I-INTRODUCTION:

London, which offers us a wide range in terms of hotel industry, is taken as a center in this research project. The data obtained as a result of the user opinions are passed through the steps of data processing and visualization and it is aimed to observe the effect of the comments of the visitors on the hotels.

II. RELATED WORK

Real-time data collection in hospitality applications and the use of APIs in the analysis processes of this data has become increasingly important. As seen in this study, the use of APIs in studies such as hotel scoring and analysis of comments provides facilities in terms of efficiency. This data is an important resource for emotion analysis and behavior modeling studies.

In the study carried out by Dey and at al. [6], the Smart Hotel Reservation System, which can analyse customer reviews and make hotel recommendations, is developed and data is collected with third-party APIs; it is analysed with natural language processing (NLP) techniques and more suitable hotel options are offered for users with the obtained data.

As shown in this study, the impact of models obtained with API data on decision-making systems and response speed is growing positively.

The integration of APIs with databases such as MongoDB enables the collection and analysis of large-scale unstructured data in the hospitality sector, allowing the flexible and scalable storage of this data with the use of MongoDB. Data is stored in JSON format at this stage because it is compatible with data withdrawn from the API and supported by MongoDB. In Scholz and Jeznik's studies [7] they recorded the Twitter data they obtained through the API in MongoDB JSON format and performed spatial emotion analysis using this data. The combination of API, JSON and MongoDB enables the analysis of data in an efficient and scalable way.

II. DATA PROCESSING METHODOLOGY

Google Maps Places API was used to obtain the hotel data used in this project.[1] The API provides hotel information made within a certain kilometer limit based on geographical coordinates or cities. API responses were based on place\_id, rating, user\_rates\_total, name, geometry.location. API access is provided by the personal Google API key, which is defined in a separate .env file. Data capture is done using Python's request library. [2]

To perform the data fetching;

-Using three dots, text-based enquiries such as "hotels in London" were also made within certain coordinates.

-More records were obtained in each query using next\_page\_token in queries by Google Places API.

-Replies from API in JSON format were converted to pandas. DataFrame in Python and saved to the database. .[3] The database used at this stage is MongoDB[4]

IV. DATA VISUALISATION METHODOLOGY

When visualizing data, the Plotly library of Python was used, with the aim of achieving more effective graphics using plotly.express.[5] As a result of these visualizations, it is aimed to facilitate the interpretation of the previously obtained variables such as users' scores and comment counts. In addition, the interactive features of all the graphics created with the Plotly library are an important factor in selecting this library for this study. Especially when examining the graphics during the analysis phase, it provides advantages due to the fact that it allows us to easily reach the details such as hotel name, score and number of comments. The resulting graphics are exported as .html and made openable in the browser.

Chart 1: Rating Distribution:

The google\_rating values of hotels are visualized using this histogram shown above. As can be seen in the histogram, most of the hotels are in the 4.0 and 4.7 points range. For this reason, we can say that the hotels in London have a high quality of service and leave a good impression on the passengers traveling here.

Chart 2: Review Distribution:

In this graph, based on the values in the column google\_review\_count, it is aimed to interpret the interest of the users to the hotels by starting from the number of comments. The positive distortion of the distribution we have achieved has shown us that; while the comments of the majority hotels are within a limited number, certain hotels have thousands of interpretations, especially the tourists affect their accommodation preferences with the comments they receive.

Chart 3: The Relationship Between Ratings and Comments

It is a graph intended to co-analyze the values of google\_rating and google\_review\_count of the two graphs we have previously analysed. As a result of this graph, it is aimed to observe whether there is a correlation between the two variables. Even if a strong linear relationship cannot be found, the clusters that are seen between some groups; the fact that small hotels with high ratings but few interpretations and hotels with very high interpretations but average ratings are located in the same slice indicates that users are concentrated on these two groups.

V. RESULTS AND EVALUATION

As a result of the analysis, it was understood that the hotels in London had high scores, but the number of reviews varied greatly in general. Our results from the scatter chart have shown us that there is not as strong a correlation between hotel quality and popularity as expected. Results from histograms showed that the hotel scores were concentrated at the mid-to-high level, indicating a positive distorted distribution in the number of reviews. The findings show us that service quality is high across London, but popularity is also influenced by parameters outside of hotel ratings.

References:

[1] <https://developers.google.com/maps/documentation/places/web-service/text-search?hl=tr>

[2] <https://requests.readthedocs.io/en/latest>

[3] W. McKinney, *Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython*, 2nd ed., O'Reilly Media, 2017.

[4] MongoDB, Inc., “PyMongo Manual.” [Online]. Available: <https://pymongo.readthedocs.io>  
[5] Plotly Technologies Inc., “Plotly for Python.” [Online]. Available: <https://plotly.com/python/>

[6] K. B. Dey, T. L. Sahoo, R. R. Rout, D. Puthal, and R. Buyya, “Smart Hotel Booking System Using Sentiment Analysis,” *2020 IEEE Region 10 Symposium (TENSYMP)*, Dhaka, Bangladesh, 2020, pp. 834–839, doi: 10.1109/TENSYMP50017.2020.9342744.

[7] J. Scholz and J. Jeznik, “Evaluating Geo-Tagged Twitter Data to Analyze Tourist Flows in Styria, Austria,” *ISPRS International Journal of Geo-Information*, vol. 9, no. 11, p. 681, Nov. 2020, doi: 10.3390/ijgi9110681.

A. Data Selection:

The hotel data used in this project was obtained programmatically through the Google Places API, filtering accommodation points classified as "hotel". Google Places API is semi-structured because the data returned is in JSON format.

The data was processed temporarily in Python and then designated as a MongoDB database and saved there. Recording the data here has made it easy for it to manage complex and nested structures from the API.

If you want, I can combine all the texts we have translated so far and present them to you as a regular document. If you want to continue, I can translate the next text.

B. Pre-Processing:

The raw data saved to MongoDB has been passed through some steps using the Pandas library in Python. The selection of the required fields and the parameters to be used during the analysis period were selected. For example; name, rating, user\_rates\_total. Incomplete records of data to be used throughout the analysis, especially rating and user\_ratings\_total, were extracted. The flattening of nested structures was positioned in the data so that the location information was separate columns. Cleaning of double records, records with the same name were extracted from the data.

C. Donusturme: Veri donusturme asamasinda, MongoDB den cekilen ve temizlenen veri PostgreSQL e aktarilarak PostgreSQL uzerinde bir veritabani olusturulmustur ve otel verilerini barindiran tablo semalari tanimlanmistir. Veri PosgreSQL e yazilmadan once yeniden yapilandirilma asamalarindan gecirilip sayi, metin gibi formatlar sadelestirilmistir. Veritabani olarak PostgreSQL in bulut surumu olan Neon kullanildi bunun baslica sebepleri Neon un kolay entegrasyon ve