Experiment-3.3

Student Name: Milan Sharma **UID: 23MAI10003**
Branch: ME-CSE AI&ML **Section/Group: 23MAI-1/A**
Semester: 1
Subject Name: Advanced Python Programming **Subject Code: 23CSH-623**

1. **Aim:** Breast Cancer case study using Python Libraries

2. **Objective:** A breast cancer case study using Python typically involves the analysis of breast cancer data to build a predictive model for classification (malignant or benign tumors). In this example, I'll use the Breast Cancer Wisconsin (Diagnostic) dataset, which is commonly used for such studies. We'll use the scikit-learn library for machine learning and other libraries for data analysis and visualization. This case study covers loading the dataset, exploring and visualizing the data, preprocessing, and building a simple Logistic Regression model for breast cancer classification. further explore more advanced models, hyperparameter tuning, and feature selection based on your specific requirements.

3. **Algorithm/ Steps for Experiment :**

    1. Import necessary libraries for data manipulation, analysis, visualization, and machine learning.
    2. Load the Breast Cancer dataset (e.g., Breast Cancer Wisconsin dataset.
    3. Visualize relationships between features using pairplots and correlation heatmaps.
    4. Split the data into features (X) and target (y).
    5. Split the data into training and testing sets.
    6. Standardize features using StandardScaler.
    7. Choose a machine learning model (e.g., Logistic Regression).
    8. Train the model using the training set.
    9. Make predictions on the test set.
    10. Evaluate the model's performance using accuracy, confusion matrix, and classification report.

4. Code for Experiment and output :

In [35]:
```
import sklearn
import numpy as np
import pandas as pd
import plotly as plot
import plotly.express as px
import plotly.graph_objs as go


import matplotlib.pyplot as plt
import seaborn as sns
import os
from sklearn.metrics import accuracy_score
import plotly.offline as pyo
from plotly.offline import init_notebook_mode,plot,iplot
```

In [36]:
```
df = pd.read_csv(r"C:\Users\kanis\Desktop\Python\Multi_Disease_Predictor\models\breastcancer.csv")
```

In [37]:
```
df = df.rename(columns={'BreastCancerPedigreeFunction':'BCPF'})
```

In [38]:
```
print(df.columns)
```

```
Index(['diagnosis', 'radius_1ean', 'texture_1ean', 'peri1eter_1ean',
       'area_1ean', 's1oothness_1ean', 'co1pactness_1ean', 'concavity_1ean',
       'concave points_1ean', 'sy11etry_1ean', 'radius_se', 'peri1eter_se',
       'area_se', 'co1pactness_se', 'concavity_se', 'concave points_se',
       'fractal_di1ension_se', 'radius_worst', 'texture_worst',
       'peri1eter_worst', 'area_worst', 's1oothness_worst',
       'co1pactness_worst', 'concavity_worst', 'concave points_worst',
       'sy11etry_worst', 'fractal_di1ension_worst'],
      dtype='object')
```

In [39]:
```
df
```

Out[39]:

	diagnosis	radius_1ean	texture_1ean	peri1eter_1ean	area_1ean	s1oothness_1ean	co1pactness_1ean	concavity_1ean	concave points_1ean	sy11etry_1ean	...
0	1	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.30010	0.14710	0.2419	...
1	1	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.08690	0.07017	0.1812	...
2	1	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.19740	0.12790	0.2069	...
3	1	11.42	20.38	77.58	386.1	0.14250	0.28390	0.24140	0.10520	0.2597	...
4	1	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.19800	0.10430	0.1809	...
...	...	...	...	...	...	...	...	...	...	...	...
564	1	21.56	22.39	142.00	1479.0	0.11100	0.11590	0.24390	0.13890	0.1726	...
565	1	20.13	28.25	131.20	1261.0	0.09780	0.10340	0.14400	0.09791	0.1752	...

In [40]:
```
df.shape
```

Out[40]: (569, 27)

In [41]:
```
df.size
```

Out[41]: 15363

In [42]:
```
df.describe()
```

Out[42]:

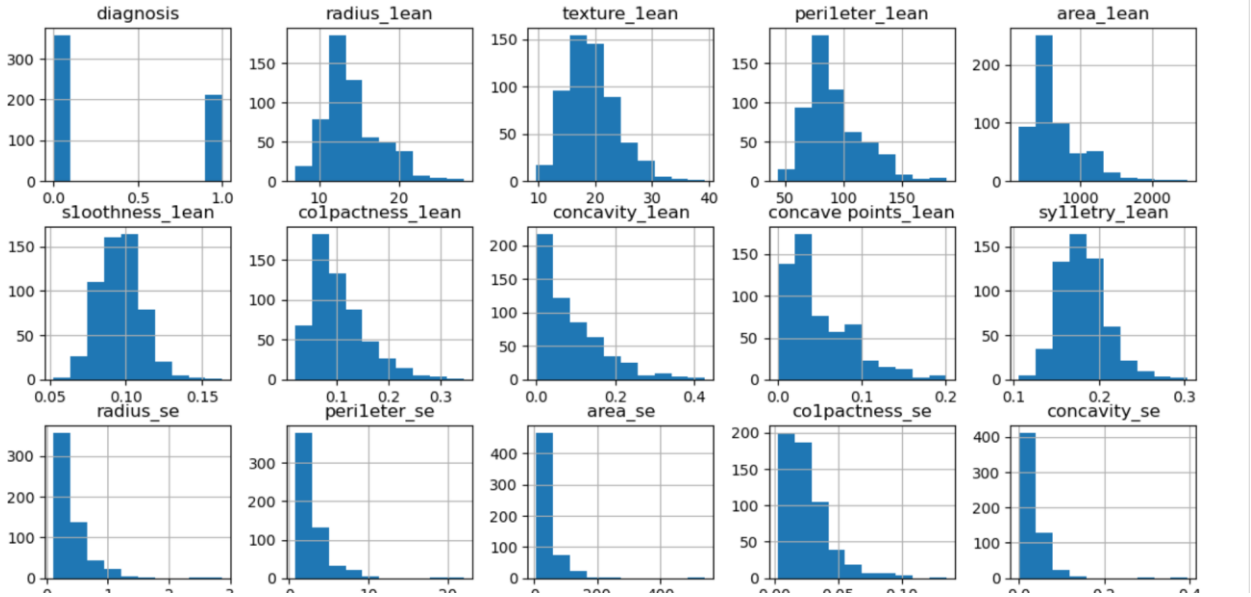
	diagnosis	radius_1ean	texture_1ean	peri1eter_1ean	area_1ean	s1oothness_1ean	co1pactness_1ean	concavity_1ean	concave points_1ean	sy11etry_1ea
count	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.0000(
mean	0.372583	14.127292	19.289649	91.969033	654.889104	0.096360	0.104341	0.088799	0.048919	0.1811(
std	0.483918	3.524049	4.301036	24.298981	351.914129	0.014064	0.052813	0.079720	0.038803	0.0274(
min	0.000000	6.981000	9.710000	43.790000	143.500000	0.052630	0.019380	0.000000	0.000000	0.1060(
25%	0.000000	11.700000	16.170000	75.170000	420.300000	0.086370	0.064920	0.029560	0.020310	0.1619(
50%	0.000000	13.370000	18.840000	86.240000	551.100000	0.095870	0.092630	0.061540	0.033500	0.1792(
75%	1.000000	15.780000	21.800000	104.100000	782.700000	0.105300	0.130400	0.130700	0.074000	0.1957(
max	1.000000	28.110000	39.280000	188.500000	2501.000000	0.163400	0.345400	0.426800	0.201200	0.3040(

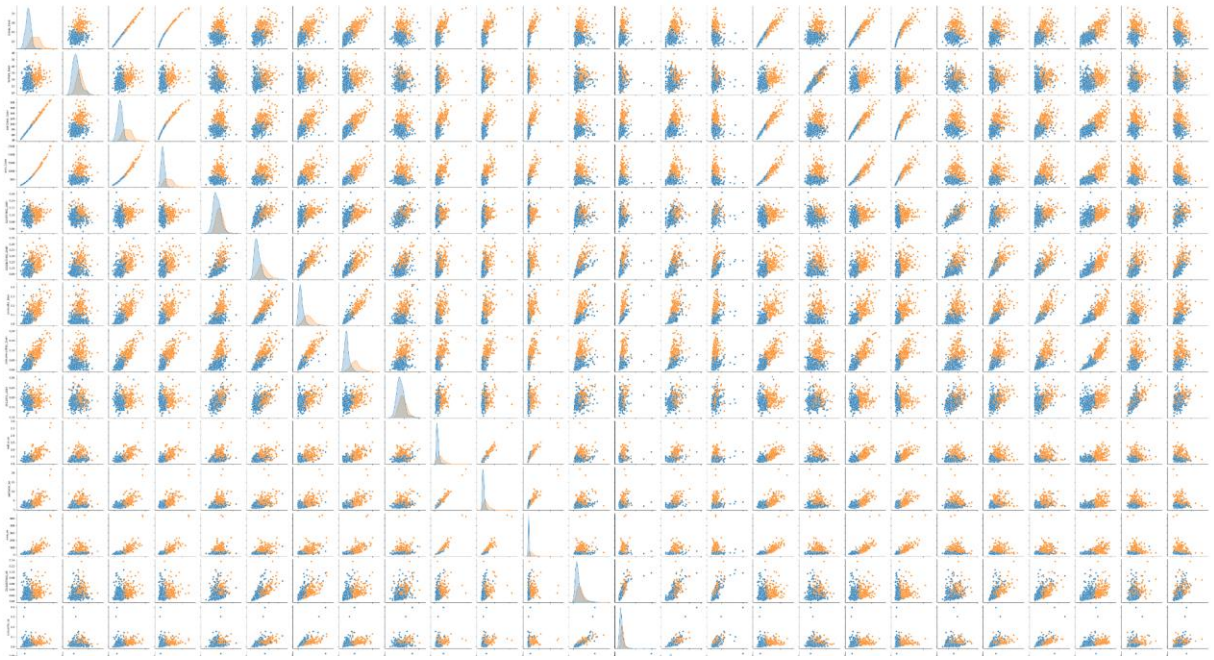
8 rows × 27 columns

In [43]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 569 entries, 0 to 568
Data columns (total 27 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   diagnosis               569 non-null    int64
 1   radius_1ean             569 non-null    float64
 2   texture_1ean            569 non-null    float64
 3   peri1eter_1ean          569 non-null    float64
 4   area_1ean               569 non-null    float64
 5   s1oothness_1ean         569 non-null    float64
 6   co1pactness_1ean        569 non-null    float64
 7   concavity_1ean          569 non-null    float64
 8   concave points_1ean     569 non-null    float64
 9   sy11etry_1ean           569 non-null    float64
 10  radius_se               569 non-null    float64
 11  peri1eter_se            569 non-null    float64
 12  area_se                 569 non-null    float64
 13  co1pactness_se          569 non-null    float64
 14  concavity_se            569 non-null    float64
 15  concave points_se       569 non-null    float64
 16  fractal_di1ension_se    569 non-null    float64
 17  radius_worst            569 non-null    float64
 18  texture_worst           569 non-null    float64
 19  peri1eter_worst         569 non-null    float64
 20  area_worst              569 non-null    float64
 21  s1oothness_worst        569 non-null    float64
```

In [44]:
```
df.hist(figsize=(14,14))
plt.show()
```





```
<Figure size 700x700 with 0 Axes>
```

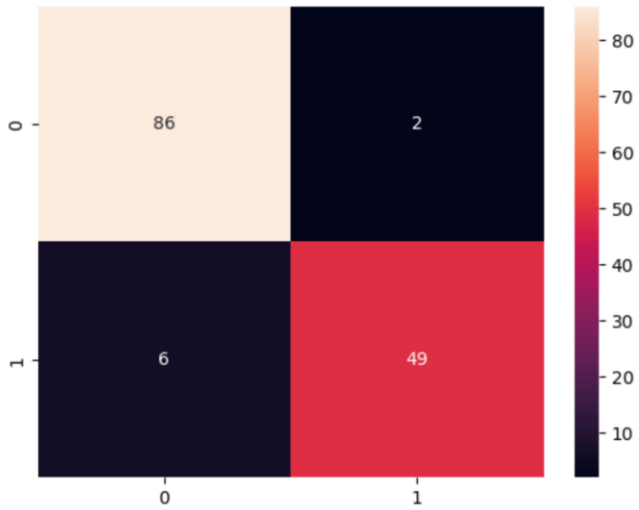
In [46]:
```
#define X variables and our target(y)
X = df.drop(['diagnosis'],axis=1).values
y = df['diagnosis'].values
#split Train and Test
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=101)
```

In [47]:
```
from sklearn.svm import SVC
svc_model = SVC()
svc_model.fit(X_train, y_train)
```

Out[47]:
```
SVC()
```

In [48]:
```
from sklearn.metrics import classification_report, confusion_matrix
y_predict = svc_model.predict(X_test)
cm = confusion_matrix(y_test, y_predict)
sns.heatmap(cm, annot=True)
```

Out[48]:
```
<Axes: >
```



Learning Outcomes:

1. Learnt how to import libraries in python program.
2. Learnt to load dataset in python.
3. Learnt how to train model in python.
4. Learnt how to evaluate model's performance using accuracy, confusion matrix, and classification report.