CAPSTONE PROJECT

IRIS FLOWER CLASSIFICATION

Presented By:
Anuciya.P
University College Of Engineering ,Villupuram
B.Tech-Information Technology

edunet foundation

OUTLINE

- Problem Statement
- Proposed System/Solution
- System Development Approach
- Algorithm & Deployment
- Result
- Conclusion
- Future Scope
- References

PROBLEM STATEMENT

The Iris flower dataset consists of three species: setosa, versicolor,and virginica.These species can be distinguished based on their measurements. Now, imagine that you have the measurementsof Iris flowers categorized by their respective species. Yourobjective is to train a machine learning model that can learn fromthese measurements and accurately classify the Iris flowers intotheir respective species.Use the Iris dataset to develop a model that can classify irisflowers into different species based on their sepal and petalmeasurements. This dataset is widely used for introductoryclassification tasks.

PROPOSED SOLUTION

1. Import Necessary Libraries

2. Load and Explore the Dataset

3. Data Preprocessing

- Handle missing values
- Outlier detection and treatment
- Feature scaling

4. Split the Data

5. Model Selection and Training

6. Model Evaluation

- Evaluate the trained models on the test set using metrics like accuracy, and confusion matrix.
- Choose the best-performing model based on the evaluation metrics.

7. Model Deployment

- If required, deploy the chosen model for real-time predictions.

edunet foundation

SYSTEM APPROACH

The "System Approach" section outlines the overall strategy and methodology for developing and implementing the Iris flower classification. Here's a suggested structure for this section:

- Data Acquisition and Preprocessing
- Data Splitting
- Model Selection and Training
- Model Evaluation
- Deployment Environment

ALGORITHM & DEPLOYMENT

1. Algorithm

The Iris dataset is a classic example of a multi-class classification problem. Given its relatively small size and clear separation between classes, several algorithms can achieve high accuracy. Here are some popular choices:

- Logistic Regression
- Training
- Testing
- Accuracy
- Confusion Matrix

2.Deployment

once you've otrained and evaluated your model, the next step is deployment. This involves making the model accessible for use in applications.

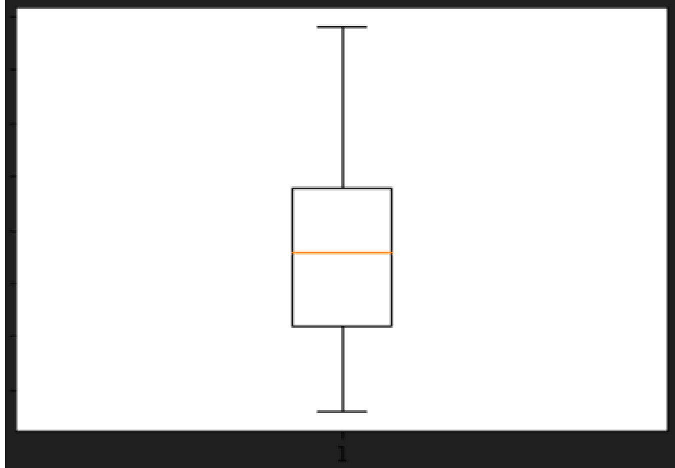
- Web application
- API
- Mobile application
- Cloud platform

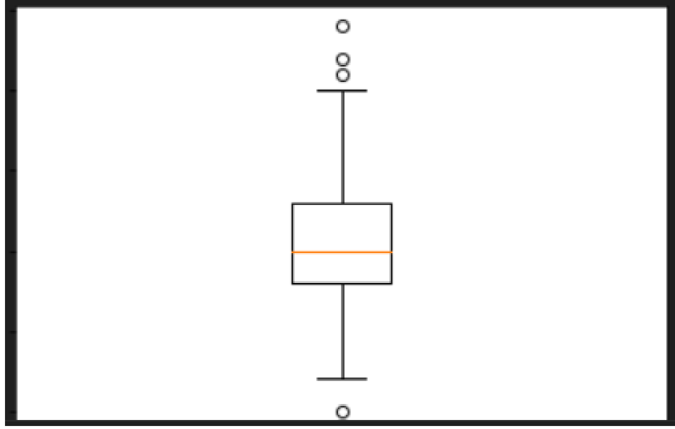
```python
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

```python
iris=pd.read_csv("/content/irisdata.csv")
print(iris)
```

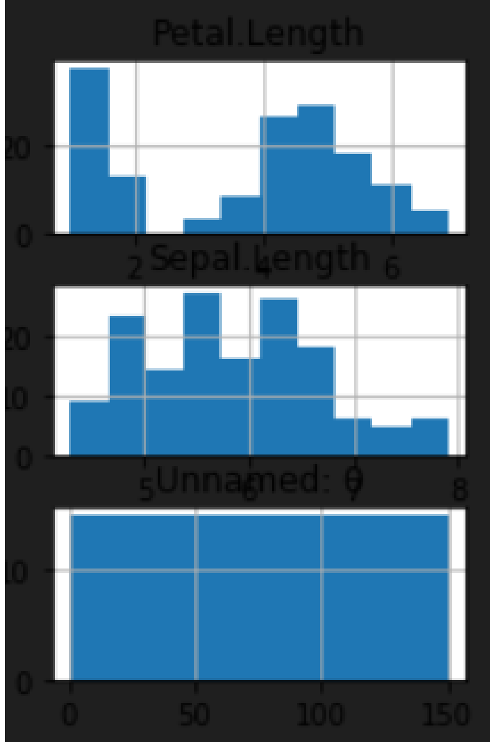
```
     sepal_length  sepal_width  petal_length  petal_width         species
0             5.1          3.5           1.4          0.2     Iris-setosa
1             4.9          3.0           1.4          0.2     Iris-setosa
2             4.7          3.2           1.3          0.2     Iris-setosa
3             4.6          3.1           1.5          0.2     Iris-setosa
4             5.0          3.6           1.4          0.2     Iris-setosa
..            ...          ...           ...          ...             ...
145           6.7          3.0           5.2          2.3  Iris-virginica
146           6.3          2.5           5.0          1.9  Iris-virginica
147           6.5          3.0           5.2          2.0  Iris-virginica
148           6.2          3.4           5.4          2.3  Iris-virginica
149           5.9          3.0           5.1          1.8  Iris-virginica

[150 rows x 5 columns]
```

```
import matplotlib.pyplot as plt
plt.figure(1)
plt.boxplot([iris['Sepal.Length']])
plt.figure(2)
plt.boxplot([iris['Sepal.Width']])
plt.show()
```



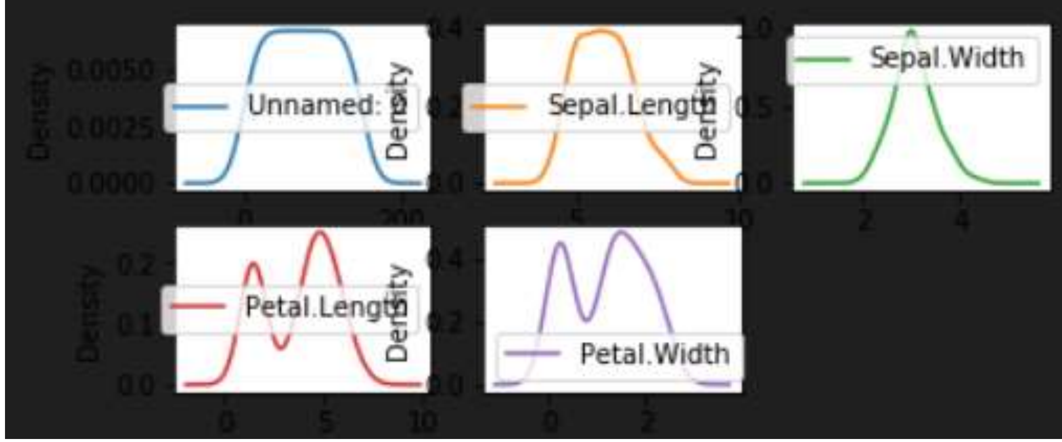


```
iris.hist()
plt.show()
```



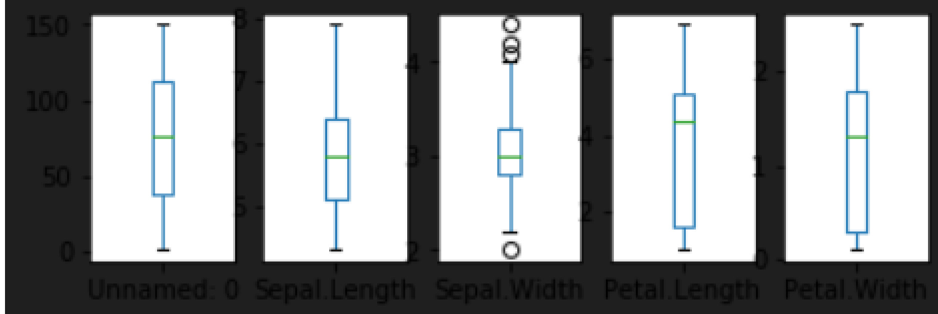
```python
iris.plot(kind ='density',subplots = True, layout =(3,3),sharex = False)
```

```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x000002900CD2EA48>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x000002900CDA7048>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x000002900CDD9E88>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x000002900CE0F0C8>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x000002900CE43A88>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x000002900CE7D488>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x000002900CEB6448>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x000002900CEF4E48>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x000002900CEFC248>]],
      dtype=object)
```

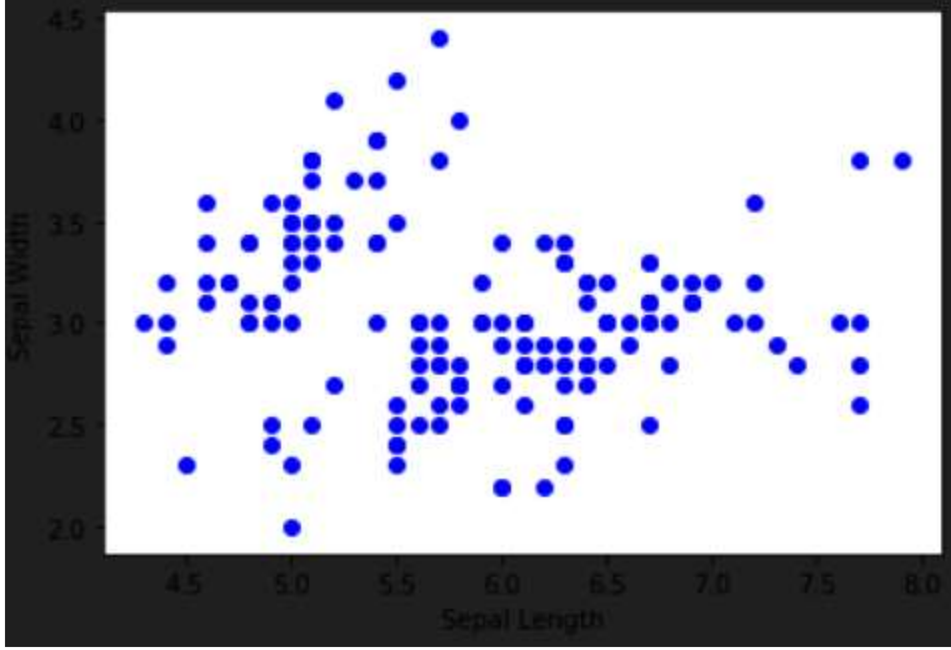


```
iris.plot(kind ='box',subplots = True, layout =(2,5),sharex = False)
```

```
Unnamed: 0        AxesSubplot(0.125,0.536818;0.133621x0.343182)
Sepal.Length    AxesSubplot(0.285345,0.536818;0.133621x0.343182)
Sepal.Width      AxesSubplot(0.44569,0.536818;0.133621x0.343182)
Petal.Length    AxesSubplot(0.606034,0.536818;0.133621x0.343182)
Petal.Width     AxesSubplot(0.766379,0.536818;0.133621x0.343182)
dtype: object
```



```
plt.xlabel("Sepal Length")
plt.ylabel("Sepal Width")
plt.scatter(X,Y,color='b')
plt.show()
```



```python
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn import svm
from sklearn import metrics
from sklearn.tree import DecisionTreeClassifier
```

```python
train, test = train_test_split(iris, test_size = 0.25)
print(train.shape)
print(test.shape)
```

```
(112, 6)
(38, 6)
```

```python
train_X = train[['Sepal.Length', 'Sepal.Width', 'Petal.Length',
                 'Petal.Width']]
train_y = train.Species

test_X = test[['Sepal.Length', 'Sepal.Width', 'Petal.Length',
               'Petal.Width']]
test_y = test.Species
```

```python
train_X.head()
```

```python
train_X.head()
```

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
99	5.7	2.8	4.1	1.3
22	4.6	3.6	1.0	0.2
86	6.7	3.1	4.7	1.5
50	7.0	3.2	4.7	1.4
33	5.5	4.2	1.4	0.2

```python
test_y.head()
```

```
90     versicolor
92     versicolor
147     virginica
16         setosa
82     versicolor
Name: Species, dtype: object
```

OUTPUT – ACCURACY AND CONFUSION MATRIX

```
model = LogisticRegression()
model.fit(train_X, train_y)
prediction = model.predict(test_X)
print('Accuracy:',metrics.accuracy_score(prediction,test_y))
```

```
Accuracy: 0.9210526315789473
```

```
#Confusion matrix
from sklearn.metrics import confusion_matrix
confusion_mat = confusion_matrix(test_y,prediction)
print("Confusion matrix: \n",confusion_mat)
```

```
Confusion matrix:
 [[11  0  0]
 [ 0 15  3]
 [ 0  0  9]]
```

RESULT

Typically, when applying various machine learning algorithms to the Iris dataset, you can expect high accuracy rates. This is due to the dataset's relatively simple nature with clear separation between the three Iris species.

Accuracy: Often above 90%, frequently reaching 95-98% or even higher.

Confusion Matrix: Generally, a diagonal matrix with high values on the diagonal, suggesting correct classifications.

CONCLUSION

- The Iris flower classification project has been a successful exploration of machine learning concepts. The ability to accurately predict Iris species based on their measurements highlights the potential of these techniques for various applications. As we move forward, applying these learnings to more complex datasets will be crucial for developing robust and scalable machine learning solutions.

FUTURE SCOPE

The Iris dataset has served as a valuable stepping stone for many machine learning practitioners,

its simplicity limits its potential for groundbreaking resources.

1.Expand the dataset.

2. Advanced Machine Learning Techniques.

3.Real world applications.

4. Edge Computing.

REFERENCES

- The Iris dataset is a classic dataset often used as a starting point for machine learning projects. While it might not have extensive research dedicated solely to it, there are numerous resources available that cover the fundamentals of machine learning and classification, which can be applied to the Iris dataset.

COURSE CERTIFICATE 1

In recognition of the commitment to achieve professional excellence

Getting Started with Enterprise-grade AI

IBM

ANUCIYA P

Has successfully satisfied the requirements for:

Getting Started with Enterprise-grade AI

Issued on: 09 JUL 2024

Issued by IBM

Verify: https://www.credly.com/go/mbuLxGFY

IBM

edunet foundation

COURSE CERTIFICATE 2

In recognition of the commitment to achieve professional excellence

Getting Started with Enterprise Data Science

ANUCIYA P

Has successfully satisfied the requirements for:

Getting Started with Enterprise Data Science

Issued on: 11 JUL 2024

Issued by IBM

Verify: https://www.credly.com/go/ZaXOoGm1

IBM

edunet foundation

THANK YOU

edunet foundation