



IBM Developer  
SKILLS NETWORK

# Winning Space Race with Data Science

Fanilo Nomen'Aina Rafanomezantsoa  
May 5, 2022



# Outline

---

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

# Executive Summary

---

- **Summary of methodologies**
  - Data collection
  - Data wrangling
  - EDA with data visualization
  - EDA with SQL
  - Building an interactive map with Folium
  - Building a Dashboard with Plotly Dash
  - Predictive analysis (Classification)
- **Summary of all results**
  - Exploratory data analysis results
  - Interactive analytics demo in screenshots
  - Predictive analysis results

# Introduction

---

- Project background and context

SpaceX advertises Falcon 9 rocket launches on its website, with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage.

Determining whether the first stage will land or not will allow us to estimate the cost of a launch

- Problems you want to find answers

- Find the correlation between each rocket variables (such as landing site location, payload mass, booster version ...) and the landing outcome, without any rocket science knowledges



Section 1

# Methodology

# Methodology

---

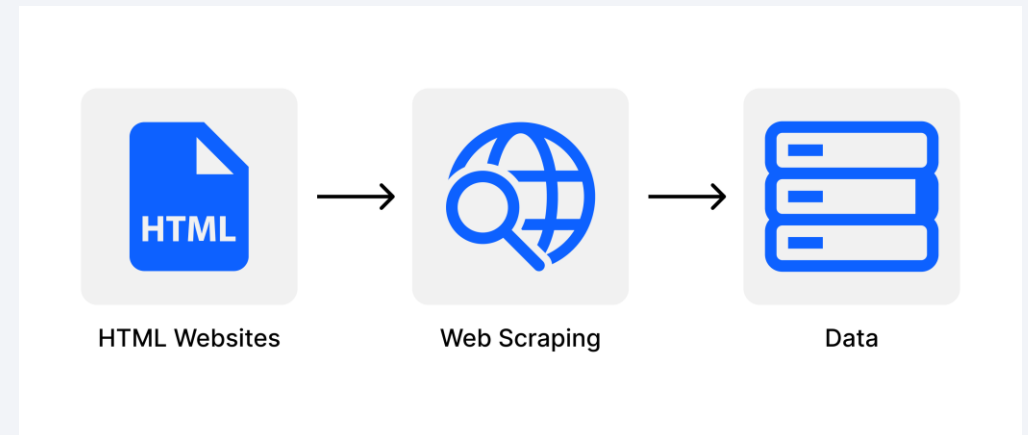
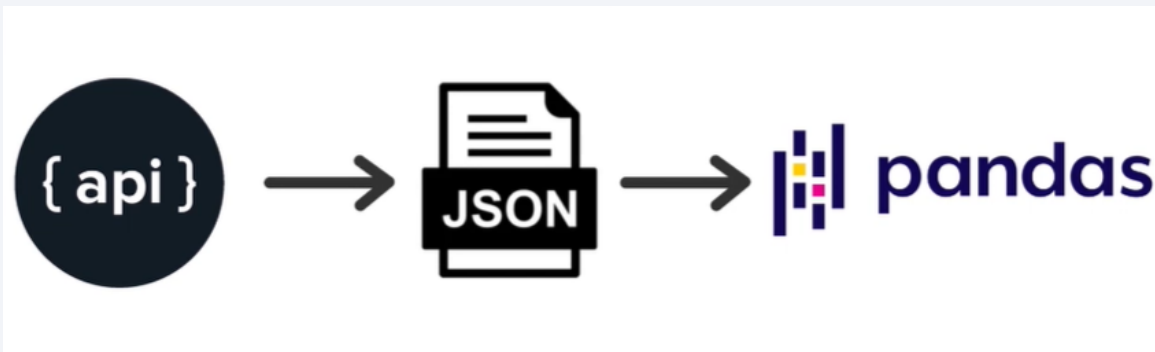
## Executive Summary

- Data collection methodology:
  - Calling a Rest Endpoint and scraping web page
- Perform data wrangling
  - Convert outcomes into binary output which categorizes the success or failure of a booster landing
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Select the models (SVM, KNN, Classification Trees and Logistic Regression), obtain the optimal Hyperparameter using grid search method, then evaluate the accuracies.

# Data Collection

---

- The data used comes from 2 different sources, the first comes from a REST Api ([SpaceX Api](#)), the second comes from [Falcon 9 and Falcon Heavy Launches Records](#) web data from [Wikipedia](#), obtained by web scraping.



# Data Collection – SpaceX API

---


- Request data from SpaceX API
- Format the response data into JSON
- Load the data into a data frame
- Export the data to CSV

[Github source](#)

Requesting data from SpaceX API by calling REST endpoint with python's library **requests**



Normalizing the response into JSON



Load the JSON data as **pandas** data frame and persist it locally as CSV file




# Data Collection - Scraping

---

- Request the Falcon9 Launch from wiki page
- Extract all column/variable names from the HTML table header
- Create a data frame by parsing the launch HTML tables
- Persist the data frame as CSV file

Request the Falcon 9 launch from wiki page using **requests**



Using **BeautifulSoup** library, extract all tables contents



Load the obtained data as pandas data frame



Persist the result to CSV format

# Data Wrangling

---

- In our data we clearly distinct that the attribute **outcome** provides us the result of the booster landing:
  - **True Ocean**: the mission result has successfully landed in a specific area of the ocean
  - **False Ocean**: the mission result has not successfully landed in a specific area of the ocean
  - **True RTLS**: the mission result successfully landed on the ground pad
  - **False RTLS**: the mission result has not successfully landed on the ground pad
  - **True ASDS**: the mission result has successfully landed on the drone ship
  - **False ASDS**: the mission result has not landed on the drone ship
- To facilitate learning of the models, it seems reasonable to categorize the landing results in a binary way (2 classes):
  - 0 will be assigned to launches with failed landing
  - 1 to all successes

# Data Wrangling

---

- Calculating the number of launches at each site
- Calculating the number and occurrence of each orbit
- Creating a landing outcome label from Outcome column
- Calculating the success rate for every landing in dataset
- Exporting dataset to a CSV

[Github source](#)

Determine the number of launches on each site

Determine the number and occurrence of each **orbit** in the column

Using the **Outcome**, create a list where the element is zero if the corresponding row in Outcome is in the set bad outcome

Compute the mean of the column **Class** to determine the average landing success rate

Export the dataset to CSV format

# EDA with Data Visualization

---

- The **Scatter chart** enlightens The relationship between variables and is used in:
  - *Flight Number vs. Launch Site*
  - *Payload vs. Launch Site*
  - *Flight Number vs. Orbit Type*
  - *Payload vs. Orbit Type*
- **Bar chart** display the magnitude of the value taken by a categorical variables and is used in:
  - *Orbit Type vs. Success Rate*
- **Line chart**, like scatter chart, enlightens The relationship between variables and is used in:
  - *Year vs. Success Rate*

# EDA with SQL

---

- Querying the dataset from a Db2 database, and with SQL queries bring responses to some key question:
  - Displaying the names of the unique launch sites in the space mission
  - Displaying 5 records where launch sites begin with the string 'CCA'
  - Displaying the total payload mass carried by boosters launched by NASA (CRS)
  - Displaying average payload mass carried by booster version F9 v1.1
  - Listing the date when the first successful landing outcome in ground pad was achieved
  - Listing the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000



# EDA with SQL

---

- Listing the total number of successful and failure mission outcomes
- Listing the names of the booster versions which have carried the maximum payload mass
- Listing the failed landing outcomes in drone ship, their booster versions, and launch site names for in year 2015
- Ranking the count of landing outcomes (such as Failure (drone ship) or Success (groundpad)) between the date 2010-06-04 and 2017-03-20 in descending order

[Github source](#)

# Build an Interactive Map with Folium

---

- Objects created and added to a folium map:
  - Markers that show all launch sites on a map
  - Markers that show the success/failed launches for each site on the map
  - Lines that show the distances between a launch site to its proximities
- These objects tell us about the common properties (logistics and safeness) of the launch sites:
  - Are launch sites in close proximity to railways? Yes
  - Are launch sites in close proximity to highways? Yes
  - Are launch sites in close proximity to coastline? Yes
  - Do launch sites keep certain distance away from cities? Yes

[Github souce](#)

# Build a Dashboard with Plotly Dash

---

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

- Pie chart
  - For showing total success launches by sites
  - This chart can be selected to indicate a successful landing distribution across all launch sites or to indicate the success rate of individual launch sites.
- Scatter chart
  - For showing the relationship between Outcomes and Payload mass(Kg) by different boosters
  - Has 2 inputs: All sites/individual site & Payload mass on a slider between 0 and 10000 kg
  - This chart helps determine how success depends on the launch point, payload mass and booster version categories.

[Github source](#)

# Predictive Analysis (Classification)

1. Select the features and the target for the models
2. Choose the models
3. Obtain the optimal hyperparameter for each models by using grid search method
4. Train each models
5. Compare their performances
6. Repeat the process and make adjustment if necessary



[Github source](#)

# Results

## SpaceX Launch Records Dashboard

All Sites

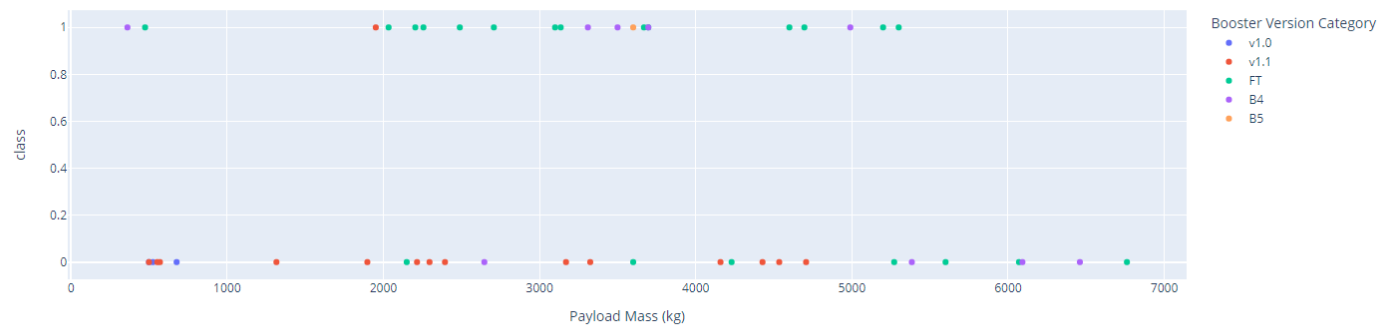
Total Success Launches By Site



Payload range (Kg):



Correlation between Payload and Success for all Sites





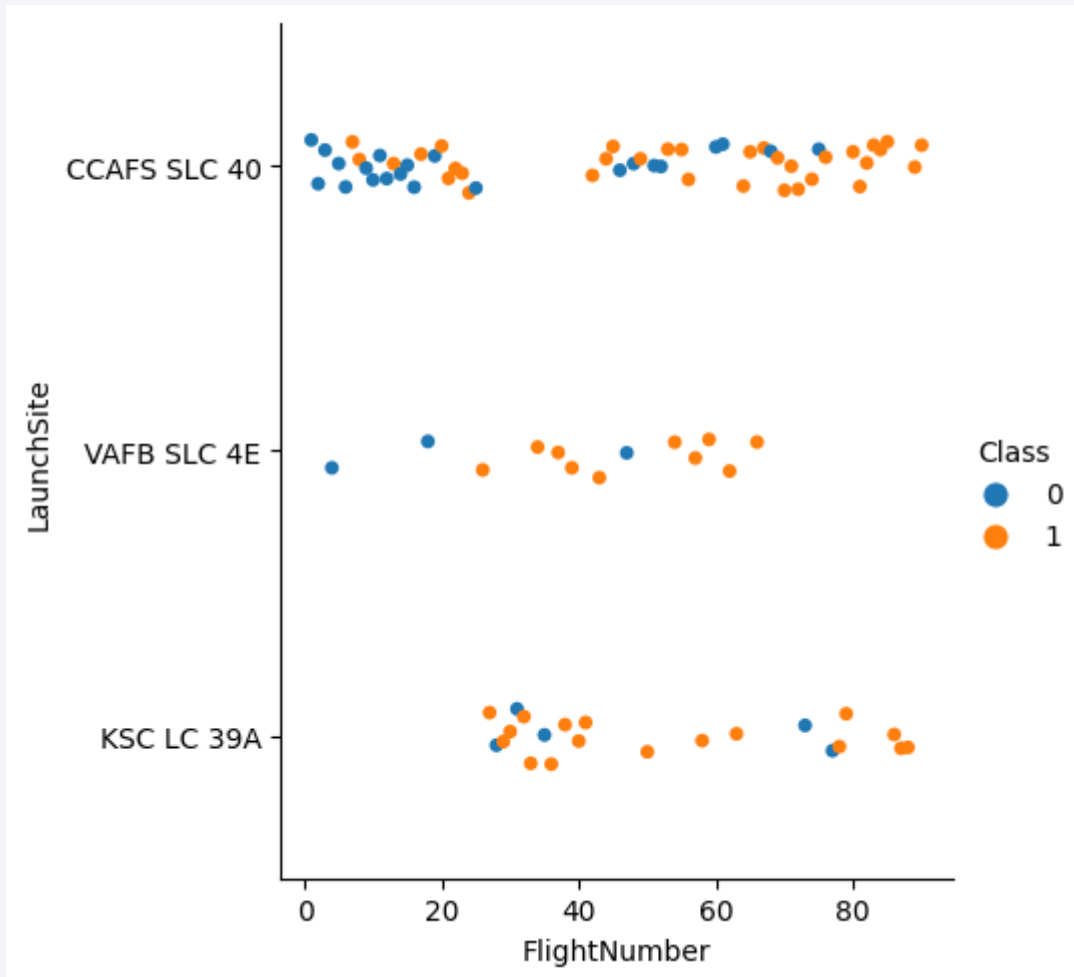
The background of the slide is an abstract composition. It features a dark blue base color. Overlaid on this are numerous diagonal streaks in shades of red and cyan. A faint, light blue grid pattern is also visible, particularly in the lower half of the image. The overall effect is dynamic and technological.

Section 2

# Insights drawn from EDA

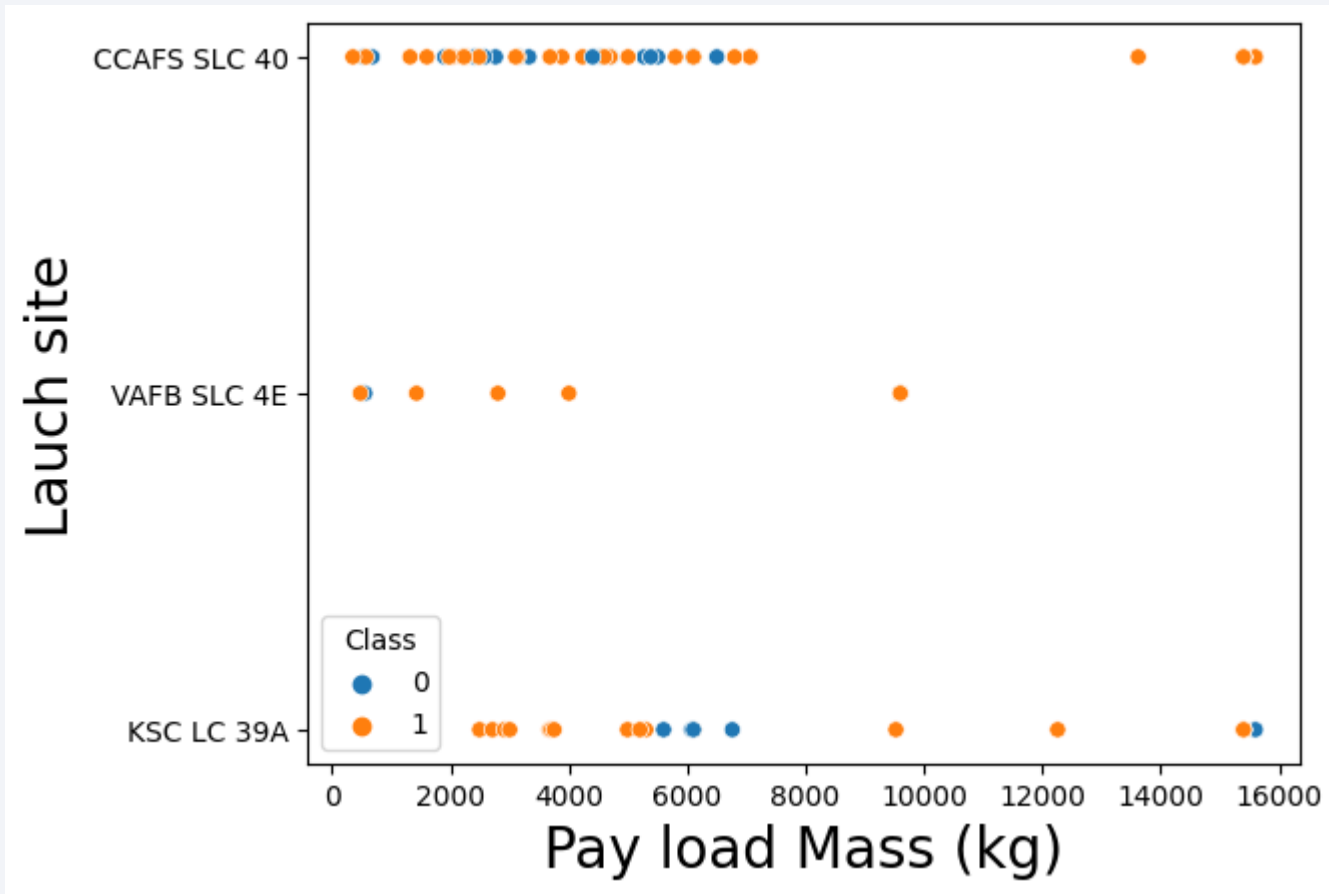


# Flight Number vs. Launch Site



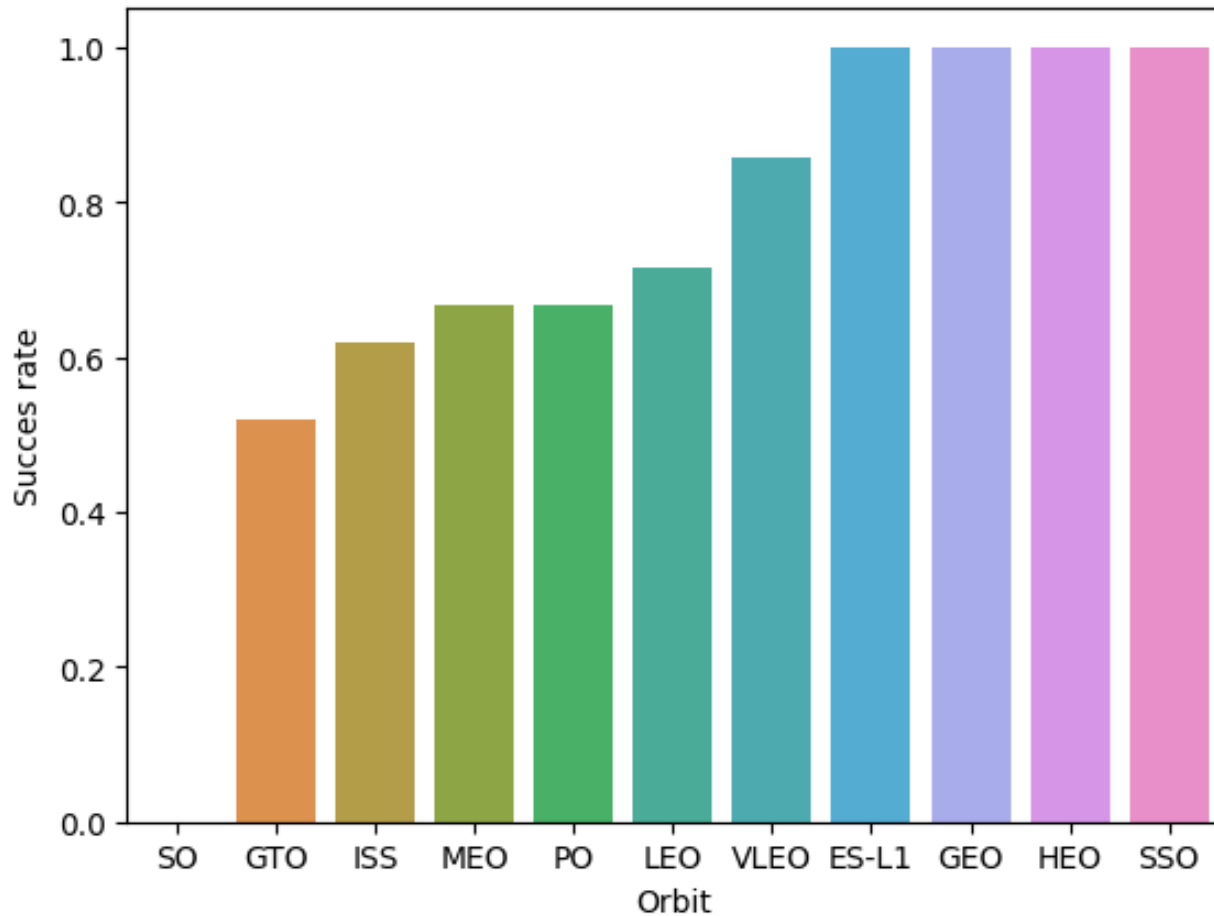
- It is clearly perceived that the success rate of landings increases considerably, as the number of launches increases. This is explained by the improvement over time of the boosters used

# Payload vs. Launch Site



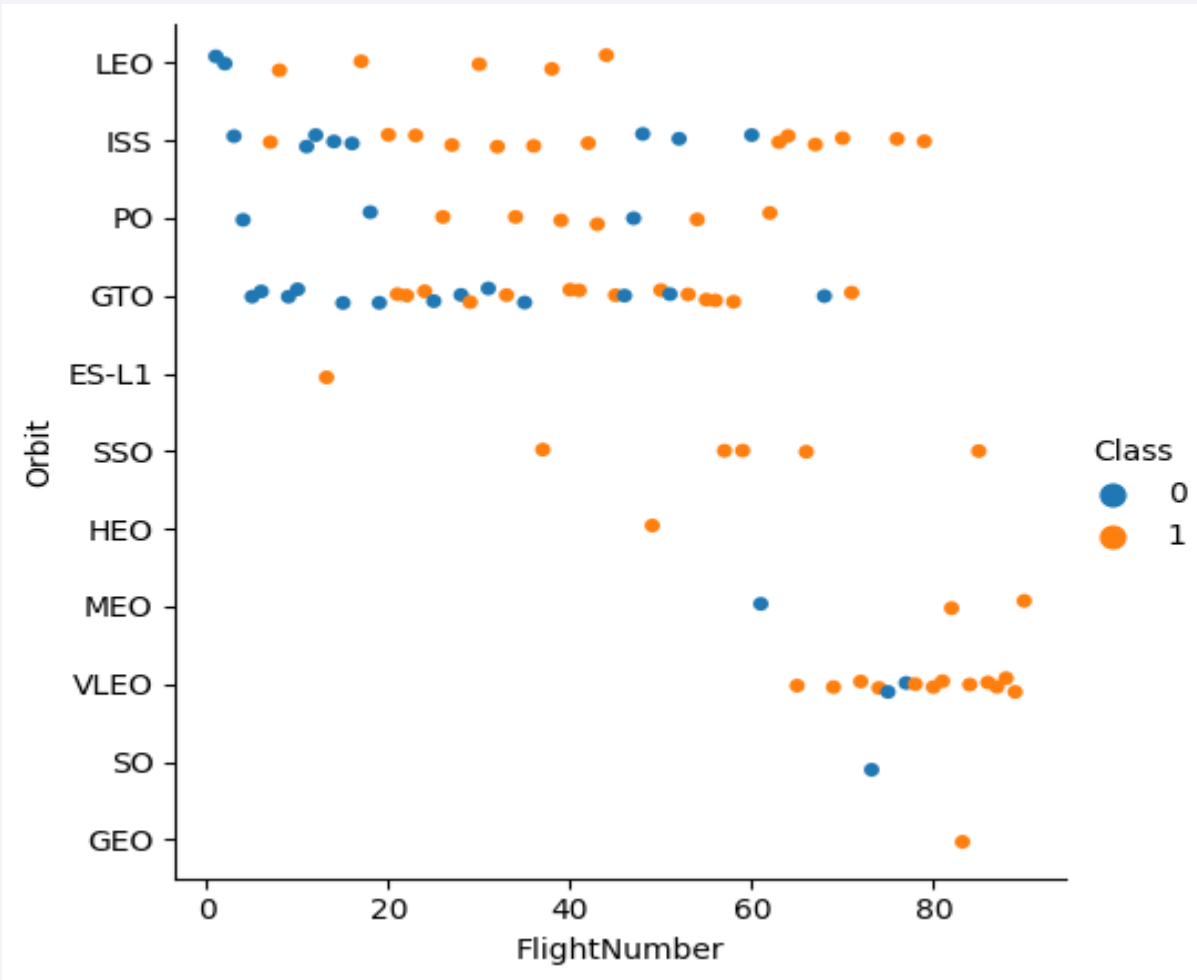
- No clear pattern can be found between successful launch site and Pay Load Mass

# Success Rate vs. Orbit Type



- the disparity in the success rate of the different orbits is explained by the inequality in the number of launches carried out for each respective orbit. That is, more tests were performed in low orbit (near Earth). In addition, the 100% success rate of some orbits is explained by the rarity of the launches and their launch years, where the boosters have already performed very well.

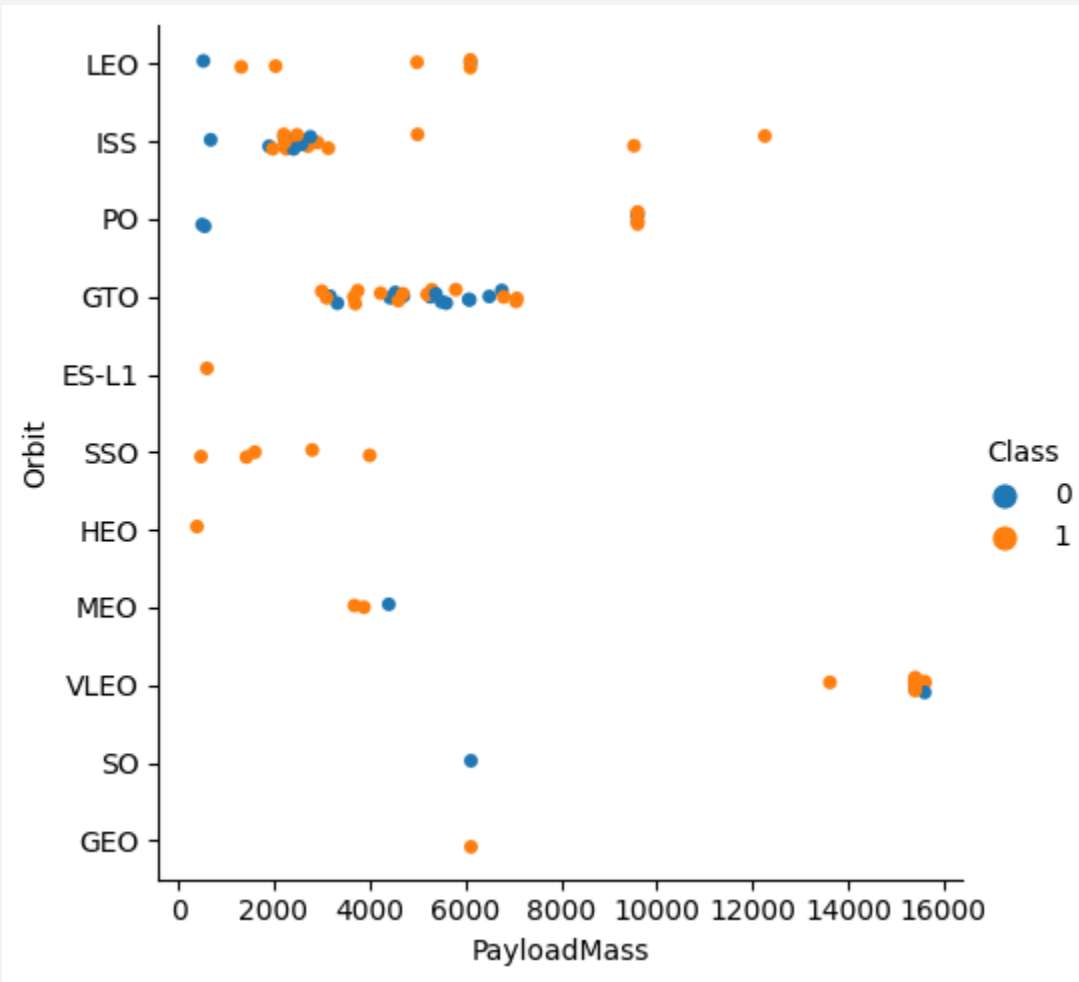
# Flight Number vs. Orbit Type



- It can be observed that the launch failure rate decreases with the increase in flights, however for some orbits like GTO, it is difficult to determine the success or failure of the landing based on the number of flights alone .



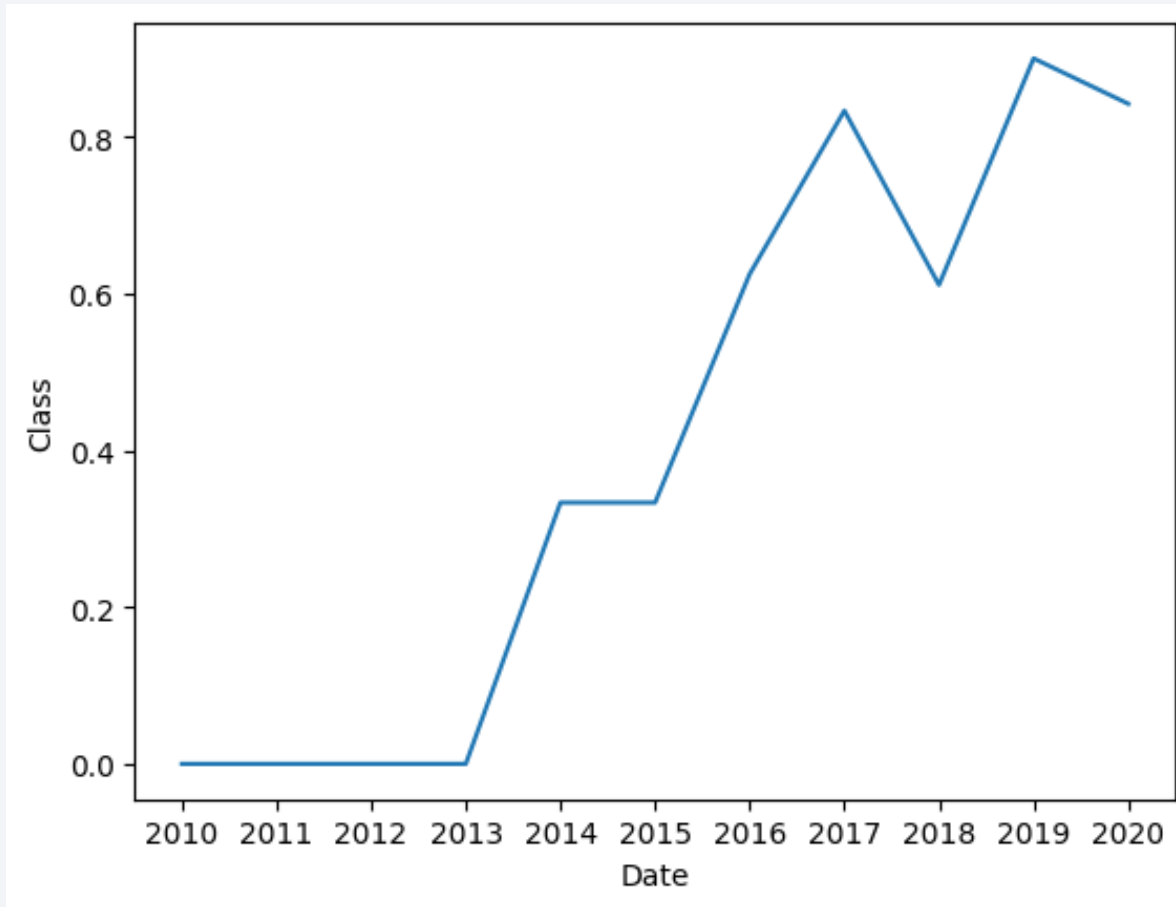
# Payload vs. Orbit Type



- With heavy payloads the successful landing or positive landing rate are more for LEO and ISS.
- However, in the case of GTO, it is hard to distinguish between the positive landing rate and the negative landing because they are all gathered together.

# Launch Success Yearly Trend

---



- We can observe that the success rate since 2013 kept increasing till 2020, which is explained by the improvement the boosters used

# All Launch Site Names

---

```
SELECT DISTINCT(launch_site) FROM SPACEXTBL
```

- Display all distinct launch site name from the table

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

# Launch Site Names Begin with 'CCA'

```
SELECT * FROM SPACEXTBL WHERE launch_site LIKE 'CCA%' LIMIT 5
```

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

- Display the record with name beginning with 'CCA...'
- Only five records of the SpaceX table were displayed using LIMIT 5 clause in the query.

# Total Payload Mass

---

```
SELECT SUM(payload_mass__kg_) as total_payload_mass FROM SPACEXTBL WHERE customer = 'NASA (CRS)'
```

total\_payload\_mass

45596

- Use SUM() to get the sum of the total payload mass where the customer is NASA



# Average Payload Mass by F9 v1.1

---

```
SELECT AVG(payload_mass__kg_) as avg_payload_mass FROM SPACEXTBL WHERE booster_version LIKE 'F9 v1.1%'
```

avg\_payload\_mass

2534

- Use AVG() to get the average payload mass carried by the booster F9 v1.1

# First Successful Ground Landing Date

---

```
SELECT min(DATE) as first_ground_landing FROM SPACEXTBL WHERE landing__outcome = 'Success (ground pad)'
```

first\_ground\_landing

2015-12-22

- User MIN() to determinate the earliest date of the launch record where the landing outcome on ground was successful

## Successful Drone Ship Landing with Payload between 4000 and 6000

---

```
SELECT DISTINCT(booster_version) FROM SPACEXTBL WHERE landing__outcome = 'Success (drone ship)' AND payload_mass__kg_ between 4000 AND 6000
```

**booster\_version**

F9 FT B1021.2

F9 FT B1031.2

F9 FT B1022

F9 FT B1026

- Use 'Distinct()' to eliminate the duplicates from booster version
- 'Between 4000 and 6000' restrict the payload mass to be fetched only between those numbers.

# Total Number of Successful and Failure Mission Outcomes

---

```
WITH
  S (SUCCESSFUL) as (SELECT count(landing__outcome) as SUCCESSFUL FROM SPACEXTBL WHERE landing__outcome like 'Success%'),
  F (FAILURE) as (SELECT count(landing__outcome) as FAILURE FROM SPACEXTBL WHERE landing__outcome like 'Failure%')
SELECT SUCCESSFUL as SUCCESSFUL, FAILURE as FAILURE from S, F;
```

successful	failure
61	10

- 'WITH' creates from 'SELECT' queries, two or more in memories temporary tables
- We create 2 temp tables which contain respectively, the count of successful and failed landing

# Boosters Carried Maximum Payload

---

```
SELECT DISTINCT(booster_version) FROM SPACEXTBL WHERE payload_mass__kg_ = (SELECT MAX(payload_mass__kg_) FROM SPACEXTBL)
```

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

- The subquery use the 'MAX()' to get the max payload mass from table
- The value obtained from the subquery is then used to get all distinct booster whose mass is equal the max

# 2015 Launch Records

---

```
SELECT * FROM SPACEXTBL WHERE landing__outcome = 'Failure (drone ship)' AND YEAR(DATE) = 2015
```

DATE	time_utc_	booster_version	launch_site	payload	payload_mass_kg_	orbit	customer	mission_outcome	landing_outcome
2015-01-10	09:47:00	F9 v1.1 B1012	CCAFS LC-40	SpaceX CRS-5	2395	LEO (ISS)	NASA (CRS)	Success	Failure (drone ship)
2015-04-14	20:10:00	F9 v1.1 B1015	CCAFS LC-40	SpaceX CRS-6	1898	LEO (ISS)	NASA (CRS)	Success	Failure (drone ship)

- 'YEAR()' extract the year value from a date which is used to restrict the records on 2015 launch only

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

---

```
SELECT landing_outcome, count(landing_outcome) AS FREQ FROM(SELECT * FROM SPACEXTBL WHERE DATE >= '2010-06-04' AND DATE <= '2017-03-20')  
GROUP BY landing_outcome ORDER BY FREQ DESC
```

landing_outcome	freq
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

- Use a subquery to select landing Outcomes between 2010-06-04 and 2017-03-20
- Then from the subquery group by landing outcome
- Order the result by frequency or the landing outcome count



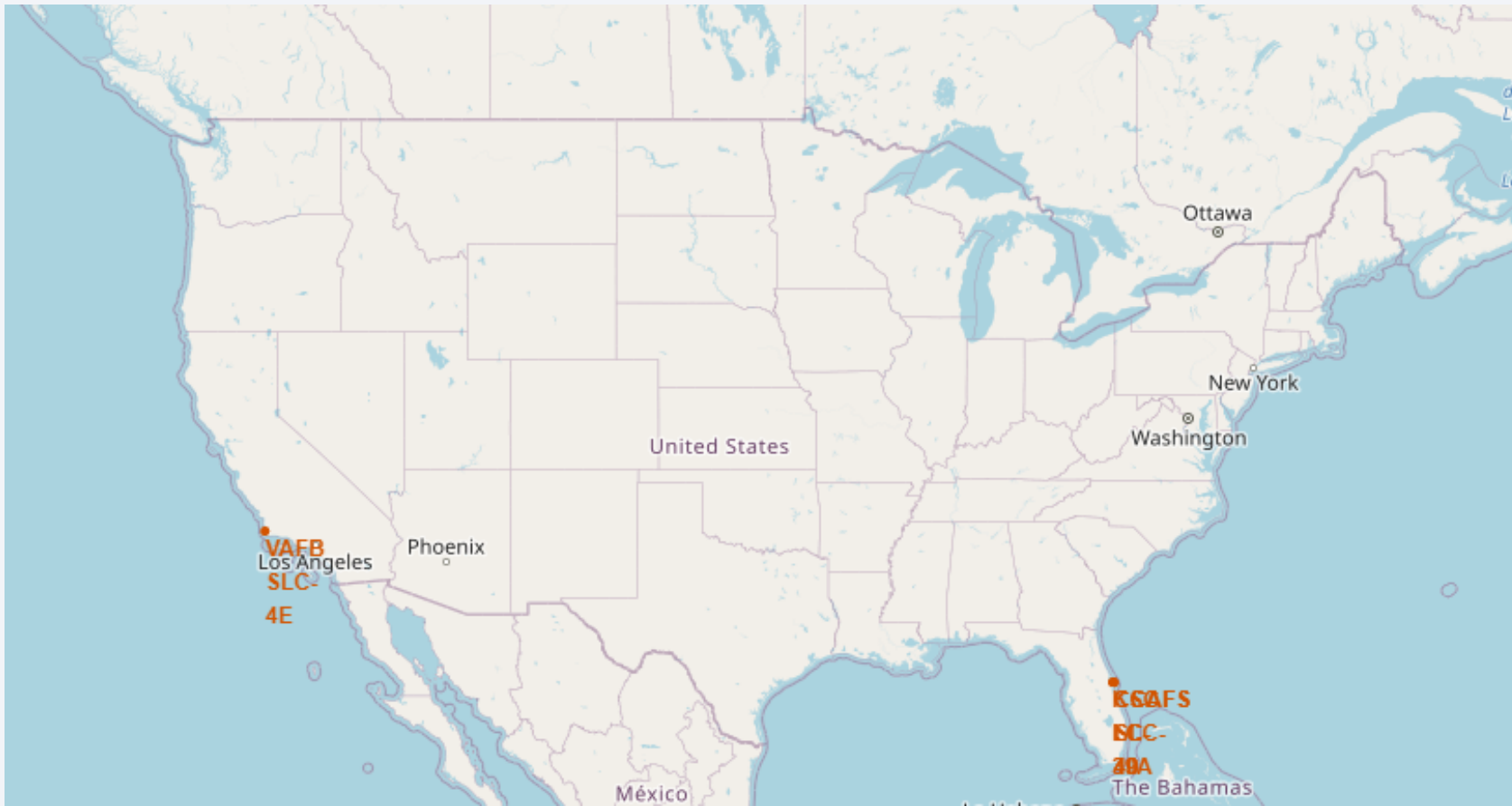
A satellite view of Earth from space, showing the curvature of the planet and city lights at night. The background is a deep blue gradient.

Section 3

# Launch Sites Proximities Analysis

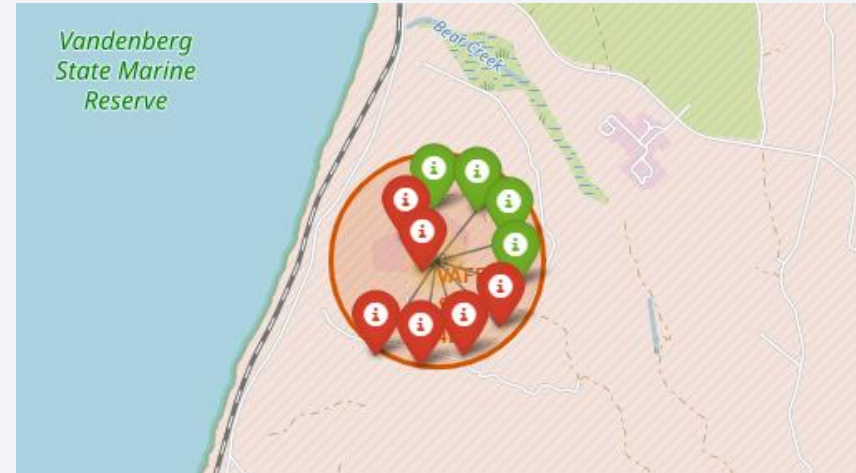
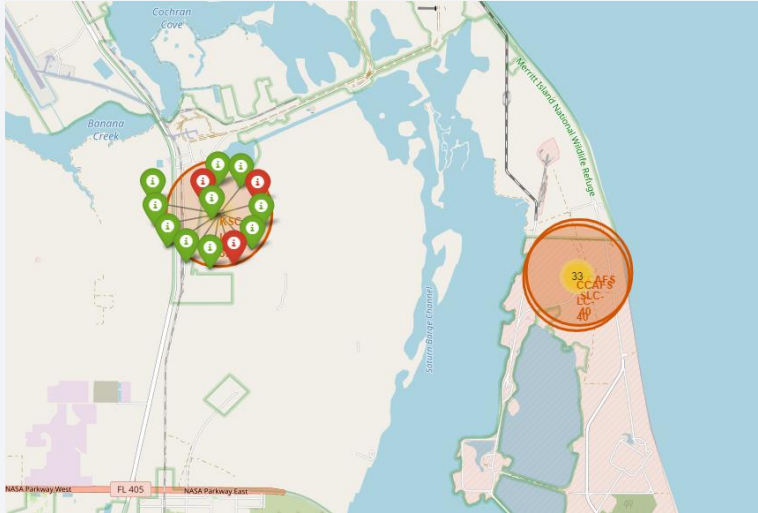
# All Launch Sites' Locations

---



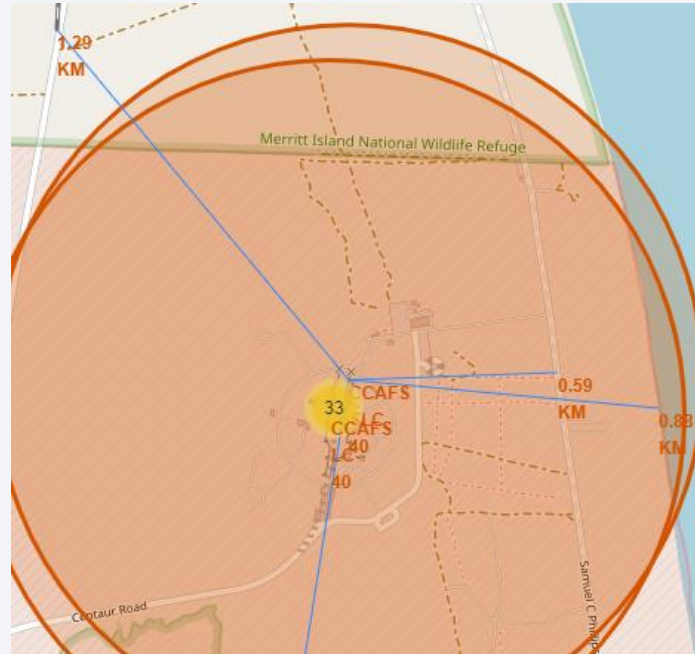
- Shows the launch sites on the world map which are all located on the USA

# Launch site with landing outcomes



- Respectively Florida and California launch site
- Successful landing is represented by green marker
- Failed landing is represented by red marker

# Launch Sites Proximities



- The launch sites are all placed at strategic locations
- Close to railways and highways to facilitate logistics (<1km)
- And for security measures, the site is close to water points(<1km) and far enough from the city(> 12km)





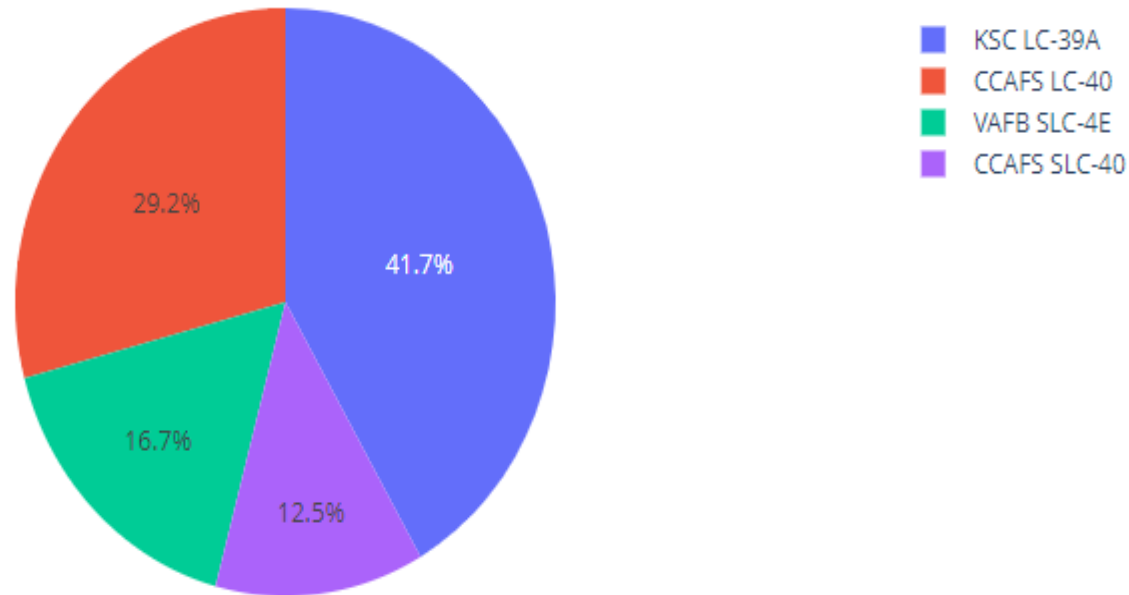
Section 4

# Build a Dashboard with Plotly Dash

# Success Landing distribution

---

Total Success Launches By Site

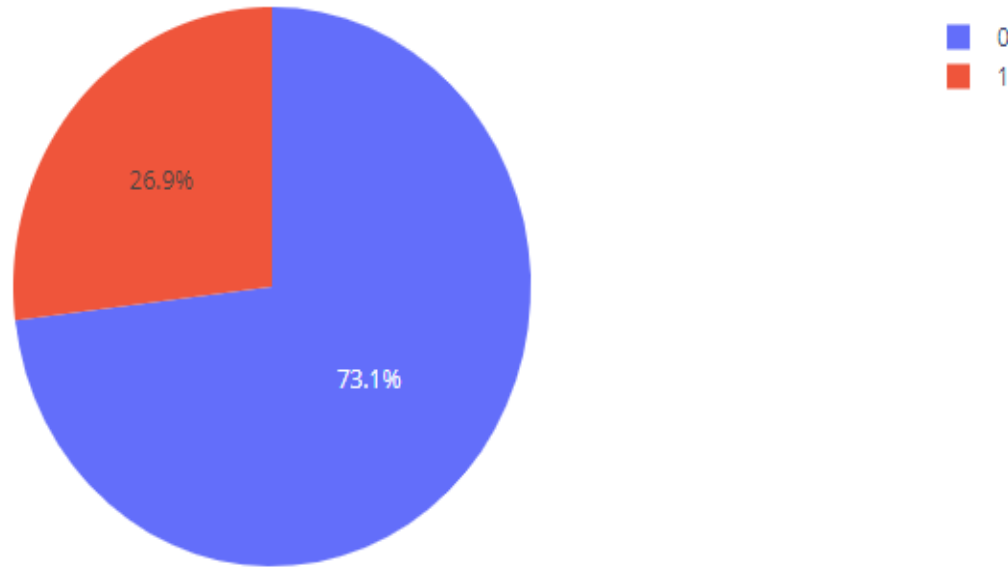


- This chart shows the distribution of the success landing among all sites

# Proportion of the successful/failed landing

---

Total Success Launched for site CCAFS LC-40

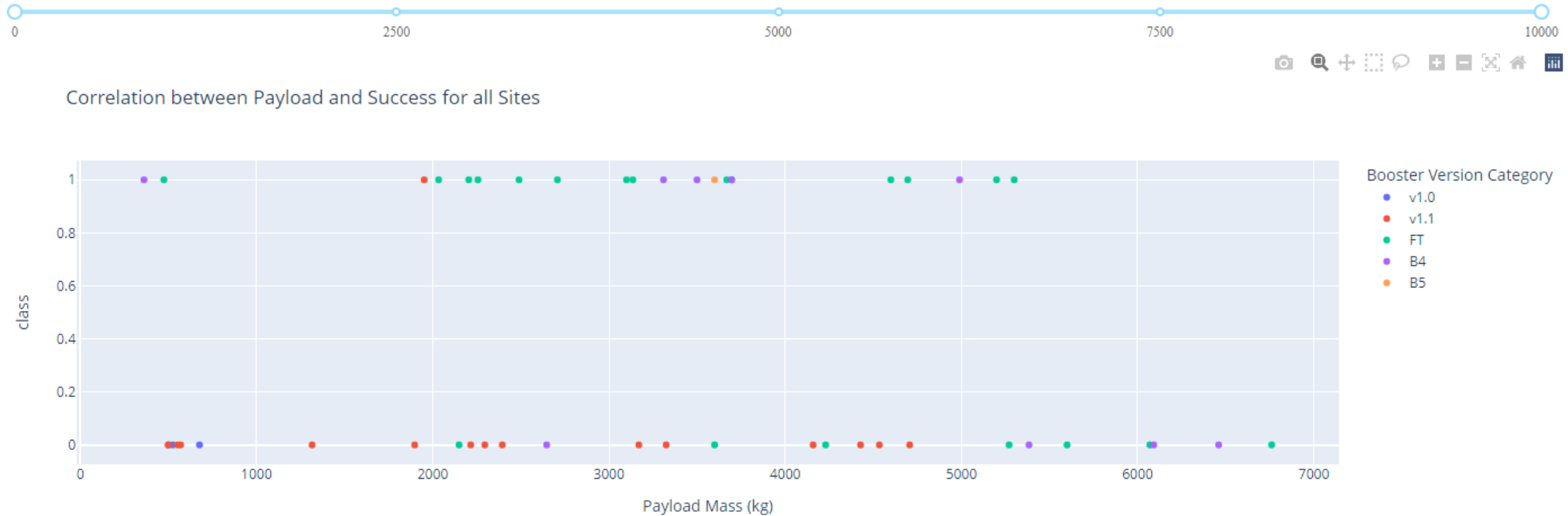


- This chart shows the proportion of the successful/failed landing of a specific site (i.e., CCAFS LC-40)



# Payload vs. Launch Outcome

Payload range (Kg):



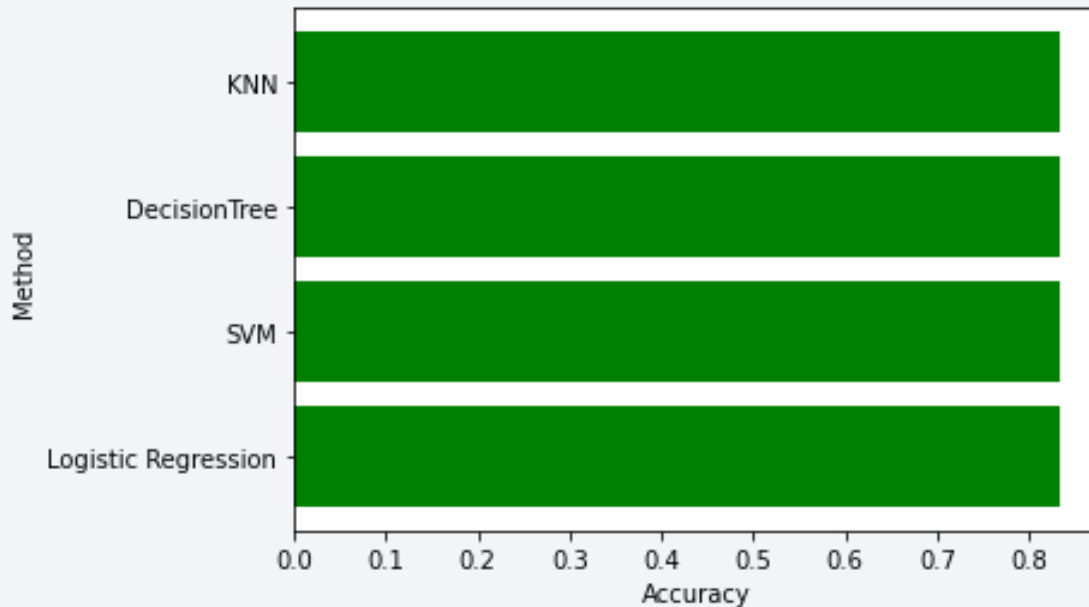
- These figures show the relation between the payload and the launch success rate

Section 5

# Predictive Analysis (Classification)

# Classification Accuracy

---



- Using the test set, all models was accurate at 83.33% of time
- We should note that test set only contains 18 instances, which may explain this behavior

# Confusion Matrix



- The models predicted 100% (12/12) of the 12 successful landing
- But only classified 50% (3 of the 6) of failed landings
- This model is better at classifying successful landings

# Conclusions

---

- Displaying launch sites on a real-world map, revealed the strategies behind their location.
- Exploring visually and analytically, give us an overview of the evolution of the rocket of time. Evolution in payload mass, in successful landing rate, in the max reach of the rocket (different orbits)
- Exploring the data visually and analytically gives us insight into the evolution of the time rocket. Evolution of payload mass, successful landing rate, maximum rocket range (different orbits), the frequency of the launch.
- Finally, it is not yet recommended to build a predictive model (a relevant model) on spaceX launches due to lack of data

# Appendix

---

- [GitHub Project](#)



Thank you!

