Module 21 Report

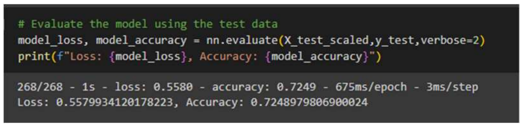
1. **Overview** of the analysis: Explain the purpose of this analysis.

The purpose of this analysis is to determine if a charity is successful or unsuccessful based on several input variables. The inputs are the application type, affiliation, classification, use case, organization, status, income amount, special considerations, and ask amount. The model takes these inputs and determines if the charity will be successful based on other existing charities.

2. **Results**: Using bulleted lists and images to support your answers, address the following questions:

- Data Preprocessing
    - What variable(s) are the target(s) for your model?
        - Targets: IS_SUCCESSFUL
    - What variable(s) are the features for your model?
        - Features: application type, affiliation, classification, use case, organization, status, income amount, special considerations, and ask amount
    - What variable(s) should be removed from the input data because they are neither targets nor features?
        - EIN and NAME, these are identification columns which are not useful in the model

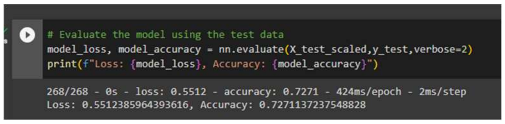
- Compiling, Training, and Evaluating the Model
    - How many neurons, layers, and activation functions did you select for your neural network model, and why?
        - For the initial model, I chose 2 hidden layers, each with 5 neurons and 1 activation function, as I thought this would be a high enough number to ensure the model has a high success rate
        - For the optimized model, I chose to add an additional hidden layer for a total of 3 hidden layers. Another optimization I made was to increase the number of neurons in each hidden layer to 7. For my final optimization, I increased the number of epochs from 100 to 150.
    - Were you able to achieve the target model performance?
        - The initial model did not reach 75% (image shown after Q2), it only reached 72.49%
        - The optimized model did not reach 75% (image shown after Q2), it only reached 72.71%
    - What steps did you take in your attempts to increase model performance?
        - Add an additional hidden layer for a total of 3 hidden layers.
        - Increase the number of neurons in each hidden layer to 7
        - Increased the number of epochs from 100 to 150.



```
# Evaluate the model using the test data
model_loss, model_accuracy = nn.evaluate(X_test_scaled,y_test,verbose=2)
print(f"Loss: {model_loss}, Accuracy: {model_accuracy}")
```

```
268/268 - 1s - loss: 0.5580 - accuracy: 0.7249 - 675ms/epoch - 3ms/step
Loss: 0.5579934120178223, Accuracy: 0.7248979806900024
```

Figure 1 Optimized Model Performance



```
# Evaluate the model using the test data
model_loss, model_accuracy = nn.evaluate(X_test_scaled,y_test,verbose=2)
print(f"Loss: {model_loss}, Accuracy: {model_accuracy}")
```

```
268/268 - 0s - loss: 0.5512 - accuracy: 0.7271 - 424ms/epoch - 2ms/step
Loss: 0.5512385964393616, Accuracy: 0.7271137237548828
```

Figure 2 Initial Model Performance

3. **Summary**: Summarize the overall results of the deep learning model. Include a recommendation for how a different model could solve this classification problem, and then explain your recommendation.

Overall, the model predicts only at around 72.5% accuracy. The optimizations helped slightly but did not bring the model’s performance up to 75%. Therefore, another model may be more helpful to solve this problem.