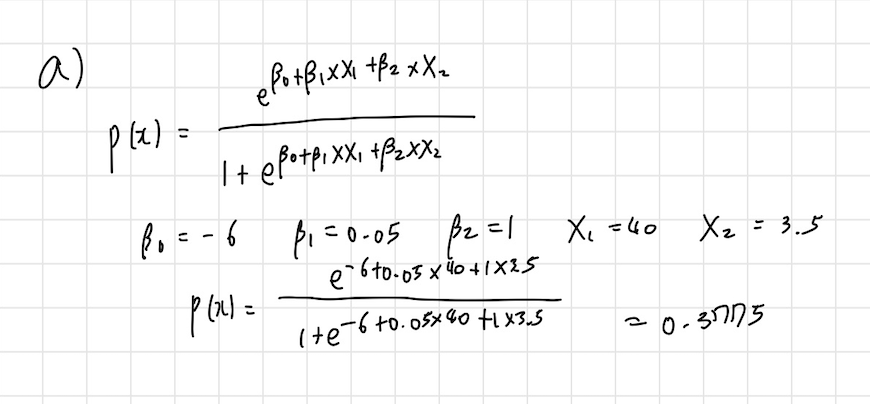
Yurim Park hw3

1.

a) 37.75%



b) 50hours

A math equations on a grid paper

Description automatically generated

c) Specifically, β0=−6 implies that when both X1(hours studied) and X2 (undergrad GPA) are zero, the log odds of receiving an A is -6.

2.

a)

**Mean:**

Year 1999.993019

Lag1 0.203541

Lag2 0.203747

Lag3 0.207269

Lag4 0.205614

Lag5 0.206440

Volume 1.011219

Today 0.200951

dtype: float64

**Standard Deviation:**

Year 3.166690

Lag1 2.289741

Lag2 2.289738

Lag3 2.291947

Lag4 2.292765

Lag5 2.292686

Volume 0.506743

Today 2.290949

dtype: float64

**Median:**

Year 2000.0000

Lag1 0.3120

Lag2 0.3120

Lag3 0.3120

Lag4 0.3120

Lag5 0.3120

Volume 0.9818

Today 0.3120

dtype: float64

**Minimum Values:**

Year 1995.000000

Lag1 -11.050000

Lag2 -11.050000

Lag3 -11.050000

Lag4 -11.050000

Lag5 -11.050000

Volume 0.241088

Today -11.050000

dtype: float64

**Maximum Values:**

Year 2005.00000

Lag1 7.78000

Lag2 7.78000

Lag3 7.78000

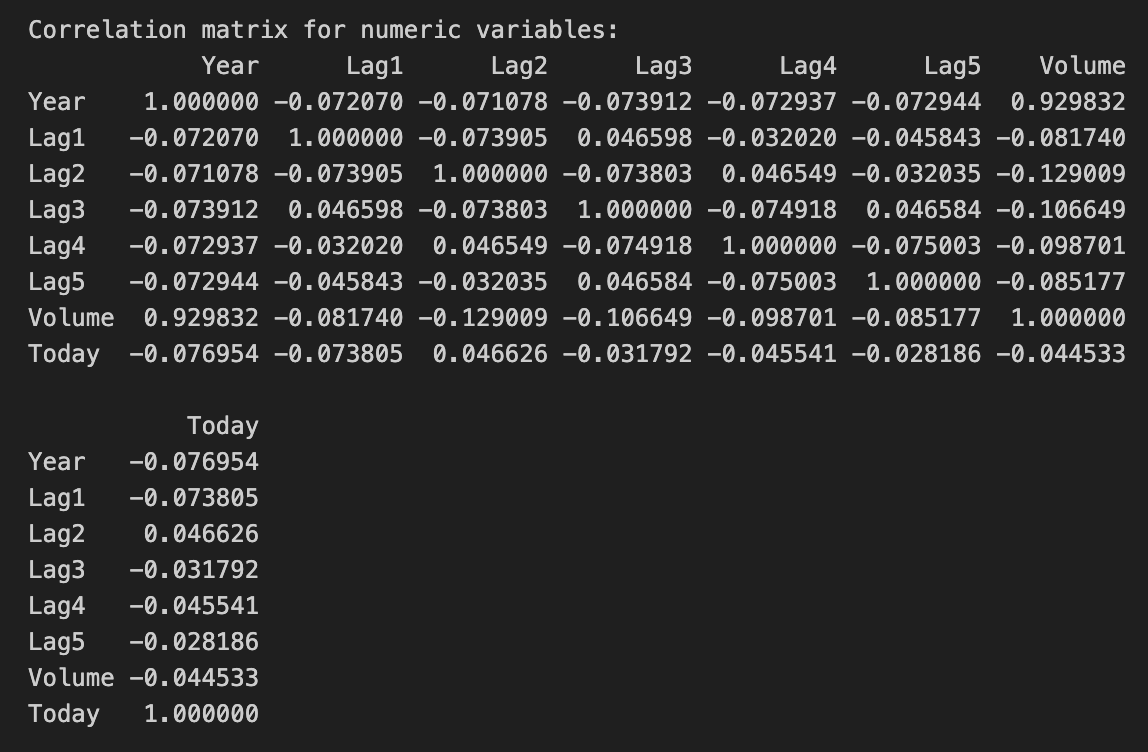
Lag4 7.78000

Lag5 7.78000

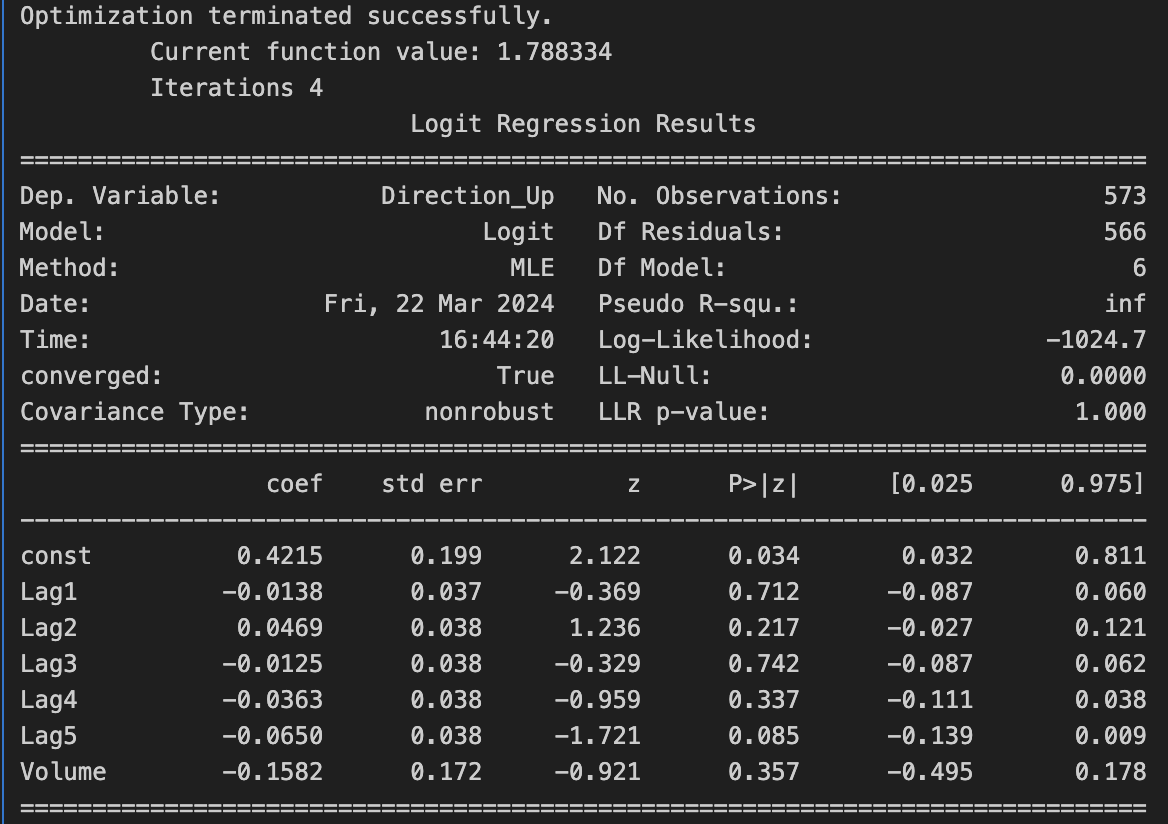
Volume 2.48811

Today 7.78000

dtype: float64



b)



None of the lag variables (Lag1, Lag2, Lag3, Lag4, Lag5) or the Volume variable appear to be statistically significant, as their p-values are all greater than 0.05. Therefore, based on the logistic regression results, we cannot conclude that any of these predictors are statistically significant in predicting the direction of the response variable.

c)

Confusion Matrix:

[[ 27 225]

[ 35 286]]

Accuracy: 0.5462

False Positive Rate: 89.2857%

False Negative Rate: 10.9034%

d)

Confusion Matrix with Lag2 Predictor:

[[134 8]

[108 11]]

Overall, Fraction of Correct Predictions with Lag2 Predictor: 0.5555555555555556

Percent of False Positives with Lag2 Predictor: 42.10526315789473

Percent of False Negatives with Lag2 Predictor: 44.62809917355372

e)

Confusion Matrix:

[[134 8]

[108 11]]

Overall, Fraction of Correct Predictions: 0.5556

Percent of False Positives: 41.38%

Percent of False Negatives: 3.07%

f)

Confusion Matrix:

[[83 59]

[68 51]]

Overall, Fraction of Correct Predictions: 0.5134

Percent of False Positives: 26.05%

Percent of False Negatives: 22.61%

g)

3.

a)

A diagram with colored lines

Description automatically generated

b)

A graph with blue and orange lines

Description automatically generated

c)

Observations 1 and 2 are in Cluster A and 3 and 4 in Cluster B.

d)

Observations 1, 2 and 3 are in Cluster A and 4 in Cluster B.

e)

A diagram with colored squares

Description automatically generated with medium confidence

4.

a)

A graph with blue dots

Description automatically generated

b)

A graph with green and red dots

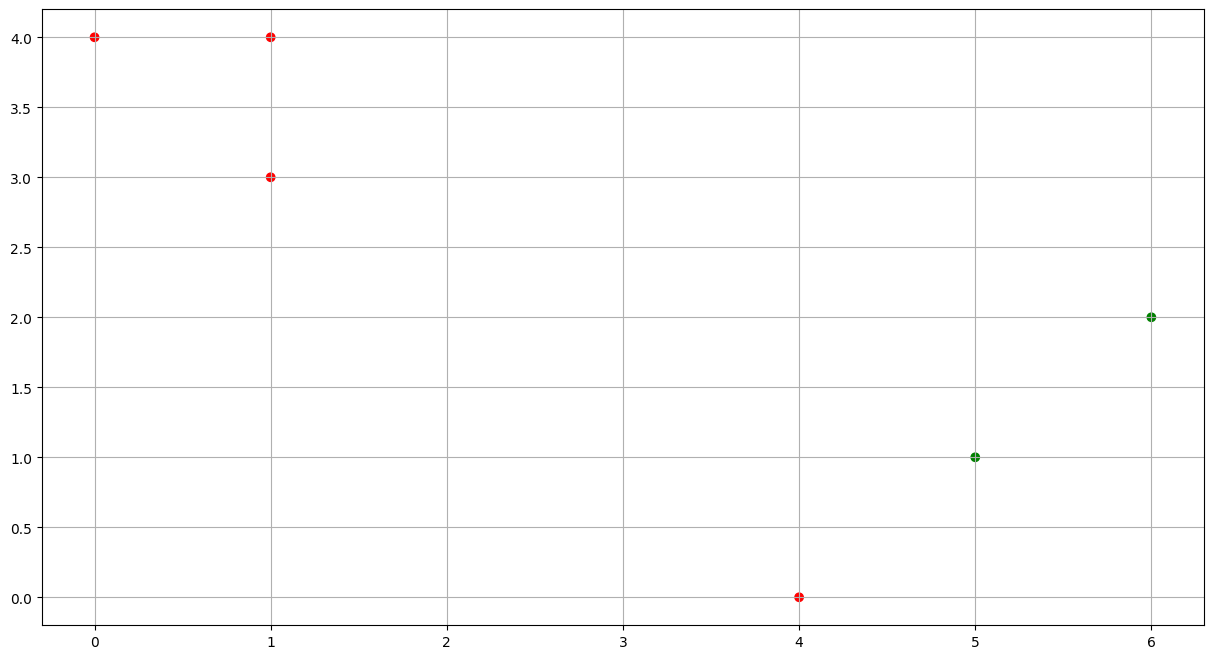
Description automatically generated

c)

Centriod for Clutser 0 is: 3.0, 2.5

Centriod for Clutser 1 is: 2.75, 2.25

d)



e)

A graph with numbers and points

Description automatically generated

f)

A graph with numbers and points

Description automatically generated

5.

a)

import pandas as pd

from scipy.cluster.hierarchy import dendrogram, linkage

import matplotlib.pyplot as plt

df = pd.read\_csv('CEV2021(1).csv')

states = df['State'].

data = df.drop(['State'], axis=1)

Z = linkage(data, method='complete', metric='euclidean')

plt.figure(figsize=(10, 8))

dendrogram(Z, labels=states, above\_threshold\_color='y', orientation='top')

plt.title('Hierarchical Clustering Dendrogram (Complete Linkage)')

plt.xlabel('State')

plt.ylabel('Euclidean Distance')

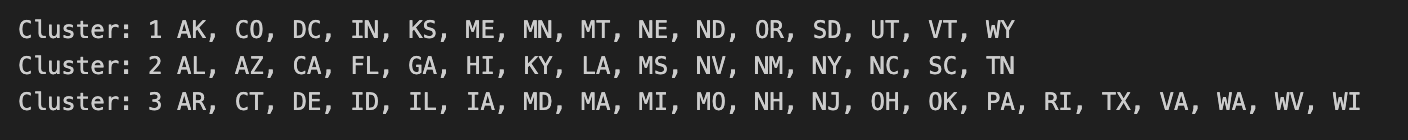
plt.xticks(rotation=90)

plt.show()

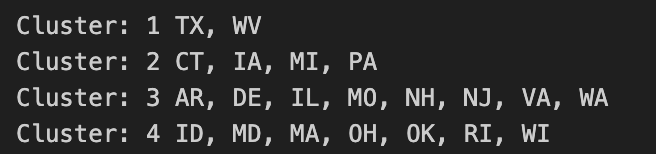
A diagram of a clustering structure

Description automatically generated

b)



c)



d)

A black background with white text

Description automatically generated

e)

scaler = StandardScaler()

scaled\_data = scaler.fit\_transform(data)

Z = linkage(scaled\_data, method='complete', metric='euclidean')

# Plot the dendrogram

plt.figure(figsize=(10, 7)) # Adjust the size as needed

dendrogram(Z, above\_threshold\_color='y', orientation='top')

plt.title('Hierarchical Clustering Dendrogram (Complete Linkage)')

plt.xlabel('State')

plt.ylabel('Distance')

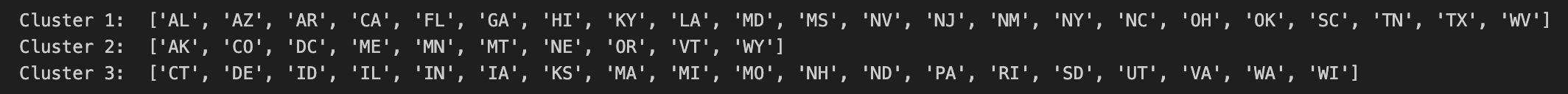
plt.xticks(rotation=90) # Rotate state names for better readability

plt.show()

A diagram of a cluster of buildings

Description automatically generated

f)



g)

from sklearn.cluster import KMeans

# Perform K-means clustering with K=3 on the scaled data

kmeans = KMeans(n\_clusters=3, random\_state=42)

kmeans.fit(scaled\_data)

# Get cluster labels for each state

cluster\_labels = kmeans.labels\_

# Mapping clusters to states

cluster\_assignment = {state: cluster for state, cluster in zip(states, cluster\_labels)}

# Initialize dictionaries to hold lists of states for each cluster

clusters\_states = {i: [] for i in range(3)}

# Populate the dictionaries with states grouped by their cluster

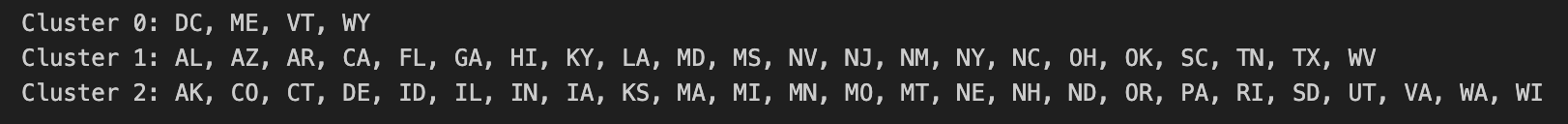
for state, cluster in cluster\_assignment.items():

clusters\_states[cluster].append(state)

# Print the states in each cluster

for cluster, states in clusters\_states.items():

print(f"Cluster {cluster}: {', '.join(states)}")



h)

A screen shot of a graph

Description automatically generated