```
#!/usr/bin/env python
# coding: utf-8

# In[12]:


import os
from flask import Flask, jsonify,
request
from flask_restful import Api, Resource
import joblib

app = Flask(__name__)
api =
Api(app)

model = joblib.load('LR_model.model')
class MakePrediction(Resource):

@staticmethod
    def post():
        posted_data = request.get_json()
        Gender =
posted_data['Gender']
        Customer_Type = posted_data['Customer_Type']
        Age =
posted_data['Age']
        Type_of_Travel = posted_data['Type_of_Travel']
        Class =
posted_data['Class']
        Flight_Distance = posted_data['Flight_Distance']

Inflight_wifi_service = posted_data['Inflight_wifi_service']

Departure_Arrival_time_convenient = posted_data['Departure_Arrival_time_convenient']

Ease_of_Online_booking = posted_data['Ease_of_Online_booking']
        Gate_location =
posted_data['Gate_location']
        Food_and_drink = posted_data['Food_and_drink']

Online_boarding = posted_data['Online_boarding']
        Seat_comfort =
posted_data['Seat_comfort']
        Inflight_entertainment =
posted_data['Inflight_entertainment']
        On_board_service =
posted_data['On_board_service']
        Leg_room_service = posted_data['Leg_room_service']

   Baggage_handling = posted_data['Baggage_handling']
        Checkin_service =
posted_data['Checkin_service']
        Inflight_service = posted_data['Inflight_service']

  Cleanliness = posted_data['Cleanliness']
        Departure_Delay_in_Minutes =
posted_data['Departure_Delay_in_Minutes']
        Arrival_Delay_in_Minutes =
posted_data['Departure_Delay_in_Minutes']

        prediction =
model.predict([[Gender,Customer_Type,Age,Type_of_Travel,Class,Flight_Distance,Inflight_wifi_ser
vice,

Departure_Arrival_time_convenient,Ease_of_Online_booking,Gate_location,

            Food_and_drink,Online_boarding,Seat_comfort,Inflight_entertainment,

                    On_board_service,Leg_room_service,Baggage_handling,

            Checkin_service,Inflight_service,Cleanliness,

Departure_Delay_in_Minutes,Arrival_Delay_in_Minutes]])[0]
```

```
        if prediction == 0:

  predicted_class = 'neutral or dissatisfied'
        else:
            predicted_class =
'satisfied'

        return jsonify({
            'Prediction': predicted_class

    })


api.add_resource(MakePrediction, '/predict')

if __name__ == '__main__':

app.run(debug=True)



# In[ ]:




# In[ ]:
```