Basics concepts of Naive Bayes algorithm

- Naive Bayes is a classification algorithm that works based on the Bayes theorem.
- Bayes theorem is used to find the probability of a hypothesis with given evidence.

$$P(A|B) = \frac{P(B|A) * P(A)}{P(B)}$$

- Bayes theorem helpsus to find the probability of A, given that B occurred.
- A is the hypothesis and B is the evidence.

1. P(B|A) is the probability of B given that A is True.
2. P(A) and P(B) is the independent probabilities of A and B.

ustad g ap ny jo samjhaya us ki samajh to ai lakin equation ma is sy xiyada samjh ni ai

source code used for this assignment

- below is the source code

In [ ]:

```
# Importing the libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

# Importing the dataset
dataset = pd.read_csv('Social_Network_Ads.csv')
X = dataset.iloc[:, [2, 3]].values
y = dataset.iloc[:, -1].values

# Splitting the dataset into the Training set and Test set
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.20, random

# Feature Scaling
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)

# Training the Naive Bayes model on the Training set
from sklearn.naive_bayes import GaussianNB
```

```
classifier = GaussianNB()
classifier.fit(X_train, y_train)

# Predicting the Test set results
y_pred = classifier.predict(X_test)

# Making the Confusion Matrix
from sklearn.metrics import confusion_matrix, accuracy_score
ac = accuracy_score(y_test,y_pred)
cm = confusion_matrix(y_test, y_pred)
```

importing libraries

In [ ]:

```
# importing libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

importing data sets

In [ ]:

```
# Importing the dataset
dataset = pd.read_csv('position_salaries.csv')
dataset.head()
```

Out[ ]:

	ids	gender	age	Salary	purchased
0	1567544	male	19	45000	0
1	1567545	female	45	50000	0
2	1567546	male	24	60000	0
3	1567547	female	28	80000	0
4	1567548	male	33	110000	0

converting string column into values

In [ ]:

```
dataset['gender']=dataset['gender'].replace('male',1)
dataset['gender']=dataset['gender'].replace('female',0)
```

confirmation of convetred columns

In [ ]:

```
dataset.head()
```

Out[ ]:

	ids	gender	age	Salary	purchased
0	1567544	1	19	45000	0
1	1567545	0	45	50000	0
2	1567546	1	24	60000	0
3	1567547	0	28	80000	0
4	1567548	1	33	110000	0

selecting columns for training model

In [ ]:
```
X = dataset.iloc[:, [2, 3]].values
y = dataset.iloc[:, -1].values
```

splitting datasets

In [ ]:
```
# Splitting the dataset into the Training set and Test set
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.20, random
```

do not have any idea about feature scaling/

fitting the model

In [ ]:
```
# Feature Scaling
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

training of Model

In [ ]:
```
# Training the Naive Bayes model on the Training set
from sklearn.naive_bayes import GaussianNB
classifier = GaussianNB()
classifier.fit(X_train, y_train)
```

Out[ ]:
```
GaussianNB()
```

predicting the results

In [ ]:
```
# Predicting the Test set results
y_pred = classifier.predict(X_test)
print(y_pred)
```

```
[0 0]
```

applying confusion matrix

In [ ]:

```
# Making the Confusion Matrix
from sklearn.metrics import confusion_matrix, accuracy_score
ac = accuracy_score(y_test,y_pred)
cm = confusion_matrix(y_test, y_pred)
```

plotting confusion matrix

In [ ]:

```
import seaborn as sns
plt.figure(figsize=(9,9))
sns.heatmap(cm, annot=True, fmt=".3f", linewidth=.5, square=True, cmap='Spectral')
plt.ylabel('Actual outputs');
plt.ylabel('Predicted outputs');
all_sample_title='Accuracy score: {0}'.format(ac)
plt.title(all_sample_title, size=15);
```