INTERNSHIP PROJECT PHASE- DEVELOPMENT

PROJECT 6 – DEVELOPMENT AND TESTING OF YELP DATA ANALYSIS PROJECT ON IBM CLOUD

1. Introduction

The rapid growth of online reviews and location-based services has led to the availability of large-scale, structured, and semi-structured datasets. Yelp, being a major platform for user reviews, provides a rich dataset for analysis. This project focuses on analyzing Yelp's business and check-in data using IBM Cloud infrastructure, targeting patterns in consumer behavior, business characteristics, and activity trends. It uses IBM Watson Studio and other IBM Cloud services for scalable data processing, visualization, and analysis.

2. Project Requirements and Dataset Overview

Requirements:

1. IBM Cloud account with active services.
2. Yelp academic dataset (JSON format).
3. Python and Jupyter environment for data analysis.
4. Basic understanding of data science libraries: pandas, numpy, seaborn, matplotlib.

3. Dataset Description:

yelp_academic_dataset_business.json – Contains metadata about businesses: name, location, star ratings, review counts, and categories.

yelp_academic_dataset_checkin.json – Contains time-based check-in data for each business.

These datasets are used to build insights regarding geographical distribution, popular business categories, customer activity hours, and business ratings.

4.Development Environment on IBM Cloud

The development of the project was carried out using the following IBM Cloud services:

IBM Watson Studio: For Jupyter Notebook creation and Python-based data analysis.

IBM Cloud Object Storage: To store input datasets and retrieve them programmatically.

IBM Cloud Functions (Optional): For automation and batch jobs.

IBM Cognos Dashboard Embedded: Optional tool for data visualization.

The following development stack was used:

Python 3.9+

Jupyter Notebooks

Libraries: pandas, seaborn, matplotlib, json, datetime

5. Implementation Steps (Detailed)

Step 1: Cloud Setup

Register/Login to IBM Cloud.

Create and configure Object Storage instance.

Launch Watson Studio.

Step 2: Project Creation

Create a new project in Watson Studio.

Associate the Object Storage service.

Step 3: Data Upload

Upload business.json and checkin.json to Object Storage.

Generate access credentials to connect from the notebook.

Step 4: Notebook Initialization

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

business_df = pd.read_json('business.json', lines=True)
checkin_df = pd.read_json('checkin.json', lines=True)
```

Step 5: Preprocessing

Handle missing/null values.

Normalize nested categories and attributes.

Convert date strings into datetime objects.

Merge datasets using business_id as key.

Step 6: Analysis and Visualization

City-wise business count

Most common categories

Distribution of check-ins over time

Correlation between ratings and check-ins

Step 7: Save Outputs

Export graphs and tables as images and CSVs.

Store final reports in IBM Cloud Object Storage.

6. Key Modules

Data Cleaning Module:

Replaces nulls.

Parses categories.

Filters out businesses with zero reviews.

Check-in Analysis Module:

Aggregates check-ins by day and hour.

Visualizes patterns using heatmaps.

Correlation Module:

Finds relation between business attributes (like stars, categories) and customer engagement.

7. Output and Reporting

Visualizations created:

Histogram of star ratings.

Bar graph of business count by city.

Heatmap of check-in times.

Pie chart of top 10 categories.

Reports stored as:

.png images for plots.

.csv and .xlsx for processed data.

8. Challenges

- **Large Dataset Size**: Processed using pandas.read_json() with lines=True in chunks to manage memory efficiently.
- **Nested JSON Fields**: Flattened using json_normalize() and dictionary expansion for easier tabular conversion.
- **Memory Constraints**: Leveraged IBM Watson Studio cloud resources to avoid local memory limitations.
- **Future Improvements**: Plan to connect to IBM DB2 or Cloud Object Storage (COS) with Apache Spark for scalable distributed computing.