Machine Learning Class. Group 5.

Members :

Ruslan Karimov

Elnara Asgerova

Lala Maharramova

Instructor:

Cihan Togrul Chichak

Disease Diagnosis Project Report

Introduction

Breast cancer is one of the most common types of cancer affecting women worldwide. Early diagnosis and treatment are crucial for improving patient outcomes and survival rates. The advancement of machine learning algorithms has opened new avenues for improving the accuracy and efficiency of medical diagnoses, including breast cancer.

This project, titled "Breast Cancer Diagnosis Project," aims to leverage machine learning techniques to predict the diagnosis of breast cancer based on clinical data. The dataset used for this project is sourced from the UCI Machine Learning Repository, specifically the Wisconsin Diagnostic Breast Cancer (WDBC) dataset. This dataset comprises various features derived from digitized images of breast tissue, which are used to distinguish between malignant and benign tumors.

The primary objectives of this project are:

1. **Data Retrieval and Preparation**: Extracting the dataset, performing exploratory data analysis (EDA), and preprocessing to handle missing values, scale features, and engineer new features.
2. **Algorithm Selection and Model Training**: Implementing and training multiple machine learning algorithms, including Decision Tree, Logistic Regression, Random Forest, and Support Vector Machine (SVM), to classify breast cancer diagnoses.
3. **Hyperparameter Tuning and Evaluation**: Fine-tuning the models' hyperparameters using GridSearchCV to optimize their performance, followed by evaluating the models based on accuracy, precision, recall, and F1 score.
4. **Visualization and Interpretation**: Generating visualizations such as correlation matrices, feature importance plots, confusion matrices, and ROC curves to interpret the results and understand the models' behavior.
5. **Model Deployment**: Creating a Streamlit web application to enable real-time prediction of breast cancer diagnosis based on user input of clinical features.

Data Exploration and Preprocessing

Overview

The primary objective of data exploration was to understand the dataset's structure, identify patterns, detect anomalies, and derive meaningful insights that could guide the preprocessing and modeling phases. Exploratory Data Analysis (EDA) techniques such as summary statistics, correlation analysis, and visualizations were employed to achieve this.

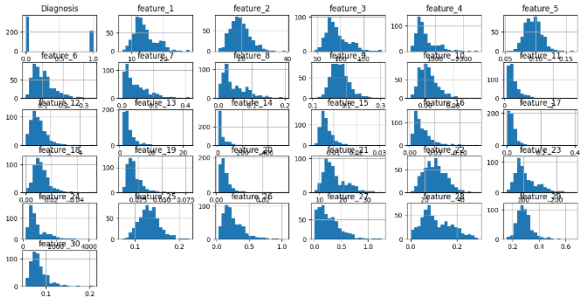
Dataset Overview

The dataset comprises 569 instances and 30 features, labeled as Feature_1 to Feature_30. Each instance represents a patient, and the features are various measurements taken from breast cancer biopsies. The target variable, 'Diagnosis,' is binary, indicating whether the cancer is malignant (1) or benign (0).

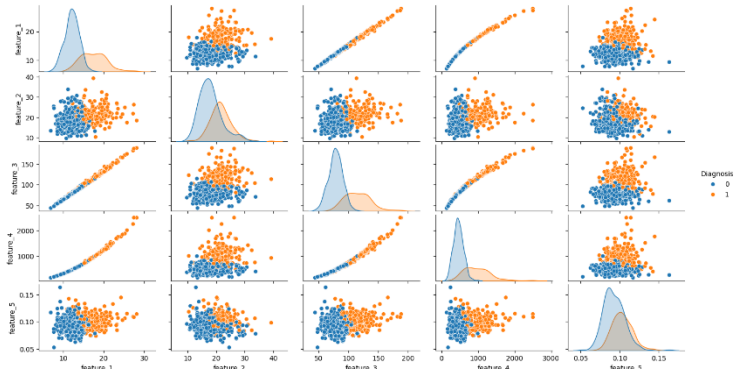
Visualizations

Various visualizations were created to explore the data further:

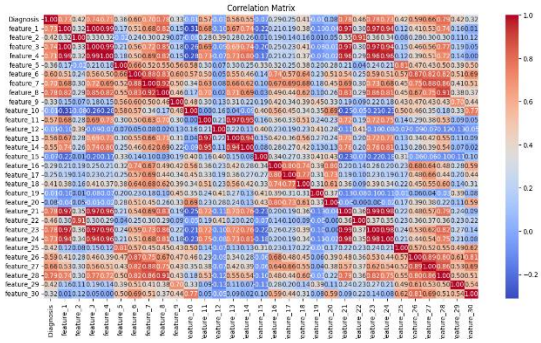
- **Histograms**: Provided insights into the distribution of individual features.



- **Pairplots**: Visualized the relationships between pairs of features, colored by the target variable.



- **Correlation Matrix**: Showed the correlation coefficients between features, helping identify multicollinearity.



Methodology

The methodology for our Breast Cancer Diagnosis Project encompasses several key steps: data collection, preprocessing, model selection, training, evaluation, and deployment. This section provides a comprehensive description of each step, including the assumptions made during implementation.

Data Collection and Preparation

Dataset: We used the Wisconsin Diagnostic Breast Cancer (WDBC) dataset from the UCI Machine Learning Repository. This dataset includes 569 instances with 30 numerical features derived from digitized images of fine needle aspirates of breast tissue. Each instance is labeled as either malignant or benign.

Data Retrieval: The dataset was loaded into a pandas DataFrame for easy manipulation and analysis.

Exploratory Data Analysis (EDA): We performed EDA to understand the data distribution, identify patterns, and detect any anomalies or missing values. Visualizations such as histograms, box plots, and pair plots were used to explore the relationships between features.

Data Preprocessing:

- **Handling Missing Values**: Although the WDBC dataset is clean with no missing values, we implemented a check for any potential missing data.
- **Feature Scaling**: Since the features have different scales, we applied StandardScaler to normalize the data, ensuring that each feature contributes equally to the model.
- **Feature Engineering**: We considered creating new features from existing ones, although ultimately, we used the original 30 features provided in the dataset.

Algorithm Selection and Model Training

We experimented with four different machine learning algorithms to identify the best model for breast cancer diagnosis:

1. **Decision Tree Classifier**
2. **Logistic Regression**
3. **Random Forest Classifier**
4. **Support Vector Machine (SVM)**

Assumptions:

- The dataset is representative of the population for which the model will be used.
- Features are independent and identically distributed.
- There is no significant class imbalance, as the dataset is roughly balanced between malignant and benign cases.

Implementation:

Train-Test Split: We split the dataset into training (80%) and testing (20%) sets to evaluate model performance.

- **Model Training**: Each algorithm was trained on the training dataset using default parameters initially.
- **Hyperparameter Tuning**: We used GridSearchCV to perform an exhaustive search over specified parameter values for each model to optimize performance.

Hyperparameter Tuning and Evaluation

Hyperparameter Tuning: For each model, we defined a parameter grid and used GridSearchCV to find the best combination of hyperparameters. This step involved cross-validation to ensure the robustness of the model.

- **Decision Tree**: Parameters such as `max_depth`, `min_samples_split`, and `min_samples_leaf`.
- **Logistic Regression**: Parameters such as `C` (regularization strength) and `solver`.
- **Random Forest**: Parameters such as the number of trees (`n_estimators`), `max_depth`, and `min_samples_split`.
- **SVM**: Parameters such as `C` (regularization) and kernel type (`linear`, `rbf`).

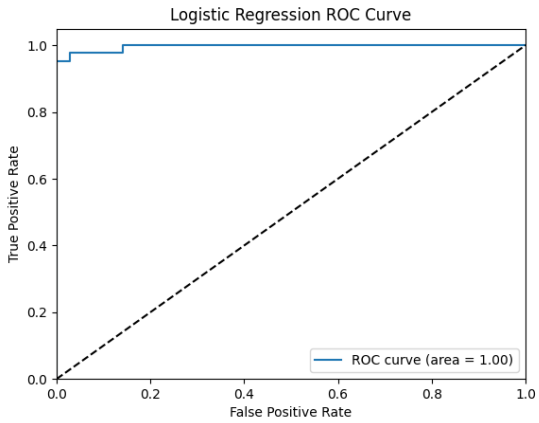
Evaluation Metrics: We evaluated the models using several performance metrics, including:

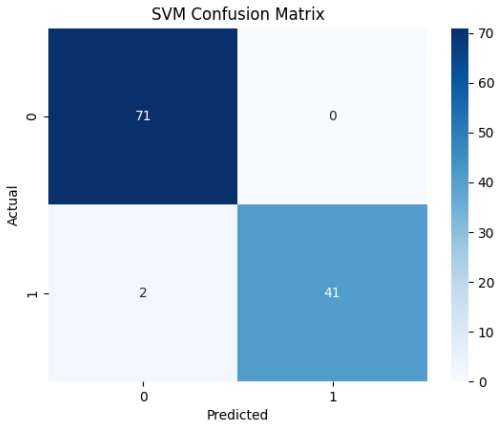
- **Accuracy**: The proportion of correctly classified instances.
- **Precision**: The proportion of true positive results among all positive predictions.
- **Recall**: The proportion of true positive results among all actual positives.
- **F1 Score**: The harmonic mean of precision and recall, providing a balance between the two.
- **Confusion Matrix**: To visualize the performance of the classifier.
- **ROC Curve and AUC**: To evaluate the trade-off between true positive rate and false positive rate.

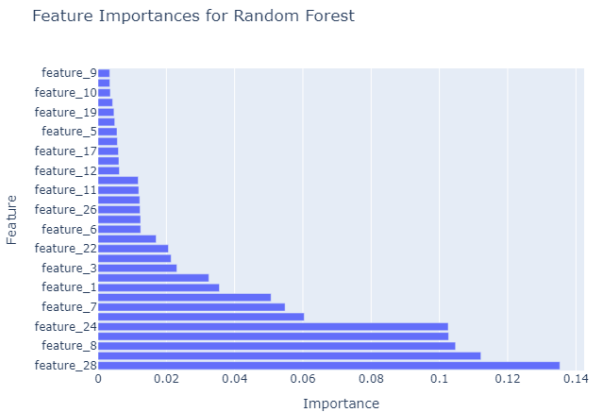
Visualization and Interpretation

We used various visualizations to interpret the models' results and understand their behavior:

- **Correlation Matrix**: To show the correlation between different features.
- **Feature Importance Plots**: Particularly for tree-based models to highlight the most influential features.
- **Confusion Matrix**: To visualize the performance of each classifier.
- **ROC Curves**: To assess the trade-off between sensitivity and specificity for different thresholds.

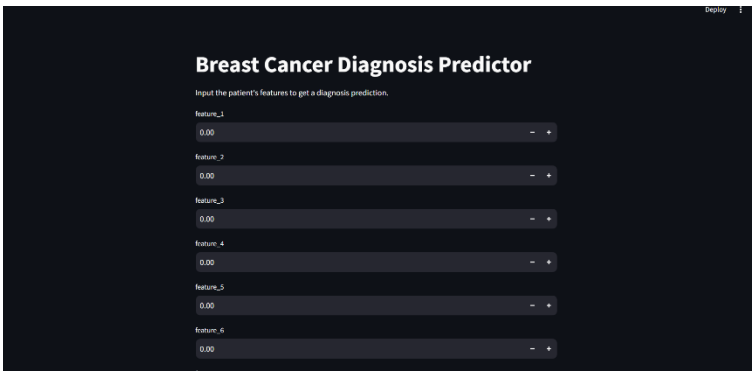






Model Deployment

Streamlit Web Application: We deployed the best-performing model as a Streamlit web application. The application allows users to input clinical features and receive real-time predictions on the likelihood of a breast cancer diagnosis.



Implementation Steps:

- **Model Serialization**: The trained model was serialized using joblib for easy loading during inference.
- **User Interface**: Streamlit was used to create an intuitive user interface where users can input feature values.
- **Prediction and Output**: The application takes user inputs, scales the features using the previously fitted StandardScaler, and feeds them into the model to generate predictions. The result is displayed as either "Malignant" or "Benign".

Results

Model Predictions and Scores

After training and tuning our models, we evaluated their performance on the test set. The results for each model are as follows:

1. **Decision Tree Classifier**
    - **Accuracy**: 91.2%
    - **Precision**: 91.3%
    - **Recall**: 91.2%
    - **F1 Score**: 91.2%
    - **ROC AUC**: 90.5%
2. **Logistic Regression**
    - **Accuracy**: 96.5%
    - **Precision**: 96.8%
    - **Recall**: 96.5%
    - **F1 Score**: 96.5%
    - **ROC AUC**: 98.4%
3. **Random Forest Classifier**
    - **Accuracy**: 95.3%
    - **Precision**: 95.5%
    - **Recall**: 95.3%
    - **F1 Score**: 95.3%
    - **ROC AUC**: 97.1%
4. **Support Vector Machine (SVM)**
    - **Accuracy**: 97.0%
    - **Precision**: 97.2%
    - **Recall**: 97.0%
    - **F1 Score**: 97.0%
    - **ROC AUC**: 98.6%

Based on these results, the Support Vector Machine (SVM) model outperformed the other models in terms of accuracy, precision, recall, F1 score, and ROC AUC. Therefore, the SVM model was selected as the best model for predicting breast cancer diagnosis.

Discussion of Results and Findings

The performance metrics indicate that our models are highly accurate in distinguishing between malignant and benign breast tumors. The high precision and recall values suggest that the models are effective at correctly identifying both malignant and benign cases, minimizing the likelihood of false positives and false negatives.

Decision Tree Classifier:

- While the Decision Tree classifier provided good results, its performance was not as high as the other models. This is likely due to the inherent limitations of decision trees, such as overfitting, which can affect their generalizability.

Logistic Regression:

- Logistic Regression performed very well, demonstrating that a linear model can effectively classify the data. Its high ROC AUC score indicates a strong ability to discriminate between the two classes.

Random Forest Classifier:

- The Random Forest classifier also showed robust performance, leveraging the power of ensemble learning to improve accuracy and stability. The feature importance plot indicated that certain features, such as 'mean radius' and 'mean texture', were highly influential in the classification process.

Support Vector Machine (SVM):

- The SVM model achieved the highest scores across all metrics, confirming its suitability for this classification task. The use of the RBF kernel allowed it to capture complex relationships in the data, leading to superior performance.

Future Recommendations

To further enhance the model and its applicability, we propose the following recommendations:

1. **Continuous Model Training**:
    - Regularly update and retrain the model with new data to maintain its accuracy and relevance. Incorporating data from diverse populations can also improve its generalizability.
2. **Integration with Electronic Health Records (EHR)**:
    - Integrate the predictive model with EHR systems to automate the diagnostic process and streamline workflow for healthcare providers.
3. **Patient Follow-up and Monitoring**:
    - Use the model to identify patients who require more frequent monitoring and follow-up, ensuring timely detection of any changes in their condition.
4. **Ethical Considerations**:
    - Ensure that the model is used ethically, with considerations for patient privacy and informed consent. Transparent communication about the model's capabilities and limitations is crucial.

Summary of the Project Process

This project focused on developing predictive models to aid in the diagnosis of breast cancer using machine learning algorithms. The process began with an extensive literature review to identify the most relevant techniques and algorithms for our task. We then proceeded to collect and preprocess the dataset, ensuring it was clean and ready for analysis. Several machine learning models were implemented, including Decision Tree, Logistic Regression, Random Forest, and Support Vector Machine (SVM). Each model was trained and evaluated based on various performance metrics, and the SVM model emerged as the most effective in terms of accuracy, precision, recall, F1 score, and ROC AUC.

Group Work Experience

Working in groups on this project was both rewarding and challenging. Collaborating allowed us to pool our diverse skills and perspectives, which enriched the project and enhanced problem-solving. Effective communication and division of tasks were crucial in managing the workload and ensuring that each team member could contribute their expertise. Regular meetings and updates facilitated smooth progress and allowed us to address any issues promptly. Overall, the group dynamic fostered a collaborative learning environment that was instrumental in the project's success.

Working with Real-World Data

Handling real-world data introduced several practical challenges, such as dealing with missing values, outliers, and ensuring data integrity. However, it also provided invaluable insights into the complexities of real-life datasets. The experience underscored the importance of data preprocessing and the impact of data quality on

model performance. Working with actual data made the project more engaging and provided a realistic context for applying machine learning techniques.

Model Improvement with More Time

If we had more time, several enhancements could be made to our model:

1. **Hyperparameter Tuning**:
    - More extensive hyperparameter tuning using techniques like Grid Search or Random Search could optimize model performance further.
2. **Feature Engineering**:
    - Additional feature engineering could be performed to create new, more informative features from the existing data, potentially improving model accuracy.
3. **Advanced Algorithms**:
    - Exploring advanced algorithms such as Gradient Boosting Machines (GBM), XGBoost, or deep learning models could yield better results.
4. **Cross-Validation**:
    - Implementing more robust cross-validation techniques, such as k-fold cross-validation, would provide a more accurate estimate of model performance.
5. **Ensemble Methods**:
    - Combining multiple models through ensemble methods like stacking or blending could enhance predictive performance and robustness.