# Analysis of Yelp dataset

Team "The Data Analytics Group"

Akshay Karki (avk1063@rit.edu)
Gaurav Gawade (gdg6776@rit.edu)
Shristika Yadav (sy2109@rit.edu)

## ABSTRACT

Yelp ratings are often seen as a respectable metric for businesses. But, how can we better something which is already good? Wouldn't it be cool if we could predict the number of stars a business would get just by looking at the number of stars it gets on each review? Or we could predict the busiest hours a business should expect based on the frequency of people? These are some of the questions that we hope to tackle with this project.

## 1. INTRODUCTION

The dataset contains about 1.6M records. The dataset is spread over multiple JSON files and the data in all the files is interlinked. The task involved in this project will be to parse the JSON files using Python and Pandas library. The parsed data will then be stored in a dictionary and then further used for visualizing trends and analyzing patterns.

There is a review file where multiple users have reviewed a business by giving it some ratings as well as comments. A business file consists of all the demographic information for that business while a user file gives information about the reviewer. A check-in file tells which time the business is usually the busiest at.

With the given dataset, some of the important questions which we will be able to answer are:

- How well a business is doing in terms of check-ins
- Is the business one of the best in its city
- Depending on user reviews what is the next business he will review
- What are the best businesses in a given category
- Depending on the review whether it is it a good review or a bad review.

.

## 2. IMPLEMENTATION

The implementation part of the project will mainly consist of three stages such as dataset collection/assembly, clean data and analyze, and data mining and visualization. Our main focus will mainly be on cleaning the data as much as possible in order to have optimum analysis result [2] [1]. The data mining algorithm that will be used will be KMeans and visualizations will be done using the Python libraries.

### 2.1 Tools used

In this section, we will consolidate the data collected from various sources for Yelp [3]. Given the dataset and how it could be analyzed, we used Python to code. We also used Python libraries like SciPy, Matplotlib, and Pandas for data visualization.

We used Weka for clustering purposes, but given the size of the dataset, it took a long time to load the dataset and perform clustering on the same. So, we switched over to Python and wrote the code to visualize our data. We used Weka only to run the KMeans algorithm since it saved us the time of writing the algorithm.

### 2.2 Data Cleaning and Analysis

The data collected from [3] had to be cleaned a lot since it had missing data and outliers. We wrote a Python code and used the 'json' library to convert the JSON files in the dataset into CSV files. The files generated were:

1. 'checkins.csv', which had the business names and the count of check-ins for each day.
2. 'time.csv', which had the business names and the count of check-ins for each time range in the entire day.
3. 'rating.csv', which had the business name and the count of ratings given to that business.
4. 'business.csv', which had the business ID, business name, state, city and major category of each business.

We had lots of check-ins with 0 values. But, that is not a missing value since a day can have 0 check-ins for a given day. So, we decided to see if we had any records with 0 values over a period of five days. About 28K records out of the given 140K records had such values in the 'checkin.json' file. So, we got rid of these records since we felt that those records could not sufficiently contribute to the analysis.

In the 'business.json' file, we encounter businesses with multiple categories. So, we first select 100 major categories from all the given categories and assign it to the business which contains one of these major categories. Then, we remove the categories which contain a minimal number of businesses since we feel that it will not be useful for the overall data mining procedure. We also remove businesses which have no categories.

In the 'time.json' file, we have counts for check-ins for the businesses and the time they were made during the day. We merge together the times to make 6 distinct time ranges. We name these ranges as Late night, Early morning, Early afternoon, Afternoon, Evening, Late evening. So, counts of check-ins from 0:00 am to 4:00 am will be added together and stored in the Late night time interval. This helps us better understand the time when the check-in was made.

We did not face the problem of incorrect column names or useless attributes since the CSV file was generated by us and we made sure to keep these points in mind. We do not remove duplicate records if the business ID is different but all the days have exactly same check-ins. This is because it is probable for two different business IDs to have the same order of check-ins.

## 2.3 Data Mining and Visualization

Initially, we decided to see which days might be attracting a lot of customers. We plotted a bar graph which helped us to see which day might have the highest number of check-ins in the given dataset. The figure below shows the bar graph plotted using the Matplotlib library.

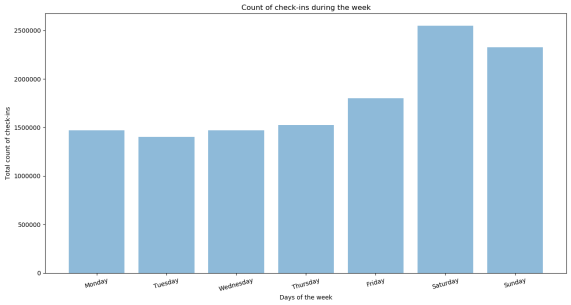


Figure 1: Count of check-ins during the week

We see that Saturday has the highest count of check-ins. This is unsurprising since usually it is a holiday and so people may want to get out of their houses. This could be said regarding Sunday too but since the next day is usually working, it might cause a slightly reduced count in check-ins.

We were also interested in finding out what time of the day might be attracting the most number of customers for businesses. A bar graph was generated for the same showing the six-time ranges that we have created spanning the entire day and the count of check-ins during that time for all businesses combined.

We see that the time range for late night, which is 0:00 am

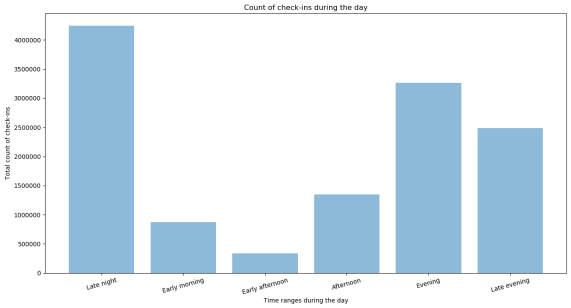


Figure 2: Count of check-ins during the day

to 4:00 am, has the highest count of check-ins. This is kind of surprising since people would usually be at home during that time. The lowest count of check-ins can be seen during the early afternoon period, which is 8:00 am to 12:00 pm. which could be put down to the fact that most people might be busy in offices or schools.

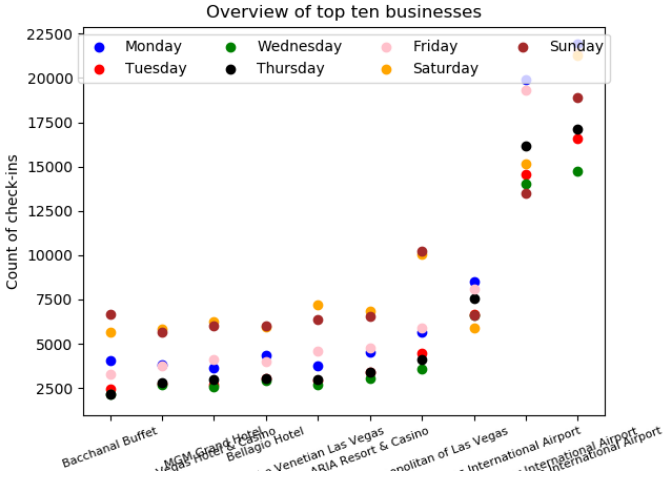


Figure 3: Count of check-ins for top ten businesses

From the generated 'checkins.csv' file, we extract the list of business names and the sum of all its corresponding check-ins for the entire week. We sort the list containing the business names and the sum of the check-ins for the week. Then we extract the list of top ten businesses in terms of most check-ins. This helps us to see how the businesses compare to each other in terms of check-ins over the period of a week.

We also individually compare the businesses over each day and see which business out of the top ten have a higher check-in count for that particular day. Unsurprisingly, we find that businesses with higher check-in counts dominate over the entire week for any given day.

From the 'checkin.csv' file, we also extract the list of business names and the sum of all its corresponding check-ins for the entire day. We use the list of top ten businesses that we compiled earlier and see how these businesses compare to each other in terms of check-ins over the entire day.

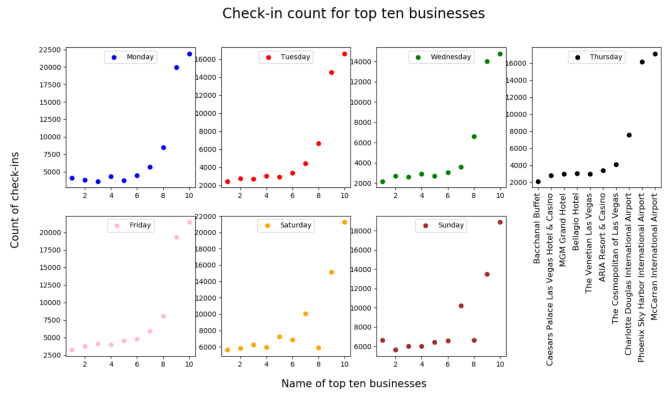


Figure 4: Individual analysis of each business on a given day

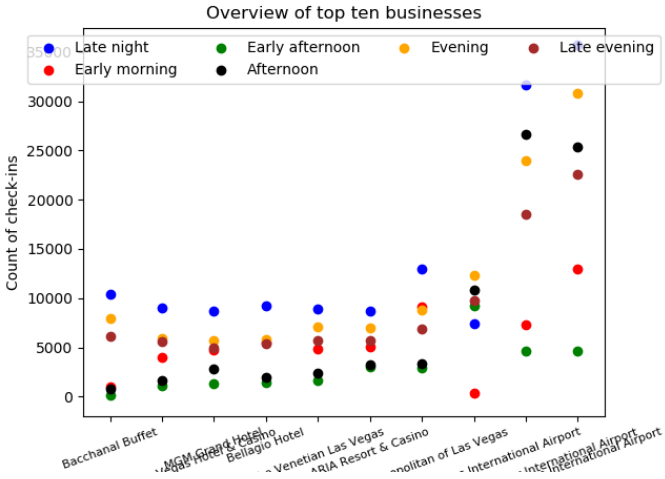


Figure 5: Count of check-ins for top ten businesses

We also individually check which business has a higher check-in count for a particular time range. Again as was the case for the entire week, businesses with higher check-in counts dominate over the entire day for any given range.

We also use KMeans algorithm to cluster the days of the week when we had most check-ins using the 'checkins' file. Given there are seven days of the week, we have seven clusters each representing the day of the week and the time when it was the busiest. For the time of the day using the 'time' file, since we have built six-time ranges covering the entire day, we have to have six clusters.

## 3. FUTURE WORK

We can go one step further than what we have done until now in terms of data cleaning. We can perform text mining on the generated 'ratings.csv'file. This would help us to understand whether a given review was good or bad for a business. We can also perform analysis on the 'business.csv'to show which businesses have the most check-ins in a given city or state. Another thing we could try, given time constraints is finding out which businesses from a particular major category gathered most reviews and check-ins.

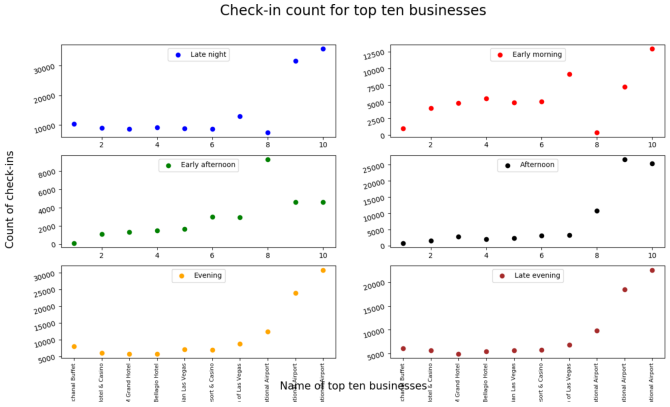


Figure 6: Analysis of each business for each time range

## References

[1] M. M. Hamad and A. A. Jihad. "An Enhanced Technique to Clean Data in the Data Warehouse". In: *2011 Developments in E-systems Engineering*. Dec. 2011, pp. 306–311. DOI: 10.1109/DeSE.2011.32.

[2] Dr. Arvind Kalia Sapna Devi. "Study of Data Cleaning and Comparison of Data Cleaning Tools". In: *International Journal of Computer Science and Mobile Computing* 4.3 (Mar. 2015), pp. 360–370.

[3] *Yelp Dataset*. 2013. URL: https://www.yelp.com/dataset%5C_challenge/.