

exp5.py

```

1  '''
2  5) Write a program to implement the naïve Bayesian classifier for a sample training data set
   stored as a
3  .CSV file. Compute the accuracy of the classifier, considering few test data sets.
4  '''
5  import pandas as pd
6  import numpy as np
7  from sklearn.model_selection import train_test_split
8  from collections import defaultdict
9
10 class NaiveBayesClassifier:
11     def __init__(self):
12         self.priors = {}
13         self.likelihoods = defaultdict(dict)
14
15     def fit(self, X, y):
16         self.classes = np.unique(y)
17         total_samples = len(y)
18
19         for cls in self.classes:
20             X_cls = X[y == cls]
21             self.priors[cls] = len(X_cls) / total_samples
22             for column in X.columns:
23                 self.likelihoods[column][cls] = X_cls[column].value_counts(normalize=True)
24                 .to_dict()
25
26     def predict(self, X):
27         results = []
28         for i in range(len(X)):
29             posteriors = {}
30             for cls in self.classes:
31                 prior = np.log(self.priors[cls])
32                 likelihood = sum(np.log(self.likelihoods[col].get(cls, {})).get(X.iloc[i][col]
33 , 1e-6)) for col in X.columns)
34                 posteriors[cls] = prior + likelihood
35             results.append(max(posteriors, key=posteriors.get))
36         return results
37
38     def accuracy(self, y_true, y_pred):
39         return np.mean(np.array(y_true) == np.array(y_pred))
40
41 # Load data from CSV file
42 data = pd.read_csv('iris.csv')
43
44 # Separate features and target variable
45 X = data.iloc[:, :-1]
46 y = data.iloc[:, -1]
47
48 # Split data into training and testing sets
49 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
50
51 # Initialize and train the Naive Bayes Classifier
52 nb_classifier = NaiveBayesClassifier()
53 nb_classifier.fit(X_train, y_train)
54
55 # Make predictions on the test set
56 y_pred = nb_classifier.predict(X_test)
57

```

```
56 | # Compute the accuracy
57 | accuracy = nb_classifier.accuracy(y_test, y_pred)
58 | print(f'Accuracy: {accuracy * 100:.2f}%')
59 |
60 | '''OUTPUT
61 | Accuracy: 93.33%
62 | '''
```