STEP 1: Import Libraries

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

from sklearn.datasets import load_breast_cancer
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score
import joblib
```

STEP 2: Load Dataset

```
data = load_breast_cancer()
X = data.data
y = data.target
```

STEP 3: Train-Test Split (Same Split for All Models)

```
X_train, X_test, y_train, y_test = train_test_split(
    X, y,
    test_size=0.2,
    random_state=42,
    stratify=y
)
```

STEP 4: Feature Scaling (Needed for LR & SVM)

```
scaler = StandardScaler()

X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

STEP 5: Initialize Models

```
models = {
    "Logistic Regression": LogisticRegression(max_iter=2000),
    "Decision Tree": DecisionTreeClassifier(random_state=42),
    "Random Forest": RandomForestClassifier(n_estimators=100, random_state=42),
```

```
    "SVM": SVC(kernel='rbf', probability=True)
}
```

STEP 6: Train & Evaluate All Models

```
results = []

for name, model in models.items():

    if name in ["Logistic Regression", "SVM"]:
        model.fit(X_train_scaled, y_train)
        y_pred = model.predict(X_test_scaled)
        train_score = model.score(X_train_scaled, y_train)
    else:
        model.fit(X_train, y_train)
        y_pred = model.predict(X_test)
        train_score = model.score(X_train, y_train)

    test_score = accuracy_score(y_test, y_pred)

    results.append([
        name,
        train_score,
        test_score,
        precision_score(y_test, y_pred),
        recall_score(y_test, y_pred),
        f1_score(y_test, y_pred)
    ])
```

STEP 7: Create Comparison Table

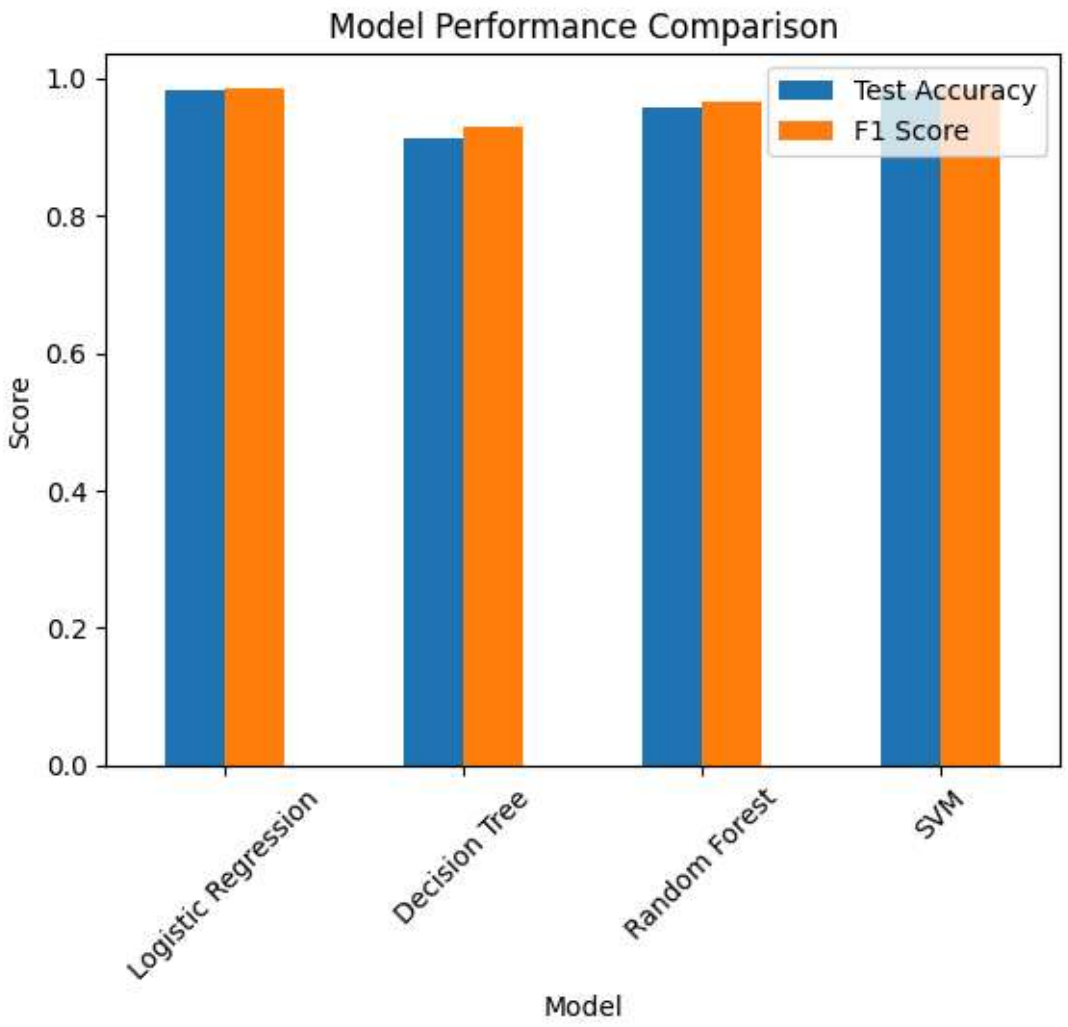
```
columns = [
    "Model",
    "Train Accuracy",
    "Test Accuracy",
    "Precision",
    "Recall",
    "F1 Score"
]

comparison_df = pd.DataFrame(results, columns=columns)
print(comparison_df)
```

```
                Model  Train Accuracy  Test Accuracy  Precision    Recall  \
0  Logistic Regression        0.989011       0.982456   0.986111  0.986111
1        Decision Tree        1.000000       0.912281   0.955882  0.902778
2        Random Forest        1.000000       0.956140   0.958904  0.972222
3                  SVM        0.982418       0.982456   0.986111  0.986111

   F1 Score
0  0.986111
```

```
1  0.928571
2  0.965517
3  0.986111
```

STEP 8: Plot Model Comparison

```
comparison_df.set_index("Model")[["Test Accuracy", "F1 Score"]].plot(kind="bar")

plt.title("Model Performance Comparison")
plt.ylabel("Score")
plt.xticks(rotation=45)
plt.show()
```



STEP 9: Identify Best Model

```
best_model_name = comparison_df.sort_values(
    by="F1 Score", ascending=False
).iloc[0]["Model"]

print("Best Model Based on F1 Score:", best_model_name)
```

```
Best Model Based on F1 Score: Logistic Regression
```

STEP 10: Save Best Model

```
best_model = models[best_model_name]

joblib.dump(best_model, "best_model.pkl")
```

```
['best_model.pkl']
```