**NAME OF THE COURSE : ARTIFICIAL INTELLIGENCE**

**NAME OF THE PROJECT : SENTIMENT ANALYSIS FOR MARKETING**

**PHASE - 1**

**PROBLEM STATEMENT:**

 The problem is to understand and predict customer sentiment using machine learning techniques. The objective is to develop a model that accurately predicts how customers feel about a product, service, or brand based on a set of data such as social media posts, online reviews, customer service tickets, and other relevant sources. This project involves data preprocessing, feature engineering, model selection, training, and evaluation.

**WHAT I UNDERSTAND :**

1. **Data Collection**: Gathering relevant data is the first step. This dataset should contain information about houses, such as their location, size, number of bedrooms and bathrooms, as any other relevant attributes. Additionally, historical sales data can be useful.
2. **Data** **Preprocessing**: Raw data usually needs cleaning and formatting. This involves handling missing values, removing duplicates, and encoding categorical variables. Data outliers may also need to be addressed.
3. Feature Engineering: Creating new features or transforming existing ones can enhance the model's predictive power. For example, you can calculate the price per square foot, create a feature indicating the age of the house, or consider proximity to amenities like schools and parks.
4. Data Splitting: Split the data into two sets: a training set used to train the model and a test set used to evaluate its performance. A common split is 80% for training and 20% for testing.
5. Model Selection: Choose an appropriate machine learning algorithm. Regression models like Linear Regression, Decision Trees, Random Forests, or Gradient Boosting are commonly used for predicting house prices.
6. Model Training: Train the selected model using the training data. The model learns the relationship between the features and the target variable (house prices) during this phase.
7. Model Evaluation: Evaluate the model's performance on the test data using metrics like Mean Absolute Error (MAE), Mean Squared Error (MSE), or Root Mean Squared Error (RMSE). These metrics quantify how well the model's predictions align with the actual prices.
8. Hyperparameter Tuning: Adjust the model's hyperparameters to optimize its performance. Techniques like cross-validation can help fine-tune the model.
9. Deployment: Once satisfied with the model's performance, it can be deployed for making real predictions. Users can input house features, and the model will estimate the price.
10. Monitoring and Maintenance: Continuously monitor the model's performance in a production environment and retrain it periodically with new data to ensure it remains accurate over time.

**BASIC SOURCE CODE :**

# Import necessary libraries

import pandas as pd

import numpy as np

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LogisticRegression # Example algorithm

# Step 1: Data Collection

# Load the dataset containing customer sentiment data

data = pd.read\_csv('sentiment\_data.csv')

# Step 2: Data Preprocessing

# Clean and format the data

# Handle missing values, duplicates, and encode categorical variables

# Step 3: Feature Engineering

# Create new features or transform existing ones

# For example, you can create a feature that counts the number of positive or negative words in a customer's feedback, or a feature that identifies the topic of the feedback

# Step 4: Data Splitting

# Split the data into training and testing sets

X = data[['Feature1', 'Feature2', 'Feature3']] # Features

y = data['Sentiment'] # Target variable

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Step 5: Model Selection

# Choose a machine learning algorithm (e.g., Logistic Regression)

model = LogisticRegression()

# Step 6: Model Training

model.fit(X\_train, y\_train)

# Step 7: Model Evaluation

# Evaluate the model's performance on the test data

y\_pred = model.predict(X\_test)

# Calculate metrics like accuracy, precision, and recall

# Step 8: Hyperparameter Tuning (if needed)

# Fine-tune the model's hyperparameters using techniques like cross-validation

# Step 9: Deployment (Optional)

# Deploy the model for making real predictions in a production environment

# Step 10: Monitoring and Maintenance (Continuous)

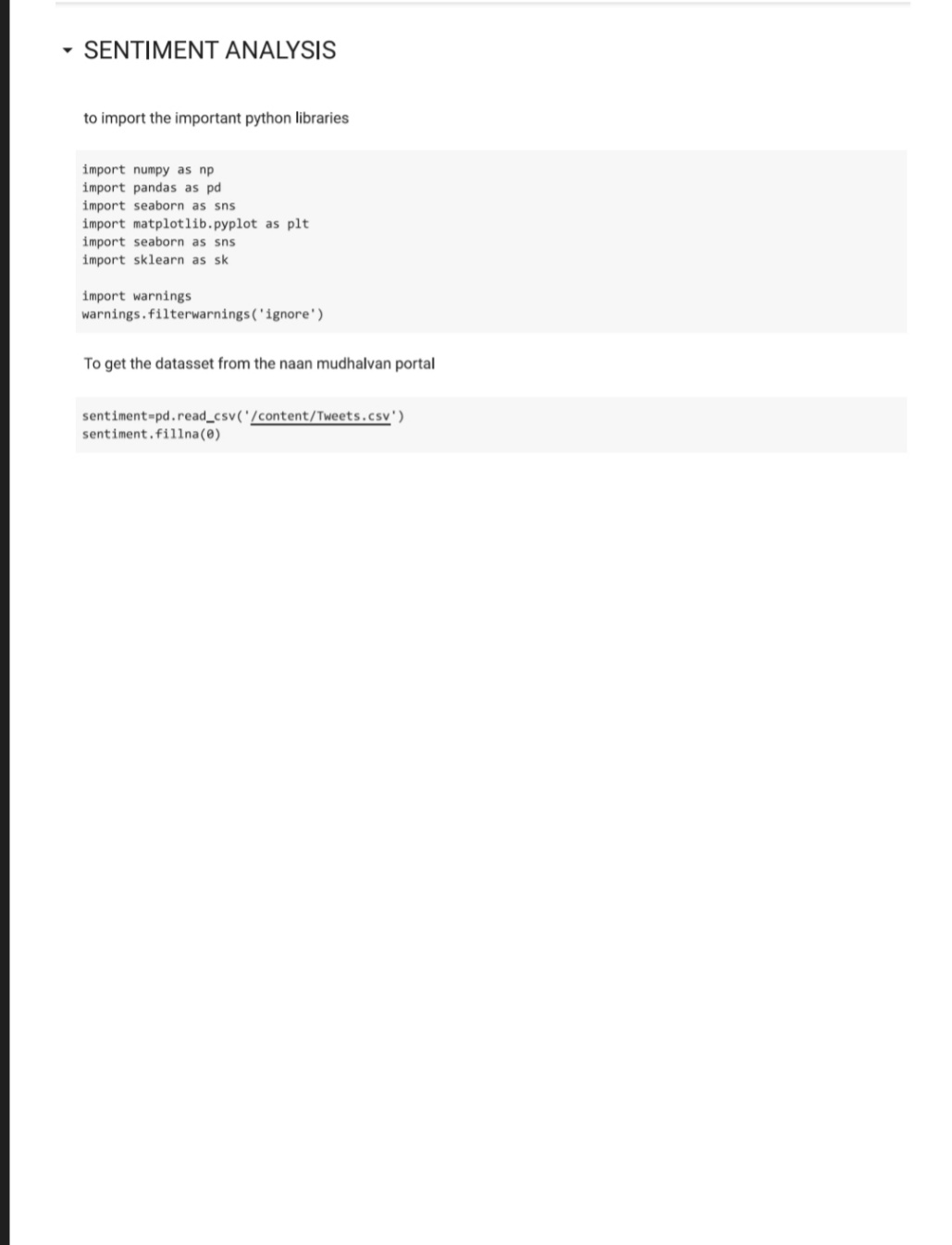
# Continuously monitor the model's performance and retrain it with new data

# Predicting customer sentiment for new input data

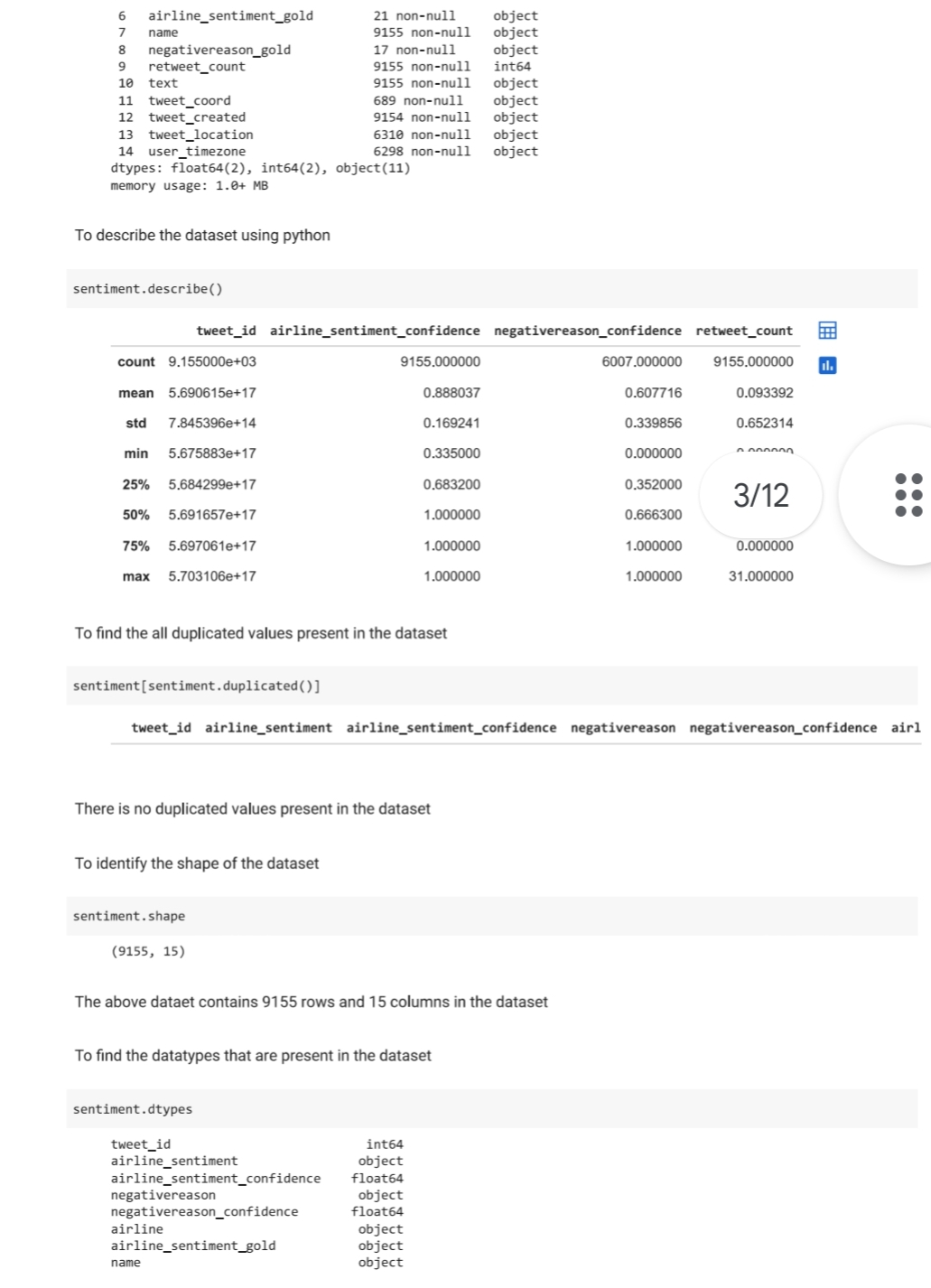
new\_data = pd.DataFrame({'Feature1': ['New\_Value1'], 'Feature2': ['New\_Value2'], 'Feature3': ['New\_Value3']})

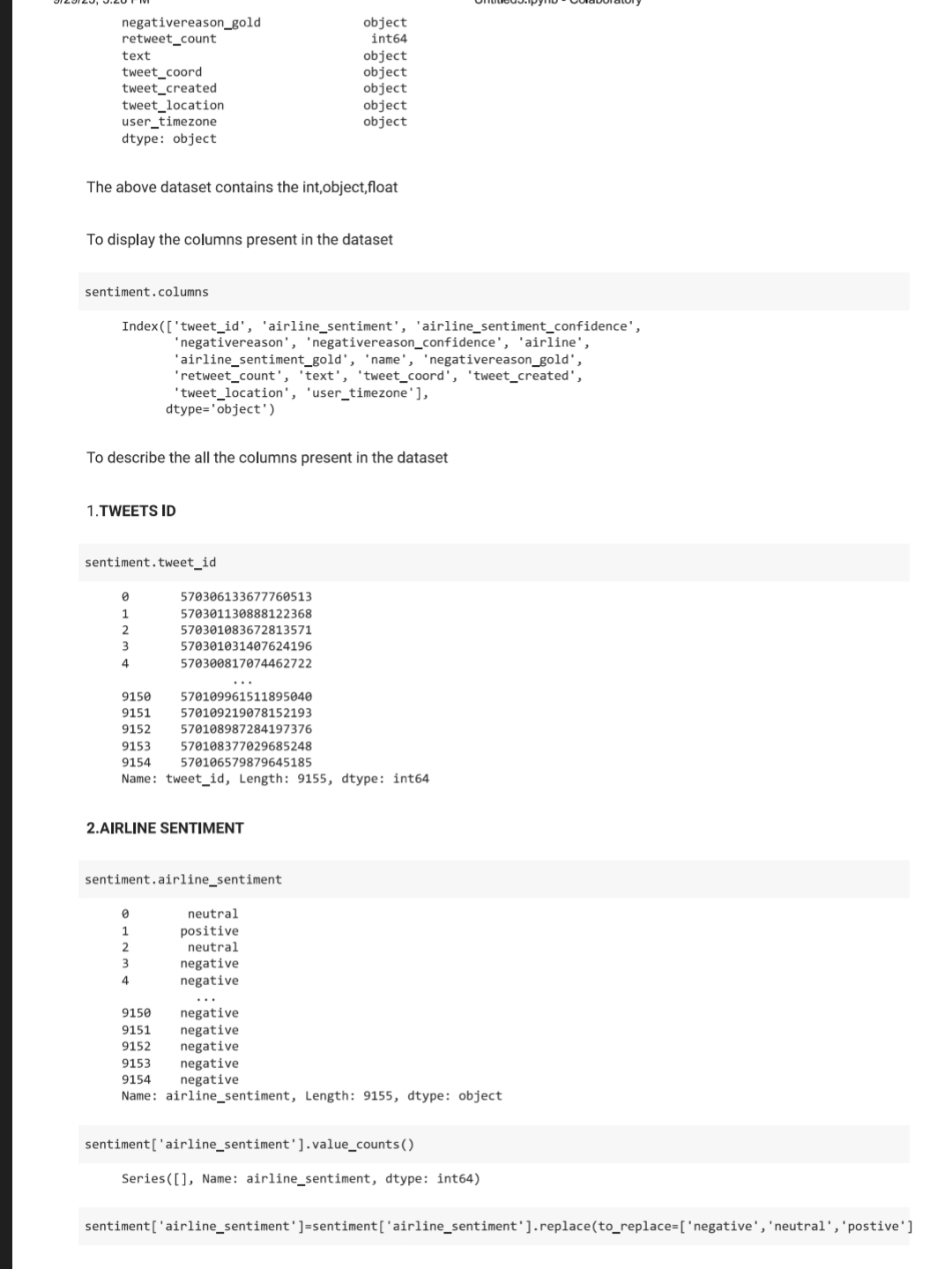
predicted\_sentiment = model.predict(new\_data)

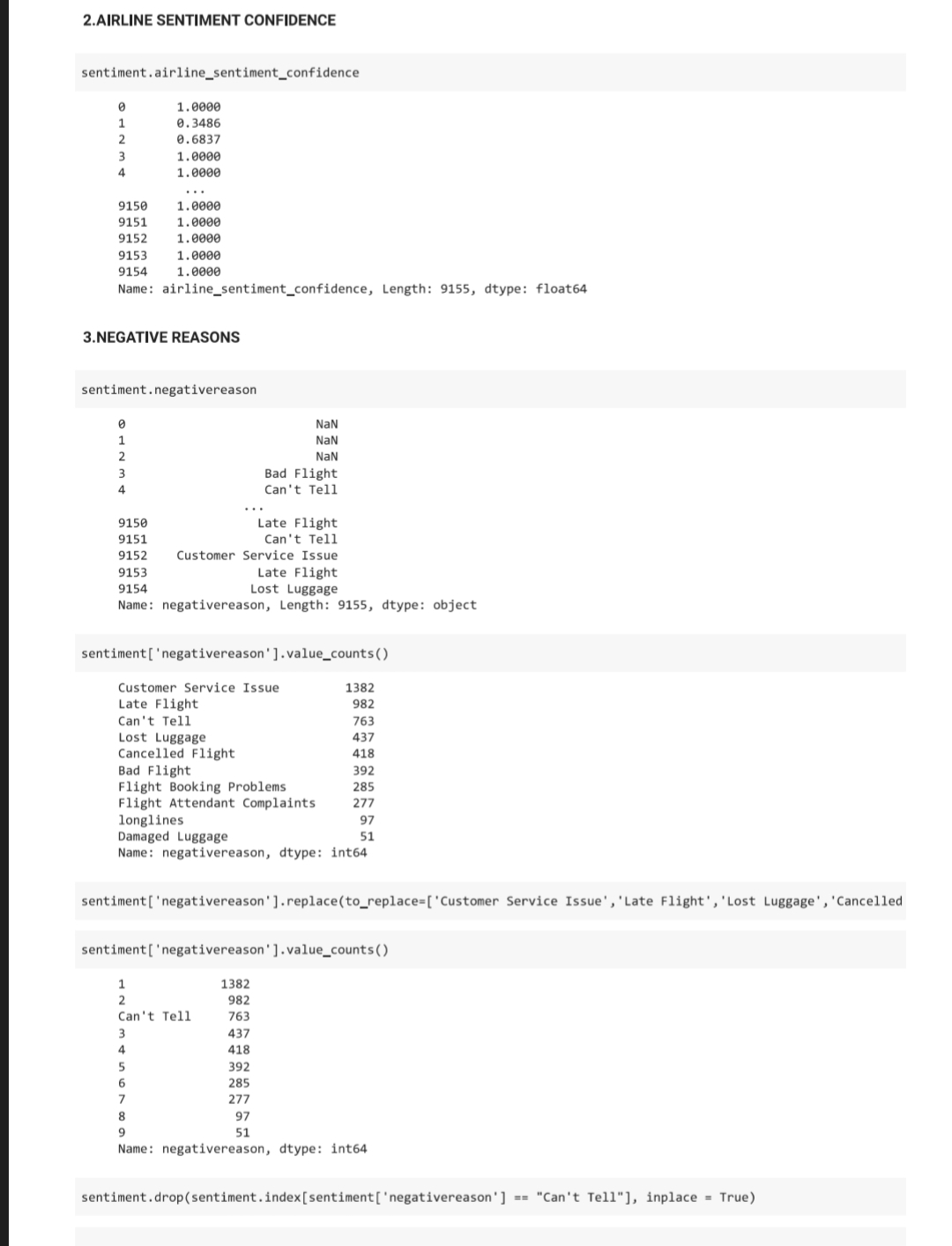
print("Predicted Customer Sentiment:" , predicted\_sentiment)

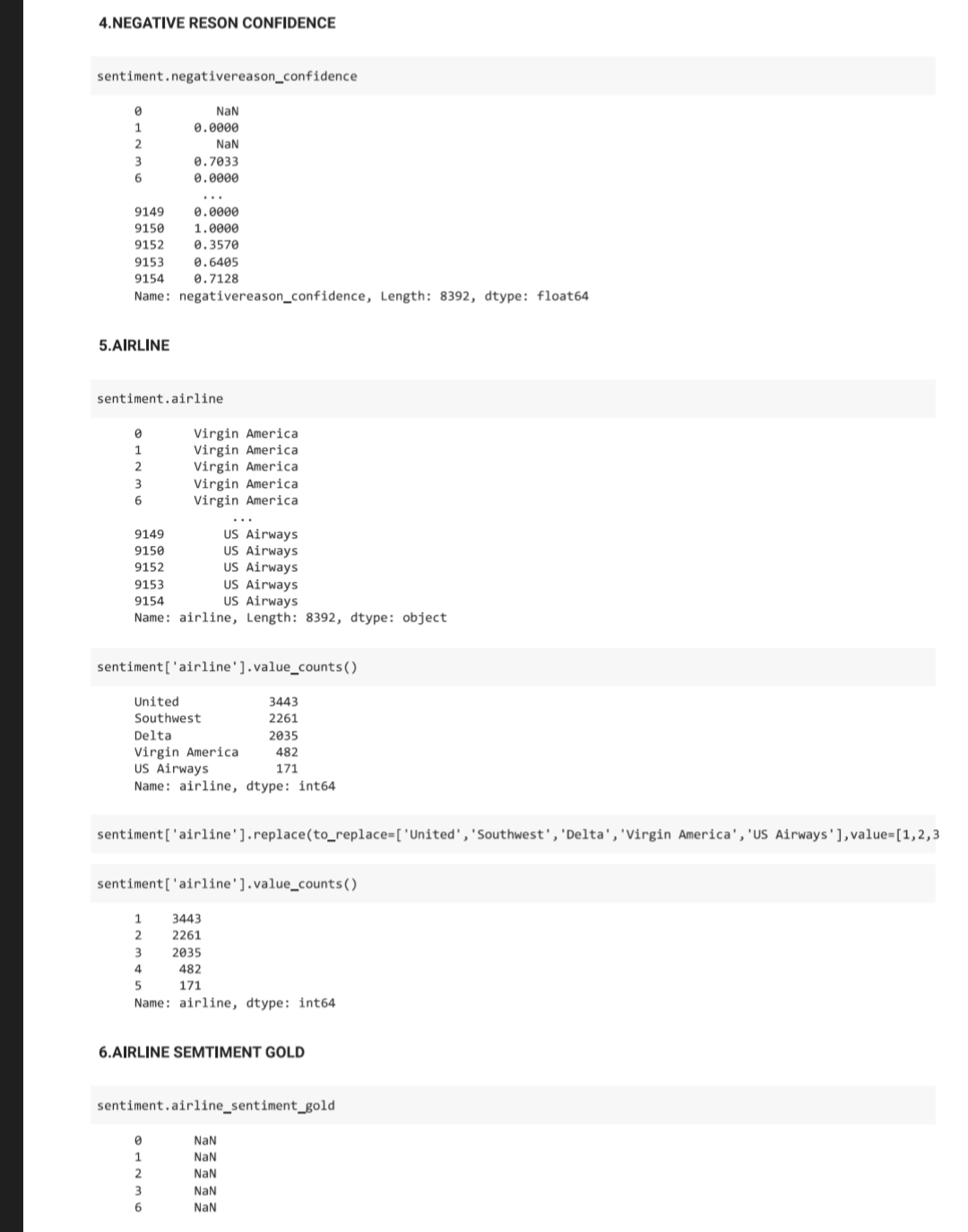






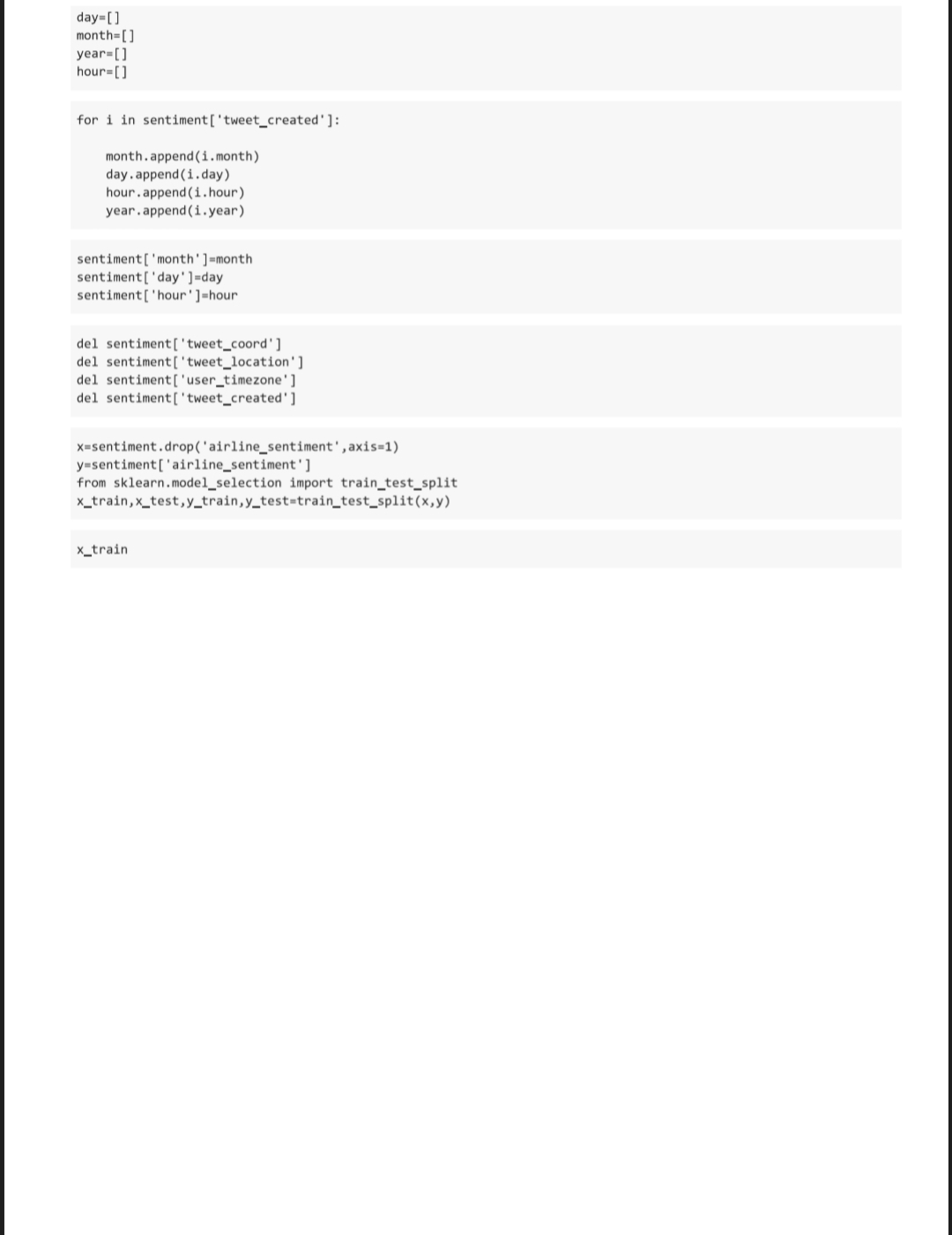




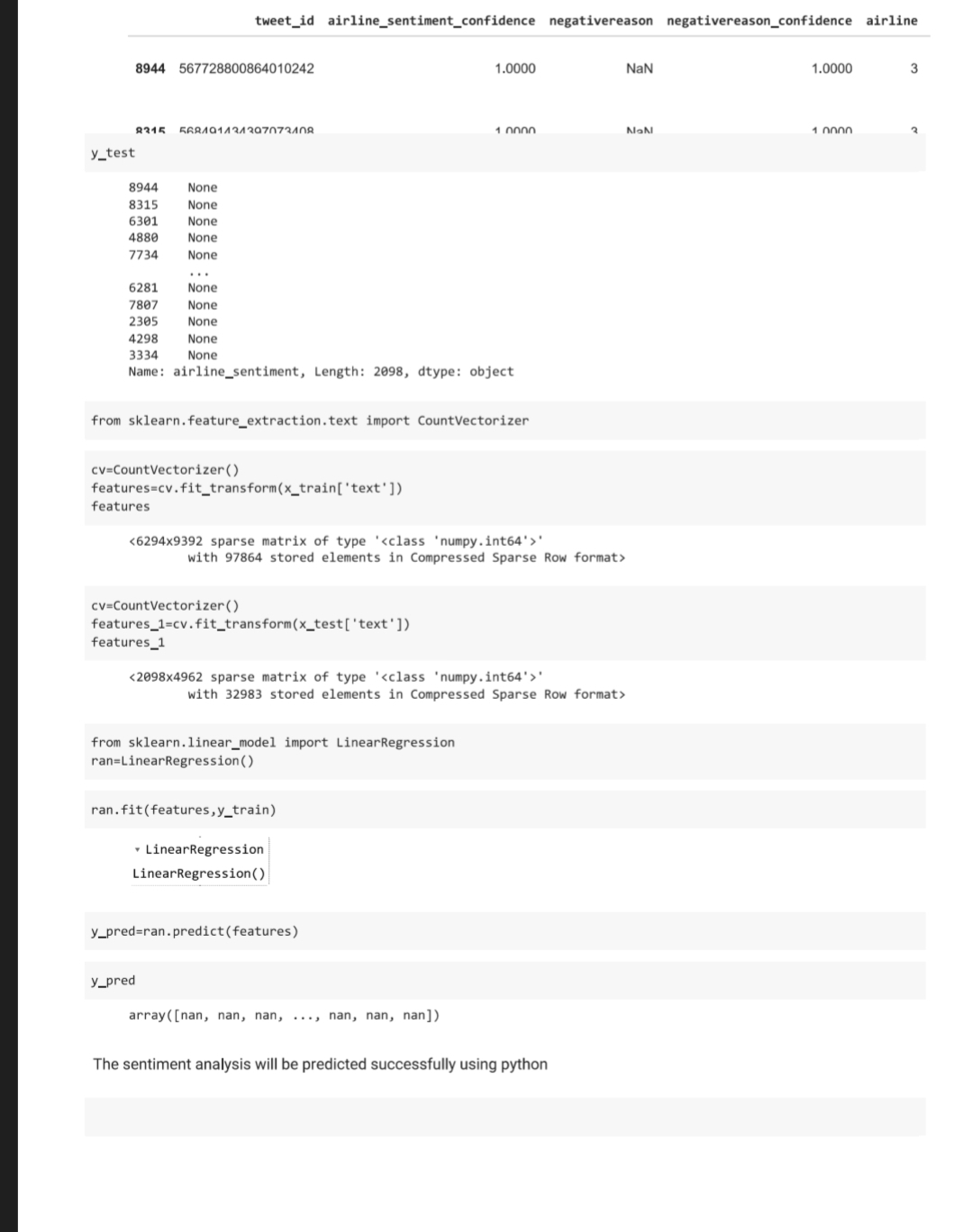












* **NAME OF THE STUDENT : JANANII.P**
* **DEPARTMENT : BE-ECE / 3RD YEAR**
* **COLLEGE : PANIMALAR INSTITUTE OF TECHNOLOGY**