**Instructions**

Open the starter file in Google Colab and complete the following steps, which are divided into three parts:

**Part 1: Preprocessing**

1. Import the data and view the first five rows.
2. Determine the number of unique values in each column.
3. Create y\_df with the attrition and department columns.
4. Create a list of at least 10 column names to use as X data. You can choose any 10 columns you’d like EXCEPT the attrition and department columns.
5. Create X\_df using your selected columns.
6. Show the data types for X\_df.
7. Split the data into training and testing sets.
8. Convert your X data to numeric data types however you see fit. Add new code cells as necessary. Make sure to fit any encoders to the training data, and then transform both the training and testing data.
9. Create a StandardScaler, fit the scaler to the training data, and then transform both the training and testing data.
10. Create a OneHotEncoder for the department column, then fit the encoder to the training data and use it to transform both the training and testing data.
11. Create a OneHotEncoder for the attrition column, then fit the encoder to the training data and use it to transform both the training and testing data.

**Part 2: Create, Compile, and Train the Model**

1. Find the number of columns in the X training data.
2. Create the input layer. Do NOT use a sequential model, as there will be two branched output layers.
3. Create at least two shared layers.
4. Create a branch to predict the department target column. Use one hidden layer and one output layer.
5. Create a branch to predict the attrition target column. Use one hidden layer and one output layer.
6. Create the model.
7. Compile the model.
8. Summarize the model.
9. Train the model using the preprocessed data.
10. Evaluate the model with the testing data.
11. Print the accuracy for both department and attrition.

**Part 3: Summary**

Briefly answer the following questions in the space provided:

1. Is accuracy the best metric to use on this data? Why or why not?
2. What activation functions did you choose for your output layers, and why?
3. Can you name a few ways that this model could be improved?

**Hints and Considerations**

* Review previous modules if you need help with data preprocessing.
* Make certain that your training and testing data are preprocessed in the same ways.
* Review Day 3 of this module for information on branching neural networks.

**Requirements**

**Preprocessing (40 points)**

* Import the data. (5 points)
* Create y\_df with the attrition and department columns. (5 points)
* Choose 10 columns for X. (5 points)
* Show the data types of the X columns. (5 points)
* Split the data into training and testing sets. (5 points)
* Encode all X data to numeric types. (5 points)
* Scale the X data. (5 points)
* Encode all y data to numeric types. (5 points)

**Model (40 points)**

* Find the number of columns in the X training data. (5 points)
* Create an input layer. (5 points)
* Create at least two shared hidden layers. (10 points)
* Create an output branch for the department column. (10 points)
* Create an output branch for the attrition column. (10 points)

**Summary (20 points)**

* Answer the questions briefly. (10 points)
* Show understanding of the concepts in your answers. (10 points)