Report on Iris Flower Classification

Asmaa Nasr

2024/9/18

1 Introduction

This report outlines the approach taken to classify iris flowers based on their physical dimensions using a dataset containing measurements of different iris species. The classification task aimed to predict the species of iris (Iris-setosa, Iris-versicolor, Iris-virginica) based on four features: sepal length, sepal width, petal length, and petal width.

2 Data Cleaning and Preprocessing

2.1 Steps Taken

1. **Loading the Dataset**: The dataset was loaded using `pandas` from a CSV file.

   ```
   df = pd.read_csv('dataset.csv')
   ```

2. **Checking for Missing Values**: We examined the dataset for any missing values and duplicates to ensure data integrity.

   ```
   print(df.isnull().sum())
   print(f"Duplicates: {df.duplicated().sum()}")
   ```

   The check for missing values showed that there were no missing entries in any of the features:

   ```
   Sepal_Length    0
   Sepal_Width     0
   Petal_Length    0
   Petal_Width     0
   Class           0
   dtype: int64
   ```

3. **Encoding Categorical Variables**: The target variable `Class` was encoded into numeric values using `LabelEncoder`. This step was crucial for the model to process the categorical data correctly.

   ```
   from sklearn.preprocessing import LabelEncoder
   label_encoder = LabelEncoder()
   df['Class'] = label_encoder.fit_transform(df['Class'])
   ```

4. **Check for class balance**: Understanding the class distribution in our dataset is crucial for building an effective machine learning model. Class distribution refers to the number of instances belonging to each category—in our case, the different species of iris flowers.

   ```
   class_distribution = data_cleaned['Class'].value_counts()
   ```

   The distribution of classes was checked, revealing a balanced dataset:

   ```
   Iris-versicolor    50
   Iris-virginica     49
   Iris-setosa        48
   ```

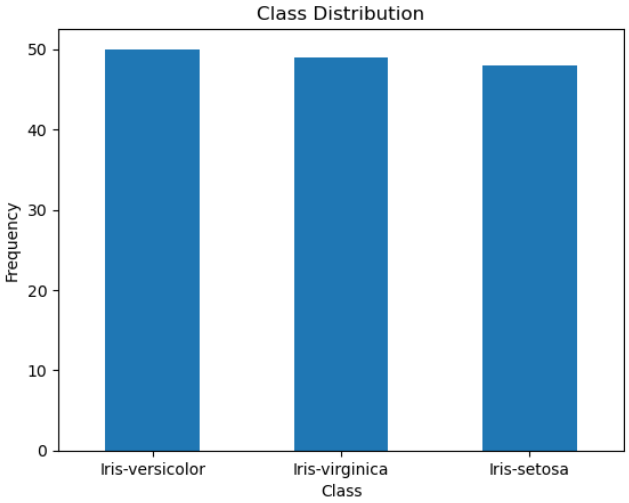

Figure 1: Class distribution

5. **Checking for Duplicates**: The dataset contained duplicates, which were identified and counted:

```
Number of duplicates: 3
```

6. **Removing Duplicates** After removing duplicates, the dataset was cleaned, and the index was reset:

   ```
   data_cleaned = data.drop_duplicates()
   data_cleaned.reset_index(drop=True, inplace=True)
   ```

3 Feature Selection

The selected features were:

- **Sepal Length**
- **Sepal Width**
- **Petal Length**
- **Petal Width**

3.1 Rationale for Selection

These features were chosen because they provide distinct measurements that differentiate between the iris species. The dataset does not contain redundant or irrelevant features, making it straightforward for model training.

4 Machine Learning Algorithm

4.1 Chosen Algorithm: Random Forest Classifier

The Random Forest algorithm was selected for several reasons:

- **Robustness**: It can handle non-linear relationships and interactions between features effectively.
- **Performance**: Random Forest generally provides high accuracy and is less prone to overfitting compared to a single decision tree.
- **Feature Importance**: It provides insights into the importance of various features, which is useful for understanding the model's decisions.

5 Model Evaluation

5.1 Evaluation Metrics

The following metrics were used to evaluate the model's performance:

- **Accuracy**: The proportion of correctly predicted instances over the total instances.
- **Precision**: The ratio of true positive predictions to the total predicted positives.
- **Recall**: The ratio of true positive predictions to the total actual positives.
- **F1-Score**: The harmonic mean of precision and recall, providing a balance between the two.

5.2 Results

The feature importance was as:

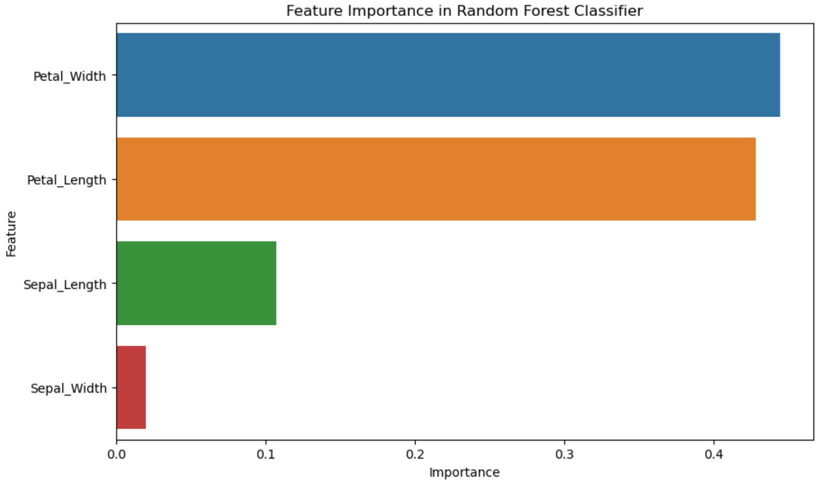

Figure 2: Feature Importance

After training the model and evaluating it on the test set, the results were as follows:

- **Accuracy**: 90%
- **Precision**: 0.91 (average for all classes)
- **Recall**: 0.91 (average for all classes)
- **F1-Score**: 0.91 (average for all classes)

The confusion matrix as shown in indicated that the model performed excellently across all classes, with very few misclassifications.

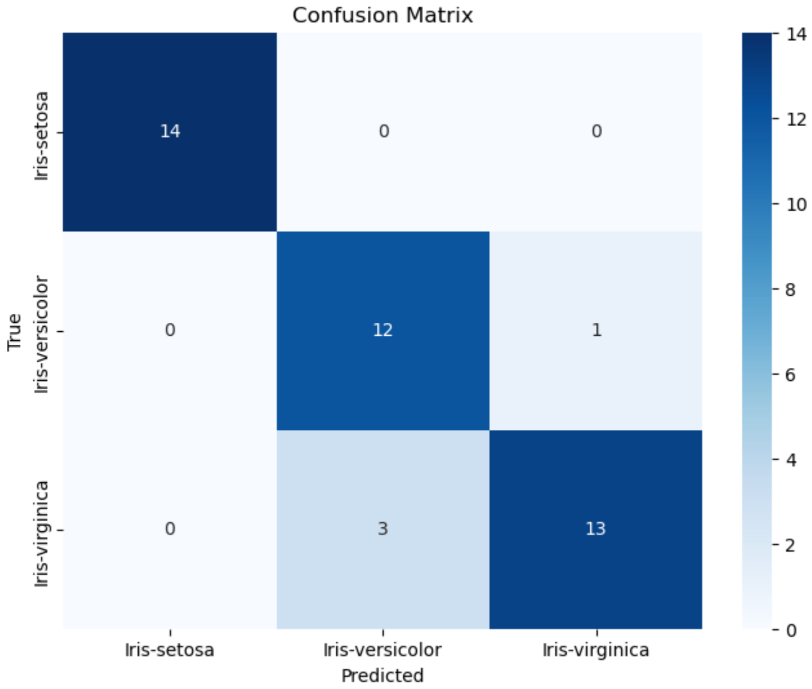

Figure 3: Confusion Matrix for Iris Flower Classification

6 Challenges Faced

6.1 Challenges

1. **Handling Categorical Data**: Initially, the target variable was in string format, which caused issues when fitting the model. This was resolved by applying `LabelEncoder`.

2. **Overfitting Concern**: Given the simplicity of the dataset, there was a potential risk of overfitting. However, the inherent nature of the Random Forest algorithm mitigated this issue.

6.2 Solutions

- **Encoding**: The use of `LabelEncoder` resolved the issue with categorical data.

- **Model Selection**: Choosing Random Forest helped in managing overfitting due to its ensemble nature.

7 Conclusion

The classification of iris flowers using the Random Forest algorithm proved to be highly effective, achieving a high accuracy of 90.7%. The steps taken in data cleaning and preprocessing were crucial for preparing the dataset for machine learning. The feature selection was justified based on domain knowledge, and the algorithm was chosen for its robustness and performance. Overall, this project exemplifies a successful application of machine learning techniques in a classic dataset.

7.1 Future Work

Future work could involve exploring other algorithms such as Support Vector Machines or Neural Networks, as well as performing hyperparameter tuning to potentially improve model performance further. Additionally, incorporating more features or expanding the dataset could provide more insights into iris classification.