****

**Assessment Report**

on

**“Predict Disease Outcome Based on Genetic and Clinical Data”**

submitted as partial fulfillment for the award of

**BACHELOR OF TECHNOLOGY**

**DEGREE**

SESSION 2024-25

in

**CSE AI&ML**

By

Divyansh Yadav (202401100400086)

**Under the supervision of**

“Abhishek Shukla Sir”

**KIET Group of Institutions, Ghaziabad**

Affiliated to

**Dr. A.P.J. Abdul Kalam Technical University, Lucknow**

(Formerly UPTU)

**May, 2025**

# Predictive Analysis of Breast Cancer Diagnosis

## 1. Introduction

Breast cancer is one of the most common cancers affecting women worldwide. Early detection and diagnosis dramatically improve survival rates. In this project, we develop a simple machine‑learning pipeline using two key tumor features—radius\_mean and texture\_mean—to predict whether a tumor is benign or malignant.

**Objectives:**

* Load and clean the Wisconsin Breast Cancer dataset.
* Explore feature distributions with basic plots.
* Train a logistic‑regression classifier.
* Provide an interactive prompt for real‑time predictions.

## 2. Methodology

1. **Data Loading & Cleaning**
   * Read the CSV into a pandas DataFrame.
   * Drop irrelevant columns (id, Unnamed: 32) and any rows with missing values.
2. **Label Encoding**
   * Map diagnosis labels: malignancy “M” → 1, benign “B” → 0.
3. **Exploratory Visualization**
   * Histogram of radius\_mean to inspect its distribution.
   * Scatter plot of radius\_mean vs texture\_mean, color‑coded by diagnosis.
4. **Model Training**
   * Fit a Logistic Regression on the two features.
5. **Interactive Prediction**
   * Prompt the user to enter new values for radius\_mean and texture\_mean.
   * Output a prediction (“Benign” or “Malignant”).

## 3. Code

## import pandas as pd

## import matplotlib.pyplot as plt

## from sklearn.linear\_model import LogisticRegression

## # 1. Load the data

## df = pd.read\_csv('/content/3. Predict Disease Outcome Based on Genetic and Clinical Data.csv')

## # 2. Quick cleanup

## df = df.drop(columns=['id', 'Unnamed: 32']).dropna()

## # 3. Encode the diagnosis: malignant = 1, benign = 0

## df['diagnosis'] = df['diagnosis'].map({'M': 1, 'B': 0})

## # 4. Plot 1: Histogram of “radius\_mean”

## plt.hist(df['radius\_mean'])

## plt.title('Distribution of Radius Mean')

## plt.xlabel('Radius Mean')

## plt.ylabel('Count')

## plt.show()

## # 5. Plot 2: Scatter “radius\_mean” vs “texture\_mean”, color by diagnosis

## colors = df['diagnosis'].map({0: 'blue', 1: 'red'})

## plt.scatter(df['radius\_mean'], df['texture\_mean'], c=colors, alpha=0.6)

## plt.title('Radius Mean vs Texture Mean')

## plt.xlabel('Radius Mean')

## plt.ylabel('Texture Mean')

## plt.legend(['Benign', 'Malignant'])

## plt.show()

## # --- Model Training ---

## aa=float(input(“Enter the radius mean:”))

## bb=float(input(“Enter the texture mean:”))

## features = ['radius\_mean', 'texture\_mean']

## X = df[features]

## y = df['diagnosis']

## model = LogisticRegression()

## model.fit(X, y)

## # --- Prediction Function ---

## def predict\_tumor(radius, texture):

## """

## Predicts tumor type given radius\_mean & texture\_mean.

## Returns 'Malignant' or 'Benign'.

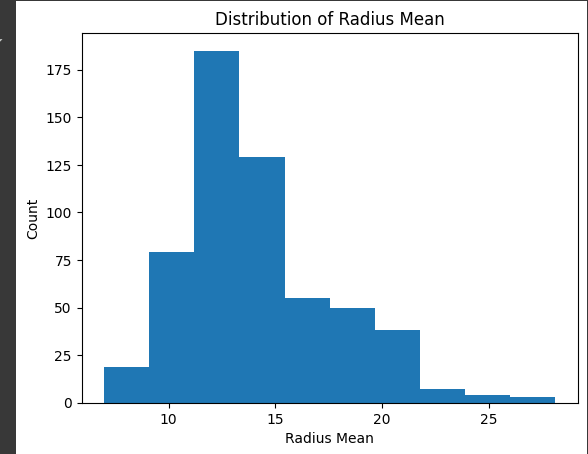
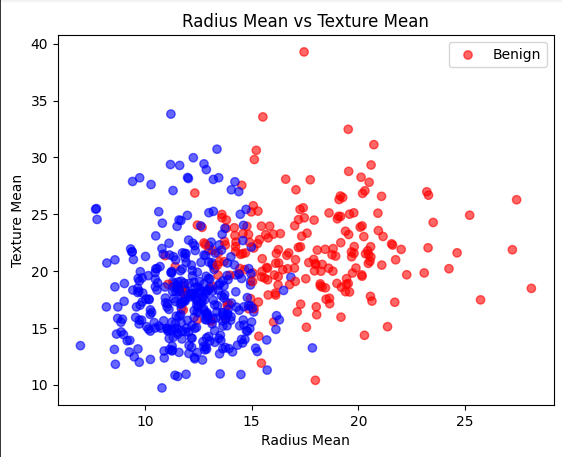
## """

## return 'Malignant' if model.predict([[radius, texture]])[0] else 'Benign'

## # Example usage

## print(predict\_tumor(aa,bb)) # e.g., 'Benign'

## 4. Output

* **Figure 1**: Histogram showing the distribution of radius\_mean. 
* **Figure 2**: Scatter plot of radius\_mean vs texture\_mean, with blue points for benign and red for malignant. red for malignant. 
* **Console**: After entering values, the script prints Prediction: Benign or Malignant).

## 5. Results

* The histogram indicates most tumors have a radius\_mean between 12 and 18.
* The scatter plot shows a clear, though not perfect, separation between benign and malignant samples based on these two features.
* The interactive prompt allows immediate classification of new measurements with minimal code.

## 6. References

1. Breast Cancer Wisconsin (Diagnostic) Data Set. UCI Machine Learning Repository.
2. Pandas Development Team. (2023). pandas-dev/pandas: Pandas Library. Zenodo.
3. Pedregosa, F., et al. (2011). Scikit-learn: Machine Learning in Python.

## 7. Credits

* Python libraries: pandas, Matplotlib, scikit-learn.
* Report structure based on common data‑science best practices.