import os

import numpy as np

import pandas as pd

from scipy.stats import skew, kurtosis

from sklearn.datasets import load\_diabetes

from sklearn.linear\_model import LinearRegression, LogisticRegression

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import StandardScaler

from sklearn.metrics import r2\_score, mean\_squared\_error, accuracy\_score, roc\_auc\_score, confusion\_matrix, classification\_report

import statsmodels.api as sm

uci = load\_diabetes()

df\_uci = pd.DataFrame(uci.data, columns=uci.feature\_names)

df\_uci["target"] = uci.target

pima\_path = "diabetes.csv"

if not os.path.exists(pima\_path):

raise FileNotFoundError("Please place 'diabetes.csv' in the same folder as this script.")

df\_pima = pd.read\_csv(pima\_path)

if "Outcome" not in df\_pima.columns:

df\_pima.rename(columns={df\_pima.columns[-1]: "Outcome"}, inplace=True)

def univariate\_analysis(df, name):

print(f"\n===== Univariate Analysis: {name} =====")

results = []

for col in df.select\_dtypes(include=[np.number]).columns:

data = df[col].dropna()

results.append({

"Feature": col,

"Count": len(data),

"Mean": data.mean(),

"Median": data.median(),

"Mode": data.mode().iloc[0] if not data.mode().empty else np.nan,

"Variance": data.var(),

"Std Dev": data.std(),

"Skewness": skew(data),

"Kurtosis": kurtosis(data)

})

return pd.DataFrame(results)

uni\_uci = univariate\_analysis(df\_uci, "UCI Diabetes")

uni\_pima = univariate\_analysis(df\_pima, "Pima Indians Diabetes")

print("\n===== Bivariate Analysis: UCI Diabetes (Linear Regression) =====")

X = df\_uci[["bmi"]] # using 'bmi' as strongest predictor

y = df\_uci["target"]

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

lin\_reg = LinearRegression().fit(X\_train, y\_train)

y\_pred = lin\_reg.predict(X\_test)

print(f"R²: {r2\_score(y\_test, y\_pred):.3f}")

print(f"RMSE: {np.sqrt(mean\_squared\_error(y\_test, y\_pred)):.3f}")

print("\n===== Bivariate Analysis: Pima (Logistic Regression) =====")

X\_pima = df\_pima[["Glucose"]] # using Glucose as single predictor

y\_pima = df\_pima["Outcome"]

Xp\_train, Xp\_test, yp\_train, yp\_test = train\_test\_split(X\_pima, y\_pima, test\_size=0.2, random\_state=42, stratify=y\_pima)

log\_reg = LogisticRegression(max\_iter=1000).fit(Xp\_train, yp\_train)

yp\_pred = log\_reg.predict(Xp\_test)

yp\_prob = log\_reg.predict\_proba(Xp\_test)[:, 1]

print(f"Accuracy: {accuracy\_score(yp\_test, yp\_pred):.3f}")

print(f"AUC: {roc\_auc\_score(yp\_test, yp\_prob):.3f}")

print("\n===== Multiple Regression: UCI Diabetes =====")

X\_all = sm.add\_constant(df\_uci.drop(columns=["target"]))

model = sm.OLS(df\_uci["target"], X\_all).fit()

print(model.summary())

print("\n===== Multiple Logistic Regression: Pima =====")

X\_pima\_all = df\_pima.drop(columns=["Outcome"])

scaler = StandardScaler()

X\_scaled = scaler.fit\_transform(X\_pima\_all)

Xp\_train, Xp\_test, yp\_train, yp\_test = train\_test\_split(X\_scaled, y\_pima, test\_size=0.2, random\_state=42, stratify=y\_pima)

log\_reg\_full = LogisticRegression(max\_iter=2000).fit(Xp\_train, yp\_train)

yp\_pred\_full = log\_reg\_full.predict(Xp\_test)

yp\_prob\_full = log\_reg\_full.predict\_proba(Xp\_test)[:, 1]

print(f"Accuracy: {accuracy\_score(yp\_test, yp\_pred\_full):.3f}")

print(f"AUC: {roc\_auc\_score(yp\_test, yp\_prob\_full):.3f}")

print(confusion\_matrix(yp\_test, yp\_pred\_full))

print(classification\_report(yp\_test, yp\_pred\_full))

print("\n===== COMPARISON =====")

print(f"UCI R² (Multiple Regression): {model.rsquared:.3f}")

print(f"Pima Accuracy (Multiple Logistic): {accuracy\_score(yp\_test, yp\_pred\_full):.3f}, AUC: {roc\_auc\_score(yp\_test, yp\_prob\_full):.3f}")

Output:-

===== Bivariate Analysis: UCI Diabetes (Linear Regression) =====

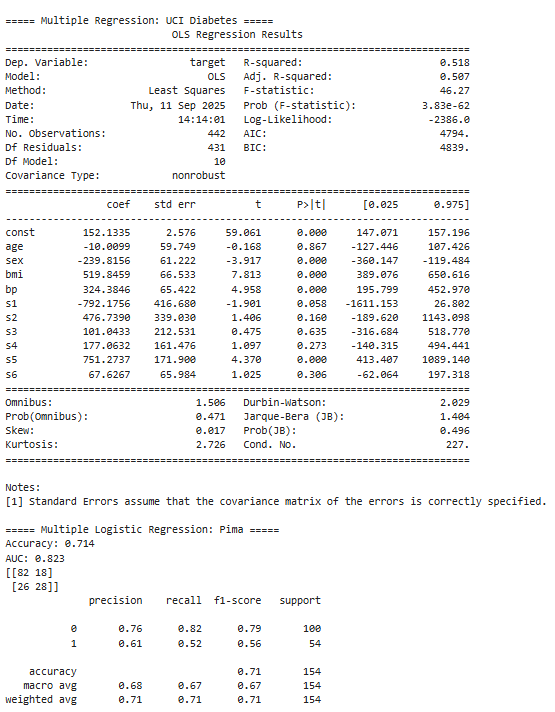
R²: 0.233

RMSE: 63.732

===== Bivariate Analysis: Pima (Logistic Regression) =====

Accuracy: 0.708

AUC: 0.767



===== COMPARISON =====

UCI R² (Multiple Regression): 0.518

Pima Accuracy (Multiple Logistic): 0.714, AUC: 0.823