German Credit Risk Score

Group 3

Wahad Latif, Bradley Childs, Siu Sing Li, Neerja Natu, Kuangzhong Niu

Background

- Used by financial institutions, employers, insurance companies to determine an individual’s creditworthiness
- They are a huge determining factor for whether you can:
  - Get a loan
  - Buy a house
  - Get a job
- It is important that these scores are an accurate representation of an individual

Dataset

- German Credit dataset from UCI Machine Learning Repository
- 1000 instances and 20 attributes, binary label indicating good or bad credit risk
    - 1 = good, 2 = bad

Attribute 1	Integer	Checking account
Attribute 2	Categorical	Duration
Attribute 3	Integer	Credit history
Attribute 4	Categorical	Purpose
Attribute 5	Categorical	Credit amount
Attribute 6	Integer	Savings account
Attribute 7	Categorical	Employment
Attribute 8	Integer	Installment rate
.....	...	Marital status
Attribute 20	Binary	Foreign worker

Data Preprocessing

- Converting primarily categorical data into numeric or one-hot encoding for easier use
- For variables like employment history or savings, we could use numeric label encoding
- For variables like sex and marital status we used one-hot encoding
- We chose to split the dataset into 10 sub-datasets based on the attribute ‘purpose of loan’, models were trained and tested on these sub-datasets separately

Models

Logistic Regression

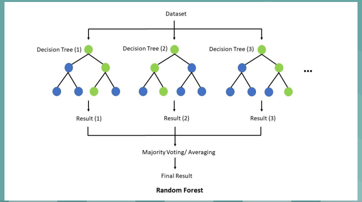
- Hyperparameter tuning
- Separate hyperparameters for each purpose to develop different models for different types of loans

Purpose	C, penalty, solver, tol
Business	0.01, l2, liblinear, 1e-3
New Car	1, l2, saga, 1e-4
Used Car	0.1, l2, saga, 1e-3
Education	0.001, l2, liblinear, 1e-3
Furniture	0.1, l2, liblinear, 1e-3
Radio/TV	1, l1, liblinear, 1e-3
Repairs	0.001, l1, liblinear, 1e-3

Random Forest

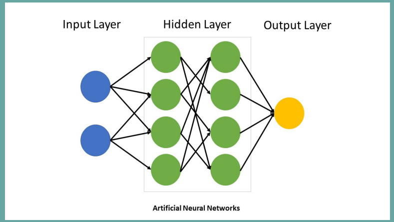
- The following were the hyperparameters we selected
- We again decided to split the dataset by purpose and build separate models

Purpose	estimators, split, leaf
Appliance	50, 5, 1
Business	300, 5, 2
New Car	50, 2, 3
Used Car	100, 2, 1
Education	50, 2, 3
Furniture	300, 2, 2
Other	50, 2, 1
Radio/TV	150, 5, 1
Repairs	50, 2, 2



Multilayer Perceptron (ANN)

- Because our dataset was only 1000 entries, we decided not to further split it up for the Neural Network



Results

- Logistic regression produced the following accuracies:

Purpose	Accuracy	MSE
Business	0.750	0.250
New Car	0.702	0.298
Used Car	0.952	0.048
Education	0.600	0.400
Furniture	0.730	0.270
Radio/TV	0.839	0.161
Repairs	0.800	0.200

Average Accuracy across all the logistic regression models: **77%**

- Random Forest produced the following accuracies:

Purpose	Accuracy	MSE
Appliance	0.667	0.333
Business	0.750	0.250
New Car	0.809	0.191
Used Car	0.904	0.095
Education	0.500	0.500
Furniture	0.702	0.297
Other	0.333	0.667
Radio/TV	0.768	0.232
Repairs	0.400	0.600

Average Accuracy across all the random forest models: **65%**

- Neural network produced the following results:

```
Accuracy :  0.752
Mean Square Error :  0.248
Confusion Matrix for each label :
[[[ 60  86]
  [ 38 316]]

 [[316  38]
  [ 86  60]]]
Classification Report :
              precision    recall  f1-score   support

           0       0.79      0.89      0.84       354
           1       0.61      0.41      0.49       146

   micro avg       0.75      0.75      0.75       500
   macro avg       0.70      0.65      0.66       500
weighted avg       0.74      0.75      0.74       500
 samples avg       0.75      0.75      0.75       500
```

Average Accuracy on the ANN model: **75%**

Github Link: https://github.com/siusing-li/171_final

Video Demo