Elements of AIML

LAB – 5



Name - Ronak Singh

SAP ID - 500120683

Batch – 12

Roll No. – R2142230381

**Topic: K-fold cross validation**

**Question**

**How does K-Fold Cross-Validation influence the accuracy of various machine learning classification algorithms (Logistic Regression, Decision Tree, Support Vector Machine, K-Nearest Neighbors, and Linear Discriminant Analysis) when applied to the Pima Indians Diabetes Dataset, Wine Quality Dataset, and Breast Cancer Wisconsin Dataset?**

**Introduction**

In the field of machine learning, evaluating the performance of models is crucial for ensuring their reliability and accuracy. One widely used technique for this purpose is K-Fold Cross-Validation, which helps mitigate the risk of overfitting by dividing the dataset into K subsets, or "folds." This method enables models to be trained on different portions of the data while being tested on unseen data, providing a robust estimate of their performance.

This project involves building a classification model to predict the quality of red wines based on various chemical properties. The workflow includes data preprocessing, encoding, splitting the dataset, and evaluating multiple classification algorithms..

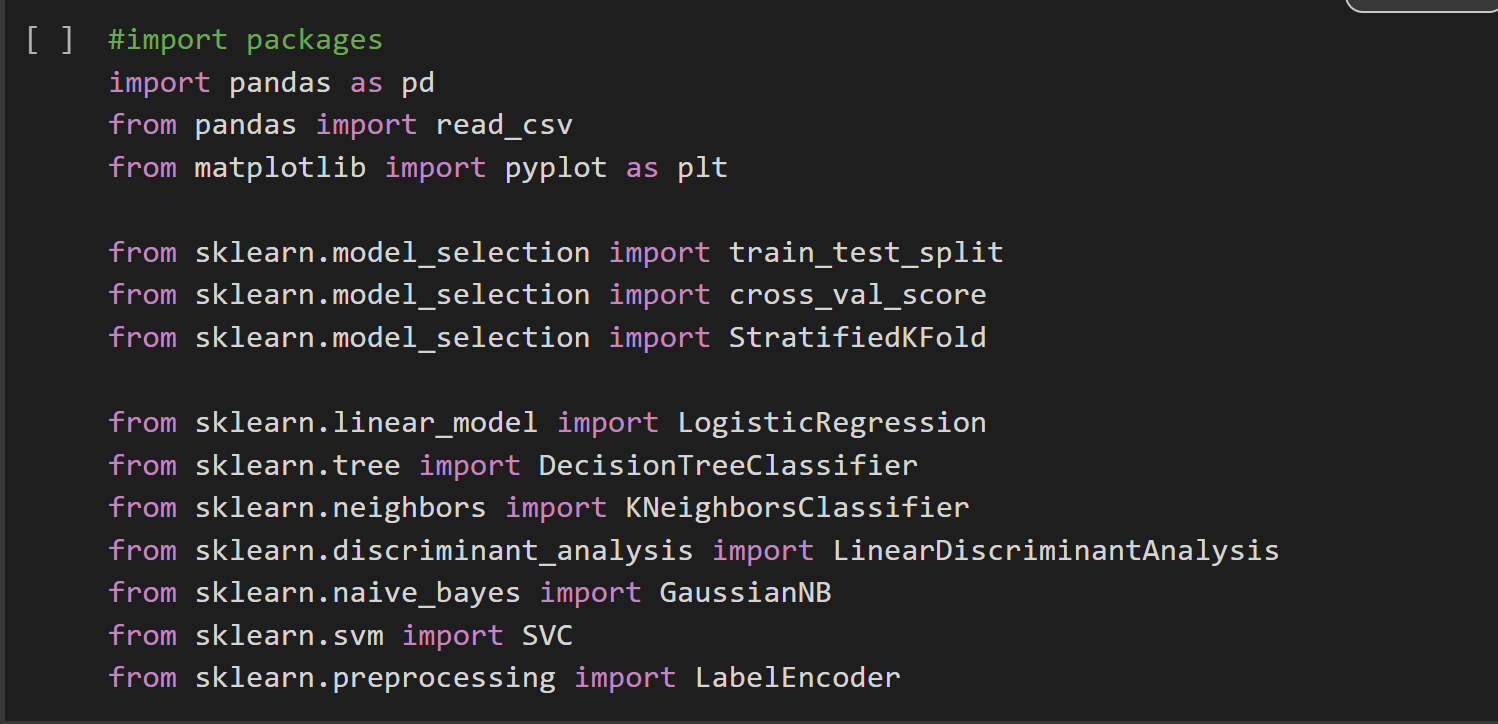
In this report, we will apply K-Fold Cross-Validation to the Wine Quality Dataset(from UCI repo). We will evaluate five classification algorithms—Logistic Regression, Decision Tree, Support Vector Machine (SVM), K-Nearest Neighbours (KNN), and Linear Discriminant Analysis (LDA)—to determine which method yields the highest accuracy in predicting outcomes. By analysing this datasets, we aim to gain insights into the effectiveness of different classification techniques and their applicability in real-world scenarios.

Wine Quality Dataset

**STEPS OF THE CODE:**

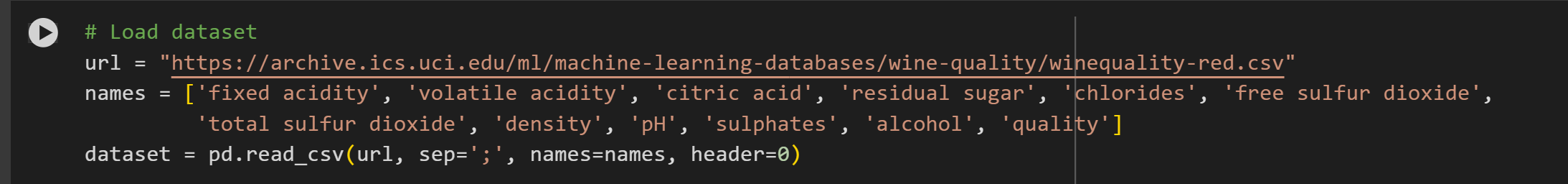
**1. Import Packages**

This section imports the necessary libraries required for the experiment. **Pandas** is used for data manipulation, **Matplotlib** for data visualization, and **Sklearn** for applying machine learning algorithms. These libraries provide the essential tools for data preprocessing, model evaluation, and visualization.



**2. Loading the Dataset**

The first step in the project is to import the **Wine Quality Dataset** from the UCI Machine Learning Repository. The dataset consists of several chemical attributes (e.g., fixed acidity, pH, sulphates) along with a quality column that indicates the wine’s quality rating.



**3.Quality Binning**

The wine quality ratings in the original dataset range from 0 to 10, making it a continuous value. To simplify the classification problem, the quality ratings are binned into three distinct categories: low, medium, and high. This is done using defined bins:

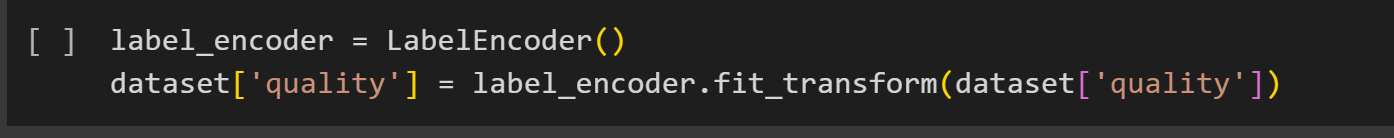
* low (0-5),
* medium (6), and
* high (7-10).

This approach converts the quality rating into a discrete classification problem, which is easier to handle using standard machine learning algorithms.



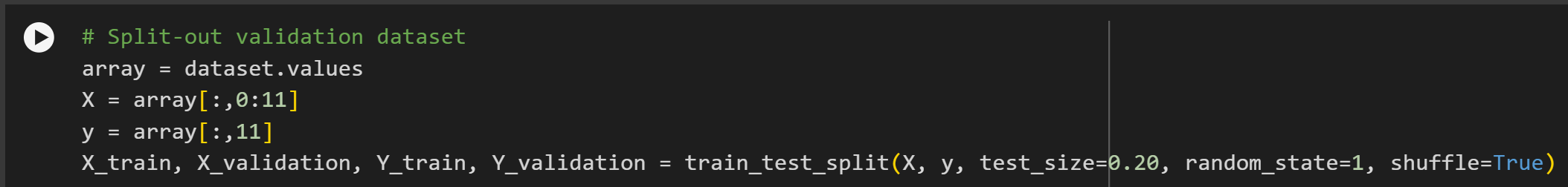
**4. Label Encoding**

After binning the quality values into three categories, a **label encoder** is applied. Label encoding converts the string labels (low, medium, high) into numerical labels (0, 1, 2). This transformation is essential as many machine learning models work only with numerical inputs.



**5. Data Splitting**

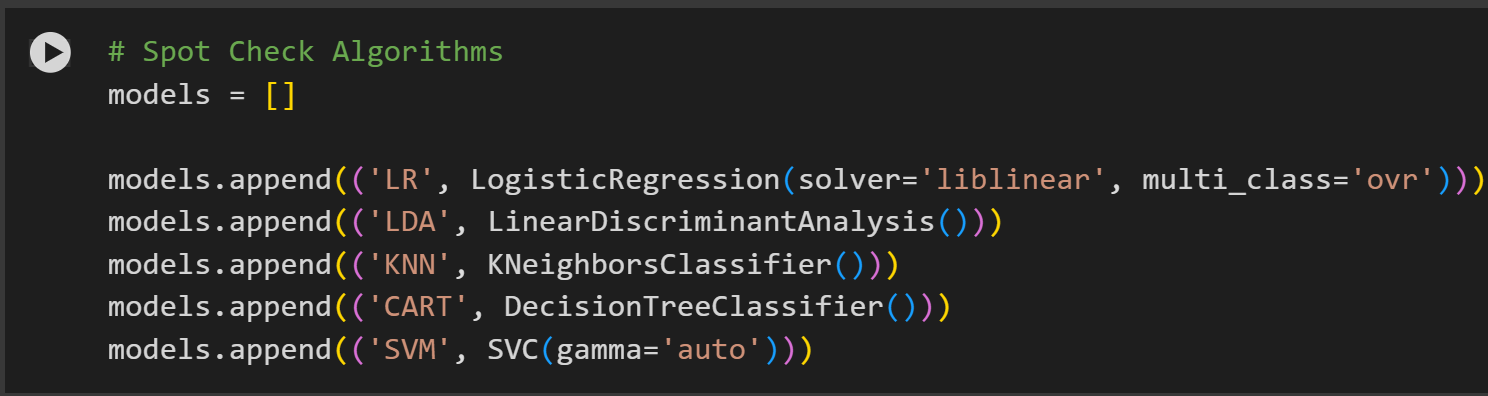
To evaluate the performance of the classification models, the dataset is split into training and validation sets. The independent variables (X) and the target variable (y) are separated, and 20% of the data is reserved for validation purposes.



**6. Model Definition**

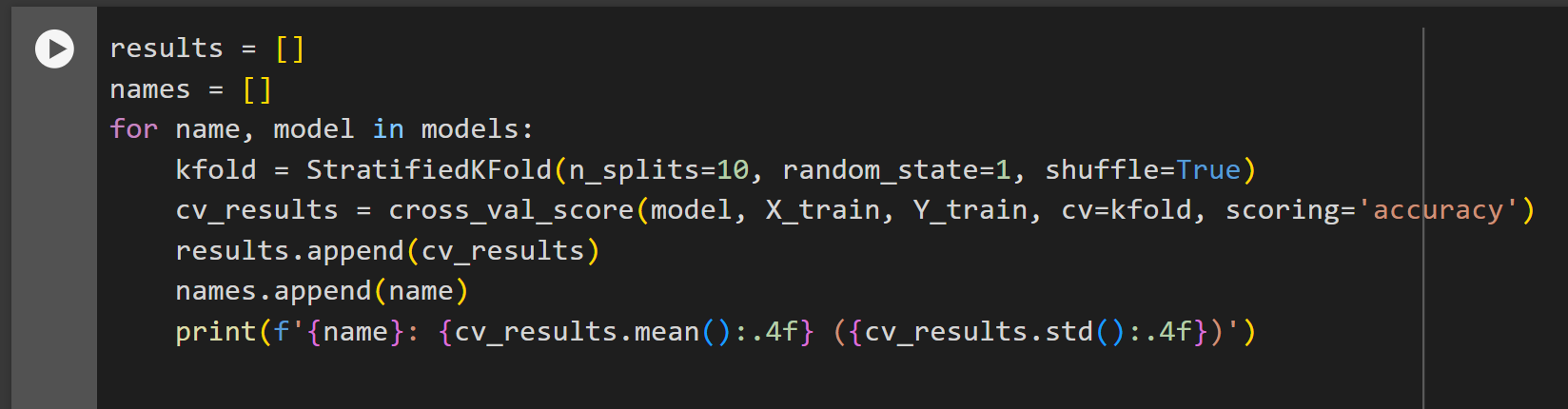
A variety of classification models are chosen to be evaluated on the dataset. The are Logistic Regression (LR), Linear Discriminant Analysis (LDA), K-Nearest Neighbours (KNN), Decision Tree Classifier (CART) and Support Vector Machine (SVM).

Each model is appended to a list, and their performance is tested using the validation dataset. The purpose of spot checking is to quickly compare different algorithms to identify which model performs the best for this classification task.



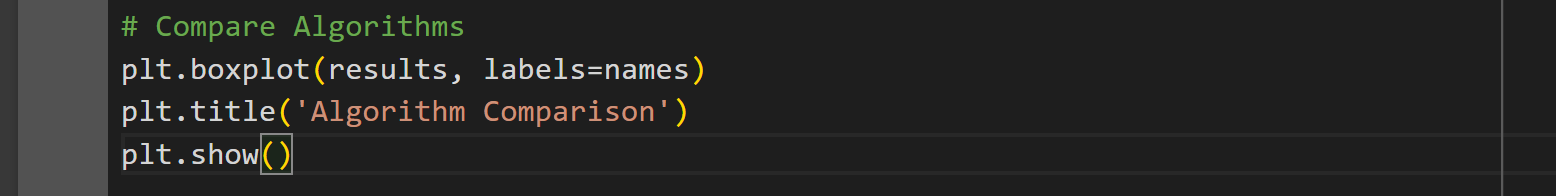
**7. Cross-Validation**

Each machine learning model is evaluated using **Stratified K-Fold Cross-Validation** to ensure robust performance testing. This technique divides the training data into 10 folds, allowing each model to be trained and tested on different parts of the dataset. The accuracy scores for each model are calculated and recorded.

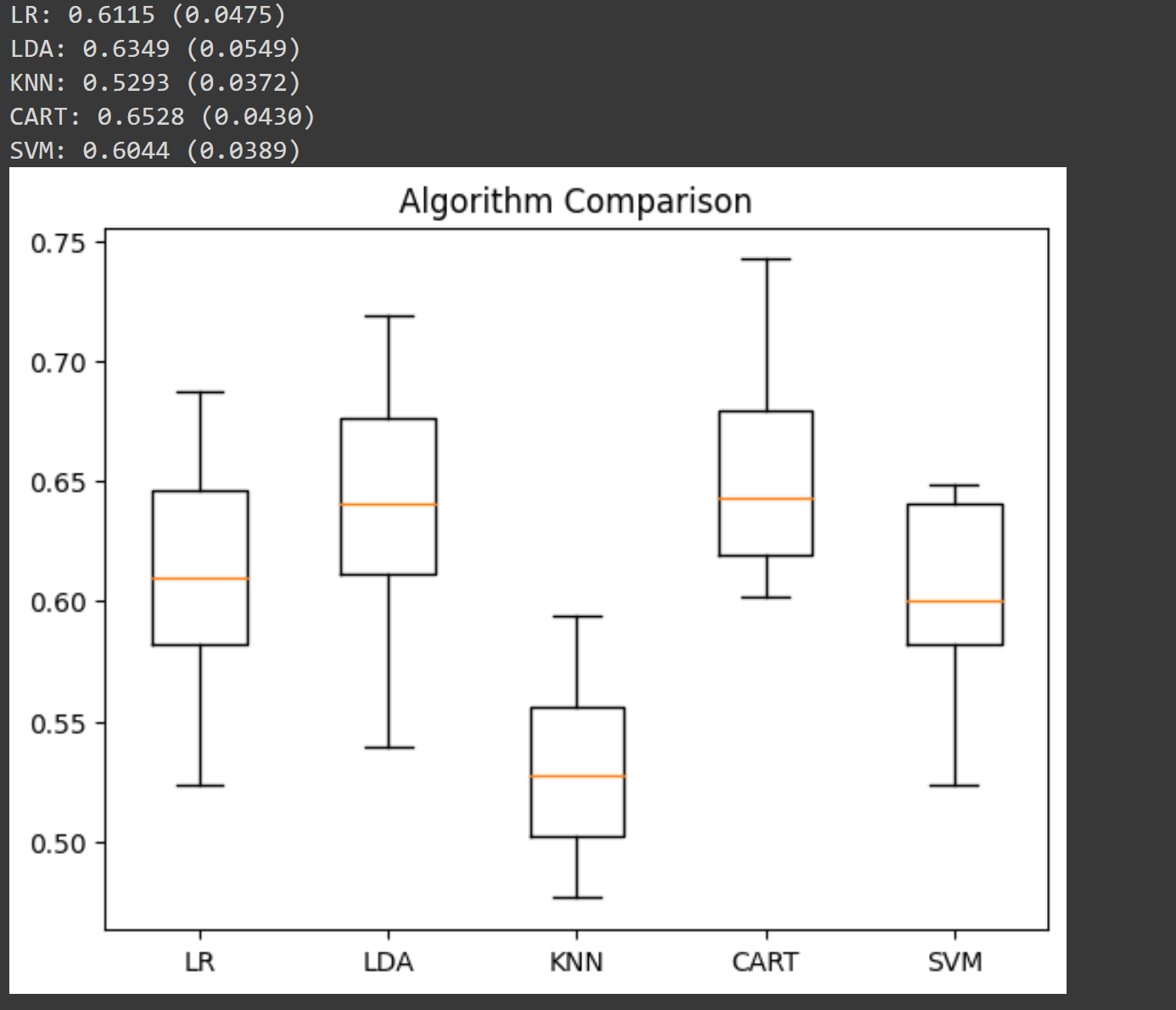


**8. Visualization**

To visualize the performance of each model, the accuracy scores from cross-validation are plotted using a **boxplot**. This allows for a quick comparison of the models’ accuracy distribution, highlighting the best-performing models and the variance in their results.



**9. Result:**



Final Conclusion:

After analysing the accuracy scores and visualizing the performance of five machine learning models—Logistic Regression (LR), Linear Discriminant Analysis (LDA), K-Nearest Neighbors (KNN), Classification and Regression Trees (CART), and Support Vector Machines (SVM)—the following conclusions can be drawn:

1. **Best Performing Model**:

**CART** (Classification and Regression Trees) showed the highest mean accuracy (0.6528) among the models with a relatively low standard deviation (0.0430), indicating better stability and overall performance in predicting wine quality.

1. **Moderate Performing Models**:

**LDA** and **SVM** achieved comparable mean accuracies of 0.6349 and 0.6044, respectively. Both models show promising results but fall slightly behind CART.

1. **Lowest Performing Model**:

**KNN** had the lowest mean accuracy (0.5293) with higher variance, indicating poor consistency in its performance. This suggests that KNN is not suitable for this particular problem without further tuning.

1. **Model Robustness**:

The boxplot visualization shows that the accuracy distribution for CART is more stable compared to others, making it a good choice for generalization.

Recommendation:

CART is recommended as the primary model for wine quality prediction based on its higher accuracy and robustness. Further optimization, such as hyperparameter tuning, could be applied to potentially improve results.

If computational resources or other constraints exist, LDA or SVM could also be considered as secondary options.