Third Capstone Project

Bank Churning Prediction Model

By: Omar Alnimasi

Problem Identification:

The dataset I am working here is for credit card customers with the goal to predict churning. A bank manager is attempting to check for churners to retain them beforehand by using the data provided for the card they are using, their salary group, age, gender, credit limit, total revolving balance, and more features.

I am working on a dataset with 20 features and attempting to build a model with high recall since the objective is to predict who is going to churn. We are aiming to have a high recall, ideally over 85% to provide the valuable information to the bank manager so they can act accordingly.

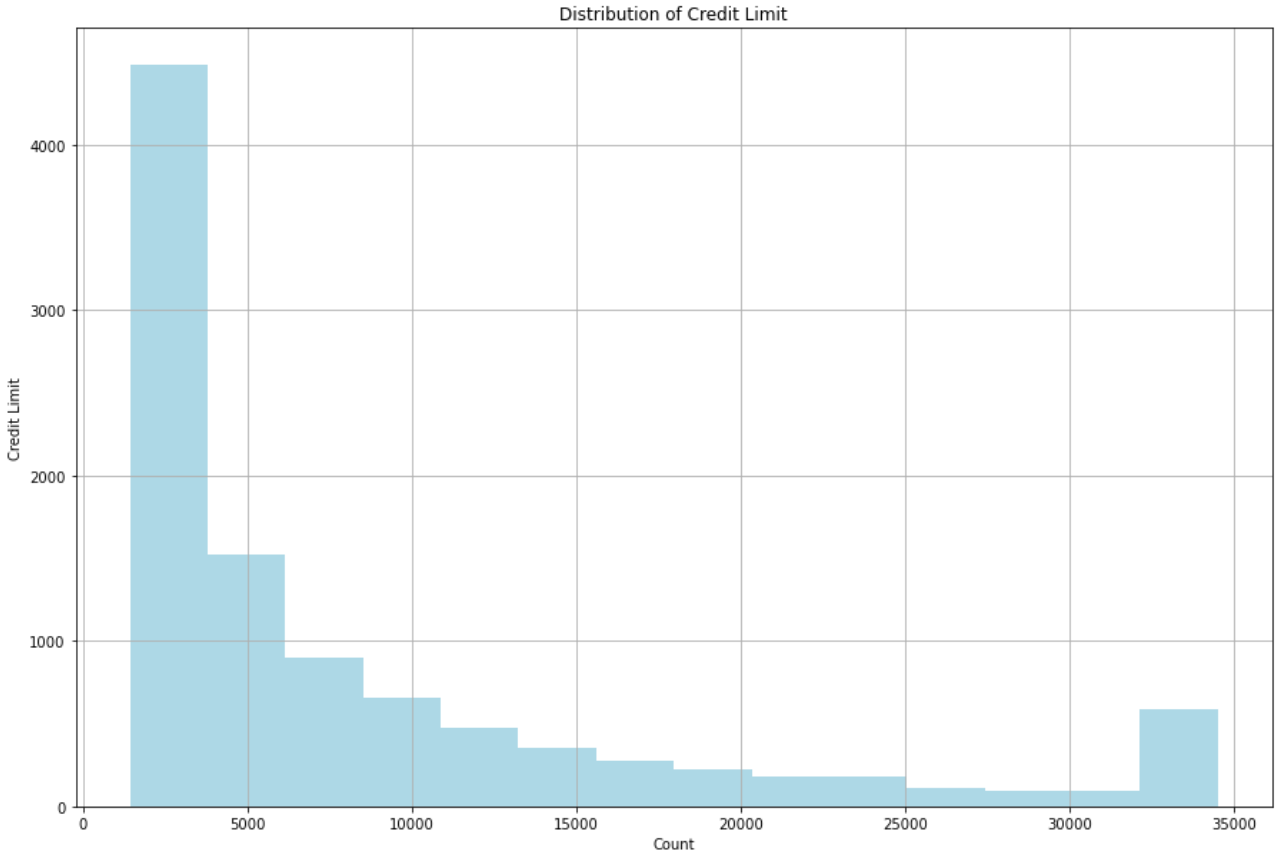
I will be attempting to use both machine learning and deep learning to achieve the best possible model. The data consists of many observations in categorical features that are marked as 'Unknown', which is information that could be valuable, but is missing. Furthermore, the dataset has the income as a categorical feature, and is divided into brackets instead of numerical values.

The main stakeholder here is the bank manager of the credit card portfolio, and the dataset was found in kaggle in the following link:
https://www.kaggle.com/datasets/sakshigoyal7/credit-card-customers

Data Wrangling:

In this dataset, there aren't any null values, however, we have a few columns with "Unknown" values as stated earlier, namely, Education_Level, Marital_Status, and Income_Category. Since we do not have a way to deal with these, we will have to leave them as they are.

Furthermore, There is an usual behavior in the distribution of Credit_Limit, as shown in the following figure:



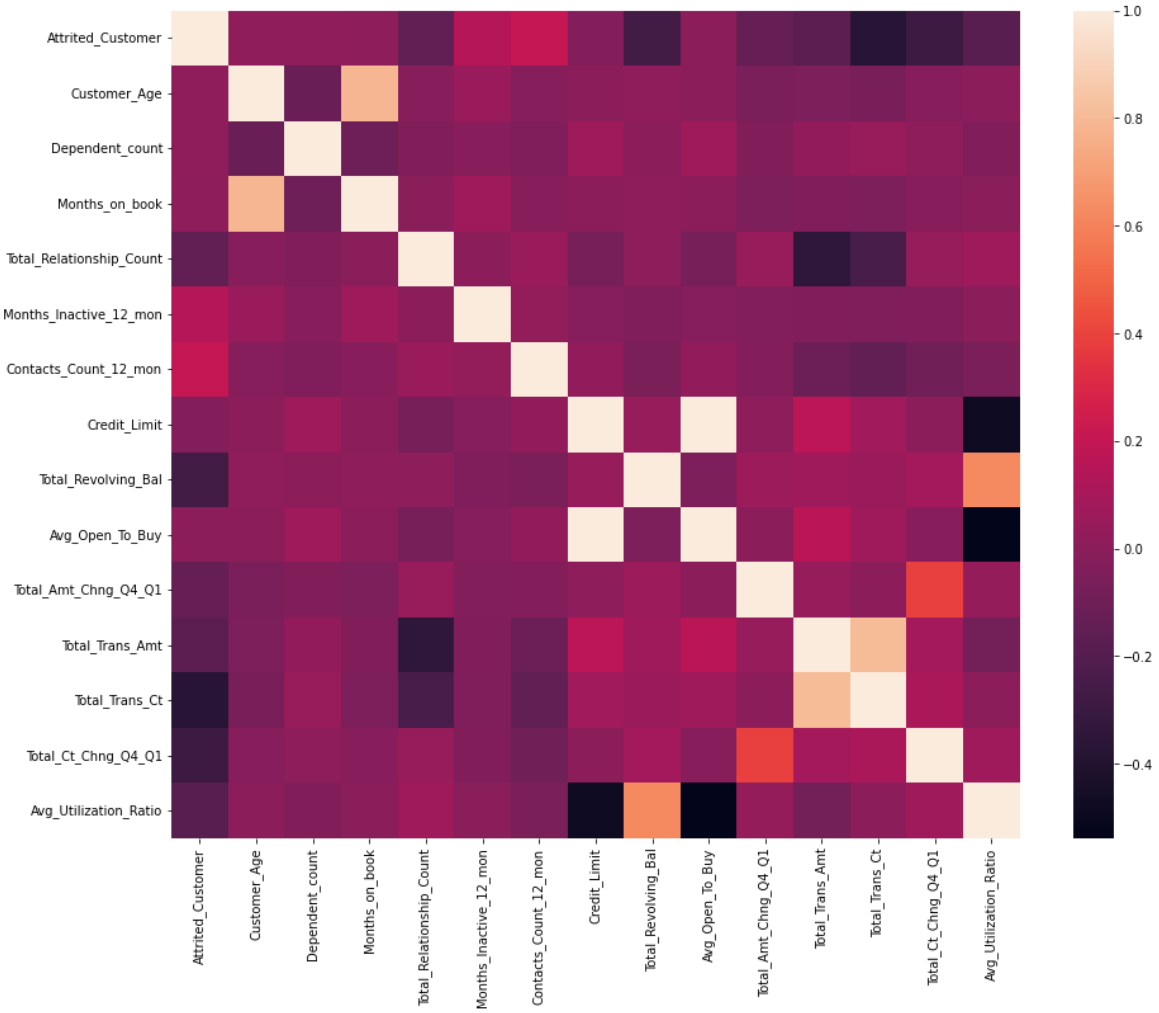
As it can be seen above, the figure is skewed to the left, however there is a peak at the rightmost bin which is unusual. After digging into the count of values, we noticed that in the credit limit column, there are two values that have over 500 occurrences, and every other value has less than 20 occurrences.

We concluded that these two values are likely to be the maximum and minimum possible values, which could explain why they are overcrowded. We also checked for any duplicated observations to make sure that these aren't just repeated observations and there aren't.

Finally, we changed our target feature to a numerical one by choosing attired customer as 1, and existing customer as 0.

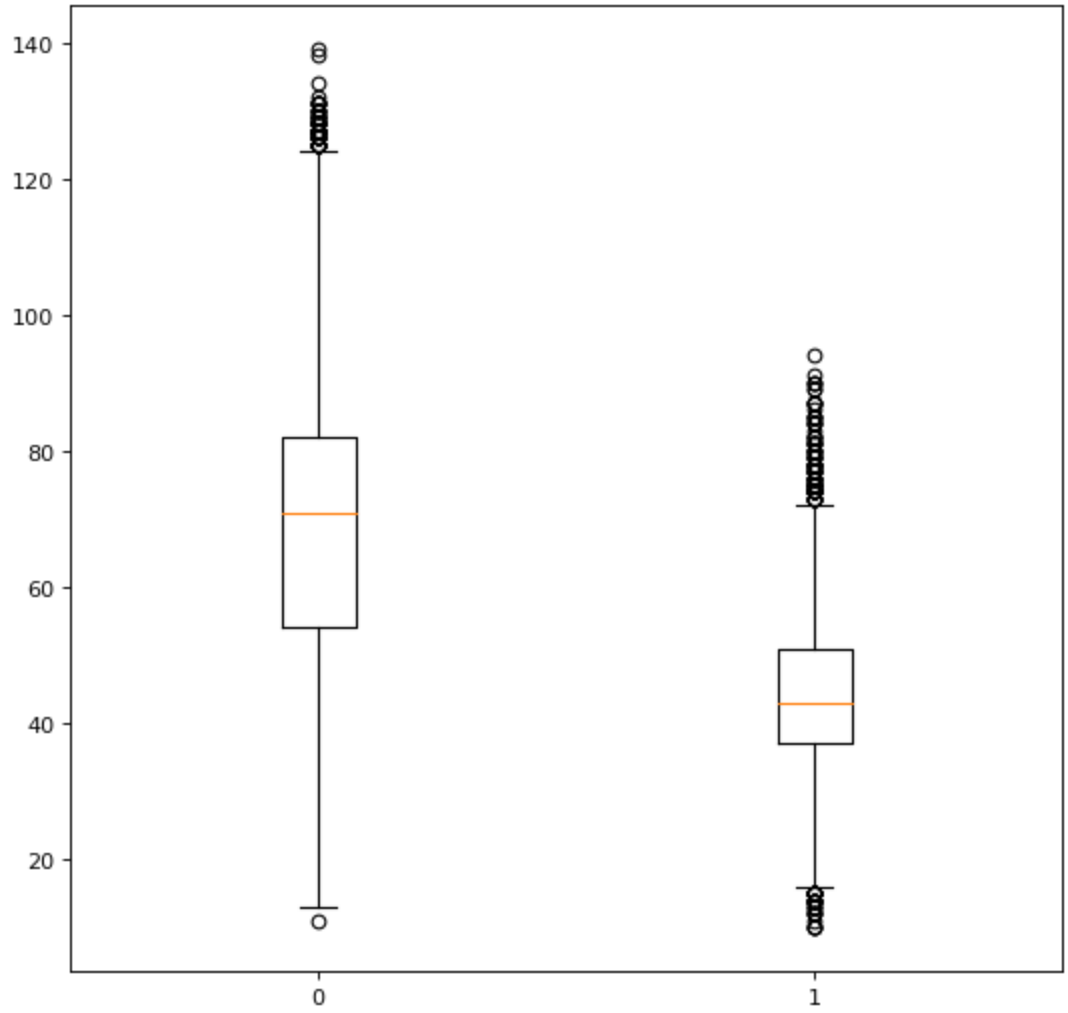
Data Exploratory Analysis:

Now that we have cleaned our data, we can explore it to find relations with our target variable. The following heatmap shows us some relations between our numerical features and could give us some information about our target as well.

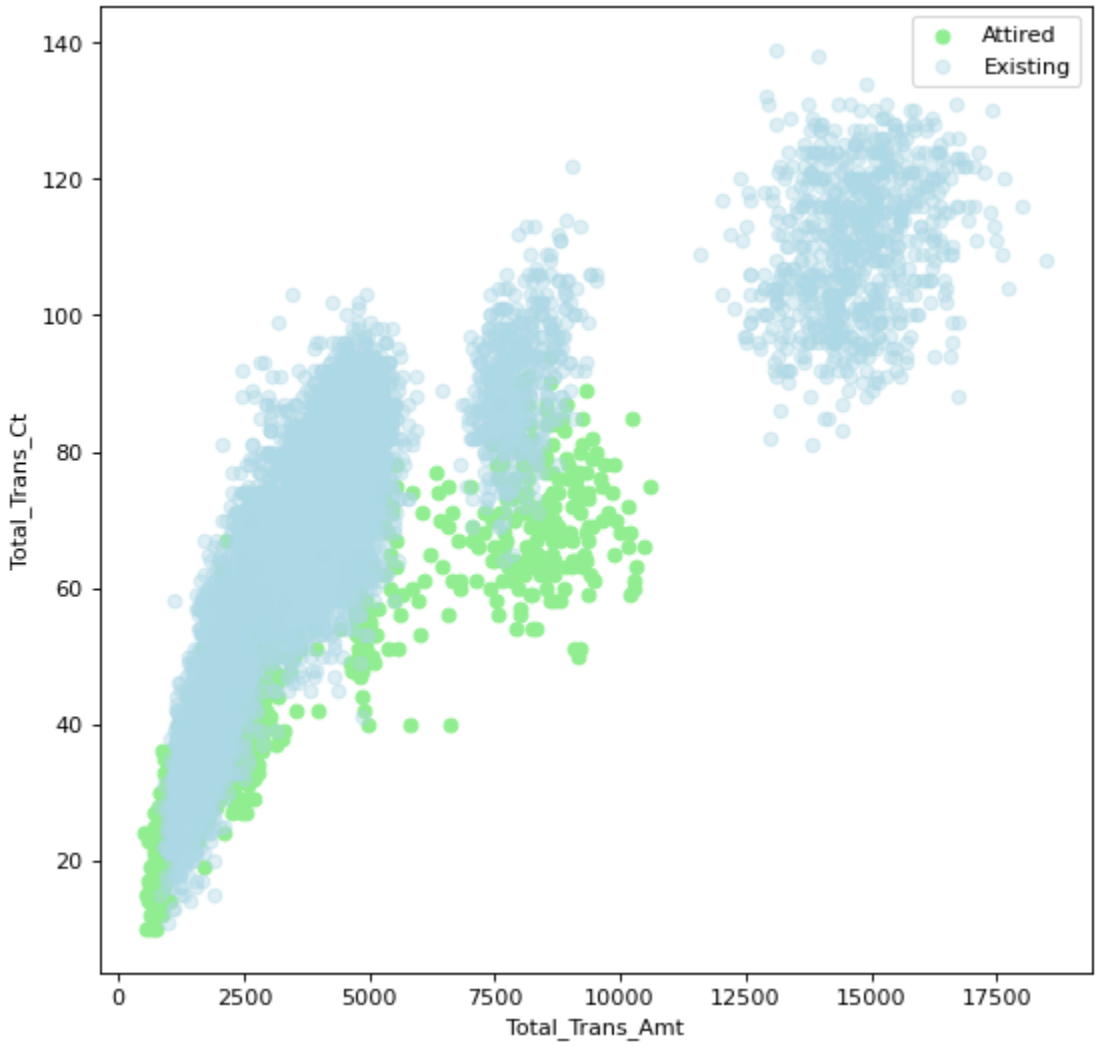


Here we have the highest correlation between our target with Months_Inactive_12_mon and Contacts_Count_12_mon, and the lowest with Total_Trans_Ct. We can also notice a high correlation between Credit_Limit and Avg_Open_To_Buy.

I then explored the distributions of Total_Trans_Ct, and as expected, the existing customers Total_Trans_Ct average is much higher than that of the attired customer.



Finally I explored the correlation of Total_Trans_Amt, and Total_Trans_Ct and the distribution of existing vs attired customers which gave us insights that the higher the total transactions the more likely the customer is existing as shown in the following figure.



Pre-Processing:

After we cleaned the data, and understood the relationships between our features, now we need to prepare it for modeling. Here, I started by separating my categorical and numerical features. I started with using the standard scaler from the sklearn library, which I used to standardize my numerical columns to having a mean of zero, and standard deviation of one.

Then, I used one hot encoding to change the categorical columns to multiple numerical columns, each corresponding to a specific unique value within that column. This was done by using get_dummies from pandas. Finally, we used the train_test_split function

from sklearn to separate our dataset so that we have a testing set to check for the performance of our models.

Modeling:

Using the train and test sets, I explored a few models to check for the best performing one. I started with a support vector machine model, which was not a high performing one, offering a recall of 67.5%. Then after optimizing some of the hyperparameters, I got the recall up to 75%.

Next I started with a decision tree, and achieved a 78% recall, which was the same result even after optimizing the hyperparameters of that model. After that, I tested out a random forest classifier, which resulted in a 74.5% recall. However, after optimizing the hyperparameters, I achieved a 78.5% recall to the model, which is slightly better than the decision tree model.

The Final machine learning model I tried was the Gaussian naive bayes, which resulted in a 57% and a 52% with and without hyperparameter optimization. Finally I improved the best model which was the random forest classifier by choosing the best number of features, which after searching through all possible numbers of features turned out to be 16. So that resulted in a recall of 84%.

After checking on potential machine learning models. I moved on to deep learning. I used a sequential model for this. I started checking for different hyperparameters, and different numbers of hidden layers and hidden nodes. After checking multiple models with different parameters and numbers of hidden layers, I achieved a model with 99.9% recall on my training set, and 88.6% recall on my test set, this has given us a much better model. There is however some level of overfitting which resulted in having a much higher recall in the training test as opposed to the test set, but this was not something any of the other deep learning models could have addressed, and since I have the best recall score of any model, this means this is the ideal model for this case as it predict about nine out of ten churning customers.

Conclusion:

After going through the data science pipeline, I cleaned the data while checking for outliers. Then I explored the data to have a better understanding of it. After that, I prepared the data for modeling in the pre-processing step, and finally I modeled the data using multiple options both in machine learning and deep learning, achieving a model that has a recall that is 88.6%. As the goal of the business is detecting the attired customers, this is the best possible model that the business can use.