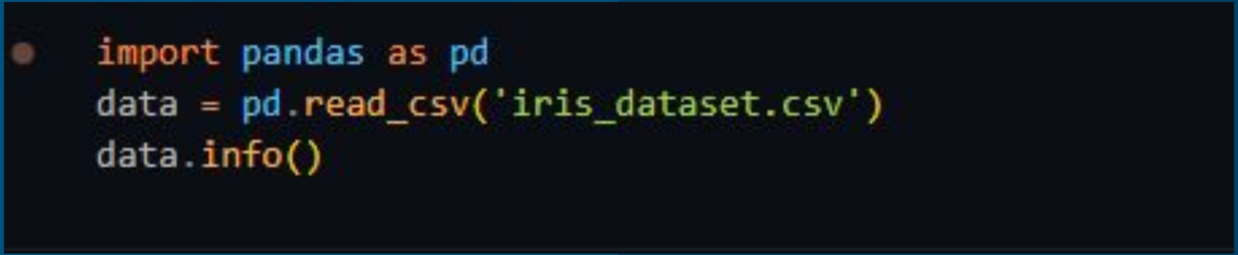
Machine Learning

By EvolkAI

Model Name: MultinomialNB

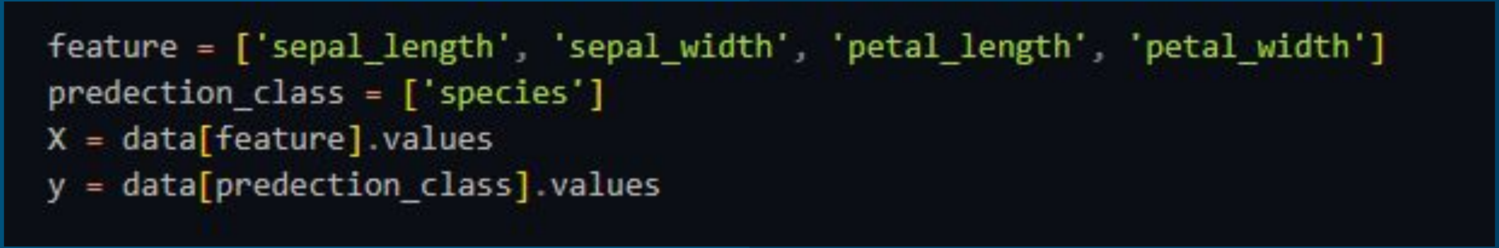
MultinomialNB, short for Multinomial Naive Bayes, is a simple yet effective machine learning algorithm used for text classification tasks. It assumes that the features (words) in the data are independent of each other, given the class labels. It's called "multinomial" because it works well with data that has multiple categories or classes.

The algorithm calculates the probability of each word occurring in a specific class and combines these probabilities to predict the most likely class for a given input. It's commonly used in spam filtering, sentiment analysis, and document categorization tasks. Despite its simplicity, MultinomialNB often produces surprisingly accurate results.



```
import pandas as pd
data = pd.read_csv('iris_dataset.csv')
data.info()
```

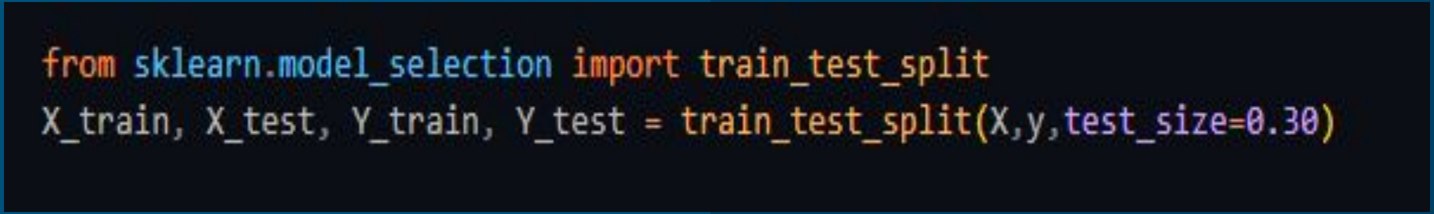
In First line we Import pandas library as pd, then we read iris_dataset.csv file using the read_csv() function, and prints information about the data using the info() method.



```
feature = ['sepal_length', 'sepal_width', 'petal_length', 'petal_width']
predection_class = ['species']
X = data[feature].values
y = data[predection_class].values
```

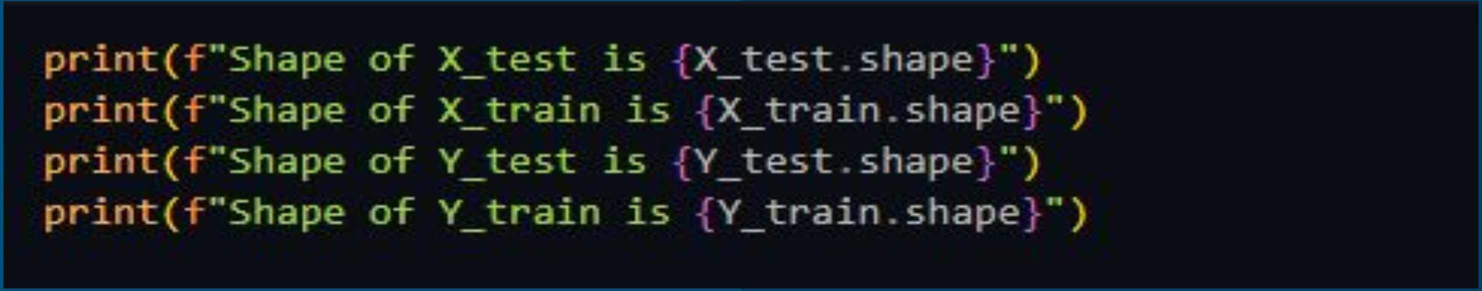
This defines the ‘feature’ and ‘predection_class’ variables, which specify the columns of the data to use as ‘features’ and the column to use as the prediction target.

The code then creates ‘X’ and ‘y’ arrays containing the values of these columns from the data DataFrame.



```
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X,y,test_size=0.30)
```

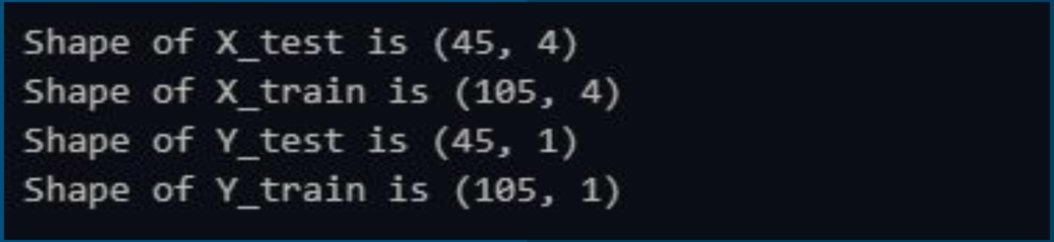
This imports the train_test_split function from the sklearn.model_selection module and uses it to split the data into training and testing sets. The test_size parameter specifies that 30% of the data should be used for testing.



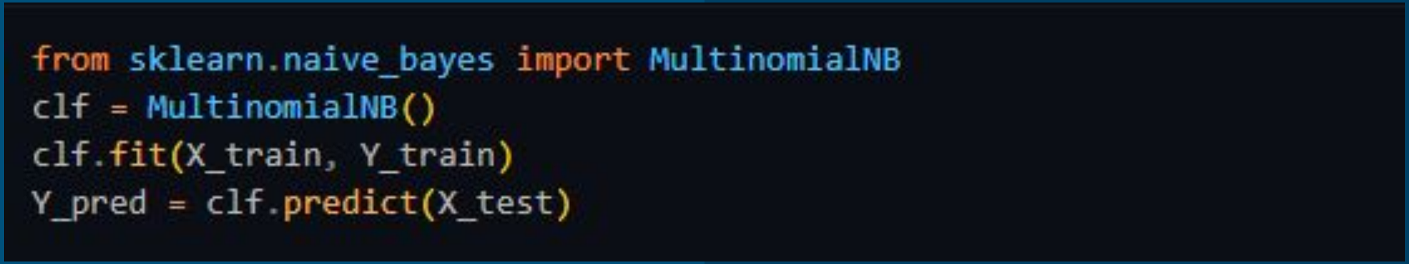
```
print(f"Shape of X_test is {X_test.shape}")
print(f"Shape of X_train is {X_train.shape}")
print(f"Shape of Y_test is {Y_test.shape}")
print(f"Shape of Y_train is {Y_train.shape}")
```

These lines print the shapes of the training and testing data arrays. This output totally depends on the test size we took while train_test_split.

OUTPUT

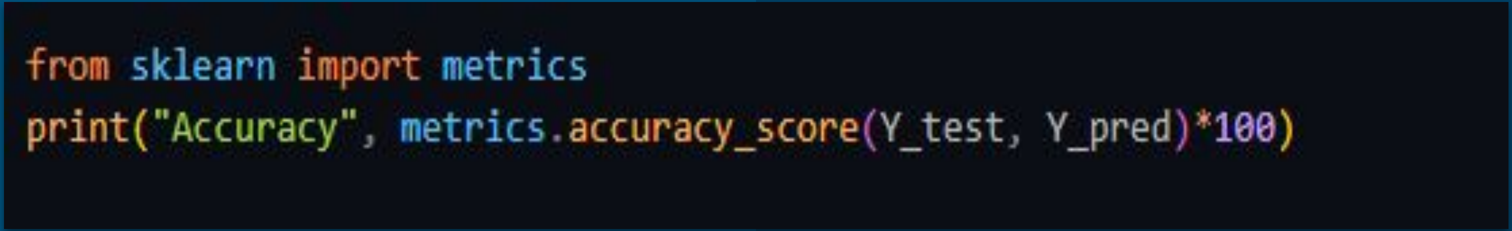


```
Shape of X_test is (45, 4)
Shape of X_train is (105, 4)
Shape of Y_test is (45, 1)
Shape of Y_train is (105, 1)
```



```
from sklearn.naive_bayes import MultinomialNB
clf = MultinomialNB()
clf.fit(X_train, Y_train)
Y_pred = clf.predict(X_test)
```

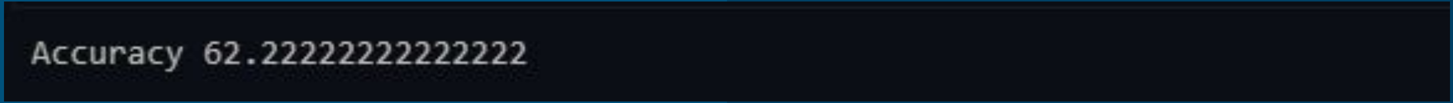
This imports the *MultinomialNB* module from the sklearn tree library, creates a *MultinomialNB* classifier object, fits the classifier with the training data using the *fit()* method, and use the *predict()* method to generate predictions for the testing data.



```
from sklearn import metrics
print("Accuracy", metrics.accuracy_score(Y_test, Y_pred)*100)
```

This imports the metrics module from sklearn and uses the accuracy_score() function to calculate the accuracy of the model on the testing data. The result is printed in the console.

OUTPUT



```
Accuracy 62.2222222222222
```

Conclusion

In conclusion, Multinomial Naive Bayes (NB) is a versatile and widely used algorithm for text classification tasks. Specifically designed for discrete feature data, Multinomial NB assumes a multinomial distribution and calculates the probabilities of data points belonging to different classes. It has proven to be effective in various applications, including document classification, sentiment analysis, and spam filtering.

Multinomial NB is a powerful and widely adopted algorithm for text classification tasks. With its ability to handle discrete features, computational efficiency, and simplicity of implementation, Multinomial NB provides accurate predictions and contributes to the field of text analysis, empowering researchers and practitioners in various domains.

Thank You