BANKRUPTCY

BANKRUPTCY PREVENTION PROJECT

P_334 TEAN MEMBER

1 SURESH RATHOD
2 DIPALI PATIL
3 SAURABH MESHRAM
4 HARSHAL BHENDARE
5 AMEY ZENDE
6 HARSHA MALLISETTY
7 GANDAMALLA ARAVIND

MENTOR NAME

ROHIT MISHRA

DATE : 06/02/2024

BANKRUPTCY PREVENTION PROJECT

CONTENT

- Business Objective
- Project Architecture
- Data Collection and Details
- Exploratory Data Analysis
- Visualization
- Modeling
- Evaluating
- Deployment

BANKRUPTCY

Objectives of Bankruptcy

01 To maximize the return to creditors.

02 To restore the debtor company to profitable trading where possible.

03 To establish a fair and equitable system for the ranking of claims and the distribution of assets among creditors.

04 To provide a mechanism by which the causes of failure can be identified and those guilty of mismanagement brought to book.

BANKRUPTCY

Types of Bankruptcy

BANKRUPTCY

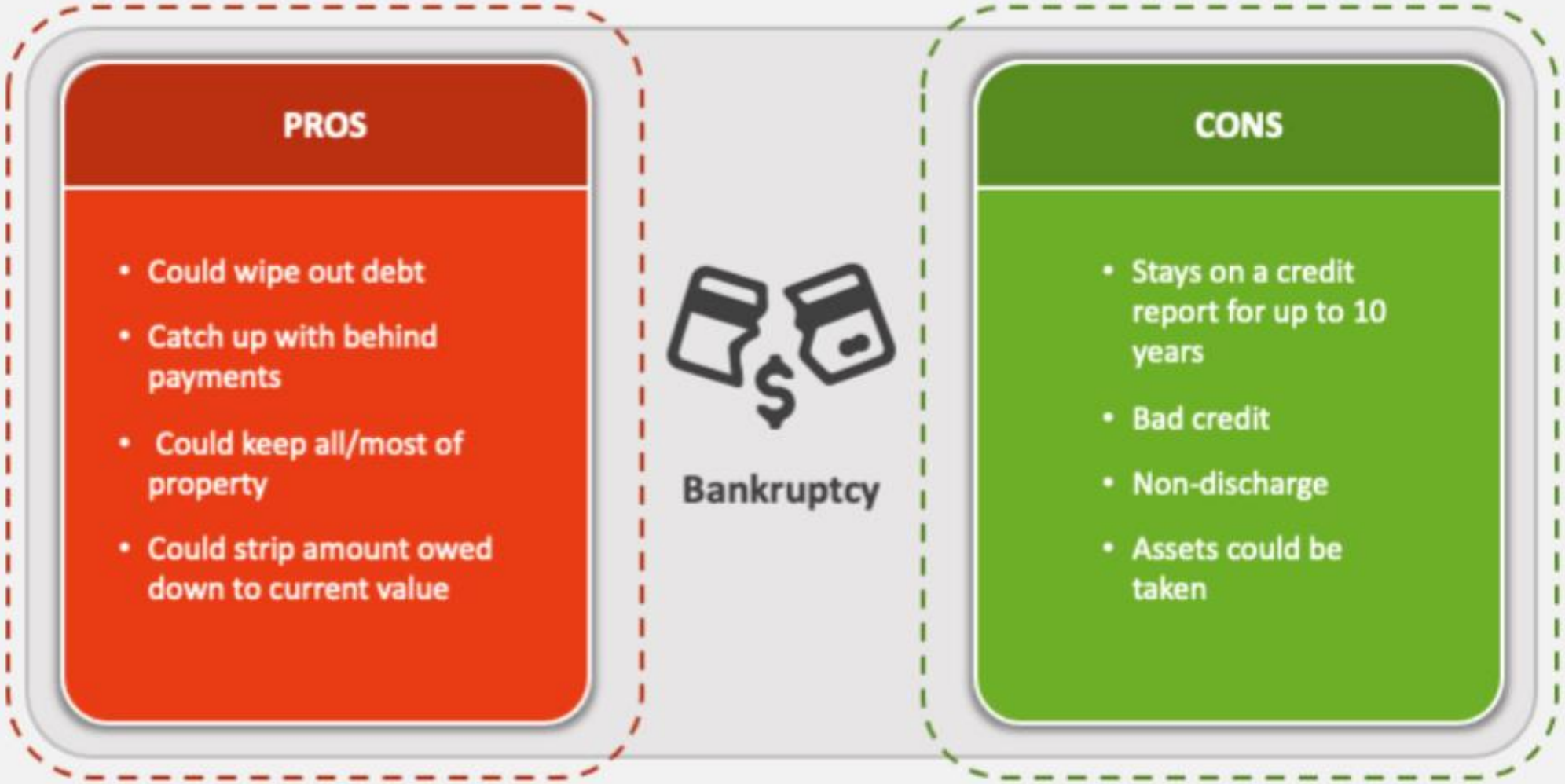
Risk Identification in Bankruptcy

BANKRUPTCY

Misconceptions about Bankruptcy

BANKRUPTCY

Pros & Cons of Bankruptcy



Data Set Includes The Following Variables

1. industrial_risk: 0=low risk, 0.5=medium risk, 1=high risk.

2. management_risk: 0=low risk, 0.5=medium risk, 1=high risk.

3. financial flexibility: 0=low flexibility, 0.5=medium flexibility, 1=high flexibility.

4. credibility: 0=low credibility, 0.5=medium credibility, 1=high credibility.

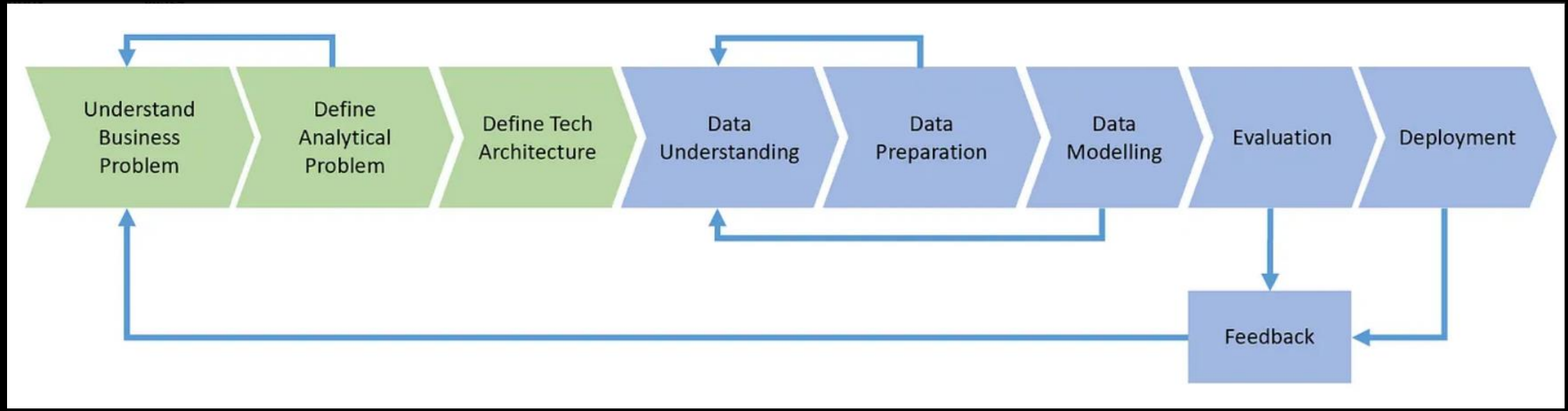
5. competitiveness: 0=low competitiveness, 0.5=medium competitiveness, 1=high

competitiveness.

6. operating_risk: 0=low risk, 0.5=medium risk, 1=high risk.

7. class: bankruptcy, non-bankruptcy (target variable).

PROJECT WORKFLOW



DATASET INFORMATION

```
bank.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
```

```
bank.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

```
bank.value_counts().sum()
```

```
250
```

EDA (Exploratory Data Analysis)

```
bank.corr()
```

	industrial_risk	management_risk	financial_flexibility	credibility	competitiveness	operating_risk	target
industrial_risk	1.000000	0.255127	-0.162624	-0.014438	-0.257814	0.144507	-0.227823
management_risk	0.255127	1.000000	-0.254845	-0.303341	-0.306568	0.213874	-0.370838
financial_flexibility	-0.162624	-0.254845	1.000000	0.524951	0.686612	-0.116903	0.751020
credibility	-0.014438	-0.303341	0.524951	1.000000	0.675689	-0.288458	0.755909
competitiveness	-0.257814	-0.306568	0.686612	0.675689	1.000000	-0.211383	0.899452
operating_risk	0.144507	0.213874	-0.116903	-0.288458	-0.211383	1.000000	-0.279786
target	-0.227823	-0.370838	0.751020	0.755909	0.899452	-0.279786	1.000000

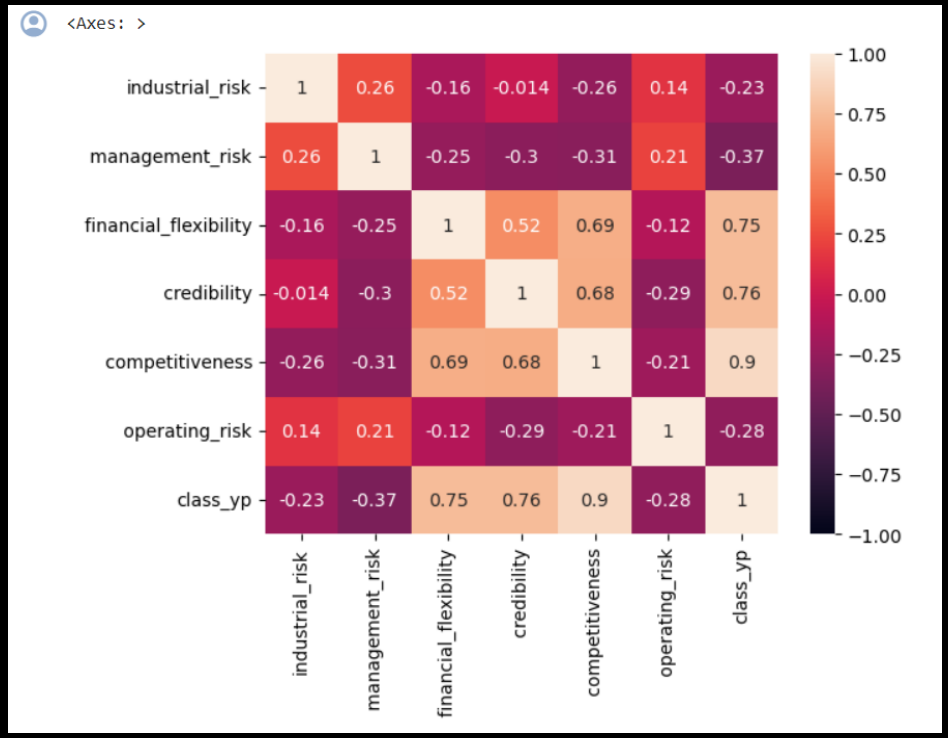
```
bank.isnull().sum()
```

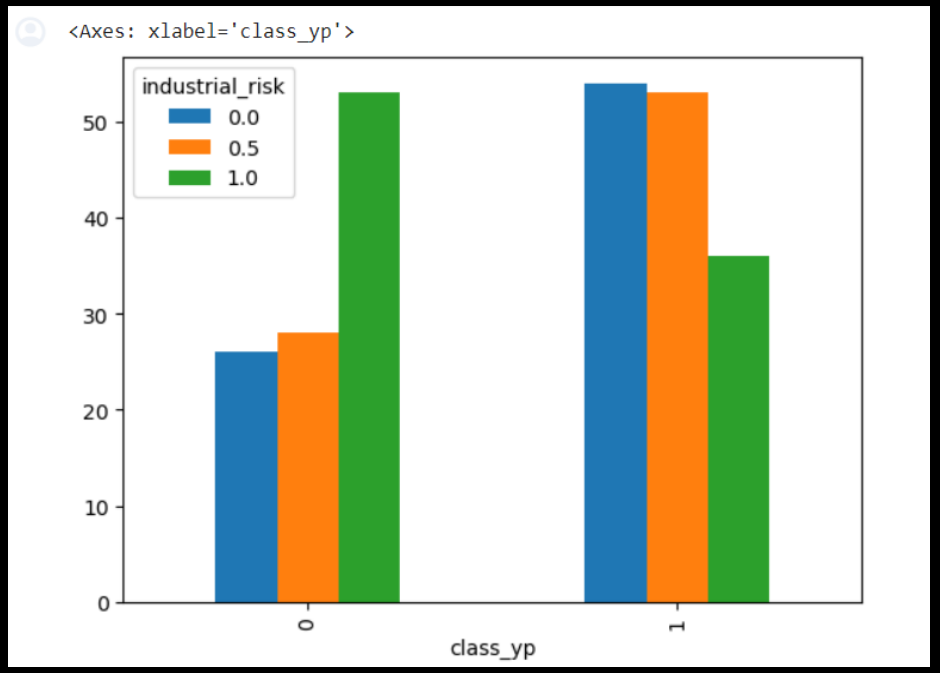
```
industrial_risk         0
 management_risk        0
 financial_flexibility  0
 credibility            0
 competitiveness        0
 operating_risk         0
 class                  0
dtype: int64
```

```
[ ] bank.drop(' class', axis=1, inplace=True)
```

```
[ ] le = LabelEncoder()
    bank['target'] = le.fit_transform(bank[' class'])
```

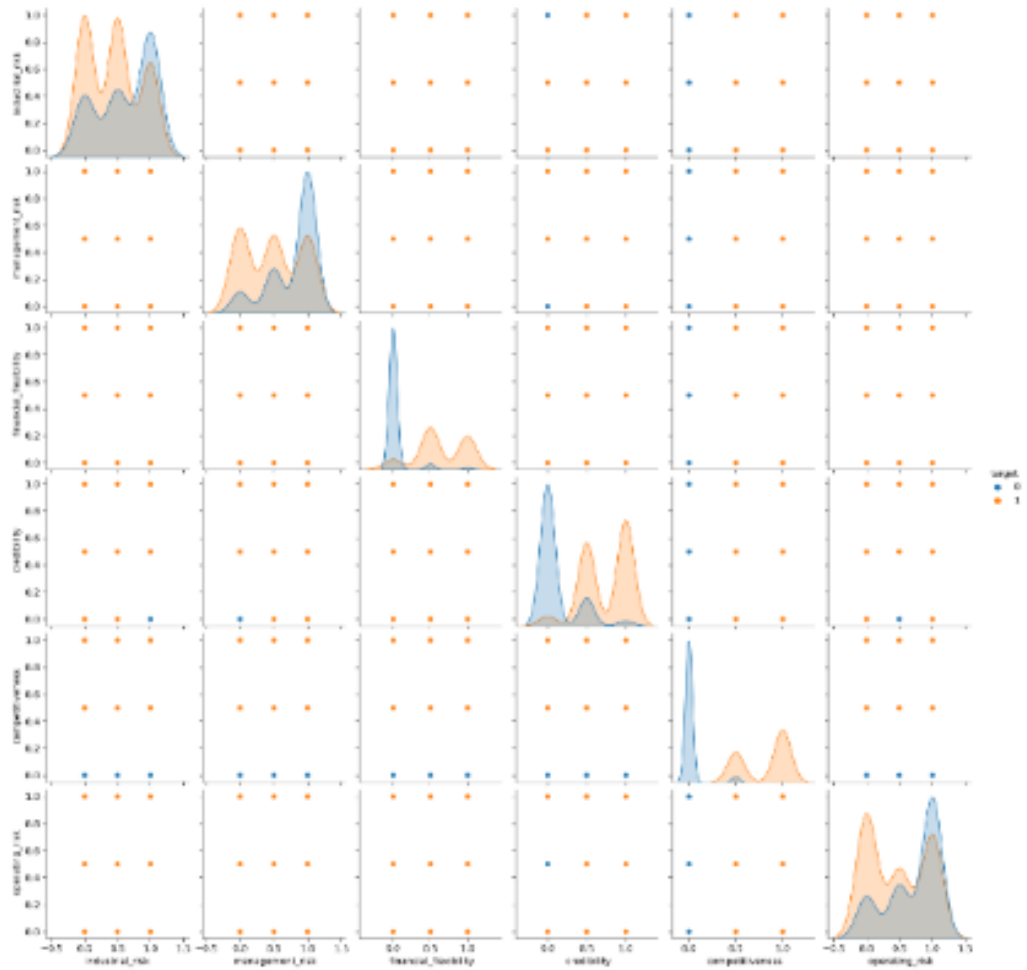
DATA VISUALIZATION

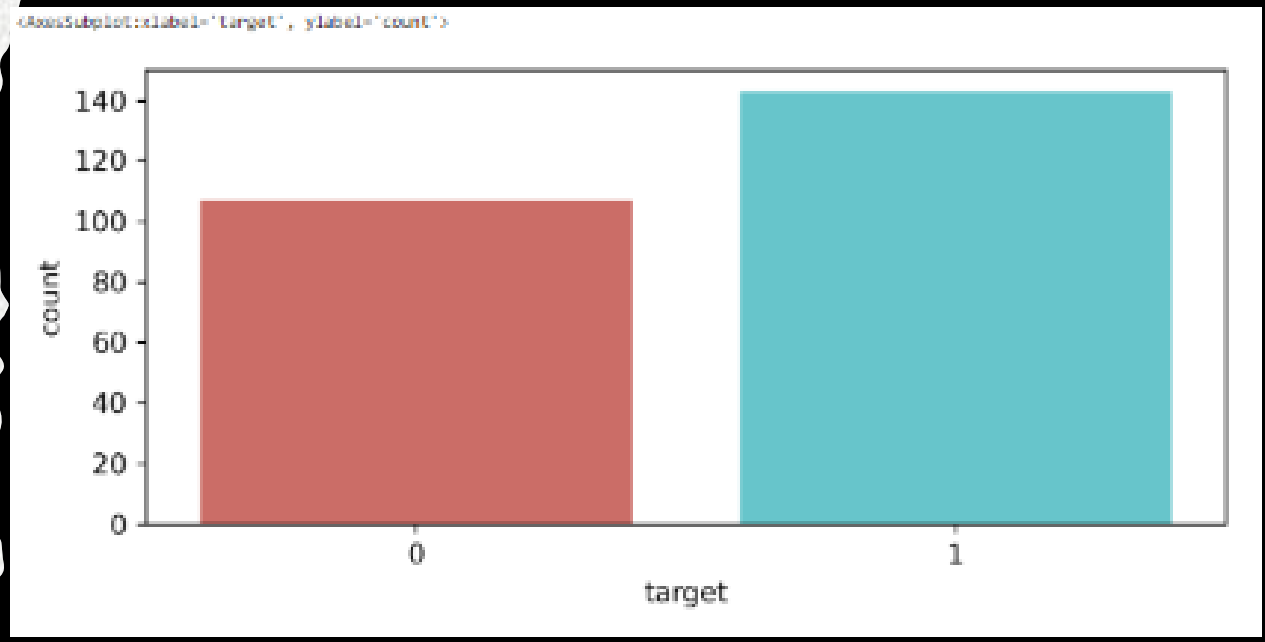




Scatter Plot

In [17]:
```
sns.pairplot(bank,hue ='target')    #Creating a pair plot for the numeric features in the DataFrame
plt.show()
```





MODEL BUILDING

1. LOGISTIC REGRESSION

```
[ ] from sklearn.linear_model import LogisticRegression

    lr=LogisticRegression()
```

```
    lr.fit(x_train,y_train)

    #Coefficients of features

    lr.coef_
```

```
array([[-0.424582  , -0.751384  ,  2.3109167 ,  2.24230331,  3.63350884,
        -0.45312639]])
```

2. KNN MODEL

```
[ ] from sklearn.neighbors import KNeighborsClassifier as knc
    import warnings
    warnings.filterwarnings("ignore")
```

To choose the K value

```
    import math
    math.sqrt(len(y_test))
```

```
8.660254037844387
```

3. Random Forest Classifier

```
[ ] from sklearn.ensemble import RandomForestClassifier
    from sklearn.model_selection import train_test_split
    from sklearn.metrics import accuracy_score
```

Create a Random Forest Classifier

```
[ ] rf= RandomForestClassifier(n_estimators=100, random_state=42)
    rf
```

```
RandomForestClassifier
RandomForestClassifier(random_state=42)
```

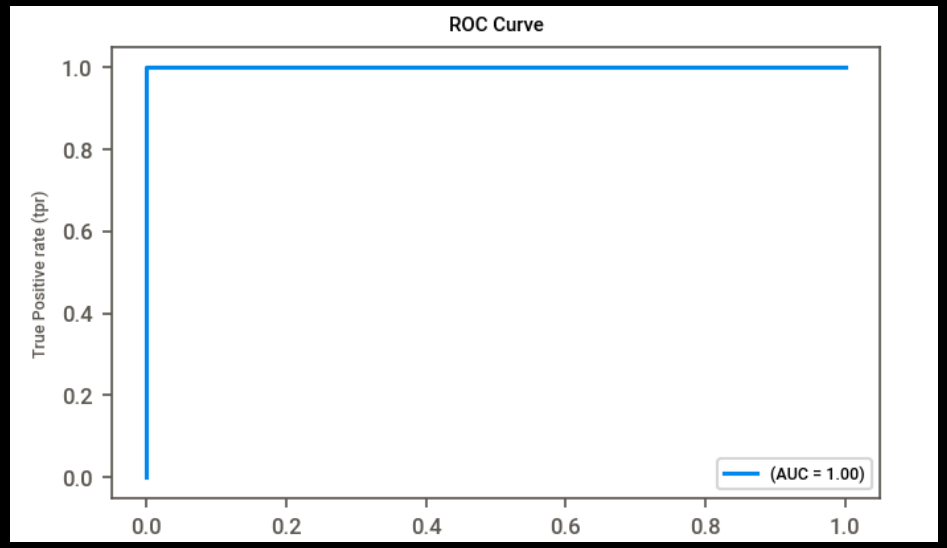
4. Support Vector Machine

```
[ ] from sklearn.svm import SVC
```

```
[ ] # Creating and training the model
    model_bankruptcy = SVC(kernel='linear').fit(x_train, y_train)
```

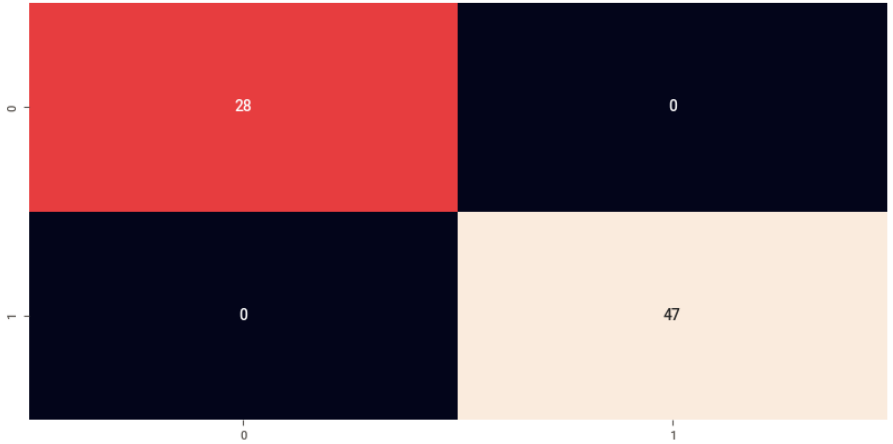
```
    # Making predictions on the test set
    pred_test_linear = model_bankruptcy.predict(x_test)
```

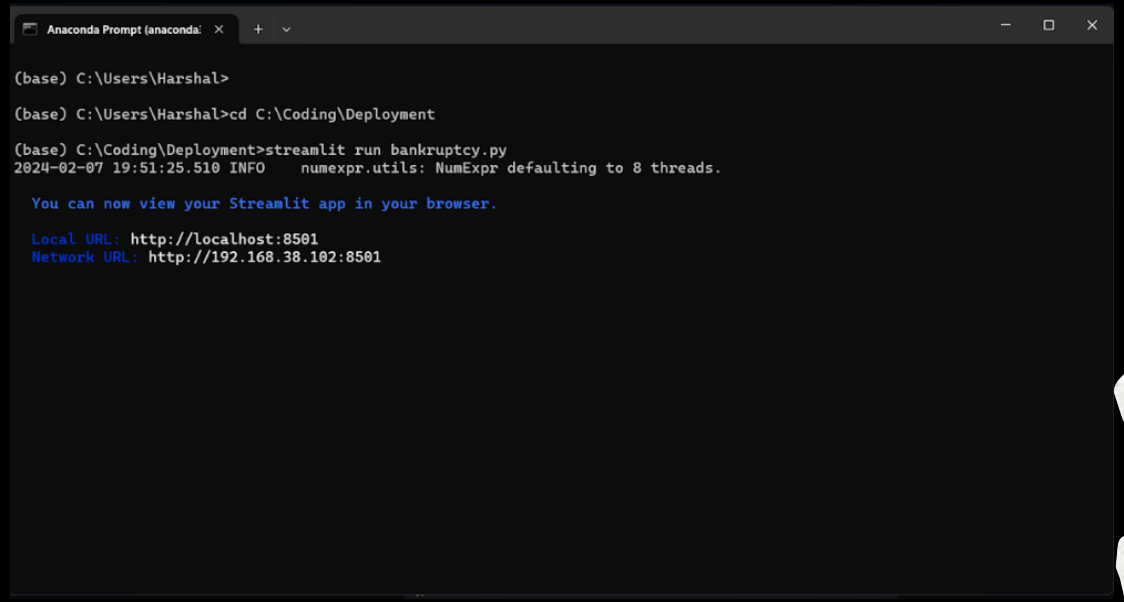
MODEL EVALUATION




```
sns.heatmap(confusion_matrix(y_test, y_pred),annot= True, cbar= False)
```

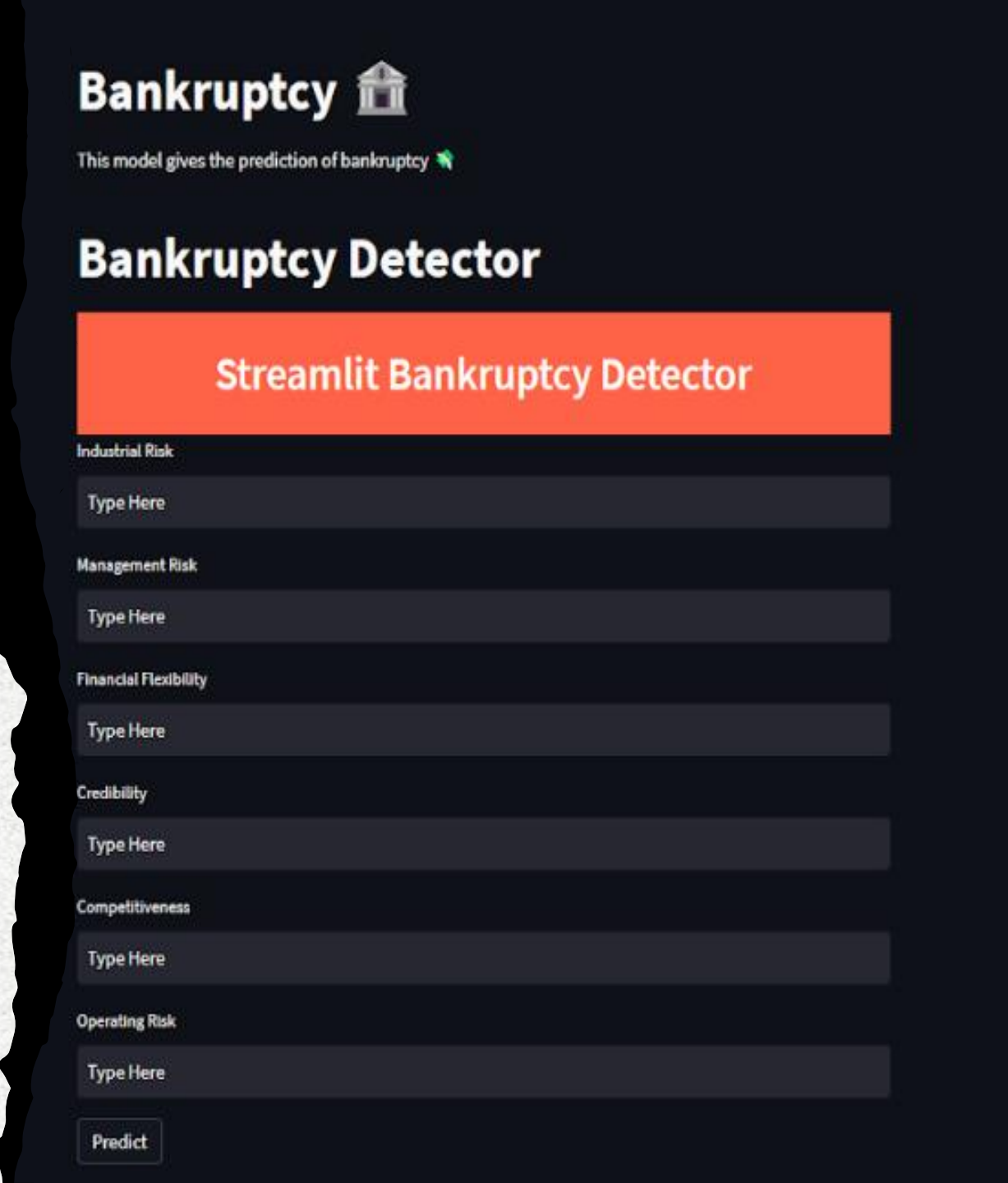
```
<AxesSubplot:>
```




Anaconda Prompt (anaconda
(base) C:\Users\Harshal>
(base) C:\Users\Harshal>cd C:\Coding\Deployment
(base) C:\Coding\Deployment>streamlit run bankruptcy.py
2024-02-07 19:51:25.510 INFO numexpr.utils: NumExpr defaulting to 8 threads.
You can now view your Streamlit app in your browser.
Local URL: http://localhost:8501
Network URL: http://192.168.38.102:8501

STREAMLIT

PROJECT DEPLOYMENT



THANK YOU!