Invest in Space Industry

Yilin Elaine Liu

July 8, 2022

OUTLINE

- Executive Summary
- Introduction
- Methodology
  - Data Collection – SpaceX API
  - Data Collection – Web Scraping
  - Data Wrangling
  - EDA with Data Visualization
  - EDA with SQL
  - Build an Interactive Map with Folium
  - Dashboard with Plotly Dash
  - Predictive Analysis (Classification)
- Results
  - Data insights from SQL
  - Visualization – Charts
  - All launch sites on Folium Map
  - Dashboard
  - Predictive Analysis
- Conclusion
- Appendix

EXECUTIVE SUMMARY

- The SpaceX Falcon 9 First Stage Analysis
- Methodologies
  - Data collection using web scraping & SpaceX API
  - Exploratory Data Analysis(EDA), including data wrangling, data visualization, and interactive visual analytics
  - Machine learning prediction
- Summary of all results
  - Collected valuable data from public sources
  - Identify which features are the best to predict
  - Compare Logistic Regression, SVM, Decision Tree, KNN

INTRODUCTION

- Project background
  - Space X advertises Falcon 9 rocket launches on its website with a cost of 62 million
  - Less than other competitors' 2.67 times
- Problems
  - Predict SpaceX Falcon 9 First Stage Landing
- Challenges
  - What factors determine if rockets will land successfully ?
  - Which features are the most correlated to determine the success rate of a successful landing ?
  - What conditions does SpaceX have to achieve to get the best results and ensure the best rocket success landing rate?

METHODOLOGY

METHODOLOGY

- Data collection methodology
  - SpaceX Rest API
  - Web Scraping from Wikipedia
- Perform data wrangling
  - Perform EDA(Exploratory Data Analysis) and determine Training Labels
- Perform exploratory data analysis(EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune and evaluate classification models

```
spacex_url="https://api.spacexdata.com/v4/launches/past"
```

```
response = requests.get(spacex_url)
```

```
# Use json_normalize meethod to convert th
data = pd.json_normalize(response.json())
```

```
launch_dict = {'FlightNumber': list(data['flight_number']),
'Date': list(data['date']),
'BoosterVersion':BoosterVersion,
'PayloadMass':PayloadMass,
'Orbit':Orbit,
'LaunchSite':LaunchSite,
'Outcome':Outcome,
'Flights':Flights,
'GridFins':GridFins,
'Reused':Reused,
'Legs':Legs,
'LandingPad':LandingPad,
'Block':Block,
'ReusedCount':ReusedCount,
'Serial':Serial,
'Longitude': Longitude,
'Latitude': Latitude}
```

```
data_falcon9.isnull().sum()
```

Data Collection – SpaceX AP → Request to get SpaceX API → To make the requested JSON results more consistent → *Use json_normalize meethod to convert the json result into a dataframe* → The data from these requests will be stored in lists → Combine the columns into a dictionary → **Filter the dataframe to only include Falcon 9 launches** → **Dealing with Missing Values** → Export new data to a **CSV**

```
static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.
```

We should see that the request was successfull with the 200 status response code

```
response.status_code
```

```
#Global variables
BoosterVersion = []
PayloadMass = []
Orbit = []
LaunchSite = []
Outcome = []
Flights = []
GridFins = []
Reused = []
Legs = []
LandingPad = []
Block = []
ReusedCount = []
Serial = []
Longitude = []
Latitude = []
```

```
# Hint data['BoosterVersion']!='Falcon 1'
data_falcon9 = data2[data2['BoosterVersion'] != 'Falcon 1']
```

Now that we have removed some values we should reset the FlgihtNumber column

```
data_falcon9.loc[:,'FlightNumber'] = list(range(1, data_falcon9.shape[0]+1))
data_falcon9
```

```
data_falcon9.to_csv('dataset_part_1.csv', index=False)
```

```
static_url = "https://en.wikipedia.c
```

Next, request the HTML page from the abov

TASK 1: Request the Falcon9 La

First, let's perform an HTTP GET method to

```
# use requests.get() method with the
# assign the response to a object
response = requests.get(static_url)
```

```
column_names = []

# Apply find_all() function with `th` eleme
tc = first_launch_table.find_all('th')
# Iterate each th element and apply the pro
# Append the Non-empty column name (`if nam
for th in tc:
    name = extract_column_from_header(th)
    if name is not None and len(name) > 0:
        column_names.append(name)
```

```
extracted_row = 0
#Extract each table
for table_number,table in enumerate(soup.find_all('table',"wikitable plainrowheaders collapsible")):
    # get table row
    for rows in table.find_all("tr"):
        #check to see if first table heading is as number corresponding to launch a number
        if rows.th:
            if rows.th.string:
                flight_number=rows.th.string.strip()
                flag=flight_number.isdigit()
        else:
            flag=False
        #get table element
        row=rows.find_all('td')
        #if it is number save cells in a dictonary
        if flag:
            extracted_row += 1
            # Flight Number value
            # TODO: Append the flight_number into launch_dict with key `Flight No.`
            launch_dict['Flight No.'].append(flight_number)
            #print(flight_number)
            datatimelist=date_time(row[0])

            # Date value
            # TODO: Append the date into launch_dict with key `Date`
            date = datatimelist[0].strip(',')
            launch_dict['Date'].append(date)
            #print(date)

            # Time value
            # TODO: Append the time into launch_dict with key `Time`
            time = datatimelist[1]
            #print(time)
            launch_dict['Time'].append(time)
```

```
df.to_csv('spacex_web_scraped.csv', index=False)
```

Data Collection – Web Scraping → Request the Falcon 9 Launch HTML page → Create a BeautifulSoup Object → Find all tables & *print the third table* → Extract column name one by one → Create a dictionary → Appending data to keys → *All arrays must be of the same length* → Convert dictionary to dataframe then export csv

```
# Use the find_all function in the BeautifulSoup object, with element type `table`
# Assign the result to a list called `html_tables`
html_tables = soup.find_all('table')
```

Starting from the third table is our target table contains the actual launch records.

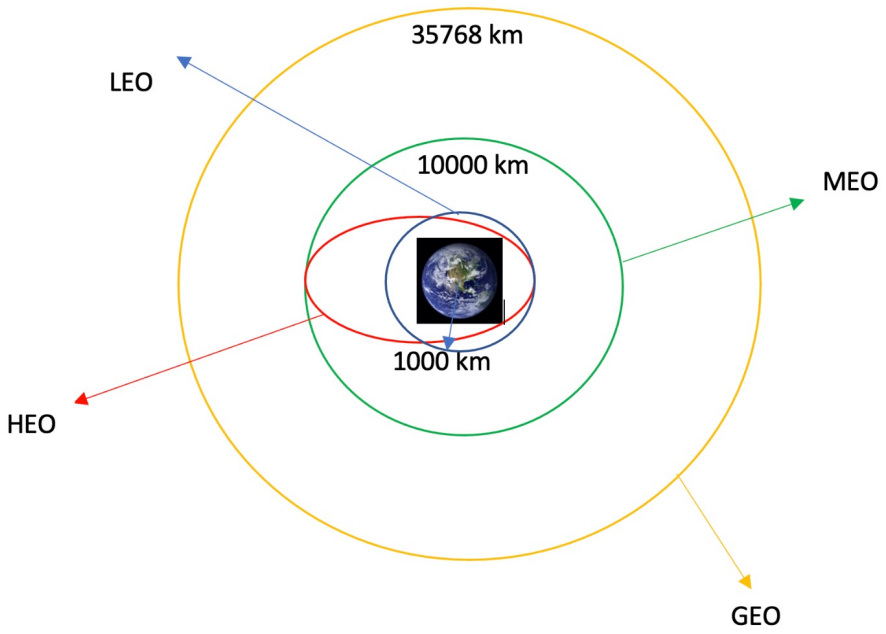
```
# Let's print the third table and check its content
first_launch_table = html_tables[2]
print(first_launch_table)
```

```
launch_dict= dict.fromkeys(column_names)

# Remove an irrelvant column
del launch_dict['Date and time ( )']

# Let's initial the launch_dict with eac
launch_dict['Flight No.'] = []
launch_dict['Launch site'] = []
launch_dict['Payload'] = []
launch_dict['Payload mass'] = []
launch_dict['Orbit'] = []
launch_dict['Customer'] = []
launch_dict['Launch outcome'] = []
# Added some new columns
launch_dict['Version Booster']=[]
launch_dict['Booster landing']=[]
launch_dict['Date']=[]
launch_dict['Time']=[]
```

```
#All arrays must be of the same length
for k in launch_dict.keys():
    print("Key {} => Len {}".format(k, len(k)))
    launch_dict
```

```
# Use BeautifulSoup() to create a BeautifulS
soup = BeautifulSoup(response.text, 'html')
```

Data Wrangling



Perform EDA(Exploratory Data Analysis) and determine Training Labels

Outcomes into Training Labels

- 1 = the booster successfully landed
- 0 = the booster unsuccessful landed

Calculate the number of launches on each site

```
df['LaunchSite'].value_counts()
```

Calculate the number and occurrence of each orbit

```
df['Orbit'].value_counts()
```

Calculate the number and occurrence of mission outcome per orbit type

```
landing_outcomes = df['Outcome'].value_counts()
```

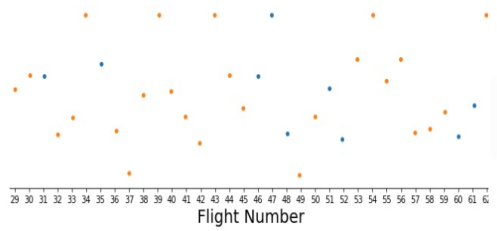
Create a landing outcome label from outcome column

```
landing_class = df['Outcome'].map(lambda x: 0 if x in bad_outcomes else 1)
```

Export dataframe to csv

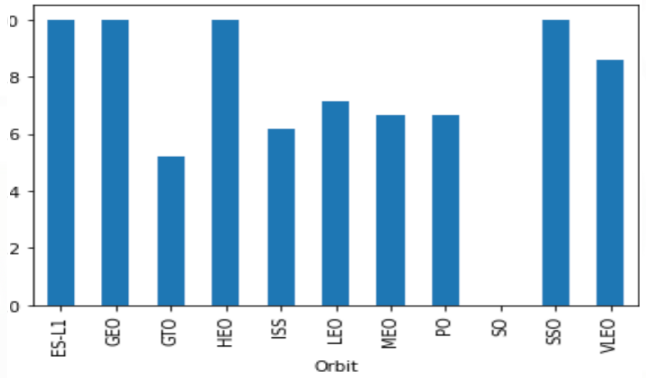
```
df.to_csv("dataset_part_2.csv", index=False)
```

EDA with Data Visualization



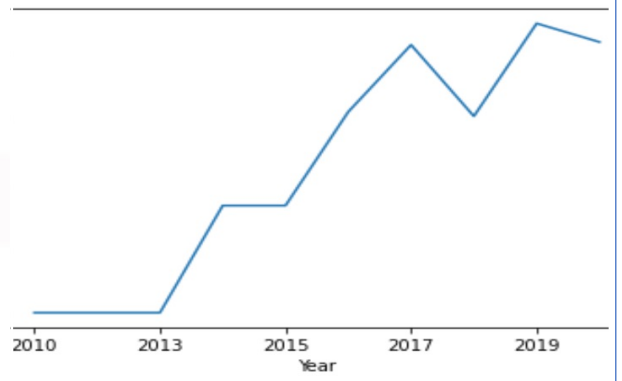
Scatter Graphs

PayloadMass vs. FlightNumber
LaunchSite vs. FlightNumber
LaunchSite vs. PayloadMass
Orbit Vs. FlightNumber
Orbit vs. PayloadMass



Bar Graph

The success rate of each orbit



Line Graph

The success rate of each orbit

EDA with SQL

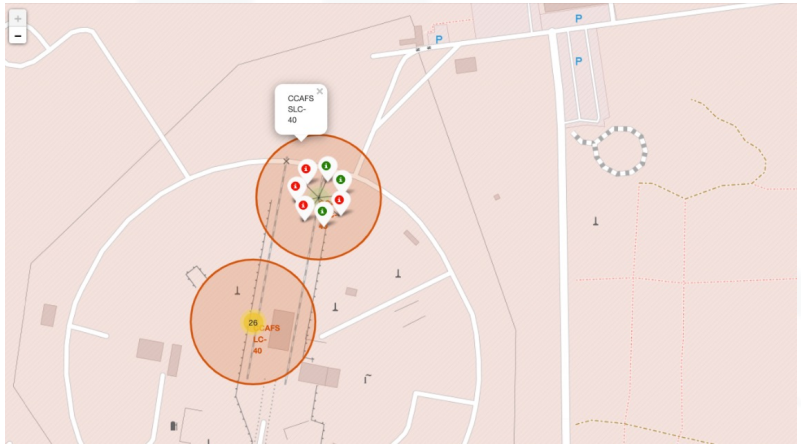
Performed EDA with SQL queries to gather information from a DB2 database

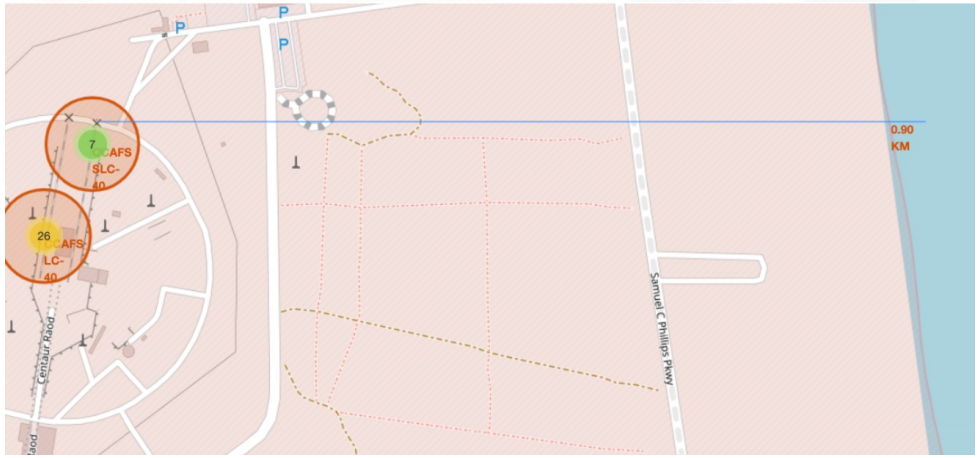
- Display the names of the unique launch sites in the space mission
- Display 5 records where launch sites begin with the string ‘CCA’
- Display the total payload mass carried by booster launched by NASA(CRS)
- Display average payload mass carried by booster version F9 v1.1
- List the date when the first successful landing outcome in ground pad was achieved
- List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
- List the total number of successful and failure mission outcomes
- List the names of the booster versions which have carried the maximum payload mass
- List the records which will display the month names, failure landing_outcomes in drone ship, booster versions, launch_site for the months in year 2015
- Rank the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order

Build an Interactive Map with Folium

Markers, circles, lines and marker clusters were used with Folium maps

- Markers indicate points like launch sites
- Mark the success/ failed launches for each site on the map
  - Green = launch outcome success 1
  - Red = launch outcome failure 0
- Add a MousePosition on the map to get coordinate for a mouse over a point on the map
- Mark down a point on the closest coastline using MousePosition and calculate the distance between the coastline point and the launch site
- Create a folium.Marker to show the distance
- Draw a PolyLine between a launch site to the selected coastline point

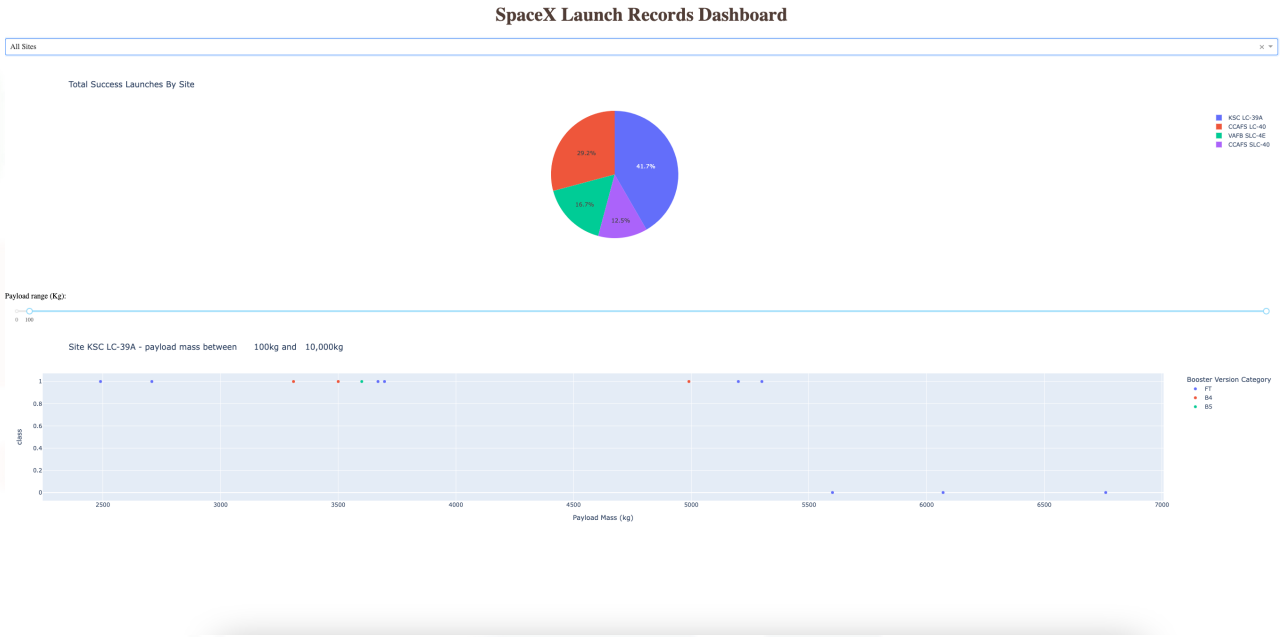




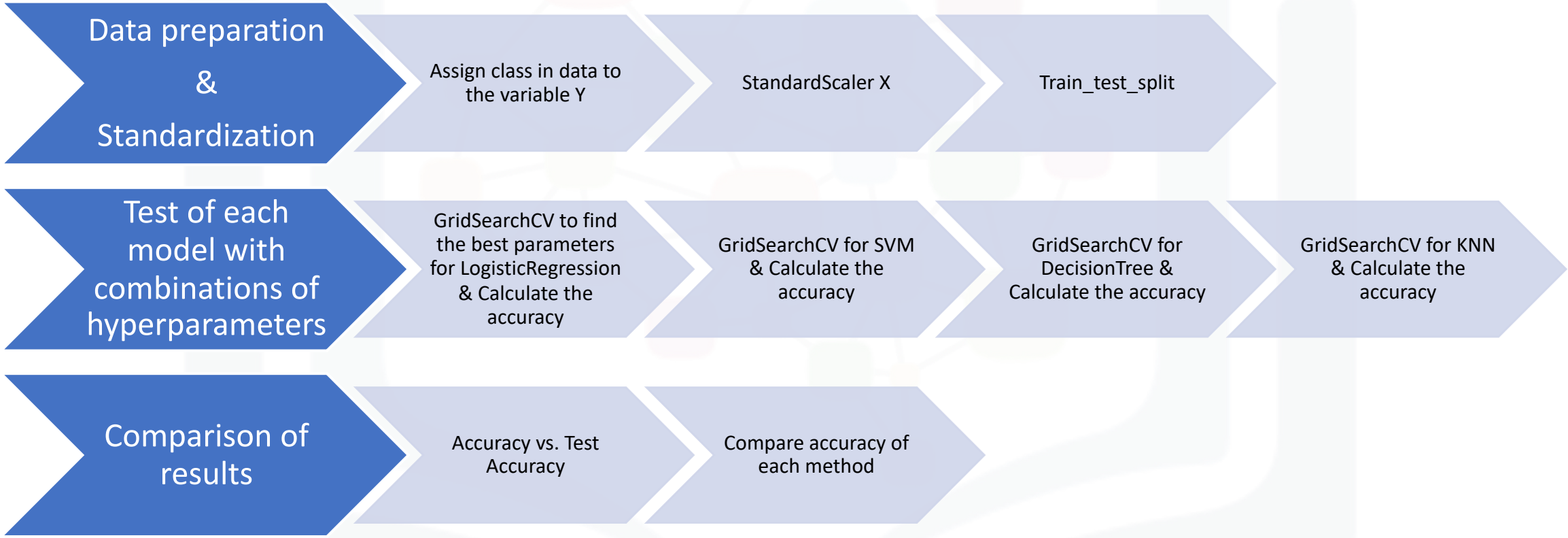
Dashboard with Plotly Dash

- The following graphs and plots were used to visualize data
  - Percentage of launches by site
  - Payload range

- This combination allowed to quickly analyze the relation between payloads and launch sites, helping to identify where is best place to launch according to payloads



Predictive Analysis (Classification)



RESULTS

Data insights from SQL

- Four launch sites in the space mission
  - CCAFS LC-40
  - CCAFS SLC-40
  - KSC LC-39A
  - VAFB SLC-4E
- Five samples of Cape Canaveral launches

DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-05-22	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Data insights from SQL

- The total payload mass is 111268
- The average payload mass carried by booster version F9 V1.1 is 2928
- The first successful landing outcome in ground date is 2015-12-22
- The name of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
  - F9 FT B1021.2
  - F9 FT B1031.2
  - F9 FT B1022
  - F9 FT B1026
- The total number of successful and failure mission outcomes
  - Failure (in flight): 1
  - Success: 99

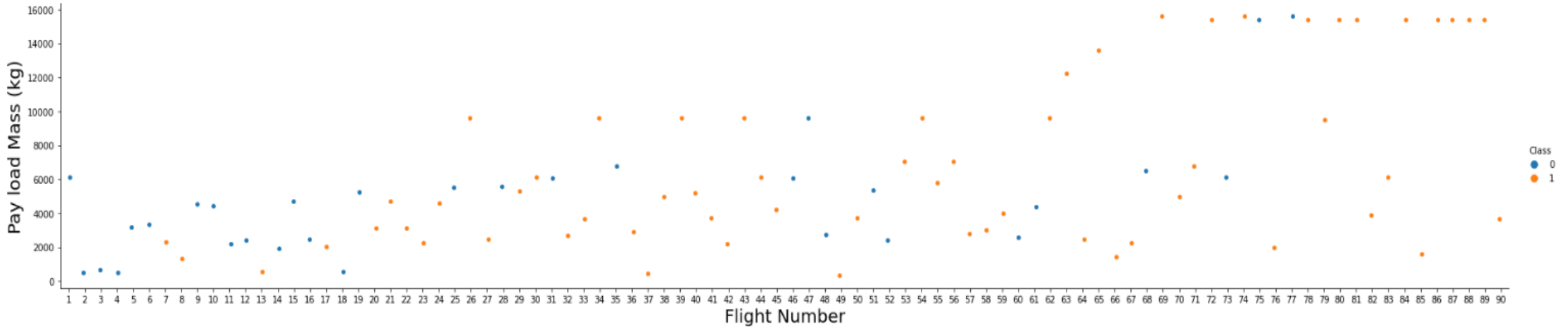
Data insights from SQL

- The top three of the booster versions which have carried the maximum payload mass:
  - F9 B5 B1048.4
  - F9 B5 B1048.5
  - F9 B5 B1049.4
- The failure landing outcomes in drone ship

MONTH	landing__outcome	booster_version	launch_site
1	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
4	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

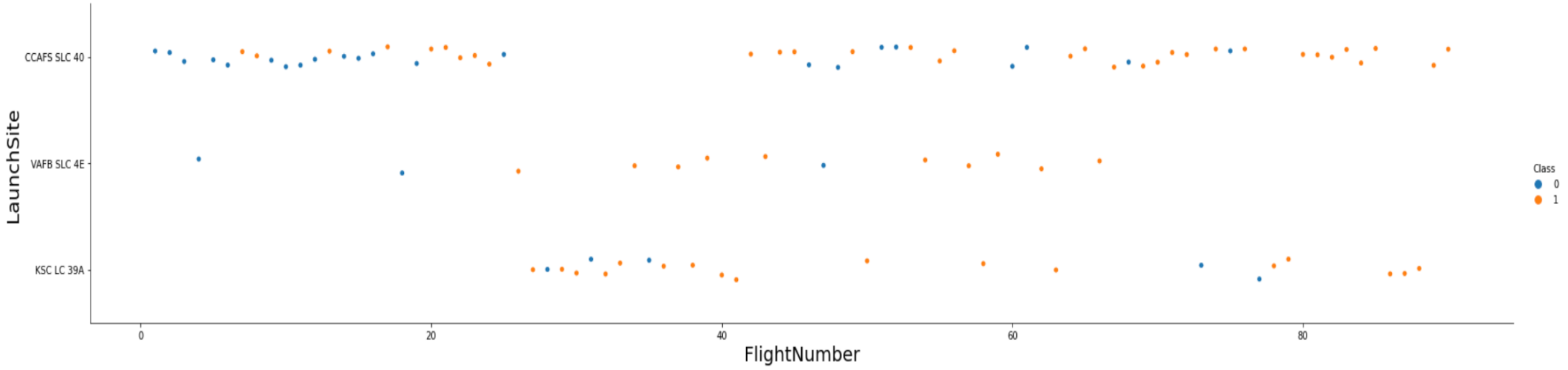
Exploring and Preparing Data

- CCAFS LC-40 has a success rate of 60%
- KSC LC-39A and VAFB SLC 4E have a success rate of 77%



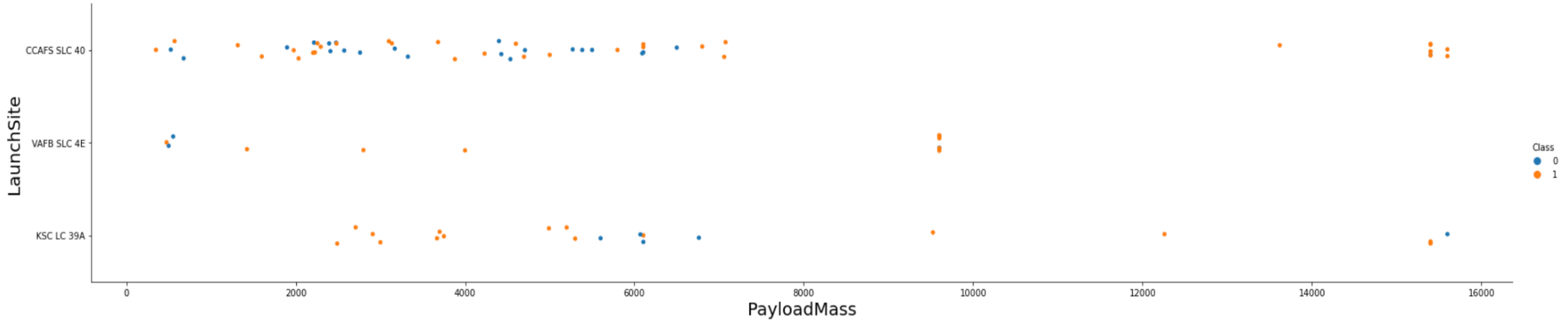
Exploring and Preparing Data

- CCAFS SLC 40 has a higher success rate and most flight numbers



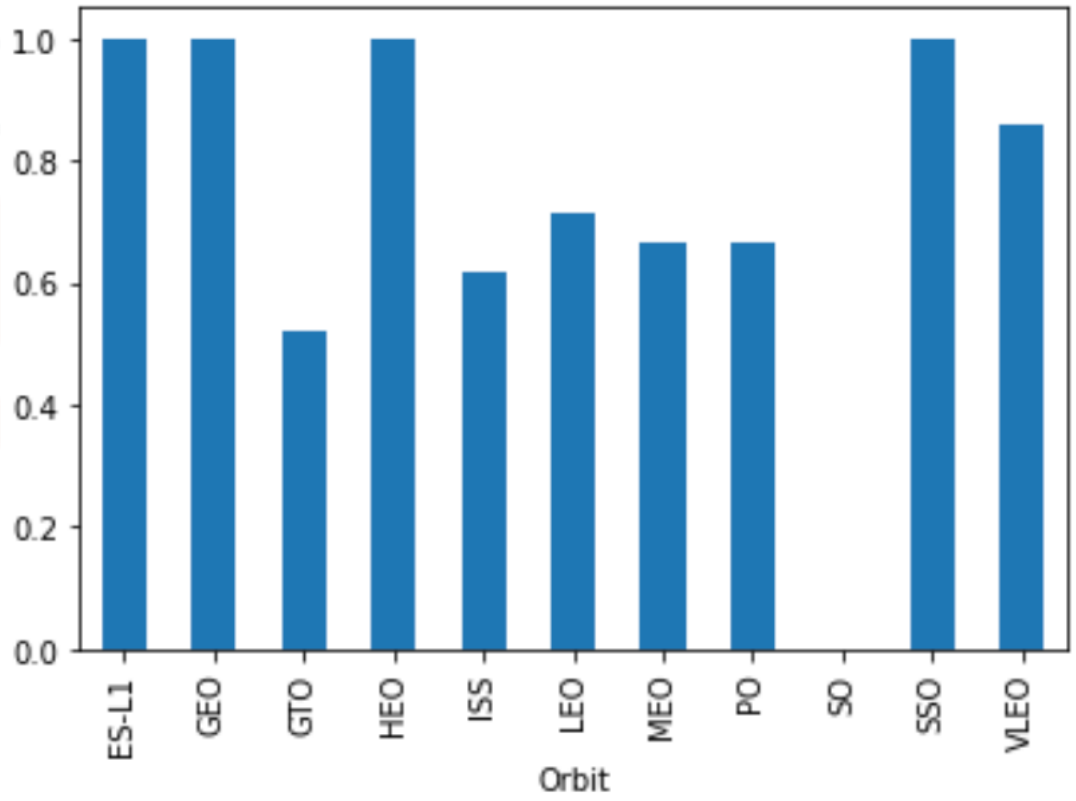
Exploring and Preparing Data

- The VAFB-SLC launch site there are no rockets launched for heavy payload mass (greater than 10000)



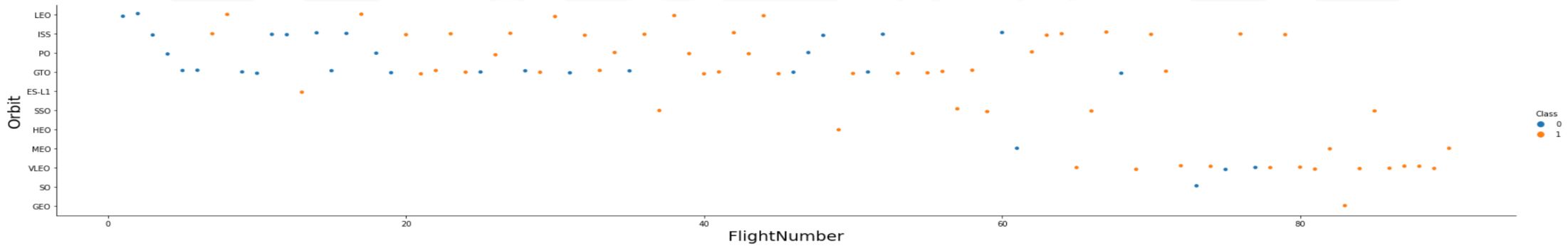
Exploring and Preparing Data

- The biggest success rates happens to orbits
  - ES-L1
  - GEO
  - HEO
  - SSO
- Followed by
  - VLEO
  - LFO



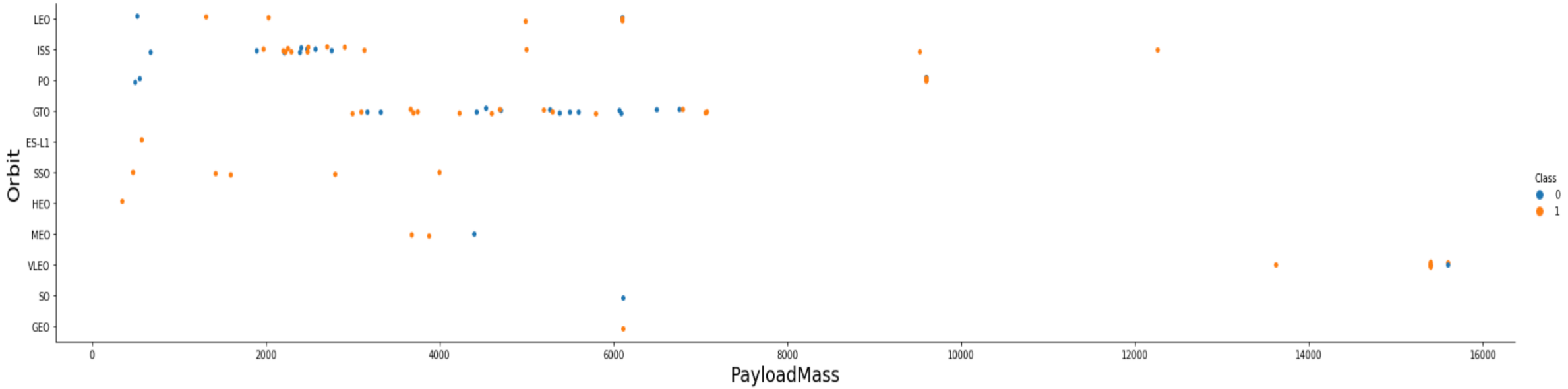
Exploring and Preparing Data

- The LEO orbit the success appears related to the number of flights
- There seems to be no relationship between flight number when in GTO orbit



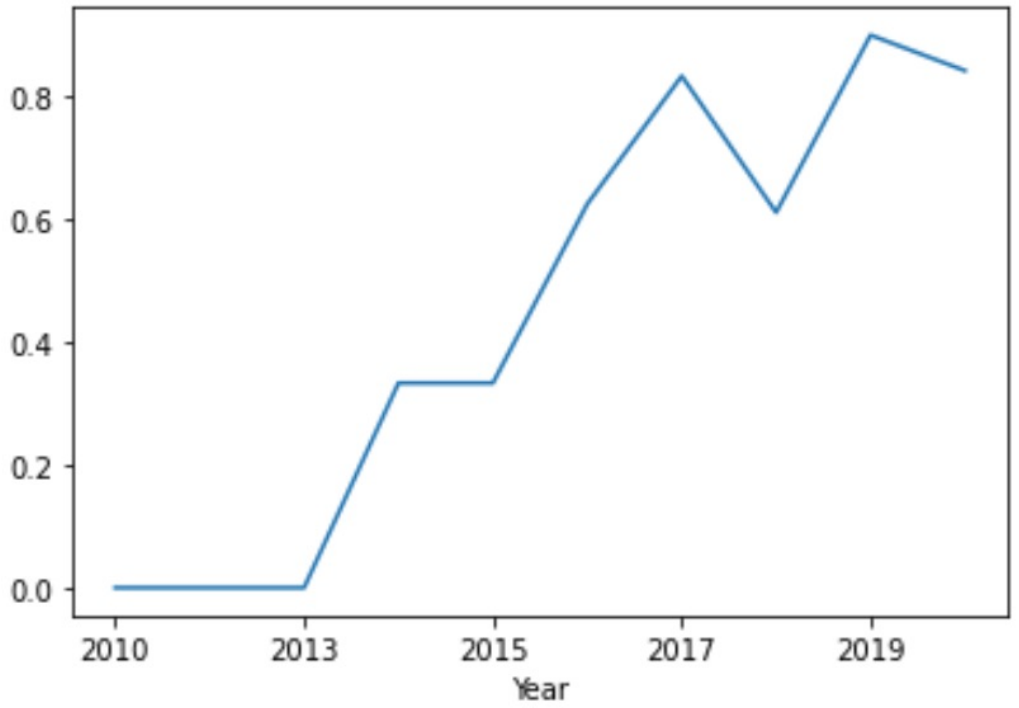
Exploring and Preparing Data

- Polar, LEO, ISS have more positive landing rate

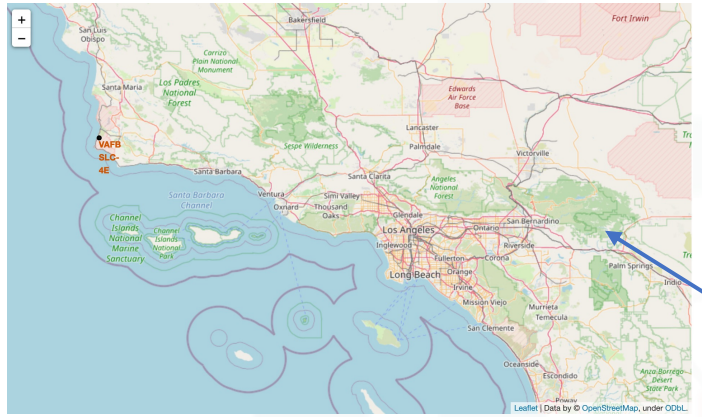


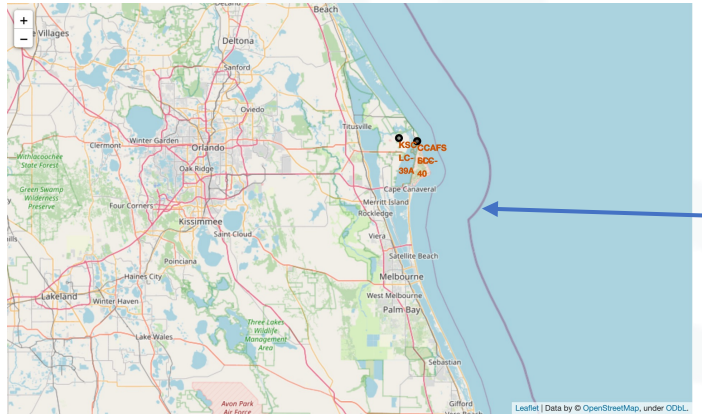
Exploring and Preparing Data

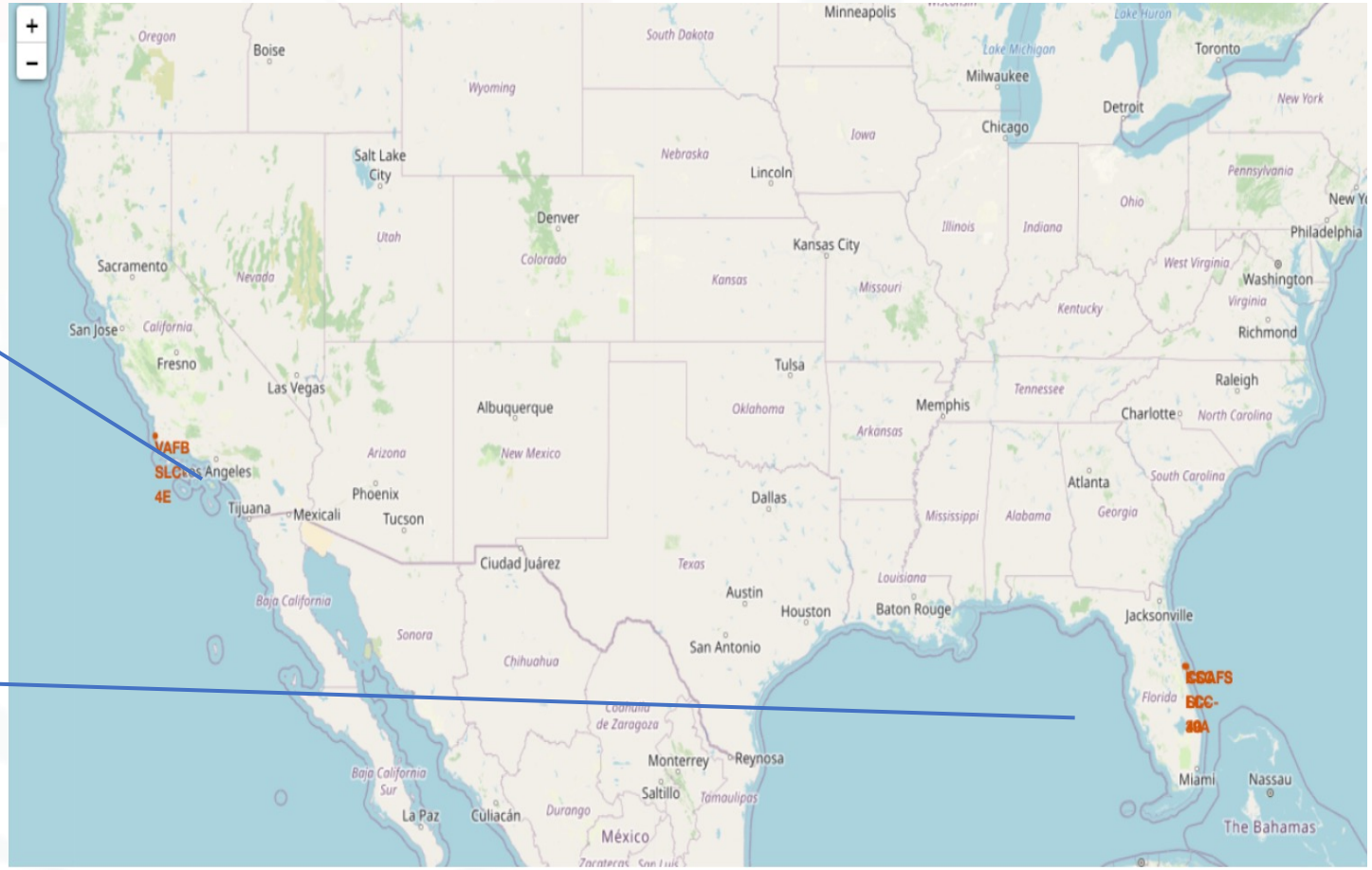
- Since 2013, the success rate kept increasing till 2020
- The first three years were a period of adjusts and improvement of technology



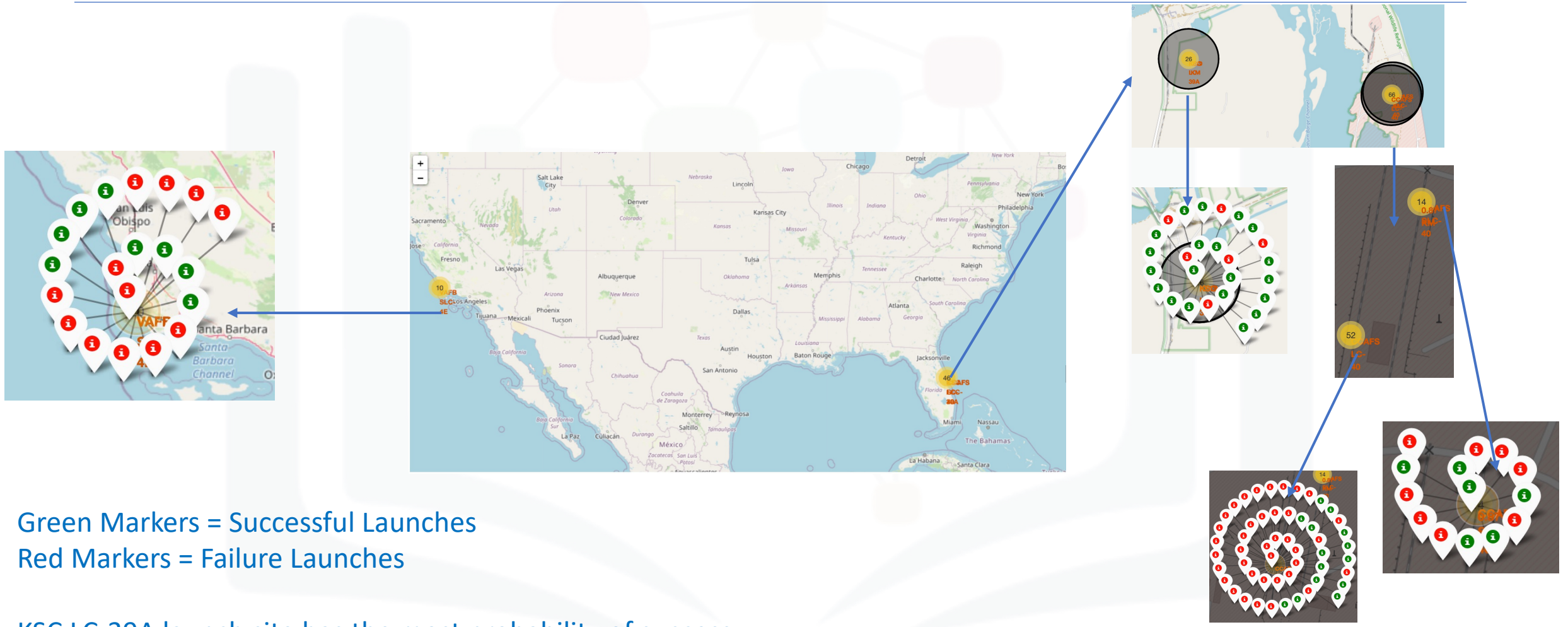
All Launch Sites on Folium Map





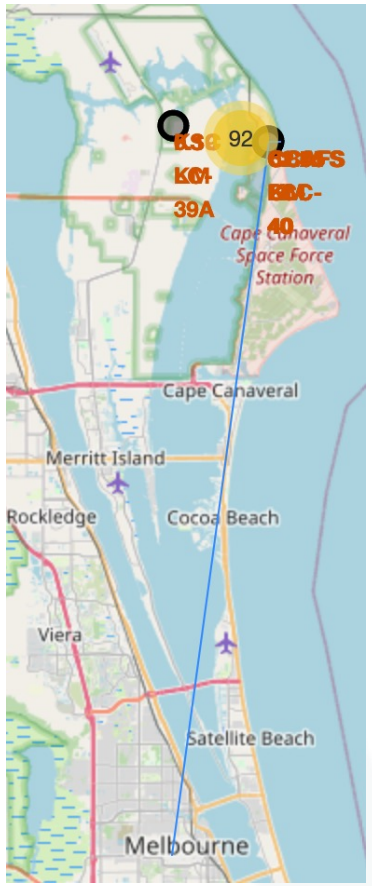


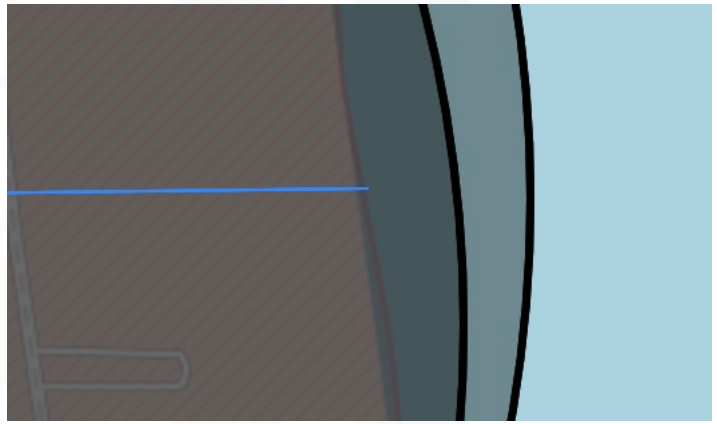
Color labels for each site on the map



KSC LC-39A launch site has the most probability of success

Launch Site Distances from railways/highways/cities/coastlines

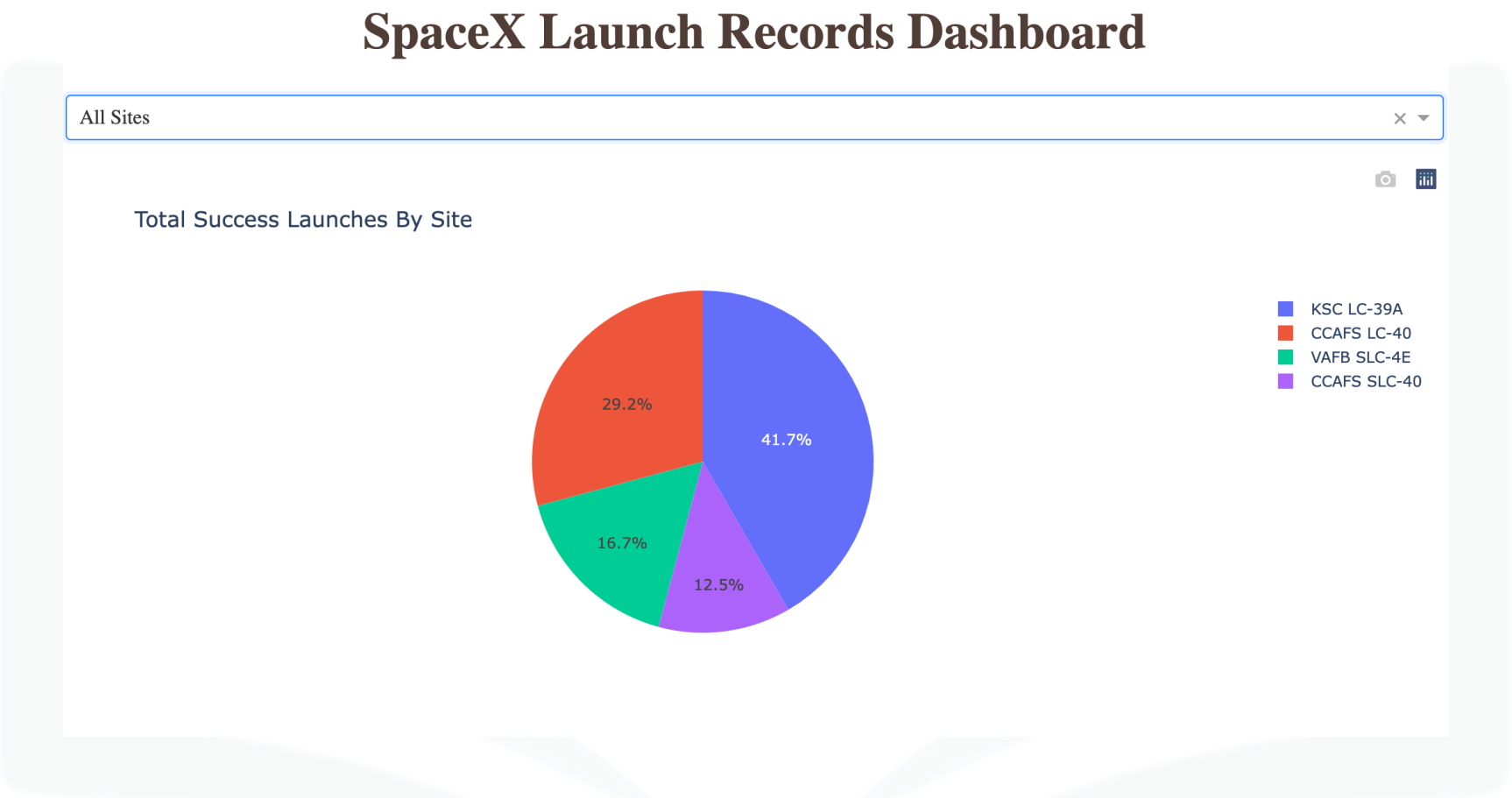




DASHBOARD

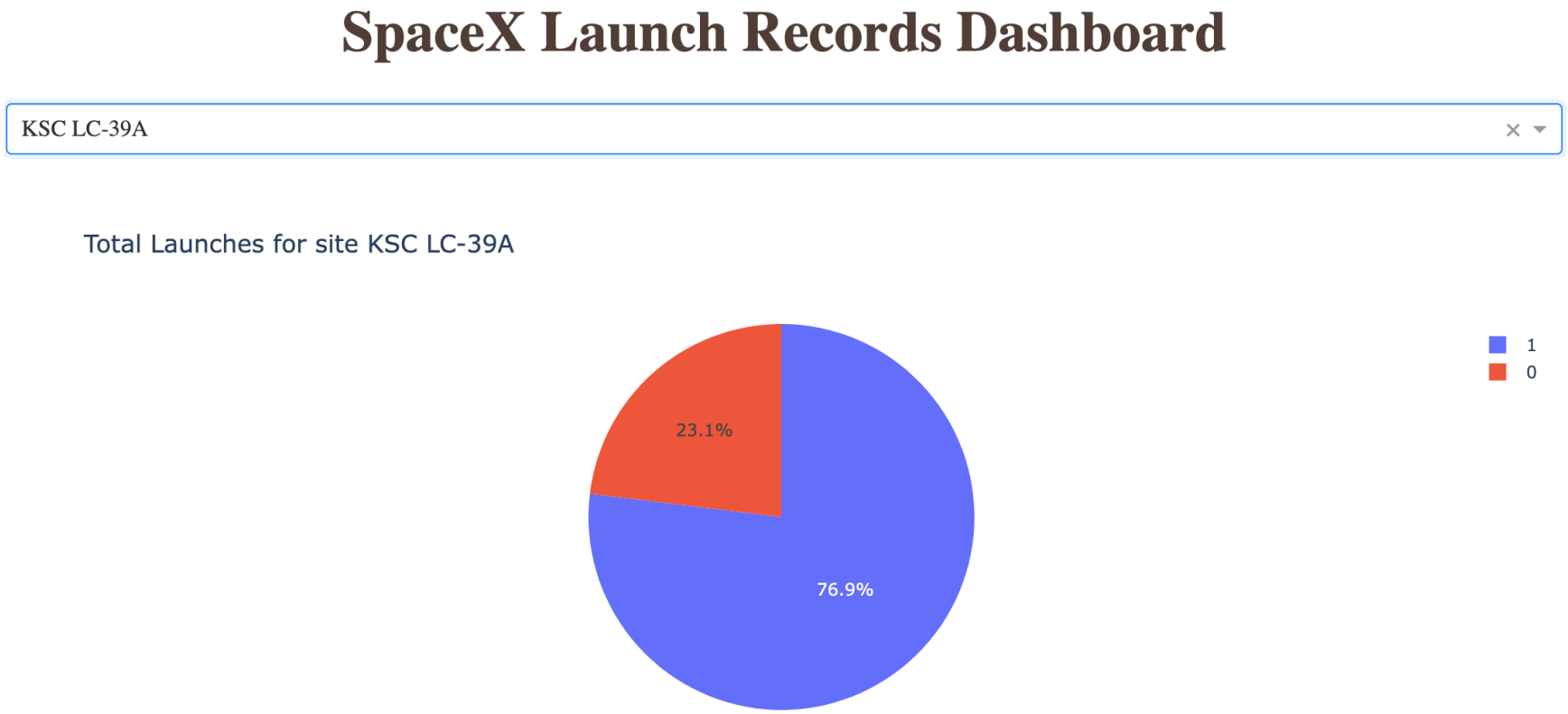
https://yilinliu520-8050.theiadocker-2-labs-prod-theiak8s-4-tor01.proxy.cognitiveclass.ai/

Successful Launches by Site



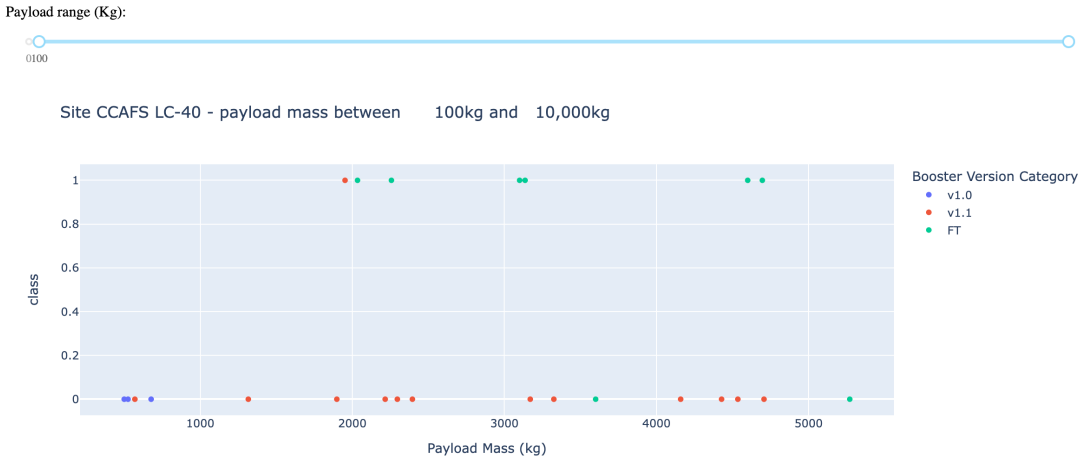
The place from where launches are done seems to be a very important factor of success of missions

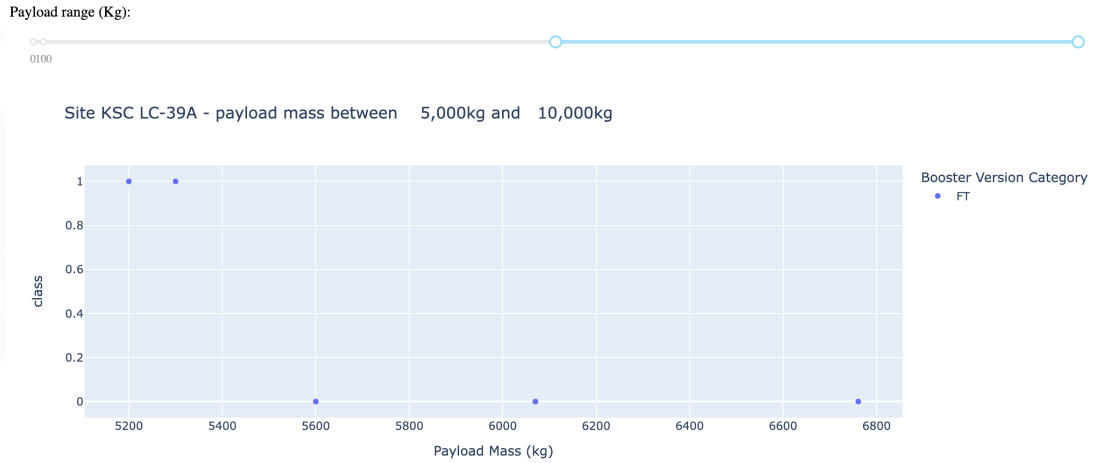
Pie Chart with Highest Success Ratio



KSC LC-39A has 76.9% of success rate while getting 23.1% of failure rate

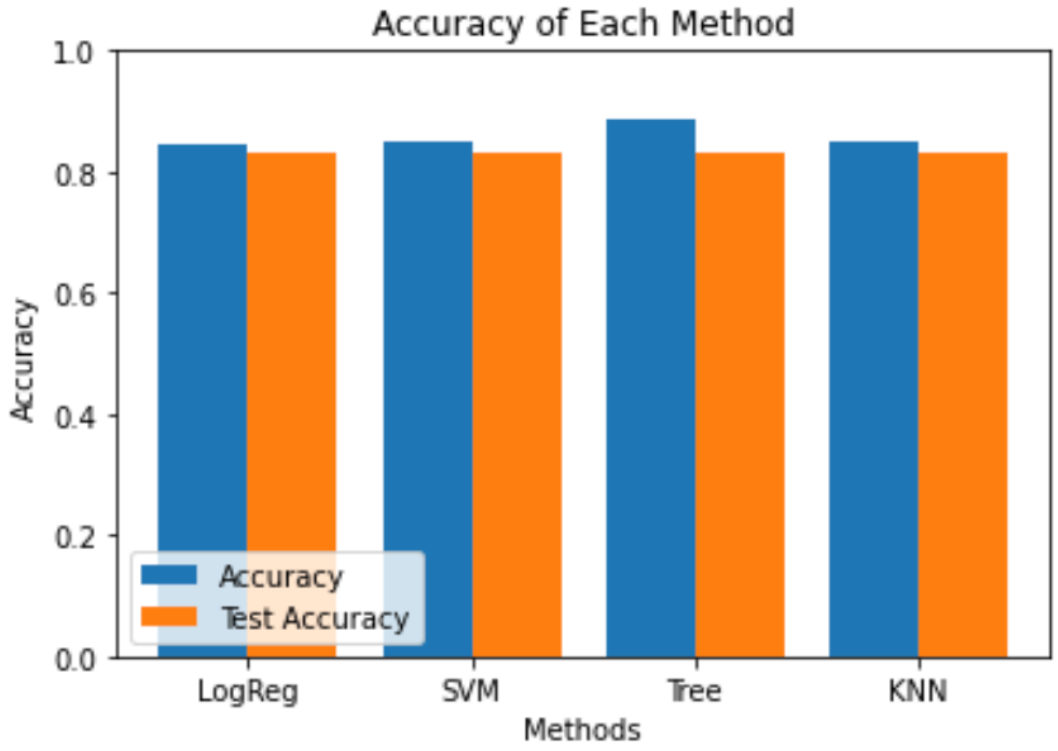
Scatter Plot of Payload vs. Launch outcome for KSC-39A





Predictive Analysis (Classification)

Model	Accuracy	Test Accuracy
Logistic Regression	0.846	0.833
SVM	0.848	0.833
Decision Tree	0.885	0.833
KNN	0.848	0.833

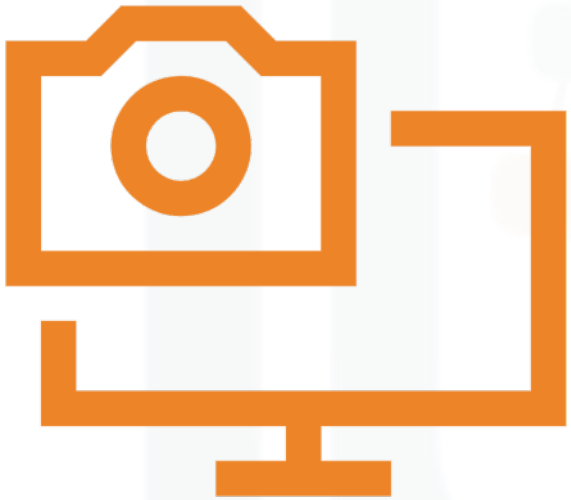


CONCLUSION

CONCLUSION

- More the number of flights increases, more the success rate increases at a launch site
- Orbits ‘Es-L1’, ‘GEO’, ‘ HEO’, ‘SSO’ had the most success rate
- Success rate keep increasing since 2013 till 2020
- KSC LC-39A had the most successful launches from all sites
- Low payloads mass(kg) perform better than the heavier payloads
- Most payload mass with highest success rate is between 1952 kg and 5300 kg
- FT is the Booster version with highest launch success rate
- The Decision Tree Classifier is the best machine learning algorithm for this project with provided dataset

APPENDIX



- a Wikipedia page titled List of Falcon 9 and Falcon Heavy launches
  - https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches
- SpaceX API
  - https://api.spacexdata.com/v4/launches/past

THANK YOU!