Model Evaluation Using Cross-Validation and Random Forest on Iris Dataset

Submitted by - RATIK KRISHNA M P - 727723EUCS184 - III CSE C

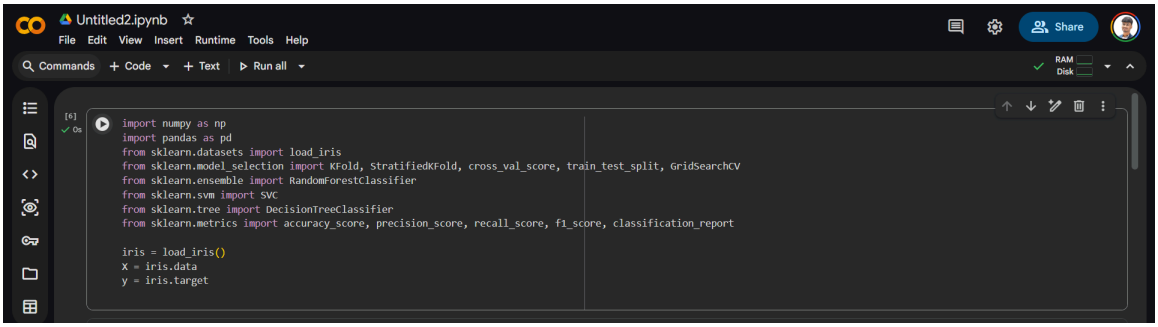
Abstract

This project focuses on evaluating machine learning models using cross-validation techniques and improving performance using a Random Forest classifier. The Iris dataset is used as a benchmark classification dataset. K-Fold and Stratified K-Fold cross-validation methods are applied to ensure reliable model evaluation. Random Forest hyperparameters are tuned using GridSearchCV to achieve optimal performance. The results are compared with Support Vector Machine (SVM) and Decision Tree classifiers using metrics such as accuracy, precision, recall, and F1-score. The project demonstrates the importance of model generalization and proper evaluation techniques.

Introduction

Machine learning models must not only perform well on training data but also generalize effectively to unseen data. A simple train-test split may give misleading results because performance can vary depending on how the data is split.

To solve this problem, cross-validation is used. Cross-validation divides the dataset into multiple parts and evaluates the model several times, ensuring more reliable performance estimation.



In this project:
- We use the Iris dataset
- Apply K-Fold and Stratified K-Fold Cross-Validation
- Train and tune a Random Forest model
- Compare it with SVM and Decision Tree

Feature	Random Forest	Support Vector Machine (SVM)	Decision Tree
Type of Model	Ensemble (Multiple Trees)	Margin-Based Classifier	Single Tree Model
Working Principle	Combines predictions from many decision trees	Finds optimal hyperplane separating classes	Splits data based on feature thresholds
Accuracy	High	High	Moderate
Overfitting Risk	Low	Low to Moderate	High
Generalization Ability	Strong	Good	Weak
Feature Scaling Required	No	Yes	No
Training Speed	Medium	Slow (for large datasets)	Fast
Prediction Speed	Medium	Fast	Fast
Best Use Case	Complex datasets with high accuracy needs	Problems with clear margin separation	Simple or small datasets
Number of Hyperparameters	Many (n_estimators, max_depth, etc.)	Moderate (C, kernel, gamma)	Few (max_depth, min_samples_split)
Robust to Noise	Yes	Moderate	No

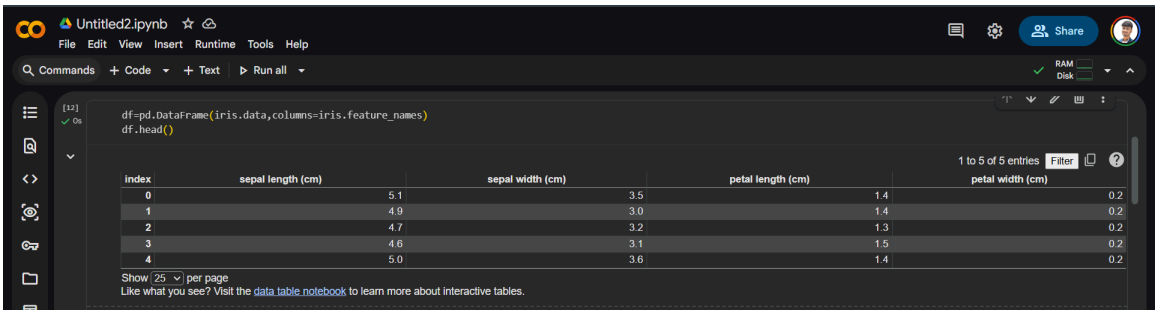
Dataset Description

The Iris dataset is a well-known dataset used for classification problems.

Number of Samples: 150
Number of Features: 4
Number of Classes: 3
Classes: Setosa, Versicolor, Virginica

Features:
- Sepal Length: Length of sepal (cm)
- Sepal Width: Width of sepal (cm)
- Petal Length: Length of petal (cm)
- Petal Width: Width of petal (cm)

The dataset is balanced, meaning each class has 50 samples.



Data Preprocessing

The Iris dataset does not contain missing values, so no data cleaning was required.

However, feature scaling (normalization) is applied because models like SVM are sensitive to feature magnitude.

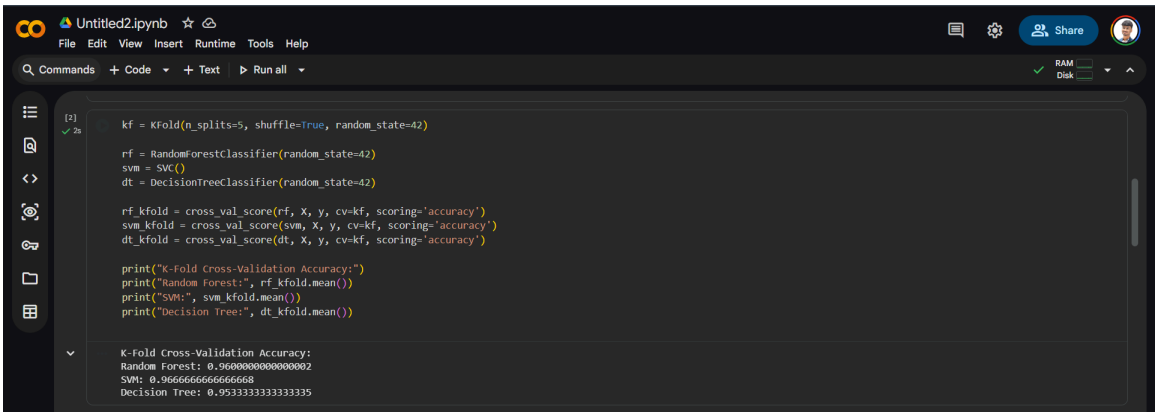
We use Standardization, which transforms data to have:
- Mean = 0
- Standard Deviation = 1

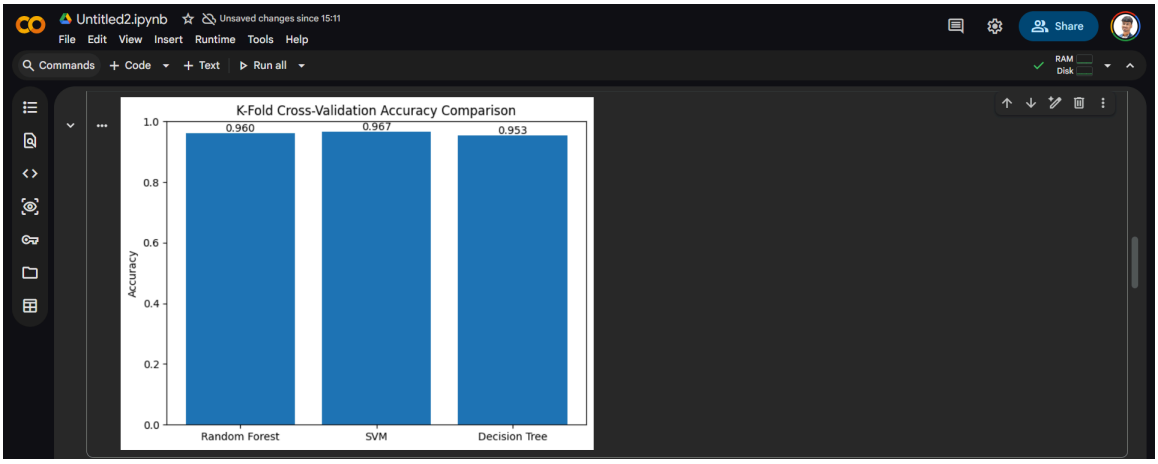
This ensures that all features contribute equally.

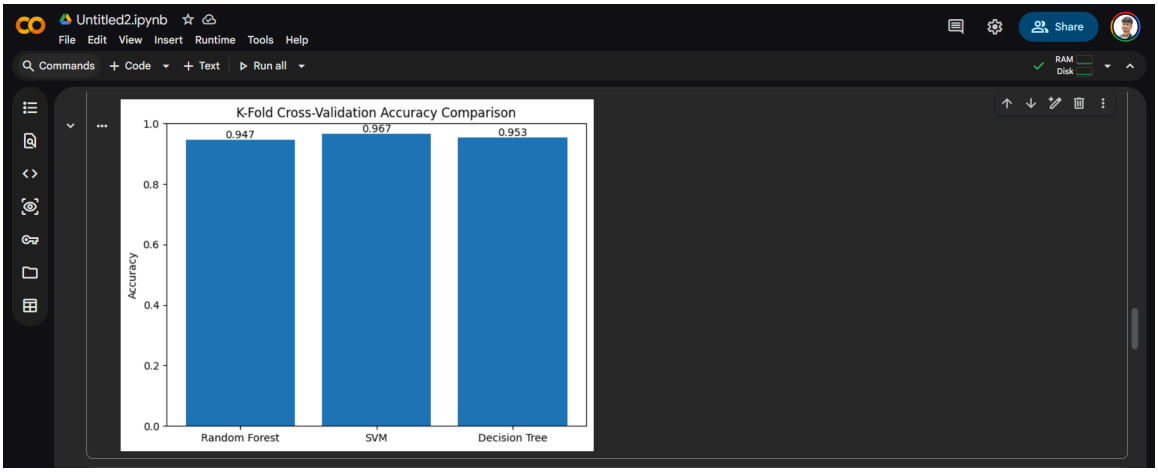
Cross-Validation

K-Fold cross-validation divides the dataset into K equal parts. The model is trained on K-1 parts and tested on the remaining part. This process repeats K times.

Stratified K-Fold ensures that each fold has the same class distribution as the original dataset. This is especially important in classification problems.







Models Used

Decision Tree: A tree-based model that splits data based on feature values. It is simple but prone to overfitting.

Support Vector Machine (SVM): A powerful classifier that finds the best boundary between classes. It requires normalized data.

Random Forest: An ensemble model that combines multiple decision trees. It reduces overfitting and improves accuracy by averaging predictions.

Hyperparameter Tuning (Random Forest)

Random Forest has parameters that control model complexity:

- n_estimators: Number of trees
- max_depth: Maximum depth of trees
- min_samples_split: Minimum samples to split node
- min_samples_leaf: Minimum samples at leaf

We use GridSearchCV to try different combinations and select the best one using cross-validation.

Model Evaluation Metrics

To evaluate the final model, we use:

- Accuracy: Overall correctness
- Precision: Correct positive predictions
- Recall: Ability to find all true positives
- F1-score: Balance of precision & recall

Classification Report:

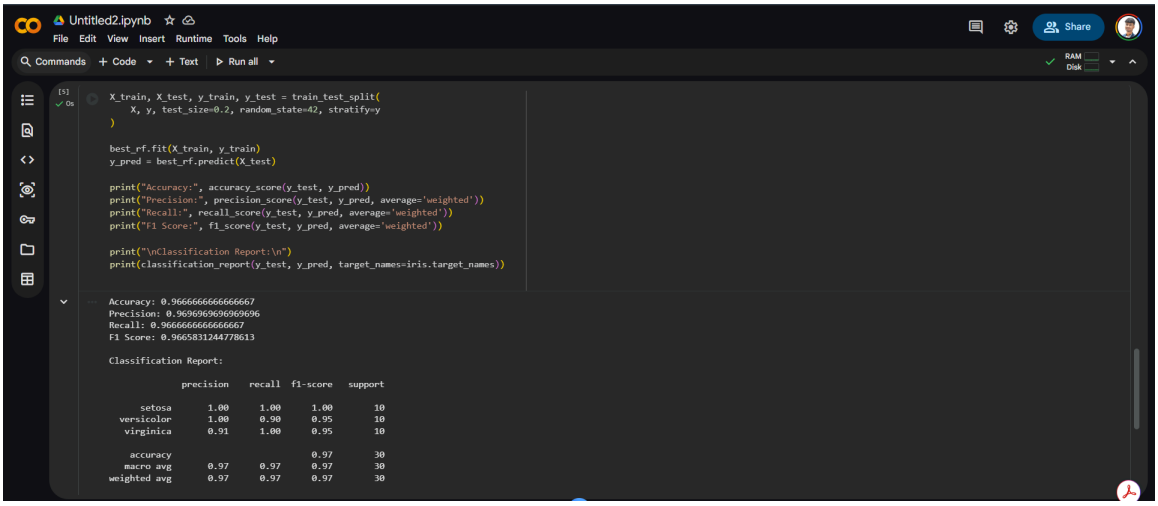
	precision	recall	f1-score	support
setosa	1.00	1.00	1.00	10
versicolor	1.00	0.90	0.95	10
virginica	0.91	1.00	0.95	10
accuracy			0.97	30
macro avg	0.97	0.97	0.97	30
weighted avg	0.97	0.97	0.97	30

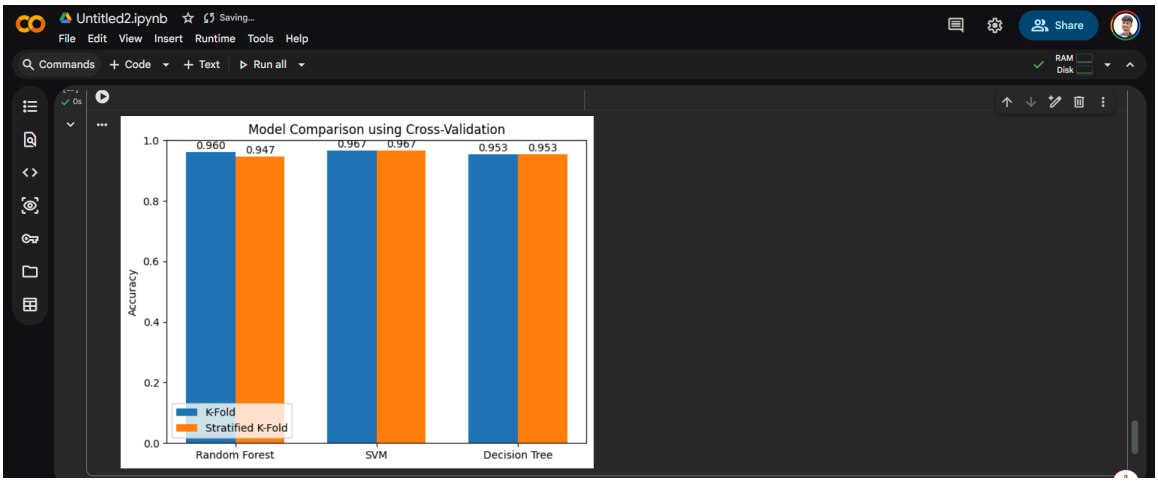
Results

Cross-Validation Accuracy Comparison:

Random Forest – High
SVM – High
Decision Tree – Moderate

Random Forest achieved the best performance due to ensemble learning.





Generalization

Generalization refers to a model's ability to perform well on unseen data.

Because the cross-validation accuracy and test accuracy are similar, the model shows good generalization and is not overfitting.

Random Forest improves generalization by combining multiple trees and reducing variance.

- **Random Forest with K-Fold:** 0.96000000000002
- **Random Forest with Stratified K-Fold:** 0.9466666666666667
- **Random Forest with Hyper-Parameter Tuning:** 0.9666666666666668

Conclusion

This project demonstrated how cross-validation improves model evaluation reliability. Stratified K-Fold ensured balanced class distribution across folds. Random Forest outperformed Decision Tree and SVM due to ensemble learning. Hyperparameter tuning further improved performance. The final model showed strong generalization and high accuracy, making Random Forest a suitable model for classification tasks.

References

- Scikit-learn Documentation
- UCI Machine Learning Repository – Iris Dataset