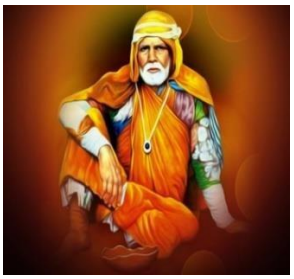


RAYAT SHIKSHAN SANSTHA'S

SADGURU GADGE MAHARAJ COLLEGE, KARAD

(An Autonomous College)

A Project Report On

"Predictive Analysis of Breast Cancer Diagnostic"

Department of Statistics

By

Miss. Anjali Deelip Mali

M.Sc.-II (2025-26)

Under The Guidance of

Mrs. Patil A. S.

CERTIFICATE

This is to certify that the project report on "**Predictive analysis of breast cancer diagnostic**". Being submitted by Miss. Anjali Deelip Mali as partial fulfillment for the award of degree of masters in Statistics at Sadguru Gadge Maharaj College, Karad is a record of Bonafide work carried out by them under supervision and guidance.

To the best of my knowledge and belief, the matter presented in the project report is original and has not been submitted elsewhere for any other purpose.

Place: Karad

Date:

Sr. No	Seat No.	Roll No.	Name of the student	Signature
1.			Anjali Deelip Mali	

Teacher in-charge Examiner PG Co-Ordinator Head

Department of Statistics

ACKNOWLEDGEMENT

This project entitled **“Predictive analysis of breast cancer diagnostic”.** I have great pleasure in presenting this report of successful completion of my project.

I sincerely thank my project guide, Mrs. Patil A. S., for her expert guidance, encouragement, and insightful suggestions throughout the course of this project. Her support played a vital role in shaping both the analytical depth and clarity of this work.

I am grateful to the faculty of the Department of Statistics, Sadguru Gadge Maharaj College, Karad, for providing a strong academic foundation and fostering a research-driven environment. Their teachings in statistical modelling and data analysis were instrumental in this endeavour.

I also extend my appreciation to my peers for their valuable feedback, and to my family and friends for their unwavering support and motivation.

This project has been a meaningful learning experience, and I am truly thankful to everyone who contributed to its successful completion.

INDEX

Sr. No	Title	Page No.
1	Introduction	05
2	Objective	06
3	Literature Review	07
4	Methodology	08
5	Tools and Techniques	09
6	Statistical Analysis	10-23
7	Conclusion	24
8	Future scope	25
9	Appendix	26-31

INTRODUCTION

Breast cancer remains one of the most prevalent and life-threatening diseases affecting women worldwide. Early and accurate diagnosis plays a critical role in improving survival rates and guiding effective treatment strategies. With the growing availability of medical datasets and advancements in data science, statistical modeling and machine learning have emerged as powerful tools in supporting clinical decision-making.

This project focuses on the analysis of the Breast Cancer Wisconsin (Diagnostic) dataset, which contains detailed measurements of cell nuclei extracted from digitized images of fine needle aspirate (FNA) of breast masses. The primary objective is to identify key features that distinguish benign from malignant tumors and to build predictive models that can classify diagnoses with high accuracy.

The study begins with exploratory data analysis and visualization, followed by statistical testing to validate feature significance. Dimensionality reduction techniques such as Principal Component Analysis (PCA) are used to assess class separation. Finally, classification models, including Logistic Regression and Random Forest, are developed and evaluated to determine their effectiveness in predicting tumor diagnosis.

By integrating statistical rigor with machine learning, this project aims to demonstrate how data-driven approaches can enhance diagnostic reliability and contribute meaningfully to healthcare analytics.

OBJECTIVES

- **To explore the Breast Cancer Wisconsin (Diagnostic) dataset**
  – This includes data cleaning, analysing class distribution, and generating descriptive statistics.
- **To identify statistically significant features**
  – Using t-tests and correlation analysis to find features that differentiate benign and malignant tumors.
- **To apply Principal Component Analysis (PCA)**
  – For dimensionality reduction and visualizing class separation.
- **To build and evaluate classification models**
  – Specifically Logistic Regression and Random Forest, using the selected features.
- **To compare model performance**
  – Using metrics like accuracy, precision, recall, and F1-score, and interpreting feature importance.

LITERATURE REVIEW

1. **Rahman, M. M., et al. (2025).** *Deep Learning Applications in Breast Cancer Diagnosis: A Global Review.*
   Published in the Journal of Biomedical Informatics, this study reviewed deep learning models applied to breast cancer imaging and clinical data. It found that CNN-based architectures achieved up to 99.96% accuracy and emphasized the value of hybrid models for improving diagnostic precision.

2. **Chakraborty, S., & Jha, A. (2024).** *Machine Learning Approaches for Cancer Detection: A Comparative Study.*
   This research compared SVM, Random Forest, and deep learning models across multiple cancer datasets. It highlighted the importance of feature selection and interpretability, recommending ensemble methods for clinical reliability.

3. **Zuo, Y., et al. (2023).** *Predicting Breast Cancer Recurrence Using Machine Learning and SHAP Analysis.*
   Published in Computers in Biology and Medicine, this study evaluated 11 algorithms and found AdaBoost to be most effective. It used SHAP values to interpret feature impact, demonstrating how explainable AI can support clinical decisions.

METHODOLOGY

Dataset Description:

- **Dataset Name**: Breast Cancer Wisconsin (Diagnostic)
- **Source**: UCI Machine Learning Repository
- **Link**: https://archive.ics.uci.edu/dataset/17/breast+cancer+wisconsin+diagnostic
- **Shape of the Data**: 569 rows × 32 columns
- **Type of Data**: Tabular, numeric (continuous features + categorical target)

Context & Purpose:

- The dataset contains diagnostic measurements from digitized images of fine needle aspirate (FNA) of breast masses.
- Each record represents a tumor case labelled as either **Benign (B)** or **Malignant (M)**.
- The goal is to predict tumor diagnosis using statistical and machine learning techniques.

Feature Overview:

- **Total Features**: 30 numeric predictors + 1 target variable + 1 ID column
- **Target Variable**: Diagnosis
  - Categorical: B (Benign) or M (Malignant)
- **ID Column**: Unique patient identifier (excluded from analysis)
- **Feature Groups**:
  - Each of the 10 base features is recorded in three forms:
    - _mean: average value
    - _se: standard error
    - _worst: worst case value
  - Base features include:
    - Radius, Texture, Perimeter, Area, Smoothness
    - Compactness, Concavity, Concave Points, Symmetry, Fractal Dimension

Data Characteristics:

- **No missing values**
- **Balanced class distribution**:
  - 357 Benign cases
  - 212 Malignant cases
- **All predictors are continuous** and suitable for statistical testing, PCA, and classification modelling.

TOOLS AND TECHNIQUES

Statistical Techniques:

- **Descriptive Statistics**
  Used summary statistics and t-tests to identify significant differences between benign and malignant tumors.
- **Principal Component Analysis (PCA)**
  Applied for dimensionality reduction and visualizing class separation.
- **Correlation Analysis**
  Explored relationships among features to detect multicollinearity
- **Performance Metrics**
  Evaluated models using Accuracy, Precision, Recall, F1-score, and AUC to assess predictive effectiveness.

Machine Learning Techniques

- **Hold-out Validation**
  Used an 80/20 train-test split to ensure unbiased model evaluation.
- **Logistic Regression**
  Built a baseline classification model using statistically significant features for interpretability.
- **Random Forest Classifier**
  Developed an ensemble model for robust classification and feature importance ranking.
- **Model Comparison**
  Compared Logistic Regression and Random Forest across multiple metrics to determine optimal performance.

Tools and Platforms

- **Python (Jupyter Notebook)**
  Used for data preprocessing, statistical analysis, model training, and visualization.
- **Scikit-learn**
  Core library for implementing machine learning algorithms and computing evaluation metrics.
- **Pandas & NumPy**
  Facilitated data manipulation, cleaning, and numerical operations.
- **Matplotlib & Seaborn**
  Created visualizations including histograms, boxplots, heatmaps, and ROC curves.

STATISTICAL ANALYSIS

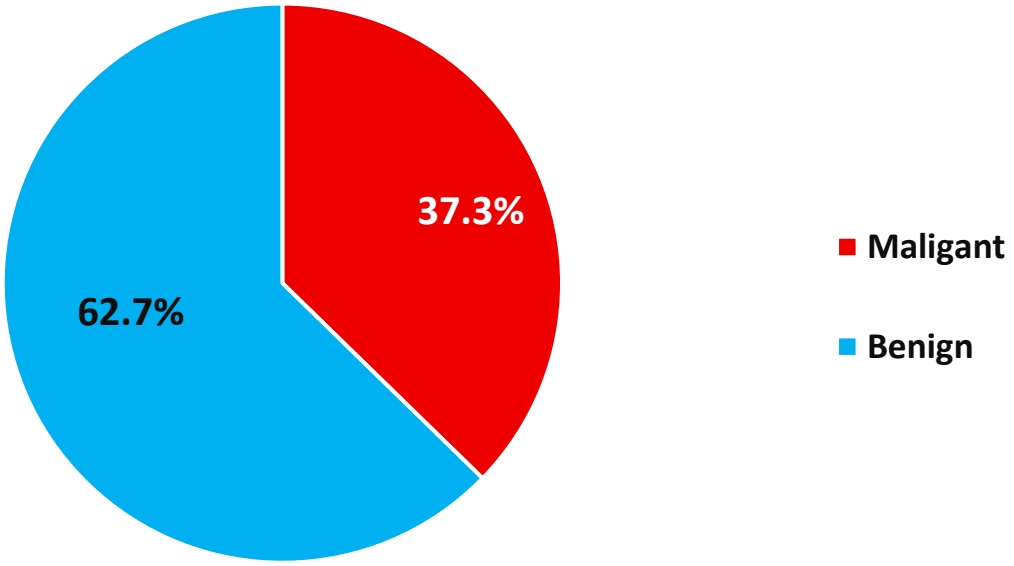
1. DATA PREPARATION AND ANALYSIS

Data Cleaning:

- Checked for missing values: None found.
- Verified data types: All features are numeric and suitable for statistical analysis.
- Renamed columns for clarity (e.g., radius_mean → radius1, texture_mean → texture1, etc.).
- Encoded diagnosis labels: Malignant = 1, Benign = 0.

Class Distribution:

Figure 1: Pie chart showing the proportion of benign and malignant cases in the dataset



- The dataset contains 569 samples, with 212 (37.3%) malignant and 357 (62.7%) benign cases. This moderate class balance supports unbiased model training and evaluation

2. DESCRIPTIVE STATISTICS

The following table summarizes the key descriptive statistics for selected features in the Breast Cancer Diagnostic dataset. It provides insights into the central tendency, spread, and distribution of each variable.

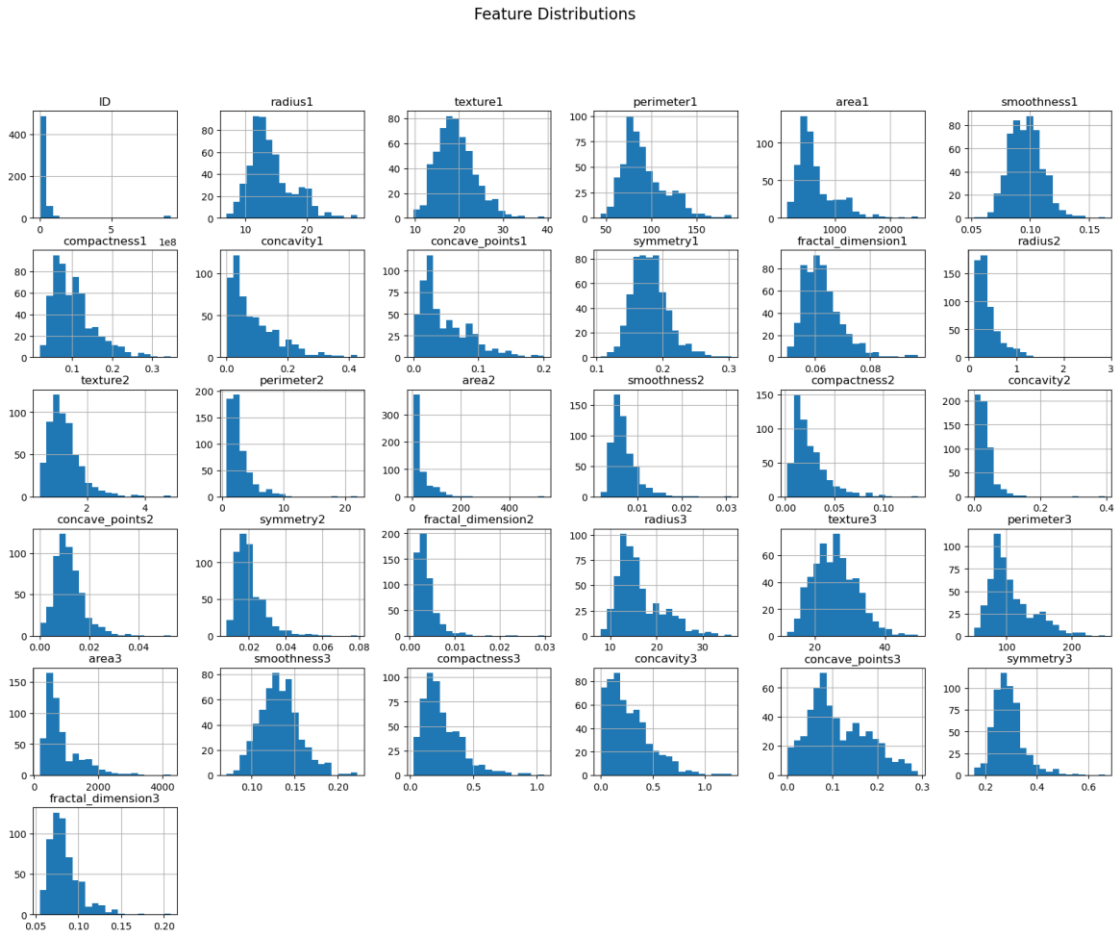
	count	mean	std	Min	25%	50%	75%	max
ID	569	30371831	1.25E+08	8670	869218	906024	8813129	9.11E+08
radius1	569	14.12729	3.524049	6.981	11.7	13.37	15.78	28.11
texture1	569	19.28965	4.301036	9.71	16.17	18.84	21.8	39.28
perimeter1	569	91.96903	24.29898	43.79	75.17	86.24	104.1	188.5
area1	569	654.8891	351.9141	143.5	420.3	551.1	782.7	2501
smoothness1	569	0.09636	0.014064	0.05263	0.08637	0.09587	0.1053	0.1634
compactness1	569	0.104341	0.052813	0.01938	0.06492	0.09263	0.1304	0.3454
concavity1	569	0.088799	0.07972	0	0.02956	0.06154	0.1307	0.4268
concave_points1	569	0.048919	0.038803	0	0.02031	0.0335	0.074	0.2012
symmetry1	569	0.181162	0.027414	0.106	0.1619	0.1792	0.1957	0.304
fractal_dimension1	569	0.062798	0.00706	0.04996	0.0577	0.06154	0.06612	0.09744
radius2	569	0.405172	0.277313	0.1115	0.2324	0.3242	0.4789	2.873
texture2	569	1.216853	0.551648	0.3602	0.8339	1.108	1.474	4.885
perimeter2	569	2.866059	2.021855	0.757	1.606	2.287	3.357	21.98
area2	569	40.33708	45.49101	6.802	17.85	24.53	45.19	542.2
smoothness2	569	0.007041	0.003003	0.001713	0.005169	0.00638	0.008146	0.03113
compactness2	569	0.025478	0.017908	0.002252	0.01308	0.02045	0.03245	0.1354
concavity2	569	0.031894	0.030186	0	0.01509	0.02589	0.04205	0.396
concave_points2	569	0.011796	0.00617	0	0.007638	0.01093	0.01471	0.05279
symmetry2	569	0.020542	0.008266	0.007882	0.01516	0.01873	0.02348	0.07895
fractal_dimension2	569	0.003795	0.002646	0.000895	0.002248	0.003187	0.004558	0.02984
radius3	569	16.26919	4.833242	7.93	13.01	14.97	18.79	36.04
texture3	569	25.67722	6.146258	12.02	21.08	25.41	29.72	49.54
perimeter3	569	107.2612	33.60254	50.41	84.11	97.66	125.4	251.2
area3	569	880.5831	569.357	185.2	515.3	686.5	1084	4254
smoothness3	569	0.132369	0.022832	0.07117	0.1166	0.1313	0.146	0.2226
compactness3	569	0.254265	0.157336	0.02729	0.1472	0.2119	0.3391	1.058
concavity3	569	0.272188	0.208624	0	0.1145	0.2267	0.3829	1.252
concave_points3	569	0.114606	0.065732	0	0.06493	0.09993	0.1614	0.291
symmetry3	569	0.290076	0.061867	0.1565	0.2504	0.2822	0.3179	0.6638
fractal_dimension3	569	0.083946	0.018061	0.05504	0.07146	0.08004	0.09208	0.2075
Diagnosis	569	0.372583	0.483918	0	0	0	1	1

Interpretation:

- Features such as radius1, area1, and concave_points1 show higher mean values, indicating their potential relevance in distinguishing tumor types.
- The standard deviations suggest moderate variability across samples, especially in size-related features.
- No missing values are present, and the ranges confirm that all features are on compatible scales for modelling.

3. Univariate analysis

Figure 2: Distribution of continuous features in the Breast Cancer Diagnostic dataset.

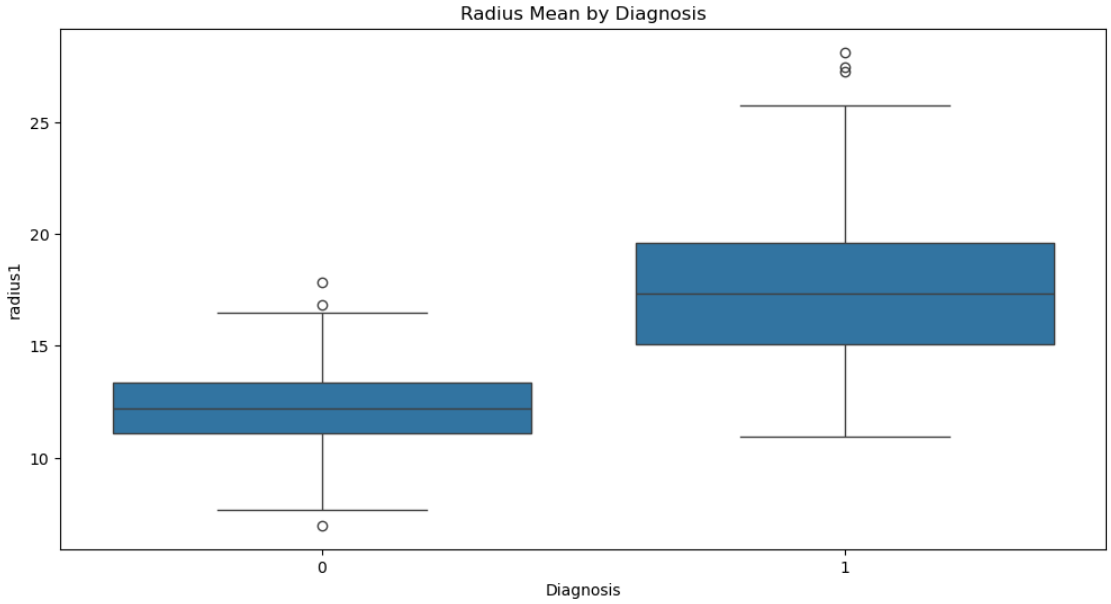


Interpretation:

- Most features show right-skewed distributions, especially area, concavity, and compactness, indicating the presence of outliers or extreme values.
- Features like smoothness and fractal_dimension appear more symmetrically distributed, suggesting consistent texture characteristics.
- The spread and shape of distributions vary across the three measurement sets (1, 2, 3), which may reflect different statistical summaries (e.g., mean, standard error, worst).

4. Bivariate analysis

Figure 3: Comparison of radius mean values between malignant and benign diagnoses.

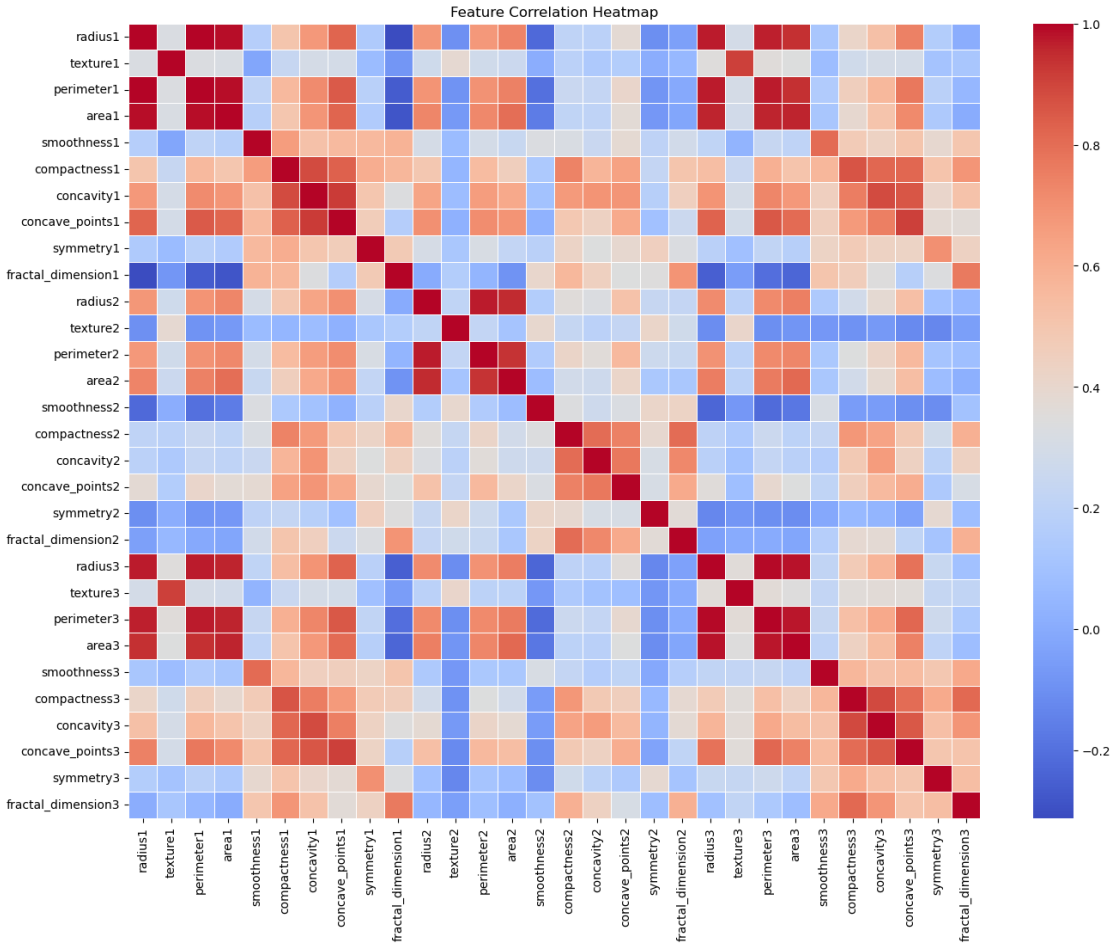


Interpretation:

- Malignant tumors (label 1) tend to have higher median radius values and a wider range, with several outliers.
- Benign tumors (label 0) show lower and more compact radius values, indicating less variability.
- This feature shows strong discriminative power, making it a potential candidate for classification modelling.

5. Correlation analysis

Figure 4: Feature Correlation Heatmap showing relationships among continuous variables.



Interpretation:

- radius1, perimeter1, and area1 are strongly correlated, indicating overlap in size-related features.
- concavity1, compactness1, and concave_points1 show high correlation, reflecting shared shape irregularities.
- Texture and fractal dimension features have weaker correlations, suggesting they add unique value.
- Consistent patterns across all three measurement sets support feature reliability.
- High correlations suggest dimensionality reduction or feature selection may improve model performance.

6. Statistical Significance of Key Features

To validate whether specific features show statistically significant differences across diagnosis groups (malignant vs benign).

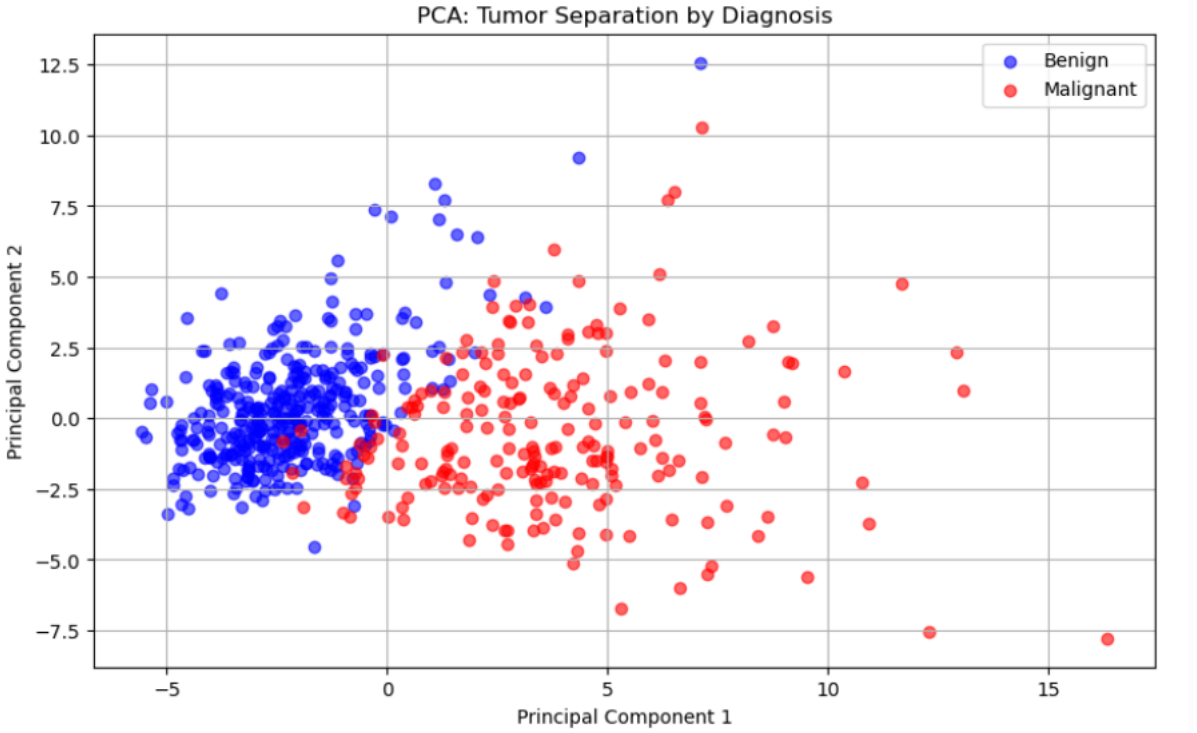
Feature	T-statistic	P-value	Interpretation
Radius1	-25.436	<0.001	Highly significant difference between groups
Area1	-23.939	<0.001	Strong evidence of group separation
Concave points1	-29.354	<0.001	Most discriminative among tested features
Texture1	-10.867	<0.001	Significant, but less pronounced than others

Interpretation:

- All four features show extremely low p-values, indicating that the differences in their means between malignant and benign tumors are statistically significant.
- The negative t-statistics suggest that malignant tumors tend to have higher values for these features.
- These results reinforce earlier visual findings and support their use in classification models.

7. Principal Component Analysis

figure 5: PCA plot showing separation between benign and malignant tumors.



Interpretation:

- The PCA scatter plot shows clear separation between benign (blue) and malignant (red) tumors.
- Principal Component 1 (PC1) captures most of the variance and contributes heavily to class separation.
- The two clusters are distinct and minimally overlapping, indicating that the original features effectively differentiate tumor types.
- This visual confirms that the dataset is well suited for classification tasks, and dimensionality reduction preserves diagnostic patterns.

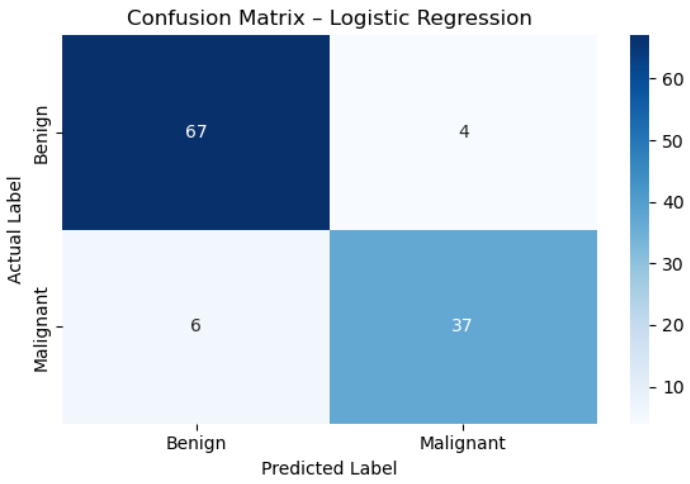
8. Logistic Regression Modeling

Preprocessing steps:

- Diagnosis labels encoded: Benign = 0, Malignant = 1
- Selected features: radius1, area1, concave_points1, texture1 (based on statistical significance)
- Train-test split: 80% training, 20% testing

Confusion Matrix:

	Predicted Benign	Predicted Malignant
Actual Benign	67	4
Actual Malignant	6	37



Classification Report:

Class	Precision	Recall	F1-score	Support
Benign	0.92	0.94	0.93	71
Malignant	0.90	0.86	0.88	43
accuracy			0.91	114

Interpretation:

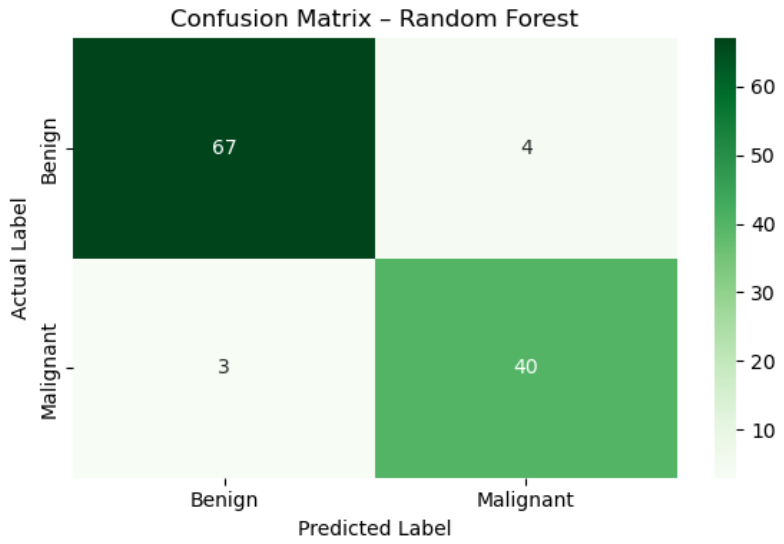
- The model achieved 91% accuracy, indicating strong overall performance.
- High precision for both classes means few false positives.

- Recall for malignant cases (0.86) is slightly lower, suggesting some false negatives, important in medical diagnosis.
- The selected features proved to be highly predictive, validating earlier EDA findings.

9. Random Forest Classification

Confusion Matrix:

	Predicted Benign	Predicted Malignant
Actual Benign	67	4
Actual Malignant	3	40



Classification Report:

Class	Precision	Recall	F1-score	Support
Benign	0.96	0.94	0.95	71
Malignant	0.91	0.93	0.92	43
accuracy			0.94	114

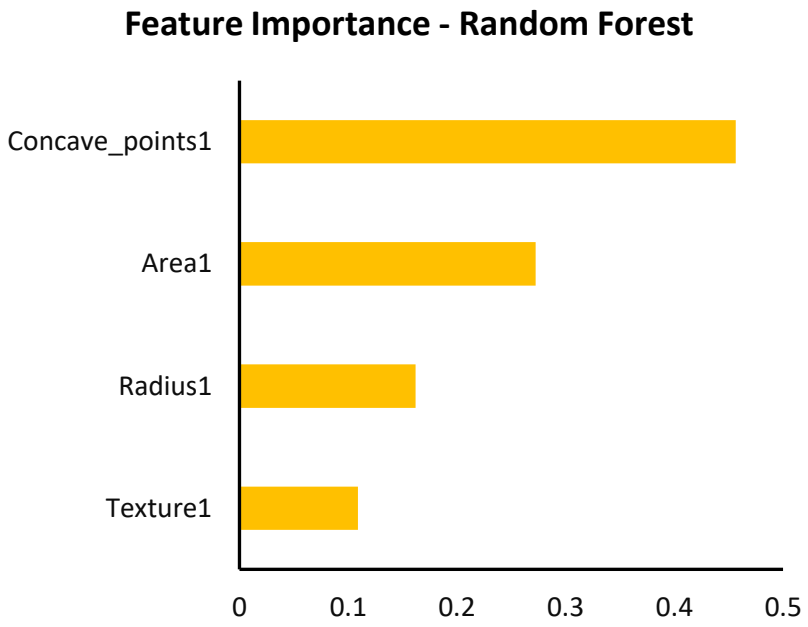
Interpretation

- The Random Forest model achieved 94% accuracy, outperforming logistic regression slightly.
- Recall for malignant cases improved to 0.93, reducing false negatives, critical in medical diagnosis.
- Ensemble methods like Random Forest offer robust performance and feature ranking, making them valuable for clinical prediction tasks.

Feature importance:

Feature	Importance
Concave_points1	0.4565
Area1	0.2723
Radius1	0.1621
Texture1	0.1091

Figure 6: Random Forest Feature Importance – Concave Points, Area, Radius, and Texture



Interpretation

- Concave_points1 is the most important feature, contributing nearly 46% to the model's predictions.
- Area1 ranks second with 27% importance, indicating tumor size is a strong diagnostic factor.
- Radius1 contributes 16%, reinforcing its relevance as a size-related feature.
- Texture1 has the lowest importance ,11% among the selected features, adding complementary but less critical information.

10. Model Comparison – Logistic Regression vs Random Forest

Metric	Logistic Regression	Random Forest
Accuracy	91.23%	93.86%
Precision	90.34%	90.91%
Recall	86.05%	93.02%
F1 score	88.09%	91.95%
AUC SCORE	90.21%	93.69%

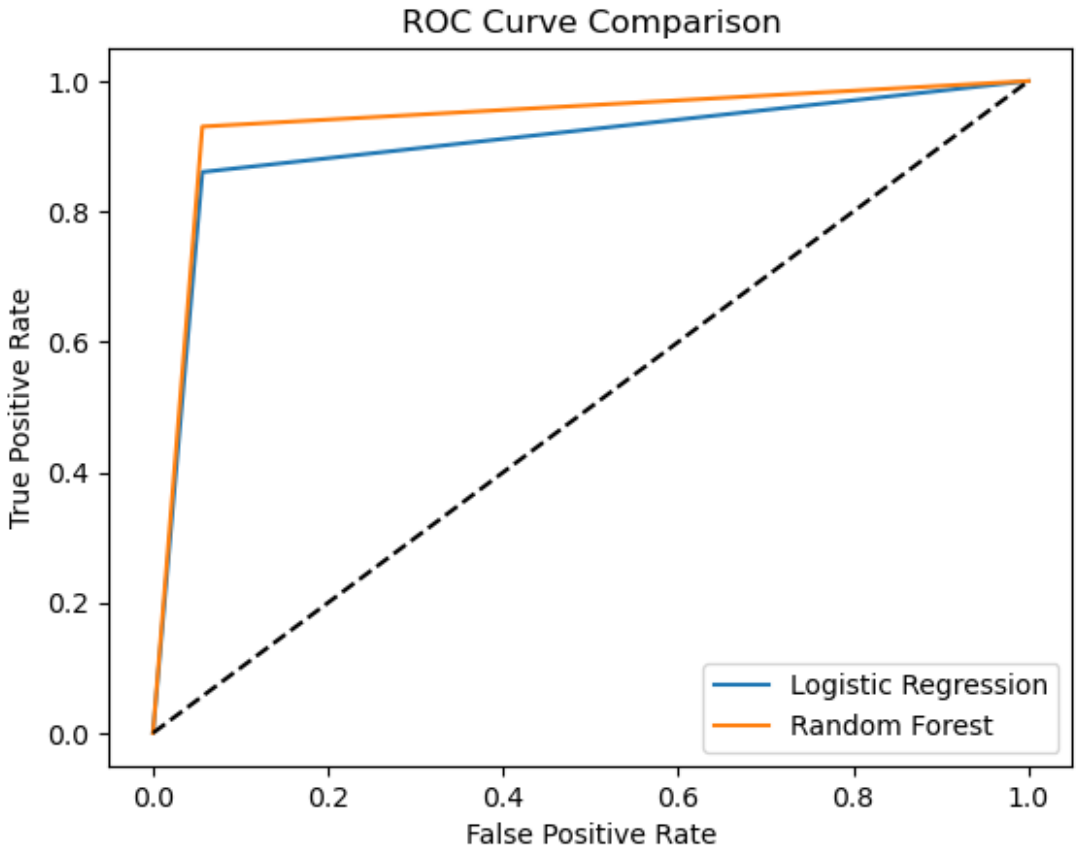
Interpretation

- Random Forest outperforms Logistic Regression across all metrics, especially in recall and F1-score, indicating better sensitivity and overall balance.
- Logistic Regression still performs well and offers greater interpretability, making it useful for understanding feature effects.
- The AUC scores suggest both models have strong discriminatory power, with Random Forest slightly ahead.

11. ROC Curve Comparison – Logistic Regression vs Random Forest

The ROC curve shows that Random Forest consistently achieves a higher true positive rate at lower false positive rates compared to Logistic Regression. This indicates superior discriminatory power.

Figure 7: ROC curves for Logistic Regression (blue) and Random Forest (orange).



Interpretation:

Random Forest demonstrates better sensitivity and overall classification performance, as reflected by its higher AUC score (93.69% vs 90.21%).

CONCLUSION

- Key features such as radius1, area1, and concave_points1 showed strong statistical significance and were selected for modelling.
- Malignant tumors consistently showed higher values in these features, confirmed by statistical tests and visual analysis.
- Strong correlations among size and shape features suggested redundancy; Principal Component Analysis helped reduce dimensionality and revealed clear class separation.
- Logistic Regression achieved **91.23%** accuracy, offering good baseline performance and interpretability.
- Random Forest achieved **93.86 %** accuracy, with higher recall and F1-score, making it more effective in detecting malignant tumors.
- ROC curve and AUC scores confirmed Random Forest's superior classification performance, with an AUC of **93.69 %**.
- Feature importance analysis ranked concave_points1 as the most influential predictor, followed by area1 and radius1.
- The dataset was clean, balanced, and highly suitable for classification modelling and clinical decision support.
- The project demonstrates how combining statistical analysis with machine learning can enhance breast cancer diagnosis and support data driven clinical decisions.

FUTURE SCOPE

- **Advanced Modelling**: Beyond logistic regression and random forest, more sophisticated algorithms like Support Vector Machines, Gradient Boosting, or Neural Networks can be explored to capture complex patterns and improve diagnostic accuracy.

- **Model Explainability**: Tools such as SHAP and LIME can be applied to interpret individual predictions, helping clinicians understand which features drive model decisions, critical for trust and transparency in healthcare.

- **Dashboard Development**: Translating insights into interactive dashboards using platforms like Power BI or Streamlit can support real-time decision-making and make results more accessible to non-technical users.

- **External Validation**: Testing the models on external datasets or real-world hospital data will help assess generalizability and ensure the models perform reliably beyond the current sample.

APPENDIX

```
pip install ucimlrepo

from ucimlrepo import fetch_ucirepo

# fetch dataset
breast_cancer_wisconsin_diagnostic = fetch_ucirepo(id=17)

# data (as pandas dataframes)
X = breast_cancer_wisconsin_diagnostic.data.features
y = breast_cancer_wisconsin_diagnostic.data.targets

# metadata
print(breast_cancer_wisconsin_diagnostic.metadata)

# variable information
print(breast_cancer_wisconsin_diagnostic.variables)

import pandas as pd

url = "https://archive.ics.uci.edu/static/public/17/data.csv"

df = pd.read_csv(url)

print(df.head())

columns = [
    "ID", "Diagnosis",
    "radius1", "texture1", "perimeter1", "area1", "smoothness1",
"compactness1", "concavity1", "concave_points1", "symmetry1",
"fractal_dimension1",
    "radius2", "texture2", "perimeter2", "area2", "smoothness2",
"compactness2", "concavity2", "concave_points2", "symmetry2",
"fractal_dimension2",
    "radius3", "texture3", "perimeter3", "area3", "smoothness3",
"compactness3", "concavity3", "concave_points3", "symmetry3",
"fractal_dimension3"
]

df.columns = columns

import pandas as pd
# to drop non-numeric columns like 'ID' and 'Diagnosis'
numeric_df = df.select_dtypes(include='number')
desc_stats = numeric_df.describe().transpose()
desc_stats

# Dropping ID column if not needed
df.drop("ID", axis=1, inplace=True)

# Encoding Diagnosis (M = 1, B = 0)
df["Diagnosis"] = df["Diagnosis"].map({"M": 1, "B": 0})
```

```
import matplotlib.pyplot as plt
import seaborn as sns

# Histograms for all features
df.drop("Diagnosis", axis=1).hist(figsize=(20, 15), bins=20)
plt.suptitle("Feature Distributions", fontsize=16)
plt.show()

# Boxplots grouped by Diagnosis
plt.figure(figsize=(12, 6))
sns.boxplot(data=df, x="Diagnosis", y="radius1")
plt.title("Radius Mean by Diagnosis")
plt.show()



# Converting fractal_dimension3 to numeric
df["fractal_dimension3"] = pd.to_numeric(df["fractal_dimension3"],
errors="coerce")

# Checking again
print(df.dtypes.tail())
```

```
import pandas as pd

url = "https://archive.ics.uci.edu/static/public/17/data.csv"
df = pd.read_csv(url, header=None)

columns = [
    "ID",
    "radius1", "texture1", "perimeter1", "area1", "smoothness1",
"compactness1", "concavity1", "concave_points1", "symmetry1",
"fractal_dimension1",
    "radius2", "texture2", "perimeter2", "area2", "smoothness2",
"compactness2", "concavity2", "concave_points2", "symmetry2",
"fractal_dimension2",
    "radius3", "texture3", "perimeter3", "area3", "smoothness3",
"compactness3", "concavity3", "concave_points3", "symmetry3",
"fractal_dimension3",
    "Diagnosis"
]
df.columns = columns

# to remove any spaces and map M/B to labels
df["Diagnosis"] = df["Diagnosis"].str.strip()
df["Diagnosis"] = df["Diagnosis"].map({"M": "Malignant", "B": "Benign"})

print(df["Diagnosis"].value_counts())
```

```
import pandas as pd

df =
pd.read_csv("C:/Users/Anjali/OneDrive/Documents/breast_cancer_clean.csv")
print(df.head())
```

```python
# Grouped descriptive stats by Diagnosis
desc_stats = df.groupby('Diagnosis').describe().transpose()
print(desc_stats)

import seaborn as sns
import matplotlib.pyplot as plt

# Selecting only numeric columns (excluding ID)
numeric_df = df.drop(columns=['ID', 'Diagnosis'])

# Compute correlation matrix
corr_matrix = numeric_df.corr()

# Plot heatmap
plt.figure(figsize=(16, 12))
sns.heatmap(corr_matrix, annot=False, cmap='coolwarm', linewidths=0.5)
plt.title('Feature Correlation Heatmap')
plt.show()


from scipy.stats import ttest_ind

# Separate groups
benign = df[df['Diagnosis'] == 'Benign']
malignant = df[df['Diagnosis'] == 'Malignant']

# Choose key features to test
features_to_test = ['radius1', 'area1', 'concave_points1', 'texture1']

# Run t-tests
for feature in features_to_test:
    stat, p = ttest_ind(benign[feature], malignant[feature])
    print(f"{feature}: t-stat = {stat:.3f}, p-value = {p:.5f}")

from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
import matplotlib.pyplot as plt

# Select numeric features only
features = df.drop(columns=['ID', 'Diagnosis'])
X = StandardScaler().fit_transform(features)

# Apply PCA
pca = PCA(n_components=2)
principal_components = pca.fit_transform(X)

# Create a new DataFrame for plotting
pca_df = pd.DataFrame(data=principal_components, columns=['PC1', 'PC2'])
pca_df['Diagnosis'] = df['Diagnosis']

# Plot
plt.figure(figsize=(10, 6))
colors = {'Benign': 'blue', 'Malignant': 'red'}
for label in colors:
```

```
    subset = pca_df[pca_df['Diagnosis'] == label]
    plt.scatter(subset['PC1'], subset['PC2'], c=colors[label], label=label,
alpha=0.6)

plt.xlabel('Principal Component 1')
plt.ylabel('Principal Component 2')
plt.title('PCA: Tumor Separation by Diagnosis')
plt.legend()
plt.grid(True)
plt.show()
```

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, confusion_matrix

# Encode Diagnosis
df['Diagnosis'] = df['Diagnosis'].map({'Benign': 0, 'Malignant': 1})

# Selecting features
X = df[['radius1', 'area1', 'concave_points1', 'texture1']]
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 model
model = LogisticRegression()
model.fit(X_train, y_train)

# Predict
y_pred = model.predict(X_test)

# Evaluate
print(confusion_matrix(y_test, y_pred))
print(classification_report(y_test, y_pred))
```

```
from sklearn.ensemble import RandomForestClassifier

# Train model
rf_model = RandomForestClassifier(random_state=42)
rf_model.fit(X_train, y_train)

# Predict
rf_pred = rf_model.predict(X_test)

# Evaluate
print(confusion_matrix(y_test, rf_pred))
print(classification_report(y_test, rf_pred))

# Feature importance
importances = rf_model.feature_importances_

for feature, importance in zip(X.columns, importances):
from sklearn.model_selection import train_test_split
```

```
from sklearn.linear_model import LogisticRegression

# Feature selection
X = df[['radius1', 'area1', 'concave_points1', 'texture1']]
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 model
model = LogisticRegression()
model.fit(X_train, y_train)

# Predict
y_pred = model.predict(X_test)
```

```
from sklearn.metrics import confusion_matrix
import seaborn as sns
import matplotlib.pyplot as plt

# Generate confusion matrix
cm = confusion_matrix(y_test, y_pred)

# Plot heatmap
plt.figure(figsize=(6, 4))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=['Benign',
'Malignant'], yticklabels=['Benign', 'Malignant'])
plt.title('Confusion Matrix – Logistic Regression')
plt.xlabel('Predicted Label')
plt.ylabel('Actual Label')
plt.tight_layout()
plt.show()
```

```
rf_model = RandomForestClassifier(random_state=42)
rf_model.fit(X_train, y_train)
rf_pred = rf_model.predict(X_test)
```

```
from sklearn.metrics import confusion_matrix
import seaborn as sns
import matplotlib.pyplot as plt


# rf_pred = predictions from your Random Forest model
# y_test = actual labels from your test set

# Generate confusion matrix
cm_rf = confusion_matrix(y_test, rf_pred)

# Plot heatmap
plt.figure(figsize=(6, 4))
sns.heatmap(cm_rf, annot=True, fmt='d', cmap='Greens', xticklabels=['Benign',
'Malignant'], yticklabels=['Benign', 'Malignant'])
```

```
plt.title('Confusion Matrix – Random Forest')
plt.xlabel('Predicted Label')
plt.ylabel('Actual Label')
plt.tight_layout()
plt.show()
```

```
from sklearn.metrics import accuracy_score, precision_score, recall_score,
f1_score, roc_auc_score

# Logistic Regression
print("Logistic Regression:")
print("Accuracy:", accuracy_score(y_test, y_pred))
print("Precision:", precision_score(y_test, y_pred))
print("Recall:", recall_score(y_test, y_pred))
print("F1 Score:", f1_score(y_test, y_pred))
print("AUC:", roc_auc_score(y_test, y_pred))

# Random Forest
print("\nRandom Forest:")
print("Accuracy:", accuracy_score(y_test, rf_pred))
print("Precision:", precision_score(y_test, rf_pred))
print("Recall:", recall_score(y_test, rf_pred))
print("F1 Score:", f1_score(y_test, rf_pred))
print("AUC:", roc_auc_score(y_test, rf_pred))
```

```
from sklearn.metrics import roc_curve
import matplotlib.pyplot as plt

fpr_lr, tpr_lr, _ = roc_curve(y_test, y_pred)
fpr_rf, tpr_rf, _ = roc_curve(y_test, rf_pred)

plt.plot(fpr_lr, tpr_lr, label='Logistic Regression')
plt.plot(fpr_rf, tpr_rf, label='Random Forest')
plt.plot([0, 1], [0, 1], 'k--')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('ROC Curve Comparison')
plt.legend()
plt.show()
```