1.Problem Statement:

You are requested to create an Indian credit risk(default) model, using the data provided in the spreadsheet.
Hints:
Dependent variable - We need to create a default variable which should take the value of 1 when net worth next year is negative & 0 when net worth next year is positive.
Validation Dataset - We need to build the model on train dataset and check the model performance measures on validation dataset.

2.EXPLORATORY DATA ANALYSIS (EDA)

Data has 51 variables and 4256 rows. There are 1 integer type,50 float type data point. Here is a snapshot of the data.

	Num	Networth Next Year	Total assets	Net worth	Total income	Change in stock	Total expenses	Profit after tax	PBDITA	PBT	...	Debtors turnover	Finished goods turnover	WIP turnover	Raw material turnover	Shares outstanding	Equity face value	EPS	Adjust E
0	1	395.3	827.6	336.5	534.1	13.5	508.7	38.9	124.4	64.6	...	5.65	3.99	3.37	14.87	8760056.0	10.0	4.44	4
1	2	36.2	67.7	24.3	137.9	-3.7	131.0	3.2	5.5	1.0	...	NaN	NaN	NaN	NaN	NaN	NaN	0.00	0
2	3	84.0	238.4	78.9	331.2	-18.1	309.2	3.9	25.8	10.5	...	2.51	17.67	8.76	8.35	NaN	NaN	0.00	0
3	4	2041.4	6883.5	1443.3	8448.5	212.2	8482.4	178.3	418.4	185.1	...	1.91	18.14	18.62	11.11	10000000.0	10.0	17.60	17
4	5	41.8	90.9	47.0	388.6	3.4	392.7	-0.7	7.2	-0.6	...	68.00	45.87	28.67	19.93	107315.0	100.0	-6.52	-6

There is no duplicate data in the dataset.

	count	mean	std	min	25%	50%	75%	max
Num	4256.0	2.128500e+03	1.228746e+03	1.000000e+00	1064.750	2128.500	3.192250e+03	4.256000e+03
Networth_Next_Year	4256.0	1.344741e+03	1.593674e+04	-7.426560e+04	3.975	72.100	3.308250e+02	8.057734e+05
Total_assets	4256.0	3.573617e+03	3.007444e+04	1.000000e-01	91.300	315.500	1.120800e+03	1.176509e+06
Net_worth	4256.0	1.351950e+03	1.296131e+04	0.000000e+00	31.475	104.800	3.898500e+02	6.131516e+05
Total_income	4025.0	4.688190e+03	5.391895e+04	0.000000e+00	107.100	455.100	1.485000e+03	2.442828e+06
Change_in_stock	3706.0	4.370248e+01	4.369150e+02	-3.029400e+03	-1.800	1.600	1.840000e+01	1.418550e+04
Total_expenses	4091.0	4.356301e+03	5.139809e+04	-1.000000e-01	96.800	426.800	1.395700e+03	2.366035e+06
Profit_after_tax	4102.0	2.950506e+02	3.079902e+03	-3.908300e+03	0.500	9.000	5.330000e+01	1.194391e+05
PBDITA	4102.0	6.059406e+02	5.646231e+03	-4.407000e+02	6.925	36.900	1.587000e+02	2.085765e+05
PBT	4102.0	4.102590e+02	4.217415e+03	-3.894800e+03	0.800	12.600	7.417500e+01	1.452926e+05
Cash_profit	4102.0	4.082675e+02	4.143926e+03	-2.245700e+03	2.900	19.400	9.625000e+01	1.769118e+05
PBDITA_as_%_of_total_income	4177.0	3.179892e+00	1.722566e+02	-6.400000e+03	4.970	9.680	1.647000e+01	1.000000e+02
PBT_as_%_of_total_income	4177.0	-1.819683e+01	4.199111e+02	-2.134000e+04	0.560	3.340	8.940000e+00	1.000000e+02
PAT_as_%_of_total_income	4177.0	-2.003367e+01	4.235762e+02	-2.134000e+04	0.350	2.370	6.420000e+00	1.500000e+02
Cash_profit_as_%_of_total_income	4177.0	-9.021278e+00	2.999574e+02	-1.502000e+04	2.000	5.660	1.073000e+01	1.000000e+02
PAT_as_%_of_net_worth	4256.0	1.016786e+01	6.153240e+01	-7.487200e+02	0.000	8.040	2.020250e+01	2.466670e+03
Sales	3951.0	4.645685e+03	5.308090e+04	1.000000e-01	113.350	468.600	1.481200e+03	2.384984e+06
Income_from_fincial_services	3145.0	8.136006e+01	1.042759e+03	0.000000e+00	0.500	1.900	9.800000e+00	5.193820e+04
Other_income	2700.0	5.595289e+01	1.178415e+03	0.000000e+00	0.400	1.500	6.200000e+00	4.285670e+04
Total_capital	4251.0	2.245577e+02	1.684951e+03	1.000000e-01	13.200	42.600	1.031500e+02	7.827320e+04
Reserves_and_funds	4158.0	1.210562e+03	1.281623e+04	-6.525900e+03	5.300	55.150	2.825250e+02	6.251378e+05

Here is a snapshot of data description.

As per the instruction we have created a new ‘default’ column. If next year net worth is -ve then we mark as default and encode it 1 and if not then we mark as not default and encode it 0.

As we can see from the snapshot data has few missing values and data also have outliers.

Num	0
Networth_Next_Year	0
Total_assets	0
Net_worth	0
Total_income	231
Change_in_stock	550
Total_expenses	165
Profit_after_tax	154
PBDITA	154
PBT	154
Cash_profit	154
PBDITA_as_%_of_total_income	79
PBT_as_%_of_total_income	79
PAT_as_%_of_total_income	79
Cash_profit_as_%_of_total_income	79
PAT_as_%_of_net_worth	0
Sales	305
Income_from_fincial_services	1111
Other_income	1556
Total_capital	5
Reserves_and_funds	98
Borrowings	431
Current_liabilities_&_provisions	110
Deferred_tax_liability	1369
Shareholders_funds	0
Cumulative_retained_profits	45
Capital_employed	0
TOL/TNW	0
Total_term_liabilities_/_tangible_net_worth	0
Contingent_liabilities_/_Net_worth_(%)	0
Contingent_liabilities	1402
Net_fixed_assets	132
Investments	1715
Current_assets	80
Net_working_capital	37
Quick_ratio_(times)	105
Current_ratio_(times)	105
Debt_to_equity_ratio_(times)	0
Cash_to_current_liabilities_(times)	105
Cash_to_average_cost_of_sales_per_day	100
Creditors_turnover	391
Debtors_turnover	385
Finished_goods_turnover	874
WIP_turnover	764
Raw_material_turnover	428
Shares_outstanding	810
Equity_face_value	810

```
Default
0    4022
1     234
Name: count, dtype: int64
```

#	Column	Non-Null Count	Dtype
0	Num	4256 non-null	int64
1	Networth_Next_Year	4256 non-null	float64
2	Total_assets	4256 non-null	float64
3	Net_worth	4256 non-null	float64
4	Total_income	4025 non-null	float64
5	Change_in_stock	3706 non-null	float64
6	Total_expenses	4091 non-null	float64
7	Profit_after_tax	4102 non-null	float64
8	PBDITA	4102 non-null	float64
9	PBT	4102 non-null	float64
10	Cash_profit	4102 non-null	float64
11	PBDITA_as_%_of_total_income	4177 non-null	float64
12	PBT_as_%_of_total_income	4177 non-null	float64
13	PAT_as_%_of_total_income	4177 non-null	float64
14	Cash_profit_as_%_of_total_income	4177 non-null	float64
15	PAT_as_%_of_net_worth	4256 non-null	float64
16	Sales	3951 non-null	float64
17	Income_from_fincial_services	3145 non-null	float64
18	Other_income	2700 non-null	float64
19	Total_capital	4251 non-null	float64
20	Reserves_and_funds	4158 non-null	float64
21	Borrowings	3825 non-null	float64
22	Current_liabilities_&_provisions	4146 non-null	float64
23	Deferred_tax_liability	2887 non-null	float64
24	Shareholders_funds	4256 non-null	float64
25	Cumulative_retained_profits	4211 non-null	float64
26	Capital_employed	4256 non-null	float64
27	TOL/TNW	4256 non-null	float64
28	Total_term_liabilities_/_tangible_net_worth	4256 non-null	float64
29	Contingent_liabilities_/_Net_worth_(%)	4256 non-null	float64
30	Contingent_liabilities	2854 non-null	float64
31	Net_fixed_assets	4124 non-null	float64
32	Investments	2541 non-null	float64
33	Current_assets	4176 non-null	float64
34	Net_working_capital	4219 non-null	float64
35	Quick_ratio_(times)	4151 non-null	float64
36	Current_ratio_(times)	4151 non-null	float64
37	Debt_to_equity_ratio_(times)	4256 non-null	float64
38	Cash_to_current_liabilities_(times)	4151 non-null	float64
39	Cash_to_average_cost_of_sales_per_day	4156 non-null	float64
40	Creditors_turnover	3865 non-null	float64
41	Debtors_turnover	3871 non-null	float64
42	Finished_goods_turnover	3382 non-null	float64
43	WIP_turnover	3492 non-null	float64
44	Raw_material_turnover	3828 non-null	float64

We check for missing values and find that many columns have missing values. Here is a snippet of the column info.

We see that ‘PE_on_BSE’ has 50% rows missing . So we remove that column to make sure that it doesn’t degrade the model. We also removed ‘num’ column as it is just a index column.

We also check for weights of the target variable in the data. We find that data is imbalanced.

3.Missing value treatment

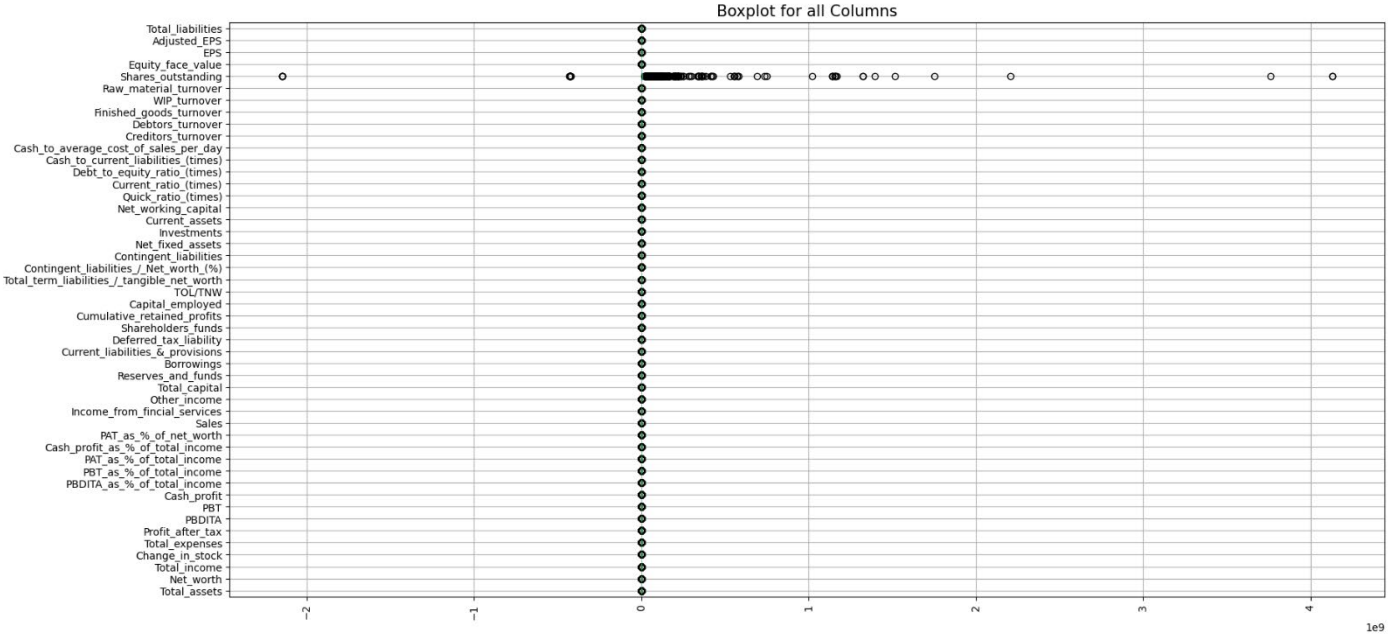
We then split the data in target and predictor variables.

We then use KNN-imputer with n_neighbors=5 and impute missing values, here is a snapshot of the data after the treatment.

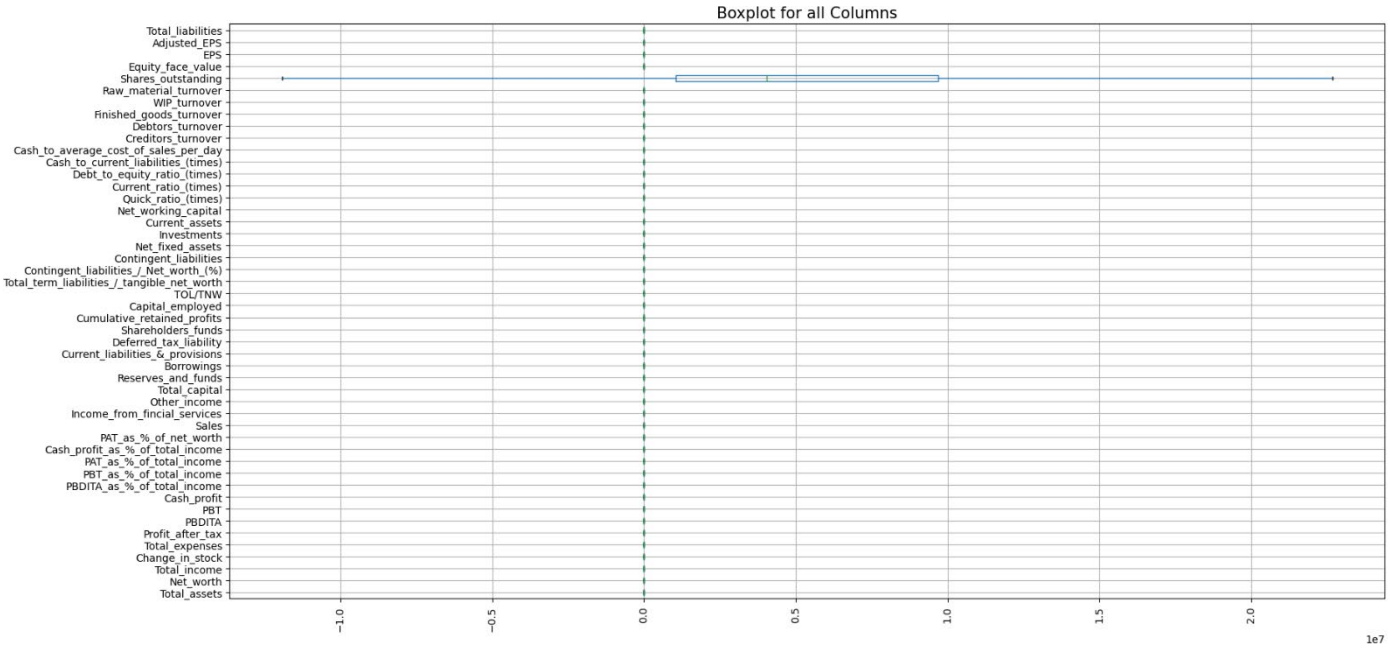
Total_assets	0
Net_worth	0
Total_income	0
Change_in_stock	0
Total_expenses	0
Profit_after_tax	0
PBDITA	0
PBT	0
Cash_profit	0
PBDITA_as_%_of_total_income	0
PBT_as_%_of_total_income	0
PAT_as_%_of_total_income	0
Cash_profit_as_%_of_total_income	0
PAT_as_%_of_net_worth	0
Sales	0
Income_from_fincial_services	0
Other_income	0
Total_capital	0
Reserves_and_funds	0
Borrowings	0
Current_liabilities_&_provisions	0
Deferred_tax_liability	0
Shareholders_funds	0
Cumulative_retained_profits	0
Capital_employed	0
TOL/TNW	0
Total_term_liabilities_/_tangible_net_worth	0
Contingent_liabilities_/_Net_worth_(%)	0
Contingent_liabilities	0
Net_fixed_assets	0
Investments	0
Current_assets	0
Net_working_capital	0
Quick_ratio_(times)	0
Current_ratio_(times)	0
Debt to equity ratio (times)	0

After this there is no missing value present in the dataset.

4.Outlier Treatment

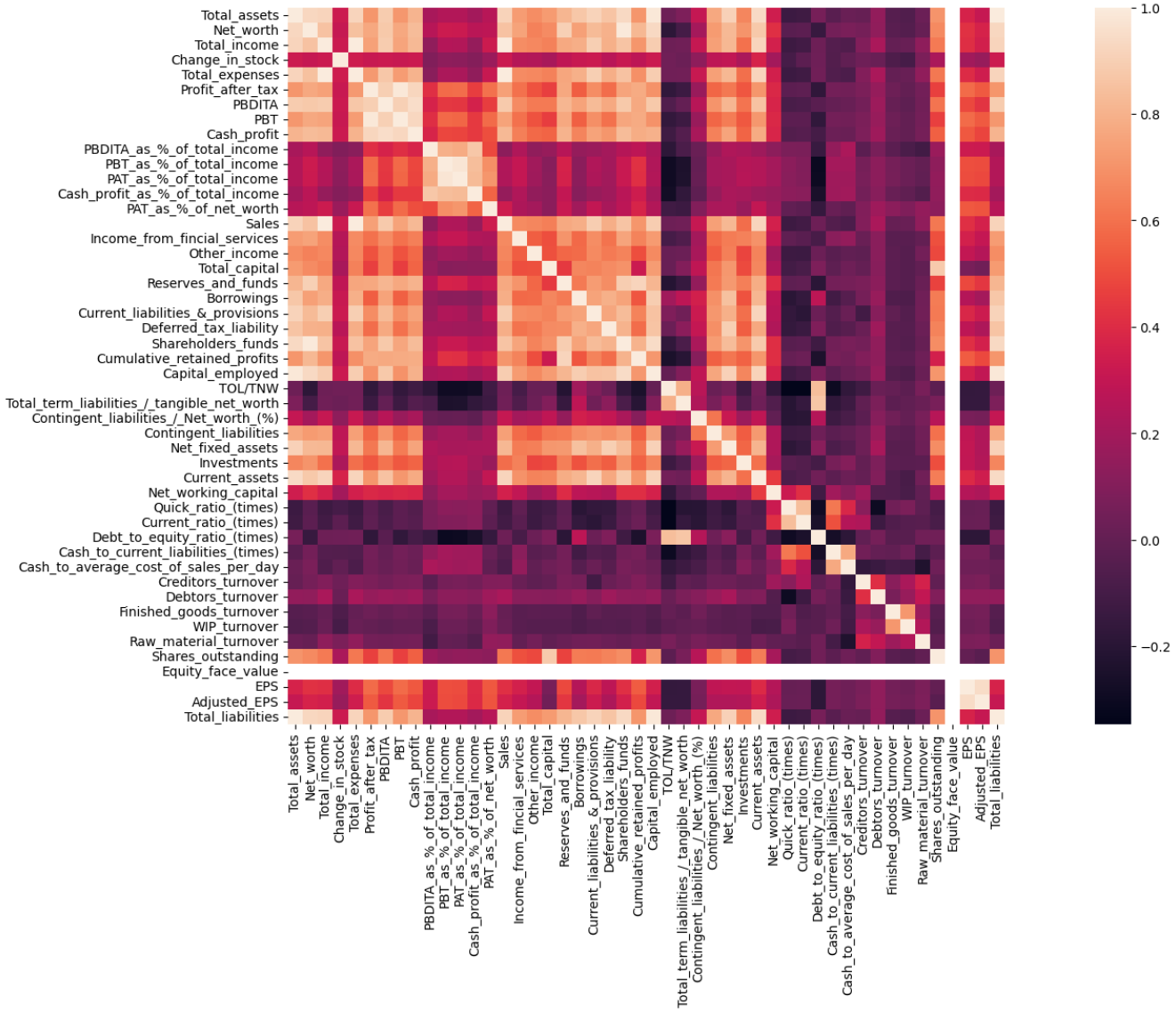


As we can see from the snippet that all the columns have outliers. We will impute these with upper bound or lower bound.



Here is the data after treating ourliers. We can see that now the data has no outliers.

5.Checking Correlation amongst the variables using heatmap:



As we can see from the heatmap that we have a few columns which are correlated. We will remove those in further workflow.

6.Train Test Split

As we have previously split the data into predictor and target variables. We will now scale the data using standardscaler library.

	Total_assets	Net_worth	Total_income	Change_in_stock	Total_expenses	Profit_after_tax	PBDITA	PBT	Cash_profit	PBDITA_as_%_of_total_income	...	Creditor
0	0.058649	0.217928	-0.362212	0.411188	-0.367228	0.159964	0.192649	0.310311	0.436732	1.220551	...	
1	-0.763498	-0.746209	-0.703303	-0.581727	-0.708342	-0.550989	-0.713776	-0.614255	-0.647458	-0.690043	...	
2	-0.578816	-0.577593	-0.536890	-1.413004	-0.547403	-0.537049	-0.559021	-0.476152	-0.581031	-0.313864	...	
3	2.046613	2.042786	2.094487	1.829122	2.088184	1.925398	2.025511	1.923395	1.418909	-0.595008	...	
4	-0.738398	-0.676106	-0.487474	-0.171861	-0.471992	-0.628656	-0.700816	-0.637515	-0.646272	-0.901891	...	

Here is the snapshot of the data after scaling.

Now we split the data into train and test data with test data size consisting of 33% of whole data. We will also stratify the data so that imbalances in the target variable can be captured in train and test data.

7.Build Logistic Regression Model (using statsmodels library) on most important variables on train dataset and choose the optimum cut-off. Also showcase your model building approach.

We have to work on feature selection and remove values which are correlated. To ensure there is no multicollinearity in the data we use VIF and we eliminate any variable which have VIF of more than 5.

We drop each column with VIF >=5 one by one and measure VIF at the end of removal.

	variables	VIF
0	Total_assets	inf
47	Total_liabilities	inf
2	Total_income	350.152085
4	Total_expenses	331.201488
14	Sales	242.360115
22	Shareholders_funds	124.888253
1	Net_worth	113.888996
24	Capital_employed	93.527403
7	PBT	52.619278
5	Profit_after_tax	50.499478
6	PBDITA	31.872569
10	PBT_as_%_of_total_income	28.437633
11	PAT_as_%_of_total_income	27.390223
8	Cash_profit	22.417658
31	Current_assets	20.788353
20	Current_liabilities_&_provisions	16.464904
18	Reserves_and_funds	13.318399
29	Net_fixed_assets	11.325544
45	EPS	11.226386
46	Adjusted_EPS	10.119625
23	Cumulative_retained_profits	9.086896
19	Borrowings	8.260666
12	Cash_profit_as_%_of_total_income	7.667169
35	Debt_to_equity_ratio_(times)	6.638005
17	Total_capital	6.406083

	variables	VIF
1	Total_income	350.152085
3	Total_expenses	331.201488
13	Sales	242.360115
21	Shareholders_funds	124.888253
46	Total_liabilities	115.538699
0	Net_worth	113.888996
23	Capital_employed	93.527403
6	PBT	52.619278
4	Profit_after_tax	50.499478
5	PBDITA	31.872569
9	PBT_as_%_of_total_income	28.437633
10	PAT_as_%_of_total_income	27.390223
7	Cash_profit	22.417658
30	Current_assets	20.788353
19	Current_liabilities_&_provisions	16.464904
17	Reserves_and_funds	13.318399
28	Net_fixed_assets	11.325544
44	EPS	11.226386
45	Adjusted_EPS	10.119625
22	Cumulative_retained_profits	9.086896
18	Borrowings	8.260666
11	Cash_profit_as_%_of_total_income	7.667169
34	Debt_to_equity_ratio_(times)	6.638005
16	Total_capital	6.406083
8	PBDITA_as_%_of_total_income	6.285787
32	Quick_ratio_(times)	6.030624

Here is the snapshot of the VIF chart after removing ‘Total_assets’ which had the highest VIF. We do this process until we find that our VIF chart doesn’t have values more than 5.

	count	mean	std	min	25%	50%	75%	max
Change_in_stock	2851.0	0.005052	0.985248	-1.778853	-0.420090	-0.315025	0.477575	1.829122
Profit_after_tax	2851.0	-0.016632	0.986779	-2.129225	-0.608741	-0.457390	0.342183	1.925398
PBDITA_as_%_of_total_income	2851.0	0.001735	0.997835	-2.339291	-0.611837	-0.128744	0.517690	2.269892
PAT_as_%_of_total_income	2851.0	-0.003810	0.990568	-1.999181	-0.486150	-0.150104	0.512054	2.039985
PAT_as_%_of_net_worth	2851.0	0.005467	0.987614	-2.252236	-0.583539	-0.142463	0.507036	2.197622
Income_from_fincial_services	2851.0	-0.029650	0.983412	-0.756316	-0.706058	-0.578740	0.265583	1.974331
Other_income	2851.0	-0.013118	0.997666	-0.833464	-0.737672	-0.514155	0.347978	2.056280
Borrowings	2851.0	-0.021000	0.989186	-0.801447	-0.741814	-0.505241	0.289425	2.047759
Deferred_tax_liability	2851.0	-0.021759	0.982348	-0.780956	-0.722012	-0.521474	0.316237	2.042914
Cumulative_retained_profits	2851.0	-0.006473	0.987739	-2.121331	-0.608586	-0.432646	0.378272	1.911327
TOL/TNW	2851.0	-0.005349	1.000498	-2.530016	-0.758398	-0.316159	0.461755	2.222717
Total_term_liabilities_/_tangible_net_worth	2851.0	-0.001802	1.003426	-2.591981	-0.780000	-0.411245	0.427988	2.239970
Contingent_liabilities_/_Net_worth_(%)	2851.0	0.000440	1.003692	-0.741933	-0.741933	-0.550088	0.433766	2.151825
Contingent_liabilities	2851.0	-0.018200	0.987012	-0.726953	-0.705017	-0.569955	0.302993	1.980107
Investments	2851.0	-0.014074	0.987667	-0.761282	-0.707124	-0.555480	0.280365	1.918477
Net_working_capital	2851.0	-0.004946	0.991388	-1.882616	-0.478239	-0.288970	0.442690	1.859505
Current_ratio_(times)	2851.0	0.003246	1.002094	-1.865603	-0.623374	-0.226918	0.447057	2.072527
Cash_to_current_liabilities_(times)	2851.0	-0.004437	0.995355	-0.910936	-0.717817	-0.460324	0.376527	2.018042
Cash_to_average_cost_of_sales_per_day	2851.0	-0.003187	0.995362	-0.895464	-0.740321	-0.455803	0.370105	2.025006
Creditors_turnover	2851.0	-0.011229	1.000786	-1.199194	-0.693675	-0.358032	0.394082	2.090447
Debtors_turnover	2851.0	-0.017902	0.996309	-1.228741	-0.703560	-0.330255	0.392225	2.137566
Finished_goods_turnover	2851.0	-0.012829	0.998620	-1.022621	-0.741709	-0.444631	0.370393	2.099040
WIP_turnover	2851.0	-0.002677	1.005619	-1.134982	-0.740099	-0.372530	0.403798	2.153183
Raw_material_turnover	2851.0	-0.016556	0.991229	-1.417751	-0.749837	-0.288787	0.445378	2.244960
Shares_outstanding	2851.0	-0.020708	0.982150	-2.460832	-0.767624	-0.378747	0.311741	2.073442

Here is the data description after removing columns which had VIF of more than 5.

We then join the target and predictor variables as statsmodel requires that.

We now create model with these variables.

Model 1:

Here is the equation we used for model 1

f_1= 'Default ~ Change_in_stock + Profit_after_tax + PBDITA_as_%_of_total_income + PAT_as_%_of_total_income + PAT_as_%_of_net_worth + Income_from_fincial_services + Other_income + Borrowings + Deferred_tax_liability + Cumulative_retained_profits + TOL/TNW + Total_term_liabilities_/_tangible_net_worth + Contingent_liabilities_/_Net_worth_(%) + Contingent_liabilities + Investments + Net_working_capital + Current_ratio_(times) + Cash_to_current_liabilities_(times) + Cash_to_average_cost_of_sales_per_day + Creditors_turnover + Debtors_turnover + Finished_goods_turnover + WIP_turnover + Raw_material_turnover + Shares_outstanding + Equity_face_value + Adjusted_EPS '

Snapshot of model summary

Logit Regression Results

Dep. Variable:	Default	**No. Observations:**	2851
Model:	Logit	**Df Residuals:**	2824
Method:	MLE	**Df Model:**	26
Date:	Mon, 18 Dec 2023	**Pseudo R-squ.:**	0.4210
Time:	01:49:07	**Log-Likelihood:**	-351.89
converged:	True	**LL-Null:**	-607.77
Covariance Type:	nonrobust	**LLR p-value:**	1.294e-91

	coef	std err	z	P>|z|	[0.025	0.975]
Intercept	-5.3361	0.343	-15.571	0.000	-6.008	-4.664
Change_in_stock	0.3530	0.157	2.249	0.024	0.045	0.661
Profit_after_tax	-0.2866	0.273	-1.048	0.295	-0.822	0.249
Q("PBDITA_as_%_of_total_income")	-0.1559	0.116	-1.343	0.179	-0.383	0.072
Q("PAT_as_%_of_total_income")	-0.2049	0.180	-1.137	0.255	-0.558	0.148
Q("PAT_as_%_of_net_worth")	-0.3488	0.152	-2.298	0.022	-0.646	-0.051
Income_from_fincial_services	0.0779	0.214	0.364	0.716	-0.342	0.497
Other_income	0.2508	0.175	1.429	0.153	-0.093	0.595
Borrowings	-0.0565	0.301	-0.188	0.851	-0.647	0.533
Deferred_tax_liability	-0.2321	0.293	-0.791	0.429	-0.807	0.343
Cumulative_retained_profits	-1.1502	0.320	-3.598	0.000	-1.777	-0.524
Q("TOL/TNW")	0.7807	0.139	5.604	0.000	0.508	1.054
Q("Total_term_liabilities_/_tangible_net_worth")	-0.1639	0.135	-1.211	0.226	-0.429	0.101
Q("Contingent_liabilities_/_Net_worth_(%)")	0.1985	0.111	1.790	0.073	-0.019	0.416
Contingent_liabilities	-0.5355	0.246	-2.181	0.029	-1.017	-0.054
Investments	-0.1661	0.197	-0.843	0.399	-0.552	0.220
Net_working_capital	-0.1234	0.173	-0.713	0.476	-0.463	0.216

We can see from the snapshot that model 1 still has some columns in which p value is not significant. We remove those at once to build a better model with less predictors.

Model 2:

Here is the equation we used for model 2

f_2= 'Default ~ Q("PAT_as_%_of_net_worth")+ Cumulative_retained_profits + Q("TOL/TNW")+ Contingent_liabilities + Q("Cash_to_current_liabilities_(times)") + Adjusted_EPS'

Logit Regression Results

Dep. Variable:	Default	**No. Observations:**	2851
Model:	Logit	**Df Residuals:**	2844
Method:	MLE	**Df Model:**	6
Date:	Mon, 18 Dec 2023	**Pseudo R-squ.:**	0.3790
Time:	01:49:07	**Log-Likelihood:**	-377.40
converged:	True	**LL-Null:**	-607.77
Covariance Type:	nonrobust	**LLR p-value:**	2.411e-96

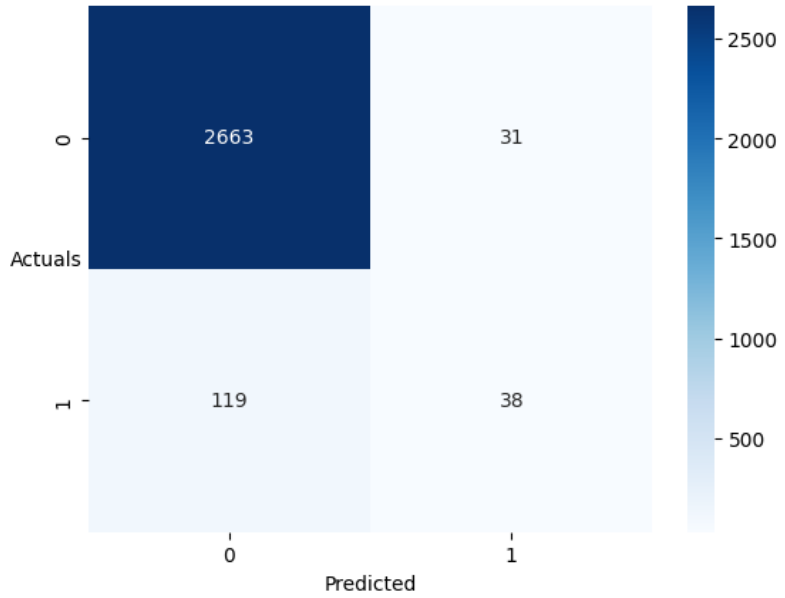
	coef	std err	z	P>|z|	[0.025	0.975]
Intercept	-4.8491	0.264	-18.359	0.000	-5.367	-4.331
Q("PAT_as_%_of_net_worth")	-0.6332	0.113	-5.628	0.000	-0.854	-0.413
Cumulative_retained_profits	-1.1074	0.226	-4.897	0.000	-1.551	-0.664
Q("TOL/TNW")	0.7264	0.085	8.513	0.000	0.559	0.894
Contingent_liabilities	-0.5224	0.157	-3.320	0.001	-0.831	-0.214
Q("Cash_to_current_liabilities_(times)")	0.2839	0.101	2.800	0.005	0.085	0.483
Adjusted_EPS	-0.9534	0.183	-5.211	0.000	-1.312	-0.595

Here is our model with all insignificant variables removed.

8.Checking the accuracy of the model using confusion matrix for training set

Here is our classification report and confusion matrix of the train data with 0.5 threesold.

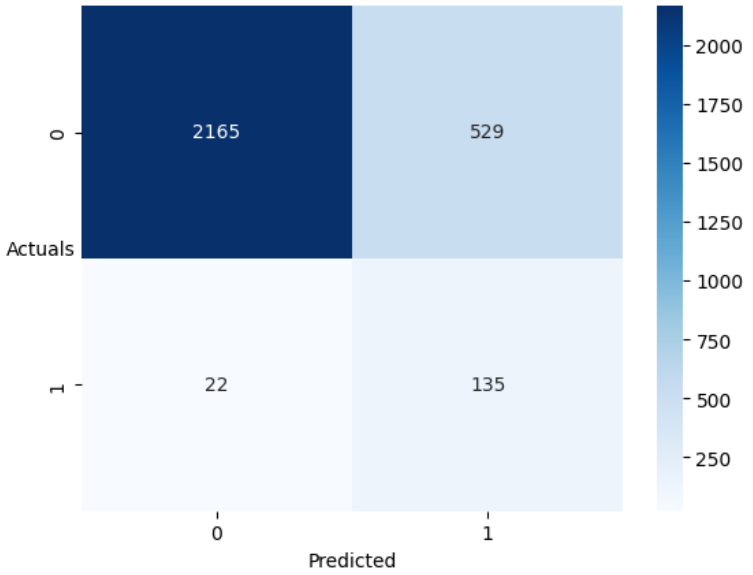
	precision	recall	f1-score	support
0	0.957	0.988	0.973	2694
1	0.551	0.242	0.336	157
accuracy			0.947	2851
macro avg	0.754	0.615	0.654	2851
weighted avg	0.935	0.947	0.938	2851



9.Checking the accuracy of the model using confusion matrix for test set

We checked that our threshold is 0.044222496849801535. We change our threshold and check our model performance on train data again.

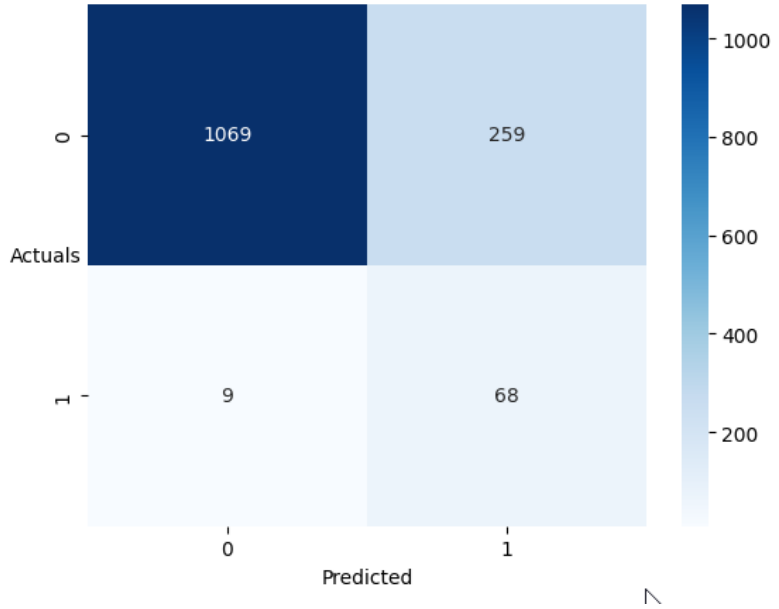
	precision	recall	f1-score	support
0	0.990	0.804	0.887	2694
1	0.203	0.860	0.329	157
accuracy			0.807	2851
macro avg	0.597	0.832	0.608	2851
weighted avg	0.947	0.807	0.856	2851



As we can see that even though precision has decreased, recall has improved greatly.

We will now check our model performance on test data with changed threshold.

	precision	recall	f1-score	support
0	0.992	0.805	0.889	1328
1	0.208	0.883	0.337	77
accuracy			0.809	1405
macro avg	0.600	0.844	0.613	1405
weighted avg	0.949	0.809	0.858	1405



As we can see from the confusion matrix and classification report that the model performance was consistent across train and test data.

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

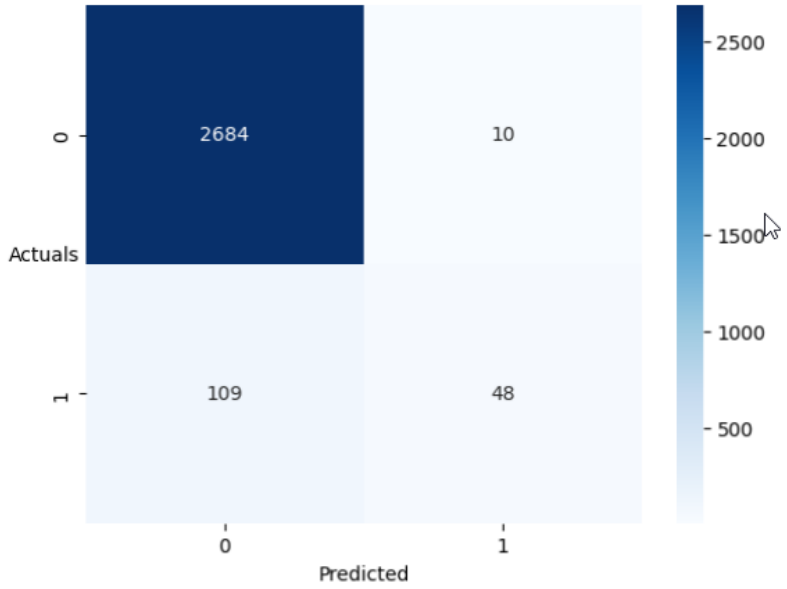
We now build a random forest model with hyper parameter turning. We use Bayesian optimisation to search for best parameter. We will use optuna library for our hyperparameter search.

```
{'n_estimators': 25,
 'max_depth': 5,
 'min_samples_split': 47,
 'min_samples_leaf': 1}
```

This is the best parameter, and we build our model with these hyperparameters.

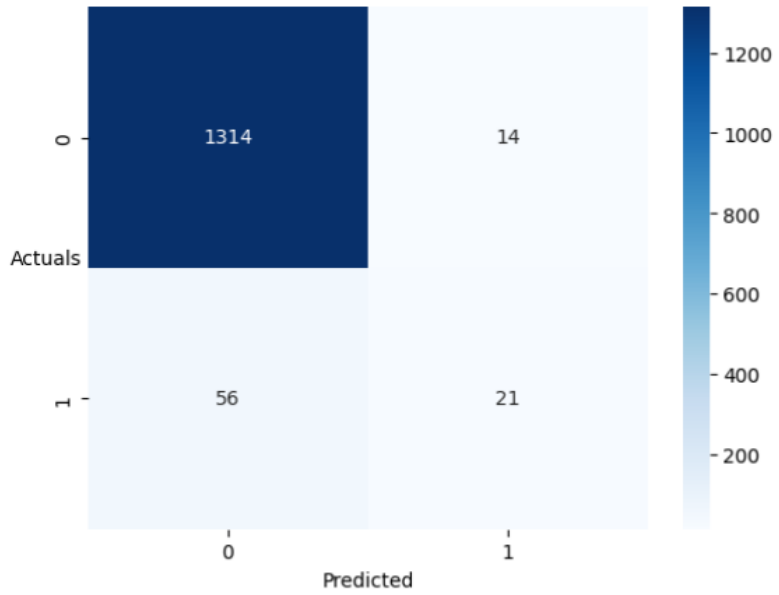
11.Validate the Random Forest Model on test Dataset and state the performance metrics. Also state interpretation from the model

	precision	recall	f1-score	support
0	0.96	1.00	0.98	2694
1	0.83	0.31	0.45	157
accuracy			0.96	2851
macro avg	0.89	0.65	0.71	2851
weighted avg	0.95	0.96	0.95	2851



As we can see our precision is good, but recall has dropped. We check test data performance.

	precision	recall	f1-score	support
0	0.96	0.99	0.97	1328
1	0.60	0.27	0.37	77
accuracy			0.95	1405
macro avg	0.78	0.63	0.67	1405
weighted avg	0.94	0.95	0.94	1405



We can see precision and recall is almost consistent with train data. Recall is very low.

We can see that our model performance has not changed.

We can see that the model performance is not that good, recall is very low. Reason for this could be the imbalance of target variable classes.

We will now try to oversample the data using SMOTE and try doing prediction again.

```
Default
0    2694
1     157
Name: count, dtype: int64
```

This is the target variable class distribution before oversampling.

```
Default
0    2694
1    2020
Name: count, dtype: int64
```

This is the target variable class distribution after oversampling. We can see imbalance has been solved.

Now we will again tune the random forest model with optuna hyperparameter search and check for best hyperparameters.

This are our best hyperparameters-

{'n_estimators': 16,

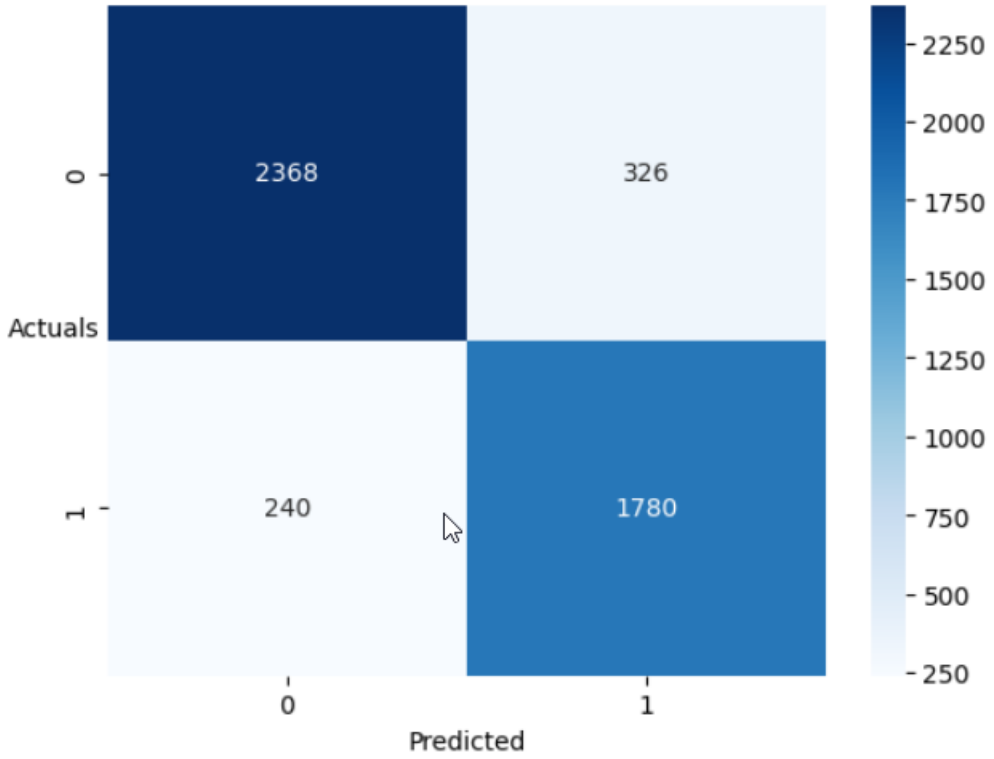
'max_depth': 4,

'min_samples_split': 13,

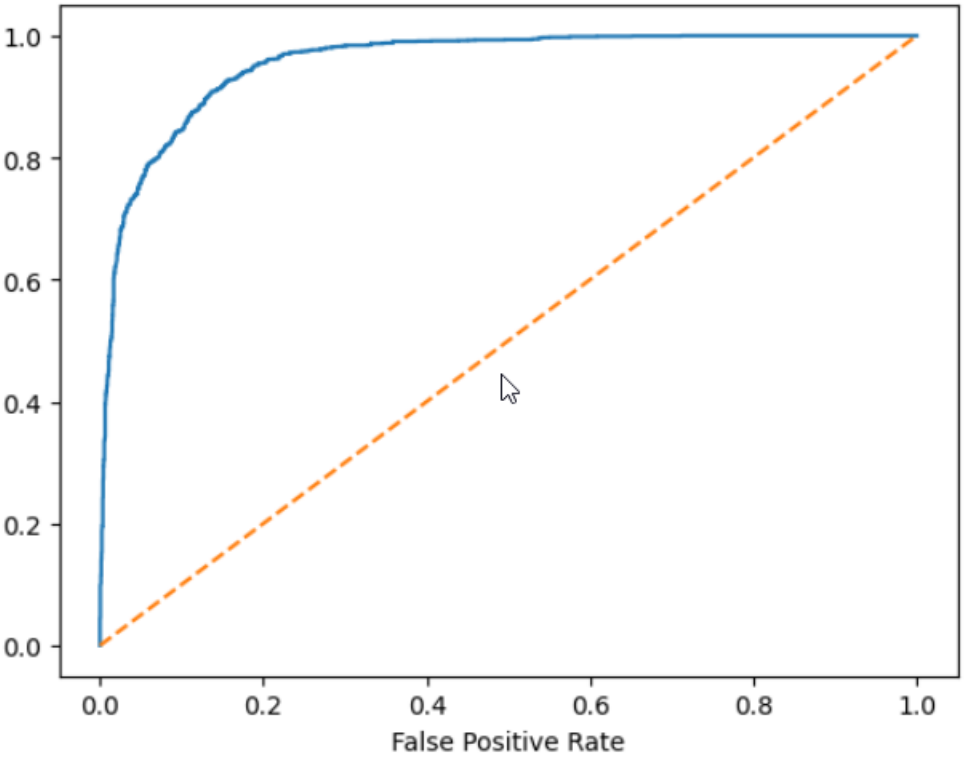
'min_samples_leaf': 66}

We again check for train and test data performance with new model.

	precision	recall	f1-score	support
0	0.91	0.88	0.89	2694
1	0.85	0.88	0.86	2020
accuracy			0.88	4714
macro avg	0.88	0.88	0.88	4714
weighted avg	0.88	0.88	0.88	4714

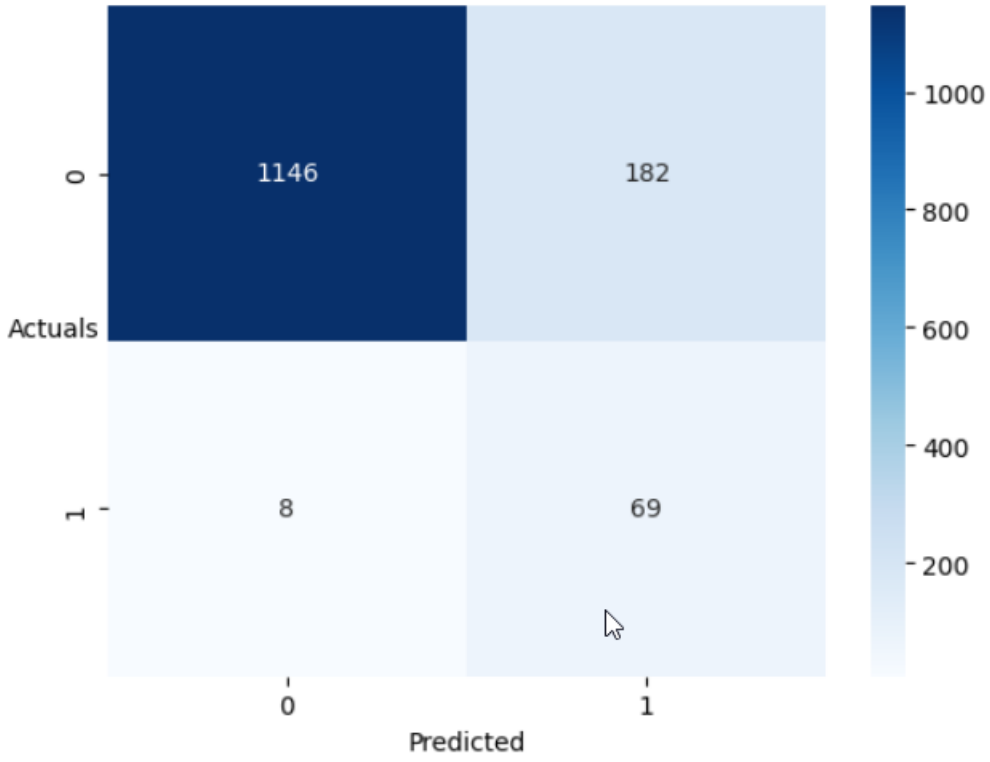


AUC: 0.956

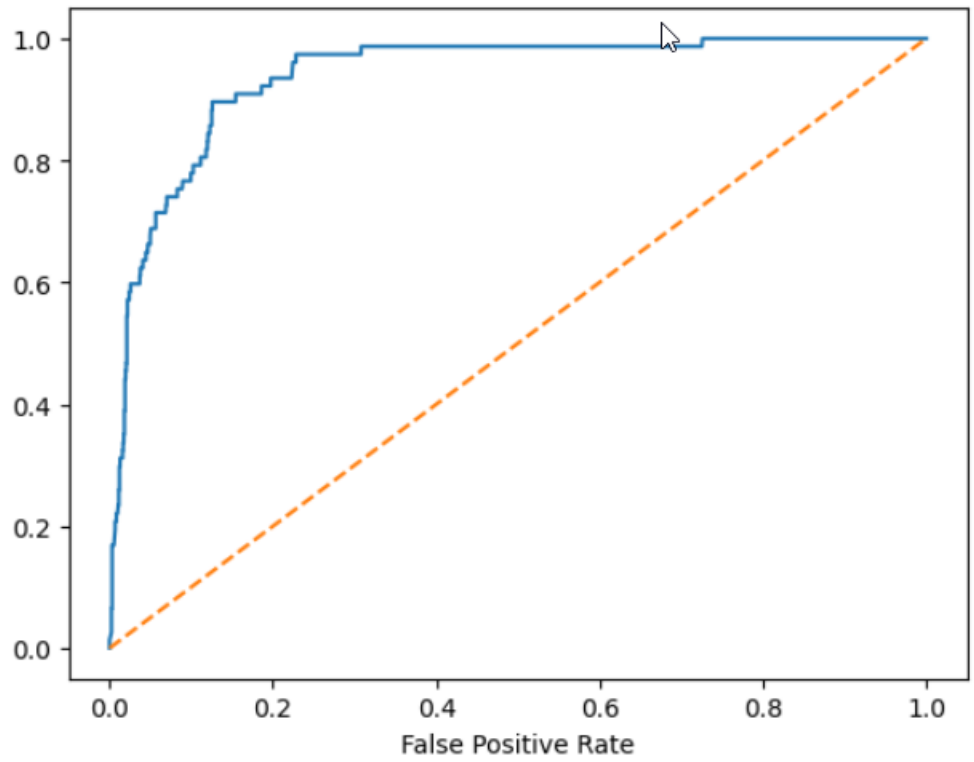


As we can see model performance has improved vastly and our train data recall has improved to 88%. Now we check test performance.

	precision	recall	f1-score	support
0	0.99	0.86	0.92	1328
1	0.27	0.90	0.42	77
accuracy			0.86	1405
macro avg	0.63	0.88	0.67	1405
weighted avg	0.95	0.86	0.90	1405

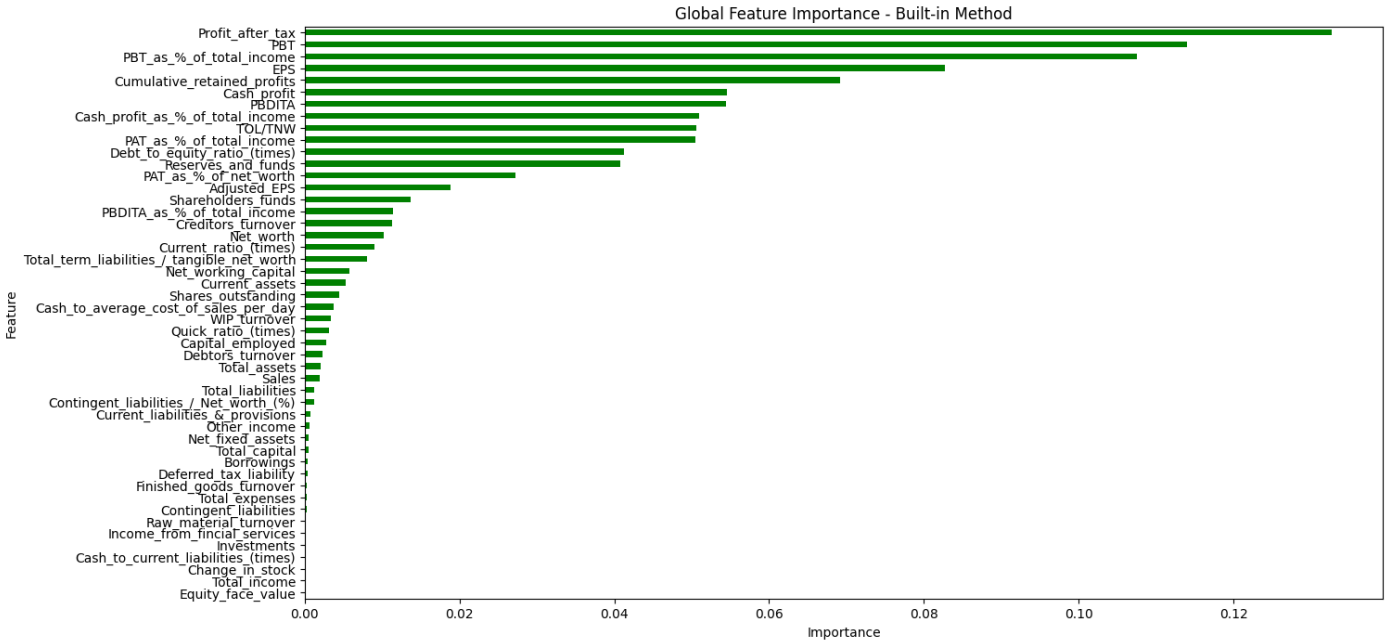


AUC: 0.939



We can see our test data recall has improved to 90% and train and test performances are consistent.

Interpretation from the model



Here is our feature importance chart.

We can see that ‘profit_after_tax’ is our most important variable and ‘equity_face_value’ is our least important variable.

12.Comparison of Random Forest model with logistic regression model:

	Accuracy	Recall	Precision
Logistic Regression	0.947	0.242	0.551
Logistic Regression after cutoff change	0.809	0.883	0.208
Random Forest	0.95	0.27	0.60
Random Forest optimized	0.86	0.90	0.27

We can see that our hyper tuned Random Forest model which was performed on over-sampled data, has performed best and given best recall value.

13.Conclusion and Recommendations: -

Conclusion: -

1. We conclude that data had too many missing values. While we imputed them with KNN imputer but they will never be a perfect reflection of the original data.
2. As the number of default class is always lower than number of non-default classes, our model performs poorly in those cases.
3. We are using SMOTE over-sampling to address class imbalances, but they are generated data and not original data. We should improve our data collection to remedy class imbalances.
4. We use Bayesian hyperparameter search techniques to find our best hyperparameter. They work better after class imbalance was addressed.
5. Our final tuned Random Forest model has recall of 90% which is huge improvement and best among rest of the models.

Recommendations: -

1. Data collection should be improved to address missing values and class imbalances.
2. Our model has a recall of 90% ,so it is a good predictor of default. We should use this model to know who are going to default and improve our business.
3. Profit_after_tax,PBT,PBT_as_%_of_total_income,EPS, Cumulative_retained_profits ,Cash_profit,PBITDA are the most important variables which are most significant part of our model. Data collection team should try to get most accurate data in these parameters.
4. Company should keep these important parameters in notice and warn customers in advance if they see any default probabilities.
5. Company should cut off trade and stop business with the customers whom they think will default according to the model.