Bankruptcy prediction model

Middle project /
feat Kirill Tiufanov and Andrey Rogatin

18.11.2022

Table of contents

Goals

Dataset
Data exploration
Data transformation
Data rebalancing with SMOTE()
RFE
statsmodels.api
confusion_matrix
classification_report
ROC_curve
RandomForestClassifier
Conclusions

Goals

To predict bankruptcy of companies traded on stock exchange based on their financial data

1. **Task 1** - Find the most important features (financial parameters)

2. **Task 2** - Build a model that can that predict company bankruptcy based on limited amount of data

Dataset

The data were collected from the Taiwan Economic Journal for the years 1999 to 2009. Company bankruptcy was defined based on the business regulations of the Taiwan Stock Exchange.

Data consists from 95 features and 6819 instances.

```
In [33]: #checking df info
         df.head()
```

Out[33]:

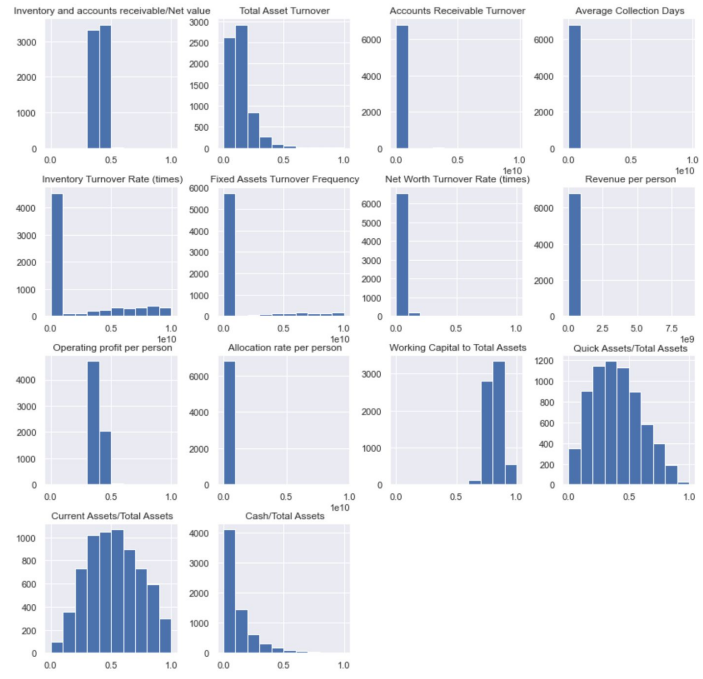
	Bankrupt?	ROA(C) before interest and depreciation before interest	Operating Gross Margin	Operating Expense Rate	Research and development expense rate	Cash flow rate	Tax rate (A)	Net Value Per Share (B)	Persistent EPS in the Last Four Seasons	Cash Flow Per Share	...	Current Asset Turnover Rate	Quick Asset Turnover Rate	Working capitcal Turnover Rate	Cash Turnover Rate
0	1	0.370594	0.601457	1.258227e-14	0.000000	0.458143	0.0	0.147950	0.169141	0.311664	...	7.010000e-02	6.550000e-01	0.593831	0.0458
1	1	0.464291	0.610235	2.900751e-14	0.000000	0.461867	0.0	0.182251	0.208944	0.318137	...	1.065198e-14	7.700000e-01	0.593916	0.2490
2	1	0.426071	0.601450	2.363661e-14	0.002555	0.458521	0.0	0.177911	0.180581	0.307102	...	1.791094e-13	1.022676e-13	0.594502	0.0761
3	1	0.399844	0.583541	1.079968e-14	0.000000	0.465705	0.0	0.154187	0.193722	0.321674	...	8.140000e-01	6.050000e-01	0.593889	0.2030
4	1	0.465022	0.598783	7.897898e-01	0.000000	0.462746	0.0	0.167502	0.212537	0.319162	...	6.680000e-01	5.050000e-01	0.593915	0.0824

5 rows × 41 columns

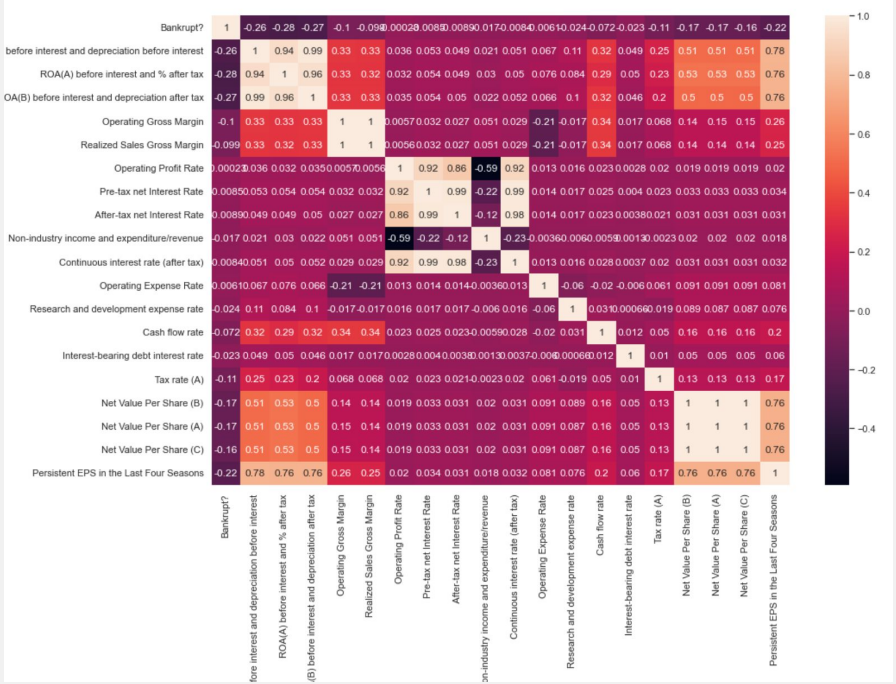
Data exploration

Two types of unimportant features were manually found (with histogram and heatmap graphs).
We've dropped 55 out of 95

1. Features with no distribution



2. Features with high correlation



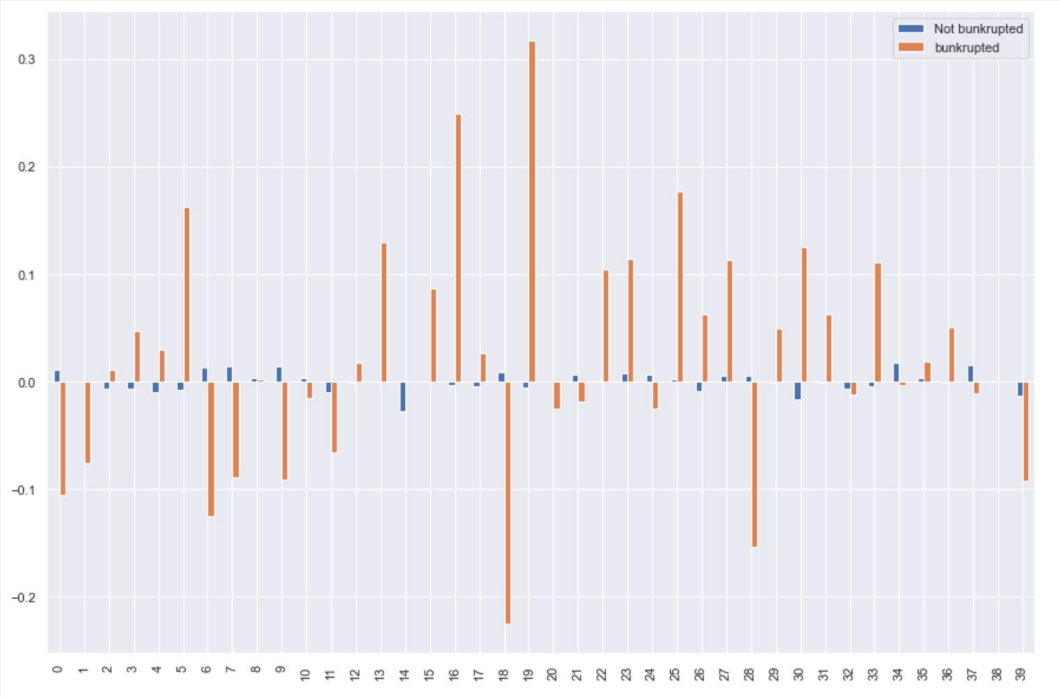
Data transformation

1. Data split into train and test parts

2. Then we've tried MinMaxScaler() and StandardScaler() functions, and got significantly higher results with SC.

3. And visually checked the difference in feature means grouped by target value (Bunkruptcy 0/1)

After transformation features showed good difference bases on target value



Data rebalancing with SMOTE()

We had only **220** bankrupted companies VS **6599** not bankrupted companies in our train dataset. The model would ignore the minority label.

SMOTE - Synthetic Minority Oversampling Technique

It helps to rebalance data and artificially generate more instances of the target value. It's called oversampling

Before

```
0    6599
1     220
Name: Bankrupt?, dtype: int64
```

After

```
smote = SMOTE()
y.value_counts()
X_sm, y_sm = smote.fit_resample(X_train, y_train)
y_sm.value_counts()
```

```
0    5279
1    5279
Name: Bankrupt?, dtype: int64
```

RFE

RFE - Feature ranking with recursive feature elimination.
Running determined model with gradual extraction of different features and ranking them based on the outcome model score.
The rank shows how important features for model performance.

It labeled 20 the most important features
that we now can feed into the model.

In [23]:
```
#checking features with high importancy
from sklearn.feature_selection import RFE
data_final_vars=df.columns.values.tolist()
logreg = LogisticRegression(max_iter=1000)
rfe = RFE(logreg, n_features_to_select=20)
rfe = rfe.fit(X_sm, y_sm.values.ravel())
print(rfe.support_)
print(rfe.ranking_)
```

```
[ True False False False False False  True  True  True  True False  True
  True  True  True  True  True False False  True False False False False
  True  True  True  True False False False  True False  True False  True
  True False False False]
[ 1 12 17 19 13  2  1  1  1  1 18  1  1  1  1  1  1  6 11  1  5  4 15  3
  1  1  1  1 16 14 20  1 10  1  7  1  1  9 21  8]
```

statsmodels.api

```
Optimization terminated successfully.
         Current function value: 0.460275
         Iterations 7
                                    Results: Logit
==========================================================================================================
Model:                          Logit                       Pseudo R-squared:             0.336
Dependent Variable:             Bankrupt?                   AIC:                          9759.1664
Date:                           2022-11-18 09:26            BIC:                          9904.4592
No. Observations:               10558                       Log-Likelihood:               -4859.6
Df Model:                       19                          LL-Null:                      -7318.2
Df Residuals:                   10538                       LLR p-value:                  0.0000
Converged:                      1.0000                      Scale:                        1.0000
No. Iterations:                 7.0000
----------------------------------------------------------------------------------------------------------
                                                      Coef.  Std.Err.    z    P>|z|   [0.025  0.975]
----------------------------------------------------------------------------------------------------------
 ROA(C) before interest and depreciation before interest -0.3802   0.0548 -6.9354 0.0000 -0.4876 -0.2727
 Cash flow rate                                          -0.0584   0.0639 -0.9140 0.3607 -0.1837  0.0669
 Tax rate (A)                                            -0.1708   0.0275 -6.2188 0.0000 -0.2246 -0.1170
 Net Value Per Share (B)                                  0.0105   0.0476  0.2200 0.8259 -0.0828  0.1037
 Persistent EPS in the Last Four Seasons                 -0.9255   0.1141 -8.1124 0.0000 -1.1491 -0.7019
 Operating Profit Per Share (Yuan ¥)                      0.7292   0.0841  8.6697 0.0000  0.5643  0.8940
 Total Asset Growth Rate                                 -0.0170   0.0285 -0.5964 0.5509 -0.0729  0.0389
 Debt ratio %                                             1.2984   0.1210 10.7326 0.0000  1.0613  1.5355
 Contingent liabilities/Net worth                        -0.0337   0.0374 -0.9017 0.3672 -0.1069  0.0396
 Inventory and accounts receivable/Net value             -0.1068   0.0488 -2.1871 0.0287 -0.2026 -0.0111
 Fixed Assets Turnover Frequency                          0.1162   0.0253  4.5888 0.0000  0.0666  0.1658
 Net Worth Turnover Rate (times)                         -0.3317   0.0432 -7.6695 0.0000 -0.4165 -0.2469
 Cash/Total Assets                                       -0.0538   0.0383 -1.4058 0.1598 -0.1289  0.0212
 Current Liability to Assets                             -0.4678   0.1405 -3.3303 0.0009 -0.7432 -0.1925
 Operating Funds to Liability                             0.2304   0.0634  3.6322 0.0003  0.1061  0.3548
 Current Liabilities/Liability                            0.3857   0.0794  4.8580 0.0000  0.2301  0.5414
 Quick Asset Turnover Rate                               -0.0537   0.0277 -1.9362 0.0528 -0.1080  0.0007
 Cash Turnover Rate                                      -0.2125   0.0286 -7.4386 0.0000 -0.2684 -0.1565
 Cash Flow to Total Assets                               -0.0467   0.0442 -1.0579 0.2901 -0.1333  0.0398
 Cash Flow to Liability                                  -0.1026   0.0394 -2.6002 0.0093 -0.1799 -0.0253
==========================================================================================================
```

P> | z |

If it's < 0.05, that means the coefficient really impact our target value statistically.

From our model we can drop 5 columns to optimize it.

confusion_matrix

```
confusion_matrix = confusion_matrix(y_test, y_pred)
print(confusion_matrix)
```

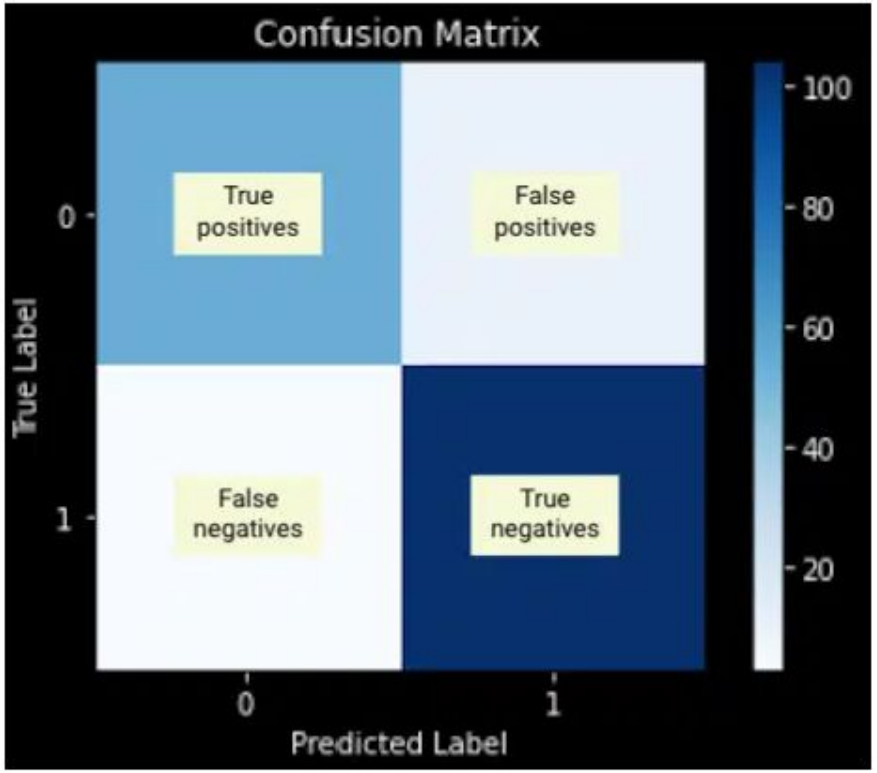
```
[[1180  140]
 [   4   40]]
```

True Positive
The model predicted true and it is true.

True Negative
The model predicted false and it is false.

False Positive
The model predicted True and it is false.

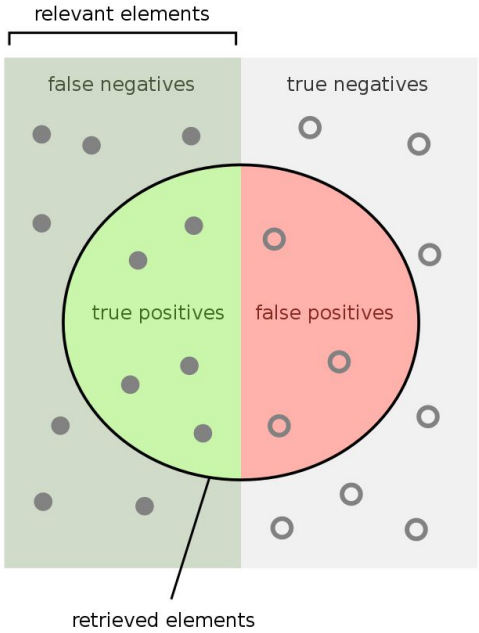
False Negative
The model predicted false and it is true.



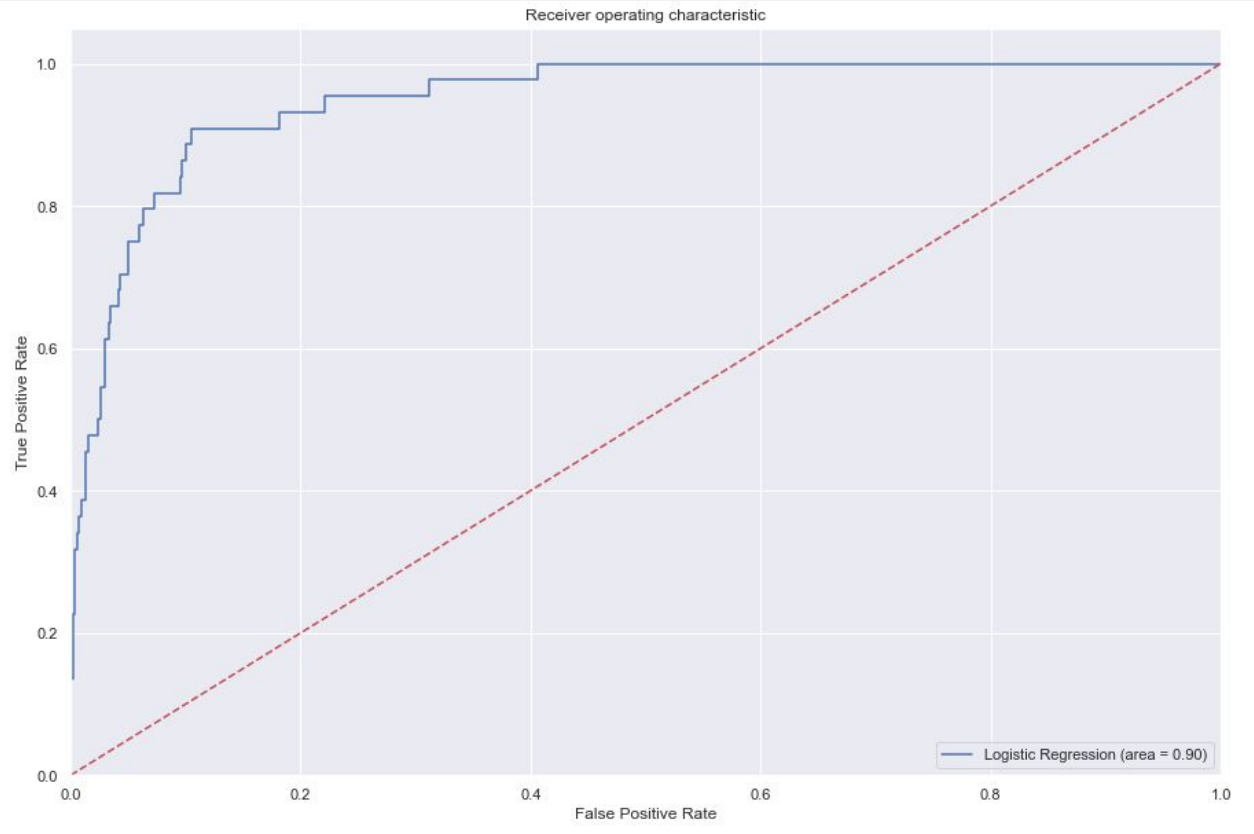
classification_report

	precision	recall	f1-score	support
0	1.00	0.89	0.94	1320
1	0.22	0.91	0.36	44
accuracy			0.89	1364
macro avg	0.61	0.90	0.65	1364
weighted avg	0.97	0.89	0.92	1364

Precision (also called positive predictive value) is the fraction of relevant instances among the retrieved instances, while **recall** (also known as sensitivity) is the fraction of relevant instances that were retrieved. Both precision and recall are therefore based on relevance.



ROC_curve



A receiver operating characteristic curve, or **ROC curve**, is a graphical plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied.

sklearn.ensemble.RandomForestClassifier

Random forest is a *Supervised Machine Learning Algorithm* that is *used widely in Classification and Regression problems*. It builds decision trees on different samples and takes their majority vote for classification and average in case of regression.

RandomForestClassifier()

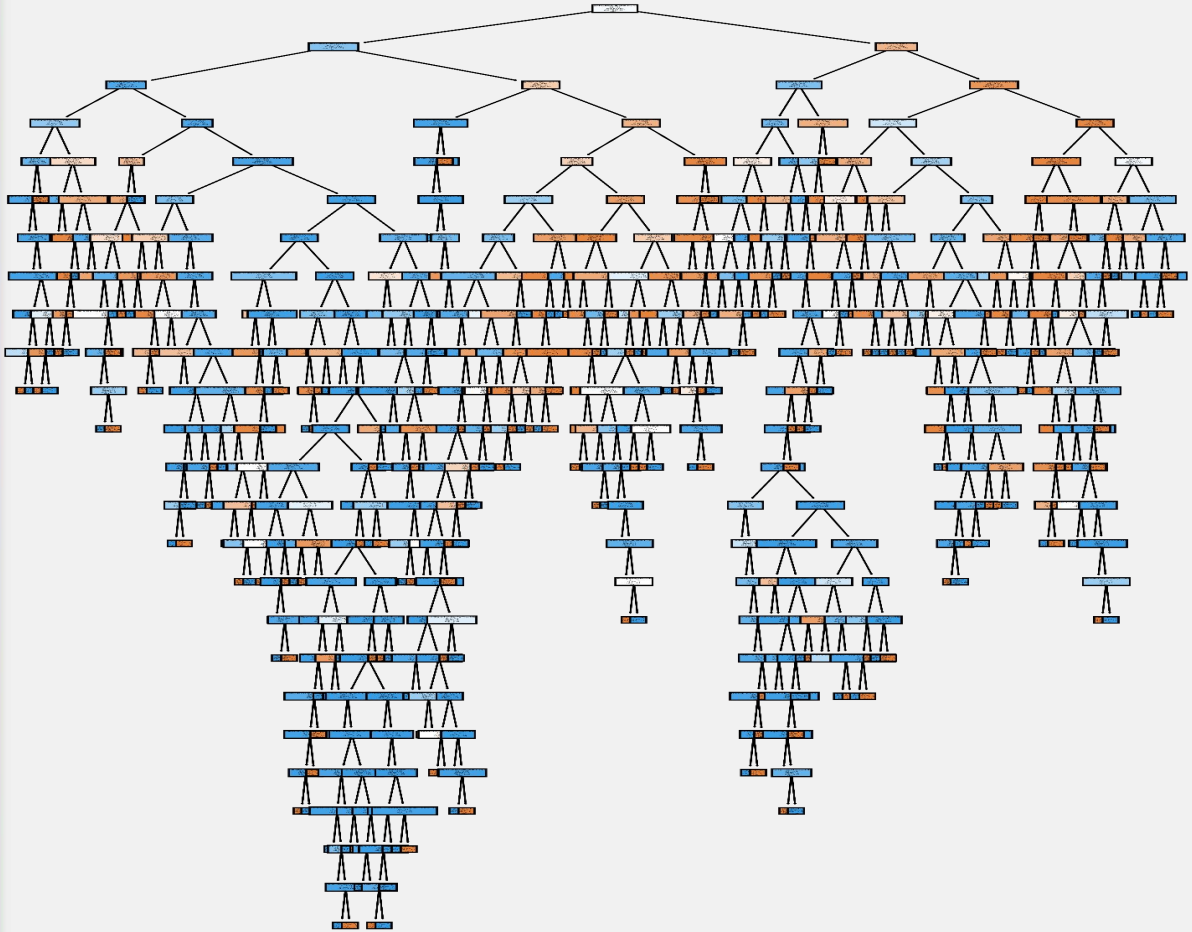
Results with the same data preparations and SMOTE:

```
confusion_matrix(y_test, y_pred)
```

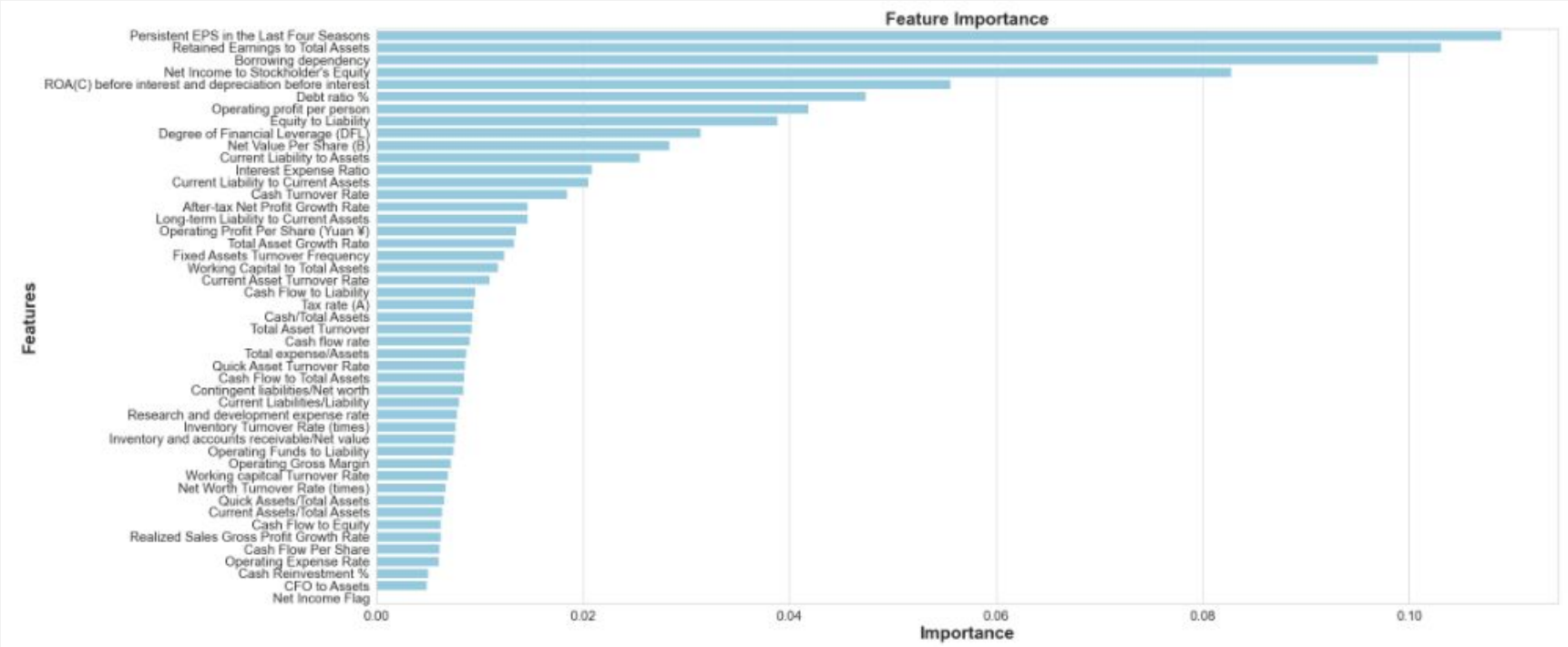
```
array([[1292,   42],
       [   5, 1301]], dtype=int64)
```

	precision	recall	f1-score	support
0	1.00	0.97	0.98	1334
1	0.97	1.00	0.98	1306
accuracy			0.98	2640
macro avg	0.98	0.98	0.98	2640
weighted avg	0.98	0.98	0.98	2640

sklearn.ensemble.RandomForestClassifier



The random forest is a classification algorithm consisting of many decisions trees. It uses bagging and feature randomness when building each individual tree to try to create an uncorrelated forest of trees whose prediction by committee is more accurate than that of any individual tree.



The most important features are -

	Features	Gini-Importance
0	Persistent EPS in the Last Four Seasons	0.108947
1	Retained Earnings to Total Assets	0.103113
2	Borrowing dependency	0.097019
3	Net Income to Stockholder's Equity	0.082778
4	ROA(C) before interest and depreciation befor...	0.055619
5	Debt ratio %	0.047413

How can we use and improve this model?

Business outcomes:
- predict bankruptcy risks by using information from open sources
- measure most important coefficients and use it in decision making

Risks:
- information from open sources can be manipulated
- the model doesn't consider the global conjecture (economy crises, wars, natural disasters)

Hypotheses for improvement:
- to add economic cycles to the model
- take into consideration political situation, regional conflicts and wars, economical crises and use this data it in the model prediction.

Thank you for your attention!