SSS (State Service Site) Encyclopedia

Team: Random Seed
Team Members: Junwei (Ivy) Sun, Liana Hill, Hanbo Dong, and Kedi Le

Project Overview

Planning trips to the U.S. National Parks is surprisingly difficult for travelers due to fragmented information across multiple websites and the lack of systematic comparison tools. Existing platforms, like the National Park Service website, provide extensive data but lack analytical modeling or personalized recommendations. Another limitation of these platforms is their professional focus, which may deter casual travelers from efficiently planning their trips and identifying parks that match their interests. These gaps create friction in the trip planning process, particularly for travelers seeking to match destinations with specific activities, seasonal preferences, or multiple interest areas. This project builds a unified R Shiny App that consolidates park data, enables systematic comparisons, and delivers tailored recommendations to streamline the travel planning experience.

Objectives

This project aims to create an interactive Shiny application that transforms how recreational travelers explore and select National Parks. We had three primary objectives: to integrate multiple data sources, including visitation patterns, park topics, and geographic information into a cohesive, visually compelling platform; to develop a recommendation engine that accounts for both user preferences and park popularity, allowing travelers to balance personal interests with popular destinations; and to implement unsupervised learning to reveal natural groupings among parks, helping users discover comparable alternatives they might not have considered.

Existing Work

The National Park Service operates search tools for exploring their park system. Their website includes topic-based search functionality. For example, searching "Birds & Birding" generates a US map with relevant parks. Additionally, separate dashboards exist for visitation statistics and individual park details. However, these tools have notable limitations for typical travelers. The search interface displays all matching parks nationwide without ranking or filtering options, creating overwhelming information. Seasonal information is absent, making it difficult to determine optimal visit timing for activities like wildlife viewing or wildflower observation. The system only supports single-topic searches, so users cannot explore parks matching multiple interests simultaneously. Crucially, visitation data and activity information are in separate tools, preventing users from understanding which parks are both suitable and popular. These platforms function more as comprehensive databases for researchers than as tools for trip planning.

Programming Paradigms

Our implementation leverages two complementary paradigms: functional programming and unsupervised machine learning. The functional programming approach supported our data processing infrastructure and recommendation logic. We constructed modular, composable functions for data transformation and designed the recommendation system as a pure function with deterministic outputs. Our recommend_parks() function calculates park suitability through Pearson correlation between user interest vectors and park feature vectors, then blends these scores with normalized popularity metrics using a user-controlled weighting parameter. The machine learning component employs K-means clustering to identify parks with similar characteristics across 30+ features. Users can dynamically select which features drive the clustering and adjust the number of groups, with an accompanying elbow plot visualization to guide optimal cluster selection. This ensures we deliver both personalized, explainable recommendations and exploratory pattern discovery capabilities.

Data Collection

The data for this project was collected by the National Park Service. Using an API, we collected the geographical information (i.e., latitude, longitude, and state location) of each national park and interesting features found within each national park. Interesting features of the park could be whether the park contained mountains, scenic views, waterfalls, etc. The features of interest were recorded as binary variables (i.e., 0 if the feature was not found within the park and 1 if the feature was found in the park) using functional programming. This included using pipe operators as well as the tidyverse() and dplyr() packages to organize the data.

The NPS also provided CSV summary reports containing the monthly visitor counts for the national parks for the annual summary reports for visitation counts. Since we were interested in the years 2023-2024 for our machine learning model, each monthly report was downloaded, cleaned, and combined into one dataframe using the purrr() and dplyr() packages in R.

Challenges & Accomplishments

There were multiple challenges that we encountered while building the national park travel recommendation Shiny App. Firstly, reflecting the exact locations of national parks onto the U.S. map was a task we spent time investigating. While we were able to display the U.S. map successfully with the usmap() package in R, we were not aware of approaches that enable adding an additional layer of coordinates on top. We conducted extensive research and utilized the National Park Service API to download the exact longitude and latitude of each park, thereby building the foundation for many of our visualizations later in the App.

Another challenge was to balance model choice with ease of interpretability. While we initially proposed other modeling techniques, such as implementing a multinomial logistic regression model instead of simple correlation analysis for recommendation systems, we decided to pick models that can offer easy-to-interpret, informative suggestions to the users, while also being statistically sound. Through this project, we kept the utility of the App in mind so we avoid implementing complex models that might hinder users from understanding the App output.

We achieved the following accomplishments in this project:

1. **National Park Summary Statistics -** We summarised the **location**, **name**, and **number of national parks** along with **popular topics** aggregated at the state-level, and presented the information as an introductory tab in our Shiny App. This tab allows users to freely explore state-level park characteristics while also learning the specific parks within each state.

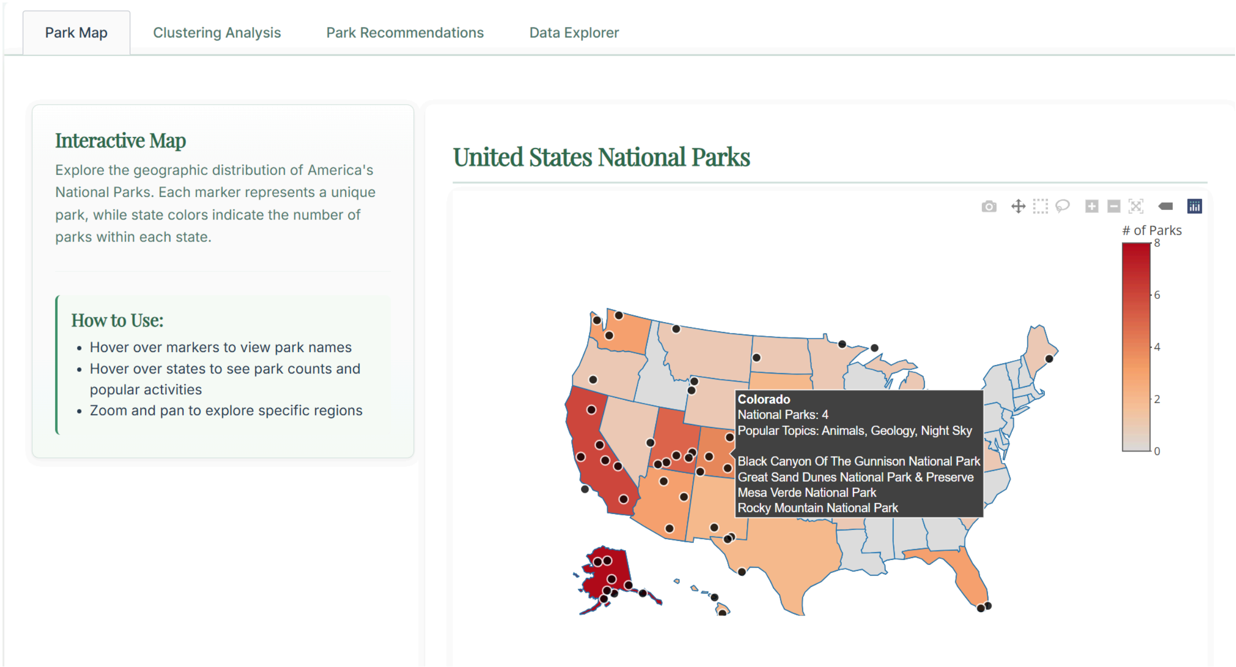
2. **Park Grouping System -** We developed a flexible K-means clustering framework that clusters parks based on binary (0/1) topic vectors representing their thematic features. The system is designed to be easily extendable, allowing additional attributes such as entry fees, park area, or other quantitative features to be incorporated in the future. This provides a robust and scalable foundation for grouping parks by similarity as new data becomes available.

3. **Recommendation System -** We build a correlation-based model to generate personalized recommendations for national parks based on both the user's preferences (**Suitability**) and park **Popularity**. **Suitability** is determined by the similarity between the user's selected topics and preferred month of travel and the corresponding features of the parks, while **popularity** is measured using annual visitor numbers. A user-specified parameter allows flexibility in balancing between suitability and popularity. The model produces an overall score for each of the parks based on the user's selection and ranks the top parks accordingly.

4. **An App that Unifies Information -** Combining accomplishments above, we were able to build a user-facing R Shiny App that combines multiple domains of information (park characteristics as well as visitation data) to build customized recommendations for users.

Functionality/Shiny App Overview

Interactive Park Map - The first tab on our Shiny app is an Interactive Park Map, where users can access some state-level summary statistics and the geographical location of the national parks. The color scale gradient represents the number of national parks located within each state. This means darker colored states have more national parks. Hovering over the states allows users to see the names of the parks as well as popular park features within the state.



Park Map | Clustering Analysis | Park Recommendations | Data Explorer

Interactive Map

Explore the geographic distribution of America's National Parks. Each marker represents a unique park, while state colors indicate the number of parks within each state.

How to Use:

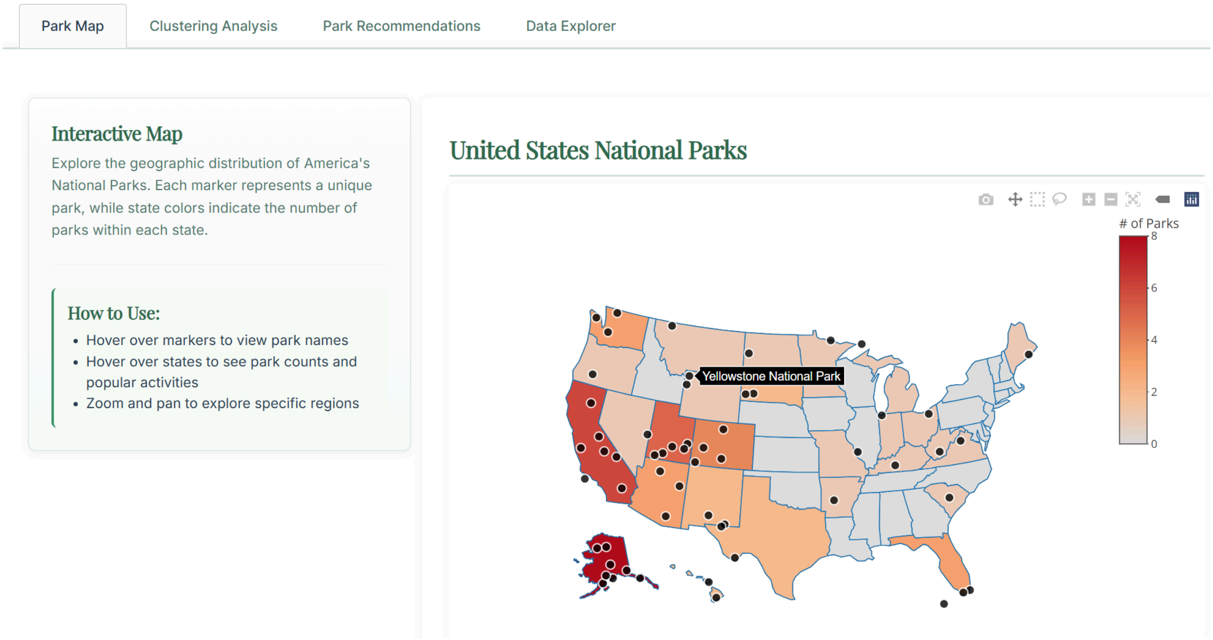
- Hover over markers to view park names
- Hover over states to see park counts and popular activities
- Zoom and pan to explore specific regions

United States National Parks

Colorado
National Parks: 4
Popular Topics: Animals, Geology, Night Sky

Black Canyon Of The Gunnison National Park
Great Sand Dunes National Park & Preserve
Mesa Verde National Park
Rocky Mountain National Park

Figure 1a. The image above showcases the information displayed on the "Park Map" tab when users hover over a state.



Park Map | Clustering Analysis | Park Recommendations | Data Explorer

Interactive Map

Explore the geographic distribution of America's National Parks. Each marker represents a unique park, while state colors indicate the number of parks within each state.

How to Use:

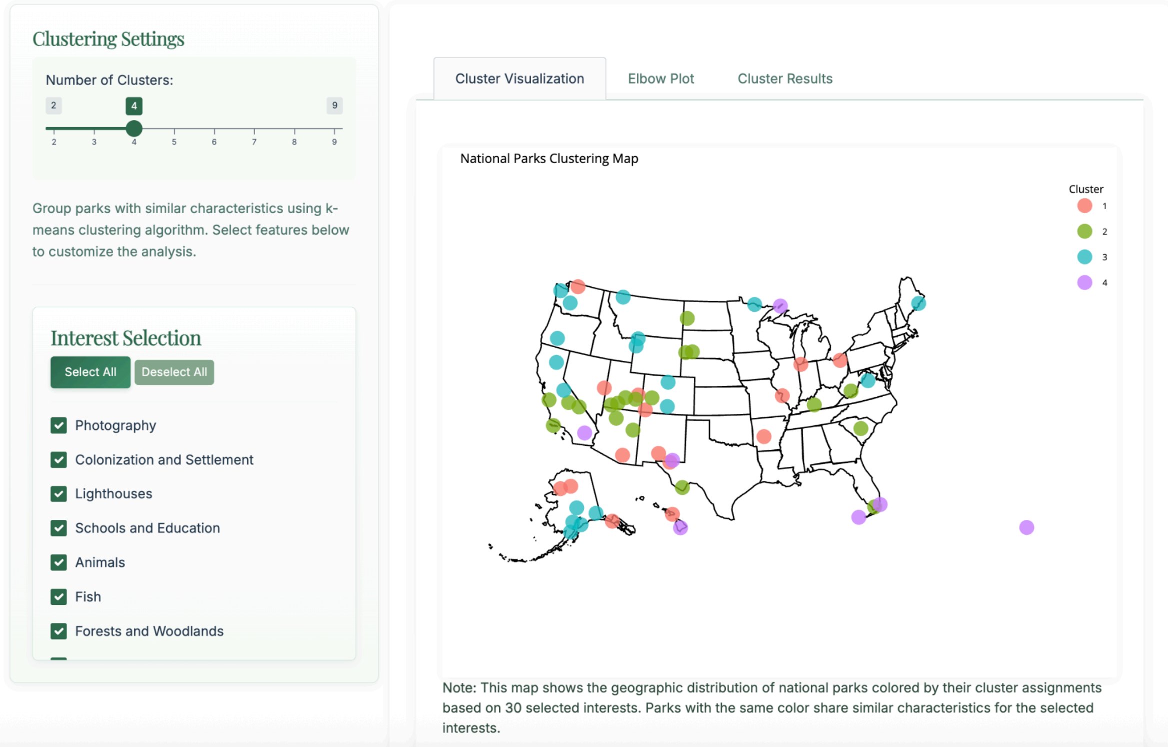
- Hover over markers to view park names
- Hover over states to see park counts and popular activities
- Zoom and pan to explore specific regions

United States National Parks

Yellowstone National Park

Figure 1b. The image above displays what appears on the "Park Map" when users hover over the black dots, which represent each national park.

K-means Clustering - The second tab in our R Shiny App is a grouping system that groups parks based on similarity in topics supplied by the users. Internally, it utilizes a K-means clustering algorithm where each park is represented by a binary 0/1 feature vector.



Number of Clusters: 4

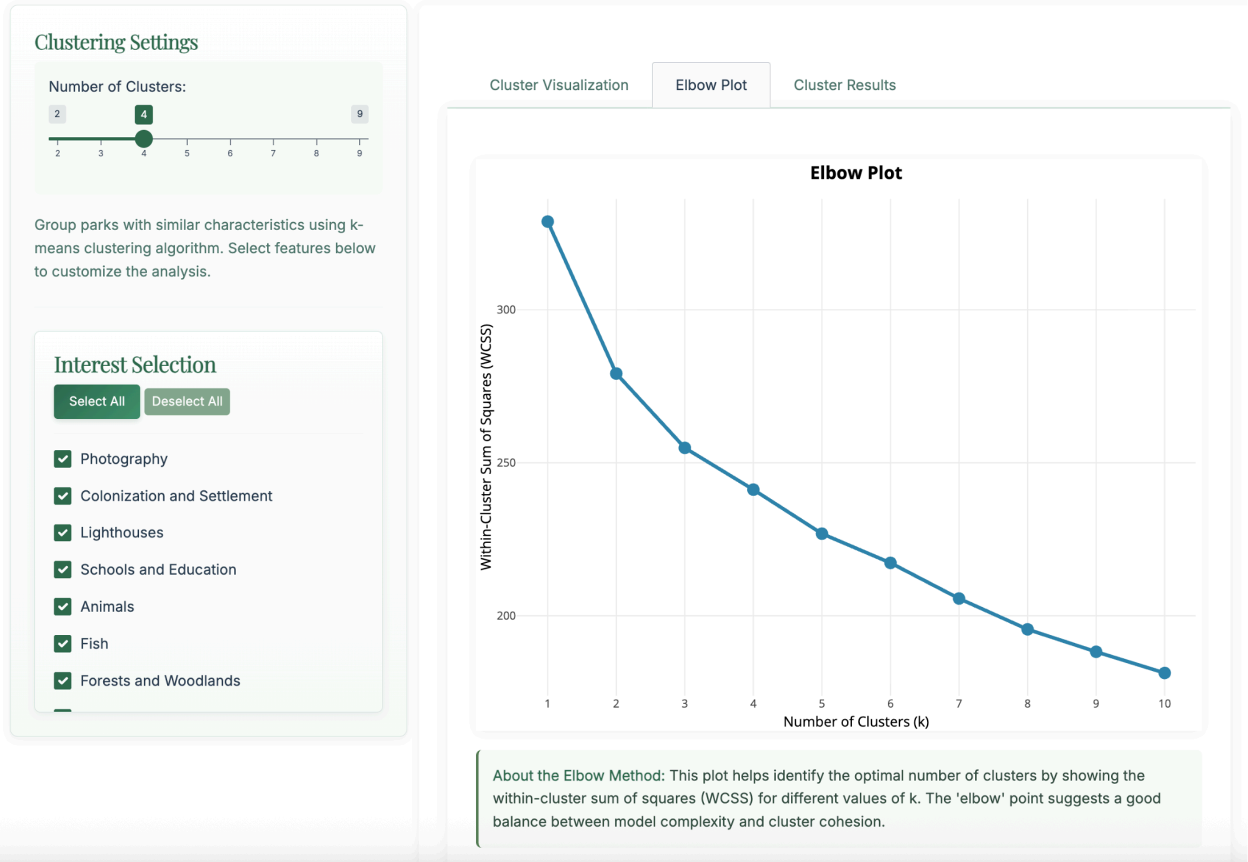
Group parks with similar characteristics using k-means clustering algorithm. Select features below to customize the analysis.

- ☑ Photography
- ☑ Colonization and Settlement
- ☑ Lighthouses
- ☑ Schools and Education
- ☑ Animals
- ☑ Fish
- ☑ Forests and Woodlands

Note: This map shows the geographic distribution of national parks colored by their cluster assignments based on 30 selected interests. Parks with the same color share similar characteristics for the selected interests.

Figure 2a. The "Clustering Analysis" tab. The tab allows interactive user inputs, such as changing the number of clusters and the topics used for clustering. Clustering results will then be reflected onto the interactive map on the right with each dot representing a national park and the color representing the cluster it is assigned to.

K-means clustering conventionally uses the Elbow method to determine the optimal number of clusters to use. It is a method to choose the hyperparameter k such that the determined number of clusters separates the data well without overfitting. We provide such an Elbow plot on our Shiny App to guide users' cluster number choice.



Clustering Settings

Number of Clusters: 4 (2–9)

Group parks with similar characteristics using k-means clustering algorithm. Select features below to customize the analysis.

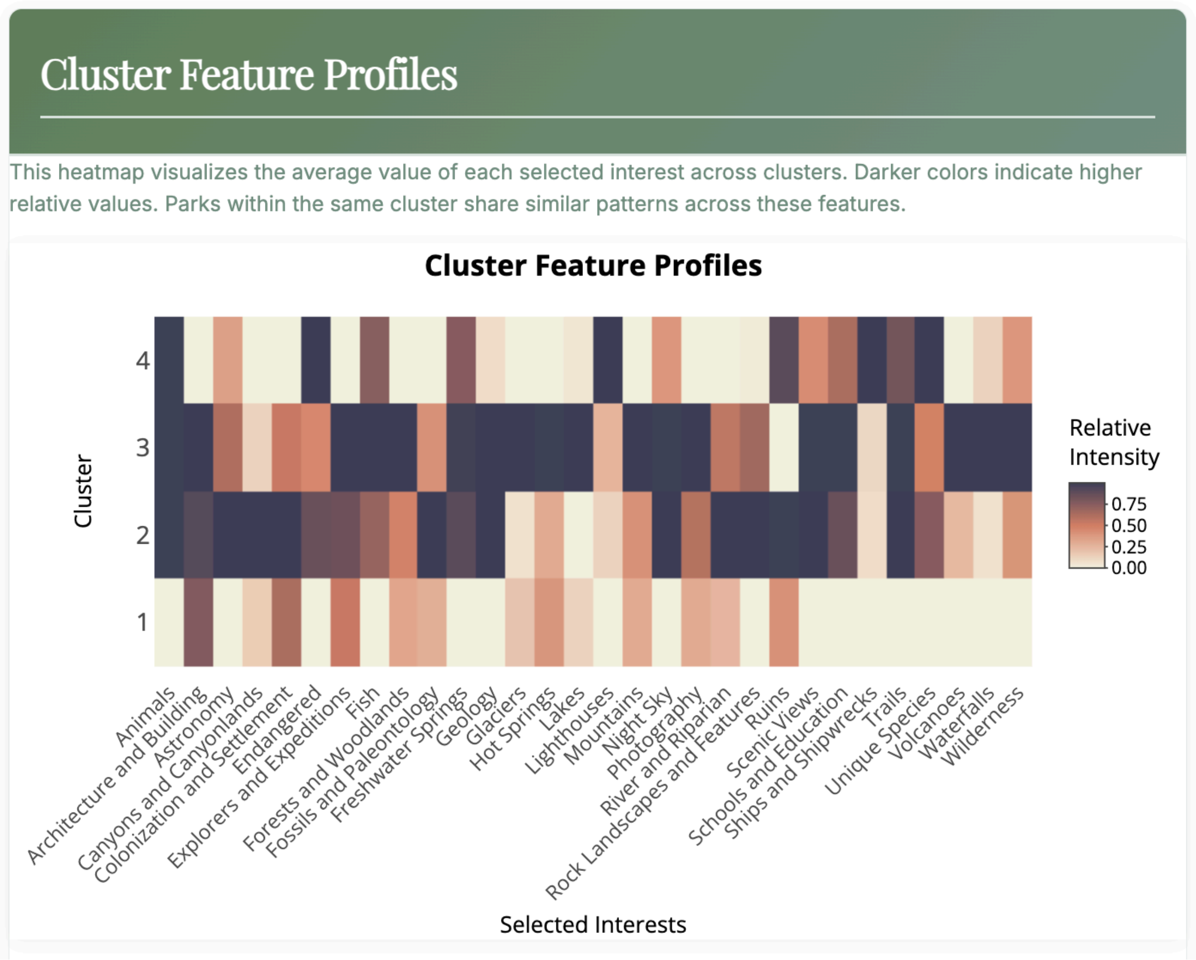
Interest Selection

- ☑ Photography
- ☑ Colonization and Settlement
- ☑ Lighthouses
- ☑ Schools and Education
- ☑ Animals
- ☑ Fish
- ☑ Forests and Woodlands

About the Elbow Method: This plot helps identify the optimal number of clusters by showing the within-cluster sum of squares (WCSS) for different values of k. The 'elbow' point suggests a good balance between model complexity and cluster cohesion.

Figure 2b. Elbow plot that supplements K-means clustering. The elbow is defined as the value of k where the within-cluster sum of squares has the largest drop in value.

Finally, to help users interpret the clustering results, we provide a third sub-tab that presents a summary for each cluster. This summary includes the number of parks in each cluster as well as the specific park names. In addition, we include a heatmap that highlights the defining characteristics of each cluster. The heatmap displays the average value of each selected interest across clusters, with darker colors indicating greater enrichment of that interest within a given cluster.

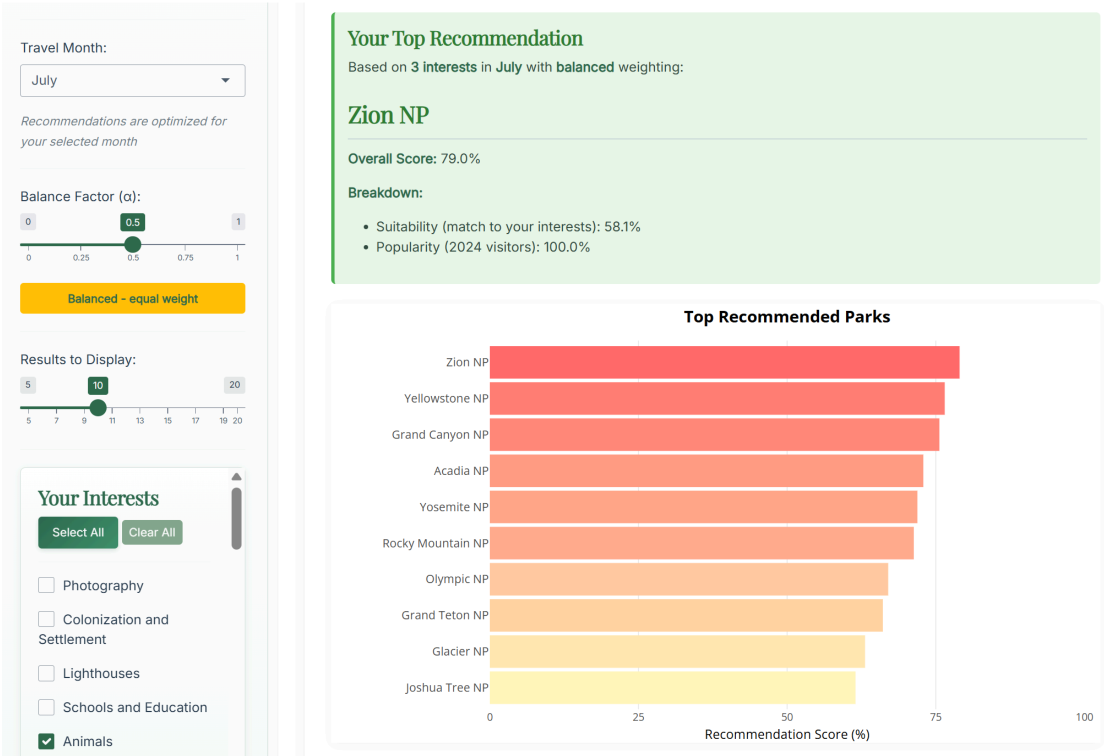


Cluster Feature Profiles

This heatmap visualizes the average value of each selected interest across clusters. Darker colors indicate higher relative values. Parks within the same cluster share similar patterns across these features.

Figure 2c. Heatmap that summarises cluster characteristics. Each row of the heatmap represents a cluster, and each column a user-selected interest. The color intensity shows how strongly that interest characterizes the cluster.

Recommendation System - The third tab in our R Shiny App is our recommendation system, where we will help users identify the parks that best fit their travel needs.



Travel Month:

July

Recommendations are optimized for your selected month

Balance Factor (α): 0.5

Results to Display: 10

Your Interests

☐ Photography

☐ Colonization and Settlement

☐ Lighthouses

☐ Schools and Education

☑ Animals

Your Top Recommendation

Based on **3 interests** in **July** with **balanced** weighting:

Zion NP

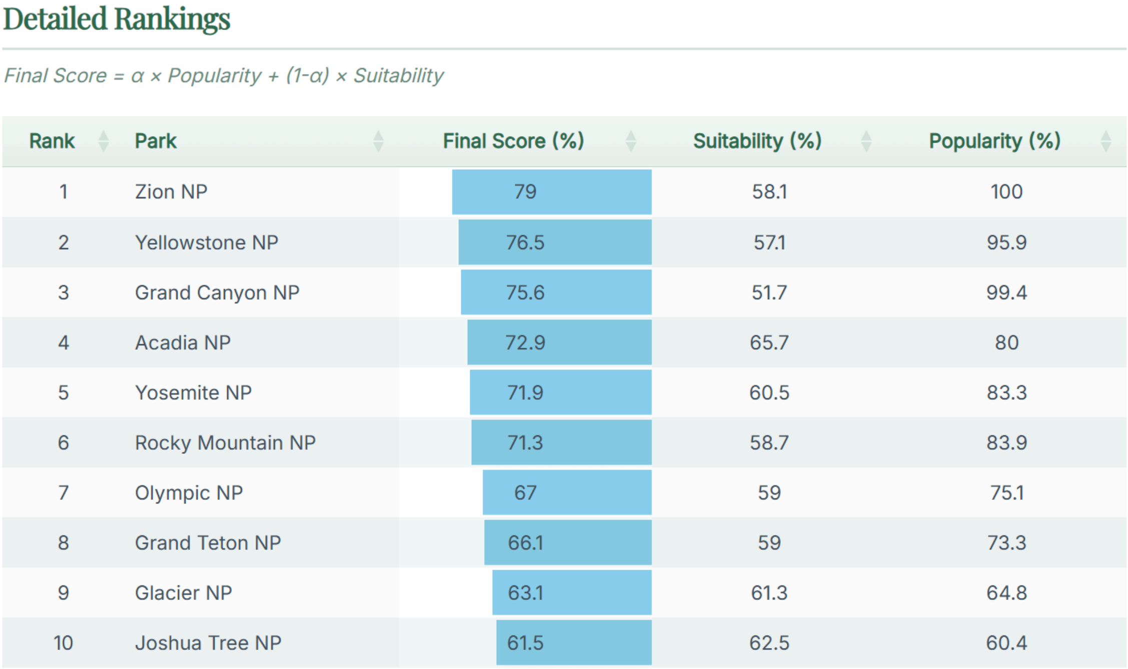
Overall Score: 79.0%

Breakdown:

- Suitability (match to your interests): 58.1%
- Popularity (2024 visitors): 100.0%

Top Recommended Parks

Figure 3a. Users can specify their topics of interest and preferred time on the left, and they also need to enter the parameter to balance between suitability and popularity. The recommendation system will generate the results for top-rank parks on the right.

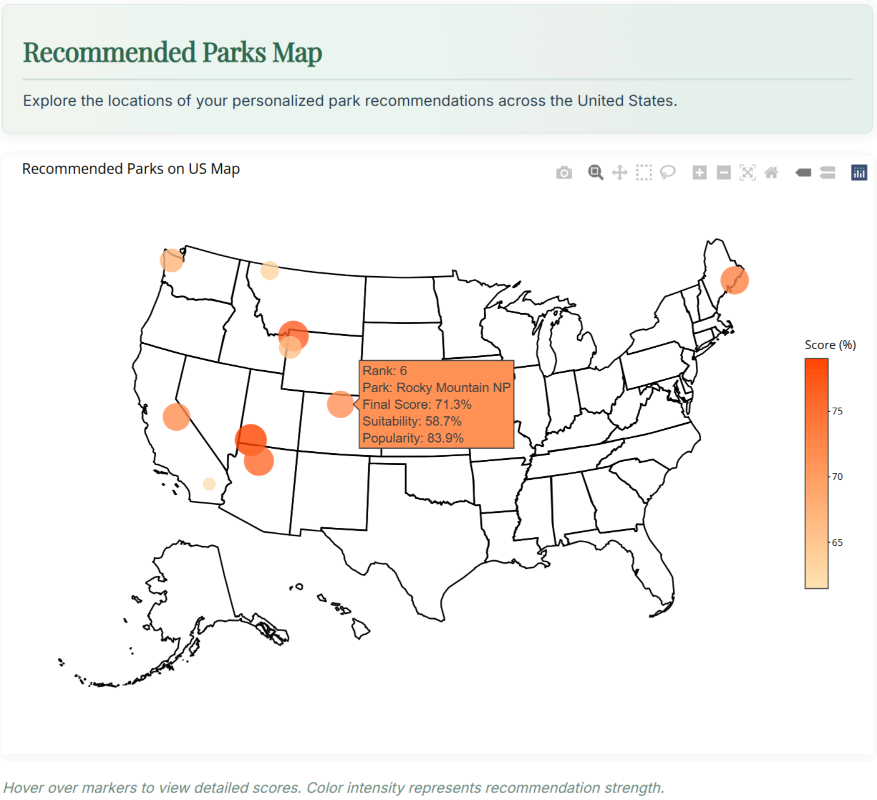


Detailed Rankings

Final Score = α × Popularity + (1-α) × Suitability

Rank	Park	Final Score (%)	Suitability (%)	Popularity (%)
1	Zion NP	79	58.1	100
2	Yellowstone NP	76.5	57.1	95.9
3	Grand Canyon NP	75.6	51.7	99.4
4	Acadia NP	72.9	65.7	80
5	Yosemite NP	71.9	60.5	83.3
6	Rocky Mountain NP	71.3	58.7	83.9
7	Olympic NP	67	59	75.1
8	Grand Teton NP	66.1	59	73.3
9	Glacier NP	63.1	61.3	64.8
10	Joshua Tree NP	61.5	62.5	60.4

Figure 3b. Score breakdown for the top recommendation parks.



Recommended Parks Map

Explore the locations of your personalized park recommendations across the United States.

Recommended Parks on US Map

Rank: 6
Park: Rocky Mountain NP
Final Score: 71.3%
Suitability: 58.7%
Popularity: 83.9%

Hover over markers to view detailed scores. Color intensity represents recommendation strength.

Figure 3c. Visualization of the recommendation results on the map. Users can hover over to see the detailed information for that park.

Data Explorer - The last tab in our R Shiny App provides a brief search table for data we used for clustering. This is meant to serve as additional information for the users if they wish to explore the raw data on top of the clustering and recommendation system results.

Usability and Documentation

We tested the usability of our Shiny App to ensure all functionalities are implemented as desired. Edge cases such as when the user did not select enough interests for clustering were marked by a warning message upon the edge case being detected. Finally, we uploaded all codes associated with our R Shiny App to a Github Repo with a detailed README file to illustrate our design and app usage.

Originality and Complexity

The originality of our project comes from building an intuitive and user-friendly R Shiny App to summarize the information for national parks and to provide informed travel suggestions not addressed by previous work. We generated state-level summary statistics and visualized them on an interactive map. We applied k-means clustering to group parks with similar profiles, and we developed a correlation-based recommendation system to help identify the parks tailored to user's preferences. We integrated the results of these models into the dashboard, where users can explore the results of interest by easily adjusting the inputs, enabling flexible and efficient exploration of national parks and informed travel decisions.

Limitations and Future Steps

Our project was limited in a few ways. First, our data contained information for 59 national parks, although there are 63 national parks in the U.S. Future steps could be made to research and manually add the information about these missing parks into the existing data. The National Park Service also manages other national landmarks such as historic sites, monuments, and memorials. Our project could be expanded to examine these other landmarks in our dashboard. Another possible extension could be incorporating the user's personal location into our machine learning.

Another limitation lies within the layout of our data. From the data obtained using the API, multiple variables contained textual information about the parks. Future efforts could be made to summarize this information using AI tools and integrate it into our recommendation system. Similarly, another future direction could be including additional information into our dashboard and clustering analysis, such as the average price of visiting each national park, including entrance and parking fees.

For more demonstrations of our product:
Link for our deployed Shiny app: https://team-randomseed.shinyapps.io/SSS_dashboard/
Link for our codes: https://github.com/jhu-statprogramming-fall-2025/project04-random-seed