SCS 3208 – MACHINE LEARNING & NEURAL COMPUTING

Assignment 01

19001908 | V K Widana Gamage

University of Colombo School of Computing

Table of Contents

Multivariate Line Regression

Selection of the data set

An Automobile data set was selected for the multivariate linear regression modelling because of the characteristics and associated tasks of the data set. It is a data set with 26 attributes and 205 instances. This data set consists of three types of entities: (a) the specification of an auto in terms of various characteristics, (b) its assigned insurance risk rating, (c) its normalized losses in use as compared to other cars.

See Data description

Objective

Using Automobile data to determine the correlation between the following variables.

Independent variables → 'wheel-base', 'curb-weight', 'engine-size', 'horsepower'
Dependent variable → 'price'

Data preprocessing

Since the data set contains missing values, to avoid them the instances has been removed as shown in the below code segment.
Other than that, data has been typecasted to float to get an accurate prediction.

```
import pandas
from sklearn import linear_model
from sklearn.metrics import r2_score
from sklearn.model_selection import train_test_split

#importing dataset
url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/autos/imports-85.data'
df = pandas.read_csv(url, header=None)

#removing rows with empty data
df = df[(df != '?').all(axis=1)]

#typecasting data to float
X = df[[9,13,16,21]].astype(float)
y = df[25].astype(float)
```

Model fitting

For the modelling purpose, data set was divided to two portions as a test data set of 30% and a train dataset of 70%. And the following library was used to do the task.

```
from sklearn.model_selection import train_test_split
```

Linear Regression model was used as follows using the necessary libraries.

```
from sklearn import linear_model
```

```
#Divide data to train set and test set. 30% as the test data
Xtrain, Xtest, Ytrain, Ytest = train_test_split( X, y, test_size = 0.3,
random_state=42)

#multivariative linear regression model
regr = linear_model.LinearRegression()
regr.fit(Xtrain, Ytrain)
```

Evaluating model

Predicting the price for the test data set and evaluating the R^2 score which is one of the metrics used to evaluate regression models . R^2 score is 0.785 which represents a strong correlation between independent and dependent variables. Hence, the model is successful.

```
from sklearn.metrics import r2_score
```

```
#predicted values for the test data
predicted = regr.predict(Xtest)

#r2 score
r2 = r2_score(Ytest,predicted)

print(r2)
```

```
Output: 0.7854644532757995
```

Checkout the code segment Link to Google Colabs

Logistic Regression

Selection of the data set

A room occupancy estimation data set has been selected which has multivariate characteristics and suitable for classification tasks. The selected data set has 16 attributes and 10,129 instances in total.

The experimental testbed for occupancy estimation was deployed in a 6m Ã— 4.6m room. The setup consisted of 7 sensor nodes and one edge node in a star configuration with the sensor nodes transmitting data to the edge every 30s using wireless transceivers. No HVAC systems were in use while the dataset was being collected.

Five different types of non-intrusive sensors were used in this experiment: temperature, light, sound, CO2 and digital passive infrared (PIR). The CO2, sound and PIR sensors needed manual calibration. For the CO2 sensor, zero-point calibration was manually done before its first use by keeping it in a clean environment for over 20 minutes and then pulling the calibration pin (HD pin) low for over 7s. The sound sensor is essentially a microphone with a variable-gain analog amplifier attached to it. Therefore, the output of this sensor is analog which is read by the microcontrollerâ€™s ADC in volts. The potentiometer tied to the gain of the amplifier was adjusted to ensure the highest sensitivity. The PIR sensor has two trimpots: one to tweak the sensitivity and the other to tweak the time for which the output stays high after detecting motion. Both of these were adjusted to the highest values. Sensor nodes S1-S4 consisted of temperature, light and sound sensors, S5 had a CO2 sensor and S6 and S7 had one PIR sensor each that were deployed on the ceiling ledges at an angle that maximized the sensorâ€™s field of view for motion detection.

The data was collected for a period of 4 days in a controlled manner with the occupancy in the room varying between 0 and 3 people. The ground truth of the occupancy count in the room was noted manually.

See Data description

Objective

Following data was taken into consideration for the logistic regression model.

Independent variables → 'S1_Temp','S2_Temp', 'S3_Temp','S4_Temp','S1_Light','S2_Light', 'S3_Light','S4_Light','S1_Sound','S2_Sound','S3_Sound', 'S4_Sound','S5_CO2', 'S5_CO2_Slope'

Dependent variable → ‘Room_Occupancy_Count’

Data preprocessing

Since the data set contains missing values, to avoid them the instances has been removed as shown in the below code segment.

Other than that, independent variables has been typecasted to float and the dependant variable has been typecasted to int with a value of 0 & 1. For the logistic regression model the prediction variable needs to be Boolean type. Therefore, when there is no occupancy the data is set to 0 and 1 otherwise.

```
import pandas
from sklearn import linear_model
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
scale = StandardScaler()

#importing dataset
url = 'https://archive.ics.uci.edu/ml/machine-learning-
databases/00640/Occupancy_Estimation.csv'
df = pandas.read_csv(url, header=None, names=['Date','Time','S1_Temp',
'S2_Temp', 'S3_Temp',  'S4_Temp',  'S1_Light', 'S2_Light', 'S3_Light',
'S4_Light', 'S1_Sound', 'S2_Sound', 'S3_Sound', 'S4_Sound', 'S5_CO2', '
S5_CO2_Slope','S6_PIR',  'S7_PIR', 'Room_Occupancy_Count'])

df = df.iloc[1: , :]

#removing rows with empty data
df = df[(df != '?').all(axis=1)]

#typecasting data to float
X = df.loc[:,['S1_Temp','S2_Temp',  'S3_Temp',  'S4_Temp',  'S1_Light',
 'S2_Light', 'S3_Light', 'S4_Light', 'S1_Sound', 'S2_Sound', 'S3_Sound'
, 'S4_Sound', 'S5_CO2', 'S5_CO2_Slope']].values.astype(float)
y = df.loc[:,'Room_Occupancy_Count'].values.astype(int)

#decision 0 = data: 0
#decision 1, 2 & 3 = data: 1
y = [0 if i==0 else 1 for i in y]

#scaling data
X = scale.fit_transform(X)
```

Model fitting

For the modelling purpose, data set was divided to two portions as a test data set of 30% and a train dataset of 70%. And the following library was used to do the task.

```
from sklearn.model_selection import train_test_split
```

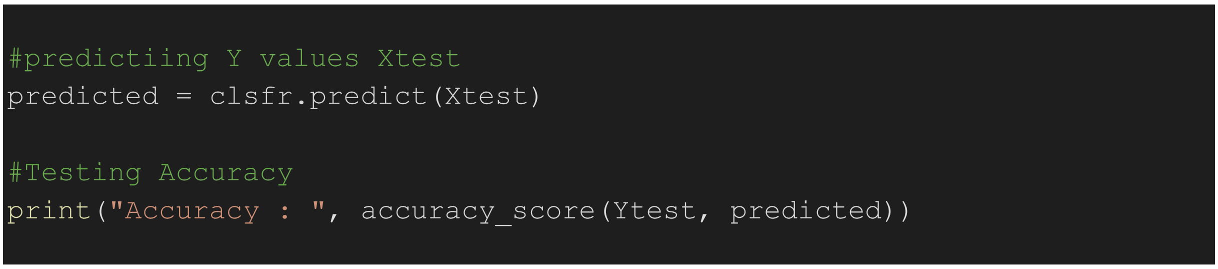
And, logistic regression model was used using the following python libraries.

```
from sklearn import linear_model
```

```
#Applying Logistic Regression
clsfr = linear_model.LogisticRegression(random_state = 0)
clsfr.fit(Xtrain, Ytrain)
```

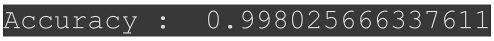
Evaluating model

Predicting values for the room occupancy for the test data and testing accuracy comparing predicted and actual test data. Accuracy is 0.998 which represents a strong succesful model.



```
#predictiing Y values Xtest
predicted = clsfr.predict(Xtest)

#Testing Accuracy
print("Accuracy : ", accuracy_score(Ytest, predicted))
```



```
Accuracy :  0.998025666337611
```

Checkout the code segment Link to Google Colabs