IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Wandy Suhendra
17 Jan 2025

Outline

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

Executive Summary

- Summary of methodologies

By accessing and manipulating data available online provided by SpaceX and Wikipedia, I managed to build models to predict the outcome of launches.

- Summary of all results

- We can use either one of 4 models built to predict the outcome of launching with expected 83% accuracy.
- Best performing booster version are FT and V1.0 Point 3.
- Best performing launch site is LC-39A with 76% success rate. From the map we can see that it is not the nearest to the coast line but still quite near.
- ES-L1, GEO, HEO and SSO orbits show a 100% success rate.
- The heavier the payloads the better the chance for success, especially 10k kg or higher.
- Experience matters, as the more launches performed the more success rate SpaceX gets.

Introduction

Project background and context

Space X advertises that Falcon 9 rocket launches on its website with a cost of 62 million dollars much cheaper than 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. With this in mind, we then created a machine learning pipeline to predict if the first stage will land based on data collected

Problems to be answered

My assignment is to find out factors that determine the success of rocket launchings and find a way to predict the success with high enough accuracy.

Section 1

Methodology

Methodology

Executive Summary

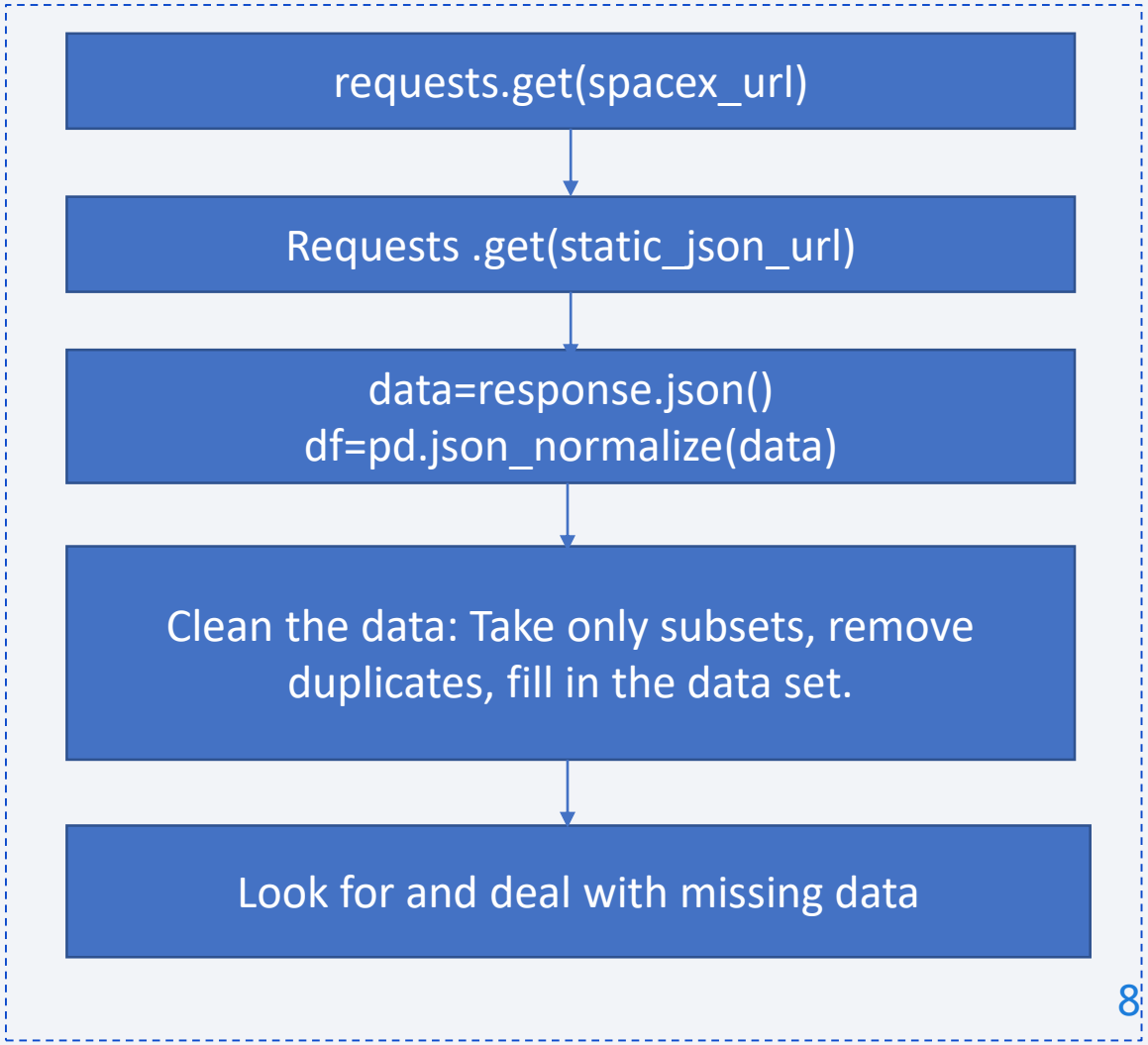
- Data collection methodology:
  - Data sources: SpaceX website available data and Wikipedia (JSON, html, etc.).
  - Collection methods: REST API, web scrapping, etc.
- Perform data wrangling
  - Cleaning the data by: finding and removing duplicates and finding and imputing missing values
  - Normalizing the data
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - We build and compare 4 models: logistic regression, support vector machine, decision tree classifier and K-nearest Neighbor classifier.
  - By comparing their accuracy of predictions and data fitting,

Data Collection

Description as to how data sets were collected will be provided in the following slides in form of narrative data collection process and flowcharts, please refer to coming slides.

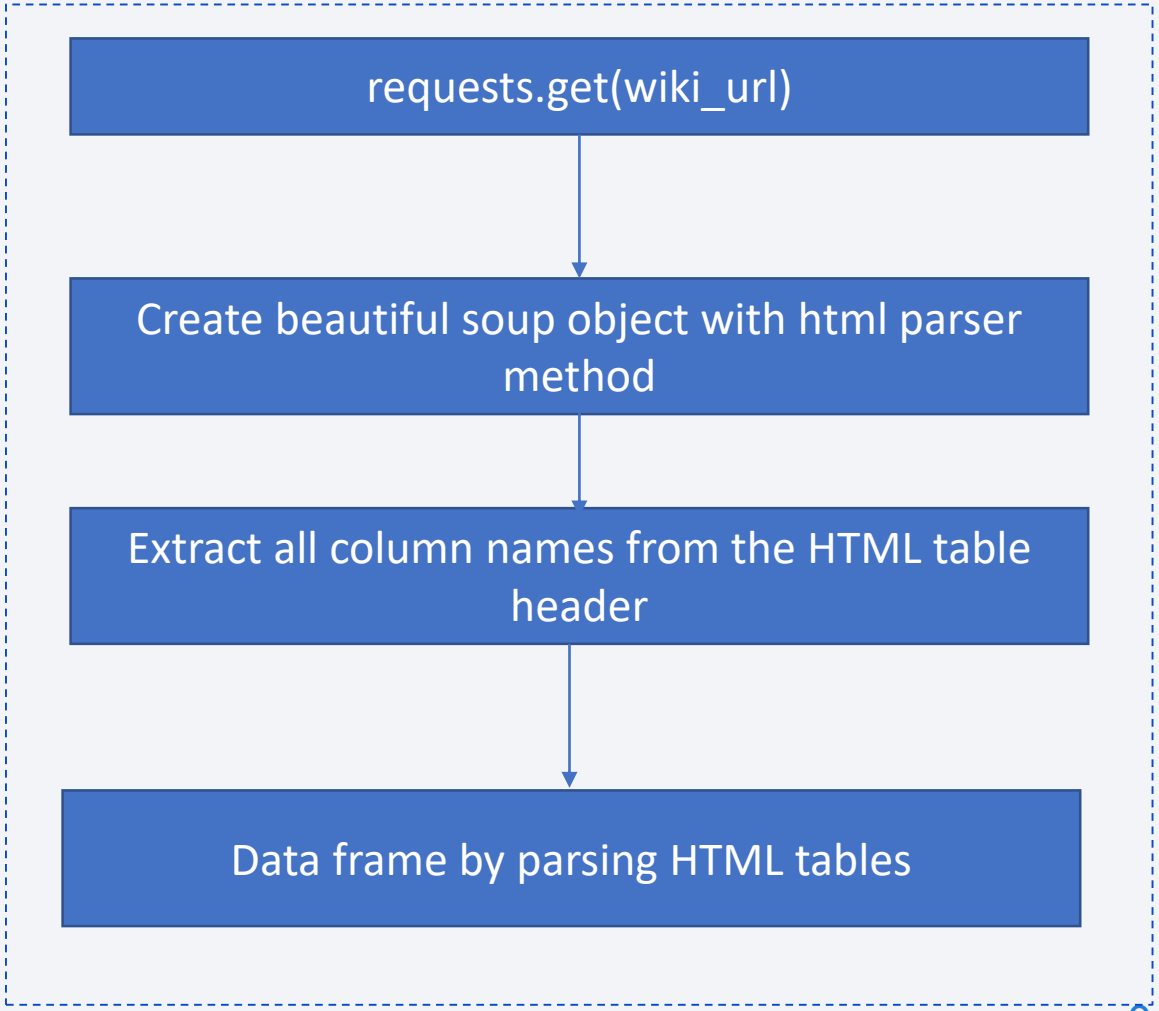
Data Collection - SpaceX API

- We target https://api.spacexdata.com/v4/launches/past to get past lauch data
- We perform GET request and load it into “response”.
- Calling the data in JSON format from “response” using request.get(JSON_url).
- Use .json_normalize() to make the JSON format fit to a DataFrame.
- take a subset of our dataframe keeping only the features we want and the flight number, and date_utc.
- Filter the dataframe to only include Falcon 9 launches
- Look for and replace the missing payload data with its mean()
- Leave the missing Landing Pad missing data for the time being.
- Result can be seen in: https://github.com/wsuhendra/IBM-Data-Science/blob/2c363247dd9181c97b3d262a64d947558806f414/jupyter-labs-spacex-data-collection-api.ipynb



Data Collection - Scraping

- Request the Falcon9 Launch Wiki page from its Wikipedia URL
- Create Beautifulsoup object: soup = BeautifulSoup(html_doc, 'html.parser')
- Extract all column names from HTML table heder
- Create a data frame by parsing the launch HTML tables. We created an empty dictionary with keys from the extracted column names, fill it in with parsing data and then convert the dictionary into a Pandas dataframe
- Result can be seen in: https://github.com/wsuhendra/IBM-Data-Science/blob/2c363247dd9181c97b3d262a64d947558806f414/jupyter-labs-webscraping.ipynb

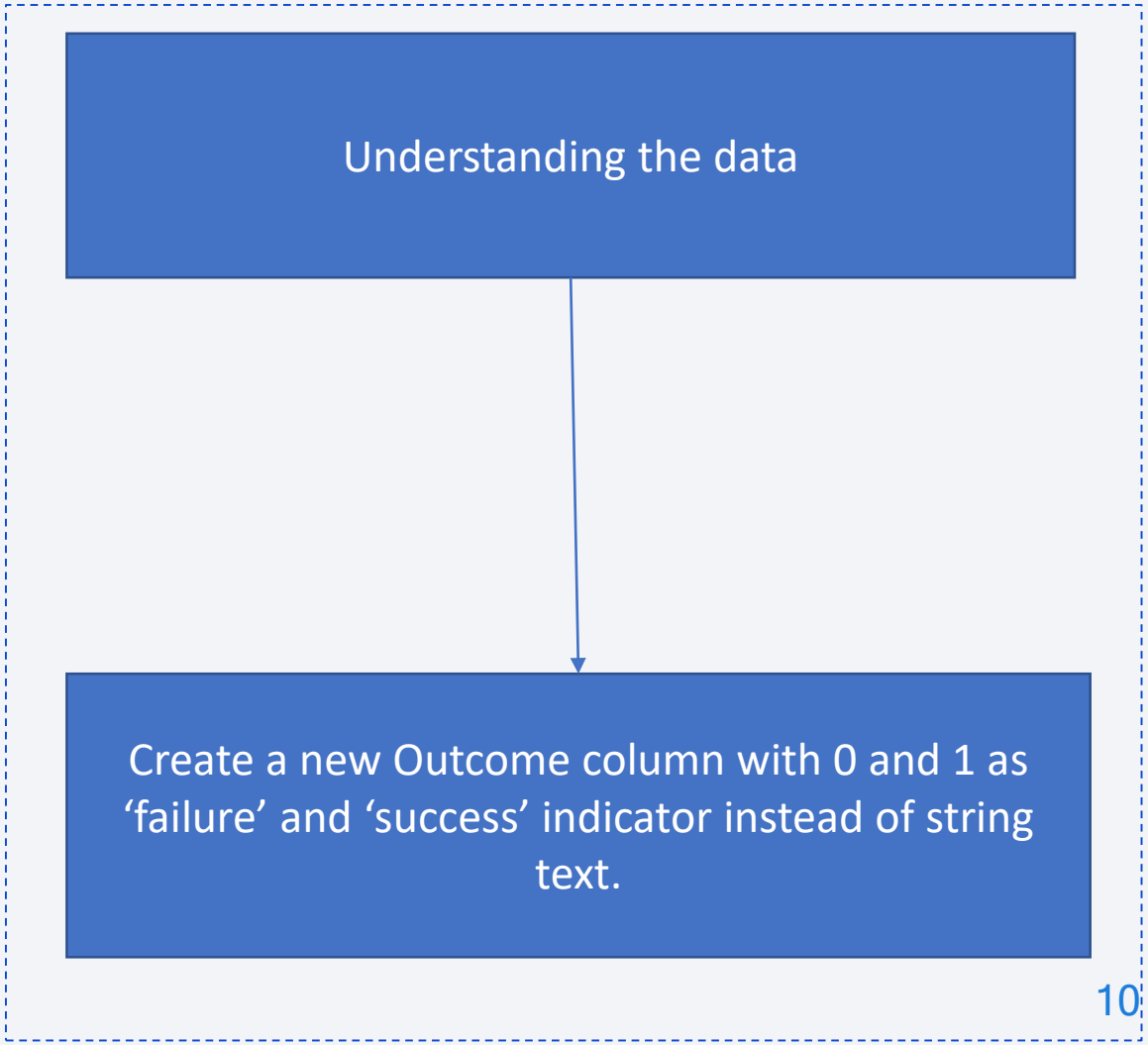


Data Wrangling

- We get understanding about the data by:

a. Loading the data into data frame.
b. Identifying and quantifying the missing data
c. Look at the data types (using .dtype method)
d. Understanding the launch sites and their launching activities (by applying value_counts() on column LaunchSite)
e. determine the number and occurrence of each orbit (by applying value_counts() on column Orbit)
f. Calculate the number and occurence of mission outcomes. (by applying value_counts() on column Outcomes)
g. Create a landing outcome label from Outcome column, we use 0 as fail and 1 as success.

- Result can be seen in: https://github.com/wsuhendra/IBM-Data-Science/blob/2c363247dd9181c97b3d262a64d947558806f414/jupyter-labs-spacex-data-collection-api.ipynb



EDA with Data Visualization

- Charts plotted:

a. Scatter plot: FlightNumber vs. PayloadMass to see the correlation between number of flights and Palyload Mass with the success of the launch.
b. Scatter plot: FlightNumber vs LaunchSite to see the correlation between number of flights and location of the launches with the success of the launch
c. Scatter plot: Payload Mass and Launch Site to see the correlation between payload mass and location of the launches with the success of the launch
d. Bar plot: success rate of each orbit type, to see correlation between orbit type and success rate
e. Scatter plot: FlightNumber and Orbit type to see the correlation between number of flights and orbit with the success of the launch
f. Scatter plot: Payload Mass and Orbit type to see the correlation between payload mass, orbit type with the success of the lauch.
g. Line chart: Lauch success over the year. To see the success rate over the years.

- Result can be seen at: https://github.com/wsuhendra/IBM-Data-Science/blob/b6b8e81af36dc6885bdf8598dcd2caa4b3d0ec71/edadataviz.ipynb

EDA with SQL

queries	purpose
select distinct Launch_Site from SPACEXTABLE	Display the names of the unique launch sites in the space mission
select * from SPACEXTABLE where Launch_Site LIKE 'CCA%' limit 5	Display 5 records where launch sites begin with the string 'CCA'
SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE Customer = 'NASA (CRS)'	Display the total payload mass carried by boosters launched by NASA (CRS)
SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE Booster_Version = 'F9 v1.1'	Display average payload mass carried by booster version F9 v1.1
SELECT MIN(Date) FROM SPACEXTABLE WHERE Landing_Outcome LIKE 'Success%';	List the date when the first succesful landing outcome in ground pad was acheived.
SELECT Booster_Version **FROM** SPACEXTABLE **WHERE** Landing_Outcome **=** 'Success (drone ship)' **AND** "PAYLOAD_MASS__KG_" **>** 4000 **AND** "PAYLOAD_MASS__KG_" **<** 6000;	List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
SELECT MISSION_OUTCOME, **COUNT**(MISSION_OUTCOME) **AS** TOTAL **FROM** SPACEXTABLE **GROUP BY** MISSION_OUTCOME;	List the total number of successful and failure mission outcomes
SELECT Booster_Version FROM SPACEXTABLE WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE);	List the names of the booster_versions which have carried the maximum payload mass
select substr(Date, 6,2) as MONTH, substr(Date,0,5) as YEAR, Landing_Outcome, Booster_Version, Launch_Site from SPACEXTABLE where substr(Date,0,5)='2015'	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.

Result can be seen at: https://github.com/wsuhendra/IBM-Data-Science/blob/b6b8e81af36dc6885bdf8598dcd2caa4b3d0ec71/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

Map objects added	purpose
Circle and marker on NASA location	To pinpoint location of NASA and help reader with a marker stating location name.
Circles and markers on all SpaceX launch sites	To pinpoint location of SpaceX lauch sites and help reader with a marker stating sites' names.
Marker clusters of success (in green) and failure (in red) to each sites according to number of launches	To help reader understand the magnitudes of failures and successes on each site.
Polylines connecting a launch site (i.e. CCAFS SLC-40) to: a. Nearest coastline b. Nearest town c. Nearest Railway d. Nearest Highway	To help reader to appreciate the distance between launch location and its surrounding important public facilities and town.

- Result is in: https://github.com/wsuhendra/IBM-Data-Science/blob/b6b8e81af36dc6885bdf8598dcd2caa4b3d0ec71/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

Plots and interaction added	purpose
Dropdown list, populated with options: All, and each site's name.	Allow the reader to pick and choose (interact) with the visualization based on launch sites.
Pie chart of success rate per site	Allow reader to see the success rate of all or each site.
Scatter plot: payload mass vs success rate	Allow reader to see the correlation between payload mass and success of each site or all sites
Payload-slider	Allow reader to interact with payload and see success rate based on different payload.

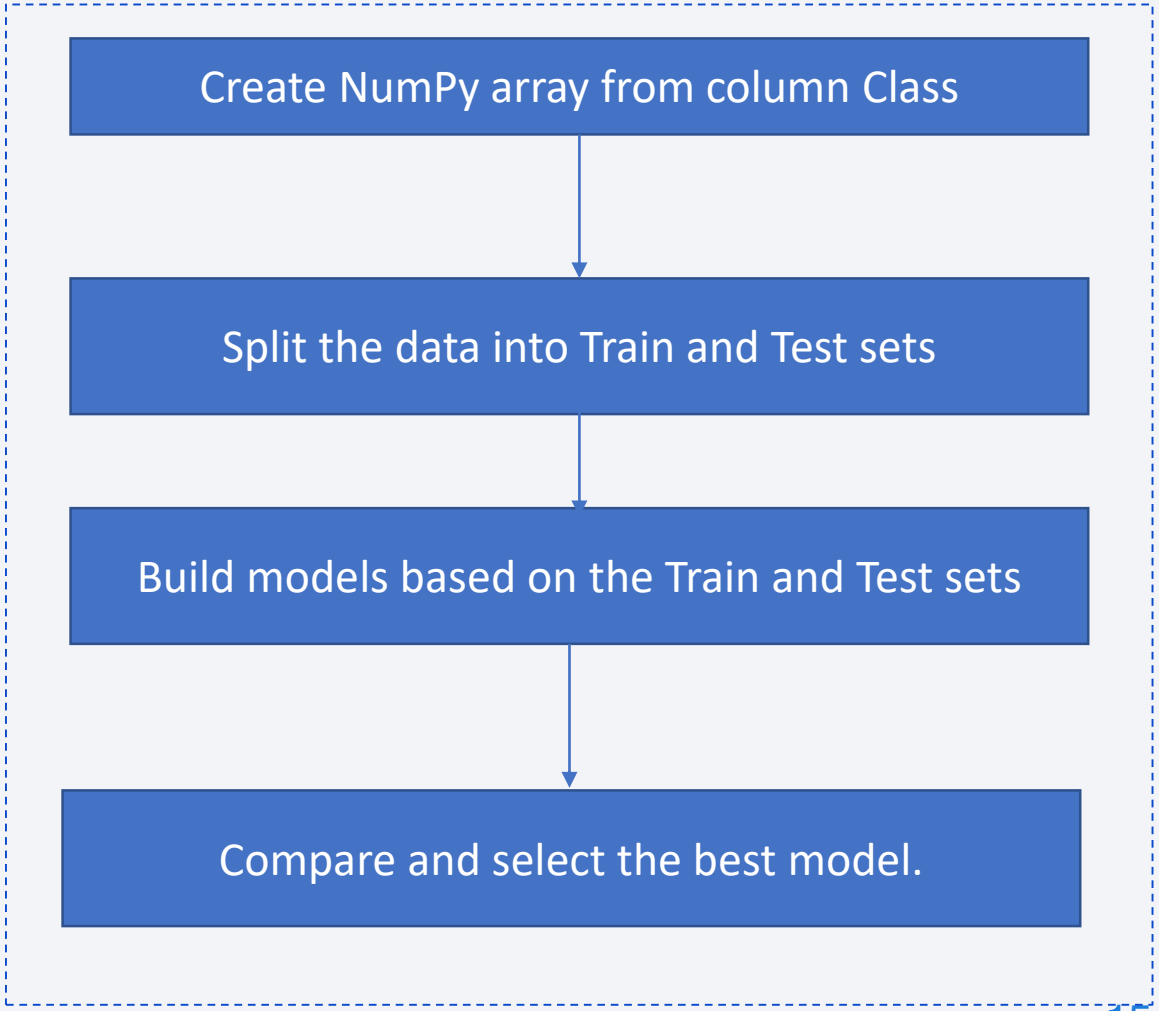
Result is in : https://github.com/wsuhendra/IBM-Data-Science/blob/b6b8e81af36dc6885bdf8598dcd2caa4b3d0ec71/spacex_dash_app.py

Predictive Analysis (Classification)

- Create a NumPy array from the column Class in data, by applying the method to_numpy() then assign it to the variable Y
- Standardize the data in X then reassign it to the variable X using the preprocessing.StandarScaler() method
- Use the function train_test_split to split the data X and Y into training and test data.
- Use the split data (train and test set) to build following models:

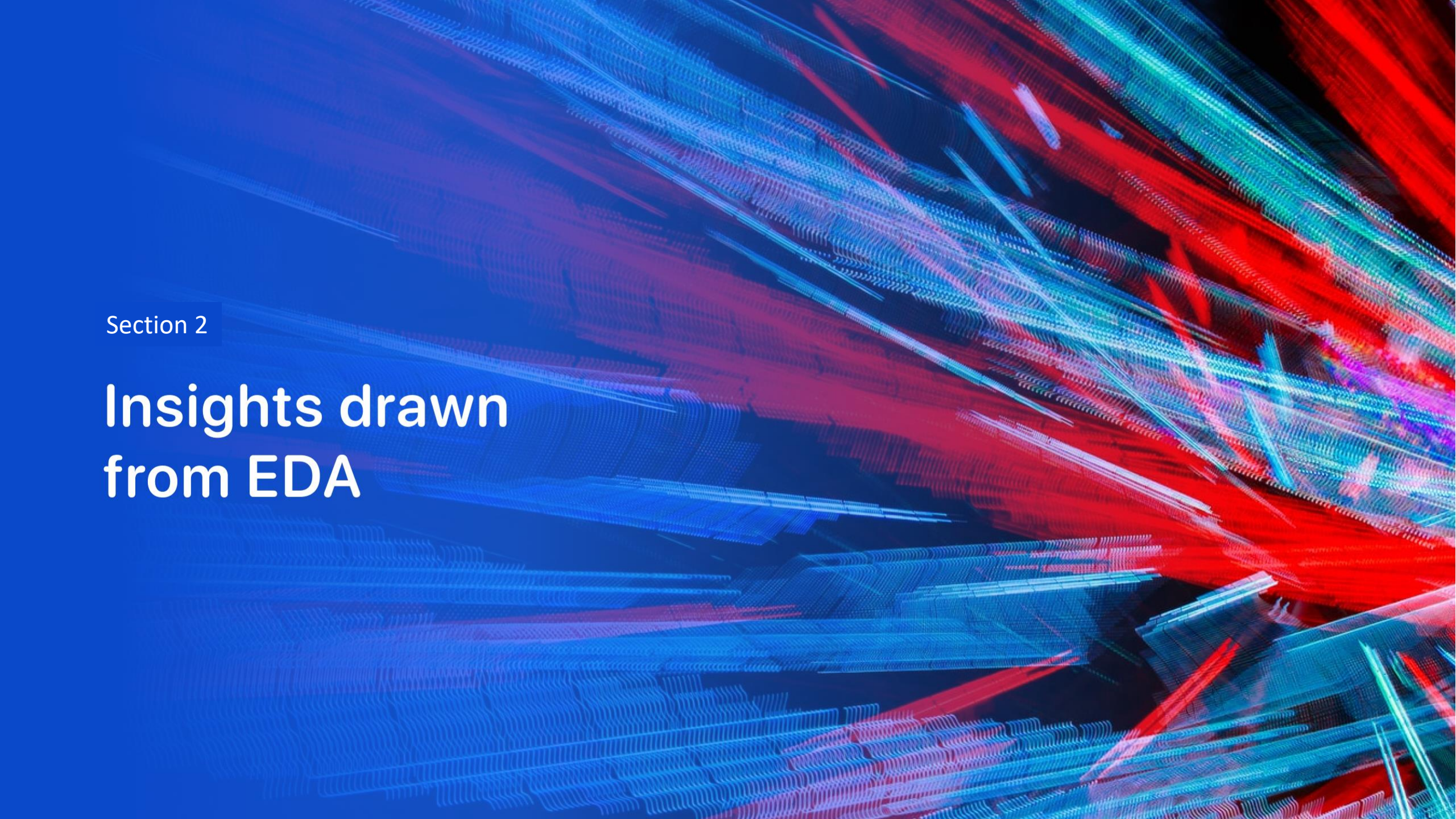
a. Logistic Regression
b. Support Vector Machine
c. Decision Tree Classifier
d. K-nearest Neighbour.

- Compare and select the best model
- Result can be seen in: https://github.com/wsuhendra/IBM-Data-Science/blob/b6b8e81af36dc6885bdf8598dcd2caa4b3d0ec71/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb



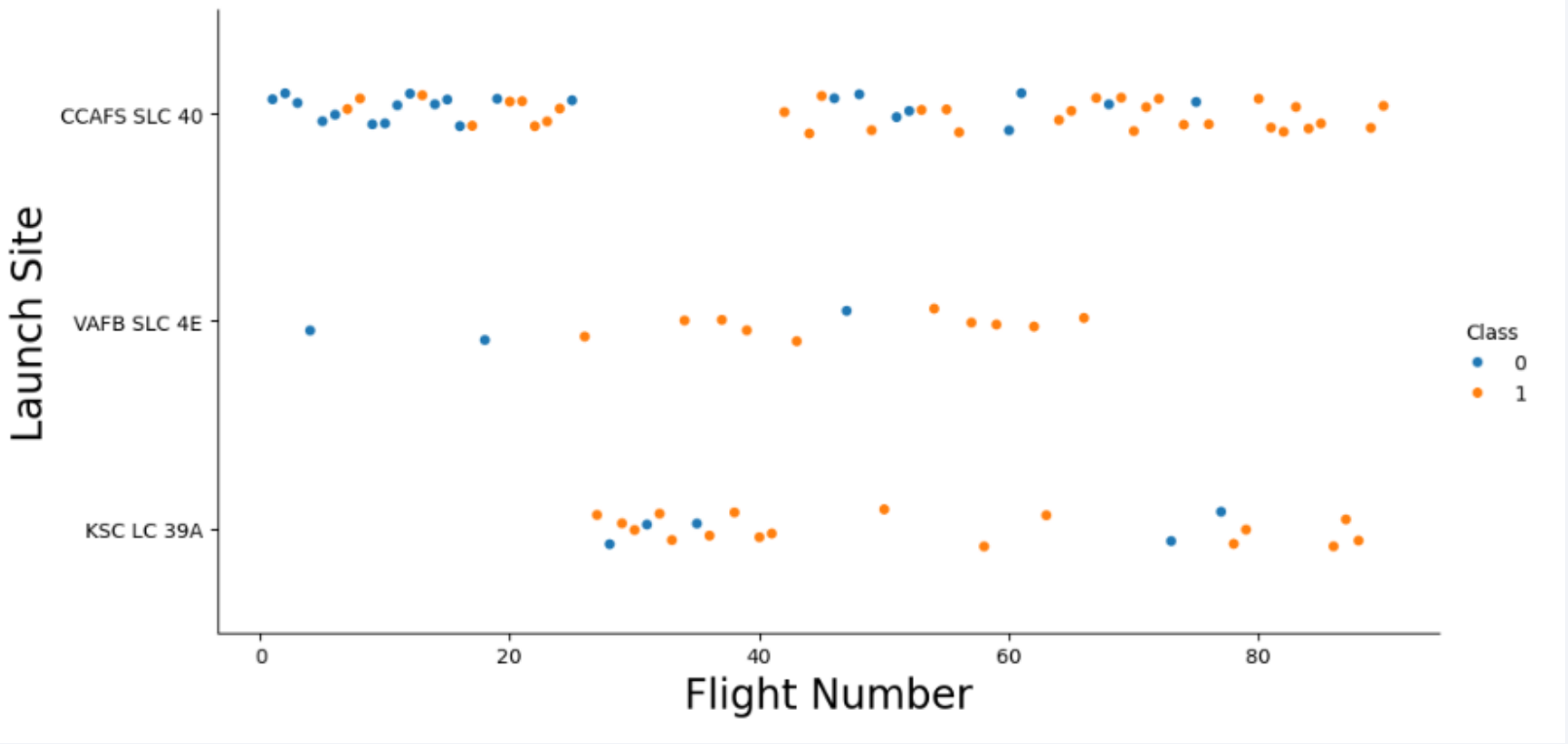
Results

- Exploratory data analysis results, please refer to the following **section 2** (slides 17-33)
- Interactive analytics demo in screenshots, please refer **to section 3 & 4** (slides 34-41)
- Predictive analysis results, please refer **to section 5** (slides 42 – 46)

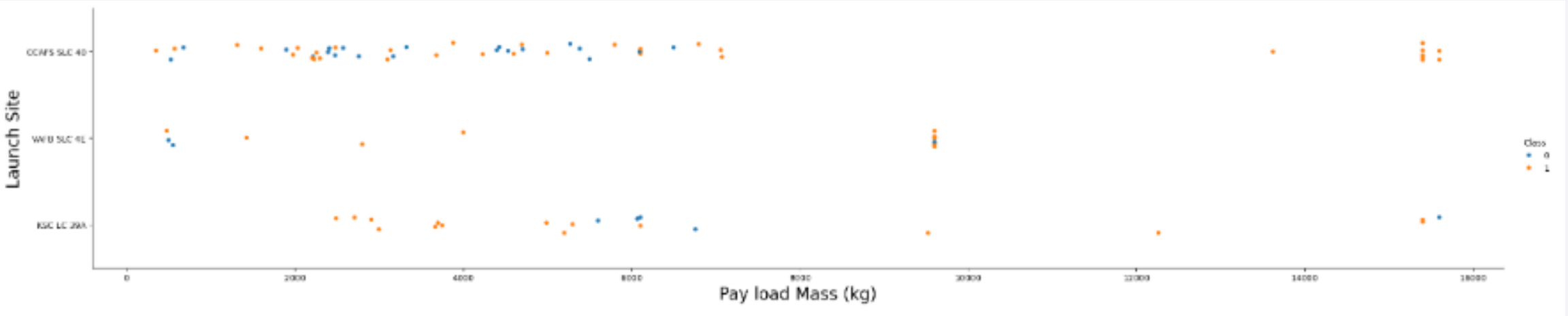

Section 2
Insights drawn from EDA

Flight Number vs. Launch Site

From the chart we can infer that the more launches done, the more success we can get.



Payload vs. Launch Site

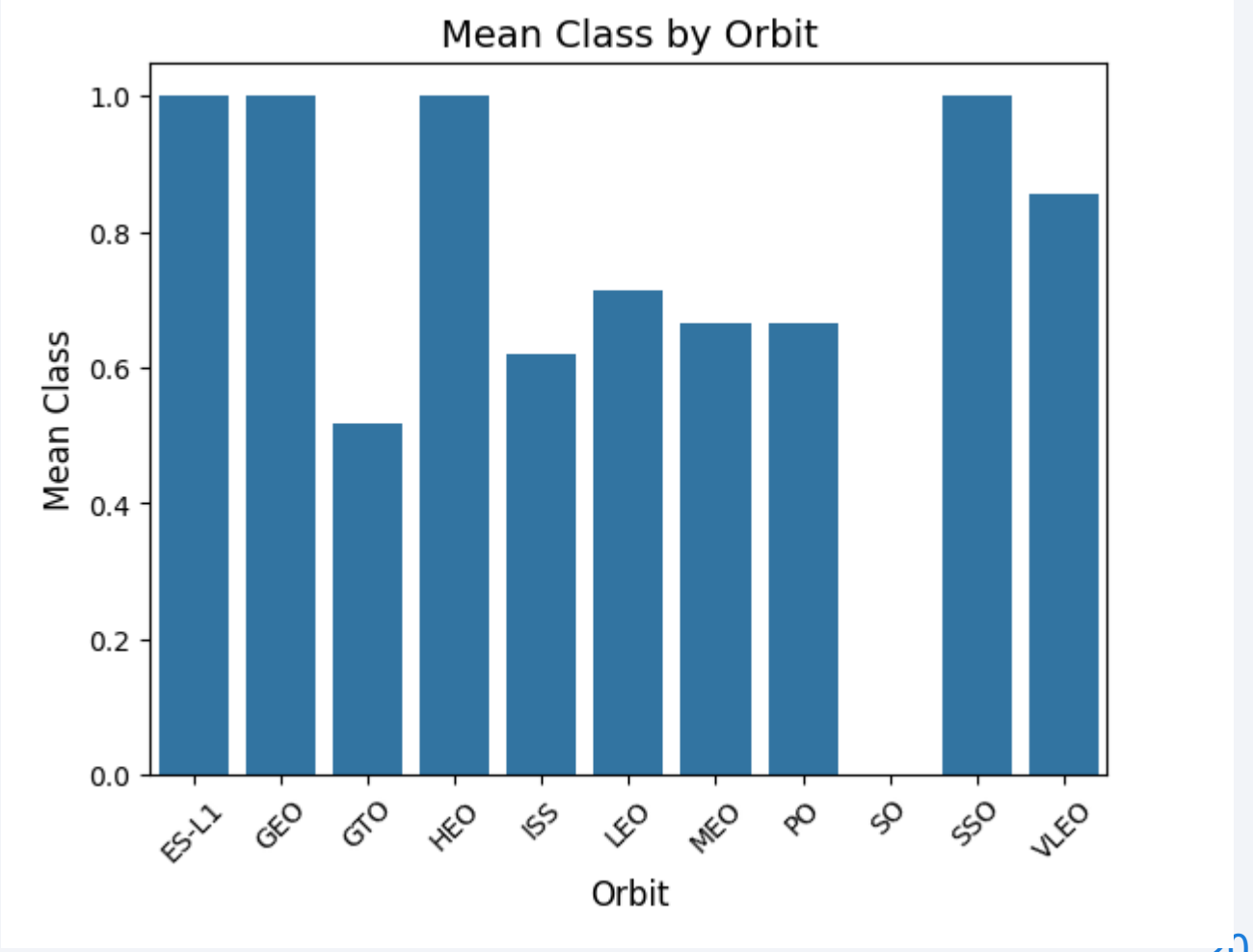


The heavier the payload, the better the results (especially 10k kg or heavier)

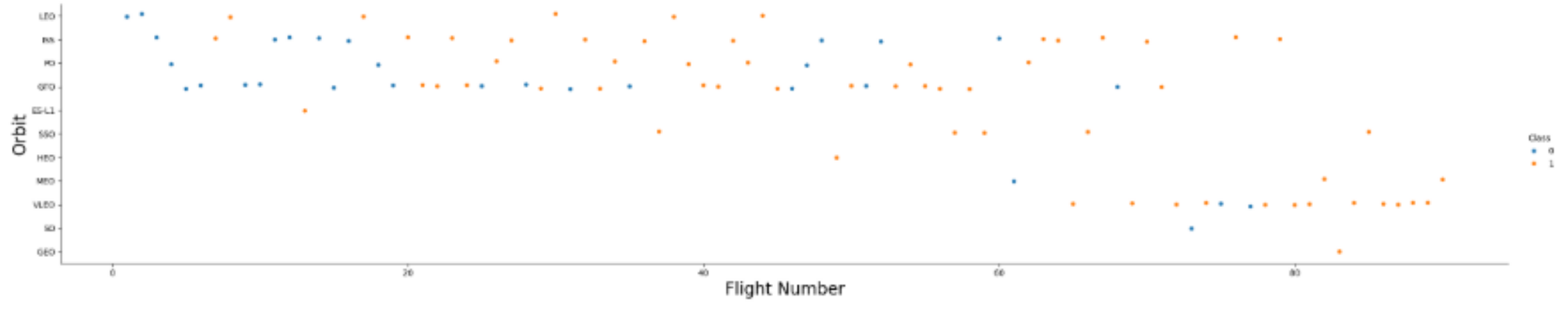
Success Rate vs. Orbit Type

ES-L1, GEO, HEO and SSO shows a 100% success rate.

The lowest is GTO with less than 60% success rate.

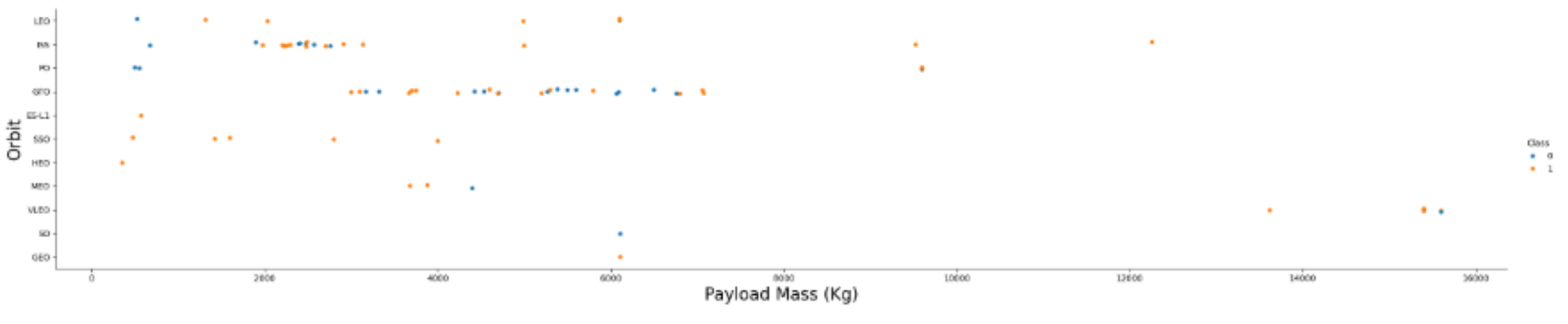


Flight Number vs. Orbit Type



It seems that for the LEO orbit, success is related to the number of flights. However, in the GTO orbit, there appears to be no relationship between flight number and success.

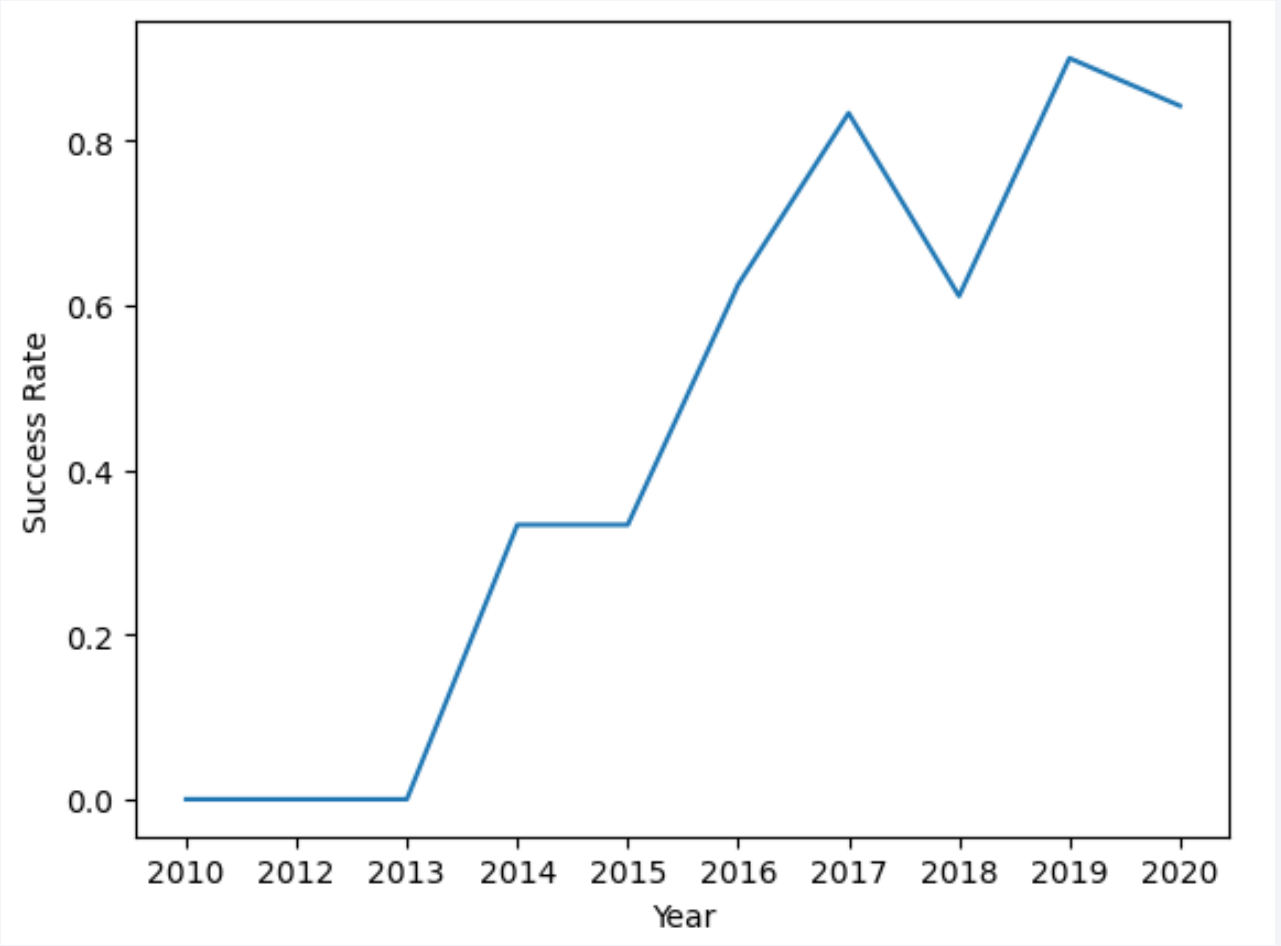
Payload vs. Orbit Type



- With heavy payloads the successful landing are more for Polar,LEO and ISS.
- However, for GTO, it's difficult to distinguish between successful and unsuccessful landings.

Launch Success Yearly Trend

- Seems like the more seasoned you are the better the chance of success
- The success rate increase surpassing 50% starting year 2016 and keep going up, except for year 2018, a slight drop in success rate.



All Launch Site Names

```
%%sql
    select distinct Launch_Site from SPACEXTABLE
```

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

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

I use the “select distinct” to select only distinct entries in the column Launch_Site from SPACEXTABLE table.

Launch Site Names Begin with 'CCA'

I use select * (all records) from the SPCEXTABLE, and apply the LIKE ‘CCA%’ (begins with CCA) and limit 5 (to limit the result to 5 only).

```
%%sql
    select * from SPACEXTABLE 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_Outc
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parach
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parach
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No atte
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No atte
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No atte

Total Payload Mass

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

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

SUM(PAYLOAD_MASS__KG_)
45596

I use select sum with the column to be summed in the bracket and give the WHERE a requirement that the Customer=“NASA(CRS)”, because I want to know only the NASA(CRS) total payload.

Average Payload Mass by F9 v1.1

```
%%sql
    SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE Booster_Version = 'F9 v1.1'
```

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

AVG(PAYLOAD_MASS__KG_)
2928.4

I use select with AVG (to calculate the average) of payload mass from column (Payload_Mass_kg_) but filter the result with Booster_Version = ‘F9 v1.1’.

First Successful Ground Landing Date

```
%%sql
SELECT MIN(Date) FROM SPACEXTABLE WHERE Landing_Outcome LIKE 'Success%';
```

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

MIN(Date)
2015-12-22

I use select min on the date. Min to yield the minimum date value, i.e. the earliest date. With filter set to Landing_Outcome column giving the record starting with the word “success”, so I use LIKE ‘Success%’.

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%%sql
SELECT Booster_Version FROM SPACEXTABLE WHERE Landing_Outcome = 'Success (drone ship)' AND "PAYLOAD_MASS__KG_" > 4000
            AND "PAYLOAD_MASS__KG_" < 6000;
```

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

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

I select the column Booster_Version with following 3 filters:

a. Landing_Outcome set to ‘Success (dron ship)’ AND
b. Payload_Mass_Kg_ > 4000 AND
c. Payload_Mass_Kg < 6000

Total Number of Successful and Failure Mission Outcomes

```
%%sql
SELECT MISSION_OUTCOME, COUNT(MISSION_OUTCOME) AS TOTAL
FROM SPACEXTABLE
GROUP BY MISSION_OUTCOME;
```

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

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

I selected Mission_outcome column and count the Mission_Outcome and assigned it to column TOTAL, all grouped by Mission_Outcome.

Boosters Carried Maximum Payload

- This is a nested query. First I need to get the Maximum Payload by querying “select Max from column Payload Mass kg
- Then fit the result as filter (using WHERE) to the query for Booster_Version.
- Seems all are B5 booster.

```
%%sql
SELECT Booster_Version FROM SPACEXTABLE
WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE);
```

```
 * 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

2015 Launch Records

```
%%sql
select substr(Date, 6,2) as MONTH, substr(Date,0,5) as YEAR, Landing_Outcome, Booster_Version, Launch_Site
from SPACEXTABLE
where substr(Date,0,5)='2015'
```

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

MONTH	YEAR	Landing_Outcome	Booster_Version	Launch_Site
01	2015	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
02	2015	Controlled (ocean)	F9 v1.1 B1013	CCAFS LC-40
03	2015	No attempt	F9 v1.1 B1014	CCAFS LC-40
04	2015	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40
04	2015	No attempt	F9 v1.1 B1016	CCAFS LC-40
06	2015	Precluded (drone ship)	F9 v1.1 B1018	CCAFS LC-40
12	2015	Success (ground pad)	F9 FT B1019	CCAFS LC-40

- First I need to substract the month number from the Date column text, by using substr(Date,6,2) starting on 6th digit pick 2 digits and assigned it to column MONTH, as well as substracting the year from the Date string using substr(Date,0,5) and assigned it to column YEAR, select also other related columns.
- Then filter the records using 2015.

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

- Select the Landing_Outcome column and count its content.
- Filter the result with between 2010-06-04 and 2017-03-20
- Finally group them according the the Landing_Outcome.

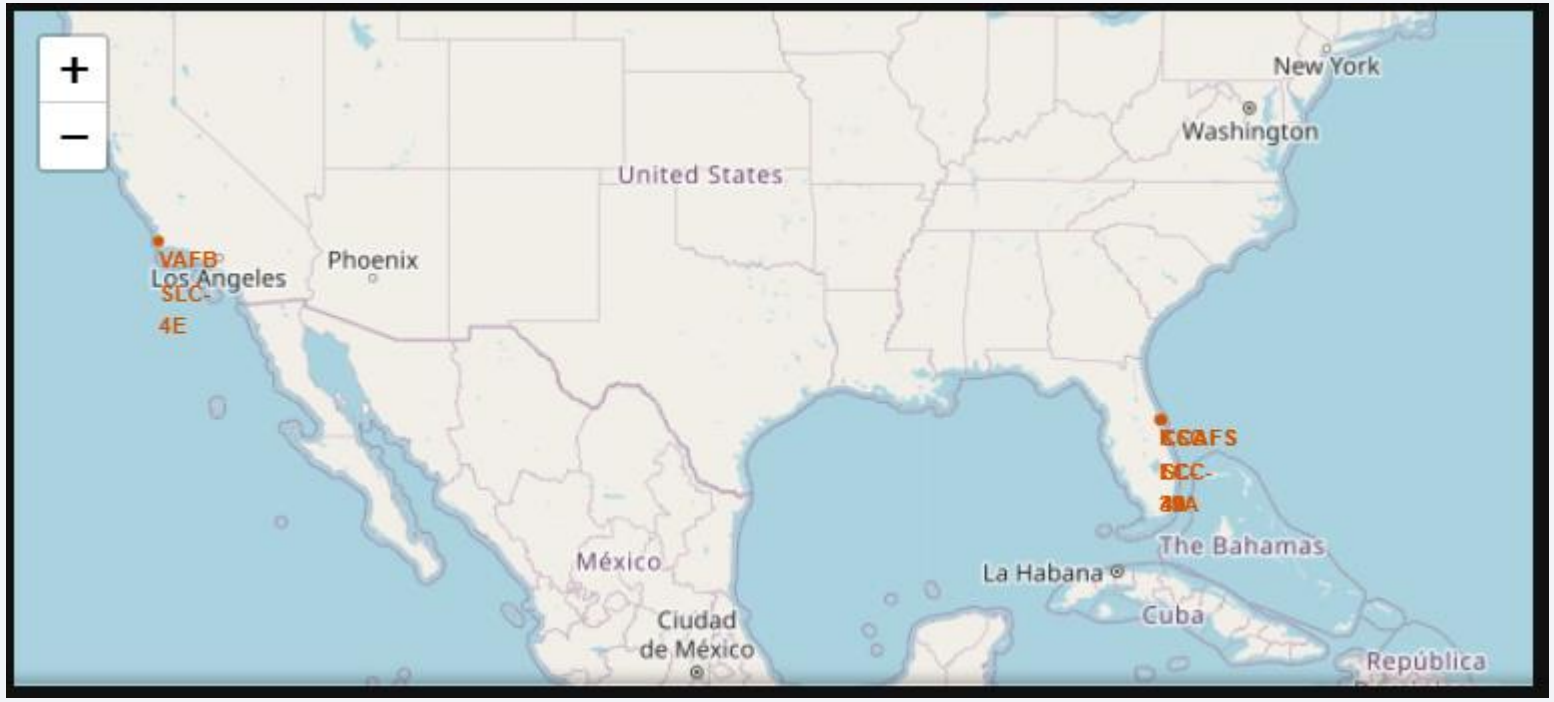
```
%%sql
select Landing_Outcome, count(Landing_Outcome)
from SPACEXTABLE
where Date >= ' 2010-06-04' and Date <='2017-03-20'
Group by Landing_Outcome
```

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

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


Section 3
Launch Sites
Proximities Analysis

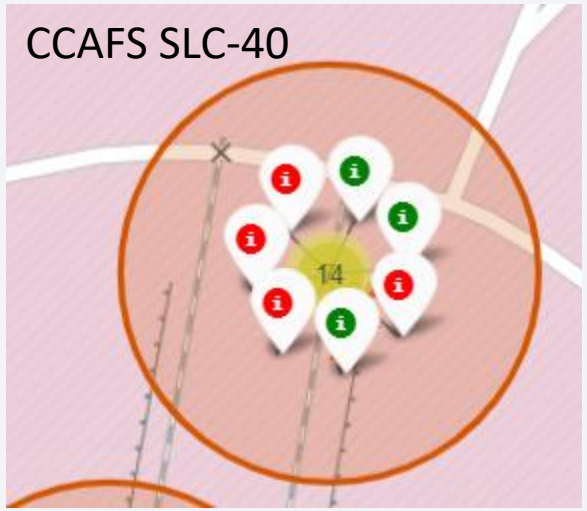
Map of Launch Sites

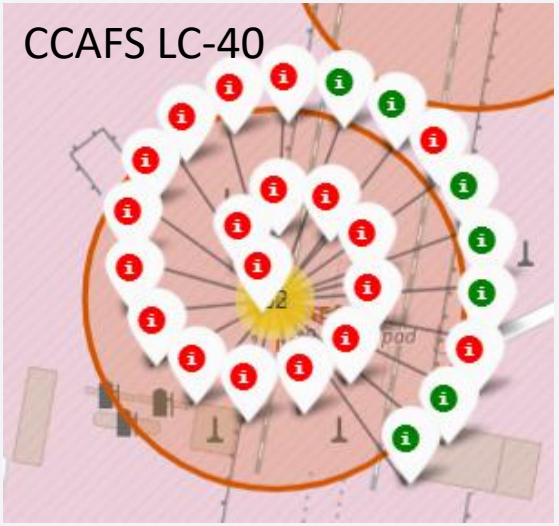


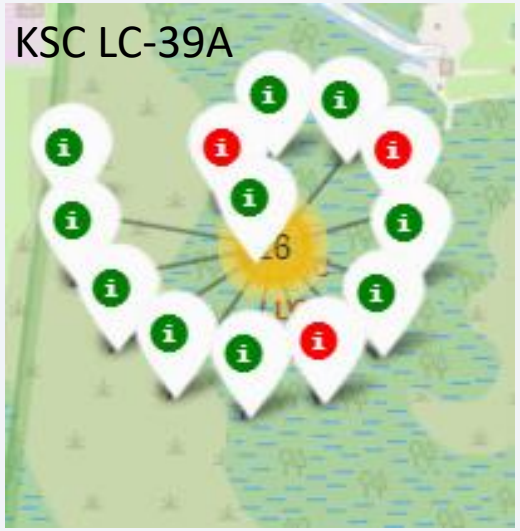
- SpaceX has 4 lauch sites, 3 sites are clustered around the Florida peninsula (i.e. CCAFS-SLC 40, CCAFS-LC-40 and KSC LC-40) and one is far away in California (i.e. VAFB-SLC-4E)

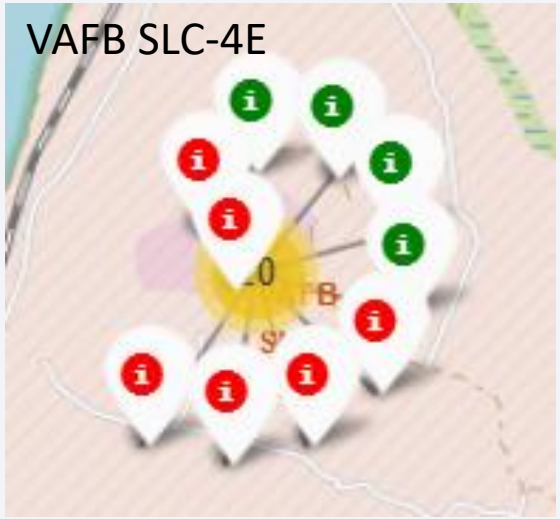
Map of Success vs Failures of Each Lauch Site

- CCAFS SLC-40 has the least number of launches
- CCAFS LC-40 has most of the launces, but also highest rate of failure as the red marks are more in this cluster.
- KSC LC-39A seems to have the highest rate of success with only 3 failures out of 13 launches.
- VAFB SLC-4E has 60% of failure with 10 launches.





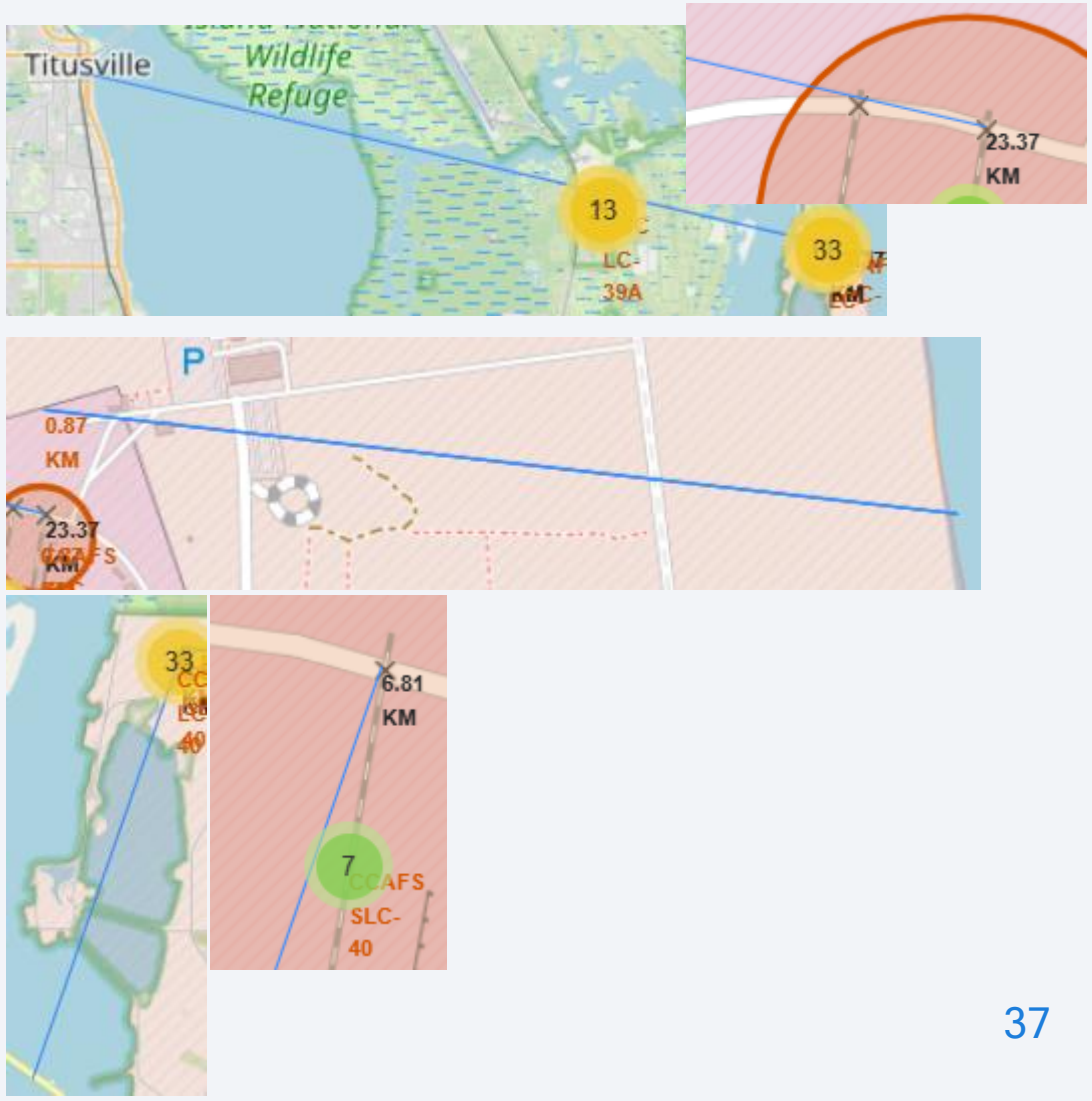


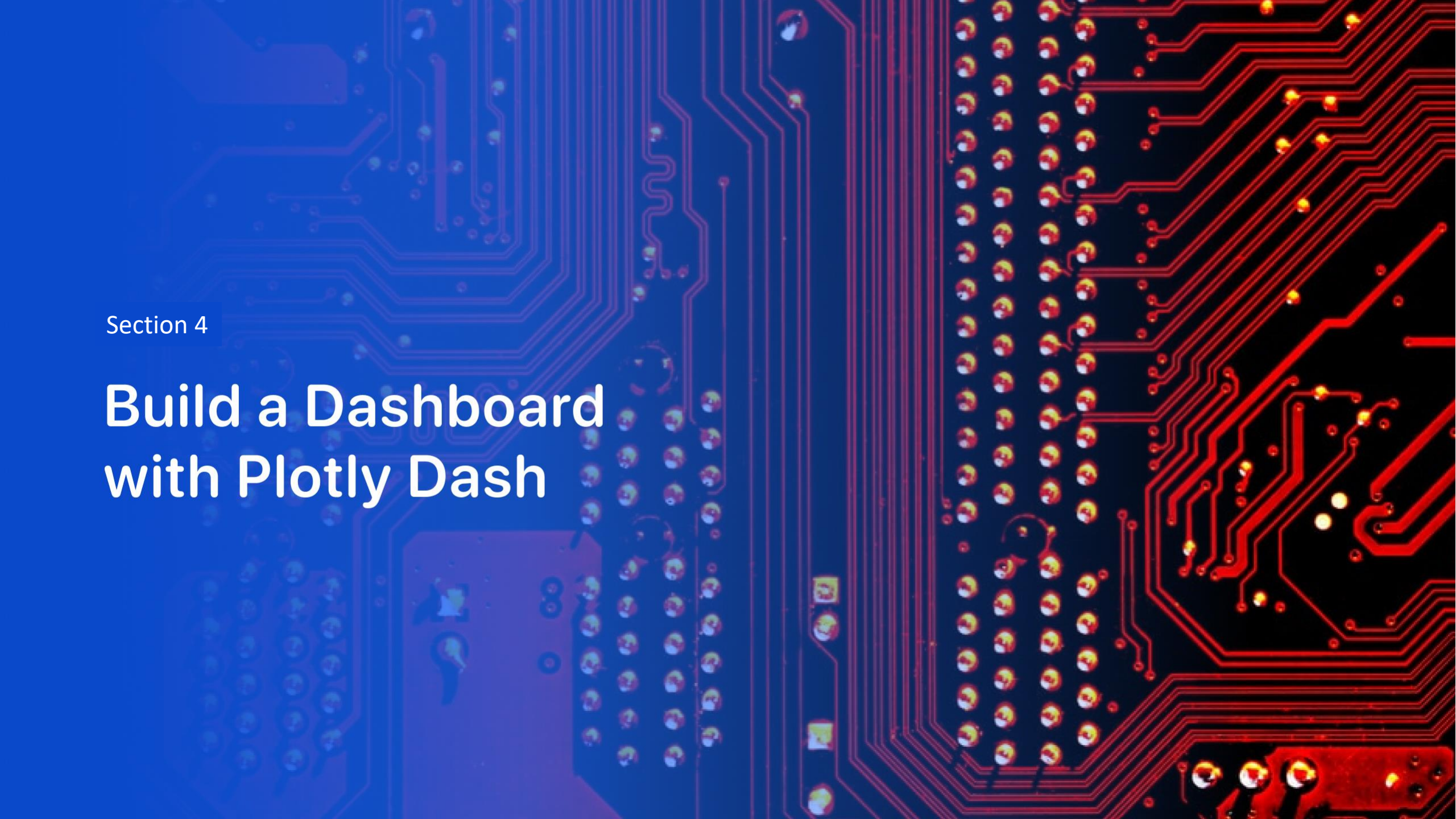


Launch Site proximities in Km.

- Nearest town (Titusville is 23.37 km away)

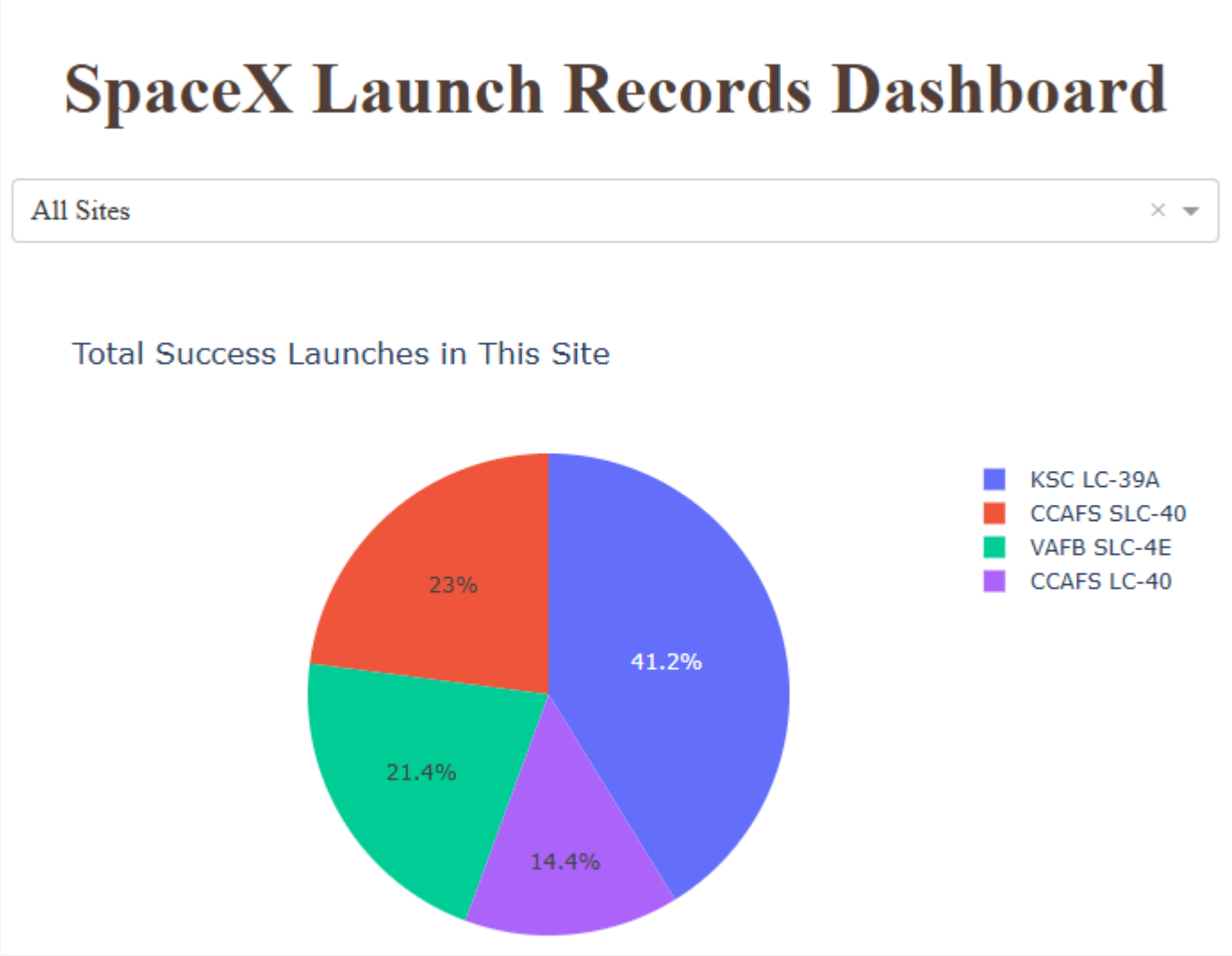
- Nearest coast line is 0.87 km away.
- Nearest highway is 6.8 km away.
- It appears that the lauch site is quite a safe distance away from public amenities, except for the coast line.




Section 4
Build a Dashboard with Plotly Dash

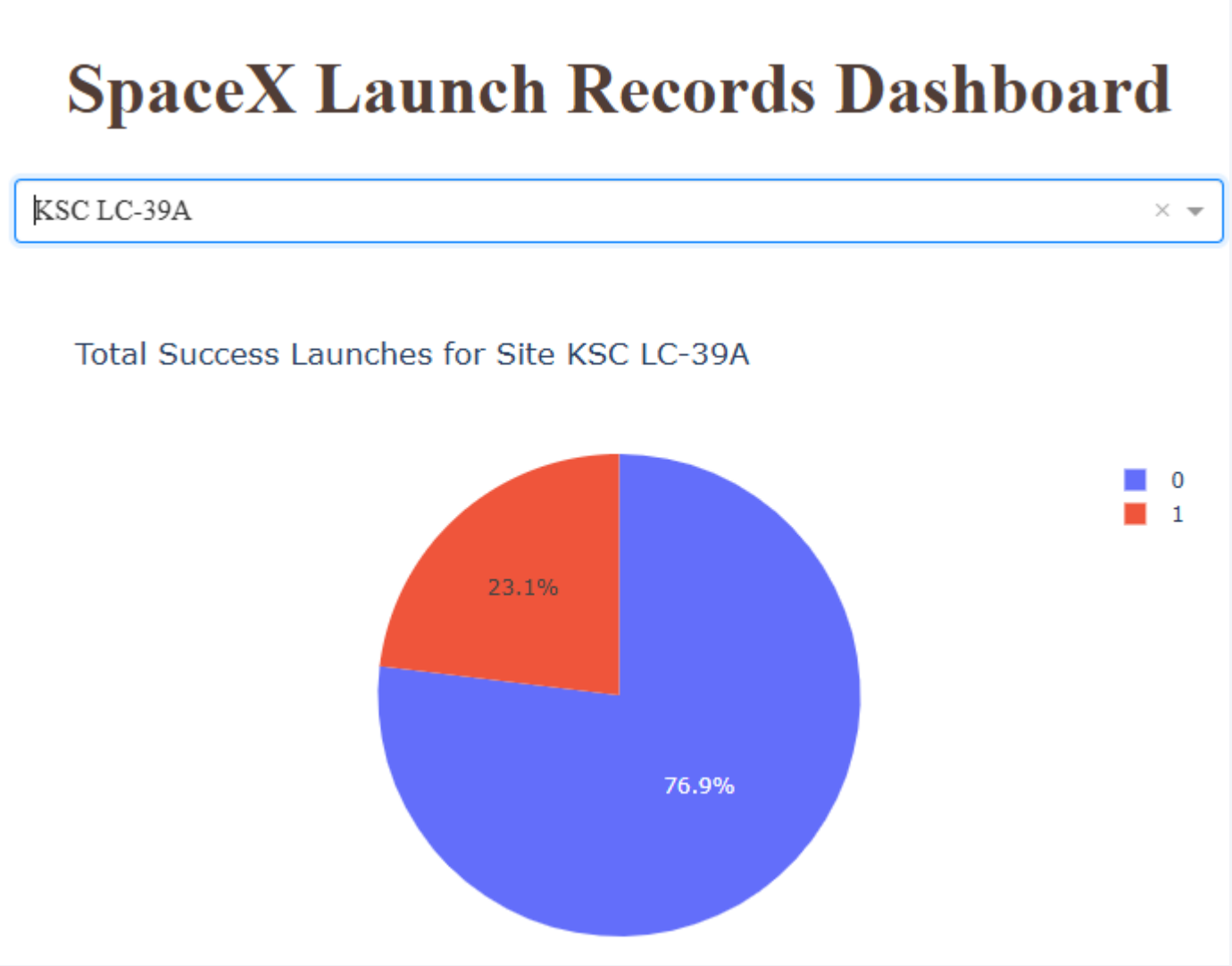
Dashboard: Success Rate of All Sites

- KSC LC-39A has the highest success rate of 41.2% of all launces across SpaceX.
- Lowest success is with CCAFS LC-40 with only 14.4% share.

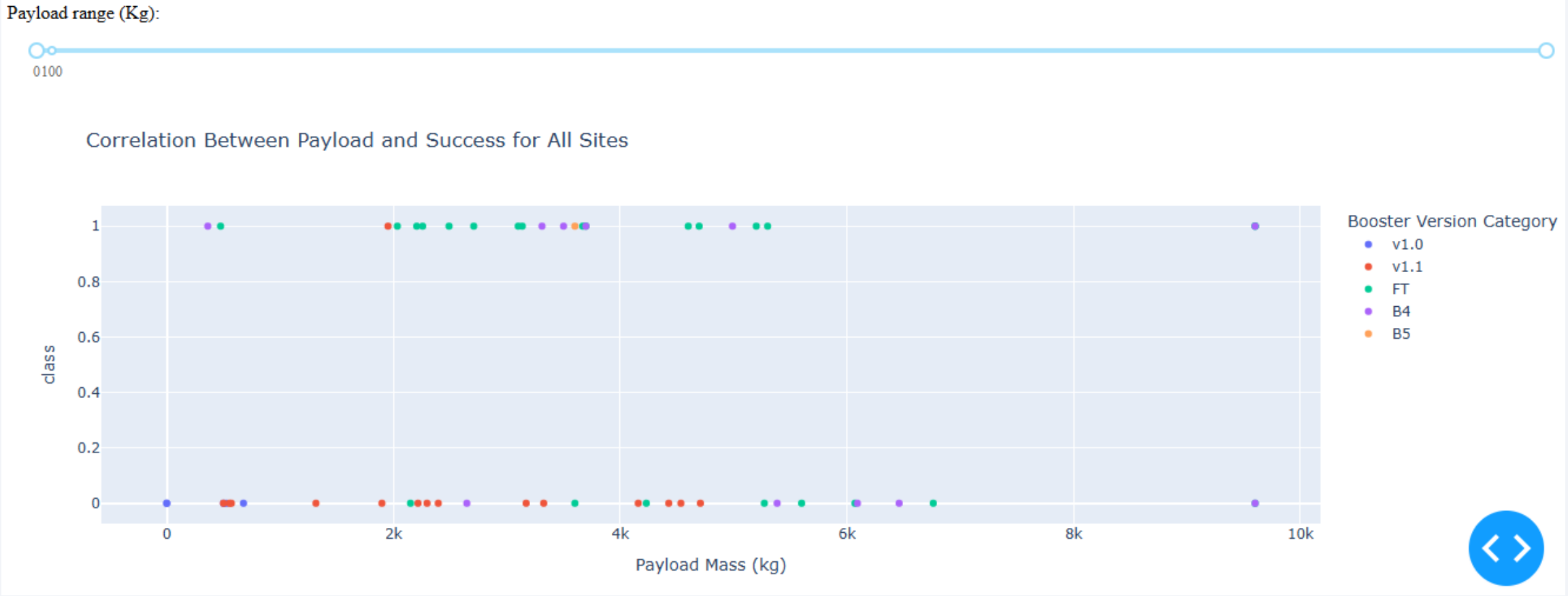


Dashboard: Site with Highest Success Rate

- KSC LC-39A has a 76% of success rate in its launching attempts.



Dashboard Payload and Booster Version



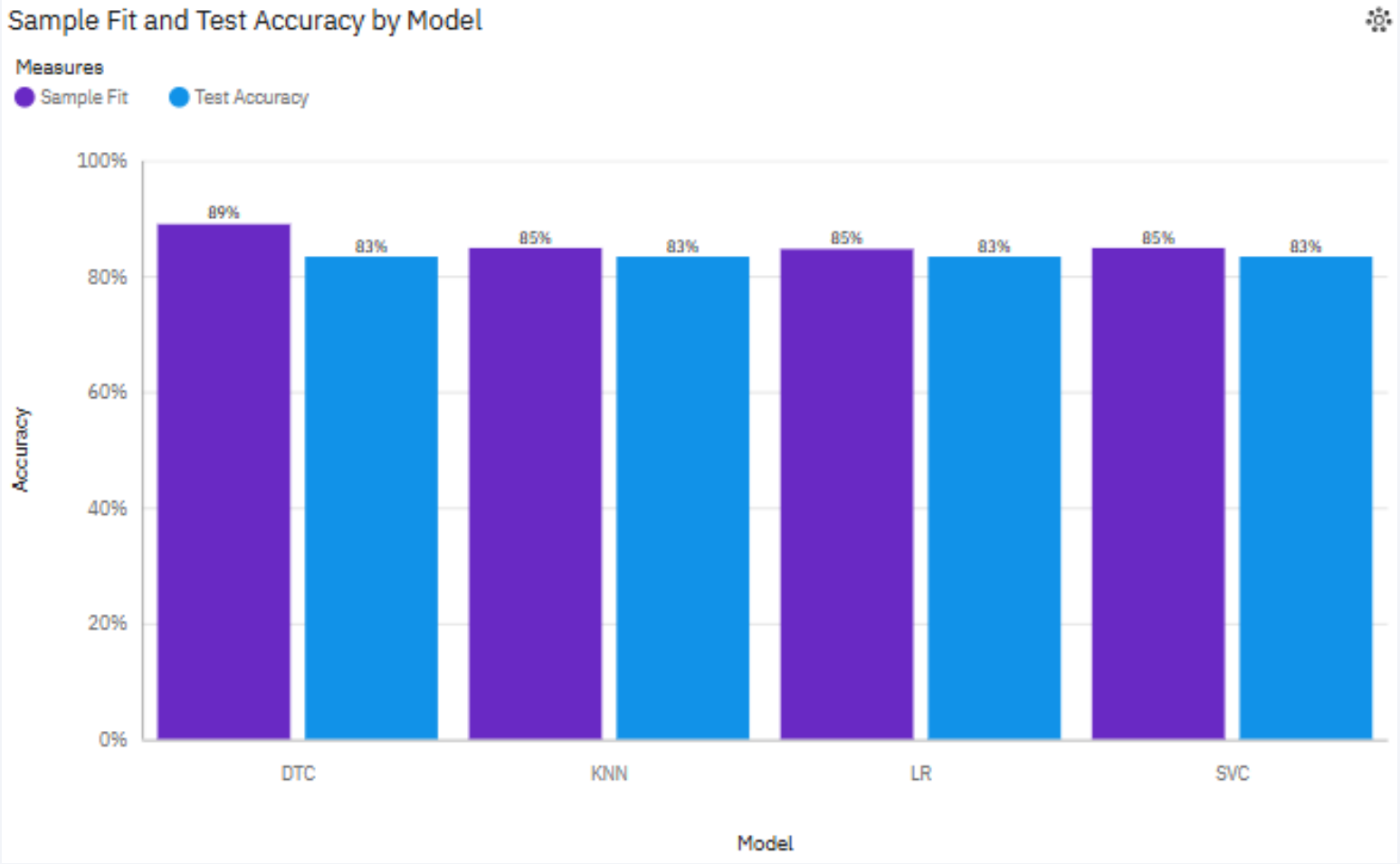
It seems the booster version FT and V1.0 dominate the contribution of success, with B4 trailing at about 50:50 chance.

Section 5

Predictive Analysis (Classification)

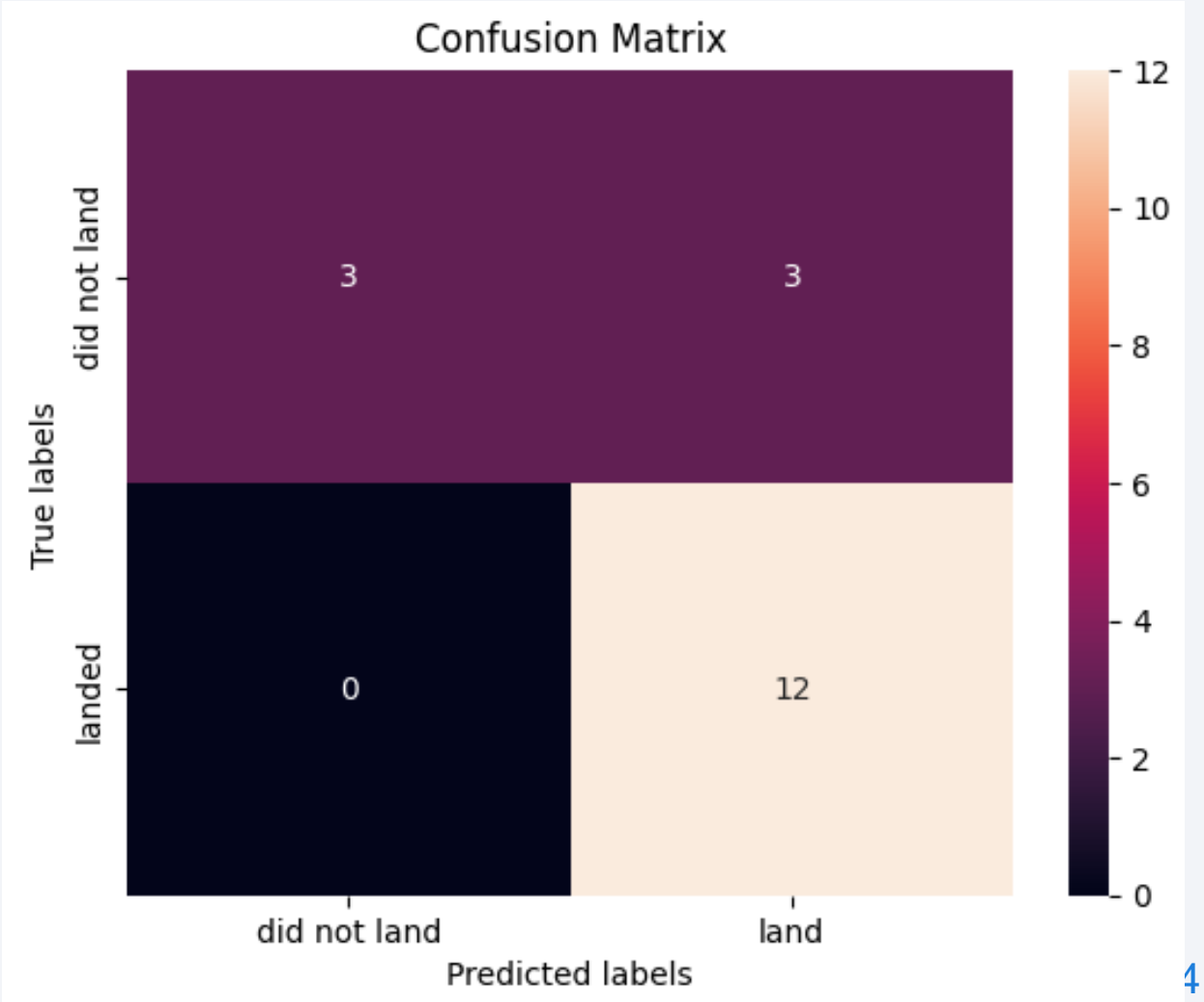
Classification Accuracy

- All models yield the same test accuracy results, i.e. 83%.
- However, DTC (Decission Tree Classification) fits better to the data.



Confusion Matrix

- All models result in the same confusion matrix.
- All can predict quite well with true positives (TP) of 12 and only 3 false positives (FP).
- The problem is with the false the False positives (FP).
- Precision = TP / (TP +FP) = 80%
- Recall = 12/12 = 100%
- F1-score = = 2 * (Precision * Recall) / (Precision +Recall) = 89%
- Accuracy = (TP + TN) / (TP + TN + FP + FN) = 83%



Conclusions

- We can use either one of 4 models built to predict the outcome of launching with expected 83% accuracy.
- Best performing booster version are FT and V1.0 Point 3.
- Best performing launch site is LC-39A with 76% success rate. From the map we can see that it is not the nearest to the coast line but still quite near.
- ES-L1, GEO, HEO and SSO orbits show a 100% success rate.
- The heavier the payloads the better the chance for success, especially 10k kg or higher.
- Experience matters, as the more launches performed the more success rate SpaceX gets.

Thank you!