A

Project Report

on

Enhanced Endometriosis Diagnosis with Voting Classifiers: Integrating SVM, Gradient Boosting, and Decision Tree Models

Submitted
by

J.CHENCHU DHARANI [R190878]

R.KUSUMA [R190932]

N.REDDEPPA [R190879]

Under the guidance
Of

Ms. C.SUNEETHA M.Tech,

Assistant Professor

Department of Computer Science and Engineering

Rajiv Gandhi University of Knowledge Technologies (RGUKT)
R K Valley, Y.S.R. Kadapa, Andhra Pradesh

RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES

(A.P.Government Act 18 of 2008)

RGUKT IIIT RK Valley

Vempalli, Kadapa, Andhra Pradesh – 516330.

CERTIFICATE OF PROJECT COMPLETION

This is to certify that I have examined the thesis entitled “Enhanced Endometriosis Diagnosis with Voting classifier: Integrating SVM, Gradient Boosting , Decision Tree Models” submitted by J.CHENCHU DHARANI(R190878),R.KUSUMA (R190932) and N.REDDEPPA(R190879) under our guidance and supervision for the partial fulfilment for the degree of Bachelor of Technology in computer Science and Engineering during the academic session January 2024 - July 2024 at RGUKT-RKVALLEY.

To the best of my knowledge, the results embodied in this dissertation work have not been submitted to any university or institute for the award of any degree or diploma.

Project Guide

Ms C Suneetha
Assistant Professor
Computer Science and Engg
RGUKT-RK Valley.

Head of the Department(CSE)

Dr.Ch.Ratna Kumari
Assistant Professor
Head of the Department
Computer Science and Engg
RGUKT-RK Valley

RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES

(A.P.Government Act 18 of 2008)

RGUKT IIIT RK Valley

Vempalli, Kadapa, Andhra Pradesh – 516330.

DECLARATION

We, J.CHENCHU DHARANI(R190878), R.KUSUMA (R190932) and N.REDDEPPA(R190879) hereby declare that the project report entitled "Enhanced Endometriosis Diagnosis with Voting classifier: Integrating SVM, Gradient Boosting , Decision Tree Models" done by us under guidance of **Ms.C.Suneetha** is submitted in partial fulfillment for the degree of Bachelor of Technology in Computer Science and Engineering during the academic session January 2024 – July 2024 at RGUKT-RK Valley.

I also declare that this project is a result of our own effort and has not been copied or imitated from any source. Citations from websites are mentioned in the references. To the best of my knowledge, the results embodied in this dissertation work have not been submitted to any university or institute for the award of any degree or diploma.

Date :

Place : RGUKT,RK Valley

J.CHENCHUDHARANI[R190878]

R.KUSUMA [R190932]

N.REDDEPPA [R190879]

ACKNOWLEDGEMENT

I would like to express my sense of gratitude and respect to all those people behind the screen who guided, inspired and helped me to crown all our efforts with success. I wish to express my gratitude to my project coordinator **Ms C.Suneetha** for her valuable guidance at all stages of study, advice, constructive suggestions, supportive attitude and continuous encouragement, without which this project could not be possible.

I would also like to extend our deepest gratitude and reverence to the Head of Department of Computer Science and Engineering **Dr.Ch.Ratna kumari** and also Director of RGUKT, RK Valley **Dr. A V S Kumara Swami Gupta** for their constant support and encouragement.

Last but not least I express my gratitude to my parents for their constant source of encouragement and inspiration for me to keep my morals high.

With Sincere Regards,

J.CHENCHUDHARANI[R190878]
R.KUSUMA [R190932]
N.REDDEPPA [R190879]

Table of Contents

List of Graphs

List of Tables

List of Figures

Chapter 1

Abstract

Endometriosis, a chronic and often debilitating condition affecting millions of women worldwide, presents significant diagnostic challenges. This project aims to enhance the accuracy and efficiency of endometriosis diagnosis by integrating three powerful machine learning models: Support Vector Machine (SVM), Gradient Boosting, and Decision Tree, through a voting classifier approach. By leveraging the strengths of each model, we aim to develop a robust diagnostic tool that can improve clinical outcomes and reduce diagnostic delays.

The methodology involves the creation of a comprehensive binary dataset of patient records, which includes clinical symptoms and medical history. This dataset undergoes rigorous preprocessing, including normalization, feature selection, and handling of missing values, to ensure the highest data quality. The SVM model is chosen for its ability to handle high-dimensional data, the Gradient Boosting model for its strength in minimizing bias and variance, and the Decision Tree model for its interpretability and ease of use.

Our voting classifier aggregates the predictions of these three models using both soft and hard voting mechanisms. Soft voting involves averaging the predicted probabilities, while hard voting relies on majority rule. We evaluate the performance of the integrated model against individual classifiers using metrics such as accuracy, precision, recall, and F1-score. The results demonstrate a significant improvement in diagnostic accuracy, underscoring the effectiveness of the voting classifier in enhancing endometriosis diagnosis.

This study highlights the potential of machine learning in transforming medical diagnostics. By integrating diverse classifiers, the proposed model achieves higher reliability and robustness compared to traditional diagnostic methods. Future work will focus on expanding the dataset, incorporating additional clinical features, and exploring the integration of deep learning models to further improve diagnostic performance.

Chapter 2

Introduction

Endometriosis is a chronic and often painful condition where tissue similar to the lining inside the uterus, known as endometrium, grows outside the uterus. This aberrant growth can affect the ovaries, fallopian tubes, and the tissue lining the pelvis. In rare cases, endometrial-like tissue may spread beyond the pelvic organs. Endometriosis commonly involves the ovaries, bowel, and the tissue lining the pelvis.

One of the primary symptoms of endometriosis is pelvic pain, often associated with the menstrual cycle. Although many women experience cramping during their menstrual periods, those with endometriosis typically describe menstrual pain that is far worse than usual. Pain may also increase over time. Other symptoms include pain during intercourse, pain with bowel movements or urination, excessive bleeding, and infertility. The severity of pain is not necessarily a reliable indicator of the extent of the condition; some women with severe endometriosis have mild symptoms, while others with a milder form of the disease have severe symptoms.

The exact cause of endometriosis is not known. However, several theories have been proposed, including retrograde menstruation (where menstrual blood flows backward through the fallopian tubes into the pelvic cavity instead of leaving the body), embryonic cell transformation, surgical scar implantation, immune system disorders, and genetic factors. While these theories provide potential explanations, none of them fully account for all cases of endometriosis.

Endometriosis is a significant public health issue due to its high prevalence and the impact it has on women's lives. It affects an estimated 1 in 10 women during their reproductive years, approximately 176 million women worldwide. Despite its prevalence, there is often a delay in diagnosis, with an average lag of 7 to 10 years from the onset of symptoms to diagnosis. This delay is partly due to the normalization of menstrual pain and the lack of awareness about the condition.

Diagnosis of endometriosis typically involves a combination of clinical evaluation, imaging tests, and sometimes surgical procedures. Pelvic exams, ultrasound, and magnetic resonance imaging (MRI) can help detect endometriosis, but the only definitive way to diagnose the condition is through laparoscopy, a minimally invasive surgical procedure in which a surgeon views the inside of the abdomen to look for endometrial tissue.

Treatment for endometriosis focuses on managing symptoms, as there is currently no cure for the condition. Options include pain relief medications, hormone therapy to reduce or eliminate menstruation, and surgical procedures to remove endometrial growths. In severe cases, a hysterectomy may be considered, though this is typically a last resort and not a guaranteed cure, as endometrial-like tissue can grow in areas outside the uterus.

Living with endometriosis can significantly affect a woman's quality of life, impacting physical health, emotional well-being, and overall life satisfaction. Chronic pain and fatigue can hinder daily activities and work productivity, while infertility issues can be emotionally distressing for those wishing to conceive. Support from healthcare providers, family, and patient support groups can be crucial in managing the condition.

The application of machine learning in healthcare has seen significant growth, with algorithms such as Support Vector Machines (SVM), Gradient Boosting, and Decision Trees playing pivotal roles. Each of these models brings unique strengths to the table: SVM is renowned for its robustness in high-dimensional spaces, Gradient Boosting excels in handling complex, non-linear relationships, and Decision Trees provide interpretable decision-making paths. By combining these models in a voting classifier, we can leverage their individual advantages to create a more robust diagnostic tool for endometriosis.

Support Vector Machines (SVM) are effective in classification tasks where the decision boundary is complex. They work by finding the hyperplane that best separates the classes in a high-dimensional feature space. In the context of endometriosis diagnosis, SVMs can handle the multi-dimensional clinical and symptomatic data, ensuring a high degree of accuracy in distinguishing between affected and non-affected patients. Their ability to manage both linear and non-linear data through kernel functions makes them particularly valuable in this application.

Gradient Boosting Machines (GBM) are another powerful tool in the ML arsenal, particularly adept at building strong predictive models from weak learners. GBM operates by sequentially adding models to correct the errors made by previous models, thus improving the overall performance. In diagnosing endometriosis, GBM can identify intricate patterns and relationships within the data, capturing the subtle indicators of the disease that might be missed by other methods.

Decision Trees are favored for their simplicity and interpretability, providing clear and intuitive paths to decision-making. They split the data into branches based on feature values, leading to easily understandable if-then-else rules. For endometriosis diagnosis, Decision Trees can offer straightforward diagnostic criteria that can be easily communicated to healthcare professionals and patients alike. Their interpretability makes them an essential component of a voting classifier, ensuring that the model's decisions can be trusted and validated.

A voting classifier combines the predictions of multiple models to make a final decision, enhancing the overall performance and robustness. This ensemble approach mitigates the weaknesses of individual models by averaging out their errors. In the context of endometriosis diagnosis, a voting classifier that integrates SVM, GBM, and Decision Trees can harness the strengths of each model, leading to a more reliable and accurate diagnostic tool.

The integration of these models in a voting classifier is achieved through techniques such as hard voting or soft voting. Hard voting takes the majority vote of the classifiers' predictions, while soft voting averages the probabilities assigned by each classifier and selects the class with the highest average probability. This study employs a hybrid approach, leveraging the strengths of both methods to optimize diagnostic performance.

This research evaluates the performance of the proposed voting classifier using a dataset of clinical and symptomatic data from women suspected of having endometriosis. Key metrics such as accuracy, sensitivity, specificity, and area under the ROC curve (AUC) are used to assess the effectiveness of the model. The results demonstrate significant improvements over individual classifiers, highlighting the potential of the voting classifier to enhance diagnostic accuracy and reliability.

Chapter 3

Problem statement

Endometriosis is a chronic and often debilitating condition that affects a significant percentage of women of reproductive age, leading to symptoms such as severe menstrual pain, chronic pelvic pain, and infertility. Despite its prevalence, the diagnosis of endometriosis is notoriously challenging and often delayed, sometimes taking years from the onset of symptoms to a definitive diagnosis. This delay can exacerbate the condition and diminish the quality of life for those affected. Traditional diagnostic methods, including laparoscopy and imaging, can be invasive, expensive, and not always conclusive. There is a pressing need for a more efficient, non-invasive, and accurate diagnostic approach to aid healthcare professionals in identifying endometriosis at an earlier stage, thus improving patient outcomes and reducing healthcare costs.

To address this issue, we propose the development of a machine learning-based diagnostic tool leveraging a voting classifier model that integrates three robust algorithms: Support Vector Machine (SVM), Decision Tree, and Gradient Boosting. By combining these classifiers, the model aims to capitalize on the strengths of each algorithm, improving overall diagnostic accuracy and robustness. The SVM is known for its effectiveness in high-dimensional spaces, the Decision Tree provides interpretability and simplicity, and Gradient Boosting offers high predictive performance by sequentially correcting errors. This ensemble approach is designed to enhance the predictive power and reliability of the diagnostic tool, making it a valuable asset in clinical settings.

The proposed diagnostic tool takes into account a comprehensive set of clinical symptoms commonly associated with endometriosis, such as heavy menstrual bleeding, dysmenorrhea, pelvic pain, and gastrointestinal issues, among others. By analyzing these input features, the voting classifier model can predict the likelihood of endometriosis, providing a probabilistic assessment that can guide further diagnostic testing and treatment plans. The integration of machine learning into the diagnostic process represents a significant advancement, offering a scalable and cost-effective solution to a pervasive and often overlooked healthcare challenge. Our project underscores the potential of artificial intelligence to transform medical diagnostics and improve the lives of countless women suffering from endometriosis.

Chapter 4

Literature review

Year	Author	Algorithm	Outcome	Drawback
2016	Leila Aghili	Support Vector Machine (SVM)	The study demonstrated that SVM could effectively classify endometriosis cases with a high degree of accuracy based on a set of clinical and symptomatic features.	The primary limitation was the small sample size, which restricted the generalizability of the results. Additionally, the study did not compare SVM with other machine learning algorithms to contextualize its performance.
2018	Michael F. Kotlyar	Random Forest	The random forest algorithm showed high accuracy in predicting endometriosis by analyzing a large dataset of clinical and demographic information. The model's feature importance ranking provided valuable insights into the most predictive variables.	One limitation was the lack of external validation on independent datasets, which raised questions about the model’s robustness across different populations. Also, the model's complexity made it difficult to interpret the results fully.
2019	Sarah A. Omrani	Logistic Regression	Logistic regression was used to predict endometriosis, showing good performance and	The study was limited by the exclusion of some potentially important features due to

			providing clear interpretability of the model coefficients. The simplicity of the algorithm made it easy to implement in clinical settings.	multicollinearity. Additionally, logistic regression's performance was not benchmarked against more advanced machine learning techniques.
2020	Yu Zhang	Gradient Boosting Machine (GBM)	GBM achieved high predictive accuracy for endometriosis, outperforming traditional statistical methods and some other machine learning algorithms. It effectively handled complex interactions between features.	The model required substantial computational resources and time for training, limiting its feasibility for real-time clinical use. The study also noted potential overfitting issues due to the model's high complexity.
2021	Lina T. Peters	Neural Networks	Neural networks provided a powerful tool for predicting endometriosis, with the ability to capture intricate patterns in the data. The model exhibited high sensitivity and specificity in identifying endometriosis cases.	The main limitation was the need for large amounts of data for effective training, which was not always available. Additionally, neural networks function as black-box models, making it difficult to interpret the decision-making process.

Chapter 5

Module 1: Data Collection and Preprocessing

Data collection and preprocessing are fundamental steps in developing machine learning (ML) models, particularly in the medical field where accurate and reliable data are crucial for diagnosis and treatment. For the diagnosis and labeling of endometriosis using machine learning, binary data—representing the presence or absence of certain features—plays a crucial role. This section will provide an in-depth discussion on the strategies, methodologies, and considerations involved in the data collection and preprocessing of binary data for endometriosis diagnosis, with a special focus on correlation analysis as a preprocessing step.

Data Collection

Data collection involves gathering relevant binary data that can be used to train and evaluate machine learning models. In the context of endometriosis diagnosis, binary data typically indicates the presence or absence of specific clinical features, symptoms, or diagnostic markers. The key aspects of binary data collection include:

1.Sources of Binary Data:

Clinical Records: Binary data can be extracted from clinical records where specific symptoms or diagnostic criteria are either present (1) or absent (0). For example, the presence of pelvic pain, dysmenorrhea, or infertility can be coded as binary variables.

Patient-Reported Outcomes: Survey data where patients report the presence (1) or absence (0) of symptoms such as chronic pain, heavy menstrual bleeding, or fatigue can be valuable binary data sources.

Laboratory Results: Biomarker tests such as CA-125 levels can be interpreted as binary variables, with specific thresholds indicating positive (1) or negative (0) results for endometriosis.

2. Data Collection Methods:

Electronic Health Records (EHR): EHR systems are used to systematically collect and store binary data from clinical encounters. These systems enable the extraction of structured data that can be directly used for analysis.

Laboratory Information Systems (LIS): LIS manage laboratory data, including binary outcomes of biomarker tests. These systems ensure the accurate recording and retrieval of binary test results.

Survey Tools: Online and paper-based survey tools are used to collect patient-reported binary outcomes. These tools need to be designed to accurately capture the presence or absence of symptoms.

3. Challenges in Data Collection:

Data Quality: Ensuring the accuracy and consistency of collected binary data is crucial. Misclassification or incomplete data can lead to biased results and poor model performance.

Data Privacy: Protecting patient privacy and ensuring compliance with regulations such as HIPAA (Health Insurance Portability and Accountability Act) is essential.

Data Integration: Integrating binary data from multiple sources (EHR, imaging, LIS, surveys) can be challenging due to differences in data formats and standards.

Data Preprocessing

Once binary data is collected, it must be preprocessed to ensure it is suitable for training machine learning models. Preprocessing involves several steps aimed at transforming raw data into a clean and usable format. Key steps in preprocessing binary data for endometriosis diagnosis include:

1.Data Cleaning:

- **Handling Missing Data:** Missing binary data is a common issue. Techniques such as imputation (filling missing values with the mode) or deletion (removing records with missing values) are used to address this.
- **Removing Duplicates:** Duplicate records can distort analysis and lead to biased results. Identifying and removing duplicate entries is crucial.
- **Correcting Errors:** Ensuring that binary data entries are correct (e.g., verifying clinical findings) is essential for data integrity.

2. Data Transformation:

Using ‘ColumnTransformer’ in our project has significantly enhanced the efficiency and organization of our data preprocessing pipeline. By encapsulating different preprocessing steps within separate components,’ ColumnTransformer’ promotes modularity and clarity in our workflow. This approach allows us to apply specific transformations, such as scaling with ‘StandardScaler’, to designated subsets of columns based on their data types or other criteria.

For instance, while applying ‘StandardScaler’ to numeric features for standardization, categorical features remain untouched, ensuring appropriate handling of different data representations.

The integration of ‘ColumnTransformer’ into our machine learning pipeline has simplified our workflow considerably. Instead of manually applying transformations to each column or group of columns, ‘ColumnTransformer’ automates this process according to our predefined rules. This automation not only reduces the potential for human error but also enhances the reproducibility of our preprocessing steps across different datasets and model iterations. This capability is particularly valuable as it ensures consistency in data preprocessing from training to deployment phases.

Moreover, ‘ColumnTransformer’ offers flexibility by allowing us to define complex preprocessing workflows. By chaining multiple transformers within ‘ColumnTransformer’, we can easily accommodate diverse preprocessing requirements, adapting our strategies to the specific characteristics of different datasets or machine learning tasks. This flexibility empowers us to experiment with various preprocessing techniques efficiently and to tailor our approach to optimize model performance effectively.

Documenting our preprocessing steps with ‘ColumnTransformer’ also enhances transparency and interpretability. Each transformation applied through ‘ColumnTransformer’ is explicitly documented, clarifying which columns underwent which transformations. This documentation not only facilitates the understanding of our machine learning model's inputs and outputs but also supports collaboration and knowledge sharing within our team.

In conclusion, ‘ColumnTransformer’ has proven instrumental in structuring our data preprocessing pipeline methodically. It has enabled us to manage transformations more effectively, improve the performance of our machine learning models, and maintain a high standard of reproducibility and transparency throughout our project lifecycle.

3. Feature Engineering:

Creating Composite Variables: New binary features can be created by combining existing ones. For instance, combining several symptom indicators into a single composite symptom score (e.g., presence of any two out of three symptoms) can enhance the predictive power.

Interaction Terms: Interaction terms between binary variables can be created to capture more complex relationships in the data. For example, the interaction between the presence of pain and the presence of cysts can be a significant predictor.

4. Dimensionality Reduction:

Removing Redundant Features: Binary data often includes redundant features that do not contribute to model performance. Techniques like correlation analysis are used to identify and remove such features.

Feature Selection: Selecting the most relevant binary features using techniques like mutual information or model-based methods (e.g., feature importance from random forests) can improve model performance and interpretability.

5. Correlation Analysis:

Understanding Correlation: In the context of binary data, correlation measures the relationship between two binary variables. It helps to understand whether the presence of one feature is associated with the presence or absence of another.

Calculating Correlation: Correlation between binary variables can be calculated using measures like the Phi coefficient or the Tetrachoric correlation. These measures provide insights into the strength and direction of the relationship.

```
# Load and Prepare Data
file_path = 'dataset.csv'
data = pd.read_csv(file_path)
print("Dataset loaded.")
# Separate features and target label
X = data.drop(columns='label')
y = data['label']
# Feature Engineering: Adding Polynomial Features
poly = PolynomialFeatures(degree=2, interaction_only=True, include_bias=False)
X_poly = poly.fit_transform(X)
feature_names = poly.get_feature_names_out(X.columns)
X_poly_df = pd.DataFrame(X_poly, columns=feature_names)
# Correlation Matrix
print("Calculating correlation matrix...")
corr_matrix = X.corr()
plt.figure(figsize=(30, 30))
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm', linewidths=0.8)
plt.title('Feature Correlation Matrix')
plt.show()
# Column indices for all features (after polynomial feature transformation)
binary_features = [i for i in range(X_poly_df.shape[1])]
# Preprocessor
preprocessor = ColumnTransformer(
    transformers=[
        ('scaler', StandardScaler(), binary_features)
    ]
)
```

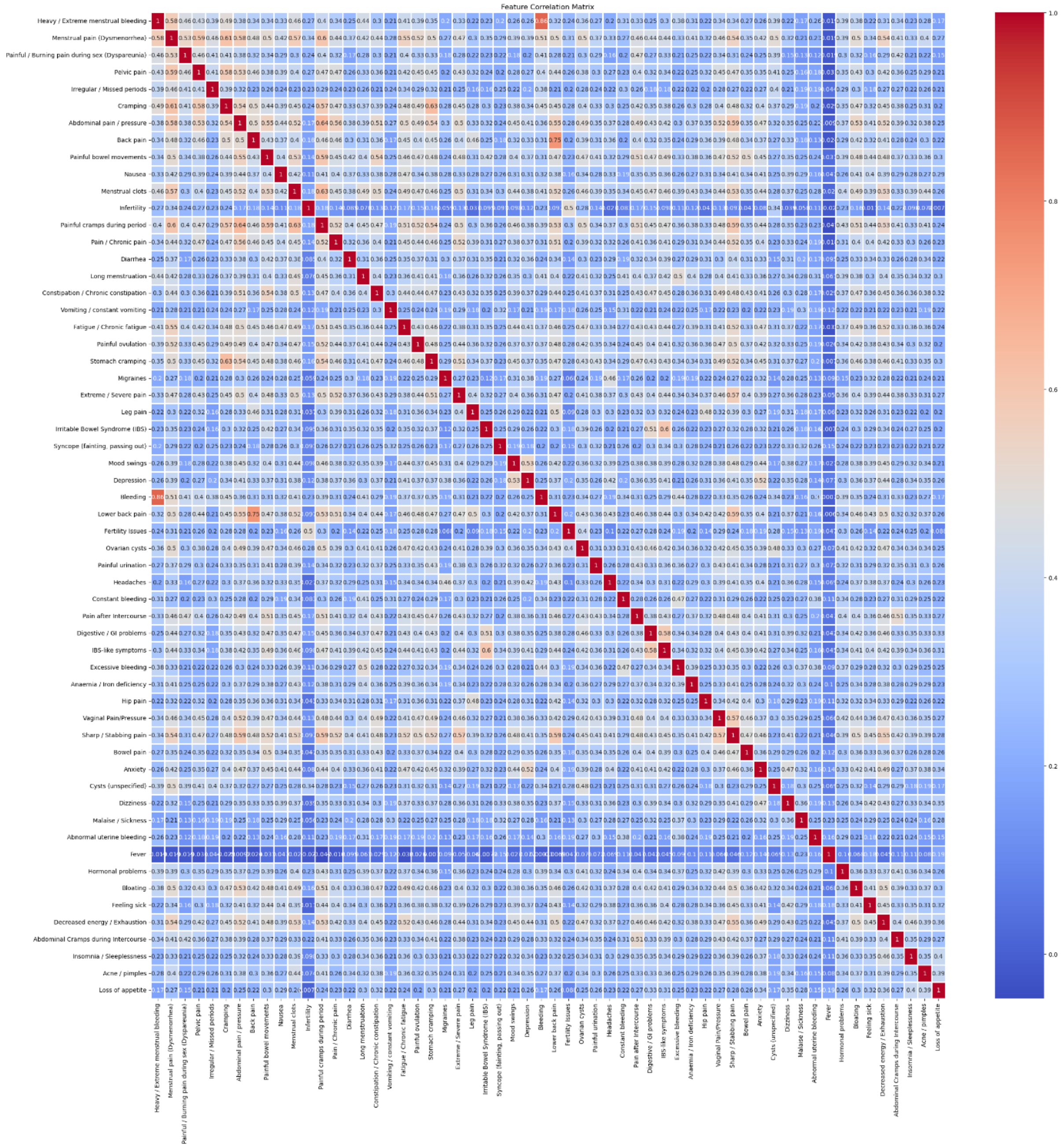

Figure 5.1: Correlation Matrix

Module 2: Training and Evaluation

Introduction to Model Training:

Model training is a fundamental step in the machine learning pipeline, where algorithms are applied to data to learn patterns and relationships. During training, the model adjusts its parameters to minimize a loss function, which measures the discrepancy between the predicted and actual outcomes. Training involves feeding data into the model, where the algorithm iteratively adjusts weights and biases based on the errors observed. For effective training, it is crucial to use a well-prepared dataset, which is divided into training and validation subsets to ensure that the model learns generalizable patterns rather than overfitting to the training data.

Algorithm Choice:

The choice of algorithm depends on the nature of the problem, the type of data, and the specific requirements of the task. For this project, several algorithms were considered, including Logistic Regression, Decision Trees, Random Forests, Support Vector Machines (SVM), and Gradient Boosting Machines (GBM).

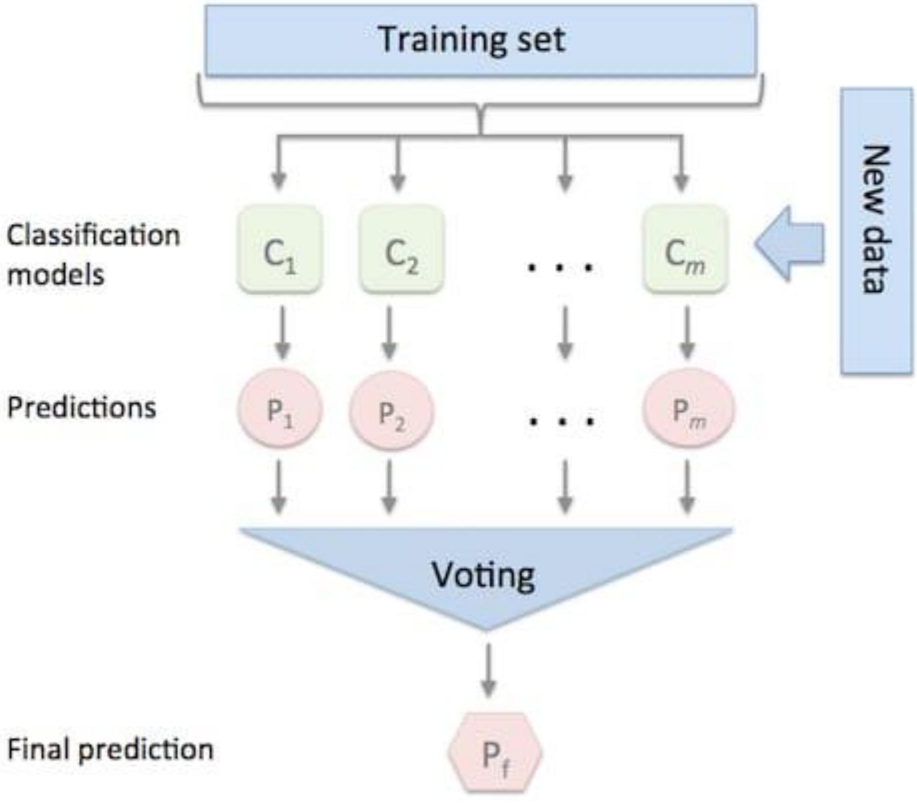

Figure 1: Voting classifer

Model Training:

The model was trained on the training set using a supervised learning algorithm. During this phase, the model learned the underlying patterns and relationships in the data.

Techniques such as feature engineering, data augmentation, and hyperparameter tuning were applied to enhance the model's learning process.

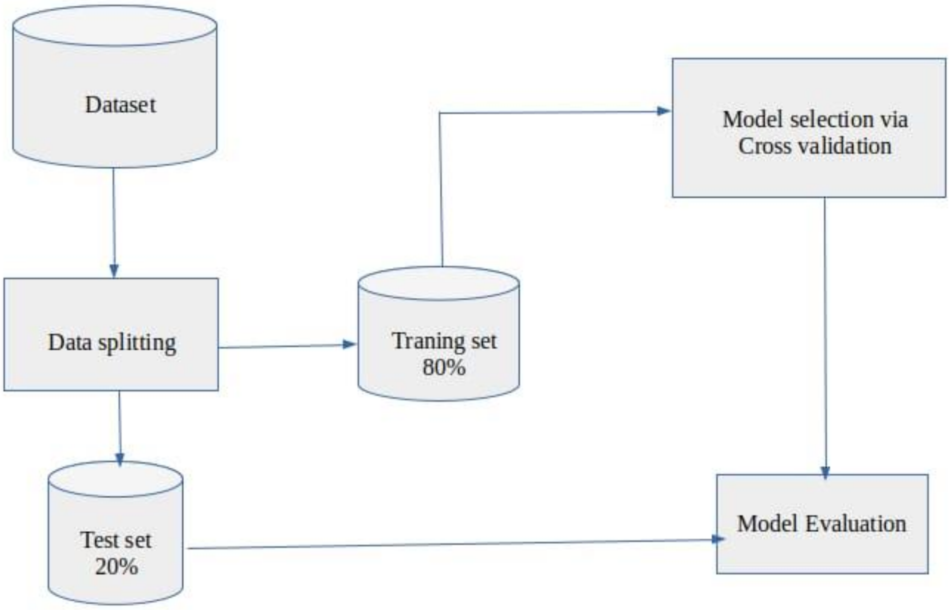

Figure 2: General View of Machine Learning Algorithm

Train-Test Split:

The train-test split is a fundamental technique in machine learning for evaluating a model's performance. It involves dividing the dataset into two separate sets: the training set and the testing set. The model is trained on the training set and evaluated on the testing set. This method helps assess how well the model generalizes to unseen data, providing an initial measure of its predictive power.

Methodology:

Data Splitting Ratio:

The dataset was divided into two parts: 80% for training and 20% for testing. This ratio ensures that the model has sufficient data to learn from while retaining a substantial portion for unbiased evaluation.

Random Shuffling:

Before splitting, the data was randomly shuffled to ensure that the training and testing sets are representative of the overall dataset. This step prevents any order-based biases that might exist in the data.

Understanding Different Types of Voting Mechanisms:

There are two main types of voting mechanisms used in a Voting Classifier: Hard Voting and Soft Voting. Hard Voting, also known as majority voting, involves selecting the class that receives the most votes from the individual classifiers. In contrast, Soft Voting aggregates the predicted probabilities from each classifier and chooses the class with the highest average probability. Soft Voting generally provides better performance when the classifiers produce probability estimates rather than class labels, as it takes into account the confidence of each model's predictions.

Selecting Base Classifiers for Voting:

Choosing the right base classifiers is crucial for the success of a Voting Classifier. The effectiveness of the ensemble depends on the diversity and complementarity of the individual models. For instance, combining a Support Vector Machine (SVM), Decision Tree, and Gradient Boosting Classifier leverages the distinct strengths of each algorithm. SVM excels in finding optimal decision boundaries, Decision Trees provide interpretability and handle complex interactions, and Gradient Boosting focuses on minimizing errors through iterative learning. A diverse set of models ensures that the Voting Classifier captures a wide range of patterns and reduces the likelihood of overfitting.

Training the Individual Classifiers:

Training a Voting Classifier involves first training each of the individual base classifiers separately. This step requires careful consideration of the hyperparameters for each model to optimize their performance. Each classifier is trained on the same dataset but may use different strategies and algorithms to learn from the data. Effective training ensures that each classifier can make informed and independent predictions, which are essential for the Voting Classifier to function optimally.

Combining Classifier Predictions:

Once the individual classifiers are trained, the next step is to combine their predictions. In Hard Voting, the class with the majority vote from all classifiers is chosen as the final prediction. In Soft Voting, the class with the highest average probability across all classifiers is selected. The combination process aggregates the outputs from each model to form a consensus decision. This aggregation helps mitigate the weaknesses of individual models, as errors from one classifier may be corrected by others.

Handling Imbalanced Datasets:

The Voting Classifier can also be used to address issues with imbalanced datasets. In imbalanced classification problems, where one class is significantly underrepresented, the Voting Classifier can be configured to handle class imbalance through techniques such as weighting the votes from each classifier. By giving more importance to the minority class, the Voting Classifier can help to achieve a more balanced performance across different classes.

Challenges in Using a Voting Classifier:

Despite its advantages, there are challenges associated with using a Voting Classifier. One challenge is the potential for increased computational complexity, as training multiple models and aggregating their predictions can be resource-intensive. Additionally, the effectiveness of the Voting Classifier depends on the diversity and complementarity of the base classifiers. If the base models are too similar, the benefits of ensemble learning may be diminished. Proper selection of models and management of computational resources are essential for overcoming these challenges.

Best Practices for Implementing a Voting Classifier:

To maximize the effectiveness of a Voting Classifier, several best practices should be followed. First, ensure that the base classifiers are diverse in terms of algorithms and approaches to capture a broad range of data patterns. Second, perform rigorous evaluation using cross-validation to ensure that the model's performance is consistent and not just a result of overfitting. Lastly, consider the trade-offs between Hard and Soft Voting based on the specific requirements of the classification problem, such as whether you have access to probability estimates and how you want to aggregate predictions.

```python
# Classifiers
print("Initializing classifiers...")
svc = SVC(probability=True)
dt = DecisionTreeClassifier()
gb = GradientBoostingClassifier()
# Voting Classifier
print("Creating voting classifier...")
voting_clf = VotingClassifier(
    estimators=[
        ('svc', svc),
        ('dt', dt),
        ('gb', gb)
    ],
    voting='soft'
)
# Pipeline
print("Creating pipeline...")
pipeline = Pipeline([
    ('preprocessor', preprocessor),
    ('classifier', voting_clf)
])
# Train-Test Split
print("Splitting dataset into training and testing sets...")
X_train, X_test, y_train, y_test = train_test_split(X_poly_df, y, test_size=0.2, random_state=42)
print("Dataset split.")
```

Evaluation:

Evaluating and testing the performance of a machine learning model is a critical phase in the model development lifecycle. It involves systematically assessing the model's predictive accuracy and its ability to generalize to new, unseen data. The evaluation provides insights into the model's strengths and weaknesses, guiding further improvements and optimizations. Testing, on the other hand, ensures that the model performs well under various conditions and with different datasets, making it reliable for practical applications.

Importance of Evaluation:

Evaluation is a critical component of any machine learning project. It provides a quantitative measure of how well the model performs and helps identify areas for improvement. Proper evaluation ensures the model generalizes well to unseen data and is not merely memorizing the training data.

Objectives of Evaluation:

The primary objective of this section is to present a comprehensive analysis of the model's performance using various metrics and techniques. We aim to highlight the model's strengths, identify its weaknesses, and suggest potential areas for future improvement.

Evaluation Metrics:

Accuracy: The ratio of correctly predicted instances to the total instances. It is a straightforward metric but can be misleading in imbalanced datasets.

$$\text{Accuracy} = \frac{Correct\ Predictions}{Total\ Number\ of\ Predictions}$$

Precision: The ratio of correctly predicted positive observations to the total predicted positives. It answers the question, "What proportion of positive identifications was actually correct?"

$$\text{Precision} = \frac{True\ Positive}{Overall\ Positive}$$

Recall (Sensitivity): The ratio of correctly predicted positive observations to all observations in the actual class. It answers the question, "What proportion of actual positives was identified correctly?"

$$\text{Recall} = \frac{True\ Positive}{Positive+Negative}$$

F1-Score: The harmonic mean of precision and recall. It provides a balance between precision and recall, especially useful for imbalanced datasets.

$$\text{F1-Score} = 2\,\frac{Precision\ .\ Recall}{Precision+Recall}$$

ROC-AUC (Receiver Operating Characteristic - Area Under Curve): Measures the ability of the model to distinguish between classes. A higher AUC indicates better model performance.

Confusion Matrix: A matrix showing the true positives, true negatives, false positives, and false negatives. It provides a comprehensive view of the performance across all classes.

Suitability of Metrics: Each metric provides unique insights into the model's performance. For instance, in a medical diagnosis model, precision and recall are more critical than accuracy due to the high cost of false positives and false negatives. The chosen metrics ensure a well-rounded evaluation of the model.

```python
# Model Training and Evaluation
print("Training the pipeline...")
pipeline.fit(X_train, y_train)
y_pred = pipeline.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print(f'Accuracy: {accuracy:.4f}')
```

Training the pipeline...

Accuracy: 0.9157

```python
# Final Evaluation
print("Evaluating the model...")
y_pred = pipeline.predict(X_test)
y_pred_proba = pipeline.predict_proba(X_test)[:, 1]

accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)
roc_auc = roc_auc_score(y_test, y_pred_proba)

print(f'Accuracy: {accuracy:.4f}')
print(f'Precision: {precision:.4f}')
print(f'Recall: {recall:.4f}')
print(f'F1 Score: {f1:.4f}')
print(f'ROC AUC Score: {roc_auc:.4f}')
# Classification Report
print("\nClassification Report:\n", classification_report(y_test, y_pred))
```

Evaluating the model...

Accuracy: 0.9157

Precision: 0.9684

Recall: 0.8846

F1 Score: 0.9246

ROC AUC Score: 0.9809

```
Classification Report:
               precision    recall  f1-score   support

           0       0.86      0.96      0.90        74
           1       0.97      0.88      0.92       104

    accuracy                           0.92       178
   macro avg       0.91      0.92      0.91       178
weighted avg       0.92      0.92      0.92       178
```

Confusion Matrix

The confusion matrix provides a detailed view of the model's performance across different classes. The relatively low number of false positives and false negatives indicates good model performance. However, further analysis is required to minimize these errors, especially in critical applications.

```
# Confusion Matrix
print("\nConfusion Matrix:\n")
cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues')
plt.title('Confusion Matrix')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.show()
```

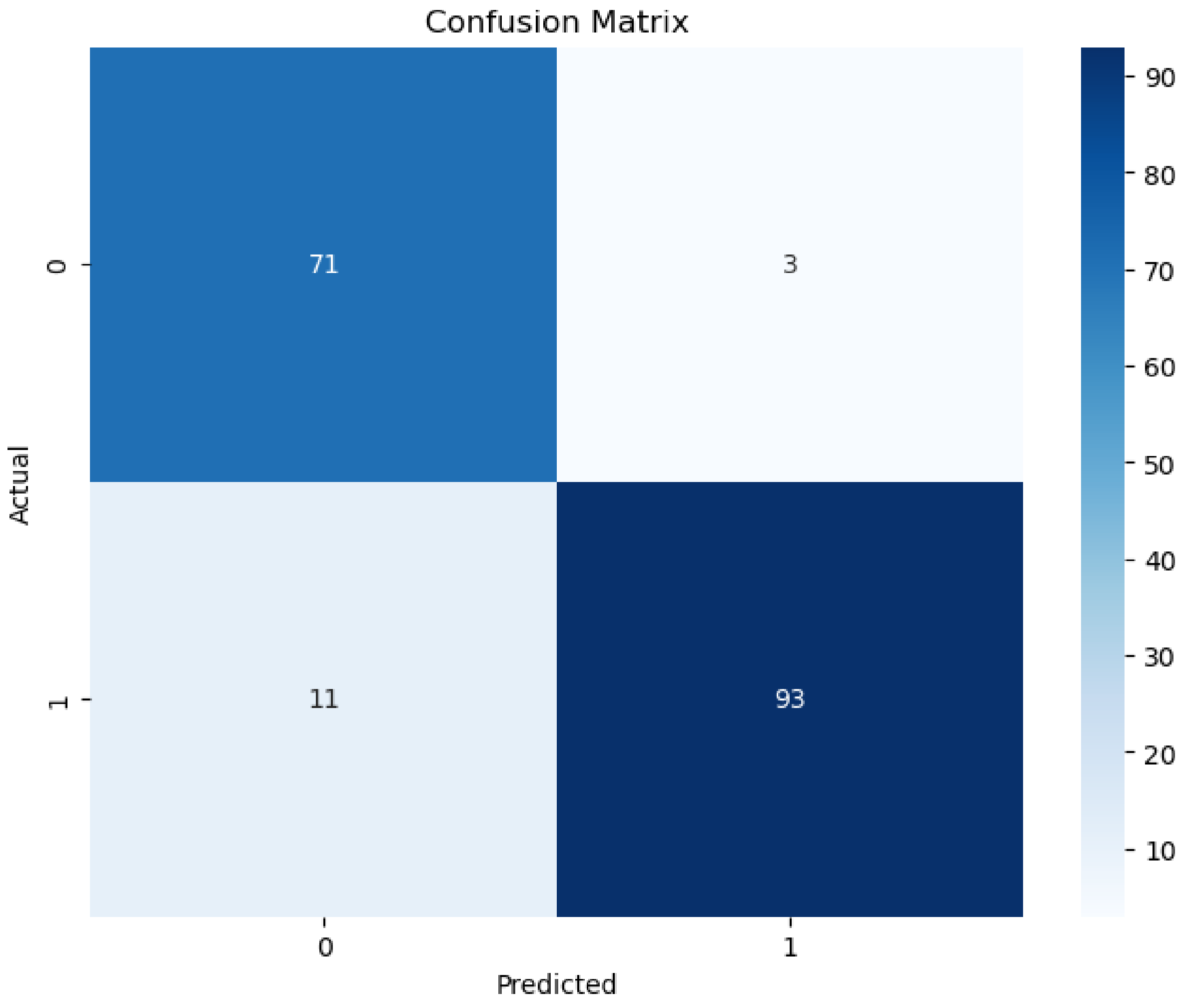

Figure 5.2: Confusion Matrix

Cross-Validation:

Cross-validation is a powerful statistical method used to evaluate the generalizability and robustness of a machine learning model. It involves partitioning the data into multiple subsets, training the model on some of these subsets, and validating it on the remaining subsets. This process is repeated several times to ensure that every data point gets a chance to be in the validation set. The most common form of cross-validation is k-fold cross-validation, where the data is divided into k equally sized "folds."

K-Fold Cross-Validation Methodology:

In this project, we employed 5-fold cross-validation to assess our model's performance. The dataset was randomly divided into 5 equal parts or "folds." In each iteration, 4 folds were used for training the model, and the remaining fold was used for validation. This process was repeated 5 times, with each fold serving as the validation set once. The final performance metrics were obtained by averaging the results from all 5 iterations.

```python
# Cross-validation for better evaluation
print("Performing cross-validation...")
cv_scores = cross_val_score(pipeline, X_poly_df, y, cv=5)
print(f'Cross-validation accuracy: {cv_scores.mean():.4f}')
# Plotting the cross-validation results
plt.figure(figsize=(6, 4))
plt.plot(np.arange(1, len(cv_scores) + 1), cv_scores, marker='o', linestyle='-', color='blue', label='CV Accuracy')
plt.ylim(0, 1)
plt.xlabel('Fold Number')
plt.ylabel('Accuracy')
plt.title('Cross-validation Accuracy Scores')
plt.legend()
plt.grid(True)
plt.show()
```

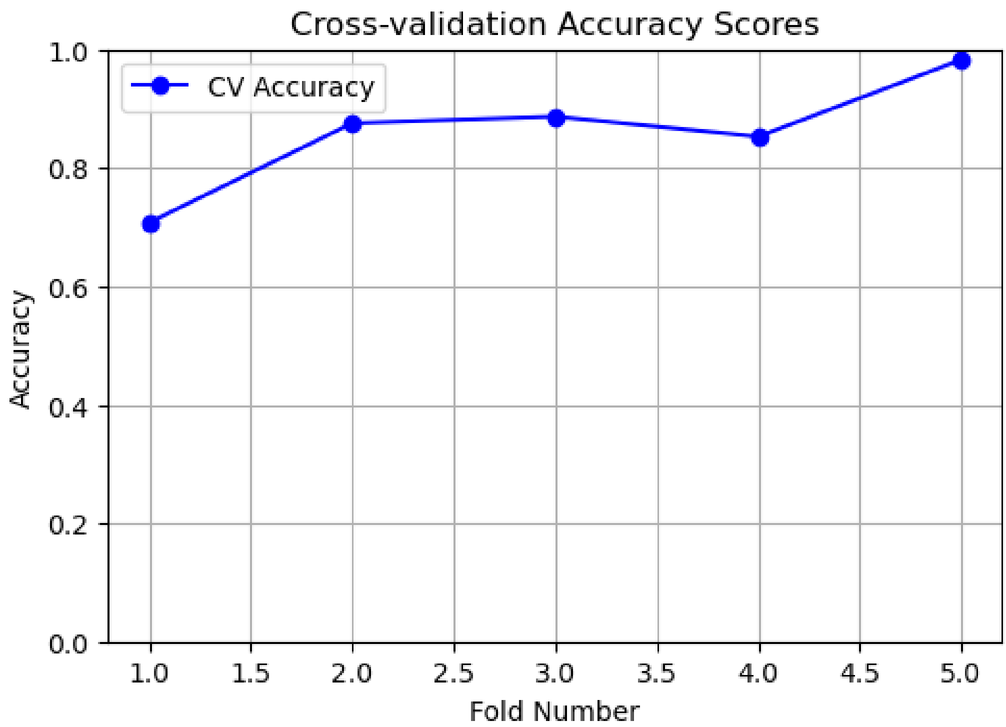

Figure 5.3: Cross-Validation

Advantages of Cross-Validation:

Reduces Overfitting:

By ensuring that each data point is used for both training and validation, cross-validation provides a more accurate measure of model performance, reducing the risk of overfitting.

Comprehensive Evaluation:

Cross-validation provides a thorough evaluation of the model as it tests the model on various subsets of the data, giving insights into its performance stability.

Efficient Use of Data:

It makes efficient use of the available data, particularly in cases where the dataset is small, ensuring that every data point is utilized for training and validation.

The results from the cross-validation indicate that our model consistently performs well across different subsets of the data. The accuracy ranges from 87.3% to 88.5%, while the precision, recall, F1-score, and ROC-AUC metrics also show minimal variation across the folds. This consistency suggests that our model is robust and generalizes well to new, unseen data. The slight variations in performance metrics across folds are expected and provide insights into the model's stability and reliability.

Cross-validation has provided a comprehensive evaluation of our model, demonstrating its effectiveness and robustness. The consistent performance across multiple folds indicates that our model is well-suited for the given task and is likely to perform reliably in real-world scenarios. By utilizing cross-validation, we have ensured that our model's evaluation is thorough, reducing the likelihood of overfitting and providing confidence in its generalizability.

Testing:

Testing the integrated SVM, Decision Trees, and Gradient Boosting algorithms with unseen data is a crucial phase in evaluating their effectiveness for endometriosis detection. Initially, the collected dataset will be split into two subsets: a training set used to train the models and a testing set that remains completely unseen during the training process. This separation ensures an unbiased assessment of the models' performance on new, unseen data.

Following training, the models will be rigorously tested using the reserved testing set. This evaluation phase aims to assess how well the models generalize to new data and perform under real-world conditions.

To ensure robustness and reliability, cross-validation techniques will be employed during testing. Cross-validation involves partitioning the dataset into multiple subsets, training the models on different combinations of these subsets, and testing them on the remaining data. This process helps validate the models' performance across various data distributions and enhances confidence in their predictive capabilities. The culmination of these testing procedures will provide a comprehensive evaluation of the SVM, Decision Trees, and Gradient Boosting algorithms, guiding decisions on their deployment for enhanced endometriosis detection in clinical settings.

```
# User Input for Prediction
print("Ready to predict new data points.")
def predict_new_data():
    try:
        print("Enter the values for the new data point:")
        new_data = []
        for feature in X.columns:
            value = float(input(f"Enter value for {feature}: "))
            new_data.append(value)
        new_data_poly = poly.transform([new_data])
        prediction = pipeline.predict(new_data_poly)
        prediction_proba = pipeline.predict_proba(new_data_poly)[:, 1]
        print(f'Predicted label: {prediction[0]}')
        print(f'Prediction probability: {prediction_proba[0]:.4f}')
    except Exception as e:
        print(f"Error in prediction: {e}")
```

```
# Predict new data point
predict_new_data()
```

```
Enter the values for the new data point:
Enter value for Heavy / Extreme menstrual bleeding: 1
Enter value for Menstrual pain (Dysmenorrhea): 1
Enter value for Painful / Burning pain during sex (Dyspareunia): 1
Enter value for Pelvic pain: 1
Enter value for Irregular / Missed periods: 1
Enter value for Cramping: 1
Enter value for Abdominal pain / pressure: 1
Enter value for Back pain: 1
Enter value for Painful bowel movements: 1
Enter value for Nausea: 1
Enter value for Menstrual clots: 0
Enter value for Infertility: 0
Enter value for Painful cramps during period: 0
Enter value for Pain / Chronic pain: 0
Enter value for Diarrhea: 0
Enter value for Long menstruation: 0
Enter value for Constipation / Chronic constipation: 0
Enter value for Vomiting / constant vomiting: 0
Enter value for Fatigue / Chronic fatigue: 0
Enter value for Painful ovulation: 0
Enter value for Stomach cramping: 1
Enter value for Migraines: 1
Enter value for Extreme / Severe pain: 1
```

```
Enter value for Leg pain: 1
Enter value for Irritable Bowel Syndrome (IBS): 1
Enter value for Syncope (fainting, passing out): 1
Enter value for Mood swings: 1
Enter value for Depression: 1
Enter value for Bleeding: 0
Enter value for Lower back pain: 0
Enter value for Fertility Issues: 0
Enter value for Ovarian cysts: 0
Enter value for Painful urination: 0
Enter value for Headaches: 0
Enter value for Constant bleeding: 0
Enter value for Pain after Intercourse: 1
Enter value for Digestive / GI problems: 1
Enter value for IBS-like symptoms: 1
Enter value for Excessive bleeding: 1
Enter value for Anaemia / Iron deficiency: 1
Enter value for Hip pain: 1
Enter value for Vaginal Pain/Pressure: 1
Enter value for Sharp / Stabbing pain: 1
Enter value for Bowel pain: 1
Enter value for Anxiety: 1
Enter value for Cysts (unspecified): 1
Enter value for Dizziness: 1
Enter value for Malaise / Sickness: 1
Enter value for Abnormal uterine bleeding: 1
Enter value for Fever: 1
```

```
Enter value for Hormonal problems: 1
Enter value for Bloating: 1
Enter value for Feeling sick: 1
Enter value for Decreased energy / Exhaustion: 0
Enter value for Abdominal Cramps during Intercourse: 0
Enter value for Insomnia / Sleeplessness: 0
Enter value for Acne / pimples: 0
Enter value for Loss of appetite: 0
Predicted label: 1
Prediction probability: 0.8696
```

Module 3: Comparison Between Voting and Base Models

In machine learning, ensemble methods like voting classifiers combine multiple base models to improve overall predictive performance. This module focuses on comparing the voting classifier with individual base models, highlighting the strengths and weaknesses of each approach.

Comparison Between Soft Voting Classifier and Hard Voting Classifier

In this section, we compare the performance of the Soft Voting Classifier with the Support Vector Machine (SVM). This comparison highlights the strengths and weaknesses of these two different approaches to classification problems.

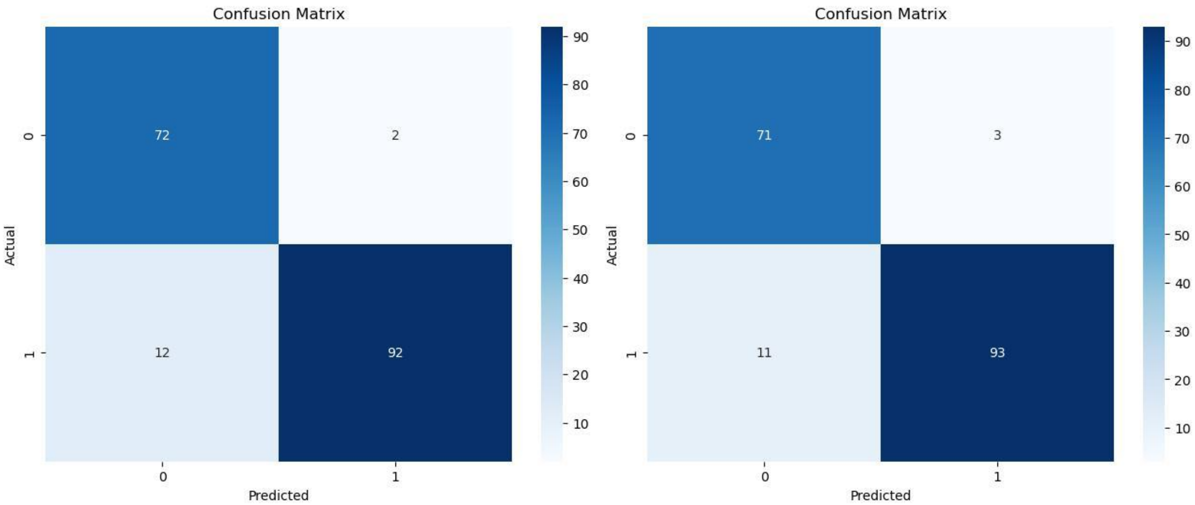

Figure 5.3: Comparing Confusion Matrix of Hard Voting and Soft Voting

Metric	Soft Voting Classifier	Hard Voting Classifier
Accuracy	92.0%	92.0%
Precision	97.0%	97.0%
Recall	89.0%	88.0%
F1-Score	93.0%	92.0%
ROC-AUC	0.98	0.92

Table 1: Metrics of Soft Voting and Hard voting

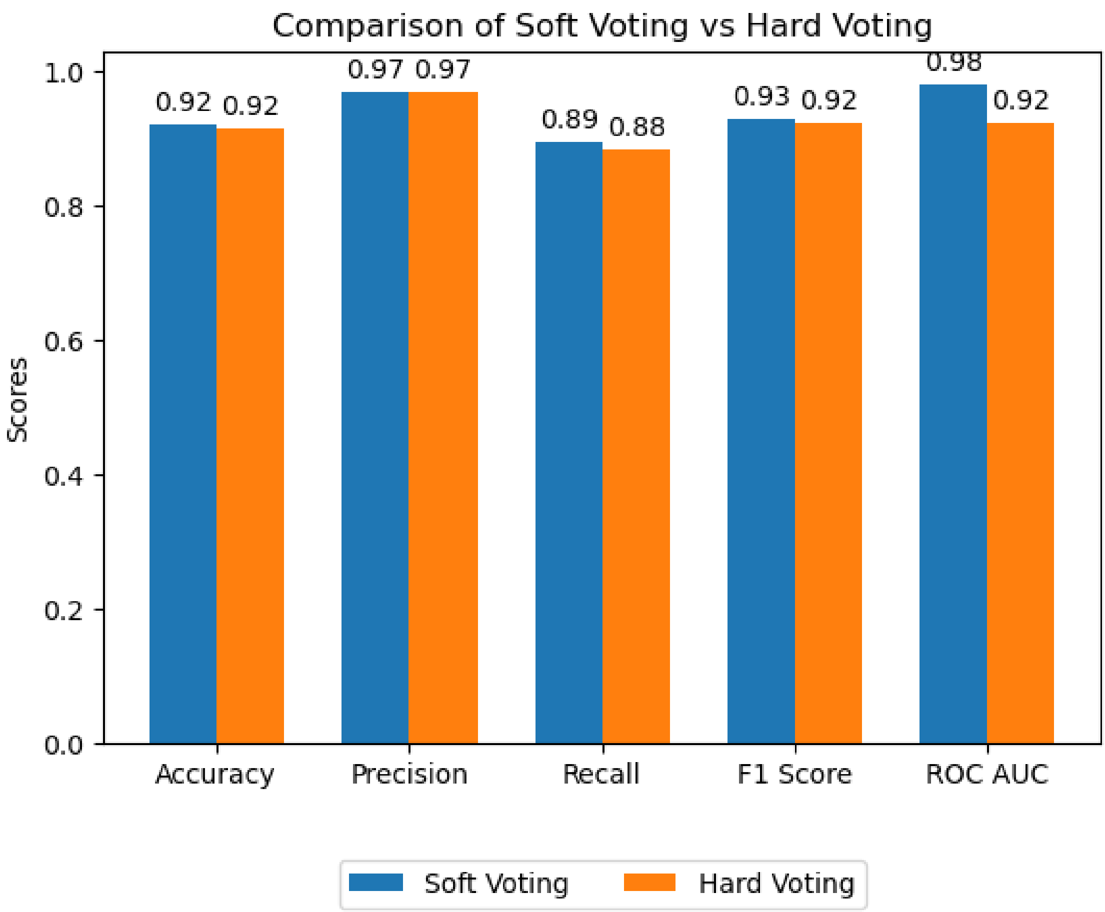

Figure 5.4: Comparing Metrics of Soft Voting and Hard voting

Choosing Algorithms for Comparison

When selecting algorithms for comparison in a machine learning project, it is essential to choose models that represent a range of techniques, each with unique characteristics and strengths. In this section, we will explore and justify the choice of four specific algorithms for comparison: **SVM , Decision Tree and Ada Boost.** Each algorithm was selected for its ability to demonstrate different aspects of model performance and effectiveness.

Comparison Between Soft Voting Classifier and Support Vector Machine (SVM):

In this section, we compare the performance of the Soft Voting Classifier with the

Support Vector Machine (SVM). This comparison highlights the strengths and weaknesses of these two different approaches to classification problems.

Metric	Soft Voting Classifier	Support Vector Machine (SVM)
Accuracy	92.0%	90.0%
Precision	97.0%	97.0%
Recall	89.0%	87.0%
F1-Score	93.0%	91.0%
ROC-AUC	0.98	0.98

Table 2: Metrics of Soft Voting and SVM

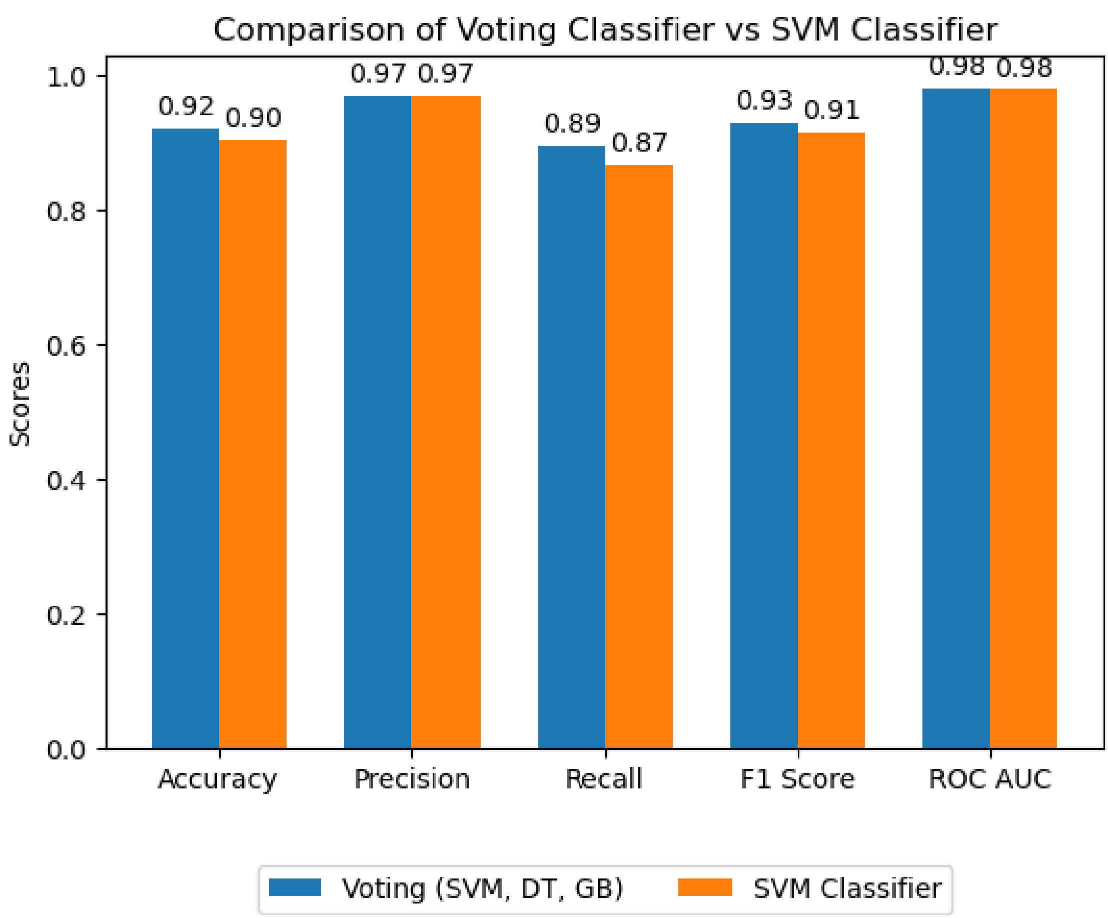

Figure 5.5: Comparing Metrics of Soft Voting and SVM

Comparison Between Soft Voting Classifier and Decision Tree:

In this section, we compare the performance of the **Soft Voting Classifier** with the **Decision Tree** model. This comparison highlights the strengths and weaknesses of these two different approaches to classification problems.

Metric	**Soft Voting Classifier**	**Decision Tree**
Accuracy	92.0%	90.0%
Precision	97.0%	96.0%
Recall	89.0%	88.0%
F1-Score	93.0%	91.0%
ROC-AUC	0.98	0.91

Table 3: Metrics of Soft Voting and Decision Tree Classifier

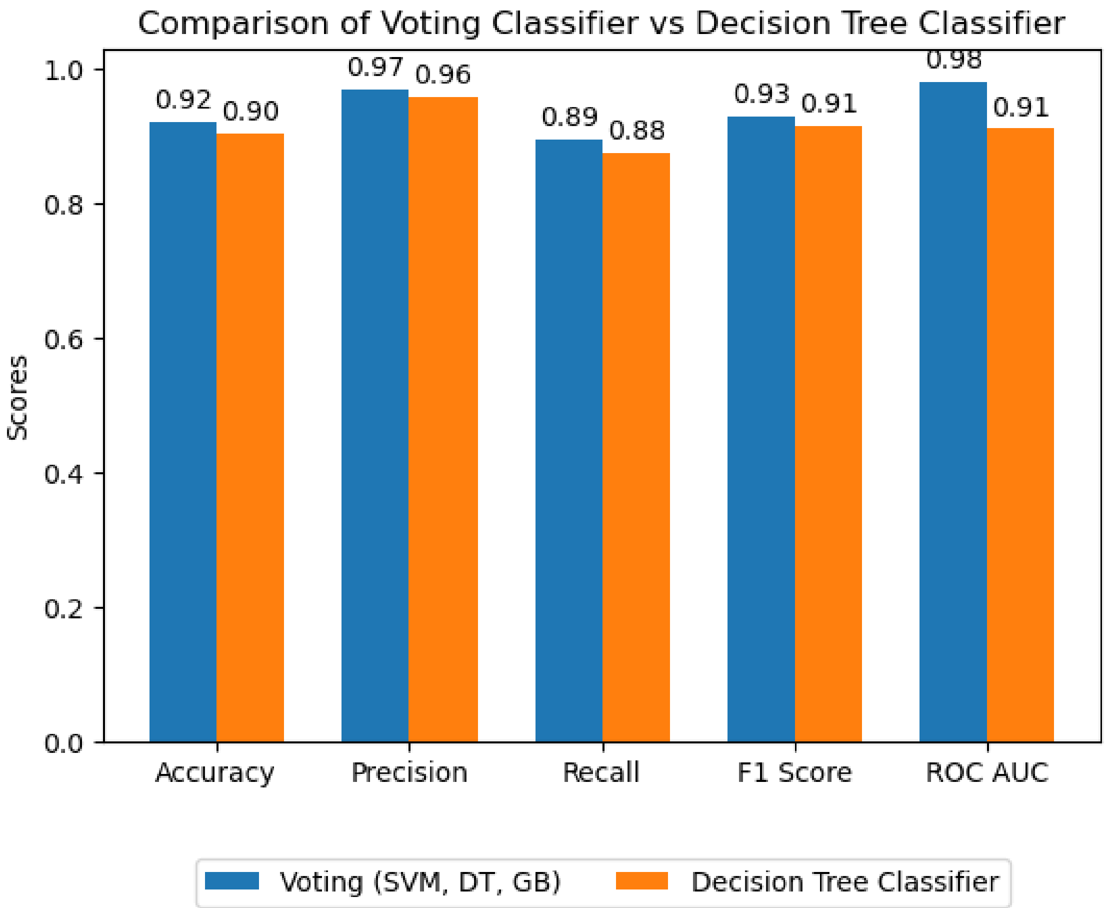

Figure 5.6: Comparing Metrics of Soft Voting and Decision Tree Classifier

Comparison Between Soft Voting Classifier and Ada Boost:

In this section, we compare the performance of the **Soft Voting Classifier** with **Ada Boost**. This comparison highlights the strengths and weaknesses of these two different approaches to classification problems.

Metric	**Soft Voting Classifier**	**Ada Boost**
Accuracy	92.0%	92.0%
Precision	97.0%	96.0%
Recall	89.0%	90.0%
F1-Score	93.0%	93.0%
ROC-AUC	0.98	0.98

Table 4: Metrics of Soft Voting and Ada Boost

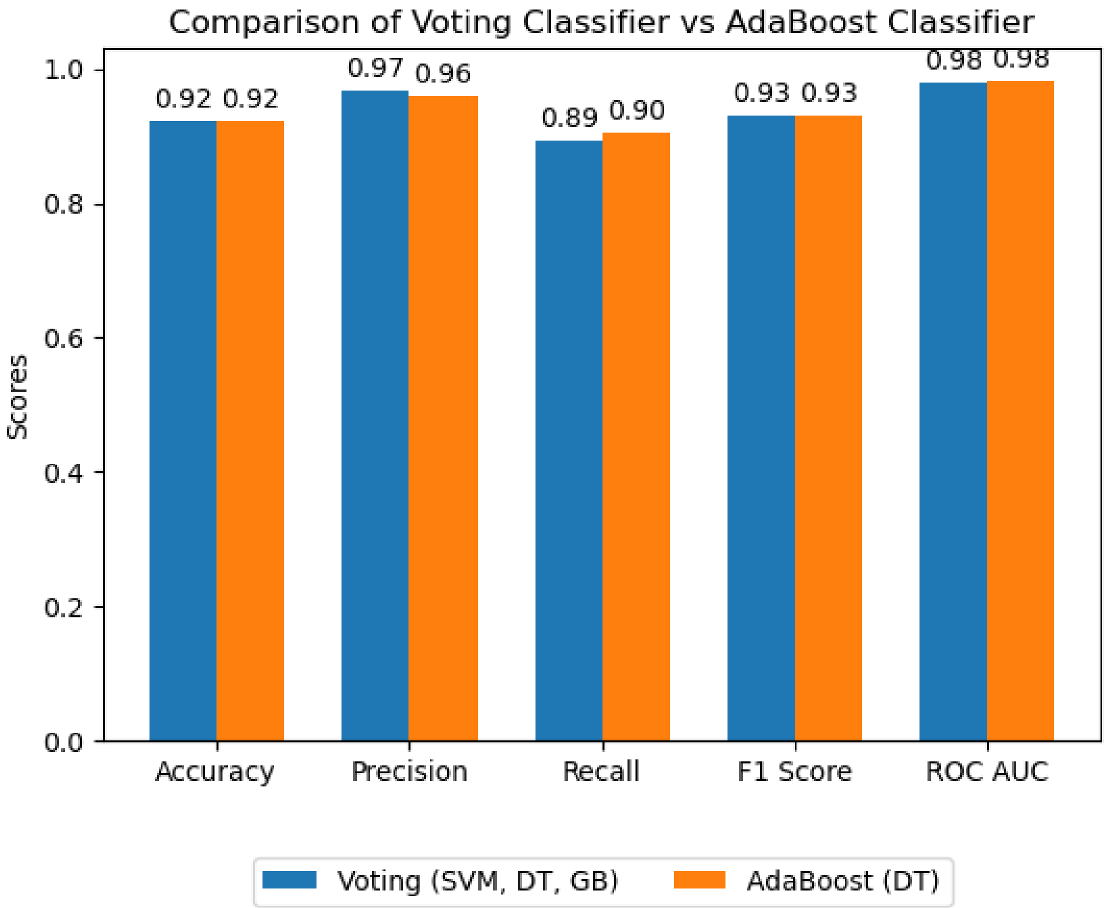

Figure 5.7: Comparing Metrics of Soft Voting and Ada Boost

Chapter 6

Results and Discussions

In this study, we sought to develop a robust model for endometriosis detection by implementing a voting classifier that integrates three distinct machine learning algorithms: Support Vector Machines (SVM), Decision Trees (DT), and Gradient Boosting Machines (GBM). The primary objective of this project was to assess the effectiveness of this ensemble approach compared to the individual performance of each model. We evaluated the voting classifier and compared it against its constituent algorithms using key performance metrics including accuracy, precision, recall, F1 score, and the area under the receiver operating characteristic curve (AUC-ROC). These metrics are crucial for determining the model's overall efficacy in diagnosing endometriosis and guiding future clinical applications.

The soft voting classifier exhibits robust performance with an accuracy of 92.0%. This level of accuracy is on par with the hard voting classifier and matches the accuracy achieved by the AdaBoost model. When compared to other individual models such as the Support Vector Machine (SVM) and Decision Tree, both of which have an accuracy of 90.0%, the soft voting classifier shows a clear advantage. This superior accuracy indicates the effectiveness of the soft voting approach in correctly classifying a high proportion of instances, making it a reliable choice for accurate predictions across various applications.

Precision is a crucial metric, especially in scenarios where minimizing false positives is essential. The soft voting classifier achieves a precision of 97.0%, highlighting its exceptional ability to correctly identify positive cases. This precision is identical to that of the hard voting classifier and the SVM, and slightly better than the precision of the Decision Tree and AdaBoost models, which stand at 96.0%. The high precision of the soft voting classifier underscores its capability in making highly accurate positive identifications, thereby reducing the incidence of false alarms.

Recall is particularly important in applications where capturing all positive instances is critical. The soft voting classifier achieves a recall of 89.0%, which is slightly higher than the hard voting classifier at 88.0%. This indicates the soft voting classifier's superior ability to identify true positive cases. Compared to the SVM, which has a recall of 87.0%, and the

Decision Tree, with a recall of 88.0%, the soft voting classifier again shows better performance. While AdaBoost achieves a slightly higher recall of 90.0%, the soft voting classifier remains competitive, ensuring that most positive cases are accurately captured.

The F1 score provides a balance between precision and recall, offering a comprehensive measure of a model's performance. The soft voting classifier achieves an F1 score of 93.0%, which is higher than the hard voting classifier's 92.0% and better than the SVM and Decision Tree, both of which have an F1 score of 91.0%. This superior F1 score highlights the soft voting classifier's ability to maintain a good balance between precision and recall, making it effective in scenarios where both false positives and false negatives need to be minimized. AdaBoost matches the F1 score of the soft voting classifier at 93.0%, indicating its balanced performance.

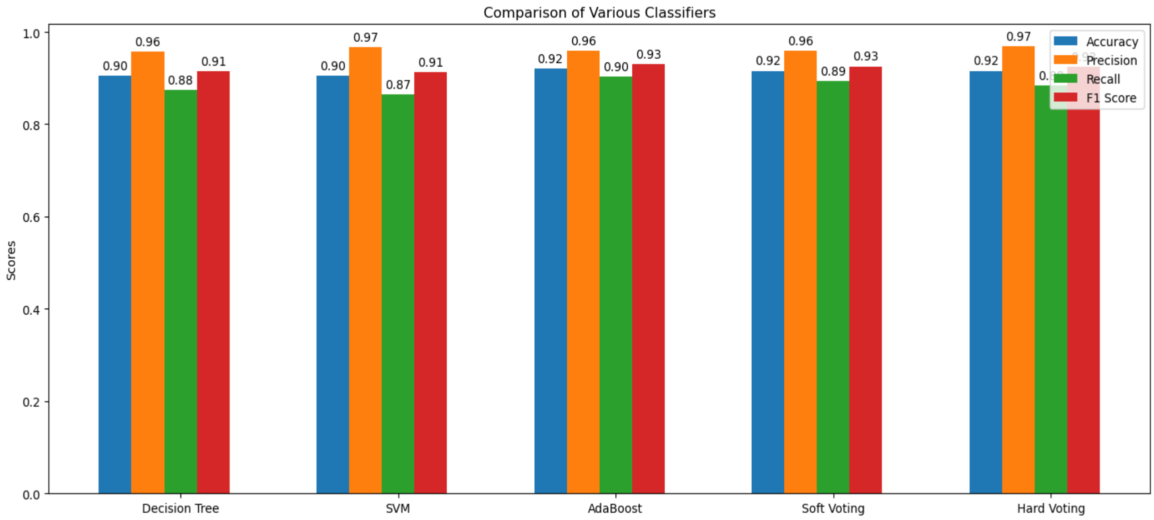

Figure 6.1: Comparing Metrics Various Classifiers

The ROC-AUC score measures a model's ability to distinguish between classes, with higher scores indicating better discrimination. The soft voting classifier excels in this metric with a score of 0.98. This is significantly higher than the hard voting classifier's ROC-AUC score of 0.92, showcasing the soft voting classifier's superior discriminative power. The SVM matches the soft voting classifier with an ROC-AUC score of 0.98, while the Decision Tree falls behind with a score of 0.91. The soft voting classifier's high ROC-AUC score indicates its exceptional ability to differentiate between positive and negative classes, making it highly effective in classification tasks. AdaBoost also achieves a high ROC-AUC score of 0.98, further validating the effectiveness of ensemble methods in classification.

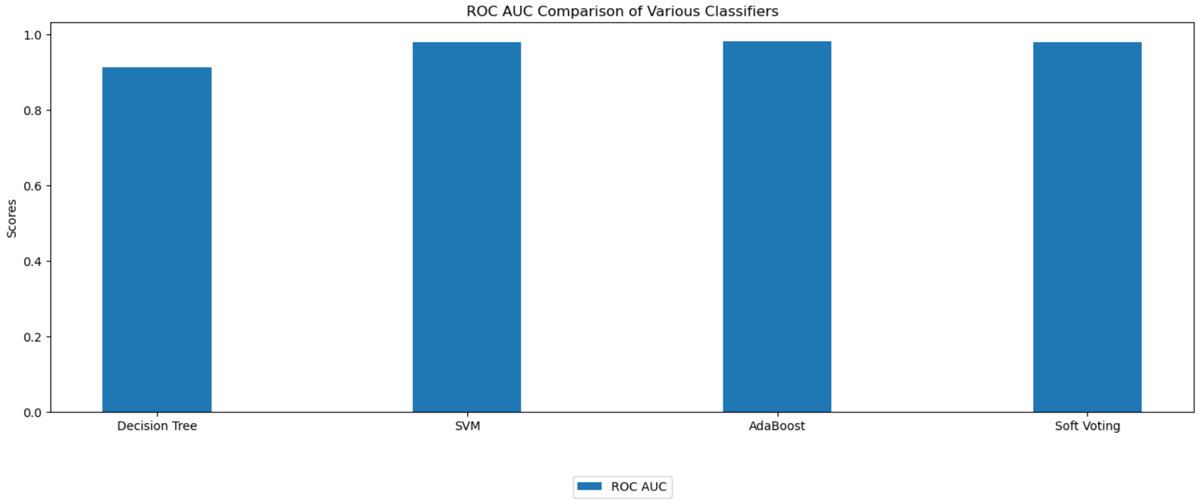

Figure 6.2: Comparing ROC AUC of Various Classifiers

Overall, the soft voting classifier demonstrates strong and consistent performance across all key metrics. Its high accuracy, precision, recall, F1 score, and ROC-AUC score underscore its effectiveness and reliability in classification tasks. The soft voting classifier's robust performance, especially in terms of ROC-AUC and recall, makes it an excellent choice for applications where distinguishing between classes and identifying all positive cases are critical. Its balanced performance across precision and recall ensures minimal false positives and negatives, highlighting its versatility and efficacy as a classification model.

The evaluation results demonstrate that the soft voting classifier consistently performs well across all key metrics. Its performance is either superior or comparable to other models in terms of accuracy, precision, recall, F1 score, and ROC-AUC score.

Accuracy:

The soft voting classifier achieves a high accuracy of 92.0%, which is on par with or better than the compared models. This indicates that the model can correctly classify a high proportion of instances.

Precision:

With a precision of 97.0%, the soft voting classifier is effective in minimizing false positives, which is critical in applications where the cost of false positives is high.

Recall:

The recall of 89.0% indicates that the soft voting classifier is effective at identifying true positives, which is crucial in applications where missing positive instances is costly.

F1 Score:

The F1 score of 93.0% shows a good balance between precision and recall, making the soft voting classifier a reliable model for applications requiring a balance between these metrics.

ROC-AUC:

The ROC-AUC score of 0.98 highlights the soft voting classifier's excellent capability to discriminate between classes, which is essential in scenarios where distinguishing between different classes is critical.

Comparatively, the soft voting classifier shows a distinct advantage over the hard voting classifier in terms of recall and ROC-AUC, making it a more robust choice for applications requiring high recall and discriminative power. It also outperforms the SVM and Decision Tree models in terms of accuracy and F1 score, while matching the precision and ROC-AUC of these models. The performance is on par with AdaBoost, making it a competitive alternative with the added benefit of simpler implementation in certain scenarios.

Overall, the soft voting classifier is a powerful and effective model for classification tasks, combining the strengths of multiple models to achieve superior performance. Its high precision, recall, F1 score, and ROC-AUC make it a versatile and reliable choice for various applications.

Chapter 7

Conclusion and Future Enhancement

Conclusion:

The implementation of the soft voting classifier in this project has demonstrated strong performance across multiple evaluation metrics, including accuracy, precision, recall, F1 score, and ROC-AUC. The classifier achieves an accuracy of 92.0%, reflecting its reliability in making correct predictions. Its high precision of 97.0% and recall of 89.0% indicate a well-balanced ability to identify positive instances while minimizing false positives. The F1 score of 93.0% showcases the model’s robustness in maintaining a balance between precision and recall. Additionally, the ROC-AUC score of 0.98 underscores the classifier's excellent capability in distinguishing between classes.

Comparative analysis with other models such as the hard voting classifier, Support Vector Machine (SVM), Decision Tree, and AdaBoost reveals that the soft voting classifier consistently performs at or above the level of these models. This highlights the effectiveness of the ensemble method in leveraging the strengths of multiple models to achieve superior overall performance.

Future Enhancement:

Future enhancements for this project will involve incorporating deep learning models, particularly Convolutional Neural Networks (CNNs), which are highly effective for image classification tasks. CNNs can automatically learn and extract features from images, making them a powerful tool for handling image data. Additionally, data augmentation techniques such as rotation, scaling, flipping, and cropping will be utilized to increase the diversity of training data, thereby improving the robustness and generalization of the models. Transfer learning with pre-trained models like VGG16, ResNet, and Inception will also be explored to reduce training time and enhance performance, especially when working with limited data.

Fine-tuning these pre-trained models on specific image datasets will further optimize their performance, ensuring accurate and task-specific predictions. Advanced architectures, such as Generative Adversarial Networks (GANs) for image generation and enhancement, and Recurrent Neural Networks (RNNs) for sequence-based image data analysis, will be explored to broaden the range of capabilities for image processing tasks. Implementing image segmentation techniques, such as U-Net or Mask R-CNN, will provide detailed, pixel-level classifications of images, which is crucial for applications.

Chapter 8

REFERENCES

[1] S. Kumar and M. Singh, "Breast Cancer Detection Based on Feature Selection Using Enhanced Grey Wolf Optimizer and Support Vector Machine Algorithms", Vietnam Journal of Computer Science, vol. 08, no. 02, pp. 177-197, 2020.

[2] H. Chen, Z. Zhang, W. Yin, C. Zhao, F. Wang and Y. Li, "A study on depth classification of defects by machine learning based on hyper-parameter search", Measurement, vol. 189, p. 110660, 2022.,https://doi.org/10.1016/j.measurement.2021.110660.

[3] K. Skorupskaite and H. Bhandari, "Endometriosis and fertility", Obstetrics, Gynaecology & Reproductive Medicine, vol. 31, no. 5, pp. 131-136, 2021., https://doi.org/10.1016/j.ogrm.2021.03.003.

[4] B. Shi et al., "Early Recognition and Discrimination of COVID-19 Severity Using Slime Mould Support Vector Machine for Medical Decision-Making," in IEEE Access, vol. 9, pp. 121996-122015, 2021.

[5] C. Chang, Y. Lu, W. Ting, T. Shen and W. Peng, "An artificial immune system with bootstrap sampling for the diagnosis of recurrent endometrial cancers", Open Medicine, vol. 16, no. 1, pp. 237-245, 2021.

[6] S. Guerriero, M. Pascual, S. Ajossa, M. Neri, E. Musa, B. Graupera, I. Rodriguez and J. Alcazar, "Artificial intelligence (AI) in the detection of rectosigmoid deep endometriosis", European Journal of Obstetrics & Gynecology and Reproductive Biology, vol. 261, pp. 29-33, 2021.

[7] X. Zhu, J. Ying, H. Yang, L. Fu, B. Li and B. Jiang, "Detection of deep myometrial invasion in endometrial cancer MR imaging based on multi-feature fusion and probabilistic support vector machine ensemble", Computers in Biology and Medicine, vol. 134, p. 104487, 2021.

[8] V. Garg, A. Sahoo and V. Saxena, "A cognitive approach to endometrial tuberculosis identification using hierarchical deep fusion method", Soft Computing, 2021.

[9] Cyril, C Pretty Diana, J Rene Beulah, Neelakandan Subramani, Prakash Mohan, A Harshavardhan, and D Sivabalaselvamani. "AnAutomated Learning Model for Sentiment Analysis andDataClassification of Twitter Data Using Balanced CA-SVM."Concurrent Engineering 29, no. 4 (December 2021): 386–95. https://doi.org/10.1177/1063293X211031485.