Project 4 - Big data

Hosted by Julia Begley, Hayley Fuller, Alemseghed Ghebrezghi, Diego Lorenzo, Lewis Russell

Project overview

Focus:

- Tableau shows us that around 75-80% of campaigns were classified as successful, so how can machine learning make this % better?
- Using Tableau what further analysis can we find. How do these findings compare to machine learning?
- What will machine learning do to predict a successful campaign?
- Can machine learning predict a higher success rate? If so then what does a successful campaigns need?
- Is the success rate prediction in machine learning the same as the data in Tableau?

Tableau Analysis

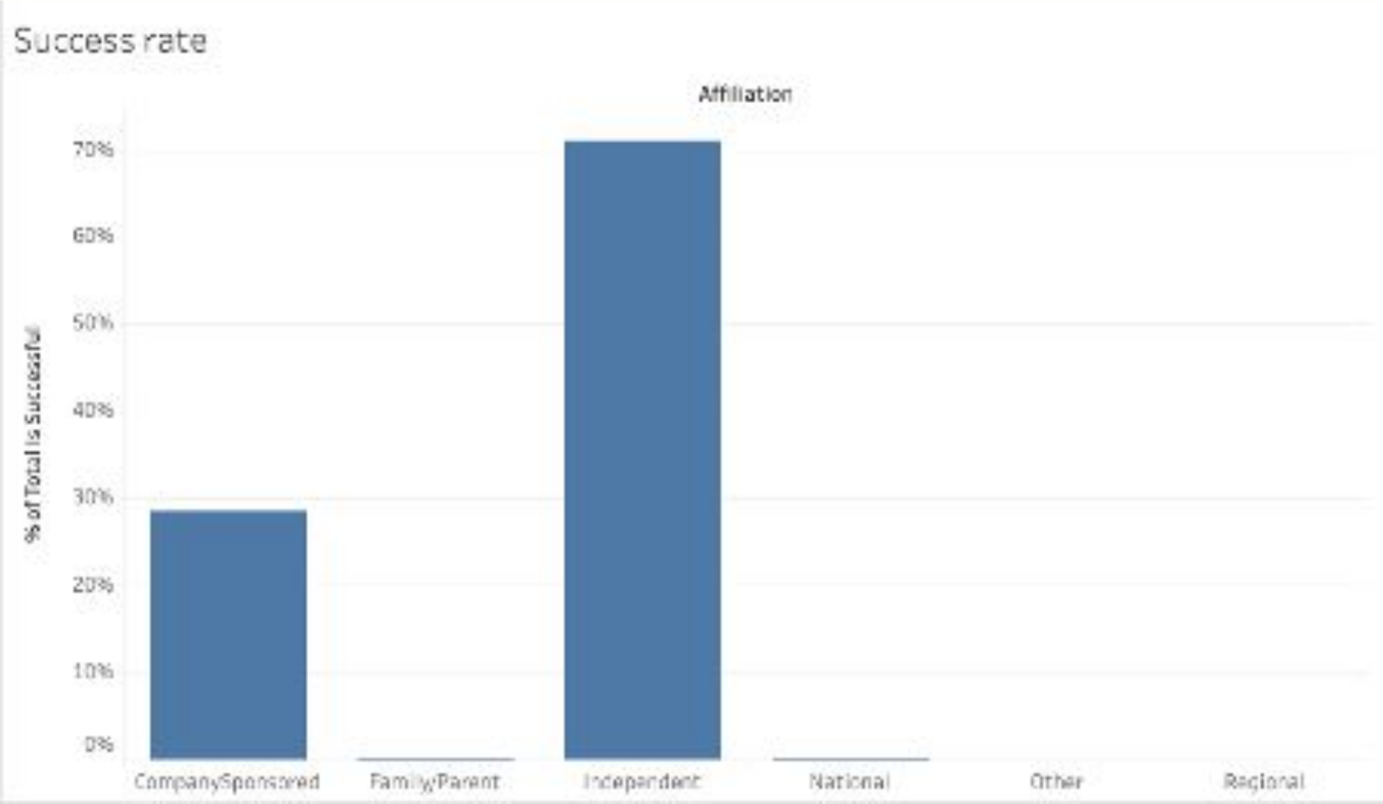
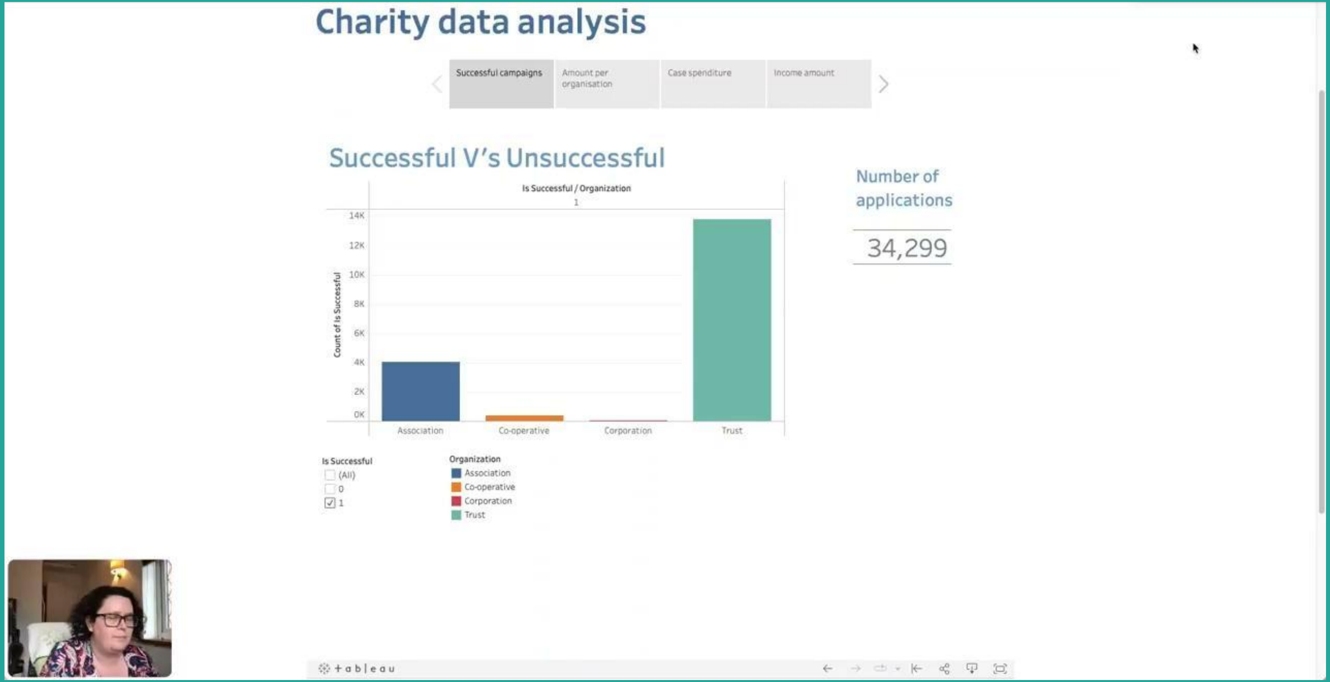


Tableau Analysis

Story 1 Dashboard



Charity data analysis

Successful V's Unsuccessful

Number of applications

34,299

Is Successful
☐ (All)
☐ 0
☑ 1

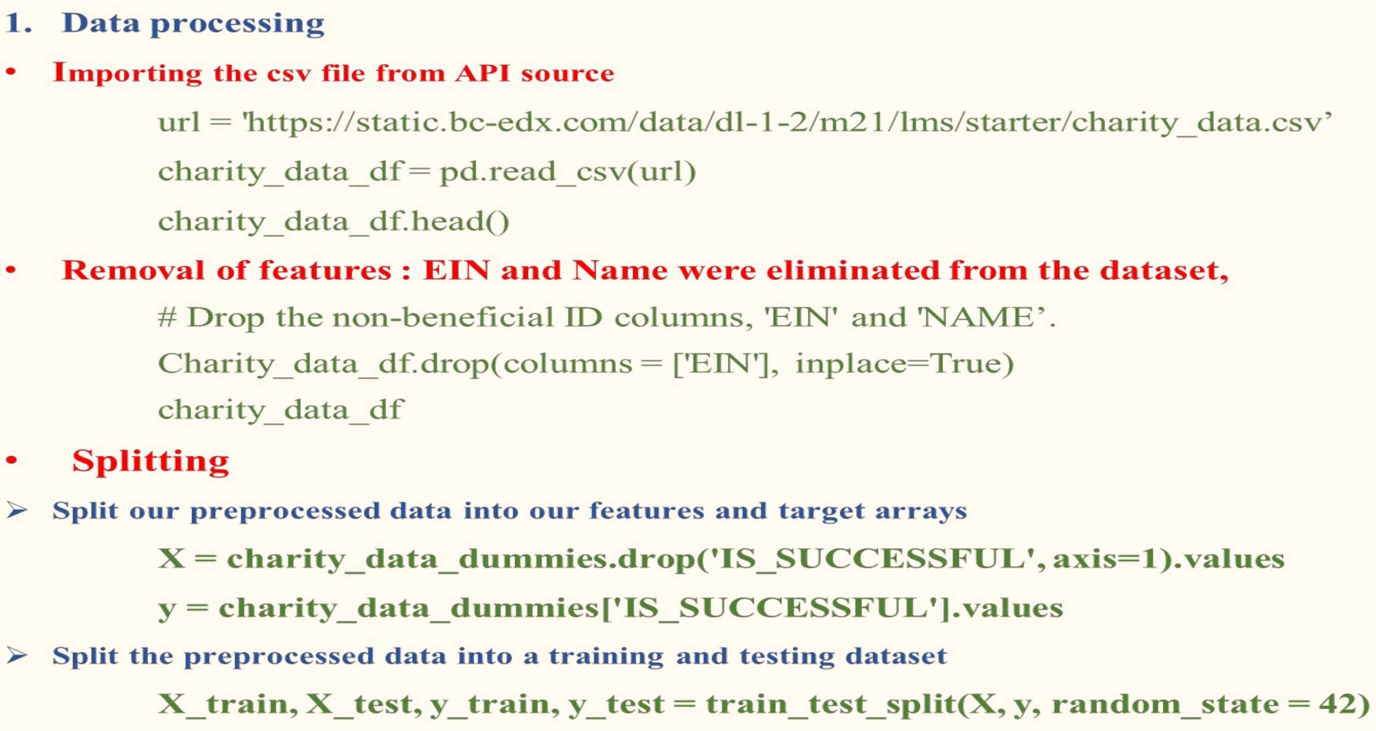
Organization
🟦 Association
🟧 Co-operative
🟥 Corporation
🟩 Trust

ML Base code

1. Overview

- The Alphabet Soup Nonprofit Foundation sought a binary classifier in order to facilitate the selection of applicants for funding.
- The binary classifier is designed to assess the likelihood of successful applicants being funded by Alphabet Soup.
- Preprocessing, compiling, training, evaluating the model and model optimization have been done all in a single jupyter notebbok file.

Project 4_ group 1: ML Base code



1. **Data processing**

- **Importing the csv file from API source**

```
url = 'https://static.bc-edx.com/data/dl-1-2/m21/lms/starter/charity_data.csv’
charity_data_df = pd.read_csv(url)
charity_data_df.head()
```

- **Removal of features : EIN and Name were eliminated from the dataset,**

```
# Drop the non-beneficial ID columns, 'EIN' and 'NAME’.
Charity_data_df.drop(columns = ['EIN'],  inplace=True)
charity_data_df
```

- **Splitting**

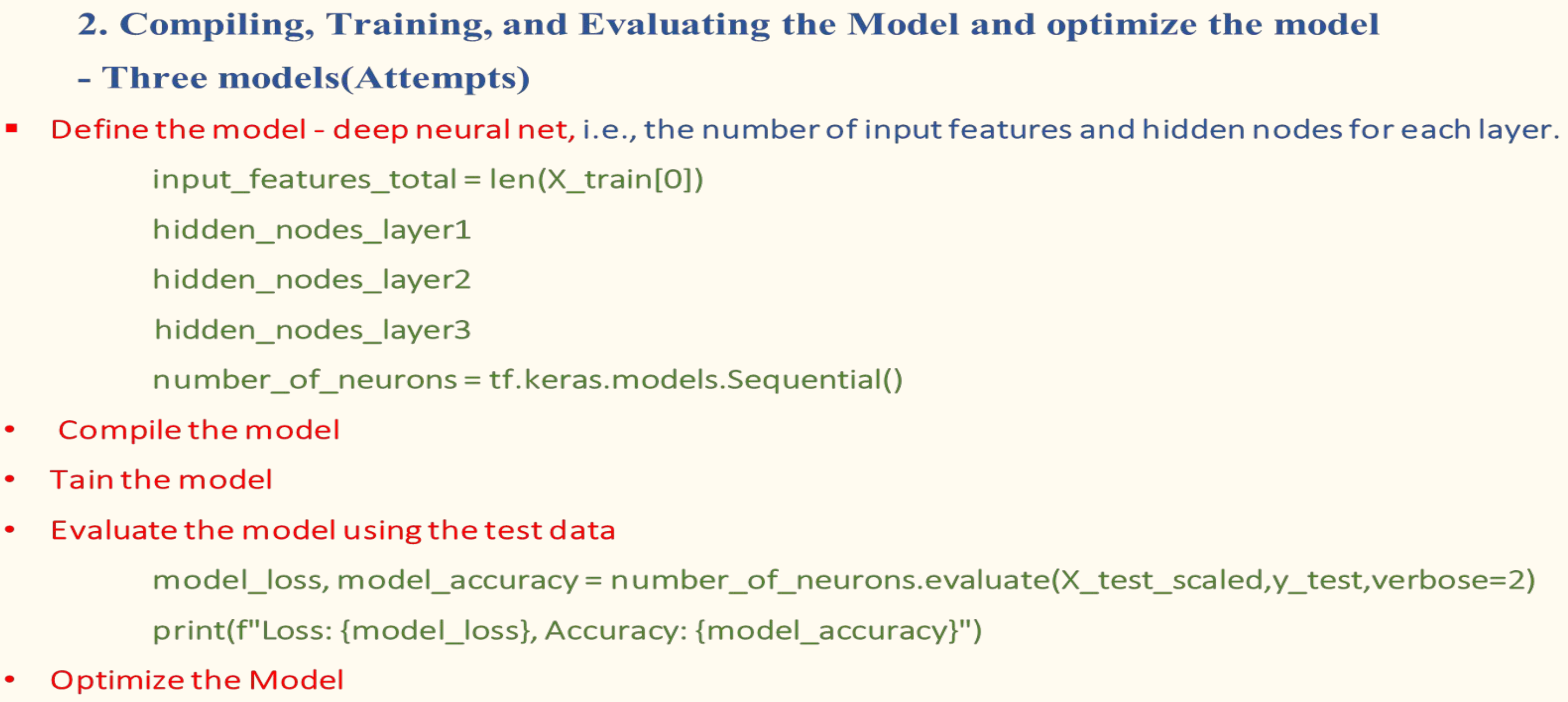
➢ **Split our preprocessed data into our features and target arrays**

```
X = charity_data_dummies.drop('IS_SUCCESSFUL', axis=1).values
y = charity_data_dummies['IS_SUCCESSFUL'].values
```

➢ **Split the preprocessed data into a training and testing dataset**

```
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state = 42)
```

Project 4_ group 1: ML Base code



2. Compiling, Training, and Evaluating the Model and optimize the model

- Three models(Attempts)

- Define the model - deep neural net, i.e., the number of input features and hidden nodes for each layer.

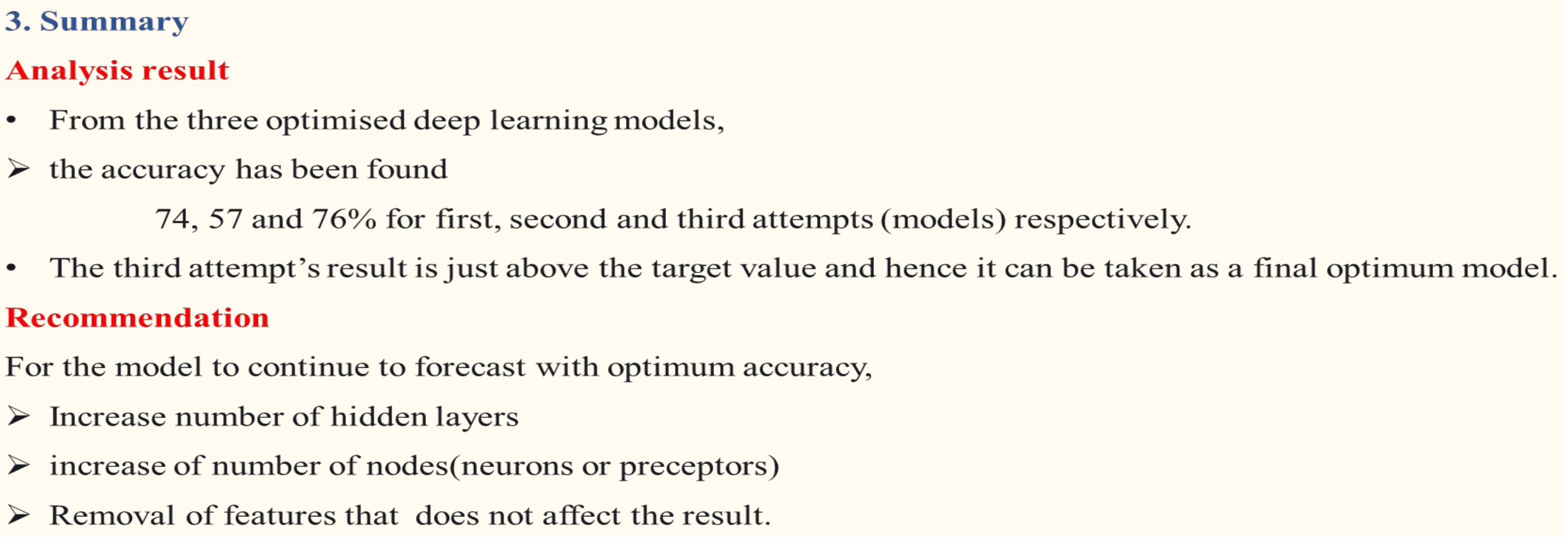
```
input_features_total = len(X_train[0])
hidden_nodes_layer1
hidden_nodes_layer2
hidden_nodes_layer3
number_of_neurons = tf.keras.models.Sequential()
```

- Compile the model
- Tain the model
- Evaluate the model using the test data

```
model_loss, model_accuracy = number_of_neurons.evaluate(X_test_scaled,y_test,verbose=2)
print(f"Loss: {model_loss}, Accuracy: {model_accuracy}")
```

- Optimize the Model

Project 4_ group 1: ML Base code



3. Summary

Analysis result

- From the three optimised deep learning models,

➢ the accuracy has been found

74, 57 and 76% for first, second and third attempts (models) respectively.

- The third attempt's result is just above the target value and hence it can be taken as a final optimum model.

Recommendation

For the model to continue to forecast with optimum accuracy,

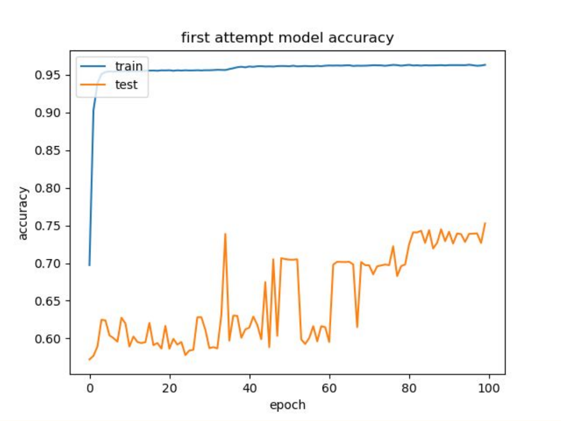
➢ Increase number of hidden layers

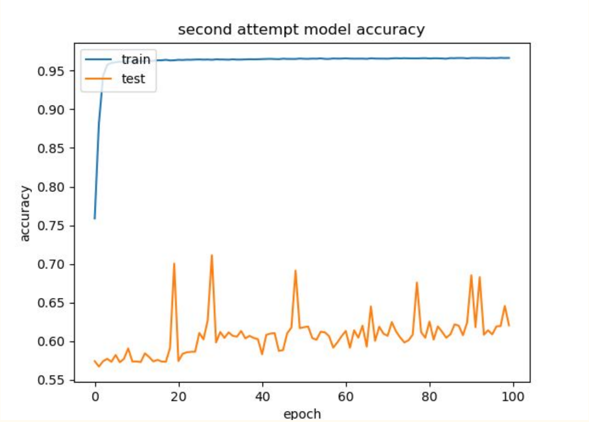
➢ increase of number of nodes(neurons or preceptors)

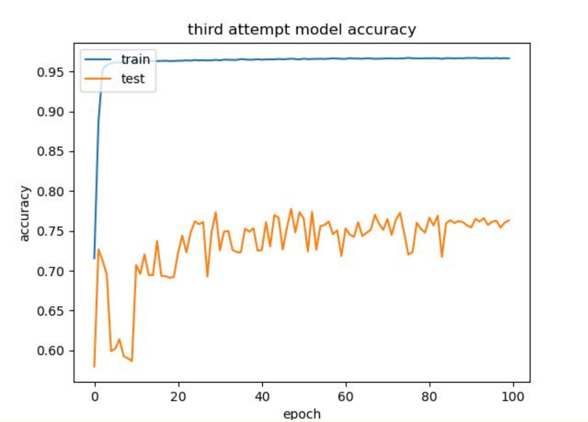
➢ Removal of features that does not affect the result.

Model accuracy by attempt

These graphs show us that the models are working as expected because the training and validation lines both show accuracy increasing over time.


first attempt model accuracy

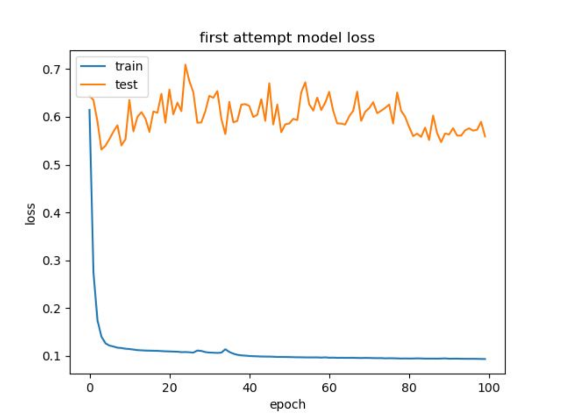

second attempt model accuracy

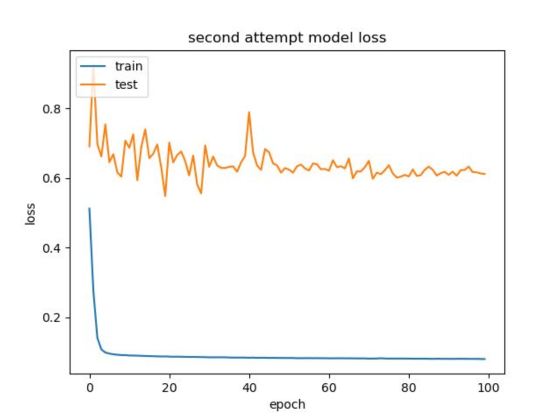

third attempt model accuracy

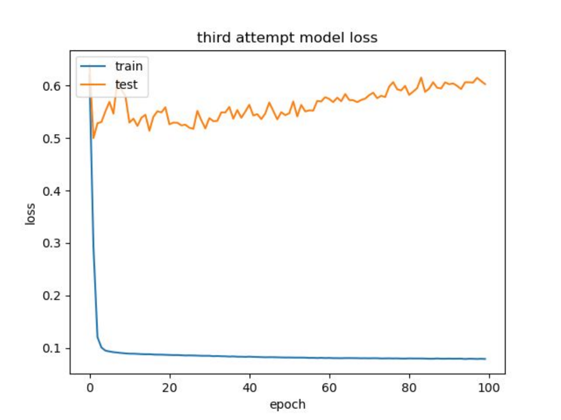
Model accuracy by model attempt

The loss graphs for the 1st and 2nd attempts at the model show they are working as expected as the training and validation loss scores both decrease over time.

However, the graph for the 3rd attempt indicates that it may be ‘overfitting’, i.e. matching too closely to the training data, and failing to fit well to the additional, testing data.

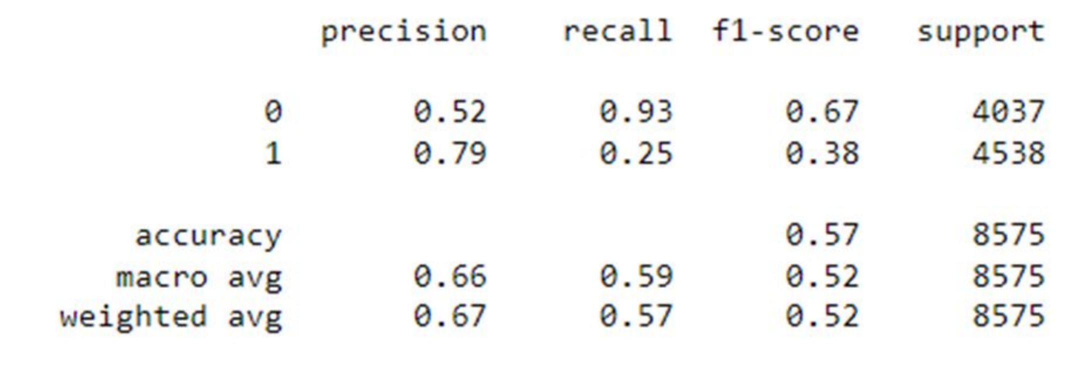






Classification Report for the test data

A Classification Report measures a model's quality of predictions using 3 metrics



```
              precision    recall  f1-score   support

           0       0.52      0.93      0.67      4037
           1       0.79      0.25      0.38      4538

    accuracy                           0.57      8575
   macro avg       0.66      0.59      0.52      8575
weighted avg       0.67      0.57      0.52      8575
```

1. **Accuracy:** how often the model is correct, the percentage of correctly predicted observations to the total number of observations for the entire dataset. How many Trues (TP +TN) over all outcomes (TP + TN) / (TP + TN + FP + FN).

   The model was right/accurate **57%** of the time.

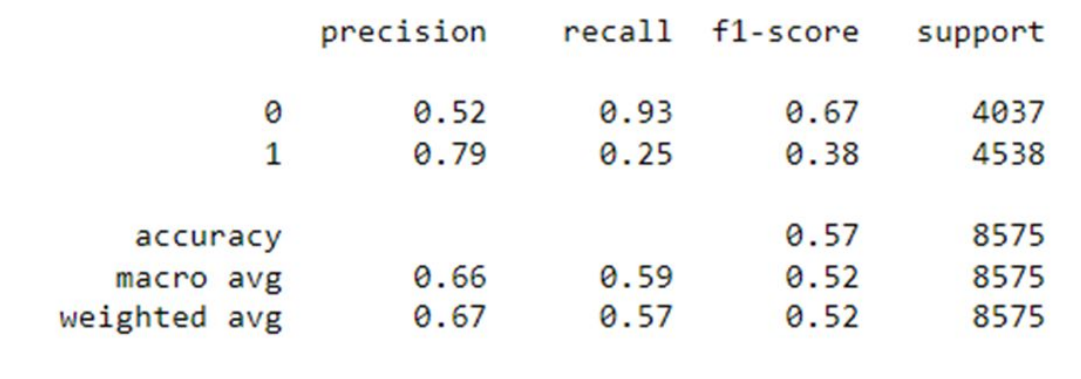
2. **Precision**: percentage of correctly predicted positive observations to the total predicted positive observations. High precision relates to a low false positive rate, how many true positives I had over all positives: TP / (TP + FP).

   Out of the 34,000+ organisations that received funding from Alphabet Soup over the years, **79%** used the money effectively.

3. **Recall**: percentage of correctly predicted positive observations to all predicted observations for that class: TP / (TP + FN). .High recall correlates to a low false negative rate.

Of all the organisations that received funding the model correctly predicted **93%** of the time which organisations didn’t use the money effectively; and out of all the organisations that actually used the money effectively, the model only predicted this outcome correctly for 25% of those organisations.

Classification Report for the test data



```
              precision    recall  f1-score   support

           0       0.52      0.93      0.67      4037
           1       0.79      0.25      0.38      4538

    accuracy                           0.57      8575
   macro avg       0.66      0.59      0.52      8575
weighted avg       0.67      0.57      0.52      8575
```

CONCLUSION

a) According to the Classification Report results, the model was correct 57% of the time which is not very accurate.

b) Recall is the most reliable metric to consider, as the model is able to correctly predict 93% of the time which organisations will fail to use the money effectively.

<u>In order to improve the model Accuracy it would be recommended to</u>:

I. **Add more datapoints** in order to improve the model accuracy, as well as to

II. **Further Exploratory Data Analysis**:
E.g.: Avoid Class Imbalance (Oversampling vs Undersampling) in order to have a balanced dataset. For this dataset this analysis was done using the **class_counts** function, which showed that this dataset is slightly unbalanced as ~53% of organisations used the money effectively against ~47% that didn't.

III. Feature Engineering:
Selecting the features that best capture the pattern in the dataset (give us the best correlation) in order to improve a model's quality of predictions, by using tools like **Random Forest** -helps selecting those columns that improve a model prediction-, or Gridsearch that gives the best combination of parameters.

Conclusions

“We will then consider the differences, advantages and disadvantages of data analysis done by only humans vs. analysis supported/directed by ML”

In Tableau successful applications have

1. An income amount of between 25,000 to 99,999
2. Use the money to Preserve the charity
3. Are classified as a Trust
4. Be independently affiliated

In machine learning successful applications…