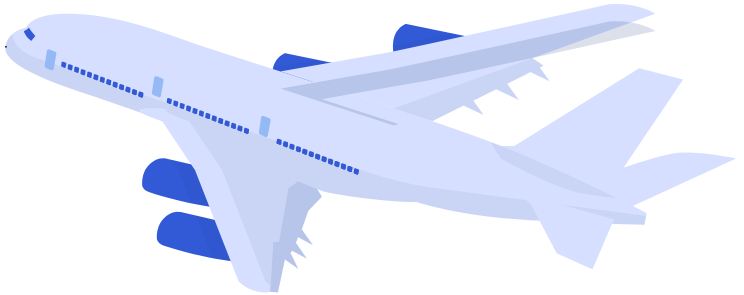
Smart Travel App



Project 4 - Group 4
Project Due: June 12, 2023

Our team

Jose Santos

Dominique Villarreal

Ricky Garcia

Table of contents

1

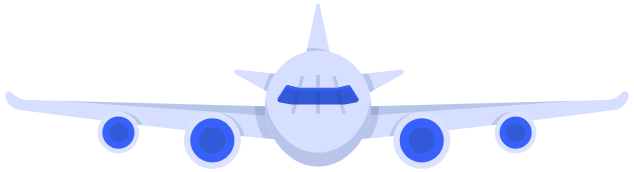
Purpose

The Purpose

Our Original Proposal: To build an app that personalizes travel destination recommendations based on user preferences to help solve the age old question of, “where should I travel to”.

Our Purpose: To simplify a travelers destination options and improve their selection experience

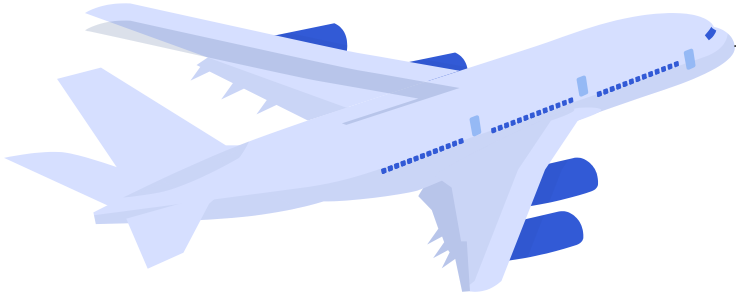
Our Scope: To construct a machine learning algorithm that can personalize travel recommendations based on user preferences and multiple data sources such as: travel, flight, location, experiences/events, demographic information, etc.



2,290,000,000

of Domestic Trips Booked in 2021

TOURISM ACADEMY



$38.65B

2021

$48.53B

2022 (forecast)

$32.96B

2020

TOURISM ACADEMY

Data Pulls

2

Yelp Fusion API

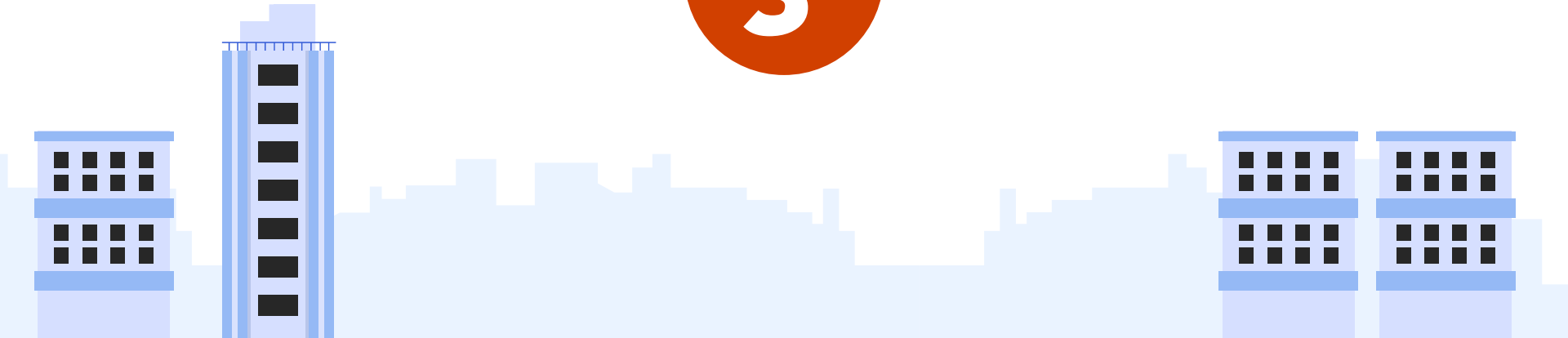
The one stop shop for business data.

Calling the data

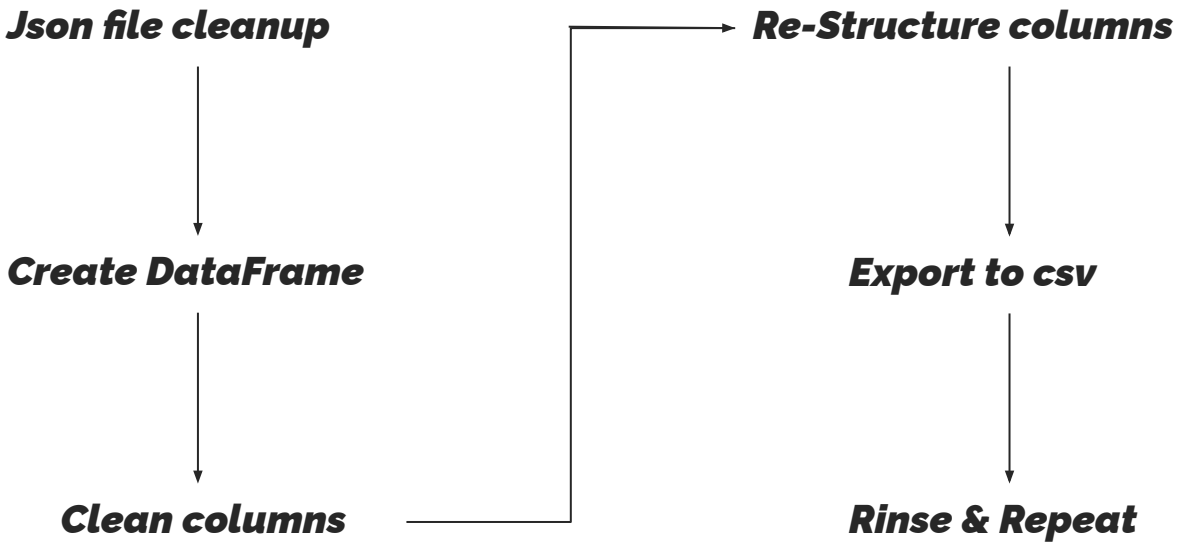
- Identify the ***parent aliases*** we believed would best suit our project.
- Individually call each parent alias for each city. (Denver, Miami, and New York)
- The API limits each call to ***50*** rows of data, so we use a loop to call up to ***1000*** rows of data (when available) and included a ***2.4 second rest***
- During the loop, the ***loop appends*** the call data to a ***json file*** and is written out to the folder and printed on the screen
- We ***repeat*** this process for each of the selected parent aliases ***for each city***.

Data Cleaning

3



Data Cleanup



Data Frame Cleanup

id	alias	name	image_url	is_closed	url	review_	categories	rating	coordinates	transactions	price	location	phone	display_phone	distance
d5A9FtUA6vJp9dnh7v4K0g	jackalope-arts-arvada	Jackalope Arts	https://s3-media2	FALSE	https://ww	8	[{'alias': 'festivals', 'title': 'Fe	5.0	{'latitude': 39.800724136818	[]	$$	{'address1': '5738 Olde Wadsworth Blvd', 'ad	13239892278	(323) 989-2278	11940.339958964500
9zC9Yz0wWFu3PVVR4UP2Uw	special-occasions-events-denver-3	Special Occasions Events	https://s3-media1	FALSE	https://ww	4	[{'alias': 'venues', 'title': 'Ven	5.0	{'latitude': 39.76623, 'longi	[]		{'address1': '3550 Federal Blvd', 'address2': '	13032222136	(303) 222-2136	5746.85264956664

categories	name	rating	review_count	location	coordinates	city
['Festivals', 'Arts & Crafts', 'Local Flavor']	Jackalope Arts	5.0	8	{'address1': '5738 Olde Wadsworth Blvd', 'address2': '', 'address3':	{'latitude': 39.800724136818026, 'longitude': -105.08124974830933}	Denver
['Venues & Event Spaces', 'Party & Event Planning', 'Festivals']	Special Occasions Events	5.0	4	{'address1': '3550 Federal Blvd', 'address2': '', 'address3': '', 'city': '	{'latitude': 39.76623, 'longitude': -105.02439}	Denver

Data Processing

4

Merging the DataFrames

Create a file path to each cities data frame → ***Merge all three data frames into one***

```
print(denver_df.info())
print(miami_df.info())
print(newyork_df.info())
```

```
Output exceeds the size limit. Open the full output data in a text editor
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6078 entries, 0 to 6077
Data columns (total 7 columns):
```

```
# Counting number of categories in newyork_df
ny_unique_categories = newyork_df['categories'].explode().unique()
ny_num_unique_categories = len(ny_unique_categories)
print("New York number of unique categories:", ny_num_unique_categories)
```

```
Denver number of unique categories: 2197
Miami number of unique categories: 2074
New York number of unique categories: 2142
```

```
# Merging all Dataframes

merge_df = pd.concat([denver_df, miami_df, newyork_df], ignore_index = True)
merge_df
```

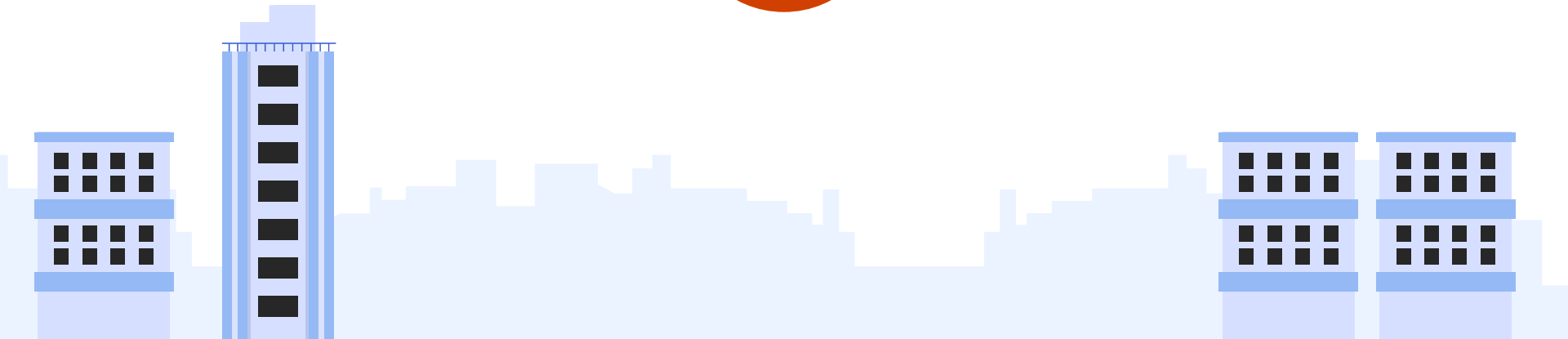
```
# Counting number of categories in merge_df

merge_df_categories = merge_df['categories'].explode().unique()
merge_df_unique_categories = len(merge_df_categories)
print("The merged data frame number of unique categories:", merge_df_unique_categories)
```

Pyt

```
The merged data frame number of unique categories: 5548
```

Machine Learning Models 5



Two ideas

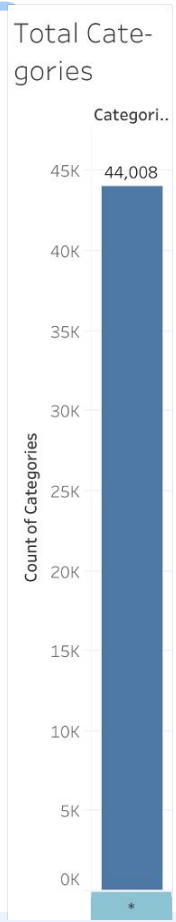
All Data

An attempt to use as much data as we could to predict the city for our random set of selected categories aka User

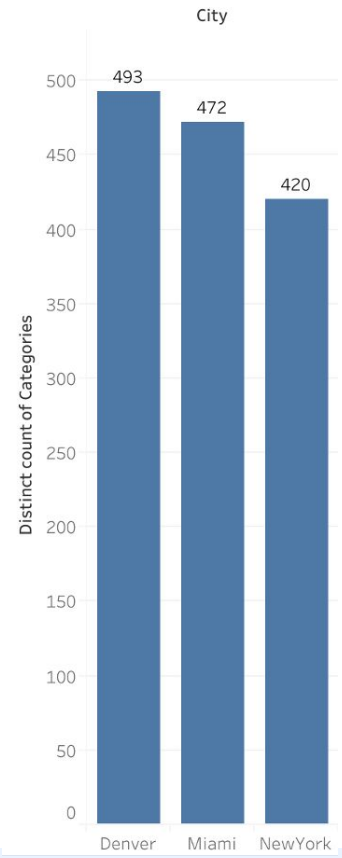
Selected Data

An Attempt to use a cleaner, smaller and more distinct data set to predict off of the randomly selected categories for our User

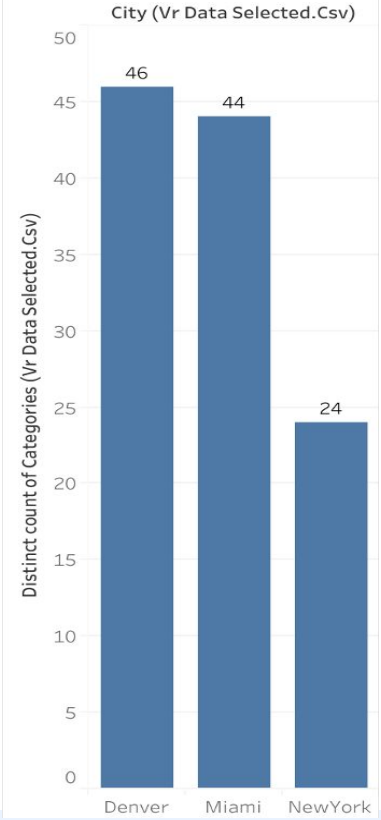
Total Categories



Total Categories x City



Selected Categories x City



MLvAllData

Machine Learning version - All Data

MLvAllData

MLvAllData stands for us using our RandomForestClassifier model on all 44009 rows of data to predict our city and list the top 3 unique experiences.

```
# View df
city_category_predict_df
```

	city	name	rating	review_count	categories	categories_encoded
0	Denver	Jackalope Arts	5.0	8	Festivals	220
0	Denver	Jackalope Arts	5.0	8	Arts & Crafts	38
0	Denver	Jackalope Arts	5.0	8	Local Flavor	348
1	Denver	Special Occasions Events	5.0	4	Venues & Event Spaces	605
1	Denver	Special Occasions Events	5.0	4	Party & Event Planning	416
...	...	...	...	...	...	...
19625	NewYork	Harlem Nights Bar	4.0	185	Bars	58
19625	NewYork	Harlem Nights Bar	4.0	185	Music Venues	384
19626	NewYork	Cardiff Giant	4.0	56	Bars	58
19626	NewYork	Cardiff Giant	4.0	56	Beer, Wine & Spirits	70
19626	NewYork	Cardiff Giant	4.0	56	Cideries	146

44009 rows × 6 columns

MLvAllData

Model Creation

```
# select X & y and reshape df
X = city_category_predict_df['categories_encoded'].values.reshape(-1, 1
y = city_category_predict_df['city']

# Split the data into training and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2

# Create and train the Random Forest Classifier
model = RandomForestClassifier(n_estimators=100, random_state=42)

# View model
model
```

53.82%

Model: Training Data Accuracy

54.58%

Model: Test Data Accuracy

Prediction

We Generate a random sample of 25 categories.

This process is meant to mimic anyone one persons 25 category selection.

The city is predicted based on the random group of 25 categories.

```
# Predict the city based on the random group of 25 categories
prediction = model.predict(sample_categories_encoded)
predicted_city = prediction[0]
predicted_city
```

```
'Miami'
```

```
# Print the readable categories being used to predict the city
print("Categories Used:")
for category in category_names:
    print(category)
```

```
Categories Used:
Barbeque
Music Venues
American (New)
Bars
Beer, Wine & Spirits
Cocktail Bars
Breakfast & Brunch
Tapas/Small Plates
Trainers
Fishing
Parks
Art Galleries
Batting Cages
Italian
Japanese
Ice Cream & Frozen Yogurt
Ramen
Cajun/Creole
Chicken Wings
Gastropubs
Hookah Bars
Beaches
Art Museums
```

Prediction

Predicted City: Miami

Experiences Listed:

- A loop that goes through each of the categories used to predict the city, and that are contained within the city predicted to predict the top 3 unique experiences for each category

Predicted user experiences based on predicted city:

	Sample Category	Predicted City	Experience Name	Rating	Review Count
0	Batting Cages	Miami	Swing Kings	5.0	1
1	Batting Cages	Miami	AllGolf at CB Smith	3.5	83
2	Batting Cages	Miami	Batter's Box - Miami	2.5	15

MLvSelectedData

Machine Learning version - Selected Data

MLvSelectedData

MLvSelectedData stands for us using our RandomForestClassifier model on 350 rows of data to predict our city and list the top 3 unique experiences.

```
# View df
city_category_selected_predict_df
```

	city	categories	categories_encoded
0	Denver	Local Flavor	348
7	Denver	Public Markets	467
20	Denver	Brewing Supplies	97
39	Denver	Brewing Supplies	97
234	Denver	Chiropractors	142
...	...	...	...
19413	NewYork	Fondue	231
19536	NewYork	Music Production Services	383
19578	NewYork	Malaysian	354
19592	NewYork	Vermouth Bars	606
19612	NewYork	South African	532

350 rows × 3 columns

MLvSelectedData

Model Creation

```
# select X & y and reshape df
X = city_category_selected_predict_df['categories_encoded'].values.resha
y = city_category_selected_predict_df['city']

# Split the data into training and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2

# Create and train the Random Forest Classifier
model = RandomForestClassifier(n_estimators=100, random_state=42)

# View model
model
```

```
RandomForestClassifier(random state=42)
```

100%

Model: Training Data Accuracy

100%

Model: Test Data Accuracy

Prediction

We Generate a random sample of 10 categories.

This process is meant to mimic anyone one persons 10 category selection.

The city is predicted based on the random group of 10 categories.

```
# Print the categories being used to predict the city
print("Categories Used:")
for category in category_names:
    print(category)
```

```
Categories Used:
Themed Cafes
Flea Markets
Piercing
Scooter Rentals
Parasailing
Hostels
Champagne Bars
Nicaraguan
Moroccan
Georgian
```

```
# Predict the city based on the random group of 10 categories
prediction = model.predict(sample_categories_encoded)
predicted_city = prediction[0]
predicted_city
```

```
'Miami'
```

Prediction

Predicted City: Miami

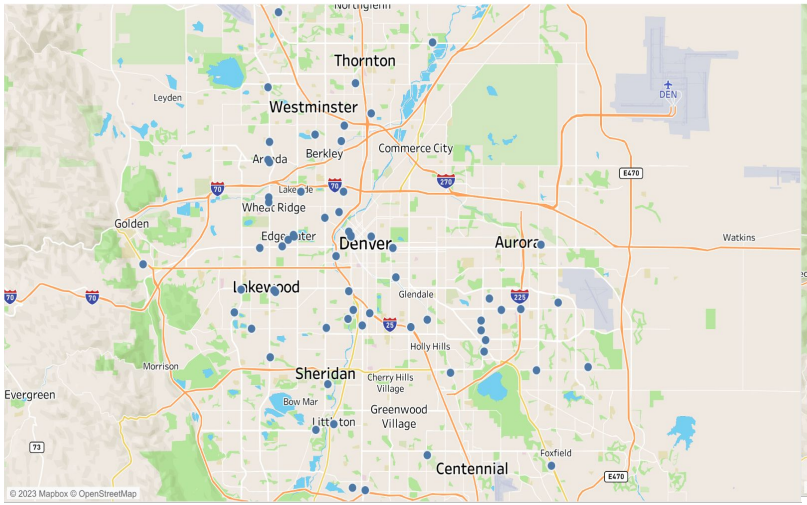
Experiences Listed:

- A loop that goes through each of the categories used to predict the city, and that are contained within the city predicted to predict the top 3 unique experiences for each category

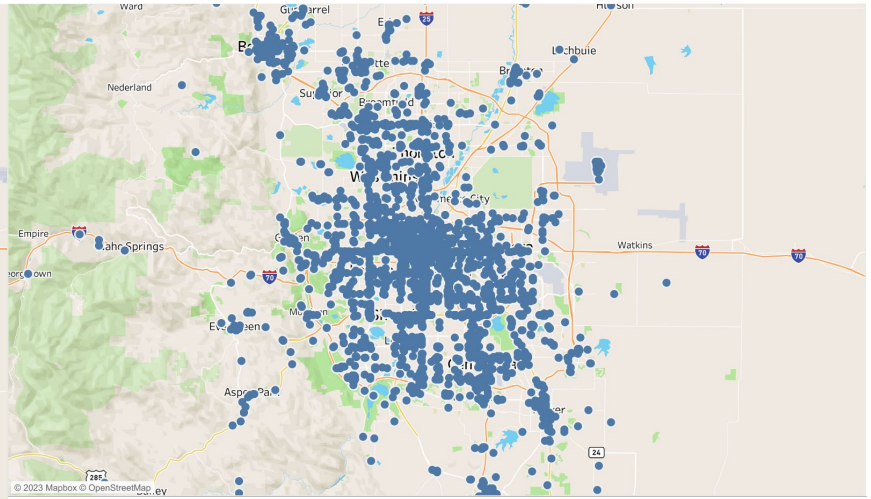
	Sample Category	Predicted City	Experience Name	Rating	Review Count
0	Nicaraguan	Miami	Madrono Restaurant	4.5	360
1	Nicaraguan	Miami	Fritanga Cana Brava	4.5	274
2	Nicaraguan	Miami	Fritanga 505	4.5	48
3	Hostels	Miami	Freehand Miami	3.5	168

Denver - Side x Side

Dener selected

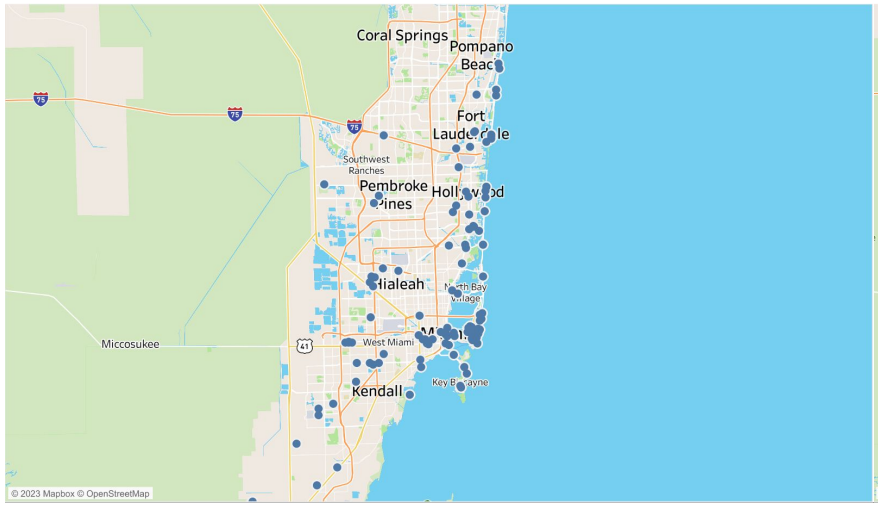


Denver all

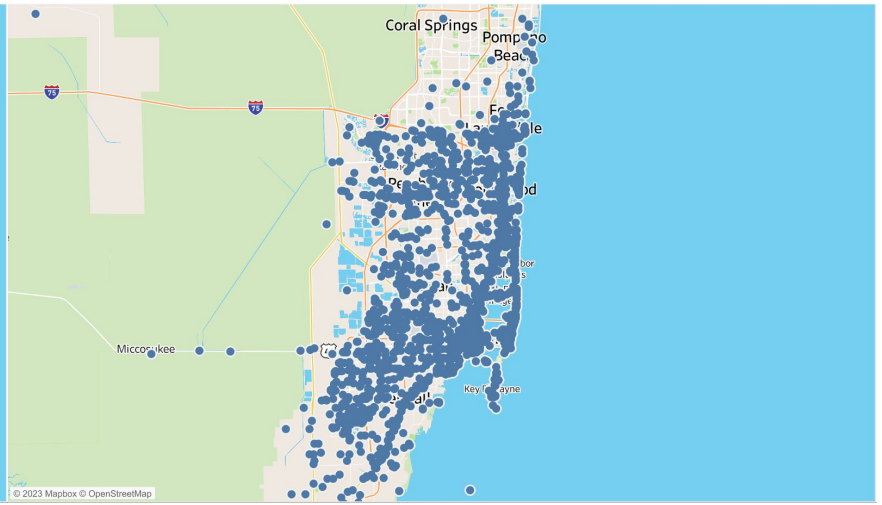


Miami - Side x Side

Miami selected

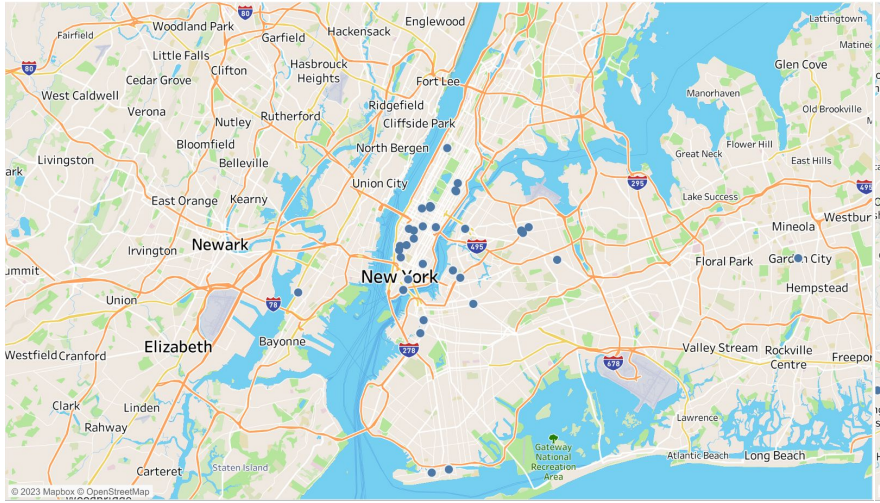


Miami all

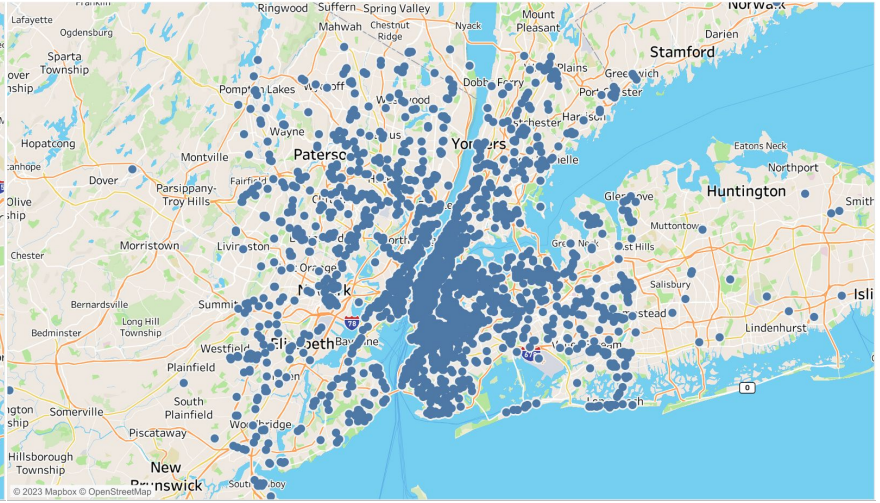


New York - Side x Side

NewYork selected



NewYork all



Practice Makes (Better)

Random Forest n=1,000 Model

76.767%

Model: Test Accuracy

Random Forest n=10,000 Model

77.085%

Model: Test Accuracy

XGB Model

77.630%

Model: Test Accuracy

Conclusion

6

Limitations & Constraints

1. API had limitations on volume of data pulled. 5,000+ rows of data were pulled daily over the span of 3 days to achieve our data set
2. Couldn't figure out how to get more than 50 rows worth of data
3. Each call appended a new dictionary to the json file which we then had to go back and manually manipulate into 1 single dictionary per file
4. Machine Learning Models - Random Forest Model on all data was large and wouldn’t settle at first. Moved on to a manually selected set of unique data. It was during the finalization of the unique set model that we reapplied it to all the data and were able to achieve a 53%+ accuracy on all data.
5. Unique data - we could have used an additional data set with more city unique data to help achieve a higher accuracy rate.
6. Time - with more time we could have pulled & cleaned more data to house more unique categories within our dataset and give the model more to calculate on.

Future Uses

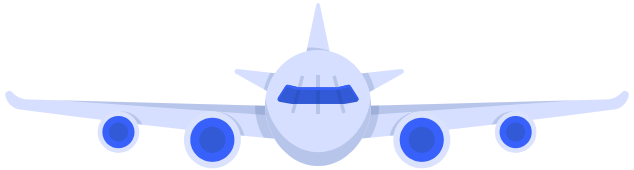
- Build an App that personalizes a users experience based on a lot more data points.
- Expand across the US and Globally
- Add in weather data
- Add itinerary capabilities
- Build in a social platform
- Create a monetized social model that allows users to earn money the more they put into the app
- Build on a blockchain with user wallet
- Add passport functionality

Conclusion

- Conducting data analysis across all categories:
  - Train accuracy: 53.82%
  - Test accuracy: 54.58%
- Hand-selecting unique categories
- Resulting in improved accuracy:
  - Train accuracy: 100%
  - Test accuracy: 100%
- Importance of unique category filtering:
  - Optimizes accuracy and ensures reliable results
- There’s opportunity for refining our analysis process and narrowing down categories:
  - This would unlock valuable insights with higher precision

“Data are just summaries of thousands of stories—tell a few of those stories to help make the data meaningful.”

—Dan Heath, bestselling author



Thank you!

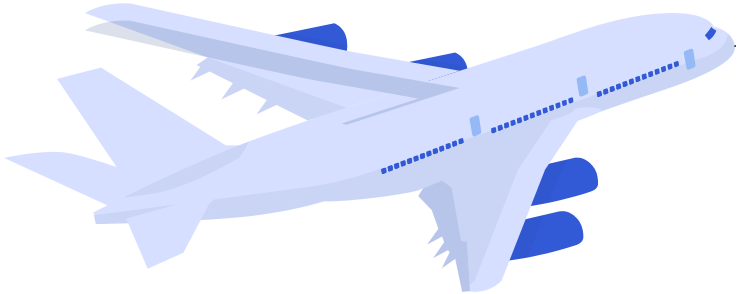
References

1. https://docs.developer.yelp.com/reference/v3_business_search

2. https://blog.tourismacademy.org/us-tourism-travel-statistics-2020-2021#:~:text=US%20domestic%20travel%20increased%20by%20%2B2%25%20YTD%20in,travel%20in%202019%20accounted%20for%20464%20million%20trips.

3. https://quotefancy.com/quote/1716228/Dan-Heath-Data-are-just-summaries-of-thousands-of-stories-tell-a-few-of-those-stories-to

4. https://slidesgo.com/theme/flying-airplane#search-Travel&position-6&results-285



Thanks!

Do you have any questions?
addyouremail@freepik.com
+91 620 421 838
yourwebsite.com