HW 2 Report

Results:

1 & 2. Load and Describe the Dataset

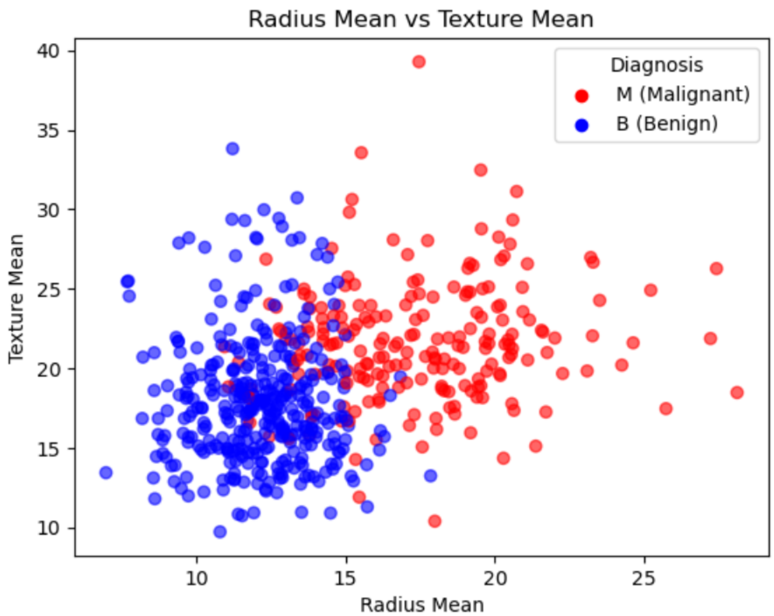
- **Code Output**:
  The cancer dataset was loaded; it contains **569 rows** and **33 columns** The column types were a mix of float values and one categorical target column (diagnosis).

```
           id diagnosis  radius_mean  texture_mean  perimeter_mean  area_mean  \
0      842302         M        17.99         10.38          122.80     1001.0
1      842517         M        20.57         17.77          132.90     1326.0
2    84300903         M        19.69         21.25          130.00     1203.0
3    84348301         M        11.42         20.38           77.58      386.1
4    84358402         M        20.29         14.34          135.10     1297.0
..        ...       ...          ...           ...             ...        ...
564    926424         M        21.56         22.39          142.00     1479.0
565    926682         M        20.13         28.25          131.20     1261.0
566    926954         M        16.60         28.08          108.30      858.1
567    927241         M        20.60         29.33          140.10     1265.0
568     92751         B         7.76         24.54           47.92      181.0

     smoothness_mean  compactness_mean  concavity_mean  concave points_mean  \
0            0.11840           0.27760         0.30010              0.14710
1            0.08474           0.07864         0.08690              0.07017
2            0.10960           0.15990         0.19740              0.12790
3            0.14250           0.28390         0.24140              0.10520
4            0.10030           0.13280         0.19800              0.10430
..               ...               ...             ...                  ...
564          0.11100           0.11590         0.24390              0.13890
565          0.09780           0.10340         0.14400              0.09791
566          0.08455           0.10230         0.09251              0.05302
567          0.11780           0.27700         0.35140              0.15200
568          0.05263           0.04362         0.00000              0.00000

     ...  texture_worst  perimeter_worst  area_worst  smoothness_worst  \
0    ...          17.33           184.60      2019.0           0.16220
1    ...          23.41           158.80      1956.0           0.12380
2    ...          25.53           152.50      1709.0           0.14440
3    ...          26.50            98.87       567.7           0.20980
4    ...          16.67           152.20      1575.0           0.13740
..   ...            ...              ...         ...               ...
564  ...          26.40           166.10      2027.0           0.14100
565  ...          38.25           155.00      1731.0           0.11660
566  ...          34.12           126.70      1124.0           0.11390
567  ...          39.42           184.60      1821.0           0.16500
568  ...          30.37            59.16       268.6           0.08996

     compactness_worst  concavity_worst  concave points_worst  symmetry_worst  \
0              0.66560           0.7119                0.2654          0.4601
1              0.18660           0.2416                0.1860          0.2750
2              0.42450           0.4504                0.2430          0.3613
3              0.86630           0.6869                0.2575          0.6638
4              0.20500           0.4000                0.1625          0.2364
..                 ...              ...                   ...             ...
564            0.21130           0.4107                0.2216          0.2060
565            0.19220           0.3215                0.1628          0.2572
566            0.30940           0.3403                0.1418          0.2218
567            0.86810           0.9387                0.2650          0.4087
568            0.06444           0.0000                0.0000          0.2871

     fractal_dimension_worst  Unnamed: 32
0                    0.11890          NaN
1                    0.08902          NaN
2                    0.08758          NaN
3                    0.17300          NaN
4                    0.07678          NaN
..                       ...          ...
564                  0.07115          NaN
565                  0.06637          NaN
566                  0.07820          NaN
567                  0.12400          NaN
568                  0.07039          NaN

[569 rows x 33 columns]
Shape: (569, 33)
```

```
Column Names: Index(['id', 'diagnosis', 'radius_mean', 'texture_mean', 'perimeter_mean',
       'area_mean', 'smoothness_mean', 'compactness_mean', 'concavity_mean',
       'concave points_mean', 'symmetry_mean', 'fractal_dimension_mean',
       'radius_se', 'texture_se', 'perimeter_se', 'area_se', 'smoothness_se',
       'compactness_se', 'concavity_se', 'concave points_se', 'symmetry_se',
       'fractal_dimension_se', 'radius_worst', 'texture_worst',
       'perimeter_worst', 'area_worst', 'smoothness_worst',
       'compactness_worst', 'concavity_worst', 'concave points_worst',
       'symmetry_worst', 'fractal_dimension_worst', 'Unnamed: 32'],
      dtype='object')
id                           int64
diagnosis                   object
radius_mean                float64
texture_mean               float64
perimeter_mean             float64
area_mean                  float64
smoothness_mean            float64
compactness_mean           float64
concavity_mean             float64
concave points_mean        float64
symmetry_mean              float64
fractal_dimension_mean     float64
radius_se                  float64
texture_se                 float64
perimeter_se               float64
area_se                    float64
smoothness_se              float64
compactness_se             float64
concavity_se               float64
concave points_se          float64
symmetry_se                float64
fractal_dimension_se       float64
radius_worst               float64
texture_worst              float64
perimeter_worst            float64
area_worst                 float64
smoothness_worst           float64
compactness_worst          float64
concavity_worst            float64
concave points_worst       float64
symmetry_worst             float64
fractal_dimension_worst    float64
Unnamed: 32                float64
dtype: object
```

3. Scatter Plot of Radius Mean vs Texture Mean

- **Plot Description**: The scatter plot shows the relationship between the radius_mean and texture_mean features.
- **Legend**: Red points represent Malignant tumors (M), and blue points represent Benign tumors (B).
- **Observations**: The data points are **not linearly separable** based on these two features alone. This suggests that a simple linear model might not perform well for classification



4. Encoding the Target Variable

- The target column diagnosis was encoded using LabelEncoder.
- **Encoding Scheme**:
  - Malignant (M) → 1
  - Benign (B) → 0
- This encoding enables the model to handle the categorical target as numeric values.

```
0    1
1    1
2    1
3    1
4    1
Name: diagnosis, dtype: int64
```

6. Train-Test Split (70-30)

- The dataset was successfully split into **70% training data** and **30% test data**.
- **Training Data Shape**: (398, 32)
- **Testing Data Shape**: (171, 32)
  This split ensures that the model has sufficient data for training and testing.

```
Training data shape: (398, 32), (398,)
Testing data shape: (171, 32), (171,)
```

7. Imputation of Missing Values

- Missing values in the dataset were handled using a **SimpleImputer** with the mean strategy.
- This step ensures that the Gaussian Naive Bayes model can train without errors caused by NaN values.

8,9. Confusion Matrix and Classification Report

- The Gaussian Naive Bayes model had 60% accuracy. It did a good job finding benign tumors (98% recall) but struggled a lot with malignant ones (only 3% recall), leading to many false negatives. It guessed benign cases correctly most of the time (60% precision), but it was bad at catching malignant ones, with an F1-score of just 0.06. This might be because the data had more benign cases, making the model lean towards those predictions. Since missing malignant tumors is dangerous, this model isn't good enough to use in real life. To make it better, we could try balancing the data, scaling the features, or using different models like Logistic Regression or Random Forest.
- I chose Gaussian Naive Bayes because it works well with continuous data, which matches the features in this dataset (radius_mean and texture_mean). It assumes the data follows a normal distribution, and since it's simple and fast, it made sense to start with it for this classification task. Naive Bayes is also easy to train and handles smaller datasets well. But, because it makes strong guesses about how the data is shaped, the model didn't do great with finding malignant tumors, which shows it might not be the best fit here.

```
Confusion Matrix:
 [[101   2]
 [ 66   2]]

Classification Report:
              precision    recall  f1-score   support

           0       0.60      0.98      0.75       103
           1       0.50      0.03      0.06        68

    accuracy                           0.60       171
   macro avg       0.55      0.50      0.40       171
weighted avg       0.56      0.60      0.47       171
```