Supervised Learning

Classification Project:
AllLife Bank Personal Loan Campaign

July 20, 2023

- Vaibhav Pradhan

Contents / Agenda

- Executive Summary
- Business Problem Overview and Solution Approach
- EDA Results
- Data Preprocessing
- Model Performance Summary
- Appendix

Executive Summary

- ❑ Customer data from AllLife Bank was analysed and Logistic Regression and Decision Tree models were trained and tested to identify the best model to predict factors that will lead customers to opt for personal loans

- ❑ The best performing model was derived from the Decision Tree Modelling technique where the original tree was post pruned via **ccp_alpha=0.0006209286209286216** and gave the below Recall values for test and training data sets:
  - ❑ Recall for best performing model on Train Data = 0.963746
  - ❑ Recall for best performing model on Test Data = 0.90604

- ❑ Decision Tree model indicates that most customers that go for loans are the ones with higher income (> $116.5K)

Customer attributes that drive personal loan purchase decisions.

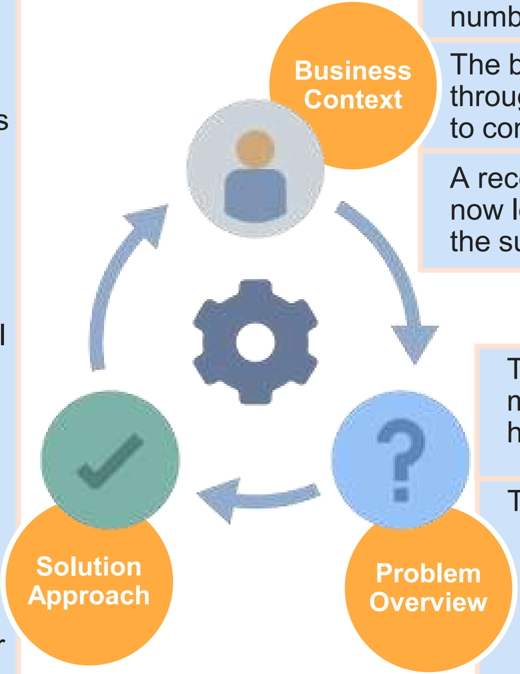
- Education, Income, Family, CCAvg, CD_Account and Age are the important features in predicting the potential loan customers
- From the decision tree model, income is the most important feature
- The higher the income, the more chances the customer will accept a personal loan
- Customers with UnderGraduate Education level are more willing to accept a personal loan than higher levels and is a feature with high importance
- As Family grows, customers are more willing to accept personal loans. As the monthly spending of customers increase, the more they are willing to accept personal loan
- Customers with a CD_Account tend to opt for personal loans
- Although to a less extent, Age also plays a factor in opting for personal loans

Recommendations

- The marketing team is recommended to study the customers profiles first before approaching them for a personal loan offer.
- The top 6 features stated in the features list above need to be considered as the target customer profile for a personal loan campaign.
- Target customers with income > $116.5 K
- Target customers with families > 3
- Target customers with Undergraduate education levels

Business Problem Overview and Solution Approach

- The model will be used to predict whether a liability customer will buy a personal loan, understand which customer attributes are most significant in driving purchases, and identify which segment of customers to target more.
- The model will be built using a supervised learning algorithm, such as logistic regression or decision trees.
- The data used to train the model will include customer demographics, financial information, and past purchase behavior.
- Once the model is trained, it will be used to score potential customers. Customers with a high score will be more likely to purchase a personal loan.
- The model will be a valuable tool for the marketing department, as it will help them to target their marketing campaigns more effectively.



AllLife Bank is a US bank with a growing customer base. Most of the customers are liability customers (depositors), while the number of borrowers is quite small.

The bank wants to expand its loan business and earn more through interest on loans. The management wants to explore ways to convert liability customers to personal loan customers.

A recent campaign showed that this is possible, and the bank is now looking for ways to improve its target marketing to increase the success rate.

The requirement is to build a model that will help the marketing department identify potential customers who have a higher probability of purchasing a personal loan.

The specific objectives of the model are:

- Predict customer likelihood of purchasing a personal loan.
- Identify customer attributes that drive purchase decisions.
- Determine which customer segment to target with marketing campaigns.

Data Overview

Table below outlines the data dictionary

Variable	Description
ID	Customer ID
Age	Customer's age in completed years
Experience	#years of professional experience
Income	Annual income of the customer (in thousand dollars)
ZIP Code	Home Address ZIP code.
Family	the Family size of the customer
CCAvg	Average spending on credit cards per month (in thousand dollars)
Education	Education Level. 1: Undergrad; 2: Graduate;3: Advanced/Professional
Mortgage	Value of house mortgage if any. (in thousand dollars)
Personal_Loan	Did this customer accept the personal loan offered in the last campaign? (0: No, 1: Yes)
Securities_Account	Does the customer have securities account with the bank? (0: No, 1: Yes)
CD_Account	Does the customer have a certificate of deposit (CD) account with the bank? (0: No, 1: Yes)
Online	Do customers use internet banking facilities? (0: No, 1: Yes)
CreditCard	Does the customer use a credit card issued by any other Bank (excluding All life Bank)? (0: No, 1: Yes)

Note:

- The dataset provided included a total of 5000 rows (customer account details)
- A total of 14 data columns / variables are included
- All variables are of *int* datatype except for the attribute CCAvg which is of the type *float*

Data Overview

The statistical summary of the dataset for all data columns/variables is represented below.

	count	mean	std	min	25%	50%	75%	max
ID	5000.0	2500.500000	1443.520003	1.0	1250.75	2500.5	3750.25	5000.0
Age	5000.0	45.338400	11.463166	23.0	35.00	45.0	55.00	67.0
Experience	5000.0	20.104600	11.467954	-3.0	10.00	20.0	30.00	43.0
Income	5000.0	73.774200	46.033729	8.0	39.00	64.0	98.00	224.0
ZIPCode	5000.0	93169.257000	1759.455086	90005.0	91911.00	93437.0	94608.00	96651.0
Family	5000.0	2.396400	1.147663	1.0	1.00	2.0	3.00	4.0
CCAvg	5000.0	1.937938	1.747659	0.0	0.70	1.5	2.50	10.0
Education	5000.0	1.881000	0.839869	1.0	1.00	2.0	3.00	3.0
Mortgage	5000.0	56.498800	101.713802	0.0	0.00	0.0	101.00	635.0
Personal_Loan	5000.0	0.096000	0.294621	0.0	0.00	0.0	0.00	1.0
Securities_Account	5000.0	0.104400	0.305809	0.0	0.00	0.0	0.00	1.0
CD_Account	5000.0	0.060400	0.238250	0.0	0.00	0.0	0.00	1.0
Online	5000.0	0.596800	0.490589	0.0	0.00	1.0	1.00	1.0
CreditCard	5000.0	0.294000	0.455637	0.0	0.00	0.0	1.00	1.0

Note:

- Certain columns like ***Education*** are categorical columns but are represented as integers
- Certains columns like ***Online*** , ***CreditCard*** etc are Boolean values represented as 1 and 0
- ***Experience*** column has negative values which indicates errorneous data that will need to be corrected

Data Preprocessing

Preprocessing Checks	Actions Taken
Duplicate & Missing Value Checks	▪ No duplicates or missing values detected
Anomalous Values	▪ ***Experience*** variable had a few values that were negative. These negative values are assumed as data input errors and were corrected to be positive. ▪ ***Education*** variable values were represented with meaningful category labels instead of numeric values
Feature Engineering	▪ **Zip Code** variable had 467 unique values across 5000 customer records. ▪ First two digits of ZIP Code were analyzed to be 7 unique values across the dataset ▪ Using the first two digits, Zip Code can be considered a "Category" ▪ Categorical features for following variables were encoded as "Category" ✓ *Education* ✓ *Personal_Loan* ✓ *Securities_Account* ✓ *CD_Account* ✓ *Online* ✓ *CreditCard* ✓ *ZIPCode*

Data Preprocessing

Preprocessing Checks	Actions Taken	

Outlier Checks

- Q1, Q3 and IQR was calculated to identify the lower and upper bounds and outliers in the data
- Table on the right summarizes the outliers for the numeric columns. As can be seen, there are no significant outliers

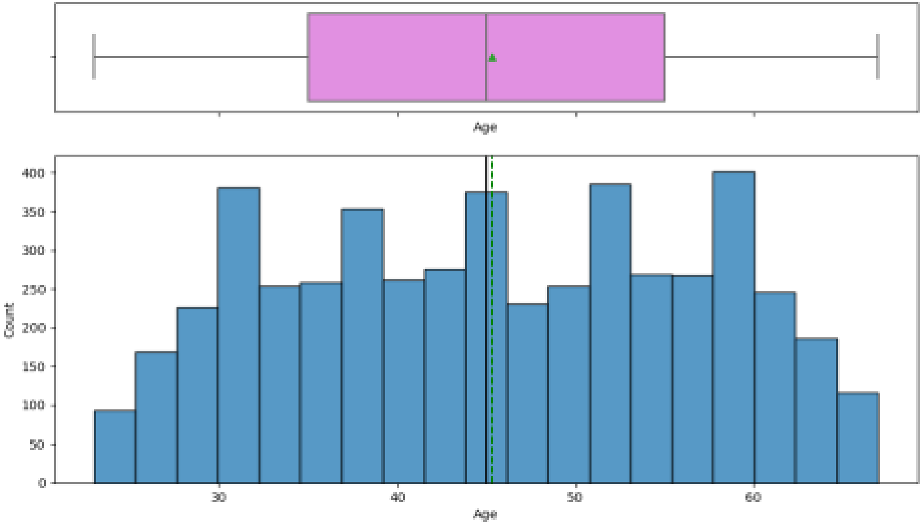
Variable	# of Outliers	% of Outliers
Age	0	0
Experience	0	0
Income	96	1.92
Family	0	0
CCAvg	324	6.48
Mortgage	291	5.82

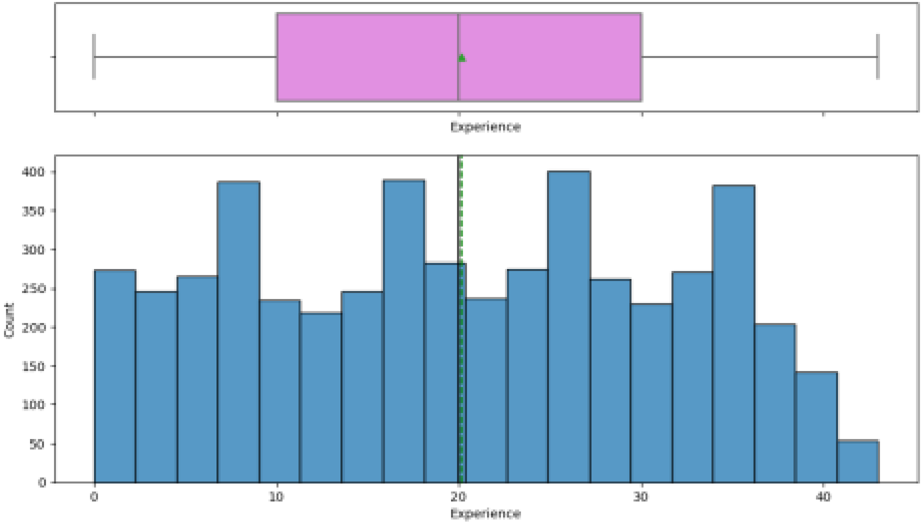
Data Preparation for Modeling

- Before proceeding to build a model, we need to split the data into train, test and validation to be able to evaluate the model that we build on the training data
- Following steps were taken to prepare the data
  - ✓ Separate independent and dependent variables
  - ✓ Categorical features e.g Education and ZIP Code are encoded in dummy variables
  - ✓ Dataset is split into train and test data for model development

Shape of “Train” data set	(3500, 17)
Shape of “Test” data set	(1500, 17)
% of classes in “Train” data set	0: 90.54% & 1: 9.46%
% of classes in “Test” data set	0: 90.07% & 1: 9.93%

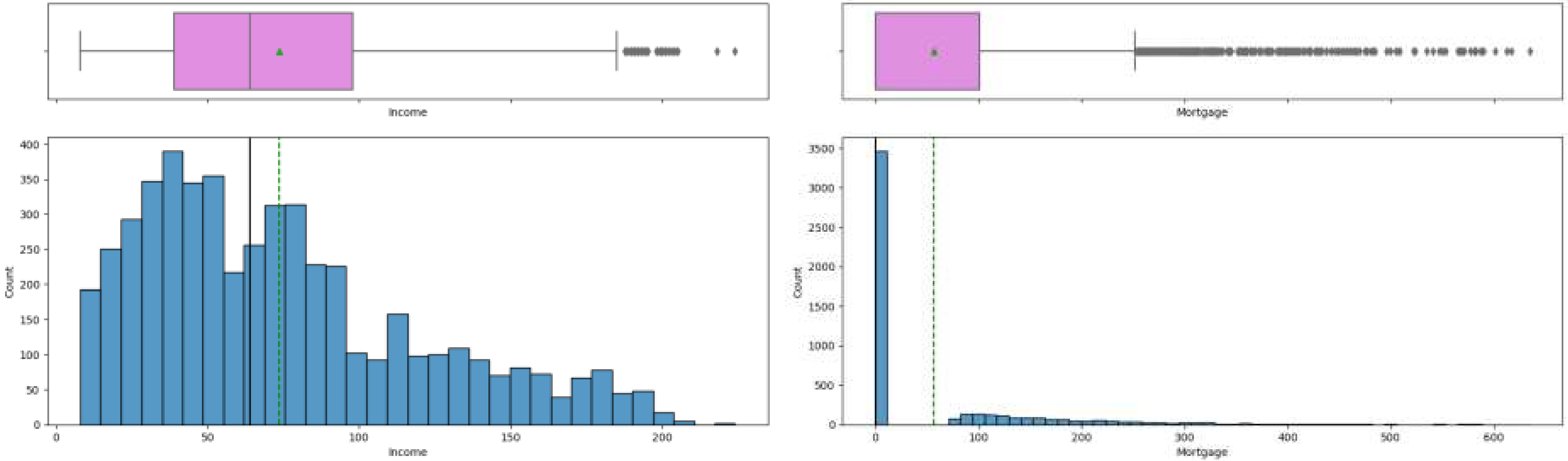
EDA Results – Univariate Analysis





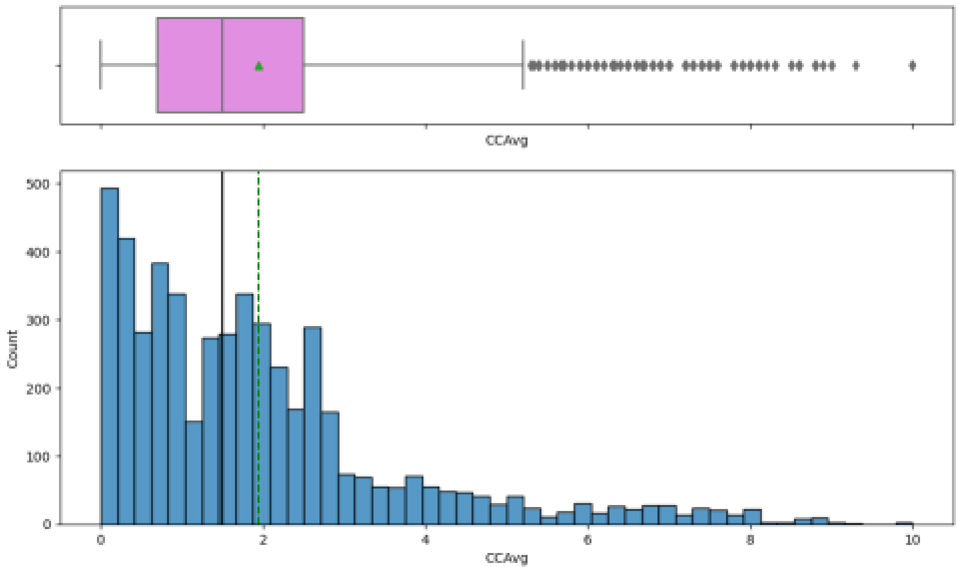
- Average age is about 45.33 years old.
- The age distribution is uniform.
- Average Experience is around 20.1 years.
- Experience has a uniform distribution.
- There are no outliers for both Age and Experience

EDA Results – Univariate Analysis



- Income is right skewed with many outliers on the higher side.
- Average Income is $ 73.77K
- Mortgage is right skewed with many outliers on the higher side.
- Mortgage also has several “zero” values

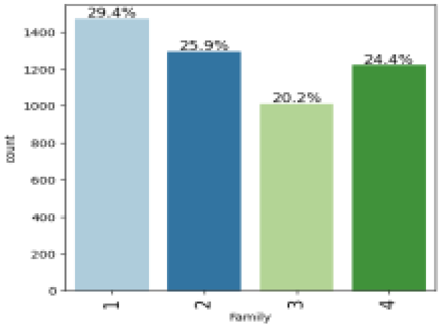
EDA Results – Univariate Analysis

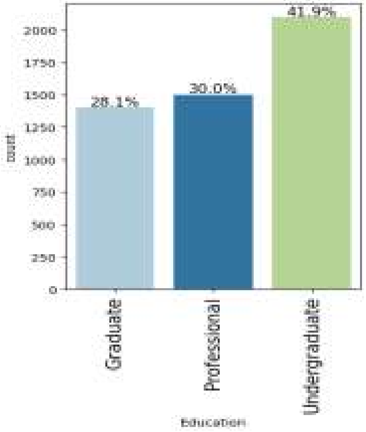


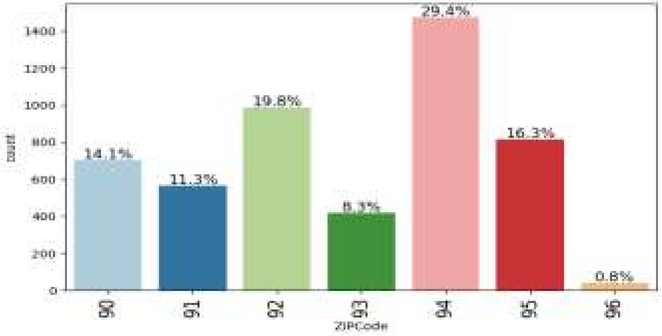
- CCavg is right skewed with many outliers on the higher side.
- CCAvg has a mean value of $ 1.93K

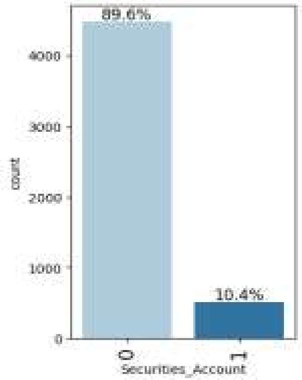
EDA Results – Univariate Analysis

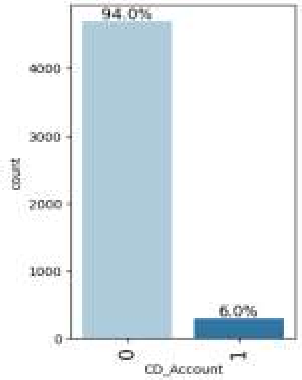
Below charts are self explanatory and depict the distribution across the various values for the categorical variables

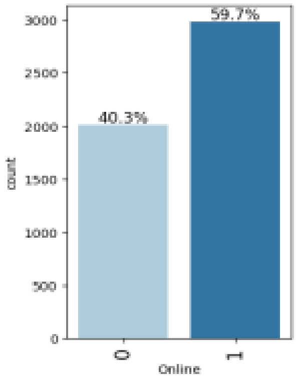


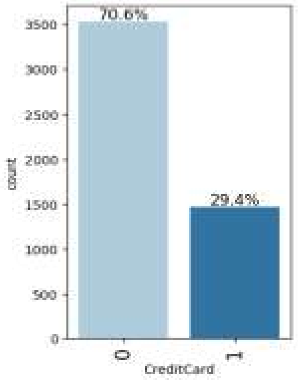








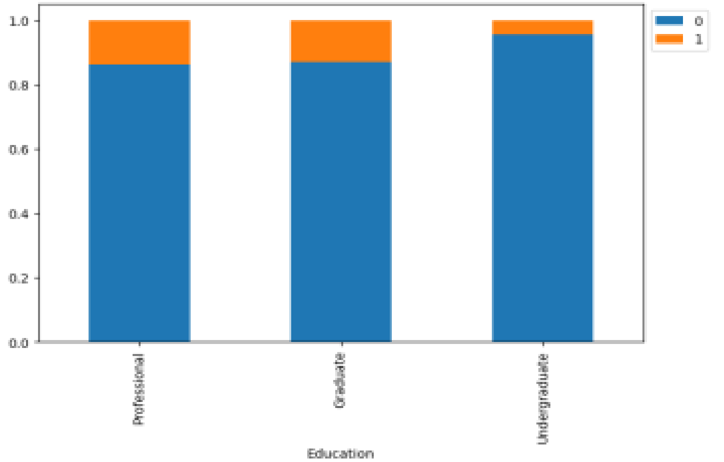


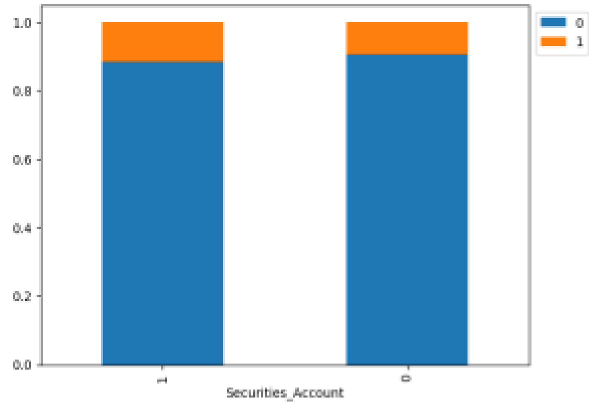


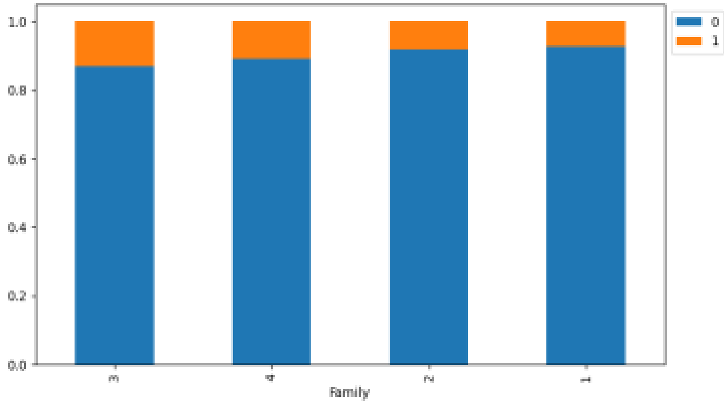
EDA Results – Bivariate Analysis

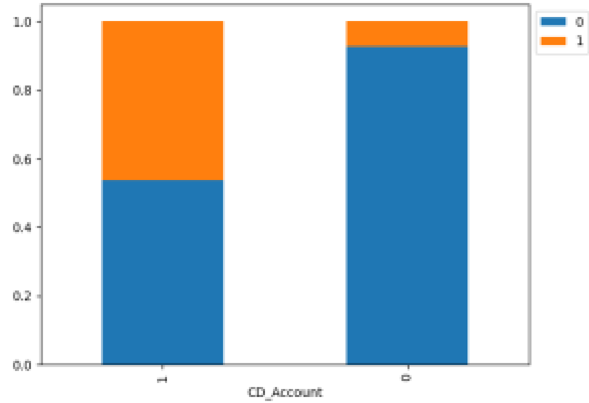
- Age and Experience are very highly correlated
- CCAvg has a high correlation with Income

EDA Results – Bivariate Analysis



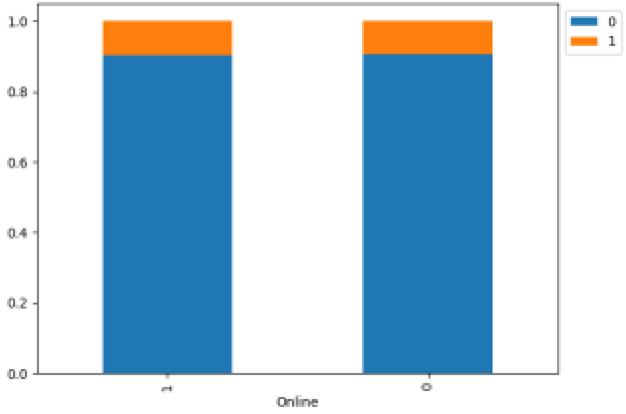


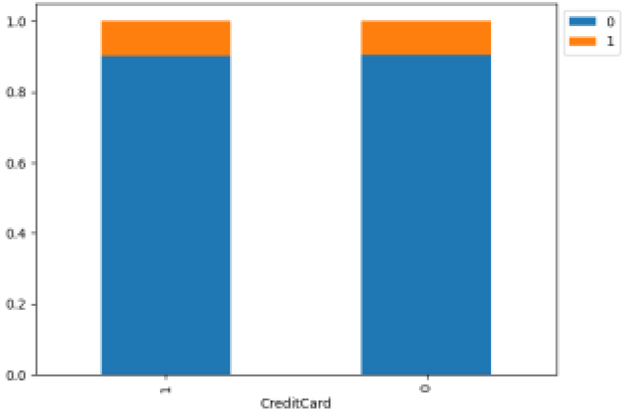


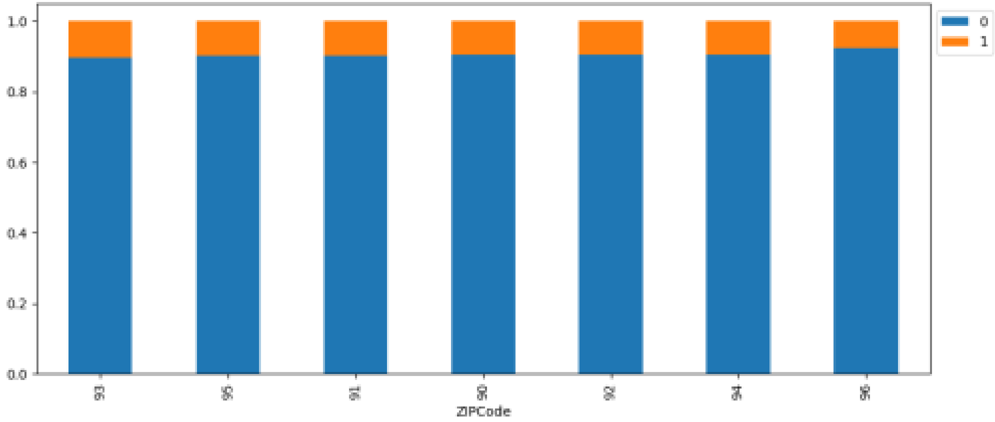


- Customers with Professional and Graduate level of education have more Personal Loans than Undergraduates
- Customers with family of 3 have higher number of loans than family of 4, followed by family of 2 and 1 respectively
- Customers with Securities account have higher number of personal loans
- Customers with CD accounts have a significantly higher number of personal loans

EDA Results – Bivariate Analysis

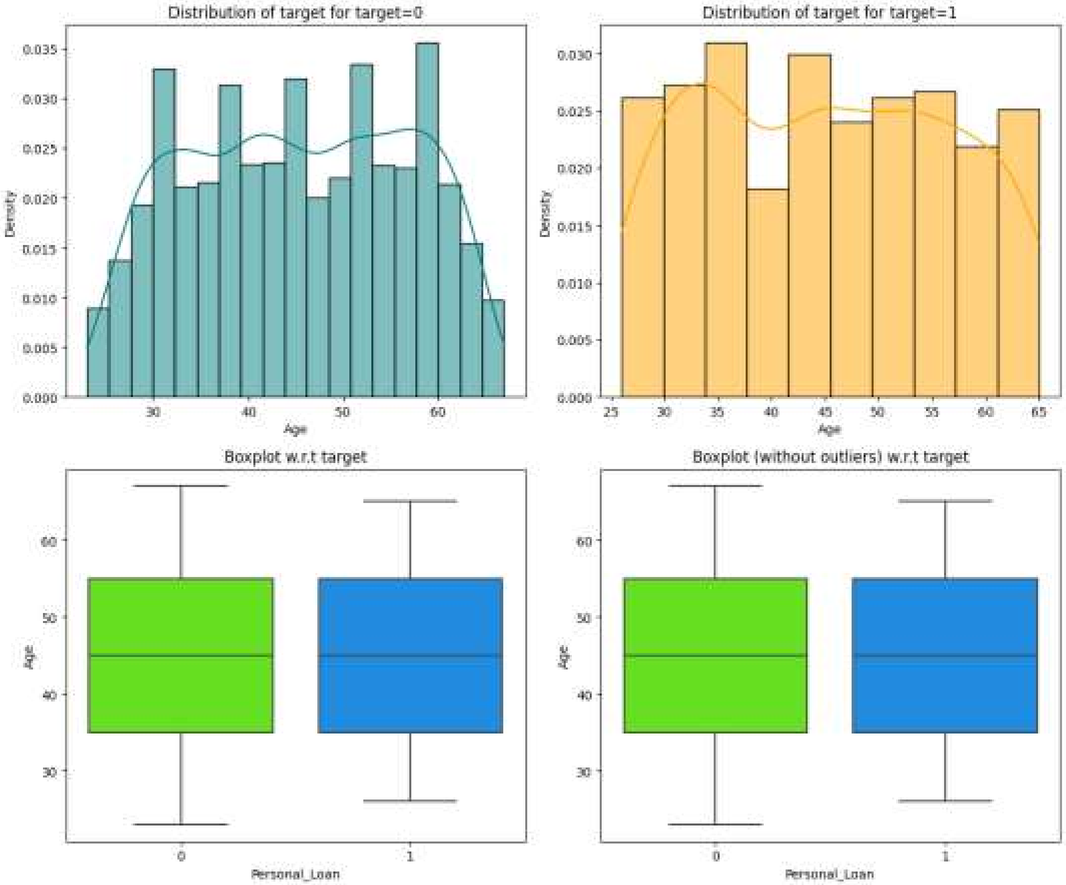






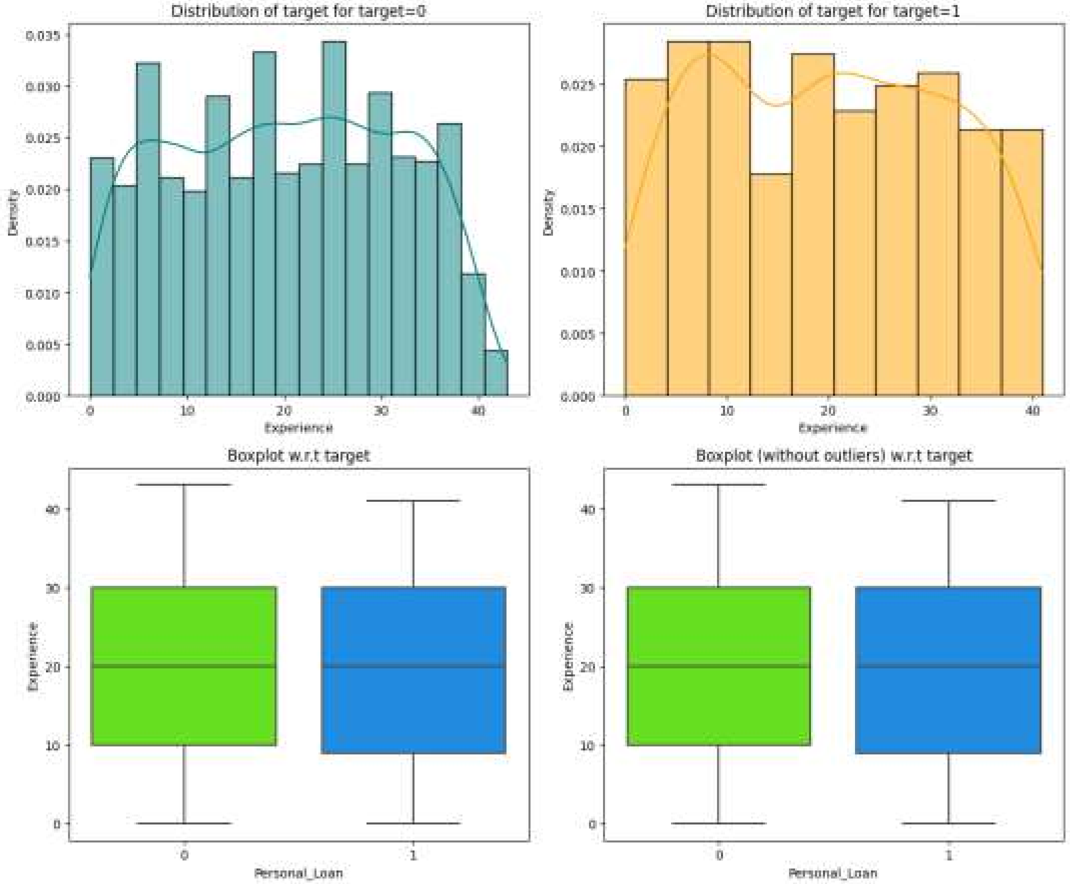
- Customers that use Online and CreditCard have more personal loans
- Customers that are within ZipCode starting with 94 have more personal loans
- No other significant observations

EDA Results – Bivariate Analysis



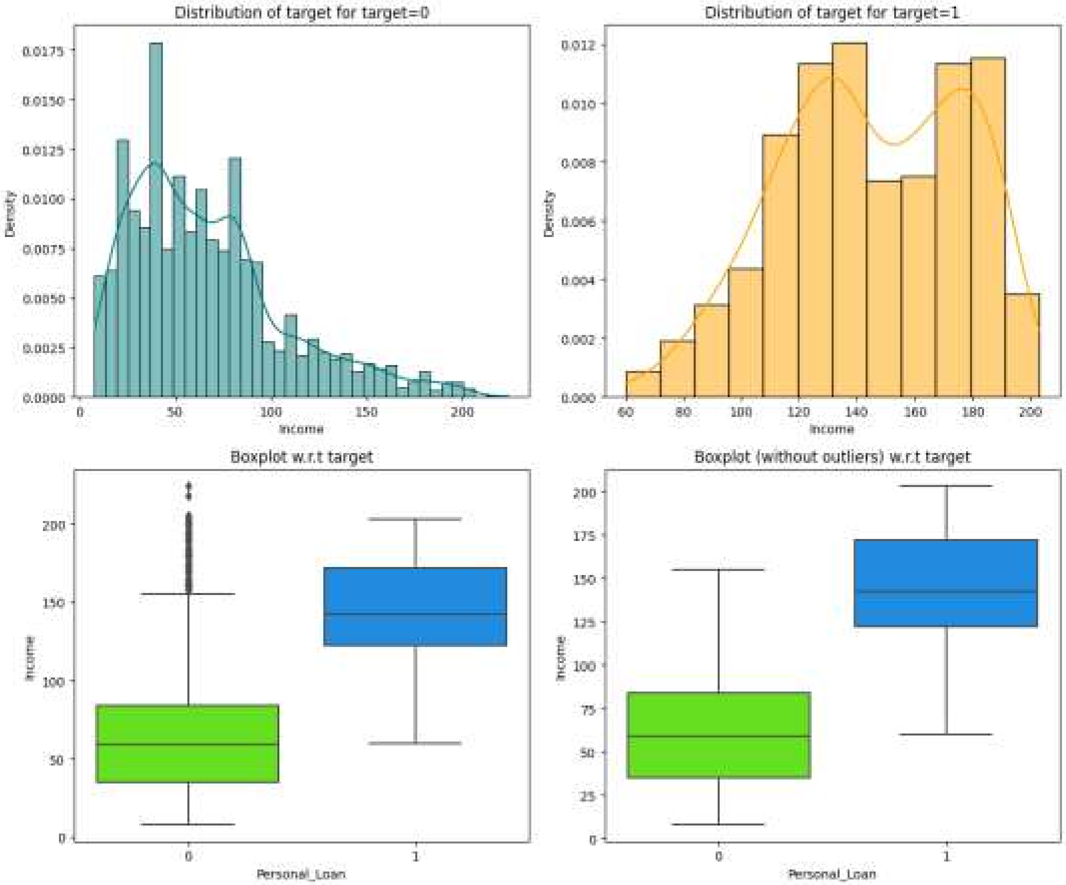
- Customers with age between 26 and 65 have personal loans with highest distribution of loans around age 35 followed by age 45

EDA Results – Bivariate Analysis



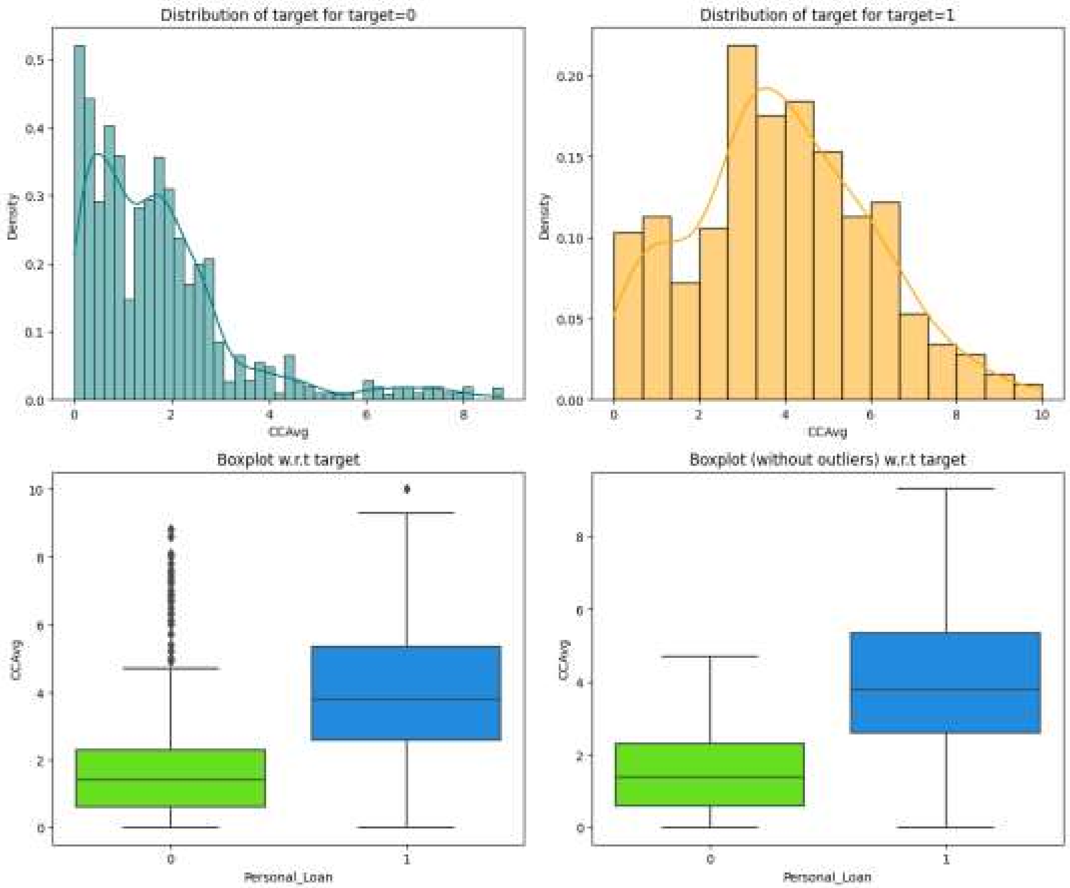
- Highest personal loan distributions are in the experience group of 4 to 12 years

EDA Results – Bivariate Analysis



- Customers with higher incomes have more personal loans

EDA Results – Bivariate Analysis



- On average, those customers with higher credit card usage have personal loans

Model Performance Summary - Logistic Regression: Model Evaluation Criteria

Model can make wrong predictions as below:

- **False Positives (FP):** Predicting a customer will take the personal loan but in reality the customer will not take the personal loan - Loss of resources
- **False Negatives (FN):** Predicting a customer will not take the personal loan but in reality the customer was going to take the personal loan - Loss of opportunity

Which case is more important?

- Losing a potential customer by predicting that the customer will not be taking the personal loan but in reality the customer was going to take the personal loan. Hence, **False Negatives (FN) need to be minimized**

How to reduce this loss of opportunity i.e need to reduce False Negatives?

- Bank would want **Recall** to be **maximized**, Greater the **Recall** higher the chances of minimizing false negatives. Hence, the focus should be on increasing **Recall** or minimizing the false negatives.

Accuracy works best if false positives and false negatives have similar cost. If the cost of false positives and false negatives are very different, it's better to look at both Precision and Recall.

F1 Score is a function of Precision and Recall and is a better measure to use if we need to seek a balance between Precision and Recall AND there is an uneven class distribution (large number of Actual Negatives).

*Thus, **our model performance evaluation will need to be based on - Recall and then on Precision and Recall vakues***

Model Performance Summary – Logistic Regression

Training Performance Comparison

	Logistic Regression		
	Initial - sklearn	Threshold using ROC-AUC Curve	Threshold using Precision-Recall
Accuracy	0.957714	0.913714	0.952857
Recall	0.646526	0.882175	0.740181
Precision	0.873469	0.526126	0.756173
F1 Score	0.743056	0.659142	0.748092

Testing Performance Comparison

	Logistic Regression		
	Initial - sklearn	Threshold using ROC-AUC Curve	Threshold using Precision-Recall Curve
Accuracy	0.956000	0.915333	0.956667
Recall	0.651007	0.845638	0.731544
Precision	0.873874	0.547826	0.813433
F1 Score	0.746154	0.664908	0.770318

Leveraging on our model performance evaluation criterion to be based on – Recall and then on F1 Score and Precision values

- *ROC-AUC based regression approach provides the best Recall values of 0.882175 and 0.845638 for training and test data sets respectively. However, the precision values a significantly lower*
- *Using the Threshold based on Precision-Recall curve balances out both Precision and Recall and provide the highest F1 scores 0.748092 and 0.770318 for training and test data sets respectively, amongst the 3 models*
- *Thus the regression model that using optimal threshold based on Precision-Recall curve is recommended Logistic Regression model*

Model Performance Summary – Decision Tree

Training Performance Comparison

	Decision Tree		
	Initial - Simple	Pre-Prunning	Post Prunning Cost Complexity
Accuracy	1	0.990286	0.993143
Recall	1	0.927492	0.963746
Precision	1	0.968454	0.963746
F1 Score	1	0.947531	0.963746

Testing Performance Comparison

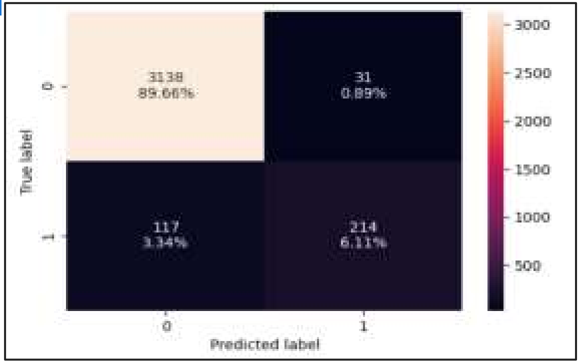
	Decision Tree		
	Initial - Simple	Pre-Prunning	Post Prunning Cost Complexity
Accuracy	0.981333	0.980000	0.984000
Recall	0.899329	0.865772	0.906040
Precision	0.911565	0.928058	0.931034
F1 Score	0.905405	0.895833	0.918367

Leveraging on our model performance evaluation criterion to be based on – Recall and then on F1 Score and Precision values

- *As you can see highlighted on the left, the Recall value on the test data set is the best with the Post Pruning Cost Complexity based model*
- *Precision and F1 Score are also optimized on the test data set with the Cost Complexity based model*
- *Thus the decision tree model based on Cost Complexity Pruning provides the best recall value*
- Following key variables that contribute to predicting target in order of importance –
  - ✓ Education_UnderGraduate
  - ✓ Income
  - ✓ Family
  - ✓ CCAvg
  - ✓ CD_Account
  - ✓ Age

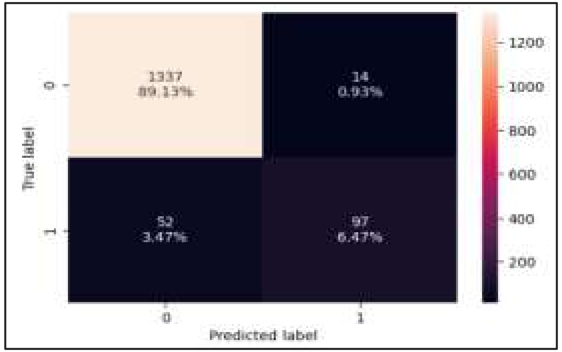
APPENDIX

Model Building: Logistic Regression - Initial



Training performance:

	Accuracy	Recall	Precision	F1
0	0.957714	0.646526	0.873469	0.743056



Test performance:

	Accuracy	Recall	Precision	F1
0	0.956	0.651007	0.873874	0.746154

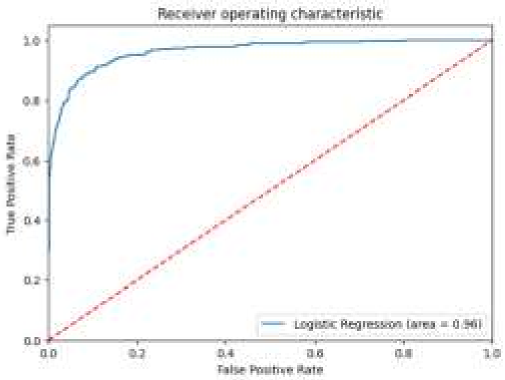
Coefficients

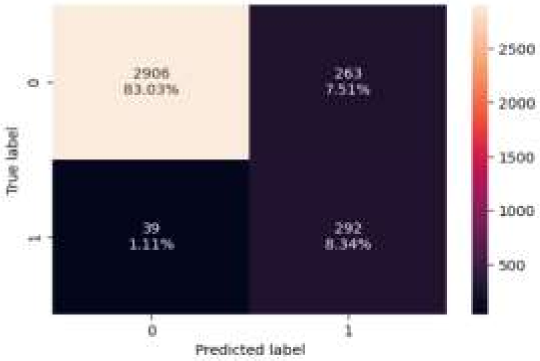
Age: -0.014586219948962118
Income: 0.05264394043893691
Family: 0.4729521306065663
CCAvg: 0.1673810274207664
Mortgage: 0.0007453840217538943
Securities_Account: -1.1819380732550249
CD_Account: 3.852001561510488З
Online: -0.517522314029169
CreditCard: -1.0586010577084575
ZIPCode_91: -1.2032598523806524
ZIPCode_92: -0.18305741460407737
ZIPCode_93: -0.589177080064161
ZIPCode_94: -0.45802954756407516
ZIPCode_95: -0.7208577034737931
ZIPCode_96: -0.215147499969581
Education_Professional: -0.09018737095015456
Education_Undergraduate: -3.5798789390435948

- Preliminary Logistic Regression on the training and the test data was conducted
- Probability of classification labels is used to arrive at a threshold

- **Accuracy** for training set and testing set was **0.9577 and 0.956** respectively
- **Recall** values for both train and test data set are also comparable at **0.6465 and 0.6510**. In fact test set gives a better Recall value.
- Since our objective is to minimize False Negatives and maximize Recall we will try to optimize this model further and see if we can improve on Recall and Accuracy
- Coefficients are also represented. The positive coefficient indicate a feature that predicts class 1, whereas the negative scores indicate a feature that predicts class 0. CD_Account, Family, CCAvg, Income, Mortgage have positive coefficient

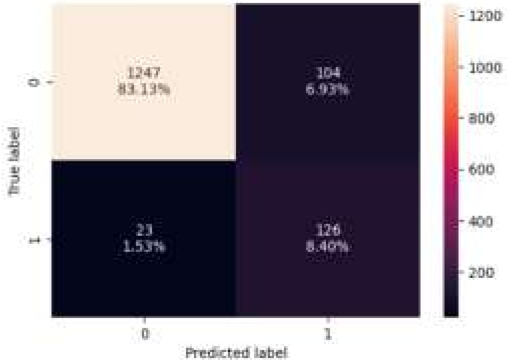
Model Performance Improvement: Logistic Regression : ROC-AUC Threshold





Training performance:

	Accuracy	Recall	Precision	F1
0	0.913714	0.882175	0.526126	0.659142



Test performance:

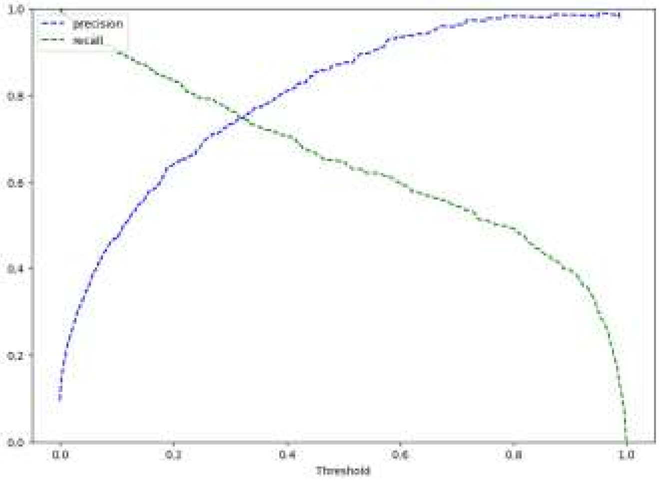
	Accuracy	Recall	Precision	F1
0	0.915333	0.845638	0.547826	0.664908

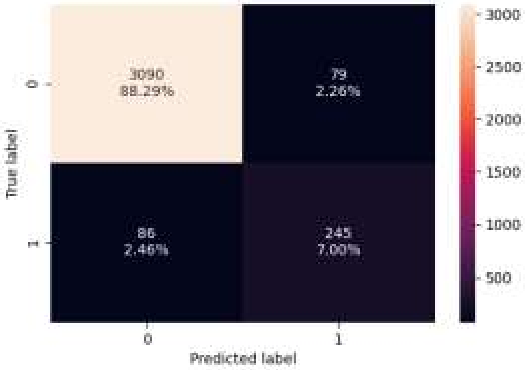
- Logistic Regression is further optimized by calculating Optimal Threshold using ROC-AUC curve. ROC-AUC curve returns the FPR, TPR and Threshold values which takes the original data and predicted probabilities for the class 1. The optimal cut off would be where TPR is high and FPR is low
- **Optimal Threshold** using ROC-AUC curve is **0.12589220403804577**

With this approach

- **Accuracy** for training set and testing set is now **0.9137 and 0.9153** respectively
- **Recall** values for both train and test data set are now at **0.8821 and 0.8456**.
- As evident from the numbers although Accuracy has slightly decreased the Recall value has a sizeable jump due to the usage of optimal threshold from ROC-AUC curve
- This model performance betters than the initial one. Next we can use the Precision-Recall Curve to see if we can optimize the model performance further

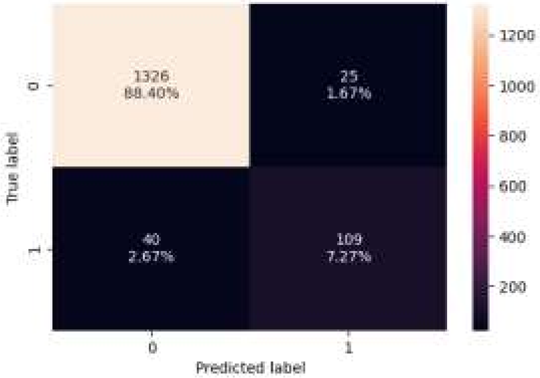
Model Performance Improvement: Logistic Regression : Precision-Recall Curve Threshold





Training performance:

	Accuracy	Recall	Precision	F1
0	0.952857	0.740181	0.756173	0.748092



Test performance:

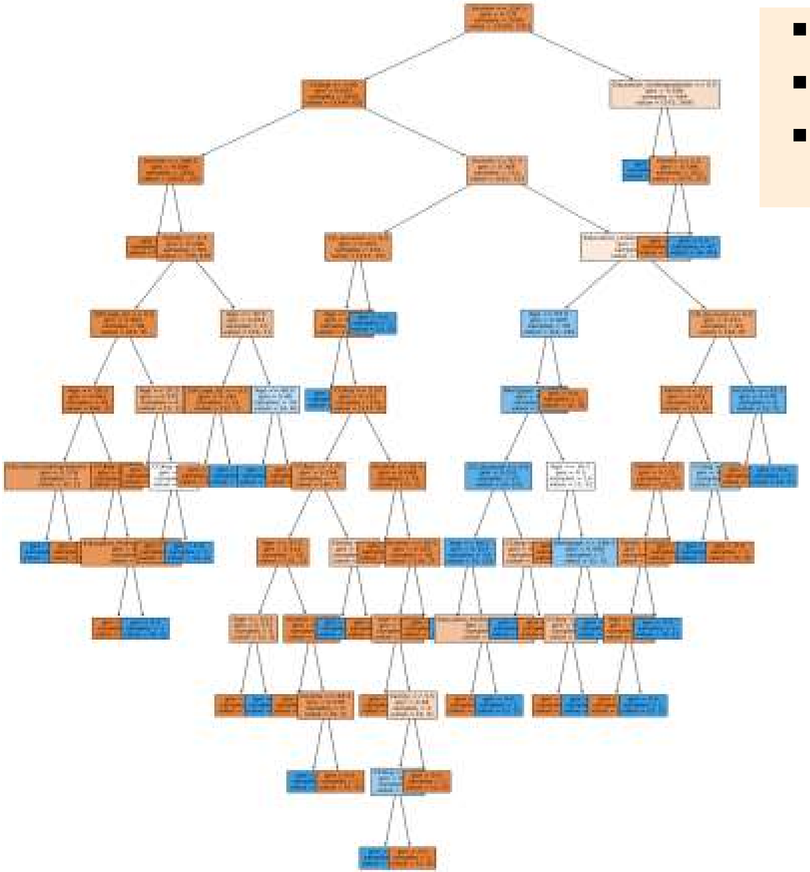
	Accuracy	Recall	Precision	F1
0	0.956667	0.731544	0.813433	0.770318

- Logistic Regression is further optimized by calculating Optimal Threshold using Precision-Recall curve. The Precision-Recall curve shows the tradeoff between Precision and Recall for different thresholds. It can be used to select optimal threshold as required to improve the model improvement.
- **Optimal Threshold** using Precision-Recall curve is **0.33** (Observed from the curve above)

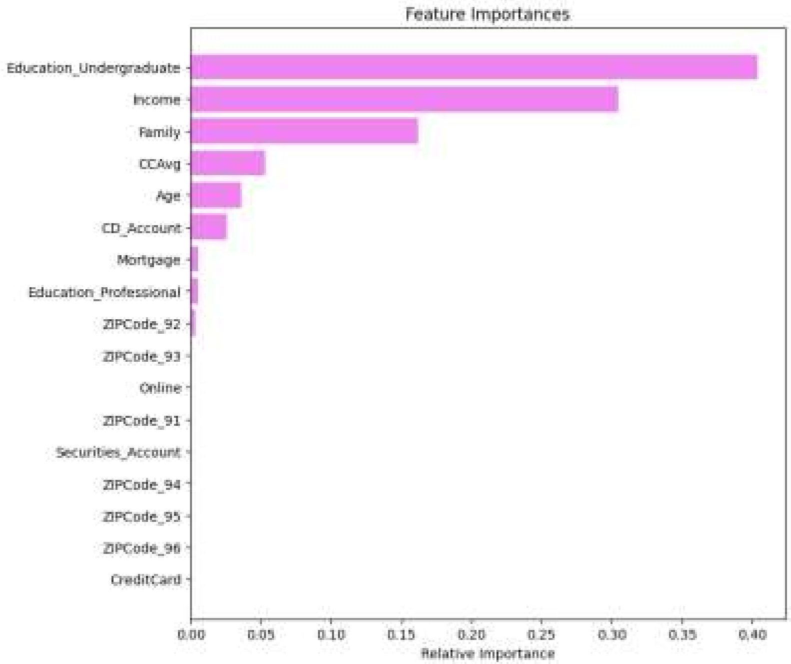
With this approach

- **Accuracy** for training set and testing set is now **0.9529 and 0.9567** respectively
- **Recall** values for both train and test data set are now at **0.7401 and 0.7315**.
- As evident from the numbers although Accuracy has now increased however the Recall value has slightly decreased

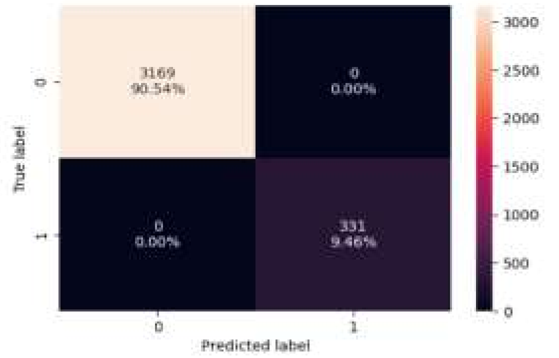
Model Building: Decision Tree - Initial



- The initial model is built using DecisionTreeClassifier
- We are going to be using the 'gini' impurity criteria
- The goal is to find the best splits with the lowest possible Gini Impurity at every step.

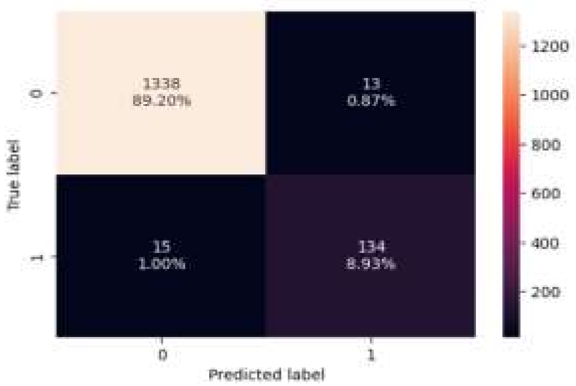


Model Building: Decision Tree - Initial



	Accuracy	Recall	Precision	F1
0	1.0	1.0	1.0	1.0

Training Performance



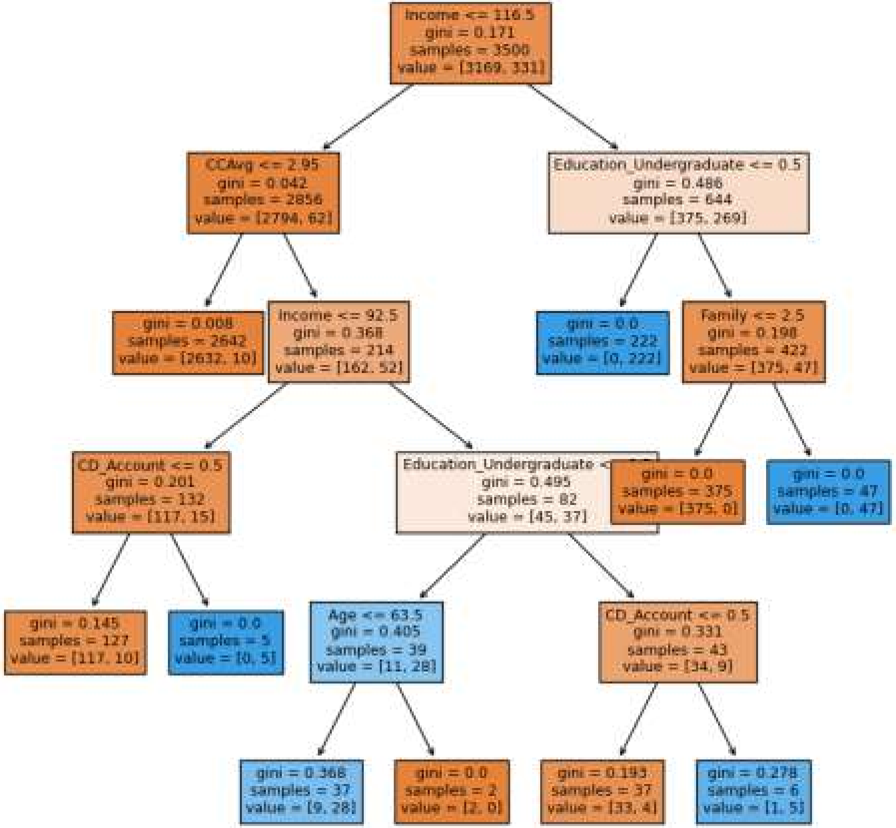
	Accuracy	Recall	Precision	F1
0	0.981333	0.899329	0.911565	0.905405

Testing Performance

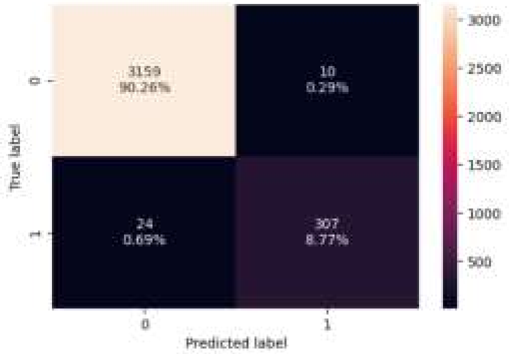
- **Accuracy** for training set and testing set was **1.0 and 0.981333** respectively
- **Recall** values for both train and test data set are at **1.0 and 0.899329**.
- Based on the training results, it is very clear that the model is overfitting. This is also depicted by the decision tree visual on the prior slide
- The Feature Importance chart represented on the prior slide indicates following key variables that contribute to predicting target in order of importance – Education_UnderGraduate, Income, Family, CCAvg, Age, CD_Account and a few others

Model Performance Improvement: Decision Tree : Pre-Pruning

- The improved model is built using DecisionTreeClassifier and pre-pruning techniques are used to avoid overfitting
- Pre-Pruning is based on the following hyper parameters
  - ✓ max_depth: The maximum depth of the decision tree
  - ✓ min_samples_leaf: The minimum number of samples required to split a leaf node
  - ✓ max_leaf_nodes: The maximum number of leaf nodes in the decision tree
  - ✓ Hyper Parameter Values: max_depth key has values from 6 to 15. The min_samples_leaf key has a list with values 1, 2, 5, 7, and 10. The max_leaf_nodes key has a list with values 2, 3, 5, and 10.
- The above dictionary of hyperparameters is used to train a decision tree model using a grid search.
- A GridSearch Cross Validation is a technique that can be used to find the best hyperparameters for a model by evaluating the model with different combinations of hyperparameters.

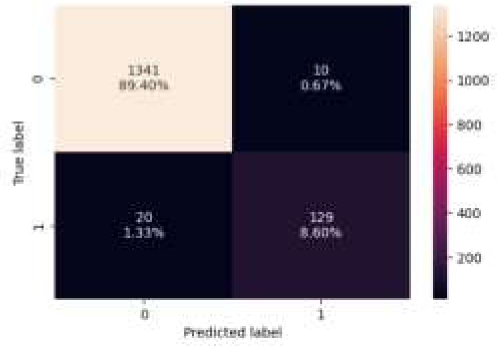


Model Performance Improvement: Decision Tree : Pre-Pruning



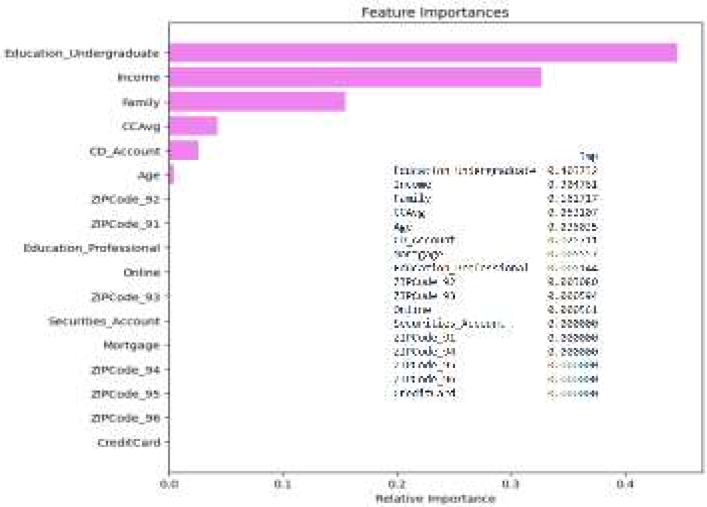
	Accuracy	Recall	Precision	F1
0	0.990286	0.927492	0.968454	0.947531

Training Performance



	Accuracy	Recall	Precision	F1
0	0.98	0.865772	0.928058	0.895833

Testing Performance



Feature Importance

- **Accuracy** for training set and testing set is now **0.990286 and 0.98** respectively
- **Recall** values for both train and test data set are at **0.92 and 0.865772**.
- The Feature Importance chart represented above indicates following key variables that contribute to predicting target in order of importance – Education_UnderGraduate, Income, Family, CCAvg, CD_Account and Age

Model Performance Improvement: Decision Tree : Post-Pruning (Cost Complexity Pruning)

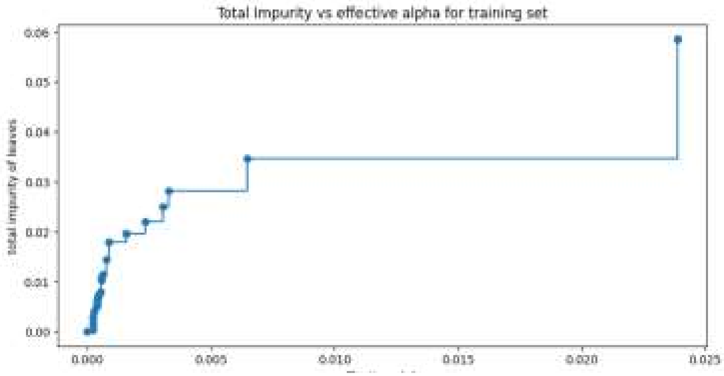
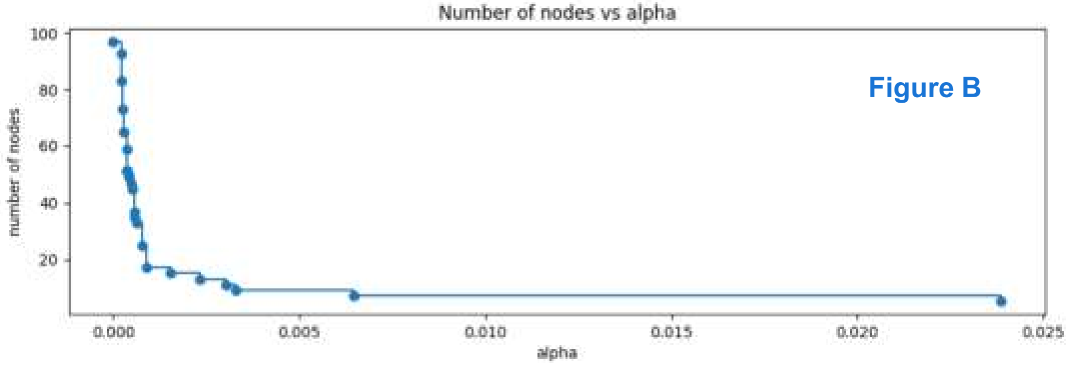


Figure A



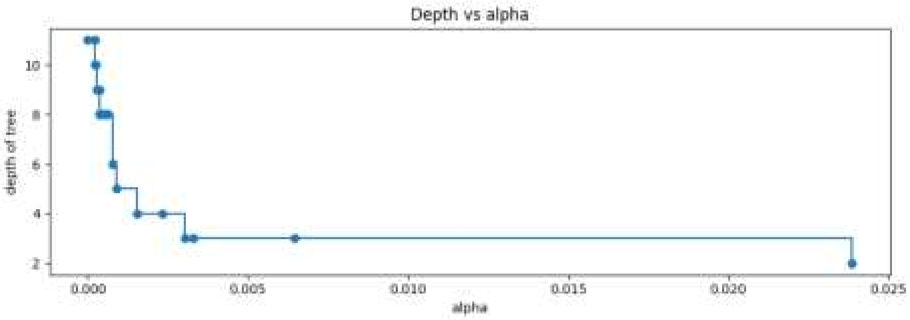


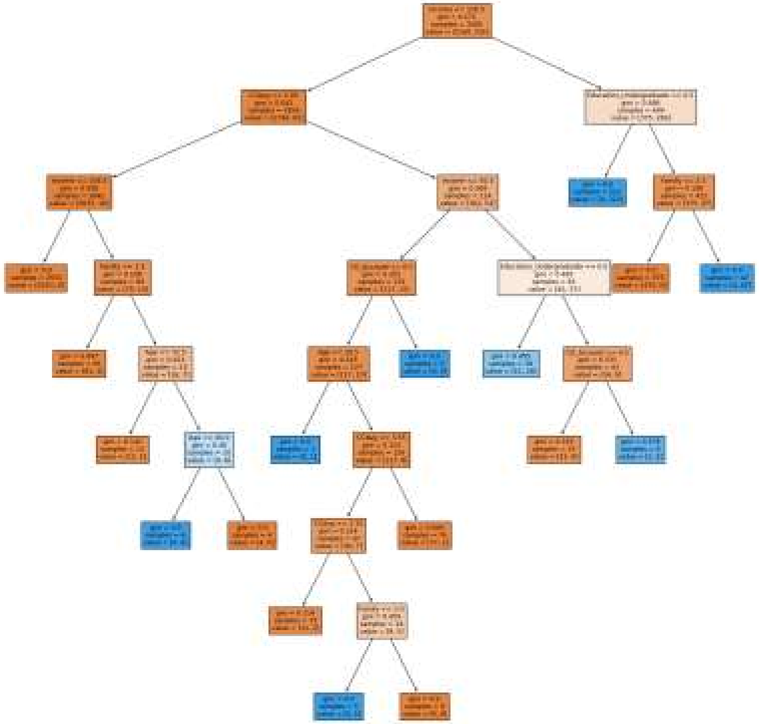
Figure C

- The improved model is built using DecisionTreeClassifier and post-pruning techniques leverage cost complexity pruning
- Cost complexity pruning technique is used to reduce the size of a decision tree by removing nodes that do not contribute significantly to the model's performance.
- We train the decision tree using effective alphas and try to find the best **alpha**
- The higher the **alpha** value, the more nodes will be pruned from the tree. As **alpha** increases, the cost of nodes with higher depths becomes too high and they are pruned from the tree.

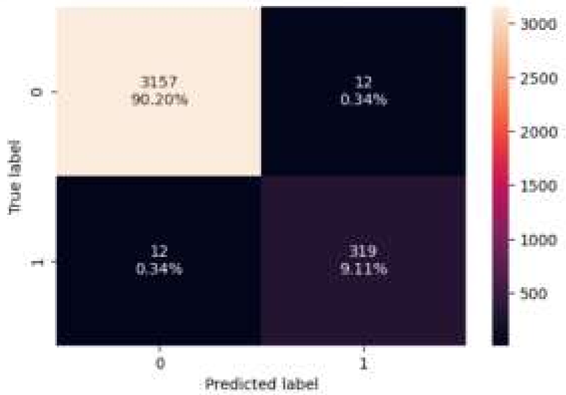
Model Performance Improvement: Decision Tree : Post-Pruning (Cost Complexity Pruning)

Best Fit Model is when ccp_alpha=0.0006209286209286216

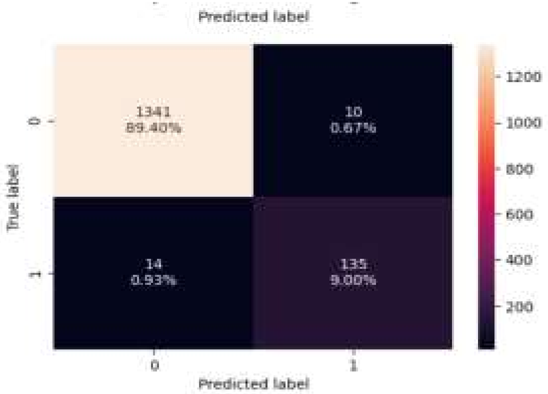
Model Performance Improvement: Decision Tree : Post-Pruning (Cost Complexity Pruning)



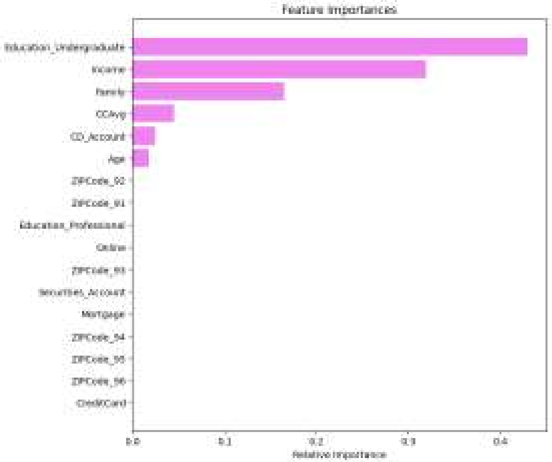
Model Performance Improvement: Decision Tree : Post-Pruning (Cost Complexity Pruning)



	Accuracy	Recall	Precision	F1
0	0.993143	0.963746	0.963746	0.963746



	Accuracy	Recall	Precision	F1
0	0.984	0.90604	0.931034	0.918367



- **Accuracy** for training set and testing set is now **0.993143 and 0.984** respectively. Better than the prior model
- **Recall** values for both train and test data set are at **0.963746 and 0.90604**
- **F1 Score** is also optimized at 0.963746 and 0.918367 for train and test data
- The Feature Importance chart represented above indicates following key variables that contribute to predicting target in order of importance – Education_UnderGraduate, Income, Family, CCAvg, CD_Account and Age
- These are the most optimized values

Happy Learning !