IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Dan
1/11/2024

Outline

- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion
- Appendix

Executive Summary

- Summary of methodologies
  - Data collection
  - Data wrangling
  - EDA with data visualization
  - EDA with SQL
  - Building an interactive map with Folium
  - Building a dashboard with Plotly Dash
  - Predictive analysis
- Summary of all results
  - EDA results
  - Interactive analytics
  - Predictive analytics

Introduction

- Project background and context
    - SpaceX is launching its rockets at a lower cost than its competition. This is due to SpaceX reusing its first stage rockets.
- Problems you want to find answers
    - Predicting if the first stage of the SpaceX Falcon 9 Rocket will land successfully

Section 1

Methodology

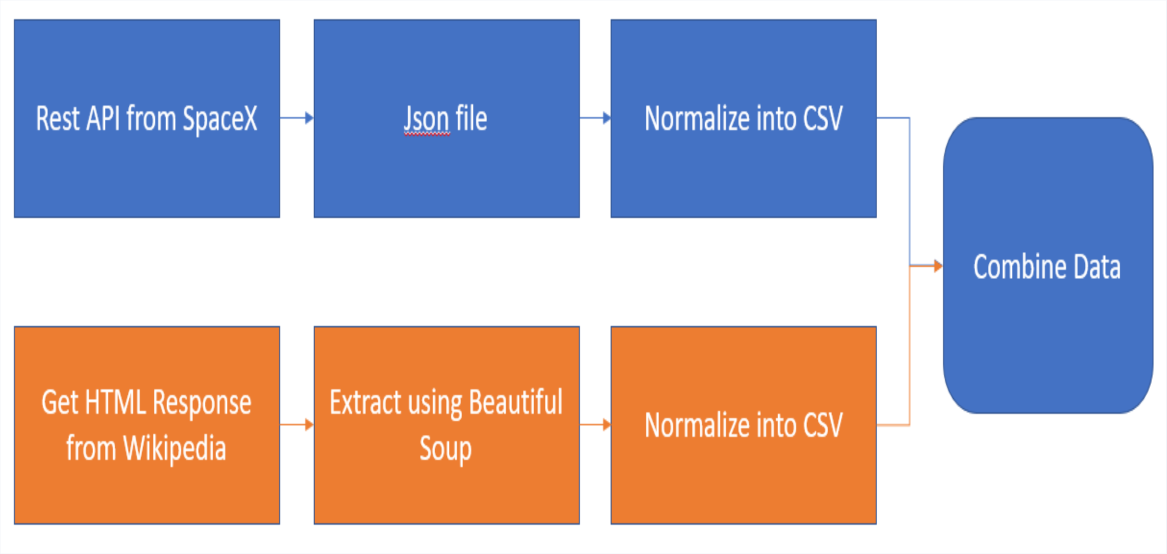
Methodology

Executive Summary

- Data collection methodology:
  - SpaceX Rest API
  - Web Scrapping from Wikipedia
- Perform data wrangling
  - Cleansing nulls and irrelevant columns
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - LR, KNN, SVM, DT models have been built and evaluated for the best classifier

Data Collection

- Launch data is retrieved from SpaceX API
- Other supplementary launch data is retrieved from Wikipedia



Data Collection – SpaceX API

- Launch data is retrieved from SpaceX API
- Other supplementary launch data is retrieved from Wikipedia

- Add the GitHub URL of the completed SpaceX API calls notebook (must include completed code cell and outcome cell), as an external reference and for peer-review purposes

https://github.com/xxShellxShockxx/IBM-Capstone/blob/main/jupyter-labs-webscraping.ipynb

1. **Getting a Response from API**

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

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

2. **Converting Response to a .json file**

```
response = requests.get(spacex_url)
data = pd.json_normalize(response)
```

3. **Apply custom functions to clean data**

```
BoosterVersion[0:5]
```

```
# Call getLaunchSite
getLaunchSite(data)
```

```
# Call getPayloadData
getPayloadData(data)
```

```
# Call getCoreData
getCoreData(data)
```

4. **Assign list to dictionary then dataframe**

```
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}
```

5. **Filter dataframe and export to flat file (.csv)**

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

Data Collection - Scraping

- Present your web scraping process using key phrases and flowcharts

- Add the GitHub URL of the completed web scraping notebook, as an external reference and for peer-review purpose

https://github.com/xxShellxShockxx/IBM-Capstone/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

1. **HTML Response**

```
r = requests.get(static_url)
data = r.text
```

2. **Creating BeautifulSoup Object**

```
soup = BeautifulSoup(data, "html.parser")
```

3. **Finding tables**

```
html_tables = soup.find_all('table')
```

4. **Getting column names**

```
column_names = []
table_headers = first_launch_table.find_all('th')
# print(table_headers)
for j, table_header in enumerate(table_headers):
    name = extract_column_from_header(table_header)
    if name is not None and len(name) > 0:
        column_names.append(name)

print(column_names)
```

5. **Appending data to keys**

```
launch_dict= dict.fromkeys(column_names)

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

# Let's initial the launch_dict with each value to be an empty list
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']=[]
```

6. **Converting dictionary to dataframe**

```
extracted_row = 0
for table_number,table in enumerate(soup.find_all('table',"wikitable plainrowheaders collapsible")):
    for rows in table.find_all("tr"):  # get table row
        if rows.th:  #check to see if first table heading is a number corresponding to launch a number
            if rows.th.string:
                flight_number=rows.th.string.strip()
                flag=flight_number.isdigit()
        else:
```

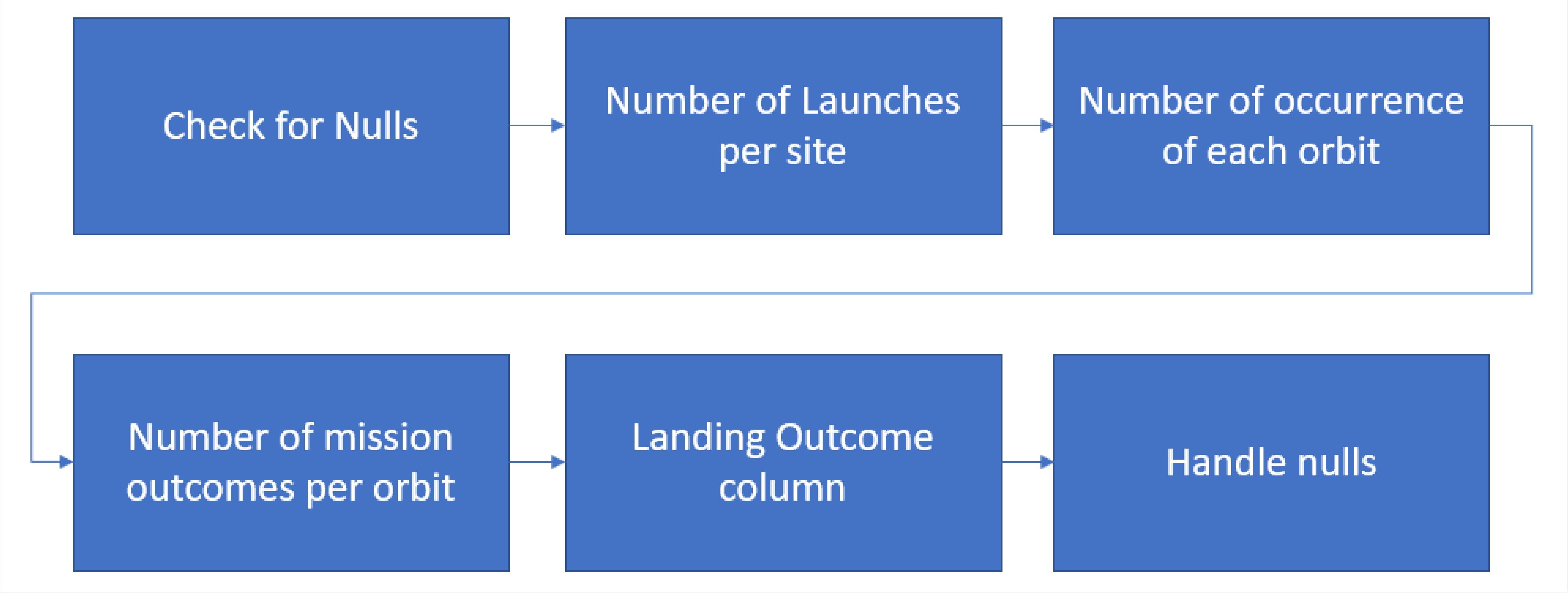
7. **Converting dictionary to dataframe**

```
df=pd.DataFrame(launch_dict)
```

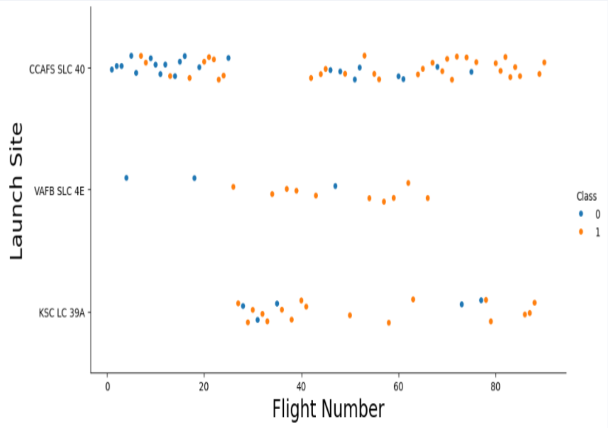
8. **Dataframe to .CSV**

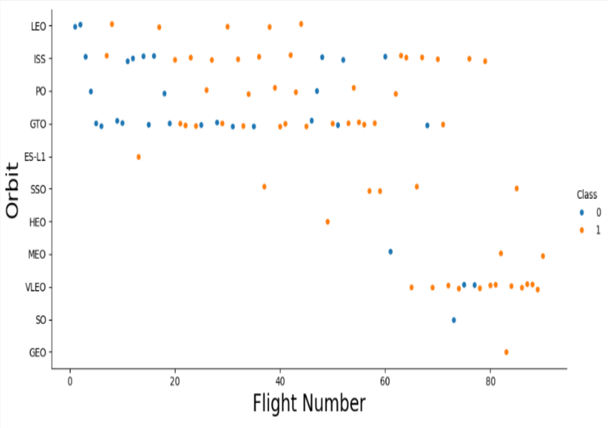
```
df.to_csv('spacex_web_scraped.csv', index=False)
```

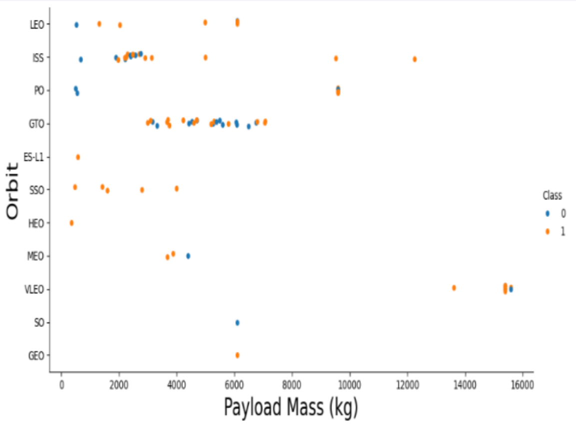
Data Wrangling

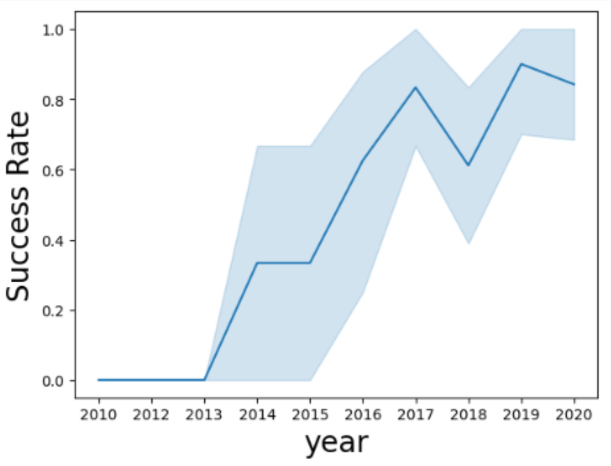


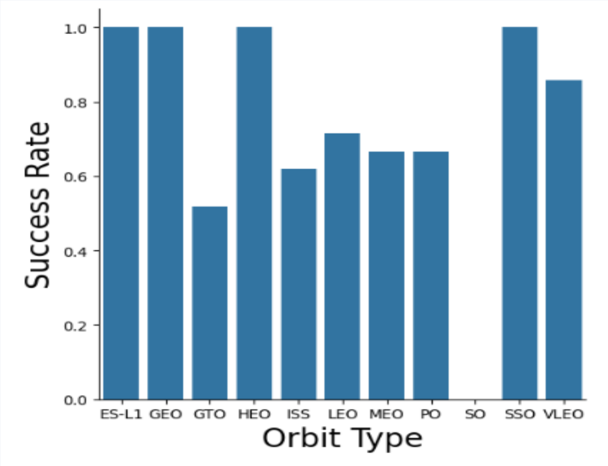
EDA with Data Visualization

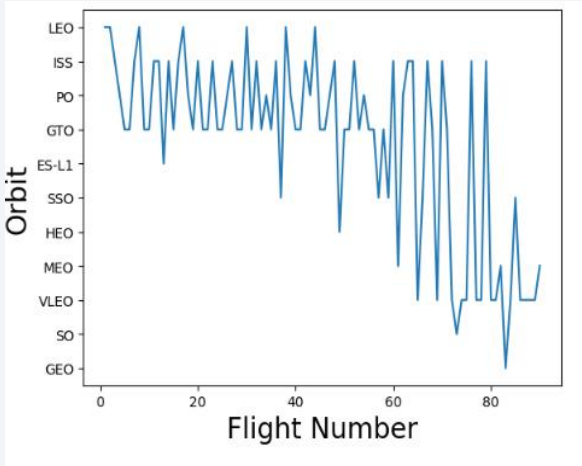










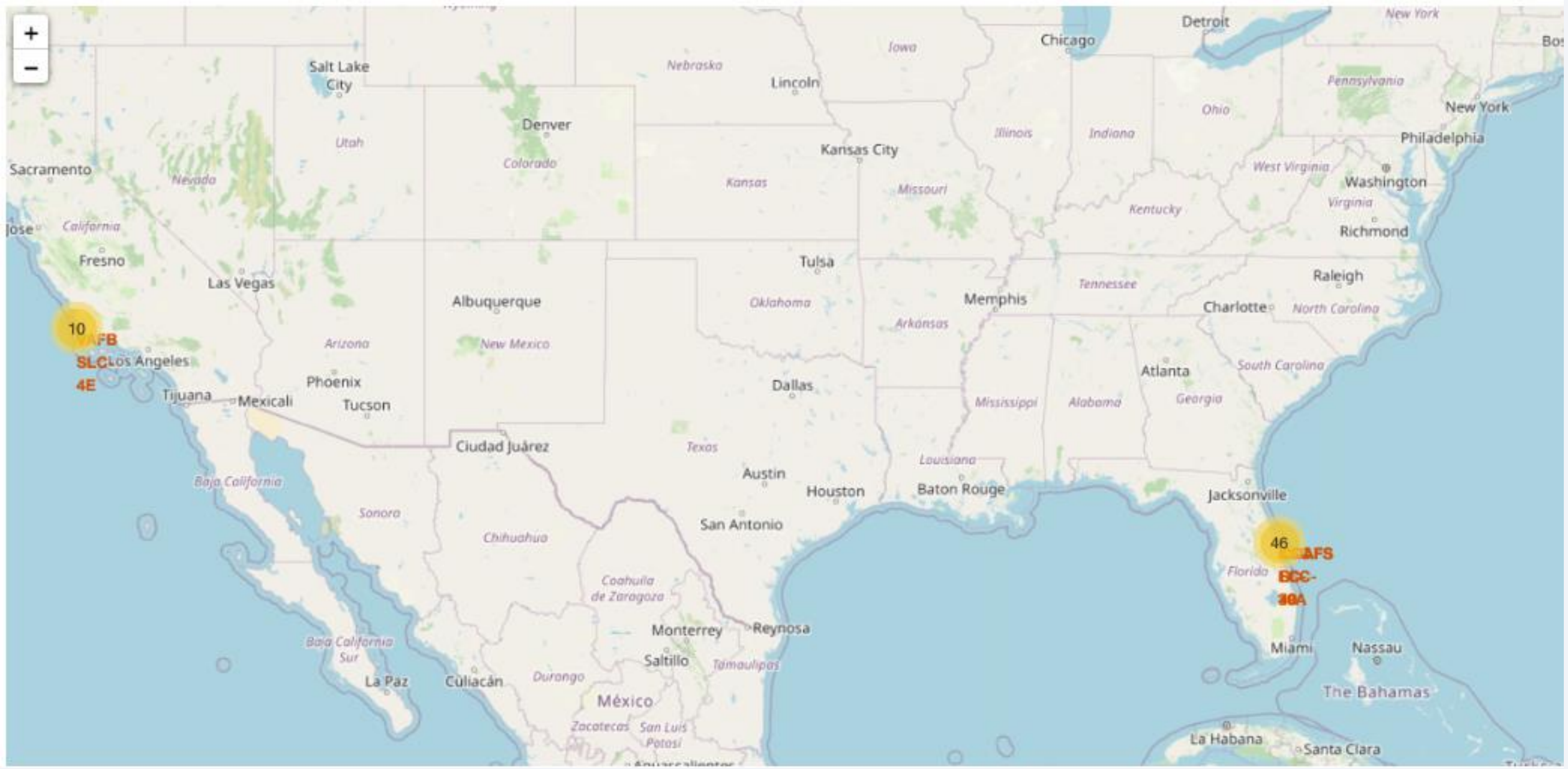


https://github.com/xxShellxShockxx/IBM-Capstone/blob/main/jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb

EDA with SQL

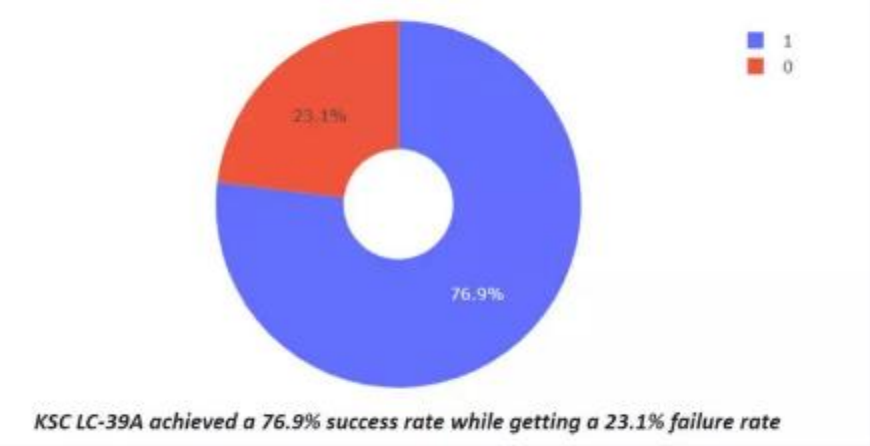
- SQL queries performed include:
    - Display 5 records where the launch site begins with ‘KSC’.
    - Display avg payload carried by F9 v1.1
    - List of dates for successful landing outcome on the drone ship was achieved
    - List boosters with a payload greater than 4000 but less than 6000 that was successful at ground pad.
    - List of successful and failure missions.

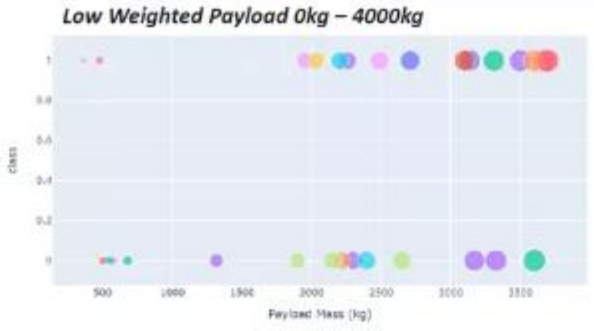
Build an Interactive Map with Folium

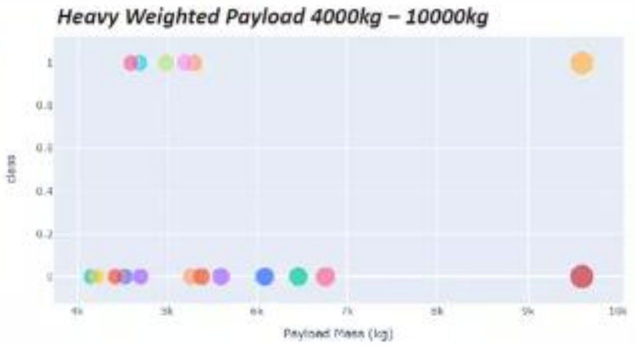


Map Markers identify the optimal location for launch sites

Build a Dashboard with Plotly Dash

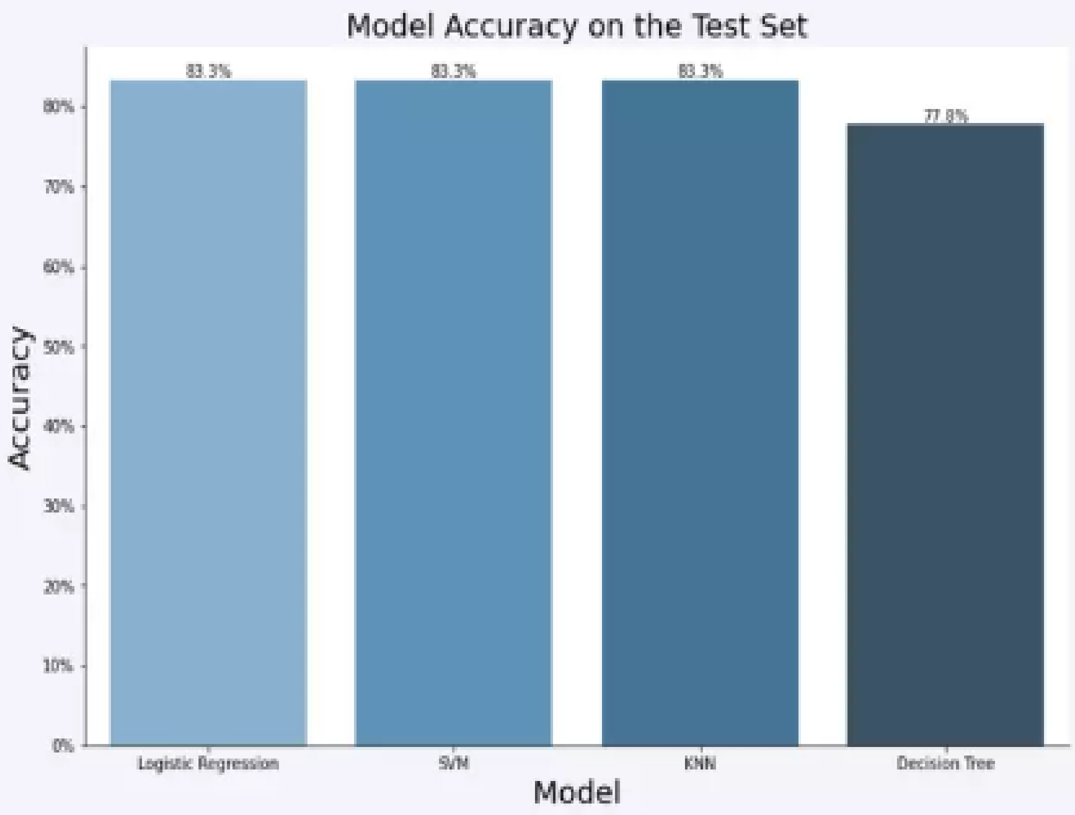


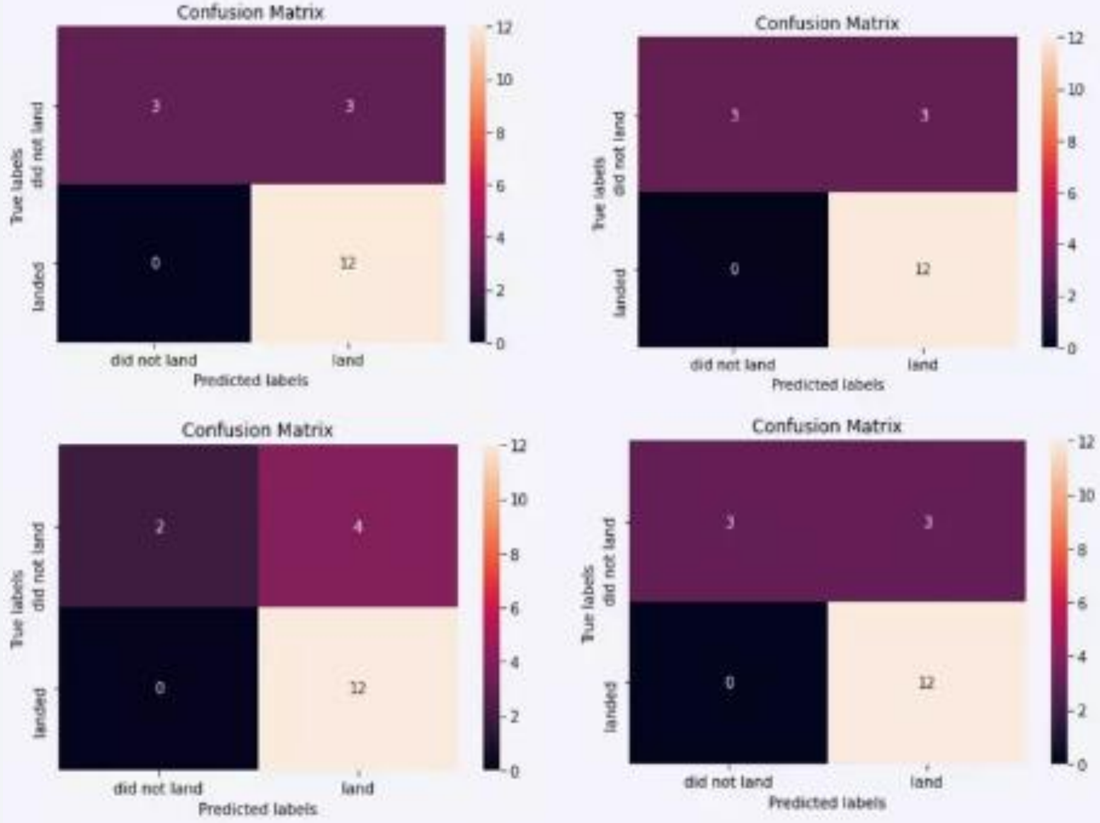




We can see the success rates for low weighted payloads is higher than the heavy weighted payloads

Predictive Analysis (Classification)





LR, SVM, and KNN had the highest accuracy. The SVM model performed at 0.958 in the area under the curve.

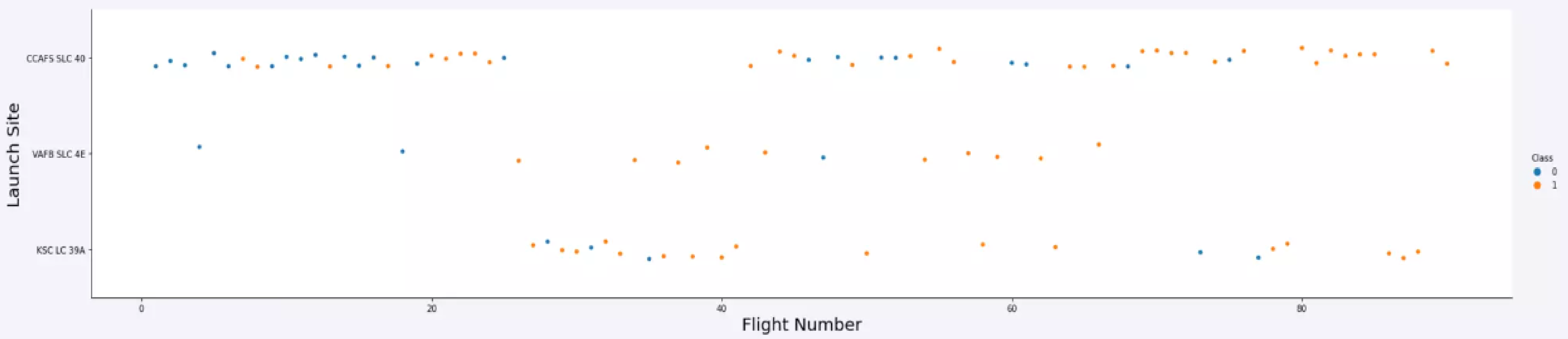
Results

- LR, SVM, and KNN have the best accuracy.
- Low-weight payloads perform the best.
- KSC LC 39A had the most successful launches.
- Orbit GEO, HEO, SSO, ES L1 have the best success rate.

Section 2

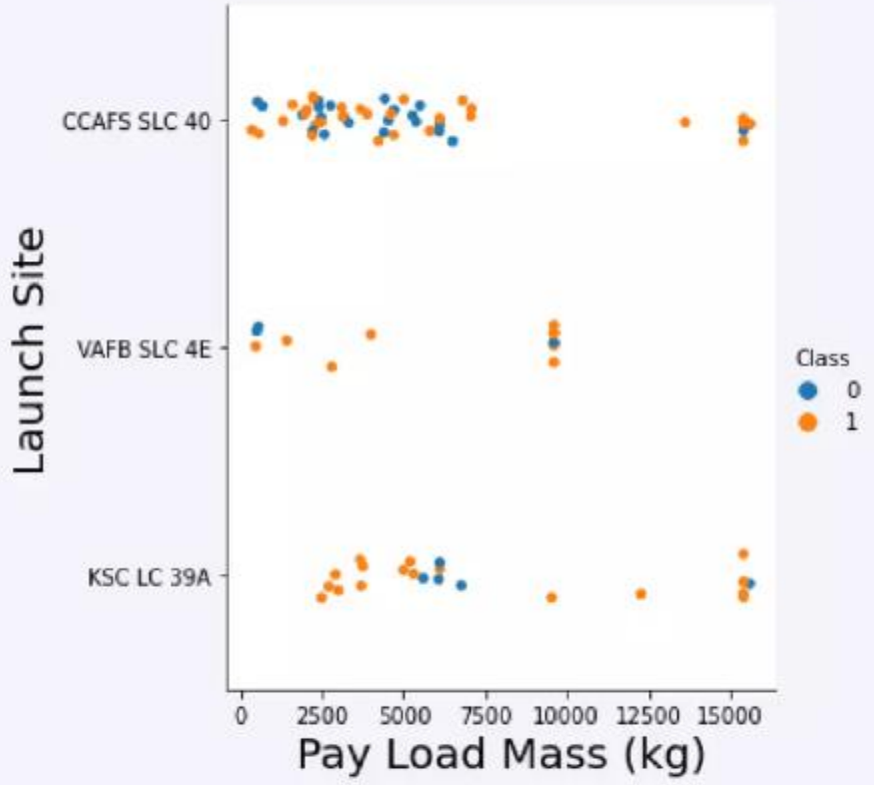
Insights drawn from EDA

Flight Number vs. Launch Site



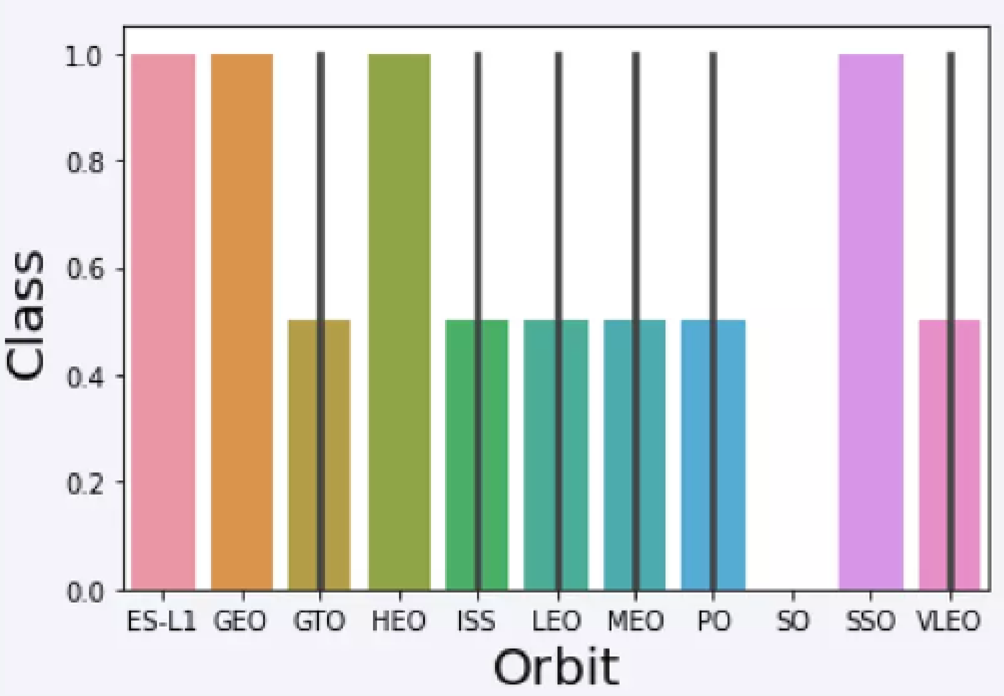
CCAFS SLC 40 has significantly more launches

Payload vs. Launch Site



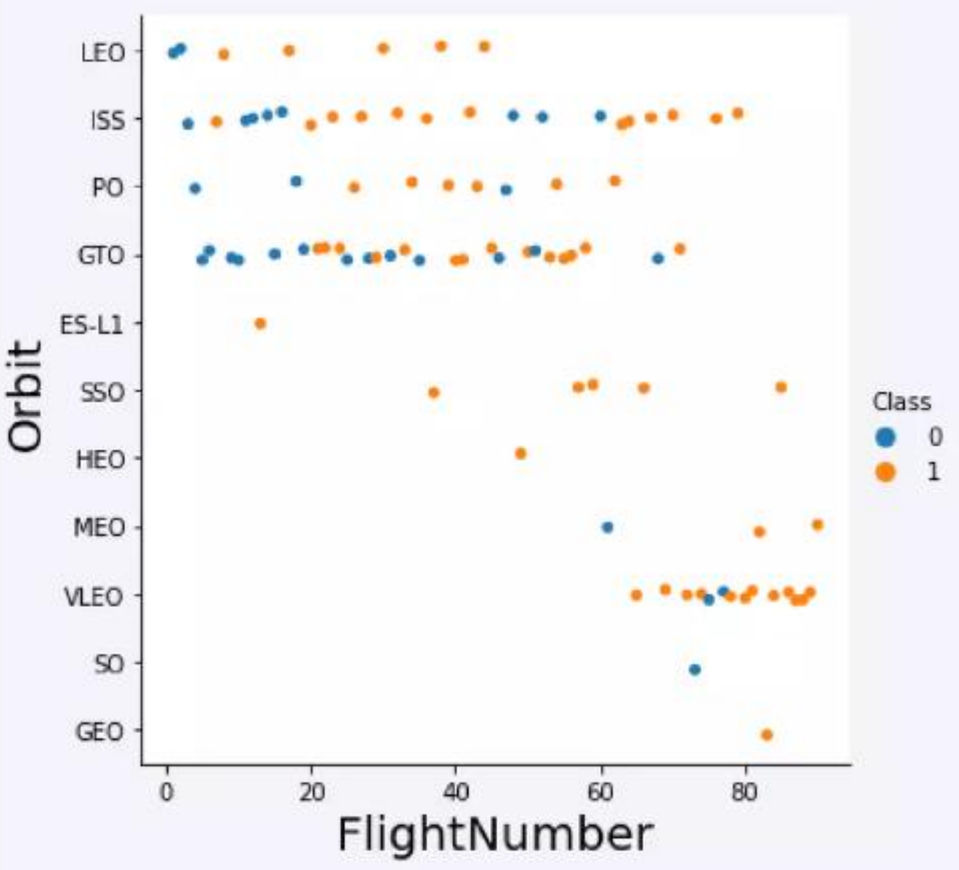
IPay loads with lower mass have been launched from CCAFS SLC 40.

Success Rate vs. Orbit Type



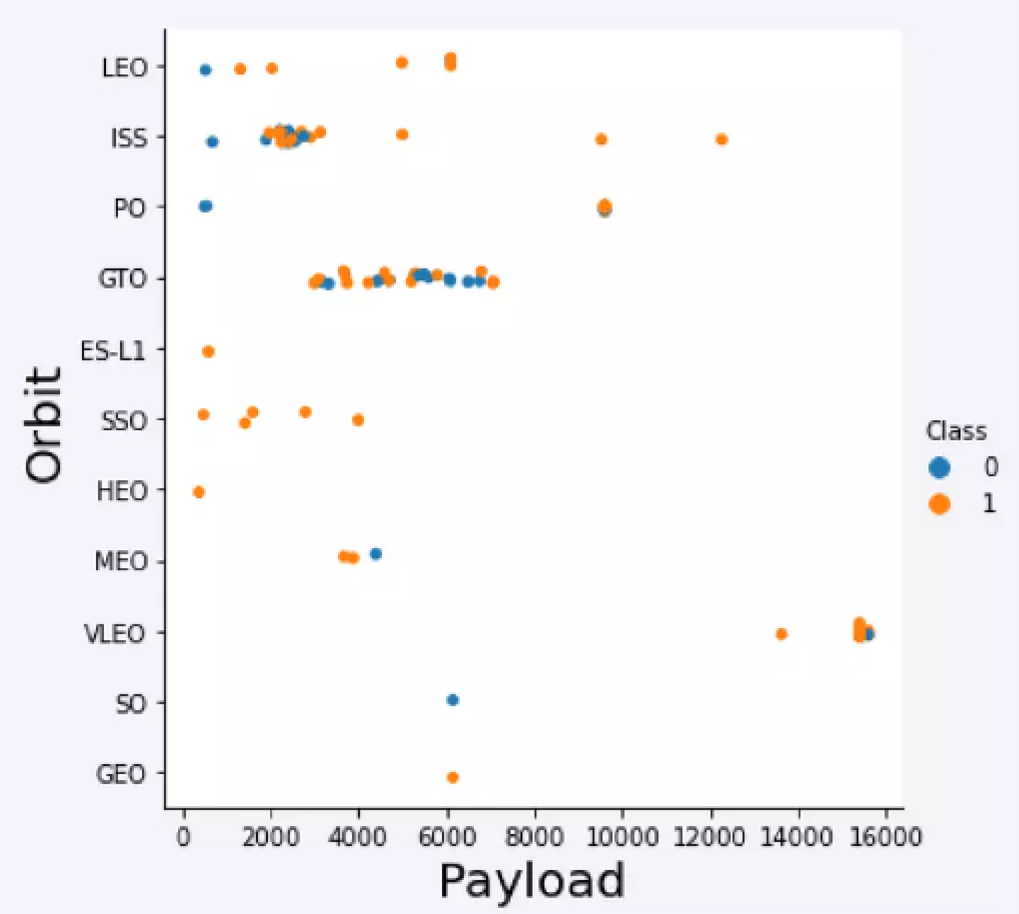
ES-L1, GEO, HEO, and SSO have the highest success rate.

Flight Number vs. Orbit Type



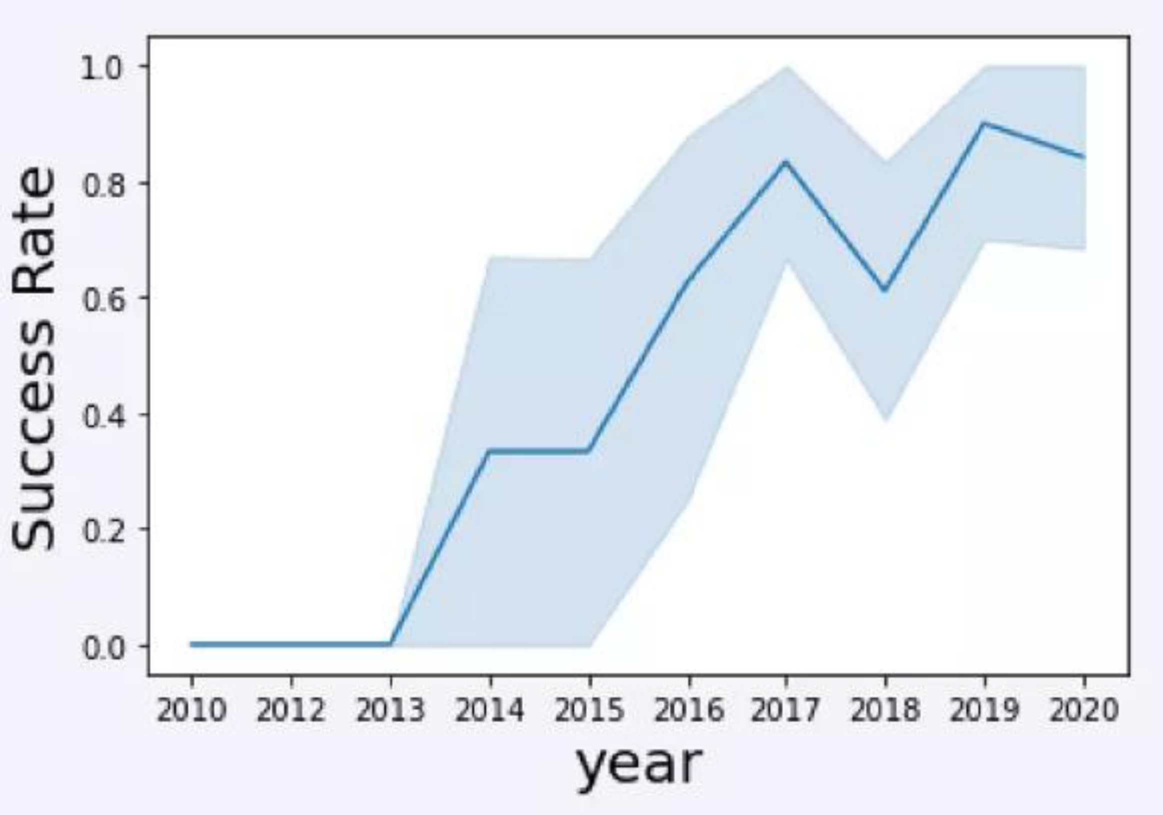
VLEO launches have been shifted to in recent years

Payload vs. Orbit Type



Strong correlation between ISS and payload at 2000 as well as Gto and the range 4000-8000.

Launch Success Yearly Trend



The launch success rate has stabilized as of 2019 due to advances in technology.

All Launch Site Names

launch_site
CCAFS LC-40
CCAFS SLC-40
KSC LC-39A
VAFB SLC-4E

```
%sql select distinct(LAUNCH_SITE) from SPACEXTBL
```

Launch Site Names Begin with 'CCA'

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

%sql select * from SPACEXTBL where LAUNCH_SITE like ‘CCA%’ limit 5

Total Payload Mass

%sql select sum(PAYLOAD_MASS_KG_) from SPACEXTBL where CUSTOMER = ‘NASA (CRS)’

45596

Average Payload Mass by F9 v1.1

%sql select avg(PAYOAD_MASS_KG_) from SPACEXTBL where BOOSTER_VERSION = ‘F9 v1.1’

2928.400000

First Successful Ground Landing Date

%sql select min(DATE) from SPACEXTBL where Landing_Outcome = 'Success (ground pad)'

2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

%sql select BOOSTER_VERSION from SPACEXTBL where Landing_Outcome = 'Success(drone ship)' and PAYLOAD_MASS_KG_> 4000 and PAYLOAD_MASS_KG_ < 6000

booster_version
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
F9 FT B1031.2

Total Number of Successful and Failure Mission Outcomes

```
%sql select count(MISSION_OUTCOME) from SPACEXTBL where
MISSION_OUTCOME = 'Success' or MISSION_OUTCOME = 'Failure (in flight)'
```

100

Boosters Carried Maximum Payload

%sql select BOOSTER_VERSION from SPACEXTBL where PAYLOAD_MASS_KG_ = (select max(PAYLOAD_MASS_KG_) from SPACEXTBL)

booster_version
F9 B5 B1048.4
F9 B5 B1049.4
F9 B5 B1051.3
F9 B5 B1056.4
F9 B5 B1048.5
F9 B5 B1051.4
F9 B5 B1049.5
F9 B5 B1060.2
F9 B5 B1058.3
F9 B5 B1051.6
F9 B5 B1060.3
F9 B5 B1049.7

2015 Launch Records

%sql select * from SPACEXTBL where Landing_Outcome like 'Success%' and (DATE between '2015-01-01' and '2015-12-31') order by date desc

time_utc_	booster_version	launch_site	payload	payload_mass_kg_	orbit	customer	mission_outcome	landing_outcome
14:39:00	F9 FT B1031.1	KSC LC-39A	SpaceX CRS-10	2490	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
17:54:00	F9 FT B1029.1	VAFB SLC-4E	Iridium NEXT 1	9600	Polar LEO	Iridium Communications	Success	Success (drone ship)
05:26:00	F9 FT B1026	CCAFS LC-40	JCSAT-16	4600	GTO	SKY Perfect JSAT Group	Success	Success (drone ship)
04:45:00	F9 FT B1025.1	CCAFS LC-40	SpaceX CRS-9	2257	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
21:39:00	F9 FT B1023.1	CCAFS LC-40	Thaicom 8	3100	GTO	Thaicom	Success	Success (drone ship)
[illegible]	[illegible]	CCAFS LC-	[illegible]	[illegible]	[illegible]	SKY Perfect JSAT	[illegible]	[illegible]

Rank Landing Outcomes Between 2010-06-04 and 2017-03-20

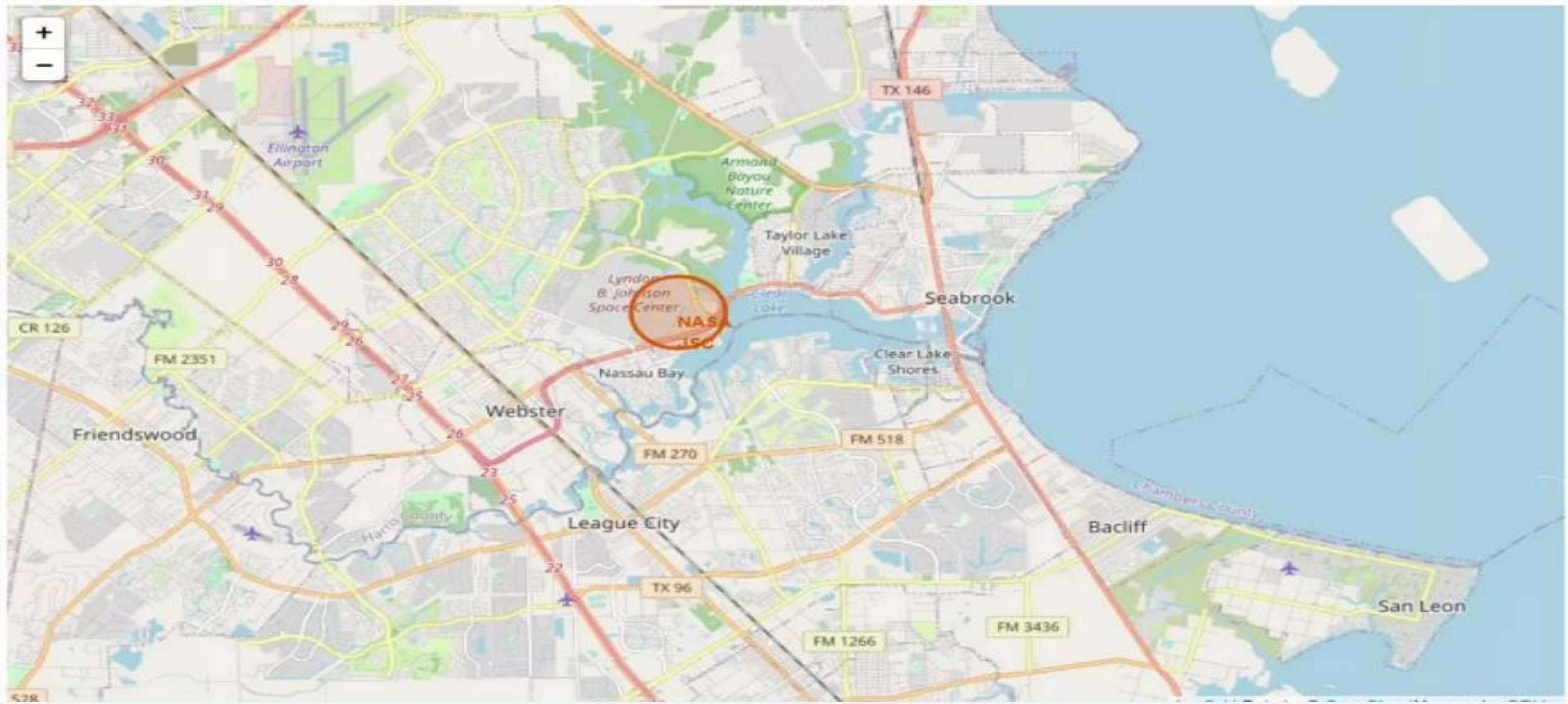
%sql select * from SPCEXTBL where Landing_Outcome like ‘Success%’ and (DATE between ‘2010-06-04’ and ‘2017-03-20’) order by date desc

2016-05-27	21:39:00	F9 FT B1023.1	CCAFS LC-40	Thaicom 8	3100	GTO	Thaicom	Success	Success (drone ship)
2016-05-06	05:21:00	F9 FT B1022	CCAFS LC-40	JCSAT-14	4696	GTO	SKY Perfect JSAT Group	Success	Success (drone ship)
2016-04-08	20:43:00	F9 FT B1021.1	CCAFS LC-40	SpaceX CRS-8	3136	LEO (ISS)	NASA (CRS)	Success	Success (drone ship)
2015-12-22	01:29:00	F9 FT B1019	CCAFS LC-40	OG2 Mission 2 11 Orbcomm-OG2 satellites	2034	LEO	Orbcomm	Success	Success (ground pad)

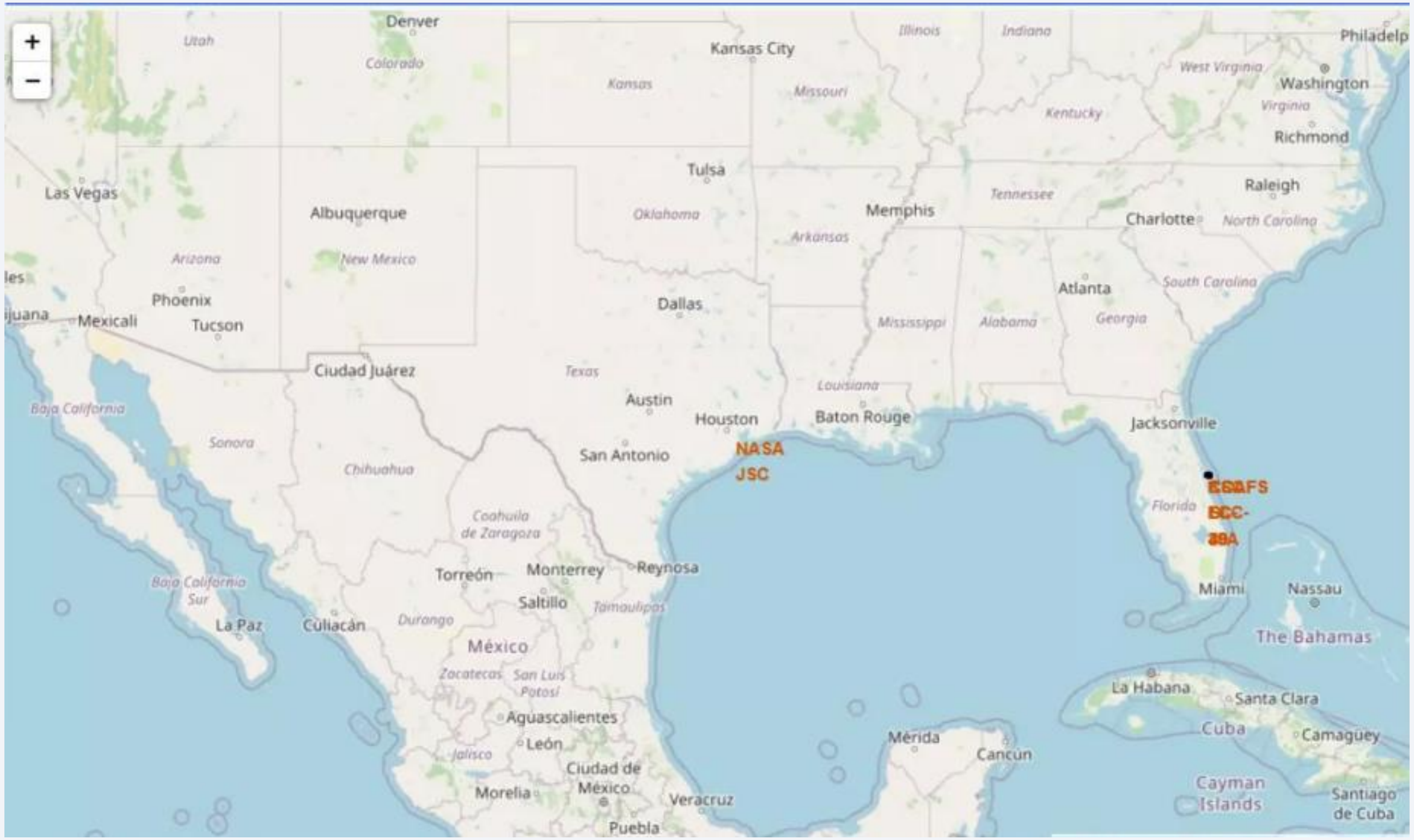
Section 3

Launch Sites Proximities Analysis

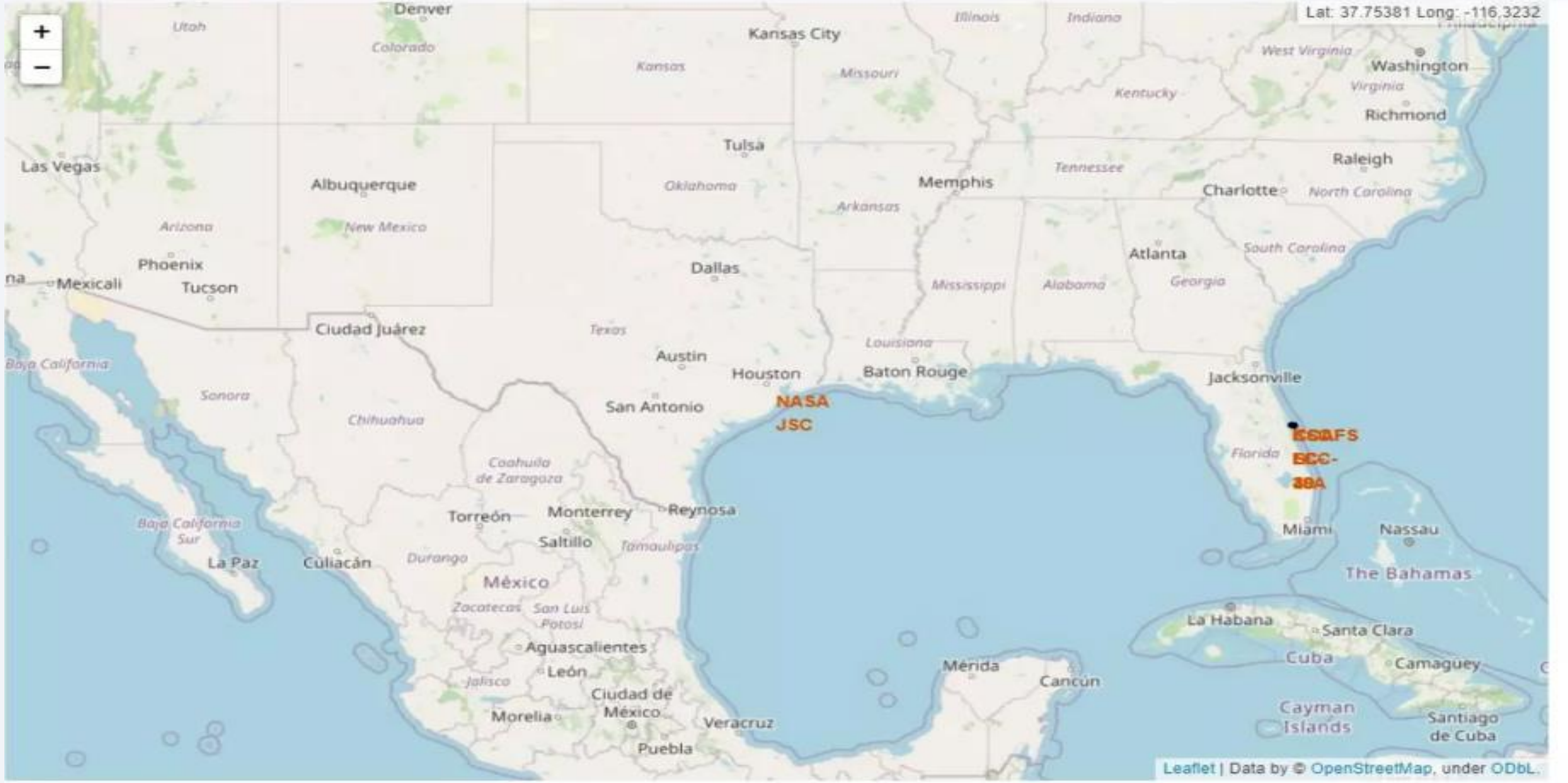
Launch sites marked on map



Launches marked on the map



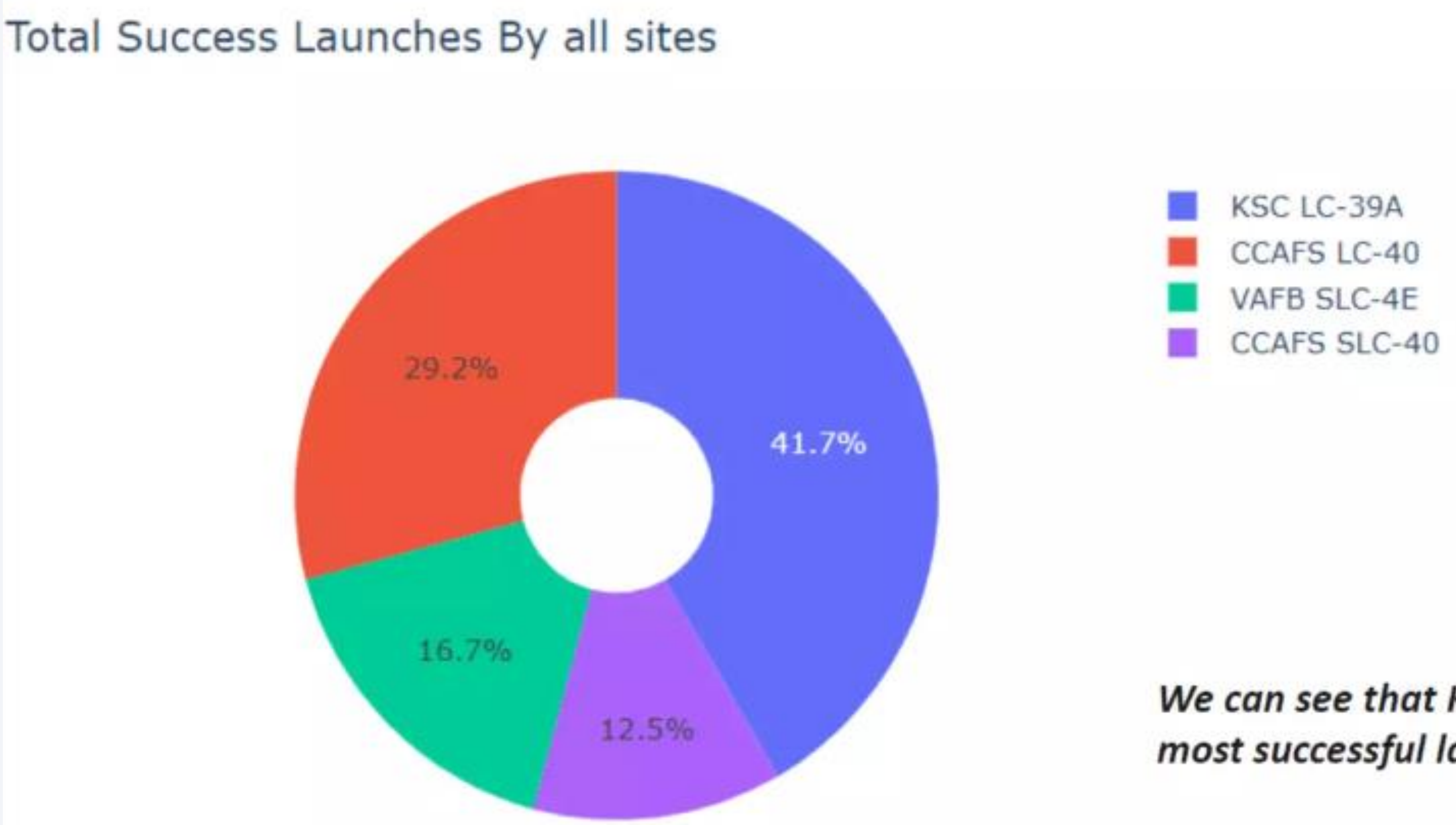
Distance between launch site to its proximities



Section 4

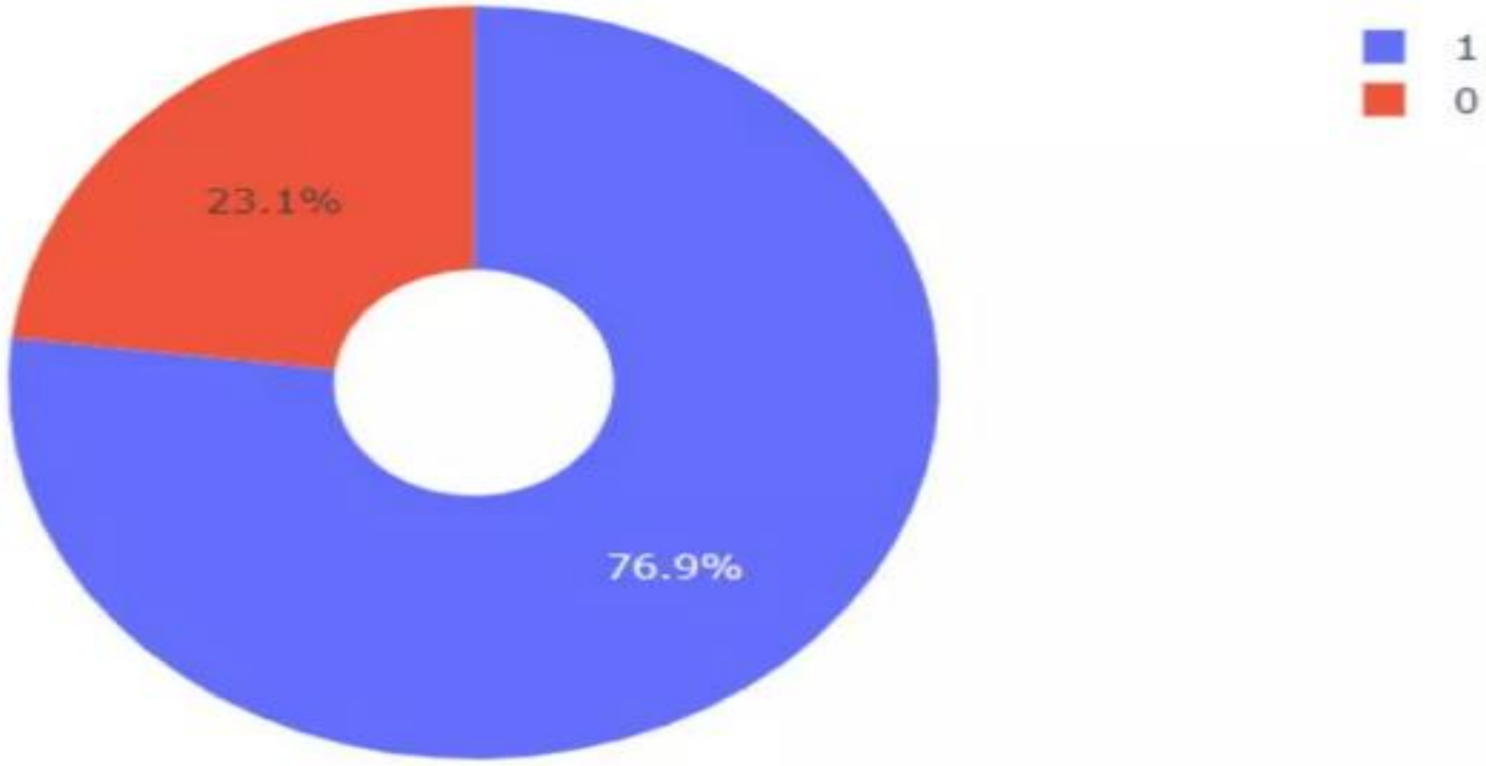
Build a Dashboard with Plotly Dash

Successful launches by all sites



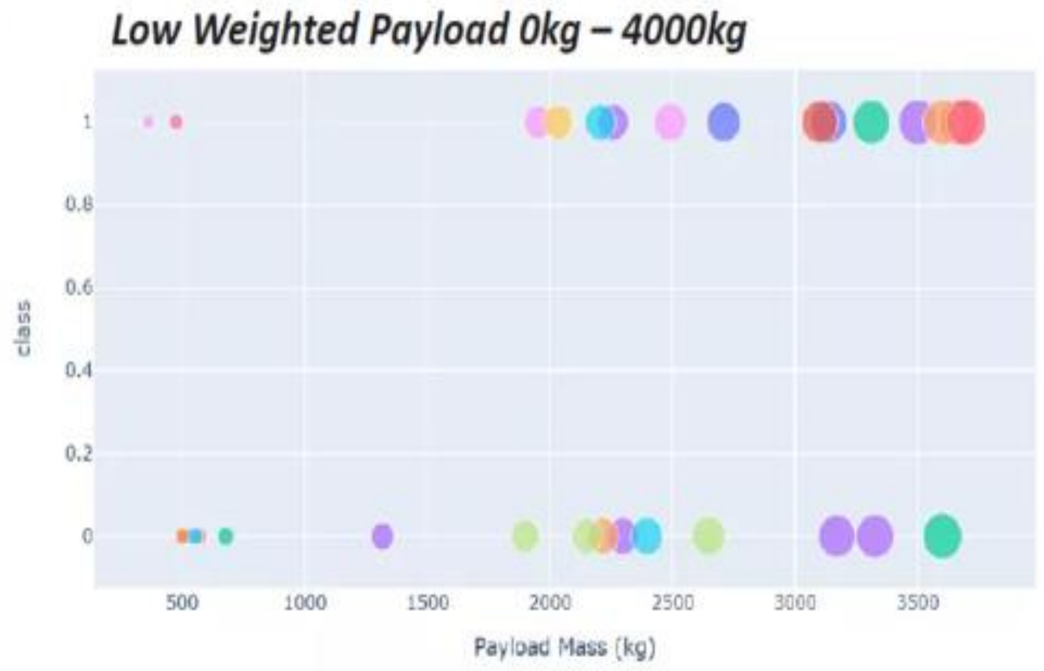
We can see that KSC LC-39A had the most successful launches from all the sites

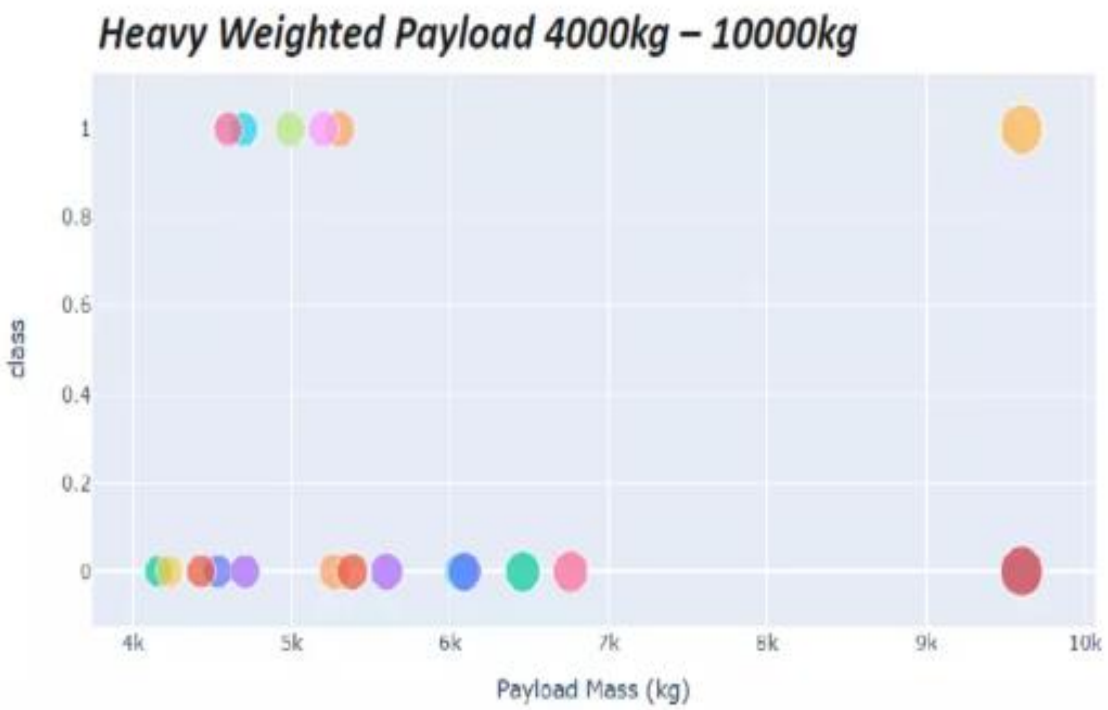
Success rate by site



KSC LC-39A achieved a 76.9% success rate while getting a 23.1% failure rate

Payload vs launch outcome



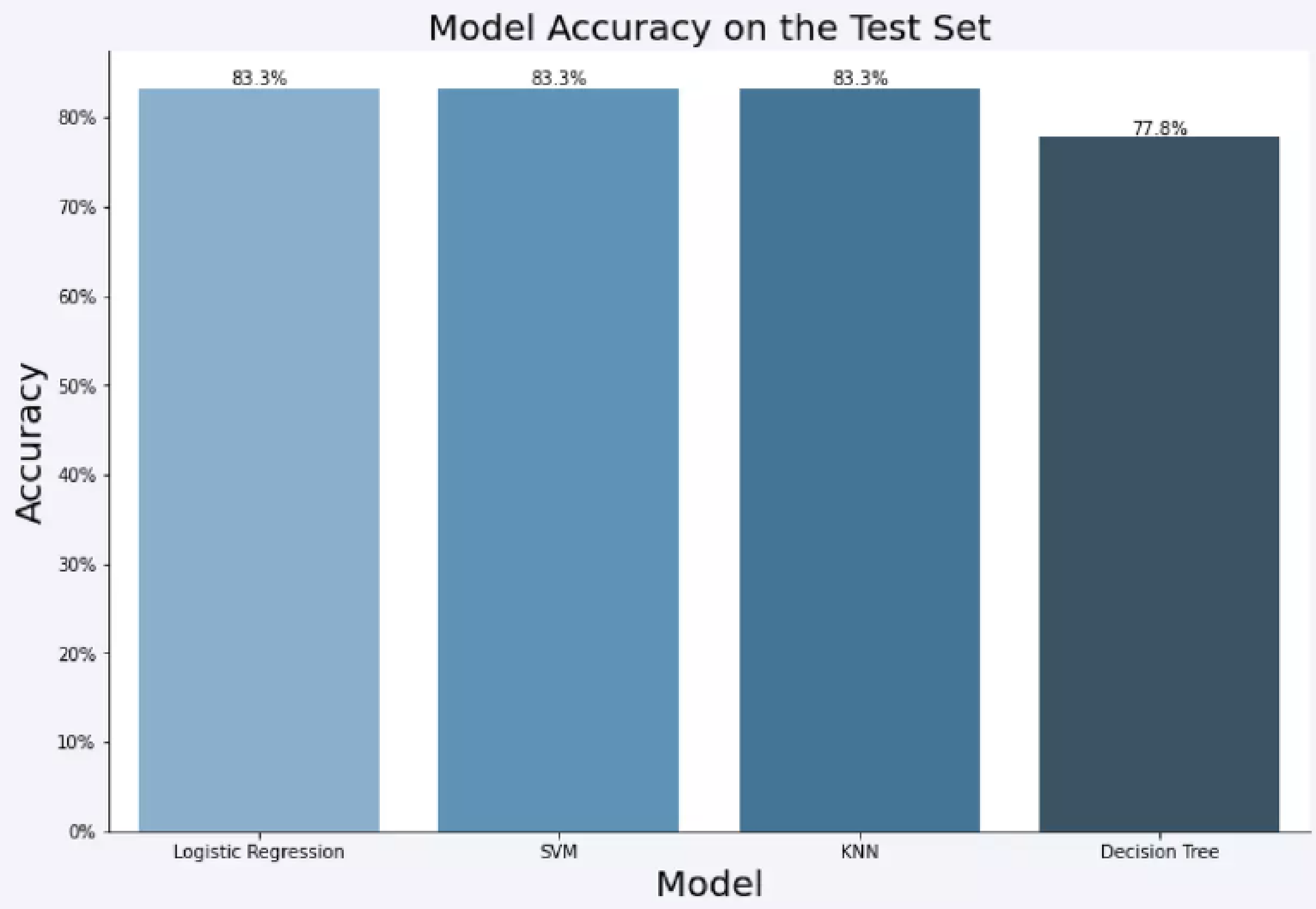


We can see the success rates for low weighted payloads is higher than the heavy weighted payloads

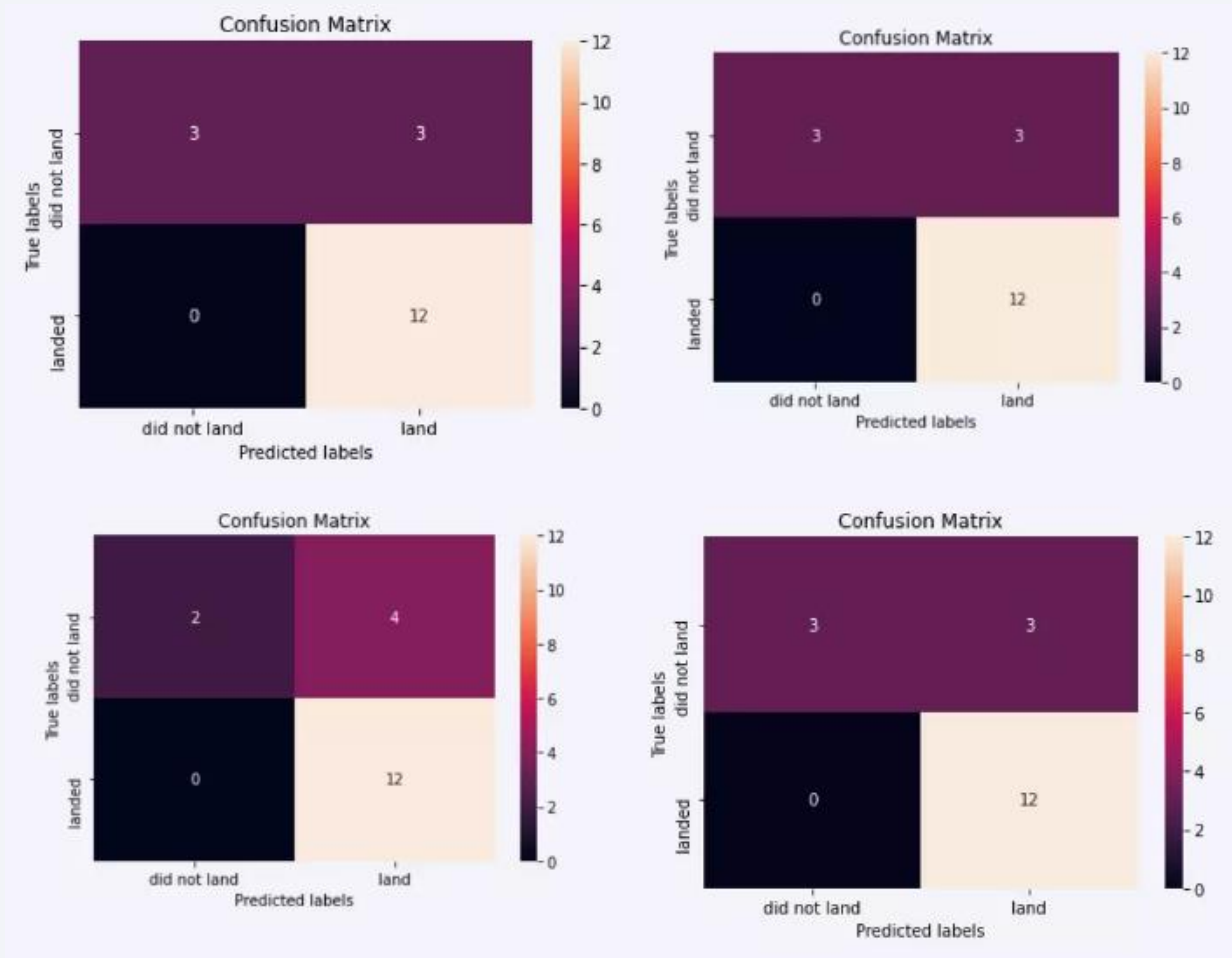
Section 5

Predictive Analysis (Classification)

Classification Accuracy



Confusion Matrix



Conclusions

- The LG, KNN, and SVM models have the best accuracy.
- Low-weighted payloads perform the best
- KSC LC 39A had the most success
- Orbit GEO, HEO, SSO, ES L1 has the best Success Rate

Thank you!