Experiment No: 4

Aim:Implementation on Naive Bayesian Classification

Code:

```python
import pandas as pd

# getting data to be trained
file_path = input("Enter file path containing training data: ")

# reading csv data and storing it as panda dataframe dataset =
pd.read_csv(file_path, dtype=str)

print("Training started ")

# extracting columns columns = dataset.columns

# extracting class column class_column = columns[-1]

# extracting non class column non_class_column = columns[: len(columns) - 1]

# extracting class column unique values class_column_unique_attributes =
dataset.loc[:, class_column].unique()

# calculating class column unique values count
class_column_unique_attributes_count = dataset.loc[:,
class_column].value_counts()

# calculating probability of class values class_attributes_probability =
pd.Series()

for class_attribute in class_column_unique_attributes:
    class_attributes_probability[
        class_attribute
        ] = class_column_unique_attributes_count[class_attribute] /
(len(dataset))

# classify data based on class values class_dataset = pd.Series()

for class_value in class_column_unique_attributes:
    class_dataset[class_value] = dataset.loc[dataset[class_column] ==
class_value]
```

```python
# calculating classified dataset coloumns unique values count
class_dataset_columns_unique_count = pd.Series()
for index, dataset in class_dataset.items():
    column_unique_count = pd.Series()
for column in non_class_column:
    column_unique_count[column] = dataset.loc[:, column].value_counts()
    class_dataset_columns_unique_count[index] = column_unique_count

# claculating classified dataset attributes_probability
class_dataset_attributes_probability = pd.Series()

for dataset_index, dataset in class_dataset.items():
    column_unique_probability = pd.Series()
for column in non_class_column:
    column_unique_probability[column] = dataset.loc[:, column].value_counts() /
len(
        dataset
    )

class_dataset_attributes_probability[dataset_index] = column_unique_probability
print("Training completed...")
# printing calculated probabilities
print("######################################################")
print("")
print("--------------TRAINING DATA RESULTS ")
print("")

for dataset_index, dataset in class_dataset.items():
    print("class attribute: ", dataset_index)
print(
    f"'{dataset_index}' attribute count: ",
    class_column_unique_attributes_count[dataset_index],
)
print(
    f"'{dataset_index}' probability: %.3f" %
class_attributes_probability[dataset_index]
)
print("")
for column in non_class_column:
    print("attribute: ", column)
for attr_index, attribute in class_dataset_columns_unique_count[dataset_index][
    column
].items():
```

```python
    print(
        f"'{attr_index} count: ",
        attribute,
        " probability: %.3f"
        %
class_dataset_attributes_probability[dataset_index][column][attr_index],
    )
print("")

print("")
print("####################################################")
# capturing random sample from the user print("")
print("Enter your random sample")
print("")

query = {}
for column in non_class_column:

    data = input(f"{column}: ")
    query[column] = data

# calculating the probabilities (using Bayes Theorem) probabilities = {}

for attribute in class_column_unique_attributes:
    probabilities[attribute] = 1

for dataset_index, dataset in class_dataset.items():
    for column in non_class_column:
        probabilities[dataset_index] = (
            probabilities[dataset_index]
            *
class_dataset_attributes_probability[dataset_index][column][query[column]]
        )

# making predictions
probabilities = pd.Series(probabilities)
maxValue = probabilities.max()
print("\nClassified as ", probabilities[probabilities == maxValue].index[0])
```

Output:
 Enter file path containing training data: play_conditions.csv
 Training started....

Training completed...

##

---------------TRAINING DATA RESULTS------------------

class attribute: no 'no'
attribute count: 5 'no'
probability: 0.357

attribute: outlook

'sunny count: 3 probability: 0.600

'rainy count: 2 probability: 0.400

attribute: temperature

'hot count: 2 probability: 0.400

'mild count: 2 probability: 0.400

'cool count: 1 probability: 0.200

attribute: humidity

'high count: 4 probability: 0.800

'normal count: 1 probability: 0.200

attribute: windy

'strong count: 3 probability: 0.600

'weak count: 2 probability: 0.400

class attribute: yes

'yes' attribute count: 9 'yes'
probability: 0.643

attribute: outlook

'overcast count: 4 probability: 0.444

'rainy count: 3 probability: 0.333

'sunny count: 2 probability: 0.222

attribute: temperature

'mild count: 4 probability: 0.444

'cool count: 3 probability: 0.333

'hot count: 2 probability: 0.222

attribute: humidity

'normal count: 6 probability: 0.667

'high count: 3 probability: 0.333

attribute: windy

'weak count: 6 probability: 0.667

'strong count: 3 probability: 0.333

##

Enter your random sample

outlook: sunny
temperature: cool
humidity: high windy:
strong

Classified as no