Machine Learning Project in Python — Iris-Step-by-Step

a)Purpose of the project

The main objective of this project involves performing a complete machine learning workflow which includes dataset loading followed by exploration and proper train/test split creation and multiple algorithm comparison through cross-validation and model tuning for the best candidate and final evaluation on unseen test data and model saving. The implementation follows the “Step-by-Step” for this project. The dataset contains 150 observations which include four numeric attributes for sepal length performance predictions on new data.

b) Description of the dataset

The Iris flower dataset serves as the “Step-By-Step” lesson structure to achieve a tuned Support Vector Machine (SVM) that and width and petal length and width and three classes that represent Iris-setosa and Iris-versicolor and Iris-virginica with equal distribution of 50 samples each. The CSV file originated from the public URL used in the lesson and I preserved the original column names for better understanding. The program saved a copy of the dataset to disk storage to achieve run reproducibility without needing internet access. The dataset maintains class balance and small size for visualization purposes and demonstrates supervised classification effectively.

c) Statistical methods used

The training data received 10-fold stratified cross-validation to determine how well each algorithm would perform in real-world applications. The stratification method maintains the same class distribution in each fold because it matters for multi-class problems. The evaluation of each candidate model included recording both the average accuracy and standard deviation from 10 repeated folds. A grid search followed the initial spot-check to optimize SVM parameters (kernel, C, gamma) through the same cross-validation process.

The selection of the final model occurred through evaluation of training-fold results only. The evaluation included test set accuracy assessment together with confusion matrix visualization and classification report output that displayed precision and recall and F1-score values for each class. The evaluation process separates model tuning from testing to prevent optimistic bias.

d) Visualization methods used

The exploratory data analysis **(EDA)** required me to develop three fast visualization tools.

1. **Boxplots** for each feature to check spread and potential outliers.
2. **Histograms** to see distribution shape and skew.
3. A **scatter-matrix** to visualize pairwise relationships between features and to see the class separation trend—particularly how petal measurements often separate the classes.

The evaluation process included three visualization elements which consisted of a boxplot displaying cross-validated accuracy results for different algorithms and a bar chart showing the best CV mean accuracy compared to test accuracy for the selected model and a confusion-matrix heatmap for test predictions. The figures were stored in a figures/ directory to include them in my report.

e) Models evaluated, chosen model, and rationale

The evaluation included six standard classification algorithms which were Logistic Regression (LR), Linear Discriminant Analysis (LDA), k-Nearest Neighbors (KNN), Decision Tree (CART), Gaussian Naive Bayes (NB) and Support Vector Machine (SVM). The models LR/KNN/SVM received StandardScaler treatment as a pipeline component to ensure proper scaling during cross-validation operations. The cross-validation results showed high accuracy values ranging from 0.93 to 0.97 with NB performing slightly better than others in the initial spot-check mean evaluation. The best SVM model with linear kernel and C value of 0.1 achieved a CV accuracy of 0.973 which provided both high performance and a simple linear decision boundary and stable results on these specific features. The final model selection went to SVM because it achieved the highest cross-validated accuracy and produced interpretable linear decision boundaries in the petal/sepal feature space.

f) Prediction result

The test accuracy of the tuned SVM model reached approximately 0.92 when applied to the hold-out test data. The confusion matrix revealed perfect Setosa classification but showed minimal errors between Versicolor and Virginica samples because these two species share overlapping features in their space. The classification report showed that macro precision and recall and F1-score values reached approximately 0.92 in total. The trained model received joblib saving and reloading which resulted in identical test accuracy from the reloaded model to verify proper model persistence.

g) Business use discussion

Although Iris is a teaching dataset, the workflow generalizes directly to business problems. A nearly identical pipeline could:

- **Quality inspection:** classify products as pass/fail or into defect categories from measured dimensions (analogous to petal/sepal features).
- **Customer segmentation and targeting:** assign customers to segments via behavioral features, then personalize offers.
- **Service triage and routing:** route tickets or emails to the right team using text-derived features and a trained classifier.

Conclusion.

After evaluating six classifiers with 10-fold stratified cross-validation and tuning hyperparameters, a linear SVM was selected and achieved ~0.92 test accuracy. The confusion matrix shows perfect Setosa identification and minor Versicolor/Virginica confusions. The end-to-end process— load→ EDA → CV → tuning → clean test evaluation— makes the result robust and reproducible.

References

Jason Brownlee. (2023, September 26). Your First Machine Learning Project in Python Step-By-Step. https://machinelearningmastery.com/machine-learning-in-python-step-by-step/

Jason Brownlee. (2021, October 13). Python Machine Learning Mini-Course. https://machinelearningmastery.com/python-machine-learning-mini-course/

Molin, S. (2021). *Hands-On Data Analysis with Pandas: A Python data science handbook for data collection, wrangling, analysis, and visualization*. Packt Publishing Ltd.

Retrieved from https://raw.githubusercontent.com/jbrownlee/Datasets/master/iris.csv