Company Bankruptcy Prediction

Group 59

Jiangying Zhong

Jingxuan Li

781-518- 1626

617-755-5192

zhong.ji@northeastern.edu

li.jingxuan1@northeastern.edu

Percentage of Effort Contributed by Student1 : 50%

Percentage of Effort Contributed by Student2 : 50%

Signature of Student 1 : Jiangying Zhong

Signature of Student 2 : Jingxuan Li

Submission Date : 4 / 2 4 /2022

Problem Setting:

Profitability is the most important thing in the development of an enterprise. Whether a company is profitable depends on many factors, and every change in detail will affect the final income. Therefore, analyzing the bankruptcy probability of a company through its various data is crucial. Through this analysis, enterprises can see which important indicators will seriously affect the development, so as to change their development strategies in time. This analysis is conducive to adjusting the development direction of enterprises and avoiding risk of bankruptcy. However, the data types of different industries are quite different and unable to be unified. The same analysis method can only be applied to similar industries, so the applicability rate is low and likely to cause different degrees of result error.

Problem Definition:

The main goal of this project is to use several machine learning techniques to identify whether the company with different accounting indicators has bankrupted(classification problem) and compare four factors: accuracy, precision, AUC and F1-score with these techniques to find the best one. We would complete the project in the following aspects:

1. Classify the features that lead to Bankruptcy for companies.
2. Figure out the correlations across categories provided.
3. Apply The Synthetic Minority Over-sampling Technique(SMOTE) to deal with the data imbalance.
4. Implement several machine learning methods such as KNN, Logistic Regression, Random Forest, SVM, Decision Tree, Naive Bayes and XGBoost to create models and then test the accuracy, precision, AUC, F1-score, confusion matrix, etc. to find the most suitable one to do the prediction.

Data Source:

https://www.kaggle.com/fedesoriano/company-bankruptcy-prediction

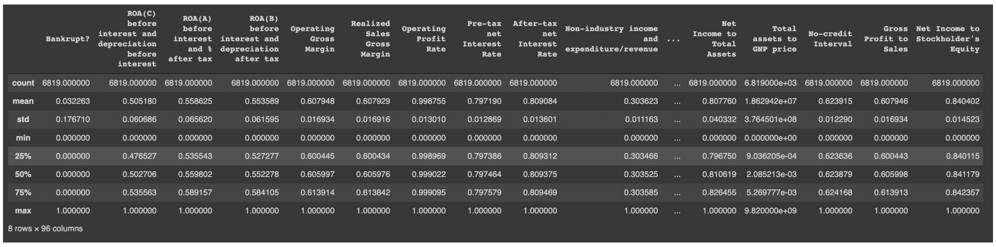
Data Description:

The data set only contains one csv file called data.csv, including several attributes of companies. The data size is approximately 11.46MB, pretty large enough to do a reasonable prediction. There are 6819 rows and 96 columns, 95 of them are the input features while one is the output feature. Data attributes cover information regarding the company’s various indicators and background, such as: return-on total assets, realized gross profit/net sales, cash flow rate, tax rate, Cash/Current Liability and inventory turnover rate etc. The dataset has 93 decimals, 2 integers and one boolean. The boolean indicates the output feature (whether the company is bankrupt or not) : “0” for bankrupt while “1” for not.

<u>Data Exploration:</u>

1. <u>Exploratory Data Analysis:</u>

Firstly we check the shape of data, which is the dimensions of a matrix or array. It has 6819 instances and 96 attributes. Then we check statistical variables of data frame to observe the range, size, fluctuation trend, etc. Of this series of data.



	Bankrupt?	ROA(C) before interest and depreciation before interest	ROA(A) before interest and % after tax	ROA(B) before interest and depreciation after tax	Operating Gross Margin	Realized Sales Gross Margin	Operating Profit Rate	Pre-tax net Interest Rate	After-tax net Interest Rate	Non-industry income and expenditure/revenue	...	Net Income to Total Assets	Total assets to GNP price	No-credit Interval	Gross Profit to Sales	Net Income to Stockholder's Equity
count	6819.000000	6819.000000	6819.000000	6819.000000	6819.000000	6819.000000	6819.000000	6819.000000	6819.000000	6819.000000	...	6819.000000	6.819000e+03	6819.000000	6819.000000	6819.000000
mean	0.032263	0.505180	0.558625	0.553589	0.607948	0.607929	0.998755	0.797190	0.809084	0.303623	...	0.807760	1.862942e+07	0.623915	0.607946	0.840402
std	0.176710	0.060686	0.065620	0.061595	0.016934	0.016916	0.013010	0.012869	0.013601	0.011163	...	0.040332	3.764501e+08	0.012290	0.016934	0.014523
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000e+00	0.000000	0.000000	0.000000
25%	0.000000	0.476527	0.535543	0.527277	0.600445	0.600434	0.998969	0.797386	0.809312	0.303466	...	0.796750	9.036205e-04	0.623636	0.600443	0.840115
50%	0.000000	0.502706	0.559802	0.552278	0.605997	0.605976	0.999022	0.797464	0.809375	0.303525	...	0.810619	2.085213e-03	0.623879	0.605998	0.841179
75%	0.000000	0.535563	0.589157	0.584105	0.613914	0.613842	0.999095	0.797579	0.809469	0.303585	...	0.826455	5.269777e-03	0.624168	0.613913	0.842357
max	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	...	1.000000	9.820000e+09	1.000000	1.000000	1.000000

8 rows × 96 columns

Fig1 Data Description

2. <u>Data Visualization:</u>

At the beginning of exploring the data, we discovered the distribution of bankrupt. Based on it, we calculate the amount of bankrupt companies and non-bankrupt companies separately in the data set, where 1 represents for bankrupt and 0 for non-bankrupt. It can be discovered that there are 6599 companies are not bankrupt while 220 companies are bankrupt. A plot is generated for the distribution.

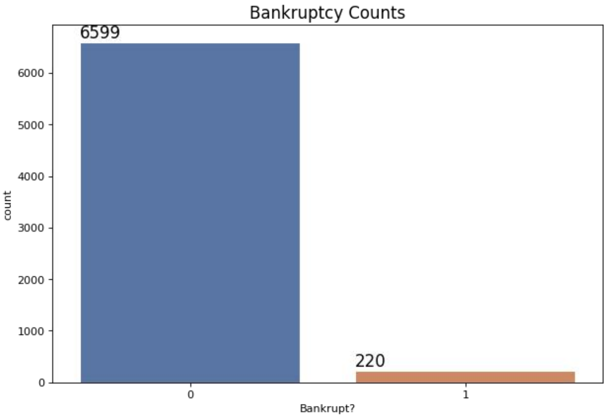


Fig2 Data Distribution

We also calculate the proportion of bankrupt and non-bankrupt companies in the data set basically judge the reliability and deviation of the data.

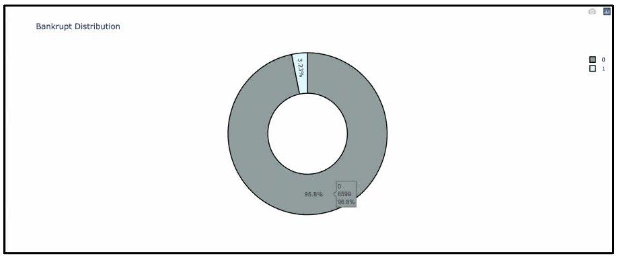


Fig3 Data Proportion

From the bar plot and pie chart above, it is obvious that the data set is imbalanced. Hence, we should focus on “Label==1” f1-score. In addition, we utilized SMOTE oversampling method to correct the imbalanced data.

SMOTE actually creates as many synthetic examples for minority class as are required so that finally two target class are well represented. It does so by synthesising samples that are close to the feature space, for the minority target class.

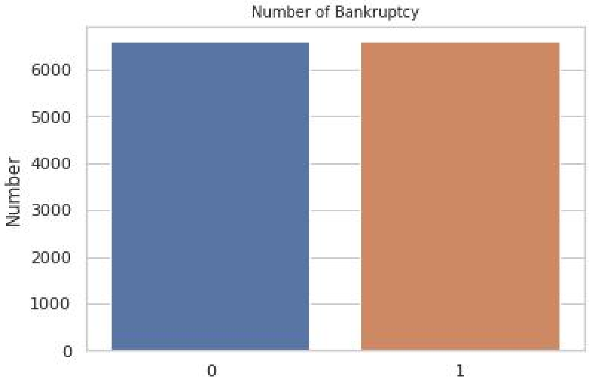


Fig4 Data Distribution after Oversampling

Also, we picked the top ten and made a box plot to see their geometric distribution.

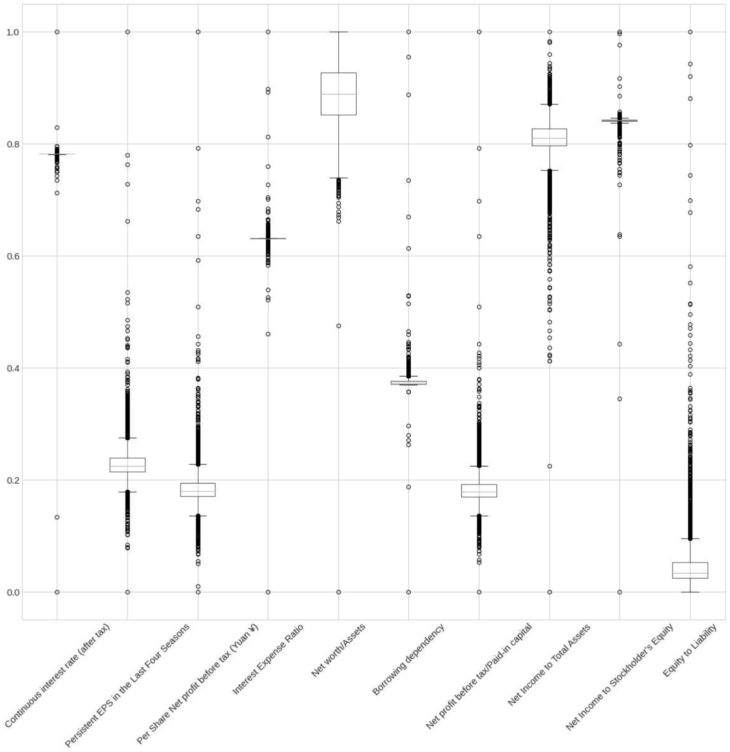


Fig5 Box plot of top10 features

Besides, for the data exploration part of the project, we decided to explore which variables in the data set are correlated to each other. Specifically, we were trying to explore which variables correlate the most of value. We created a heat map of attributes which significantly contributed to predicting whether a company is bankrupt or not.

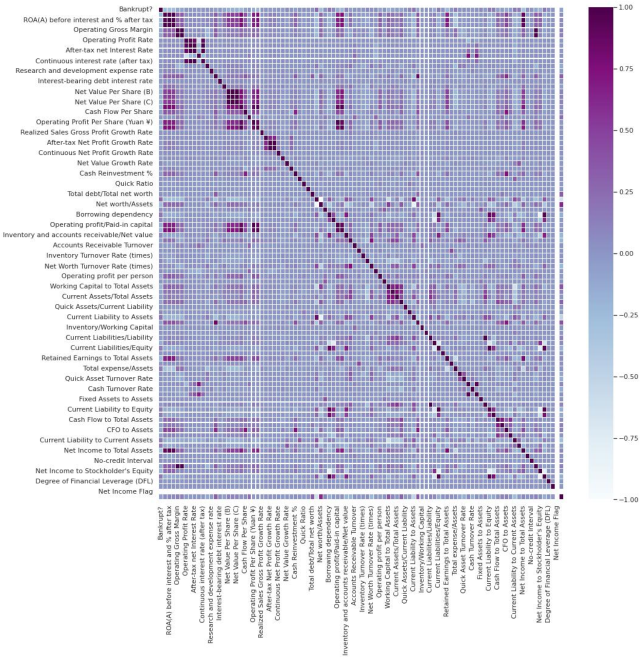
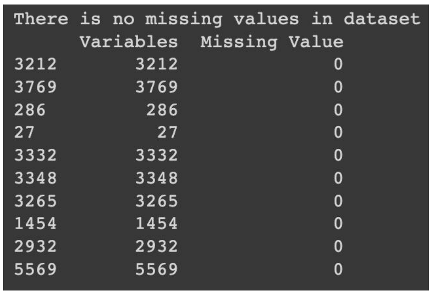


Fig6 Correlation between all variables

Data Mining Tasks:

1. Missing Data:

Since the data is the original data downloaded directly from Kaggle, we checked the missing values of the data. Fortunately, the data is clean and there are no missing values.



```
There is no missing values in dataset
      Variables  Missing Value
3212       3212              0
3769       3769              0
286         286              0
27           27              0
3332       3332              0
3348       3348              0
3265       3265              0
1454       1454              0
2932       2932              0
5569       5569              0
```

Fig7 Missing Values

2. Outliers Removal:

After that, we checked whether there are outliers in the data set and removed all the outliers to make the data clean. We defined fractionoal columns, which is a way to find out the probability that there will be outliers in the data. We separate the columns that have values between 0 and 1 from the columns that have higher and different values. The fractional columns are the columns that contain the values between 0 and 1 ,while non-fractional columns are columns that contain different values. So we should focus on non_fraction_columns because they have different values, the probability of having outliers is high.

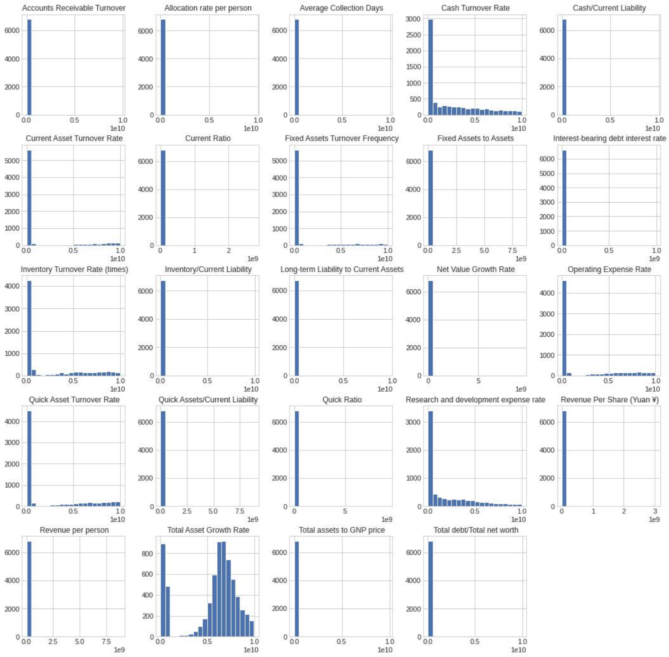


Fig8 Histograms non-fractional columns

When we tried to remove the outliers, the processing rules are as followings:

a) outliers larger than 75% percentile are replaced with upper quartile values;

b) outliers smaller than 25% percentile are replaced with lower quartile values.

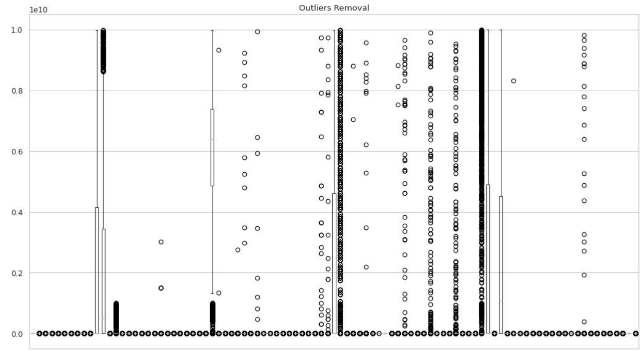


Fig9 Outliers Removal

3. <u>Dimension Reduction:</u>

Since most features are uncorrelated with each other but there are also some of them correlated. Hence, we need to reduce dimension by PCA.

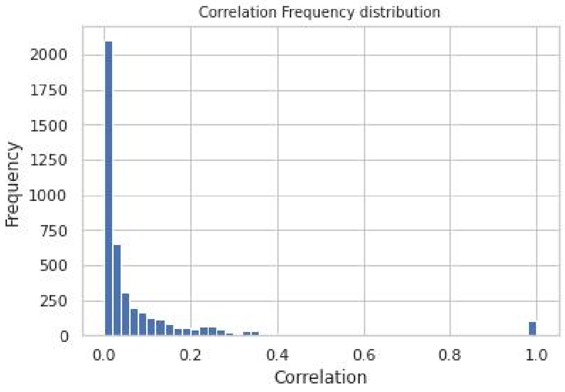
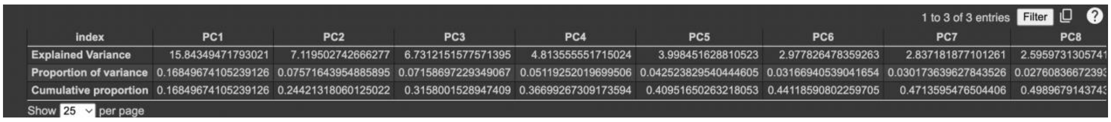


Fig10 Correlation Frequency Distribution

PCA is an orthogonal linear transformation that projects data onto a lower-dimensional space such that the total variance of the original data is mostly retained. The goal of PCA is to reduce the number of numerical variables while retaining as much predictive information.

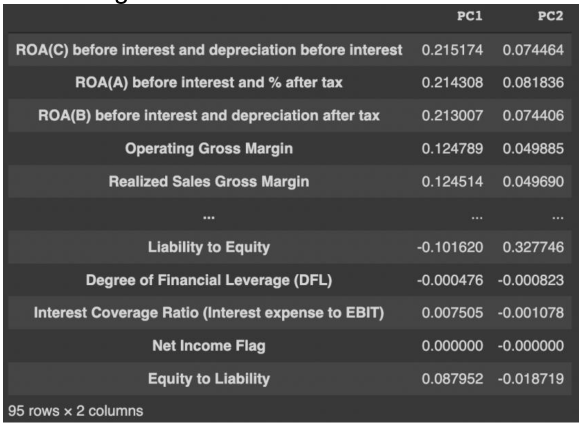


1 to 3 of 3 entries Filter

index	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8
Explained Variance	15.84349471793021	7.119502742666277	6.7312151577571395	4.813555551715024	3.998451628810523	2.977826478359263	2.837181877101261	2.5959731305741
Proportion of variance	0.16849674105239126	0.07571643954885895	0.07158697229349067	0.05119252019699506	0.042523829540444605	0.03166940539041654	0.030173639627843526	0.02760836672393
Cumulative proportion	0.16849674105239126	0.24421318060125022	0.3158001528947409	0.36699267309173594	0.40951650263218053	0.44118590802259705	0.4713595476504406	0.4989679143743

Show 25 per page

Fig11 Statistic Information of PCA



	PC1	PC2
ROA(C) before interest and depreciation before interest	0.215174	0.074464
ROA(A) before interest and % after tax	0.214308	0.081836
ROA(B) before interest and depreciation after tax	0.213007	0.074406
Operating Gross Margin	0.124789	0.049885
Realized Sales Gross Margin	0.124514	0.049690
...	...	...
Liability to Equity	-0.101620	0.327746
Degree of Financial Leverage (DFL)	-0.000476	-0.000823
Interest Coverage Ratio (Interest expense to EBIT)	0.007505	-0.001078
Net Income Flag	0.000000	-0.000000
Equity to Liability	0.087952	-0.018719

95 rows × 2 columns

Fig12 Scores of PCA

4. <u>Feature Extraction:</u>

We calculated the mutual information between the variables and the target, and the smaller the value of the mutual information, the less information we can infer from the feature about the target. We sorted the features based on their mutual information value and make a bar plot.

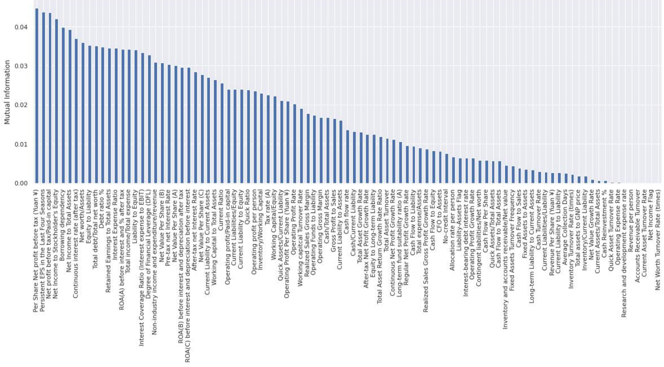


Fig13 Mutual Information

From the histogram above, we can illustrate that there are a few features (at the left of the plot) with higher mutual information values. Besides, there are still features with almost zero mutual information values on the right of the plot. In order to determine a threshold, we selected top 10 features.

Comparing PCA and Feature Extraction, we found that feature extraction has better performance, we utilized the result of feature extraction to reduce dimension.

<u>Data Mining Models/Methods:</u>

In our predictions, we used a total of seven models, which are Logistic Regression, KNN, Decision Tree, Random Forest, Naive Bayes, SVM and XGBoost. The predictor variables were collectively represented using 'Variable X' and the target variable 'bankrupt' was represented using the 'Variable y'. We split the whole data set into training data set and validation data set. 75% was used for training the classification models and the remaining 25% was used as validation data for evaluation of the classification performance of those models, along with the random state set to 42 to reduce variability.

Since there are too many attributes in the dataset, we use SelectKBest() to select the top10 representative variables for subsequent model construction based on the mutual information between the variables and target.

train_X contained 5114 records and 10 attributes, while test_X contained 1705 records and 10 variables respectively. For the train_y and test_y, we used overall as the variable, since it is so highly correlated to value. Train_y contained 5114 records and 1 variable, while test_y contained 1705 records and 1 variable respectively.

Performance Evaluation:

1. Logistic Regression:

Logistic regression is a discriminative model that estimates parameters of P(Ck|X) directly from the training data. The logistic regression is used for both prediction and profiling. The prediction is aimed to estimate the probability that an observation belongs to a particular class while the profiling is aimed to explain the contribution of predictors for explanatory purposes.

Pros:

a) The rate function is derivable in any order, and has good mathematical properties. Many existing numerical optimization algorithms can be used to find the optimal solution, and the training speed is fast;
b) Simple and easy to understand, the interpretability of the model is very good, and the influence of different features on the final result can be seen from the weight of the feature;
c) Suitable for binary classification problems, no need to scale input features;
d) The memory resource occupation is small, because only the eigenvalues of each dimension need to be stored;
e) Model the classification possibility directly, without assuming data distribution in advance, avoiding the problems caused by inaccurate assumption distribution

Cons:

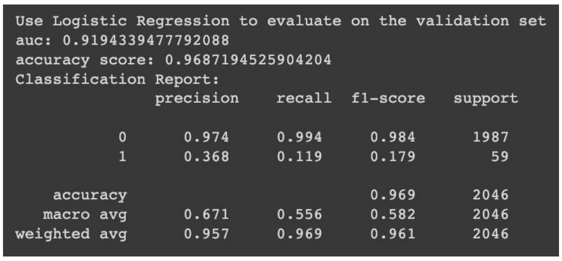
a) Logistic regression cannot be used to solve nonlinear problems, because Logistic's decision-making interview is linear;

b) It is more sensitive to multi-collinearity data;

c) It is difficult to deal with the problem of data imbalance;

d) Logistic regression itself cannot filter features, sometimes gbdt is used to filter features, and then logistic regression is used;

e) The accuracy rate is not very high, because the form is very simple (very similar to a linear model), and it is difficult to fit the true distribution of the data.



```
Use Logistic Regression to evaluate on the validation set
auc: 0.9194339477792088
accuracy score: 0.9687194525904204
Classification Report:
              precision    recall  f1-score   support

           0      0.974     0.994     0.984      1987
           1      0.368     0.119     0.179        59

    accuracy                          0.969      2046
   macro avg      0.671     0.556     0.582      2046
weighted avg      0.957     0.969     0.961      2046
```

Fig14 Classification Report of Logistic regression Model

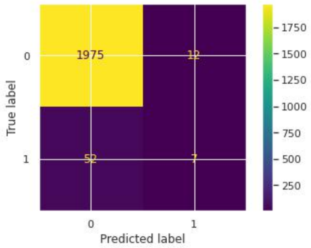


Fig15 Confusion Matrix of Logistic regression Model

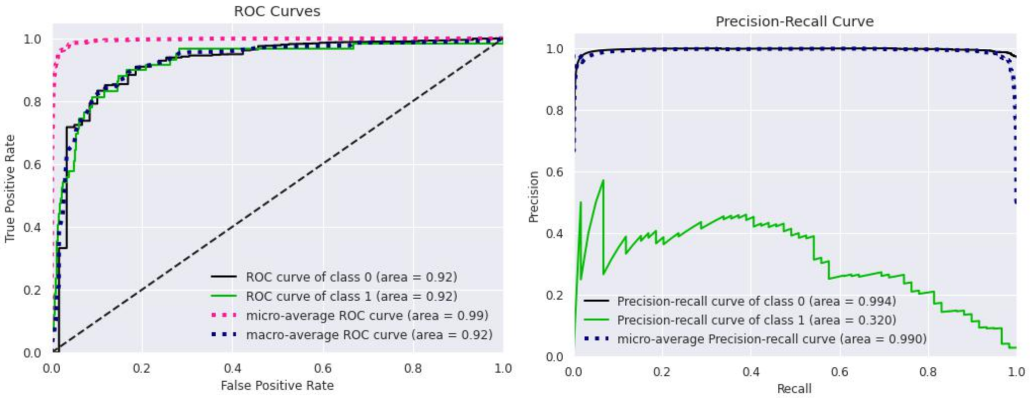


Fig16 ROC&PR Curve of Logistic regression Model

The accuracy of logistic regression model is 0.969, and f1-score is 0.984, which are both pretty high. Besides, the ROC curve is close to the top-left corner, with an AUC value of 0.92, which means the model works very well. Also, the PR curve fully focused on positive samples and the line of class of interest (class 0) is close to the top-right corner, with a value of 0.994.

2. KNN Classifier:

In KNN classification algorithm, the K-Nearest Neighbors model identifies observations that are similar to the new observation in the existing data set. It then uses the response value of the similar observation to make predictions for the new observations. Due to the majority rule, the new record should be classified as a member of the majority class of the k-neighbors.

Pros:

a) No parametric assumptions. KNN doesn't have any requirement for the distribution of predictors, which makes it versatile for real-world data sets;

b) The ability to evolve while adding new data. KNN automatically adapts while adding new data to the training set;

c) Easy implementation. KNN model operates directly on the training data not requiring any model developed on the training data. Model only requires two hyperparameters: the number of neighbors and the distance function.

Cons:

a) Time consuming at prediction time with a large data set. For a new observation, the model needs to compute the distance from the new observation to all observations in the training data set to find the nearest neighbors. Calculation cost will be huge for large data sets;

b) Curse of dimensionality;

c) Sensitive to noisy data. In the absence of training process, the model isn't robust to the noise. It may not be able to make accurate predictions for outliers.

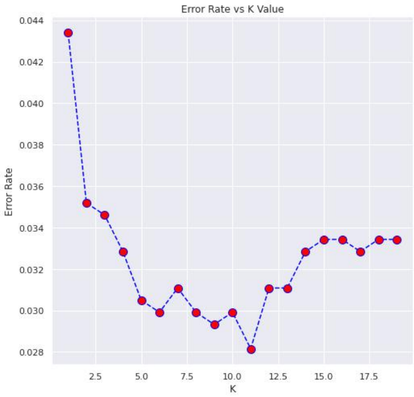
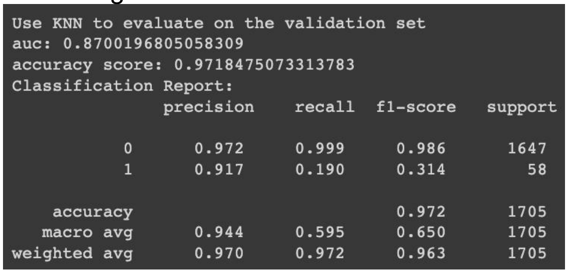


Fig17 Best K value of KNN Model



```
Use KNN to evaluate on the validation set
auc: 0.8700196805058309
accuracy score: 0.9718475073313783
Classification Report:
              precision    recall  f1-score   support

           0      0.972     0.999     0.986      1647
           1      0.917     0.190     0.314        58

    accuracy                          0.972      1705
   macro avg      0.944     0.595     0.650      1705
weighted avg      0.970     0.972     0.963      1705
```

Fig18 Classification Report of KNN Model

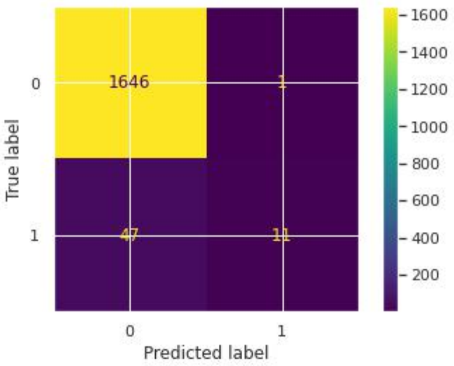


Fig19 Confusion Matrix of KNN Model

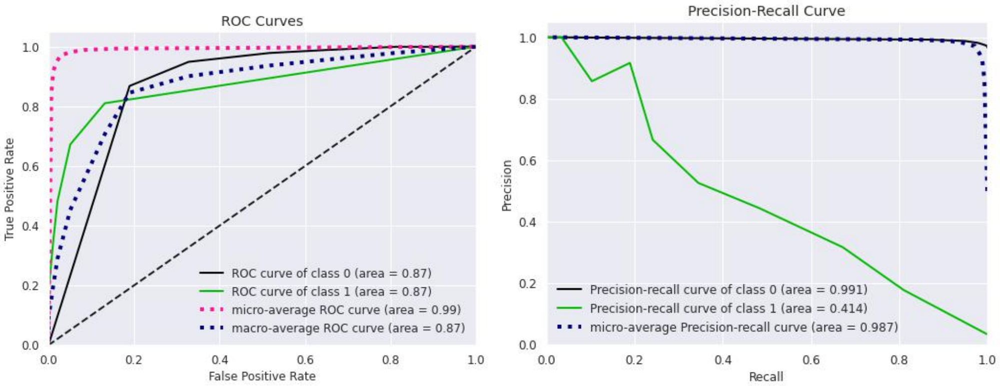


Fig20 ROC&PR Curve of KNN Model

In KNN model, we chose the best k=11, where the error dropped the lowest. The accuracy of KNN model is 0.972 and f1-score is 0.986, which are both better than those of logistic regression model. However, the ROC curve is close to the top-left corner, with an AUC value of 0.87, which means the model works not that good as logistic regression model. Also, the PR curve fully focused on positive samples and the line of class of interest (class 0) is close to the top-right corner, with a value of 0.991.

3. Decision Tree:

A decision tree can handle both categorical and numerical predictors. Classification trees return categorical response while regression trees return numeric response.

Pros:

a) Produce interpretable rules for small-sized trees;

b) Trees are fast when predicting outcomes for new records;

c) Automatic variable selection and dimension reduction possible;

d) No assumption on the distribution of training data and classifier structure;

e) Tree methods are good off-the-shelf classifiers and predictors. They are also useful for variable selection, with the most important predictors usually showing up at the top of the tree. Trees require relatively little effort from users in the following senses: First, there is no need for transformation of variables (any monotone transformation of the variables will give the same trees). Second, variable subset selection is automatic since it is part of the split selection.

f) Trees are also intrinsically robust to outliers, since the choice of a split depends on the ordering of values and not on the absolute magnitudes of these values.

g) Classification and regression trees are nonlinear and nonparametric. This allows for a wide range of relationships between the predictors and the outcome variable.

h) Trees can handle missing data without having to impute values or delete records with missing values.

i) A very important practical advantage of trees is the transparent rules that they generate. Such transparency is often useful in managerial applications, though this advantage is lost in the ensemble versions of trees (random forests, boosted trees).

Cons:

a) Trees tend to be instable and are sensitive to changes in the data, and even a slight change can cause very different splits;

b) CART algorithm is a greedy search algorithm. It doesn't guarantee an absolute optimal solution. However, it produces a relatively good solution;

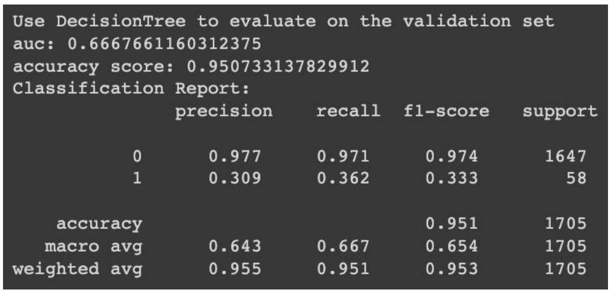
c) Decision boundaries are orthogonal. The trees are sensitive to rotation on the training data;

d) The trees failed to represent complex variable interactions in a precise way compared to other models;

e) Since the splits are done on one predictor at a time, rather than on combinations of predictors, the tree is likely to miss relationships between predictors, in particular linear structures like those in linear or logistic regression models. Classification trees are useful classifiers in cases where horizontal and vertical splitting of the predictor space adequately divides the classes. In such cases, a classification tree is expected to have lower performance than methods such as discriminant analysis;

f) Classification trees require a large dataset in order to construct a good classifier. From a computational aspect, trees can be relatively expensive to grow, because of the multiple sorting involved in computing all possible splits on every variable. Methods for avoiding overfitting, such as cross-validation or pruning the data using the validation set, add further computation time;

g) Although trees are useful for variable selection, one challenge is that they "favor" predictors with many potential split points. This includes categorical predictors with many categories and numerical predictors with many different values. Such predictors have a higher chance of appearing in a tree.



```
Use DecisionTree to evaluate on the validation set
auc: 0.6667661160312375
accuracy score: 0.950733137829912
Classification Report:
              precision    recall  f1-score   support

           0      0.977     0.971     0.974      1647
           1      0.309     0.362     0.333        58

    accuracy                          0.951      1705
   macro avg      0.643     0.667     0.654      1705
weighted avg      0.955     0.951     0.953      1705
```

Fig21 Classification Report of Decision Tree Model

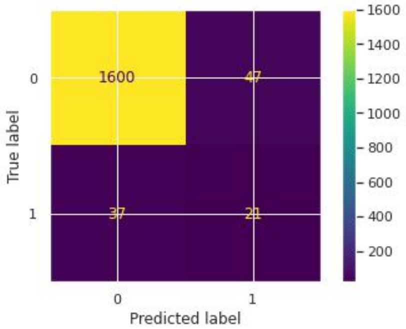


Fig22 Confusion Matrix of Decision Tree Model

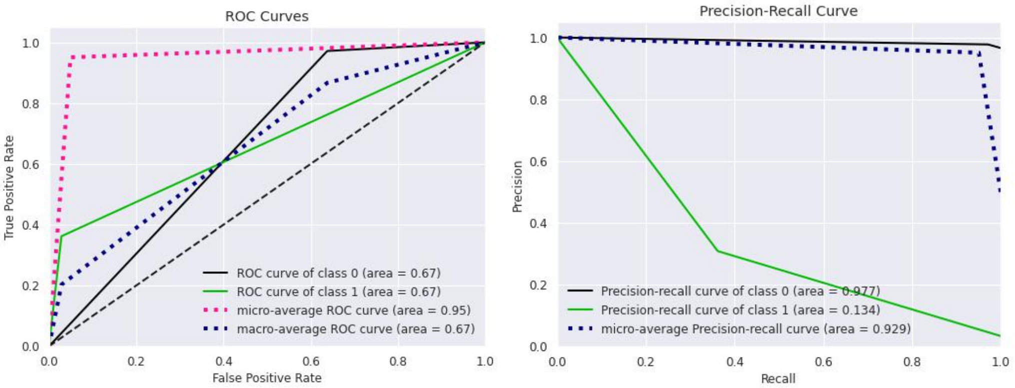


Fig23 ROC&PR Curve of Decision Tree Model

The accuracy of decision tree model is 0.95, and f1-score is 0.977, which are both lower than those of logistic regression model. Besides, the ROC curve is close to the diagonal, with an AUC value of 0.67, which means the model works just good. Also, the PR curve fully focused on positive samples and the line of class of interest (class 0) is close to the top-right corner, with a value of 0.977.

4. Random Forest:

Random Forest is an ensemble of decision trees. Each decision tree is trained on a bootstrap sample while final prediction is made considering predictions of all individual trees. A random forest may search for the best predictor to split among a random subset of predictors (instead of searching for the best predictor among all predictors), thereby introducing extra randomness. Extremely randomized trees, instead of searching for the best splitting point for a predictor, may choose a random split point to grow the trees.

Pros:

a) It has very high accuracy;

b) Ability to operate efficiently on large data sets;

c) Randomness is introduced, which is not easy to overfit;

d) Random forest has good anti-noise ability, but it will overfit when the data noise is relatively large;

e) Can handle very high-dimensional data without dimensionality reduction;

f) Not only can handle discrete data, but also continuous data, and no need to normalize the data set;

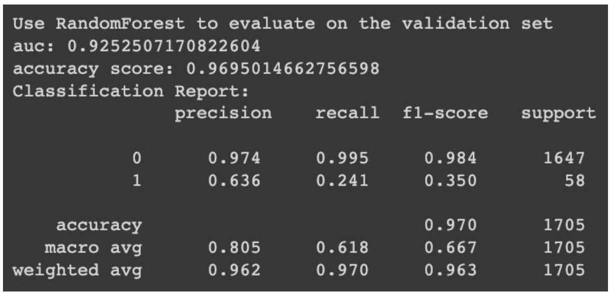
g) The training speed is fast, and the importance of variables can be ranked;

h) Easy to parallelize;

i) Can handle missing values well.

Cons:

Although the random forest algorithm is fast enough, when the number of decision trees in the random forest is large, the space and time required for training will be very large, which will lead to slower models. Therefore, in practical applications, if the real-time requirements are very high, it is better to choose other algorithms.



```
Use RandomForest to evaluate on the validation set
auc: 0.9252507170822604
accuracy score: 0.9695014662756598
Classification Report:
              precision    recall  f1-score   support

           0      0.974     0.995     0.984      1647
           1      0.636     0.241     0.350        58

    accuracy                          0.970      1705
   macro avg      0.805     0.618     0.667      1705
weighted avg      0.962     0.970     0.963      1705
```

Fig24 Classification Report of Random Forest Model

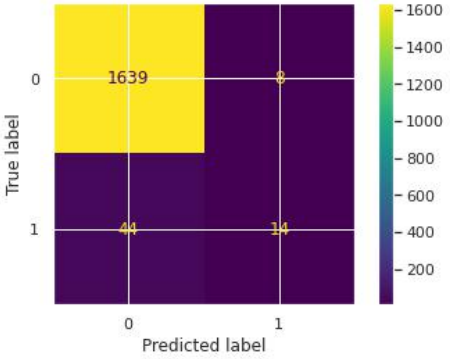


Fig25 Confusion Matrix of Random Forest Model

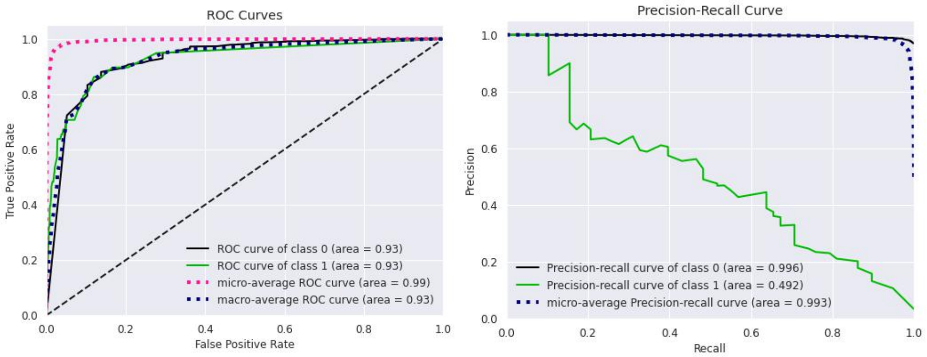


Fig26 ROC&PR Curve of Random Forest Model

The accuracy of random forest model is 0.97, and f1-score is 0.984, which are both pretty high, similar to those in logistic regression model. Besides, the ROC curve is close to the top-left corner, with an AUC value of 0.93, which means the model works very well. Also, the PR curve fully focused on positive samples and the line of class of interest (class 0) is close to the top-right corner, with a value of 0.996.

5. SVM:

Support vector machines are supervised learning models and related learning algorithms that analyze data in classification and regression analysis. Given a set of training instances, each marked as belonging to one or the other of the two classes, the SVM training algorithm creates a model that assigns new instances to one of the two classes, making it a non-probabilistic two Metalinear classifier. The SVM model represents instances as points in space, such that the mapping makes instances of individual classes separated by as wide a noticeable interval as possible. Then, map the new instances to the same space and predict the class they belong to based on which side of the interval they fall on.

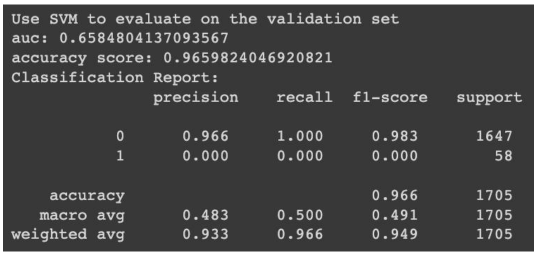
Pros:

a) Efficient. It does not involve probability measurement and the law of large numbers, etc., and realizes efficient "transduction reasoning" from training samples to forecast samples, which greatly simplifies the usual problems of classification and regression.
b) The final decision function of SVM is only determined by a small number of support vectors, and the computational complexity depends on the number of support vectors, not the dimension of the sample space, which avoids the "curse of dimensionality" in a sense.
c) Robust: adding and deleting non-support vector samples has no effect on the model; the support vector sample set has a certain robustness; the SVM method is not sensitive to the selection of the kernel.

Cons:

a) SVM algorithm is difficult to implement for large-scale training samples.

b) It is difficult to solve the multi-classification problem with SVM.



```
Use SVM to evaluate on the validation set
auc: 0.6584804137093567
accuracy score: 0.9659824046920821
Classification Report:
              precision    recall  f1-score   support

           0      0.966     1.000     0.983      1647
           1      0.000     0.000     0.000        58

    accuracy                          0.966      1705
   macro avg      0.483     0.500     0.491      1705
weighted avg      0.933     0.966     0.949      1705
```

Fig27 Classification Report of SVM Model

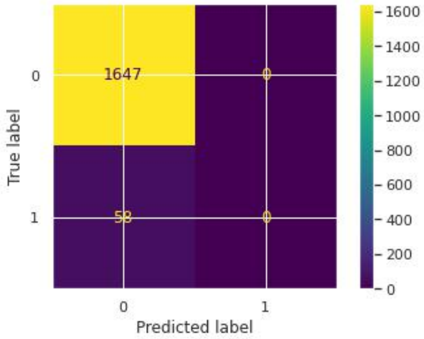


Fig28 Confusion Matrix of SVM Model

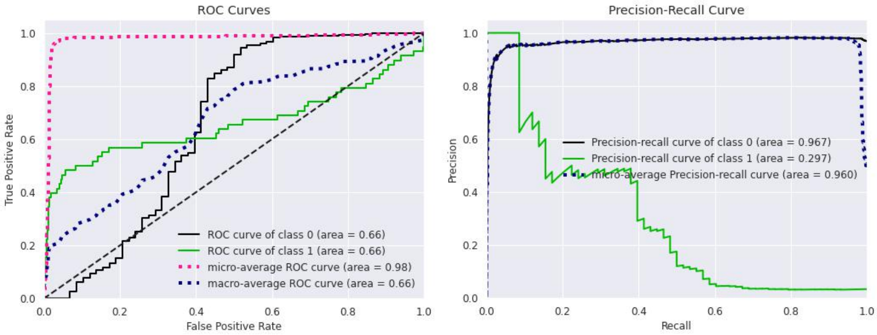


Fig29 ROC&PR Curve of SVM Model

The accuracy of SVM model is 0.966, and f1-score is 0.983, which are both pretty high. However, the ROC curve is close to the diagonal, with an AUC value of 0.66, which means the model works just good. Also, the PR curve fully focused on positive samples and the line of class of interest (class 0) is close to the top-right corner, with a value of 0.967.

6. XGBoost:

XGBoost is one of the boosting algorithms. The idea of Boosting algorithm is to integrate many weak classifiers together to form a strong classifier. Because XGBoost is a boosted tree model, it is the integration of many tree models to form a strong classifier. The tree model used is the CART regression tree model.

Pros:

a) Can handle missing values;

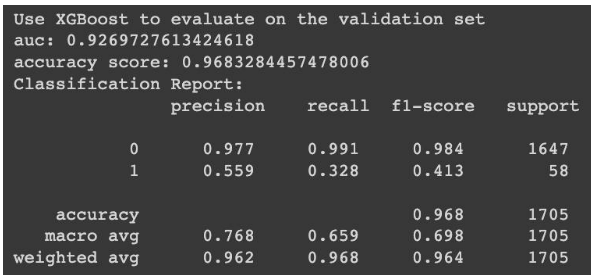
b) XGBoost draws on the practice of random forest and supports column sampling, which can not only reduce overfitting, but also reduce calculation;

c) Higher precision and greater flexibility.

Cons:

a) Time consuming. Although the use of pre-sorting and approximation algorithms can reduce the amount of computation to find the best split point, it still needs to traverse the data set during the node splitting process;

b) The space complexity of the presort process is too high.



```
Use XGBoost to evaluate on the validation set
auc: 0.9269727613424618
accuracy score: 0.9683284457478006
Classification Report:
              precision    recall  f1-score   support

           0      0.977     0.991     0.984      1647
           1      0.559     0.328     0.413        58

    accuracy                          0.968      1705
   macro avg      0.768     0.659     0.698      1705
weighted avg      0.962     0.968     0.964      1705
```

Fig30 Classification Report of XGBoost Model

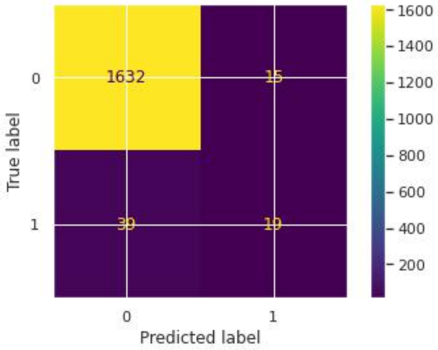


Fig31 Confusion Matrix of XGBoost Model

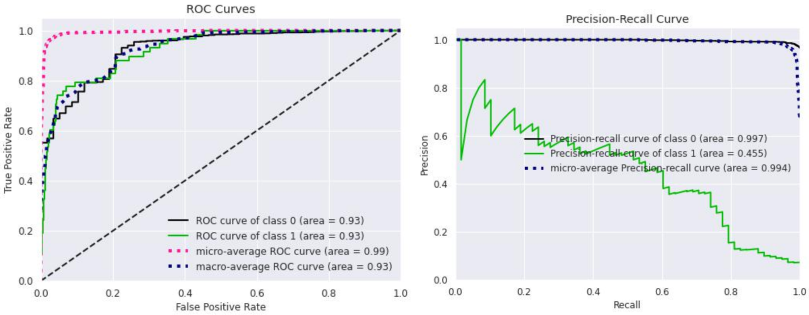


Fig32 ROC&PR Curve of XGBoost Model

The accuracy of XGBoost model is 0.968, and f1-score is 0.984, which are both pretty high,similar to those in logistic regression model and random forest model. Besides, the ROC curve is close to the top-left corner, with an AUC value of 0.93, which means the model works very well,higher than those in logistic regression and random forest model. Also, the PR curve fully focused on positive samples and the line of class of interest (class 0) is close to the top-right corner, with a value of 0.997.

7. Naive Bayes:

Naive Bayes assumes that the predictors are conditionally independent given the target class. This simplifies the computation of joint distribution over the predictors.

Pros:

a) It is simpler and faster comparing to the Exact Bayes;

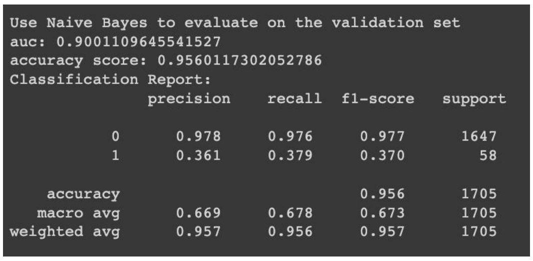
b) It is highly scalable as it scales linearly with number of predictors;

c) It is relatively robust to noisy data.

Cons:

a) Even though Naive Bayes can work with less data, it requires a large number of records to obtain reliable parameter estimates;

b) Compared to the Exact Bayes, Naive Bayes retains the order of the propensity order but fails to generate the accurate propensity.



```
Use Naive Bayes to evaluate on the validation set
auc: 0.9001109645541527
accuracy score: 0.9560117302052786
Classification Report:
              precision    recall  f1-score   support

           0      0.978     0.976     0.977      1647
           1      0.361     0.379     0.370        58

    accuracy                          0.956      1705
   macro avg      0.669     0.678     0.673      1705
weighted avg      0.957     0.956     0.957      1705
```

Fig33 Classification Report of Naive Bayes Model

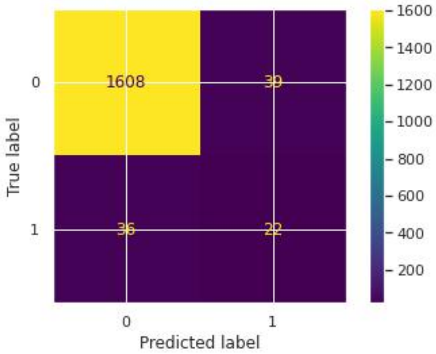


Fig34 Confusion Matrix of Naive Bayes Model

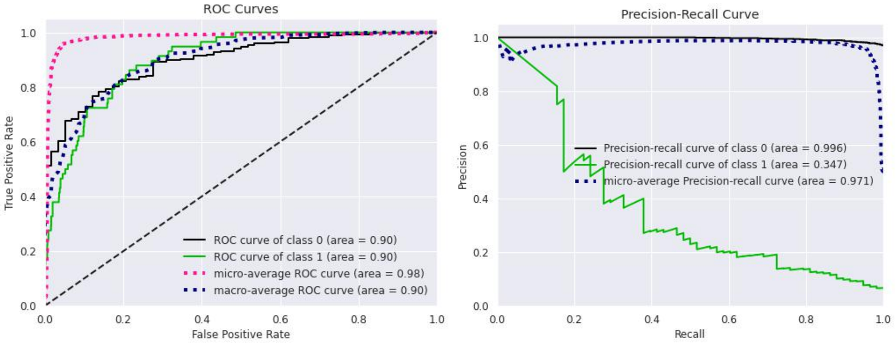


Fig35 ROC&PR Curve of Naive Bayes Model

The accuracy of Naive Bayes model is 0.956, and f1-score is 0.977, which are both pretty high. Besides, the ROC curve is close to the top-left corner, with an AUC value of 0.9, which means the model works very well. Also, the PR curve fully focused on positive samples and the line of class of interest (class 0) is close to the top-right corner, with a value of 0.996.

Project Results:

Our final result determined that the best prediction model for our dataset would be the XGBoost Model. Although the accuracy of XGBoost model is just 0.968, ranking the 4th place, it is not far behind the top three, only within 0.04. Besides the precision of XGBoost reaches 0.977, ranking the second place, where the difference between XGBoost and Naive Bayes is just 0.001. At the same time, the F1-score of XGBoost achieves 0.984, ranking second, only 0.02 behind KNN. Also the AUC of XGBoost is 0.927, which is a really high value, representing that this model works very well.

Overall, taking accuracy, precision, f1-score and AUC into account, we think that XGBoost is the best model to run on our dataset.

Method	Accuracy	Precision	Recall	AUC	F1-score
Logistic Regression	0.969	0.971	0.998	0.903	0.984
KNN	0.972	0.972	0.999	0.87	0.986
Decision Tree	0.951	0.977	0.971	0.667	0.974
Random Forest	0.97	0.974	0.995	0.925	0.984
SVM	0.966	0.966	1	0.658	0.983
XGBoost	0.968	0.977	0.991	0.927	0.984
Naive Bayes	0.956	0.978	0.976	0.9	0.977

Impact of the Project Outcomes:

With the development of the world economy, private enterprises have gradually become the main carrier of current economic development.Under the condition of market economy, the bankruptcy and establishment of enterprises are just like the birth and death of human beings, which exist objectively. For investors and other stakeholders who have an interest in the enterprise, accurately judging whether the enterprise is facing bankruptcy will be able to minimize their financial risks and obtain better economic benefits. It can be seen that it is very important to predict the future development situation of the company. However, the development trend of an enterprise is affected by many factors. Without considering the social environment and other factors, our model only focuses on the parameters provided in the data set.

In this project, the goal was to be able to identify whether a company is bankrupt or not. This problem was to be solved using the real-world data, through which it was observed that the number of non-bankrupt is far more than that of bankrupt. Therefore, the class of interest represented by label 0 was the class of interest. To some extent, our project can better predict whether a business will go bankrupt.

Among all the evaluation indicators, XGBoost has pretty high accuracy, precision, F1-score and the highest AUC score. Accuracy is the proportion of correctly classified records, which represents the ability to classify observations correctly into their respective classes, while precision is the proportion of the observations that model classifies as the positive observations are class of interest. F1-score, which can find the optimal combination of precision and recall. Considering all factors, we believed that XGBoost is the best classification model for our data.