CS 484: Introduction to Machine Learning

Fall Semester 2023 Assignment 4

# 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.

1. (10 points) How many policies are in each of the four groups in the Training partition? Also, in the Testing partition?
2. (10 points) What is the lowest AIC value on the Training partition? Also, which model produces that AIC value?
3. (10 points) What is the lowest BIC value on the Training partition? Also, which model produces that BIC value?
4. (10 points) What is the highest Accuracy value on the Testing partition? Also, which model produces that Accuracy value?
5. (10 points) What is the lowest Root Average Squared Error value on the Testing partition? Also, which model produces that RASE value?

# 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) where is the proportion of observations where *quality\_grp* = 1 in the training partition. Otherwise, the predicted *quality\_grp* is 0.

1. (10 points). What is the proportion of observations where *quality\_grp* = 1 in the training partition?
2. (10 points). What is the proportion of observations where *quality\_grp* = 1 in the testing partition?
3. (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.
4. (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.
5. (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.
6. (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.