Project 4: Iris Classification

Introduction

The Iris Classification Project demonstrates the application of machine learning techniques to a classic classification problem. The goal is to classify iris flowers into one of three species (**Setos, Versicolour, Virginica**) based on sepal and petal dimensions. This project includes extensive exploratory data analysis, robust model development, and practical deployment of the predictive model. This report provides a detailed account of the project workflow, including insights, results, and the deployment process.

Objectives

- Develop a machine learning model to classify iris flowers into three species with high accuracy.
- Analyze the dataset to uncover patterns and feature importance.
- Deploy the model to provide a real-world application for predicting iris species.

Problem Statement

The model must:

- Achieve high classification accuracy.
- Provide consistent and reliable predictions validated through cross-validation.
- Be deployable as an application for real-world use.

Dataset Overview

-Dataset Source: Unfied Mentor Data Analytics Project File

-Features:

- SepalLengthCm
- SepalWidthCm
- PetalLengthCm
- PetalWidthCm

-**Target Variable:** Species (Setosa, Versicolour, Virginica)

- **Sample Size:** 150 observations with no missing values. The dataset is balanced, with 50 samples for each species.

Exploratory Data Analysis (EDA)

a. Initial Observations:

- The dataset contains no missing or duplicate values.
- Numerical columns have varying distributions.

b. Key Visualizations:

- **Pairwise Relationships**: Scatter plots showed clear separability for Setosa and overlapping clusters for Versicolour and Virginica.
- **Feature Distributions**:
    - PetalWidthCm and PetalLengthCm are the most distinct features.
    - Sepal dimensions overlap significantly between classes.
- **Correlation Heatmap**:
    - Strong correlation between PetalWidthCm and PetalLengthCm.

c. Feature Ratios:

- Sepal and Petal ratios provided additional insights into separability.

Model Development

a. Preprocessing Steps:

- Encoded the categorical target variable.
- Standardized numerical features to ensure consistent scaling.

b. Models Evaluated:

- Logistic Regression
- K-Nearest Neighbors (KNN)
- Support Vector Machine (SVM)
- Decision Tree
- Random Forest (Best Model)

c. Hyperparameter Tuning:

- Random Forest parameters were optimized using GridSearchCV.

d. Best Model Performance:

- **Accuracy**: 98%
- **Precision, Recall, F1-Score**: High values across all species.
- **Confusion Matrix**: Minimal misclassifications for Versicolour and Virginica.

Results & Insights

Results:

-Model Performance:

- The Random Forest model outperformed other algorithms, achieving an impressive **98% accuracy** on the test data.
- Evaluation metrics such as **precision, recall, and F1-score** confirmed the model's reliability and consistency across all three species.
- The **confusion matrix** highlighted minimal misclassifications, particularly between **Versicolour** and **Virginica**, showcasing the model's robustness.

- Feature Importance:

- **Petal features (PetalLengthCm and PetalWidthCm)** were the most significant predictors of species classification.
- Sepal features were less influential but still contributed to the overall accuracy of the model.
- These findings align with the biological characteristics of iris flowers, where petal dimensions exhibit more distinct variation between species.

-EDA Findings:

- The **pairwise scatter plots** and **correlation heatmap** revealed that Setosa species are linearly separable, making them easier to classify.
- **Versicolour** and **Virginica** overlap significantly in feature space, requiring complex decision boundaries for accurate separation.
- Feature engineering, such as creating petal and sepal length-to-width ratios, provided additional insights and separability.

Insights:

-Dataset Characteristics:

- The Iris dataset is balanced, with an equal number of samples for each species, which ensures unbiased training and evaluation of the model.
- No missing or duplicate values were present, reducing the need for extensive data cleaning.

-Algorithm Effectiveness:

- Simpler models like Logistic Regression and KNN showed good performance but struggled with overlapping species clusters.
- Ensemble methods like Random Forest handled complex decision boundaries effectively, justifying its selection as the final model.

-Visual Interpretations:

- Visualizations confirmed the biological distinction of the Setosa species and highlighted the challenges of differentiating between Versicolour and Virginica.
- These insights can guide future studies or real-world applications in botany or horticulture.

Model Deployment (http://127.0.0.1:5000)

a. Deployment Tools:

- Flask for web application.
- Pickle for saving the trained model.

b. Folder Structure:

```
iris_classification/
│
├── app.py              # Flask application file
├── iris_model.pkl      # Saved machine learning model
├── templates/          # Folder for HTML templates
│   ├── index.html      # Main UI file
└── static/             # Folder for CSS files
    ├── style.css       # Optional styling file
```

c. Deployment Code:

Created Flask App in app.py

```python
app.py > home
from flask import Flask, request, render_template
import pickle
import numpy as np

app = Flask(__name__)
model = pickle.load(open('iris_model.pkl', 'rb'))

@app.route('/')
def home():
    return render_template('index.html')

@app.route('/predict', methods=['POST'])
def predict():
    try:
        features = [float(x) for x in request.form.values()]
        final_features = np.array(features).reshape(1, -1)
        prediction = model.predict(final_features)
        species = {0: 'Setosa', 1: 'Versicolour', 2: 'Virginica'}
        output = species[prediction[0]]
        return render_template('index.html', prediction_text=f'The Iris species is: {output}')
    except:
        return render_template('index.html', prediction_text="Error in prediction. Please check your input.")

if __name__ == '__main__':
    app.run(debug=True)
```

HTML Interface

```html
templates > <> index.html > ...
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Iris Flower Classifier</title>
</head>
<body>
    <h1>Iris Flower Classification</h1>
    <form action="/predict" method="POST">
        <label>Sepal Length:</label>
        <input type="text" name="sepal_length" required>

        <label>Sepal Width:</label>
        <input type="text" name="sepal_width" required>

        <label>Petal Length:</label>
        <input type="text" name="petal_length" required>

        <label>Petal Width:</label>
        <input type="text" name="petal_width" required>

        <button type="submit">Predict</button>
    </form>
    <h2>{{ prediction_text }}</h2>
</body>
```

Conclusion

The Iris Classification Project represents a successful implementation of machine learning techniques applied to a real-world dataset. The project highlights the importance of following a structured pipeline, from data exploration to model deployment, to achieve accurate and reliable results.

The detailed exploratory data analysis (EDA) uncovered significant insights about the dataset, such as the importance of petal dimensions and the unique separability of the Setosa species. The results confirmed that petal features are the primary determinants of species classification, while sepal features play a secondary role. This understanding is essential for both machine learning applications and biological studies.

The development and evaluation of multiple machine learning models, including Logistic Regression, KNN, SVM, Decision Trees, and Random Forests, demonstrated the necessity of experimenting with various algorithms. Random Forest emerged as the best-performing model due to its robustness in handling overlapping feature spaces and its ability to deliver high accuracy consistently. Hyperparameter tuning further optimized its performance, showcasing the effectiveness of grid search techniques in machine learning workflows.

One of the most significant achievements of this project is the deployment of the model as a Flask application. This deployment bridges the gap between theoretical work and practical application, allowing users to predict iris species based on flower measurements interactively. The organized folder structure, modular code design, and user-friendly interface ensure the application's scalability and ease of use.

Future Work:

1. Incorporate additional features such as flower color or geographical location to improve classification accuracy and applicability.
2. Explore advanced algorithms like XGBoost or Neural Networks for potential performance gains.
3. Deploy the model on a cloud platform like AWS, Heroku, or Google Cloud for global accessibility.
4. Extend the deployment by creating APIs or mobile applications to integrate the model with broader systems.

This project exemplifies how data science and machine learning can transform raw data into actionable insights and practical solutions. It not only achieved the objective of accurate iris species classification but also demonstrated the potential for machine learning applications in various domains, including agriculture, biology, and education.