Using Machine Learning to Predict Traffic Accidents

Lewis Quick – *22016949*, William Forber – *22015706*, Yamin Shwe Yi Htay - *23019880*

Abstract

In this study, we attempted to predict the severity of traffic accidents in Bristol, UK using machine learning classification. We were aiming to provide a classification between slight and severe accidents, based on the number of vehicles involved in the incident. After selecting Support Vector Machines and Random Forests as our algorithms to use, we tuned the parameters of each to obtain the strongest classification possible. The Random Forest model provided the best performance, with an average test accuracy of 0.609. These results were not as conclusive as we had hoped, but factors of the dataset contributed towards this.

Introduction

As recently as 2022, road traffic accidents attributed to 135,480 casualties in the UK (GOV.UK, 2023). This remains a significant number of people, despite the trend being a general decrease over recent decades. Machine Learning technologies enable versatile classification of many different problems, including humanitarian issues such as this. In this study, we aim to identify factors of traffic accidents and their severity in Bristol, UK using machine learning methods. We aimed to predict the severity of a car accident depending on the number/type of vehicles involved.

Related Work

Support Vector Machines (SVM)

Breast Cancer Prediction (Min-Wei, H., et al., 2017)

This study used SVMs to predict breast cancer susceptibility, focusing on different configurations and kernels, including linear, polynomial, and RBF kernels. The study found linear SVM with bagging worked best for smaller datasets. This aligns with our project, as it also uses a relatively small dataset.

Mountain Freeways in China (Li, J., et al., 2023)

This study applied SVMs to predict the accident severity on mountain freeways in Yunnan Province, China. It also used Random Forest feature selection to boost accuracy, with Ada_SVM achieving high precision and accuracy.

Artificial Neural Networks (ANN)

Road Traffic Accident Prediction (Gatarić, D., et al., 2023)

This study predicted road traffic accidents in Eastern Europe, comparing ANN to logistic regression. The ANN showed positive results, with road length being a significant factor in predicting accidents. In contrast, our project does not use ANN, focusing on SVM for smaller datasets.

Deep Neural Network (Formosa, N., et al., 2020)

This study employed a Deep Neural Network (DNN) to predict accidents based on vehicle telemetry data, using complex inputs such as speed and distance from other vehicles. This study's use of DNN differs from our project, which primarily uses text-based data and focuses on SVM.

Random Forest and Other Algorithms

UK Traffic Accident Prediction (Obasi, I.C. and Benson, C., 2023)

This study analysed 10 years of UK traffic accident data, using Random Forest, Naive Bayes, Logistic Regression, and ANN. Random Forest and Logistic Regression outperformed the other models, achieving high accuracy rates. This study also performed feature importance analysis using Random Forest, identifying key factors like engine capacity and vehicle age. Compared to our project, this study has a broader dataset and uses various machine learning algorithms.

Indian Highways Accident Prediction (Khanum, H., Garg, A. and Faheem, M.I., 2023)

This study used Random Forest to predict accident severity on Indian highways, achieving moderate accuracy on the training set but lower accuracy on the test set due to possible data imbalances. Our project focuses on SVM with a smaller dataset and k-fold cross-validation to prevent overfitting, unlike the Indian study.

Traffic Accident Severity in China (Yang, J., Han, S. and Chen, Y., 2023)

This study used a large Chinese dataset to predict traffic accident severity, achieving high accuracy with the Random Forest algorithm. It highlighted collision patterns and car structure as key factors, recommending improvements in road infrastructure and driver training. This study's broader scope and higher accuracy with larger datasets differ from our project's more localized focus with SVM on a smaller dataset.

Compared to these studies, the major differences between all these cases and our project include:

Dataset Size: Many studies use large datasets covering broader regions (like China, India, and the UK). In contrast, our project uses a smaller dataset from Bristol, UK.

Machine Learning Algorithms: While other studies use various algorithms (such as Random Forest, ANN, Logistic Regression, and DNN), our project focuses primarily on SVM and Random Forest emphasizing their effectiveness with smaller datasets.

Scope and Focus: Other studies often have a broader scope, exploring various factors and environmental conditions. Our project is localized, focusing on a specific city with a smaller dataset.

Evaluation Techniques: Our project uses k-fold cross-validation to prevent overfitting, while other studies might use different evaluation techniques, such as feature importance analysis or SHAP analysis for model interpretation.

Data

The dataset we used for this project was available from the Bristol Council website (Open Data Bristol, 2017). It is a CSV file containing over 4000 records of Bristol traffic accidents between 2017-2021. Each record represents an incident, with several fields to represent details:

Field	Description
Date	Date of accident.
Time	Time of accident.
Severity	Integer value of severity. (3, 2, 1).
Severity Description	Label of severity. (Slight, severe, fatal).
Accident Type	Accident code. (LC, A, HO, …)
Accident Description	Description of accident code. (Loss of Control, Adult Pedestrian, Head On, …).
Vehicles	Number of vehicles involved.
Casualties	The number of casualties involved.
Pedestrian	The number of pedestrians involved.
Cycles	Number of cycles involved.
MCycles	The number of motorcycles involved.
Children	Number of children involved.
OAPs	Number of OAPs involved.
X	X coordinates of location.
Y	Y coordinates of location.
Render	The main cause of the accident. (Cars, Cyc, A, …) {Cars, Cycles, Adult pedestrian, …} e.g. Many different accident types can be attributed to 'Cars'.

Pre-processing the data was required since we were predicting the severity depending on the number and types of vehicles involved. To adjust the data to the classification problem we were solving, we removed all columns other than:

- Severity.
- Number of vehicles.
- Number of pedestrians involved.
- Number of cycles involved.
- Number of motorcycles involved.

As is accustomed to pre-processing data, we also checked for any null values in the dataset to ensure that the models were not trained on erroneous data.

To create a classification on the severity of incidents, we found it best to convert the 3 categories of severity into a binary classification. Seeing as there were very few records classed as 'fatal', we combined these with the much more abundant 'serious' records to reduce the number of severity categories to 2. This could in turn be made into a binary classification – an accident would be classed as slight or 'not slight' (serious/fatal). This is optimal for training machine learning algorithms as it's a very simple classification. Furthermore, we identified that since there was a large imbalance in the dataset, with 3861 slight accidents and only 405 other examples. This would have caused misclassifications of severe accidents. Therefore, we under-sampled the dataset removing slight cases until there was an even split between severe and slight accidents.

Methods

In consideration of the related works, we found that Support Vector Machines, Random Forests and Artificial Neural Networks should be considered for our research.

Support Vector Machine

A Support Vector Machine is "a supervised machine learning algorithm that classifies data by finding an optimal line or hyperplane that maximizes the distance between each class in an N-dimensional space" (IBM, 2023). SVMs are essentially an extension of logistic regression, which is one of the earliest ideas of classification algorithms. They are known for their effective classification across small to medium-sized datasets, via a relatively simple implementation. Large datasets are to be avoided, as the algorithm can be computationally expensive. This is particularly true when the classification is completed on a multi-dimensional dataset. To improve this performance, most SVMs utilise kernels. Kernels supplement SVMs with mathematical functions in the algorithm which simulate distances between data points, rather than an actual calculation of said distances. Different kernels are found to produce different results, which is why kernel selection is a key part of SVM experimentation.

Random Forest

An ensemble is a collection of classifiers each trained using different parts of a dataset, resulting in an aggregation of the output of each of the classifiers. A common type of ensemble method is a Random Forest Classifier. This method utilises an ensemble of **Decision Trees** and combines their results with the ambition of producing a classification (IBM, 2023). The nuance in the method is in the behaviour of the decision trees, and how the data is handled between the trees. Random Forests are known for their low computational cost and effectiveness on relatively simple classifications. This makes them a good fit for quick and easy classification of small to medium datasets.

Artificial Neural Network

ANNs are a highly sophisticated classification methodology. They consist of multiple 'layers' which represent the data flow. Between an input and output layer, there are many 'hidden layers'. The specification of these hidden layers dictates the model's output. For example, certain hidden layers can be set to analyse certain portions of the input. Traits like this make the methodology useful for the classification of highly complex input, such as images. ANNs are best used on larger datasets, as they've been seen to overfit with too little data (BuiltIn, 2023).

In consideration of the previous 3 methods, we chose to use SVMs and Random Forest classifiers on our dataset. The ANNs would have proved to be a lengthy and difficult implementation and would have yielded sub-par results on our small dataset.

Grid Search

To optimize the performance of both SVM and Random Forest models we used grid search, which is a common technique for identifying the best hyperparameters. Grid search involves defining a range of potential values for various hyperparameters and systematically exploring these combinations to identify the optimal set of variables. Considering it is computationally expensive, grid search is only viable for a relatively small number of possible variables. However, it must be said that it is far more effective than a manual 'trial and error' test of the variables.

For each SVM kernel variant (linear, polynomial, and Radial Basis Function), we defined parameter grids for the key hyperparameters. These typically included the regularization (C), the degree of polynomial kernel, and the gamma parameter for non-linear kernels. By conducting a comprehensive search across different values, we aimed to identify the combination that yielded the best performance.

The grid search process was paired with cross-validation to ensure that the tuning was robust and not overly reliant on specific data splits. This approach helped us identify hyperparameters that consistently performed well across different training and testing datasets, leading to models with greater accuracy and reduced risk of overfitting.

Performance Evaluation

K-fold Cross Validation

K-fold cross-validation is where the dataset is split k-number of times and the model is trained on each split dataset (Anguita *et al.,* 2012), the results are then averaged giving a more stable and reliable estimate of model accuracy and precision. Using k-fold cross-validation helped us ensure that our evaluation process was robust and minimized the risk of overfitting, which is when a model performs well on training data but poorly on unseen data and allowed us to compare how each model performs with different data and consequently how the model would perform in a real-world scenario.

Confusion Matrix

We used a confusion matrix coupled with k-fold cross-validation to evaluate different models. A confusion matrix is used to measure the performance of a classifier numbering the occurrences of true positives against false positive cases and true negatives against false negative cases. This allowed us to easily compare the classification performance of each model.
To enhance our evaluation, we calculated additional metrics like precision, recall, F1-score, and accuracy. Precision shows the proportion of correct positive

predictions, recall reflects how many actual positives were identified, and the F1-score balances precision and recall. Accuracy provides an overall measure of correct predictions.

Languages & Libraries

For software tools and libraries, we used Python as the primary language, with a range of supporting libraries for data analysis, machine learning, and data visualization. Python's '**Scikit-learn**' served as a highly useful module for all aspects of machine learning. It provides easy application of classifiers like '**SVC**', '**RandomForestClassifier**', and '**AdaBoostClassifier**', offers utilities for data splitting ('**train_test_split'**), hyperparameter tuning('**GridSearchCV**'), and model evaluation ('**confusion_matrix**', '**accuracy_score**', and '**classification_report**'). For data manipulation, we used '**pandas**', which allowed us to pre-process, filter, and manage the dataset effectively. '**Numpy'** was employed for numerical operations, while '**seaborn**' and '**matplotlib**' were used for data visualization and plotting results. Together, these tools created a comprehensive and efficient environment for our project, enabling us to build, evaluate and optimise our models with flexibility and ease.

Ethical Considerations

When working with data related to traffic accidents, ensuring ethical standards is critical. Our project used a dataset from the Bristol Council, which included details about traffic accidents, such as date, time, severity, vehicle types, and casualties. Given the potential sensitivity of this information, we took several steps to ensure the data was anonymized and that no personally identifiable information (PII) could be used to discriminate against any individual or group.

Anonymization of Data (Regulation (EU) 2016/679 of the European Parliament and of the Council, 2016)

Any personal information, such as names, addresses, or vehicle registration numbers, was not included in the dataset. While location data was present, it was generalized to avoid pinpointing specific addresses or exact locations where accidents occurred. Any demographic data that could identify individuals, such as age or gender, was not described to prevent discrimination against specific groups.

Avoiding Discrimination

The data was processed and analysed without reference to protected characteristics, focusing solely on factors related to traffic accidents. We avoided any analysis that could lead to biased conclusions or discriminatory practices (e.g. We did not use demographic data to conclude accident causes.). The project complied with ethical guidelines and regulations, ensuring that all data handling, processing, and analysis were conducted concerning privacy and non-discrimination.
By focusing on anonymization and avoiding discrimination, we aimed to contribute to a broader understanding of traffic accident patterns while respecting ethical considerations.

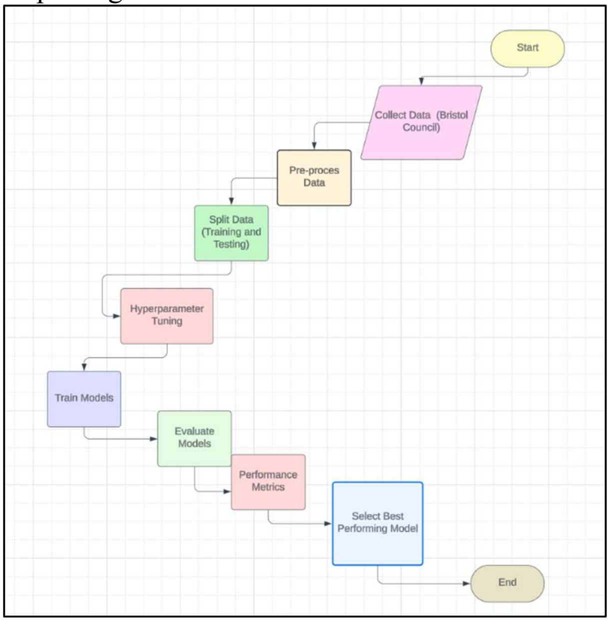

Fig 1.0 – The machine learning workflow which we planned to follow.

Experiments

We experimented using different kernels for the Support Vector Machine and comparing them to a Random Forest classifier.

We started by training each model using *train_test_split* from *Sklearns*, which splits the dataset into training and test data, the test data to be used to evaluate each model. Furthermore, the random state parameter was set to 0 so we could tune each model using the same data to get repeatable results. Afterwards, we used the *make_pipeline* feature from *Sklearns* and scaled the data using the standardization technique. To get a control baseline each model was evaluated with no hyperparameters specified using the test data mentioned previously. We then started tuning the hyperparameters for each model.

Hyperparameter Tuning

Linear Kernel SVM

The linear kernel Support Vector Machine only hyperparameter that can be tuned is the **regularization** parameter (how much error is allowed). We started by running a grid search which incremented the regularization parameter in factors of 10, which allowed us to narrow down on a range of values.

```
parameter_grid = {'C': [0.001, 0.01, 0.1, 1, 10, 100, 1000]
```

Here, the best value for the hyperparameter was 1. With this information, we used the *linspace* feature from *Numpy* to search through 100 values in the range 0.1 to 10.

```
{'C': np.linspace(0.1, 10, num=100, dtype=float)}
```

This search found that the best value was:

- **regularization, C** = 3.

Polynomial SVM

This kernel is affected by more than one hyperparameter, being the **regularization**, **degree** (the complexity of the model), and **gamma** (kernel co-efficient). Since there were more parameters to tune than the previous method, to get the best possible hyperparameters using *linspace* will simply take too long to complete. Therefore, we instead created a grid search with the **regularization** and **gamma** hyperparameters incrementing in factors of 10 and the degree hyperparameter incrementing in steps of 1.

```
{'C': [0.001, 0.01, 0.1, 1, 10, 100],
'degree': [1, 2, 3, 4],
'gamma': [0.01, 0.1, 1, 10]}
```

This grid search found that the best hyperparameters for the polynomial kernel were:

- **regularization, C** = 0.01
- **degree** = 2
- **gamma** = 1

Radial Basis Function SVM

The RBF SVM has two hyperparameters which can be adjusted, being the **regularization** and **gamma**. This means that we can follow the same method as for the linear kernel for tuning the hyperparameters, starting with narrowing down the range of possible best values and finishing using *linspace* to find the best possible combination. So, we started by creating a grid search to search the **regularization** and **gamma** parameters incrementing in factors of 10.

```
{'C': [0.01, 0.1, 1, 1, 10, 100, 1000],
'gamma': [0.001, 0.01, 0.1, 1, 10, 100, 1000]}
```

This search found the best combination to be **C** = 10, and **gamma** = 0.1. We then used *linspace* feature to generate 100 values between 1 and 100 for **C**, and between 0.01 and 1 for the **gamma**.

```
{'C': np.linspace(1, 100, num=100, dtype=float)
'gamma': np.linspace(0.01, 1, num=100, dtype=float)}
```

This grid search found that the best hyperparameters for the RBF SVM were:

- **regularization, C** = 18.
- **gamma** = 0.09.

Random Forest

Random Forests have 4 different hyperparameters to tune, being **n_estimators** (number of trees in the forest), **max_features** (the number of features to consider when looking for the best split), **max_depth** (the maximum depth of the tree) and **max_leaf_nodes** (grows the tree by the number of max_leaf_nodes).

Because there are many hyperparameters in Random Forests, it was infeasible to use *linspace* as it would have taken too much time. We used a grid search with **n_estimators** in increments of 25, with **max_depth** and **max_leaf_nodes** searched through in increments of 3.

```
'n_estimators': [25, 50, 100, 150],
'max_features': ['sqrt', 'log2', None],
'max_depth': [3, 6, 9]
'max_leaf_nodes': [3, 6, 9],
```

The results from this search concluded that the best parameters for the Random Forest classifier were:

- **max_depth** = 3.
- **max_features** = 'sqrt'.
- **max_leaf_nodes** = 9.
- **n_estimators** = 25.

Performance Evaluation

To evaluate the models, we used k-fold cross-validation coupled with a confusion matrix (see more detail in Methods). We focused the most on the f1-score, which is the combined average of precision and recall (GeeksForGeeks, 2023), and average test accuracy.

Results

Model	F1 score positive	F1 score negative	Avg. Test Accuracy
Linear SVM	0.59	0.64	0.584
Polynomial SVM	0.59	0.62	0.584
RBF SVM	0.65	0.62	0.602
Random Forest	0.64	0.65	0.609

The best model was the Random Forest classifier (ensemble model), producing the highest f1-score of 0.65 for negative classifications and 0.64 for positive classifications. The Random Forest classifier also had the highest average test accuracy of 0.609.

The runner-up model was the Radial Basis Function Support Vector Machine, which had a lower f1-score of 0.62 for negative classifications and 0.65 for positive classifications. The model had a lower test average accuracy of 0.602.

The worst model was the polynomial SVM, producing the lowest f1-score of 0.62 for negative classifications and 0.59 for positive classifications. This was coupled with the tied lowest test average accuracy of 0.584 with the linear kernel SVM, which had a higher f1-score for negative classifications of 0.64 but had the same f1-score for positive classifications of 0.59.

Conclusion

Through the use of both SVMs and Random Forest classifiers, we were unable to make as clear of a classification as we'd have liked. This is even though both models were optimised effectively using grid search. However, the models did produce correct classifications for the majority of cases.

A lesson we have learned is to pay closer attention to the dataset, and the range of data which it holds. The results of the models may have been negatively affected by the under-sampling used on the dataset. While this was necessary for the training on this dataset, ideally it would have been avoided. Clearly, it is best to have the highest number of cases possible when training a model, as small datasets are always prone to producing inaccurate or invalid results.

It is unlikely that our choice of algorithm affected the results. An ANN would have likely suffered more than our SVM and RF because, as mentioned in Methods, they are prone to overfitting on small datasets.

References

Anguita, D., Ghelardoni, L., Ghio, A., Oneta, L. and Ridella, S. (2012) The 'K' in K-fold Cross Validation. *Esann 2012 Proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning* [online]. [Accessed 15 April 2024].

BuiltIn (2023) 4 Disadvantages of Neural Networks. Available from: https://builtin.com/data-science/disadvantages-neural-networks [Accessed 15/03/2024].

GeeksforGeeks (2023) F1 Score in Machine Learning. Available from: https://www.geeksforgeeks.org/f1-score-in-machine-learning/.

Formosa, N., Quddus, M., Ison, S., Abdel-Aty, M. and Yuan, J. (2020) Predicting Real-time Traffic Conflicts Using Deep Learning. *Accident Analysis & Prevention* [online]. 136 [Accessed 23 April 2024].

Gatarić, D., Ruškić, N., Aleksić, B., Đurić, T., Pezo, L., Lončar, B., Pezo, M. (2023) Predicting Road Traffic Accidents - Artificial Neural Network Approach. *Algorithms* [online]. pp. 257. 16 (5). [Accessed 17 April 2024].

GOV.UK (2023) Reported road casualties Great Britain, annual report: 2022. Available from: https://www.gov.uk/government/statistics/reported-road-casualties-great-britain-annual-report-2022/reported-road-casualties-great-britain-annual-report-2022#:~:text=In%20reported%20road%20collisions%20in,of%2012%25%20compared%20to%202019 [Accessed 23/04/2024].

IBM (2023) What are Support Vector Machines (SVMs)? Available from: https://www.ibm.com/topics/support-vector-machine [Accessed 30/04/2024].

IBM (2023) What is Random Forest? Available from: https://www.ibm.com/topics/random-forest [Accessed 30/04/2024].

Khanum, H., Garg, A. and Faheem, M.I. (2023) Accident Severity Prediction Modelling For Road Safety Using Random Forest Algorithm: *An Analysis of Indian Highways*. F1000research [online]. 2 [Accessed 23 April 2024].

Li, J., Guo, F., Zhou, Y., Yang, W. and Ni, D. (2023) Predicting the Severity of Traffic Accidents on Mountain Freeways with Dynamic Traffic and Weather Data. *Transportation Safety and Environment* [online]. 5 (4) [Accessed 23 April 2024].

Min-Wei, H., Chen, C., Wei-Chao, L., Shih-Wen, K. and Chih-Fong, T. (2017) SVM and SVM Ensembles in Breast Cancer Prediction. *PLoS One* [online]. 12 (1) [Accessed 16 April 2024].

Obasi, I.C. and Benson, C. (2023) Evaluating the Effectiveness of Machine Learning Techniques in Forecasting the Severity of Traffic Accidents. *Heliyon* [online]. 9 (8) [Accessed 23 April 2024].

Open Data Bristol (2017) Traffic Accidents. Available from: https://opendata.bristol.gov.uk/datasets/bcc::traffic-accidents-1/about [Accessed 05/03/2024].

Regulation (EU) 2016/679 of the European Parliament and of the Council [online]. Chapter 1. (2016) legislation.gov.uk. Available from: https://www.legislation.gov.uk/eur/2016/679/contents#/ [Accessed 04 April 2024].

Yang, J., Han, S. and Chen, Y. (2023) Prediction of Traffic Accident Severity Based on Random Forest. *Journal of Advanced Transportation* [online]. [Accessed 23 April 2024].