# **Project Report**

**Name :** Prem M **Course :** Machine Learning Fundamentals / U24CST362 **Registration Number :**  TU6243202111036

### **Project Title:** Breast Cancer Diagnosis Using Machine Learning

## **1. Project Description**

This project classifies tumors as malignant or benign using medical diagnostic data. The goal is to build a model that accurately identifies cancerous tumors based on features extracted from cell nuclei in breast tissue samples. This has real-world applications in early cancer detection and improving patient outcomes.

## **2. Learning Objectives**

* **Objective 1:** Apply classification algorithms (KNN, Logistic Regression) to healthcare datasets.
* **Objective 2**: Understand preprocessing steps for medical data.
* **Objective 3**: Evaluate model performance using metrics like accuracy, precision, recall, and ROC-AUC.
* **Objective 4**: Visualize decision boundaries and interpret model behavior.
* **Objective 5**: Handle imbalanced data and compare classification models.

## **3. Timeline**

* **Submission date:** Oct 6, 2025

## **4. Algorithm Used**

* **Algorithm Name:** K-Nearest Neighbors (KNN) and Logistic Regression
* **Explanation:** **KNN** classifies a tumor based on the majority label of its nearest neighbors. **Logistic Regression** models the probability of malignancy using a sigmoid function and interpretable coefficients.

## **5. Tools & Libraries**

**Programming Language:** Python

**Libraries Used:**

* **Pandas** → for handling and manipulating datasets
* **NumPy** → for numerical computations and calculations
* **Scikit-learn** → for building regression models and evaluation
* **Matplotlib / Seaborn** → for data visualization and plotting graphs

## **6. Dataset Description**

* **Source:** Provided CSV file (Brest Cancer Wisconsin.csv)
* **Size:** 569 rows X 30 Features
* **Target Variable:** Diagnosis (Malignant = M, Benign = B)
* **Description of Features:** Includes Radius Mean, Texture Mean, Perimeter Mean, Area Mean, Smoothness Mean, Compactness, Concavity, Symmetry, Fractal Dimension (mean, worst, standard error)

## **7. Methodology**

**🔧 Data Preprocessing**

* Removed missing/null values
* Encoded target variable (M = 1, B = 0)
* Scaled features using StandardScaler
* Split dataset into 80% training and 20% testing

**🧪 Model Training**

* Trained KNN and Logistic Regression models
* Used cross-validation to tune hyperparameters (e.g., K value)

**📊 Evaluation**

* Metrics: Accuracy, Precision, Recall, F1-score, ROC-AUC
* Visualized confusion matrix and ROC curves
* Compared decision boundaries

**🔁 Improvements**

* Tuned K in KNN using GridSearchCV
* Applied regularization in Logistic Regression
* Handled class imbalance using SMOTE

## **8. Results**

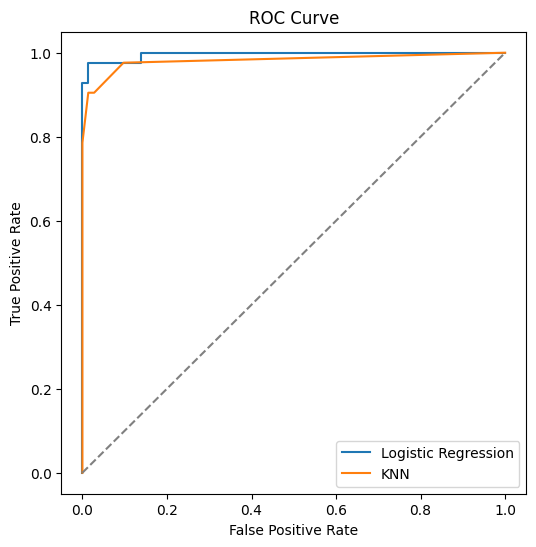
* Visualizations:
* Confusion Matrix
* ROC Curve
* Decision Boundary Plot
* Feature Correlation Heatmap
* **Insights**:
* **Radius Mean**, **Perimeter Mean**, and **Area Mean** are the most predictive features
* Logistic Regression slightly outperforms KNN
* ROC-AUC confirms strong classification performance

## **9. Questions Answered**

1. **Which features are most predictive?** Radius, Perimeter, Area, and Texture Mean
2. **How does KNN work?** Classifies based on majority label of nearest neighbors using distance metrics
3. **What is accuracy, precision, recall?** Accuracy = overall correctness, Precision = correct positives, Recall = sensitivity
4. **How to choose K in KNN?** Use cross-validation; small K = sensitive, large K = smoother but may underfit
5. **What is ROC-AUC?** Measures model’s ability to distinguish classes; closer to 1 = better
6. **How to handle imbalanced data?** Use SMOTE, class weighting, or resampling techniques
7. **What preprocessing is needed?** Encode labels, scale features, handle missing values, split dataset
8. **How to visualize decision boundaries?** Use meshgrid and predict over 2D feature space with Matplotlib
9. **What is overfitting?** Model fits training data too well but fails on test data; solved with regularization
10. **How to compare models?** Use metrics (Accuracy, Precision, Recall, ROC-AUC), confusion matrix, and visual plots

**🔍 Extra 5 Questions**

1. **How does feature scaling affect KNN and Logistic Regression?** KNN is distance-based, so scaling is essential; Logistic Regression benefits from faster convergence.
2. **What is the role of distance metrics in KNN?** Euclidean and Manhattan distances affect neighbor selection and classification.
3. **How to interpret logistic regression coefficients?** Coefficients show log-odds change in malignancy per unit increase in feature.
4. **What is cross-validation and why use it?** Evaluates model performance across multiple splits to avoid overfitting.
5. **How to tune hyperparameters in KNN and Logistic Regression?** Use GridSearchCV to find optimal K and regularization strength.



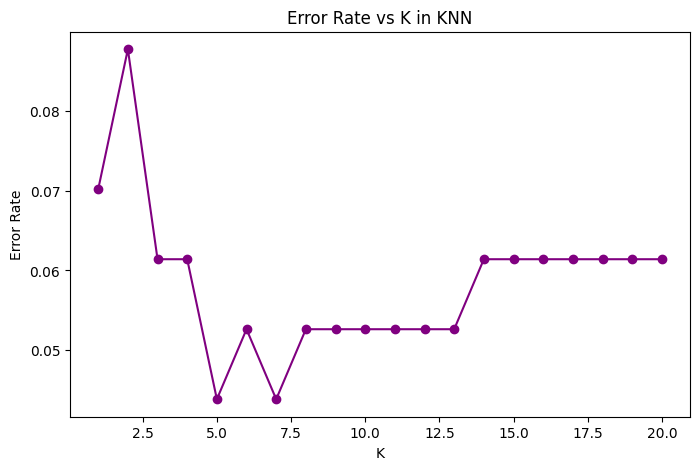
**ROC Curve** shows how well a model separates classes.

It plots **True Positive Rate** vs **False Positive Rate**.

**AUC (Area Under Curve)** measures performance:

* + **AUC = 1.0** → perfect
  + **AUC = 0.5** → random guessing

Higher AUC = better cancer detection.



* **K** = number of neighbors used to classify a point. **Small K (e.g., 1–3)** → high accuracy on training but may overfit.
* **Large K (e.g., 10+)** → smoother predictions but may underfit.
* Use **cross-validation** to test multiple K values.
* Plot **Accuracy vs K** → pick the K with highest stable accuracy.
* Use **Confusion Matrix** and **ROC Curve** to evaluate performance.
* Visualize **Decision Boundaries** → sharp edges = low K, smooth = high K.

## **10. Challenges & Improvements**

* **Challenges:**
* Scaling features for KNN
* Handling class imbalance
* Visualizing decision boundaries in high dimensions
* **Future Improvements:**
* Add Support Vector Machines and Decision Trees
* Use ensemble methods like Gradient Boosting
* Deploy model with Streamlit for interactive diagnosis

## **11. References**

* + **Dataset Links: Custom Excel dataset provided for the project.**
* **Scikit-learn Documentation:** [**https://scikit-learn.org/**](https://scikit-learn.org/)
* **Extra Reading: Blogs & tutorials on text classification, TF-IDF, and Logistic Regression.**

## **12. GitHub Link**

https://github.com/premcodemaster005m-star/ML\_Project\_4