IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Valdon Vitija
01/01/2023

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection via API, Web Scraping
  - Exploratory Data Analysis (EDA) with Data Visualization
  - EDA with SQL
  - Interactive Map with Folium
  - Dashboardswith Plotly Dash
  - Predictive Analysis
- Summary of all results
  - Exploratory Data Analysis results
  - Interactive mapsand dashboard
  - Predictive results

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 or failed landing ?
  - What are the effects of each relationship of the rocket variables on the success or failure of a landing ?
  - What are the conditions which will allow SpaceX to achieve the best landing success rate ?

Section 1

Methodology

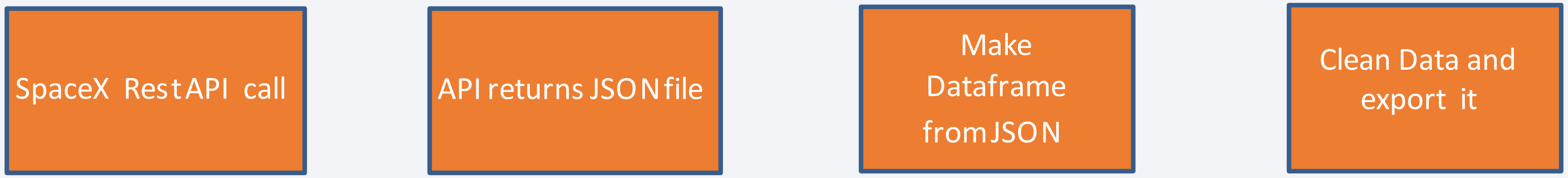
Methodology

Executive Summary

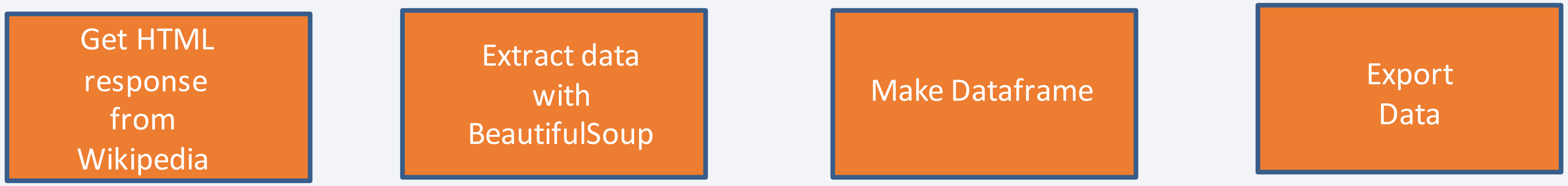
- Data collection methodology:
  - SpaceX REST API
  - Web Scrapping from Wikipedia
- Perform data wrangling
  - Dropping unnecessary columns
  - One Hot Encoding for classification models
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection

- Data had to be collected from SpaceX (REST-API) and through webscrapping (Wikipedia)
- Collected information using the provided API for rockets, launches, etc.
  - Space X (REST API) - url : "api.spacexdata.com/v4/"

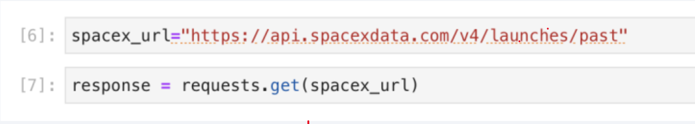


  - Webscrapping Wikipedia for information
  - URL is https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922


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

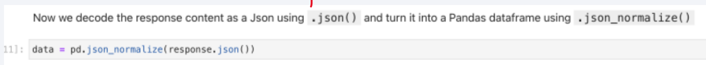
Data Collection - SpaceX API

The following flow chart represents some important steps that were being done while collecting data through the API



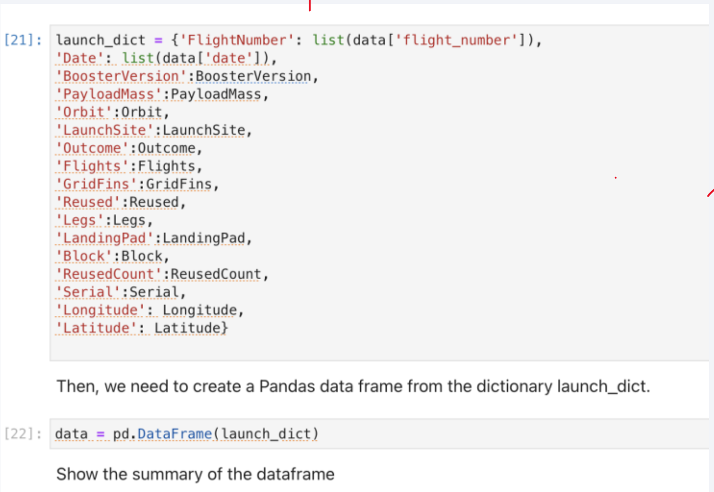
```
[6]: spacex_url="https://api.spacexdata.com/v4/launches/past"

[7]: response = requests.get(spacex_url)
```



Now we decode the response content as a Json using `.json()` and turn it into a Pandas dataframe using `.json_normalize()`

```
11]: data = pd.json_normalize(response.json())
```

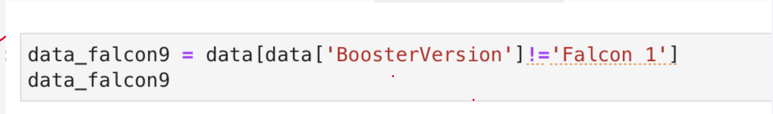


```
[21]: launch_dict = {'FlightNumber': list(data['flight_number']),
      'Date': list(data['date']),
      'BoosterVersion':BoosterVersion,
      'PayloadMass':PayloadMass,
      'Orbit':Orbit,
      'LaunchSite':LaunchSite,
      'Outcome':Outcome,
      'Flights':Flights,
      'GridFins':GridFins,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

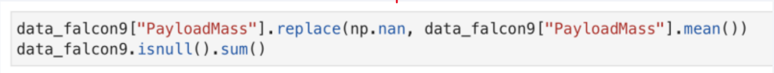
Then, we need to create a Pandas data frame from the dictionary launch_dict.

```
[22]: data = pd.DataFrame(launch_dict)
```

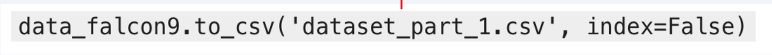
Show the summary of the dataframe



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



```
data_falcon9["PayloadMass"].replace(np.nan, data_falcon9["PayloadMass"].mean())
data_falcon9.isnull().sum()
```

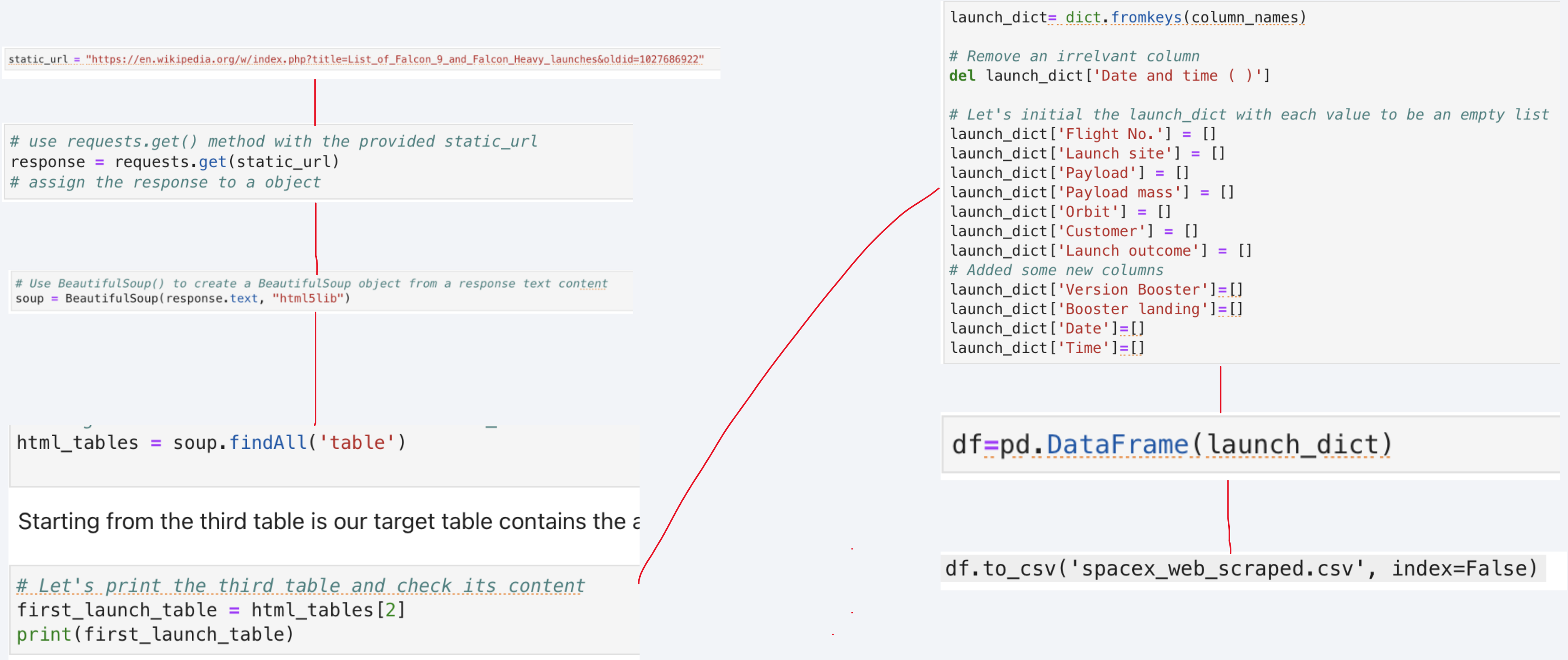


```
data_falcon9.to_csv('dataset_part_1.csv', index=False)
```

GitHub

Data Collection - Scraping

The following flow chart represents some important steps that were being done while scrapping data from wikipedia



```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"
```

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

```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(response.text, "html5lib")
```

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

Starting from the third table is our target table contains the a

```
# Let's print the third table and check its content
first_launch_table = html_tables[2]
print(first_launch_table)
```

```
launch_dict= dict.fromkeys(column_names)

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

# Let's initial the launch_dict with each value to be an empty list
launch_dict['Flight No.'] = []
launch_dict['Launch site'] = []
launch_dict['Payload'] = []
launch_dict['Payload mass'] = []
launch_dict['Orbit'] = []
launch_dict['Customer'] = []
launch_dict['Launch outcome'] = []
# Added some new columns
launch_dict['Version Booster']=[]
launch_dict['Booster landing']=[]
launch_dict['Date']=[]
launch_dict['Time']=[]
```

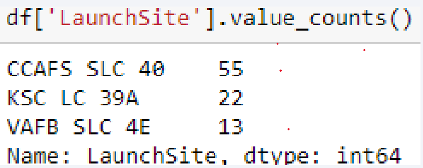
```
df=pd.DataFrame(launch_dict)
```

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

GitHub

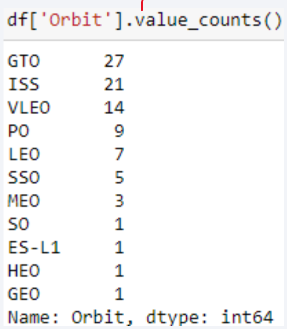
Data Wrangling

- In the dataset, there are several cases where the booster did not land successully.
  - True Ocean, True RTLS, True ASDS means the mission has been successful.
  - False Ocean, False RTLS, False ASDS means the mission was a failure.
- Categorical variables need to be converted into continuous variables.



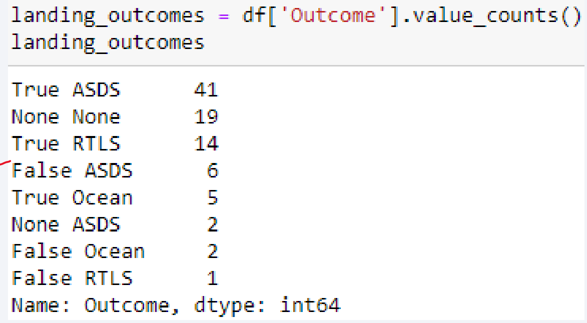
```
df['LaunchSite'].value_counts()

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



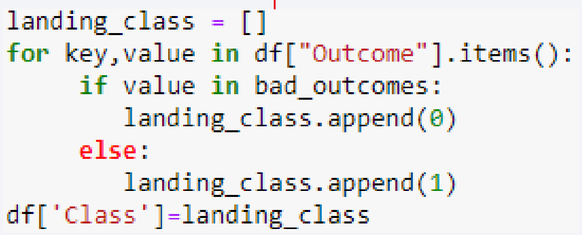
```
df['Orbit'].value_counts()

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

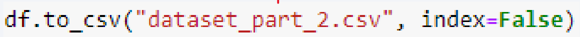


```
landing_outcomes = df['Outcome'].value_counts()
landing_outcomes

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



```
landing_class = []
for key,value in df["Outcome"].items():
     if value in bad_outcomes:
        landing_class.append(0)
     else:
        landing_class.append(1)
df['Class']=landing_class
```



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

GitHub

EDA with Data Visualization

- Scatter Graphs
  - Flight Number vs. Payload Mass
  - Flight Number vs. Launch Site
  - Payload vs. Launch Site
  - Orbit vs. Flight Number
  - Payload vs. Orbit Type
  - Orbit vs. Payload Mass

Scatter plots show relationship between variables. This relationship is called the correlation.

- Bar Graph
  - Success rate vs. Orbit

Bar graphs show the relationship between numeric and categoric variables.

- Line Graph
  - Success rate vs. Year

Line graphs show data variables and their trends. Line graphs can help to show global behavior and make prediction for unseen data.

GitHub

EDA with SQL

- Show the unique launch sites used in the space mission.
- Display 5 records where the launch site name begins with "CCA".
- Display the total payload mass carried by NASA (CRS) boosters.
- Show the average payload mass carried by booster version F9 v1.1.
- List the date of the first successful ground pad landing outcome.
- Show the names of boosters with success in drone ship landings and payload mass between 4000 and 6000.
- Display the total number of successful and failed mission outcomes.
- List the booster versions that have carried the maximum payload mass.
- Display records showing the month names, failed drone ship landing outcomes, booster versions, and launch sites for the months in 2015.
- Rank the count of successful landing outcomes between 04-06-2010 and 20-03-2017 in descending order.

GitHub

Build an Interactive Map with Folium

- Create a Folium map object centered on the NASA Johnson Space Center in Houston, Texas.
- Add a red circle at the coordinates of the NASA Johnson Space Center, with a label showing its name (using folium.Circle and folium.map.Marker).
- Add red circles at the coordinates of each launch site, with a label showing the launch site name (using folium.Circle, folium.map.Marker, and folium.features.DivIcon).
- Use the folium.plugins.MarkerCluster to group points together in a cluster and display multiple pieces of information for the same coordinates.
- Add markers to show successful and unsuccessful landings, using green markers for successful landings and red markers for unsuccessful landings (using folium.map.Marker and folium.Icon).
- Add markers to show the distance between launch sites and key locations (such as railways, highways, coastways, and cities), and plot a line between them (using folium.map.Marker, folium.PolyLine, and folium.features.DivIcon).

The reason :

  - These objects are used to better understand the problem and data, allowing us to easily visualize all launch sites, their surroundings, and the number of successful and unsuccessful landings.

GitHub

Build a Dashboard with Plotly Dash

The Dashboard has four components: a dropdown, a pie chart, a range slider, and a scatter plot. The dropdown allows a user to choose a specific launch site or view all launch sites. The pie chart displays the total number of successful and unsuccessful launches for the chosen launch site. The range slider allows a user to select a payload mass within a fixed range. The scatter plot shows the relationship between two variables, specifically the success of a launch and the payload mass. These components are implemented using the **dash_core_components.Dropdown**, **plotly.express.pie**, **dash_core_components.RangeSlider**, and **plotly.express.scatter** libraries, respectively.

GitHub

Predictive Analysis (Classification)

•In data preparation, the first step is to load the dataset. The data is then normalized to ensure that all features are on the same scale. The data is then split into training and test sets to evaluate the model's performance.

•In model preparation, machine learning algorithms are selected and their parameters are set using GridSearchCV. The models are then trained on the training dataset.

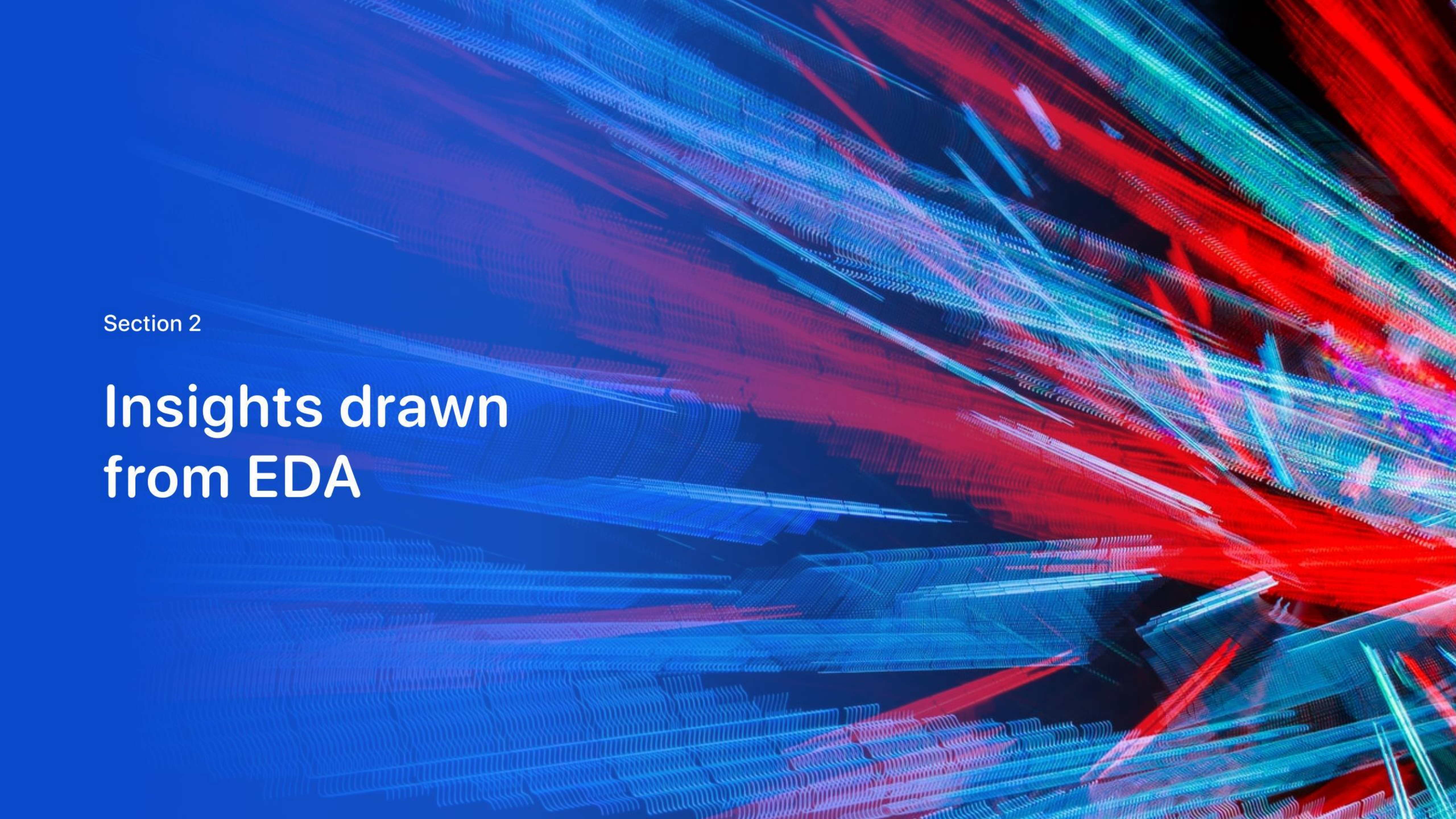
•In model evaluation, the best hyperparameters for each model are determined and the model's accuracy is computed using the test dataset. A confusion matrix is plotted to visualize the model's performance.

•In model comparison, the models are compared based on their accuracy and the model with the best accuracy is chosen.

GitHub

Results

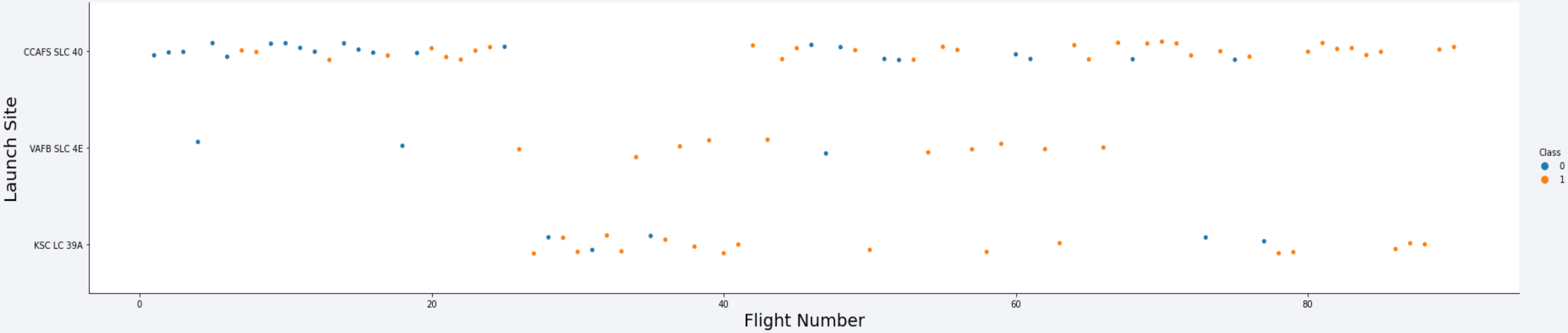
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results



Section 2

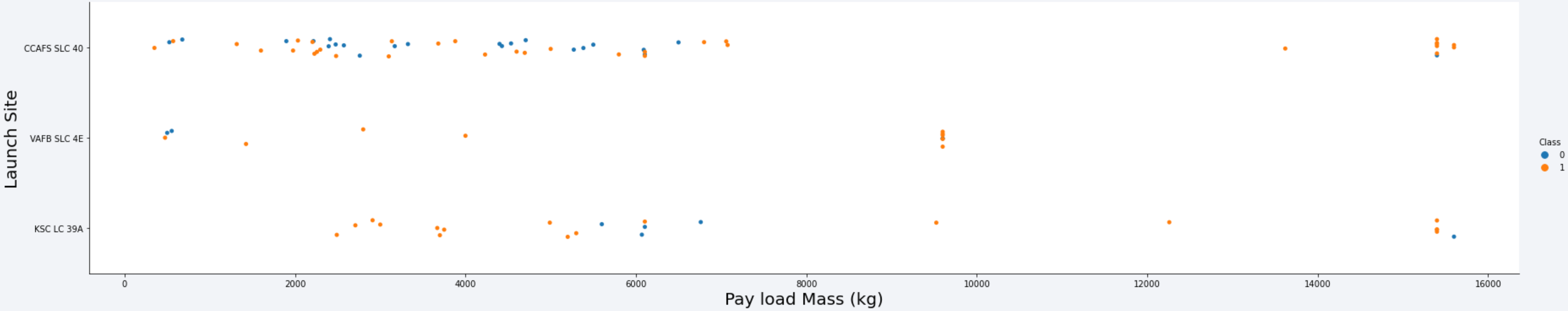
Insights drawn from EDA

Flight Number vs. Launch Site



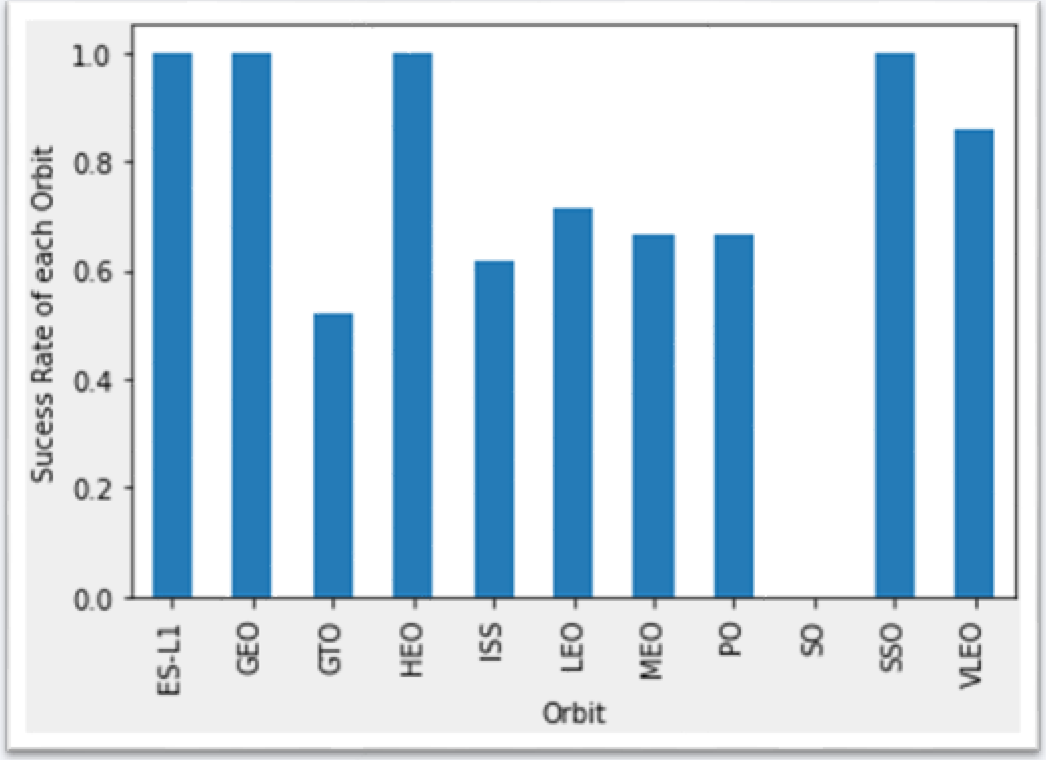
the success rate is increasing for each site.

Payload vs. Launch Site



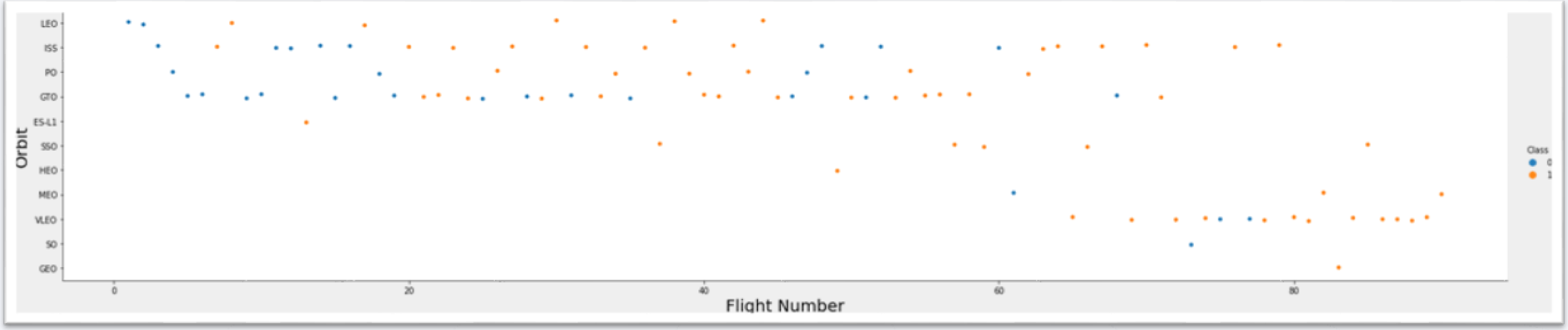
The weight of the payload can affect the success of a landing. If the payload is too heavy, it may cause the landing to fail. However, depending on the launch site, a heavier payload may be necessary for a successful landing. It is important to consider the weight of the payload when planning a launch.

Success Rate vs. Orbit Type



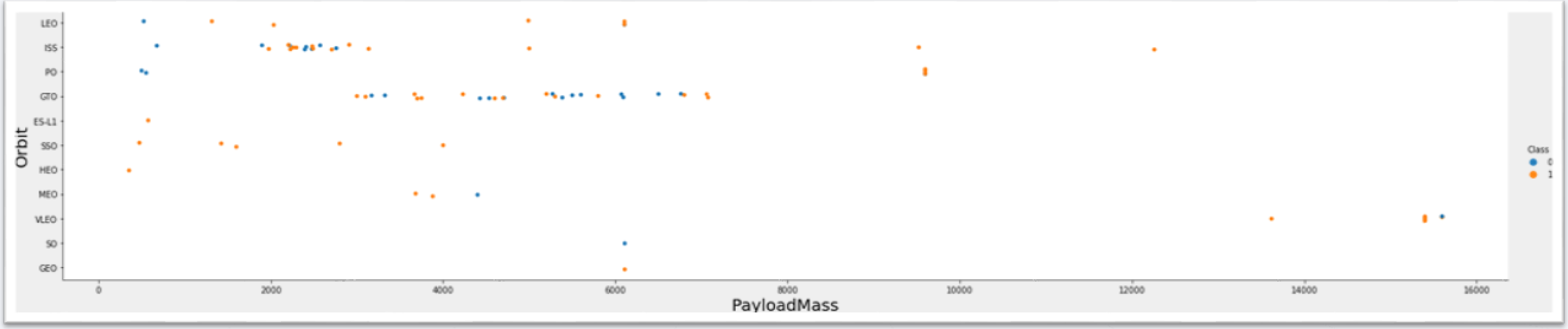
This plot shows the success rate for different orbit types. It appears that the ES-L1, GEO, HEO, and SSO orbits have the best success rate. This information may be useful for determining the most reliable orbit for a particular mission.

Flight Number vs. Orbit Type



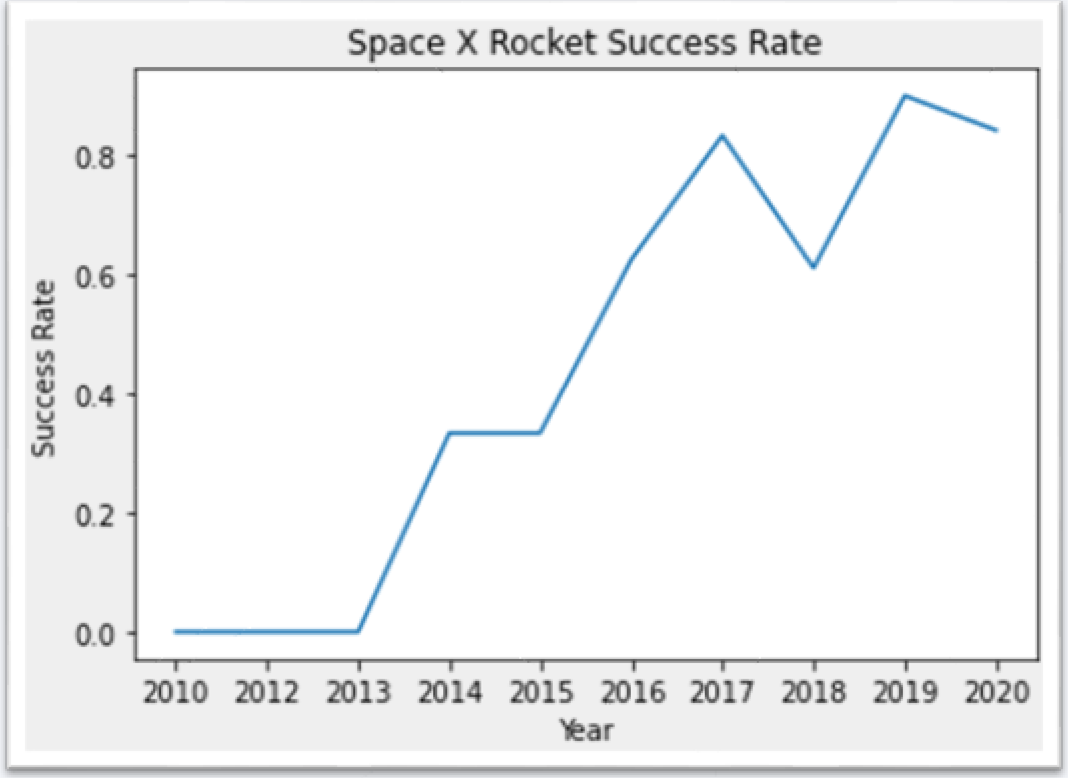
This analysis suggests that for the LEO orbit, the success rate increases with the number of flights. However, for some orbits like GTO, there is no relationship between the success rate and the number of flights. It is possible that the high success rate of orbits like SSO or HEO is due to the knowledge and experience gained from previous launches in other orbits. This suggests that experience and learning from past launches may play a role in the success rate of certain orbits.

Payload vs. Orbit Type



The weight of the payloads can have a great influence on the success rate of the launches in certain orbits. For example, heavier payloads improve the success rate for the LEO orbit. Another finding is that decreasing the payload weight for a GTO orbit improves the success of a launch.

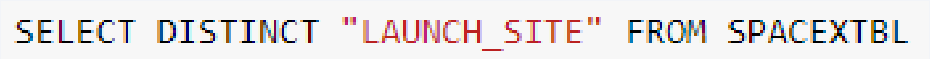
Launch Success Yearly Trend



This analysis shows that the success rate of Space X rockets has increased since 2013. This may indicate that the company has made improvements to its rockets or launch processes over time, resulting in a higher success rate.

All Launch Site Names

SQL Query

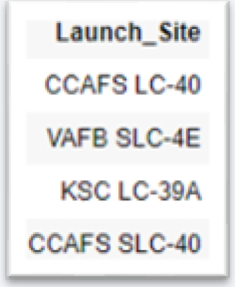


```
SELECT DISTINCT "LAUNCH_SITE" FROM SPACEXTBL
```

Explanation

The use of DISTINCT in the query allows to remove duplicate LAUNCH_SITE.

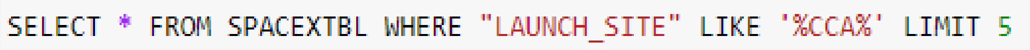
Results



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

Launch Site Names Begin with 'CCA'

SQL Query

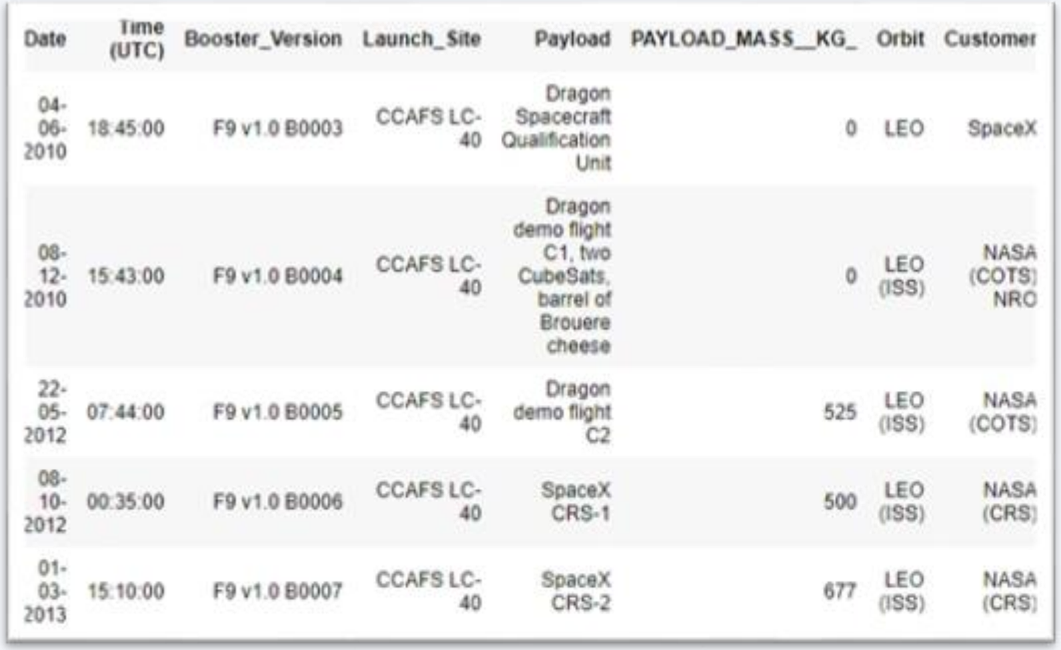


```
SELECT * FROM SPACEXTBL WHERE "LAUNCH_SITE" LIKE '%CCA%' LIMIT 5
```

Explanation

"The WHERE and LIKE clauses in this statement are used to filter the rows returned so that only those with **launch_sites** containing the substring **CCA** are included. The LIMIT clause then limits the number of rows returned to 5, displaying only these 5 records after filtering."

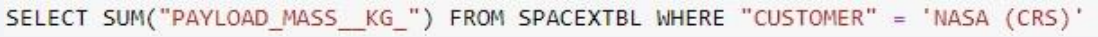
Results



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

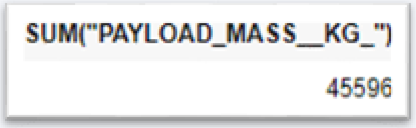


```
SELECT SUM("PAYLOAD_MASS__KG_") FROM SPACEXTBL WHERE "CUSTOMER" = 'NASA (CRS)'
```

Explanation

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

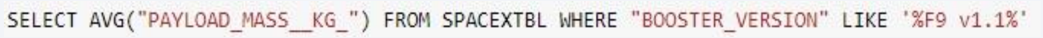
Results



SUM("PAYLOAD_MASS__KG_")
45596

Average Payload Mass by F9 v1.1

SQL Query

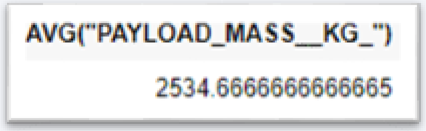


```
SELECT AVG("PAYLOAD_MASS__KG_") FROM SPACEXTBL WHERE "BOOSTER_VERSION" LIKE '%F9 v1.1%'
```

Explanation

"This query uses the WHERE clause and LIKE operator to filter the rows returned to only those with a **booster_version** containing the substring **F9 v1.1**. It then returns the average of all payload masses in these rows."

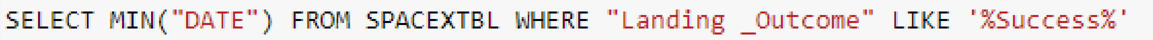
Results



AVG("PAYLOAD_MASS__KG_")
2534.6666666666665

First Successful Ground Landing Date

SQL Query

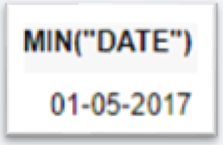


```
SELECT MIN("DATE") FROM SPACEXTBL WHERE "Landing _Outcome" LIKE '%Success%'
```

Explanation

"This query filters the dataset to include only rows where the landing was successful, using the WHERE clause. It then uses the MIN function to select the row with the oldest date, which represents the oldest successful landing."

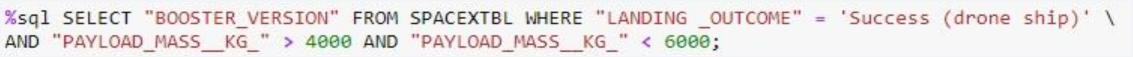
Results



MIN("DATE")
01-05-2017

Successful Drone Ship Landing with Payload between 4000 and 6000

SQL Query

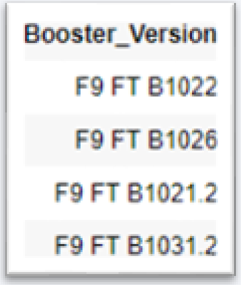


```
%sql SELECT "BOOSTER_VERSION" FROM SPACEXTBL WHERE "LANDING _OUTCOME" = 'Success (drone ship)' \
AND "PAYLOAD_MASS__KG_" > 4000 AND "PAYLOAD_MASS__KG_" < 6000;
```

Explanation

"This query uses the WHERE and AND clauses to filter the dataset to include only rows where the landing was successful and the payload mass is between 4000 and 6000 kg. It then returns the **booster_version** for these rows."

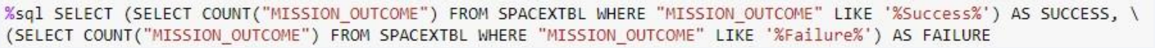
Results



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

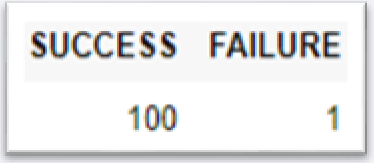


```
%sql SELECT (SELECT COUNT("MISSION_OUTCOME") FROM SPACEXTBL WHERE "MISSION_OUTCOME" LIKE '%Success%') AS SUCCESS, \
(SELECT COUNT("MISSION_OUTCOME") FROM SPACEXTBL WHERE "MISSION_OUTCOME" LIKE '%Failure%') AS FAILURE
```

Explanation

"This query uses two subqueries, both of which use the WHERE clause and LIKE operator to filter the rows based on the **mission_outcome** column. The first subquery counts the number of rows where the outcome is successful, while the second subquery counts the number of rows where the outcome is unsuccessful. The outer SELECT statement displays the results of these subqueries, which are the counts of successful and unsuccessful missions."

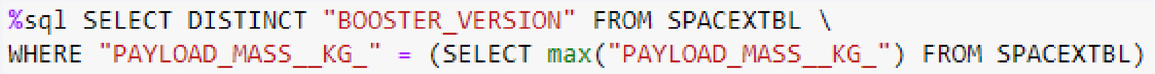
Results



SUCCESS	FAILURE
100	1

Boosters Carried Maximum Payload

SQL Query

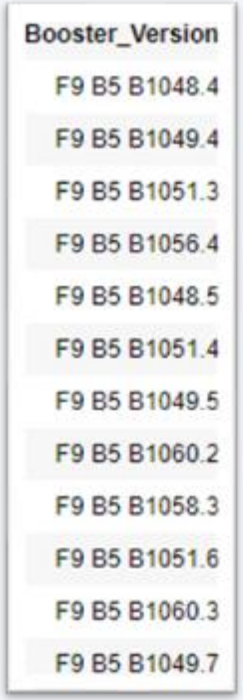


```
%sql SELECT DISTINCT "BOOSTER_VERSION" FROM SPACEXTBL \
WHERE "PAYLOAD_MASS__KG_" = (SELECT max("PAYLOAD_MASS__KG_") FROM SPACEXTBL)
```

Explanation

"This query uses a subquery to filter the data by returning the heaviest payload mass using the MAX function. The main query then uses the results of the subquery and returns only unique **booster_version** values using the SELECT DISTINCT statement, along with the heaviest payload mass."

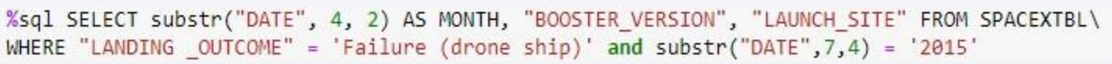
Results



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

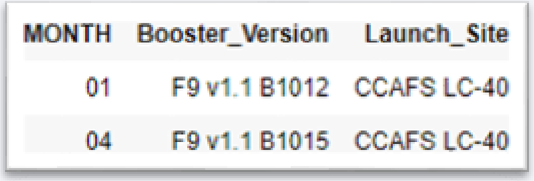


```
%sql SELECT substr("DATE", 4, 2) AS MONTH, "BOOSTER_VERSION", "LAUNCH_SITE" FROM SPACEXTBL\
WHERE "LANDING _OUTCOME" = 'Failure (drone ship)' and substr("DATE",7,4) = '2015'
```

Explanation

"This query returns the month, **booster_version**, and **launch_site** for rows where the landing was unsuccessful and the landing took place in 2015. It uses the SUBSTR function to extract the month and year from the **DATE** column. SUBSTR(DATE, 4, 2) returns the month, while SUBSTR(DATE,7, 4) returns the year."

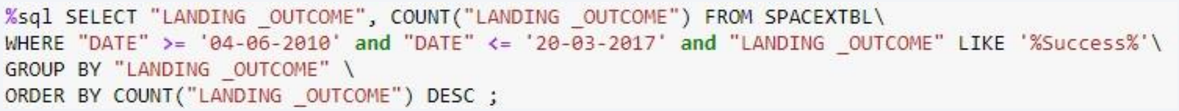
Results



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

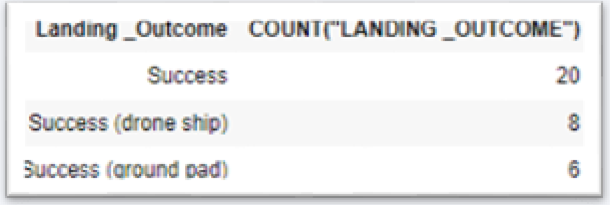


```
%sql SELECT "LANDING _OUTCOME", COUNT("LANDING _OUTCOME") FROM SPACEXTBL\
WHERE "DATE" >= '04-06-2010' and "DATE" <= '20-03-2017' and "LANDING _OUTCOME" LIKE '%Success%'\
GROUP BY "LANDING _OUTCOME" \
ORDER BY COUNT("LANDING _OUTCOME") DESC ;
```

Explanation

"This query returns the **landing_outcome** values and their counts for rows where the mission was successful and the date is between 04/06/2010 and 20/03/2017. It uses the GROUP BY clause to group the results by **landing_outcome**, and the ORDER BY clause to sort the results in decreasing order based on the count of each outcome."

Results

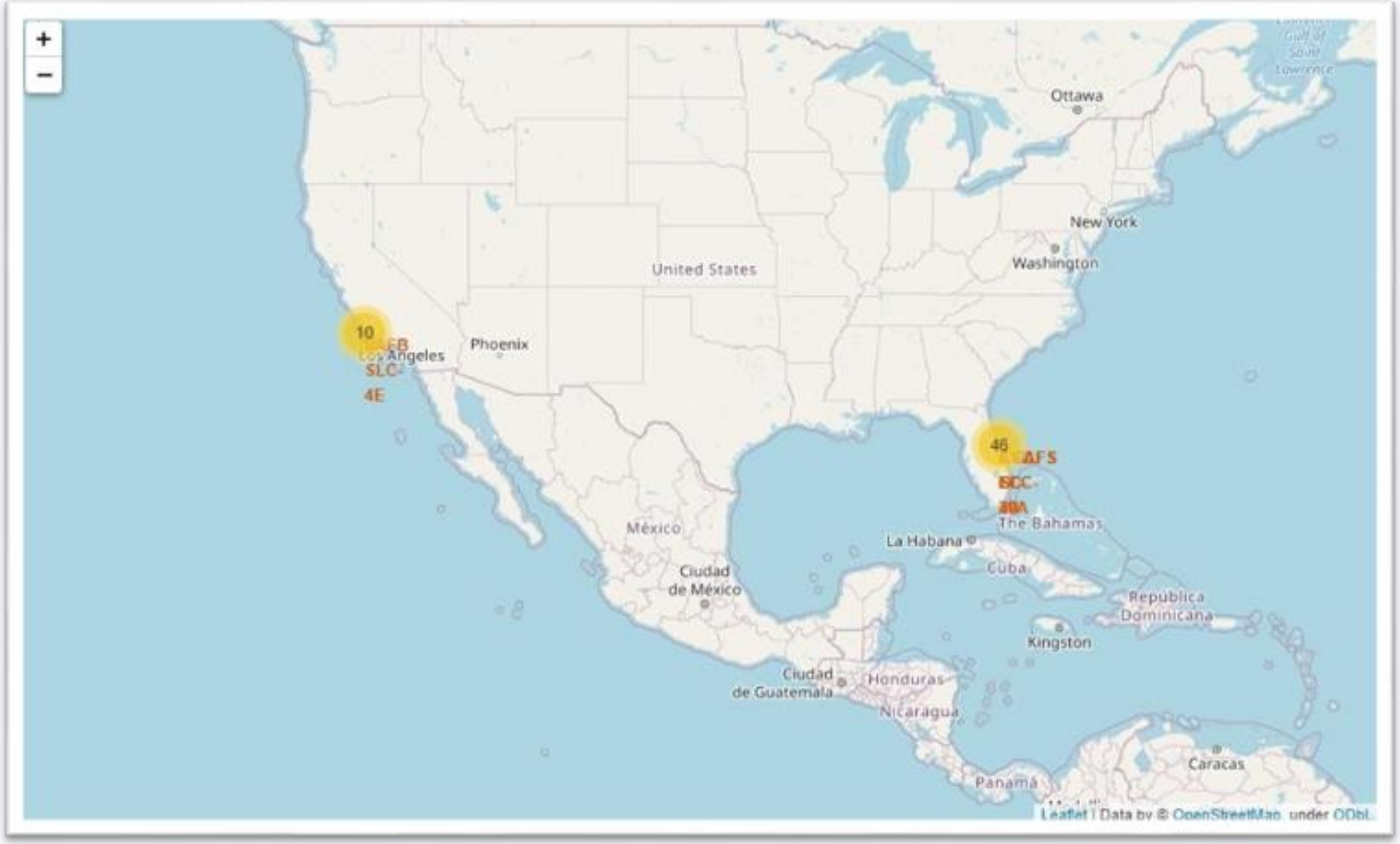


Landing _Outcome	COUNT("LANDING _OUTCOME")
Success	20
Success (drone ship)	8
Success (ground pad)	6

Section 4

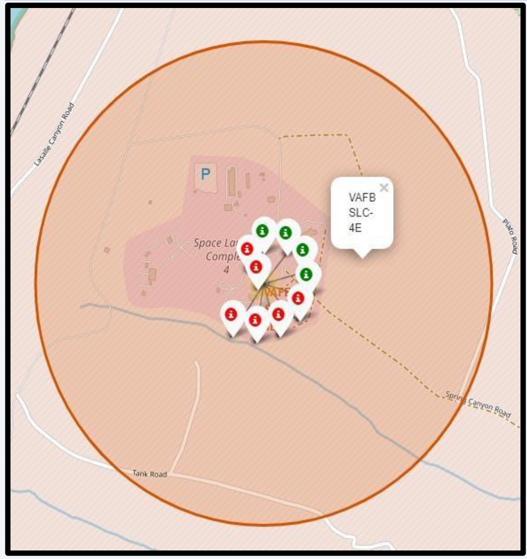
Launch Sites Proximities Analysis

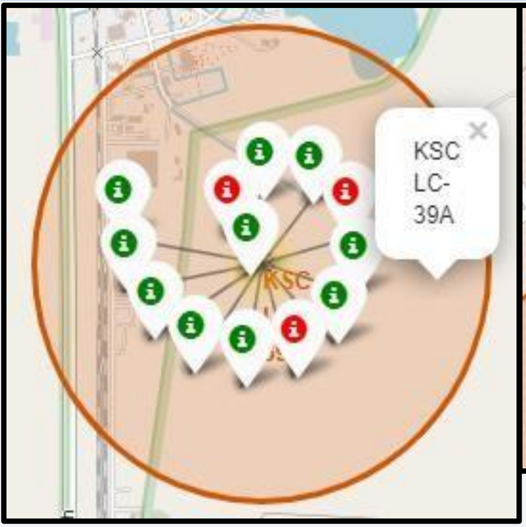
Folium map - Ground stations

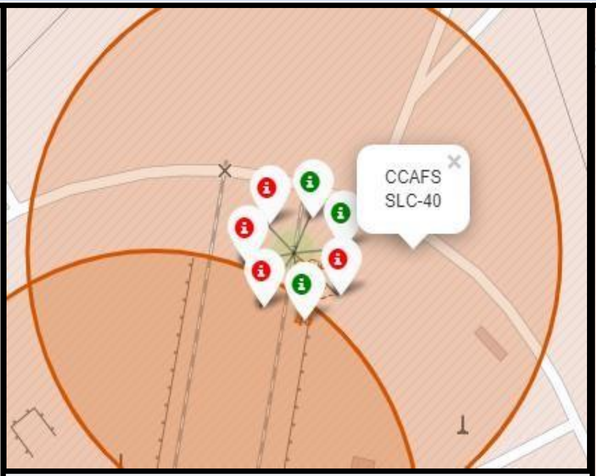


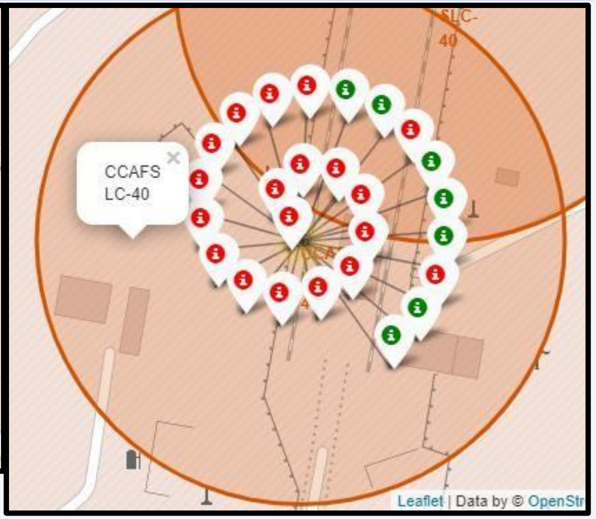
"SpaceX launch sites are located along the coast of the United States."

Folium map - Color Labeled Markers



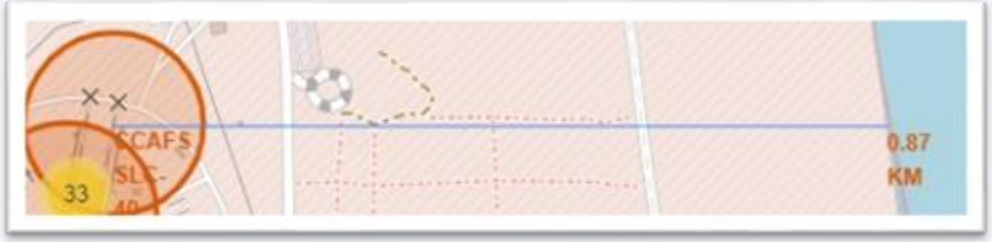


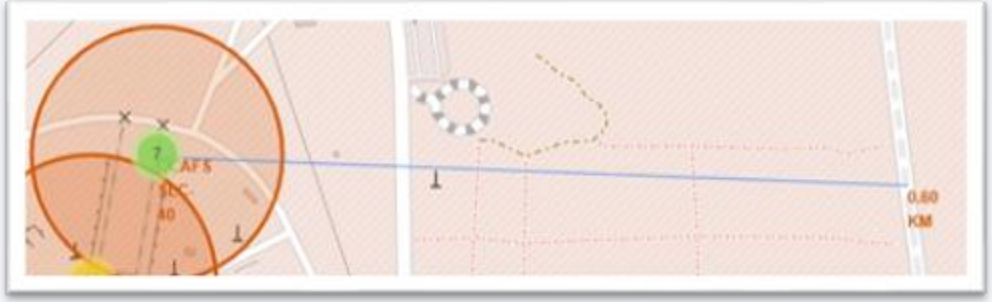


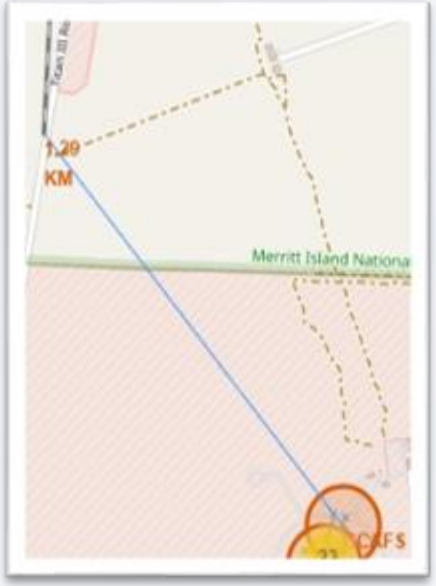


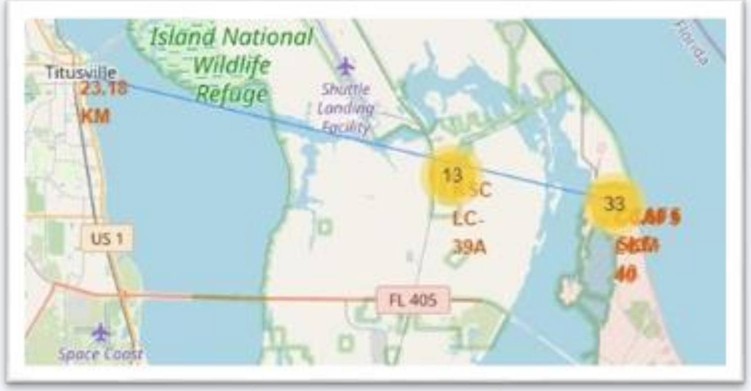
"The green markers on the map represent successful launches, while the red markers represent unsuccessful launches. It appears that KSC LC-39A has a higher launch success rate based on the distribution of these markers."

Folium Map - Distances between CCAFS SLC-40 and its proximities

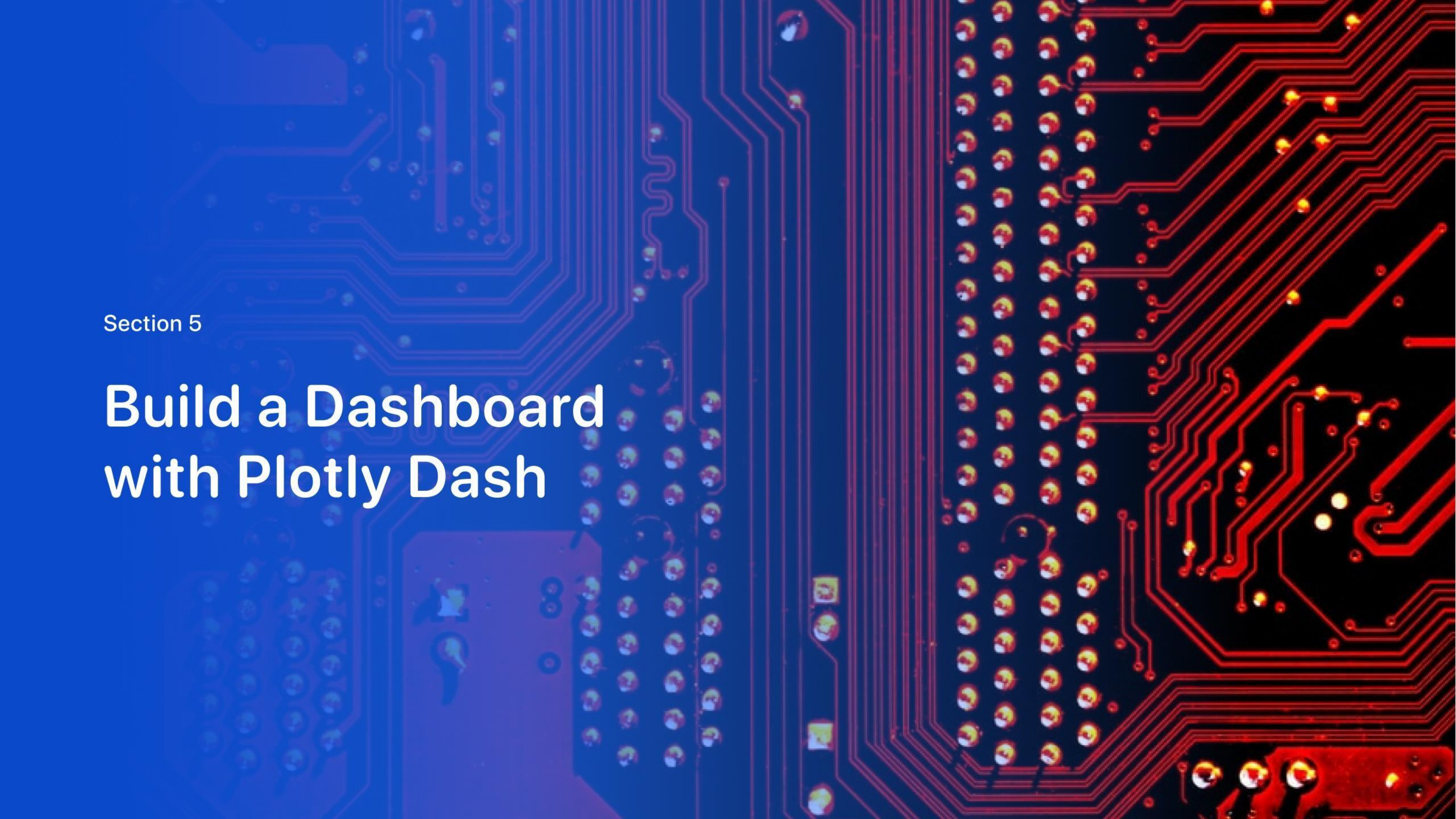








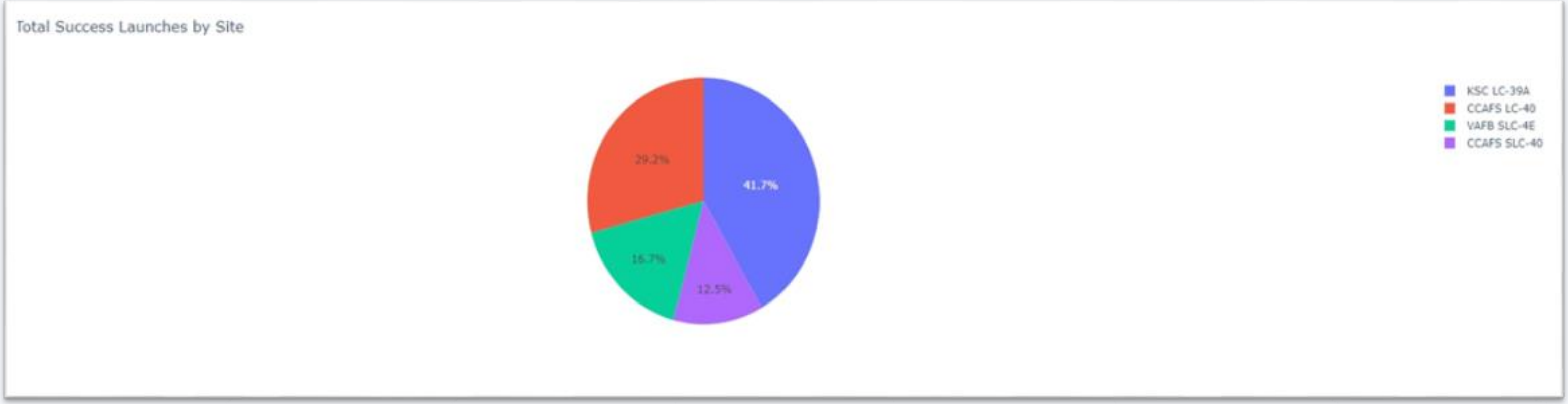
- CCAFS SLC-40 is located in close proximity to railways, meaning that it is situated relatively near to railway lines.
- CCAFS SLC-40 is located near highways, meaning that it is situated relatively close to major roads.
- CCAFS SLC-40 is situated close to the coastline, meaning that it is located relatively near to the edge of the land where it meets the sea."
- CCAFS SLC-40 is not located a significant distance away from cities. It is situated relatively close to urban areas



Section 5

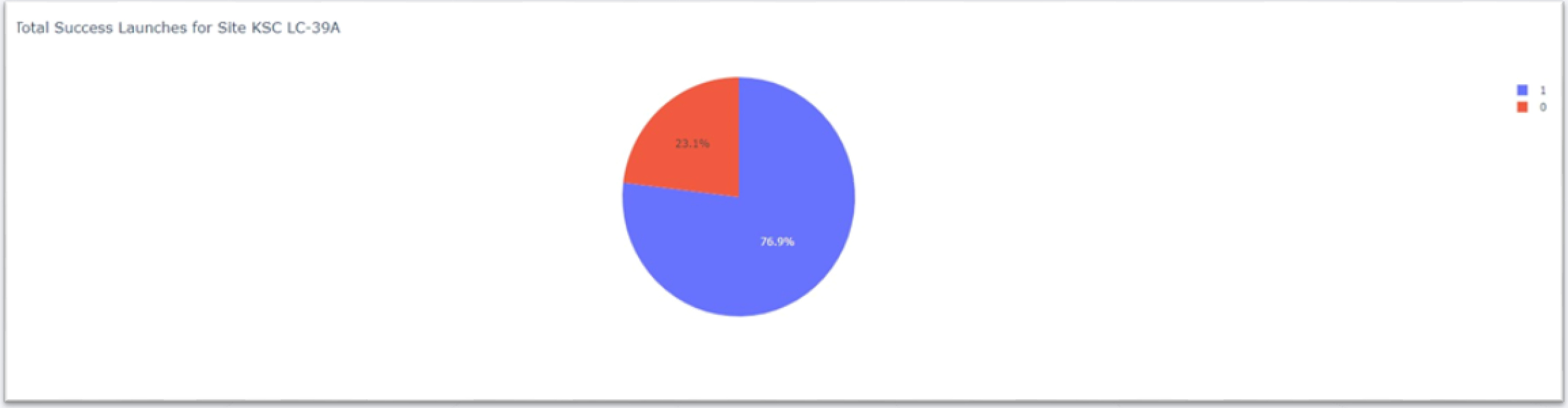
Build a Dashboard with Plotly Dash

Dashboard - Total success by Site



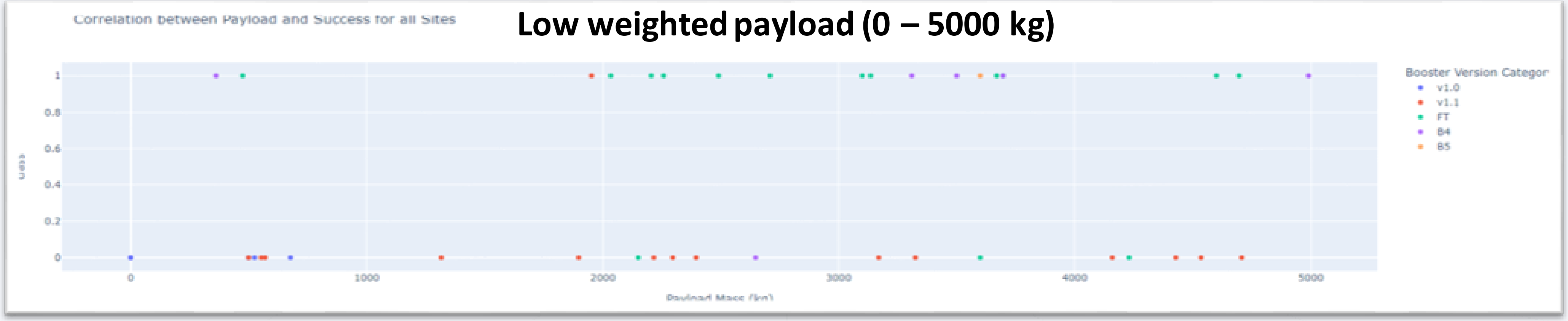
Based on the data, KSC LC-39A has the highest success rate of launches among the available sites.

Dashboard - Total success launches for Site KSC LC-39A

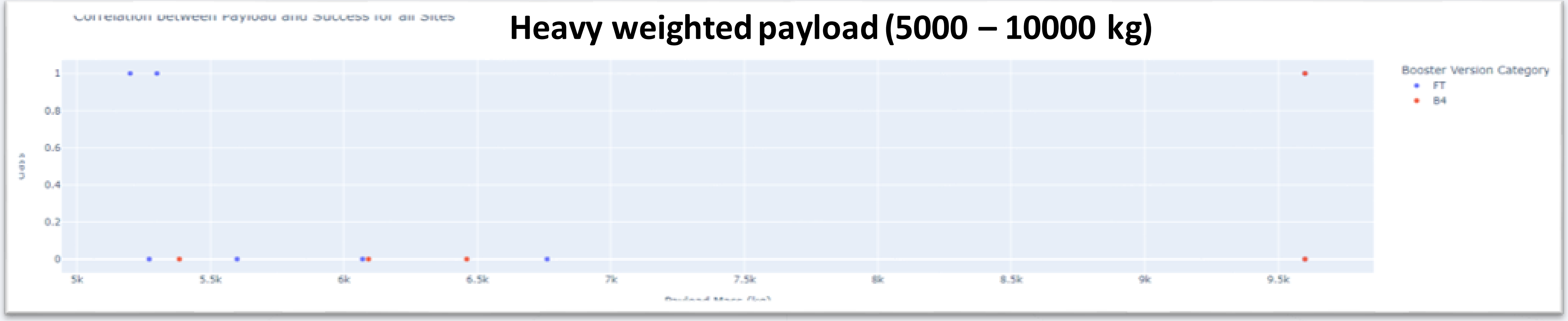


"KSC LC-39A has achieved a success rate of 76.9% and a failure rate of 23.1% based on the available data."

Dashboard - Payload mass vs Outcome for all sites with different payload mass selected



Low weighted payload (0 – 5000 kg)



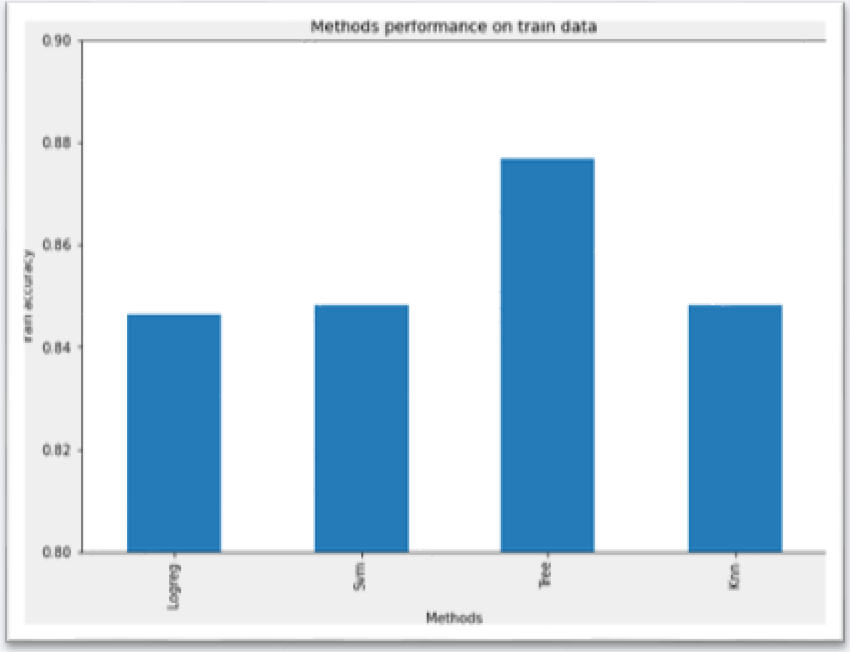
Heavy weighted payload (5000 – 10000 kg)

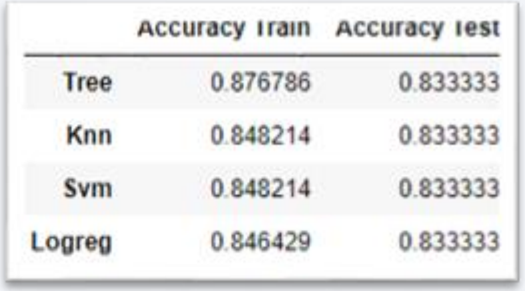
"The success rate for launches with low-weighted payloads is higher than the success rate for launches with heavy-weighted payloads."

Section 6

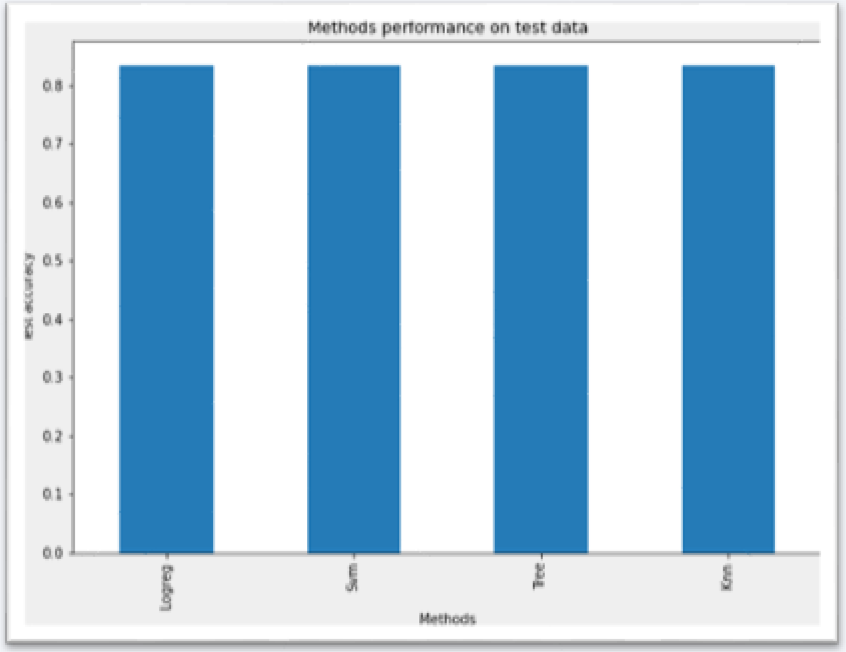
Predictive Analysis (Classification)

Classification Accuracy



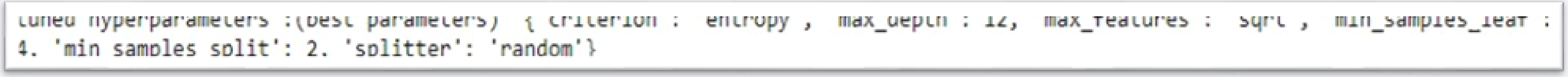


	Accuracy Train	Accuracy Test
Tree	0.876786	0.833333
Knn	0.848214	0.833333
Svm	0.848214	0.833333
Logreg	0.846429	0.833333



"According to the accuracy test results, all methods performed similarly. It would be beneficial to gather more test data in order to make a more informed decision, but if a choice must be made immediately, the decision tree method would be the most advisable option."

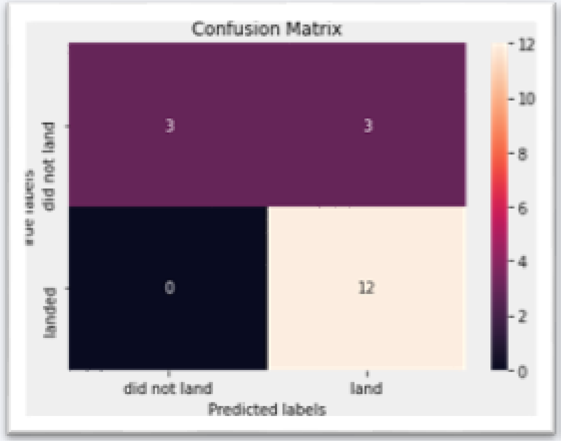
Decision tree optimal parameters



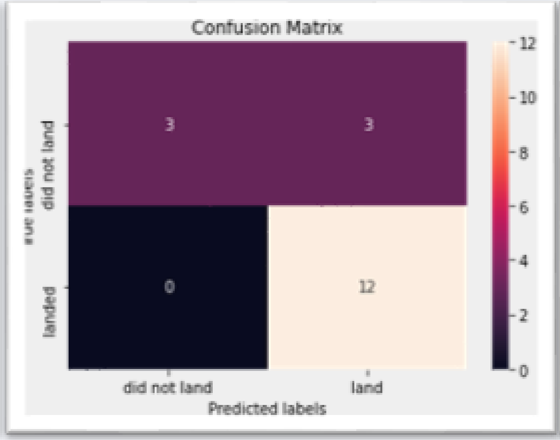
```
tuned hyperparameters :(best parameters)  { criterion :  entropy ,  max_depth : 12,  max_features :  sqrt ,  min_samples_leaf :
4. 'min samples split': 2. 'splitter': 'random'}
```

Confusion Matrix

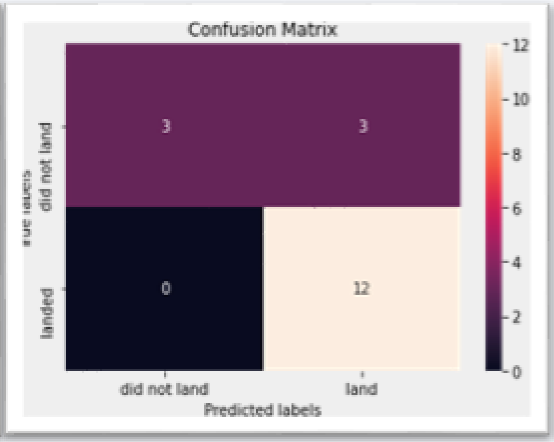
Logistic regression



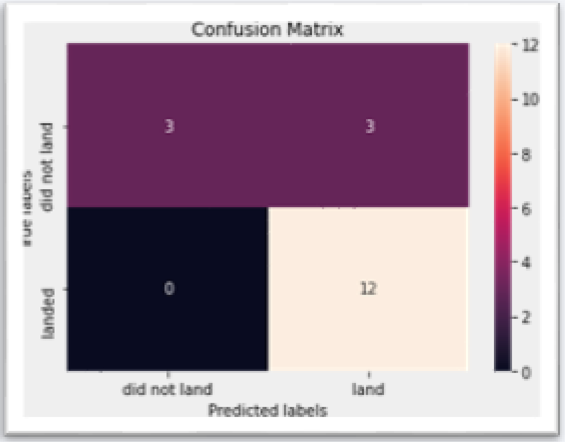
kNN



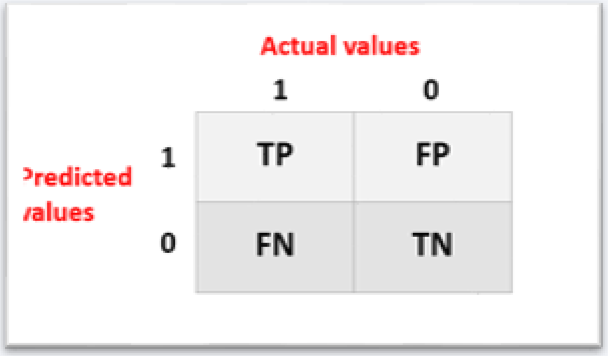
SVM



Decision Tree



"Since the test accuracies for all methods are equal, the confusion matrices are also identical. The main issue with these models is a high number of false positives."



		Actual values	
		1	0
Predicted values	1	TP	FP
	0	FN	TN

Conclusions

There are several factors that can contribute to the success of a mission, including the launch site, orbit, and number of previous launches. Gains in knowledge and experience gained from previous launches may help increase the chances of success. The orbits with the highest success rates are GEO, HEO, SSO, and ES-L1. The payload mass may also be a factor in the success of a mission, depending on the orbit, but generally, low-weighted payloads have a higher success rate than heavy-weighted payloads. Based on the current data, it is not possible to determine why some launch sites have higher success rates than others (such as KSC LC-39A). To answer this question, it would be necessary to gather additional data, such as atmospheric conditions or other relevant information. In this dataset, the Decision Tree Algorithm was chosen as the best model because it had the highest train accuracy, even though the test accuracies for all the models were identical.

Thank you!