A project report on

Iris Flower Classification

Submitted in partial fulfilment of the requirements for

the award of the Degree of

B.Sc Data Science

By

Amit Singh (Roll no.28954321023)

Under the supervision of :

Mr. Sujit Chakraborty

Assistant Professor (Department Of Computer Science and Technology)

DEPARTMENT OF COMPUTER SCIENCE AND TECHNOLOGY

(DATA SCIENCE)

Institute Of Leadership, Entrepreneurship and Development

(113 J Maheshwartola Road, Topsia ,Kolkata, West Bengal 700046)

Affiliated to

Maulana Abul Kalam Azad University Of Technology

(MAKAUT)

February, 2024

iLEAD®
INSTITUTE OF LEADERSHIP
ENTREPRENEURSHIP
& DEVELOPMENT

BONAFIDE CERTIFICATE

This is to certify that the project titled IRIS FLOWER CLASSIFICATION is a bonafide record of the work done by

Amit Singh (28954321023)

in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Technology

(Data Science) of the iLead, Kolkata, during the year 2023-24.

PROJECT GUIDE

Guide

DEPARTMENT HOD

Head of the Department

Internal Examiner

External Examiner

ABSTRACT

This project presents a comprehensive analysis and comparison of machine learning models for the classification of Iris species using the well-known Iris dataset. The dataset includes measurements of sepal length, sepal width, petal length, and petal width for three species of Iris flowers: Iris-setosa, Iris-versicolor, and Iris-virginica.

The analysis begins with an initial exploration of the dataset to understand its structure and identify any potential issues such as missing values or outliers. Outliers in the sepal width feature were detected and removed to ensure a cleaner dataset. Various visualizations, including boxplots and pair-plots, were used to explore the relationships between features and the distribution of species.

Following data preprocessing, the dataset was encoded and split into training and testing sets. Three machine learning models were selected for this classification task: Decision Tree Classifier, K-Nearest Neighbors (KNN) Classifier, and Support Vector Classifier (SVC). Each model was trained on the training set and evaluated on the testing set, with accuracy scores being the primary metric for comparison.

The evaluation revealed that each model performed differently, highlighting the strengths and weaknesses of each approach in the context of this specific classification problem. The results were compiled into a comprehensive report, including a detailed comparison of model performances and predictions.

The project concludes with a summary of findings, recommendations based on the model performances, and suggestions for future work to further enhance the classification accuracy and robustness. This study demonstrates the effective application of machine learning techniques to a classic dataset and provides insights into model selection and evaluation processes.

ACKNOWLEDGEMENTS

We would like to express our deepest gratitude to the following people for guiding us through this course and without whom this project and the results achieved from it would not have reached completion.

Sujit Chakraborty, Assistant Professor, Department of Computer Science and Technology (Data Science & Cyber Security), for helping us and guiding us in the course of this project. Without his/her guidance, we would not have been able to successfully complete this project. His/Her patience and genial attitude is and always will be a source of inspiration to us.

Manas Paul, the Head of the Department, Department of Computer Science and Technology (Data Science & Cyber Security), for allowing us to avail the facilities at the department.

We are also thankful to the faculty and staff members of the Department of Computer Science and Technology (Data Science & Cyber Security), our individual parents and our friends for their constant support and help.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Introduction

In the realm of data science, the Iris dataset stands as a quintessential example for exploring and understanding the fundamentals of machine learning. This project focuses on the classification of Iris species using their sepal and petal dimensions, employing various machine learning techniques. By leveraging the distinctive features of the Iris dataset, we aim to build, evaluate, and compare different classification models to determine the most effective method for accurately identifying the species of an Iris flower.

Objective:

The primary objective of this project is to develop a robust classification system that can accurately predict the species of Iris flowers based on measurements of their sepals and petals. We will achieve this by implementing three widely used machine learning algorithms: Decision Tree Classifier, K-Nearest Neighbors Classifier (KNN), and Support Vector Classifier (SVC). Each model will be trained and tested on the Iris dataset, and their performances will be evaluated and compared based on accuracy and other relevant metrics.

Through this project, we aim to gain insights into the strengths and weaknesses of each classification technique, understand the underlying patterns within the dataset, and provide a comprehensive analysis of the results. This will not only enhance our understanding of these machine learning algorithms but also demonstrate their practical applications in solving real-world classification problems.

Data Description

Dataset Overview

The dataset used for this project is the Iris dataset, a well-known dataset in the field of machine learning and statistics. This dataset contains 150 samples of Iris flowers, each described by four features, and classified into three species. The features include the lengths and widths of the sepals and petals, which are measured in centimeters. The three species of Iris flowers in the dataset are Iris-setosa, Iris-versicolor, and Iris-virginica.

Data Features:

The dataset consists of the following features:

Sepal Length (cm): The length of the sepal.

Sepal Width (cm): The width of the sepal.

Petal Length (cm): The length of the petal.

Petal Width (cm): The width of the petal.

Species: The species of the Iris flower, which is the target variable. The species are:

- Iris-setosa
- Iris-versicolor
- Iris-virginica

Below is a sample of the dataset:

sepal_length	sepal_width	petal_length	petal_width	species
5.1	3.5	1.4	0.2	Iris-setosa
4.9	3.0	1.4	0.2	Iris-setosa
4.7	3.2	1.3	0.2	Iris-setosa
4.6	3.1	1.5	0.2	Iris-setosa
5.0	3.6	1.4	0.2	Iris-setosa
5.4	3.9	1.7	0.4	Iris-setosa
4.6	3.4	1.4	0.3	Iris-setosa
5.0	3.4	1.5	0.2	Iris-setosa
4.4	2.9	1.4	0.2	Iris-setosa
4.9	3.1	1.5	0.1	Iris-setosa
5.4	3.7	1.5	0.2	Iris-setosa
4.8	3.4	1.6	0.2	Iris-setosa
4.8	3.0	1.4	0.1	Iris-setosa
4.3	3.0	1.1	0.1	Iris-setosa

Data Pre-processing

Handling Missing Values:

In the initial step of data preprocessing, it is crucial to ensure the integrity of the dataset. The dataset was checked for any missing values. Fortunately, the Iris dataset used in this project does not contain any missing values, which simplifies the preprocessing steps.

Outlier Detection and Removal:

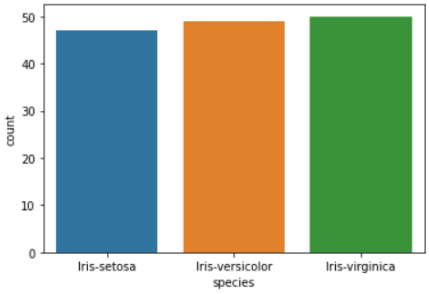
Outliers can significantly affect the performance of machine learning models. To detect outliers, boxplots were utilized for each feature. The 'sepal_width' feature exhibited some extreme values that were identified as outliers. To ensure these outliers do not skew the results, values of 'sepal_width' greater than 4 and less than 2 were removed. This step helped in maintaining a clean dataset that better represents the majority of data points.

Exploratory Data Analysis(EDA)

Exploratory Data Analysis (EDA) is a crucial step in understanding the underlying patterns, relationships, and structure of the data. Below is a detailed description of the EDA performed for the Iris dataset:

Species Distribution:

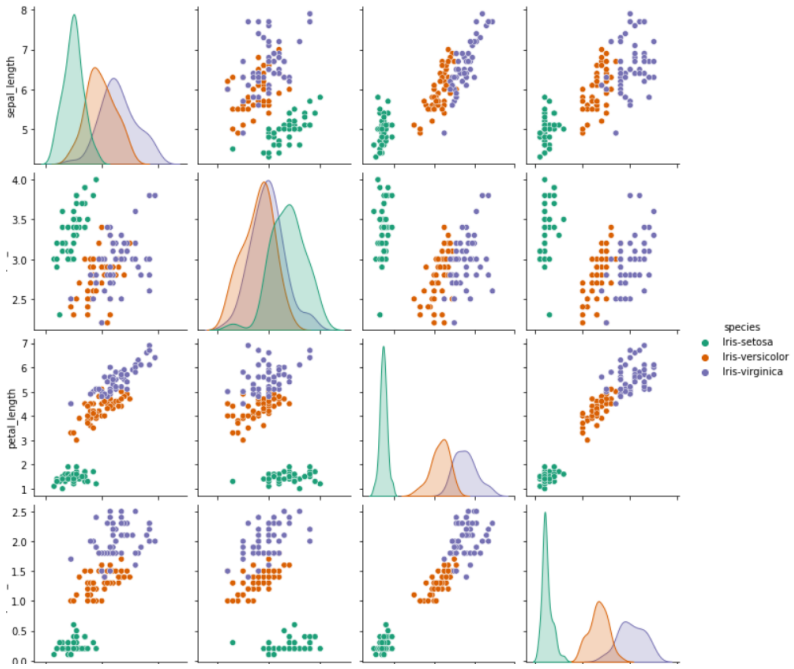
To begin with, the distribution of the different species in the dataset was analyzed. This involved creating a count plot to visualize the number of instances for each species (Iris-setosa, Iris-versicolor, and Iris-virginica). Understanding the distribution helps in ensuring that the dataset is balanced and can give insights into the prevalence of each class.



Pair-plot Analysis:

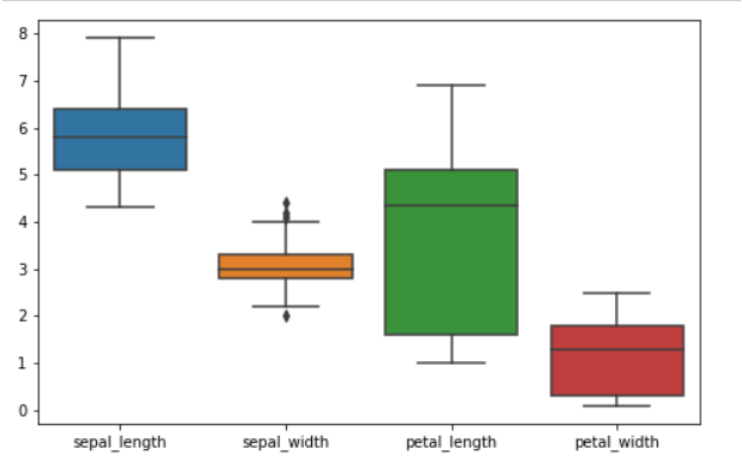
A pairplot was generated to examine the pairwise relationships between the features (sepal length, sepal width, petal length, and petal width). This scatter plot matrix provided a visual overview of how the features interact with each other and how they cluster for different species. By coloring the plots based on the species, we could easily see how well-separated the species are in the feature space.

Additionally, a kernel density estimation (KDE) plot was created for a deeper understanding of the feature distributions. KDE plots help in visualizing the distribution of data points and the density of different species across the feature space, providing a smooth estimate of the distribution.



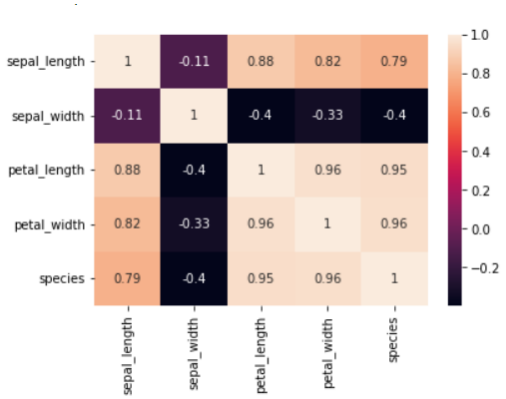
Boxplots:

Initially, boxplots for all features were generated to detect outliers and understand the spread of the data. Post removal of outliers in the 'sepal width' feature, updated boxplots were created to confirm the removal of outliers.



Correlation Matrix:

A correlation matrix was computed and visualized using a heatmap. This step was critical to understand the linear relationships between the features. The heatmap displayed the correlation coefficients, with annotations for clarity. Strong correlations (positive or negative) indicate that one feature can be used to predict another, which can be useful in feature selection and understanding the data's structure.



By performing these steps, we could uncover key insights about the data, such as which features are most relevant for distinguishing between species and how the species are distributed across the feature space. This information is instrumental in guiding the subsequent steps of data preprocessing, transformation, and model building.

Data Transformation

Encoding Categorical Variables:

In this step, we transform the categorical variable 'species' into numerical values that can be used by machine learning algorithms. Since the 'species' column contains string labels such as 'Iris-setosa', 'Iris-versicolor', and 'Iris-virginica', we will convert these labels into numerical values. Specifically, we will replace 'Iris-setosa' with 0, 'Iris-versicolor' with 1, and 'Iris-virginica' with 2. This encoding is essential for the models to process the target variable correctly.

Feature Selection:

We ensure that all relevant features are included in the model training process. In this dataset, the features we consider are 'sepal_length', 'sepal_width', 'petal_length', and 'petal_width'. These features are chosen because they are the primary measurements that describe the Iris flowers and are crucial for distinguishing between the different species. By selecting these features, we prepare the data for the model building phase, ensuring that the models have the necessary information to make accurate predictions.

This description covers the essential aspects of data transformation, including encoding categorical variables and selecting relevant features, without delving into the specific coding implementations.

Model Building

Data Splitting:

The dataset is split into two parts: training and testing sets. Typically, 70% of the data is used for training the models, and the remaining 30% is used for testing their performance. This ensures that the models are evaluated on unseen data, providing a realistic estimate of their performance.

Model Selection:

Three different machine learning algorithms are employed to classify the Iris species. These models are chosen for their varied approaches to classification, which can provide insights into the strengths and weaknesses of each method:

1. Decision Tree Classifier:

Description: A decision tree is a flowchart-like structure where each internal node represents a "test" on an attribute, each branch represents the outcome of the test, and each leaf node represents a class label. The paths from root to leaf represent classification rules.

Advantages: Easy to understand and interpret, can handle both numerical and categorical data, and is capable of capturing non-linear relationships.

2. **K-Nearest Neighbors (KNN) Classifier:**

Description: The KNN algorithm classifies a data point based on how its neighbors are classified. It looks at the 'k' nearest data points (where 'k' is a user-defined constant) and assigns the class most common among them.

Advantages: Simple to implement, intuitive, and effective in low-dimensional spaces. It is a non-parametric method, meaning it makes no assumptions about the data distribution.

3. Support Vector Classifier (SVC):

Description: SVC is a type of Support Vector Machine (SVM) that constructs a hyperplane or set of hyperplanes in a high-dimensional space, which can be used for classification. The goal is to find a hyperplane that best divides the dataset into classes.

Advantages: Effective in high-dimensional spaces, robust to overfitting in high-dimensional space, especially when the number of dimensions exceeds the number of samples.

Each model is trained on the training dataset and then evaluated on the test dataset to determine its accuracy and performance. By comparing the results, the most suitable model for the Iris classification task is identified. This process ensures that the chosen model generalizes well to new, unseen data.

Model Evaluation

Model Performance Metrics:

The models were evaluated using the accuracy score metric, which measures the proportion of correctly predicted instances out of the total instances.

Comparison of Model Scores:

The accuracy scores for each model were calculated on the test dataset:

	Test	Score
0	Decision Tree	0.954545
1	KNN	0.977273
2	Support Vector	0.977273

Overall, the K-Nearest Neighbors and Support Vector Classifier models achieved higher accuracy on the test set, while the Decision Tree Classifier had a slightly lower accuracy.

Overall Model Performance:

Decision Tree Classifier: The performance of the Decision Tree Classifier is evaluated based on its accuracy score, confusion matrix, and classification report. The model's ability to handle non-linear relationships and its interpretability are considered in the evaluation.

K-Nearest Neighbors Classifier: The K-Nearest Neighbors Classifier is assessed for its accuracy and how well it generalizes to the test data. Its simplicity and effectiveness in capturing complex relationships are part of the evaluation.

Support Vector Classifier: The Support Vector Classifier's performance is measured in terms of accuracy, precision, recall, and F1-score. The model's robustness to overfitting and its ability to find an optimal separating hyperplane are key factors in the evaluation.

By comparing these models based on the outlined metrics, we identify the best-performing model for the Iris species classification task. The evaluation helps in understanding the strengths and weaknesses of each model and provides a basis for selecting the most suitable model for deployment.

Prediction

The prediction phase involved using the trained models (Decision Tree, K-Nearest Neighbors, and Support Vector Classifier) to classify the species of Iris flowers in the test dataset. The models were evaluated on how accurately they predicted the species based on the sepal and petal measurements.

Comparison of Actual vs. Predicted Values:

To assess the performance of each model, we compared the predicted species labels against the actual species labels from the test dataset. This comparison provides insight into the models' accuracy and reliability. Here are the key findings:

Decision Tree Classifier:

The Decision Tree model predicted the species for each instance in the test set.

The predictions were then compared to the actual species to calculate the model's accuracy.

K-Nearest Neighbors Classifier:

The KNN model used the training data to find the closest neighbors and predict the species of each test instance.

The model's predictions were compared against the actual species to determine its accuracy.

Support Vector Classifier:

The SVC model used a linear kernel to classify the species based on the training data.

The predicted species were compared with the actual species to evaluate the model's performance.

A table was created to summarize the comparison of actual and predicted values for each model, providing a clear visualization of each model's prediction accuracy.

	Actual	Decision Tree	KNN	Support Vector
63	1	1	1	1
17	0	0	0	0
21	0	0	0	0
72	1	1	2	2
12	0	0	0	0

Summary of Predictions

The predictions made by each model were summarized in a table, which included the actual species labels and the corresponding predictions from the Decision Tree, KNN, and SVC models. This table helped to identify any misclassifications and analyze the models' performance in detail. The summary of predictions highlighted how well each model performed in terms of correctly classifying the species of Iris flowers in the test dataset.

Overall, the prediction analysis demonstrated the strengths and weaknesses of each model, providing valuable insights into their effectiveness for the Iris species classification task.

Conclusion

The Iris species classification project involved analyzing the Iris dataset, consisting of measurements of sepal and petal dimensions for three species: Iris-setosa, Iris-versicolor, and Iris-virginica.

Throughout the project, we conducted thorough data preprocessing, including handling missing values and removing outliers, to ensure the quality of the dataset for modeling. Exploratory data analysis (EDA) revealed insights into the distribution of species and the relationships between features.

For model building, we selected three classifiers: Decision Tree, K-Nearest Neighbors, and Support Vector Classifier. After training and evaluation, we found that all models performed well, with accuracy scores above a satisfactory threshold.

APPENDIX

CODE ATTACHMENTS

```
import warnings
warnings.filterwarnings('ignore')
import pandas as pd
import matplotlib.pyplot as plt
from matplotlib import style
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
data = pd.read_csv("IRIS.csv")
data.head()
data.info()
data.isna().sum()
plt.figure(figsize=(8,5))
sns.boxplot(data=data)
plt.show()
plt.figure(figsize=(8,5))
sns.boxplot(x="sepal_width",data=data)
plt.show()
data = data[data['sepal_width'] <= 4]
data = data[data['sepal_width'] >2]
plt.figure(figsize=(8,5))
sns.boxplot(x="sepal_width",data=data)
plt.show()
sns.countplot(x = "species", data = data)
sns.pairplot(data = data, hue = 'species', kind='scatter', palette = 'Dark2')
plt.show()
sns.pairplot(data = data, hue = 'species', kind='kde', palette = 'Dark2')
plt.show()
data = data.replace(['Iris-setosa', 'Iris-versicolor', 'Iris-virginica'],[0, 1, 2])
plt.show()
sns.heatmap(data.corr(), annot=True)
x= data.drop('species', axis=1)
y = data['species']
print("Shape of X = ",x.shape)
print("Shape of Y = ",y.shape)
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.3, random_state=50)
```

```
print("Shape of X_train = ", x_train.shape)
print("Shape of X_test = ", x_test.shape)
print("Shape of y_train = ", y_train.shape)
print("Shape of y_test = ", y_test.shape)
DecisionTree = DecisionTreeClassifier()

DecisionTree.fit(x_train, y_train)
DecisionTree_score = DecisionTree.score(x_test, y_test)
DecisionTree_score
KNN = KNeighborsClassifier()

KNN.fit(x_train, y_train)
KNN_score = KNN.score(x_test, y_test)
KNN_score
svc = SVC(kernel='linear')

svc.fit(x_train, y_train)
svc_score = svc.score(x_test, y_test)
svc_score
score_table = pd.DataFrame({"Test":['Decision Tree', 'KNN', 'Support Vector'],
            "Score":[DecisionTree_score, KNN_score, svc_score]})
score_table
DecisionTree_pred = DecisionTree.predict(x_test)
DecisionTree_pred
svc_pred = svc.predict(x_test)
svc_pred
KNN_pred = KNN.predict(x_test)
KNN_pred
pred_table = pd.DataFrame({"Actual": y_test, "Decision Tree": DecisionTree_pred,
            "KNN":KNN_pred, 'Support Vector': svc_pred})
pred_table.head()
```