Mohana Krishna Narsupalli

HW2: Credit Risk Prediction

username:rook_in_defence

Rank: 72

Score: 0.68

Approach:

Data Cleaning:

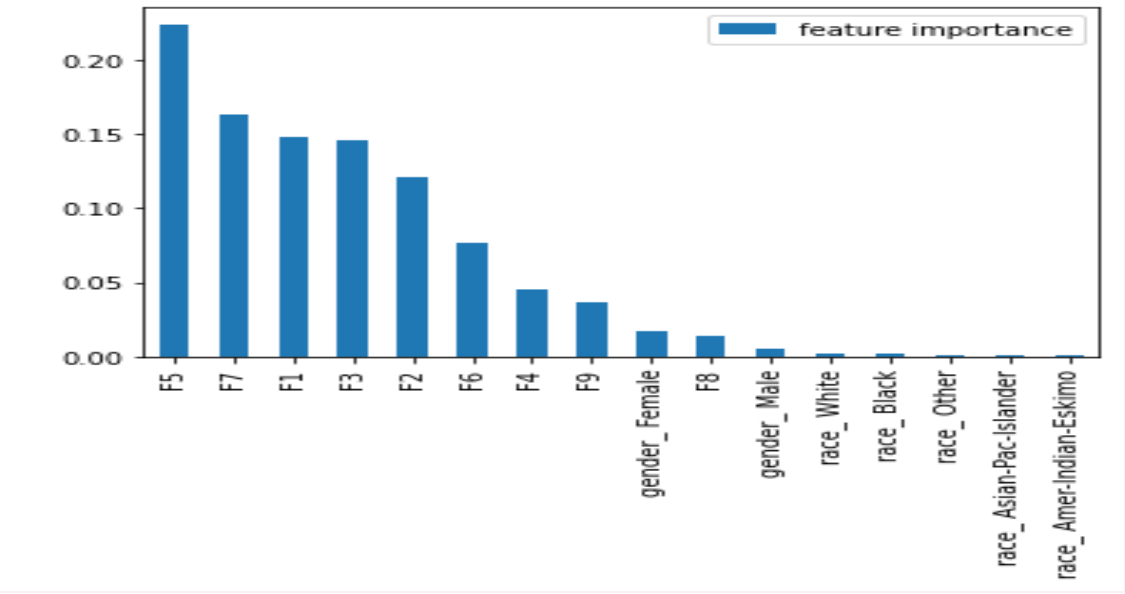
Looking at the training dataset and test dataset, there's high imbalance i.e 24720 0's and 7841 1's are present in the training dataset. I have tried to balance this dataset by shuffling the data. i.e by combining records with target variable as 1 is almost on third of records with target variable with 0. So I multiplied the records three times with target variable 1 and tried to balance the data. After that , I checked for null records if any value of any column is null. There are no null values and after that, I tried to check for test records.

Similar procedure is followed for test dataset. Checking for null records and any corrupt records with either missing fields or columns. Then I need to predict the target variable. Then I started to look at the features i.e feature selection. I checked the columns of the train dataset. Columns years since last degree was completed and hours worked per week Seem to be more appropriate and so gave importance for them.

Next column being relationship, I felt it as moderately affecting column and column occupation too the same. In contrast, continuous columns of Gains and Losses are critical to establish Credit Risk and so I felt them to be included in my model. I felt it column employment too is as moderately affecting column. Column employment type is bit of redundant column and am skeptical about it. Column education type too is a bit critical for credit risk evaluation and so I included it in train dataset.

But columns race and gender, am not sure so I want to play with those columns if it affects much. So those columns which are moderately affecting are to be played aroundwith. I plotted Feature importance and so want to check feature importances.

Based on the feature importance plot, I dropped the Categorical columns and proceeded with other columns. Below is the feature importance plot done initially.



In this, the last columns gender and race are done with one-hot encoding asone-hot encoding is better than just labelling. So when checked with both cases, One-hot encoding is giving better results, so I used it for both categorical column values.In this the value of current value is given 1 whereas others become 0. Same logic is applied for all records. Similarly, test dataset is also done with same procedure.But with feature impotance plot, they aren't to be given much weight so I dropped these columns.

Now, I wanted to play with different permutations of columns with different models. So, I started with SVM(Support Vector Machine) and tried with kernel=rbf. It did not give much accuracy and it's around 81% accurate. Next,I tried with Neural Networks. In this,I used Sequential model. I built Sequential Layer, then added Dense Layer with 9 units, again added Dense Layer with 9 units, and a Layer with 5 units and finally last layer with 1 unit.I used sigmoid function in the last layer to decide the output. Optimizers used is adam with loss = 'binary_crossentropy'. Then I fitted with the Layers to Artificial Neural Network and ran it for 200 epochs.

The above resulted inaccuracy around 83 and I tried with other parameters.It did not give much better results and I thought to try other models.I tried with RandomForestClassifier and it's accuracy is also around the same. The accuracy is around 84% and from then I wanted to try other models too.

I also tried xgbclassifier with following parameters:
XGBClassifier(learning_rate=0.5, max_depth=10, min_child_weight=6, n_estimators=10

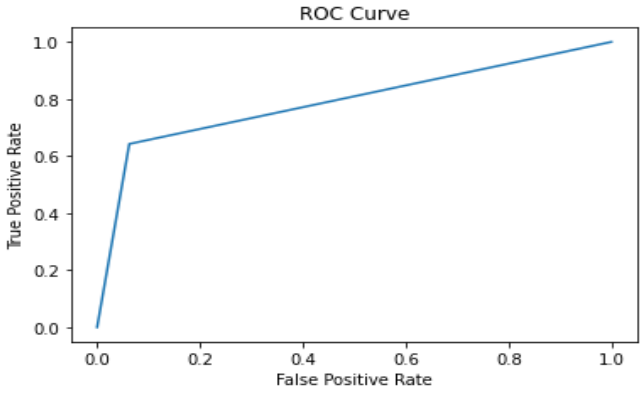
0, nthread=1, subsample=0.9500000000000001). It gave a bit high accuracy around 85 and I want to continue further.

So, I again tried with GradientBoostingClassifier without base parameters. It boosted the accuracy. So I want to continue with this model and started Hyper parameter tuning. I got best result when used parametrs:

GradientBoostingClassifier(learning_rate=0.5,max_depth=5,max_features=0.3,min_samples_leaf=25,min_samples_split=4,n_estimators=100,subsample=1.0)

Score Table

Model	Accuracy	F1-Score
SVM(rbf kernel)	78	63
Neural Network	83	65
XGBClassifier	85	67
GBMClassifier	85	69



The above shows various results for different models. ROC Curve corresponds to GBMClassifier with above parameters. It showed auc value around 78 and that's best result so far I have got. Boosting is a method of converting weak learners into strong learners. In boosting, each new tree is a fit on a modified version of the original data set. GBM constructs a forward stage-wise additive model by implementing gradient descent in function space. Gradient Boosting trains many models in a gradual, additive and sequential manner.

References:

Sklearn documentation: https://scikit-learn.org/stable/
Numpy and Pandas Libraries and Functions