DURHAM
COLLEGE
SUCCESS MATTERS

Introduction to Data Analysis

(DATA 1200)

Assignment - 3

Submitted by: Kartik Sojitra

(100723768)

To Professor: Sam Plati

1) Using Python develop the Multivariate Regression Algorithm script.

Step 1: Load libraries

```
#Load Libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

Step 2: Read dataset to pandas dataframe

```
#Read dataset to pandas dataframe
leuanalysisdata = pd.read_csv('./leuanalysis.csv')
leuanalysisdata.head()
```

Step 3: Identify how many classes we have

```
#Identify how many classes we have
leuanalysisdata.REMISS.unique()
```

Step 4: Assign data from first four column to X variable

```
# Assign data from first four columns to X variable
X = leuanalysisdata.drop('REMISS',axis=1).values
y = leuanalysisdata['REMISS'].values
```

Step 5: Train the data

```
#Train the Data
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y,
stratify=y,test_size = 0.20,random_state=100)
```

Step 6: Scale the data

```
#Scale the data
from sklearn.preprocessing
import StandardScaler scaler = StandardScaler()
scaler.fit(X_train)
X_train = scaler.transform(X_train)
X_test = scaler.transform(X_test)
```

Step 7: Neural Network

```
#Neural Network
from sklearn.neural_network import MLPClassifier
mlp = MLPClassifier(hidden_layer_sizes=(12, 8, 1),
max_iter=10000,random_state=100)
mlp.fit(X_train, y_train)
predictions = mlp.predict(X_test)
```

Step 8: Evaluate the Neural Network Algorithm

```
#Evaluate the Neural Network Algorithum
from sklearn.metrics import classification_report, confusion_matrix
print(confusion_matrix(y_test,predictions))
print(classification_report(y_test,predictions))
```

Step 9: Script for Logistic Regression

```
#Script for Logistic Regression
from sklearn.linear_model import LogisticRegression
for name,method in [('Logistic Regression',
LogisticRegression(solver='liblinear',random_state=100))]:
method.fit(X_train,y_train)
predict = method.predict(X_test)
print('Method: {}'.format(name))
```

Step 10: Evaluate the Neural Network Algorithm

```
#Evaluate the Neural Network Algorithum
from sklearn.metrics import classification_report, confusion_matrix
print(confusion_matrix(y_test,predictions))
print(classification_report(y_test,predictions))
```

Step 11: Script for Decision Tree

```
#Script for Decision Tree
from sklearn.tree import DecisionTreeClassifier
for name,method in [('DT', DecisionTreeClassifier(random_state=100))]:
method.fit(X_train,y_train)
predict = method.predict(X_test)
print('\nEstimator: {}'.format(name))
print(confusion_matrix(y_test,predict))
print(classification_report(y_test,predict))
```

(Plati, 2019)

2) Summery of key findings:

We have created a Neural Network Algorithm to predict the better accuracy of analysis using the six-dependent variable (CELL, SMEAR, INFIL, LI, BLAST, TEMP) and independent variable (REMISS). As most of the times, simple models would not be able to predict the analysis with higher accuracy, therefore we are using Neural Network Algorithm to get high accuracy.

- In this model, as per given information, **hidden layers considered as 12, 8 and 1** and **max_iter is 10,000.**
- Also, we used 20% of the dataset for training set as text size and 80% for the test data and **random state as 100.**

- To evaluate an algorithm, we used a **confusion matrix, precision, recall, and f1 score**, besides, confusion matrix and classification methods are used to find the scores.
- The half of the network **incorrectly classified by the confusion matrix** which demonstrates that the value of **True positive is 3** and **True Negative is 0 and the f1-score of 0.50 is not good** to create model**.**
- **Recall** describing the probability or positivity which determines that probability of individual do **not have cancer** and network does not define the value of **one has cancer**-based on **f1-score of 0.50**.
- **The linear logistic regression model** suggests that the connection between REMISS and the independent variable. The value is between 0 and 1. This model value is 50% which means the model can explain more than 50% of the variation.

(Plati, 2019)

3) Conclusion and Next step:

The value of total precision, recall, f1-score is 0.40, 0.50 and 0.50 respectively which is not good that implies our algorithm **was not exceptionally good to make a sensibly great prediction.**

The total precision, recall, and f1-score would be the ultimate accuracy of the model based on the Neural Network which defines the accuracy of the model. To explain, if the precision REMISS would be 95% and the value of the total precision, recall and f1-score is more precious than it would be a specific model.

The three hidden layers and sizes of the dataset are considered to increase the accuracy of the model. To improve the algorithm, we can add the 2nd layer is to be considered half of the 1st parameter and 3rd should be always one. In addition, precision level, recall, f1 score should be higher about 95% to get a more accurate neural network model.

References:

Plati, S. (2019). Neural Networks and Decision Trees.