ECE 5464 HW4 – Prof. Jones – HW 4 – UPDATE II

Due Tuesday, April 9, 2024 – 11:59 PM via Canvas

Part 1

In the Datasets section of Canvas, you will find a file called "VWXYZ.xlsx". It contains five possible predictors – V through Z – and one target variable, called "Binary".

You are to write a Python program to try a few logistic regression models on this data, and to measure the performance of each. To do this you should do the following:

1. Read in the data set
2. Prepare all data for use in modeling, as we have discussed.
3. For each of four scenarios (original data only, adding polynomial features, adding log features and adding both polynomial and log features):
    a. Per the scenario, add polynomial and/or log basis functions applied on the original features
    b. Normalize the dataset as we have learned in class
    c. Split the data set into training and test partitions, using a 70-30 split.
    d. Fit a logistic model to the training set
    e. Evaluate the model performance on the test set through a set of metrics:
        i. Classification accuracy
        ii. Area under the ROC curve (using the probability output of the model)
        iii. Precision
        iv. Recall
        v. Confusion matrix
        vi. TPR, FNR, FPR and TNR
    f. Print these metrics out, in a format similar to what I show below (note: the numbers are not right)
4. Write a paragraph explaining your observations on the differences, if any.

Part 2

In the Datasets section of Canvas, you will find a file called "Census_Supplement_Data.xlsx". It contains information about individuals and households that received some public assistance. There are many variables, with special attributes. I don't expect you to perform a lot of preprocessing that is based on the meaning of the variables, in this case. I give the types below, to guide your data preparation.

- For this problem, we will predict one continuous target variable, called "AGI".
- There are some "census weight" variables, which are NOT weights as we have been considering them in the course, and should not be used: HSUP_WGT, MARSUPWT and FSUP_WGT.
- Some columns are binary features: A_SEX and HAS_DIV.
- Some columns are ordinal: PEINUSYR.

- Some columns are categorical: PAW_YN, A_MARITL and PENATVTY.
- The other columns are numeric features.

You are to develop a set of neural network regression models to predict AGI. Your program should:

1. Read in the data set
2. Prepare all data for use in modeling, as we have discussed, including normalization.
3. Split the data set into training and test partitions, using a 70-30 split.
4. For each of the following set of possible neural network architectures, do the following:
    a. *Use the following hidden layer sizes:*
        i. *(4, 4)*
        ii. *(10, 6)*
        iii. *(32, 16)*
        iv. *(8, 3, 5)*
        v. *(12, 9, 10)*
    b. Define an ANN regressor with the given architecture (use the 'relu' activation function and early stopping with tol=0.0005)
    c. Fit the model to the training set (be sure that the model has converged)
    d. Plot the validation accuracy and training loss versus epoch for the model training
    e. Write out the following information about the model training:
        i. architecture (hidden layer sizes)
        ii. number of epochs
        iii. training set coefficient of determination, MSE and MAE
        iv. test set coefficient of determination, MSE and MAE
        v. generalization gap
5. Compare the various models; write a paragraph explaining your observations on the differences, if any.

Here is an example of the output from my program for Part 1 (don't pay attention to the numbers):

```
Classification test set: [182] iterations, accuracy = 0.6302, AUC = 0.7263
Precision = 0.636543,  Recall = 0.646464
[[12000  5432]
 [ 2222 14564]]
  TPR = 0.7777, FNR = 0.2222, FPR = 0.1111, TNR = 0.8888
```

Notes and tips:

- Calculations of the various metrics can be done using routines in the "metrics" section of the scikit-learn API; see https://scikit-learn.org/stable/modules/classes.html
- Note that the mean square error and area under the ROC curve (auroc or auc) are calculated using the continuous score output of the logistic model; pay careful attention to the difference in output of the model.predict() and model.predict_proba() functions.
- When I create my logistic regression classifier, I specify the number of iterations to be 1000, to give it more time to converge.

Your submission should include:

- A single WORD or PDF file including all of your Python code pasted in as plain text (no dark-mode or screen shots), the console output from your program, the plots for parts 1 and 2, and your paragraph of conclusions; and
- Your .py file(s) as an attachment (if you use Jupyter, do NOT attach the ipynb file, you must download your code as a .py).