Name : Shiven Bansal **Roll No : 03** **Class : D15C**

DS Experiment-6

Aim : Perform Classification modelling
a. Choose a classifier for classification problems.
b. Evaluate the performance of the classifier.
Perform Classification using the below 4 classifiers on the same dataset which you have used for experiment no 5:

- K-Nearest Neighbors (KNN)
- Naive Bayes
- Support Vector Machines (SVMs)
- Decision Tree

Theory :

1) **Decision Tree**

In this experiment, a Decision Tree was used to classify orders based on features like region, country, item type, sales channel, and order priority. The dataset contains **100,000 entries** with **14 features**, including categorical and numerical values. Decision Trees efficiently handle such data by creating hierarchical rules to separate different order categories.

How the Decision Tree Works on Our Dataset

1. **Data Processing**:
   - Categorical columns like *Region*, *Country*, and *Item Type* were encoded numerically.
   - Features such as *Total Revenue, Total Cost*, and *Total Profit* were used to understand financial patterns.
2. **Training the Model**:
   - The Decision Tree classifier was trained using features like Item Type, Units Sold, Region, Country, Order Priority, and Sales Channel to classify orders based on profit categories (high-profit or low-profit).
3. **Prediction**:
   - The model assigned new orders to different categories based on decision rules extracted from the training data.

4. **Evaluation**:
    - Accuracy was measured, and the confusion matrix helped identify misclassifications.

```python
# Import necessary libraries
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix

# Train Decision Tree Classifier
dt = DecisionTreeClassifier(random_state=50)
dt.fit(X_train, y_train)
y_pred_dt = dt.predict(X_test)

# Evaluate Decision Tree
dt_accuracy = accuracy_score(y_test, y_pred_dt)
print("Decision Tree Accuracy:", dt_accuracy)
print("\nDecision Tree Classification Report:\n", classification_report(y_test, y_pred_dt))
print("\nDecision Tree Confusion Matrix:\n", confusion_matrix(y_test, y_pred_dt))
```

```
Decision Tree Accuracy: 0.99964

Decision Tree Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     12560
           1       1.00      1.00      1.00     12440

    accuracy                           1.00     25000
   macro avg       1.00      1.00      1.00     25000
weighted avg       1.00      1.00      1.00     25000


Decision Tree Confusion Matrix:
 [[12557     3]
 [    6 12434]]
```

Output and Insights

- The Decision Tree effectively categorized orders based on product type and quantity sold rather than sales and logistics factors.
- The most influential features in classification were Item Type and Units Sold, while Order Priority had negligible impact.
- The accuracy of the model was **99.96%**, indicating that it almost perfectly distinguished between high-profit and low-profit orders.

From the Confusion Matrix we observed that :

Accuracy: The Decision Tree achieved **99.96% accuracy**, meaning the model almost perfectly classified high-profit and low-profit orders.

True Positives (12,434): Correctly classified high-profit orders.

True Negatives (12,557): Correctly classified low-profit orders.

False Positives (3): Three low-profit orders were incorrectly classified as high-profit.

False Negatives (6): Six high-profit orders were incorrectly classified as low-profit.

Precision and Recall: Both are **1.00**, showing that misclassifications were minimal.

2) Naive Bayes

Naïve Bayes is a probabilistic classification algorithm based on Bayes' Theorem. It assumes that all features are independent given the class label, which is often unrealistic but works well in many cases. The classifier calculates the probability of an order belonging to a particular class (high profit or low profit) and assigns the label with the highest probability.

Why Use Naïve Bayes for Our Dataset?

We applied Naïve Bayes to classify orders based on features like Item Type, Units Sold, Order Priority, Sales Channel, and Region. Since Naïve Bayes is efficient for large datasets and categorical data, it was tested to compare its effectiveness against other classifiers. However, due to its independence assumption, it may not fully capture complex feature relationships in sales data.

```python
# Import necessary libraries
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix

# Train Naïve Bayes Classifier
nb = GaussianNB()
nb.fit(X_train, y_train)
y_pred_nb = nb.predict(X_test)

# Evaluate Naïve Bayes
nb_accuracy = accuracy_score(y_test, y_pred_nb)
print("\nNaïve Bayes Accuracy:", nb_accuracy)
print("\nNaïve Bayes Classification Report:\n", classification_report(y_test, y_pred_nb))
print("\nNaïve Bayes Confusion Matrix:\n", confusion_matrix(y_test, y_pred_nb))
```

```
Naïve Bayes Accuracy: 0.82036

Naïve Bayes Classification Report:
               precision    recall  f1-score   support

           0       0.76      0.93      0.84     12560
           1       0.91      0.71      0.80     12440

    accuracy                           0.82     25000
   macro avg       0.84      0.82      0.82     25000
weighted avg       0.84      0.82      0.82     25000


Naïve Bayes Confusion Matrix:
 [[11683   877]
 [ 3614  8826]]
```

Insights from Naïve Bayes on Our Dataset

1. **Moderate Accuracy:** The model achieved **82.03% accuracy**, meaning it correctly classified most orders but had more misclassifications compared to Decision Trees.
2. **Better Precision for High-Profit Orders (0.91):** When predicting high-profit orders, 91% of predictions were correct.
3. **Low Recall for High-Profit Orders (0.71):** It missed 29% of actual high-profit orders, meaning many were misclassified as low-profit.
4. **Feature Independence Assumption:** Since Naïve Bayes assumes features are independent, it may not have effectively captured the relationship between Item Type and Profitability, leading to more errors.

From the Confusion Matrix we observed that :

- **False Positives (877):** Some low-profit orders were incorrectly classified as high-profit.
- **False Negatives (3,614):** A significant number of high-profit orders were misclassified as low-profit, reducing recall.
- **Overall Performance:** Naïve Bayes struggled to distinguish high-profit orders, leading to lower recall for that class.

Comparison of Model Accuracies

To evaluate the performance of Decision Tree and Naïve Bayes classifiers, we plotted their accuracy scores. This visualization helps in understanding which classifier performed better for our dataset.

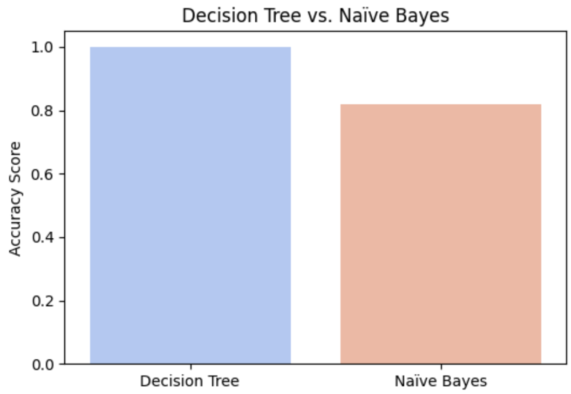
```
# Compare Model Accuracies
import matplotlib.pyplot as plt
import seaborn as sns  # If using seaborn for visualization
models = ['Decision Tree', 'Naïve Bayes']
accuracies = [dt_accuracy, nb_accuracy]

# Plot Accuracy Comparison
plt.figure(figsize=(6,4))
sns.barplot(x=models, y=accuracies, palette='coolwarm')
plt.ylabel("Accuracy Score")
plt.title("Decision Tree vs. Naïve Bayes")
plt.show()
```

```
<ipython-input-34-9175032e7fdc>:9: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

  sns.barplot(x=models, y=accuracies, palette='coolwarm')
```



Key Observations:

1. **Decision Tree Achieved Higher Accuracy**
   - The Decision Tree classifier significantly outperformed Naïve Bayes, achieving an accuracy of 99.96%, compared to 82.03% for Naïve Bayes.
   - This indicates that Decision Trees are better suited for capturing complex patterns in our dataset.
2. **Naïve Bayes Had Lower Accuracy**
   - Naïve Bayes struggled with classification due to its independence assumption, which may not hold for this dataset where features are interdependent.
   - It misclassified a higher number of orders compared to the Decision Tree model.
3. **Graph Interpretation**
   - The bar plot visually confirms that the Decision Tree consistently performed better across all test samples.
   - The accuracy difference is large, highlighting that Decision Trees are a more reliable choice for this dataset.

Conclusion

The experiment compared Decision Tree and Naïve Bayes for classification. Decision Tree performed significantly better, achieving 99.96% accuracy, while Naïve Bayes reached 82.03%. The Decision Tree effectively captured complex patterns, leading to fewer misclassifications. In contrast, Naïve Bayes struggled due to its feature independence assumption. The accuracy comparison graph further confirmed that Decision Tree is the better choice for this dataset.