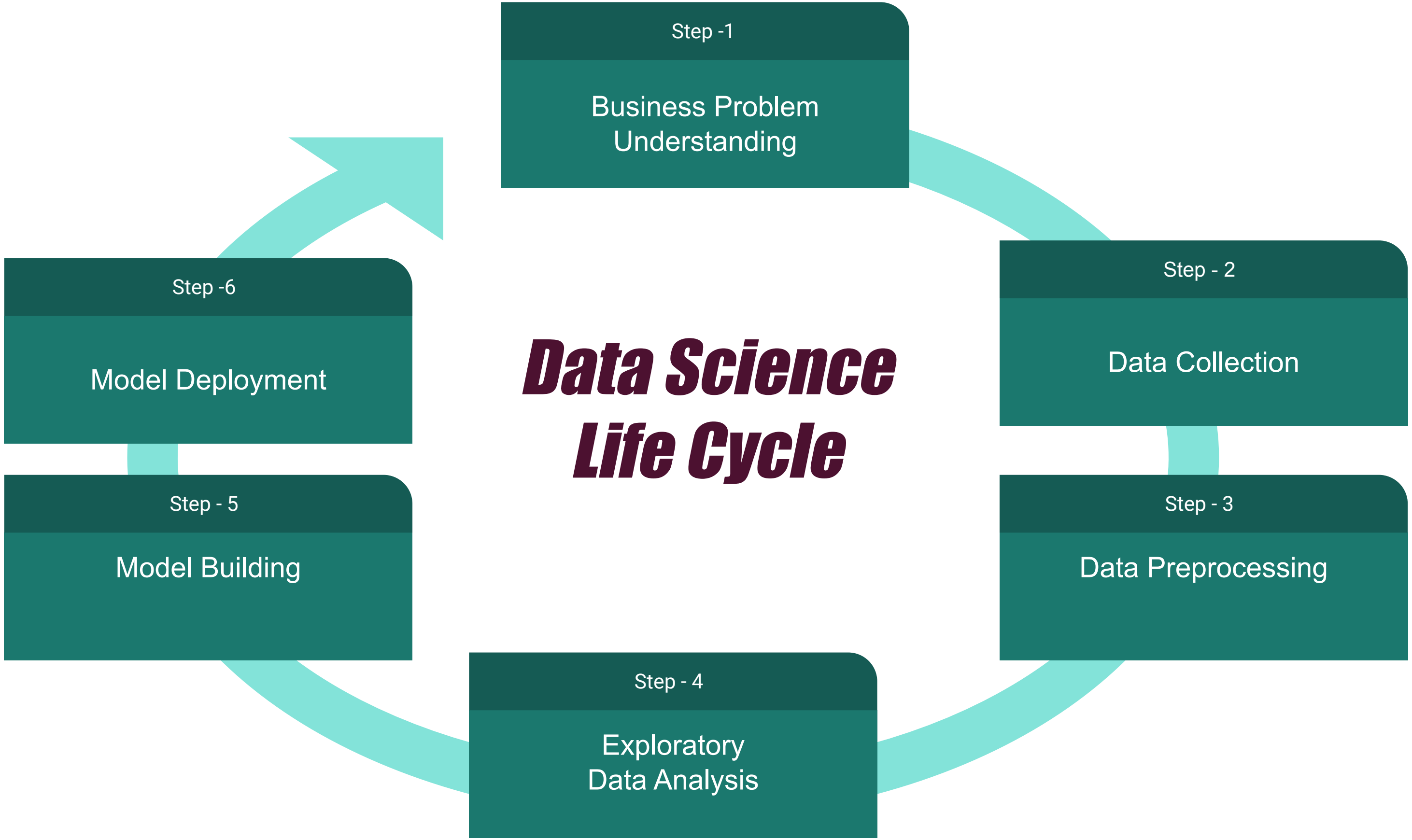
Big Data Analytics Project

Company Bankruptcy Prediction

Team Members :
Basava Chari Boppudi (S20200010043)
Balarajaiah Kalluri (S20200010085)

Instructor:
Dr. Arun PV

Data Science Life Cycle



Step -1

Business Problem Understanding

Step - 2

Data Collection

Step - 3

Data Preprocessing

Step - 4

Exploratory Data Analysis

Step - 5

Model Building

Step -6

Model Deployment

Business Problem Understanding

- As bankruptcy due to business failure can negatively affect the enterprise as well as the global economy, it is crucial to understand and predict whether a company is showing symptoms of getting bankrupt or not.
- The problem statement is to develop a prediction model which will predict whether a company can go bankrupt or not. This will help the company to take appropriate decisions.These distresses often lead to bankruptcy of the company if not alerted at the right time.
- This is a typical example of **classification** problem.

Artifacts - Problem statements

- Bankruptcy prediction is a complex problem due to the many factors that can contribute to a company's financial distress.
- Machine learning algorithms can be used to develop effective bankruptcy prediction models.
- These models can be used to identify companies at risk of bankruptcy early on, allowing for timely intervention.
- Studying this problem helps
    - Protect jobs and investments
    - Promote economic stability

Data Collection

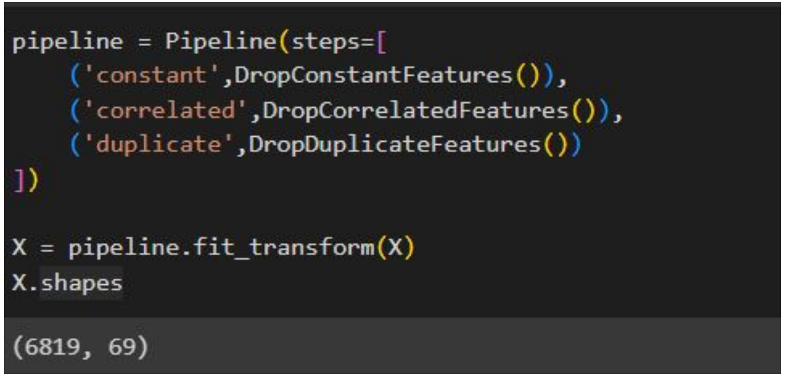
The data were collected from the Taiwan Economic Journal for the years 1999 to 2009. Company bankruptcy was defined based on the business regulations of the Taiwan Stock Exchange.

Dataset Characteristics Multivariate	**Subject Area** Business	**Associated Tasks** Classification
Feature Type Integer	**# Instances** 6819	**# Features** 96

Data Preprocessing

Data Preprocessing

- The dataset was clean so not much pre-processing was required.
- Removal of Constant features, Correlated features, duplicate features.
- After there steps almost 25 column dropped.
- 95 to 69 features.



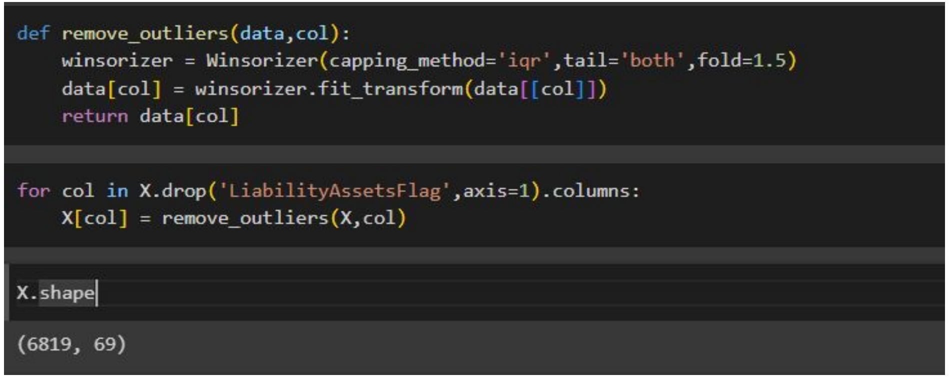
```
pipeline = Pipeline(steps=[
    ('constant',DropConstantFeatures()),
    ('correlated',DropCorrelatedFeatures()),
    ('duplicate',DropDuplicateFeatures())
])

X = pipeline.fit_transform(X)
X.shapes
```

```
(6819, 69)
```

Data Preprocessing

- Observed no NAN values.
- Checked for outlier in features.
- Observed no outliers.



```
def remove_outliers(data,col):
    winsorizer = Winsorizer(capping_method='iqr',tail='both',fold=1.5)
    data[col] = winsorizer.fit_transform(data[[col]])
    return data[col]
```

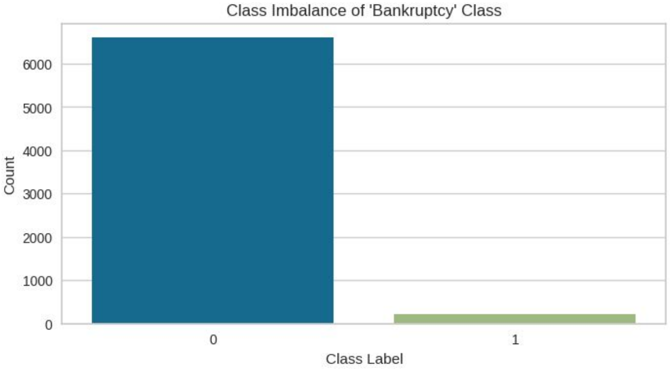
```
for col in X.drop('LiabilityAssetsFlag',axis=1).columns:
    X[col] = remove_outliers(X,col)
```

```
X.shape
```

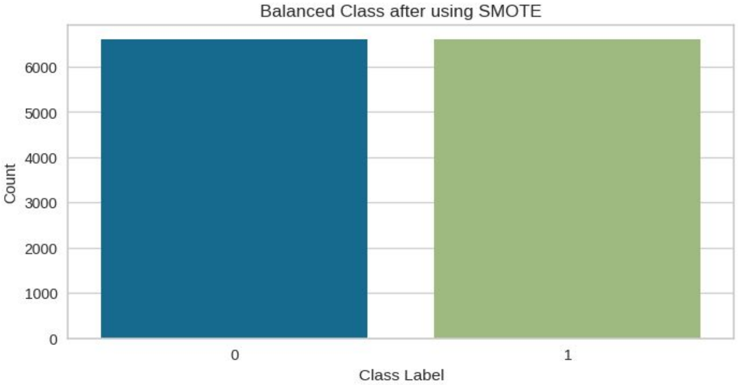
```
(6819, 69)
```

Data Preprocessing

SMOTE is an oversampling technique which creates synthetic data in the dimension of original data by drawing points on the line connecting two points of same class. This is a very famous data augmentation technique. I performed SMOTE on training data.



Class Imbalance of 'Bankruptcy' Class



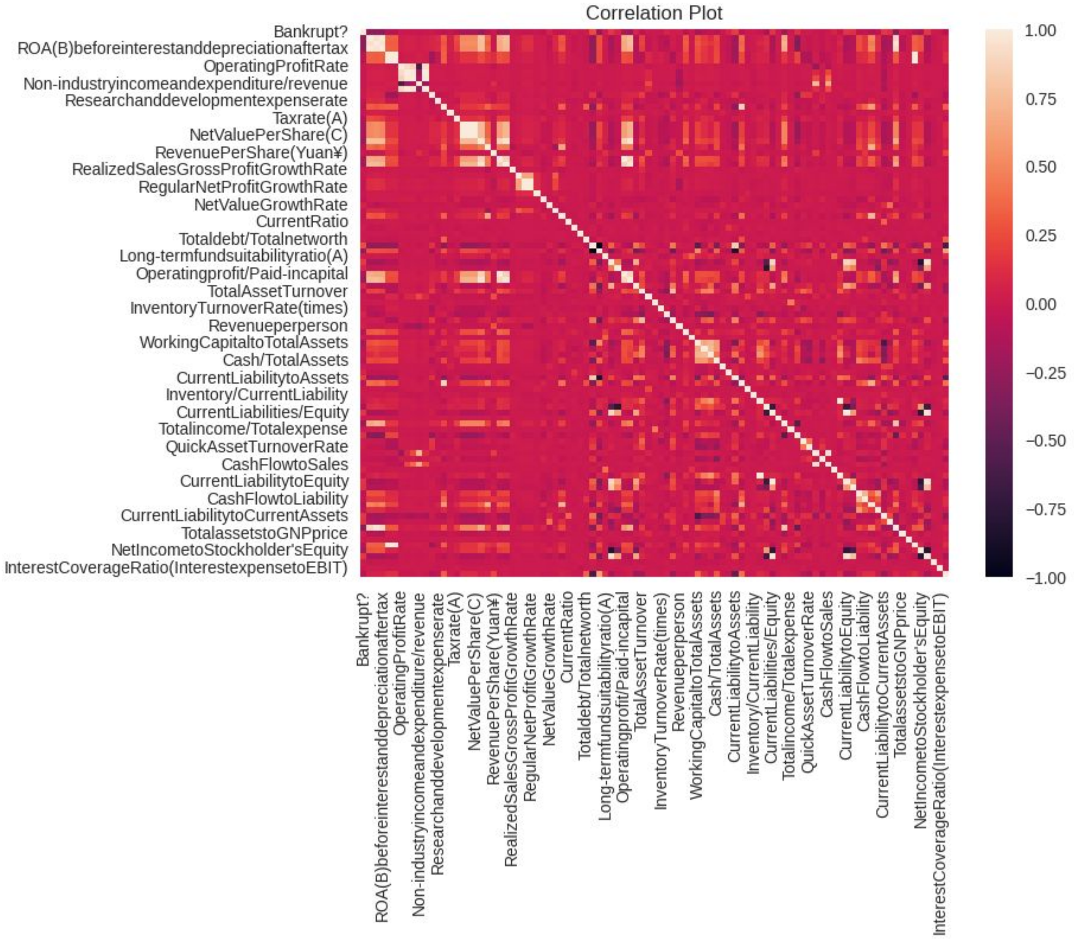
Balanced Class after using SMOTE

Artifact - Data Preprocessing

- Did necessary data preprocessing to get model evaluation better.
    - Removal of nan, duplicate and correlated columns.
    - Observed No outliers in dataset.
- Main observation is imbalanced data of output feature.
- Dataset has more data for non bankruptcy data than bankruptcy data.
- Used **SMOTE** method to get data balance which make data almost double the size.

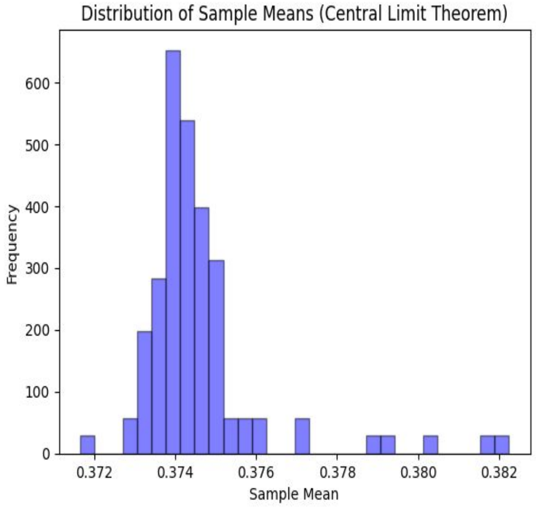
Exploratory Data Analysis

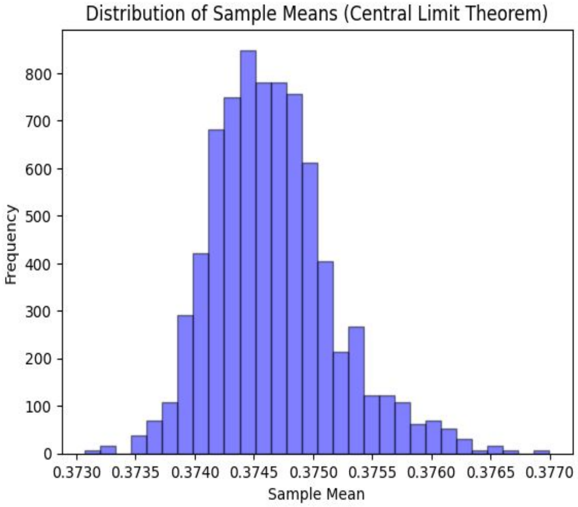
- Plotted the Correlation plot for features.
- Removed the most correlated features.

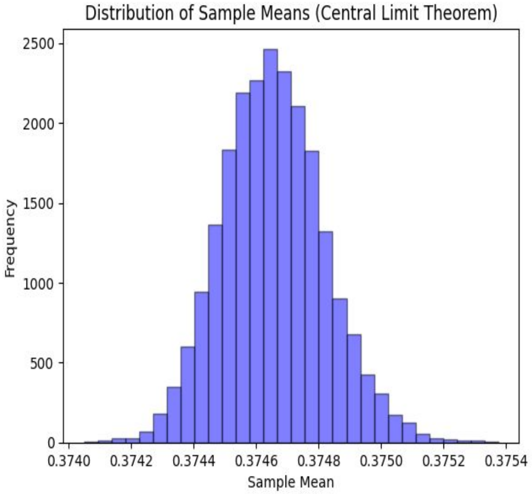


Central Limit Theorem

- To assess whether the ***Borrowing Dependency*** feature adheres to a normal distribution, we examined its normality across varying sample sizes (n1 = 100, n2 = 1,000, and n3 = 10,000).







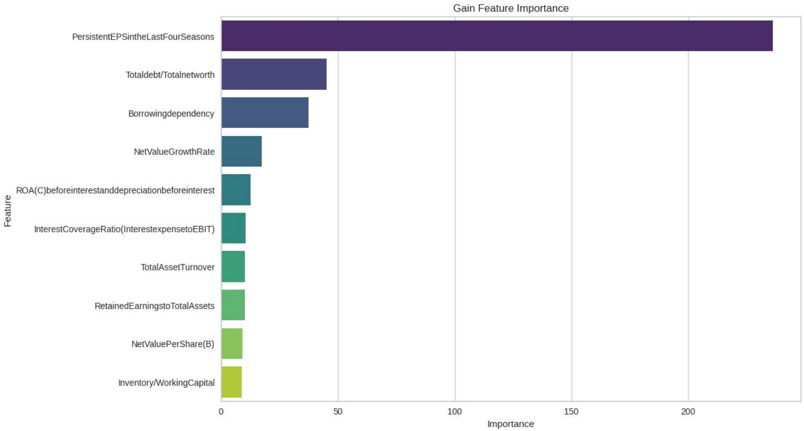
Hypothesis Test(t-test)

Variable (or)Feature	T-statistic	P-value	Hypothesis
Working Capital/Equity	-12.28921	2.386080	Reject H_0
Net Value Growth Rate	5.40547	6.68258	Reject H_0
Total debt/Total net worth	1.01674	0.30930	Fail to Reject H_0
Debt ratio %	21.33286	8.37395	Reject H_0
Net worth/Assets	-21.33286	8.3739	Reject H_0

- Variables with significant differences between bankrupt and non-bankrupt groups
  [' *Working Capital/Equity*', ' *Net Value Growth Rate*', ' *Debt ratio %*', ' *Net worth/Assets*']

Feature selection

- As there are around 70 feature main task is feature selection.
- Feature Importance Extraction
    - Utilizes XGBoost to extract feature importances from a machine learning model
- Taken to top 10 features.



Artifact - EDA and Feature selection

- Exploratory Data Analysis
  - Identified the relation among the features using Correlation Matrix.
  - Understanded the features using their distribution plots.
  - Identified the range of the each feature and normalized them using **Standard normalization**.
- Feature selection
  - Using XGboost feature selection, selected most important 10 features.

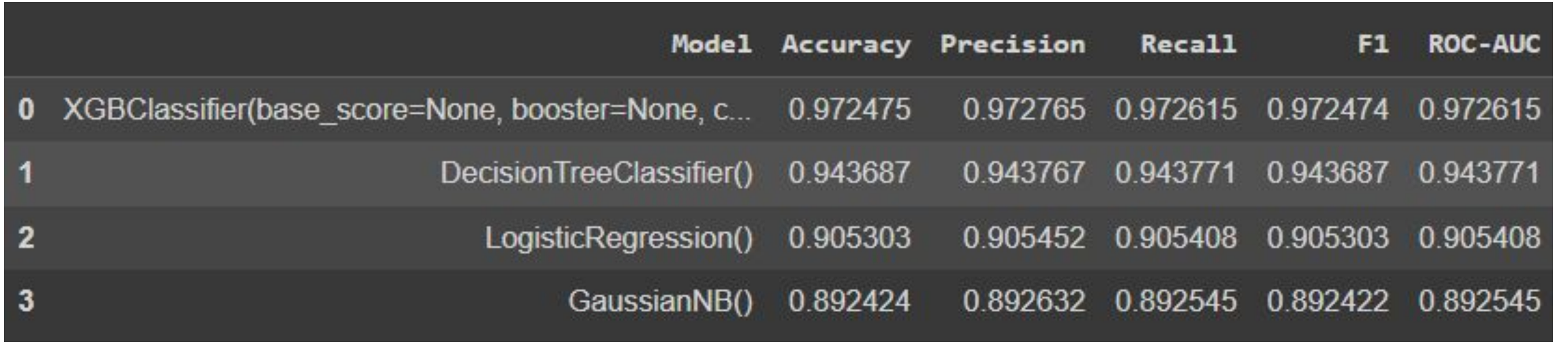
MODEL BUILDING AND EVALUATION

Logistic regression using Map Reduce

- Chunk-Based Logistic Regression in Hadoop
  - Leveraged Hadoop to handle large datasets efficiently by dividing them into manageable chunks.
  - Implemented a distributed logistic regression model using MapReduce, a powerful parallel processing framework.
- Parallel Processing in Mapper and Reducer
  - Mapper tasks were responsible for calculating chunk-specific weights, optimizing the logistic regression model for each data segment.
  - Reducer tasks played a vital role in aggregating the chunk-specific weights into final model weights.
  - Utilized the average function for aggregating.

Different Models and results:

- Tried different Base ML models.
- Among all the baseline models, the XGBoost Classifier is the best performing model which outperforms all other models by achieving a remarkable accuracy of almost 97% on the test set.

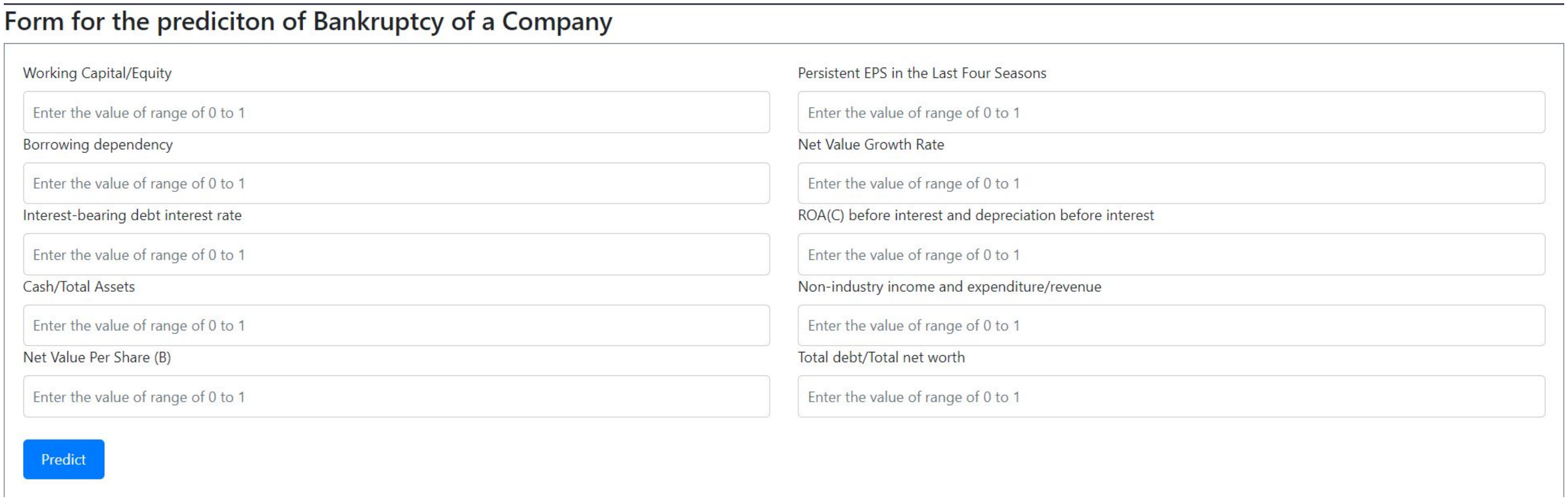


	Model	Accuracy	Precision	Recall	F1	ROC-AUC
0	XGBClassifier(base_score=None, booster=None, c...	0.972475	0.972765	0.972615	0.972474	0.972615
1	DecisionTreeClassifier()	0.943687	0.943767	0.943771	0.943687	0.943771
2	LogisticRegression()	0.905303	0.905452	0.905408	0.905303	0.905408
3	GaussianNB()	0.892424	0.892632	0.892545	0.892422	0.892545

Model Deployment

- Developed a Flask-based API to provide predictive insights into company bankruptcy.
- Integrated the API with frontend applications, enabling stakeholders to access bankruptcy predictions via HTTP requests.
- Implemented the basic chatbot using Google dialog flow.
- It helps to get understand of the website.

Demo for frontend page.



Form for the prediciton of Bankruptcy of a Company

Working Capital/Equity

Enter the value of range of 0 to 1

Borrowing dependency

Enter the value of range of 0 to 1

Interest-bearing debt interest rate

Enter the value of range of 0 to 1

Cash/Total Assets

Enter the value of range of 0 to 1

Net Value Per Share (B)

Enter the value of range of 0 to 1

Persistent EPS in the Last Four Seasons

Enter the value of range of 0 to 1

Net Value Growth Rate

Enter the value of range of 0 to 1

ROA(C) before interest and depreciation before interest

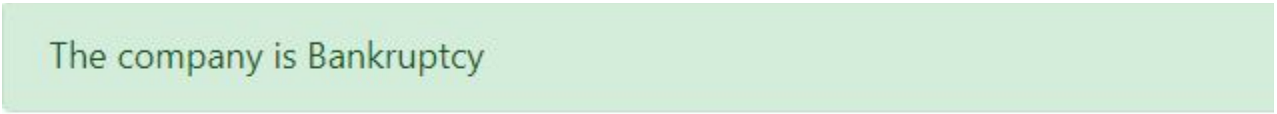
Enter the value of range of 0 to 1

Non-industry income and expenditure/revenue

Enter the value of range of 0 to 1

Total debt/Total net worth

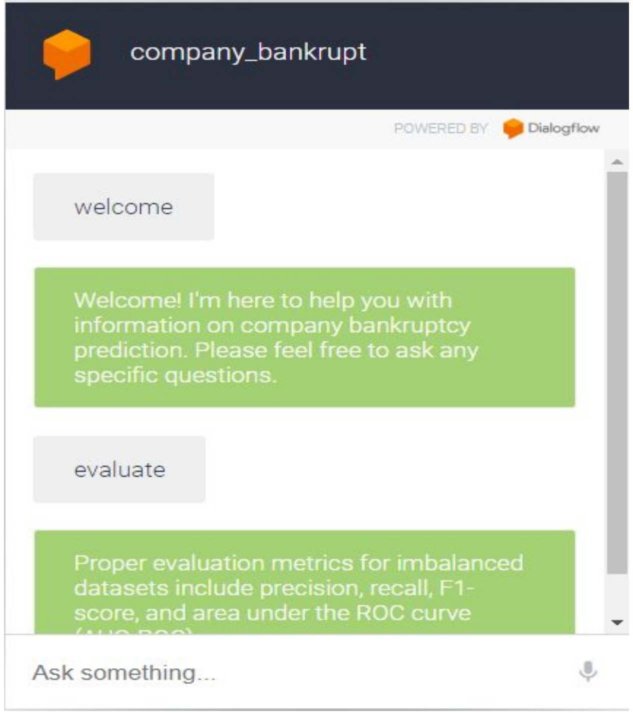
Enter the value of range of 0 to 1



The company is Bankruptcy

Chatbot using Dialogflow

- Dialogue flow for company bankrupt prediction.
- Help the user to understand the website using interaction with chatbot.



welcome

Welcome! I'm here to help you with information on company bankruptcy prediction. Please feel free to ask any specific questions.

evaluate

Proper evaluation metrics for imbalanced datasets include precision, recall, F1-score, and area under the ROC curve

Artifacts - Model Building

- Model building
  - Developed a logistic regression model for bankruptcy prediction using MapReduce and Hadoop to handle large datasets.
  - Evaluated various machine learning models, including XGBoost, decision tree, and linear regression, on the same data.
  - Achieved an accuracy of 97% with the XGBoost model, demonstrating its superior predictive performance.
- Model deployment
  - Built a user-friendly web interface using Flask framework to integrate the most accurate model (XGBoost).
  - Implemented a chat interface using Dialogflow framework to enhance user engagement and provide contextual assistance.

Thank you.