

[← Go Back to Introduction to Neural Networks](#)[☰ Course Content](#)

FAQ - Bank Churn Prediction

Q1. How should one approach the Bank Churn project?

- Before starting the project, please read the problem statement carefully and go through the criteria and descriptions mentioned in the rubric.
- Once you understand the task, download the dataset and import it into a Jupyter Notebook or Google Colab to get started with the project.
- To work on the project, you should start with data preprocessing and EDA using descriptive statistics and data visualization.
- In data preprocessing, you need to encode the independent target variables and standardize/normalize only numerical variables.
- Once the EDA is completed and the data is preprocessed, you can use the data to build a model, check its performance based on the desired metric, and if it is not good then train other neural networks.
- You should find the optimal value of the threshold using the ROC-AUC curve and use this threshold to convert probabilities into labels.
- You should include all the models that you have trained in your notebook.
- It is important to close the analysis with key findings and conclusions.

Q2. I keep getting the following error while installing TensorFlow

```
ImportError: DLL load failed while importing _pywrap_tensorflow_internal: The specified module could not be found. Failed to load
```

How to fix the above error?

You have installed a recent version that supports CPU and GPU and you don't have the correct version of CUDNN and CUDA libraries installed. Please try the below solutions one by one and see if any of them work:-

Solution 1

First, uninstall the existing TensorFlow:

```
!pip uninstall tensorflow
```

For re-installation, download this [TensorFlow wheel](#) file
Install this in the same Directory

```
!pip install tensorflow-1.6.0-cp36-cp36m-win_amd64.whl
```

Solution 2

If you want to have Tensorflow Version 2 with the least dependencies, use the below command:

```
!pip install tensorflow==2.15.0
```

Solution 3

Download MSVCP140.dll, unzip it, and then paste it into the system32 folder.

Link to the file: dli-files.com/msvc140.dll.html

Solution 4

Run this line of code in your Jupyter notebook cell

```
!pip install tensorflow --upgrade --force-reinstall
```

Final Solution

Did you try using the conda command?

```
conda install tensorflow==1.14.0 ## (Use this command in your anaconda prompt)
```

Please follow this [article](#) for more details

You can also use Google Colab for Deep Learning because installing TensorFlow sometimes gets difficult due to system configuration issues.

Q3. Is it mandatory to use Google Colab for this project?

No, it's not mandatory. You can use either Jupyter or Google Colab based on your convenience, but we would recommend using Colab as it offers comparatively more compute power and free access to GPU that will speed up code execution

Q4. I keep getting the below error while executing the fit function for the neural network. Why?

```
ValueError: Input 0 of layer sequential_55 is incompatible with the layer: expected axis -1 of input shape to have value 10000
```

This error is due to the incompatible shape of the input layer. Please pass the input shape in the first layer as follows:

```
v=X_train.shape
s=v[1]
Model2.add(Dense(number_of_units, input_shape=s, kernel_initializer='he_normal', activation='relu'))
```

This should use a compatible shape for the input layer that prevents the error from occurring.

Q5. I keep getting the below error while running the code on the Jupyter Notebook Why?

```
ModuleNotFoundError: No module named 'google.colab'
```

If you are using a Jupyter Notebook, then you need not install *google.colab*. This library is only used on Google Colab and it comes pre-installed on Colab.

Q6. Why does my model have a high training accuracy but low validation accuracy? What should I do to get a better validation accuracy?

In the above scenario, your model is probably overfitting. You should decrease the capacity of the model by reducing the number of layers or nodes per layer. You can also try using a Dropout layer between the hidden layers.

Q7. Why does my model have a high validation accuracy and low training accuracy? What should I do then?

There are a few possible reasons for this:

- You may have a small validation set
- You may have highly imbalanced data in the validation set (Use stratify= **Y** in the train_test_split method)

- If you use regularization methods such as L2, L1, or Dropout, while the model calculates training accuracy it uses a regularized model but when it tests accuracy on the validation set, it processes your data through an unregularized model. Regularization introduces some noise in loss value during training, because of this the training accuracy decreases more than expected, but while evaluating the model, the model doesn't use regularization and hence there's no noise, which is why the validation accuracy doesn't decrease.

Q8. Getting an error while importing the Optimizers from the Keras library?

Please try to import the optimizers using the TensorFlow library as below.

```
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.optimizers import RMSprop
```

Q9. I keep getting errors while importing libraries. Why?

Some of the libraries have version issues in the Jupyter Notebook, so use Google Colab to avoid any installation issues while importing libraries. Google Colab is a separate platform similar to Jupyter Notebook that has Pre-Installed Libraries. To get familiar with Google Colab, watch the Google Colab video in Pre: Work Deep Learning.

Q10. Do we need to build Machine Learning models for this project?

No need to build any Machine Learning models in this project since there are no points assigned in the rubric for Machine Learning models.

Q11. The CSV file name is bank.csv, but after downloading it, the name of the file is changed to Churn.csv. Are both files the same?

Yes, both the bank.csv and Churn.csv files are the same.

Q12. The dataset provided for this project is imbalanced. Is it necessary for me to employ any sampling techniques?

Yes, one can use sampling techniques to balance data and see if evaluation metrics are improved. No points will be deducted if sampling techniques are not applied to the data.

Q13. Is it required to bin numerical columns such as Estimated Salary, Age, and Balance before building a Neural Network model? What effect might binning have on a model?

Binning can be applied to all relevant numerical columns in the data to see if the evaluation metrics can be improved. The effect of binning on any ML or DL algorithm will only be seen after it has been implemented. It is difficult to interpret whether binning will have a positive or negative impact on any algorithm and will rather be a trial-and-error method.

Q14. Is it necessary to treat outliers in data?

Treating outliers mainly depends on the problem statement. Outliers cannot always be removed or treated because that is how the data is and should be kept for analysis.

Consider a bank as an example. In a bank, account balance might have high outliers as few people will have a high amount in their accounts, but should we be treating these values, or removing them? So, we have to consider that in practical life situations, we cannot remove these accounts, and if we replace them with any other value, that will make the data biased.

So the best option for treating the outliers would be taken for the problem statement at hand.

Q15. I keep getting the below error while executing the fit function for the neural network. Why?

```
ValueError: Input 0 of layer "sequential_1" is incompatible with the layer: expected shape=(None, 29), found shape=(None, 11)
```

This error is due to the incompatible shape of the input layer. The shape of the input layer given to the model should be the same as X_train. Thus, please pass the input shape in the first layer as follows:

```
Model2.add(Dense(number_of_units, input_shape=X_train.shape[1],  
kernel_initializer='he_normal', activation='relu'))
```

Q16. When should one-hot encoding and label encoding be used for categorical variables?

We generally prefer label encoding when there is a sense of order in the values. For example, suppose a variable has values bad, good, and very good in such a case we know that there is an order, and we can encode them as 1,2,3 respectively.

But let's say the values are red, blue, and green. In this case, there is no definite order in values, and hence, using one-hot encoding would be a better choice.

Q17. Is it necessary to use the stratify function in the train test split?

The parameter stratify is used to split the dataset into train and validation/test sets in such a way that the proportions of data points in each class remain the same as in the original dataset. Since the data is imbalanced, we would recommend you use the stratify function in the train test split.

[< Previous](#)[Next >](#)