Assignment 3

Lending Club is a US peer-to-peer lending company, headquartered in San Francisco, California. It was the first peer-to-peer lender to register its offerings as securities with the Securities and Exchange Commission (SEC), and to offer loan trading on a secondary market. Lending Club is the world's largest peer-to-peer lending platform.

Data Set: Lending Club DataSet (lending_club_loan.csv)
Data Set Overview: Please have a look at lending_club_info.csv regarding the meaning and description of each column in the given data set.

Goal: Given historical data on loans given out with information on whether or not the borrower defaulted (charge-off), we can build a model that can predict whether or nor a borrower will pay back their loan **using ANN**. This way in the future when we get a new potential customer, we can assess whether or not they are likely to pay back the loan.The "loan_status" column contains our label.

Note: It is mandatory to implement the tasks given below. But apart from doing these tasks, you can include your steps too. Creativity with justification is encouraged.

Assignment Tasks:

TASK 1: Exploratory Data Analysis

OVERALL GOAL: Get an understanding for which variables are important, view summary statistics, and visualize the data

TASK 1.1 : Analyze and visualize loan_status column. (Hint: Use countplot and histogram)

TASK 1.2: Explore correlation between the continuous feature variables. Calculate the correlation between all continuous numeric variables.

TASK 1.3: Visualize the task given in 1.2 using a heatmap.

TASK 1.4: Based on correlation analysis in step 1.2 and 1.3, do you think there is duplicate information here? If yes, please make a note, but do not remove anything at this step.

TASK 1.5 : Analyse the relationship between the loan_status and the Loan Amount using Boxplot.

TASK 1.6: Calculate the summary statistics for the loan amount, grouped by the loan_status.

TASK 1.7: Analyse the Grade and SubGrade columns that Lending Club attributes to the loans. What are the unique possible grades and subgrade? Create a countplot per grade.

TASK 1.8: Display a count plot per subgrade. Explore all loans made per subgrade as well being separated based on the loan_status.

TASK 1.9: Based on the above tasks, are there any subgrades which don't get paid back that often. Isolate those and recreate the countplot just for those subgrades.

TASK 1.10: Create a new column called 'loan_repaid' which will contain a 1 if the loan status was "Fully Paid" and a 0 if it was "Charged Off".

Bonus Task: Create a bar plot showing the correlation of the numeric features to the new loan_repaid column.

Task 2: Data Pre-Processing

Goal: Remove or fill any missing data. Remove unnecessary or repetitive features. Convert categorical string features to dummy variables.

TASK 2.1: Display the total count of missing values per column and also show the same in terms of percentage.

TASK 2.2 : Examine emp_title and emp_length to see whether it will be okay to drop them.

TASK 2.3: Create a count plot of the emp_length feature column. Sort the order of the values.

TASK 2.4 : Plot out the countplot separating Fully Paid vs Charged Off

TASK 2.5 : What is the percentage of charge offs per category. What percent of people per employment category didn't pay back their loan? Once you've created it, visualize it with a bar plot.

TASK 2.6: Is there any kind of relation between Charge off rates and employment lengths? Are there any chances of dropping something based on this analysis?

TASK 2.7: Is there any kind of relation between title column vs the purpose column**.** Are there any chances of dropping something based on this analysis?

TASK 2.8 : Is there any kind of relation between title column vs the purpose column.

TASK 2.9 : Analyse mort_acc feature and create a value_counts of the mort_acc column. Fill the missing values if any using appropriate approach along with proper reasoning. Read next task before attempting this task

(Hint: There are many ways we could deal with this missing data. We could attempt to build a simple model to fill it in, such as a linear model, we could just fill it in based on the mean of the other columns, or you could even bin the columns into categories and then set NaN as its own category.)

TASK 2.10: How mort_acc is related to other fields? Can we use the other related fields to fill the missing value of this column?

TASK 2.11: Are there any missing values in revol_util and the pub_rec_bankruptcies? If yes, then what is the appropriate approach to deal with this issue. Please do a practical implementation of your solution.

TASK 2.12 : List all the columns that are currently non-numeric.

TASK 2.13: Convert the term feature into either a 36 or 60 integer numeric data type

[hint: Use .apply() or .map()]

TASK 2.14: Can we drop grade column? If no, then what are the columns which are correlated with this column? Will it result into loss of information?

Before doing the next tasks, understand what are dummy variables using following links and how to use them. It is very similar to one hot encoding method

1. https://www.geeksforgeeks.org/python-pandas-get_dummies-method/

2. https://www.marsja.se/how-to-use-pandas-get_dummies-to-create-dummy-variables-in-python/]

TASK 2.15: Convert the subgrade into dummy variables. Then concatenate these new columns to the original dataframe. Remember to drop the original subgrade column and to add drop_first=True to your get_dummies call.

TASK 2.16 : Convert these columns: ['verification_status', 'application_type','initial_list_status','purpose'] into dummy variables and concatenate them with the original dataframe. (Hint: Use drop_first=True and to drop the original columns.)

TASK 2.17: Display the count for the various possible values of home_ownership column.

TASK 2.18: Convert these to dummy variables, but replace NONE and ANY with OTHER, so that we end up with just 4 categories, MORTGAGE, RENT, OWN, OTHER. Then concatenate them with the original dataframe. Set drop_first=True and to drop the original columns.

TASK 2.19: Create a column called 'zip_code' that extracts the zip code from the address column.

TASK 2.20: Now make this zip_code column into dummy variables. Concatenate the result and drop the original zip_code column along with dropping the address column.

TASK 2.21: Analyze "**earliest_cr_line".** Extract the year from this feature then convert it to a numeric feature. Set this new data to a feature column called 'earliest_cr_year'.Then drop the earliest_cr_line feature**.**

TASK 3: Train Test Split

TASK 3.1: Set X and y variables to the values of the features and label.

TASK 4: Grabbing a Sample for Training Time

TASK 4.1 : Perform a train/test split with test_size=0.2 and a random_state of 101.

TASK 4.2: Use a MinMaxScaler to normalize the feature data X_train and X_test.

TASK 5: **Creating the Model**

TASK 5.1: Build a sequential model which will be trained on the data. OPTIONAL: You can add Dropout layers in order to avoid the overfitting problem.

TASK 5.2 : Fit the model to the training data for **at least** 25 epochs and batch_size of 256.

TASK 5.3: Save your model.

TASK 6: Evaluating Model Performance

TASK 6.1: Plot out the validation loss versus the training loss.

TASK 6.2: Create predictions from the X_test set and display a classification report and confusion matrix for the X_test set.

TASK 6.3 : Select any random customer from the dataset, would you offer this person a loan?

Hint: For random selection of the customer for testing, you can use the following code:

```
import random

random.seed(101)

random_ind = random.randint(0,len(df))

new_customer = df.drop('loan_repaid',axis=1).iloc[random_ind]

new_customer
model.predict_classes(new_customer.values.reshape(1,78))
```

TASK 6.4 : Now check, did this person actually end up paying back their loan?