# PHASE 05

PROJECT NAME:- SENTIMENT ANALYSIS FOR MARKETING

NAME :- KAIF ALI

NM ID :-E3B31ACF0DF7A773B2E540D1D907827F8

YEAR :- 3RD

-----------------------------------------------------PROGRAM START----------------------------------------------------------

* **AI\_CODE\_PHASE :-05**
* . **Loading the Dataset**:

import pandas as pd

# Load the dataset

dataset = pd.read\_csv('marketing\_data.csv')

* **Text Cleaning**
* import re
* from nltk.tokenize import word\_tokenize
* def clean\_text(text):
* text = re.sub(r'[^A-Za-z\s]', '', text)
* text = text.lower()
* tokens = word\_tokenize(text)
* return ' '.join(tokens)

dataset['Cleaned\_Text'] = dataset['Text'].apply(clean\_text)

* **Stopword Removal**:

from nltk.corpus import stopwords

stop\_words = set(stopwords.words('english'))

def remove\_stopwords(text):

tokens = word\_tokenize(text)

filtered\_tokens = [word for word in tokens if word not in stop\_words]

return ' '.join(filtered\_tokens)

dataset['Cleaned\_Text'] = dataset['Cleaned\_Text'].apply(remove\_stopwords)

* **Sentiment Analysis**

import nltk

from nltk.sentiment.vader import SentimentIntensityAnalyzer

# Initialize the Sentiment Intensity Analyzer

sia = SentimentIntensityAnalyzer()

def analyze\_sentiment(text):

sentiment\_scores = sia.polarity\_scores(text)

compound\_score = sentiment\_scores['compound']

if compound\_score >= 0.05:

return "positive"

elif compound\_score <= -0.05:

return "negative"

else:

return "neutral"

dataset['Sentiment\_VADER'] = dataset['Cleaned\_Text'].apply(analyze\_sentiment)

* **Result Analysis**:
* from sklearn.metrics import accuracy\_score
* import matplotlib.pyplot as plt
* accuracy = accuracy\_score(dataset['Sentiment'], dataset['Sentiment\_VADER'])
* # Visualize sentiment distribution
* sentiment\_counts = dataset['Sentiment\_VADER'].value\_counts()
* sentiment\_counts.plot(kind='bar')
* plt.title('Sentiment Distribution')
* plt.xlabel('Sentiment')
* plt.ylabel('Count')
* plt.show()
* print("Accuracy:", accuracy)