the dataset with only 2-3 severity rating as these two are the major severity values.
    - The second model is a multilevel logistic regression model, where the data is first classified into more and less severe categories.
    - Less severe data is then classified into 1 and 2 categories of severity, while more severe data is classified into 3 and 4 categories of severity.

4.4. Decision Trees and Boosting

1. **Algorithm**

    - The model reads the input data upon calling the fit function
    - If it is at the node, then split is made such that Info Gain is the highest. Else, it is at the leaf it checks if all the data is being classified properly or not

```
Accuracy on training set:  0.9999228454594553
Accuracy on the test set:  0.8854415274463007

Classification Report
              precision    recall  f1-score   support

           1       0.00      0.00      0.00        14
           2       0.90      0.93      0.91     30828
           3       0.87      0.82      0.84     16821
           4       0.85      0.53      0.65       941

    accuracy                           0.89     48604
   macro avg       0.65      0.57      0.60     48604
weighted avg       0.88      0.89      0.88     48604
```

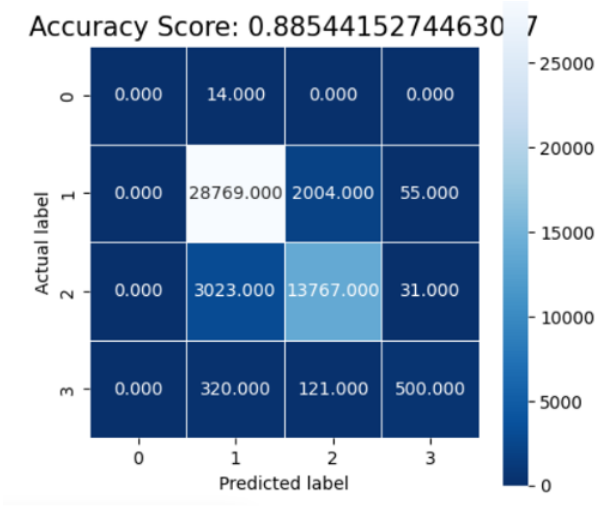

Figure 6. Random Forest Analysis

- If the classification is incorrect, then the algorithm makes a split is continued if termination condition is not being fulfilled
- Algorithm stops on reaching the termination condition

2. **Algorithm Random Forests**

- The model reads the input data upon calling the fit function and sees the number of decision trees to be made(n)
- The model proceeds to make a decision tree and randomly chooses m features to be used for building the tree and train data is created by means of bootstrapping.
- In the end, the model ends by making n different decision trees, each trained on a different bootstrap data and made by splitting on m randomly chosen features

3. **Boosting Algorithm**

- Ada Boosting - This works by creating decision stumps(depth=1) and having equal weights for all, the misclassified examples are given higher weightage as the algorithm progresses
- Gradient Boosting - The primary aim of this boosting method is to decrease the loss function
- Xtreme Gradient Boosting - It is a modified version of Gradient Boosting which gives higher weights and minimizes the loss along with the use of parallelization and cache

Technique	Train Accuracy	Test Accuracy
Logistic Regression	0.645	0.646
Mixed Naive Bayes	0.665	0.66
Support Vector Machine	0.643	0.681
Decision Tree	0.999	0.83
Random Forest	0.999	0.885
Adaptive Boosting	0.6533	0.6538
Gradient Boosting	0.944	0.858
XG Boosting	**0.987**	**0.913**
Multi Layer Precepton	0.707	0.709

Figure 7. Models Accuracies

4.5. Multi-Layer Perceptron (MLP)

MLP is a type of artificial neural network that 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.Its used for both classification and regression tasks

1. **Algorithm**

- The MLP model was implemented using the MLPClassifier from the sklearn.neural_network library.
- Grid search was performed using different activation functions, including 'tanh', 'relu', 'logistic', and 'identity' on subset dataset of 5k entries.
- The best parameters from the grid search were then used to initialize the model.
- The model with the highest accuracy on the training set was selected as the best model and then trained individually on the complete dataset.
- Using the best-performing activation function, the model was learning and improving its performance on the training set and had better accuracy scores on both the training set and test set.

5. Results And Analysis

We implemented the following models: Naive Bayes Classifier, which makes use of Laplace and weighted cor-

rection, Support Vector Machine, which takes in various kernel functions like linear and Radial Basis Function(RBF), logistic regression classifiers where the first classifier is a simple logistic regression and the other is a multilevel logistic regression, Decision Trees and Random forests, boosting algorithms like Gradient Boost and XGBoost, and finally a Multi-Layer Perceptron.

The reason why linear regression is not used is that severity can take up only 4 values which are discrete values, while linear regression works best for predicting real numbers given the parameters hence, we chose models like Naives Bayes, SVM and Decision Trees, which give discrete values.

SVM's main use is to classify binary data, but it works well on multiclass data. This is possible as scikit-learn's implementation of SVM considers 2 classifying factors, whether it is a part of a severity class or not. In other words, it breaks down the data internally into binary classes.

We also performed K-means clustering and the K-Nearest Neighbors Algorithm on our dataset. However, both these gave low accuracy scores because our dataset can't be clustered properly.

It is observed that the XGBoost gave the best accuracy of 0.913 on the testing test, followed by Random Forest with a high accuracy of 0.885, Gradient Boost with an accuracy of 0.858, and Decision Tree also having a high accuracy of 0.83. Apart from these, Support Vector Machine and Mixed Naive Bayes also gave a decent accuracy of 0.681 and 0.66, respectively.

6. Conclusion

1. In this report, we explored the prediction of traffic severity using machine learning models and tried to analyze the dataset with various techniques to determine the best models.

2. Among classification models, Random Forest demonstrated very high accuracy among the models tested, achieving 89% accuracy on the test dataset.

3. Algorithm boosting algorithms trained on different decision trees also gave great results. For example, we were able to achieve 91% accuracy by using XGBoost.

4. We will need to save and restore/reload later our ML Model so as to test our model with new data or to compare multiple models or anything else. Hence, serialization and deserialization of models are required, which we will complete before the final evaluation of all the final models.

5. A real-world applicability is that with such high accuracy, these models could be potentially used in real-world applications, such as in traffic management systems, to predict and manage traffic severity which would lead to a better quality of life.

References

[1] Hong Chen, Songhua Hu, Rui Hua, and Xiuju Zhao. Improved naive bayes classification algorithm for traffic risk management. *Journal on Advances in Signal Processing*, 2021.

[2] Mu-Ming Chen and Mu-Chen Chen. Modeling road accident severity with comparisons of logistic regression, decision tree and random forest. 2020.

[3] Zeinab Farhat, Ali Karouni, Bassam Daya, Pierre Chauvet, and Nizar Hmadeh. Traffic accidents severity prediction using support vector machine models. *International Journal of Innovative Technology and Exploring Engineering*, 2020.

[4] Victor Olutayo and Adekunle Eludire. Traffic accident analysis using decision trees and neural networks. *International Journal of Information Technology and Computer Science*, 6:22–28, 01 2014.