**CIND820 XJH - Big Data Analytics Project - W2024**

Final Report

of

**Final Results and Project Report**

Title: **Predictive Analysis of Traffic Severity**

TMU Student Number: **501274795**

Under the supervision of – **Dr. Tamer Abdou, PhD**

Submitted by: **Prashanth Vijayaraja**

on 01 April, 2024.

**Table of Contents**

[1 Abstract 3](#_Toc162904333)

[2 Introduction 3](#_Toc162904334)

[3 Literature Survey 3](#_Toc162904335)

[3.1 Improved naive Bayes classification algorithm for traffic risk management 3](#_Toc162904336)

[3.2 Modeling Road Accident Severity with Logistic Regression 4](#_Toc162904337)

[3.3 Traffic Accidents Severity Prediction using Support Vector Machine Models 4](#_Toc162904338)

[3.4 Traffic Accident Analysis Using Decision Trees and Neural Networks 5](#_Toc162904339)

[4 Dataset 6](#_Toc162904340)

[4.1 Dataset Details 6](#_Toc162904341)

[4.2 Data Pre-processing Techniques 6](#_Toc162904342)

[4.2.1 Handling missing values 6](#_Toc162904343)

[4.2.2 Handling duplicate values 6](#_Toc162904344)

[4.2.3 Slicing the dataset 6](#_Toc162904345)

[4.2.4 Encoding Categorical Variables 7](#_Toc162904346)

[4.2.5 Splitting dataset into Training and Testing set 7](#_Toc162904347)

[4.2.6 Handling of Outliers 7](#_Toc162904348)

[4.2.7 Feature scaling 7](#_Toc162904349)

[4.3 Data Inferences 8](#_Toc162904350)

[5 Methodology - Model Details 10](#_Toc162904351)

[5.1 Mixed Naive Bayes 10](#_Toc162904352)

[5.1.1 Algorithm 10](#_Toc162904353)

[5.2 Support Vector Machine (SVM) 12](#_Toc162904354)

[5.2.1 Algorithm 12](#_Toc162904355)

[5.3 Logistic Regression 13](#_Toc162904356)

[5.3.1 Algorithm 13](#_Toc162904357)

[5.4 Decision Trees and Random Forests 13](#_Toc162904358)

[5.4.1 Algorithm 13](#_Toc162904359)

[5.5 Random Forests 14](#_Toc162904360)

[5.5.1 Algorithm 14](#_Toc162904361)

[5.6 Boosting 17](#_Toc162904362)

[5.6.1 Ada Boosting Algorithm 17](#_Toc162904363)

[5.6.2 Gradient Boosting Algorithm 18](#_Toc162904364)

[5.7 Multi-Layer Perceptron (MLP) 19](#_Toc162904365)

[5.7.1 Algorithm 19](#_Toc162904366)

[6 Results And Analysis 21](#_Toc162904367)

[7 Conclusion 22](#_Toc162904368)

[8 Future Scope 22](#_Toc162904369)

[9 References 22](#_Toc162904370)

**Table of Figures**

[Figure 1. Percentage Severity Distribution 9](#_Toc162904895)

[Figure 2. Correlation Heatmap 10](#_Toc162904896)

[Figure 3. Traffic Severity during Day and Night 11](#_Toc162904897)

[Figure 4. Traffic Severity during different weather conditions 11](#_Toc162904898)

[Figure 5. NBC - Train Accuracy 12](#_Toc162904899)

[Figure 6. NBC - Test Accuracy 12](#_Toc162904900)

[Figure 7. SVM – Train Accuracy Score 13](#_Toc162904901)

[Figure 8. SVM – Test Accuracy Score 13](#_Toc162904902)

[Figure 9. Logistic Regression – Train and Test Accuracy Score 14](#_Toc162904903)

[Figure 10. Decision Tree Train and Test Accuracy Score 15](#_Toc162904904)

[Figure 11. Decision Tree (Criterion – Gini) Train and Test Accuracy Score 15](#_Toc162904905)

[Figure 12. Decision Tree (Criterion – Entropy) Train and Test Accuracy Score 15](#_Toc162904906)

[Figure 13. Random forest (Criterion – Gini) Train and Test Accuracy Score 16](#_Toc162904907)

[Figure 14. Random forest (Criterion – Gini) Confusion Matrix 16](#_Toc162904908)

[Figure 15. Random forest (Criterion – Entropy) Train and Test Accuracy Score 17](#_Toc162904909)

[Figure 16. Random forest (Criterion – Entropy) Confusion Matrix 17](#_Toc162904910)

[Figure 17. ADA Boosting Train & Test Accuracy Score and Confusion Matrix 18](#_Toc162904911)

[Figure 18. Gradient Boosting Train & Test Accuracy Score and Confusion Matrix 19](#_Toc162904912)

[Figure 19. Gradient Boosting - Loss VS Number of Iterations 20](#_Toc162904913)

[Figure 20. MLP Train and Test Accuracy Score 21](#_Toc162904914)

[Figure 21. MLP Confusion Matrix 21](#_Toc162904915)

[Figure 22. Models Accuracies 22](#_Toc162904916)

1. Abstract

With the advancement and modernization of technology, various modes of transportation are rapidly evolving to reduce the time of travel. But contrary to the advancement, the road mode of transport has always been considered important. Even though the distance is short, the frequency of crashes is considerably high due to the highly concentrated traffic. Road Accidents have a substantial economic impact. However, their effects on lost lives are more significant. The National Highway Traffic Safety Administration released its latest counts estimating 19,515 people died in motor vehicle traffic crashes, for the first half of 2023. 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 accident 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.

1. Literature Survey

The various research papers about Traffic Severity Analysis were reviewed, which used the following models:

* 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. 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.

* 1. 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 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. 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 property damage) 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.

* 1. 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 pre-processing 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:

Yi(wT ϕ(xi) + 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.

* 1. 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 24 months from January 2002 to December 2003. As part of pre-processing, the dataset was divided into both categorical and continuous data and the 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 of 54.73% and 40.56% respectively, with 0.3478 of mean absolute error. The next model trained was a Decision Tree algorithm, which used 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.

1. Dataset
   1. Dataset Details

In this project, the dataset used is a countrywide traffic accident dataset available on [Kaggle.](https://www.kaggle.com/datasets/sobhanmoosavi/us-accidents)

It comprises 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.

* 1. Data Pre-processing Techniques

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

* + 1. Handling missing values

All the NULL value entries in our dataset were filtered, which were around 10,000 in total, and all such entries were deleted.

* + 1. Handling duplicate values

The dataset contained around 5,000 repeated entries. All the duplicates were deleted to make all rows unique.

* + 1. 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, filtered only entries from 2016 to 2018, bringing down the number of rows to around 3,00,000.

* + 1. Encoding Categorical Variables

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

* + 1. Splitting dataset into Training and Testing set

We divided the number of entries into Training and Testing sets in the ratio 80 : 20.

* + 1. Handling of Outliers

Outliers have been removed with a threshold of 1.5 from all the columns.

* + 1. Feature scaling

We scaled the features in our dataset to the same range, so no feature dominates over the others. We used Standardization using the StandardScaler class of sklearn.preprocessing library to do so.

* 1. Data Inferences

A pie chart with numbers and a number on it

Description automatically generated

Figure 1. Percentage Severity Distribution

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.

A screenshot of a computer screen

Description automatically generated

Figure 2. Correlation Heatmap

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.

A graph with blue and white stripes

Description automatically generated

Figure 3. Traffic Severity during Day and Night

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

A graph of different weather conditions

Description automatically generated

Figure 4. Traffic Severity during different weather conditions

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

1. Methodology - Model Details

We have experimented with the following machine-learning models:

* 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 Naive Bayes Classification (NBC) and used scikit learn for testing purposes such as f1 score from sklearn.metrics.

* Upon running the train on the initiated class for NBC, the model counts the required parameters conditioned on each value of the output stores them for categorical columns and stores the relevant values of mean and std for Numerical data.
* The predict function takes in the row for which the prediction is to be performed
* It checks which category a particular column lies. If it’s a category, then Naive Bayes is applied using the Naive formula along with a Laplacian method with alpha =5. If it’s a numeric category, then it applies the relevant Gaussian model.
* Predict alpha takes a custom alpha to find the prediction
* The accuracy score takes in the test set and its ground truth and for each row in the 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 Naive Bayes.

A screenshot of a computer code

Description automatically generated

Figure 5. NBC - Train Accuracy

A screenshot of a computer code

Description automatically generated

Figure 6. NBC - Test Accuracy

* 1. 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 the 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, use 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 the model.
* We use the cross-validation technique to determine the performance of the model.

A black text on a white background

Description automatically generated

Figure 7. SVM – Train Accuracy Score

A screenshot of a computer code

Description automatically generated

Figure 8. SVM – Test Accuracy Score

* 1. Logistic Regression
     1. Algorithm
* The first model runs on the dataset with only 2-3 severity ratings 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.

A screenshot of a computer

Description automatically generated

Figure 9. Logistic Regression – Train and Test Accuracy Score

* 1. Decision Trees and Random Forests
     1. Algorithm
* The model reads the input data upon calling the fit function.
* If it is at the node, then the split is made such that Info Gain is the highest. Otherwise, it is at the leaf that checks if all the data is being classified properly or not.
* If the classification is incorrect, then the algorithm makes a split continue if the termination condition is not being fulfilled.
* The algorithm stops on reaching the termination condition.

A number of numbers on a white background

Description automatically generated

Figure 10. Decision Tree Train and Test Accuracy Score

A number with numbers on it

Description automatically generated with medium confidence

Figure 11. Decision Tree (Criterion – Gini) Train and Test Accuracy Score

A number with black text

Description automatically generated with medium confidence

Figure 12. Decision Tree (Criterion – Entropy) Train and Test Accuracy Score

* 1. Random Forests
     1. Algorithm
* 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

A screenshot of a computer

Description automatically generated

Figure 13. Random forest (Criterion – Gini) Train and Test Accuracy Score

A blue squares with white numbers and a blue bar

Description automatically generated

Figure 14. Random forest (Criterion – Gini) Confusion Matrix

A white paper with numbers and black text

Description automatically generated

Figure 15. Random forest (Criterion – Entropy) Train and Test Accuracy Score

A blue squares with white numbers and numbers

Description automatically generated

Figure 16. Random forest (Criterion – Entropy) Confusion Matrix

* 1. Boosting
     1. Ada Boosting Algorithm
* 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

A screenshot of a graph

Description automatically generated

Figure 17. ADA Boosting Train & Test Accuracy Score and Confusion Matrix

* + 1. Gradient Boosting Algorithm
* The primary aim of this boosting method is to decrease the loss function

A screenshot of a graph

Description automatically generated

Figure 18. Gradient Boosting Train & Test Accuracy Score and Confusion Matrix

A graph with a line

Description automatically generated

Figure 19. Gradient Boosting - Loss VS Number of Iterations

* 1. 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 MLP Classifier from the sklearn.neural network library.
* Grid search was performed using different activation functions, including ‘tanh’, ‘relu’, ‘logistic’, and ‘identity’ on a 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.

A screenshot of a test

Description automatically generated

Figure 20. MLP Train and Test Accuracy Score

A blue squares with white numbers and black text

Description automatically generated

Figure 21. MLP Confusion Matrix

1. Results And Analysis

We implemented the following models: Naive Bayes Classifier, which makes use of Laplace and weighted correction, Support Vector Machine, which takes in various kernel functions like linear and Radial Basis Function (RBF), logistic regression classifier 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 finally a Multi-Layer Perceptron.

A table with numbers and text

Description automatically generated

Figure 22. Models Accuracies

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 Naive 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 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 Neighbours 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, the Support Vector Machine and Mixed Naive Bayes also gave a decent accuracy of 0.681 and 0.66 respectively.

1. Conclusion

* 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.
* Among classification models, Random Forest demonstrated very high accuracy among the models tested, achieving 89% accuracy on the test dataset.
* Algorithm boosting algorithms trained on different decision trees also gave great results. For example, we were able to achieve 91% accuracy by using XGBoost.

1. Future Scope

* We will need to save and restore/reload later our ML Model to test our model with new data or to compare multiple models or anything else. Hence, serialization and deserialization of models are required.
* 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.
* Balancing of the dataset may yield a better result and it has to be performed on the dataset preferably Synthetic Minority Oversampling Technique (SMOTE) algorithm which works by randomly picking a point from the minority class and computing the k-nearest neighbors for this point.

1. 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.