IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

<Name>
<Date>

Outline

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

Executive Summary

- Summary of methodologies
  - Collect data from SpaceX API.
  - Collect data with web scraping techniques.
  - Data Wrangling
  - Exploratory Data Analysis using SQL
  - EDA with matplotlib
  - Interactive dashboard
  - Use of machine learning for prediction.
- Summary of all results
  - Visual analytics
  - Predictive analysis

Introduction

- Project background and context
- SpaceX has been able to reduce launch costs for the Falcon 9 primarily due to its ability to reuse the first and second stages of the rocket.
- In terms of cost per launch, the Falcon 9 is significantly cheaper than the United Launch Alliance's Atlas V and the same company's Delta IV Heavy. In addition, the Falcon 9 is cheaper than the European Space Agency's Ariane 5 rocket and the Russian Proton-M rocket.Overall, SpaceX's focus on rocket reuse has significantly reduced launch costs and changed the landscape of the space industry by making launches more affordable and accessible for a wide variety of missions.
- Problems you want to find answers
- We want to predict if the first stage of the falcon 9 will land correctly using the data provided by SpaceX.

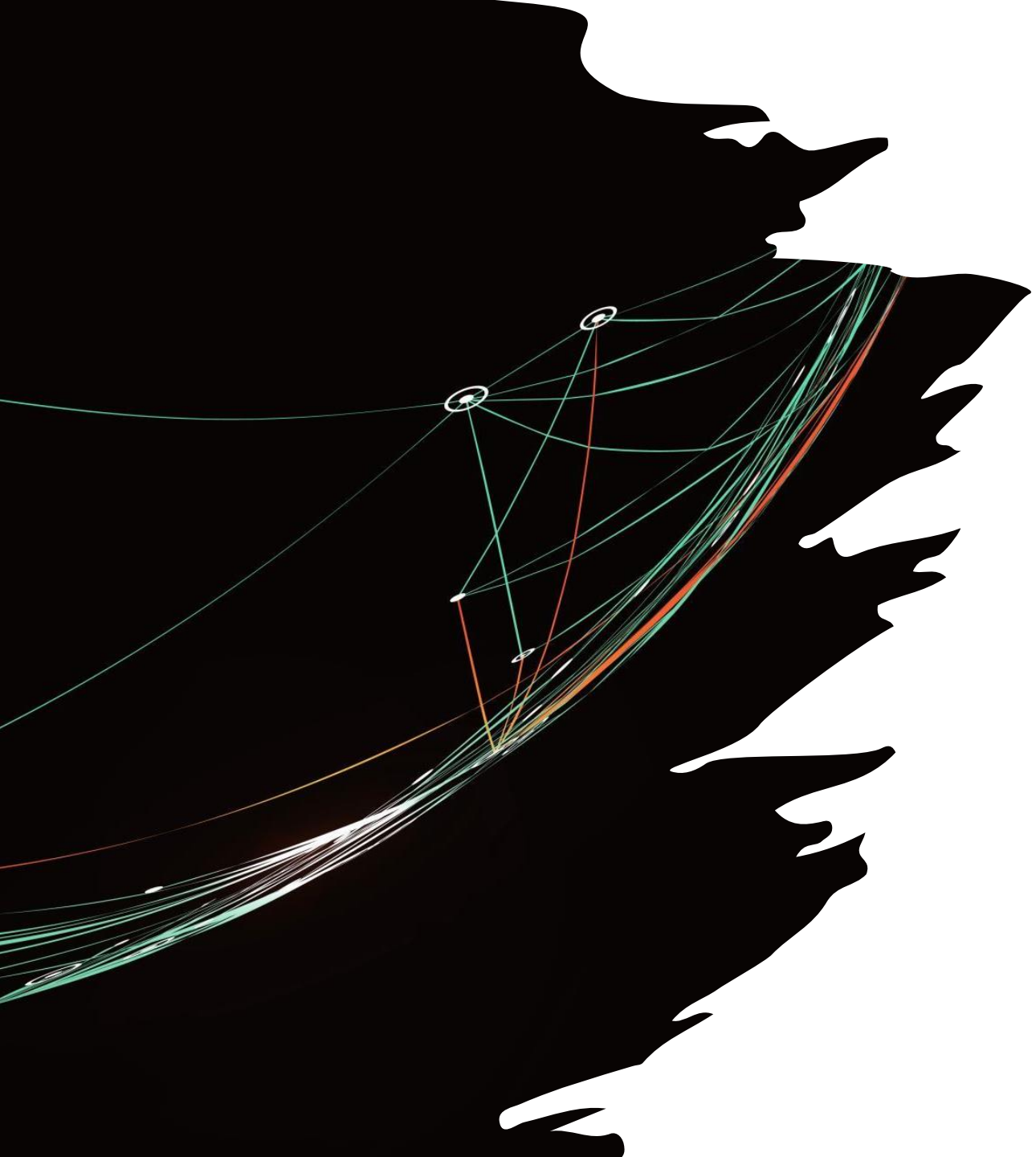
Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Describe how data was collected
- Perform data wrangling
  - Describe how data was processed
- 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

- Firstly, the necessary libraries for the data collection process were imported, later a series of functions necessary for the task were created and finally with the request module we obtained the information from the API.
- Another methodology for obtaining data, webscraping, was also used, in which we took care of collecting information from https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches, this information was used to complement the information found in the API.
- The data obtained in the API and webscraping have been converted into a Dataframe for better analysis.

Data Collection – SpaceX API

- We have used the url that has been provided to us, we have used the get method of the request library that returns a response object that contains information about the server's response.later we decode the response content as a Json and turn it into a Pandas dataframe .

- https://github.com/xb1t2/IBM-Proyect/blob/master/Space_Y.ipynb

Now let's start requesting rocket launch data from SpaceX API with the following URL:

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

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

```
static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_call_spacex_a
```

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

```
response.status_code
```

```
200
```

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

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

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

Next, request the HTML page from the above URL and get a `response` object

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.

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

```
'<!DOCTYPE html>\n<html class="client-nojs vector-feature-language-in-header-enabled vector-feature-language-in-main-pag
vector-feature-language-alert-in-sidebar-enabled vector-feature-sticky-header-disabled vector-feature-page-tools-disable
age-tools-pinned-disabled vector-feature-main-menu-pinned-disabled vector-feature-limited-width-enabled vector-feature-
nt-enabled" lang="en" dir="ltr">\n<head>\n<meta charset="UTF-8"/>\n<title>List of Falcon 9 and Falcon Heavy launches - W
```

Data Collection - Scraping

- Using Beautifulsoup, information has been collected from the data found in the url https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches , information has been extracted from the tables found.

- Github URL :https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches

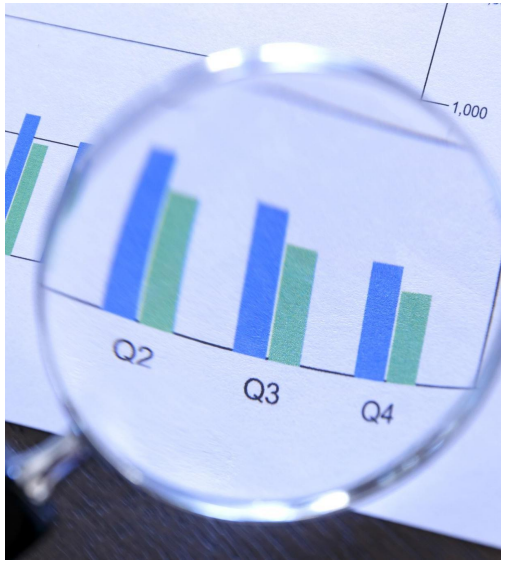
```
df.isnull().sum()/df.count()*100
```

```
FlightNumber      0.000
Date              0.000
BoosterVersion    0.000
PayloadMass       0.000
Orbit             0.000
LaunchSite        0.000
Outcome           0.000
Flights           0.000
GridFins          0.000
Reused            0.000
Legs              0.000
LandingPad       40.625
Block             0.000
ReusedCount       0.000
Serial            0.000
Longitude         0.000
Latitude          0.000
dtype: float64
```

Identify which columns are numerical and categorical:

```
df.dtypes
```

```
FlightNumber      int64
Date             object
BoosterVersion   object
PayloadMass     float64
Orbit            object
LaunchSite       object
Outcome          object
Flights           int64
GridFins           bool
Reused             bool
```



Data Analysis

Load Space X dataset, from last section.

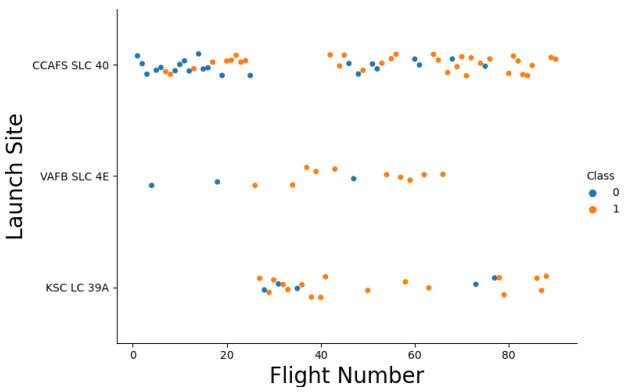
```
df=pd.read_csv("https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/dataset_part_1.csv")
df.head(10)
```

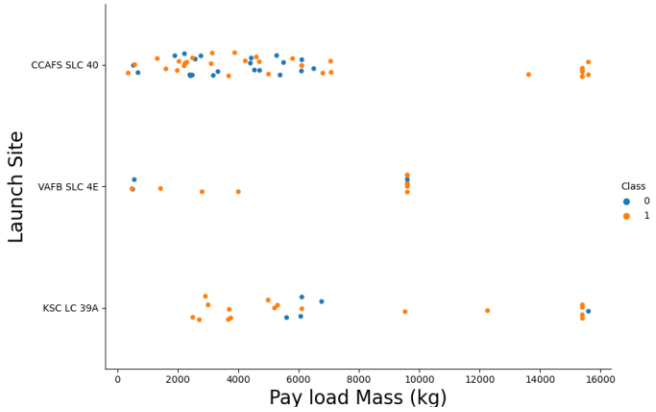
	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	Longitude
0	1	2010-06-04	Falcon 9	6104.959412	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0003	-80.5773
1	2	2012-05-22	Falcon 9	525.000000	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0005	-80.5773
2	3	2013-03-01	Falcon 9	677.000000	ISS	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0007	-80.5773
3	4	2013-09-29	Falcon 9	500.000000	PO	VAFB SLC 4E	False Ocean	1	False	False	False	NaN	1.0	0	B1003	-120.6108
4	5	2013-12-03	Falcon 9	3170.000000	GTO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B1004	-80.5773
5	6	2014-01-06	Falcon 9	3325.000000	GTO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B1005	-80.5773
6	7	2014-04-18	Falcon 9	2296.000000	ISS	CCAFS SLC 40	True Ocean	1	False	False	True	NaN	1.0	0	B1006	-80.5773
7	8	2014-07-14	Falcon 9	1316.000000	LEO	CCAFS SLC 40	True Ocean	1	False	False	True	NaN	1.0	0	B1007	-80.5773
8	9	2014-08-05	Falcon 9	4535.000000	GTO	CCAFS	None	1	False	False	False	NaN	1.0	0	B1008	-80.5773

Data Wrangling

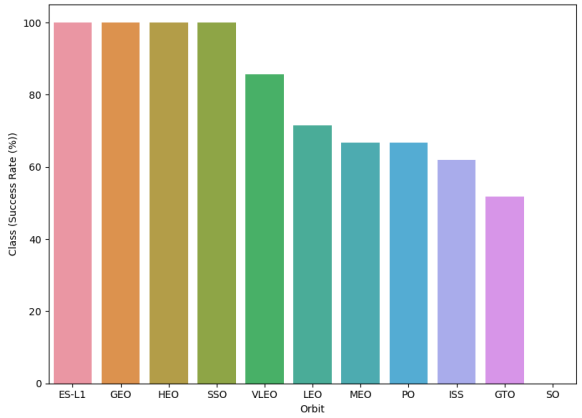
- To carry out the cleaning, transformation and preparation of data, the pandas module was used to read a dataset and obtain a dataframe, the null data was investigated and the data types of the dataframe were obtained, then different analysis metrics were calculated, such as for example the use of value_counts to know the number of occurrences of each orbit

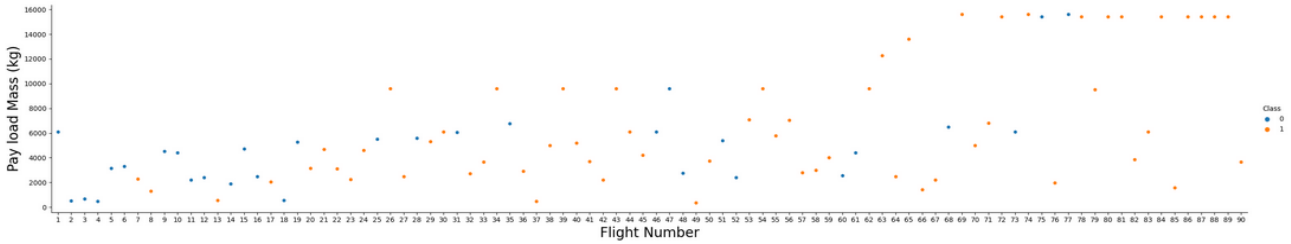
- GitHub URL :https://github.com/xb1t2/IBM-Proyect/blob/master/Data_Wrangling.ipynb





EDA with Data Visualization





- we have used those tables to see the effect that each independent variable has on the dependent variable.
- https://github.com/xb1t2/IBM-Proyect/blob/master/IBM-DS0321EN-SkillsNetwork_labs_module_2_jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb

EDA with SQL

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

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

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

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

- Display 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

EDA with SQL

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

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

```
%%sql
SELECT MIN(Date)
FROM SPACEXTBL
WHERE "Landing _Outcome" = 'Success (ground pad)';
```

- 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

```
%sql SELECT BOOSTER_VERSION,PAYLOAD_MASS__KG_ FROM SPACEXTBL WHERE "LANDING _OUTCOME" = 'Success (ground pad)' AND PAYLOAD_MASS__KG_ BETW
```

- https://github.com/xb1t2/IBM-Proyect/blob/master/SQL.ipynb

Build an Interactive Map with Folium

- Marks were created in the launch areas both in the eastern and western parts of the country in the form of a circle and the distances from the launch area to different parts that may be of interest and relevant areas were measured, lines were used to carry out the task
- GitHub URL : https://github.com/xb1t2/IBM-Proyect/blob/master/IBM-DS0321EN-SkillsNetwork_labs_module_3_lab_jupyter_launch_site_location.jupyterlite.ipynb

Build a Dashboard with Plotly Dash

- Summarize what plots/graphs and interactions you have added to a dashboard
    - Incorporating a drop-down input component for selecting launch sites.
    - Adding a callback function to render a success-pie-chart based on the selected site from a dropdown menu.
    - Implementing a callback function to display a scatter plot of the success payload based on the user's selection in a dropdown menu.

- https://github.com/xb1t2/IBM-Proyect/blob/master/IBM-DS0321EN-SkillsNetwork_labs_module_3_lab_jupyter_launch_site_location.jupyterlite.ipynb

Predictive Analysis (Classification)

- The phases that have been followed have been:
- selecting training labels , the class column is created, the data is standardized and the dataset is divided into training and test.The best hyperparameters for each model are searched for and the algorithm with the best performance results is selected.
- To find the best model, the parameters for each machine learning algorithm are created, then the Gridsearch function is instantiated for each model. This function exhaustively searches for the best hyperparameters for each instance of each model according to the parameters provided to each one

Predictive Analysis (Classification)

- The table presented below corresponds to the precision presented by each model with the hyperparameters already adjusted.

	Test_Accuracy
KNN	0.833333
Decision_tree	0.944444
SVM	0.833333
Log_Reg	0.833333

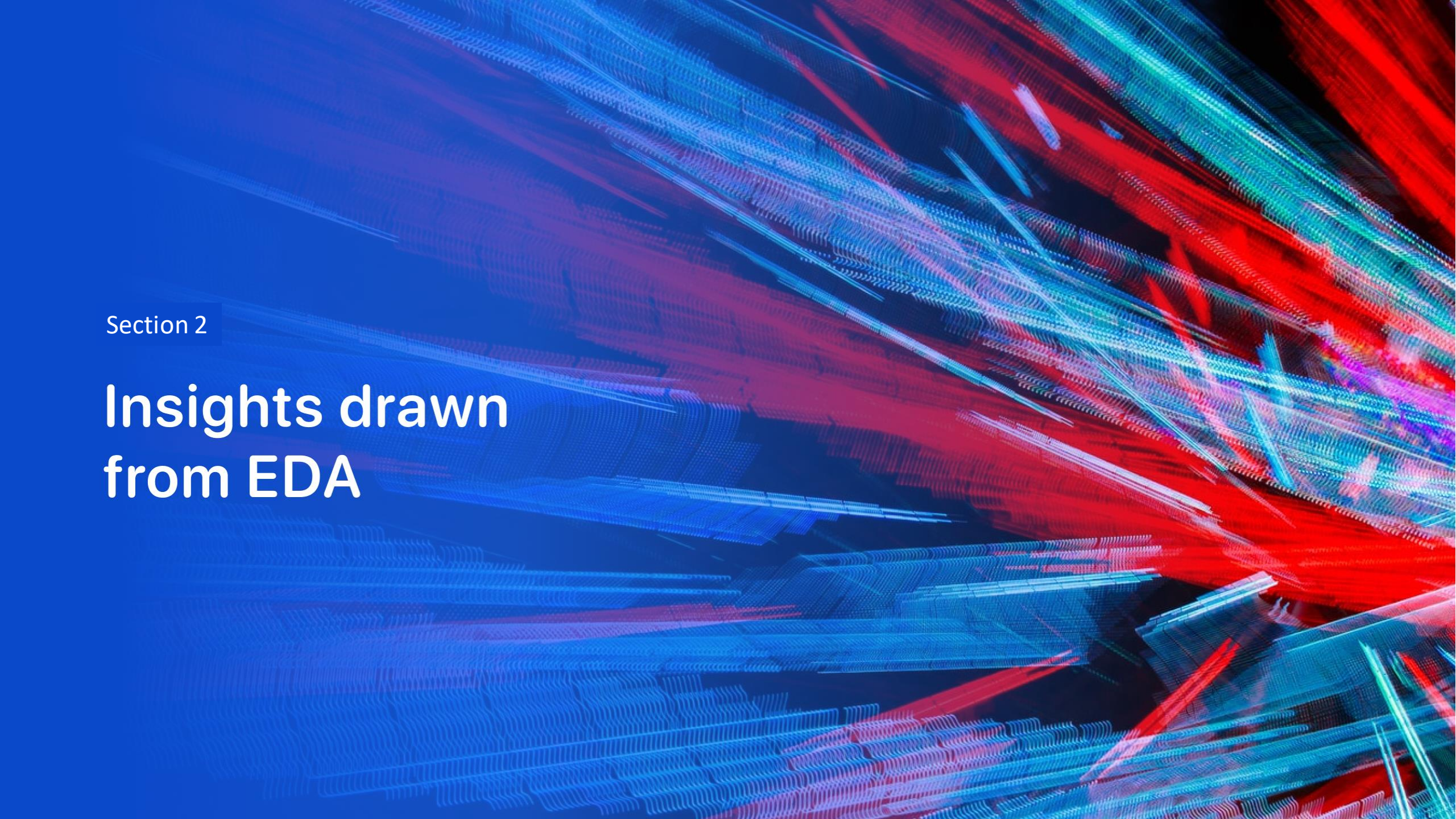
- //github.com/xb1t2/IBM-Proyect/blob/master/IBM-DS0321EN-SkillsNetwork_labs_module_4_SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

Results

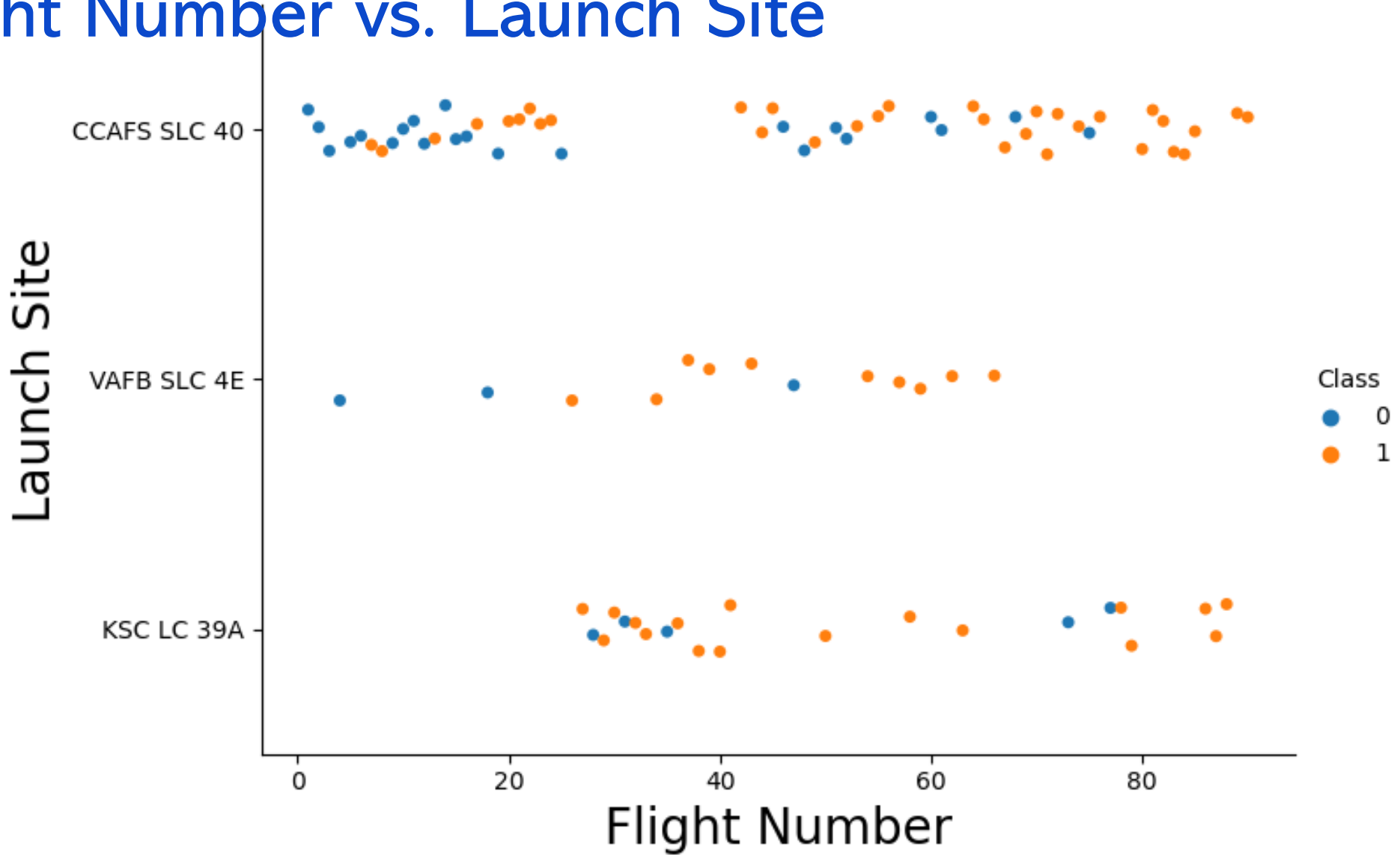
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

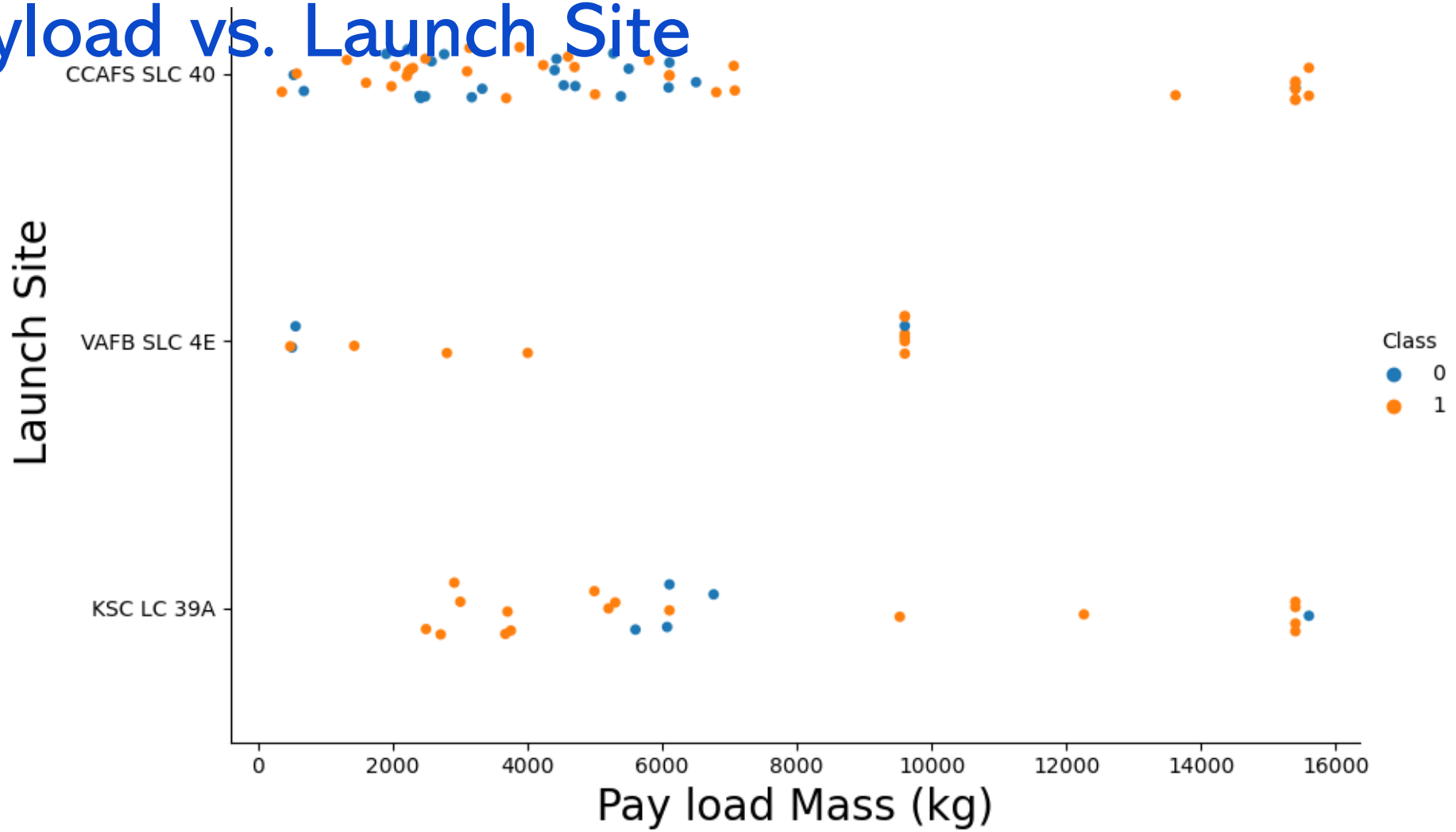
Insights drawn from EDA



Flight Number vs. Launch Site

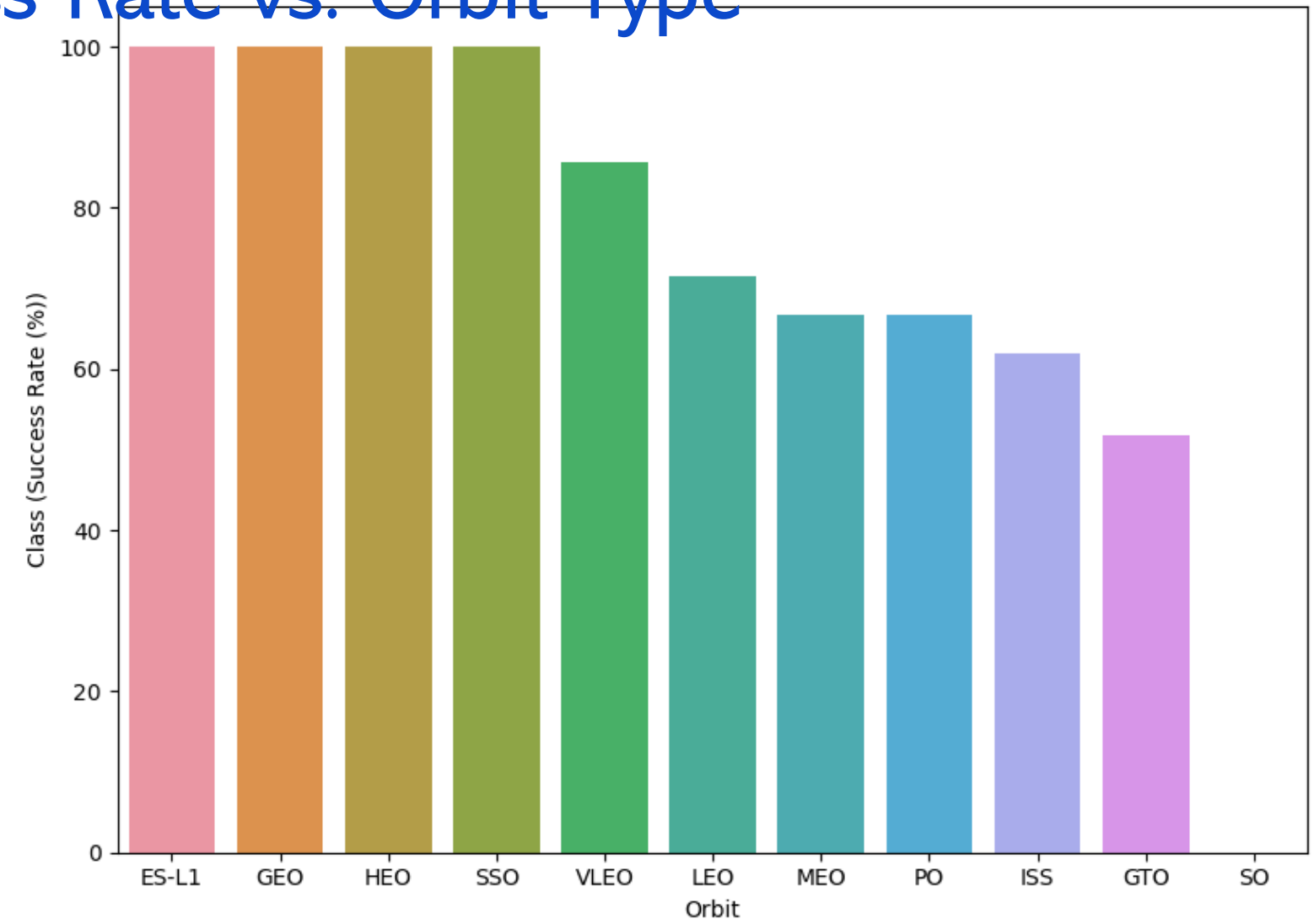


Payload vs. Launch Site

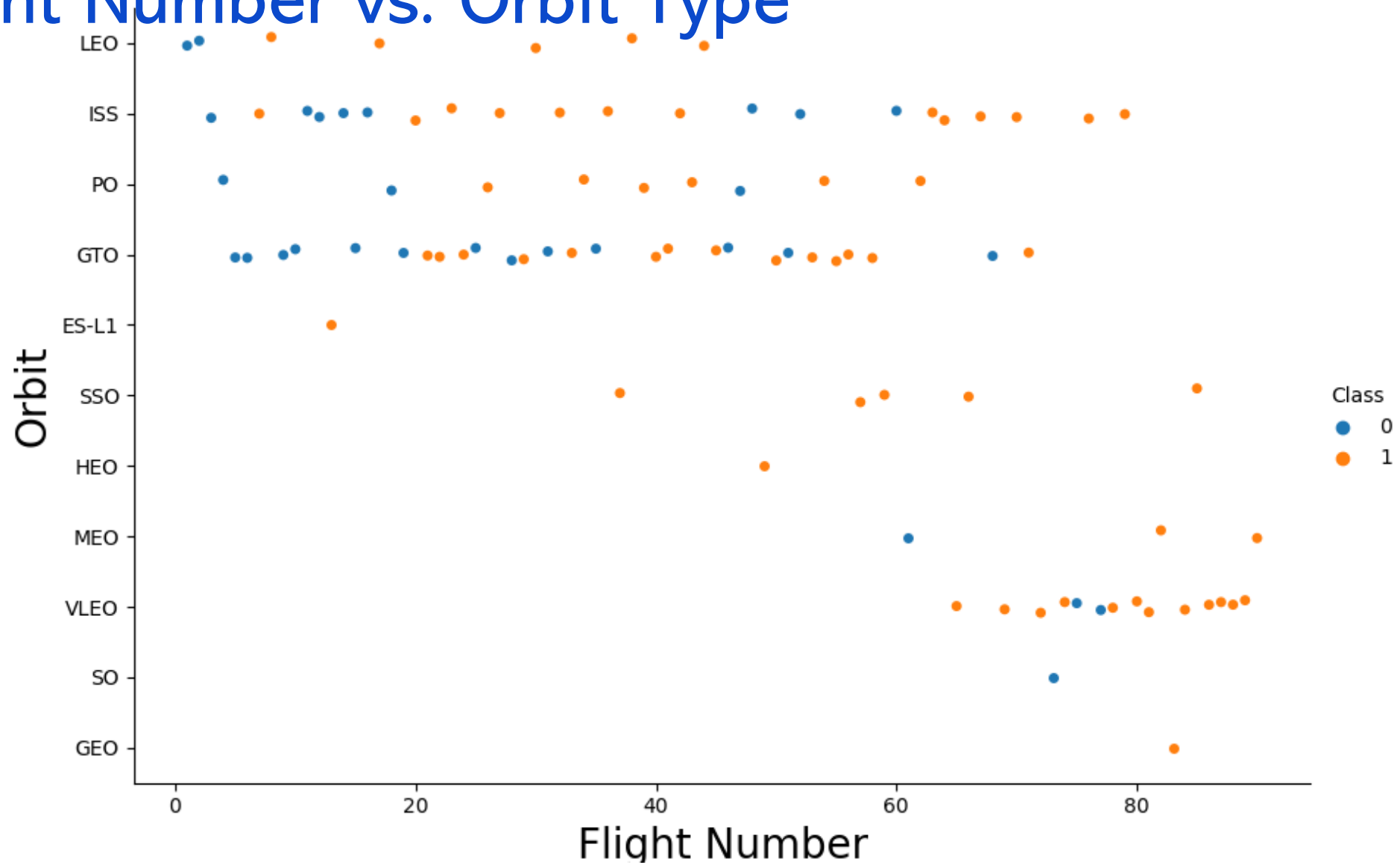


Now if you observe Payload Vs. Launch Site scatter point chart you will find for the VAFB-SLC launchsite there are no rockets launched for heavypayload mass(greater than 10000).

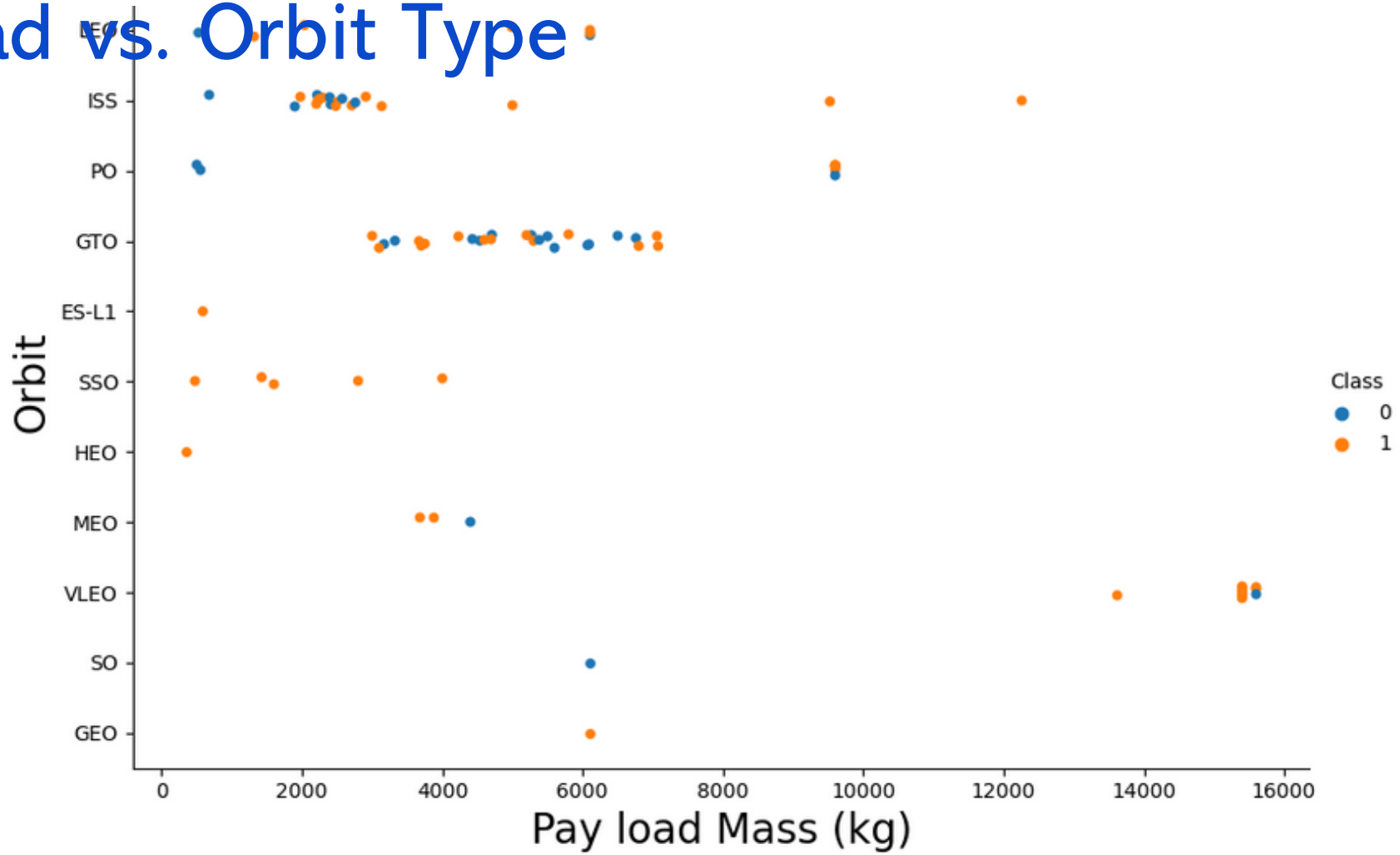
Success Rate vs. Orbit Type



Flight Number vs. Orbit Type



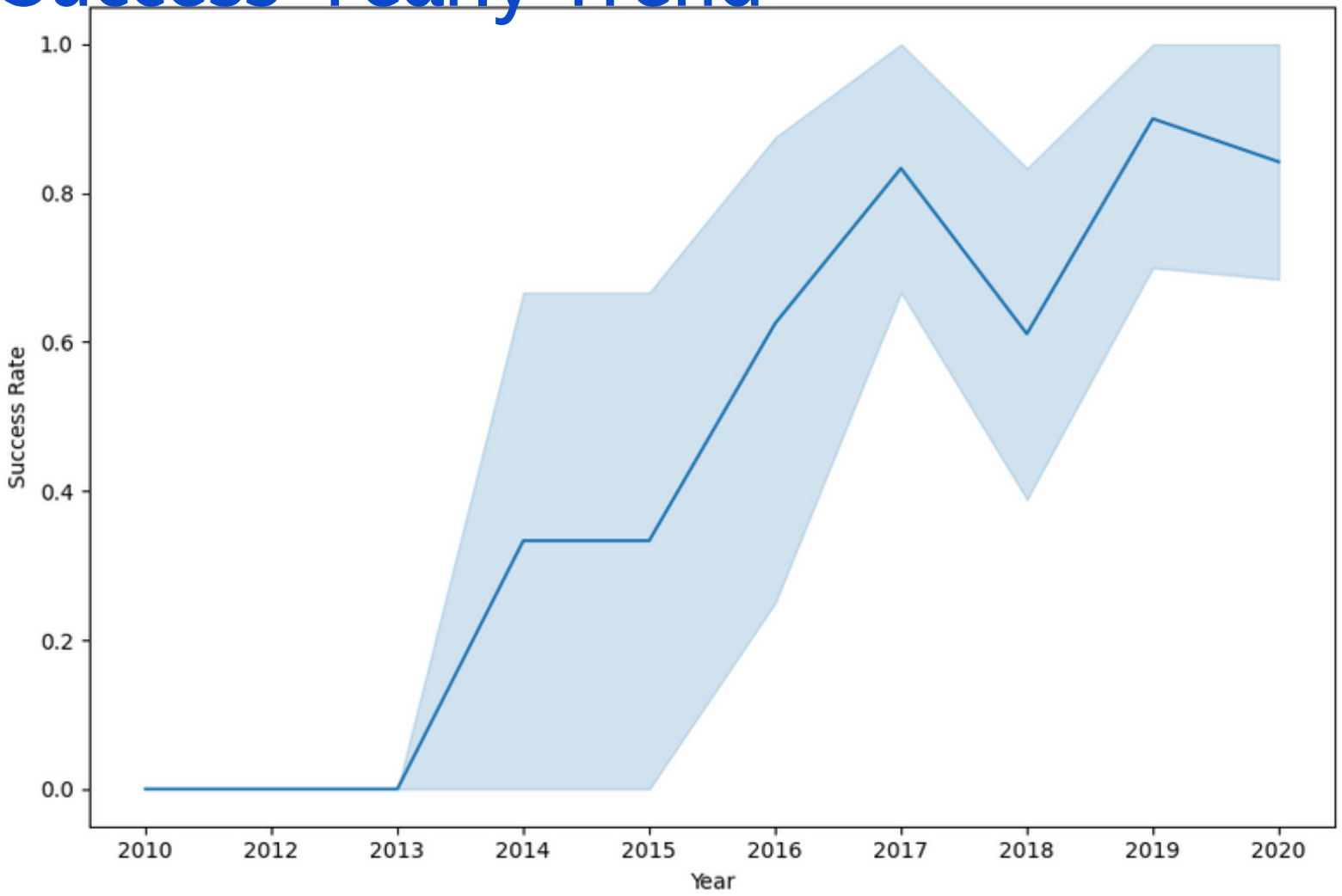
Payload vs. Orbit Type



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

However for GTO we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission) are both there here.

Launch Success Yearly Trend



you can observe that the sucess rate since 2013 kept increasing till 2020

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

```
 * sqlite:///my_data1.db
Done.
```

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

All Launch Site Names

- Find the names of the unique launch sites.
- with this query we get the unique launch sites

Launch Site Names Begin with 'CCA'

- With this query we limit the launch sites where CCA appears to 5

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

```
 * sqlite:///my_data1.db
Done.
```

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing _Outcome
04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
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	Success	Failure (parachute)
22-05-2012	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
01-03-2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

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

```
 * sqlite:///my_data1.db
Done.
```

SUM(PAYLOAD_MASS__KG_)
45596

- with this query we add the payload mass column where customer = Nasa(CRS)

Average Payload Mass by F9 v1.1

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

```
 * sqlite:///my_data1.db
Done.
```

AVG(PAYLOAD_MASS__KG_)
2928.4

- ith this query we make the average of the payload mass column where the boosterversion column is F9

First Successful Ground Landing Date

```
%%sql
SELECT MIN(Date)
FROM SPACEXTBL
WHERE "Landing _Outcome" = 'Success (ground pad)';
```

```
 * sqlite:///my_data1.db
Done.
```

MIN(Date)
01-05-2017

- In this query we use the min function to select the oldest successful landing date.

```
%sql SELECT DISTINCT BOOSTER_VERSION,PAYLOAD_MASS__KG_ FROM SPACEXTBL WHERE "LANDING__OUTCOME" = 'Success (ground pad)' AND PAYLOAD_MASS_
```

```
 * sqlite:///my_data1.db
Done.
```

Booster_Version	PAYLOAD_MASS__KG_

Successful Drone Ship Landing with Payload between 4000 and 6000

Query error

Total Number of Successful and Failure Mission Outcomes

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

```
 * sqlite:///my_data1.db
Done.
```

Mission_Outcome	TOTAL_NUMBER
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

- MISSION_OUTCOME from the "SPACEXTBL" table and counts how many times each value in this column is repeated using the COUNT() function. It then uses the GROUP BY clause to group the results by the value of the MISSION_OUTCOME column.

Boosters Carried Maximum Payload

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

```
 * sqlite:///my_data1.db
Done.
```

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

- The query uses a subquery to get the maximum value of "PAYLOAD_MASS__KG_" in the table "SPACEXTBL" and then compares the value of "PAYLOAD_MASS__KG_" for each row in the table against that maximum value. If a value of "PAYLOAD_MASS__KG_" matches the maximum value, the query returns the value of the "BOOSTER_VERSION" column for that row.

2015 Launch Records

- Selects the "Date", "booster_version" and "Landing_Outcome" columns from the "SPACEXTBL" table where the value of the "Landing_Outcome" column is "Failure (drone ship)" and the year in the "Date" column is "2015" . The query also uses the substr() function to extract the characters corresponding to the month of the date in the "Date" column and names it as "month".

```
%%sql
SELECT substr(Date, 4, 2) as month,booster_version,"Landing _Outcome"
from SPACEXTBL where "Landing _Outcome"
='Failure (drone ship)' and substr(Date,7,4)='2015'
```

```
 * sqlite:///my_data1.db
Done.
```

month	Booster_Version	Landing _Outcome
01	F9 v1.1 B1012	Failure (drone ship)
04	F9 v1.1 B1015	Failure (drone ship)

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

```
%%sql
SELECT "Landing _Outcome",count("Landing _Outcome")as LANDING_OUTCOME_COUNT,DATE
from SPACEXTBL where substr(Date,7,4) || substr(Date,4,2) || substr(Date,1,2) between '20100604'
and '20170320'and "Landing _Outcome" like "Success%"
group by "Landing _Outcome" order by count("Landing _Outcome") desc
```

```
 * sqlite:///my_data1.db
Done.
```

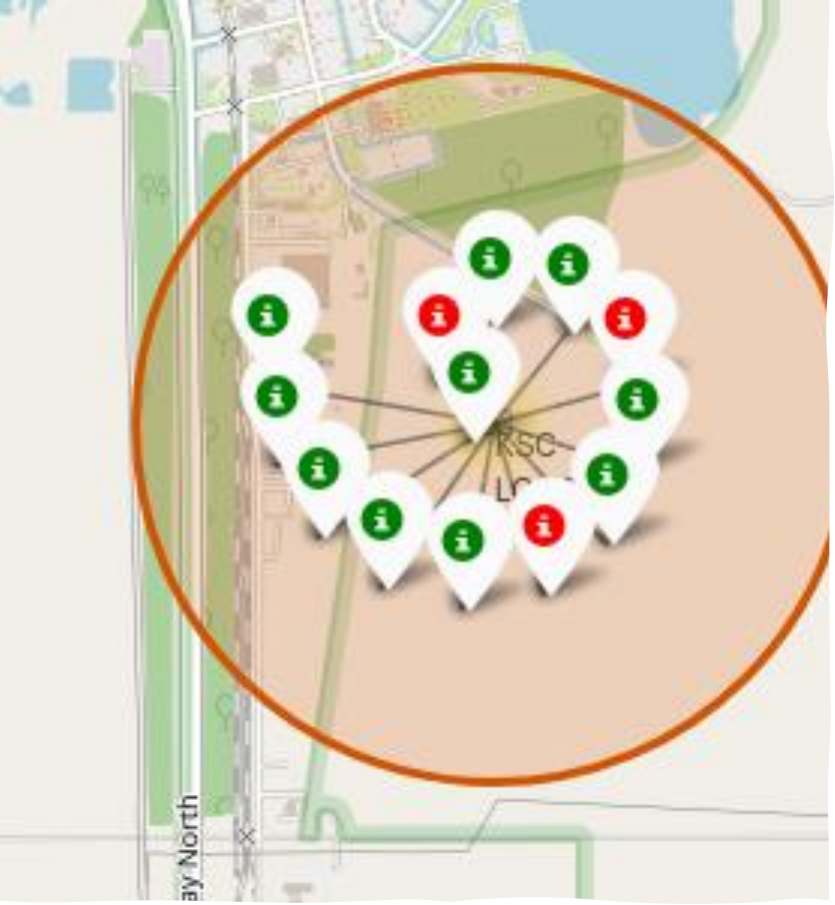
Landing _Outcome	LANDING_OUTCOME_COUNT	Date
Success (drone ship)	5	08-04-2016
Success (ground pad)	3	22-12-2015

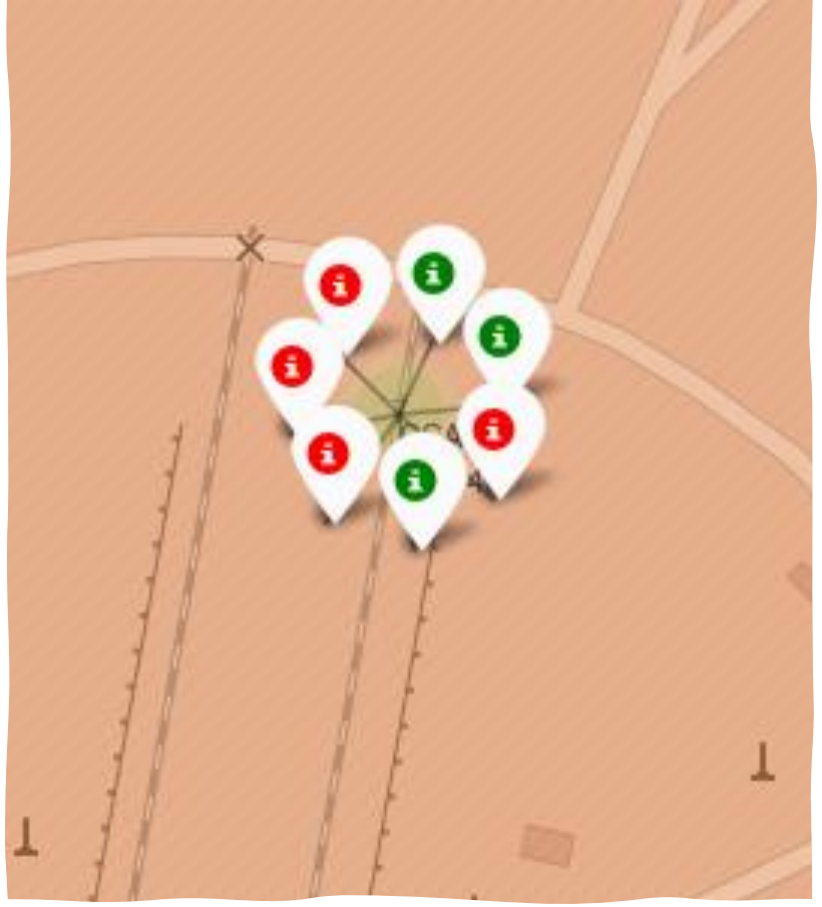
- returns a list of landing results beginning with "Success" during a specified time period, along with the number of times each result was recorded in the SPACEXTBL table

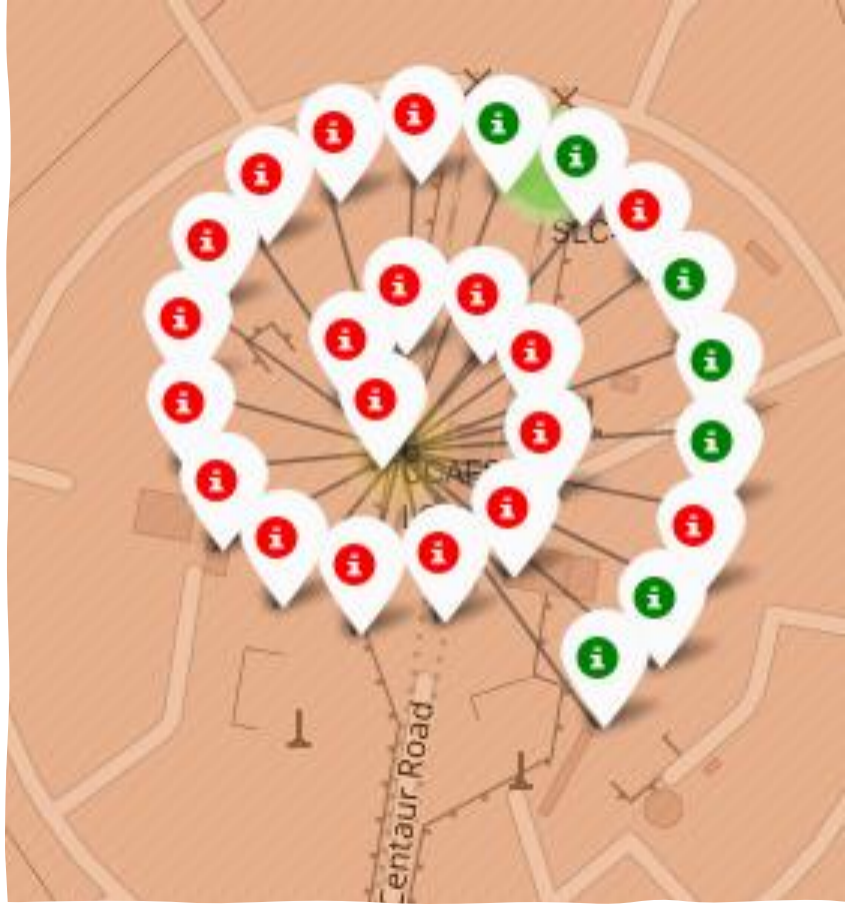

Section 3
Launch Sites
Proximities Analysis

LAUNCH SITES

- In this image are the launch sites

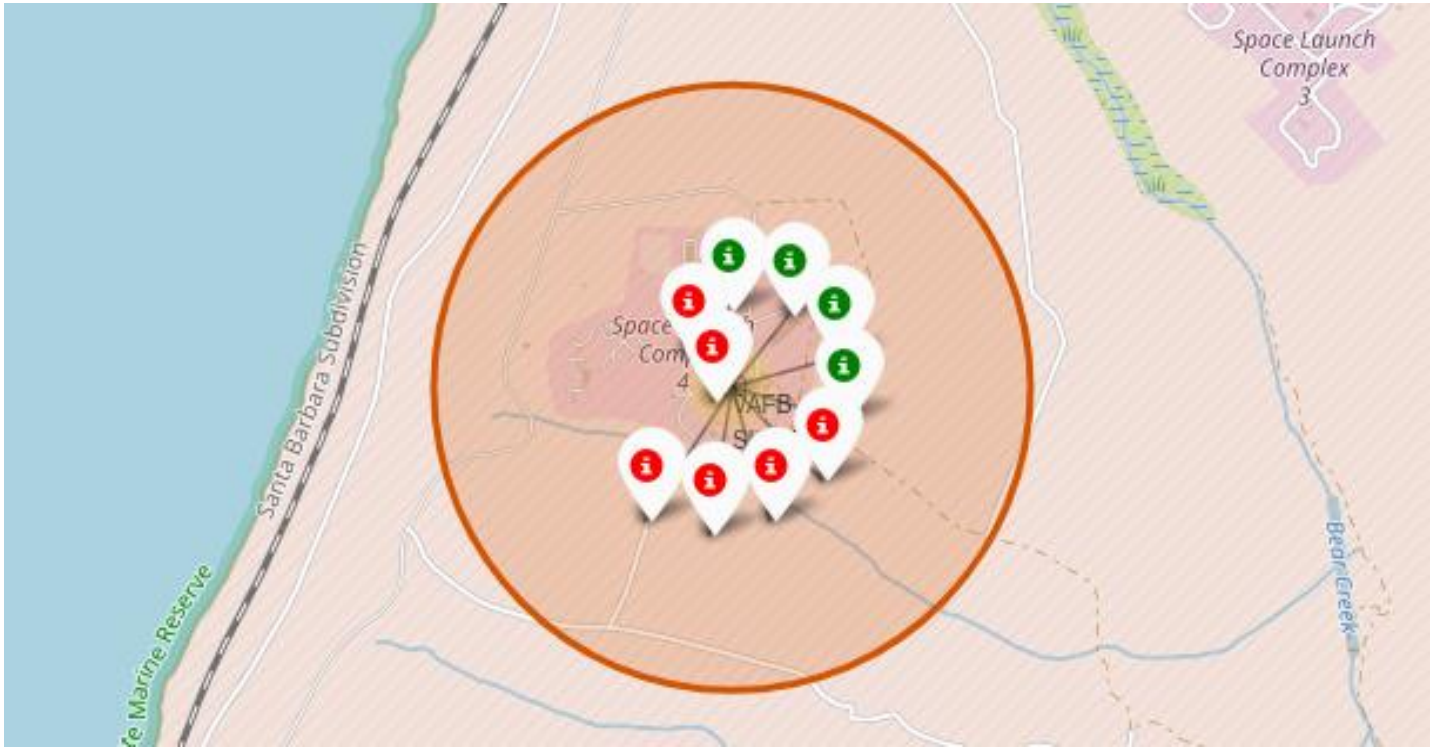






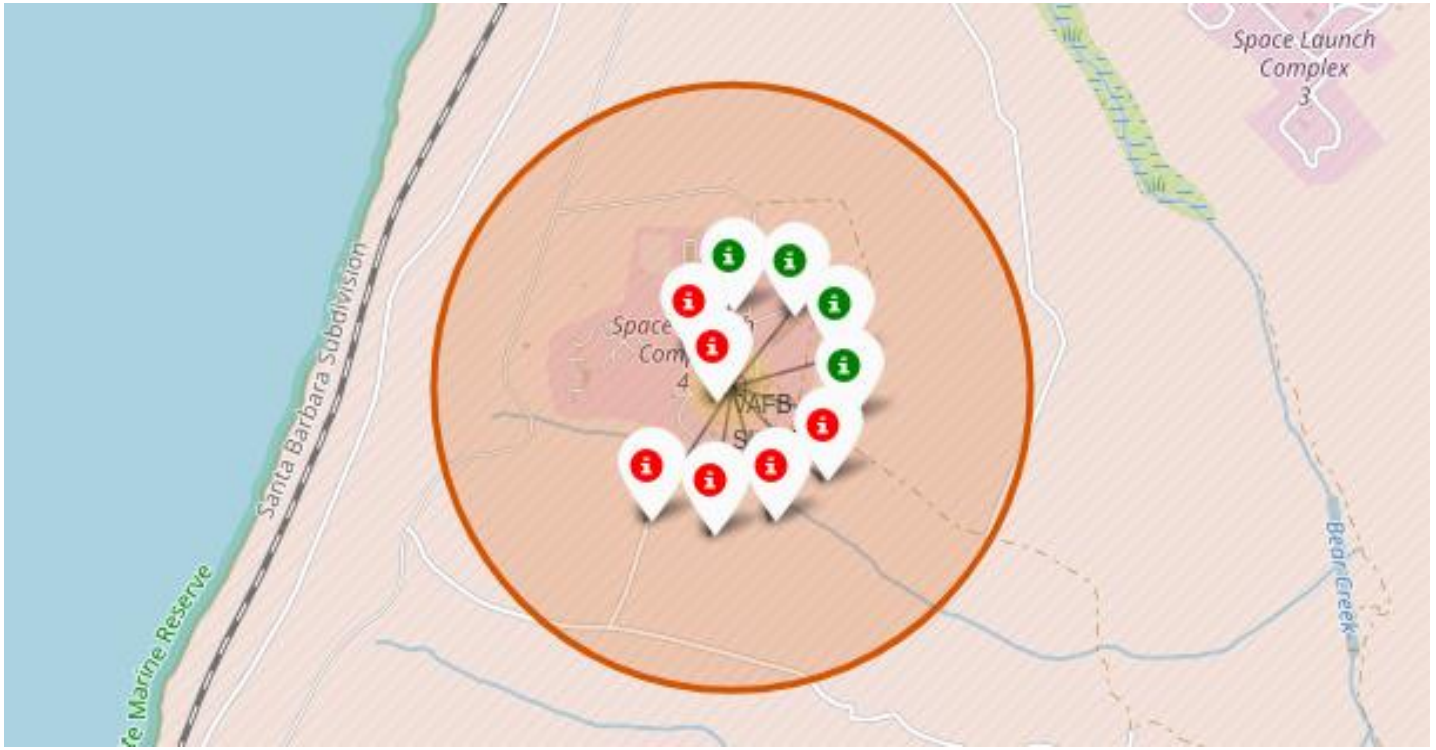
LAUNCH OUTCOMES WITH COLOR MARKERS

The launch site KSC LC-39A has a comparatively higher success rate when compared to CCAFS SLC-40 and CCAFS LC-40



LAUNCH OUTCOMES WITH COLOR MARKERS(California)

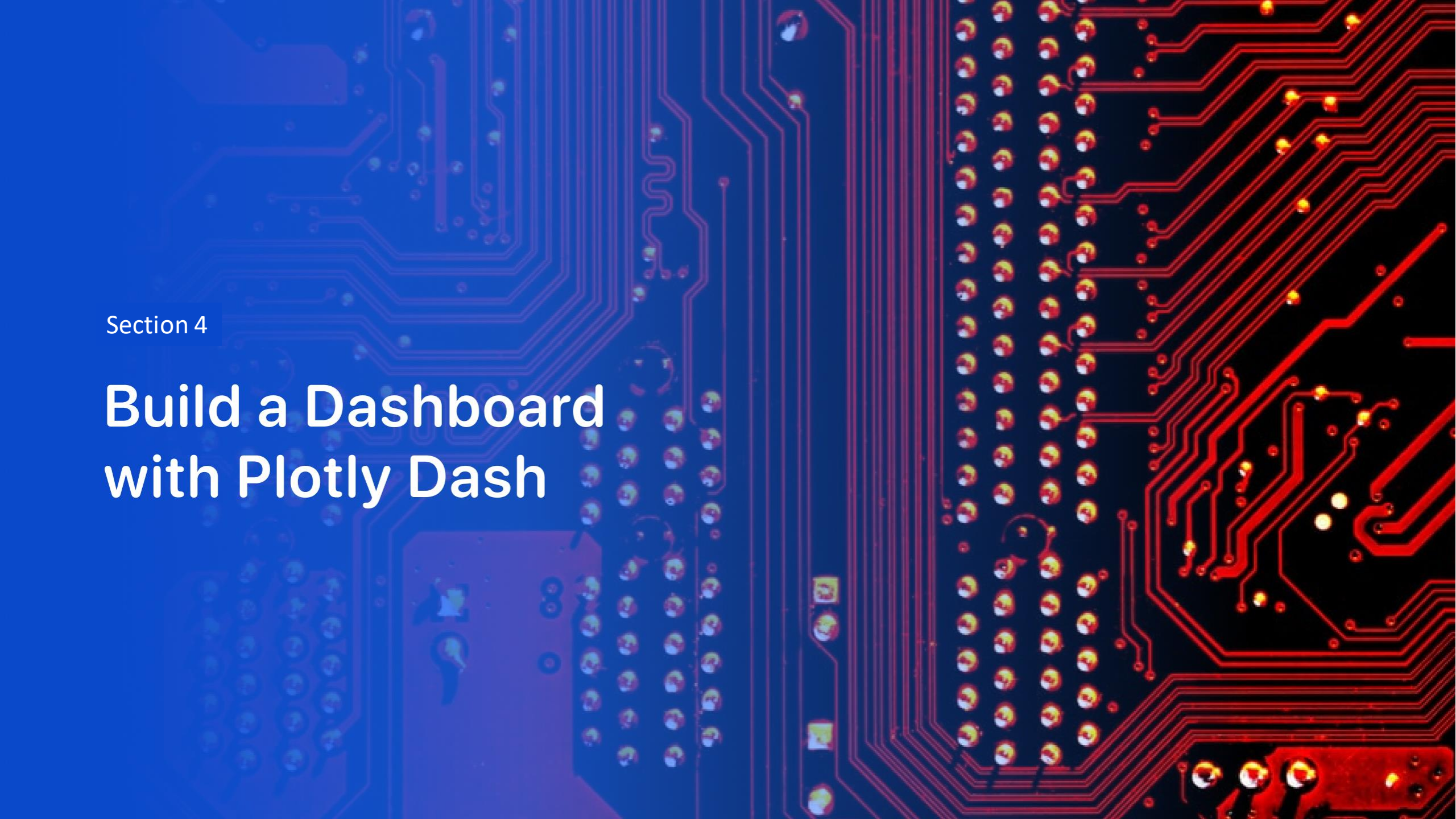
When it comes to launch sites, the KSC LC-39A site on the Eastern Coast of Florida has a higher success rate compared to the VAFB SLC-4E site on the West Coast (California), which has a lower success rate 40% .



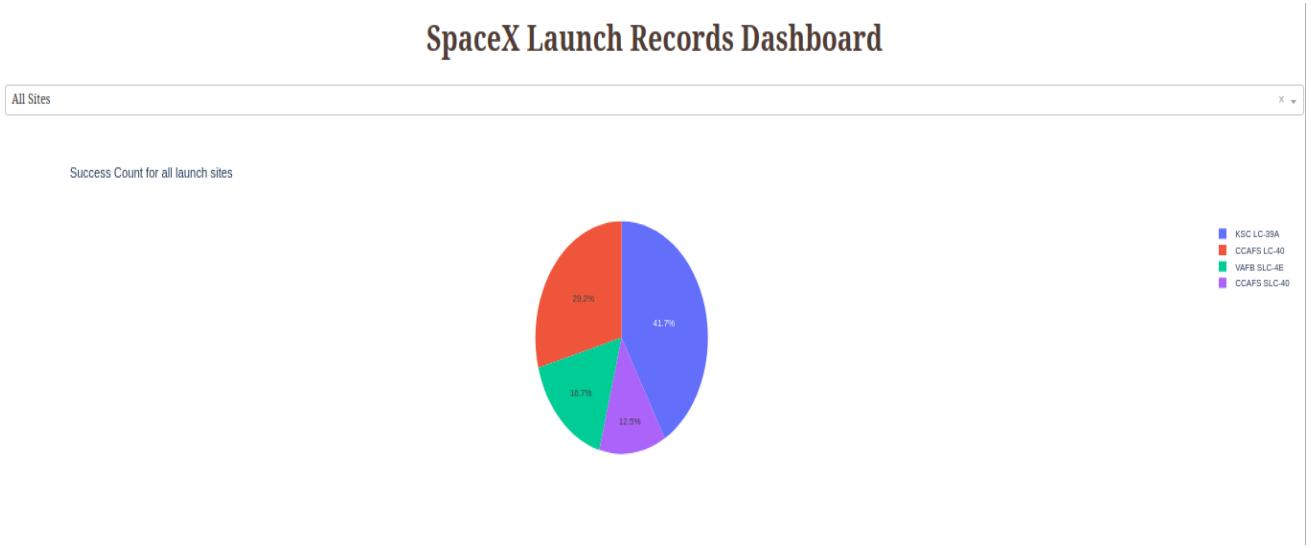
LAUNCH OUTCOMES WITH COLOR MARKERS(California)

When it comes to launch sites, the KSC LC-39A site on the Eastern Coast of Florida has a higher success rate compared to the VAFB SLC-4E site on the West Coast (California), which has a lower success rate 40% .

DISTANCE LINES

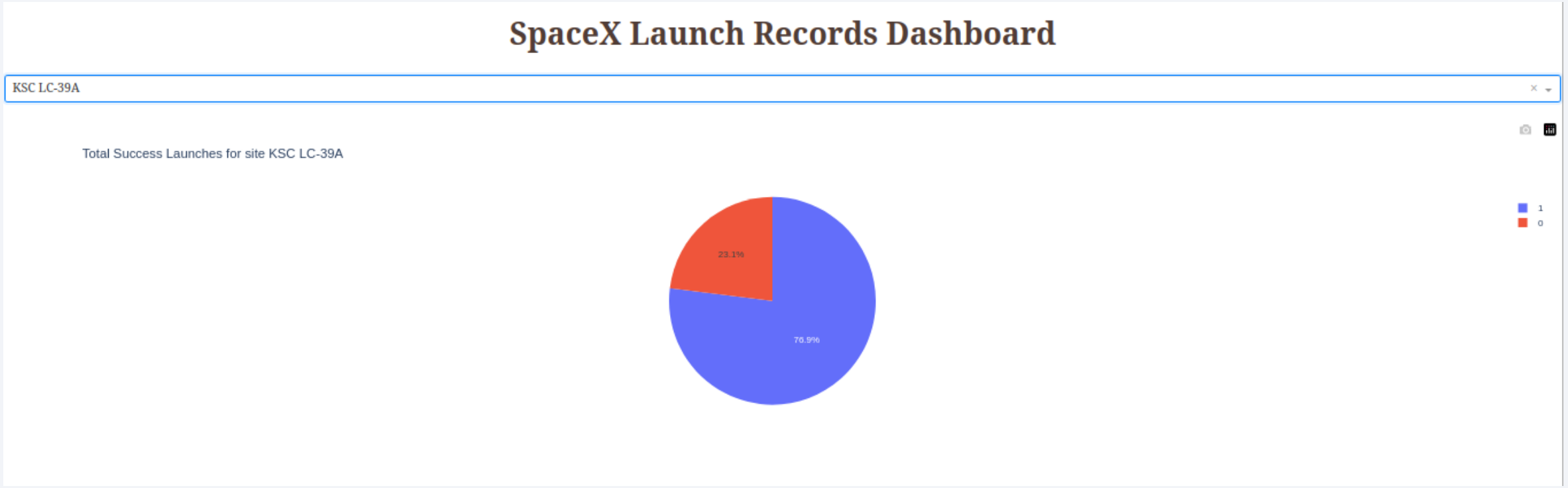

Section 4
Build a Dashboard with Plotly Dash

PIE-CHART OF LAUNCH SUCCESS

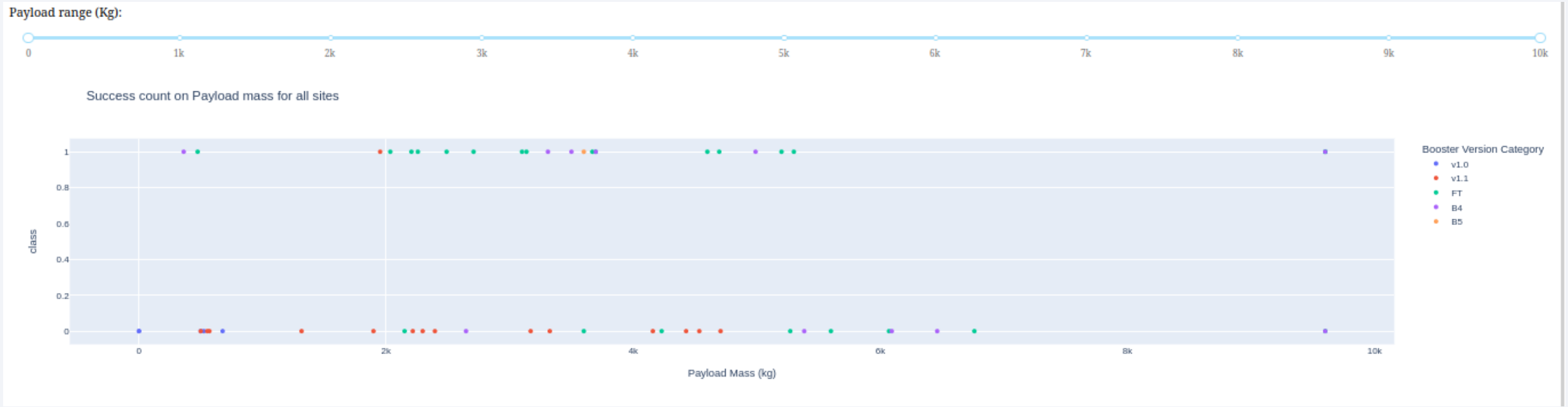


You can see in the graph the success rate according to the launch zone, the largest launch success ratio is taken by KSC with 41.7

PIE-CHART BEST LAUNCH SITE



PAYLOAD VS LAUNCH OUTCOME



In Launch site CCAFS LC-40, the FT booster version has the highest success rate for a payload mass greater than 2000 kg. In other words, the FT booster model is the most successful at launching payloads weighing over 2000 kg from Launch site CCAFS LC-40

Section 5

Predictive Analysis (Classification)

Classification Accuracy

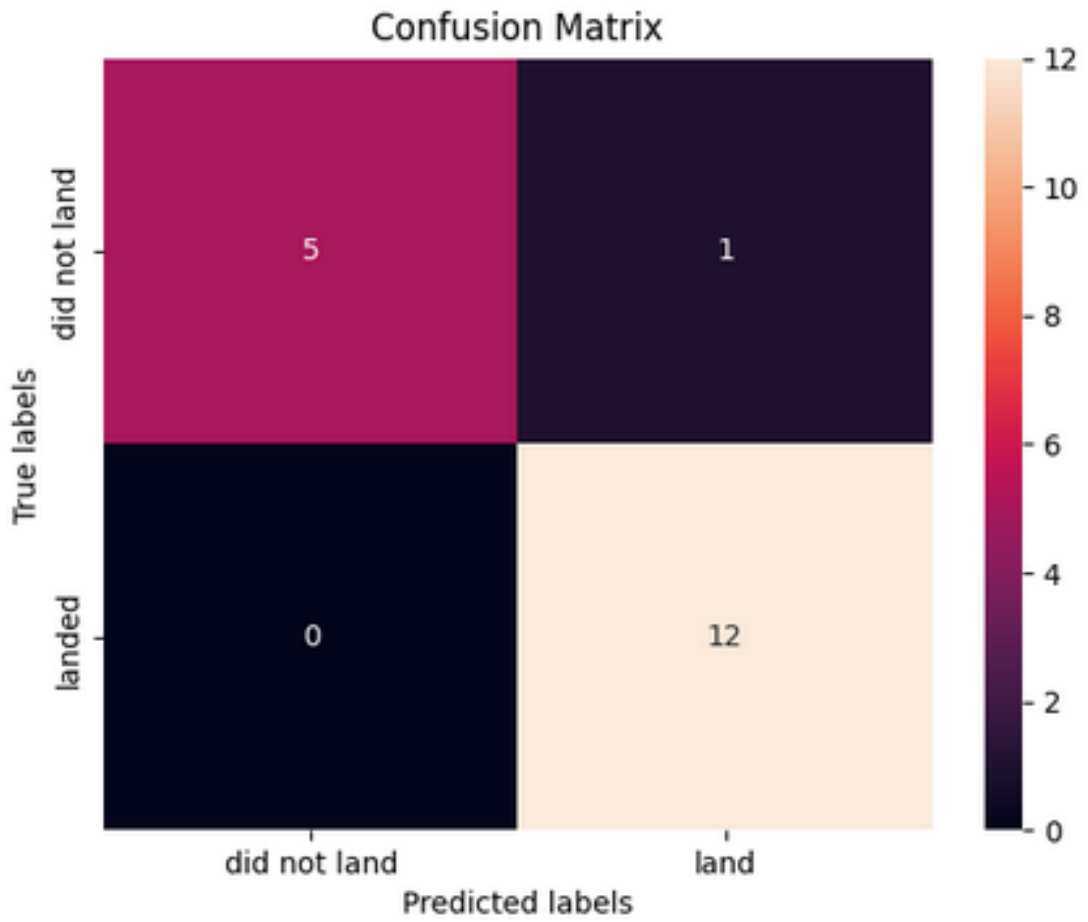
```
df= pd.DataFrame({'KNN':knn_cv.score(X_test, Y_test),
                  'Decision_tree':tree_cv.score(X_test, Y_test),
                  'SVM':svm_cv.score(X_test, Y_test),
                  'Log_Reg':logreg_cv.score(X_test, Y_test)},index=['Test_Accuracy']).transpose()

df
```

	Test_Accuracy
KNN	0.833333
Decision_tree	0.944444
SVM	0.833333
Log_Reg	0.833333

the decision tree seems to have a higher hit rate than the other algorithms

Confusion Matrix

the decision tree has a very high rate of success, it seems that it only fails on a false positive



Conclusions

- We can conclude that the time factor is a fundamental variable when it comes to obtaining a positive result. On the other hand, we also have to mention the importance of the areas in which the launch sites are located and other important variables, such as the incident load, significantly in whether a landing is successful or fails.

Thank you!