Winning Space Race with Data Science

Bin Cui

13. April 2022

Outline

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

Executive Summary

- Summary of methodologies
  - ❑ Data Collection through API and Web Scraping
  - ❑ Data Wrangling
  - ❑ Exploratory Data Analysis with SQL and Data Visualization (Seaborn)
  - ❑ Interactive Visual Analytics with Folium/Plotly
  - ❑ Machine Learning Prediction (Classification)
- Summary of all results
  - ❑ Exploratory Data Analysis results
  - ❑ Interactive Analytics
  - ❑ Predictive Analytics

Introduction

Project background and context

Space X advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because Space X can reuse the first stage. Therefore, if we can determine if the first stage will land, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against space X for a rocket launch. This goal of the project is to create a machine learning pipeline to predict if the first stage will land successfully.

- What factors determine if the rocket will land successfully?
- How does various features interact that determine the success rate of a successful landing?
- What operating conditions need to be ensured for a successful landing?

Section 1

Methodology

Methodology

Executive Summary

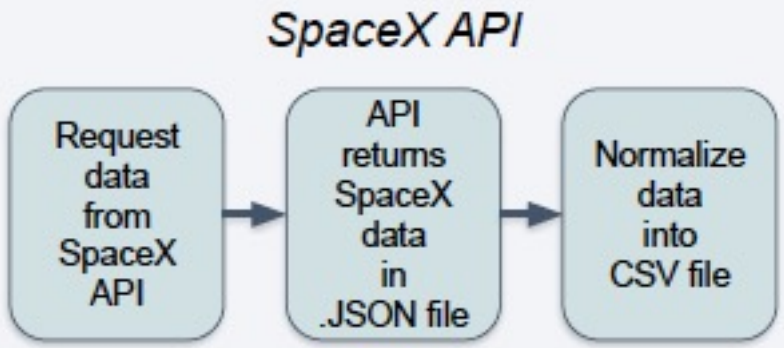
- Data collection methodology:
  - SpaceX API & Web Scraping Falcon 9 and Falcon Heavy Launches Records from Wikipedia
- Perform data wrangling
  - One-hot Encoding to categorical features
  - Missing values were taken care of by proper replacements
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Finding best Hyperparameters for SVM, KNN, Decision Trees and Logistic Regression

Data Collection

- Data was collected using the get request to the SpaceX API.
- Decoding the responste content as a Json using .json() function call and convert it into a pandas dataframe using .json_normalize()
- Cleaning the data, checking for missing values and fill in missing values where needed.
- In addition, we performed web scraping from Wikipedia for Falcon 9 launch records with BeautifulSoup.
- Objective was to extract the launch records as HTML table, parse table and convert it into pandas dataframe for further analysis.

Data Collection – SpaceX API

GitHub URL



We used the get request to the SpaceX API to collect data, clean the requested data and did some basic data wrangling and formatting.

Task 1: Request and parse the SpaceX launch data using the GET request

To make the requested JSON results more consistent, we will use the following static response object for this project:

In [44]:
```
static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-Sk
```

We should see that the request was successfull with the 200 status response code

In [45]:
```
response.status_code
```

Out[45]:
```
200
```

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

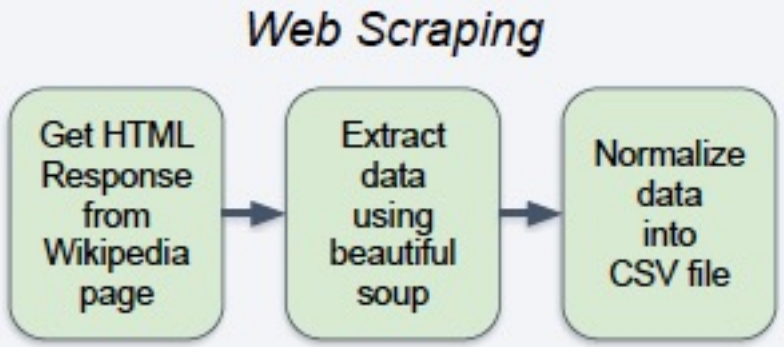
In [46]:
```
# Use json_normalize meethod to convert the json result into a dataframe

#response=requests.get(static_json_url)
data=pd.json_normalize(response.json())
```

Using the dataframe `data` print the first 5 rows

In [47]:
```
# Get the head of the dataframe
data.head()
```

Data Collection - Scraping

GitHub URL



We applied web scrapping to Falcon 9 launch records with BeautifulSoup

We parsed the table and converted it into a pandas dataframe.

TASK 1: Request the Falcon9 Launch Wiki page from its URL

First, let's perform an HTTP GET method to request the Falcon9 Launch HTML page, as an HTTP response.

n [17]:
```
# use requests.get() method with the provided static_url
# assign the response to a object
response=requests.get(static_url).text
#response
```

Create a `BeautifulSoup` object from the HTML `response`

n [18]:
```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(response, 'html5lib')
```

Print the page title to verify if the `BeautifulSoup` object was created properly

n [19]:
```
# Use soup.title attribute
soup.title
```

ut[19]:
```
<title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>
```

TASK 2: Extract all column/variable names from the HTML table header

Next, we want to collect all relevant column names from the HTML table header

Let's try to find all tables on the wiki page first. If you need to refresh your memory about `BeautifulSoup`, please check the external reference link towards the end of this lab

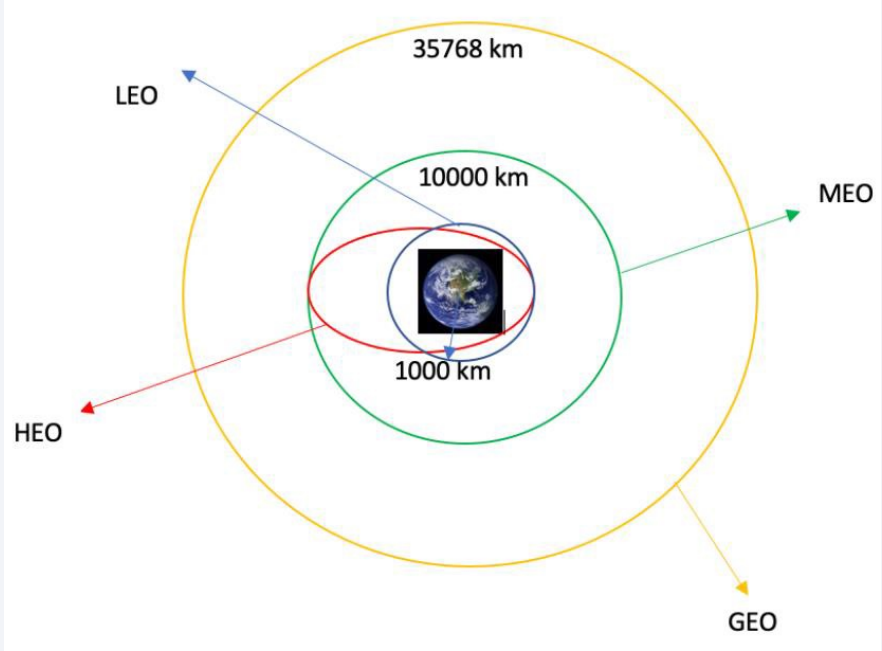
n [27]:
```
# Use the find_all function in the BeautifulSoup object, with element type `table`
# Assign the result to a list called `html_tables`

#for row in soup.find("tbody").find_all('tr'):
html_tables=soup.find_all('tr')
```

Starting from the third table is our target table contains the actual launch records.

Data Wrangling

GitHub URL

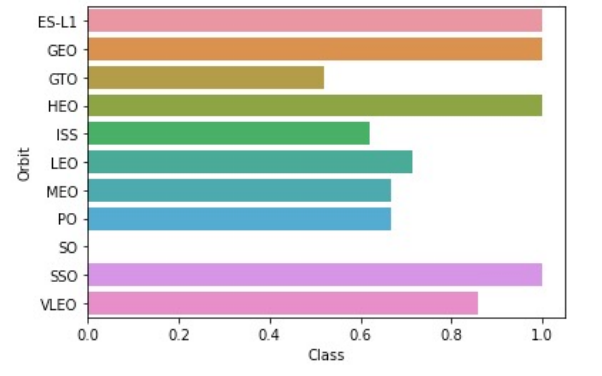
- We performed exploratory data analysis (EDA) to find some patterns in the data and determined the training labels for supervised models.
- We calculated the number of launches at each site, and the number and occurrence of each orbits
- We also calculated the mission outcome for each orbit type
- Finally, we created landing outcome label from outcome column and exported the results to csv.

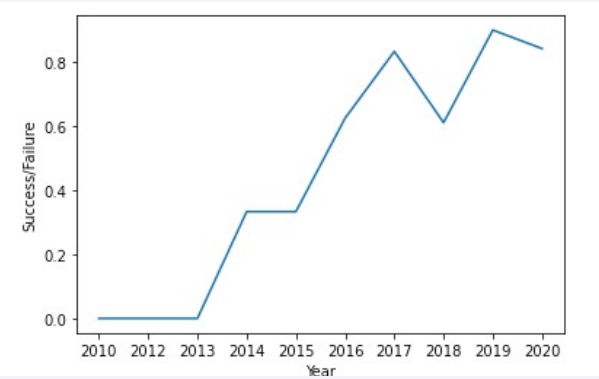


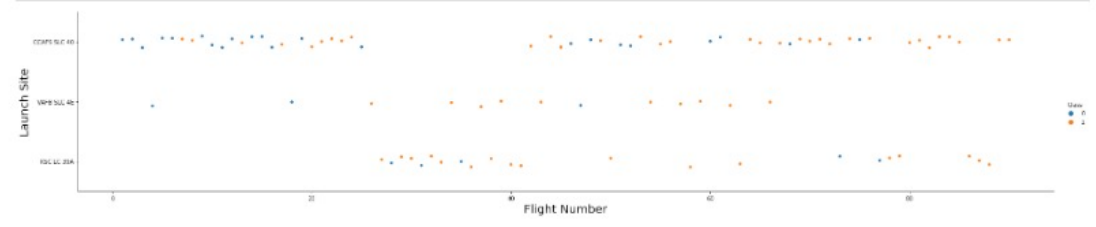
EDA with Data Visualization

GitHub URL

We explored the data by visualizing the relationship between flight number and launch Site, payload and launch site, success rate of each orbit type, flight number and orbit type, the launch success yearly trend.







EDA with SQL

GitHub URL

- We loaded the SpaceX dataset into a PostgreSQL database without leaving the Jupyternotebook.

- We applied EDA with SQL to get insight from the data. We wrote queries to find out for instance:

  - The names of unique launch sites in the space mission.
  - The total payload mass carried by boosters launched by NASA (CRS)
  - The average payload mass carried by booster version F9 v1.1
  - The total number of successful and failure mission outcomes
  - The failed landing outcomes in drone ship, their booster version and launch site names.

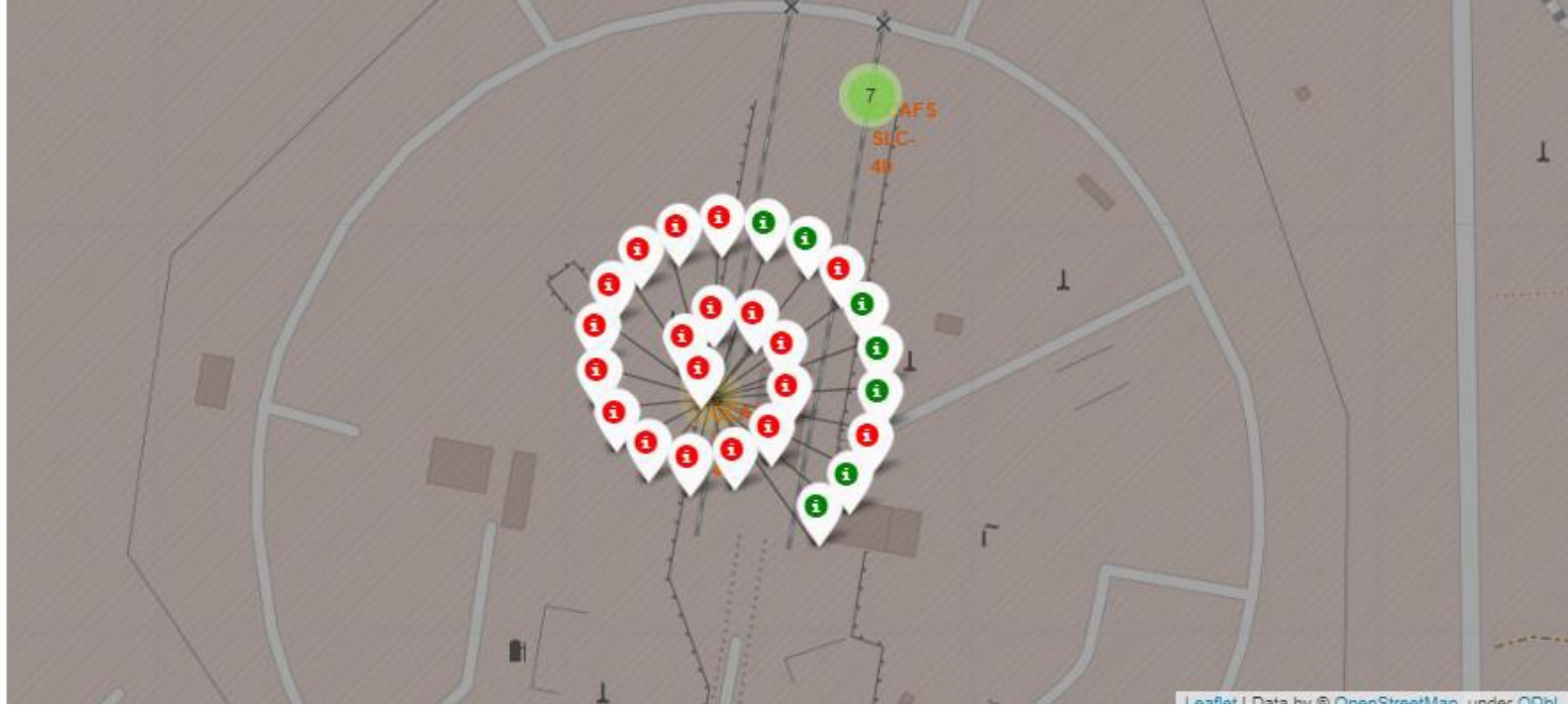
Build an Interactive Map with Folium

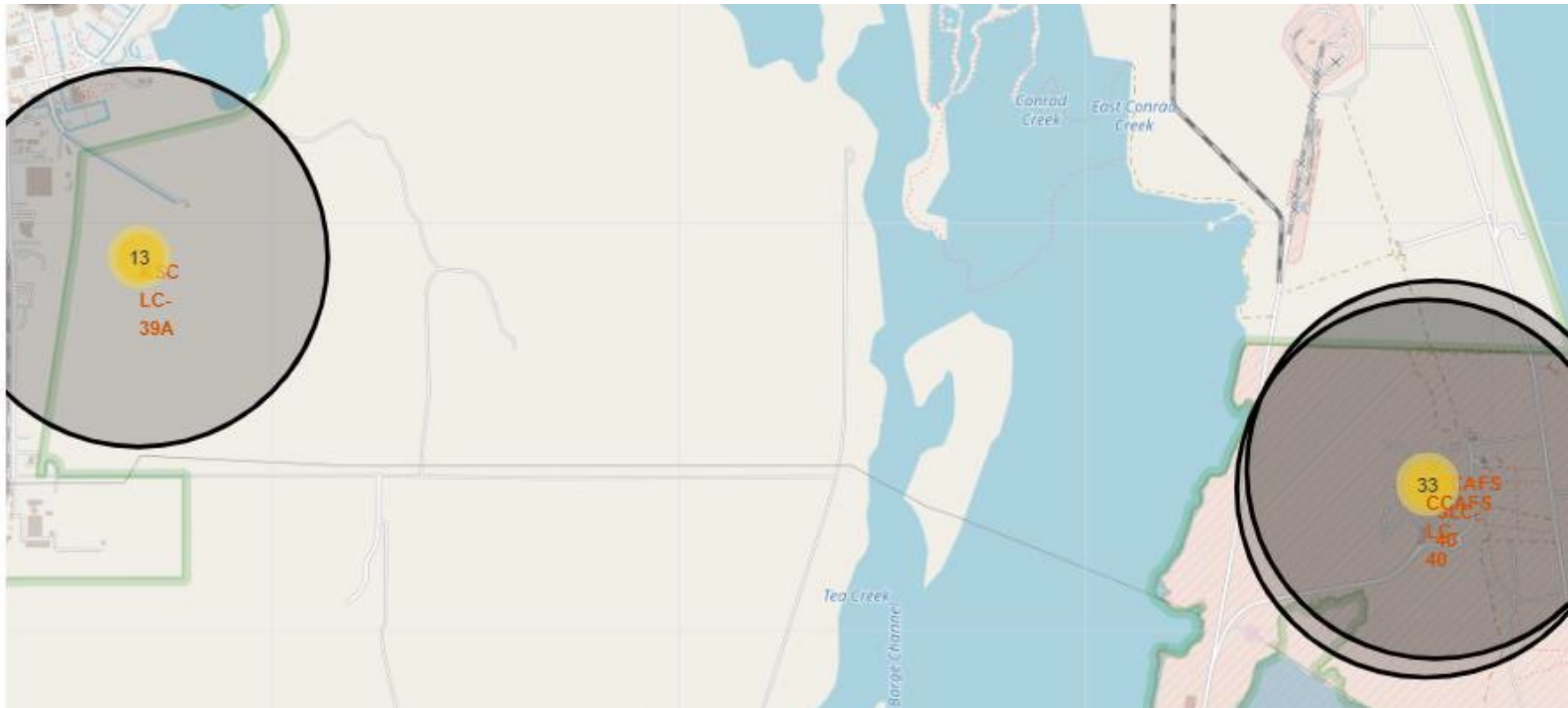
GitHub URL

- Objects created and added to a folium map:
  - Markers that show all launch sites on a map
  - Markers that show the success/failed launches for each site on the map
  - Lines that show the distances between a 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

Folium


7
AFS
SLC-
40


13
LC-
39A
Conrad
Creek
East Conrad
Creek
33
CCAFS
SLC-
40
Tee Creek

Build a Dashboard with Plotly Dash

GitHub URL

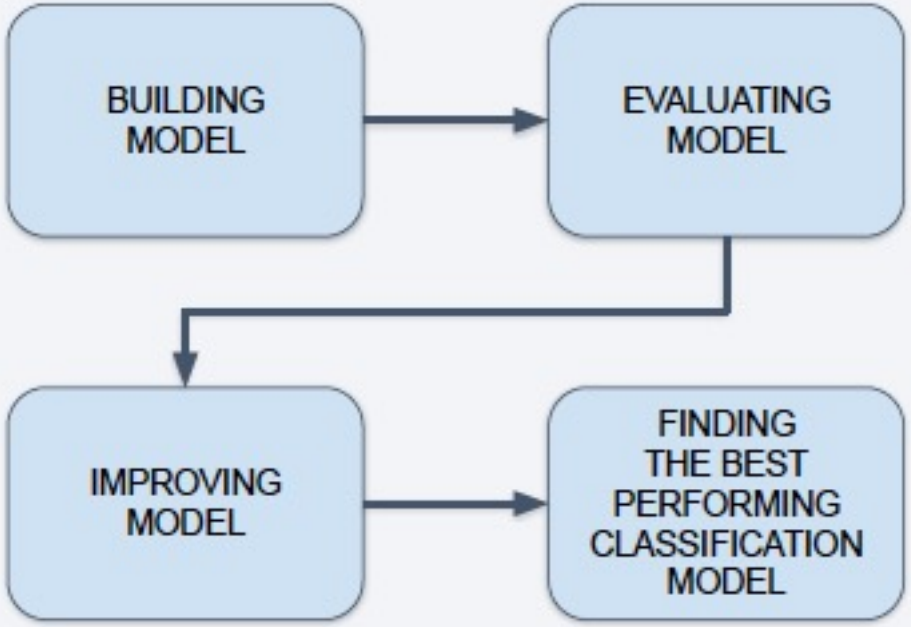
The dashboard application contains a pie chart and a scatter point chart:

- 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.
- Scatter chart
  - For showing the relationship between Outcomes and Payload mass (Kg) by different boosters
  - Has 2 inputs: All sites/individual site & Payload mass on a slider between 0 and 10000 kg
  - This chart helps determine how success depends on the launch point, payload mass, and booster version categories.

Predictive Analysis (Classification)

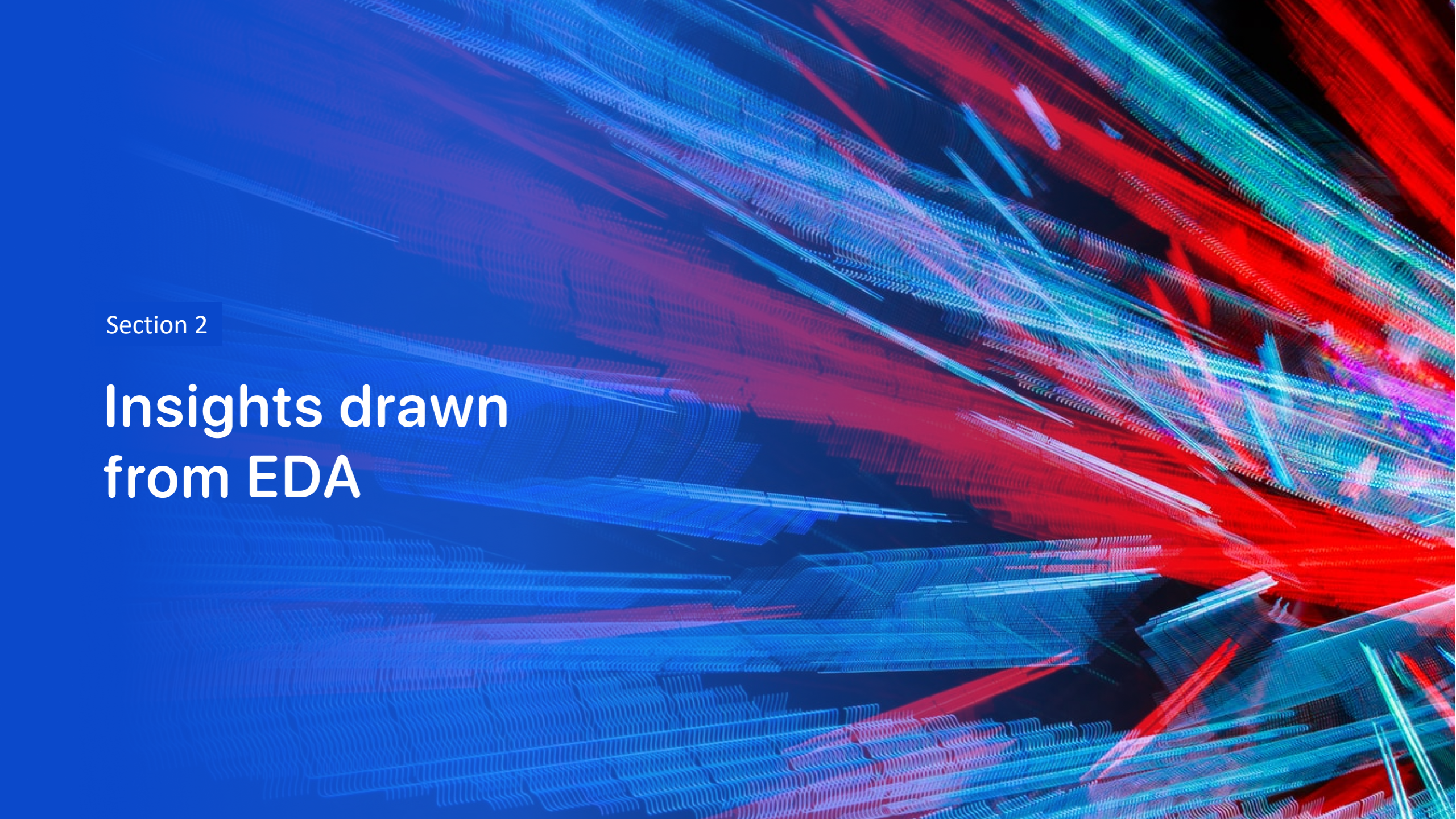
GitHub URL

- Perform exploratory Data Analysis and determine Training Labels
  - Create a column for the class
  - Standardize the data
  - Split into training data and test data
- Find best Hyperparameter for SVM, Classification Trees and Logistic Regression
  - Find the method performs best using test data



Results

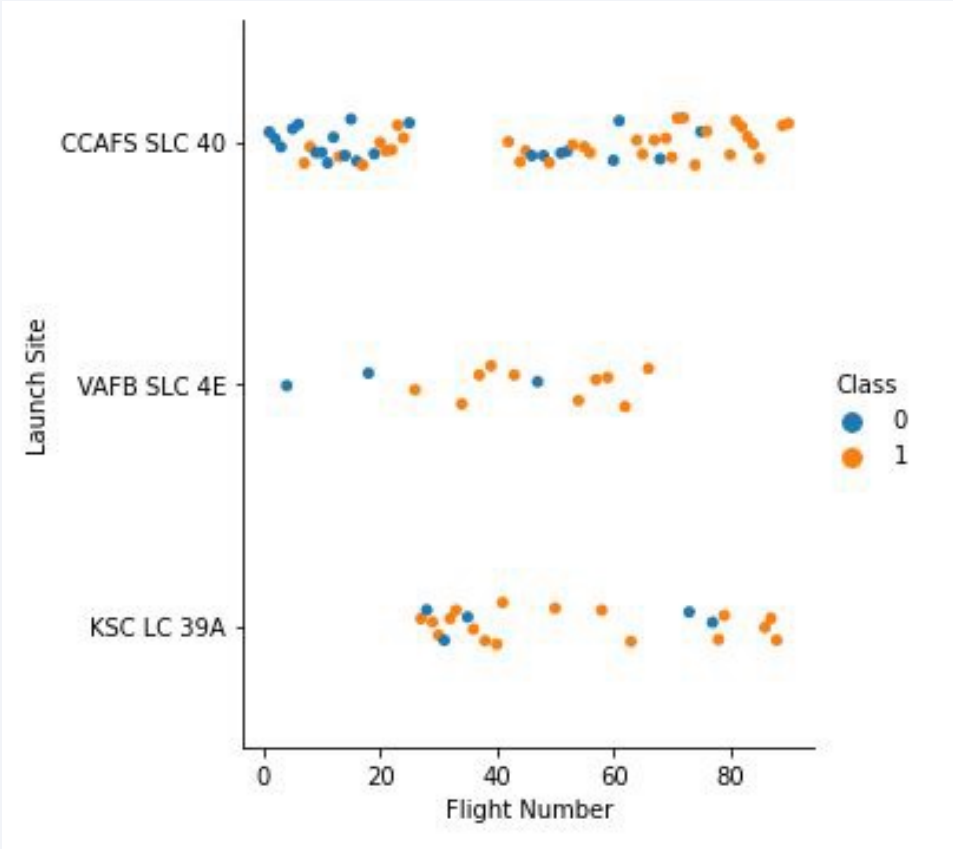
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results


Section 2
Insights drawn from EDA

Flight Number vs. Launch Site

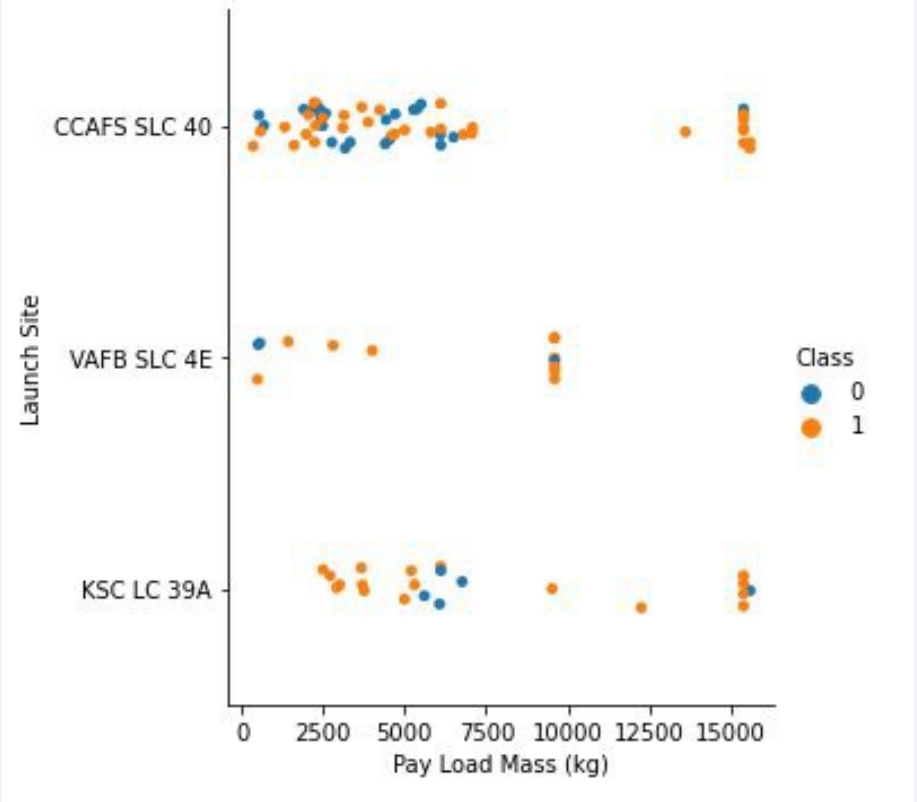
This figure shows that the success rate increased as the number of flights increased.

As the success rate has increased considerably since the 20th flight, this point seems to be a big breakthrough.



Payload vs. Launch Site

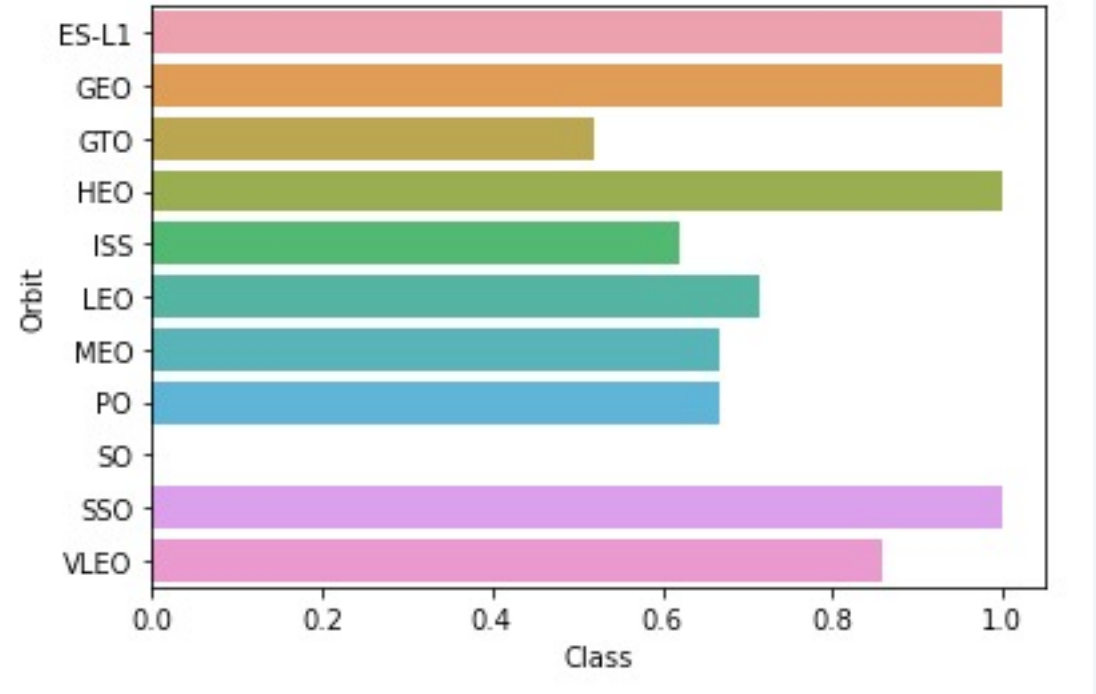
At first glance, the larger pay load mass, the higher the rocket's success rate, but it seems difficult to make decisions based on this figure because no clear pattern can be found between successful launch and Pay Load Mass.



Success Rate vs. Orbit Type

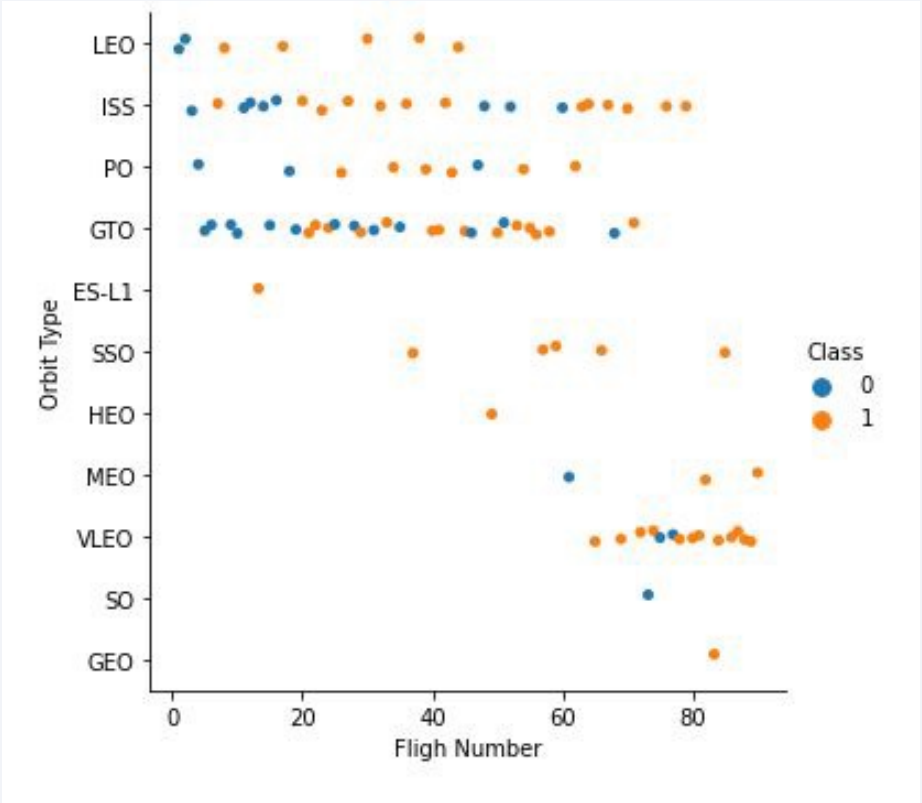
Orbit types SSO, HEO, GEO, and ES-L1 have the highest success rates (100%).

On the 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

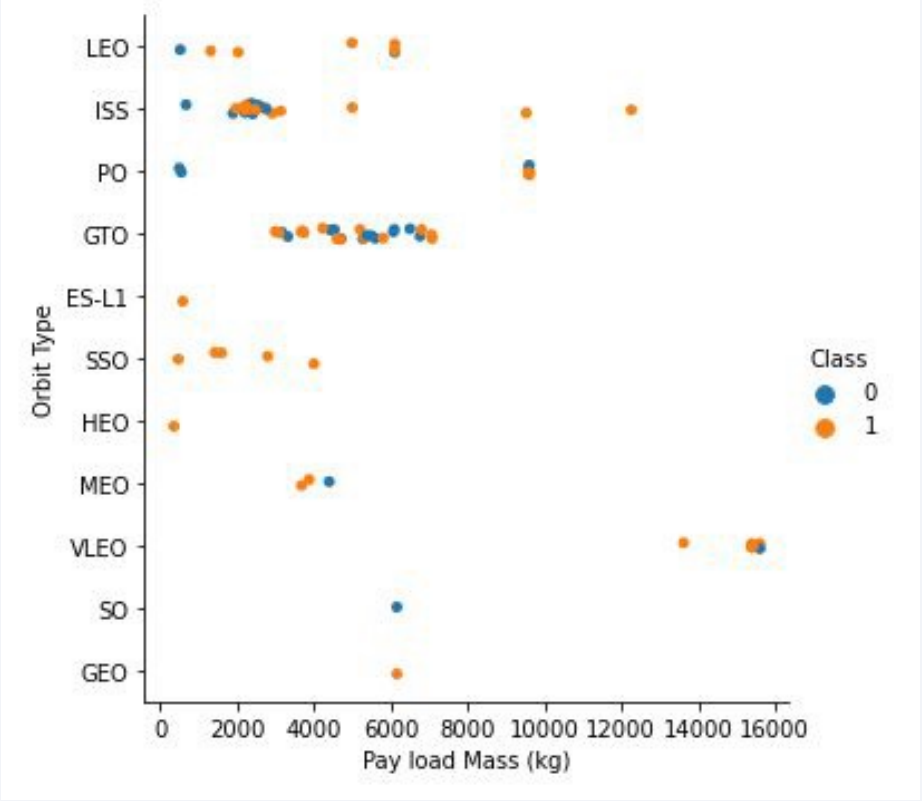
- In most cases, the launch outcome seems to be correlated with the flight number.
- On the other hand, in GTO orbit, there seems to be no relationship between flight numbers and success rate.
- SpaceX starts with LEO with a moderate success rate, and it seems that VLEO, which has a high success rate,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



Payload vs. Orbit Type

With heavy payloads the successful landing or positive landing rate are more for LEO and ISS.

However, in the case of GTO, it is hard to distinguish between the positive landing rate and the negative landing because they are all gathered together.

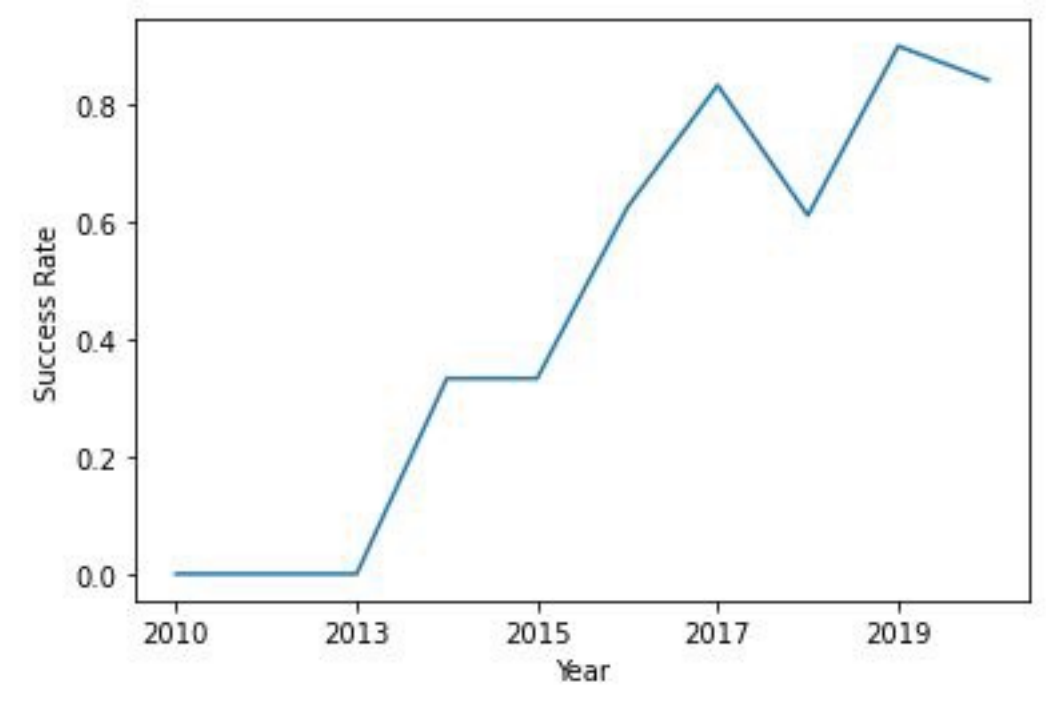


Launch Success Yearly Trend

Since 2013, the success rate has continued to increase until 2017.

The rate decreased slightly in 2018.

Recently, it has shown a success rate of about 80%.



All Launch Site Names

Task 1

Display the names of the unique launch sites in the space mission

```
In [8]: %sql SELECT DISTINCT LAUNCH_SITE FROM SPACEXDATASET
```

```
 * ibm_db_sa://hfd24292:***@3883e7e4-18f5-4afe-be8c-fa31c41761d2.bs2io90l
08kqblod8lcg.databases.appdomain.cloud:31498/bludb
Done.
```

Out[8]:

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

With SELECT DISTINCT, only the unique launch sites will be presented

Launch Site Names Begin with 'CCA'

Task 2

Display 5 records where launch sites begin with the string 'CCA'

In [9]:
```
%%sql SELECT LAUNCH_SITE
FROM SPACEXDATASET
WHERE LAUNCH_SITE LIKE 'CCA%'
LIMIT 5
```

```
 * ibm_db_sa://hfd24292:***@3883e7e4-18f5-4afe-be8c-fa31c41761d2.bs2io901
08kqblod8lcg.databases.appdomain.cloud:31498/bludb
Done.
```

Out[9]:

launch_site
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40

To display only 5 records where launch sites begin with ‘CCA’, the LIMIT and LIKE Operation is used

Total Payload Mass

Task 3

Display the total payload mass carried by boosters launched by NASA (CRS)

In [12]:
```
%%sql SELECT SUM(PAYLOAD_MASS__KG_) AS TOTALPAYLOAD
FROM SPACEXDATASET
WHERE CUSTOMER = 'NASA (CRS)'
```

```
 * ibm_db_sa://hfd24292:***@3883e7e4-18f5-4afe-be8c-fa31c41761d2.bs2io901
08kqblod8lcg.databases.appdomain.cloud:31498/bludb
Done.
```

Out[12]:

totalpayload
45596

The total payload mass can be determined with the SELECT SUM-operation

Average Payload Mass by F9 v1.1

Task 4

Display average payload mass carried by booster version F9 v1.1

In [16]:
```
%%sql SELECT AVG(PAYLOAD_MASS__KG_) AS AVGPAYLOAD
FROM SPACEXDATASET
WHERE BOOSTER_VERSION LIKE 'F9 v1.1'
```

```
 * ibm_db_sa://hfd24292:***@3883e7e4-18f5-4afe-be8c-fa31c41761d2.bs2io90l
08kqb1od8lcg.databases.appdomain.cloud:31498/bludb
Done.
```

Out[16]:

avgpayload
2928

Average payload mass can be calculated with SELECT AVG()

First Successful Ground Landing Date

Task 5

List the date when the first successful landing outcome in ground pad was acheived.

Hint:Use min function

In [18]:
```
%%sql SELECT min(DATE)
FROM SPACEXDATASET
WHERE LANDING__OUTCOME ='Success (ground pad)'
```

```
 * ibm_db_sa://hfd24292:***@3883e7e4-18f5-4afe-be8c-fa31c41761d2.bs2io901
08kqblod8lcg.databases.appdomain.cloud:31498/bludb
Done.
```

Out[18]:

1
2015-12-22

First successful landing date can be determined with SELECT min()

Successful Drone Ship Landing with Payload between 4000 and 6000

Task 6

List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

In [25]:
```
%%sql SELECT BOOSTER_VERSION
FROM SPACEXDATASET
WHERE LANDING__OUTCOME ='Success (drone ship)' AND PAYLOAD_MASS__KG_ BETWE
EN 4000 AND 6000
```

```
 * ibm_db_sa://hfd24292:***@3883e7e4-18f5-4afe-be8c-fa31c41761d2.bs2io901
08kqblod8lcg.databases.appdomain.cloud:31498/bludb
Done.
```

Out[25]:

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

To determined the Payload between 4000 and 6000 the WHERE AND BETWEEN functions are used

Total Number of Successful and Failure Mission Outcomes

Task 7

List the total number of successful and failure mission outcomes

In [37]:
```
%%sql
Select MISSION_OUTCOME, count(MISSION_OUTCOME) as count from SPACEXDATASET
GROUP BY MISSION_OUTCOME
```

```
 * ibm_db_sa://hfd24292:***@3883e7e4-18f5-4afe-be8c-fa31c41761d2.bs2io901
08kqblod8lcg.databases.appdomain.cloud:31498/bludb
Done.
```

Out[37]:

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

We counted the mission outcome and used a group by on Mission_Outcome to find number of success or failure.

Boosters Carried Maximum Payload

Task 8

List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

In [39]:
```
%%sql
select BOOSTER_VERSION FROM SPACEXDATASET where payload_mass__kg_ = (selec
t max(payload_mass__kg_) from SPACEXDATASET) group by BOOSTER_VERSION
```

```
 * ibm_db_sa://hfd24292:***@3883e7e4-18f5-4afe-be8c-fa31c41761d2.bs2io901
08kqblod8lcg.databases.appdomain.cloud:31498/bludb
Done.
```

Out[39]:

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

We determined the booster that have carried the maximum payload using a subquery in the WHERE clause and the MAX() function.

2015 Launch Records

Task 9

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

In [41]:
```
%%sql
select landing__outcome, booster_version, launch_site from SPACEXDATASET w
here (landing__outcome = 'Failure (drone ship)') and (year(date) = '2015')
```

```
 * ibm_db_sa://hfd24292:***@3883e7e4-18f5-4afe-be8c-fa31c41761d2.bs2io901
08kqb1od8lcg.databases.appdomain.cloud:31498/bludb
Done.
```

Out[41]:

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

To determine the landing_outcomes, again the WHERE & AND clause is used

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

Task 10

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

In [43]:
```
%%sql
select landing__outcome, count(landing__outcome) as Rank from SPACEXDATASE
T where date between '2010-06-04' and '2017-03-20' group by landing__outco
me order by count(landing__outcome) desc
```

```
 * ibm_db_sa://hfd24292:***@3883e7e4-18f5-4afe-be8c-fa31c41761d2.bs2io901
08kqblod8lcg.databases.appdomain.cloud:31498/bludb
Done.
```

Out[43]:

landing__outcome	RANK
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

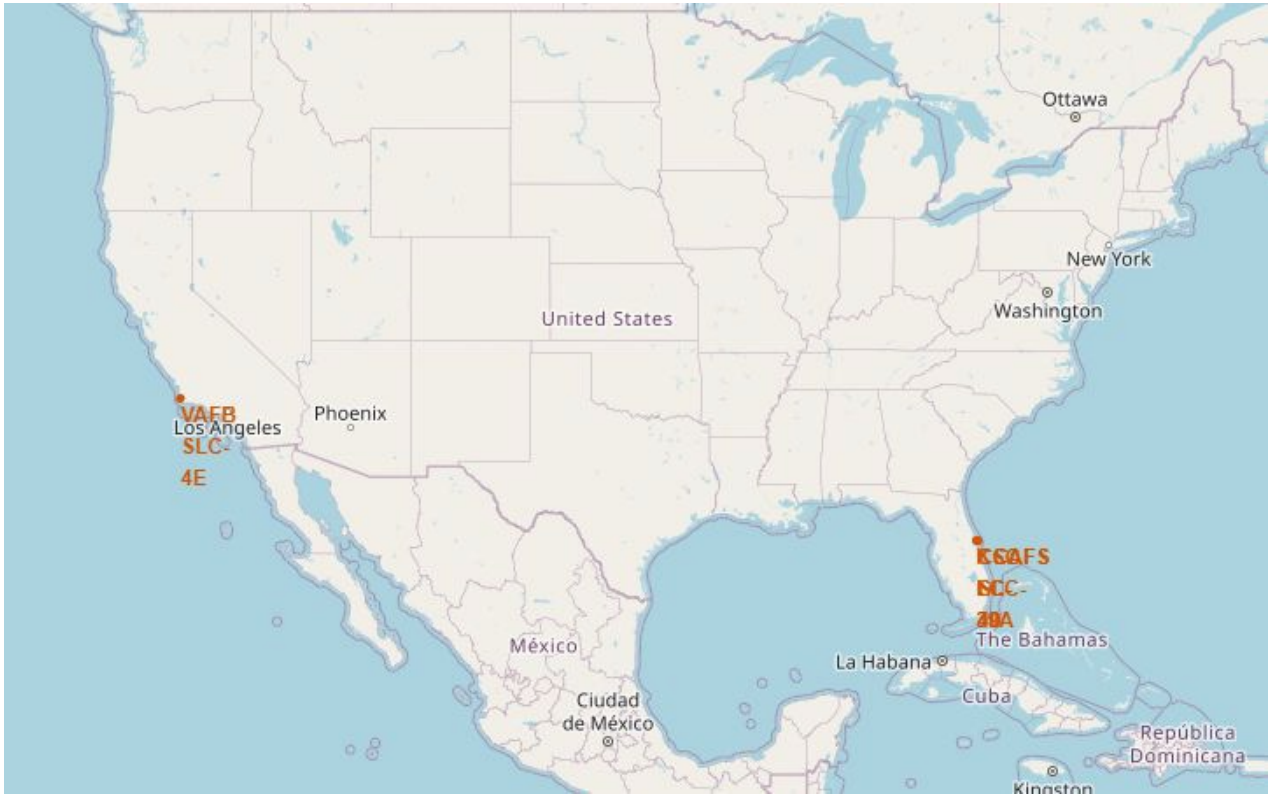
We selected Landing outcomes and the COUNT of landing outcomes from the data and used the WHERE clause to filter for landing outcomes BETWEEN2010-06-04 to 2010-03-20.

We applied the GROUP BY clause to group the landing outcomes and the ORDER BY clause to order the grouped landing outcome in descending order.

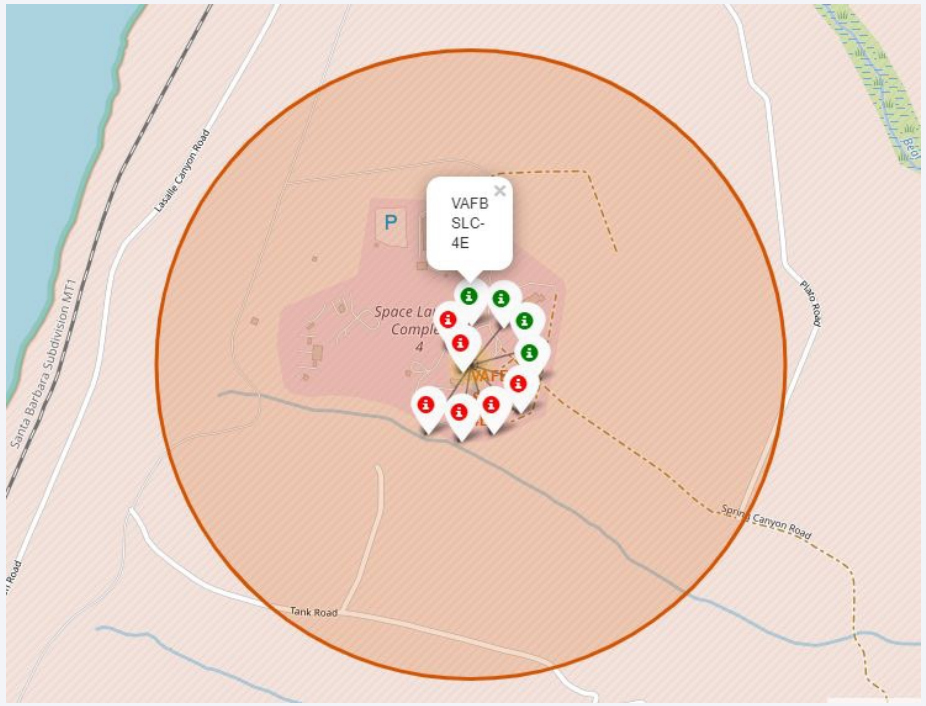

Section 3
Launch Sites
Proximities Analysis

All Launch Sites' Locations

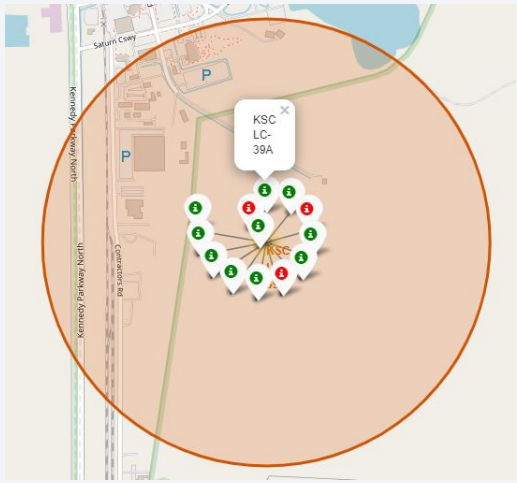
We can see that the SpaceX launch sites are in the coasts of United States of America, Florida and California.

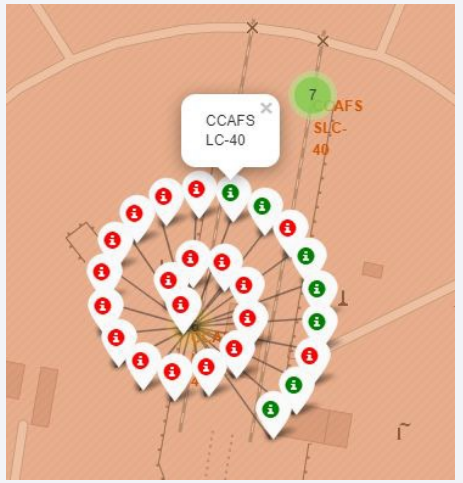


Markers showing launch sites with color labels



Launch site in California

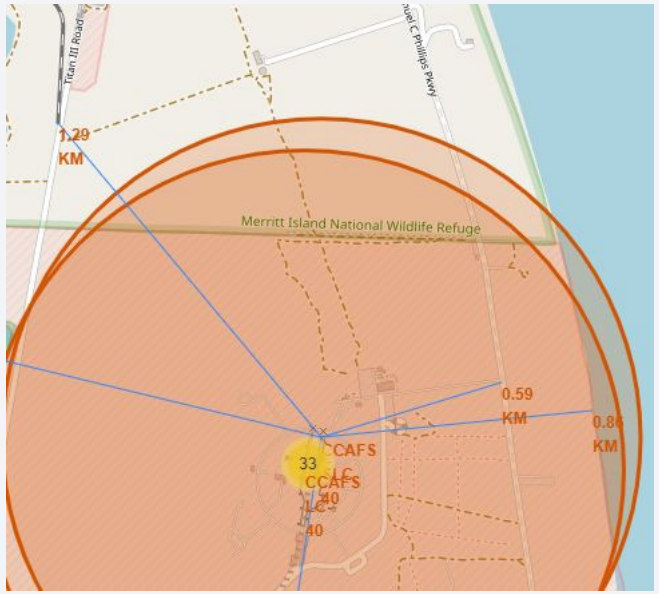




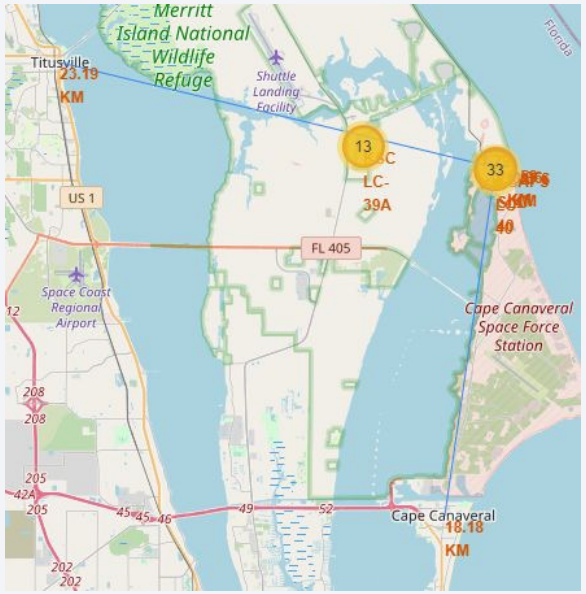
Launch sites in Florida

By clicking on the marker clusters, successful landing (green) or failed landing (red) are displayed

Proximities of Launch Sites

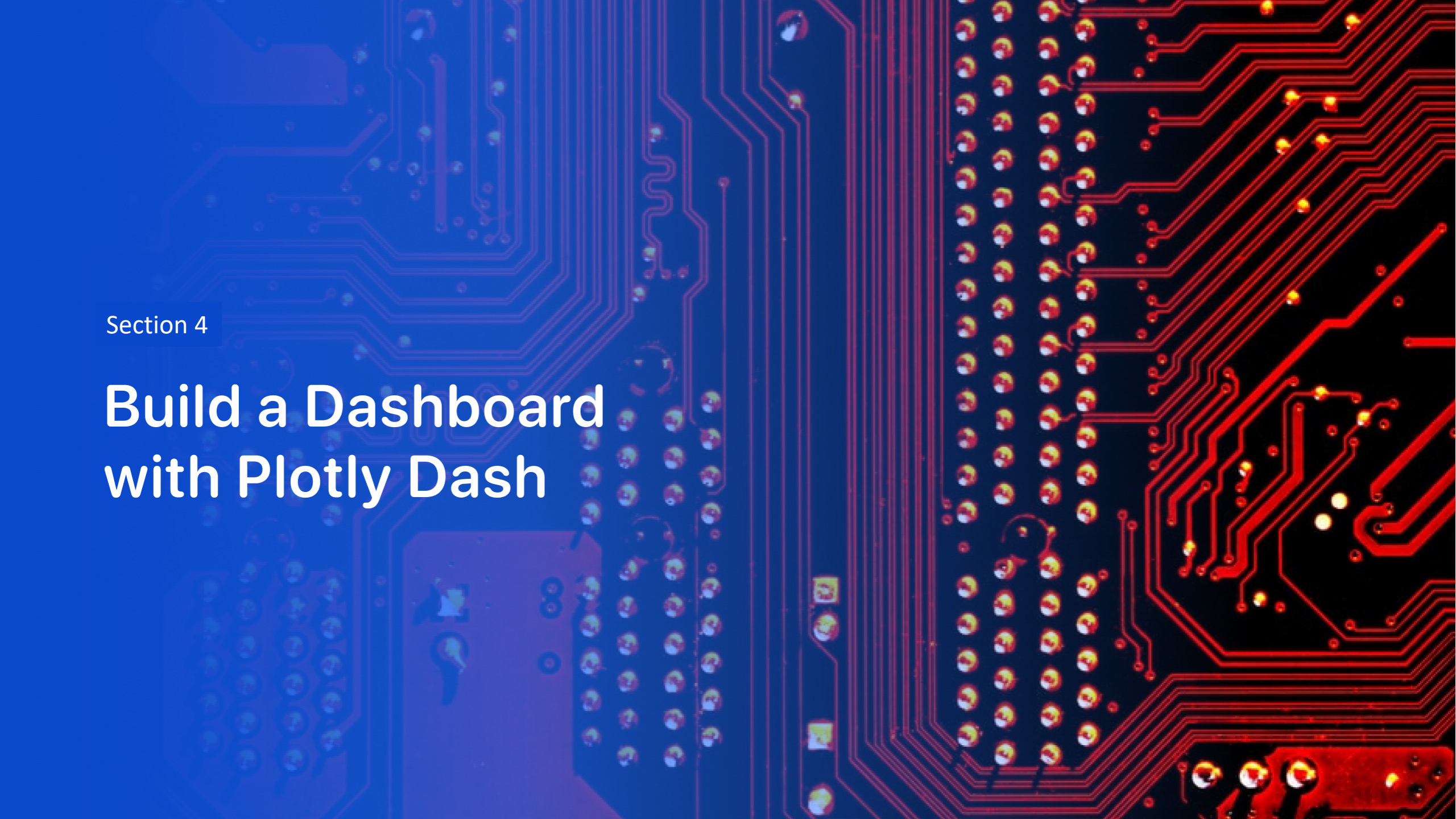


Launch sites close proximity to railways, highways and coastline



Launch sites keeping distance to other cities

It can be found 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.

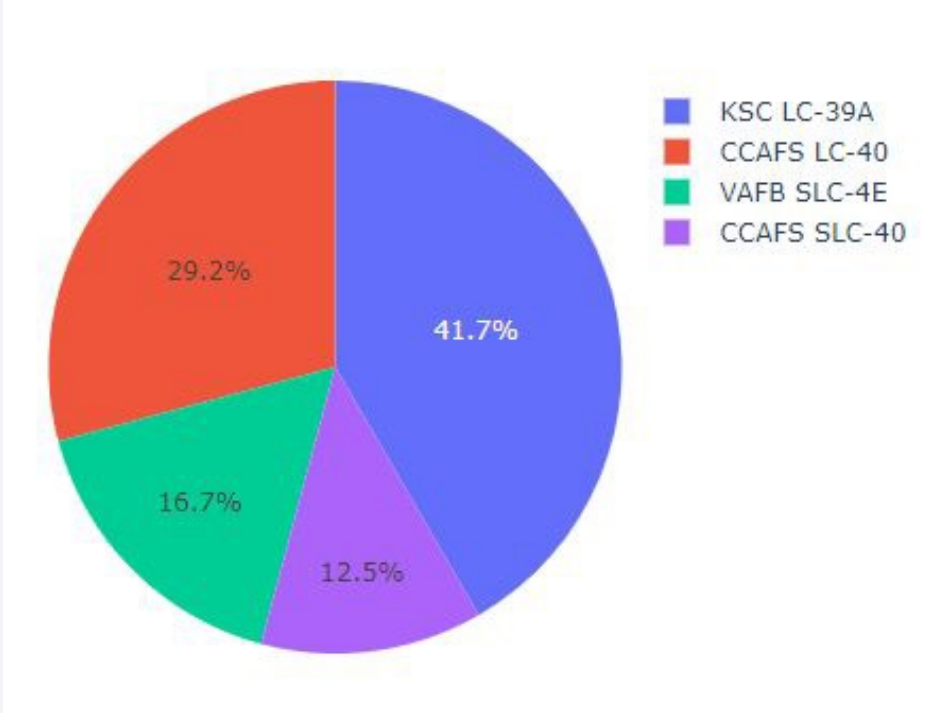

Section 4
Build a Dashboard with Plotly Dash

Total Success Launches By all sites

KSLC-39A records the most launch success among all sites.

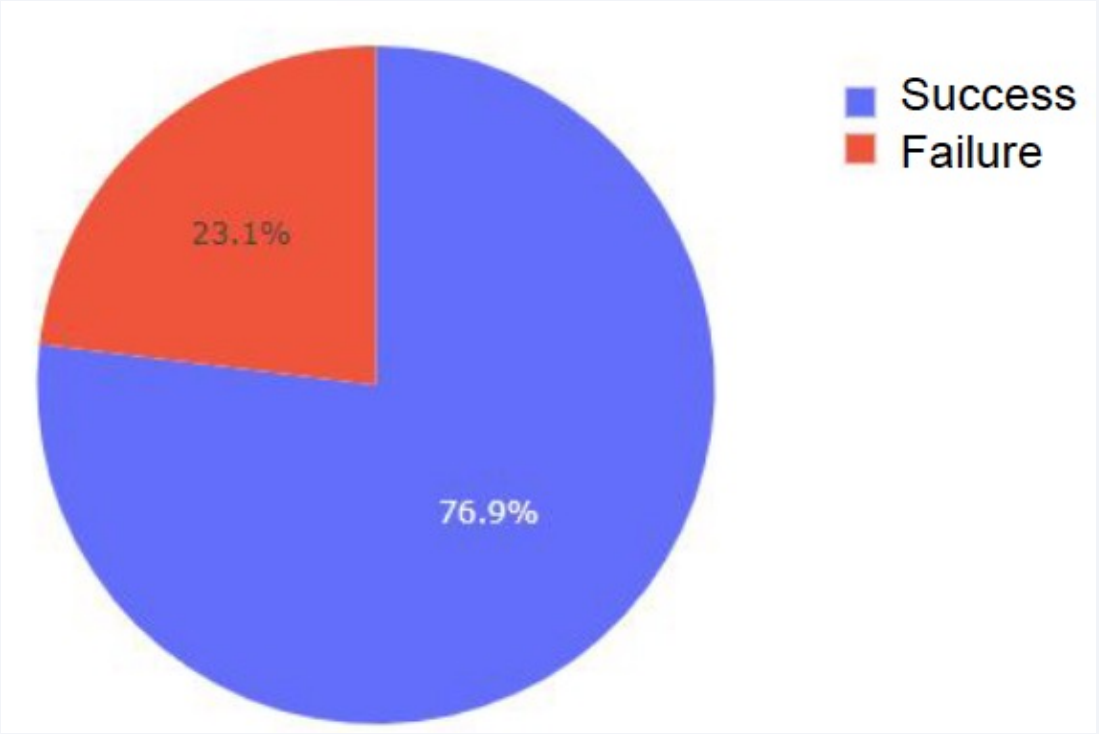
The VAFB SLC-4E has the fewest launch success, possibly because

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



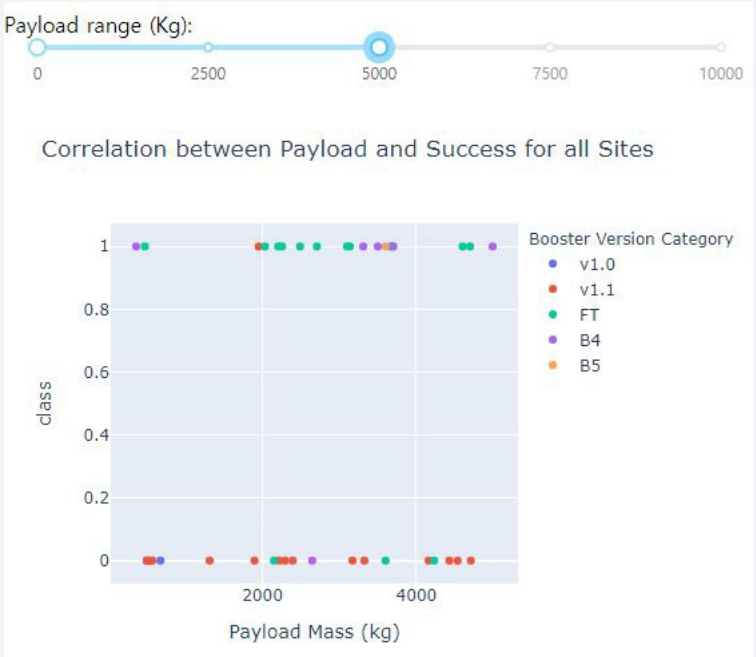
Launch Site with Highest Launch Success Ratio

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



Payload vs. Launch Outcome Scatter Plot for all sites

These figures show that the launch success rate (class 1) for low weighted payloads(0-5000 kg) is higher than that of heavy weighted payloads(5000-10000 kg).

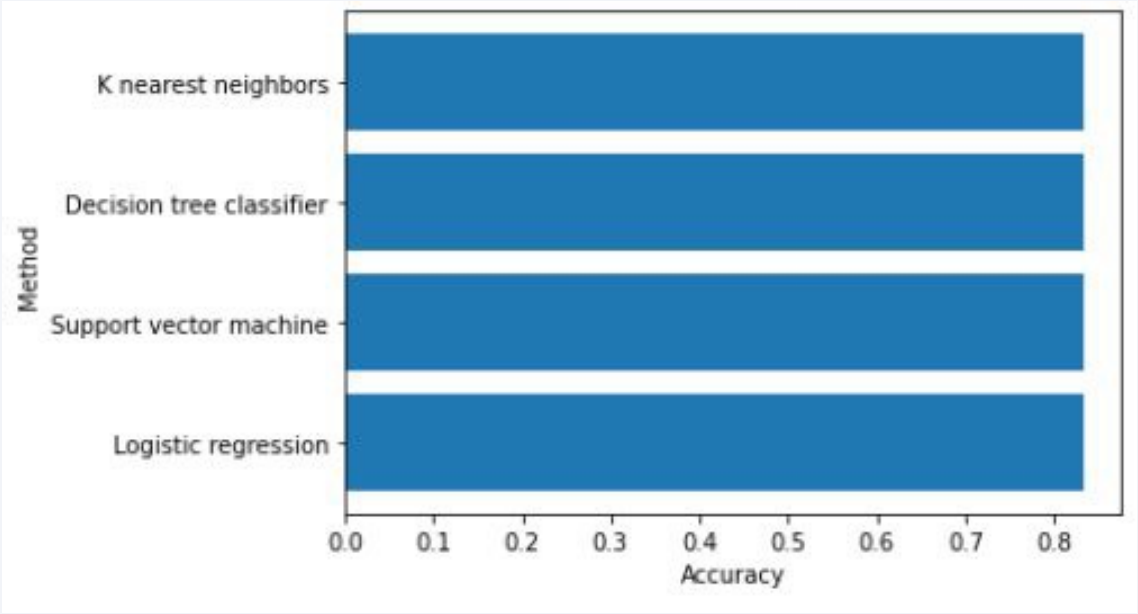


Section 5

Predictive Analysis (Classification)

Classification Accuracy

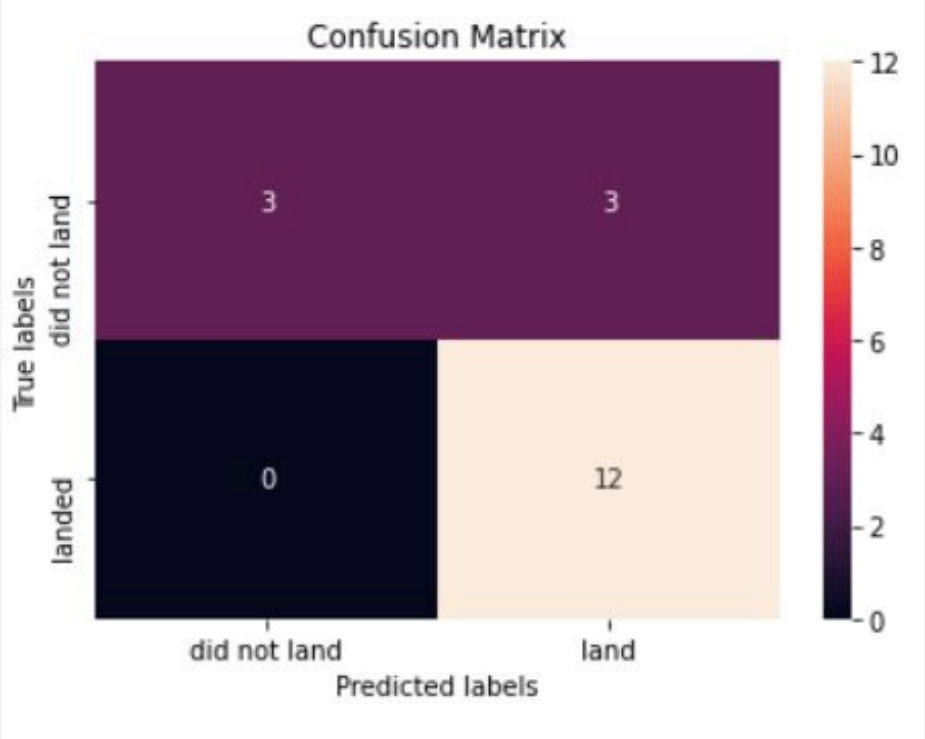
- In the test set, the accuracy of all models was virtually the same at 83.33%.
- It should be noted that the test size was small at 18.
- Therefore, more data is needed to determine the optimal model.



	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

- The confusion matrix is the 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 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

- By increasing the number of flights, the success rate is also increasing and has exceeded 80%.
- Orbital types SSO, HEO, GEO and ES-L1 have the highest success rate (100%).
  - KSLC-39A has the highest number of launch successes and success rate.
- Launch sites are mostly close to railways, highways and coastline but far from cities.
- The classification models showed an accuracy of 83,33%. However, more data is needed to determine the optimal model and parameters due to the small training and test data size.

Appendix

- GitHub Repository
- SpaceX Falcon 9 Wikipedia

Thank you!