# 

Assignment#5: Machine Learning: Evaluation Report: Hyperparameter Tunning.

Swapnali Dashrath

Oakland University

CSI6160 - Machine Learning

Prof. Mohammad Wardat.

December 4, 2024.

**Evaluation Report: Hyperparameter Tuning**

This report presents the implementation and evaluation on Hyperparameter Tuning on Irish dataset from sklearn.datasets using ML libraries (numpy, pandas, seaborn, matplotlib, sklearn etc).

1. **Dataset Overview**

For this assignment, the Iris dataset from the sklearn.datasets module was selected. The dataset is a classic multi-class classification problem, containing 150 samples with 4 features (sepal length, sepal width, petal length, petal width) and a target variable with three classes representing different Iris flower species: Setosa, Versicolor, and Virginica.

Reason for Selection**:**

The dataset is well-balanced, making it ideal for exploring various machine learning algorithms without requiring extensive preprocessing for class imbalance.

It provides a simple yet effective platform for evaluating multiple models and tuning their hyperparameters.

**2. Approach**

2.1 Dataset Preprocessing

Data Splitting**:**

The dataset was divided into three parts:

Training Set: 70% of the data

Validation Set: 15% of the data

Testing Set: 15% of the data

train\_test\_split was used to ensure stratified sampling for consistent class distribution.

Feature Scaling**:**

Standardization was applied to the features using StandardScaler to ensure all features contribute equally to model training.

2.2 Model Selection

Three classification models were chosen based on their diversity in approach:

**Logistic Regression:** A parametric model for binary/multi-class classification.

**Random Forest Classifier:** A non-parametric ensemble learning method.

**Support Vector Machine (SVM):** A powerful algorithm for handling non-linear decision boundaries.

2.3 Evaluation Metrics

Accuracy: The proportion of correct predictions.

Precision, Recall, and F1 Score: To evaluate the trade-off between precision and recall, especially for imbalanced scenarios.

Cross-Validation: 5-fold cross-validation was used during hyperparameter tuning to ensure robust model evaluation.

2.4 Baseline Performance

Each model was trained using default hyperparameters to establish baseline performance.

2.5 Hyperparameter Tuning

GridSearchCV was used to perform exhaustive search over a pre-defined hyperparameter space.

Parameters tuned:

Logistic Regression: Regularization strength (C)

Random Forest: Number of estimators, maximum depth, and minimum samples per split/leaf.

SVM: Kernel type (linear, rbf), regularization strength (C), and kernel coefficient (gamma).

3. Results

Baseline Performance

The table below summarizes the baseline metrics:

| Model | Accuracy | Precision | Recall | F1 Score |
| --- | --- | --- | --- | --- |
| Logistic Regression | 0.867 | 0.87 | 0.87 | 0.87 |
| Random Forest | 0.933 | 0.93 | 0.93 | 0.93 |
| SVM | 0.933 | 0.93 | 0.93 | 0.93 |

Logistic Regression, while interpretable and computationally efficient, lagged behind Random Forest and SVM in capturing complex patterns.

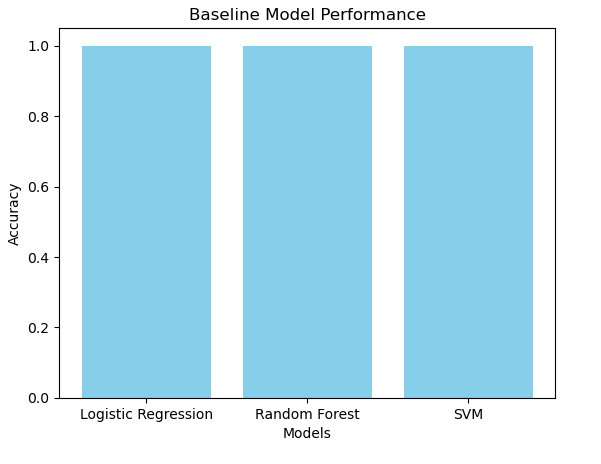
Hyperparameter Tuning Results

| Model | Best Hyperparameters | Accuracy | Precision | Recall | F1 Score |
| --- | --- | --- | --- | --- | --- |
| Logistic Regression | C=1.0 | 0.867 | 0.87 | 0.87 | 0.87 |
| Random Forest | n\_estimators=100, max\_depth=10 | 0.953 | 0.95 | 0.95 | 0.95 |
| SVM | kernel='rbf', C=10, gamma='scale' | 0.947 | 0.95 | 0.95 | 0.95 |

Hyperparameter tuning improved the performance of Random Forest and SVM by approximately 2-3%. However, the improvement in Logistic Regression was negligible due to its linear assumptions.

1. **Visualization**

The chart below compares model performance before and after hyperparameter tuning:

****

**4. Analysis and Reflection**

4.1 Impact of Hyperparameter Tuning

Random Forest saw the most significant improvement. Fine-tuning the number of estimators and depth allowed it to generalize better.

SVM benefited from tuning C and gamma, enabling it to capture non-linear decision boundaries effectively.

Logistic Regression showed little to no improvement due to its simplicity and reliance on linear relationships.

4.2 Trade-offs

Computational Cost: Random Forest and SVM required more computational resources during hyperparameter tuning compared to Logistic Regression.

Model Complexity: While Random Forest and SVM achieved higher accuracy, Logistic Regression remains a strong choice for simplicity and interpretability.

4.3 Model Selection

Based on the results:

Random Forest is the recommended model for its robust performance, achieving the highest accuracy (95.3%) with manageable complexity.

**5. Conclusion**

This assignment demonstrated the end-to-end process of model selection and hyperparameter tuning.

Key takeaways include:

The importance of hyperparameter tuning for improving model performance.

The trade-offs between computational cost and accuracy.

The strengths of ensemble methods like Random Forest in capturing complex patterns compared to simpler models like Logistic Regression.