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

Random forest builds multiple decision trees and merges them together to get a more accurate and stable prediction.

We got the following best parameters through GridSearchCV for the dataset,

RandomForestClassifier(max_depth=**8**, max_features=**4**, min_samples_leaf=**30**, min_samples_split=**90**, n_estimators=**200**)

```
RandomForestClassifier(max_depth=8, max_features=4, min_samples_leaf=30,
                       min_samples_split=90, n_estimators=200)
```

The highest importance feature denoted by this method is **Networth (49.21% importance).**

Train_Test Split :

```
X_train.head()
```

	Networth	TotalDebt	GrossBlock	CurrentLiabilitiesandProvisions	PBIDT	PBIT	BookValueAdjUnitCurr	CurrentRatioLatest	FixedAssetsRatioLatest
662	-0.702412	-0.683014	-0.698523	-0.722541	-0.625732	-0.605635	-0.623993	-0.372287	-0.765576
1373	-0.600008	-0.683014	-0.698523	-0.713137	-0.614916	-0.587930	0.503399	0.352270	-0.696731
3268	0.263459	-0.530709	-0.650638	-0.354174	-0.892836	-0.702028	0.314003	-0.655853	0.697387
3246	1.878272	0.569079	1.842891	2.168455	2.432545	1.898613	2.112590	-0.181561	0.383280
1456	-0.621330	-0.498962	-0.610615	-0.699165	-0.475018	-0.435472	-0.590156	0.393645	1.136276

```
X_test.head()
```

	Networth	TotalDebt	GrossBlock	CurrentLiabilitiesandProvisions	PBIDT	PBIT	BookValueAdjUnitCurr	CurrentRatioLatest	FixedAssetsRatioLatest
3163	1.654536	0.465345	0.599983	0.924928	1.686256	1.423928	0.044357	-0.322839	2.267921
3133	1.893879	0.834174	0.197061	3.225247	1.196182	0.880685	0.355176	-0.686127	0.056265
937	-0.677609	-0.683014	-0.711581	-0.721735	-0.685585	-0.684323	-0.866240	-0.062483	-0.868844
196	-0.889378	-0.276866	-0.364257	-0.400388	-0.581023	-0.642028	-1.692695	-1.160419	-0.541829
2852	2.074376	0.752703	3.108037	0.279117	0.575656	-1.743661	-0.667112	0.141362	-0.176088

```
train_labels
```

```
662     0
1373    0
3268    0
3246    0
1456    0
       ..
1130    0
1294    0
860     0
3507    0
3174    0
Name: default, Length: 2402, dtype: int64
```

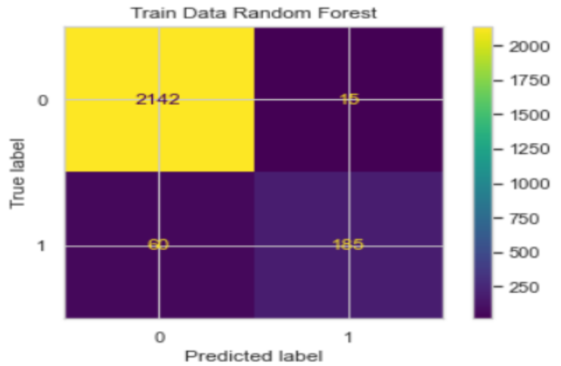
```
test_labels
```

```
3163    0
3133    0
937     0
196     1
2852    0
       ..
2953    0
3116    0
1010    0
1292    0
2130    0
Name: default, Length: 1184, dtype: int64
```

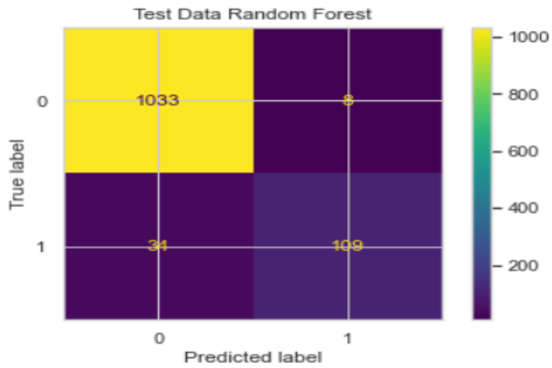
1.9)Validate the Random Forest Model on test Dataset and state the performance matrices. Also state interpretation from the model

<u>Confusion Matrix:</u>

```
Text(0.5, 1.0, 'Train Data Random Forest')
```




```
Text(0.5, 1.0, 'Test Data Random Forest')
```

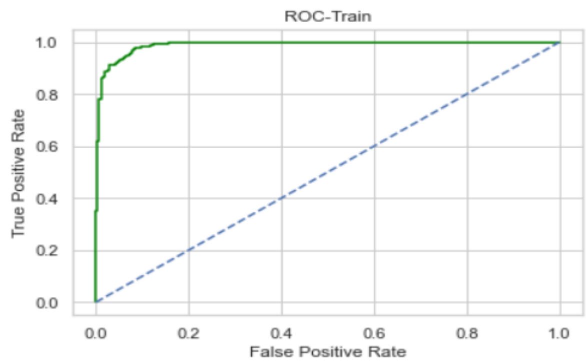


<u>Classification Report:</u>

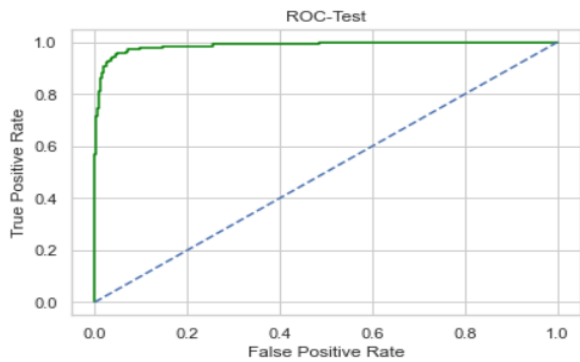
```
Train Data Random Forest
              precision    recall  f1-score   support

           0       0.97      0.99      0.98      2157
           1       0.93      0.76      0.83       245

    accuracy                           0.97      2402
   macro avg       0.95      0.87      0.91      2402
weighted avg       0.97      0.97      0.97      2402

TestData Random Forest
              precision    recall  f1-score   support

           0       0.97      0.99      0.98      1041
           1       0.93      0.76      0.84       143

    accuracy                           0.96      1184
   macro avg       0.95      0.88      0.91      1184
weighted avg       0.96      0.96      0.96      1184
```

<u>ROC Curve:</u>

```
Area under Curve is 0.8740739689478016
```




```
Area under Curve is 0.8772764219450098
```



Training and Test set results are almost similar, and with the overall measures high, the model is a good model.

Networth is again the most important variable for predicting default status.

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

Linear discriminate analysis and logistic regression are the most widely used statistical methods for analyzing categorical outcome variable. While both are appropriate for the development of linear classification models, linear discriminate analysis makes more assumptions about the underlying data and LDA is preferred when it is nominal (more than two groups).

```
clf = LinearDiscriminantAnalysis()
model1=clf.fit(X_train1,Y_train1)
```

The Coefficients of different variables as per LDA model are as below:

```
The coefficient for Networth is -1.200194123134492
The coefficient for TotalDebt is 0.2938148767264865
The coefficient for GrossBlock is 0.6036888357420934
The coefficient for CurrentLiabilitiesandProvisions is 0.9108548618214206
The coefficient for PBIDT is -1.2121824268143804
The coefficient for PBIT is 0.4690372590191507
The coefficient for BookValueAdjUnitCurr is -2.004421364264967
The coefficient for CurrentRatioLatest is -1.0395436957623165
The coefficient for FixedAssetsRatioLatest is -0.4637332080302533
The coefficient for InterestCoverRatioLatest is -0.565646114426516
```

The highest importance feature denoted by this method is **BookValueAdjUnitCurr (-2.0044213).**

Train_Test Split :

```
X_train1.head()
```

	Networth	TotalDebt	GrossBlock	CurrentLiabilitiesandProvisions	PBIDT	PBIT	BookValueAdjUnitCurr	CurrentRatioLatest	FixedAssetsRatioLatest
662	-0.702412	-0.683014	-0.698523	-0.722541	-0.625732	-0.605635	-0.623993	-0.372287	-0.765576
1373	-0.600008	-0.683014	-0.698523	-0.713137	-0.614916	-0.587930	0.503399	0.352270	-0.696731
3268	0.263459	-0.530709	-0.650638	-0.354174	-0.892836	-0.702028	0.314003	-0.655853	0.697387
3246	1.878272	0.569079	1.842891	2.168455	2.432545	1.898613	2.112590	-0.181561	0.383280
1456	-0.621330	-0.498962	-0.610615	-0.699165	-0.475018	-0.435472	-0.590156	0.393645	1.136276

```
X_test1.head()
```

	Networth	TotalDebt	GrossBlock	CurrentLiabilitiesandProvisions	PBIDT	PBIT	BookValueAdjUnitCurr	CurrentRatioLatest	FixedAssetsRatioLatest
3163	1.654536	0.465345	0.599983	0.924928	1.686256	1.423928	0.044357	-0.322839	2.267921
3133	1.893879	0.834174	0.197061	3.225247	1.196182	0.880685	0.355176	-0.686127	0.056265
937	-0.677609	-0.683014	-0.711581	-0.721735	-0.685585	-0.684323	-0.866240	-0.062483	-0.868844
196	-0.889378	-0.276866	-0.364257	-0.400388	-0.581023	-0.642028	-1.692695	-1.160419	-0.541829
2852	2.074376	0.752703	3.108037	0.279117	0.575656	-1.743661	-0.667112	0.141362	-0.176088

```
Y_train1
```

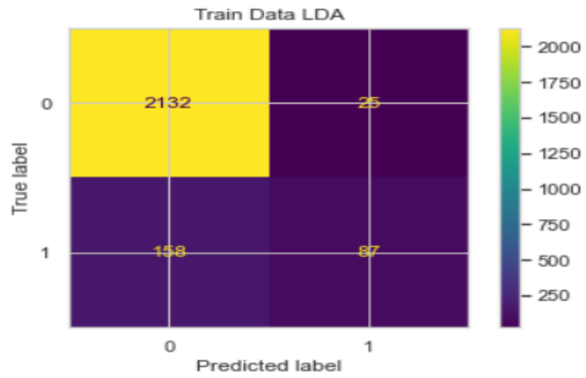
```
662     0
1373    0
3268    0
3246    0
1456    0
       ..
1130    0
1294    0
860     0
3507    0
3174    0
Name: default, Length: 2402, dtype: int64
```

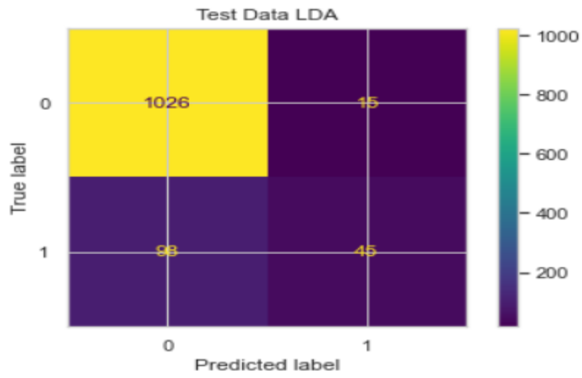
```
Y_test1
```

```
3163    0
3133    0
937     0
196     1
2852    0
       ..
2953    0
3116    0
1010    0
1292    0
2130    0
Name: default, Length: 1184, dtype: int64
```

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

<u>Confusion Matrix:</u>



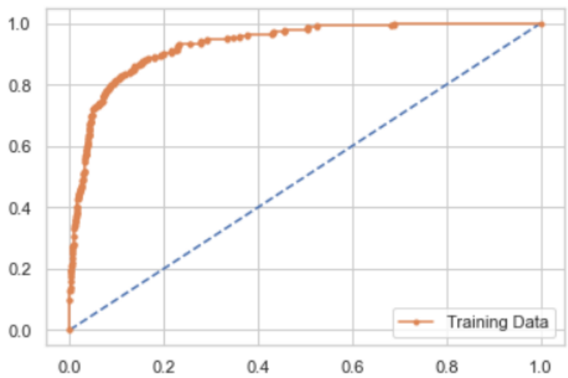


<u>Classification Report:</u>

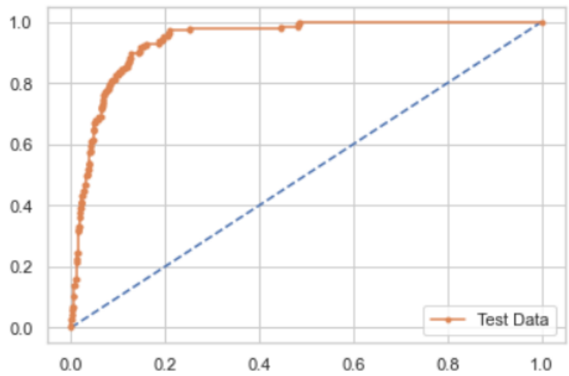
```
Classification Report of the training data:
              precision    recall  f1-score   support

           0       0.93      0.99      0.96      2157
           1       0.78      0.36      0.49       245

    accuracy                           0.92      2402
   macro avg       0.85      0.67      0.72      2402
weighted avg       0.92      0.92      0.91      2402

Classification Report of the test data:
              precision    recall  f1-score   support

           0       0.91      0.99      0.95      1041
           1       0.75      0.31      0.44       143

    accuracy                           0.90      1184
   macro avg       0.83      0.65      0.70      1184
weighted avg       0.89      0.90      0.89      1184
```

<u>ROC Curve:</u>

AUC for the Training Data: 0.932



AUC for the Test Data: 0.941



Training and Test set results are almost similar, and with the overall measures high, the model is a good model.

BookValueAdjUnitCurr is again the most important variable for predicting default status.

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

Comparing the performance metrics from the three models, we can summarize as below,

	Logistic Train	Logistic Test	LDA Train	LDA Test	Random Forest Train	Random Forest Test
Accuracy	0.94	0.93	0.92	0.9	0.97	0.96
AUC	0.934	0.946	0.932	0.941	0.87	0.87
Recall	0.49	0.58	0.36	0.31	0.76	0.76
Precision	0.83	0.83	0.78	0.75	0.93	0.93
F1 Score	0.62	0.68	0.49	0.44	0.83	0.84

Looking at the details got from **test data** from the three models ,

Accuracy : Random Forest models has highest value of 0.96

AUC : Logistic Regression model has highest value of 0.946 and Random Forest model has least value of 0.87

Recall : Random Forest model has highest value of 0.76 and LDA model has least value 0.31

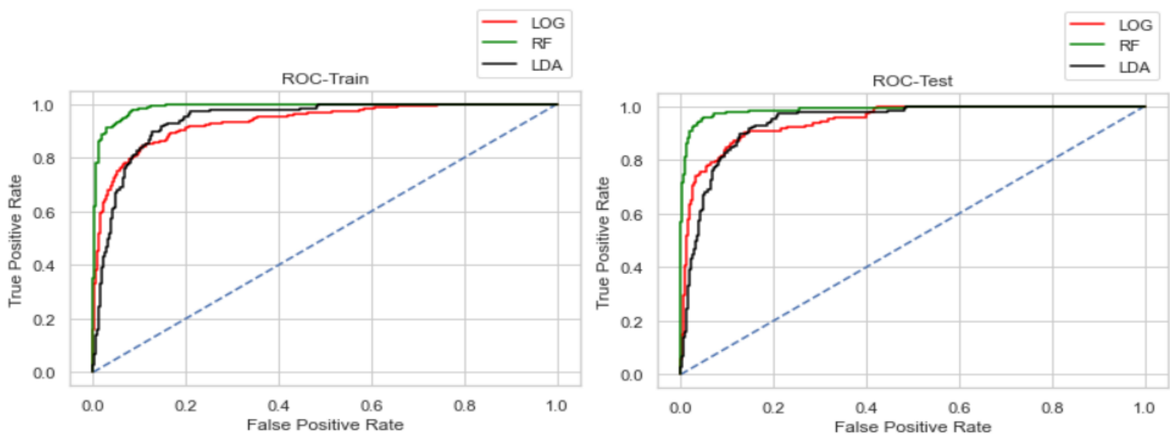
Precision : Random Forest has highest value of 0.93 and LDA model has least value of 0.75

F1 Score : Random Forest has highest value 0.84 and LDA model has least value of 0.44

Training and Test set results are almost similar in all the three models and overall measures are high in Random Forest.

Therefore, **Random Forest has slightly better performance than the Logistic Regression and LDA model**

Overall all the 3 models are reasonably stable enough to be used for making any future predictions. From Logistic and LDA Model, the variable **BookValueAdjUnitCurr** is found to be the most useful feature amongst all other features for predicting default status.



1.13) State Recommendations from the above models.

Due to the importance of understanding and managing the risks in volatile business domains, it is required to find an effective aid in making decisions. The results from model shows that the above algorithm is a promising opportunity in predicting whether a company will go for the default or not through the cause and effect relationship between the independent and dependent variables of the given dataset.

Also Random forest has proven to be a great algorithm if the dataset is in tabular format. Random Forests requires less pre-processing and the training process is also much simpler. Moreover hyper-parameter tuning is easier with random forest when compared to other models. This gives random forest the edge above remaining models

The above model will be helpful in predicting the dependent variables through the independent variables by assigning the probability of company going for the default to the every predictor variable to give the best predictive/dependent variable.

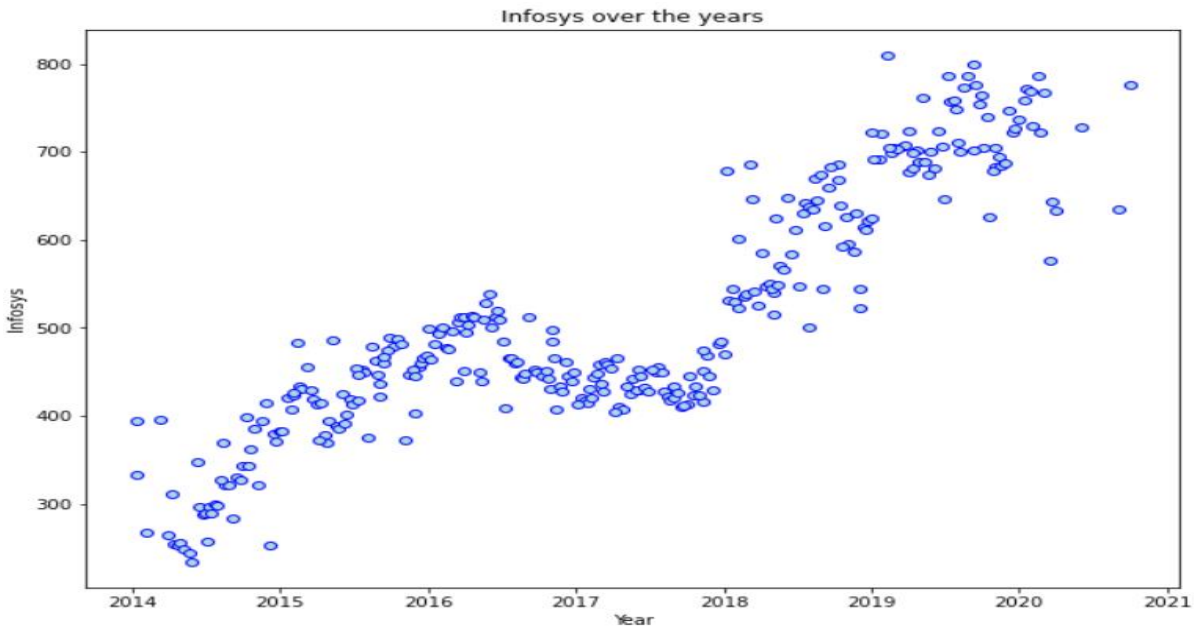
As per predictions of the model, The variable **BookValueAdjUnitCurr** is found to be the most useful feature amongst all other features for predicting default status.

We must look about company based on the feature importance to get the better results in predicting whether an company will go for the default or not.

So, The Overall analysis of given dataset definitely helped to get insights that would help in predicting about the company.

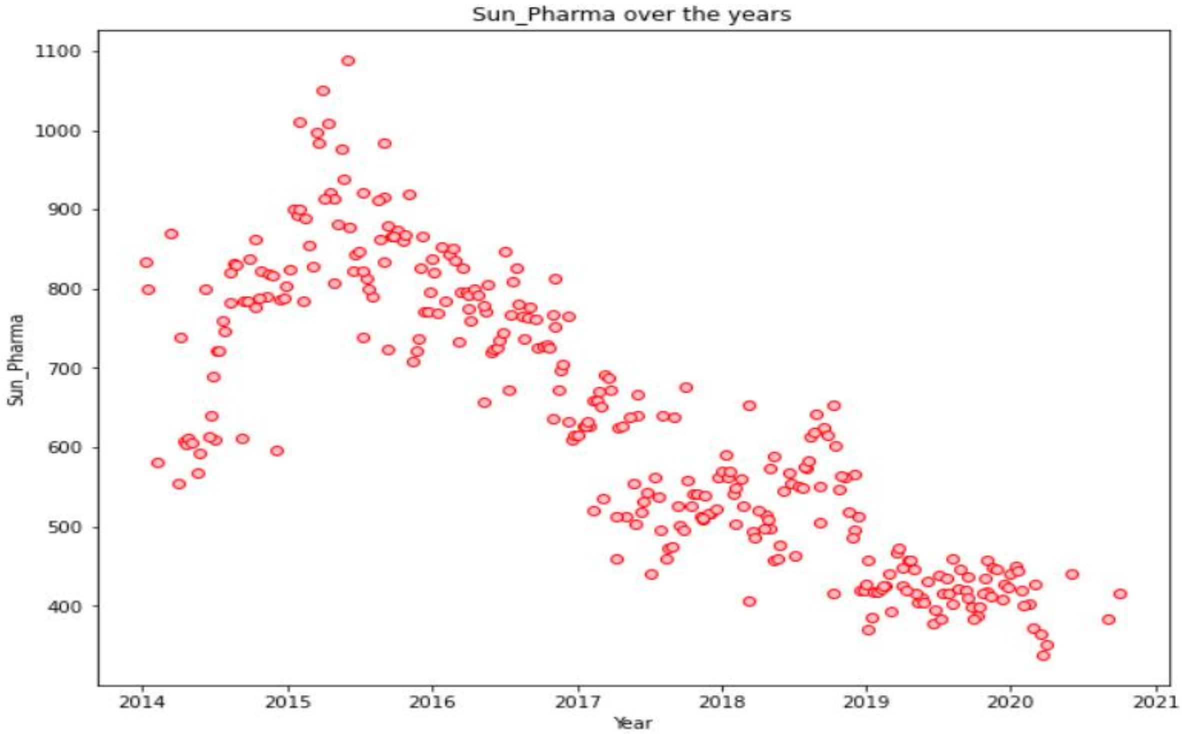
2.1) Draw Stock Price Graph(Stock Price vs Time) for any 2 given stocks with inference

Stock Price Graph for Infosys:



The Stock price for the Infosys is on increasing trend from 2014 to 2021. There is an almost increase of 500 points within the span of 7 years.

Stock Price Graph for Sun_Pharma:



The Stock price for the Sun_Pharma is on decreasing trend from 2014 to 2021. There is an almost decrease of 700 points within the span of 7 years.

2.2) Calculate Returns for all stocks with inference.

Returns is the **difference** between two consecutive week prices for the stock.

```
stock_returns.head()
```

	Infosys	Indian_Hotel	Mahindra_&_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

The **negative** value of Return means there is **decrease** in price compared to previous week and the **positive** value of Return means there is **increase** in price compared to previous week.

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

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

```
stock_means

Infosys                0.002794
Indian_Hotel           0.000266
Mahindra_&_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
```

Shree_Cement has **highest** Stock Means and **Jet_Airways** has **lowest** Stock Means.

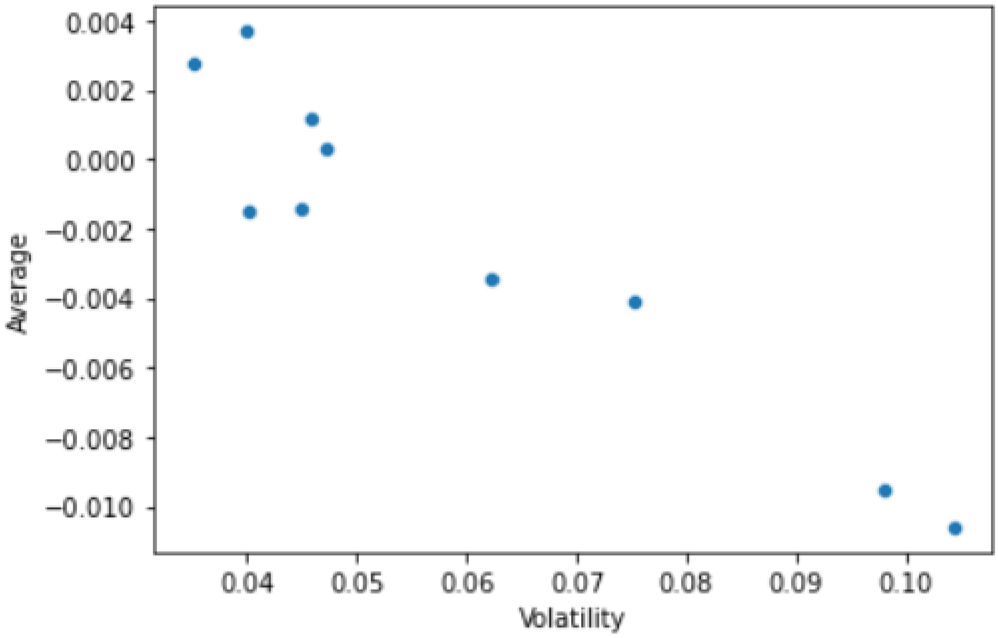
- **Stock Standard Deviation** : It is a measure of volatility meaning the more a stock's returns vary from the stock's average return, the more volatile the stock.

```
stock_sd

Infosys                0.035070
Indian_Hotel           0.047131
Mahindra_&_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
```

Idea_Vodafone has **highest** Volatility and **Infosys** has **lowest** Volatility.

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

<u>Plot between Stock Means & Stock standard Deviation:</u>



From above plot, we can understand that stock with higher average value has lower volatility. There is a decrease in the average value with the increase in the volatility.

	Average	Volatility
Infosys	0.002794	0.035070
Indian_Hotel	0.000266	0.047131
Mahindra_&_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

2.5) Conclusion and Recommendations.

Of all the above stocks, only the following stocks are having positive average means.

Infosys – 0.002794
Indian_Hotel – 0.000266
Axis_Bank – 0.001167
Shree_Cement – 0.003681

Stock with a lower mean & higher standard deviation do 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:

	Average	Volatility
Infosys	0.0028	0.0351
Shree_Cement	0.0037	0.0399
Axis_Bank	0.0012	0.0458
Indian_Hotel	0.0003	0.0471

Among the above stocks, **Infosys & Shree_Cement** stocks are having **best average** with **low volatility** .

Therefore , the stocks with higher return for a comparative or lower risk are considered better among all the available stocks.