CF969-7-SP

Assignment 2: Loan Default Prediction

APRIL 15, 2024

UNIVERSITY OF ESSEX
Osama Sher 2312569

Table of Contents

Abstract:

This assignment focuses on different machine learning methodologies and applies them to data on the financial status of individuals who have taken loans. The machine learning objective is to train a model that accurately predicts if an individual defaults or not. Three regression methods (Linear, Lasso, Ridge) are used and additionally, Random Forest and Neural Network methods are used for training models. The models are evaluated and compared to identify the best model.

Introduction:

Machine learning is an important part of various domains, including the financial sector. In the Financial Sector, predicting individual behavior for example loan default is essential for risk assessment and decision-making. The dataset being analyzed includes information about the financial profiles of individuals who have applied for loans. The objective is to develop robust predictive models that can accurately classify individuals into default or non-default categories using different regression methodologies such as Linear, Lasso, and Ridge, alongside more complex algorithms like Random Forest and Neural Networks. Each method has its unique advantage in addressing specific nuances present in the dataset, thus providing a comprehensive exploration of modeling approaches. After training of each model, it is evaluated on unseen data, and the metrics are compared among the models to identify the best-performing model.

Preprocessing:

The data set provided has 33 variables and 220608 observations. The data set size is large and this poses a challenge when training models. The first step is to clean the data as much as possible to improve the predictive power of the models.

Null Values:

The dataset has 4 columns with null values. The column "mths_since_last_delinq" has the most null values. Initially, the null values were imputed with mean values, but this can result in bias. Removing the observations with null values in other columns won't result in significant data loss.

Categorical Values:

During the preprocessing of datasets, categorical columns such as 'grade', 'emp_length', 'home_ownership', and 'application_type' are transformed into numerical values using 'LabelEncoder' from sklearn, following the removal of null values. This enables regression models to effectively handle these columns.

Correlation with Target Variable:

After the previous steps, the absolute correlation of each variable with the target variable is calculated to identify the highest and lowest correlated columns. This can be used to identify significant or insignificant columns in our dataset, which can be dealt with accordingly.

Variable	*Correlation*
Recoveries	0.523448
Collection_recovery_fee	0.504759
Grade	0.224438
Total_rec_prncp	0.213357
Int_rate	0.199086
Last_pymnt_amnt	0.173894
Last_pymnt_inv	0.129822
Total_pymnt	0.129816
Total_rec_late_fee	0.100838
Inq_last_6mths	0.090857

Variable	*Correlation*
Tot_coll_amt	0.000463
Collections_12_mths_ex_med	0.004578
Mths_since_last_delinq	0.006298
Acc_now_delinq	0.011173
Emp_length	0.013985
Total_acc	0.017877
Delinq_2yrs	0.017902
Revol_bal	0.019001
Open_acc	0.020041
Loan_amnt	0.020313

Figure 1 : 10 Highest correlated (Top figure) 10 lowest correlated(Bottom) Variables to Target Variable

Based on the correlation analysis between predictor variables and the target variable, we have identified only two variables that exhibit high correlation, while the rest of the variables have correlations close to zero. Given the low correlation values for most of the variables, removing these variables may result in a loss of valuable information. Therefore, only the three variables with the lowest correlations are eliminated, including the "Mths_since_last_delinq" column, which contains a high number of null values. This approach avoids the need

to impute missing values and preserves essential data by preventing the removal of rows containing empty values for this variable.

Scaling Data:

The data is scaled for better performance of the models. Scaling the data helps improve convergence of algorithms resulting in finding optimal solutions efficiently. Additionally, scaling helps prevent dominant features in the data. Our dataset has variables with large scales that dominate the learning process resulting in the model ignoring the small-scale features. Scaling prevents this by ensuring that all features contribute equally. In the assignment, this is achieved by "StandardScaler" from sklearn.

Training Models:

Linear Regression:

Linear regression model is fit to the pre-processed data and the mean squared error of the model on the training set and test set are calculated.

Training set MSE	**0.06683**
Test set MSE	0.06768

Table 1: Linear Regression Model MSE

Ridge Regression:

The ridge regression model is fit to the same data and the mean squared error of the model on both train and test sets are calculated. Ridge regression has a hyperparameter alpha that is used as a penalty to cater to multicollinearity between predictor variables and avoid overfitting. The Tuning select alpha = 99.99 as the best value for the model. Cros-validation is used to find the best alpha value, which is then used in the model, and MSE values for training and test sets are calculated.

Alpha	**99.99**
Training set MSE	0.06683
Test set MSE	0.06768

Table 2: Ridge Regression Model MSE

Lasso Regression:

Lasso regression model is fit to the same data and the mean squared error of the model on both train and test sets are calculated. Hyperparameter tuning is done to identify the best alpha value for the model and then MSE values are calculated to evaluate the model.

Alpa	**0.01**
Training set MSE	0.06847
Test set MSE	0.06939

Table 3: Lasso Regression Model MSE

Random Forest:

Random Forest is a robust method for classification and regression tasks. This method unlike other methods does not require the removal of insignificant variables as it ignores such variables automatically during the learning process. This method creates a forest of decision trees and then aggregates their predictions for better performance. Two hyperparameters require tuning to create the best random forest classifier, these are max_depth and n_estimators. These decide how many random decision trees need to be created and the depth of each tree. Having a higher number of both makes the model more complex and low values can result in inaccurate predictions. Therefore an optimal value for these hyperparameters is calculated and the model is then run on both training and test sets for evaluation.

R-Forest Hyperparameters	
N_estimators	50
Max_depth	3

Table 4: Random Forest Hyperparameters (Best Model)

Training set MSE	**0.03769**
Test set MSE	0.03920
Train set Accuracy	0.96230
Test set Accuracy	0.96079

Table 5: Random Forest Evaluation

Random Forest is a robust classifier and it calculates the importances of the features during the learning process. We can check how important each feature is and which feature holds the most significance. The below Plot was obtained for importances of each feature. According to the Random Forest Classifier the most important features are “recoveries”, “collection_recovery_fee”, “last_pymnt_amnt”, “total_rec_prncp”, “grade” and “int_rate”. The Below figure shows the plot of feature importances from highest to lowest.

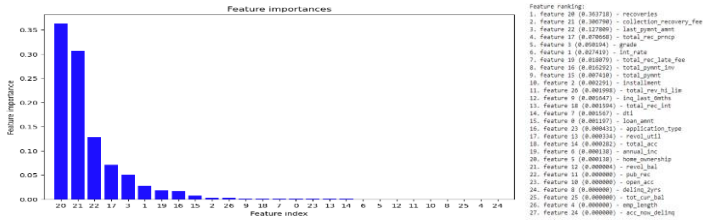

Figure 2: Random Forest Feature Importances.

Neural Network:

Neural Networks are a machine learning model that imitates the structural organization of the human brain. They consist of different interconnected layers of nodes called neurons that receive input signals, process them, and produce output signals. Specific settings need to be adjusted to optimize the network's performance for specific tasks. In our binary classification task, a neural network with a sigmoid activation function is implemented that generates a probability between 1 and 0. The network comprises fully connected layers with two hidden layers containing 64 and 32 neurons, respectively. To capture complex interactions and achieve

better prediction of default, ReLU activation is used to introduce non-linearity to the model. The loss function used is "binary_crossentropy," which is best suited for binary classification tasks. This function measures the difference between predicted probabilities and actual labels and guides the model to minimize misclassification. Adam optimizer is used for efficient training to minimize the loss function. This combination of parameters effectively defines our neural network to suit our specific task.

Training set MSE	**0.02748**
Test set MSE	0.03160
Train set Accuracy	0.96737
Test set Accuracy	0.96289

Table 5: Neural Network (ANNs) Evaluation

Evaluation:

Multiple models are trained on our data to anticipate defaults. Each model possesses distinct advantages and necessitates varying evaluation metrics. Mean squared error is ideal for assessing regression tasks, whereas accuracy is well-suited for classification tasks. To ensure a thorough comparison between our models, we have calculated MSE for all of them. For models such as Random Forest and Neural Network, we have additionally computed accuracy for individual performance evaluation. The table below presents a comparison of the metrics for each model (Best Model Metric Highlighted in Yellow).

Model	**Training MSE**	**Testing MSE**	**Training Accuracy**	**Testing Accuracy**
Linear Regression	0.06683	0.06768	NA	NA
Ridge Regression	0.06683	0.06768	NA	NA
Lasso Regression	0.06847	0.06939	NA	NA
Random Forest	0.03867	0.03984	0.96132	0.96015
Neural Network	0.02748	0.03160	0.96737	0.96289

Table 6: Models Evaluation

Conclusion:

After evaluating the performance of different models on the datasets, we have found that all models perform well on unseen data. Ridge regression with L2 regularization provides a high alpha and produces low Mean Squared Error (MSE) values for both the test and training sets, indicating no overfitting. Lasso with L1 regularization, on the other hand, produces an alpha of 0.01, suggesting that no regularization is required and essentially performs linear regression. With tuned hyperparameters, Random Forest Classifier performs well, with a low maximum depth to ensure no overfitting, and high accuracy on unseen data. The Neural Network model has the best performance, with the highest accuracy and lowest MSE, but with slightly more overfitting compared to other models. Therefore, the best model for loan default prediction in this assignment is the Neural Network model.