Credit Risk Analysis using Machine Learning

Once someone submit a loan application (applicant) to a bank or any other financial institution (lender), it is important that the lender carefully assess if the applicant will or have the capability to pay back the loan. This should be done vigilantly as there are two types of risks involved here. First is that, if the applicant is likely to repay the loan, but the lender decide not to approve the loan, lender will lose a business. On the other hand if the applicant is not likely to pay the loan, but the lender approves the loan, then it will be a financial loss for the lender. Because of this, modeling credit risk has become a crucial study and researchers have been using machine learning techniques to model the credit risk so that the loan managers can make data driven decisions on the massive number of loan applications they receive frequently.

With this project, I make an attempt to understand the most features in regard to the credit risk. Further, I will develop a Machine Learning model that predicts the likelihood of good credit risk.

Data

I use German Credit Risk data set prepared by Dr. Hofmann, which is available at UCI Machine Learning Repository. The dataset has 1000 records and each record is associated with a unique person. This has 22 variables including the target variable "Risk". Seven of the variables are numerical, while all the rest are categorical. Also, those categorical variables were coded differently. So, I used the data dictionary presented with the original dataset and manually encoded them into meaningful categories.

I decided to remove some of the variables in the dataset, as I believe they will not be useful to be further analyzed. The ones I removed were, 'Personal Status", "Property", "Telephone", and "foreign worker".

Data type assigned for each variable appeared good and there were no missing values in the data set. However, I renamed few of the variables to make them more clear and readable.

Data Exploration

After the removal of certain variables, dataset now have 18 variables with 1000 records. It was noticeable that we have an imbalanced dataset here as 70% of the records were classified as good risk while only 30% were classified as bad risk.

It can be observed that for applicants with bad risk have a higher duration in average than that of good risk applicants. Also, mean credit amount of bad risk applicants and the mean installment rate are considerably higher than those with the applicants identified as good risk.

On the other hand, mean number of existing credits in the bank is higher for the applicants labeled as good risk.

However, since we have two imbalanced groups, comparing average values might not be a wise thing to do. But they could be useful in understanding data elements.

Table 1: Data Summary between Good and Bad Risk

Numerical Variables	Risk	
	Bad	**Good**
Duration	24.86	19.21
Credit Amount	3938.13	2985.46
Installment Rate	3.1	2.92
Resident Duration	2.85	2.84
Age	33.96	36.22
Number of Existing Credits in the bank	1.37	1.42
Dependents	1.15	1.16

Duration, Age, and Credit amount all seem to be right-skewed distributions, meaning most of the observations have less duration, small credit amounts, and young applicants. This seem to be the situation with both good and bad risk groups. Also, it can be noticed that, credit amount and duration (in month) is positive correlated.

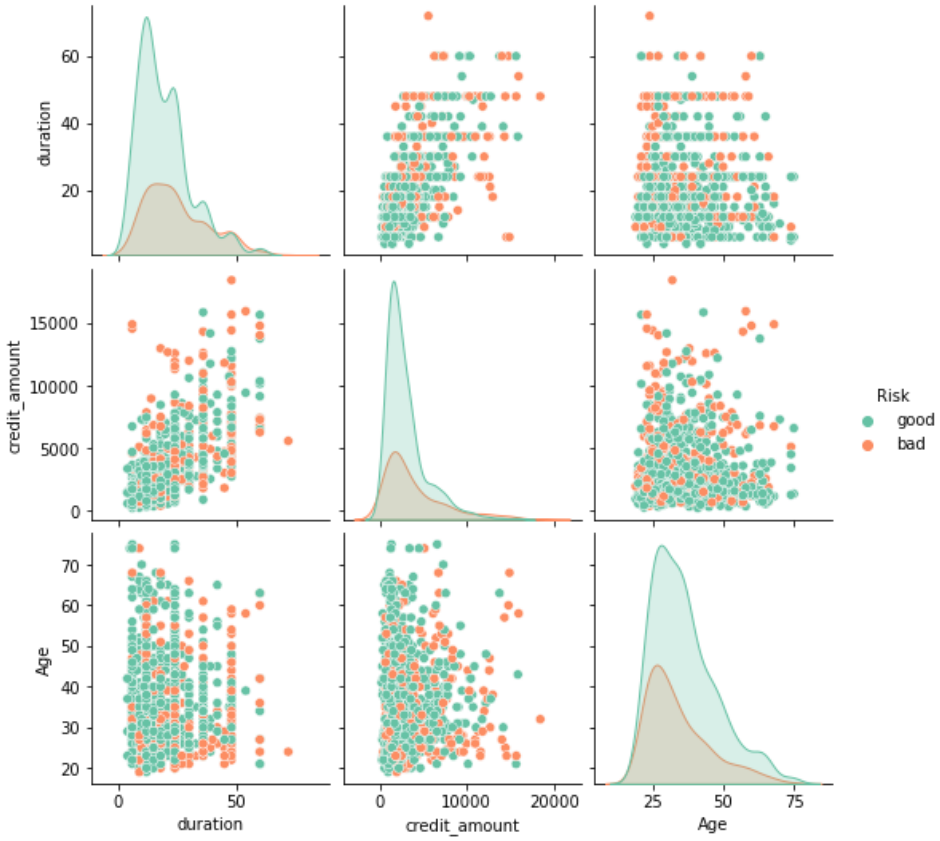

Figure 1: Distributions of Age, Credit Amount, and Duration

Of those with bad risk, only 15% does not have a checking account and with good risk applicants, 49% does not have checking account. With the status of having no checking account, there is a substantial difference in the number of records between good and bad risk applicants.

72% of bad risk applicants have had savings account/bonds with <100 DM (DM is the German Currency), and 45% of them have had checking accounts with <0 DM.

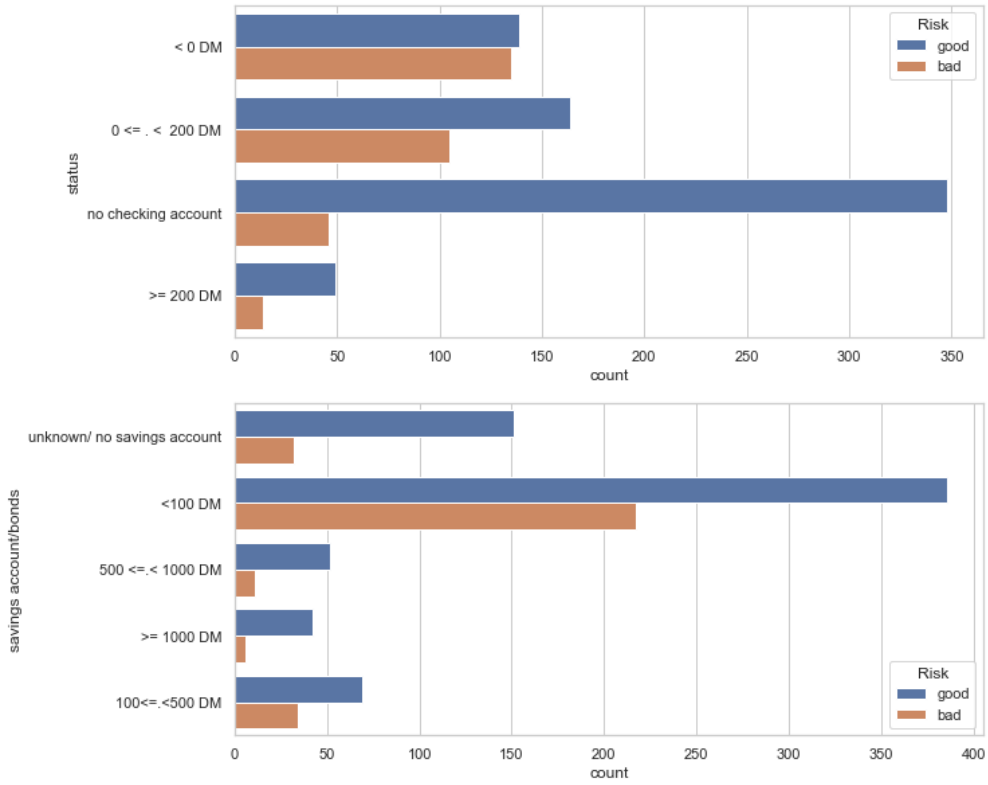

Figure 2: Checking Accounts and Savings Accounts/Bonds

When loan duration is close to 10 months majority of the applicants have a good risk, and with 20 month loan duration it is the other way around. And once the loan duration go beyond about 30 months, it seems most of the applicants have been identified as bad risk applicants.

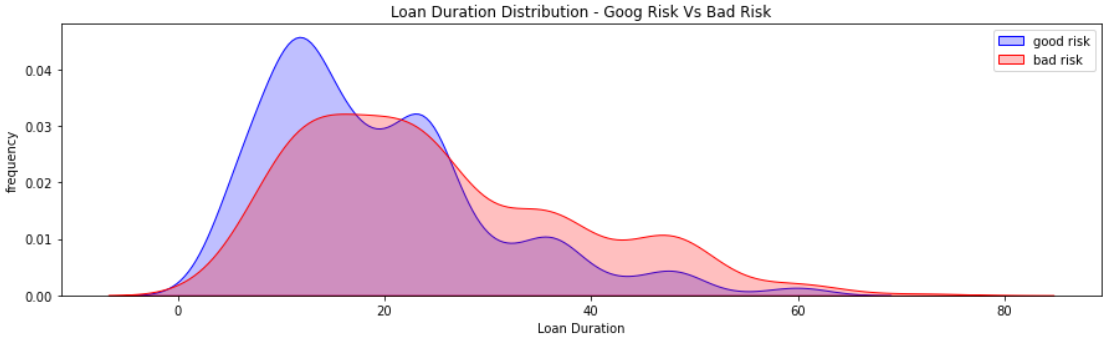

Figure 3: Loan Duration Distributions

Data Preprocessing and Modeling

We need to convert categorical variables into numerical variables because most of the machine learning models require the input data to be numerical. Since “status”, “savings account/bonds”, and “present employment” variables are ordinal, I manually encoded them because ordinal encoding was not assigning the labels in a way that demonstrates the order between the categories of these variables. One thing I could try here is target encoding. Or if I use CatBoost or LightGBM algorithms, this step would not be necessary as they can deal with categorical variables when fit the model (These remarks will be used in future to improve the performance of the model and predictions).

For the rest of the categorical variables, one hot encoding was used. As a result, we now have 31 features in our dataset.

As previously mentioned, 70% of the loan applications are good risk, while 30% are bad.

There are many approaches to handle imbalance data. Few popular ones among them are,

- **Random Under-Sampling**: randomly eliminates majority class instances
- **Random Over-Sampling**: increases the number of instances in the minority class by randomly replicating them
- **Cluster-Based Over Sampling**: K-means clustering algorithm is independently applied to minority and majority class instances. Each cluster is oversampled such that all classes have the same size.
- **Synthetic Minority Oversampling Algorithm (SMOTE)**: A subset of data from minority class is taken and then new synthetic similar instances are added to the original dataset.
- **Modified synthetic minority oversampling technique (MSMOTE)**

I will focus on only random under-sampling, random over-sampling, and SMOTE here.

Next step was to split the dataset into train and test dataset. 70/30 ratio was used for this process. And then resampled train datasets with above-mentioned resampling techniques. Standardized the features, and used each resampled dataset separately to fit the Logistic Regression model. Then selected the best resampled data set based on the F1 score values, and original train/test dataset gave the best F1 score.

Logistic regression was used without scaling the features with original train/test data set and the AUC was much better with this way. So, I followed the same steps when modeling with other models.

Other models I used are,

1. Random Forest Classifier
2. Gradient Boosting Classifier
3. CatBoost Algorithm (Did not encode categorical variables into numerical. Also, none of the scaling techniques were not used as well for this algorithm)

Model Evaluation

Evaluation metrics which can be used select the best classification model are,

1. Accuracy Score
2. Precision
3. Recall
4. F1 Score
5. ROC-AUC

Best metrics among these for classification of imbalance dataset are known as F1 Score, ROC curve, and the AUC value. Comparing the AUC values, we can conclude that the best model, among the models we used here, is the CatBoost algorithm.

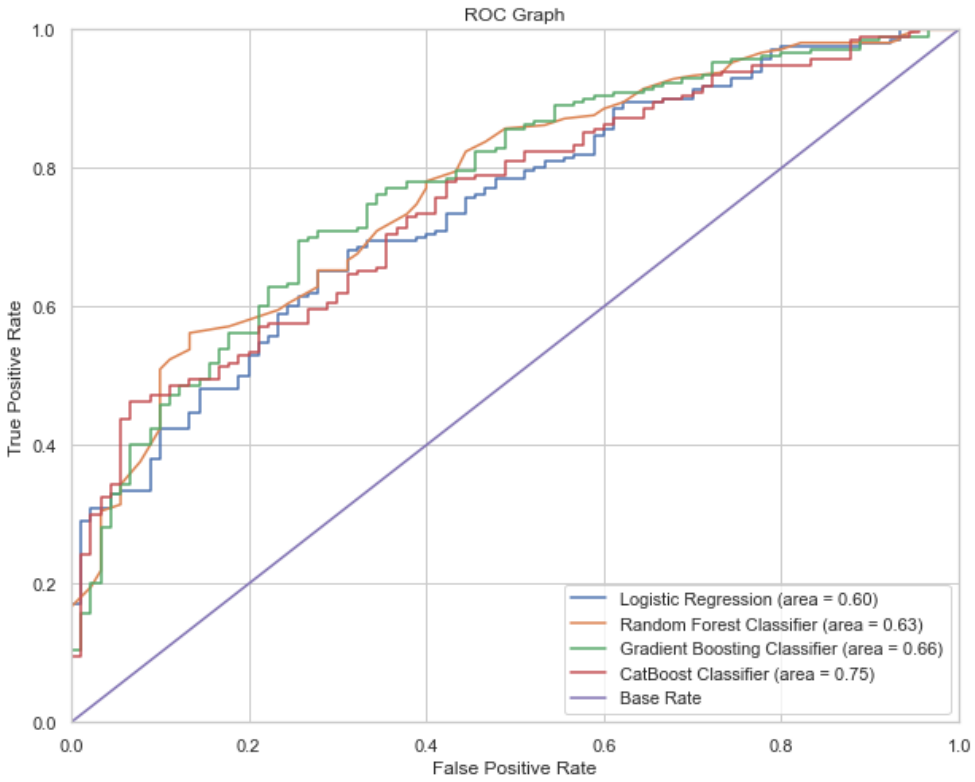

Figure 4: ROC curves of all models

Future Work

- Use Bayesian Hyper Parameter Optimization improve the performance of the models.
- Come up with a strategy to use this information effectively.