PREDICTING THE SEVERITY OF ROAD ACCIDENTS IN THE UK

EXECUTIVE SUMMARY

The objective of this report is to forecast the severity of road accidents in the UK employing machine learning methodologies. The goal is to create a system that assists emergency services in responding more efficiently to accidents. The supplied dataset encompasses information regarding accidents in 2019, their severity, and additional variables that could be beneficial for predicting severity.

To address this issue, I utilized both conventional machine learning algorithms and neural networks, comparing their effectiveness. I employed various metrics to assess the models, including accuracy and precision.

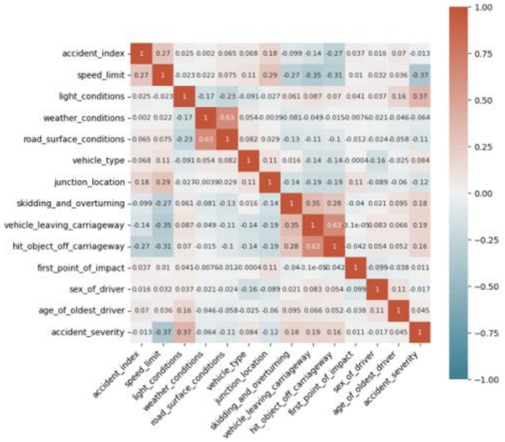
The findings indicate that both conventional machine learning algorithms and neural networks can attain high accuracy in classifying accident severity. However, I discovered that neural networks slightly outperform traditional algorithms, particularly in classifying the most severe accidents.

1. EXPLORATORY DATA ANALYSIS

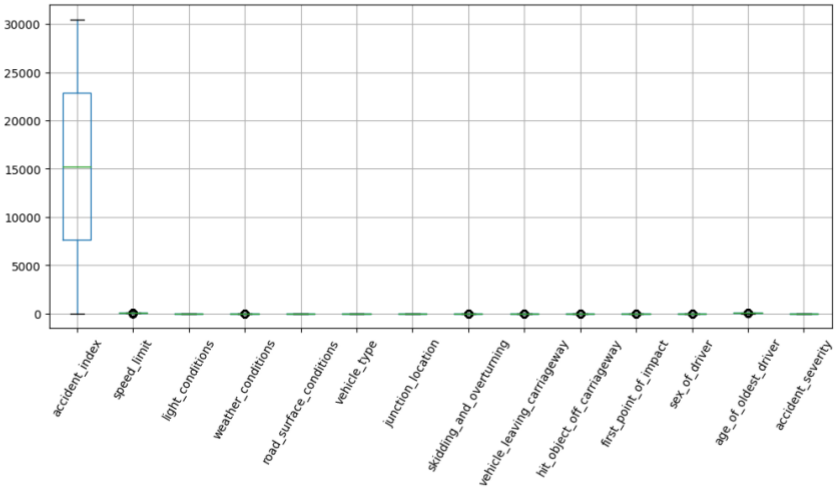
Before creating a machine learning model, exploratory data analysis (EDA) is an essential step in understanding the dataset and its properties. It entails analyzing the distribution, connections, and trends of the data, which can affect the model selection and preprocessing methods.

As part of the EDA, I performed the following tasks:

- Data Cleaning: I checked for missing values, duplicates, and outliers, then handled them accordingly.
- Descriptive Statistics: To understand the central tendency and dispersion of each feature, I generated summary statistics including mean, median, mode, and standard deviation.
- Visualizations: I created a Line chart, boxplot, heatmap and Scatterplot observe the distribution of each feature and the correlations between features.

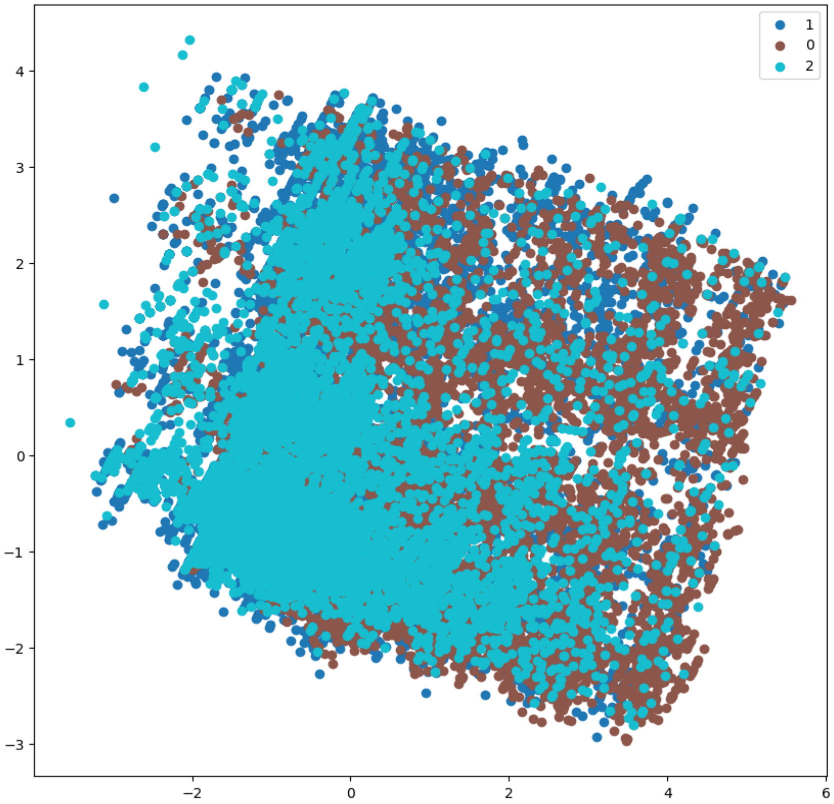


Heatmap showing the correlation



Boxplot chart showing the distribution

In order to visualise the dataset in a 2-dimensional plot, I used dimensionality reduction techniques. The high-dimensional dataset was projected onto a 2D plane using Principal Component Analysis (PCA), preserving as much variance as possible. I applied PCA to build a 2D plot that shows the data points and target labels for each data point. (accident severity: fatal, serious, slight).



PCA Scatterplot

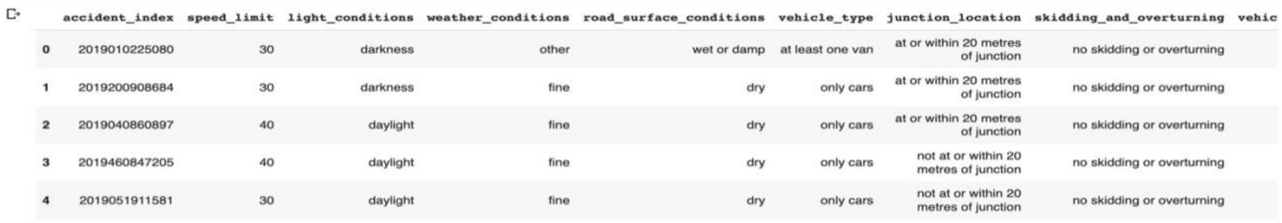
From the above plot, I could observe that the classes were heavily overlapping, this suggests that the classification task might be more challenging and may require more sophisticated models or feature engineering.

2. DATA PREPROCESSING

Pre-processing is the process of transforming and cleaning raw data to make it suitable for use in machine learning algorithms **(George Lawton, 2021).**

THE DATASET:

The data given contains detailed information about UK accidents in 2019, it has 31,647 rows and 14 column and some of the information include; Weather conditions, Age, Speed limit, Vehicle type, etc.



	accident_index	speed_limit	light_conditions	weather_conditions	road_surface_conditions	vehicle_type	junction_location	skidding_and_overturning	vehic
0	2019010225080	30	darkness	other	wet or damp	at least one van	at or within 20 metres of junction	no skidding or overturning	
1	2019200908684	30	darkness	fine	dry	only cars	at or within 20 metres of junction	no skidding or overturning	
2	2019040860897	40	daylight	fine	dry	only cars	at or within 20 metres of junction	no skidding or overturning	
3	2019460847205	40	daylight	fine	dry	only cars	not at or within 20 metres of junction	no skidding or overturning	
4	2019051911581	30	daylight	fine	dry	only cars	not at or within 20 metres of junction	no skidding or overturning	

The dataset

In this work, I performed several data preprocessing steps to prepare the data for the machine learning task. The steps taken for this task are given below;

INSPECT THE DATA:

I first inspected the dataset to understand its structure, size, and basic statistics such as mean, median, number of rows, and columns. This initial inspection provided me with valuable insights into the nature of the data and potential preprocessing steps required.

DATA CLEANING:

To correct any discrepancies, errors, or flaws in the dataset, I performed data cleaning. This process ensures the accuracy of the modelling data and lessens the possibility of inaccurate outcomes. Some of the steps done in data cleaning includes;

- Fixing the details in the "accident_severity" table; changing words like "Fatal" and "fatal" to just one word "fatal".
- Converting words writing as "data missing or out of range" to "np.nan".

HANDLING MISSING DATA:

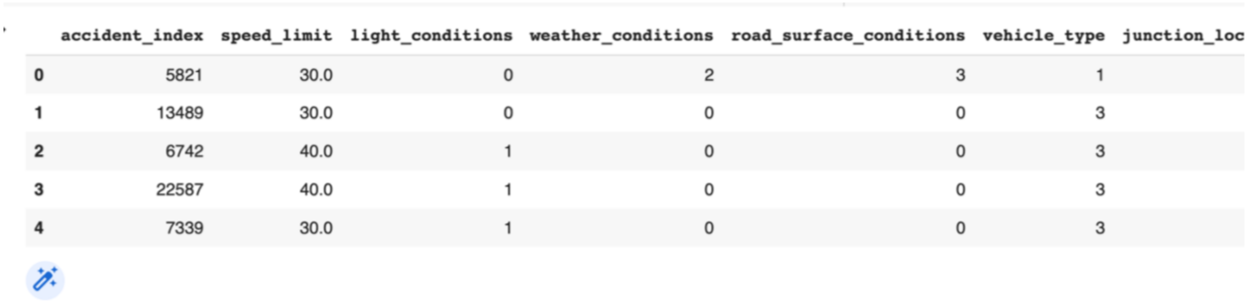
I used the “Simple Imputer” from “sklearn.impute” to handle missing values in the dataset. For categorical features, I replaced missing values with the “most frequent” category, while for numerical features, I replaced missing values with the “median”. This approach helps maintain the distribution of the data and prevents the introduction of biases.

REMOVING DUPLICATE ROWS:

During my data cleaning process, I identified and removed 1,172 duplicate records from the dataset. Removing duplicates is essential to prevent overfitting and ensure the model generalizes well to new data.

ENCODING CATEGORICAL DATA:

I applied label encoding to convert categorical features into binary features. This step is necessary because most machine learning algorithms work with numerical data. Label encoding is a common technique for encoding categorical variables.



	accident_index	speed_limit	light_conditions	weather_conditions	road_surface_conditions	vehicle_type	junction_loc
0	5821	30.0	0	2	3	1	
1	13489	30.0	0	0	0	3	
2	6742	40.0	1	0	0	3	
3	22587	40.0	1	0	0	3	
4	7339	30.0	1	0	0	3	

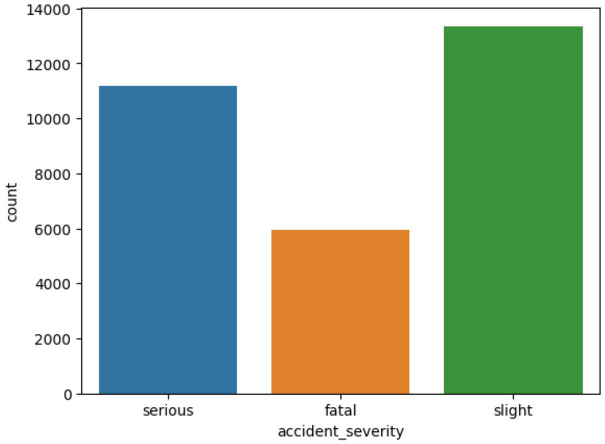
The encoded data

FEATURE SCALING:

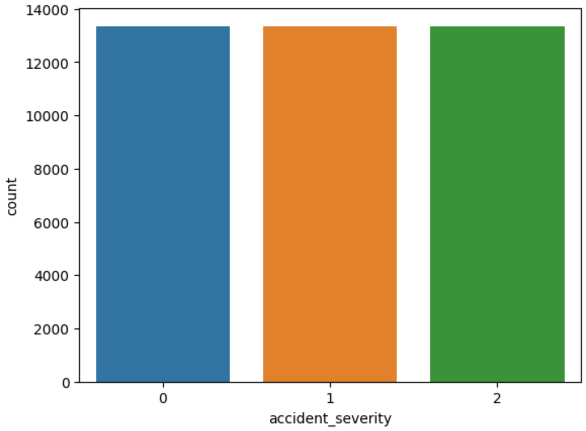
I used the “StandardScaler” from “sklearn.preprocessing” to scale the numerical features. This step is crucial for algorithms that are sensitive to the scale of the input features. Standard scaling transforms the features to have zero mean and unit variance, making the data more comparable across different features.

OVERSAMPLING:

I used oversampling to balance the distribution of the target variable in the training set. Synthetic Minority Over-sampling Technique (SMOTE) is used to resolve any class imbalances in the dataset. This action helps the model perform better when applied to underrepresented classes.



Target features before oversampling



Target features after oversampling

TRAIN-TEST SPLIT:

To classify the accident severity, I had to split the dataset into independent and dependent variables. The dependent variable "y" is the accident severity, and the independent variable "x" is the other features/attributes (excluding the accident index). In order to provide the machine with some attributes to train with and use the experience for classification, I then had to split the divided dataset into train and test splits.

3. CLASSIFICATION USING TRADITIONAL MACHINE LEARNING

I used traditional machine learning techniques to classify accident severity into three categories: "fatal", "serious", and "slight". I employed several classification algorithms, such as SGD Classifier, Logistic Regression, Naïve Bayes, Random Forest, Decision Tree and k-Nearest Neighbour.

ALGORITHMS USED AND THEIR HYPER PARAMETERS:

SGD Classifier:

HYPER PARAMETERS	**VALUES**	**BEST VALUE**
loss	“hinge”, “log”, “modified_huber”, “squared_hinge”, “perceptron”	log
penalty	“l1”, “l2”, “elasticnet”	l2
alpha	1e-4, 1e-3, 1e-2, 1e-1, 1e0, 1e1, 1e2, 1e3	0.01
learning_rate	“constant”, “optimal”, “invscaling”, “adaptive”	optimal

Logistic Regression:

HYPER PARAMETERS	**VALUES**	**BEST VALUE**
c	-4, 4, 20	11.2
Solver	‘newton-cg’, ‘lbfgs’, ‘liblinear’, ‘sag’, ‘saga’	sag
Max iter	20000	20000

Naïve Bayes:

HYPER PARAMETERS	**VALUES**	**BEST VALUE**
“var_smoothing”	-10, -1, 10	-10

Random Forest:

HYPER PARAMETERS	VALUES	BEST VALUE
criterion	“gini”, “entropy”, “log_loss”	gini
n_estimators	10, 50, 100, 200	200
max_features	“sqrt”, “log2”	log2

Decision Tree:

HYPER PARAMETERS	VALUES	BEST VALUE
criterion	“gini”, “entropy”, “log_loss”	entropy
max_features	“auto”,”sqrt”, “log2”	sqrt
splitter	“best”, “random”	random

KNN:

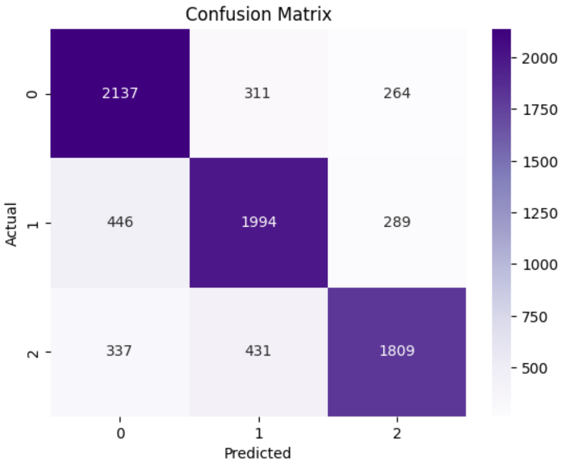
HYPER PARAMETERS	VALUES	BEST VALUE
n_neighbors	range(1, 31)	9
weights	“uniform”, “distance”	uniform
metric	“8uclidean”, “manhattan”, “minkowski″	manhattan

To optimize the model, I used a grid search with cross-validation to explore different hyperparameter combinations. Based on the accuracy and Precision of the optimized SGD Classifier, Logistic Regression, Random Forest Classifier, Decision Tree, and KNN models, I evaluated each model’s performance. With an appropriate level of complexity, Random Forest and KNN model showed higher performance in terms of accuracy and precision.

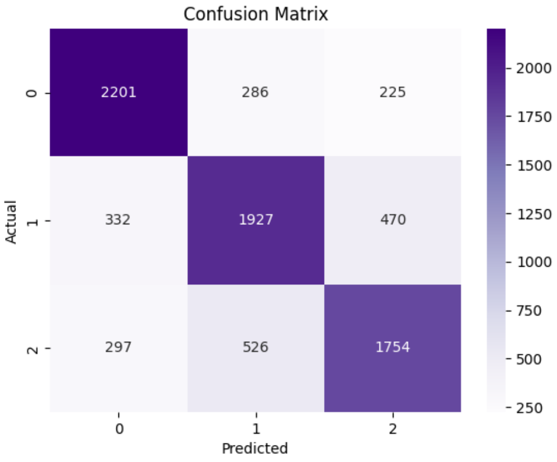
CONFUSION MATRIX:

By comparing predicted labels to actual labels, a confusion matrix is a table that summarises how well a classification system performed. It shows the proportion of accurate and inaccurate predictions for each class, making it easier to see the model’s accuracy, precision, recall, and other performance measures **(John D. Kelleher, 2015)**. Below, we can see the confusion matrix of the five-algorithm used;

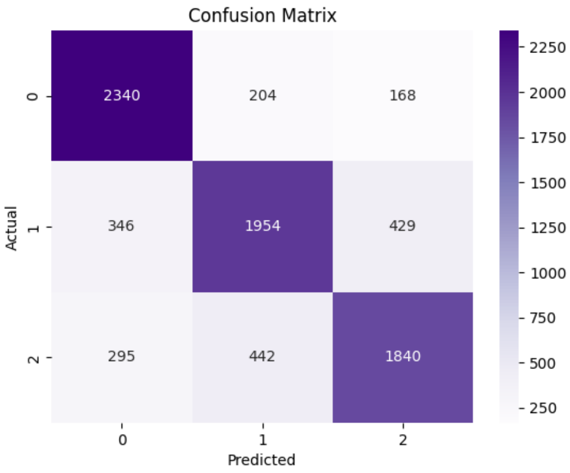
CONFUSION MATRIX OF THE TRADITIONAL CLASSIFICATION ALGORITHM



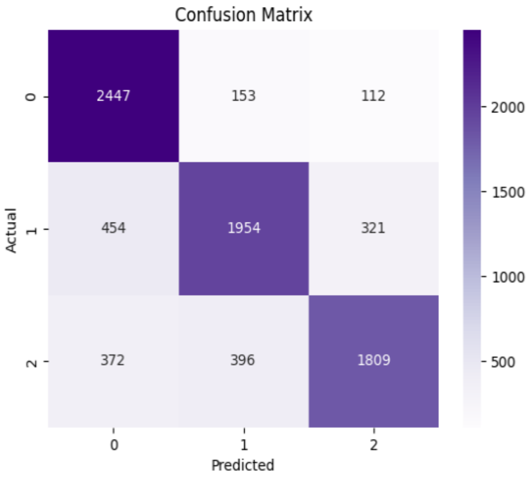
Logistic Regression



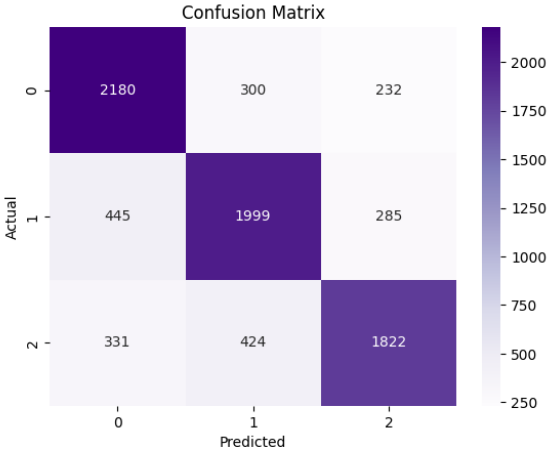
Decision Tree



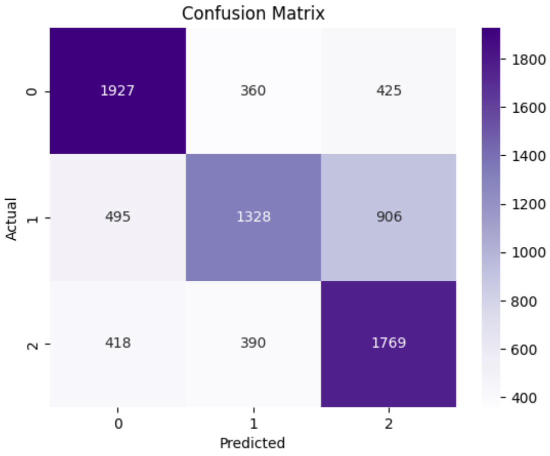
Random Forest.



K Nearest Neighbour



SGD Classifier



Naïve Bayes

PERFORMANCE METRICS

The performance of a classification model is evaluated using classification assessment measures. These metrics measure the difference between the predicted values and the true values and provide a way to evaluate the model's accuracy. There are several types of evaluation metrics and a few include; precision, recall, F1-score, accuracy, etc.

The evaluation metrics I used were Precision and accuracy:

Accuracy: Accuracy is defined as the ratio of correct predictions to total predictions. It is calculated using the following formula:

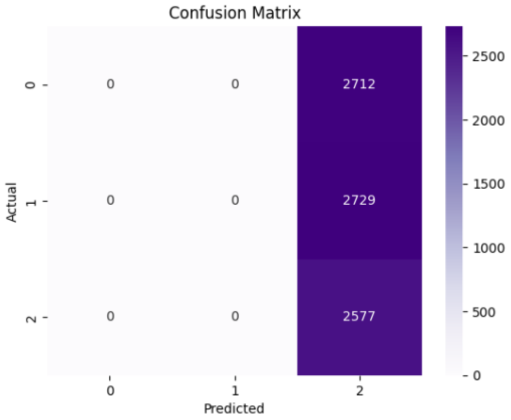
$$Classifier\ accuracy\ = \frac{TP + TN}{TP + TN + FP + FN} \times 100$$

Precision: The proportion of true positive predictions among all positive predictions

$$\boldsymbol{Precision\ =\ \frac{Tp}{(TP\ +\ FP))}}$$

COMPARISON WITH A TRIVIAL BASELINE:

To compare our model's performance with a trivial baseline, I used the DummyClassifier with the strategy set to "prior" from the sklearn library. Regardless of the input features, this classifier consistently predicts the dataset's most common accident severity class.



Confusion matrix of the dummy classifier

I was able to determine whether or not my model was learning significant patterns from the data by comparing its accuracy and Precision with the trivial baseline.

Comparing the classification algorithm:

Model	**Accuracy**	**Precision (Fatal)**	**Precision (Slight)**	**Precision (Serious)**
Logistic Regression	0.74	0.73	0.77	0.73
Decision Tree	0.73	0.78	0.72	0.70
Random Forest	0.77	0.78	0.76	0.75
KNN	0.77	0.75	0.81	0.78
SGD Classifier	0.74	0.74	0.78	0.73
Naïve Bayes	0.68	0.71	0.69	0.49
Dummy Classifier	0.32	0.00	0.32	0.00

After comparing the classification algorithms, here are my conclusions:

- With an accuracy of 0.77, the Random Forest and KNN classifiers outperformed the other models. This shows that these two classifiers perform better at predicting accident severity. Furthermore, they perform better than the other models in terms of precision ratings for each class of accident severity (Fatal, Slight, and Serious).
- Both the SGD Classifier and Logistic Regression perform similarly, obtaining an accuracy of 0.74.
- The Decision Tree classifier shad an accuracy of 0.73. While it outperformed the other models in the Fatal accident severity class, it fell short in the Slight and Serious classes.
- Naïve Bayes showed a significantly lower accuracy of 0.73 when compared to the other models.

As a result, in our dataset, the Random Forest and KNN classifiers performed the best in predicting accident severity.

4. CLASSIFICATION USING NEURAL NETWORKS

I classified accident severity into three groups using a neural network: "fatal," "serious," and "slight." I created a multi-layer perceptron (MLP) model, which is a feedforward artificial neural network with three layers: one input layer, one hidden layer, and an output layer.

I performed a grid search using several hyperparameters, including learning rate and the quantity of hidden layer nodes, to optimise the neural network model. In order to prevent overfitting and make sure that the model generalises effectively to unknown data, I used GridSearchCV to conduct a thorough search over the supplied parameter values.

Neural Network Hyperparameters;

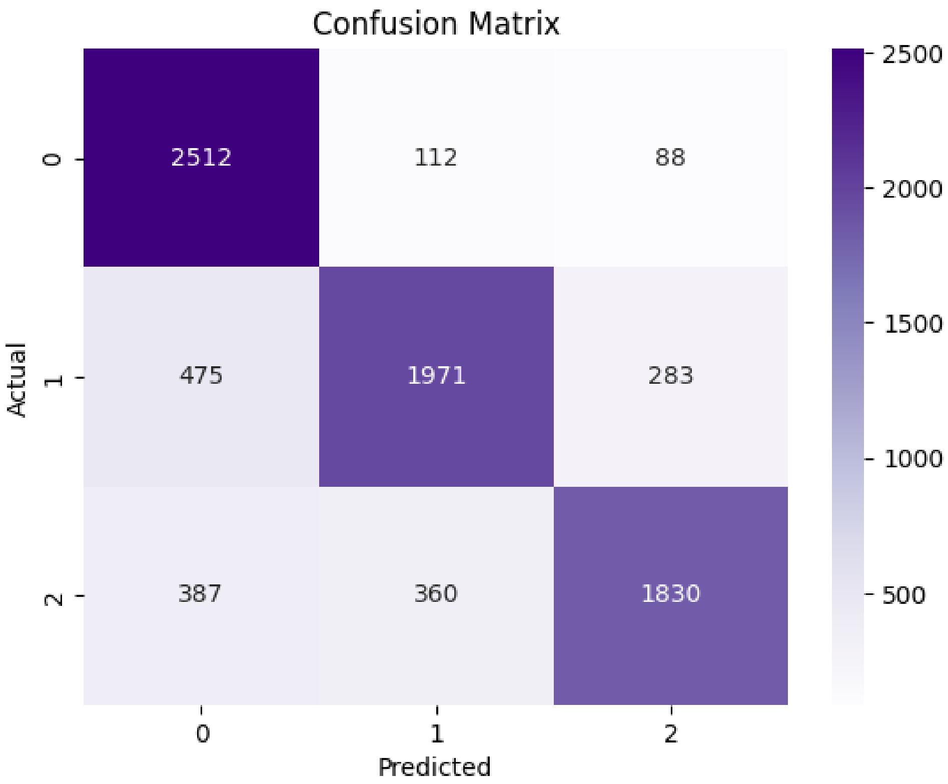
HYPER PARAMETERS	**BEST VALUE**
Number of hidden layers	1
Number of neurons in the hidden layer	20
Activation function	ReLU
Output layer activation	Softmax
Loss function	Sparse Categorical Cross-Entropy
Optimizer	Adam
Learning rate	0.001
Batch size	64
Number of epochs	50

To evaluate the performance of the chosen neural network model, I employed the following methods:

- Confusion matrix
- Performance metrics
- Comparison with trivial baseline
- Comparison with traditional machine learning classification model

CONFUSION MATRIX:

The figure below displays the proportions of true positives, true negatives, false positives, and false negatives for each class. It aids in the visualisation of the model's performance by pointing out instances that the model has classified properly or inaccurately.



Confusion matrix for neural network

PERFORMANCE METRICS:

Accuracy: Accuracy is defined as the ratio of correct predictions to total predictions. It is an appropriate metric when dealing with imbalanced datasets, as it considers both false positives and false negatives, providing a balanced view of the classifier's performance. This model achieved a balanced accuracy of 78%, indicating that the model performed well on the classification task.

Precision: Precision is calculated for each accident severity class (Fatal, Slight, and Serious) in this classification task, providing us an understanding of how well the model can distinguish between these classifications.

COMPARISON WITH A TRIVIAL BASELINE:

I evaluated the effectiveness of my neural network model in comparison to the DummyClassifier. With the help of this comparison, we can determine the importance of our model's performance and whether it has any real advantages over a naive method.

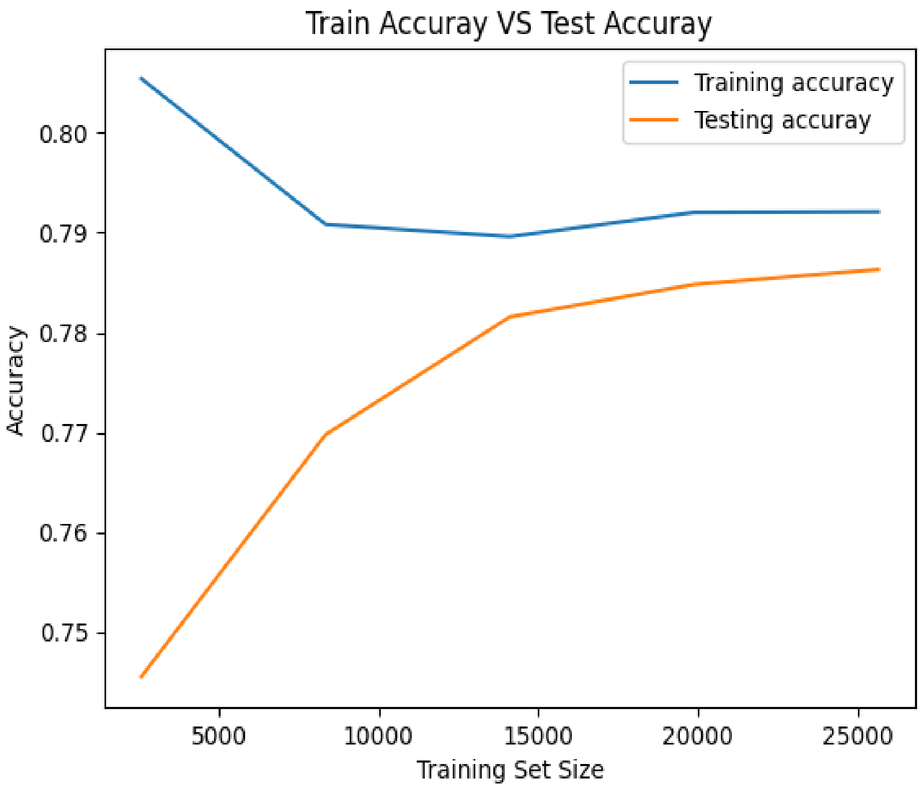
Model	**Accuracy**	**Precision (Fatal)**	**Precision (Slight)**	**Precision (Serious)**
Dummy Classifier	0.32	0.00	0.32	0.00
Neural Network	0.78	0.75	0.83	0.79

COMPARISON WITH TRADITIONAL MACHINE LEARNING CLASSIFICATION MODEL

Model	**Accuracy**	**Precision (Fatal)**	**Precision (Slight)**	**Precision (Serious)**
Random Forest	0.77	0.78	0.76	0.75
KNN	0.77	0.75	0.81	0.78
Neural Network	0.78	0.75	0.83	0.79

The comparison table reveals that among all classifiers, the Neural Network model achieved the highest performance metrics. It outperformed the other algorithms with an accuracy of 0.78.

I implemented a learning curve to further investigate the neural network model's performance. The training set size and model are correlated, as shown by the learning curve. This aids in identifying potential model problems, such as overfitting or underfitting.



Learning curve

Considering both the model's performance metrics and the learning curve, it reveals that the Neural Network model provides the most credible and accurate solution for determining accident severity.

5. ETHICAL DISCUSSION

Using Data Hazard Labels as a structure, I will examine some of the social and ethical issues of the accident severity prediction problem, taking into account the data collecting and processing stages, as well as the ML forecast and its possible influence on communities and individuals.

Label name	**Label description**	**Label image**	**Reason for applying**	**Relevant safety precautions**
Reinforces existing biases	Ensuring equal treatment of demographic groups and locations in the ML model		To prevent unfair predictions	Diverse data collection, bias-aware modelling techniques
Privacy and Security	Protecting sensitive information about individuals and locations		To maintain trust and respect privacy rights	Anonymizing personal data, adhering to data protection regulations
Difficult to understand	Providing clear and interpretable explanations for the ML predictions		To build trust and facilitate understanding of the system	Transparent documentation of data collection, preprocessing, and modeling techniques
Data Quality and Reliability	Ensuring accurate and unbiased data for effective ML predictions		To avoid misleading predictions and harmful consequences	Transparent and well-documented data collection process

Ethical and Legal Compliance	Adhering to ethical guidelines and legal regulations related to data collection, processing, and prediction		To minimize potential harm to communities and individuals	Ensuring compliance with relevant ethical guidelines and legal regulations

6. RECOMMENDATIONS

- Best Machine Learning Model: The Neural Network model is the best choice for the classification problem because it outperformed all other examined classifiers in terms of accuracy (0.78). In comparison to other algorithms, its performance implies that it has superior generalisation and adaptability in classifying accident severity.
- Model Suitability for Practice: The final Neural Network model performs better than the competition in terms of accuracy, precision, and learning curve, making it suitable for usage in practise. The model is a reliable choice for real-world use since it can effectively manage the complexity of the data and classify accident severity.
- Top Suggestion for Future Improvements: For future improvements, my top suggestion would be to explore more sophisticated neural network architectures, such as deep learning models or convolutional neural networks. These advanced models can capture more complex patterns in the data, potentially leading to even better performance and more accurate predictions of accident severity. Additionally, it would be beneficial to investigate more feature engineering techniques, as well as the impact of different data preprocessing methods on model performance.

7. RETROSPECTIVE

If I were to start the coursework again, an aspect I would investigate more in-depth would be ensemble learning methods. By combining various models' strengths and compensating for their weaknesses, it might be possible to achieve even better performance on the classification task. This exploration would provide valuable insights into the effectiveness of different ensemble techniques in the context of this specific problem.

8. REFERENCE

George Lawton, 2021. *TechTarget.* [Online]
Available at: https://www.techtarget.com/searchdatamanagement/definition/data-preprocessing
[Accessed 3 April 2023].

John D. Kelleher, B. M. N. A. D., 2015. *Fundamentals of Machine Learning for Predictive Data Analytics: Algorithms, Worked Examples, and Case Studies.* 2nd Edition ed. s.l.:MIT Press.