Statistics 652 - Quiz

Sajith Gowthaman

February 19, 2020

In [1]:

```
import pandas as pd
import numpy as np

# Import CSV mtcars
data = pd.read_csv('https://gist.githubusercontent.com/ZeccaLehn
/4e06d2575eb9589dbe8c365d61cb056c/raw/64f1660f38ef523b2a1a13be77
b002b98665cdfe/mtcars.csv')
# Edit element of column header
data.rename(columns={'Unnamed: 0':'brand'}, inplace=True)
```

In [2]:

```
data.head()
```

Out[2]:

	brand	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear
0	Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4
1	Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4
2	Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4
3	Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3
4	Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.02	0	0	3

In [3]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32 entries, 0 to 31
Data columns (total 12 columns):
brand    32 non-null object
mpg      32 non-null float64
cyl      32 non-null int64
disp     32 non-null float64
hp       32 non-null int64
drat     32 non-null float64
wt       32 non-null float64
qsec     32 non-null float64
vs       32 non-null int64
am       32 non-null int64
gear     32 non-null int64
carb     32 non-null int64
dtypes: float64(5), int64(6), object(1)
memory usage: 3.1+ KB
```

In [4]:

```
data['brand'] = pd.get_dummies(data['brand'])
data['brand'].unique()
```

Out[4]:

```
array([0, 1], dtype=uint64)
```

3) For the mtcars dataset give the code for creating a training data set of 27 examples and a test data set of 5. examples.

In [5]:

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

In [6]:

```
X_train, X_test, y_train, y_test = train_test_split(data.drop('mpg', 1), data['mpg'], train_size =0.87, random_state=40)
```

In [7]:

```
print("The number of observations in training set is {}".format(X_train.shape[0]))
print("The number of observations in test set is {}".format(X_test.shape[0]))
```

```
The number of observations in training set is 27
The number of observations in test set is 5
```

4) Build a Linear Regression model using the training data set you have created and use it to predict the values of the test dataset. Type the final model using LaTeX using βi and xi.

In [8]:

```
from sklearn import linear_model
lm = linear_model.LinearRegression()

# fit method estimates the coefficients using OLS
lm.fit(X_train,y_train)

# Inspect the results.
print('\nCoefficients: \n', lm.coef_)
print('\nIntercept: \n', lm.intercept_)
```

```
Coefficients:
 [-4.03935505  1.48004624  0.01410524 -0.01501797  1
.93098623 -4.48594043
  1.00726231  0.87391927  1.75320314  3.18560552 -1.
15229589]

Intercept:
 -9.987575784554377
```

In [9]:

```
from sklearn.metrics import mean_absolute_error
import statsmodels.api as sm
from statsmodels.tools.eval_measures import mse, rmse
from sklearn.metrics import classification_report

y_preds = lm.predict(X_test)

print("Mean absolute error of the prediction is: {}".format(mean
_absolute_error(y_test, y_preds)))
print("Mean squared error of the prediction is: {}".format(mse(y
_test, y_preds)))
print("Root mean squared error of the prediction is: {}".format(
rmse(y_test, y_preds)))
print("Mean absolute percentage error of the prediction is: {}".
format(np.mean(np.abs((y_test - y_preds) / y_test)) * 100))
```

```
Mean absolute error of the prediction is: 4.55732963
1379973
Mean squared error of the prediction is: 27.04830398
3657327
Root mean squared error of the prediction is: 5.2007
983986747
Mean absolute percentage error of the prediction is:
25.21409052286393
```

In [10]:

```
X_train = sm.add_constant(X_train)

# We fit an OLS model using statsmodels
results = sm.OLS(y_train, X_train).fit()

# We print the summary results
print(results.summary())
```

```
                            OLS Regression Results
==============================================================
==========================
Dep. Variable:                        mpg   R-squared:
0.938
Model:                                OLS   Adj. R-squar
ed:                   0.892
```

```
Method:                 Least Squares   F-statistic:
20.60
Date:                Wed, 26 Feb 2020   Prob (F-stat
istic):           4.48e-07
Time:                        21:40:42   Log-Likeliho
od:                -48.067
No. Observations:                  27   AIC:
120.1
Df Residuals:                      15   BIC:
135.7
Df Model:                          11
Covariance Type:            nonrobust
==============================================
==========================
                 coef    std err          t      P>|
t|      [0.025      0.975]
----------------------------------------------
--------------------------
const         -9.9876     17.232     -0.580      0.5
71     -46.716      26.741
brand         -4.0394      2.171     -1.861      0.0
83      -8.667       0.588
cyl            1.4800      0.963      1.537      0.1
45      -0.572       3.532
disp           0.0141      0.014      0.996      0.3
35      -0.016       0.044
hp            -0.0150      0.017     -0.865      0.4
01      -0.052       0.022
drat           1.9310      1.416      1.363      0.1
93      -1.088       4.950
wt            -4.4859      1.594     -2.814      0.0
13      -7.884      -1.088
qsec           1.0073      0.558      1.805      0.0
91      -0.182       2.197
vs             0.8739      1.621      0.539      0.5
98      -2.580       4.328
am             1.7532      1.646      1.065      0.3
04      -1.755       5.261
gear           3.1856      1.386      2.298      0.0
36       0.231       6.140
carb          -1.1523      0.796     -1.447      0.1
69      -2.850       0.545
==============================================
==========================
Omnibus:                        3.779   Durbin-Watso
```

```
n:                          2.332
Prob(Omnibus):                  0.151   Jarque-Bera
(JB):                       2.377
Skew:                           0.699   Prob(JB):
0.305
Kurtosis:                       3.401   Cond. No.
1.39e+04
================================================
==========================

Warnings:
[1] Standard Errors assume that the covariance matri
x of the errors is correctly specified.
[2] The condition number is large, 1.39e+04. This mi
ght indicate that there are
strong multicollinearity or other numerical problems
.

/Users/sajithgowthaman/opt/anaconda3/lib/python3.7/s
ite-packages/numpy/core/fromnumeric.py:2495: FutureW
arning: Method .ptp is deprecated and will be remove
d in a future version. Use numpy.ptp instead.
  return ptp(axis=axis, out=out, **kwargs)
```

$$y = \beta_0 + \beta_1 x_1 + \epsilon$$

mpg = -3.460brand + 0.07237854cyl + 0.91153236disp + 0.02582739hp -0.02222715drat + 1.65389797wt -5.28828152qsec + 0.88793436vs + 1.26223391am + 1.69942195gear + 2.56575396carb

5) Give the formula for the Logistic Regression model.

The Logistic Regression:

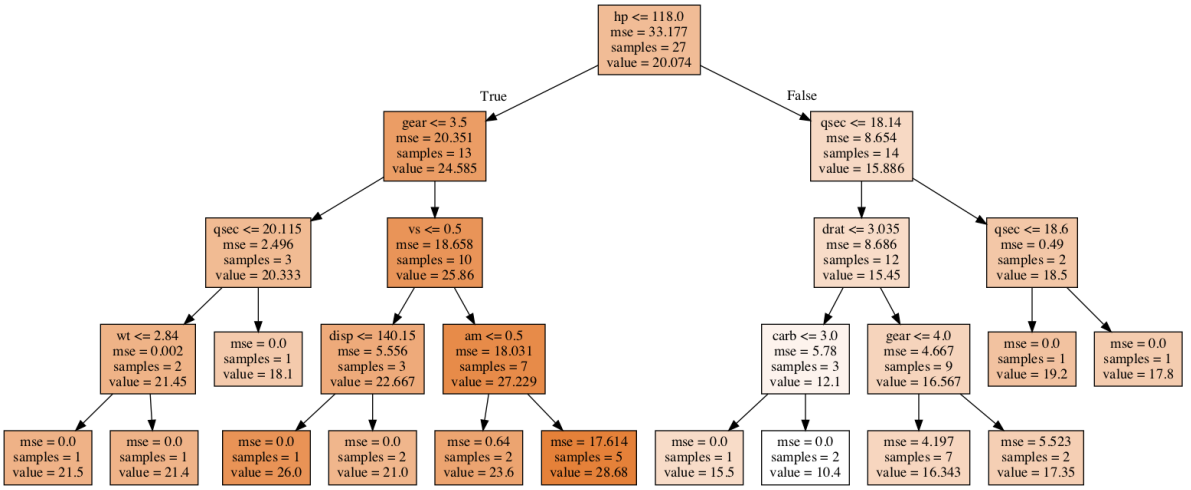
$$g(z) = \frac{1}{1 + e^{-z}}$$

7) Build a Regression Tree model using the training data set you have created and use it to predict the values of the test data set.

In [11]:

```
# This is the model we'll be using.
from sklearn import tree

# A convenience for displaying visualizations.
from IPython.display import Image

# Packages for rendering our tree.
import pydotplus
import graphviz

# Initialize and train our tree.
decision_tree = tree.DecisionTreeRegressor(
    max_features=1,
    max_depth=4,
    random_state = 1337
)
decision_tree.fit(X_train, y_train)

# Render our tree.
dot_data = tree.export_graphviz(
    decision_tree, out_file=None,
    feature_names=X_train.columns,
    class_names=['high mpg', 'low mpg'],
    filled=True
)
graph = pydotplus.graph_from_dot_data(dot_data)
Image(graph.create_png())
```

Out[11]:



1)Explain the terms Statistical Learning and Machine Learning.

Statistical learning is a way of classifying data into supervised and unsupervised. This involves predicting a function with the help of analyzing the given data. They can be used to study the data and come up with a decision based on the data that we study. It helps to study the pattern of the data which can later be classified into machine learning categories.

Machine Learning is the study of algorithms and systems that improve the knowledge and performance of the respective machine. This process involves creating machines or computer systems that can learn how to perform tasks. Making the system learn by feeding input that predict the dependent variables by comparing it with the independent variables.

3) Explain the terms Supervised Learning and Unsupervised Learning.

Supervised learning is the process of feeding the target, features variables and is asked to predict learn what the pattern in the dataset is. They are classified into classification tasks and regression tasks. Eg of a classification task is predicting the Rating, and regressing task would be predicting or learning the pattern of a houseprice.

Unsupervised learning: Unlike supervised learning, here the features that has no label and can be used to discover pattern or predict the variables that can be fed in to a supervised learning model if needed.

These are are of four types: Clustering (finding groups that come from data), Association (learning the rules from the data), Neural Networks (strengthening the neuron and machine learning process), and Anomaly detection (discovering anomalies from data.)

6) Explain what type of Supervised Learning task Linear Regression is used for. Explain what type of Supervised Learning tast Logistic Regression is used for.

Linear regression is used for predicting models that come under regression models. Regression problems have a continuous outcome variable. It has a linear relationship between the input variables (x) and the single output variable (y). y can be calculated from a linear combination of the input variables (x). It can be performed with OLS (ordinary least square) technique.

Logistic Regression is a classifier model. It usually works with a binary variable as target however, it can be solved by using multi-class to predict the outcome.

8)Explain what an Ensemble is used for. Give the name of a Machine Learning algorithm that creates an ensemble. What are the benefits of using an ensemble method?

These are models that are built based on another model. The machine learning model that used in ensemble model is Random Forest. Here the submodes are decision tree. It improves the model by combining multiple models to predict better results.