Artificial Intelligence

Assignment – 03

Name: Muhammad Tayyab

Roll No: BSDSF21M007

Mushroom Classification Analysis

Introduction

This report presents the analysis and comparison of two classification models, the Decision Tree (using the ID3 algorithm) and the Naïve Bayes Classifier, applied to the Mushroom dataset. The dataset consists of various attributes of mushrooms, classified as either edible or poisonous.

Methodology

The dataset was preprocessed to handle missing values and encode categorical features. It was then split into a training set (80%) and a test set (20%). Both classifiers were implemented and evaluated using this split.

1. Decision Tree Classifier (ID3)

The Decision Tree Classifier was implemented using the ID3 algorithm. The 'entropy' criterion was used for the decision nodes. The classifier was trained on the training set and its performance was evaluated on the test set.

Accuracy: 100%

2. Naïve Bayes Classifier

The Naïve Bayes Classifier was implemented using the GaussianNB model. It was also trained on the training set and evaluated on the test set.

Accuracy: 70.95%

Conclusion

The Decision Tree classifier significantly outperformed the Naïve Bayes Classifier on the Mushroom dataset. While the Decision Tree achieved perfect accuracy, the Naïve

Bayes Classifier's performance was considerably lower. This could be due to the different assumptions and methods used by these classifiers.

criterion for building the tree. The model achieved a remarkable accuracy of 100% on the test set, indicating excellent performance in classifying the mushrooms as edible or poisonous.

2. Naïve Bayes Classifier

The Naïve Bayes Classifier was implemented next. Despite being a powerful classifier for many scenarios, it achieved an accuracy of approximately 70.95% on the test set, which is lower compared to the Decision Tree model.

Performance Comparison

In comparing the two models, the Decision Tree classifier significantly outperformed the Naïve Bayes Classifier on the Mushroom dataset. This might be attributed to the Decision Tree's ability to handle the specific features and their interactions more effectively than the assumption of feature independence in Naïve Bayes.

Conclusion

The analysis indicates that the Decision Tree Classifier is more suited for this particular dataset, owing to its higher accuracy and effectiveness in handling complex attribute relationships. However, the choice of classifier should always be based on the specific characteristics and requirements of the dataset.

Python Code

Below is the Python code used for the analysis:

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
from sklearn.naive_bayes import GaussianNB

# Loading the mushroom dataset
dataset_path = '/tayyab/AI/agaricus-lepiota.data'
column_names = ["class", "cap-shape", "cap-surface", "cap-color", "bruises", "odor",
                "gill-attachment", "gill-spacing", "gill-size", "gill-color", "stalk-shape",
                "stalk-root", "stalk-surface-above-ring", "stalk-surface-below-ring",
                "stalk-color-above-ring", "stalk-color-below-ring", "veil-type", "veil-color",
                "ring-number", "ring-type", "spore-print-color", "population", "habitat"]

mushroom_data = pd.read_csv(dataset_path, names=column_names)

# Preprocessing
mushroom_data.replace('?', pd.NA, inplace=True)
mushroom_data.dropna(inplace=True)
```

```
label_encoders = {}
for column in mushroom_data.columns:
    encoder = LabelEncoder()
    mushroom_data[column] = encoder.fit_transform(mushroom_data[column])
    label_encoders[column] = encoder

X = mushroom_data.drop('class', axis=1)
y = mushroom_data['class']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Decision Tree Classifier
dt_classifier = DecisionTreeClassifier(criterion='entropy', 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)

# Naive Bayes Classifier
nb_classifier = GaussianNB()
nb_classifier.fit(X_train, y_train)
y_pred_nb = nb_classifier.predict(X_test)
accuracy_nb = accuracy_score(y_test, y_pred_nb)

# Print accuracies
print("Accuracy of Decision Tree:", accuracy_dt)
print("Accuracy of Naive Bayes:", accuracy_nb)
```