# Machine Learning: Supervised Learning

Nick Lutostanski
PGP AI/ML for Business Applications

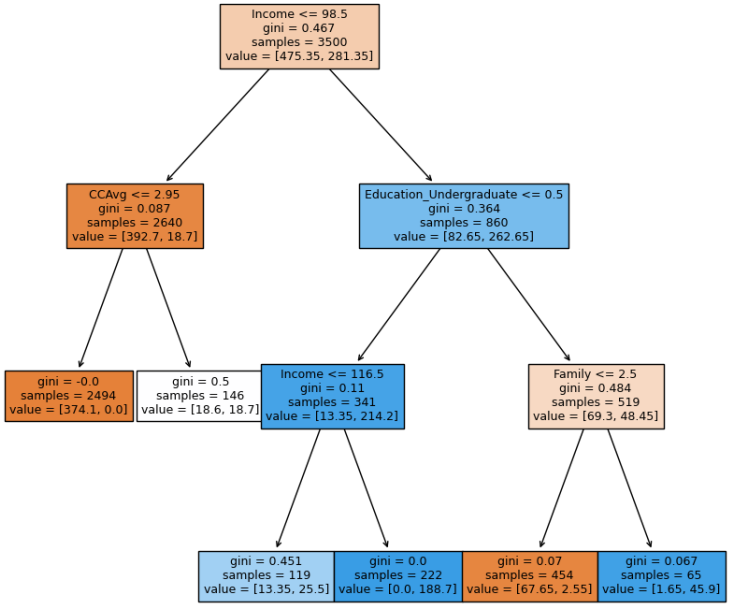
2.1.24

# Contents / Agenda

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

# Executive Summary

- Income, education, family, and monthly credit card spend are the most important features in predicting personal loan customers.
- The marketing campaign should be focused on customers who have higher incomes (over $98.5K) with an undergraduate education and a growing family (1 or 2 children). These customers are dramatically more likely to take a personal loan and thus convert into asset customers.
- The best model to use is the Cost-Complexity Model with a ccp_value of .01. It has a .986 recall value on the test set.



# Business Problem Overview and Solution Approach

- AllLife Bank is looking for a way to predict which liability customers (depositors) they can convert to asset customers (borrowers) by predicting which of them are likely to take out a personal loan with the bank. To do this, we need to understand which customer attributes are most significant in driving predicting whether or not a customer will take a personal loan so that we can target them with a new campaign.
- We are going to build 3 supervised learning decision tree algorithms and determine which one is most likely to produce the best results to predict which customers should be targeted in the new campaign.
- We are going to consider maximized Recall the best evaluator of the models since we want to minimize False Negatives. A false negative in this context means a customer who was predicted NOT to take a loan would have taken it, leading to a loss of opportunity/profit.

# Data Overview – Features and Description

## Data Dictionary

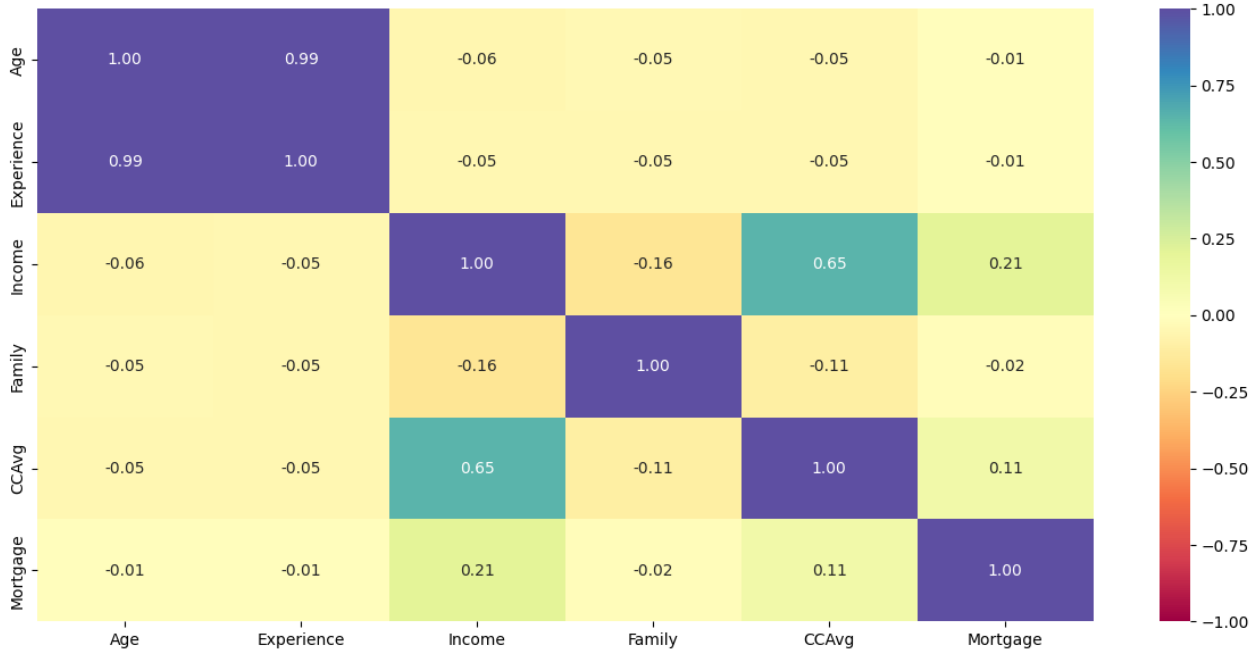
- `ID`: Customer ID
- `Age`: Customer's age in completed years
- `Experience`: #years of professional experience
- `Income`: Annual income of the customer (in thousand dollars)
- `ZIP Code`: Home Address ZIP code.
- `Family`: the Family size of the customer
- `CCAvg`: Average spending on credit cards 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)

# EDA Results - Describe

| | count | mean | std | min | 25% | 50% | 75% | max |
|---|---|---|---|---|---|---|---|---|
| **Age** | 5000.0 | 45.338400 | 11.463166 | 23.0 | 35.0 | 45.0 | 55.0 | 67.0 |
| **Experience** | 5000.0 | 20.134600 | 11.415189 | 0.0 | 10.0 | 20.0 | 30.0 | 43.0 |
| **Income** | 5000.0 | 73.774200 | 46.033729 | 8.0 | 39.0 | 64.0 | 98.0 | 224.0 |
| **Family** | 5000.0 | 2.396400 | 1.147663 | 1.0 | 1.0 | 2.0 | 3.0 | 4.0 |
| **CCAvg** | 5000.0 | 1.937938 | 1.747659 | 0.0 | 0.7 | 1.5 | 2.5 | 10.0 |
| **Mortgage** | 5000.0 | 56.498800 | 101.713802 | 0.0 | 0.0 | 0.0 | 101.0 | 635.0 |

- Customers at the bank are 23 – 67 years old, with a mean and median around 45, meaning it's somewhat normally distributed.
- Income is from $8,000 to $224,000, with a mean of 73,774 and a median of $64,000
- Average spending (CCAvg) ranges from 0 to $10,000 per month.
- It appears that a significant amount of customers who have a bank account with AllLife Bank do not have their mortgages through the bank. We will keep an eye out to see how much this impacts our predictions.
- Age and Experience columns have similar standard deviations and similar increments in the interquartile range. Would dropping one be beneficial since it essentially captures the same information twice?

# EDA Results - Correlation

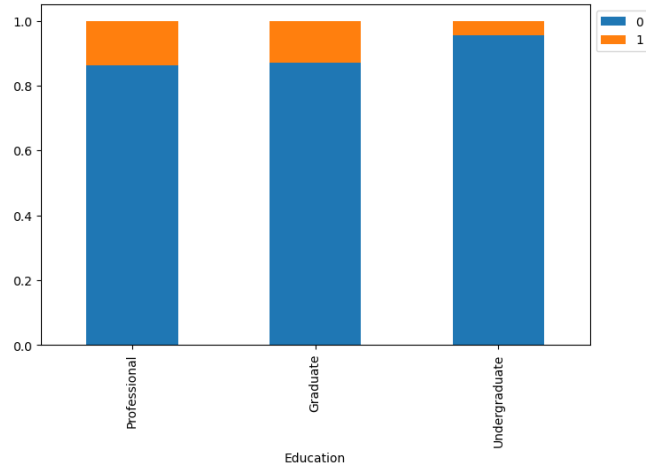



- Experience and age are highly correlated. We might consider that we are simply counting the same feature twice or not (i.e. can we drop one and still capture the same data without including noise).
- Income is highly correlated with average monthly credit card spending, mortgage, and negatively correlated with family size.

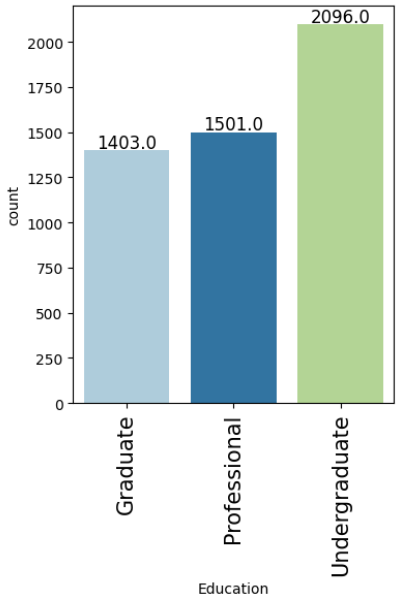
# EDA Results - Education

```
stacked_barplot(data, "Education", "Personal_Loan")
```

```
Personal_Loan     0    1   All
Education
All            4520  480  5000
Professional   1296  205  1501
Graduate       1221  182  1403
Undergraduate  2003   93  2096
```




```
labeled_barplot(data, 'Education')   ## Complete th
```

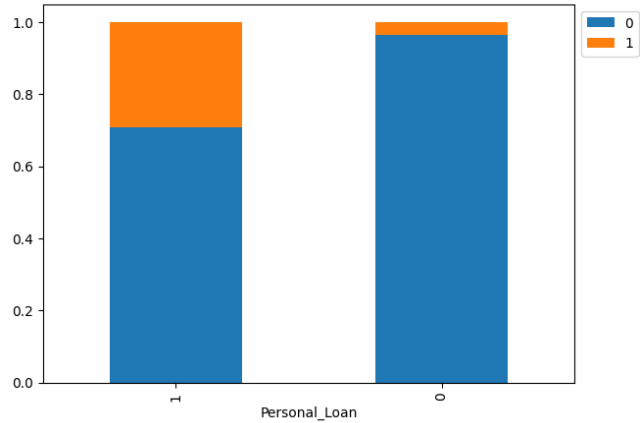



- There appears to be a large difference between undergraduate customers and Graduate/Professional customers and their propensity to take out a personal loan. We will keep an eye on how significant this is when we examine the feature importance of our decision tree models.
- Customers with an undergraduate degree are overrepresented in our data, so this might have a significant impact.

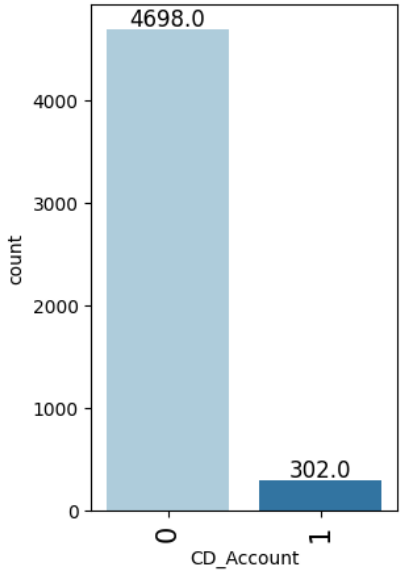
# EDA Results – CD Accounts

```
stacked_barplot(data, "Personal_Loan", "CD_Account") ## Complete the code to plot sta
```

```
CD_Account        0    1   All
Personal_Loan
All            4698  302  5000
0              4358  162  4520
1               340  140   480
```

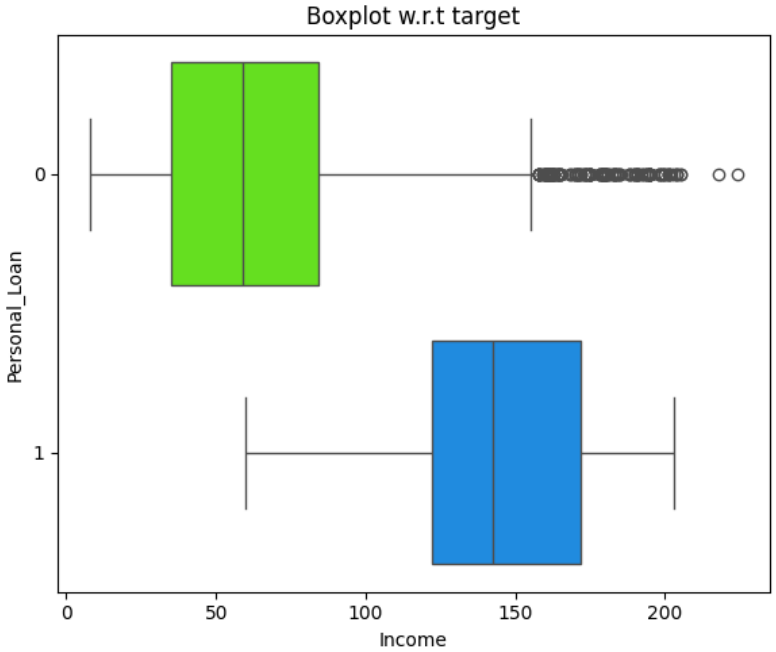


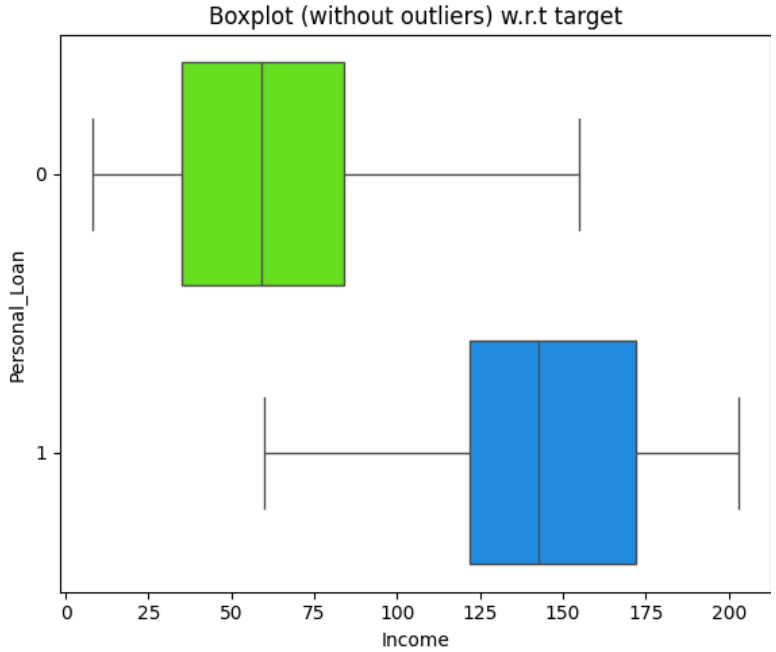

```
labeled_barplot(data, "CD_Account")   ## Complete the code t
```




- There seems to be a significantly higher proportion of personal loan borrowers that also have CD Accounts with the bank. However, it appears that a very small percentage of customers at the bank own CD accounts.
- We will examine the feature importance in our models.

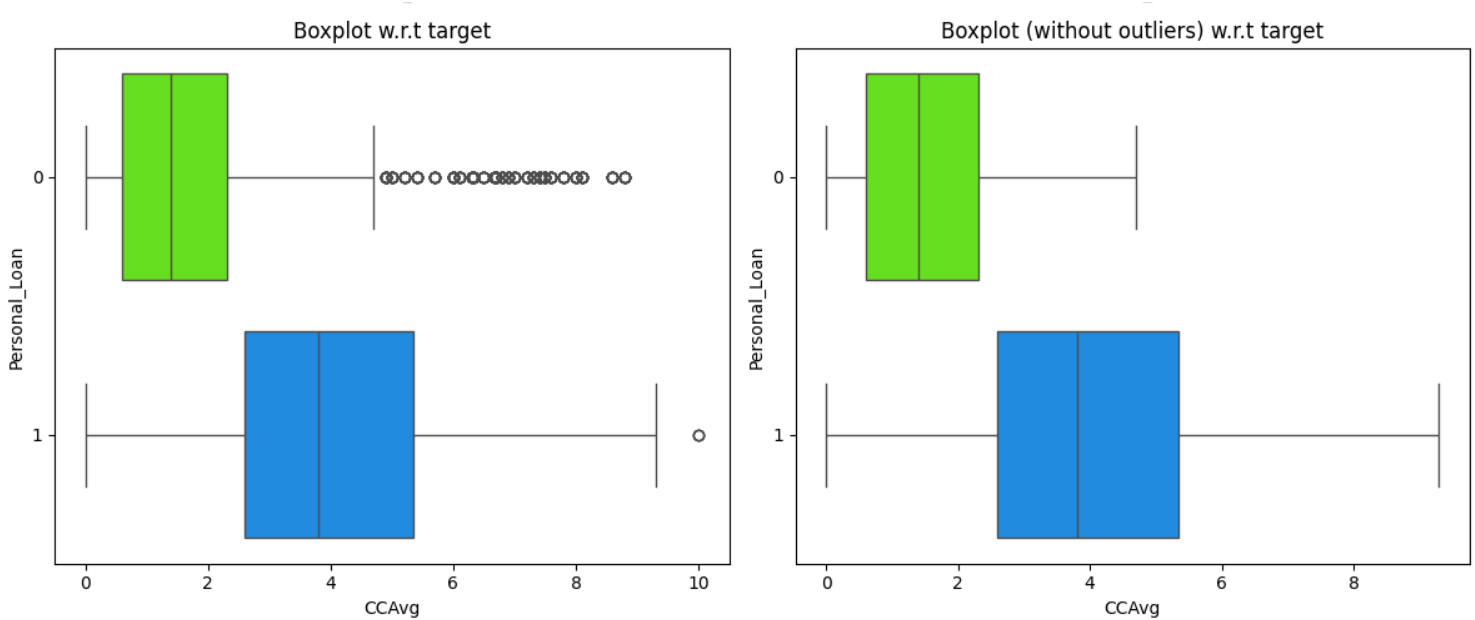
# EDA Results – Income




- Higher income individuals seem to, on average, take more personal loans.
- There are quite a few outliers in income that do not take personal loans, despite having a higher than average income. These customers might possibly just be averse to borrowing.
- One more variable we will look at when considering the feature importance of our models.

# EDA Results – CCAvg




- Those with higher average spending on credit cards are more likely to take personal loans
- There are quite a few outliers in income that do not take personal loans, despite having a higher than average monthly credit card spending.
- We will consider this in feature importance of our models.

# Data Preprocessing – Duplicate and Missing Values

- There were no found duplicate values. Every ID had a unique transaction row.
- No missing values in the dataset, but there were some values in the experience column that were negative. Instead of deleting these rows, we will assume they were data entry error and convert to absolute values to align with the rest of the data set.

# Data Preprocessing – Outlier Check & Feature Eng.

```
Age           0.00
Experience    0.00
Income        1.92
Family        0.00
CCAvg         6.48
Mortgage      5.82
dtype: float64
```

- These values represent the % of rows in the data set that contain outliers (defined as outside 1.5*IQR). Nothing major in any category. Highest % of outliers are in credit card average spend and if they have a mortgage, but the outliers are not too egregious to treat.

- **Feature engineering choices** – truncate zipcode to first two values. Change Education, Personal_Loan, Securities_Account, CD_Account, Online, CreditCard, And Zipcode from int64 variables into categories for our modeling purposes.

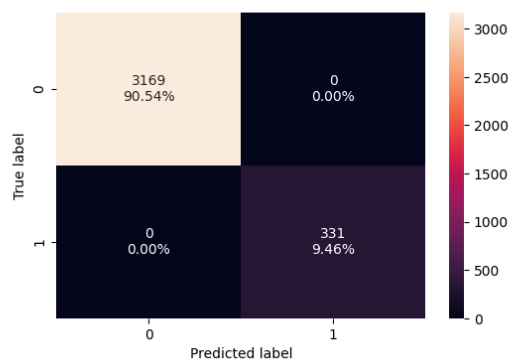
# Data Preprocessing – Data Preparation for Modeling

- To prepare data for modeling, we need to separate the independent variables (Mortgage, CCAvg, Family, Income, Age) from the dependent variable that we are looking to predict (Personal_Loan)
- We made a subjective decision to drop “Experience” from the data set, since Age is so highly correlated to it that we do not need to count the influence on our dataset twice.
- We create dummy variables for ZipCode And Education, resulting in the creation of additional rows of Boolean variables.
- We create a train/test split of 70% train/30% test for our first model. The first model will grow until each of the terminal nodes is pure, meaning contains 100% of each condition. It is understood that this will inevitably overfit the data.

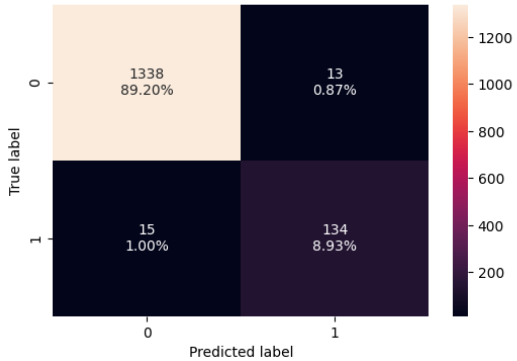
# Model Building – Decision Tree – Train v. Test Data

- In the initial decision tree model where the decision nodes continue until the all nodes are pure, the performance is perfect for the training data, however, that makes sense due to the overfitting of the data. Plotting performance data of the test data shows how accurately it fits. Recall is dropped by .11. Decent model, but can we improve it?

```
confusion_matrix_sklearn(model, X_train, y_train)
```




```
confusion_matrix_sklearn(model, X_test, y_test)  # Complete the code to get the confusion matrix for test data
```



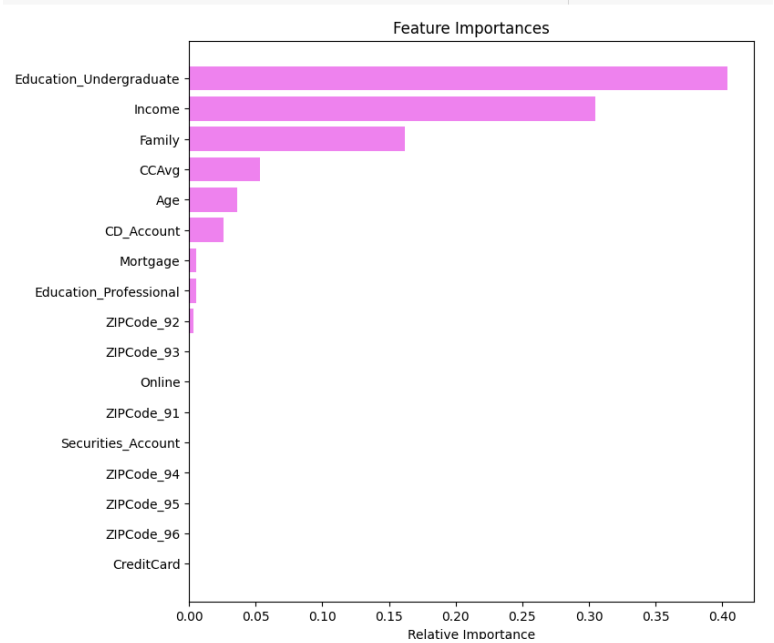

```
decision_tree_perf_train = model_performance_classification_sklearn(model, X_train, y_train)
decision_tree_perf_train
```

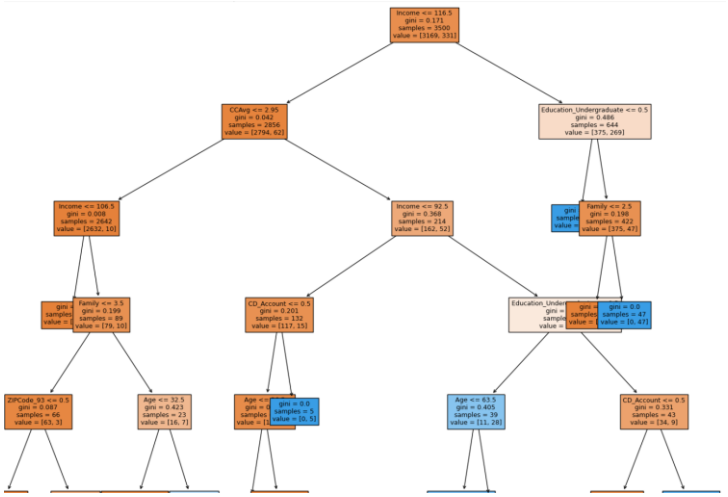
| | Accuracy | Recall | Precision | F1 |
|---|---|---|---|---|
| 0 | 1.0 | 1.0 | 1.0 | 1.0 |

```
# Complete the code to get the model formance on test data
decision_tree_perf_test = model_performance_classification_sklearn(model, X_test, y_test)
decision_tree_perf_test
```

| | Accuracy | Recall | Precision | F1 |
|---|---|---|---|---|
| 0 | 0.981333 | 0.899329 | 0.911565 | 0.905405 |

# Model Building – Decision Tree – Feature Importance & Decision rules



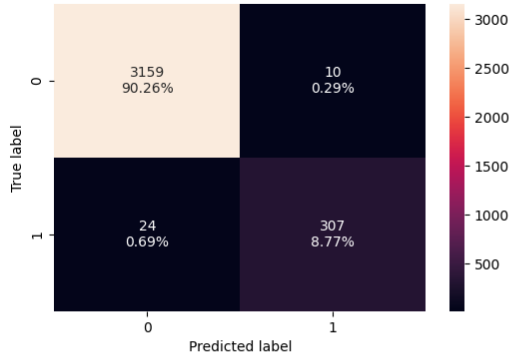



In this model, having an undergraduate degree, your income, number of members in your family, your average monthly credit card spend, and having a CD account were the most important features.
Only the top nodes are shown on the decision tree as it is very large.

# Model Building – Decision Tree – Pre-Pruning with Hyperparameters

- This time, we attempt to improve our model by pre-pruning the decision tree using hyperparameters. These are limitations in the Python code that are placed on requirements necessary to generate a subsequent node. If these requirements are not met, a new node is stopped. By doing this, we will be limiting the growth of the tree as it is currently being generated.
- Training data for the model (left) is very close to test data (right). Accuracy, precision, and F1 values of the test data are very similar to the trained data, however, there was a significant delta created for the Recall value, which we identified as the most important to model success for our situation. If we are inaccurate in this area, we will produce more false negatives than we want. Meaning we will miss opportunities to offer loans to people who would have taken them if offered, by assuming that they would not have.

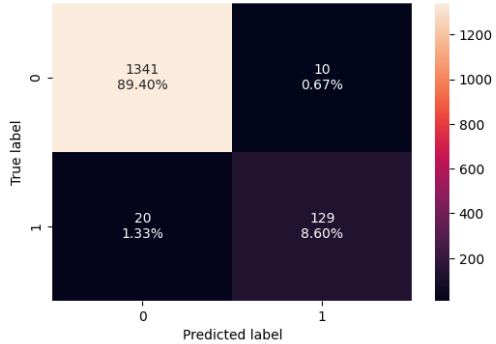
```
confusion_matrix_sklearn(estimator, X_train, y_train) ## Complete the code to create confusion matrix for train da
```




```
decision_tree_tune_perf_train = model_performance_classification_sklearn(estimator, X_train, y_train) ## Complete
decision_tree_tune_perf_train
```

| | Accuracy | Recall | Precision | F1 |
|---|---|---|---|---|
| 0 | 0.990286 | 0.927492 | 0.968454 | 0.947531 |

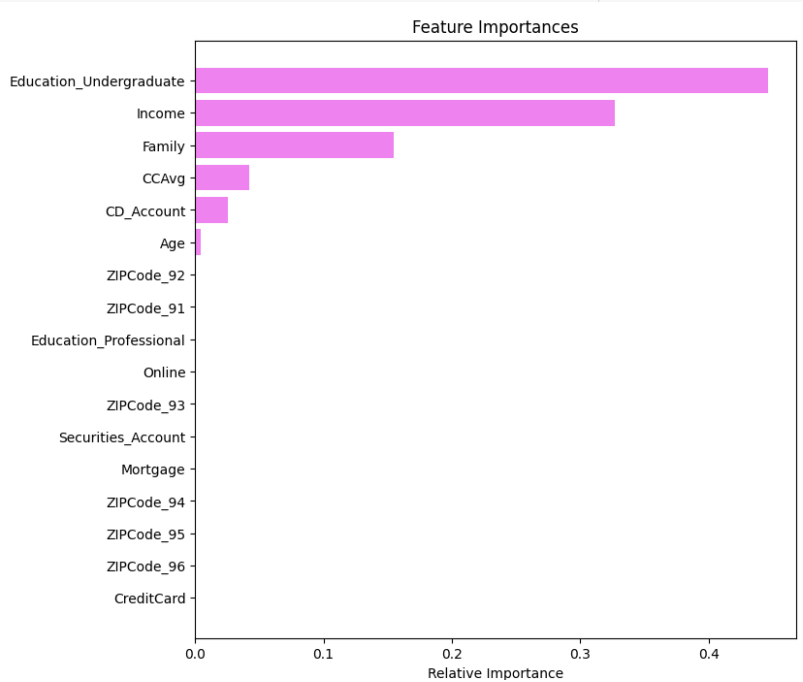
```
confusion_matrix_sklearn(estimator, X_test, y_test)  # Complete the code to get the c
```

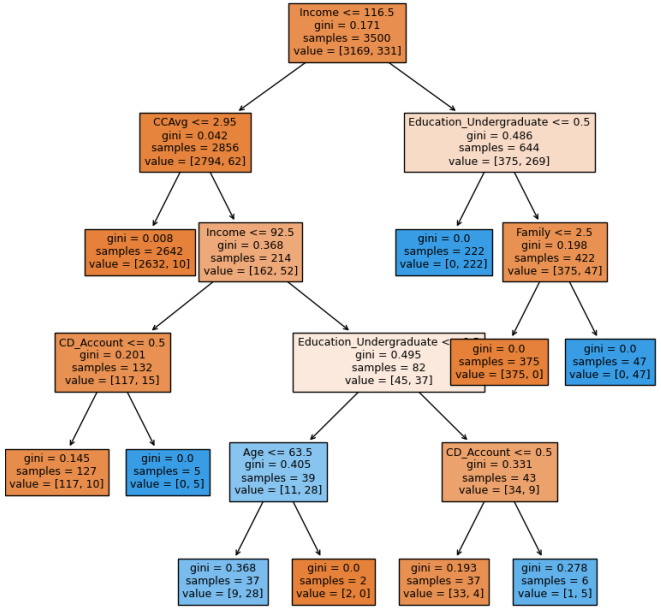



```
# Complete the code to get the model performance on test data
decision_tree_tune_perf_test = model_performance_classification_sklearn(estimator, X_
decision_tree_tune_perf_test
```

| | Accuracy | Recall | Precision | F1 |
|---|---|---|---|---|
| 0 | 0.98 | 0.865772 | 0.928058 | 0.895833 |

# Model Building – Pre-Pruning – Feature Importance

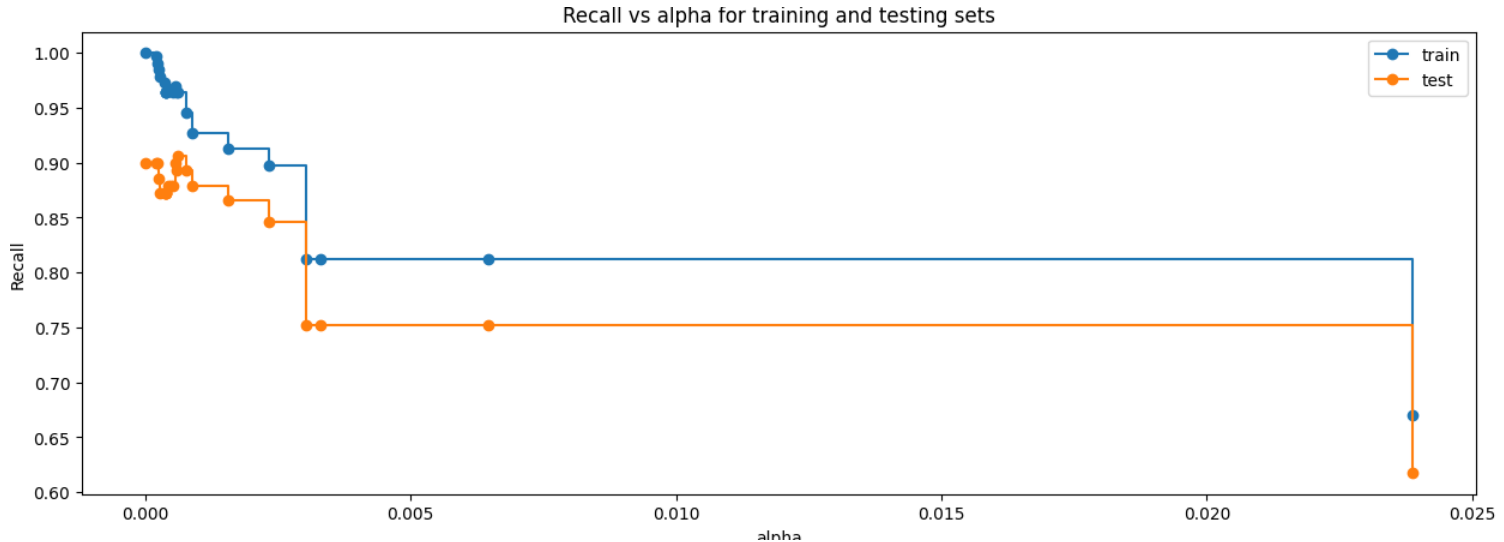



In this model, having an undergraduate degree, your income, the number of members in your family, your average monthly credit card spend, and having a CD account were the most important features, but slightly more muted than the initial decision tree model.
The decision tree is much simpler.

# Model Building – Cost Complexity Tuning

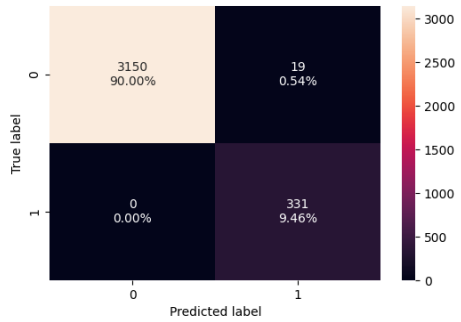
- In this model, we are going to prune the decision tree after it has already been created, which is known as Post-Pruning. The recall for training and test set are best between .006 and ~.024. We will choose our alpha to be .01.



# Model Building – Cost Complexity Tuning

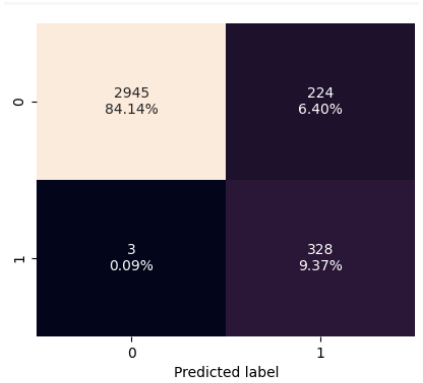
- Using alpha = .01 yields the following results. Recall of test set is .990, which is very good performance. This should be the recommended model to use for our purposes.

```
confusion_matrix_sklearn(estimator_2, X_train, y_train) ## Complet
```




```
decision_tree_tune_post_train = model_performance_classification_sk
decision_tree_tune_post_train
```

| | Accuracy | Recall | Precision | F1 |
|---|---|---|---|---|
| 0 | 0.994571 | 1.0 | 0.945714 | 0.9721 |

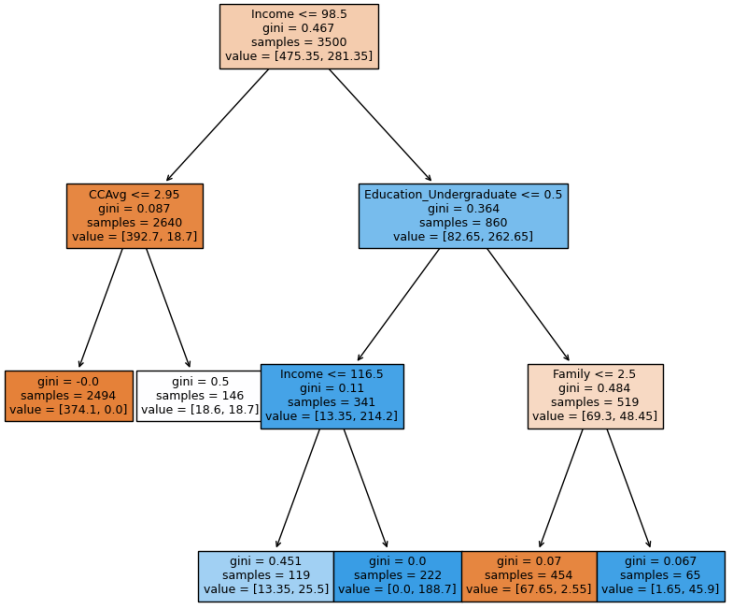



```
ision_tree_tune_post_train = model_performance_classi
ision_tree_tune_post_train
```

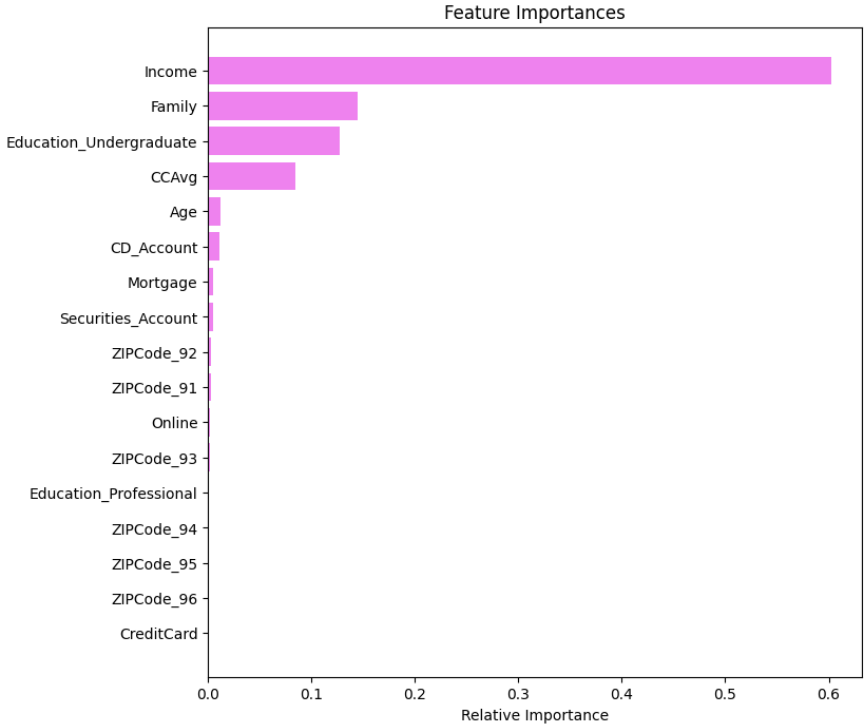
| Accuracy | Recall | Precision | F1 |
|---|---|---|---|
| 0.935143 | 0.990937 | 0.594203 | 0.742922 |

# Model Building – Cost Complexity Tuning

- Using alpha = .01 yields the following results. Recall of test set is .990, which is very good performance. The decision tree is much simpler looking as well.



# Model Building – Cost Complexity Tuning – Feature Importance




In the Post-Pruning Model, it seems that income has overtaken the top spot for feature importance, followed by family. Undergraduate education is less important compared to the decision tree and pre-pruning model. CCAvg is still important.

# Model Performance Summary

- We are evaluating our decision tree models with a preference for the model with the highest Recall value, to minimize false negatives.
- We have a decision tree model that grows until all leaves are pure, one that is pre-pruned using hyperparameters, and one that is post-pruned using cost-complexity and finding the idea ccp_alpahs value.
- In all 3 decision trees, Income, family, Undergraduate education, and monthly spending on credit card were the most important features. In the cost complexity post-pruned tree, the importance of undergraduate education was reduced.
- Comparing models with the training performance on the left and test performance on the right. It appears that the cost-complexity post-pruning model had the highest recall ability on the test set (.986), so this would be the tree that we recommend for AllBank.

Training performance comparison:

| | Decision Tree sklearn | Decision Tree (Pre-Pruning) | Decision Tree (Post-Pruning) | Decision Tree sklearn | Decision Tree (Pre-Pruning) | Decision Tree (Post-Pruning) |
|---|---|---|---|---|---|---|
| **Accuracy** | 1.0 | 0.990286 | 0.994571 | 0.981333 | 0.980000 | 0.939333 |
| **Recall** | 1.0 | 0.927492 | 1.000000 | 0.899329 | 0.865772 | 0.986577 |
| **Precision** | 1.0 | 0.968454 | 0.945714 | 0.911565 | 0.928058 | 0.622881 |
| **F1** | 1.0 | 0.947531 | 0.972100 | 0.905405 | 0.895833 | 0.763636 |

# APPENDIX

Great Learning

# Happy Learning !