

Comp1804 Report: Predicting the severity of road accidents in the UK.

Student ID: 001249887

DATE: 11/04/2023

Word count: 3050

## **Executive summary**

The report contains the process and results of prediction of road accidents severity in the UK. The process includes exploratory data analysis and preprocessing of the dataset. This prediction is important to identify if neural networks outperform traditional machine learning algorithms. The traditional machine learning models implemented are random forest classifier, decision tree classifier, linear svm classifier and stochastic gradient classifier. After implementing all models, it is identified that random forest classifier gave the highest accuracy, which means neural networks did not outperform conventional models.

## 1. Exploratory data analysis(EDA)

EDA involves analyzing datasets and summarizing their key features which helps identify trends, patterns, outliers, errors, unusual occurrences like invalid and missing data. (Barla, 2023)

Identifications of such errors and occurrences, is important to know what errors to correct during preprocessing of data. The dataset contains 14 columns and 31647 rows. Figure 1 shows each column's datatypes, where only two columns have numeric datatype, and the rest are categorical.

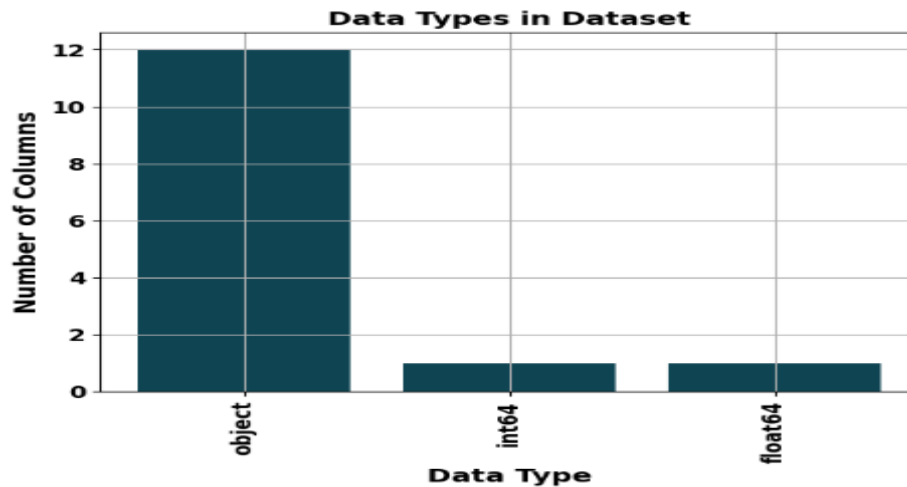


Figure 1: Data types.

The statistical summary of the numerical columns shows that the minimum speed in "speed\_limit" is -1, which is an invalid value because speed can never be less than 0. Figure 2 shows the incorrect value.

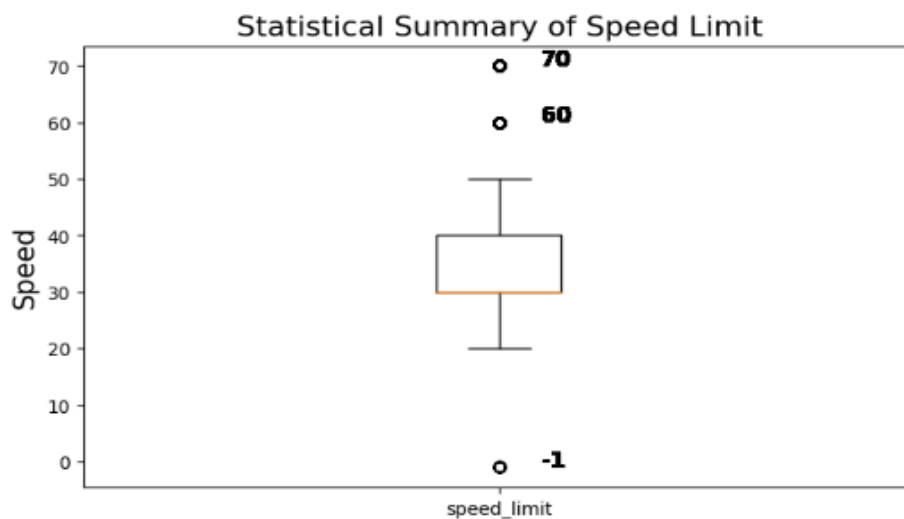


Figure 2: Invalid value.

To find missing values, I used “isna()” function, which returns all NaN values that the dataset contains. It is found that "age\_of\_oldest\_driver" has 6450 NaN values and “accident\_severity” has 1172 NaN values. This is shown in figure 3.

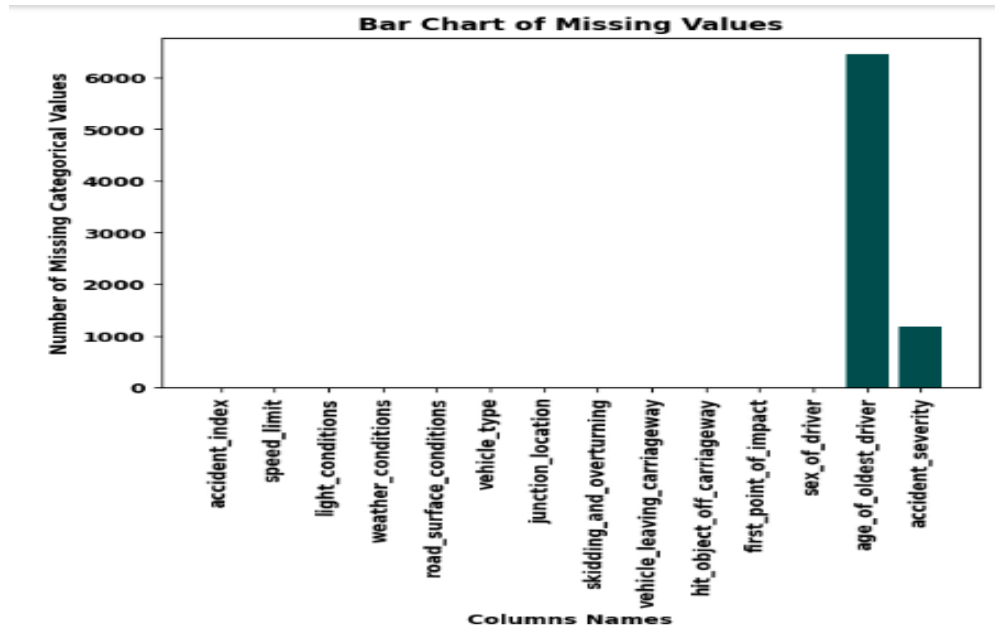


Figure 3: Missing values.

I have then found all the values with its number of occurrences for all categorical columns which is shown in table 2. All columns except “light\_conditions” have data missing or out of range. Also, “accident\_severity” has duplicates.

Column Name: “light_conditions”	
daylight	22210
darkness	9437

Column Name: “weather_conditions”	
fine	25152
other	5407
<b>data missing or out of range</b>	<b>970</b>
fog or mist	118

Column Name: “road_surface_conditions”	
dry	22092
wet or damp	8761
other	374
<b>data missing or out of range</b>	<b>365</b>
flood over 3cm. deep	55

Column Name: “vehicle_type”	
Only cars	17487
at least one biped	8662
atleast one van	4040
biped and van	1000
other	395
<b>data missing or out of range</b>	<b>63</b>

Column Name: “junction_location”	
at or within 20 metres of junction	17204
not at or within 20 metres of junction	12973
<b>data missing or out of range</b>	<b>1470</b>

Column Name: “skidding_and_overturning”	
no skidding or overturning	24387
at least one vehicle skidded or overturned	5226
<b>data missing or out of range</b>	<b>2034</b>

Column Name: “vehicle_leaving_carriageway”	
none leaving carriageway	24019
at least one vehicle leaving carriageway	5661
<b>data missing or out of range</b>	<b>1967</b>

Column Name: “hit_object_off_carriageway”	
none hit an object	26129
at least one vehicle hit an object	3655
<b>data missing or out of range</b>	<b>1863</b>

Column Name: “first_point_of_impact”	
at least one vehicle with frontal impact	23002
other points of impact	6008
no impact	1333

<b>data missing or out of range</b>	<b>1304</b>
-------------------------------------	-------------

<b>Column Name: “sex_of_driver”</b>	
all males	15494
male and female	7015
<b>data missing or out of range</b>	<b>5122</b>
all females	4016

<b>Column Name: “accident_severity”</b>	
slight	12672
serious	11592
fatal	6159
<b>Serious</b>	<b>25</b>
<b>Slight</b>	<b>19</b>
<b>Fatal</b>	<b>8</b>

Table 2: Values of the categorical columns and their number of occurrences.

There are many variables to consider while performing machine learning tasks which makes it challenging to model. The dataset I am working with has 14 variables. By identifying a collection of principal variables, dimension reduction is a technique for minimizing the number of random variables in a task. (Anon., 2018) The dimensionality reduction technique I have used is Principal Component Analysis(PCA) which creates a new principal component coordinate system from data. PCA plots help to identify how many components are required to capture a high variance in data.

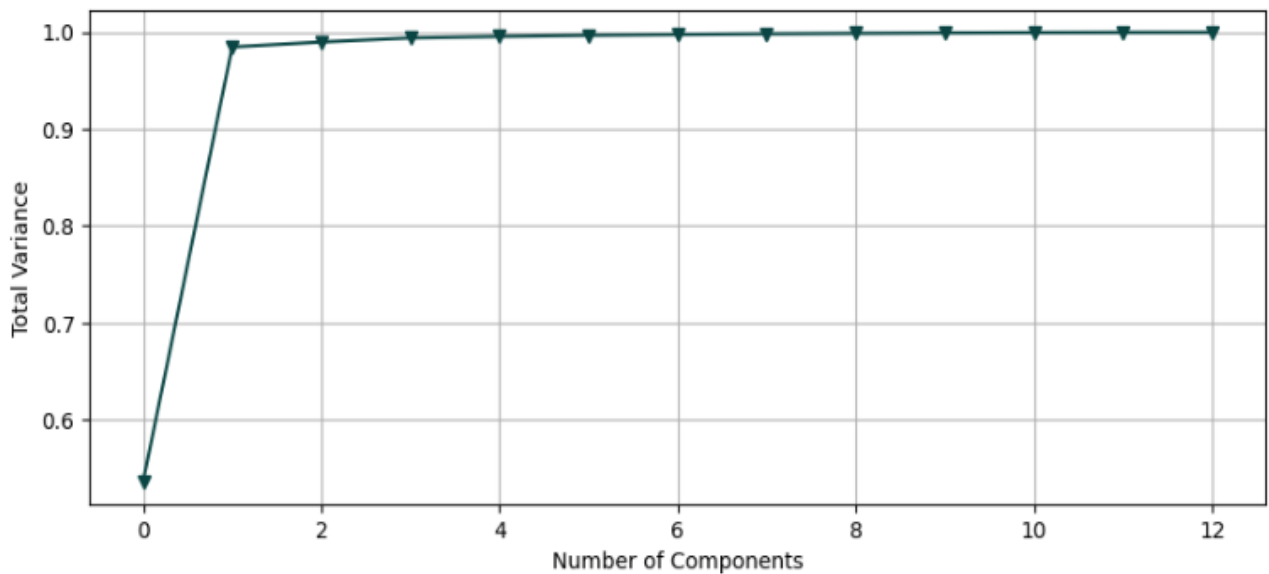


Figure 4: PCA.

Figure 4 shows PCA of all the variables including the target variable. The plot shows variance of

each component. Using 1 component explains around 80% of the variance. Using 3 to 12 components explains 100% of the variance. This indicates that using the first 3 components is sufficient to retain majority of the information from the dataset. I have also performed PCA after scaling the data.

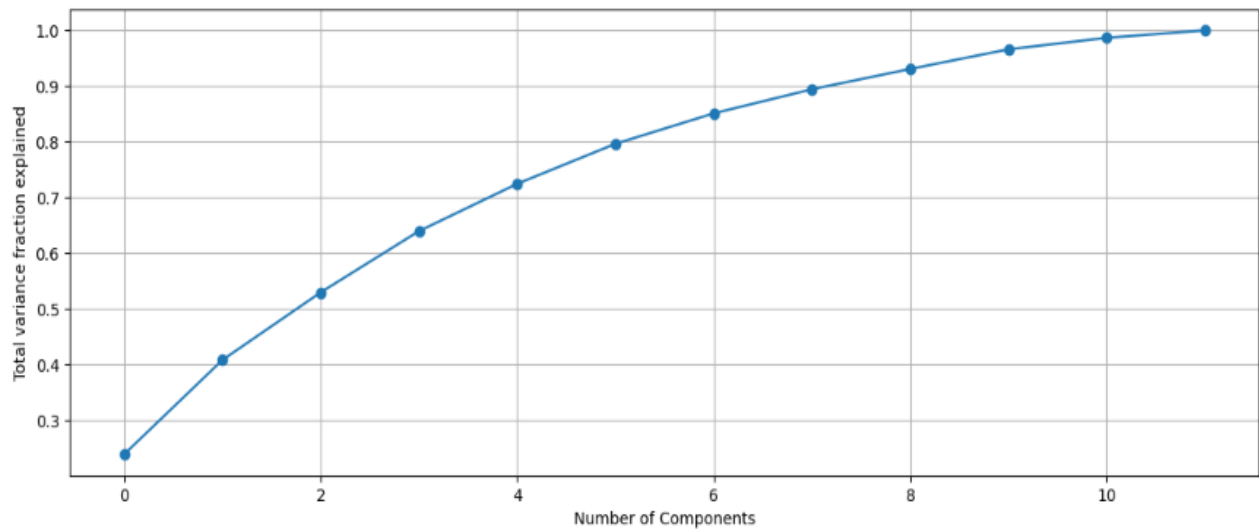


Figure 5: PCA after Scaling.

Figure 5 shows using 3 components explains 65% variance, whereas it was different in figure 4. Scaling may be the reason for the differences, as without it, some variables may have dominated the analysis due to their high values. After scaling, each variable contributes to the PCA analysis evenly.

## 2. Data preprocessing

Data preprocessing is modifying the dataset to make it correctly formatted, free of errors and missing data. This is important when implementing models as the data may have errors, missing, inaccurate and inconsistent data. Such data may not give accurate results, may take longer training times, or give biased results.(Singh, 2023)

In figure 3 it has been identified that “age\_of\_oldest\_driver” has NaN values. To correct this, I have used imputation which is the process of substituting other values for missing data.(Anon., 2023) I have used median imputation where the missing values are replaced by the median value. In this way all information is preserved.

In figure 2 an invalid speed limit of -1 has been identified, for which I have used a conditional statement to drop all rows with value less than 0. The result is shown in figure 6.

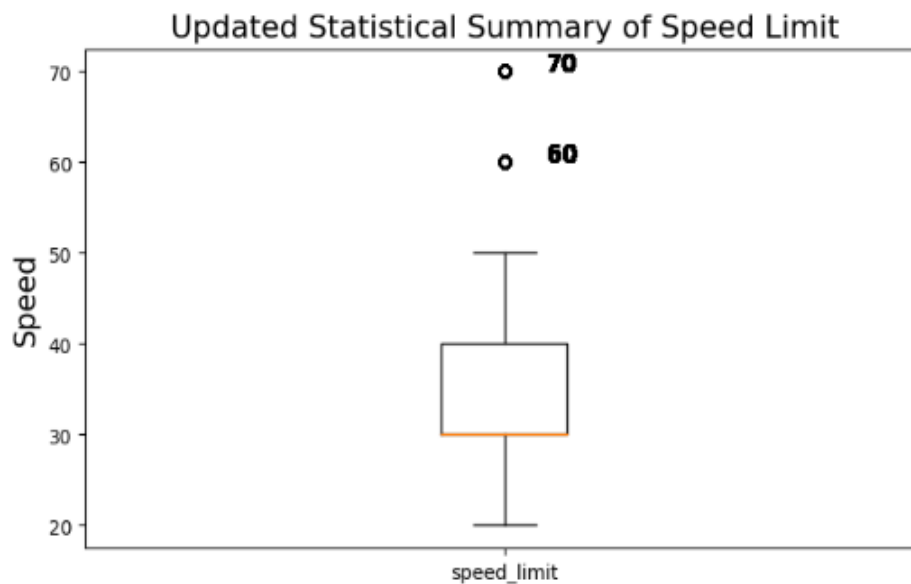


Figure 6: Updated “speed\_limit”.

To correct the duplicates of “accident\_severity” identified in table 2, I used a conditional statement, and the result is shown in table 3.

Column Name: “accident_severity”	
slight	12674
serious	11613
fatal	6164

Table 3: Column “accident\_severity” after removing duplicates.

Next, I converted all missing or out of range values identified in table 2 to NaN so that I can impute.



Figure 7 below shows the number of converted NaN values.

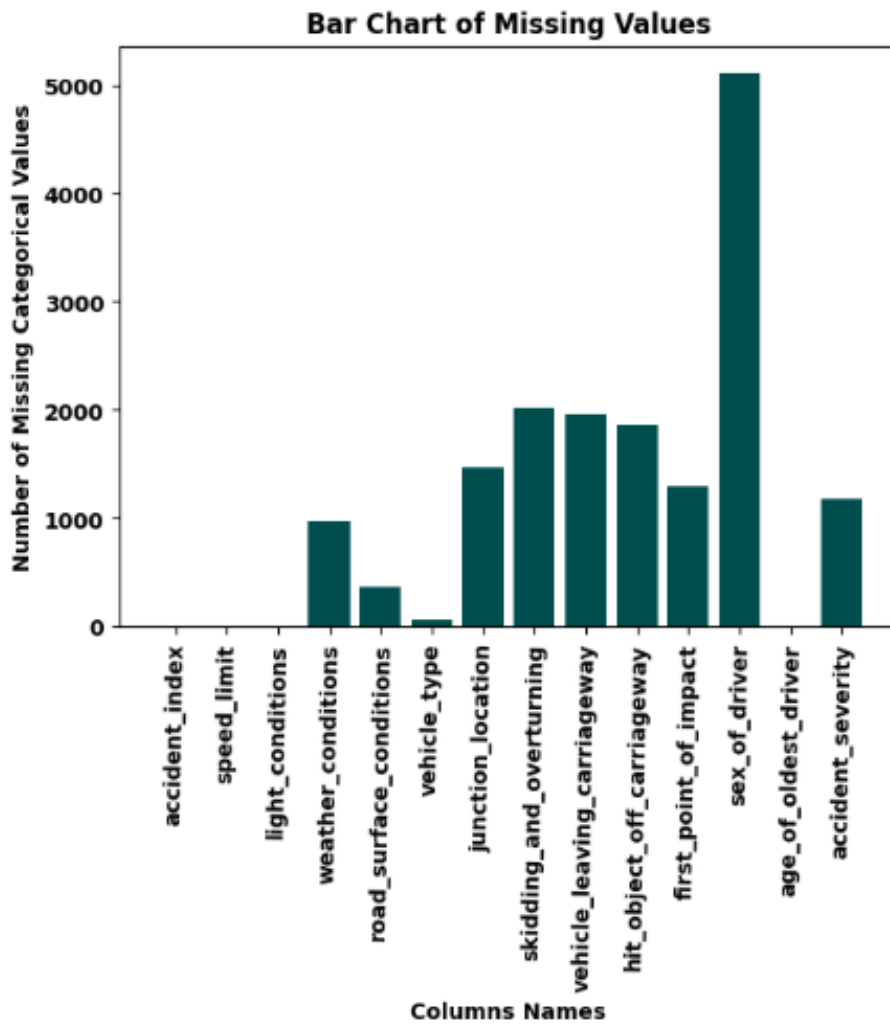


Figure 7: NaN values.

All NaN values are then replaced with most frequent values using imputation. Then, I have converted all the categorical values into a numerical form using label encoding. Then I found the imbalance of the target column shown in figure 8.

## Accident Severity Categories

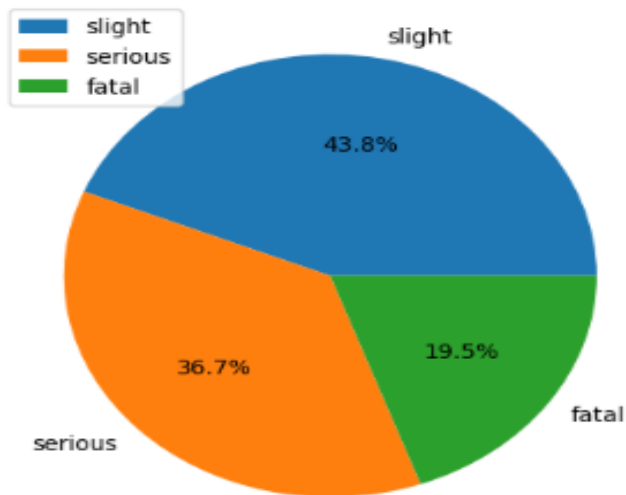


Figure 8: Imbalance.

Figure 8 shows an imbalance, where “fatal” consists of only 19.5% which is less compared to other variables. This may result in bad performance. Next, I split the dataset into X and Y(target variable). As data is imbalanced, I performed oversampling using SMOTE(Synthetic Minority Over-sampling Technique) which increases the sample size of the minority classes without overfitting. Figure 9 shows the data is balanced after oversampling.

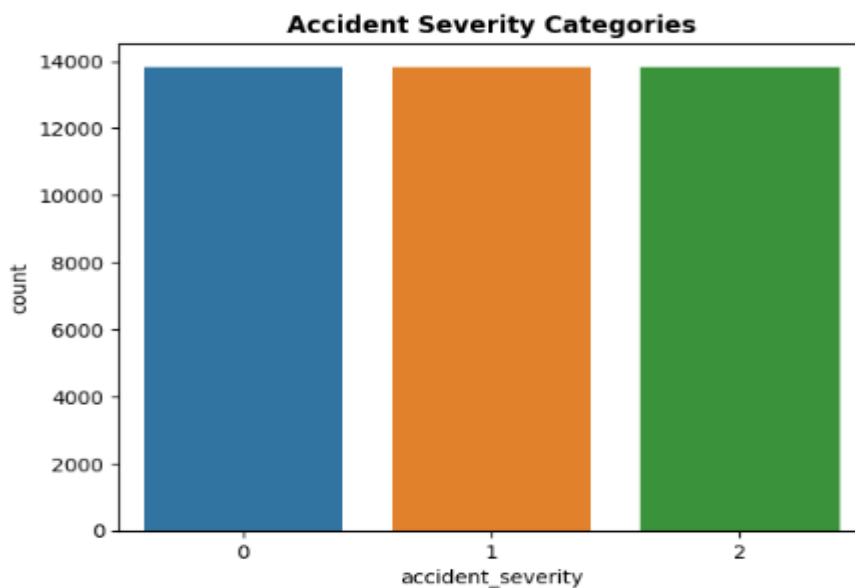


Figure 9: Balanced data.

Scaling is a technique for distributing the independent features in data uniformly over a predetermined range. Without scaling, machine learning algorithms tend to give greater weight to larger values, despite the value's unit. For scaling I have used Min-Max scaling. I have then split the dataset into training and testing sets with a test size of 20%.

### 3. Classification using traditional machine learning

A supervised machine learning technique known as classification predicts the correct label of a given input data. The classification models I have used are:

- Random Forest Classifier
- Decision Tree Classifier
- Linear Support Vector Classification.
- Gradient Boosting Classifier

The preprocessed data has been used to implement all these classification models with the best hyperparameters that control the learning process. I have used GridSearchCV to determine the optimum hyperparameter, which uses cross validation to improve the hyperparameters of the model. For all models I have implemented the same process. First, I have mentioned the different hyper parameters of the model. The model is evaluated for each combination of hyperparameters using the cross-validation approach as GridSearchCV examines all possible combinations of hyperparameters. (Anon., 2022)

After identification of the best hyperparameter combination, it is used to predict. For evaluating model performance, I have used two performance metrics, which are accuracy and recall. I used accuracy, because when the target class is evenly distributed, it provides reliable results, and the reason for using recall is to identify the number of accurate positive predictions made among all possible positive predictions. (Brownlee, 2020)

For comparison with a “trivial” baseline I used the Dummy Classifier with the strategy “most frequent” with which an accuracy of 33% is achieved. Figure 11 shows the model made 2743 correct predictions of label 0.

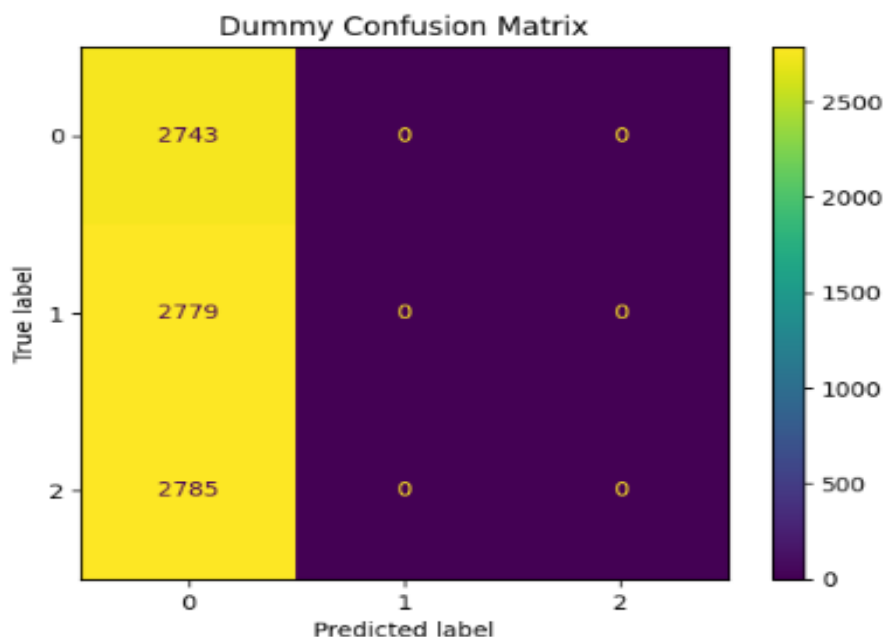


Figure 11: Dummy Classifier Confusion Matrix

Below I have compared the different hyperparameters for each model and showed its results using confusion matrix and performance metrics.

- **Random Forest Classifier**

For random forest classifier, I have used three hyperparameters which are criterion, max\_features and min\_samples\_split. For these hyperparameters I have given a range of different values. For example, for max features, I have given the values “sqrt” and “log2”. As mentioned above GridSearchCv examines all possible combinations of hyperparameters to get the best results. Table 5 shows some combinations of hyperparameters with their accuracy.

	<b>criterion</b>	<b>max_features</b>	<b>min_samples_split</b>	<b>Validation Accuracy</b>
0	gini	sqrt	2	0.776424
11	entropy	log2	10	0.795173
16	log_loss	log2	5	0.788733
<b>17</b>	<b>log_loss</b>	<b>log2</b>	<b>10</b>	<b>0.794661</b>

Table 5: Hyperparameter combinations for Random Forest Classifier.

It can be seen that different hyperparameter combinations have resulted in different accuracies for the model. It can be identified that with criterion=log\_loss, max\_features=log2 and min\_samples\_split=10, the validation accuracy is 0.795, which is the highest accuracy among all other combinations. Thus, this set of hyperparameters is chosen for prediction.

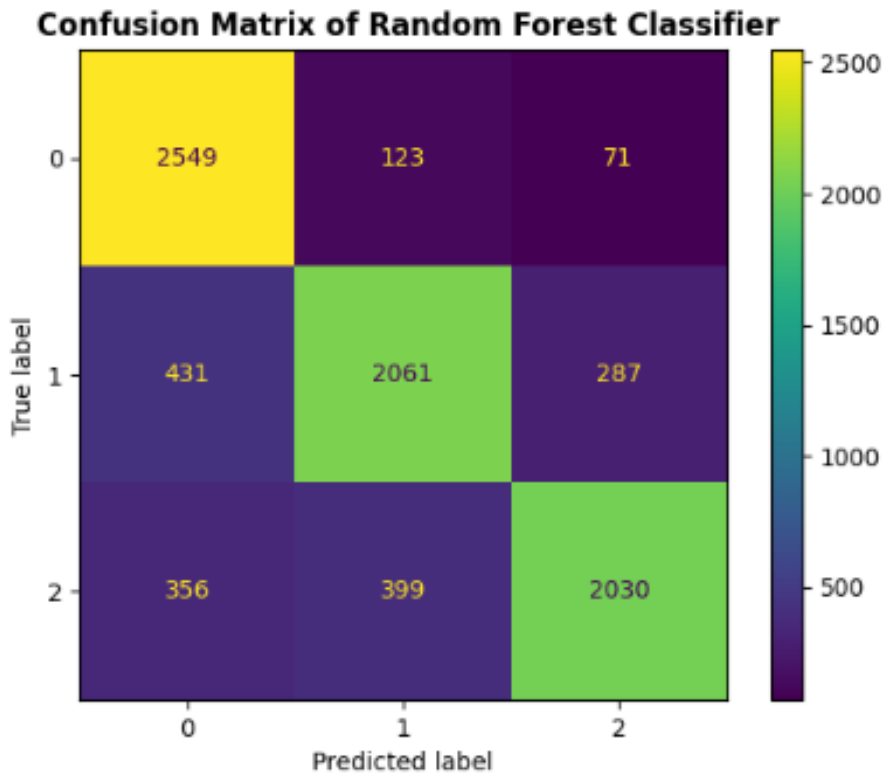


Figure 12: Confusion Matrix of Random Forest Classifier

Figure 12 shows the amount of correct and incorrect predictions. It can be seen that 2549 predictions of the label 0 are correctly made and 787 predictions are incorrect. Similarly, for the label 1, 2061 predictions are correctly made, and 522 predictions are incorrect. Lastly, for label 2, 2030 predictions are correct, and the rest 358 predictions are incorrect.

Performance Metrics		
Accuracy	0.80	
Recall	0	0.93
	1	0.74
	2	0.73

Table 6: Performance Metrics for Random Forest Classifier

Table 6 shows the accuracy is 80% which shows the model is predicting very well. It can also be said from recall, that the model made 93% correct predictions for label 0, 74% correct predictions for label 1 and 73% correct predictions for label 2.

- **Decision Tree Classifier**

For decision tree classifier I have used 4 hyperparameters which are criterion, max\_features, min\_samples\_split and splitter. Table 6 shows the combinations of hyperparameters with its accuracies.

	criterion	max_features	min_samples_split	splitter	Validation Accuracy
50	log_loss	log2	5	best	0.750693
51	log_loss	log2	5	random	0.754455
52	log_loss	log2	10	best	0.762219
<b>53</b>	<b>log_loss</b>	<b>log2</b>	<b>10</b>	<b>random</b>	<b>0.763483</b>

Table 7: Hyperparameter combinations for Decision Tree Classifier.

Table 7 shows, that with criterion=log\_loss, max\_features=log2, min\_samples\_split=10 and splitter=random, the validation accuracy is 0.7635, which is the highest accuracy among all other combinations. Thus, this set of hyperparameters is chosen for prediction.

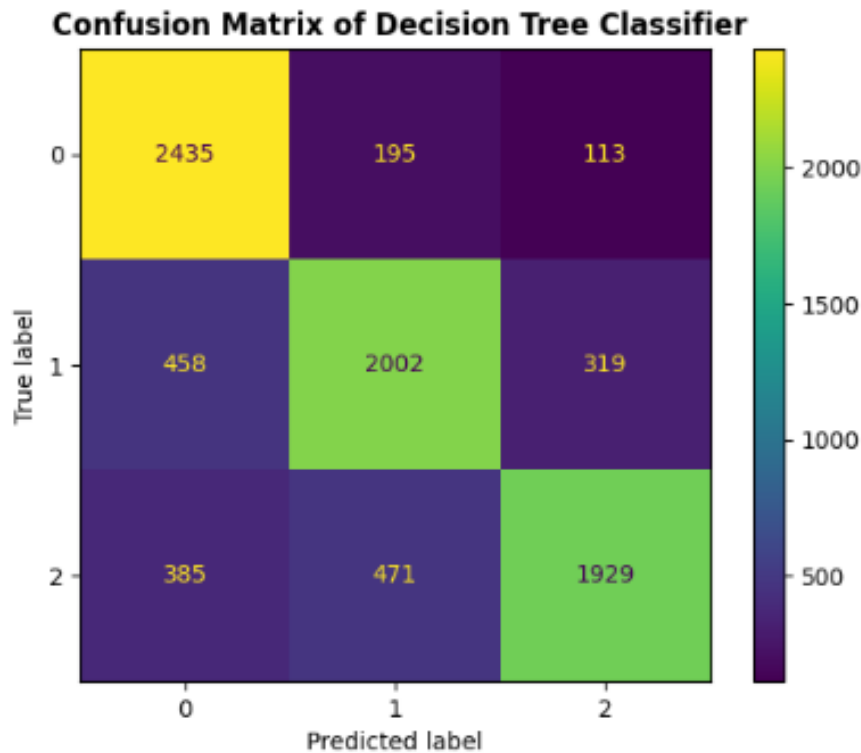


Figure 13: Confusion Matrix of Decision Tree Classifier

Figure 13 shows there are 2435 correct predictions for the label 0, 2002 correct predictions for the label 1 and 1929 correct predictions for the label 2.

Performance Metrics		
Accuracy	0.77	
Recall	0	0.89
	1	0.72
	2	0.69

Table 8: Performance Metrics for Decision Tree Classifier.

Table 8 shows the accuracy is 77% which indicates the model is predicting well. Moreover, it can be said from recall, that the model made 89% correct predictions for label 0, 72% correct predictions for label 1 and 69% correct predictions for label 2.



- **Linear Support Vector Classification.**

For linear SVM I have used 3 hyperparameters which are loss, multi\_class and penalty. Table 9 shows combinations of hyperparameters and its accuracies.

	loss	multi_class	penalty	Validation Accuracy
3	hinge	crammer_singer	l2	0.698598
<b>5</b>	<b>squared_hinge</b>	<b>ovr</b>	<b>l2</b>	<b>0.736066</b>
7	squared_hinge	crammer_singer	l2	0.698718

Table 9: Hyperparameter combinations for Linear Support Vector Classification.

Table 9 shows that with loss= squared\_hinge, multi\_class=ovr, penalty=l2, the validation accuracy is 0.74, which is the highest accuracy among all other combinations. Thus, this set of hyperparameters is chosen for prediction.

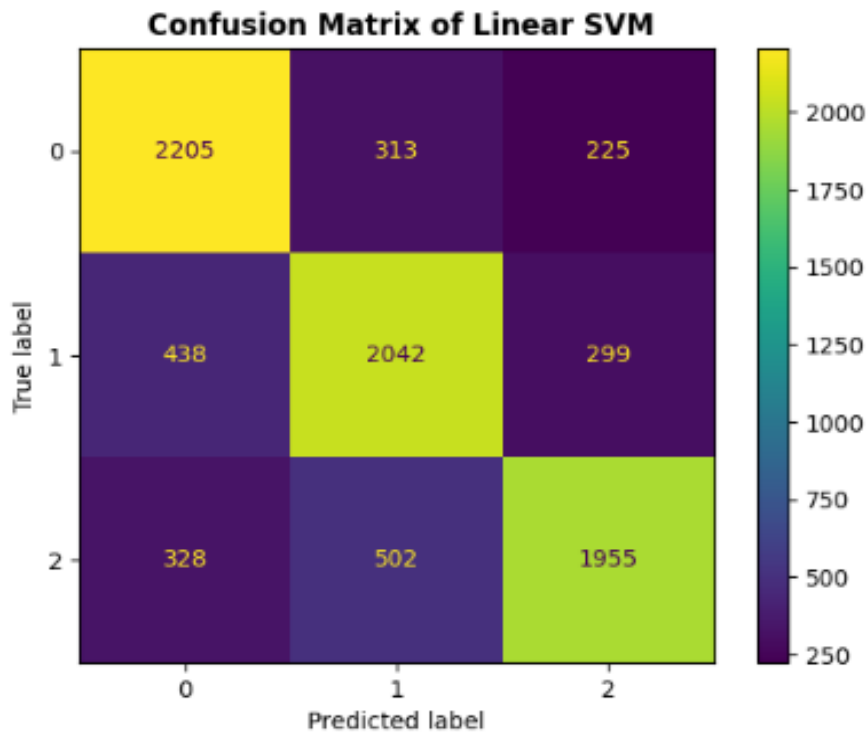


Figure 14: Confusion Matrix of Linear SVM.

Figure 14 shows 2205 correct predictions for label 0, 2042 correct predictions for label 1 and 1955 correct predictions for label 2.

Performance Metrics		
Accuracy	0.75	
Recall	0	0.80
	1	0.73
	2	0.70

Table 10: Performance Metrics for Linear SVM.

Table 10 shows the accuracy is 75%. Moreover, it can be said from recall, that the model made 80% correct predictions for label 0, 73% correct predictions for label 1 and 70% correct predictions for label 2.

- **Gradient Boosting Classifier**

For gradient boosting classifier I have used 3 hyperparameters which are loss, learning\_rate and min\_samples\_leaf. The contribution of each tree decreases as learning\_rate increases. Table 11 shows hyperparameter combinations.

	learning_rate	loss	min_samples_leaf	Validation Accuracy
20	1.00	log_loss	4	0.791080
22	1.00	deviance	2	0.790297
23	1.00	deviance	4	0.791080

Table 11: Hyperparameter combinations for Gradient Boosting Classifier.

Table 11 shows, that with loss= deviance, min\_samples\_leaf =4 and learning\_rate=1, the validation accuracy is 0.791, which is the highest accuracy among all other combinations. Thus, this set of hyperparameters is chosen for prediction.

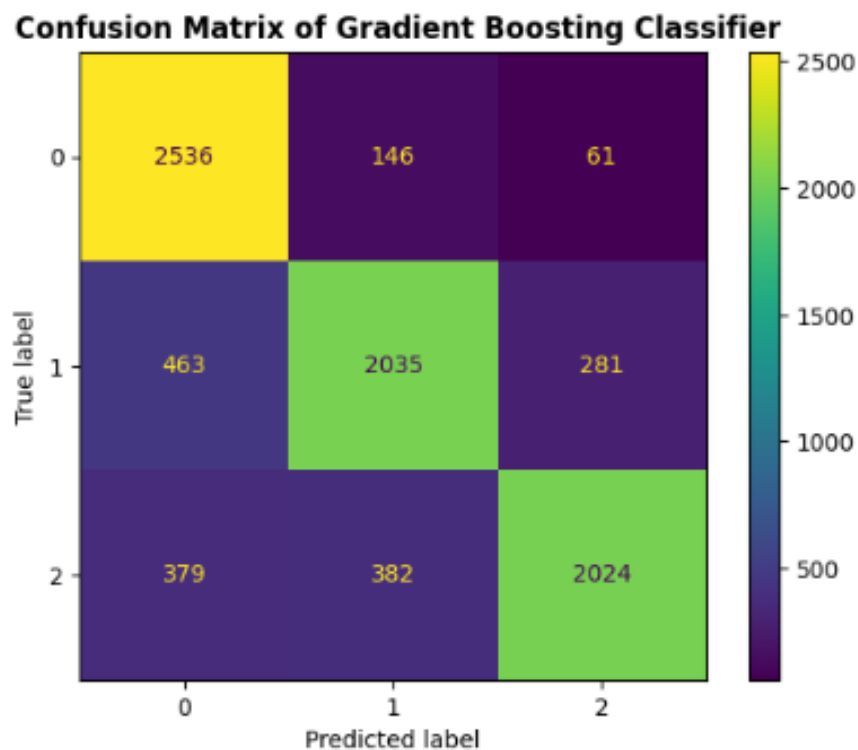


Figure 15: Confusion Matrix of Gradient Boosting Classifier.

Figure 15 shows 2536 correct predictions for label 0, 2035 correct predictions for label 1 and 2024 correct predictions for label 2.

<b>Performance Metrics</b>		
Accuracy	0.79	
Recall	0	0.92
	1	0.73
	2	0.73

Table 12: Performance Metrics for Gradient Boosting Classifier.

Table 12 shows the accuracy is 79% which shows the model is predicting very well. In addition, it can be said from recall, that the model made 92% correct predictions for label 0, 73% correct predictions for label 1 and 73% correct predictions for label 2.

### **Experiments:**

For experimenting I have tried different models and compared accuracies to find the best model. Moreover, I have tried different hyperparameters. In random forest regressor I experimented using the hyperparameters “max\_features” and min\_samples\_split and achieved 79%. But later using more hyperparameters I received 80%.

### **Comparison of Models:**

<b>Classification Models</b>	<b>Accuracy</b>
<b>Random Forest Classifier</b>	<b>80%</b>
Decision Tree Classifier	77%
Linear SVM Classifier	75%
Gradient Boosting Classifier	79%
Dummy Classifier	33%

Table 12: Comparison between accuracy scores.

From table 12, it can be identified that the best performing model is Random Forest Classifier with the highest accuracy of 80%. It also performed well in comparison to dummy classifier. This shows that the random forest classifier outperformed dummy classifier significantly because it accurately predicted 47% more instances.

## 4. Classification using neural networks

A neural network is used for training computers to process data in a manner that is similar to the way the human brain does. (Anon., n.d.) In neural network, data enters the model from input layer, passes it through hidden layers, and then is sent to the output layer. The hidden layer multiplies the sum of the weights and bias and generates output through an activation function.

To predict accident severity, I have used sequential model which created a linear stack of layers. For the layers, I have used the dense layer which is a fully connected layer with 20 hidden layer nodes and a batch size of 64. I have used the activation function “softmax” as it is appropriate for multi-class classification problems. Also, I have used the optimizer “adam” with a learning rate of 0.001, which updates the networks parameters. For loss function I have used the “SparseCategoricalCrossentropy”, which is appropriate for multi-class classification and when labels are integer. As I have performed label encoding, all the labels are integers. These hyperparameters are passed to GridSearchCV to find the best combination of hyperparameter along with a dictionary of two other hyperparameters 'optimizer\_\_learning\_rate' and 'model\_\_hidden\_layer\_nodes'.

	<b>model__hidden_layer_nodes</b>	<b>optimizer__learning_rate</b>	<b>Validation Accuracy</b>
0	20	0.010	0.784901
2	30	0.010	0.787148
<b>4</b>	<b>40</b>	<b>0.010</b>	<b>0.786340</b>
5	40	0.001	0.778100

Table 13: Hyperparameter combination of Neural Network.

Table 13 shows that with model\_hidden\_layer\_nodes=40 and optimizer\_\_learning\_rate=0.010, the model gives the best accuracy of 79%. This hyperparameter combination is then used to predict.

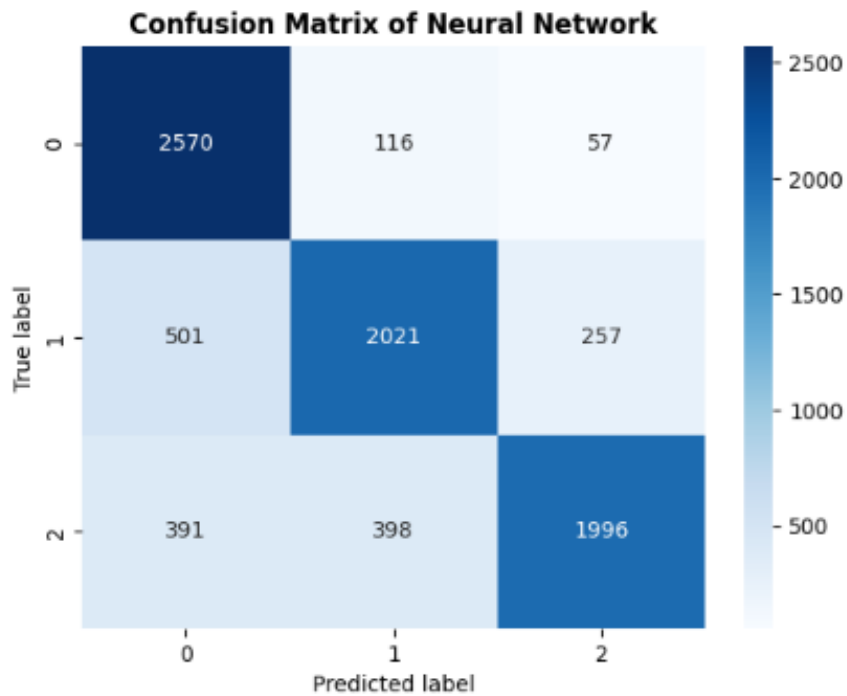


Figure 16: Confusion Matrix of Neural Network.

Figure 16 shows that for the label 0, 2570 predictions are correct. There are 2021 correct predictions for label 1 and 1996 correct predictions for label 2.

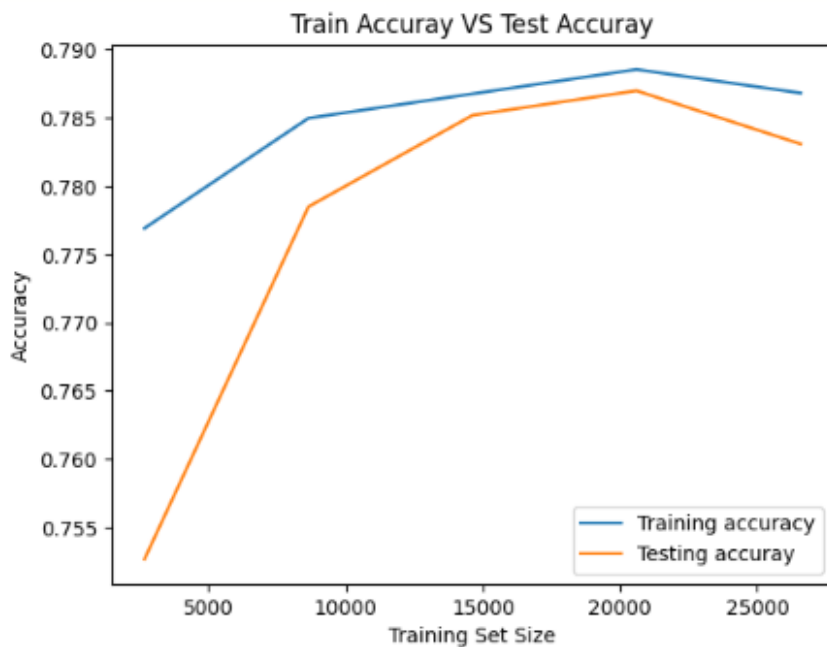


Figure 17: Train Score VS Test Score.

Figure 17 shows that as the training set size increases, the difference between training accuracy and testing accuracy reduces, showing the model is performing well.

Performance Metrics		
Accuracy	0.79	
Recall	0	0.94
	1	0.73
	2	0.72

Table 14: Performance Metrics for Neural Network.

Table 14 shows the accuracy is 79% which shows the neural network model is predicting very well. In addition, it can be said from recall, that the model made 92% correct predictions for label 0, 72% correct predictions for label 1 and 74% correct predictions for label 2.

### **Experiments:**

For experimenting I have tried different hyperparameters to achieve the best accuracy. I have used different values of epochs and batch size, and with 20 epochs and a batch size of 24, I received 77% accuracy, which gave a low accuracy compared to the accuracy received with the chosen combination of final hyperparameter.

### **Comparison of Models:**

Classification Models	Accuracy
<b>Random Forest Classifier</b>	<b>80%</b>
Dummy Classifier	33%
Neural Network	79%

Table 15: Comparison

Table 15 shows an accuracy score of 33% with dummy classifier which indicates neural network outperformed random guessing significantly because it accurately predicted 46% more instances than the dummy classifier. However random forest classifier performed slightly better than neural network.

## 5. Ethical discussion




Label name	Label description	Label image	Reason for applying	Relevant safety precautions
Reinforcing existing biases.	The prediction accuracies may be impacted for biasness in dataset. The bias may have happened due to an unfair way of recording accidents. For example-the accidents that are not severe may not be recorded.		Machine learning models cannot identify biasness due to unfair way of recording.	Before predicting, equality of each label in the dataset should be checked to find biasness.
Difficult to understand	The codes used for predictions is important as the hyperparameters used in the models make a difference in the results. The wrong usage of hyperparameters may result in wrong results. However, the code is hard for people with no machine learning knowledge to interpret.		The codes maybe complicated for general people to understand.	The codes should contain in-depth information as comments, for public to understand.
Automates decision making.	GridSearchCV automates the hyperparameter tuning procedure. The combination of hyperparameters chosen by GridSearchCV has an impact on the model's accuracy. Wrong predictions may impact the accident severity forecasts.		Finding a wrong combination of hyperparameter may cause a wrong prediction. This may impact the accident severity forecasts.	The results should be reviewed and tested multiple times to find errors.

Table 16: Data Hazard Labels.



## **6. Recommendations**

- The best model is Random Forest Classifier, because it gave the highest accuracy of 80%. Moreover, the recall score for 3 of the labels is above 70%, which indicates it predicted most of the labels correctly.
- The final model is good to be used in practice, because it identifies the best combinations of hyperparameter using GridSearchCV, which results in a good accuracy score.
- There are other models for classification which I did not implement. My suggestion is to implement other models to check if they give better accuracy. Also, more data should be used, as it may result in better training of the model.

## **7. Retrospective**

If I were to start the coursework again, I would invest more time in preprocessing the data to find more ways of removing all the problems from the dataset to get better accuracy. For example- I have used label encoding, but I would also have tried one hot encoding to see if it gives better accuracy. In addition, I would utilize the heat map to identify correlation and use the strongly correlated features in the dataset to make predictions to check if they give better results.

## 8. References

Anon., 2018. *dimensionality reduction*. [Online]

Available at: <https://www.techtarget.com/whatis/definition/dimensionality-reduction#:~:text=Dimensionality%20reduction%20is%20a%20machine,a%20set%20of%20principa%20variables>.

[Accessed 9 April 2023].

Anon., 2022. *perparameter Tuning with GridSearchCV*. [Online]

Available at:

<https://www.mygreatlearning.com/blog/gridsearchcv/#:~:text=GridSearchCV%20tries%20all%20the%20combinations,one%20with%20the%20best%20performance>.

[Accessed 9 April 2023].

Anon., 2023. *Introduction to Data Imputation*. [Online]

Available at: <https://www.simplilearn.com/data-imputation-article#:~:text=Imputation%20in%20statistics%20refers%20to,constituent%20of%20a%20data%20p oint>.

[Accessed 8 April 2023].

Anon., n.d. *What Is A Neural Network?*. [Online]

Available at: <https://aws.amazon.com/what-is/neural-network/#:~:text=A%20neural%20network%20is%20a,that%20resembles%20the%20human%20brai n>.

[Accessed 09 April 2023].

Barla, N., 2023. *Dimensionality Reduction for Machine Learning*. [Online]

Available at: <https://neptune.ai/blog/dimensionality-reduction#:~:text=Advantages%20and%20disadvantages->

[What%20is%20dimensionality%20reduction%3F,variables%20are%20also%20called%20features](https://neptune.ai/blog/dimensionality-reduction#:~:text=Advantages%20and%20disadvantages-).

[Accessed 9 April 2023].

Brownlee, J., 2020. *How to Calculate Precision, Recall, and F-Measure for Imbalanced Classification*. [Online]

Available at: <https://machinelearningmastery.com/precision-recall-and-f-measure-for-imbalanced-classification/#:~:text=Recall%20is%20a%20metric%20that,indication%20of%20missed%20positive%20predictions>.

[Accessed 9 April 2023].

Singh, S., 2023. *Importance of Pre-Processing in Machine Learning*. [Online]

Available at: <https://www.kdnuggets.com/2023/02/importance-preprocessing-machine-learning.html#:~:text=In%20conclusion%2C%20preprocessing%20data%20before,the%20interpreta bility%20of%20the%20model>.

[Accessed 8 April 2023].

