MINISTRY OF EDUCATION AND TRAINING
HUNG YEN UNIVERSITY OF TECHNOLOGY AND EDUCATION

# BIG PROJECT

## BIG DATA ANALYSIS

# YELP REVIEWS ANALYSIS

MAJOR: COMPUTER SCIENCE

STUDENT: **VU QUANG PHUC**
CLASS: **124221**
SUPERVISOR: **PhD NGUYEN VAN QUYET**

**HUNG YEN – 2025**

## COMMENTS

**Comments from supervisor:**

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

**SUPERVISOR**

**(**Signature and Full Name)

**COMMITMENT**

I solemnly declare that the project for the Big Data Analysis course, titled “Yelp Reviews Analysis” is the result of my independent work.

All references and materials use in this report have been properly cited in the References section.

All data, figures and results presented in this project are truthful and accurate.

If any information is found to be incorrect or plagiarized. I will take full responsibilities and accept my disciplinary actions imposed by the faculty and university

*Hung Yen November 1, 2025*

Student

*Phuc*

*Vu Quang Phuc*

**ACKNOWLEDGEMENT**

Completing this project, “Yelp Reviews Analysis” has been a challenging yet rewarding journey, and it would not have been possible without the invaluable guidance and support I have received along the way

First and foremost, I would like to express my deepest gratitude to the Department of Computer Science, Faculty of Information Technology – Hung Yen University of Technical Education. The resources, academic environment, and opportunities provided by the department have been instrumental in enabling me to carry out this project. I feel incredibly fortunate to have been part of a program that prioritizes academic excellence and hands-on learning.

I owe a special debt of gratitude to my mentor, PhD Nguyen Van Quyet, whose dedication, patience, and expertise have been a cornerstone of my progress. His thoughtful feedback and encouragement throughout the process not only enhanced the quality of this project but also deepened my understanding of advanced Python programming and data analysis. His ability to inspire confidence while challenging me to push my boundaries has left a lasting impact on my learning journey.

I would also like to thank all the faculty members of Hung Yen University of Technical Education. Their dedication to teaching and their tireless efforts to impart knowledge have equipped me with a strong foundation in computer science. The skills and insights I gained during their lectures and practical sessions have proven invaluable in overcoming the obstacles encountered in this project.

I acknowledge that, despite my best efforts, there may still be areas for improvement in this work. I warmly welcome constructive criticism and feedback from my professors, as I believe that every critique is an opportunity for growth.

Thank you for your encouragement and support throughout this journey!

# TABLE OF CONTENTS

GLOSSARY OF TERMS....4

LIST OF FIGURES....5

CHAPTER 1: INTRODUCTION....6

1.1 Yelp Review dataset....6

1.2 Why's this dataset?....6

1.3 Project pipeline....8

CHAPTER 2: BACKGROUND....9

2.1 Hadoop HDFS....9

2.2 MapReduce....10

2.3 Data Lake & Data Warehouse....11

2.4 Spark & Pyspark....11

2.5 Kafka....12

CHAPTER 3: IMPLEMENTATION....14

3.1 Dataset....14

3.2 Exploratory Data Analysis....16

3.3 Data Visualization....19

3.4 Data Preparation....25

3.5 Modeling....25

3.6 Actionable Insights....26

CONCLUSION....28

REFERENCES....29

# GLOSSARY OF TERMS

| Index | Term | Full form | Meaning |
|---|---|---|---|
| 1 | HDFS | Hadoop Distributed File System | |
| 2 | RDD | Resilient Distributed Dataset | |
| 3 | MLlib | Machine Learning Library (Spark MLlib) | |
| 4 | TF-IDF | Term Frequency–Inverse Document Frequency | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

# LIST OF FIGURES

Figure 1.1: Yelp Reviews Homepage....7
Figure 1.2: Yelp Reviews Dataset HomePage....8
Figure 1.3: Project pipeline....8
Figure 3.1: Yelp Business relational diagram....15
Figure 3.2: Yelp Business object in json format....15
Figure 3.3: Yelp Business samples in Spark Dataframe format....16
Figure 3.4: Yelp Review object in json format....16
Figure 3.5: Top 10 businesses by city....16
Figure 3.6: Top categories (explode categories string into array)....17
Figure 3.7: How many businesses are currently open?....17
Figure 3.8: What are the average stars and review counts for open businesses?....17
Figure 3.9: Which cities have the highest number of businesses?....18
Figure 3.10: What are the top 10 most common business categories?....18
Figure 3.11: What is the relationship between review count and rating?....19
Figure 3.12: Weekday with the highest check-ins....19
Figure 3.13: Longtitude and Lattitube range cover?....20
Figure 3.14: Top 15 Most Common Yelp Categories....20
Figure 3.15: What is the distribution of average_stars given by users?....21
Figure 3.16: Distribution of stars in reviews (how many 1–5 stars)?....21
Figure 3.17: Review length distribution (characters / words) and extreme lengths.....22
Figure 3.18: Time-series: check-in trends by month or year....22
Figure 3.19: Temporal trends: reviews per year/month....23
Figure 3.20: Most frequent words/bigrams in positive vs negative reviews....23
Figure 3.21: Busiest hours of day (aggregated across all businesses)....24
Figure 3.22: Peak hours by business category....24
Figure 3.23: Label creation and drop null text....25
Figure 3.24: Train/Test split....25
Figure 3.25: Creating modeling pipeline included preprocessing data....25
Figure 3.26: Evaluating result of Logistic Regression model....26

# CHAPTER 1: INTRODUCTION

## 1.1 Yelp Review dataset

Yelp is a globally recognized online platform that connects people with local businesses through user-generated reviews, ratings, and recommendations. Founded in San Francisco in 2004, Yelp has grown into one of the largest repositories of crowdsourced business information worldwide, covering restaurants, services, entertainment, and more. To promote academic research and innovation in data science, Yelp periodically releases a public subset of its internal data known as the Yelp Open Dataset for educational and non-commercial purposes. The current version of this dataset, maintained and distributed through the Yelp Dataset Challenge and Yelp Open Dataset Program, provides millions of real-world business reviews, user profiles, and location data across multiple countries.

This dataset is exceptionally valuable to data scientists and analysts because it offers a rare opportunity to work with large-scale, authentic, and heterogeneous data. It combines natural language text (reviews), structured numerical data (ratings, votes, and attributes), and geospatial information (business locations) enabling exploration across diverse domains such as sentiment analysis, recommendation systems, data mining, and big data processing. Moreover, because the Yelp data reflects real consumer behavior and market dynamics, it serves as a practical foundation for understanding user engagement, business performance, and the ethical implications of working with user-generated content in large-scale analytics.

## 1.2 Why's this dataset?

The Yelp Open Dataset is extremely valuable because it provides a large-scale, real-world, multi-modal dataset that includes text (user reviews), numerical ratings, business metadata (categories, location, attributes), and temporal & geospatial information. Because it contains millions of reviews across thousands of businesses, it allows data scientists to not only perform standard tasks like sentiment analysis and rating prediction, but also to explore more advanced solutions: topic modeling (to uncover what people talk about most),

recommendation systems (both collaborative & content-based), temporal trend analysis (how tastes / reviews evolve over time), geospatial analytics (compare cuisines / reviews by area), fraud / spam detection, and even image-based attribute inference (e.g. from photos or ambience tags).

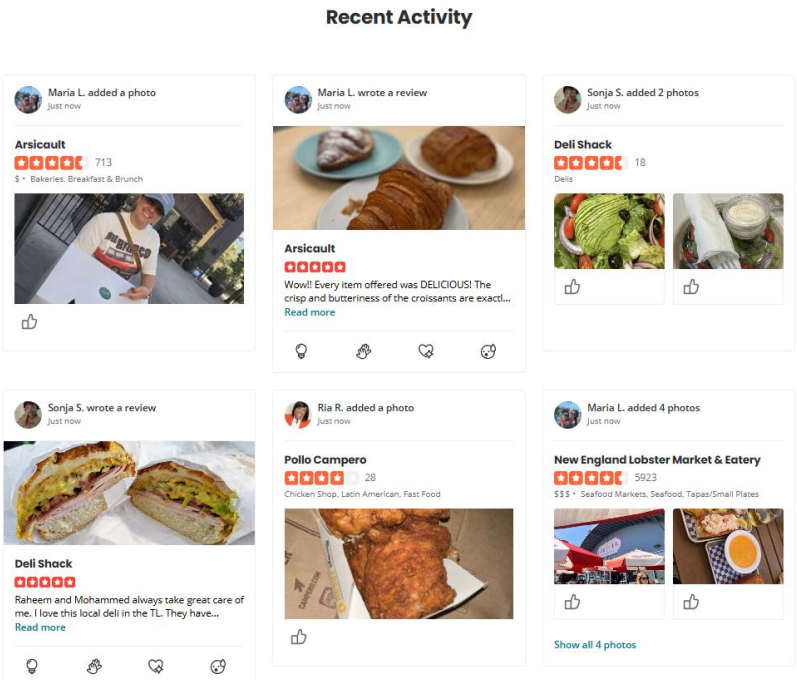


*Figure 1.1: Yelp Reviews Homepage*

Working with this dataset gives you multiple benefits. First, you gain experience handling big data: reading, cleaning, transforming JSON line files, managing memory, possibly using Spark or distributed processing, dealing with missing values, etc. Second, you improve on natural language processing skills: text preprocessing, feature extraction (TF-IDF, embeddings like Word2Vec or BERT), sentiment classification, maybe fine-tuning or transfer learning. Third, you develop modeling & recommendation skills: building and evaluating recommender systems, comparing content-based vs collaborative filtering, creating explainable recommendations. Fourth, you strengthen your analytical insight: by comparing different cuisine types, you'll learn to interpret review patterns, extract actionable business insight, see how customer sentiment links to ratings, identify pain points and strengths in operations. And finally, working with such a "real, imperfect" dataset boosts your problem-solving: dealing with noise, outliers, inconsistent data, biases, class imbalance, etc., which are all critical in real industrial / applied research settings.

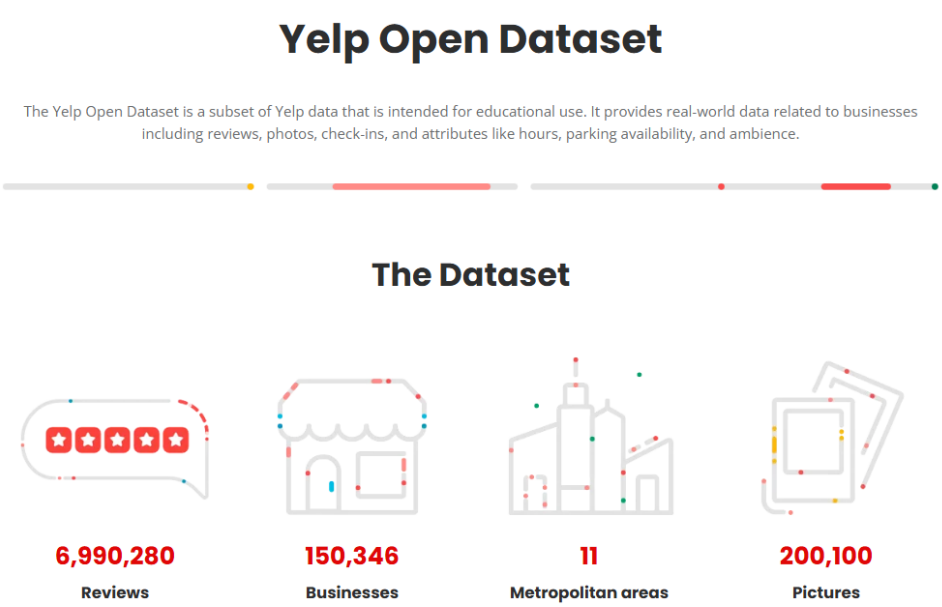


*Figure 1.2: Yelp Reviews Dataset HomePage*

## 1.3 Project pipeline

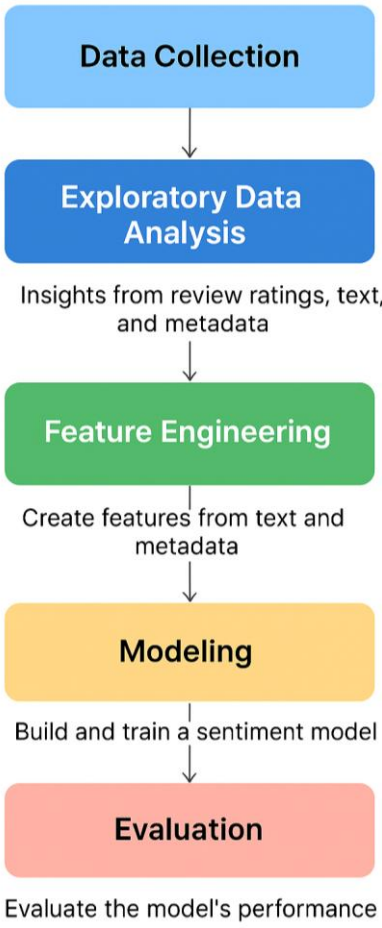


*Figure 1.3: Project pipeline*

# CHAPTER 2: BACKGROUND

## 2.1 Hadoop HDFS

Hadoop is an open-source framework for storing and processing large datasets across clusters of computers. Instead of using one powerful machine, it uses a collection of less expensive computers (commodity hardware) to store and process data in parallel, making it efficient for handling "big data". Key components include the Hadoop Distributed File System (HDFS) for storage and MapReduce for processing, along with YARN for resource management.

How it works:

- Distributed Storage: HDFS stores large files by splitting them into smaller blocks and distributing them across multiple computers in a cluster.
- Parallel Processing: It processes the data by breaking large workloads into smaller tasks that can be run simultaneously on different machines.
- Resource Management: YARN (Yet Another Resource Negotiator) manages and allocates cluster resources to these processing tasks.

Core components:

- Hadoop Distributed File System (HDFS): Provides high-throughput access to application data and scales to hundreds of nodes.
- MapReduce: A programming model for processing large datasets in parallel.
- YARN: Manages cluster resources and schedules jobs.
- Apache Hive: A data warehousing system built on top of Hadoop that provides an SQL-like query language.

Why it's used:

- It can handle massive amounts of structured, semi-structured, and unstructured data.
- It provides a cost-effective solution by using clusters of commodity hardware.
- It is highly fault-tolerant, designed to handle the failure of individual machines in the cluster.

## 2.2 MapReduce

MapReduce is a programming model and framework for processing large data sets in a parallel and distributed manner across a cluster of computers. It works by breaking down a big data task into two main phases: a "map" phase that processes data in parallel, and a "reduce" phase that aggregates the results of the map phase to produce a final output. This approach, famously associated with Apache Hadoop, enables massive scalability and fault tolerance for big data processing.

Map phase:

- Takes raw input data and splits it into smaller chunks.
- Processes each chunk independently and in parallel.
- Transforms the data into a set of key/value pairs.
- This is where initial filtering, sorting, and transformation occurs.

Reduce phase:

- Gathers the key/value pairs from the map phase.
- Groups pairs with the same key together.
- Performs a summary or aggregation operation on the grouped data.
- Produces the final output of the job.

Key features and benefits:

- Parallel processing: Breaks down large tasks into smaller, manageable pieces that are processed simultaneously across many machines, which significantly speeds up processing.
- Scalability: Can scale to handle massive amounts of data by distributing the workload across hundreds or thousands of servers in a cluster.
- Fault tolerance: If a node fails during a process, the system can automatically reassign the task to another node, ensuring the job completes reliably without manual intervention.
- Data locality: Tries to perform the processing on the same machine where the data is stored, which minimizes the amount of data that needs to be transferred over the network.

## 2.3 Data Lake & Data Warehouse

A data lake stores vast amounts of raw, unstructured data in its native format, making it flexible for future exploration and use cases like machine learning.

A data warehouse stores processed, structured data that is organized for specific analytics and reporting, providing a central repository for business intelligence and decision-making. Key differences include data type (raw vs. processed), schema (undefined vs. defined), and primary use case (exploration vs. reporting).

Data Lake

- Data Type: Stores all types of data, including raw, unstructured, and semi-structured data, in its native format.
- Schema: Uses a flat architecture and a "schema-on-read" approach, meaning the structure is applied when the data is retrieved, not when it's stored.
- Purpose: A flexible, cost-effective storage solution for data scientists and analysts to explore, test, and prepare data for future projects.
- Agility: Highly agile, as it's easy to add new data without needing to pre-define its structure.

Data Warehouse

- Data Type: Stores clean, structured data, often transformed through an Extract, Transform, Load (ETL) process.
- Schema: Uses a defined, structured schema (schema-on-write) before data is loaded, which optimizes it for queries and reporting.
- Purpose: A central repository for business intelligence (BI), analytics, and reporting, providing consistent and reliable data for decision-making.
- Agility: Less agile due to its rigid, structured nature. Changing the structure requires re-engineering the processes tied to it.

## 2.4 Spark & Pyspark

Apache Spark is an open-source, distributed computing framework designed for large-scale data processing and analytics. It provides a unified engine for various big data tasks, including batch processing, real-time streaming, SQL analytics, and machine learning.

Spark is known for its speed, scalability, and ability to handle diverse data workloads efficiently across clusters of computers.

PySpark is the Python API for Apache Spark. It enables Python developers to interact with Spark and leverage its distributed computing capabilities using familiar Python syntax and libraries. PySpark acts as a bridge between Python and the Spark engine (which is written in Scala), allowing users to write Spark applications in Python and process large-scale data in a distributed environment.

Key aspects of PySpark:

- Pythonic Interface: Provides a Python-friendly way to access Spark's features, making it accessible to a wide range of data professionals and developers.
- Distributed Data Processing: Allows you to work with large datasets by distributing computations across a cluster of machines, enabling efficient processing of big data.
- Integration with Python Ecosystem: Seamlessly integrates with popular Python libraries like pandas, NumPy, and scikit-learn, enhancing data manipulation and machine learning capabilities.
- Spark DataFrames: Utilizes Spark DataFrames as a key data type, providing a tabular, distributed data structure for structured data processing, similar to pandas DataFrames but designed for distributed environments.
- Access to Spark Features: Supports all of Spark's core features, including Spark SQL for structured data, Structured Streaming for real-time data, and MLlib for machine learning.

## 2.5 Kafka

Apache Kafka is a distributed data store optimized for ingesting and processing streaming data in real-time. Streaming data is data that is continuously generated by thousands of data sources, which typically send the data records in simultaneously. A streaming platform needs to handle this constant influx of data, and process the data sequentially and incrementally.

Kafka provides three main functions to its users:

- Publish and subscribe to streams of records
- Effectively store streams of records in the order in which records were generated
- Process streams of records in real time
- Kafka is primarily used to build real-time streaming data pipelines and applications that adapt to the data streams. It combines messaging, storage, and stream processing to allow storage and analysis of both historical and real-time data.

What is Kafka used for?

- Kafka is used to build real-time streaming data pipelines and real-time streaming applications. A data pipeline reliably processes and moves data from one system to another, and a streaming application is an application that consumes streams of data. For example, if you want to create a data pipeline that takes in user activity data to track how people use your website in real-time, Kafka would be used to ingest and store streaming data while serving reads for the applications powering the data pipeline. Kafka is also often used as a message broker solution, which is a platform that processes and mediates communication between two applications.

What are the benefits of Kafka's approach?

- Scalable: Kafka’s partitioned log model allows data to be distributed across multiple servers, making it scalable beyond what would fit on a single server.
- Fast: Kafka decouples data streams so there is very low latency, making it extremely fast.
- Durable: Partitions are distributed and replicated across many servers, and the data is all written to disk. This helps protect against server failure, making the data very fault-tolerant and durable.

# CHAPTER 3: IMPLEMENTATION

## 3.1 Dataset

The relational structure of the Yelp Dataset is centered around the Business entity, which serves as the core of the analytical model. Each business, uniquely identified by a business_id, contains essential metadata such as name, address, city, state, average rating (stars), review count, and categorical tags describing the nature of the business. The User entity represents individual Yelp users, identified by user_id, and stores personal metrics such as the number of reviews written, date of account creation, average rating behavior, and engagement features like compliments and fan counts. The Review table forms the critical bridge between users and businesses, where each review links a single user to a specific business and contains textual feedback, a star rating, and engagement attributes (useful, funny, cool). Complementing these, the Check-in entity records temporal user interactions, providing timestamps of visits that can be aggregated to study temporal trends in business activity. In addition, auxiliary entities such as Business_Category and Tip further enrich the schema by detailing business classifications and short user-generated advice, respectively.

Quantitatively, the Yelp Dataset encompasses approximately 209,000 businesses, 1.98 million users, and 8.6 million reviews, making it a rich and complex environment for large-scale data analysis. On average, each business is associated with about 41 reviews, while each user contributes around four reviews, illustrating a highly skewed participation distribution with a small proportion of users generating the majority of the content. Furthermore, each business records roughly nine check-ins, highlighting the uneven engagement patterns across categories and locations. The schema's many-to-one relationships—particularly between businesses and reviews, and users and reviews—form the foundation for exploring customer sentiment, behavioral patterns, and business performance dynamics.

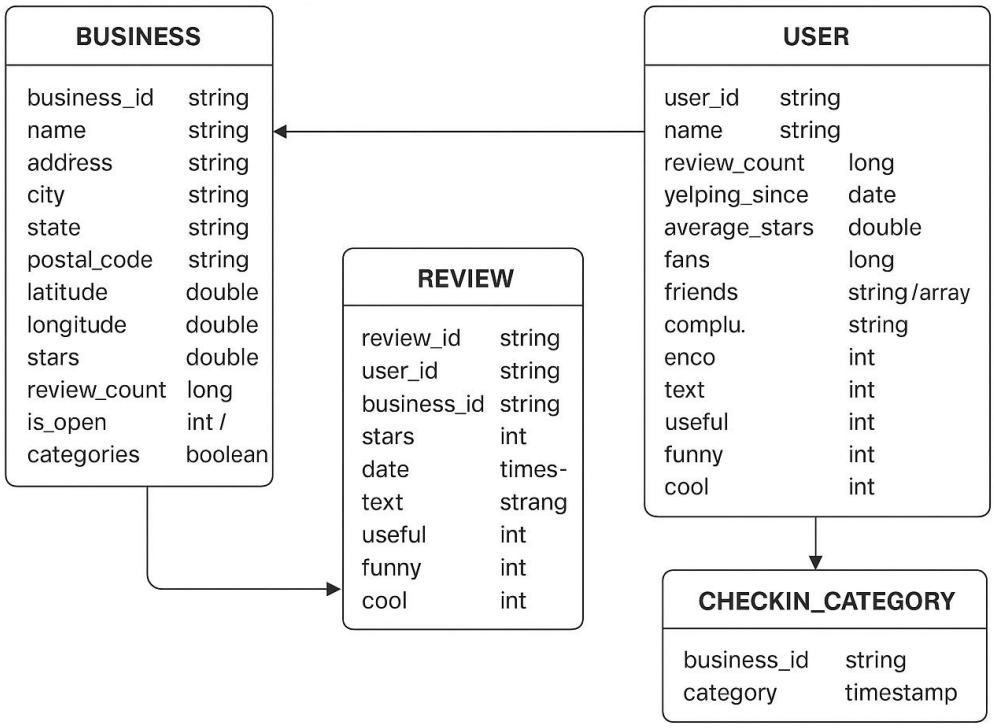


*Figure 3.1: Yelp Business relational diagram*

This is the Yelp Business object saved in the JSON format in *yelp_academic_dataset_business.json* file.

```
{
  "business_id": "1SWheh84yJXfytovILXOAQ",                    **main_objects**
  "name": "The Range at Lake Norman",
  "address": "10913 Bailey Rd",
  "city": "Cornelius",
  "state": "NC",
  "postal_code": "28031",
  "latitude": 35.4627242,
  "longitude": -80.8526119,
  "stars": 4.0,
  "review_count": 36,
  "is_open": 1,
  "categories": "Active Life, Gun/Rifle Ranges, Guns & Ammo",

  "attributes": {                                              **nested oject -> analysis later**
    "BusinessAcceptsCreditCards": "True",
    "WiFi": "free"
  },
  "hours": {                                                   **nested oject -> analysis later**
    "Monday": "10:00-18:00",
    "Tuesday": "10:00-18:00"
  }
}
```

*Figure 3.2: Yelp Business object in json format*

```
+--------------------+--------------------+--------------------+--------------+-----+-----------+----------+------------+-----+------------+-------+--------------------+
|         business_id|                name|             address|          city|state|postal_code|  latitude|   longitude|stars|review_count|is_open|          categories|
+--------------------+--------------------+--------------------+--------------+-----+-----------+----------+------------+-----+------------+-------+--------------------+
|Pns214eNsfO8kk83d...|Abby Rappoport, L...|1616 Chapala St, ...|Santa Barbara|   CA|      93101|34.4266787|-119.7111968|  5.0|           7|      0|Doctors, Traditio...|
|mpf3x-BjTdTEA3yCZ...|       The UPS Store|87 Grasso Plaza S...|       Affton|   MO|      63123| 38.551126|  -90.335695|  3.0|          15|      1|Shipping Centers,...|
|tUFrWirKiKi_TAnsV...|              Target|5255 E Broadway Blvd|       Tucson|   AZ|      85711| 32.223236| -110.880452|  3.5|          22|      0|Department Stores...|
|MTSW4McQd7CbVtyjq...|  St Honore Pastries|         935 Race St| Philadelphia|   PA|      19107|39.9555052| -75.1555641|  4.0|          80|      1|Restaurants, Food...|
|mWMc6_wTdE0EUBKIG...|Perkiomen Valley ...|       101 Walnut St|   Green Lane|   PA|      18054|40.3381827| -75.4716585|  4.5|          13|      1|Brewpubs, Breweri...|
+--------------------+--------------------+--------------------+--------------+-----+-----------+----------+------------+-----+------------+-------+--------------------+
only showing top 5 rows
```

*Figure 3.3: Yelp Business samples in Spark Dataframe format*


```
{
  "review_id": "xQY8N_XvtGbearJ5X0KlyQ",
  "user_id": "OwjRMXRC0KyPrIlcjYv4-A",
  "business_id": "f9NumwFMBDn751xgFiRbNA",
  "stars": 4,
  "date": "2016-03-09",
  "text": "Great food, friendly staff...",
  "useful": 2,
  "funny": 0,
  "cool": 1
}
```


*Figure 3.4: Yelp Review object in json format*

## 3.2 Exploratory Data Analysis

Top 10 businesses by city

```
1  city_counts = (
2      df_business.groupBy("city")
3      .agg(F.countDistinct("business_id").alias("num_businesses"))
4      .orderBy(F.desc("num_businesses"))
5  )
6  city_counts.show(10, truncate=False)
7
```

```
+-------------+--------------+
|city         |num_businesses|
+-------------+--------------+
|Philadelphia |14569         |
|Tucson       |9250          |
|Tampa        |9050          |
|Indianapolis |7540          |
|Nashville    |6971          |
|New Orleans  |6209          |
|Reno         |5935          |
|Edmonton     |5054          |
|Saint Louis  |4827          |
|Santa Barbara|3829          |
+-------------+--------------+
only showing top 10 rows
```

*Figure 3.5: Top 10 businesses by city*

Top categories (explode categories string into array)

```
df_business = df_business.withColumn("categories_array", F.split(F.col("categories"), ",\\s*"))
df_business.select(F.explode(F.col("categories_array")).alias("category")) \
    .groupBy("category").count().orderBy(F.desc("count")).show(15, truncate=False)
```

```
+-------------------------+-----+
|category                 |count|
+-------------------------+-----+
|Restaurants              |52268|
|Food                     |27781|
|Shopping                 |24395|
|Home Services            |14356|
|Beauty & Spas            |14292|
|Nightlife                |12281|
|Health & Medical         |11890|
|Local Services           |11198|
|Bars                     |11065|
|Automotive               |10773|
|Event Planning & Services|9895 |
|Sandwiches               |8366 |
|American (Traditional)   |8139 |
|Active Life              |7687 |
|Pizza                    |7093 |
+-------------------------+-----+
only showing top 15 rows
```

*Figure 3.6: Top categories (explode categories string into array)*

[?] How many businesses are currently open?

**Insight:**

- *Understand how many businesses are still operating — an indicator of data freshness and business survival rate.*
- *If most businesses have is_open = 1, it means Yelp's dataset is still relevant and updated.*

```
from pyspark.sql import functions as F

df_business.groupBy("is_open").count().show()
```

```
+-------+------+
|is_open| count|
+-------+------+
|      0| 30648|
|      1|119698|
+-------+------+
```

*Figure 3.7: How many businesses are currently open?*

[?] What are the average stars and review counts for open businesses?

**Purpose:**

- *Compare customer satisfaction among active businesses.*

**Insight:**

- *Open businesses often have higher ratings because poorly rated ones tend to close — a natural selection effect.*

```
df_business.filter(F.col("is_open") == 1).select(
    F.avg("stars").alias("avg_stars_open"),
    F.avg("review_count").alias("avg_reviews_open")
).show()

```

```
+-----------------+-----------------+
|   avg_stars_open| avg_reviews_open|
+-----------------+-----------------+
|3.618907584086618|46.68396297348327|
+-----------------+-----------------+
```

*Figure 3.8: What are the average stars and review counts for open businesses?*

[?] Which cities have the highest number of businesses?

**Purpose:**

- *Identify major business hubs and Yelp coverage areas.*

**Insight:**

- *Cities like Las Vegas, Phoenix, Toronto, and Charlotte usually dominate — showing where Yelp is most active.*

```
df_business.groupBy("city")\
    .agg(F.count("business_id").alias("num_businesses"))\
    .orderBy(F.desc("num_businesses"))\
    .show(15, truncate=False)
```

```
+----------------+--------------+
|city            |num_businesses|
+----------------+--------------+
|Philadelphia    |14569         |
|Tucson          |9250          |
|Tampa           |9050          |
|Indianapolis    |7540          |
|Nashville       |6971          |
|New Orleans     |6209          |
|Reno            |5935          |
|Edmonton        |5054          |
|Saint Louis     |4827          |
|Santa Barbara   |3829          |
|Boise           |2937          |
|Clearwater      |2221          |
|Saint Petersburg|1663          |
|Metairie        |1643          |
|Sparks          |1624          |
+----------------+--------------+
only showing top 15 rows
```

*Figure 3.9: Which cities have the highest number of businesses?*

[?] What are the top 10 most common business categories?

**Purpose:**

- *Discover which business types dominate Yelp.*

**Insight:**

- *Expect to see "Restaurants", "Shopping", and "Beauty & Spas" — gives a macro-view of Yelp's domain coverage.*

```
df_exploded = df_business.withColumn("category", F.explode(F.split(F.col("categories"), ",\\s*")))

df_exploded.groupBy("category")\
    .agg(F.count("business_id").alias("num_businesses"))\
    .orderBy(F.desc("num_businesses"))\
    .show(15, truncate=False)
```

```
+------------------------+--------------+
|category                |num_businesses|
+------------------------+--------------+
|Restaurants             |52268         |
|Food                    |27781         |
|Shopping                |24395         |
|Home Services           |14356         |
|Beauty & Spas           |14292         |
|Nightlife               |12281         |
|Health & Medical        |11890         |
|Local Services          |11198         |
|Bars                    |11065         |
|Automotive              |10773         |
|Event Planning & Services|9895         |
|Sandwiches              |8366          |
|American (Traditional)  |8139          |
|Active Life             |7687          |
|Pizza                   |7093          |
+------------------------+--------------+
only showing top 15 rows
```

*Figure 3.10: What are the top 10 most common business categories?*

## 3.3 Data Visualization

**Purpose**

- *Check whether highly rated businesses also have more reviews.*

**Insight**

- *Businesses with many reviews tend to have more stable ratings — fewer extremes due to averaging effects.*

```
sample_df = df_business.select("stars", "review_count").sample(False, 0.01, 42).toPandas()

import seaborn as sns
import matplotlib.pyplot as plt

sns.scatterplot(data=sample_df, x="review_count", y="stars", alpha=0.5)
plt.xlim(0, 500)
plt.title("Relationship between Review Count and Star Rating")
plt.show()
```

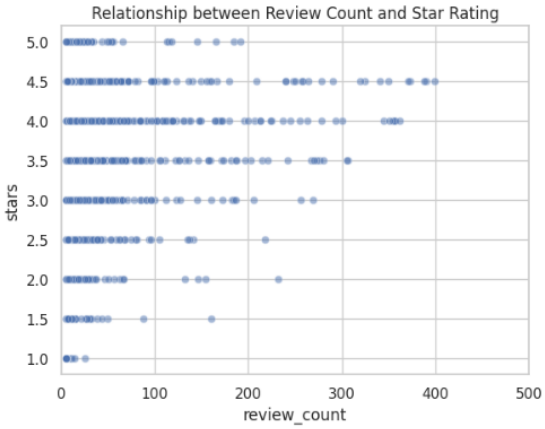


*Figure 3.11: What is the relationship between review count and rating?*

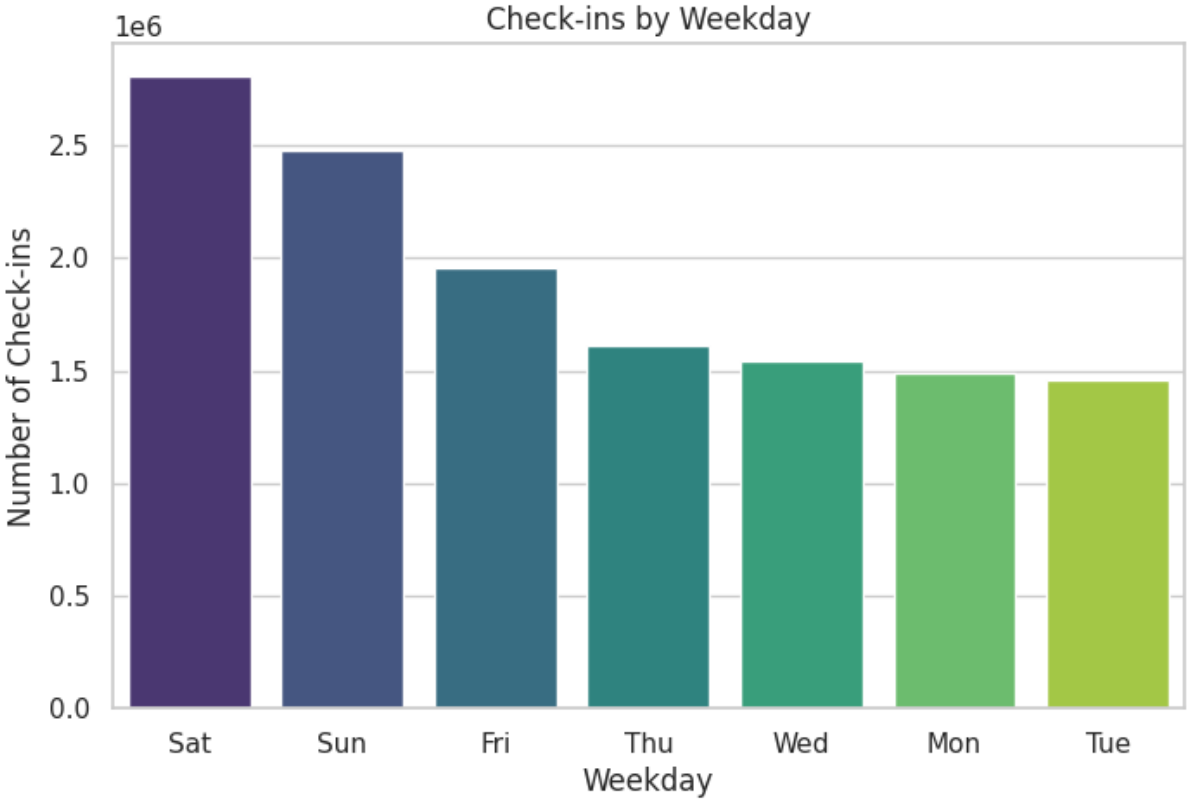


*Figure 3.12: Weekday with the highest check-ins*

Longtitude and Lattitube range cover?

```
from pyspark.sql import functions as F

df_business.select(
    F.min("latitude").alias("min_lat"),
    F.max("latitude").alias("max_lat"),
    F.min("longitude").alias("min_lon"),
    F.max("longitude").alias("max_lon")
).show()
```

```
+---------+----------+-----------+--------------+
|  min_lat|   max_lat|    min_lon|       max_lon|
+---------+----------+-----------+--------------+
|27.555127|53.6791969|-120.095137|-73.2004570502|
+---------+----------+-----------+--------------+
```

*Latitude range 25–56 → roughly covers southern US to northern UK*

*Longitude range -124–8 → western US to western Europe*

*Asia (longitudes 60–150) and Africa (latitudes -35–35, longitudes -20–55) are completely absent.*

**Yelp's market presence**

- *Yelp primarily operates in the US, Canada, and a few EU countries.*

**Asia's local markets are dominated by other platforms:**

-* CN Dianping / Meituan*

- JP Tabelog / Gurunavi*
- VN *Foody.vn*
- KR *Naver Map / KakaoMap*

*-SG Burpple / HungryGoWhere*

**Privacy and data protection**

- *European GDPR and other local laws (like PDPA in Asia) restrict sharing of user review data, especially if it can be traced back to individuals.*

**Legal licensing**

- *Yelp only granted data use for "academic and educational purposes" in a limited subset of locations.&*

*Figure 3.13: Longtitude and Lattitube range cover?*

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

pdf_top = top_categories.limit(15).toPandas()

plt.figure(figsize=(10,6))
sns.barplot(data=pdf_top, y="category", x="num_businesses", orient="h", palette="viridis")
plt.title("Top 15 Most Common Yelp Categories")
plt.xlabel("Number of Businesses")
plt.ylabel("Category")
plt.tight_layout()
plt.show()
```

```
/tmp/ipykernel_4142/1431418309.py:7: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same effect.

  sns.barplot(data=pdf_top, y="category", x="num_businesses", orient="h", palette="viridis")
```

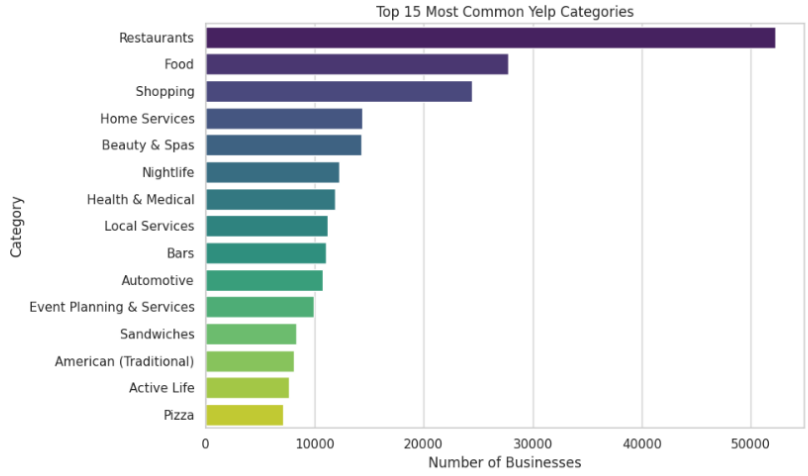


*Figure 3.14: Top 15 Most Common Yelp Categories*

What is the distribution of average_stars given by users?

```
user_star_stats = df_user.select("average_stars").toPandas()
sns.histplot(user_star_stats["average_stars"], bins=40, kde=True)
plt.title("Distribution of Users' Average Ratings")
plt.xlabel("Average Stars")
plt.ylabel("Number of Users")
plt.show()
```

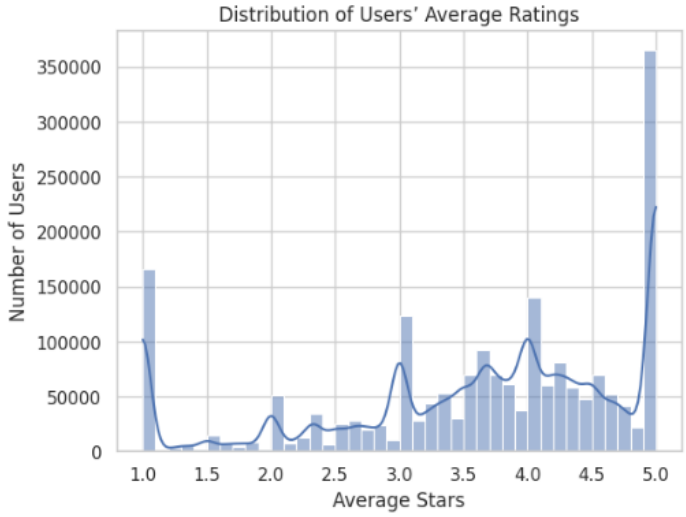


*Figure 3.15: What is the distribution of average_stars given by users?*

```
from pyspark.sql import functions as F
import matplotlib.pyplot as plt
import seaborn as sns

# Count by star rating
star_dist = (
    df_reviews.groupBy("stars")
    .agg(F.count("*").alias("count"))
    .orderBy("stars")
    .toPandas()
)

plt.figure(figsize=(7,5))
sns.barplot(data=star_dist, x="stars", y="count", palette="viridis")
plt.title("Distribution of Yelp Star Ratings")
plt.xlabel("Stars")
plt.ylabel("Number of Reviews")
plt.show()
```

```
/tmp/ipykernel_4142/2367776542.py:14: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

  sns.barplot(data=star_dist, x="stars", y="count", palette="viridis")
```

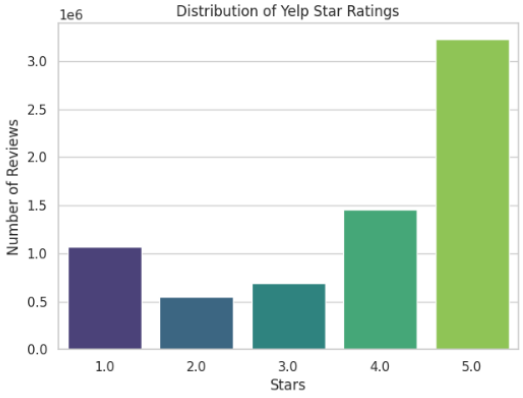


*Figure 3.16: Distribution of stars in reviews (how many 1–5 stars)?*

```
df_len = (
    df_reviews.withColumn("char_len", F.length("text"))
              .withColumn("word_len", F.size(F.split(F.col("text"), " ")))
)

# Summary stats
df_len.select(F.mean("char_len"), F.mean("word_len"),
              F.max("char_len"), F.max("word_len")).show()

# For plotting (sample down)
pdf_len = df_len.sample(False, 0.002, seed=42).select("char_len", "word_len").toPandas()

plt.figure(figsize=(8,5))
sns.histplot(pdf_len["word_len"], bins=100, kde=True)
plt.title("Distribution of Review Length (in Words)")
plt.xlabel("Words per Review")
plt.ylabel("Frequency")
plt.show()
```

```
+-----------------+------------------+-------------+-------------+
|  avg(char_len)|     avg(word_len)|max(char_len)|max(word_len)|
+-----------------+------------------+-------------+-------------+
|567.7644364746477|105.79662159455701|         5000|         3079|
+-----------------+------------------+-------------+-------------+
```

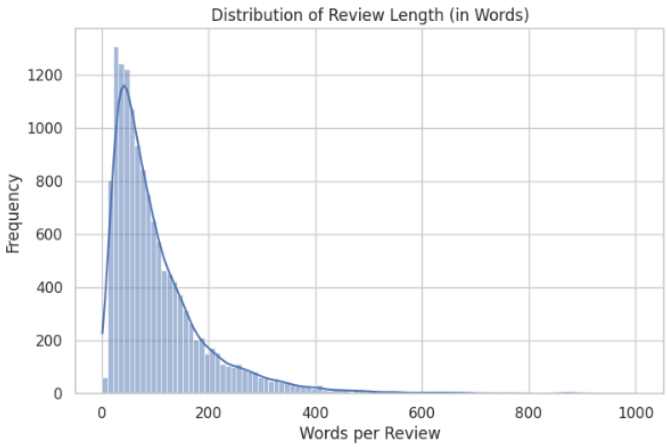


*Figure 3.17: Review length distribution (characters / words) and extreme lengths.*

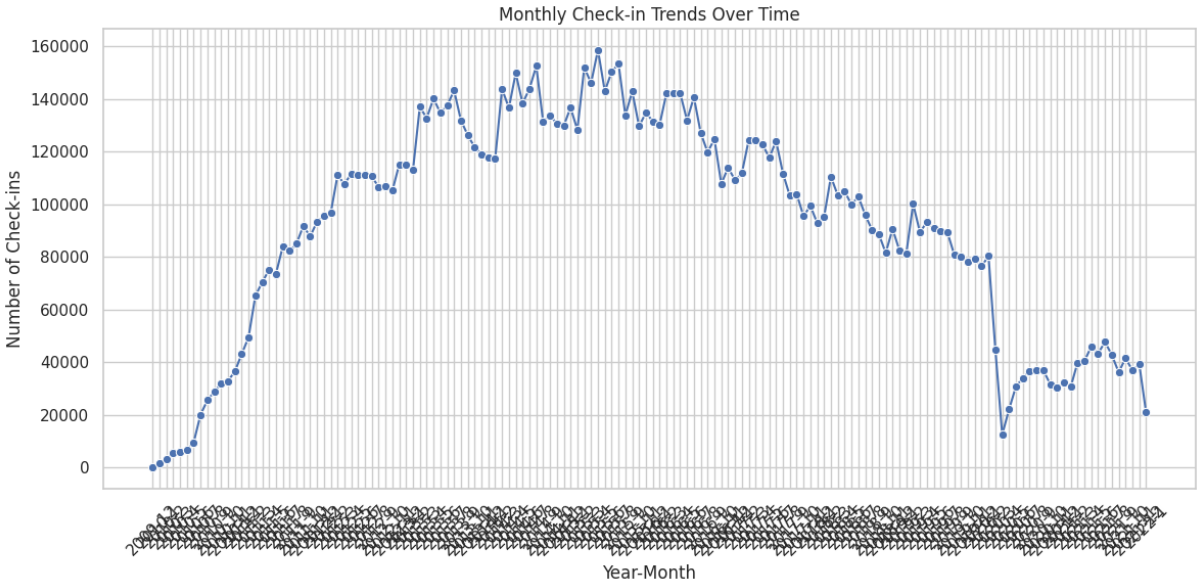


*Figure 3.18: Time-series: check-in trends by month or year*

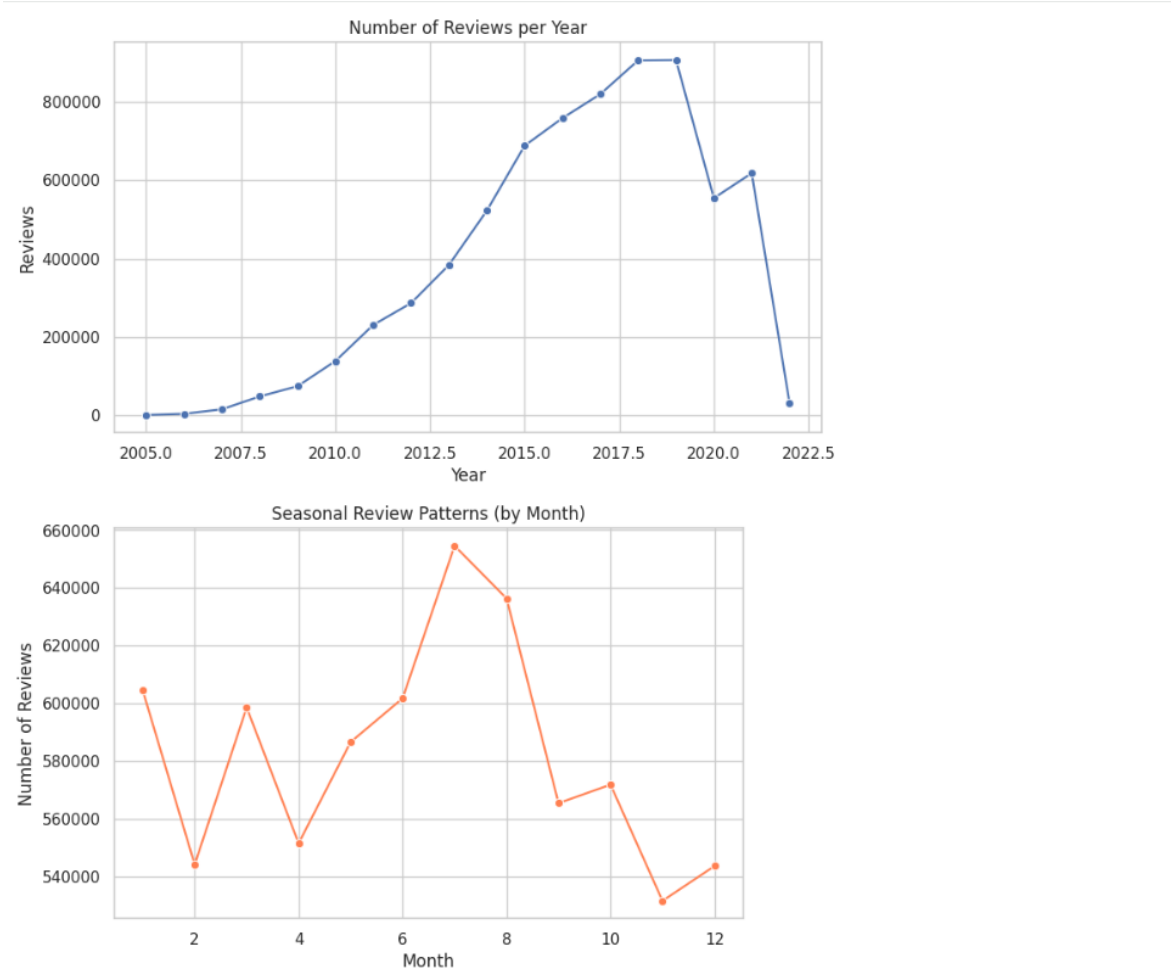


*Figure 3.19: Temporal trends: reviews per year/month*

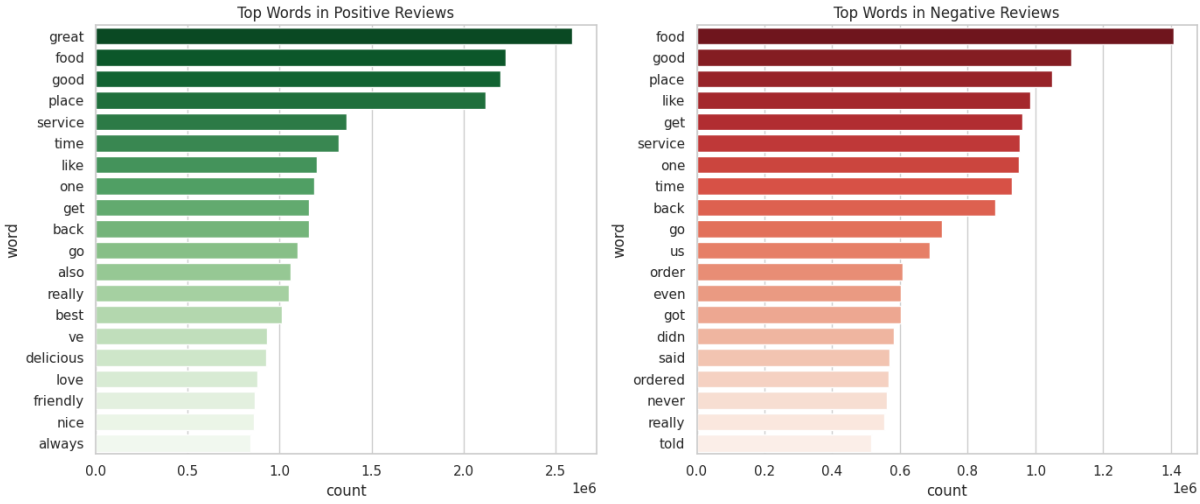


*Figure 3.20: Most frequent words/bigrams in positive vs negative reviews*

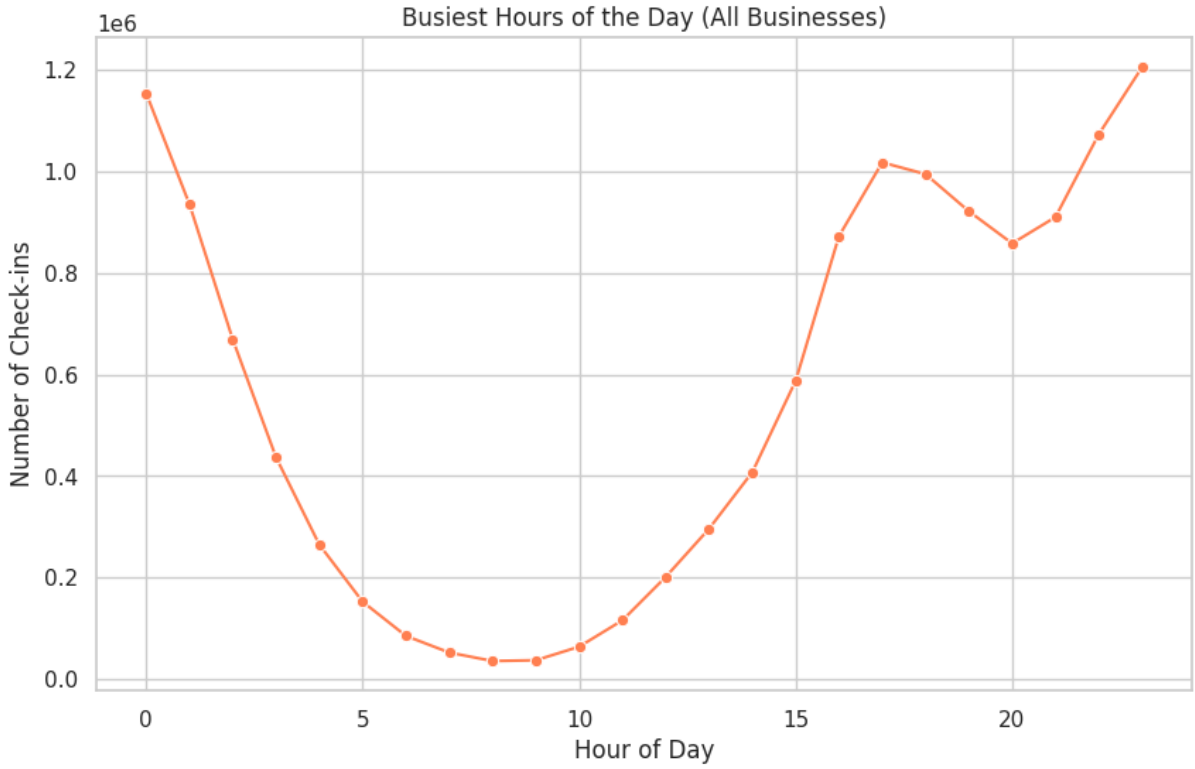


*Figure 3.21: Busiest hours of day (aggregated across all businesses)*

- Peak hours by business category
  - *Restaurants/Cafes: Noon & evening peaks*
  - *Bars/Nightlife: Late evening peaks*
  - *Gyms/Studios: Morning rush (6–9 AM)*

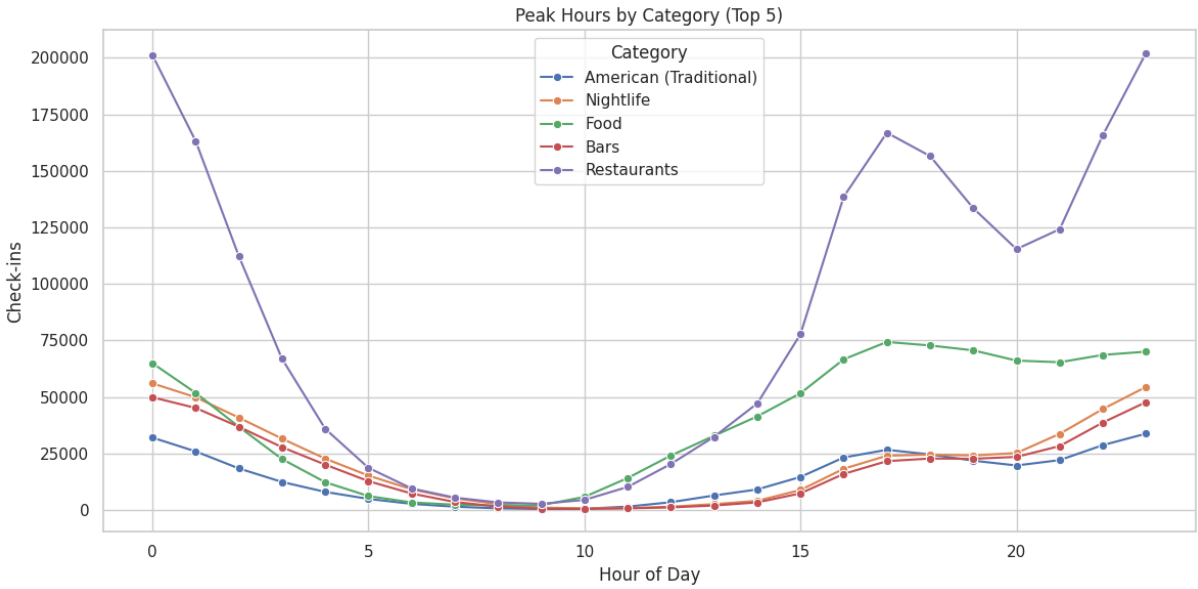


*Figure 3.22: Peak hours by business category*

## 3.4 Data Preparation

- Label creation (positive/negative) base on the rating
- Drop Null

Label Creation

```
# Create binary label: positive (1) if stars >= 4, else negative (0). Adjust threshold as desired.
df_reviews_labeled = df_reviews.withColumn('label', F.when(F.col('stars') >= 4, 1.0).otherwise(0.0))
```

```
recol_reviews.append("label")

df_reviews_labeled.select(recol_reviews).limit(4).show()
```

```
+--------------------+--------------------+--------------------+-------------------+--------------------+----+-----+-----+------+-----+
|           review_id|         business_id|             user_id|               date|                text|cool|funny|stars|useful|label|
+--------------------+--------------------+--------------------+-------------------+--------------------+----+-----+-----+------+-----+
|KU_O5udG6zpxOg-Vc...|XQfwVwDr-v0ZS3_Cb...|mh_-eMZ6K5RLWhZyI...|2018-07-07 22:09:11|If you decide to ...|   0|    0|  3.0|     0|  0.0|
|BiTunyQ73aT9WBnpR...|7ATYjTIgM3jUlt4UM...|OyoGAe7OKpv6SyGZT...|2012-01-03 15:28:18|I've taken a lot ...|   1|    0|  5.0|     1|  1.0|
|saUsX_uimxRlCVr67...|YjUWPpI6HXG5301wP...|8g_iMtfSiwikVnbP2...|2014-02-05 20:30:30|Family diner. Had...|   0|    0|  3.0|     0|  0.0|
|AqPFMleE6RsU23_au...|kxX2SOes4o-D3ZQBk...|_7bHUi9Uuf5__HHc_...|2015-01-04 00:01:03|Wow!  Yummy, diff...|   1|    0|  5.0|     1|  1.0|
+--------------------+--------------------+--------------------+-------------------+--------------------+----+-----+-----+------+-----+
```

```
# Drop null texts
df_reviews_labeled_dnull = df_reviews_labeled.filter(F.col('text').isNotNull())
print('Rows after filtering null text:', df_reviews_labeled_dnull.count())
# ----------------------------------------------------------------------
```

```
Rows after filtering null text: 6990280
```

- *Rows after filtering null text: 6990280*

```
Start coding or generate with AI.
```

*Figure 3.23: Label creation and drop null text*

- Train/Test split

Train/Test Split

```
train_df, test_df = df_reviews_labeled_dnull.randomSplit([0.8, 0.2], seed=42)
print('Train rows:', train_df.count(), 'Test rows:', test_df.count())
```

```
Train rows: 5591048 Test rows: 1399232
```

*Figure 3.24: Train/Test split*

## 3.5 Modeling

Modeling

```
from pyspark.ml.classification import LogisticRegression
from pyspark.ml.evaluation import BinaryClassificationEvaluator, MulticlassClassificationEvaluator

from pyspark.ml.classification import LogisticRegression
from pyspark.ml import Pipeline
from pyspark.ml.feature import RegexTokenizer, StopWordsRemover, HashingTF, IDF
```

```

train_df_sample = train_df.sample(False, 0.005, seed = 16)

tokenizer = RegexTokenizer(inputCol="text", outputCol="words", pattern="\\W+")
remover = StopWordsRemover(inputCol="words", outputCol="filtered")
tf = HashingTF(inputCol="filtered", outputCol="rawFeatures", numFeatures=1000)
idf = IDF(inputCol="rawFeatures", outputCol="features")
lr = LogisticRegression(maxIter=10, regParam=0.3, elasticNetParam=0.8)

pipeline = Pipeline(stages=[tokenizer, remover, tf, idf, lr])
model = pipeline.fit(train_df_sample)
model

```

```
25/10/31 07:14:30 WARN InstanceBuilder: Failed to load implementation from:dev.ludovic.netlib.blas.JNIBLAS
PipelineModel_99f8cb5a4fbd
```

*Figure 3.25: Creating modeling pipeline included preprocessing data*

```
# Evaluate
bce = BinaryClassificationEvaluator(labelCol='label', rawPredictionCol='rawPrediction', metricName='areaUnderROC')
auc = bce.evaluate(pred_lr)
print('Logistic Regression AUC:', auc)
```

```
Logistic Regression AUC: 0.5
```

```
mce = MulticlassClassificationEvaluator(labelCol='label', predictionCol='prediction', metricName='accuracy')
acc = mce.evaluate(pred_lr)
print('Accuracy:', acc)
```

```
Accuracy: 0.6707372329963865
25/10/31 14:33:01 ERROR TaskSchedulerImpl: Lost executor 1 on 192.168.56.103: worker lost: Not receiving heartbeat for 60 seconds
```

*Figure 3.26: Evaluating result of Logistic Regression model*

## 3.6 Actionable Insights

- In general, We found out that for most restaurant types, delicious ranks first among all positive words, indicating that tastes might weight more than other factors like service and price when people are judging a restaurant. For most cuisine types, the word friendly rank first before the word reasonable, which means the friendly service is more likely to be the reason for the high score rather than reasonable price. It could also be observed that when it comes to the flavor of food, customers value freshness more than tastiness.
- Different characteristics are also shown for different restaurant categories. Vietnamese and Italian food received positive feedback because of freshness, while French restaurants received positive reviews for their sweet food. However, sweet food is the reason for Korean restaurants to have negative reviews. Korean, Japanese, Chinese, and Thai have positive reviews mainly for their friendly service, especially for Korean restaurants, since attentive ranks third. The variety of food is also the reason of high score for Korean, Japanese and Thai cuisine types. Fun and creative are special charateristics for Japanese restaurants. For Italian cuisine type, customers prefer classic Italian food. The reason of high score in French cuisine type is related to the romantic and beautiful appearences or environment.
- From the negative word list, we could observe that bland is one of the main problems for Korean, Thai and Vietnamese restaurants, which means customers expect food of those three cuisine type should be spicy. For French, Italian and Japanese cuisine types of restaurants, it is likely to have the low score because the food is cold. The low score of Japanese cuisine type is also due to the dark and crowded environment. Sour is one of the main problems

for Chinese cuisine type. Slow service is the main negative characteristic for Korean and French. French cuisine type receive negative reviews also for the expensive price. Thai receive negative reviews mainly for greasy food.

- Since our analysis may help to extract specific features from any set of reviews, restaurant owners can make good use of it for essential information once they received a certain amount of Yelp reviews. From those reviews they can understand why customers love or dislike their restaurants, maybe great reviews primarily due to fresh food, or perhaps unsatisfied reviews caused by too high price. Meanwhile they can also compare the restaurant with similar restaurants within the same type.

# CONCLUSION

Through the completion of this project, I have gained a comprehensive understanding of how large-scale datasets can be processed, analyzed, and modeled using modern big data technologies. By working with the Yelp Dataset, I learned to manage complex data structures distributed across multiple files such as business information, user profiles, reviews, and check-ins while developing a coherent relational schema that supports scalable analytics. Using PySpark, I explored how distributed computing enables efficient handling of massive data volumes that traditional tools like Pandas could not manage.

The project strengthened my skills in data cleaning, exploratory data analysis (EDA), and feature engineering, helping me uncover meaningful insights about customer behavior, business performance, and temporal engagement trends. I also applied machine learning techniques such as sentiment analysis using Spark MLlib, which provided practical exposure to building and evaluating predictive models on large text datasets. Additionally, analyzing check-in and review data helped me understand how user interactions can be translated into actionable insights for business recommendation and improvement.

Overall, this project enhanced my ability to design end-to-end big data analytics pipelines from ingestion and preprocessing to modeling and visualization. It not only deepened my technical expertise in distributed data processing but also improved my analytical thinking, data interpretation, and problem-solving skills within real-world data environments.

**Achievements:**

- The project has significantly improved the early detection and prediction of

**Weaknesses:**

- Data Limitations: The quality and availability of data remain a challenge,
- populations or regions, leading to concerns about their generalizability.

**Future Enhancements:**

- Data Enhancement: Efforts should be made to acquire and integrate diverse datasets, including genetic data, biomarkers, and longitudinal studies, to improve

# REFERENCES

[1] Dataset link: *Open Dataset | Yelp Data Licensing*

[2] Documents and slides of mentor Nguyen Van Quyet