Homework 21

Purpose :

The purpose of this analysis is to help select the applicants for funding with best chances of success in their ventures. This can be achieved by machine learning and neural networks.

Data Preprocessing :

Before doing analysis on the data we have to make sure that there are no null values and dataset is cleaned to perform analysis. Below are some of the data cleaning done on the dataset.

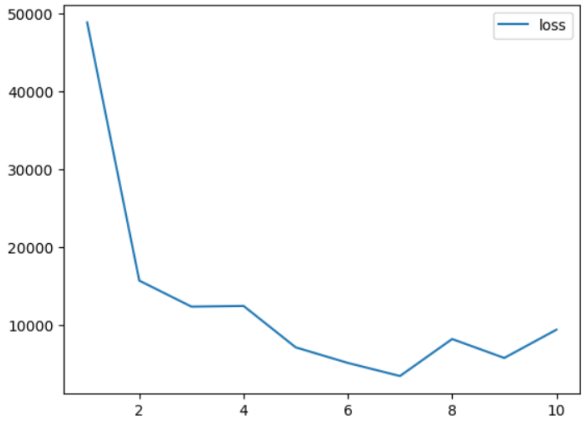
a. Below is the list of all columns dataset originally had
   ['EIN', 'NAME', 'APPLICATION_TYPE', 'AFFILIATION', 'CLASSIFICATION', 'USE_CASE', 'ORGANIZATION', 'STATUS', 'INCOME_AMT', 'SPECIAL_CONSIDERATIONS', 'ASK_AMT', 'IS_SUCCESSFUL']
b. The target for the model is the 'IS_SUCCESSFUL' column, signifying if the money was use effectively
c. Column 'EIN' and "NAME" were dropped as they are not numeric and may be not adding any value in the analysis.
   i. Application_type less than 200 were grouped into others.
   ii. Classification less than 500 were grouped into others.
d. One- hot encoding was performed using get_dummies() on the dataframe.
e. The dataset was scaled for ASK_AMT, all the binary columns were not scaled.

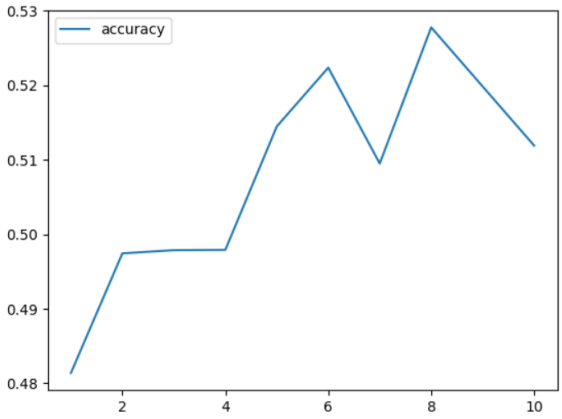
Compiling, Training, and Evaluating the Model:

1. Iteration 1

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 7)	308
dense_1 (Dense)	(None, 5)	40
dense_2 (Dense)	(None, 1)	6

```
268/268 - 0s - 767us/step - accuracy: 0.5324 - loss: 6.4255
Loss: 6.4254937171936035, Accuracy: 0.5323615074157715
```

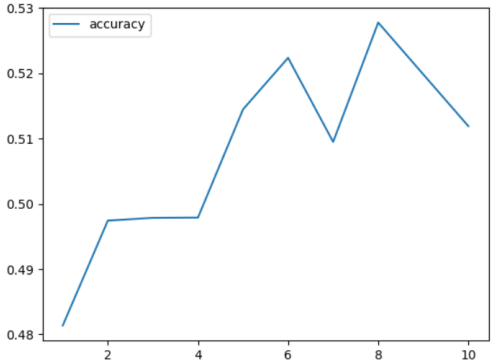




2. Iteration 2

Layer (type)	Output Shape	Param #
dense_3 (Dense)	(None, 15)	660
dense_4 (Dense)	(None, 11)	176
dense_5 (Dense)	(None, 7)	84
dense_6 (Dense)	(None, 1)	8

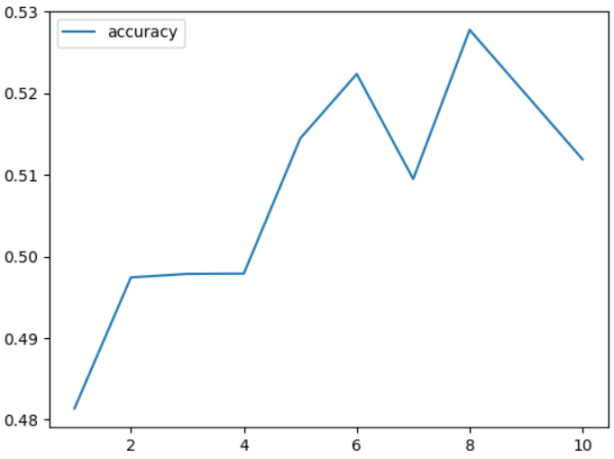
```
268/268 - 0s - 1ms/step - accuracy: 0.4083 - loss: 0.7002
Loss: 0.700214147567749, Accuracy: 0.4082798957824707
```



3. Iteration 3

Layer (type)	Output Shape	Param #
dense_7 (Dense)	(None, 19)	836
dense_8 (Dense)	(None, 13)	260
dense_9 (Dense)	(None, 9)	126
dense_10 (Dense)	(None, 5)	50
dense_11 (Dense)	(None, 1)	6

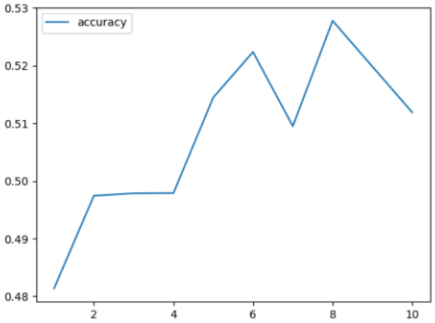
```
268/268 - 0s - 2ms/step - accuracy: 0.5284 - loss: 0.6939
Loss: 0.6938753724098206, Accuracy: 0.528396487236023
```



4. Iteration 4

Layer (type)	Output Shape	Param #
dense_12 (Dense)	(None, 111)	4,884
dense_13 (Dense)	(None, 61)	6,832
dense_14 (Dense)	(None, 31)	1,922
dense_15 (Dense)	(None, 11)	352
dense_16 (Dense)	(None, 1)	12

```
268/268 - 0s - 2ms/step - accuracy: 0.5324 - loss: 0.6913
Loss: 0.691345751285553, Accuracy: 0.5323615074157715
```



5. Random Forest Classifier:

```
]: from sklearn.metrics import accuracy_score
   from sklearn.ensemble import RandomForestClassifier
```

```
]: # Create a random forest classifier.
   rf_model = RandomForestClassifier(n_estimators=128, random_state=78)

   # Fitting the model
   rf_model = rf_model.fit(X_train_scaled, y_train)

   # Evaluate the model
   y_pred = rf_model.predict(X_test_scaled)
   print(f" Random forest model accuracy: {accuracy_score(y_test,y_pred):.3f}")
```

```
 Random forest model accuracy: 0.713
```

In this model the first, second, third and fourth hidden layers were modified along with neurons on each layer. Also, shown above are the accuracy for each iteration. The accuracy varies but not much from 0.40 to 0.53. The target value of accuracy was not able to achieved by NN.

A good model to recommend is the Random Forest model because Random Forest are good for classification problems. Using this model produces a 72% accuracy, still less than requested.