CS 484: Introduction to Machine Learning

Fall Semester 2023 Assignment 4

Oduri Sai Ram

A20522183

Question 1 (50 points)

The **Homeowner_Claim_History.xlsx** contains the claim history of 27,513 homeowner policies. The following table describes the eleven columns in the HOCLAIMDATA sheet.

Name	**Description**	**Categories**
policy	Policy Identifier	
exposure	Duration a Policy is Exposed to Risk Measured in Portion of a Year	
num_claims	Number of Claims in a Year	
amt_claims	Total Claim Amount in a Year	
f_primary_age_tier	Age Tier of Primary Insured	< 21, 21 - 27, 28 - 37, 38 - 60, > 60
f_primary_gender	Gender of Primary Insured	Female, Male
f_marital	Marital Status of Primary Insured	Not Married, Married, Un-Married
f_residence_location	Location of Residence Property	Urban, Suburban, Rural
f_fire_alarm_type	Fire Alarm Type	None, Standalone, Alarm Service
f_mile_fire_station	Distance to Nearest Fire Station	< 1 mile, 1 - 5 miles, 6 - 10 miles, > 10 miles
f_aoi_tier	Amount of Insurance Tier	< 100K, 100K - 350K, 351K - 600K, 601K - 1M, > 1M

We want to predict the *Frequency* which is *number of claims per unit of exposure* using the above features. We first divide the reported number of claims by the exposure. This gives the *Frequency*. Next, we put the policies into four groups according to their *Frequency* values.

Frequency Group	**Frequency Value**
0	Frequency = 0
1	0 < Frequency <= 1
2	1 < Frequency <= 2
3	2 < Frequency <= 3
4	3 < Frequency

We will use the above Frequency Group as our target variable which has four levels.

After dropping the missing target values, we will divide the observations into the training and the testing partitions. Observations whose Policy Identifier starts with the letters A, G, and P will go to the training partition. The remaining observations go to the testing partition.

Since we have sufficient computing resources, we will train multinomial logistic models for all the possible subsets of combinations of the seven categorical predictors, namely, *f_aoi_tier*, *f_fire_alarm_type*, *f_marital*, *f_mile_fire_station*, *f_age_tier*, *f_primary_gender*, and *f_residence_location*. All models must include the Intercept term. To help us select our "optimal" model, we will calculate the AIC and the BIC criteria of the Training partition, the Accuracy of the Testing partition, and the Root Average Squared Error of the Testing partition.

(a) (10 points) How many policies are in each of the four groups in the Training partition? Also, in the Testing partition?

```
Training Partition - Frequency Group Counts:
0    12184
1     5567
2     2584
3      746
4      929
Name: Frequency Group, dtype: int64

Testing Partition - Frequency Group Counts:
0    3010
1    1469
2     618
3     157
4     249
Name: Frequency Group, dtype: int64
```

(b) (10 points) What is the lowest AIC value on the Training partition? Also, which model produces that AIC value?

(c) (10 points) What is the lowest BIC value on the Training partition? Also, which model produces that BIC value?

B and C Ans.

```
Lowest AIC on Training Partition: 39.10713312130849
Model producing Lowest AIC: K=10, C=0.001, Max Iter=100

Lowest BIC on Training Partition: 220084.64749431767
Model producing Lowest BIC: K=10, C=0.001, Max Iter=100
```

(d) (10 points) What is the highest Accuracy value on the Testing partition? Also, which model produces that Accuracy value?

```
  warnings.warn(
Best Model Hyperparameters (Based on AIC/BIC): (10, 0.001, 100)
Accuracy (Testing): 0.5473378157368708
F1 Score (Testing): 0.3901946954001588
```

(e) (10 points) What is the lowest Root Average Squared Error value on the Testing partition? Also, which model produces that RASE value?

```
Lowest RASE on Testing Partition: 1.298106646228551
Model producing Lowest RASE: K=10, C=10, Max Iter=100
```

Question 2 (50 points)

The Center for Machine Learning and Intelligent Systems at the University of California, Irvine manages the Machine Learning Repository (https://archive.ics.uci.edu/ml/index.php). We will use two of the datasets in the repository for analyses, namely, the **WineQuality_Train.csv** for training and the **WineQuality_Test.csv** for testing.

The categorical target variable is *quality_grp*. It has two categories, namely, 0 and 1. The input features are *alcohol*, *citric_acid*, *free_sulfur_dioxide*, *residual_sugar*, and *sulphates*. These five input features are considered interval variables.

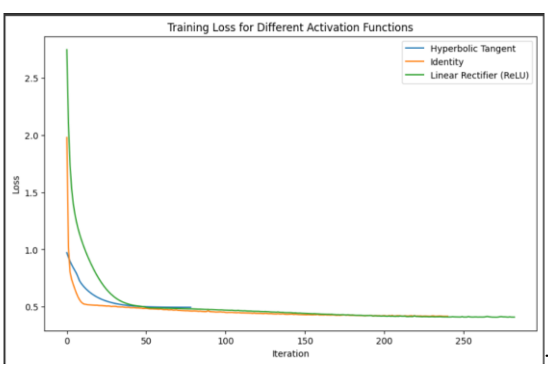
We will train a Multi-Layer Perceptron neural network with the following specifications.

1. Perform a grid search to select the most desired network structure.
2. The maximum number of iterations is 10000.
3. The random seed is 2023484.
4. Try all the **Hyperbolic Tangent**, the **Identity**, and the **Linear Rectifier** activation functions.
5. Try the number of layers from 1 to 10 inclusively with an increment of 1.
6. Try the common number of neurons per layer from 2 to 10 inclusively with an increment of 2.

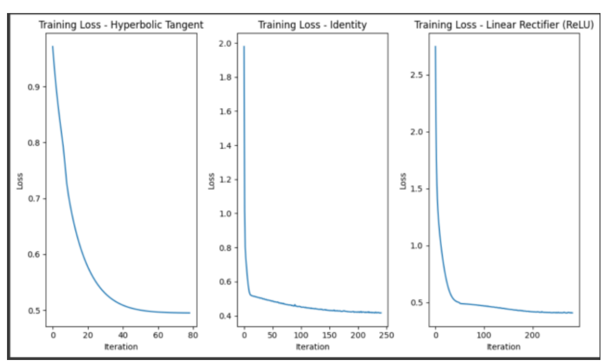
We will predict an observation with *quality_grp* of 1 if Prob(*quality_grp* = 1) ≥ 1.5c where c is the proportion of observations where *quality_grp* = 1 in the training partition. Otherwise, the predicted *quality_grp* is 0.

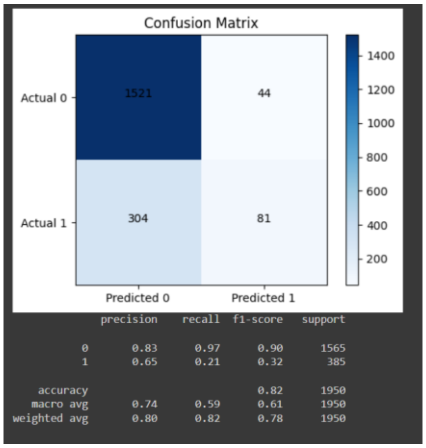
```
Fitting 3 folds for each of 150 candidates, totalling 450 fits
Best Model Parameters: {'activation': 'logistic', 'hidden_layer_sizes': (10, 10)}
Accuracy on Test Data: 0.8215384615384616
```

Some plots for the training_loss for activation functions



+





(a) (10 points). What is the proportion of observations where *quality_grp* = 1 in the training partition?

```
# Calculate the proportion of quality_grp = 1 in the training partition
proportion_quality_1 = (y_train_split == 1).sum() / len(y_train_split)

print(f"Proportion of quality_grp = 1 in the training partition: {proportion_quality_1:.2f}")
```

```
Proportion of quality_grp = 1 in the training partition: 0.20
Proportion of quality_grp = 1 in the training partition: 0.20
```

(b) (10 points). What is the proportion of observations where *quality_grp* = 1 in the testing partition?

```
[9] # Calculate the proportion of quality_grp = 1 in the testing partition
proportion_quality_1_test = (test_data["quality_grp"] == 1).sum() / len(test_data)

print(f"Proportion of quality_grp = 1 in the testing partition: {proportion_quality_1_test:.2f}")
```

```
Proportion of quality_grp = 1 in the testing partition: 0.20
```

(c) (10 points). Show your grid search results in a table. The table should contain (1) the activation function type, (2) the number of layers, (3) the common number of neurons per layer, (4) the number of iterations performed (*n_iter_* attribute), (5) the best loss value (*best_loss_* attribute), (6) the root average squared error of the testing partition, (7) the misclassification rate of the testing partition, and (9) the elapsed time in seconds.

	Activation Function	Number of Layers	Number of Neurons	Number of Iterations	Best Loss	Root Mean Squared Error	Misclassification Rate	Elapsed Time (s)
0	logistic	1	2	695	0.411692	0.433235	0.187692	5.207618
1	logistic	2	2	656	0.410626	0.425471	0.181026	4.744748
2	logistic	3	2	47	0.498748	0.444338	0.197436	0.404819
3	logistic	4	2	108	0.498919	0.444338	0.197436	1.109493
4	logistic	5	2	104	0.498918	0.444338	0.197436	1.197575
..	...	...	...	...	...	...	...	...
145	relu	6	10	123	0.412049	0.412621	0.170256	2.491989
146	relu	7	10	111	0.397612	0.430861	0.185641	3.764477
147	relu	8	10	92	0.406278	0.462435	0.213846	2.938288
148	relu	9	10	94	0.411402	0.431455	0.186154	2.252472
149	relu	10	10	93	0.418122	0.417563	0.174359	2.282679

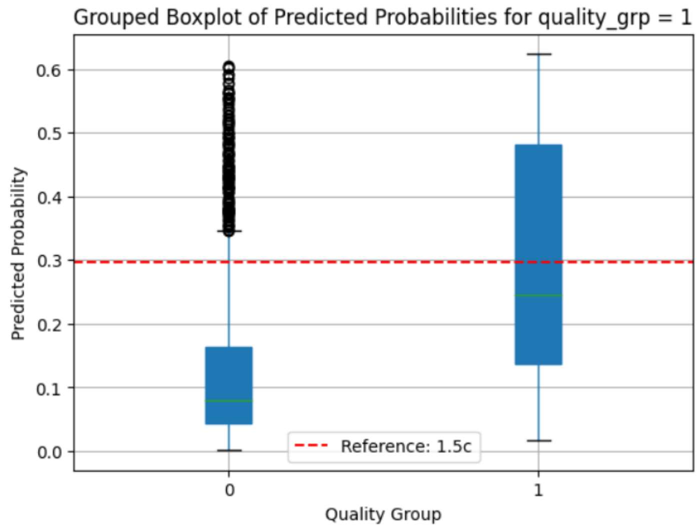
(d) (5 points). Among the networks that converged, which network structure yields the lowest misclassification rate on the testing partition? In the case of ties, choose the network with fewer neurons overall.

```
Best Network Structure:
Activation Function             relu
Number of Layers                   6
Number of Neurons                  4
Number of Iterations             255
Best Loss                   0.400065
Root Mean Squared Error     0.408876
Misclassification Rate      0.167179
Elapsed Time (s)            6.045437
Name: 115, dtype: object
```

(e) (5 points). Among the networks that converged, which network structure yields the lowest root average squared error on the testing partition? In the case of ties, choose the network with fewer neurons overall.

```
Best Network Structure:
Activation Function             relu
Number of Layers                   6
Number of Neurons                  4
Number of Iterations             255
Best Loss                   0.400065
Root Mean Squared Error     0.408876
Misclassification Rate      0.167179
Elapsed Time (s)            6.045437
Name: 115, dtype: object
```

(f) (10 points) We will choose the network structure that yields the lowest root average squared error as our final model. Based on the final model, generate a grouped boxplot for the predicted probability for *quality_grp* = 1 (i.e., if Prob(*quality_grp* = 1)) on the Testing data. The groups are the observed *quality_grp* categories. Add one reference line for Prob(*quality_grp* = 1) equals$1.5c$.


Grouped Boxplot of Predicted Probabilities for quality_grp = 1
Predicted Probability
0.0
0.1
0.2
0.3
0.4
0.5
0.6
Reference: 1.5c
0
1
Quality Group