Predicting the Severity of Road Accidents in the UK

001277636

Word count: 2800

Executive summary

This project work has been employed on a dataset based on the UK road accident in 2019. For the prediction of the severity of road accidents in the UK. The dataset used in this project is a real-time dataset and is extremely imbalanced. The traffic-road accident is considered to be one of the main reasons for the unnatural death globally. The governmental organizations are working hard to increase the awareness about the laws which should be followed while driving to avoid the accidents. So, it is important to have a model which forecast the accident's severity and helps in minimizing the death rate (Avikumar Talaviya, 2023). In this project work, a machine and deep learning model is developed and evaluate on the testing dataset in order to predict severity of road traffic accident for following the important precaution by the enquiry agency. For employing such models, the raw data needs to be cleaned properly in order to get the accurate results. All the ethical and social issues regarding the project is also discussed and some recommendation or future scope of the project is also discussed.

1. Exploratory data analysis

EDA or exploratory data analysis is used to analyze and examine the dataset and concludes its important characteristics, which mainly implements methods of data visualization. It helps to regulate how to manipulate the data source for acquiring the needed answers, and providing an easy way to explore new patterns, check predictions, spot irregularities or evaluate a hypothesis.

- To have an overview to the dataset ".()head" function is employed. By default, this function displays the first five row to the dataset.

	accident_index	speed_limit	light_conditions	weather_conditions	road_surface_conditions	vehicle_type	junction_location	skidding_and_overturning	vehicle_leaving_carriageway	hit_object_off_carriagew
0	2019010225080	30	darkness	other	wet or damp	at least one van	at or within 20 metres of junction	no skidding or overturning	none leaving carriageway	none hit an obje
1	2019200908684	30	darkness	fine	dry	only cars	at or within 20 metres of junction	no skidding or overturning	at least one vehicle leaving carriageway	at least one vehicle hit obje
2	2019040860897	40	daylight	fine	dry	only cars	at or within 20 metres of junction	no skidding or overturning	none leaving carriageway	none hit an obje
3	2019460847205	40	daylight	fine	dry	only cars	not at or within 20 metres of junction	no skidding or overturning	none leaving carriageway	none hit an obje
4	2019051911581	30	daylight	fine	dry	only cars	not at or within 20 metres of junction	no skidding or overturning	none leaving carriageway	none hit an obje

Figure 1 overview of dataset

- The number of rows and columns in a dataset is determined by the '()shape" function. This particular dataset includes 14 columns and 31647 rows.
- The information of the entire dataset including column name, non-null value, data type can be determined by using "()info" function.

```
[6]: accident.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 31647 entries, 0 to 31646
Data columns (total 14 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   accident_index              31647 non-null  object
 1   speed_limit                 31647 non-null  int64
 2   light_conditions            31647 non-null  object
 3   weather_conditions          31647 non-null  object
 4   road_surface_conditions     31647 non-null  object
 5   vehicle_type                31647 non-null  object
 6   junction_location           31647 non-null  object
 7   skidding_and_overturning    31647 non-null  object
 8   vehicle_leaving_carriageway 31647 non-null  object
 9   hit_object_off_carriageway  31647 non-null  object
 10  first_point_of_impact       31647 non-null  object
 11  sex_of_driver               31647 non-null  object
 12  age_of_oldest_driver        25197 non-null  float64
 13  accident_severity           30475 non-null  object
dtypes: float64(1), int64(1), object(12)
memory usage: 3.4+ MB
```

Figure 2 information about dataset

The following dataset includes data in float, object and integer type.

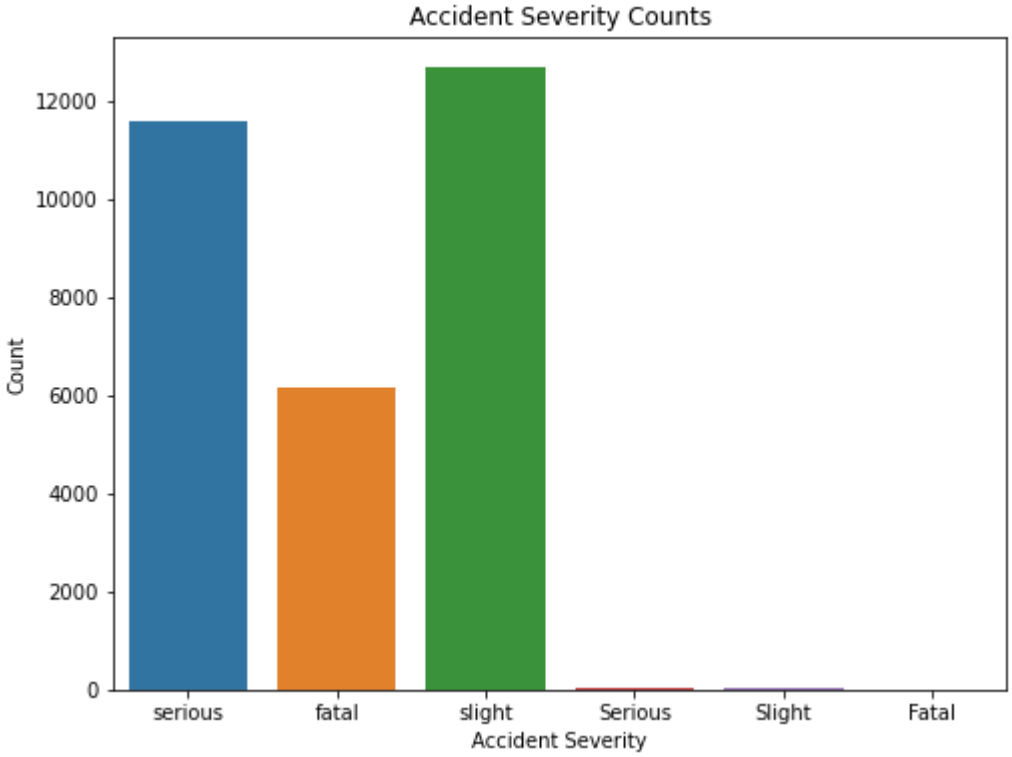

Figure 3 accident severity counts

The following plot is a bar graph which demonstrate the count of severity of accident. The x-axis displays the severity status of an accident and y-axis displays the count for it. The graph demonstrates that 'slight' shows maximum count which is more than 12000 meanwhile, the minimum count is displayed by 'fatal' which is approx. 6000. Although the severity labels is repeating again which give rise to a confusion and will get clean in the data cleaning and pre-processing phase.

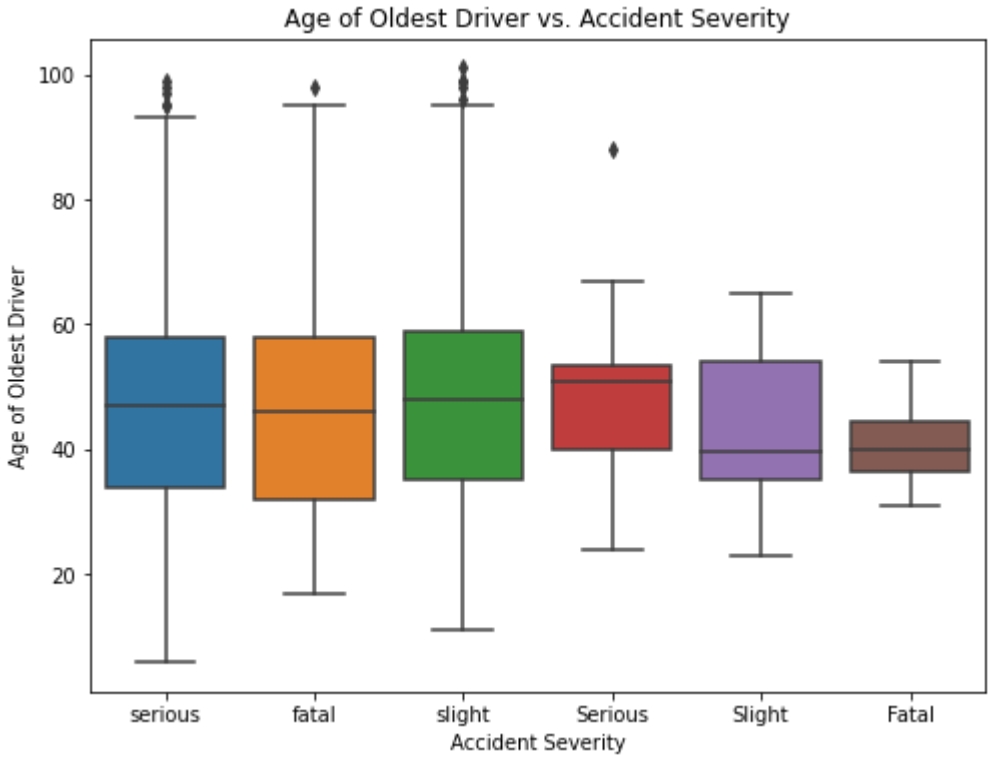

Figure 4 accident severity vs age of the oldest driver

The data present in this dataset is highly imbalanced and labels are continuously repeating which give rise to a lot of confusion while interpreting the data. This value will be replaced and cleaned properly.

The dataset has been cleaned by removing all the NaN values in the ‘accident severity’ column and values present in this column has been transferred in uppercase only. The following graph is plotted after the process of data cleaning which provides more accurate result.

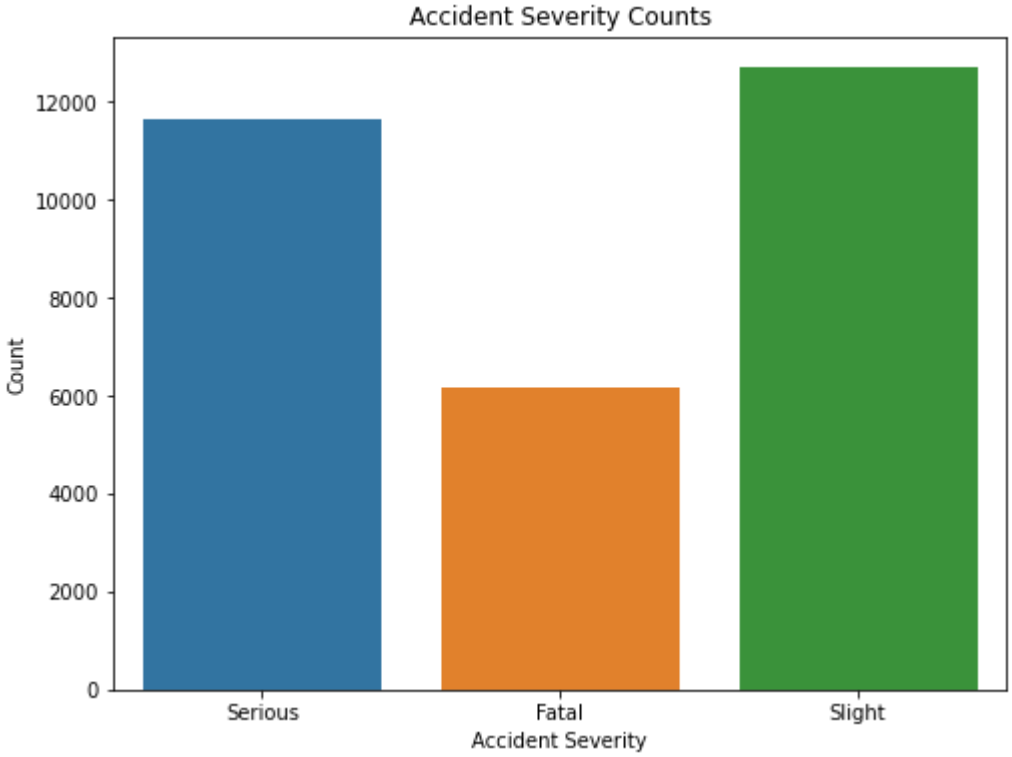

Figure 5 accident severity count

The following graph demonstrate the count of severity status of accidents. The outcome of the graph demonstrate that ‘slight’ accident has the maximum count which is 12691 where else, ‘fatal’ accident achieve least count which is 6167. The ‘serious’ accident has the count of 11617.

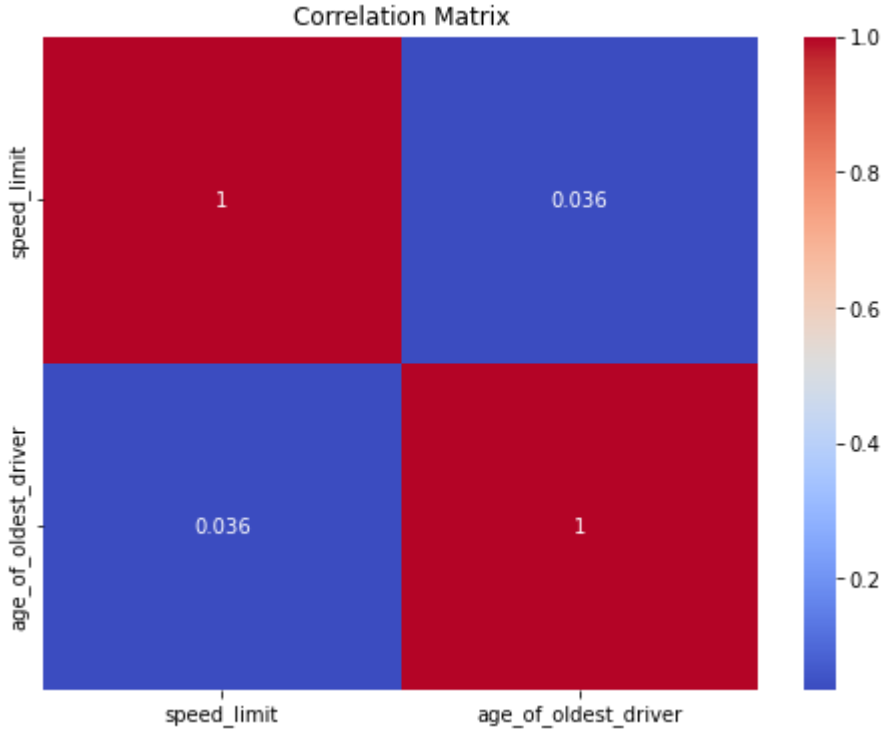

Figure 6 correlation matrix

The following plot is a correlation matrix which is used here to determine the relationship between age of the oldest driver with the speed limit. The plot shows the value as 1 which means that it is positively correlate with each other. In other words, if the value of a variable is increase or decrease, the other variable also shows the same behavior.

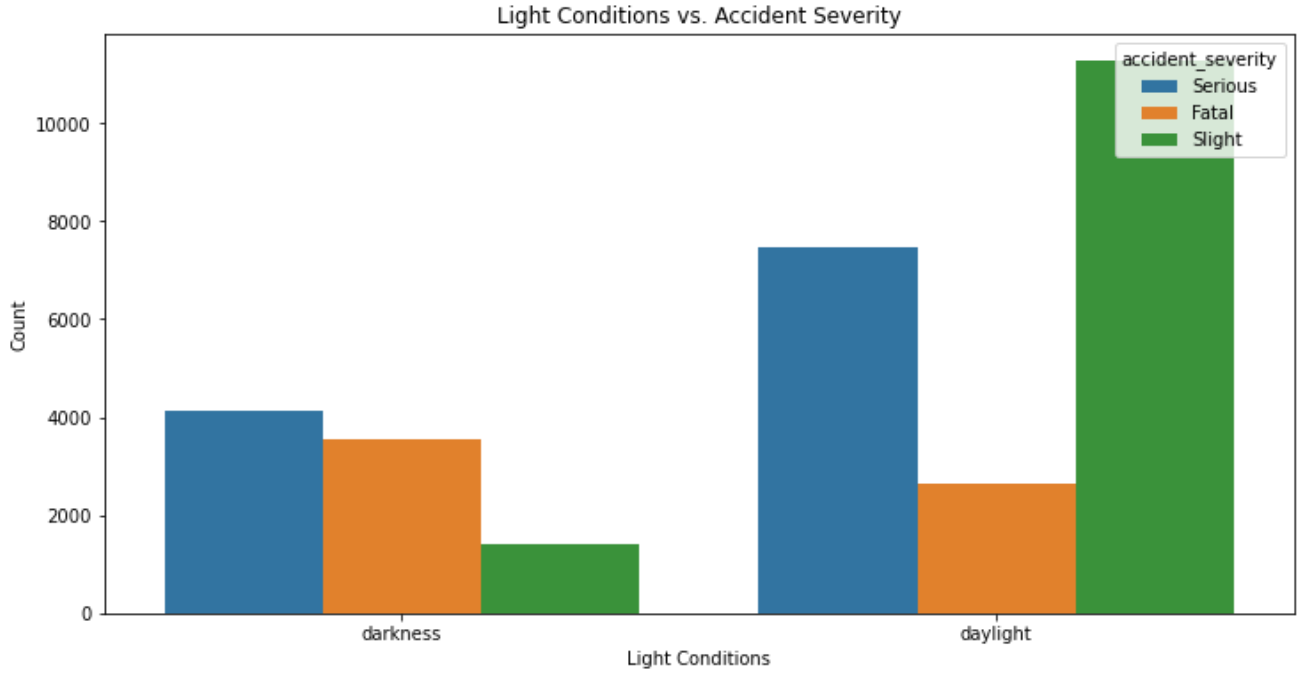

Figure 7 light condition vs accident severity

The following plot is stacked bar graph which demonstrate the count of severity status of accident according to the light condition. The graph demonstrates that in the dark condition ‘serious’ accident occurs the most and in the day time, ‘slight’ accident occurs the most.

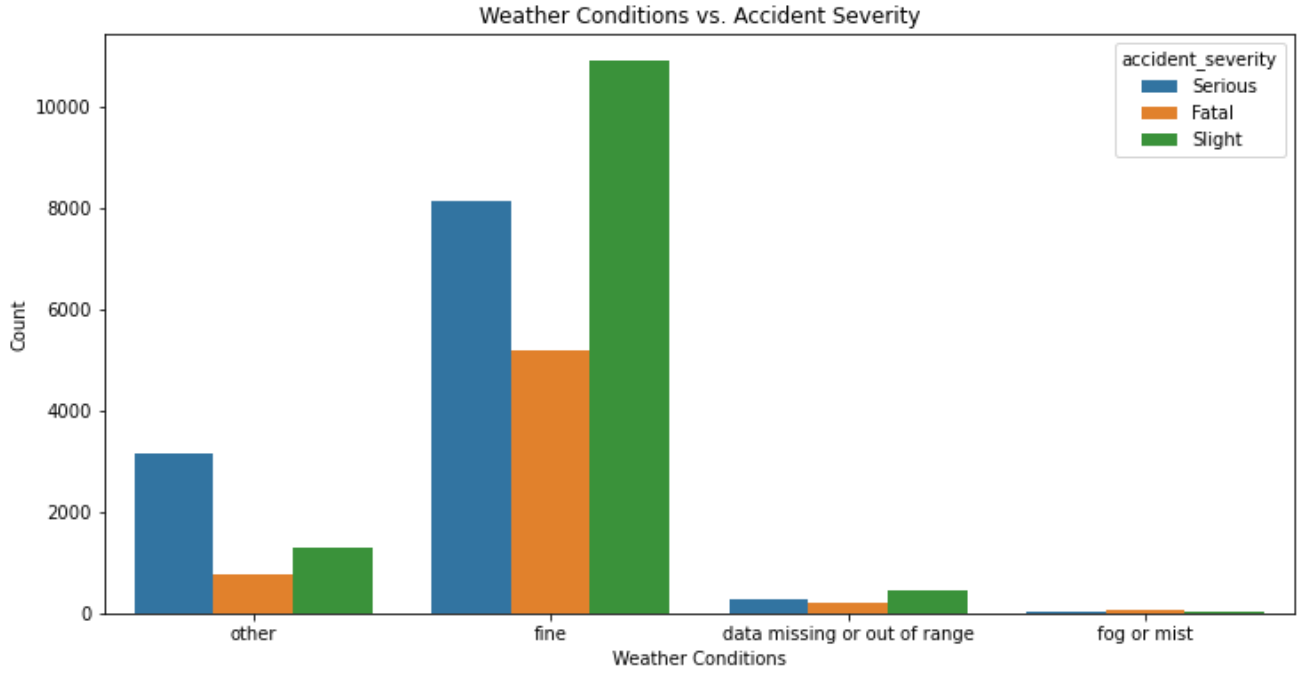

Figure 8 weather conditions vs accident severity

The following stacked bar plot demonstrate the count of accident according to their occurs under

different weather condition. The plot demonstrates that under fine condition highest number of ‘slight’ accidents occurs. In other condition ‘serious’ accident occurs the most. In the condition of mist or fog very a smaller number of accidents occurs. Few accidents also occur in missing or out of range weather condition.

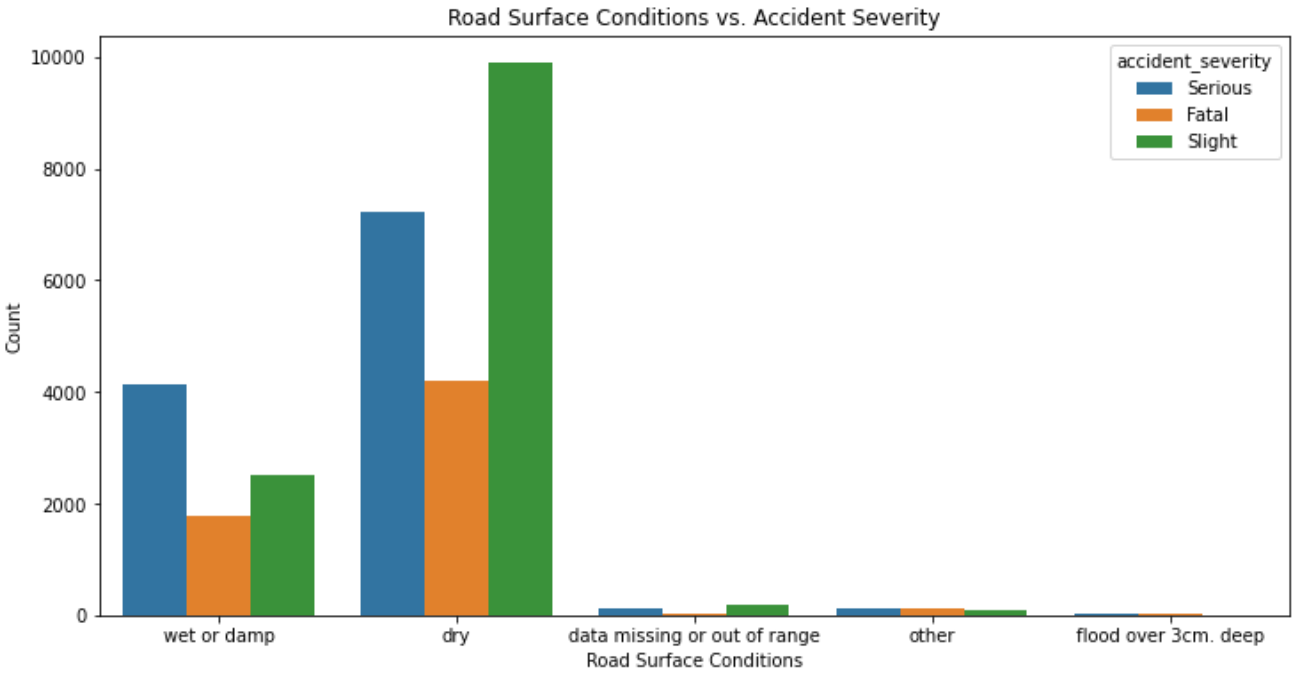

Figure 9 road surface condition vs accident severity

The following plot is a stacked bar graph which represents the count of severity of road accident occurs under different road surface conditions. The graph demonstrates that under dry road condition highest number of ‘slight’ accidents occurs. Under the damp or wet condition mostly ‘serious’ accident takes place. Under flood and other condition very few number of accident takes place. Some accident also occurs in the unknown or missing condition.

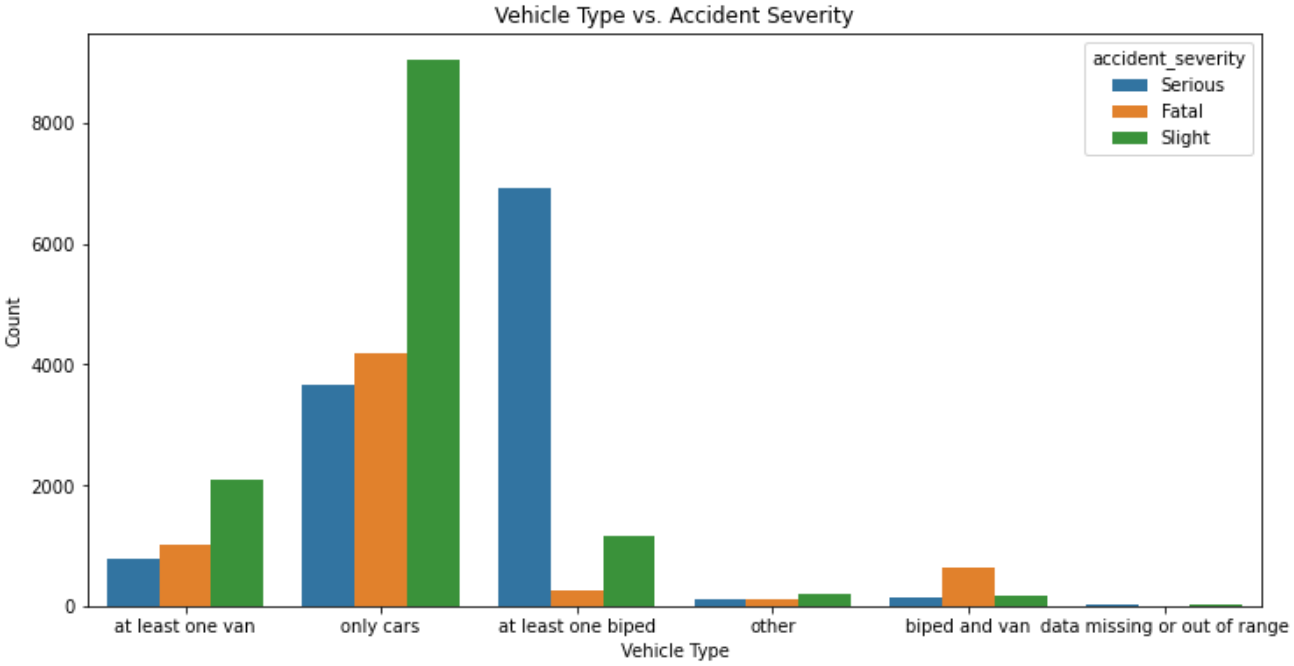

Figure 10 vehicle type vs accident severity

The following stacked bar graph demonstrated the count of accident occurred by which type of vehicle. The output of the graph demonstrates that while driving car, ‘slight’ accidents occur the most. While driving van, ‘slight’ accident occurs the most. When at least one biped vehicle is presented then ‘serious’ accident occurs the most. With biped and van, ‘fatal’ accident occurs the most. While some of the data is missing or out of range.

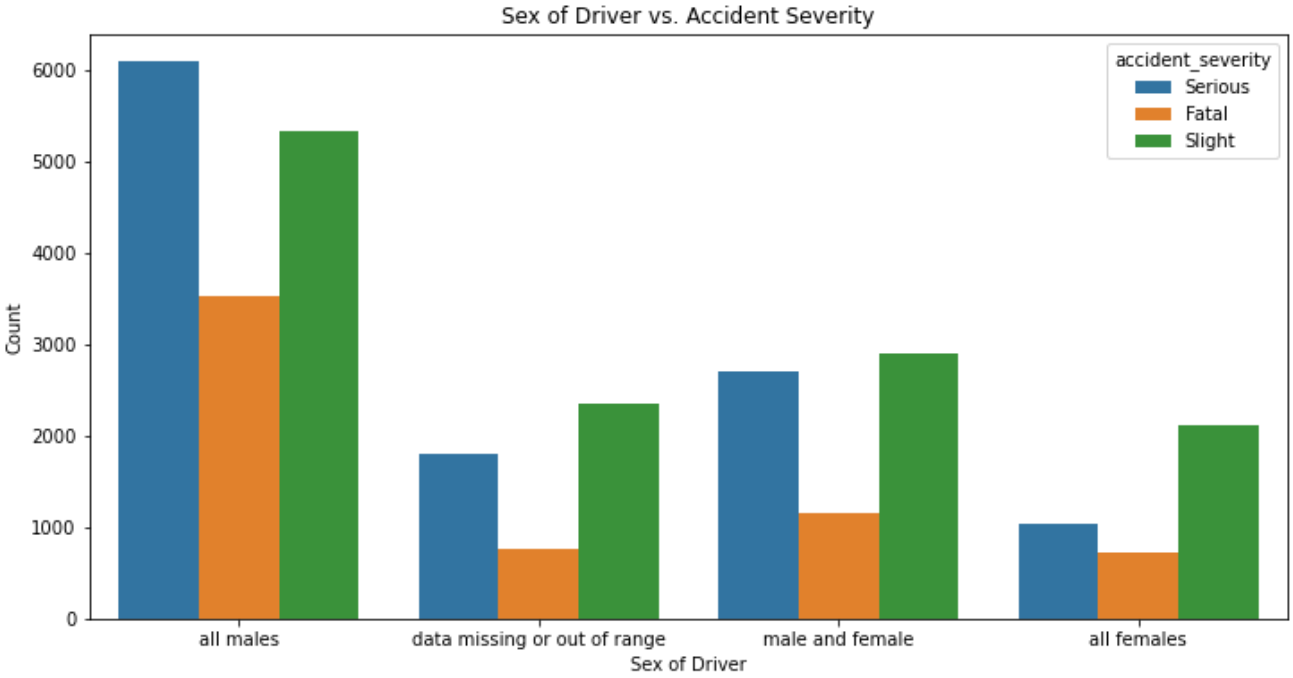

Figure 11 sex of driver vs accident severity

The following stacked bar graph illustrates the count of accident severity based on the sex of driver. when males are driving the vehicle 'serious' accident occurs the most. When only females are driving 'sight' accident occurs the most and when both male and female are driving then 'slight' accident occur the most. The rest of the data is missing or is not clearly provided by the dataset.

```
In [16]: accident.isna().sum()

Out[16]: accident_index                 0
         speed_limit                    0
         light_conditions               0
         weather_conditions             0
         road_surface_conditions        0
         vehicle_type                   0
         junction_location              0
         skidding_and_overturning       0
         vehicle_leaving_carriageway    0
         hit_object_off_carriageway     0
         first_point_of_impact          0
         sex_of_driver                  0
         age_of_oldest_driver        6188
         accident_severity              0
         dtype: int64
```

Figure 12 checking null values

Here the null values for the columns is checked through '.isna() . sum()' function. These missing values

can be filled with an appropriate value which includes mode, mean or median of the given ages.

```
In [18]: accident.isna().sum()

Out[18]: accident_index                0
         speed_limit                   0
         light_conditions              0
         weather_conditions            0
         road_surface_conditions       0
         vehicle_type                  0
         junction_location             0
         skidding_and_overturning      0
         vehicle_leaving_carriageway   0
         hit_object_off_carriageway    0
         first_point_of_impact         0
         sex_of_driver                 0
         age_of_oldest_driver          0
         accident_severity             0
         dtype: int64
```

Figure 13 NaN values

Fillna() function is used here to replace the null values with an suitable value.

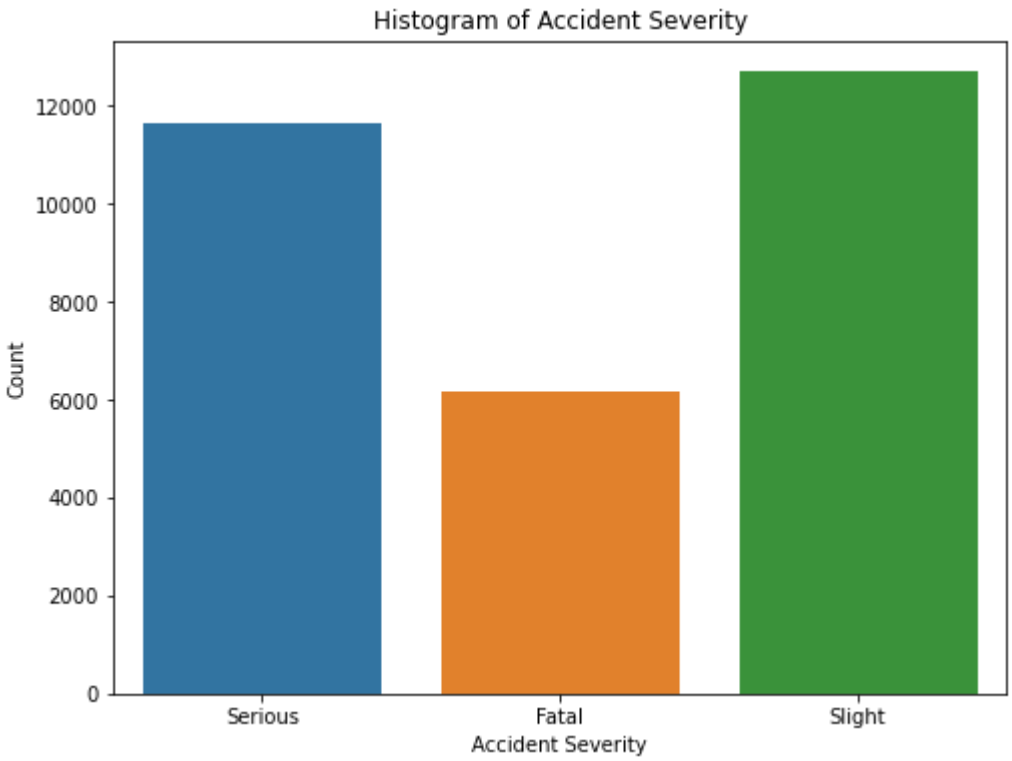

Figure 14 accident severity

The following graph demonstrate the count of severity of accident. The outcome of the graph

demonstrates that ‘slight’ accident occurs the most which has the count of more than 12000. Meanwhile, ‘fatal’ accident occurs the most having count of approx. 6000.

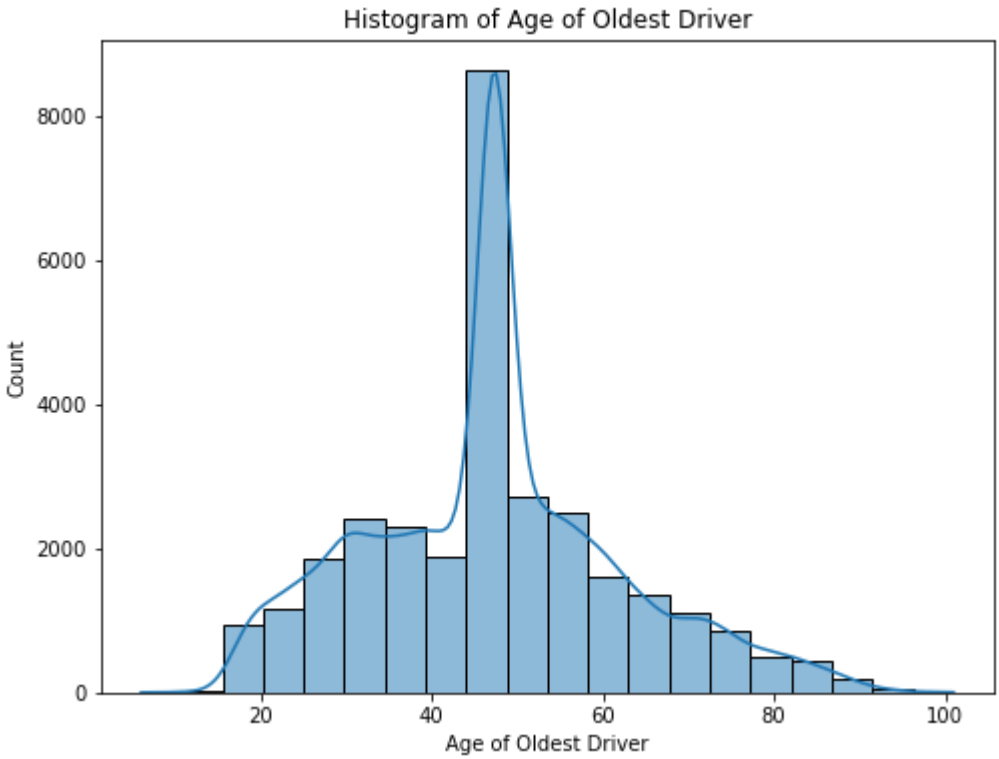

Figure 15 histogram of age of oldest driver

The following plot is a histogram which demonstrates the count of age of oldest driver. the x-axis of the graph represents the age of drivers and the y-axis has the count of drivers. The graph demonstrates that the maximum age of driver as 50-55 which has the count of more than 8000.

The ‘accident_index’ columns is removed by .drop() function and then value of each column is count by .value_count() function.

2. Data preprocessing

The process of data preprocessing includes transformation of raw data in an understandable way. It is important step as machine and deep learning algorithms is not applied on the raw and uncleaned data and data quality should also be determined before applying deep and machine learning algorithms. This process of majorly used to check the quality of data. In this dataset, all the column values gets converted into the lowercase strings through .str.lower() function. the label ‘data missing or out of range’ gets

combined with ‘other’ in the columns having ‘data missing or out of range’ label. Also, in ‘sex of driver’ column, the label ‘male and female’ combined with ‘other’. The value for each column is counted after processing. Also, all the rows having null values is removed by .dropna() function. and the value of categorical variables is replaced by using label encoders.

3. Classification using traditional machine learning

Decision tree

Decision tree is a supervised machine learning algorithm which is used for to solve the programs regarding regression and classification. This algorithm has a tree like- hierarchical structure which includes root node, internal nodes, branches and leaf nodes. The main aim of this algorithm to generate a model which forecasts value of a target variable by employing some easy decision rules concluded from the features of data. A decision tree consists of two nodes which includes leaf node and decision node. The decision node is used to generate a decision and have many branches, were the leaf nodes illustrate the result or output of the decisions and don’t comprise of any additional branches. A decision tree can be considered as a visual presentation for acquiring all the possible outcomes for a decision or a condition based on the given conditions. For generating a tree, an algorithm is used called as CART which stands for classification and regression tree algorithm. It basically, asks a question and generates the answer in binary form (yes/no) and it additionally divides the tree into sub-trees.

- It usually impersonators the thinking of human while generating any decision which is easy to comprehend.
- Th main reason for using a decision tree is that it can be easily understood as it shows a tree- like structure

Working of decision tree algorithm-

In the decision tree algorithm, for assuming a class for the given dataset, the method starts from the tree’s root node. This method likens the root features value with the attribute of real dataset and based on the evaluation, it follows the branch and moves forward to the further node. For the further node, this algorithm again evaluates the feature value with another sub-node and move forward. The process will be continued till it reaches to the leaf node of the tree.

the data gets divide into features and the target variable. The target variable used here is ‘accident

severity' as most of the features can be determined on this variable. And then it gets divided into the two sets which is training and testing. The decision tree algorithm is trained on the training dataset and then gets predicted on the testing dataset. A classification report is generated based on the prediction made on testing set and the results get evaluate based on the below-mentioned metrics.
For evaluating the performance of the model few of the metrics is used which includes a classification report which further consist of:

- Accuracy:

It has the ability which demonstrates the performance of the model across all the classes. It is the fraction of correct number of predictions with the number of total predictions.

- Precision: TP/(TP+FP)

It has capability of a model not to label a sample as positive which is negative in actual. For a class, it can be demonstrated as the fraction of true positives with the total of false positive and true positive.

- Recall: TP/ (TP/FN)

It has the capability of a classifier to explore all the positive samples. For a class, it can be defined as the fraction of true positive to the total of true positive and false negative values.

- F1 score: 2 (Recall * Precision) / (Recall + Precision)

It is said to the harmonic mean value of the recall and precision in such a way that the best score ranges 1.0 and the worst one is 0.0. it is said to low than the accuracy as it implants the value of recall and precision into its calculation.

- Support:

It is said to the frequency of the actual occurrences of a class in a particular dataset. The imbalanced score of support shows the weakness of the structure in the training phase and also shows the requirement of rebalancing. (Ahmed Fawzy Gad, 2020)

	precision	recall	f1-score	support
0	0.60	0.65	0.62	1222
1	0.68	0.70	0.69	2308
2	0.76	0.71	0.73	2565
accuracy			0.69	6095
macro avg	0.68	0.69	0.68	6095
weighted avg	0.70	0.69	0.69	6095

Figure 16 classification report

The classification report conveys the outcome of the classifier that it attains 0.69 accuracy.

Tuned Decision tree

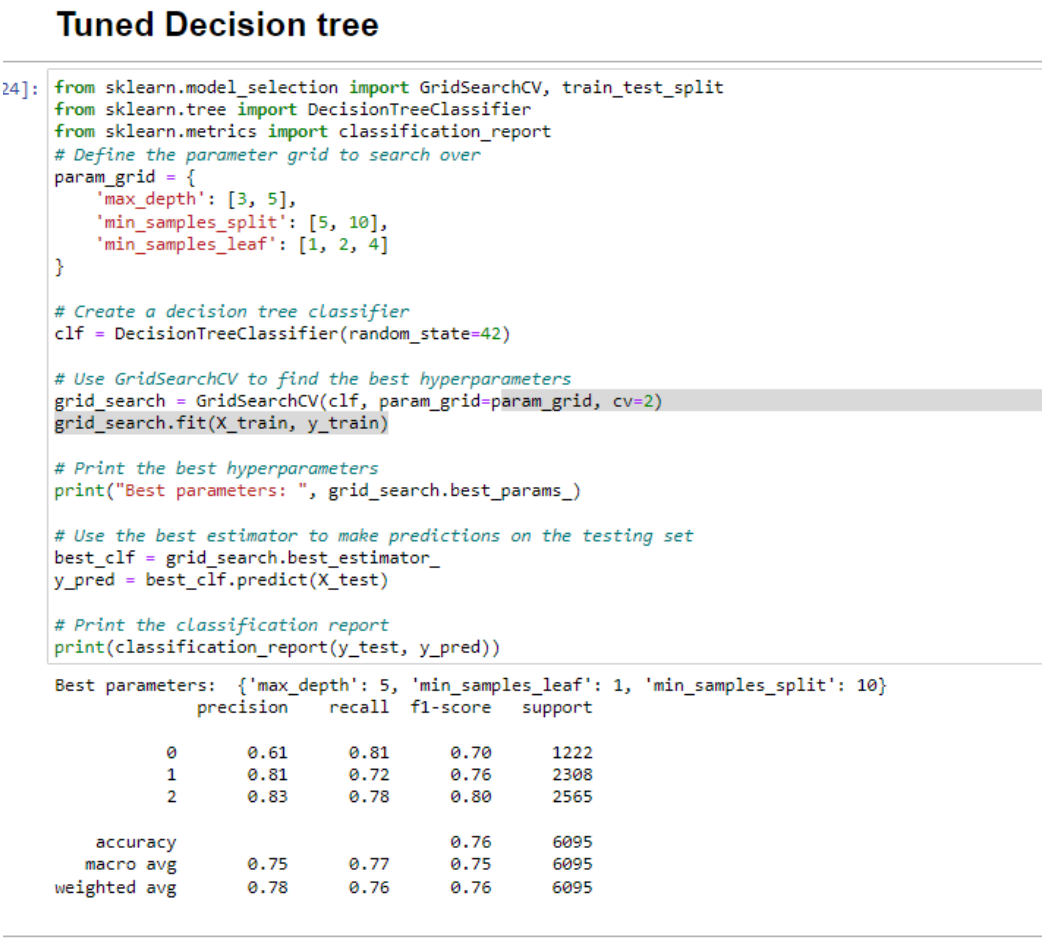

Figure 17:Tuned Decision Tree

According to the outcomes , GridSearchCV's top hyperparameters were:

min samples leaf: 1 max depth: 5 min samples split: 10
To manage the decision tree's complexity, certain hyperparameters are specially used:

max depth: The decision tree's maximum depth. While a lower max depth can result in underfitting, a larger max depth can cause overfitting.
min samples leaf is the bare minimum of samples that must be present at a leaf node. By requiring more samples at each leaf node, a larger min samples leaf can stop overfitting.
The minimum number of samples needed to separate an internal node is known as min samples split. By requiring more samples to split a node, a greater min samples split can also prevent overfitting.
As can be seen from the classification report, the adjusted decision tree classifier has an accuracy of 0.76.
The findings show that the tuned decision tree classifier has fairly high precision, recall, and f1-score for each of the three classes. This shows that the classifier can distinguish between incidents of varying degrees of severity with some degree of accuracy. The classifier is doing well overall, as evidenced by the fact that both the macro and weighted averages are both rather high.

4. Classification using neural Network

Neural networks are said to be the machine learning algorithms and is considered to the main part of algorithm. Neural network is also called as SNN (simulated neural networks) or ANN (artificial neural network). The structure and name of this algorithm is stimulated by a human brain and it impersonators the working of a human neural system. The neural networks are included of different node layers, which comprises of an input layer, an output layer and one or more than one of hidden layers. Each of the node links to another node which has a related threshold and weight. If the result of any node is more than a quantified value of threshold, then the node gets activated and transferring the data to further network layer. Else, no data is transferred to the next network layer. These networks are based on the training data to analyze and enhance their accuracy across time. but, once they achieved a satisfactory accuracy, they become one of the most powerful algorithms and used to classify data at a good velocity. (Hardesty, 2017)

Working of neural network:
Neural network is made up to thousands of neurons or nodes which contain various layers such as an input layer, an output layer and a hidden layer. The real-world data enters in this network via a input layer. The layer processes, evaluate and classify the data and move it forward to the another layer which is hidden layer. The neural network includes many hidden layers. The main focus of this hidden layer is to compare the result provided by the prior layer, process that result and then pass it forward to the nest hidden layer or to the output layer. The output layer obtains the final finding the entire procedure by the neural network. It may include one or more than one node.

One-hot encoded is used on the target variable which converts the categorical variable into the encoder as machine learning or deep learning networks aren't capable to work with the categorical data directly. The model requires to explore input's shape for that it the first layer is in 'sequential' model. Before the process of training a mode, it needs to compile first, for which three different argument is used such as optimizer, loss and metrics. And then the model is trained having epochs=50 as it helps in generating an accurate result without consuming much time.

```
In [29]: model.summary()
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 64)                832

 dense_1 (Dense)             (None, 32)                2080

 dense_2 (Dense)             (None, 3)                 99

=================================================================
Total params: 3,011
Trainable params: 3,011
Non-trainable params: 0
```

Figure 18 model summary

The entire summary of a model is provided which demonstrates about the type of layer, shape of the output and parameters it includes.

Epochs

Epochs is a way of training network with the trained data used for a single cycle. In this process, the data is only used for a single time. When a forward and a backward is together then it is called as a pass. Epochs are generated from a single batch or more than one batch, in which only few parts of the dataset is used in the training of neural network. (SHARMA, 2017)

the trained model is evaluated and generates a confusion matrix which is used for measuring the metrics of a classification report such as accuracy, recall, precision, and specificity.

```
[[1048  126   48]
 [ 362 1697  249]
 [ 287  327 1951]]
```

Figure 19 confusion matrix

The result of confusion matrix is evaluated by few parameters which includes true positive value, true negative value, false positive value and false negative value.

	precision	recall	f1-score	support
0	0.62	0.86	0.72	1222
1	0.79	0.74	0.76	2308
2	0.87	0.76	0.81	2565
accuracy			0.77	6095
macro avg	0.76	0.78	0.76	6095
weighted avg	0.79	0.77	0.77	6095

Figure 20 classification report

A classification report is generated based on the result of confusion matrix, which demonstrates that the model trained on neural networks attains accuracy of 0.77 and outperformed the decision tree classifier.

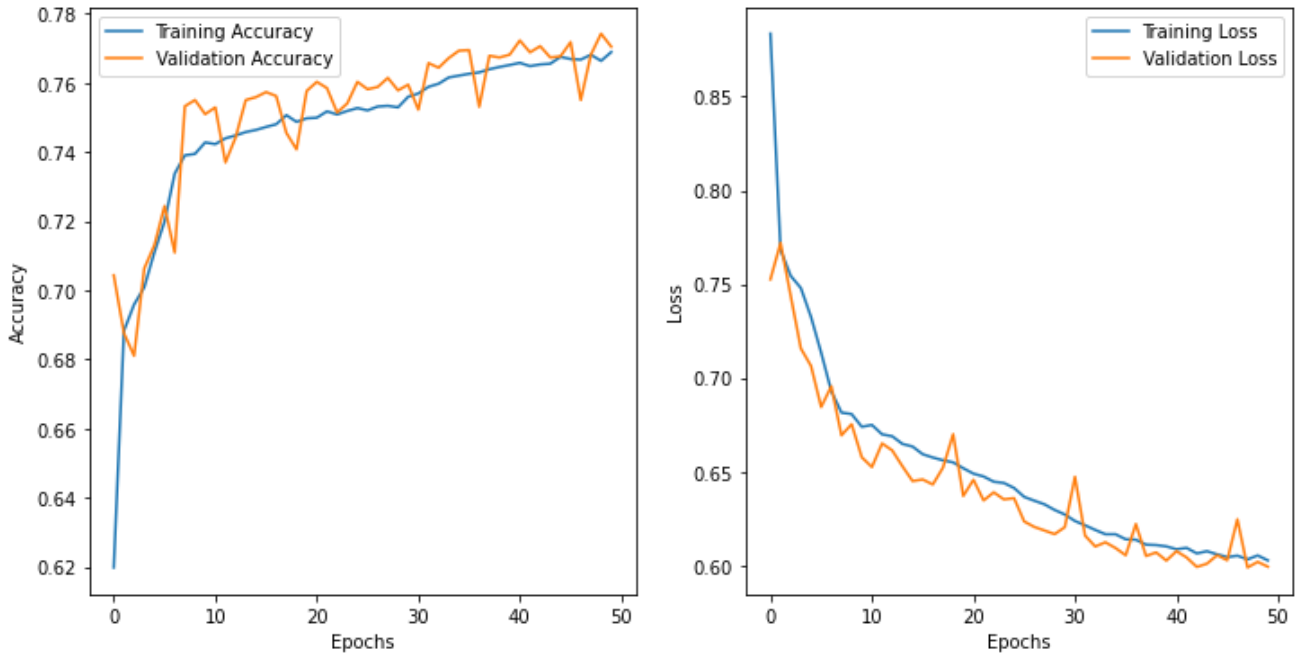

Figure 21 epoch vs accuracy/loss

The following graph is used to determine whether the model leads to overfitting or is still in the learning phase.

In the graph generated between the epoch and accuracy demonstrates little accuracy as there is not of gap between the training and validation accuracy trend, meanwhile in the case of epoch vs loss, the graph demonstrates that the model shows a improve rate of performance.

Trained Neural Net

Tuned Neural Net

```
]: callbacks import EarlyStopping
   mport classification_report

   meters to tune over

   m', 'sgd'],
   ]


   y_test to one-hot encoded format
   eras.utils.to_categorical(y_train, num_classes=3)
   ras.utils.to_categorical(y_test, num_classes=3)



   ations of hyperparameters
   _grid['optimizer']:
    param_grid['hidden_size']:

   eural network model
   as.Sequential([
   ayers.Dense(64, activation='relu', input_shape=(X_train.shape[1],)),
   ayers.Dense(hidden_size, activation='relu'),
   ayers.Dense(3, activation='softmax')


   model with the current hyperparameters
   ptimizer=optimizer, loss='categorical_crossentropy', metrics=['accuracy'])

   del with early stopping
   arlyStopping(patience=5, restore_best_weights=True)
   l.fit(X_train, y_train_encoded, validation_data=(X_test, y_test_encoded), epochs=100, batch_size=32, callbacks=[early_stop], verb

    model and store the best score and hyperparameters
   evaluate(X_test, y_test_encoded, verbose=0)[1]
   t_score:
    = score
   s = {'optimizer': optimizer, 'hidden_size': hidden_size}
   oded = model.predict(X_test)
   ncoder.inverse_transform(y_pred_encoded)
```

Figure 22:Tuned Neural Net

```
evaluate(X_test, y_test_encoded, verbose=0)[1]
t_score:
 = score
s = {'optimizer': optimizer, 'hidden_size': hidden_size}
oded = model.predict(X_test)
ncoder.inverse_transform(y_pred_encoded)
```

```
191/191 [==============================] - 0s 720us/step
```

In [30]:
```
# Print the best hyperparameters and classification report
print("Best parameters: ", best_params)
print(classification_report(y_test, y_pred))
```

```
Best parameters:  {'optimizer': 'adam', 'hidden_size': 16}
              precision    recall  f1-score   support

           0       0.62      0.88      0.72      1222
           1       0.80      0.72      0.76      2308
           2       0.86      0.77      0.81      2565

    accuracy                           0.77      6095
   macro avg       0.76      0.79      0.76      6095
weighted avg       0.79      0.77      0.77      6095
```

Figure 23: Results of Neural Net

GridSearchCV's optimal hyperparameters for the tuned neural network were "optimizer": "adam" and "hidden size": 16. The hidden size parameter controls the amount of neurons in the hidden layer of the neural network, while the optimizer parameter specifies the optimisation algorithm used to change the weights of the neural network during training. The grid search discovered that the highest performance was achieved using the Adam optimizer and a smaller hidden layer size of 16.

We can see that both models have comparable precision, recall, and f1-score for all three classes by comparing the classification report of the main model with the tuned model. But the accuracy, macro-average f1-score, and weighted-average f1-score of the tuned model are marginally higher.

Results

Model	Hyperparameters	Accuracy	Macro Avg F1-Score	Weighted Avg F1-Score
Default Decision Tree	N/A	0.70	0.68	0.70
Tuned Decision Tree	{'max_depth': 5, 'min_samples_leaf': 1, 'min_samples_split': 10}	0.76	0.75	0.76
Basic Neural Net	{'optimizer': 'adam', 'hidden_size': 32}	0.77	0.76	0.77
Tuned Neural Net	{'optimizer': 'adam', 'hidden_size': 16}	0.77	0.76	0.77

The table demonstrates that all models have comparable accuracy, with the basic and adjusted neural networks having the best accuracy (0.77). The tuned decision tree has the highest macro-average and weighted-average f1-scores of 0.75 and 0.77, respectively, while the basic neural network has the highest macro-average and weighted-average f1-score of 0.75 among all models.

The maximum depth of the tuned decision tree was five, the minimum sample size per leaf was one, and the minimum sample size per split was ten. The 'adam' optimizer and a hidden layer size of 32 were utilised by the basic neural network, whereas a hidden layer size of 16 was employed by the tuned neural network.

5. Ethical discussion

the following research work conducted was interesting as well as challenging. The exploratory data analysis provides a great insight the data regarding the topic of “prediction of road accident cases in UK” in which most of the labels are explained properly and in the process of data preprocessing and

cleaning, the data is cleaned properly in order to obtain the best result possible.

6. Recommendations

- The machine learning model for the prediction task provides average accuracy instead of decision tree classifier, random tree can be employed for the improved results.
- The final model depicts not so good result, in order to improve the result more epochs should be employed.
- There is still some room of scope by the model which can be achieved by in-depth analysis and employing more accurate machine learning models.

7. Retrospective

In this report a machine learning based and a deep learning-based algorithm for the prediction of severity of road accident in UK and the proposed model achieved average rate of accuracy.
if I've to start this work all over again then I would try to learn about some more machine learning algorithm and try to implement clearer and more precise dataset.

8. References

Ahmed Fawzy Gad (2020). Accuracy, Precision, and Recall in Deep Learning | Paperspace Blog.[online] Paperspace Blog. Available at: https://blog.paperspace.com/deep-learning-metrics precision-recall-accuracy/.

Hardesty, L. (2017). Explained: Neural networks. [online] MIT News | Massachusetts Institute ofTechnology. Available at: https://news.mit.edu/2017/explained-neural-networks-deep-learning-0414 .

Jose, G.V. (2019). Useful Plots to Diagnose your Neural Network - Towards Data Science. [online]Medium. Available at: https://towardsdatascience.com/useful-plots-to-diagnose-your-neural-network-521907fa2f45.

SHARMA, S. (2017). *Epoch vs Batch Size vs Iterations - Towards Data Science*. [online] Medium. Available at: https://towardsdatascience.com/epoch-vs-iterations-vs-batch-size-4dfb9c7ce9c9 [Accessed 7 Apr. 2023].