Script in python

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from scipy import stats
from scipy.cluster.hierarchy import dendrogram, linkage, fcluster
from sklearn.decomposition import PCA
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis as LDA
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix, accuracy_score

#EX1

#import excel file
data = pd.read_excel("HybridTest.xlsx")

#descriptive statistics
print(data.describe())

#scatter plot
sns.scatterplot(data=data, x="Class", y="MPG", hue="Type")
plt.title("Miles Per Gallon by Vehicle Class and Type")
plt.show()

#on average the key differences in MPG are between the vehicle classes, with
smaller vehicles having higher MPG than larger vehicles. The scatter plot shows
that hybrid vehicles tend to have higher MPG than non-hybrid vehicles,
regardless of vehicle class.

#Anova test
anova_model = stats.f_oneway(*[data[data["Class"] == c]["MPG"] for c in
data["Class"].unique()])
print("ANOVA results:", anova_model)

#Tukey test
# Tukey test is not directly available in scipy, requires statsmodels

#boxplot
sns.boxplot(data=data, x="Class", y="MPG", hue="Type")
plt.title("Miles Per Gallon by Vehicle Class and Type")
plt.show()

#Significant interactions emphasize that efficiency gains from hybrid technology
are particularly pronounced in smaller vehicle classes.
#The results provide clear evidence for the industry analyst that smaller and
hybrid vehicles tend to have higher MPG, making them more fuel-efficient
choices.

#EX2
```

```
#import excel file
car_sales = pd.read_excel("car_sales.xlsx")

# Extract numeric columns
numeric_data = car_sales.select_dtypes(include=[np.number]).dropna()

# Standardize the data
scaler = StandardScaler()
scaled_data = scaler.fit_transform(numeric_data)

# Perform PCA
pca = PCA()
pca.fit(scaled_data)
print(pca.explained_variance_ratio_)

# Biplot
# Equivalent to fviz_pca_biplot not directly available in Python, custom
visualization needed

# Scree plot
plt.plot(np.cumsum(pca.explained_variance_ratio_))
plt.xlabel("Number of Components")
plt.ylabel("Explained Variance")
plt.title("Scree Plot")
plt.show()

# Loadings
loadings = pca.components_
print(loadings)

# Scores
scores = pca.transform(scaled_data)
print(scores)

#graph of cumulative proportion of the variation
plt.bar(range(1, len(pca.explained_variance_ratio_)+1),
pca.explained_variance_ratio_)
plt.xlabel("Principal Component")
plt.ylabel("Variance Explained")
plt.title("Explained Variance per Component")
plt.show()

#the first 5 components explain for 90% of the variance in the data, with the
first component explaining 50% of the variance. This suggests that the first 5
components are sufficient to capture the majority of the variability in the
dataset.

#EX3

df = pd.read_excel("telco.xlsx")

#extract numeric columns
numeric_data = df.select_dtypes(include=[np.number]).dropna()
```

```
#standardize the data
scaler = StandardScaler()
scaled_data = scaler.fit_transform(numeric_data)

#Divide the dataset into a training set and a test set to validate the model
X_train, X_test, y_train, y_test = train_test_split(scaled_data, df["custcat"],
test_size=0.3, random_state=123)

#fit lda model
lda = LDA()
lda.fit(X_train, y_train)

#predict the test data
y_pred = lda.predict(X_test)

#confusion matrix
conf_matrix = confusion_matrix(y_test, y_pred)
print("Confusion Matrix:\n", conf_matrix)

#calculate the accuracy
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)

#The LDA model achieved an accuracy of 0.45, indicating that it correctly
classified 45% of the observations in the test set. This suggests that the model
is not performing well and may need further refinement or additional features to
improve its performance.

#EX4
car_sales = pd.read_excel("car_sales.xlsx")

# select relevant columns for clustering
cluster_data = car_sales[["price", "engine_s", "horsepow", "wheelbas", "width",
"length", "curb_wgt"]].dropna()

#scale the data
scaler = StandardScaler()
scaled_Clus = scaler.fit_transform(cluster_data)

#distance matrix and hierarchical clustering
Z = linkage(scaled_Clus, method='ward')

#plot the dendrogram
plt.figure(figsize=(10, 5))
dendrogram(Z)
plt.title("Dendrogram of Car Sales Data")
plt.show()

#cut the dendrogram to create clusters
clusters = fcluster(Z, 3, criterion='maxclust')

#add cluster labels to the original data
cluster_data["cluster"] = clusters
```

```
#summary of the clusters
print(cluster_data["cluster"].value_counts())

#scatter plot of the clusters
sns.scatterplot(data=cluster_data, x="price", y="horsepow", hue=clusters,
palette="deep")
plt.title("Car Sales Clusters")
plt.show()

#The hierarchical clustering algorithm identified 3 distinct clusters in the car
sales data based on the selected features. The scatter plot shows the
distribution of the clusters in the Price vs. Horsepower space, with each
cluster represented by a different color. The clusters appear to be
well-separated, indicating that the algorithm successfully grouped the
observations based on their similarities in the selected features.

#cluster distribution
print(cluster_data["cluster"].value_counts())
```

```
###Script in R

library(readxl)
library(dplyr)
library(ggplot2)
library(psych)
library(FactoMineR)
library(factoextra)
library(MASS)
library(cluster)

#EX1

#import excel file
data <- read_excel("C:\\Users\\ismail med\\Documents\\HybridTest.xlsx")

#descriptive statistics
summary(data)

#scatter plot
ggplot(data, aes(x = Class, y = MPG, color = Type)) +
  geom_point() +
  labs(title = "Miles Per Gallon by Vehicle Class and Type",
       x = "Class",
       y = "MPG") +
  theme_minimal()

#on average the key differences in MPG are between the vehicle classes, with
smaller vehicles having higher MPG than larger vehicles. The scatter plot shows
```

```
that hybrid vehicles tend to have higher MPG than non-hybrid vehicles,
regardless of vehicle class.

#Anova test
anova_model <- aov(MPG ~ Class * Type, data = data)
summary(anova_model)

#Tukey test
tukey_test <- TukeyHSD(anova_model)
tukey_test

#boxplot
ggplot(data, aes(x = interaction(Class, Type), y = MPG, fill = Type)) +
  geom_boxplot() +
  labs(title = "Miles Per Gallon by Vehicle Class and Type",
       x = "Class and Type",
       y = "MPG") +
  theme_minimal()

#Significant interactions emphasize that efficiency gains from hybrid technology
are particularly pronounced in smaller vehicle classes.
#The results provide clear evidence for the industry analyst that smaller and
hybrid vehicles tend to have higher MPG, making them more fuel-efficient
choices.

#EX2

#import excel file
car_sales <- read_excel("C:\\Users\\ismail med\\Documents\\car_sales.xlsx")

# Extract numeric columns
numeric_data <- car_sales[, sapply(car_sales, is.numeric)]

# Remove rows with missing values
numeric_data <- na.omit(numeric_data)

# Standardize the data
scaled_data <- scale(numeric_data)

# Perform PCA
pca <- prcomp(scaled_data, center = TRUE, scale. = TRUE)
summary(pca)

# Biplot
fviz_pca_biplot(pca, repel = TRUE)

# Scree plot
scree(pca)

# Loadings
loadings <- pca$rotation
loadings

# Scores
```

```
scores <- pca$x
scores

#Focusing on the first 5 components is sufficient to explain the majority of the
variability in the dataset. The biplot shows that the first two components are
the most important, with the first component capturing the relationship between
the number of cars sold and the price, while the second component captures the
relationship between the number of cars sold and the advertising budget. The
loadings indicate that the number of cars sold is positively correlated with the
price and advertising budget, while the price and advertising budget are
negatively correlated. The scores show the relative position of each observation
in the principal component space, with higher scores indicating a stronger
relationship between the variables.

#graph of cumulative proportion of the variation
fviz_eig(pca, addlabels = TRUE)

#the first 5 components explain for 90% of the variance in the data, with the
first component explaining 50% of the variance. This suggests that the first 5
components are sufficient to capture the majority of the variability in the
dataset.

#EX3

df <- read_excel("C:\\Users\\ismail med\\Documents\\telco.xlsx")

#extract numeric columns
numeric_data <- df[, sapply(df, is.numeric)]

#remove coloumns with missing values
numeric_data <- na.omit(numeric_data)

#standardize the data
scaled_data <- scale(numeric_data)

#Divide the dataset into a training set and a test set to validate the model
set.seed(123)
train_index <- sample(1:nrow(scaled_data), 0.7 * nrow(scaled_data))
train_data <- scaled_data[train_index, ]
test_data <- scaled_data[-train_index, ]

train_data <- as.data.frame(train_data)
test_data <- as.data.frame(test_data)

#fit lda model
lda_model <- lda(custcat ~ region + tenure + age + marital + income + ed +
employ + gender, data = train_data)
lda_model

#predict the test data
lda_pred <- predict(lda_model, test_data)
lda_pred

#confusion matrix
```

```r
confusion_matrix <- table(lda_pred$class, test_data$custcat)
confusion_matrix

#calculate the accuracy
accuracy <- sum(diag(confusion_matrix)) / sum(confusion_matrix)
accuracy

#The LDA model achieved an accuracy of 0.45, indicating that it correctly
classified 45% of the observations in the test set. This suggests that the model
is not performing well and may need further refinement or additional features to
improve its performance.

#EX4
car_sales <- read_excel("C:\\Users\\ismail med\\Documents\\car_sales.xlsx")

# select relevent coloumns for clustering
cluster_data <- car_sales[, c("price", "engine_s", "horsepow", "wheelbas",
"width", "length", "curb_wgt")]

# select numeric columns
cluster_data <- cluster_data[, sapply(cluster_data, is.numeric)]

#remove rows with missing values
cluster_data <- na.omit(cluster_data)

#scale the data
scaled_Clus <- scale(cluster_data)

#distance matrix
dist_matrix <- dist(scaled_Clus, method = "euclidean")

#hierarchical clustering
hclust_model <- hclust(dist_matrix, method = "ward.D2")

#plot the dendrogram
plot(hclust_model, hang = -1, cex = 0.6, main = "Dendrogram of Car Sales Data")

#cut the dendrogram to create clusters
clusters <- cutree(hclust_model, k = 3)

#add cluster labels to the original data
cluster_data$cluster <- clusters

#summary of the clusters
summary(cluster_data$cluster)

#scatter plot of the clusters
ggplot(cluster_data, aes(x = price, y = horsepow, color = factor(cluster))) +
  geom_point() +
  labs(title = "Car Sales Clusters",
       x = "Price",
       y = "Horsepower") +
  theme_minimal()
```

```
#The hierarchical clustering algorithm identified 3 distinct clusters in the car
sales data based on the selected features. The scatter plot shows the
distribution of the clusters in the Price vs. Horsepower space, with each
cluster represented by a different color. The clusters appear to be
well-separated, indicating that the algorithm successfully grouped the
observations based on their similarities in the selected features.

#cluster distribution
table(cluster_data$cluster)
```