Great Learning

Financial Risk Analytics Project - Business Report

Submitted by:

N. Aishwarya
PGP-DSBA

October 2022

Business Report Outline

List of Tables

List of Formulas

List of Figures

Problem Statement:

Businesses or companies can fall prey to default if they are not able to keep up their debt obligations. Defaults will lead to a lower credit rating for the company which in turn reduces its chances of getting credit in the future and may have to pay higher interests on existing debts as well as any new obligations. From an investor's point of view, he would want to invest in a company if it is capable of handling its financial obligations, can grow quickly, and is able to manage the growth scale.

A balance sheet is a financial statement of a company that provides a snapshot of what a company owns, owes, and the amount invested by the shareholders. Thus, it is an important tool that helps evaluate the performance of a business.

Data that is available includes information from the financial statement of the companies for the previous year (2015). Also, information about the Net worth of the company in the following year (2016) is provided which can be used to drive the labeled field.

We need to create a default variable that should take the value of 1 when net worth next year is negative & 0 when net worth next year is positive.

Data dictionary as described below:

S. No	Field Name	Description	New Field Name
1	Co_Code	Company Code	Co_Code
2	Co_Name	Company Name	Co_Name
3	Networth Next Year	Value of a company as on 2016 - Next Year (difference between the value of total assets and total liabilities)	Networth_Next_Year
4	Equity Paid Up	Amount that has been received by the company through the issue of shares to the shareholders	Equity_Paid_Up
5	Networth	Value of a company as on 2015 - Current Year	Networth
6	Capital Employed	Total amount of capital used for the acquisition of profits by a company	Capital_Employed
7	Total Debt	The sum of money borrowed by the company and is due to be paid	Total Debt
8	Gross Block	Total value of all the assets that a company owns	Gross Block
9	Net Working Capital	The difference between a company's current assets (cash, accounts receivable, inventories of raw materials and finished goods) and its current liabilities (accounts payable).	Net_Working_Capital
10	Current Assets	All the assets of a company that are expected to be sold or used because of standard business operations over the next year.	Curr_Assets
11	Current Liabilities and Provisions	Short-term financial obligations that are due within one year (includes amount that is set aside cover a future liability)	Curr_Liab_and_Prov
12	Total Assets/Liabilities	Ratio of total assets to liabilities of the company	Total_Assets_to_Liab
13	Gross Sales	The grand total of sale transactions within the accounting period	Gross_Sales
14	Net Sales	Gross sales minus returns, allowances, and discounts	Net_Sales
15	Other Income	Income realized from non-business activities (e.g., sale of long-term asset)	Other Income
16	Value Of Output	Product of physical output of goods and services produced by company and its market price	Value_Of_Output

17	Cost of Production	Costs incurred by a business from manufacturing a product or providing a service	Cost_of_Prod
18	Selling Cost	Costs which are made to create the demand for the product (advertising expenditures, packaging and styling, salaries, commissions and travelling expenses of sales personnel, and the cost of shops and showrooms)	Selling_Cost
19	PBIDT	Profit Before Interest, Depreciation & Taxes	PBIDT
20	PBDT	Profit Before Depreciation and Tax	PBDT
21	PBIT	Profit before interest and taxes	PBIT
22	PBT	Profit before tax	PBT
23	PAT	Profit After Tax	PAT
24	Adjusted PAT	Adjusted profit is the best estimate of the true profit	Adjusted PAT
26	CP	Commercial paper, a short-term debt instrument to meet short-term liabilities.	CP
27	Revenue earnings in forex	Revenue earned in foreign currency	Rev_earn_in_forex
28	Revenue expenses in forex	Expenses due to foreign currency transactions	Rev_exp_in_forex
29	Capital expenses in forex	Long term investment in forex	Capital_exp_in_forex
30	Book Value (Unit Curr)	Net asset value	Book_Value_Unit_Curr
31	Book Value (Adj.) (Unit Curr)	Book value adjusted to reflect asset's true fair market value	Book_Value_Adj_Unit_Curr
32	Market Capitalisation	Product of the total number of a company's outstanding shares and the current market price of one share	Market_Capitalisation
33	CEPS (annualised) (Unit Curr)	Cash Earnings per Share, profitability ratio that measures the financial performance of a company by calculating cash flows on a per share basis	CEPS_annualised_Unit_Curr
34	Cash Flow from Operating Activities	Use of cash from ongoing regular business activities	Cash_Flow_From_Opr
35	Cash Flow from Investing Activities	Cash used in the purchase of non-current assets–or long-term assets– that will deliver value in the future	Cash_Flow_From_Inv
36	Cash Flow from Financing Activities	Net flows of cash that are used to fund the company (transactions involving debt, equity, and dividends)	Cash_Flow_From_Fin
37	ROG-Net Worth (%)	Rate of Growth - Networth	ROG_Net_Worth_perc
38	ROG-Capital Employed (%)	Rate of Growth - Capital Employed	ROG_Capital_Employed_perc
39	ROG-Gross Block (%)	Rate of Growth - Gross Block	ROG_Gross_Block_perc
40	ROG-Gross Sales (%)	Rate of Growth - Gross Sales	ROG_Gross_Sales_perc
41	ROG-Net Sales (%)	Rate of Growth - Net Sales	ROG_Net_Sales_perc
42	ROG-Cost of Production (%)	Rate of Growth - Cost of Production	ROG_Cost_of_Prod_perc
43	ROG-Total Assets (%)	Rate of Growth - Total Assets	ROG_Total_Assets_perc
44	ROG-PBIDT (%)	Rate of Growth- PBIDT	ROG_PBIDT_perc
45	ROG-PBDT (%)	Rate of Growth- PBDT	ROG_PBDT_perc
46	ROG-PBIT (%)	Rate of Growth- PBIT	ROG_PBIT_perc
47	ROG-PBT (%)	Rate of Growth- PBT	ROG_PBT_perc
48	ROG-PAT (%)	Rate of Growth- PAT	ROG_PAT_perc
49	ROG-CP (%)	Rate of Growth- CP	ROG_CP_perc
50	ROG-Revenue earnings in forex (%)	Rate of Growth - Revenue earnings in forex	ROG_Rev_earn_in_forex_perc
51	ROG-Revenue expenses in forex (%)	Rate of Growth - Revenue expenses in forex	ROG_Rev_exp_in_forex_perc

52	ROG-Market Capitalisation (%)	Rate of Growth - Market Capitalisation	ROG_Market_Capitalisation_perc
53	Current Ratio [Latest]	Liquidity ratio, company's ability to pay short-term obligations or those due within one year	Curr_Ratio_Latest
54	Fixed Assets Ratio [Latest]	Solvency ratio, the capacity of a company to discharge its obligations towards long-term lenders indicating	Fixed_Assets_Ratio_Latest
55	Inventory Ratio [Latest]	Activity ratio, specifies the number of times the stock or inventory has been replaced and sold by the company	Inventory_Ratio_Latest
56	Debtors Ratio [Latest]	Measures how quickly cash debtors are paying back to the company	Debtors_Ratio_Latest
57	Total Asset Turnover Ratio [Latest]	The value of a company's revenues relative to the value of its assets	Total_Asset_Turnover_Ratio_Latest
58	Interest Cover Ratio [Latest]	Determines how easily a company can pay interest on its outstanding debt	Interest_Cover_Ratio_Latest
59	PBIDTM (%) [Latest]	Profit before Interest Depreciation and Tax Margin	PBIDTM_perc_Latest
60	PBITM (%) [Latest]	Profit Before Interest Tax Margin	PBITM_perc_Latest
61	PBDTM (%) [Latest]	Profit Before Depreciation Tax Margin	PBDTM_perc_Latest
62	CPM (%) [Latest]	Cost per thousand (advertising cost)	CPM_perc_Latest
63	APATM (%) [Latest]	After tax profit margin	APATM_perc_Latest
64	Debtors Velocity (Days)	Average days required for receiving the payments	Debtors_Vel_Days
65	Creditors Velocity (Days)	Average number of days company takes to pay suppliers	Creditors_Vel_Days
66	Inventory Velocity (Days)	Average number of days the company needs to turn its inventory into sales	Inventory_Vel_Days
67	Value of Output/Total Assets	Ratio of Value of Output (market value) to Total Assets	Value_of_Output_to_Total_Assets
68	Value of Output/Gross Block	Ratio of Value of Output (market value) to Gross Block	Value_of_Output_to_Gross_Block

Table 1: Data dictionary for the dataset

There is total 68 variables in this dataset. It contains various measures related to company business.

Exploratory Data Analysis:

FRA dataset data is loaded using pandas and the dataset has 3,586 observations (rows) and 67 variables (columns). A quick glimpse of the data is shown below:

	Co_Code	Co_Name	Networth Next Year	Equity Paid Up	Networth	Capital Employed	Total Debt	Gross Block	Net Working Capital	Current Assets	Current Liabilities and Provisions	Total Assets/Liabilities	Gross Sales	Net Sales
0	16974	Hind.Cables	-8021.60	419.36	-7027.48	-1007.24	5936.03	474.30	-1076.34	40.50	1116.85	109.60	0.00	0.00
1	21214	Tata Tele. Mah.	-3986.19	1954.93	-2968.08	4458.20	7410.18	9070.86	-1098.88	486.86	1585.74	6043.94	2892.73	2892.73
2	14852	ABG Shipyard	-3192.58	53.84	506.86	7714.68	6944.54	1281.54	4496.25	9097.64	4601.39	12316.07	392.13	392.13
3	2439	GTL	-3054.51	157.30	-623.49	2353.88	2326.05	1033.69	-2612.42	1034.12	3646.54	6000.42	1354.39	1354.39
4	23505	Bharati Defence	-2967.36	50.30	-1070.83	4675.33	5740.90	1084.20	1836.23	4685.81	2849.58	7524.91	38.72	38.72

Figure 1: Dataset overview

Description of variables are as below, to understand the data better:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3586 entries, 0 to 3585
Data columns (total 67 columns):
 #   Column                                Non-Null Count  Dtype
---  ------                                --------------  -----
 0   Co_Code                               3586 non-null   int64
 1   Co_Name                               3586 non-null   object
 2   Networth_Next_Year                    3586 non-null   float64
 3   Equity_Paid_Up                        3586 non-null   float64
 4   Networth                              3586 non-null   float64
 5   Capital_Employed                      3586 non-null   float64
 6   Total_Debt                            3586 non-null   float64
 7   Gross_Block                           3586 non-null   float64
 8   Net_Working_Capital                   3586 non-null   float64
 9   Current_Assets                        3586 non-null   float64
 10  Current_Liabilities_and_Provisions    3586 non-null   float64
 11  Total_Assets_by_Liabilities           3586 non-null   float64
 12  Gross_Sales                           3586 non-null   float64
 13  Net_Sales                             3586 non-null   float64
 14  Other_Income                          3586 non-null   float64
 15  Value_Of_Output                       3586 non-null   float64
 16  Cost_of_Production                    3586 non-null   float64
 17  Selling_Cost                          3586 non-null   float64
 18  PBIDT                                 3586 non-null   float64
 19  PBDT                                  3586 non-null   float64
 20  PBIT                                  3586 non-null   float64
 21  PBT                                   3586 non-null   float64
 22  PAT                                   3586 non-null   float64
 23  Adjusted_PAT                          3586 non-null   float64
 24  CP                                    3586 non-null   float64
 25  Revenue_earnings_in_forex             3586 non-null   float64
 26  Revenue_expenses_in_forex             3586 non-null   float64
 27  Capital_expenses_in_forex             3586 non-null   float64
 28  Book_Value_Unit_Curr                  3586 non-null   float64
 29  Book_Value_Adj._Unit_Curr             3582 non-null   float64
 30  Market_Capitalisation                 3586 non-null   float64
 31  CEPS_annualised_Unit_Curr             3586 non-null   float64
 32  Cash_Flow_From_Operating_Activities   3586 non-null   float64
 33  Cash_Flow_From_Investing_Activities   3586 non-null   float64
 34  Cash_Flow_From_Financing_Activities   3586 non-null   float64
 35  ROG_Net_Worth_perc                    3586 non-null   float64
 36  ROG_Capital_Employed_perc             3586 non-null   float64
 37  ROG_Gross_Block_perc                  3586 non-null   float64
 38  ROG_Gross_Sales_perc                  3586 non-null   float64
 39  ROG_Net_Sales_perc                    3586 non-null   float64
 40  ROG_Cost_of_Production_perc           3586 non-null   float64
 41  ROG_Total_Assets_perc                 3586 non-null   float64
 42  ROG_PBIDT_perc                        3586 non-null   float64
 43  ROG_PBDT_perc                         3586 non-null   float64
 44  ROG_PBIT_perc                         3586 non-null   float64
 45  ROG_PBT_perc                          3586 non-null   float64
 46  ROG_PAT_perc                          3586 non-null   float64
 47  ROG_CP_perc                           3586 non-null   float64
 48  ROG_Revenue_earnings_in_forex_perc    3586 non-null   float64
 49  ROG_Revenue_expenses_in_forex_perc    3586 non-null   float64
 50  ROG_Market_Capitalisation_perc        3586 non-null   float64
 51  Current_Ratio_Latest                  3585 non-null   float64
 52  Fixed_Assets_Ratio_Latest             3585 non-null   float64
 53  Inventory_Ratio_Latest                3585 non-null   float64
 54  Debtors_Ratio_Latest                  3585 non-null   float64
 55  Total_Asset_Turnover_Ratio_Latest     3585 non-null   float64
 56  Interest_Cover_Ratio_Latest           3585 non-null   float64
 57  PBIDTM_perc_Latest                    3585 non-null   float64
 58  PBITM_perc_Latest                     3585 non-null   float64
 59  PBDTM_perc_Latest                     3585 non-null   float64
 60  CPM_perc_Latest                       3585 non-null   float64
 61  APATM_perc_Latest                     3585 non-null   float64
 62  Debtors_Velocity_Days                 3586 non-null   int64
 63  Creditors_Velocity_Days               3586 non-null   int64
 64  Inventory_Velocity_Days               3483 non-null   float64
 65  Value_of_Output_by_Total_Assets       3586 non-null   float64
 66  Value_of_Output_by_Gross_Block        3586 non-null   float64
dtypes: float64(63), int64(3), object(1)
```

Figure 2: Description of variables

Observations:

- The special characters in the variable names (Field names) have been replaced to get to the suggested variable names mentioned in data dictionary.
- There are 3586 rows and 67 columns (variables).
- All the variables are numeric type except one variable (Co_Name) which is object type.
- For our analysis, Co_Code and Co_Name are dropped.
- There is no duplicate entry in the dataset.
- The problem statement requires to predict “default” status of the company where the “Networth Next Year” of the company is used to drive the “default” field. The “default” is 1 when “Networth Next Year” is negative, and it is 0 when “Networth Next Year” is positive. The “Default” field is created and added to the dataset based on the condition mentioned above. Subsequently “Networth Next Year” is not considered further as it became redundant.
- There are missing values in 13 of the variables. Missing values will be treated with either mean or median values of corresponding variables.
- There are outliers in the dataset. It will be treated for our analysis.

Statistical description of the dataset:

	count	mean	std	min	25%	50%	75%	max
Co_Code	3586.0	16065.388734	19776.817379	4.00	3029.2500	6077.500	24269.5000	72493.00
Networth_Next_Year	3586.0	725.045251	4769.681004	-8021.60	3.9850	19.015	123.8025	111729.10
Equity_Paid_Up	3586.0	62.966584	778.761744	0.00	3.7500	8.290	19.5175	42263.46
Networth	3586.0	649.746299	4091.988792	-7027.48	3.8925	18.580	117.2975	81657.35
Capital_Employed	3586.0	2799.611054	26975.135385	-1824.75	7.6025	39.090	226.6050	714001.25
Total_Debt	3586.0	1994.823779	23652.842746	-0.72	0.0300	7.490	72.3500	652823.81
Gross_Block	3586.0	594.178829	4871.547802	-41.19	0.5700	15.870	131.8950	128477.59
Net_Working_Capital	3586.0	410.809665	6301.218546	-13162.42	0.9425	10.145	61.1750	223257.56
Current_Assets	3586.0	1960.349172	22577.570829	-0.91	4.0000	24.540	135.2775	721166.00
Current_Liabilities_and_Provisions	3586.0	391.992078	2675.001631	-0.23	0.7325	9.225	65.6500	83232.98
Total_Assets_by_Liabilities	3586.0	1778.453751	11437.574690	-4.51	10.5550	52.010	310.5400	254737.22
Gross_Sales	3586.0	1123.738985	10603.703837	-62.59	1.4425	31.210	242.2500	474182.94
Net_Sales	3586.0	1079.702579	9996.574173	-62.59	1.4400	30.440	234.4400	443775.16
Other_Income	3586.0	48.729824	426.040665	-448.72	0.0200	0.450	3.6350	14143.40
Value_Of_Output	3586.0	1077.187292	9843.880293	-119.10	1.4125	30.895	235.8375	435559.09
Cost_of_Production	3586.0	798.544621	9076.702982	-22.65	0.9400	25.990	189.5500	419913.50
Selling_Cost	3586.0	25.554997	194.244466	0.00	0.0000	0.160	3.8825	5283.91
PBIDT	3586.0	248.175282	1949.593350	-4655.14	0.0400	2.045	23.5250	42059.26

Figure 3: Statistical description of the dataset

The values of mean, standard deviation, minimum and maximum, 25th, 50th and 75th percentile is mentioned in the above tables.

The statistical summary shows that there are many outliers present in the dataset in almost all the variables. The median Networth is 18.58 units whereas minimum is -7027 units and maximum are 81657 units. This shows that how much deviation is there which is also proved by the difference in mean and std. deviation.

ROG-Revenue earnings in forex and expense variable shows that the 25th & 75th percentile are 0 which means they have a large chunk of data having zero which might not be contributing enough towards the output and can be eliminated. We will look further into the outliers to see how bad the situation is, whether any treatment is required or not

Checking for Duplicate values in dataset:

Number of duplicate rows = 0

Checking for NULL value:

Co_Code	0
Co_Name	0
Networth_Next_Year	0
Equity_Paid_Up	0
Networth	0
Capital_Employed	0
Total_Debt	0
Gross_Block	0
Net_Working_Capital	0
Current_Assets	0
Current_Liabilities_and_Provisions	0
Total_Assets_by_Liabilities	0
Gross_Sales	0
Net_Sales	0
Other_Income	0
Value_Of_Output	0
Cost_of_Production	0
Selling_Cost	0
PBIDT	0
PBDT	0
PBIT	0
PBT	0
PAT	0
Adjusted_PAT	0
CP	0
Revenue_earnings_in_forex	0
Revenue_expenses_in_forex	0
Capital_expenses_in_forex	0
Book_Value_Unit_Curr	0
Book_Value_Adj._Unit_Curr	4

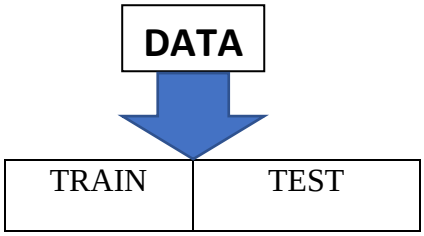
ROG_Gross_Sales_perc	0
ROG_Net_Sales_perc	0
ROG_Cost_of_Production_perc	0
ROG_Total_Assets_perc	0
ROG_PBIDT_perc	0
ROG_PBDT_perc	0
ROG_PBIT_perc	0
ROG_PBT_perc	0
ROG_PAT_perc	0
ROG_CP_perc	0
ROG_Revenue_earnings_in_forex_perc	0
ROG_Revenue_expenses_in_forex_perc	0
ROG_Market_Capitalisation_perc	0
Current_Ratio_Latest	1
Fixed_Assets_Ratio_Latest	1
Inventory_Ratio_Latest	1
Debtors_Ratio_Latest	1
Total_Asset_Turnover_Ratio_Latest	1
Interest_Cover_Ratio_Latest	1
PBIDTM_perc_Latest	1
PBITM_perc_Latest	1
PBDTM_perc_Latest	1
CPM_perc_Latest	1
APATM_perc_Latest	1
Debtors_Velocity_Days	0
Creditors_Velocity_Days	0
Inventory_Velocity_Days	103
Value_of_Output_by_Total_Assets	0
Value_of_Output_by_Gross_Block	0
Default	0

Table 2: Null value check

There are null values in 13 of the variables. These null values are imputed with median values as mean may not be correct one as the data variations are more and skewed.

Train Test Split

The original data frame except the variables Co_Code, Co_Name, Networth_Next_Year is divided independent and independent variable type data frame. Then both independent and dependent variable data frame is split into 67:33 (train: test) ratio. One requirement for Stats model is that dependent and independent variables should be contained in same data frame. So, concatenation was performed to combine dependent and independent variables arrays.




```
The number of rows (observations) in TRAIN set is 2402
The number of columns (variables) in TRAIN set is 65

The number of rows (observations) in TEST set is 1184
The number of columns (variables) in TEST set is 65
```

Logistic Regression Model:

The equation of the Logistic Regression by which we predict the corresponding probabilities and then go on predict a discrete target variable is

$$y = \frac{1}{1+e^{-z}}$$

Note: $z = \beta_0 + \sum_{i=1}^{n}(\beta_i X_i)$

Formula 1: Logistic Regression Model

1.8. Build a Random Forest Model on Train Dataset. Also showcase your model building approach.

Solution:

RF Model Performance Evaluation on Training data:

We have created two models using Random Forest one without SMOTE data and one with SMOTE data. We did not scale the data for Random Forest as tree-based models are not distance based models and can handle varying ranges of features.

We also used GridSearchCV for hyper parameter tuning. PFB the grid which was used. Same grid was used for both the models i.e., with and without smote to maintain uniformity.

We now fit our model to the GridSearchCV for Random Forest model by training the model with our independent variable and dependent variables:

- n_estimators = number of trees in the forest
- max_features = max number of features considered for splitting a node
- max_depth = max number of levels in each decision tree
- min_samples_split = min number of data points placed in a node before the node is split
- min_samples_leaf = min number of data points allowed in a leaf node

Below are the performance metrics on train data for random forest model without Smote:

Grid Search Best Parameters:

```
RandomForest
--------------------Best Parameters--------------------
{'max_depth': 20, 'max_features': 5, 'min_samples_leaf': 25, 'min_samples_split': 75, 'n_estimators': 100}
```

Best Model Parameters:

```
--------------------Best Model Params--------------------
RandomForestClassifier(max_depth=20, max_features=5, min_samples_leaf=25,
                       min_samples_split=75)
```

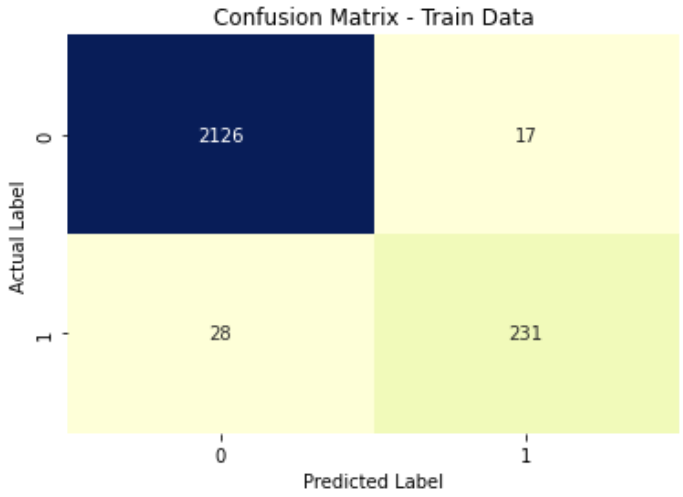
Figure 4: Train data performance metrics for random forest model without Smote

A custom function named apply_evl which takes below arguments was created. This function is able to handle modelling for different type of models and returns the performance metrics as the output. The same function has been used to build and evaluate the different models here.

Predicted Probability on train data:

	0	1
0	0.976599	0.023401
1	0.996066	0.003934
2	0.963224	0.036776
3	0.948924	0.051076
4	1.000000	0.000000

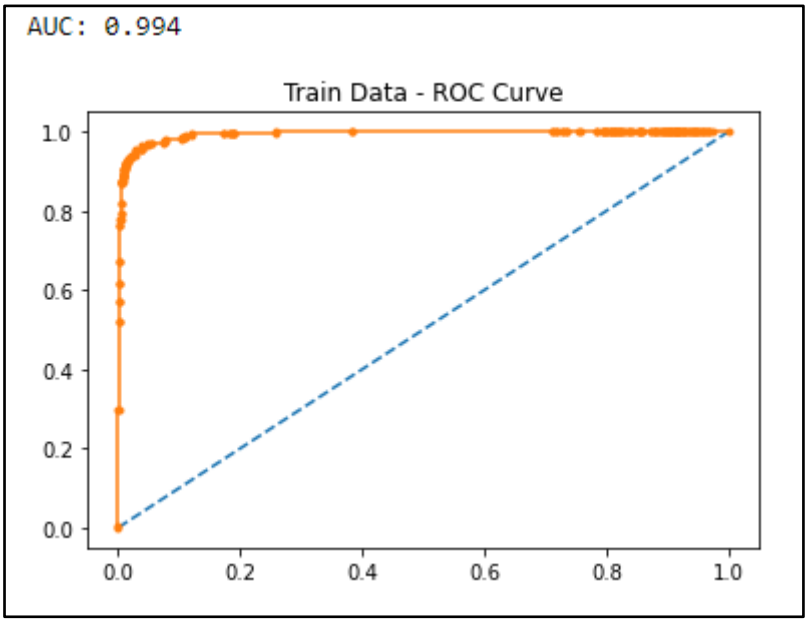
Confusion Matrix:



Classification Report:

	precision	recall	f1-score	support
0	0.99	0.99	0.99	2143
1	0.93	0.89	0.91	259
accuracy			0.98	2402
macro avg	0.96	0.94	0.95	2402
weighted avg	0.98	0.98	0.98	2402

Accuracy for training data – 96.7%



Variable Importance:

	Imp
Book_Value_Unit_Curr	0.255619
Book_Value_Adj._Unit_Curr	0.207487
Networth	0.197744
Capital_Employed	0.046181
Current_Ratio_Latest	0.040263
PBIDT	0.029316
PBDT	0.029011
CP	0.027483
CEPS_annualised_Unit_Curr	0.016372
ROG_Net_Worth_perc	0.015038
Net_Working_Capital	0.014392
PAT	0.012378
PBIT	0.012148
ROG_Capital_Employed_perc	0.011902
PBT	0.011808
Adjusted_PAT	0.010224
Total_Asset_Turnover_Ratio_Latest	0.005309
PBIDTM_perc_Latest	0.005187
PBDTM_perc_Latest	0.004334
Interest_Cover_Ratio_Latest	0.004150
Total_Debt	0.004127
CPM_perc_Latest	0.003869
PBITM_perc_Latest	0.003544
ROG_Total_Assets_perc	0.002702
Total_Assets_by_Liabilities	0.002583

Random forest classifier gives high importance to Net worth, Book_Value_Unit_Curr and Book_Value_Adj_Unit_Curr with more than 20% and rest of the variables have less than 5% importance. Moreover, in this model we could see that there are many variables with very less importance.

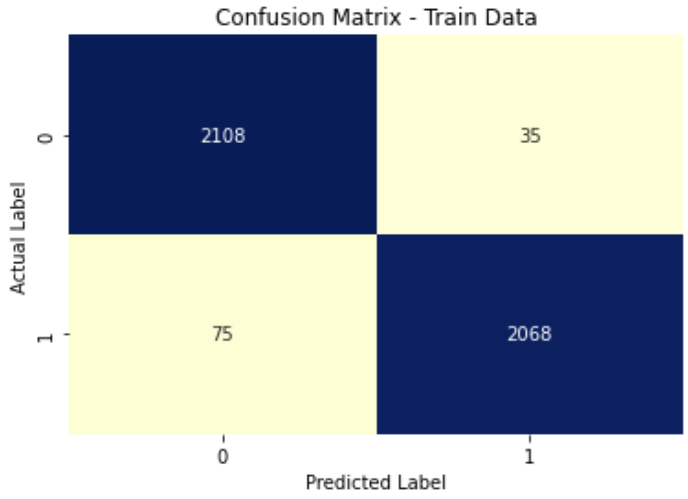
Below are the performance metrics on train data for random forest model with Smote:

```
RF_With_Smote
--------------------Best Parameters--------------------
{'max_depth': 30, 'max_features': 7, 'min_samples_leaf': 25, 'min_samples_split': 50, 'n_estimators': 50}
```

```
---------------------Best Model Params---------------------
RandomForestClassifier(max_depth=30, max_features=7, min_samples_leaf=25,
                       min_samples_split=50, n_estimators=50)
```

Figure 5: Train data performance metrics for random forest model with Smote

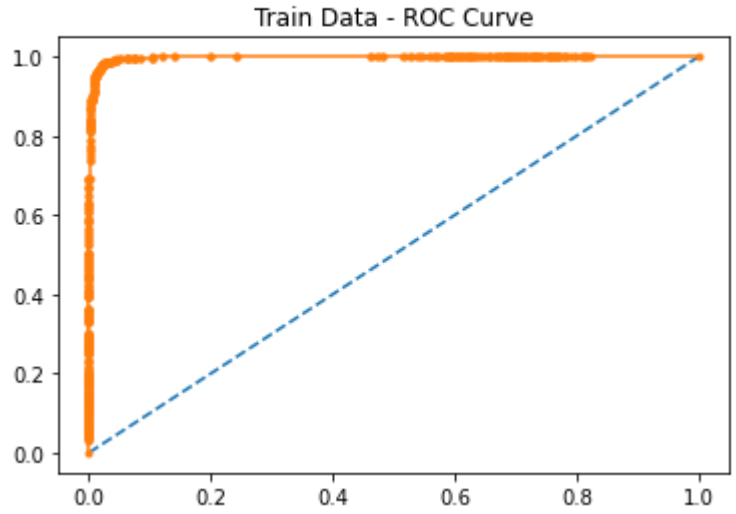
Confusion Matrix:



Classification Report:

```
--------------------Classification Report - Train Data------
              precision    recall  f1-score   support

           0       0.97      0.98      0.97      2143
           1       0.98      0.97      0.97      2143

    accuracy                           0.97      4286
   macro avg       0.97      0.97      0.97      4286
weighted avg       0.97      0.97      0.97      4286

------------------------------------------------------------
```

AUC: 0.998



1.9. Validate the Random Forest Model on test Dataset and state the performance matrices. Also state interpretation from the model.

Solution:

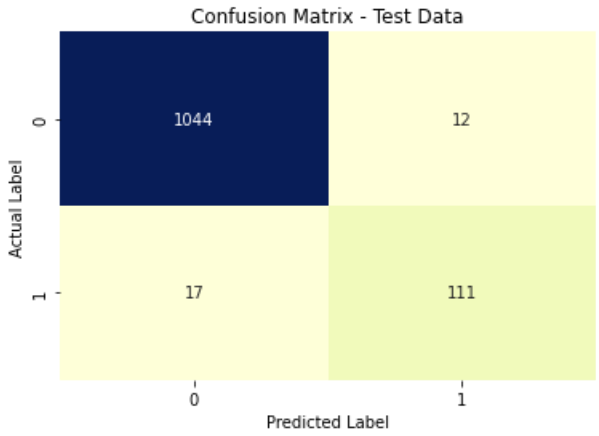
Random Forest on test data:

Both the random forest models were then evaluated on the test data. Below are the results for test data for random forest model without smote data.

Predicted Probability on train data:

	0	1
0	0.998850	0.001150
1	0.994463	0.005537
2	0.917903	0.082097
3	0.163215	0.836785
4	0.965124	0.034876

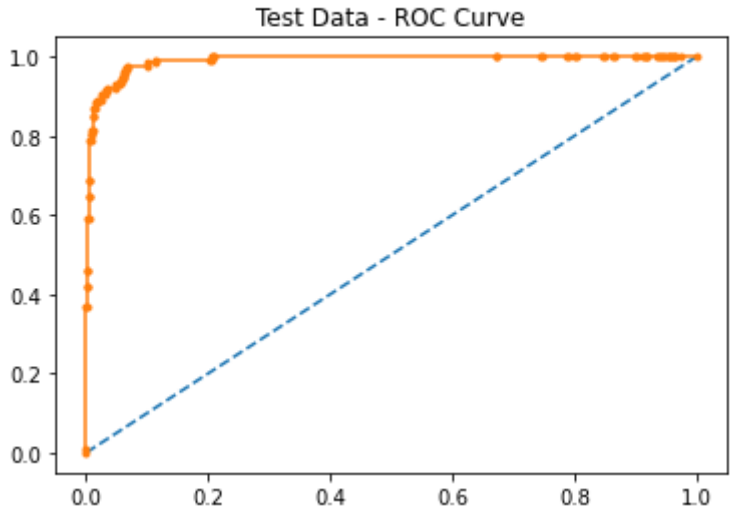
Confusion Matrix:



Classification Report:

```
--------------------Classification Report - Test Data----
              precision    recall  f1-score   support

           0       0.98      0.99      0.99      1056
           1       0.90      0.87      0.88       128

    accuracy                           0.98      1184
   macro avg       0.94      0.93      0.94      1184
weighted avg       0.98      0.98      0.98      1184
```

AUC: 0.990

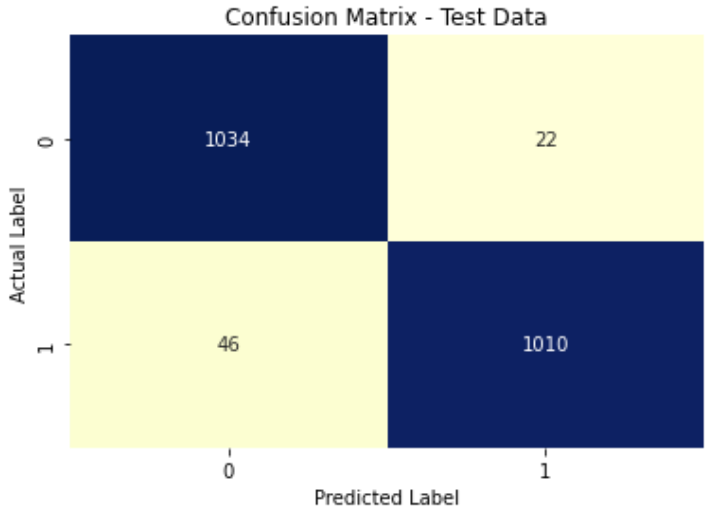


Below are the results for test data for random forest model with smote data:

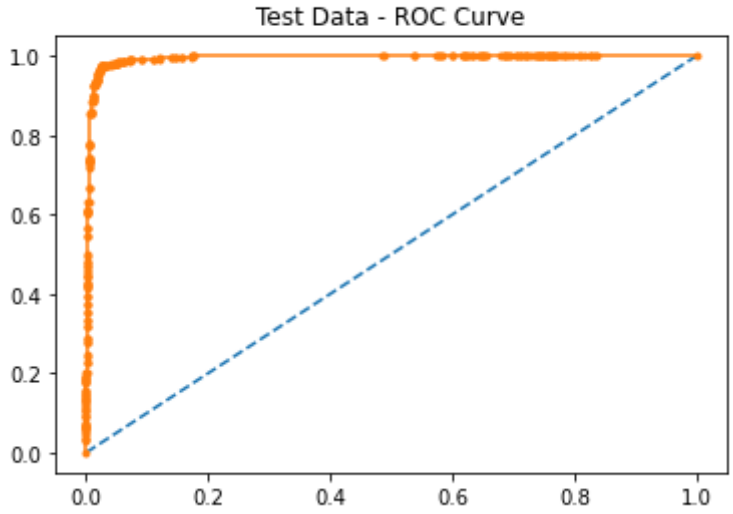
Classification Report:

```
--------------------Classification Report - Test Data-----
              precision    recall  f1-score   support

           0       0.96      0.98      0.97      1056
           1       0.98      0.96      0.97      1056

    accuracy                           0.97      2112
   macro avg       0.97      0.97      0.97      2112
weighted avg       0.97      0.97      0.97      2112
```

Confusion Matrix:

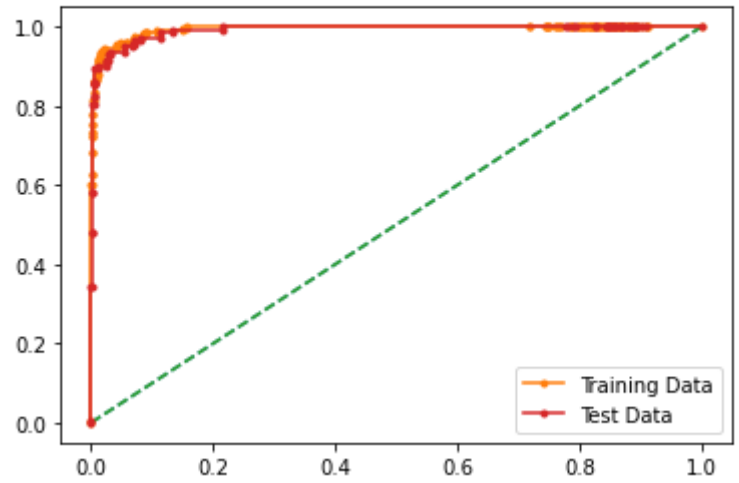


AUC: 0.994



AUC and ROC for the training and testing data:

```
AUC for the Training Data: 0.994
AUC for the Test Data: 0.991
```



Below is the comparison of various evaluation matric in tabular format for both the models i.e., with and without smote.

	Accuracy	Precision	Recall	F1	AUC
RandomForest_Train	0.980849	0.927711	0.891892	0.909449	0.993687
RandomForest_Test	0.975507	0.902439	0.867188	0.884462	0.989975

	Accuracy	Precision	Recall	F1	AUC
RF_With_Smote_Train	0.974335	0.983357	0.965002	0.974093	0.997926
RF_With_Smote_Test	0.967803	0.978682	0.956439	0.967433	0.994474

Figure 6: Comparison of various evaluation metrics – with and without SMOTE

Accuracy and precision of defaulter class have good values compared to Logistic Regression.

We can see in the above comparison that even though Random Forest without SMOTE has higher accuracy, but Random Forest with SMOTE surpasses the other model significantly in terms of Precision, Recall, F1 score as well as AUC score.

Also, there is not a significant difference in the accuracies of both the models.

Hence, we can say Random Forest with SMOTE data is the better of the two models with 97% accuracy, 97.86% precision, 95.64% recall and F1 score of 0.9674 and 0.9944 respectively.

1.10. Build a LDA Model on Train Dataset. Also showcase your model building approach.

Solution:

LDA takes the help of prior probabilities to predict the corresponding target probabilities. Prior probabilities are the probability of y (say equal to 1) without considering any other data or variables. The corresponding updated probabilities when the covariates (Xs) are available is called the posterior probabilities. We want to find P(Y=1|X). Thus, a Linear Discriminant Analysis (LDA) discriminates between the two classes by looking at the features (Xs).

We have created two models using LDA, one without SMOTE data and one with SMOTE data. We did not scale the data for LDA as it finds its coefficients using the variation between the classes, hence scaling doesn't matter.

We also used GridSearchCV for hyper parameter tuning. PFB the grid which was used. Same grid was used for both the models i.e., with and without smote to maintain uniformity.

Predicted Probability on train data:

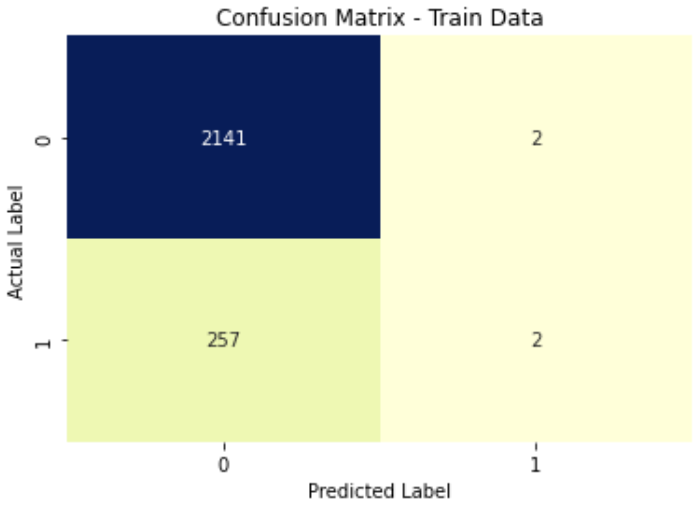
	0	1
0	0.996038	0.003962
1	0.998997	0.001003
2	0.999642	0.000358
3	0.985928	0.014072
4	0.995863	0.004137
5	0.969313	0.030687
6	0.997646	0.002354
7	0.998537	0.001463
8	0.996886	0.003114
9	0.998630	0.001370

Below are the performance metrics on train data for LDA model without Smote:

```
LDA
--------------------Best Parameters--------------------
{'shrinkage': 'auto', 'solver': 'lsqr', 'tol': 1e-06}
```

```
--------------------Best Model Params--------------------
LinearDiscriminantAnalysis(shrinkage='auto', solver='lsqr', tol=1e-06)
```

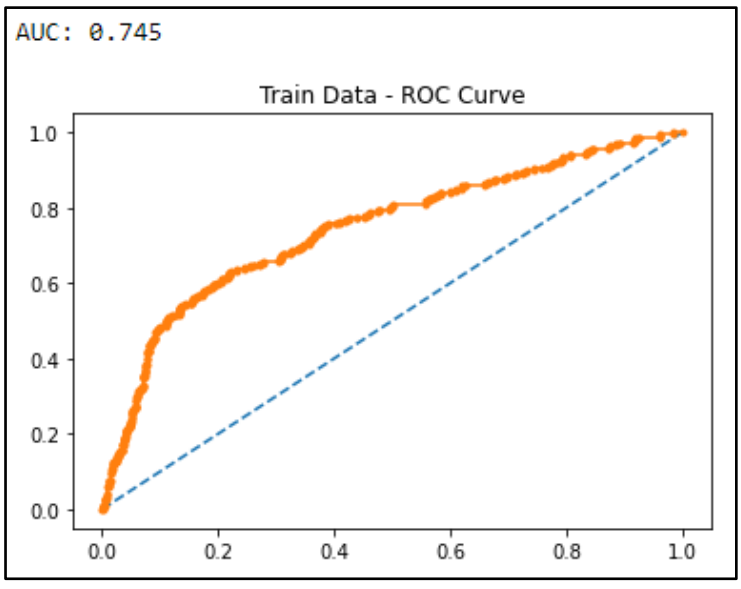
Confusion matrix on the training data:



Classification report of Train Data:

```
--------------------Classification Report - Train Data----
              precision    recall  f1-score   support

           0       0.89      1.00      0.94      2143
           1       0.50      0.01      0.02       259

    accuracy                           0.89      2402
   macro avg       0.70      0.50      0.48      2402
weighted avg       0.85      0.89      0.84      2402
```

ROC Curve for train data:



Below are the performance metrics on train data for LDA model with Smote:

```
LDA_With_Smote
--------------------Best Parameters--------------------
{'shrinkage': 'auto', 'solver': 'lsqr', 'tol': 1e-06}
```

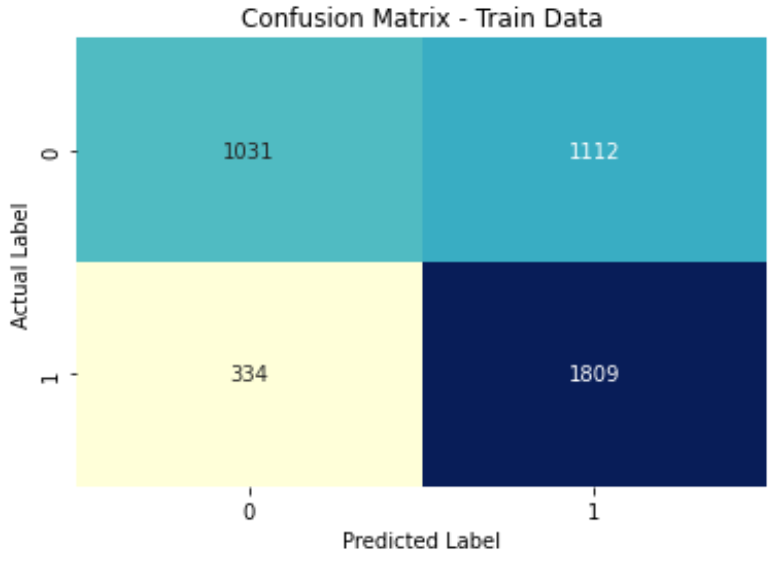
```
--------------------Best Model Params--------------------
LinearDiscriminantAnalysis(shrinkage='auto', solver='lsqr', tol=1e-06)
```

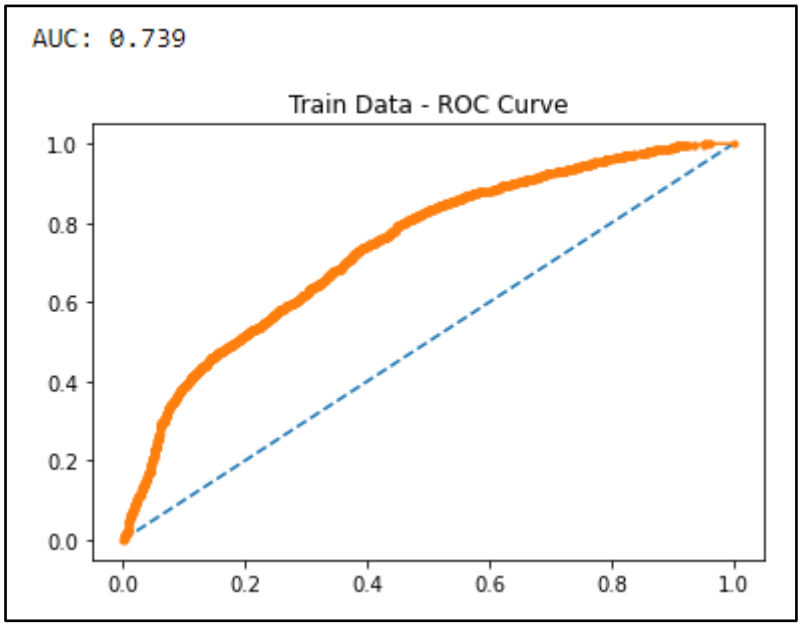
Figure 7: Performance metrics on train data for LDA model with Smote

Classification Report:

```
--------------------Classification Report - Train Data--
              precision    recall  f1-score   support

           0       0.76      0.48      0.59      2143
           1       0.62      0.84      0.71      2143

    accuracy                           0.66      4286
   macro avg       0.69      0.66      0.65      4286
weighted avg       0.69      0.66      0.65      4286
```

Confusion Matrix:





1.11. Validate the LDA Model on test Dataset and state the performance matrices. Also state interpretation from the model.

Solution:

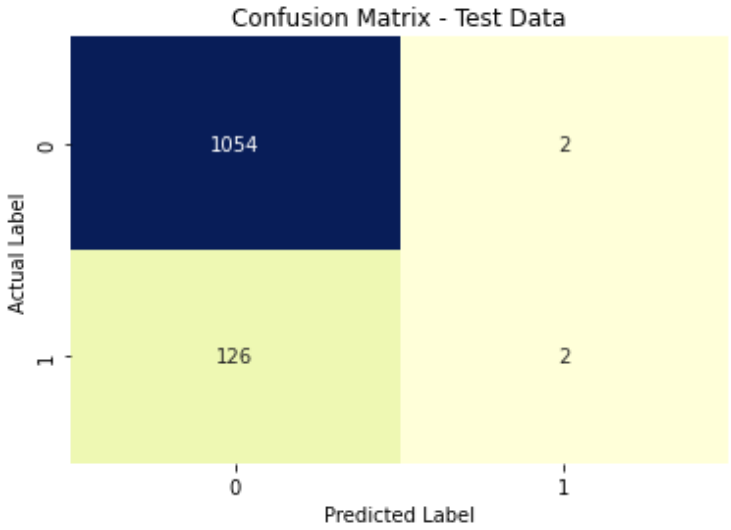
LDA Model on test data:

Both the LDA models were then evaluated on the test data. Below are the results for test data for LDA model without smote data.

Predicted Probability on test data:

	0	1
0	0.996328	0.003672
1	0.993790	0.006210
2	0.980624	0.019376
3	0.919425	0.080575
4	0.998789	0.001211
5	0.295149	0.704851
6	0.999714	0.000286
7	0.000226	0.999774
8	0.531435	0.468565
9	0.999695	0.000305

Confusion matrix on the test data:

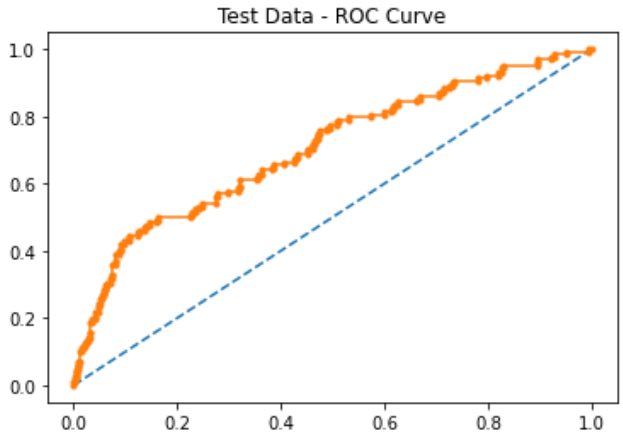


Classification report of Test Data:

	precision	recall	f1-score	support
0	0.89	1.00	0.94	1056
1	0.50	0.02	0.03	128
accuracy			0.89	1184
macro avg	0.70	0.51	0.49	1184
weighted avg	0.85	0.89	0.84	1184

ROC Curve for test data:

AUC: 0.704



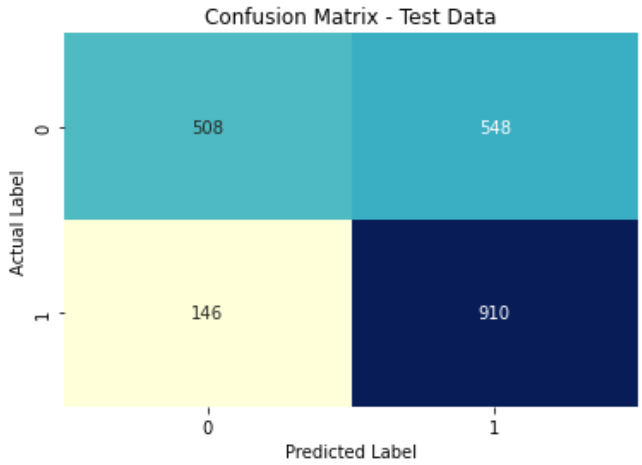
Below are the results for test data for LDA model with smote data:

Classification Report:

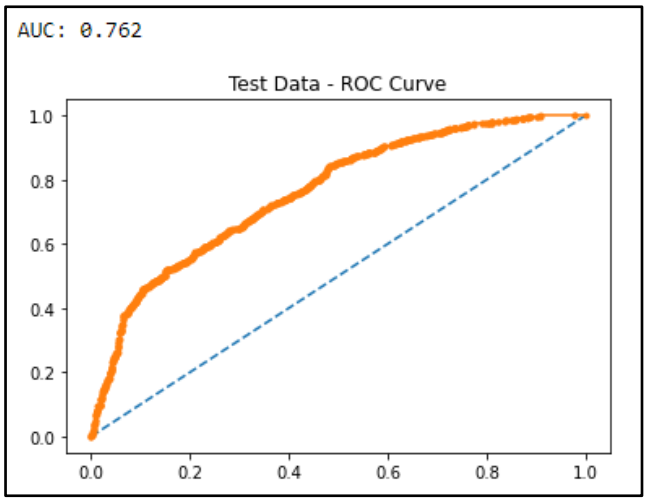
--------------------Classification Report - Test Data----

	precision	recall	f1-score	support
0	0.78	0.48	0.59	1056
1	0.62	0.86	0.72	1056
accuracy			0.67	2112
macro avg	0.70	0.67	0.66	2112
weighted avg	0.70	0.67	0.66	2112

Confusion Matrix:



ROC Curve:



Below is the comparison of various evaluation matric in tabular format for both the models i.e., with and without smote.

	Accuracy	Precision	Recall	F1	AUC
LDA_Train	0.892173	0.5	0.007722	0.015209	0.745186
LDA_Test	0.891892	0.5	0.015625	0.030303	0.703843

	Accuracy	Precision	Recall	F1	AUC
LDA_With_Smote_Train	0.662622	0.619308	0.844144	0.714455	0.739134
LDA_With_Smote_Test	0.671402	0.624143	0.861742	0.723946	0.762228

Comparing the two models, we can see LDA model without SMOTE data has higher accuracy on test data. However, it has really bad precision, recall and F1 scores compared to the other model.

On purely accuracy perspective model without smote data performs better. But in real world scenario model with smote will perform better given better recall and precision values.

1.12. Compare the performances of Logistics, Radom Forest and LDA models (include ROC Curve)

Solution:

Below is the comparison chart for all the models:

	Accuracy	Precision	Recall	F1	AUC
RandomForest_Train	0.980849	0.927711	0.891892	0.909449	0.993687
RandomForest_Test	0.975507	0.902439	0.867188	0.884462	0.989975
Logit_SM_Train	0.974604	0.863971	0.907336	0.885122	0.965672
RF_With_Smote_Train	0.974335	0.983357	0.965002	0.974093	0.997926
LogisticRegression_Unscaled_Train	0.970858	0.939535	0.779923	0.852321	0.962838
RF_With_Smote_Test	0.967803	0.978682	0.956439	0.967433	0.994474
Logit_SM_Test	0.963682	0.819549	0.851562	0.835249	0.960138
LogisticRegression_Unscaled_Test	0.956081	0.865385	0.703125	0.775862	0.94215
LogisticRegression_Scaled_Train	0.947127	0.958333	0.532819	0.684864	0.962358
Logit_SM_Train_SMOTE	0.946104	0.973736	0.916939	0.944484	0.98049
Logit_SM_Test_SMOTE	0.936553	0.961924	0.909091	0.934761	0.973314

LogisticRegression_Scaled_Test	0.934966	0.859155	0.476562	0.613065	0.947347
LogisticRegression_Scaled_With_Smote_Train	0.930705	0.91316	0.951937	0.932145	0.980486
LogisticRegression_With_Smote_Train	0.928138	0.910515	0.949603	0.929648	0.979997
LogisticRegression_Scaled_With_Smote_Test	0.923769	0.903517	0.948864	0.925635	0.973239
LogisticRegression_With_Smote_Test	0.921402	0.899461	0.948864	0.923502	0.973695
LDA_Train	0.892173	0.5	0.007722	0.015209	0.745186
LDA_Test	0.891892	0.5	0.015625	0.030303	0.703843
LDA_With_Smote_Test	0.671402	0.624143	0.861742	0.723946	0.762228
LDA_With_Smote_Train	0.662622	0.619308	0.844144	0.714455	0.739134

Table 3: Comparison chart for all the models

Looking at the above comparison chart we can see Random Forest model has surpassed all other models in terms of accuracy. Both Random Forest with and without smote feature in the top 2 models.

While Random Forest was the best performing model, LDA models were the worst performers of all. LDA with or without were both really bad in terms of accuracies, LDA with smote being the worst of the lot.

Best Model on Train Data:

	Accuracy	**Precision**	**Recall**	**F1**	**AUC**
RandomForest_Train	0.980849	0.927711	0.891892	0.909449	0.993687
Logit_SM_Train	0.974604	0.863971	0.907336	0.885122	0.965672
RF_With_Smote_Train	0.974335	0.983357	0.965002	0.974093	0.997926
LogisticRegression_Unscaled_Train	0.970858	0.939535	0.779923	0.852321	0.962838
LogisticRegression_Scaled_Train	0.947127	0.958333	0.532819	0.684864	0.962358
Logit_SM_Train_SMOTE	0.946104	0.973736	0.916939	0.944484	0.98049
LogisticRegression_Scaled_With_Smote_Train	0.930705	0.91316	0.951937	0.932145	0.980486
LogisticRegression_With_Smote_Train	0.928138	0.910515	0.949603	0.929648	0.979997
LDA_Train	0.892173	0.5	0.007722	0.015209	0.745186
LDA_With_Smote_Train	0.662622	0.619308	0.844144	0.714455	0.739134

Table 4: Best model on Training data

Best Model on Test Data:

	Accuracy	Precision	Recall	F1	AUC
RandomForest_Test	0.975507	0.902439	0.867188	0.884462	0.989975
RF_With_Smote_Test	0.967803	0.978682	0.956439	0.967433	0.994474
Logit_SM_Test	0.963682	0.819549	0.851562	0.835249	0.960138
LogisticRegression_Unscaled_Test	0.956081	0.865385	0.703125	0.775862	0.94215
Logit_SM_Test_SMOTE	0.936553	0.961924	0.909091	0.934761	0.973314
LogisticRegression_Scaled_Test	0.934966	0.859155	0.476562	0.613065	0.947347
LogisticRegression_Scaled_With_Smote_Test	0.923769	0.903517	0.948864	0.925635	0.973239
LogisticRegression_With_Smote_Test	0.921402	0.899461	0.948864	0.923502	0.973695
LDA_Test	0.891892	0.5	0.015625	0.030303	0.703843
LDA_With_Smote_Test	0.671402	0.624143	0.861742	0.723946	0.762228

Table 5: Best model on Test data

Below is a comparison of different evaluation metrics for all the models on the training data set.

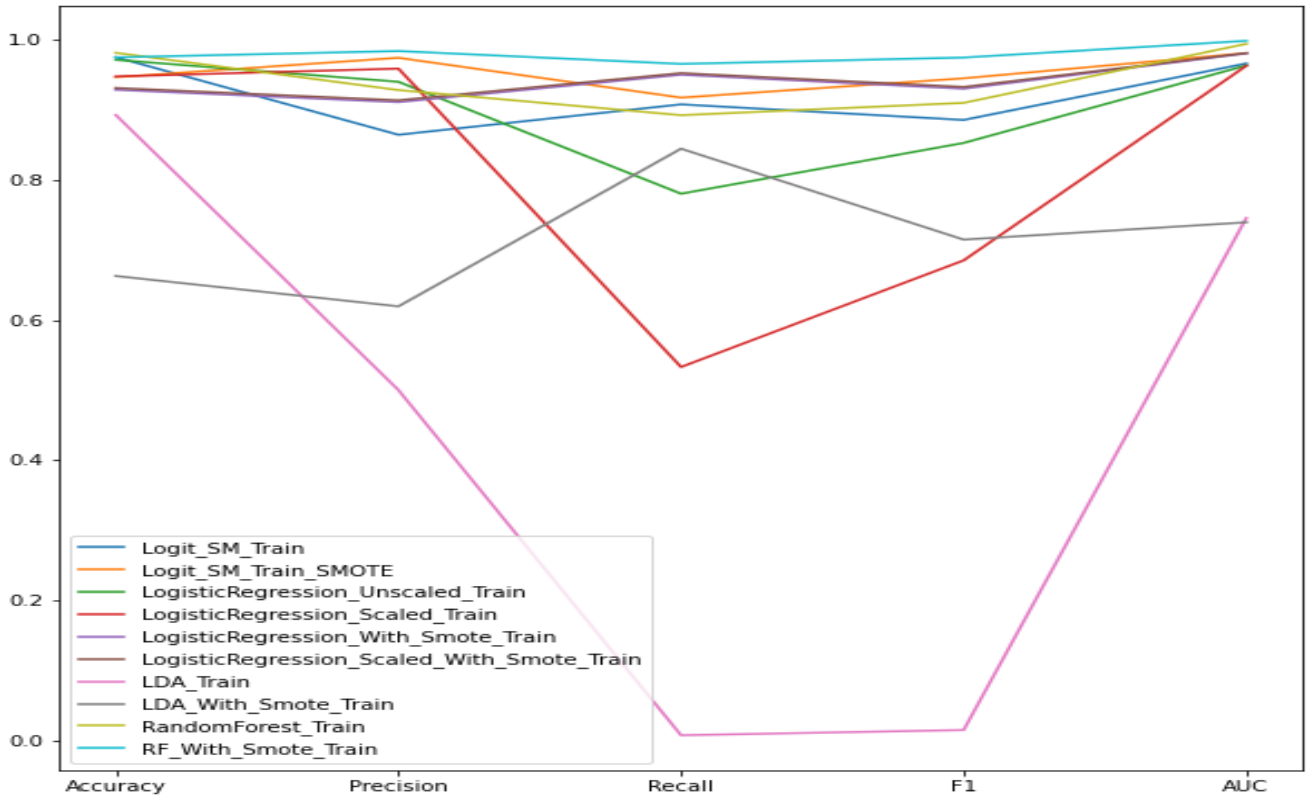


Figure 8: Comparison of different evaluation metrics for all the models - Training data

Looking at the above chart, we clearly see the blue line indicating "Random Forest With SMOTE" being o n top of all other models, consistently for all the evaluation parameters.

While the pink line for LDA without smote is slightly higher than LDA with smote in terms of accuracy and Auc score, it slips down significantly on all other parameters, making it the worst performing model of the lot.

Below is a comparison of different evaluation metrics for all the models on the test data set.

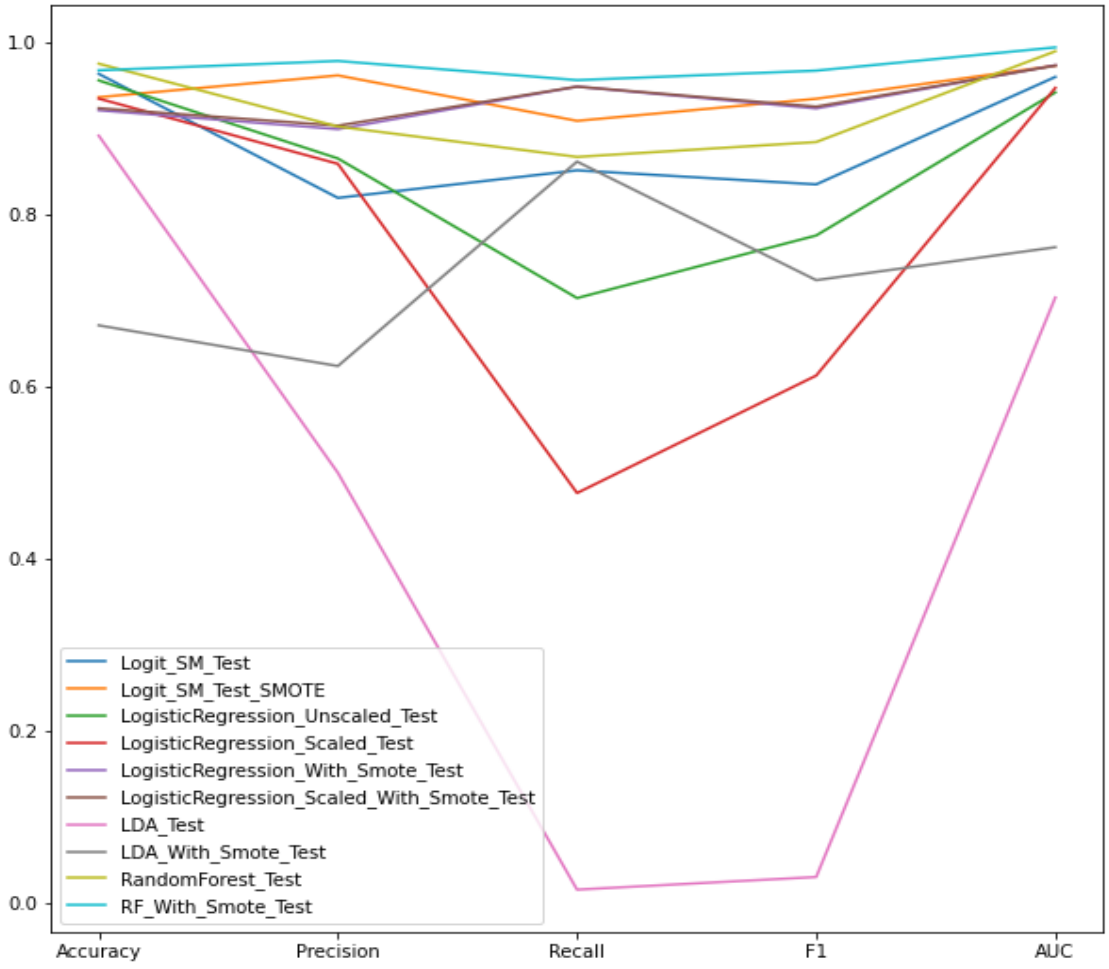


Figure 9: Comparison of different evaluation metrics for all the models - Test data

On the test data set, which is the deciding factor, here too we see Random Forest with SMOTE indicated by blue line performing the best on almost all the fronts. We can see the blue line dip only very slightly o n accuracy compared to Random Forest without smote, on all other fronts the blue line stays on the top and is the clear winner amongst all the models.

Random Forest model with smote dataset is the best model amongst all the models with 96.7% accuracy, 97.86% precision, 95.64% recall and F1 score of 0.9674 respectively.

1.13. State Recommendations from the above models.

Solution:

With increasing amount of data, companies and industries try to remain competitive by keeping themselves ahead of the curve. By analysing huge amounts of financial data, companies are able to obtain valuable information to determine their strategic plans such as risk control, crisis management or growth management. Logistic regression, Random Forest and LDA models have been employed in predicting the defaulters of companies.

Using the information gained from above exercise, we can say Random Forest with smote data is the best model. We also looked at the coefficients derived from the best Logit model built using Stats model to derive some more insights.

LDA Model without Smote:

Logit Regression Results

Dep. Variable:	Default	No. Observations:	2402
Model:	Logit	Df Residuals:	2397
Method:	MLE	Df Model:	4
Date:	Sat, 15 Oct 2022	Pseudo R-squ.:	0.4909
Time:	23:42:10	Log-Likelihood:	-418.18
converged:	True	LL-Null:	-821.36
Covariance Type:	nonrobust	LLR p-value:	3.232e-173

	coef	std err	z	P>|z|	[0.025	0.975]
Intercept	-1.0835	0.096	-11.294	0.000	-1.272	-0.895
Book_Value_Unit_Curr	-0.1147	0.008	-13.749	0.000	-0.131	-0.098
Cash_Flow_From_Operating_Activities	-0.0008	0.000	-3.761	0.000	-0.001	-0.000
Interest_Cover_Ratio_Latest	-0.0015	0.001	-2.603	0.009	-0.003	-0.000
Value_of_Output_by_Gross_Block	-0.0176	0.005	-3.603	0.000	-0.027	-0.008

Figure 10: LDA Model without Smote

LDA Model with Smote:

Logit Regression Results

Dep. Variable:	Default	No. Observations:	2402
Model:	Logit	Df Residuals:	2376
Method:	MLE	Df Model:	25
Date:	Sat, 15 Oct 2022	Pseudo R-squ.:	0.5160
Time:	23:42:53	Log-Likelihood:	-397.55
converged:	True	LL-Null:	-821.36
Covariance Type:	nonrobust	LLR p-value:	1.072e-162

	coef	std err	z	P>|z|	[0.025	0.975]
Intercept	-0.8530	0.123	-6.940	0.000	-1.094	-0.612
Net_Working_Capital	-0.0002	0.000	-0.999	0.318	-0.000	0.000
Current_Assets	0.0001	4.95e-05	2.349	0.019	1.92e-05	0.000
Other_Income	-0.0058	0.004	-1.375	0.169	-0.014	0.002
Selling_Cost	0.0048	0.003	1.461	0.144	-0.002	0.011
Revenue_expenses_in_forex	0.0003	0.000	0.851	0.395	-0.000	0.001
Book_Value_Unit_Curr	-0.1158	0.009	-12.683	0.000	-0.134	-0.098
Market_Capitalisation	-0.0009	0.001	-1.495	0.135	-0.002	0.000
CEPS_annualised_Unit_Curr	0.0268	0.015	1.792	0.073	-0.003	0.056
Cash_Flow_From_Operating_Activities	-0.0032	0.001	-3.151	0.002	-0.005	-0.001
Cash_Flow_From_Investing_Activities	-0.0033	0.001	-3.258	0.001	-0.005	-0.001
ROG_Capital_Employed_perc	-0.0006	0.001	-0.869	0.385	-0.002	0.001
ROG_Gross_Block_perc	-0.0023	0.002	-0.991	0.321	-0.007	0.002
ROG_Net_Sales_perc	-0.0006	0.001	-1.275	0.202	-0.002	0.000
ROG_PAT_perc	-2.117e-05	1.92e-05	-1.104	0.269	-5.87e-05	1.64e-05
ROG_Revenue_earnings_in_forex_perc	-0.0069	0.003	-2.427	0.015	-0.012	-0.001
Current_Ratio_Latest	-0.0131	0.008	-1.718	0.086	-0.028	0.002
Inventory_Ratio_Latest	-0.0013	0.002	-0.668	0.504	-0.005	0.003
Debtors_Ratio_Latest	-0.0012	0.002	-0.709	0.478	-0.005	0.002
Total_Asset_Turnover_Ratio_Latest	0.0490	0.039	1.245	0.213	-0.028	0.126
Interest_Cover_Ratio_Latest	-0.0014	0.001	-2.340	0.019	-0.003	-0.000
CPM_perc_Latest	-0.0001	0.000	-0.924	0.356	-0.000	0.000
Debtors_Velocity_Days	-6.909e-05	3.97e-05	-1.738	0.082	-0.000	8.8e-06
Creditors_Velocity_Days	8.76e-06	5.91e-06	1.483	0.138	-2.82e-06	2.03e-05
Value_of_Output_by_Total_Assets	-0.1551	0.122	-1.276	0.202	-0.393	0.083
Value_of_Output_by_Gross_Block	-0.0148	0.005	-2.896	0.004	-0.025	-0.005

Figure 11 - LDA Model with Smote

From the above analysis, we can infer below business insights. Following things should be kept in mind while investing in these companies.

1) Lower the Book_value_unit_curr i.e., Net assets, higher is the chance of a default, which would mean the net worth next year for this company is expected to be negative.

2) Lower the CEPS_annualised_Unit_Curr i.e., Cash earning per share, higher is the change of a default.

3) Higher the Curr_Ratio_Latest, i.e., the company's ability to pay short term dues, lower are its chances of defaulting or having a negative net worth in the next year.

4) Higher the Interest_Cover_Ratio_Latest lower the chances of default. Which means easier the company is able to pay the interest on its outstanding debt, lower are its chances to default. Curr_Ratio_Latest is most important criteria amongst the above parameters, while Interest_Cover_Ratio_Latest is the least important when considering only these 4 parameters. However, all these 4 parameters remain important compared to the other variables in the data set.

Business problem 2 - Market Risk

The dataset contains 6 years of information (weekly stock information) on the stock prices of 10 different Indian Stocks. Calculate the mean and standard deviation on the stock returns and share insights.

2.1. Draw Stock Price Graph (Stock Price vs Time) for any 2 given stocks with inference.

Solution:

Exploratory data analysis:

Head of the dataset (after renaming the column headers)

	Date	Infosys	Indian_Hotel	Mahindra_N_Mahindra	Axis_Bank	SAIL	Shree_Cement	Sun_Pharma	Jindal_Steel	Idea_Vodafone	Jet_Airways
0	31-03-2014	264	69	455	263	68	5543	555	298	83	278
1	07-04-2014	257	68	458	276	70	5728	610	279	84	303
2	14-04-2014	254	68	454	270	68	5649	607	279	83	280
3	21-04-2014	253	68	488	283	68	5692	604	274	83	282
4	28-04-2014	256	65	482	282	63	5582	611	238	79	243

Shape of the dataset:

The number of rows (observations) is 314
The number of columns (variables) is 11

Information of the dataset:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 314 entries, 0 to 313
Data columns (total 11 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   Date                 314 non-null    object
 1   Infosys              314 non-null    int64
 2   Indian_Hotel         314 non-null    int64
 3   Mahindra_N_Mahindra  314 non-null    int64
 4   Axis_Bank            314 non-null    int64
 5   SAIL                 314 non-null    int64
 6   Shree_Cement         314 non-null    int64
 7   Sun_Pharma           314 non-null    int64
 8   Jindal_Steel         314 non-null    int64
 9   Idea_Vodafone        314 non-null    int64
 10  Jet_Airways          314 non-null    int64
dtypes: int64(10), object(1)
memory usage: 27.1+ KB
```

Summary statistics of the dataset:

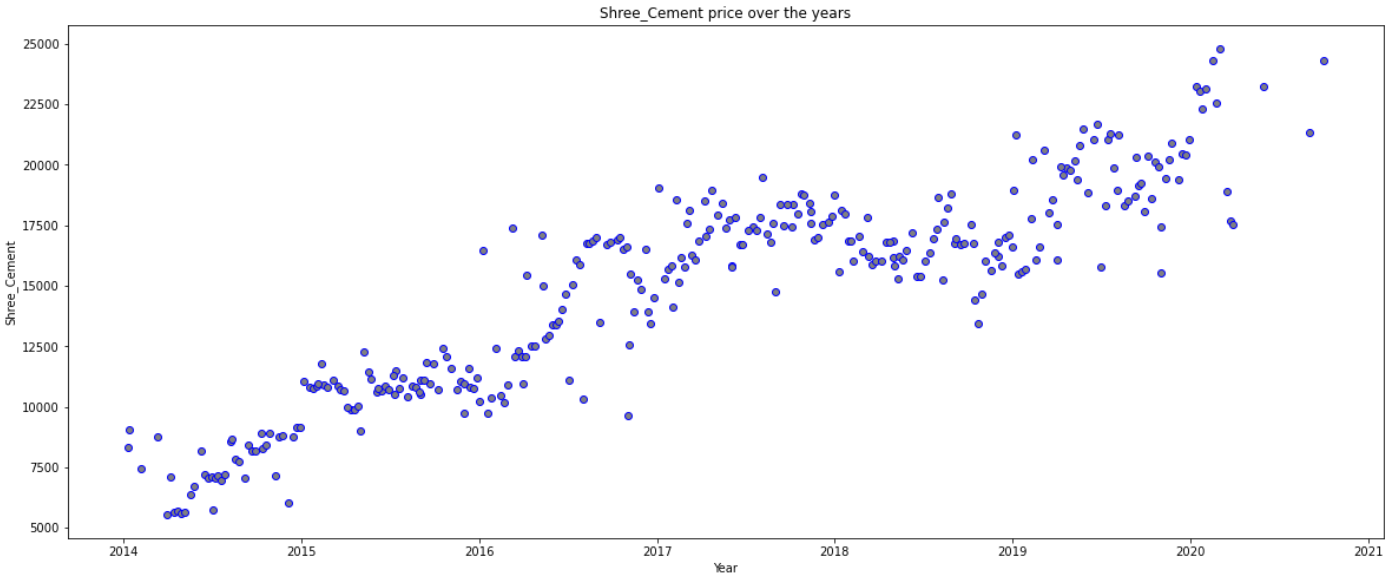
	Infosys	Indian_Hotel	Mahindra_N_Mahindra	Axis_Bank	SAIL	Shree_Cement	Sun_Pharma	Jindal_Steel	Idea_Vodafone	Jet_Airways
count	314.000000	314.000000	314.000000	314.000000	314.000000	314.000000	314.000000	314.000000	314.000000	314.000000
mean	511.340764	114.560510	636.678344	540.742038	59.095541	14806.410828	633.468153	147.627389	53.713376	372.659236
std	135.952051	22.509732	102.879975	115.835569	15.810493	4288.275085	171.855893	65.879195	31.248985	202.262668
min	234.000000	64.000000	284.000000	263.000000	21.000000	5543.000000	338.000000	53.000000	3.000000	14.000000
25%	424.000000	96.000000	572.000000	470.500000	47.000000	10952.250000	478.500000	88.250000	25.250000	243.250000
50%	466.500000	115.000000	625.000000	528.000000	57.000000	16018.500000	614.000000	142.500000	53.000000	376.000000
75%	630.750000	134.000000	678.000000	605.250000	71.750000	17773.250000	785.000000	182.750000	82.000000	534.000000
max	810.000000	157.000000	956.000000	808.000000	104.000000	24806.000000	1089.000000	338.000000	117.000000	871.000000

2.1 Draw Stock Price Graph (Stock Price vs Time) for any 2 given stocks with inference.

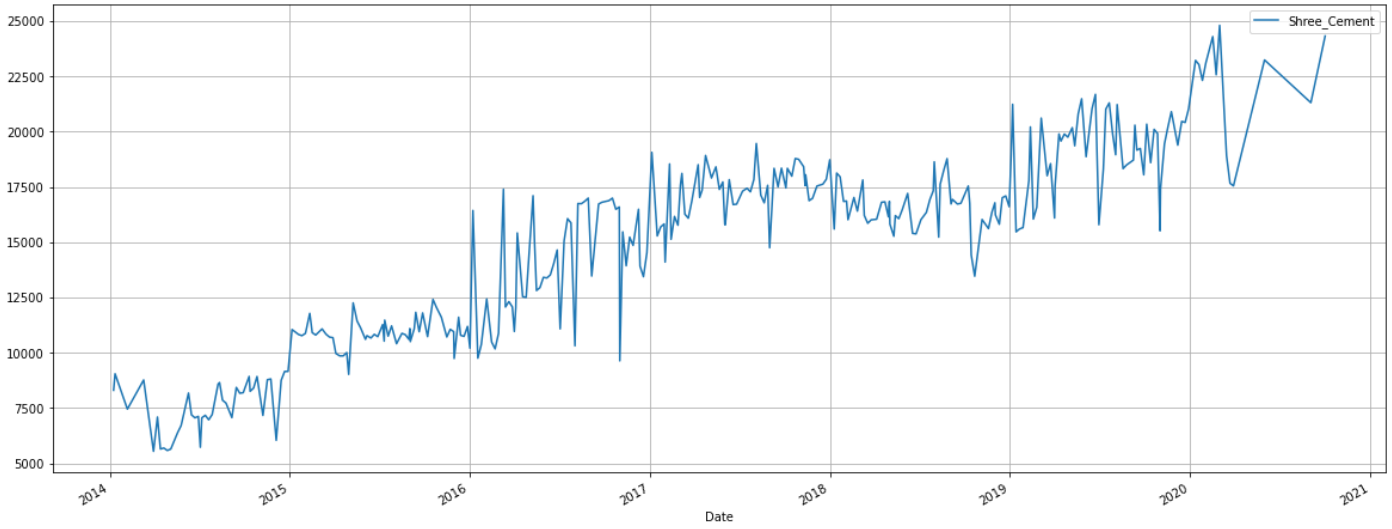
Solution:

Taken two stocks 'Shree_Cement' and 'Idea_Vodafone' to explain the stock price graph.

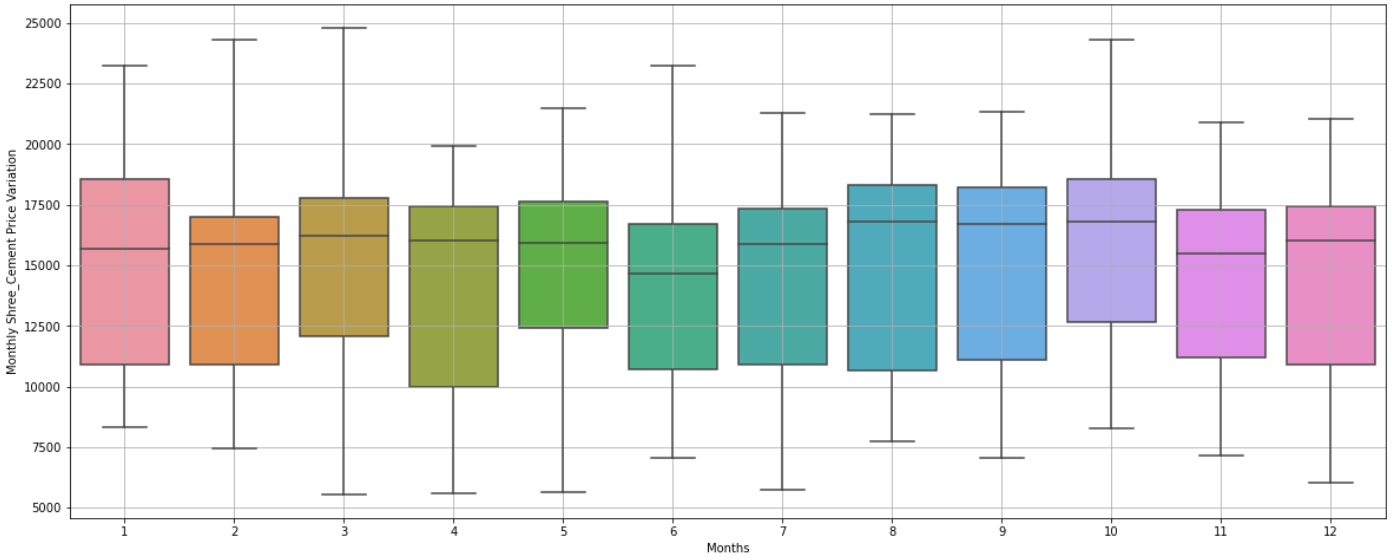
Shree cement price over time:



This scatterplot shows the trend of Shree_Cement price over time. We can observe an upward trend over the years. Price was at 5000 during 2014 and it has increased to 25000 at 2020.

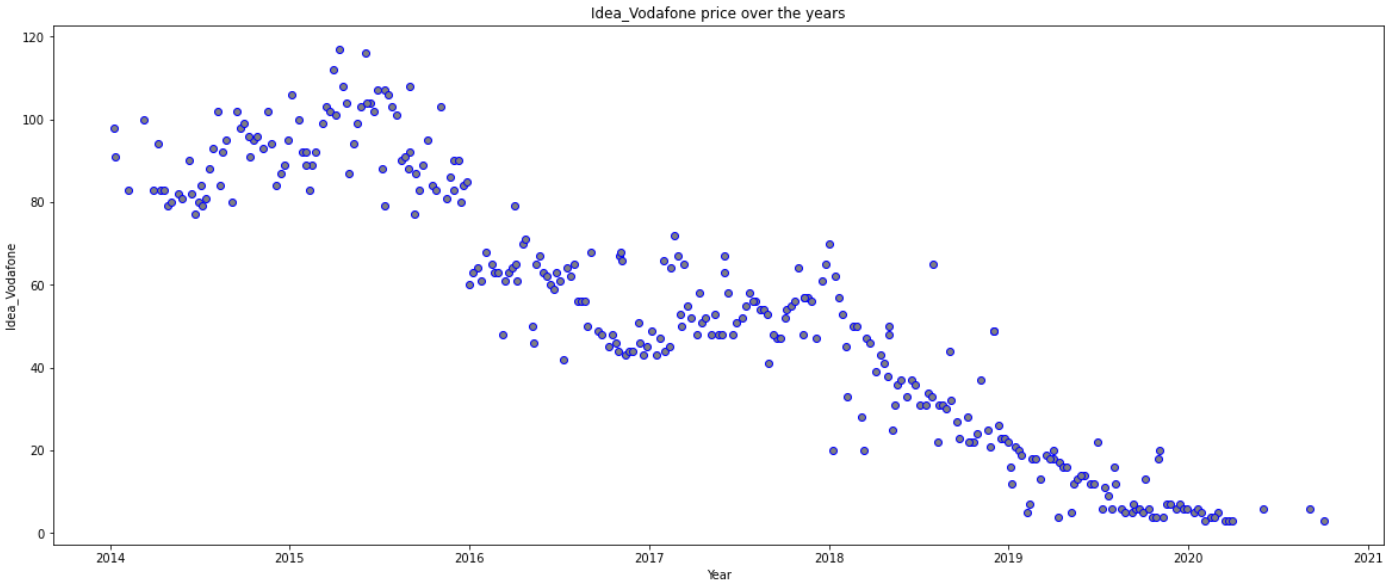


Month wise plot to check for seasonality:

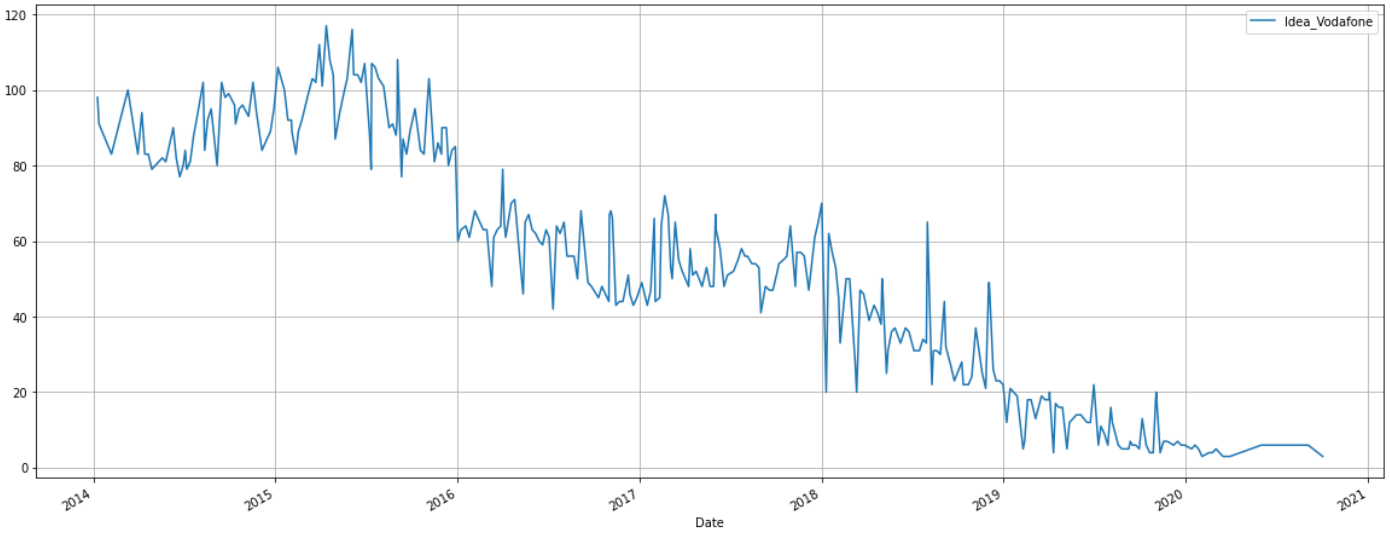


There is no visible seasonality present in the dataset for Shree_Cement price since the median is in and around 15000. But we can observe that the range of the prices are widely spread.

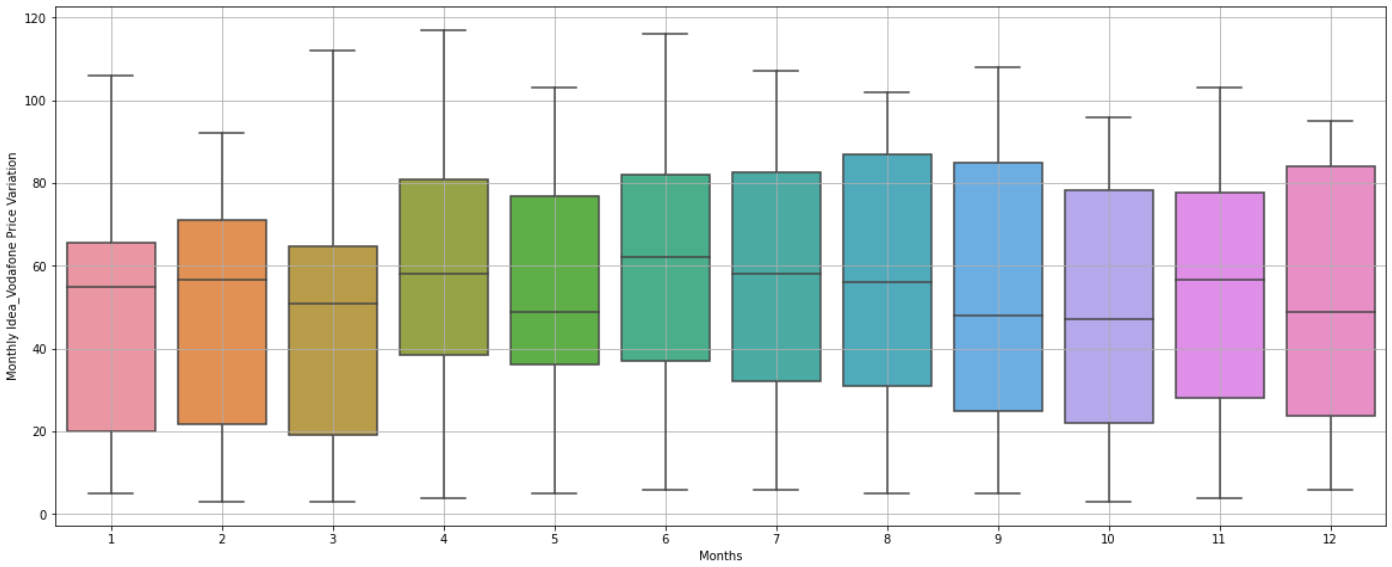
Idea_Vodafone price over time:



This scatterplot shows the trend of Shree_Cement price over time. We can observe a downward trend over the years. Price was at 100 during 2014, peaks are available at 120 during 2015 end and it has decreased to 0 at 2019 itself.



Month wise plot to check for seasonality:



There is no visible seasonality present in the dataset for Idea_Vodafone price since the median is between the range 50-60. Here also, we can observe that the range of the prices are widely spread.

2.2 Calculate Returns for all stocks with inference.

Solution:

Returns are the change in stock price as a proportion of what the stock price was in the earlier time period. Calculating returns by taking the log is preferred when we look at multiple time periods.

	Infosys	Indian_Hotel	Mahindra_N_Mahindra	Axis_Bank	SAIL	Shree_Cement	Sun_Pharma	Jindal_Steel	Idea_Vodafone	Jet_Airways
0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	-0.026873	-0.014599	0.006572	0.048247	0.028988	0.032831	0.094491	-0.065882	0.011976	0.086112
2	-0.011742	0.000000	-0.008772	-0.021979	-0.028988	-0.013888	-0.004930	0.000000	-0.011976	-0.078943
3	-0.003945	0.000000	0.072218	0.047025	0.000000	0.007583	-0.004955	-0.018084	0.000000	0.007117
4	0.011788	-0.045120	-0.012371	-0.003540	-0.076373	-0.019515	0.011523	-0.140857	-0.049393	-0.148846

A positive return is the profit, or money made, on the stock. Likewise, a negative return represents the loss or money lost on the stock. This is an important metric to calculate how well the stock has performed.

Summary statistics of the return's dataset:

	Infosys	Indian_Hotel	Mahindra_N_Mahindra	Axis_Bank	SAIL	Shree_Cement	Sun_Pharma	Jindal_Steel	Idea_Vodafone	Jet_Airways
count	313.000000	313.000000	313.000000	313.000000	313.000000	313.000000	313.000000	313.000000	313.000000	313.000000
mean	0.002794	0.000266	-0.001506	0.001167	-0.003463	0.003681	-0.001455	-0.004123	-0.010608	-0.009548
std	0.035070	0.047131	0.040169	0.045828	0.062188	0.039917	0.045033	0.075108	0.104315	0.097972
min	-0.167300	-0.236389	-0.285343	-0.284757	-0.251314	-0.129215	-0.179855	-0.283768	-0.693147	-0.458575
25%	-0.014514	-0.023530	-0.020884	-0.022473	-0.040822	-0.019546	-0.020699	-0.049700	-0.045120	-0.052644
50%	0.004376	0.000000	0.001526	0.001614	0.000000	0.003173	0.001530	0.000000	0.000000	-0.005780
75%	0.024553	0.027909	0.019894	0.028522	0.032790	0.029873	0.023257	0.037179	0.024391	0.036368
max	0.135666	0.199333	0.089407	0.127461	0.309005	0.152329	0.166604	0.243978	0.693147	0.300249

2.3 Calculate Stock Means and Standard Deviation for all stocks with inference.

Solution:

Stock mean – Average: The mean return of the selected stocks is intended to represent the behaviour of the market and to report the composite change in prices of the stocks.

```
Infosys                0.002794
Indian_Hotel           0.000266
Mahindra_N_Mahindra   -0.001506
Axis_Bank              0.001167
SAIL                  -0.003463
Shree_Cement           0.003681
Sun_Pharma            -0.001455
Jindal_Steel          -0.004123
Idea_Vodafone         -0.010608
Jet_Airways           -0.009548
dtype: float64
```

Each average reflects the general movement of each stock and serves as a benchmark for the performance of individual stocks in its sphere. Positive mean value is when we can see increase in stock prices than the initial and negative mean value is the decrease in stock prices.

Stock standard deviation – Volatility: It is a statistical measure of volatility, measuring how widely prices are dispersed from the average price. If prices trade in a narrow trading range, the standard deviation will return a low value that indicates low volatility. Conversely, if prices swing wildly up and down, then standard deviation returns a high value that indicates high volatility.

```
Infosys                 0.035070
Indian_Hotel            0.047131
Mahindra_N_Mahindra     0.040169
Axis_Bank               0.045828
SAIL                    0.062188
Shree_Cement            0.039917
Sun_Pharma              0.045033
Jindal_Steel            0.075108
Idea_Vodafone           0.104315
Jet_Airways             0.097972
dtype: float64
```

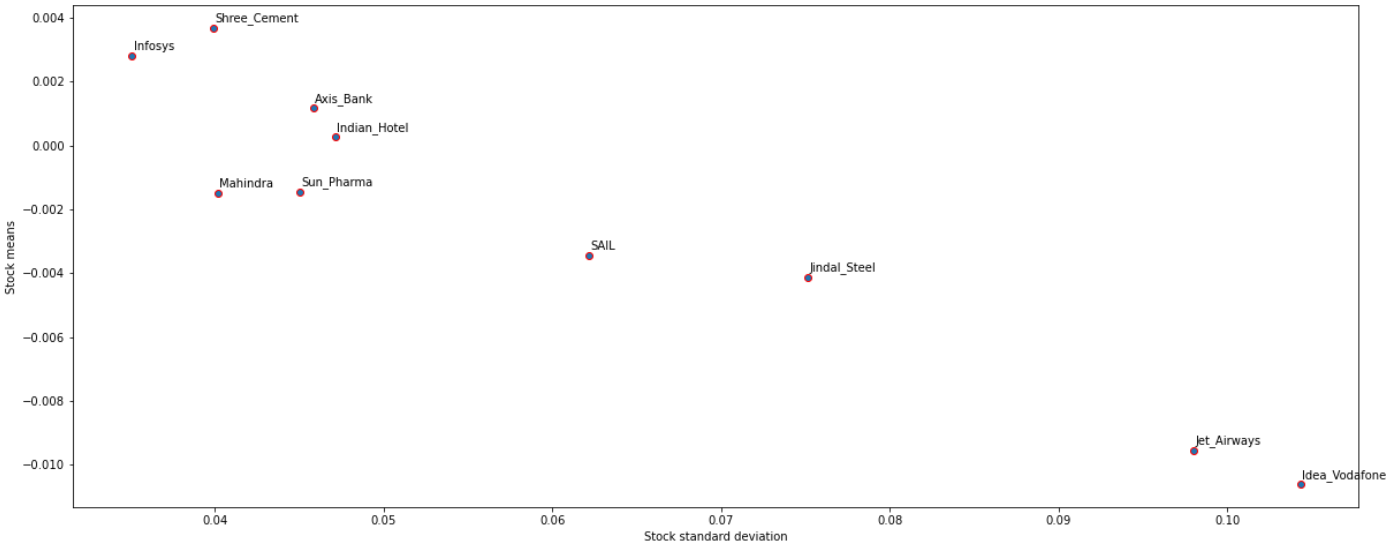
Highest standard deviation among the given companies is for Idea_Vodafone. This can be observed from the graphs before where can see a steep decrease in stock prices from 2018.

Created a dataframe with the stock mean values and stock standard deviation.

	Average	Volatility
Infosys	0.002794	0.035070
Indian_Hotel	0.000266	0.047131
Mahindra_N_Mahindra	-0.001506	0.040169
Axis_Bank	0.001167	0.045828
SAIL	-0.003463	0.062188
Shree_Cement	0.003681	0.039917
Sun_Pharma	-0.001455	0.045033
Jindal_Steel	-0.004123	0.075108
Idea_Vodafone	-0.010608	0.104315
Jet_Airways	-0.009548	0.097972

2.4. Draw a plot of Stock Means vs Standard Deviation and state your inference.

Solution:



Here is a combined plot of volatility vs average of all the given stocks.

Standard deviation is a measure of risk. It is used to capture the uncertainty of variables. If the value is high, then the returns are more uncertain and if the value is low, then the returns are less uncertain and risky.

From the above plot we can see that the stocks of Idea_Vodafone and Jet_Airways have the highest standard deviation and low mean values which means these stocks are a risky investment since they are in the declining stage.

The stocks of Shree_Cement, Infosys, Axis_Bank and Indian_Hotel have positive means which means the stock prices are increasing and they have correspondingly less standard deviation which represents that these stocks are less risky to invest in.

Comparing Shree_Cement and Mahindra which has comparative standard deviation values but the mean of Shree_Cement stocks are higher than that of Mahindra. At this point of time, Mahindra might be performing less as compared to other stock. Same goes with Mahindra and Sun pharma, these two stocks have same average stock prices but the volatility of Sun Pharma is higher than that of Mahindra, hence concluding that Sun Pharma might be a risky option.

2.5. Conclusion and Recommendations.

Solution:

Arranging the stocks by increasing order of volatility:

	Average	Volatility
Infosys	0.002794	0.035070
Shree_Cement	0.003681	0.039917
Mahindra_N_Mahindra	-0.001506	0.040169
Sun_Pharma	-0.001455	0.045033
Axis_Bank	0.001167	0.045828
Indian_Hotel	0.000266	0.047131
SAIL	-0.003463	0.062188
Jindal_Steel	-0.004123	0.075108
Jet_Airways	-0.009548	0.097972
Idea_Vodafone	-0.010608	0.104315

Simple definition of volatility is a reflection of the degree to which price moves A stock with a price that fluctuates wildly—hits new highs and lows or moves erratically—is considered highly volatile. A stock that maintains a relatively stable price has low volatility. A highly volatile stock is inherently riskier, but that risk cuts both ways. When investing in a volatile security, the chance for success is increased as much as the risk of failure. For this reason, we have to keep in mind the financial position of the company, performance and other metrics to determine what to invest in.

We can observe that Idea Vodafone and Jet Airways are the highly volatile stocks in the decreasing trend since the prices of these stocks have reduced drastically because of the greater number of selling movements involved rather than buying. As these companies are out of market now, they are not an ideal option for investing.

On the other hand, Infosys and Shree Cements are less volatile but their price is now showing increasing trend and selling of these stocks would be ideal since it can come down any time after this. Investing in a portfolio of first four stocks can be a better option since averaging out the prices gives the benefits of diversification.