FINANCE & RISK ANALYTICS

(Milestone 2)

Mohamed Rifaz Ali K S

PGP-DSBA Online

May’ 22

Contents

List of Figures

Problem: Probability at default

Introduction

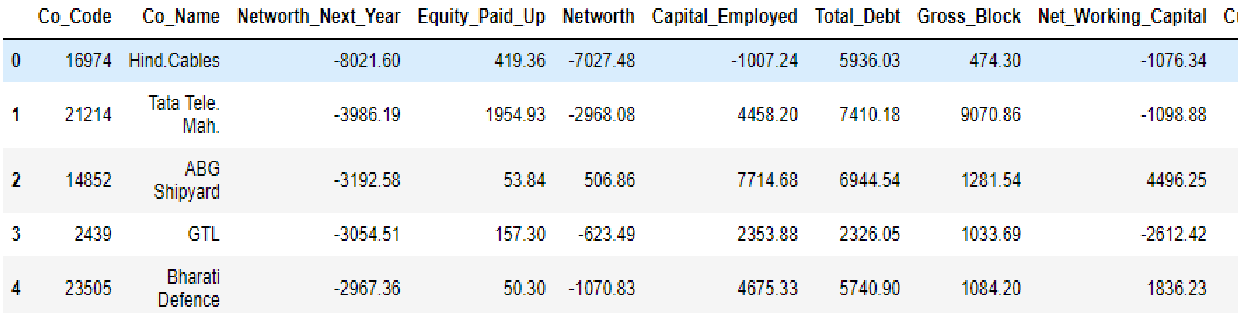
Businesses or companies can fall prey to default if they are not able to keep up their debt obligations. Defaults will lead to a lower credit rating for the company which in turn reduces its chances of getting credit in the future and may have to pay higher interests on existing debts as well as any 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.

About Data

- The number of rows (observations) is 3586 **and t**he number of columns (variables) is 67**.**
- There are very few missing values present in the entire dataset. By default, **numerical &** categorical columns properly mapped**.**
- **Default value can be created using** Networth Next Year **variable.**
- **Total cell size is** 240262 **and the total missing value is 118.** It is just 0.05% of the missing values present in the dataset.
- **For the Inventory velocity days variable, only** 3% of the total is missing**.**
- **There is one bad data in the Inventory velocity day column and it is in negative. This is definitely data entry issue** -199 can be **substituted** to 199**.**



	Co_Code	Co_Name	Networth_Next_Year	Equity_Paid_Up	Networth	Capital_Employed	Total_Debt	Gross_Block	Net_Working_Capital	C
0	16974	Hind.Cables	-8021.60	419.36	-7027.48	-1007.24	5936.03	474.30	-1076.34	
1	21214	Tata Tele. Mah.	-3986.19	1954.93	-2968.08	4458.20	7410.18	9070.86	-1098.88	
2	14852	ABG Shipyard	-3192.58	53.84	506.86	7714.68	6944.54	1281.54	4496.25	
3	2439	GTL	-3054.51	157.30	-623.49	2353.88	2326.05	1033.69	-2612.42	
4	23505	Bharati Defence	-2967.36	50.30	-1070.83	4675.33	5740.90	1084.20	1836.23	

Figure 1. Sample Company Finance Dataset

Outlier Treatment

Almost, all the variables contain outliers.

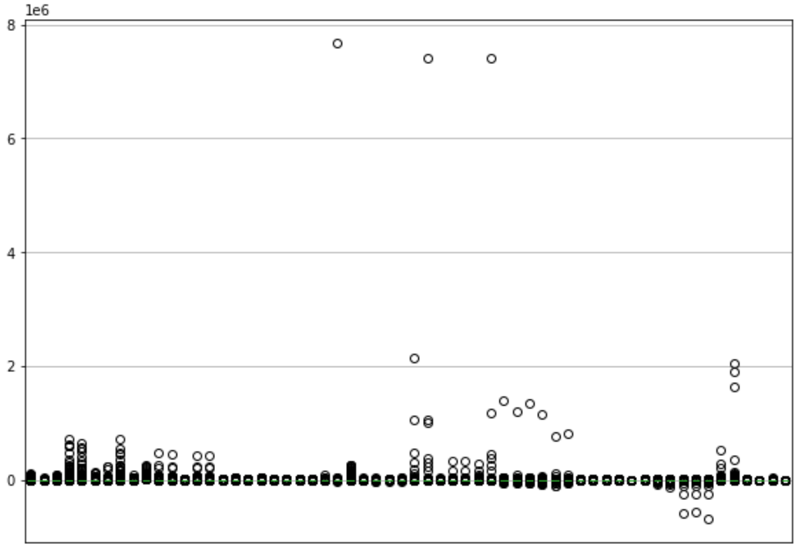


Figure 2. Boxplot to check the outliers

- Lets use the capping method to treat the outliers.
- Out of 3585 records, there are just 388(11% of the) records with default status. Data is highly imbalanced.
- We have 42449 outliers in total (i.e., 18% of the data's are outliers), of which 19% (~ 73 in number) are with a default status 1. As such we have only 388 default instances in our data therefore dropping these 73 records. i.e., 19% will not be a good option for this dataset.
- Before treating the outliers, let's include only the significant columns and remove the unwanted columns from the dataframe.
- The variables such as Revenue_earnings_in_forex, Revenue_expenses_in_forex, Capital_expenses_in_forex, ROG_Revenue_earnings_in_forex_perc, ROG_Revenue_expenses_in_forex_perc, ROG_Market_Capitalisation_perc do not have much variablility in the values, their 25%, 50% and 75% are just constant as 0. It might not be useful for our model building approach. We can keep Book Value (Unit Curr) column and drop Book Value (Adj.) (Unit Curr) column because both the columns are mimic to each other. Let's drop these redundant variables from the dataframe.

- For the remaining numerical columns, let’s calculate the Inter Quartile range between 0.25 and 0.75 and treat the lower and upper range range values.

  Q1,Q3=np.percentile(col,[25,75])

  IQR=Q3-Q1

  *lower_range= Q1-(1.5 * IQR)*

  *upper_range= Q3+(1.5 * IQR)*

- After treating the outliers, boxplots of the numerical variables looks neat.

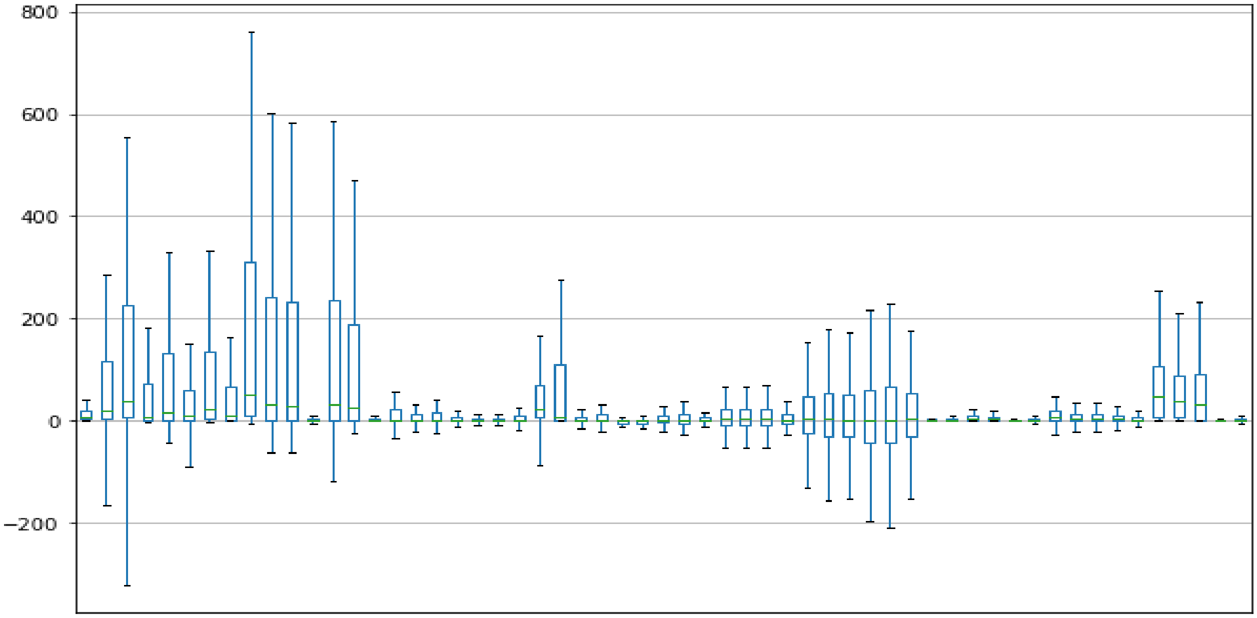


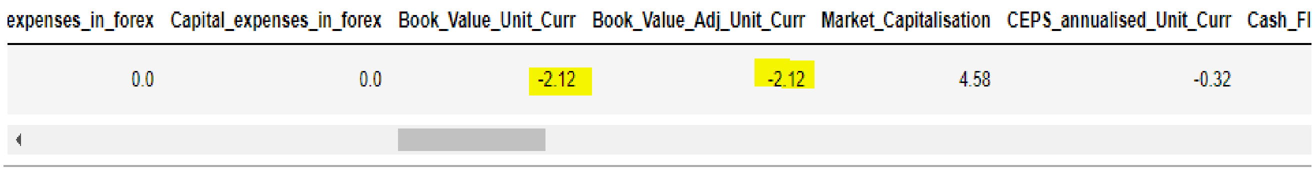
Figure 3. Boxplot after outlier treatments

Missing Value Treatment

- There are few columns with just 1 missing value in it.
- There are very few missing values in the entire dataset and present only in few columns.
- Total cell size is 240195 and Total missing values are 118. It is just 0.05% of the missing values present in the entire dataset.
- There is one row contains 11 missing values in it. We can identify and drop the row. The company named ‘**G M Breweries (Co_code=3240)**’ missed to provide the much financial information.

- After dropping the ‘G M Breweries’ company data, it looks much better now.

- Only 2 columns have missing values now. The columns are **Inventory_Velocity_Day** and **Book_Value_Adj_Unit_Curr.**
- **Inventory_Velocity_Day** – 103 missing values. Approximately, 3% of this column contains missing entries.
- Inventory_Velocity_Day has the data entry issue with the value -199, can be replaced with 199 days.
- Upon reviewing, **31%** of the company needs 0 days to convert the inventory into sales. So, let's impute the missing entries with 0 days using *fillna(value=0)* method.
- **Book_Value_Adj_Unit_Curr** – 4 missing values. It is just 0.1% of this column contains missing entries. On further analyzing the data, the column '*Book Value (Unit Curr)*' and '*Book Value (Adj.) (Unit Curr)*' contains the same values. We can drop the '*Book Value (Adj.) (Unit Curr)*' from the dataframe. Because 2 columns with the same data do not add any values to the model building. However, using the *fillna()* method, we copied the missing values from the *'Book Value (Unit Curr)'* column(*where the data is present*).
- Ensure to check for one record, we checked for **company code=495**, the value -2.12 is imputed in missing *Book Value (Adj.) (Unit Curr) column.*



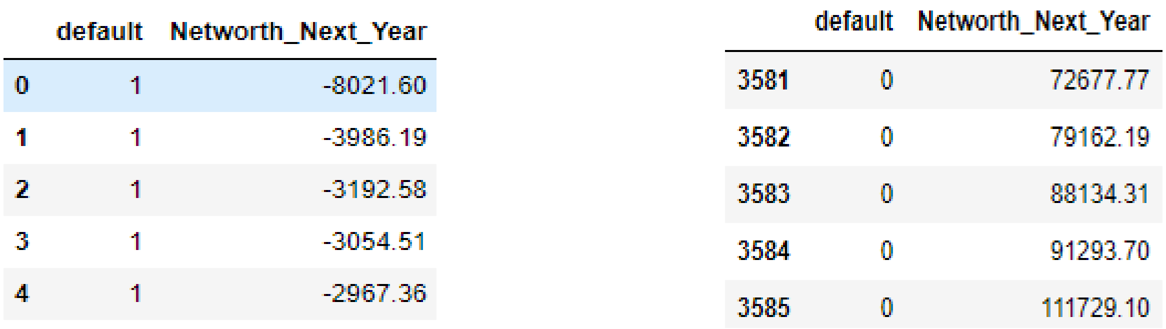
expenses_in_forex	Capital_expenses_in_forex	Book_Value_Unit_Curr	Book_Value_Adj_Unit_Curr	Market_Capitalisation	CEPS_annualised_Unit_Curr	Cash_Fl
0.0	0.0	-2.12	-2.12	4.58	-0.32	

Figure 4. After imputation of Book Value (Adj.) (Unit Curr)

- Now, there is no missing entry present in the entire dataset.

Transform Target variable into 0 and 1

Create a dependent variable named 'default' that should take the value of 1 when '*Networth_Next_Year*' column value is **negative** & **0** when '*Networth_Next_Year'* value is positive. It is achieved using the *numpy* where function.



	default	Networth_Next_Year
0	1	-8021.60
1	1	-3986.19
2	1	-3192.58
3	1	-3054.51
4	1	-2967.36

	default	Networth_Next_Year
3581	0	72677.77
3582	0	79162.19
3583	0	88134.31
3584	0	91293.70
3585	0	111729.10

Figure 5. Default variable created using Networth Next Year column

Train Test Split

We see that proportions of dependent values (*default variable*) are highly imbalanced. i.e., 0 with 89% and 1 with 11%

Using *train_test_split()* method from sklearn library, split the data into train & test dataset in a ratio of 67:33 & used random_state =42 with the *stratify= Company['default']* parameters.

This stratify parameter makes a split so that the proportion of values in the sample produced will be the same as the proportion of values provided to parameter stratify.

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

Import the RandomForestClassifier() package from sklearn library. Using the GridSearchCV method with various parameters to tune the model as shown below

```
max_depth: [3, 5, 7],min_samples_leaf: [5, 10, 15],min_samples_split: [15,
30, 45],n_estimators:[25, 50]
```

n_estimators – Number of trees want to build before taking the maximum voting.
max_features- Maximum number of features is allowed to train in individual tree.

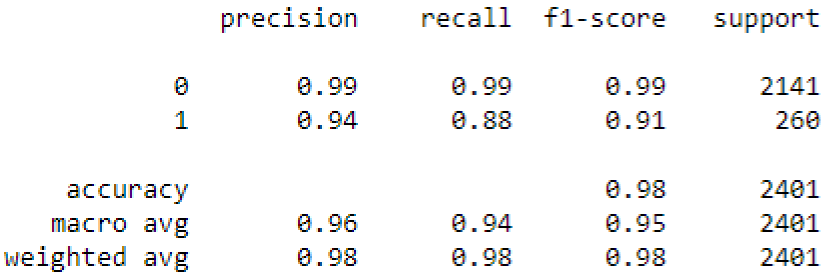
Fit the model with the training data and figure out the best parameters.

The best estimators are as follows

```
max_depth:5,min_samples_leaf:5,min_samples_split:30,n_estimators:50
```

Using the best estimators, predict the train and test record and produce the classification report.

Classification Report for Training Data



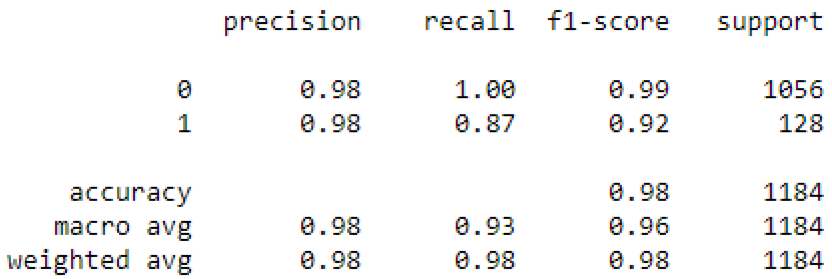
```
              precision    recall  f1-score   support

           0       0.99      0.99      0.99      2141
           1       0.94      0.88      0.91       260

    accuracy                           0.98      2401
   macro avg       0.96      0.94      0.95      2401
weighted avg       0.98      0.98      0.98      2401
```

- As observed above, for the training record, the accuracy of the model. i.e. %overall correct a prediction is 98%.

- Sensitivity of the model is 88% i.e. 88% of those defaulted were correctly identified as defaulters by the model.
- Precision of the model is 94% i.e. 94% of those defaulters were correctly identified as defaulters out of all defaulters.

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

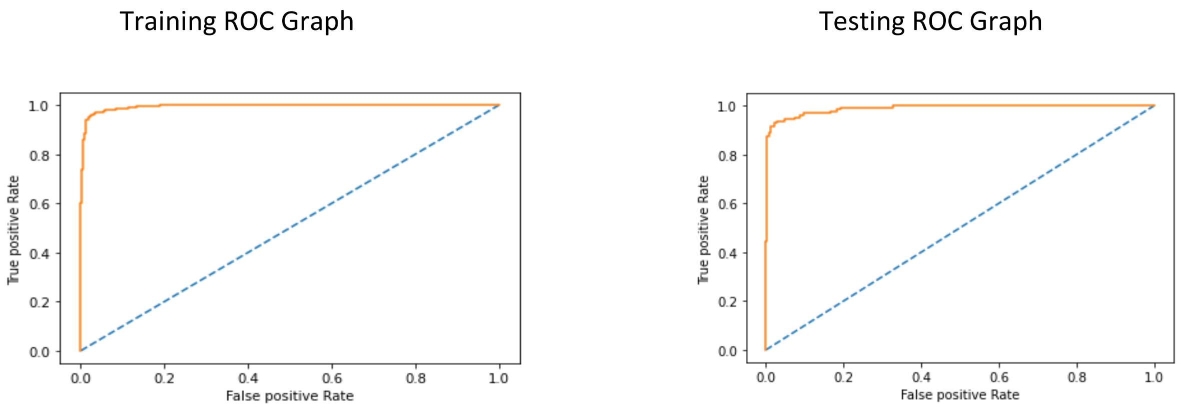
Classification Report for Testing Data



```
              precision    recall  f1-score   support

           0       0.98      1.00      0.99      1056
           1       0.98      0.87      0.92       128

    accuracy                           0.98      1184
   macro avg       0.98      0.93      0.96      1184
weighted avg       0.98      0.98      0.98      1184
```

- As observed above, for the testing record, the accuracy of the model. i.e. %overall correct predictions are 98%.
- Sensitivity of the model is 87% i.e. 87% of those defaulted were correctly identified as defaulters by the model.
- Precision of the model is 98% i.e. 98% of those defaulters were correctly identified as defaulters out of all defaulters.
- ‘Random Forest model’ is neither overfitting nor underfitting. It is a good valid model; however we will evaluate the model using the ROC graph for both training and testing records.

Performance Matrix



Training ROC Graph

Testing ROC Graph

Figure 6. ROC Curve for Random Forest Model both train & test data

- AUC score for training record is 0.996. AUC score for testing record is 0.990.

Feature Importance's

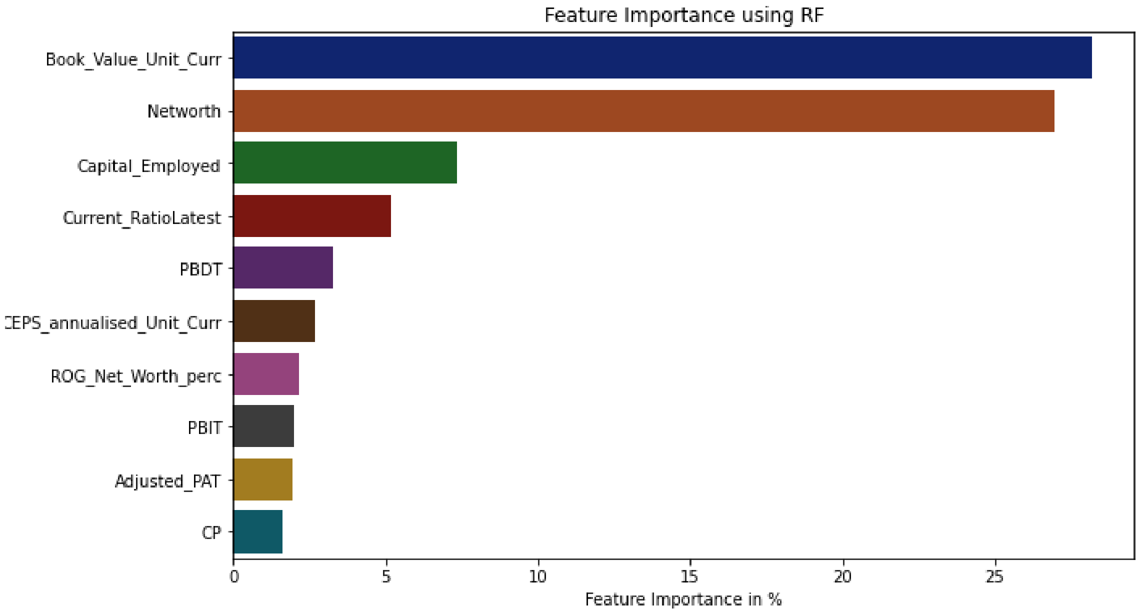


Figure 7. Important variables to predict defaulters using RF Model

In our model, the variable such as *Book Value Unit Curr(28%), Networth(27%),Capital Employed(8%)* itself constitutes 63% of the contribution to the prediction and the remaining variables such as *CEPS annualized Unit Curr, Current Ratio Latest, Net Working Capital, CP, ROG Net Worth perc, PBIDTM perc Latest, PBT* are the top 10 variables are most important features to predict the defaulters.

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

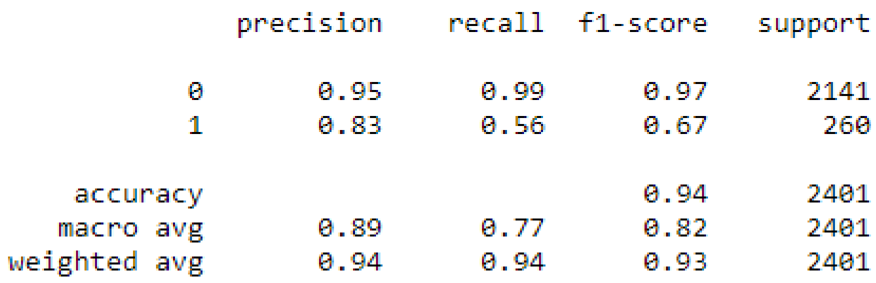
LDA is a dimensionality reduction technique that is commonly used for supervised classification problems. It is used for modeling differences in groups i.e. separating two or more classes. It works well with the small datasets.

Using the sklearn package, we import the LinearDiscriminantAnalysis() function.

There is not much scope to tune the model. LDA divided the dataset into 2 groups (0 and 1), once it is separated out, now looks at the X variables and uses the method of unsupervised learning (PCA) to try and find the structure of X's. LDA model maximizes the between class variances (like ANOVA technique) and minimizes the within class variance (like PCA). It uses the Bayes' theorem to estimate the probabilities for the every new input, the class which has the highest probability is considered as the output class

Using the best estimators, predict the train and test record and produce the classification report.

Classification Report for Training Data

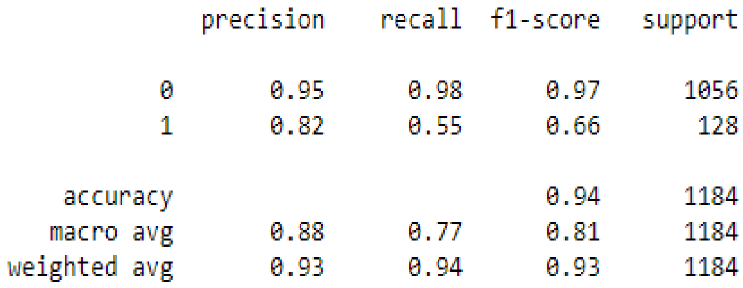


```
              precision    recall  f1-score   support

           0       0.95      0.99      0.97      2141
           1       0.83      0.56      0.67       260

    accuracy                           0.94      2401
   macro avg       0.89      0.77      0.82      2401
weighted avg       0.94      0.94      0.93      2401
```

- As observed above, for the training record, the accuracy of the model. i.e. %overall correct a prediction is 94%.
- Sensitivity of the model is 56% i.e. 56% of those defaulted were correctly identified as defaulters by the model.
- Precision of the model is 83% i.e. 83% of those defaulters were correctly identified as defaulters out of all defaulters.

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

Classification Report for Testing Data



```
              precision    recall  f1-score   support

           0       0.95      0.98      0.97      1056
           1       0.82      0.55      0.66       128

    accuracy                           0.94      1184
   macro avg       0.88      0.77      0.81      1184
weighted avg       0.93      0.94      0.93      1184
```

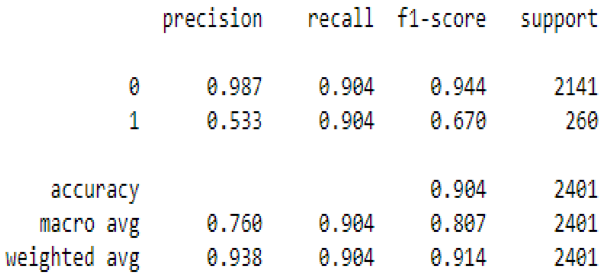
- As observed above, for the testing record, the accuracy of the model. i.e. %overall correct predictions are 94%.

- Sensitivity of the model is 55% i.e. 55% of those defaulted were correctly identified as defaulters by the model.
- Precision of the model is 82% i.e. 82% of those defaulters were correctly identified as defaulters out of all defaulters.
- As we assumed, LDA works well for the smaller datasets, this is not good valid model and the recall score is poor compared to the Random Forest Model.

Using the thresholds method with a True & False positive rate, figure out the optimum threshold to predict the defaulters. It results in 0.074.

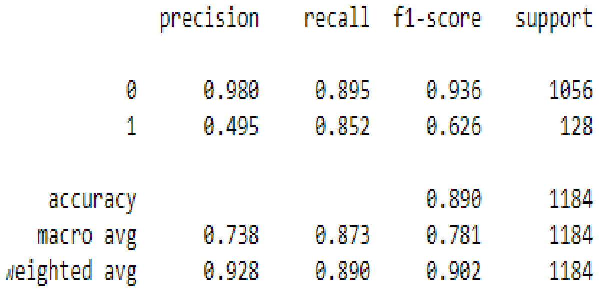
Rerun the model with 0.074 threshold value.

Training data Classification Report



```
              precision    recall  f1-score   support

           0      0.987     0.904     0.944      2141
           1      0.533     0.904     0.670       260

    accuracy                          0.904      2401
   macro avg      0.760     0.904     0.807      2401
weighted avg      0.938     0.904     0.914      2401
```

Testing data Classification Report

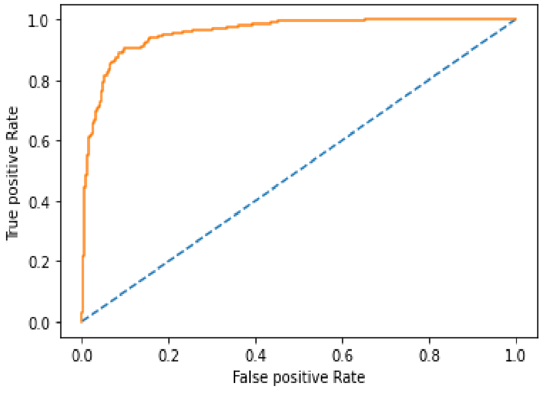


```
              precision    recall  f1-score   support

           0      0.980     0.895     0.936      1056
           1      0.495     0.852     0.626       128

    accuracy                          0.890      1184
   macro avg      0.738     0.873     0.781      1184
weighted avg      0.928     0.890     0.902      1184
```

- As observed above, the accuracy of the model. i.e. %overall correct prediction for train data is 90% & test data is 89%.
- Sensitivity of the model is 90% for train data & 85% for test data. i.e. % of those defaulted were correctly identified as defaulters by the model.
- Precision of the model is 53% for train data & 50% for test data. i.e. % of those defaulters was correctly identified as defaulters out of all defaulters.
- Sensitivity rate is increased to 30% when we set up the correct threshold to predict defaulters.

Performance Matrix for LDA

Training ROC Graph



Testing ROC Graph

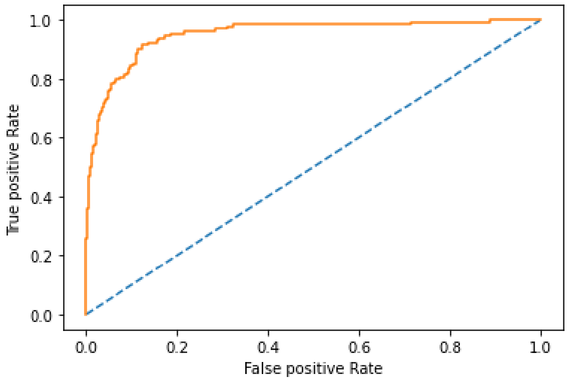


Figure 8. ROC Curve for LDA classification Model both train & test data

- AUC score for training record is 0.958.
- AUC score for testing record is 0.950.

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

	Train				Test			
	Accuracy	Recall	Precision	AUC Score	Accuracy	Recall	Precision	AUC Score
Random Forest	**98**	**88**	**94**	**99.7**	**98**	**86**	**98**	**99.1**
LDA with threshold 0.074	53.2	90.4	82	95.8	88.9	84.4	49.3	95
Logistic with threshold 0.4	96	78	80	97.7	95	74	77	97

Without doing much tuning in the Random Forest Model, we have achieved 86% of recall for the test record.

So, Random Forest is good valid model for this classification problem.

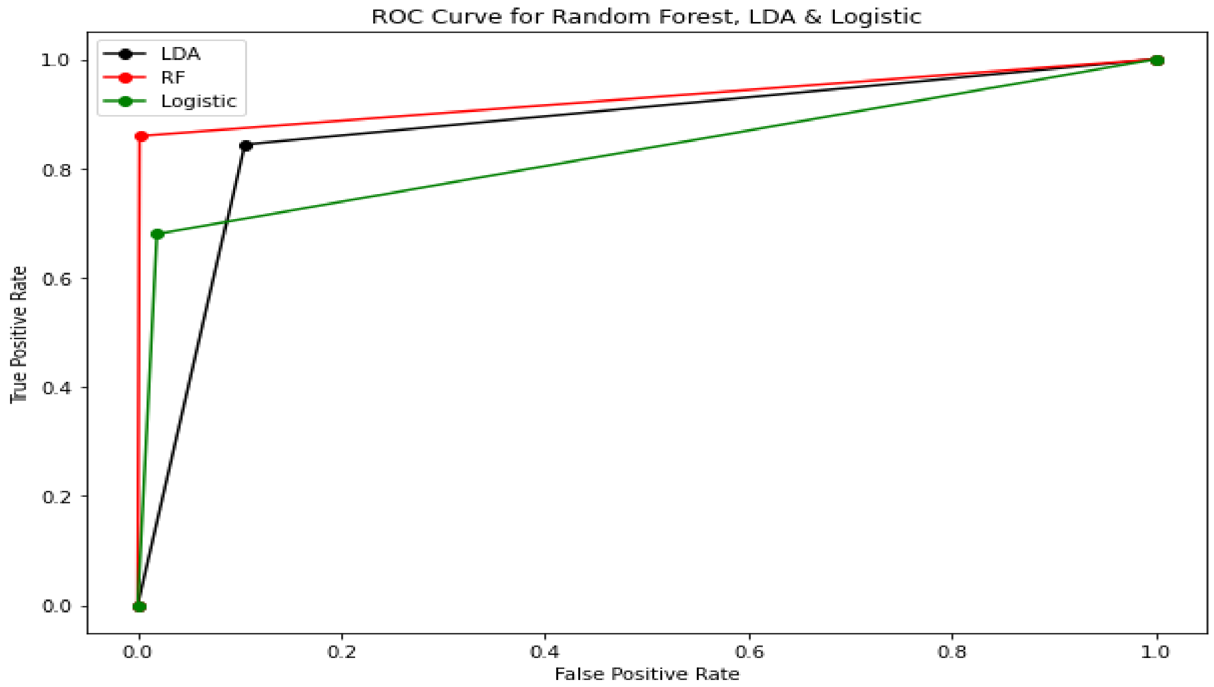


Figure 9. ROC Curve for Random Forest, LDA & Logistic classification model

- Area under the curve for LDA Classification Model is 87%.
- Area under the curve for Random Forest Classification Model is 93%.
- Area under the curve for Logistic Classification Model is 83%.

To evaluate the model, with respect to AUC score, it clearly states that Random Forest model results out with 93%.

1.13 State Recommendations from the above models

- Random Forest Model performs well with all measures. It produces the overall accuracy with 98%, precision with 98%, important measure for this problem is recall with 86% and very good AUC score with 99%. Whereas, LDA & Logistic classification model is nowhere matching to Random Forest Model for this problem.
- As an investor or Financial Advisor, we would recommend investing in a company able to handle the financial obligations, can grow quickly, and is able to manage the growth scale. Using the model, we'd be able to differentiate the companies fall under defaulters and non-defaulters.
- Invest in a company with high NetWorth, Capital used for the acquisition of profits, cash earnings per share, Liquidity ratio & profit before Interest Depreciation & tax.
- Reach out to the lender and request them to consider paying the debt on time, set up a team for continuous follow-ups with them to avoid missing the deadlines.
- At a minimum, financial institution should have a regular assessment of its liquidity position, cash flow; automate the process to send a note to the lenders.

Problem: Predicting Defaulters of a financial companies

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

Infosys & Axis Bank Stock price over the years

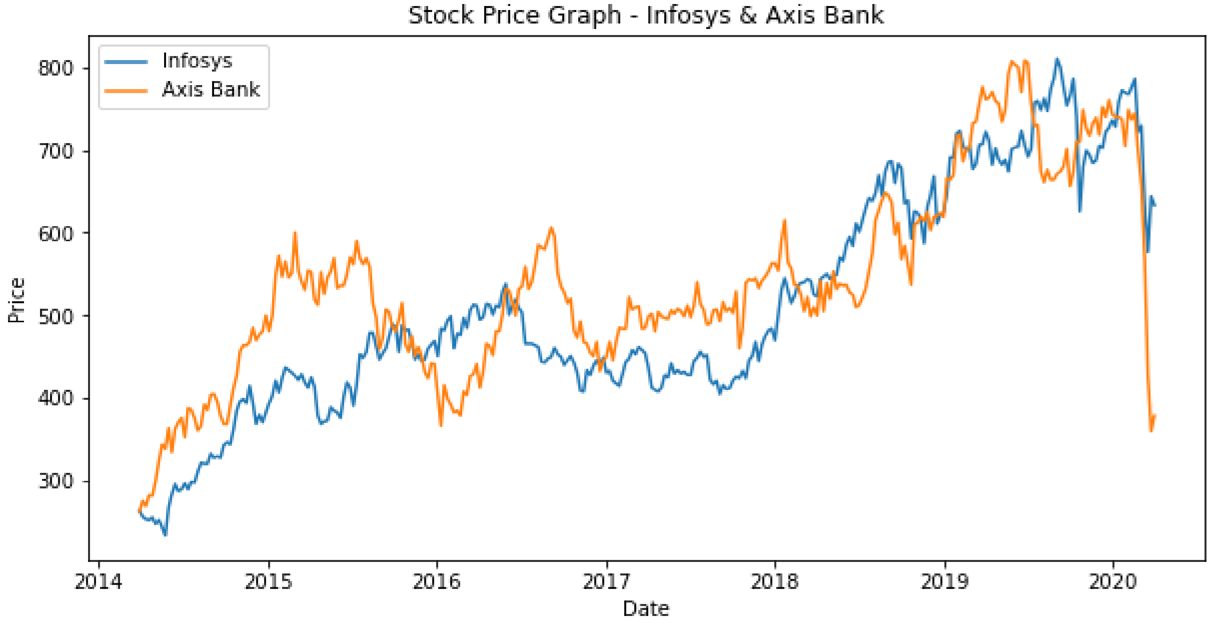
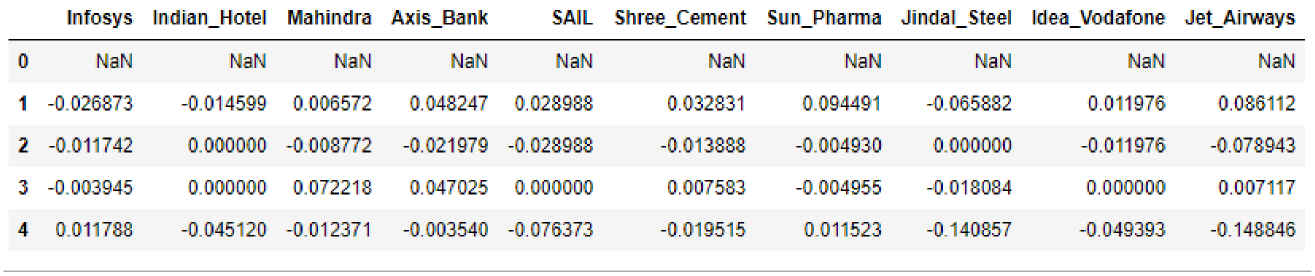


Figure 10. Stock Price graph of Infosys Vs Axis Bank

- Axis Bank stock prices were considerably high during 2014 & 2015.
- Beginning 2016 & till the Q2 2016, Infosys stock prices overcome the Axis Bank, and again from Q3 of 2016 till Q1 of 2018 Axis Bank stock prices were high.
- From Q2 of 2018 till the end of 2019, Infosys stock prices were high marginally high compared to Axis Bank stock price.
- Stock prices of both are almost same during Q1 of 2019.
- During Q2 of 2019, Axis Bank stock prices were high, where as in Q3, Infosys stock prices were high.
- Since Q4 2019 till the data we have (2020), prices of both were coming down.
- Overall, Axis Bank stock prices were high most of the times from 2014 to 2020.

2.2 Calculate Returns for all stocks with inference

We have the data on week to week basis. Let's take the logarithmic difference for the stocks.



	Infosys	Indian_Hotel	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

We now have the logarithmic returns available for the all the 10 stocks.

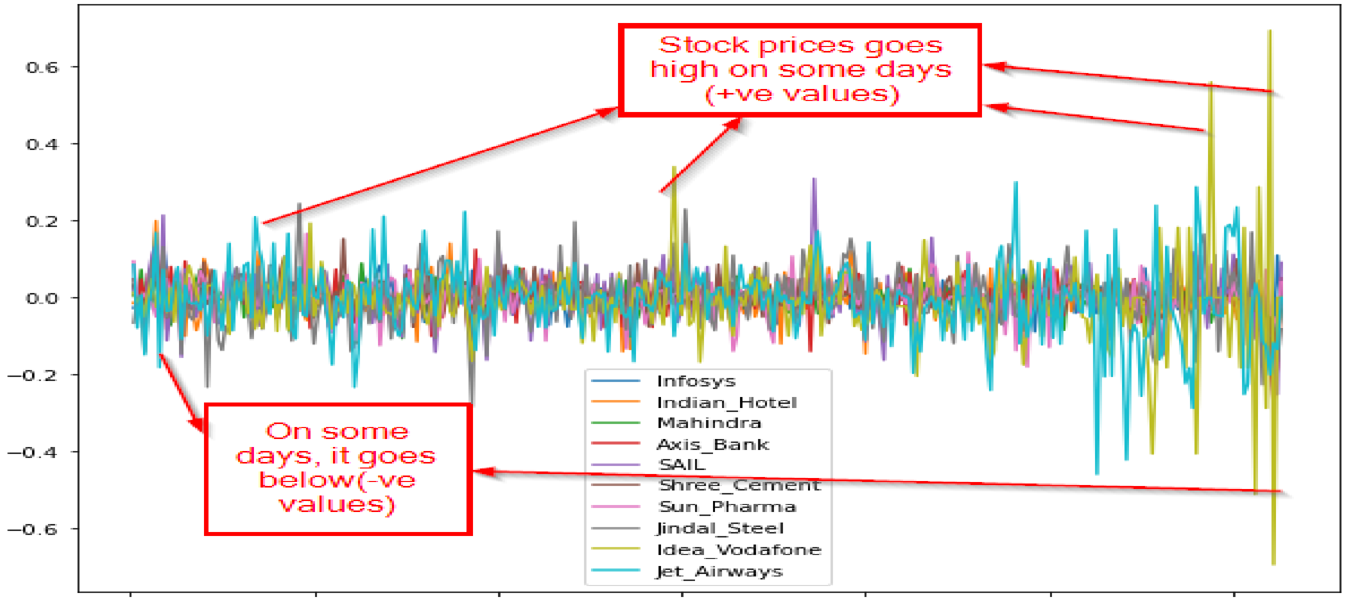


Figure 11. Stock Means & Standard Deviation for all stocks

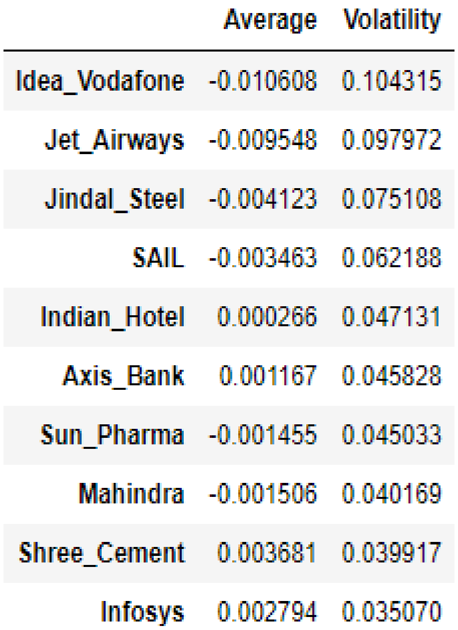
Analyzing the returns for all the stocks is just random and not predictable. Anywhere in the world, returns of the stocks are unpredictable.

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

We now look at Means & Standard Deviations of these returns

Stock Means: Average returns that the stock is making on a week to week basis.

Stock Standard Deviation: It is a measure of volatility meaning the more a stock's returns vary from the stock's average return, the more volatile the stock.



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

Figure 12. Stock Means & Standard Deviation for all stocks

- Here, the data’s are displayed from high to low volatility.
- Stocks such as *Idea-Vodafone, Jet Airways, Jindal, SAIL* yiedls very low return & their volatility is high.
- *Sun Pharma & Mahindra* stocks volatility is relatively low; however their average weekly return is in negative.
- *Indian hotel, Axis Bank, Shree Cement & Infosys* stocks results out with positive weekly average returns & low volatility.

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

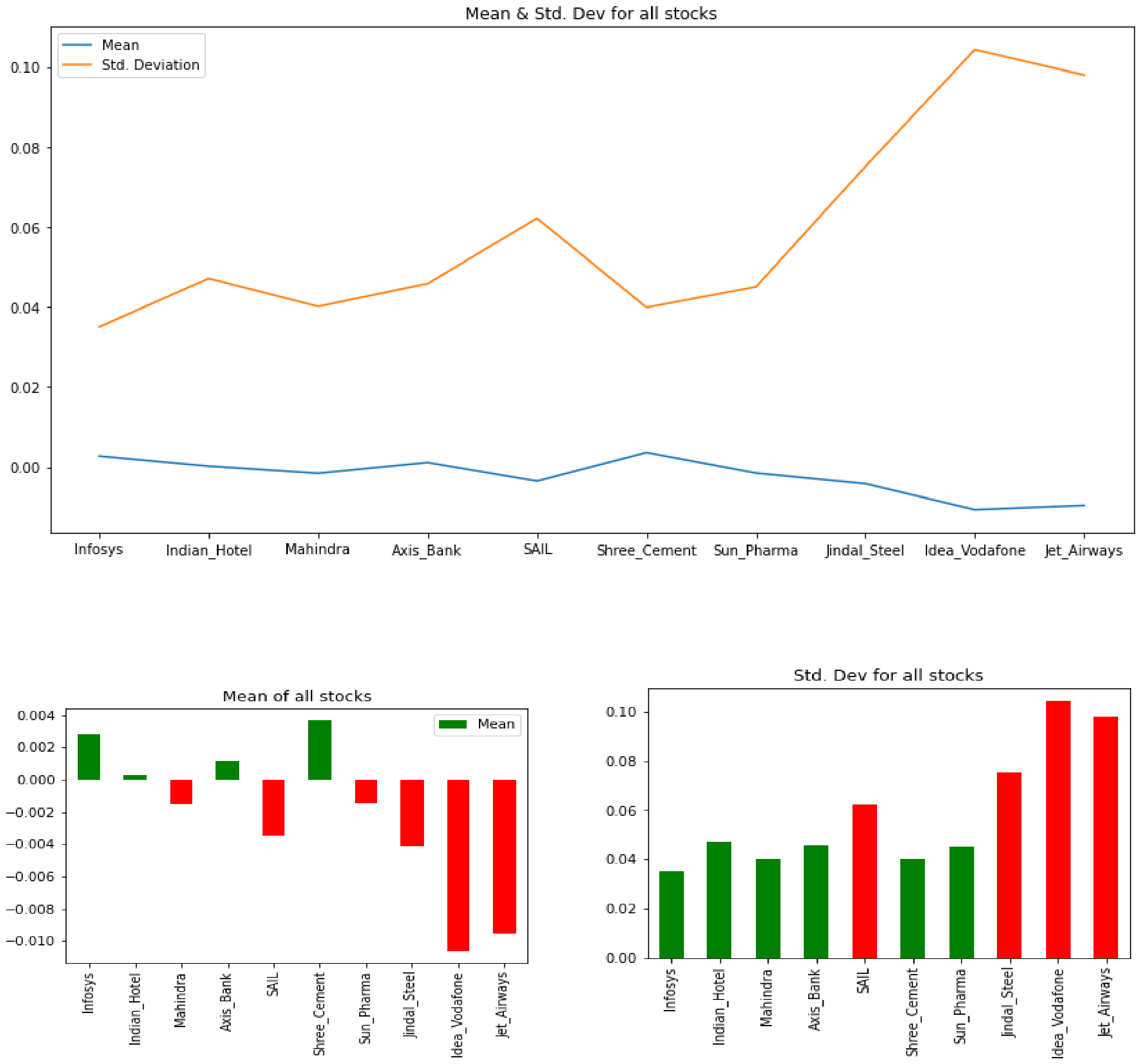


Figure 13. Graph of Stock Means & Standard Deviation for all stocks

- The standard deviation for the stocks such as *Idea-Vodafone, Jet Airways* is super high compared to the other stocks. Whereas, their average weekly return mean is super low compared to the other stocks.
- *SAIL* stock volatility is also marginally high & its return is also moderate.

- The average weekly returns of the stocks such as *Infosys & Shree Cement* are marginally high & their volatility is low compared to the other stocks.
- Stocks such as *Indian Hotel, Mahindra, Axis Bank, SAIL & Sun pharma* average weekly return is indistinguishable.
- Mahindra & Sun pharma stocks volatility is low & their returns are in negative values. There is a moderate risk investing money in these stocks.

2.5 Conclusion and Recommendations.

Stock with a lower mean & higher standard deviation does not play a role in a portfolio that has competing stock with more returns & less risk. Thus for the data we have here, we are only left few stocks:

Ones with higher return for a comparative or lower risk are considered better.

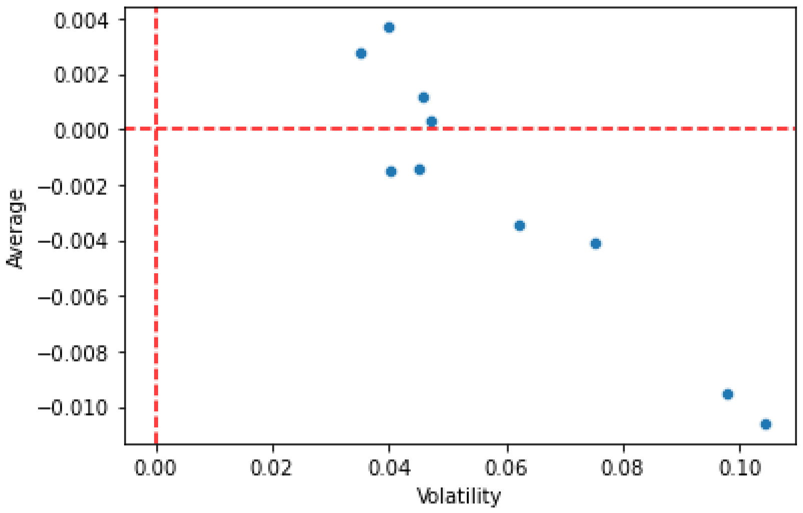
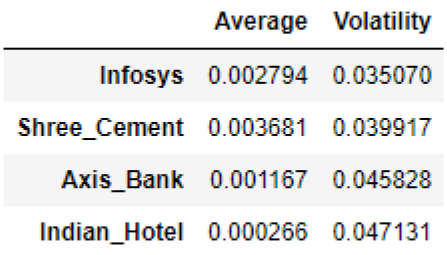


Figure14. Plot to predict good returns stocks with low risk

Let’s identify the 4 stocks with weekly average returns is above 0 & with the low volatility.



	Average	Volatility
Infosys	0.002794	0.035070
Shree_Cement	0.003681	0.039917
Axis_Bank	0.001167	0.045828
Indian_Hotel	0.000266	0.047131

Figure15. Stocks with good returns & minimal volatility

- It is advisable to invest in the Stocks such as *Infosys, Shree Cement, Axis Bank* and *Indian Hotel*. These stocks results out with minimal risk and positive return. We would not be losing the money if we invest in these stocks.
- There is a minimal risk investing money in the stocks of *Mahindra* & *Sun Pharma*.
- It is not likely to invest the money in the stocks such as *Idea-Vodafone, Jet Airway, Jindal Steel & SAIL* as their market volatility is quite high & returns are in negative.

THE END!