Written By: Isis Vasquez

4/3/2025

Imports

```
# Basic libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# AutoViz
!pip install autoviz
from autoviz.AutoViz_Class import AutoViz_Class

# PyCaret
!pip install pycaret
from pycaret.classification import *

# Warnings
import warnings
warnings.filterwarnings('ignore')
```

Show hidden output

Input Data

```
try:
  df = pd.read_csv('https://raw.githubusercontent.com/Ivasquez2003/Data-Mining-Project/re
  display(df.head())
except FileNotFoundError:
  print("Error: 'test_Y3wMUE5_7gLdaTN.csv' not found. Please upload the file or provide t
except pd.errors.EmptyDataError:
  print("Error: 'test_Y3wMUE5_7gLdaTN.csv' is empty.")
except pd.errors.ParserError:
  print("Error: Could not parse 'test_Y3wMUE5_7gLdaTN.csv'. Please check the file format.
except Exception as e:
  print(f"An unexpected error occurred: {e}")
```

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome
0	LP001002	Male	No	0	Graduate	No	5849

0	LP001002	Male	No	0	Graduate	No	5849
1	LP001003	Male	Yes	1	Graduate	No	4583
2	LP001005	Male	Yes	0	Graduate	Yes	3000
3	LP001006	Male	Yes	0	Not Graduate	No	2583
4	LP001008	Male	No	0	Graduate	No	6000

Exploring Data

```
# Statistical Summary
df.describe()
```

	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_Histo
count	614.000000	614.000000	592.000000	600.00000	564.0000
mean	5403.459283	1621.245798	146.412162	342.00000	0.8421
std	6109.041673	2926.248369	85.587325	65.12041	0.3648
min	150.000000	0.000000	9.000000	12.00000	0.0000
25%	2877.500000	0.000000	100.000000	360.00000	1.0000
50%	3812.500000	1188.500000	128.000000	360.00000	1.0000
75%	5795.000000	2297.250000	168.000000	360.00000	1.0000
max	81000.000000	41667.000000	700.000000	480.00000	1.0000

```
# Summary Statistics
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 614 entries, 0 to 613
Data columns (total 13 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Loan_ID            614 non-null    object
 1   Gender             601 non-null    object
 2   Married            611 non-null    object
 3   Dependents         599 non-null    object
 4   Education          614 non-null    object
 5   Self_Employed      582 non-null    object
```

```
 6   ApplicantIncome    614 non-null    int64
 7   CoapplicantIncome  614 non-null    float64
 8   LoanAmount         592 non-null    float64
 9   Loan_Amount_Term   600 non-null    float64
 10  Credit_History     564 non-null    float64
 11  Property_Area      614 non-null    object
 12  Loan_Status        614 non-null    object
dtypes: float64(4), int64(1), object(8)
memory usage: 62.5+ KB
```

```
# Analysis of Nulls
df.isnull().sum()
```

	0
Loan_ID	0
Gender	13
Married	3
Dependents	15
Education	0
Self_Employed	32
ApplicantIncome	0
CoapplicantIncome	0
LoanAmount	22
Loan_Amount_Term	14
Credit_History	50
Property_Area	0
Loan_Status	0

dtype: int64

```
# AutoViz visualization
AV = AutoViz_Class()
auto_viz = AV.AutoViz('https://raw.githubusercontent.com/Ivasquez2003/Data-Mining-Project
```

```
Shape of your Data Set loaded: (614, 13)
#######################################################################################
####################### C L A S S I F Y I N G   V A R I A B L E S  ####################
#######################################################################################
Classifying variables in data set...
    Number of Numeric Columns =  3
    Number of Integer-Categorical Columns =  1
```

```
    Number of String-Categorical Columns =  2
    Number of Factor-Categorical Columns =  0
    Number of String-Boolean Columns =  5
    Number of Numeric-Boolean Columns =  1
    Number of Discrete String Columns =  0
    Number of NLP String Columns =  0
    Number of Date Time Columns =  0
    Number of ID Columns =  1
    Number of Columns to Delete =  0
    13 Predictors classified...
        1 variable(s) removed since they were ID or low-information variables
        List of variables removed: ['Loan_ID']
To fix these data quality issues in the dataset, import FixDQ from autoviz...
    All variables classified into correct types.
```

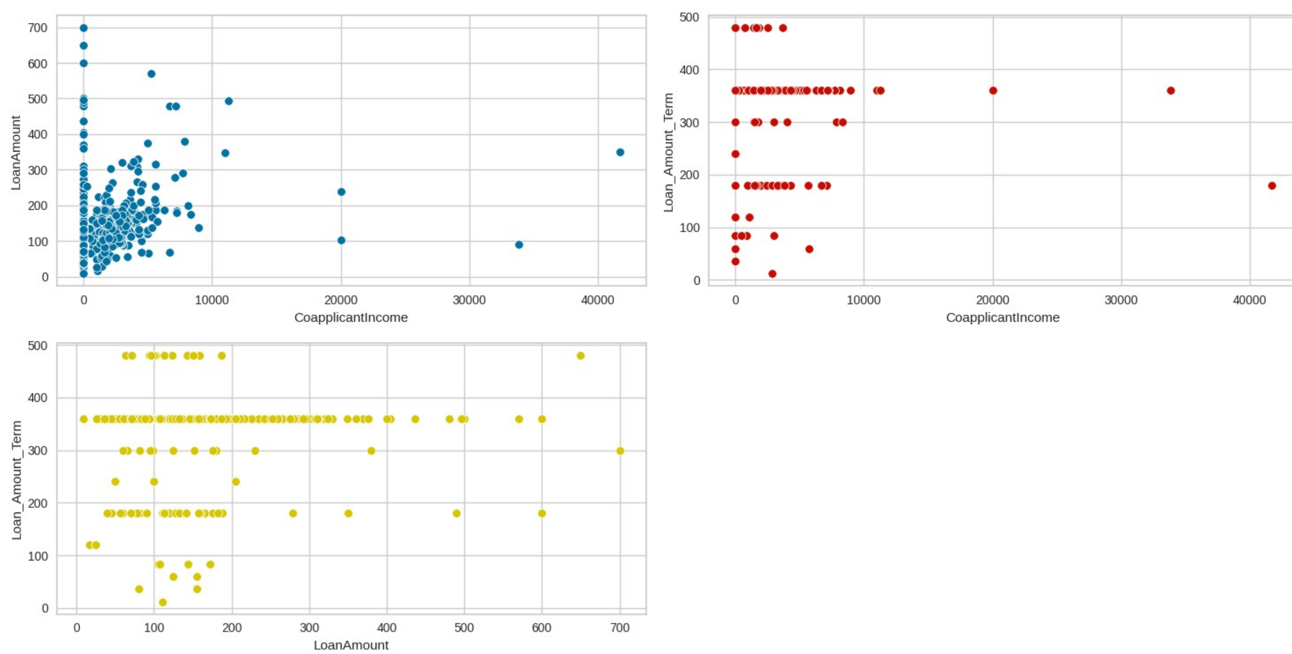
	Data Type	Missing Values%	Unique Values%	Minimum Value	Maximum Value	DQ Issue
Loan_ID	object	0.000000	100			Possible ID column: drop before modeling step.
Gender	object	2.117264	0			13 missing values. Impute them with mean, median, mode, or a constant value such as 123., Mixed dtypes: has 2 different data types: object, float,
Married	object	0.488599	0			3 missing values. Impute them with mean, median, mode, or a constant value such as 123., Mixed dtypes: has 2 different data types: object, float,
Dependents	object	2.442997	0			15 missing values. Impute them with mean, median, mode, or a constant value such as 123., Mixed dtypes: has 2 different data types: object, float,
Education	object	0.000000	0			No issue

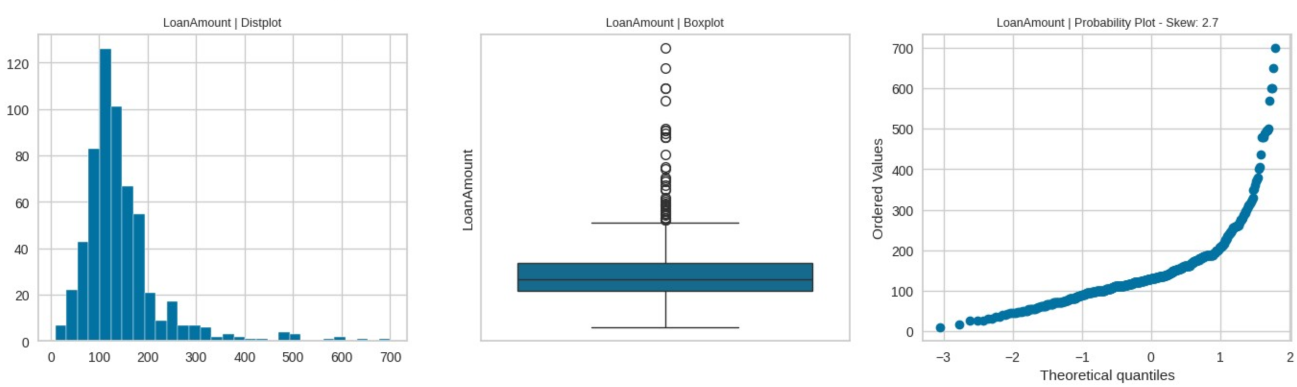
	object	0.000000	0			No issue
Self_Employed	object	5.211726	0			32 missing values. Impute them with mean, median, mode, or a constant value such as 123., Mixed dtypes: has 2 different data types: object, float,
ApplicantIncome	int64	0.000000	82	150.000000	81000.000000	Column has 50 outliers greater than upper bound (10171.25) or lower than lower bound(-1498.75). Cap them or remove them.
CoapplicantIncome	float64	0.000000	NA	0.000000	41667.000000	Column has 18 outliers greater than upper bound (5743.12) or lower than lower bound(-3445.88). Cap them or remove them.
LoanAmount	float64	3.583062	NA	9.000000	700.000000	22 missing values. Impute them with mean, median, mode, or a constant value such as 123., Column has 39 outliers greater than upper bound (270.00) or lower than lower bound(-2.00). Cap them or remove them.
Loan_Amount_Term	float64	2.280130	NA	12.000000	480.000000	14 missing values. Impute them with mean, median, mode, or a constant value such as 123., Column has 88 outliers greater

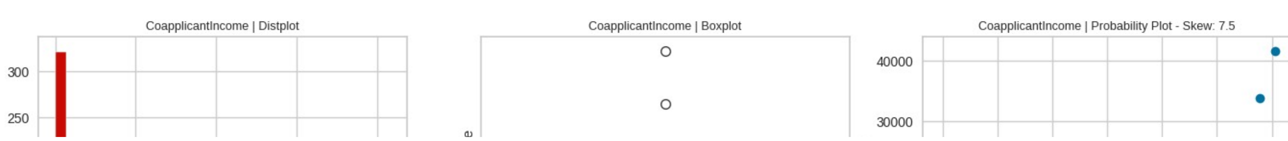
				than upper bound (360.00) or lower than lower bound(360.00). Cap them or remove them.
Credit_History	float64	8.143322	0	50 missing values. Impute them with mean, median, mode, or a constant value such as 123.
Property_Area	object	0.000000	0	No issue
Loan_Status	object	0.000000	0	No issue

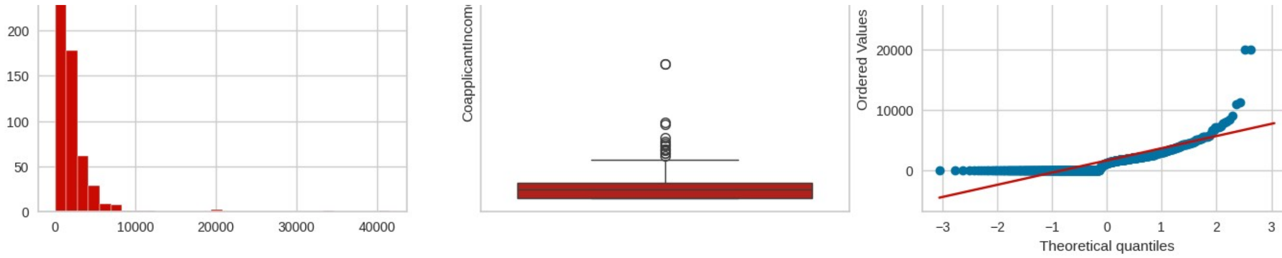
```
Number of All Scatter Plots = 6
```

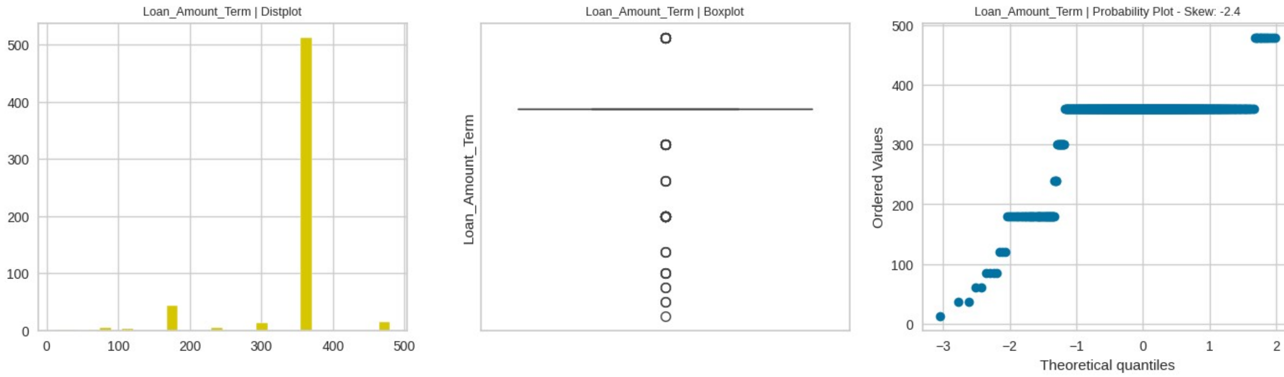
Pair-wise Scatter Plot of all Continuous Variables



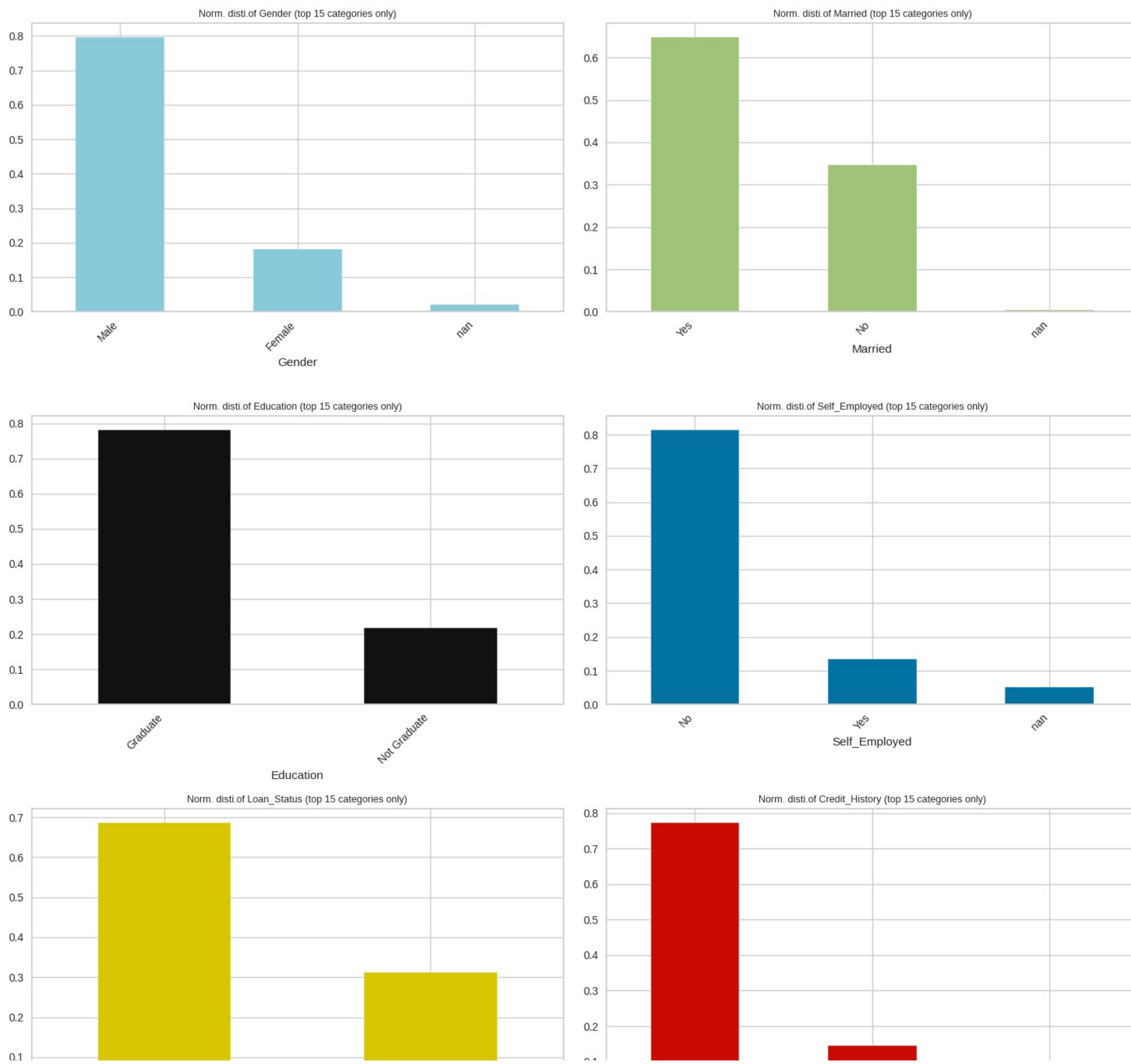


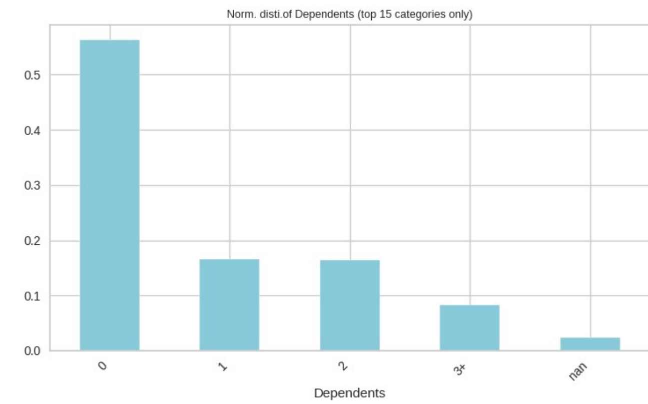


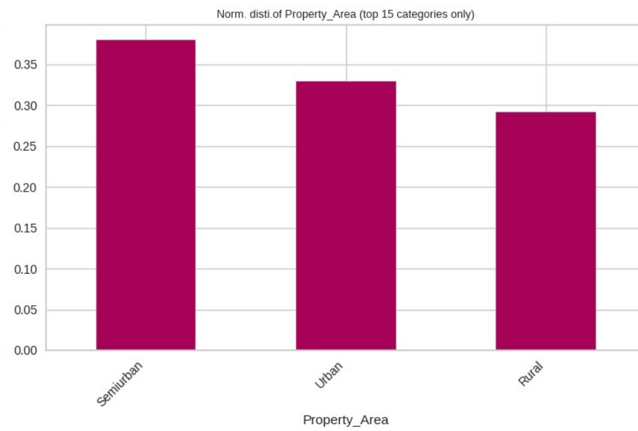


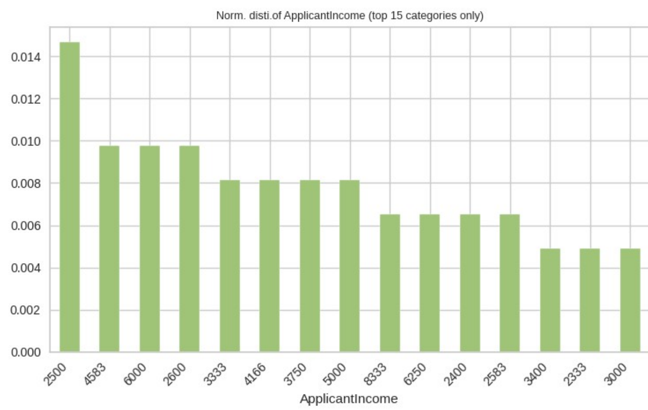


Histograms and Normalized distributions of all variables

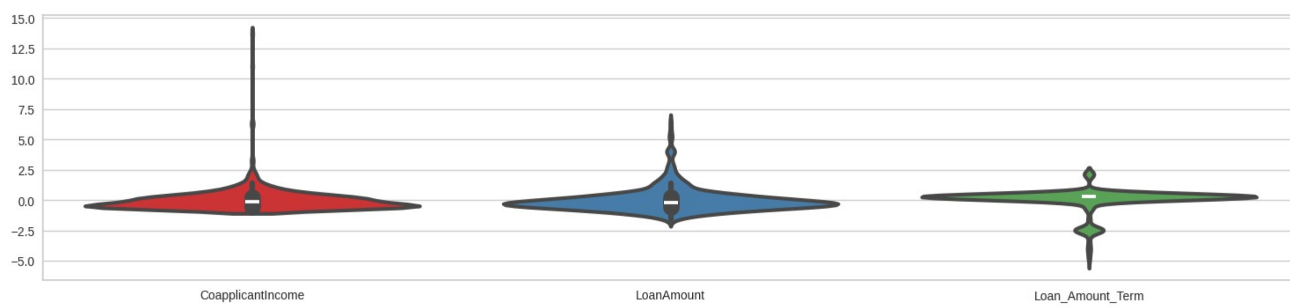






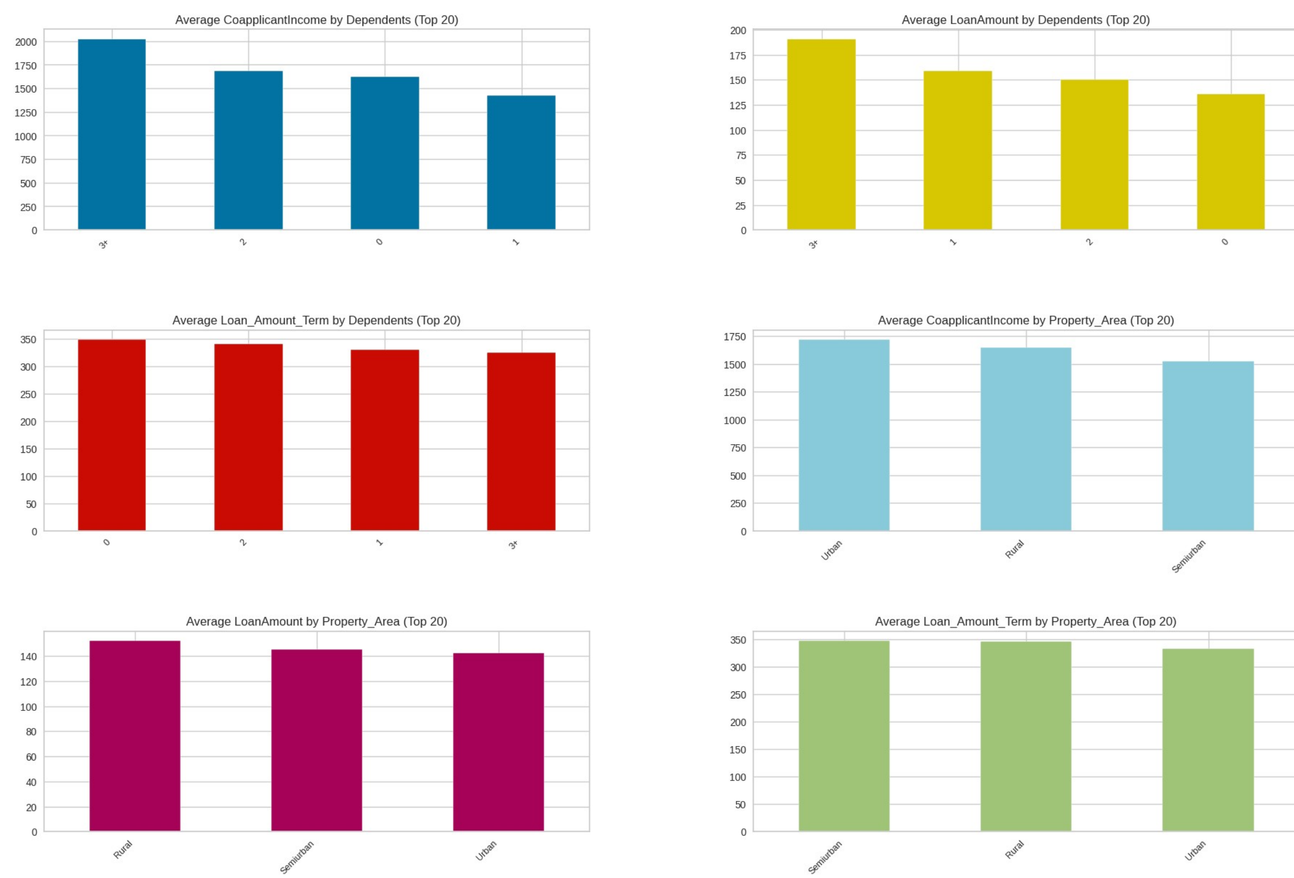


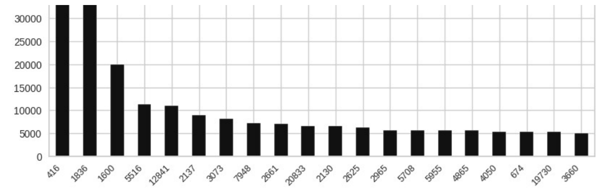
Violin Plot of all Continuous Variables

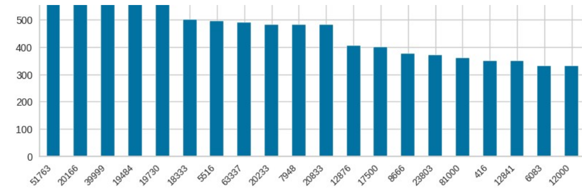


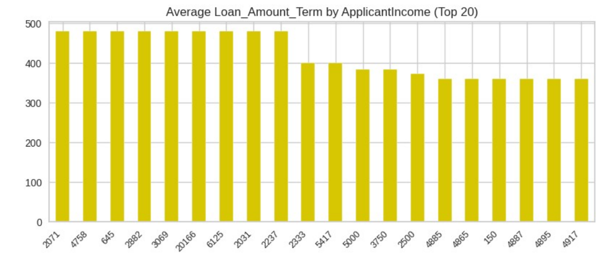
Heatmap of all Numeric Variables including target:

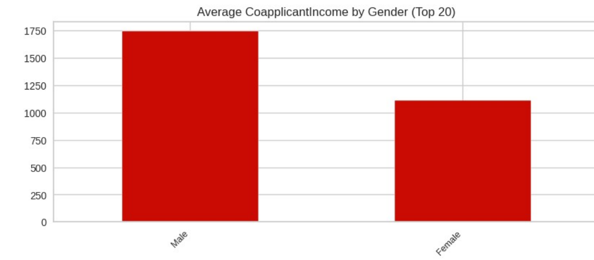
Bar plots for each Continuous by each Categorical variable

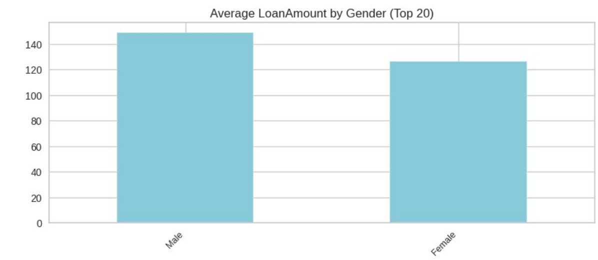


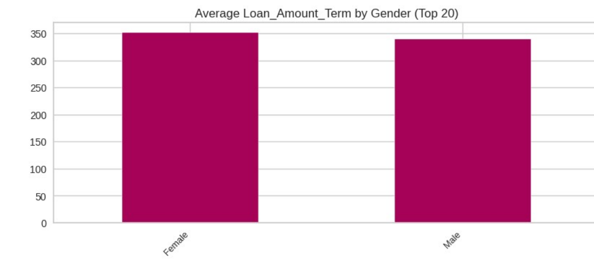

416
1836
1600
5516
12841
2137
3073
7948
2661
20833
2130
2625
2965
5708
5955
4865
4050
674
19730
3660

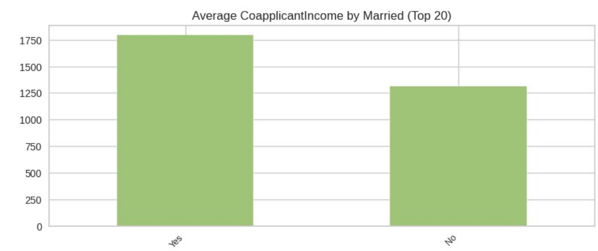

51763
20166
39999
19484
19730
18333
5516
63337
20233
7948
20833
12876
17500
8666
23803
81000
416
12841
6083
12000

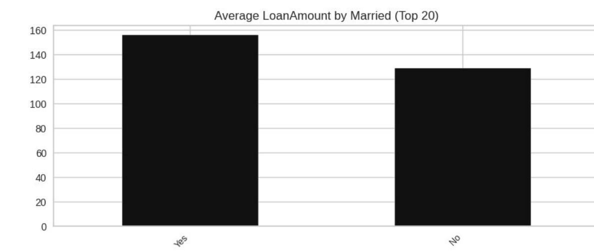

Average Loan_Amount_Term by ApplicantIncome (Top 20)
2071
4758
645
2882
3069
20166
6125
2031
2237
2333
5417
5000
3750
2500
4885
4865
150
4887
4895
4917

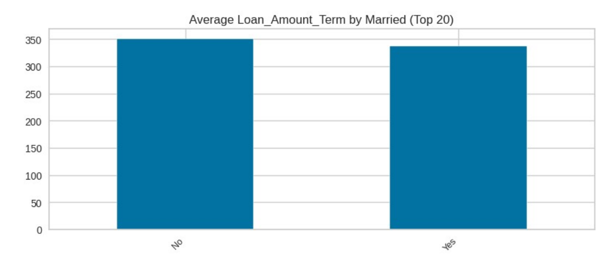

Average CoapplicantIncome by Gender (Top 20)
Male
Female

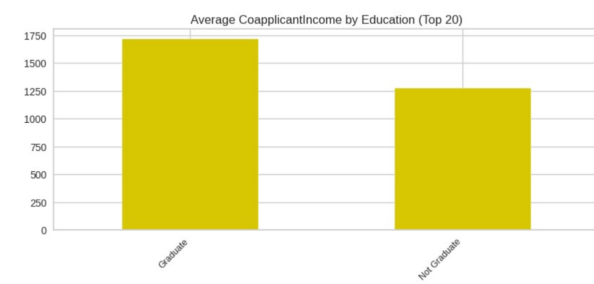

Average LoanAmount by Gender (Top 20)
Male
Female

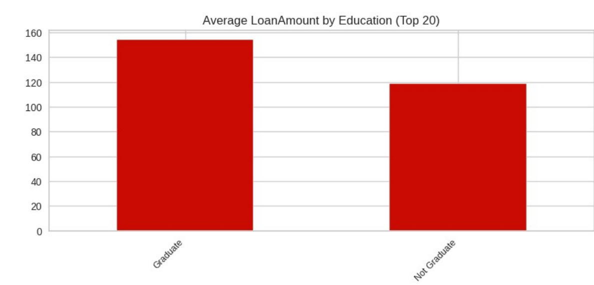

Average Loan_Amount_Term by Gender (Top 20)
Female
Male

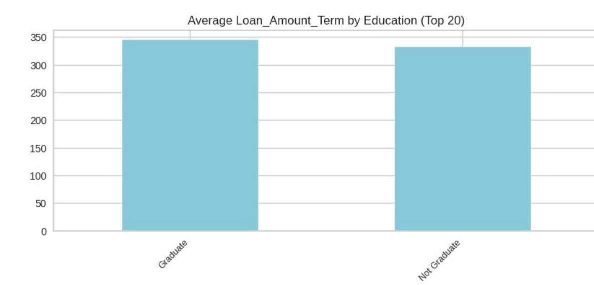

Average CoapplicantIncome by Married (Top 20)
Yes
No

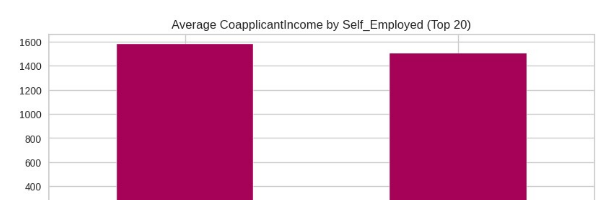

Average LoanAmount by Married (Top 20)
Yes
No

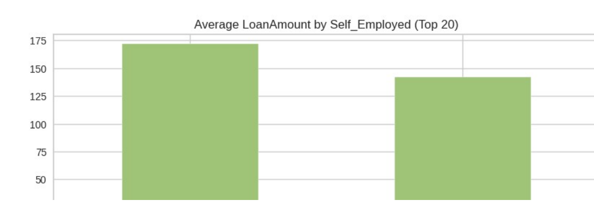

Average Loan_Amount_Term by Married (Top 20)
No
Yes


Average CoapplicantIncome by Education (Top 20)
Graduate
Not Graduate

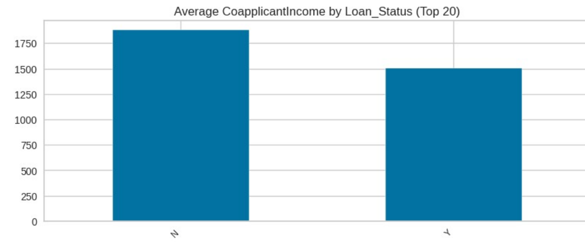

Average LoanAmount by Education (Top 20)
Graduate
Not Graduate

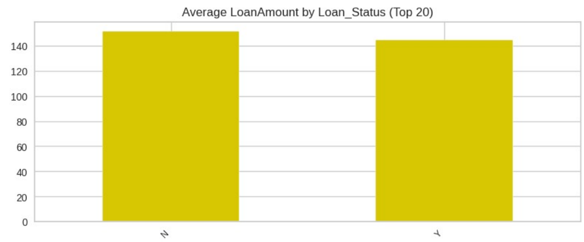

Average Loan_Amount_Term by Education (Top 20)
Graduate
Not Graduate

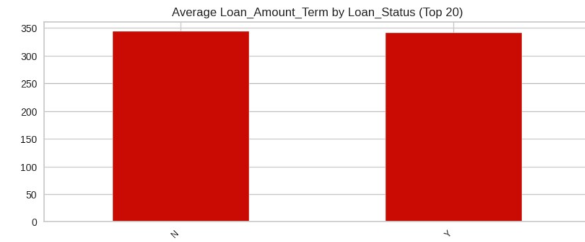

Average CoapplicantIncome by Self_Employed (Top 20)


Average LoanAmount by Self_Employed (Top 20)










```
All Plots done
Time to run AutoViz = 19 seconds

 ###################### AUTO VISUALIZATION Completed ########################
```

Preparing Data

```
# Drop columns with many missing values or irrelevant
df_clean = df.drop(columns=['Loan_ID'])

# Fill missing values
df_clean['Gender'] = df_clean['Gender'].fillna(df_clean['Gender'].mode()[0])
df_clean['Married'] = df_clean['Married'].fillna(df_clean['Married'].mode()[0])
df_clean['Dependents'] = df_clean['Dependents'].fillna(df_clean['Dependents'].mode()[0])
df_clean['Self_Employed'] = df_clean['Self_Employed'].fillna(df_clean['Self_Employed'].mo
df_clean['LoanAmount'] = df_clean['LoanAmount'].fillna(df_clean['LoanAmount'].median())
df_clean['Loan_Amount_Term'] = df_clean['Loan_Amount_Term'].fillna(df_clean['Loan_Amount_
df_clean['Credit_History'] = df_clean['Credit_History'].fillna(df_clean['Credit_History']

df_clean.dropna(inplace=True)
df_clean.head()
```

	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	Coapplican
0	Male	No	0	Graduate	No	5849	
1	Male	Yes	1	Graduate	No	4583	
2	Male	Yes	0	Graduate	Yes	3000	
3	Male	Yes	0	Not Graduate	No	2583	
4	Male	No	0	Graduate	No	6000	

Model

```
# Setup PyCaret
clf1 = setup(data=df_clean, target='Loan_Status', session_id=123)

# Compare all models
#compare_models()

compare_models(include=['rf', 'xgboost'])
```

```
compare_models(include=['rf','xgboost'])
```

	Description	Value
0	Session id	123
1	Target	Loan_Status
2	Target type	Binary
3	Target mapping	N: 0, Y: 1
4	Original data shape	(614, 12)
5	Transformed data shape	(614, 17)
6	Transformed train set shape	(429, 17)
7	Transformed test set shape	(185, 17)
8	Numeric features	5
9	Categorical features	6
10	Preprocess	True
11	Imputation type	simple
12	Numeric imputation	mean
13	Categorical imputation	mode
14	Maximum one-hot encoding	25
15	Encoding method	None
16	Fold Generator	StratifiedKFold
17	Fold Number	10
18	CPU Jobs	-1
19	Use GPU	False
20	Log Experiment	False
21	Experiment Name	clf-default-name
22	USI	8a69

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
rf	Random Forest Classifier	0.7904	0.7597	0.7904	0.7935	0.7717	0.4508	0.4822	1.1320
xgboost	Extreme Gradient Boosting	0.7693	0.7645	0.7693	0.7637	0.7576	0.4208	0.4344	0.8130

```
▾ RandomForestClassifier
RandomForestClassifier(bootstrap=True, ccp_alpha=0.0, class_weight=None,
                       criterion='gini', max_depth=None, max_features='sqrt',
                       max_leaf_nodes=None, max_samples=None,
                       min_impurity_decrease=0.0, min_samples_leaf=1,
```

Model Comparison:

Random Forest outperformed XGBoost in overall accuracy (0.7904 vs 0.7693), precision (0.7935 vs 0.7637), and F1-score (0.7717 vs 0.7576). XGBoost achieved a slightly higher AUC score (0.7645 vs 0.7597), indicating it may perform slightly better in distinguishing between classes under imbalance.

Final Decision: Random Forest was selected as the final model due to its superior performance in accuracy, precision, and F1-score — key metrics for the loan approval prediction.

Interpretation: The most influential features in the Random Forest model were:

- Credit_History: Applicants with good credit history had significantly higher approval chances.
- LoanAmount: Higher requested amounts generally correlated with a lower approval probability.
- ApplicantIncome: Income had moderate influence, especially when considered with Coapplicant Income.

Evaluate

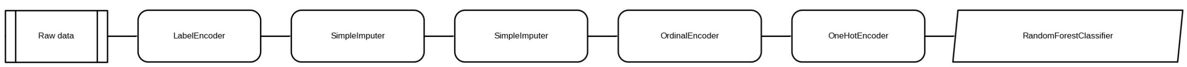
```
# Create and evaluate a Random Forest model
rf_model = create_model('rf')
evaluate_model(rf_model)
predict_model(rf_model)
```

Fold	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	0.8140	0.8077	0.8140	0.8212	0.7945	0.4926	0.5327
1	0.6977	0.6859	0.6977	0.6694	0.6733	0.1957	0.2058
2	0.7674	0.6436	0.7674	0.7743	0.7294	0.3323	0.3931
3	0.7907	0.8449	0.7907	0.8390	0.7489	0.3828	0.4865
4	0.7907	0.8154	0.7907	0.7867	0.7882	0.4928	0.4936
5	0.7907	0.7241	0.7907	0.7888	0.7763	0.4749	0.4965
6	0.7907	0.7303	0.7907	0.7888	0.7763	0.4749	0.4965

7	0.8372	0.7734	0.8372	0.8448	0.8260	0.5916	0.6185
8	0.7674	0.6749	0.7674	0.7645	0.7466	0.4044	0.4330
9	0.8571	0.8966	0.8571	0.8571	0.8571	0.6658	0.6658
Mean	0.7904	0.7597	0.7904	0.7935	0.7717	0.4508	0.4822
Std	0.0411	0.0773	0.0411	0.0512	0.0487	0.1251	0.1192

Plot Type:

Pipeline Plot	Hyperparameters	AUC	Confusion Matrix
Threshold	Precision Recall	Prediction Error	Class Report
Feature Selection	Learning Curve	Manifold Learning	Calibration Curve
Validation Curve	Dimensions	Feature Importance	Feature Importance ...
Decision Boundary	Lift Chart	Gain Chart	Decision Tree
KS Statistic Plot			



	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	Random Forest Classifier	0.7892	0.7748	0.7892	0.8015	0.7622	0.4271	0.4797

	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	Coapplic
354	Female	Yes	0	Graduate	No	2423	
426	Female	No	1	Not Graduate	No	4606	
611	Male	Yes	1	Graduate	No	8072	
214	Male	Yes	0	Graduate	No	3173	
72	Male	No	0	Graduate	No	3500	
...	...	...	...	...	...	...	
574	Male	Yes	3+	Graduate	No	6406	
42	Male	Yes	0	Graduate	No	2400	
50	Female	Yes	0	Not Graduate	No	1928	
189	Male	Yes	0	Graduate	No	9328	
215	Male	Yes	3+	Not Graduate	No	3850	

185 rows × 14 columns