IBM Developer
SKILLS NETWORK

# Winning Space Race with Data Science

Stefano Fagioli
30/11/2021

# Outline

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

# Executive Summary

- Companies are making space travel **much more affordable** than in the past.
- **SpaceX** is one of the **leading companies** in the industry, due to the capacity of **cutting costs** due to the **reutilization of the first stage**.
- With the Falcon 9 Rocket, SpaceX offers launches at **62 million USD**.
- In this presentation is provided some **exploratory data analysis** to better understand, via the SpaceX historical data, their launch history.
- A **predictive model** is built to predict **if the first stage will land**, and to determine the **cost of the launch.**
- Exploratory data analysis with **visualization**, **SQL**, by providing **Folium maps** to better understand the geographical configuration of the launch sites, and an **interactive Dash dashboard.**
- From the application of the **GridSearch pipeline**, all the models perform equally, with an **accuracy of 83.3%.** Confusion Matrixes are also provided.

# Introduction

- As the **commercial age of space travel** unfolds before us, the company **Space Y** wants to propose itself as a competitor to greater players like SpaceX.
- The stronger point of the SpaceX business plan is to **reutilize the first and most expensive stage of the rocket to contain costs to 62 million dollars per launch.** Therefore, a deeper understanding of the mechanics of the launches is crucial to the developing of a solid and competitive business plan.
- The goal of this analysis is to have a **deep understanding of all the variables** that determine if the first reusable stage of the SpaceX rocket makes a successful landing or not and to build on this knowledge to create first-stage boosters that can always successfully land, minimizing costs.

Section 1

# Methodology

# Methodology

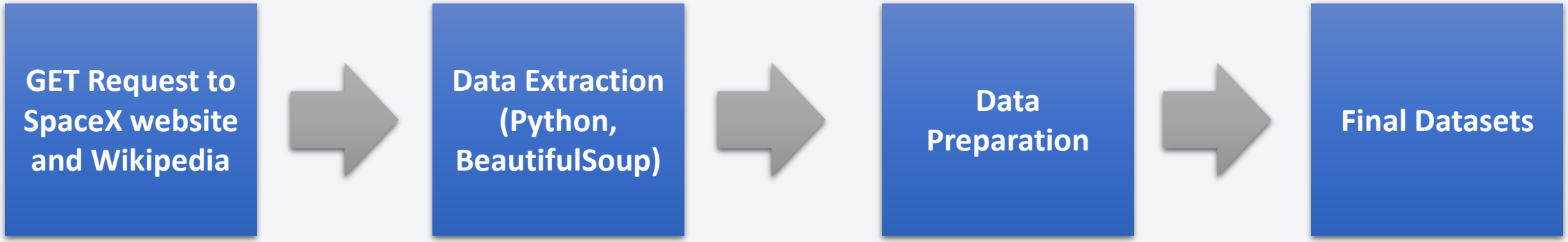
# Executive Summary

- Data collection methodology:
  - The data was collected from two sources: the SpaceX website, and Wikipedia.
- Perform data wrangling
  - For the data frame obtained from the SpaceX website, the missing values in the Payload column was replaced with the mean. For the data obtained from Wikipedia, the correct table was selected, the column headers were used to create a dictionary, which was populated with the values of the selected table. Then the data frame was created.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - The data was standardized, split into train and test sets and then a GridSearchCV was made with Logistic Regression, Support Vector Machine, Decision Tree Classifier and K Nearest Neighbor. Then the score for each method was calculated, and the Decision Matrixes were plotted.

# Data Collection

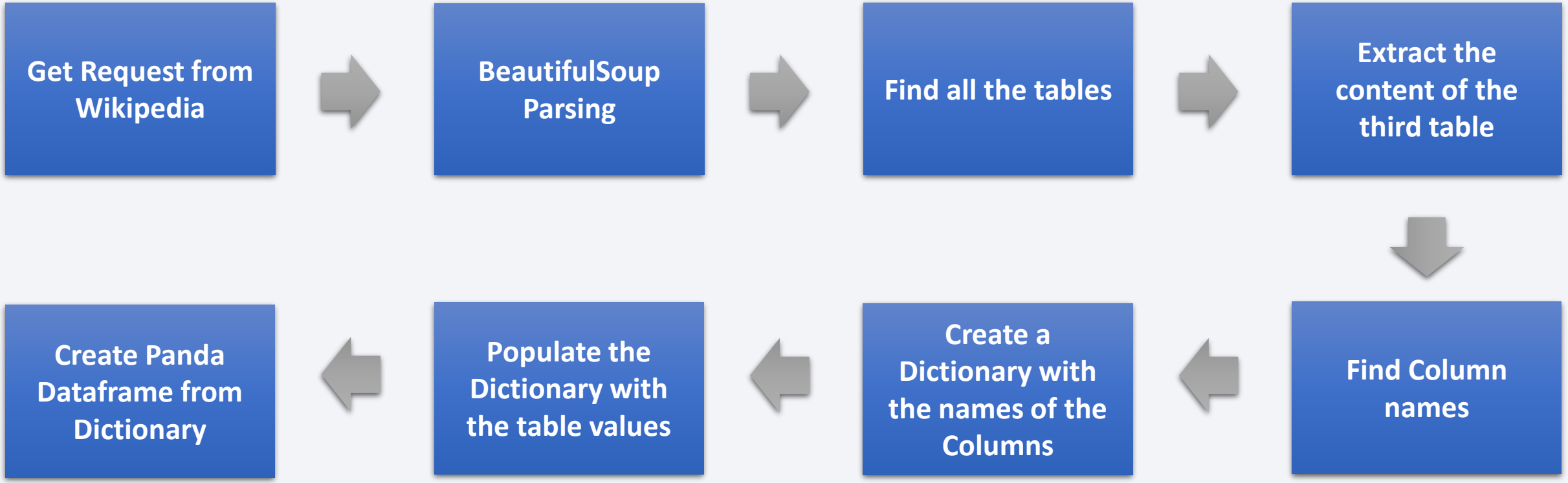
- Data were obtained from two sources: from the SpaceX website, and from Wikipedia.
- Four functions were used to extract the values needed for the analysis: booster name, launch site, payload data and core data.
- After getting the whole SpaceX database, a subset is created with the needed variables. The entries with more than 1 booster and 1 payload are removed. The date format is changed, and the date is limited to 2020/11/13.
- Lists with the variable names are created, to be populated and then combined in a dictionary and then into a data frame.
- A subset is created with only the Falcon 9 launches.

# Data Collection – SpaceX API




GitHub: Data Collection API Lab

# Data Collection - Scraping




GitHub: [Data Collection with Web Scraping]

# Data Wrangling

- The Data Wrangling process started with the identification of the percentage of the missing attributes in each value, and of the types of each value. The missing values in the Payload column were replaced with the mean.
- Then the number of launches for each site was calculated, the number of launches for each Orbit type, and the number of occurrences for each launch outcome. The outcomes were divided in “Successful” and “Unsuccessful” outcomes and a variable was given to each.
- The outcomes were divided in “Successful” and “Unsuccessful” outcomes and a variable “Class” was created, with a value for each outcome.
- Then, the column “Class” with all the landing outcomes was added to the data frame.

GitHub: EDA

# EDA with Data Visualization

- A bar graph was used to show the relationship between success rate and orbit type.
- A line plot graph was used to show the increasing success percentage during the years.
- Scatter plots were used to show the relationship between:
  payload mass, flight number and success rate;
  launch site, flight number and success rate;
  launch site, payload mass and success rate;
  orbit, flight number and success rate;
  payload mass, orbit type and success rate;

GitHub: EDA with Data Visualization

# EDA with SQL

## The following queries were performed:

- Select Launch Site
- Display first 5 records of Launch Site that begins with the string “CCA”
- Display total payload mass carried by boosters launched by NASA
- Display average payload mass carried by booster version F9 v1.1
- Displayed the date when the first successful landing outcome in ground pad was achieved
- Displayed names of boosters that have success in drone ship and payload between 4000 and 6000
- Displayed total number of successful and failed missions
- Displayed the list of the boosters that have carried the maximum payload mass
- Displayed the failed landing outcomes in drone ship, their booster versions and launch site names for year 2015
- Ranked the count of landing outcomes between 2010-06-04 and 2017-03-20 in descending order

GitHub: EDA with SQL

# Build an Interactive Map with Folium

- Markers and circles on the coordinates of all the launch sites were created.
  Then markers were added to each launch site in color green or red.
  Then the distance between launch sites and its proximities (road, cities, railroads, coast) was drawn.
- The markers were added to give a geographical representation of the location of each launch site, with the exact number of successful and unsuccessful launches for each site. The distances to the proximities were drawn to have a deeper understanding of the surroundings as a key factor in the success of the launch.

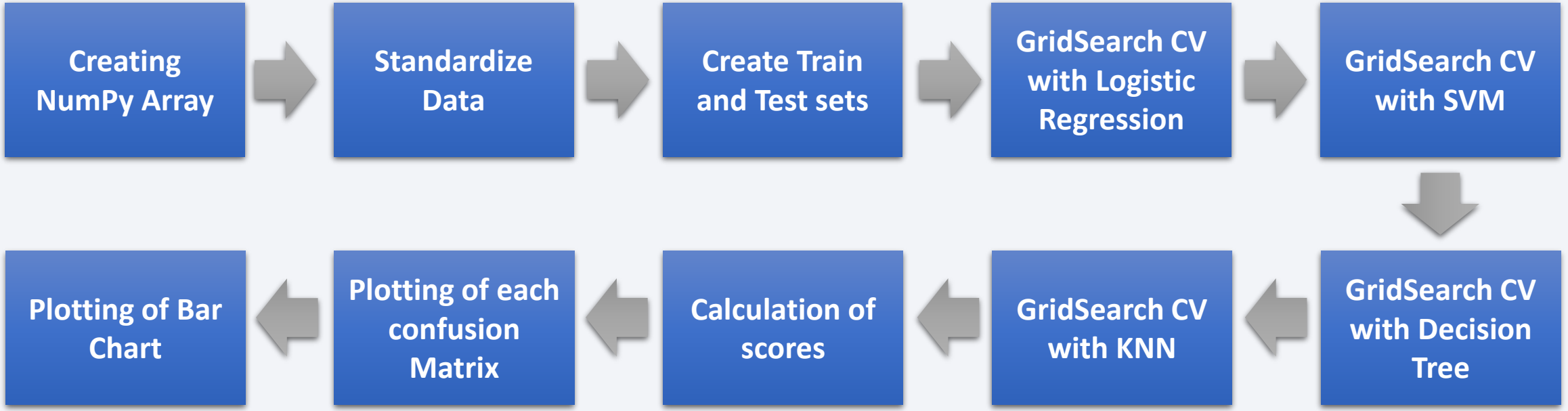
GitHub: Interactive Visual Analytics with Folium

# Build a Dashboard with Plotly Dash

- A dropdown menu with all the launch sites, and two graphics were added, a pie chart that shows the successful launches for all the launch sites combined or the successful and unsuccessful launches for each selected launch site and a scatter plot with all the launch outcomes for all the launch sites or for the single launch. A slider was added too, to filter the launches to display by the minimum and maximum of carried payload mass.
- All the plots and interactions were added to have a visual representation of the geography of the launch sites, to have a deeper understanding of the relationship between the launch sites, each launch payload and even the presence of cities, railroads and roads to the launch success rate.

GitHub: SpaceX Dash App Python Code

# Predictive Analysis (Classification)




- The "Class" column was used to create a NumPy array, then assigned to Y.
- The data was then standardized through a transform, and assigned to X.
- After splitting the data in train and test sets, we created four different GridSearchCV object for Logistic Regression, Support Vector Machine, Decision Tree Classifier and K Nearest Neighbors. The score for each of the methods was calculated, and the Confusion Matrix plotted. Then, a bar graph was created to show the best performing model



GitHub: Machine Learning Prediction

# Results

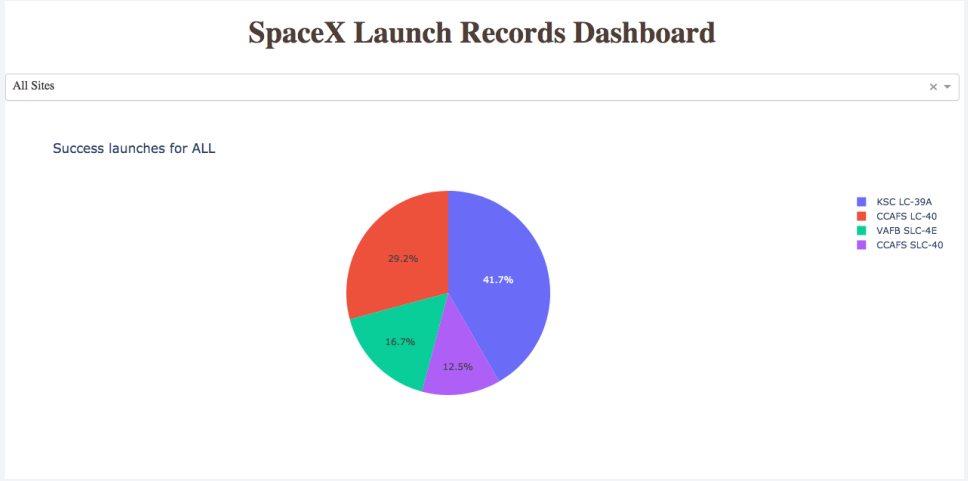
## Exploratory Data Analysis

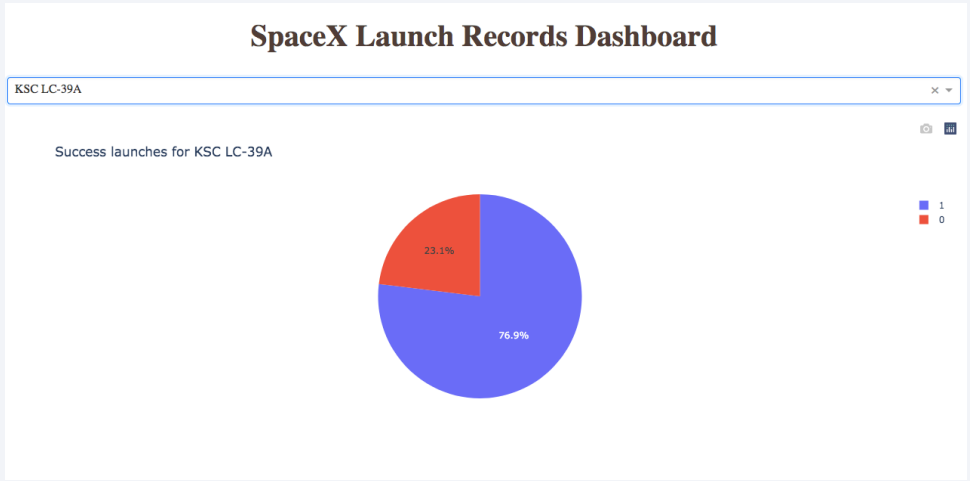
- For **around 60% of the launches**, the **CCAFS SLC 40** (Cape Canaveral Space Launch Complex 40) site was used. The launches were mainly directed to the **GTO, ISS and VLEO orbits**, and most of the successful landings were on **drone ships (41)** or ground pads.
- The **CCAFS SLC 40 has a success rate of 60%** while **KSC LC-39A and VAFB SLC 4E has a success rate of 77%.** The success rate of the launches is **steadily increasing since 2013**. The most successful orbits are **ES-L1, GEO, HEO and SSO**.
- Regarding the delivered payloads, there are **no payloads over 10000 delivered from VAFB SCL 4E**. For the heavier payloads, the more successful orbits are Polar, LEO and ISS.
- The SQL analysis shows that the total payload mass delivered for NASA is short of 100k and that SpaceX needed approximately **2 years to make a successful landing on ground pad.**

# Results

## Interactive Analytics

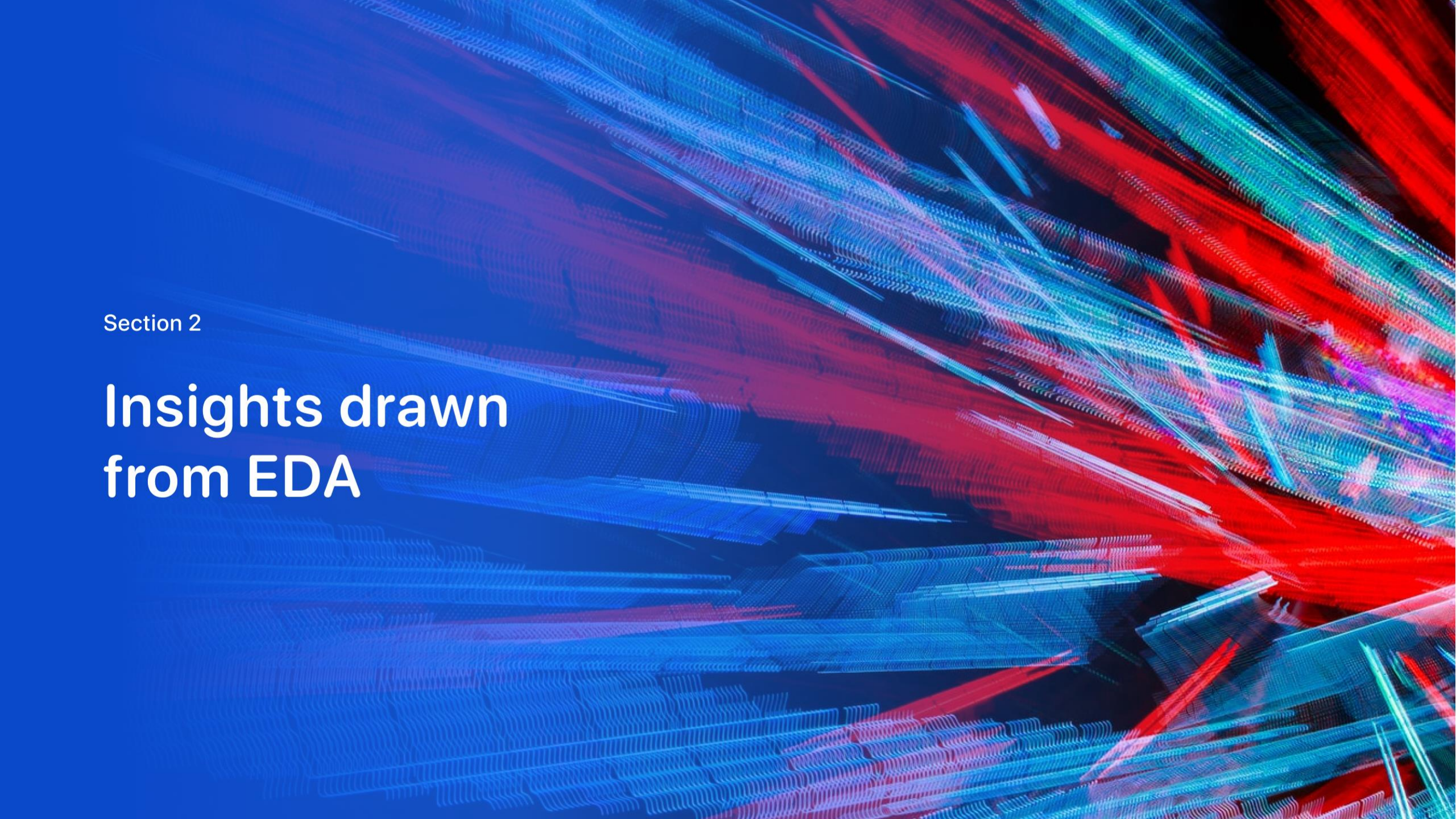
- The launch sites must be near the coastline, and have logistic facilities like railroads, roads in the proximities. It is imperative that the launch sites are far from populated cities and are best positioned in empty areas or in Air Force bases.
- The most successful launch site is KSC LC39A and its success rate is of 76.9%



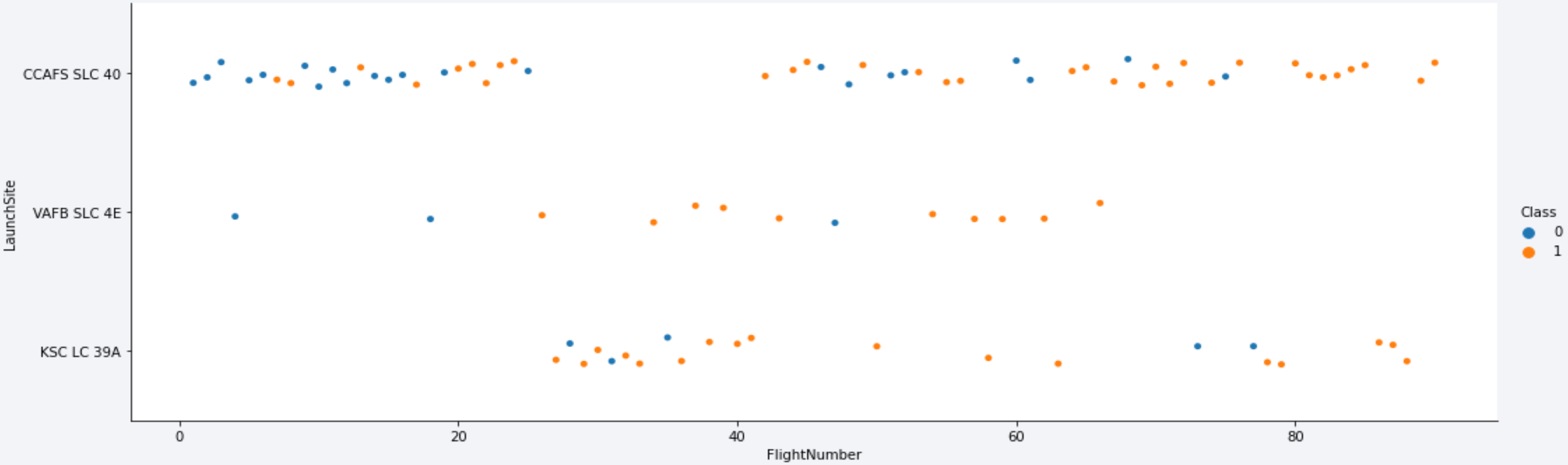



Section 2

# Insights drawn from EDA

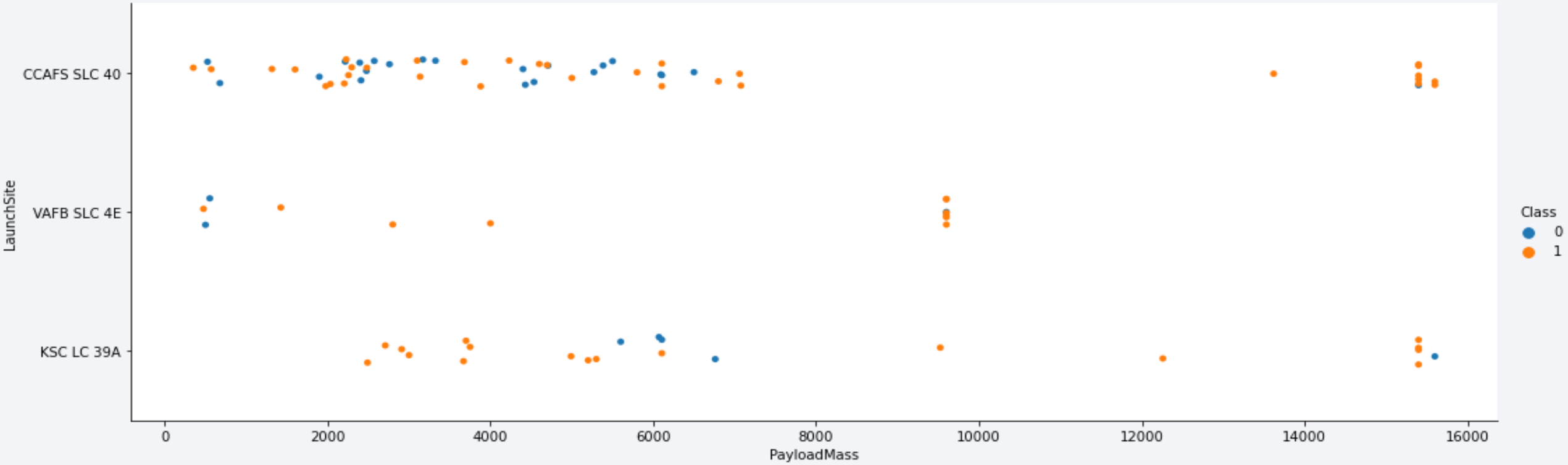


# Flight Number vs. Launch Site




We can see that for Launch Site CCAFS SLC 40, the success rate increases with the payload mass. This is a general trend also in the other two launch sites, but it's more evident in CCAFS SLC 40 due to the number of launches.

# Payload vs. Launch Site

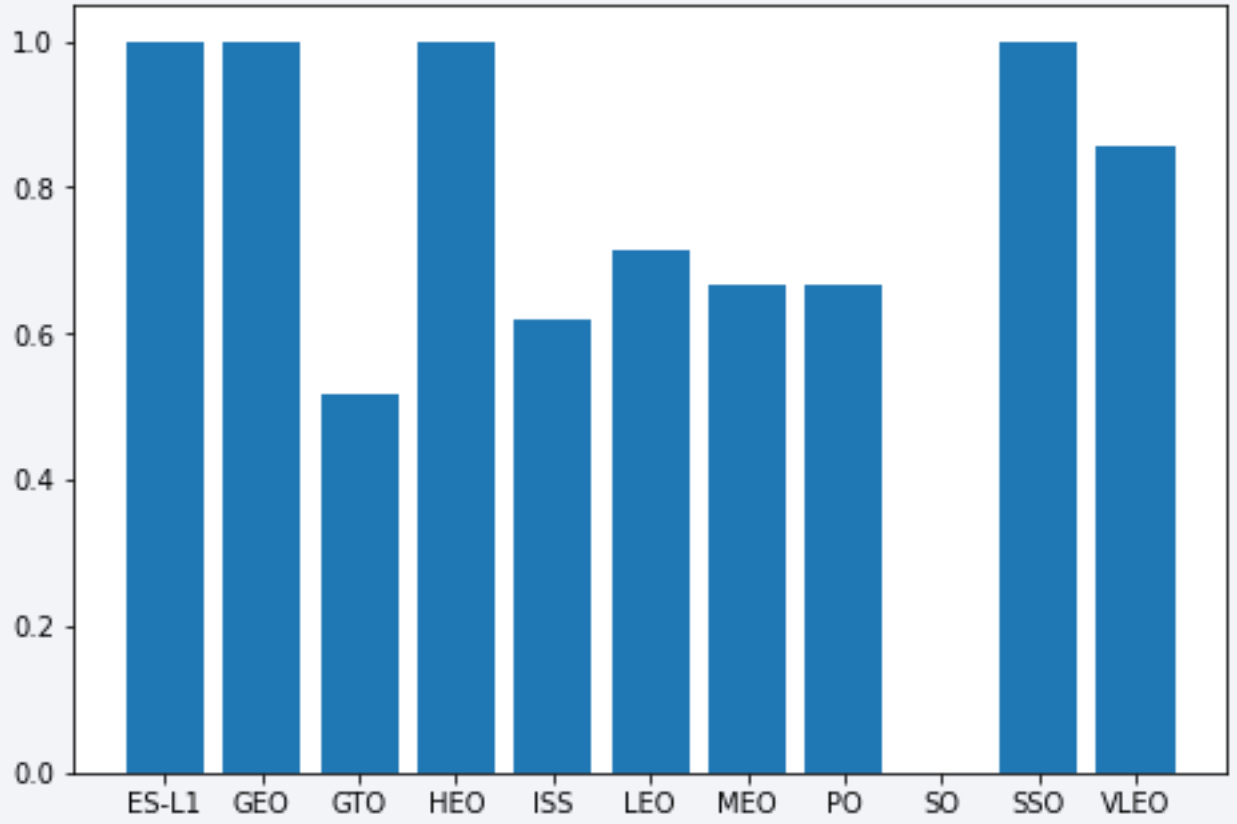



Here we have no real insights, except for the fact that no payloads greater than 10000 were launched from VAFB SLC 4E.

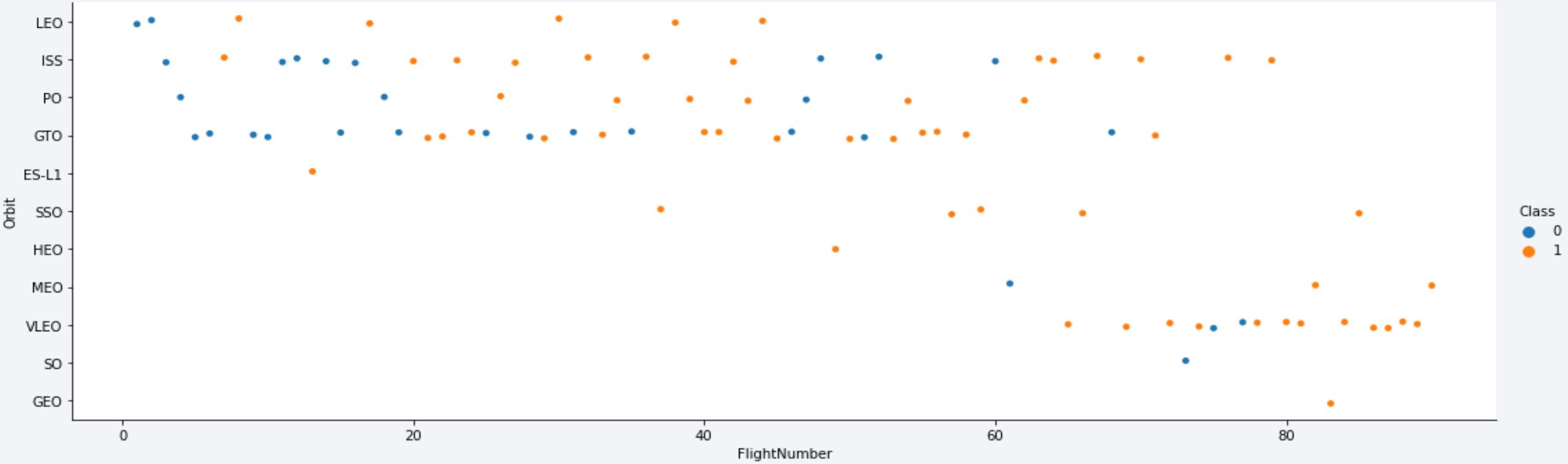
# Success Rate vs. Orbit Type

From the bar chart, we can see that the orbit with the highest success rate are:

- ES-L1 (the Lagrange Point)
- GEO (geosynchronous 35,786 km)
- HEO (Elliptic orbit with high eccentricity
- SSO (Sun-synchronous orbit)

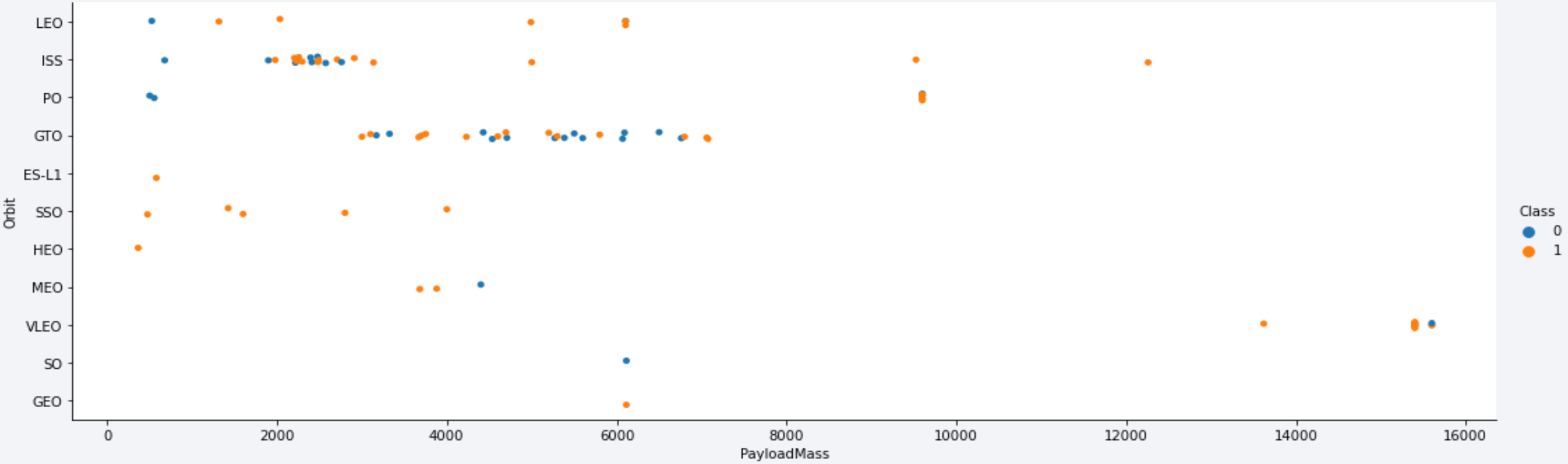


# Flight Number vs. Orbit Type




This chart shows that the success in the LEO orbit is related to the number of flights; for all the launches there is a first 'learning phase' of approx. 20 flights, then a spike in success rate. Also, the VLEO orbit gains being the last orbit approached by SpaceX.

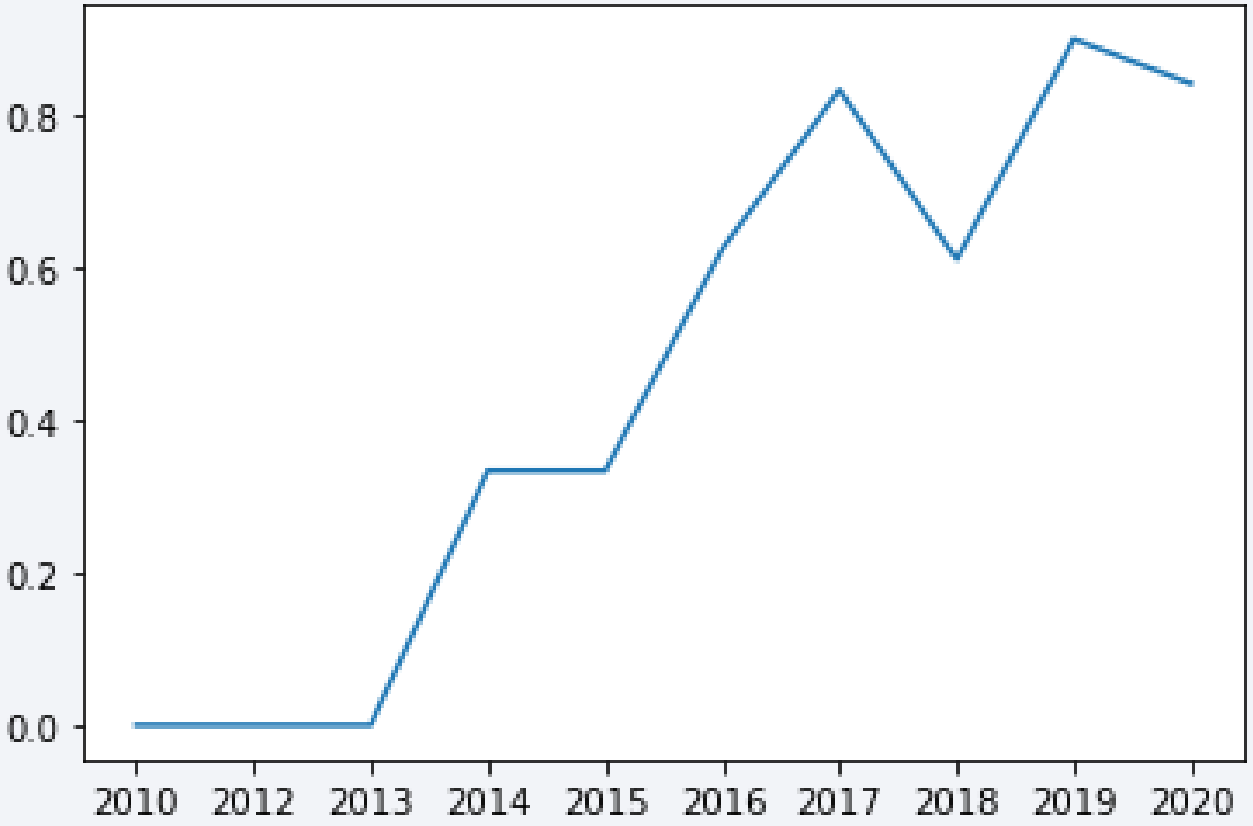
# Payload vs. Orbit Type




The chart shows that for higher payloads the success rate for LEO, Polar and ISS is higher. For GTO, we can't really see a trend in this direction, due to the presence, for every payload, of both successful and unsuccessful missions.

# Launch Success Yearly Trend

From the graphic, we can see that the success rate has **steadily improved since 2013**, with a little drop in 2018.



SELECT LAUNCH_SITE FROM SPACEXTBL GROUP BY LAUNCH_SITE

# All Launch Site Names

- Find the names of the unique launch sites
- Here we select the column LAUNCH_SITE from the SPACEXTBL database, and we group it by LAUNCH_SITE to have only one value of each launch site.

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

# Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with `CCA`
- Here we select all the values of the DB with *, then we filter it with a WHERE clause in the LAUNCH_SITE column, adding the LIKE '%CCA%' which returns only the values with "CCA" in it. We then add the LIMIT 5 to show only the first 5 occurrences.

| DATE | time__utc_ | booster_version | launch_site | payload | payload_mass__kg_ | orbit | customer | mission_outcome | landing__outcome |
|---|---|---|---|---|---|---|---|---|---|
| 2010-06-04 | 18:45:00 | F9 v1.0 B0003 | CCAFS LC-40 | Dragon Spacecraft Qualification Unit | 0 | LEO | SpaceX | Success | Failure (parachute) |
| 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 (parachute) |
| 2012-05-22 | 07:44:00 | F9 v1.0 B0005 | CCAFS LC-40 | Dragon demo flight C2 | 525 | LEO (ISS) | NASA (COTS) | Success | No attempt |
| 2012-10-08 | 00:35:00 | F9 v1.0 B0006 | CCAFS LC-40 | SpaceX CRS-1 | 500 | LEO (ISS) | NASA (CRS) | Success | No attempt |
| 2013-03-01 | 15:10:00 | F9 v1.0 B0007 | CCAFS LC-40 | SpaceX CRS-2 | 677 | LEO (ISS) | NASA (CRS) | Success | No attempt |

# Total Payload Mass

- Calculate the total payload carried by boosters from NASA
- Here we select the sum of the total payload mass, and we add the WHERE CUSTOMER LIKE clause to filter the values that begin with “NASA”. We return the value in the query as “total_payload_mass”.

| total_payload_mass |
| --- |
| 99980 |

# Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1
- We select the average of the payload mass, and we add the clause WHERE to select the booster versions that begin with “F9 v1.1”.

| avg_payload_mass |
| --- |
| 2534 |

# First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad
- We select MIN(DATE), the “smallest” date and we give to the return data the name “first_succ_landing_gp”, adding a filtering WHERE clause to show the “Success (ground pad)” on the LANDING__OUTCOME column.

| first_succ_landing_gp |
| --- |
| 2015-12-22 |

# Successful Drone Ship Landing with Payload between 4000 and 6000

- List the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000
- We select the booster version, and we give it the name “booster_name” and the payload mass adding two filtering WHERE clauses; “Success (drone ship) in LANDING__OUTCOME and PAYLOAD_MASS__KG between 4000 and 6000

| booster_name | payload_mass |
|---|---|
| F9 FT B1022 | 4696 |
| F9 FT B1026 | 4600 |
| F9 FT B1021.2 | 5300 |
| F9 FT B1031.2 | 5200 |

# Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes
- Here we select the mission outcome and its count, naming it “launches” and we group it by mission outcome.

| mission_outcome | launches |
|---|---|
| Failure (in flight) | 1 |
| Success | 99 |
| Success (payload status unclear) | 1 |

# Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass
- Here we select with “booster name” the booster version, and the payload mass, adding a WHERE clause for filtering. Then we add a subquery, selecting the all the maximum values from PAYLOAD_MASS__KG_.

| booster_name | payload_mass |
|---|---|
| F9 B5 B1048.4 | 15600 |
| F9 B5 B1049.4 | 15600 |
| F9 B5 B1051.3 | 15600 |
| F9 B5 B1056.4 | 15600 |
| F9 B5 B1048.5 | 15600 |
| F9 B5 B1051.4 | 15600 |
| F9 B5 B1049.5 | 15600 |
| F9 B5 B1060.2 | 15600 |
| F9 B5 B1058.3 | 15600 |
| F9 B5 B1051.6 | 15600 |
| F9 B5 B1060.3 | 15600 |
| F9 B5 B1049.7 | 15600 |

# 2015 Launch Records

- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
- We select date, booster version, launch site and landing outcome, and we add a WHERE filtering clause joined to a LIKE clause to display the values that have 2015 as DATE.

| DATE | booster_version | launch_site | landing__outcome |
|---|---|---|---|
| 2015-01-10 | F9 v1.1 B1012 | CCAFS LC-40 | Failure (drone ship) |
| 2015-04-14 | F9 v1.1 B1015 | CCAFS LC-40 | Failure (drone ship) |

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

- 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
- Here we select the landing outcome and its count, adding a filtering WHERE clause to the DATE column, and a BETWEEN clause to select the dates of our interest. At the end we order the results by the count of the LANDING__OUTCOME column and we add the DESC clause, to display bigger results first.

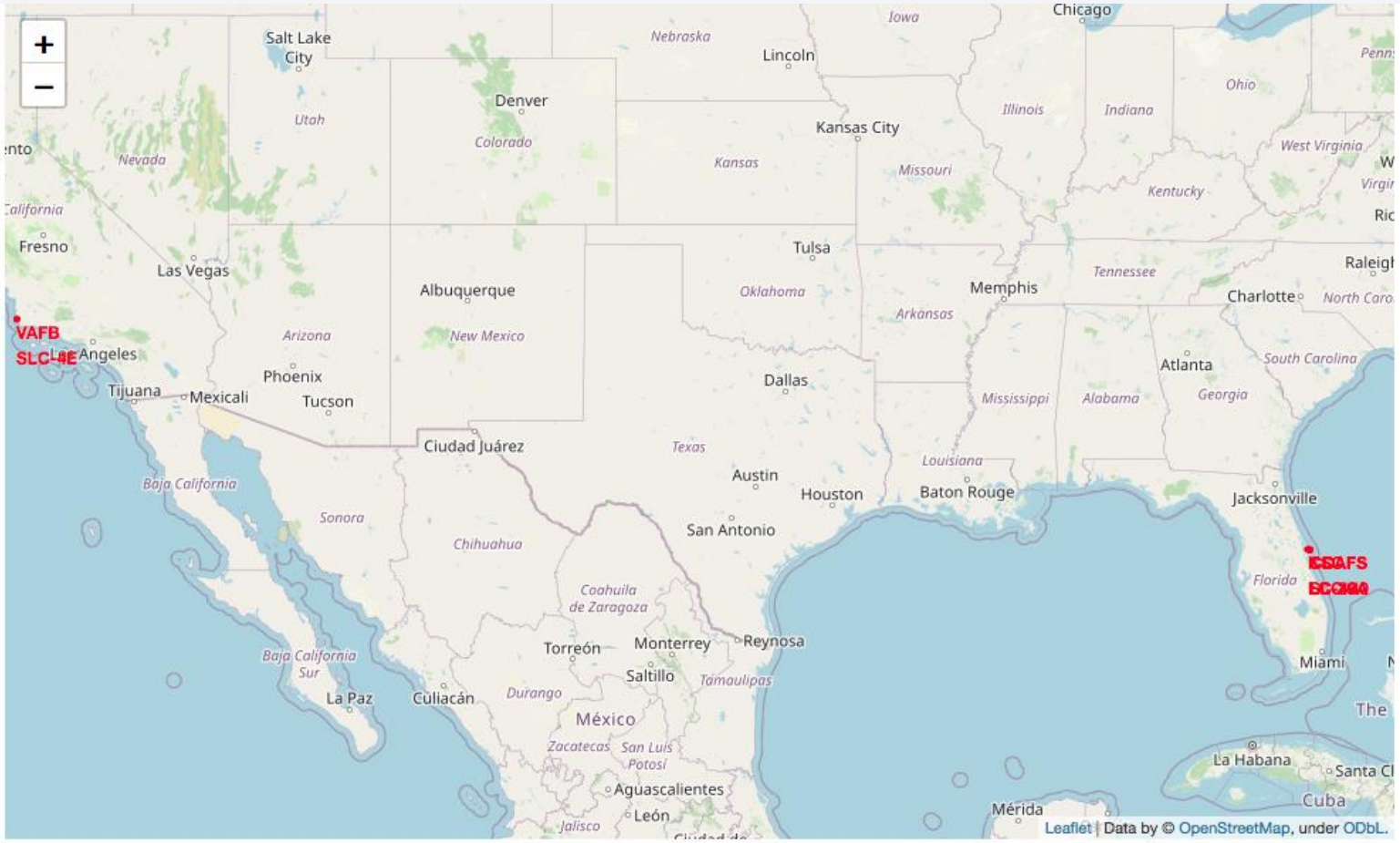
| LANDING OUTCOME | |
|---|---|
| No attempt | 10 |
| Failure (drone ship) | 5 |
| Success (drone ship) | 5 |
| Controlled (ocean) | 3 |
| Success (ground pad) | 3 |
| Failure (parachute) | 2 |
| Uncontrolled (ocean) | 2 |
| Precluded (drone ship) | 1 |

Section 4

# Launch Sites Proximities Analysis

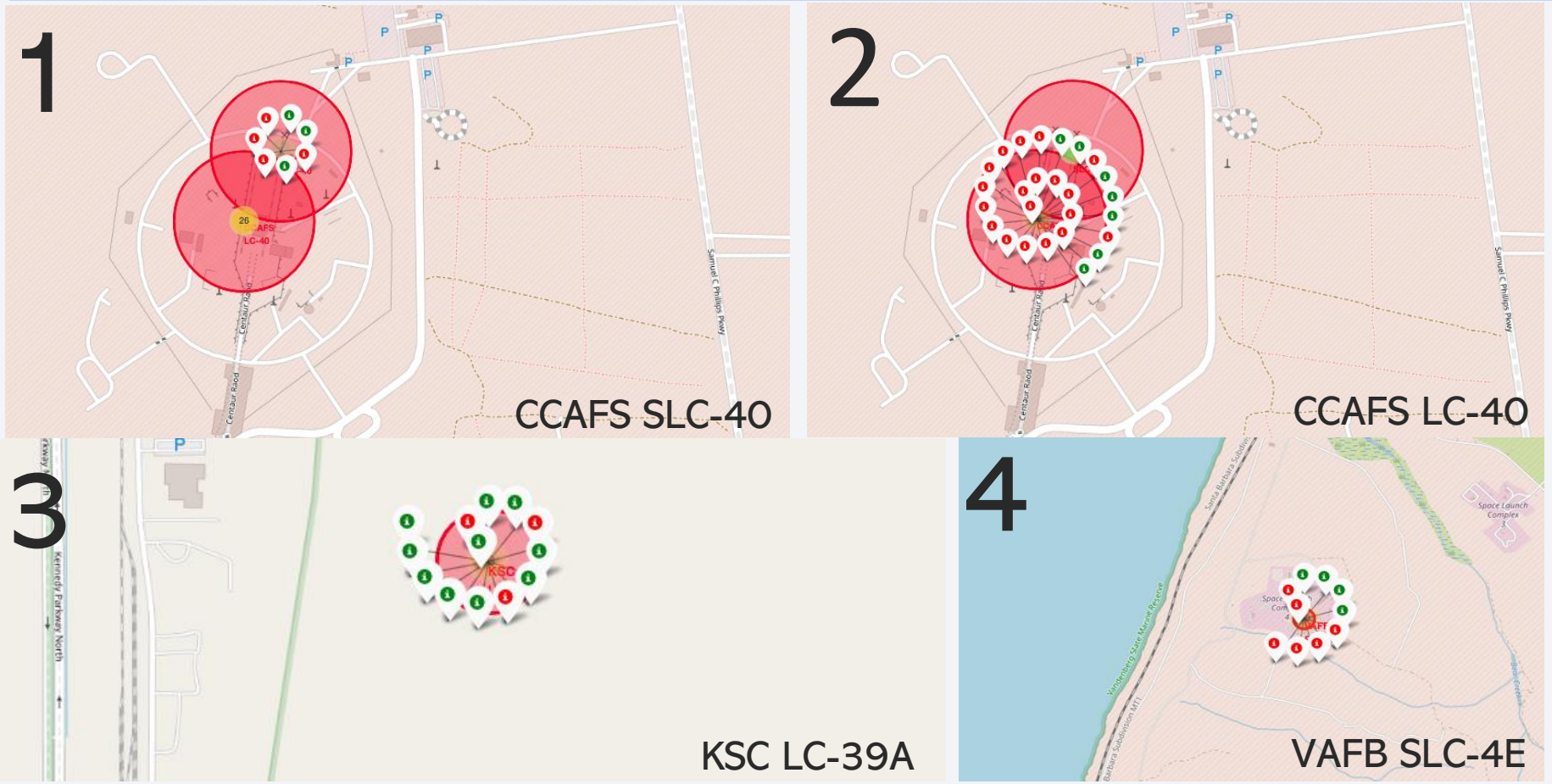
# Launch Sites Locations

Out[10]:



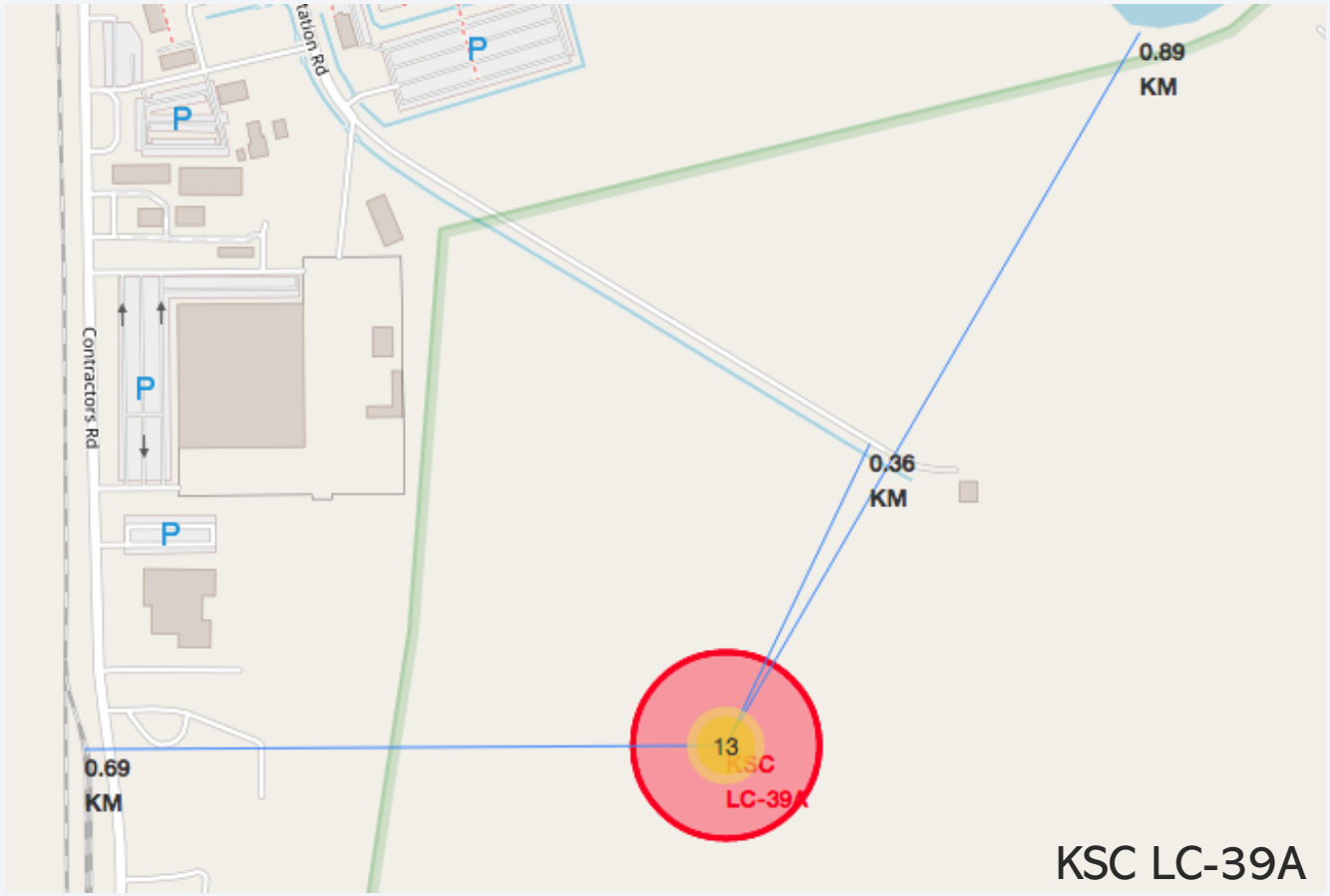

All the launch sites are in isolated areas, but near to the coastline and to logistic resources such as roads and railroads.

# Success & Failure Markers of each Launch Site




From the markers on each launch site, we can see that site number 3 (KSC LC-39A) has the highest success rate.

# Distances to a selected Launch Site

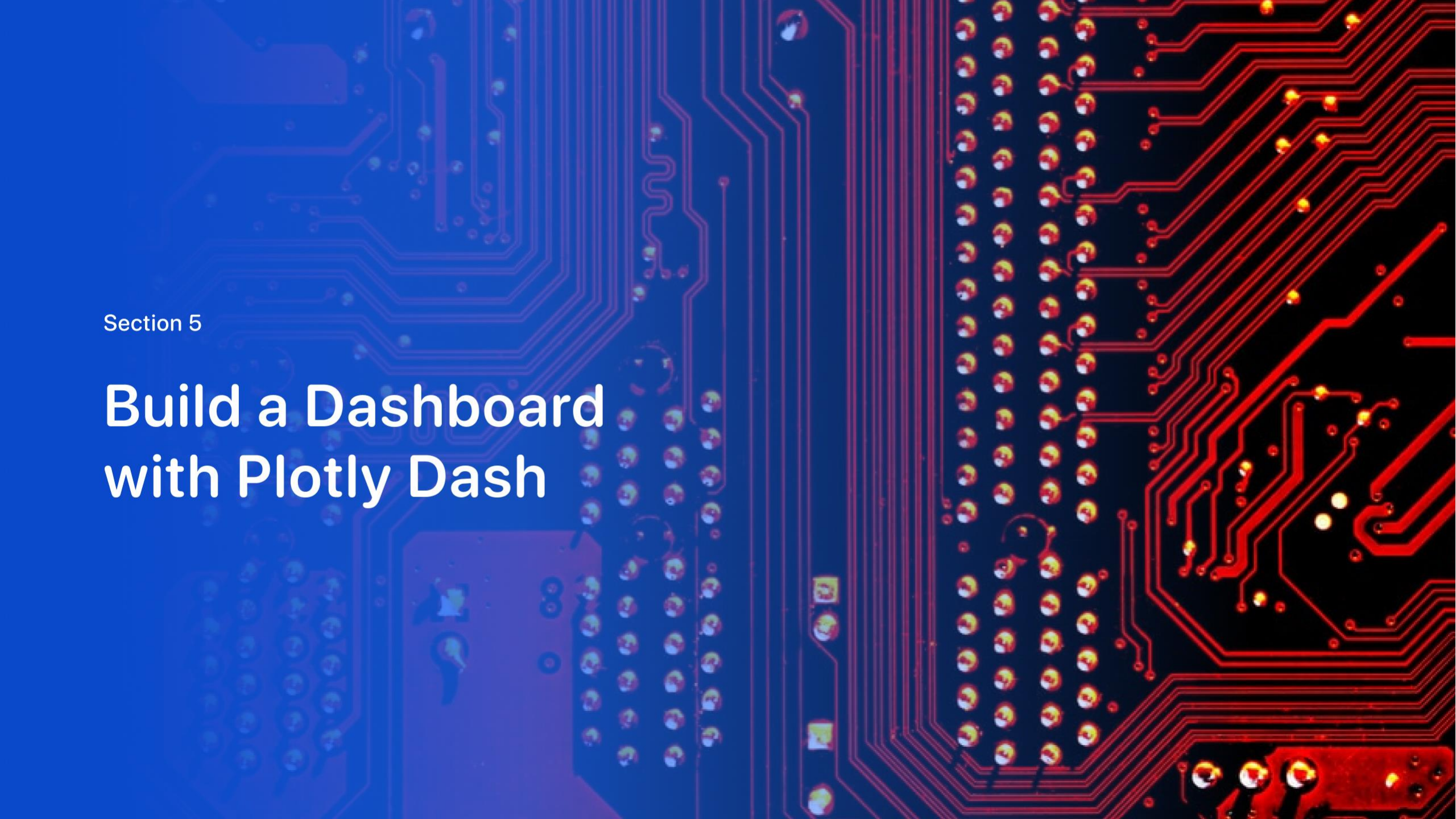


KSC LC-39A

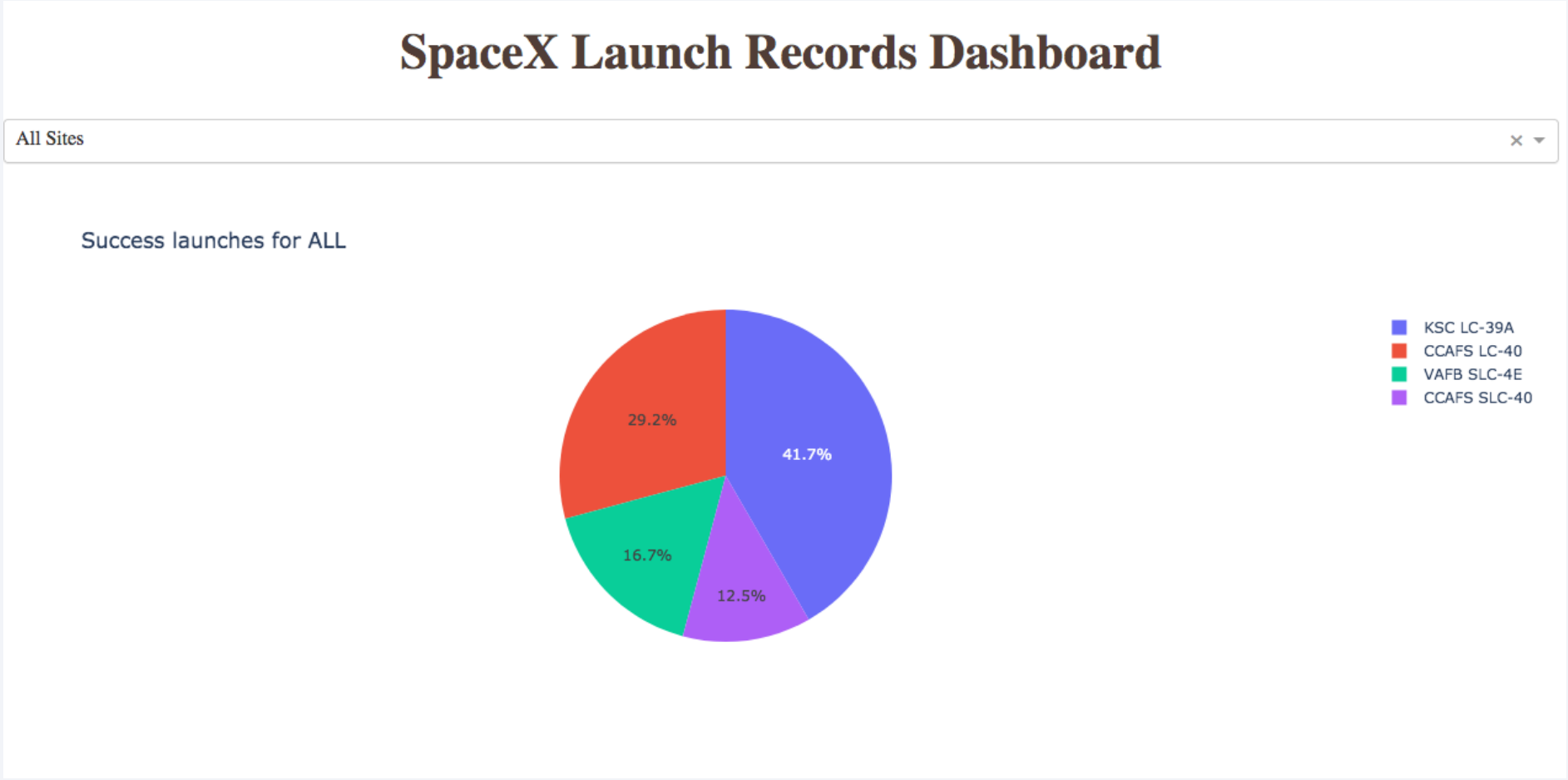
- The drawing shows that the launch site with the highest success rate has in its proximities a railroad, a road and is also near to the coastline.

Section 5

# Build a Dashboard with Plotly Dash

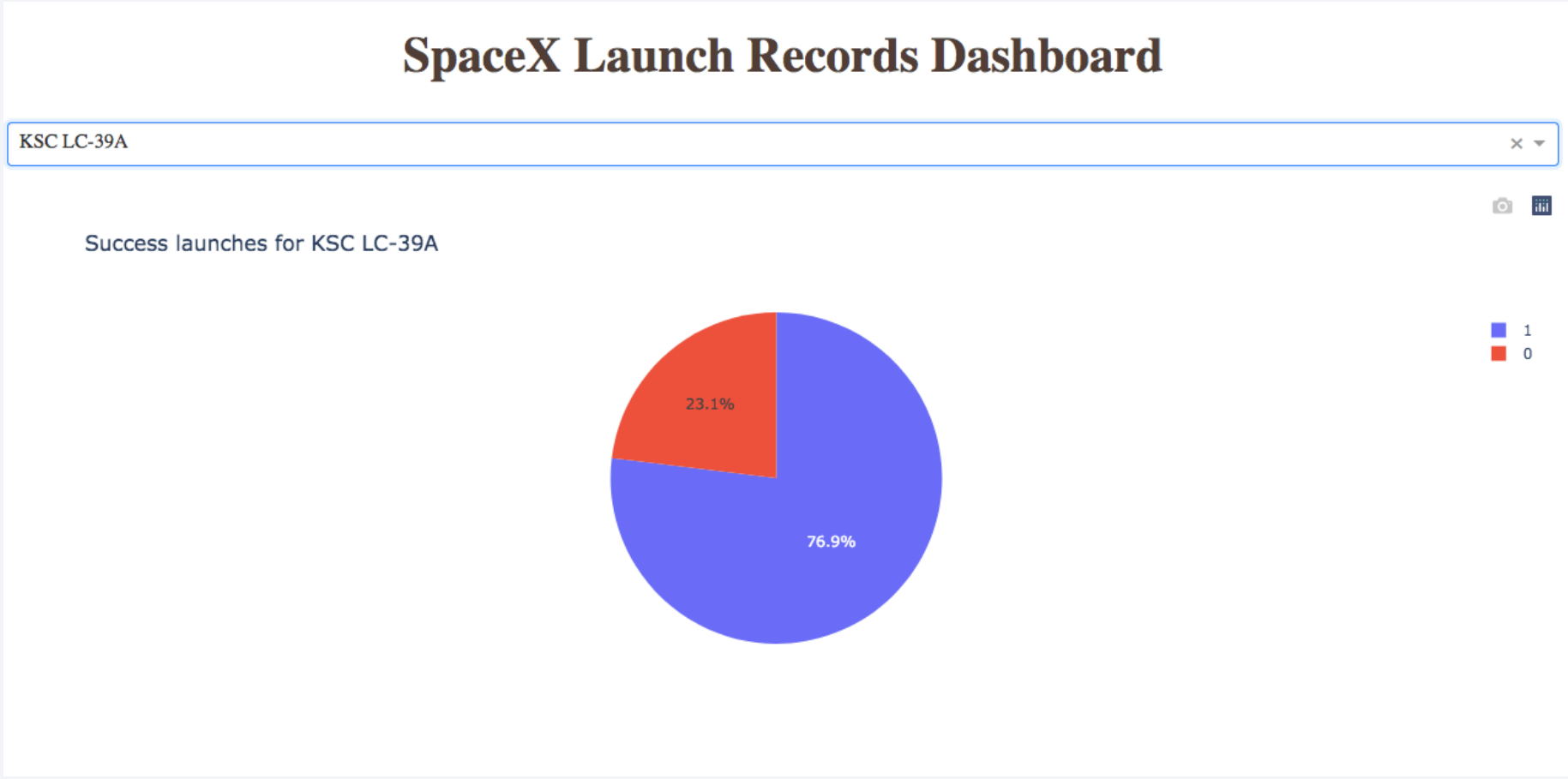


# Success Launch Dashboard for All Sites



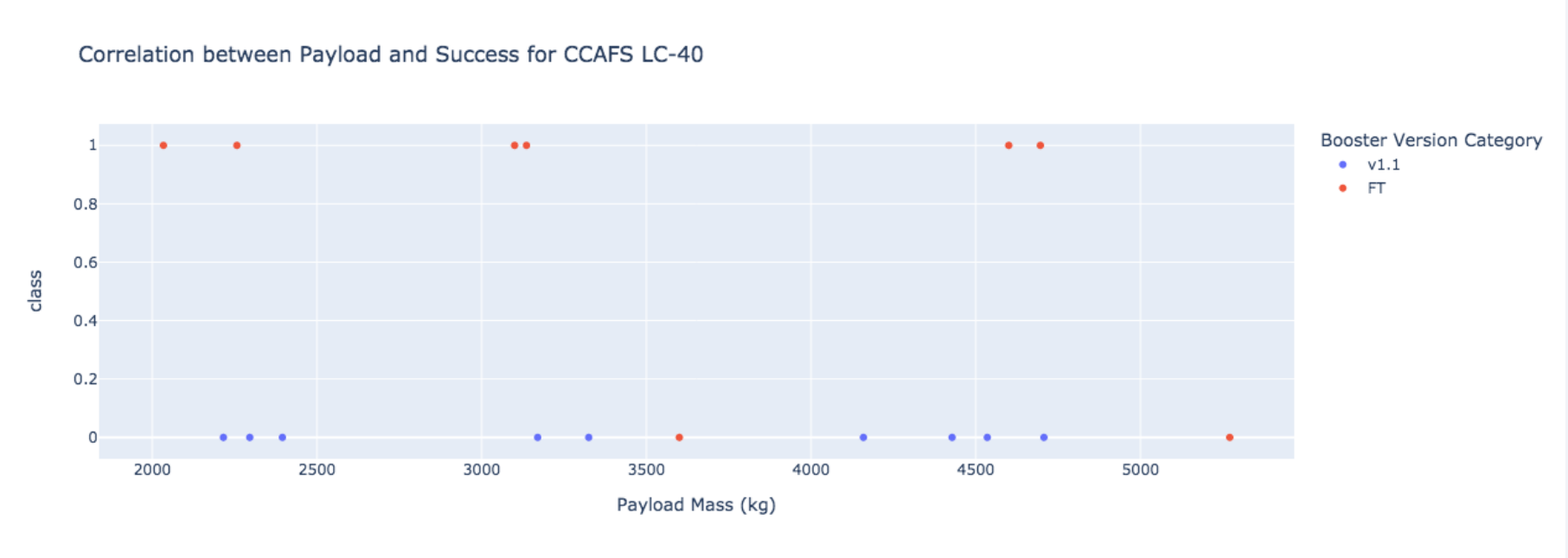

From the pie chart we can see that the launch site with the highest success rate is KSC LC - 39A, the Kennedy Space Center Launch Complex 39A.

# Success Launch Rate for KSC LC-39A



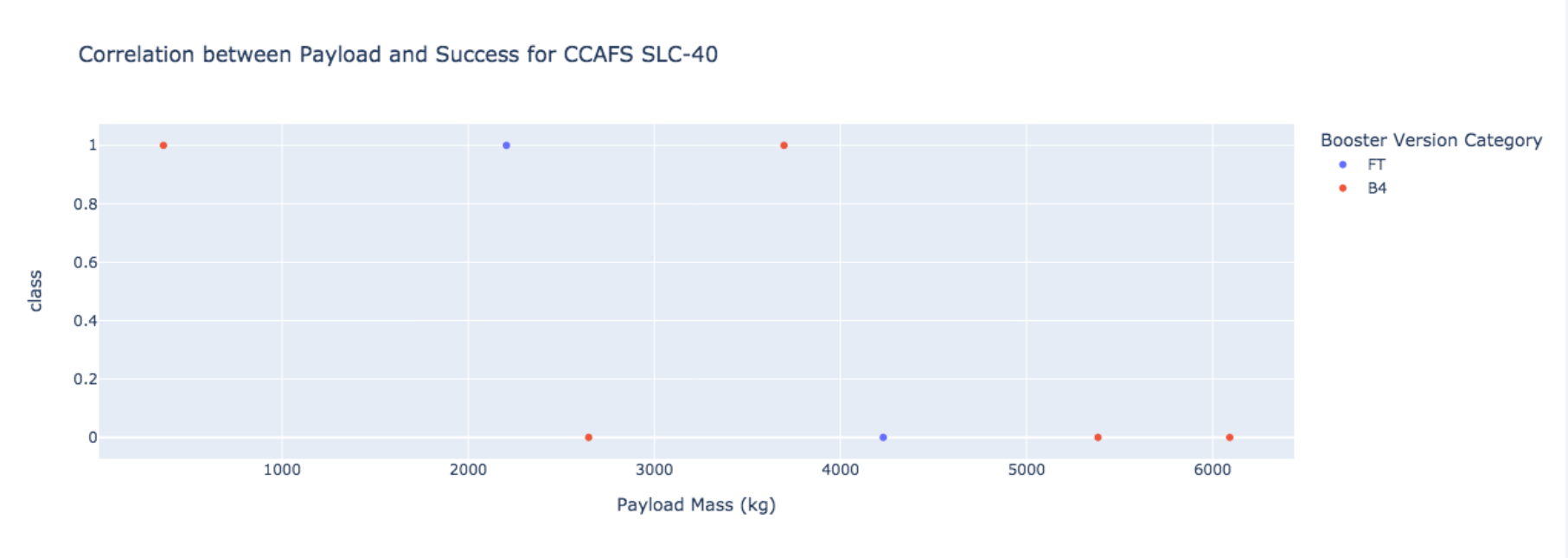

From this pie chart we can see that the success rate for the KSC LC-39A launch site is 76.9%.

# Analysis of the Success Rate for CCAFS LC-40



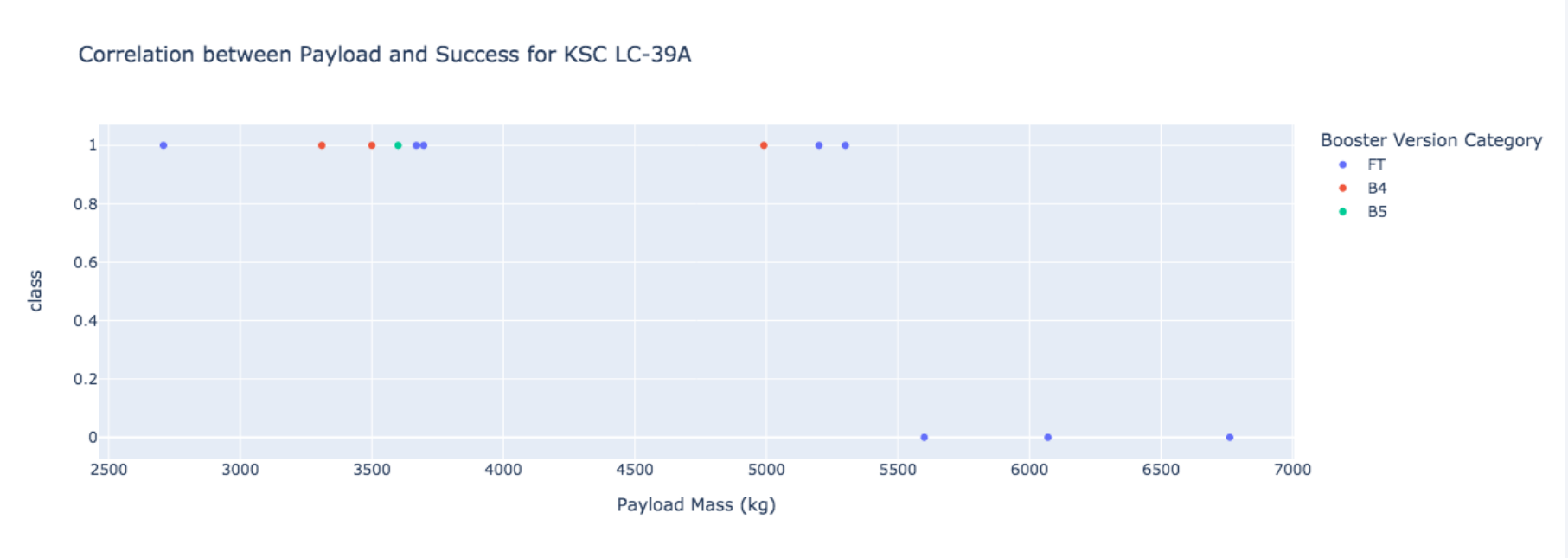

Here we can see that all the successful launches had a payload mass between 2000 and 5000 and the booster FT was utilized.

# Analysis of the Success Rate for CCAFS SLC-40




We can't really draw any conclusion from this plot, except for the fact that the booster B4 has a 66% success rate for payloads under 4000.

# Analysis of the Success Rate for KSC LC-39A




Here we can see that all the launches with payload over 2500 and under 5500 were successful. The launches with payload over 5500 were all unsuccessful.

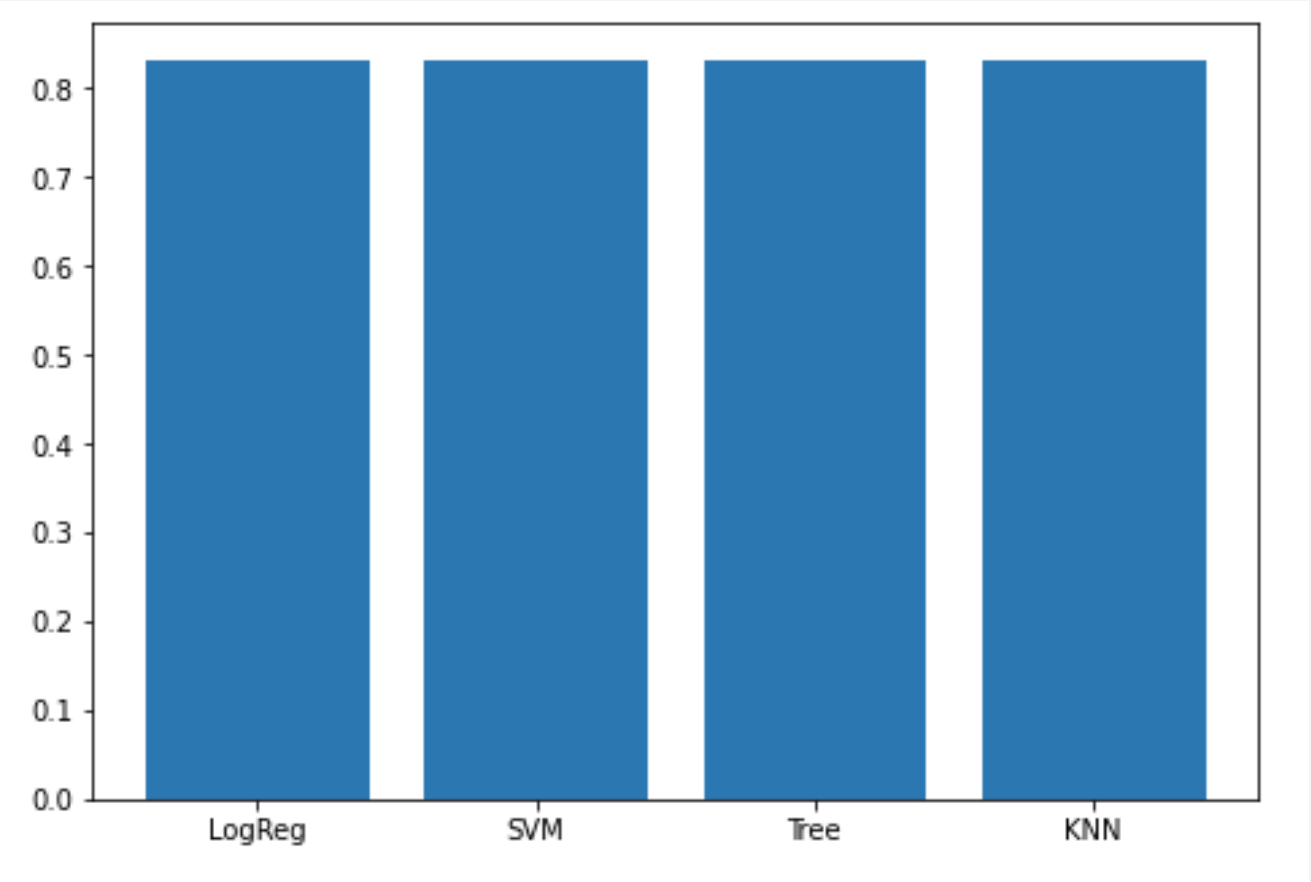
# Analysis of the Success Rate for VAFB SLC-4E


The graphic shows that at this launch site, most of the launches were unsuccessful, and that the booster B4 was the most used one for bigger payloads.

Section 6

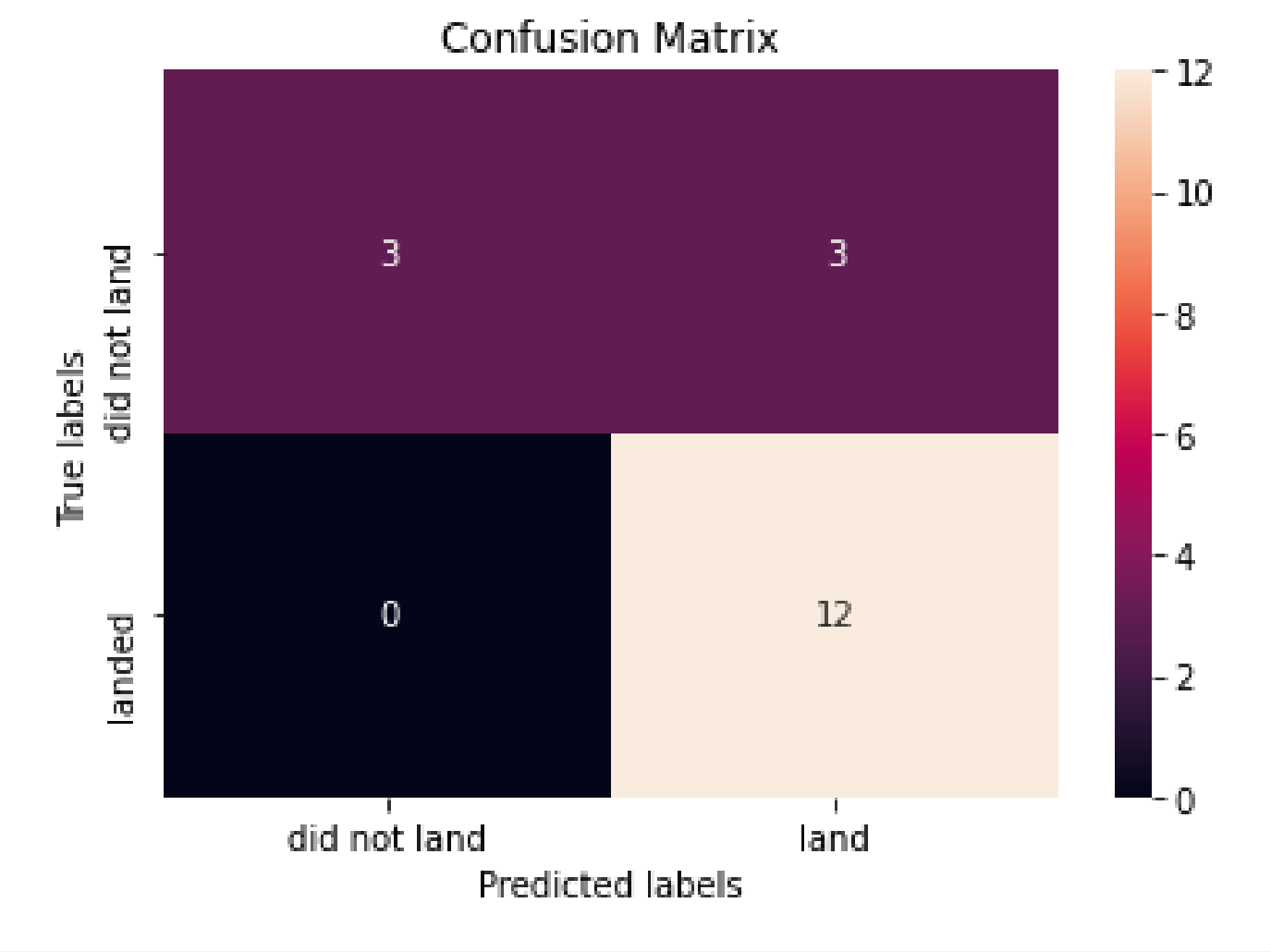
# Predictive Analysis (Classification)

# Classification Accuracy




All methods give the result of 83.3%. Therefore, all methods perform equally.

# Confusion Matrix




- Due to the limited number of elements in the test set (18) all the models have the same accuracy score, and the same confusion matrix. The major problem is in the false positives.

# Conclusions

- It is needed to **fund at least 2 years of failures** to have a successful landing.
- The two launch sites that should be preferred are **KSC LC39A** and **VAFB SLC 4E**; however, **for bigger payloads CCAFS SLC40 is preferred**.
- For SpaceX there was a first '**learning phase**' of approx. **20 flights**, then a spike in success rate.
- The most successful launches are to the orbits **GEO, HEO, SSO** and **ES-L1**.
- The boosters with code **F9 B5 B104X.X, 105X.X** and **106X.X** are **the best performing with higher payloads**.
- For building new launch sites, if necessary, it's best to choose a location with **good railroads and roads connections**, **close to the coastline and far from any populated city**.

# Appendix

## The following SQL queries were performed:

- SELECT LAUNCH_SITE FROM SPACEXTBL GROUP BY LAUNCH_SITE
- SELECT * FROM SPACEXTBL WHERE LAUNCH_SITE LIKE '%CCA%' LIMIT 5
- SELECT SUM(PAYLOAD_MASS__KG_) AS TOTAL_PAYLOAD_MASS FROM SPACEXTBL WHERE CUSTOMER LIKE 'NASA%'
- SELECT AVG(PAYLOAD_MASS__KG_) AS AVG_PAYLOAD_MASS FROM SPACEXTBL WHERE BOOSTER_VERSION LIKE 'F9 v1.1%'
- SELECT MIN(DATE) AS FIRST_SUCC_LANDING_GP FROM SPACEXTBL WHERE LANDING__OUTCOME = 'Success (ground pad)'
- SELECT BOOSTER_VERSION AS BOOSTER_NAME, PAYLOAD_MASS__KG_ AS PAYLOAD_MASS FROM SPACEXTBL WHERE LANDING__OUTCOME = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000
- SELECT MISSION_OUTCOME, COUNT(MISSION_OUTCOME) AS LAUNCHES FROM SPACEXTBL GROUP BY MISSION_OUTCOME
- SELECT BOOSTER_VERSION AS BOOSTER_NAME, PAYLOAD_MASS__KG_ AS PAYLOAD_MASS FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL)
- SELECT DATE, BOOSTER_VERSION, LAUNCH_SITE, LANDING__OUTCOME FROM SPACEXTBL WHERE LANDING__OUTCOME = 'Failure (drone ship)' AND DATE LIKE '%2015%'
- SELECT LANDING__OUTCOME AS "LANDING OUTCOME", COUNT(LANDING__OUTCOME) AS " " FROM SPACEXTBL WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY LANDING__OUTCOME ORDER BY COUNT(LANDING__OUTCOME) DESC

# Appendix

The following code was added:

```
labels = ['LogReg', 'SVM', 'Tree','KNN']
values_y = [logr, svm, tree, knn]
fig = plt.figure()
ax = fig.add_axes([0 , 0, 1 ,1])
ax.bar(labels, values_y)
plt.show()
```

# Thank you!