**Documentation for Sentiment Analysis Streamlit Application**

**Overview**

This Streamlit application performs sentiment analysis on user-provided text using a Hugging Face Transformers model. It also records the user's name and the timestamp when they first use the app in a MySQL database.

**Dependencies**

Ensure you have the following Python packages installed:

* streamlit
* transformers
* scipy
* numpy
* mysql-connector-python

You can install them using pip:

bash

Copy code

pip install streamlit transformers scipy numpy mysql-connector-python

**Database Setup**

The application uses a MySQL database to store user information. Configure the MySQL connection with the following parameters:

* host: "your host"
* port: your port
* user: "username"
* password: ""
* database: ""

Ensure the user\_data table exists with the following schema:

Sql query

CREATE TABLE user\_data (

name VARCHAR(255) NOT NULL,

timestamp DATETIME NOT NULL

);

**Application Structure**

**Imports**

The necessary libraries and modules are imported at the beginning of the script:

import streamlit as st

import mysql.connector

from transformers import AutoTokenizer, AutoConfig, AutoModelForSequenceClassification

from scipy.special import softmax

import numpy as np

from datetime import datetime

**Database Connection**

A connection to the MySQL database is established:

python

Copy code

connection = mysql.connector.connect(

* host= "your host"
* port= your port
* user= "username"
* password=""
* database= ""

)

mycursor = connection.cursor(buffered=True)

**Text Preprocessing**

A function to preprocess text by replacing user mentions and URLs:

python

Copy code

def preprocess(text):

new\_text = []

for t in text.split(" "):

t = '@user' if t.startswith('@') and len(t) > 1 else t

t = 'http' if t.startswith('http') else t

new\_text.append(t)

return " ".join(new\_text)

**Model Loading**

Loading the sentiment analysis model and tokenizer:

python

Copy code

MODEL = "cardiffnlp/twitter-roberta-base-sentiment-latest"

tokenizer = AutoTokenizer.from\_pretrained(MODEL)

config = AutoConfig.from\_pretrained(MODEL)

model = AutoModelForSequenceClassification.from\_pretrained(MODEL)

**Sentiment Prediction**

A function to predict sentiment from the provided text:

python

def predict(text):

preprocessed\_text = preprocess(text)

encoded\_input = tokenizer(preprocessed\_text, return\_tensors='pt')

output = model(\*\*encoded\_input)

scores = output[0][0].detach().numpy()

scores = softmax(scores)

ranking = np.argsort(scores)

ranking = ranking[::-1]

output\_text = ""

for i in range(scores.shape[0]):

label = config.id2label[ranking[i]]

score = np.round(float(scores[ranking[i]]), 4)

output\_text += f"{i+1}) {label}: {score}\n"

return output\_text

**Main Application**

The main function checks if the user’s name is stored in the session state and navigates to the appropriate page:

python

Copy code

def main():

if 'name' not in st.session\_state:

st.session\_state['name'] = None

if st.session\_state['name'] is None:

name\_page()

else:

sentiment\_analysis\_page()

**Name Page**

A page to collect the user's name and store it in the database:

def name\_page():

st.title("Welcome!")

st.write("Please enter your name:")

name = st.text\_input("Name")

if st.button("Submit Name"):

if name:

timestamp = datetime.now()

query = "INSERT INTO user\_data (name, timestamp) VALUES (%s, %s)"

mycursor.execute(query, (name, timestamp))

connection.commit()

st.session\_state['name'] = name

st.experimental\_rerun()

else:

st.write("Please enter a valid name.")

**Sentiment Analysis Page**

A page for performing sentiment analysis on user-provided text:

def sentiment\_analysis\_page():

st.title("Sentiment Analysis with Hugging Face Transformers")

st.write(f"Hello, {st.session\_state['name']}!")

st.write("Enter some text and see the predicted sentiment with confidence scores.")

user\_input = st.text\_area("Enter your text here:")

if st.button("Analyze"):

if user\_input:

result = predict(user\_input)

st.text(result)

else:

st.write("Please enter some text for analysis.")

**Running the Application**

Run the Streamlit application by executing the following command in your terminal:

bash

Copy code

streamlit run app.py

Replace app.py with the filename of your script.

**Conclusion**

This Streamlit application allows users to perform sentiment analysis on text input using a pre-trained model from Hugging Face. It also stores user information in a MySQL database. The application consists of two main pages: one for collecting user names and another for performing sentiment analysis.