IBM Developer
SKILLS NETWORK

# Winning Space Race with Data Science

Petrusca Mello Costa Filha
19.09.2023

# Outline

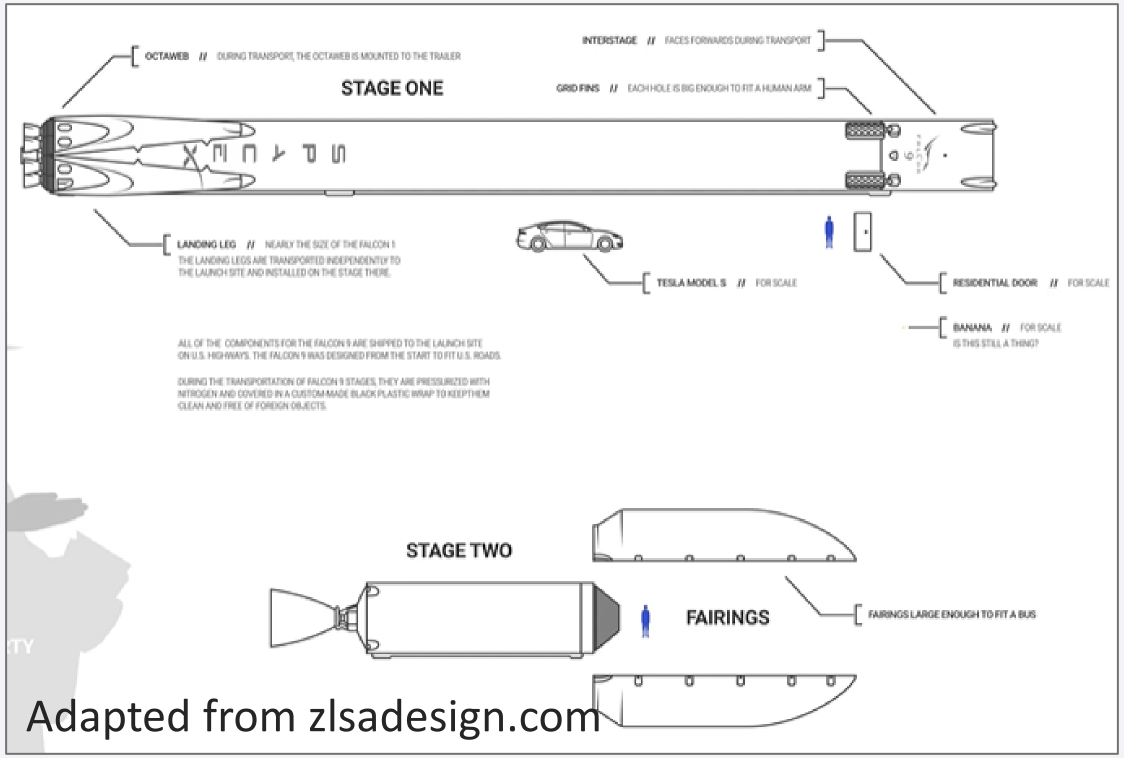
# Executive Summary

- This project aims to analyze the SpaceX launch data and predict the landing success of the Falcon 9 first stage. The data sources include an API that provides structured JSON data and web pages that contain relevant information. The data was cleaned, normalized, and converted into a Pandas data frame for further analysis. Using visualization and SQL, the project explored the factors that influence the landing success, such as launch site, payload mass, orbit, and booster version. The project also used interactive visual analytics tools such as Folium and Plotly Dash to examine the geographical and temporal aspects of the launch data. Finally, the project built and evaluated several classification models using machine learning techniques to predict the landing outcome based on the available features.

- The data analysis reveals that SpaceX has improved its technology and experience over the years, resulting in more successful landings. The analysis also shows that the most successful payload range is between 2k and 6k kg, and that the FT booster version has the highest success rate. Moreover, the analysis identifies four orbits that have a perfect success rate: ES-L1, SSO, HEO and GEO. The analysis also finds that the success rate is more related to the payload range than the orbit type. Furthermore, the analysis shows that the KSC LC site has the best performance among the launch sites, and that all sites have some common features, such as proximity to highways, railways, equator line and coastlines. Finally, the analysis demonstrates that the tree classifier is the best predictive model for launch success, with an accuracy of 94%.

# Introduction

We are in the age of commercial space, for numerous reasons that range from satellite communications through space tourism. In this context, the main challenge is the actual cost of launching. In this scenario one company, SpaceX has been successful in reducing costs by implementing the reuse the first stage.



Adapted from zlsadesign.com

Seeing the opportunity to better understand this amazing technology and develop our own, we will use data science to predict the features that best correlates to the success of reuse of the first stage landing, as well as predicting their success rate implementing multiple machine learning models.

Section 1

# Methodology

# Methodology

## Executive Summary

- Data collection methodology:
  - SpaceX launch data was gathered from an API (SpaceX REST API)
  - Web scraping in related Wiki pages.
- Perform data wrangling
  - Data from API was normalized from structured JSON data into a flat table.
  - The web scrapped data was converted into a Pandas data frame using the Python Beautiful Soup package.
- Perform exploratory data analysis (EDA) using visualization and SQL
  - Determined what attributes are correlated with successful landings then use these features with machine learning to automatically predict if the first stage can land successfully.
- Perform interactive visual analytics using Folium and Plotly Dash
  - Analyzed launch site geo and proximities with Folium to explain how to choose an optimal launch site.
- Perform predictive analysis using classification models
  - Build, tuned, and evaluated classification models to predict if the first stage of the Falcon 9 lands. successfully.

# Data Collection

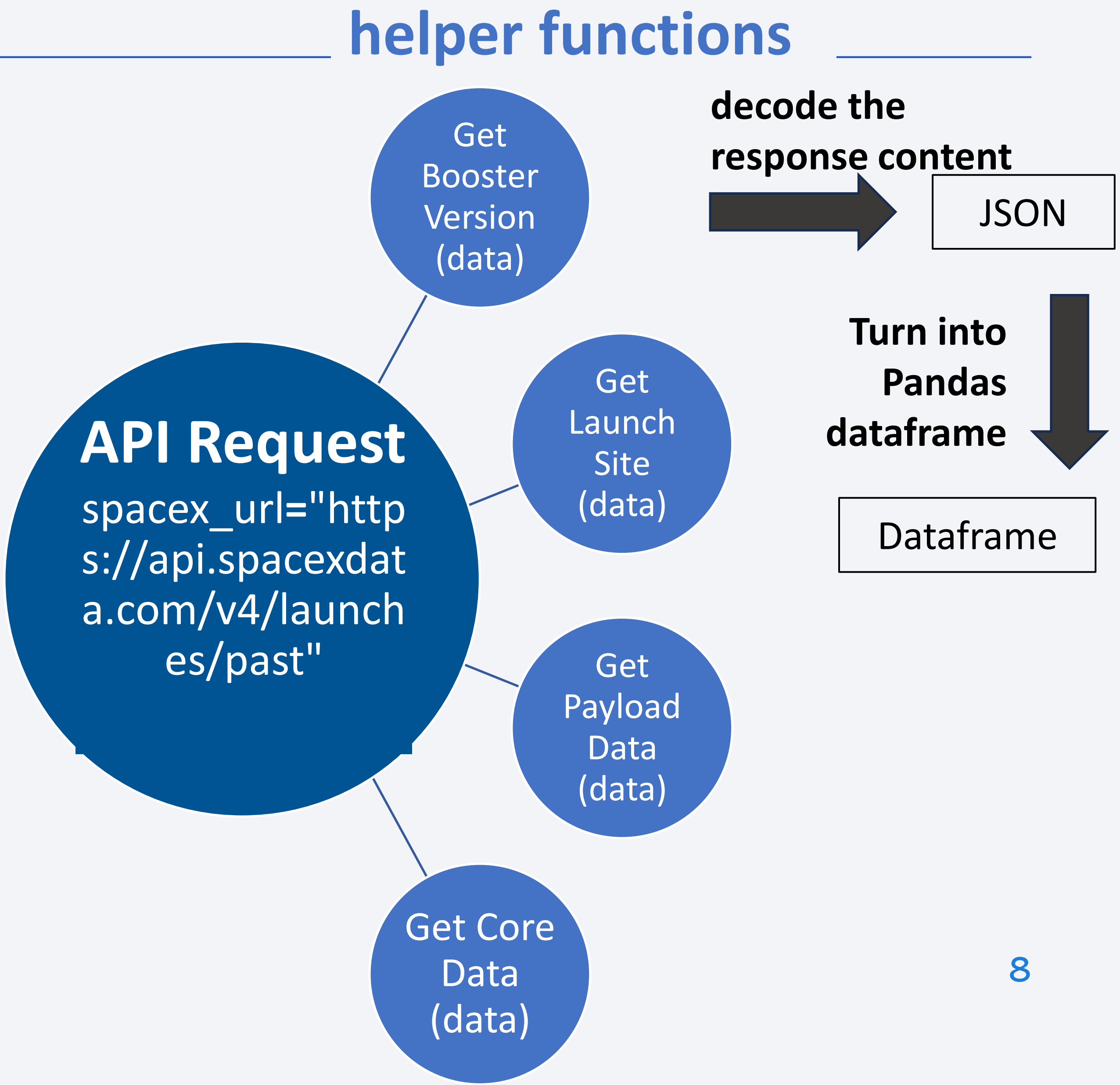
- SpaceX launch data was gathered from an API (SpaceX REST API) getting information about the rocket used, payload delivered, launch specifications, landing specifications, and landing outcome, as well as using web scraping related Wiki pages.

For the complete walk-through access links:

- API (SpaceX REST API)
  - https://github.com/PetruscaF/capstoneIBM/blob/main/jupyter-labs-spacex-data-collection-api.ipynb
- Web scraping
  - https://github.com/PetruscaF/capstoneIBM/blob/main/jupyter-labs-webscraping.ipynb

# Data Collection – SpaceX API

- The first step was to request rocket launch data from SpaceX API with the URL
- Then defined Helper functions that uses API to extract information using identification numbers in the launch data.
- Then decoded the response content as a Json using .json() and turn it into a Pandas data frame using .json_normalize()
- Finally used the helper functions to get information about the launches using the IDs given for each launch. Specifically, the columns rocket, payloads, launchpad, and cores then store in lists and used to create a new data frame.

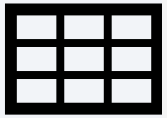


# Data Collection - Scraping

- The first step was to use HTTP GET method to request the Falcon9 Launch HTML page on Wikipedia (using static version of page), as an HTTP response.
- Then create a BeautifulSoup object from the HTML response
- In the response content, found all tables on the wiki page then collected all relevant column names from the HTML table header
- Then extract each table and then Append the column on each correlated dictionary created with the names of each column.
- Finally filed in the parsed launch record values into launch_dict, to create a data frame from it.


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

BeautifulSoup object



Data frame

```
launch_dict['Flight No.'] = flight_number_array
launch_dict['Date'] = date_array
launch_dict['Time'] = time_array
launch_dict['Version Booster'] = bv_array
launch_dict['Launch site'] = launch_site_array
launch_dict['Payload'] = payload_array
launch_dict['Payload mass'] = payload_mass_array
launch_dict['Orbit'] = orbit_array
launch_dict['Customer'] = customer_array
launch_dict['Launch outcome'] = launch_outcome_array
launch_dict['Booster landing'] = booster_landing_array
```

# Data Wrangling

https://github.com/PetruscaF/capstoneIBM/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

- The objective for the data wrangling was to convert the mission outcomes from categorical values to numerical ones on a column Class, where 1 means the booster successfully landed 0 means it was unsuccessful, using pandas and NumPy libraries.

| # | FlightNumber | Date | BoosterVersion | PayloadMass | Orbit | LaunchSite | Outcome | Flights | GridFins | Reused | Legs | LandingPad | Block | ReusedCount | Serial | Longitude | Latitude |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2010-06-04 | Falcon 9 | 6104.959412 | LEO | CCAFS SLC 40 | None None | 1 | False | False | False | NaN | 1.0 | 0 | B0003 | -80.577366 | 28.561857 |
| 1 | 2 | 2012-05-22 | Falcon 9 | 525.000000 | LEO | CCAFS SLC 40 | None None | 1 | False | False | False | NaN | 1.0 | 0 | B0005 | -80.577366 | 28.561857 |
| 2 | 3 | 2013-03-01 | Falcon 9 | 677.000000 | ISS | CCAFS SLC 40 | None None | 1 | False | False | False | NaN | 1.0 | 0 | B0007 | -80.577366 | 28.561857 |

| # | FlightNumber | Date | BoosterVersion | PayloadMass | Orbit | LaunchSite | Outcome | Flights | GridFins | Reused | Legs | LandingPad | Block | ReusedCount | Serial | Longitude | Latitude | Class |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2010-06-04 | Falcon 9 | 6104.959412 | LEO | CCAFS SLC 40 | None None | 1 | False | False | False | NaN | 1.0 | 0 | B0003 | -80.577366 | 28.561857 | 0 |
| 1 | 2 | 2012-05-22 | Falcon 9 | 525.000000 | LEO | CCAFS SLC 40 | None None | 1 | False | False | False | NaN | 1.0 | 0 | B0005 | -80.577366 | 28.561857 | 0 |
| 2 | 3 | 2013-03-01 | Falcon 9 | 677.000000 | ISS | CCAFS SLC 40 | None None | 1 | False | False | False | NaN | 1.0 | 0 | B0007 | -80.577366 | 28.561857 | 0 |

# EDA with Data Visualization

https://github.com/PetruscaF/capstoneIBM/blob/main/jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb

To better understand the data some charts were plotted:

- **FlightNumber** vs. **PayloadMass** – Scatter plot used to observe that the amount of flights had a positive relation with the success landings
- **Flight Number** vs. **Launch Site** - Scatter plot used to observe that the preferred launch site is CCAFS, but the success rate of VAFB and KSC are higher (~77%) compared with CCAFS (60%).
- **Payload** vs. **Launch Site** - Scatter plot that shows VAFB-SLC launchsite there are no rockets launched for heavypayload mass (greater than 10000kg)
- **Orbit** vs. **Success rate** – Bar chart that shows the ES-L1 , SSO, HEO and GEO are the orbits with 100% sucess rate
- **FlightNumber** vs. **Orbit** - Scatter plot that shows LEO orbit the success appears related to the number of flights; on the other hand, not conclusive for GTO orbit.
- **Payload** vs. **Orbit** - Scatter plot that shows with heavy payloads the successful landing or positive landing rate are more for Polar, LEO and ISS; However, not conclusive for GTO.
- **Year** vs. **Success rate** – Line chart that shows the tendency of increasing success rate since 2013.

# EDA with SQL

https://github.com/PetruscaF/capstoneIBM/blob/main/jupyter-labs-eda-sql-edx.ipynb

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

```
%sql select distinct LAUNCH_SITE FROM SPACEX
```

- Display 5 records where launch sites begin with the string 'KSC'

```
%sql select * from spacex where launch_site like 'KSC%'limit 5
```

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

```
%sql select sum(payload_mass__kg_) as total_payload ,customer from spacex where customer = 'NASA (CRS)' group by customer
```

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

```
%sql select avg(payload_mass__kg_) as avg_payload ,booster_version from spacex where booster_version = 'F9 v1.1' group by booster_version
```

- List the date where the first successful landing outcome in drone ship was achieved.

```
%sql select min(DATE), landing__outcome from spacex where landing__outcome = 'Success (drone ship)' group by landing__outcome
```

- List the names of the boosters which have success in ground pad and have payload mass greater than 4000 but less than 6000

```
%sql select booster_version,landing__outcome,payload_mass__kg_ from spacex where payload_mass__kg_ >4000 and payload_mass__kg_<6000 and landing__outcome = 'Success (ground pad)'
```

- List the total number of successful and failure mission outcomes

```
%sql select count(*) as total, mission_outcome from spacex group by mission_outcome
```

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

```
%sql select payload_mass__kg_ , booster_version  from (select payload_mass__kg_, booster_version,(select max(payload_mass__kg_) as max_mass from spacex) from spacex) where payload_mass__kg_ = max_mass
```

- List the records which will display the month names, successful landing outcomes in ground pad ,booster versions, launch site for the months in year 2017

```
%sql select to_char(DATE, 'MONTH') as month, booster_version, launch_site, landing__outcome from spacex where landing__outcome = 'Success (ground pad)' and to_char(DATE, 'YYYY') = '2017'
```

- Rank the count of successful landing outcomes between the date 2010-06-04 and 2017-03-20 in descending order.

```
%sql select landing__outcome, count(*) as qnt from spacex where DATE >= '2010-06-04' and DATE <= '2017-03-20' and landing__outcome like '%Success%' GROUP BY landing__outcome ORDER BY count(*) desc
```

# Build an Interactive Map with Folium

https://github.com/PetruscaF/capstoneIBM/blob/main/lab_jupyter_launch_site_location.ipynb

- To find some geographical patterns and answer analytic questions about launch sites some objects were added on the folium map.

- Fisrt all launches sites were marked with it's name and with a circle using site's latitude and longitude coordinates.

- Afterward were inserted markers clusters to sum and illustrate If a launch was successful (green marker) or if a launch was failed (red marker)

- Finally, were calculated distancies between a selected launch site (CCAFS SLC) and a railway, highway, coastline and neatest city using polylines and showing the distance in km

# Build a Dashboard with Plotly Dash

https://github.com/PetruscaF/capstoneIBM/blob/main/spacex_dash_app.py

- A dashboard was created using a HTML application, in it you have 2 main graphics representations, one as pie chart that shows the success launches by site and the other a scatter plot with the correlation between payload and success for all sites classified by booster version (colors).
- The pie chart has 5 different inputs made by a drop-down component, one for each site and one for `all sites`, this way its possible to know what is the most successful landing site and by how much, other possibilities is to investigate each site individually to know the percentage of success and fail of each site.
- The scatter plot has a range of inputs made by a slider component additionally to the inputs of drop-down component; the slider it's ranged by the payload mass from 0 to 10000kg. This range permits the investigation of which range has the better results, and the color makes easy to identify the correlation of success and the booster version. The drop-down permits an easy correction with the different sites.

# Predictive Analysis (Classification)

https://github.com/PetruscaF/capstoneIBM/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

- First step is to download the data, afterwards we define the y target varible (Class) and save as pandas series. Then we standardize the data in X then reassign it to the variable X using the transform method.
- Secondly we split the data as train and test data.
- The trird step is repeated for each model (KNN, Logistic regression, tree classifier, SVM) and is used to improve and implement the models; which consist in creating a \`model\` object then create a GridSearchCV object with parameter cv = 10, then fitting the object to find the best parameters to that model from the dictionary parameters of that model and applying it in the model.
- The next step is evaluating the results calculating the accuracy on the test data using the method score and then plotting the confusion matrix, to complement the confusion matrix results we can use the classification_report which gives the precision, recall, F1-score, accuracy and support numbers.
- Finally, we can use the evaluation metrics to find **the best performing classification model**, in this case it was the tree classifier.

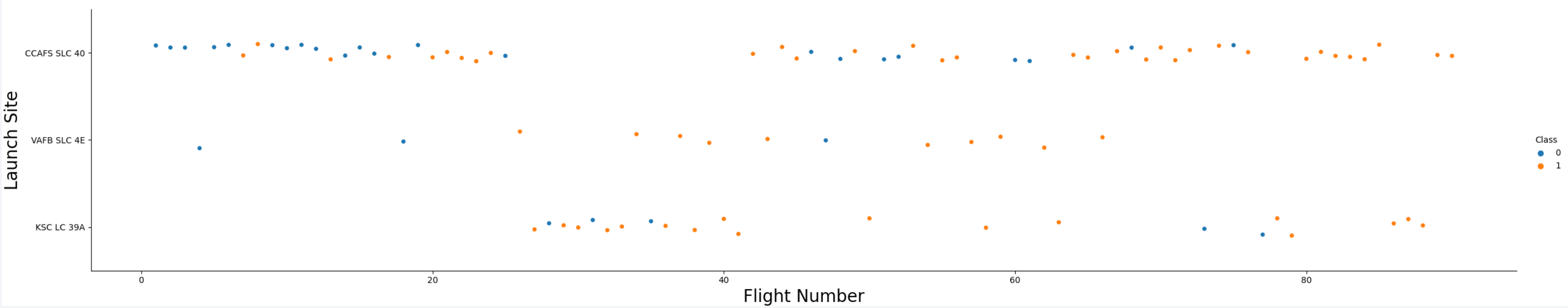
# Results

- The Flight Number vs. Launch Site and Launch Success Yearly Trend shows a positive tendency to improvement on technology and experience with the years culminating in more successful landings over the years.
- Payload vs. Launch Site and Correlation between Payload and Success from Dashboard shows that overall, the payload range with greater success are from 2k until 6k kg and the booster version that have the largest success rate is the FT.
- Success Rate vs. Orbit Type shows that 4 orbits that has perfect success rate: ES-L1, SSO, HEO and GEO.  Flight Number vs. Orbit Type shows that some orbits weren't used initially starting only after flight number 60. Payload vs. Orbit Type shows that most of the successful landing were related with payloads 2k to 6k more than the specific orbits, being GTO and ISS the most used.
- From the maps analysis it is easy to identify that the KSC LC site have higher success rates in13 launches and that all sites have important point in proximity, such as highways, railway, are close to the equator line and close to the ocean (coastlines).
- From the predictive analysis is clear that the best model for prediction is the tree classifier, with 94% of accuracy.

Section 2

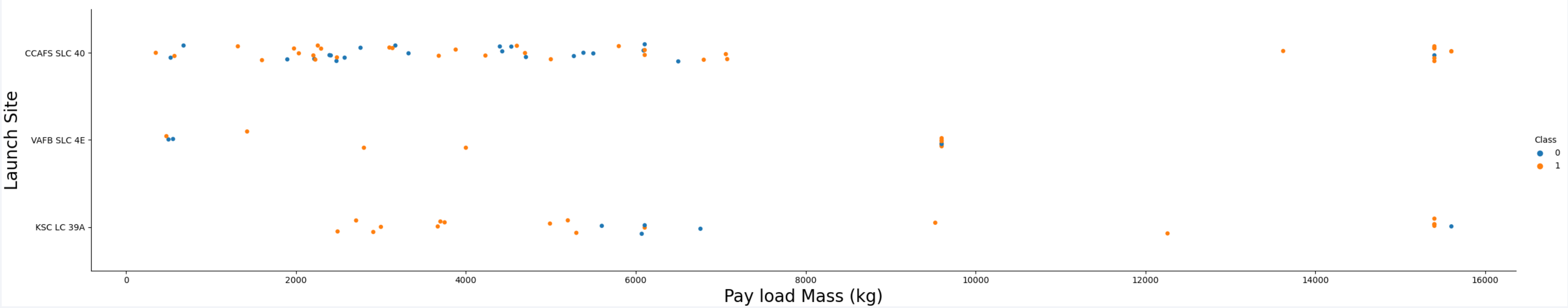
# Insights drawn from EDA

# Flight Number vs. Launch Site



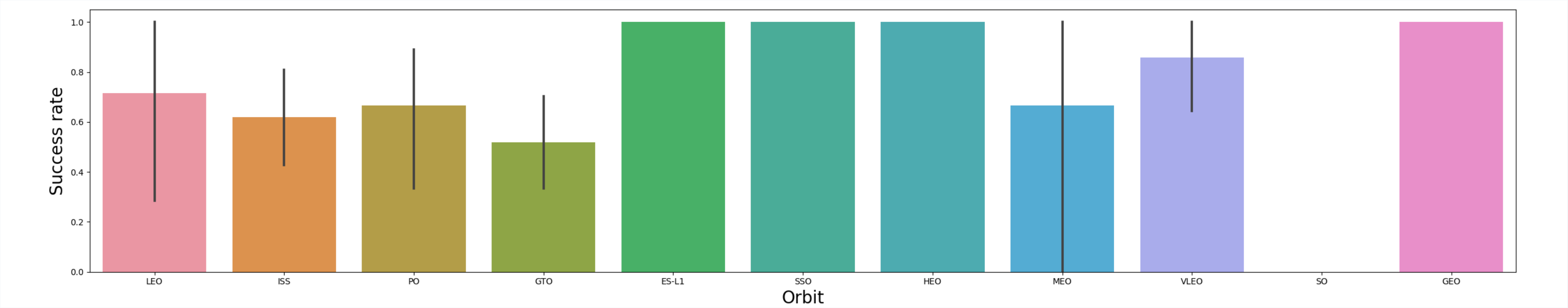

- We see that different launch sites have different success rates. CCAFS LC-40, has a success rate of 60 %, while KSC LC-39A and VAFB SLC 4E has a success rate of 77%.
- The graph shows more success rate on both VAFB and KSC launching sites, however the number of launchings are quite different from the CCAFS site, that fact alone can change de perspective of the results, for better understanding of success and launching sites, we need more data samples and better representation of each sites.

# Payload vs. Launch Site



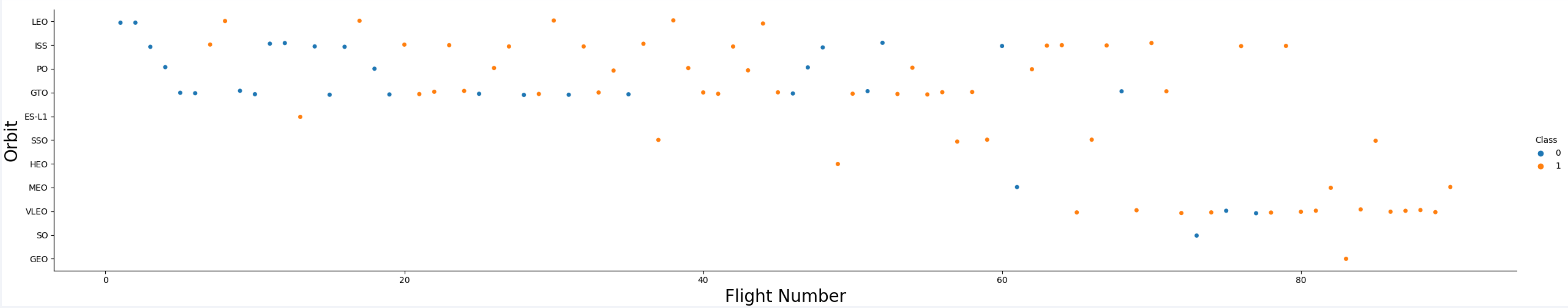

- The VAFB-SLC launch site didn't launch any rocket with heavy payload mass (greater than 10000).
- The payload above 8000 kg has a great success rate on both sites.
- The payload below 6000 kg has great success on KSC LC and VAFB sites, and mixed results on CCAFS SLC site.

# Success Rate vs. Orbit Type



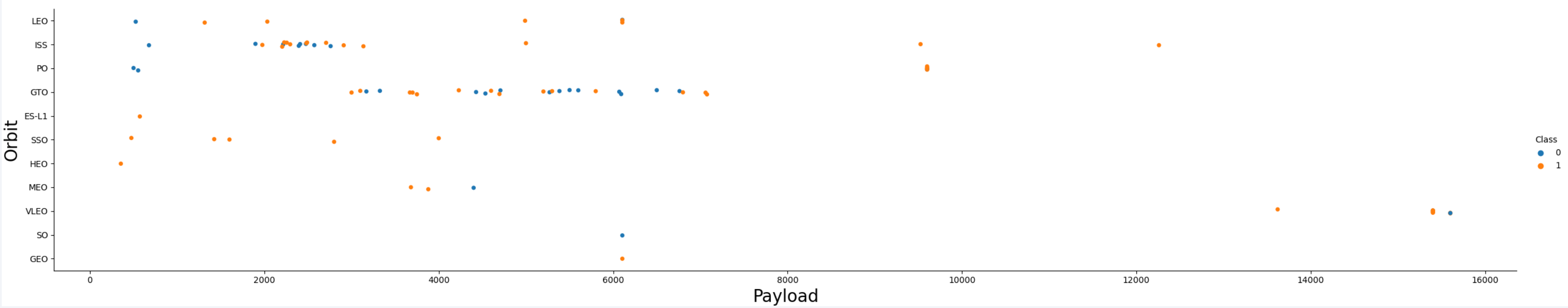

- There are 4 orbits that has perfect success rate: ES-L1, SSO, HEO and GEO.

# Flight Number vs. Orbit Type



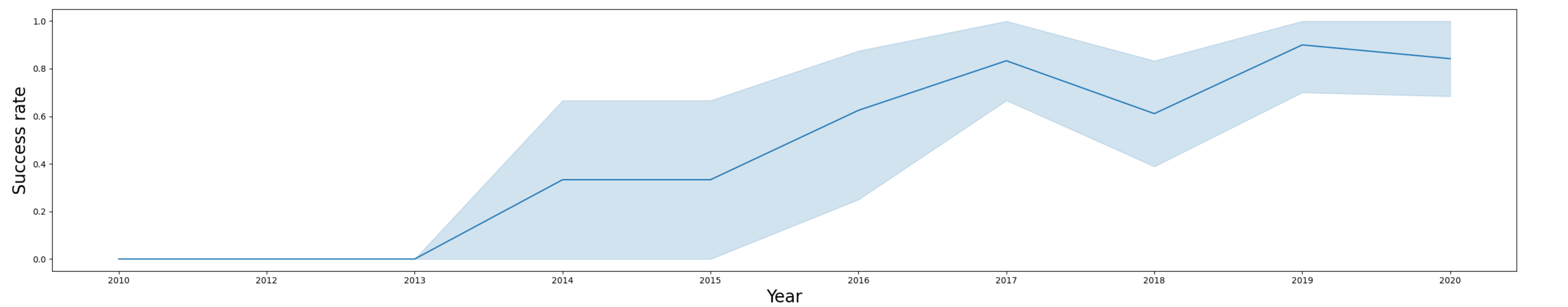

- LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.

# Payload vs. Orbit Type




- With heavy payloads the successful landing rate rises for Polar, LEO and ISS.
- However, for GTO we cannot distinguish this well as both positive and negative landing rate are almost equally present.

# Launch Success Yearly Trend




- Since 2013 the success rate kept increasing till 2020 with partial decrease on 2018. This indicates a positive tendency to improvement on technology and experience with the years. It's not clear the reason for decrease on 2018.

# All Launch Site Names

- Find the names of the unique launch sites
- %sql select distinct LAUNCH_SITE FROM SPACEX
- There are 4 unique registries on dataset, used the `distinct` argument to select only unique values

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

# Launch Site Names Begin with 'KSC'

- Find 5 records where launch sites' names start with `KSC`
- %sql select * from spacex where launch_site like 'KSC%'limit 5

| DATE | time__utc_ | booster_version | launch_site | payload | payload_mass__kg_ | orbit | customer | mission_outcome | landing_outcome |
|---|---|---|---|---|---|---|---|---|---|
| 2017-02-19 | 14:39:00 | F9 FT B1031.1 | KSC LC-39A | SpaceX CRS-10 | 2490 | LEO (ISS) | NASA (CRS) | Success | Success (ground pad) |
| 2017-03-16 | 06:00:00 | F9 FT B1030 | KSC LC-39A | EchoStar 23 | 5600 | GTO | EchoStar | Success | No attempt |
| 2017-03-30 | 22:27:00 | F9 FT B1021.2 | KSC LC-39A | SES-10 | 5300 | GTO | SES | Success | Success (drone ship) |
| 2017-05-01 | 11:15:00 | F9 FT B1032.1 | KSC LC-39A | NROL-76 | 5300 | LEO | NRO | Success | Success (ground pad) |
| 2017-05-15 | 23:21:00 | F9 FT B1034 | KSC LC-39A | Inmarsat-5 F4 | 6070 | GTO | Inmarsat | Success | No attempt |

- Used the `like` argument to narrow down the results, and limit argument to show specified number of rows

# Total Payload Mass

- Calculate the total payload carried by boosters from NASA
- %sql select sum(payload_mass__kg_) as total_payload ,customer from spacex where customer = 'NASA (CRS)' group by customer
- Used where clause to filter the data , grouped by customer, and used sum() to perform the calculation.

| total_payload | customer |
|---:|---:|
| 45596 | NASA (CRS) |

# Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1
- %sql select avg(payload_mass__kg_) as avg_payload ,booster_version from spacex where booster_version = 'F9 v1.1' group by booster_version
- Used avg() to perform the calculation and where clause to filter the data

| avg_payload | booster_version |
|---:|---:|
| 2928 | F9 v1.1 |

# First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on drone ship.
- %sql select min(DATE), landing__outcome from spacex where landing__outcome = 'Success (drone ship)' group by landing__outcome
- Used the min() to select the closest date, where clause to filter the data and grouped by landing outcome to single out the result.

| 1 | landing__outcome |
|---:|---:|
| 2016-04-08 | Success (drone ship) |

# 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
- %sql select booster_version,landing__outcome,payload_mass__kg_ from spacex where payload_mass__kg_ >4000 and payload_mass__kg_<6000 and landing__outcome = 'Success (ground pad)'
- Used 3 obligatory conditions (>4000kg ,<6000kg, success (groundpad)) to filter the results with where clause

| booster_version | landing__outcome | payload_mass__kg_ |
|---:|---:|---:|
| F9 FT B1032.1 | Success (ground pad) | 5300 |
| F9 B4 B1040.1 | Success (ground pad) | 4990 |
| F9 B4 B1043.1 | Success (ground pad) | 5000 |

# Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes
- %sql select count(*) as total, mission_outcome from spacex group by mission_outcome
- Used count() to perform the calculation and * to select all data then grouped by `outcome` to classify the results

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

# Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass
- %sql select payload_mass__kg_ , booster_version  from (select payload_mass__kg_, booster_version,(select max(payload_mass__kg_) as max_mass from spacex) from spacex) where payload_mass__kg_ = max_mass
- Used a subquery to single out the maximum mass from the dataset using another selection (max()), then used this argument to filter the original dataset and list the booster version

| payload_mass__kg_ | booster_version |
|---|---|
| 15600 | 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 |

# 2015 Launch Records

- List the records which will display the month names, succesful landing_outcomes in ground pad ,booster versions, launch_site for the months in year 2017
- %sql select to_char(DATE, 'MONTH') as month, booster_version, launch_site, landing__outcome from spacex where landing__outcome = 'Success (ground pad)' and to_char(DATE, 'YYYY') = '2017’
- Parsed the data first to filter the year the used the success (ground pad) as another filter and finally parsed the data to show the results as months names

| MONTH | booster_version | launch_site | landing__outcome |
|---|---|---|---|
| FEBRUARY | F9 FT B1031.1 | KSC LC-39A | Success (ground pad) |
| MAY | F9 FT B1032.1 | KSC LC-39A | Success (ground pad) |
| JUNE | F9 FT B1035.1 | KSC LC-39A | Success (ground pad) |
| AUGUST | F9 B4 B1039.1 | KSC LC-39A | Success (ground pad) |
| SEPTEMBER | F9 B4 B1040.1 | KSC LC-39A | Success (ground pad) |
| DECEMBER | F9 FT B1035.2 | CCAFS SLC-40 | Success (ground pad) |

# 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
- %sql select landing__outcome, count(*) as qnt from spacex where DATE >= '2010-06-04' and DATE <= '2017-03-20' and landing__outcome like '%Success%' GROUP BY landing__outcome ORDER BY count(*) desc
- Used 3 conditions to filter the data, based on data range and them using $ argument to select any combination of the string that contains `success` in the `middle`  then grouped by outcome e ordinated the results from the select count() from bigger to smaller

| landing__outcome | qnt |
|---|---|
| Success (drone ship) | 5 |
| Success (ground pad) | 3 |

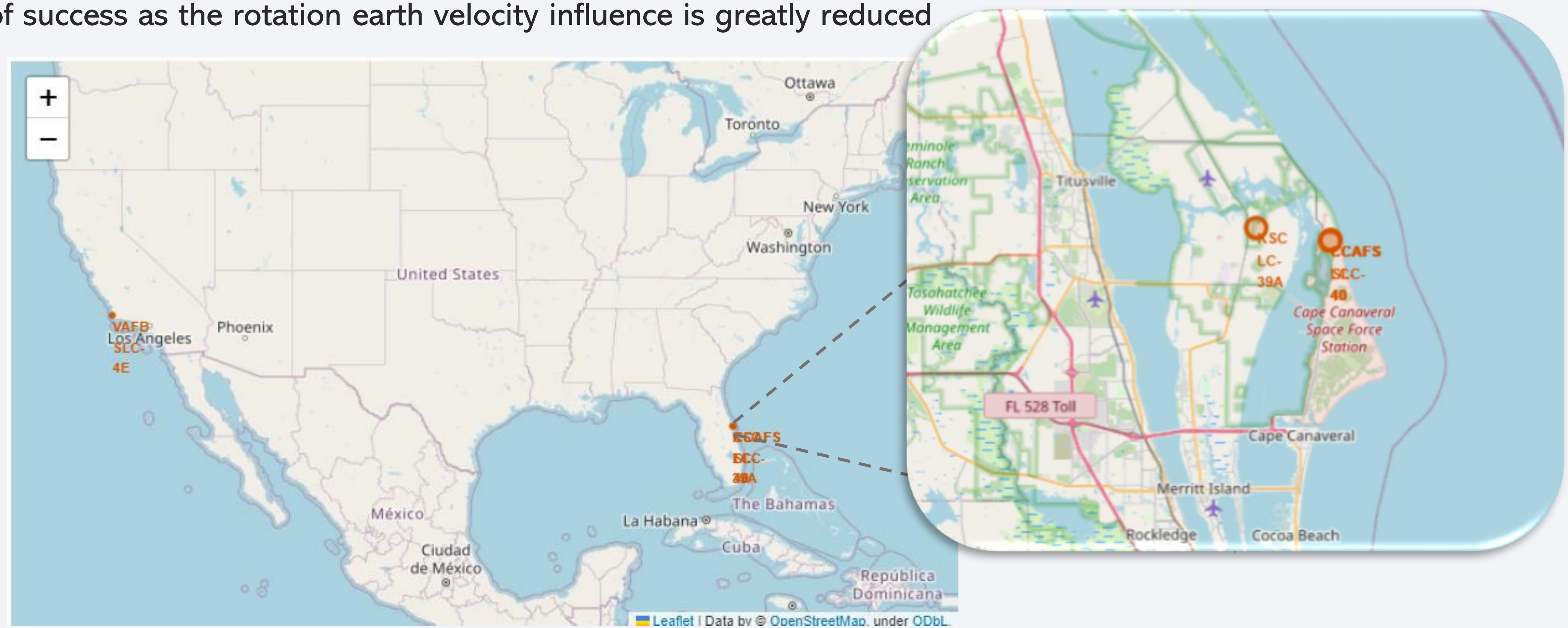
Section 3

# Launch Sites Proximities Analysis

# Launch site's locations on Folium Map

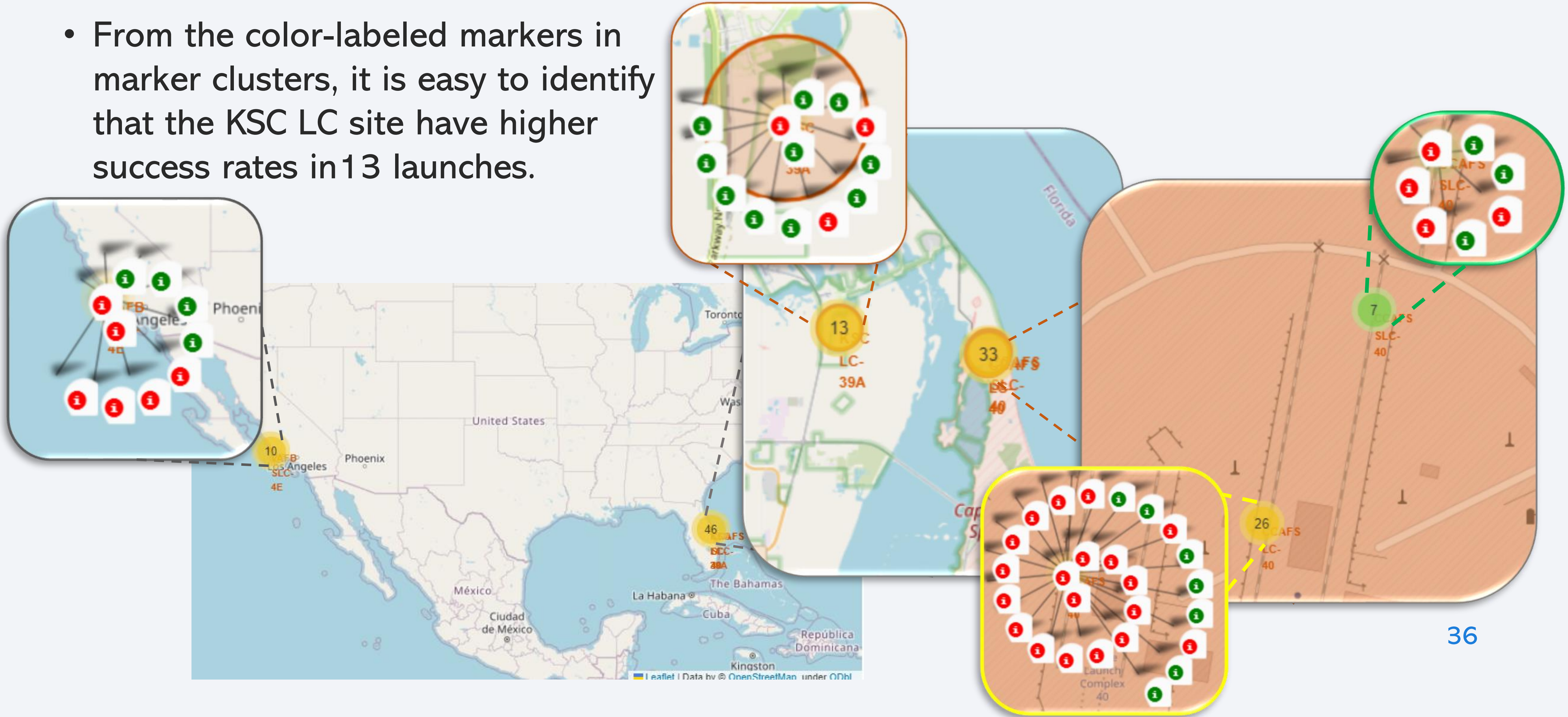
- From the map representation we can see that the sites KSC and CCAFS are in close proximity and on the coastline whereas the site VAFB it's on the opposite coastline of the United States. Locations by the sea makes errors in landings or launchings less likely to cause damage to urban areas, as the ocean occupies most of the surrounding areas.

- All sites are relatively close to the equator line. It is known that launches closer to the equator line have better chances of success as the rotation earth velocity influence is greatly reduced



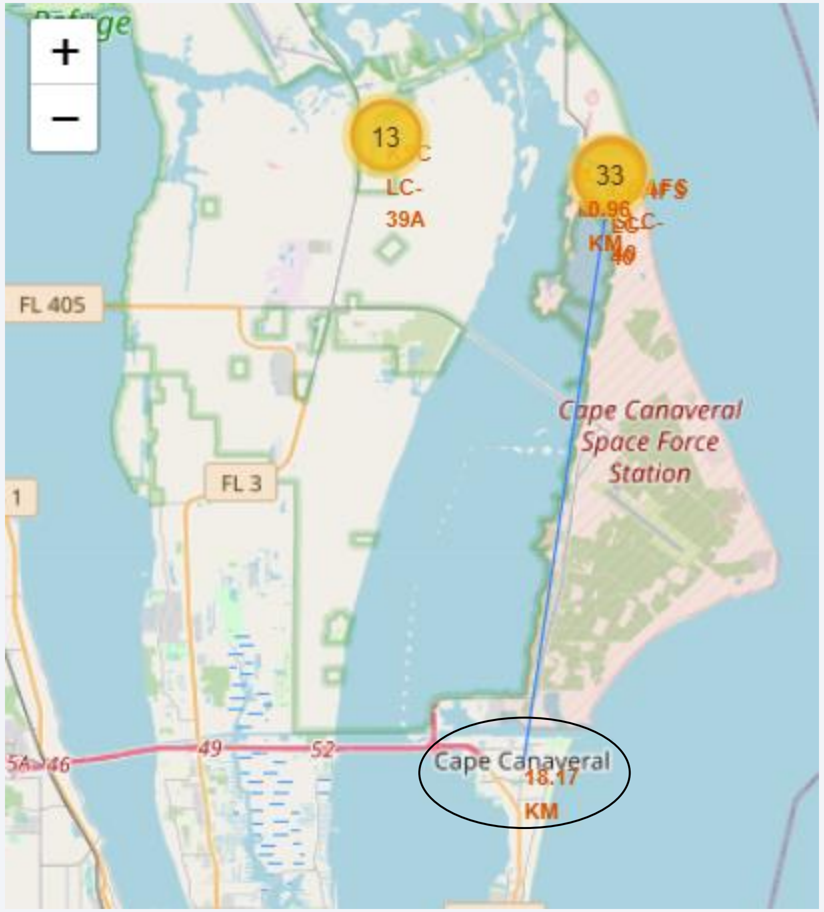
# Success/failed outcomes per Launch site

- From the color-labeled markers in marker clusters, it is easy to identify that the KSC LC site have higher success rates in13 launches.

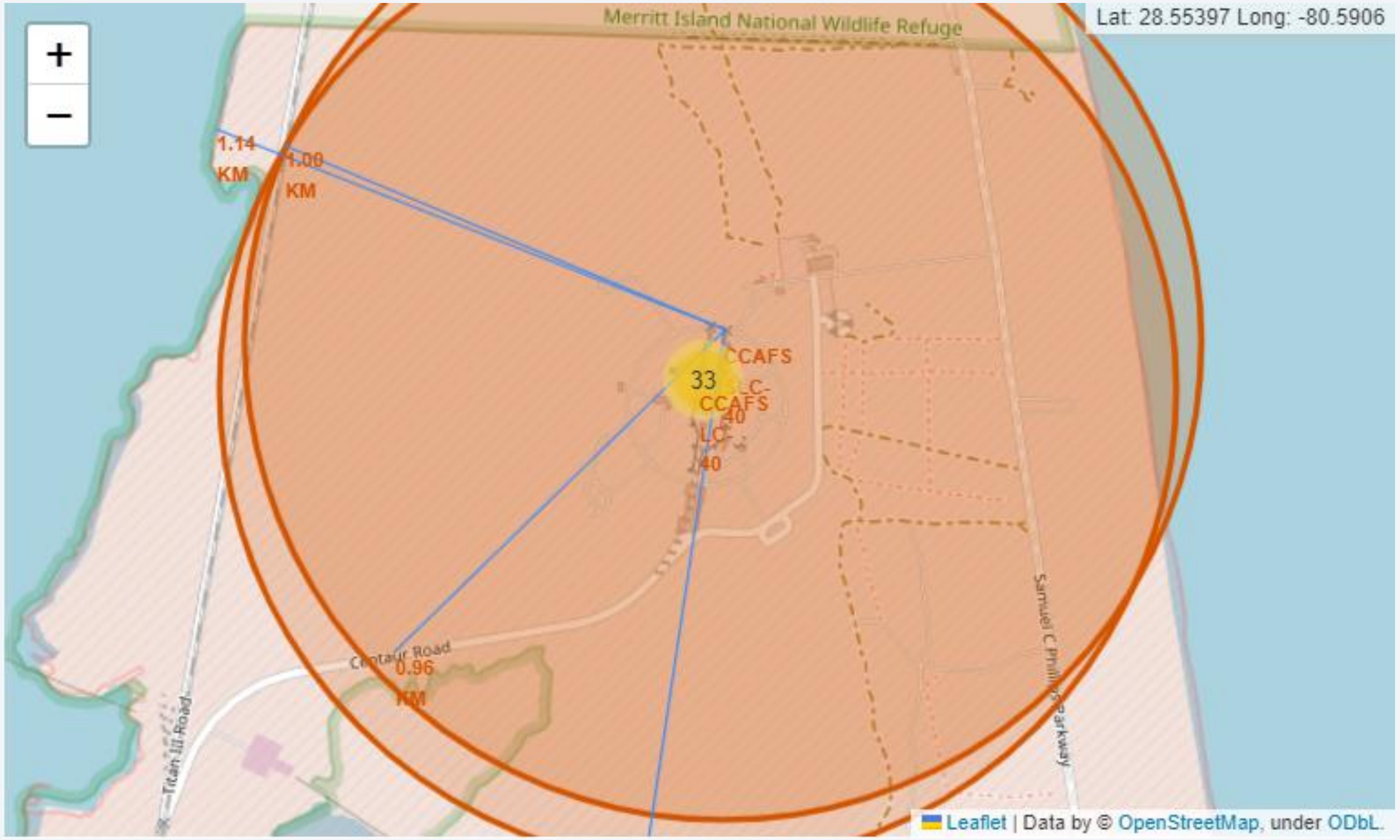


# Distance from CCAFS SLC to proximity's elements

- The closest city (Cape Canaveral) from site is at least 18 km away from the base, probably for security reasons in case of unsuccessful landing or launching.




- The coastline is 1.14km from the site also for security reasons making the errors in landings or launchings less likely to cause damage to urban areas, as the ocean occupies most of the surrounding areas
- There are a highway (Centaur Road) in 0.96km and a railway (Titan III Road) in 1km from the site, they are important to supply and to transport materials and rocket's components.

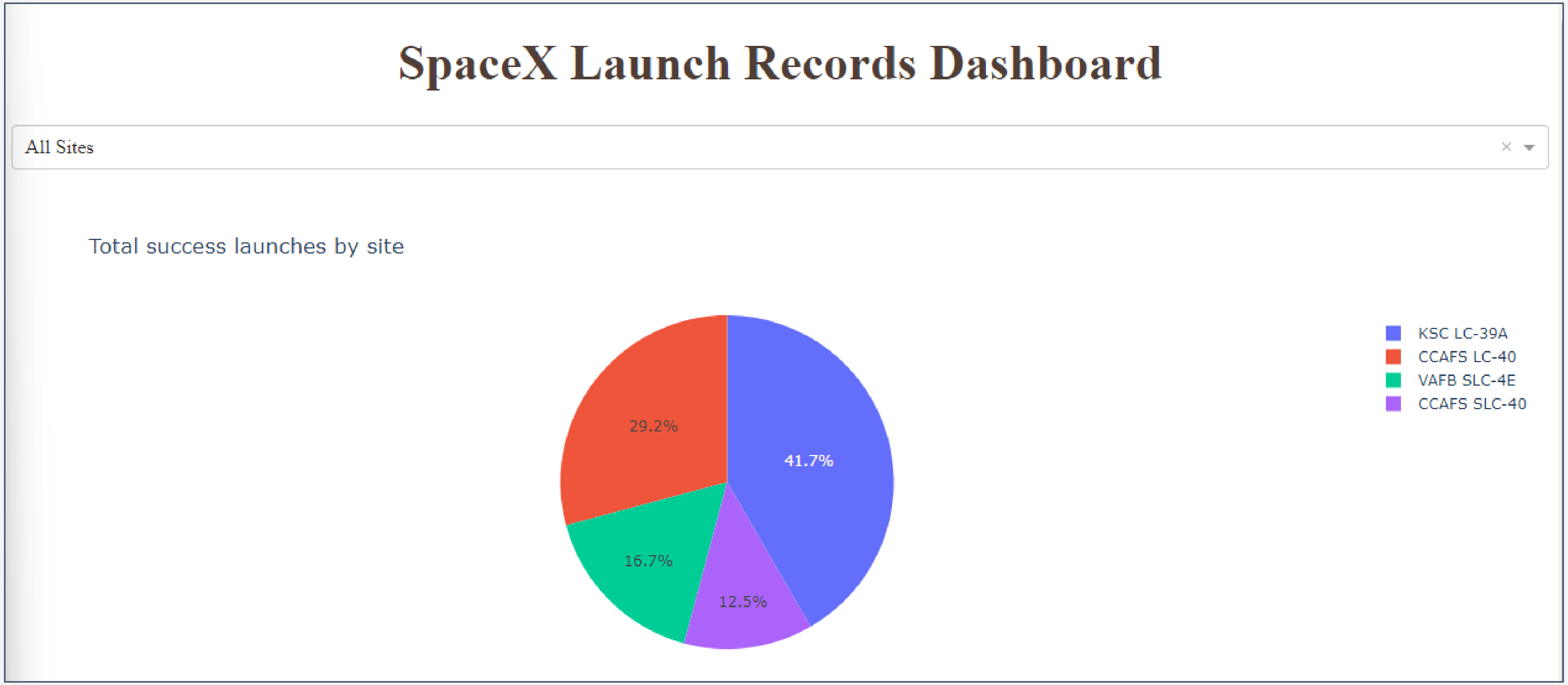


Section 4

# Build a Dashboard with Plotly Dash

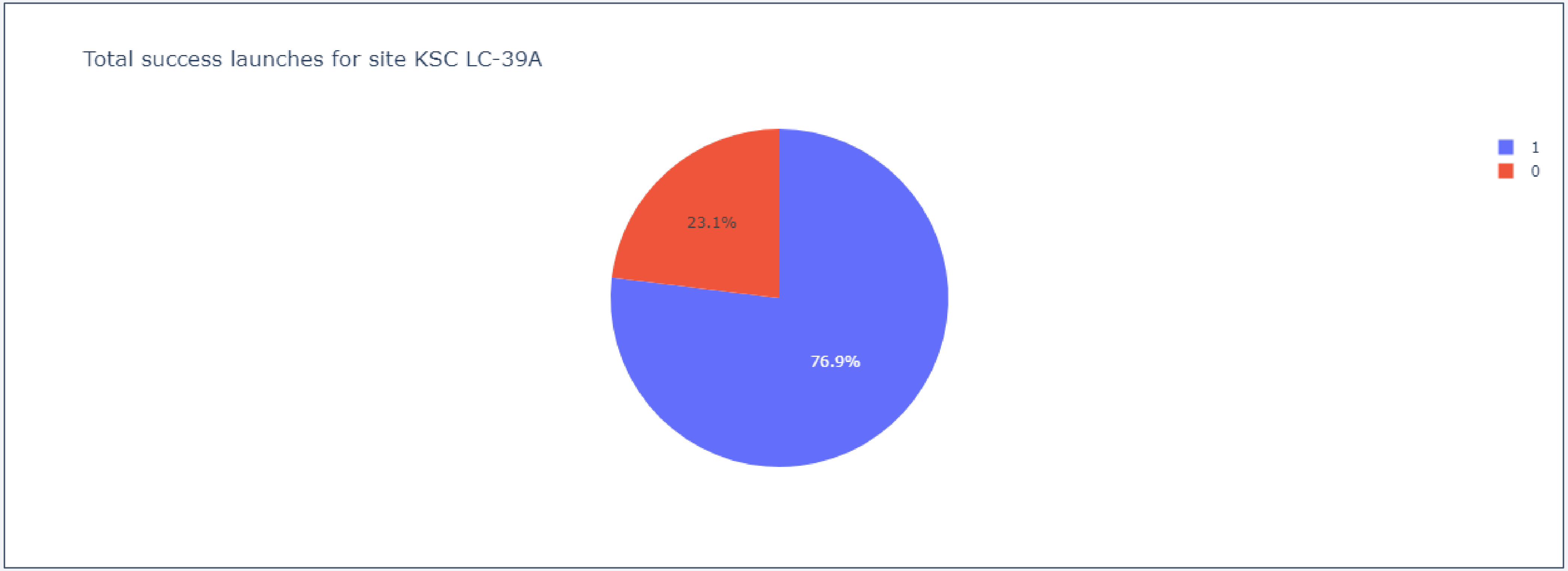
# Total success launches by site

- This graphic shows that the most successful site is the KSC LC 39-A with 41.7%



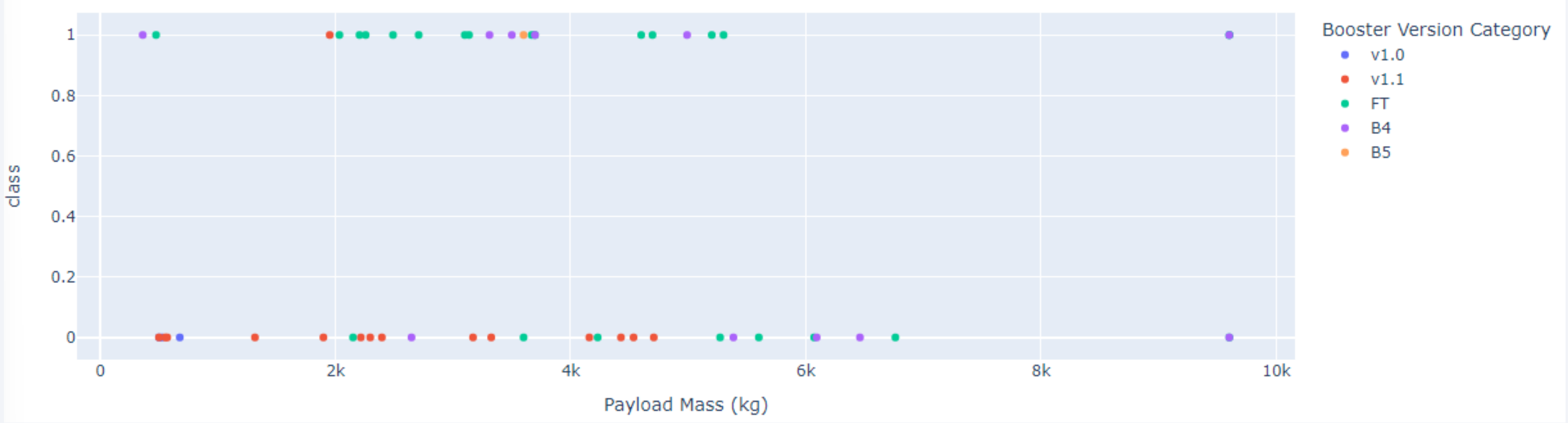
# Total success launches for site KSC LC-39A

- The launch site with highest launch success ratio is the KSC, with 76.9% successful landings and 23.1% of failed ones.

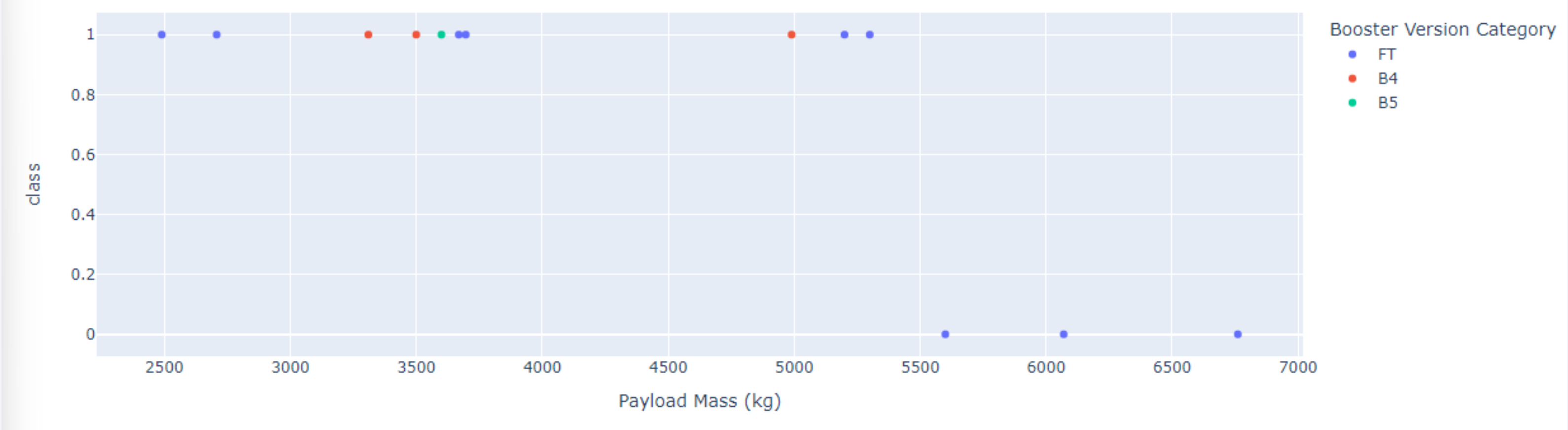


# Correlation between Payload and Success

Correlation between Payload and Success for all sites




Correlation between Payload and Success for KSC LC-39A




- Overall, the payload range with greater success are from 2k until 6k kg and the booster version that have the largest success rate is the FT (green). We can also see that the v1.1 has the worst performance.

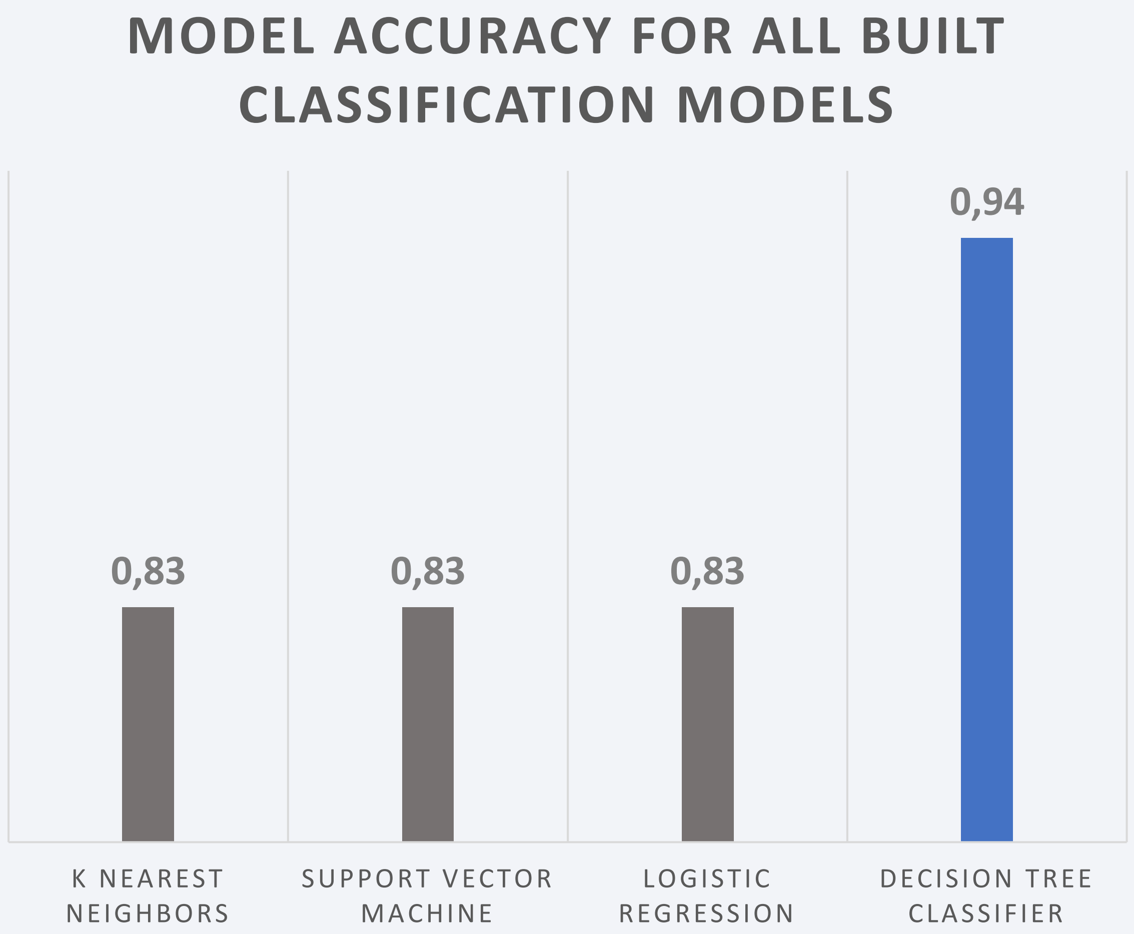
- Narrowing down the results by selecting the best site (KSC) and best payload range (2k~6k) we can confirm the performance of the booster FT (blue) and see the addition of booster B4 (red), this way we can filter even more, then the best payload mass for this site is from 2500 to 5300 kg, as heavier mass ends in failure.

Section 5

# Predictive Analysis (Classification)

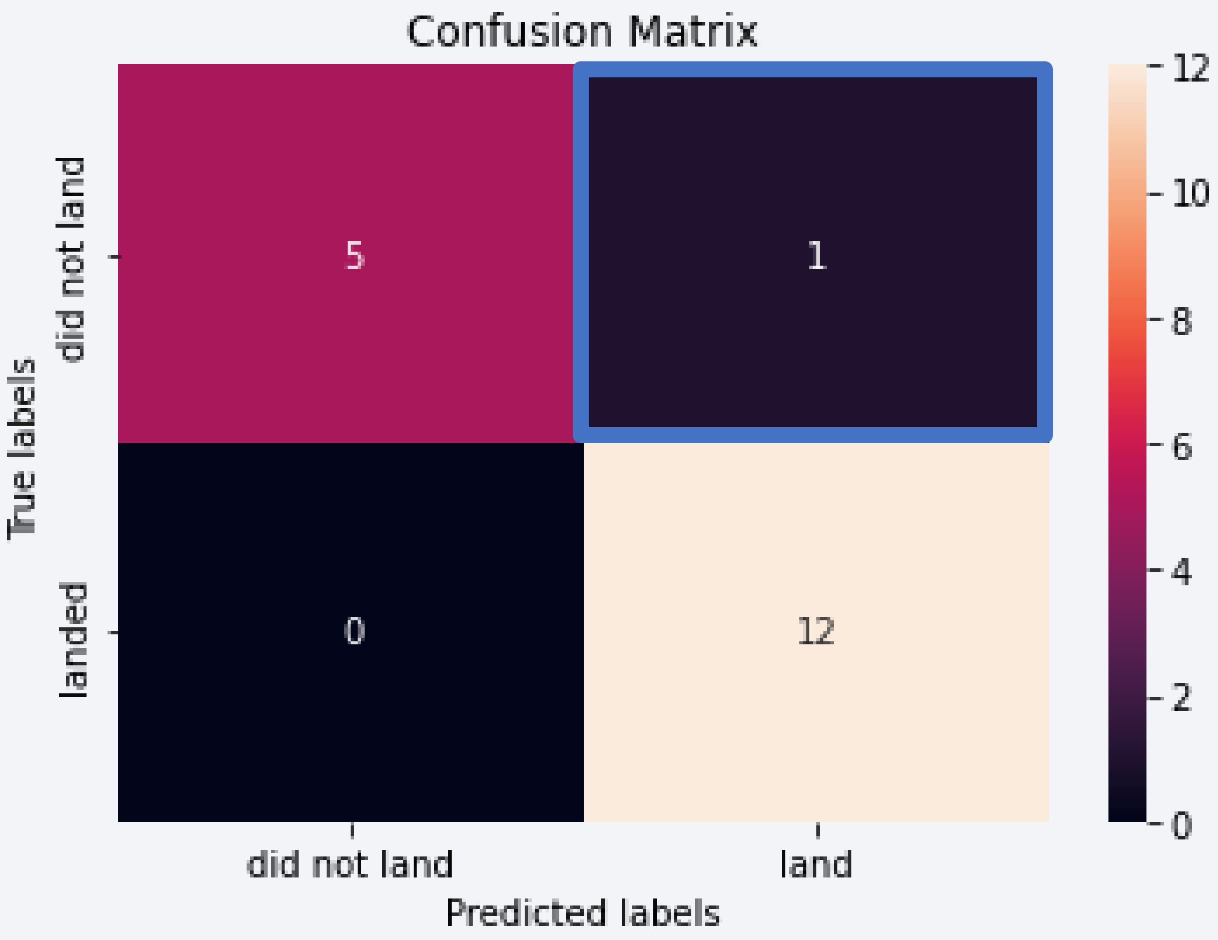
# Classification Accuracy

- The overall accuracy of all evaluated models are 0.83 except the **decision tree classifier** that had the best accuracy of 0.94 which means it **is the best model used to predict landings**.



# Confusion Matrix

- The confusion matrix of the tree classified shows only **one error of false positive** landing in the prediction as show by the blue square and predicted correctly all successfully landed launches.



# Conclusions

- The **decision tree classifier** had the highest accuracy of 0.94 which means it **is the best model used to predict landings**.
- The confusion matrix of the tree classifier had only one error of false positive.
- The models: k nearest neighbors, support vector machine and logistic regression had the same results were the confusion matrix shows that all of them had problems with false positives (3 wrong predictions of successfully landing).
- The closeness of accuracy on this models are related with the small dataset size to train and predict with larger datasets the results may vary a lot more.

# Thank you!