Project: Bankruptcy Prevention

Business Objective

- Using the given information, build a model that predict whether the business is going to bankrupt or not.
- In dataset, it contains 7 features about 250 companies.
- In each columns, data values varies only with 0, 0.5 and 1 which means Low risk, Medium risk and High risk.
- Class column contains categorical data called bankruptcy and non-bankruptcy (Target Veriables).

Dataset Details & EDA

1) Import dataset in Jupyter Notebook and separate columns by removing semi-colon.
2) Removed 147 duplicate values for better accuracy.
3) There is no null values in dataframe.
4) Plotting Box-Plot, there is no outliers.
5) Convert categorical data into numerical data in class column using LabelEncoder. Convert bankruptcy = 0 and non-bankruptcy = 1
6) In class, most of the companies are non-bankrupt.
7) In the dataset, columns like competitiveness, credibility and financial_flexibility has strong positive correlation.
8) Whenever the risk is high the bankruptcy is low and whenever the risk is low bankruptcy is high.

Model Building and Evaluation

1) Selecting only those columns which have strong positive correlation with class (output).
2) Train Test split = Splitting our dataset into train and test using train_test_split(), what we are doing here is taking 80% of data to train our model, and 20% that we will hold back as a validation dataset.
3) Using the Lazypredict classifier to check accuracy of different algorithms on data.
4) As we have only 107 data points so that most of the algorithms giving100% accuracy. This indicates the model is overfit.
5) Using DecisionTreeClassifier, the accuracy is 0.95.

Snapshots

- Importing Dataset :

Importing Dataset

```
In [3]: data = pd.read_excel("bankruptcy-prevention.xlsx")

In [4]: data
```

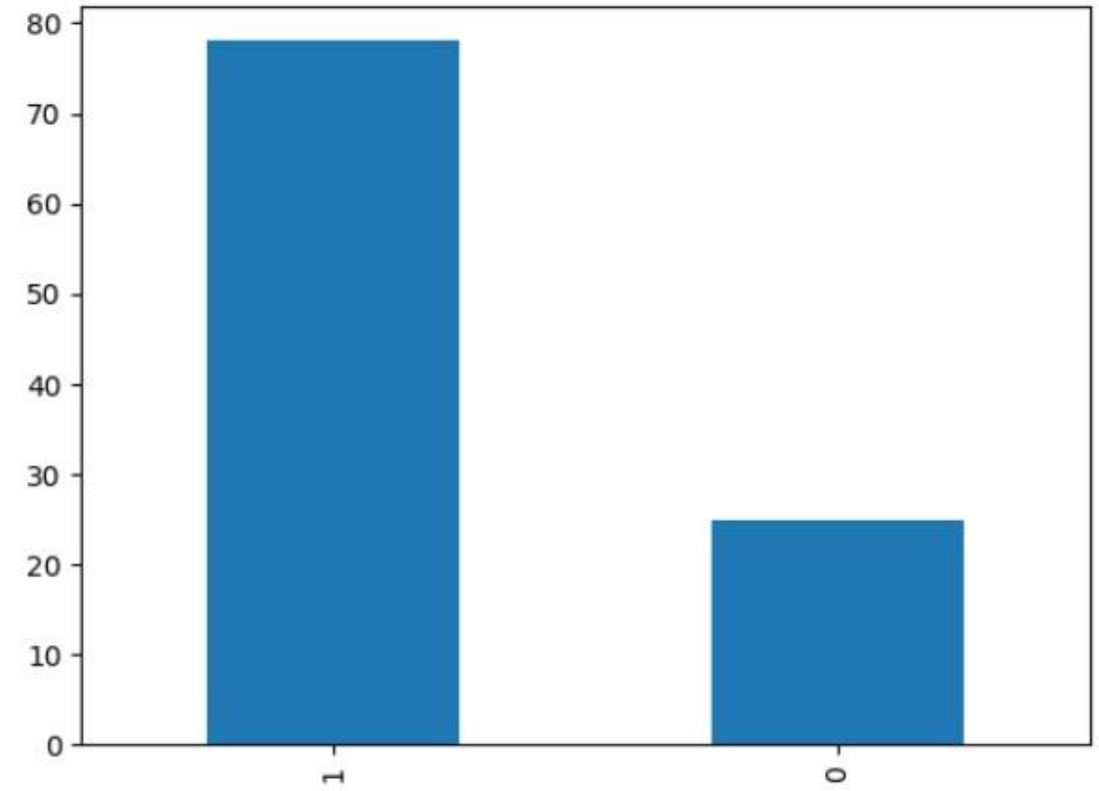
Out[4]:

	industrial_risk	management_risk	financial_flexibility	credibility	competitiveness	operating_risk	class
0	0.5	1.0	0.0	0.0	0.0	0.5	bankruptcy
1	0.0	1.0	0.0	0.0	0.0	1.0	bankruptcy
2	1.0	0.0	0.0	0.0	0.0	1.0	bankruptcy
3	0.5	0.0	0.0	0.5	0.0	1.0	bankruptcy
4	1.0	1.0	0.0	0.0	0.0	1.0	bankruptcy
...	...	...	...	...	...	...	...
245	0.0	1.0	1.0	1.0	1.0	1.0	non-bankruptcy
246	1.0	1.0	0.5	1.0	1.0	0.0	non-bankruptcy
247	0.0	1.0	1.0	0.5	0.5	0.0	non-bankruptcy
248	1.0	0.0	0.5	1.0	0.5	0.0	non-bankruptcy
249	1.0	0.0	0.5	0.5	1.0	1.0	non-bankruptcy

250 rows × 7 columns

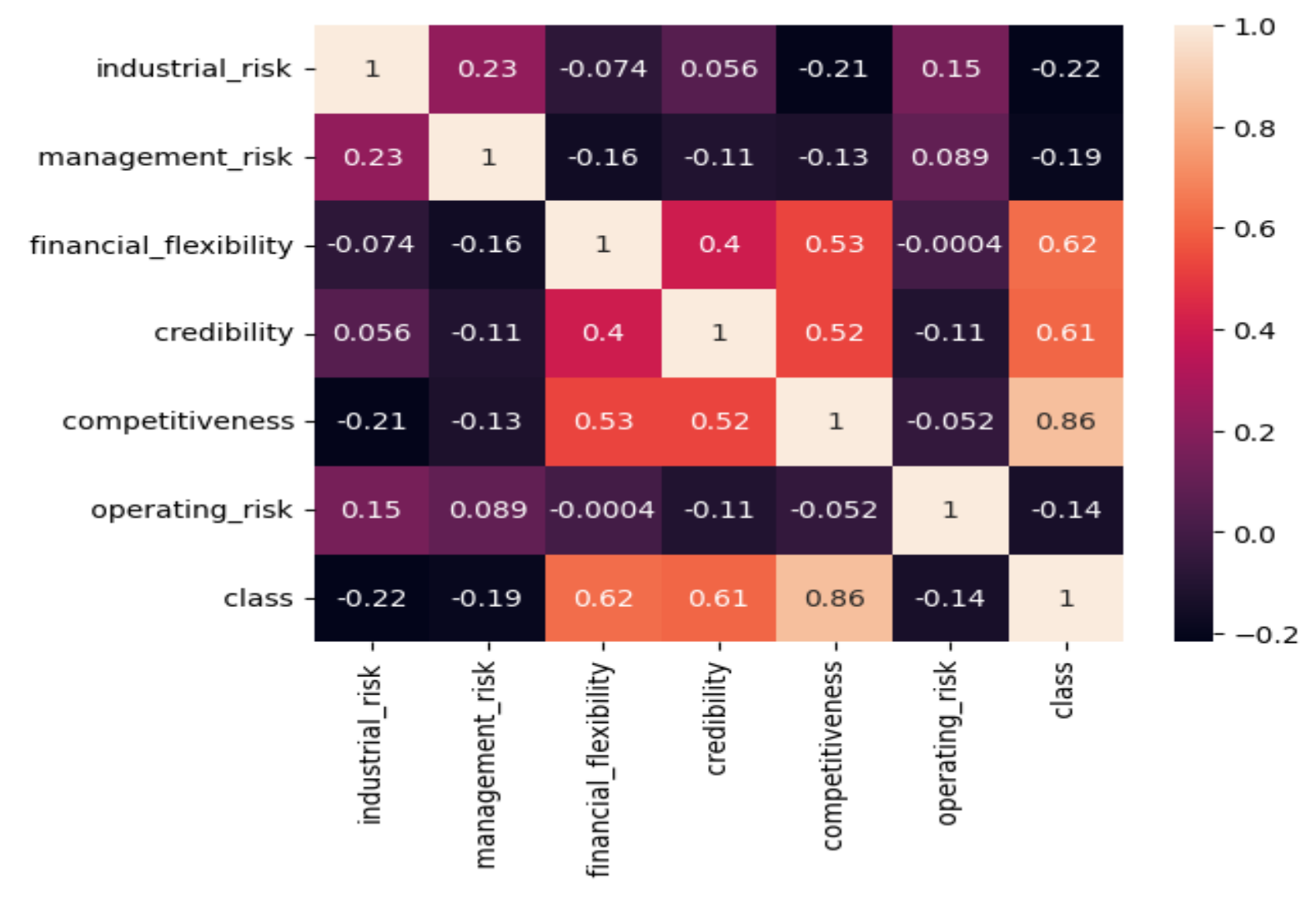
- Number of Companies went bankruptcy and non-bankruptcy :

```
In [28]: df3[' class'].value_counts().plot(kind = 'bar')
         plt.show()
```



In class, most of the companies are nonbankrupt

- Bivariate Analysis :



Model Building and Evaluation (1) :

Using the Lazypredict to check accuraccy of different algorethem on data

In [72]:
```
from lazypredict.Supervised import LazyClassifier
X_train, X_test, y_train, y_test = train_test_split(df4.drop(columns = [' class']),
                                                    df4[' class'],
                                                    test_size=0.20,
                                                    random_state =42)
clf = LazyClassifier(verbose=0,ignore_warnings=True, custom_metric=None)
models,predictions = clf.fit(X_train, X_test, y_train, y_test)
print(models)
```

```
100%|| 29/29 [00:00<00:00, 39.76it/s]
```

Model	Accuracy	Balanced Accuracy	ROC AUC	F1 Score
NearestCentroid	1.00	1.00	1.00	1.00
RidgeClassifierCV	1.00	1.00	1.00	1.00
RidgeClassifier	1.00	1.00	1.00	1.00
KNeighborsClassifier	1.00	1.00	1.00	1.00
LabelPropagation	1.00	1.00	1.00	1.00
LabelSpreading	1.00	1.00	1.00	1.00
LinearDiscriminantAnalysis	1.00	1.00	1.00	1.00
BernoulliNB	0.95	0.97	0.97	0.95
AdaBoostClassifier	0.95	0.90	0.90	0.95
XGBClassifier	0.95	0.90	0.90	0.95
SVC	0.95	0.90	0.90	0.95
SGDClassifier	0.95	0.90	0.90	0.95
RandomForestClassifier	0.95	0.90	0.90	0.95
Perceptron	0.95	0.90	0.90	0.95
PassiveAggressiveClassifier	0.95	0.90	0.90	0.95
LinearSVC	0.95	0.90	0.90	0.95
LogisticRegression	0.95	0.90	0.90	0.95
BaggingClassifier	0.95	0.90	0.90	0.95
GaussianNB	0.95	0.90	0.90	0.95
ExtraTreesClassifier	0.95	0.90	0.90	0.95
ExtraTreeClassifier	0.95	0.90	0.90	0.95
DecisionTreeClassifier	0.95	0.90	0.90	0.95
CalibratedClassifierCV	0.95	0.90	0.90	0.95
LGBMClassifier	0.95	0.90	0.90	0.95
QuadraticDiscriminantAnalysis	0.24	0.50	0.50	0.09
DummyClassifier	0.76	0.50	0.50	0.66

Model Building and Evaluation (2) :

Creating an object of DecisionTreeClassifier as clf and fit the object so that model will learn the parameter.

In [75]:
```
clf = DecisionTreeClassifier()
clf.fit(X_train,y_train)
```

Out[75]:
```
DecisionTreeClassifier()
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

Predicting the output of 20% traning data.

In [76]:
```
y_pred = clf.predict(X_test)
y_pred
```

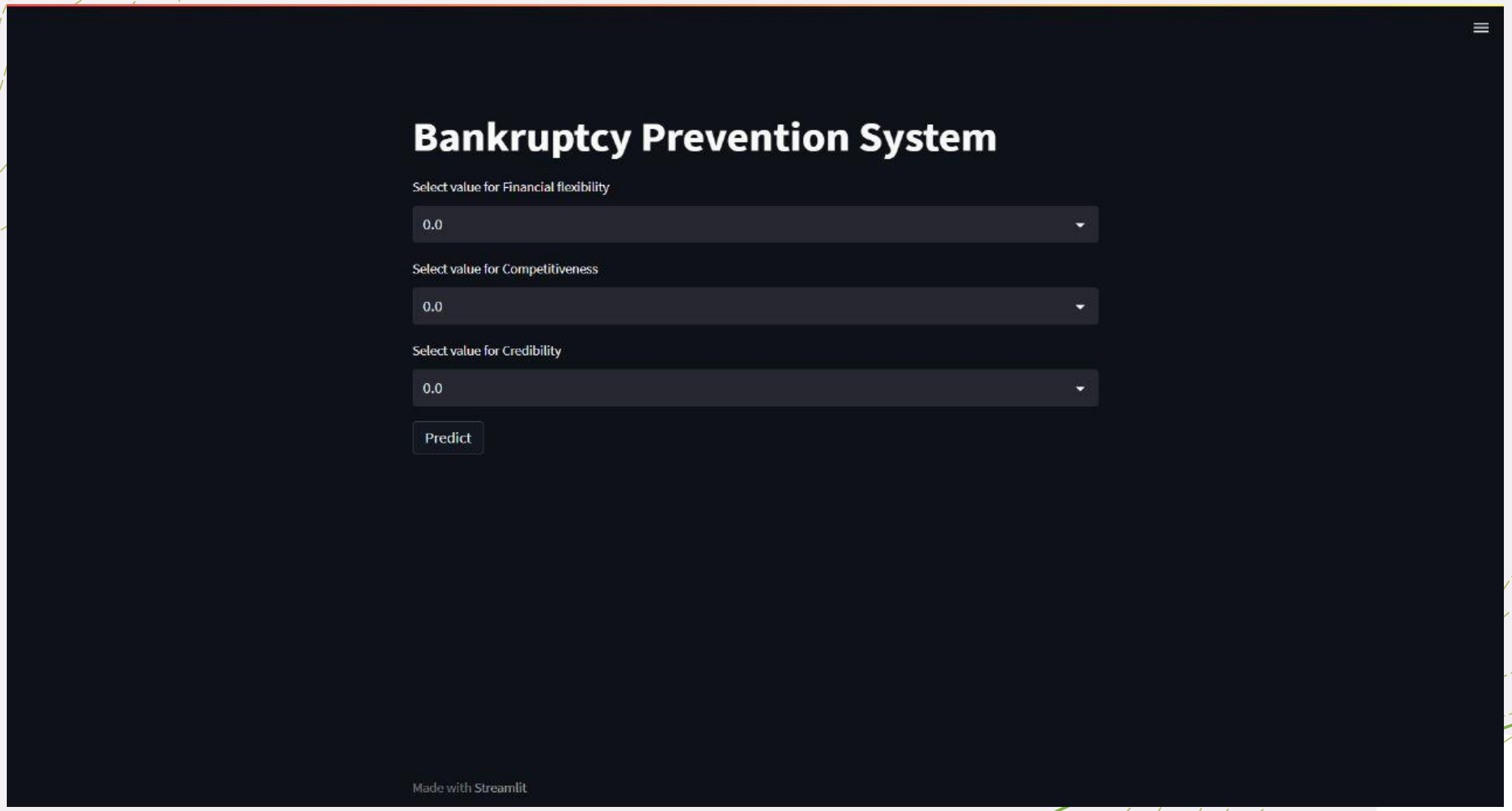
Out[76]:
```
array([1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1])
```

Calculate the accuracy score.

In [77]:
```
accuracy_score(y_test,y_pred)
```

Out[77]:
```
0.9523809523809523
```

- Deployment :


Bankruptcy Prevention System
Select value for Financial flexibility
0.0
Select value for Competitiveness
0.0
Select value for Credibility
0.0
Predict
Made with Streamlit

Conclusion

- In dataset, all columns are categorical the categories varies with 0.00,0.5 and 1.00.
- In the dataset, there are 57.2% duplicate values.
- The complete data ranges between 0-1 so that there are no outlier in the data and the data is uniform.
- In the dataset, the column name called competitiveness, credibility and financial_flexibility has strong positive correlation.
- Wherever the risk is high the bankruptcy is low and wherever the risk is low bankruptcy is high.
- We need to have LOW(industrial risk,management risk,operating risk) and HIGH(financial flexibility,credibility,competitiveness) in order to avoid Bankruptcy

- Contribution Name :

- ✓ Lavkush Chaudhary
- ✓ Shubhangi Birajdar
- ✓ Chandru Chivatgundi
- ✓ Tejas Sawant

Thank You