IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Serkan MERAL
February 25, 2022

Outline

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

Executive Summary

- Summary of methodologies
  - ➢ Data Collection with Web Scraping
  - ➢ Data Wrangling
  - ➢ Exploratory Data Analysis with Visualization
  - ➢ Exploratory Data Analysis with SQL
  - ➢ Building Interactive Map with Folium
  - ➢ Building Interactive Dashboard with Plotly Dash
  - ➢ Predictive Analysis
- Summary of all results
  - ➢ Exploratory data analysis results
  - ➢ Interactive analytics demo in screenshots
  - ➢ Predictive analysis results

Introduction

- Project background and context

The commercial space age is here, companies including SpaceY are making space travel affordable for everyone. Perhaps the most successful one among our competitors is SpaceX. One reason for SpaceX's success is that their rocket launches are relatively inexpensive. SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upwards of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage. Therefore, if we can determine if the first stage will land, we can determine the cost of a launch. This information can be used if we want to bid against SpaceX for a rocket launch.

- Problems we want to find answers
  - ➢ The price of each launch
  - ➢ Whether first stage will be reused
  - ➢ Factors determining a successful landing
  - ➢ The relationship between different parameters and their affect on successful landing rate
  - ➢ Optimal conditions to ensure a successful landing

Section 1

Methodology

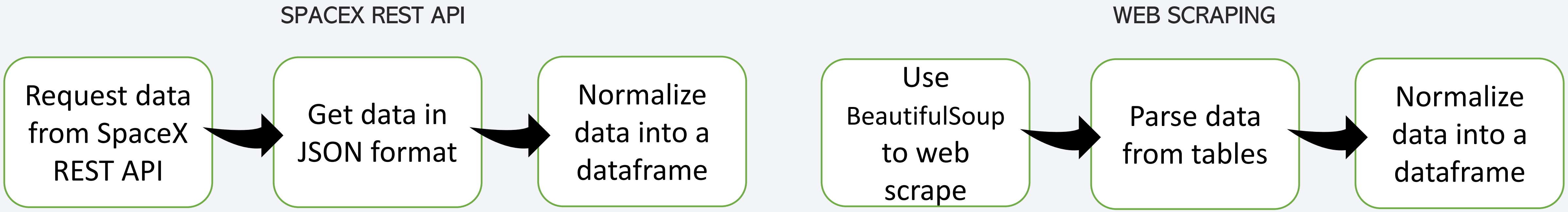
Methodology

Executive Summary

- Data collection methodology:
  - Data was collected using SpaceX API and web scraping List of Falcon 9 and Falcon Heavy launches Wikipedia page.
- Perform data wrangling
  - One hot encoding was applied to the column «Outcome» which was a categorical feature.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Data was standardized and split into test and training data. Best hyperparameters for SVM, Classification Trees and Logistic Regression was found and by using test data the model performing best was determined.

Data Collection

- The data was gathered from SpaceX REST API using a get request. Since the response was in the form of JSON, json_normalize function was used to convert data into a dataframe.
- Also, Python BeautifulSoup package was used to web scrape some HTML tables from List of Falcon 9 and Falcon Heavy launches Wikipedia page. Necessary data was parsed from tables and converted into a Pandas dataframe for further analysis.



SPACEX REST API

Request data from SpaceX REST API

Get data in JSON format

Normalize data into a dataframe

WEB SCRAPING

Use BeautifulSoup to web scrape

Parse data from tables

Normalize data into a dataframe

Data Collection – SpaceX API

1- Requesting rocket launch data from SpaceX API through get request

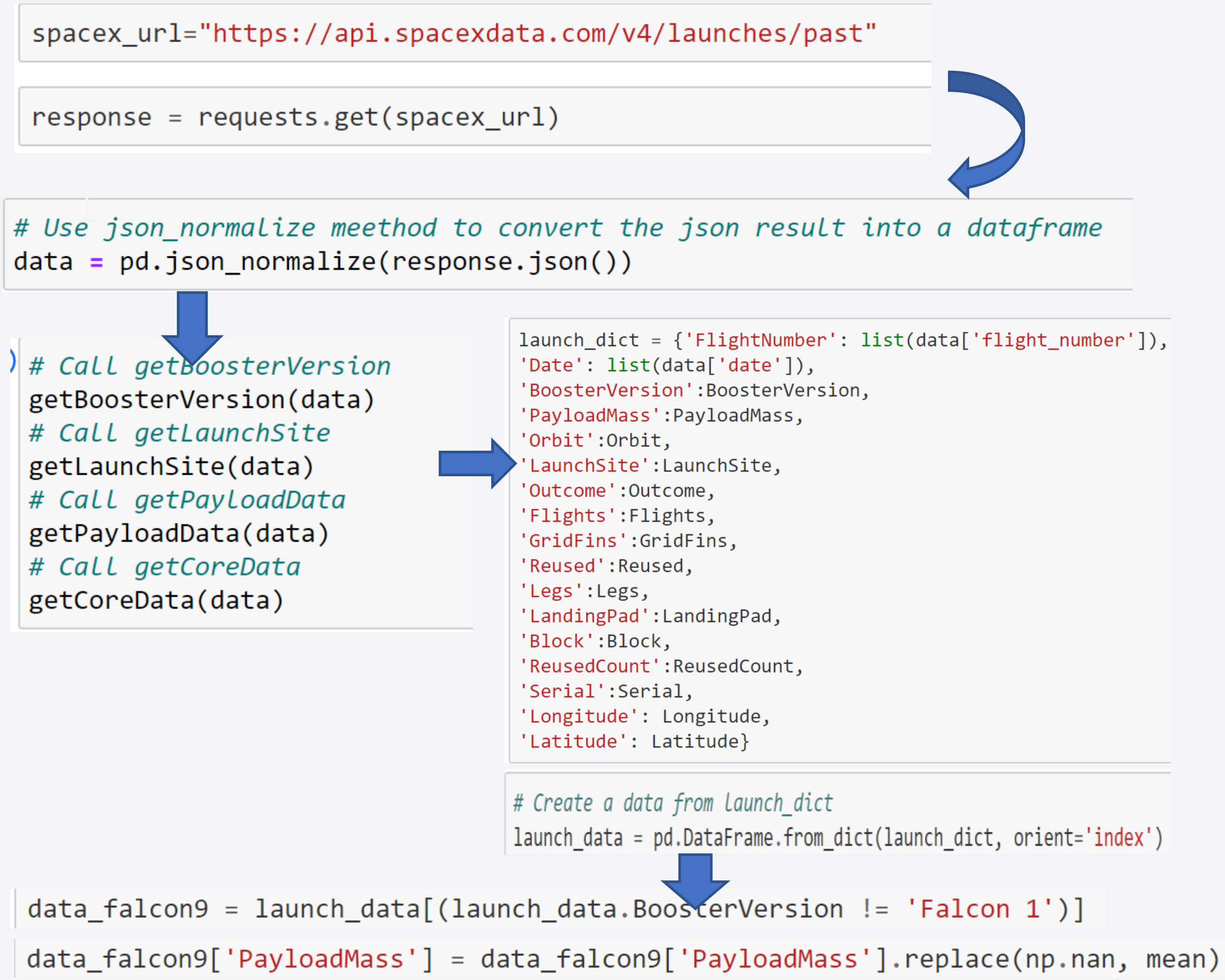
2- Normalizing data and turning it into a Pandas dataframe using json.normalize

3- Using customs functions to get information about the launches using the IDs given for each launch

4- Constucting dataset using the data obtained

5- Filtering the data and dealing with the missing labels

[GitHub URL of the completed SpaceX API calls notebook]()



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

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

```
# Use json_normalize meethod to convert the json result into a dataframe
data = pd.json_normalize(response.json())
```

```
# Call getBoosterVersion
getBoosterVersion(data)
# Call getLaunchSite
getLaunchSite(data)
# Call getPayloadData
getPayloadData(data)
# Call getCoreData
getCoreData(data)
```

```
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}
```

```
# Create a data from launch_dict
launch_data = pd.DataFrame.from_dict(launch_dict, orient='index')
```

```
data_falcon9 = launch_data[(launch_data.BoosterVersion != 'Falcon 1')]
```

```
data_falcon9['PayloadMass'] = data_falcon9['PayloadMass'].replace(np.nan, mean)
```

Data Collection - Scraping

1- Requesting data from web page

2- Creating a BeautifulSoup object

3- Finding all tables

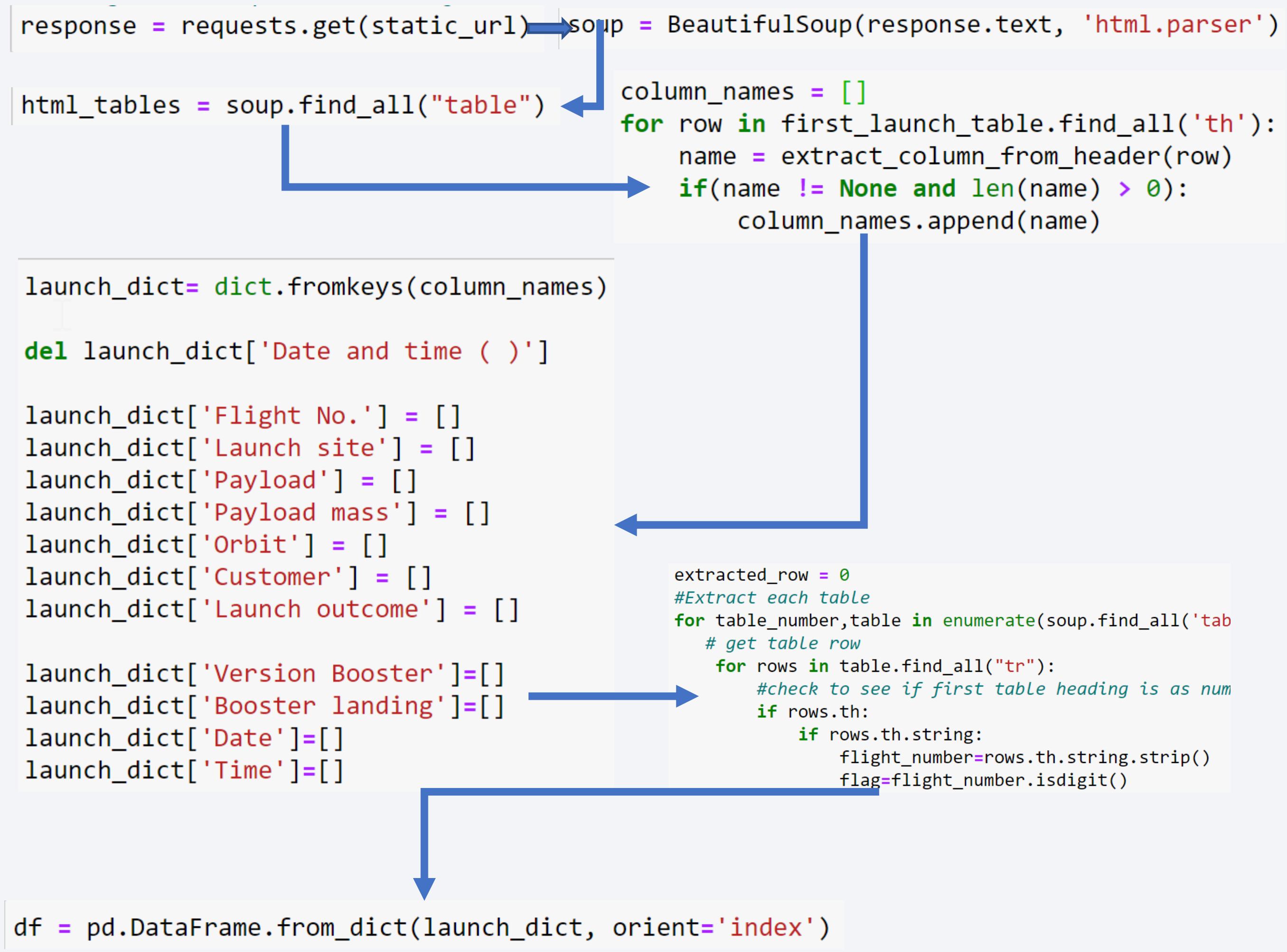
4- Extracting column names

5- Create dictionary

6- Fill up the dictionary

7- Create a dataframe

[GitHub URL of the completed web scraping notebook]()



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

```
soup = BeautifulSoup(response.text, 'html.parser')
```

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

```
column_names = []
for row in first_launch_table.find_all('th'):
    name = extract_column_from_header(row)
    if(name != None and len(name) > 0):
        column_names.append(name)
```

```
launch_dict= dict.fromkeys(column_names)

del launch_dict['Date and time ( )']

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'] = []

launch_dict['Version Booster']=[]
launch_dict['Booster landing']=[]
launch_dict['Date']=[]
launch_dict['Time']=[]
```

```
extracted_row = 0
#Extract each table
for table_number,table in enumerate(soup.find_all('tab
   # get table row
    for rows in table.find_all("tr"):
        #check to see if first table heading is as num
        if rows.th:
            if rows.th.string:
                flight_number=rows.th.string.strip()
                flag=flight_number.isdigit()
```

```
df = pd.DataFrame.from_dict(launch_dict, orient='index')
```

Data Wrangling

In the data set, there are several different cases where the booster did not land successfully. Sometimes a landing was attempted but failed due to an accident; for example, True Ocean means the mission outcome was successfully landed to a specific region of the ocean while False Ocean means the mission outcome was unsuccessfully landed to a specific region of the ocean. True RTLS means the mission outcome was successfully landed to a ground pad False RTLS means the mission outcome was unsuccessfully landed to a ground pad. True ASDS means the mission outcome was successfully landed on a drone ship False ASDS means the mission outcome was unsuccessfully landed on a drone ship. Those outcomes were converted into Training Labels with 1 means the booster successfully landed 0 means it was unsuccessful.

Perform exploratory Data Analysis and determine Training Labels

Calculate the number of launches on each site

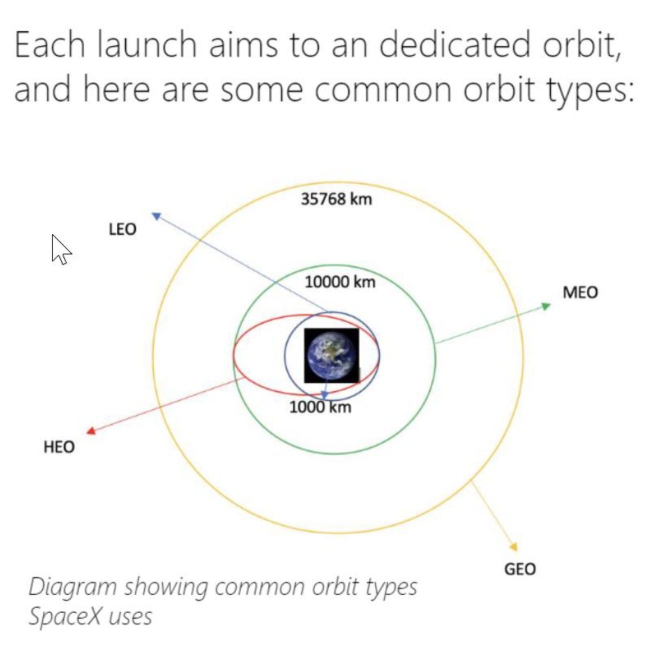
Calculate the number and occurrence of each orbit

Calculate the number and occurence of mission outcome per orbit type

Create a landing outcome label from Outcome column

Calculate the success rate for every landing in dataset

[GitHub URL of completed data wrangling related notebook]

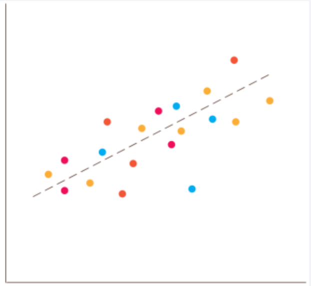


Each launch aims to an dedicated orbit, and here are some common orbit types:

Diagram showing common orbit types SpaceX uses

EDA with Data Visualization

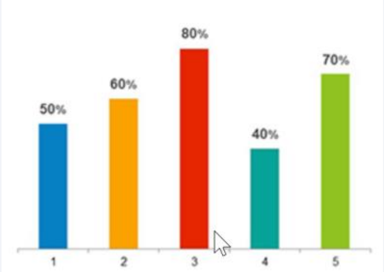
- Scatter Charts Drawn:
  - Flight Number vs Payload Mass
  - Flight Number vs Launch Site
  - Payload vs Launch Site
  - Flight Number vs Orbit
  - Payload Mass vs Orbit



Scatter plots show how much one variable is affected by another. The relationship between two variables is called their correlation.

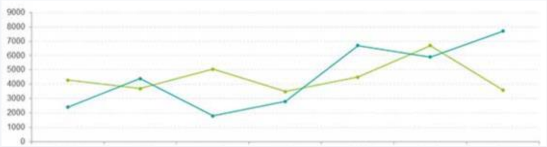
[GitHub URL of completed EDA with data visualization notebook.](#)

- Bar Graph Drawn
  - Orbit vs Success Rate



A bar diagram makes it easy to compare sets of data between different groups at a glance. The graph represents categories on one axis and a discrete value in the other. The goal is to show the relationship between the two axes.

- Line Chart Drawn
  - Year vs Success Rate



Line charts are useful in that they show data variables and trends very clearly and can help to make predictions about the results of data not yet recorded

EDA with SQL

SQL queries performed to answer following questions:

- Displaying the names of the unique launch sites in the space mission
- Displaying 5 records where launch sites begin with the string 'CCA’
- Displaying the total payload mass carried by boosters launched by NASA (CRS)
- Displaying average payload mass carried by booster version F9 v1.1
- Listing the date when the first successful landing outcome in ground pad was achieved
- Listing the names of the boosters which have success in drone ship and have payload mass greater than 4.000 but less than 6.000
- Listing the total number of successful and failure mission outcomes
- Listing the names of the booster versions which have carried the maximum payload mass
- Listing the failed landing outcomes in drone ship, their booster versions, and launch site names for year 2015
- Ranking 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 URL of completed EDA with SQL notebook

Build an Interactive Map with Folium

- Map objects added to folium map:
  - Circles and markers to display all launch sites on the map
  - Green/red markers to display successful/failed launches on the map
  - Markers to display distances between selected points and launch sites
  - Polylines between selected points and launch sites

Questions answered:

- Are launch sites in close proximity to railways? YES
- Are launch sites in close proximity to highways? YES
- Are launch sites in close proximity to coastline? YES
- Do launch sites keep certain distance away from cities? YES

GitHub URL of completed interactive map with Folium map

Build a Dashboard with Plotly Dash

- Dashboard contains a dropdown menu, a pie chart, a slider and a scatter chart.
- Drop-down Input Component and pie chart was used to first see which launch site has the largest success count and then to select one specific site and check its detailed success rate.
- Slider was added to find if variable payload is correlated to mission outcome. The idea was to be able to easily select different payload range in order to identify some visual patterns.
- Scatter plot was used to visually observe how payload and different boosters may be correlated with mission outcomes for selected site(s).

GitHub URL of completed Plotly Dash lab.

Predictive Analysis (Classification)

- Building the Model
    - Load data and create a column for class
    - Standardize the data and split it into training and test data
    - Determine the algorithms to be used (Logistic regression, SVM, Decision tree classifier, K-nearest neighbors)
    - Set parameters and train the model
- Evaluating Model
    - Check accuracy for each model and plot confusion matrix
- Improving Model
    - Create dictionaries containing different parameter values for each model
    - Create a GridSearchCV object to find the hyperparameter values that allow each model to perform best
    - Repeat evaluation process
- Finding the Best Performing Classification Model
    - Determine the model with the best accuracy using the training data

GitHub URL of completed predictive analysis lab

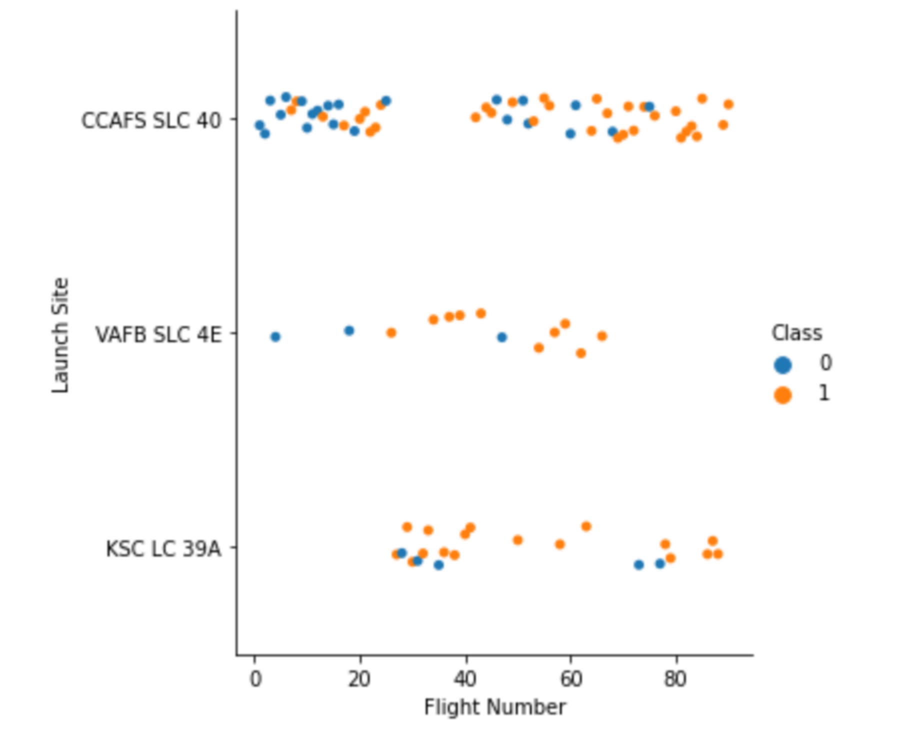
Results

- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

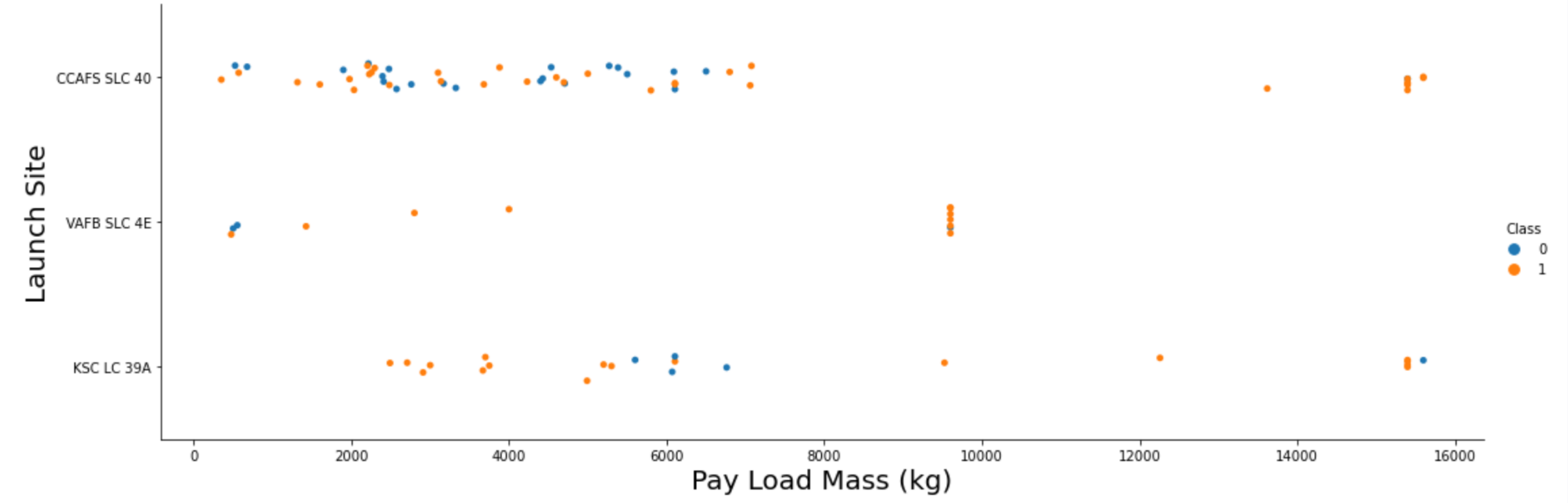
Insights drawn from EDA

Flight Number vs. Launch Site



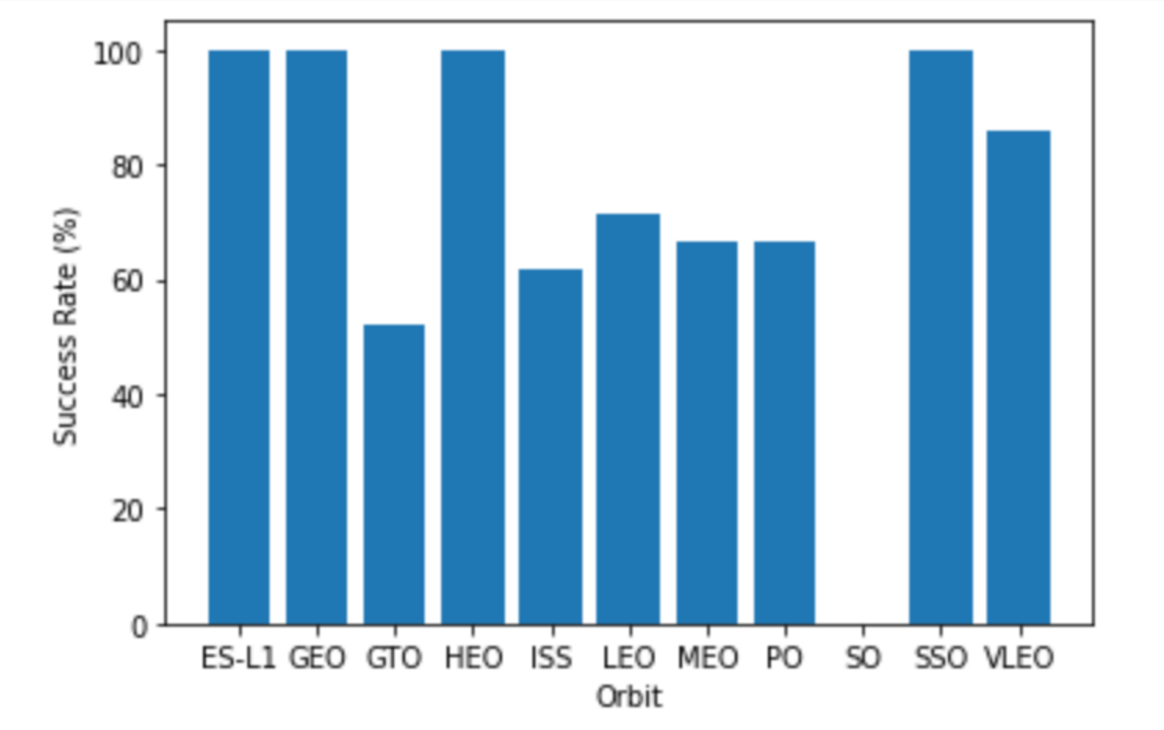
- Class 0 represents unsuccessful landings and Class 1 represents successful landings.
- As the number of flight increases so as the success rate.

Payload vs. Launch Site



- Class 0 represents unsuccessful landings and Class 1 represents successful landings.
- It seems that as the payload mass increases so as the success rate however there is not a clear relationship between payload mass and launch site.

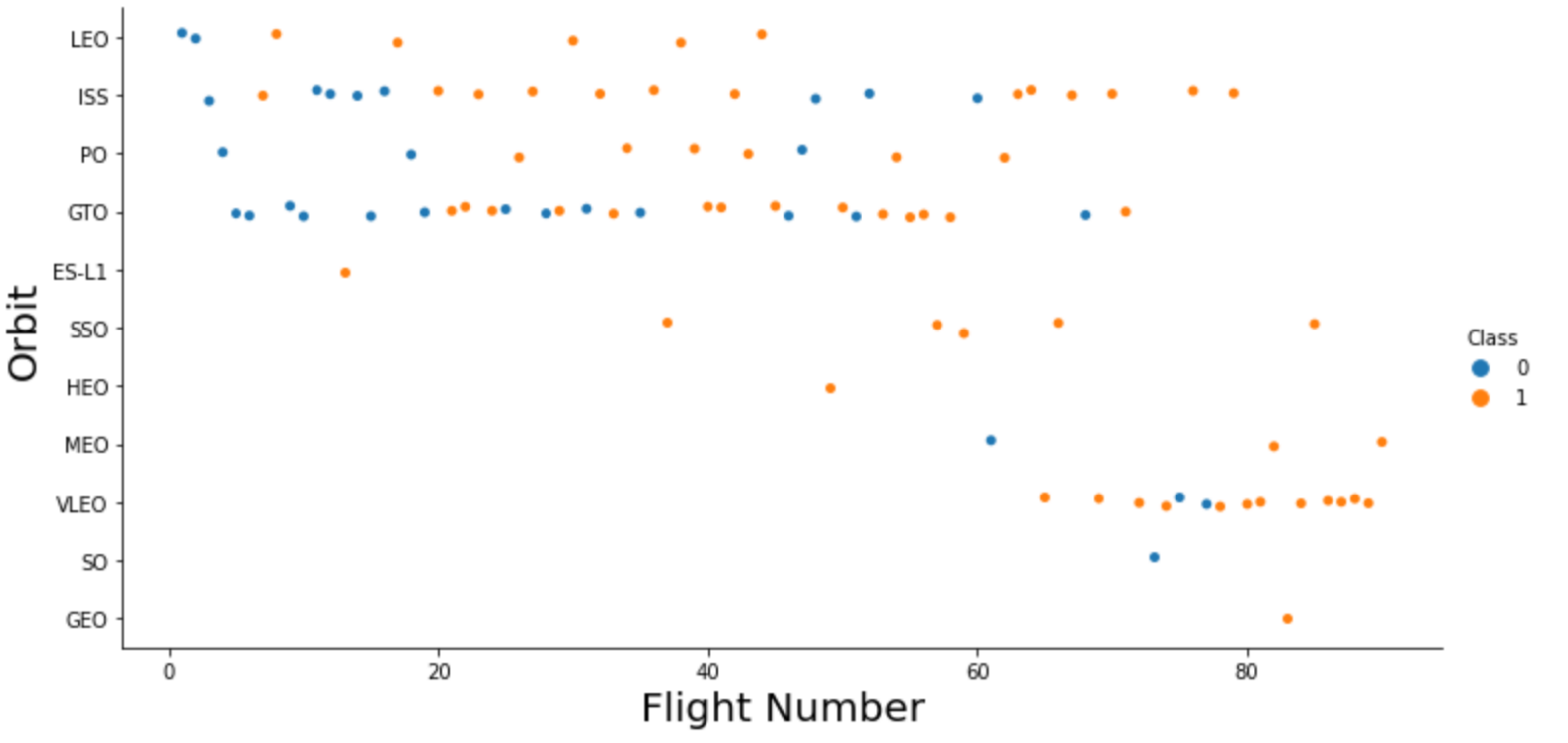
Success Rate vs. Orbit Type



- Orbit types ES-L1, GEO, HEO and SSO has a success rate of 100 %. However, these orbit types only have a total launch attempt of 8.

- Top 3 orbit types in terms of total number of attempts GTO (27 attempts), ISS (21 attempts) and VLEO (14 attempts) have success rates of 52%, 62% and 86% respectively.

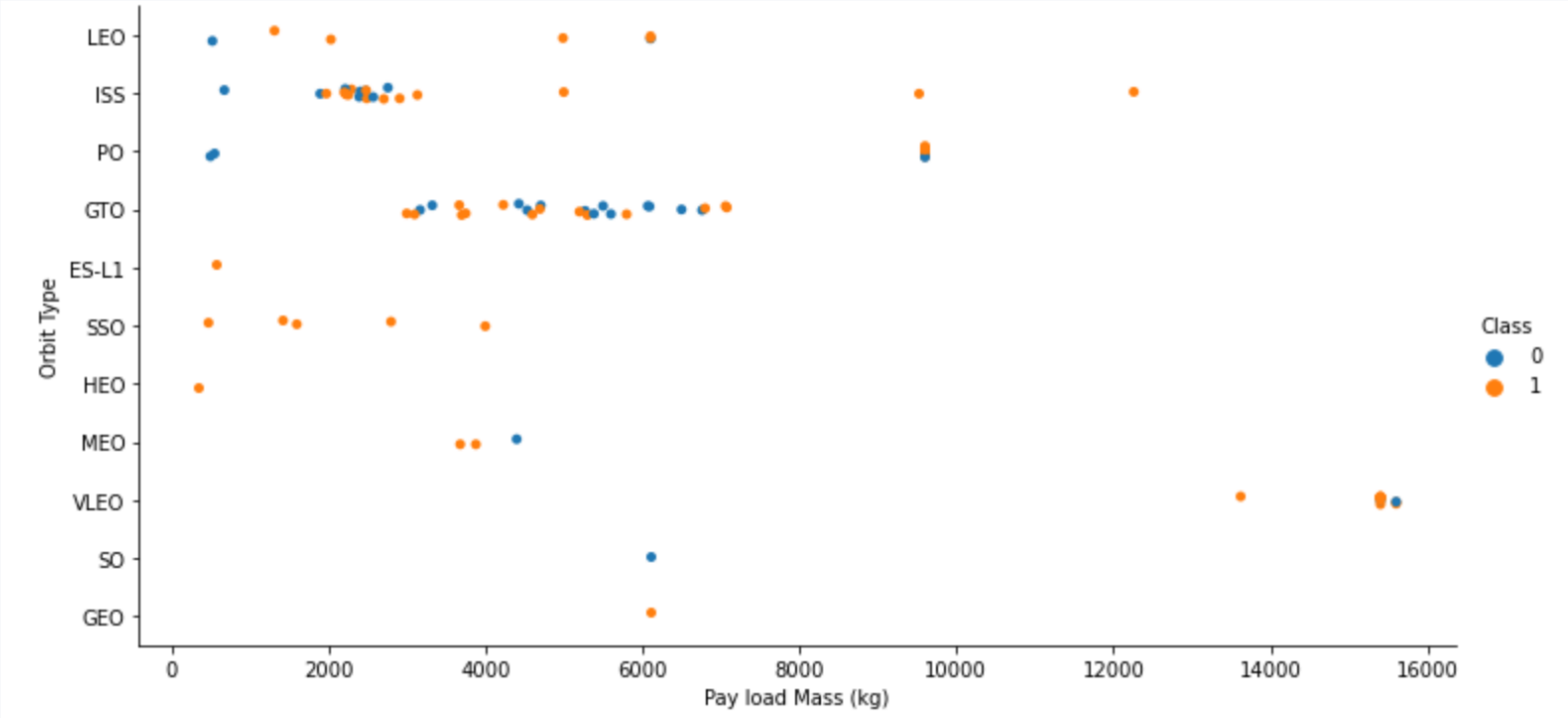
Flight Number vs. Orbit Type



- There seems to be a positive relation with landing outcome and flight number (As the flight number increases so as the possibility of a successful landing)

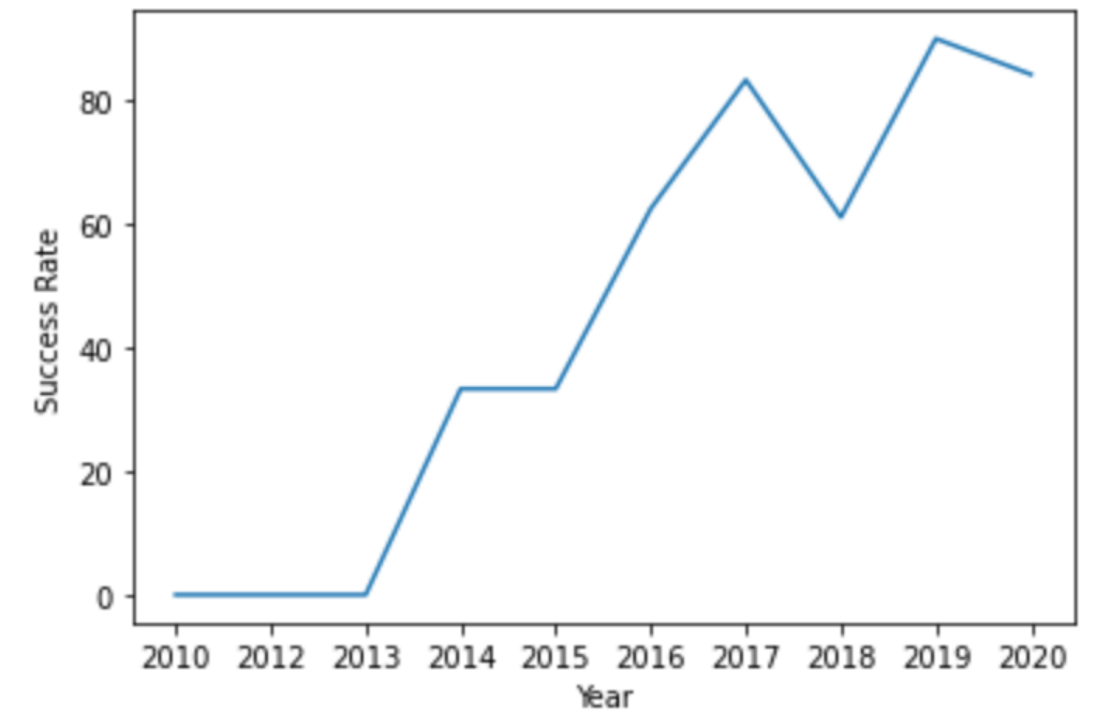
- The relationship seems stronger for LEO and VLEO orbit types and lower for GTO orbit type.

Payload vs. Orbit Type



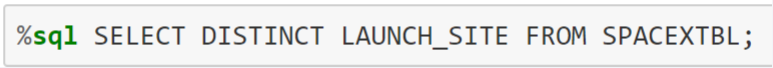
- For LEO and ISS orbit types successful landing rate seems to increase with heavier payloads.
- For GTO orbit type there is not a clear relationship between successful landing rate and payloads.
- For the other orbit types there isn’t sufficient data to comment on the correlation of successful landing rate and payload mass.

Launch Success Yearly Trend

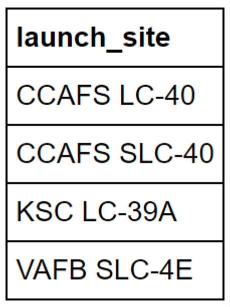


- Success rates for the first 3 years were 0%.
- From 2013 to 2017 there was a steady increase in success rate.
- There was a slight decrease in 2018 however success rates for the last 2 years are above 80%.

All Launch Site Names



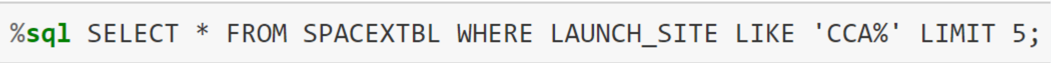
```
%sql SELECT DISTINCT LAUNCH_SITE FROM SPACEXTBL;
```



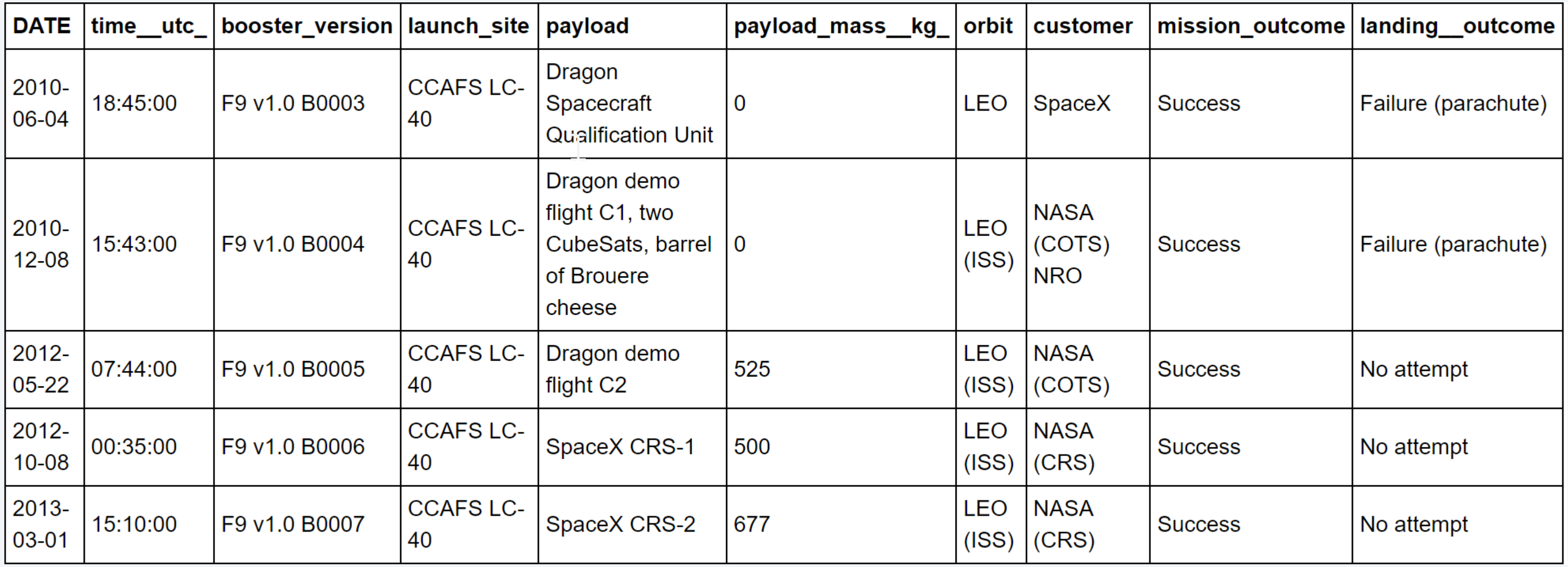
launch_site
CCAFS LC-40
CCAFS SLC-40
KSC LC-39A
VAFB SLC-4E

- There are 4 distinct launch sites.
- Using DISTINCT clause in the query ensures only unique values in the launch_site column from SPACEXTBL table are displayed.

Launch Site Names Begin with 'CCA'



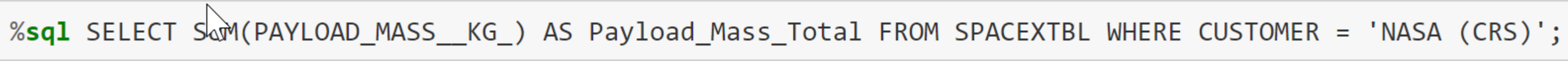
```
%sql SELECT * FROM SPACEXTBL WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5;
```



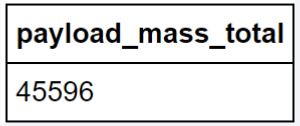
DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight 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

- LIMIT 5 clause is used to select only 5 records from SPACEXTBL
- LIKE operator is used to search for a specified pattern, % wildcard represents zero, one or multiple characters. LIKE ‘CCA%’ clause is used to select all launch sites starting with CCA.

Total Payload Mass



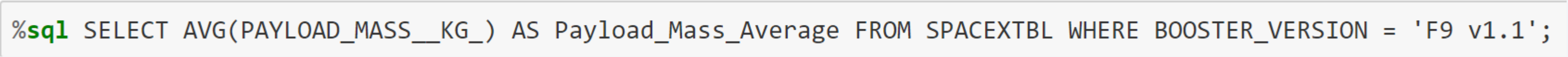
```
%sql SELECT SUM(PAYLOAD_MASS__KG_) AS Payload_Mass_Total FROM SPACEXTBL WHERE CUSTOMER = 'NASA (CRS)';
```



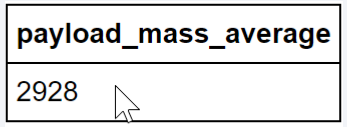
payload_mass_total
45596

- SUM function calculates the sum of values in the column PAYLOAD_MASS__KG_
- WHERE clause filters the dataset to perform calculations only if the value in customer column is NASA (CRS)

Average Payload Mass by F9 v1.1



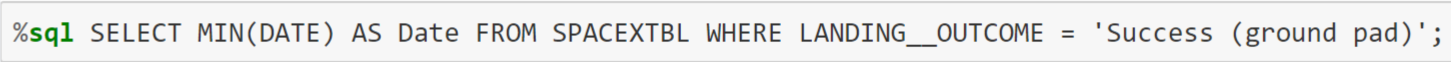
```
%sql SELECT AVG(PAYLOAD_MASS__KG_) AS Payload_Mass_Average FROM SPACEXTBL WHERE BOOSTER_VERSION = 'F9 v1.1';
```



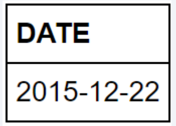
payload_mass_average
2928

- AVERAGE function calculates the average value of the column PAYLOAD_MASS__KG_
- WHERE clause filters the dataset to perform calculations only if the value in BOOSTER_VERSION column is F9 v1.1

First Successful Ground Landing Date



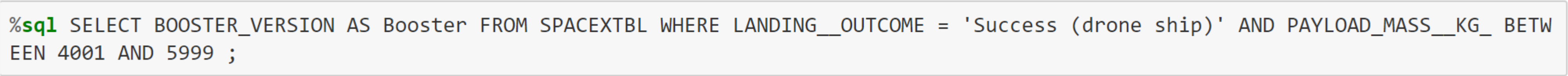
```
%sql SELECT MIN(DATE) AS Date FROM SPACEXTBL WHERE LANDING__OUTCOME = 'Success (ground pad)';
```



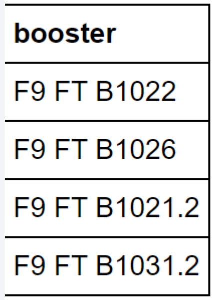
DATE
2015-12-22

- MIN function brings the earliest date in DATE column
- WHERE clause filters the dataset to perform search only if the value in LANDING_OUTCOME column is Success (ground pad)

Successful Drone Ship Landing with Payload between 4000 and 6000



```
%sql SELECT BOOSTER_VERSION AS Booster FROM SPACEXTBL WHERE LANDING__OUTCOME = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ BETW
EEN 4001 AND 5999 ;
```

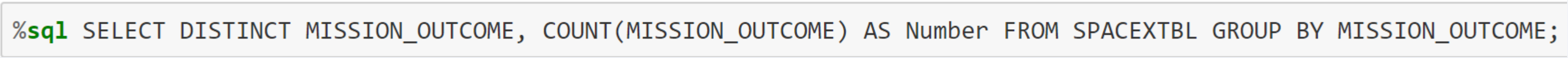


booster
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
F9 FT B1031.2

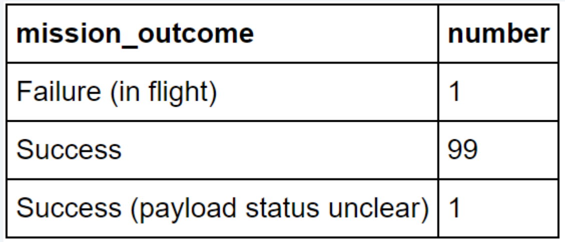
- Using an AND operator inside a WHERE clause, dataset is filtered to perform search only if the value in LANDING_OUTCOME column is Success (drone ship) and value in PAYLOAD_MASS__KG_ column is greater than 4.000 and less than 6.000

- BETWEEN operator selects all values in column PAYLOAD_MASS__KG_ in 4.001 – 5.999 range. Begin and end values are included.

Total Number of Successful and Failure Mission Outcomes



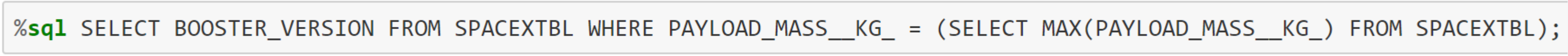
```
%sql SELECT DISTINCT MISSION_OUTCOME, COUNT(MISSION_OUTCOME) AS Number FROM SPACEXTBL GROUP BY MISSION_OUTCOME;
```



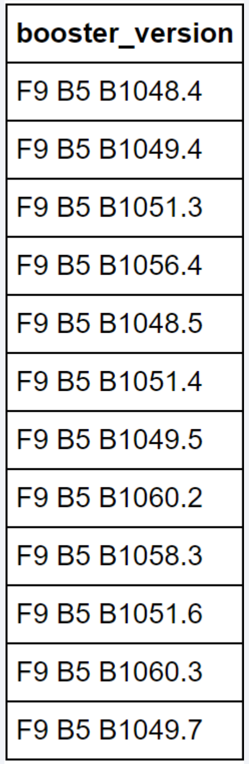
mission_outcome	number
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

- COUNT function counts the total number of values in MISSION_OUTCOME column
- GROUPBY statement groups rows that have the same values into summary rows

Boosters Carried Maximum Payload



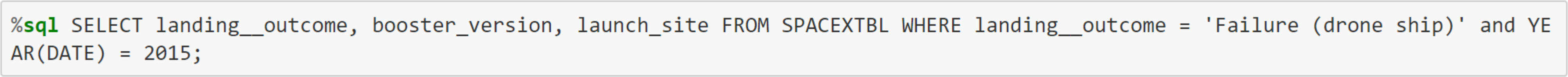
```
%sql SELECT BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL);
```



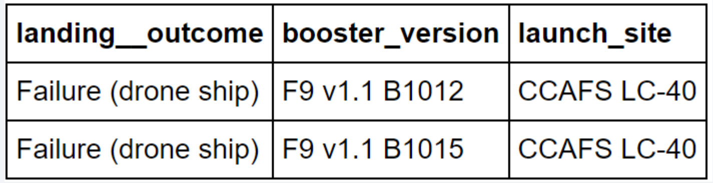
booster_version
F9 B5 B1048.4
F9 B5 B1049.4
F9 B5 B1051.3
F9 B5 B1056.4
F9 B5 B1048.5
F9 B5 B1051.4
F9 B5 B1049.5
F9 B5 B1060.2
F9 B5 B1058.3
F9 B5 B1051.6
F9 B5 B1060.3
F9 B5 B1049.7

- Using a subquery first MAX function filters the maximum value in PAYLOAD_MASS__KG_ column
- Then WHERE clause filters the dataset to perform search only if the value in PAYLOAD_MASS__KG_ column is equal to maximum value

2015 Launch Records



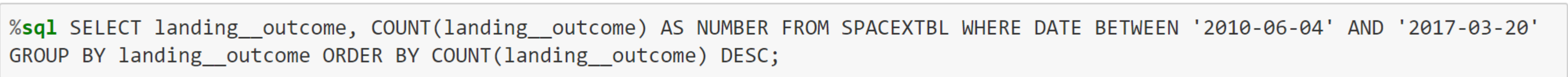
```
%sql SELECT landing__outcome, booster_version, launch_site FROM SPACEXTBL WHERE landing__outcome = 'Failure (drone ship)' and YE
AR(DATE) = 2015;
```



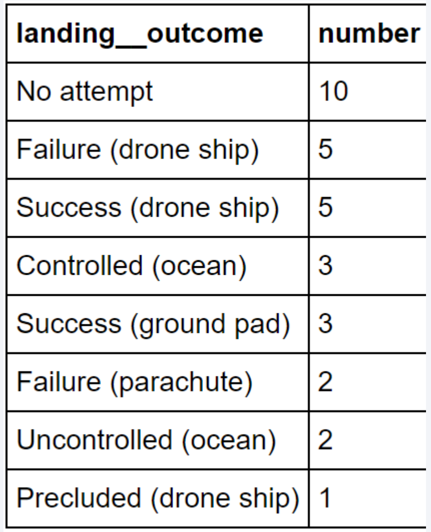
landing__outcome	booster_version	launch_site
Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

- Using an AND operator inside a WHERE clause, dataset is filtered to perform search only if the value in LANDING_OUTCOME column is Failure (drone ship) and year in DATE column is 2015
- YEAR function returns the year part of a date

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



```
%sql SELECT landing__outcome, COUNT(landing__outcome) AS NUMBER FROM SPACEXTBL WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY landing__outcome ORDER BY COUNT(landing__outcome) DESC;
```



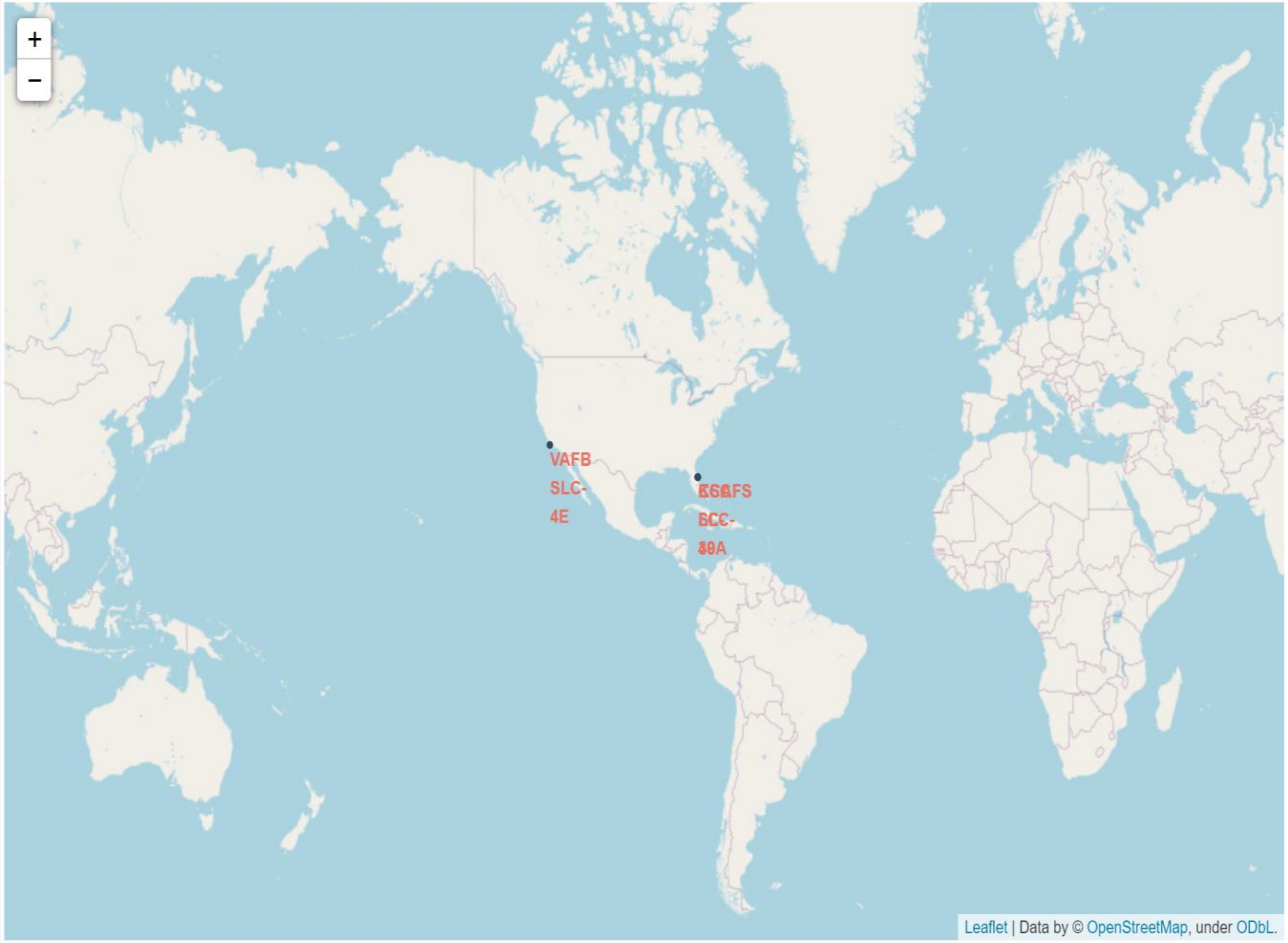
landing__outcome	number
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

- Using an AND and BETWEEN operator inside a WHERE clause, dataset is filtered to perform search only if the value in DATE column is between 2010-06-04 and 2017-03-20
- GROUPBY statement groups rows that have the same values into summary rows
- COUNT function counts the total number of values in landing_outcome column
- ORDER BY is used to sort the result set by total number of landing and DESC word sorts the result set in descending order

Section 3

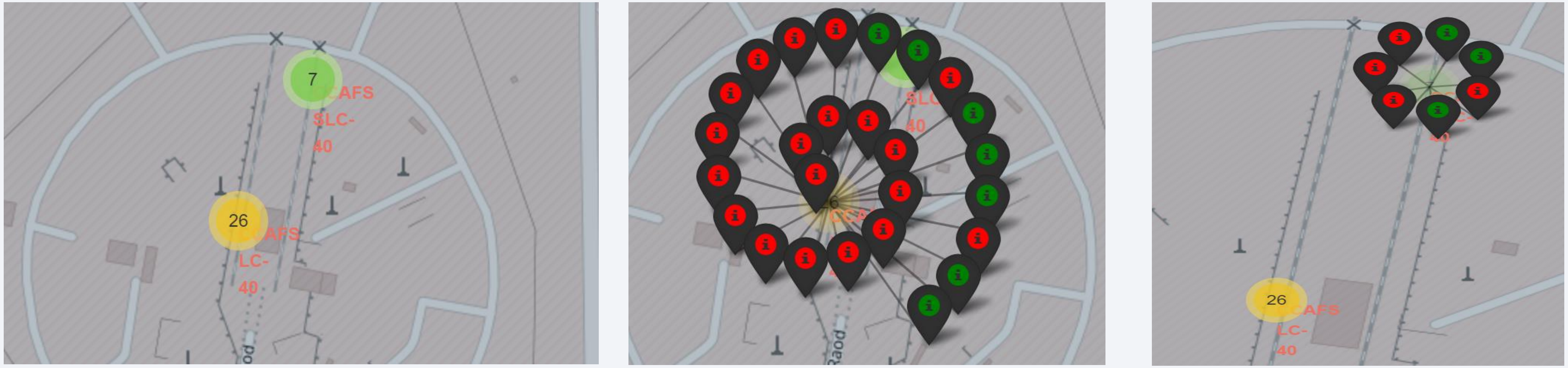
Launch Sites Proximities Analysis

All Launch Sites Location



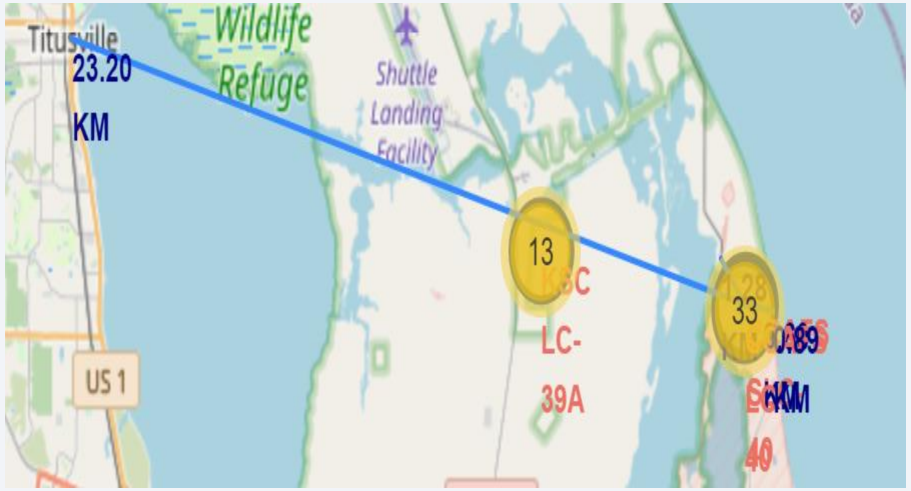
- All launch sites are in the USA and near coastline. (One of them near North Atlantic the others near North Pacific coastline)

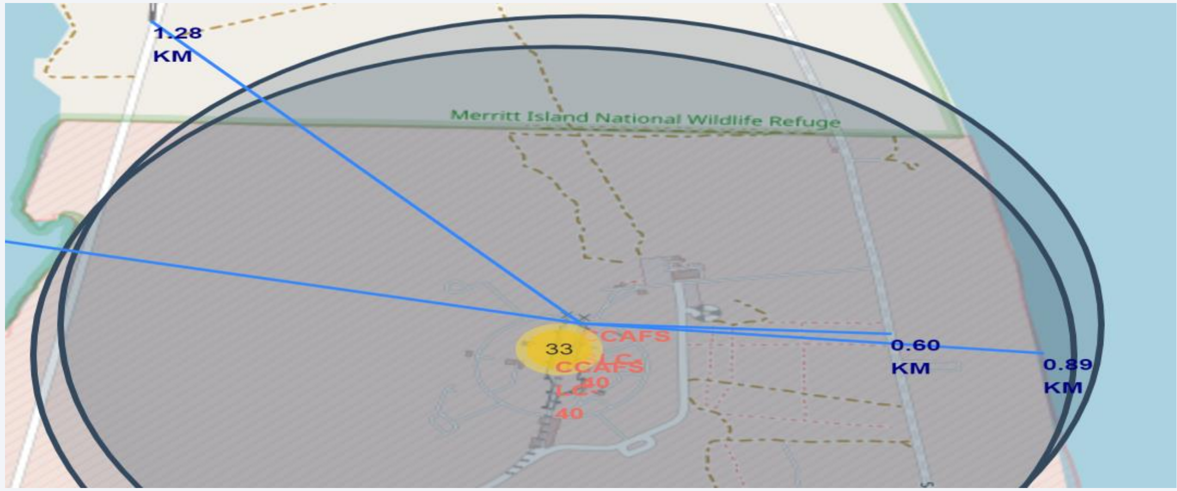
Color Labeled Launch Outcomes



- Clicking on the marker clusters, landing outcomes can be displayed. Green markers display successful landings and red markers display failed ones.

Launch Site Distance to Proximities

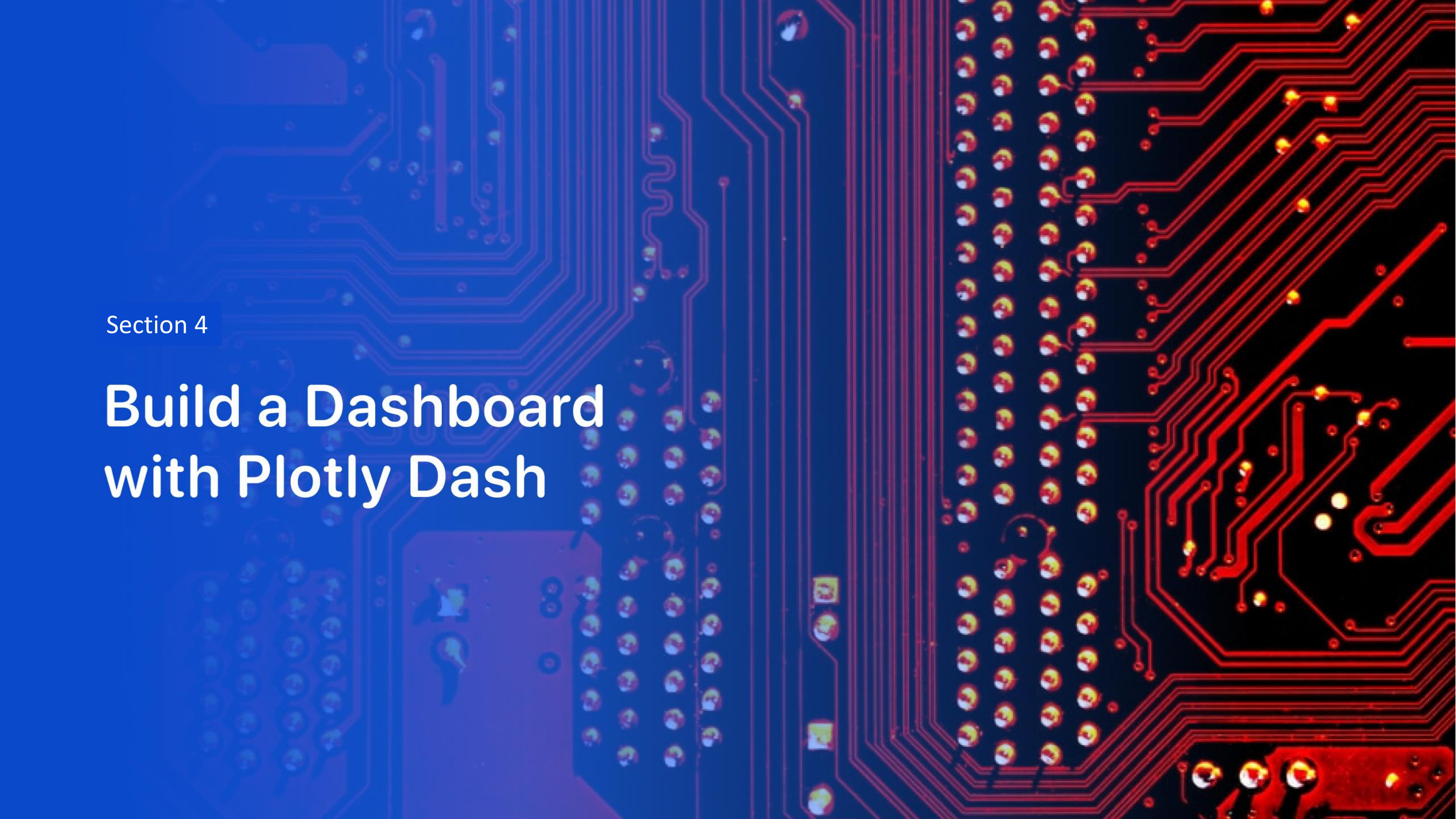




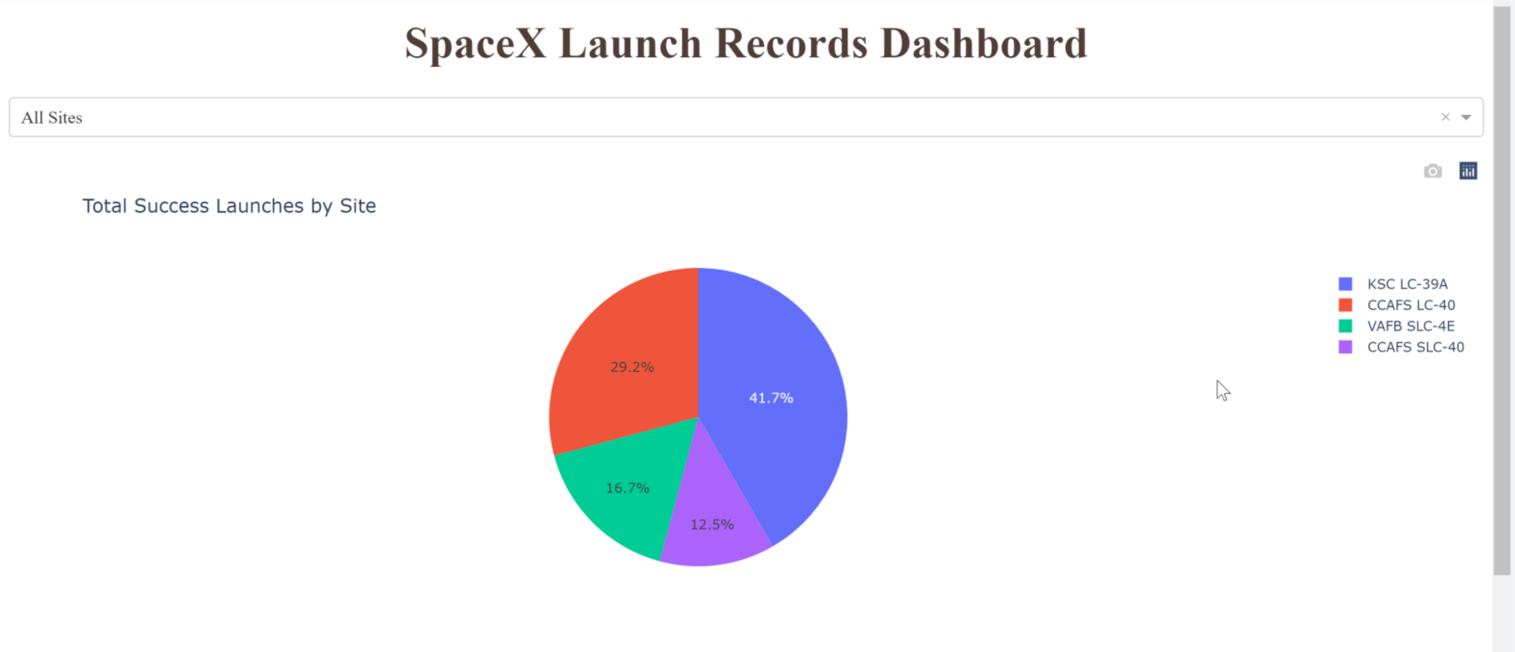
- While the launch sites are pretty close to railways, highways and coastline for logistic purposes, they are relatively far away from cities in order to ensure public safety.

Section 4

Build a Dashboard with Plotly Dash



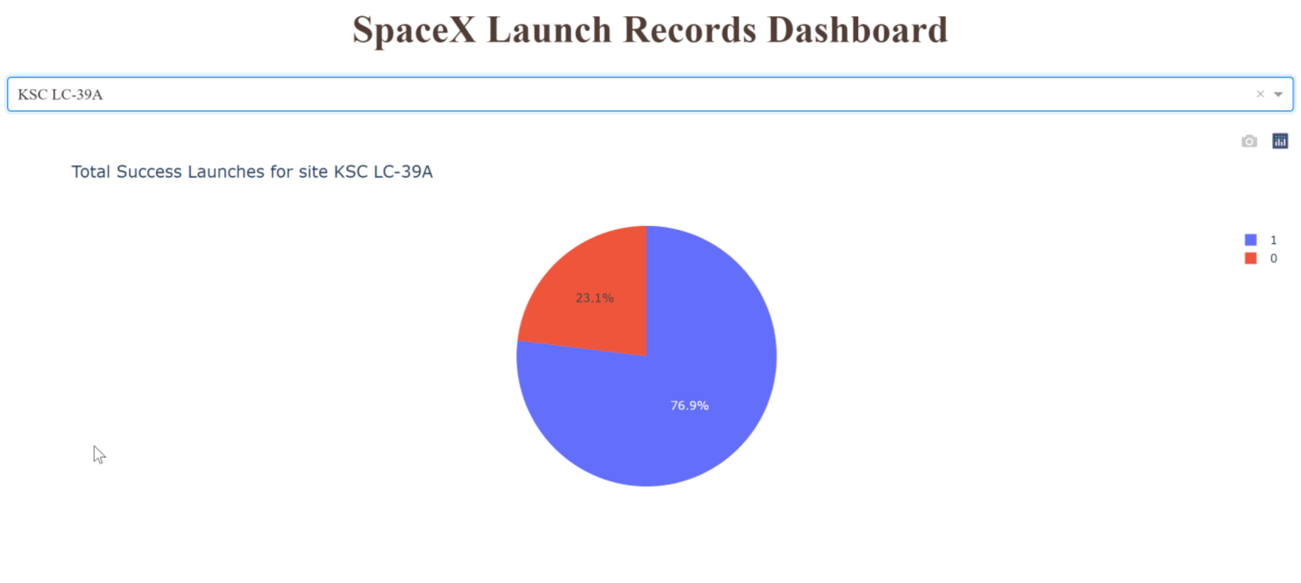
Successful Launches for All Sites



- KSC LC-39A site has the most number of successful launches and CCAFS SLC-40 has the fewest.

- Although site CCAFS LC-40 has the 2nd most number of successful launches, it has a success rate of only 27% which is lower from both VAFB SLC-4E and CCAFS SLC-40.

Launch Site with the Highest Launch Success Ratio



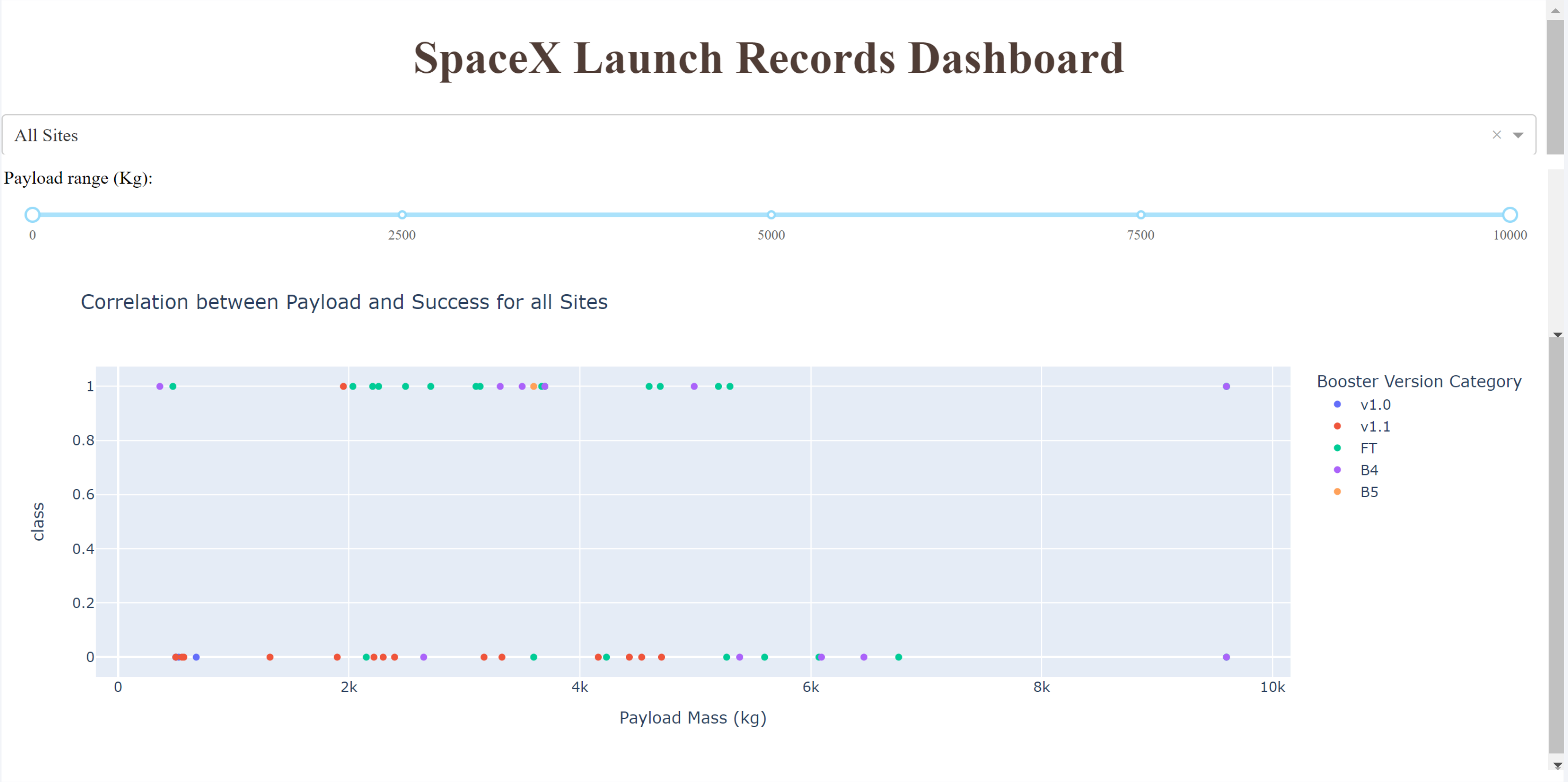
SpaceX Launch Records Dashboard

KSC LC-39A

Total Success Launches for site KSC LC-39A

- KSC LC-39A has the highest launch success ratio with 10 successful and 3 failed landings.

Payload vs Launch Outcome Scatter Plot for All Sites

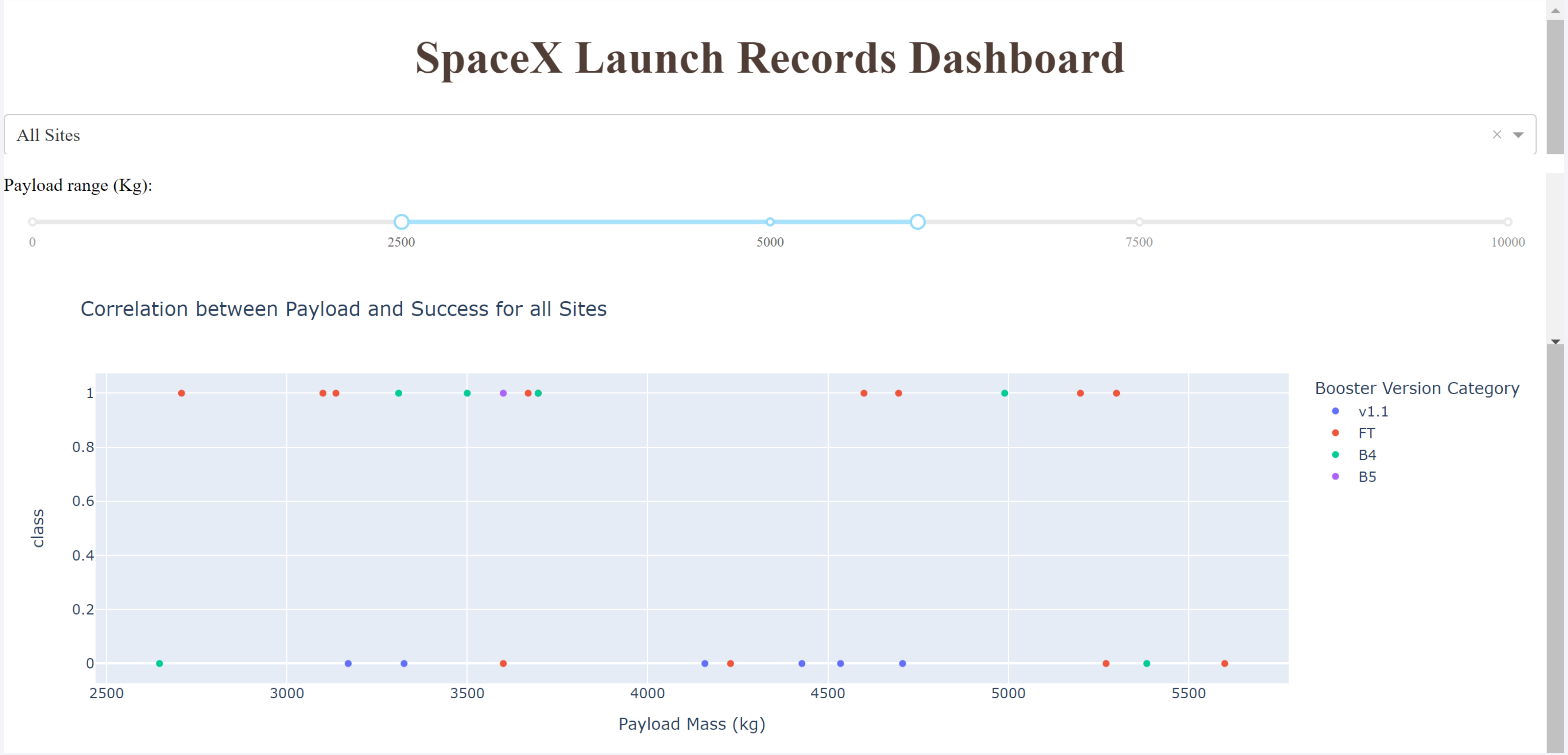


SpaceX Launch Records Dashboard

All Sites

Payload range (Kg):

Payload vs Launch Outcome Scatter Plot for All Sites



SpaceX Launch Records Dashboard

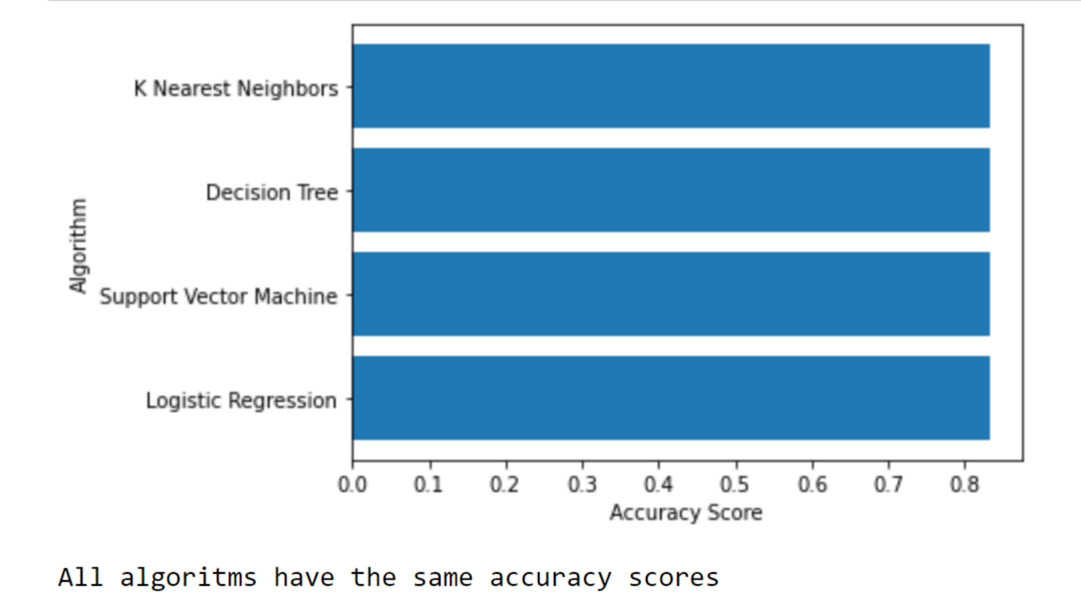
Payload vs Launch Outcome Scatter Plot for All Sites

- FT Booster Version has the largest success rate while almost all of launches with V1.0 and V1.1Booster Version failed.
- 3.000 – 4.000 KG payload range seems to have the largest success rate.
- Within 6.000 – 8.000 KG payload range all launches have failed.
- 0 – 2.000 KG payload range also seems to have a pretty low success rate.

Section 5

Predictive Analysis (Classification)

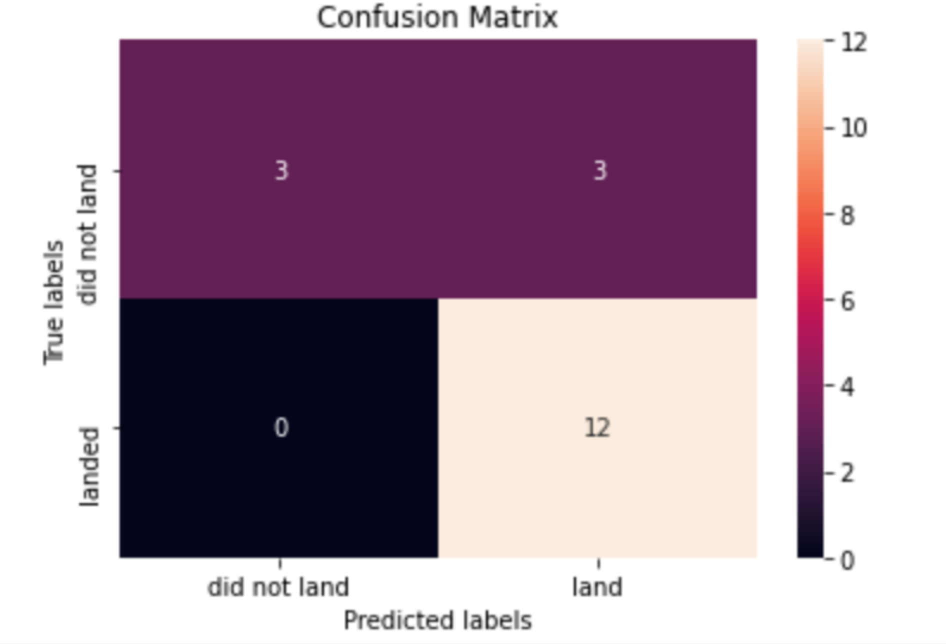
Classification Accuracy



```
All algoritms have the same accuracy scores
```

- The accuracy scores of all the models were pretty close on training data and all models produced an accuracy score of 83,33% on test data.

Confusion Matrix



- Confusion matrices of all the models were the same.
- Confusion matrix displays that the problem with the model was false positive classifications. 3 launches were classified as successful landings when in reality they were failed landings.
- However considering the fact that the model classified 15 out of 18 landing outcomes correctly, overall the accuracy of the model is satisfiying.

Conclusions

- As the number of flights increased so as the success rate. The average success rate for the last two years was over 86%.
- Orbit types ES-L1, GEO, HEO and SSO has a success rate of 100 %. However, these orbit types only have a total launch attempt of 8. Top 3 orbit types in terms of total number of attempts GTO (27 attempts), ISS (21 attempts) and VLEO (14 attempts) have success rates of 52%, 62% and 86% respectively.
- KSC LC-39A has the highest launch success ratio. However, as mentioned above as the number of flights increased so did the success rate. The other launch sites have pretty high success rates for the last two years as well, thus number of flights seems to have a higher affect on launch outcome than launch site.
- The launch success rate of heavy payloads are higher than lower weighted payloads but some of that success is also related with the increase in number of flights. (The earliest launch date with a payload over 9.000 KG is 14th January 2017)
- The payload range 4.000 – 8.000 KG seems to have an especially lower success rate with only 15 successful landings out of 29. Even after 2017 (where flight numbers increased and overall success rate is pretty high) the successful landings in this payload range is only 13 out of 21.

Conclusions

- The launch sites are close to railways, highways and coastline for logistic purposes but a reasonable distance away from the cities in order to ensure public safety and security.
- All the models trained with training dataset produced the same accuracy score (83,33%) for the test data. Although the model creates some false positive predictions (positive outcomes that the model predicts incorrectly), overall its predictions are satisfying.

Appendix

- Github repository

Thank you!