Experiment - 06.

(Group A)

- Title of the Assignment: Data Analytics III.

- Problem Statement -
  1. Implement simple Naive Bayes classification algorithm using python/R on iris.csv dataset.
  2. Compute confusion matrix to find TP, FP, TN, FN, Accuracy, error rate, precision, recall on given dataset.

- Objective of the assignment - Students should be able to data analysis using Naive Bayes classification algorithm using python for any open source dataset.

- Prerequisite:
  1. Basic of python
  2. Concept of join and marginal probability.

- THEORY :

1) Concepts used for Naive Bayes classifier.

- Naive Bayes classifier can be used for classification of categorical data.

- Let there be j number of classes $C = \{1, 2 \cdots j\}$
- Let input observation is specified by 'p' features. Therefore, input observation x is given,

$x = \{F1, F2 \cdots Fp\}$.

- The naive Bayes classifier depends on Bayes rule from probability theory.
- Prior probabilities: which are calculated for some event based on no information.
- For eg: P(A), P(B), P(C) are prior probabilities because while calculating P(A), occurrences of B, or C are not concerned i.e. no information about occurence of any other event is used.
- Conditional probabilities -

$$P\left(\frac{A}{B}\right) = \frac{P(A \cap B)}{P(B)} \quad \text{if } P(B) \neq 0$$

$$P\left(\frac{B}{A}\right) = \frac{P(B \cap A)}{P(A)}$$

$$P(A \cap B) = P\left(\frac{A}{B}\right) \cdot P(B) = P\left(\frac{B}{A}\right) \cdot P(A)$$

$$P\left(\frac{A}{B}\right) = \frac{P(B/A) \cdot P(A)}{P(B)}$$

Is called the Bayes rule.

2) Example of Naive Bayes -

We have a dataset with some features outlook, temp, Humidity, windy and the target here is to predict whether a person or team will play tennis or not.

Outlook	Temp	Humidity	Windy	Play
sunny	hot	high	FALSE	no
sunny	hot	high	TRUE	no
rainy	mild	normal	TRUE	yes
overcast	cool	high.	FALSE	yes.
⋮				

$$X = [\underbrace{\text{Outlook}}_{x1}, \underbrace{\text{Temp}}_{x2}, \underbrace{\text{Humidity}}_{x3}, \underbrace{\text{Windy}}_{x4}]$$

$$C_k = [\underbrace{\text{Yes}}_{C1}, \underbrace{\text{No}}_{C2}]$$

- Conditional probability -

- Here we are predicting the probability of class1 and class2 based on given condition If I try to write same formula in terms of classes and features

$$P(C_k \mid X) = \frac{P(X \mid C_k) * P(C_k)}{P(X)}$$

- Now we have two classes and four features, so if we write this formula for class C1.

$$P(C_1 | x_1 \cap x_2 \cap x_3 \cap x_4) = \frac{P(x_1 \cap x_2 \cap x_3 \cap x_4 | C_1) * P(C_1)}{P(x_1 \cap x_2 \cap x_3 \cap x_4)}$$

- The naive Bayes algorithm assumes that all the features are independent of each other or in other words all features are unrelated.
- With that assumption, the equation will be -

$$P(C_1 | x_1 \cap x_2 \cap x_3 \cap x_4) = \frac{P(x_1|C_1) * P(x_2|C_1) * P(x_3|C_1) * P(x_4|C_1) * P(C_1)}{P(x_1) * P(x_2) * P(x_3) * P(x_4)}.$$

- This is the final equation of Naive Bayes and we have to calculate probability of both C_1 and C_2 for this particular example.

Outlook	Temp	Humidity	Windy	Play.
Rainy	Cool	High	True	?

$$P(Yes|x) = P(Rainy|Yes) \times P(Cool|Yes) \times P(High|Yes) \times P(True|Yes) \times P(Yes).$$

$$P(Yes|x) = 2/9 \times 3/9 \times 3/9 \times 3/9 + 9/14 = 0.00529$$

$$0.2 = \frac{0.00529}{0.02057 + 0.00529}$$

$$P(No|x) = P(Rainy|No) \times P(Cool|No) \times P(High|No) \times P(True|No) \times P(No)$$

$$P(No|x) = 3/5 \times 1/5 \times 4/5 \times 3/5 \times 5/14 = 0.02057$$

P(No | Today) > P(Yes | Today). So, the prediction that golf would be played is 'No'.

3] **Stepwise Algorithm for Naive Bayes classification.**

Step 1: Import required libraries. necessary for data handling, visualization, model training. and evaluation.

Step 2: Load the dataset.
data = pd.read_csv("iris.csv").

Step 3: Perform EDA on dataset. Check for null values.

Step 4: Split dataset into dependent and independent variables Y and X.
x = data.drop(['Species'], axis=1)
y = data['Species']

Step 5: Split the dataset into training and testing sets:

xtrain, xtest, ytrain, ytest = train_test_split (x, y, test_size = 0.2, random_state=0).

Step 6: Train the Naive Bayes model. Using gaussian Naive Bayes.

nb = GaussianNB()
nb.fit(xtrain, ytrain).

Step 7: Make predictions.

ytrain-pred = nb.predict (xtrain)
ytest-pred = nb.predict (xtest).

Step 8: Evaluate Model performance.

Accuracy :

accuracy = accuracy-score (ytest, ytest-pred).
error-rate = 1 - accuracy.
cm = confusion-matrix (ytest, ytest-pred)
precision = precision-score (ytest, ytest-pred)
recall = recall-score (ytest, ytest-pred)
report = classification-report (ytest, ytest-pred).
f1 = f1-score (ytest, ytest-pred, average = "weighted").

Step 9: Print all above values and plot if neccessary.

Hence, this simple algorithm helps us to do classification using naive bayes model.

* **Conclusion:** In this way we have done data analysis using Naive Bayes Algorithm for Iris dataset and evaluated performance of the model.

NB

* * * * * * *