**MACHINE LEARNING ALGORITHM APPLCATION FOR PROCESSING OF BANK LOAN APPLICATIONS**

By:

Ramkishore Rao

DSA 5900 / Credit Hrs: 4 hrs

Summer 2022

Faculty Advisors: Dr. Trafalis/Dr. Radhakrishnan

Faculty Coordinator: Dr. Beattie

**TABLE OF CONTENTS**

[1.0 Introduction 2](#_Toc106628605)

[2.0 Objectives 2](#_Toc106628606)

[3.0 Exploratory Data Analysis 2](#_Toc106628607)

[3.1 Analysis Summary 3](#_Toc106628608)

[3.2 Analysis Findings 7](file:///C:\Data%20Science%20and%20Analytics\DSA%205900\Ramkishore_Rao_DSA5900_Project.docx#_Toc106628609)

[4.0 Feature Evaluation/Extraction 8](file:///C:\Data%20Science%20and%20Analytics\DSA%205900\Ramkishore_Rao_DSA5900_Project.docx#_Toc106628610)

[4.1 Missing Value Analysis 8](file:///C:\Data%20Science%20and%20Analytics\DSA%205900\Ramkishore_Rao_DSA5900_Project.docx#_Toc106628611)

[4.2 Correlation Analysis 9](file:///C:\Data%20Science%20and%20Analytics\DSA%205900\Ramkishore_Rao_DSA5900_Project.docx#_Toc106628612)

[4.3 PCA Analysis 11](file:///C:\Data%20Science%20and%20Analytics\DSA%205900\Ramkishore_Rao_DSA5900_Project.docx#_Toc106628613)

[5.0 Machine Learning Modeling 13](#_Toc106628614)

[5.1 Logistic Regression 14](file:///C:\Data%20Science%20and%20Analytics\DSA%205900\Ramkishore_Rao_DSA5900_Project.docx#_Toc106628615)

[5.2 Multinomial Bayes 15](file:///C:\Data%20Science%20and%20Analytics\DSA%205900\Ramkishore_Rao_DSA5900_Project.docx#_Toc106628616)

[5.3 Decision Tree 16](file:///C:\Data%20Science%20and%20Analytics\DSA%205900\Ramkishore_Rao_DSA5900_Project.docx#_Toc106628617)

[5.4 Ensemble Trees 17](file:///C:\Data%20Science%20and%20Analytics\DSA%205900\Ramkishore_Rao_DSA5900_Project.docx#_Toc106628618)

[5.5 Deep Neural Network with Tensorflow/Keras 18](file:///C:\Data%20Science%20and%20Analytics\DSA%205900\Ramkishore_Rao_DSA5900_Project.docx#_Toc106628619)

[5.6 Tensorflow Federated/Keras 19](#_Toc106628620)

[5.7 Summary of Findings 20](#_Toc106628621)

[6.0 Conclusions 21](#_Toc106628622)

[7.0 References 22](#_Toc106628623)

**List of Tables, forthcoming**

Preliminary tables included now

**List of Exhibits, forthcoming**

Preliminary exhibits included now

**List of Appendices, forthcoming**

Preliminary appendix included now

# 1.0 Introduction

This project serves as my final practicum for my master’s degree in Data Science and Analytics being completed at the University of Oklahoma. As part of this project, various machine learning algorithms were applied to a bank loan dataset[[1]](#footnote-1) to aid in the processing of loan applications from consumers at a bank. For this study, a git hub repository developed by Dr. Jeff Heaton for his Deep Learning (DL)[[2]](#footnote-2) class at Washington University at St. Louis and his accompanying book[[3]](#footnote-3) were leveraged. In addition, class notes from Dr. Nicholson and from Dr. Diochnos were also utilized during the study.

The primary programming language used was Python, with its pre-existing modules. PostgreSQL was used for storing the data. Tableau has been used during the initial exploration phase of the data.

# 2.0 Objectives

Themain objective of the project is to use the existing bank loan dataset to develop back-end statistics models in order to provide a decision on the loan applications. Training, validation, and testing were performed using the existing dataset. An implementation plan is provided below.

# 3.0 Exploratory Data Analysis

**Table 1: Data Breakdown by Target Class**

Table

Description automatically generated

A bank loan dataset that contained 112 features was utilized in this study. Of the 112 features, one of the features was default\_date, i.e., this feature had the data on which default occurred. This feature was the target class, and if default had occurred, it was assigned a value of 1 and if default had not occurred, it was assigned a value of 0.

More explanation to follow in final deliverable.

## 3.1 Analysis Summary

A few tables and exhibits are provided in the following pages. They presented breakout of aggregated values of several features by target class value (i.e., 0 if debtor has not defaulted and 1 if debtor has defaulted). Further explanation to be provided in final deliverable.

**Table 2: Income Breakouts by Target Class**

A picture containing timeline

Description automatically generated

**Table 3: Interest Servicing Breakouts by Target Class**

A picture containing timeline

Description automatically generated

**Table 4: Liability Breakouts by Target Class**

A picture containing chart

Description automatically generated

**Table 5: Credit Rating by Median Probability of Default**

Timeline

Description automatically generated with medium confidence

**Table 6: Employment Status Counts Breakdown by Target Class**

Table

Description automatically generated

**Table 8: Education/Country Type Counts Breakdown by Target Class**

Table

Description automatically generated

**Table 7: Work Experience/Home Ownership Type Counts Breakdown by Target Class**

Table

Description automatically generated

**Table 10: Days to Payments Percentage of Total Breakdown by Target Class**

Table

Description automatically generated

**Table 9: Amount of Previous Credit Breakdown by Target Class**

Table

Description automatically generated

## 3.2 Analysis Findings

To be Explained More Thoroughly in Final Deliverable.

When aggregated, following can be ascertained from preliminary data analysis:

1. Higher income appears to result in lower default
2. Higher interest appears to result in higher default
3. Poorer credit rating was assigned a higher median probability of default – a decision variable used by the loan processors prior to processing of the loan
4. Higher previous credit obtained appears to result in lower default
5. Higher education appears to result in lower default
6. More prompt appears to result in lower default

# 4.0 Feature Evaluation/Extraction

This section does further data exploration:

1. Missing value analysis
2. Multi collinearity effects
3. Correlation between predictor variable and target variable
4. PCA analysis to reduce the high dimensional data set – need to evaluate further in final analysis. Will determine if this reduction can still result in reasonable prediction power of the models presented in Section 5.0
5. Final report may also include feature engineering to combine select features for better model interpretability and prediction power.

## 4.1 Missing Value Analysis

Missing values for key predictor variables is provided in Exhibit 1.

Further explanation to follow in final deliverable to help guide the reader in how features were lowered from the original 112 to what’s presented herein.

**Exhibit 1: Missing Values Count**

Chart

Description automatically generated

## 4.2 Correlation Analysis

Analysis was conducted to assess for multi-collinearity of predictor variables. The predictor variables that have correlation coefficient greater than 0.75 between each other are presented in Table 11.

**Table 11: Correlation Coefficients Between Variables**

Table

Description automatically generatedTable

Description automatically generated

In the final analysis, I will evaluate if variables that exhibit multi-collinearity need to be eliminated for models to have better interpretability and prediction power.

**Table 12 Continued: Correlation Coefficients Between Variables and Target Variable**

Table

Description automatically generated

**Table 12: Correlation Coefficients Between Variables and Target Variable**

Table

Description automatically generated

**Final Summary**

This shows that the correlation is not significant (less than 0.5, except for Status\_Late) between retained variables and the target class variable.

In final deliverable, will remove Status\_Late from the predictor variable set, and will explain this analysis in more detail.

This section will attempt to connect the “aggregated” breakout summaries presented in Section 2.0 with the correlation analysis presented herein.

Also, there are some variables may need to be grouped together as part of feature engineering to evaluate if prediction power of the models is better.

## 4.3 PCA Analysis

Preliminary PCA analysis was conducted to perform exploratory analysis and to evaluate whether the variance in the target class values can be explained by reducing dimensions of the predictor variables.

Initial analysis was conducted using only 5,000 dataset points. This number will be increased prior to the final deliverable.

Initial analysis indicates that 50% of the variance can be explained with 5 principal components (see Exhibit 2).

Separability in the target class is not clearly discernable when 3 principal components are evaluated (see Exhibit 3.

**Exhibit 2: Explained Variance vs Principal Component No.**

Chart, line chart

Description automatically generated

**Exhibit 3: Target Class Separation from**

**3 Principal Components**

Chart, scatter chart

Description automatically generated

PCAA Bi Plot results from this preliminary analysis is presented on Exhibit 4. Based on the “vector” representation of some of the features, it does appear that the first two components may be a reasonable assimilator of some of the predictor variables.

This will be further evaluated during the final analysis to be included in the final deliverable.

**Exhibit 4: PCA Bi Plot**

Chart, scatter chart

Description automatically generated with medium confidence

# 5.0 Machine Learning Modeling

Pythion’s sklearn and tensorflow/keras is utilized for this analysis. Explanation to follow in final deliverable. Tensorflow federated with keras will also be checked in the final analysis.

Data preparation included the following steps:

* 1. Prior to further evaluation, the predictor variables were segregated into continuous or categorical variables.
  2. The continuous predictor variables were scaled to ensure that the range of input values for the various variables are similar.
  3. One hot encoding was utilized to encode the categorical variables as either dummy or ordinal[[4]](#footnote-4).

Dataset was split into train, validation, and test components using sklearn. The weights for the train, validation, and test components are being developed. At this time, results from the test dataset following training on the train set is presented.

Final analysis will include testing on the validation data set and will also include results from 5-fold validation prior to testing on the test dataset.

Model optimization will be performed to tune hyperparameters in each type of model for the final analysis. Typical hyperparameters for the various models are as follows.

* + 1. Logistic Regression with its variants, include lasso, ridge regression, and elastic net – parameters for training may include L1 and L2 regularization parameters, solver, or class\_weights [[5]](#footnote-5).
    2. Naïve Bayes - parameter will be alpha, Additive (Laplace/Lidstone) smoothing parameter (0 for no smoothing and 1 for default)[[6]](#footnote-6).
    3. K nearest neighbors – parameter will be number of nearest neighbors to be used for model classification.
    4. Decision Tree Classifiers – parameters will include criterion (gini or logloss), splitter (best or random), max\_depth (depth of tree), etc[[7]](#footnote-7).
    5. ANN, tensorflow and tensorflow federated – parameters may include number of hidden layers, type of activation functions, regularization – L1, L2, dropout, and learning\_rate. Tensorflow’s in-built Bayesian Optimization will be utilized for hyperparameter tuning[[8]](#footnote-8).

## 5.1 Logistic Regression

#Basic Logistic Regression Model Fitting,

w/Default, Explanation to be provided in Final Deliverable

log\_reg = LogisticRegression()

log\_reg.fit(X\_train1, y\_train1)

y\_predict = log\_reg.predict(X\_train1)

Optimization, Hyerparameter Tuning is forthcoming

**Exhibit 5: Performance Evaluation: Logistic Regression**

Confusion Matrix, Test Dataset:

|  |  |  |
| --- | --- | --- |
|  | **Predicted**  **No Default** | **Predicted**  **Yes Default** |
| **Actual**  **No Default** | 25,341 | 1,846 |
| **Actual**  **Yes Default** | 2,721 | 11,894 |

|  |  |
| --- | --- |
| **Parameter** | **Value** |
| RMSE | 0.33 |
| Precision | 0.866 |
| Accuracy | 0.891 |
| Recall | 0.814 |
| F1\_Score | 0.839 |

**Exhibit 6: ROC Curve: Logistic Regression**

Chart, line chart

Description automatically generated

AUC = 0.873

## 5.2 Multinomial Bayes

#Basic MNB Model Fitting,

w/Default, Explanation to be provide in Final Deliverable

clf = MultinomialNB()

clf.fit(X\_train1, y\_train1)

y\_predict = clf.predict(X\_train1)

Optimization, Hyerparameter Tuning is forthcoming

**Exhibit 7: Performance Evaluation: Multinomial Bayes**

Confusion Matrix, Test Dataset:

|  |  |  |
| --- | --- | --- |
|  | **Predicted**  **No Default** | **Predicted**  **Yes Default** |
| **Actual**  **No Default** | 24,278 | 2,909 |
| **Actual**  **Yes Default** | 3,775 | 10,840 |

|  |  |
| --- | --- |
| **Parameter** | **Value** |
| RMSE | 0.40 |
| Precision | 0.788 |
| Accuracy | 0.840 |
| Recall | 0.741 |
| F1\_Score | 0.764 |

**Exhibit 8: ROC Curve: Multinomial Bayes**

Chart, line chart

Description automatically generated

AUC = 0.817

## 5.3 Decision Tree

5 stumps model, explanation to be provided in final deliverable

tree\_clf1 = DecisionTreeClassifier(criterion='entropy', max\_depth = 5)

tree\_clf1.fit(X\_train1, y\_train1)

y\_predict = tree\_clf1.predict(X\_train1)

Optimization, Hyerparameter Tuning is forthcoming

**Exhibit 9: Performance Evaluation: Decision Tree**

Confusion Matrix, Test Dataset:

|  |  |  |
| --- | --- | --- |
|  | **Predicted**  **No Default** | **Predicted**  **Yes Default** |
| **Actual**  **No Default** | 26,544 | 643 |
| **Actual**  **Yes Default** | 1,684 | 12,931 |

|  |  |
| --- | --- |
| **Parameter** | **Value** |
| RMSE | 0.236 |
| Precision | 0.953 |
| Accuracy | 0.944 |
| Recall | 0.885 |
| F1\_Score | 0.917 |

**Exhibit 10: ROC Curve: Decision Tree**

Chart, line chart

Description automatically generated

AUC = 0.931

## 5.4 Ensemble Trees

Explanation to be provided in final deliverable

ada\_clf1 = AdaBoostClassifier(DecisionTreeClassifier(max\_depth =1), n\_estimators = 20)

ada\_clf1.fit(X\_train1, y\_train1)

y\_predict = ada\_clf1.predict(X\_train1)

Optimization, Hyerparameter Tuning is forthcoming

**Exhibit 11: Performance Evaluation: Ensembles Trees**

Confusion Matrix, Test Dataset:

|  |  |  |
| --- | --- | --- |
|  | **Predicted**  **No Default** | **Predicted**  **Yes Default** |
| **Actual**  **No Default** | 26,063 | 1,124 |
| **Actual**  **Yes Default** | 1,972 | 12,643 |

|  |  |
| --- | --- |
| **Parameter** | **Value** |
| RMSE | 0.272 |
| Precision | 0.918 |
| Accuracy | 0.926 |
| Recall | 0.865 |
| F1\_Score | 0.891 |

**Exhibit 12: ROC Curve: Ensemble Trees**

Chart, line chart

Description automatically generated

AUC = 0.912

## 5.5 Deep Neural Network with Tensorflow/Keras

Explanation to be provided in Final Deliverable

1 input layer, 2 hidden layers, and 1 output layer, RELU activation for all except output – sigmoid activation

model = Sequential()

model.add(Dense(100, input\_dim=X\_train1.shape[1], activation='relu',

                kernel\_initializer='random\_normal'))

model.add(Dense(50,activation='relu',kernel\_initializer='random\_normal'))

model.add(Dense(25,activation='relu',kernel\_initializer='random\_normal'))

model.add(Dense(1,activation='sigmoid',kernel\_initializer='random\_normal'))

model.compile(loss='binary\_crossentropy',

              optimizer=tensorflow.keras.optimizers.Adam(),

              metrics =['accuracy'])

monitor = EarlyStopping(monitor='val\_loss', min\_delta=1e-3,

    patience=5, verbose=1, mode='auto', restore\_best\_weights=True)

model.fit(X\_train1,y\_train1,validation\_data=(X\_test,y\_test),

          callbacks=[monitor],verbose=2,epochs=1000)

Optimization, Hyerparameter Tuning is forthcoming

**Exhibit 13: Performance Evaluation: Tensor Flow/Keras**

Confusion Matrix, Test Dataset:

|  |  |  |
| --- | --- | --- |
|  | **Predicted**  **No Default** | **Predicted**  **Yes Default** |
| **Actual**  **No Default** | 26,101 | 1,086 |
| **Actual**  **Yes Default** | 1,768 | 12,847 |

|  |  |
| --- | --- |
| **Parameter** | **Value** |
| RMSE | 0.261 |
| Precision | 0.922 |
| Accuracy | 0.931 |
| Recall | 0.879 |
| F1\_Score | 0.900 |

**Exhibit 14: ROC Curve: Tensor Flow/Keras**

A picture containing text, screenshot

Description automatically generated

## 5.6 Tensorflow Federated/Keras

forthcoming

## 5.7 Summary of Findings

Forthcoming

Preliminary results are promising, with decision tree and deep neural networks providing the best results. Final analysis will include optimization and hyperparameter tuning.

# 6.0 Conclusions

forthcoming

# 7.0 References

Footnote referencing will be replaced with traditional referencing to be included in this section.

**APPENDIX**

Feature No Feature Name

1 ReportAsOfEOD

2 LoanId

3 LoanNumber

4 ListedOnUTC

5 BiddingStartedOn

6 BidsPortfolioManager

7 BidsApi

8 BidsManual

9 PartyId

10 NewCreditCustomer

11 LoanApplicationStartedDate

12 LoanDate

13 ContractEndDate

14 FirstPaymentDate

15 MaturityDate\_Original

16 MaturityDate\_Last

17 ApplicationSignedHour

18 ApplicationSignedWeekday

19 VerificationType

20 LanguageCode

21 Age

22 DateOfBirth

23 Gender

24 Country

25 AppliedAmount

26 Amount

27 Interest

28 LoanDuration

29 MonthlyPayment

30 County

31 City

32 UseOfLoan

33 Education

34 MaritalStatus

35 NrOfDependants

36 EmploymentStatus

37 EmploymentDurationCurrentEmployer

38 EmploymentPosition

39 WorkExperience

40 OccupationArea

41 HomeOwnershipType

42 IncomeFromPrincipalEmployer

43 IncomeFromPension

44 IncomeFromFamilyAllowance

45 IncomeFromSocialWelfare

46 IncomeFromLeavePay

47 IncomeFromChildSupport

48 IncomeOther

49 IncomeTotal

50 ExistingLiabilities

51 LiabilitiesTotal

52 RefinanceLiabilities

53 DebtToIncome

54 FreeCash

55 MonthlyPaymentDay

56 ActiveScheduleFirstPaymentReached

57 PlannedPrincipalTillDate

58 PlannedInterestTillDate

59 LastPaymentOn

60 CurrentDebtDaysPrimary

61 DebtOccuredOn

62 CurrentDebtDaysSecondary

63 DebtOccuredOnForSecondary

64 ExpectedLoss

65 LossGivenDefault

66 ExpectedReturn

67 ProbabilityOfDefault

68 PrincipalOverdueBySchedule

69 PlannedPrincipalPostDefault

70 PlannedInterestPostDefault

71 EAD1

72 EAD2

73 PrincipalRecovery

74 InterestRecovery

75 RecoveryStage

76 StageActiveSince

77 ModelVersion

78 Rating

79 EL\_V0

80 Rating\_V0

81 EL\_V1

82 Rating\_V1

83 Rating\_V2

84 Status

85 Restructured

86 ActiveLateCategory

87 WorseLateCategory

88 CreditScoreEsMicroL

89 CreditScoreEsEquifaxRisk

90 CreditScoreFiAsiakasTietoRiskGrade

91 CreditScoreEeMini

92 PrincipalPaymentsMade

93 InterestAndPenaltyPaymentsMade

94 PrincipalWriteOffs

95 InterestAndPenaltyWriteOffs

96 PrincipalBalance

97 InterestAndPenaltyBalance

98 NoOfPreviousLoansBeforeLoan

99 AmountOfPreviousLoansBeforeLoan

100 PreviousRepaymentsBeforeLoan

101 PreviousEarlyRepaymentsBefoleLoan

102 PreviousEarlyRepaymentsCountBeforeLoan

103 GracePeriodStart

104 GracePeriodEnd

105 NextPaymentDate

106 NextPaymentNr

107 NrOfScheduledPayments

108 ReScheduledOn

109 PrincipalDebtServicingCost

110 InterestAndPenaltyDebtServicingCost

111 ActiveLateLastPaymentCategory

112 Target Class: Defaulted

1. : Loan Dataset file from <https://www.bondora.com/en/public-reports> [↑](#footnote-ref-1)
2. https://github.com/jeffheaton/t81\_558\_deep\_learning [↑](#footnote-ref-2)
3. Applications of Deep Neural Networks with Keras, Jeff Heaton, Fall 2022.0 [↑](#footnote-ref-3)
4. Refer to Section 2.2.2 Encoding Categorical Variables as dummies, Applications of Deep Neural Networks with Keras, Jeff Heaton, Fall 2022.0 [↑](#footnote-ref-4)
5. https://scikit-learn.org/stable/modules/generated/sklearn.linear\_model.LogisticRegression.html [↑](#footnote-ref-5)
6. https://scikit-learn.org/stable/modules/generated/sklearn.naive\_bayes.MultinomialNB.html [↑](#footnote-ref-6)
7. https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html [↑](#footnote-ref-7)
8. 8.4 – Bayesian Hyperparameter Optimization for Keras, Applications of Deep Neural Networks with Keras, Jeff Heaton, Fall 2022.0 [↑](#footnote-ref-8)