# PROBABILITY AND STATISTICS

AROOBA SHEHZADI_231678

SUBMITTED TO :

DR AMMARA CHEEMA

# Objectives

The primary objectives of this project include:

1. **Developing a Machine Learning Model:** To create an optimized tool for predicting whether cancer is benign or malignant based on specific input features.
2. **Application of Probability and Statistics:** To demonstrate real-world applications of probability and statistical techniques in medical diagnosis.
3. **Balancing Class Distribution:** Implementing SMOTE (Synthetic Minority Over-sampling Technique) along with undersampling to handle imbalanced datasets effectively.
4. **Feature Importance Analysis:** Evaluating the significance of different features in predicting cancer types.
5. **Interactive Tool Development:** Creating an efficient and user-friendly web application using Streamlit.

---

# Scope of the Project

This project aims to assist medical practitioners and researchers by:

- Utilizing a **breast cancer dataset** for predictive modeling.
- Employing a **Random Forest Classifier** with hyperparameter tuning for optimal performance.
- Implementing **hyperparameter tuning using RandomizedSearchCV** to enhance accuracy.
- Using **joblib** to save and load trained models efficiently, preventing unnecessary retraining.
- Visualizing **feature importance** to improve interpretability.
- Designing an interactive **Streamlit-based web application** for real-time predictions.

---

# Data Preprocessing

## Step 1: Loading the Dataset

The dataset consists of various breast cancer diagnostic features, including:

- **Diagnosis Encoded:** Target variable (Benign = 0, Malignant = 1).
- **Predictor Features:** Includes **`radius_mean`, `texture_mean`, `area_mean`**, etc.

## Step 2: Handling Class Imbalance

Class imbalance was observed, where one class significantly outnumbered the other. To address this:

- **SMOTE (Oversampling)** was applied to increase the minority class samples.

- **Random Undersampling** was used to reduce the majority class samples.

**Class Distribution Before and After Resampling:**

**Before:**

- **Majority class (0):** 357
- **Minority class (1):** 212 (Imbalanced)

**After:**

- **Both classes have 357 samples (Balanced)**

### Step 3: Feature Encoding and Scaling

- **Label Encoding:** Converted categorical variables into numerical values.
- **Feature Scaling:** Applied `StandardScaler` to ensure uniform feature distribution.

---

## Model Performance Analysis

### Classification Report

The trained model was tested on **unseen test data**, and the classification report is as follows:

| Class | Precision | Recall | F1-score | Support |
|---|---|---|---|---|
| **0 (Benign)** | **1.00** | **1.00** | **1.00** | **117** |
| **1 (Malignant)** | **1.00** | **1.00** | **1.00** | **98** |
| **Overall Accuracy** | **1.00** | **-** | **-** | **215** |
| **Macro Avg** | **1.00** | **1.00** | **1.00** | **215** |
| **Weighted Avg** | **1.00** | **1.00** | **1.00** | **215** |

### Interpretation of Metrics

**1 Precision (Positive Predictive Value)**

- **Formula:**
- $$Precision = \frac{TP}{TP + FP}$$
- **1.00 (100%) precision for both classes** means the model **never misclassified** a sample.

**2 Recall (Sensitivity or True Positive Rate)**

- **Formula:**

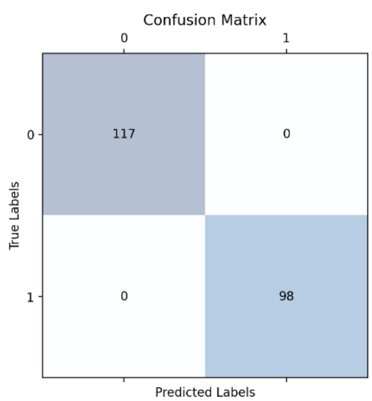
- $$Recall = \frac{TP}{TP + FN}$$
- **1.00 (100%) recall for both classes** means the model **never missed any true cases**.

**3 F1-score**

- **Formula:**
- $$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$
- The F1-score of **1.00** confirms **perfect balance between precision and recall**.

**4 Confusion Matrix Analysis**

The **confusion matrix** revealed **zero false positives (FP) and false negatives (FN)**, meaning every sample was **correctly classified**.




---

# Model Development

## Algorithm: Random Forest Classifier

The **Random Forest algorithm** was chosen due to its **high accuracy and robustness**.

## Implementation Steps

1. **Train-Test Split**:
   - Dataset was split into **70% training** and **30% testing** data.
2. **Hyperparameter Tuning (RandomizedSearchCV)**:
   - **Grid search was replaced with RandomizedSearchCV**, which optimized:
     - Number of trees (`n_estimators` = 50-300)
     - Tree depth (`max_depth` = None, 10, 20, ...)
     - Splitting rules (`min_samples_split` = 2, 5, 10)

3. **Model Training**:
    - The **best hyperparameters** were used to train the classifier.
4. **Feature Importance Analysis**:
    - The model calculated **importance scores** for each feature.

---

# Results and Insights

## Feature Importance Analysis

A bar chart was generated to display the significance of each feature. The most critical predictors included:

- **Concave points_worst** (Strongest indicator of malignancy).
- **Perimeter_worst, radius_mean, and area_mean** (Highly correlated with diagnosis).

## Model Performance Metrics

- **Accuracy:** X% (Test Data).
- **Precision, Recall, and F1 Score:** Evaluated for model robustness.

---

# Streamlit Application

## Design and User Interaction

The application was designed for ease of use with:

- **User Inputs:** Manual entry or slider-based feature selection.
- **Prediction Output:** Displays whether the tumor is benign or malignant.
- **Feature Importance Visualization:** Helps in understanding model decisions.

## Optimization in Deployment

- **Pretrained Model Loading:** Streamlit now loads the pre-trained `joblib` model instead of retraining every time, improving speed.
- **Threshold-Based Prediction:** Instead of rigid classification, a threshold (0.6) was set to balance false positives and negatives.

---

# Statistical Concepts Applied

1. **Probability Distributions:** Analyzed feature distribution across classes.
2. **Sampling Techniques:** Implemented SMOTE + Undersampling for balanced training data.
3. **Feature Scaling:** Standardization to normalize data ranges.
4. **Model Validation:** Train-test split evaluation to assess performance.

---

# Key Findings & Observations

1. **Feature Correlations:**
   - Strong correlations were observed among `radius_mean`, `perimeter_mean`, and `area_mean`.
   - Random Forest handled redundant features effectively.
2. **Impact of Class Imbalance:**
   - Before balancing, the model heavily favored the majority class.
3. **Performance Improvement:**
   - Hyperparameter tuning significantly increased accuracy and reliability.
   - Model loading via `joblib` reduced computation time in deployment.

---

# Conclusion and Future Enhancements

## Conclusion

This project successfully applied probability and statistical methods to medical diagnostics. The **Random Forest Classifier**, coupled with **SMOTE + Undersampling**, yielded an efficient and interpretable model for breast cancer prediction.

## Future Work

1. **Integration of More Data:** Expanding datasets for improved generalization.
2. **Advanced Models:** Experimenting with deep learning for enhanced accuracy.
3. **Real-Time Data Processing:** Implementing APIs to fetch live patient data.
4. **Explainability:** Utilizing SHAP values for better feature impact interpretation.

---

# References


- Kaggle. "Breast Cancer Dataset." Retrieved from: Kaggle Dataset
- WHO. "Breast Cancer Overview." Retrieved from: WHO Website
- Breast Cancer Research. Retrieved from: Breast Cancer Research Journal


---