Machine Learning - 2nd Assignment

Task 3: Model Comparison (Logistic Regression, SVM, Random Forest)

Name: Aishwarya **Sub Code:** BCS602 **Year:** 2024-2025 **Semester:** VI

---

1. Detailed Explanation of Dataset Used

a. Dataset Name and Source

The dataset used for this task is the **Breast Cancer Wisconsin (Diagnostic) Dataset**. * **Source:** This dataset is included in `sklearn.datasets` module (`load_breast_cancer()`). It was chosen for consistency with Task 2 and its suitability for binary classification model comparison.

b. Size and Structure

- **Size:** The dataset contains **569 instances (samples)** and **30 numeric, predictive attributes (features)**.
- **Target Variable:** The target variable is binary: **malignant (0)** or **benign (1)**.
    - Malignant samples: [Count from notebook, e.g., 212]
    - Benign samples: [Count from notebook, e.g., 357]

c. Preprocessing of Dataset

The preprocessing steps were identical to those performed in Task 2 for consistency: 1. **Loading Data:** Dataset loaded via `sklearn.datasets.load_breast_cancer()`. 2. **Train-Test Split:** Data split into 70% training and 30% testing sets (`random_state=42`, `stratify=y`). 3. **Feature Scaling:** Features scaled using `StandardScaler` on both training and testing sets.

---

2. Explain the Working of Algorithms on Selected Dataset with Necessary Figures

Three classification models were trained and compared: Logistic Regression, Support Vector Machine (SVM), and Random Forest.

a. Logistic Regression

Logistic Regression is a linear model used for binary classification. It models the probability of a binary outcome using a logistic function (sigmoid function) applied to a linear combination of input features. * **Working:** It estimates coefficients for each feature, and the output is transformed by the sigmoid function to give a probability between 0 and 1. A threshold (typically 0.5) is used to classify the instance into one of the two classes. * **Parameters:** Default parameters were used from `sklearn.linear_model.LogisticRegression`, with `random_state=42` and `max_iter=10000` for convergence.

b. Support Vector Machine (SVM)

SVM aims to find the optimal hyperplane that best separates the classes in the feature space by maximizing the margin between the classes. * **Working:** (Briefly reiterate from Task 2, or state "As described in Task 2 report…"). For this comparison, an SVM with [mention kernel, C, gamma if you used specific ones from Task 2, e.g., "an RBF kernel with C=X, gamma=Y"] or default parameters (`kernel='rbf'`, `C=1.0`, `gamma='scale'`) was used, with `random_state=42`.

c. Random Forest Classifier

Random Forest is an ensemble learning method that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) of the individual trees. * **Working:** It builds multiple decision trees on various sub-samples of the dataset (using bootstrapping) and uses averaging (or voting) to improve predictive accuracy and control over-fitting. Each tree is trained on a random subset of features at each split. * **Parameters:** Default parameters were used from `sklearn.ensemble.RandomForestClassifier`, with `random_state=42`.

d. Evaluation Metrics

The models were evaluated based on the following metrics calculated on the test set: * **Accuracy:** The proportion of correctly classified instances. * **Precision:** The ability of the classifier not to label as positive a sample that is negative (TP / (TP + FP)). Calculated on a weighted average basis. * **Recall (Sensitivity):** The ability of the classifier to find all the positive samples (TP / (TP + FN)). Calculated on a weighted average basis. * **F1-Score:** The harmonic mean of precision and recall (2 * (Precision * Recall) / (Precision + Recall)). Calculated on a weighted average basis. * **Confusion Matrix:** A table showing the number of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN).

e. Results and Figures

Figure 1: Confusion Matrix - Logistic Regression *(Insert the CM plot for Logistic Regression. Save as `cm_logistic_regression.png` in Task3_Model_Comparison/report/)* `![CM Logistic Regression](./cm_logistic_regression.png)`

Figure 2: Confusion Matrix - SVM *(Insert the CM plot for SVM. Save as `cm_svm.png`)* `![CM SVM](./cm_svm.png)`

Figure 3: Confusion Matrix - Random Forest *(Insert the CM plot for Random Forest. Save as `cm_random_forest.png`)* `![CM Random Forest](./cm_random_forest.png)`

Figure 4: Comparison of Model Performance Metrics *(Insert the bar plot comparing Accuracy, Precision, Recall, F1 for all models. Save as `model_metrics_comparison.png`)* `![Model Metrics Comparison](./model_metrics_comparison.png)`

Summary Table of Performance Metrics:

Model	Accuracy	Precision (Weighted)	Recall (Weighted)	F1-Score (Weighted)
Logistic Regression	[Acc_LR]	[Prec_LR]	[Recall_LR]	[F1_LR]
SVM	[Acc_SVM]	[Prec_SVM]	[Recall_SVM]	[F1_SVM]
Random Forest	[Acc_RF]	[Prec_RF]	[Recall_RF]	[F1_RF]

(Fill this table with values from your notebook's `display_df` output)

---

3. Interpretation of Tasks and Results

The three models were successfully trained and evaluated on the Breast Cancer dataset. The performance metrics provide insights into their respective strengths and weaknesses for this particular problem.

Overall Performance (referencing Figure 4 and the Summary Table): * **Random Forest Classifier** demonstrated the highest overall performance with an accuracy of **[Acc_RF]%**, and also led in [mention other metrics like F1-score if it was the highest, e.g., F1-score of [F1_RF]%]. * **Support Vector Machine (SVM)** performed very competitively, achieving an accuracy of **[Acc_SVM]%** and an F1-score of **[F1_SVM]%**. * **Logistic Regression**, while being the simplest model, provided a strong baseline with an accuracy of **[Acc_LR]%** and an F1-score of **[F1_LR]%**.

Analysis of Confusion Matrices (referencing Figures 1, 2, 3): * **Logistic Regression:** * True Positives (Benign correctly classified): [Value] * True Negatives (Malignant correctly classified): [Value] * False Positives (Malignant -> Benign): [Value] * False Negatives (Benign -> Malignant): [Value] * **SVM:** * TP: [Value], TN: [Value], FP: [Value], FN: [Value] * **Random Forest:** * TP: [Value], TN: [Value], FP: [Value], FN: [Value]

Critical Error Analysis (False Negatives): In medical diagnosis for conditions like cancer, a False Negative (classifying a malignant tumor as benign) is often considered more critical than a False Positive. * Logistic Regression resulted in [FN_LR] False Negatives. * SVM resulted in [FN_SVM] False Negatives. * Random Forest resulted in [FN_RF] False Negatives. * [Comment on which model(s) performed best in minimizing these critical errors. E.g., “Both SVM and Random Forest showed a low number of False Negatives, with [Model X] having the absolute lowest at [Number].”]

Model Characteristics and Suitability: * **Logistic Regression:** It’s a good starting point due to its simplicity, interpretability, and fast training time. However, it may not capture complex non-linear relationships as effectively as the other models. * **SVM:** With appropriate kernel selection and hyperparameter tuning (as explored in Task 2), SVMs can achieve high accuracy and generalize well. They are effective in high-dimensional spaces. * **Random Forest:** This ensemble method is robust to overfitting (more so than individual decision trees), handles non-linear data well, and often requires less hyperparameter tuning to achieve good results compared to SVMs. It can also provide feature importance measures (not explored in this task but a known benefit).

Conclusion for Task 3: For the Breast Cancer Wisconsin dataset, all three models (Logistic Regression, SVM, and Random Forest) demonstrated strong predictive capabilities. **Random Forest and SVM generally outperformed Logistic Regression** in terms of overall accuracy and F1-score. Random Forest [or SVM, depending on your specific results] showed a slight edge, particularly in [mention specific metric or error type if applicable, e.g., minimizing false negatives or highest F1]. The choice of the “best” model in a real-world application would also consider factors beyond just these metrics, such as computational cost, model interpretability, and the specific costs associated with different types of misclassification.
