IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Dagmawi B. Tadesse
24/05/2022

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection with API and Web Scraping
  - Data Wrangling
  - Exploratory Data Analysis (EDA) with Data Visualization
  - EDA with SQL
  - Building an interactive map with Folium
  - Building a Dashboard with Plotly Dash
  - Machine Learning Predictive Analysis (Classification)
- Summary of all results
  - Exploratory Data Analysis results
  - Interactive maps and dashboard
  - Predictive results from Machine Learning

Introduction

- Project background and context:
  - The aim of this project is to predict if the Falcon 9 first stage will successfully land. SpaceX says on its website that the Falcon 9 rocket launch cost 62 million dollars. Other providers cost upward of 165 million dollars each. The price difference is explained by the fact that SpaceX can reuse the first stage. By determining if the stage will land, we can determine the cost of a launch. This information is interesting for another company if it wants to compete with SpaceX for a rocket launch.
- Problems you want to find answers:
  - What are the main characteristics of a successful landing?
  - How are the rocket variables correlated and how they are affecting successful landing rate ?
  - What are the conditions to get the best results and ensure the best successful landing rate?

Section 1

Methodology

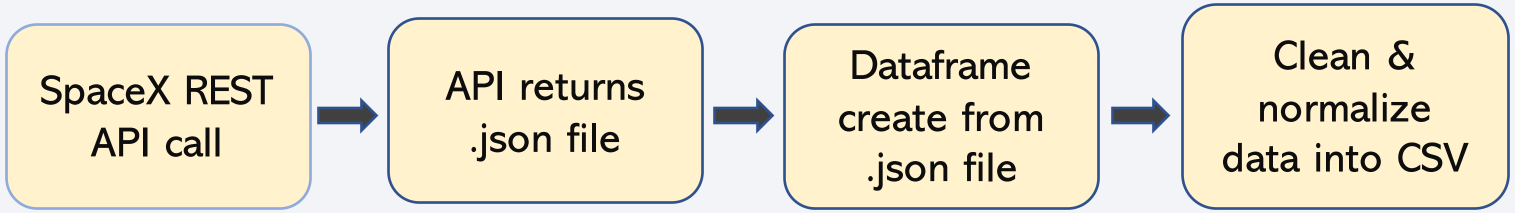
Methodology

Executive Summary

- Data collection methodology:
  - Data was collected using SpaceX REST API and web scrapping Falcon 9 and Falcon Heavy Launches Records from Wikipedia
- Perform data wrangling
  - Data was processed using one-hot encoding for categorical features
- 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

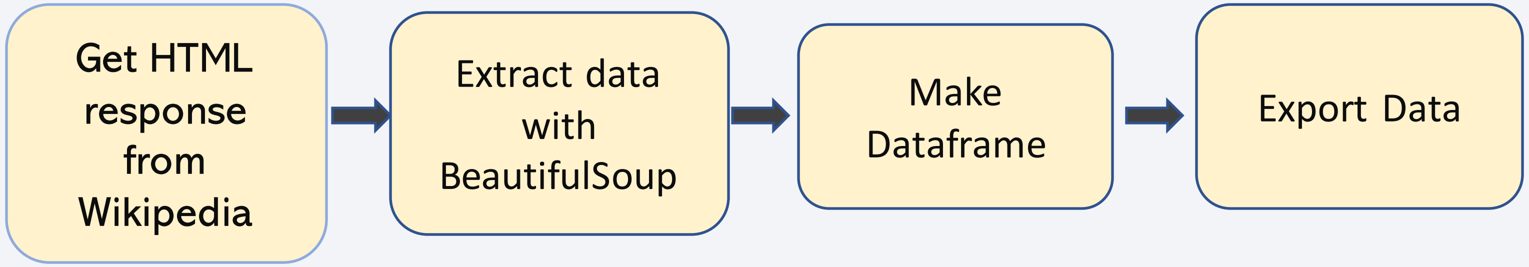
Data Collection

- Data was collected using SpaceX REST API and web scrapping from Wikipedia
  - **SpaceX REST API Data Columns:** `FlightNumber, Date, BoosterVersion, PayloadMass, Orbit, LaunchSite, Outcome, Flights, GridFins, Reused, Legs, LandingPad, Block, ReusedCount, Serial, Longitude, Latitude`



SpaceX REST API call → API returns .json file → Dataframe create from .json file → Clean & normalize data into CSV

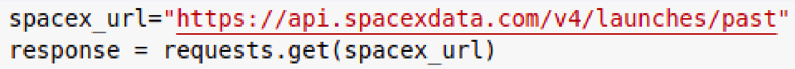
- **Wikipedia Web Scrape Data Columns:** `Flight No., Launch site, Payload, PayloadMass, Orbit, Customer, Launch outcome, Version Booster, Booster landing, Date, Time`



Get HTML response from Wikipedia → Extract data with BeautifulSoup → Make Dataframe → Export Data

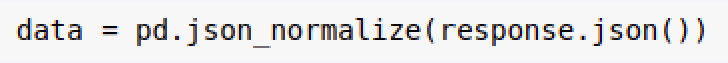
Data Collection – SpaceX API

1. Getting Response from API



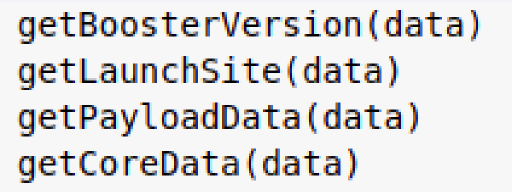
```
spacex_url="https://api.spacexdata.com/v4/launches/past"
response = requests.get(spacex_url)
```

2. Convert Response to JSON file



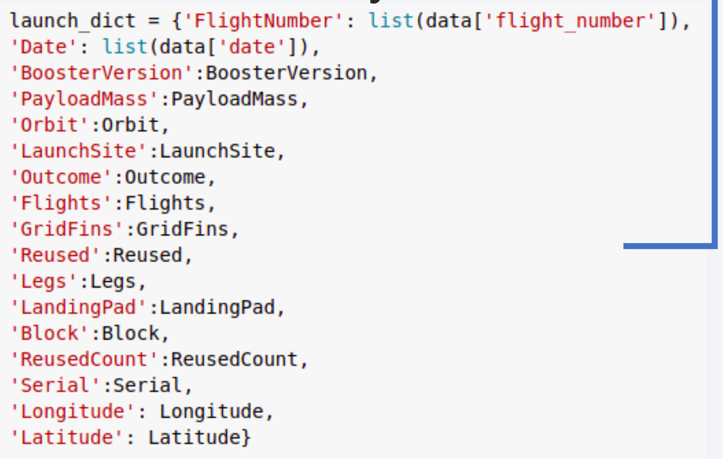
```
data = pd.json_normalize(response.json())
```

3. Transform data



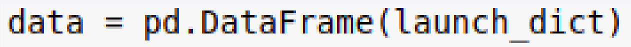
```
getBoosterVersion(data)
getLaunchSite(data)
getPayloadData(data)
getCoreData(data)
```

4. Create dictionary



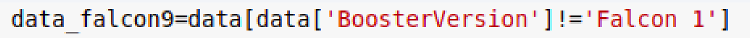
```
launch_dict = {'FlightNumber': list(data['flight_number']),
'Date': list(data['date']),
'BoosterVersion':BoosterVersion,
'PayloadMass':PayloadMass,
'Orbit':Orbit,
'LaunchSite':LaunchSite,
'Outcome':Outcome,
'Flights':Flights,
'GridFins':GridFins,
'Reused':Reused,
'Legs':Legs,
'LandingPad':LandingPad,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

5. Create dataframe



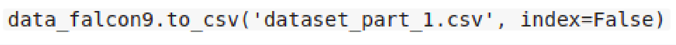
```
data = pd.DataFrame(launch_dict)
```

6. Filter dataframe



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

7. Export to csv file

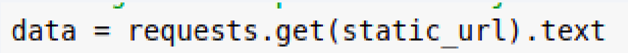


```
data_falcon9.to_csv('dataset_part_1.csv', index=False)
```

Link to github to access the code

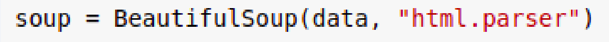
Data Collection - Scraping

1. Getting Response from HTML



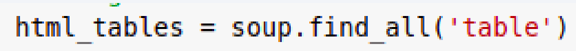
```
data = requests.get(static_url).text
```

2. Create BeautifulSoup object



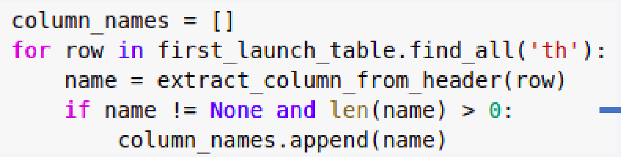
```
soup = BeautifulSoup(data, "html.parser")
```

3. Find all tables



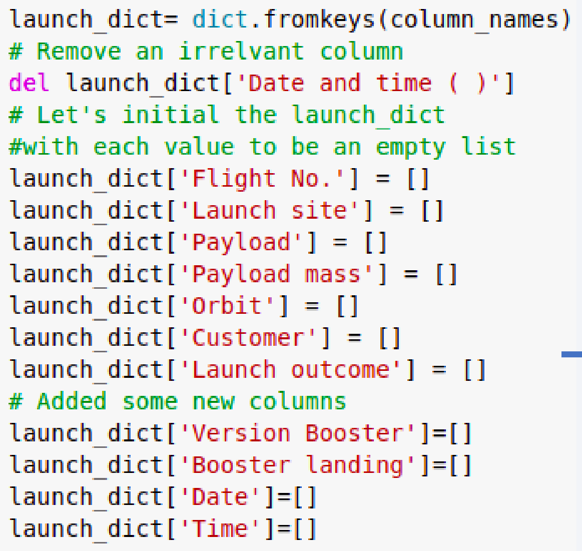
```
html_tables = soup.find_all('table')
```

4. Get column names



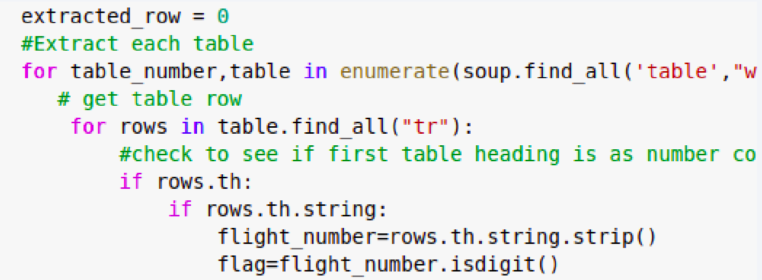
```
column_names = []
for row in first_launch_table.find_all('th'):
    name = extract_column_from_header(row)
    if name != None and len(name) > 0:
        column_names.append(name)
```

5. Create dictionary



```
launch_dict= dict.fromkeys(column_names)
# Remove an irrelvant column
del launch_dict['Date and time ( )']
# Let's initial the launch_dict
#with each value to be an empty list
launch_dict['Flight No.'] = []
launch_dict['Launch site'] = []
launch_dict['Payload'] = []
launch_dict['Payload mass'] = []
launch_dict['Orbit'] = []
launch_dict['Customer'] = []
launch_dict['Launch outcome'] = []
# Added some new columns
launch_dict['Version Booster']=[]
launch_dict['Booster landing']=[]
launch_dict['Date']=[]
launch_dict['Time']=[]
```

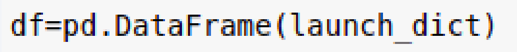
6. Add data to keys



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

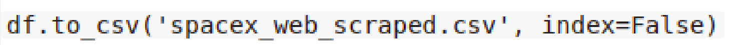
Note: Check notebook for the rest of the code

7. Create dataframe from dictionary



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

8. Export to csv file



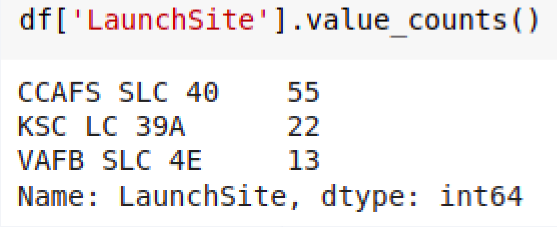
```
df.to_csv('spacex_web_scraped.csv', index=False)
```

Link to github to access the code

Data Wrangling

- There are several cases where the booster failed to land successfully
  - True Ocean, True RTLS, True ASDS means the mission has been successful.
  - False Ocean, False RTLS, False ASDS means the mission was a failure.
- Transform string variables into categorical variables where 1 = successful / 0 = failure

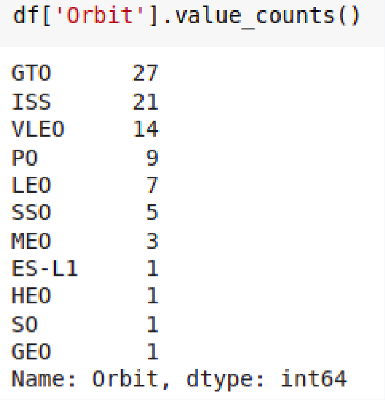
1. Calculate launches number for each site



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

```
CCAFS SLC 40    55
KSC LC 39A      22
VAFB SLC 4E     13
Name: LaunchSite, dtype: int64
```

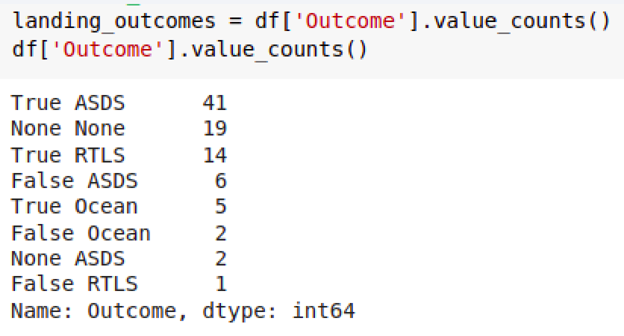
- 2. Calculate the number and occurrence of each orbit



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

```
GTO      27
ISS      21
VLEO     14
PO        9
LEO       7
SSO       5
MEO       3
ES-L1     1
HEO       1
SO        1
GEO       1
Name: Orbit, dtype: int64
```

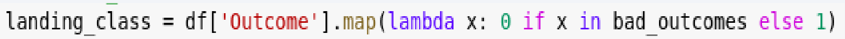
3. Calculate the number and occurrence of mission outcome per orbit type



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

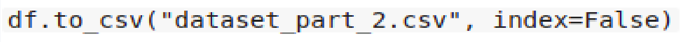
```
True ASDS      41
None None      19
True RTLS      14
False ASDS      6
True Ocean      5
False Ocean     2
None ASDS       2
False RTLS      1
Name: Outcome, dtype: int64
```

4. Create landing outcome label from outcome column



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

5. Export to csv file



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

Link to github to access the code

EDA with Data Visualization

- **Scatter chart** (show how the variables are correlated):
  - Flight Number vs. Launch Site
  - Payload vs. Launch Site
  - Flight Number vs. Orbit Type
  - Payload vs. Orbit Type
- **Bar charts** (show the relationship between numeric and categoric variables):
  - Success Rate vs Orbit Type vs.
- **Line charts** (show data variables and their trends. They can help to show global behavior and make prediction for unseen data.):
  - Year vs. Success Rate

Link to github to access the code

EDA with SQL

- We performed SQL queries to gather and understand data from dataset:
  - Displaying 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 successful landing outcome in ground pad was achieved.
  - List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000.
  - List the total number of successful and failure mission outcomes.
  - List the names of the booster_versions which have carried the maximum payload mass.
  - List the records which will display the month names, failure landing_outcomes in drone ship, booster versions, launch_site for the months in year 2015.
  - Rank the count of successful landing_outcomes between the date 04/06/2010 and 20/03/2017 in descending order.

Link to github to access the code

Build an Interactive Map with Folium

- Folium map object is a map centered on NASA Johnson Space Center at Houson, Texas
  - Red circle at NASA Johnson Space Center's coordinate with label showing its name *(folium.Circle, folium.map.Marker)*.
  - Red circles at each launch site coordinates with label showing launch site name *(folium.Circle, folium.map.Marker, folium.features.DivIcon)*.
  - The grouping of points in a cluster to display multiple and different information for the same coordinates *(folium.plugins.MarkerCluster)*.
  - Markers to show successful and unsuccessful landings. Green for successful landing and Red for unsuccessful landing *(folium.map.Marker, folium.Icon)*.
  - Markers to show distance between launch site to key locations (railway, highway, coastway, city) and plot a line between them *(folium.map.Marker, folium.PolyLine, folium.features.DivIcon)*
- These objects are created in order to understand better the problem and the data. We can show easily all launch sites, their surroundings and the number of successful and unsuccessful landings.

Link to github to access the code

Build a Dashboard with Plotly Dash

- Dashboard has dropdown, pie chart, range-slider and scatter plot components
  - Dropdown allows a user to choose the launch site or all launch sites *(dash_core_components.Dropdown)*.
  - Pie chart shows the total success and the total failure for the launch site chosen with the dropdown component *(plotly.express.pie)*.
  - Range-slider allows a user to select a payload mass in a fixed range *(dash_core_components.RangeSlider)* .
  - Scatter chart shows the relationship between two variables, in particular Success vs Payload Mass *(plotly.express.scatter)* .

Link to github to access the code

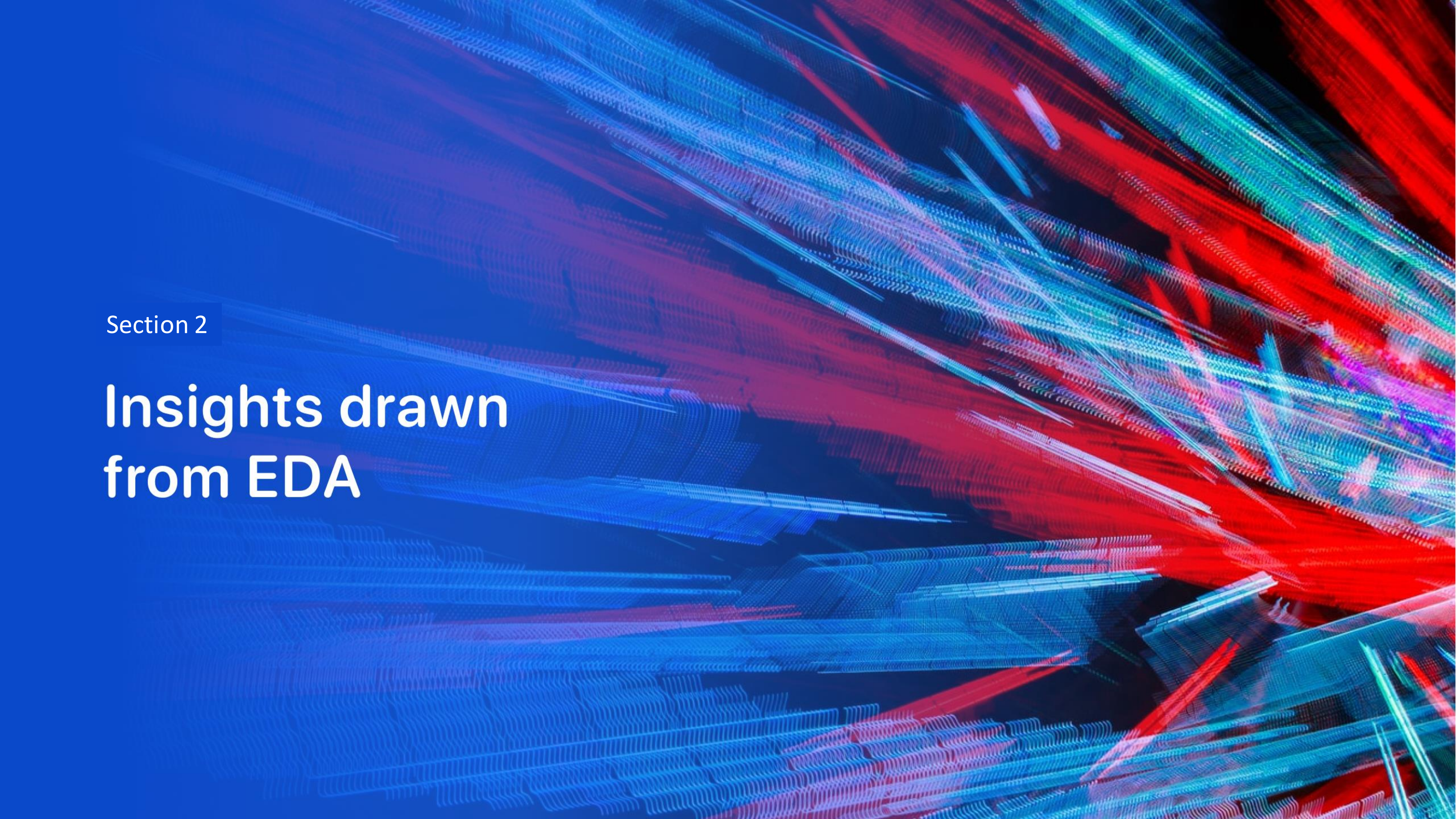
Predictive Analysis (Classification)

- Data preparation
  - Load dataset
  - Normalize data
  - Split data into training and test sets.
- Model preparation
  - Selection of machine learning algorithms
  - Set parameters for each algorithm to GridSearchCV
  - Training GridSearchModel models with training dataset
- Model evaluation
  - Get best hyperparameters for each type of model
  - Compute accuracy for each model with test dataset
  - Plot Confusion Matrix
- Model comparison
  - Comparison of models according to their accuracy
  - The model with the best accuracy will be chosen (see notebook for results)

Link to github to access the code

Results

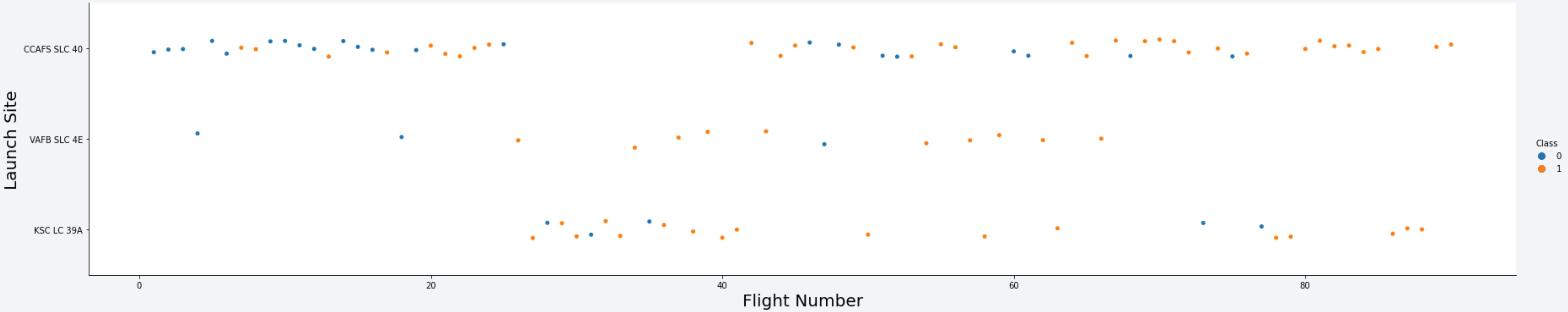
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results



Section 2

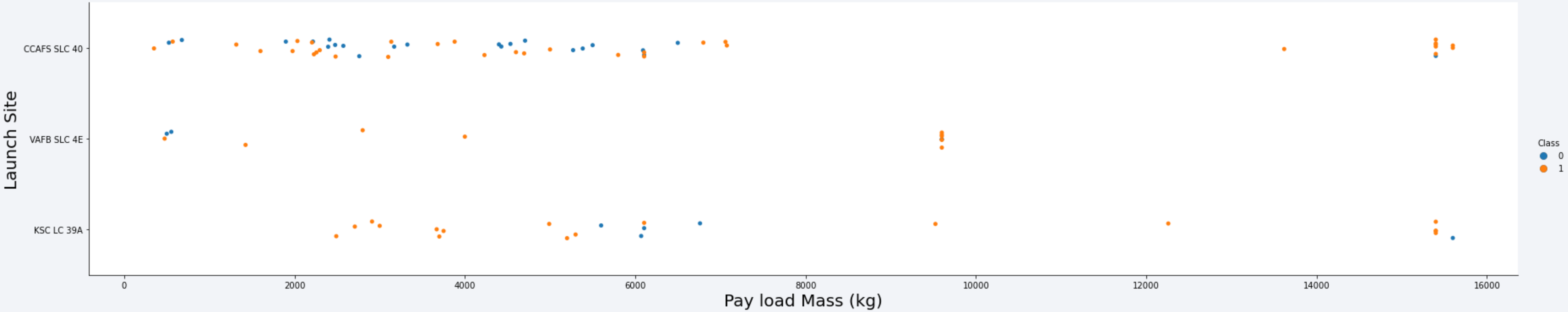
Insights drawn from EDA

Flight Number vs. Launch Site



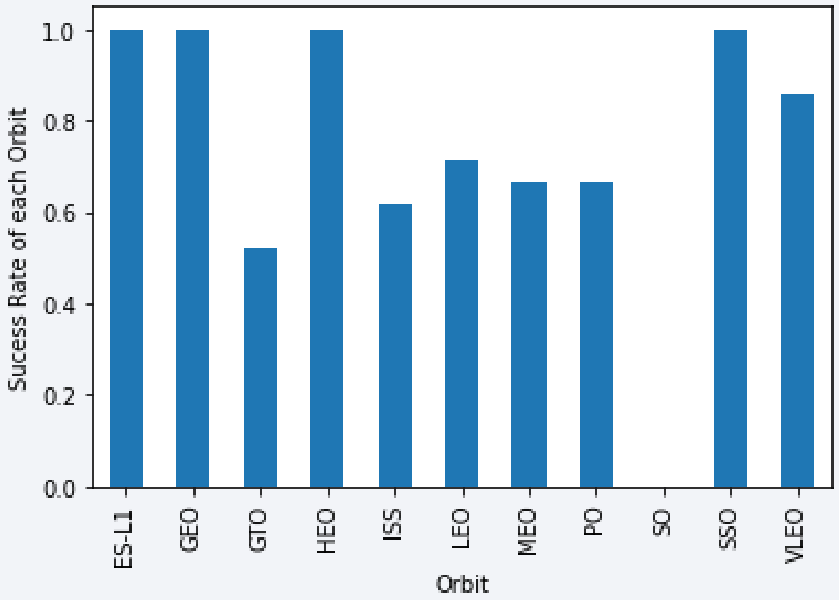
- This scatter plot shows that the success rate is increasing with increasing flights with the exception of site CCAFS SLC40.
- Flight number 25 looks to be the success rate increase for the launch sites KSC LC 39A and VAFB SLC 4E. Whereas for site CCAFS SLC40 the success rate drops relative to those below 25.

Payload vs. Launch Site



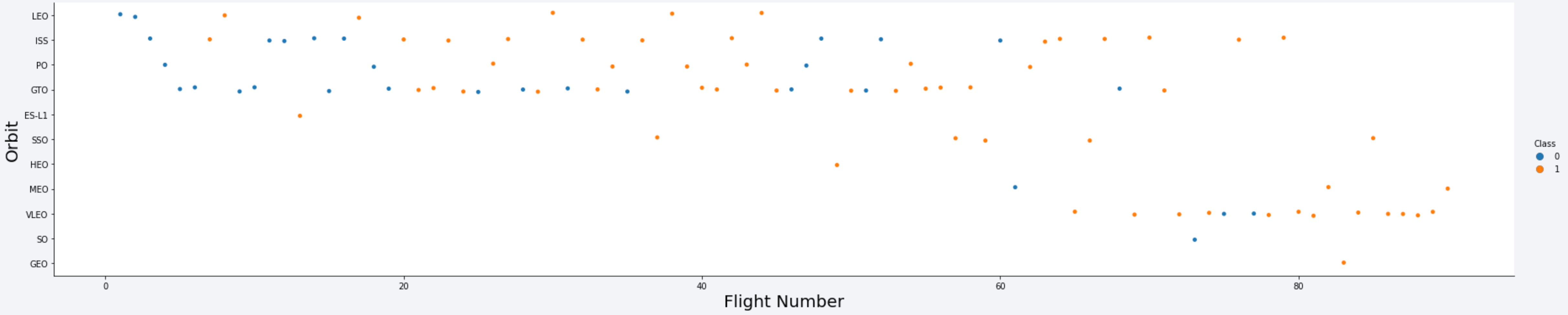
- We can observe that for payload mass less than 7000 Kg, the site CCAFS SLC40 has roughly equal probability of success to failure.
- Whereas for payload mass over 7000 Kg, in all the three launch sites we observe a significant increase in the success rate.

Success Rate vs. Orbit Type



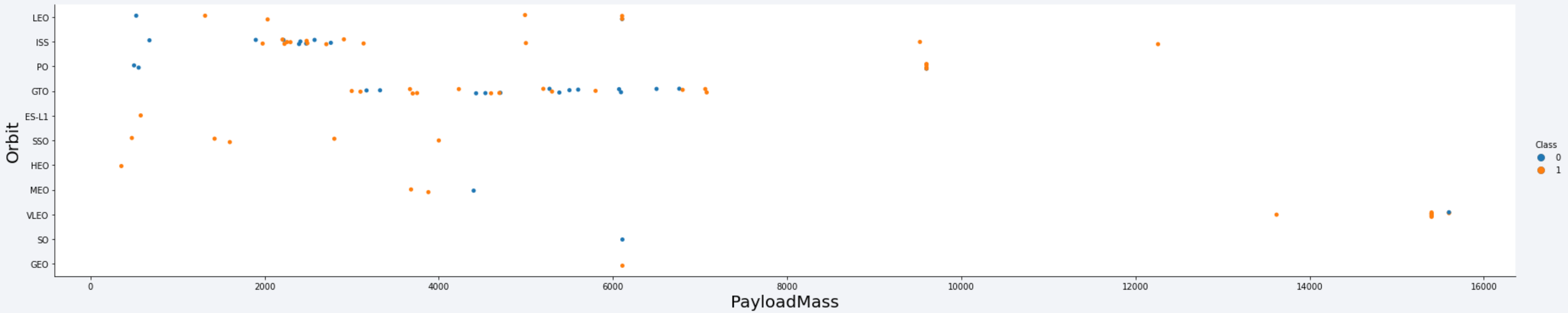
- We can observe from this bar chart that, with the exception of orbit SO which has 0% success rate, all the other orbits have a success rate over 50% with orbits ES-L1, GEO, HEO and SSO, having a success rate of100%.
- However deeper analysis show that some of these orbits such as GEO, SO, HEO and ES-L1 have only 1 occurrence which explains the outlier in 0 and 100%. Thus, we need more dataset for these orbits to draw any conclusion.

Flight Number vs. Orbit Type



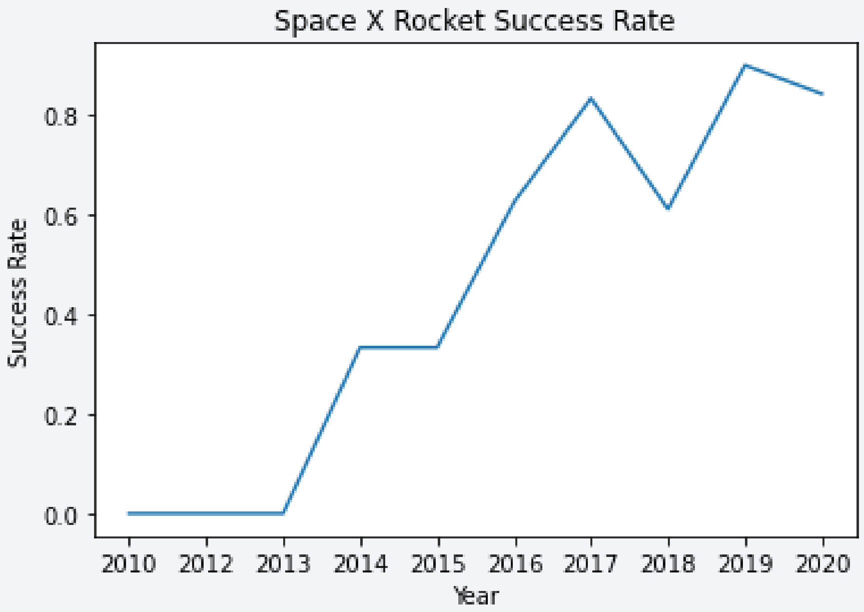
- From the scatter plot we can infer, in general, that there seems to be a correlation between success rate with the flight number (especially for LEO orbit) with the exception of GTO orbit, where there is no clear relationship between flight numbers and success rate.
- For orbits with only one data set, additional data sets needed to determine if there is a correlation or not.

Payload vs. Orbit Type



- Heavier payload has positive impact on LEO, ISS and PO orbit. However, it has negative impact on MEO and VLEO orbit.
- GTO orbit seem to depict no relation between the attributes.
- Meanwhile, again, SO, GEO and HEO orbit need more dataset to see any pattern or trend.

Launch Success Yearly Trend



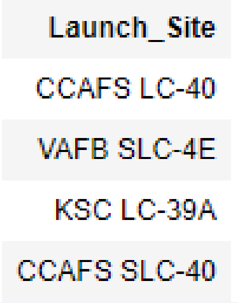
- From this line plot, we can clearly observe that the success rate has been improving since 2013 reaching maximum of 90% in 2019.

All Launch Site Names

- SQL Query:

```
SELECT DISTINCT LAUNCH_SITE FROM SPACEXTBL
```

- Results:

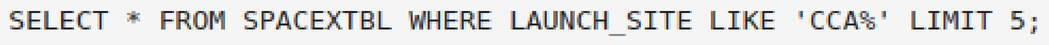


Launch_Site
CCAFS LC-40
VAFB SLC-4E
KSC LC-39A
CCAFS SLC-40

- Explanation: The use of DISTINCT in the query is to show unique launch sites.

Launch Site Names Begin with 'CCA'

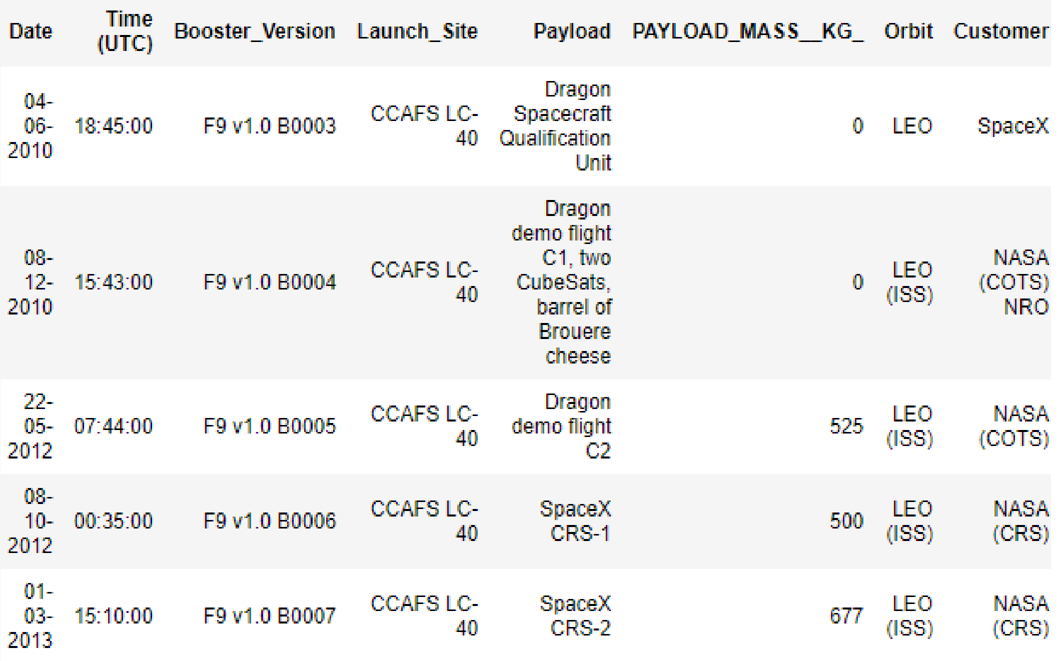
- SQL Query:



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

- Explanation: The WHERE clause followed by LIKE clause filters launch sites that contain the substring CCA. LIMIT 5 shows 5 records from filtering.

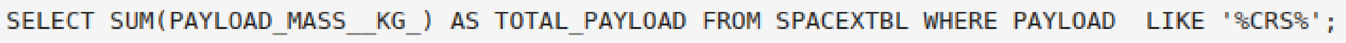
- Results:



Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer
04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX
08-12-2010	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
22-05-2012	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)
08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)
01-03-2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)

Total Payload Mass

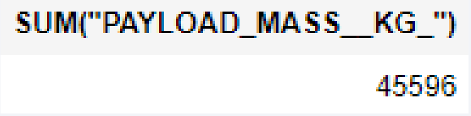
- SQL Query:



```
SELECT SUM(PAYLOAD_MASS__KG_) AS TOTAL_PAYLOAD FROM SPACEXTBL WHERE PAYLOAD  LIKE '%CRS%';
```

- Explanation: This query returns the sum of all payload masses where the customer is NASA (CRS).

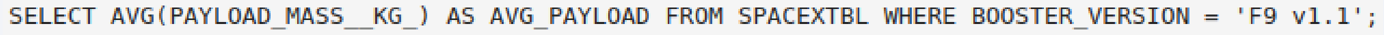
- Results:



SUM("PAYLOAD_MASS__KG_")
45596

Average Payload Mass by F9 v1.1

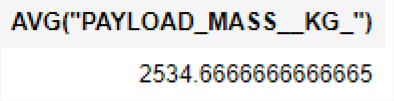
- SQL Query:



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

- Explanation: This query returns the average of all payload masses where the booster version contains the substring F9v1.1.

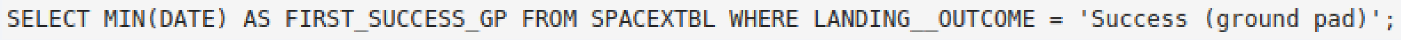
- Results:



AVG("PAYLOAD_MASS__KG_")
2534.6666666666665

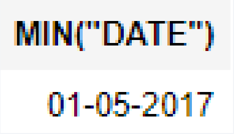
First Successful Ground Landing Date

- SQL Query:



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

- Results:

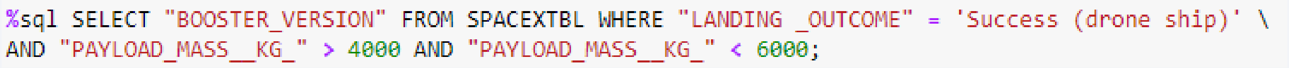


MIN("DATE")
01-05-2017

- Explanation: With this query, we select the oldest successful landing. The WHERE clause filters dataset in order to keep only records where landing was successful. With the MIN function, we select the record with the oldest date.

Successful Drone Ship Landing with Payload between 4000 and 6000

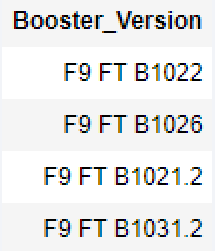
- SQL Query:



```
%sql SELECT "BOOSTER_VERSION" FROM SPACEXTBL WHERE "LANDING _OUTCOME" = 'Success (drone ship)' \
AND "PAYLOAD_MASS__KG_" > 4000 AND "PAYLOAD_MASS__KG_" < 6000;
```

- Explanation: This query returns the booster version where landing was successful and payload mass is between 4000 and 6000 kg. The WHERE and AND clauses filter the dataset.

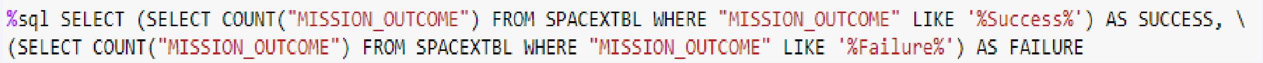
- Results:



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

Total Number of Successful and Failure Mission Outcomes

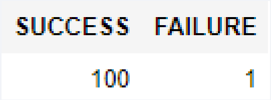
- SQL Query:



```
%sql SELECT (SELECT COUNT("MISSION_OUTCOME") FROM SPACEXTBL WHERE "MISSION_OUTCOME" LIKE '%Success%') AS SUCCESS, \
(SELECT COUNT("MISSION_OUTCOME") FROM SPACEXTBL WHERE "MISSION_OUTCOME" LIKE '%Failure%') AS FAILURE
```

- Explanation: With the first SELECT, we show the subqueries that return results. The first subquery counts the successful mission. The second subquery counts the unsuccessful mission. The WHERE clause followed by LIKE clause filters mission outcome. The COUNT function counts records filtered.

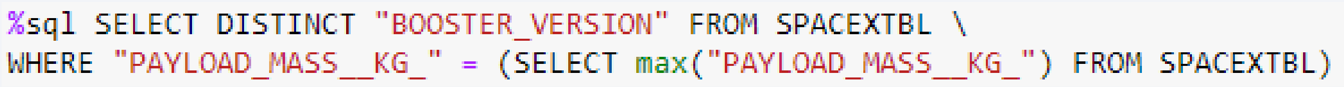
- Results:



SUCCESS	FAILURE
100	1

Boosters Carried Maximum Payload

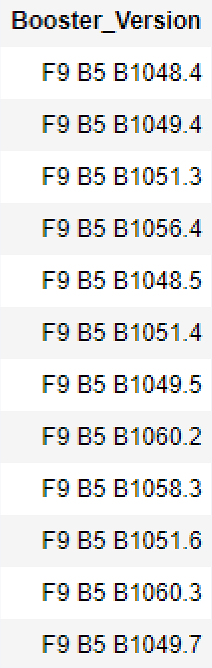
- SQL Query:



```
%sql SELECT DISTINCT "BOOSTER_VERSION" FROM SPACEXTBL \
WHERE "PAYLOAD_MASS__KG_" = (SELECT max("PAYLOAD_MASS__KG_") FROM SPACEXTBL)
```

- Explanation: We used a subquery to filter data by returning only the heaviest payload mass with MAX function. The main query uses subquery results and returns unique booster version (SELECT DISTINCT) with the heaviest payload mass.

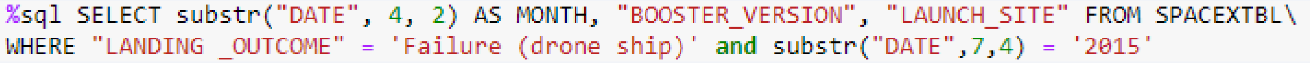
- Results:



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

2015 Launch Records

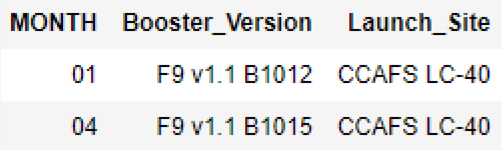
- SQL Query:



```
%sql SELECT substr("DATE", 4, 2) AS MONTH, "BOOSTER_VERSION", "LAUNCH_SITE" FROM SPACEXTBL\
WHERE "LANDING _OUTCOME" = 'Failure (drone ship)' and substr("DATE",7,4) = '2015'
```

- Explanation: This query returns month, booster version, launch site where landing was unsuccessful and landing date took place in 2015. Substr function process date in order to take month or year. Substr(DATE, 4, 2) shows month. Substr(DATE,7, 4) shows year.

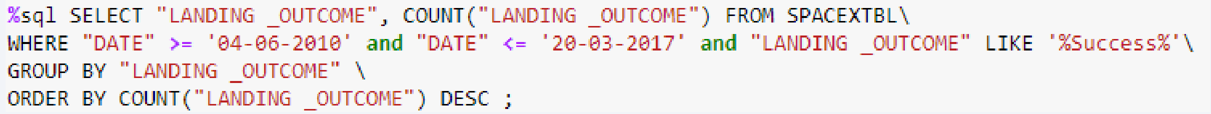
- Results:



MONTH	Booster_Version	Launch_Site
01	F9 v1.1 B1012	CCAFS LC-40
04	F9 v1.1 B1015	CCAFS LC-40

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

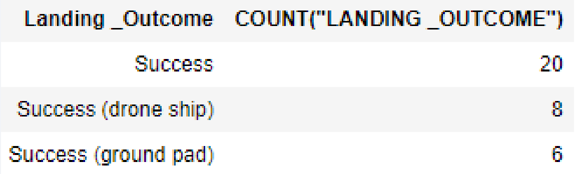
- SQL Query:



```
%sql SELECT "LANDING _OUTCOME", COUNT("LANDING _OUTCOME") FROM SPACEXTBL\
WHERE "DATE" >= '04-06-2010' and "DATE" <= '20-03-2017' and "LANDING _OUTCOME" LIKE '%Success%'\
GROUP BY "LANDING _OUTCOME" \
ORDER BY COUNT("LANDING _OUTCOME") DESC ;
```

- Explanation: This query returns landing outcomes and their count where mission were successful and dates are between 04/06/2010 and 20/03/2017. The GROUP BY clause groups results by landing outcome and ORDER BY COUNT DESC shows results in decreasing order.

- Results:

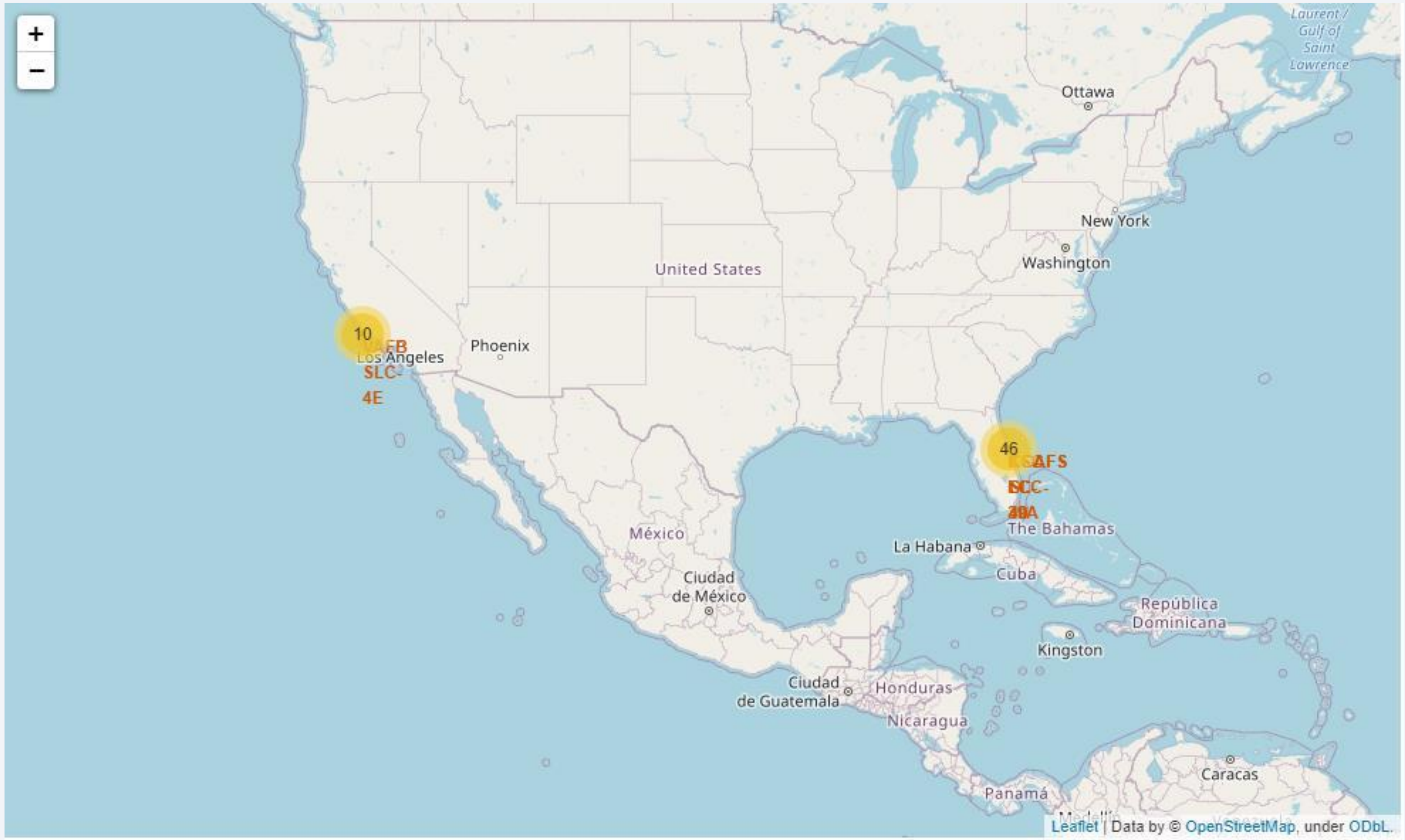


Landing _Outcome	COUNT("LANDING _OUTCOME")
Success	20
Success (drone ship)	8
Success (ground pad)	6

Section 3

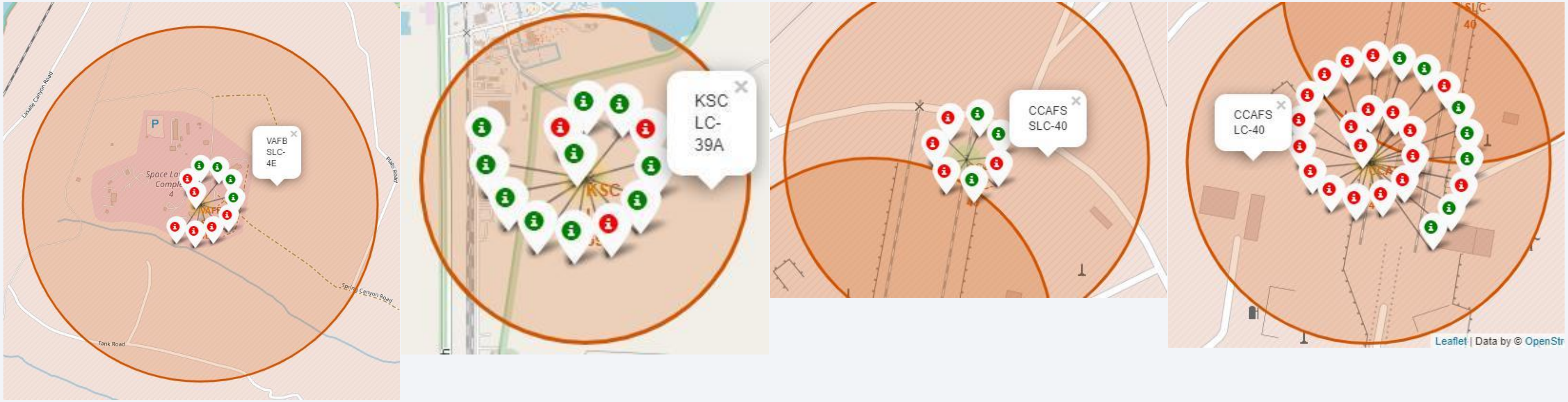
Launch Sites Proximities Analysis

Launch sites' locations



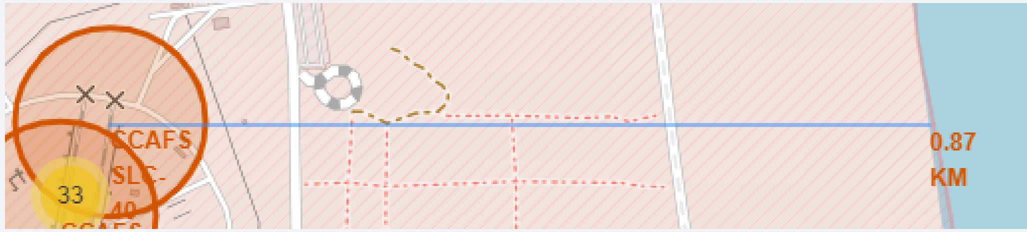
- The map shows all SpaceX launch sites are near the coast of the United States.

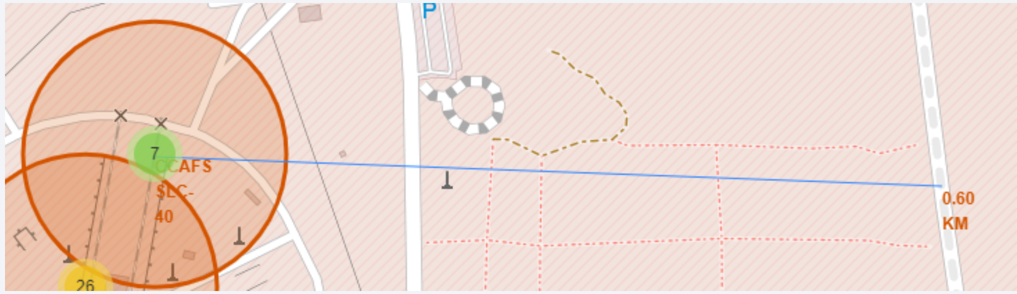
Color-labelled launch outcomes



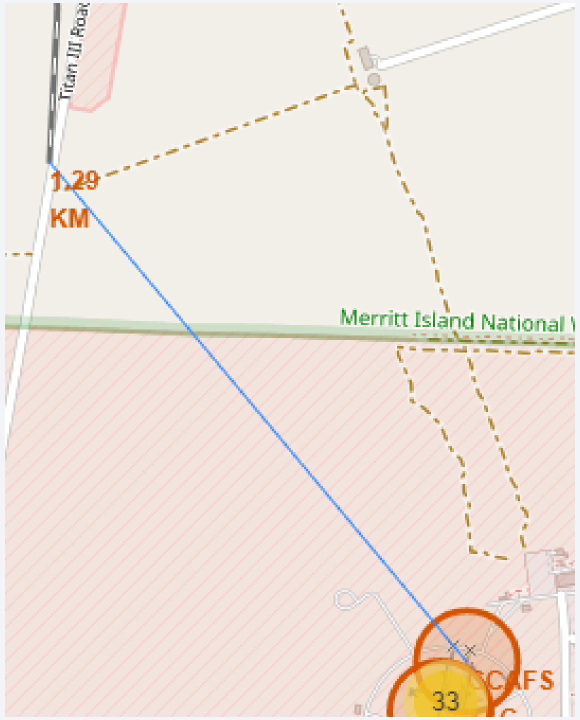
•By clicking on the marker clusters, successful landing (green) or failed landing (red) are displayed. We note that KSC LC-39A has a higher launch success rate.

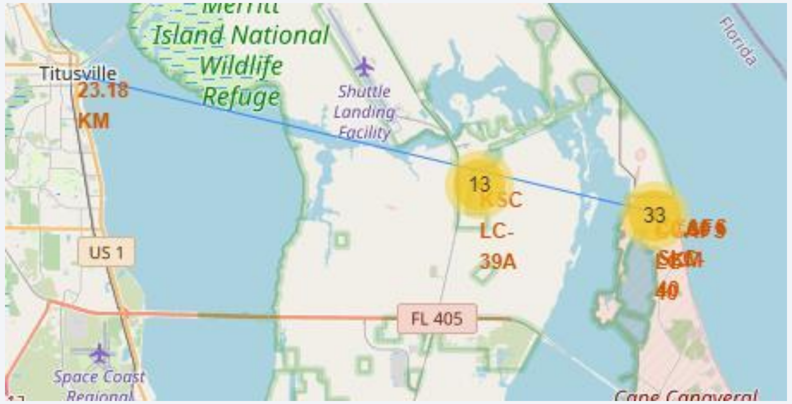
Proximities of launch sites





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**



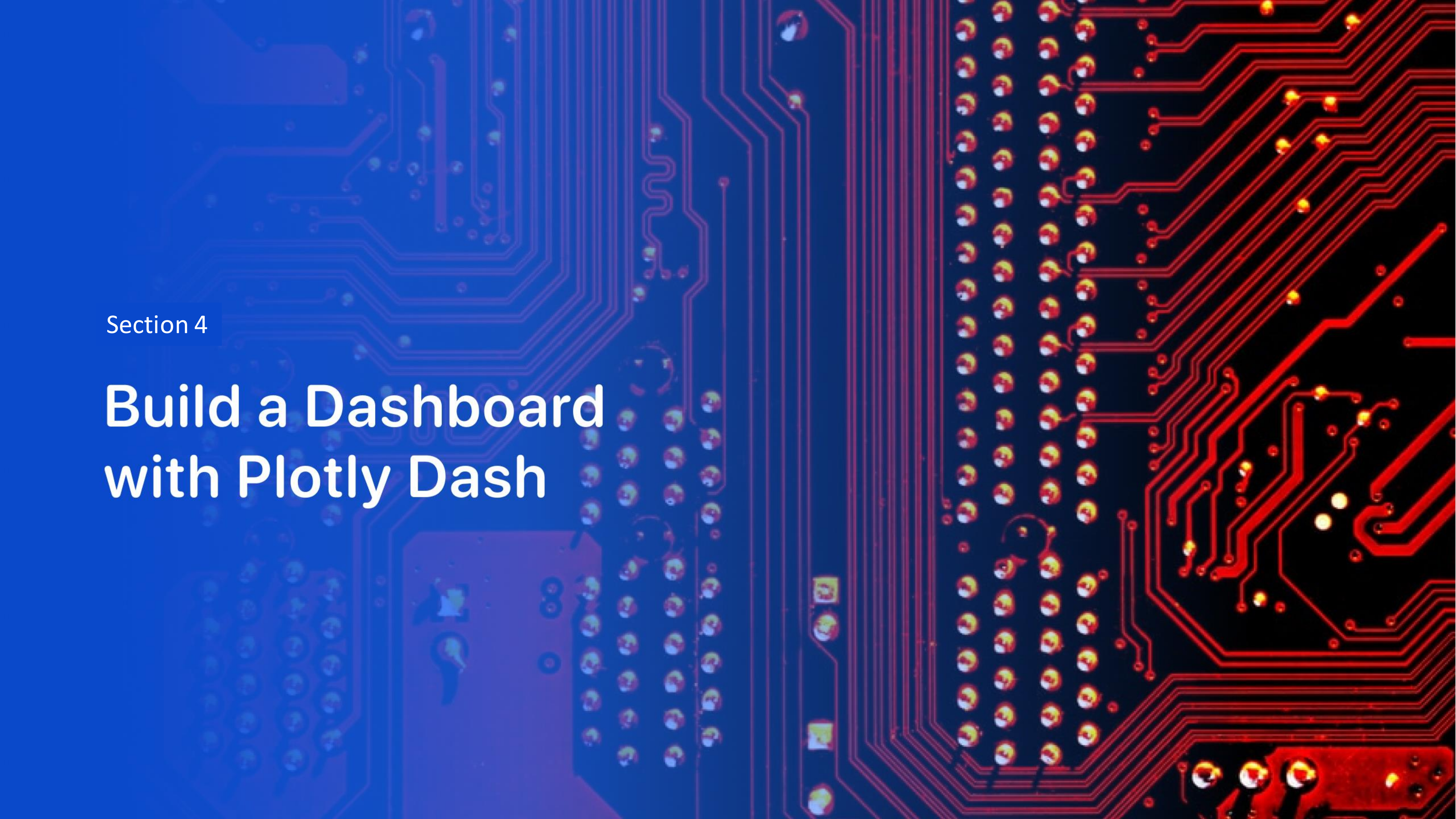


Do launch sites keep certain distance away from cities? **Yes**

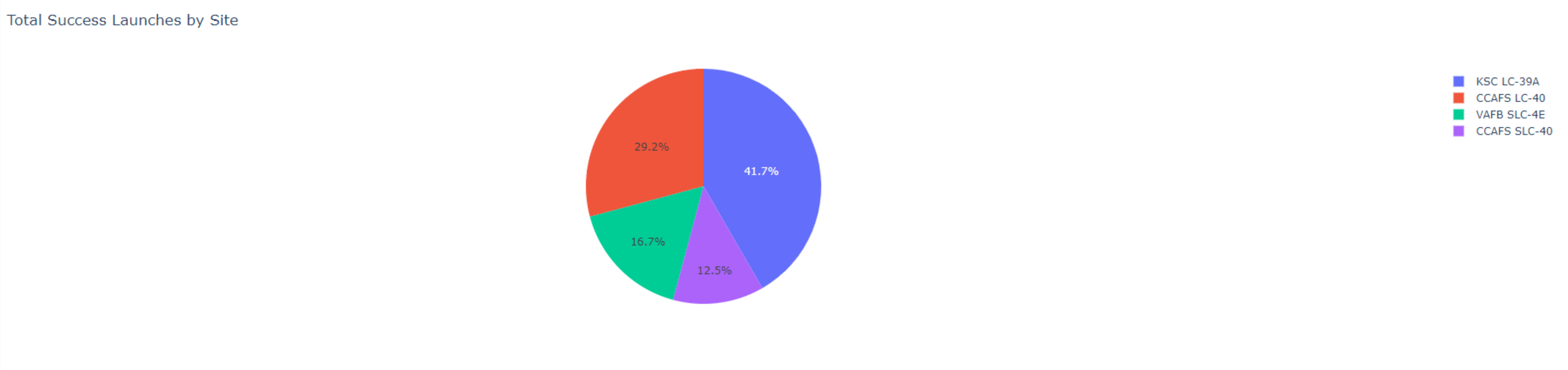
- It can be seen that the launch site is close to railways and highways for transportation of equipment or personnel, and is also close to coastline and relatively far from the cities so that launch failure does not pose a threat.

Section 4

Build a Dashboard with Plotly Dash

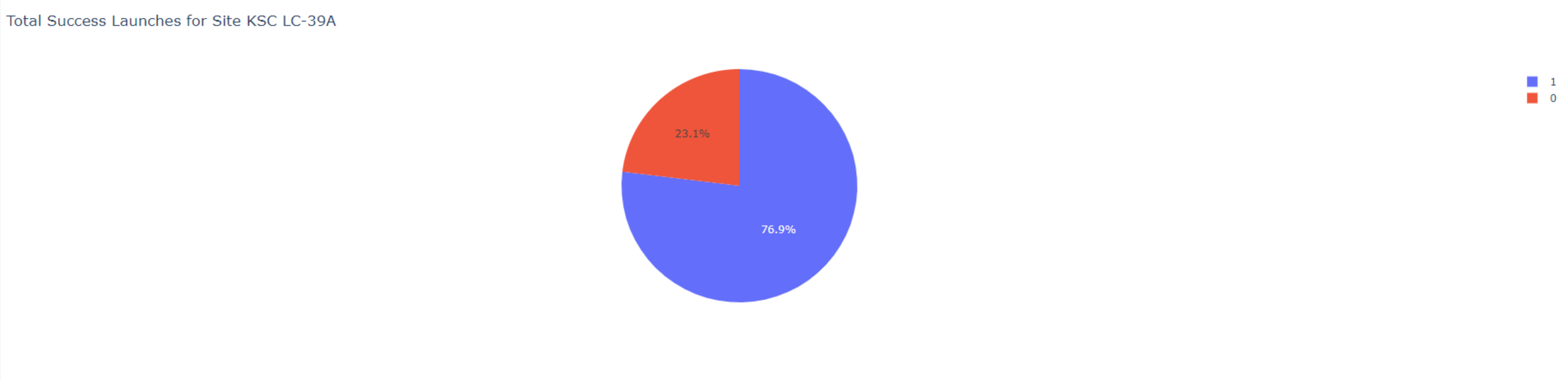


Total success launches by all sites



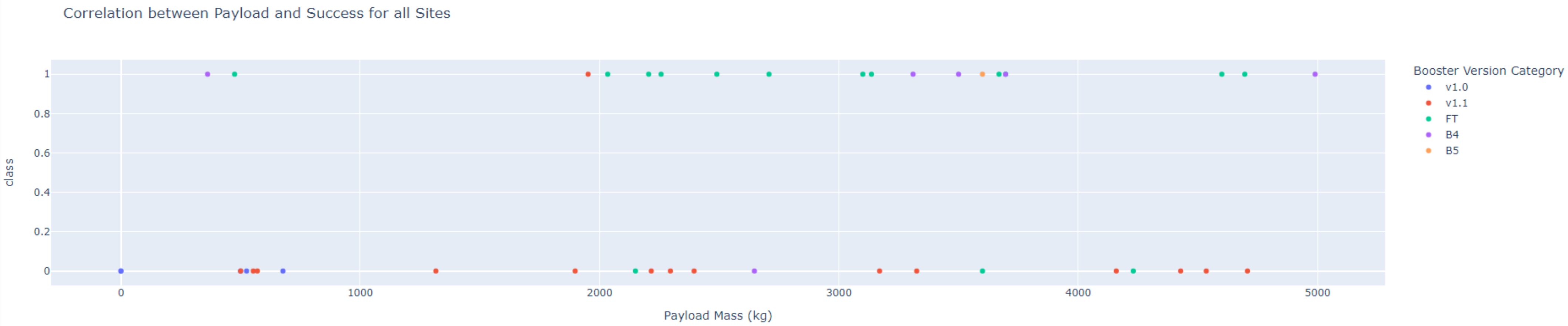
- We observe that KSC LC-39A has the best success rate of launches.

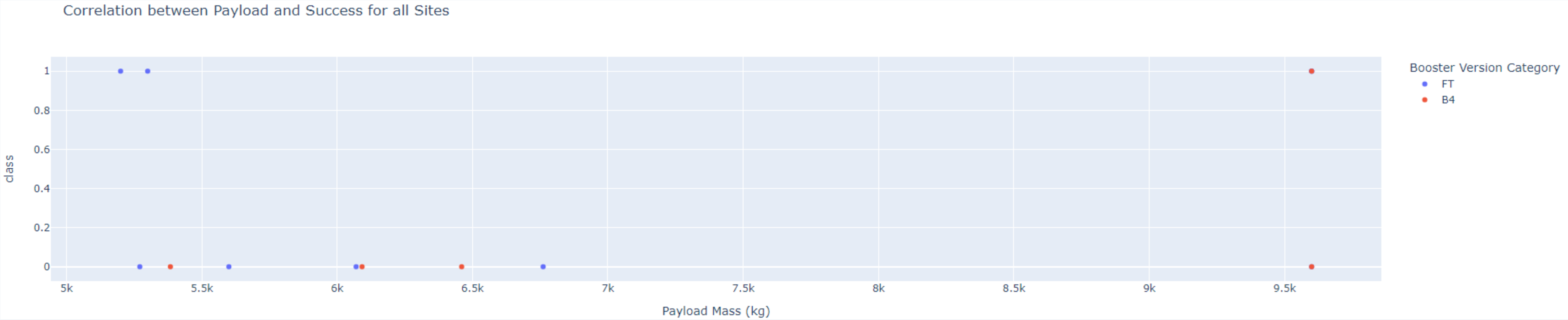
Total success launches for site KSC LC-39A



We note that the failure rate for launch site KSC LC-39A is less than a quarter.

Payload vs. Launch Outcome Scatter Plot for all sites


Correlation between Payload and Success for all Sites

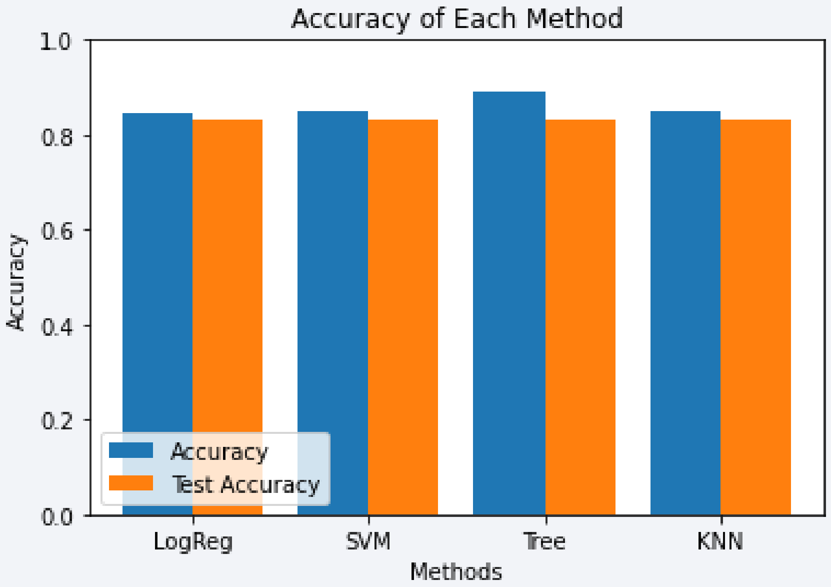

Correlation between Payload and Success for all Sites

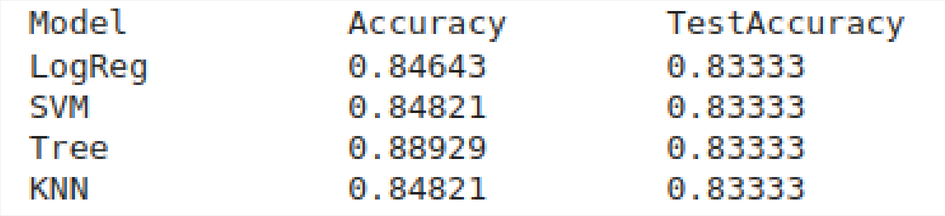
- These figures show that the launch success rate for low weighted payloads (top: 0-5000 kg) is higher than that of heavy weighted payloads (bottom: 5000-10000 kg) .

Section 5

Predictive Analysis (Classification)

Classification Accuracy

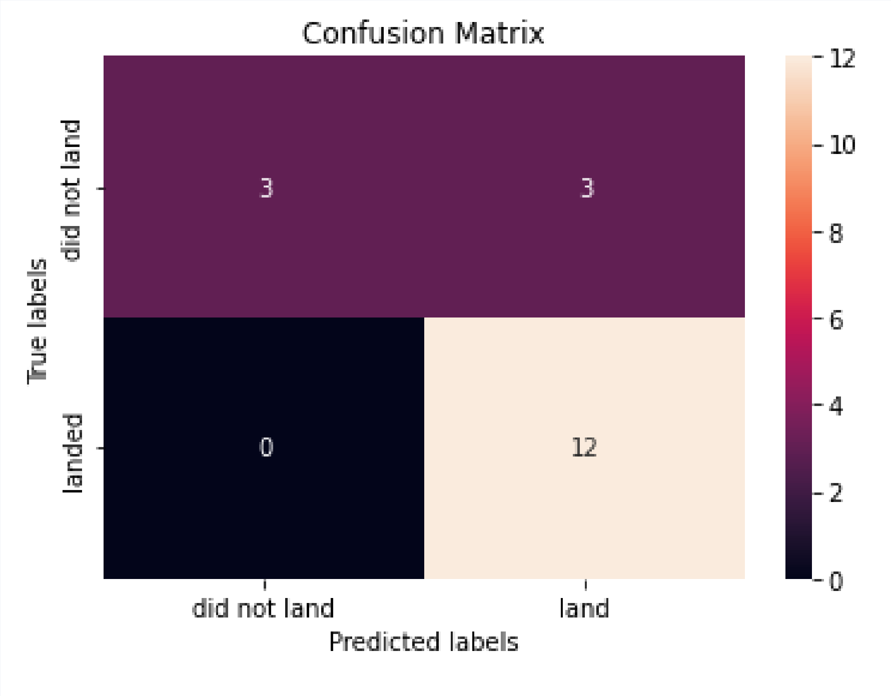




```
Model           Accuracy        TestAccuracy
LogReg          0.84643         0.83333
SVM             0.84821         0.83333
Tree            0.88929         0.83333
KNN             0.84821         0.83333
```

- For accuracy test, all models performed similarly.
- It should be noted that the test size were limited to only 18. Therefore, more data needed to determine the optimal model.
- However, with the available results we have at hand, we would take the decision tree model.

Confusion Matrix



- The confusion matrix is identical 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 failure. But there were also 3 predictions that said successful landings when the true label was failure (false positive).
- Overall, these models predict successful landings.

Conclusions

- The success of a mission can be explained by several factors such as the launch site, the orbit and especially the number of previous launches. Indeed, we can assume that there has been a gain in knowledge between launches that allowed to go from a launch failure to a success.
- Orbital types SSO, HEO, GEO, and ES-L1 have the highest success rate (100%).
- Depending on the orbits, the payload mass can be a criterion to take into account for the success of a mission. Some orbits require a light or heavy payload mass. But generally low weighted payloads perform better than the heavy weighted payloads.
- With the current data, we cannot explain why some launch sites are better than others (KSC LC-39A is the best launch site). To get an answer to this problem, we could obtain atmospheric or other relevant data.
- The launch success rate of low weighted payloads is higher than that of heavy weighted payloads.
- For this dataset, we choose the Decision Tree Algorithm as the best model even if the test accuracy between all the models used is identical. We choose Decision Tree Algorithm because it has a better train accuracy.

Thank you!