Indian Institute of Technology, Dharwad

CS 209/214 - Artificial Intelligence
Term Project Report

Course Instructor:
Dr. Dileep A. D.

Mentor:
Ms. Neha Rudrappa Pudakalakatti

Team 21

Raghuveer Verma	CS23BT041
Annavarapu Vijwal	MC23BT001
Shaikh Aariz	MC23BT030
Abhinandan Jain	CS23BT012

Abstract

In this project, we explore and compare various Machine Learning models for three tasks: Classification, Regression, and Clustering. The data set used contains features that describe the financial situation and personal attributes of individuals.

In classification, the goal is to predict whether a loan is approved or not. For the regression task, we use models to estimate an individual's risk score. Clustering involves identifying clusters in the data, where we use techniques such as the Elbow Method and Silhouette Analysis to get the optimal number of clusters.

Contents

Chapter 1

The Dataset

We use a synthetic dataset comprising of 20,000 records of personal and financial data. The original dataset can be found here.

Features

The dataset contains the following thirty-six columns:

1. Demographic and Employment Information
    - Age
    - Marital Status
    - Number of Dependents
    - Education Level
    - Employment Status
    - Experience
    - Job Tenure
2. Financial and Credit Information
    - Annual Income
    - Monthly Income
    - Total Assets
    - Total Liabilities
    - Net Worth
    - Home Ownership Status
    - Savings Account Balance
    - Checking Account Balance
    - Debt-to-Income Ratio
    - Total Debt-to-Income Ratio
    - Monthly Debt Payments
    - Utility Bills Payment History
3. Credit History and Loan Information
    - Credit Score
    - Credit Card Utilization Rate
    - Number of Open Credit Lines
    - Number of Credit Inquiries
    - Bankruptcy History
    - Length of Credit History
    - Previous Loan Defaults
    - Payment History
4. Loan Attributes
    - Loan Amount
    - Loan Duration
    - Loan Purpose
    - Base Interest Rate
    - Interest Rate
    - Monthly Loan Payment
    - Application Date
5. Target Variables
    - Loan Approved
    - Risk Score

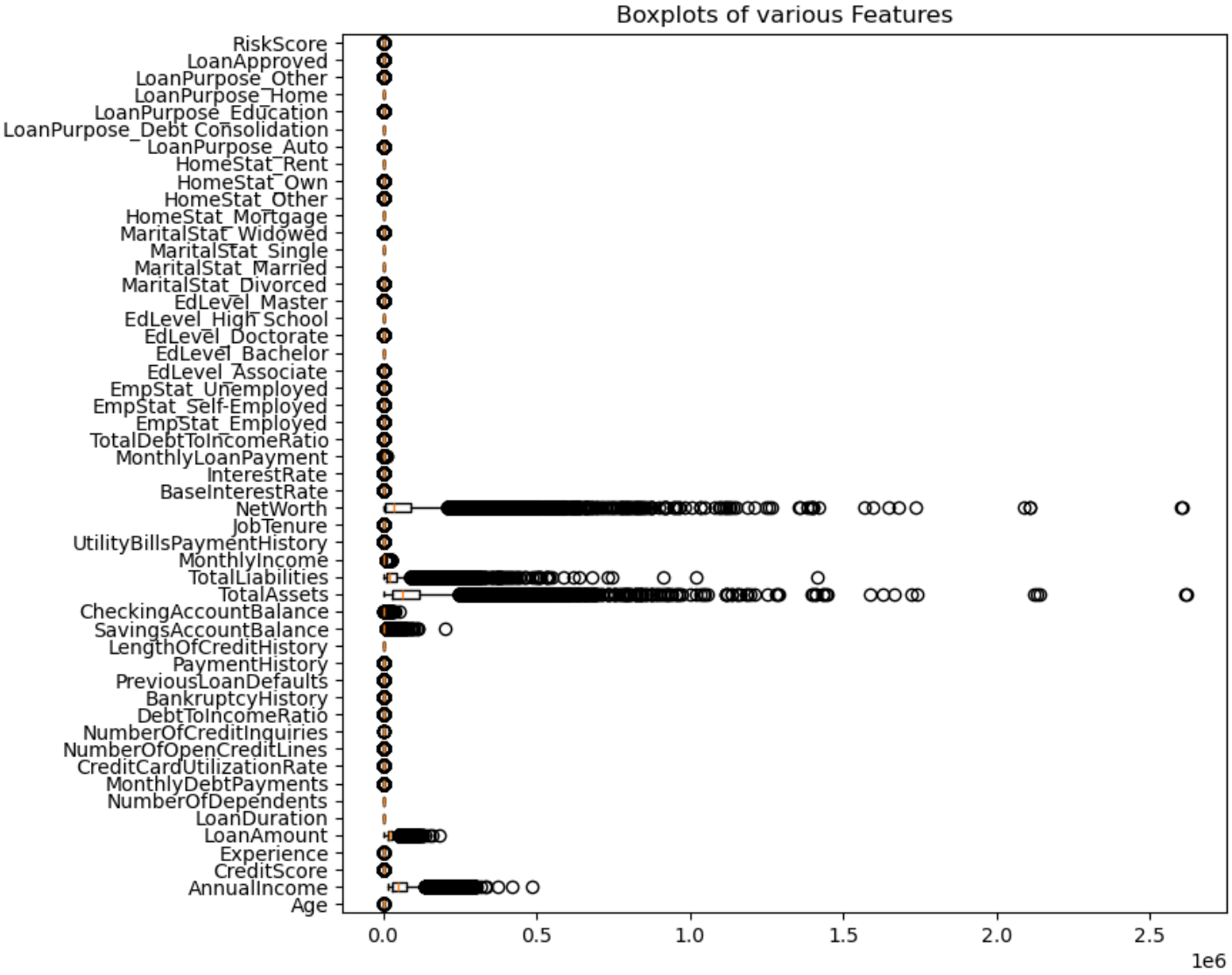
Analysis and Preprocessing

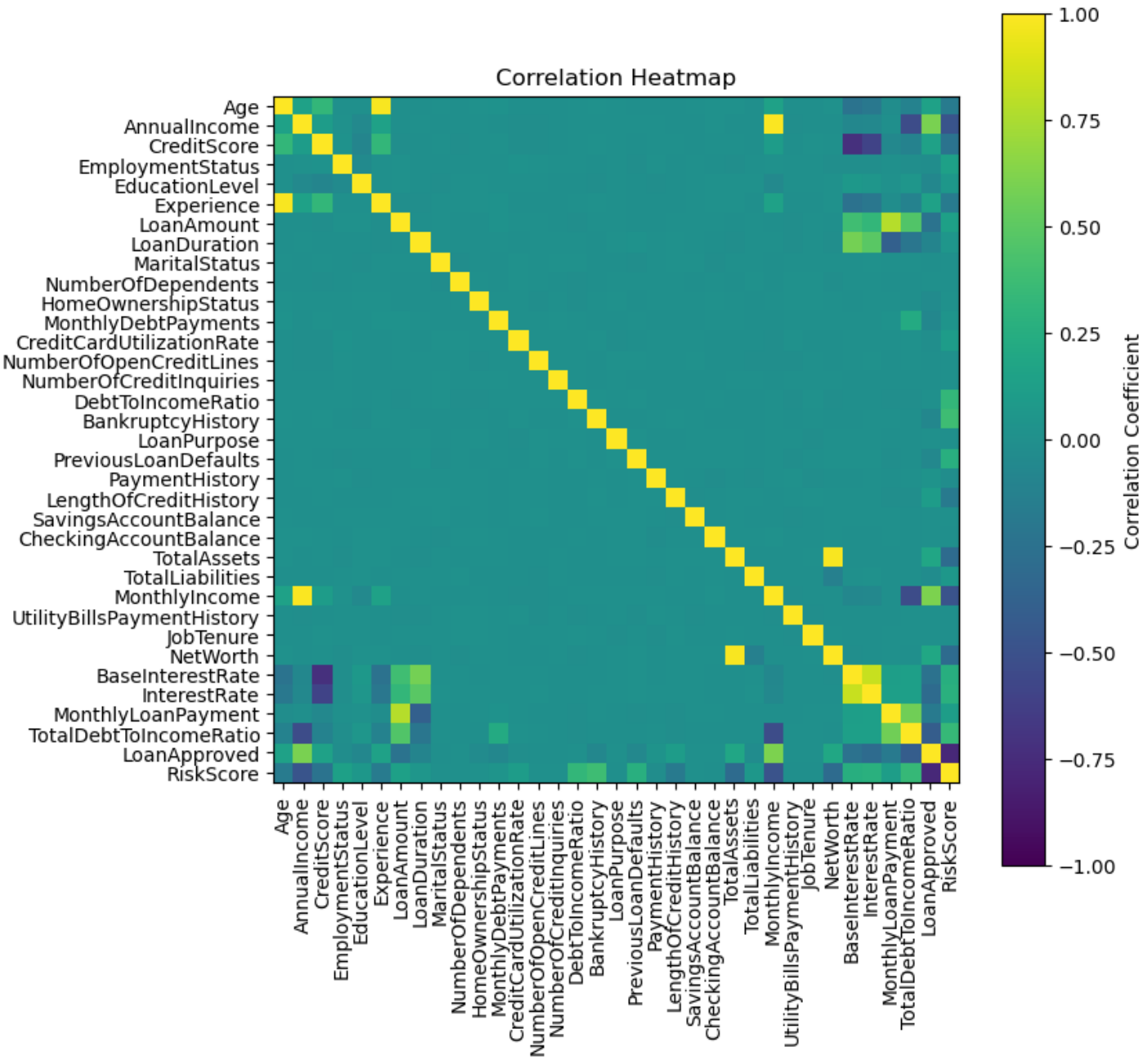
Duplicate and Missing Values: No duplicate rows or rows with missing values were found.

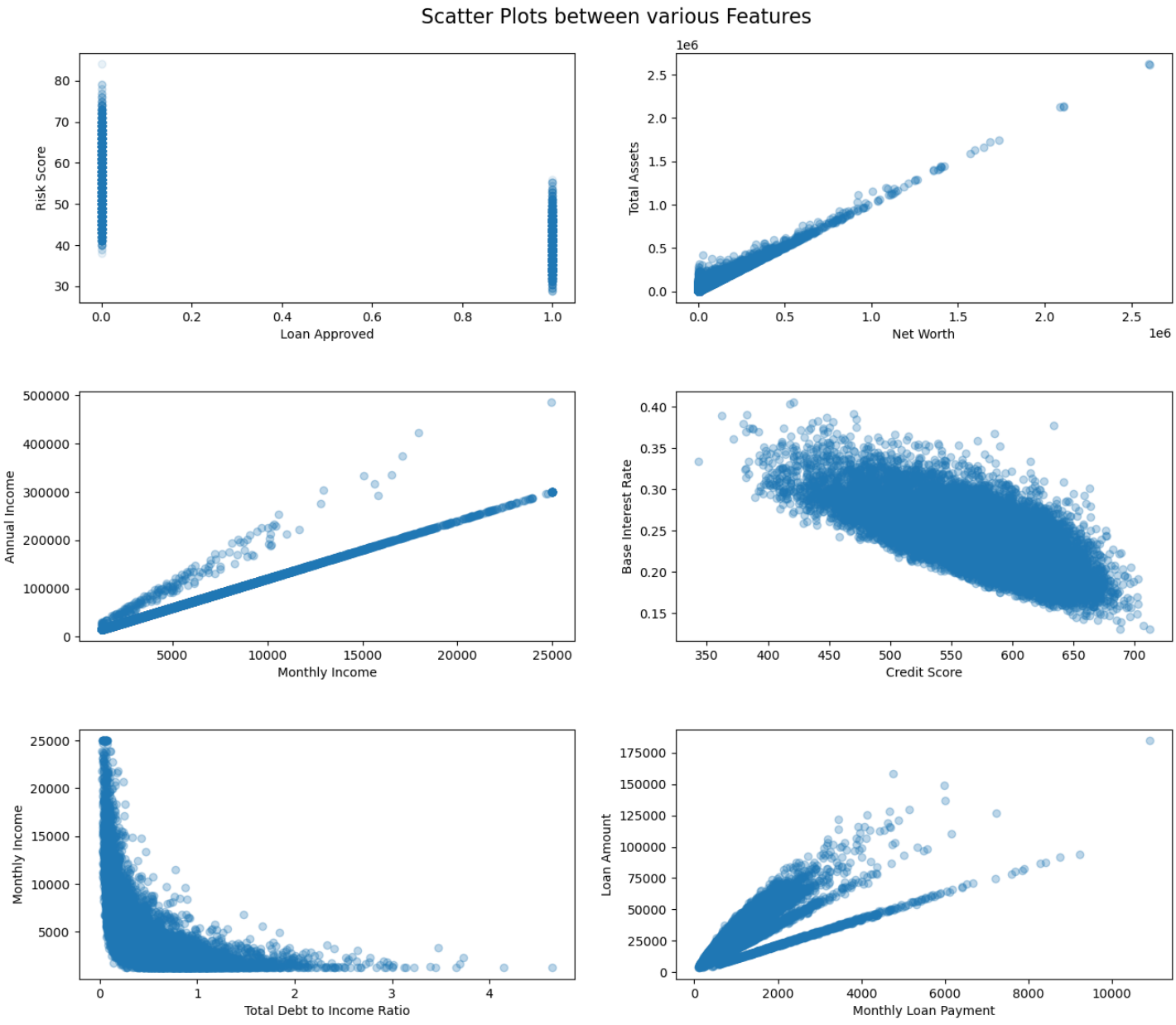
Columns with multiple classes: The following columns described multiple classes as strings which were later converted to integer labels (given on the left) and one-hot encodings in separate datasets:

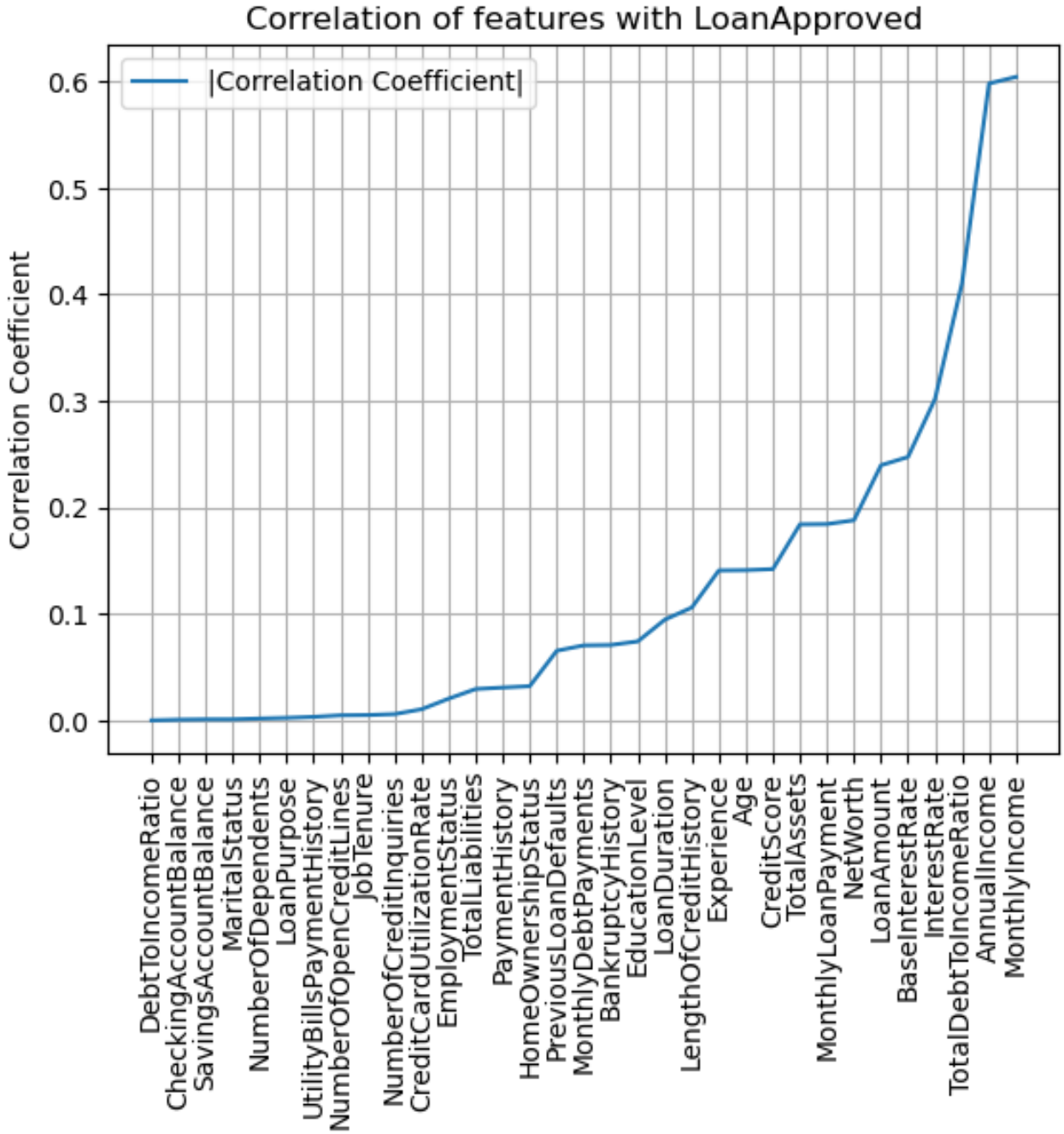
- **Employment Status**
  - 0: Employed
  - 1: Self-Employed
  - 2: Unemployed
- **Education Level**
  - 0: Master
  - 1: Associate
  - 2: Bachelor
  - 3: High School
  - 4: Doctorate
- **Marital Status**
  - 0: Married
  - 1: Single
  - 2: Divorced
  - 3: Widowed
- **Home Ownership Status**
  - 0: Own
  - 1: Mortgage
  - 2: Rent
  - 3: Other
- **Loan Purpose**
  - 0: Home
  - 1: Debt Consolidation
  - 2: Education
  - 3: Other
  - 4: Auto

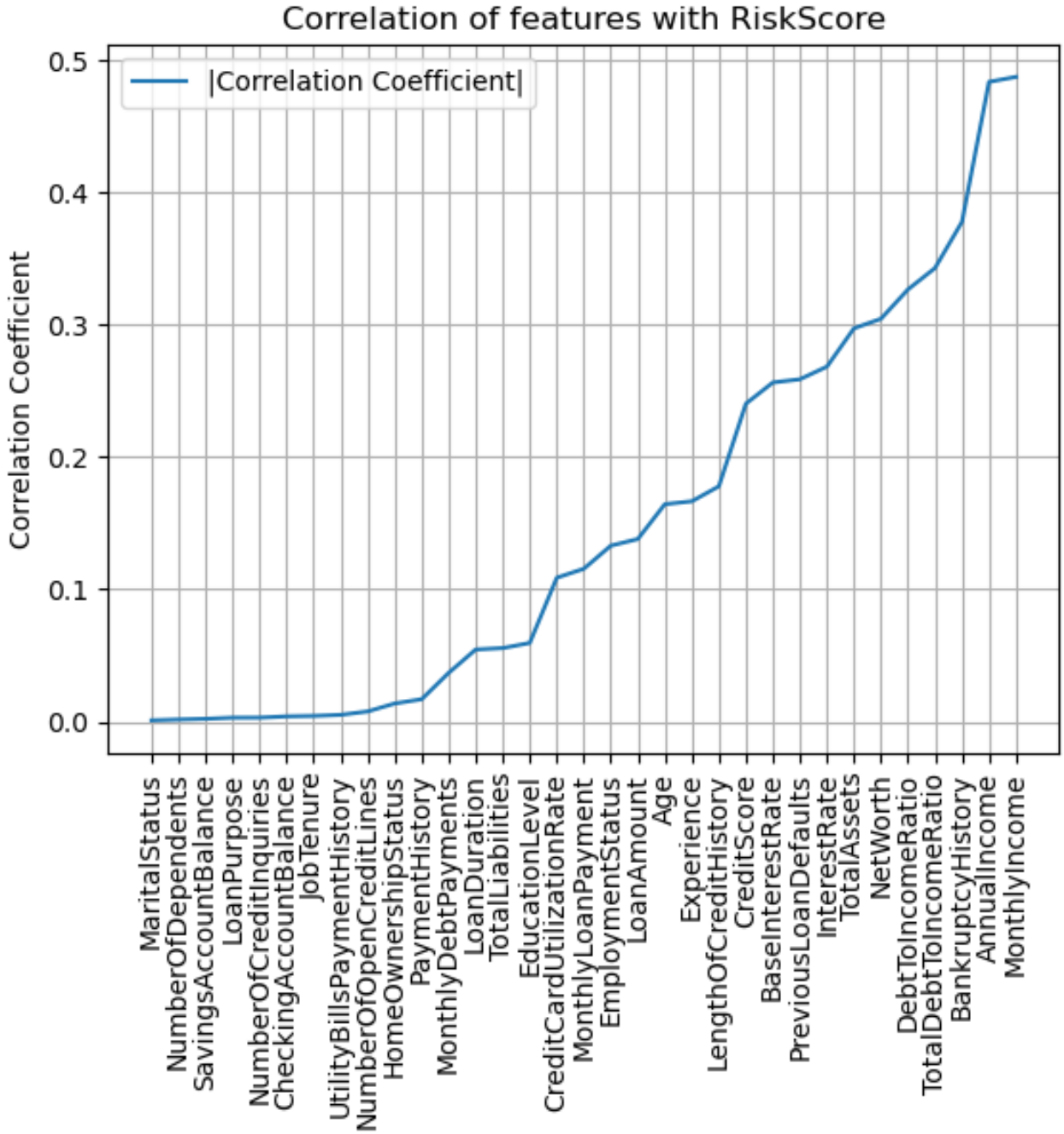
Visualizations




Correlation Heatmap
Age
AnnualIncome
CreditScore
EmploymentStatus
EducationLevel
Experience
LoanAmount
LoanDuration
MaritalStatus
NumberOfDependents
HomeOwnershipStatus
MonthlyDebtPayments
CreditCardUtilizationRate
NumberOfOpenCreditLines
NumberOfCreditInquiries
DebtToIncomeRatio
BankruptcyHistory
LoanPurpose
PreviousLoanDefaults
PaymentHistory
LengthOfCreditHistory
SavingsAccountBalance
CheckingAccountBalance
TotalAssets
TotalLiabilities
MonthlyIncome
UtilityBillsPaymentHistory
JobTenure
NetWorth
BaseInterestRate
InterestRate
MonthlyLoanPayment
TotalDebtToIncomeRatio
LoanApproved
RiskScore
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
Correlation Coefficient


Scatter Plots between various Features
Risk Score
Loan Approved
Total Assets
Net Worth
1e6
Annual Income
Monthly Income
Base Interest Rate
Credit Score
Monthly Income
Total Debt to Income Ratio
Loan Amount
Monthly Loan Payment


Correlation of features with LoanApproved
|Correlation Coefficient|
Correlation Coefficient
0.0
0.1
0.2
0.3
0.4
0.5
0.6
DebtToIncomeRatio
CheckingAccountBalance
SavingsAccountBalance
MaritalStatus
NumberOfDependents
LoanPurpose
UtilityBillsPaymentHistory
NumberOfOpenCreditLines
JobTenure
NumberOfCreditInquiries
CreditCardUtilizationRate
EmploymentStatus
TotalLiabilities
PaymentHistory
HomeOwnershipStatus
PreviousLoanDefaults
MonthlyDebtPayments
BankruptcyHistory
EducationLevel
LoanDuration
LengthOfCreditHistory
Experience
Age
CreditScore
TotalAssets
MonthlyLoanPayment
NetWorth
LoanAmount
BaseInterestRate
InterestRate
TotalDebtToIncomeRatio
AnnualIncome
MonthlyIncome


Correlation of features with RiskScore
|Correlation Coefficient|
Correlation Coefficient
0.0
0.1
0.2
0.3
0.4
0.5
MaritalStatus
NumberOfDependents
SavingsAccountBalance
LoanPurpose
NumberOfCreditInquiries
CheckingAccountBalance
JobTenure
UtilityBillsPaymentHistory
NumberOfOpenCreditLines
HomeOwnershipStatus
PaymentHistory
MonthlyDebtPayments
LoanDuration
TotalLiabilities
EducationLevel
CreditCardUtilizationRate
MonthlyLoanPayment
EmploymentStatus
LoanAmount
Age
Experience
LengthOfCreditHistory
CreditScore
BaseInterestRate
PreviousLoanDefaults
InterestRate
TotalAssets
NetWorth
DebtToIncomeRatio
TotalDebtToIncomeRatio
BankruptcyHistory
AnnualIncome
MonthlyIncome

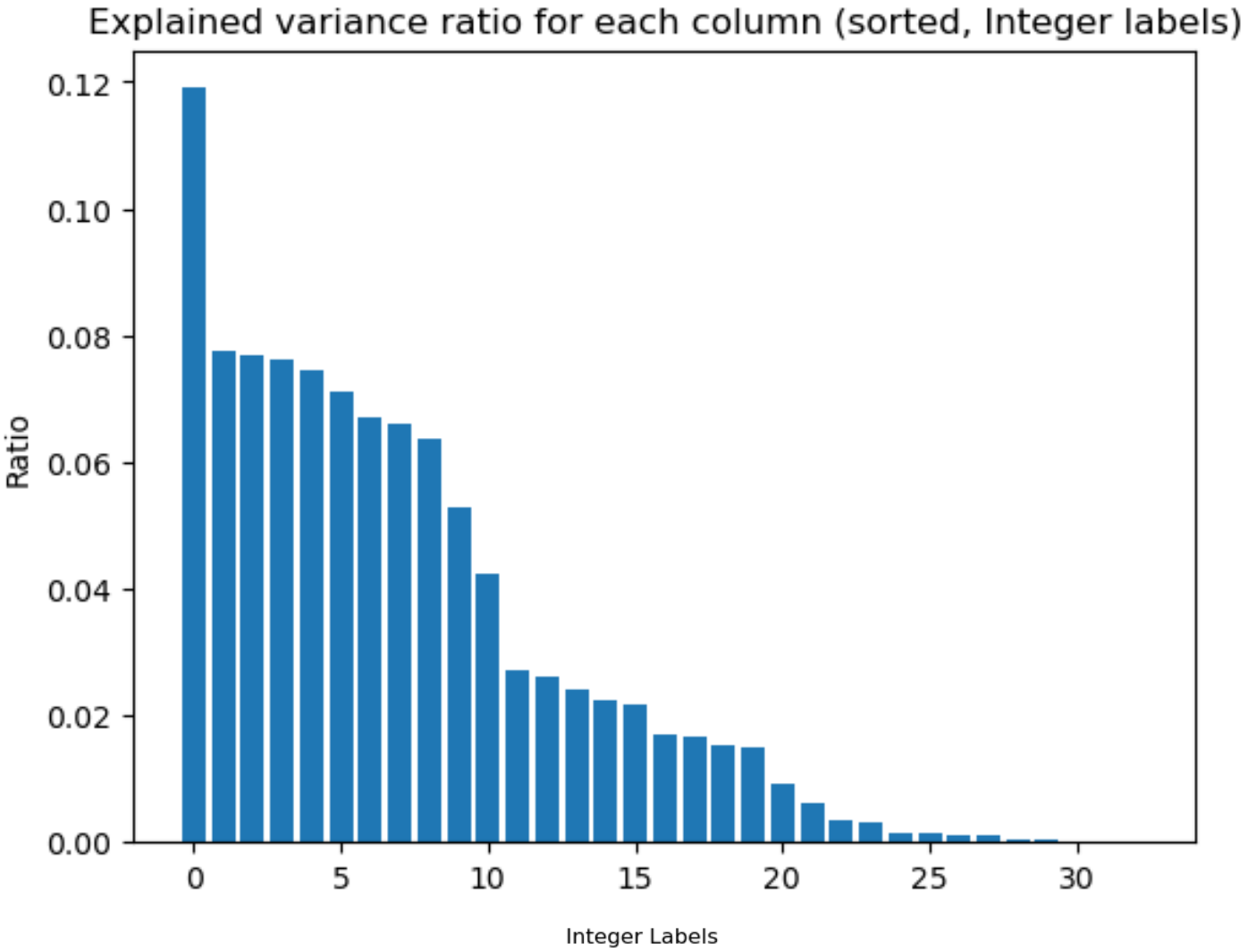
Train-Validation-Test Split: We used a 60-20-20 split which resulted in a training set with 12000 examples and validation and test sets with 4000 examples each.

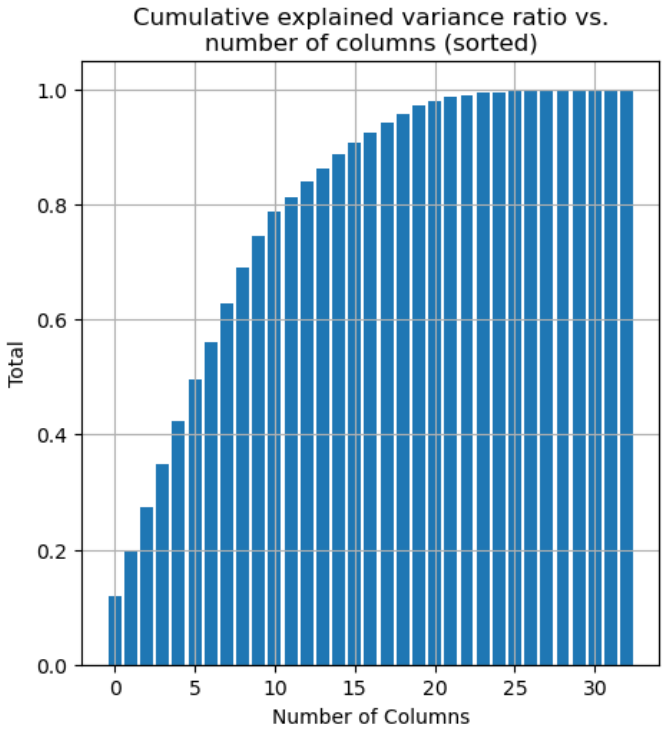
Outlier Processing: The boxplot shows presence of many outliers in the data. However, these cannot be simply removed or altered since they are valid data points. Hence we proceed with the data as it is.

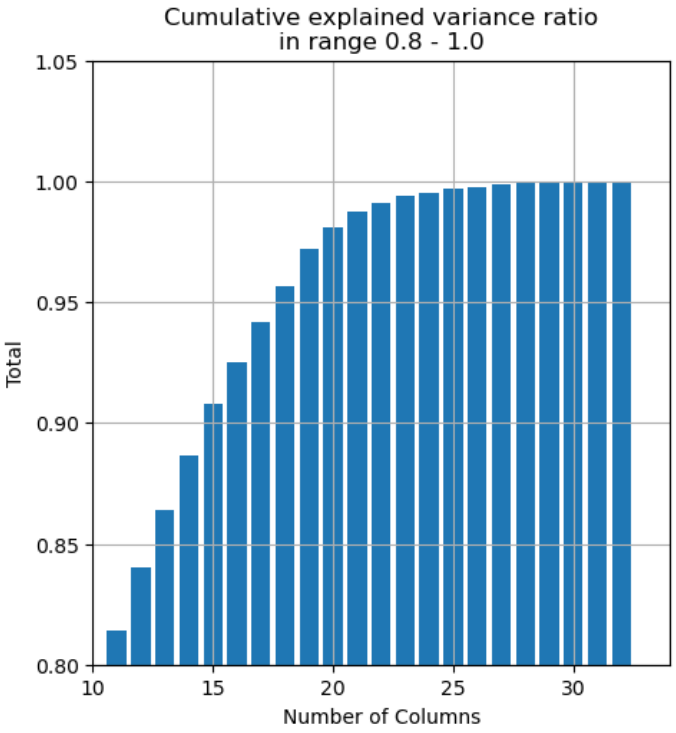
Data Scaling: Data has been processed through min-max scaling, Z-score scaling, and Robust Scaling (to deal with outliers) separately.

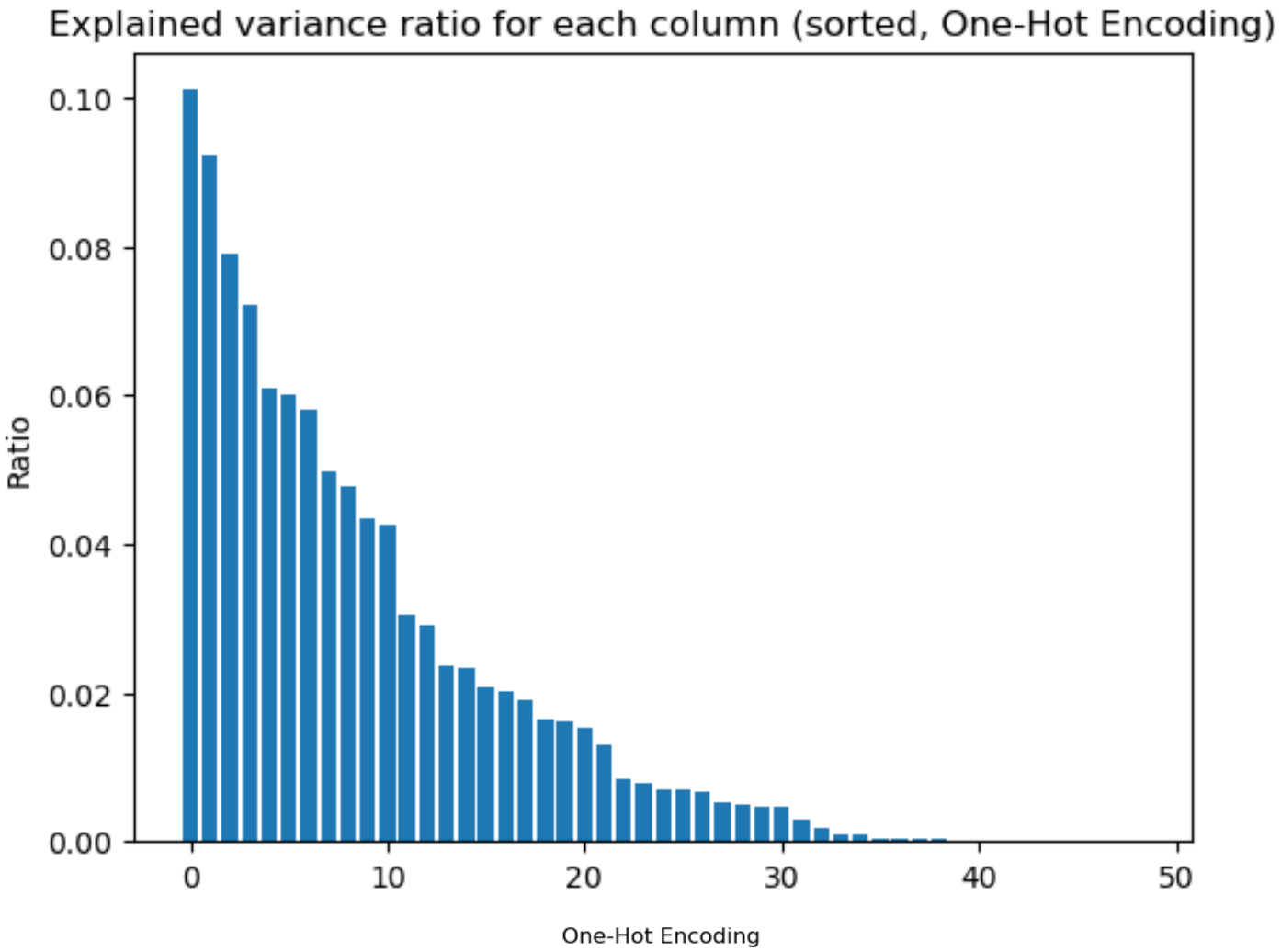
Dimensional Reduction: The column `ApplicationDate` was dropped since it does not affect `LoanApproved` or `RiskScore`. Principal Component Analysis has been used to further reduce the dimensions of the data.

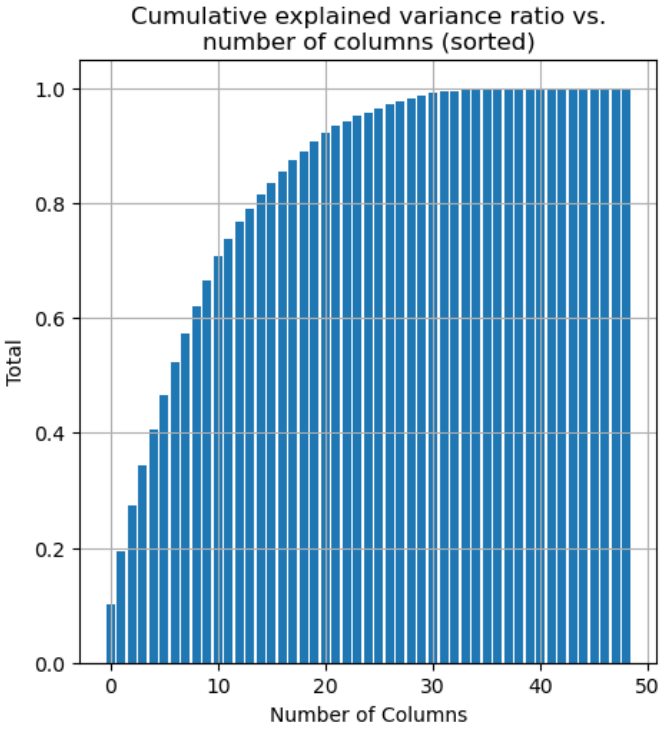
For data with integer labels, dimensions have been reduced from 33 to 10, 15, 20, and 25. For data with one-hot encoding, dimensions have been reduced from 49 to 15, 20, 25, 30, and 35.

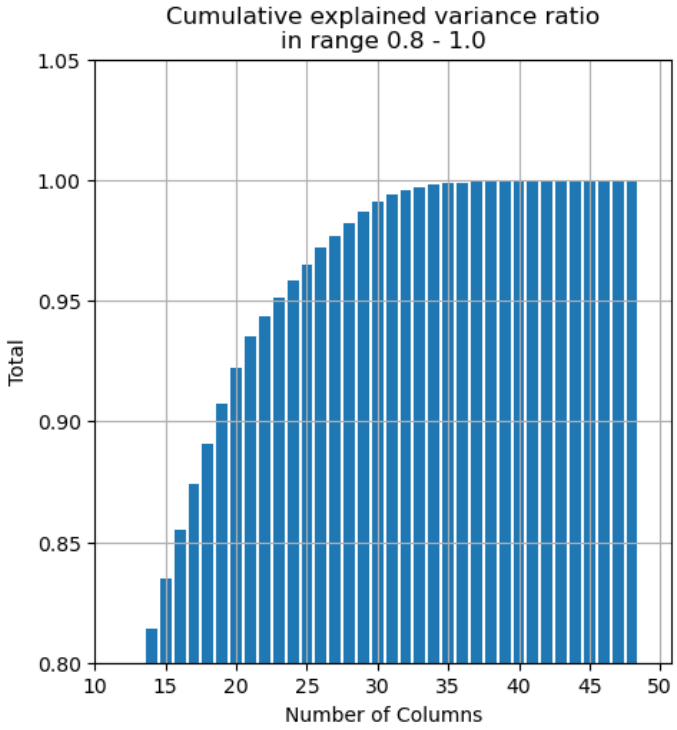

Explained variance ratio for each column (sorted, Integer labels)
Ratio
0.12
0.10
0.08
0.06
0.04
0.02
0.00
0
5
10
15
20
25
30
Integer Labels


Cumulative explained variance ratio vs.
number of columns (sorted)
Total
1.0
0.8
0.6
0.4
0.2
0.0
0
5
10
15
20
25
30
Number of Columns


Cumulative explained variance ratio
in range 0.8 - 1.0
Total
1.05
1.00
0.95
0.90
0.85
0.80
10
15
20
25
30
Number of Columns


Explained variance ratio for each column (sorted, One-Hot Encoding)
Ratio
0.10
0.08
0.06
0.04
0.02
0.00
0
10
20
30
40
50
One-Hot Encoding


Cumulative explained variance ratio vs.
number of columns (sorted)
Total
1.0
0.8
0.6
0.4
0.2
0.0
0
10
20
30
40
50
Number of Columns


Cumulative explained variance ratio
in range 0.8 - 1.0
Total
1.05
1.00
0.95
0.90
0.85
0.80
10
15
20
25
30
35
40
45
50
Number of Columns

Chapter 2

Classification

The target variable `LoanApproved` is the second last column in the preprocessed datasets. We use the following models for classifying whether a loan request is approved or not:

- k-Nearest Neighbors
- Logistic Regression
- Support Vector Machine
- Artificial Neural Networks

k-Nearest Neighbors

Using sklearn.neighbors' KNeighborsClassifier with $k = 3, 5, 7, 11$ and Mahalanobis Variant, we got best accuracy of 82.50% and precision of 81.84% with $k = 11$, best recall of 44.35% and best F1-score of 52.65%.

The best performing dataset was with:

- One-Hot Encoding
- Z-score normalization

The test performance metrics for $k = 5$ are:

- Accuracy: 81.93%
- Precision: 70.40%
- Recall: 42.05%
- F1-Score: 52.65%

The test performance metrics for $k = 11$ are:

- Accuracy: 82.50%
- Precision: 81.84%
- Recall: 34.41%
- F1-Score: 48.45%

Logistic Regression

We used the sklearn LogisticRegression model with max_iter=1000 across different preprocessed datasets which resulted in the best accuracy of 96.70% and the best F1-score of 93.09% on the Z-score data with one-hot encoding.

Best precision of 93.98% was obtained on the min-max scaled data with one-hot encoding and best recall of 92.99% was obtained on RobustScaled data with one-hot encoding.

Overall, the best performing dataset for logistic regression had:

- One-Hot Encoding
- Z-score normalization

The test performance metrics are:

- Accuracy: 96.70%
- Precision: 93.19%
- Recall: 92.99%
- F1-Score: 93.09%

Support Vector Machine

RBF (Gaussian Kernel)
The sklearn.svm's SVC was used with kernel='rbf', C=0.1, 1, 10 and gamma='scale'. Best results were obtained with C=1.

The best performing dataset was with:

- One-Hot Encoding
- Z-score normalization

The test performance metrics are:

- Accuracy: 96.20%
- Precision: 93.23%
- Recall: 90.69%
- F1-Score: 91.94%

Polynomial Kernel
The sklearn.svm's SVC was used with kernel='poly', C=0.1, 1, 10 and gamma='scale'. Best results were obtained with C=1.

The best performing dataset was with:

- One-Hot Encoding
- Z-score normalization

The test performance metrics are:

- Accuracy: 94.20%
- Precision: 94.80%
- Recall: 80.13%
- F1-Score: 86.85%

Artificial Neural Network

We used torch.nn's neural network along with loss criterion of BCELoss and optimzer Adam. The number of hidden layers were varied from 1 to 3 and the number of neurons in hidden layers were varied as 8, 16, 32, 64, and 128.

Single Layer Architecture
Best Performance was achieved with 128 hidden neurons.

The best performing dataset was with:

- One-Hot Encoding
- Min-Max normalization

The test performance metrics are:

- Accuracy: 96.57%
- Precision: 92.44%
- Recall: 93.31%
- F1-Score: 92.87%

Double Layer Architecture
Best Performance was achieved with 8 and 64 hidden neurons.

The best performing dataset was with:

- One-Hot Encoding
- Z-score normalization

The test performance metrics are:

- Accuracy: 96.75%
- Precision: 92.93%
- Recall: 93.51%
- F1-Score: 93.22%

Triple Layer Architecture
Best Performance was achieved with 8, 32 and 64 hidden neurons.

The best performing dataset was with:

- One-Hot Encoding
- Z-score normalization

The test performance metrics are:

- Accuracy: 96.77%
- Precision: 93.21%
- Recall: 93.31%
- F1-Score: 93.26%

Feature	Weight
Employment Status	45.8074
Interest Rate	35.0623
Number of Open Credit Lines	34.3788
Total Debt-To-Income Ratio	31.8159
Number of Dependents	31.4804
Number of Credit Inquiries	31.2968
Job Tenure	30.9944
Utility Bills Payment History	29.5157
Debt-To-Income Ratio	29.2051
Credit Card Utilization Rate	28.5385
Savings Account Balance	28.3232
Checking Account Balance	26.4220
Marital Status	23.6274
Monthly Income	23.1984
Payment History	22.3997
Length of Credit History	22.0455
Experience	21.6128
Loan Duration	20.9622
Age	20.3916
Home Ownership Status	19.7985
Total Liabilities	19.3334
Loan Amount	18.8017
Monthly Debt Payments	18.2229
Previous Loan Defaults	17.5500
Net Worth	17.5360
Annual Income	17.4526
Loan Purpose	16.5604
Bankruptcy History	15.3973
Education Level	13.8001
Credit Score	12.0731
Total Assets	11.9530
Base Interest Rate	9.7190
Monthly Loan Payment	7.9780

Table 2.1: Feature weights calculated at the first layer of double layer architecture

Overall Performance

One-Hot encoded and Z-score normalized data gave the best performance across almost all the models. Logistic Regression and Triple Layer Neural Network models gave the best performance.

Model	**Accuracy**	**Precision**	**Recall**	**F1-Score**
Logistic Regression	96.70%	93.19%	92.99%	93.09%
Artificial Neural Network	96.77%	93.21%	93.31%	93.26%

Table 2.2: Performance Metrics of Classification Models

Chapter 3

Regression

The target variable `RiskScore` is the last column in the preprocessed datasets. We use the following models for predicting the risk score:

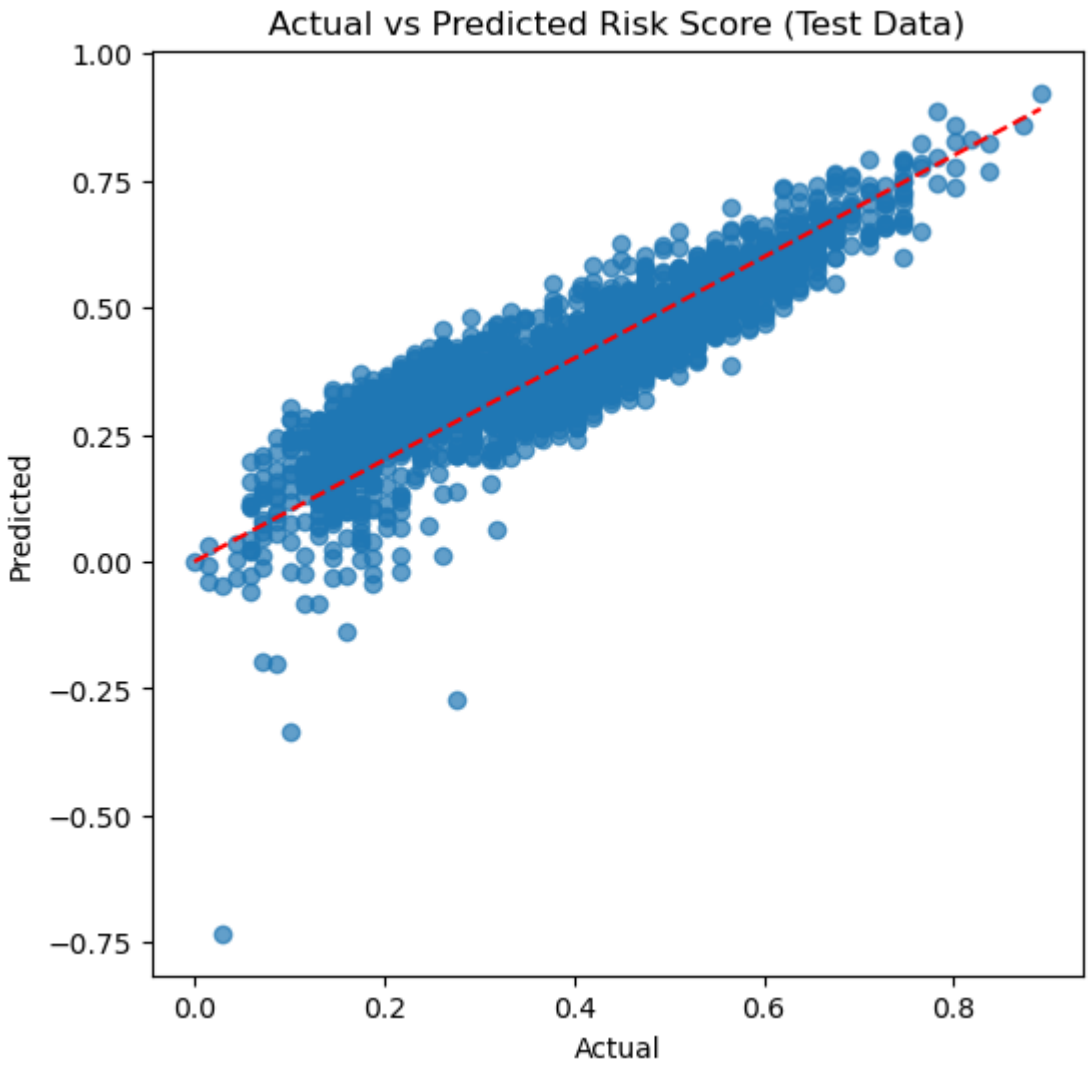
- Linear and Polynomial Regression
- Random Forest Regression
- XGBoost
- Artificial Neural Network

Linear Regression

The best performing dataset was with:

- One-Hot Encoding
- Min-Max normalization

The test RMSE is 0.066.

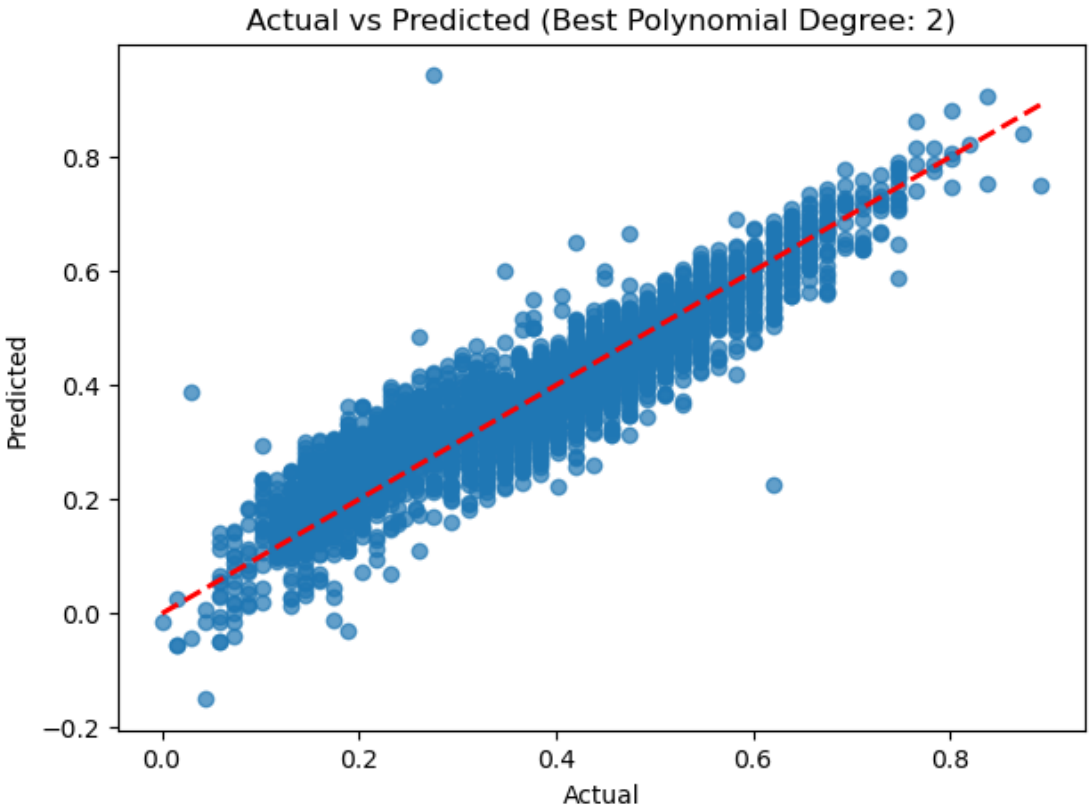


Polynomial Regression

The best performing dataset was with:

- One-Hot Encoding
- Min-Max normalization

The test RMSE is 0.066.



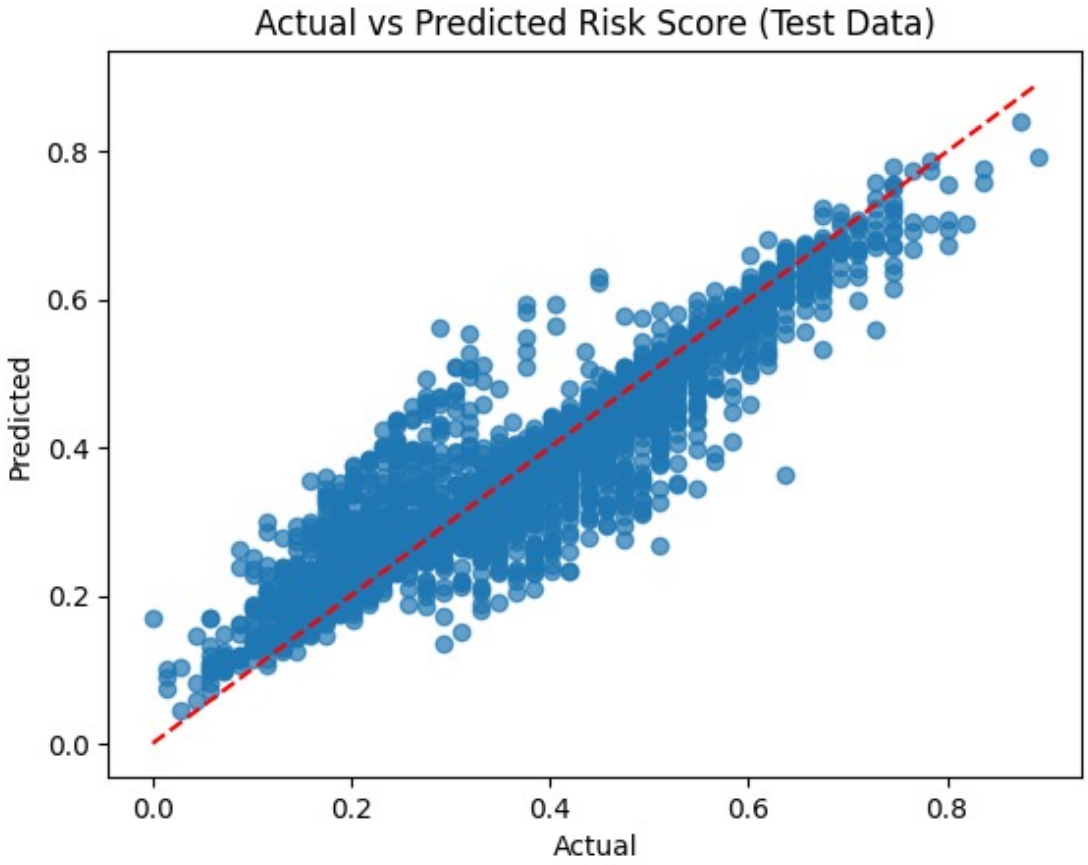
Random Forest Regression

For n_estimators=100, 200, 300, 500, all values gave similar MSE of 0.0042 and R^2 of 0.7950 on validation data.

The best performing dataset was with:

- One-Hot Encoding
- Min-Max normalization

The test RMSE is 0.051.



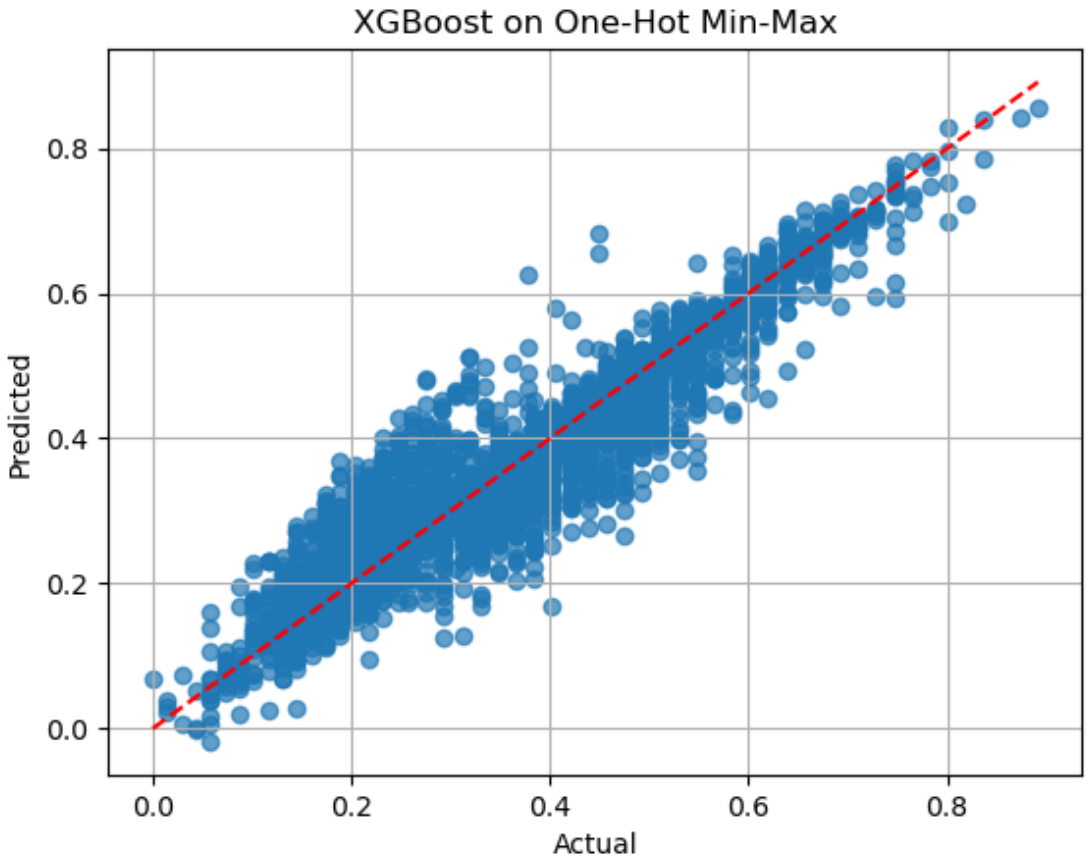
XGBoost

Using xgboost's XGBRegressor with objective='reg:squarederror', max_depth=6, and min_child_weight=10, we got the following results.

The best performing dataset was with:

- One-Hot Encoding
- Min-Max normalization

The test RMSE is 0.046.



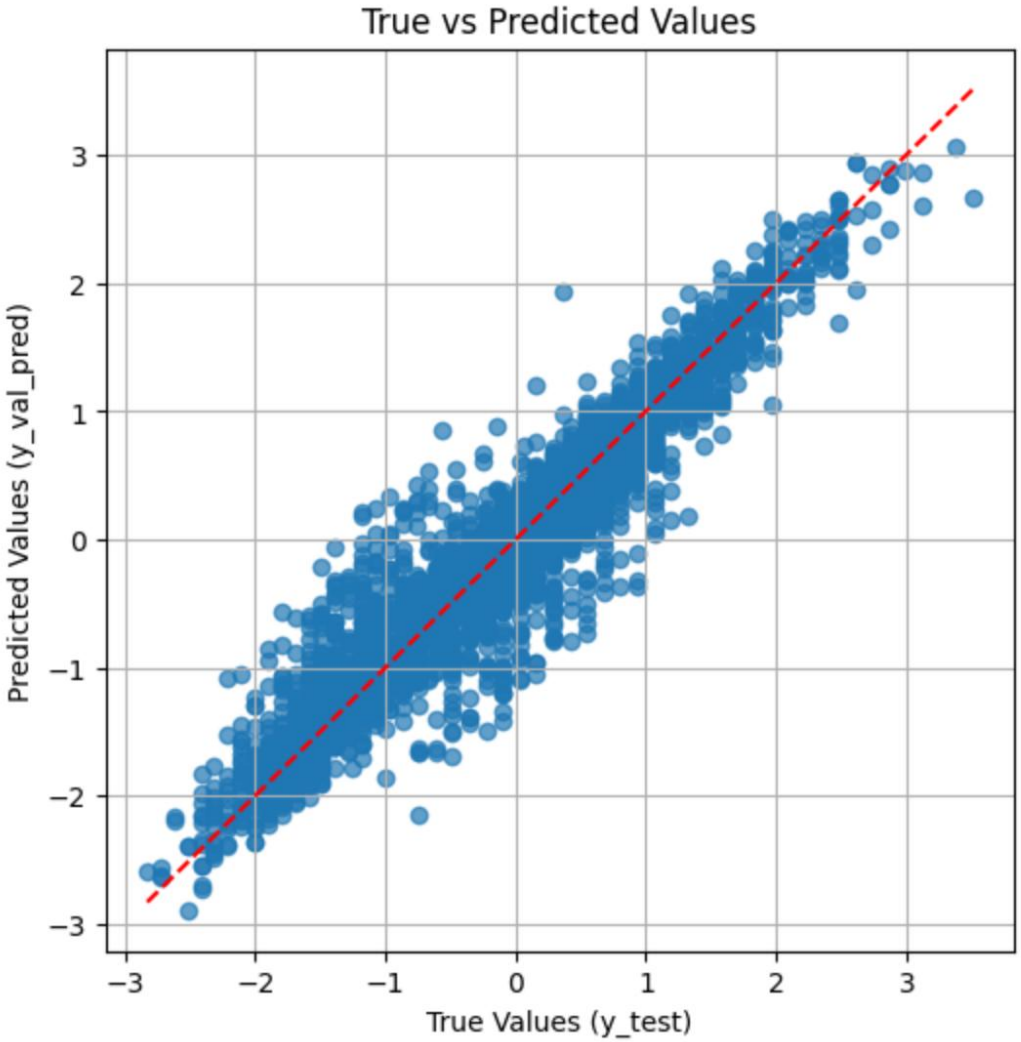
Artificial Neural Network

We used torch.nn's neural network and varied the number of hidden layers from 1 to 3 and the number of neurons in hidden layers were varied as 8, 16, 32, 64, and 128.

The best performing dataset was with:

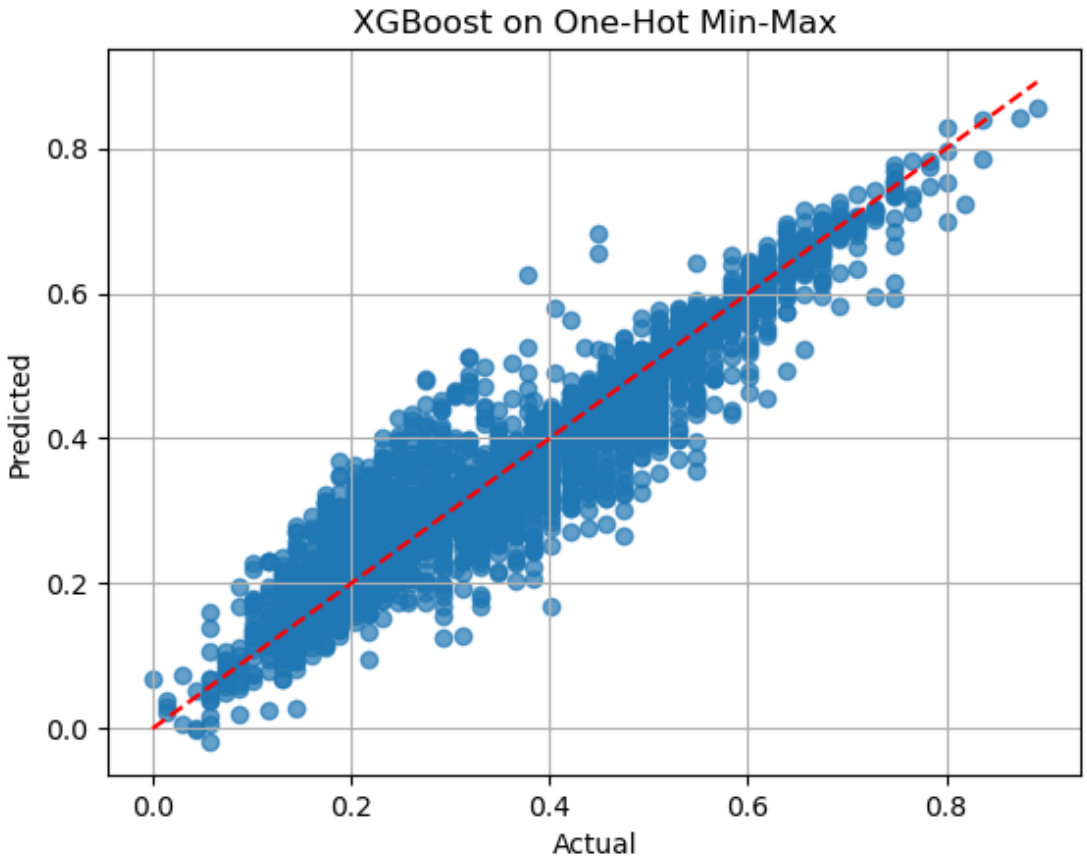
- One-Hot Encoding
- Z-score normalization

The test RMSE is 0.325 for single layer, 0.304 for double layer, and 0.313 for triple layer architecture.



Overall Performance

For the regression task, one-hot encoded and min-max normalized data showed better results. XGBoost model was the most suitable model with RMSE of 0.046.



Chapter 4

Clustering

For clustering analysis, the following models were used:

- k-Medoid
- DBSCAN
- Agglomerative Hierarchical Clustering

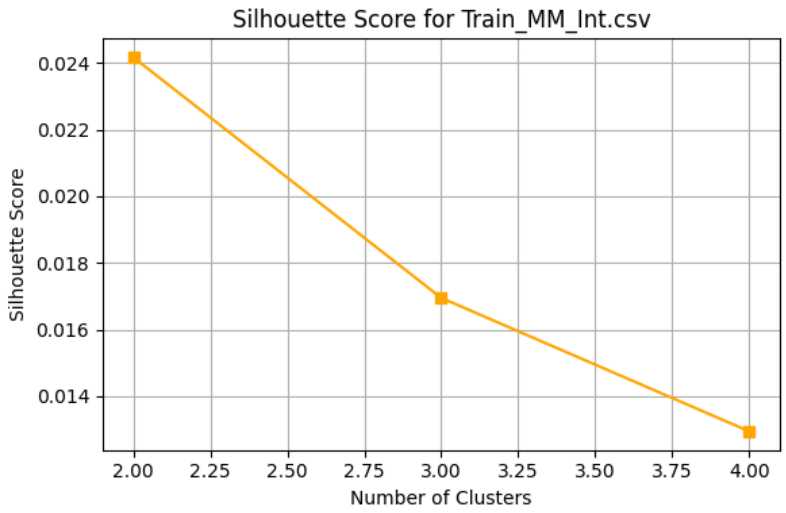
For visualizations, the two most import principal components were chosen.

k-Medoid

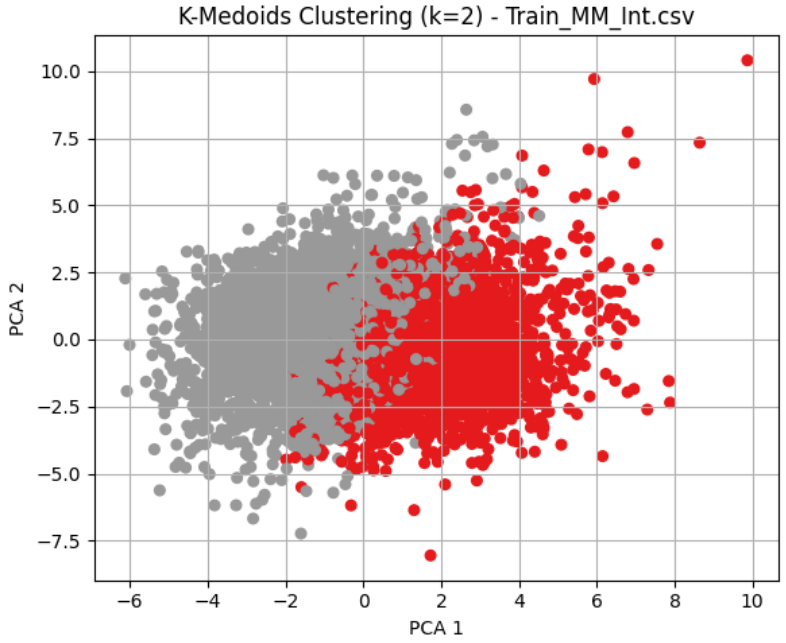
We used the elbow method and silhouette scores to determine the optimal value of k. The graph was found to be almost linear after $k = 2$.

Table 4.1: Performance Metrics for Different Datasets using K-Medoids Clustering ($k = 2$)

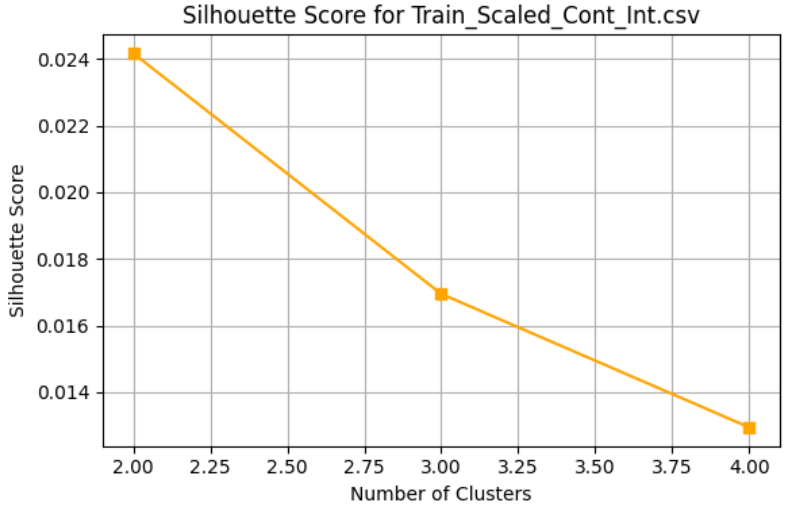
S.No	Dataset	Silhouette Score	Inertia
0	Train_Scaled_Cont_OH.csv	0.020905	91323.673402
1	Train_MM_OH.csv	0.020905	91323.673402
2	train_OH_MM_PCA15.csv	0.071245	48832.189519
3	train_OH_MM_PCA20.csv	0.032095	57938.435981
4	train_OH_MM_PCA25.csv	0.032465	64564.495058
5	train_OH_MM_PCA30.csv	0.022067	71582.557634
6	train_OH_MM_PCA35.csv	0.020132	77578.418111
7	Train_Scaled_Cont_Int.csv	0.024165	70411.072568
8	Train_MM_Int.csv	0.024165	70411.072568
9	train_Int_MM_PCA10.csv	0.068794	36153.897856
10	train_Int_MM_PCA15.csv	0.045488	46035.836064
11	train_Int_MM_PCA20.csv	0.032297	55019.387744
12	train_Int_MM_PCA25.csv	0.026684	62399.740681



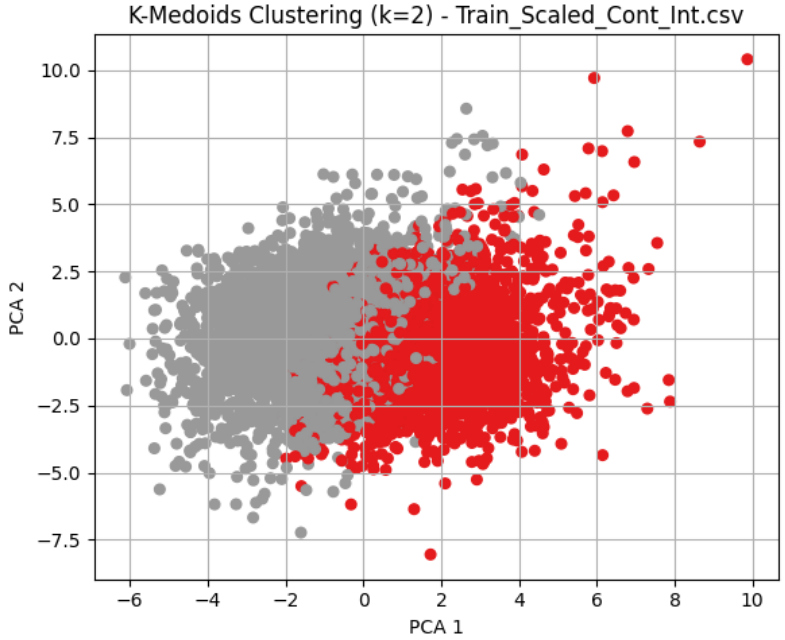
Silhouette Score Graph for Integer labeled, min-max normalized dataset
Inertia = 70411.072



Clusters for Integer labeled, min-max normalized dataset



Silhouette Score Graph for Integer labeled, Z-score normalized dataset
Inertia = 91323.673

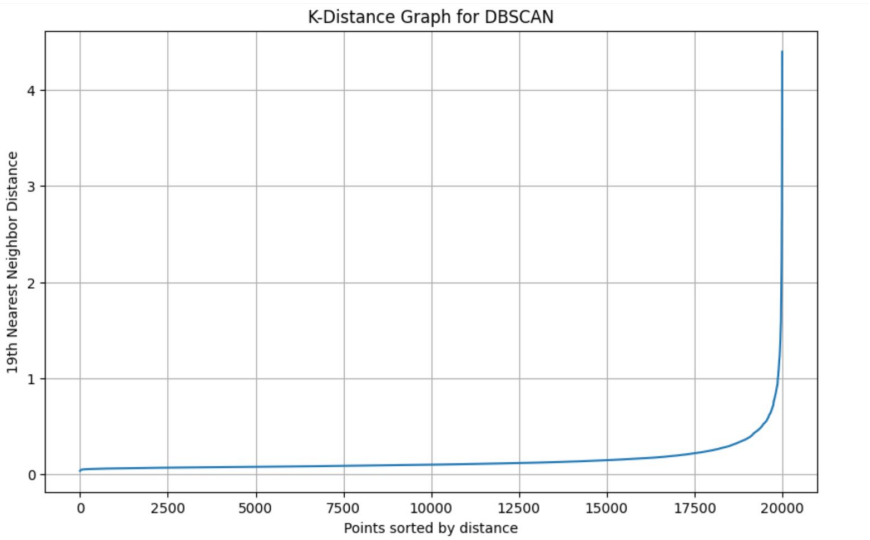


Clusters for Integer labeled, Z-score normalized dataset

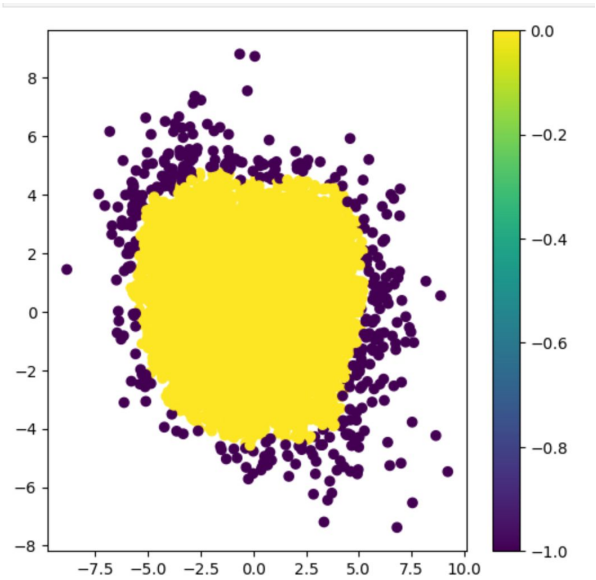
DBSCAN

The performance of DBSCAN heavily depends on the choice of the eps (neighborhood radius) and min_samples parameters. To determine a suitable value for eps, we plotted the k**-distance graph**, which shows the distance to the k-th nearest neighbor for each point, sorted in ascending order.

Based on the graphs, eps=0.5 and min_samples=5 was chosen.



k-Distance Graph for Selecting eps in DBSCAN



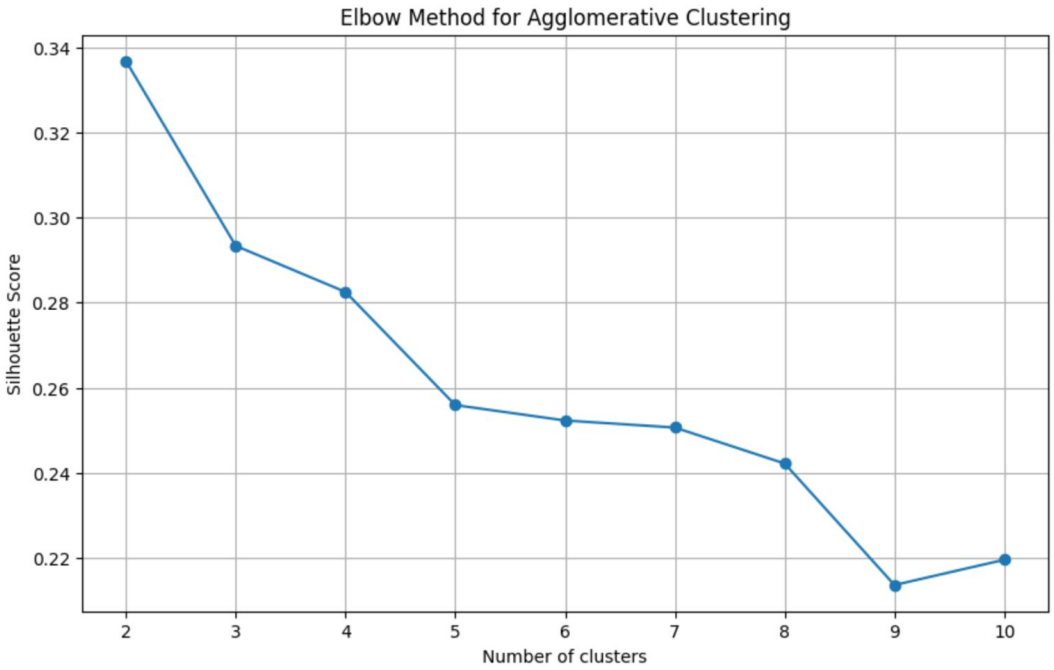
DBSCAN clustering result: Due to absence of a low-density separation, almost every point has been grouped into a single cluster which is a caveat of DBSCAN

Agglomerative Hierarchical Clustering

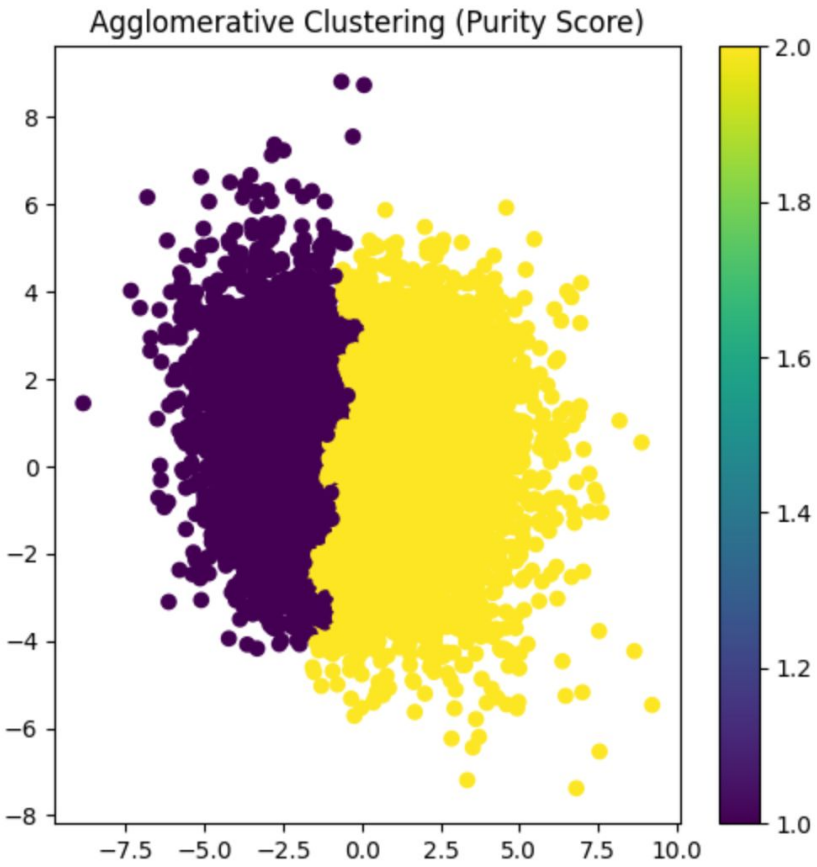
We experimented with four commonly used linkage methods: `ward`, average, `complete`, and `single`. After assigning clusters using each method, we evaluated the clustering quality using the **Purity Score**. The results are summarized below:

Linkage Method	**Purity Score**
Ward	0.85
Average	0.78
Complete	0.81
Single	0.65

Table 4.2: Purity Scores for Different Linkage Methods



Elbow for Agglomerative Clustering



Agglomerative Clustering Result

Chapter 5

Final Results

For the three tasks of classification, regression, and clustering, we have explored different models on differently preprocessed datasets. The following sections summarize the findings.

Classification

One-Hot encoding and Z-score normalizing was the preprocessing that gave the best results. Logistic Regression and Triple Layer Neural Network models gave the best performance.

Model	**Accuracy**	**Precision**	**Recall**	**F1-Score**
Logistic Regression	96.70%	93.19%	92.99%	93.09%
Artificial Neural Network	96.77%	93.21%	93.31%	93.26%

Table 5.1: Performance Metrics of Classification Models

Regression

One-Hot encoding and min-max normaling was the preprocessing that gave the best results. XGBoost model was the most suitable model with RMSE of 0.046.

Clustering

DBSCAN failed to identify clusters due to lack of low-density separating region. Agglomerative Clustering and k-Medoid Clustering were better suited for the given dataset.