Default Prediction

16.01.2023

Trinanjan Saha

Overview

Given a dataset with different features, we need to predict the probability of default.

Project Structure

1. After analyzing the data, It is clear that we need to do default prediction from the financial data of the user.
2. The first step is to do **EDA** to figure out the different characteristics of the data columns.
3. Once the EDA is done, **Preprocessing** needs to happen before we can send the data to the model.
4. **Feature selection / Feature engineering** to figure out relevant and important features.
5. Finally, the **Model** is built with this data, and since the data is imbalanced , we need to choose an appropriate metric for **evaluation**.
6. Since the dataset size is not that big I have only experimented with machine learning models, not deep learning models
7. Please refer to the **Future Work** section at the end. It talks about the possible strategy to improve the model metrics.
8. **Also, I feel the default =1 class is more important than the default = 0. The initial draft of the model without any preprocessing and feature selection performs well on 0 class but not on 1 class. After doing the following steps the F1 score for the 1 class has improved from .3 to .6. Although the class 0 f1 score came down from .8 to .7**

1. EDA

After doing EDA on the data I had the following observations on the dataset

a. Not all the columns have a similar data type, some are numeric, and some are binary in nature, for example, columns v18, v34

b. Some of the columns have the same values more than 90 % of the time. We can ignore these columns because they don't contribute any significant information to predict default. For example v8,v20,v22,v26,v29,v40,v47
c. Some of the columns have very skewed data, we can do log transformation for them. For example columns v6, v10, etc.
d. There are a lot of missing values, we need to impute them(based on the choice of model)
e. log transformed columns will be imputed using mean, and other columns are imputed with median(because of skewness).
f. The final label column has imbalanced data, there are more numbers 0(no default) vs 1 (default). We need to use some minority sampling strategy.
g. After doing a bivariate analysis I found out that there are a lot of correlated columns having a correlation coefficient of more than .9. For Example **v2 vs v30.** We can remove these columns.
h. Also, some columns are the user's device information, they will not matter while doing default prediction.
i. Finally, I brought down the features/columns from 62 to 37. This helps to reduce the complexity of data and helps us to focus on more important features.

2. Preprocessing

According to the observations on the EDA, the following preprocessing was performed

a. First, we transform the skewed columns with log transformation
b. Then Impute the data according to skewness, if too much is skewed use the median otherwise mean.
c. I have also tried KNN imputation, but the results were similar.
d. Since label 1 has less number of examples compared to label 0, we need to do some oversampling on label 1 to make them an equal number to the majority class.
e. Also tried 0 centering the data, but the performance didn't improve.

3. Feature Engineering / Feature selection

a. I have only selected those features that were relevant according to the EDA, We can further reduce them by using some sort of dimensionality reduction technique like PCA (Future work).
b. Feature engineering is basically coming up with new features using the existing feature. This needs a lot of understanding of the domain.
c. To figure out which features are similar so that we can somehow combine them to create a new feature, I tried to group features using their name. I used a

transformer-based model to generate the embedding of all the column names and then group them using **KMEANS**.

d. Example avg. monthly credit transactions amount to last 180 days vs avg. monthly debit transactions amount the last 180 days could be used as a ratio to check the user spending habits.
e. This is one example of feature engineering, Since the f1 score didn't improve after doing these feature engineering I have not included them in the final notebook.

4. Model Training / Model Evaluation

a. For model training, I have tried out multiple models like Logistic Regression, Random forest, XGboost, etc.
b. Also tried out ensembling these models with voting. The results were similar.
c. The random forest model was giving the best results.
d. These models' hyperparameters were decided through grid search(randomize).
e. For evaluation, I was using weighted F1 score. We need to check individual classes' f1 scores. After doing the preprocessing and feature selection the class 1 probability improved from .3 to .6

5. Future Work and Improvement

a. We can further reduce them by using some sort of dimensionality reduction technique like PCA.
b. Feature engineering could improve the f1 score.
c. Need to check the feature importance, This will tell us what features are more important in terms of default prediction.