Final term project

Peiwen Zhao

Machine learning 5805

Table of content

Table of figures and tables

Abstract

Banks play a pivotal role in the economy by providing loans, which constitute a substantial portion of their revenue. However, lending inherently entails the risk of borrowers defaulting on their loans. To mitigate this risk, banks are increasingly adopting Machine Learning (ML) techniques to predict potential defaulters. In this project, our objective is to develop a robust ML model capable of classifying whether new borrowers are likely to default based on historical loan data.

The dataset utilized for this study, sourced from Kaggle, encompasses various deterministic factors that can influence loan defaults, including borrower income, gender, loan purpose, and more. Nevertheless, the dataset also presents challenges, such as multicollinearity among variables and missing values. To address these challenges, we employ appropriate preprocessing techniques to clean and transform the data.

We subsequently employ diverse classification algorithms to construct and evaluate predictive models. We compare the performance of these models using standard metrics such as accuracy, precision, recall, and F1 score. Our analysis provides insights into the key factors that influence loan defaults and identifies the classifier that most effectively predicts potential defaulters.

The outcomes of this project offer actionable recommendations for banks to enhance their risk assessment processes. By identifying the most influential features and classifiers, banks can make informed lending decisions and minimize default-related losses.

The project is divided into four distinct phases:

Phase I: Feature Engineering and Exploratory Data Analysis (EDA)

This phase is dedicated to preparing the dataset for modeling by identifying the target variable and attributes. Key tasks include handling missing values, detecting and eliminating duplicates, transforming and encoding variables, reducing dimensionality, and addressing multicollinearity.

Phase II: Regression Analysis

In this phase, multiple linear regression is employed to predict a continuous numerical feature within the dataset. The analysis involves statistical tests, such as T-tests and F-tests, to evaluate the significance of the predictors. The model's performance is assessed using metrics like R-squared, Adjusted R-squared, AIC, BIC, and Mean Squared Error (MSE). Confidence intervals and stepwise regression are applied to fine-tune the model and ensure precise predictions.

Phase III: Classification Analysis

This phase encompasses applying various classification algorithms, including Decision Trees, Logistic Regression, K-Nearest Neighbors (KNN), Support Vector Machines (SVM), Naïve Bayes, Random Forest, and Neural Networks. Hyperparameter tuning is achieved through grid search. Model performance is assessed using metrics: Precision, Recall, F-score, ROC curves, and AUC. K-fold cross-validation is employed to prevent overfitting and ensure reliable performance. The most effective classifier is identified and recommended based on a thorough comparison of all models.

Phase IV: Clustering and Association Analysis

In this phase, independent research is conducted on clustering and association rule mining techniques. Clustering algorithms, such as K-means and DBSCAN, are applied to uncover patterns within the dataset. Silhouette analysis and within-cluster variation plots are used to determine the optimal number of clusters. And the Apriori algorithm is employed to discover association rules, offering valuable insights into the relationships within the data.

Description of the dataset

The dataset used in this project contains **148,670 rows** and **34 columns**, providing detailed information about loan applications. The dependent variable is "Status". Below is a comprehensive description of each variable in the dataset[1]:

Variable	Description
ID	client loan application id
year	year of loan application
loan_limit	indicates whether the loan is conforming (cf) or non-conforming (ncf)
Gender	gender of the applicant (male, female, joint, sex not available)
approv_in_adv	indicates whether the loan was approved in advance (pre, nopre)
loan_type	type of loan (type1, type2, type3)
loan_purpose	purpose of the loan (p1, p2, p3, p4)
Credit_Worthiness	credit worthiness (l1, l2)
open_credit	indicates whether the applicant has any open credit accounts (opc, nopc)
business_or_commercial	indicates whether the loan is for business/commercial purposes (ob/c - business/commercial, nob/c - personal)
loan_amount	amount of money being borrowed
rate_of_interest	interest rate charged on the loan
Interest_rate_spread	difference between the interest rate on the loan and a benchmark interest rate

Upfront_charges	initial charges associated with securing the loan
term	duration of the loan in months
Neg_ammortization	indicates whether the loan allows for negative ammortization (neg_amm, not_neg)
interest_only	indicates whether the loan has an interest-only payment option (int_only, not_int)
lump_sum_payment	indicates if a lump sum payment is required at the end of the loan term (lpsm, not_lpsm)
property_value	value of the property being financed
construction_type	type of construction (sb - site built, mh - manufactured home)
occupancy_type	occupancy type (pr - primary residence, sr- secondary residence, ir - investment property)
Secured_by	specifies the type of collateral securing the loan (home, land)
total_units	number of units in the property being financed (1U, 2U, 3U, 4U)
income	applicant's annual income
credit_type	applicant's type of credit (CIB - credit information bureau , CRIF - CIRF credit information bureau, EXP - experian , EQUI - equifax)
Credit_Score	applicant's credit score
co-applicant_credit_type	co-applicant's type of credit (CIB - credit information bureau EXP - experian)
age	the age of the applicant.

submission_of_application	indicates how the application was submitted (to_inst - to institution, not_inst - not to institution)
LTV	loan-to-value ratio, calculated as the loan amount divided by the property value
Region	geographic region where the property is located (North, south, central, North-East)
Security_Type	type of security or collateral backing the loan (direct, indirect)
Status	indicates whether the loan has been defaulted (1) or not (0)
dtir1	debt-to-income ratio

Phase 1

- **Handling Missing Values**:
  - **Numerical Columns with Missing Values**:
    - `rate_of_interest`
    - `Interest_rate_spread`
    - `Upfront_charges`
    - `term`, `income`
    - `dtir1`

```
Numerical Columns with missing values:
 ['rate_of_interest', 'Interest_rate_spread', 'Upfront_charges', 'term', 'income', 'dtir1']

Numerical Data missing percentage:
 Upfront_charges         18.369211
Interest_rate_spread    16.122197
rate_of_interest        15.972446
income                   6.834637
dtir1                    6.752274
term                     0.021714
dtype: float64
```

Picture 1.0 Description of the numerical columns with missing values

- Missing values were filled with appropriate imputation methods (mean/ median).[2][3]
- Mean:
  - `dtir1`
  - `Interest_rate_spread`
  - `rate_of_interest`
- Median:
  - `Income`
  - `LTV`
  - `property_value`
  - `Term`
  - `Upfront_charges`

- **Categorical Columns with Missing Values**:
  - Neg_ammortization
  - age
  - approv_in_adv
  - loan_limit
  - loan_purpose
  - submission_of_application

```
categorical Data missing percentage:
 loan_limit                 2.228295
approv_in_adv               0.607240
age                         0.149751
submission_of_application   0.149751
loan_purpose                0.087604
Neg_ammortization           0.084609
dtype: float64
```

Picture 1.1 Description of the categorical columns with missing values

  - Missing values were filled with the mode.
  - Mode:
    - Neg_ammortization
    - age
    - approv_in_adv
    - loan_limit
    - loan_purpose
    - submission_of_application
- **Categorical Columns encoding**:
  - Performed One hot encoding for all the categorical columns. [4]

- **Duplicate Data**:
  - No duplicate rows were found in the dataset. [5]

- **Data balance**:

Feature	Imbalance Ratio
loan_amount	0.00
rate_of_interest	0.00
Interest_rate_spread	0.00
Upfront_charges	0.00
term	0.00
property_value	0.00
income	0.00
Credit_Score	0.76
LTV	0.00
Status	0.19
dtir1	0.04
Credit_Worthiness_l2	0.04
Gender_Joint	0.39
Gender_Male	0.40
Gender_Sex Not Available	0.33
Neg_ammortization_not_neg	0.10
Region_North-East	0.01
Region_central	0.06
Region_south	0.75
Secured_by_land	0.00
Security_Type_direct	0.00
age_35-44	0.28
age_45-54	0.31
age_55-64	0.28
age_65-74	0.16
age_<25	0.01
age_>74	0.05
approv_in_adv_pre	0.19
business_or_commercial_nob/c	0.16
co-applicant_credit_type_EXP	0.80

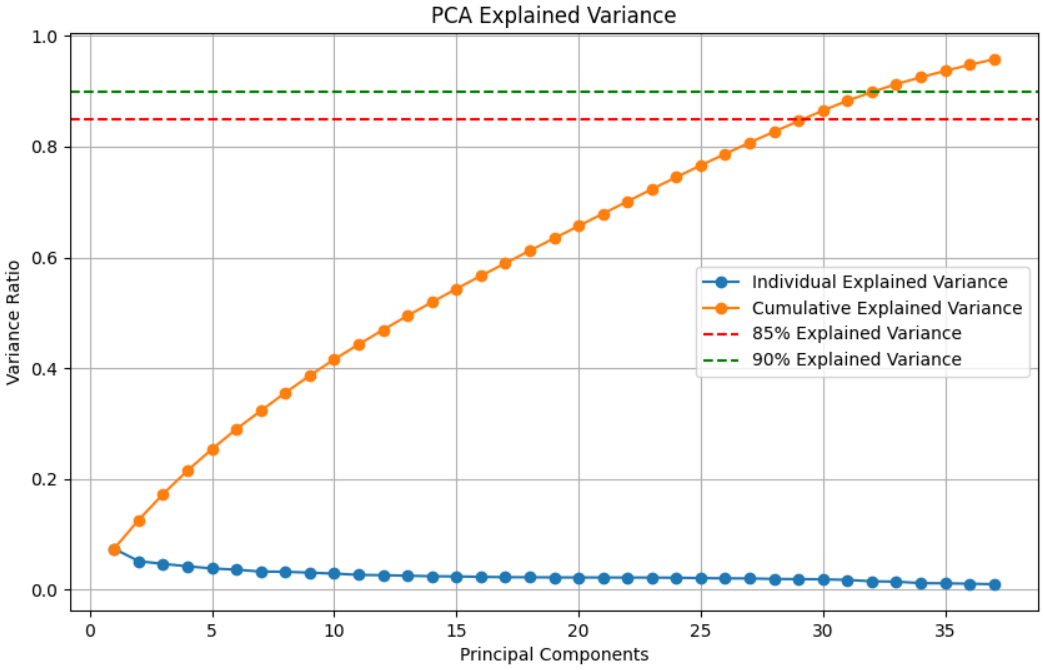
construction_type_sb	0.00
credit_type_CRIF	0.49
credit_type_EQUI	0.00
credit_type_EXP	0.45
interest_only_not_int	0.05
loan_limit_ncf	0.07
loan_purpose_p2	0.02
loan_purpose_p3	0.62
loan_purpose_p4	0.59
loan_type_type2	0.16
loan_type_type3	0.11
lump_sum_payment_not_lpsm	0.02
occupancy_type_pr	0.08
occupancy_type_sr	0.02
open_credit_opc	0.00
submission_of_application_to_inst	0.56
total_units_2U	0.01
total_units_3U	0.00
total_units_4U	0.00

- **Data collinearity**:[12][13]
    - Removed columns due to high collinearity(threshold = 0.85):
        - `Security_Type_direct`
        - `Construction_type_sb`
        - `loan_type_2`
- **Random Forest**:
    - Top ten important features:[6]

Feature Importance	rate_of_interest
rate_of_interest	0.386678
Interest_rate_spread	0.312017

Upfront_charges	0.198904
dtir1	0.019407
LTV	0.009439
income	0.009160
lump_sum_payment_not_lpsm	0.008954
business_or_commercial_nob/c	0.007872
submission_of_application_to_inst	0.007649
Neg_ammortization_not_neg	0.007031

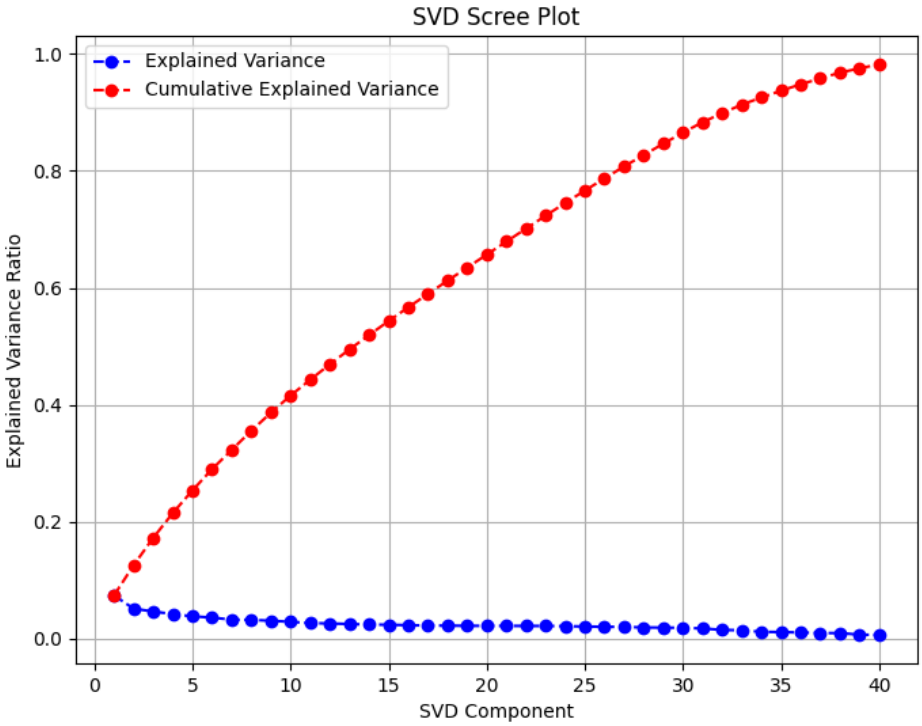
- PCA:[7]



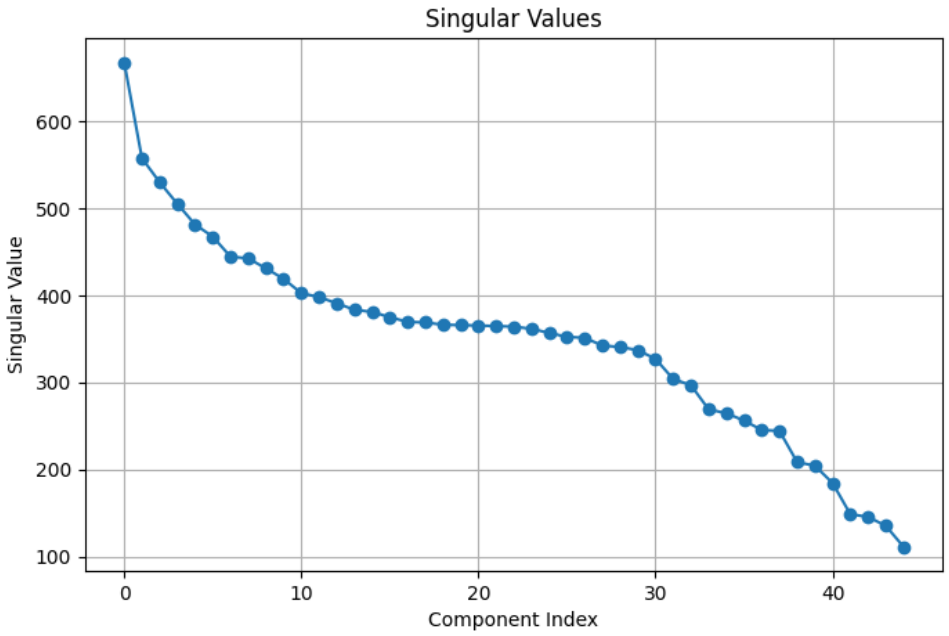
Picture 1.2 Result of PCA for cumulative explained variance and individual explained variance

- The first few principle components explain most of the variance.
- The cumulative explained variance reaches 90% with about 32 components.

- SVD:[8][9]
    - SVD analysis shows a similar pattern of variance distribution as PCA.

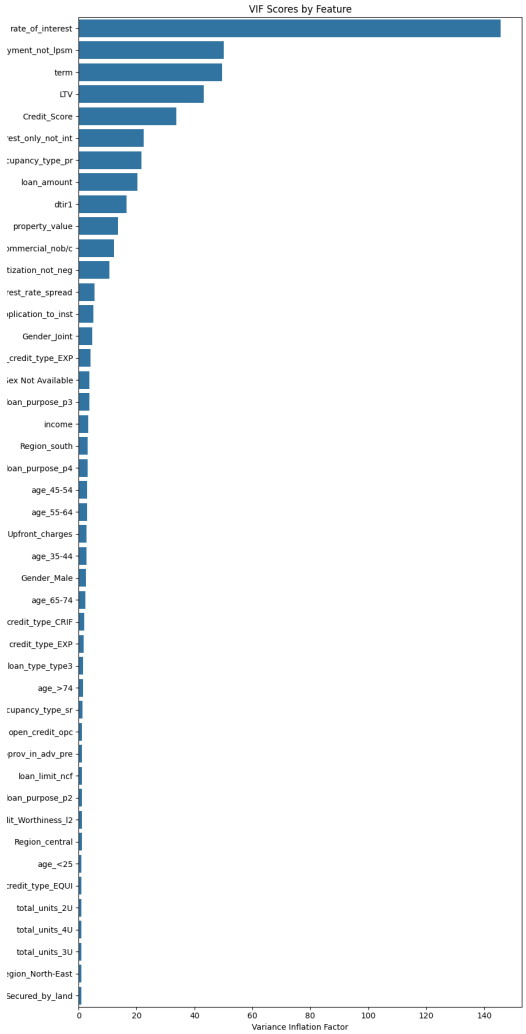


Picture 1.3 Result of SVD for explained variance vs cumulative explained variance



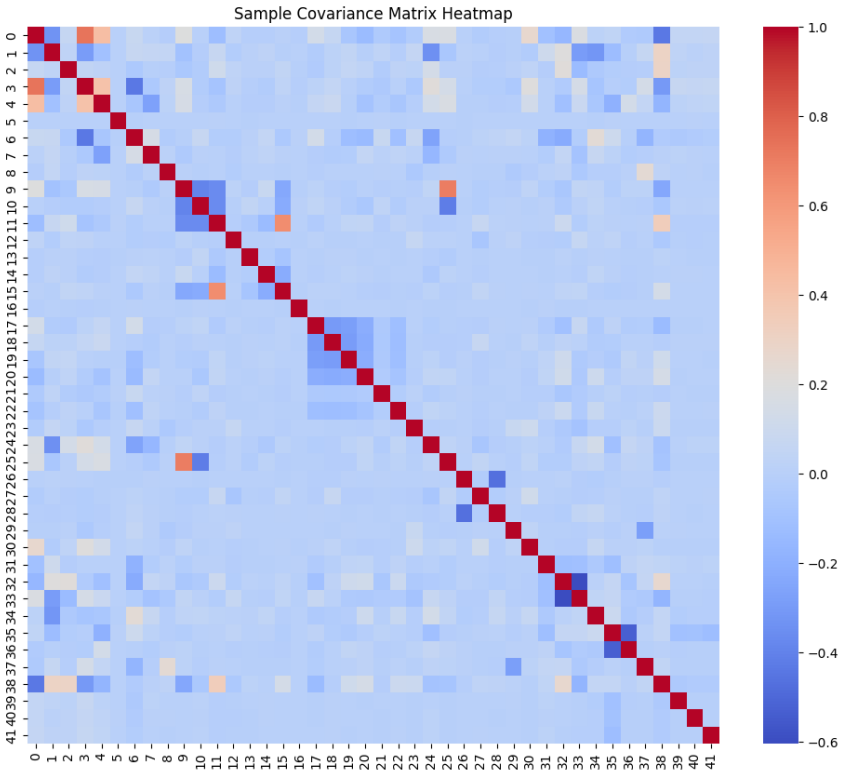
Picture 1.4 Result of Singular values for each component

- **VIF**:[10][11]
  - Identified features with high multicollinearity:
    - `rate_of_interest`
    - `Term`
    - `lump_sum_payments_not_lpsm`
  - Solution: Drop the selected features.



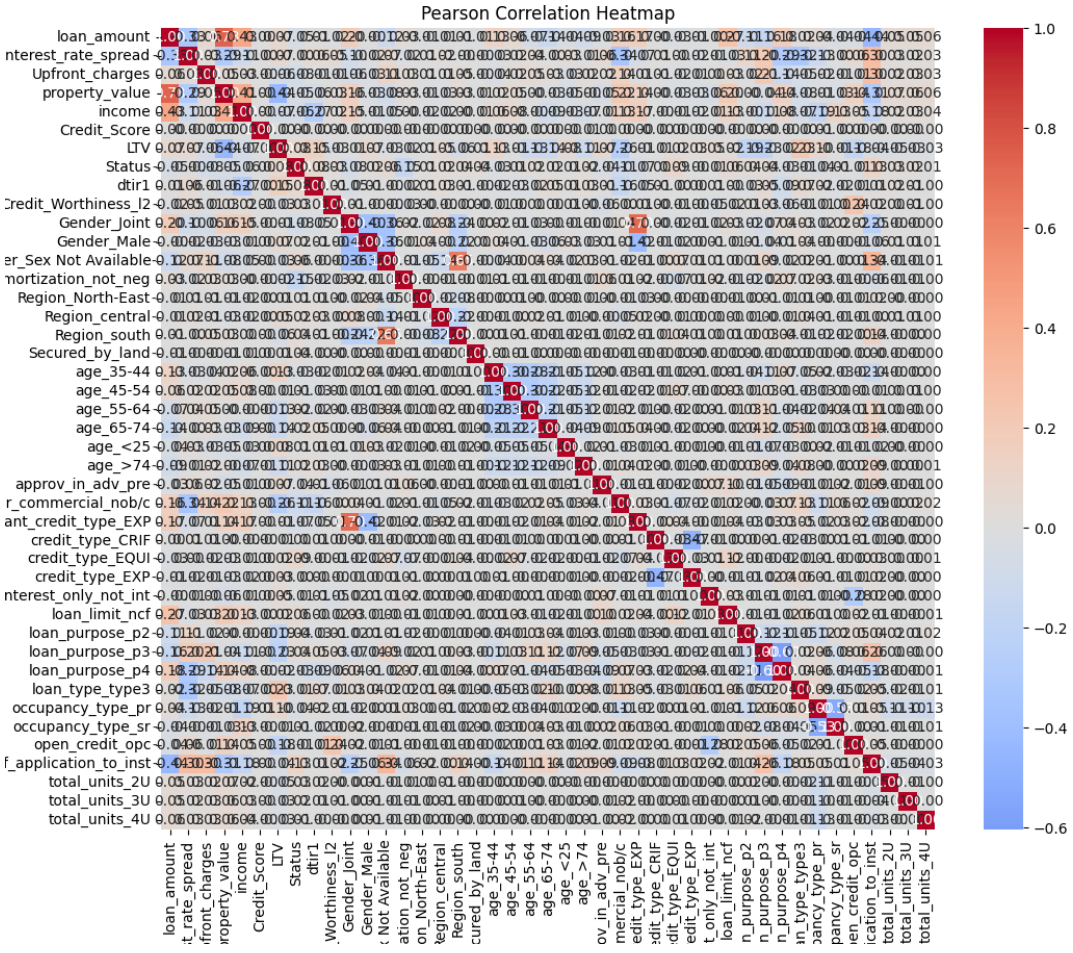
Picture 1.5 Result of VIF

- **Isolation Forest**:[14][15]
    - Detected 1336 anomalies in the dataset.
    - Detected 1336 anomalies.
- **Data Transformation**:
    - One hot encoding applied to all the categorical features and avoiding the dummy variable trap.
    - Standardization and Normalization applied to numerical features to scale the data uniformly.
- **Covariance Matrix Heatmap**:[16][17]
    - Most of the features have negative covariance relationship



Picture 1.6 Sample Covariance Matrix Heatmap

- **Pearson Correlation coefficients Matrix Heatmap**:[16][17]
    - Most of the features have close to zero Pearson correlation value.



Picture 1.7 Pearson correlation coefficients Matrix Heatmap

- **Data Imbalance**:
    - The target variable Status is imbalanced with a ratio of around 0.19.
    - Performed downsampling to the dataset for a balanced target variable.

Phase 2

- **Target**:
    - Aiming to predict the continuous variable Interest_rate_spread using multi-linear regression model.
- Model Summary:[18][19][20]

R-Squared	0.408
Adjusted R-squared	0.408
AIC	85468.37
BIC	85880.27
MSE	0.3131254
F-test(p-value)	0

    - R-squared value means the model explains 40.8% of the variance of the target variable.
    - With a value of 0.408 for the adjusted R-squared , the model remains robust after training.
    - The f test value shows the model is statistically significant.

```
                            OLS Regression Results
==============================================================================
Dep. Variable:     Interest_rate_spread   R-squared:                       0.408
Model:                              OLS   Adj. R-squared:                  0.408
Method:                   Least Squares   F-statistic:                     1756.
Date:                  Sun, 08 Dec 2024   Prob (F-statistic):               0.00
Time:                          18:19:05   Log-Likelihood:                -42691.
No. Observations:                106844   AIC:                         8.547e+04
Df Residuals:                    106801   BIC:                         8.588e+04
Df Model:                            42
Covariance Type:             nonrobust
```

Picture 2.0 OLS Regression Summary

- The t value of many variables are quite high, shows many variables are statistically significant.

	coef	std err	t	P>|t|	[0.025	0.975]
const	0.7485	0.012	60.089	0.000	0.724	0.773
business_or_commercial_nob/c	-0.2860	0.004	-79.131	0.000	-0.293	-0.279
loan_purpose_p2	0.2571	0.008	30.976	0.000	0.241	0.273
loan_purpose_p3	0.1081	0.003	32.028	0.000	0.101	0.115
property_value	-2.047e-08	6.74e-09	-3.038	0.002	-3.37e-08	-7.26e-09
loan_purpose_p4	-0.1076	0.003	-33.789	0.000	-0.114	-0.101
loan_type_type3	-0.5141	0.004	-126.576	0.000	-0.522	-0.506
submission_of_application_to_inst	0.2093	0.003	71.687	0.000	0.204	0.215
occupancy_type_pr	-0.3553	0.005	-64.692	0.000	-0.366	-0.345
occupancy_type_sr	-0.3865	0.009	-41.541	0.000	-0.405	-0.368
Status	-0.1281	0.003	-40.738	0.000	-0.134	-0.122
open_credit_opc	0.4304	0.019	22.618	0.000	0.393	0.468
LTV	0.0044	9.95e-05	44.262	0.000	0.004	0.005
loan_amount	-4.537e-07	1.32e-08	-34.399	0.000	-4.8e-07	-4.28e-07
Upfront_charges	-8.382e-06	4.21e-07	-19.912	0.000	-9.21e-06	-7.56e-06
loan_limit_ncf	0.0980	0.005	20.588	0.000	0.089	0.107
credit_type_EQUI	-0.3862	0.030	-13.017	0.000	-0.444	-0.328
total_units_2U	0.1436	0.011	12.551	0.000	0.121	0.166
total_units_4U	0.2589	0.024	10.687	0.000	0.211	0.306
Credit_Worthiness_l2	0.0522	0.006	9.081	0.000	0.041	0.063
total_units_3U	0.1903	0.022	8.568	0.000	0.147	0.234
Region_south	-0.0164	0.002	-7.037	0.000	-0.021	-0.012
dtir1	0.0010	0.000	8.198	0.000	0.001	0.001
income	1.241e-06	2.19e-07	5.668	0.000	8.12e-07	1.67e-06
age_<25	0.0825	0.012	6.824	0.000	0.059	0.106
Region_central	0.0275	0.005	5.660	0.000	0.018	0.037
age_45-54	0.0237	0.003	7.742	0.000	0.018	0.030
age_55-64	0.0213	0.003	6.851	0.000	0.015	0.027
Neg_ammortization_not_neg	0.0109	0.004	2.837	0.005	0.003	0.018
age_35-44	0.0089	0.003	2.836	0.005	0.003	0.015

Picture 2.1 T-test summary

- **Final Model coefficient** :

	coef
const	0.7485
business_or_commercial_nob/c	-0.2860
loan_purpose_p2	0.2571
loan_purpose_p3	0.1081
property_value	-2.047e-08
loan_purpose_p4	-0.1076
loan_type_type3	-0.5141
submission_of_application_to_inst	0.2093
occupancy_type_pr	-0.3553
occupancy_type_sr	-0.3865
Status	-0.1281
open_credit_opc	0.4304
LTV	0.0044
loan_amount	-4.537e-07
Upfront_charges	-8.382e-06
loan_limit_ncf	0.0980
credit_type_EQUI	-0.3862
total_units_2U	0.1436
total_units_4U	0.2589
Credit_Worthiness_l2	0.0522
total_units_3U	0.1903
Region_south	-0.0164
dtir1	0.0010
income	1.241e-06
age_<25	0.0825
Region_central	0.0275
age_45-54	0.0237
age_55-64	0.0213
Neg_ammortization_not_neg	0.0109
age_35-44	0.0089

Picture 2.2 Final regression model and prediction of dependent variable

- **Confidence interval analysis** :
    - Not significant predictors:
        - Secured_by_land
        - age_>74
        - credit_type_CRIF
        - credit_type_EXP

- **Stepwise regression and adjusted R-square analysis** :[21]
    - With threashold value of 0.01, the following predictors are included:

```
business_or_commercial_nob/c
loan_purpose_p2
loan_purpose_p3
property_value
loan_purpose_p4
loan_type_type3
submission_of_application_to_inst
occupancy_type_pr
Status
open_credit_opc
LTV
loan_amount
Upfront_charges
loan_limit_ncf
credit_type_EQUI
total_units_2U
total_units_4U
Credit_Worthiness_l2
total_units_3U
Region_south
dtir1
income
age_<25
Region_central
age_45-54
```

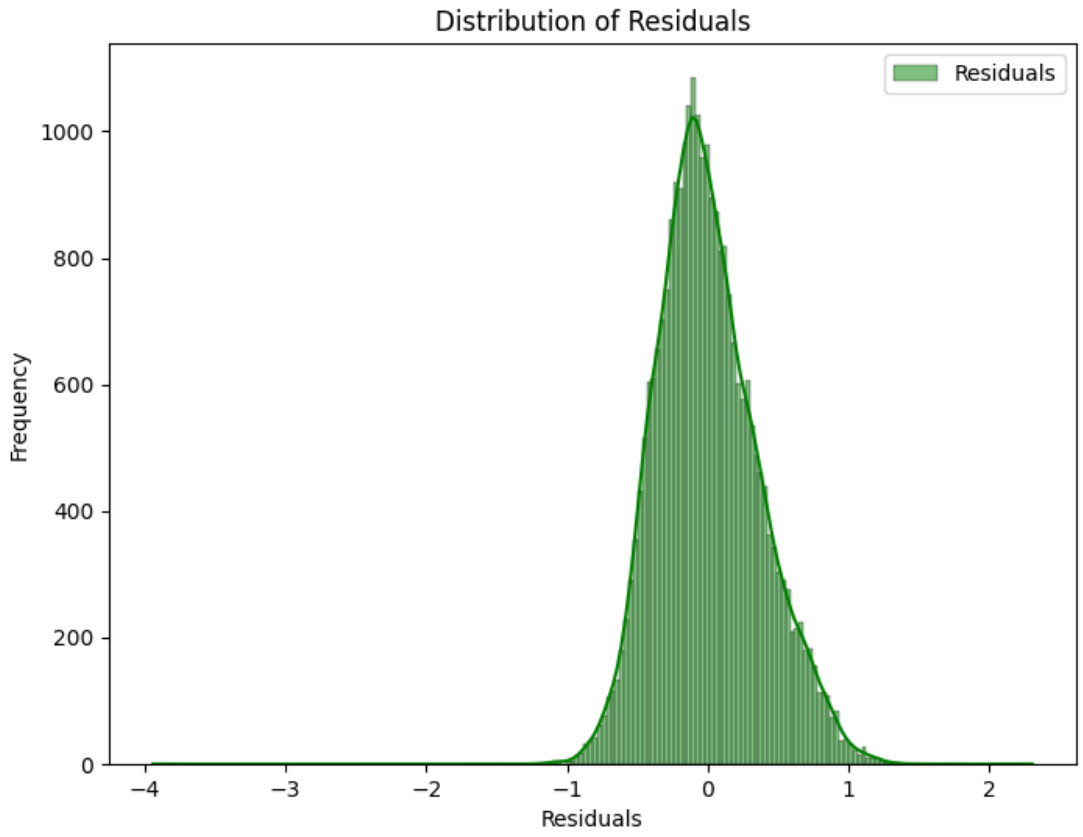
```
age_55-64
Neg_ammortization_not_neg
age_35-44
```

- Comparison between initial model and final model:

Initial model Adjusted R-squared	0.408
Final model after stepwise selection Adjusted R-squared	0.408057

  - The adjusted R-squared value remained almost the same, shows that the selected predictors from the stepwise selection explain the same amount of variance for the dependent variable.
  - The slight increase in Adjusted R-squared means that the selected predictors are optimal for maintaining model performance.

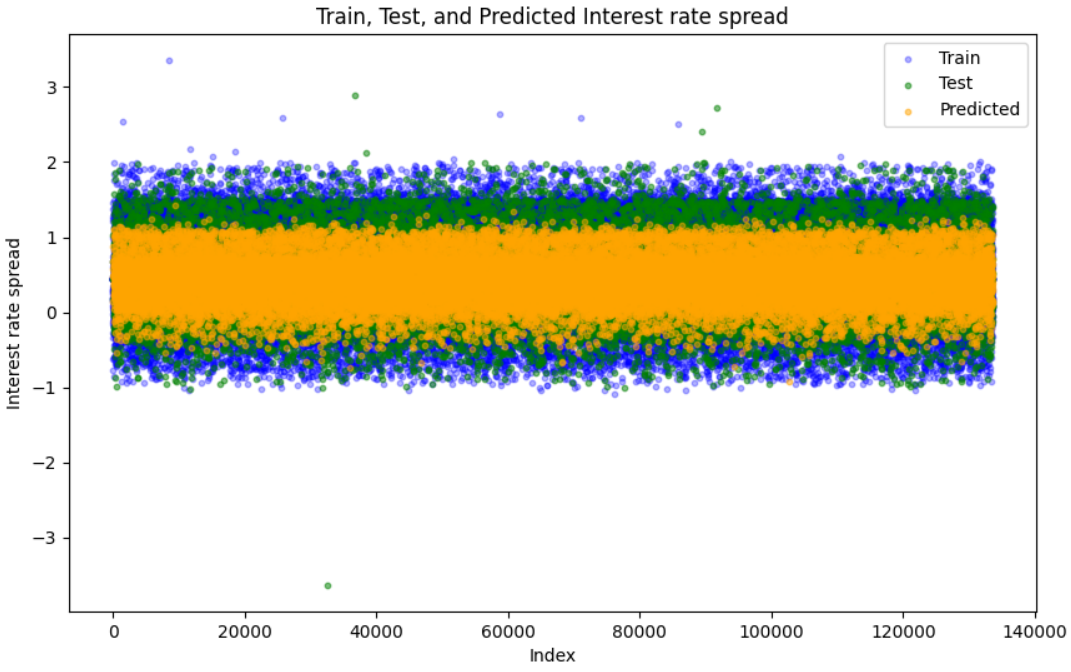
- **Distribution of Residuals** :



Picture 2.3 Distribution graph of the residuals.

- The bell shaped residuals distribution shows that the residuals are centered around zero.
- The normal distribution of residuals shows a well-fitted model that captures the patterns in the data.
- The distribution shows the model assumption for the dependent variable holds.

- **Train, test, and predicted interest rate spread result** :



Picture 2.4 Graph for Train, Test, and Predicted Interest rate spread

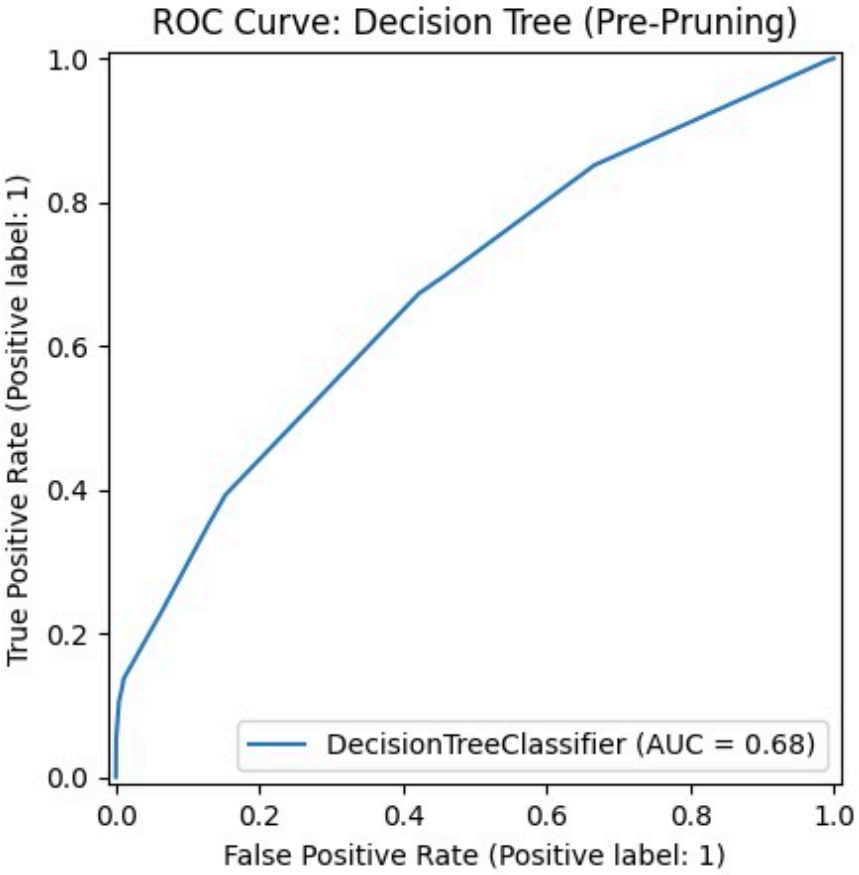
Phase 3 classification analysis

- **Target**:
    - Aiming to classify the variable Status using different classification models.

- **Decision Tree with pre-pruning**:[22][23][24][25]
    - Confusion Matrix:

```
[[2489 1817]
[1410 2897]]
```

Picture 3.0 decision Tree with pre-pruning confusion matrix

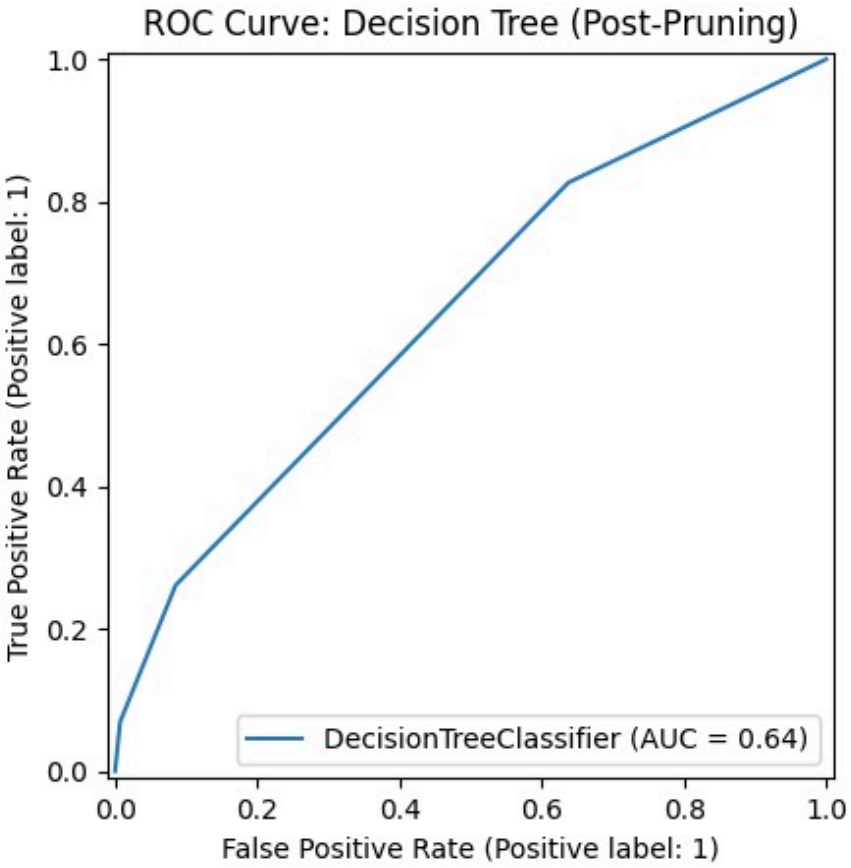


Picture 3.1 decision Tree with pre-pruning ROC curve

- **Decision Tree with post-pruning**:[22][24][25]
    - Confusion Matrix:

```
[[1560 2746]
[ 746 3561]]
```

Picture 3.2 decision Tree with post-pruning confusion matrix

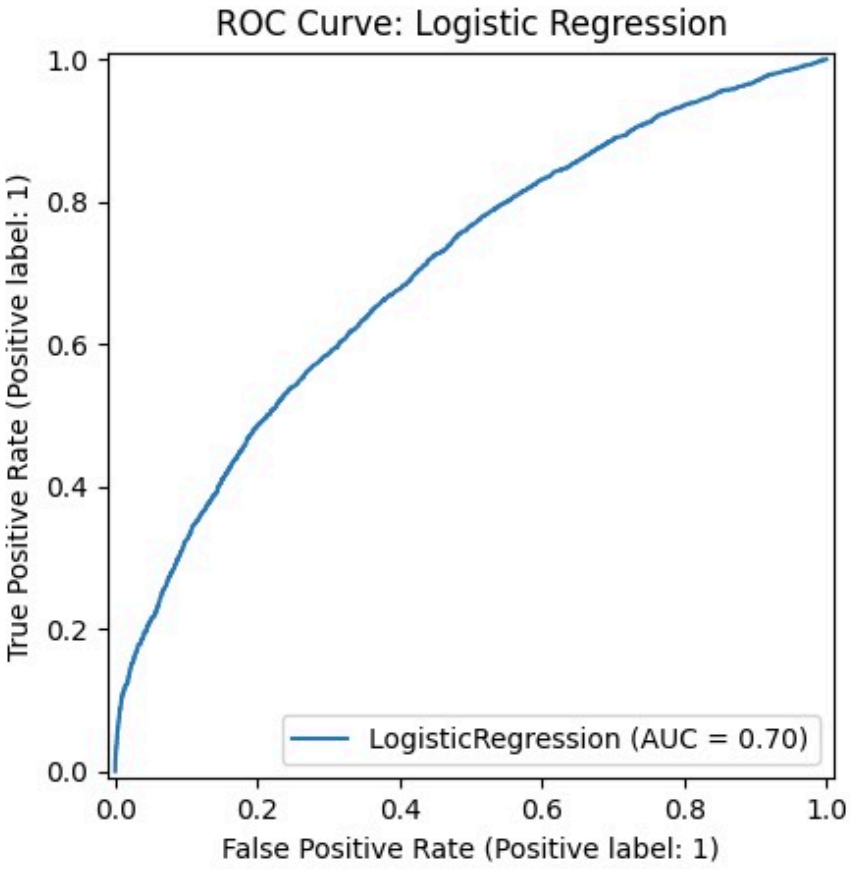


Picture 3.3 decision Tree with post-pruning ROC curve

- **Logistic regression**:**[26][27]**
    - Confusion Matrix:

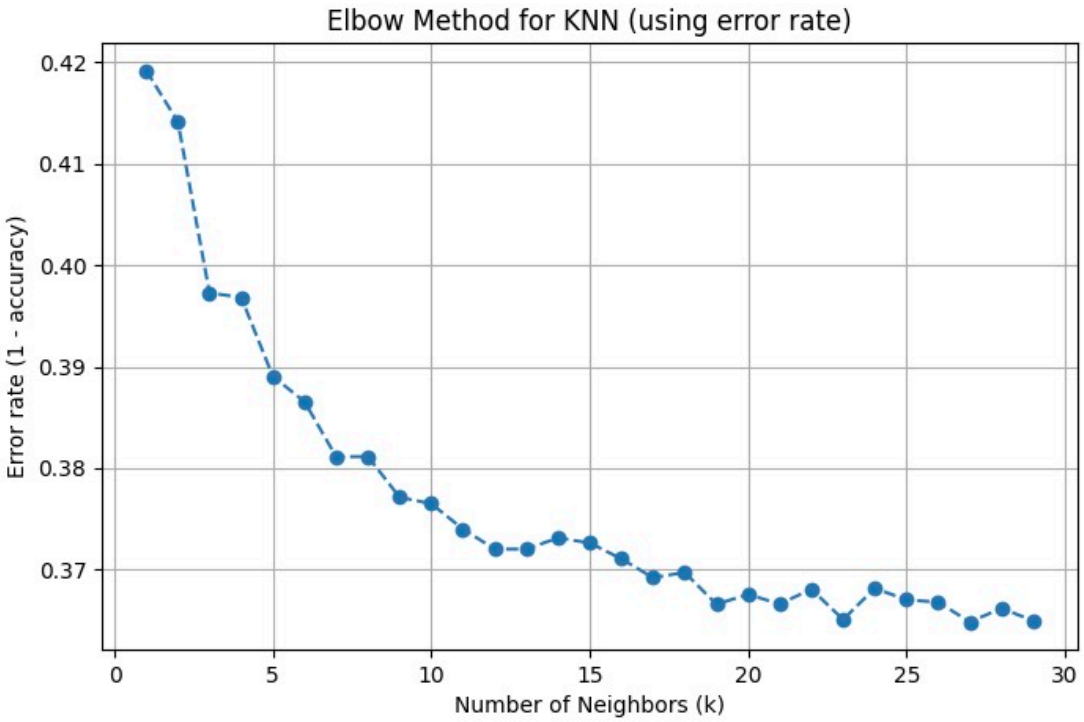
```
[[3132 1174]
[1877 2430]]
```

Picture 3.4 Logistic regression confusion matrix



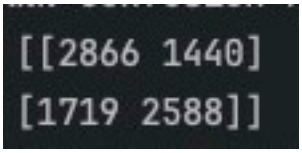
Picture 3.5 Logistic regression ROC curve

- **KNN**:[28]
    - Optimum K:
        - K = 19

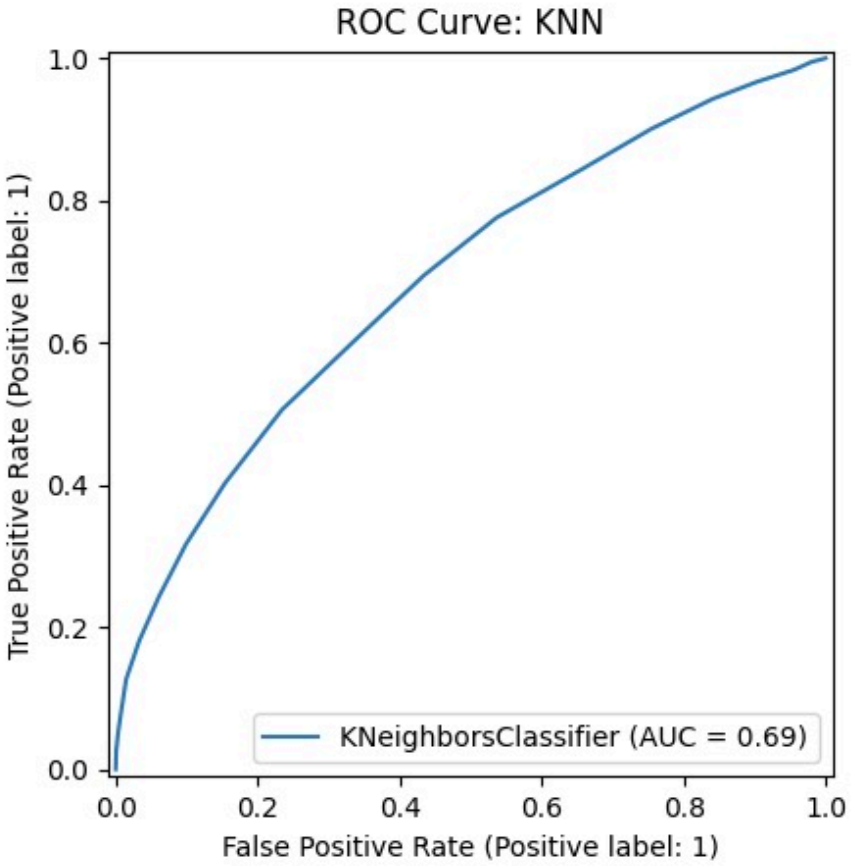


Picture 3.6 KNN optimum K with elbow method

- Confusion Matrix:



Picture 3.7 KNN confusion matrix

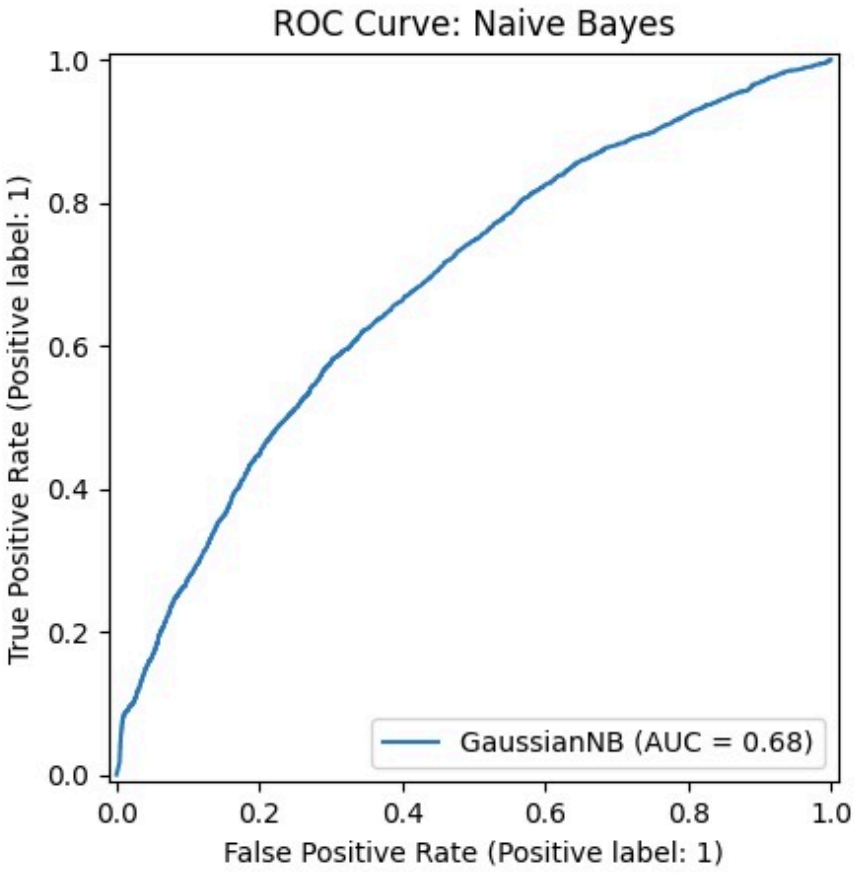


Picture 3.8 KNN ROC curve

- **Naive Bayes**:[29]
    - Confusion Matrix:

```
[[4259   47]
[3942  365]]
```

Picture 3.9 Naive Bayes confusion matrix

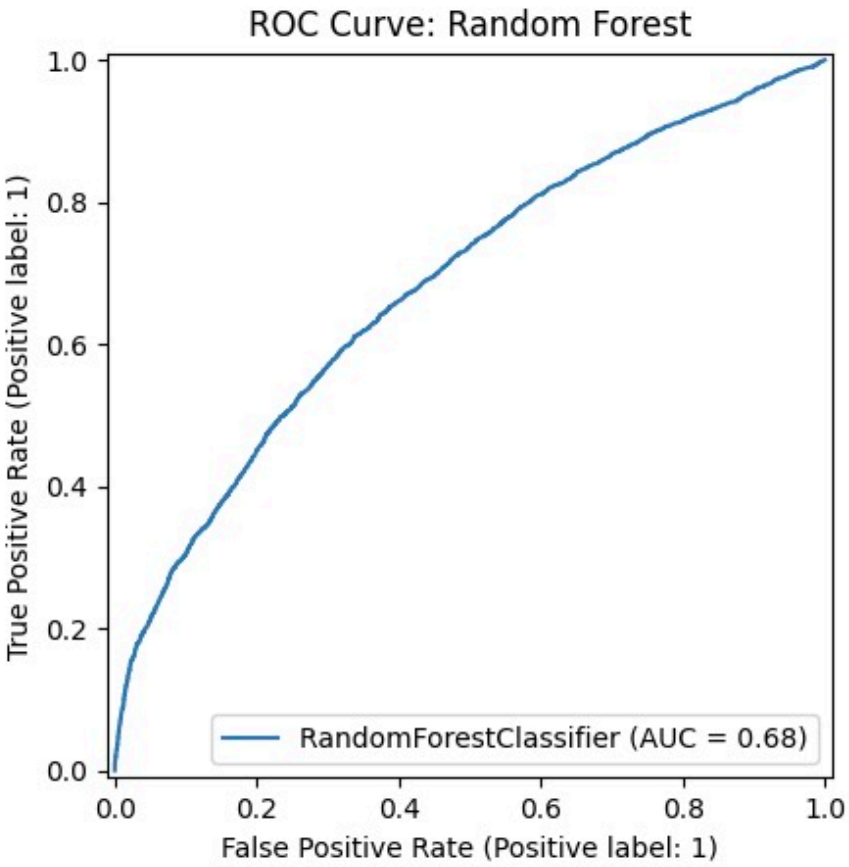


Picture 3.10 Naive Bayes ROC curve

- **Random Forest**:[30]
    - Confusion Matrix:

```
[[2927 1379]
[1759 2548]]
```

Picture 3.11 Random Forest confusion matrix

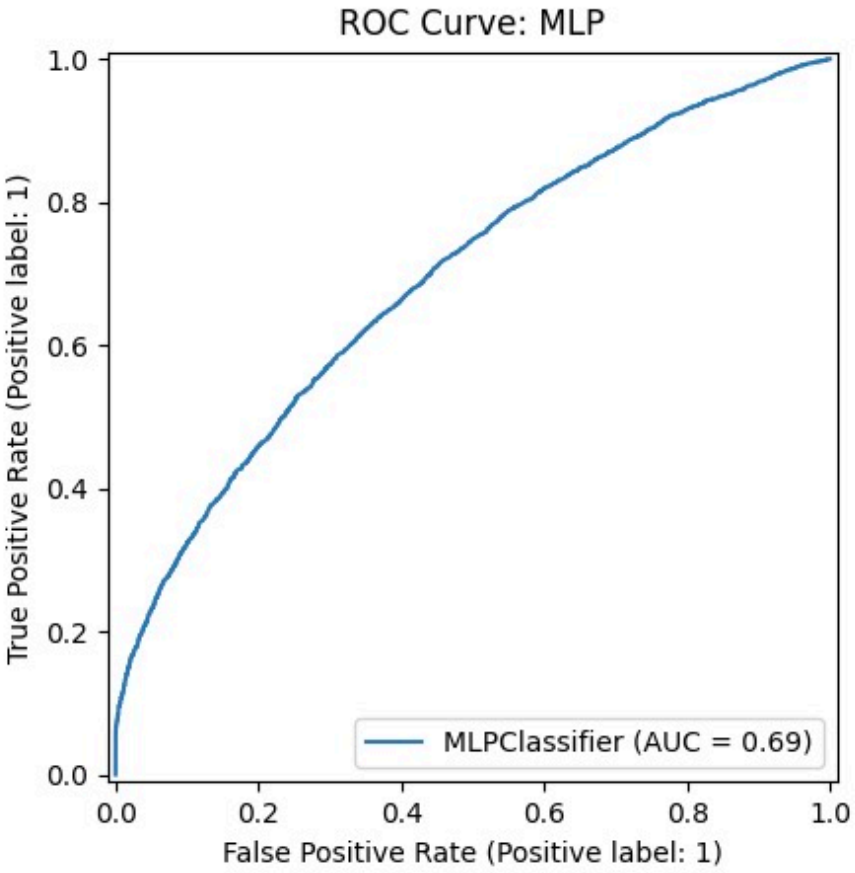


Picture 3.12 Random Forest ROC curve

- **Neural Network with MLP**:[31][32]
    - Confusion Matrix:

```
[[3002 1304]
[1827 2480]]
```

Picture 3.13 Neural Network with MLP confusion matrix

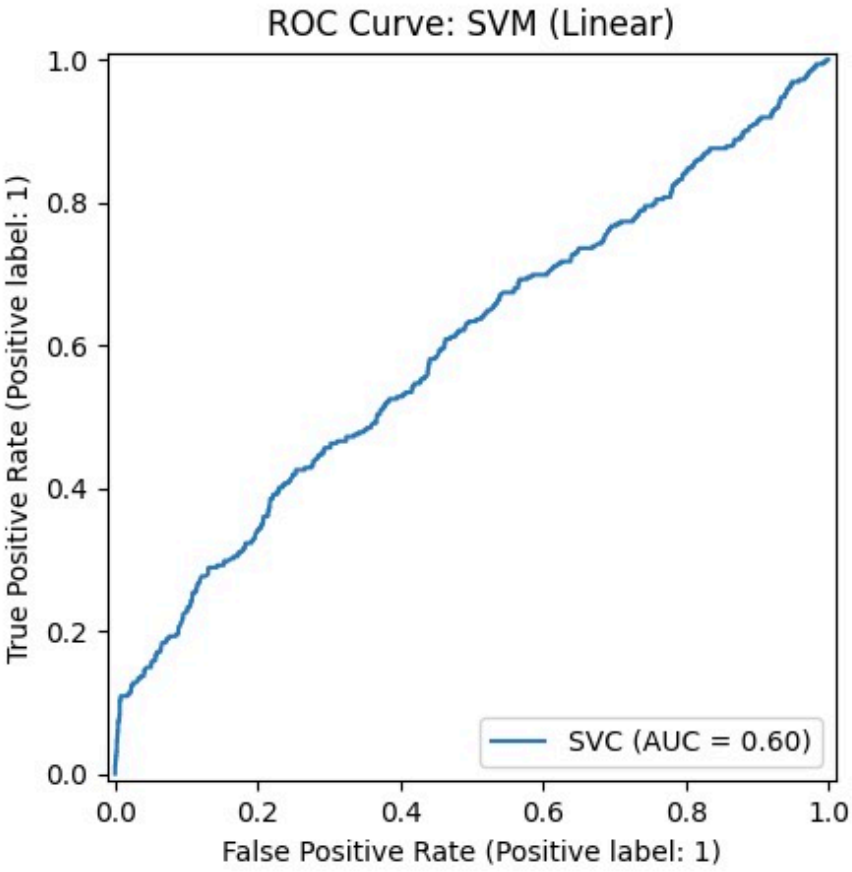


Picture 3.14 Neural Network with MLP ROC curve

- **SVM with Linear Kernel**:[33][34]
    - Confusion Matrix:

```
[[1667   11]
[ 289   33]]
```

Picture 3.15 SVM with linear kernel confusion matrix

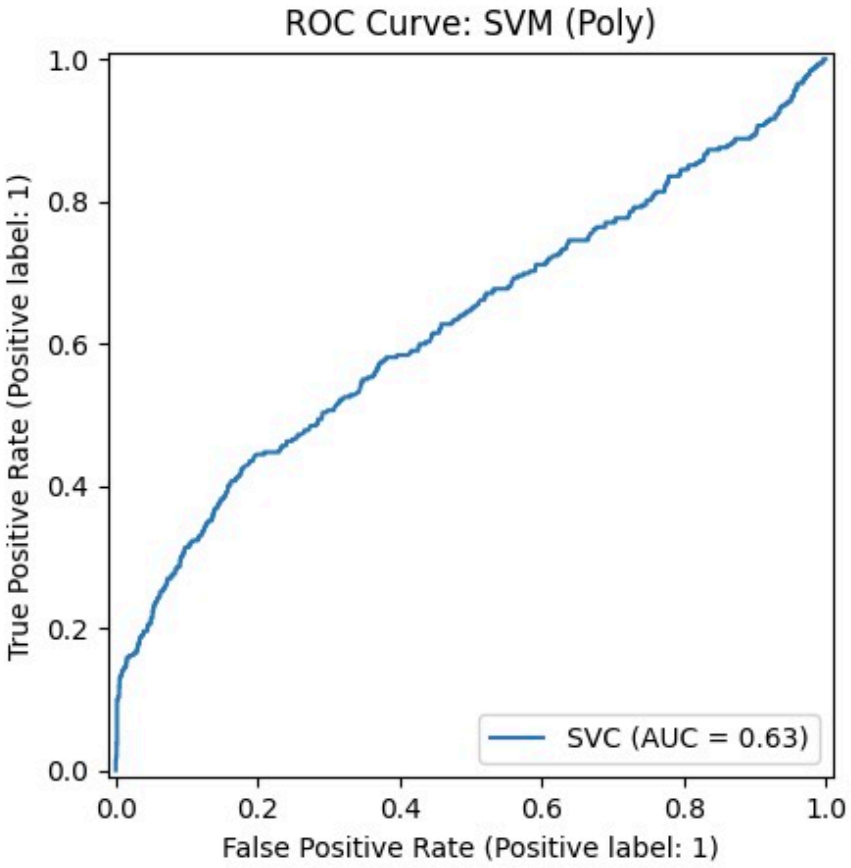


Picture 3.16 SVM with linear kernel ROC curve

- **SVM with Polynomial Kernel**:[33][34][35]
    - Confusion Matrix:

```
[[1657   21]
[ 276   46]]
```

Picture 3.17 SVM with polynomial kernel confusion matrix

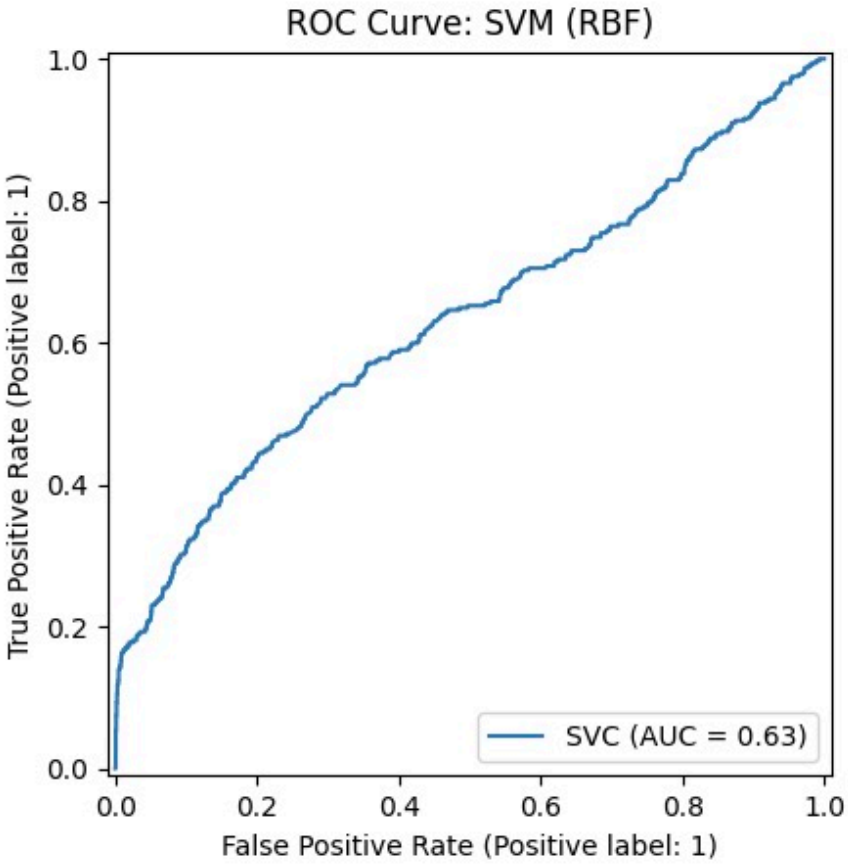


Picture 3.18 SVM with polynomial kernel ROC curve

- **SVM with radial base Kernel**:[33][34][36][37]
- Confusion Matrix:

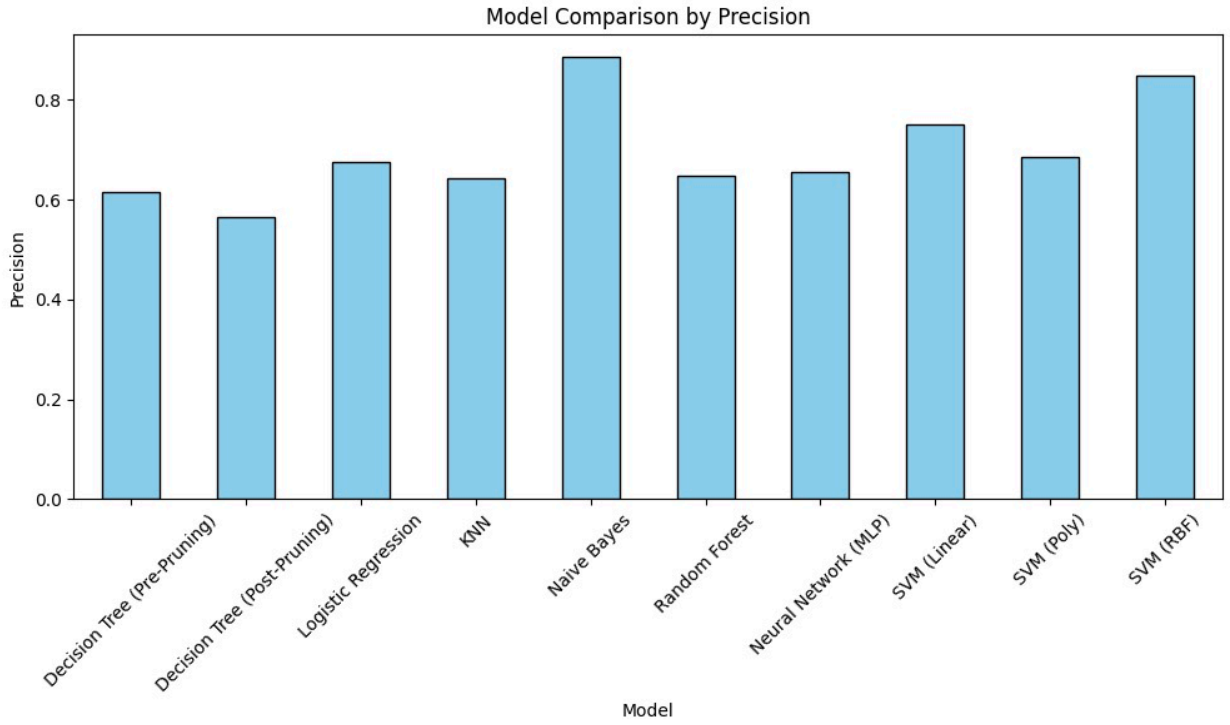
```
[[1670    8]
 [ 277   45]]
```

Picture 3.19 SVM with radial base kernel confusion matrix



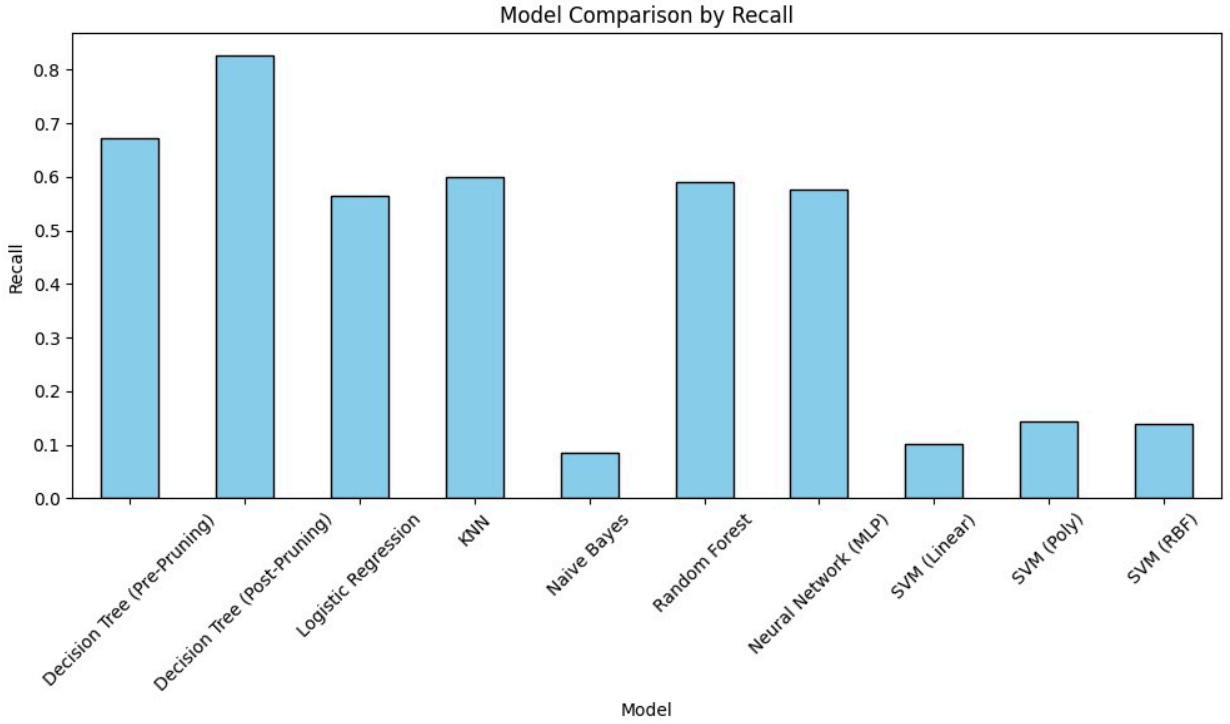
Picture 3.20 SVM with radial base ROC curve

- **Performance evaluation**:
    - Precision:



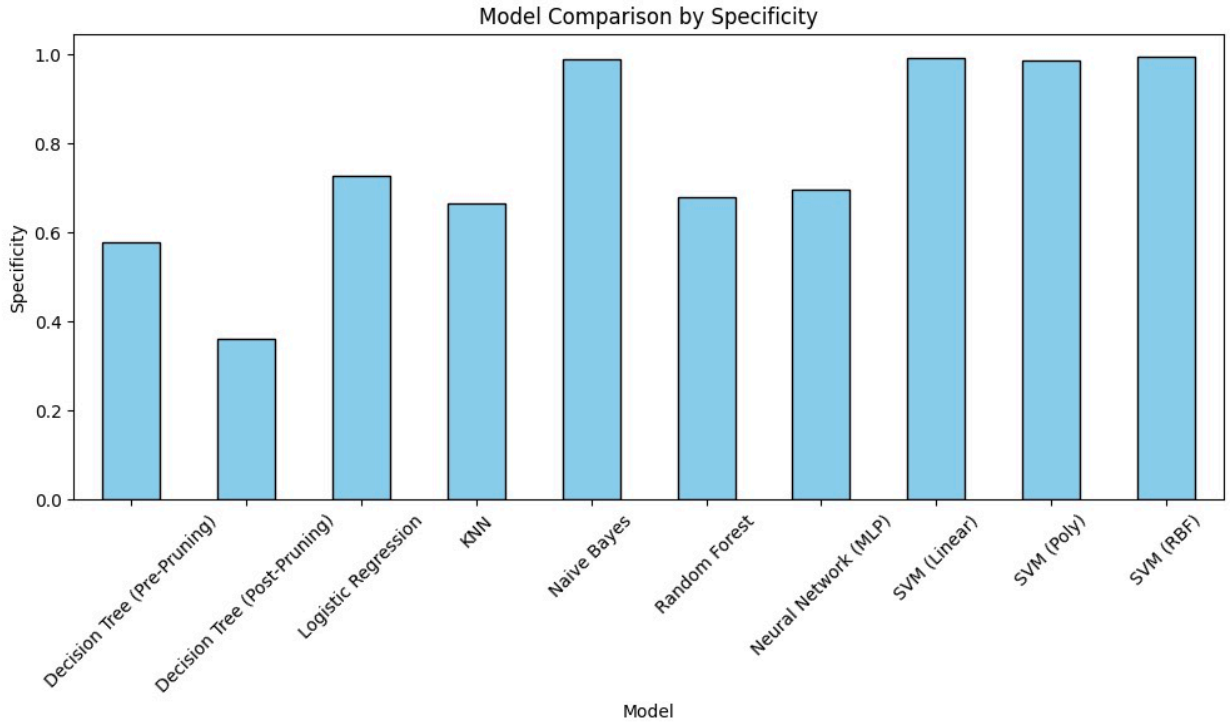
Picture 3.21 Model Precision comparison

- Recall:



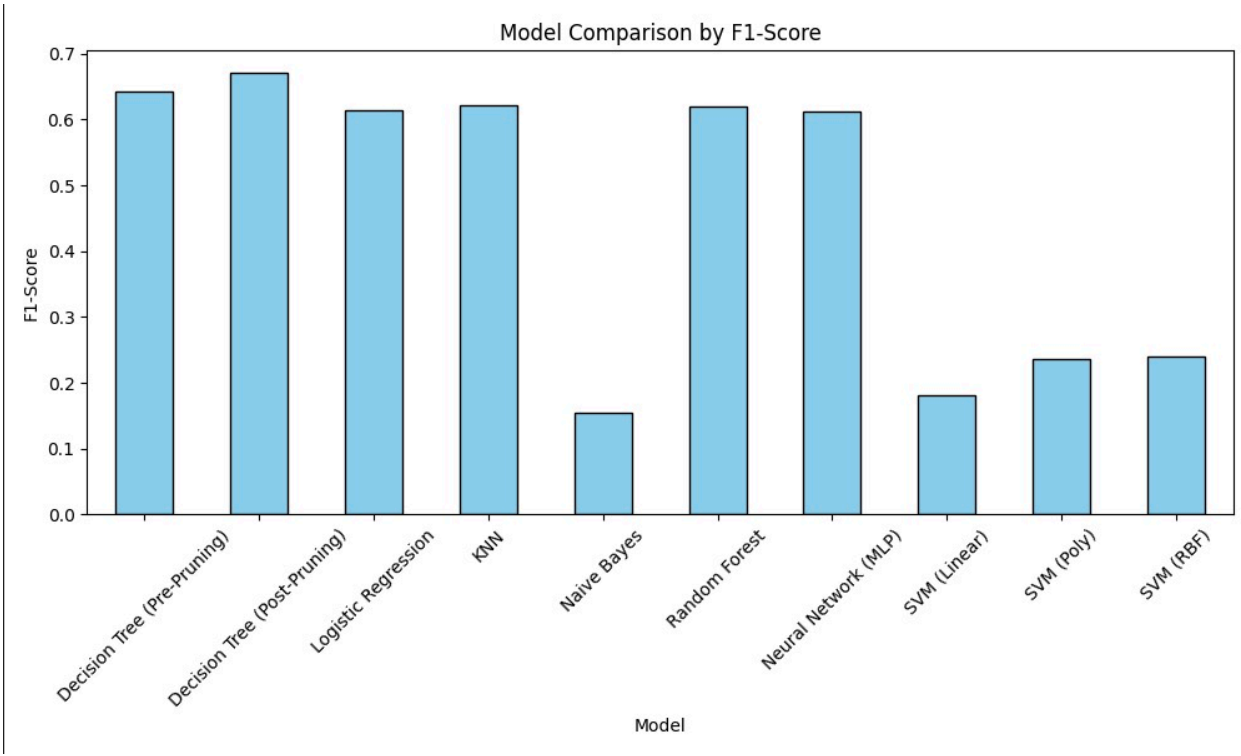
Picture 3.22 Model Recall comparison

- Specificity:



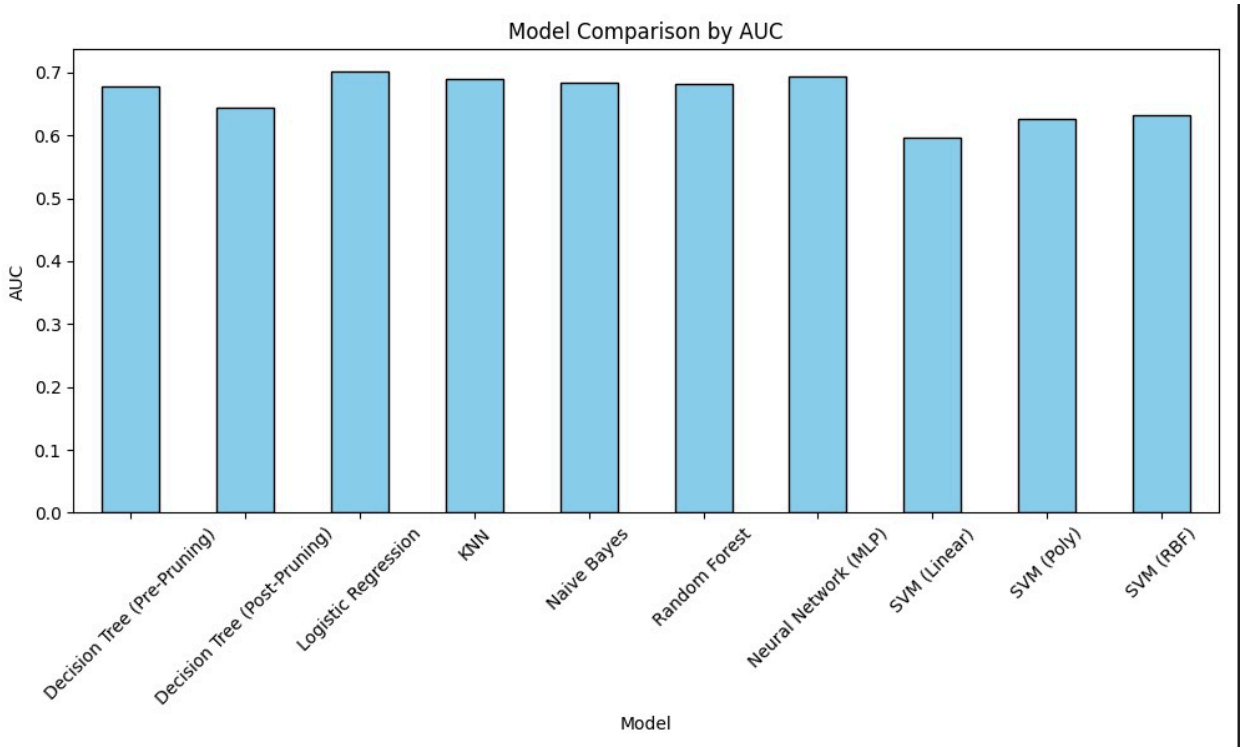
Picture 3.23 Model Specificity comparison

- F1-score:



Picture 3.24 Model F1-score comparison

- AUC



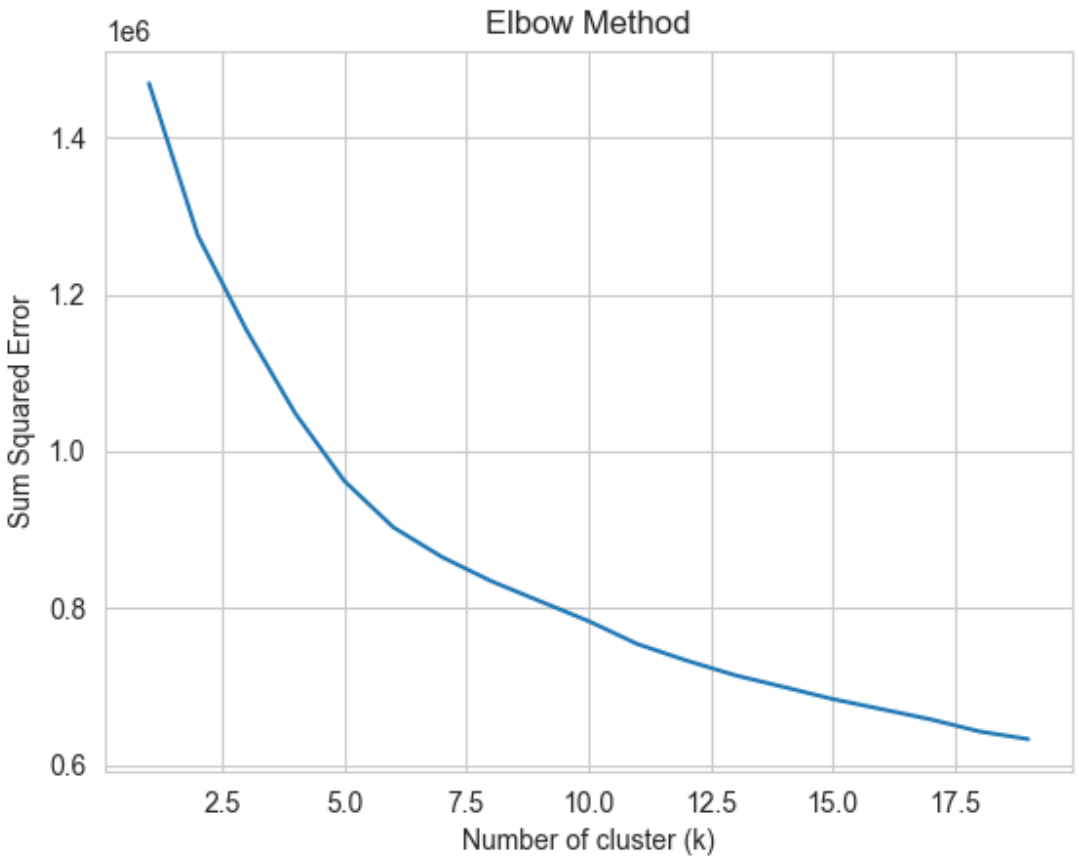
Picture 3.25 Model AUC comparison

- **Performance Summary**
    - Best model: Logistic Regression
        - Logistic regression has the best over all performance because it has the best combination of the AUC, precision, and F1-score across all the models.

Classifier	Precision	Recall	Specificity	F1-Score	AUC
Decision Tree (Pre-Pruning)	0.6146	0.6726	0.5780	0.6423	0.6780
Decision Tree (Post-Pruning)	0.5646	0.8268	0.3623	0.6710	0.6446
Logistic Regression	0.6743	0.5642	0.7274	0.6143	0.7017
KNN	0.6425	0.6009	0.6656	0.6210	0.6889
Naive Bayes	0.8859	0.0847	0.9890	0.1547	0.6844
Random Forest	0.6488	0.5916	0.6797	0.6189	0.6827
Neural Network	0.6554	0.5758	0.6972	0.6130	0.6927
SVM (RBF Kernel)	0.8491	0.1398	0.9952	0.2400	0.6317
SVM (Polynomial Kernel)	0.6866	0.1429	0.9875	0.2365	0.6270
SVM (Linear Kernel)	0.7500	0.1028	0.9934	0.1803	0.5958

Phase 4 clustering and association

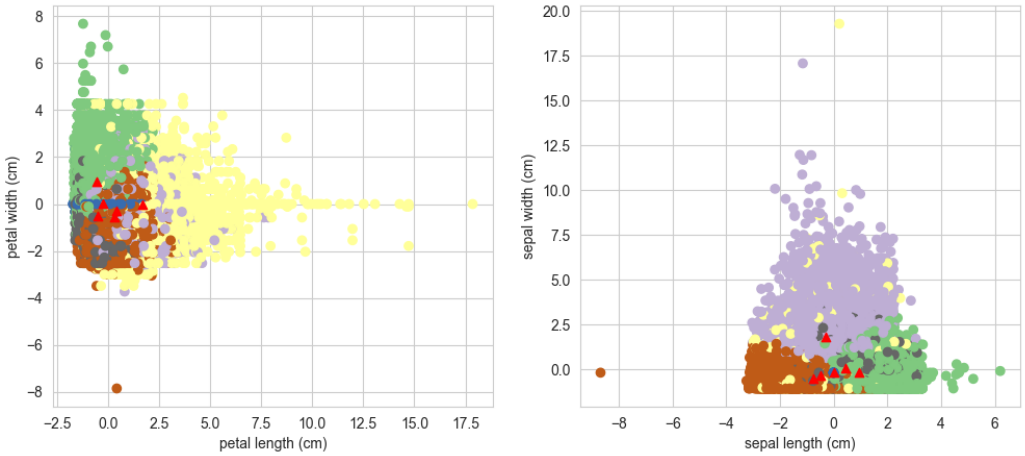
- **K-mean[38][39]**
    - Elbow method result:



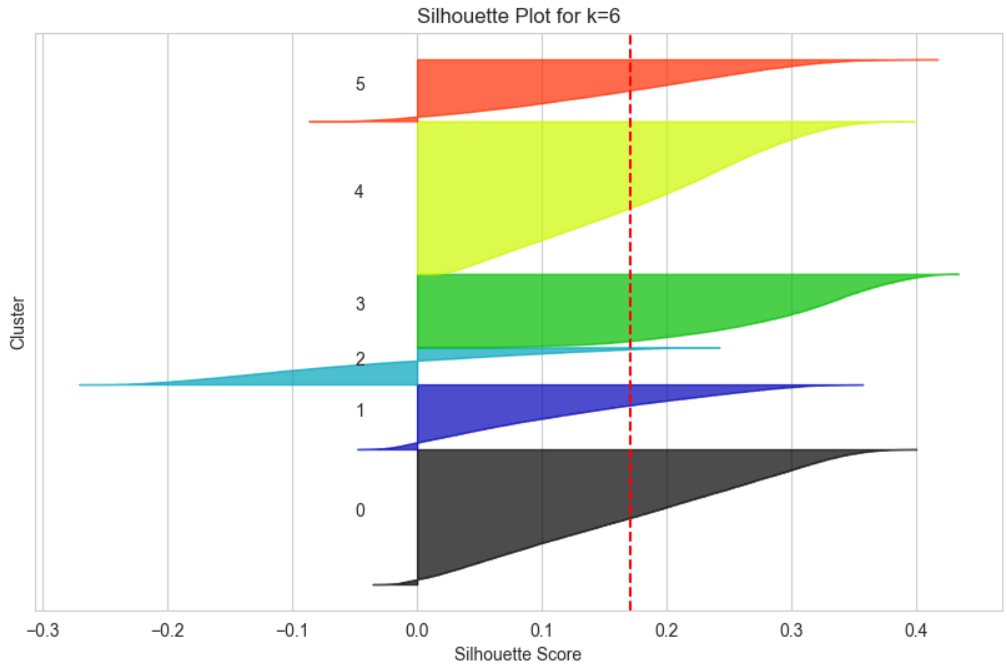
Picture 4.0 K-mean elbow method result

- The SSE decreases more when K = 6, which supports the selection to be 6 clusters for the dataset.

- Cluster Visualization
    - The scatter plots shows the dat points in 6 clusters.
    - Some clusters overlaps shows that some groups have similarity.
    - Most clusters shows a clear separation between other clusters.
    - Most clusters shows a positive silhouette score, means classification are correct.
    - The average score is around 1.8, means the clusters are reasonably well-separated.
    - Each cluster has different size showing on the plot.

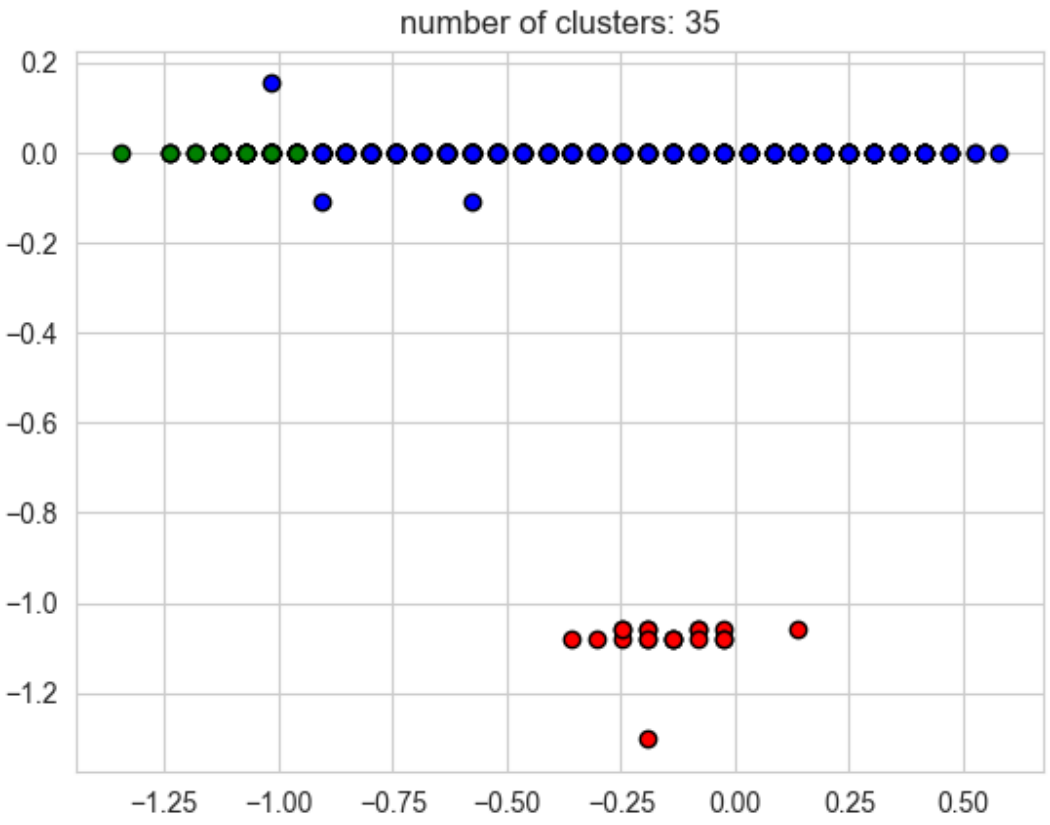


Picture 4.1 Visualization of the clusters



Picture 4.2 Visualization of the Silhouette analysis scores

- **DBSCAN[40]**
    - The plot shows that the dataset has 35 clusters, which means the radius of the algorithm might be too small.
    - Very few outliers of the dataset exist.
    - The flat structure of the plot dots shows that the clusters are more pronounced along one dimension,



Picture 4.2 Visualization of the clusters

Recommendations

- **Information and knowledge gained from the project:**
    - Data preparation
        - I learned the importance of data cleaning before training or building any models. The quality of the data plays a huge role in the quality of the model.
        - With PCA and VIF, we can quickly identify important features and eliminate not important features.
    - Model evaluation
        - Every model have different complexity when training. And every model have different performance.
        - The most time consuming model does not always perform the best.
        - The dataset quality and structure plays a huge role in the performance of different model.
        - Grid search improves model performance quite a bit.
    - Clustering analysis
        - Elbow method is essential and very effective when trying to identify the best number of clusters.
- **Best classifier**
    - Logistic regression has the best performance with the highest AUC and balanced precision, recall, and F1-score.
    - KNN and Random Forest also have promising results and can be good alternatives with different scenarios.
- **Improvements**
    - Feature engineering
        - Possibly create new features or combine existing features for better capturing patters in the dataset.
    - Hyperparameter optimization
        - Perform more extensive grid search or random search to fine-tune the hyper parameters for classifiers.

    - Handling imbalanced data
        - Try different ways of handling imbalanced data such as SMOTE or weight penalization and see which method gives the best performance.

- **Features that are associated with the target variable**
    - loan_amount
    - rate_of_interest
    - LTV
    - Credit_score
    - Upfront_charges
    - dtir1
    - loan_purpose
- **Number of clusters in the feature space**
    - With K-means clustering, the number of clusters is 6.

Python code:

- **phaseI.py**
- **phaseII.py**
- **phaseIII.py**
- **phaseIV.py**

References

[1]: https://www.kaggle.com/datasets/yasserh/loan-default-dataset/discussion/522084
[2]: https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.mean.html
[3]: https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.median.html
[4]: https://www.geeksforgeeks.org/ml-one-hot-encoding/
[5]: https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.duplicated.html
[6]: https://www.geeksforgeeks.org/random-forest-classifier-using-scikit-learn/
[7]: https://www.geeksforgeeks.org/principal-component-analysis-pca/
[8]: https://scikit-learn.org/dev/modules/generated/sklearn.decomposition.TruncatedSVD.html
[9]: https://numpy.org/doc/2.1/reference/generated/numpy.linalg.svd.html
[10]: https://www.statsmodels.org/stable/generated/
statsmodels.stats.outliers_influence.variance_inflation_factor.html
[11]: https://www.geeksforgeeks.org/detecting-multicollinearity-with-vif-python/
[12]: https://www.statsmodels.org/stable/generated/
statsmodels.stats.outliers_influence.variance_inflation_factor.html
[13]: https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.corr.html
[14]: https://scikit-learn.org/1.5/modules/generated/sklearn.ensemble.IsolationForest.html
[15]: https://www.geeksforgeeks.org/what-is-isolation-forest/
[16]: https://seaborn.pydata.org/generated/seaborn.heatmap.html
[17]: https://www.geeksforgeeks.org/seaborn-heatmap-a-comprehensive-guide/
[18]: https://www.statsmodels.org/dev/generated/
statsmodels.regression.linear_model.OLS.html
[19]: https://www.geeksforgeeks.org/ordinary-least-squares-ols-using-statsmodels/
[20]: https://www.geeksforgeeks.org/how-to-create-a-residual-plot-in-python/
[21]: https://www.geeksforgeeks.org/stepwise-regression-in-python/
[22]: https://medium.com/analytics-vidhya/post-pruning-and-pre-pruning-in-decision-
tree-561f3df73e65
[23]: https://www.geeksforgeeks.org/pruning-decision-trees/
[24]: https://www.w3schools.com/python/python_ml_grid_search.asp
[25]: https://www.geeksforgeeks.org/how-to-tune-a-decision-tree-in-hyperparameter-tuning/
[26]: https://stackoverflow.com/questions/70264157/logistic-regression-and-gridsearchcv-
using-python-sklearn
[27]: https://www.kaggle.com/code/enespolat/grid-search-with-logistic-regression
[28]: https://stackoverflow.com/questions/72067663/train-test-split-gridsearch-and-cross-
validation
[29]: https://stackoverflow.com/questions/39828535/how-to-tune-gaussiannb
[30]: https://stackoverflow.com/questions/53782169/random-forest-tuning-with-
randomizedsearchcv
[31]: https://scikit-learn.org/1.5/modules/generated/sklearn.neural_network.MLPClassifier.html
[32]: https://michael-fuchs-python.netlify.app/2021/02/03/nn-multi-layer-perceptron-classifier-
mlpclassifier/
[33]: https://www.geeksforgeeks.org/support-vector-machine-algorithm/
[34]: https://scikit-learn.org/1.5/modules/svm.html
[35]: https://blog.devgenius.io/machine-learning-algorithm-series-polynomial-kernel-svm-
understanding-the-basics-and-applications-89b4b42df137
[36]: https://www.geeksforgeeks.org/major-kernel-functions-in-support-vector-machine-svm/
[37]: https://www.geeksforgeeks.org/rbf-svm-parameters-in-scikit-learn/
[38]: https://www.geeksforgeeks.org/k-means-clustering-introduction/
[39]: https://scikit-learn.org/1.5/auto_examples/cluster/plot_kmeans_silhouette_analysis.html
[40]: https://www.geeksforgeeks.org/dbscan-clustering-in-ml-density-based-clustering/