IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Maximiliano Ivan Vega
30/04/2023

Outline

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

Executive Summary

- Summary of methodologies
  - Data collection using SpaceX API
  - Data Collection with Web Scrapping
  - Data Wrangling
  - Exploratory Data Analysis using SQL
  - EDA Data Visualization using Python, Pandas and Matplotlib
  - Launch Sites Analysis with Folium Interactive Visual Analytics and Ploty Dash
  - Machine Learning Landing Prediction
- Summary of all results
  - EDA Results
  - Interactive Visual Analytics & Dashboards
  - Predictive Analysis

Introduction

- Project background and context
  - SpaceX charges $62 million for Falcon 9 rocket launches, much cheaper than other providers due to the ability to reuse the first stage. Predicting if the first stage will land can determine the cost of the launch, useful for competitors bidding against SpaceX.
- Problems you want to find answers
  - Predict if the SpaceX Falcon 9 rocket first stage will land successfully

Section 1

Methodology

Methodology

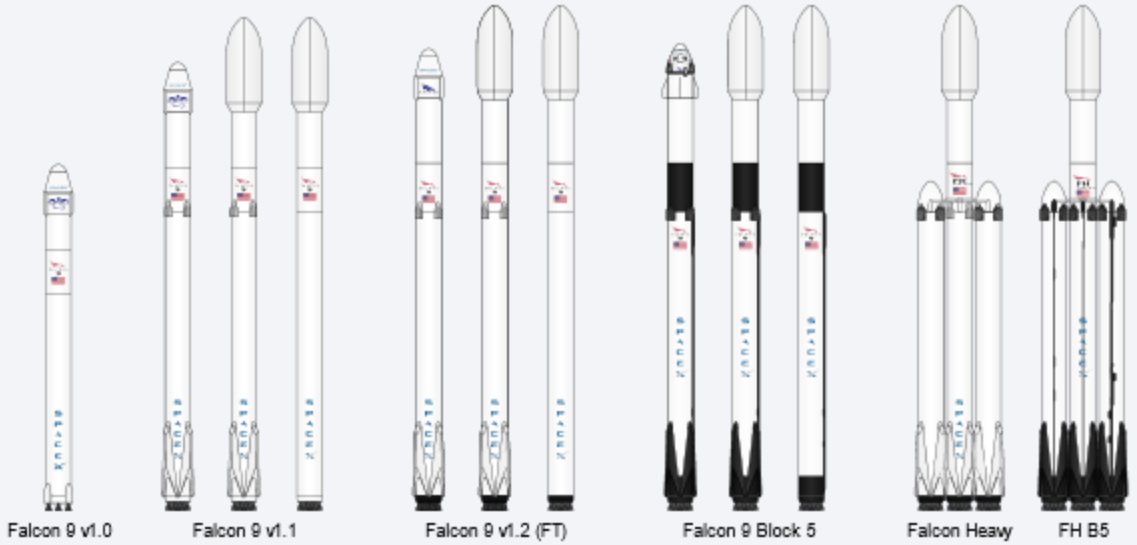
Executive Summary

- Data collection methodology:
  - SpaceX API, Web Scrapping from Wikipedia.
- Perform data wrangling
  - One-Hot Encoding of the data fields, cleaning null & invalid 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
  - LR, SVM, KNN, DT were used to choose the best classifier

Data Collection

Datasets used:

- SpaceX REST API v4
  - From the API we gathered information about the launches, launchpad, success, cores, payloads, etc.
  - We normalized this dataset into a JSON file to use it inside a dataframe
- Web Scrapping of Wikipedia
  - This is another popular data source for obtaining Falcon 9 Launch data.
  - To help us do this web scrapping me made use of BeautifulSoup



Data Collection – SpaceX API

Requesting SpaceX API

1. Request the API
2. Save the response and convert it to a JSON
3. We apply custom functions to format the data
4. Combine the columns into a dictionary to construct our dataset
5. Filter dataframe and export it to a csv file

```
spacex_url="https://api.spacexdata.com/v4/launches/past"
```

```
response = requests.get(spacex_url)
```

```
# Use json_normalize meethod to convert the json result into a dataframe
data = pd.json_normalize(response.json())
```

```
# Call getLaunchSite
getLaunchSite(data)
```

```
# Call getPayloadData
getPayloadData(data)
```

```
# Call getCoreData
getCoreData(data)
```

```
launch_dict = {'FlightNumber': list(data['flight_number']),
'Date': list(data['date']),
'BoosterVersion':BoosterVersion,
'PayloadMass':PayloadMass,
'Orbit':Orbit,
'LaunchSite':LaunchSite,
'Outcome':Outcome,
'Flights':Flights,
'GridFins':GridFins,
'Reused':Reused,
'Legs':Legs,
'LandingPad':LandingPad,
'Block':Block,
'ReusedCount':ReusedCount,
'Serial':Serial,
'Longitude': Longitude,
'Latitude': Latitude}
```

```
data_falcon9.to_csv('dataset_part_1.csv', index=False)
```

[Github Data Collection (https://github.com/powerOFMAX/SpaceX-Landing-Prediction/blob/main/1.%20Data%20Collection.ipynb)](https://github.com/powerOFMAX/SpaceX-Landing-Prediction/blob/main/1.%20Data%20Collection.ipynb)

Data Collection - Scraping

Scraping Wikipedia

1. Make the request and convert the result using BeautifulSoup
2. Find all tables and extract the columns
3. Create a dictionary and append the data previously extracted
4. Save it into a dataframe and export a csv

```
response = requests.get(static_url)
soup = BeautifulSoup(response.text, 'html.parser')
html_tables = soup.find_all('table')
```

```
column_names = []

# Apply find_all() function with `th` element on
# Iterate each th element and apply the provided
# Append the Non-empty column name (`if name is
th_elements = first_launch_table.find_all('th')

for th in th_elements:
    name = extract_column_from_header(th)
    if name is not None and len(name) > 0:
        column_names.append(name)
```

```
launch_dict= dict.fromkeys(column_names)

# Remove an irrelvant column
del launch_dict['Date and time ( )']

# Let's initial the launch_dict with each
launch_dict['Flight No.'] = []
launch_dict['Launch site'] = []
launch_dict['Payload'] = []
launch_dict['Payload mass'] = []
launch_dict['Orbit'] = []
launch_dict['Customer'] = []
launch_dict['Launch outcome'] = []
# Added some new columns
launch_dict['Version Booster']=[]
launch_dict['Booster landing']=[]
launch_dict['Date']=[]
launch_dict['Time']=[]
```

```
for table_number,table in enumerate(soup.find_all('
  # get table row
   for rows in table.find_all("tr"):
       #check to see if first table heading is as
       if rows.th:
           if rows.th.string:
               flight_number=rows.th.string.strip(
               flag=flight_number.isdigit()
       else:
           flag=False
       #get table element
       row=rows.find_all('td')
       #if it is number save cells in a dictonary
       if flag:
           extracted_row += 1
           # Flight Number value
```

```
df=pd.DataFrame(launch_dict)
df.head()
df.to_csv('spacex_web_scraped.csv', index=False)
```

[Github Web Scraping](https://github.com/powerOFMAX/SpaceX-Landing-Prediction/blob/main/2.%20Web%20scraping%20Falcon%209%20and%20Falcon%20Heavy%20Launches%20Records%20from%20Wikipedia.ipynb)

Data Wrangling

1. Identified and calculated number of missing values
2. Calculated number of launches on each site and number of occurrence of each orbit
3. Calculate the number and occurrence of mission outcome per orbit
4. Create a landing outcome label from Outcome column

```
# Apply value_counts() on column LaunchSite
df['LaunchSite'].value_counts()
```

```
CCAFS SLC 40    55
KSC LC 39A      22
VAFB SLC 4E     13
```

```
df['Orbit'].value_counts()
```

```
GTO      27
ISS      21
VLEO     14
PO        9
LEO       7
SSO       5
MEO       3
ES-L1     1
HEO       1
SO        1
GEO       1
```

```
landing_outcomes = df['Outcome'].value_counts()
landing_outcomes.head()
```

```
True ASDS      41
None None      19
True RTLS      14
False ASDS      6
True Ocean      5
```

```
landing_class = [0 if outcome in bad_outcomes else 1 for i, outcome in enumerate(df['Outcome'].values)]
# landing_class = 1 otherwise
print(landing_class)
```

```
[0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 0, 1,
0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
```

[Github Data Wrangling](https://github.com/powerOFMAX/SpaceX-Landing-Prediction/blob/main/3.%20Data%20Wrangling.jupyterlite.ipynb)
https://github.com/powerOFMAX/SpaceX-Landing-Prediction/blob/main/3.%20Data%20Wrangling.jupyterlite.ipynb

EDA with Data Visualization

Performed Data Analysis using Pandas and Matplotlib

- Exploratory Data Analysis
- Preparing Data Feature Engineering

Charts used:

- Scatter Plots to visualize the relationship between Flight Number and Launch Site, Payload and Launch Site, Flight Number and Obit type, Payload and Orbit type.
    - This type of chart make us easier to see correlation between the attributes, for all of these we separated the values in two groups, Class 0 and 1, being the last one the successful
- Used Bar chart to visualize the relationship between success rate of each orbit type
- Line Plot to visualize the launch success yearly trend

Github Data Visualization
https://github.com/powerOFMAX/SpaceX-Landing-Prediction/blob/main/5.%20Data%20Visualization%20Pandas%20and%20Matplotlib%20-%20SpaceX.ipynb

EDA with SQL

- SQL Queries performed:
  - Display the names of the unique launch sites in the space mission
  - Display 5 records where launch sites begin with the string 'CCA'
  - Display the total payload mass carried by boosters launched by NASA (CRS)
  - Display average payload mass carried by booster version F9 v1.1
  - List the date when the first successful landing outcome in ground pad was achieved.
  - List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
  - List the total number of successful and failure mission outcomes
  - List the names of the booster_versions which have carried the maximum payload mass. List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.
  - Rank the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.

Github EDA SQL
https://github.com/powerOFMAX/SpaceX-Landing-Prediction/blob/main/4.%20Space-X%20EDA%20Using%20SQL.ipynb

Build an Interactive Map with Folium

- A Folium map was created to mark all the launch sites. For this we made use of objects like markers, circles, lines to mark the success or failure of launches for each launch site.

- To represent this map we used the following set of outcomes where Failure = 0 and success = 1. This helped us see which places have high success rate.

Github Folium Map
https://github.com/powerOFMAX/SpaceX-Landing-Prediction/blob/main/6.%20Launch%20Sites%20Locations%20Analysis%20with%20Folium-Interactive%20Visual%20Analytics.ipynb

Build a Dashboard with Plotly Dash

- Built an interactive dashboard app with Plotly dash:
  - Added a dropdown input component for the Launch Site.
  - Added a callback function to render a success-pie-chart based on selected site dropdown.
  - Added a Range Slider to select the Payload
  - Added a callback function to render the success-payload-scatter-chart scatter plot.
- We could filter the information using the dropdown to look all the sites or a specific one.

Github SpaceX DashApp
https://github.com/powerOFMAX/SpaceX-Landing-Prediction/blob/main/7.%20Build%20an%20Interactive%20Dashboard%20with%20Ploty%20Dash%20-%20spacex_dash_app.py

Predictive Analysis (Classification)

To find the best classifier for this problem we explored different solutions like, SVM, Classification Trees, K – Nearest Neighbors and Logistic Regression.

1. We created an object for each of the algorithms then created a GridSearchCV and assigned them a set of parameters for each model
2. The GridSearchCV Object was created with CV=10 for all the models, then fit the training data into the GridSearch object for each to find the best Hyperparameter.
3. After fitting the training set, we output GridSearchCV object for each of the models then displayed the best parameters using the data attribute **best_params_** and the accuracy on the validation data using the data attribute **best_score_**
4. Lastly, we used the method score to calculate the accuracy on the test data for each model and plotted a confusion matrix using the test and predicted outcomes

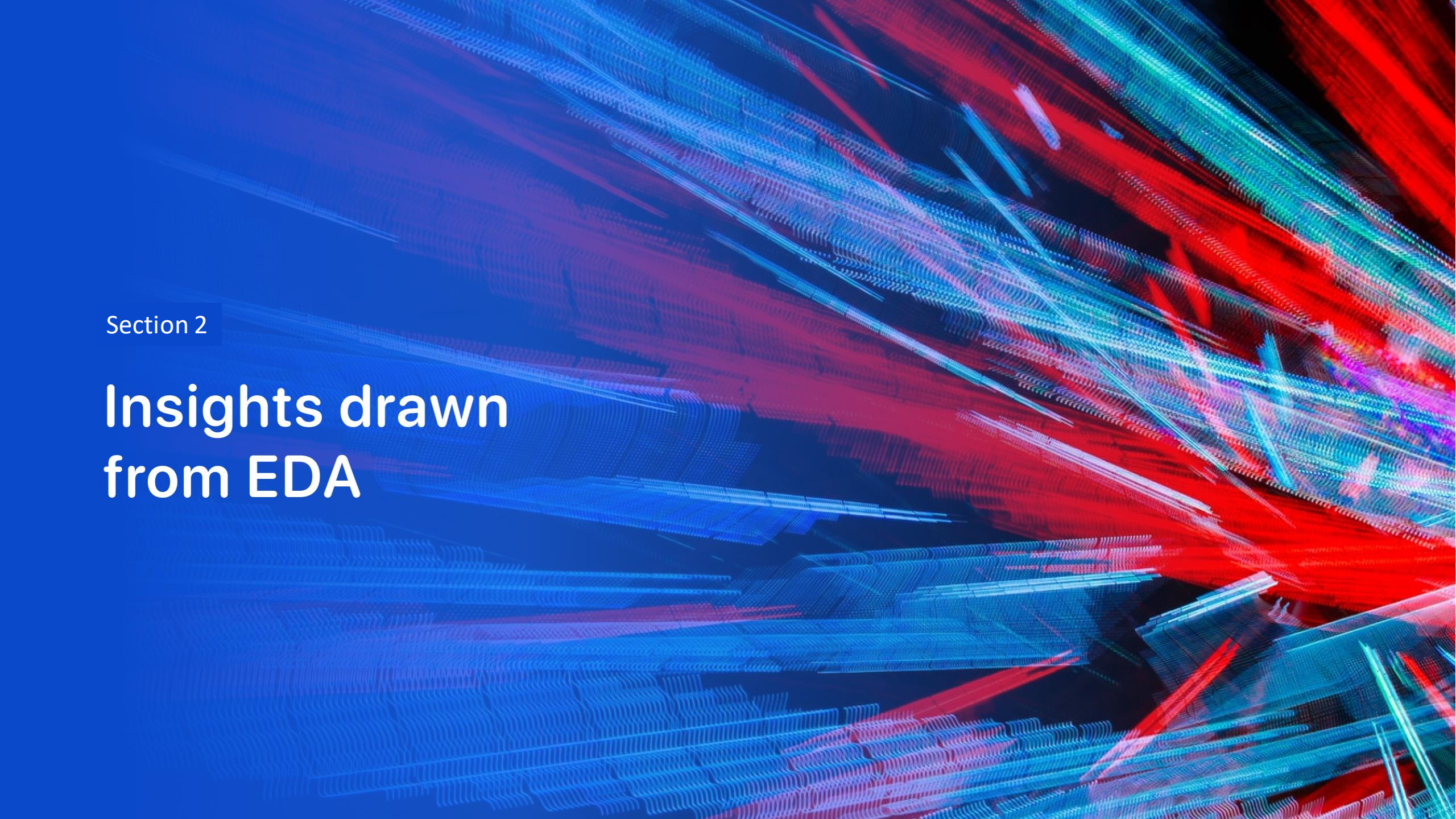
Github Prediction
https://github.com/powerOFMAX/SpaceX-Landing-Prediction/blob/main/8.%20Machine%20Learning%20Prediction.ipynb

Results

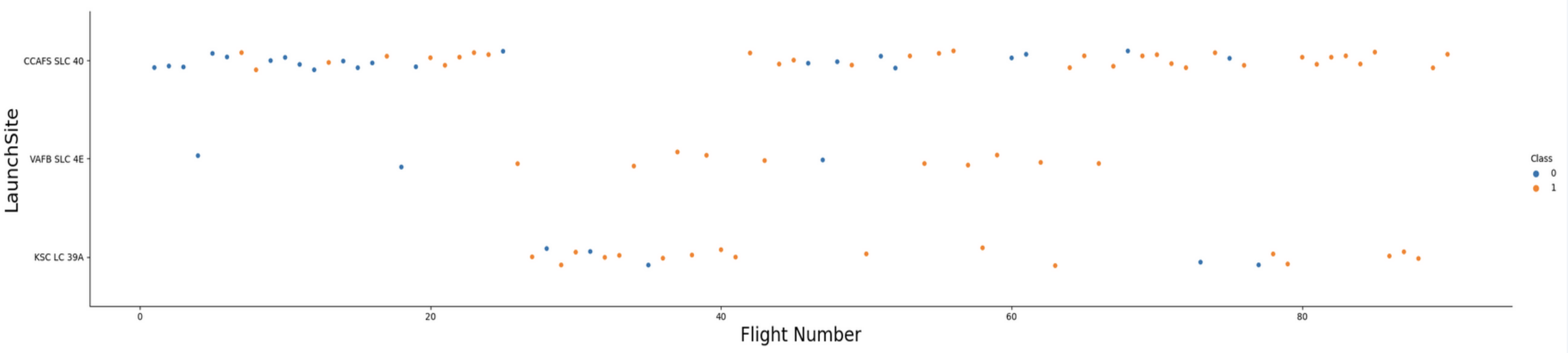
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

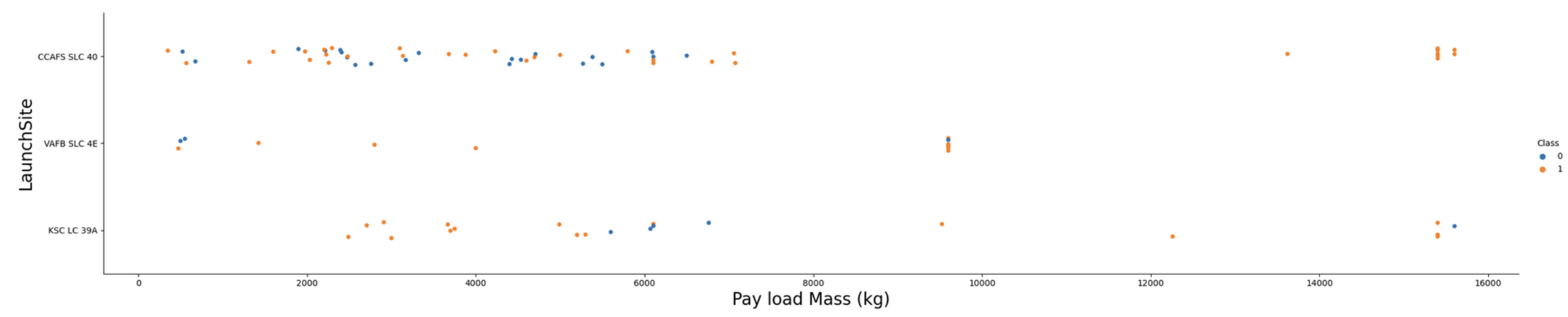


Flight Number vs. Launch Site



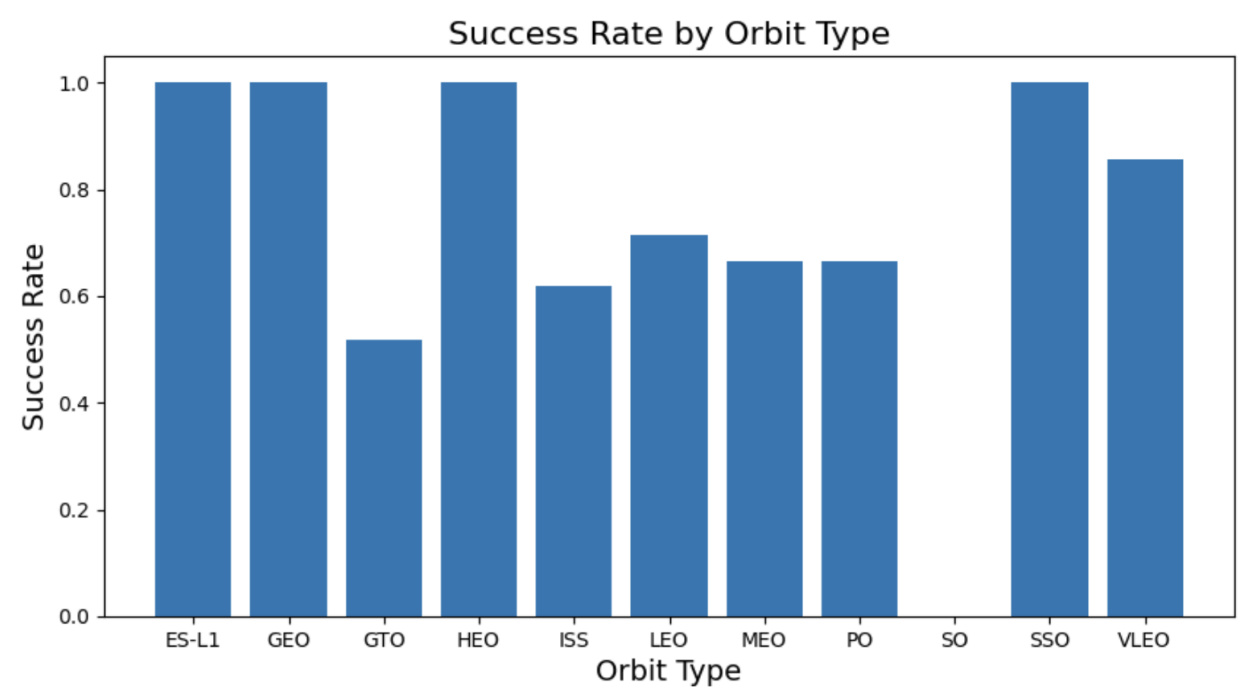
We can observe that "**CCAFS-SLC-40**" had the biggest number of flights, followed by "**KSC-LC-39A**" and that starting from flight number 60 the payload mass increased and kept constant, which could have a correlation with **CCAFS-SLC-40**

Payload vs. Launch Site



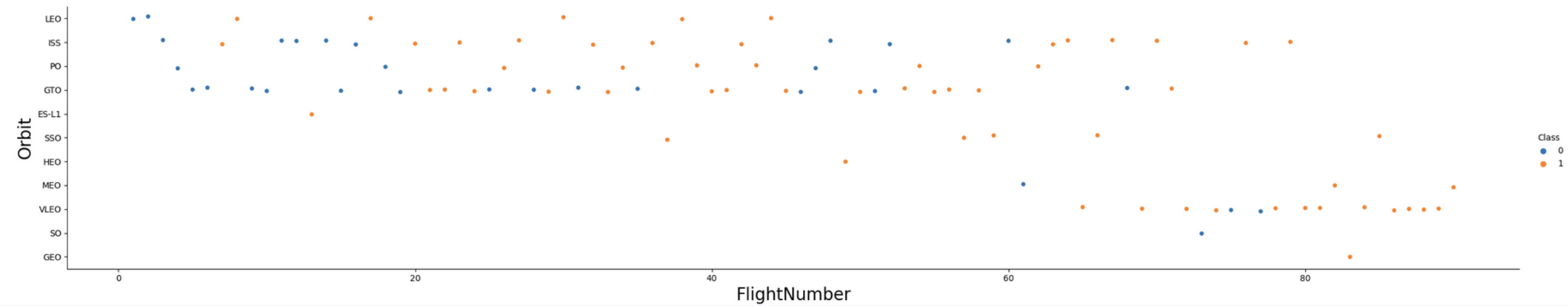
The majority of the payload with lower mass have been launched from **CCAFS SLC 40**. Also for **VAFB-SLC** launch site there are no rockets launched for heavy payload mass (greater than 10000).

Success Rate vs. Orbit Type



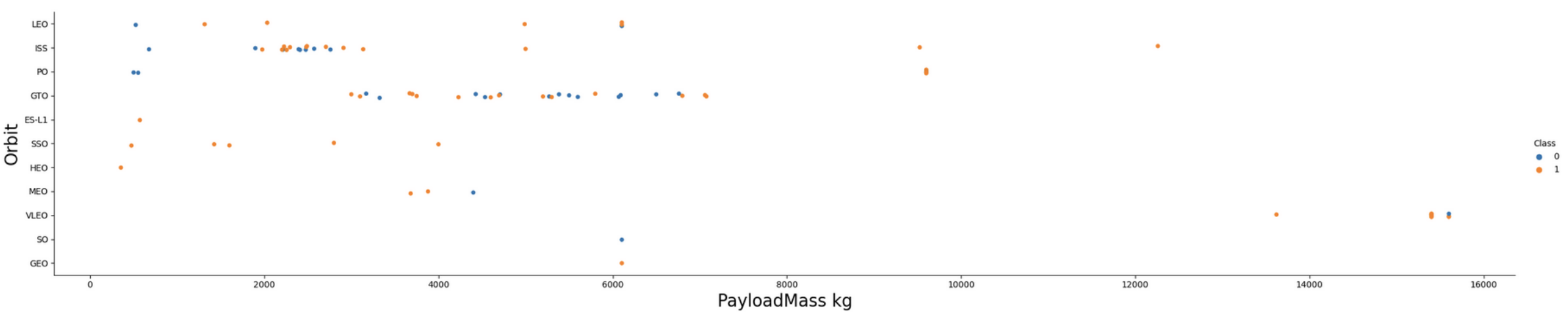
- The Orbit types ESL1, GEO, HEO and SSO are the ones with the highest success rate.

Flight Number vs. Orbit Type



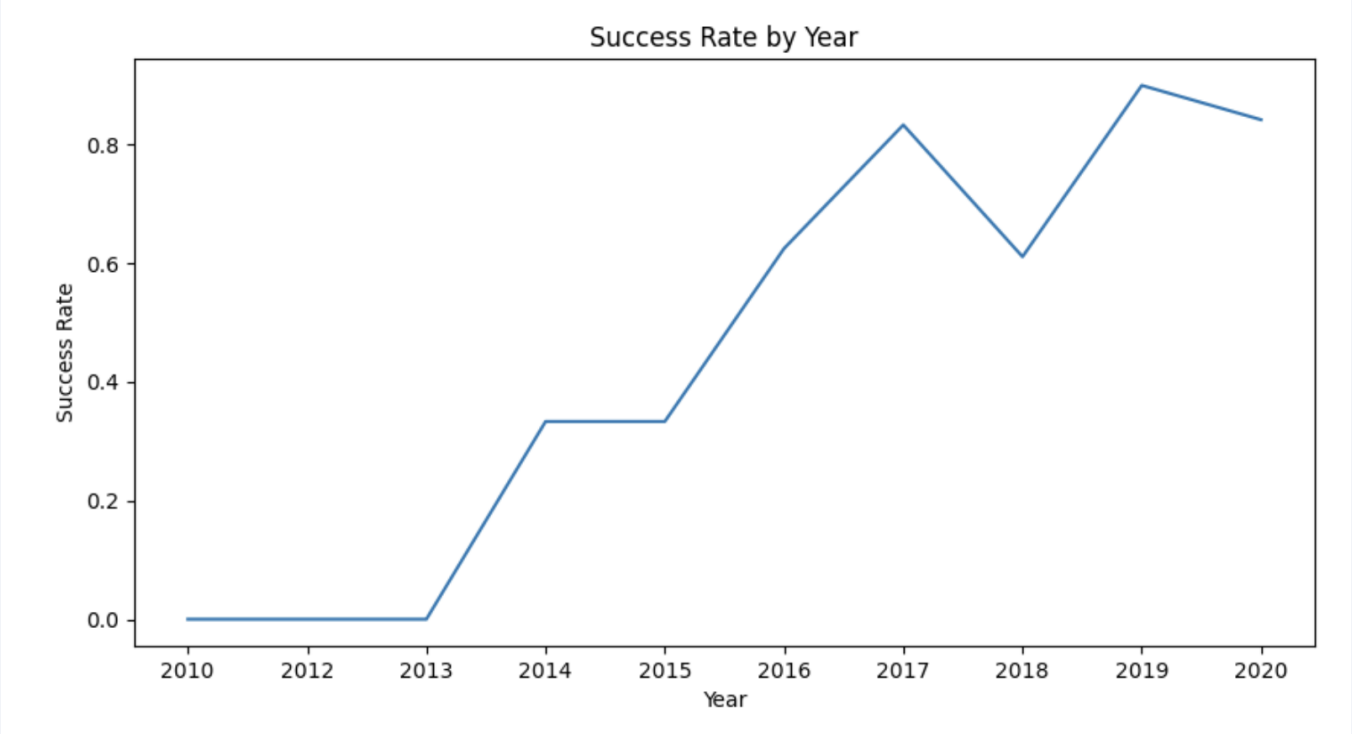
- LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit

Payload vs. Orbit Type



- With heavy payloads the successful landing or positive landing rate are more for Polar, LEO and ISS.
- For GTO we cannot distinguish this well as both positive landing rate and negative landing (unsuccessful mission) are both there here.

Launch Success Yearly Trend



- We can observe that the success rate since 2013 kept increasing till 2020

All Launch Site Names

We Queried the names of unique launch sites using a DISTINCT statement

```
%sql SELECT DISTINCT "Launch_Site" FROM "SPACEXTBL";
```

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

Launch Site Names Begin with 'CCA'

```
%sql 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	
04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	(
08-12-2010	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	(
22-05-2012	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	N
08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	N
01-03-2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	N

- Find 5 records where launch sites begin with `CCA`

Total Payload Mass

- Calculate the total payload carried by boosters from NASA

```
%%sql
SELECT SUM(PAYLOAD_MASS_KG_) as "Total Payload Mass (Kg)", Customer
FROM 'SPACEXTBL'
WHERE Customer = 'NASA (CRS)';
```

Total Payload Mass (Kg)	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_") FROM "SPACEXTBL" WHERE "Booster_Version" = 'F9 v1.1';
```

AVG("PAYLOAD_MASS__KG_")
2928.4

First Successful Ground Landing Date

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

MIN("Date")
22-12-2015

```
%sql SELECT MIN("Date") FROM "SPACEXTBL" WHERE "Landing _Outcome" = "Success (ground pad)";
```

Successful Drone Ship Landing with Payload between 4000 and 6000

- List of 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" FROM "SPACEXTBL" WHERE "Landing
_Outcome" = "Success (drone ship)" and "PAYLOAD_MASS__KG_" < 4000
and "PAYLOAD_MASS__KG_" < 6000;
```

Booster_Version
F9 FT B1021.1
F9 FT B1023.1
F9 FT B1029.2
F9 FT B1038.1
F9 B4 B1042.1
F9 B4 B1045.1
F9 B5 B1046.1

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes

Successful	Failure
98	3

```
%%sql
SELECT (SELECT count(*) FROM "SPACEXTBL" WHERE "Mission_Outcome" = "Success") as "Successful",
        (SELECT count(*) FROM "SPACEXTBL" WHERE "Mission_Outcome" != "Success") as "Failure";
```

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass

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

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

2015 Launch Records

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

```
%%sql
SELECT
  substr("Date", 4, 2) as Date,
  "Landing _Outcome",
  "Booster_Version",
  "Launch_Site"
FROM "SPACEXTBL"
WHERE substr("Date",7,4) = '2015'
AND "Landing _Outcome" = 'Failure (drone ship)';
```

Date	Landing _Outcome	Booster_Version	Launch_Site
01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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

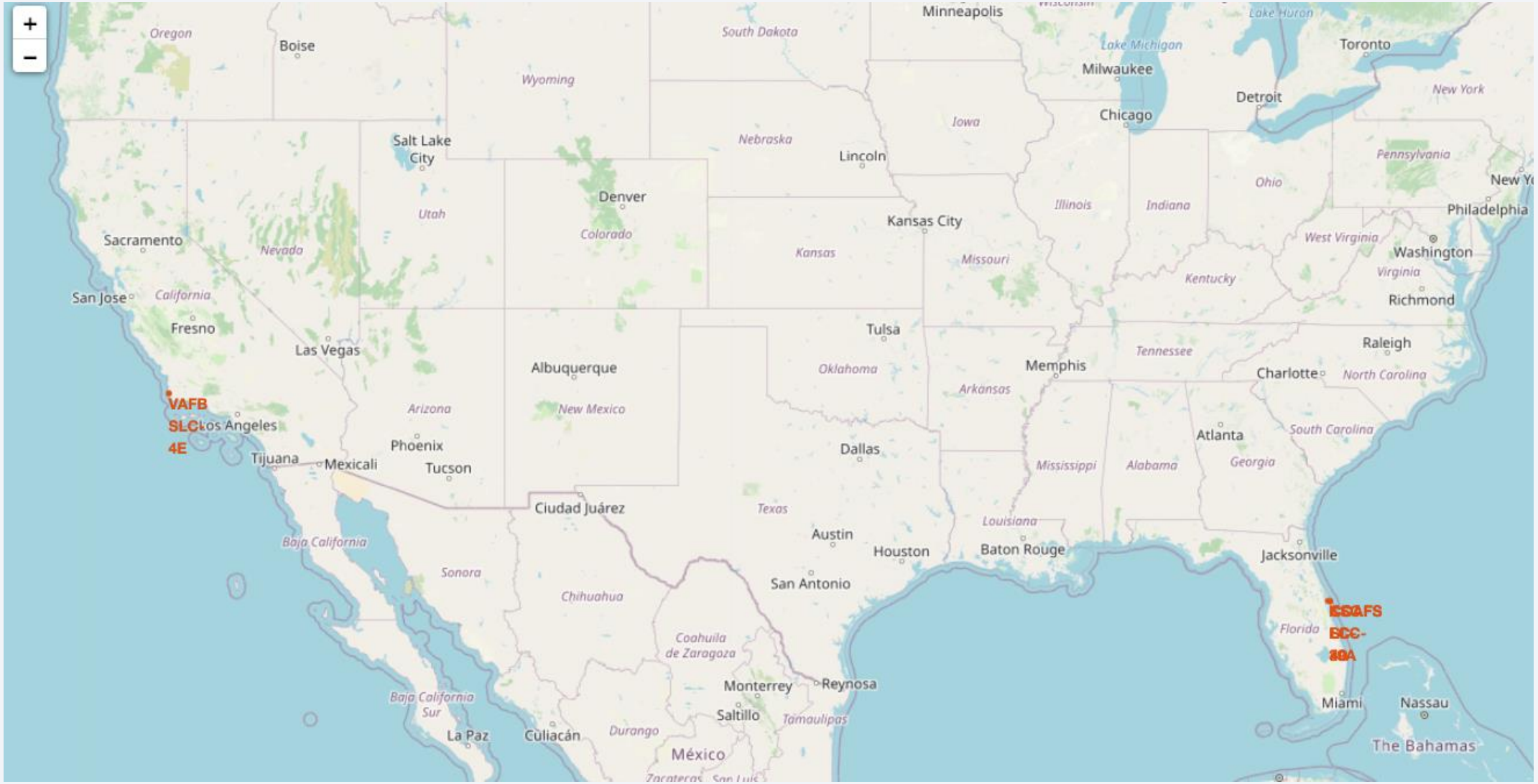
- Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order

```
%%sql
SELECT "Landing _Outcome", count (*) as "successful_landings_count", RANK() OVER (ORDER BY COUNT(*) DESC) AS rank
FROM "SPACEXTBL"
WHERE "Landing _Outcome" like 'Success%'
AND "Date" >= '04-06-2010' AND "Date" <= '20-03-2017'
GROUP BY "Landing _Outcome"
```

Landing _Outcome	successful_landings_count	rank
Success	20	1
Success (drone ship)	8	2
Success (ground pad)	6	3

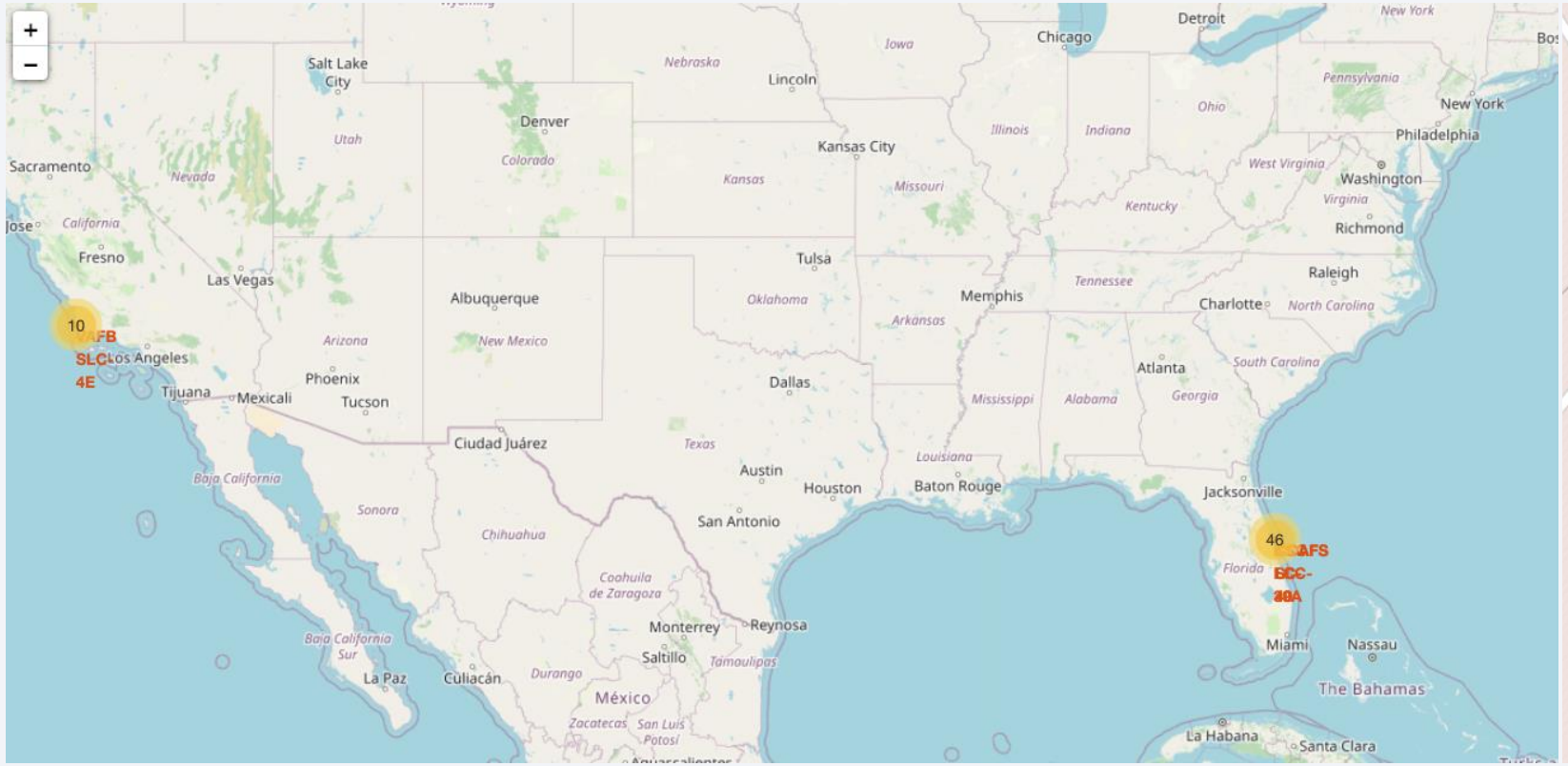

Section 3
Launch Sites
Proximities Analysis

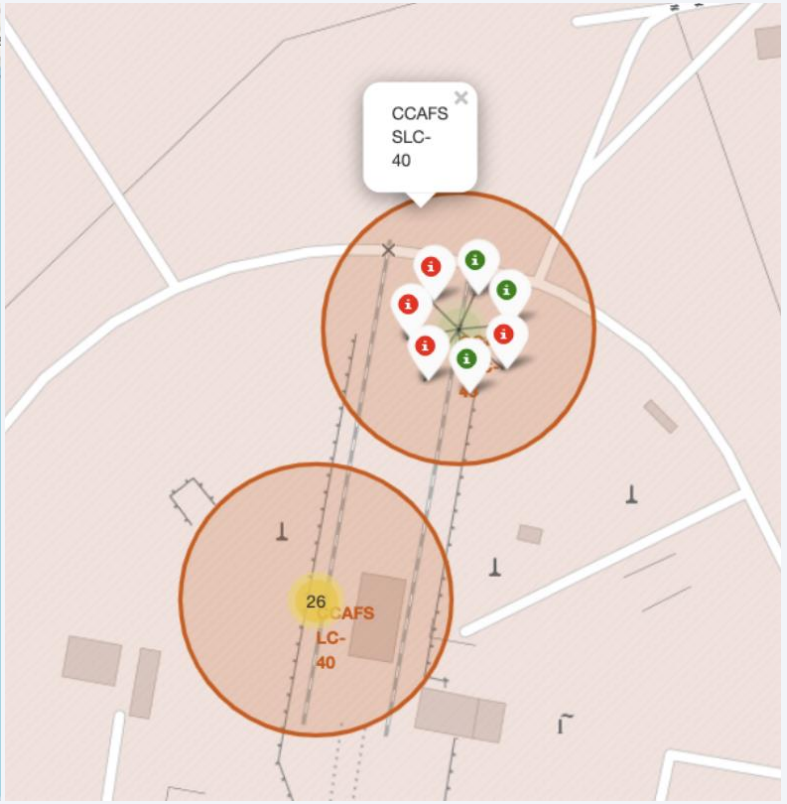
SpaceX Launch Sites



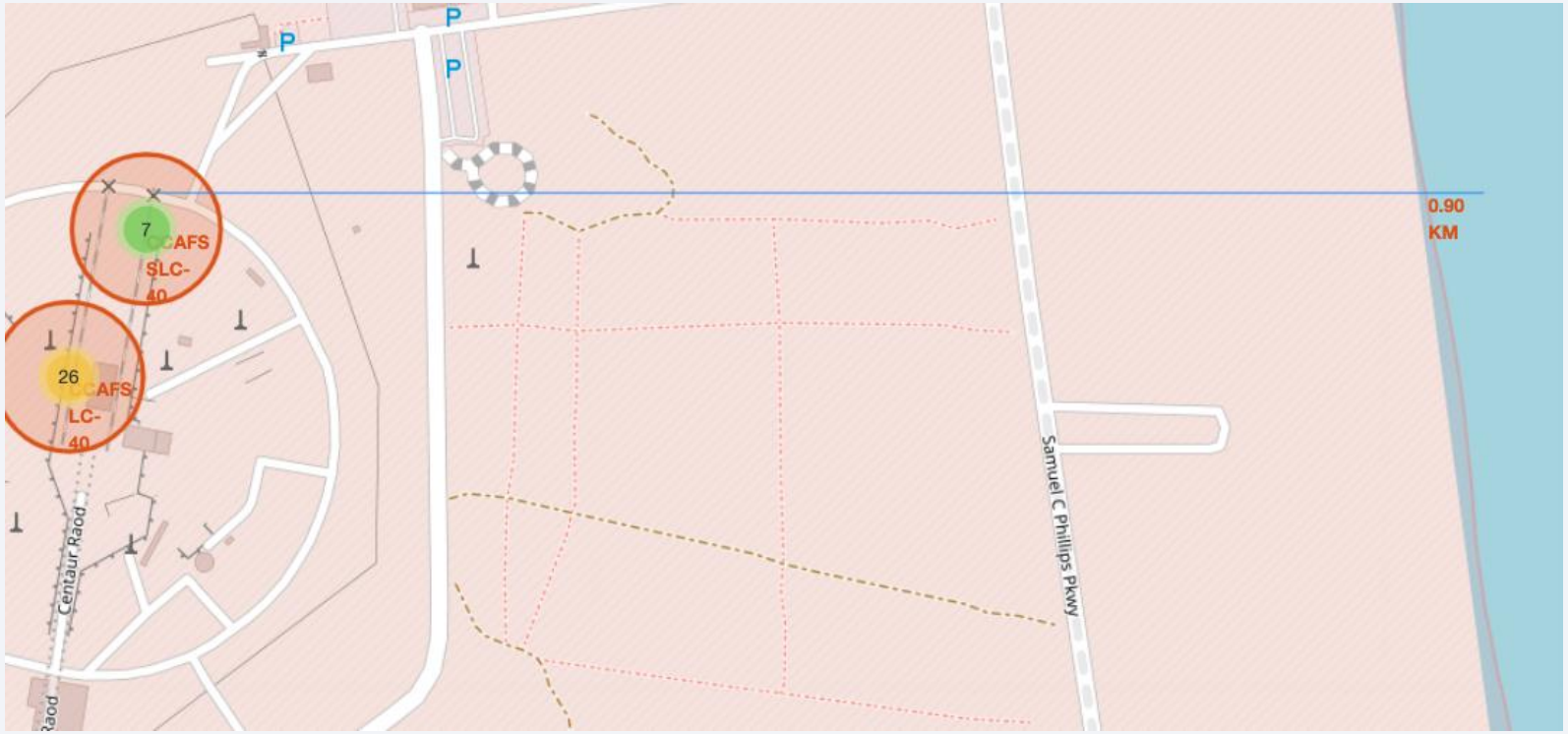
The launch sites are close to the coast and relatively close to the Equator

Launch outcomes using Color Markers

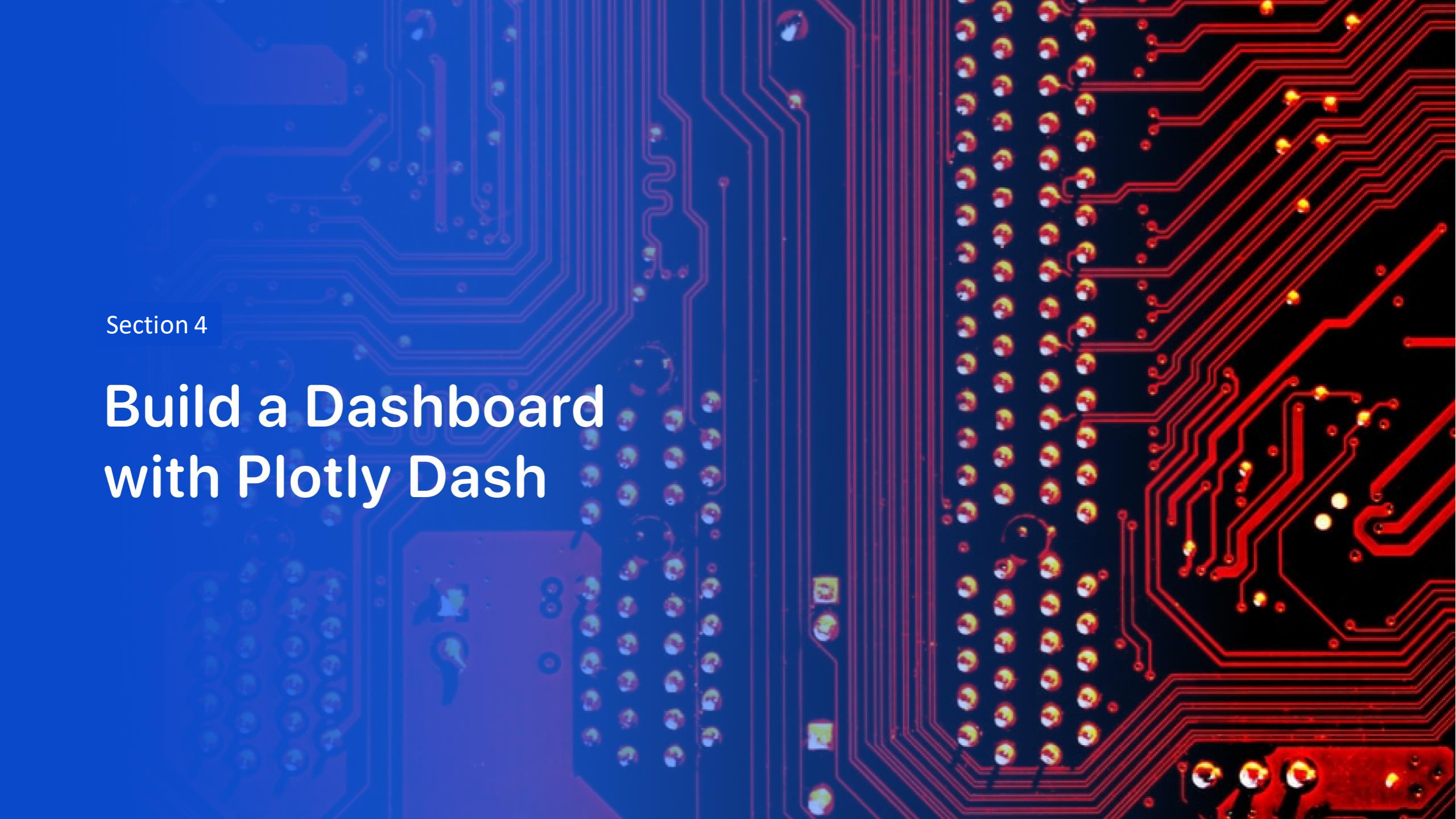




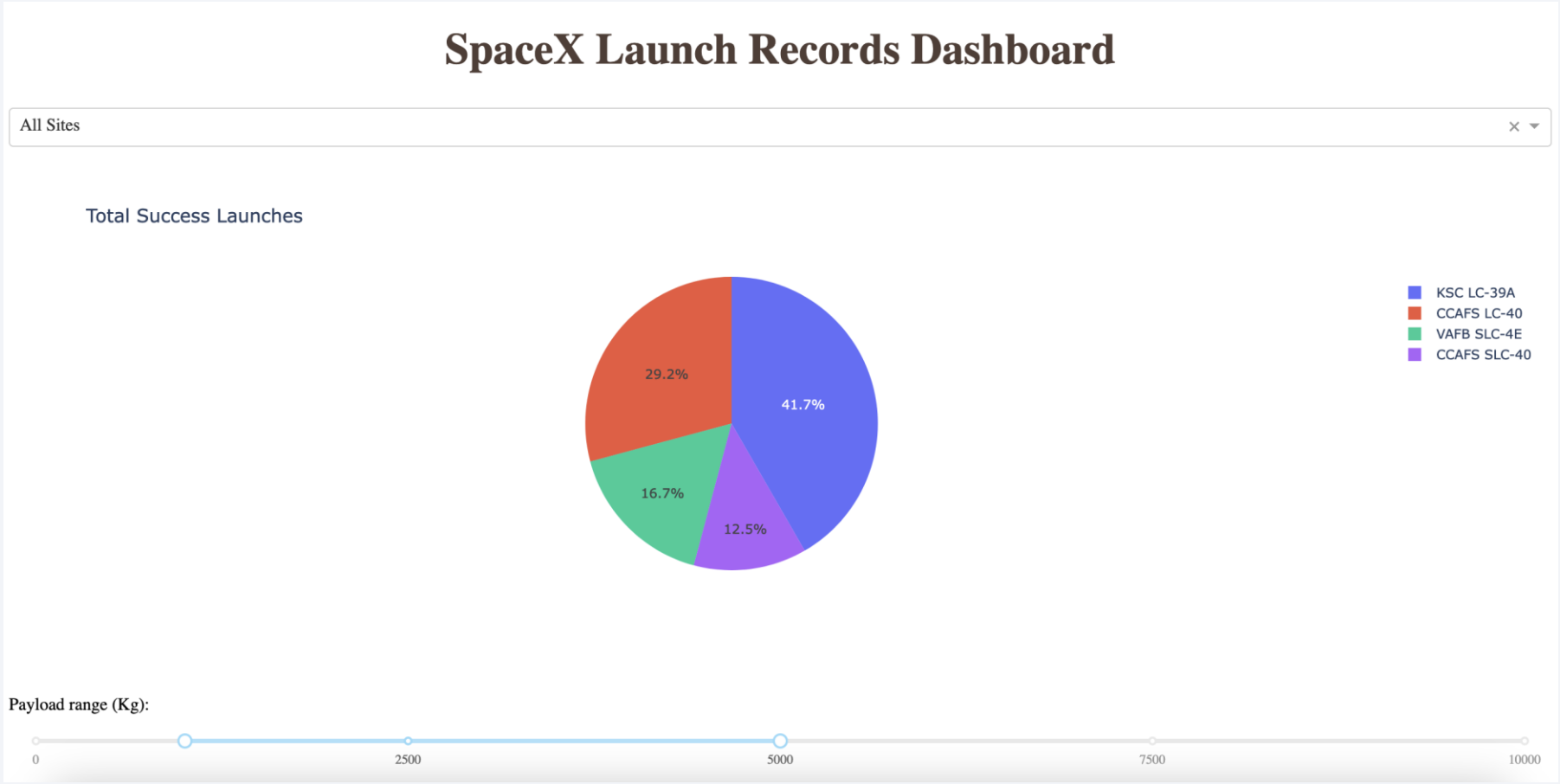
Launch site surroundings



The distance between the Launch site CCAFS SLC-40 to the coast is around 0.90km

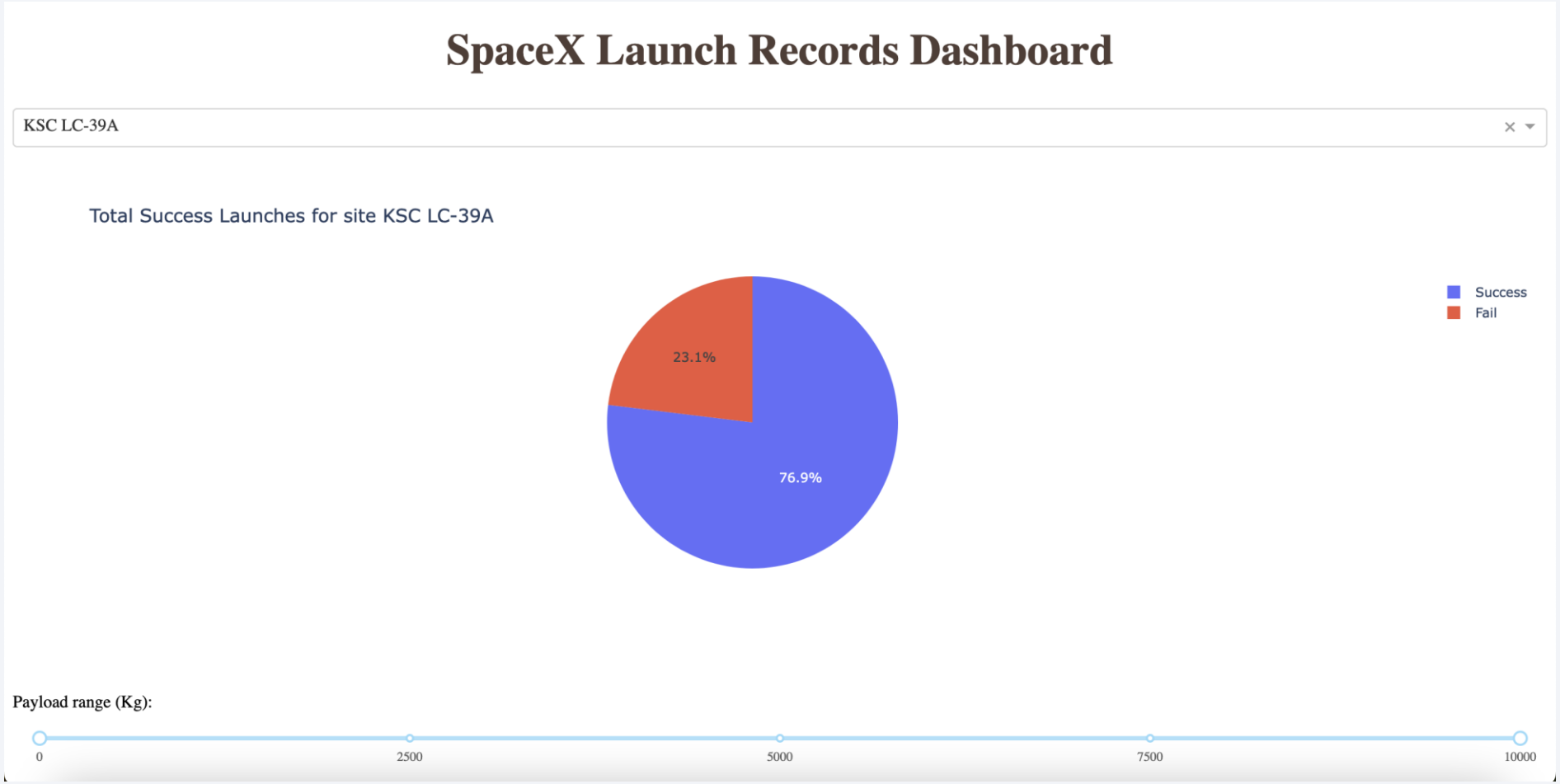

Section 4
Build a Dashboard with Plotly Dash

SpaceX Launch Success



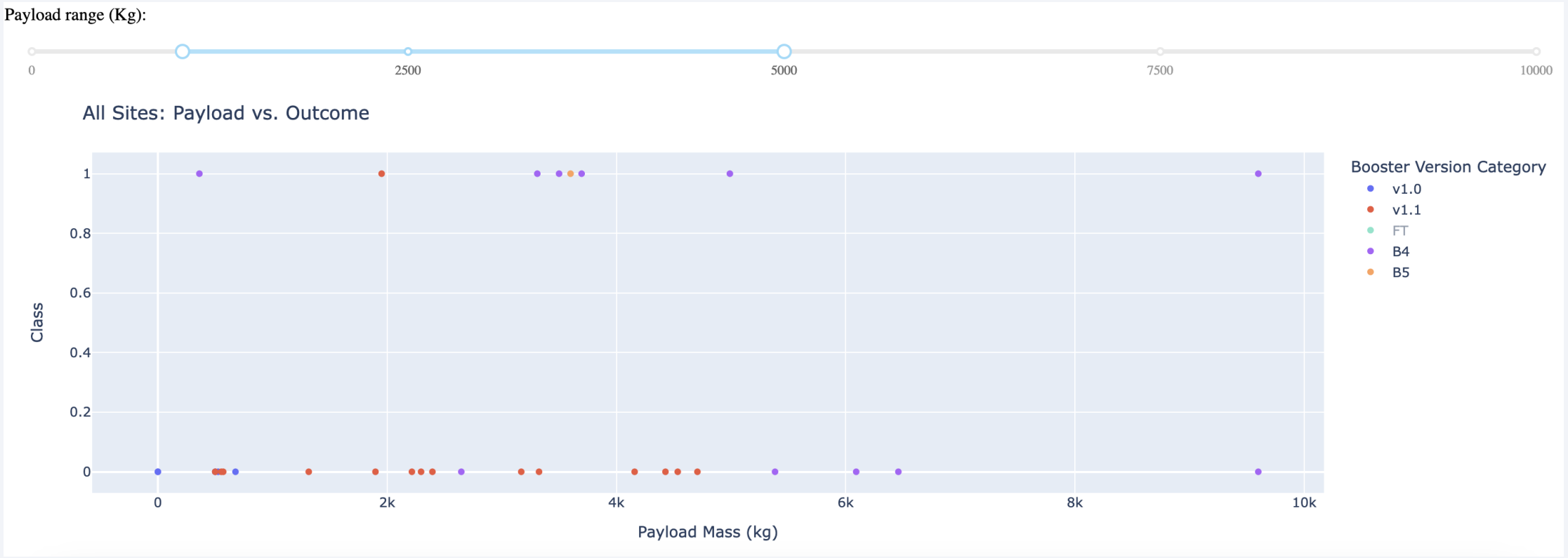
- Launch site **KSC LC 39A** has the highest launch success rate with almost 42% followed by **CCAFS LC-40** with 29%.
- The Launch site with less success rate is **CCAFS SLC-40** with 12.5%

Site with highest launch success ratio



- As we can observe we have almost 77% of success vs 23%

Payload vs Outcome for all the sites



Section 5

Predictive Analysis (Classification)

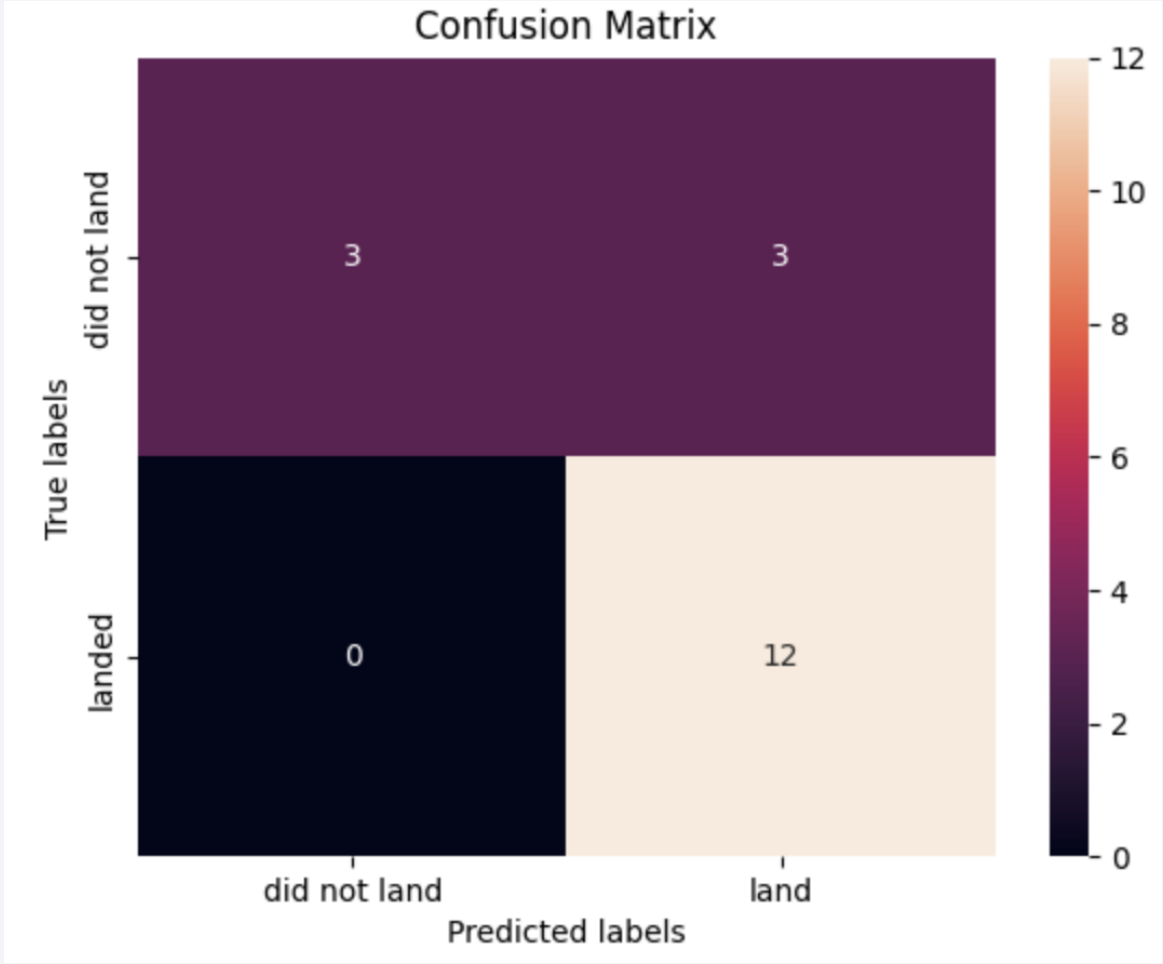
Classification Accuracy

- Using the Score value we observed no difference in the accuracy on the test data.
- We observed differences in best_score on each
- DT: 0.8875
- LG: 0.8464
- SVM: 0.8482
- KNN: 0.8482

	0
Method	Test Data Accuracy
Logistic_Reg	0.833333
SVM	0.833333
Decision Tree	0.833333
KNN	0.833333

Confusion Matrix

- All of the 4 classification models had the same Matrix.
- The main problem observed was the number of false positive for all the models



Conclusions

- The success rates of launches can vary depending on the site used. From the total successes, KSC LC 39A has the highest launch success rate with almost 42% followed by CCAFS LC-40 with 29%.
- Since 2013 we observed that the success rate kept increasing till 2020.
- Orbits GEO, HEO, SSO, and ES L1 have the highest success rates
- Low-weighted payloads perform better than heavier payloads but this can change.
- All the models used in this analysis resulted in similar results the only difference that gave better results with best_score was the decision tree model, nevertheless, the other models should be able to provide similar quality predictions

Appendix

Find all the code and examples shown in this presentation in the following git repository:

https://github.com/powerOFMAX/SpaceX-Landing-Prediction/tree/main

Contact information:

- Twitter: https://twitter.com/_maxi_vega
- LinkedIn https://www.linkedin.com/

Thank you!