Personal Loan Acceptance Prediction

Introduction to Machine Learning and Decision Trees

2024-12-29

Melissa Lindskär

Contents / Agenda

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

Executive Summary

Key Factors Influencing Loan Acceptance:

- **Income:** Higher income customers are more likely to accept loans. Marketing efforts should focus on customers with middle to high income levels.
- **Family Size:** Customers with larger families showed a higher likelihood of accepting loans. Tailoring campaigns to families with multiple dependents might increase acceptance rates.
- **Education:** Customers with advanced education (graduate or professional levels) are more likely to accept loans. Offering specialized loan packages or benefits tailored to this group could enhance engagement.
- **Credit Card Usage (CCAvg):** Higher credit card usage correlates with loan acceptance. Consider cross-promoting loans with credit card offers or benefits to this segment.

Customer Segmentation:

- Segment customers based on the identified key features (income, family size, education, credit card usage) and develop personalized marketing strategies.
- For low-income customers, emphasize smaller, short-term loans or flexible repayment options.

Model Optimization Recommendations:

- The **pre-pruned decision tree** provided the best balance between accuracy and generalization. Use this model for future predictions, ensuring a balance between model complexity and performance.
- Regularly update and retrain the model with fresh data to account for evolving customer behavior.

Targeting High-Value Customers:

- Focus on individuals with significant credit card activity and those within middle-to-high income brackets.
- Offer educational resources or financial counseling to underrepresented segments to increase loan acceptance rates.

Business Problem Overview and Solution Approach

Defining the problem

Financial institutions face challenges in identifying which customers are most likely to accept personal loans when offered. Misallocation of marketing resources can lead to unnecessary expenses and missed opportunities to engage high-potential customers. The problem is further complicated by the diverse characteristics of customers, making it difficult to predict loan acceptance with accuracy.

The key questions to address are:

- What are the key factors influencing customer decisions to accept a loan?
- How can we predict which customers are most likely to accept a loan offer?
- How can the institution optimize its marketing strategies based on customer profiles?

Business Problem Overview and Solution Approach

The solution approach / methodology

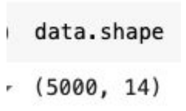
To address the problem, a systematic data-driven approach was adopted, ensuring a robust and interpretable model that balances predictive power with operational simplicity

1. **Data Collection and Understanding:**
   - A dataset of 5,000 customer records was analyzed, including features such as age, income, family size, education, credit card usage, and loan acceptance status.
   - Exploratory Data Analysis (EDA) was conducted to understand data distributions, correlations, and identify potential outliers.
2. **Data Preparation:**
   - Missing values were handled, and categorical variables (e.g., Education) were encoded using dummy variables.
   - Outliers were evaluated for their impact on model performance, with decisions made to retain them based on their potential significance.
3. **Feature Engineering:**
   - Created new features, such as "Mortgage-to-Income Ratio" and "CCAvg-to-Income Ratio," to capture meaningful relationships between variables.
   - Retained the most influential features (e.g., Income, Family, Education, and CCAvg) for model building.
4. **Model Building and Evaluation:**
   - A Decision Tree Classifier was chosen for its interpretability and suitability for binary classification tasks.
   - Three tree versions were evaluated:
     - **Default Tree:** Overfitted to the training data.
     - **Pre-Pruned Tree:** Optimized with hyperparameter tuning to balance complexity and performance.
     - **Post-Pruned Tree:** Pruned based on the Cost Complexity Pruning approach to reduce overfitting further.
   - Performance metrics such as Accuracy, Recall, Precision, and F1-score were used to evaluate the models on training and test sets.
5. **Insights and Recommendations:**
   - Feature importance analysis identified Income, Family Size, Education, and CCAvg as the most significant predictors of loan acceptance.
   - Actionable insights were provided for customer segmentation and marketing strategies.

EDA Results

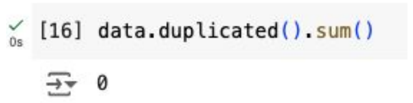
Dataset Dimensions:

- The dataset consist of **5000 rows** and **14 columns**
- This gives a substantial amount of data to analyze customer behavior and other factors that might be levers in the



```
data.shape
(5000, 14)
```

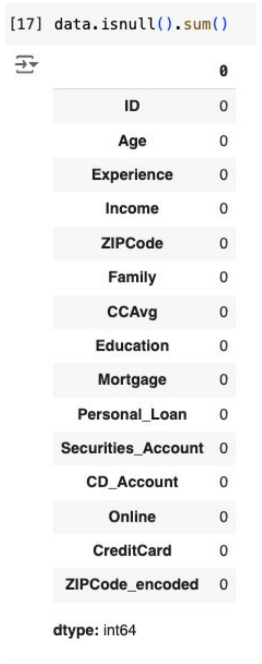
Duplicate value check - no duplicates



```
[16] data.duplicated().sum()
0
```

Datatypes of Columns

- The dataset includes only of numerical variables
- There are no missing values

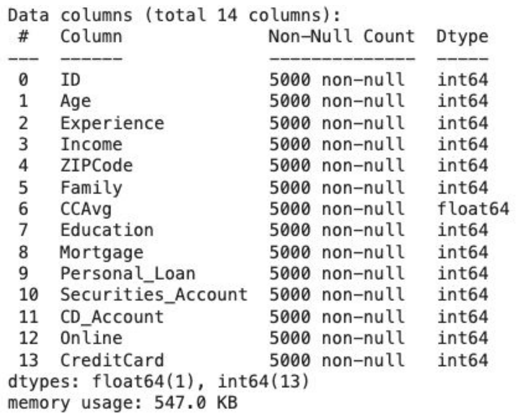


```
[17] data.isnull().sum()
```

	0
ID	0
Age	0
Experience	0
Income	0
ZIPCode	0
Family	0
CCAvg	0
Education	0
Mortgage	0
Personal_Loan	0
Securities_Account	0
CD_Account	0
Online	0
CreditCard	0
ZIPCode_encoded	0

dtype: int64

Missing value treatment - no missing values per se but “Experince” and “ZIPCode” need a deeper check



```
Data columns (total 14 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   ID                  5000 non-null   int64
 1   Age                 5000 non-null   int64
 2   Experience          5000 non-null   int64
 3   Income              5000 non-null   int64
 4   ZIPCode             5000 non-null   int64
 5   Family              5000 non-null   int64
 6   CCAvg               5000 non-null   float64
 7   Education           5000 non-null   int64
 8   Mortgage            5000 non-null   int64
 9   Personal_Loan       5000 non-null   int64
 10  Securities_Account  5000 non-null   int64
 11  CD_Account          5000 non-null   int64
 12  Online              5000 non-null   int64
 13  CreditCard          5000 non-null   int64
dtypes: float64(1), int64(13)
memory usage: 547.0 KB
```

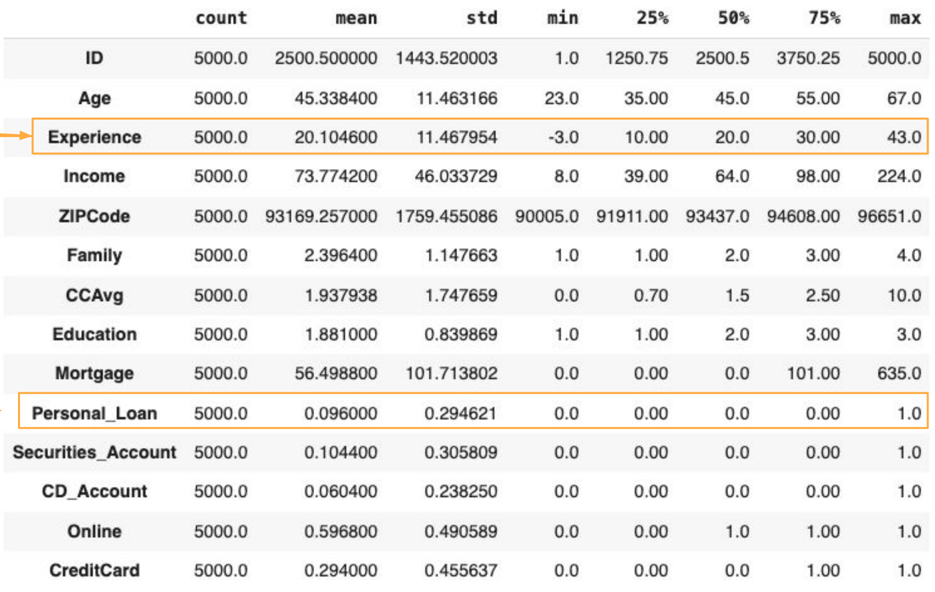
EDA Results - Statistical Summary

Negative values in "Experience":

- The Experience column has a minimum value of -3, which is illogical since years of experience cannot be negative.
- This may indicate a data entry error or an anomaly that needs to be addressed (e.g., replaced with the median or removed).

Imbalanced "Personal_Loan" column:

- The Personal_Loan column (0 = No, 1 = Yes) reveals that only 9.6% of customers accepted the loan.
- The dataset reflects the existing customer base, where only 9.6% of customers have accepted the loan. This low percentage makes sense because the bank currently has a much larger pool of liability customers compared to loan customers.
- While class imbalance is a challenge, not applying balancing techniques allows the model to maintain the dataset's real-world distribution. However, this choice emphasizes the importance of recall and F1-score over accuracy to ensure the model still identifies a substantial number of loan acceptors despite the imbalance.



	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

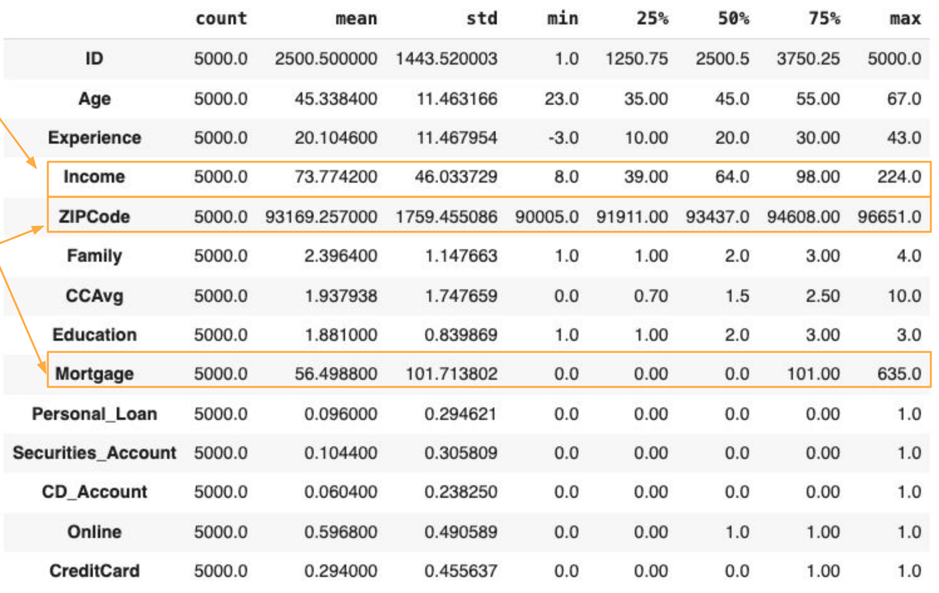
EDA Results - Statistical Summary

High standard deviation in "Income" and "Mortgage":

- Columns like Income and Mortgage show a high standard deviation, indicating a large spread in the values.
- For example, Income ranges from 8 to 224, and Mortgage ranges from 0 to 635. (This wide range might require scaling during model building)

"ZIPCode" as a non-informative numeric feature:

- The ZIPCode column appears numeric but is categorical in nature (it represents geographical locations).
- The mean (93169) and standard deviation are not meaningful here. This column may need to be transformed or dropped, depending on its correlation with the target variable.



	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

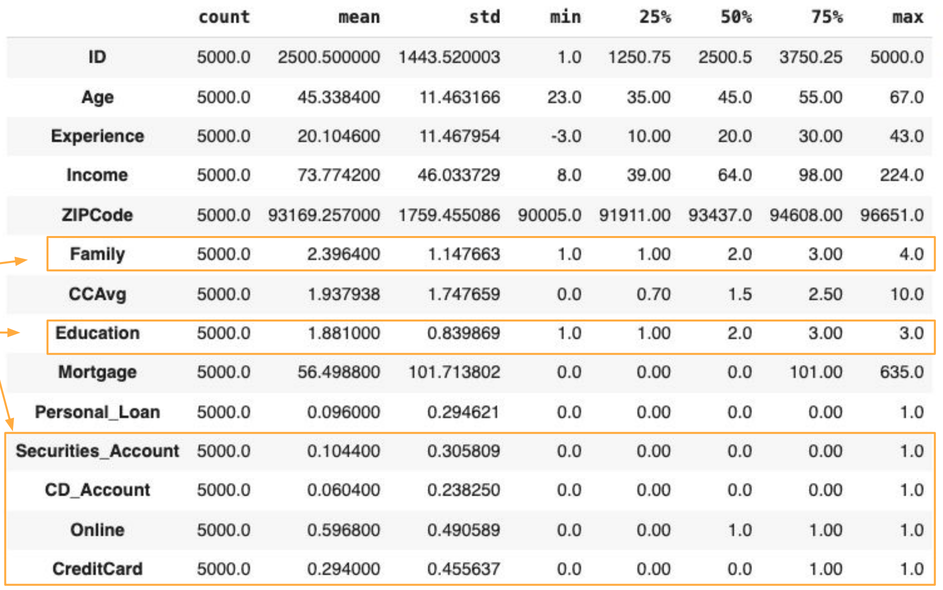
EDA Results - Statistical Summary

Binary features:

- Features such as Securities_Account, CD_Account, Online, and CreditCard are binary variables (0 or 1).
- It seems that most customers do not possess these accounts (e.g., CD_Account has only 6% with a value of 1).

"Education" and "Family":

- The Education column has only three levels (1, 2, 3), indicating an ordinal categorical variable.
- The Family column ranges from 1 to 4, with a median of 2, suggesting that most families are small.

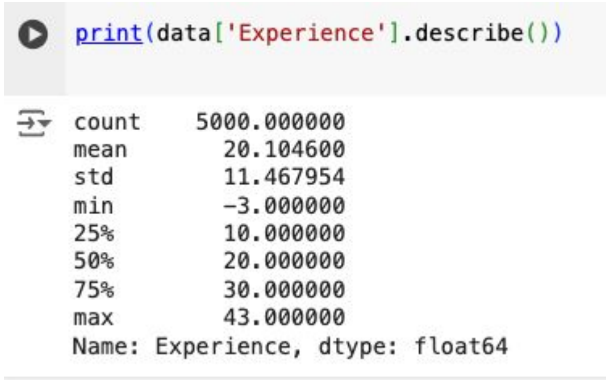


	count	mean	std	min	25%	50%	75%	max
ID	5000.0	2500.500000	1443.520003	1.0	1250.75	2500.5	3750.25	5000.0
Age	5000.0	45.338400	11.463166	23.0	35.00	45.0	55.00	67.0
Experience	5000.0	20.104600	11.467954	-3.0	10.00	20.0	30.00	43.0
Income	5000.0	73.774200	46.033729	8.0	39.00	64.0	98.00	224.0
ZIPCode	5000.0	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

Data Preprocessing

Experience check

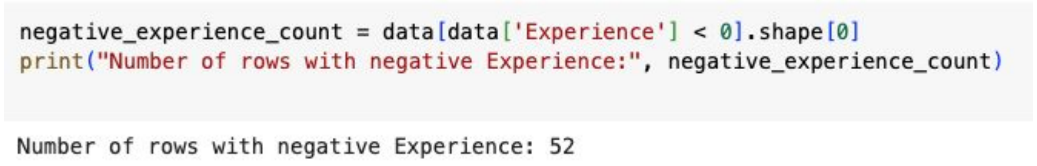
- Column “Experience” need a deeper check, -3 years does not make sense



```
print(data['Experience'].describe())
```

```
count    5000.000000
mean       20.104600
std        11.467954
min        -3.000000
25%        10.000000
50%        20.000000
75%        30.000000
max        43.000000
Name: Experience, dtype: float64
```

- There are 52 rows with negative professional experience.



```
negative_experience_count = data[data['Experience'] < 0].shape[0]
print("Number of rows with negative Experience:", negative_experience_count)
```

```
Number of rows with negative Experience: 52
```

Data Preprocessing - Choice of value treatment

Which approach to choose regarding Experience?

1. Replacing with 0, mean or median:

- By imputing negative values as `0`, would mean that saying that these customers have zero years of professional experience. This assumes that zero is a valid value, meaning the individual has no work experience, not that the value is missing.
- If the `Experience` column is important for predicting `Personal_Loan`, setting these values to zero could misrepresent the data and potentially skew results. For example:
  - If negative values indicate a data error, replacing with `0` might add noise to the model.
  - Zero might create an artificial correlation where none exists.

2. If negative values are frequent and one might suspect they indicate missing data:

- Replace them with the mean or median experience instead of 0, which would be a more neutral and representative imputation.
- This ensures we retain those rows without adding bias from a value like 0.

Data Preprocessing - Choice of value treatment

3. Dropping the Rows:

- If we drop the rows with negative values, we are removing potentially erroneous data points.
- This approach is safer **if** the number of rows with negative values is small, as it ensures the integrity of the data.
- However, dropping rows reduces the size of our dataset, which can:
    - Lead to loss of information.
    - Affect the generalizability of your model if too many rows are dropped.

4. Feature Engineering (Chosen Approach):

- Instead of imputing or dropping, **feature engineering** can retain data integrity while capturing additional insights:
    - **Flag the rows with zero or negative values:** Create binary indicator columns to represent these special cases.
    - Replace negative values with **0** for interpretability (indicating no experience).
    - Retain the context of negative values as a unique feature rather than discarding or overwriting them.
- This approach preserves the data while allowing the model to learn patterns associated with these flagged rows.
- Example:
    - Create a column, `Negative_Experience_Flag`, indicating whether the value was negative.
    - Retain rows and their potential signal, without making assumptions about data errors.
    -

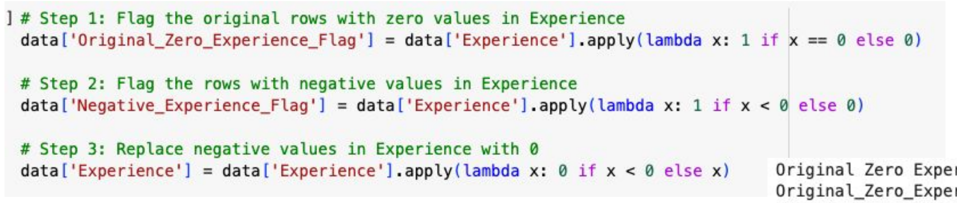
Data Preprocessing - Missing value treatment

Chosen approach - Feature Engineering

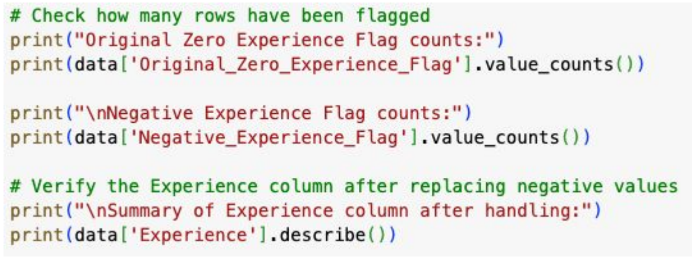
I chose to retain the negative values and perform **feature engineering** to add a new column. This approach allows us to preserve all the data and potentially gain additional insights.

Why Is This a Good Idea?

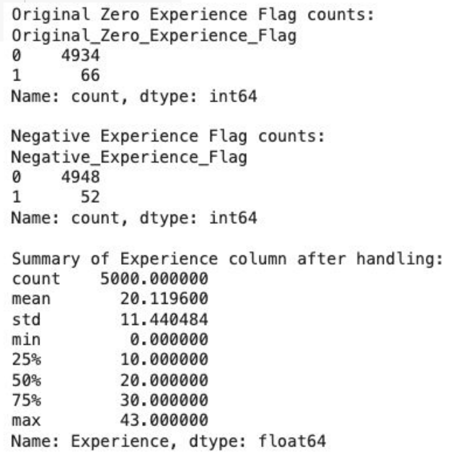
1. Retain Data Integrity: Dropping rows results in loss of information, which may not be ideal when data is limited.
2. Capture a Signal: Negative values might hold some underlying meaning (e.g., missing, incorrect entry, or special case). Engineering a new feature helps preserve that context.
3. Avoid Bias: Dropping data may introduce bias, especially if the rows with negative values share specific characteristics.



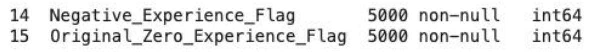
```
# Step 1: Flag the original rows with zero values in Experience
data['Original_Zero_Experience_Flag'] = data['Experience'].apply(lambda x: 1 if x == 0 else 0)

# Step 2: Flag the rows with negative values in Experience
data['Negative_Experience_Flag'] = data['Experience'].apply(lambda x: 1 if x < 0 else 0)

# Step 3: Replace negative values in Experience with 0
data['Experience'] = data['Experience'].apply(lambda x: 0 if x < 0 else x)
```



```
# Check how many rows have been flagged
print("Original Zero Experience Flag counts:")
print(data['Original_Zero_Experience_Flag'].value_counts())

print("\nNegative Experience Flag counts:")
print(data['Negative_Experience_Flag'].value_counts())

# Verify the Experience column after replacing negative values
print("\nSummary of Experience column after handling:")
print(data['Experience'].describe())
```



```
Original Zero Experience Flag counts:
Original_Zero_Experience_Flag
0    4934
1      66
Name: count, dtype: int64

Negative Experience Flag counts:
Negative_Experience_Flag
0    4948
1      52
Name: count, dtype: int64

Summary of Experience column after handling:
count    5000.000000
mean       20.119600
std        11.440484
min         0.000000
25%        10.000000
50%        20.000000
75%        30.000000
max        43.000000
Name: Experience, dtype: float64
```



```
14  Negative_Experience_Flag       5000 non-null   int64
15  Original_Zero_Experience_Flag  5000 non-null   int64
```

Data Preprocessing - Choice of value treatment ZIPCode

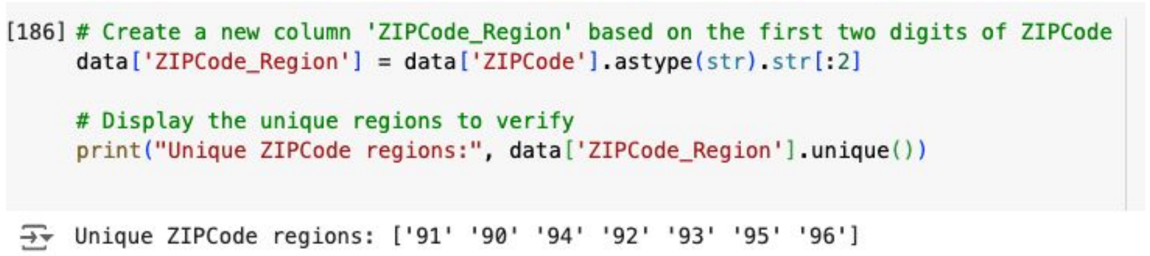
Chosen approach - Dropping

1) Categorize ZIP Codes Based on the first 2 digits:

- By grouping ZIP codes into regions (first two digits), we avoid the pitfalls of high-cardinality categorical data.
- If geographic location impacts customer behavior (loan acceptance), this method captures those patterns.

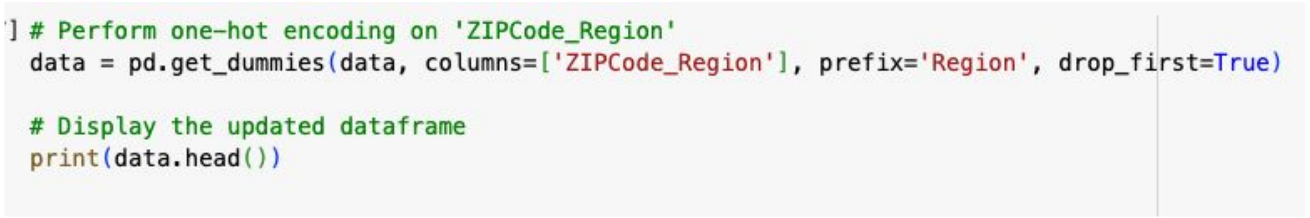
2) Perform One-Hot Encoding on the new Regions:

- One-hot encoding makes the column usable in machine learning models without introducing misleading numerical relationships



```
[186] # Create a new column 'ZIPCode_Region' based on the first two digits of ZIPCode
      data['ZIPCode_Region'] = data['ZIPCode'].astype(str).str[:2]

      # Display the unique regions to verify
      print("Unique ZIPCode regions:", data['ZIPCode_Region'].unique())
```

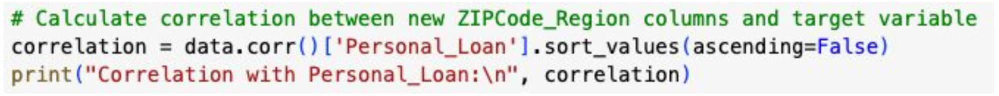
```
Unique ZIPCode regions: ['91' '90' '94' '92' '93' '95' '96']
```



```
] # Perform one-hot encoding on 'ZIPCode_Region'
  data = pd.get_dummies(data, columns=['ZIPCode_Region'], prefix='Region', drop_first=True)

  # Display the updated dataframe
  print(data.head())
```

Data Preprocessing - Choice of value treatment

3) Checking the correlation of Regions with Personal_Loan



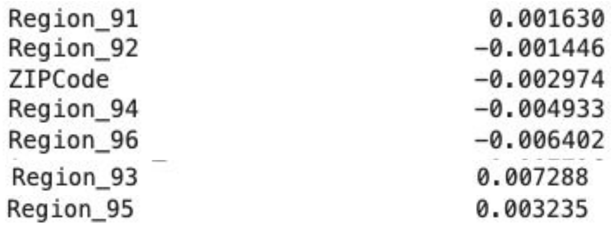
```
# Calculate correlation between new ZIPCode_Region columns and target variable
correlation = data.corr()['Personal_Loan'].sort_values(ascending=False)
print("Correlation with Personal_Loan:\n", correlation)
```

New ZIP Code Regions (e.g., Region_93, Region_95, etc.):

The correlations of these regions with Personal_Loan are extremely low (close to 0):

- Region_91 to 96: A range from -0.001 to 0.007

This indicates minimal influence of ZIP code regions on loan acceptance.



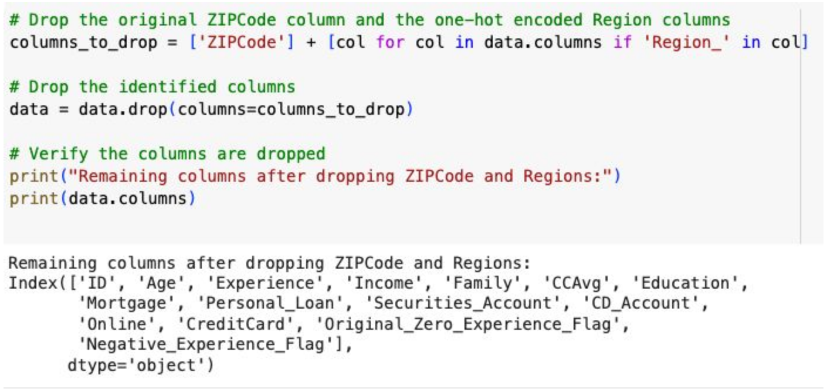
```
Region_91                     0.001630
Region_92                    -0.001446
ZIPCode                      -0.002974
Region_94                    -0.004933
Region_96                    -0.006402
Region_93                     0.007288
Region_95                     0.003235
```

Original ZIPCode and Region Columns:

- Original ZIPCode has a negative correlation (-0.002), confirming it has little predictive power when kept as is or encoded.
- The one-hot encoded Regions did not reveal meaningful patterns.

4) <u>**Dropping the column ZIPCode and the Regions**</u>

Dropping ZIPCode simplifies the dataset, reduces dimensionality and improves computational efficiency without compromising model accuracy, as ZIPCode does not add predictive value.



```
# Drop the original ZIPCode column and the one-hot encoded Region columns
columns_to_drop = ['ZIPCode'] + [col for col in data.columns if 'Region_' in col]

# Drop the identified columns
data = data.drop(columns=columns_to_drop)

# Verify the columns are dropped
print("Remaining columns after dropping ZIPCode and Regions:")
print(data.columns)
```

```
Remaining columns after dropping ZIPCode and Regions:
Index(['ID', 'Age', 'Experience', 'Income', 'Family', 'CCAvg', 'Education',
       'Mortgage', 'Personal_Loan', 'Securities_Account', 'CD_Account',
       'Online', 'CreditCard', 'Original_Zero_Experience_Flag',
       'Negative_Experience_Flag'],
      dtype='object')
```

Data Preprocessing - Dropping ID column

Advantages of Dropping "ID"

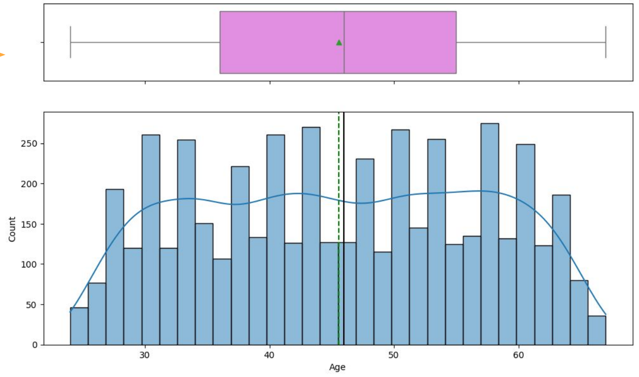
1. **No Predictive Power**:
    - The **ID column** is a unique identifier and carries no meaningful information to predict the target variable (*Personal_Loan* in this case).
    - It has no correlation with other variables and will not improve the model's performance.
2. **Reduces Noise and Complexity**:
    - Including irrelevant or unrelated variables introduces **noise** and adds unnecessary complexity to the model. Dropping "ID" makes the dataset cleaner and easier to process.
3. **Prevents Overfitting**:
    - In models like **Decision Trees**, the "ID" column might accidentally be used as a feature, leading to **overfitting**, since "ID" is unique to each row and cannot generalize.

At this stage there are 5000 rows and 14 columns. The rows ID and ZIPCode have been dropped and there are two new columns for the 0 experience and negative experience.

Univariate Analysis

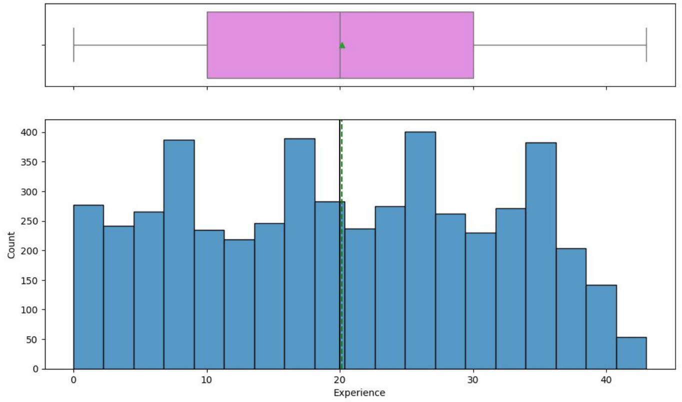
Age:

- The age distribution is even with a very slightly tendency toward right skewness.
- The bank's customers are evenly distributed across most age groups, with a concentration between 30 and 60 years, which suggest that marketing campaigns should target customers aged 30 to 60, as they represent the majority of the customer base.
- There are no clear outliers in the data.
- The mean and median are close to each other, suggesting that the data is relatively balanced but not perfectly symmetric.



Experience:

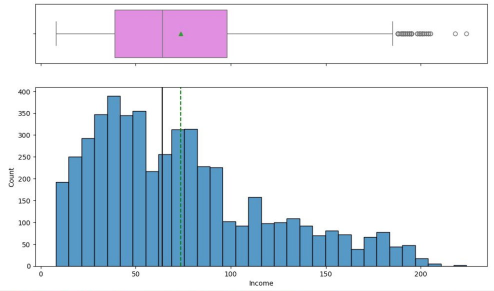
- The distribution of experience is balanced, with the majority of customers having between 10 and 30 years of experience.
- There are no visible outliers.
- The notable group of customers with 0 years of experience might need further exploration, as this could represent new customers, students, or missing data previously imputed as 0.
- The lack of skewness indicates that no transformation is necessary for this variable



Univariate Analysis

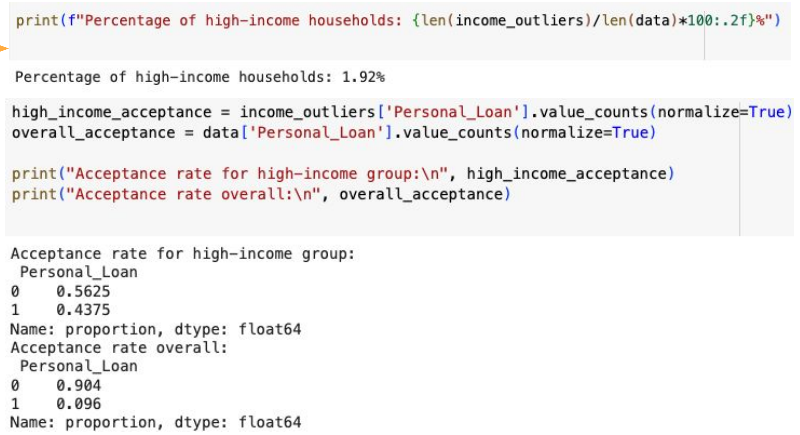
Income:

- Skewness: The income data is not normally distributed due to the right skew, primarily driven by a few high-income outliers.
- Outliers: The outliers (incomes > 200k dollars) may not need removal if they represent valid customers, as these individuals could potentially be important for loan decisions.
- The majority of customers fall into the lower-to-middle income brackets, which could influence how the bank targets customers for loans.



No treatment of outliers:

- The data set have a small percentage (1.92%) of high-income rows, which is realistic for a diverse customer base since banks often have a mix of low-income, middle-income, and high-income customers, and this diversity is reflected in the data.
- Outliers aren't always errors; they can represent meaningful patterns, such as; have a unique behaviors worth modeling or be key to the bank's strategy for offering loans. If removing them, there´s a risk of ignoring an important segment of the customer base.
- The high-income households are more likely to take personal loans (or have a different likelihood compared to the general population), removing them could bias the model, including these rows will help the model learn patterns associated with high-income loan acceptance, improving its predictive power for this segment.



```
print(f"Percentage of high-income households: {len(income_outliers)/len(data)*100:.2f}%")
```

```
Percentage of high-income households: 1.92%
```

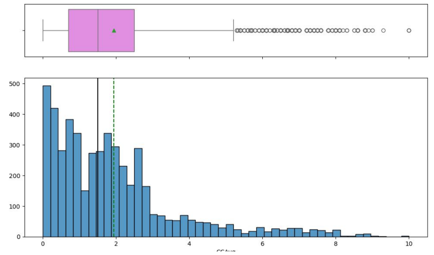
```
high_income_acceptance = income_outliers['Personal_Loan'].value_counts(normalize=True)
overall_acceptance = data['Personal_Loan'].value_counts(normalize=True)

print("Acceptance rate for high-income group:\n", high_income_acceptance)
print("Acceptance rate overall:\n", overall_acceptance)
```

```
Acceptance rate for high-income group:
 Personal_Loan
0    0.5625
1    0.4375
Name: proportion, dtype: float64
Acceptance rate overall:
 Personal_Loan
0    0.904
1    0.096
Name: proportion, dtype: float64
```

➔ 43.75% of high-income individuals accepted the loan. This is significantly higher than the overall dataset acceptance rate of 9.6%.
➔ 56.25% of high-income individuals rejected the loan. This rejection rate is much lower than the overall rejection rate of 90.4%.
➔ High-income customers are over 4 times more likely to accept loans compared to the general population (43.75% vs. 9.6%).

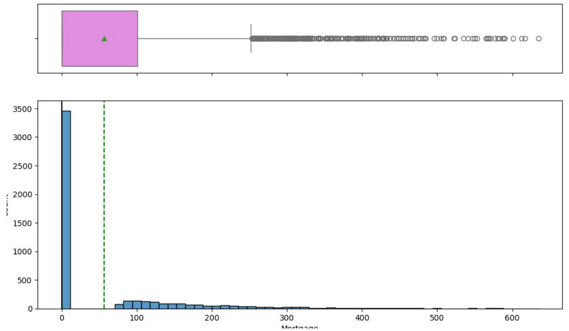
Univariate Analysis

CCAvg:



- The average credit card spending feature is positively skewed, with most customers spending less than $2,000 monthly.
- Outliers exist at higher spending levels, but they represent a small proportion of the dataset and do not significantly impact the distribution.
- The feature retains its continuous nature for analysis, as it provides meaningful granularity for capturing trends in loan acceptance

Mortgage:



- The majority of customers have a Mortgage value of 0, as evidenced by the high concentration of data at the leftmost bin of the histogram. This indicates that most customers either do not have a mortgage or it is not recorded.
- There are a significant number of outliers in the higher range of the Mortgage feature.
- For customers with non-zero mortgage values, the distribution is sparse and spread out, with very few customers having mortgages greater than 300 .
- The calculated correlation value is 0.1421, which indicates a weak positive relationship between mortgage and loan acceptance. This suggests that while there is some relationship, mortgage alone might not be a strong predictor of personal loan acceptance.
- Higher mortgage customers have a clear trend of higher acceptance rates, which might contribute useful insights to the model.

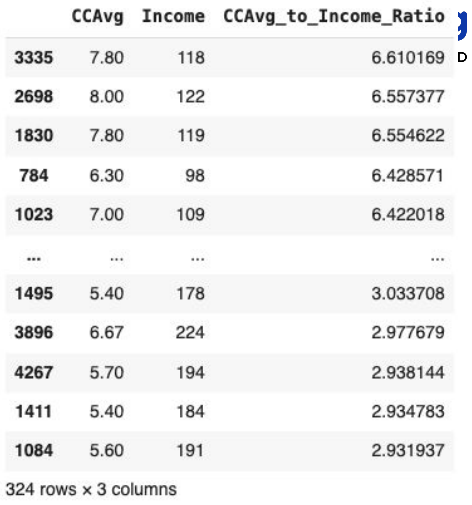
Outlier treatment - CCAvg

Chosen Approach: Retaining Outliers in the CCAvg Variable

Number of rows with high CCAvg: 324 Outliers in the CCAvg variable account for **6,48% of the data**.

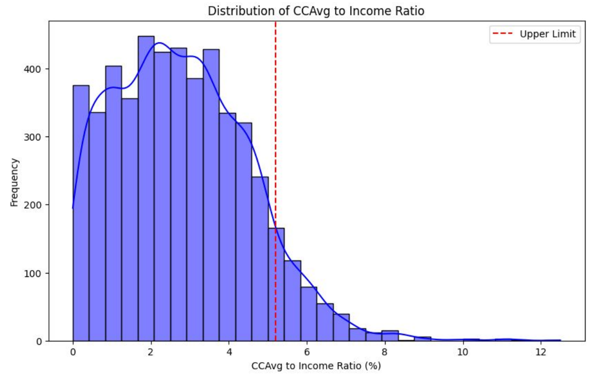
Reasons for Keeping the Outliers:

- **Number of Outliers**: There are 324 rows classified as having high credit card average (CCAvg) expenditures, representing outliers according to the defined upper limit.
- **Magnitude of Values**: These values range significantly higher than the majority of the dataset, highlighting individuals with unusually high monthly credit card spending.
- **Distribution**: The distribution of the CCAvg to Income Ratio shows a clear skew with a long tail toward higher percentages. Most ratios fall within a reasonable range, but there is a small subset of customers where credit card expenditures are disproportionately high compared to their income.
- **Outliers**: Values beyond the red dashed line (upper limit) suggest that these customers allocate an unusually large portion of their income to credit card spending. For instance:
    - Ratios above 6% could be considered financially unusual or high-risk.
- **Potential Customer Segments**: Customers with high CCAvg values might represent a unique segment:
    - **High-Income Individuals**: Some customers with a high CCAvg also have high income, suggesting they might be premium or affluent customers who spend heavily on credit cards.
    - **Financial Stress**: Others with a high CCAvg to Income Ratio may indicate financial strain or risky financial behavior.
- **Possible Anomalies**: Some of these extreme values might be data errors or anomalies, but without additional context, it's difficult to determine definitively.
- To understand customer behavior or segmentation, keeping these high CCAvg cases could provide insights into specific high-spending or high-risk groups.



	CCAvg	Income	CCAvg_to_Income_Ratio
3335	7.80	118	6.610169
2698	8.00	122	6.557377
1830	7.80	119	6.554622
784	6.30	98	6.428571
1023	7.00	109	6.422018
...	...	...	...
1495	5.40	178	3.033708
3896	6.67	224	2.977679
4267	5.70	194	2.938144
1411	5.40	184	2.934783
1084	5.60	191	2.931937

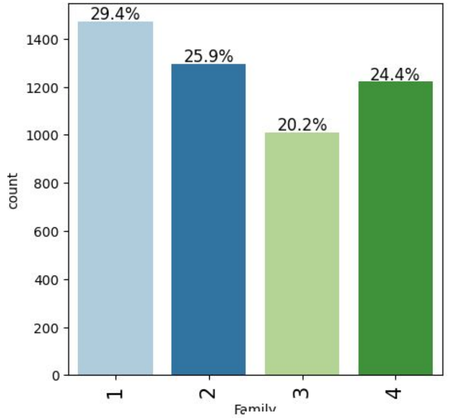
324 rows × 3 columns



Univariate Analysis

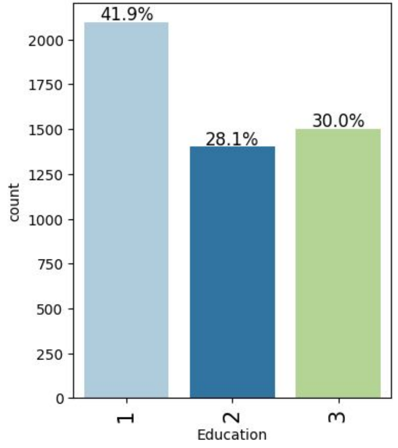
Family:

- Family size "1" is the most common, comprising 29.4% of the dataset. This might indicate a significant portion of single or individual households.
- Family sizes "2" and "4" are relatively close, making up 25.9% and 24.4% of the dataset respectively.
- Family size "3" is the least common, representing 20.2% of the dataset.



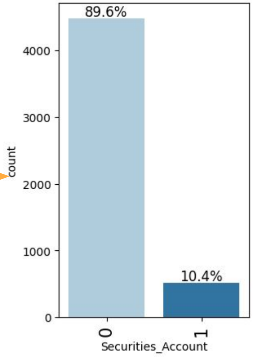
Education:

- Education level "1" (Undergrad) has the largest proportion, accounting for 41.9% of the dataset. This suggests that most of the customers are undergraduates.
- Education level "3" (Advanced/Professional) follows with 30.0%, indicating a significant portion of highly educated individuals.
- Education level "2" (Graduate) is the smallest group, representing 28.1% of the dataset.



Securities Account

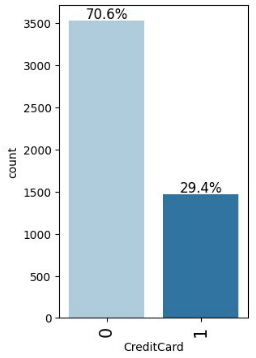
- The majority of customers (89.6%) do not have a Securities Account (0), indicating that most customers likely have simpler financial relationships with the bank.
- A smaller group (10.4%) has a Securities Account (1), which could represent more financially sophisticated or wealthier customers



Univariate Analysis

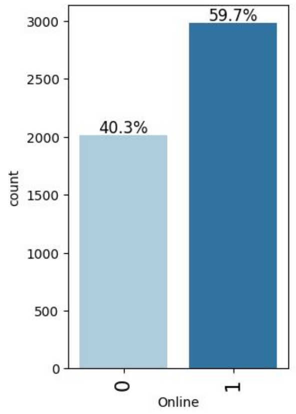
Credit Card:

- The majority of customers (70.6%) do not have credit cards from other banks, which could indicate that many customers are reliant on AllLife Bank for their financial services.
- Customers with credit cards from other banks (29.4%) might display different financial behavior compared to those without, such as higher credit usage or different spending patterns. This might influence their likelihood of accepting a personal loan.



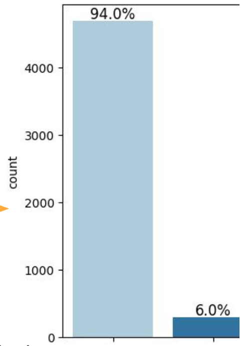
Online

- 40.3% of the customers do not use online banking (0), suggesting there is still a significant portion of customers who might rely on traditional banking methods or have no access to online services
- The higher proportion of online banking users (59.7%) indicates a strong shift towards digital banking, which aligns with broader industry trends.
- It would be useful to analyze whether online banking users have a higher acceptance rate for Personal Loans, as they may have more frequent interactions with the bank and access to its digital offers.
- This variable might serve as a valuable predictor in the model.



CD Account

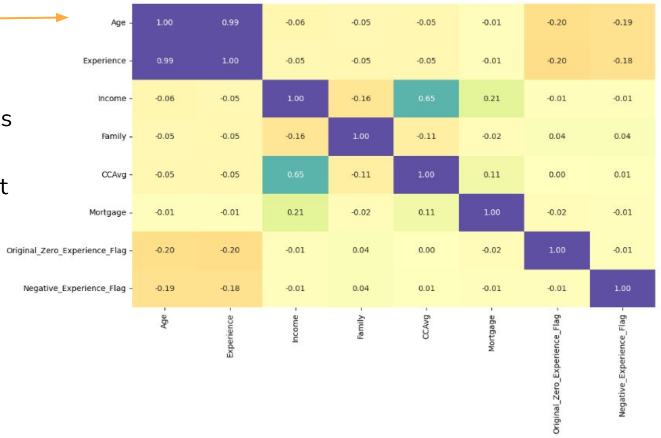
- A large majority (94.0%) of customers do not have a CD Account (0), indicating that most customers may not prefer or have access to such fixed deposit products.
- Only a small fraction (6.0%) of customers have a CD Account (1), which could represent a niche group of customers with different financial preferences or needs.



Multivariate Analysis

Analysis of the heatmap

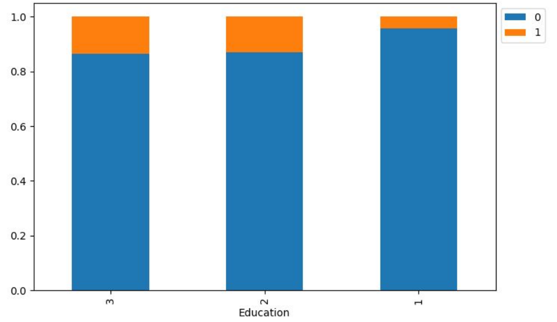
- The heatmap indicates that Age and Experience are almost linearly related. They might carry redundant information. One should be dropped to avoid multicollinearity.
- Higher Income tends to align with higher CCAvg (credit card average spending). This makes sense since individuals with higher incomes might spend more on credit cards.
- Features like Mortgage, Family, and Negative_Experience_Flag show weak or no significant correlation with other variables. This might indicate they are independent contributors to the dataset.
- The engineered flags have weak correlations with other variables, which is expected since they were derived from the Experience column.
- Now its clear that dropping Experience and engineered flags are beneficial for not overfitting the model



Insights from bar plot Education level and Personal Loan

- Education level appears to significantly influence the likelihood of purchasing a personal loan.
- Customers with a higher education level (3: Advanced/Professional) are more inclined to accept personal loans.
- Targeting campaigns towards customers with higher education levels (Levels 2 and 3) might yield better results for the bank.

```
Personal_Loan     0    1   All
Education
All            4520  480  5000
3              1296  205  1501
2              1221  182  1403
1              2003   93  2096
------------------------------------------------------------------------------------------------------------------------
```

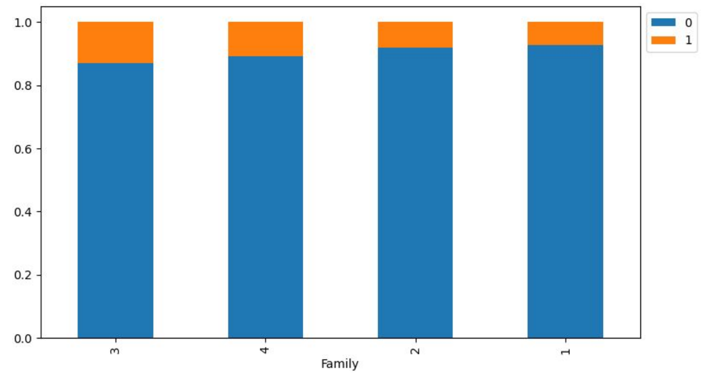


Multivariate Analysis

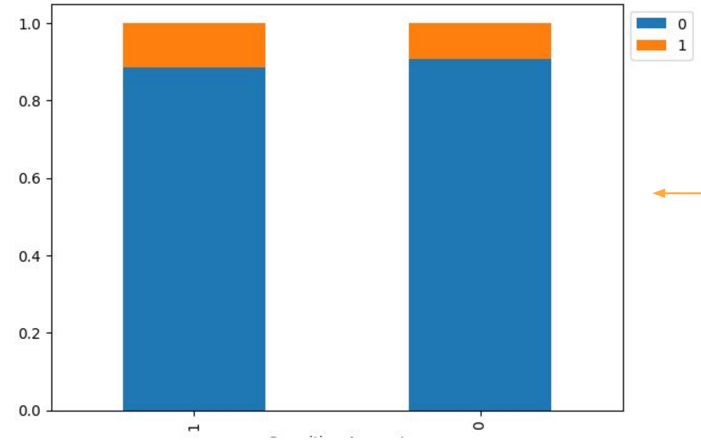
Insights from bar plot Family and Personal Loan

- Family size seems to have a mild correlation with personal loan acceptance, with larger families (3 and 4 members) being more likely to accept a loan.
- Family sizes 1 and 2 have similar lower acceptance rates, suggesting they may not be as responsive to marketing campaigns targeting loans.
- This trend might indicate that families with more members have greater financial responsibilities, making them more likely to seek loans.

```
Family
All          4520  480  5000
4            1088  134  1222
3             877  133  1010
1            1365  107  1472
2            1190  106  1296
------------------------------------------------------------------------------------------------------------------------
```



```
Securities_Account
All                 4520  480  5000
0                   4058  420  4478
1                    462   60   522
------------------------------------------------------------------------------------------
```



Insights from bar plot Securities Account and Personal Loan

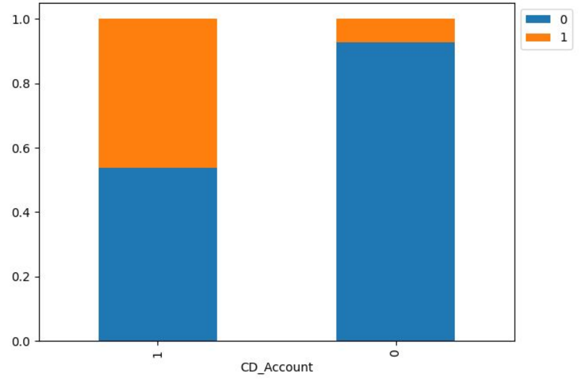
- Customers with a Securities Account are slightly more likely to accept a personal loan compared to those without one.
- However, the difference is not very pronounced, suggesting that Securities Account ownership may not be a strong indicator of personal loan acceptance.

Multivariate Analysis

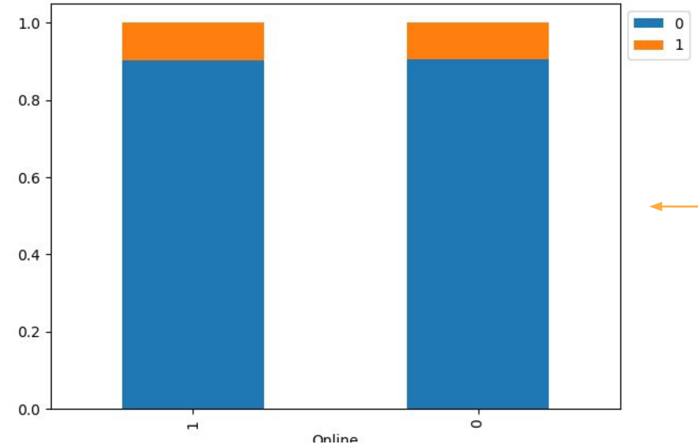
Insights from bar plot CD Account and Personal Loan

- There is a **strong association** between CD Account ownership and loan acceptance. Customers with a CD Account are significantly more likely to accept a personal loan compared to those without one.
- This suggests that owning a CD Account could be a key indicator for identifying potential personal loan customers.

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



```
Online
All            4520  480  5000
1              2693  291  2984
0              1827  189  2016
------------------------------------------------------------------------------------------------------------------------
```



Insights from bar plot Online banking and Personal Loan

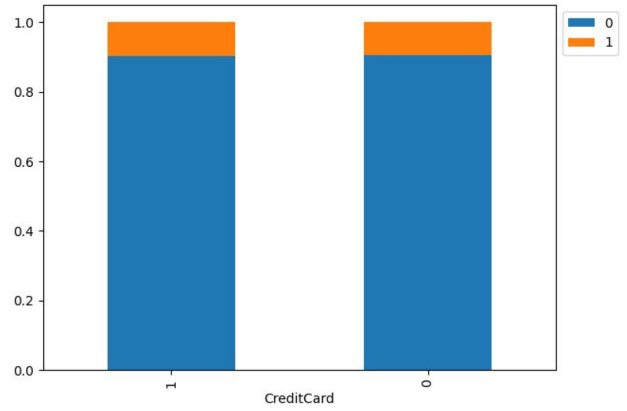
- The difference in loan acceptance rates between online banking users (~9.8%) and non-users (~9.4%) is minimal. This suggests that **online banking usage does not have a strong correlation with loan acceptance** in this dataset.
- However, the higher count of online banking users in the dataset makes them a larger group to potentially target for marketing campaigns.

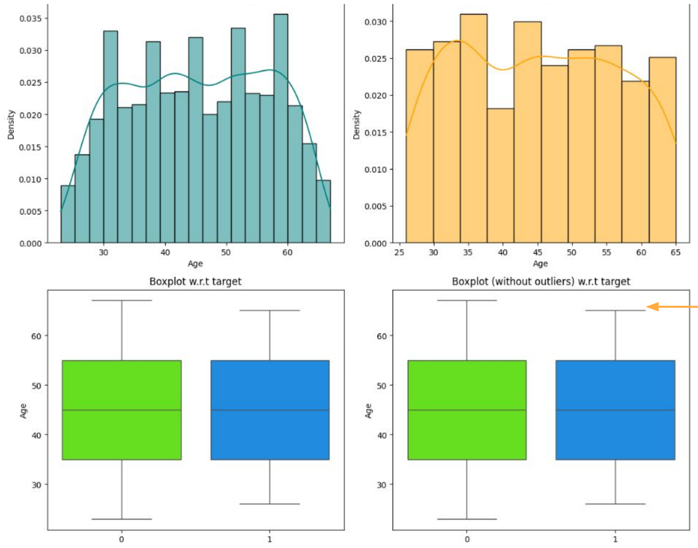
Multivariate Analysis

Insights from bar plot Credit Card and Personal Loan

- The loan acceptance rates are very similar between customers who have credit cards (~9.7%) and those who do not (~9.5%).
- This suggests that having a credit card issued by another bank does not have a strong impact on the likelihood of accepting a personal loan.

```
Personal_Loan     0    1   All
CreditCard
All            4520  480  5000
0              3193  337  3530
1              1327  143  1470
------------------------------------------------------------------------------------------------------------------------
```





Insights from Age and Personal Loan Distribution and boxplots

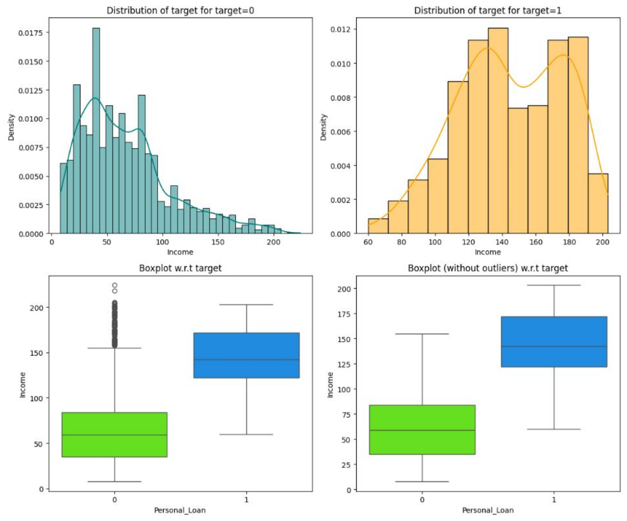
- While there is a noticeable peak in loan acceptance for individuals aged 35–45, age alone does not appear to be a dominant factor in determining loan acceptance.
- Other features in the dataset might interact with age to influence decisions. Combining age with other predictors during model training could reveal hidden patterns.

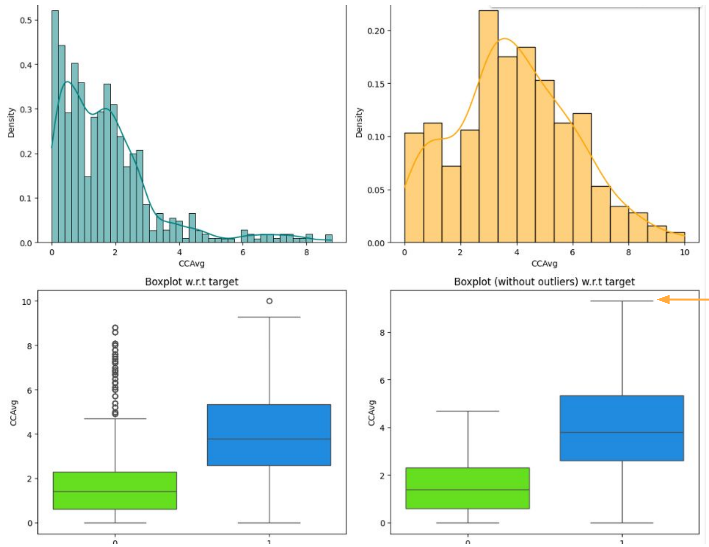
Multivariate Analysis

Insights from bar plot Income and Personal Loan

- Income appears to be a strong differentiator between customers who take personal loans and those who do not. Higher-income individuals are more likely to accept loans, while lower-income individuals tend to reject them.
- This suggests that income is a key predictive factor and should be carefully considered during model development.

Note: The data also shows that Income has a approximately **1.92%** of the values are outliers.





Insights from CCAvg and Personal Loan Distribution and boxplots

- CCAvg is a significant predictor of loan acceptance, as individuals with higher spending patterns are more inclined to take a loan.
- Further multivariate analysis could reveal if this behavior is consistent when other variables (like income or mortgage) are considered simultaneously.

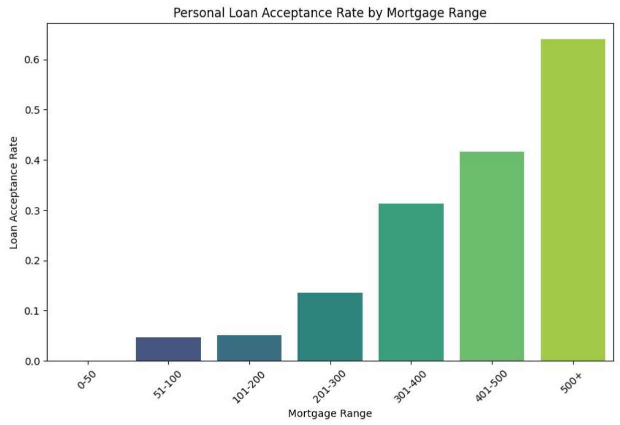
Note: The data also shows that **6.48%** of the values are outliers, indicating that spending varies significantly for some customers.

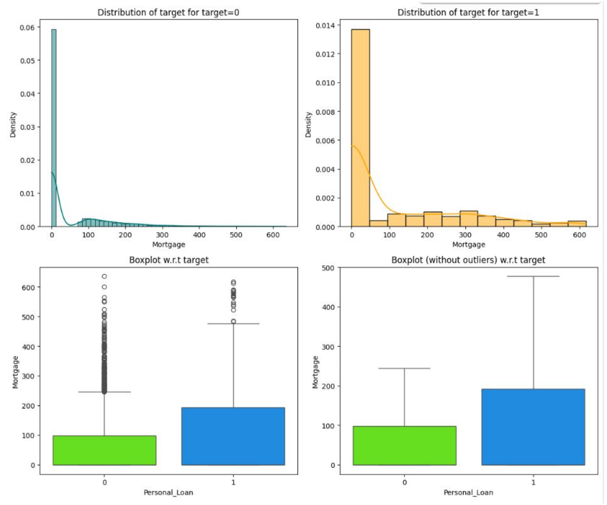
Mortgage vs Personal Loan

Mortgage range vs Personal Loan

Positive Trend: As the mortgage range increases, the acceptance rate for personal loans also increases. Customers with mortgages in the highest range (500+) have the highest loan acceptance rate (around 60%).

Low Acceptance for Smaller Mortgages: For mortgage ranges between 0-50 and 51-100, the acceptance rate is very low, nearly negligible. This suggests that customers with low or no mortgages are less likely to accept personal loans.





Insights from Mortgage and Personal Loan Distribution and boxplots

- Individuals with higher mortgages are more likely to take a personal loan. This could be because a higher mortgage might indicate greater financial needs or commitments.
- The concentration of mortgage values close to zero for target=0 suggests many individuals without a personal loan might have no mortgage obligations.

Outlier treatment - Mortgage

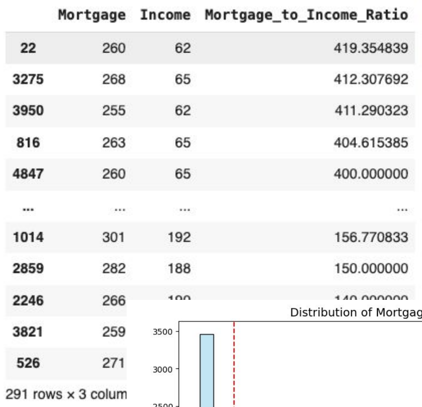
Chosen Approach: Retaining Outliers in the Mortgage Variable

Outliers in the Mortgage variable account for **5.82% of the data** and are retained as they are likely valid, represent high-mortgage customers, and show a correlation with loan acceptance.

Reasons for Keeping the Outliers:

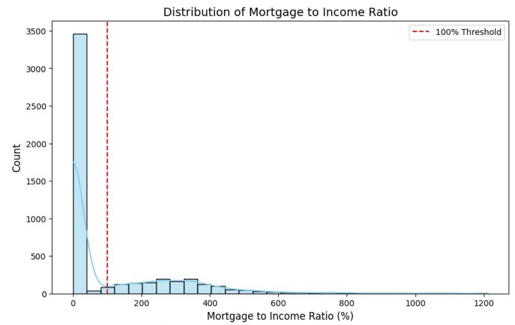
1. **Realistic Scenarios:**
   It is plausible for some individuals to have mortgage loans exceeding their reported annual income. For instance, wealthy individuals may use loans for investment purposes, which can explain a high Mortgage-to-Income Ratio.
2. **Unrealistic Scenarios:**
   While an outlier rate of 5.82% is above the typical threshold for rare cases (<5%), it is still manageable. This suggests that the extreme values are not excessively dominant, and their inclusion may provide valuable insights about high-mortgage customers.
3. **Impact on Analysis:**
   Analysis indicates that these high-mortgage values correlate with the target variable (Personal_Loan). Removing them could eliminate useful patterns, particularly since the dataset is limited in scope.
4. **Simplifying Complexity:**
   To avoid unnecessary complexity, the outliers are retained with the understanding that their realism cannot be fully verified, but they may represent unique and significant cases.

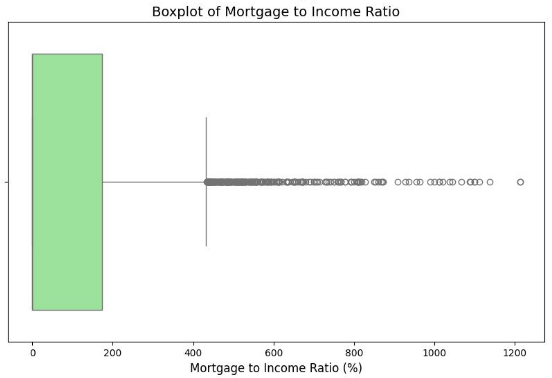
As of now there are 11 columns



	Mortgage	Income	Mortgage_to_Income_Ratio
22	260	62	419.354839
3275	268	65	412.307692
3950	255	62	411.290323
816	263	65	404.615385
4847	260	65	400.000000
...	...	...	...
1014	301	192	156.770833
2859	282	188	150.000000
2246	266	190	140.000000
3821	259		
526	271		

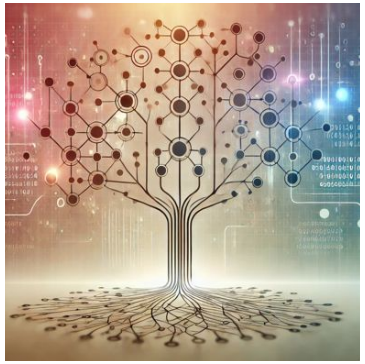
291 rows × 3 colum





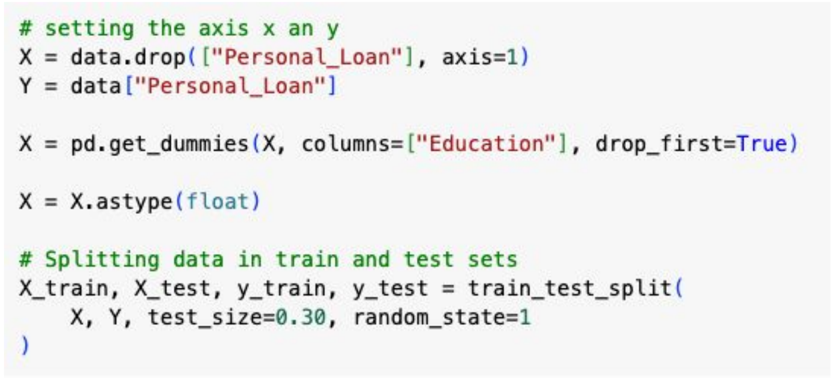
Model Building

Objective:
The goal is to create a classification model to predict whether a customer will accept a personal loan. This involves systematic data preparation, splitting, and evaluation to ensure accuracy and robustness.



Step 1: Data Preparation

- Features and Target: - Features (X):
  - All variables except "Personal_Loan"
  - Target (Y) "Personal_Loan" (1: Accepted, 0: Not Accepted).
- Encoding Categorical Variables:
  - One-hot encoding is applied to categorical variables like "Education" to prepare them for the model while preventing multicollinearity.
- Data Type Adjustment:
  - Ensuring all features are treated as floats for compatibility with machine learning algorithms.
- Train-Test Split:
  - Data is split into training (70%) and testing (30%) sets to train and evaluate the model. Example split: Training Set (3500 rows), Test Set (1500 rows).



```
# setting the axis x an y
X = data.drop(["Personal_Loan"], axis=1)
Y = data["Personal_Loan"]

X = pd.get_dummies(X, columns=["Education"], drop_first=True)

X = X.astype(float)

# Splitting data in train and test sets
X_train, X_test, y_train, y_test = train_test_split(
    X, Y, test_size=0.30, random_state=1
)
```

Model Building

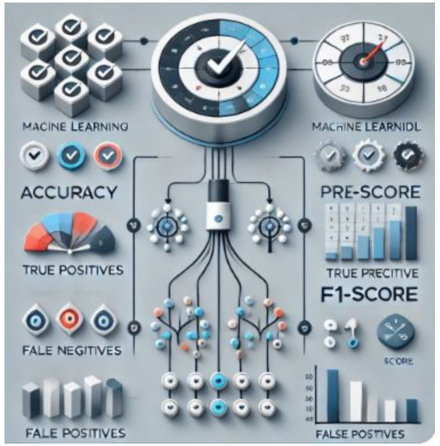
Step 2: Model Evaluation Criteria

- Evaluation Metrics:
  - Accuracy: Measures overall correctness.
  - Recall: Ability to identify true positives (important for minimizing missed opportunities).
  - Precision: Ability to avoid false positives. (important for saving marketing costs)
  - F1-Score: A balance between precision and recall.

These metrics ensure the model is both accurate and relevant for the business objective of identifying loan acceptance likelihood.

- Confusion Matrix:
  - Visual representation of predictions, showing the true positives, true negatives, false positives, and false negatives, helping identify areas for improvement.

Note: Class Imbalance and Class Weight:
In this project, class_weight was not applied despite the dataset being imbalanced, as the model's metrics (recall, precision, and F1-score) already demonstrated strong performance for both classes. This suggests that the model handled the imbalance effectively without additional adjustments. While balancing class weights can refine the tree’s focus on minority class predictions, it was not deemed necessary given the current results shown in coming slides and objectives of this analysis."



Model Building

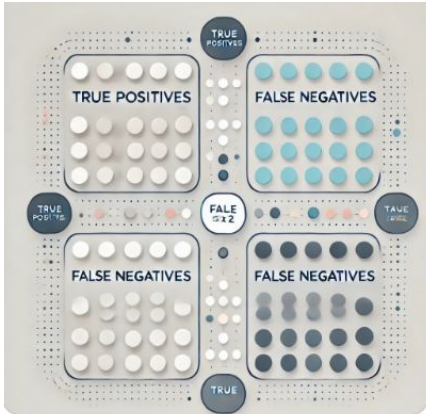
Step 3: Custom Functions for Reusability

1. Performance Metrics Function:
   - A reusable function computes and returns the model's accuracy, recall, precision, and F1-score in a concise table.

2. Confusion Matrix Visualization:
   - A function to generate a heatmap for confusion matrices, enabling clear analysis of prediction performance.

Why this approach?

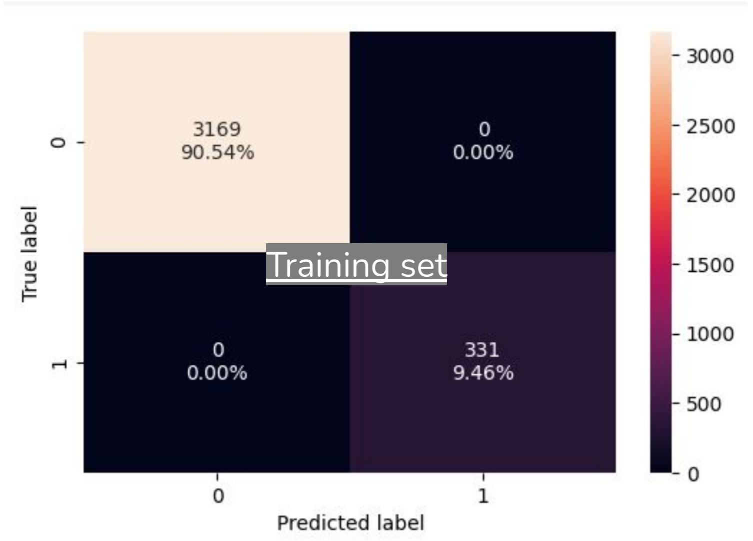
- Structured Process: Ensures clear steps for building and evaluating the model.
- Business Focus: Metrics and visuals align with decision-making needs.
- Scalability: Functions allow repeated testing of multiple models efficiently.



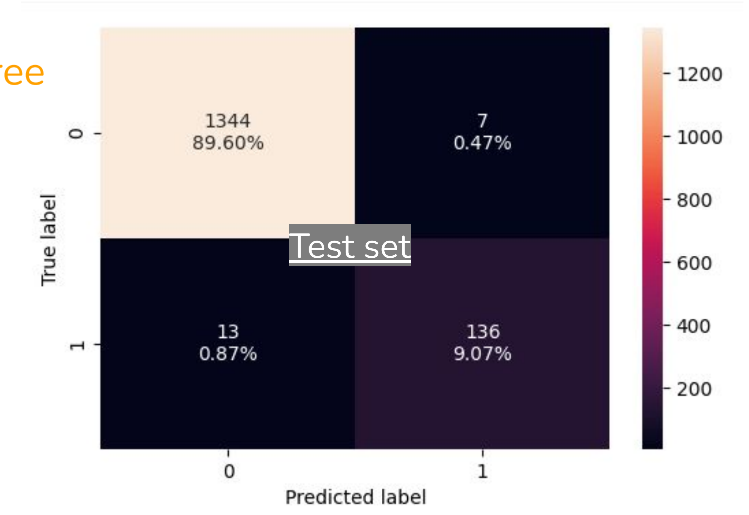
Model Building - Model performance (Default Tree)

To evaluate the decision tree models performance, I focused on metrics like **Accuracy**, **Precision**, **Recall**, and **F1-Score**. Recall and F1-score are essential for imbalanced datasets (explained in slide 7, EDA Results - Statistical Summary). The confusion matrix was used to visualize the classification results.

The evaluation on the test set ensures that the model generalizes well to unseen data, meeting the project's objective of accurately predicting personal loan acceptance while minimizing false negatives. This is critical for identifying high-potential customers without overwhelming the system with false leads.



Default tree



Model Building - Model performance (Default Tree)

1. Accuracy (98.67%): The model predicts the correct class (loan acceptance or not) 98.67% of the time on the test set.
2. Recall (91.28%): Out of all actual loan acceptances, the model successfully identifies 91.28%. This means it has a low rate of missing positive cases (false negatives).
3. Precision (95.10%): Out of all predicted loan acceptances, 95.10% are correct. This means the model rarely misclassifies non-loan acceptors as loan acceptors (false positives).
4. F1-Score (93.15%): The harmonic mean of Precision and Recall shows a strong balance between these two metrics, which is important in imbalanced datasets.

Test set

Metric	Value
Accuracy	98.68%
Recall	91.28%
Precision	95.10
F1-Score	93.15%

Comparison with Training Set:

Since the training set metrics were perfect (e.g., Accuracy, Recall, Precision, and F1 = 1.0), and the test set metrics are very high but slightly lower, it suggests that:

- The model has generalized well to unseen data, as there is no significant drop in performance.
- The model is not overfitting (due to good performance on the test set), which is a common issue when training accuracy is perfect. But to ensure its robustness further evaluation and other models needs to be considered and tested

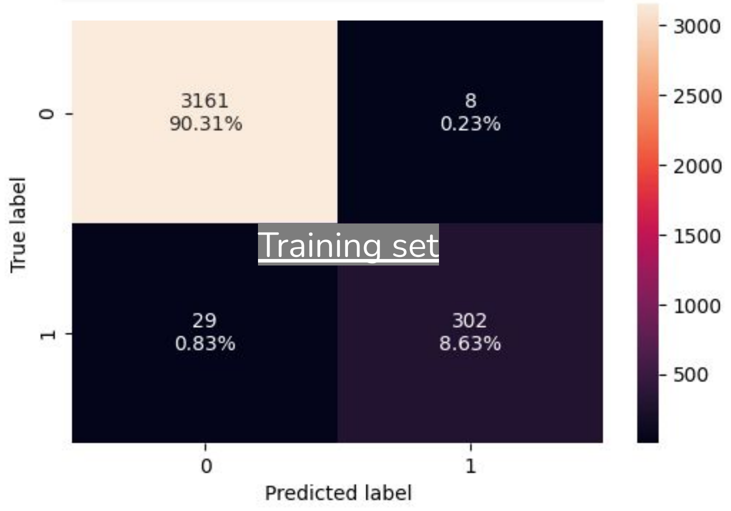
Training set

Metric	Value
Accuracy	100%
Recall	100%
Precision	100%
F1-Score	100%

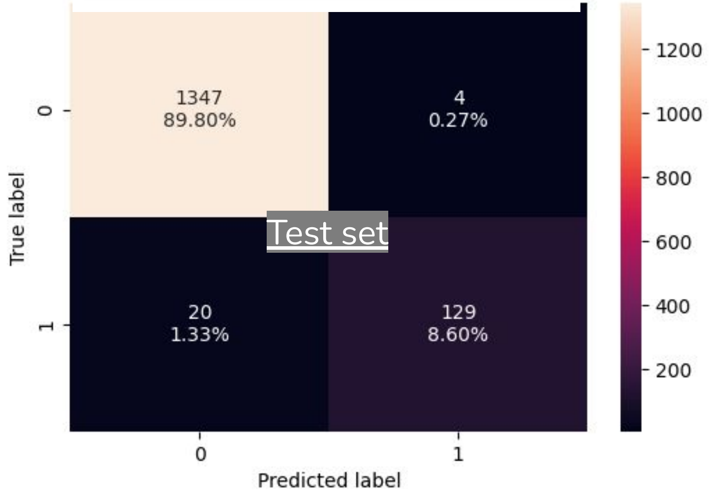
Model Building - Model performance (Pre Pruned)

Pre Pruned tree

	Accuracy	Recall	Precision	F1
0	0.989429	0.912387	0.974194	0.942278



	Accuracy	Recall	Precision	F1
0	0.984	0.865772	0.969925	0.914894



Model Building - Model performance (Pre Pruned)

1. Accuracy (98.4%): The overall correctness of the model in predicting loan acceptance or rejection. The model was right 98.4% of the time on the test set.
2. Recall (86.5%): Out of all actual loan acceptances, the model successfully identifies 86.5%. This means it has a low rate of missing positive cases (false negatives). The pre pruned tree did less good on recall then the default tree. High recall ensures that most potential loan acceptors are identified, minimizing missed opportunities.
3. Precision (96.9%): Out of all predicted loan acceptances, 96.9% are correct. This means the model rarely misclassifies non-loan acceptors as loan acceptors (false positives). Precision ensures that false positives are rare, meaning customers predicted to accept loans are likely to do so, saving marketing costs.
4. F1-Score (91.4%): The harmonic mean of Precision and Recall shows a strong balance between these two metrics. This metric is especially important in imbalanced datasets like ours, where only 9.6% of customers accept loans.

<u>Test set</u>

Metric	Value
Accuracy	98.4%
Recall	86.5%
Precision	96.9%
F1-Score	91.4%

Comparison with Training Set:

Since the training set metrics were close to perfect (e.g., Accuracy, Recall, Precision, and F1 = between 91.2% - 98.9%), and the test set metrics are very high but slightly lower, it suggests that:

- <u>Balances precision and recall effectively, making it suitable for customer targeting.</u>
- <u>By Pruning the tree we hurt the performance but also protected it from being overfit</u>
- <u>The model has generalized well to unseen data, as there is no significant drop in performance.</u>
- <u>The model is not overfitting, which is a common issue when training accuracy is nearly perfect.</u>

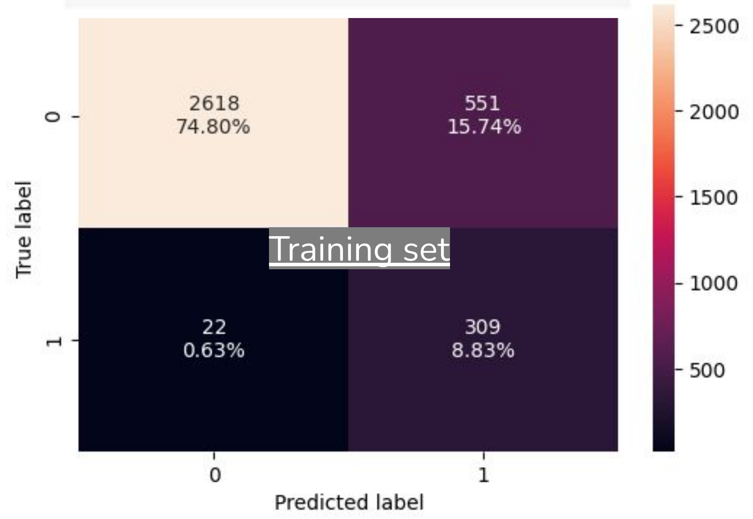
<u>Training set</u>

Metric	Value
Accuracy	98.9%
Recall	91.2%
Precision	97.4%
F1-Score	94.2%

Model Building - Model performance (Post Pruned)

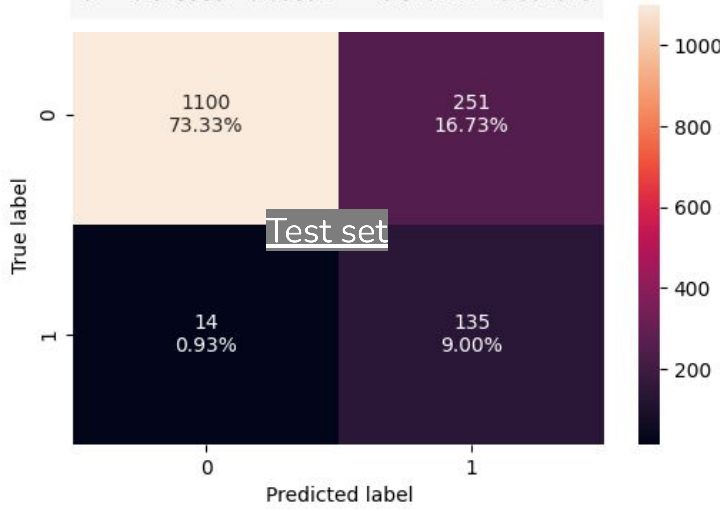
Post Pruned tree

	Accuracy	Recall	Precision	F1
0	0.836286	0.933535	0.359302	0.518892



Training set

	Accuracy	Recall	Precision	F1
0	0.823333	0.90604	0.349741	0.504673



Test set

Model Building - Model performance (Post Pruned)

1. **Accuracy (82.3% on Test Set)**:
   The model correctly predicts the class (loan acceptance or not) 82.3% of the time on the test set. This indicates a decrease in overall accuracy compared to the pre-pruned model.
2. **Recall (90.6% on Test Set)**:
   Out of all actual loan acceptances, the model successfully identifies 90.6%. This shows that the model performs well in detecting positive cases (loan acceptances), though slightly less than the pre-pruned model.
3. **Precision (34.97% on Test Set)**:
   Out of all predicted loan acceptances, only 34.97% are correct. This is a significant drop compared to the pre-pruned model, indicating a higher false-positive rate.
4. **F1-Score (50.47% on Test Set)**:
   The harmonic mean of Precision and Recall is 50.47%. This is lower than the pre-pruned model, showing a trade-off in balancing false positives and false negatives.

Comparison with Training Set: The training set accuracy is slightly higher at 83.6%, but Precision (35.93%) and F1-score (51.89%) remain close to the test set values, suggesting a better generalization of the model.

Recall on the training set (93.35%) is slightly better than the test set (90.6%). Recall remains high but at the cost of lower precision, potentially increasing marketing expenses due to false positives.

Trade-off from Post-Pruning: By pruning the tree, the model became simpler and less likely to overfit. However, this came at the cost of lower Precision and F1-score, which are critical in imbalanced datasets where false positives or false negatives can be problematic.

Model Generalization: The similarity in metrics between training and test sets indicates good generalization but reduced predictive power for identifying true loan acceptances accurately.

Test set

Metric	Value
Accuracy	82.3%
Recall	90.6%
Precision	34.9%
F1-Score	50.4%

Training set

Metric	Value
Accuracy	83.6%
Recall	93.3%
Precision	35.9%
F1-Score	51.8%

Model Performance Summary

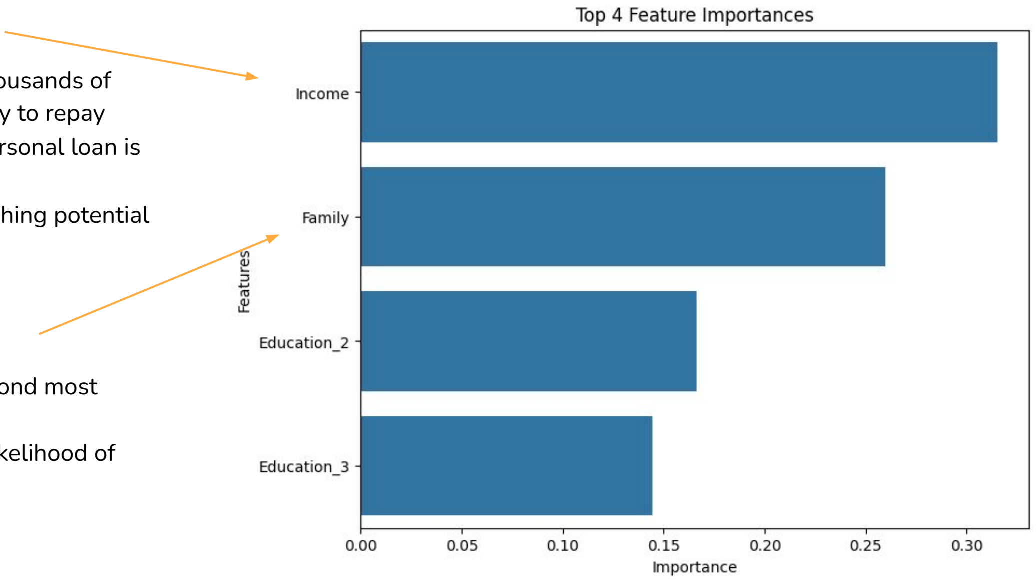
The 4 most important features

1. Income

- This feature represents the customer's annual income (in thousands of dollars). Higher income likely increases the customer's ability to repay loans, making it a crucial factor in determining whether a personal loan is accepted.
- With the highest importance, this feature is key in distinguishing potential borrowers with a stable financial capacity.

2. Family

- The feature importance chart shows that "Family" is the second most influential variable in predicting loan acceptance.
- This suggests that family size correlates strongly with the likelihood of accepting a loan.



Model Performance Summary

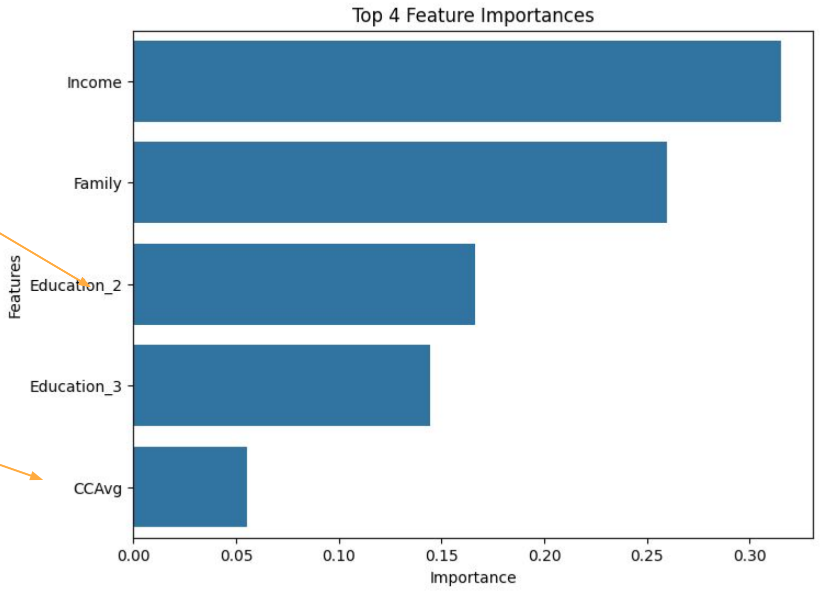
The 4 most important features

3. Education (Graduate and Advanced/Professional)

- **Explanation:** These features reflect the customer's education level. Higher education levels often correlate with higher income and financial stability, influencing loan acceptance.
- **Impact:** Education acts as a proxy for financial literacy and earning potential, which is why it's a critical determinant in the model.

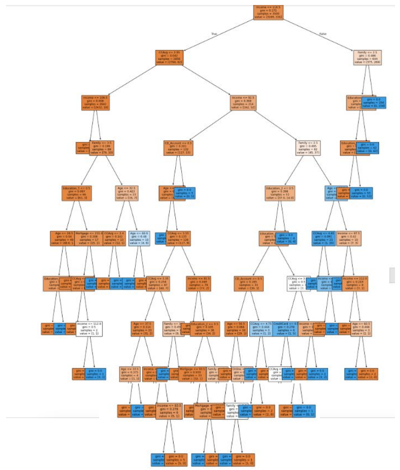
4. CCAvg (Credit Card Average Spending)

- **Explanation:** This feature represents the customer's average monthly spending on credit cards (in thousands of dollars). It indicates their spending behavior and financial habits.
- **Impact:** A moderate importance level suggests that spending patterns help assess creditworthiness and loan needs.

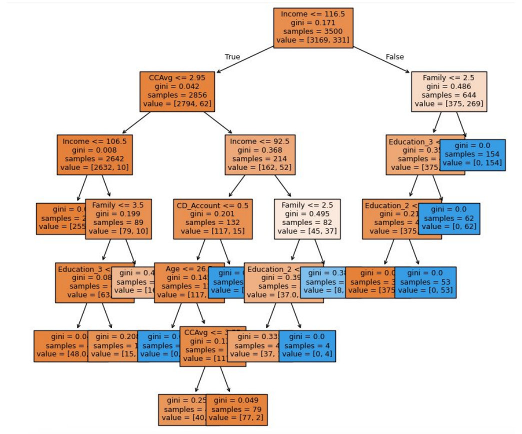


Model Performance Summary (The trees)

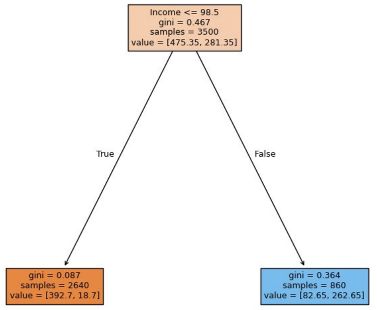
Default tree



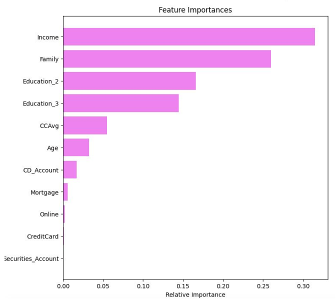
Pre Pruned tree

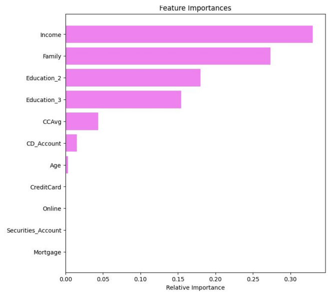


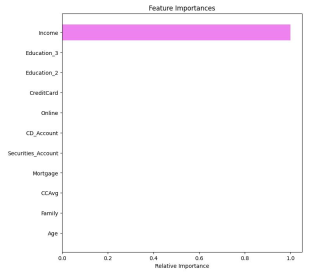
Post Pruned tree



Most important features for each tree, see previous slide for top 4.







Model Performance Improvement and Pruning Techniques

Default Decision Tree:

- The default decision tree model seems to overfit the training data, achieving perfect metrics (Accuracy, Recall, Precision, F1-score all = 1.0).
- The model is not overfitting (which is a common issue when training accuracy is perfect) since it is doing well on the testset as well, although slightly dropping. But to ensure its robustness further evaluation and other models needs to be considered and tested

Pre-Pruned Decision Tree:

- The pre-pruned tree achieves a near-perfect balance between complexity and performance.
- **Improvements over Default Tree:**
    - It avoids overfitting, as seen in the consistency between training and test set performance (Accuracy = 98.4% for both).
    - Recall drops slightly compared to the default tree (86.57% vs. 91.27%), but this is acceptable since the model is less likely to overfit.
    - The pre-pruned tree captures patterns in the data more effectively without sacrificing test performance, making it a well-generalized model.

Post-Pruned Decision Tree:

- Post-pruning significantly simplifies the model, as evident from lower metrics.
- **Impact on Performance:**
    - Recall remains relatively high (90.6% on test data), ensuring positive cases are still captured.
    - Precision and F1-score drop drastically (Precision = 34.97%, F1-score = 50.47%), indicating that the model struggles with false positives and overall balance.
- While post-pruning minimizes complexity, it comes at the cost of reduced predictive performance, particularly Precision, making it less suitable for this problem.

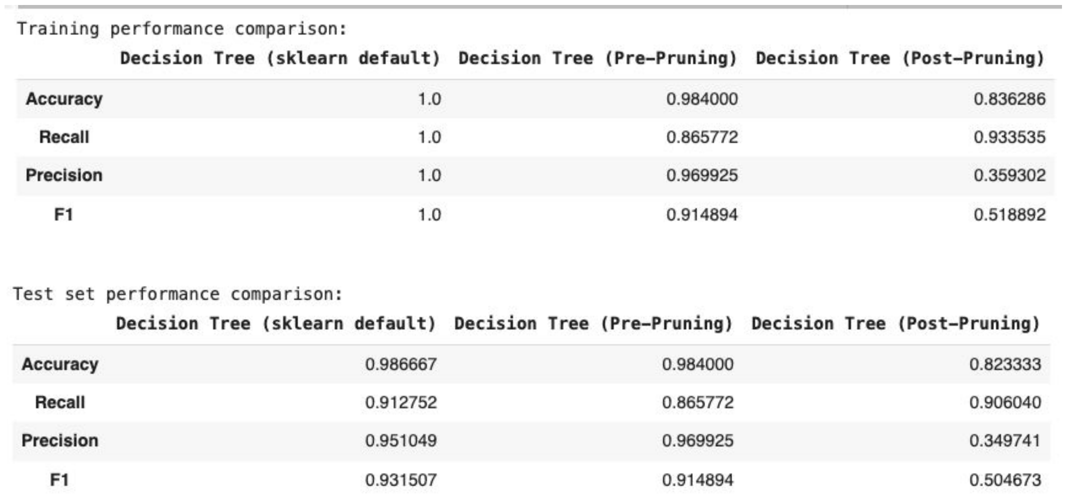
Model Performance Summary

Model Performance Comparison and Final Model Selection

The **Pre-Pruned Decision Tree** emerges as the best model:

- It strikes a balance between performance and simplicity.
- High Accuracy (98.4%), Precision (96.99%), and F1-score (91.49%) on the test set indicate reliable and generalizable predictions.
- Recall (86.57%) remains robust, ensuring minimal false negatives.
- The model avoids overfitting while retaining strong predictive power, making it ideal for deployment.

The **Post-Pruned Decision Tree**, while simpler, sacrifices significant predictive performance, particularly in Precision and F1-score, making it unsuitable for this use case.



Training performance comparison:

	Decision Tree (sklearn default)	Decision Tree (Pre-Pruning)	Decision Tree (Post-Pruning)
Accuracy	1.0	0.984000	0.836286
Recall	1.0	0.865772	0.933535
Precision	1.0	0.969925	0.359302
F1	1.0	0.914894	0.518892

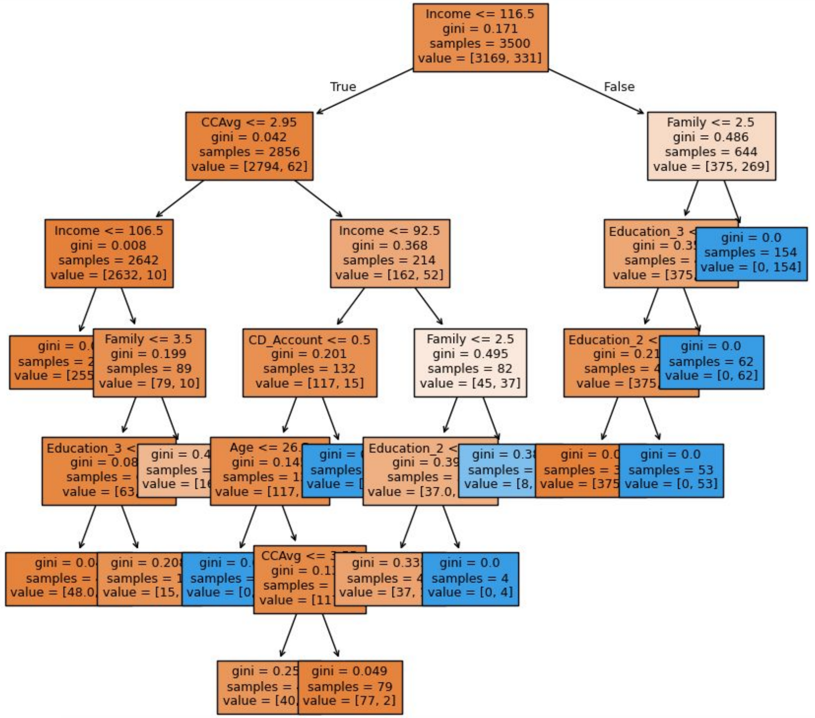
Test set performance comparison:

	Decision Tree (sklearn default)	Decision Tree (Pre-Pruning)	Decision Tree (Post-Pruning)
Accuracy	0.986667	0.984000	0.823333
Recall	0.912752	0.865772	0.906040
Precision	0.951049	0.969925	0.349741
F1	0.931507	0.914894	0.504673

Decision Rules (Pre Pruned tree)

The pre-pruned tree decision rules highlight key splits based on the following features:

1. **Income:**
   - The most critical factor for predicting loan acceptance. For example, lower-income customers (<=116.50) are typically predicted as non-loan acceptors unless other conditions (like high education or smaller family size) indicate otherwise.
2. **Family Size:**
   - Customers with smaller family sizes (≤2.5) and higher education levels (Education_2 or Education_3) are more likely to accept loans.
3. **Credit Card Spending (CCAvg):**
   - Higher average monthly spending (>2.95) is associated with higher loan acceptance, especially for higher-income customers.
4. **Education Levels:**
   - Customers with advanced education (Education_2 or Education_3) are more likely to accept loans, even in scenarios of moderate income. Further more, customers with high income, small family size, advanced education, and moderate to high credit card spending are the most likely to accept loans.

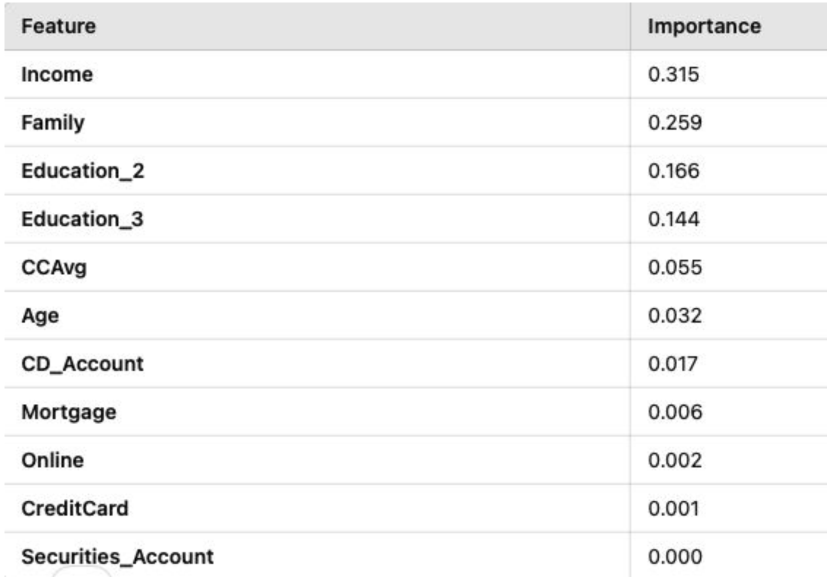


Business context:

- Target high-income individuals with tailored campaigns.
- Use education and family size as secondary indicators to refine marketing efforts.
- Analyze credit card spending patterns to better segment customers.

Model Performance - Key observations

- Income and Family size are the most significant features, together accounting for more than 50% of the tree's splits.
- Education (both graduate and advanced levels) contributes significantly, reflecting its influence on loan acceptance behavior.
- CCAvg, though less critical, plays an auxiliary role in further refining splits, particularly for higher-income groups.
- Securities Account and CreditCard have negligible importance, confirming they contribute minimally to predicting loan acceptance.



Feature	Importance
Income	0.315
Family	0.259
Education_2	0.166
Education_3	0.144
CCAvg	0.055
Age	0.032
CD_Account	0.017
Mortgage	0.006
Online	0.002
CreditCard	0.001
Securities_Account	0.000

Model Performance - Final Conclusions

Pre-Pruned Tree: The Best Overall Model:

- The Pre-Pruned Tree provides a strong balance between predictive performance and interpretability.
- It achieves high accuracy (98.4%), recall (86.5%), and precision (96.9%) on the test set, ensuring the model performs well in identifying both loan acceptors (positive class) and non-acceptors (negative class).
- The relatively moderate depth and size of the tree make it easy to interpret while retaining a strong focus on key business metrics.
- By maintaining high recall, the model ensures that most potential loan acceptors are identified, minimizing missed opportunities.
- Its precision means the model avoids making excessive false-positive predictions, ensuring marketing resources are not wasted.

Recommendations Based on Results:

- Choose the Pre-Pruned Tree for Deployment. It provides the best trade-off between performance and complexity, ensuring robust identification of loan acceptors without sacrificing interpretability.
- Target high-income customers with small families and advanced education levels as high-priority segments for loan offers.

Summary:

The Pre-Pruned Tree is the recommended model for deployment, as it achieves high predictive performance while remaining interpretable and aligned with business objectives. However, the Post-Pruned Tree offers a simplified alternative if interpretability or computational constraints are paramount, though at the cost of reduced effectiveness.

APPENDIX

Data Background and Contents

Dataset Overview:

- Source: Fictional banking dataset created for educational purposes.
- Purpose: To analyze customer behavior and predict personal loan acceptance based on demographic, financial, and banking-related features.
- Size: 5,000 rows and 11 features.

Key Variables:

- **Target Variable**: Personal_Loan – Indicates whether a customer accepted the personal loan offer.
- **Predictor Variables**: Features like Age, Income, Family, CCAvg, Education, and more, which are used to predict the target.

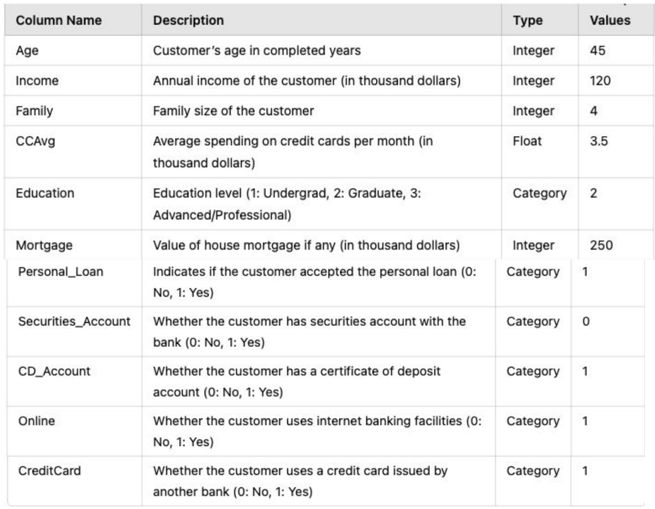
Relevance:

- This dataset is representative of real-world customer data used in banking and finance to develop predictive models for targeted marketing and customer segmentation.
- This dataset aligns with the business objective by providing insights into customer segmentation and predicting loan acceptance likelihood based on financial, demographic, and behavioral factors.

Data Preparation:

- Handled outliers in Mortgage and CCAvg variables.
- Categorical features (Education, Securities_Account, etc.) were converted into numerical representations for modeling.
- Dataset split into 70% training data and 30% test data for model evaluation.
- Handling outliers in variables like Mortgage and CCAvg was crucial to ensure the model's predictions were not skewed by extreme or erroneous values, thereby improving the model's robustness and reliability in real-world scenarios.

Dataset:



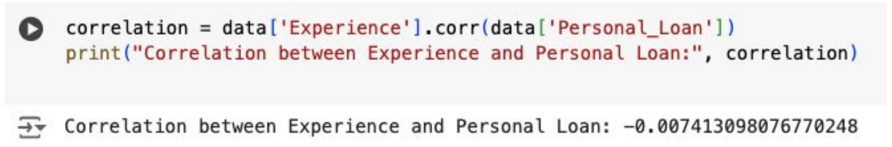
Column Name	Description	Type	Values
Age	Customer's age in completed years	Integer	45
Income	Annual income of the customer (in thousand dollars)	Integer	120
Family	Family size of the customer	Integer	4
CCAvg	Average spending on credit cards per month (in thousand dollars)	Float	3.5
Education	Education level (1: Undergrad, 2: Graduate, 3: Advanced/Professional)	Category	2
Mortgage	Value of house mortgage if any (in thousand dollars)	Integer	250
Personal_Loan	Indicates if the customer accepted the personal loan (0: No, 1: Yes)	Category	1
Securities_Account	Whether the customer has securities account with the bank (0: No, 1: Yes)	Category	0
CD_Account	Whether the customer has a certificate of deposit account (0: No, 1: Yes)	Category	1
Online	Whether the customer uses internet banking facilities (0: No, 1: Yes)	Category	1
CreditCard	Whether the customer uses a credit card issued by another bank (0: No, 1: Yes)	Category	1

#Data Preprocessing - Missing value treatment (Did not use)

Chosen approach - Dropping the negative values

The 52 rows with negative values in the 'Experience' column represent only 1.04% of the dataset, making them a small proportion. These values are likely erroneous and contribute little meaningful information. Additionally, the correlation between 'Experience' and 'Personal Loan' is extremely low (-0.0074), indicating that 'Experience' has no significant relationship with the target variable. To maintain data integrity and avoid introducing noise, the choice was to drop these rows.

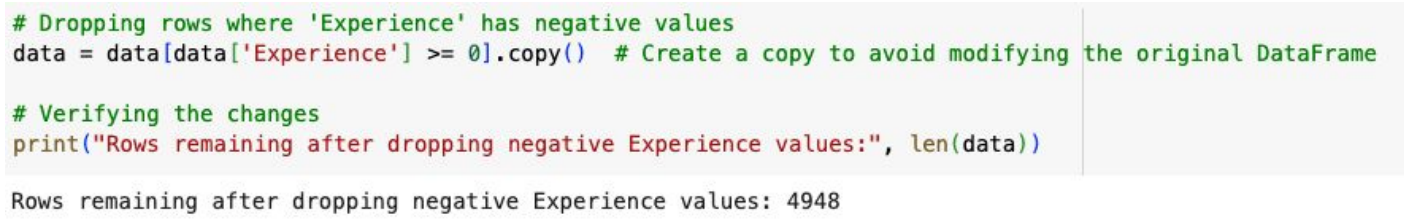
Low correlation



```
correlation = data['Experience'].corr(data['Personal_Loan'])
print("Correlation between Experience and Personal Loan:", correlation)
```

```
Correlation between Experience and Personal Loan: -0.007413098076770248
```

Dropping the rows



```
# Dropping rows where 'Experience' has negative values
data = data[data['Experience'] >= 0].copy()  # Create a copy to avoid modifying the original DataFrame

# Verifying the changes
print("Rows remaining after dropping negative Experience values:", len(data))
```

```
Rows remaining after dropping negative Experience values: 4948
```

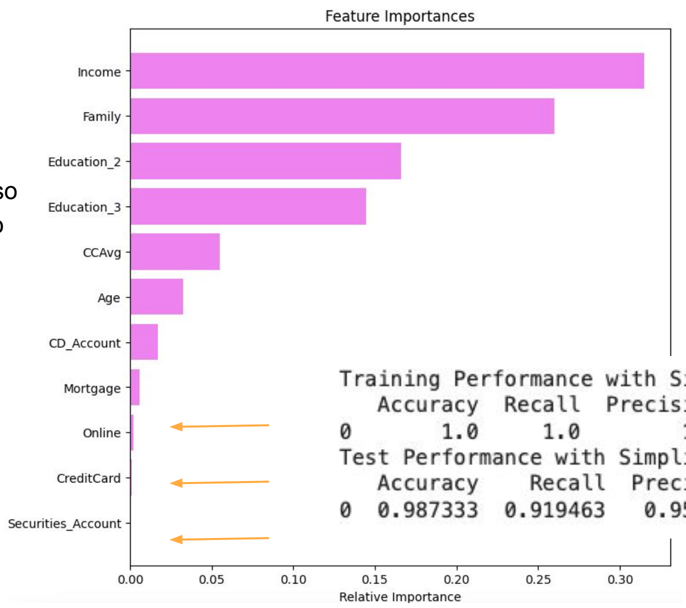
#Simplified the tree, did not use!

Simplifying the model

The model already performs well, but when simplifying the model by dropping features with very low or zero importance can further improve interpretability and also reduce computational complexity, especially if they do not contribute meaningfully to the predictions.

Why Drop These Features?

1. **Securities_Account**:
   - Feature importance is 0, indicating it has no contribution to the splits or predictions in the decision tree.
   - It adds no value and can safely be dropped.
2. **CreditCard**:
   - Feature importance is very close to zero, suggesting its contribution to the model is negligible.
   - Dropping it simplifies the model without significant impact on performance.
3. **Online:**
   - Feature importance is very close to zero, suggesting its contribution to the model is negligible.
   - Dropping it simplifies the model without significant impact on performance.



```
Training Performance with Simplified Features:
   Accuracy  Recall  Precision   F1
0       1.0     1.0        1.0  1.0
Test Performance with Simplified Features:
   Accuracy    Recall  Precision        F1
0  0.987333  0.919463   0.951389  0.935154
```

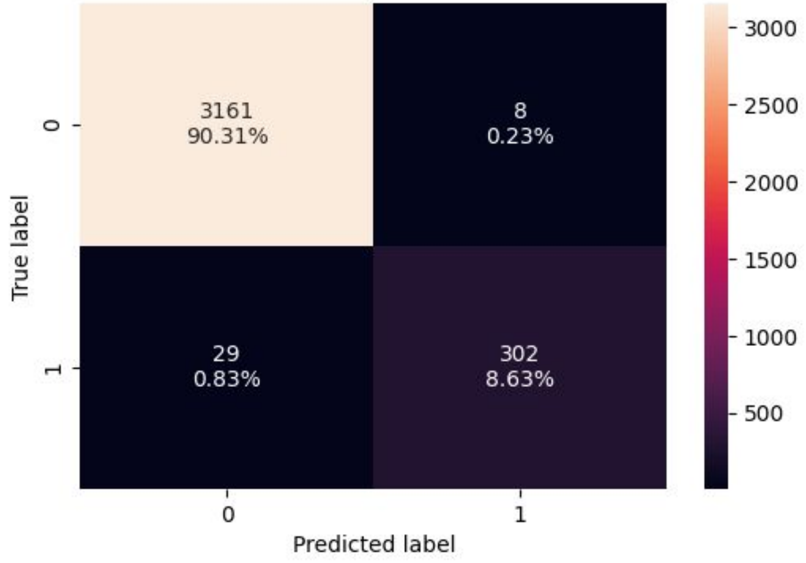
Simplified model:
The simplified model maintains excellent performance metrics, with no significant drop in accuracy, recall, precision, or F1 score, indicating that the removed features had negligible impact on the model's predictive ability.

#Simplified the tree, did not use!

Checking performance on simplified model

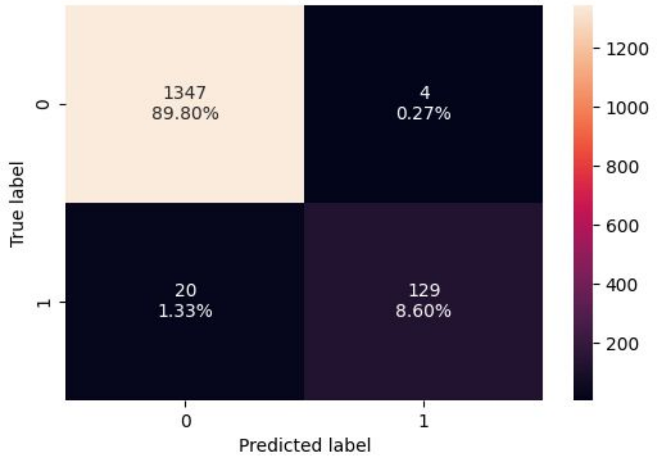
Training set (simplified)

	Accuracy	Recall	Precision	F1
0	0.989429	0.912387	0.974194	0.942278



Test set (simplified)

	Accuracy	Recall	Precision	F1
0	0.984	0.865772	0.969925	0.914894

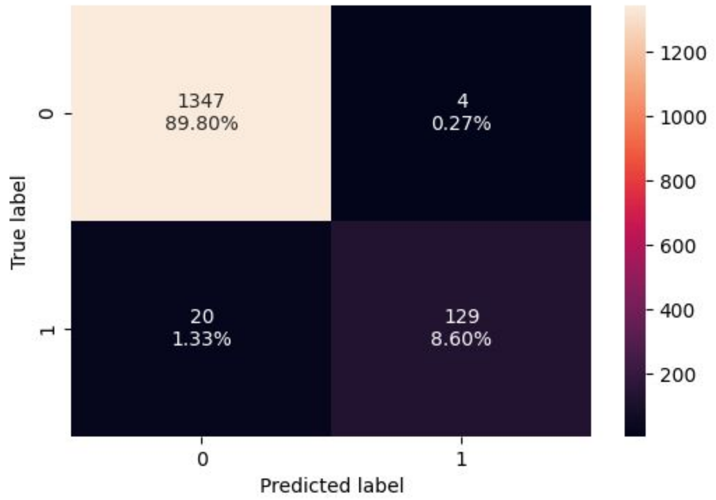


#Simplified the tree, did not use!

Pre Pruned Decision Tree

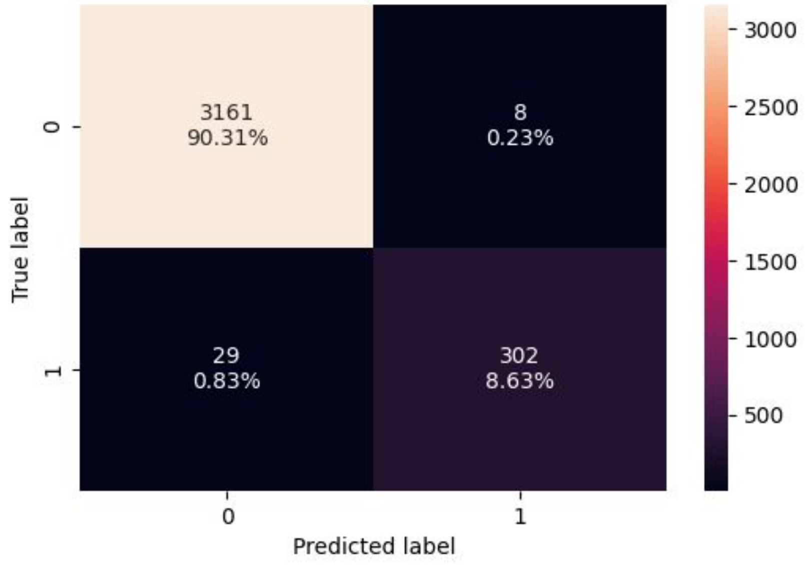
Testing set (simplified and pre pruned)

	Accuracy	Recall	Precision	F1
0	0.984	0.865772	0.969925	0.914894



Training set (simplified and pre pruned)

	Accuracy	Recall	Precision	F1
0	0.989429	0.912387	0.974194	0.942278



#Simplified the tree, did not use!

Decision Tree (Pre Pruned)

```
] # Drop low-importance features
  X_train_simplified = X_train.drop(["Securities_Account", "CreditCard", "Online", "Mortgage", "Age"], axis=1)
  X_test_simplified = X_test.drop(["Securities_Account", "CreditCard", "Online", "Mortgage", "Age"], axis=1)

  # Check new feature set
  print("Updated Features:", X_train_simplified.columns)
```

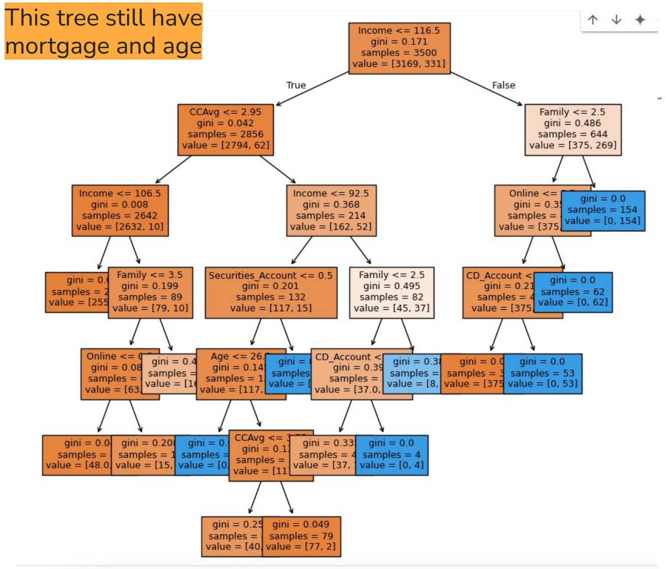
```
Updated Features: Index(['Income', 'Family', 'CCAvg', 'CD_Accou
       'Education_3'],
      dtype='object')
```

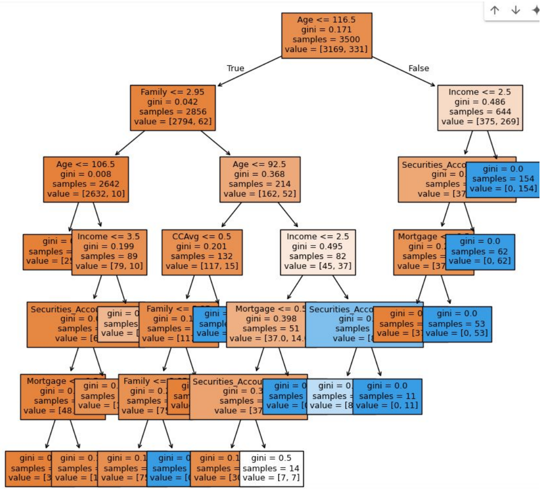
Removing more features; age and mortgage

	Imp
Income	0.330122
Family	0.273640
Education_2	0.180204
Education_3	0.154207
CCAvg	0.043562
CD_Account	0.015101
Age	0.003164
Mortgage	0.000000

The pre pruned tree show importance of mortgage and age were very low, indicating low impact on the model. Removing them did not change performance

This tree still have mortgage and age





This tree mortgage and age are dropped

Great Learning

Happy Learning !