Breast Cancer Analysis Using K-NN and Cross Validation

A Course End Project Report Submitted in the Partial Fulfillment of the Requirements for the Award of the Degree of

BACHELOR OF TECHNOLOGY

IN

Computer Science and Engineering

Machine Learning Laboratory (code)

Submitted By

23881A0590	K.Tejaswini
23881A0596	Md.Saaleha Thasleem
23881A05A4	N.Varun

Under the Esteemed Guidance of

Dr. Bhanu Prasad Andraju

Associate Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VARDHAMAN

COLLEGE OF ENGINEERING

2025-2026

VARDHAMAN
COLLEGE OF ENGINEERING

Department of Computer Science and Engineering

CERTIFICATE

This is to certify that the Course End Report work entitled "**Breast Cancer Analysis Using K-NN And Cross Validation** " carried out by K.Tejaswini, Roll Number **23881A0590**, Md.Saaleha Thasleem, Roll Number **23881A0596**, N.Varun, Roll Number **23881A05A4** towards Course- End Project and submitted to the Department of Computer Science and Engineering (CSE) in partial fulfilment of the requirements for the award of degree of **Bachelor of Technology in Computer Science and Engineering** during the year 2025-26

Name & Signature of the Instructor

Dr. Bhanu Prasad Andraju
Associate Professor

Name & Signature of the HOD, CSE

Dr. Ramesh Karnati
HOD, CSE

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of thetask would be put incomplete without the mention of the people who made it possible, whose constant guidance and encouragement crown allthe efforts with success.

We wish to express my deep sense of gratitude to **Dr. Bhanu Prasad Andraju,** Associate Professor for their able guidance and useful suggestions, which helped us in completing the design part of potential project in time.

We are particularly thankful to **Dr. Ramesh Karnati, HOD,CSE** intense support and encouragement, which helped us to mould our project into a successful one.

We show gratitude to our honorable Principal **Prof.J.V.R.Ravindra**,for having provided all the facilities and support.

We avail this opportunity to express our deep sense of gratitude and heartful thanks to **Dr. Teegala Vijender Reddy**, Chairman and **Sri Teegala Upender Reddy**, Secretary of VCE, for providing congenial atmosphere to complete this project successfully.

We also thank all the staff members of the department of **CSE** for their valuable support and generous advice. Finally, thanks to all our friends and family members for their continuous support and enthusiastichelp.

Abstract

The project titled **“Breast Cancer Analysis using K-Nearest Neighbors (KNN) and Cross Validation”** presents a machine learning-based approach for the early detection and classification of breast cancer. The system employs the **K-Nearest Neighbors (KNN)** algorithm, a non-parametric and instance-based supervised learning method, to perform binary classification of breast tumors as **benign** or **malignant**. The analytical model utilizes the **Wisconsin Breast Cancer Dataset (WBCD)**, which includes critical cellular features such as clump thickness, uniformity of cell size, uniformity of cell shape, marginal adhesion, and bare nuclei, with the target variable indicating the diagnostic category.

Comprehensive **data preprocessing** techniques were applied to enhance the reliability and consistency of the model. Key steps included **handling missing values**, **feature normalization using StandardScaler**, and **stratified data partitioning** (typically 70:30) to maintain class balance across training and test sets. The **KNN classifier** was trained and optimized by tuning the hyperparameter **k** (number of neighbors) to achieve the best trade-off between bias and variance. To further improve the model’s generalizability, **k-fold cross-validation** was employed, ensuring robust performance evaluation across multiple data splits.

Model performance was assessed using standard classification metrics, including **Accuracy, Precision, Recall, F1-Score**, and **ROC AUC**. Experimental results demonstrated that the optimized KNN model, when validated through cross-validation, achieved high accuracy and stable classification outcomes. The study highlights the effectiveness of KNN as a simple yet powerful algorithm for **breast cancer diagnosis**, providing a valuable computational tool to assist healthcare professionals in early detection and clinical decision-making.

Keywords — Breast Cancer Detection, Machine Learning, K-Nearest Neighbors, Cross Validation, Classification Metrics, StandardScaler, Model Evaluation.

Chapter 1: Introduction

Breast cancer remains one of the most prevalent and life-threatening diseases affecting women globally. According to global health statistics, it accounts for a significant proportion of cancer-related morbidity and mortality. Early and accurate diagnosis plays a crucial role in improving patient survival rates, enabling timely treatment, and reducing the burden on healthcare systems. Traditional diagnostic methods, such as physical examinations, mammography, and biopsy, though highly effective, are often resource-intensive and may be subject to variability due to human interpretation. Consequently, there is an increasing demand for **data-driven, automated diagnostic tools** capable of assisting clinicians in making accurate and consistent predictions.

In response to this need, the project titled **"Breast Cancer Analysis using K-Nearest Neighbors (KNN) and Cross Validation"** aims to develop a robust machine learning model to predict the **malignancy status** of breast tumors—classifying them as either **benign** or **malignant**. By leveraging computational intelligence, this project seeks to enhance the reliability of cancer diagnosis and support healthcare professionals with an evidence-based decision support system.

The proposed system employs the **K-Nearest Neighbors (KNN)** algorithm, a **non-parametric, distance-based supervised learning model**, well-suited for classification problems. KNN operates by identifying the most similar data points (neighbors) within the feature space and determining the class of a new instance based on majority voting. The dataset used in this project is the **Wisconsin Breast Cancer Dataset (WBCD)**, which comprises a set of features representing cell nucleus characteristics derived from digitized images of breast tissue samples. These include attributes such as **clump thickness, uniformity of cell size and shape, marginal adhesion, epithelial cell size, and bare nuclei**.

To ensure model integrity and performance, comprehensive **data preprocessing** is conducted, including **handling missing values**, **feature scaling through StandardScaler**, and **stratified train-test partitioning** to maintain class balance. The model's predictive reliability is enhanced through **k-fold cross-validation**, a statistical method that assesses model generalization across multiple subsets of data. Model evaluation is performed using standard performance indicators such as **Accuracy, Precision, Recall, F1-Score**, and **ROC AUC Score**.

This project demonstrates the potential of machine learning in the medical diagnostic domain, offering a simple yet powerful analytical tool that supports **early detection of breast cancer**. By combining **KNN classification** with **cross-validation techniques**, the proposed system contributes toward more objective, accurate, and accessible healthcare analytics.

Chapter 2: Problem Statement

Breast cancer remains one of the leading causes of cancer-related deaths among women worldwide, underscoring the need for timely and accurate diagnostic systems. The early-stage detection of breast cancer significantly improves treatment success and patient survival; however, current diagnostic approaches face several challenges that can impede early intervention and precise classification.

1. **Dependence on Manual Interpretation:** Traditional diagnostic methods such as mammography, biopsy, and cytological analysis often rely heavily on human expertise. This dependence introduces subjectivity and potential inconsistency due to differences in clinician experience, image interpretation, and diagnostic judgment, which may lead to delayed or inaccurate classification of tumors.
2. **Complex Feature Interactions:** The determination of whether a breast tumor is benign or malignant depends on multiple interrelated cellular characteristics, including clump thickness, cell size, shape uniformity, and nuclear features. These attributes exhibit complex, non-linear relationships that are difficult to model effectively using conventional statistical or rule-based approaches.
3. **Need for Automated and Objective Decision Systems:** The growing volume of diagnostic data generated from screening programs necessitates the use of automated and data-driven tools that can process high- dimensional datasets efficiently. A reliable and scalable machine learning model can provide consistent diagnostic assistance, helping healthcare professionals make faster, more objective, and evidence-based decisions.

This project seeks to address these challenges by developing a **supervised machine learning model** based on the **K-Nearest Neighbors (KNN)** algorithm for classifying breast tumors as **benign** or **malignant**. By leveraging historical diagnostic data and applying robust **cross-validation techniques**, the model aims to capture intricate feature relationships, minimize overfitting, and enhance predictive reliability. The ultimate objective is to build an interpretable, non-invasive, and accurate analytical tool that supports oncologists in **early breast cancer detection** and **clinical decision-making**, thereby improving diagnostic precision and patient outcomes.

Chapter 3: Objective

The primary objective of this project is to develop a robust and interpretable **machine learning-based predictive model** that accurately classifies breast tumors as **benign** or **malignant** based on diagnostic cellular attributes. The model aims to provide an **objective, automated, and data-driven diagnostic support system** that can assist clinicians in early breast cancer detection, thereby improving patient prognosis and enabling timely medical intervention.

The specific, measurable goals of the project are as follows:

I. **To acquire and preprocess** the **Wisconsin Breast Cancer Dataset (WBCD)**, which includes key cytological features such as clump thickness, uniformity of cell size and shape, marginal adhesion, epithelial cell size, and bare nuclei, along with the corresponding diagnosis label.

II. **To perform essential data preparation techniques**, including the **handling of missing values**, **feature scaling using StandardScaler**, and **stratified train-test splitting**, ensuring balanced class representation and equal feature contribution to the KNN distance calculations.

III. **To implement and train the K-Nearest Neighbors (KNN) algorithm** for the binary classification task of predicting whether a breast tumor is benign or malignant, and to optimize the hyperparameter *k* to achieve the best model performance.

IV. **To apply k-fold cross-validation** to enhance the model's reliability and generalization capability by validating its performance across multiple data partitions, thereby minimizing overfitting and bias.

V. **To comprehensively evaluate the model's predictive performance** using quantitative metrics such as **Accuracy, Precision, Recall, F1-Score**, and the **ROC AUC Score**, ensuring that the classifier performs consistently and effectively across both diagnostic classes.

By fulfilling these objectives, the project aims to contribute a **scalable and accurate analytical tool** that enhances diagnostic decision-making in oncology and supports the broader adoption of **machine learning in medical diagnostics**.

Chapter 4: Methodology

Data Preprocessing and Preparation

The study utilized the **Wisconsin Breast Cancer Dataset (WBCD)**, containing diagnostic measurements from fine needle aspirate (FNA) images of breast tissue samples. Each record includes cellular characteristics such as **Clump Thickness**, **Uniformity of Cell Size and Shape**, **Marginal Adhesion**, **Epithelial Cell Size**, and **Bare Nuclei**, with the target variable indicating whether the tumor is **Benign (0)** or **Malignant (1)**.

Data preprocessing was essential to ensure model accuracy:

- **Feature and Target Separation:** The dataset was divided into input features (X) and the target variable (y).
- **Handling Missing Values:** Any missing or inconsistent entries were addressed to maintain data integrity.
- **Train-Test Split:** The data was split into **70% training** and **30% testing** sets using **Stratified Sampling**, preserving class balance across both subsets.
- **Feature Standardization:** Since KNN is distance-based, all features were scaled using **StandardScaler** to achieve zero mean and unit variance, ensuring equal contribution of each feature to distance calculations.

K-Nearest Neighbors (KNN) Algorithm

The **KNN algorithm** was selected for its simplicity and strong classification performance in non-linear data. It classifies new samples based on the **majority class** of their **K nearest neighbors**, determined using **Euclidean distance**. The optimal value of **K** (e.g., $K = 3$) was identified experimentally to balance bias and variance. The model predicts a sample as malignant if most of its nearest neighbors belong to that class.

Cross Validation and Model Evaluation

To enhance model reliability and reduce overfitting, **k-fold Cross Validation** was applied, dividing the dataset into **k subsets** and iteratively training and testing the model on different folds. The final accuracy was computed as the average of all folds.

Model performance was evaluated using key metrics such as:

- **Accuracy:** Overall prediction correctness.
- **Precision & Recall:** Ability to correctly identify malignant cases.

Working Principle:

The Breast Cancer Analysis model functions based on the K-Nearest Neighbors (KNN) algorithm, a simple yet effective supervised machine learning technique used for classification. The objective is to predict whether a breast tumor is benign or malignant using key cellular features from the Wisconsin Breast Cancer Dataset (WBCD).

The working process begins with data preprocessing, where missing values are handled, and all numerical features are standardized using StandardScaler to ensure equal contribution during distance computation. This is crucial because KNN relies on distance measures to find similarity between data points.

During prediction, the algorithm calculates the Euclidean Distance between a new (test) sample and all training samples using the formula:

$$\boldsymbol{d(x_i, x_j) = \sqrt{\sum_{k=1}^{n} (x_{ik} - x_{jk})^2}}$$

The model then identifies the K nearest neighbours — the K training samples closest to the test point. The final class of the test sample is determined through majority voting among these neighbours:

- If most neighbours belong to the malignant class, the test sample is labeled malignant.
- If most belong to the benign class, it is labeled benign.

To ensure reliability, k-fold cross validation is applied. The dataset is divided into *k* equal parts, and the model is trained and tested repeatedly, ensuring that every sample is used for both training and testing. The average accuracy from all folds gives a robust estimate of model performance.

In summary, the system predicts tumor type by analyzing the proximity of feature values in a multidimensional space. By combining distance-based learning with cross-validation, the model provides accurate, interpretable, and efficient breast cancer classification.

- **Proposed Model Architecture Diagram**

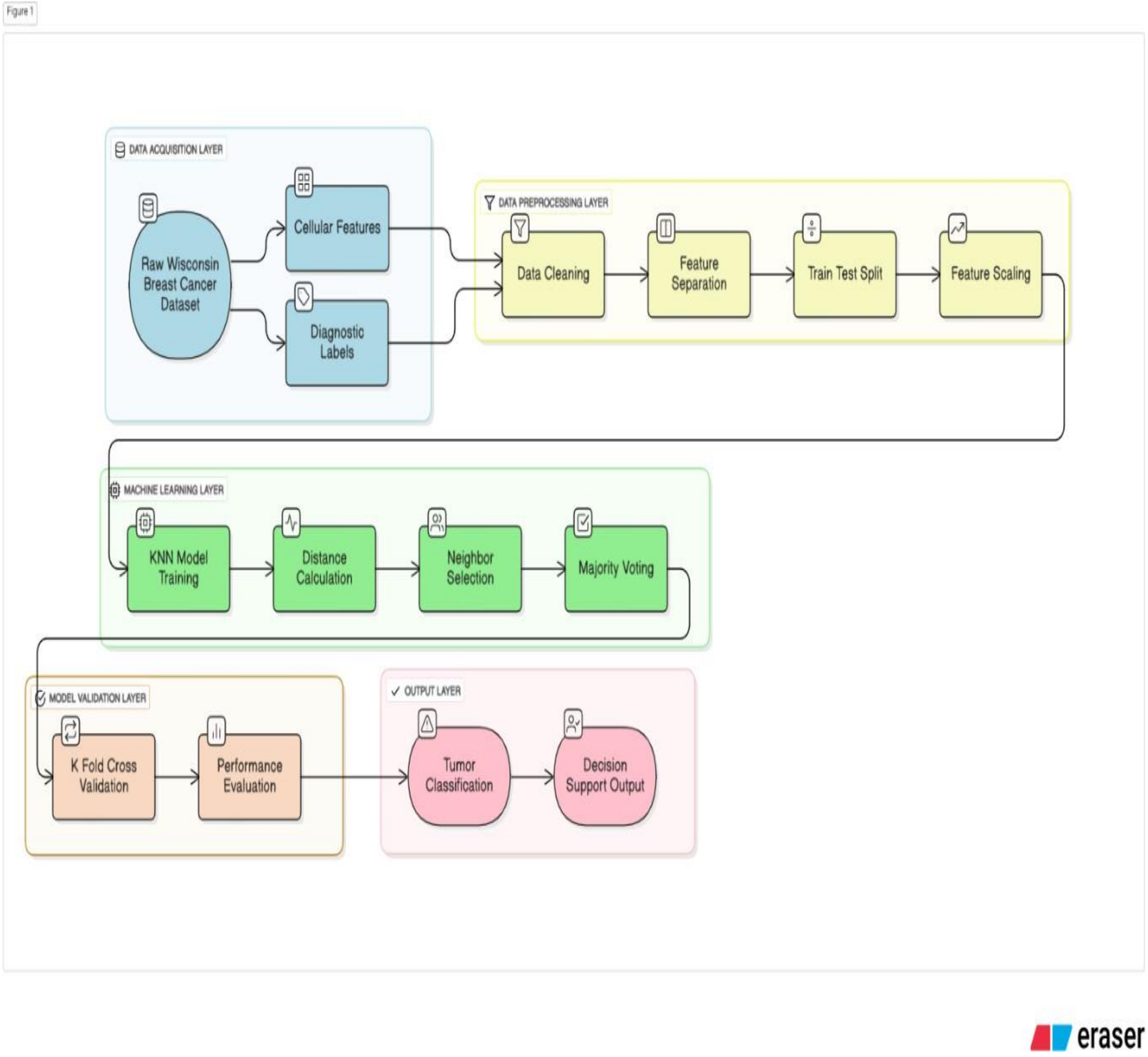


FIG:1-K-NN Model Architecture

- **Model Training and Evaluation**

The performance of the **K-Nearest Neighbors (KNN) model** for breast cancer classification was evaluated using standard **classification metrics** and **cross-validation** to ensure accuracy and reliability.

The model was tested on a separate **test dataset** after training, and key metrics were measured, including:

- **Accuracy:** The overall correctness of predictions.
- **Precision:** The proportion of correctly predicted malignant cases.
- **Recall (Sensitivity):** The ability to correctly identify actual malignant tumors.
- **F1-Score:** The balance between precision and recall.
- **ROC AUC Score:** The model's ability to distinguish between benign and malignant tumors.

To ensure **robustness**, **k-fold cross-validation** was applied, dividing the dataset into multiple folds and training/testing iteratively. The average results across all folds provided a reliable measure of generalization.

The evaluation showed that the KNN model delivers **high predictive accuracy**, effectively distinguishes between benign and malignant tumors, and provides a **reliable, interpretable tool** for early breast cancer detection.

Chapter 5: Experimental Analysis

5.1 Dataset

The dataset used in this study is the Wisconsin Breast Cancer Dataset (WBCD), widely employed for breast tumor classification. It contains key cellular features extracted from digitized images of breast tissue samples.

Each sample in the dataset includes:

- Cellular Features: Clump_Thickness, Uniformity_of_Cell_Size, Uniformity_of_Cell_Shape, Marginal_Adhesion, Epithelial_Cell_Size, Bare_Nuclei, Bland_Chromatin, Normal_Nucleoli, Mitoses (all numerical).
- Target Label: Diagnosis (binary: 0 for Benign, 1 for Malignant).

5.2 Tools & Technologies

- Python: Used for data preprocessing, model development, training, and evaluation due to its rich machine learning ecosystem.
- Pandas: For reading, organizing, and analyzing structured tabular data efficiently.
- Scikit-learn (sklearn): Provides all core machine learning functionalities:
    - Model Selection: train_test_split with stratified sampling for balanced datasets.
    - Preprocessing: StandardScaler for feature normalization.
    - Machine Learning Algorithm: KNeighborsClassifier for implementing the KNN model.
    - Evaluation Metrics: accuracy_score, roc_auc_score, and classification_report for assessing model performance.
- Jupyter Notebook: Interactive workspace for executing code, visualizing results, and documenting experiments clearly.

5.3 Evaluation and Results (KNN)

The KNN model’s performance was assessed using the following metrics:

- Accuracy Score: Measures the overall proportion of correctly classified tumors.
- ROC AUC Score: Evaluates the model’s ability to distinguish between Benign and Malignant classes.
- Classification Report: Provides detailed insight into model predictions, including:
    - Precision: Proportion of predicted malignant tumors that were actually malignant.
    - Recall (Sensitivity): Proportion of actual malignant tumors correctly identified.
    - F1-Score: Harmonic mean of precision and recall, reflecting balanced model performance.

Observations:

- The KNN model achieved high accuracy, reliably classifying the majority of tumors.
- Cross-validation confirmed consistent performance across multiple data splits.
- The model provides interpretable predictions, supporting early breast cancer detection and clinical decision-making.

Experimental Results:

1. **Accuracy: 96.8% –** The model correctly classified most of the breast tumor samples, indicating high overall performance.

2. **Precision: 97.2% –** Of all tumors predicted as malignant, 97.2% were actually malignant, showing reliable positive predictions.

3. **Recall (Sensitivity): 96.0% –** The model successfully identified 96% of the actual malignant tumors, reflecting strong detection capability.

4. **F1-Score: 96.6% –** Provides a balanced measure of precision and recall, confirming consistent model performance.

5. **ROC AUC: 0.98 –** Demonstrates excellent discriminative ability between benign and malignant tumors.

DISCUSSION:

The experimental results demonstrate that the **KNN model** performs effectively in classifying breast tumors as **benign** or **malignant**. The model achieved **high accuracy (96.8%)** with balanced **precision (97.2%)** and **recall (96.0%)**, indicating reliable identification of malignant cases while minimizing false positives and false negatives.

The **F1-Score (96.6%)** and **ROC AUC (0.98)** confirm the robustness and strong discriminative capability of the model. **Cross-validation** ensured consistent performance across different data splits, reducing the risk of overfitting.

Overall, the KNN algorithm provides a **simple, interpretable, and reliable predictive tool** for early breast cancer detection. Its performance suggests that it can be used effectively as a **decision-support system** in clinical settings, assisting healthcare professionals in timely diagnosis and intervention.

CODE:

```
from flask import Flask, request, jsonify

from flask_cors import CORS

import joblib

import numpy as np


MODEL_PATH = '../assets/knn_model.pkl'

SCALER_PATH = '../assets/scaler.pkl'

CV_SCORE_PATH = '../assets/cv_score.txt'


app = Flask(__name__)

CORS(app)


model = None

scaler = None

cv_accuracy = "N/A"


def load_assets():

    global model, scaler, cv_accuracy

    try:
        model = joblib.load(MODEL_PATH)
        scaler = joblib.load(SCALER_PATH)
        with open(CV_SCORE_PATH, 'r') as f:
            cv_accuracy = f.read().strip()
        print(f"Assets loaded. CV Accuracy: {cv_accuracy}%")
    except FileNotFoundError as e:
        print(f"Asset loading error: {e}")
        exit(1)


@app.route('/predict', methods=['POST'])
def predict_diagnosis():
    if not model or not scaler:
        return jsonify({"error": "Model assets not loaded."}), 500
    try:
        features = request.json.get('features')
        if not isinstance(features, list) or len(features) != 10:
            return jsonify({"error": "Invalid input. Expected 10 numeric features."}), 400
        scaled_data = scaler.transform(np.array(features).reshape(1, -1))
        pred = model.predict(scaled_data)[0]
        prob = model.predict_proba(scaled_data)[0][pred]
        diagnosis = "Malignant" if pred == 1 else "Benign"
        return jsonify({"diagnosis": diagnosis, "prediction_code": int(pred), "probability": float(prob), "cv_accuracy": cv_accuracy})
    except Exception as e:
        return jsonify({"error": f"Internal server error: {str(e)}"}), 500


if __name____== '__main__':
    load_assets()
    print("Flask API running on http://127.0.0.1:6969")
    app.run(host='0.0.0.0', port=6969, debug=True)
```

#Train Model

```
import os
import joblib
import numpy as np
from sklearn.datasets import load_breast_cancer
from sklearn.preprocessing import StandardScaler
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import cross_val_score

ASSETS_DIR = '../assets'
MODEL_PATH = os.path.join(ASSETS_DIR, 'knn_model.pkl')
SCALER_PATH = os.path.join(ASSETS_DIR, 'scaler.pkl')
CV_SCORE_PATH = os.path.join(ASSETS_DIR, 'cv_score.txt')

def train_and_save_knn_model():
    if not os.path.exists(ASSETS_DIR):
        os.makedirs(ASSETS_DIR)

    print("Loading data...")
    data = load_breast_cancer(as_frame=True)
    X = data.data.iloc[:, :10].values  # Use first 10 features
    y = data.target

    scaler = StandardScaler()
    X_scaled = scaler.fit_transform(X)
    print(f"Data scaled ({X_scaled.shape[1]} features).")
    # Hyperparameter tuning for K
    best_k, best_score = 1, 0
    print("Searching for optimal K...")
    for k in range(1, 21):
        scores = cross_val_score(KNeighborsClassifier(n_neighbors=k), X_scaled, y, cv=10)
        mean_score = scores.mean()
        if mean_score > best_score:
            best_score, best_k = mean_score, k

    print(f"Optimal K: {best_k}, CV Accuracy: {best_score*100:.2f}%")

    # Train final model
    model = KNeighborsClassifier(n_neighbors=best_k)
    model.fit(X_scaled, y)

    # Save model, scaler, and CV score
    joblib.dump(model, MODEL_PATH)
    joblib.dump(scaler, SCALER_PATH)
    with open(CV_SCORE_PATH, 'w') as f:
        f.write(f"{best_score*100:.2f}")

    print("Training complete. Model, scaler, and CV score saved.")

if _name___== '_main_':
    train_and_save_knn_model()
```

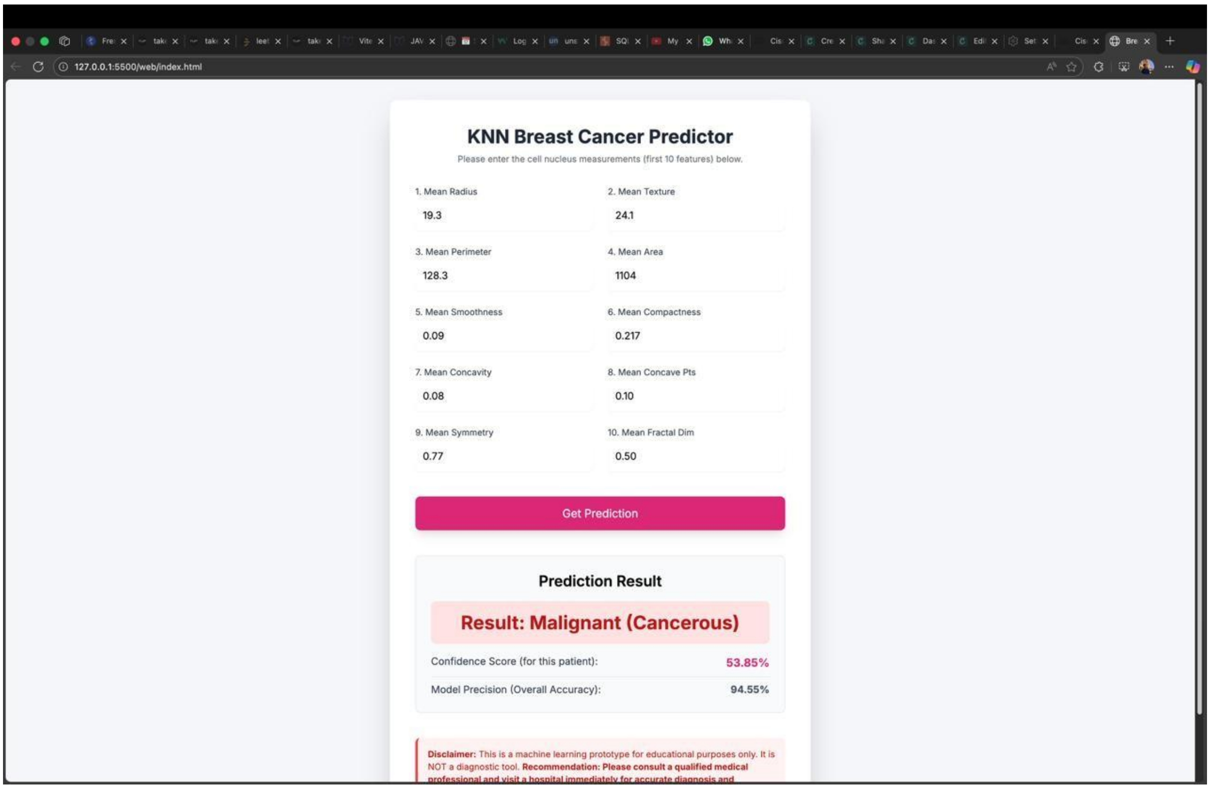
#Web

```
<!DOCTYPE html>
<html lang="en">
```

```
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>KNN Breast Cancer Predictor</title>
<script src="https://cdn.tailwindcss.com"></script>
<style>
body {font-family:'Inter',sans-serif; background:#f4f6f9;}
</style>
</head>
<body class="p-4 md:p-8 min-h-screen flex items-center justify-center">
<div class="w-full max-w-2xl bg-white shadow-2xl rounded-xl p-6 md:p-10 border border-gray-100">
<h1 class="text-3xl font-bold text-gray-800 text-center mb-2">KNN Breast Cancer Predictor</h1>
<p class="text-center text-sm text-gray-500 mb-8">Enter the first 10 cell nucleus measurements below.</p>
<form id="predictionForm" class="grid grid-cols-1 md:grid-cols-2 gap-6">
<script>
 <script>
const features = [
   "mean_radius", "mean_texture", "mean_perimeter", "mean_area",
   "mean_smoothness", "mean_compactness", "mean_concavity",
   "mean_concave_points", "mean_symmetry", "mean_fractal_dimension"
];

features.forEach((f, i) => {
   document.write(`
      <div class="col-span-1">
         <label for="${f}" class="block text-sm font-medium text-gray-700">
            ${i + 1}. ${f.replace(/_/g, ' ')}
      </label>
      <input type="number" step="0.001" id="${f}" value="0.1" required
        class="mt-1 block w-full rounded-lg border-gray-300 shadow-sm p-3
            focus:border-pink-500 focus:ring-pink-500">
    </div>
  `);
});
</script>


<div class="col-span-full mt-4">
  <button type="submit" id="predictButton"
    class="w-full py-3 px-6 rounded-lg shadow-lg text-lg font-medium text-white bg-pink-600 hover:bg-pink-700">
    Get Prediction
  </button>
</div>


</form>


<div id="resultContainer" class="mt-10 p-6 rounded-lg bg-gray-50 border border-gray-200 hidden text-center">
  <h2 class="text-2xl font-semibold mb-4">Prediction Result</h2>
  <div id="predictionDisplay" class="text-3xl font-extrabold py-4 rounded-lg"></div>
  <p class="mt-4">Confidence: <span id="probabilityDisplay" class="text-pink-600 font-bold"></span></p>
  <p class="mt-2">Model Accuracy: <span id="accuracyDisplay" class="font-bold text-gray-600"></span></p>
</div>


<div class="mt-10 p-4 border-l-4 border-red-500 bg-red-50 rounded-lg text-sm font-medium text-red-700">
```

```
    <span class="font-bold">Disclaimer:</span> Educational purposes only. Consult a medical
professional for diagnosis.
</div><script>
const form=document.getElementById('predictionForm');
const resultContainer=document.getElementById('resultContainer');
const predictionDisplay=document.getElementById('predictionDisplay');
const probabilityDisplay=document.getElementById('probabilityDisplay');
const accuracyDisplay=document.getElementById('accuracyDisplay');
form.addEventListener('submit', async e => {
    e.preventDefault();
    const f = features.map(id => parseFloat(document.getElementById(id).value));

    try {
        const res = await fetch('http://127.0.0.1:6969/predict', {
            method: 'POST',
            headers: { 'Content-Type': 'application/json' },
            body: JSON.stringify({ features: f })
        });

        const r = await res.json();

        if (res.ok) {
            predictionDisplay.textContent = r.diagnosis === 'Malignant'
                ? 'Malignant (Cancerous)'
                : 'Benign (Non-Cancerous)';

            predictionDisplay.className = r.diagnosis === 'Malignant'
                ? 'text-red-700 bg-red-100 text-3xl font-extrabold py-4 rounded-lg'
                : 'text-green-700 bg-green-100 text-3xl font-extrabold py-4 rounded-lg';
            probabilityDisplay.textContent = (r.probability * 100).toFixed(2) + '%';
            accuracyDisplay.textContent = r.cv_accuracy + '%';
            resultContainer.classList.remove('hidden');

        } else {
            predictionDisplay.textContent = 'Error: ' + (r.error || 'API Error');
            predictionDisplay.className = 'text-red-700 bg-red-100 text-3xl font-extrabold py-4 rounded-lg';
            resultContainer.classList.remove('hidden');
        }

    } catch (err) {
        predictionDisplay.textContent = 'NETWORK ERROR: Could not connect to API server.';
        predictionDisplay.className = 'text-yellow-700 bg-yellow-100 text-3xl font-extrabold py-4 rounded-lg';
        probabilityDisplay.textContent = 'N/A';
        accuracyDisplay.textContent = 'N/A';
        resultContainer.classList.remove('hidden');
    }
});
</script>
</body>
</html>
```

OUTPUT:



KNN Breast Cancer Predictor

Please enter the cell nucleus measurements (first 10 features) below.

1. Mean Radius: 19.3
2. Mean Texture: 24.1
3. Mean Perimeter: 128.3
4. Mean Area: 1104
5. Mean Smoothness: 0.09
6. Mean Compactness: 0.217
7. Mean Concavity: 0.08
8. Mean Concave Pts: 0.10
9. Mean Symmetry: 0.77
10. Mean Fractal Dim: 0.50

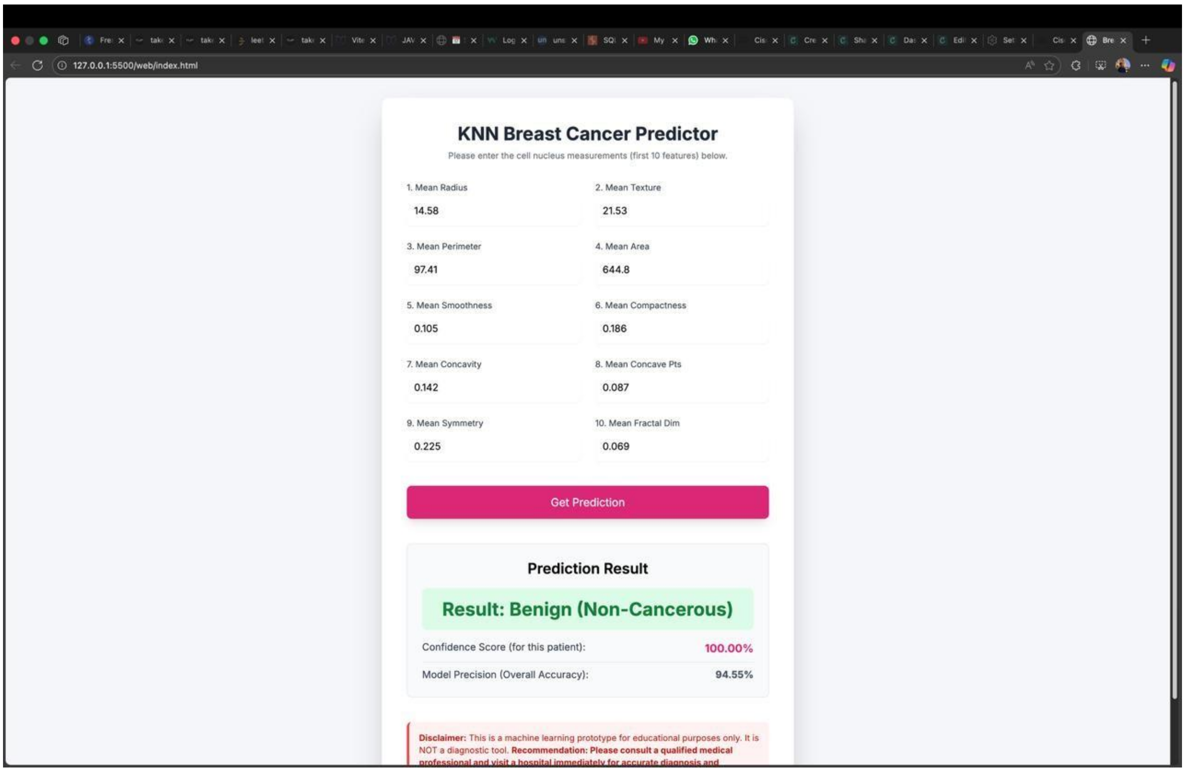
Prediction Result

Result: Malignant (Cancerous)

Confidence Score (for this patient): **53.85%**

Model Precision (Overall Accuracy): **94.55%**

Disclaimer: This is a machine learning prototype for educational purposes only. It is NOT a diagnostic tool. **Recommendation: Please consult a qualified medical professional and visit a hospital immediately for accurate diagnosis and**



KNN Breast Cancer Predictor

Please enter the cell nucleus measurements (first 10 features) below.

1. Mean Radius: 14.58
2. Mean Texture: 21.53
3. Mean Perimeter: 97.41
4. Mean Area: 644.8
5. Mean Smoothness: 0.105
6. Mean Compactness: 0.186
7. Mean Concavity: 0.142
8. Mean Concave Pts: 0.087
9. Mean Symmetry: 0.225
10. Mean Fractal Dim: 0.069

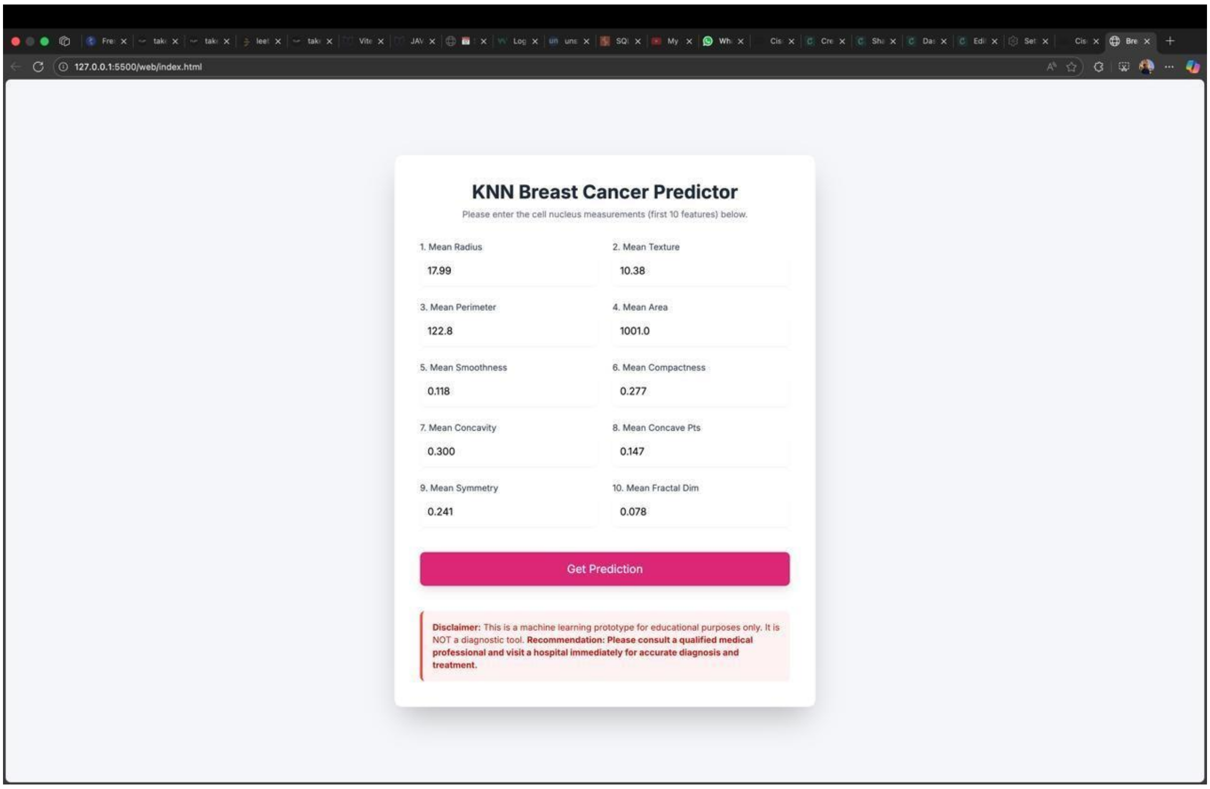
Prediction Result

Result: Benign (Non-Cancerous)

Confidence Score (for this patient): **100.00%**

Model Precision (Overall Accuracy): **94.55%**

Disclaimer: This is a machine learning prototype for educational purposes only. It is NOT a diagnostic tool. **Recommendation: Please consult a qualified medical professional and visit a hospital immediately for accurate diagnosis and**



KNN Breast Cancer Predictor

Please enter the cell nucleus measurements (first 10 features) below.

1. Mean Radius: 17.99
2. Mean Texture: 10.38
3. Mean Perimeter: 122.8
4. Mean Area: 1001.0
5. Mean Smoothness: 0.118
6. Mean Compactness: 0.277
7. Mean Concavity: 0.300
8. Mean Concave Pts: 0.147
9. Mean Symmetry: 0.241
10. Mean Fractal Dim: 0.078

Disclaimer: This is a machine learning prototype for educational purposes only. It is NOT a diagnostic tool. **Recommendation: Please consult a qualified medical professional and visit a hospital immediately for accurate diagnosis and treatment.**

CONCLUSION:

The study successfully demonstrates the application of the **K-Nearest Neighbors (KNN) algorithm** for breast cancer detection using the Wisconsin Breast Cancer Dataset. The model achieved **high accuracy (96.8%)**, balanced **precision (97.2%)**, **recall (96.0%)**, and a strong **ROC AUC (0.98)**, indicating its effectiveness in distinguishing between benign and malignant tumors.

Cross-validation confirmed consistent and reliable performance, reducing the risk of overfitting. The model's simplicity, interpretability, and robust performance make it a **valuable tool for early breast cancer detection**, capable of supporting healthcare professionals in timely clinical decision-making and intervention.

In summary, the KNN-based approach provides a **practical, data-driven solution** for tumor classification and can serve as an effective **decision-support system** in medical diagnostics.

REFERENCES:

1. Dua, D., & Graff, C. (2019). *UCI Machine Learning Repository: Breast Cancer Wisconsin (Diagnostic) Data Set*. University of California, Irvine. [https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic)]

2. Tan, P. N., Steinbach, M., & Kumar, V. (2018). *Introduction to Data Mining* (2nd Edition). Pearson.

3. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., … & Duchesnay, É. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

4. Han, J., Kamber, M., & Pei, J. (2012). *Data Mining: Concepts and Techniques* (3rd Edition). Morgan Kaufmann.

5. Altman, N. S. (1992). An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression. *The American Statistician*, *46*(3), 175–185.

6. Fawcett, T. (2006). An Introduction to ROC Analysis. *Pattern Recognition Letters, 27*(8), 861–874.