# Assignment 1

## Report

**Name**: Erum Meraj

**Roll Numbe**r: 2201CS24

**Course**: CS502 - APR

**Algorithm:** Support Vector Machine (SVM)

**Dataset:** Breast Cancer Wisconsin Dataset

In this assignment, we use the **Breast Cancer Wisconsin Dataset** from the UCI Machine Learning Repository (also built into Scikit-learn) to build a classification model using **Support Vector Machines (SVM)**.

# 1. Dataset Description

The dataset contains **569 instances** and **30 features**, which are computed from digitised images of breast masses. Each instance is labelled as either:

- **Malignant (0)** – cancerous
- **Benign (1)** – non-cancerous

Sample features include:

- *Mean radius*
- *Mean texture*
- *Mean perimeter*
- *Mean area*
- *Mean smoothness*

# 2. Algorithm – Support Vector Machine (SVM)

SVM is a supervised machine learning algorithm used for classification and regression. Its key idea is to find an **optimal hyperplane** that separates classes with the maximum margin.

- **Kernel Used:** Radial Basis Function (RBF)
- **Reason:** Non-linear separation in high-dimensional data.
- **Feature Scaling:** StandardScaler was applied since SVMs are sensitive to feature magnitude.

# 3. Implementation

The model was implemented using **Python** and the **Scikit-learn** library. The main steps included:

## 1. Load the dataset

The breast cancer dataset was loaded directly from `sklearn.datasets`. It contains **30 features** and **2 classes** (malignant and benign).

```
from sklearn import datasets
```

```
# Load Dataset
data = datasets.load_breast_cancer()
X = data.data
y = data.target
```

## 2. Split into train and test sets

The dataset was split into **70% training** and **30% testing** using `train_test_split`, ensuring stratified sampling to maintain class balance.

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.3, random_state=42, stratify=y
)
```

## 3. Standardise the features

Since SVM is sensitive to feature scaling, we standardised the dataset using `StandardScaler`.

```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

## 4. Train an SVM classifier

We trained an **SVM classifier with an RBF kernel** (`C=2.0`, `gamma=scale`).

```
from sklearn.svm import SVC
model = SVC(kernel='rbf', C=2.0, gamma='scale')
model.fit(X_train, y_train)
```

## 5. Predictions

After training, predictions were generated on the test set.

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

### 6. Evaluate performance

Performance was evaluated using a **confusion matrix**, **classification report**, and **accuracy score**.

```
from sklearn.metrics import confusion_matrix, classification_report,
accuracy_score
print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred))
print("Classification Report:\n", classification_report(y_test, y_pred))
print("Accuracy:", accuracy_score(y_test, y_pred))
```

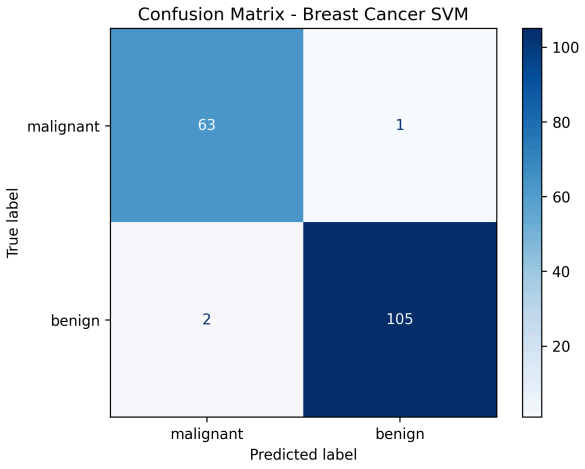
### 7. Cross-validation

To ensure generalisation, we applied **5-fold cross-validation**.

```
from sklearn.model_selection import cross_val_score
cv_scores = cross_val_score(model, X, y, cv=5)
print("Cross-validation Accuracy Scores:", cv_scores)
print("Mean CV Accuracy:", cv_scores.mean())
```

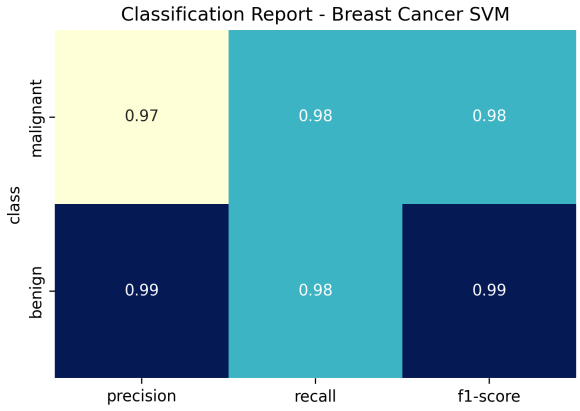
## 4. Results

### Confusion Matrix



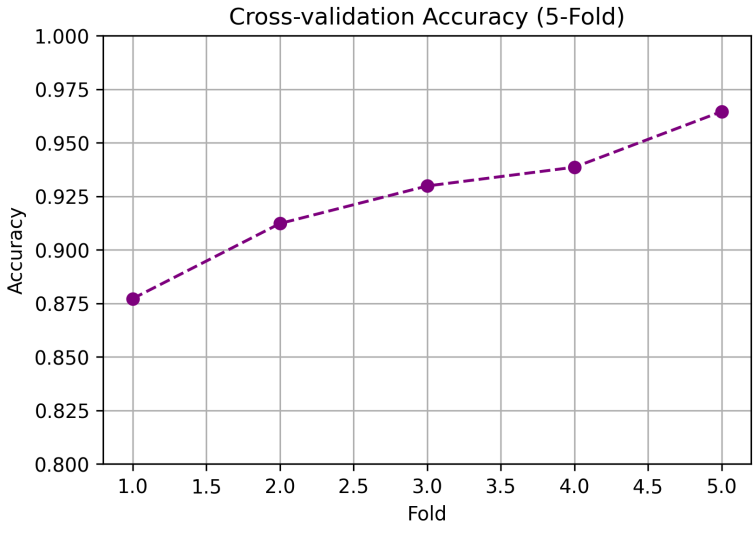
## Classification Report

| Class | Precision | Recall | F1-score | Support |
|---|---|---|---|---|
| Malignant (0) | 0.97 | 0.98 | 0.98 | 64 |
| Benign (1) | 0.99 | 0.98 | 0.99 | 107 |
| **Accuracy** | | | **0.98** | 171 |




## Cross-Validation Accuracy

- Fold scores: [0.877, 0.912, 0.930, 0.939, 0.965]
- **Mean CV Accuracy:** 0.924

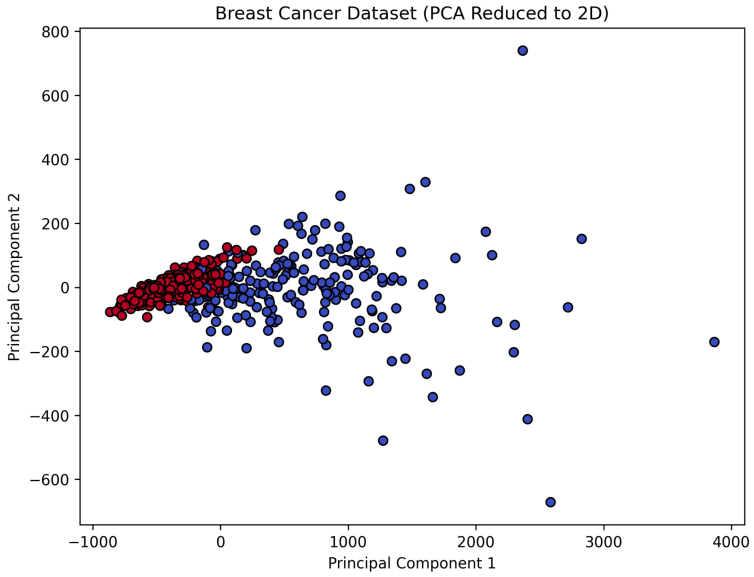



## Final Accuracy

- Test Accuracy: 0.9825

- Cross-validation Scores: [0.87719298
- 0.9122807 0.92982456 0.93859649 0.96460177]
- Mean CV Accuracy: 0.9245

# 5. Visualisation




# 6. Conclusion

The SVM classifier achieved a high test accuracy of **98.2%**, indicating that it can effectively separate malignant and benign breast tumours. The **cross-validation accuracy of ~92.4%** confirms that the model generalises well and is not overfitting.

This experiment demonstrates the potential of SVM in **medical diagnostics**, where accurate predictions are critical. Future work may include testing with other kernels, feature selection techniques, or comparing with alternative classifiers like Random Forests or Neural Networks.