BANKRUPTCY PREVENTION

PROJECT - 218

GROUP-5

GROUP MEMBERS

Shubham Ajitkumar Sutar

Suchin Biradar

Abhishek Kumar Sahu Ere

Anil Parabati Jadhav

Jagadeesh Korukonda

Margam Navya

CONTENT

- Abstract
- Introduction
  - Objective
  - Project Architecture
  - Challenges Faced
- Data Collection
- EDA
- Data Visualization
- Feature Engineering/Data Pre-processing
- Model Building
- Model Evaluation
- Model Deployment
- Result

ABSTRACT

The Classification algorithm is a Supervised Learning technique, that is used to identify the category of new observations on the basis of training data. In classification, a program uses the dataset or observations provided to learn how to categorize new observations into various classes or groups.

Classification predictive modelling is trained using data or observations, and new observations are categorized into classes or groups.

This algorithm is simple to implement, robust to noisy training data, and effective if training data is large

Classification algorithms are used to analyse discrete values, The output variable must be either continuous nature or real value.

INTRODUCTION

Objective :

- This is a classification project, since the variable to predict is binary (bankruptcy or non-bankruptcy). The goal here is to model the probability that a business goes bankrupt from different features.
- The data file contains 7 features about 250 companies

Project Architecture :

Data Collection – Data Cleaning/ EDA – Data Visualization – Model Building – Model Evaluation – Model Deployment – Result.

Challenges Faced :

- As the data set contains only 3 unique values and half of the values are identified as duplicated values and more the size of the data is small with 250 entries which results in inaccurate reporting, data integrity problem.
- The biggest challenge is identifying the result in business point of view and making a decision on selecting the best model for deployment part.

DATA COLLECTION

The dataset provides information whether companies that are going under bankruptcy and non bankruptcy. The data file contains 7 features about 250 companies.

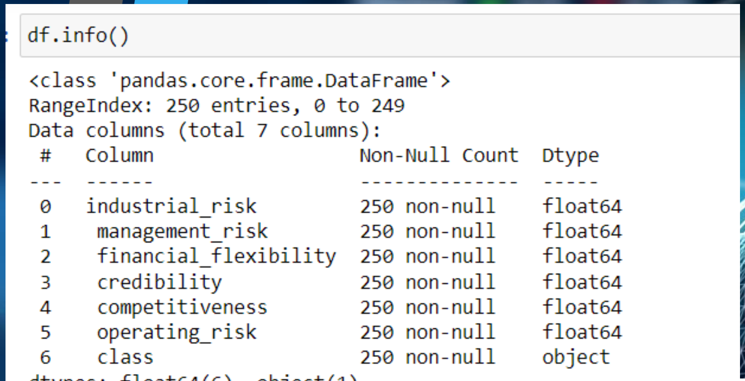
The data set includes the following variables:

- Industrial risk
- Management risk
- Financial flexibility
- Credibility
- Competitiveness
- Operating
- Class : bankruptcy, non-bankruptcy (target variable)

Further more the variables information defined into 3 groups:

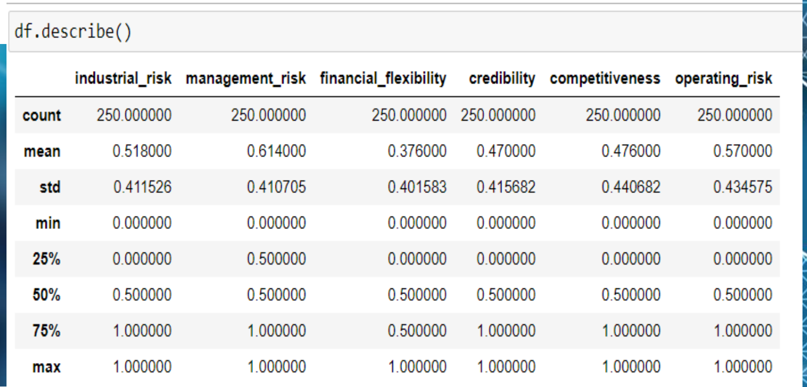
- Low risk = 0
- Medium risk = 0.5
- High risk = 1

EDA



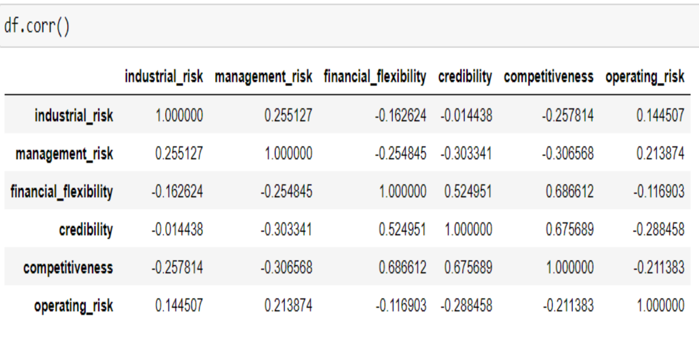
```
df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 250 entries, 0 to 249
Data columns (total 7 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   industrial_risk        250 non-null    float64
 1    management_risk       250 non-null    float64
 2    financial_flexibility 250 non-null    float64
 3    credibility           250 non-null    float64
 4    competitiveness       250 non-null    float64
 5    operating_risk        250 non-null    float64
 6    class                 250 non-null    object
dtypes: float64(6), object(1)
memory usage: 13.8+ KB
```



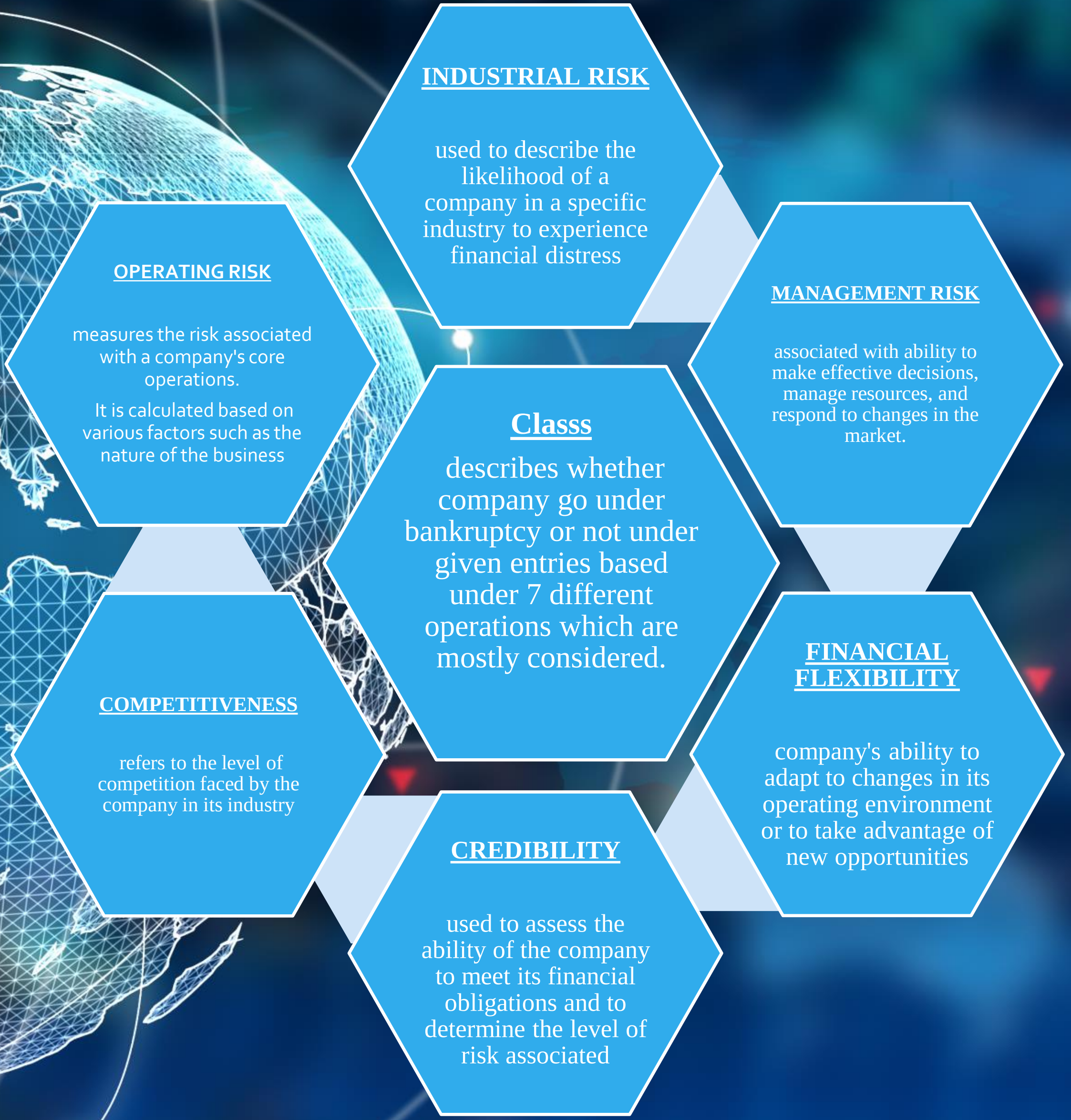
df.describe()

	industrial_risk	management_risk	financial_flexibility	credibility	competitiveness	operating_risk
count	250.000000	250.000000	250.000000	250.000000	250.000000	250.000000
mean	0.518000	0.614000	0.376000	0.470000	0.476000	0.570000
std	0.411526	0.410705	0.401583	0.415682	0.440682	0.434575
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	0.000000	0.500000	0.000000	0.000000	0.000000	0.000000
50%	0.500000	0.500000	0.500000	0.500000	0.500000	0.500000
75%	1.000000	1.000000	0.500000	1.000000	1.000000	1.000000
max	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000



df.corr()

	industrial_risk	management_risk	financial_flexibility	credibility	competitiveness	operating_risk
industrial_risk	1.000000	0.255127	-0.162624	-0.014438	-0.257814	0.144507
management_risk	0.255127	1.000000	-0.254845	-0.303341	-0.306568	0.213874
financial_flexibility	-0.162624	-0.254845	1.000000	0.524951	0.686612	-0.116903
credibility	-0.014438	-0.303341	0.524951	1.000000	0.675689	-0.288458
competitiveness	-0.257814	-0.306568	0.686612	0.675689	1.000000	-0.211383
operating_risk	0.144507	0.213874	-0.116903	-0.288458	-0.211383	1.000000



INDUSTRIAL RISK

used to describe the likelihood of a company in a specific industry to experience financial distress

OPERATING RISK

measures the risk associated with a company's core operations.

It is calculated based on various factors such as the nature of the business

MANAGEMENT RISK

associated with ability to make effective decisions, manage resources, and respond to changes in the market.

Classs

describes whether company go under bankruptcy or not under given entries based under 7 different operations which are mostly considered.

COMPETITIVENESS

refers to the level of competition faced by the company in its industry

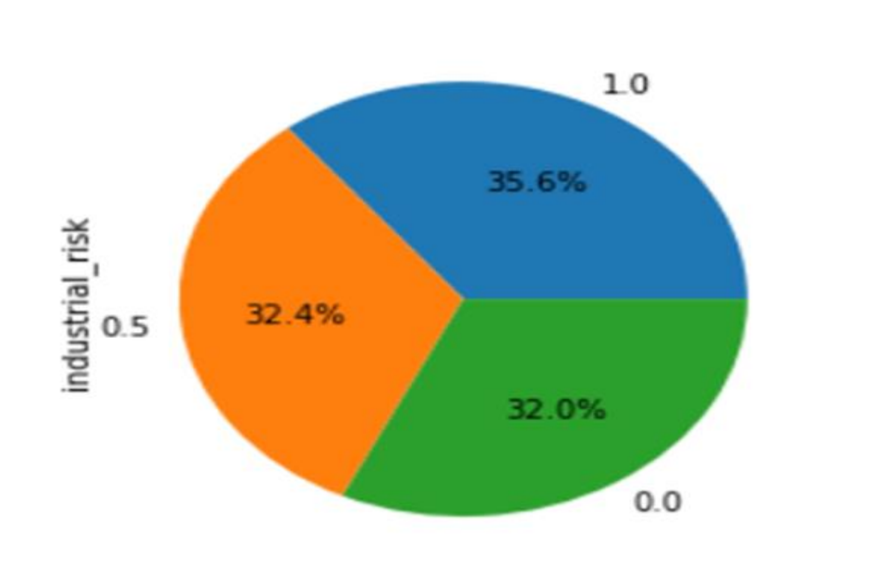
FINANCIAL FLEXIBILITY

company's ability to adapt to changes in its operating environment or to take advantage of new opportunities

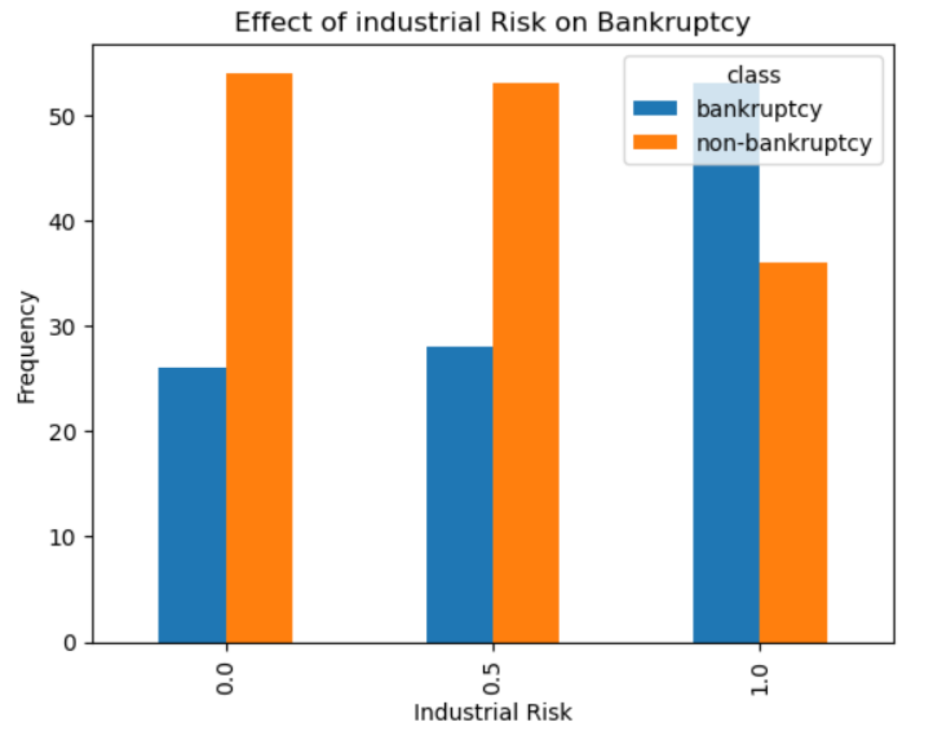
CREDIBILITY

used to assess the ability of the company to meet its financial obligations and to determine the level of risk associated

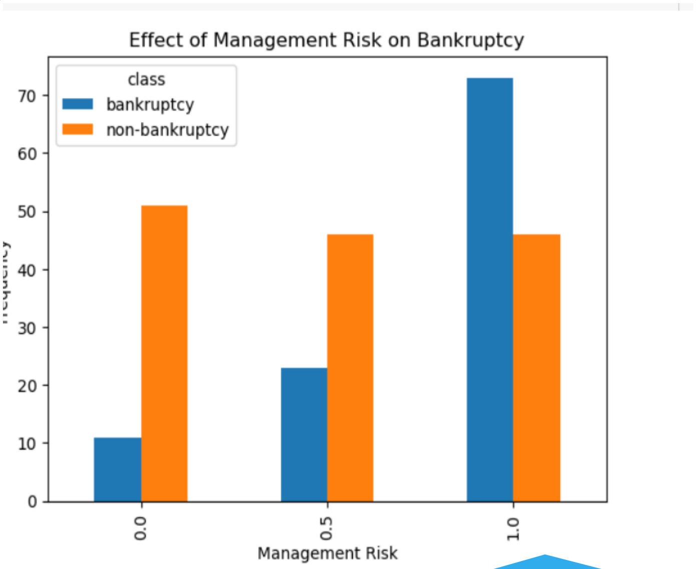
DATA VISUALIZATION



The illustration describes the value counts of an industrial risk and it shows most of the part goes under bankruptcy with 35.6%.

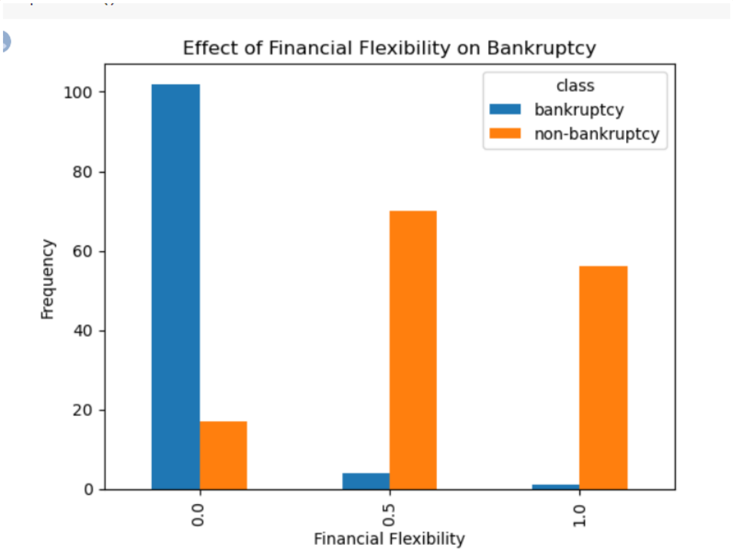


The plot shows the effect of industrial risk on bankruptcy and it illustrates risk of bankruptcy is more when the given information is classed as 1 comparing to rest division.

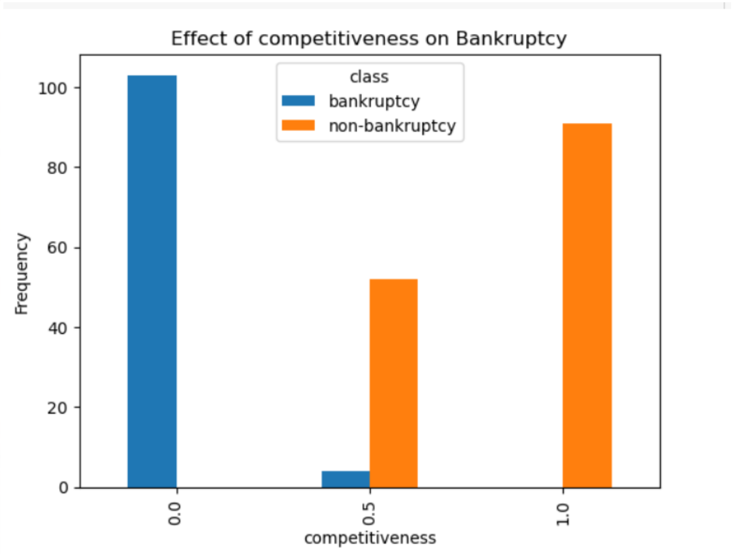


The plot shows the effect of management risk on bankruptcy and

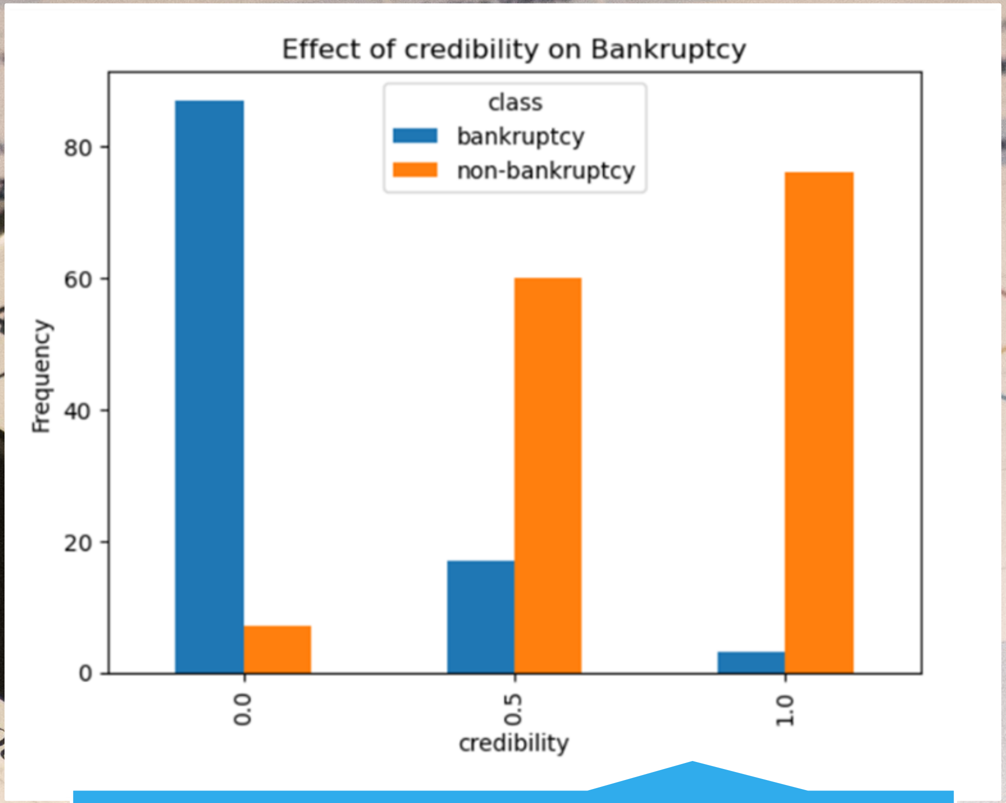
it illustrates risk of bankruptcy is more when the given information is classed as 1 comparing to rest division.



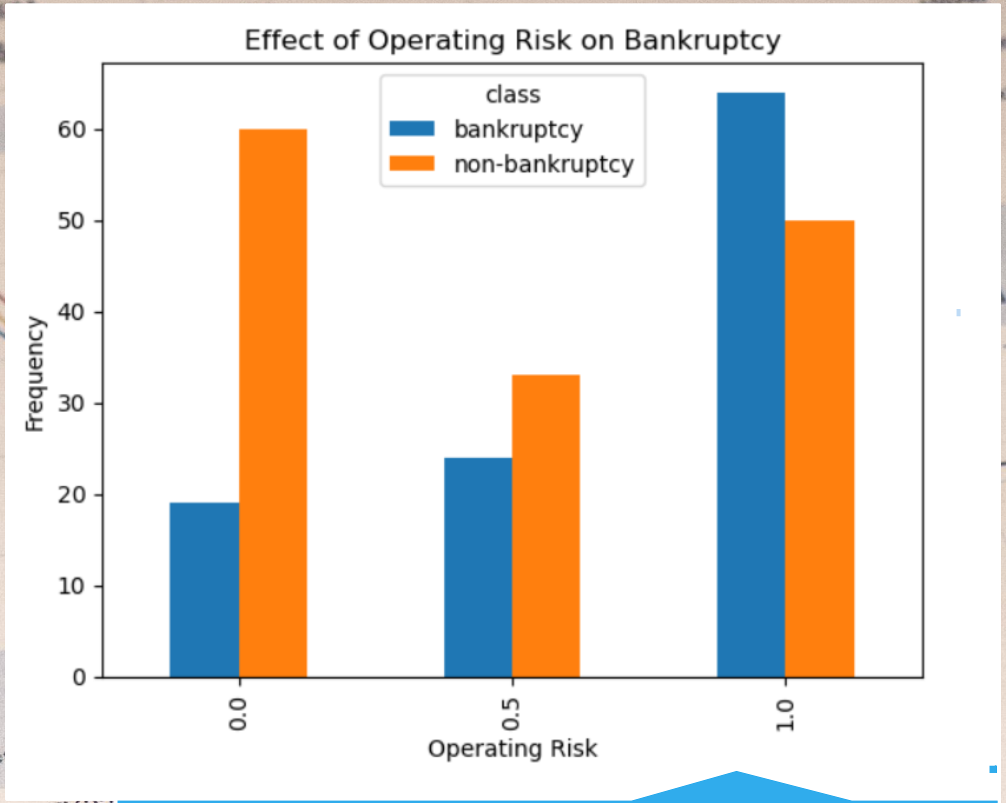
The plot shows under financial flexibility the bank doesn’t go any bankruptcy and it illustrates risk of bankruptcy is less according to the given information



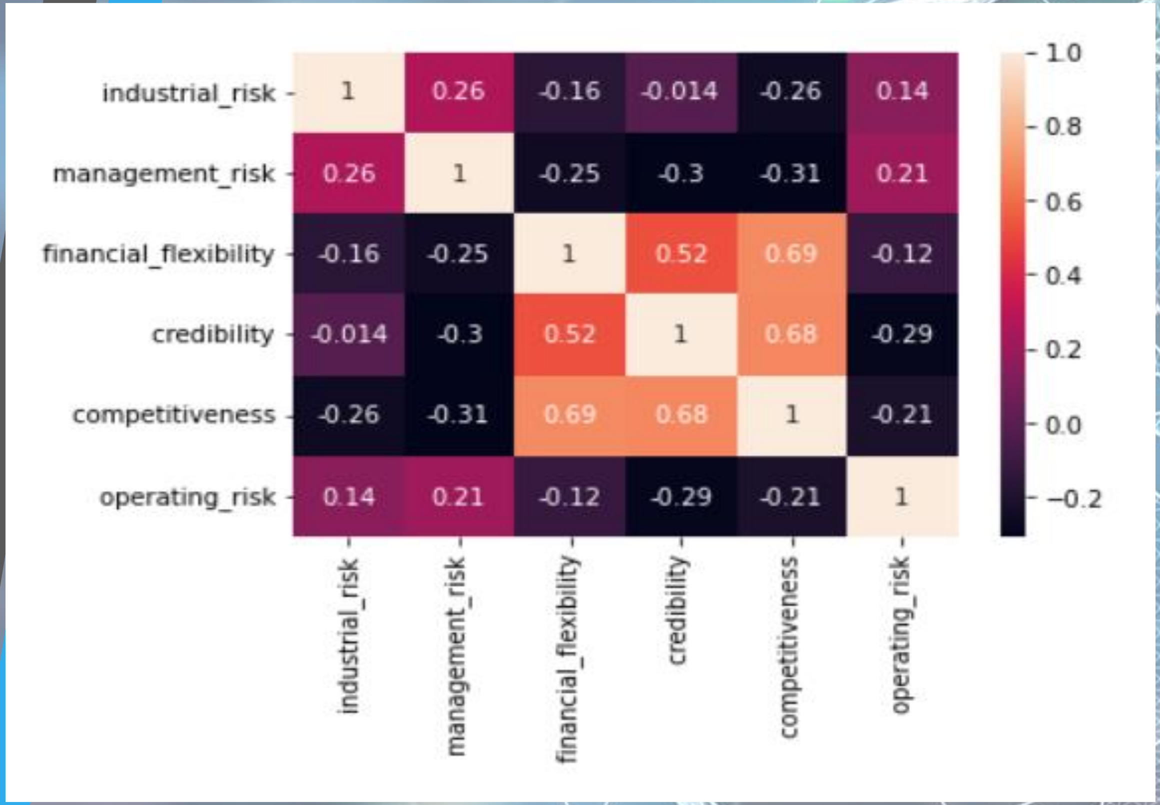
The plot shows under competitiveness the bank doesn’t go any bankruptcy and it illustrates risk of bankruptcy is less according to the given information



The plot shows under credibility the bank doesn’t go any bankruptcy and illustrates risk of bankruptcy is less according to the given information



The plot shows under operational risk the bank does go bankruptcy and illustrates risk of bankruptcy is more according to the given information

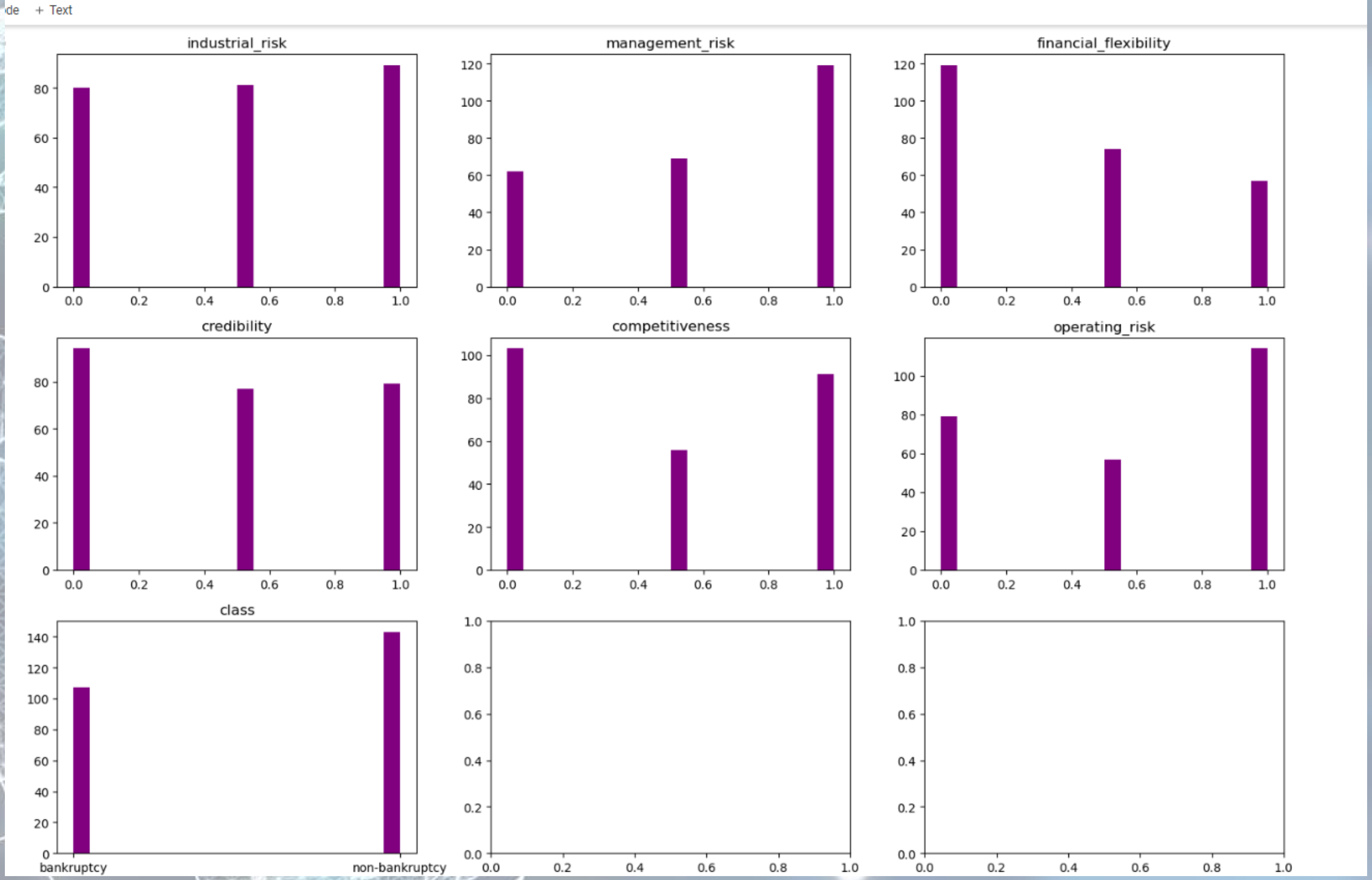


	industrial_risk	management_risk	financial_flexibility	credibility	competitiveness	operating_risk
industrial_risk	1	0.26	-0.16	-0.014	-0.26	0.14
management_risk	0.26	1	-0.25	-0.3	-0.31	0.21
financial_flexibility	-0.16	-0.25	1	0.52	0.69	-0.12
credibility	-0.014	-0.3	0.52	1	0.68	-0.29
competitiveness	-0.26	-0.31	0.69	0.68	1	-0.21
operating_risk	0.14	0.21	-0.12	-0.29	-0.21	1

Higher Financial Flexibility indicates company is flexible to adjust with changing market and operating condition. Companies with Higher Financial Flexibility are less prone to go bankrupt

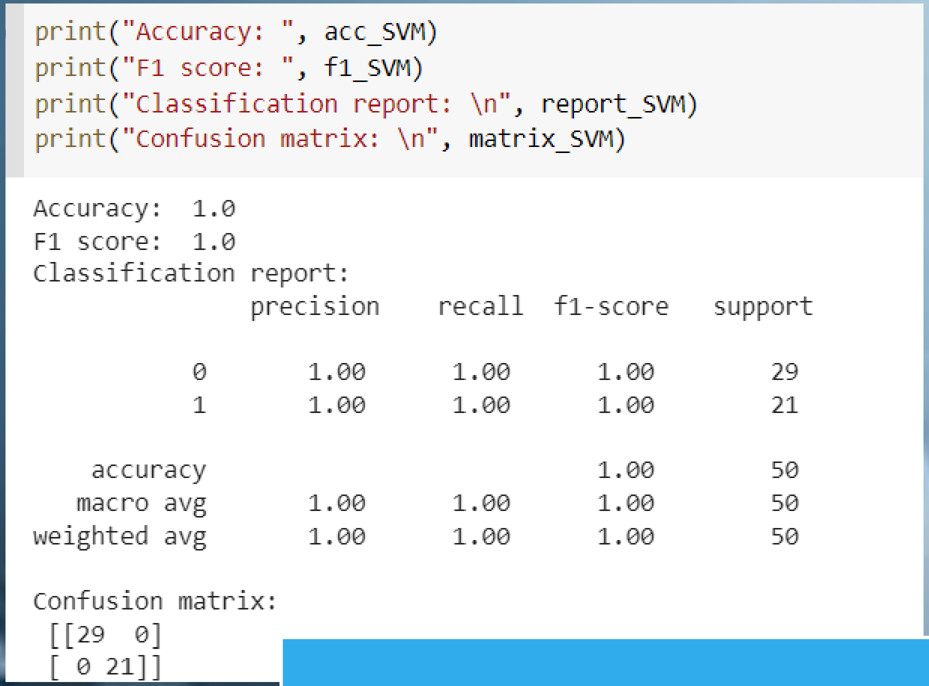
Financial flexibility is very much important to compete in a market since cash flow, cash position and level of debt plays major role in competing with other players in the market.

The plots shows the illustrations of 7 different histograms of each variables which were calculated during the finding out result whether companies are under bankruptcy or non bankruptcy.



MODEL BUILDING

The train and test split was with the size of 0.8 and 0.2 with random state 42.
Furtherly eight models are built under this train and test size along with fixed random state



```
print("Accuracy: ", acc_SVM)
print("F1 score: ", f1_SVM)
print("Classification report: \n", report_SVM)
print("Confusion matrix: \n", matrix_SVM)
```

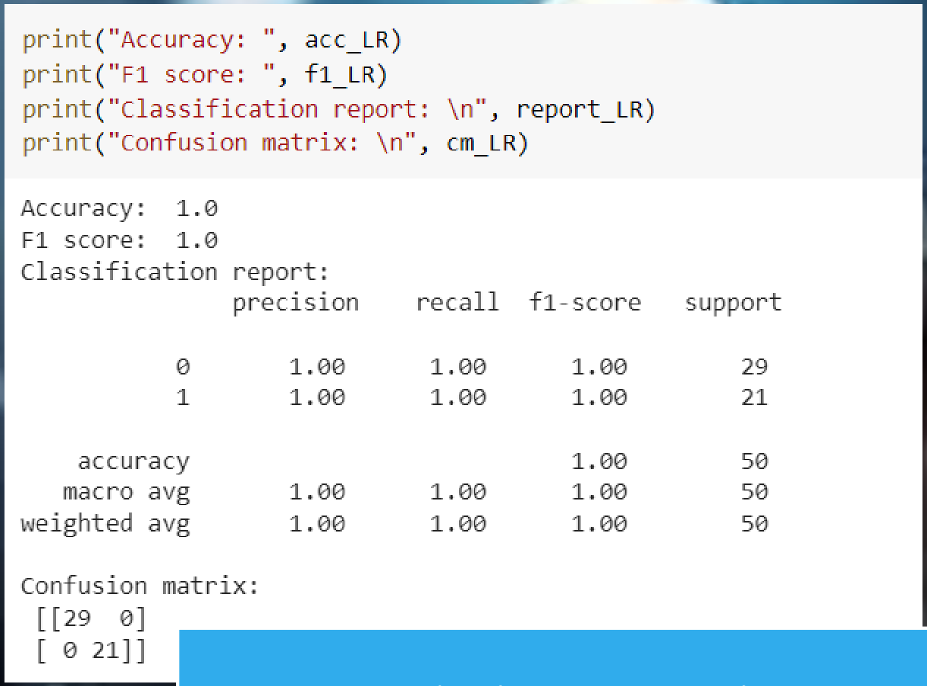
```
Accuracy:  1.0
F1 score:  1.0
Classification report:
               precision    recall  f1-score   support

           0       1.00      1.00      1.00        29
           1       1.00      1.00      1.00        21

    accuracy                           1.00        50
   macro avg       1.00      1.00      1.00        50
weighted avg       1.00      1.00      1.00        50

Confusion matrix:
 [[29  0]
 [ 0 21]]
```

Support vector machine



```
print("Accuracy: ", acc_LR)
print("F1 score: ", f1_LR)
print("Classification report: \n", report_LR)
print("Confusion matrix: \n", cm_LR)
```

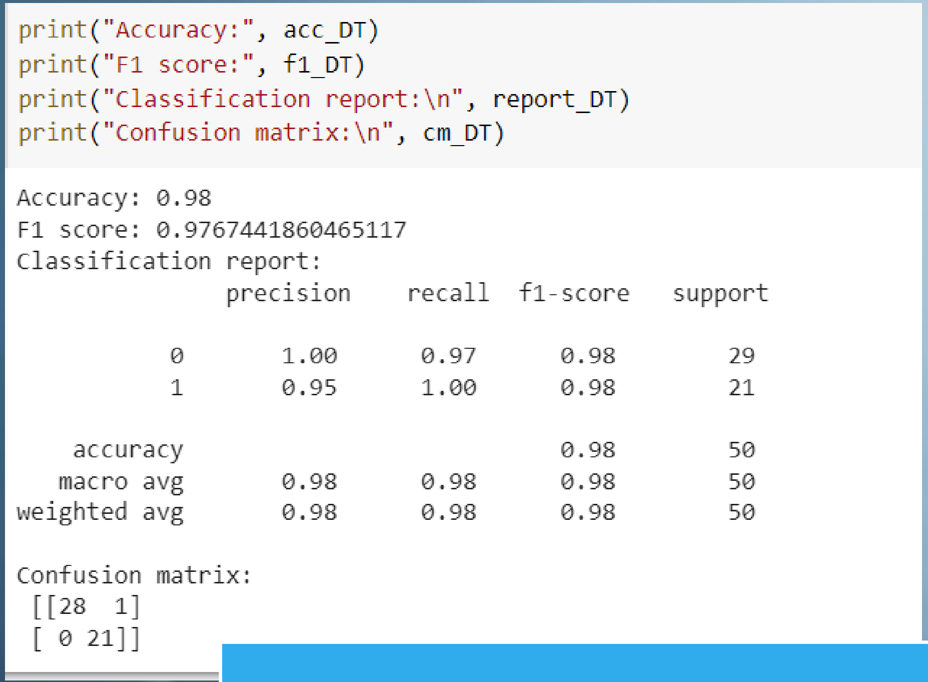
```
Accuracy:  1.0
F1 score:  1.0
Classification report:
               precision    recall  f1-score   support

           0       1.00      1.00      1.00        29
           1       1.00      1.00      1.00        21

    accuracy                           1.00        50
   macro avg       1.00      1.00      1.00        50
weighted avg       1.00      1.00      1.00        50

Confusion matrix:
 [[29  0]
 [ 0 21]]
```

Logistic regression



```
print("Accuracy:", acc_DT)
print("F1 score:", f1_DT)
print("Classification report:\n", report_DT)
print("Confusion matrix:\n", cm_DT)
```

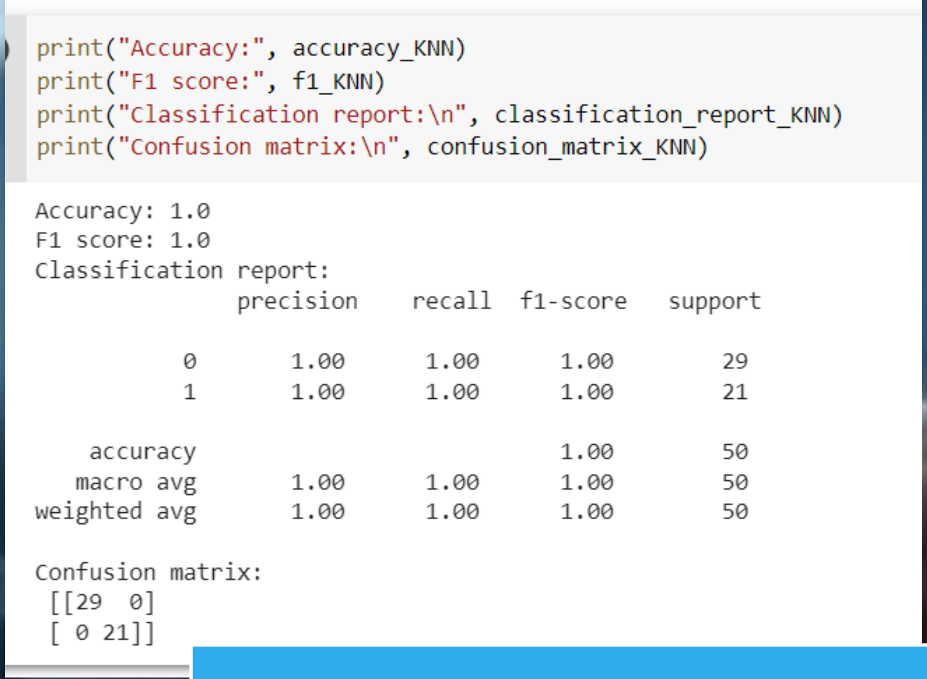
```
Accuracy: 0.98
F1 score: 0.9767441860465117
Classification report:
               precision    recall  f1-score   support

           0       1.00      0.97      0.98        29
           1       0.95      1.00      0.98        21

    accuracy                           0.98        50
   macro avg       0.98      0.98      0.98        50
weighted avg       0.98      0.98      0.98        50

Confusion matrix:
 [[28  1]
 [ 0 21]]
```

Decision tree



```
print("Accuracy:", accuracy_KNN)
print("F1 score:", f1_KNN)
print("Classification report:\n", classification_report_KNN)
print("Confusion matrix:\n", confusion_matrix_KNN)
```

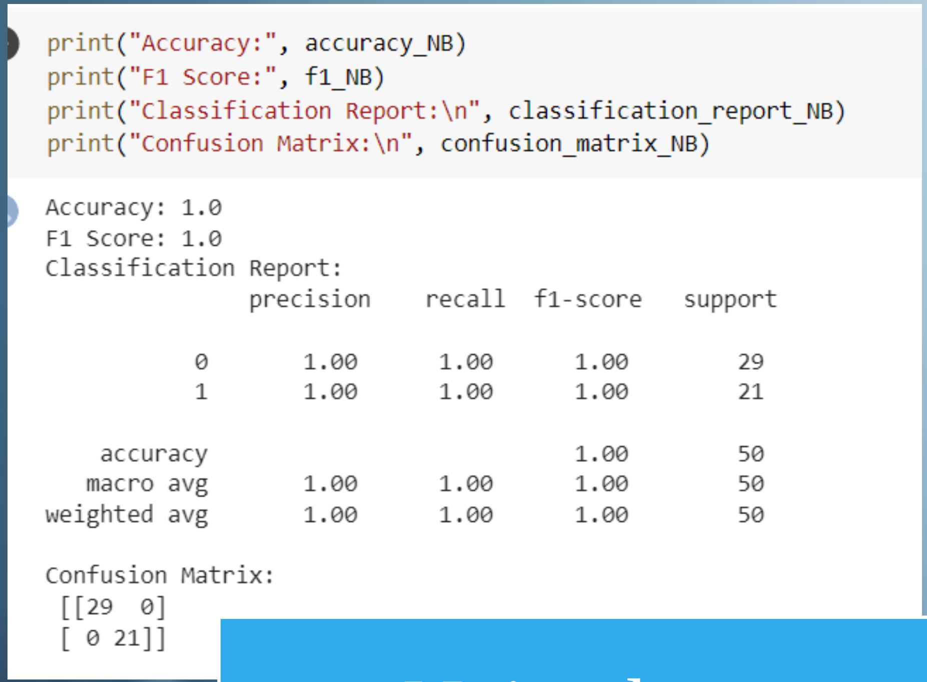
```
Accuracy: 1.0
F1 score: 1.0
Classification report:
               precision    recall  f1-score   support

           0       1.00      1.00      1.00        29
           1       1.00      1.00      1.00        21

    accuracy                           1.00        50
   macro avg       1.00      1.00      1.00        50
weighted avg       1.00      1.00      1.00        50

Confusion matrix:
 [[29  0]
 [ 0 21]]
```

KNN



```
print("Accuracy:", accuracy_NB)
print("F1 Score:", f1_NB)
print("Classification Report:\n", classification_report_NB)
print("Confusion Matrix:\n", confusion_matrix_NB)
```

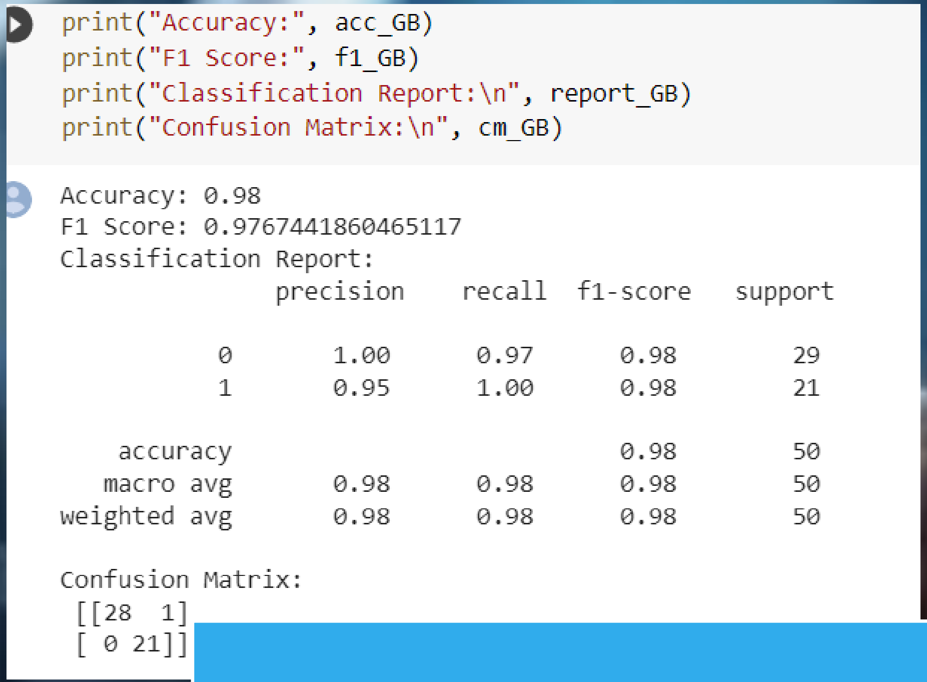
```
Accuracy: 1.0
F1 Score: 1.0
Classification Report:
               precision    recall  f1-score   support

           0       1.00      1.00      1.00        29
           1       1.00      1.00      1.00        21

    accuracy                           1.00        50
   macro avg       1.00      1.00      1.00        50
weighted avg       1.00      1.00      1.00        50

Confusion Matrix:
 [[29  0]
 [ 0 21]]
```

Naive bayes



```
print("Accuracy:", acc_GB)
print("F1 Score:", f1_GB)
print("Classification Report:\n", report_GB)
print("Confusion Matrix:\n", cm_GB)
```

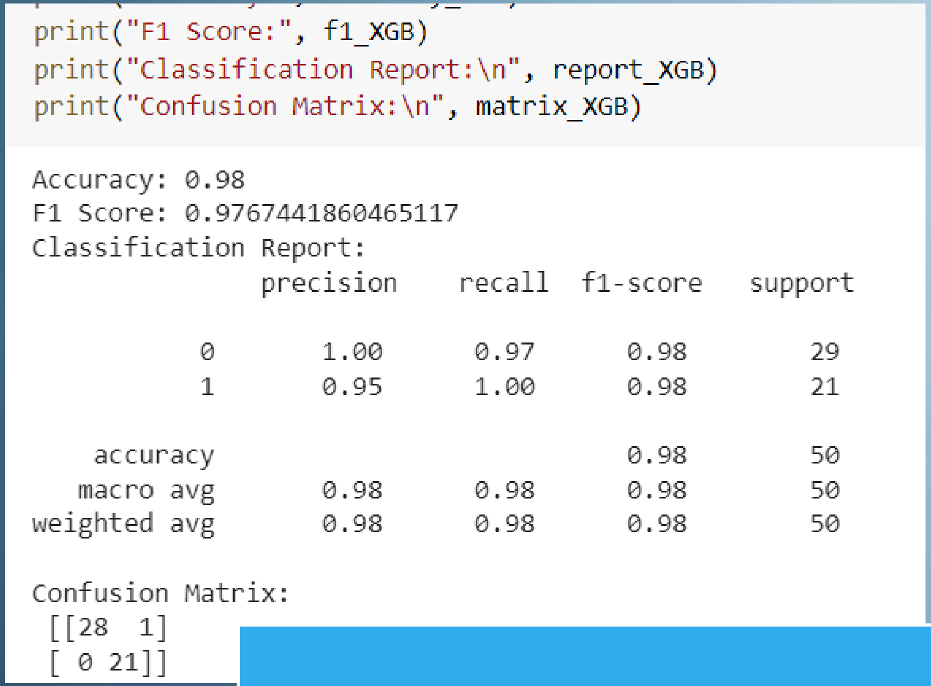
```
Accuracy: 0.98
F1 Score: 0.9767441860465117
Classification Report:
               precision    recall  f1-score   support

           0       1.00      0.97      0.98        29
           1       0.95      1.00      0.98        21

    accuracy                           0.98        50
   macro avg       0.98      0.98      0.98        50
weighted avg       0.98      0.98      0.98        50

Confusion Matrix:
 [[28  1]
 [ 0 21]]
```

Gradient Boosting



```
print("F1 Score:", f1_XGB)
print("Classification Report:\n", report_XGB)
print("Confusion Matrix:\n", matrix_XGB)
```

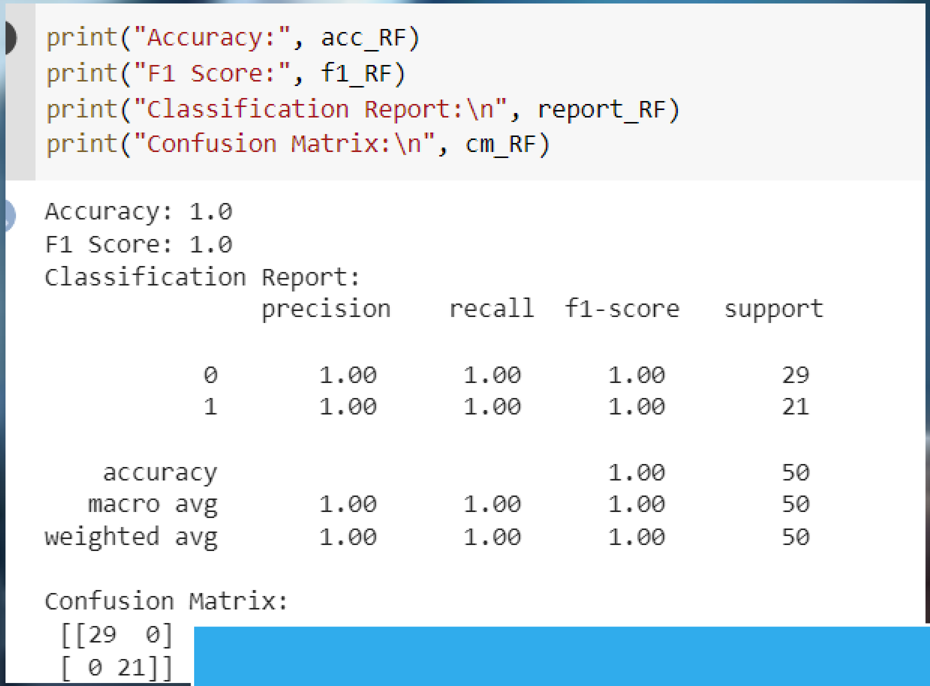
```
Accuracy: 0.98
F1 Score: 0.9767441860465117
Classification Report:
               precision    recall  f1-score   support

           0       1.00      0.97      0.98        29
           1       0.95      1.00      0.98        21

    accuracy                           0.98        50
   macro avg       0.98      0.98      0.98        50
weighted avg       0.98      0.98      0.98        50

Confusion Matrix:
 [[28  1]
 [ 0 21]]
```

XG Boost



```
print("Accuracy:", acc_RF)
print("F1 Score:", f1_RF)
print("Classification Report:\n", report_RF)
print("Confusion Matrix:\n", cm_RF)
```

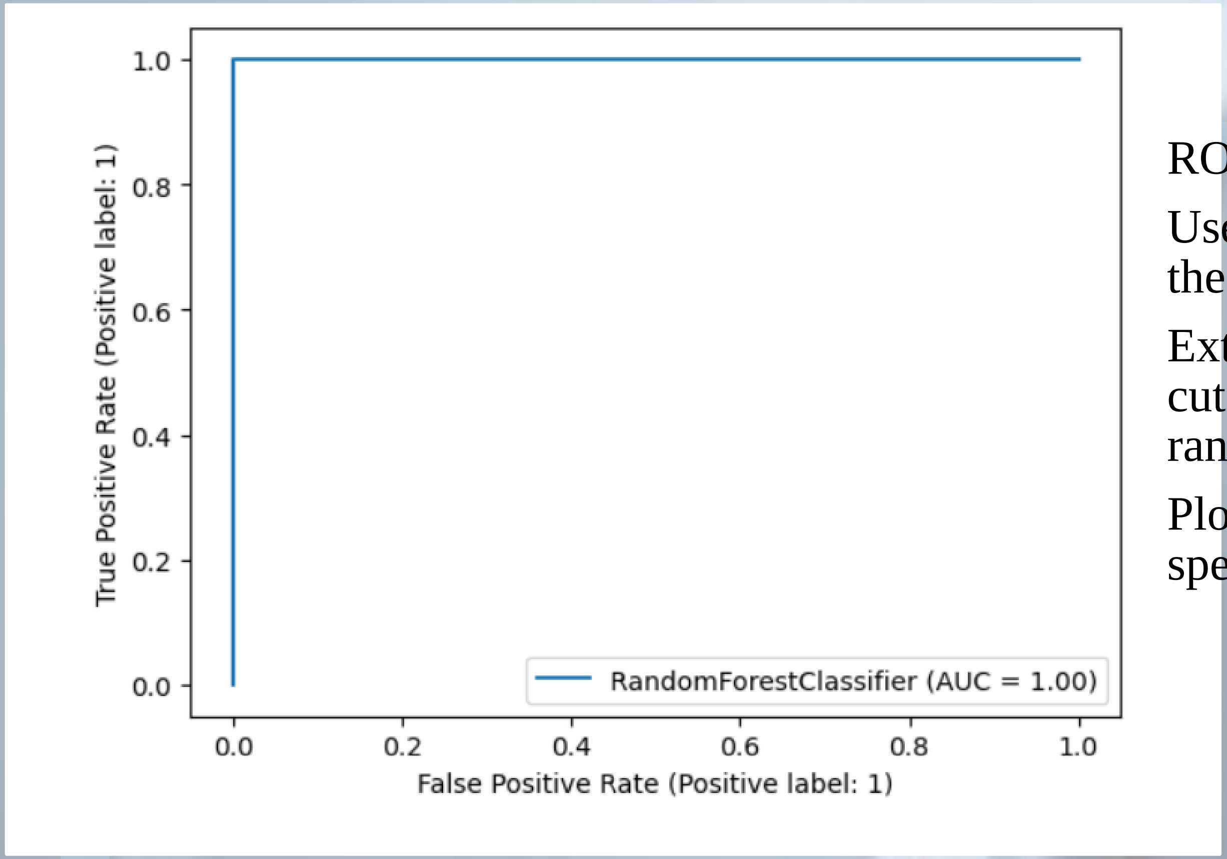
```
Accuracy: 1.0
F1 Score: 1.0
Classification Report:
               precision    recall  f1-score   support

           0       1.00      1.00      1.00        29
           1       1.00      1.00      1.00        21

    accuracy                           1.00        50
   macro avg       1.00      1.00      1.00        50
weighted avg       1.00      1.00      1.00        50

Confusion Matrix:
 [[29  0]
 [ 0 21]]
```

Random Forest



ROC – Receiver operating characteristic curve

Used to understand the overall performance of the model.

Extending 2 by 2 table idea rather than single cut off selection, we can examine the full range of cut off value from 0 – 1.

Plotting the pairs of sensitivity vs 1 - specificity .

The above ROC illustration describes that the model is working excellent under the given test size.

MODEL EVALUATION

Random Forest :

In above business problem recall is more important than accuracy and precision

Because we are more concerned about false negatives than false positives.

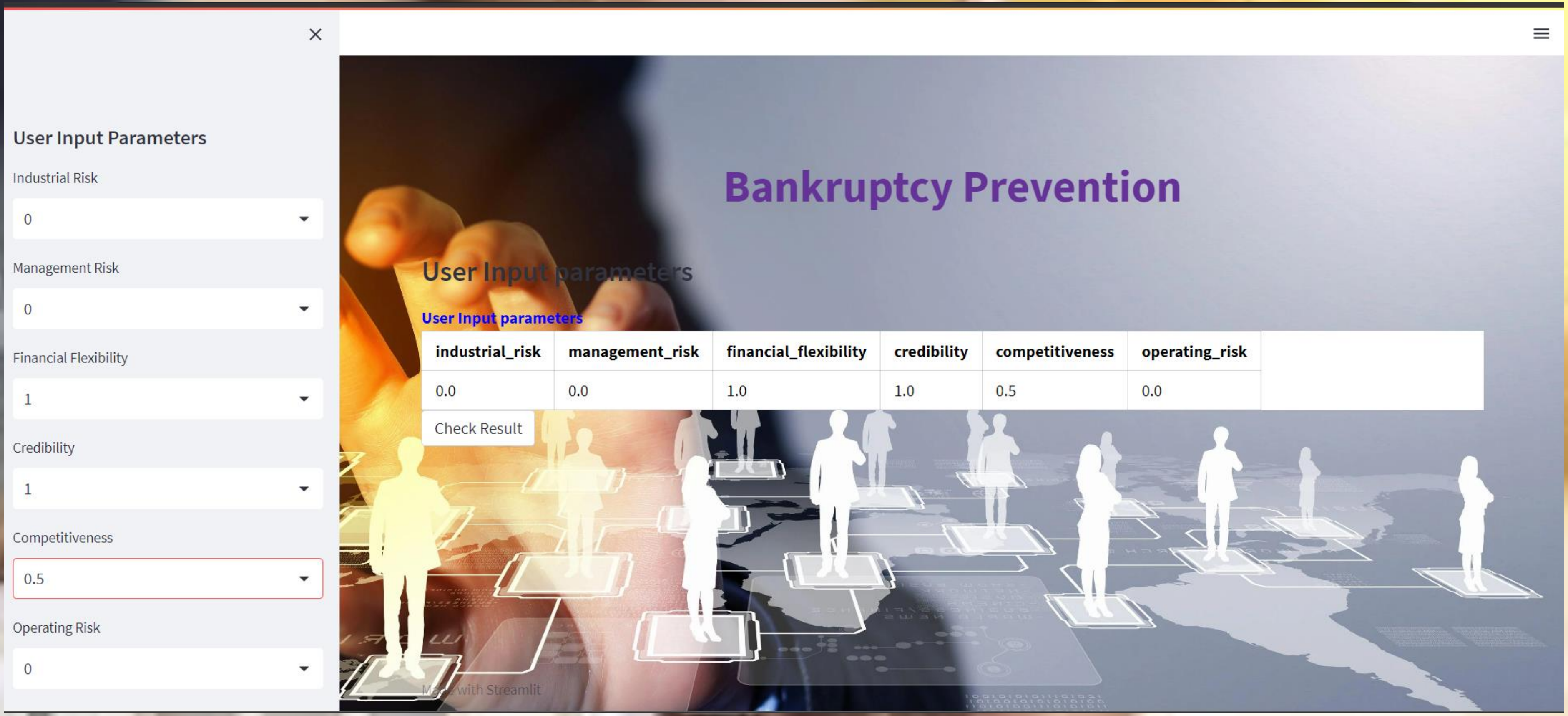
But all models are giving same values for all evaluation metric.

So we decided to go with RANDOM FOREST because it reduces overfitting at some extent and it gives more generalized result.

MODEL DEPLOYMENT

Streamlit is an open-source app framework in python language. It helps us create beautiful web apps for data science and machine learning in a little time.

It is compatible with major python libraries such as scikit-learn, keras, PyTorch, latex, numpy, pandas, matplotlib, etc.



User Input Parameters

Industrial Risk: 0

Management Risk: 0

Financial Flexibility: 1

Credibility: 1

Competitiveness: 0.5

Operating Risk: 0

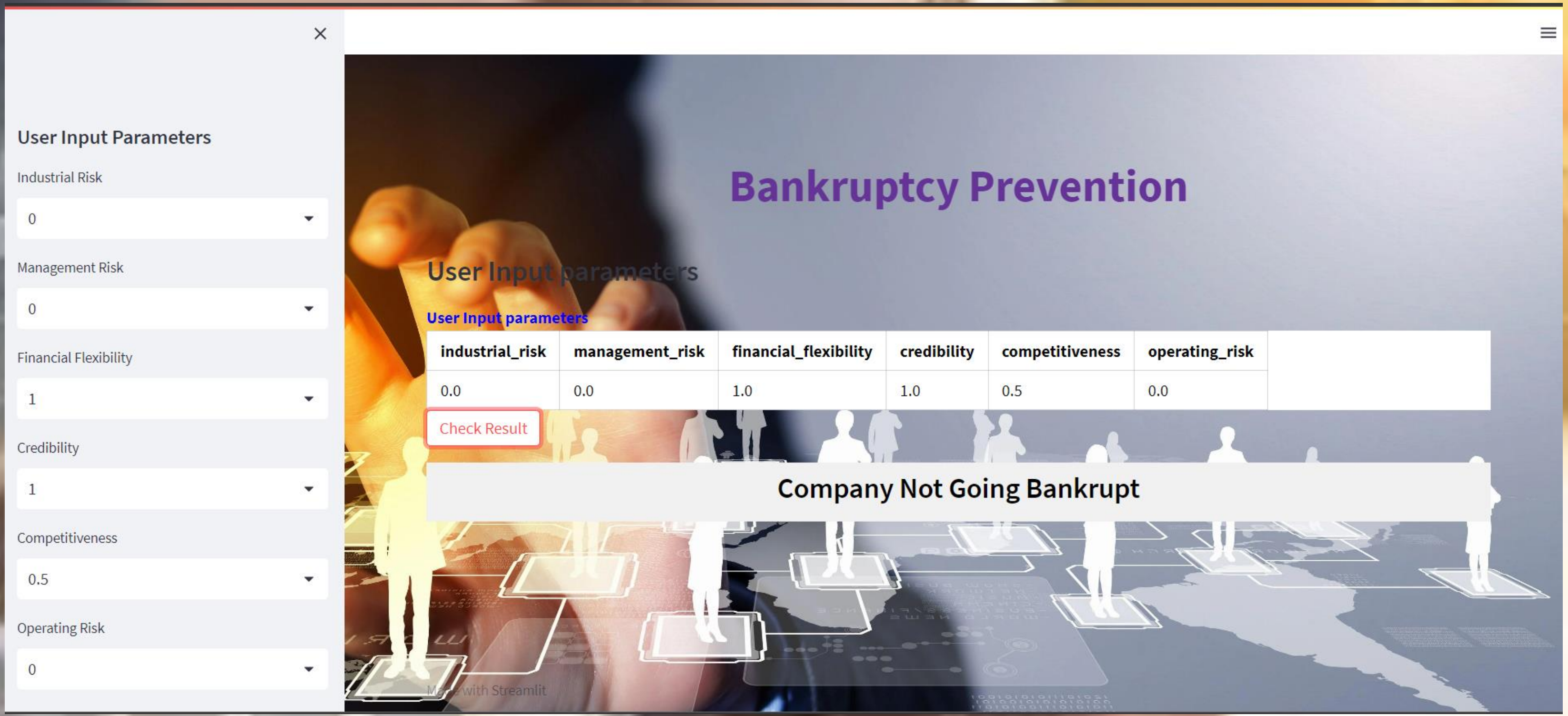
Bankruptcy Prevention

User Input parameters

User Input parameters

industrial_risk	management_risk	financial_flexibility	credibility	competitiveness	operating_risk
0.0	0.0	1.0	1.0	0.5	0.0

RESULT



User Input Parameters

Industrial Risk: 0

Management Risk: 0

Financial Flexibility: 1

Credibility: 1

Competitiveness: 0.5

Operating Risk: 0

Bankruptcy Prevention

User Input parameters

User Input parameters

industrial_risk	management_risk	financial_flexibility	credibility	competitiveness	operating_risk
0.0	0.0	1.0	1.0	0.5	0.0

Company Not Going Bankrupt