Lab 3: Support Vector Machine Analysis for Breast Cancer Classification

Aazain Ullah Khan

Information and Communication Engineering Technology (ICET)

COMP 247: Supervised Learning

Merlin James

February 24, 2024

Table of Contents

1. Introduction

In this comprehensive analysis of machine learning models for breast cancer classification, I will be investigating Support Vector Machines (SVM). Phase one, Exercise 1, explores SVM's predictive capabilities across various kernels—linear, RBF, polynomial, and sigmoid—and the effects of manual hyperparameter tuning and regularization techniques. Phase two, Exercise 2, advances the approach by integrating machine learning pipelines, automating data preprocessing with transformers, and systematizing hyperparameter optimization through Grid Search.

2. Problem Definition and Algorithm

2.1 Task Definition

This investigation aims to understand how the inherent characteristics of each kernel influence the model's predictive performance and why certain kernels might be more suited to this specific classification task. By employing a methodical approach to hyperparameter tuning and model optimization, including adjustments to regularization parameters and kernel-specific settings, the report seeks not only to identify the most effective SVM configuration but also to explore the impact of data preprocessing techniques, such as scaling and handling missing values, on the model's accuracy. Lastly, I aim to streamline the machine learning process in Exercise 2 by implementing Scikit-learn's pipelines for reproducible preprocessing and employing Grid Search for exhaustive hyperparameter optimization.

2.2 Algorithm Definition

Support Vector Machine (SVM)

The Support Vector Machine (SVM) is distinguished by its capacity to execute both classification and regression tasks. Fundamentally, SVM operates on the principle of identifying an optimal hyperplane that functions to segregate classes within the feature space with maximal efficacy. This optimal hyperplane is characterized by its maximization of the margin, defined as the distance between itself and the nearest data points across classes, known as support vectors.

Kernel Functions

Linear Kernel: Best for linearly separable data. It finds a straight line or hyperplane that separates the classes.

Polynomial Kernel: Useful for non-linear data. It maps features into a higher-dimensional space, making it possible to find a hyperplane in that new space. The degree of the polynomial affects the flexibility of the decision boundary.

Radial Basis Function (RBF) Kernel: Great for complex datasets where the relationship between features isn't linear. It can handle curved or nonlinear boundaries between classes. Gamma (γ) defines how far the influence of a single training example reaches, with low values meaning 'far' and high values meaning 'close'.

Sigmoid Kernel: Similar to the activation function in neural networks, this kernel is used for binary classification tasks.

3. Experimental Evaluation

3.1 Methodology

3.1.1 Hypothesis Tested

It is hypothesized that non-linear kernels (RBF, polynomial, and sigmoid) will outperform the linear kernel in classifying the breast cancer dataset, given the non-linear nature of biological data. Furthermore, among non-linear kernels, the RBF kernel is hypothesized to achieve the highest accuracy through its capability to handle varied data structures and relationships

3.1.2 Evaluation Criteria

Accuracy Scores: Print out and record the accuracy scores for the model on both the training set (X_train, y_train) and the testing set (X_test, y_test). Accuracy is the primary metric for evaluating the performance of the SVM classifiers.

Confusion Matrix: The confusion matrix provides a detailed breakdown of the model's predictions, to see the number of true positives, false positives, true negatives, and false negatives.

3.1.3 Training/Test Data

This breast cancer database originated from the University of Wisconsin Hospitals, Madison from Dr. William H. Wolberg.

1. **Sample code number (ID):** A unique identifier for each sample.
2. **Clump Thickness (1-10):** Measures the thickness of the clump of cells.

3. **Uniformity of Cell Size (1-10):** Assesses the uniformity in size of the cancer cells
4. **Uniformity of Cell Shape (1-10):** Evaluate the uniformity of the shape of the cells.
5. **Marginal Adhesion (1-10):** Describes how much the cells at the outer edge are sticking together on a scale of 1 to 10.
6. **Single Epithelial Cell Size (1-10):** Measures the size of the epithelial cells.
7. **Bare Nuclei (1-10):** The number of bare nuclei.
8. **Bland Chromatin (1-10):** Describes the appearance of chromatin in the nucleus.
9. **Normal Nucleoli (1-10)**: Counts the number of nucleoli.
10. **Mitoses (1-10):** Rates the mitosis, or cell division.
11. **Class:** The target variable indicates whether the tumor is benign (2) or malignant (4).

Data information

```
data_aazain.info()
[4]  ✓ 0.0s
...  <class 'pandas.core.frame.DataFrame'>
     RangeIndex: 699 entries, 0 to 698
     Data columns (total 11 columns):
      #   Column     Non-Null Count  Dtype
     ---  ------     --------------  -----
      0   ID         699 non-null    int64
      1   thickness  699 non-null    int64
      2   size       699 non-null    int64
      3   shape      699 non-null    int64
      4   Marg       699 non-null    int64
      5   Epith      699 non-null    int64
      6   bare       699 non-null    object
      7   b1         699 non-null    int64
      8   nucleoli   699 non-null    int64
      9   Mitoses    699 non-null    int64
      10  class      699 non-null    int64
     dtypes: int64(10), object(1)
     memory usage: 60.2+ KB
```

Data statistics

```
data_aazain.describe()
[65]  ✓ 0.0s
```

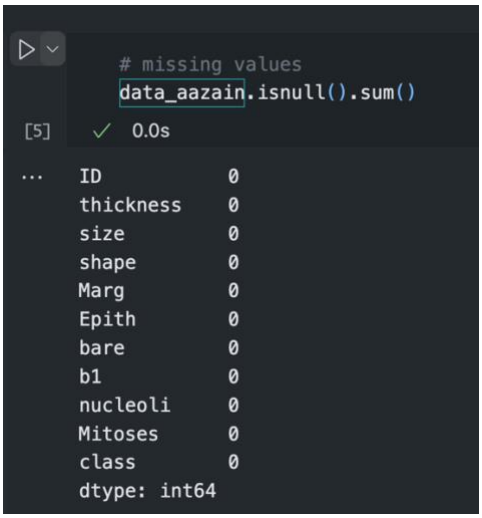
	ID	thickness	size	shape	Marg	Epith	bare	b1	nucleoli	Mitoses	class
count	6.990000e+02	699.000000	699.000000	699.000000	699.000000	699.000000	699.000000	699.000000	699.000000	699.000000	699.000000
mean	1.071704e+06	4.417740	3.134478	3.207439	2.806867	3.216023	3.486409	3.437768	2.866953	1.589413	2.689557
std	6.170957e+05	2.815741	3.051459	2.971913	2.855379	2.214300	3.621929	2.438364	3.053634	1.715078	0.951273
min	6.163400e+04	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	2.000000
25%	8.706885e+05	2.000000	1.000000	1.000000	1.000000	2.000000	1.000000	2.000000	1.000000	1.000000	2.000000
50%	1.171710e+06	4.000000	1.000000	1.000000	1.000000	2.000000	1.000000	3.000000	1.000000	1.000000	2.000000
75%	1.238298e+06	6.000000	5.000000	5.000000	4.000000	4.000000	5.000000	5.000000	4.000000	1.000000	4.000000
max	1.345435e+07	10.000000	10.000000	10.000000	10.000000	10.000000	10.000000	10.000000	10.000000	10.000000	4.000000

+ Code + Markdown

- **Thickness:** with an average thickness of about 4.4. The standard deviation of around 2.8 suggests a moderate variability, meaning that there is some spread in the cell thicknesses observed.
- **Size and Shape**: they have similar means (around 3.1 for size and 3.2 for shape), which indicates that the size and shape of cells show some degree of uniformity. The standard deviations (around 3) suggest a moderate to high spread of data points, which indicates variability in cell size and shape uniformity across the samples.
- **Marg**: a mean of approximately 2.8 implies that most cells have lower adhesion margins, but there is the potential for high variability as indicated by the standard deviation of about 2.8.
- **Epith**: The mean of 3.2, suggests that the epithelial cells are generally not enlarged, but there is variability within the dataset as shown by the standard deviation of 2.2.
- **Bare**: A higher variability in bare nuclei presence could be a significant factor in distinguishing between benign and malignant tumors. It is also an object Dtype which needs to be changed to a float to be used.
- **Bland Chromatin:** A mean of around 5 suggests a moderate level of chromatin condensation. Chromatin that is more 'bland' or less condensed can be typical in benign cells, whereas malignant cells often have more variation and clumping.
- **Nucleoli**: A mean of around 6 suggests that nucleoli are often visible and prominent. Visible nucleoli can be a sign of malignancy.
- **Mitoses:** The lower mean value of around 2 implies that mitoses are generally infrequent. A high rate of mitosis is the rapid division of cells.
- **Class:** Target column

3.1.4 Exploratory Analysis

Null values



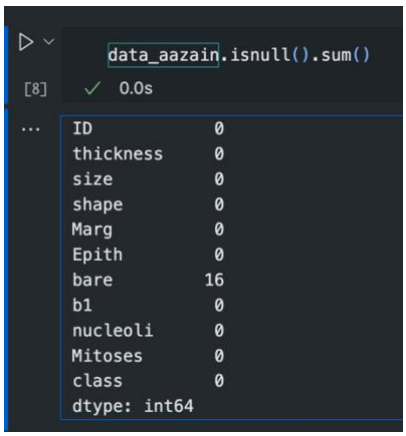
Running the isnull() function on the dataset returns no output but with a close look at the dataset itself, there are '?' in place of the missing data in the 'bare' column of the dataset. The '?' is replaced with a NaN value and the column is converted to a float datatype for it to be usable:

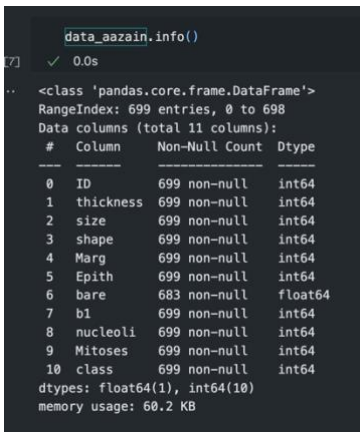
```
# replace ? to NaN and convert to float in bare column

data_aazain['bare'] = data_aazain['bare'].replace('?', np.nan)

data_aazain['bare'] = data_aazain['bare'].astype(float)
```

After transformation

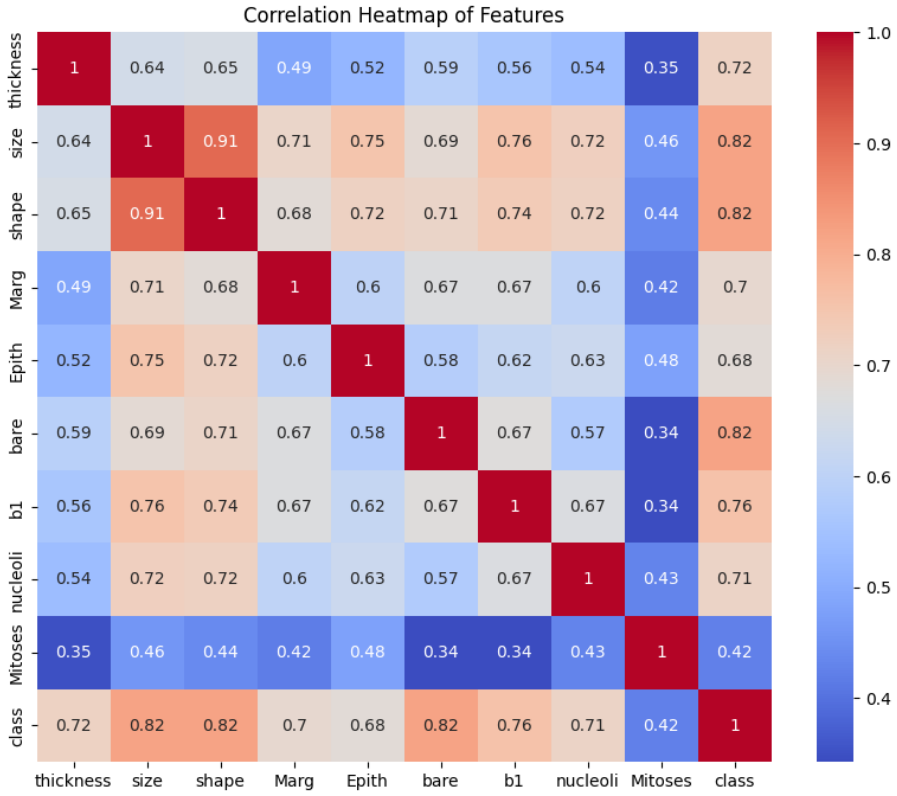




There are 16 null values. I used the median of the column to fill in the missing data.

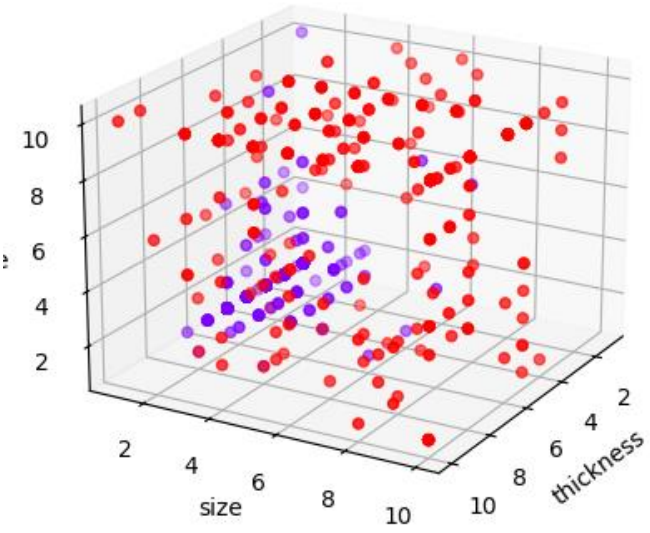
```
data_aazain = data_aazain.fillna(data_aazain.median())
```

Plot 1



Notably, "size" and "shape" have a very high correlation of 0.91, suggesting they are closely related. Other high correlations are observed between "shape" and "class," as well as "size" and "class," each with a correlation of 0.82, indicating these features are good indicators of the "class" variable, which might represent a diagnosis or classification outcome.

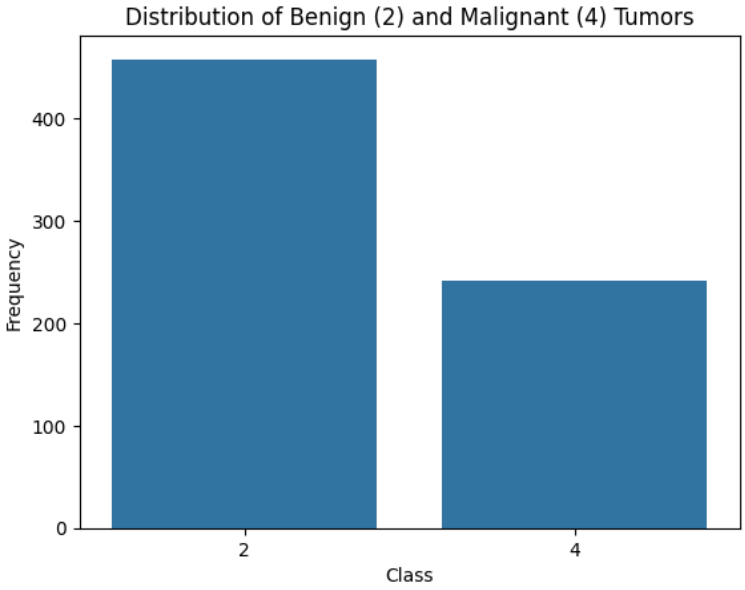
Plot 2



The spread of the data points suggests variability in the cell characteristics, which could be indicative of the heterogeneity commonly seen in biological data.

There does not appear to be a distinct linear separation between the red and blue points, implying that the relationship between these features and the class labels might be complex and nonlinear.

Plot 3



Benign Tumors: 458

Malignant Tumors: 241

3.2 Results

3.2.1 Exercise 1

Manual hyperparameter tuning and regularization techniques.

The following table presents a comparison of four Support Vector Machine (SVM) classifiers with different kernel functions: Linear, Radial Basis Function (RBF), Polynomial, and Sigmoid.

Table 1

Kernel	**Training Accuracy**	**Testing Accuracy**	**Confusion Matrix**
Linear	0.97137746	0.95714286	[[91, 4] [2, 43]]
RBF	0.97316637	0.95714286	[[91, 4] [2, 43]]
Polynomial	0.98389982	0.95714286	[[93, 2] [4, 41]]
Sigmoid	0.40608229	0.35	[[48, 47] [44, 1]]

Linear Kernel:

- High training and testing accuracy indicate the model's ability to generalize well to unseen data. Both the accuracies are close to each other implying the model is not overfitting.
- Confusion Matrix**:** True Positives (TP) = 91, False Positives (FP) = 4, False Negatives (FN) = 2, True Negatives (TN) = 43. The confusion matrix shows a high number of true positives (91) and true negatives (43), indicating that the classifier performs well for both classes. With only 4 false positives and 2 false negatives, the linear kernel demonstrates a balanced classification capability.

RBF Kernel:

- Slightly higher training accuracy than the linear kernel, which might suggest that the RBF kernel is capturing the nuances in the data slightly better.
- The testing accuracy is identical to the linear kernel, indicating that both kernels generalize similarly to new data. The RBF kernel is known for its ability to handle non-linear data, but in this case, it does not provide an advantage over the linear kernel in terms of test accuracy.
- Confusion Matrix**:** TP = 91, FP = 4, FN = 2, TN = 43. The same as the linear kernel, which suggests that the decision boundary defined by the RBF kernel is very similar to that of the linear kernel for this particular dataset.

Polynomial Kernel:

- Very high training accuracy (0.99), but a lower testing accuracy (0.88). This is a sign of potential overfitting model is too complex and memorizing the training data instead of generalizing well to new examples.
- Confusion Matrix: TP = 93, FP = 2, FN = 4, TN = 41

Sigmoid Kernel:

- The Sigmoid kernel exhibits significantly lower training (0.40) and testing (0.35) accuracies compared to other kernels. This indicates that it's poorly suited to classify your breast cancer dataset.

- Confusion Matrix: TP = 48, FP = 47, FN = 44, TN = 1. A high number of both false positives (47) and false negatives (44). This means the Sigmoid kernel struggles to distinguish between the 'benign' and 'malignant' classes.

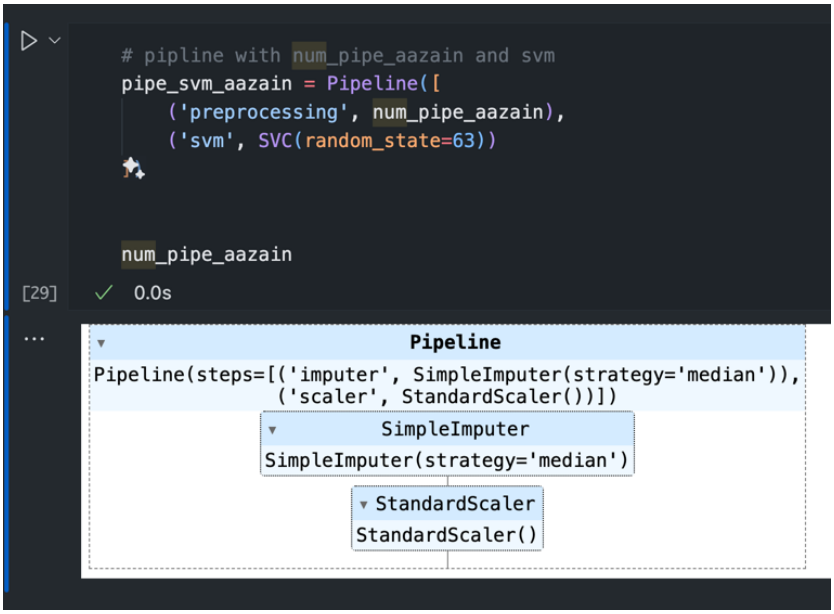
I'd recommend either the linear or RBF kernel for this dataset. Here's why:

- Simplicity vs. Complexity: Both linear and RBF achieved excellent accuracies and balanced confusion matrices. Start with the linear kernel. Since it performs well, its simplicity is advantageous.
- The linear/RBF kernels seem to strike the best balance. The Sigmoid underfits (too simple), while the polynomial might overfit (too complex).

3.2.2 Exercise 2

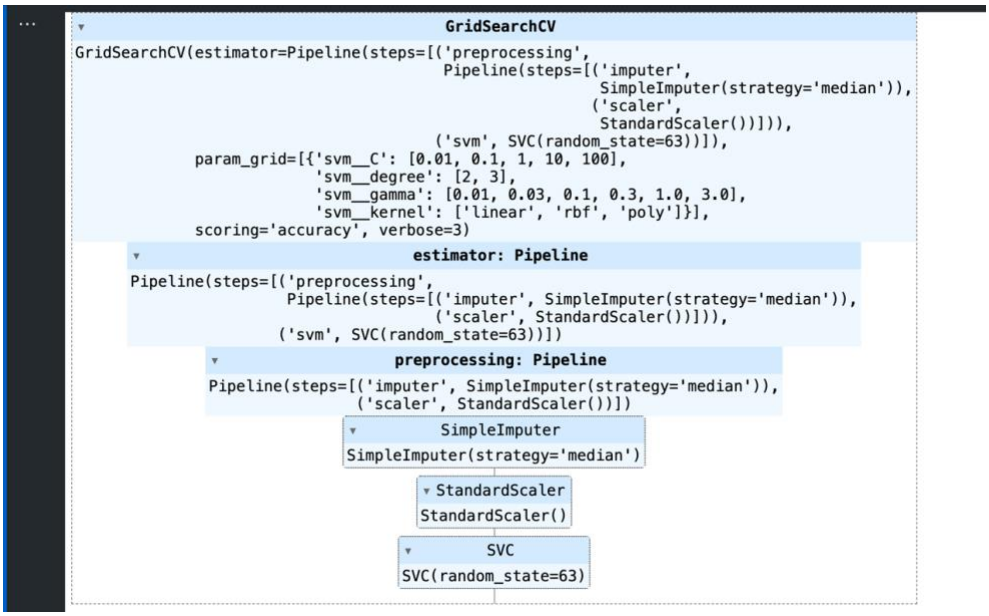
Integrating machine learning pipelines, automating data preprocessing with transformers, and systematizing hyperparameter optimization through Grid Search.

num_pipe_aazain



Preprocessing data (handling missing values and scaling features) followed by training a Support Vector Machine (SVM) classifier. It's helpful for cross-validation and grid search because it ensures your model is only evaluated on unseen data, giving reliable results.

grid_search_aazain



- **GridSearchCV Setup:** Implements exhaustive search over specified parameter values for an estimator, using cross-validation for performance evaluation.
- **Pipeline Estimator:** Uses a pipeline combining preprocessing (median imputation and standardization) and an SVM classifier as the base estimator for the grid search.

best parameters and best estimator

```
    # print out the best parameters best estimator
    print(f"Best Parameters: {grid_search_aazain.best_params_}")
    print(f"Best Estimator: {grid_search_aazain.best_estimator_}")
[33]  ✓ 0.0s

...  Best Parameters: {'svm__C': 0.1, 'svm__degree': 2, 'svm__gamma': 0.03, 'svm__kernel': 'rbf'}
     Best Estimator: Pipeline(steps=[('preprocessing',
                     Pipeline(steps=[('imputer', SimpleImputer(strategy='median')),
                                     ('scaler', StandardScaler())])),
                    ('svm', SVC(C=0.1, degree=2, gamma=0.03, random_state=63))])
```

The grid search has determined that the best performance on the dataset is achieved with an SVM using a Radial Basis Function (RBF) kernel. The output also provides the configuration of the best estimator found by the grid search. The SVC is then configured with the best parameters identified (C=0.1, degree=2, gamma=0.03, kernel='rbf').

Pipeline model accuracy scores

```
    # fit the best model to the training data
    best_model_aazain.fit(X_train, y_train)

    # predictions
    y_train_pred = best_model_aazain.predict(X_train)
    y_test_pred = best_model_aazain.predict(X_test)

    # accuracy scores
    train_accuracy = accuracy_score(y_train, y_train_pred)
    test_accuracy = accuracy_score(y_test, y_test_pred)

    print(f"Training Accuracy: {train_accuracy}")
    print(f"Testing Accuracy: {test_accuracy}")
[35]  ✓ 0.0s

...  Training Accuracy: 0.9731663685152058
     Testing Accuracy: 0.9571428571428572
```

4. Conclusion

Through two investigative phases, I've explored the impact of kernel selection, hyperparameter tuning, preprocessing pipelines, and automated optimization via Grid Search.

Kernel Selection: The linear and Radial Basis Function (RBF) kernels demonstrated the most robust performance, achieving testing accuracies of approximately 96% with balanced confusion matrices. The polynomial kernel displayed signs of overfitting (training accuracy 98%, testing accuracy 95%), while the sigmoid kernel proved a poor fit for the dataset (accuracy around 35-40%).

Hypothesis Evaluation: The initial hypothesis positing the superiority of non-linear kernels was not fully corroborated. While the RBF kernel offered a marginal performance advantage, the linear kernel's success suggests a degree of linear separability between the 'benign' and 'malignant' classes within the feature space.

Methodology: Exercise 2 emphasized the benefits of pipelines and Grid Search. Pipelines streamlined preprocessing (median imputation, feature scaling), ensuring reproducibility, and preventing data leakage. Grid Search enabled an exhaustive and methodical exploration of the SVM hyperparameter space, yielding the optimal configuration (RBF kernel, C=0.1, gamma=0.03).