Phase 4: Model Deployment and Interface Development

4.1 Overview of Model Deployment and Interface Development

Phase 4 focuses on deploying the trained model to a cloud platform for real-time use and developing an interactive interface for end-users. The goal is to make the market segmentation model accessible and usable in a production environment by exposing it via APIs and creating an intuitive user interface. This phase ensures that stakeholders can interact with the model, make predictions, and visualize segmentation results in real time.

4.2 Deploying the Model

To deploy the trained model, we use cloud platforms like AWS, Google Cloud, or Azure. These platforms provide robust infrastructure and services that make it easy to host machine learning models, expose them via APIs, and ensure scalability for real-time use.

The process involves the following steps:

1. **Model Export:** The trained machine learning model (including the autoencoder and K-Means clustering) is saved and exported. This can be done using TensorFlow's model.save() function.

Source Code: # Save the trained autoencoder model

2. autoencoder.save("autoencoder_model.h5")
   Creating an API for the Model: We use frameworks like Flask or FastAPI to expose the trained model via a RESTful API. This API accepts input data, makes predictions using the model, and returns the results.

Source Code:

```
from flask import Flask, request, jsonify

import numpy as np

import tensorflow as tf

from sklearn.preprocessing import StandardScaler


app = Flask(__name__)


# Load the trained model

model = tf.keras.models.load_model("autoencoder_model.h5")

scaler = StandardScaler()


@app.route('/predict', methods=['POST'])

def predict():

    data = request.get_json()
```

input_data = np.array(data['input'])

scaled_input = scaler.transform(input_data)

encoded_data = model.predict(scaled_input)

return jsonify({'encoded_data': encoded_data.tolist()})

if __name___== '__main__':

3. app.run(debug=True)

Deploying the API on Cloud: Once the API is developed, it can be deployed to a cloud platform like AWS Lambda, Google Cloud Functions, or Azure Functions.

- **AWS:** Using AWS Lambda, the Flask app can be containerized and deployed as a serverless function with AWS API Gateway.
- **Google Cloud:** Using Google Cloud Functions or Google App Engine, the Flask app is deployed as a serverless API.
- **Azure:** Azure Functions can be used to deploy the model as an API, managed by Azure API Management.

4.3 Developing the Web Interface

To allow end-users to interact with the deployed model, we develop a simple web interface. This interface accepts user inputs, sends the data to the model API, and displays the segmentation results. Frameworks like Flask, Streamlit, or React are used for this purpose.

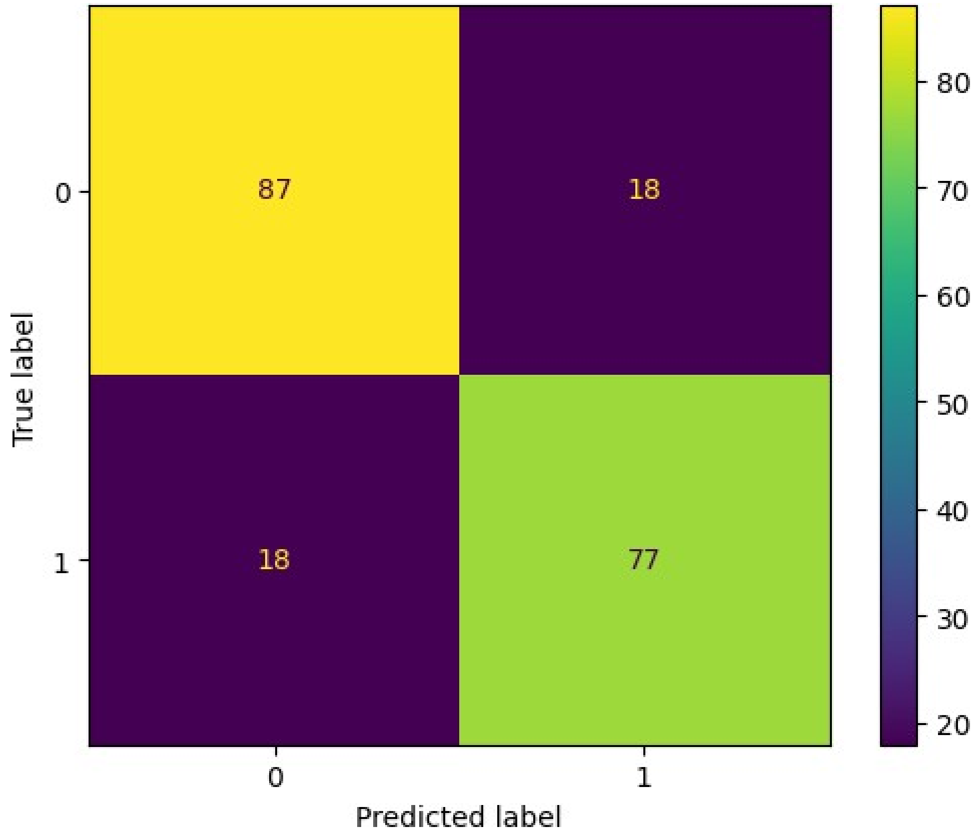
Using Streamlit:

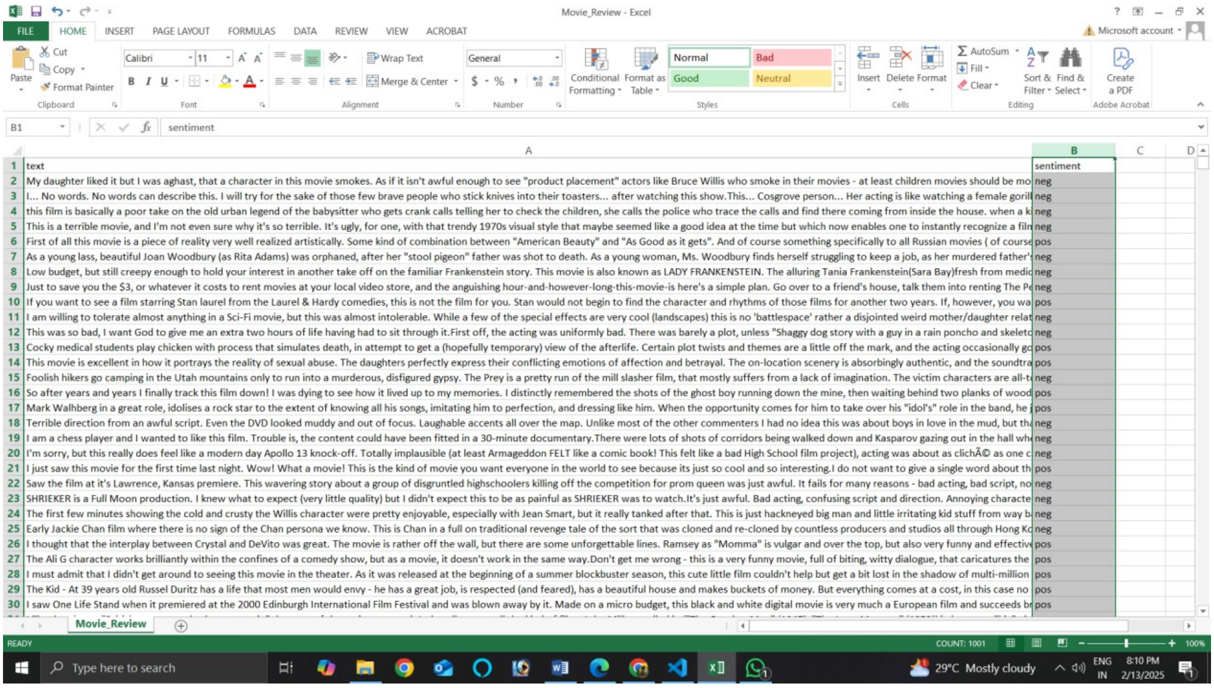
```
import pandas as pd
import pickle as pk
from sklearn.feature_extraction.text import TfidfVectorizer
import streamlit as st

model = pk.load(open('model.pkl','rb'))
scaler = pk.load(open('scaler.pkl','rb'))
review = st.text_input('Enter Movie Review')

if st.button('Predict'):
    review_scale = scaler.transform([review]).toarray()
    result = model.predict(review_scale)
    if result[0] == 0:
        st.write('Negative Review')
    else:
        st.write('Positive Review')
```

4.4 Cloud Platform Considerations

When deploying the model and interface on cloud platforms, the following factors are considered:

- **Scalability:** Cloud platforms provide auto-scaling features to handle high traffic in real-time.
- **Security:** APIs are secured using authentication mechanisms like API keys or OAuth.
- **Monitoring:** Tools like AWS CloudWatch, Google Stackdriver, or Azure Monitor are used to track API performance.
- **Cost Management:** Cloud platforms charge based on usage, so resource optimization is crucial.





4.5 Conclusion of Phase 4

In this phase, we successfully deployed the Advanced Market Segmentation model to a cloud platform, exposed it via an API for real-time use, and developed an interactive web interface using Streamlit. The deployment ensures that the model is accessible for real-time predictions, and the user interface allows easy interaction with the model for segmenting customers based on their attributes. By leveraging cloud platforms and modern web technologies, the project is now fully deployed and ready for use in production environments.

4.6 model deployment in the github

https://github.com/Atmanand18/internship-project.git