**Date: 16 oct 2023**

**Team ID: 329**

**Project ID: Proj\_227277\_Team\_1**

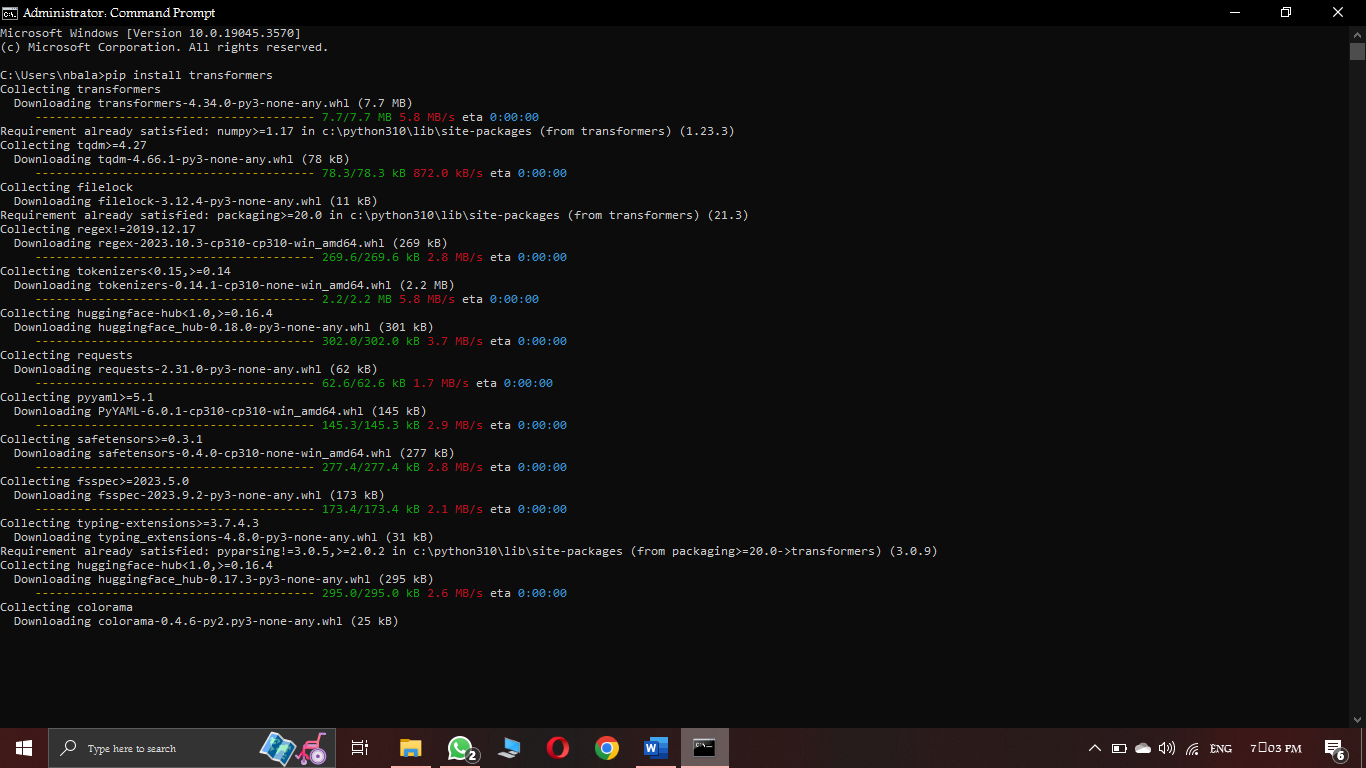
**Name: B. Hemanth**

**Installation of Required Packages**

1. **Package Name: Transformers**

**Use: For Gpt-3 integration**

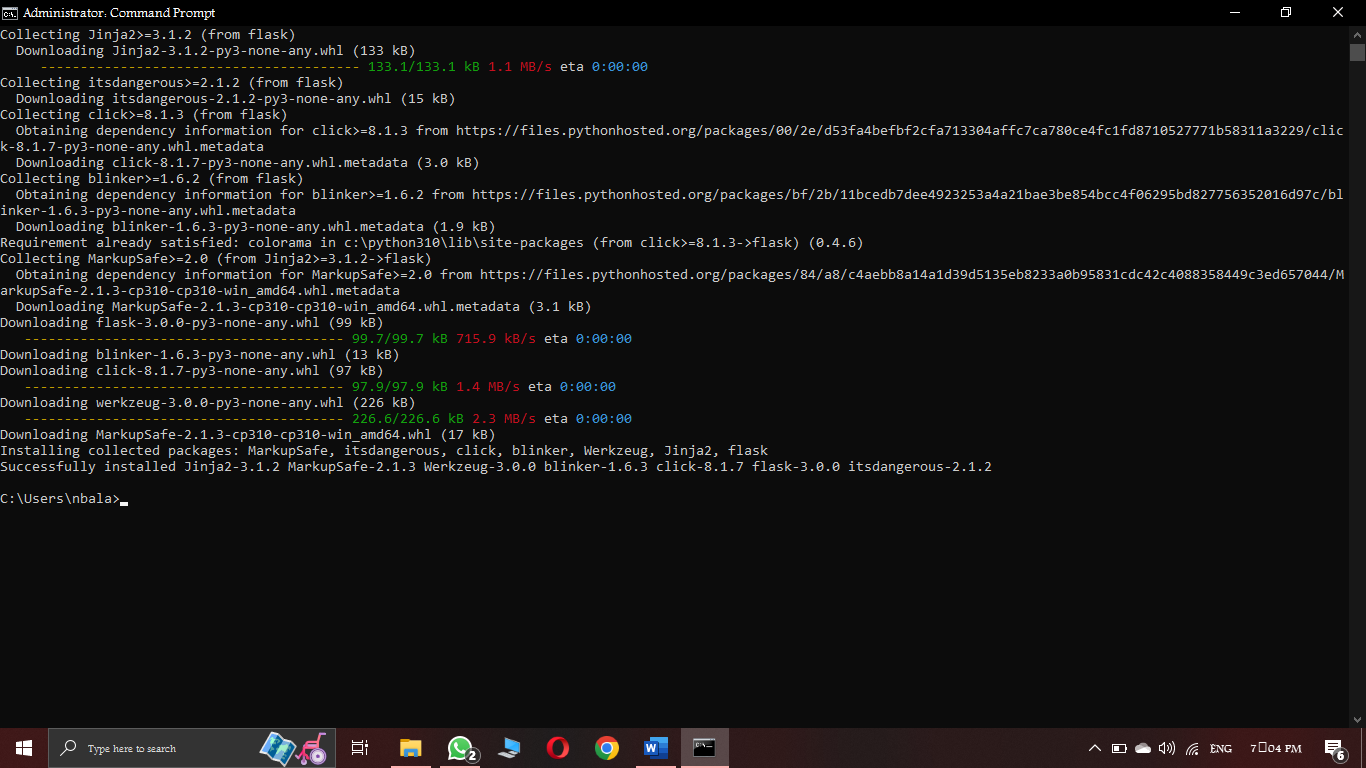
**Command to install: pip install transformers**

****

1. **Package Name: Flask**

**Use: for web app development**

**Command to install: pip install flask**

****

**Program for basic chat bot creation**

I provided source code file in my git hub repository.

# import all required libraries

import numpy as np

import string

from nltk.corpus import stopwords

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.feature\_extraction.text import CountVectorizer

from sklearn.svm import SVC

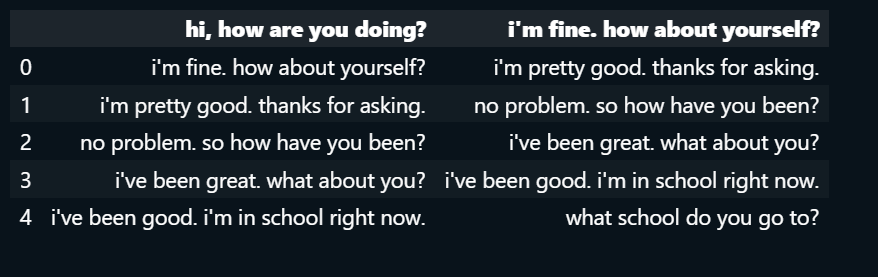
from sklearn.feature\_extraction.text import TfidfTransformer,TfidfVectorizer

from sklearn.pipeline import Pipeline

# importing the dataset

df = pd.read\_csv(r"C:\Users\nbala\Desktop\IBM\Dataset\dialogs.txt", sep='\t')

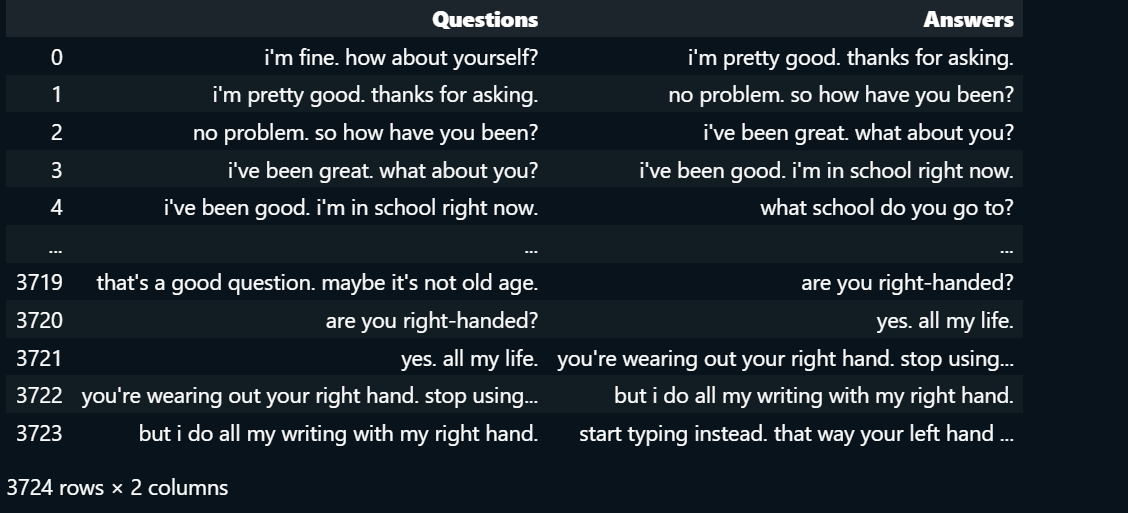
df.head()



#add column names

df.columns=['Questions','Answers']

df



# Data Preprocessing

df['question tokens'] = df['Questions'].apply(lambda x: len(x.split()))

df['answer tokens'] = df['Answers'].apply(lambda x: len(x.split()))

# Create different types of plots

plt.style.use('seaborn-darkgrid')

fig, ax = plt.subplots(nrows=2, ncols=2, figsize=(15, 10))

# Histograms

sns.histplot(x='question tokens', data=df, kde=True, ax=ax[0, 0], color='skyblue')

sns.histplot(x='answer tokens', data=df, kde=True, ax=ax[0, 1], color='salmon')

# KDE Plots

sns.kdeplot(x='question tokens', data=df, ax=ax[1, 0], color='skyblue')

sns.kdeplot(x='answer tokens', data=df, ax=ax[1, 1], color='salmon')

# Titles and labels

ax[0, 0].set\_title('Question Tokens Histogram')

ax[0, 1].set\_title('Answer Tokens Histogram')

ax[1, 0].set\_title('Question Tokens KDE')

ax[1, 1].set\_title('Answer Tokens KDE')

# Adding labels

ax[0, 0].set\_xlabel('Tokens')

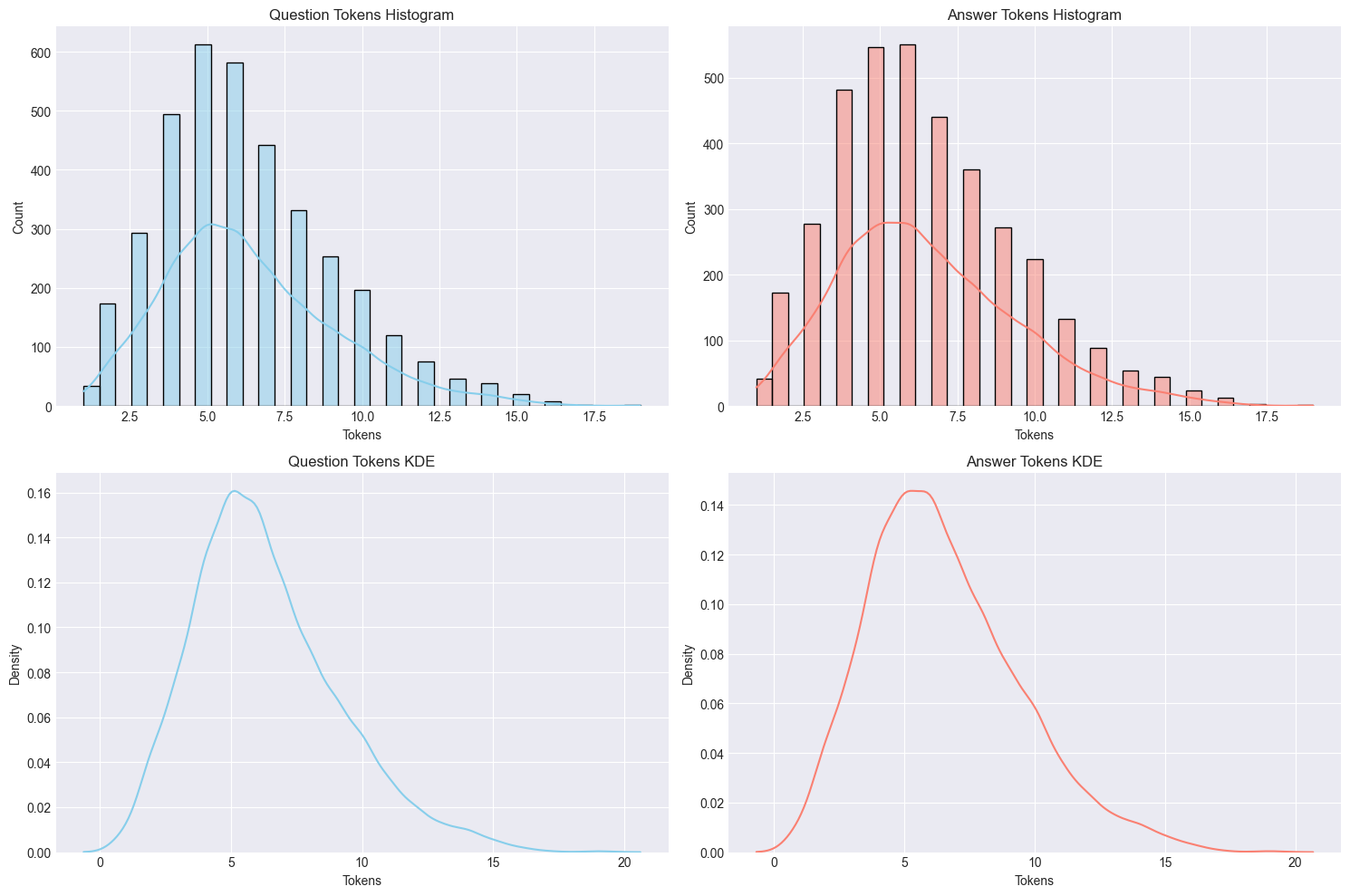
ax[0, 1].set\_xlabel('Tokens')

ax[1, 0].set\_xlabel('Tokens')

ax[1, 1].set\_xlabel('Tokens')

plt.tight\_layout()

plt.show()



#Function for converting upper to lower case

def cleaner(x):

return [a for a in (''.join([a for a in x if a not in string.punctuation])).lower().split()]

Pipe = Pipeline([

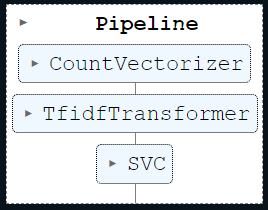
('bow',CountVectorizer(analyzer=cleaner)),

('tfidf',TfidfTransformer()),

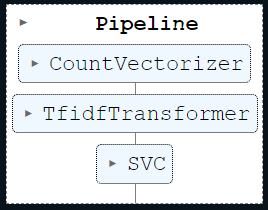
('classifier',SVC(kernel=’linear’))

])

Pipe.fit(df['Questions'],df['Answers'])



Pipe.fit(df['Questions'],df['Answers'])



Pipe.predict(["how's it going?"])[0]



Pipe.predict(["i'm fine. how about yourself?"])[0]



Pipe.predict(["i'm actually in school right now."])[0]



Pipe.predict(["which school do you attend?"])[0]



Pipe.predict(["great"])[0]



Pipe.predict(["it wouldn't rain in the middle of the summer."])[0]

