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).

A screenshot of a document

Description automatically generated

A screen shot of a computer screen

Description automatically generated

**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

A diagram of red and blue dots

Description automatically generated

**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.

A screenshot of a computer

Description automatically generated

**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.

A black text on a white background

Description automatically generated

**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.

**A screenshot of a computer screen

Description automatically generated**