



Machine Learning



By EvolkAI



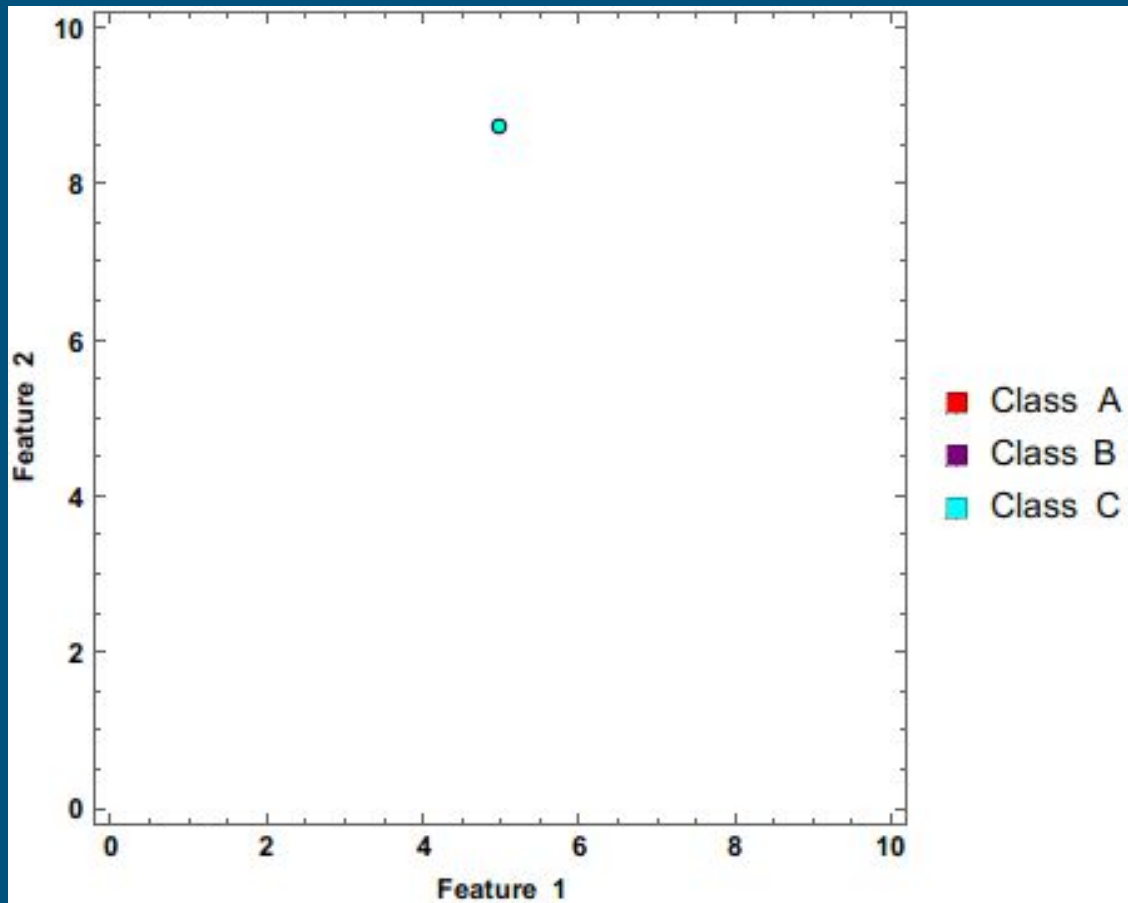


Model Name:
BernoulliNB



Bernoulli Naive Bayes (NB) is a machine learning algorithm used for classification tasks, particularly when dealing with binary or boolean features. It assumes that features follow a Bernoulli distribution, meaning they can take on binary values (0 or 1). Bernoulli NB calculates the probability of a data point belonging to each class based on the presence or absence of features. It assumes that the presence of a feature is important for classification, while the absence is irrelevant.

Bernoulli NB is simple, computationally efficient, and works well with sparse binary data. It is commonly used in text classification tasks, such as sentiment analysis or spam detection. Despite its simplifying assumptions, Bernoulli NB often produces good results and is widely applied in various domains.





```
● 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
```

```
lambda = LogisticRegression()  
lambda.fit(X, y)
```

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)
```

```
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)
```

```
Shape of X_train is (105, 4)
```




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

```
Y_pred = clf.predict(X_test)
```

This imports the *BernoulliNB* module from the sklearn tree library, creates a *BernoulliNB* 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 28.888888888888886
```



Conclusion

In conclusion, Bernoulli Naive Bayes (NB) is a powerful and versatile machine learning algorithm that excels in handling binary feature data. By assuming that the features follow a Bernoulli distribution, Bernoulli NB calculates the probabilities of data points belonging to different classes and assigns them to the class with the highest probability. It is a computationally efficient algorithm that works well with binary features commonly found in text mining and document classification tasks.

Bernoulli NB has demonstrated its effectiveness and reliability in various real-world applications, including sentiment analysis, spam detection, and document categorization. Its simplicity and ease of implementation make it a popular choice for both researchers and practitioners. The algorithm's ability to handle large-scale datasets efficiently and provide accurate predictions contributes significantly to decision-making processes.



Thank You

