به نام خدا

نام و نام خانوادگی: امیرحسین عبدیان

شماره دانشجویی: 4021661210

پرسش3 بخش1 :

این خطوط کتابخانه‌های pandas برای عملیات‌های مرتبط با داده‌ها، seaborn برای رسم نمودارهای آماری و matplotlib.pyplot برای رسم نمودار را وارد می‌کنند.

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

این بخش فایل weatherHistory.csv را بارگذاری می‌کند و داده‌ها را در یک DataFrame به نام data ذخیره می‌کند.

file\_path = '/content/weatherHistory.csv'

data = pd.read\_csv(file\_path)

در این خط، ستون‌های مورد نظر برای تحلیل همبستگی از DataFrame انتخاب می‌شوند: Temperature (C)، Apparent Temperature (C) و Humidity.

selected\_data = data[['Temperature (C)', 'Apparent Temperature (C)', 'Humidity']]

این خط ماتریس همبستگی بین ستون‌های انتخاب شده را با استفاده از تابع corr محاسبه می‌کند.

correlation\_matrix = selected\_data.corr()

این بخش نمودار حرارتی (heatmap) ماتریس همبستگی را رسم می‌کند:

plt.figure(figsize=(8, 6)): تنظیم اندازه نمودار.

sns.heatmap(correlation\_matrix, annot=True, cmap='coolwarm', fmt=".2f"): رسم نمودار حرارتی با استفاده از seaborn.heatmap و تنظیمات مختلف.

plt.title('Correlation Matrix Heatmap'): افزودن عنوان به نمودار.

plt.show(): نمایش نمودار.

plt.figure(figsize=(8, 6))

sns.heatmap(correlation\_matrix, annot=True, cmap='coolwarm', fmt=".2f")

plt.title('Correlation Matrix Heatmap')

plt.show()

این بخش نمودارهای پراکندگی و هیستوگرام‌های ویژگی‌های انتخاب شده را رسم می‌کند:

sns.pairplot(selected\_data): رسم نمودارهای پراکندگی و هیستوگرام‌ها با استفاده از seaborn.pairplot.

plt.suptitle('Scatter Plots and Histograms of Selected Features'): افزودن عنوان به نمودارها.

plt.show(): نمایش نمودارها.

sns.pairplot(selected\_data)

plt.suptitle('Scatter Plots and Histograms of Selected Features')

plt.show()

پرسش3 بخش2 :

این خطوط کتابخانه‌های pandas برای عملیات‌های مرتبط با داده‌ها، numpy برای عملیات‌های عددی، matplotlib.pyplot و seaborn برای رسم نمودارها، LinearRegression و Ridge از sklearn.linear\_model برای ایجاد مدل‌های رگرسیون، mean\_squared\_error از sklearn.metrics برای ارزیابی مدل‌ها و train\_test\_split از sklearn.model\_selection برای تقسیم داده‌ها را وارد می‌کنند.

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.linear\_model import LinearRegression, Ridge

from sklearn.metrics import mean\_squared\_error

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

این بخش فایل weatherHistory.csv را بارگذاری می‌کند و داده‌ها را در یک DataFrame به نام data ذخیره می‌کند.

file\_path = '/content/weatherHistory.csv'

data = pd.read\_csv(file\_path)

در این بخش، ستون‌های Temperature (C) و Humidity به عنوان ویژگی‌ها (X) و ستون Apparent Temperature (C) به عنوان هدف (y) انتخاب می‌شوند. سپس داده‌ها به دو بخش آموزشی (80 درصد) و تست (20 درصد) تقسیم می‌شوند. random\_state=42 برای اطمینان از تکرارپذیری نتایج استفاده می‌شود.

X = data[['Temperature (C)', 'Humidity']]

y = data['Apparent Temperature (C)']

# Split the data into training and testing sets

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

در این بخش، ابتدا مدل رگرسیون خطی با استفاده از LinearRegression ایجاد و با داده‌های آموزشی آموزش داده می‌شود. سپس پیش‌بینی‌ها برای داده‌های تست انجام شده و میانگین مربع خطا (mean\_squared\_error) محاسبه می‌شود. در ادامه، مدل رگرسیون ریج با استفاده از Ridge ایجاد و با داده‌های آموزشی آموزش داده می‌شود. سپس پیش‌بینی‌ها برای داده‌های تست انجام شده و میانگین مربع خطا محاسبه می‌شود.

# Initialize and train the Linear Regression model (Least Squares)

ls\_model = LinearRegression()

ls\_model.fit(X\_train, y\_train)

y\_pred\_ls = ls\_model.predict(X\_test)

mse\_ls = mean\_squared\_error(y\_test, y\_pred\_ls)

# Initialize and train the Ridge Regression model (Regularized Least Squares)

ridge\_model = Ridge(alpha=1.0)

ridge\_model.fit(X\_train, y\_train)

y\_pred\_ridge = ridge\_model.predict(X\_test)

mse\_ridge = mean\_squared\_error(y\_test, y\_pred\_ridge)

این بخش نتایج پیش‌بینی مدل‌های رگرسیون خطی و رگرسیون ریج را در دو نمودار جداگانه رسم می‌کند:

در نمودار اول (چپ)، پیش‌بینی‌های مدل رگرسیون خطی رسم می‌شوند.

در نمودار دوم (راست)، پیش‌بینی‌های مدل رگرسیون ریج رسم می‌شوند.

هر نمودار شامل نقاط پراکندگی (پیش‌بینی‌ها در مقابل مقادیر واقعی) و خط ایده‌آل (پیش‌بینی کامل) است.

در نهایت، عنوان کلی شامل مقادیر MSE برای هر دو مدل اضافه می‌شود.

plt.figure(figsize=(12, 6))

# Plot for Least Squares predictions

plt.subplot(1, 2, 1)

plt.scatter(y\_test, y\_pred\_ls, alpha=0.5, color='blue', label='Predictions')

plt.plot([y\_test.min(), y\_test.max()], [y\_test.min(), y\_test.max()], 'k--', lw=2, label='Ideal Fit')

plt.xlabel('Actual Apparent Temperature (C)')

plt.ylabel('Predicted Apparent Temperature (C)')

plt.title('Least Squares Regression')

plt.legend()

# Plot for Ridge Regression predictions

plt.subplot(1, 2, 2)

plt.scatter(y\_test, y\_pred\_ridge, alpha=0.5, color='red', label='Predictions')

plt.plot([y\_test.min(), y\_test.max()], [y\_test.min(), y\_test.max()], 'k--', lw=2, label='Ideal Fit')

plt.xlabel('Actual Apparent Temperature (C)')

plt.ylabel('Predicted Apparent Temperature (C)')

plt.title('Ridge Regression')

plt.legend()

plt.suptitle(f'Comparison of Regression Models\nMSE LS: {mse\_ls:.2f}, MSE Ridge: {mse\_ridge:.2f}')

plt.show()

این بخش مقادیر میانگین مربع خطا (MSE) برای هر دو مدل را چاپ می‌کند تا مقایسه‌ای بین دقت مدل‌ها انجام شود.

print(f"Mean Squared Error for Least Squares (LS): {mse\_ls:.4f}")

print(f"Mean Squared Error for Ridge Regression (RLS): {mse\_ridge:.4f}")

پرسش3 بخش3 :

این خطوط کتابخانه‌های numpy برای عملیات‌های عددی، pandas برای عملیات‌های مرتبط با داده‌ها، matplotlib.pyplot برای رسم نمودار و statsmodels.api برای مدل‌های آماری را وارد می‌کنند.

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import statsmodels.api as sm

این بخش فایل weatherHistory.csv را بارگذاری می‌کند و داده‌ها را در یک DataFrame به نام data ذخیره می‌کند.

file\_path = '/content/weatherHistory.csv'

data = pd.read\_csv(file\_path)

در این بخش، ستون Temperature (C) به عنوان ویژگی‌ها (X) و ستون Apparent Temperature (C) به عنوان هدف (y) انتخاب می‌شوند. وزن‌ها به عنوان معکوس رطوبت (Humidity) تعریف می‌شوند، با اضافه کردن یک مقدار کوچک (0.01) برای جلوگیری از تقسیم بر صفر.

X = data[['Temperature (C)']]

y = data['Apparent Temperature (C)']

weights = 1 / (data['Humidity'] + 0.01)  # Adding a small constant to avoid division by zero

در این بخش، یک ثابت به مدل اضافه می‌شود که شامل بایاس (intercept) است. سپس مدل کمترین مربعات وزن‌دار (WLS) با استفاده از داده‌ها و وزن‌ها ایجاد و آموزش داده می‌شود. خلاصه نتایج مدل چاپ می‌شود. این بخش در یک بلوک try قرار دارد تا در صورت وقوع خطاها، آنها مدیریت شوند.

X = sm.add\_constant(X)

# Fit the Weighted Least Squares model

try:

    model = sm.WLS(y, X, weights=weights)

    results = model.fit()

    print(results.summary())

این بخش مقادیر واقعی و مقادیر پیش‌بینی شده را در یک نمودار رسم می‌کند:

plt.figure(figsize=(10, 6)): تنظیم اندازه نمودار.

plt.scatter(data['Temperature (C)'], data['Apparent Temperature (C)'], color='blue', alpha=0.5, label='Actual Data'): رسم نقاط پراکندگی برای داده‌های واقعی.

plt.plot(X['Temperature (C)'], results.fittedvalues, 'r', label='Fitted values', linewidth=2): رسم خط مقادیر پیش‌بینی شده.

plt.title('Weighted Least Squares Fit'): افزودن عنوان به نمودار.

plt.xlabel('Temperature (C)'): افزودن برچسب به محور x.

plt.ylabel('Apparent Temperature (C)'): افزودن برچسب به محور y.

plt.legend(): نمایش توضیحات.

plt.show(): نمایش نمودار.

# Plotting the fitted values

    plt.figure(figsize=(10, 6))

    plt.scatter(data['Temperature (C)'], data['Apparent Temperature (C)'], color='blue', alpha=0.5, label='Actual Data')

    plt.plot(X['Temperature (C)'], results.fittedvalues, 'r', label='Fitted values', linewidth=2)

    plt.title('Weighted Least Squares Fit')

    plt.xlabel('Temperature (C)')

    plt.ylabel('Apparent Temperature (C)')

    plt.legend()

    plt.show()

این بخش خطاهای مربوط به جبر خطی (np.linalg.LinAlgError) و سایر خطاها را مدیریت و چاپ می‌کند.

except np.linalg.LinAlgError as e:

    print("Linear algebra error during the fit:", e)

except Exception as e:

    print("An error occurred:", e)