IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Antonio Martínez Vicente
August 29, 2023

Outline

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

Executive Summary

- Summary of methodologies
  - Data collection
  - Data wrangling
  - EDA with data visualization
  - EDA using SQL sentences
  - Building an interactive map with Folium
  - Building a Dashboard with Plotly Dash
  - Predictive analysis (Classification)
- Summary of all results
  - EDA results
  - Interactive analytics
  - Predictive analysis

Introduction

- **Project Background and Context:**

This project addresses the growing presence of commercial companies in the space industry, which has made space travel more accessible. Various companies are engaged in suborbital flights, satellite launches, and reusable rockets. Specifically, SpaceX has achieved remarkable feats, such as sending spacecraft to the International Space Station, creating the Starlink satellite constellation, and conducting crewed space missions. Their advantage lies in reusing the first stage of their Falcon 9 rockets, significantly reducing costs.

- **Problems We Aim to Address:**

Our project focuses on the challenge of determining the cost of a space launch by predicting whether the first stage of a SpaceX Falcon 9 rocket will be reused or not. Our company, Space Y, founded by Allon Musk to compete with SpaceX, needs to establish competitive pricing for its launches. Our task involves collecting data from various sources and extracting information by designing dashboards for the Space Y team. Instead of relying on traditional evaluation methods, we propose utilizing machine learning to predict the reuse of the first stage, leveraging publicly available information.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
    - SpaceX REST API.
    - Webscraping Falcon 9 launch records from Wikipedia
    - Perform data wrangling
    - Cleaning data of null values and non relevant columns and convert categorical data fields for Machine Learning processes
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
    - We have built KNN, LOGREG, SVM, Decision Trees models and selected best classifier

Data Collection

SpaceX REST API

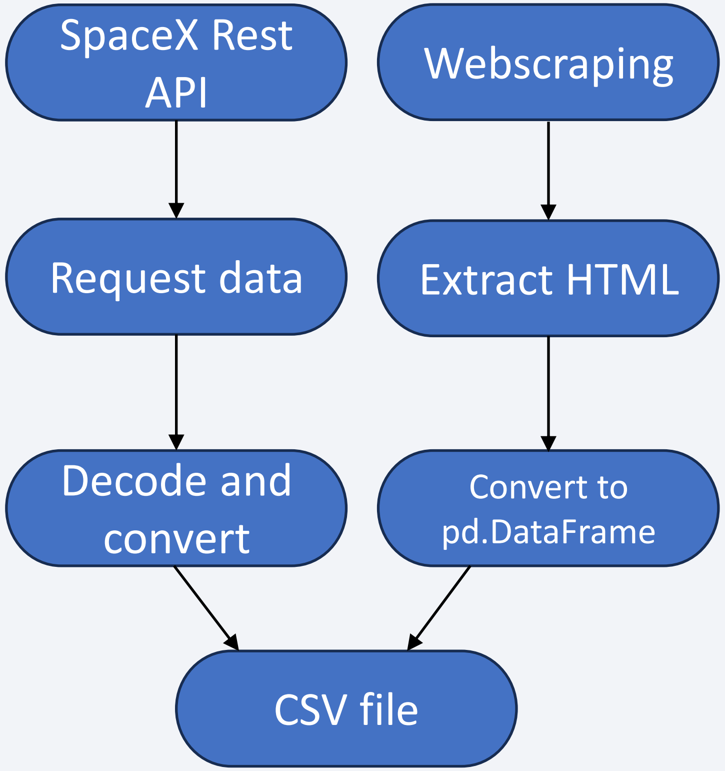
----

1 Import Libraries and Define Auxiliary Functions:

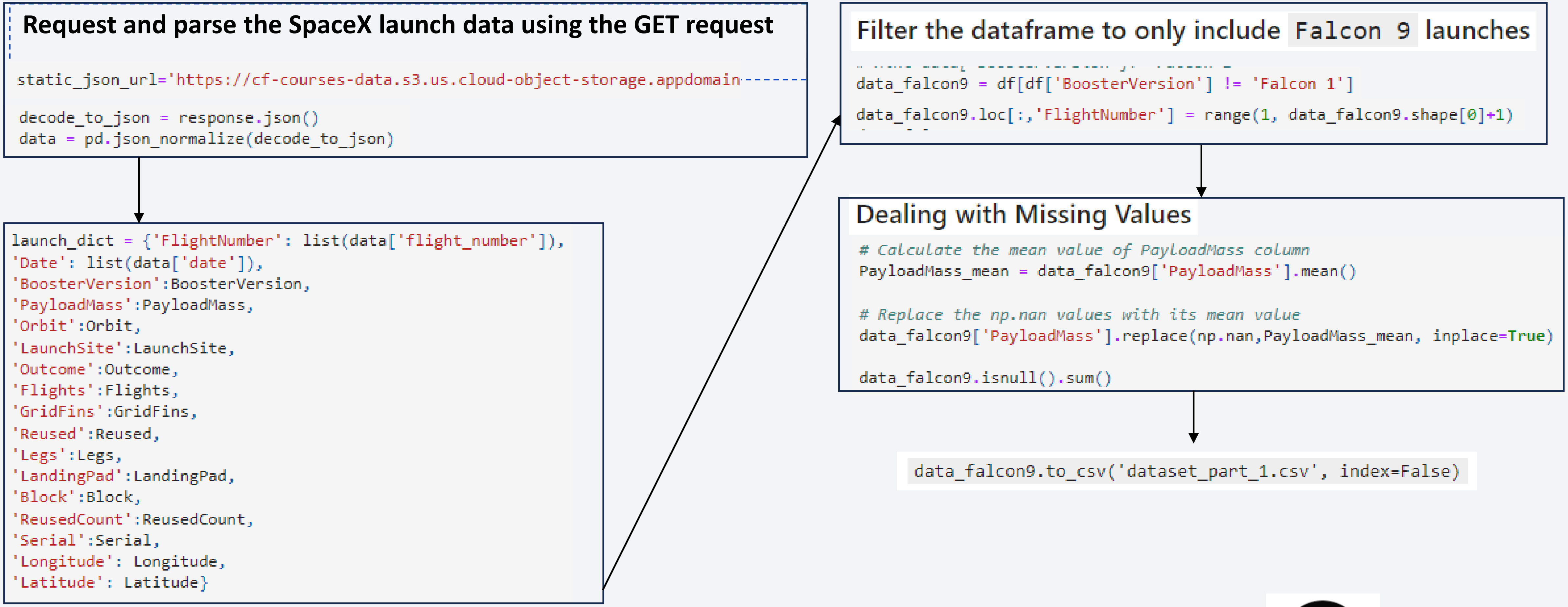
- Libraries imported: Requests, Pandas, NumPy, Datetime.
- Helper functions defined to extract specific information from APIs:
  - **getBoosterVersion**: Retrieves booster names **getLaunchSite**: Extracts launch site names, longitude, and latitude from launchpad IDs.
  - **getPayloadData**: Collects payload mass and orbit details from payload IDs.
  - **getCoreData**: Gathers landing outcomes, core details, and reusability info from core IDs.
- 2 Request SpaceX Launch Data:
  - Request data from SpaceX API using URL: https://api.spacexdata.com/v4/launches/past
  - Also, a static response object is used from static_json_url for consistency.
- 3 Decode and Convert to DataFrame:
  - Decode JSON response using .json() and convert to a Pandas DataFrame using .json_normalize().
  - Displaying Data:
- 4 Normalise data into csv file

Webscraping

- 1 Webscraping Falcon 9 launch records from Wikipedia.
  - Imported required packages: BeautifulSoup, requests, pandas.
  - Defined functions for data extraction, e.g., date_time, booster_version, landing_status, etc.
- 2 Extract an HTML table containing launch records.
  - Found all tables on the Wikipedia page
  - Parsed table rows, extracted information, and filled launch_dict with data.
  - Used functions and logic to extract relevant data from rows.
- 3 Convert the table into a Pandas DataFrame.
  - Created DataFrame from the populated launch_dict
- 4 Export data into csv file



Data Collection – SpaceX API



Request and parse the SpaceX launch data using the GET request

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

```
decode_to_json = response.json()
data = pd.json_normalize(decode_to_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}
```

Filter the dataframe to only include `Falcon 9` launches

```
data_falcon9 = df[df['BoosterVersion'] != 'Falcon 1']

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

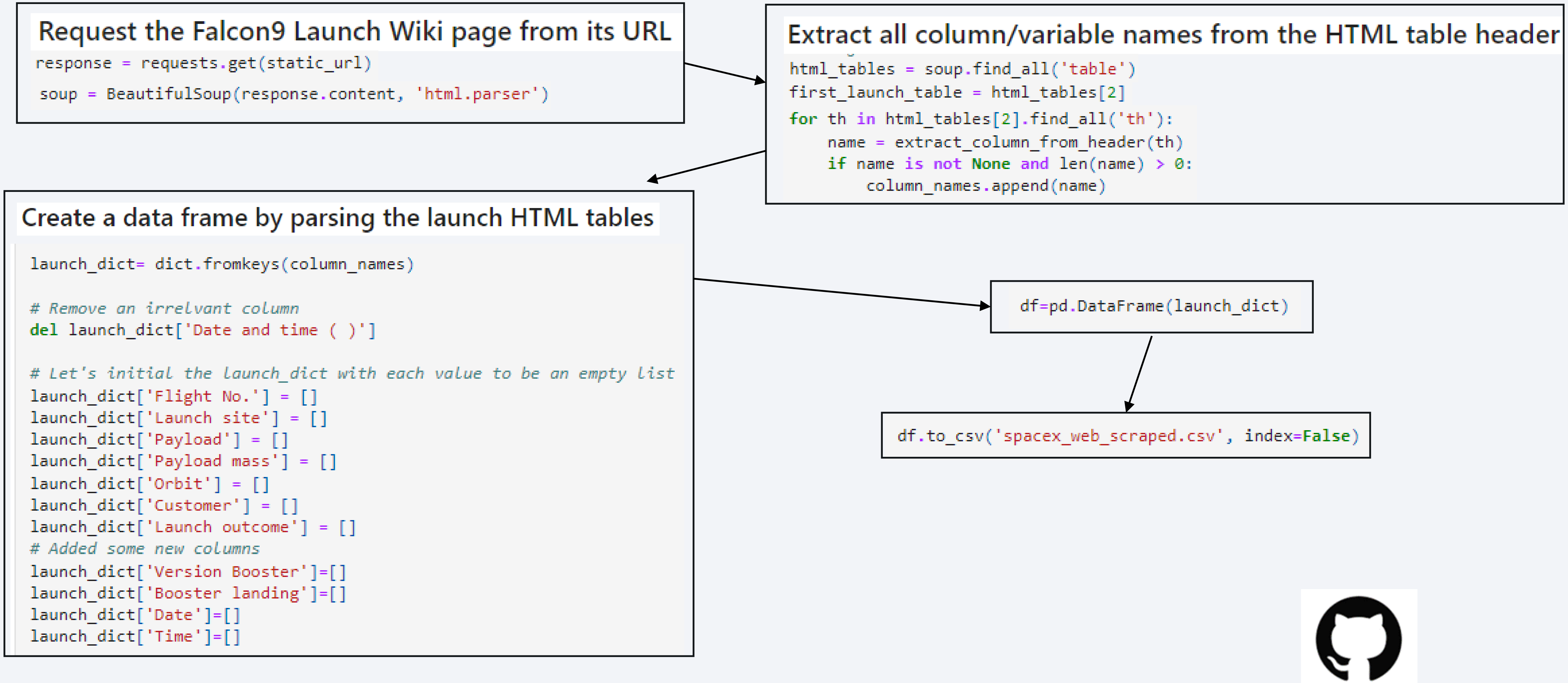
Dealing with Missing Values

```
# Calculate the mean value of PayloadMass column
PayloadMass_mean = data_falcon9['PayloadMass'].mean()

# Replace the np.nan values with its mean value
data_falcon9['PayloadMass'].replace(np.nan,PayloadMass_mean, inplace=True)

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

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

GitHub URL for SpaceX API notebook

Data Collection – Scraping from Wikipedia



Request the Falcon9 Launch Wiki page from its URL

```
response = requests.get(static_url)
soup = BeautifulSoup(response.content, 'html.parser')
```

Extract all column/variable names from the HTML table header

```
html_tables = soup.find_all('table')
first_launch_table = html_tables[2]
for th in html_tables[2].find_all('th'):
    name = extract_column_from_header(th)
    if name is not None and len(name) > 0:
        column_names.append(name)
```

Create a data frame by parsing the launch HTML tables

```
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']=[]
```

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

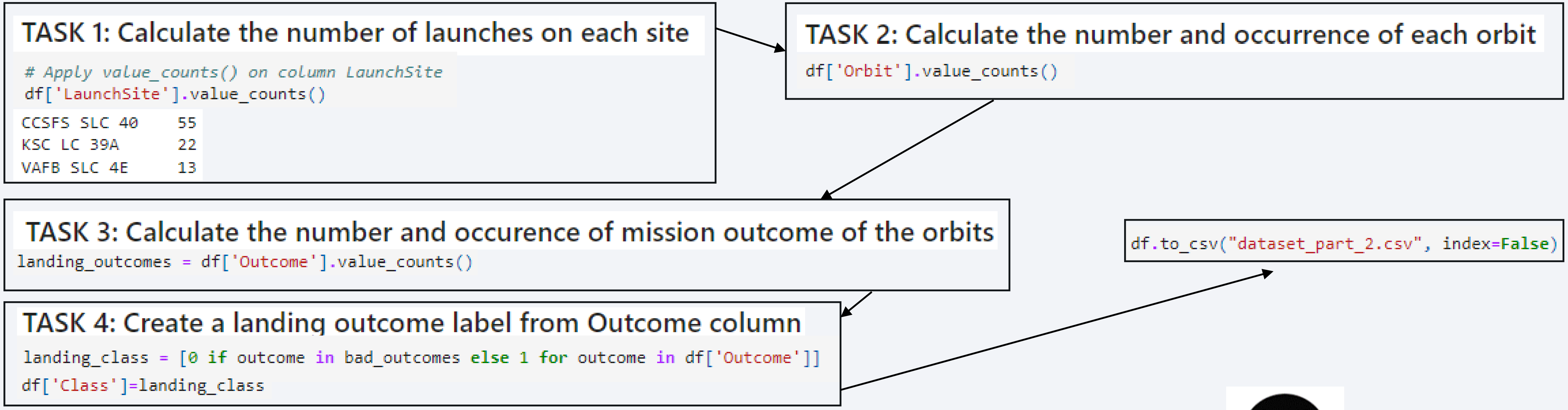
```
df.to_csv('spacex_web_scraped.csv', index=False)
```

URL to web scraping notebook

GitHub

Data Wrangling

We will perform some Exploratory Data Analysis (EDA) to find some patterns in the data and determine what would be the label for training supervised models.



TASK 1: Calculate the number of launches on each site

```
# Apply value_counts() on column LaunchSite
df['LaunchSite'].value_counts()
```

```
CCSFS SLC 40     55
KSC LC 39A       22
VAFB SLC 4E      13
```

TASK 2: Calculate the number and occurrence of each orbit

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

TASK 3: Calculate the number and occurence of mission outcome of the orbits

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

TASK 4: Create a landing outcome label from Outcome column

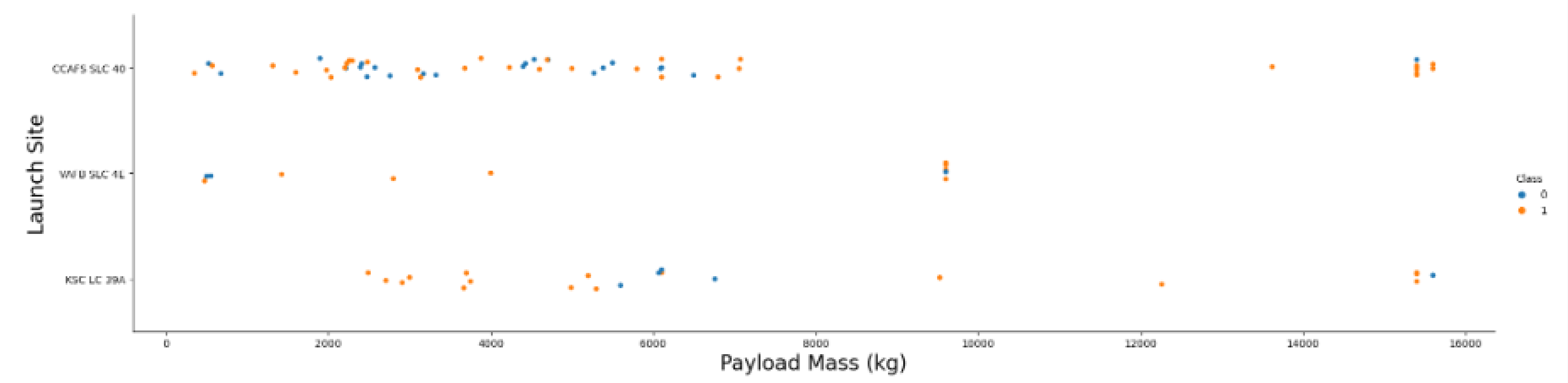
```
landing_class = [0 if outcome in bad_outcomes else 1 for outcome in df['Outcome']]
df['Class']=landing_class
```

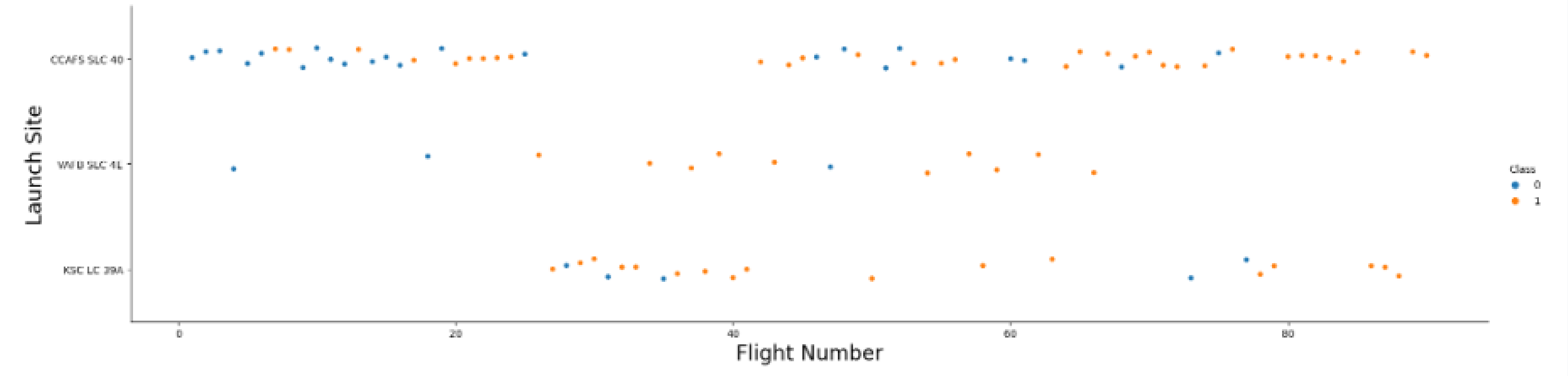
```
df.to_csv("dataset_part_2.csv", index=False)
```

URL Data wrangling notebook

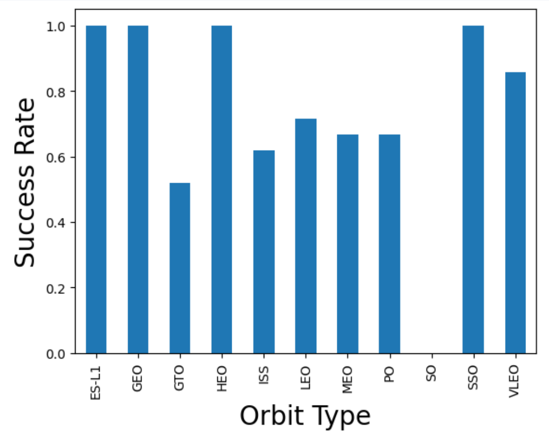
GitHub

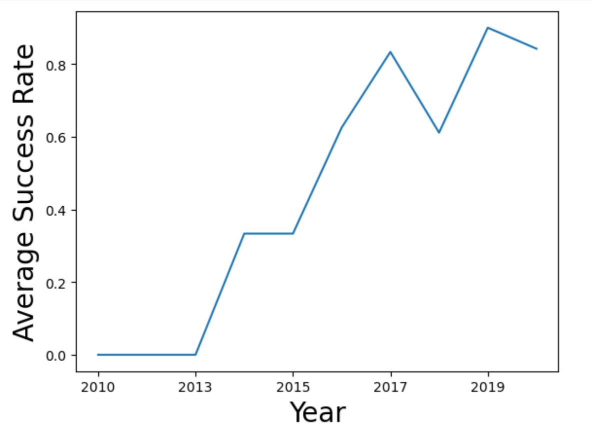
EDA with Data Visualization



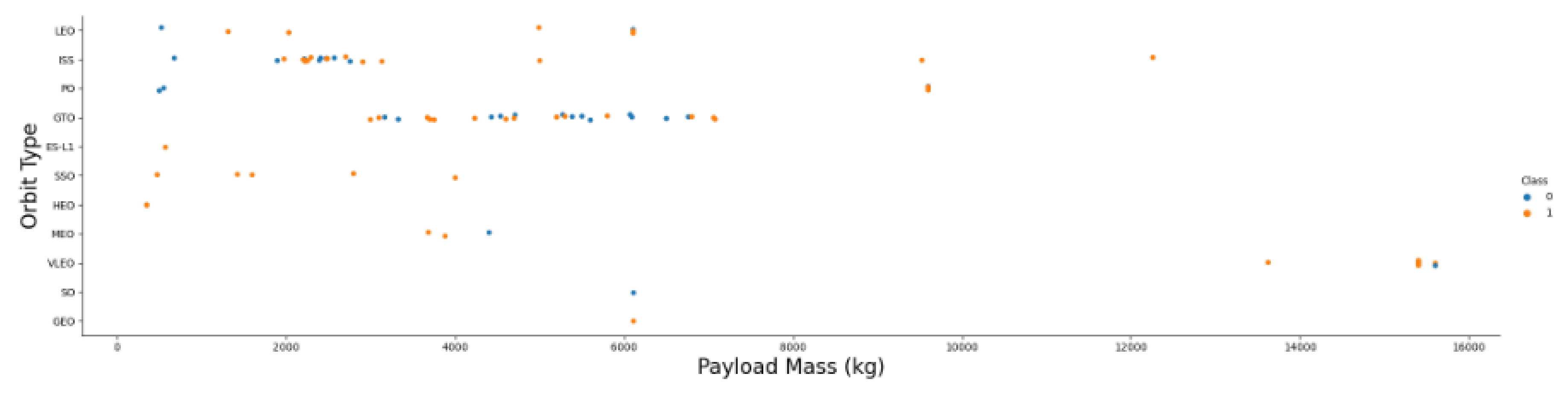


EDA with Data Visualization





EDA with Data Visualization



EDA with SQL

- **SQL QUERIES:**

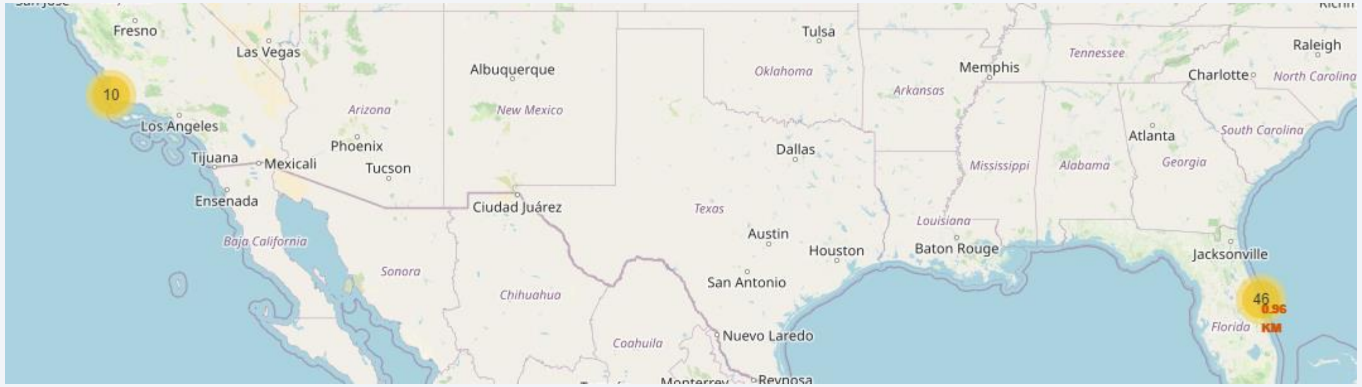
- **Display the names of the unique launch sites in the space mission**
- **Display 5 records where launch sites begin with the string 'CCA'**
- **Display the total payload mass carried by boosters launched by NASA (CRS)**
- **Display average payload mass carried by booster version F9 v1.1**
- **List the date when the first succesful landing outcome in ground pad was acheived.**
- **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 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. Use a subquery**
- **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 landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order.**

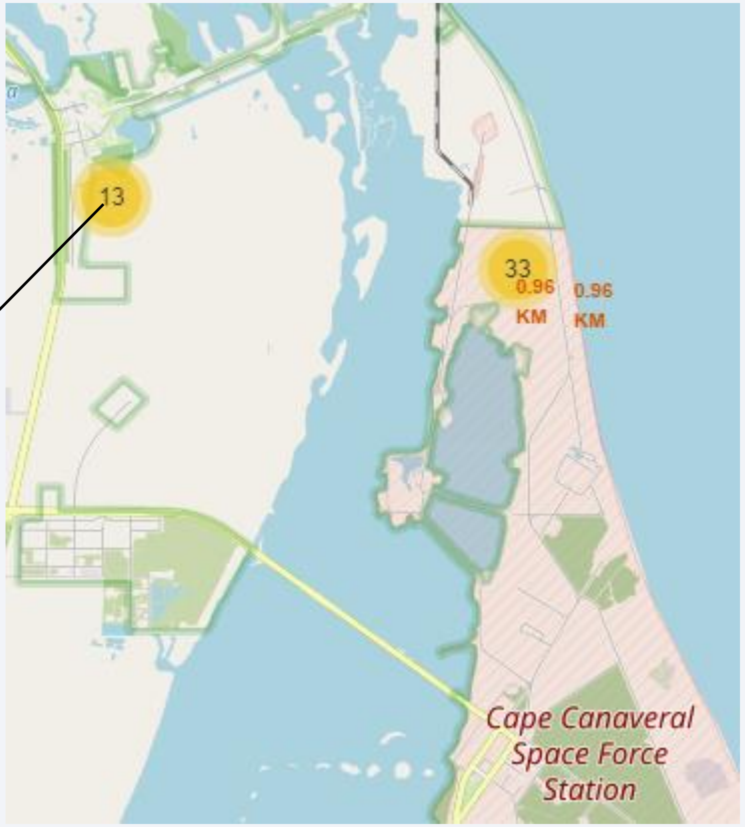
GitHub

EDA with SQL

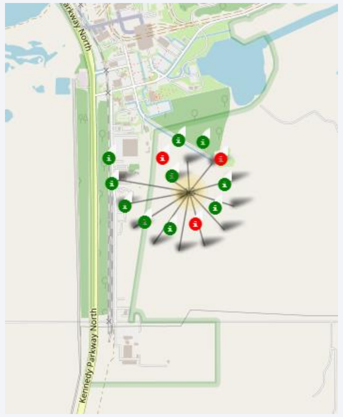
Build an Interactive Map with Folium

Map indicators have been incorporated onto the map in an effort to pinpoint the most suitable area for constructing a launch facility.





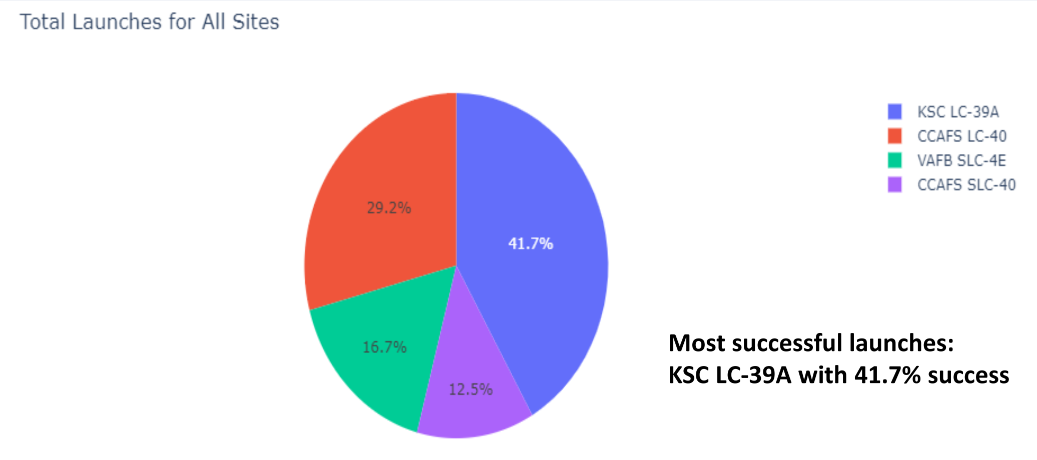
Kennedy Parkway North road in Merrit Island



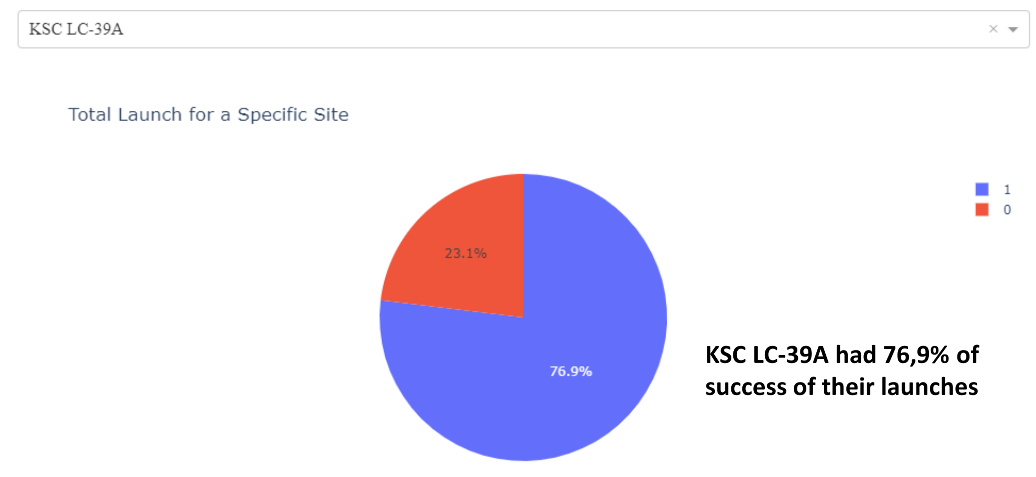
[Interactive Map with Folium]

* Download and open the notebook in a new browser. Images will not load correctly in GitHub

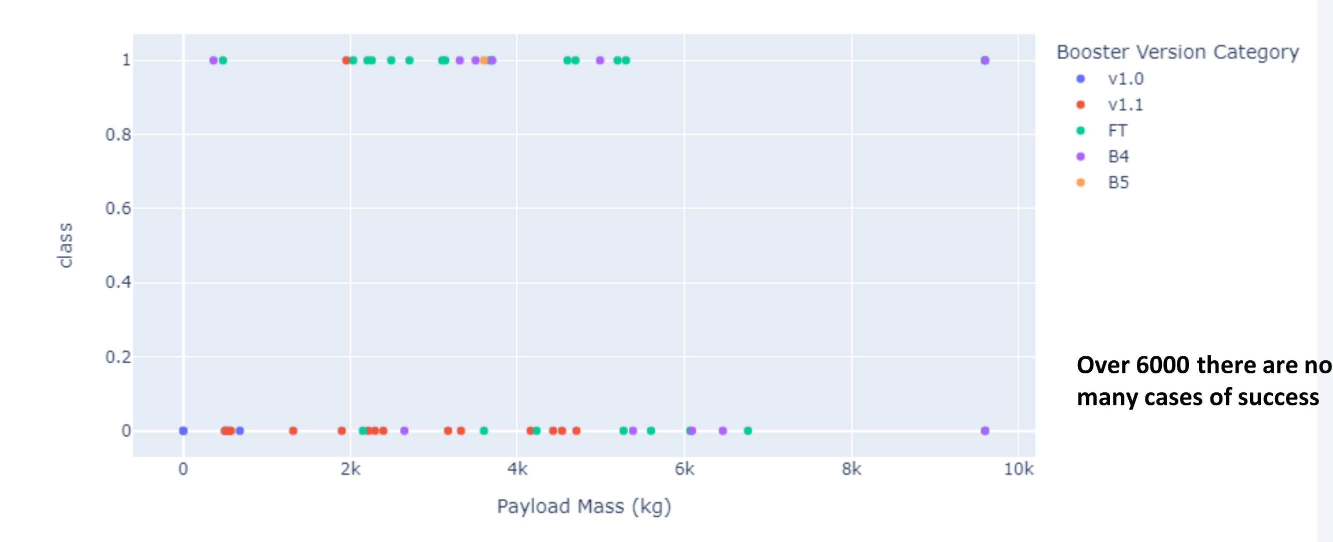
Build a Dashboard with Plotly Dash



Most successful launches:
KSC LC-39A with 41.7% success



KSC LC-39A had 76,9% of success of their launches



Over 6000 there are not too many cases of success

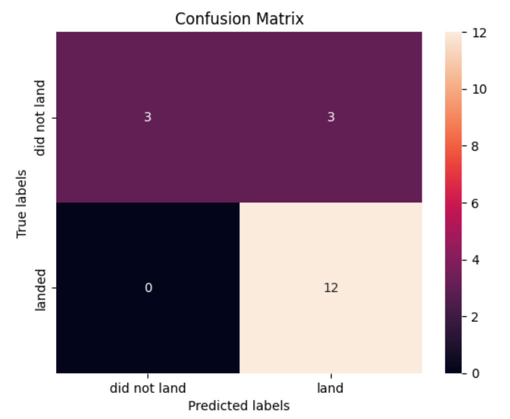
[Dashboard with Plotly Dash]

* Download and open the notebook in a new browser. Interactive dashboard will not load correctly in GitHub

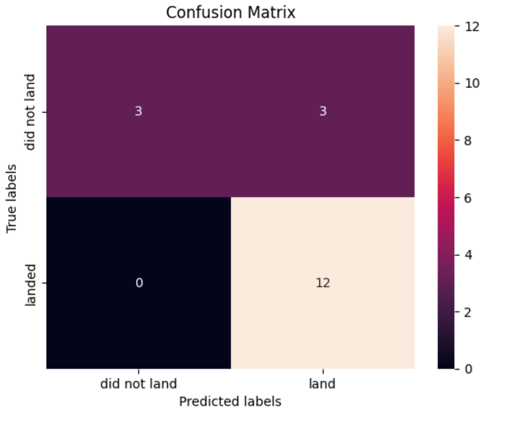
Predictive Analysis (Classification)

In predictive analysis, we have defined our target column as "Class," standardized the data, and split it into training and testing sets to determine the method that achieves the most accurate class classifications among: SVM, Classification Trees, Logistic Regression, and KNN:

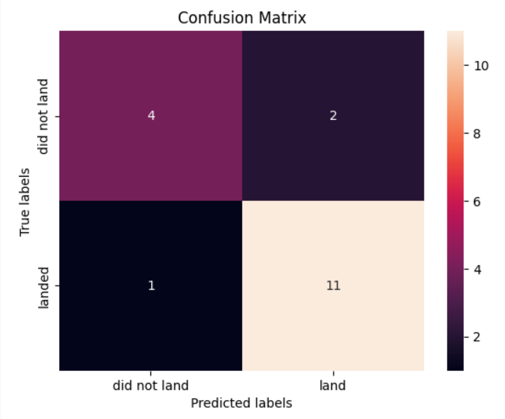
Logistic Regression



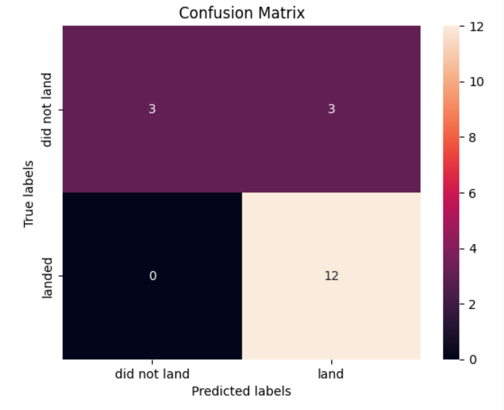
SVM



Classification Trees

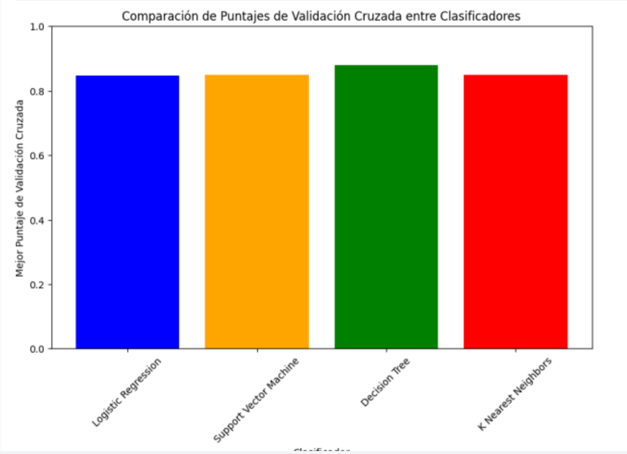


KNN



We have obtained similar results in all 4 confusion matrices. Only in the case of decision trees have we achieved a different classification, with higher accuracy in false positives, but lower accuracy in true positives.

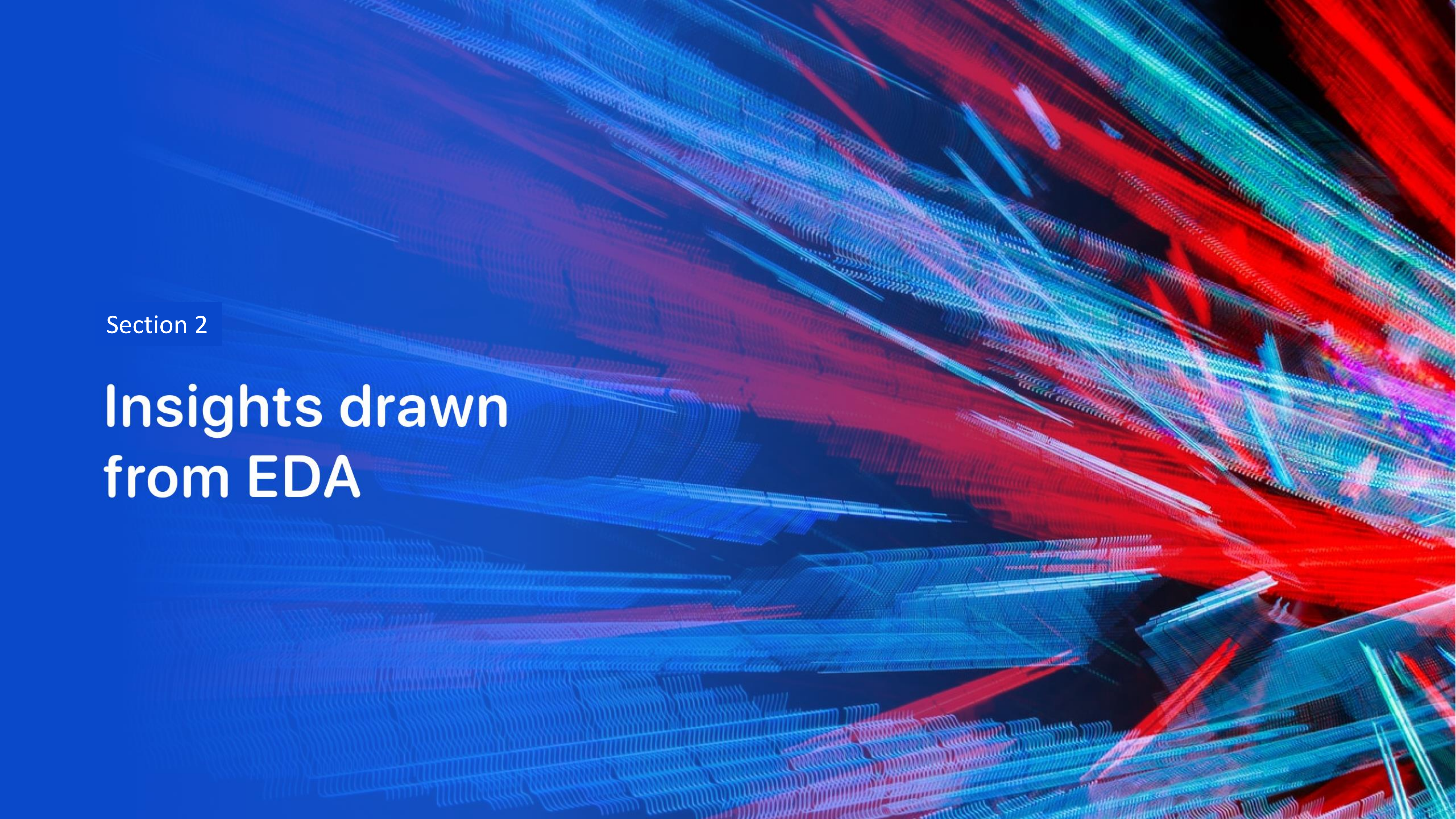
Nonetheless, it is the method that performs the best classifications with a "Best performance" of 0.8785.



[Predictive Analysis]

Results

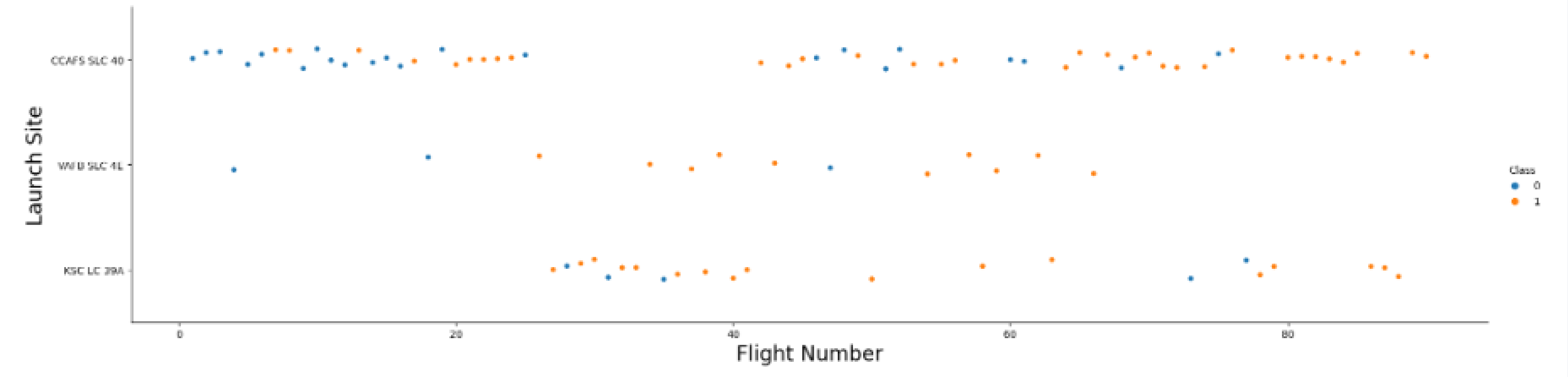
- In terms of prediction accuracy all models we analysed performed similarly. Tree Decision models have the best perform accuracy with 87,85%.
- KSC LC 39A have most successful launches
- Over 6000 kg there are not too many cases of success. Low weighted payloads perform much better.
- ES L1, HEO, GEO, SSO have the best success rate among rest.
- The average of success rate has increased in the last years
- Kennedy Parkway North road in Merrit Island is the most suitable area for constructing a launch facility.



Section 2

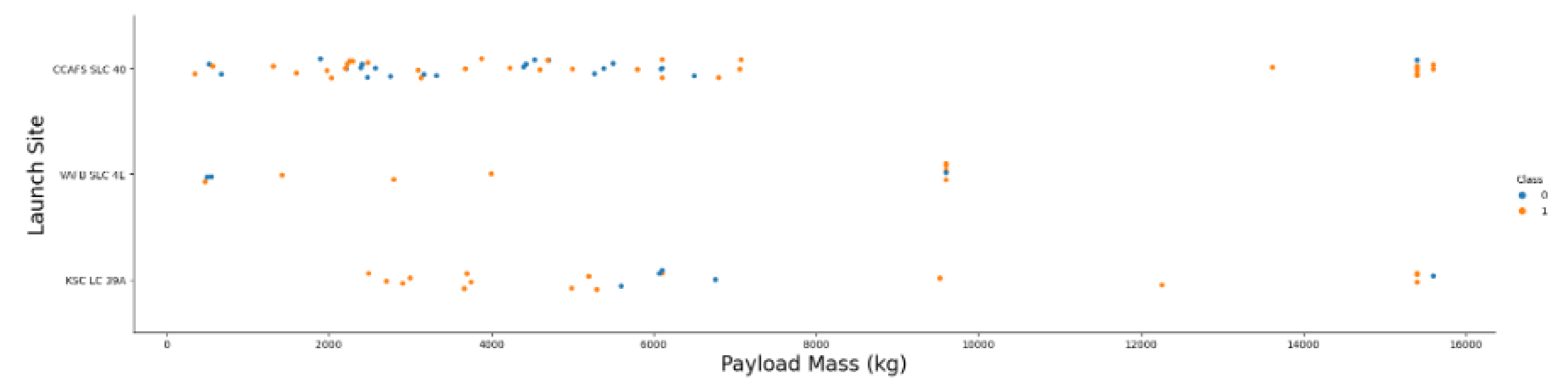
Insights drawn from EDA

Flight Number vs. Launch Site



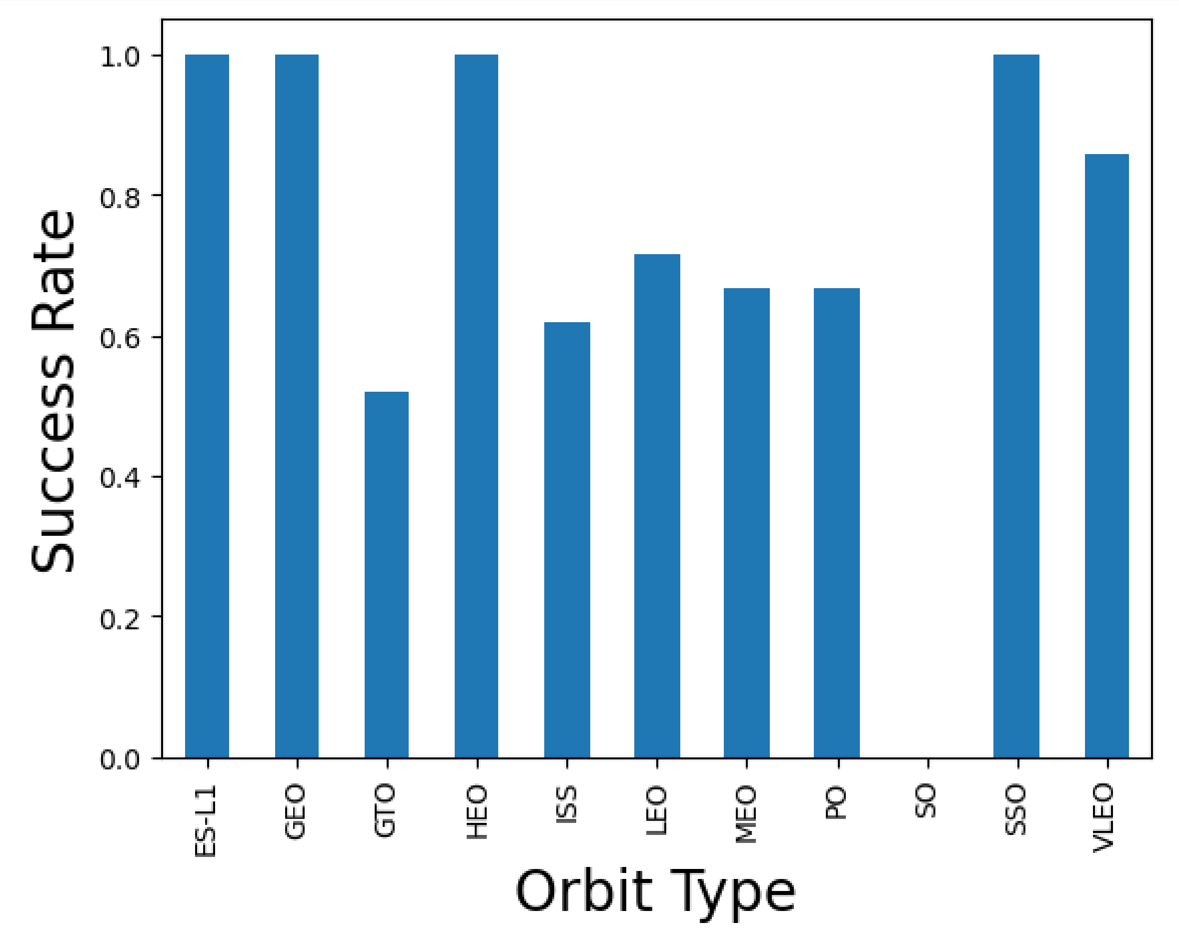
The majority of the launches have been carried out from the CCAFS SLC 40 site.

Payload vs. Launch Site



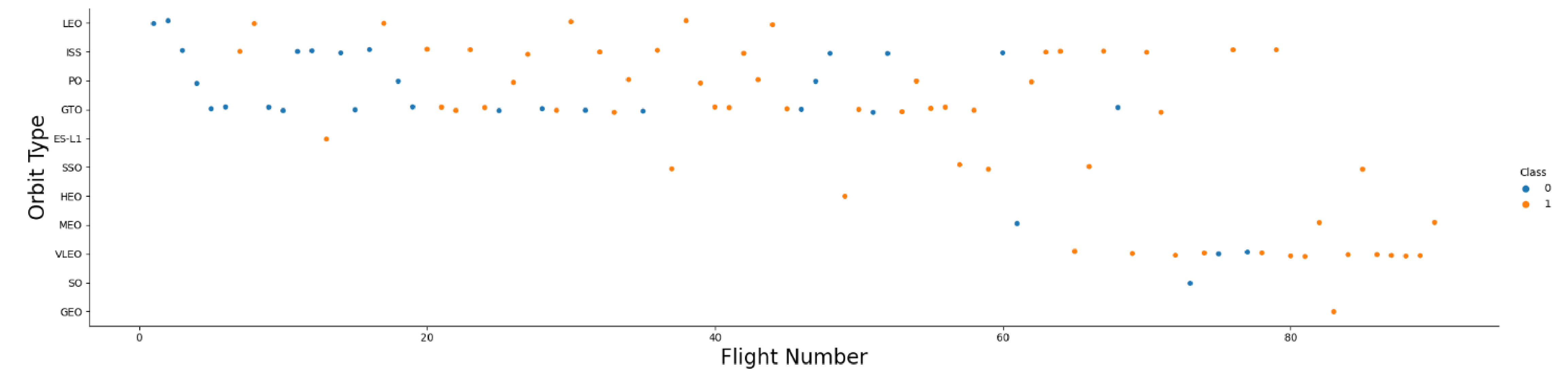
The majority of launches do not exceed 7000 kilograms in weight. Beyond that threshold, the success rate is lower.

Success Rate vs. Orbit Type



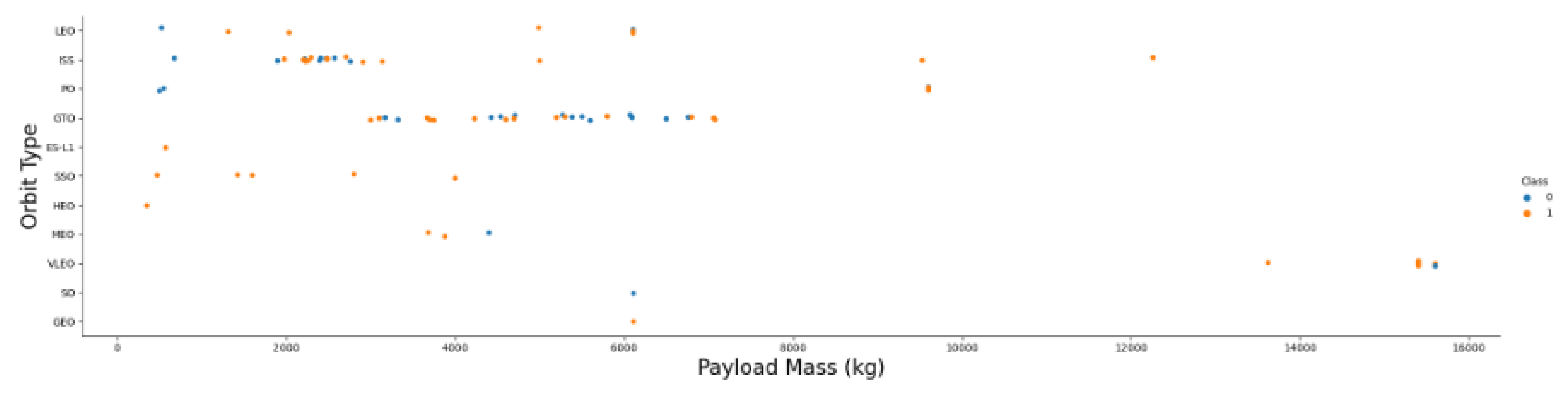
- ES L1, HEO, GEO, SSO have the best success rate among rest.

Flight Number vs. Orbit Type



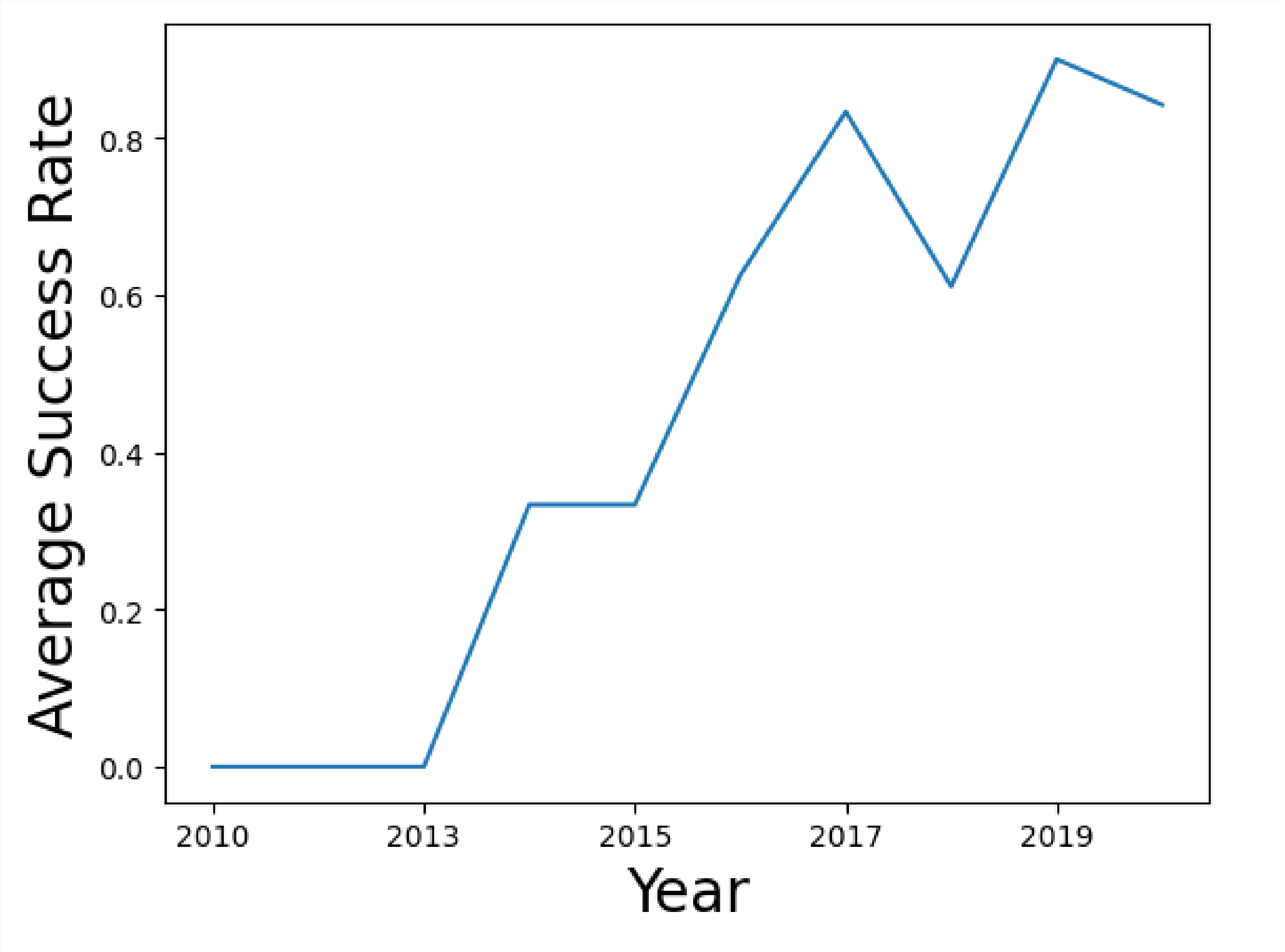
- Starting approximately from flight number 50, we observe a substantial proportion of successful launches. It is worth noting the case of VLEO orbit, where flights with a high success rate have prevailed in recent years.

Payload vs. Orbit Type



- There is a certain trend observed between launches with a payload of 2000 kg that are placed in the ISS orbit and those between 3000 and 7000 kg, which are placed in the GTO orbit.

Launch Success Yearly Trend

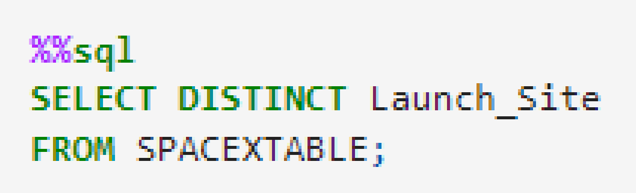


We can observe that in recent years, the average success rate in launches has significantly increased, achieving success rates of over 80%.

All Launch Site Names

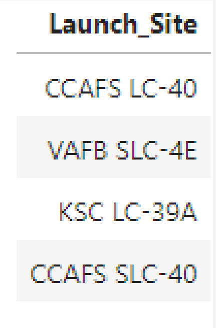
- Find the names of the unique launch site

Query:



```
%%sql
SELECT DISTINCT Launch_Site
FROM SPACEXTABLE;
```

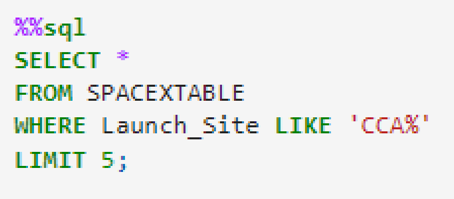
Outcome:



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

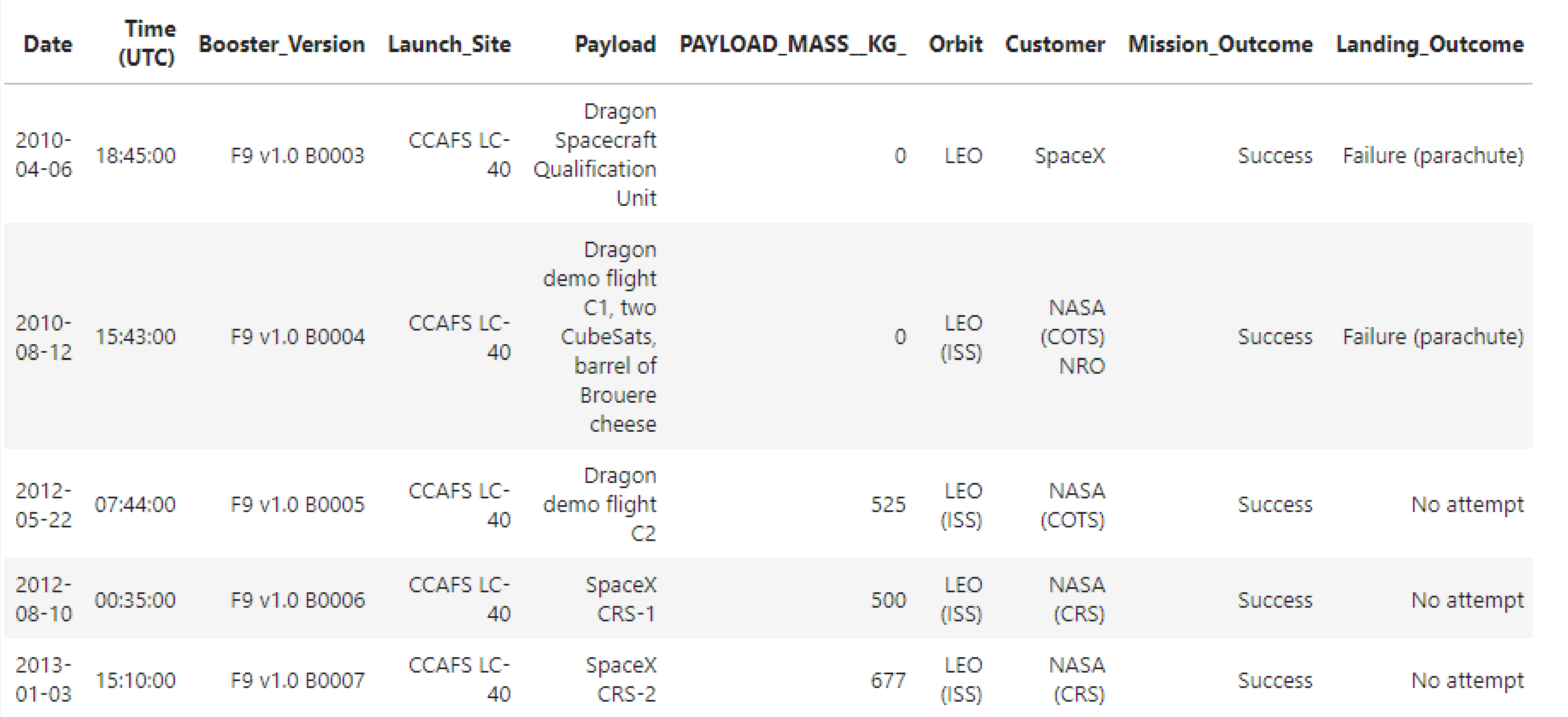
Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with `CCA`
- Query



```
%%sql
SELECT *
FROM SPACEXTABLE
WHERE Launch_Site LIKE 'CCA%'
LIMIT 5;
```

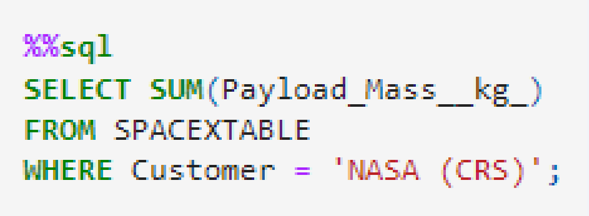
- Outcome →



Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-04-06	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-08-12	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
2012-08-10	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-01-03	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

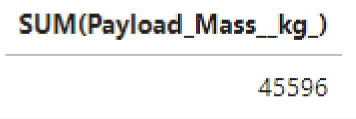
Total Payload Mass

- Calculate the total payload carried by boosters from NASA
- Query:



```
%%sql
SELECT SUM(Payload_Mass__kg_)
FROM SPACEXTABLE
WHERE Customer = 'NASA (CRS)';
```

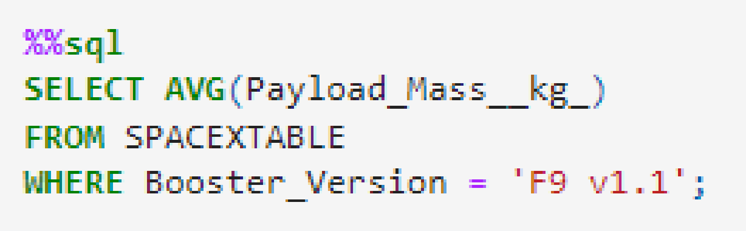
Outcome:



SUM(Payload_Mass__kg_)
45596

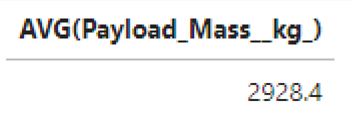
Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1
- Query:



```
%%sql
SELECT AVG(Payload_Mass__kg_)
FROM SPACEXTABLE
WHERE Booster_Version = 'F9 v1.1';
```

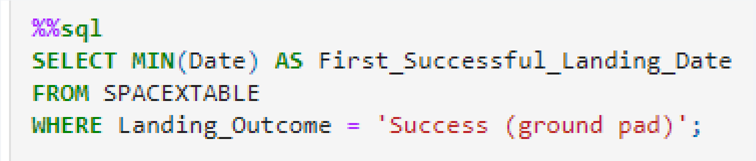
Outcome:



AVG(Payload_Mass__kg_)
2928.4

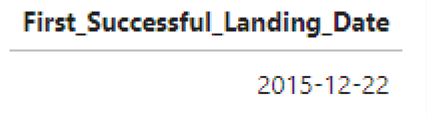
First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad
- Query:



```
%%sql
SELECT MIN(Date) AS First_Successful_Landing_Date
FROM SPACEXTABLE
WHERE Landing_Outcome = 'Success (ground pad)';
```

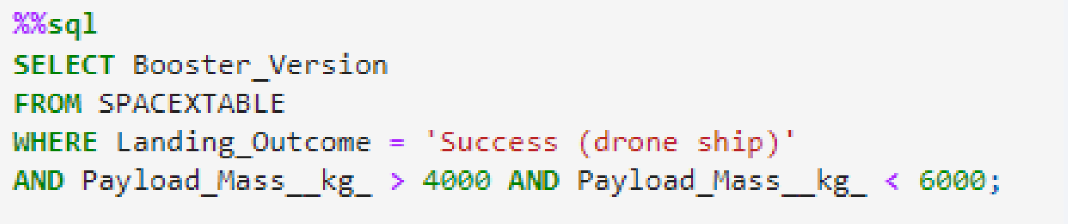
Outcome:



First_Successful_Landing_Date
2015-12-22

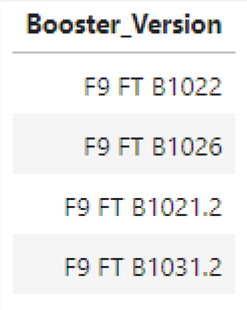
Successful Drone Ship Landing with Payload between 4000 and 6000

- List the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000
- Query:



```
%%sql
SELECT Booster_Version
FROM SPACEXTABLE
WHERE Landing_Outcome = 'Success (drone ship)'
AND Payload_Mass__kg_ > 4000 AND Payload_Mass__kg_ < 6000;
```

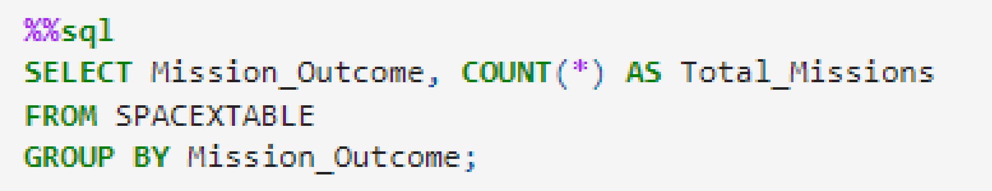
Outcome:



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

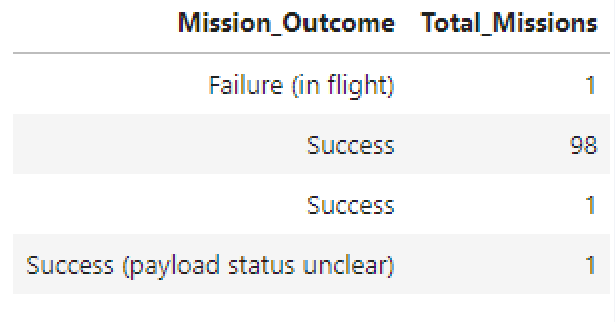
Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes
- Query:



```
%%sql
SELECT Mission_Outcome, COUNT(*) AS Total_Missions
FROM SPACEXTABLE
GROUP BY Mission_Outcome;
```

Outcome:

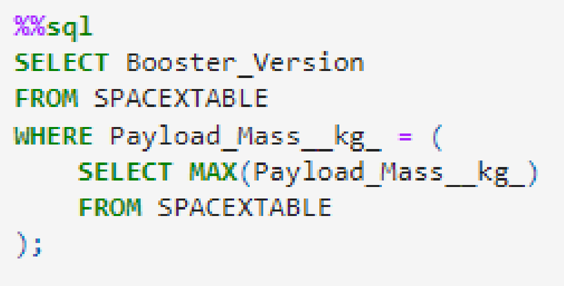


Mission_Outcome	Total_Missions
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

Boosters Carried Maximum Payload

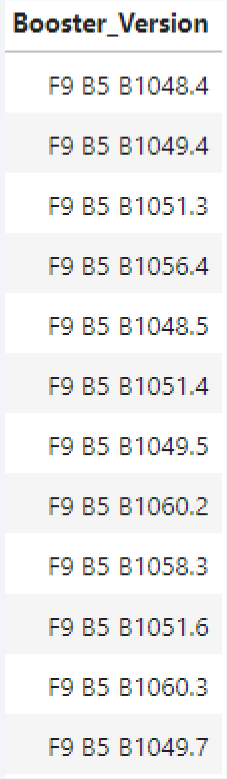
- List the names of the booster which have carried the maximum payload mass

- Query:



```
%%sql
SELECT Booster_Version
FROM SPACEXTABLE
WHERE Payload_Mass__kg_ = (
    SELECT MAX(Payload_Mass__kg_)
    FROM SPACEXTABLE
);
```

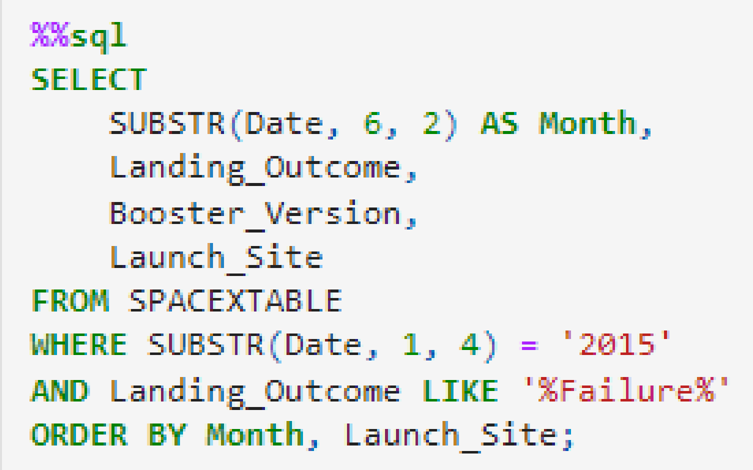
Outcome: →



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

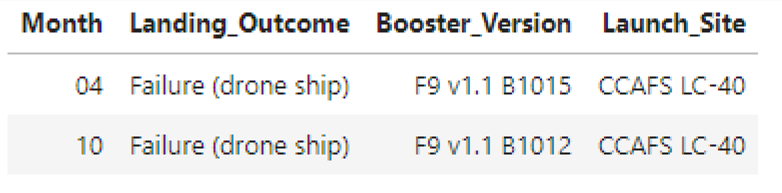
2015 Launch Records

- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
- Query:



```
%%sql
SELECT
    SUBSTR(Date, 6, 2) AS Month,
    Landing_Outcome,
    Booster_Version,
    Launch_Site
FROM SPACEXTABLE
WHERE SUBSTR(Date, 1, 4) = '2015'
AND Landing_Outcome LIKE '%Failure%'
ORDER BY Month, Launch_Site;
```

Outcome:

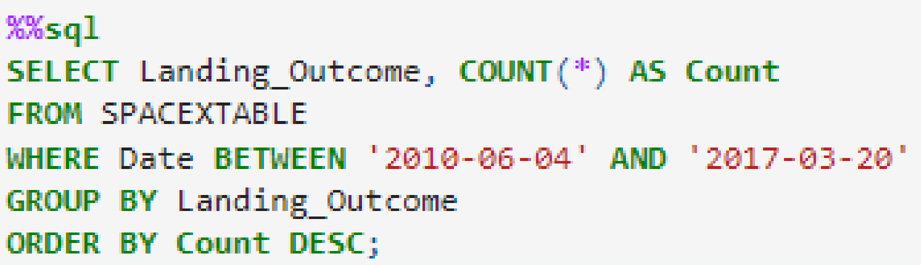


Month	Landing_Outcome	Booster_Version	Launch_Site
04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40
10	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40

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

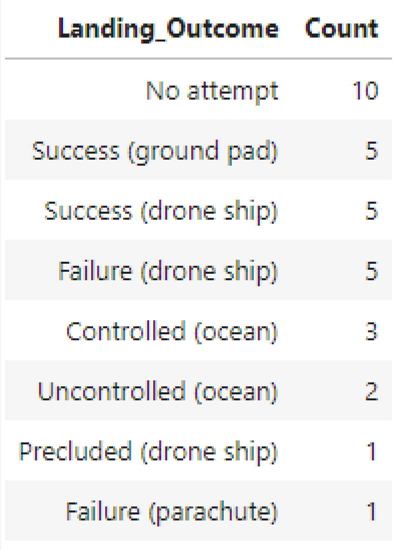
- Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order

- Query:



```
%%sql
SELECT Landing_Outcome, COUNT(*) AS Count
FROM SPACEXTABLE
WHERE Date BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY Landing_Outcome
ORDER BY Count DESC;
```

Outcome:

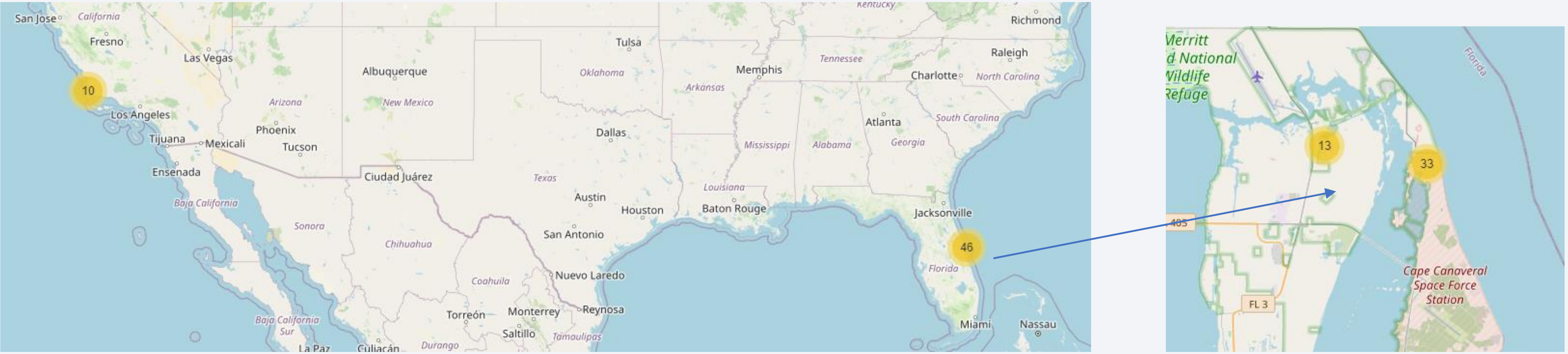


Landing_Outcome	Count
No attempt	10
Success (ground pad)	5
Success (drone ship)	5
Failure (drone ship)	5
Controlled (ocean)	3
Uncontrolled (ocean)	2
Precluded (drone ship)	1
Failure (parachute)	1

Section 3

Launch Sites Proximities Analysis

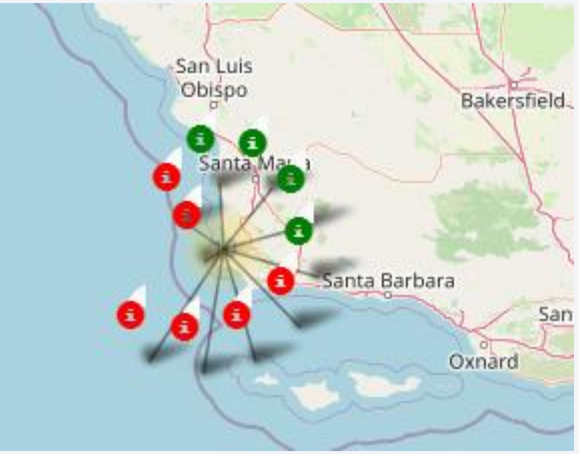
All launch sites



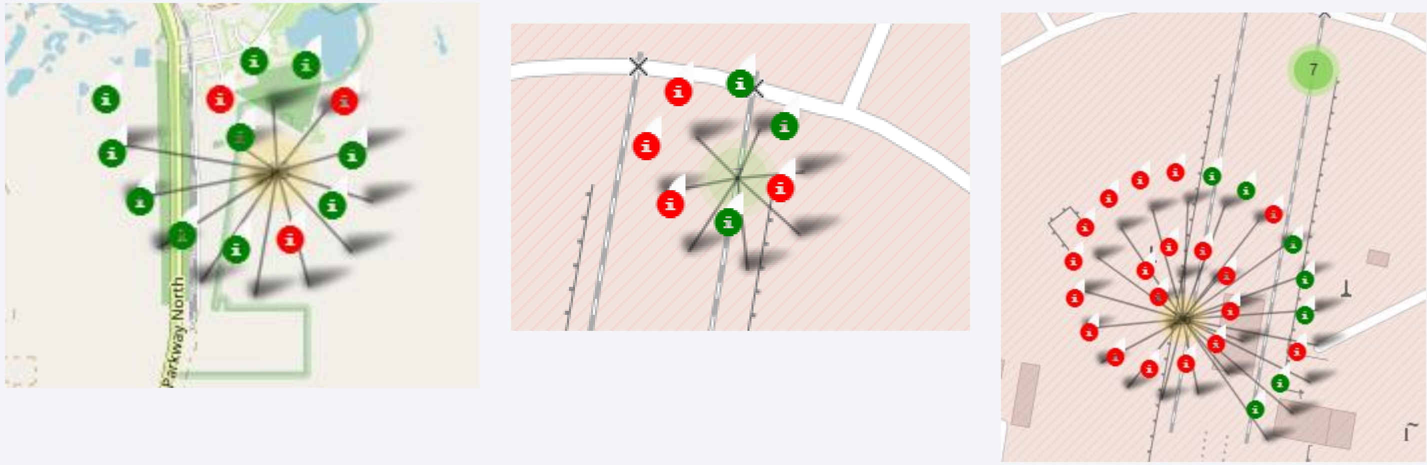
As we can observe, we find two well-differentiated points for the launch areas. On the east coast and on the west coast. If we zoom in on the map to the launch area on the east coast, we can appreciate that there are actually two sites where the takeoffs are carried out.

Color-labeled launch outcomes on the map

West Coast

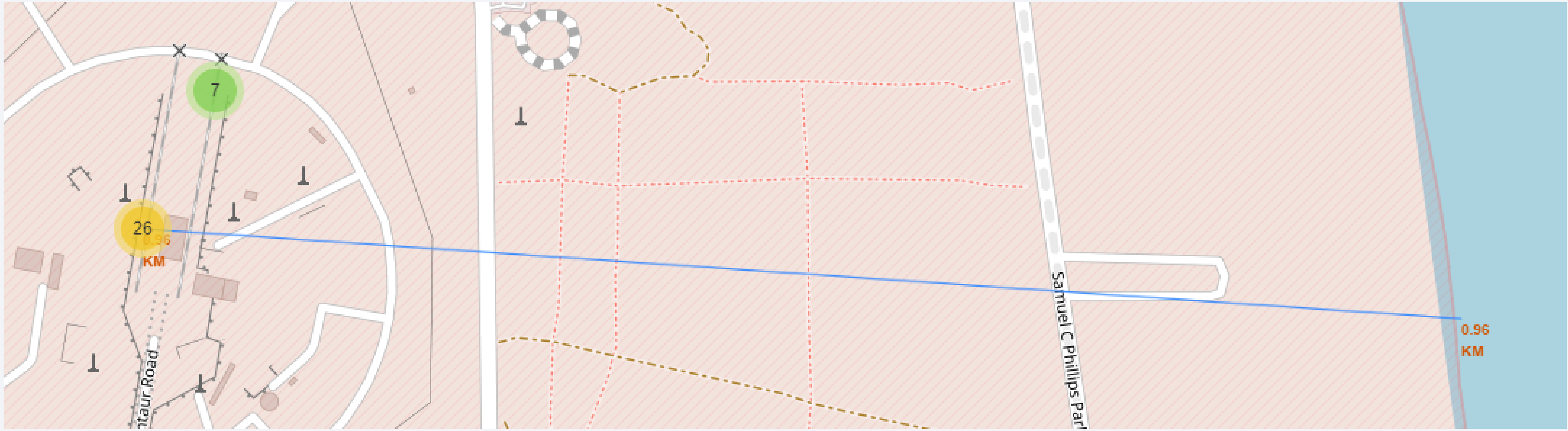


East Coast

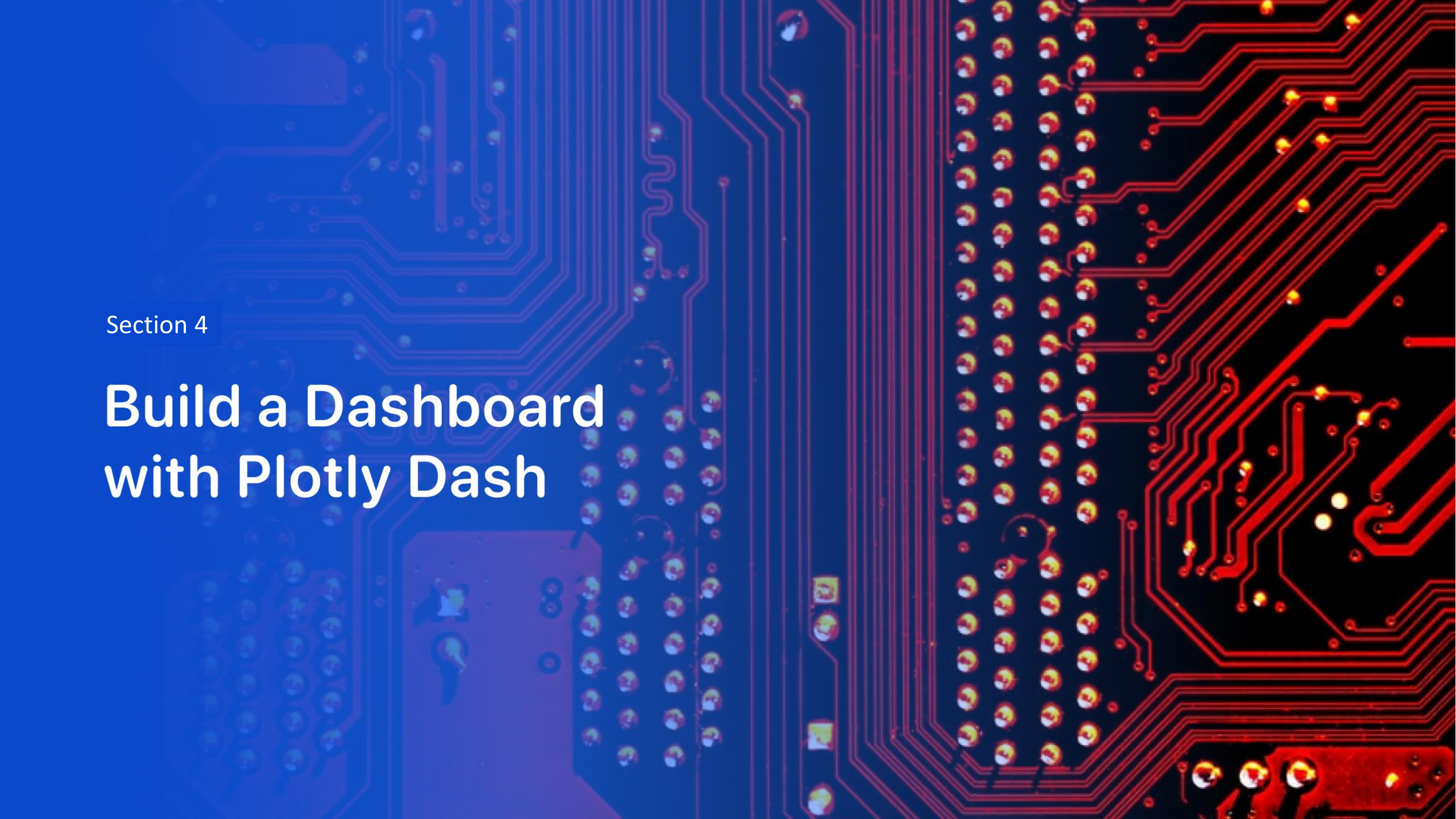


As we saw previously, there are two launch areas: the east coast and the west coast. On the east coast, many more launches take place, and it's the area near "Kennedy Parkway North" road that has the highest number of successful launch cases.

proximities to a railway, highway or coastline



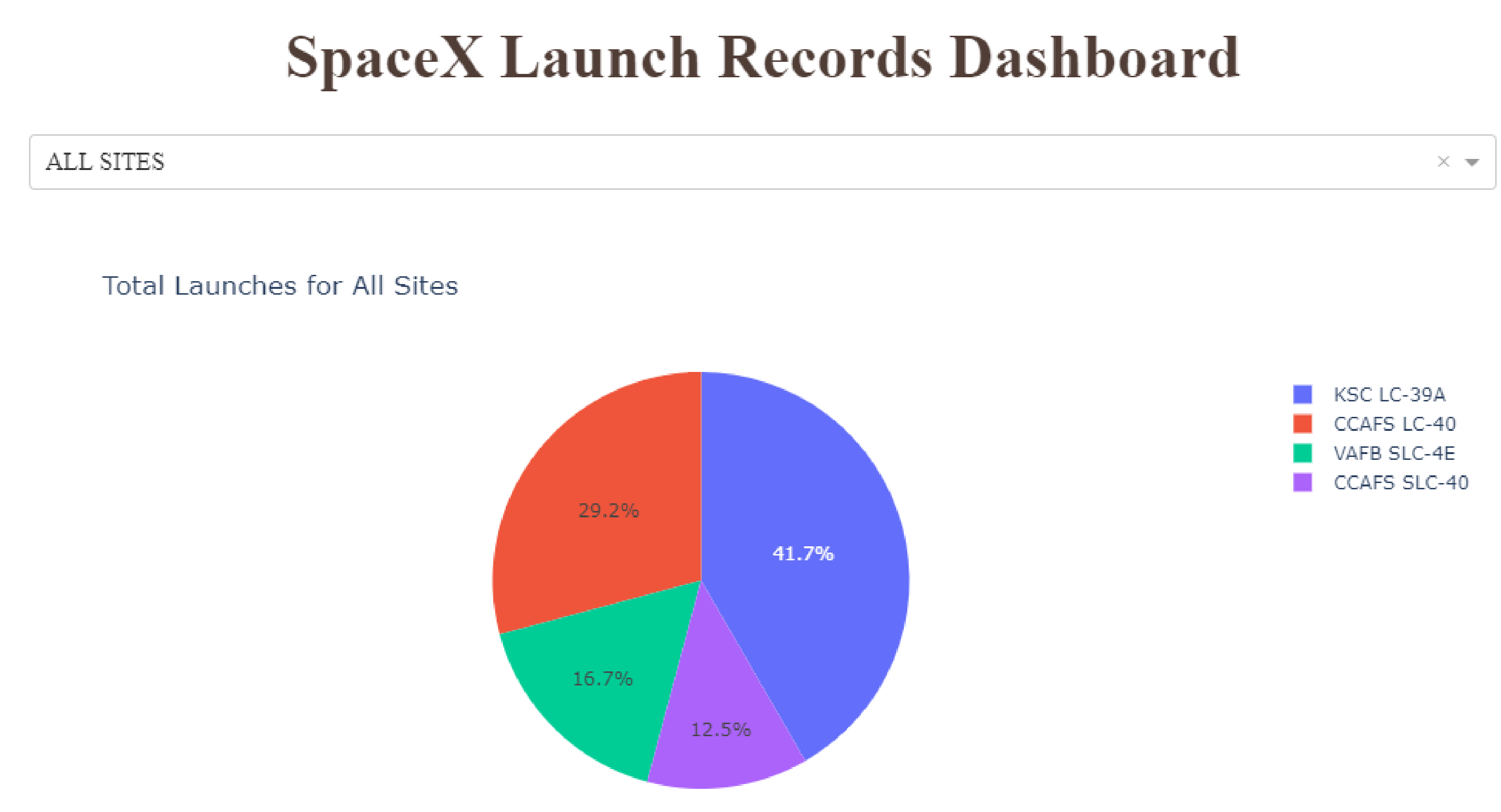
- As can be observed, there is a linear distance of 960 meters between the launch area and the coast.



Section 4

Build a Dashboard with Plotly Dash

Launch success count for all sites



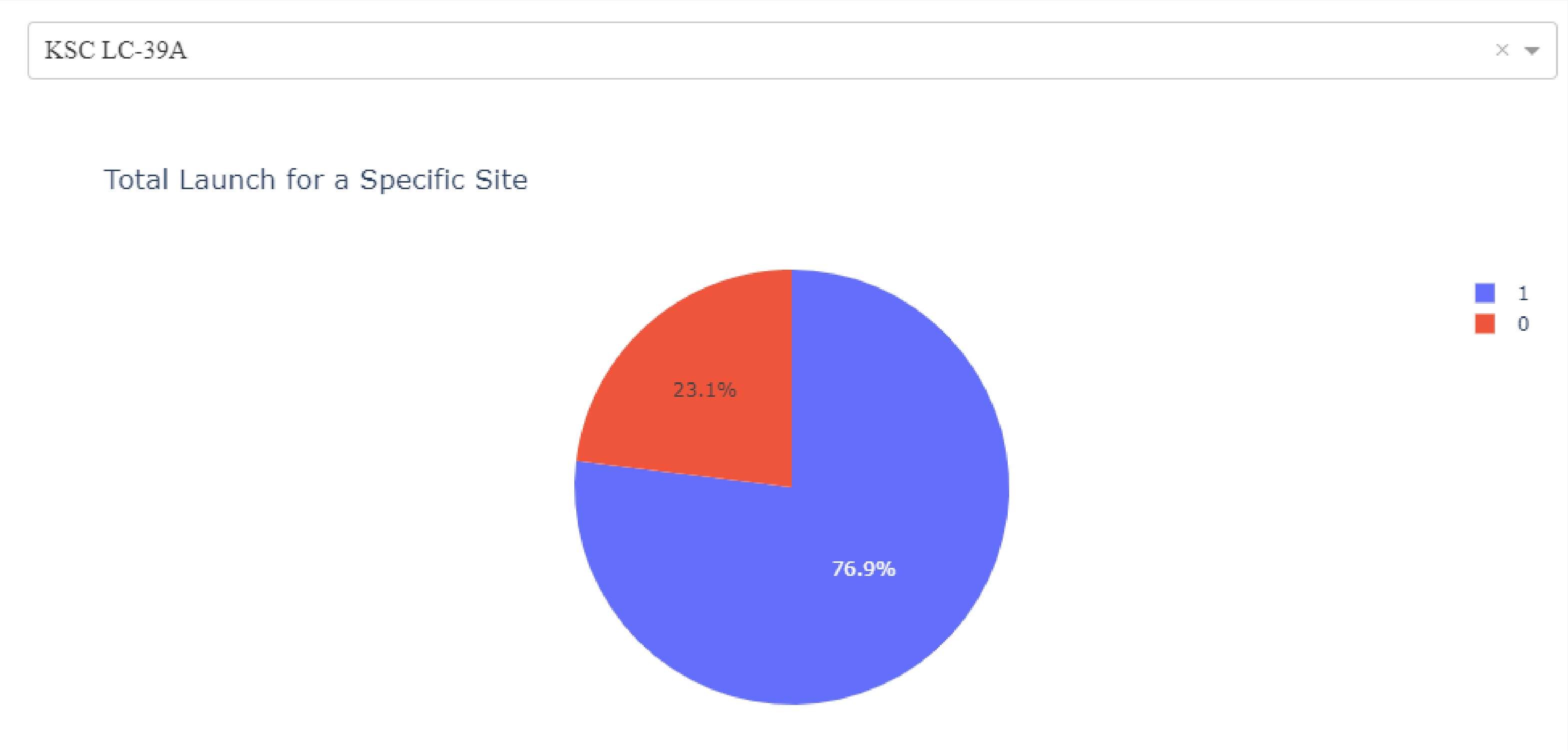
SpaceX Launch Records Dashboard

ALL SITES

- **Most successful launches:**

KSC LC-39A with 41.7% success

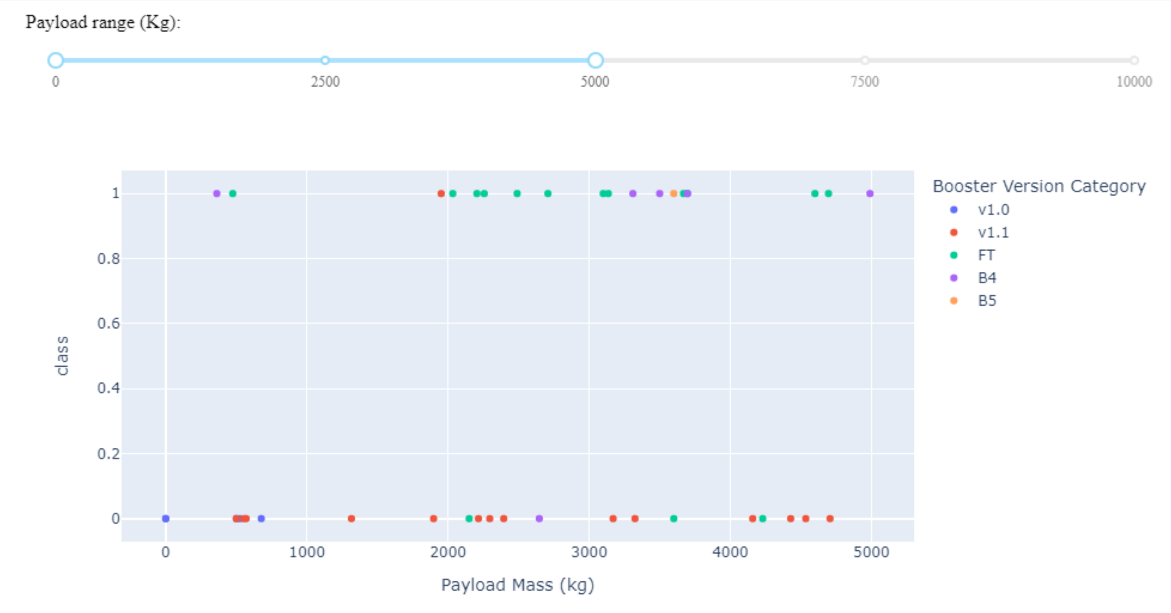
Launch site with highest launch success ratio

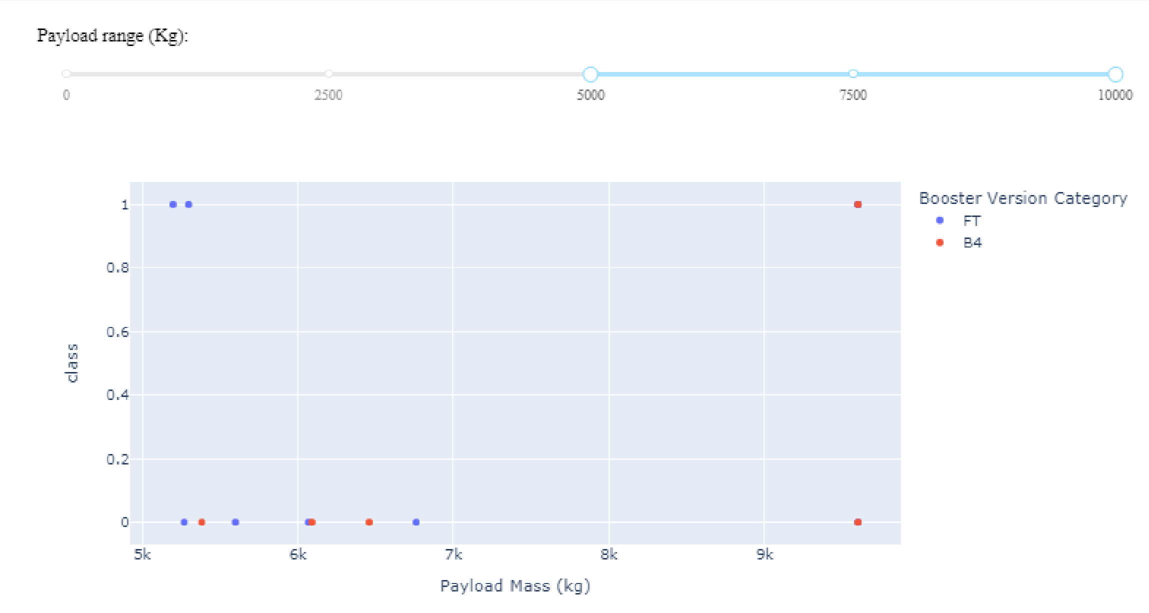


KSC LC-39A

Not only does "KSC LC-39 A" have the highest success ratio, but it also boasts a quite promising success percentage.

Different payloads ranges



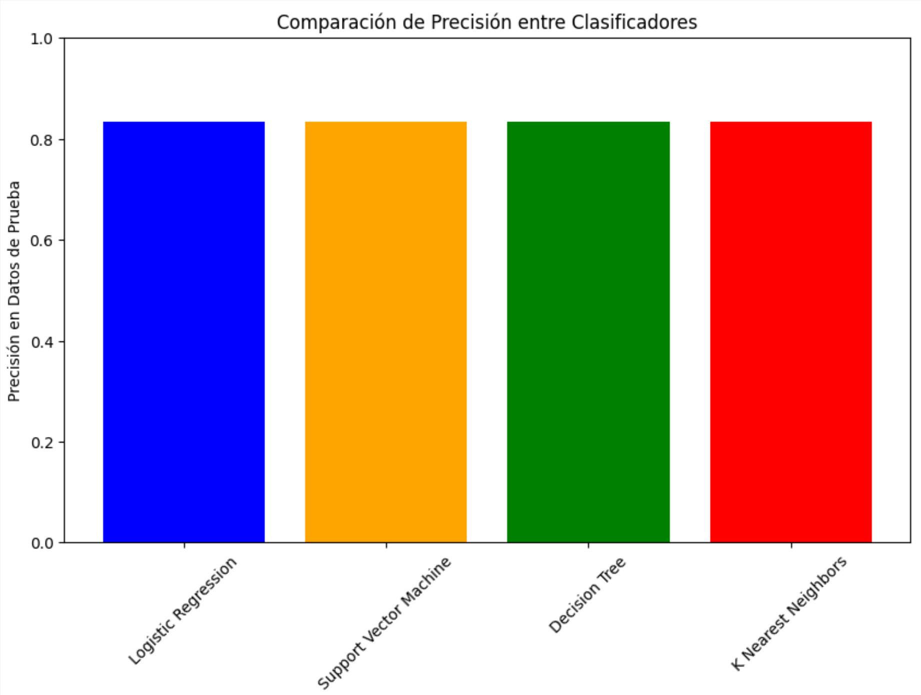


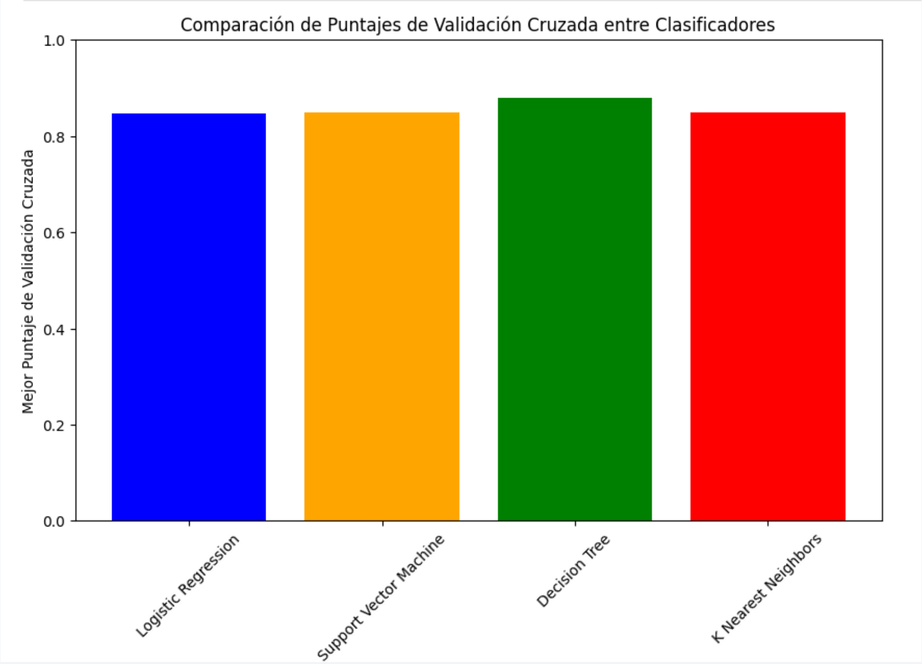
- In the first graph, we observe the range between 0 and 5 tons, and in the second graph, the range between 5 and 10 tons. Only the FT and B4 versions are manufactured for payload mass over 5000 kilograms.

Section 5

Predictive Analysis (Classification)

Classification Accuracy

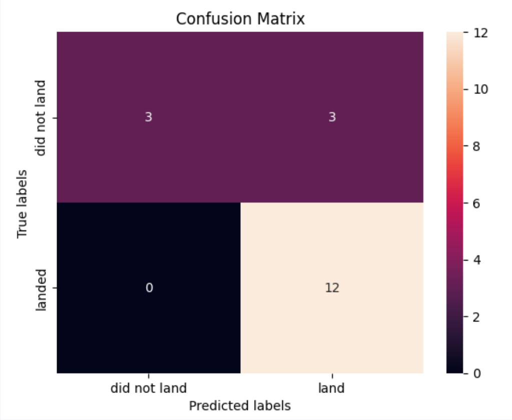




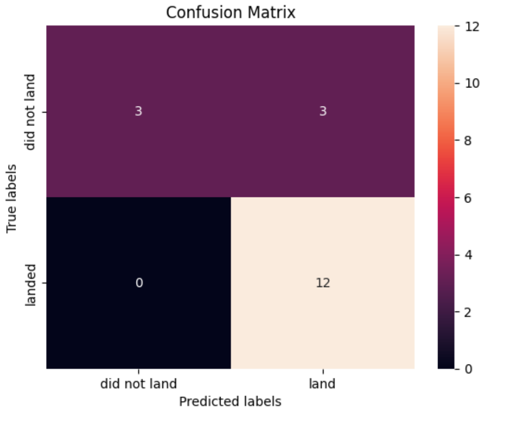
As we can see in the first graph, when comparing the accuracy among different classifiers on the test set, we obtain the same score, 0.83. However, when we compare the cross-validation scores among the classifiers, we observe that the decision tree algorithm achieves a slightly higher result with 0.87.

Confusion Matrix

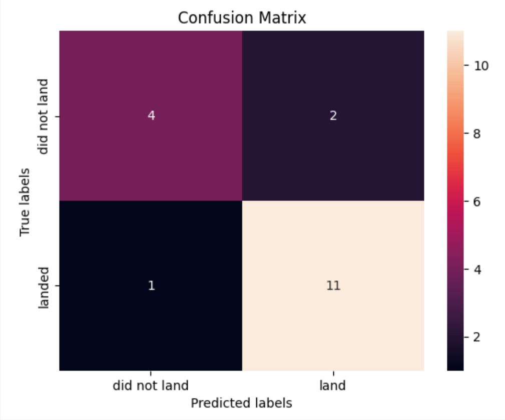
Logistic Regression



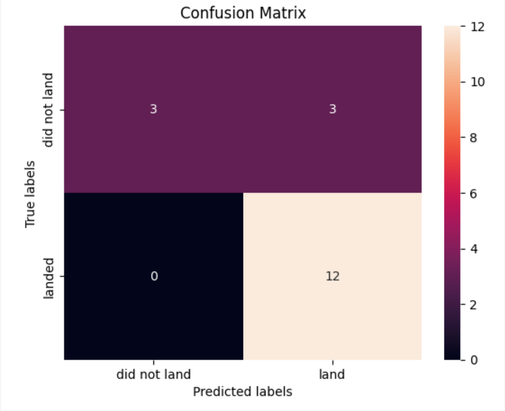
SVM



Classification Trees



KNN



As we can see, we obtain the same result in Logistic Regression, SVM, and KNN. The models predict very well when successful landings will occur, but they struggle to distinguish unsuccessful landings, yielding a 50% result due to having 3 false negatives and 3 true negatives.

The decision tree case yields somewhat different results, and while it doesn't classify cases with 100% accuracy when predicting successful landings, it achieves better outcomes when determining unsuccessful landings.

Conclusions

- When comparing the accuracy among different classifiers on the test set, we obtain the same score, however, when we compare the cross-validation scores tree algorithm achieves a slightly higher result
- Most successful launches are KSC LC-39A with 41.7% success
- The area near "Kennedy Parkway North" road that has the highest number of successful launch cases.
- ES L1, HEO, GEO, SSO orbit types have the best success rate among rest.
- Low weighted payloads perform much better than heavier.

Thank you!