ADS-500B Assignment 6.1 Machine Learning in R and Python

Created by: Gabriel Duffy

1.) Regression models and Findings of Airline Costs dataset in Python

Model #1 and import of 'statsmodels.api' libraries'

```python
# importing pandas and statsmodels.api libraries
import pandas as pd
import statsmodels.api as sm

# Load the dataset
airline_costs = pd.read_csv("C:/Users/gabed/.ipython/airline_costs.csv")

# relevant columns for the regression analysis
X = airline_costs[['FlightLength', 'DailyFlightTime']]

# Constant term
X = sm.add_constant(X)

# Dependent variable
y = airline_costs['CustomersServed']

# Fit model
model = sm.OLS(y, X).fit()

# Print summary
print(model.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:        CustomersServed   R-squared:                       0.622
Model:                            OLS   Adj. R-squared:                  0.595
Method:                 Least Squares   F-statistic:                     23.00
Date:                Sat, 06 Apr 2024   Prob (F-statistic):           1.23e-06
Time:                        11:28:48   Log-Likelihood:                -330.06
No. Observations:                  31   AIC:                             666.1
Df Residuals:                      28   BIC:                             670.4
Df Model:                           2
Covariance Type:            nonrobust
===================================================================================
                      coef    std err          t      P>|t|      [0.025      0.975]
-----------------------------------------------------------------------------------
const           -7792.0706   8396.998     -0.928      0.361    -2.5e+04    9408.401
FlightLength      183.2956     30.414      6.027      0.000     120.995     245.596
DailyFlightTime  -213.3340   1436.955     -0.148      0.883   -3156.803    2730.134
==============================================================================
Omnibus:                        2.469   Durbin-Watson:                   1.894
Prob(Omnibus):                  0.291   Jarque-Bera (JB):                1.851
Skew:                           0.598   Prob(JB):                        0.396
Kurtosis:                       2.931   Cond. No.                         653.
==============================================================================
```

Model # 2:

```python
# second linear regression model to predict the total assets of an airline

import pandas as pd
import statsmodels.api as sm

# Load the dataset
airline_costs = pd.read_csv("C:/Users/gabed/.ipython/airline_costs.csv")

# relevant columns for the regression analysis
X = airline_costs['CustomersServed']  # Independent variable
y = airline_costs['TotalAssets']  # Dependent variable

# constant term to the independent variable
X = sm.add_constant(X)

# Fit model
model = sm.OLS(y, X).fit()

# Print summary
print(model.summary())
```

```
Dep. Variable:            TotalAssets   R-squared:                       0.819
Model:                            OLS   Adj. R-squared:                  0.812
Method:                 Least Squares   F-statistic:                     130.9
Date:                Sat, 06 Apr 2024   Prob (F-statistic):           2.85e-12
Time:                        11:30:15   Log-Likelihood:                -202.95
No. Observations:                  31   AIC:                             409.9
Df Residuals:                      29   BIC:                             412.8
Df Model:                           1
Covariance Type:            nonrobust
===================================================================================
                      coef    std err          t      P>|t|      [0.025      0.975]
-----------------------------------------------------------------------------------
const             -98.5080     41.633     -2.366      0.025    -183.656     -13.360
CustomersServed     0.0217      0.002     11.442      0.000       0.018       0.026
==============================================================================
Omnibus:                       18.279   Durbin-Watson:                   1.745
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               24.501
Skew:                          -1.486   Prob(JB):                     4.78e-06
Kurtosis:                       6.183   Cond. No.                     2.92e+04
==============================================================================
```

Insight about the data from the two regression models:

```
# Insights from the data
# Regression model 1 Dependent Variable Customers Served, Independent Variable Flight Length and Daily Flight Time Findings:
The regression results for regression model 1 shows, flight length has a significant positive effect on the number of
customers served, but daily flight time per plane does not have a significant effect on customers served.The coefficient
for flight length is 183.2956, with a standard error of 30.414. It has a statistically significant
p-value indicating that flight length has a significant positive effect on the number of customers served.

# Regression model 2 Dependent Variable Total Assets, Independent Variable Customers Served Findings:
The regression results indicate a significant positive relationship between the number of customers served and the total
assets of the airline.
The coefficient of determination is 0.819, indicating that approximately 81.9% of the variability in the total assets of
the airline can be explained by the number of customers served.
```

2.) Use agglomerative clustering and divisive clustering on this dataset to find out which players have similar performance in the same season. Visualize the clusters using dendrograms for both types of clustering models.

```
# Step 1: Load the dataset
file_path <- "C:/Users/gabed/OneDrive/Documents/R/lpga2008.csv"
data <- read.csv(file_path)

# Step 2:handle any missing values
# Check for missing values
print(colSums(is.na(data)))
               Golfer        AvgDriveDistance         FairwayPercent RegulatedGreensPercent
                    0                       0                      0                      0
              AvgPutts            SandAttempts              SandSaves          TotalWinnings
                    0                       0                      0                      0
                  Log              TotalRounds                     Id
                    0                       0                      0

# Step 3: Scale the data
scaled_data <- scale(data[, -c(1, length(data))])  # Exclude 'Golfer' and 'Id' columns for cluste
ing

# Step 4: agglomerative clustering
agglomerative_result <- agnes(scaled_data, method = "average")

# Step 5: divisive clustering
divisive_result <- diana(scaled_data)

# Step 6: Visualize the clusters
# Plot dendrograms for agglomerative and divisive clustering
par(mfrow=c(1,2))
plot(agglomerative_result, main="Agglomerative Clustering")
plot(divisive_result, main="Divisive Clustering")
```