*Algorithm : NAIVE BAYES *

Project Title : EMOTION ANALYSIS

- STEP 1 : Importing required libraries[scikit-learn,Vectorizer,Multinomial,pandas]
- STEP 2 : Importing and Reading a downloaded file.View the file(.head())
- STEP 3 : Splitting the dataset [dependent and independent]
- STEP 4 : Vectorizing the dependent value.
- STEP 5 : Fitting a model.
- STEP 6 : Creating an empty Naive-Bayes model.
- STEP 7 : Predicting.

  This Emotion Classification dataset is designed to facilitate research and experimentation in the field of natural language processing and emotion analysis using NAIVE-BAYE'S algorithm.

  Target: Anger,joy,Fear Using Comments.

```
!pip install scikit-learn
from sklearn.datasets import fetch_20newsgroups
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.model_selection import train_test_split
import pandas as pd
```

```
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.10/dist-packages (1.2.2)
Requirement already satisfied: numpy>=1.17.3 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (1.25.2)
Requirement already satisfied: scipy>=1.3.2 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (1.11.4)
Requirement already satisfied: joblib>=1.1.1 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (1.4.2)
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (3.5.0)
```

```
df=pd.read_csv('/content/Emotion_classify_Data.csv')
df.head()
```

	Comment	Emotion
0	i seriously hate one subject to death but now ...	fear
1	im so full of life i feel appalled	anger
2	i sit here to write i start to dig out my feel...	fear
3	ive been really angry with r and i feel like a...	joy
4	i feel suspicious if there is no one outside l...	fear

```
x = df['Emotion']
y = df['Comment']
```

```
vectorizer=TfidfVectorizer(stop_words='english')
x=vectorizer.fit_transform(df['Comment'])
y=df['Emotion']
```

```
clf=MultinomialNB()
clf.fit(x,y)
```

```
▾ MultinomialNB
MultinomialNB()
```

```
text=input("enter the Comment:")
t=[]
t.append(text)
tv=vectorizer.transform(t)
predi=clf.predict(tv)
print(predi)
```

```
enter the Comment:afraid
['fear']
```