CS-502 APR Assignment - 1
Breast Cancer Prediction Using Support Vector Machine (SVM)

2201CS15
Anchal Dubey

1 Introduction

Breast cancer prediction is a crucial problem in healthcare, where machine learning methods can significantly aid early diagnosis. In this assignment, a Support Vector Machine (SVM) model was developed using the **Breast Cancer Wisconsin (Original) dataset** to predict whether a tumor is malignant or benign. A polynomial kernel was used to classify the samples, with hyperparameter tuning to improve performance.

2 Dataset Description

The dataset used is the Breast Cancer Wisconsin (Original) dataset from the UCI Repository [1]. It consists of 699 samples with 11 attributes describing characteristics of cell nuclei from breast mass images. The target variable *Class* indicates tumor type:

- 4 = malignant
- 2 = benign

The other features represent various morphological characteristics of the cell nuclei, and their values range from 1 to 10. These features are; Clump thickness, Uniformity of cell size, Uniformity of cell shape, Marginal adhesion, Single epithelial cell size, Bland chromatin, normal nucleoli and mitoses.

Dataset link: `https://archive.ics.uci.edu/dataset/15/breast+cancer+wisconsin+original`

3 Data Preprocessing

The following preprocessing steps were applied to ensure data quality and consistency:

1. Dropped non-feature columns: *id* and *Class*.
2. Found that the **Bare Nuclei** feature contained non-numeric values and missing entries.
    - Converted values to numeric using `pd.to_numeric` with coercion.
    - Replaced missing values with the column mean.

```
[110]
✓ 0s
# Drop ID and Class columns
raw_data2 = raw_data.drop(['id','Class'], axis=1)

# Converting the 'Bare Nuclei' column to numeric values
raw_data2['Bare Nuclei'] = pd.to_numeric(raw_data2['Bare Nuclei'], errors='coerce')

# Filling the NaN values in the 'Bare Nuclei' column with the mean of the column to handle missing data
raw_data2['Bare Nuclei'].fillna(raw_data2['Bare Nuclei'].mean(), inplace=True)

print(raw_data2.dtypes)
```

```
Clump Thickness                  int64
Uniformity of Cell Size          int64
Uniformity of Cell Shape         int64
Marginal Adhesion                int64
Single Epithelial Cell Size      int64
Bare Nuclei                    float64
Bland Chromatin                  int64
Normal Nucleoli                  int64
Mitoses                          int64
dtype: object
```

Figure 1: Removing non-feature columns and numeric conversion with missing value handling

3. Normalized the features using z-score scaling (zero mean, unit variance).
4. Mapped the target column to binary values:
   - Malignant = 1
   - Benign = 0

4 Train-Test Split

The dataset was divided into:

- 70% training data
- 30% testing data

This ensured unbiased evaluation of the trained model.

```
[113]
# Split data into training and test features and labels using 30% of data as validation/test set
X_train, X_test, y_train, y_test = train_test_split(features, labels, test_size=.3, random_state=42)
print(X_train.shape, y_train.shape)
print(X_test.shape, y_test.shape)
```

```
(489, 9) (489,)
(210, 9) (210,)
```

Figure 2: Splitting Dataset into Training and Testing subsets

5 Model Selection and Training

The Support Vector Machine model was trained using a polynomial kernel. To optimize performance, hyperparameter tuning was performed using **GridSearchCV**, exploring different values for:

- Slack penalty parameter C
- Polynomial degree

The best parameters were found to be:

$$C = 0.1, \quad \text{degree} = 1$$

The model was then trained using these parameters.

```
[120]
✓ 0s
# Create and train a polynomial kernel SVM classifier with C=0.1 and degree=1
final_svc_poly = svm.SVC(C=.1, degree=1, kernel='poly', random_state=42)
final_svc_poly.fit(X_train, y_train)

# Evaluate the trained SVM model on the test set and return the accuracy score
final_svc_poly.score(X_test, y_test)

0.9619047619047619
```

Figure 3: Training and Evaluating the SVM Model

6 Model Evaluation

The trained SVM model was evaluated on the test set. The model achieved an accuracy of:

$$\text{Accuracy} = \mathbf{0.9619}$$

This demonstrates the effectiveness of the polynomial kernel SVM in classifying breast cancer tumors with high accuracy.

7 Discussion

A correlation heatmap of dataset features is also generated which reveals the relationship between different cell nucleus characteristics:

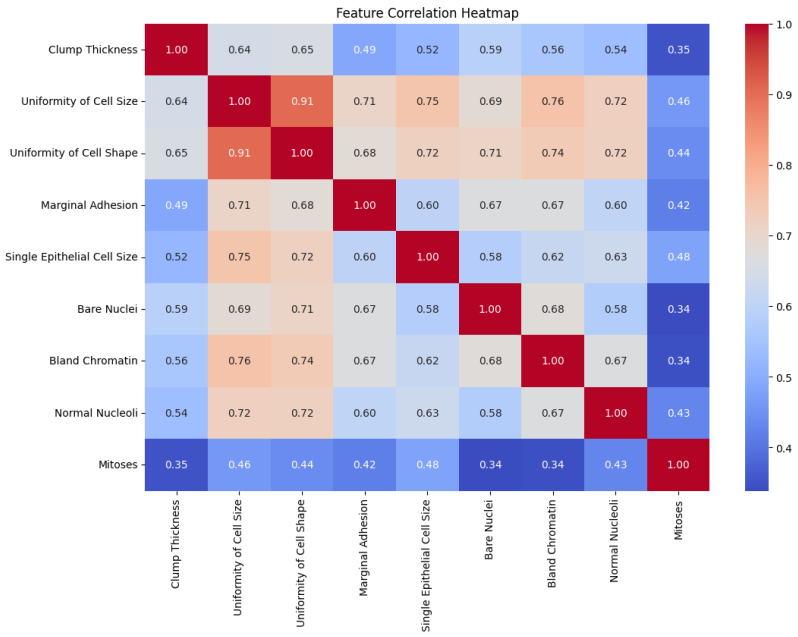

Figure 4: Correlation Heatmap of Dataset Features

Apart from the accuracy, the confusion matrix also reiterates the results of SVM, as can be seen below:

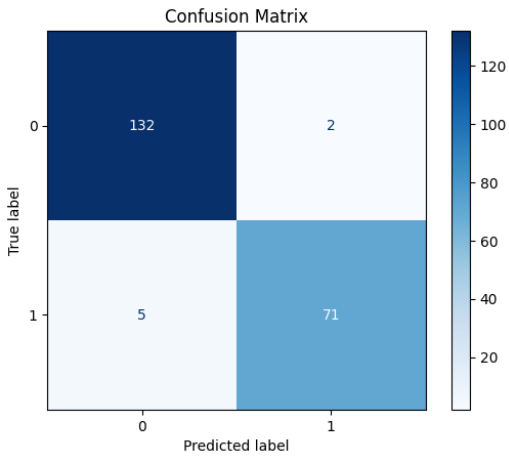

Figure 5: Confusion Matrix for SVM

8 Conclusion

This study applied a Support Vector Machine with a polynomial kernel to the Breast Cancer Wisconsin dataset. Through careful preprocessing, normalization, and hyperparameter tuning, the model achieved a high accuracy of 96.19% on the test set. The results indicate that SVM is a reliable and effective method for breast cancer prediction tasks.

References

[1] UCI Machine Learning Repository. *Breast Cancer Wisconsin (Original) Dataset*. `https://archive.ics.uci.edu/dataset/15/breast+cancer+wisconsin+original`