11/20/2022

FRA Project(Milestone-2)

Shailesh Pande
PUNE

Contents

1 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.

Explanation of data fields available in Data Dictionary, 'Credit Default Data Dictionary.xlsx'

- Read the data

Table 1: Data Provided

```
(3586, 67)
```

	Co_Code	Co_Name	Networth Next Year	Equity Paid Up	Networth	Capital Employed	Total Debt	Gross Block	Net Working Capital	Current Assets	...	PBIDTM (%) [Latest]	PBITM (%) [Latest]	PBDTM (%) [Latest]	CPM (%) [Latest]	APATM (%) [Latest]	Debt Velo (Da
0	16974	Hind.Cables	-8021.60	419.36	-7027.48	-1007.24	5936.03	474.30	-1076.34	40.50	...	0.00	0.00	0.00	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	...	-10.30	-39.74	-57.74	-57.74	-87.18	
2	14852	ABG Shipyard	-3192.58	53.84	506.86	7714.68	6944.54	1281.54	4496.25	9097.64	...	-5279.14	-5516.98	-7780.25	-7723.67	-7961.51	
3	2439	GTL	-3054.51	157.30	-623.49	2353.88	2326.05	1033.69	-2612.42	1034.12	...	-3.33	-7.21	-48.13	-47.70	-51.58	
4	23505	Bharati Defence	-2967.36	50.30	-1070.83	4675.33	5740.90	1084.20	1836.23	4685.81	...	-295.55	-400.55	-845.88	379.79	274.79	3

- We have already worked on cleaning up the column names
- Missing values were ascertained to be 118 numbers in the full data set
- The target feature, namely 'NETWORTH_NEXT_YEAR' was converted into a binary feature called 'default', which will take on the value of 0 , it the NETWORTH_NEXT_YEAR is greater than zero. 'default' will be 1 , if NETWORTH_NEXT_YEAR is negative.

Table 2: New Binary feature 'default'

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

- The above dependent features along with the categorical features –'CO_CODE' and 'CO_NAME' are dropped from the data set.

- The statistical summary of features is

Table 3:Statistical Summary

	count	mean	std	min	25%	50%	75%	max
NETWORTH_NEXT_YEAR	3586.0	725.05	4769.68	-8021.60	3.98	19.02	123.80	111729.10
EQUITY_PAID_UP	3586.0	62.97	778.76	0.00	3.75	8.29	19.52	42263.46
NETWORTH	3586.0	649.75	4091.99	-7027.48	3.89	18.58	117.30	81657.35
CAPITAL_EMPLOYED	3586.0	2799.61	26975.14	-1824.75	7.60	39.09	226.60	714001.25
TOTAL_DEBT	3586.0	1994.82	23652.84	-0.72	0.03	7.49	72.35	652823.81
GROSS_BLOCK	3586.0	594.18	4871.55	-41.19	0.57	15.87	131.90	128477.59
NET_WORKING_CAPITAL	3586.0	410.81	6301.22	-13162.42	0.94	10.14	61.18	223257.56
CURRENT_ASSETS	3586.0	1960.35	22577.57	-0.91	4.00	24.54	135.28	721166.00
CURRENT_LIABILITIES_AND_PROVISIONS	3586.0	391.99	2675.00	-0.23	0.73	9.23	65.65	83232.98
TOTAL_ASSETS_BY_LIABILITIES	3586.0	1778.45	11437.57	-4.51	10.56	52.01	310.54	254737.22
GROSS_SALES	3586.0	1123.74	10603.70	-62.59	1.44	31.21	242.25	474182.94
NET_SALES	3586.0	1079.70	9996.57	-62.59	1.44	30.44	234.44	443775.16
OTHER_INCOME	3586.0	48.73	426.04	-448.72	0.02	0.45	3.64	14143.40
VALUE_OF_OUTPUT	3586.0	1077.19	9843.88	-119.10	1.41	30.90	235.84	435559.09
COST_OF_PRODUCTION	3586.0	798.54	9076.70	-22.65	0.94	25.99	189.55	419913.50
SELLING_COST	3586.0	25.55	194.24	0.00	0.00	0.16	3.88	5283.91
PBIDT	3586.0	248.18	1949.59	-4655.14	0.04	2.04	23.52	42059.26
PBDT	3586.0	116.27	956.20	-5874.53	0.00	0.80	12.94	23215.00
PBIT	3586.0	217.66	1850.97	-4812.95	0.00	1.15	16.67	41402.96
PBT	3586.0	85.75	799.93	-6032.34	-0.06	0.31	7.42	16798.00
PAT	3586.0	61.22	620.30	-6032.34	-0.06	0.26	5.54	13383.39
ADJUSTED_PAT	3586.0	60.06	580.43	-4418.72	-0.09	0.21	5.34	13384.11
CP	3586.0	91.73	780.79	-5874.53	0.00	0.74	10.91	20760.20
REVENUE_EARNINGS_IN_FOREX	3586.0	131.17	1150.73	0.00	0.00	0.00	7.20	46158.00
REVENUE_EXPENSES_IN_FOREX	3586.0	256.33	4132.34	0.00	0.00	0.00	6.99	193979.73
CAPITAL_EXPENSES_IN_FOREX	3586.0	7.66	111.43	0.00	0.00	0.00	0.00	3722.10
BOOK_VALUE_UNIT_CURR	3586.0	157.24	1622.66	-3371.57	7.96	21.66	71.67	75790.00
BOOK_VALUE_ADJ_UNIT_CURR	3582.0	2243.15	128283.73	-33715.70	7.06	18.92	60.01	7677600.29
MARKET_CAPITALISATION	3586.0	1664.09	12805.17	0.00	0.00	8.37	111.46	260865.08
CEPS_ANNUALISED_UNIT_CURR	3586.0	36.02	828.42	-1808.00	0.00	1.14	8.77	45438.44
CASH_FLOW_FROM_OPERATING_ACTIVITIES	3586.0	65.77	1455.05	-25469.23	-0.31	0.45	12.65	44529.40
CASH_FLOW_FROM_INVESTING_ACTIVITIES	3586.0	-60.87	701.97	-23843.45	-5.12	-0.12	0.12	3732.98
CASH_FLOW_FROM_FINANCING_ACTIVITIES	3586.0	11.44	1272.26	-38374.04	-5.85	0.00	0.46	28846.00
ROG_NET_WORTH_PERC	3586.0	1237.62	41041.93	-14485.71	-1.49	1.84	11.36	2144020.00
ROG_CAPITAL_EMPLOYED_PERC	3586.0	2988.88	126472.87	-8614.63	-3.84	1.38	12.59	7412700.00
ROG_GROSS_BLOCK_PERC	3586.0	37.55	893.62	-116.12	0.00	0.25	6.72	47400.00
ROG_GROSS_SALES_PERC	3586.0	242.67	6103.53	-5503.70	-8.08	3.31	21.52	320200.00
ROG_NET_SALES_PERC	3586.0	242.59	6103.49	-5503.70	-8.12	3.20	21.57	320200.00
ROG_COST_OF_PRODUCTION_PERC	3586.0	310.49	5573.22	-2130.23	-7.24	4.42	23.12	267150.00
ROG_TOTAL_ASSETS_PERC	3586.0	2793.28	125941.65	-136.13	-3.97	1.48	12.50	7422120.00
ROG_PBIDT_PERC	3586.0	375.85	23278.40	-52200.00	-23.36	4.57	47.88	1386200.00
ROG_PBDT_PERC	3586.0	336.38	20353.40	-52200.00	-30.60	3.36	52.92	1208700.00
ROG_PBIT_PERC	3586.0	374.70	22462.79	-58500.00	-31.35	2.13	50.14	1338000.00
ROG_PBT_PERC	3586.0	224.07	19659.23	-78900.00	-41.24	0.02	61.96	1160500.00
ROG_PAT_PERC	3586.0	112.23	13480.52	-114500.00	-43.73	0.00	65.35	774200.00
ROG_CP_PERC	3586.0	221.09	13980.20	-52200.00	-29.51	4.62	52.91	822400.00
ROG_REVENUE_EARNINGS_IN_FOREX_PERC	3586.0	37.23	658.67	-100.00	0.00	0.00	0.00	29084.77
ROG_REVENUE_EXPENSES_IN_FOREX_PERC	3586.0	364.86	15233.64	-100.00	0.00	0.00	0.00	894591.69
ROG_MARKET_CAPITALISATION_PERC	3586.0	63.68	1047.93	-98.05	0.00	0.00	47.52	61865.26
CURRENT_RATIO_LATEST	3585.0	12.06	108.41	0.00	0.88	1.36	2.77	4813.00
FIXED_ASSETS_RATIO_LATEST	3585.0	51.54	681.15	0.00	0.27	1.56	4.74	22172.00
INVENTORY_RATIO_LATEST	3585.0	37.80	458.19	0.00	0.00	3.56	8.94	15472.00
DEBTORS_RATIO_LATEST	3585.0	33.03	489.56	0.00	0.42	3.82	8.52	22992.67
TOTAL_ASSET_TURNOVER_RATIO_LATEST	3585.0	1.24	2.67	0.00	0.07	0.60	1.55	57.75
INTEREST_COVER_RATIO_LATEST	3585.0	16.39	351.74	-5450.00	0.00	1.08	3.71	18639.40
PBIDTM_PERC_LATEST	3585.0	-51.16	1795.13	-78870.45	0.00	8.07	18.99	19233.33
PBITM_PERC_LATEST	3585.0	-109.21	3057.64	-141600.00	0.00	5.23	14.29	19195.70
PBDTM_PERC_LATEST	3585.0	-311.57	10921.59	-590500.00	0.00	4.69	14.11	15640.00
CPM_PERC_LATEST	3585.0	-307.01	10676.15	-572000.00	0.00	3.89	11.39	15640.00
APATM_PERC_LATEST	3585.0	-365.06	12500.05	-688600.00	0.00	1.59	7.41	15266.67
DEBTORS_VELOCITY_DAYS	3586.0	603.89	10636.76	0.00	8.00	49.00	106.00	514721.00
CREDITORS_VELOCITY_DAYS	3586.0	2057.85	54169.48	0.00	8.00	39.00	89.00	2034145.00
INVENTORY_VELOCITY_DAYS	3483.0	79.64	137.85	-199.00	0.00	35.00	96.00	996.00
VALUE_OF_OUTPUT_BY_TOTAL_ASSETS	3586.0	0.82	1.20	-0.33	0.07	0.48	1.16	17.63
VALUE_OF_OUTPUT_BY_GROSS_BLOCK	3586.0	61.88	976.82	-61.00	0.27	1.53	4.91	43404.00

- We note from the above statistical summary that there are many features which have values very similar to each other. These features are highly correlated to each other and will lead to multi-collinearity issues later.

- The Correlations are studied further with heat map

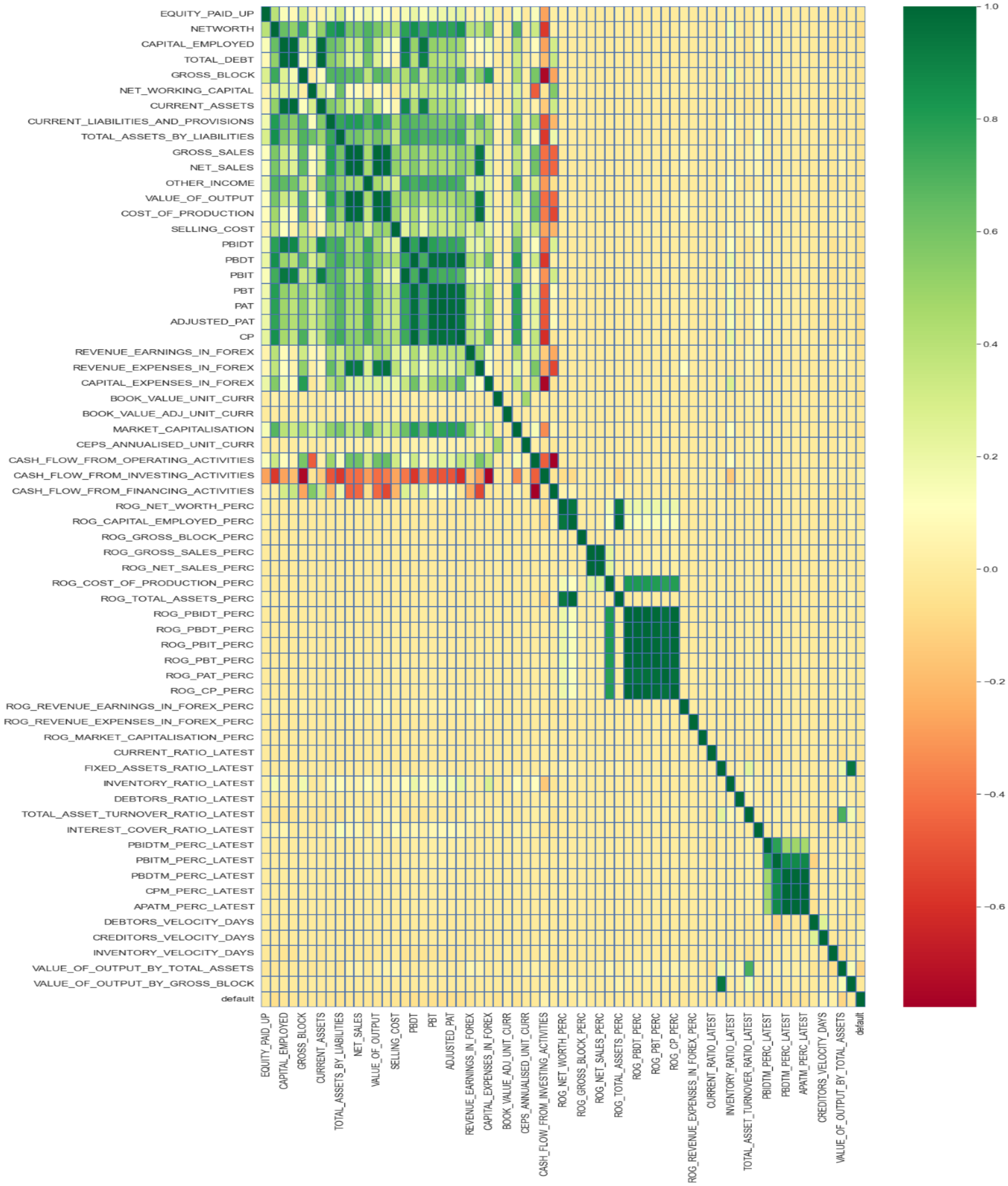


Figure 1: Heat map showing correlations amongst features

- The highly correlated features were dropped, and the data set was reduced to 24 features.
- The correlations for the selected 24 features are checked again

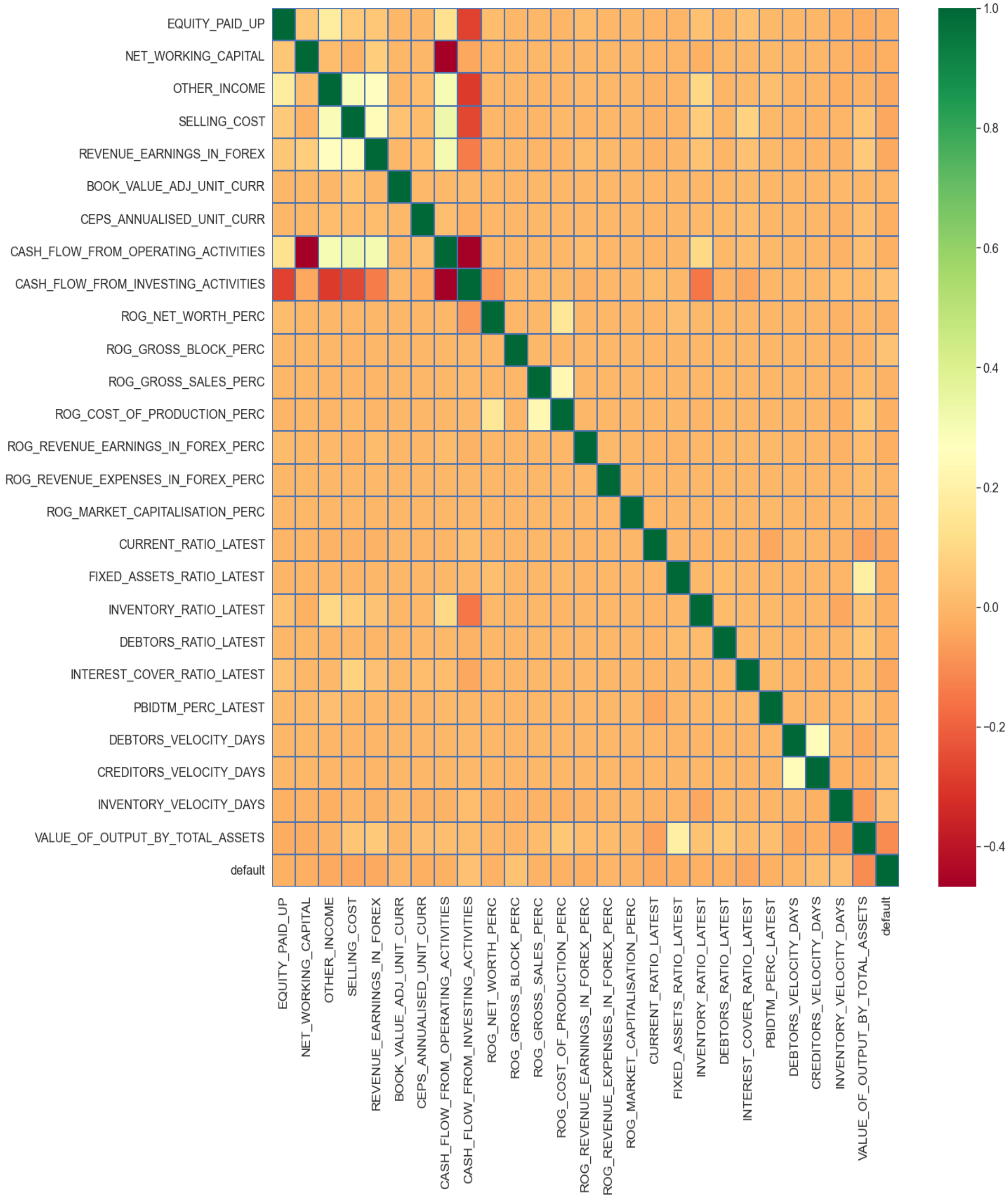


Figure 2: Heat Map after removing correlated features

- The above Data set of 24 features was further cleaned by finding out the outliers in each feature.
- The outliers were replaced by Nan values
- The total Nan values in the data set were close to 18 %
- We also studied the rows which had more than 10% missing records . Row # 2585 had more than 10 % records missing and was accordingly dropped.
- Features 'ROG_REVENUE_EARNINGS_IN_FOREX_PERC', and 'ROG_REVENUE_EXPENSES_IN_FOREX_PERC' were also dropped as they had more than 30 % records missing.
- The Data was subsequently split in the ratio of 67 : 33 in Train and Test data set. Care was taken that the proportion of ‘default’ present in the full data set at 89:11 was maintained for the Train and Test data as well.
- The Train data was scaled using the Standard Scaler tool.
- **Test Data was scaled using the mean and standard deviations of the train data features.**
- After scaling the Null values of the Train Data were imputed using the K Nearest Neighbor tool, using 10 neighbors as the parameter.
- **The null values in the Test data were imputed by values as ascertained by nearest neighbors of the Train Data**

- Train and Test Data

Table 4: Train and Test Data

Train Data Set

	EQUITY_PAID_UP	NET_WORKING_CAPITAL	OTHER_INCOME	SELLING_COST	REVENUE_EARNINGS_IN_FOREX	BOOK_VALUE_ADJ_UNIT_CURR	CEPS_AI
1913	0.124056	-0.183439	0.634139	-0.132107	0.242183	-0.132127	
2811	0.147981	2.678147	0.199973	-0.475610	-0.325055	1.291481	
3172	0.523167	1.245454	4.084114	1.630015	0.751015	1.208300	
1494	-0.343567	1.486209	-0.533545	-0.383326	-0.325055	-0.739478	
750	-0.728541	-0.536898	-0.576006	-0.506372	-0.325055	-0.479814	
...	...	...	...	...	...	...	
1436	-1.039565	-0.595954	-0.576006	-0.506372	-0.325055	-0.223611	
2301	1.609577	1.508721	-0.188547	0.201143	1.328813	-0.505340	
2024	0.935329	2.649788	3.197736	2.918410	0.051124	-0.735077	
2719	-0.321817	0.872260	0.602293	1.624888	0.428293	0.215266	
2521	0.290443	-0.117951	1.021598	0.242158	-0.272258	0.261624	

2401 rows × 24 columns

Test Data Set

	EQUITY_PAID_UP	NET_WORKING_CAPITAL	OTHER_INCOME	SELLING_COST	REVENUE_EARNINGS_IN_FOREX	BOOK_VALUE_ADJ_UNIT_CURR	CEPS_AI
350	-0.368580	-0.602678	-0.576006	-0.490991	-0.325055	-0.826327	
1196	-0.320730	-0.575489	-0.576006	-0.506372	-0.325055	-0.787597	
2141	1.628064	-0.089300	-0.512314	-0.460230	-0.325055	-0.749748	
2267	0.255643	0.075297	0.803984	0.835343	0.139559	0.175656	
1458	-0.553454	-0.528420	-0.570698	-0.506372	-0.325055	-0.410570	
...	...	...	...	...	...	...	
2067	-0.515392	1.068432	-0.565391	2.108357	-0.173264	0.720512	
2028	-0.427305	0.105994	-0.183240	2.395464	5.476024	0.501924	
1995	0.474230	1.119009	0.384679	0.266255	0.597904	-0.088703	
831	-1.028690	-0.534559	-0.570698	-0.121853	-0.325055	2.070475	
1265	-0.543667	-0.424925	-0.406161	-0.501245	-0.325055	-0.334577	

1184 rows × 24 columns

- The target variable ‘default’ in Train and Test

Table 5: Target Variable

```
Train Data Set
```

	default
1913	0
2811	0
3172	0
1494	0
750	0
...	...
1436	0
2301	0
2024	0
2719	0
2521	0

2401 rows × 1 columns

```
Test Data Set
```

	default
350	1
1196	0
2141	0
2267	0
1458	0
...	...
2067	0
2028	0
1995	0
831	0
1265	0

1184 rows × 1 columns

- X_train and X_test is concatenated so that X_train is sitting over X_test. That is the first 2401 rows of the combined data frame is Train Data and the all rows below that are the Test data of the combined Data set.

- Similar to the independent variables (X) the dependent variable (y) train and test data sets are concatenated.

- The Correlations of the scaled , imputed, and reduced dataframe are checked again.

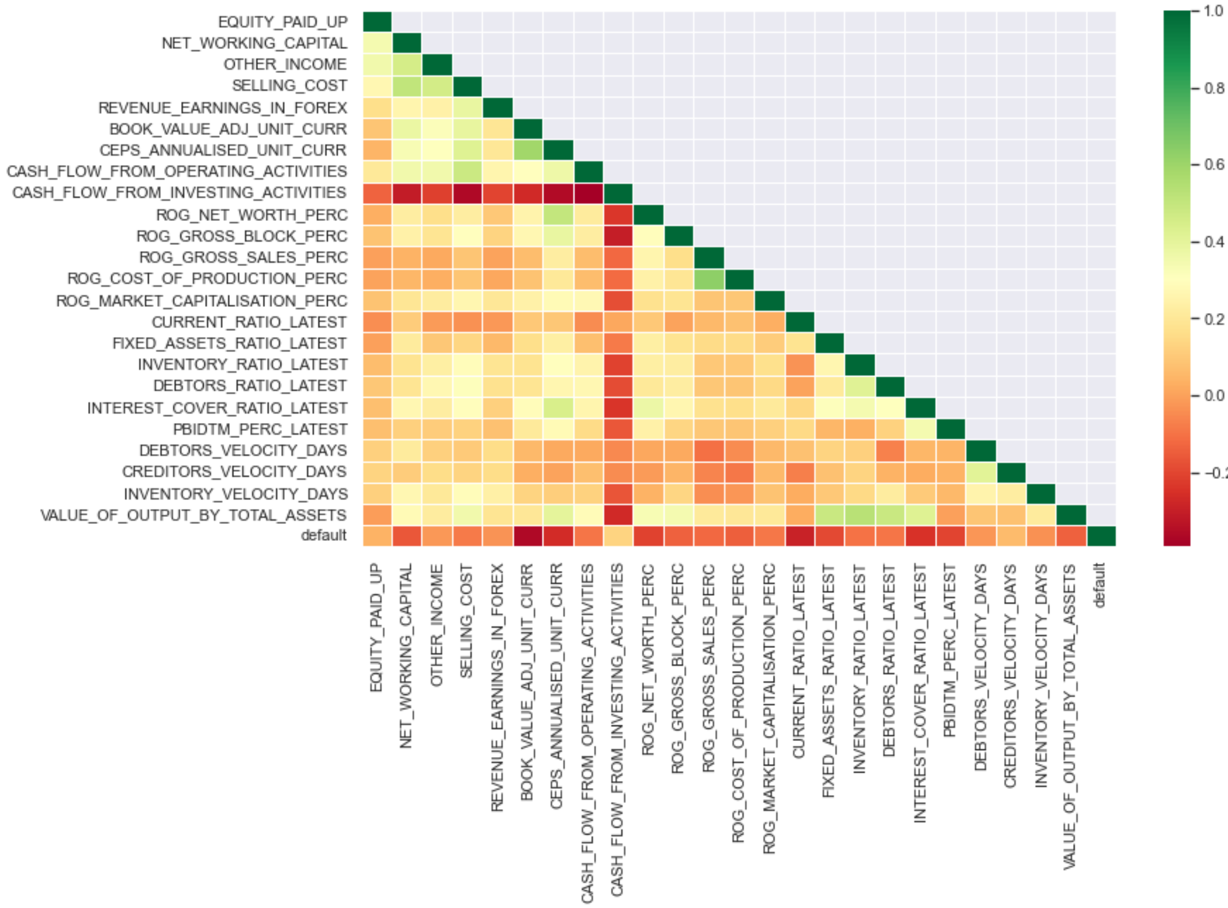


Figure 3: Correlations amongst Features after scaling and imputing

- No collinearity exists.
- Variance Influence Factor check

Table 6: VIF dataframe for features

	VIF
CEPS_ANNUALISED_UNIT_CURR	2.342458
VALUE_OF_OUTPUT_BY_TOTAL_ASSETS	2.327682
SELLING_COST	2.001325
NET_WORKING_CAPITAL	1.781558
ROG_GROSS_SALES_PERC	1.746198
ROG_COST_OF_PRODUCTION_PERC	1.713397
BOOK_VALUE_ADJ_UNIT_CURR	1.710451
INTEREST_COVER_RATIO_LATEST	1.612414
OTHER_INCOME	1.566357
CASH_FLOW_FROM_OPERATING_ACTIVITIES	1.565049
INVENTORY_RATIO_LATEST	1.551107
DEBTORS_RATIO_LATEST	1.539226
ROG_NET_WORTH_PERC	1.480845
FIXED_ASSETS_RATIO_LATEST	1.433282
CASH_FLOW_FROM_INVESTING_ACTIVITIES	1.380912
DEBTORS_VELOCITY_DAYS	1.375988
ROG_GROSS_BLOCK_PERC	1.319728
CREDITORS_VELOCITY_DAYS	1.286420
EQUITY_PAID_UP	1.283740
PBIDTM_PERC_LATEST	1.279328
REVENUE_EARNINGS_IN_FOREX	1.261746
INVENTORY_VELOCITY_DAYS	1.243324
ROG_MARKET_CAPITALISATION_PERC	1.167552
CURRENT_RATIO_LATEST	1.124366

- All the VIF values are less than 5 , which implies that the chosen features are not correlated to each other and are independent.

- The combined data set is again split into train and test sets. We had already split the data earlier and then concatenated one over the other. So now we just choose the first 2401 rows as Train and the rest as Test. It should be noted that the dependent variable ‘ default’ (y) is part of the Train and Test set , as that is the requirement of the Stats Model.

- Logistic Regression Model 1 with all the 24 features -Summary Report

Table 7: Summary report of Stats Model for Logistic Regression Model

Logit Regression Results

Dep. Variable:	default	**No. Observations:**	2401
Model:	Logit	**Df Residuals:**	2376
Method:	MLE	**Df Model:**	24
Date:	Sat, 19 Nov 2022	**Pseudo R-squ.:**	0.5288
Time:	07:10:52	**Log-Likelihood:**	-387.99
converged:	True	**LL-Null:**	-823.35
Covariance Type:	nonrobust	**LLR p-value:**	2.303e-166

	coef	std err	z	P>|z|	[0.025	0.975]
Intercept	-4.8395	0.240	-20.127	0.000	-5.311	-4.368
EQUITY_PAID_UP	0.1146	0.107	1.068	0.286	-0.096	0.325
NET_WORKING_CAPITAL	-0.2913	0.159	-1.829	0.067	-0.603	0.021
OTHER_INCOME	0.3743	0.128	2.935	0.003	0.124	0.624
SELLING_COST	0.4902	0.149	3.280	0.001	0.197	0.783
REVENUE_EARNINGS_IN_FOREX	-0.0241	0.115	-0.210	0.833	-0.249	0.201
BOOK_VALUE_ADJ_UNIT_CURR	-3.2946	0.278	-11.858	0.000	-3.839	-2.750
CEPS_ANNUALISED_UNIT_CURR	-0.2280	0.181	-1.259	0.208	-0.583	0.127
CASH_FLOW_FROM_OPERATING_ACTIVITIES	-0.1293	0.140	-0.924	0.355	-0.404	0.145
CASH_FLOW_FROM_INVESTING_ACTIVITIES	0.1717	0.148	1.163	0.245	-0.118	0.461
ROG_NET_WORTH_PERC	-0.1415	0.121	-1.168	0.243	-0.379	0.096
ROG_GROSS_BLOCK_PERC	-0.0819	0.157	-0.522	0.602	-0.389	0.226
ROG_GROSS_SALES_PERC	0.1560	0.134	1.168	0.243	-0.106	0.418
ROG_COST_OF_PRODUCTION_PERC	-0.4813	0.134	-3.579	0.000	-0.745	-0.218
ROG_MARKET_CAPITALISATION_PERC	-0.0020	0.106	-0.019	0.985	-0.210	0.206
CURRENT_RATIO_LATEST	-1.5085	0.181	-8.356	0.000	-1.862	-1.155
FIXED_ASSETS_RATIO_LATEST	-0.4430	0.203	-2.186	0.029	-0.840	-0.046
INVENTORY_RATIO_LATEST	-0.0386	0.129	-0.299	0.765	-0.292	0.215
DEBTORS_RATIO_LATEST	-0.0445	0.126	-0.354	0.724	-0.291	0.202
INTEREST_COVER_RATIO_LATEST	-0.3723	0.157	-2.369	0.018	-0.680	-0.064
PBIDTM_PERC_LATEST	-0.1880	0.116	-1.623	0.105	-0.415	0.039
DEBTORS_VELOCITY_DAYS	0.1626	0.108	1.503	0.133	-0.050	0.375
CREDITORS_VELOCITY_DAYS	0.1500	0.100	1.497	0.134	-0.046	0.346
INVENTORY_VELOCITY_DAYS	0.0018	0.117	0.016	0.988	-0.227	0.231
VALUE_OF_OUTPUT_BY_TOTAL_ASSETS	0.3834	0.165	2.328	0.020	0.061	0.706

Possibly complete quasi-separation: A fraction 0.14 of observations can be perfectly predicted. This might indicate that there is complete quasi-separation. In this case some parameters will not be identified.

- There are a lot of features whose P-value is more than 0.05 and are insignificant. They are dropped one at a time and models are built again.

- Final Model , where all P_value is We less than 0.05

Table 8: Final Summary Report with 8 Features

Logit Regression Results

Dep. Variable:	default	**No. Observations:**	2401
Model:	Logit	**Df Residuals:**	2392
Method:	MLE	**Df Model:**	8
Date:	Sat, 19 Nov 2022	**Pseudo R-squ.:**	0.5150
Time:	07:10:54	**Log-Likelihood:**	-399.33
converged:	True	**LL-Null:**	-823.35
Covariance Type:	nonrobust	**LLR p-value:**	9.034e-178

	coef	std err	z	P>|z|	[0.025	0.975]
Intercept	-4.7205	0.229	-20.634	0.000	-5.169	-4.272
NET_WORKING_CAPITAL	-0.3027	0.149	-2.028	0.043	-0.595	-0.010
OTHER_INCOME	0.4031	0.118	3.424	0.001	0.172	0.634
SELLING_COST	0.4977	0.130	3.838	0.000	0.244	0.752
BOOK_VALUE_ADJ_UNIT_CURR	-3.4553	0.270	-12.793	0.000	-3.985	-2.926
ROG_COST_OF_PRODUCTION_PERC	-0.4675	0.110	-4.256	0.000	-0.683	-0.252
CURRENT_RATIO_LATEST	-1.4824	0.164	-9.021	0.000	-1.804	-1.160
INTEREST_COVER_RATIO_LATEST	-0.4515	0.138	-3.261	0.001	-0.723	-0.180
PBIDTM_PERC_LATEST	-0.2307	0.111	-2.077	0.038	-0.448	-0.013

Possibly complete quasi-separation: A fraction 0.13 of observations can be perfectly predicted. This might indicate that there is complete quasi-separation. In this case some parameters will not be identified.

- So, starting from 66 features , we have now built a model which is only going to use 8 features as mentioned above.

- The Confusion Matrix for Train and Test Data for above model is. Please note that our focus is on 'default' . So, we shall concentrate on recall and precision readings for 1

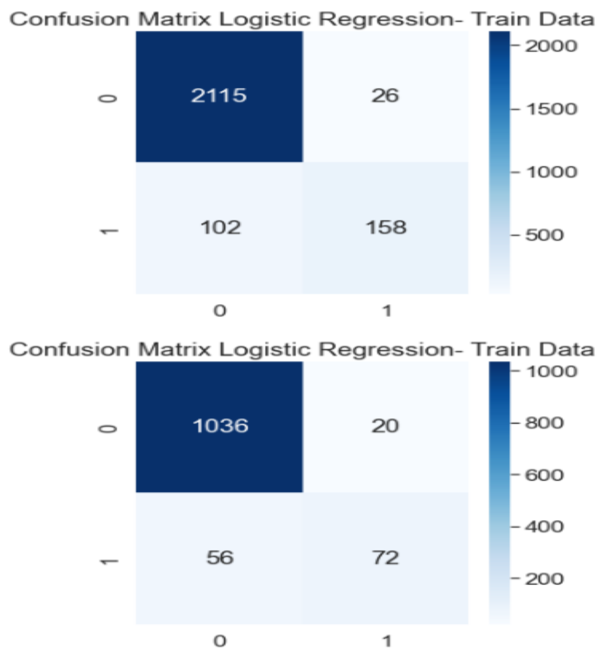


Figure 4: Confusion Matrix - Logistic Regression Model

- Classification Report

Classification Report - Logistic Regression- Train Data

	precision	recall	f1-score	support
0	0.95	0.99	0.97	2141
1	**0.86**	**0.61**	0.71	260
accuracy			0.95	2401
macro avg	0.91	0.80	0.84	2401
weighted avg	0.94	0.95	0.94	2401

Classification Report - Logistic Regression- Test Data

	precision	recall	f1-score	support
0	0.95	0.98	0.96	1056
1	**0.78**	**0.56**	0.65	128
accuracy			0.94	1184
macro avg	0.87	0.77	0.81	1184
weighted avg	0.93	0.94	0.93	1184

- The model is further improved by optimizing the threshold level to 0.165 . (if predicted probability is greater than 0.165 , outcome predicted is 1 , in the earlier model the threshold was 0.5)

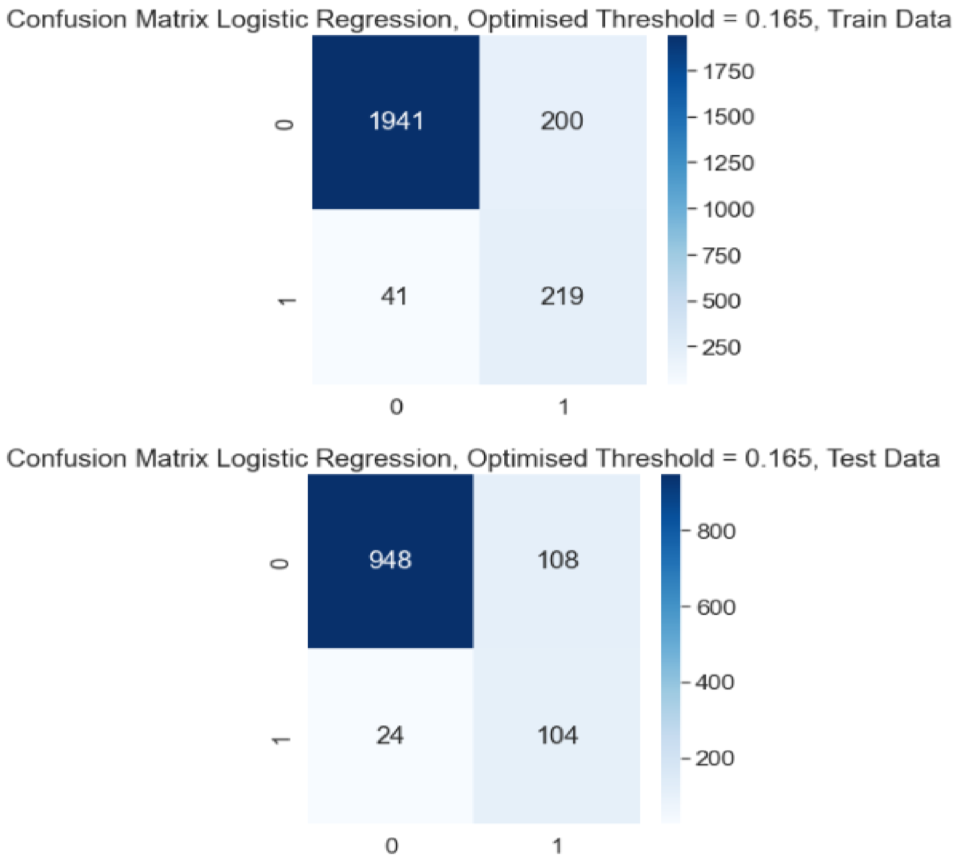


Figure 5:Confusion Matrix Log Reg Threshold Optimized

- Classification Reports (Optimized Threshold)

Classification Report - Logistic Regression, Optimized Threshold = 0.165,Train Data

	precision	recall	f1-score	support
0	0.98	0.91	0.94	2141
1	**0.52**	**0.84**	0.65	260
accuracy			0.90	2401
macro avg	0.75	0.87	0.79	2401
weighted avg	0.93	0.90	0.91	2401

Classification Report - Logistic Regression, Optimized Threshold = 0.165,Test Data

	precision	recall	f1-score	support
0	0.98	0.90	0.93	1056
1	**0.49**	**0.81**	0.61	128
accuracy			0.89	1184
macro avg	0.73	0.86	0.77	1184
weighted avg	0.92	0.89	0.90	1184

- After optimizing the threshold , we note that the recall for 1 has improved above 80 % , although the precision has gone down to close to 50% from 80 %.

- We try and balance the data using SMOTE. (engineer data , so that the proportion of 'default' increases in the Train data set , so that the machine has more data to learn from for 'default' cases)

- Sampling Strategy Ratio - Initial proportion of 'default' in original data is 388 / 3197 = 0.12 . (388 +3197 =3585)

- Using the SMOTE tool, we iterate the sampling strategy ratio and see that by increasing the proportion of default by engineering synthetic data , we get models which over fit. The most ideal ratio ascertained was 0.15 , where there was an improvement in the model performance and the Recall and Precision readings for '1' were reasonably together.

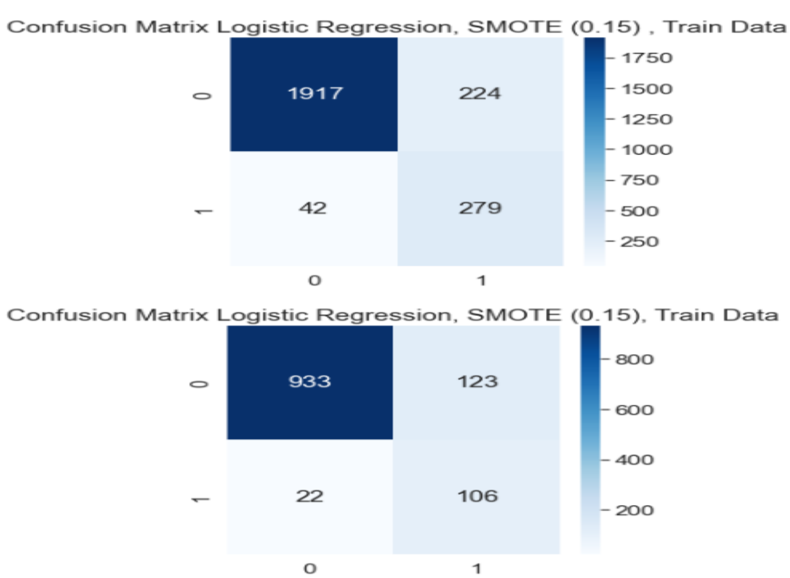


Figure 6: Confusion Matrix -Smote

- Classification Report – SMOTE , threshold = 0.5

Classification Report- Logistic Regression, SMOTE (0.15) , Train Data

	precision	recall	f1-score	support
0	0.98	0.90	0.94	2141
1	0.55	0.87	0.68	321
accuracy			0.89	2462
macro avg	0.77	0.88	0.81	2462
weighted avg	0.92	0.89	0.90	2462

Classification Report- Logistic Regression, SMOTE (0.15) , Test Data

	precision	recall	f1-score	support
0	0.98	0.88	0.93	1056
1	0.46	0.83	0.59	128
accuracy			0.88	1184
macro avg	0.72	0.86	0.76	1184
weighted avg	0.92	0.88	0.89	1184

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

- The train data set (2401 observations with 24 features) without the dependent variable ('default') is fitted into the random forest model with all parameters selected by default as offered by the model.

- The performance metrics are as follows

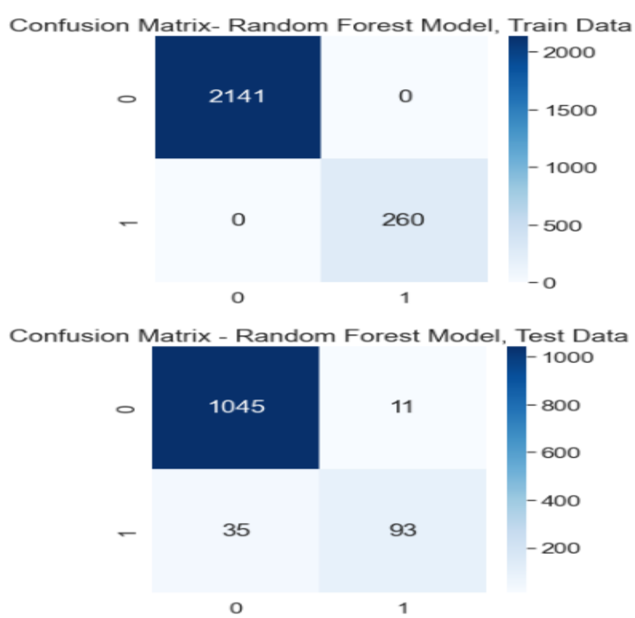


Figure 7: Confusion Matrix - Random Forest

- Classification Report for Random Forest

Classification Report- Random Forest Model, Train Data

	precision	recall	f1-score	support
0	1.00	1.00	1.00	2141
1	**1.00**	**1.00**	1.00	260
accuracy			1.00	2401
macro avg	1.00	1.00	1.00	2401
weighted avg	1.00	1.00	1.00	2401

Classification Report- Random Forest Model, Train Data

	precision	recall	f1-score	support
0	0.97	0.99	0.98	1056
1	**0.89**	**0.73**	0.80	128
accuracy			0.96	1184
macro avg	0.93	0.86	0.89	1184
weighted avg	0.96	0.96	0.96	1184

- The Random forest model is an over fit case , as in the train data it predicts everything correct both for 0 and 1. However in the train data the recall and precision comes down to 73 % and 89 %.
- We shall try and modify the model by varying the parameters by using the GridSearch module .
- The first set of parameters fed to the Grid search tool are

'max_depth': [3, 5, 7],

'min_samples_leaf': [5, 10, 15],

'min_samples_split': [15, 30, 15],

'n_estimators': [25, 50]

- The best parameters selected are

'max_depth': 7,

'min_samples_leaf': 5,

'min_samples_split': 15,

'n_estimators': 50

- Confusion Matrix of Random Forest with optimized parameters -Iteration 1

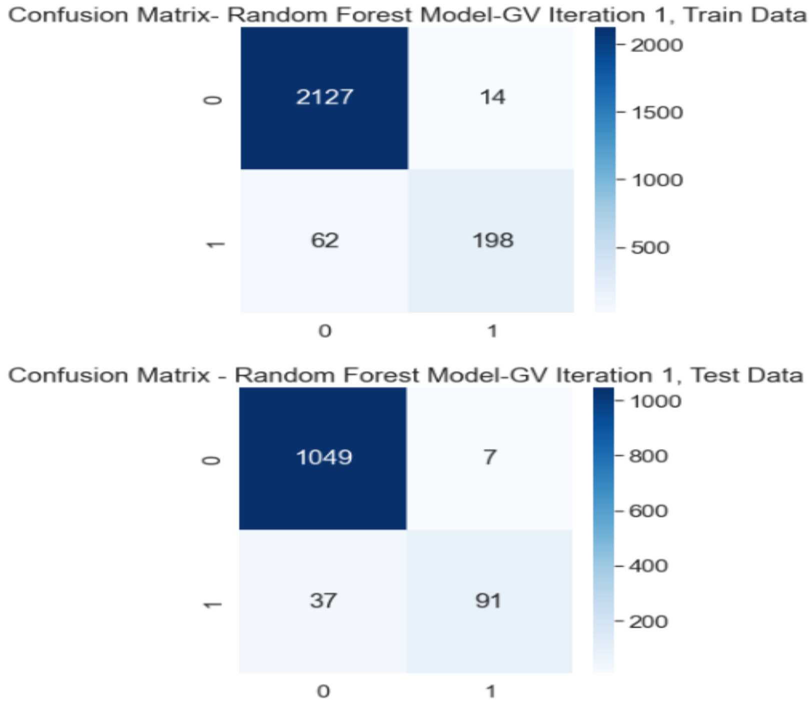


Figure 8: Confusion Matrix - Random Forest GV1

- Classification Report of Random Forest with optimized parameters -Iteration 1

Classification Report- Random Forest Model-GV Iteration 1, Train Data

	precision	recall	f1-score	support
0	0.97	0.99	0.98	2141
1	**0.93**	**0.76**	0.84	260
accuracy			0.97	2401
macro avg	0.95	0.88	0.91	2401
weighted avg	0.97	0.97	0.97	2401

Classification Report- Random Forest Model-GV Iteration 1 Train Data

	precision	recall	f1-score	support
0	0.97	0.99	0.98	1056
1	**0.93**	**0.71**	0.81	128
accuracy			0.96	1184
macro avg	0.95	0.85	0.89	1184
weighted avg	0.96	0.96	0.96	1184

- With a recall of 71 % and 93 % precision , in the test data – the model is able to correctly identify 73 % of the default cases . Of all the cases the model identifies as 'default' – 93% are correct.

The recall and precision figures for test and train data are similar and hence we can say that this model is an ideal fit.

- Random Forest – best parameters selection , Iteration 2

- The second set of parameters fed to the Grid search tool are

'max_depth': [5, 7, 9],

'min_samples_leaf': [10,15,20],

'min_samples_split': [30,45,60],

'n_estimators': [50,100]

- The best parameters selected are

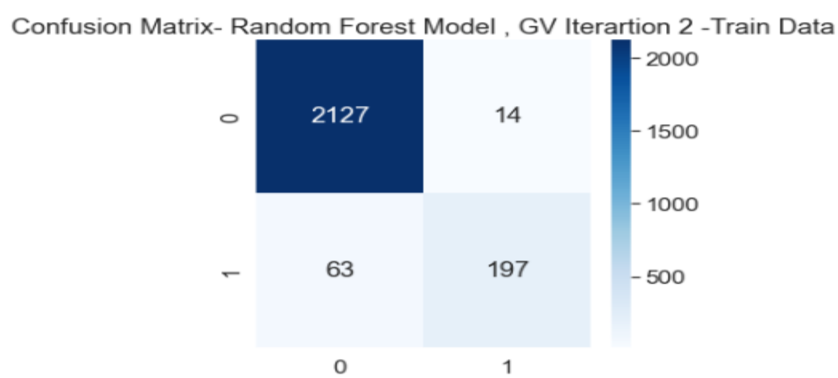
'max_depth': 9,

'min_samples_leaf': 10,

'min_samples_split': 60,

'n_estimators': 100,

- Confusion Matrix of Random Forest with optimized parameters -Iteration 2



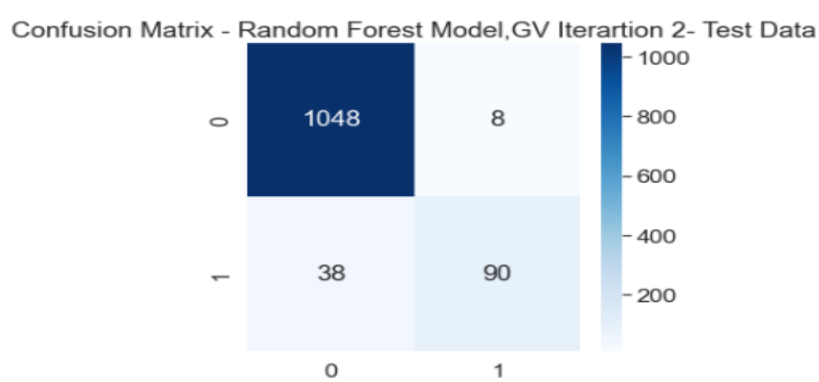


Figure 9: Confusion Matrix Random Forest GV2

- Classification Report of Random Forest with optimized parameters -Iteration 2

Classification Report- Random Forest Model,GV Iteration 2- Train Data

	precision	recall	f1-score	support
0	0.97	0.99	0.98	2141
1	**0.93**	**0.76**	0.84	260
accuracy			0.97	2401
macro avg	0.95	0.88	0.91	2401
weighted avg	0.97	0.97	0.97	2401

Classification Report- Random Forest Model , GV Iteration 2 -Test Data

	precision	recall	f1-score	support
0	0.97	0.99	0.98	1056
1	**0.92**	**0.70**	0.80	128
accuracy			0.96	1184
macro avg	0.94	0.85	0.89	1184
weighted avg	0.96	0.96	0.96	1184

- Random Forest GV 1 performs better

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

- Validation and Performance metrics of the 3 Random Forest models built have already been shown above.
- The model with best parameters as

'max_depth': 7,

'min_samples_leaf': 5,

'min_samples_split': 15,

'n_estimators': 50

performs the best

Classification Report- Random Forest Model-GV Iteration 1, Train Data

	precision	recall	f1-score	support
0	0.97	0.99	0.98	2141
1	**0.93**	**0.76**	0.84	260
accuracy			0.97	2401
macro avg	0.95	0.88	0.91	2401
weighted avg	0.97	0.97	0.97	2401

Classification Report- Random Forest Model-GV Iteration 1 Train Data

	precision	recall	f1-score	support
0	0.97	0.99	0.98	1056
1	**0.93**	**0.71**	0.81	128
accuracy			0.96	1184
macro avg	0.95	0.85	0.89	1184
weighted avg	0.96	0.96	0.96	1184

- The recall and precision for '1' (default cases) and '0' (non-default) cases for test and train data are similar as highlighted in red above. So this model is an ideal fit .
- At 71 % recall the model is able to correctly identify 71 % of all default cases.
- Of all the predictions that the model makes for 'default' cases, 93 % (precision reading)are correct.

4 Build an LDA Model on Train Dataset. Also showcase your model building approach

- LinearDiscriminantAnalysis tool is imported from the sklearn.discriminant_analysis library of Python
- The Linear Discriminant Model is built on the Train Data with 2401 observations and 24 features (scaled and outliers treated).
- LDA model from the sklearn library is capable of predicting the outcome (0 or 1) along with their probabilities.
- We predict the outcome and analyze the results

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

- Confusion Matrix of Train Data for LDA Model

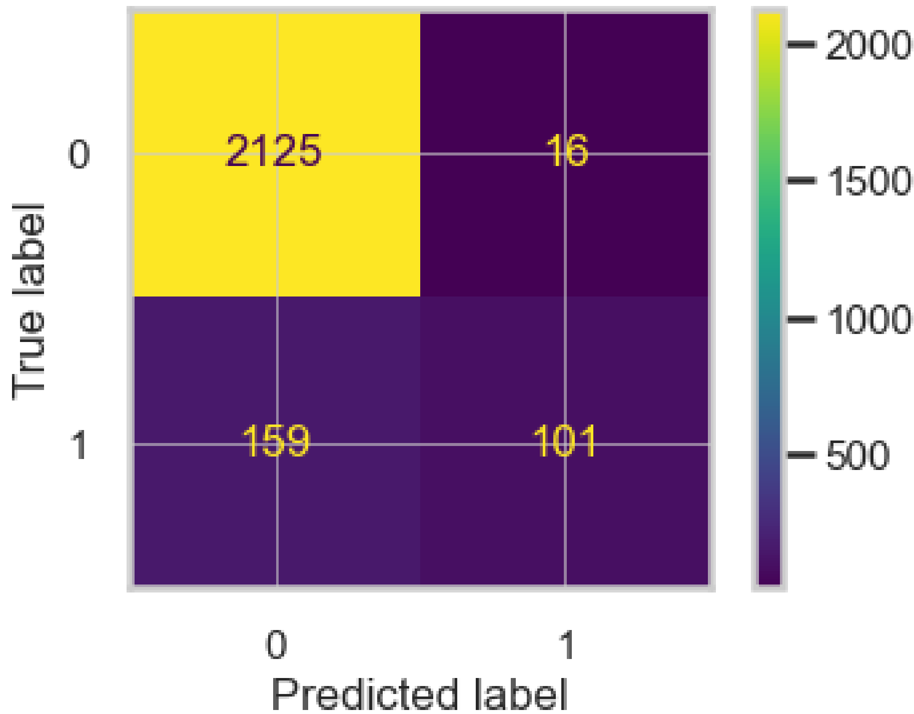


Figure 10: Confusion Matrix - Train Data LDA model

- Confusion Matrix of Test Data for LDA Model

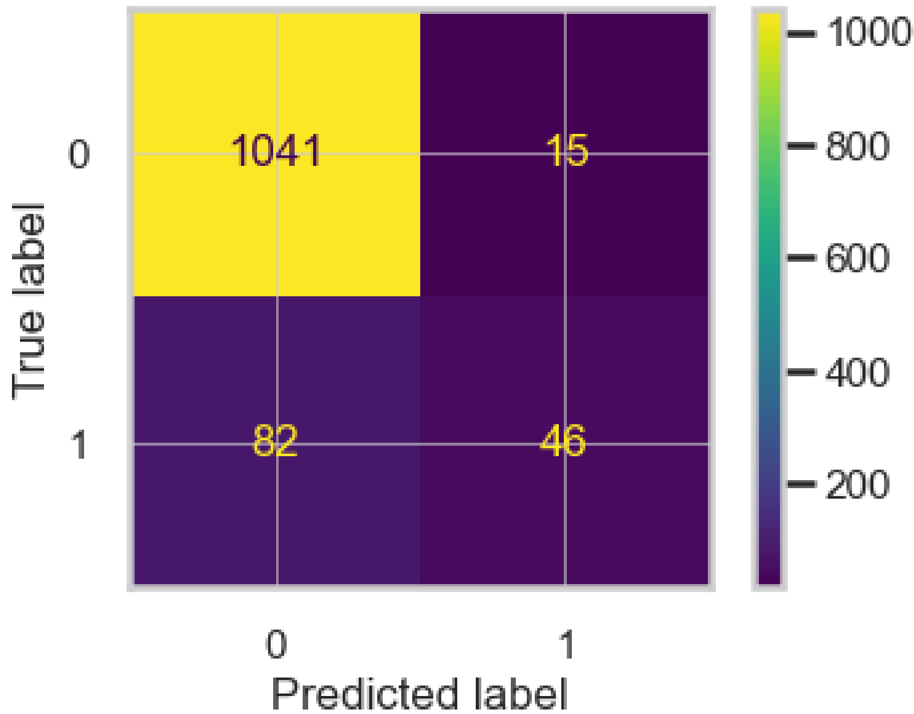


Figure 11: Confusion Matrix - Test Data LDA model

- Classification Report for the LDA model for Train and Test Data

Classification Report- LDA-Model, Train Data

	precision	recall	f1-score	support
0	0.93	0.99	0.96	2141
1	**0.86**	**0.39**	0.54	260
accuracy			0.93	2401
macro avg	0.90	0.69	0.75	2401
weighted avg	0.92	0.93	0.91	2401

Classification Report- LDA-Model, Test Data

	precision	recall	f1-score	support
0	0.93	0.99	0.96	1056
1	**0.75**	**0.36**	0.49	128
accuracy			0.92	1184
macro avg	0.84	0.67	0.72	1184
weighted avg	0.91	0.92	0.90	1184

- The LDA model is able to identify only 36 % of the defaulters in the Test data. Only 75 % of its predictions on defaulters are correct in the test data.

  The performance of this model is poor .

  The features provided are not able to segregate the two classes of 'default' and 'non-default'

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

- 3 Models were built in Logistic Regression
    - Model 1 with default settings of parameters
    - Model 2 – threshold probability revised to optimum levels to maximize the difference between True Positive and False Positive rates
    - Model 3 – Balancing of Data on Default. Proportion revised to 0.15

- The performance metrics of Logistic Models are

Table 9: Classification Reports- Logistic Regression Models

Classification Report - Logistic Regression Train Data				
	precision	recall	f1-score	support
0	0.95	0.99	0.97	2141
1	0.86	0.61	0.71	260
accuracy			0.95	2401
macro avg	0.91	0.80	0.84	2401
weighted avg	0.94	0.95	0.94	2401
Classifiaction Report - Logistic Regression Test Data				
	precision	recall	f1-score	support
0	0.95	0.98	0.96	1056
1	0.78	0.56	0.65	128
accuracy			0.94	1184
macro avg	0.87	0.77	0.81	1184
weighted avg	0.93	0.94	0.93	1184

Classification Report - Logistic Regression, Optimised Threshold = 0.165, Train Data				
	precision	recall	f1-score	support
0	0.98	0.91	0.94	2141
1	0.52	0.84	0.65	260
accuracy			0.90	2401
macro avg	0.75	0.87	0.79	2401
weighted avg	0.93	0.90	0.91	2401
Classification Report - Logistic Regression, Optimised Threshold = 0.165, Test Data				
	precision	recall	f1-score	support
0	0.98	0.90	0.93	1056
1	0.49	0.81	0.61	128
accuracy			0.89	1184
macro avg	0.73	0.86	0.77	1184
weighted avg	0.92	0.89	0.90	1184

Classification Report- Logistic Regression, SMOTE (0.15) , Train Data				
	precision	recall	f1-score	support
0	0.98	0.90	0.94	2141
1	0.55	0.87	0.68	321
accuracy			0.89	2462
macro avg	0.77	0.88	0.81	2462
weighted avg	0.92	0.89	0.90	2462
Classification Report- Logistic Regression, SMOTE (0.15) , Test Data				
	precision	recall	f1-score	support
0	0.98	0.88	0.93	1056
1	0.46	0.83	0.59	128
accuracy			0.88	1184
macro avg	0.72	0.86	0.76	1184
weighted avg	0.92	0.88	0.89	1184

- Amongst the Logistic Regression Models – we select Model 2 , as it gives more or less the same level of performance on recall and precision of '1' (default) without engineered data.

- We then created 3 Random Forest Models
    - RF Model 1 – with default setting of parameters
    - RF Model 2 – with Grid Search for Parameters – Iterartion1
    - RF Model 3 – with Grid Search for Parameters – Iterartion1

Table 10: Classification Reports- Random Forest Models

Classification Report- Random Forest Model, Train Data					Classification Report- Random Forest Model-GV Iteration 1, Train Data					Classification Report- Random Forest Model,GV Iterartion 2- Train Data				
	precision	recall	f1-score	support		precision	recall	f1-score	support		precision	recall	f1-score	support
0	1.00	1.00	1.00	2141	0	0.98	0.99	0.99	2141	0	0.97	0.99	0.98	2141
1	1.00	1.00	1.00	260	1	0.94	0.81	0.87	260	1	0.94	0.76	0.84	260
accuracy			1.00	2401	accuracy			0.97	2401	accuracy			0.97	2401
macro avg	1.00	1.00	1.00	2401	macro avg	0.96	0.90	0.93	2401	macro avg	0.95	0.88	0.91	2401
weighted avg	1.00	1.00	1.00	2401	weighted avg	0.97	0.97	0.97	2401	weighted avg	0.97	0.97	0.97	2401
Classification Report- Random Forest Model, Train Data					Classification Report- Random Forest Model-GV Iteration 1 Test Data					Classification Report- Random Forest Model,GV Iterartion 2- Test Data				
	precision	recall	f1-score	support		precision	recall	f1-score	support		precision	recall	f1-score	support
0	0.97	0.99	0.98	1056	0	0.97	0.99	0.98	1056	0	0.96	0.99	0.98	1056
1	0.91	0.73	0.81	128	1	0.91	0.71	0.80	128	1	0.93	0.69	0.79	128
accuracy			0.96	1184	accuracy			0.96	1184	accuracy			0.96	1184
macro avg	0.94	0.86	0.89	1184	macro avg	0.94	0.85	0.89	1184	macro avg	0.94	0.84	0.88	1184
weighted avg	0.96	0.96	0.96	1184	weighted avg	0.96	0.96	0.96	1184	weighted avg	0.96	0.96	0.96	1184

- We selected RF Model 2 because the parameters selected in this model are smaller as highlighted earlier.
- The Logistic , Random Forest and Linear Discriminant Analysis Models comparison is as under

Table 11: Classification Reports -final comparison

Classification Report - Logistic Regression, Optimised Threshold = 0.165, Train Data					Classification Report- Random Forest Model-GV Iteration 1, Train Data					Classification Report- LDA-Model, Train Data				
	precision	recall	f1-score	support		precision	recall	f1-score	support		precision	recall	f1-score	support
0	0.98	0.91	0.94	2141	0	0.98	0.99	0.99	2141	0	0.93	0.99	0.96	2141
1	0.52	0.84	0.65	260	1	0.94	0.81	0.87	260	1	0.86	0.39	0.54	260
accuracy			0.90	2401	accuracy			0.97	2401	accuracy			0.93	2401
macro avg	0.75	0.87	0.79	2401	macro avg	0.96	0.90	0.93	2401	macro avg	0.90	0.69	0.75	2401
weighted avg	0.93	0.90	0.91	2401	weighted avg	0.97	0.97	0.97	2401	weighted avg	0.92	0.93	0.91	2401
Classification Report - Logistic Regression, Optimised Threshold = 0.165, Test Data					Classification Report- Random Forest Model-GV Iteration 1 Test Data					Classification Report- LDA-Model, Test Data				
	precision	recall	f1-score	support		precision	recall	f1-score	support		precision	recall	f1-score	support
0	0.98	0.90	0.93	1056	0	0.97	0.99	0.98	1056	0	0.93	0.99	0.96	1056
1	0.49	0.81	0.61	128	1	0.91	0.71	0.80	128	1	0.75	0.36	0.49	128
accuracy			0.89	1184	accuracy			0.96	1184	accuracy			0.92	1184
macro avg	0.73	0.86	0.77	1184	macro avg	0.94	0.85	0.89	1184	macro avg	0.84	0.67	0.72	1184
weighted avg	0.92	0.89	0.90	1184	weighted avg	0.96	0.96	0.96	1184	weighted avg	0.91	0.92	0.90	1184

- The Random Forest model performs the best. It is able to identify 71 % of the defaulters with 91 % accuracy in Test data.

- The ROC curve of the Random Forest Model – GV1

```
AUC Train Data Random Forest GV1 model: 0.996

AUC Test Data Random Forest GV1 model: 0.975
```

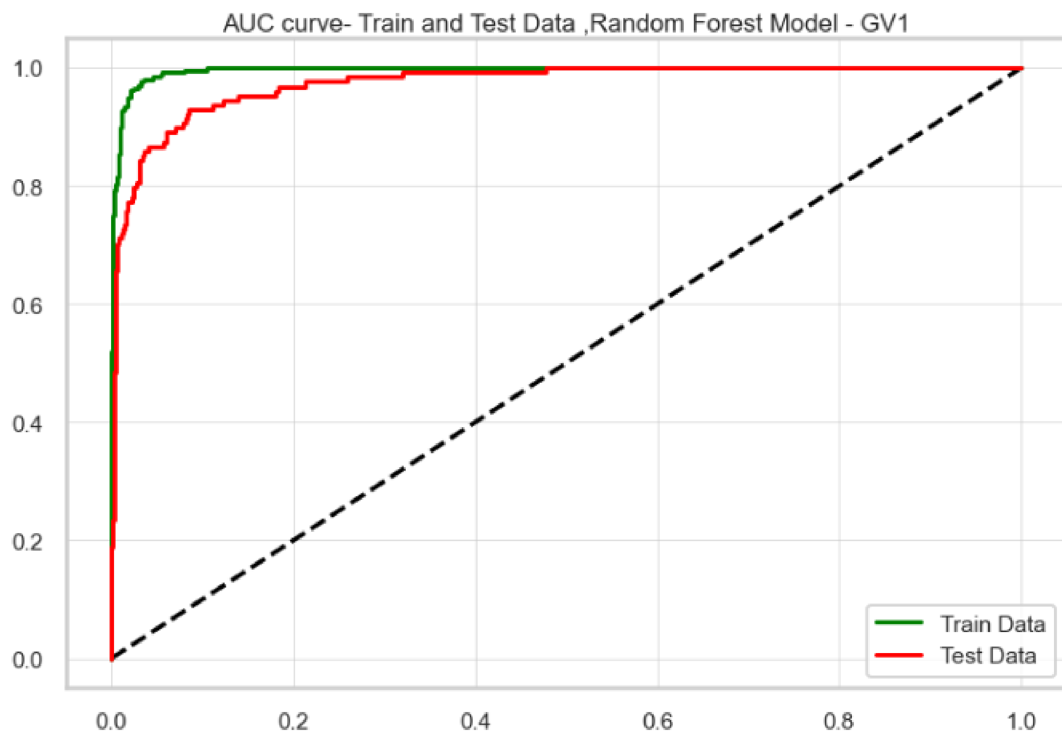


Figure 12: ROC curve of the selected model

7 State Recommendations regarding the above models

- The Random Forest model with optimized parameters performs the best amongst the models built – Logistic Regression and Linear Discriminant Analysis
- The model will be able to predict 71 % of the defaulters with 91 % accuracy.
- To improve the model further we will need to gather more meaningful data. It has to be noted that 18 % of the data provided were outliers, which were removed and imputed by values obtained the K-nearest neighbor algorithm.
- The performance of the models will improve with the quality of data.

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

- Stock Price vs Time for Infosys Stocks over the years

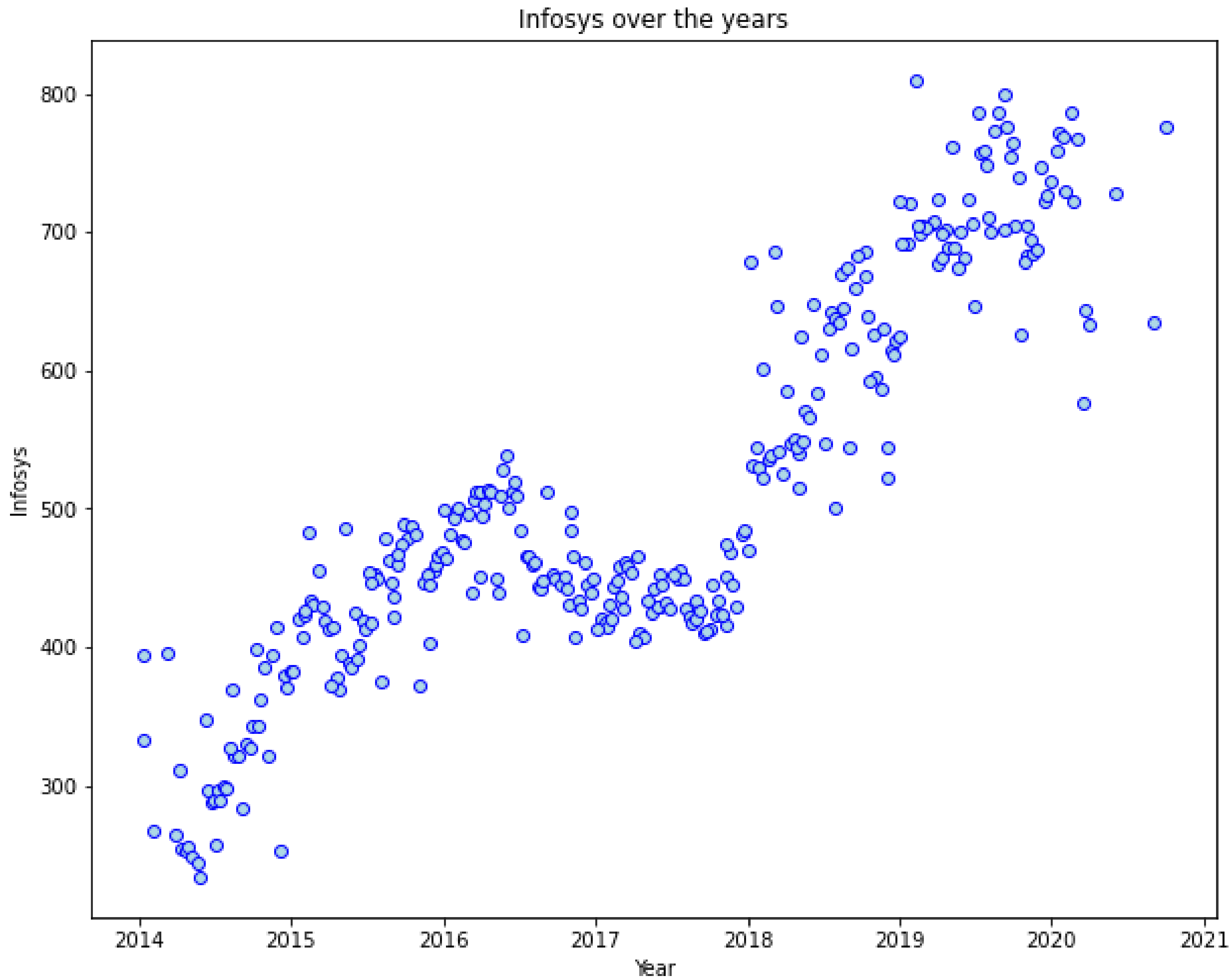


Figure 13: Infosys Stock price Graph over time

- The statistical summary of the Infosys stock for the period 31 Mar 2014 to 30 Mar 2021

mean	511.340764
std	135.952051
min	**234.000000**
25%	424.000000
50%	466.500000
75%	630.750000
max	**810.000000**

- Stock prices of Infosys are clearly rising over the years from 2014 to 2021

- Stock Price vs Time for Sail Stocks over the years

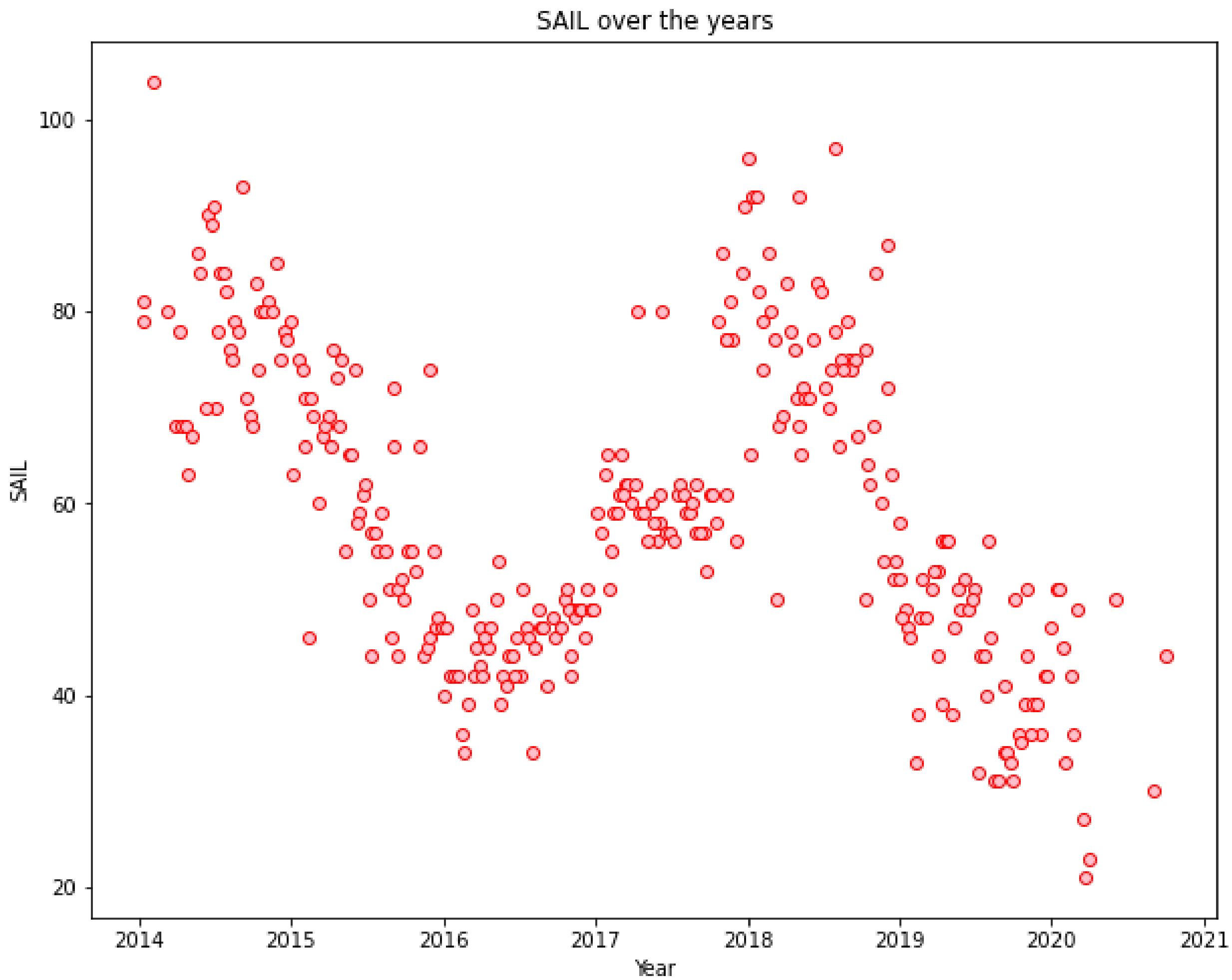


Figure 14: Sail Stock Price over time

- The statistical summary of the Sail stock for the period 31 Mar 2014 to 30 Mar 2021

mean	59.095541
std	15.810493
min	**21.000000**
25%	47.000000
50%	57.000000
75%	71.750000
max	**104.000000**

- Stock price of SAIL declined from 2014 to mid-2016

- They showed an upward trend from mid-2016 to 2019

- From 2019 the prices have been declining

9 Calculate Returns for all stocks with inference

Table 12: Weekly Stock Returns

	0	1	2	3	4	5	6	7	8	9	...	304	305	306
infosys	NaN	-0.026873	-0.011742	-0.003945	0.011788	-0.031749	0.019961	-0.036221	-0.041847	0.135666	...	-0.003894	-0.002604	0.011666
indian_hotel	NaN	-0.014599	0.000000	0.000000	-0.045120	-0.015504	0.060625	0.199333	-0.012121	0.081917	...	-0.042560	0.007220	-0.044125
mahindra_&_mahindra	NaN	0.006572	-0.008772	0.072218	-0.012371	0.040656	0.011881	0.038615	0.064183	-0.003559	...	-0.039716	0.043250	-0.084609
axis_bank	NaN	0.048247	-0.021979	0.047025	-0.003540	0.061875	0.076961	0.059898	-0.014642	0.071154	...	-0.044390	0.059205	-0.014815
sail	NaN	0.028988	-0.028988	0.000000	-0.076373	0.061558	0.112795	0.136859	-0.023530	0.213574	...	-0.125163	0.085158	-0.107631
shree_cement	NaN	0.032831	-0.013888	0.007583	-0.019515	0.011400	0.067622	0.056790	0.048090	0.105167	...	-0.031539	0.105826	-0.019663
sun_pharma	NaN	0.094491	-0.004930	-0.004955	0.011523	-0.008217	-0.016639	-0.049881	0.044835	-0.018724	...	-0.057820	0.018868	-0.028438
jindal_steel	NaN	-0.065882	0.000000	-0.018084	-0.140857	0.024898	0.097543	0.105732	-0.010084	0.132686	...	-0.123753	0.170273	-0.035994
idea_vodafone	NaN	0.011976	-0.011976	0.000000	-0.049393	0.012579	0.048790	-0.024098	-0.012270	0.024391	...	-0.182322	0.000000	-0.510826
jet_airways	NaN	0.086112	-0.078943	0.007117	-0.148846	-0.016598	0.020705	0.169258	-0.181630	0.072031	...	-0.223144	-0.036368	0.036368

10 rows × 314 columns

- The above dataframe shows the weekly returns of each stock for a total 314 weeks.

- The first week (column indexed 0) shows Nan values because that is the beginning week and does not have a reference of the previous weeks data

- Statistical Summary of the Stock Returns

Table 13: Summary of weekly Stock Returns

	infosys	indian_hotel	mahindra_&_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

- Shree Cement @ 0.368 % mean weekly return is the best performer in the period 2014 to 2021.

10 Calculate Stock Means and Standard Deviation for all stocks with inference

- Dataframe showing avg weekly returns and volatility

Table 14: Dataframe showing Average weekly returns and 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

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

- Stock Means vs Standard Deviation

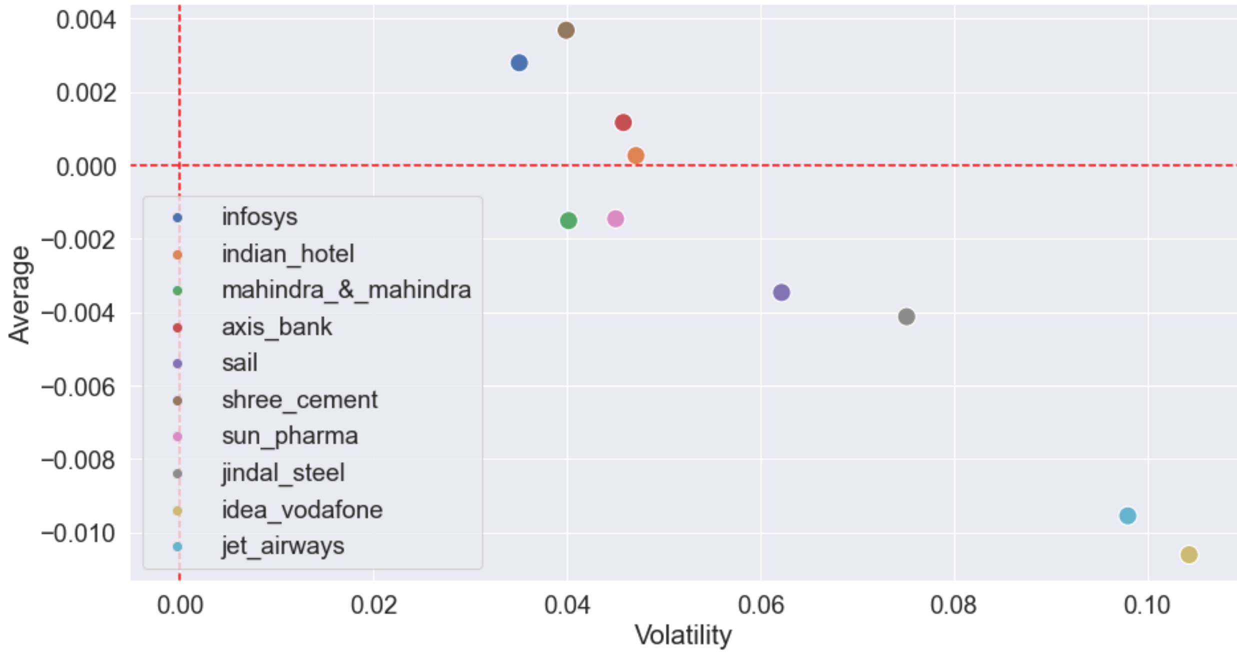


Figure 15: Average Weekly Returns vs Volatility

12 Conclusion and Recommendations

- The red doted line above represents the zero average returns and zero volatility along the respective axis.
- All points above the horizontal red dotted line are giving positive average weekly returns for the aforementioned period
- Shree Cement has been giving the highest weekly returns
- Infosys has the least volatility
- Idea Vodafaone has been the biggest looser and has been the most volatile as well
- Shares to invest in looking at the past data
    - Shree Cements
    - Infosys
    - Axis Bank

End