RP SEISO

32577443

CISM 623 PRACTICAL LAB 2

Brief description.

1. EDA Insights
    - Different Feature scales required StandardScaler application
      EDA showed different features had different ranges (sepal length: 4,3 – 7.9 cm, petal width: 0.1 – 2.5 cm). This guided the decision to standardize all features for the distance-based algorithms kNN and gradient-based models Logistic Regression.
    - String Feature corelation: action; monitored for multicollinearity
      High correlation of 0.96 between petal length and width suggested potential multicollinearity, and from this it was expected for the tree-based model: Random Forest, to perform well as they handle correlated features better than linear models.
    - Clear Class separability pattern helped to make an informed model selection.
      Setosa was perfectly separable, while versicolor/virginica had overlap, thus suggested that both linear and non-linear models could work.
    - Perfect Class Balance: Used accuracy as primary metric
      With equal class distribution of 50 samples each, accuracy was a reliable metric without needing weighted averages.
2. Best Performing Model
   All model performed well, with accuracy of 0.933 %
   Though Random Forest model was expected to perform this well because:
    - It handles non-linear boundaries, the overlap between versicolor and virginica classes required complex decision boundaries that Random Forest's ensemble approach captures effectively
    - It robust against correlated features, with petal length and width highly correlated (0.96), Random Forest's feature bagging prevents over-reliance on any single feature
3. Class Imbalance Impact: Hypothetical Scenario

   Since my dataset was perfectly balanced, imbalance didn't affect metrics. However, if there had been class imbalance:

    - Accuracy would be misleading, as a model could achieve high accuracy by simply predicting the majority class
    - Precision/Recall trade-off: the minority class would likely have lower recall (more false negatives)
    - F1-score would be crucial: would provide a better balance than accuracy for imbalanced scenarios
    - Need for stratified sampling, going to be essential to maintain class proportions in train/test splits
    - Class weighting, was going be necessary for algorithms like Logistic Regression and SVM

4. To improve the model performance, I suggest in future to use Hyperparameter Tuning as it would address the one misclassification between versicolor/virginica, optimize each algorithm for this dataset, and could push Logistic Regression and kNN up to 100% accuracy.
   Snapshot for suggested improvement

```
from sklearn.model_selection import GridSearchCV

# Logistic Regression
param_grid_lr = {
    'C': [0.001, 0.01, 0.1, 1, 10, 100],
    'penalty': ['l1', 'l2', 'elasticnet'],
    'solver': ['liblinear', 'saga']
}

# Random Forest
param_grid_rf = {
    'n_estimators': [50, 100, 200],
    'max_depth': [None, 5, 10, 20],
    'min_samples_split': [2, 5, 10]
}

# For kNN
param_grid_knn = {
    'n_neighbors': [3, 5, 7, 9, 11],
    'weights': ['uniform', 'distance'],
    'metric': ['euclidean', 'manhattan']
}
```