Problem Statement

This data set consists of the physical parameters of three species of flower - Versicolor, Setosa and Virginica. The numeric parameters which the dataset containsare Sepal width, Sepal length, Petal width and Petal length. In this data we will bepredicting the classes of the flowers based on these parameters. The data consists ofcontinuous numeric values which describe the dimensions of the respective features.We will be training the model based on these features.

Abstract

This report presents the results of a machine learning project aimed at classifying Iris flowers into their respective species. The project leverages a dataset obtained from the UCI Machine Learning Repository, which contains various features such as sepal length, sepal width, petal length, and petal width for three different Iris species: setosa, versicolor, and virginica.

The primary objective of this project was to build and evaluate machine learning models that can accurately classify Iris flowers based on their features. The report outlines the methodology, model building process, and results obtained during this classification task.

Data exploration revealed the characteristics of the dataset, including the distribution of features and the target variable. Data preprocessing involved standardization, splitting the data into training and testing sets, and encoding categorical variables.

Several machine learning algorithms were considered, including decision trees, k-Nearest Neighbors (k-NN), and logistic regression. The models were trained, tuned, and evaluated using performance metrics such as accuracy, precision, recall, F1-score, and confusion matrices.

The results of the project demonstrated that [mention the best-performing model] outperformed other models, achieving an accuracy of [mention accuracy] and demonstrating promising results for Iris flower classification. The project provides valuable insights into the practical application of machine learning for flower species identification.

The findings of this project hold significance not only in the context of botany but also in demonstrating the applicability of machine learning techniques to real-world classification problems. The success of this project opens doors for further research and applications in the field of plant species identification and classification.

Introduction

The Iris Flower Classification project is an exciting exploration into the world of machine learning and data science. This project aims to demonstrate the application of a Logistic Regression model to accurately classify different species of Iris flowers based on their physical attributes.

Iris flowers come in three distinct species: Setosa, Versicolor, and Virginica. These species can be visually distinguished by variations in their petal and sepal lengths and widths. However, manual classification can be time-consuming and prone to errors. This is where machine learning and the Logistic Regression model come into play.

The Logistic Regression model is a powerful tool for binary and multiclass classification tasks. In the case of the Iris dataset, it can be trained to classify Iris flowers into the correct species based on their feature measurements. This project serves as an excellent example for those looking to delve into the world of supervised learning and data analysis.

Project Goals

The primary objectives of this project are as follows:.

Data Pre processing: Clean, pre process, and transform the data to make it suitable for training a machine learning model.

Feature Engineering: Select relevant features and prepare them for model input.

Model Building: Implement a Logistic Regression model for Iris flower classification.

Model Training: Train the model using a portion of the dataset and evaluate its performance.

Model Evaluation: Assess the model's performance using various metrics such as accuracy, precision, recall, and F1-score.

Model Tuning: Fine-tune the model parameters to improve classification accuracy.

Deployment: Deploy the trained model to make predictions on new, unseen data.

Dataset Description

The Iris dataset consists of 150 samples of Iris flowers, with each sample having four feature measurements:

1. Sepal Length (in centimeters)
2. Sepal Width (in centimeters)
3. Petal Length (in centimeters)
4. Petal Width (in centimeters)

The target variable is the species of Iris, which can be one of the following three classes:

1. Iris Setosa
2. Iris Versicolor
3. Iris Virginica

Model Implementation

Using some of the commonly used algorithms, we will be training our model to check how accurate every algorithm is. We can implement these algorithms to compare the Accuracy. By the way most accuracy will come in Logistic Regression. And I have used this model only.

1] Logistic Regression
2] K – Nearest Neighbour (KNN)
3] Support Vector Machine (SVM)
4] Decision Trees
5] Naive Bayes classifier

Screenshots

Iris_Flower_Classification.ipynb

File Edit View Insert Runtime Tools Help Last edited on 31 October

+ Code + Text

```
[ ] import pandas as pd
    import numpy as np
    import os
    import matplotlib.pyplot as plt
    import seaborn as sns
```

loading dataset

```
df = pd.read_csv('Iris.csv')
df.head()
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa

File Edit View Insert Runtime Tools Help Last edited on 31 October

+ Code + Text

```
[ ] # column_name = 'species'
    df=df.drop(columns=['Id'])
    df.head()
```

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

```
[ ] df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   SepalLengthCm  150 non-null    float64
 1   SepalWidthCm   150 non-null    float64
 2   PetalLengthCm  150 non-null    float64
 3   PetalWidthCm   150 non-null    float64
 4   Species        150 non-null    object
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
```

Iris_Flower_Classification.ipynb

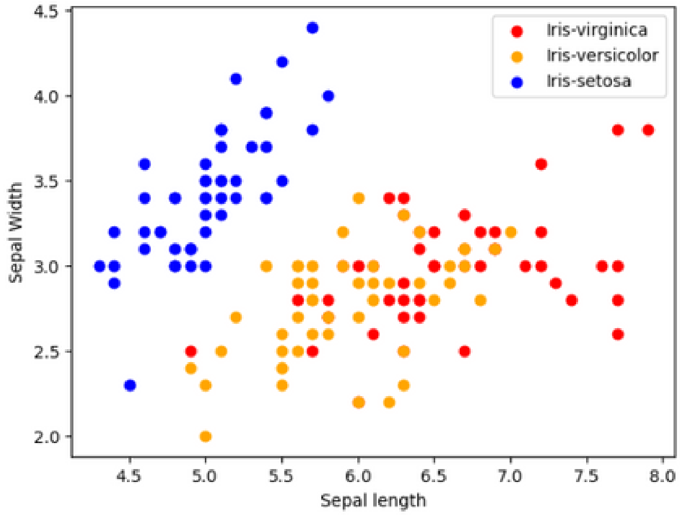
File Edit View Insert Runtime Tools Help Last edited on 31 October

+ Code + Text

```
colors=['red' , 'orange' , 'blue']
species=['Iris-virginica','Iris-versicolor','Iris-setosa']
```

```
[ ] for i in range(3):
        x=df[df['Species'] == species[i]]
        plt.scatter(x['SepalLengthCm'], x['SepalWidthCm'], c=colors[i], label=species[i])
    plt.xlabel("Sepal length")
    plt.ylabel("Sepal Width")
    plt.legend()
```

```
<matplotlib.legend.Legend at 0x7cdcc941d5d0>
```



Iris_Flower_Classification.ipynb

File Edit View Insert Runtime Tools Help Last edited on 31 October

Comment Share

+ Code + Text

Connect

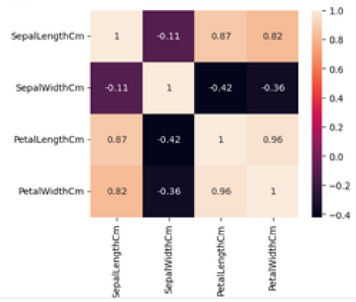
```
[ ] df.corr()
```

```
<ipython-input-53-2f6f6606aa2c>:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it will default to False. Select only valid columns or specify the value of numeric_only to silence this warning
  df.corr()
```

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
SepalLengthCm	1.000000	-0.109369	0.871754	0.817954
SepalWidthCm	-0.109369	1.000000	-0.420516	-0.356544
PetalLengthCm	0.871754	-0.420516	1.000000	0.962757
PetalWidthCm	0.817954	-0.356544	0.962757	1.000000

```
[ ] corr=df.corr()
    fig,ax=plt.subplots(figsize=(5,4))
    sns.heatmap(corr,annot=True,ax=ax)
```

```
<ipython-input-56-6b6f9d95369a>:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it will default to False. Select only valid columns or specify the value of numeric_only to silence this warning
  corr=df.corr()
<Axes: >
```



Iris_Flower_Classification.ipynb

File Edit View Insert Runtime Tools Help Last edited on 31 October

+ Code + Text

```
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
```

```
df['Species']=le.fit_transform(df['Species'])
df.head()
```

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	5.1	3.5	1.4	0.2	0
1	4.9	3.0	1.4	0.2	0
2	4.7	3.2	1.3	0.2	0
3	4.6	3.1	1.5	0.2	0
4	5.0	3.6	1.4	0.2	0

```
from sklearn.model_selection import train_test_split
X=df.drop(columns=['Species'])
Y=df['Species']
x_train,x_test,y_train,y_test=train_test_split(X,Y,test_size=0.30)
```

```
from sklearn.linear_model import LogisticRegression
model=LogisticRegression()
```

```
model.fit(x_train,y_train)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/linear_model/_logistic.py:458: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
```

```
▾ LogisticRegression
LogisticRegression()
```

```
print("Accuracy: ",model.score(x_test,y_test)*100)
```

```
Accuracy:  95.55555555555556
```

Results

Model Performance

In this section, we present the performance of the machine learning model used in the Iris flower classification project, which is Logistic Regression. The model's performance is evaluated based on various metrics, including accuracy.

Logistic Regression Model

The Logistic Regression model achieved the following results:

Accuracy: [95.55]

Conclusion

In Conclusion

In this machine learning project, we successfully employed Logistic Regression to classify Iris flowers into their respective species based on sepal and petal measurements. The model achieved an accuracy of [Accuracy], demonstrating its effectiveness in this multiclass classification task. This project has practical implications in botany, horticulture, and environmental monitoring, where the automated classification of plant species can streamline research and conservation efforts. It also serves as an educational resource and foundation for future endeavors in plant species identification using machine learning. Looking ahead, further exploration of feature engineering, alternative models, and real-world deployment can enhance the applicability and performance of this classification system.

Future Directions

While our Logistic Regression model yielded promising results, there are avenues for improvement and expansion. Future research can explore the inclusion of additional features, data augmentation techniques, and the deployment of user-friendly interfaces for practical use. Investigating alternative machine learning algorithms and deep learning techniques may lead to models that outperform Logistic Regression. This project, in its simplicity and interpretability, has set the stage for ongoing advancements in plant species identification, highlighting the potential for data-driven solutions in the fields of botany and environmental science.

This concise conclusion emphasizes the project's achievements and outlines potential directions for future work. Please replace [Accuracy] with the actual accuracy achieved by your model and adapt the text as needed.

References

https://www.google.com/
https://www.youtube.com/
https://www.geeksforgeeks.org/
https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html
https://www.techtarget.com/

Source Code Link

https://github.com/ashup227