US Parks Visitors Data Trend

Huzefa Igatpuriwala, Saurabh Swaroop, Vijayakumar Perumalsamy
Indiana University
Bloomington, Indiana

ABSTRACT

Parks are one of the most iconic symbols in America representing the beauty of nature, historic site, monument, memorial or wild-life reserve. United States has 379 parks operated by NPS (National Park Services). There could be various factors like weather, crowds, visitation fee, season, etc. which would make someone to choose an ideal park. This project shows a way to perform exploratory data analysis along with visualization of all types of parks in US.

KEYWORDS

National Parks, Park Visitors, Recreation Visitors, Tent Campers, RV Campers

1 INTRODUCTION - MOTIVATION

Cost and effort for park visit make it very important for someone to choose an ideal park. For a particular month, factors like crowd and weather needs to be analysed before visit. Increase in visitor count nowadays is also influenced by easy of availability of cameras and smartphones nowadays. As a visitor, moderate crowds and ideal weather make it worthwhile trip. It also becomes important for employment seekers like food vendors to select an ideal park for making best of sales. Currently, good data analysis and visualizations are hard to find for National Parks.

2 BACKGROUND

Google results also suggests a park name for every month. However, it is not in the form of an interactive visualization. It gives direct results. So, the user may not know why that is best park of the month. Also, what if that park has been already visited by that person and he is looking for second best park? This project is a solution for these problems. Additionally IRMA which manages and delivers information on National Parks, does not seem to have good data analysis or visualization work done for National Park Visitor stats data. Instead IRMA makes this data available at https://irma.nps.gov/Stats site.

3 DATASET

The Integrated Resource Management Applications (IRMA) Portal provides easy access to National Park Service applications that manage and deliver resource information to parks, partners and the public. This portal provides month-wise visitation count of all 18 park types from 1904 to 2019. That's a lot of data. Additionally Google Maps API can be leveraged to find geo-location of each park. The API returns Latitude and Longitude Information based on address.

3.1 Pre-Processing

We did pre-processing in following steps.

IRMA Data Pull : The National Park Services IRMA website was used to download data in CSV format for the past 10yrs. The dataset contains information about month-visit counts to parks for different recreation types i.e Recreation, Tent Campers and RV Campers.

Google Maps API : We built a jupyter notebook to query Google Maps API by providing Park Locations. This gave us latitude and longitude info about parks in a CSV.

IRMA + Lat/Long Info : The data received from IRMA was tied with latitude and longitude info using UnitCode attribute present in both the datasets. State Info was added using abbreviations present in IRMA dataset.

4 OBJECTIVES

With this project we intended to analyse National Parks data from IRMA using both NoSQL and SQL databases and club our findings into some interactive visualizations. Stated below are high level objectives for this project.

4.1 Data Pre-processing

- Successfully import 10 years worth of US parks visitor data from CSV in Python
- Get Latitude/Longitude information for each park using Google API
- Populate states against each park and merge with Geographical info for park locations dataset

4.2 NoSQL Database

The second classification experiment we performed was to classify Fear vs Anger classes. We segregated the data set from Distress class in two sub-classes. First class had all the audio files belonging to Fear. Other class had all audio files belonging to Anger. We trained all the samples with four different feature sets and measured the performance.

- Merge Park visitors and locations into a single dataset, which has Lat/Long info against visitor info.
- Import Final dataset to MongoDB Collection.
- Querying via PyMongo using Conditional Operators/Aggregation Pipelines
- Interactive Visualizations using IPyWidgets

4.3 SQL Database

- Load Parks Visitor and Parks Location dataset to MySQL DB
- Perform analysis on loaded data using Structured Query Language (SQL)
- Build a web-application using Amazon Web Services (EC2 and RDS) services for live querying on cloud.

4.4 Technologies Used and Why?

In this project a variety of tools along with multiple databases have been used. We have also leveraged the power of cloud services to build an interactive website to perform real time querying. Here's the list of technologies used and why?

4.4.1 *Jupyter Notebooks with Python Pandas Library.*

- They are a powerful way to write and iterate Python code to perform data analysis and visualizations. Rather than writing and re-writing an entire programs, lines of code can be run one at a time using notebooks.
- Pandas Library presents data in a way which is suitable for data analysis with the help of Dataframes datastructures. It provides for easy subsetting and filtering of data using minimal, concise and clear code, so that user can focus more on core goal rather than having to write lot of code.
- This library helped us mainly to perform data pre-processing before loading to MongoDB as well as notebook style of coding helped us to showcase visualizations easily.

4.4.2 *MongoDB, PyMongo and IPyWidgets.*

- MongoDB is a NoSQL database which stores documents in one collection. It also has deep-query ability which supports dynamic queries, aggregations on documents which is as powerful as SQL.
- PyMongo is a Python distribution containing tools for working with MongoDB, and is the recommended way to work with MongoDB from Python.
- Jupyter widgets 'interact' function automatically creates UI controls for exploring code and data interactively. This was used with Python's plotly library to make interactive visualizations.
- Tying all 3 together made sense with python for our use-case, which helped us to not only query data using MongoDB but also visualize them effectively.

4.4.3 *MySQL and Amazon Web Services.*

- MySQL is a popular open source database management system commonly used in web applications due to its speed, flexibility and reliability.
- AWS provides on-demand access to scalable web and application servers, storage, databases, content delivery, cache, search, and other application services that make it easier to build and run apps that deliver a great customer experience.

5 RESEARCH QUESTIONS AND RESULTS

With this project our main objective was to leverage the power of data analysis to derive insights or help make informed decisions. The insights would either help visitors choose the ideal time to visit a park with moderate crowd and best weather. It would also help employment seekers like food vendors select the best time of the year to set up food trucks and make high sales. Below are some of the questions which we tried to answer with our analysis and visualizations.

5.1 NoSQL - Mongo DB database

- Different park types and their counts – PyMongo Aggregation Pipeline to sum distinct count of parks in US, this would give an overall picture to visitors and vendors on the number of parks present across US and their types.
- Countrywide Spread of Parks as per park type – PyMongo Aggregation to sum count of visitors for past 10 yrs and display on Map. The map would help visitors locate type of parks present in their state and whether their popularity based on count of visitor in past 10yrs.
- Different kinds of visitors across parks in past 10yrs – PyMongo Aggregation to sum count of visitors monthly for past 10 yrs. The data from pipeline was fed to interactive scatter plot which would show visitor count for the chosen year and park. This would help users determine which month of the year was adequate for visiting a park.
- Monthly Average of visitors in past 10yrs for each park – PyMongo Aggregation to find average count of visitors monthly for past 10 yrs. The data from pipeline was fed to interactive scatter plot which would show average visitor count for the chosen park. This would help users approximate expected no of visitors for the current year.
- Count of visitors for each month and year per park – PyMongo Aggregation to find sum of count of each recreation type for a specific month and year. The data from pipeline was fed to interactive scatter plot which would show count of respective recreation type(tent/RV campers) count for the chosen year/month and park. This would help specific recreation type users like tent campers determine best time to visit a park.
- Statewise Parktype – PyMongo Aggregation to find count of different park types in each state. This helps users identify the different places for visit in their respective state.
- Yearly Count of Each Recreation Type(Visitors) per park – PyMongo Aggregation to find yearly count of visitors for each park. The data from pipeline was fed to interactive scatter plot which would show stacked bar chart of visitor count for the chosen park.

5.2 Interactive Web App

- Local System Design – Normalized data from MySQL was used as backend for a python based web application which would allow querying Parks and Location datasets at real-time from MySQL.
- Cloud Migration of Data – Amazon Web Services(AWS) Data Migration Service was leveraged to migrate normalized data from Local MySQL DB to AWS RDS MySQL.

6 CHALLENGES/FAILED EXPERIMENTS

- Initially, we used third party website to show location of parks on Map. However, the coordinates were not accurate and so we had to use Google API for this purpose.
- Google API was not returning coordinates for Sequoia National Park, CA and so for this park alone, we had to manually set the coordinates.

- Extra spaces in the string data and commas in the numerical caused errors while loading. So, we had to shape the data in the code.
- Showing data for all National Parks for the past 10yrs in a tabular format was not feasible. Here IPywidgets came to our rescue where we could showcase results of MongoDB aggregations with the help of nice drop down selection for each visualization.
- Migration of data from Local MySQL database to Cloud was a challenge, we compared options between Azure and AWS. AWS Data migration Service made it very easy for us to migrate from Local instance to Cloud.

7 CONCLUSION

Using this data analysis and visualizations, an ideal park can be estimated for any particular month by a recreational visitor, hiker, alone timer seeker or an employment seeker. The visualization and the estimation was built over visitation count for each park. However, other features like Weather, Visitation fee, Motorcycle allowance, stroller-friendly, Tents allowance, camping allowance, Overnight stays, etc. can be implemented to more accurately estimate the ideal park.

8 REFERENCES

- https://creately.com/blog/diagrams/aws-templates-for-architecture-diagrams/
- https://cloud.google.com/maps-platform/
- https://irma.nps.gov/STATS/SSRSReports/National Reports/Query Builder for Public Use Statistics (1979 - Last Calendar Year)