In [11]:

```
import pandas as pd
import numpy as np

from sklearn.model_selection import KFold, StratifiedKFold, cross_val_score, GridSearchC
from sklearn.ensemble import RandomForestClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.svm import SVC

from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import Pipeline

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

data = pd.read_csv("breast_cancer_large_dataset.csv")

X = data.drop("target", axis=1)
y = data["target"]

print("\nK-FOLD CROSS VALIDATION (Random Forest)")

kf = KFold(n_splits=5, shuffle=True, random_state=42)
rf = RandomForestClassifier(random_state=42)

kf_scores = cross_val_score(rf, X, y, cv=kf, scoring="accuracy")

print("Accuracy Scores:", kf_scores)
print("Mean Accuracy:", kf_scores.mean())

print("\nSTRATIFIED K-FOLD CROSS VALIDATION (Random Forest)")

skf = StratifiedKFold(n_splits=5, shuffle=True, random_state=42)

skf_scores = cross_val_score(rf, X, y, cv=skf, scoring="accuracy")

print("Accuracy Scores:", skf_scores)
print("Mean Accuracy:", skf_scores.mean())

print("\nHYPERPARAMETER TUNING (Random Forest)")

param_grid = {
    "n_estimators": [100, 200, 300],
    "max_depth": [None, 10, 20],
    "min_samples_split": [2, 5, 10]
}

grid = GridSearchCV(
    RandomForestClassifier(random_state=42),
    param_grid,
    cv=5,
    scoring="accuracy"
)

grid.fit(X, y)

best_rf = grid.best_estimator_

print("Best Parameters:", grid.best_params_)
```

```python
print("Best Cross-Validation Accuracy:", grid.best_score_)

print("\nMODEL COMPARISON")

models = {
    "Random Forest": best_rf,
    "Decision Tree": DecisionTreeClassifier(random_state=42),
    "SVM": Pipeline([
        ("scaler", StandardScaler()),
        ("svm", SVC(kernel="linear"))
    ])
}

for name, model in models.items():
    scores = cross_val_score(model, X, y, cv=5, scoring="accuracy")
    print(f"{name} Mean Accuracy: {scores.mean()}")

print("\nFINAL EVALUATION METRICS (Random Forest)")

best_rf.fit(X, y)
y_pred = best_rf.predict(X)

print("Accuracy :", accuracy_score(y, y_pred))
print("Precision:", precision_score(y, y_pred))
print("Recall   :", recall_score(y, y_pred))
print("F1 Score :", f1_score(y, y_pred))
```

```
K-FOLD CROSS VALIDATION (Random Forest)
Accuracy Scores: [0.95614035 0.96491228 0.93859649 0.96491228 0.96460177]
Mean Accuracy: 0.9578326346840551

STRATIFIED K-FOLD CROSS VALIDATION (Random Forest)
Accuracy Scores: [0.96491228 0.93859649 0.95614035 0.94736842 0.97345133]
Mean Accuracy: 0.9560937742586555

HYPERPARAMETER TUNING (Random Forest)
Best Parameters: {'max_depth': 10, 'min_samples_split': 2, 'n_estimators': 300}
Best Cross-Validation Accuracy: 0.9631113181183046

MODEL COMPARISON
Random Forest Mean Accuracy: 0.9631113181183046
Decision Tree Mean Accuracy: 0.9173420276354604
SVM Mean Accuracy: 0.9718987734823784

FINAL EVALUATION METRICS (Random Forest)
Accuracy : 1.0
Precision: 1.0
Recall   : 1.0
F1 Score : 1.0
```

In [ ]: