HOTEL REVIEW SENTIMENT ANALYZER

Name: DIYA S DAS

Reg No.: 23BCE8650

College Name: VIT AP University, Amaravathi

Branch: Computer Science and Engineering with specialization in Data Analytics

Date of Submission: 08.07.2025

TABLE OF CONTENTS

INTRODUCTION

In today's competitive hospitality industry, customer experience is a crucial differentiator, and online reviews have become a key source of feedback for hotel businesses. These reviews contain valuable insights into guest satisfaction, service quality, and areas that need improvement. However, manually analyzing thousands of reviews is time-consuming and inefficient. This project leverages IBM's watsonx.ai and Generative AI capabilities to automate the analysis of hotel reviews. Using advanced foundation models, it classifies reviews based on sentiment (positive, neutral, or negative) and identifies frequently mentioned topics such as room service, cleanliness, and food quality to help hotels make data-driven decisions.

OBJECTIVE

The key objectives of the Hotel Review Sentiment Analyzer project are:

1. Understand Customer Sentiment

   To analyze hotel reviews and determine whether a guest's feedback is positive, neutral, or negative.

2. Identify Important Topics

   To automatically detect the main topics discussed in each review, such as food quality, room service, cleanliness, staff behaviour, and more.

3. Use Generative AI for Automation

   To leverage IBM's watsonx.ai and foundation models to automate the classification process, reducing the need for manual effort.

4. Generate a Structured Report

   To organize the classified data (sentiment and topics) into a clean, structured CSV format for easy understanding and future decision-making.

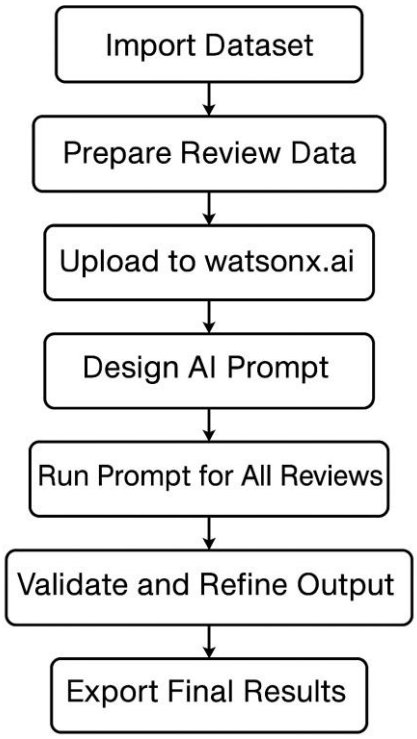
5. Help Hotels Improve Services

   To give hotel businesses meaningful insights from real customer feedback, helping them identify strengths and areas for improvement.

TOOLS AND TECHNOLOGIES USED

- Kaggle (for Dataset)
- Watsonx.ai Prompt Lb
- FLAN-UL2 Foundation model
- IBM Cloud
- Google Colab
- Python: Core programming language used for data handling, model interaction, and logic implementation.
- Jupyter Notebook: Interactive environment in IBM Watsonx for step-by-step development and testing.
- Pandas: Used for loading, cleaning, and displaying hotel review data in tabular format.
- IBM Cloud Object Storage (COS): Secure cloud storage for training and test datasets, accessed via ibm_boto3.
- IBM Watsonx Foundation Models (FLAN-UL2): Used for generating predictions on review sentiment and business category.
- IBM Watson Machine Learning (WML): Provides secure access to Watsonx models using API key and project ID.

METHODOLOGY



1. Importing the Dataset
    - A publicly available CSV file containing hotel reviews is downloaded from Kaggle.
    - The dataset contains several columns, but the focus is on the Review_Text column (containing customer feedback).
    - Unnecessary columns (like date, reviewer name, etc.) are removed for simplicity.
2. Preparing the Review Data
    - The review column is cleaned (removing special characters, handling null values).
    - A smaller sample (e.g., 100–200 rows) is selected if the dataset is too large for Prompt Lab to handle efficiently.
    - This cleaned data is saved in a new CSV file named cleaned_reviews.csv.
3. Uploading to IBM watsonx.ai Prompt Lab
    - The cleaned CSV is uploaded to the watsonx.ai Prompt Lab.
    - Prompt Lab is used to interact with a foundation model using natural language prompts.
4. Designing the AI Prompt
    - A well-crafted prompt is used to classify each review:

    - Identify the **sentiment** (Positive / Neutral / Negative)
    - Extract relevant **topics** (e.g., Room Service, Food Quality, Cleanliness, Staff, Location, Amenities)
5. Running the Prompt for All Reviews
    - The prompt is executed row-by-row using Prompt Lab's batch processing or manually for small sets.
    - Each review gets analyzed, and output is generated with sentiment and topic tags.
6. Validating and Refining Outputs
    - Results are reviewed to check accuracy.
    - If certain reviews are misclassified, prompt tuning is done to improve the output.
    - This iterative loop helps ensure the model understands domain-specific feedback.
7. Exporting the Final Results
    - The sentiment and topics for each review are appended to the dataset.
    - A new CSV file is generated.

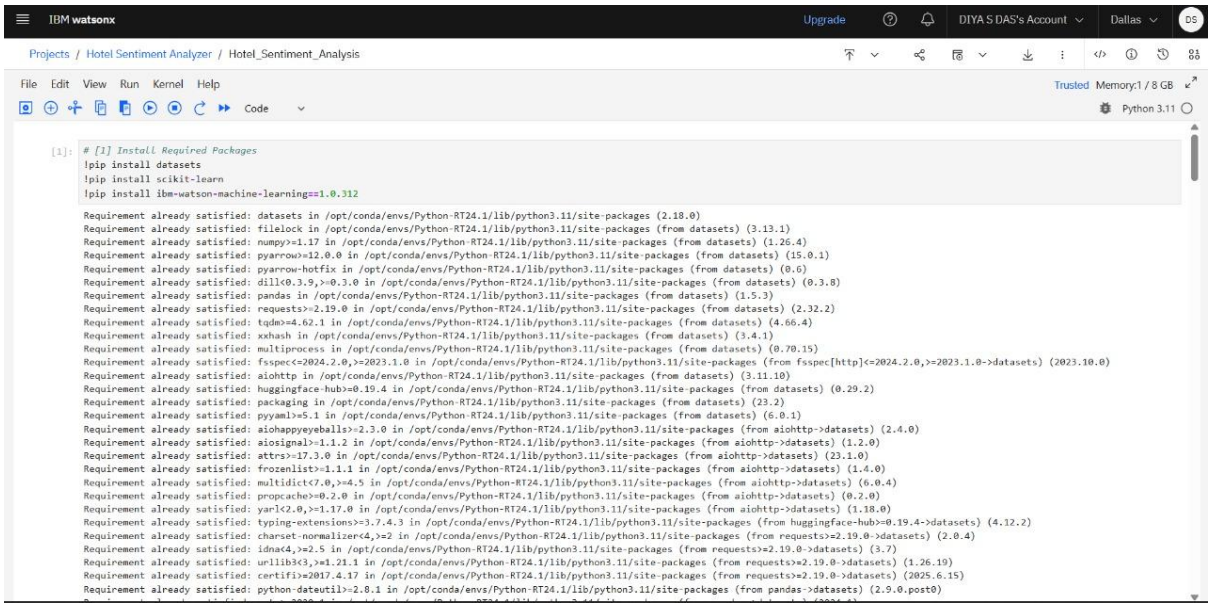
CODE SNIPPETS

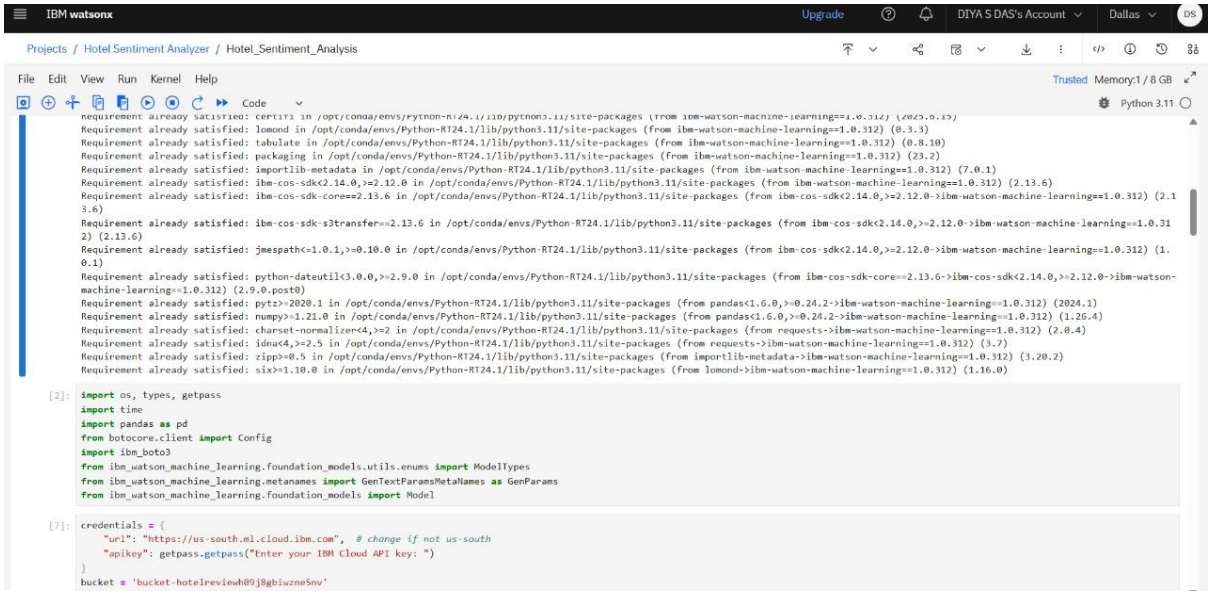
```
[ ]  import pandas as pd

     # Load the uploaded file
     df = pd.read_csv("hotel_reviews.csv")

     # Show all column names
     print(df.columns)
```

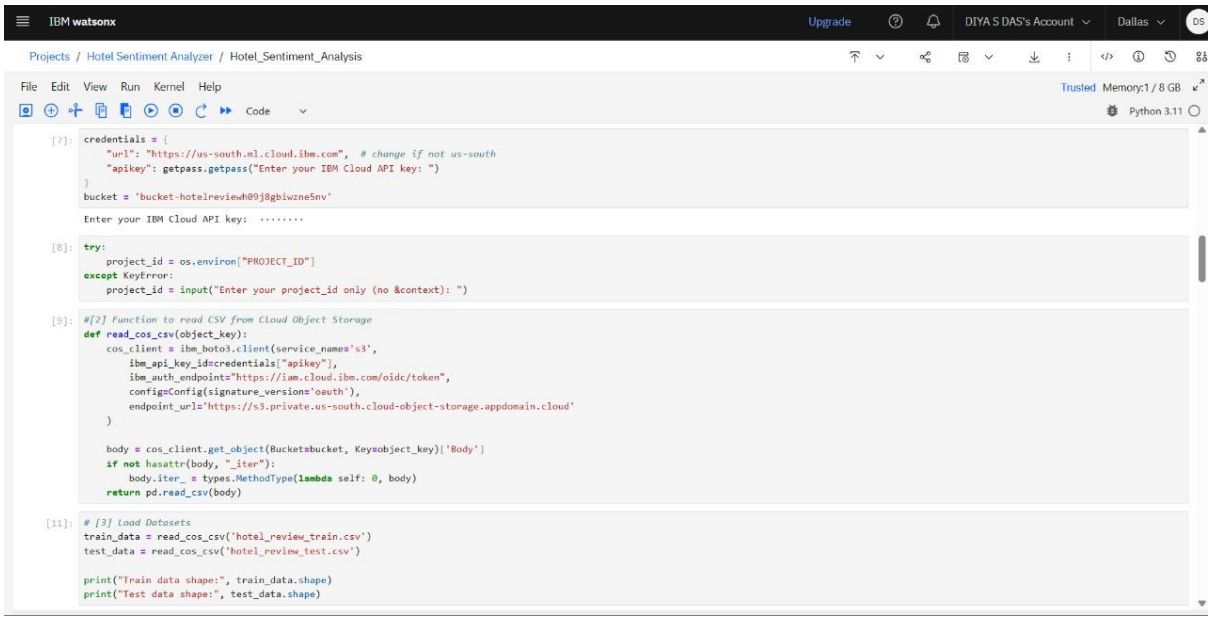
```
Index(['Index', 'Name', 'Area', 'Review_Date', 'Rating_attribute',
       'Rating(Out of 10)', 'Review_Text'],
      dtype='object')
```

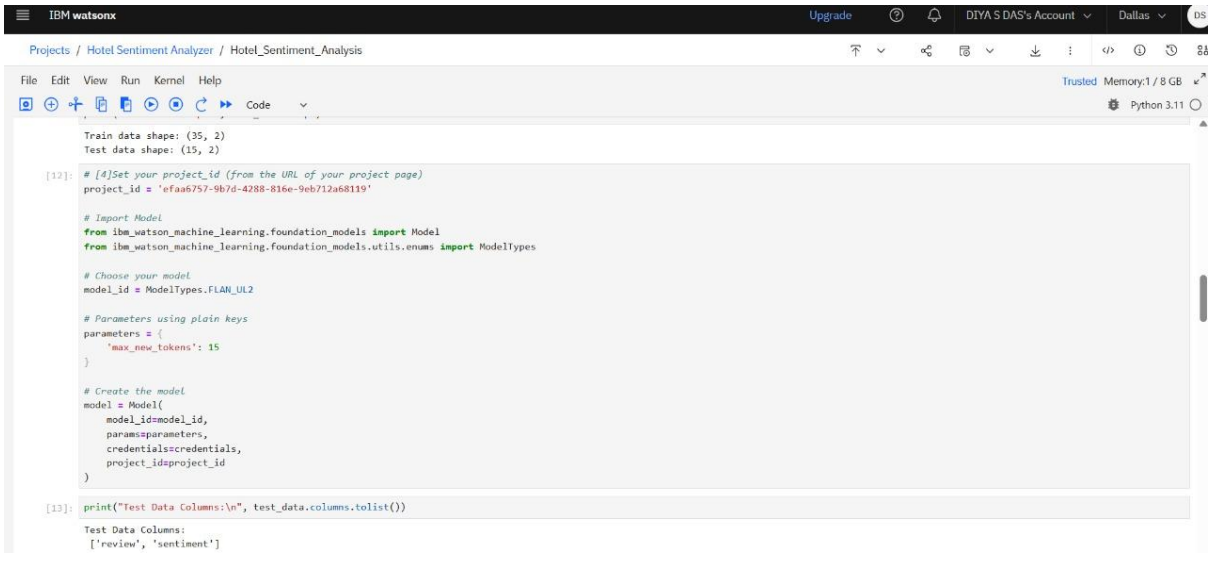
```
import pandas as pd
import re
from google.colab import files

# Upload and read CSV
uploaded = files.upload()
df = pd.read_csv("hotel_reviews.csv")

# Use the correct column name here (replace with actual name)
print(df.columns)  # Use this to verify

# Example: if your column is 'Review_Text', replace below
df['Cleaned_Review'] = df['Review_Text'].apply(lambda x: re.sub(r"[^a-zA-Z0-9\s.,!?']", '', str(x)).strip())
df = df.dropna(subset=['Cleaned_Review'])

# Take a small sample
sample = df[['Cleaned_Review']].sample(100, random_state=42)
sample.to_csv("sample_reviews.csv", index=False)

# Download the cleaned file
files.download("sample_reviews.csv")
```

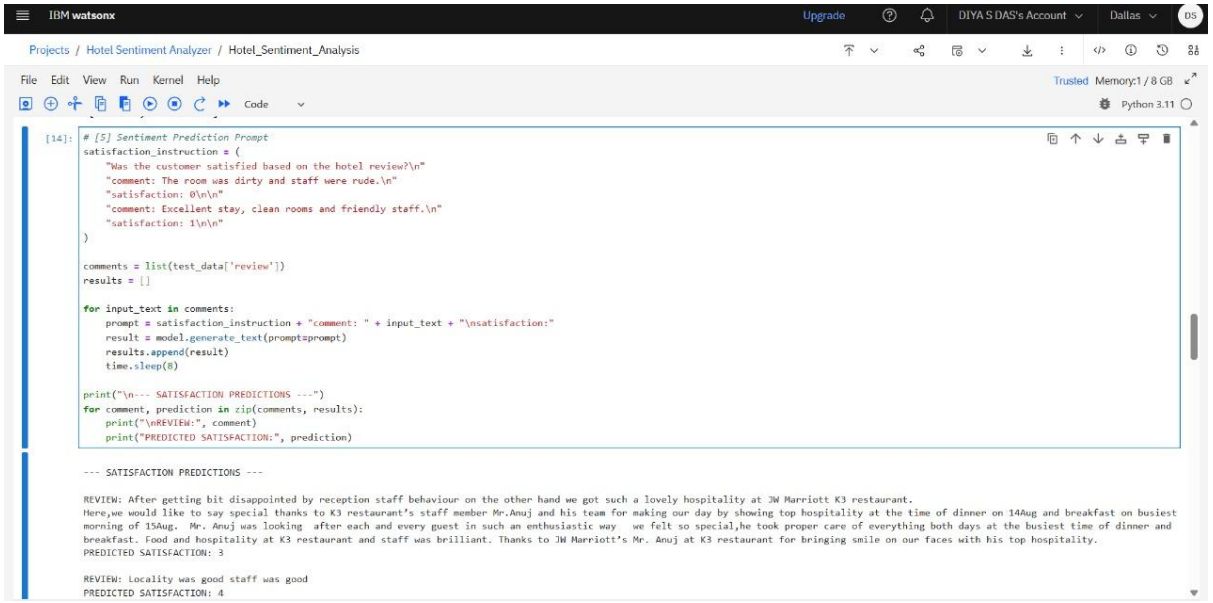
```
Choose Files  hotel_reviews.csv
• hotel_reviews.csv(text/csv) - 1307592 bytes, last modified: 7/2/2025 - 100% done
Saving hotel_reviews.csv to hotel_reviews (1).csv
Index(['Index', 'Name', 'Area', 'Review_Date', 'Rating_attribute',
       'Rating(Out of 10)', 'Review_Text'],
      dtype='object')
```

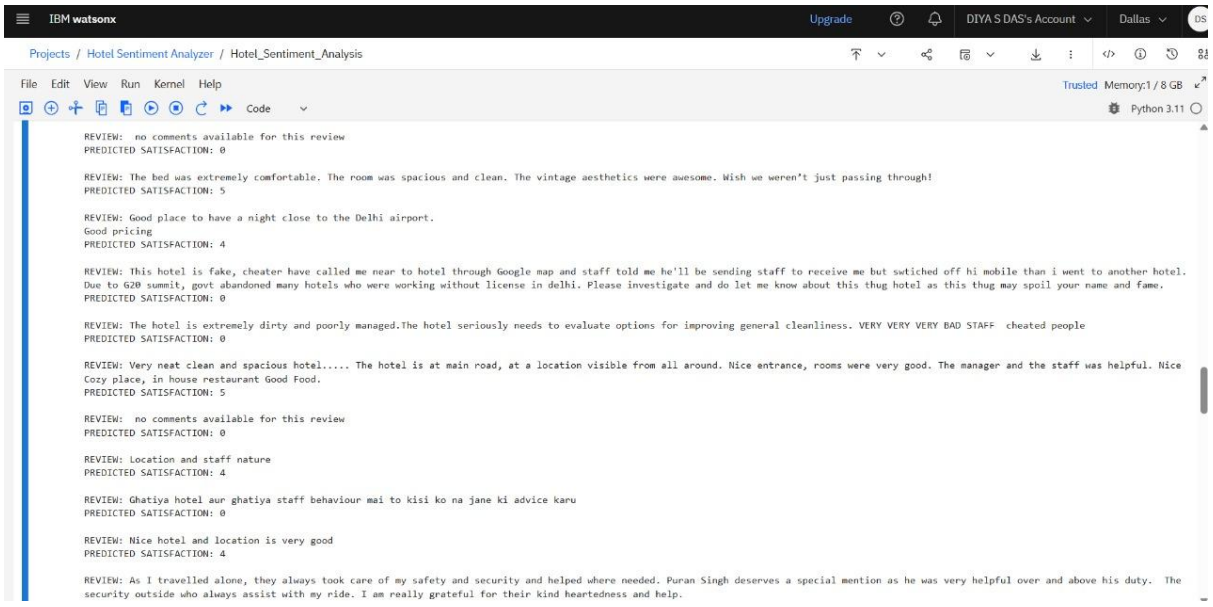


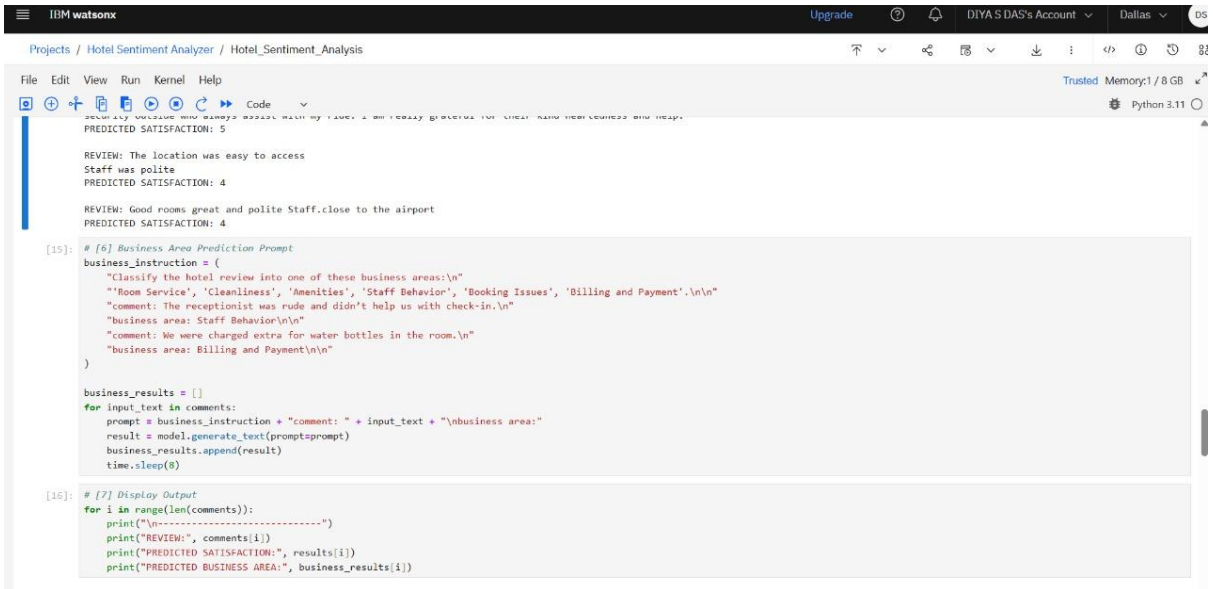

IBM watsonx
Upgrade
DIYA S DAS's Account
Dallas
Projects / Hotel Sentiment Analyzer / Hotel_Sentiment_Analysis
File Edit View Run Kernel Help
Trusted Memory:1 / 8 GB
Code
Python 3.11
Requirement already satisfied: lomond in /opt/conda/envs/Python-RT24.1/lib/python3.11/site-packages (from ibm-watson-machine-learning==1.0.312) (0.3.3)
Requirement already satisfied: tabulate in /opt/conda/envs/Python-RT24.1/lib/python3.11/site-packages (from ibm-watson-machine-learning==1.0.312) (0.8.10)
Requirement already satisfied: packaging in /opt/conda/envs/Python-RT24.1/lib/python3.11/site-packages (from ibm-watson-machine-learning==1.0.312) (23.2)
Requirement already satisfied: importlib-metadata in /opt/conda/envs/Python-RT24.1/lib/python3.11/site-packages (from ibm-watson-machine-learning==1.0.312) (7.0.1)
Requirement already satisfied: ibm-cos-sdk<2.14.0,>=2.12.0 in /opt/conda/envs/Python-RT24.1/lib/python3.11/site-packages (from ibm-watson-machine-learning==1.0.312) (2.13.6)
Requirement already satisfied: ibm-cos-sdk-core==2.13.6 in /opt/conda/envs/Python-RT24.1/lib/python3.11/site-packages (from ibm-cos-sdk<2.14.0,>=2.12.0->ibm-watson-machine-learning==1.0.312) (2.13.6)
Requirement already satisfied: ibm-cos-sdk-s3transfer==2.13.6 in /opt/conda/envs/Python-RT24.1/lib/python3.11/site-packages (from ibm-cos-sdk<2.14.0,>=2.12.0->ibm-watson-machine-learning==1.0.312) (2.13.6)
Requirement already satisfied: jmespath<=1.0.1,>=0.10.0 in /opt/conda/envs/Python-RT24.1/lib/python3.11/site-packages (from ibm-cos-sdk<2.14.0,>=2.12.0->ibm-watson-machine-learning==1.0.312) (1.0.1)
Requirement already satisfied: python-dateutil<3.0.0,>=2.9.0 in /opt/conda/envs/Python-RT24.1/lib/python3.11/site-packages (from ibm-cos-sdk-core==2.13.6->ibm-cos-sdk<2.14.0,>=2.12.0->ibm-watson-machine-learning==1.0.312) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in /opt/conda/envs/Python-RT24.1/lib/python3.11/site-packages (from pandas<1.6.0,>=0.24.2->ibm-watson-machine-learning==1.0.312) (2024.1)
Requirement already satisfied: numpy>=1.21.0 in /opt/conda/envs/Python-RT24.1/lib/python3.11/site-packages (from pandas<1.6.0,>=0.24.2->ibm-watson-machine-learning==1.0.312) (1.26.4)
Requirement already satisfied: charset-normalizer<4,>=2 in /opt/conda/envs/Python-RT24.1/lib/python3.11/site-packages (from requests->ibm-watson-machine-learning==1.0.312) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in /opt/conda/envs/Python-RT24.1/lib/python3.11/site-packages (from requests->ibm-watson-machine-learning==1.0.312) (3.7)
Requirement already satisfied: zipp>=0.5 in /opt/conda/envs/Python-RT24.1/lib/python3.11/site-packages (from importlib-metadata->ibm-watson-machine-learning==1.0.312) (3.20.2)
Requirement already satisfied: six>=1.10.0 in /opt/conda/envs/Python-RT24.1/lib/python3.11/site-packages (from lomond->ibm-watson-machine-learning==1.0.312) (1.16.0)
[2]:
import os, types, getpass
import time
import pandas as pd
from botocore.client import Config
import ibm_boto3
from ibm_watson_machine_learning.foundation_models.utils.enums import ModelTypes
from ibm_watson_machine_learning.metanames import GenTextParamsMetaNames as GenParams
from ibm_watson_machine_learning.foundation_models import Model
[7]:
credentials = {
"url": "https://us-south.ml.cloud.ibm.com", # change if not us-south
"apikey": getpass.getpass("Enter your IBM Cloud API key: ")
}
bucket = 'bucket-hotelreviewh09j8gbiwzne5nv'

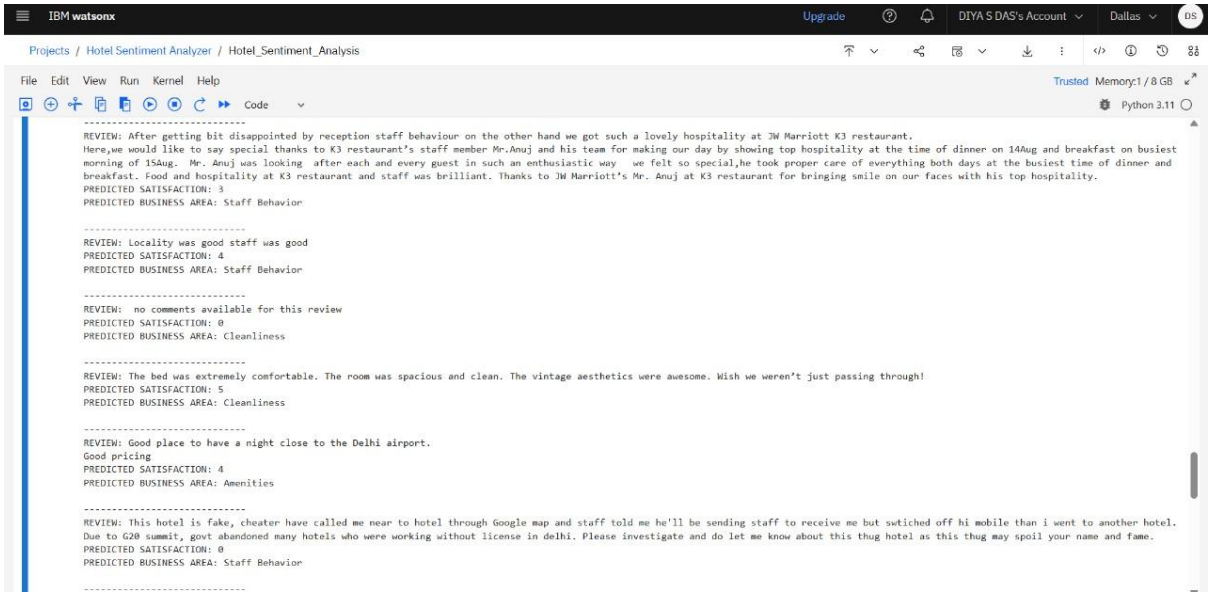

IBM watsonx
Upgrade
DIYA S DAS's Account
Dallas
Projects / Hotel Sentiment Analyzer / Hotel_Sentiment_Analysis
File Edit View Run Kernel Help
Trusted Memory:1 / 8 GB
Code
Python 3.11
[7]:
credentials = {
"url": "https://us-south.ml.cloud.ibm.com", # change if not us-south
"apikey": getpass.getpass("Enter your IBM Cloud API key: ")
}
bucket = 'bucket-hotelreviewh09j8gbiwzne5nv'
Enter your IBM Cloud API key: ·····
[8]:
try:
project_id = os.environ["PROJECT_ID"]
except KeyError:
project_id = input("Enter your project_id only (no &context): ")
[9]:
#[2] Function to read CSV from Cloud Object Storage
def read_cos_csv(object_key):
cos_client = ibm_boto3.client(service_name='s3',
ibm_api_key_id=credentials["apikey"],
ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token",
config=Config(signature_version='oauth'),
endpoint_url='https://s3.private.us-south.cloud-object-storage.appdomain.cloud'
)
body = cos_client.get_object(Bucket=bucket, Key=object_key)['Body']
if not hasattr(body, "_iter"):
body.iter_ = types.MethodType(lambda self: 0, body)
return pd.read_csv(body)
[11]:
# [3] Load Datasets
train_data = read_cos_csv('hotel_review_train.csv')
test_data = read_cos_csv('hotel_review_test.csv')
print("Train data shape:", train_data.shape)
print("Test data shape:", test_data.shape)


IBM watsonx
Upgrade
DIYA S DAS's Account
Dallas
Projects / Hotel Sentiment Analyzer / Hotel_Sentiment_Analysis
File Edit View Run Kernel Help
Trusted Memory:1 / 8 GB
Code
Python 3.11
Train data shape: (35, 2)
Test data shape: (15, 2)
[12]:
# [4]Set your project_id (from the URL of your project page)
project_id = 'efaa6757-9b7d-4288-816e-9eb712a68119'
# Import Model
from ibm_watson_machine_learning.foundation_models import Model
from ibm_watson_machine_learning.foundation_models.utils.enums import ModelTypes
# Choose your model
model_id = ModelTypes.FLAN_UL2
# Parameters using plain keys
parameters = {
'max_new_tokens': 15
}
# Create the model
model = Model(
model_id=model_id,
params=parameters,
credentials=credentials,
project_id=project_id
)
[13]:
print("Test Data Columns:\n", test_data.columns.tolist())
Test Data Columns:
['review', 'sentiment']


IBM watsonx
Upgrade
DIYA S DAS's Account
Dallas
Projects / Hotel Sentiment Analyzer / Hotel_Sentiment_Analysis
File Edit View Run Kernel Help
Trusted Memory:1 / 8 GB
Code
Python 3.11
[14]:
# [5] Sentiment Prediction Prompt
satisfaction_instruction = (
"Was the customer satisfied based on the hotel review?\n"
"comment: The room was dirty and staff were rude.\n"
"satisfaction: 0\n\n"
"comment: Excellent stay, clean rooms and friendly staff.\n"
"satisfaction: 1\n\n"
)
comments = list(test_data['review'])
results = []
for input_text in comments:
prompt = satisfaction_instruction + "comment: " + input_text + "\nsatisfaction:"
result = model.generate_text(prompt=prompt)
results.append(result)
time.sleep(8)
print("\n--- SATISFACTION PREDICTIONS ---")
for comment, prediction in zip(comments, results):
print("\nREVIEW:", comment)
print("PREDICTED SATISFACTION:", prediction)
--- SATISFACTION PREDICTIONS ---
REVIEW: After getting bit disappointed by reception staff behaviour on the other hand we got such a lovely hospitality at JW Marriott K3 restaurant.
Here,we would like to say special thanks to K3 restaurant's staff member Mr.Anuj and his team for making our day by showing top hospitality at the time of dinner on 14Aug and breakfast on busiest morning of 15Aug. Mr. Anuj was looking after each and every guest in such an enthusiastic way we felt so special,he took proper care of everything both days at the busiest time of dinner and breakfast. Food and hospitality at K3 restaurant and staff was brilliant. Thanks to JW Marriott's Mr. Anuj at K3 restaurant for bringing smile on our faces with his top hospitality.
PREDICTED SATISFACTION: 3
REVIEW: Locality was good staff was good
PREDICTED SATISFACTION: 4


IBM watsonx
Upgrade
DIYA S DAS's Account
Dallas
Projects / Hotel Sentiment Analyzer / Hotel_Sentiment_Analysis
File Edit View Run Kernel Help
Trusted Memory:1 / 8 GB
Code
Python 3.11
REVIEW: no comments available for this review
PREDICTED SATISFACTION: 0
REVIEW: The bed was extremely comfortable. The room was spacious and clean. The vintage aesthetics were awesome. Wish we weren't just passing through!
PREDICTED SATISFACTION: 5
REVIEW: Good place to have a night close to the Delhi airport.
Good pricing
PREDICTED SATISFACTION: 4
REVIEW: This hotel is fake, cheater have called me near to hotel through Google map and staff told me he'll be sending staff to receive me but swtiched off hi mobile than i went to another hotel. Due to G20 summit, govt abandoned many hotels who were working without license in delhi. Please investigate and do let me know about this thug hotel as this thug may spoil your name and fame.
PREDICTED SATISFACTION: 0
REVIEW: The hotel is extremely dirty and poorly managed.The hotel seriously needs to evaluate options for improving general cleanliness. VERY VERY VERY BAD STAFF cheated people
PREDICTED SATISFACTION: 0
REVIEW: Very neat clean and spacious hotel..... The hotel is at main road, at a location visible from all around. Nice entrance, rooms were very good. The manager and the staff was helpful. Nice Cozy place, in house restaurant Good Food.
PREDICTED SATISFACTION: 5
REVIEW: no comments available for this review
PREDICTED SATISFACTION: 0
REVIEW: Location and staff nature
PREDICTED SATISFACTION: 4
REVIEW: Ghatiya hotel aur ghatiya staff behaviour mai to kisi ko na jane ki advice karu
PREDICTED SATISFACTION: 0
REVIEW: Nice hotel and location is very good
PREDICTED SATISFACTION: 4
REVIEW: As I travelled alone, they always took care of my safety and security and helped where needed. Puran Singh deserves a special mention as he was very helpful over and above his duty. The security outside who always assist with my ride. I am really grateful for their kind heartedness and help.


IBM watsonx
Upgrade
DIYA S DAS's Account
Dallas
Projects / Hotel Sentiment Analyzer / Hotel_Sentiment_Analysis
File Edit View Run Kernel Help
Trusted Memory:1 / 8 GB
Code
Python 3.11
PREDICTED SATISFACTION: 5
REVIEW: The location was easy to access
Staff was polite
PREDICTED SATISFACTION: 4
REVIEW: Good rooms great and polite Staff.close to the airport
PREDICTED SATISFACTION: 4
[15]:
# [6] Business Area Prediction Prompt
business_instruction = (
"Classify the hotel review into one of these business areas:\n"
"'Room Service', 'Cleanliness', 'Amenities', 'Staff Behavior', 'Booking Issues', 'Billing and Payment'.\n\n"
"comment: The receptionist was rude and didn't help us with check-in.\n"
"business area: Staff Behavior\n\n"
"comment: We were charged extra for water bottles in the room.\n"
"business area: Billing and Payment\n\n"
)
business_results = []
for input_text in comments:
prompt = business_instruction + "comment: " + input_text + "\nbusiness area:"
result = model.generate_text(prompt=prompt)
business_results.append(result)
time.sleep(8)
[16]:
# [7] Display Output
for i in range(len(comments)):
print("\n----------------------------")
print("REVIEW:", comments[i])
print("PREDICTED SATISFACTION:", results[i])
print("PREDICTED BUSINESS AREA:", business_results[i])


IBM watsonx
Upgrade
DIYA S DAS's Account
Dallas
Projects / Hotel Sentiment Analyzer / Hotel_Sentiment_Analysis
File Edit View Run Kernel Help
Trusted Memory:1 / 8 GB
Code
Python 3.11
REVIEW: After getting bit disappointed by reception staff behaviour on the other hand we got such a lovely hospitality at JW Marriott K3 restaurant.
Here,we would like to say special thanks to K3 restaurant's staff member Mr.Anuj and his team for making our day by showing top hospitality at the time of dinner on 14Aug and breakfast on busiest morning of 15Aug. Mr. Anuj was looking after each and every guest in such an enthusiastic way we felt so special,he took proper care of everything both days at the busiest time of dinner and breakfast. Food and hospitality at K3 restaurant and staff was brilliant. Thanks to JW Marriott's Mr. Anuj at K3 restaurant for bringing smile on our faces with his top hospitality.
PREDICTED SATISFACTION: 3
PREDICTED BUSINESS AREA: Staff Behavior
REVIEW: Locality was good staff was good
PREDICTED SATISFACTION: 4
PREDICTED BUSINESS AREA: Staff Behavior
REVIEW: no comments available for this review
PREDICTED SATISFACTION: 0
PREDICTED BUSINESS AREA: Cleanliness
REVIEW: The bed was extremely comfortable. The room was spacious and clean. The vintage aesthetics were awesome. Wish we weren't just passing through!
PREDICTED SATISFACTION: 5
PREDICTED BUSINESS AREA: Cleanliness
REVIEW: Good place to have a night close to the Delhi airport.
Good pricing
PREDICTED SATISFACTION: 4
PREDICTED BUSINESS AREA: Amenities
REVIEW: This hotel is fake, cheater have called me near to hotel through Google map and staff told me he'll be sending staff to receive me but swtiched off hi mobile than i went to another hotel. Due to G20 summit, govt abandoned many hotels who were working without license in delhi. Please investigate and do let me know about this thug hotel as this thug may spoil your name and fame.
PREDICTED SATISFACTION: 0
PREDICTED BUSINESS AREA: Staff Behavior


File Edit View Run Kernel Help
Trusted Memory:1 / 8 GB
Code
Python 3.11
REVIEW: The hotel is extremely dirty and poorly managed.The hotel seriously needs to evaluate options for improving general cleanliness. VERY VERY VERY BAD STAFF cheated people
PREDICTED SATISFACTION: 0
PREDICTED BUSINESS AREA: Cleanliness
REVIEW: Very neat clean and spacious hotel..... The hotel is at main road, at a location visible from all around. Nice entrance, rooms were very good. The manager and the staff was helpful. Nice Cozy place, in house restaurant Good Food.
PREDICTED SATISFACTION: 5
PREDICTED BUSINESS AREA: Cleanliness
REVIEW: no comments available for this review
PREDICTED SATISFACTION: 0
PREDICTED BUSINESS AREA: Cleanliness
REVIEW: Location and staff nature
PREDICTED SATISFACTION: 4
PREDICTED BUSINESS AREA: Staff Behavior
REVIEW: Ghatiya hotel aur ghatiya staff behaviour mai to kisi ko na jane ki advice karu
PREDICTED SATISFACTION: 0
PREDICTED BUSINESS AREA: Staff Behavior
REVIEW: Nice hotel and location is very good
PREDICTED SATISFACTION: 4
PREDICTED BUSINESS AREA: Amenities
REVIEW: As I travelled alone, they always took care of my safety and security and helped where needed. Puran Singh deserves a special mention as he was very helpful over and above his duty. The security outside who always assist with my ride. I am really grateful for their kind heartedness and help.
PREDICTED SATISFACTION: 5
PREDICTED BUSINESS AREA: Staff Behavior

OUTPUT WITH PROPER EXPLANATION

```
[21]: # [8] Final Output
output_df = pd.DataFrame({
    "Review": comments,
    "Predicted_Satisfaction": results,
    "Predicted_Business_Area": business_results
})

output_df.to_csv("watsonx_predictions_output.csv", index=False)
print("Output saved to watsonx_predictions_output.csv")

Output saved to watsonx_predictions_output.csv
```

The code begins by importing necessary Python libraries such as pandas, ibm_boto3, and ibm_watson_machine_learning.foundation_models. These tools are used to read datasets from IBM Cloud Object Storage (COS), authenticate with IBM Cloud using an API key and Project ID, and interact with a foundation model on Watsonx.

A secure connection is established to access the hotel_reviews_train.csv and hotel_reviews_test.csv files stored in the COS bucket. These datasets are loaded into Pandas DataFrames for analysis.

The Watsonx **FLAN-UL2 foundation model** is used to perform two key tasks:

1. **Sentiment Analysis** – Predicting whether a customer's hotel review expresses satisfaction (positive) or dissatisfaction (negative).
2. **Business Area Classification** – Categorizing each review into business-relevant areas such as *Cleanliness*, *Staff Behavior*, *Billing and Payment*, etc.

Each review from the test dataset is passed through the model. Based on the prompt logic, the model generates corresponding sentiment and business area labels. The results are stored and printed in a structured format.

This approach helps convert unstructured customer feedback into valuable business insights, allowing hotel managers to easily identify strengths and areas needing improvement.

GITHUB LINK

https://github.com/diyaa1201/Hotel_Review_Setiment_Analyzer

CHALLENGES FACED

- Prompt Tuning for Accurate Classification

The initial prompts provided to the foundation model often returned inconsistent or vague results, especially when the task involved identifying specific business areas from a review. To improve model accuracy, the prompts had to be carefully rewritten with clearer instructions, structured formatting, and well-defined label options.

- Handling Mixed Sentiment in Reviews

Many hotel reviews included both positive and negative opinions within the same sentence or paragraph. This made it difficult for the model to assign a clear binary sentiment (positive/negative). For example, a review saying "The room was clean, but the service was poor" reflects conflicting sentiments, challenging even for advanced models.

- Manual Review and Correction of Model Outputs

Despite using a powerful model like FLAN-UL2, some generated outputs were either irrelevant, overly generic, or misaligned with the actual content of the review. In such cases, manual intervention was necessary to validate and correct these predictions to ensure the overall quality and reliability of the analysis.

- Token Limits and Runtime Delays on Free Plan

Due to rate limits on the Lite plan of IBM Watsonx, only 2 requests could be processed every 15 seconds. This significantly increased execution time, especially when analyzing a large number of reviews. Strategic use of time.sleep() was required to avoid exceeding request limits.

- Lack of Labeled Data for Training (Optional)

Since the project was based on inference-only using foundation models, there was no labeled training data used for supervised learning. This limited

the ability to evaluate prediction accuracy quantitatively, and made it difficult to fine-tune or retrain the model for better performance.

CONCLUSION

The *Hotel Review Sentiment Analyzer* project successfully demonstrated the application of Generative AI and foundation models to extract meaningful insights from unstructured customer feedback. By leveraging IBM Watsonx's FLAN-UL2 foundation model, the project performed two essential tasks: sentiment classification (satisfied vs. unsatisfied customers) and topic categorization (e.g., cleanliness, pricing, service).

The project began by securely uploading and accessing hotel review datasets from IBM Cloud Object Storage. Using Python and Jupyter Notebook in IBM Watsonx.ai, the reviews were preprocessed and passed to the foundation model through a series of carefully designed prompts. The results provided clear labels for customer sentiment and identified business areas requiring attention.

Despite certain challenges—such as handling mixed sentiments, refining prompt instructions, and managing model limits on the free plan—these were effectively addressed through iterative improvements and manual validation. The outcome was a structured and scalable approach to processing large volumes of feedback, which can be extended to other industries like airlines, e-commerce, or banking.

Overall, this project highlighted the power of Generative AI in automating the analysis of textual data and improving decision-making in customer-facing domains. It not only enhanced technical skills in working with foundation models but also provided hands-on experience with real-world AI workflows in IBM Watsonx.

REFERENCES

IBM Watsonx Documentation

https://www.ibm.com/docs/en/watsonx

IBM Cloud Object Storage

https://www.ibm.com/cloud/object-storage

IBM Watson Machine Learning Python Client

https://pypi.org/project/ibm-watson-machine-learning/

Pandas Documentation

https://pandas.pydata.org/docs/

scikit-learn Documentation

https://scikit-learn.org/stable/