**Data analysis tools for Experimental Research Tutorial 2B**

Read the instructions carefully

1. **Please label your x-axis and y-axis for all the questions. If Axis labels are not present, the answers will not be evaluated.**
2. **No discussion is allowed during the tutorial session. Evaluation of the tutorial will be performed in class.**
3. **Dimension reduction**: Using the dataset from the previous tutorial, you fit a linear as well as non linear regression model. Now you decide to visualize your predictions compared to the original data, since visualization of a 4 dimensional data isn’t feasible you decide to reduce the dimensional to a simple 2D plot. Using this 2D plot for both actual data and predicted data, you can visualize the goodness of fit of your model.

### Fit a Liner regression model to the given dataset and obtain coefficients. Make a goodness of fit plot. (5 + 5)

### Fit a Non-Linear regression model to the given dataset including the interaction terms and obtain coefficients. Make a goodness of fit plot (5 + 5)

**Use following equation:** b0 + b1\*X1 + b2\*X2 + b3\*X3 + b4\*X2\*\*2 + b5\*X3\*\*2 + b6\*X2\*X3

### Reduce the dimensions of the Linear Model to obtain Principal Components 1 & 2 for actual as well as predicted dataset. Plot both the points in the same plot were ‘blue’ points denote actual dataset, and ‘red’ points denote predicted dataset. (5 + 5)

### Reduce the dimensions of the Non - Linear Model to obtain Principal Components 1 & 2 for actual as well as predicted dataset. Plot both the points in the same plot were ‘blue’ points denote actual dataset, and ‘red’ points denote predicted dataset. (5 + 5)

### A graph of a line graph Description automatically generated with medium confidence A graph with a red line and blue dots Description automatically generated

### 

### A graph with red and blue dots Description automatically generated

## Code Hints

from sklearn.decomposition import PCA

import seaborn as sns

import numpy as np

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

data = pd.read\_excel(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_) #Fill yourself

#import regression model

from sklearn.linear\_model import LinearRegression as regression

X1 = data.x1

X2 = data.x2

X3 = data.x3

Y = data.y

X = np.column\_stack((X1, X2, X3))

#We don't have to use one's column if we are using linear regression from sklearn

linear\_model = regression().fit(X, Y)

b0 = linear\_model.intercept\_

b1, b2, b3 = linear\_model.coef\_

Y\_predicted\_linear = b0 + b1\*X1 + (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_) # Fill Yourself

print('The coefficients are: ', b0, b1, b2, b3)

#Goodness of Fit plot for linear model

sns.scatterplot(x = Y\_predicted\_linear, y = Y, color = 'blue', alpha = 0.8, linewidth = 0.5, label = 'Predicted vs Observed')

plt.plot(Y, Y, color = 'red', alpha = 1, linewidth = 0.5, label = 'y = x line')

plt.xlabel('Predicted Y')

plt.ylabel('Observed Y')

plt.title('Goodness of Fit plot for linear model')

plt.legend()

plt.show()

#Check the shape of data before dimension reduction

print('The shape of data before dimension reduction is: \n',np.shape(data))

pca\_linear = PCA(n\_components=2)

pca\_linear.fit(data)

data1\_2dim = pca\_linear.transform(data)

pca1\_actual = (\_\_\_\_\_\_\_\_\_\_\_\_\_) # Fill Yourself

pca2\_actual = (\_\_\_\_\_\_\_\_\_\_\_\_\_) # Fill Yourself

#Check the shape of data after dimension reduction

print('The shape of data after dimension reduction in PCA is: \n',np.shape(data1\_2dim))

# Now we will use the PCA to reduce the dimension of the predicted data

data\_predicted\_linear = np.column\_stack((Y\_predicted\_linear, data[['x1', 'x2', 'x3']]))

print('The matrix with predicted values is:')

print(data\_predicted\_linear)

print('The shape of data before dimension reduction is: ', data\_predicted\_linear.shape)

pca\_linear = PCA(n\_components=2)

pca\_linear.fit(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_) # Fill Yourself

data\_predicted\_linear\_2dim = pca\_linear.transform(data\_predicted\_linear)

#Check the shape of data after dimension reduction

print('The shape of data after dimension reduction is: ', data\_predicted\_linear\_2dim.shape)

pca1\_predicted\_linear = (\_\_\_\_\_\_\_\_\_\_\_\_\_) # Fill Yourself

pca2\_predicted\_linear = (\_\_\_\_\_\_\_\_\_\_\_\_\_) # Fill Yourself

import matplotlib.pyplot as plt

plt.scatter(pca1\_actual,pca2\_actual, color ='blue', label = 'Actual data')

plt.scatter(pca1\_predicted\_linear,pca2\_predicted\_linear, color ='red', label = 'Predicted data')

plt.xlabel('PCA1')

plt.ylabel('PCA2')

plt.title('PCA 1 vs PCA 2 plot (Linear Model)')

plt.legend()

plt.show()