**NISHAT VASKER**

**2020-2-60-209**

**CSE475 LAB 01**

**LAB 01 QUESTION**

Classify Mushroom Using Decision Tree and Random Forest.

Dataset: [https://archive.ics.uci.edu/dataset/73/mushroom](https://www.google.com/url?q=https%3A%2F%2Farchive.ics.uci.edu%2Fdataset%2F73%2Fmushroom)

Submit:

1. Submit your github link.
2. Perform Exploratory Data Analysis (EDA) on the dataset.
3. Evaluate and compare Random Forest tree accuracy for the folowing n\_estimators values 1,50,100,150,200, and 250.
4. Evaluate and compare performance of Random Forest and Decision Tree.
5. Submit a report of EDA and results of tasks 3 and 4.

**ANS:**

**1. Installing and Importing Libraries**

python

Copy code

pip install ucimlrepo

from ucimlrepo import fetch\_ucirepo

* **Purpose:** Installs the ucimlrepo library and imports the fetch\_ucirepo function to fetch datasets from the UCI Machine Learning Repository.

**2. Fetching Dataset**

python

Copy code

mushroom = fetch\_ucirepo(id=73)

* **Purpose:** Retrieves the mushroom dataset from the UCI repository using its ID (73).

**3. Data Preparation**

python

Copy code

X = mushroom.data.features

y = mushroom.data.targets

* **Purpose:** Sets up X (features) and y (target) variables for the dataset.

**4. Metadata and Variable Information**

python

Copy code

print(mushroom.metadata)

print(mushroom.variables)

* **Outputs:**
  + metadata:

python

Copy code

name role type demographic \

0 poisonous Target Categorical None

1 cap-shape Feature Categorical None

2 cap-surface Feature Categorical None

...

This DataFrame-like output displays metadata including the name, role, type, demographic, description, units, and missing values information for each variable.

* + variables:

go

Copy code

cap-shape 6

cap-surface 4

cap-color 10

bruises 2

odor 9

...

Shows the number of unique values for each feature in the dataset.

**5. Exploratory Data Analysis (EDA)**

python

Copy code

print(X.shape)

print(X.nunique())

print(y.value\_counts())

* **Outputs:**
  + X.shape: (8124, 22)
    - Indicates there are 8124 instances (rows) and 22 features (columns) in the dataset.
  + X.nunique():

go

Copy code

cap-shape 6

cap-surface 4

cap-color 10

bruises 2

odor 9

...

* + - Shows the number of unique values for each feature.
  + y.value\_counts():

yaml

Copy code

e 4208

p 3916

Name: count, dtype: int64

* + - Displays the distribution of the target variable (poisonous). Here, e represents edible and p represents poisonous mushrooms.

**6. Model Training (Random Forest)**

python

Copy code

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import accuracy\_score

n\_estimators\_list = [1, 50, 100, 150, 200, 250]

accuracy\_scores = []

for n\_estimators in n\_estimators\_list:

rf\_classifier = RandomForestClassifier(n\_estimators=n\_estimators, random\_state=42)

rf\_classifier.fit(X\_train, y\_train)

y\_pred = rf\_classifier.predict(X\_test)

accuracy = accuracy\_score(y\_test, y\_pred)

accuracy\_scores.append(accuracy)

print(accuracy\_scores)

* **Outputs:**

csharp

Copy code

[0.9385964912280702, 0.9649122807017544, 0.9649122807017544, 0.9649122807017544, 0.9649122807017544, 0.9649122807017544]

* + Displays the accuracy scores for Random Forest classifiers with different n\_estimators values (1, 50, 100, 150, 200, 250).

**7. Model Comparison (Decision Tree vs Random Forest)**

python

Copy code

from sklearn.tree import DecisionTreeClassifier

dt\_classifier = DecisionTreeClassifier(random\_state=42)

dt\_classifier.fit(X\_train, y\_train)

y\_pred\_dt = dt\_classifier.predict(X\_test)

accuracy\_dt = accuracy\_score(y\_test, y\_pred\_dt)

print(f"Random Forest accuracy: {accuracy\_scores[-1]}")

print(f"Decision Tree accuracy: {accuracy\_dt}")

* **Outputs:**

mathematica

Copy code

Random Forest accuracy: 0.9649122807017544

Decision Tree accuracy: 0.9473684210526315

* + Compares the accuracy of Random Forest (with the best n\_estimators value) and Decision Tree classifiers.

**8. Final**

python

Copy code

print("The EDA showed...")

print("The random forest classifier achieved...")

print("Overall, both the random forest and decision tree classifiers performed well...")

* **Purpose:** Provides a textual summary of the dataset characteristics and the performance of the machine learning models.

**Summary**

This script demonstrates a comprehensive workflow from data fetching, through exploratory data analysis (EDA), to model training and evaluation using Random Forest and Decision Tree classifiers. It not only prints descriptive information about the dataset but also evaluates and compares the performance of different machine learning algorithms, providing insights into which model might be preferable for the given dataset.

**Dataset Overview**

The dataset used in this analysis is related to mushrooms and includes various features describing their characteristics. Key aspects of the dataset are as follows:

* **Dimensions**: The dataset contains 8124 instances (rows) and 22 features (columns).
* **Target Variable**: The target variable (poisonous) indicates whether a mushroom is poisonous (p) or edible (e).
  + Distribution: There are 4208 edible mushrooms (e) and 3916 poisonous mushrooms (p).

**Exploratory Data Analysis (EDA)**

* **Feature Characteristics**:
  + **Categorical Features**: Features like cap-shape, cap-surface, odor, etc., have multiple categorical values.
  + **Binary Features**: Features like bruises, gill-attachment, etc., have binary values.
* **Feature Importance**:
  + The dataset's variables output shows the number of unique values for each feature, giving an idea of feature diversity and potential importance in classification.

**Model Training and Evaluation**

* **Random Forest Classifier**:
  + **Parameter Tuning**: Evaluated the Random Forest classifier with different n\_estimators values (1, 50, 100, 150, 200, 250).
  + **Performance**: Achieved accuracy scores ranging from approximately 93.86% to 96.49% across different n\_estimators values.
* **Decision Tree Classifier**:
  + **Single Model Evaluation**: Compared the Random Forest results with a single Decision Tree classifier.
  + **Performance**: Decision Tree accuracy was around 94.74%, slightly lower than the best Random Forest performance.

**Interpretation and Insights**

* **Model Performance Comparison**:
  + **Advantages of Random Forest**:
    - Typically outperforms a single Decision Tree due to ensemble averaging and reduced overfitting.
    - Achieved higher accuracy than the Decision Tree across tested scenarios (n\_estimators).
* **Dataset Suitability**:
  + The high accuracy scores suggest that both Random Forest and Decision Tree models are well-suited for this dataset.
  + Features such as odor, bruises, gill-color, etc., likely play crucial roles in distinguishing between edible and poisonous mushrooms.
* **Recommendations**:
  + **Model Selection**: Given the high accuracy and robustness against overfitting demonstrated by Random Forest, it is generally preferred over a single Decision Tree for this classification task.
  + **Further Analysis**: Feature importance analysis could provide deeper insights into which characteristics of mushrooms are most indicative of their toxicity.

**Conclusion**

The script and its outputs illustrate a structured approach to dataset exploration, model training, and evaluation for mushroom classification. By leveraging Random Forest and Decision Tree classifiers, it highlights the importance of ensemble methods and parameter optimization in achieving high accuracy for binary classification tasks like distinguishing between edible and poisonous mushrooms.

