

NAME: MOHAMMAD HUSSAM(2033.KHI.DEG.020)

PAIRING WITH : MAVIA ALAM KHAN (2303.KHI.DEG.017)

&

AQSA TAUHEED(2303.KHI.DEG.011)

ASSIGNMENT 2.4

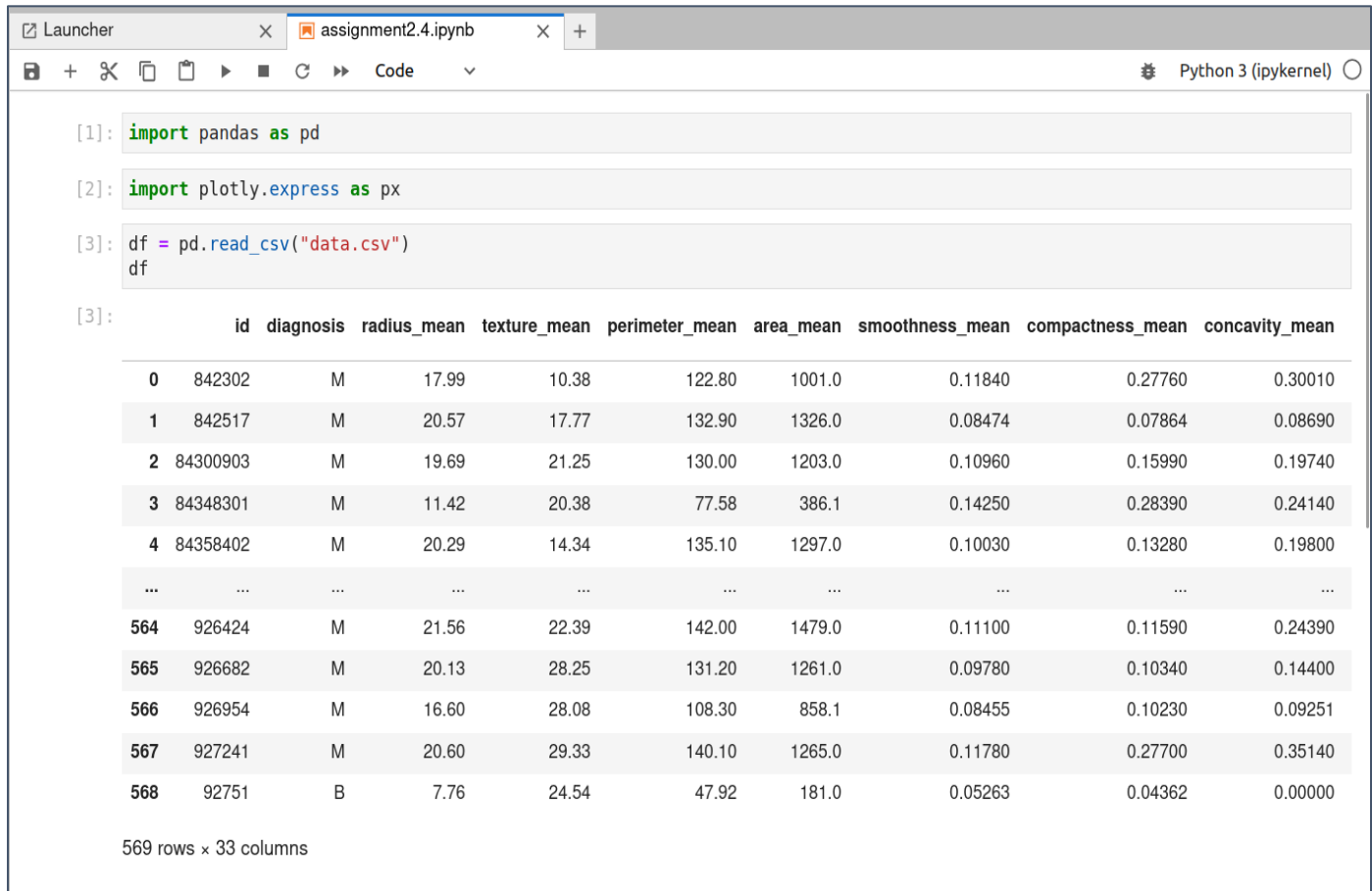
Download the Breast Cancer Wisconsin dataset from <https://www.kaggle.com/datasets/uciml/breast-cancer-wisconsin-data>.

After downloading, read about scatter matrix and implement it using plotly. Limit it to only few (5-6) features of your choice. Try to make it as readable as possible (eg. use colors to represent target class).

SOLUTION:

SCATTER MATRIX: A scatter matrix is a matrix of scatterplots, where each variable in a dataset is plotted against every other variable. It is a powerful graphical tool that helps in understanding the relationships between variables and identifying patterns or trends in the data. A scatter matrix can be especially useful in exploratory data analysis when dealing with high-dimensional datasets. By visualizing the relationships between variables, we can gain insights into the underlying structure of the data and make informed decisions about how to proceed with further analysis or modeling.

STEP:1



The screenshot shows a Jupyter Notebook window titled "assignment2.4.ipynb". The interface includes a toolbar with icons for saving, running, and other actions. The code cells are as follows:

```
[1]: import pandas as pd
```

```
[2]: import plotly.express as px
```

```
[3]: df = pd.read_csv("data.csv")
df
```

The output of the third cell is a DataFrame with 10 columns: id, diagnosis, radius_mean, texture_mean, perimeter_mean, area_mean, smoothness_mean, compactness_mean, and concavity_mean. The first few rows are displayed, with an ellipsis indicating that there are 569 rows in total.

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean
0	842302	M	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.30010
1	842517	M	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.08690
2	84300903	M	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.19740
3	84348301	M	11.42	20.38	77.58	386.1	0.14250	0.28390	0.24140
4	84358402	M	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.19800
...
564	926424	M	21.56	22.39	142.00	1479.0	0.11100	0.11590	0.24390
565	926682	M	20.13	28.25	131.20	1261.0	0.09780	0.10340	0.14400
566	926954	M	16.60	28.08	108.30	858.1	0.08455	0.10230	0.09251
567	927241	M	20.60	29.33	140.10	1265.0	0.11780	0.27700	0.35140
568	92751	B	7.76	24.54	47.92	181.0	0.05263	0.04362	0.00000

569 rows x 33 columns

After downloading data.csv file ,we have to load the file first using read_csv funtion in pandas , so first of all we imported pandas as pd , and also imported plotly.express as px for plotting functions to create visualizations from the data in the DataFrame. Then we loaded data.csv file and assigned it to df and printed output for it, as shown in image .

STEP:2

```
[4]: selected_features = ['radius_mean', 'texture_mean', 'perimeter_mean', 'area_mean', 'smoothness_mean', 'compactness_mean', 'diagnosis']  
[ ]:
```

Now for scatter matrix , we have to select the features to include in the scatter matrix which include :

(radius_mean,'texture_mean','perimeter_mean','area_mean','smoothness_mean','compactness_mean', 'diagnosis') as shown in image above . . We will use the first six features to plot the scatter matrix, and use the diagnosis variable to color code the data points.

```
[8]: color_map = {'M': '#e41a1c', 'B': '#377eb8'}
```

We define a color map for the diagnosis column, where M (malignant) is assigned the color red (#E41A1C) and B (benign) is assigned the color blue (#377EB8). We created a dictionary of these mappings and assign it to a variable called color_map as shown above in image.

STEP : 3

```
[9]: fig = px.scatter_matrix(  
    df[selected_features],  
    dimensions=selected_features[:-1],  
    color='diagnosis',  
    color_discrete_map=color_map  
)
```

We created the scatter matrix using Plotly Express , code which shown in image above generates a scatterplot matrix that shows the pairwise relationships between the selected features, colored by the diagnosis column. The resulting scatterplot matrix provides an easy way to visualize the relationships between multiple variables and to identify any patterns or trends in the data.

`df[selected_features]` selects a subset of columns from a pandas DataFrame `df` based on the list of `selected_features`.

`selected_features[:-1]` is used to select all the features in the `selected_features` list except for the last one.

`color='diagnosis'` is used to specify the variable to use for color.

`color_discrete_map=color_map` is used to specify a custom mapping between discrete values of a categorical variable and colors to be used for each value in a plot.

STEP:4

```
[11]: fig.update_layout(  
    title='Breast Cancer Diagnosis: ',  
    font=dict(size=12),  
    width=1000,  
    height=950,  
    dragmode='select',  
    hovermode='closest',  
    xaxis=dict(titlefont=dict(size=14), tickfont=dict(size=10)),  
    yaxis=dict(titlefont=dict(size=14), tickfont=dict(size=12)),  
)
```

The `fig.update_layout()` function is a method provided by Plotly's Figure class that allows you to update the layout and style of a plot that has been created using Plotly.

title: This sets the plot title to Breast Cancer Diagnosis:

font: This sets the font size for all text in the plot to 12.

width: This sets the width of the plot to 1000 pixels.

height: This sets the height of the plot to 950 pixels.

dragmode: This sets the drag mode of the plot to select, which allows the user to select a region of the plot to zoom in on.

hovermode: This sets the hover mode of the plot to closest, which displays the data point closest to the mouse cursor when hovering over the plot.

xaxis: This sets the properties of the x-axis, including the title and font size of the axis labels.

yaxis: This sets the properties of the y-axis, including the title and font size of the axis labels.

```
fig.show()
```

And then fig.show() for showing the output of scatter matrix.

OUTPUT

