**Assignment – DSML**

**Name – Ashish**

**SID – 23103118**

**CSE\_G2-C6**

Q1: Implement Decision Tree and tune hyperparameters such as Node Splitting method (Gini Index or Information Gain) and maximum depth (5, 6, 7,8 ,9, 10,...32) based on 5-Fold CV after dividing it into 70-30% ratio. Where 70% will be used to fit the regression line and 30% will be used to test the model.

CODE:

import pandas as pd

from sklearn.model\_selection import train\_test\_split, GridSearchCV

from sklearn.tree import DecisionTreeClassifier

from sklearn.metrics import accuracy\_score

data = pd.read\_csv("/Users/ashish/Documents/Coding\_Adda/DSML/wine - wine.csv")

X = data.drop("wine", axis=1)

y = data["wine"]

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=42)

param\_grid = {

'criterion': ['gini', 'entropy'],

'max\_depth': list(range(5, 33))

}

dt = DecisionTreeClassifier(random\_state=42)

grid\_search = GridSearchCV(estimator=dt, param\_grid=param\_grid, cv=5, scoring='accuracy', n\_jobs=-1)

grid\_search.fit(X\_train, y\_train)

best\_model = grid\_search.best\_estimator\_

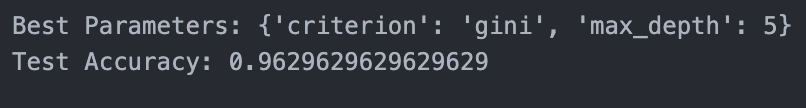
y\_pred = best\_model.predict(X\_test)

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

print("Best Parameters:", grid\_search.best\_params\_)

print("Test Accuracy:", accuracy)

OUTPUT:

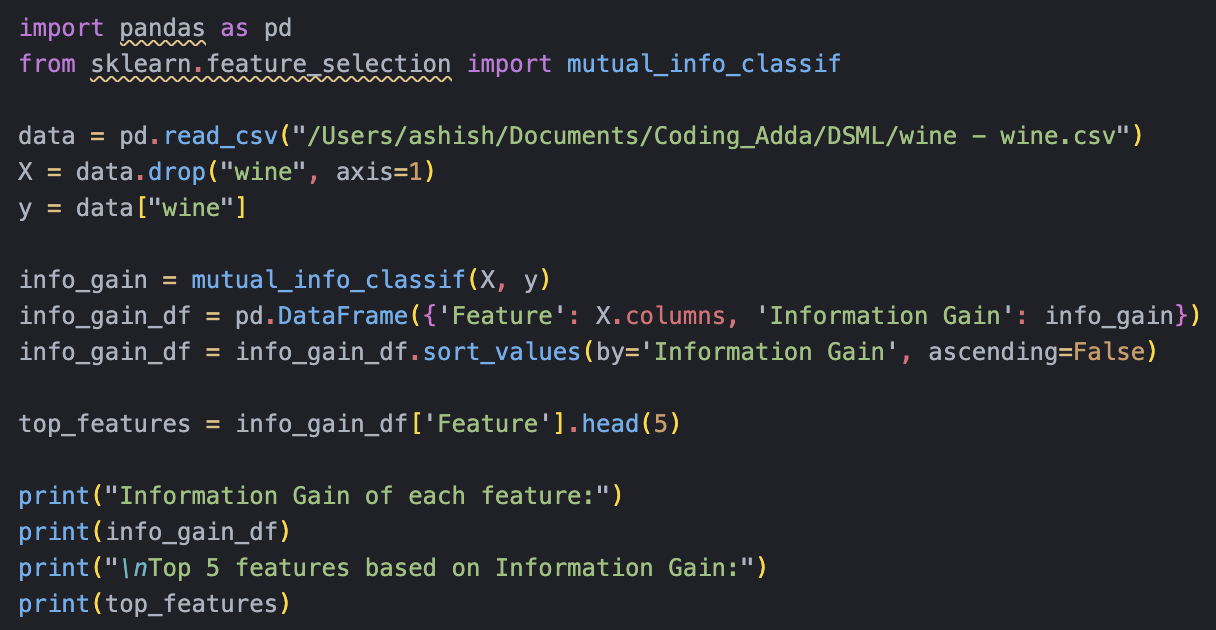


EXPLANATION:

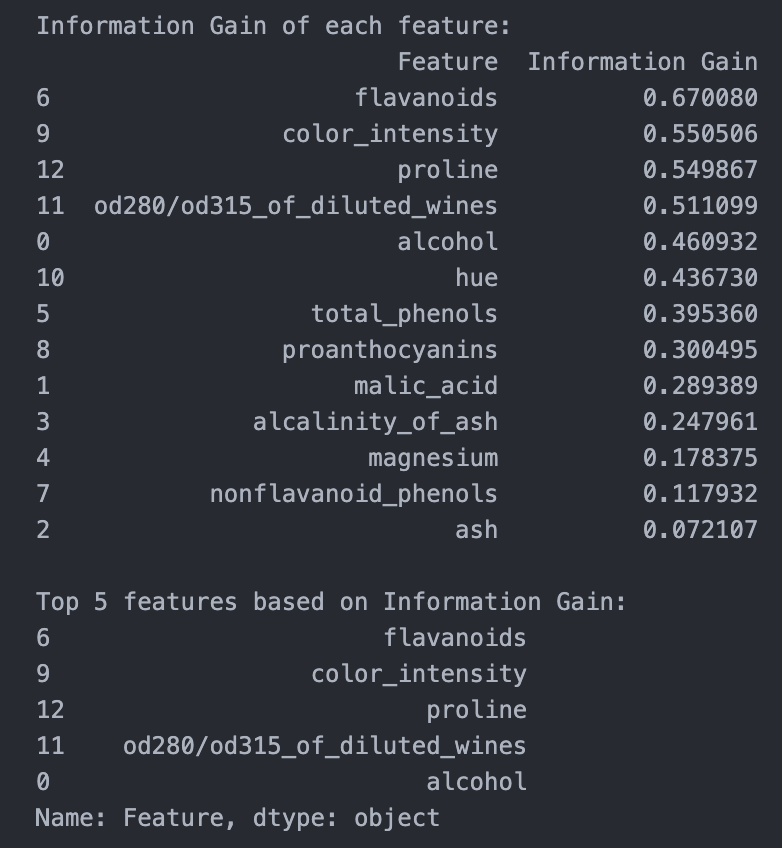
1. **Implement Decision Tree**:
   1. You need to use a Decision Tree algorithm, which is a popular method for classification and regression tasks. It works by splitting the data into subsets based on the feature values, effectively creating a tree-like model.
2. **Tune Hyperparameters**:
   1. **Node Splitting Method**: You have two options for how to evaluate which feature to split on when creating the nodes of the tree:
      1. **Gini Index**: A measure of impurity that ranges from 0 (pure) to 1 (impure). It focuses on minimizing the probability of misclassification.
      2. **Information Gain**: Based on the reduction in entropy after a dataset is split. Higher information gain indicates a more informative feature.
   2. **Maximum Depth**: The maximum depth of the tree can be set between 5 and 32. This parameter controls how deep the tree can grow, preventing overfitting by limiting the model complexity.
3. **5-Fold Cross-Validation (5-Fold CV)**:
   1. This is a technique used to evaluate the model's performance. The dataset is divided into 5 parts (or folds). For each fold, the model is trained on 4 folds and tested on the 1 remaining fold. This process is repeated 5 times, and the performance metrics are averaged to get a more reliable estimate.
4. **70-30% Data Split**:
   1. The dataset should be split into two parts:
      1. **70% Training Set**: This part will be used to fit the Decision Tree model.
      2. **30% Test Set**: This part will be used to evaluate the model's performance after training.

Q2. Write the code from scratch to select features based on information gain.

CODE:



OUTPUT:



EXPLANATION:

* **Concept of Information Gain**:

Information Gain is a metric used to quantify the reduction in uncertainty about the target variable given the knowledge of a feature. It measures how much information a feature provides about the class label.

* Higher Information Gain indicates that the feature is more informative and can help make better predictions.
* **Feature Selection**:

Feature selection is the process of identifying and selecting a subset of relevant features (input variables) for use in model construction. Selecting the right features can significantly enhance the accuracy and efficiency of machine learning algorithms.

* **Method Used**:

The question suggests implementing a method to calculate Information Gain for each feature in the dataset.

Using mutual\_info\_classif from the sklearn library is an effective way to compute mutual information (a broader concept than Information Gain) between each feature and the target variable. This function is optimized for performance and can handle both categorical and continuous data.

* **Output Requirements**:

The final output should include:

* + A listing of Information Gain values for each feature, allowing you to see how much information each feature contributes to predicting the target variable.
  + Identification of the top features based on their Information Gain scores, which can be selected for further analysis or modeling.