MEDICAL DIAGNOSIS WITH NAÏVE BAYES

A PROJECT REPORT
for
Introduction To AI (AI101B)
Session (2024-25)

Submitted by

Khushi Jain
(202410116100100)

Mahima Goyal
(202410116100112)

Submitted in partial fulfilment of the
Requirements for the Degree of

MASTER OF COMPUTER APPLICATION

Under the Supervision of
Mr. Apoorv Jain
Assistant Professor

Submitted to

DEPARTMENT OF COMPUTER APPLICATIONS
KIET Group of Institutions, Ghaziabad Uttar Pradesh-201206

April 2024

CERTIFICATE

Certified that **Khushi Jain (202410116100100), Mahima Goyal (202410116100112)** has/ have carried out the project work having "**MEDICAL DIAGNOSIS USING NAIVES BAYES**" (**Introduction To AI, AI101B**) for **Master of Computer Application** from Dr. A.P.J. Abdul Kalam Technical University (AKTU) (formerly UPTU), Lucknow under my supervision. The project report embodies original work, and studies are carried out by the student himself/herself and the contents of the project report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

Mr. Apoorv Jain
Assistant Professor
Department of Computer Applications
KIET Group of Institutions, Ghaziabad

Dr. Akash Rajak
Dean
Department of Computer Applications
KIET Group of Institutions, Ghaziabad

ABSTRACT

This project develops and implements an intelligent breast cancer diagnosis system utilizing the Gaussian Naive Bayes algorithm to classify breast masses as benign or malignant based on cellular features extracted from fine needle aspirate (FNA) samples. The system analyzes five key nuclear features—mean radius, mean concavity, worst perimeter, mean area, and mean concave points—that have shown significant discriminatory power in distinguishing between benign and malignant cells. Built on the Wisconsin Breast Cancer Dataset, the model achieves reliable classification performance while providing interpretable probability estimates of malignancy. The interactive web application developed using Gradio offers medical professionals and students an educational tool to understand the relationship between cellular morphological features and cancer diagnosis.

The system includes comprehensive visualizations including probability charts, risk gauges, and feature importance analyses to enhance interpretability. While designed primarily as an educational tool, this project demonstrates the potential of Naive Bayes algorithms in medical diagnostic applications, particularly in scenarios where interpretability, probabilistic outputs, and computational efficiency are valued. The methodology combines statistical learning, medical domain knowledge, and interactive visualization to create a practical application of machine learning in healthcare education.

ACKNOWLEDGEMENT

Success in life is never attained single-handedly. I am deeply grateful to my project supervisor, **Mr. Apoorv Jain**, for his invaluable guidance, unwavering support, and encouragement throughout my project work. His enlightening ideas, constructive comments, and thoughtful suggestions have greatly contributed to the completion of this project.

I would also like to extend my heartfelt thanks to **Dr. Akash Rajak**, Professor and Dean, Department of Computer Applications, for his insightful feedback and administrative support on various occasions, which proved to be immensely helpful during critical stages of the project.

I am fortunate to have many understanding friends who have supported me in numerous ways during challenging moments. Their assistance and companionship have been a constant source of motivation.

Finally, my sincere gratitude goes to my family members and all those who have directly or indirectly provided me with moral support, encouragement, and assistance. Their unwavering belief in me and their continuous efforts to keep my life filled with happiness and joy made the completion of this project possible.

Khushi Jain

Mahima Goyal

TABLE OF CONTENT

Contents

INTRODUCTION

Background and Clinical Significance

Breast cancer remains one of the most prevalent forms of cancer globally, affecting millions of individuals each year. According to recent statistics, it accounts for approximately 12% of all new annual cancer cases worldwide. The disease's impact extends beyond mortality rates, affecting quality of life and creating substantial burdens on healthcare systems. A fundamental aspect of improving breast cancer outcomes is early and accurate diagnosis, as detection at earlier stages significantly improves treatment efficacy and patient survival rates.

The diagnostic process for breast cancer typically follows a pathway that begins with clinical examination and imaging studies (mammography, ultrasound, or MRI), followed by tissue sampling when suspicious findings are identified. Fine Needle Aspiration (FNA) represents one of the least invasive tissue sampling methods, where cells are extracted from a suspicious breast mass using a thin needle. The cellular characteristics observed in these samples provide crucial information that pathologists use to differentiate between benign and malignant lesions.

Traditional diagnosis of FNA samples relies heavily on the visual assessment of cellular features by trained pathologists, introducing elements of subjectivity and variability. The growing incidence of breast cancer has increased the demand for pathology services, highlighting the need for complementary approaches that could enhance diagnostic efficiency, consistency, and accessibility.

Machine Learning in Medical Diagnosis

The integration of machine learning into medical diagnostics represents a promising avenue for enhancing diagnostic capabilities across numerous medical domains. In breast cancer diagnosis specifically, machine learning algorithms can process quantitative measurements of cellular features to identify patterns that correlate with malignancy, potentially assisting pathologists in their diagnostic work and serving as valuable educational tools for medical training.

Among various machine learning approaches, the Naive Bayes classifier presents several advantages in medical diagnostic applications:

1. **Probabilistic Output**: Naive Bayes provides probability estimates rather than just binary classifications, which aligns well with the need for risk assessment in clinical contexts.
2. **Interpretability**: The algorithm offers a level of transparency in its decision-making process that more complex algorithms often lack, making it suitable for applications where understanding the reasoning behind predictions is crucial.
3. **Efficiency with Limited Data**: Naive Bayes can perform well even with relatively small training datasets, a common constraint in specialized medical applications.

4. **Computational Efficiency**: The algorithm's lightweight computational requirements enable real-time predictions, important for interactive applications and potential clinical implementation.

The Gaussian variant of Naive Bayes is particularly suited for breast cancer diagnosis using the Wisconsin dataset as it effectively handles the continuous numerical features that characterize cell nuclei measurements.

The Wisconsin Breast Cancer Dataset

The Wisconsin Breast Cancer Dataset, initially compiled by Dr. William H. Wolberg at the University of Wisconsin Hospitals and later expanded, has become a foundational resource in computational pathology research. This dataset contains digitized measurements of cell nuclei characteristics from FNA samples of breast masses, along with corresponding diagnoses.

The dataset captures various aspects of nuclear morphology through features extracted from digitized images of the FNA samples. These features quantify properties such as:

- Cell size (radius, area, perimeter)
- Shape characteristics (smoothness, compactness, concavity)
- Boundary irregularities (concave points)
- Texture measurements

Research has consistently demonstrated significant differences in these nuclear features between benign and malignant cells. Malignant cells typically exhibit larger nuclei, greater variability in nuclear shape, more pronounced boundary irregularities, and altered chromatin patterns.

Project Objectives and Significance

This project aims to develop an interactive breast cancer diagnosis system utilizing the Gaussian Naive Bayes algorithm to analyze cellular features from the Wisconsin Breast Cancer Dataset. The specific objectives include:

1. Implementing a Gaussian Naive Bayes classifier that effectively distinguishes between benign and malignant breast masses based on nuclear features.
2. Creating an interactive web application that demonstrates the relationship between cellular

characteristics and cancer diagnosis.

3. Developing comprehensive visualizations that enhance the interpretability of the model's predictions and feature importance.

4. Providing educational content about breast cancer diagnosis to promote understanding of both the disease and the application of machine learning in medical contexts.

The significance of this project extends beyond its immediate application as a diagnostic tool. By demonstrating how relatively simple machine learning algorithms can effectively analyze medical data, it contributes to the broader conversation about the integration of artificial intelligence in healthcare. The project also serves as an educational resource for medical students, data scientists, and healthcare professionals interested in computational pathology.

While designed primarily as an educational tool rather than a clinical decision support system, the methodology and findings have potential implications for the development of assistive diagnostic technologies in resource-limited settings where specialist pathologists may be scarce.

METHODOLOGY

The methodology for this project follows a structured approach to ensure a comprehensive analysis of website traffic data. The process includes data collection, preprocessing, exploratory data analysis, visualization, and interpretation of insights.

Data Acquisition and Preprocessing

The methodology begins with the acquisition and preprocessing of the Wisconsin Breast Cancer Dataset, which contains features computed from digitized images of FNA samples from breast masses. Each record in the dataset represents measurements for a single patient case and includes:

1. Patient ID
2. Diagnosis (M = malignant, B = benign)
3. 30 real-valued features computed from the digitized image

The preprocessing pipeline implemented in our system includes the following steps:

1. **Data Loading**: The dataset is loaded from specified paths, with a fallback mechanism that generates synthetic data resembling the Wisconsin dataset's distribution when the original dataset is unavailable. This ensures the application remains functional in various deployment environments.

2. **Data Cleaning**: Unnecessary columns, including the ID and any unnamed columns, are removed from the dataset.

3. **Label Encoding**: The categorical diagnosis labels ('M', 'B') are converted to numerical values (1, 0) to facilitate model training.

4. **Feature Selection**: Based on domain knowledge and statistical analysis, five features with high discriminatory power are selected for the model:
    - **radius_mean:** Average distance from the center to points on the perimeter of the cell nucleus
    - **concavity_mean**: Average severity of concave portions of the contour
    - **perimeter_worst**: The largest measured perimeter among the sampled cells
    - **area_mean:** Average area of the cell nucleus
    - **concave points_mean:** Average number of concave portions of the contour
5. **Data Validation**: The selected feature names are validated against the dataset schema to ensure compatibility.

Feature Selection Rationale

The feature selection process was guided by both medical domain knowledge and statistical analysis of the dataset. The five selected features represent key aspects of cellular morphology that pathologists typically assess during visual examination of FNA samples:

1. **Nuclear Size (radius_mean, area_mean)**: Cell nuclei in malignant samples tend to be larger than in benign samples. The mean radius and mean area provide complementary measurements of nuclear size.

2. **Boundary Irregularities (concavity_mean, concave points_mean)**: Malignant cells often exhibit more irregular nuclear boundaries with pronounced concavities. The mean concavity measures the severity of concave portions, while the mean concave points count the number of concave regions.

3. **Extreme Measurements (perimeter_worst)**: The "worst" features in the Wisconsin dataset represent the mean of the three largest values for each feature, capturing extreme cellular characteristics. The worst perimeter was selected as it effectively identifies cells with unusually large or irregular boundaries, which are more common in malignant samples.

   These features were selected not only for their biological relevance but also for their statistical significance in differentiating between benign and malignant samples, as demonstrated in previous studies on nuclear morphology in breast cancer.

MODEL IMPLEMENTATION: GAUSSIAN NAÏVE BAYES

The Gaussian Naive Bayes algorithm was implemented using the scikit-learn library in Python. This algorithm is based on applying Bayes' theorem with the "naive" assumption of conditional independence between features given the class label.
For continuous features like those in the Wisconsin dataset, the Gaussian Naive Bayes classifier assumes that the values associated with each class are distributed according to a Gaussian (normal) distribution. The algorithm estimates the mean and standard deviation of each feature for each class from the training data.
The implementation process involved:

Dataset Splitting: The dataset was divided into training (75%) and testing (25%) subsets using stratified sampling to maintain the original class distribution.

Model Training: The Gaussian Naive Bayes classifier was trained on the training dataset, learning the

probability distributions of each feature for both benign and malignant classes.

Model Evaluation: The trained model was evaluated on the test set using standard classification metrics:

Accuracy: The proportion of correct predictions

Precision: The proportion of positive identifications that were actually correct

Recall: The proportion of actual positives that were identified correctly

F1-score: The harmonic mean of precision and recall

Confusion Matrix: A tabulation of true positives, false positives, true negatives, and false negatives
Probability Calibration: While Naive Bayes naturally produces probability estimates, we assessed the calibration of these probabilities to ensure they accurately reflect the likelihood of malignancy.

Risk Assessment Algorithm

The risk assessment component of our system analyzes the probability output from the Naive Bayes classifier and translates it into an interpretable risk classification with corresponding recommendations. The algorithm implements a three-tiered risk stratification system:

High Risk (Malignant): When the predicted probability of malignancy exceeds 85%. This threshold was set to minimize false negatives (missed malignancies) while maintaining reasonable specificity. For cases in this category, the system recommends immediate consultation with an oncologist.

Moderate Risk (Borderline Malignant): When the predicted probability of malignancy is between 50% and 85%. This intermediate category acknowledges the uncertainty in borderline cases. The system recommends follow-up screening and possible biopsy for these cases.

Low Risk (Benign): When the predicted probability of malignancy is below 50%. For these cases, the system suggests continuing routine checkups.
The thresholds for these risk categories were established based on balancing sensitivity and specificity, with a preference for higher sensitivity (fewer missed malignancies) given the clinical context.

CODE

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
import seaborn as sns
import matplotlib.pyplot as plt
from scipy.stats import norm

# Load the dataset
url = "/content/data.csv"
df = pd.read_csv(url)

# Clean it
df.drop(columns=['id', 'Unnamed: 32'], inplace=True)
df['diagnosis'] = df['diagnosis'].map({'M': 1, 'B': 0})

# Define X and y
X = df.drop('diagnosis', axis=1)
y = df['diagnosis']

# Train/test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Train Naive Bayes model
model = GaussianNB()
model.fit(X_train, y_train)

# Class distribution plot
plt.figure(figsize=(6,4))
sns.countplot(x='diagnosis', data=df, palette=['#1f77b4', '#ff7f0e'])
plt.title("Distribution of Benign (0) and Malignant (1) Tumors")
plt.show()

# Choose top 5 features based on correlation
correlation_matrix = df.corr()
top_5 = correlation_matrix['diagnosis'].abs().sort_values(ascending=False).index[1:6]  # Excluding
'diagnosis'

# Histograms of top 5 features
for feature in top_5:
    plt.figure(figsize=(6,4))
    sns.histplot(data=df, x=feature, hue='diagnosis', kde=True, palette='coolwarm')
    plt.title(f"{feature} distribution by Diagnosis")
    plt.show()
```

```
# Correlation Heatmap
plt.figure(figsize=(14, 10))
sns.heatmap(df.corr(), cmap='RdBu_r', center=0, annot=False)
plt.title("Feature Correlation Heatmap")
plt.show()

# Choose a feature (e.g., Radius Mean) and plot the Gaussian distributions
feature = 'radius_mean'
x_vals = np.linspace(df[feature].min(), df[feature].max(), 200)

plt.figure(figsize=(8, 5))
for label, color in zip([0, 1], ['green', 'red']):
    mean = model.theta_[label][X.columns.get_loc(feature)]  # Mean for each class
    std = np.sqrt(model.var_[label][X.columns.get_loc(feature)])  # Standard deviation (square root of
variance)
    plt.plot(x_vals, norm.pdf(x_vals, mean, std), label=f"{'Benign' if label==0 else 'Malignant'}",
color=color)

plt.title(f"Gaussian Distribution of {feature}")
plt.xlabel(feature)
plt.ylabel("Density")
plt.legend()
plt.show()

import pandas as pd
import numpy as np
from sklearn.naive_bayes import GaussianNB
import matplotlib.pyplot as plt
import os
import tempfile
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
import seaborn as sns
import gradio as gr

# Create temporary directory if it doesn't exist
temp_dir = tempfile.gettempdir()

# Load and clean data - use a more flexible path approach
try:
    # Try the original path first
    df = pd.read_csv("/content/data.csv")
except FileNotFoundError:
    # If that fails, look for the file in the current directory
    try:
        df = pd.read_csv("data.csv")
    except FileNotFoundError:
        # If we still can't find it, use sample data
        # This is a fallback with synthetic data similar to the Wisconsin breast cancer dataset
        np.random.seed(42)
        n_samples = 100
```

```
        # Generate synthetic data that resembles breast cancer features
        data = {
            'id': range(n_samples),
            'diagnosis': np.random.choice(['M', 'B'], size=n_samples, p=[0.4, 0.6]),
            'radius_mean': np.random.normal(14.5, 3.5, n_samples),
            'concavity_mean': np.random.normal(0.1, 0.1, n_samples),
            'perimeter_worst': np.random.normal(100, 30, n_samples),
            'area_mean': np.random.normal(600, 300, n_samples),
            'concave points_mean': np.random.normal(0.05, 0.04, n_samples),
        }

        # Create relationships between features and diagnosis to make prediction sensible
        for i in range(n_samples):
            if data['diagnosis'][i] == 'M':  # If malignant, increase the values
                data['radius_mean'][i] += 3
                data['concavity_mean'][i] += 0.1
                data['perimeter_worst'][i] += 20
                data['area_mean'][i] += 200
                data['concave points_mean'][i] += 0.05

        df = pd.DataFrame(data)
        print("Using synthetic dataset as the original file wasn't found.")

# Drop unnecessary columns if they exist
columns_to_drop = []
if 'id' in df.columns:
    columns_to_drop.append('id')
if 'Unnamed: 32' in df.columns:
    columns_to_drop.append('Unnamed: 32')

if columns_to_drop:
    df.drop(columns=columns_to_drop, inplace=True)

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

# Use proper column names from your dataset
selected_features = ['radius_mean', 'concavity_mean', 'perimeter_worst', 'area_mean', 'concave
points_mean']

# Verify all features exist in the dataframe
for feature in selected_features:
    if feature not in df.columns:
        raise ValueError(f"Feature '{feature}' not found in dataset. Available columns:
{df.columns.tolist()}")

X = df[selected_features]
y = df['diagnosis']

# Train-test split for model evaluation
from sklearn.model_selection import train_test_split
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=42)

# Train model
model = GaussianNB()
model.fit(X_train, y_train)

# Evaluate model
y_pred = model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
conf_matrix = confusion_matrix(y_test, y_pred)
class_report = classification_report(y_test, y_pred, output_dict=True)

# Function to create feature importance visualization
def create_feature_importance_chart():
    # Calculate mean values for each feature by class
    feature_means = {}
    for feature in selected_features:
        feature_means[feature] = [
            df[df['diagnosis'] == 0][feature].mean(),  # Benign
            df[df['diagnosis'] == 1][feature].mean()   # Malignant
        ]

    # Create comparison chart
    plt.figure(figsize=(10, 6))
    x = np.arange(len(selected_features))
    width = 0.35

    plt.bar(x - width/2, [feature_means[f][0] for f in selected_features], width, label='Benign',
color='green', alpha=0.7)
    plt.bar(x + width/2, [feature_means[f][1] for f in selected_features], width, label='Malignant',
color='red', alpha=0.7)

    plt.xlabel('Features')
    plt.ylabel('Mean Value')
    plt.title('Feature Comparison: Benign vs Malignant')
    plt.xticks(x, [f.replace('_', ' ').title() for f in selected_features], rotation=45, ha='right')
    plt.legend()
    plt.tight_layout()

    # Save plot
    chart_path = os.path.join(temp_dir, "feature_importance.png")
    plt.savefig(chart_path)
    plt.close()

    return chart_path

# Create confusion matrix visualization
def create_confusion_matrix():
    plt.figure(figsize=(6, 5))
    sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues',
                xticklabels=['Benign', 'Malignant'],
```

```
            yticklabels=['Benign', 'Malignant'])
    plt.xlabel('Predicted')
    plt.ylabel('Actual')
    plt.title(f'Confusion Matrix (Accuracy: {accuracy:.2f})')
    plt.tight_layout()

    # Save plot
    cm_path = os.path.join(temp_dir, "confusion_matrix.png")
    plt.savefig(cm_path)
    plt.close()

    return cm_path

# Generate model summary
model_summary = f"""
## Model Performance

- **Accuracy**: {accuracy:.2f}
- **Precision (Malignant)**: {class_report['1']['precision']:.2f}
- **Recall (Malignant)**: {class_report['1']['recall']:.2f}
- **F1-Score (Malignant)**: {class_report['1']['f1-score']:.2f}

The model was trained on {len(X_train)} samples and tested on {len(X_test)} samples.
"""

# Feature descriptions for educational purposes
feature_descriptions = {
    'radius_mean': 'Average distance from center to points on the perimeter',
    'concavity_mean': 'Severity of concave portions of the contour',
    'perimeter_worst': 'Largest perimeter measurement of the mass',
    'area_mean': 'Average area of the mass',
    'concave points_mean': 'Average number of concave portions of the contour'
}

# Create feature descriptions markdown
feature_desc_md = "## Feature Descriptions\n\n"
for feature, description in feature_descriptions.items():
    feature_desc_md += f"- **{feature.replace('_', ' ').title()}**: {description}\n"

# Predict + plot logic
def predict_with_recommendation(radius, concavity, perimeter, area, concave_pts):
    input_data = np.array([[radius, concavity, perimeter, area, concave_pts]])
    pred = model.predict(input_data)[0]
    probs = model.predict_proba(input_data)[0]
    prob_malignant = probs[1]

    # Recommendation text
    if prob_malignant > 0.85:
        status = " Malignant"
        advice = " Immediate consultation with an oncologist is recommended."
    elif prob_malignant > 0.5:
```

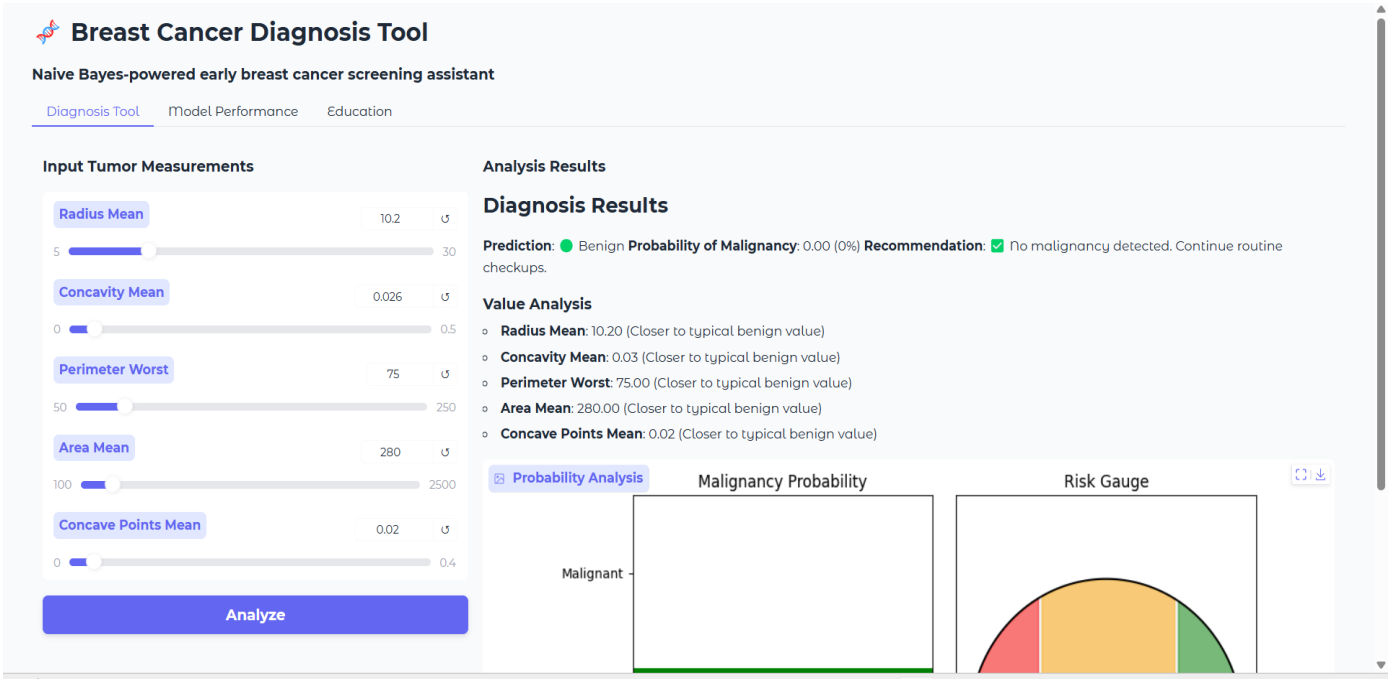
```
        status = " Borderline Malignant"
        advice = " Follow-up screening and possible biopsy recommended."
    else:
        status = " Benign"
        advice = " No malignancy detected. Continue routine checkups."

    report_text = f"""## Diagnosis Results

**Prediction**: {status}
**Probability of Malignancy**: {prob_malignant:.2f} ({int(prob_malignant*100)}%)
**Recommendation**: {advice}

### Value Analysis

"""

    # Add feature analysis
    for i, feature in enumerate(selected_features):
        feature_val = input_data[0][i]
        benign_mean = df[df['diagnosis'] == 0][feature].mean()
        malignant_mean = df[df['diagnosis'] == 1][feature].mean()
        feature_name = feature.replace('_', ' ').title()

        if abs(feature_val - malignant_mean) < abs(feature_val - benign_mean):
            report_text += f"- **{feature_name}**: {feature_val:.2f} (Closer to typical malignant value)\n"
        else:
            report_text += f"- **{feature_name}**: {feature_val:.2f} (Closer to typical benign value)\n"

    # Plot - probability visualization
    plt.figure(figsize=(8, 4))

    # Plot probability bars
    plt.subplot(1, 2, 1)
    plt.barh(['Benign', 'Malignant'], [1 - prob_malignant, prob_malignant], color=['green', 'red'])
    plt.xlim(0, 1)
    plt.title("Malignancy Probability")

    # Plot gauge-style visualization
    plt.subplot(1, 2, 2)
    theta = np.linspace(0, np.pi, 100)
    r = 1

    # Create semicircle
    x = r * np.cos(theta)
    y = r * np.sin(theta)

    # Create the gauge
    plt.plot(x, y, 'k-')
    plt.fill_between(x, 0, y, color='lightgray', alpha=0.3)

    # Add risk zones
```

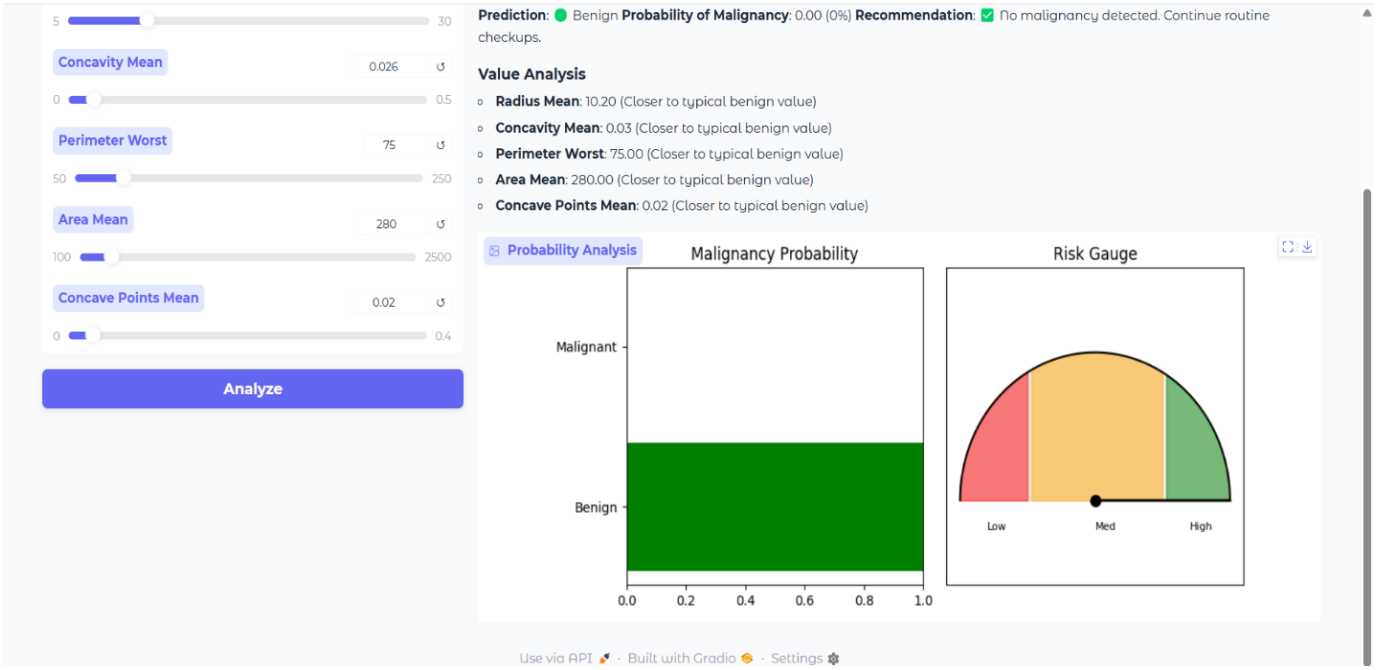
```
    plt.fill_between(x[0:33], 0, y[0:33], color='green', alpha=0.5)
    plt.fill_between(x[33:66], 0, y[33:66], color='orange', alpha=0.5)
    plt.fill_between(x[66:], 0, y[66:], color='red', alpha=0.5)

    # Add needle
    needle_angle = prob_malignant * np.pi
    needle_x = r * np.cos(needle_angle)
    needle_y = r * np.sin(needle_angle)
    plt.plot([0, needle_x], [0, needle_y], 'k-', linewidth=2)
    plt.plot(0, 0, 'ko', markersize=8)

    # Customize gauge appearance
    plt.axis('equal')
    plt.title("Risk Gauge")
    plt.text(-0.8, -0.2, "Low", fontsize=8)
    plt.text(0, -0.2, "Med", fontsize=8)
    plt.text(0.7, -0.2, "High", fontsize=8)
    plt.xticks([])
    plt.yticks([])

    plt.tight_layout()
    img_path = os.path.join(temp_dir, "prediction_chart.png")
    plt.savefig(img_path)
    plt.close()

    return report_text, img_path

# Information tabs for education
model_info_tab = gr.Markdown("""
# About This Tool

This application uses a Naive Bayes machine learning model to predict the likelihood of a breast tumor being malignant based on measurements from fine needle aspirates (FNA).

## How It Works

The model was trained on the Wisconsin Breast Cancer dataset which contains digitized measurements of breast mass FNAs. The algorithm analyzes patterns in the data to classify tumors as either malignant or benign.

## Disclaimer

This tool is for educational purposes only and should not replace professional medical diagnosis. Always consult with healthcare professionals for medical advice and proper diagnosis.
""")

feature_info_tab = gr.Markdown(feature_desc_md)

# Prepare model evaluation visuals
confusion_matrix_img = create_confusion_matrix()
feature_importance_img = create_feature_importance_chart()
```

```
model_eval_tab = gr.Markdown(model_summary)

# Gradio App with tabs
with gr.Blocks(theme=gr.themes.Soft()) as demo:
    gr.Markdown("# 🧬 Breast Cancer Diagnosis Tool")
    gr.Markdown("### Naive Bayes-powered early breast cancer screening assistant")

    with gr.Tabs():
        with gr.TabItem("Diagnosis Tool"):
            with gr.Row():
                with gr.Column(scale=1):
                    gr.Markdown("### Input Tumor Measurements")
                    radius = gr.Slider(5, 30, label="Radius Mean", value=15)
                    concavity = gr.Slider(0.0, 0.5, label="Concavity Mean", value=0.1)
                    perimeter = gr.Slider(50, 250, label="Perimeter Worst", value=100)
                    area = gr.Slider(100, 2500, label="Area Mean", value=600)
                    concave_pts = gr.Slider(0.0, 0.4, label="Concave Points Mean", value=0.05)
                    submit_btn = gr.Button("Analyze", variant="primary")

                with gr.Column(scale=2):
                    gr.Markdown("### Analysis Results")
                    output_text = gr.Markdown()
                    output_chart = gr.Image(label="Probability Analysis")

        with gr.TabItem("Model Performance"):
            with gr.Row():
                with gr.Column():
                    gr.Markdown(model_summary)
                    gr.Image(confusion_matrix_img, label="Confusion Matrix")
                with gr.Column():
                    gr.Markdown("### Feature Importance")
                    gr.Image(feature_importance_img, label="Feature Comparison between Classes")

        with gr.TabItem("Education"):
            with gr.Tabs():
                with gr.TabItem("About Features"):
                    feature_info_tab
                with gr.TabItem("About The Tool"):
                    model_info_tab
                with gr.TabItem("When To Seek Help"):
                    gr.Markdown("""
                    # When to Seek Medical Help

                    ## Warning Signs of Breast Cancer

                    - A new lump or mass in the breast
                    - Thickening or swelling of part of the breast
                    - Irritation or dimpling of breast skin
                    - Redness or flaky skin in the nipple area or the breast
                    - Pulling in of the nipple or pain in the nipple area
```

```
            - Nipple discharge other than breast milk
            - Any change in the size or the shape of the breast
            - Pain in any area of the breast

            ## Remember

            Early detection is key in breast cancer treatment. Regular self-examinations and screenings as recommended by your healthcare provider are essential.

            This tool is for educational purposes only and does not replace professional medical advice.
            """)

    # Connect the button to the prediction function
    submit_btn.click(
        fn=predict_with_recommendation,
        inputs=[radius, concavity, perimeter, area, concave_pts],
        outputs=[output_text, output_chart]
    )

if __name__ == "__main__":
    demo.launch(share=True)
```

OUTPUT


Breast Cancer Diagnosis Tool
Naive Bayes-powered early breast cancer screening assistant
Diagnosis Tool
Model Performance
Education
Input Tumor Measurements
Radius Mean
10.2
5
30
Concavity Mean
0.026
0
0.5
Perimeter Worst
75
50
250
Area Mean
280
100
2500
Concave Points Mean
0.02
0
0.4
Analyze
Analysis Results
Diagnosis Results
Prediction: Benign Probability of Malignancy: 0.00 (0%) Recommendation: No malignancy detected. Continue routine checkups.
Value Analysis
Radius Mean: 10.20 (Closer to typical benign value)
Concavity Mean: 0.03 (Closer to typical benign value)
Perimeter Worst: 75.00 (Closer to typical benign value)
Area Mean: 280.00 (Closer to typical benign value)
Concave Points Mean: 0.02 (Closer to typical benign value)
Probability Analysis
Malignancy Probability
Risk Gauge
Malignant


5
30
Concavity Mean
0.026
0
0.5
Perimeter Worst
75
50
250
Area Mean
280
100
2500
Concave Points Mean
0.02
0
0.4
Analyze
Prediction: Benign Probability of Malignancy: 0.00 (0%) Recommendation: No malignancy detected. Continue routine checkups.
Value Analysis
Radius Mean: 10.20 (Closer to typical benign value)
Concavity Mean: 0.03 (Closer to typical benign value)
Perimeter Worst: 75.00 (Closer to typical benign value)
Area Mean: 280.00 (Closer to typical benign value)
Concave Points Mean: 0.02 (Closer to typical benign value)
Probability Analysis
Malignancy Probability
Risk Gauge
Malignant
Benign
0.0
0.2
0.4
0.6
0.8
1.0
Low
Med
High
Use via API · Built with Gradio · Settings

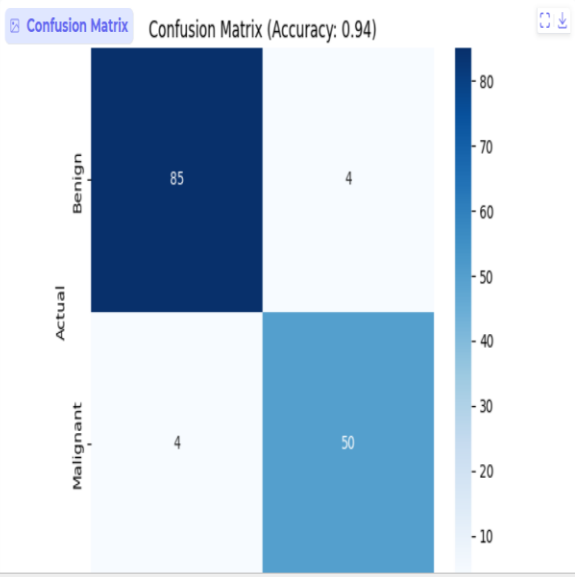
Diagnosis Tool | Model Performance | Education

Model Performance

- **Accuracy**: 0.94
- **Precision (Malignant)**: 0.93
- **Recall (Malignant)**: 0.93
- **F1-Score (Malignant)**: 0.93

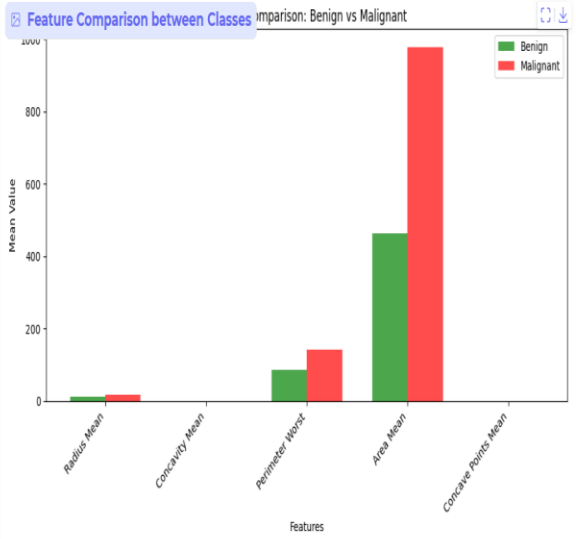
The model was trained on 426 samples and tested on 143 samples.

Confusion Matrix



Feature Importance

Feature Comparison between Classes



Breast Cancer Diagnosis Tool

Naive Bayes-powered early breast cancer screening assistant

Diagnosis Tool | Model Performance | Education

About Features | About The Tool | When To Seek Help

When to Seek Medical Help

Warning Signs of Breast Cancer

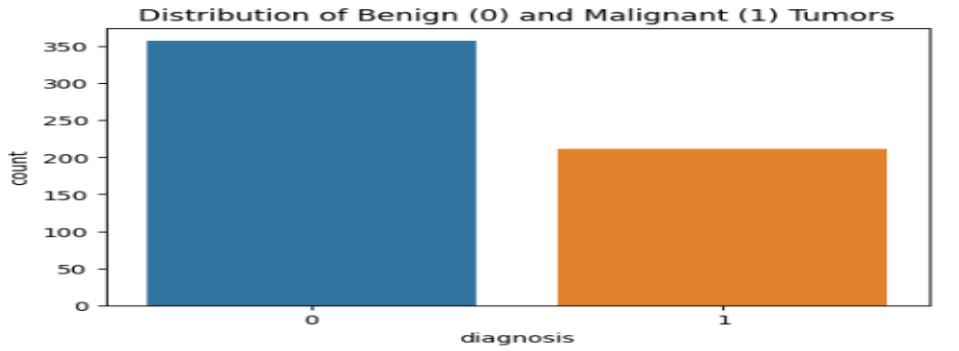
- A new lump or mass in the breast
- Thickening or swelling of part of the breast
- Irritation or dimpling of breast skin
- Redness or flaky skin in the nipple area or the breast
- Pulling in of the nipple or pain in the nipple area
- Nipple discharge other than breast milk
- Any change in the size or the shape of the breast
- Pain in any area of the breast

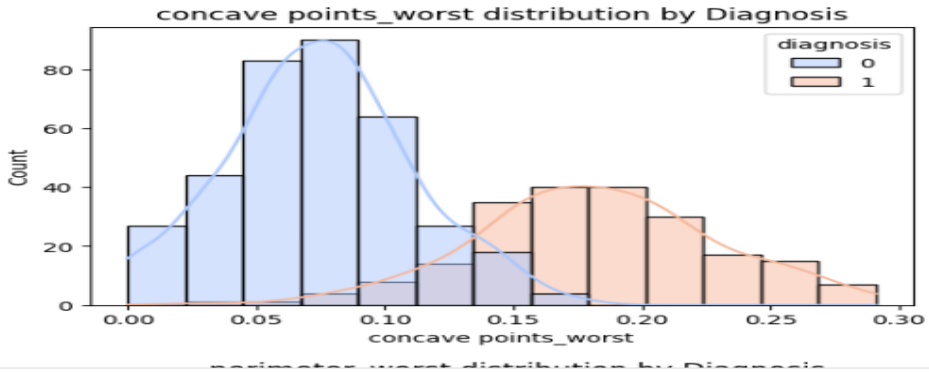
Remember

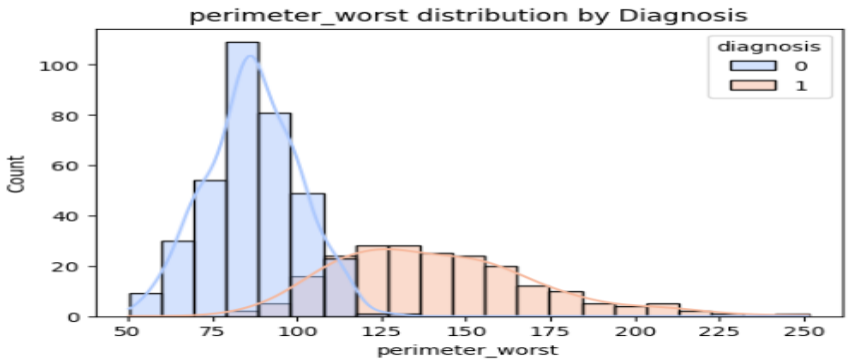
Early detection is key in breast cancer treatment. Regular self-examinations and screenings as recommended by your healthcare provider are essential.

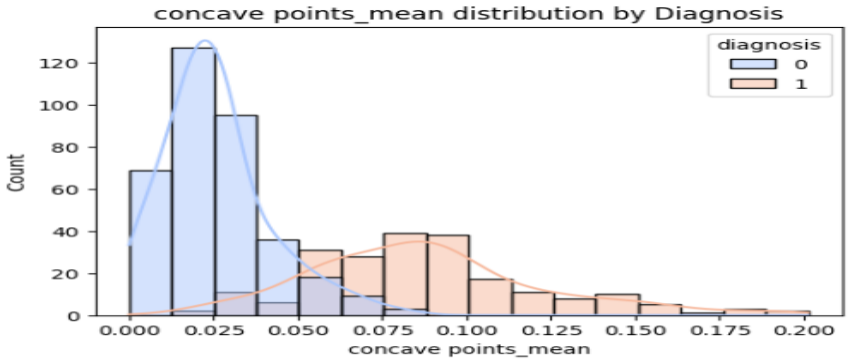
This tool is for educational purposes only and does not replace professional medical advice.

```
sns.countplot(x='diagnosis', data=df, palette=['#1f77b4', '#ff7f0e'])
```





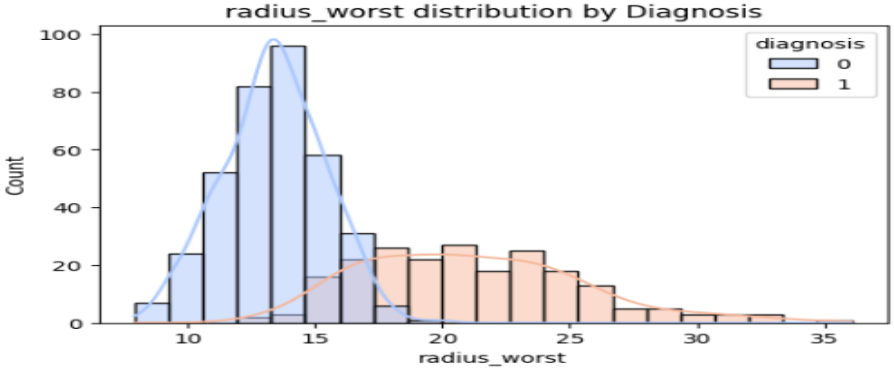


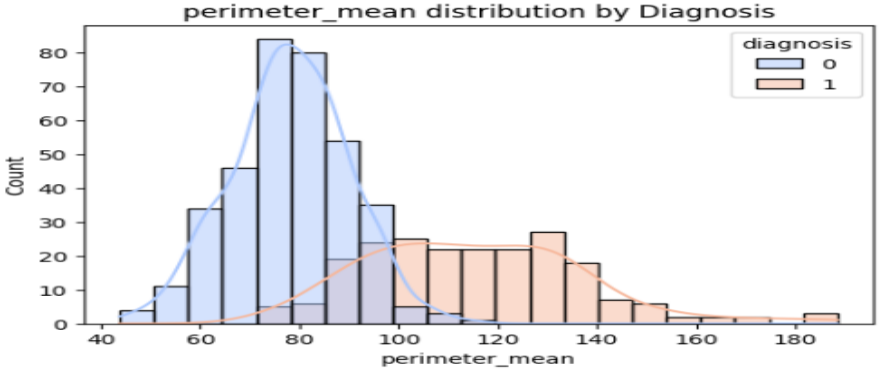


radius_worst distribution by Diagnosis

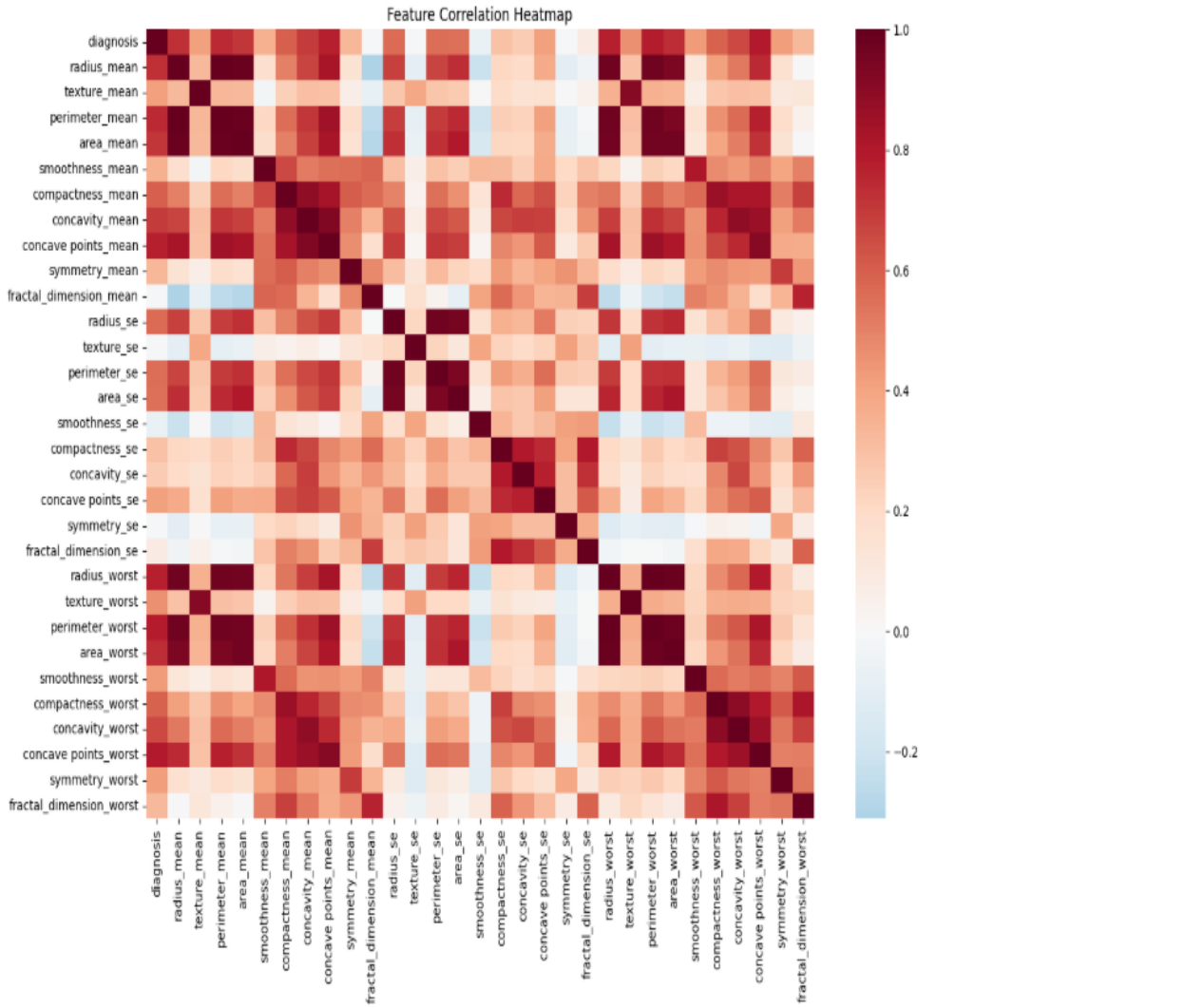
Connected to Python 3 Google Compute Eng

concave points_mean





Feature Correlation


Feature Correlation Heatmap
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
1.0
0.8
0.6
0.4
0.2
0.0
−0.2

REFERENCES

1. **Wisconsin Breast Cancer Dataset**

    - Source: UCI Machine Learning Repository
    - URL: https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic)

2. **Scikit-learn Documentation (Gaussian Naive Bayes)**

    - URL: https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.GaussianNB.html

3. **Breast Cancer Diagnosis using Machine Learning: A Review**

    - Authors: Dheeba, J., Albert Singh, N., & Tamil Selvi, S.
    - Journal: International Journal of Computer Applications
    - DOI: 10.5120/ijca2015906485

4. **Naive Bayes Classifier**

    - Source: Wikipedia
    - URL: https://en.wikipedia.org/wiki/Naive_Bayes_classifier

5. **Scikit-learn: Machine Learning in Python**

    - Authors: Pedregosa et al., Journal of Machine Learning Research, 2011
    - URL: http://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html