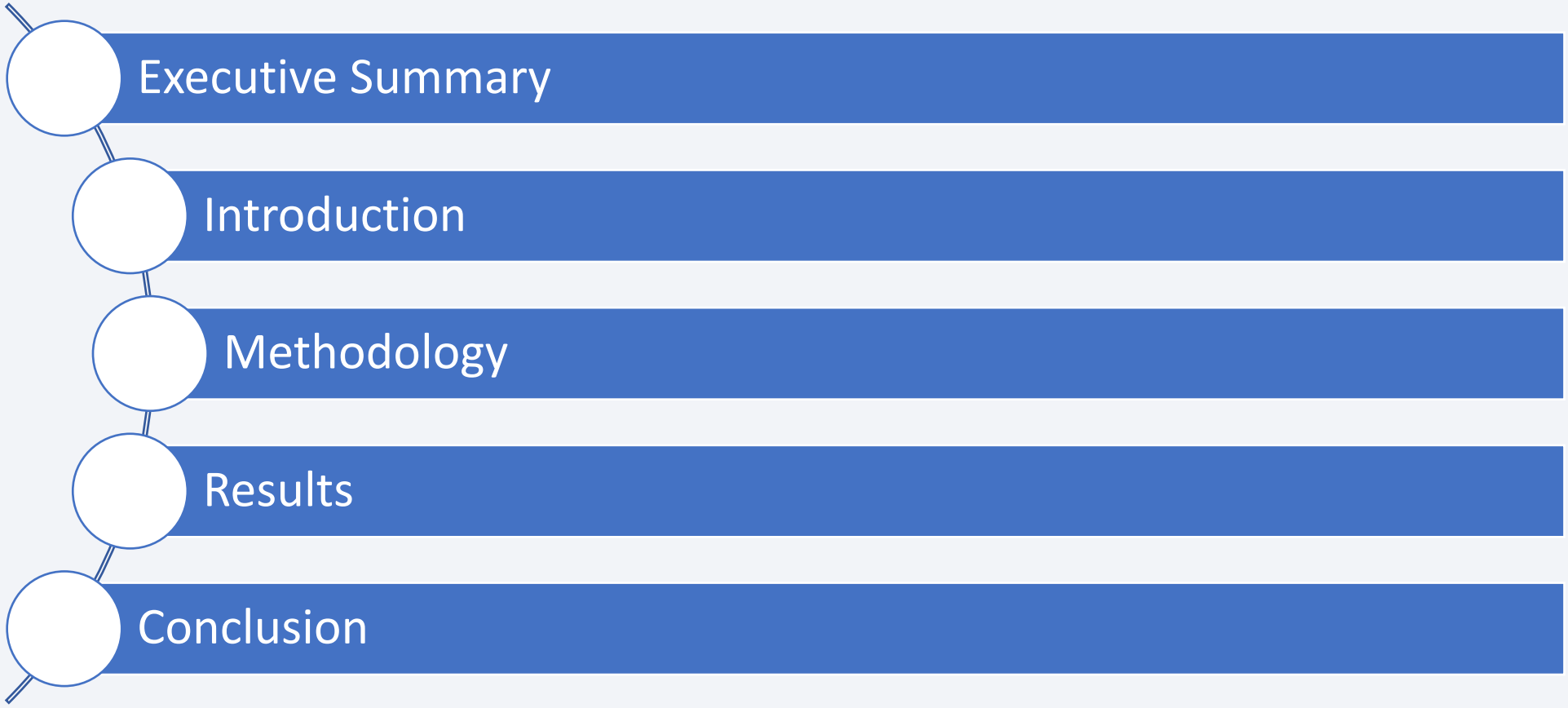
IBM Developer
SKILLS NETWORK

IBM DATA SCIENCE CAPSTONE PROJECT

Winning Space Race with Data Science

Obaidullah YAQUBI
22/03/2022

Outline


Executive Summary
Introduction
Methodology
Results
Conclusion

Executive Summary

- Summary of methodologies
  - Collecting data using API, web-scrapping, and SQL
  - Data wrangling and Analysis
  - Constructing interactive maps with Folium
  - Implementing predictive machine learning tools
- Summary of all results
  - Building a story around data and how results were obtained
  - Identification of best model for prediction using a comparative analysis of results

Introduction

- **Project background and context**

SpaceX advertises falcon 9 rocket launches on its website claiming that the cost of each launch is 62 million dollars. For other providers the cost of a similar operation is above 165 million dollars. SpaceX states much of the saving is due to the reuse of first-stage rocket parts. Therefore, if we can determine the success rate in the reuse of first stage parts, then the information could be used by an alternative company to bid against SpaceX.

- **Problems you want to find answers**

- Finding information about factors that influence successful landing?
- Determining the relative effect of each factor on the outcome (success/failure)
- What are the conditions the can help SpaceX or any other company in this business to achieve the best results?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Via Rest API of SpaceX
  - Web-scrapping data from Wikipedia
- Perform data wrangling
  - One-hot encoding was applied to transform categorical variables in numerical values
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection

- Describe how data sets were collected.
  - Data collection was done using get request to the SpaceX API.
  - Next, I decoded the response content as a Json using .json() function call and turn it into a pandas DataFrame using .json_normalize().
  - Then I cleaned the data, checked for missing values and filled in missing values where necessary with mean().
  - In addition, I performed web scraping from Wikipedia for Falcon 9 launch records with Beautiful Soup.
  - The objective was to extract the launch records as HTML table, parse the table and convert it to a pandas DataFrame for further analysis.

Data Collection – SpaceX API

- Useed get request to the SpaceX API to collect data, cleaned the requested data and performed some basic data wrangling and formatting.
- Add the GitHub URL of the completed SpaceX API calls notebook: https://github.com/ObaidYaqubi/CapstoneProject/blob/4f0e561c42c3dc3d4e574dada4084b4fac3e77f5/spacex-data-collection-api%20(1).ipynb

Task 1: Request and parse the SpaceX launch data using the GET request

To make the requested JSON results more consistent, we will use the following static response object for this project:

```
In [9]: static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNet
```

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

```
In [10]: response = requests.get(static_json_url)
         response.status_code

Out[10]: 200
```

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

```
In [11]: # Use json_normalize meethod to convert the json result into a dataframe
         static_json_df = response.json()
         data = pd.json_normalize(static_json_df)
```

Using the dataframe `data` print the first 5 rows

```
In [13]: # Get the head of the dataframe
         data.head(3)
```

Out[13]:

static_fire_date_utc	static_fire_date_unix	tbd	net	window	rocket	success	details	crew	ships	capsules
2006-03							Engine failure at 33			

Data Collection - Scraping

- Applied web scrapping to webscrap Falcon 9 launch records with Beautiful Soup
- Parsed the table and converted it into a pandas DataFrame.
- Add the GitHub URL of the completed web scraping notebook:

https://github.com/ObaidYaqubi/CapstoneProject/blob/b5ec8d94fcc783a0a5cf26df66b23f67ab8ac044/web-scraping%20wikipedia%20(1).ipynb

TASK 1: Request the Falcon9 Launch Wiki page from its URL

First, let's perform an HTTP GET method to request the Falcon9 Launch HTML page, as an HTTP response.

```
In [5]: # use requests.get() method with the provided static_url
        html_data = requests.get(static_url)
        # assign the response to a object
        html_data.status_code
Out[5]: 200
```

Create a `BeautifulSoup` object from the HTML `response`

```
In [6]: # Use BeautifulSoup() to create a BeautifulSoup object from a response text content
        soup = BeautifulSoup(html_data.text, 'html.parser')
```

Print the page title to verify if the `BeautifulSoup` object was created properly

```
In [7]: # Use soup.title attribute
        soup.title
Out[7]: <title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>
```

TASK 2: Extract all column/variable names from the HTML table header

Next, we want to collect all relevant column names from the HTML table header

Let's try to find all tables on the wiki page first. If you need to refresh your memory about `BeautifulSoup`, please check the the end of this lab

```
In [8]: # Use the find_all function in the BeautifulSoup object, with element type `table`
        # Assign the result to a list called `html_tables`
        html_tables = soup.find_all('table')
```

Starting from the third table is our target table contains the actual launch records.

```
In [1]: ## Let's print the third table and check its content
        first_launch_table = html_tables[2]
        #print(first_launch_table)
```

Next, we just need to iterate through the `<th>` elements and apply the provided `extract_column_from_header()` to ex

```
In [54]: column_names = []

         #column_names = []

         for row in first_launch_table.find_all('th'):
             columns = extract_column_from_header(row)
             if columns is not None and len(columns) > 0:
                 column_names.append(columns)
         # Apply find_all() function with `th` element on first_launch_table
```

Data Wrangling

- Performed exploratory data analysis and determined the training labels.
- Calculated the number of launches at each site, and the number and occurrence of each orbits
- Created landing outcome label from outcome column and exported the results to csv.

- Add the GitHub URL of your completed data wrangling related notebooks:

https://github.com/ObaidYaqubi/CapstoneProject/blob/38293f7fd9d02f98c56f3895facd50e83e627f32/spacex-Data%20wrangling%20(1).ipynb

EDA with Data Visualization

- Summarize what charts were plotted and why you used those charts

Data was explored by visualizing the relationship between **flight number** and **launch Site**, **payload** and launch site, success rate of each **orbit type**, flight number and orbit type, the **launch success yearly trend**.

- Add the GitHub URL of your completed EDA with data visualization notebook:

https://github.com/ObaidYaqubi/CapstoneProject/blob/52432b90e41e3eaabb180806362f29ad4f7cf12d/Exploratory%20Data%20analysis.ipynb

EDA with SQL

- Loaded the SpaceX dataset into a PostgreSQL database using the jupyter magic sql queries.
- Applied EDA with SQL to get insight from the data. We wrote queries to find out for instance:
  - The names of unique launch sites in the space mission.
  - The total payload mass carried by boosters launched by NASA (CRS)
  - The average payload mass carried by booster version F9 v1.1
  - The total number of successful and failure mission outcomes
  - The failed landing outcomes in drone ship, their booster version and launch site names.

- GitHub URL of completed EDA with SQL notebook:
https://github.com/ObaidYaqubi/CapstoneProject/blob/629122d8b2d455996a5882ac8bf79bfec4abf9fd/eda-sql-coursera%20(1).ipynb

Build an Interactive Map with Folium

- Marked all launch sites, and added map objects such as markers, circles, lines to mark the success or failure of launches for each site on the folium map.
- Assigned the feature launch outcomes (failure or success) to class 0 and 1
- Using the color-labeled marker clusters, identified which launch sites have relatively high success rate.
- Calculated the distances between a launch site to its proximities to answer questions :
  - Are launch sites near railways, highways and coastlines.
  - Do launch sites keep certain distance away from cities.

GitHub URL of completed interactive map with Folium map:

https://github.com/ObaidYaqubi/CapstoneProject/blob/951efe5e9950d1f7ab0e7993f8b11a8cacb3f4d1/launch_site_location%20(1).ipynb

Build a Dashboard with Plotly Dash

- Built an interactive dashboard with Plotly dash
- Plotted pie charts showing the total launches by a certain sites
- Plotted scatter graph showing the relationship with Outcome and Payload Mass (Kg) for the different booster version.
- Why?
  - For a more dynamic exploratory analysis
  - Give the client the ability to explore what is in the data and draw insight
- Add the GitHub URL of your completed Plotly Dash lab:

https://github.com/ObaidYaqubi/CapstoneProject/blob/61440591e72c6979a9b4207cf230ecff290d1b3e/dash_board_spacex.ipynb

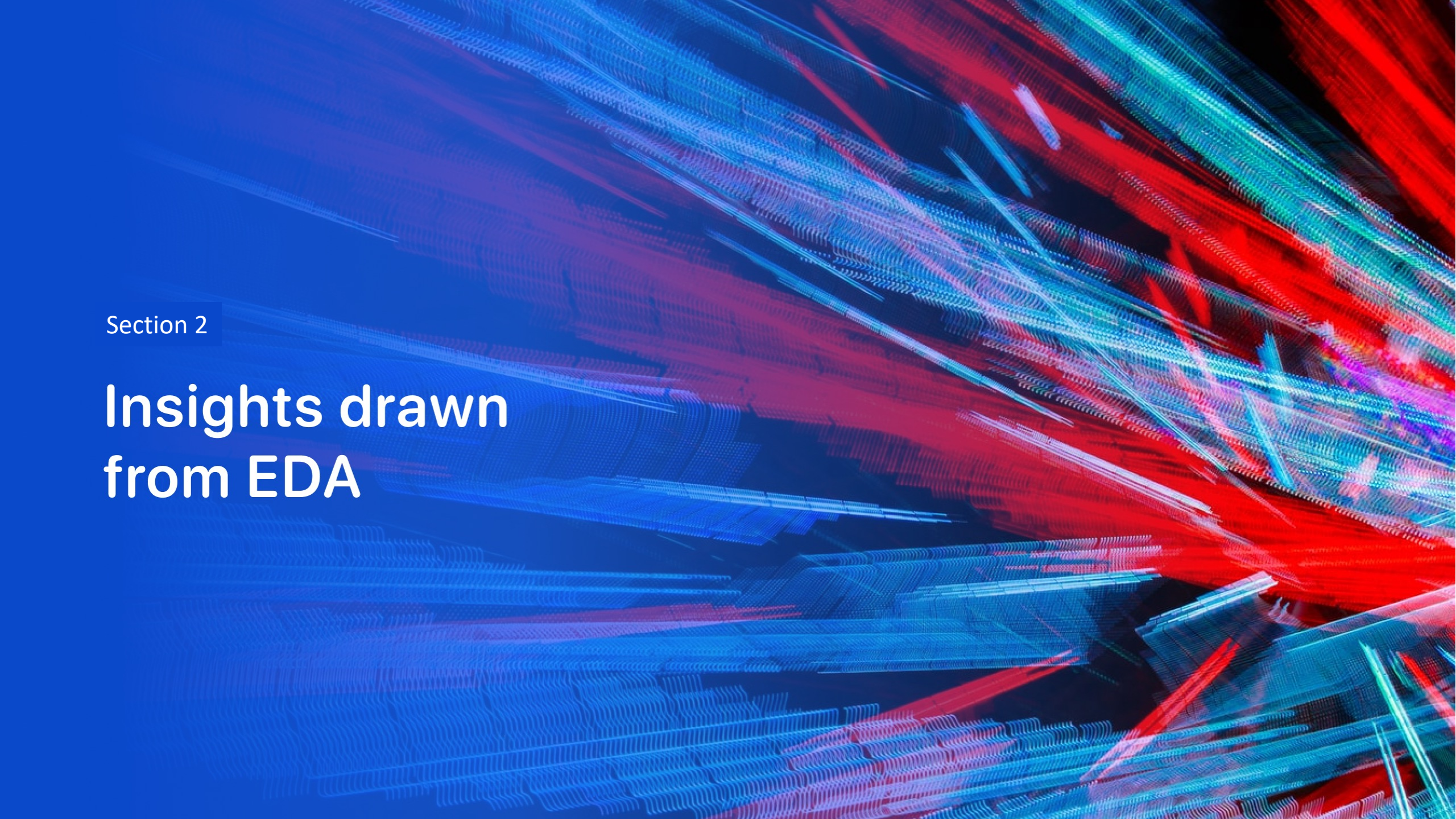
Predictive Analysis (Classification)

- Loaded the data using numpy and pandas, transformed the data, split our data into training and testing.
- Built different machine learning models and tune different hyperparameters using GridSearchCV.
- Used accuracy as the metric for our model, improved the model using feature engineering and algorithm tuning.
- Identified the best performing classification model.
- Add the GitHub URL of your completed predictive analysis lab:

https://github.com/ObaidYaqubi/CapstoneProject/blob/ce77a9b9f4c350e1c3251ef3b21de48e620a10e9/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results

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

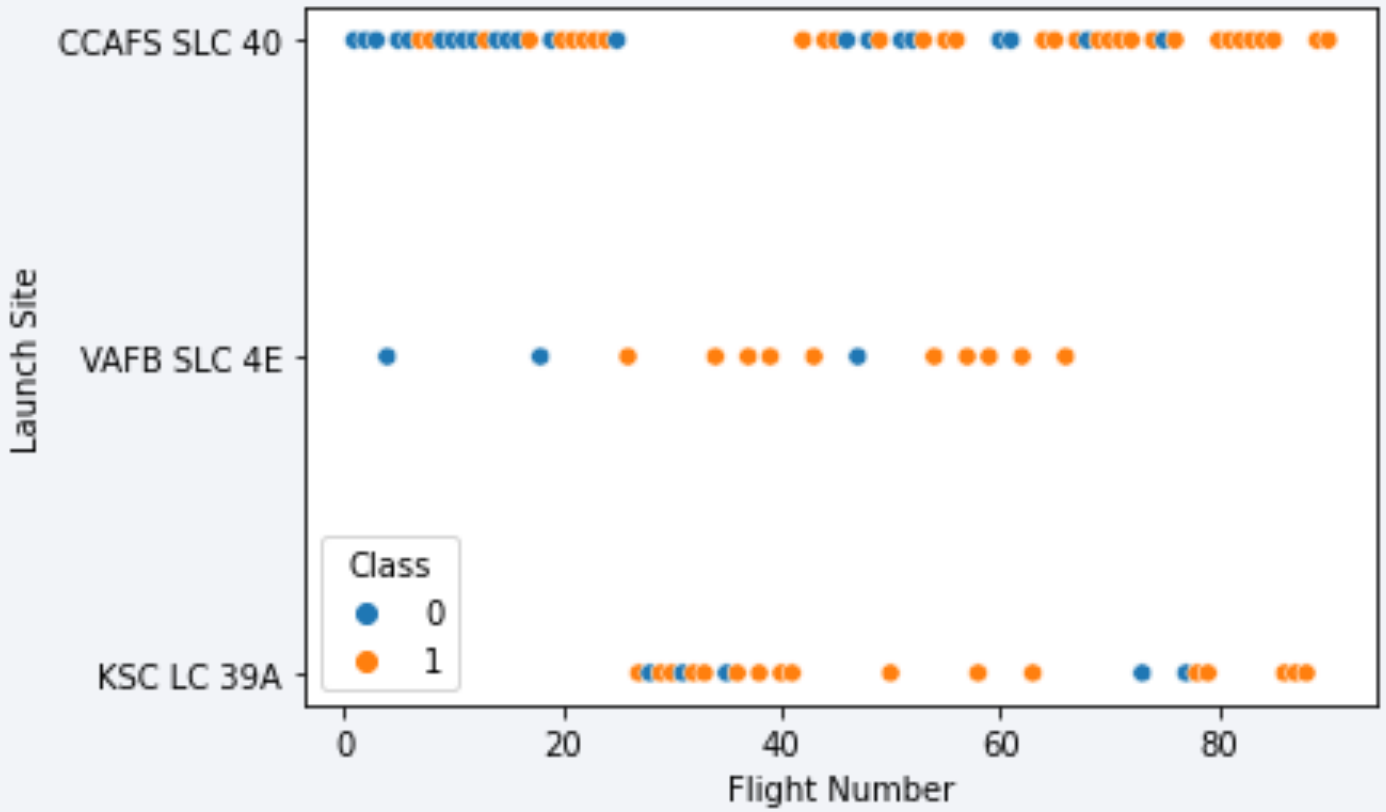

Section 2
Insights drawn from EDA

Flight Number vs. Launch Site

- From the plot, it can be seen that the larger the flight amount at a launch site, the greater the success rate at a launch site.

Class 0: failure

Class 1: success

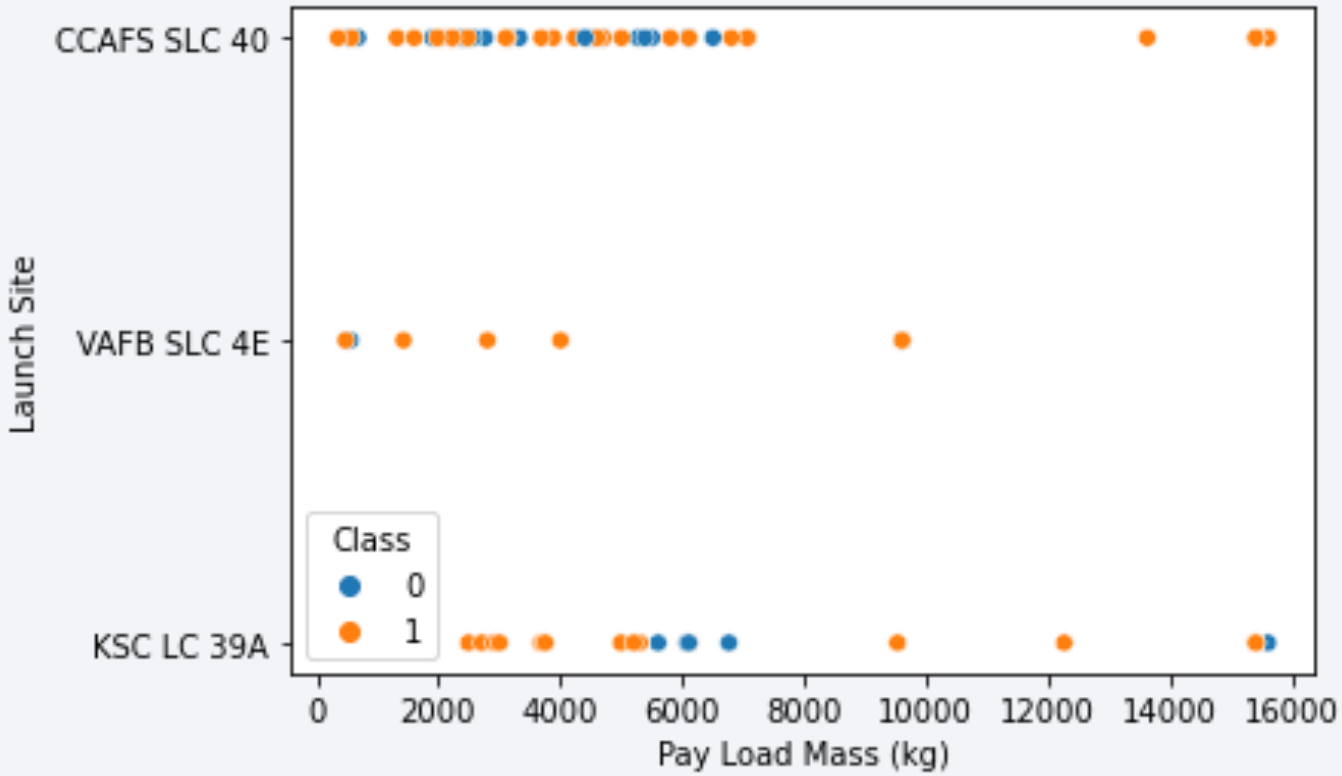


Payload vs. Launch Site

- Payload of 7000 [kg] and above has the highest success rate from all sites except for KSC LC 39A site.

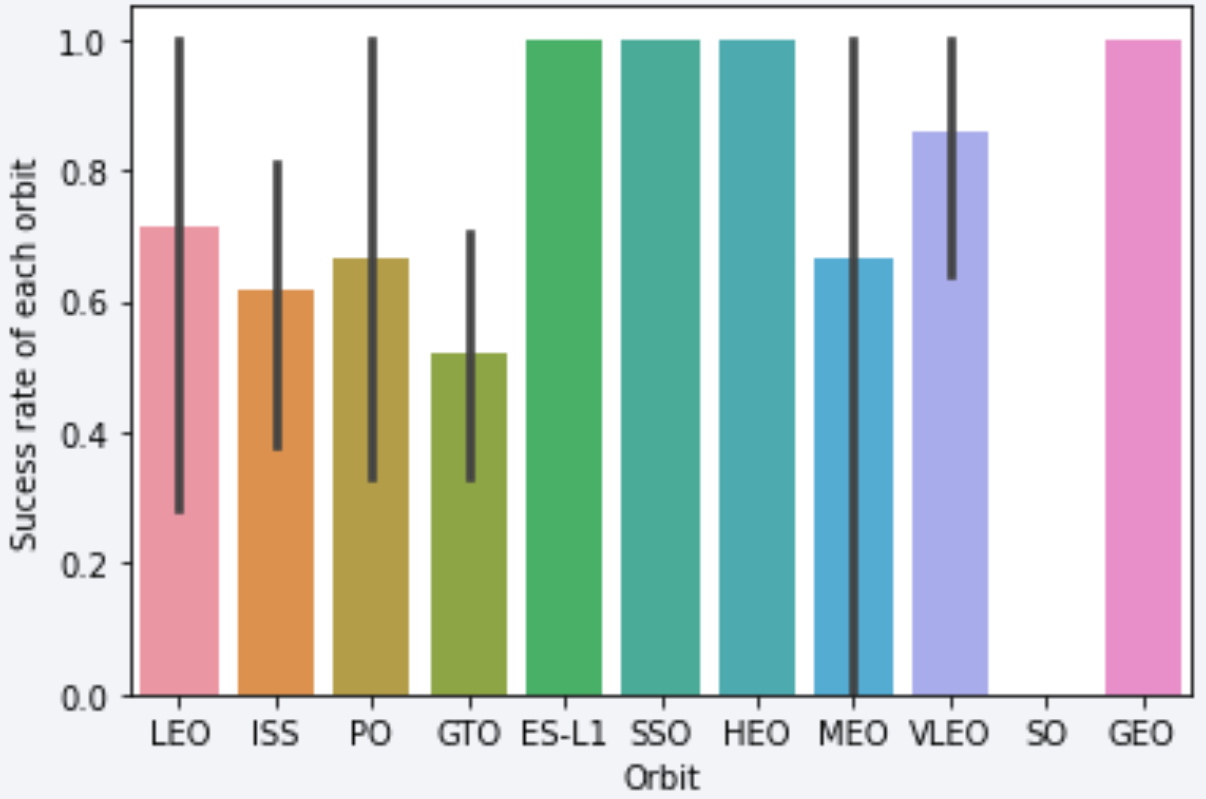
Class 0: failure

Class 1: success



Success Rate vs. Orbit Type

- From the plot, we can see that ES-L1, GEO, HEO, SSO, VLEO had the most success rate.

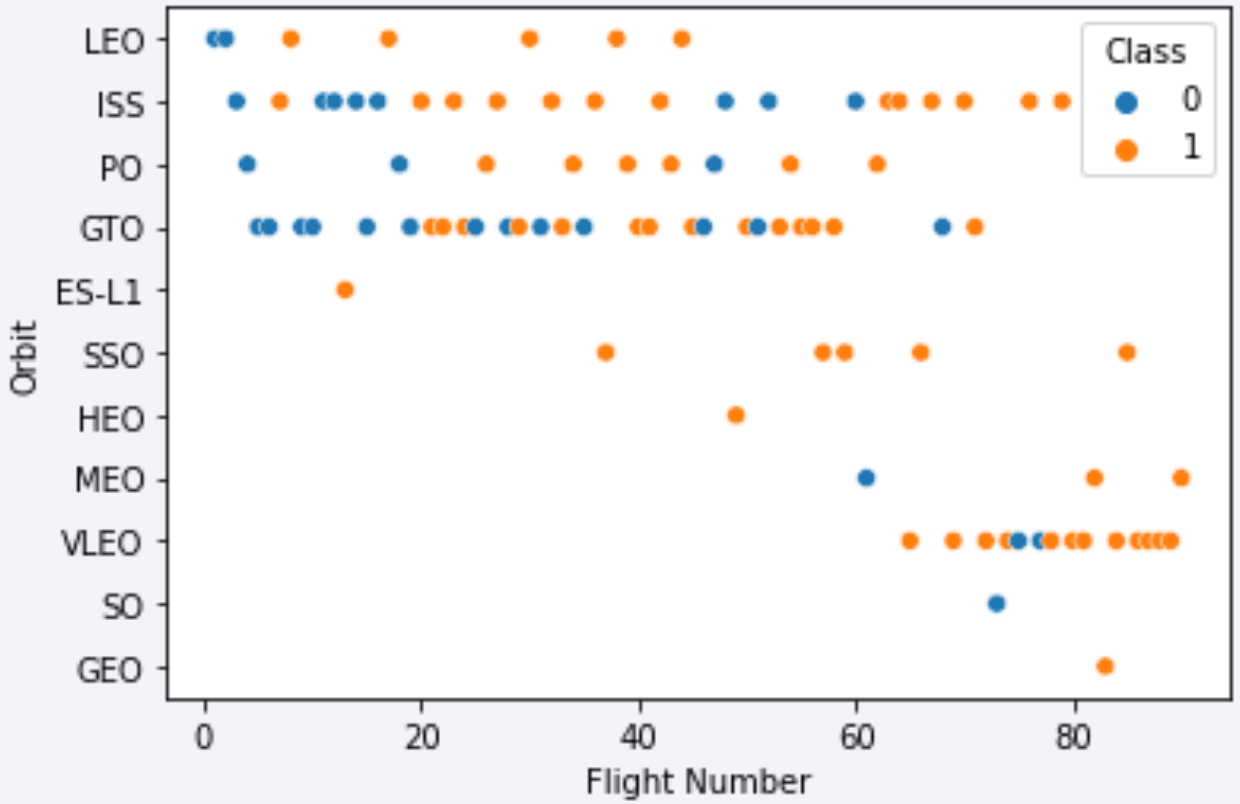


Flight Number vs. Orbit Type

- It can be observed that in the LEO orbit, success is related to the number of flights whereas in the GTO orbit, there is no relationship between flight number and the orbit.

Class 0: failure

Class 1: success

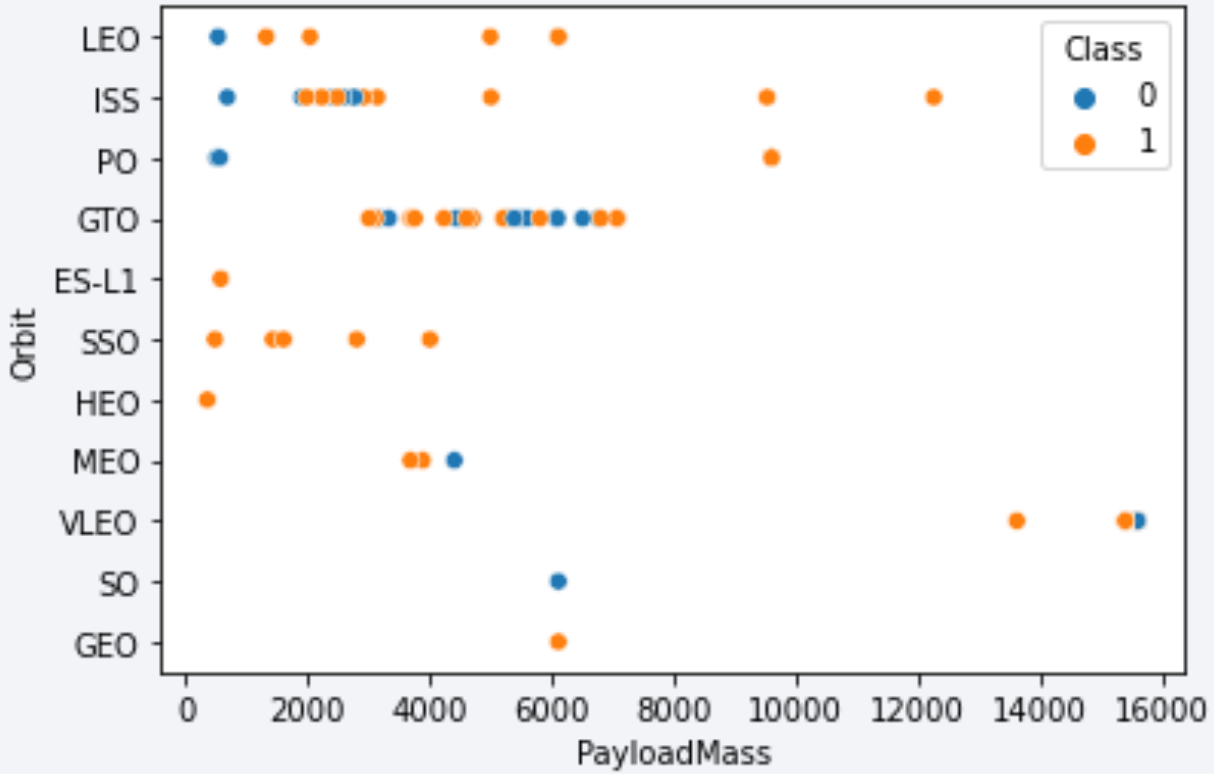


Payload vs. Orbit Type

- It can be observed that with heavy payloads, the successful landing are more for likely for PO, LEO and ISS orbits.

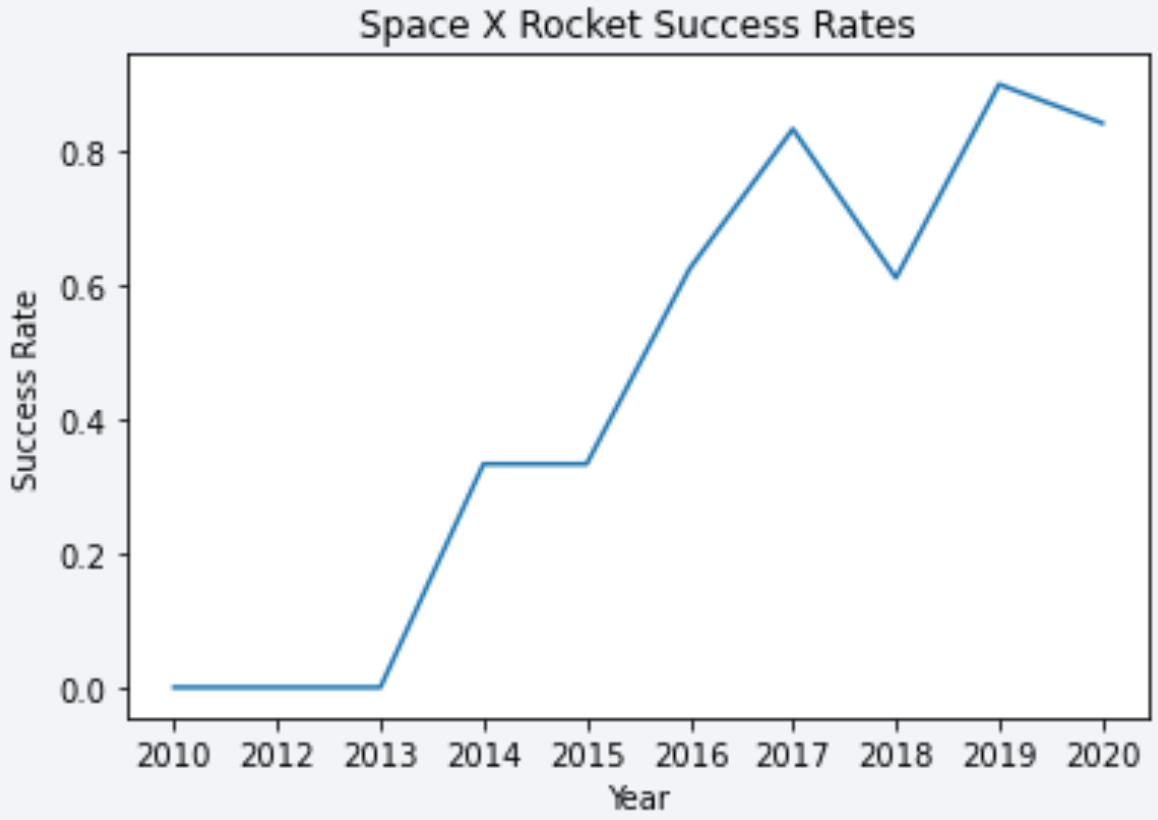
Class 0: failure

Class 1: success



Launch Success Yearly Trend

- From the plot, it can be observed that success rate has been improving since 2013 and closed in to almost 90% in 2020.



All Launch Site Names

Key word **DISTINCT** was used to show only unique launch sites from the SpaceX data.

Task 1

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

In [11]:
```
%sql SELECT DISTINCT LAUNCH_SITE as "Launch_Sites" FROM SPACEX;
```

```
 * ibm_db_sa://gzd92406:***@815fa4db-dc03-4c70-869a-a9cc13f33084.bs
7/bludb
Done.
```

Out[11]:

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

Launch Site Names Begin with 'CCA'

- The term LIKE ‘CCA%’ was used to identify records that start with CCA.

Task 2

Display 5 records where launch sites begin with the string 'CCA'

In [12]:
```
%sql SELECT * FROM SPACEX WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5;
```

```
 * ibm_db_sa://gzd92406:***@815fa4db-dc03-4c70-869a-a9cc13f33084.bs2io90l08
7/bludb
Done.
```

Out[12]:

DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_
2010-04-06	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0
2010-08-12	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0
2012-08-10	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500
2013-01-03	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677
2013-03-12	22:41:00	F9 v1.1	CCAFS LC-40	SES-8	3170

Total Payload Mass

- Query for total payload carried by boosters from NASA

Task 3

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

```
In [13]: %sql SELECT SUM(PAYLOAD_MASS__KG_) AS "Total Payload Mass by NASA (CRS)" FROM SPACEX WHERE CUSTOMER = 'NASA (CRS)';
```

Total Payload Mass by NASA (CRS)
44014

The Function SUM(PAYLOAD_MASS_KG_) was used to calculate total payload

Average Payload Mass by F9 v1.1

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

Task 4

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

In [14]:
```
%sql SELECT AVG(PAYLOAD_MASS__KG_) AS "Average Payload Mass by Booster Version F9 v1.1" FROM SPACEX \
WHERE BOOSTER_VERSION = 'F9 v1.1';
```

```
Done.
```

Out[14]:

Average Payload Mass by Booster Version F9 v1.1
3676

The Function AVG(PAYLOAD_MASS_KG_) was used to calculate average payload

First Successful Ground Landing Date

- The dates of the first successful landing outcome on ground pad

Task 5

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

Hint:Use min function

```
In [15]: %sql SELECT MIN(DATE) AS "First Succesful Landing Outcome in Ground Pad" FROM SPACEX \
WHERE LANDING__OUTCOME = 'Success (ground pad)';
```

First Succesful Landing Outcome in Ground Pad
2015-12-22

MIN(DATE) was used to filter out the first successful landing

Successful Drone Ship Landing with Payload between 4000 and 6000

Task 6

List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

In [16]:
```
%sql SELECT BOOSTER_VERSION FROM SPACEX WHERE LANDING__OUTCOME = 'Success (drone ship)' \
AND PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 6000;
```

```
Done.
```

Out[16]:

booster_version
F9 FT B1022
F9 FT B1031.2
F9 FT B1022
F9 FT B1031.2

WHERE query was used to filter for boosters which have successfully landed on drone ship and applied the **AND** condition to determine successful landing with payload mass greater than 4000 but less than 6000

Total Number of Successful and Failure Mission Outcomes

- The total number of successful and failure mission outcomes

Task 7

List the total number of successful and failure mission outcomes

```
In [17]: %sql SELECT COUNT(MISSION_OUTCOME) AS "Successful Mission" FROM SPACEX WHERE MISSION_OUTCOME LIKE 'Success%';
```

```
In [18]: %sql SELECT COUNT(MISSION_OUTCOME) AS "Failure Mission" FROM SPACEX WHERE MISSION_OUTCOME LIKE 'Failure%';
```

Successful Mission
100

Failed Mission
1

Wildcard like '%' to filter for **WHERE** MissionOutcome was a success or a failure.

Boosters Carried Maximum Payload

- Names of the booster which have carried the maximum payload mass

Task 8

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

In [19]:
```
%sql SELECT DISTINCT BOOSTER_VERSION AS "Booster Versions which carried the Maximum Payload Mass" FROM SPACEX \
WHERE PAYLOAD_MASS__KG_ =(SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEX);
```

```
Done.
```

Out[19]:

Booster Versions which carried the Maximum Payload Mass
F9 B5 B1048.4
F9 B5 B1049.4
F9 B5 B1049.5
F9 B5 B1058.3
F9 B5 B1060.2

A subquery was used to find the names of boosters that carried the maximum payload

2015 Launch Records

- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

Task 9

List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

In [20]:
```
%sql SELECT BOOSTER_VERSION, LAUNCH_SITE FROM SPACEX WHERE DATE LIKE '2015-%' AND \
LANDING__OUTCOME = 'Failure (drone ship)';
```

```
Done.
```

Out[20]:

booster_version	launch_site
F9 v1.1 B1012	CCAFS LC-40
F9 v1.1 B1012	CCAFS LC-40

A combinations of the **WHERE** clause, **LIKE**, **AND**, and **BETWEEN** conditions to filter for failed landing outcomes in drone ship, their booster versions, and launch site names for year 2015

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

Task 10

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

In [22]:
```
%sql SELECT LANDING__OUTCOME as "Landing Outcome", COUNT(LANDING__OUTCOME) AS "Total Count" FROM SPACEX \
WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' \
GROUP BY  LANDING__OUTCOME \
ORDER BY COUNT(LANDING__OUTCOME) DESC ;
```

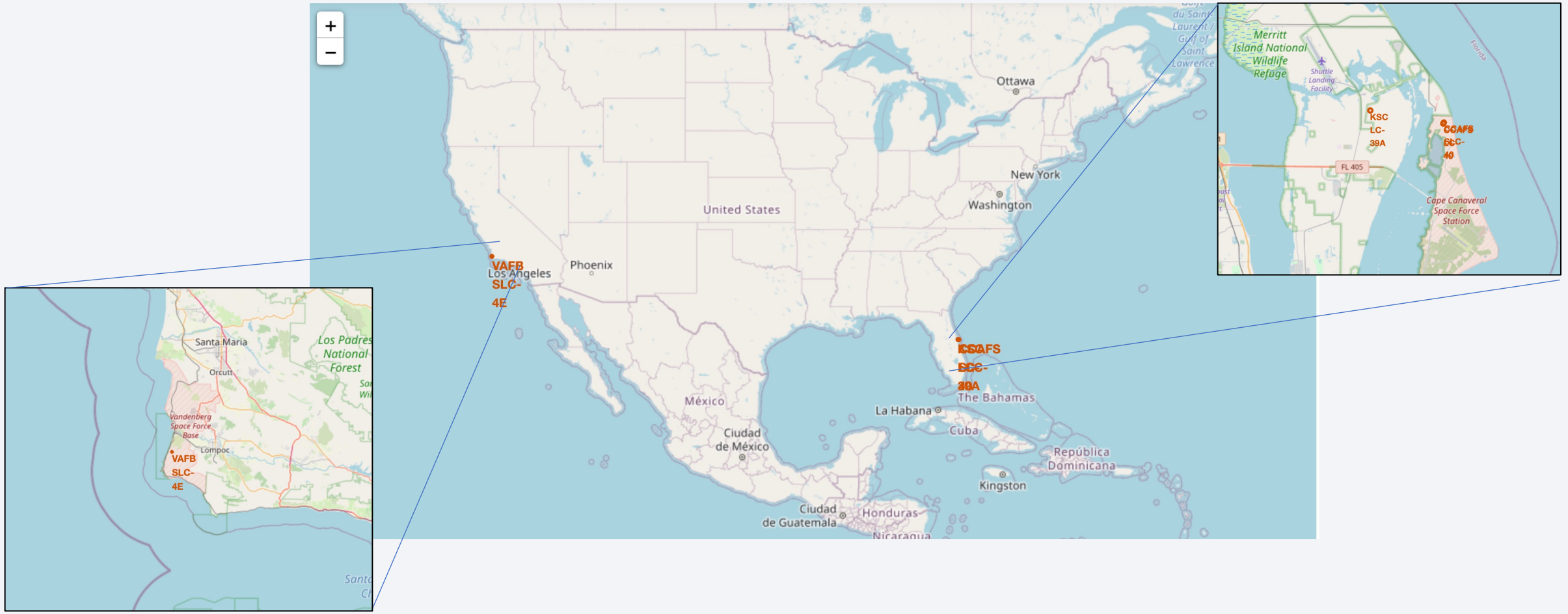
Out[22]:

Landing Outcome	Total Count
No attempt	14
Failure (drone ship)	4
Success (drone ship)	4
Success (ground pad)	4
Controlled (ocean)	2
Failure (parachute)	2

- Used **COUNT** of landing outcomes from the data and the **WHERE** query to filter for landing outcomes **BETWEEN** 2010-06-04 to 2017-03-20.
- Applied the **GROUP BY** query to group the landing outcomes and the **ORDER BY** to order the grouped landing outcome in descending order.

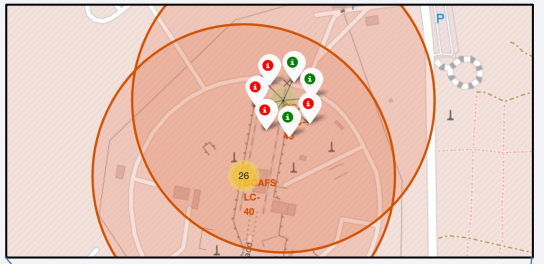

Section 3
Launch Sites
Proximities Analysis

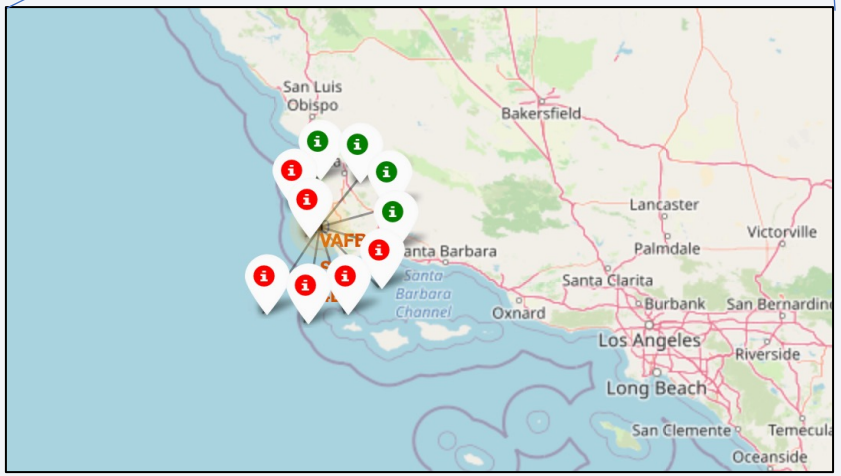
Launch Sites of SpaceX space rockets

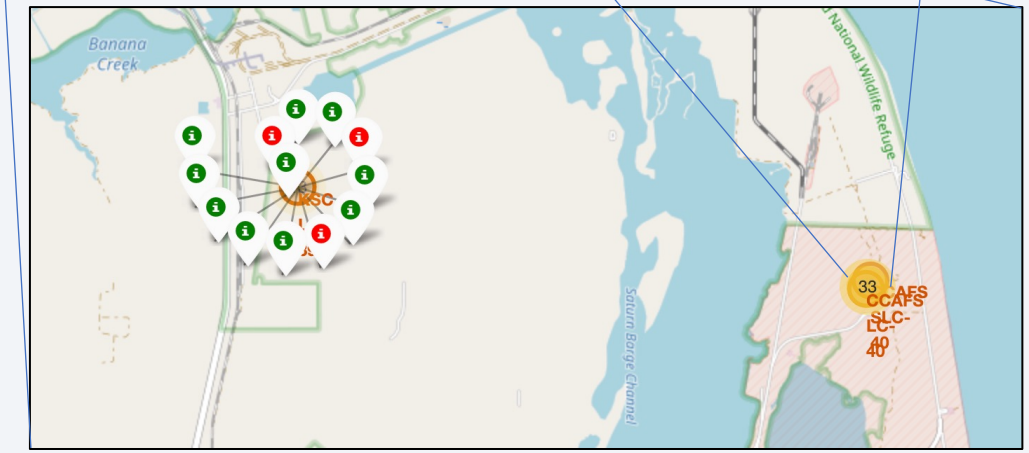


Launch sites are on both east and west coasts of United States

Clusters of Launch sites and success rate



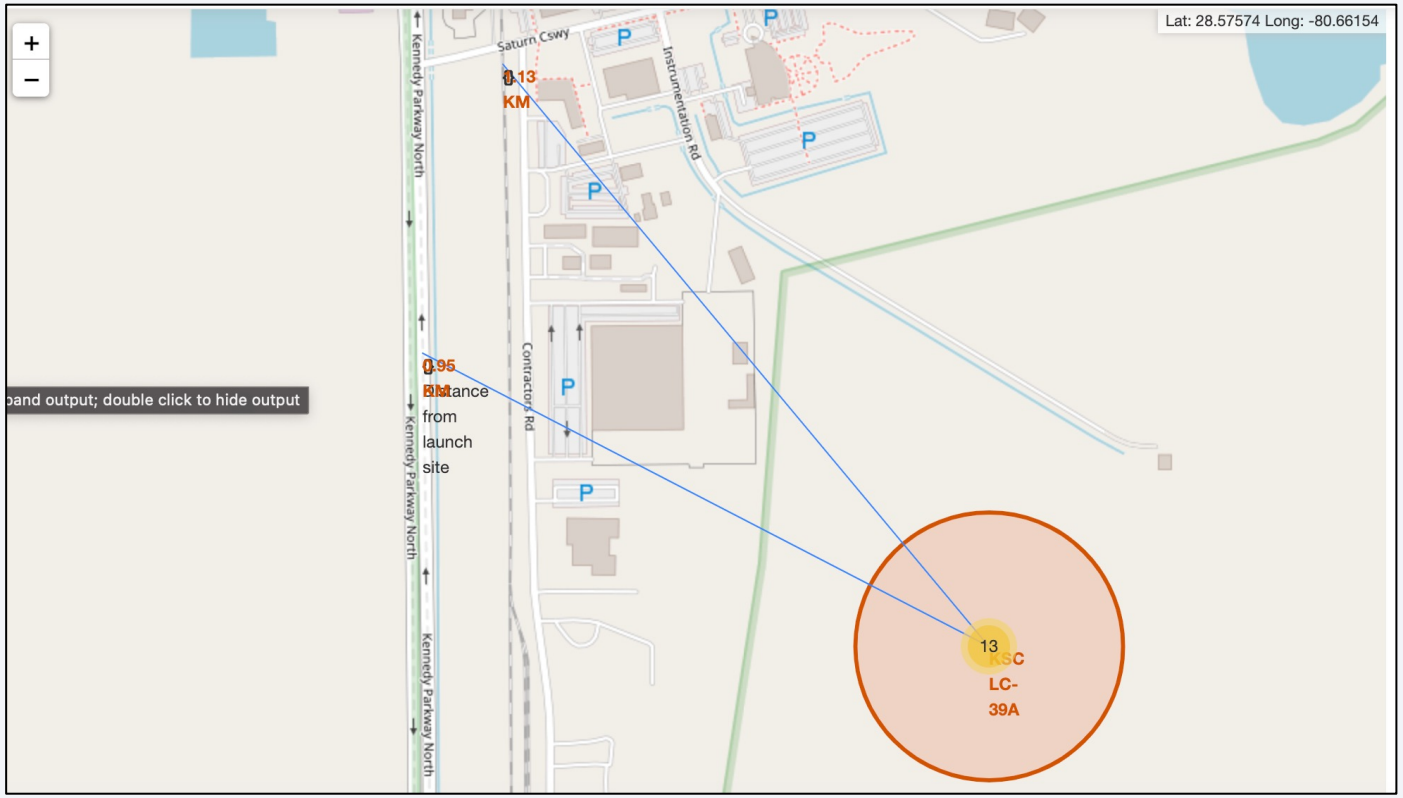




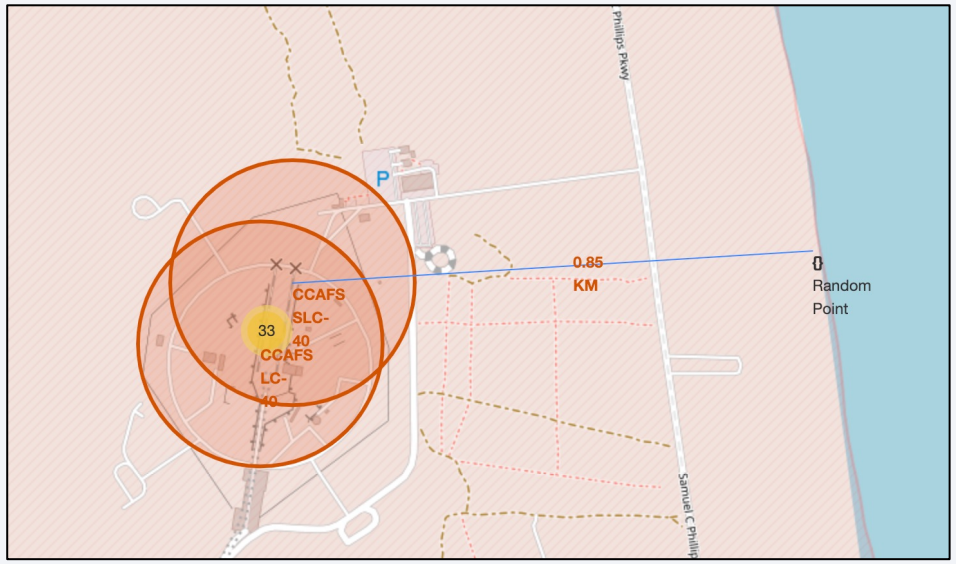
Color-labeled launch outcomes on the map: Green = success, Red = failure

Launch sites and proximity to railways, highway and coastline

Distance of launch site from railway and highway



Distance from coastline

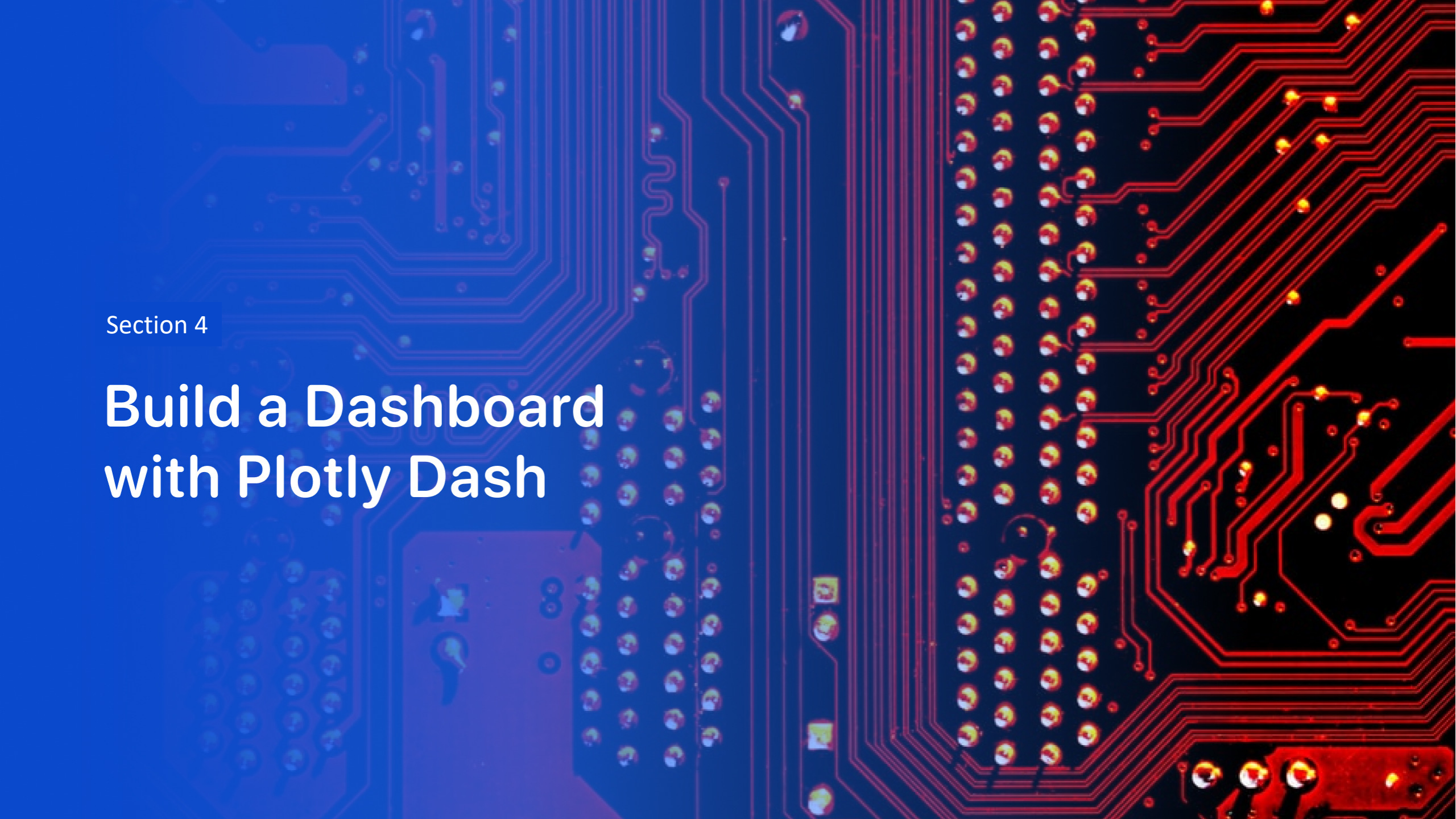


In the left the distance between launch site and close point in railway and highway is depicted

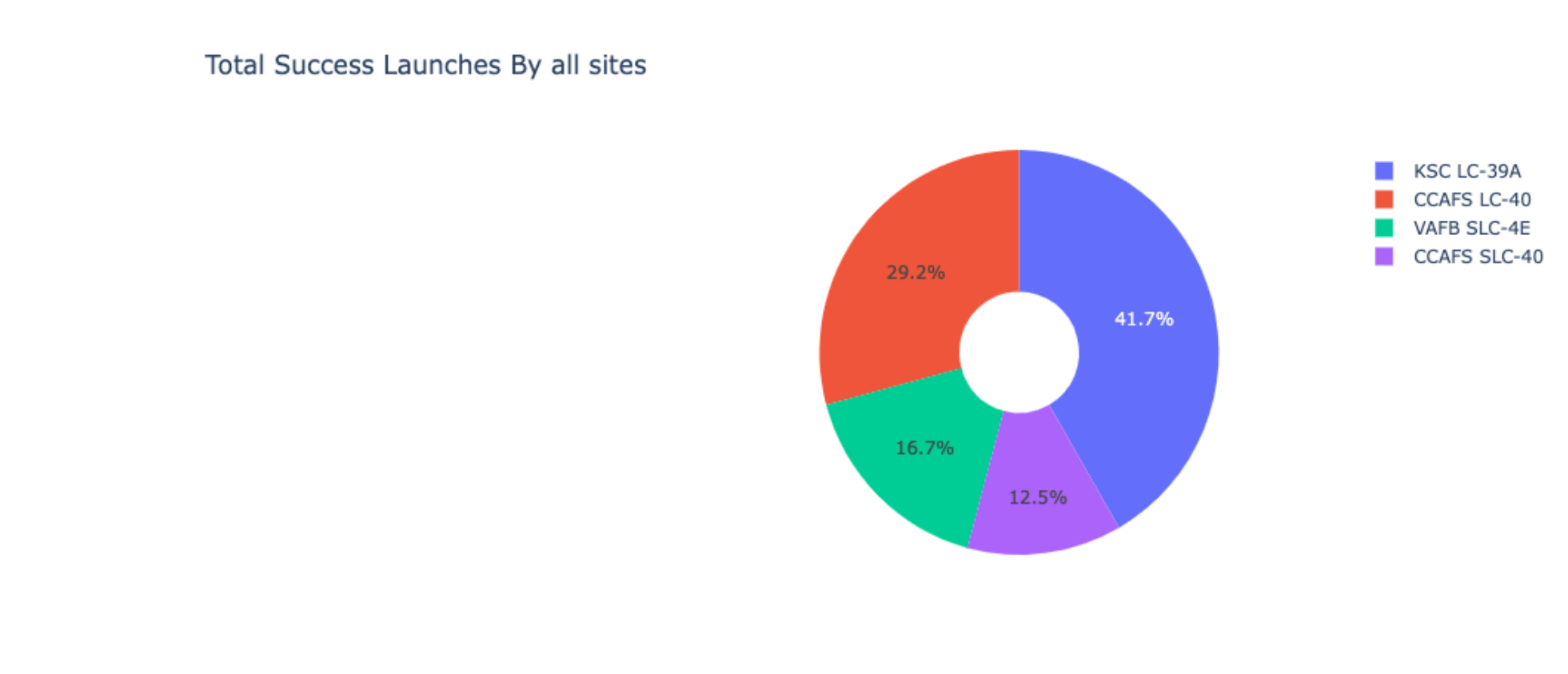
In the right the distance between launch site and closest point on the coastline is presented

Launch site and closest city

Distance of launch site from nearest city in the east coast

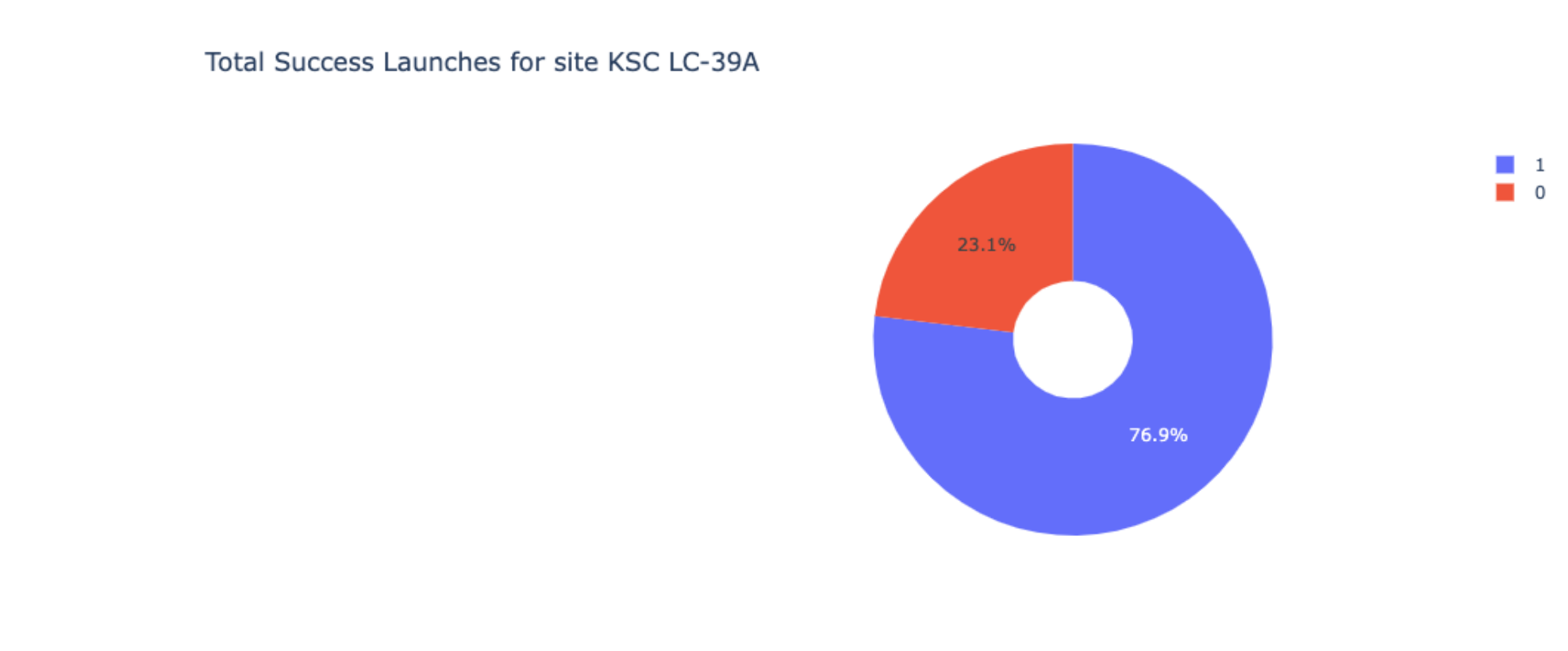

Section 4
Build a Dashboard with Plotly Dash

Launch success count rate



The pie charts shows KSC LC-39A has the highest success rate

Success percentages in the site with highest success rate



Success rate in KSC LC -39A: 0= failure, 1=Success

Payload vs. Launch Outcome scatter plot

- Payload range from 0 to 4000 [kg]

- Payload range from 4000 to 10000 [kg]

Success rate for payload range below 4000 [kg] is higher compared to above it

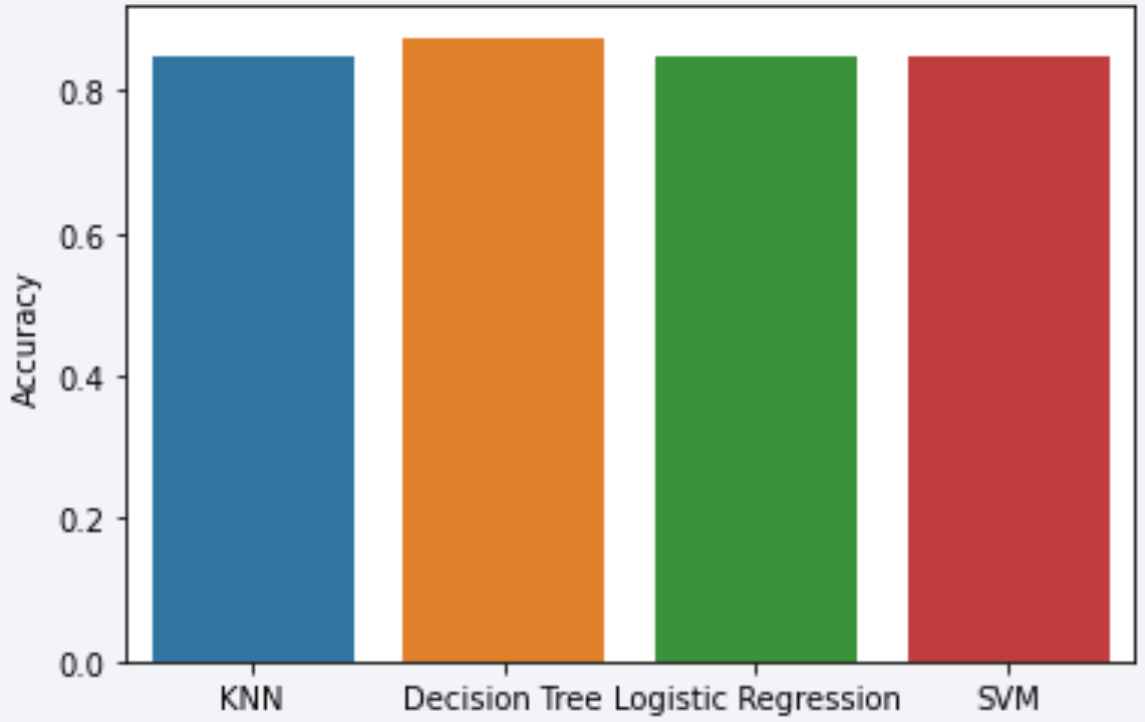
Section 5

Predictive Analysis (Classification)

Classification Accuracy

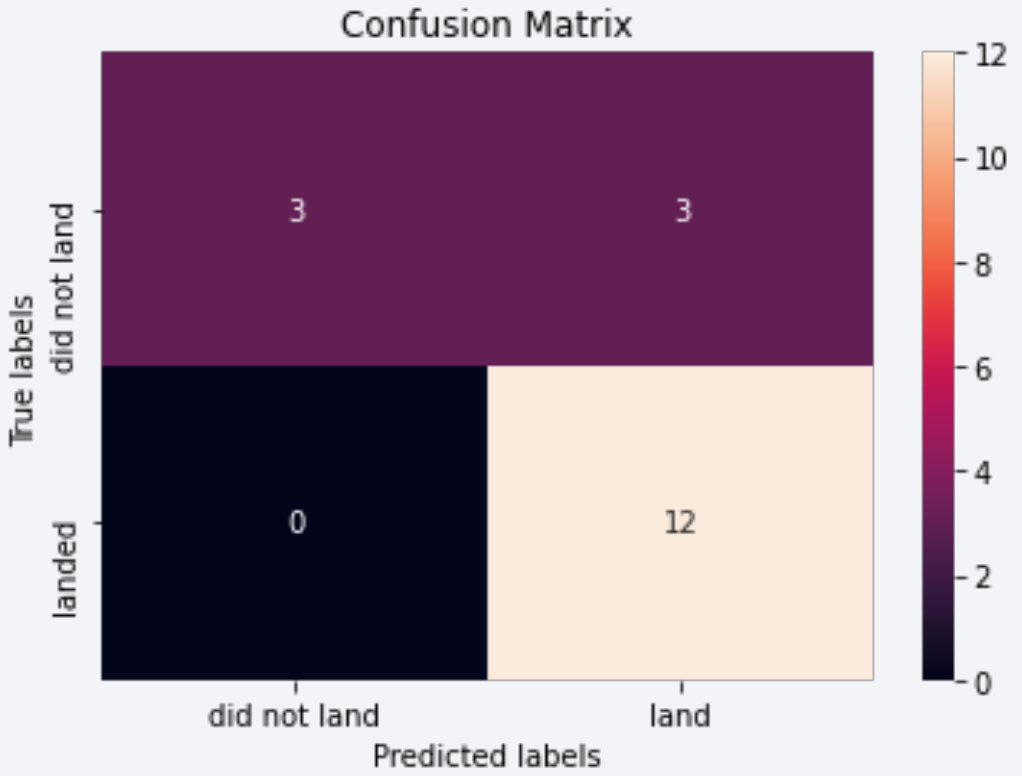
- Visualizing model accuracy for all built classification models, in a bar chart

- From chart it is clear Decision Tree has a higher performance accuracy as compared with other approaches



Confusion Matrix

- In this confusion matrix we notice that **false positive** is a **major problem**, where the ground truth says it did not land successfully but model predicts it did.



Conclusions

- Orbits ES-L1, GEO, HEO, SSO, VLEO had the most success rate.
- Launch success rate started to increase in 2013 and reached to almost 90% in 2020.
- The larger the flight amount at a launch site, the greater the success rate at a launch site.
- KSC LC-39A had the most successful launches of any sites.
- The Decision tree classifier is the best machine learning algorithm for this task.

Thank you!