Project to create Sentiment analysis for marketing in python

Certainly! To create a sentiment analysis project for marketing in Python, you can follow these steps using popular libraries such as NLTK and scikit-learn. Make sure to install the necessary libraries using pip install nltk scikit-learn.

```
# Import necessary libraries
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score, classification_report
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk.stem import PorterStemmer
from sklearn.pipeline import make_pipeline

# Download NLTK resources
import nltk
nltk.download('stopwords')
nltk.download('punkt')

# Load your dataset (replace 'your_dataset.csv' with your file)
df = pd.read_csv('your_dataset.csv')
```

```
# Preprocess the text data
stop_words = set(stopwords.words('english'))
ps = PorterStemmer()

def preprocess_text(text):
    words = word_tokenize(text)
    words = [ps.stem(word) for word in words if word.isalpha() and word not in stop_words]
    return ' '.join(words)

df['Processed_Text'] = df['Text_Column'].apply(preprocess_text)

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(df['Processed_Text'], df['Sentiment_Column'], test_size=0.2, random_state=42)

# Create a pipeline with CountVectorizer and Naive Bayes classifier
model = make_pipeline(CountVectorizer(), MultinomialNB())

# Train the model
model.fit(X_train, y_train)

# Make predictions
predictions = model.predict(X_test)

# Evaluate the model
```

```
print(f'Accuracy: {accuracy_score(y_test, predictions)}')

print('Classification Report:')

print(classification_report(y_test, predictions))


# Now, you can use this model to analyze sentiments in your marketing data.
```

Make sure to replace 'your_dataset.csv', 'Text_Column', and 'Sentiment_Column' with your actual dataset file name, text column name, and sentiment column name, respectively. Additionally, customize the preprocessing steps based on your specific requirements.

This example uses a simple Naive Bayes classifier, but you can explore other algorithms and fine-tune parameters based on your dataset and project goals.