ML Analysis for AllLife Bank Personal Loan Campaign

Project 2, PGP AI/ML: Business Analytics

Saurabh Mittal

2/1/24

Contents / Agenda

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

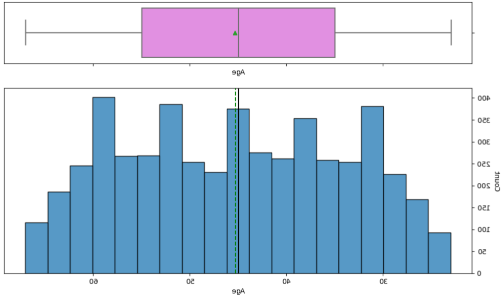
Executive Summary

- A Machine-Learning (ML)-based model was built based on recent successes with Loan campaign to identify new target groups for designing new campaigns.
- Undergraduates from high income families is the target segment that has the least likelihood of a loss (highest Recall)
- Three most important criteria (features) for identifying the target segment: (1) Income less than $116K, (2) Holding CC balance greater than $2950 and (3) Having an Undergraduate education
- Undergraduates are 42% of the dataset and are the most under-utilized segment
- No significant correlation between various variables that may impact any business decision. Multiple rules were discovered from the ML-based Decision Tree designed from 17 features.

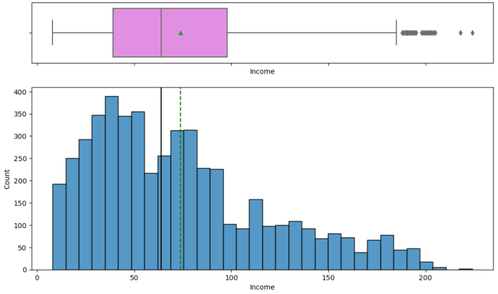
Business Problem Overview and Solution Approach

- **Problem**
  - AllLife bank intends to increase its loan business and increase income from interest from the new loans
  - Prior success with depositors (liability customers) encouraged AllLife to design a new campaign.
  - AllLife would like to analyze current data (with prior success) to learn from it and eventually identify potential target customers
- **Solution approach / methodology**
  - Identify which segment of customers to target and which customer attributes are the most significant drivers for loan success, i.e., the loan gets repaid (higher Recalls).
  - Devise a Machine-Learning (ML)-based model which can identify target groups with high Recall rate

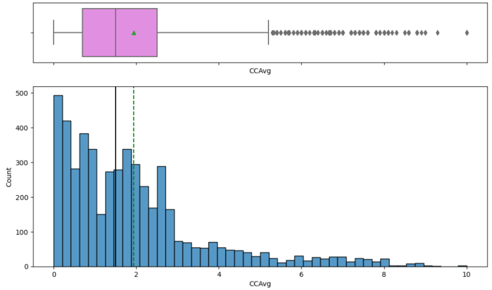
EDA Results (Key Charts)



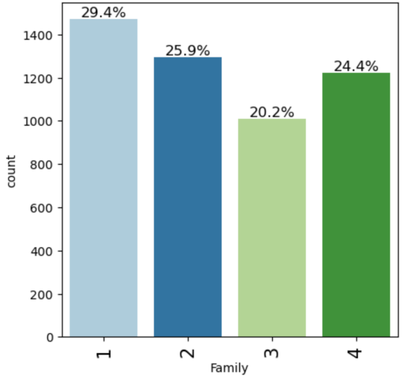
50% customers in 35-55 age group



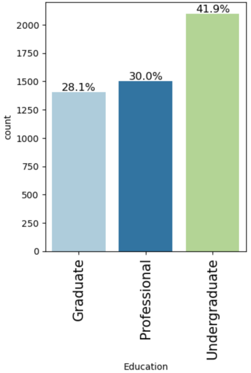
50% customers earn between ~$40K-$100K



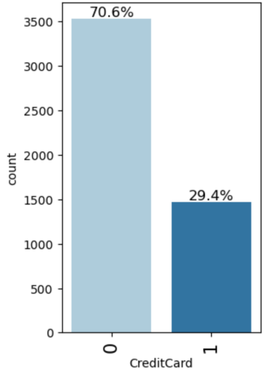
50% customers carry CC balances between ~$900-$2200



Family contains 1-4 members, with 55% families with 2 or less members

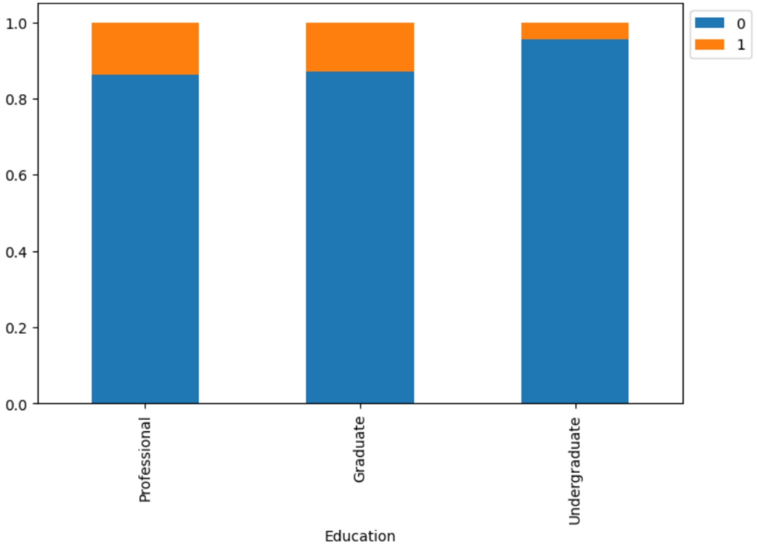


Undergraduates comprise 42% of the dataset

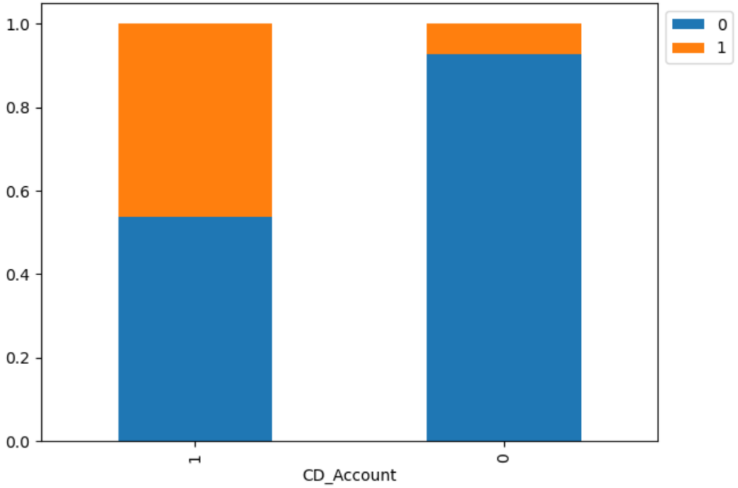


30% of customer that hold CC debt also carry Personal Loans

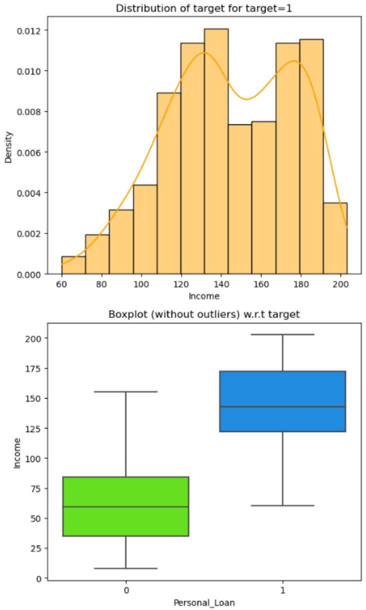
EDA Results (Contd.)



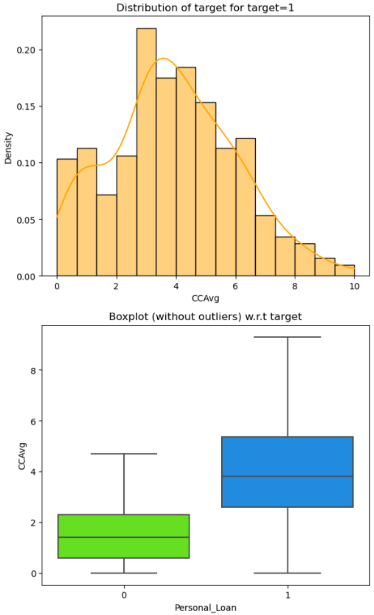
90% Undergraduates have no loans



50% of customers that have CD deposits also have Personal Loans



50% of **customers with loans** are in the income bracket of ~$125K - $170K



50% of **customers with loans** contain CC balances in the range of $2.2K - $5K

Data Preprocessing

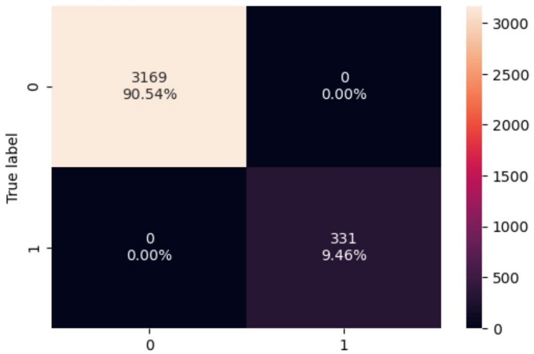
- Duplicate value check: no duplicates
- Missing value treatment: no missing values
- Outlier check: normalized data between the 0.25 and 0.75 quantiles
- Feature Engineering: 17 features
- Data preprocessing for modeling as below

```
Shape of Training set :  (3500, 17)
Shape of test set :  (1500, 17)
Percentage of classes in training set:
0    0.905429
1    0.094571
Name: Personal_Loan, dtype: float64
Percentage of classes in test set:
0    0.900667
1    0.099333
Name: Personal_Loan, dtype: float64
```

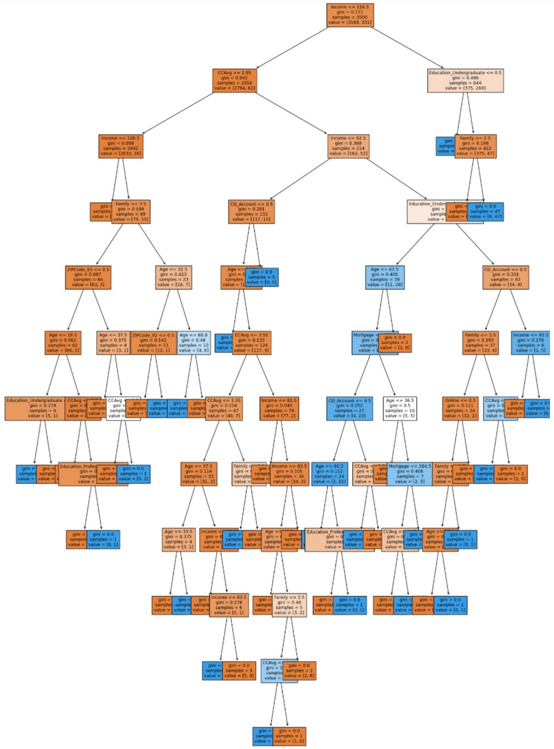
Model Building

- Method
  - To build a Decision Tree, there are 4 evaluation scores that one must implement: Accuracy, Recall, Precision and F1 scores
  - For the current problem, Recall is the score we must maximize on as this score keeps track of minimizing false negatives, i.e., cases where the model predicts the customer is not taking the loan but in reality, the customer was going to take the loan
  - Create functions to calculate the above scores and confusion matrix.
  - Utilize **DecisionTreeClassifier()** with gini criteria and random_state=1 for repeatable results
- Comment on the model performance
  - The ***model_performance_classification_sklearn_with_threshold*** function will be used to check the model performance of models.
  - The ***confusion_matrix_sklearn_with_threshold*** function will be used to plot confusion matrix.

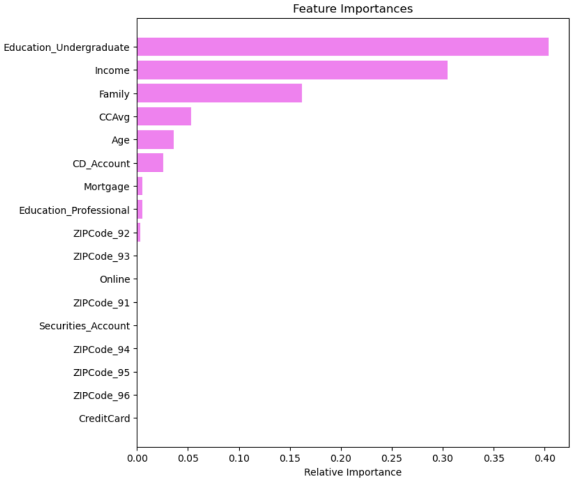
Model Performance on Training Data



	Accuracy	Recall	Precision	F1
0	1.0	1.0	1.0	1.0



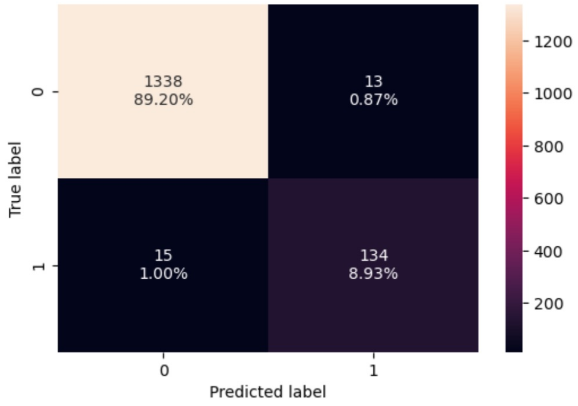
Decision Tree Depth = 11



Feature Ranking:

1. Education_Undergraduate
2. Income
3. Family
4. CCAvg
5. Age
6. CD_Account

Model Performance on Test Data

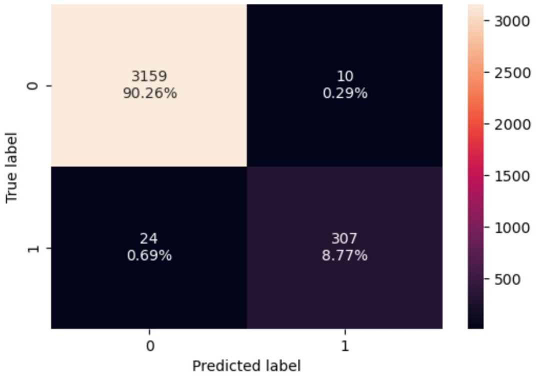


	Accuracy	Recall	Precision	F1
0	0.981333	0.899329	0.911565	0.905405

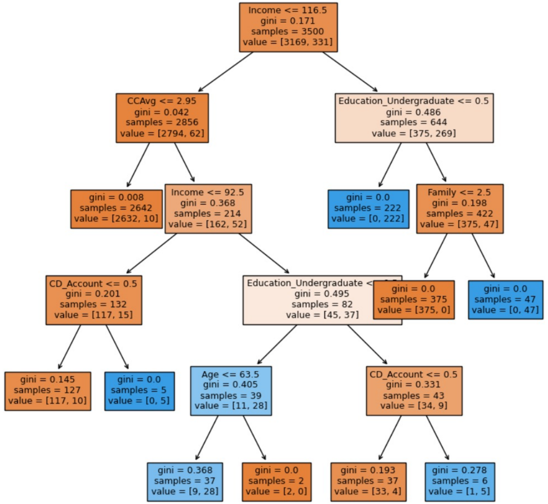
Synopsys:

- Test Model shows lower Recall numbers by 11%
- Model needs to be tuned either by pre-pruning, post-pruning or both

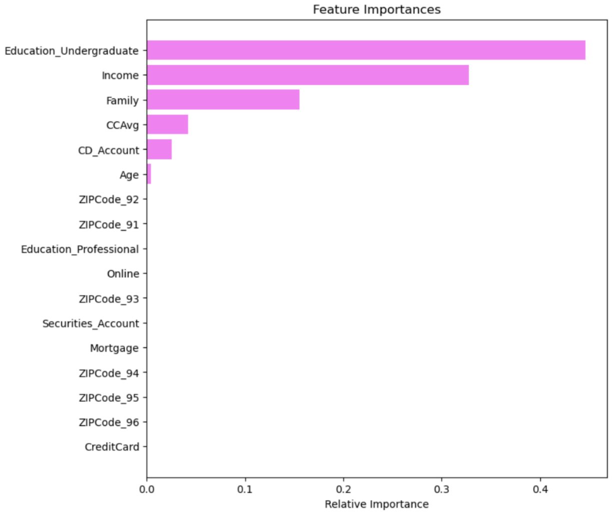
Model Pre-Pruning Performance on Training Data



	Accuracy	Recall	Precision	F1
0	0.990286	0.927492	0.968454	0.947531



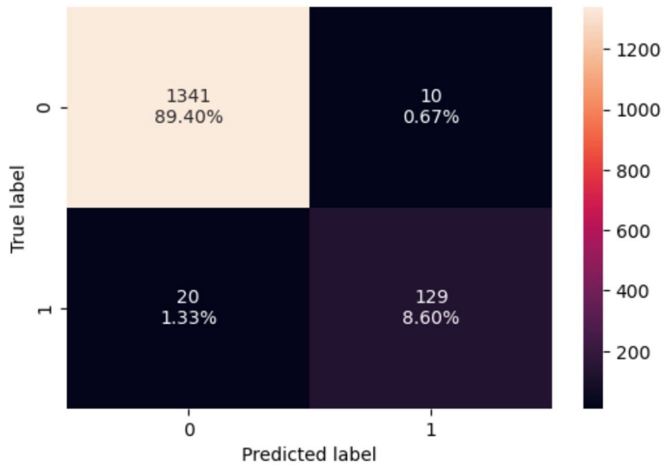
Decision Tree Depth = 6



Feature Ranking:

1. Education_Undergraduate
2. Income
3. Family
4. CCAvg
5. **CD_Account**
6. **Age**

Model Pre-Pruning Performance on Test Data



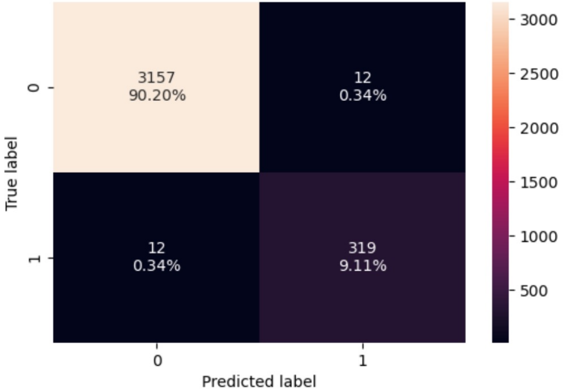
	Accuracy	Recall	Precision	F1
0	0.98	0.865772	0.928058	0.895833

Synopsys:

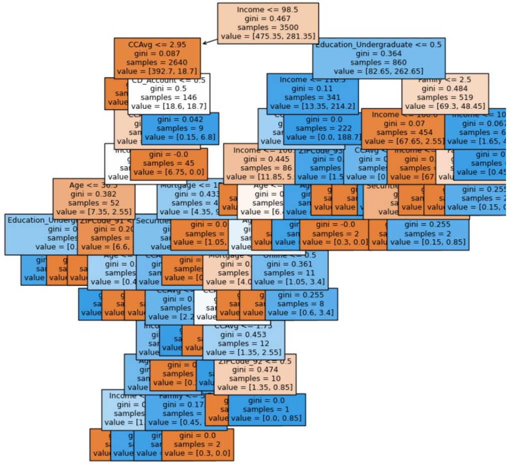
- Pre-Pruning Test Model shows reduction in Recall number from 0.9275 to 0.8658
- Model needs to be treated for cost-complexity analysis for identifying the right alpha value for the best model

Model Post-Pruning Performance on Training Data

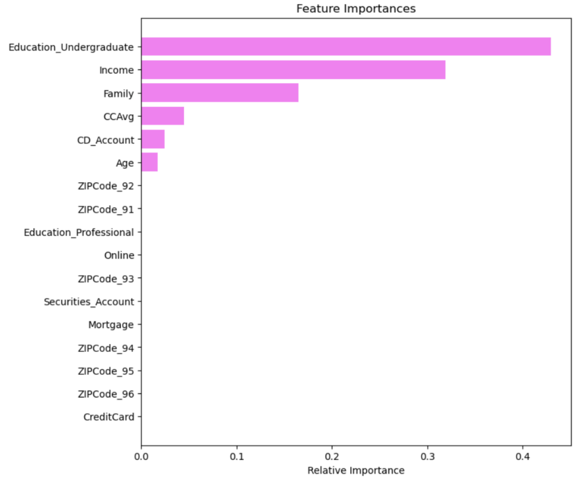
- Alpha: 0.0006209286209286216



	Accuracy	Recall	Precision	F1
0	0.993143	0.963746	0.963746	0.963746



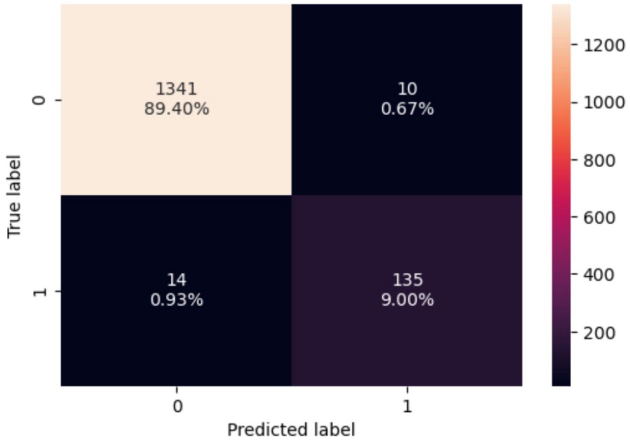
Decision Tree Depth = 12



Feature Ranking:

1. Education_Undergraduate
2. Income
3. Family
4. CCAvg
5. **CD_Account**
6. **Age**

Model Post-Pruning Performance on Test Data



	Accuracy	Recall	Precision	F1
0	0.984	0.90604	0.931034	0.918367

Synopsys:

- Post-Pruning Test Model shows reduction in Recall number from 0.9637 to 0.9060

Model Performance Comparison

Training performance comparison:

	Decision Tree sklearn	Decision Tree (Pre-Pruning)	Decision Tree (Post-Pruning)
Accuracy	1.0	0.990286	0.993143
Recall	1.0	0.927492	0.963746
Precision	1.0	0.968454	0.963746
F1	1.0	0.947531	0.963746

Test set performance comparison:

	Decision Tree sklearn	Decision Tree (Pre-Pruning)	Decision Tree (Post-Pruning)
Accuracy	0.981333	0.980000	0.984000
Recall	0.899329	0.865772	0.906040
Precision	0.911565	0.928058	0.931034
F1	0.905405	0.895833	0.918367

Conclusions

- Decision Tree with Post-pruning is giving the highest Recall
- However, the Tree with Pre-Pruning is not complex and easy to interpret with no marked difference between the feature rank order.

Outcome: Decision Tree Model Solution

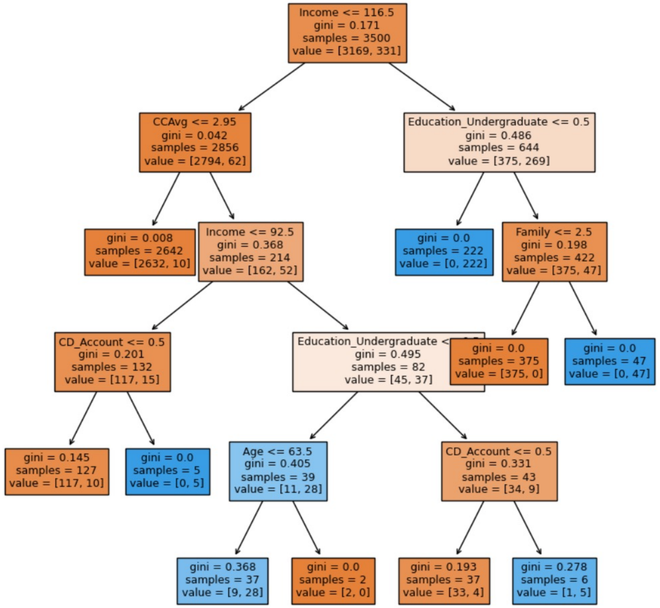
Major rules to be executed in the following order:

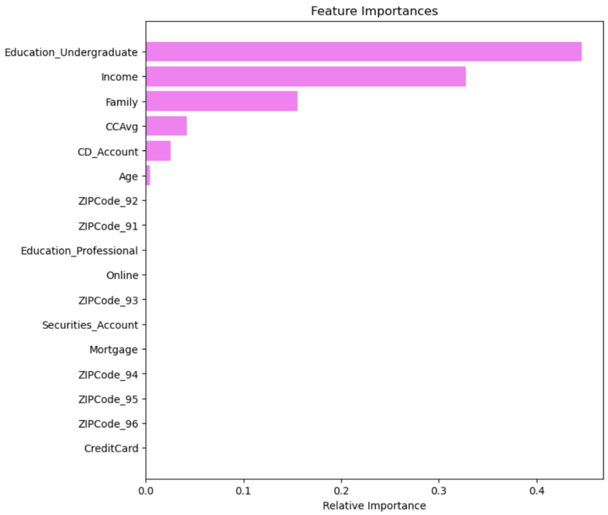
(Left tree)

- Income < = $116.5K
- CCAvg <= $2.95K

(Right tree)

- Income > $116.5K
- Undergraduate >0.5





Feature Ranking:

1. Education_Undergraduate
2. Income
3. Family
4. CCAvg
5. CD_Account
6. Age

APPENDIX

Data Background and Contents

- Data provided by AllLife in Loan_Modelling.csv file

Happy Learning !