Homework 2 - CS584

Hoang-Dung Bui
Team Name: bui1720
Mason ID: G01301478
Rank:
Highest Public Score: 0.64

1 Introduction

In homework 2, I built a KNN program, Artificial Neural Network (ANN), Bayesian Classifier, and Decision Tree to classify the Credit Risk Score Prediction. The results from the models are shown as following:

- For KNN - the highest F-1 score: 0.59 (k=3)
- For Decision tree - the highest F-1 score: 0.63
- For Bayesian classifier - the highest F-1 score: 0.64
- For Artificial Neural Network - the highest F1-score: 0.6

2 KNN

Model's selection:

- The number of neighbors are tested: k = 3 or k = 5
- Majority voting
- distance/similarity: euclidean

The KNN program is similar to the program in homework 1. It contains three functions: *data_preprocessing*, *distance* and *main*.

1. *data_preprocessing*: This function removes some neutral column data and break down the attributes and the label.
2. *distance*: calculate the distance/similarity between two data. In this case, the *euclidean* is implemented.
3. *main function*: it reads data from files, break them into attribute data and label; Then two for loops are used to compute the distance between each data from train_data set and test_data set. The top k-nn data are selected to vote to decide the label of the the test data.

To change the number of neighbors, the parameter k_nn needed to be adjusted.

Experiment Results: The highest F-1 score: 0.59

3 Decision Tree

In this model, *sklearn* and *pandas* packages are used to read the data and build the model. Is is simple program with the available functions in the libraries. It removed the neutral data attributes by *drop* commands. In the column F10 and F11, the string data are converted to ordinal data by using the fit_transform in LabelEncoder package. The decision tree model is built by calling the class DecisionTreeClassifier and trained by fit function. The data is tested by function *predict*.

Model's selection:

- The depth of the decision tree: 15
- features selected to branch: GINI/entropy

Experiment Results Reach the F1-score is 0.63

4 Bayesian Classifier

In this classifier, I built the algorithm from scratch. It consist of a pre-processing data function, and followed by a long main function.

The continuous data are divided into subsets with thresholds. In F1 column, the time from graduation were divided into three sets: before 5 years, from 5 years to 15 years and longer than 15 years. In F2 column, the working hours are separated into four sets: 0 to 10, 10 to 25, 25 to 50, and longer. In columns F6 and F7, the data were segmented into three groups: equal to zero, from zero to 500, and greater than 500. The categorical data are kept the same.

In the main function, there are two main parts. The first one calculate the probabilities of each class and attributes. In the second part, it is a loop to calculate the probabilities of each class for each data row, and the decision is based on the comparison between two class probabilities.

Experiment Results: Reach the F1-score is 0.64

5 Artificial Neural Network

The ANN was built based on Pytorch package, and ran on Google Colab. The first part is to pre-process the data, which divide the data's attributes into numerical and string. The string data are converted into categorical data type, then embedding N dimensional vector. All data are then converted to tensor form to run on GPU.

The ANN is developed as a class named Model. By Pytorch framework, the models' parameters are defined, and the forward propagation is built. The backward propagation is done by function *backward* in Pytorch. The optimizer and loss function are selected based on the application. In this case, Adam optimizer and Cross Entropy Loss are chosen for the mode.

Model's selection/parameters:

- (3-4) layers with the varied nodes numbers in each layer.
- Optimization methods: Stochastic Gradient descent and Adam optimizer
- Training episodes: 400-600 epochs

Experiment Results: The highest F1-score is 0.6.

6 Conclusion

As comparing the F1-score among, the Bayesian classifier provided the highest score with 0.64. The result surprised me because I usually think a neural network will provide better result. Otherwise, I set up inappropriate parameters for the ANN's model.