Treeleaf Technologies Nepal

Bank Loan Classification using Logistic Regression and KNN

By

Aseem Chemjong

Aug, 2023

1. Approach

First, I looked at the data in an Excel file to see what it's like. Then, I used a tool called Google Colab for this project. In Colab, I got some helpful tools ready for working with the data, like numbers and tables. I brought in the data from the Excel file using another tool called pandas. I checked to see if the main thing I want to predict, which is whether someone will accept the personal loan or not, was spread out evenly or not. It turned out it wasn't, so I looked at a graph to see that more clearly. Then, I looked at a couple of other things in the data like 'Gender' and 'Home Ownership', but they had lots of missing information, so I decided not to drop them. I checked for other columns with NaN values from the dataframe and I found columns ‘Income’ and ‘Online’ had 67 and 40 NaN values respectively. Compared to 5000 rows 67 plus 40 which is just 107 rows were insignificant so I dropped the rows which contained the NaN value in ‘Income’ and ‘Online’ class.

After that, I checked if different things in the data were connected to each other, like if one thing going up meant another thing went up. I made a heatmap of correlation between target class and each of the other features to help me see that. I also performed another visualization, that is scatterplot between Target class and every other feature. Further I also used the box plot visualization technique. The box plot along with the scatter plot helped me discover the outliers in the data. I dropped the ‘ID’ column as it doesn't possess any significance in our model building.

Then, I wanted to make a machine learning model to guess if someone will accept a personal loan or not. I used two different algorithms, one called Logistic Regression and the other called KNN. To help me with this, I imported the necessary libraries like train_test_split, StandardScaler, SMOTE, metrics, KNeighborsClassifier and LogisticRegression. After that, I divided the features and target into X and y. Then I splitted the X, y into the X_train , X_test , y_train , y_test while maintaining the test_size of ratio 0.25. I scaled the X_train and X_test using the StandardScaler. I then applied the SMOTE technique to balance the classes in the training data.

Furthermore, I built the model using Logistic Regression. For this, I created the instance of LogisticRegression Class which is the machine learning model. Then I fitted that model with our resampled X_train and y_train data. I did the same with the KNN model as well. The only difference being I tried all the values of k i.e. neighbor value iteratively."

2. Key Findings

One of the first key findings was that the Personal Loan class had 1 space string as its values. Technically, Personal Loan should have only 2 unique values, 0 and 1. But because of this space string value there were 3, which obviously needed to be removed. This space string had also made the Personal Loan column an object datatype which I converted into integer later.

Another key finding was that column ‘Experience’ also had some negative values. That didn’t make any sense. At max. people have -3 (for negative experience) as the experience with all those people being less than 30 years of year. So, there must be data entry mistakes and there should be positive values in the 'Experience'. I then converted all the negative values into the positive using the abs() function.

I also discovered that there were data entries in the age column of values more than 500. Although there were not many, these unusually high values can hamper our model prediction so I removed these unusually high values. There were also outliers in other columns as well, I also removed them taking the help of scatter plot and box plot.

3. Observation form the analysis

I carried out the analysis of the both Logistic Regression and KNN model. First let's talk about the Logistic Regression model. The accuracy of my Logistic regression was 89.8%. I also calculated the confusion matrix using sklearn.metrics. The model did a good job with prediction which has ‘0’ as target value but couldn’t do much better which has ‘1’the target value. This may be the result of imbalanced target class values. Although I performed the SMOTE technique to generate the synthetic data it seems it hasn’t worked that well. If I had more time I would have experimented with the other techniques like oversampling, undersampling etc. I would have also worked more on EDA to get more insight. I have used the .xlsx file which was given earlier which seems a bit dirty compared to the newer .xlsx file. But as I had already spent a considerable amount of time with older data, I had no time left to use the new data as well.

In the analysis of the KNN model I got an accuracy of 95%. But this accuracy again is misleading as this model also failed to correctly predict the data points with the target values as ‘1’.

From this what I conclude is that, I should use other better techniques to handle the imbalanced distribution of target class. Also check if the model is overfitted or not. And Finally build the machine learning model.

4. Message from my side

I've got my basics on machine learning very clear, right from the theories to the mathematical side. And can work extremely well under good supervision. I’ve performed quite well in my AI fellowship program of Fusemachines and have passed the Machine Learning Module with flying colors.