IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Akshata Surve
13th Feb, 2022

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 using Classification

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 are making space travel affordable for everyone. Perhaps the most successful of them is SpaceX, and one of the reasons is that their rocket launch is relatively expensive.

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, we will predict if the Falcon 9 first stage will land successfully. If we can determine if the first stage will land, we can determine the cost of launch. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch.

- **Problems you want to find answers**

- Correlation between each rocket variables and successful landing rate.

- Conditions to get best results & ensure the best successful landing rate.

Section 1

Methodology

Methodology

Executive Summary

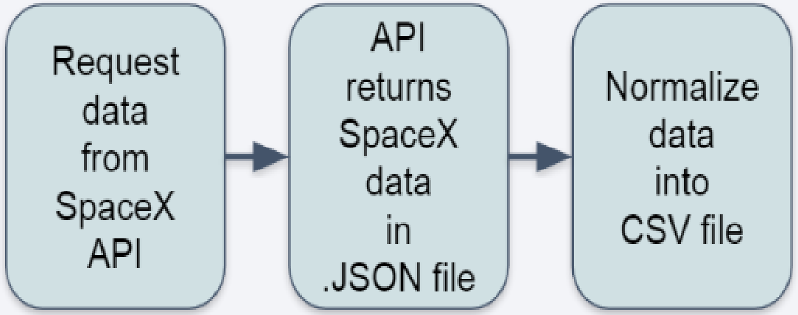
- Data collection methodology:
  - SpaceX Rest API
  - Web Scrapping from Wikipedia
- Perform data wrangling
  - Converted outcomes into training labels with booster successfully/unsuccessfully landed
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Find best hyperparameter for SVM, Classification Trees and Logistic Regression

Data Collection

GITHUB URL

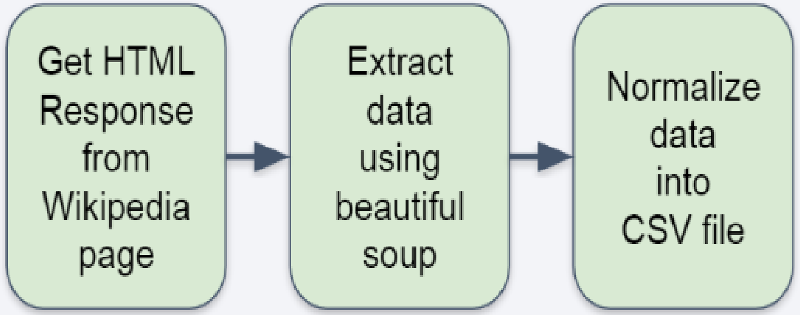
- Data collection process includes a combination of API requests from the SpaceX API & web scraping data from a table in wikipedia page of SpaceX, Falcon 9 & Falcon Heavy Launches records.
- **Columns derived from API:** FlightNumber, Date, BoosterVersion, PayloadMass, Orbit, LaunchSite, Outcome, Flights, GridFins, Reused, Legs, LandingPad, Block, ReusedCount, Serial, Longitude, Latitude
- **Flow Chart for Data Collection process :**

SpaceX API



Request data from SpaceX API → API returns SpaceX data in .JSON file → Normalize data into CSV file

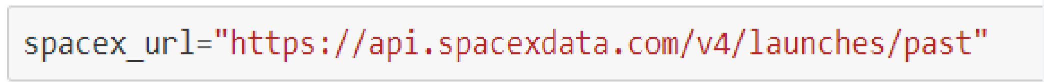
Web Scraping



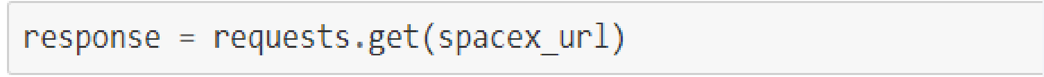
Get HTML Response from Wikipedia page → Extract data using beautiful soup → Normalize data into CSV file

Data Collection – SpaceX API

1. Requesting rocket launch data from SpaceX API

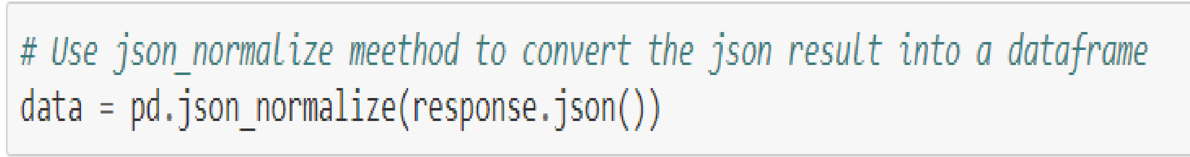


```
spacex_url="https://api.spacexdata.com/v4/launches/past"
```



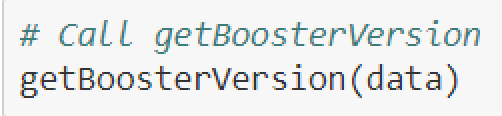
```
response = requests.get(spacex_url)
```

2. Converting response to a JSON file

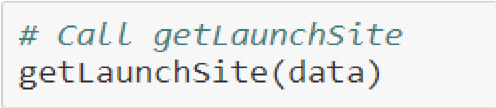


```
# Use json_normalize meethod to convert the json result into a dataframe
data = pd.json_normalize(response.json())
```

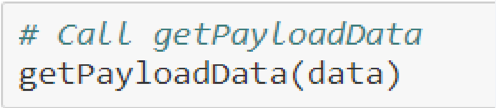
3. Using custom functions to clean data



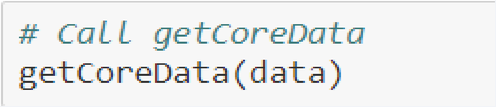
```
# Call getBoosterVersion
getBoosterVersion(data)
```



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

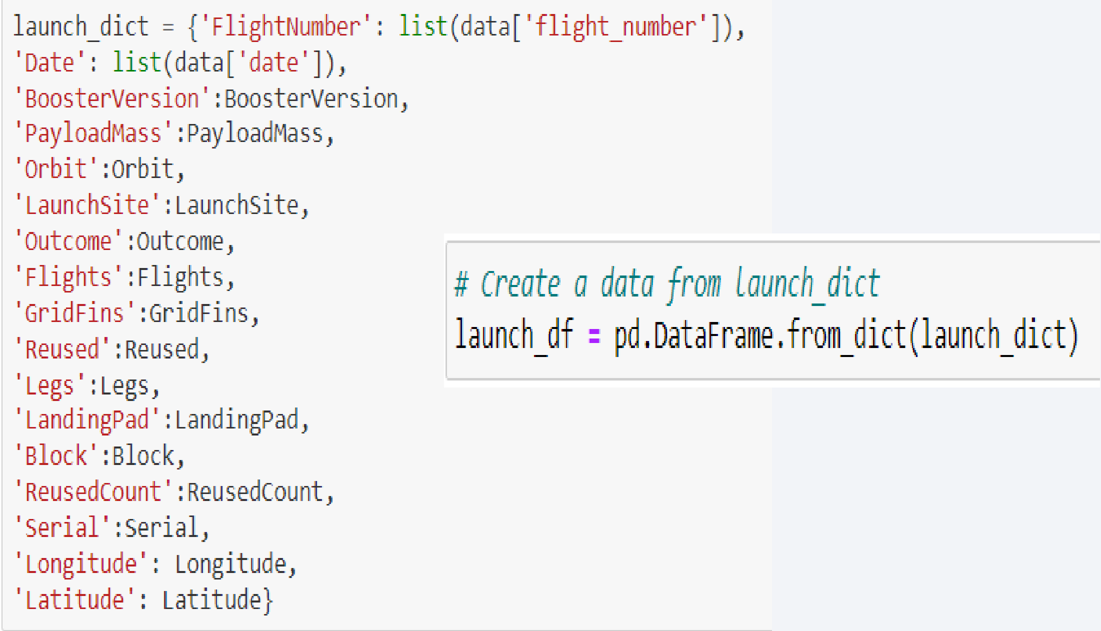


```
# Call getPayloadData
getPayloadData(data)
```



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

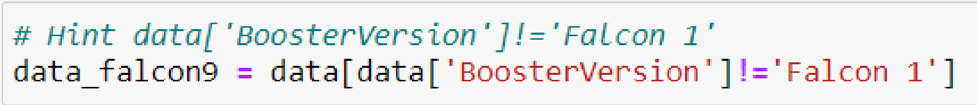
4. Combining the columns into dictionary to create a data frame



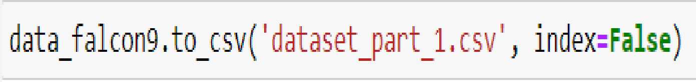
```
launch_dict = {'FlightNumber': list(data['flight_number']),
'Date': list(data['date']),
'BoosterVersion':BoosterVersion,
'PayloadMass':PayloadMass,
'Orbit':Orbit,
'LaunchSite':LaunchSite,
'Outcome':Outcome,
'Flights':Flights,
'GridFins':GridFins,
'Reused':Reused,
'Legs':Legs,
'LandingPad':LandingPad,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

```
# Create a data from launch_dict
launch_df = pd.DataFrame.from_dict(launch_dict)
```

5. Filtering data frame & exporting to CSV



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

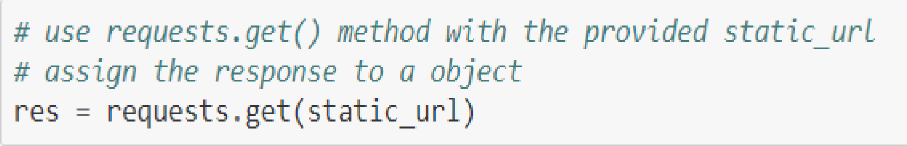


```
data_falcon9.to_csv('dataset_part_1.csv', index=False)
```

Data Collection - Scraping

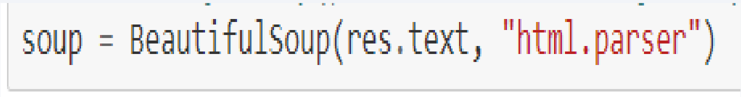
GITHUB URL

1. Getting response from HTML



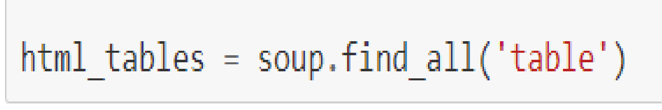
```
# use requests.get() method with the provided static_url
# assign the response to a object
res = requests.get(static_url)
```

2. Creating a BeautifulSoup object



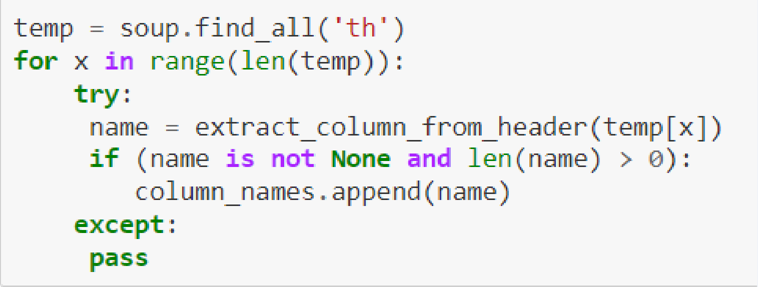
```
soup = BeautifulSoup(res.text, "html.parser")
```

3. Find all tables & assigning results to list



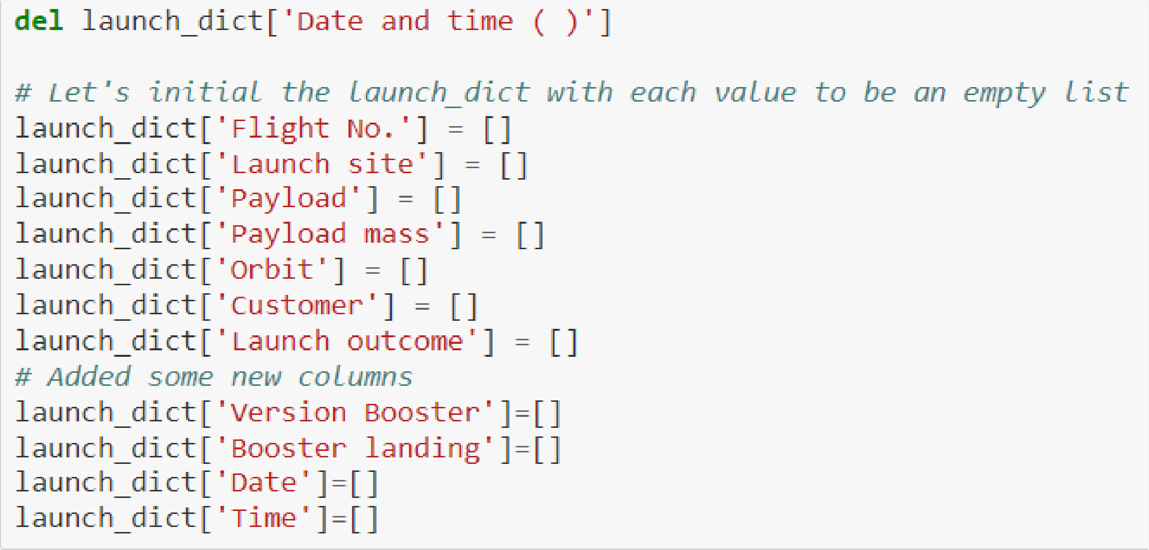
```
html_tables = soup.find_all('table')
```

4. Extracting column name one by one



```
temp = soup.find_all('th')
for x in range(len(temp)):
    try:
     name = extract_column_from_header(temp[x])
     if (name is not None and len(name) > 0):
        column_names.append(name)
    except:
     pass
```

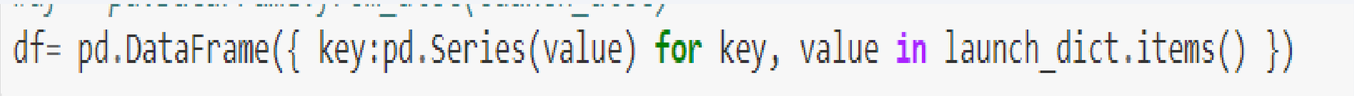
5. Creating empty dictionary with keys



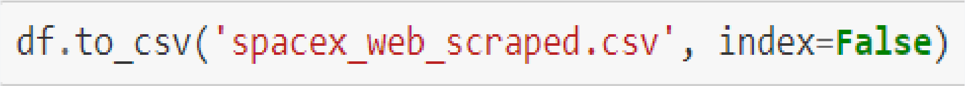
```
del launch_dict['Date and time ( )']

# 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. Filling up the launch dict with launch records

7. Creating dataframe and exporting it to csv



```
df= pd.DataFrame({ key:pd.Series(value) for key, value in launch_dict.items() })
```



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

Data Wrangling

GITHUB URL

- There are several cases in which booster failed to successfully land on the dataset, & sometimes it attempted to land but failed because of accident.

- True Ocean : the mission result has successfully landed in specific area of ocean

- False Ocean : the mission result has not successfully landed in specific area of ocean

- True RTLS : the mission result successfully landed on the ground pad

- False RTLS : the mission result has not successfully landed on the ground pad

- True ASDS : the mission result has successfully landed on the drone ship

- False ASDS : the mission result has successfully landed on the drone ship

- Convertng these results into training labels :

- 1= successful/ 0= failure

EDA with Data Visualization

- **Scatter chart**

- Flight number vs Launch site

- Payload vs Launch site

- Flight number vs Orbit type

- Payload vs Orbit type

- A scatter plot shows how much one variable is affected by another. The relationship between two variables is called correlation. This plot is generally composed of large data bodies.

- **Bar chart**

- Orbit type vs Success rate

- A bar chart makes it easy to compare datasets between multiple groups at glance. One axis represnets a category and other axis represents a discrete value. The purpose of this chart is to indicate the relationship between two axes.

- **Line chart**

- Year vs Success rate

- A line chart shows data variables & trends very clearly & helps to predict results of data that has not been recorded

EDA with SQL

- Loading the dataset into the corresponding table in a Db2 database, & executing the SQL queries to answer following questions :

- Displaying 5 records where launch sites begin with string 'CCA'.

- Displaying names of unique launch sites in the space mission.

- Displaying the total payload mass carried by boosters launched by NASA(CRS).

- Displaying avg 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 boosters which have success in drone ship & have payload mass > 4000 & < 6000.

- Listing the total number of successful & 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, & launch site names for year 2015.

- Ranking the count of landing outcomes.

Build an Interactive Map with Folium

GITHUB URL

- Objects created & added to a folium map:

- Markers that show the success/failed launches for each site on the map.

- Markers that show all launch sites on map.

-Lines that show the distances between launch site to its proximities.

- By adding these objects, following geographical patterns about launch sites are found.

- 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

Build a Dashboard with Plotly Dash

GITHUB URL

- The dashboard applications contains a pie chart & scatter point chart.

I. Pie Chart:

- For showing total success launches by sites

- This chart can be selected to indicate a successful landing distribution across all launch sites or to indicate the success rate of individual launch sites.

II. Scatter chart:

- For showing the relation between outcomes & payload mass(KG) by different boosters.

- Has 2 inputs: all sites/individual site& payload mass on slider between 0 & 10000 kg.

- This chart helps to determine how success depends on the launch point, payload mass, & booster version categories.

Predictive Analysis (Classification)

GITHUB URL

- Perform exploratory Data analysis & determine training labels.

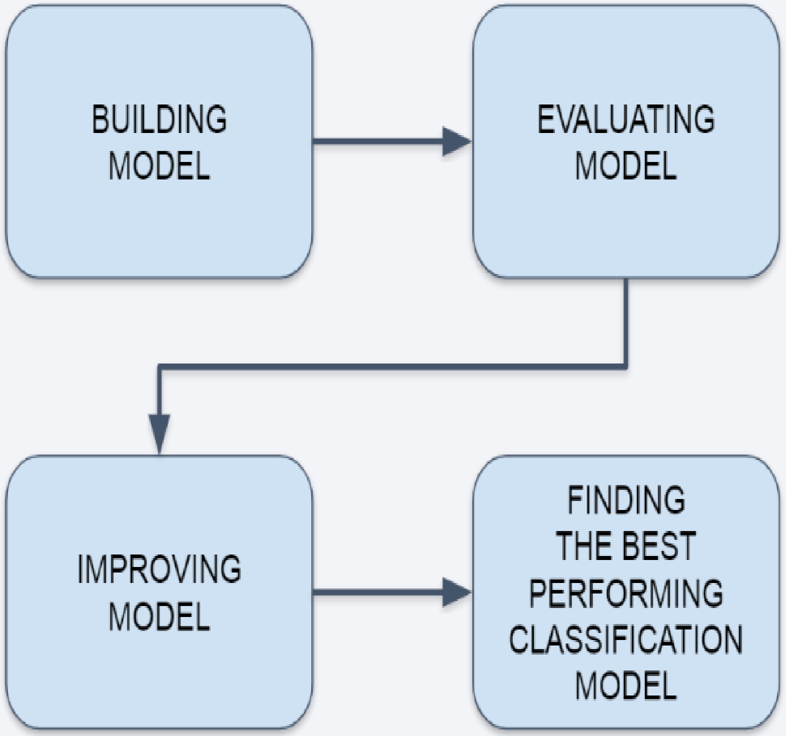
- Create a column for class

- Standardize the data

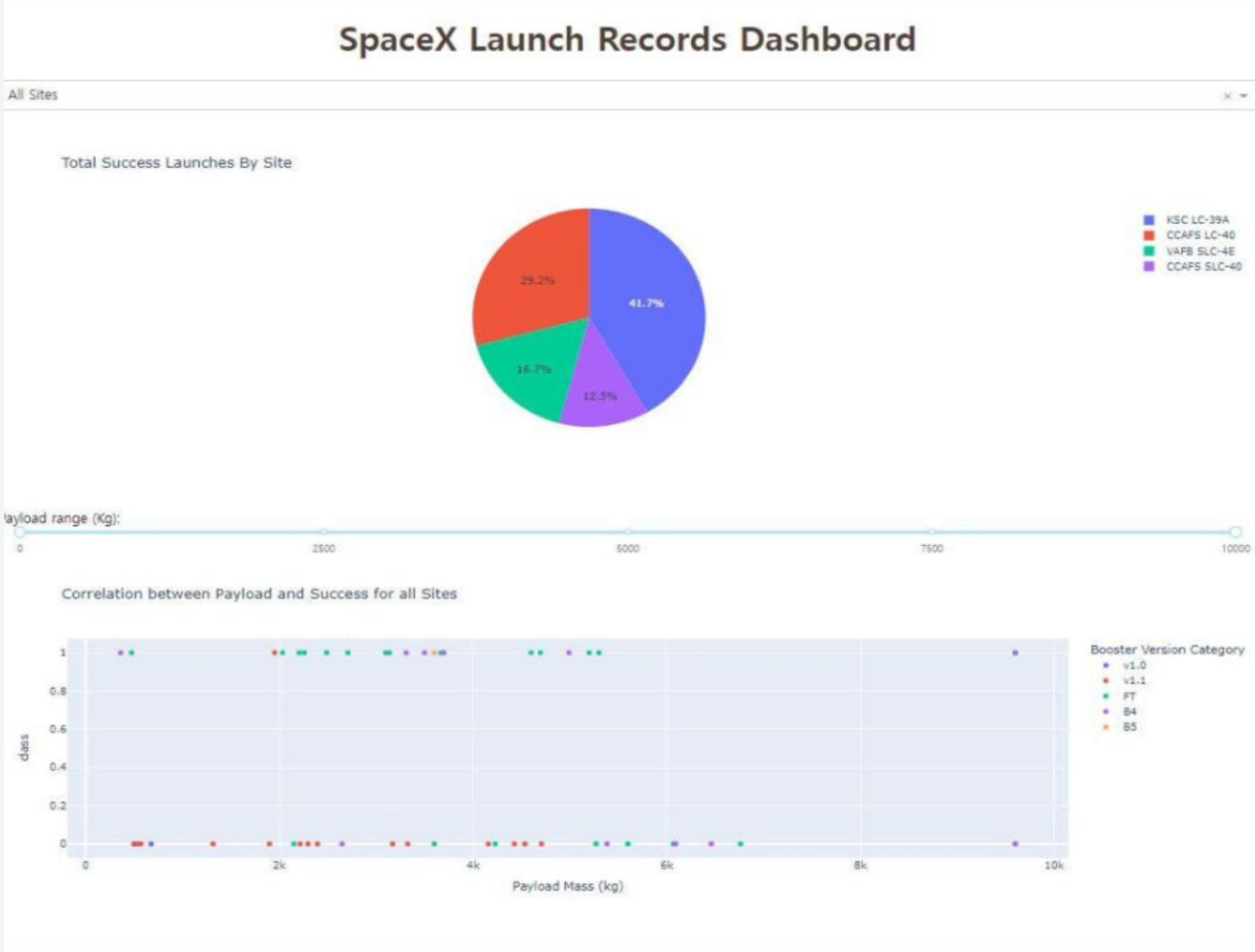
- Split data into training & test data

- Find best hyperparameter for SVM, Classification trees & logistic regression

- Find the method performs best using test data



Results



- Th left screenshot is a preview of the dashboard with Plotly Dash.

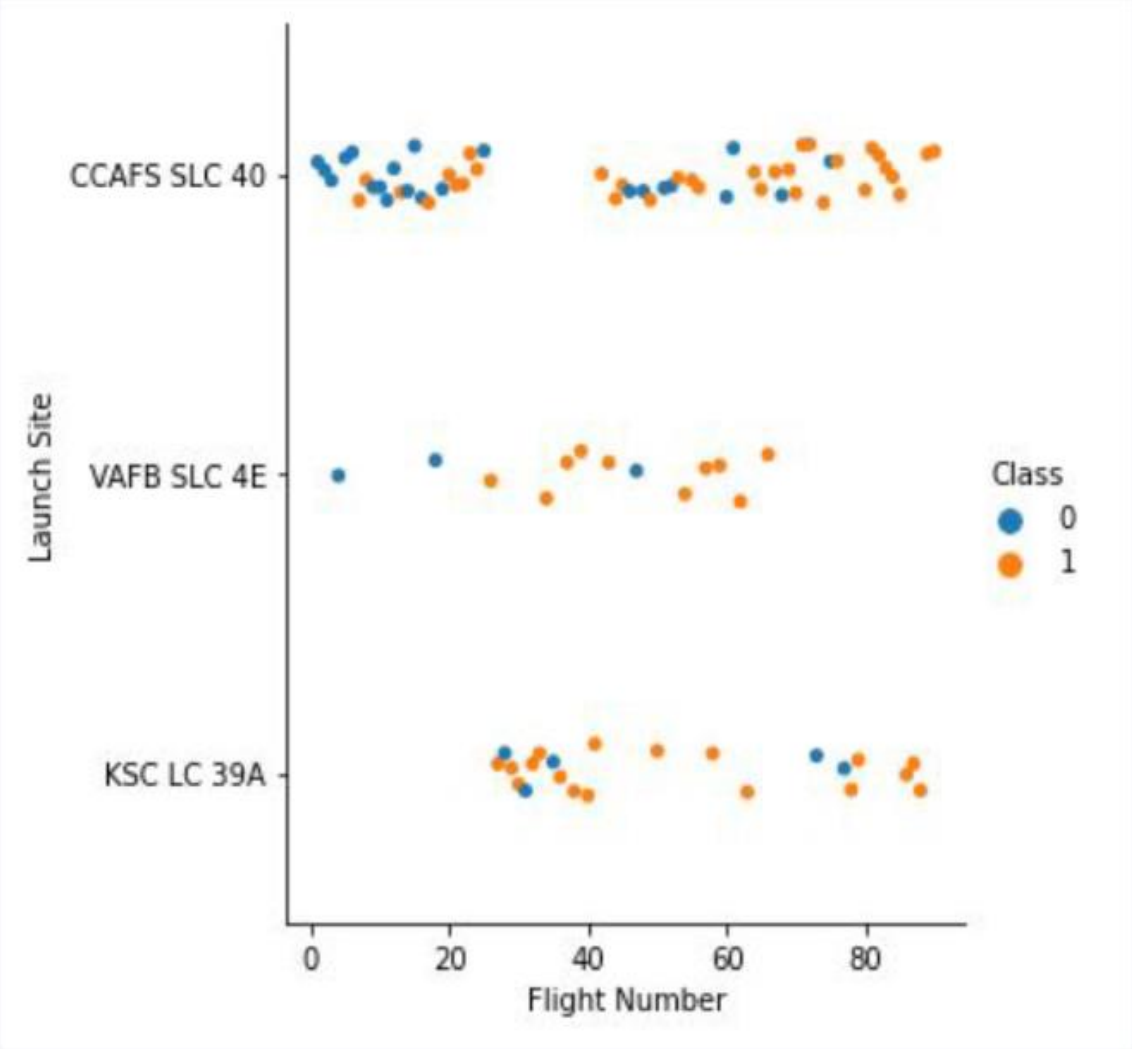
- The results of EDA with visualization, EDA with SQL, interactive map with Folium, & interactive dashboard will be shown in next slides.

- Comparing the accuracy of the four methods, all return the same accuracy of about 83% for test data.

Section 2

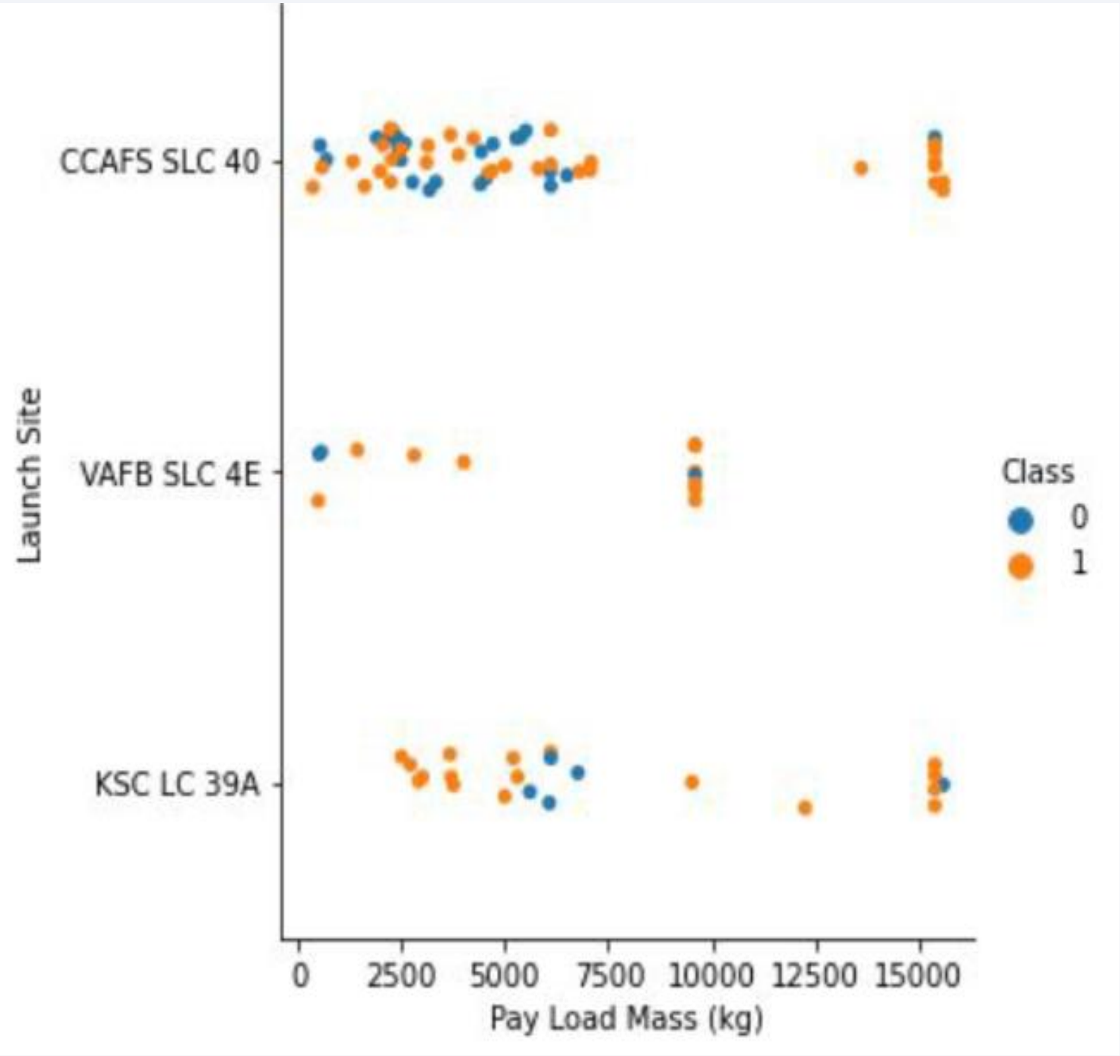
Insights drawn from EDA

Flight Number vs. Launch Site



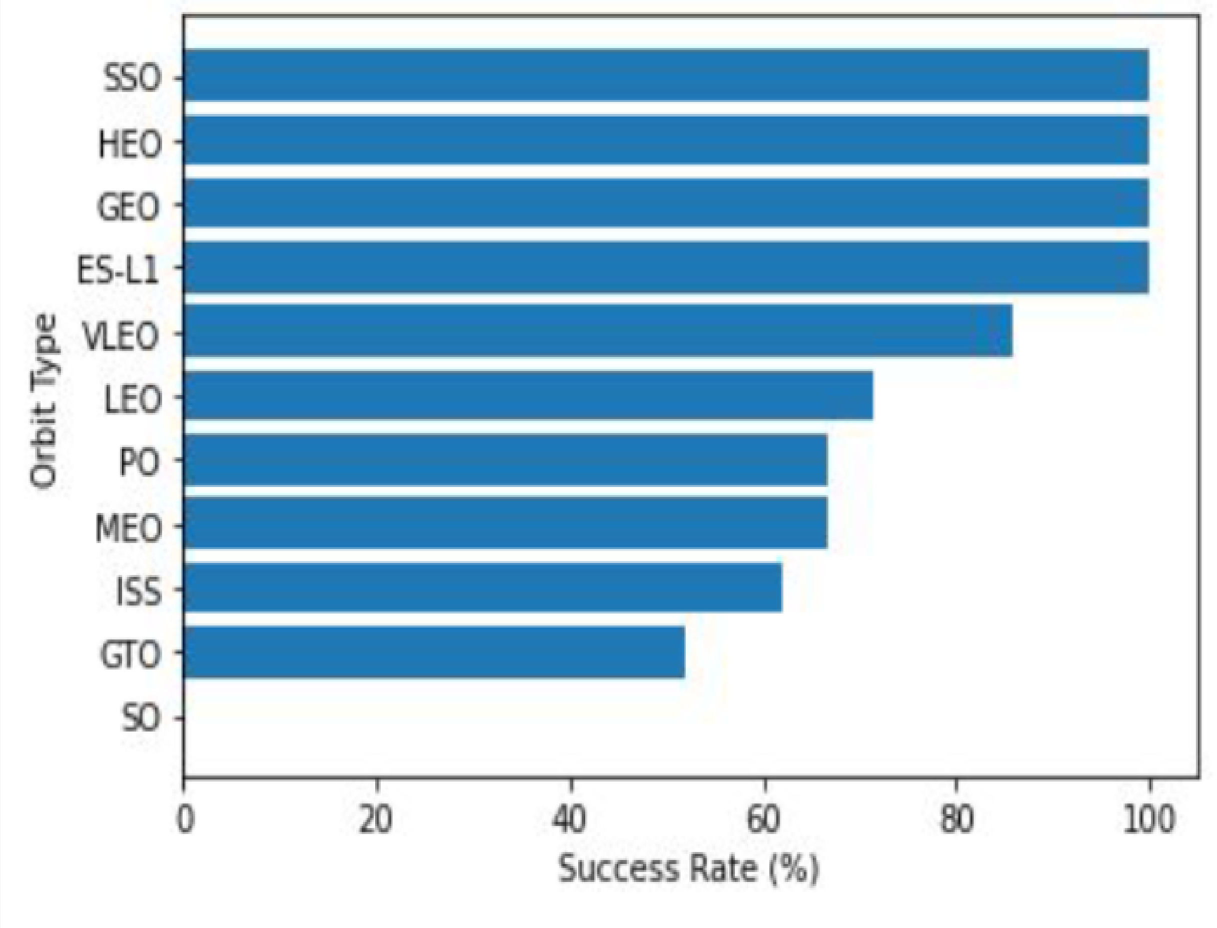
- Class 0(blue) represents unsuccessful launch, and class 1(orange) represents successful launch.
- This figure shows that the success rate increased as the number of flights increased.
- As the success rate has increased, considerably since 20th flight, this point seems to be a big breakthrough.

Payload vs. Launch Site



- Class 0(blue) represents unsuccessful launch, and class 1(orange) represents successful launch.
- At first glance, the larger payload mass, the higher the rocket's success rate. But it seems difficult to make decisions based on this figure because, no clear pattern can be found between successful launch and payload mass.

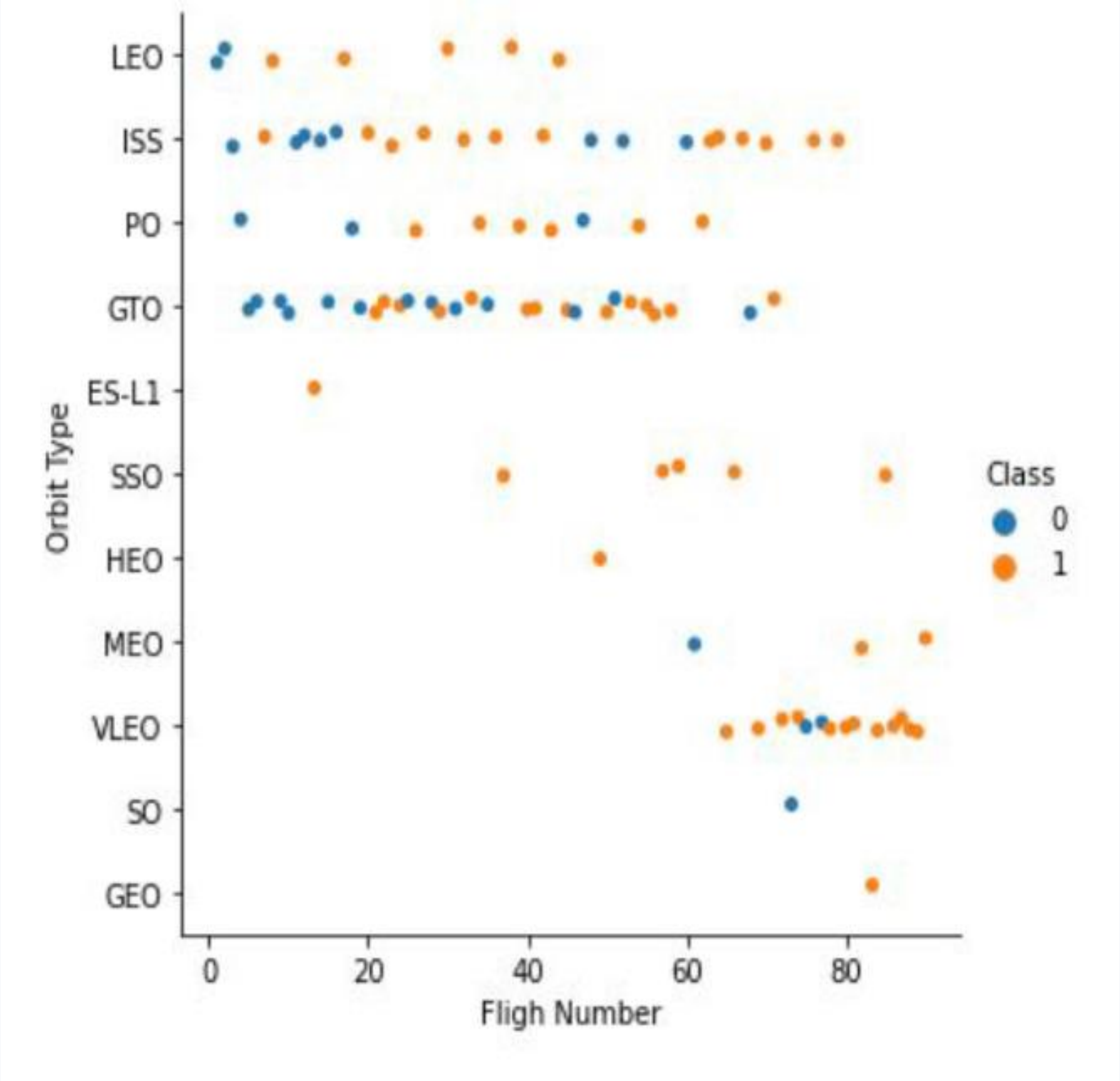
Success Rate vs. Orbit Type



- Orbit types SSO, HEO, GEO & ES-L1 has the highest success rates which is 100%.

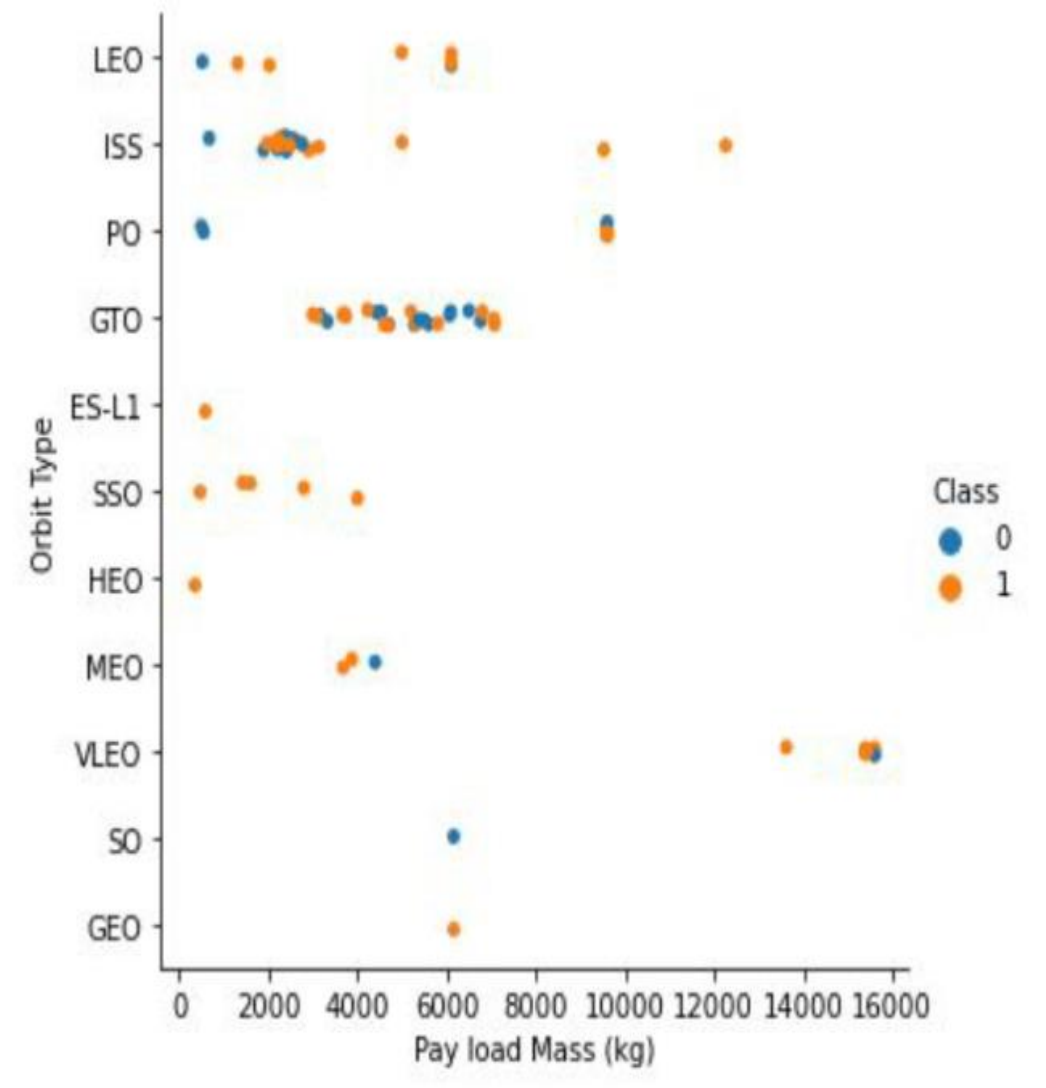
- On other hand, the success rate of orbit type GTO is only 50%, and it is the lowest except for type SO, which recorded failure in a single attempt.

Flight Number vs. Orbit Type



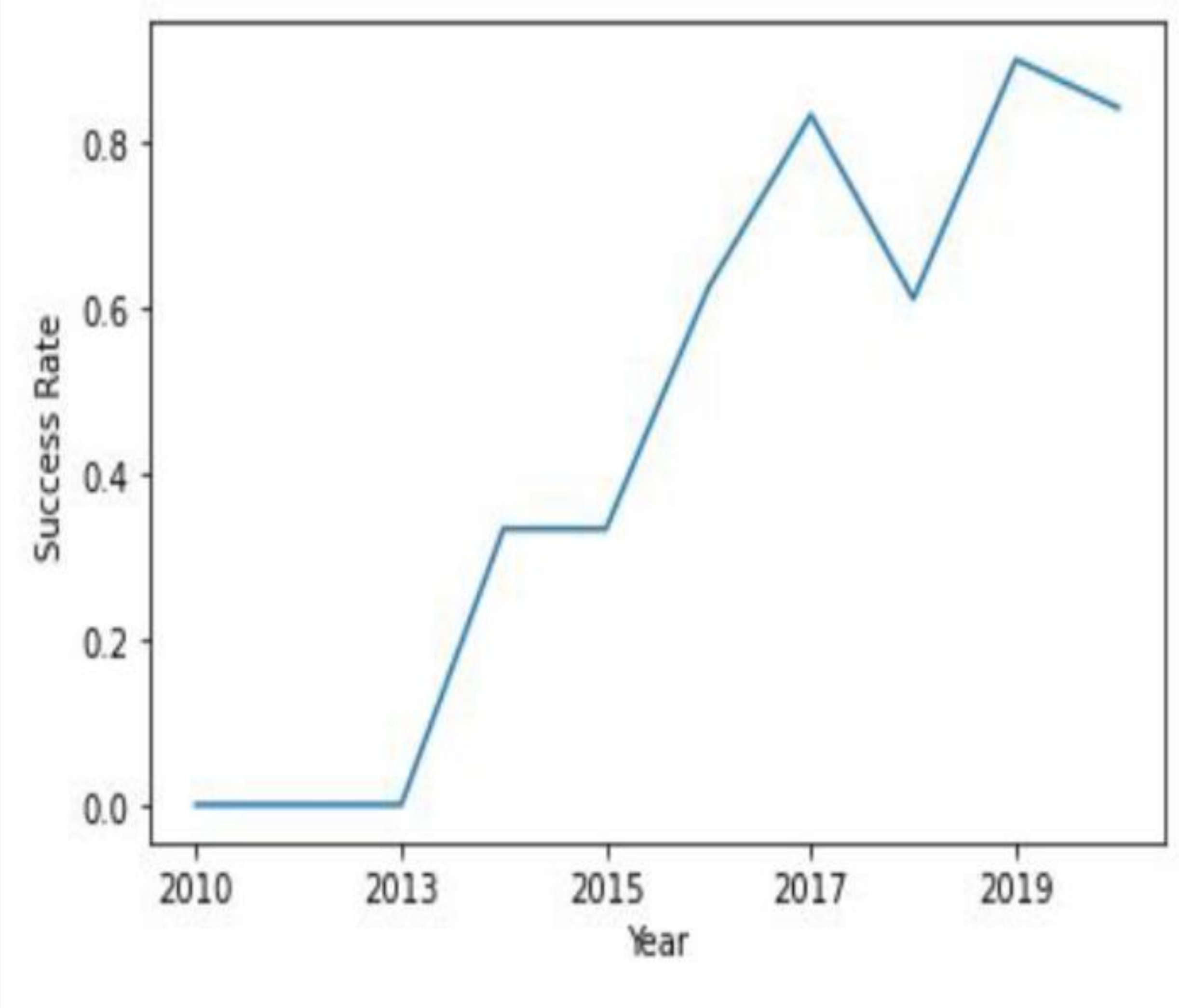
- Class0(blue) represents unsuccessful launch, and class 1(orange) represents successful launch.
- In most cases, the launch outcome seems to be correlated with the flight number.
- On the other hand, in GTO orbit, there seems to be no relationship between flight numbers & success rate.
- SpaceX starts with LEO with a moderate success rate and it seems that VLEO, which has a high success rate is used the most in recent launches.

Payload vs. Orbit Type



- Class0(blue) represents unsuccessful launch, and
  class 1(orange) represents successful launch.
- With heavy payloads, the successful landing or positive landing rate are more for LEO & ISS.
- However, in case of GTO, it is hard to distinguish between

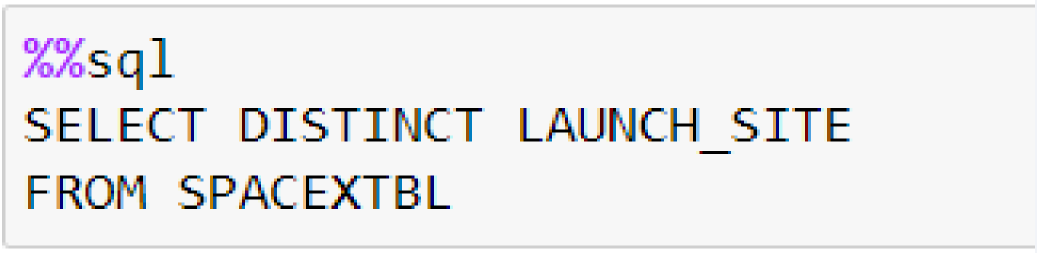
Launch Success Yearly Trend



- Since 2013, success rate has continued to increase until 2017.
- The rate decreased slightly in 2018.
- Recently, it has shown a success rate of 80%.

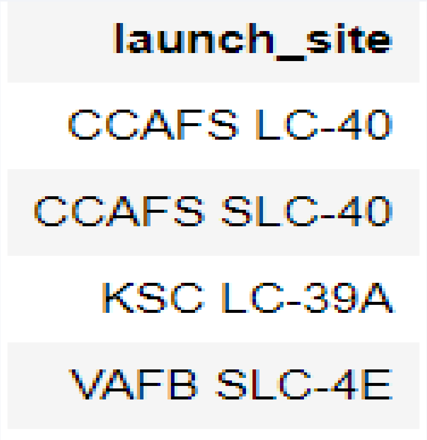
All Launch Site Names

Query :



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

Results :



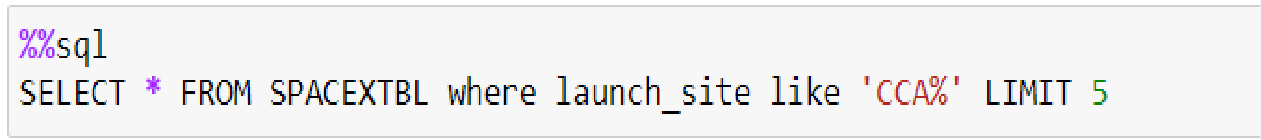
launch_site
CCAFS LC-40
CCAFS SLC-40
KSC LC-39A
VAFB SLC-4E

- When SQL DISTINCT clause is used in the query, only unique values are displayed in the launch_site column from SpaceX table.
- There are 4 unique launch sites :

CCAFS LC-40, CCAFS SLC-40, KSC LC-39A, VAFB SLC-4E.

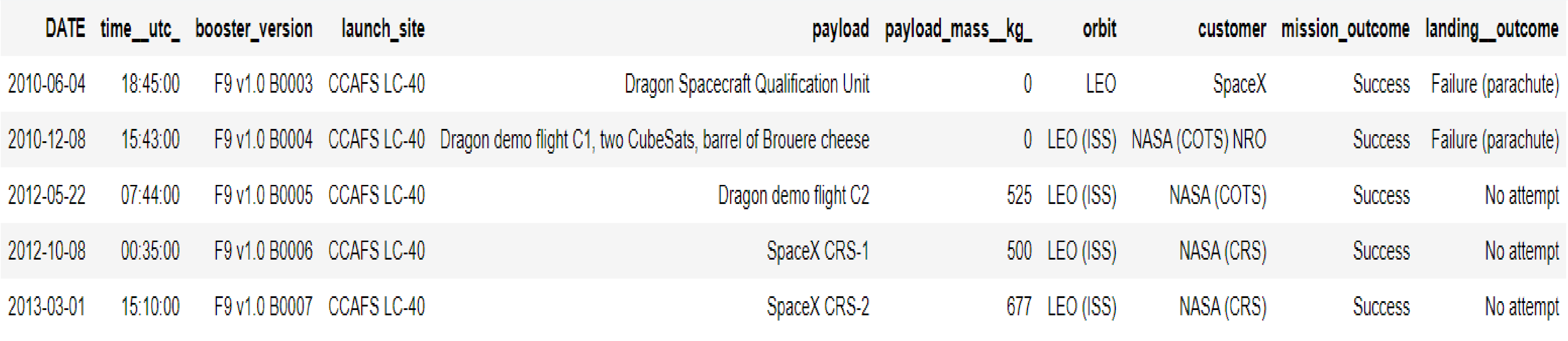
Launch Site Names Begin with 'CCA'

Query :



```
%%sql
SELECT * FROM SPACEXTBL where launch_site like 'CCA%' LIMIT 5
```

Results :

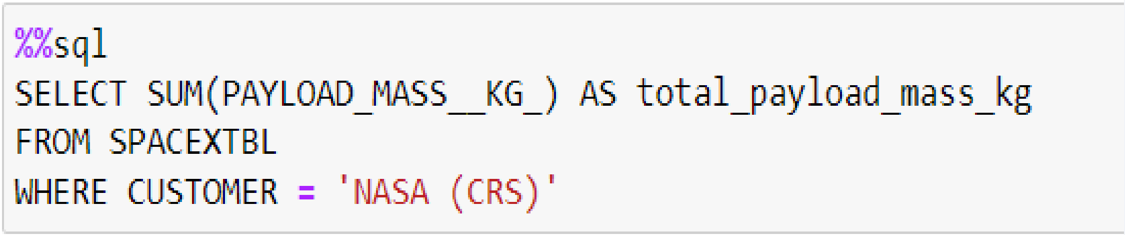


DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight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

- Only five records of SpaceX table were displayed using LIMIT 5 clause in query.
- Using the LIKE operator along with '%' sign together, the Launch_site name starting with CAA could be called.

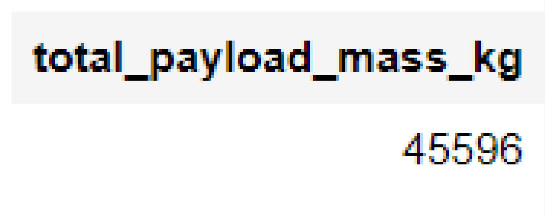
Total Payload Mass

Query :



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

Results :

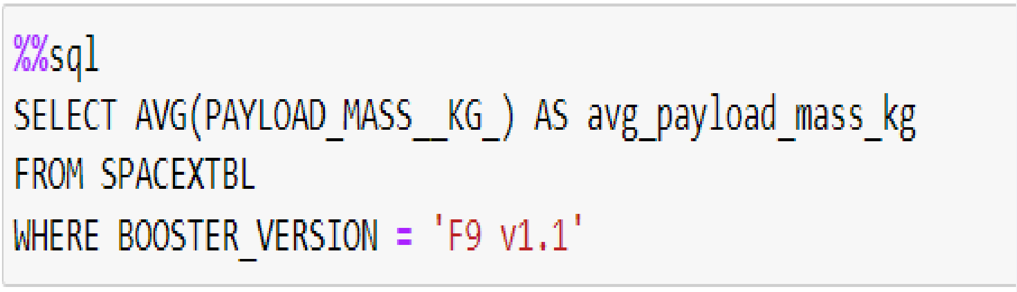


total_payload_mass_kg
45596

- Using the sum() function, to calculate the sum of the column PAYLOAD_MASS_KG_.
- In the where clause, filter the dataset to perform calculations only if Customer is NASA(CRS).

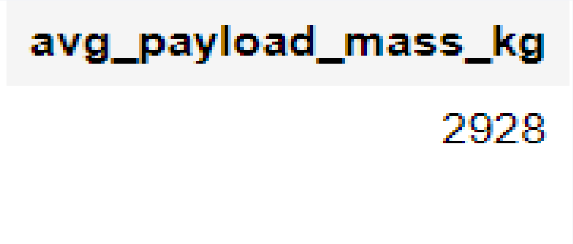
Average Payload Mass by F9 v1.1

Query :



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

Results :

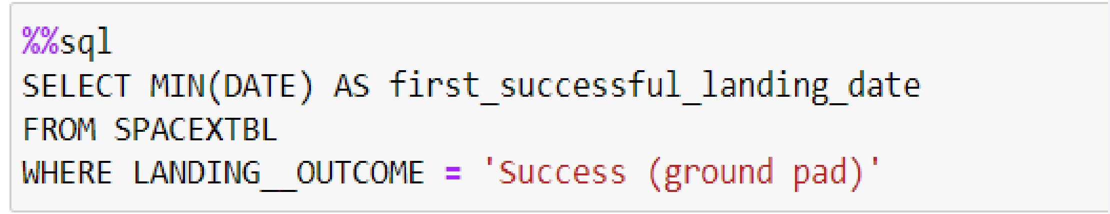


avg_payload_mass_kg
2928

- Using the AVG() function to calculate the average value of column PAYLOAD_MASS_KG_.
- In the where clause, filter the dataset to perform calculations only if Booster version is F9 v1.1.

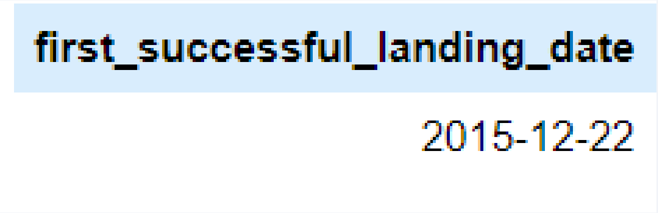
First Successful Ground Landing Date

Query :



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

Results :

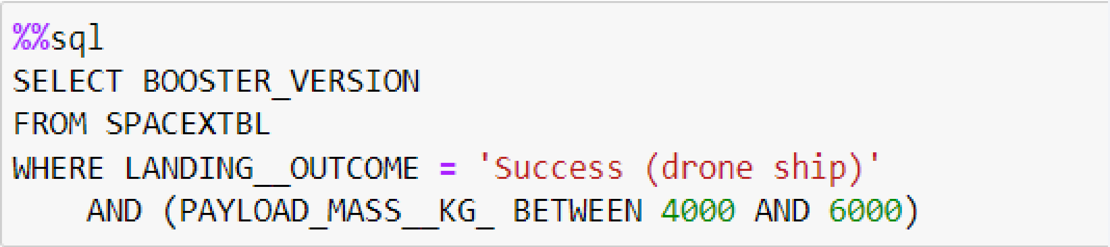


first_successful_landing_date
2015-12-22

- Using min() function to find out the earliest date in the column DATE.
- In the WHERE clause, filter the dataset to perform a search only if Landing_outcome is Success(Ground Pad).

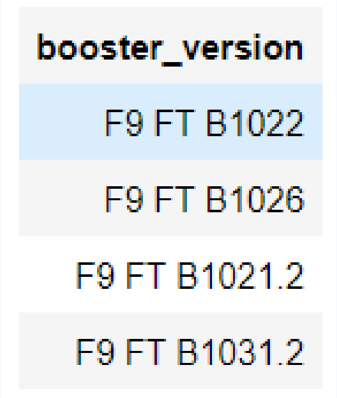
Successful Drone Ship Landing with Payload between 4000 and 6000

Query :



```
%%sql
SELECT BOOSTER_VERSION
FROM SPACEXTBL
WHERE LANDING__OUTCOME = 'Success (drone ship)'
    AND (PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000)
```

Results :

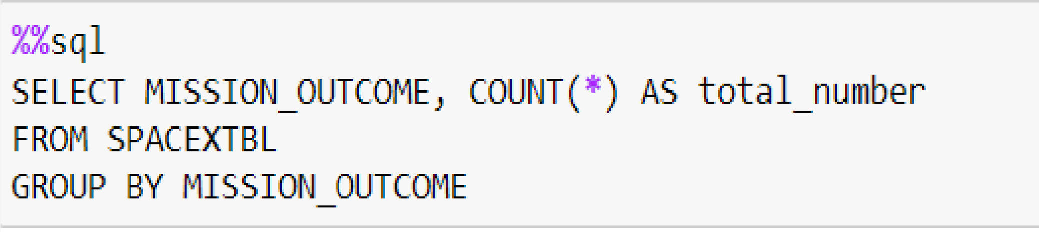


booster_version
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
F9 FT B1031.2

- In the WHERE clause, filter the dataset to perform a search if Landing_outcome is Success (drone_ship).
- Using the AND operator to display a record if additional condition PAYLOAD_MASS_KG is between 4000 & 6000.

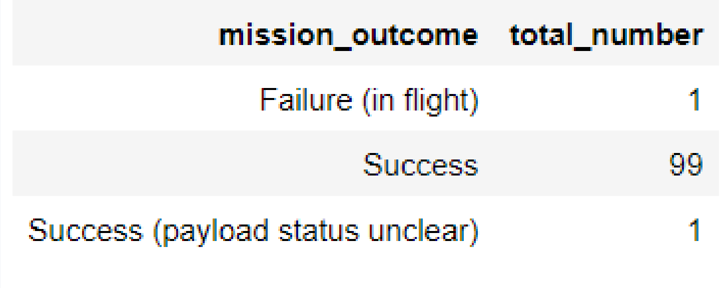
Total Number of Successful and Failure Mission Outcomes

Query :



```
%%sql
SELECT MISSION_OUTCOME, COUNT(*) AS total_number
FROM SPACEXTBL
GROUP BY MISSION_OUTCOME
```

Results :

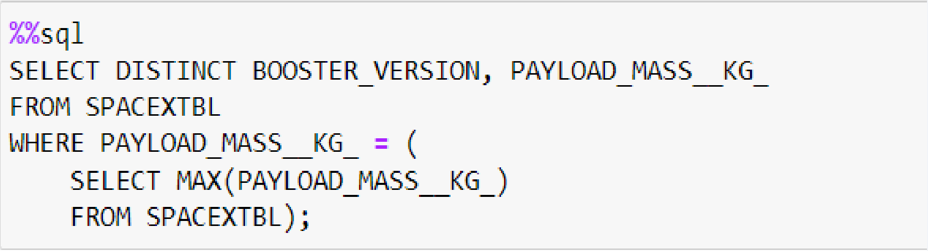


mission_outcome	total_number
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

- Using the COUNT() function to calculate the total number of columns.
- Using group by statement, groups rows that have same values into summary rows to find the total number in each Mission_outcome.
- According to the result, SpaceX seems to have successfully completed nearly 99% of its mission.

Boosters Carried Maximum Payload

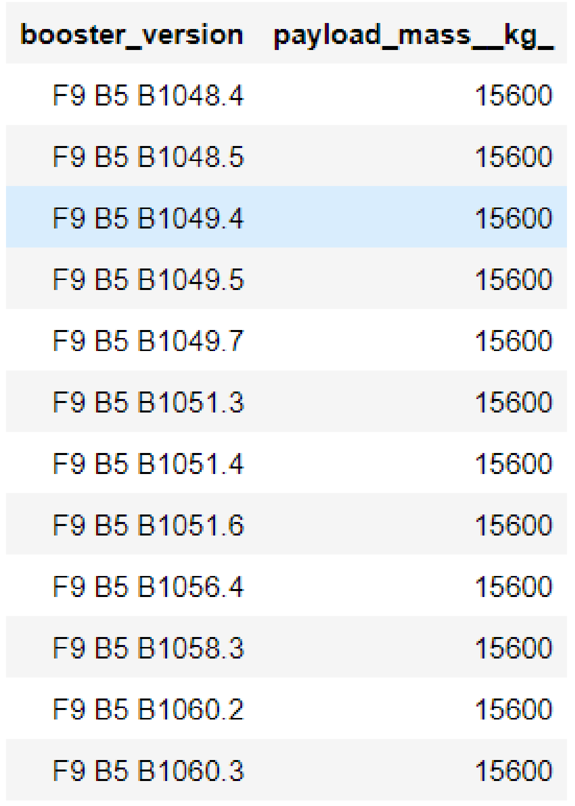
Query :



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

- Using a subquery, first find the maximum value of payload by using MAX() function, and second filter the dataset to perform a search if PAYLOAD_MASS_KG_ is the maximum value of the payload.
- According to the result, version F9 B5 B10xx.x boosters could carry the maximum payload.

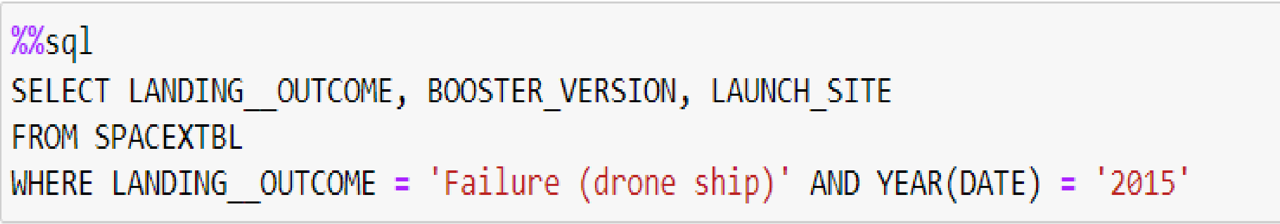
Results :



booster_version	payload_mass__kg_
F9 B5 B1048.4	15600
F9 B5 B1048.5	15600
F9 B5 B1049.4	15600
F9 B5 B1049.5	15600
F9 B5 B1049.7	15600
F9 B5 B1051.3	15600
F9 B5 B1051.4	15600
F9 B5 B1051.6	15600
F9 B5 B1056.4	15600
F9 B5 B1058.3	15600
F9 B5 B1060.2	15600
F9 B5 B1060.3	15600

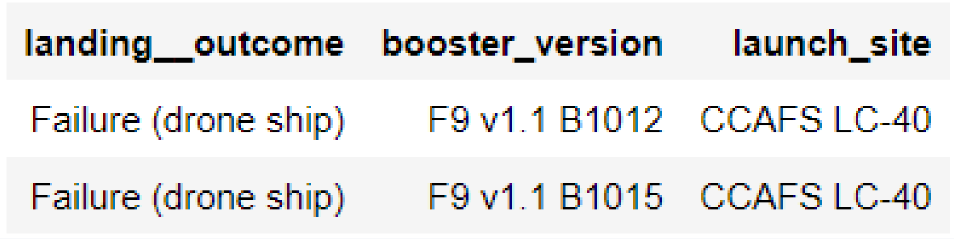
2015 Launch Records

Query :



```
%%sql
SELECT LANDING__OUTCOME, BOOSTER_VERSION, LAUNCH_SITE
FROM SPACEXTBL
WHERE LANDING__OUTCOME = 'Failure (drone ship)' AND YEAR(DATE) = '2015'
```

Results :

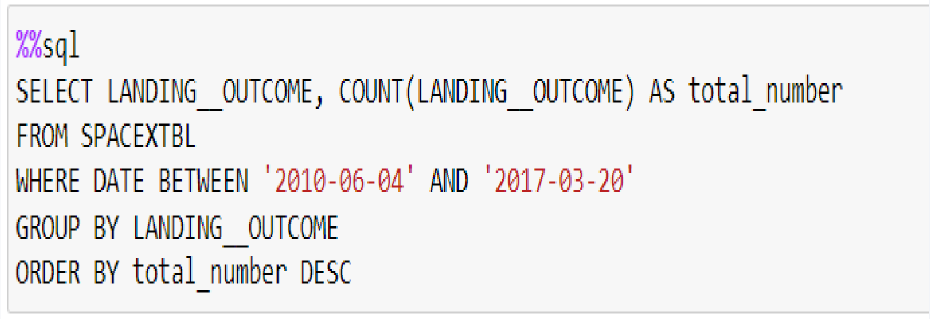


landing__outcome	booster_version	launch_site
Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

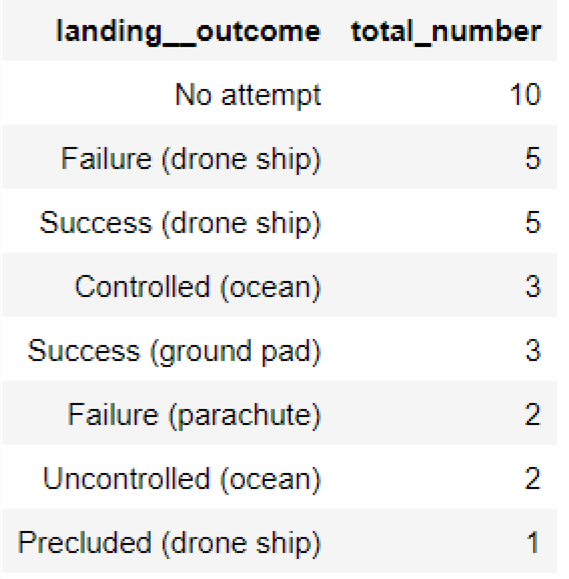
- In the WHERE clause, filter the dataset to perform to search if Landing_outcome is a failure(drone ship).
- Using the AND operator to display a record if additional condition YEAR is 2015.
- In 2015, there were 2 landing failure on drone ships.

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

Query & Results:



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



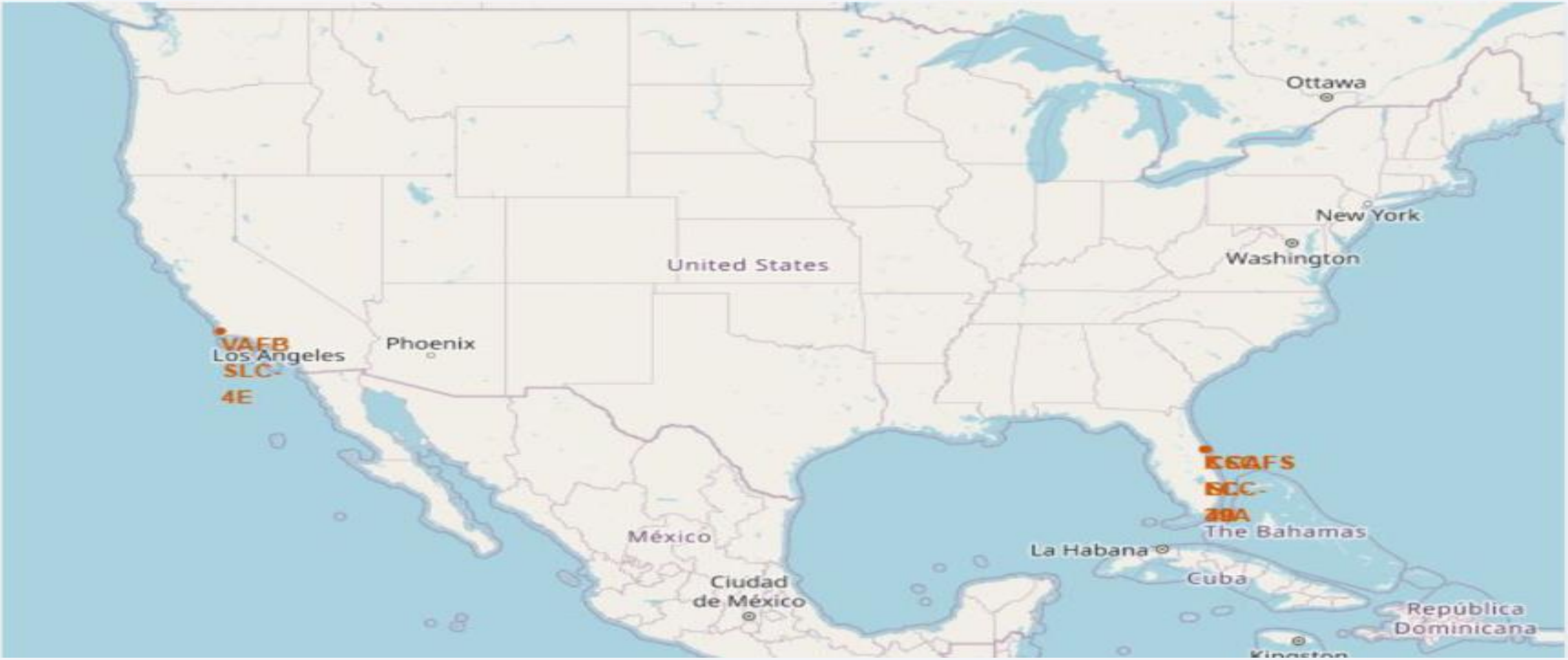
landing__outcome	total_number
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

- In the WHERE clause, filter the dataset to perform a search if the date is between 2010-06-04 & 2017-03-20.
- Using the order by keyword to sort the records by total number of landings and using DESC keyword to sort the records in the descending order.
- According to the results, the number of successes and failures between 2010-06-04 & 2017-03-20 was similar.

Section 3

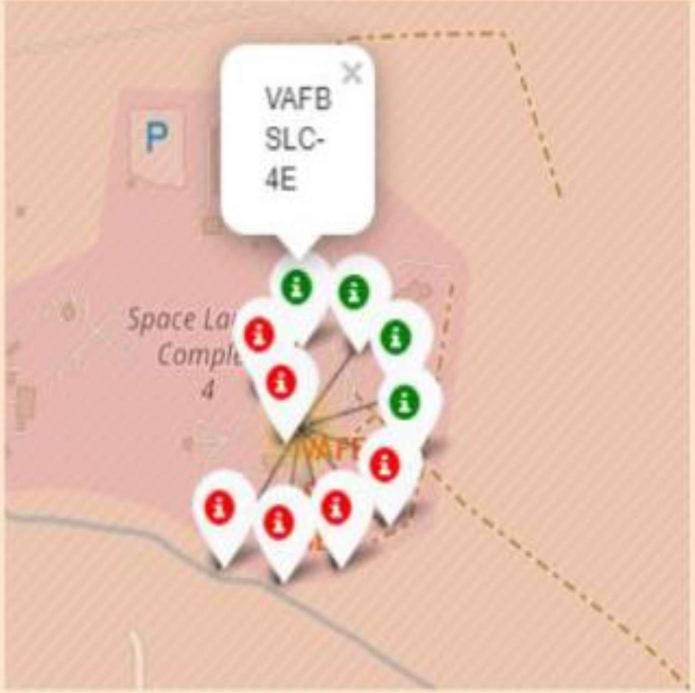
Launch Sites Proximities Analysis

All launch site locations

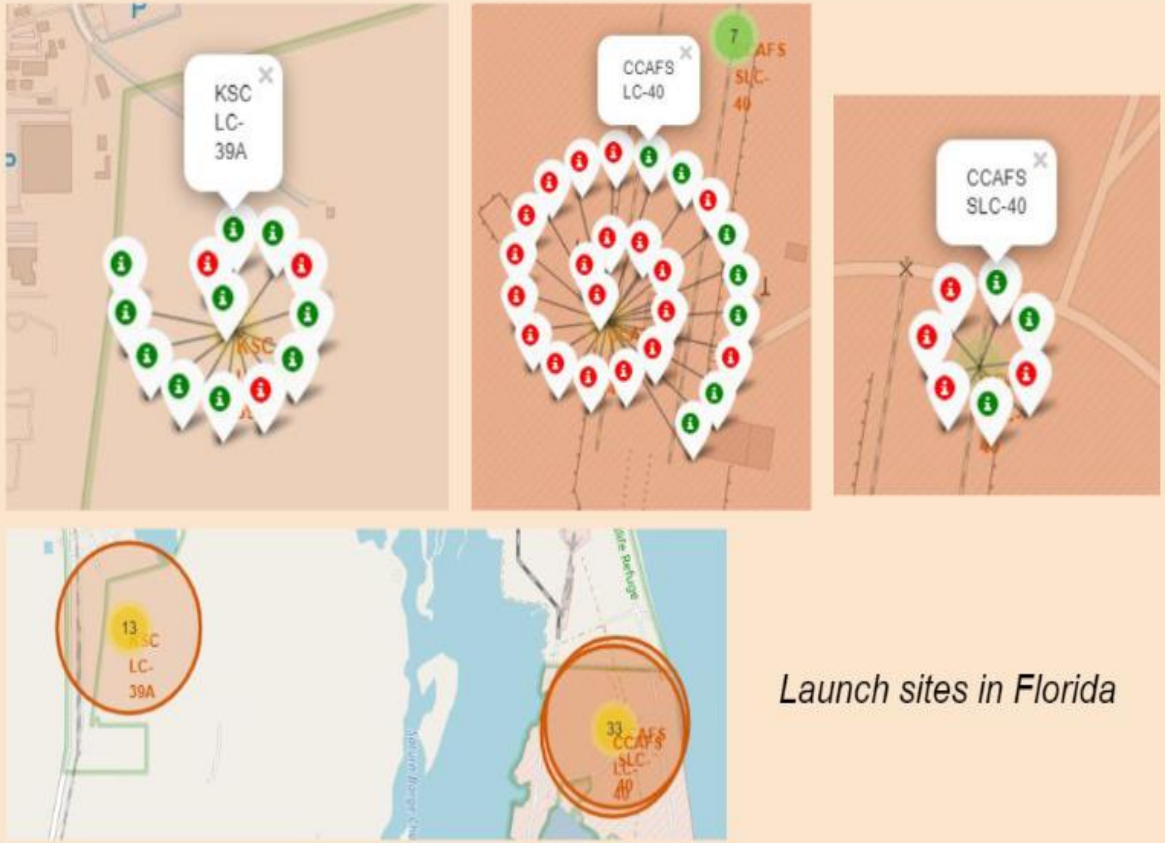


- We can see that the SpaceX launch sites are in the United States of America coasts – Florida & California.

Color labelled launch Outcomes



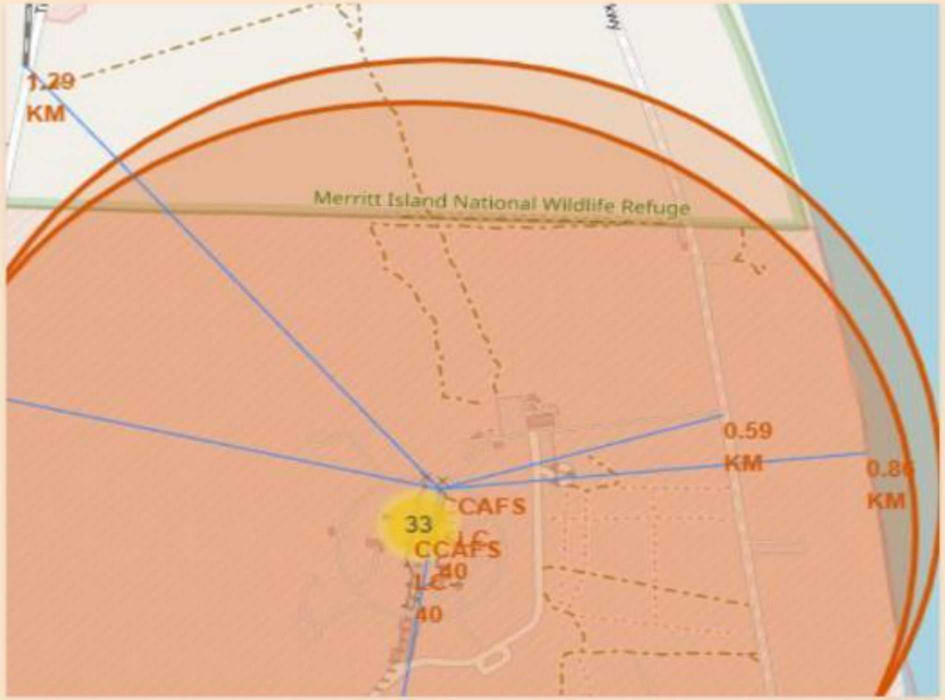
Launch site in California



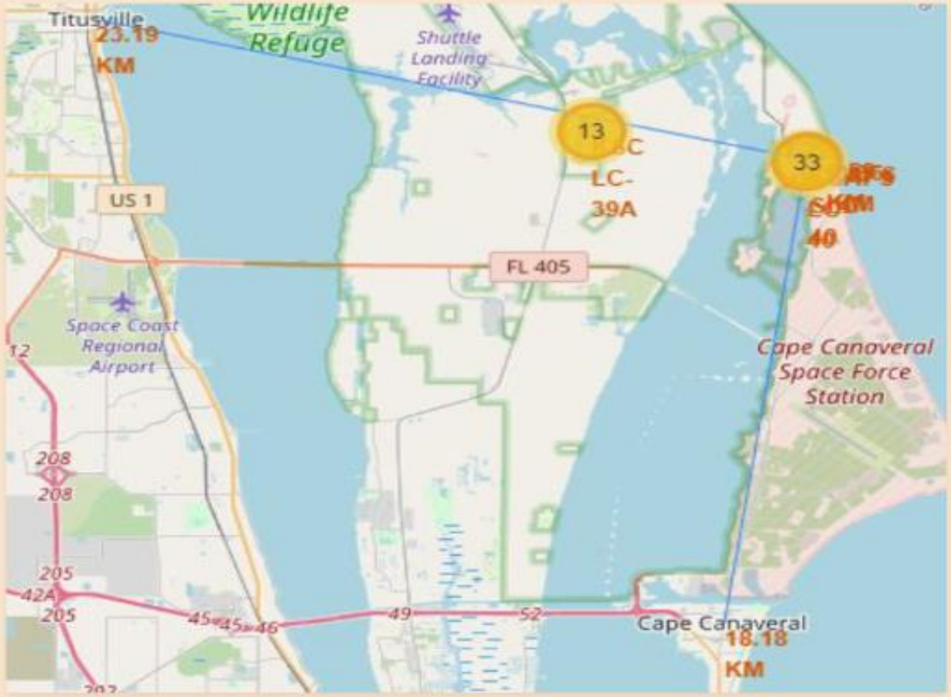
Launch sites in Florida

Green marker shows successful launches and red marker shows failures.

Proximity of launch sites



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***

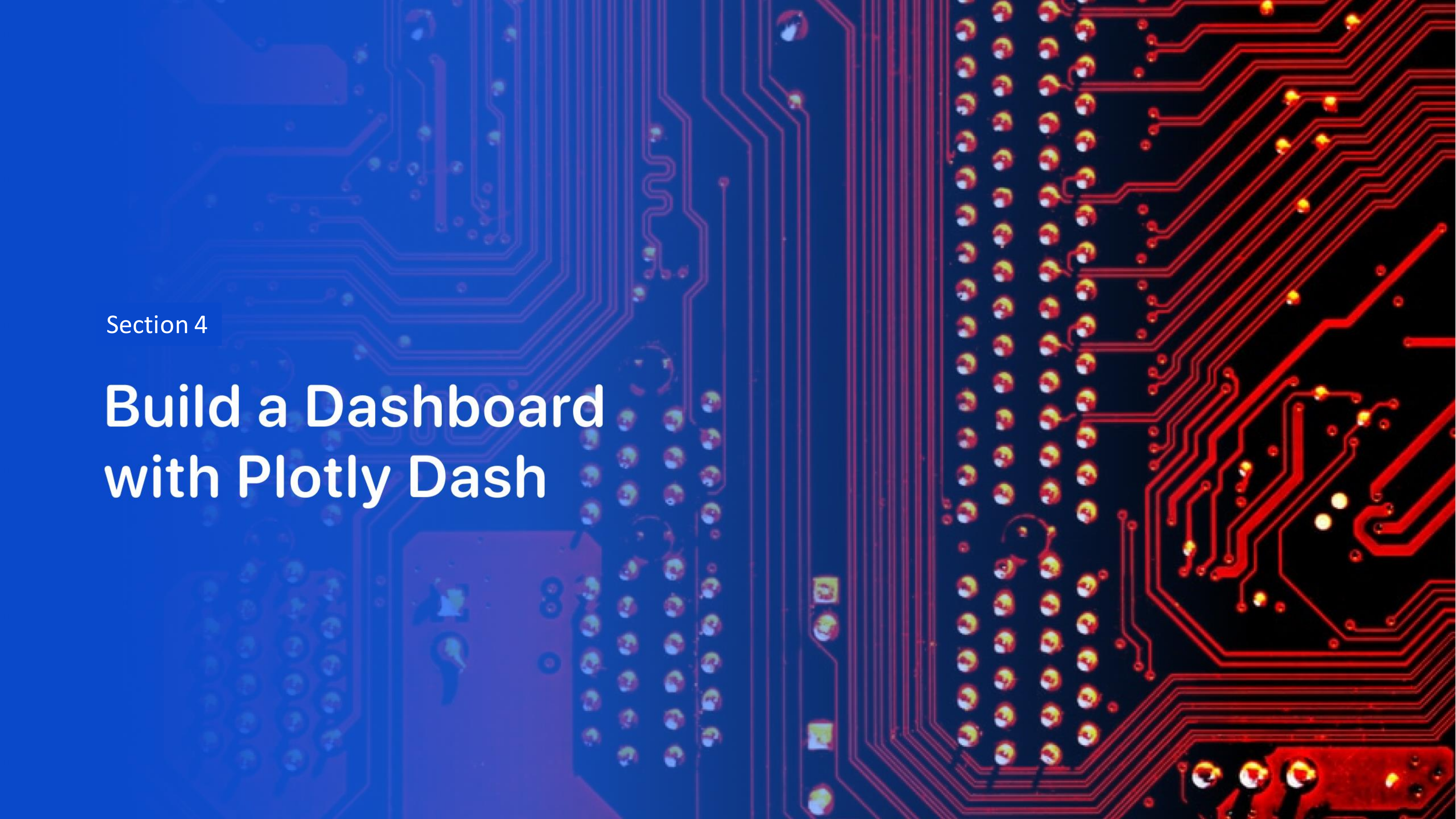


Do launch sites keep certain distance away from cities? ***Yes***

By above figures we can see that, the launch site is close to railways &highways for transportation of equipment or personnel, and is also close to coastline and relatively far from the cities such that launch failure does not pose a threat.

Section 4

Build a Dashboard with Plotly Dash

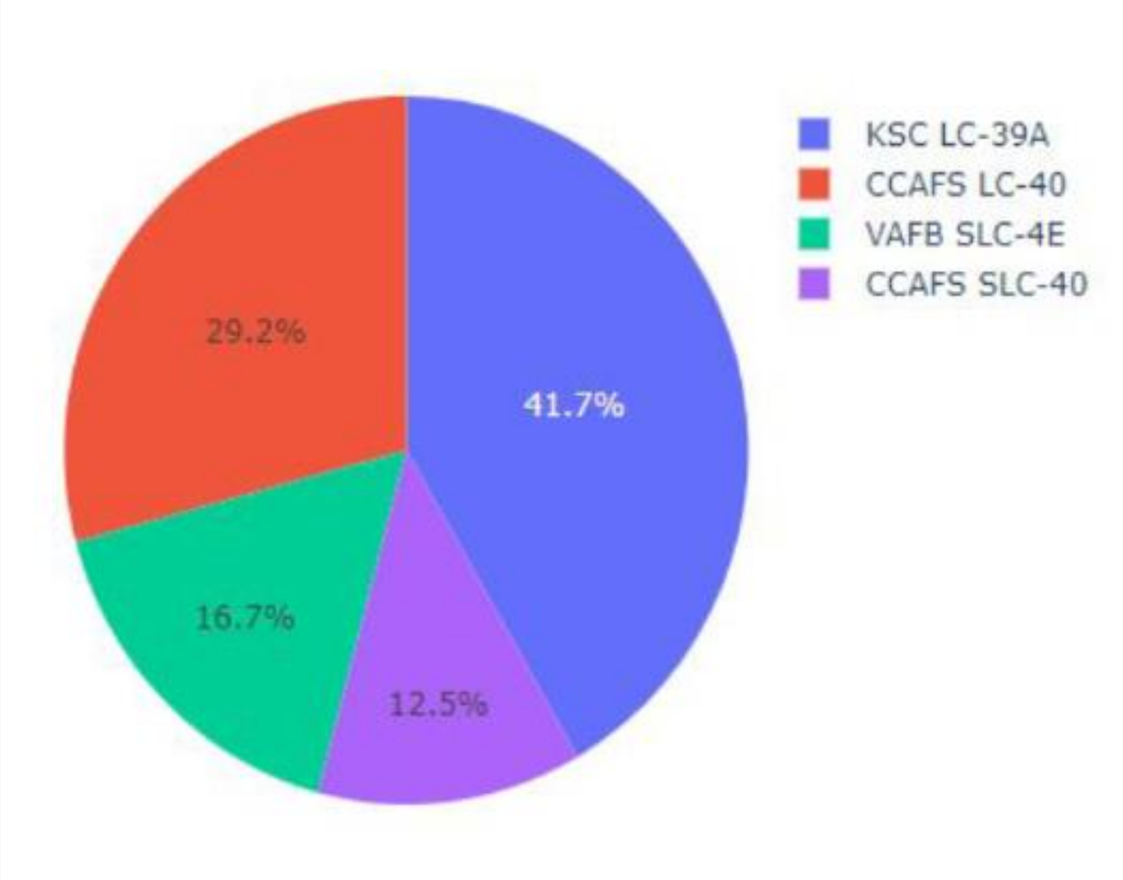


Success percentage achieved by each launch site

- KSC LC-39A records the most launch success among all the sites.
- The VAFB SLC-4E has the fewest launch success, possibly because

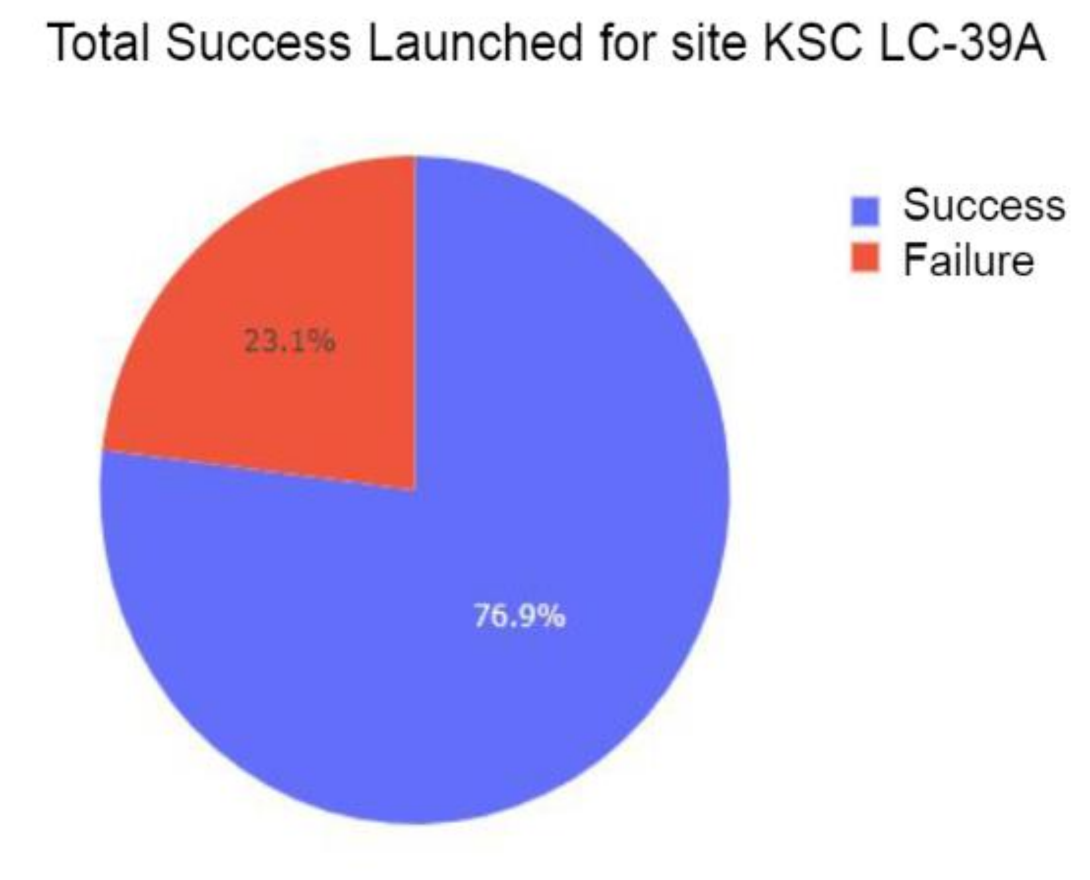
  - data sample is small or

  - because it is the only site located in California, so the launch difficulty on west coast may be higher than on the east coast.

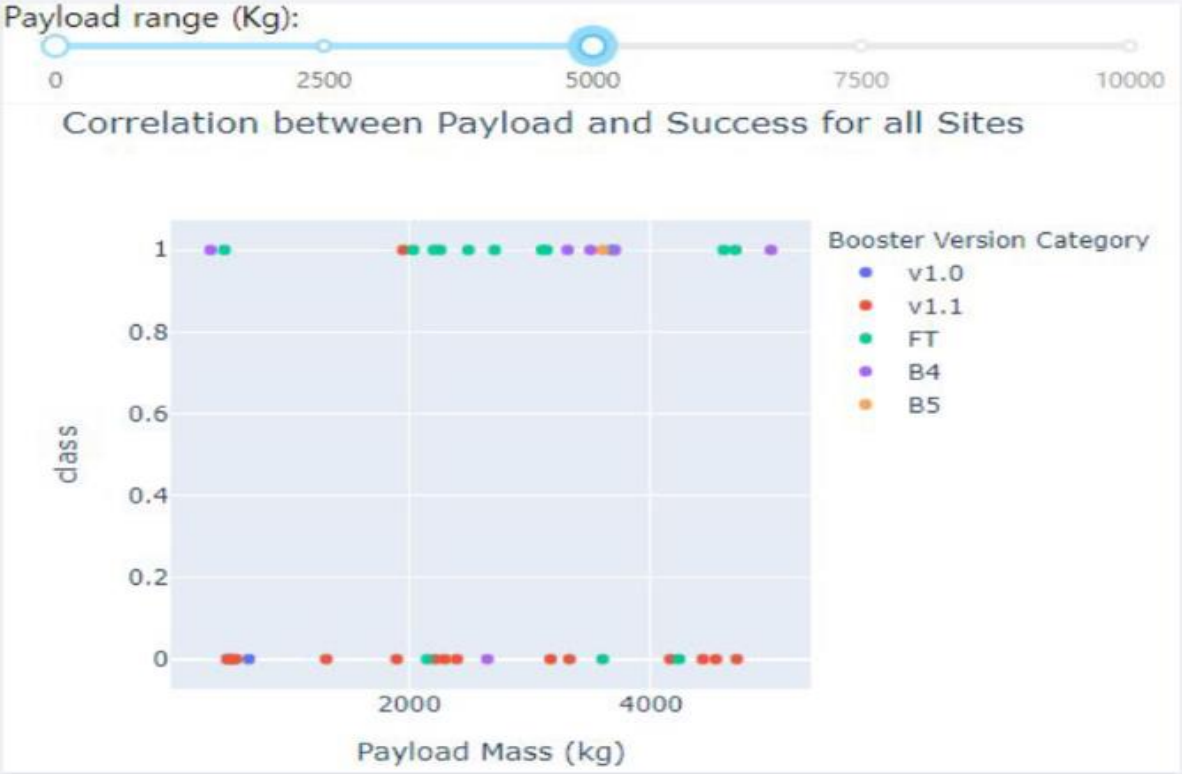


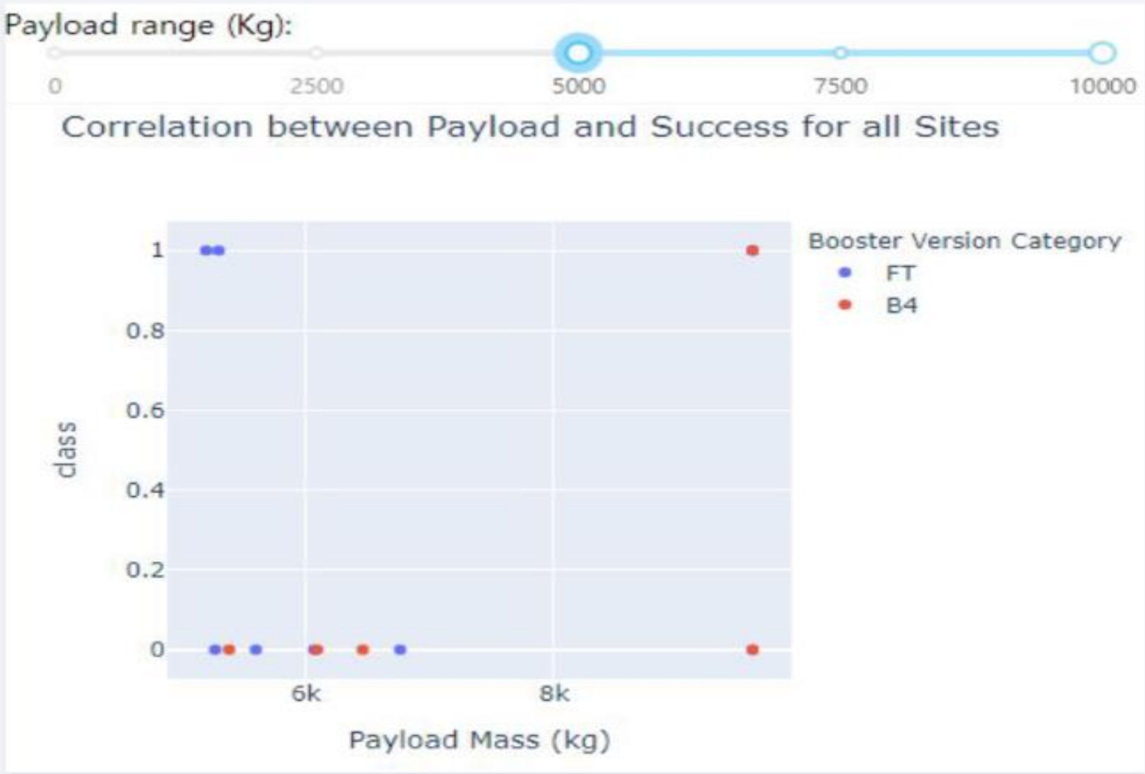
Launch site with Highest launch success ratio

- KLSC-39A has highest success rate with 10 landing successes (76.9%) and 3 landing failures (23.1%).



Payload vs Launch outcome Scatter plot for all sites





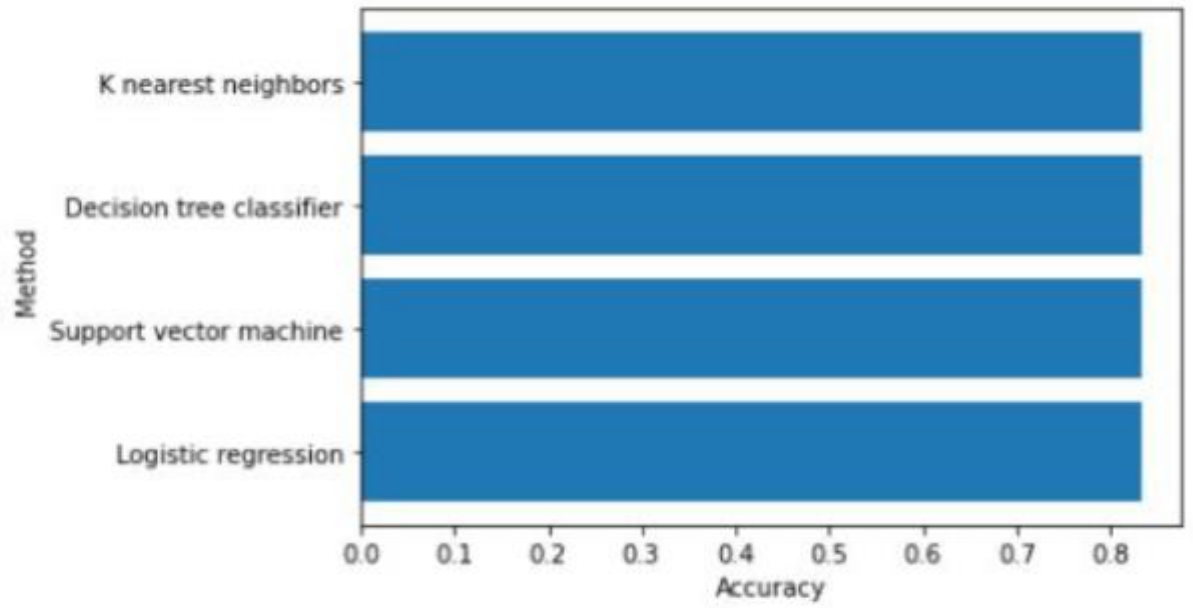
- The above figure shows that the launch success rate(class 1) for low weighted payloads(0-5000kg) is higher than that of heavy weighted payloads(5000-10000kg).

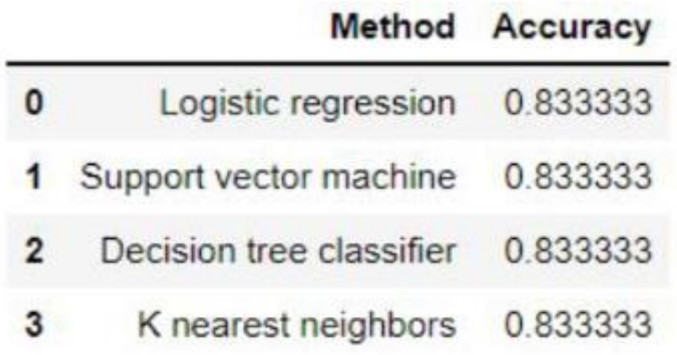
Section 5

Predictive Analysis (Classification)

Classification Accuracy

- In the test set, the accuracy of all models was virtually the same at 83.33%.
- It should be noted that the test size was small at 18.
- Therefore, more data is needed to determine optimal model.

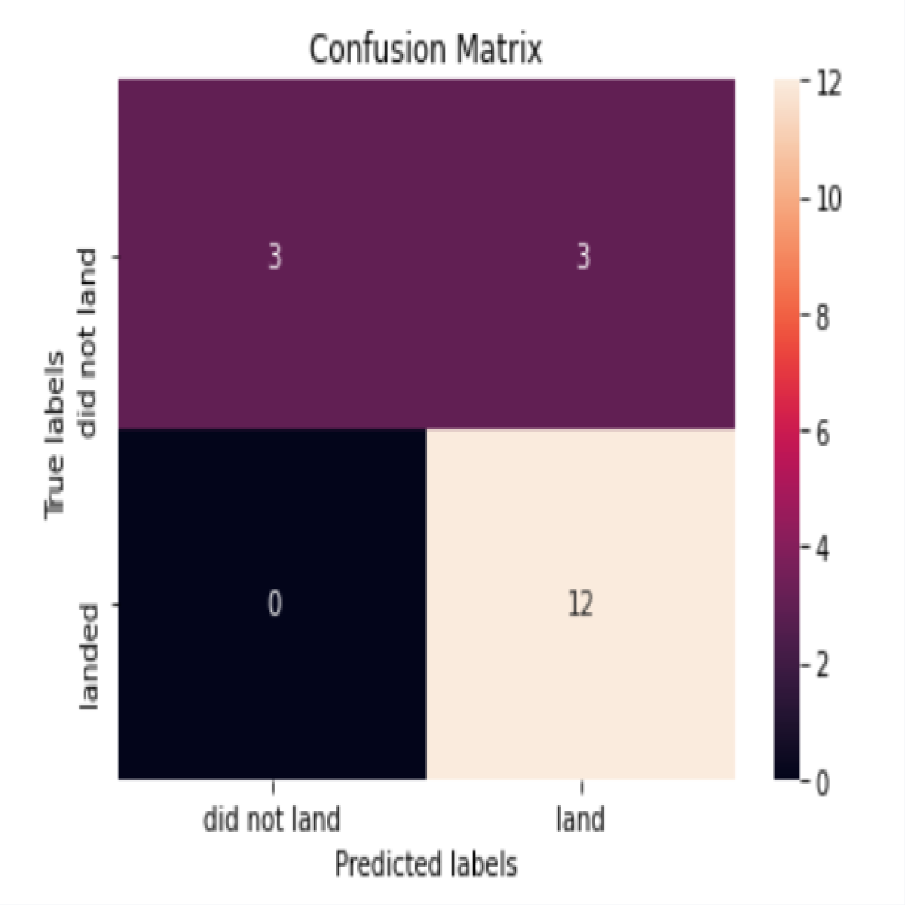




	Method	Accuracy
0	Logistic regression	0.833333
1	Support vector machine	0.833333
2	Decision tree classifier	0.833333
3	K nearest neighbors	0.833333

Confusion Matrix

- Confusion matrix is same for all models because all models performed the same for the test set.
- The models predicted 12 successful landings when the true label was successful and 3 failed landings when the true label was a failure. But there were also 3 predictions that said successful landings when the true label was a failure (false positives).
- These models predict successful landings.



Conclusions

- As the number of flights increased, the success rate increased, & recently it has exceeded 80%.
- Orbital types SSO, HEO, GEO, & ES-L1 have the highest success rate(100%).
- The launch site is close to railways, highways, & coastline, but far from cities.
- KSLC-39A has the highest no of launch successes & highest success rate among all sites.
- The launch success rate of low weighted payloads is higher than that of heavy weighted payloads.
- In this dataset, all models have the same accuracy(83.33%), but it seems that more data is needed to determine the optimal model due to the small data size.

Appendix

GITHUB URL

Thank you!