IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Heber Antony
September 29, 2023

Outline

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

Executive Summary

- Summary of methodologies
  - SpaceX Data Collection using SpaceX API
  - SpaceX Data Collection with Web Scraping
  - SpaceX Data Wrangling
  - SpaceX Exploratory Data Analysis using SQL
  - SpaceX EDA Data Visualization using Python Pandas and Matplotlib
  - SpaceX Launch Sites Analysis with Folium
  - Interactive Visual Analytics and Plotly Dash
- Summary of all results
  - EDA results
  - Interactive Visual Analytics and Dashboards
  - Predictive Analysis

Introduction

- Project background and context
  - SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars
  - Other providers cost upwards of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage
  - Therefore, if we can determine if the first stage will land, we can determine the cost of a launch.
- Problems you want to find answers
  - In this project, we will predict if the Falcon 9 first stage will land successfully using data from Falcon 9 rocket launches advertised on its website.

Section 1

Methodology

Methodology

Executive Summary

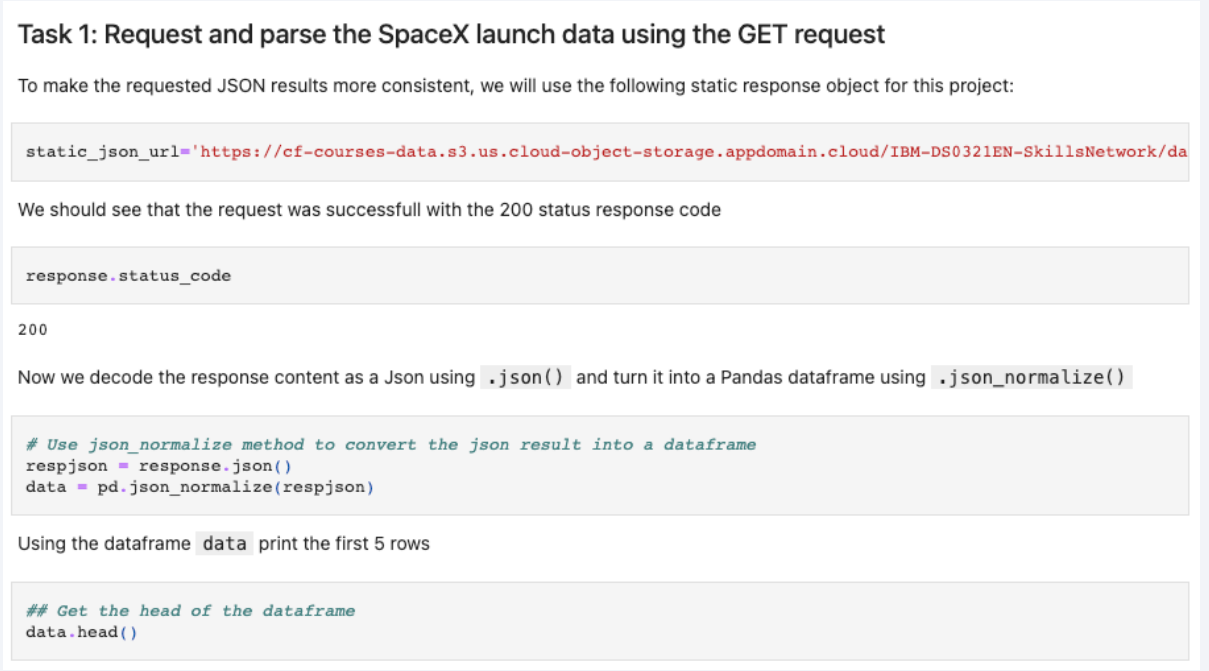
- Data collection methodology:
  - Describe how data was collected
- Perform data wrangling
  - Describe how data was processed
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection

- SpaceX Falcon 9 data collection involved utilizing the SpaceX API through custom helper functions, enabling the extraction of launch details using unique identification numbers. The API responses were decoded and transformed into a Pandas Data frame for consistent analysis.
- Additionally, historical launch records were gathered from a Wikipedia page titled "List of Falcon 9 and Falcon Heavy launches." Utilizing web scraping techniques with BeautifulSoup and request libraries, the Falcon 9 launch data was extracted, parsed, and converted into a Pandas DataFrame, enhancing the dataset for our analysis.

Data Collection – SpaceX API

- Data collection involved utilizing the SpaceX API, a RESTful interface. Initially, a GET request was made to fetch the SpaceX launch data. The retrieved JSON response was parsed and decoded, then transformed into a Pandas Data frame for comprehensive analysis.
- Here is the GitHub URL of the completed SpaceX API calls notebook: https://github.com/hebera0/SpaceX-Capstone-Project/blob/main/1.%20SpaceX%20Data%20Collection%20APIi.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:

```
static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/da
```

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

```
response.status_code
```

```
200
```

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

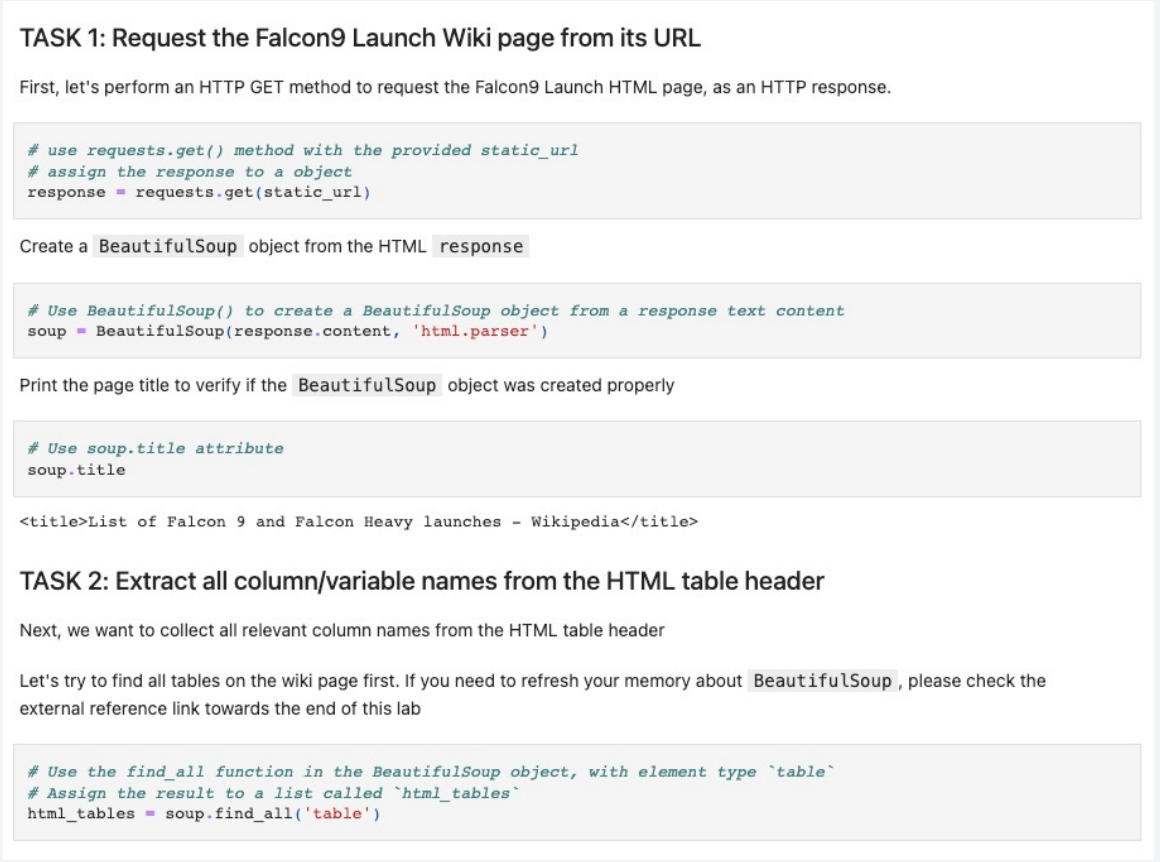
```
# Use json_normalize method to convert the json result into a dataframe
respjson = response.json()
data = pd.json_normalize(respjson)
```

Using the dataframe `data` print the first 5 rows

```
## Get the head of the dataframe
data.head()
```

Data Collection - Scraping

- Conducted web scraping to gather historical Falcon 9 launch records from a Wikipedia page. Utilized BeautifulSoup and requests to extract launch details from the HTML table on the Wikipedia page. The extracted information was parsed and organized into a Data frame for further analysis.
- Here is the GitHub URL of the completed SpaceX API calls notebook: https://github.com/hebera0/SpaceX-Capstone-Project/blob/main/2.%20SpaceX%20Web%20scraping%20from%20Wikipedia.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.

```
# use requests.get() method with the provided static_url
# assign the response to a object
response = requests.get(static_url)
```

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

```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(response.content, 'html.parser')
```

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

```
# Use soup.title attribute
soup.title
```

```
<title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>
```

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

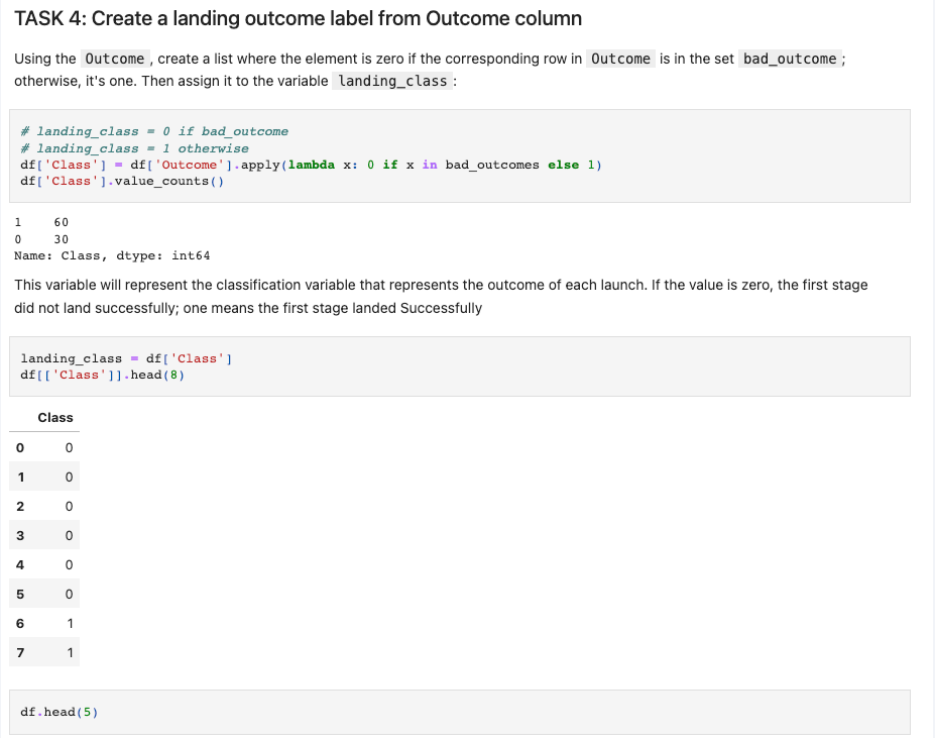
Next, we want to collect all relevant column names from the HTML table header

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

```
# 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')
```

Data Wrangling

- Upon acquiring and assembling the data into a Pandas Data frame, filtering was performed using the Booster Version column to retain only Falcon 9 launches. Subsequently, missing data in the Landing Pad and Payload Mass columns were addressed. For Payload Mass, the missing values were substituted with the mean column value.
- Additionally, Exploratory Data Analysis (EDA) was conducted to uncover patterns within the data and identify suitable labels for training supervised models.
- Here is the GitHub URL of the completed data wrangling process



TASK 4: Create a landing outcome label from Outcome column

Using the `Outcome`, create a list where the element is zero if the corresponding row in `Outcome` is in the set `bad_outcome`; otherwise, it's one. Then assign it to the variable `landing_class`:

```
# landing_class = 0 if bad_outcome
# landing_class = 1 otherwise
df['Class'] = df['Outcome'].apply(lambda x: 0 if x in bad_outcomes else 1)
df['Class'].value_counts()
```

```
1    60
0    30
Name: Class, dtype: int64
```

This variable will represent the classification variable that represents the outcome of each launch. If the value is zero, the first stage did not land successfully; one means the first stage landed Successfully

```
landing_class = df['Class']
df[['Class']].head(8)
```

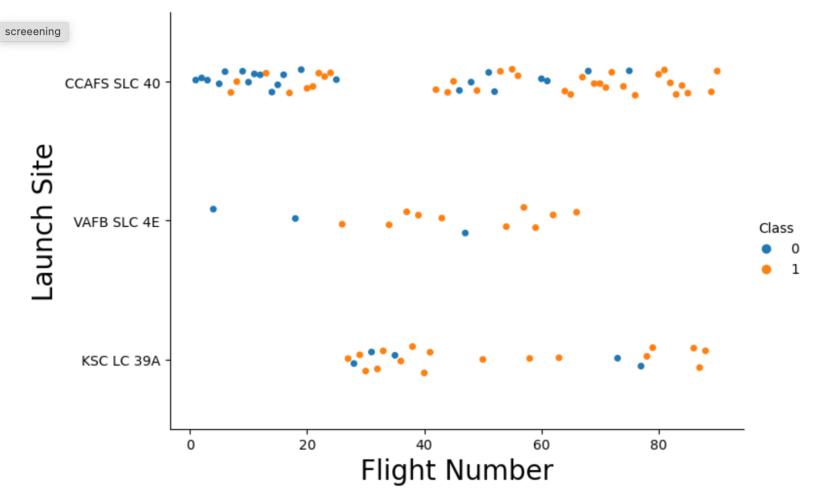
	Class
0	0
1	0
2	0
3	0
4	0
5	0
6	1
7	1

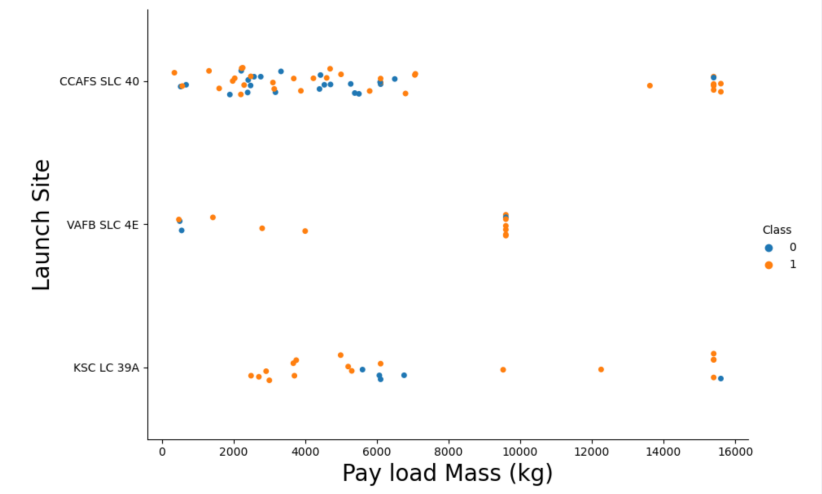
```
df.head(5)
```

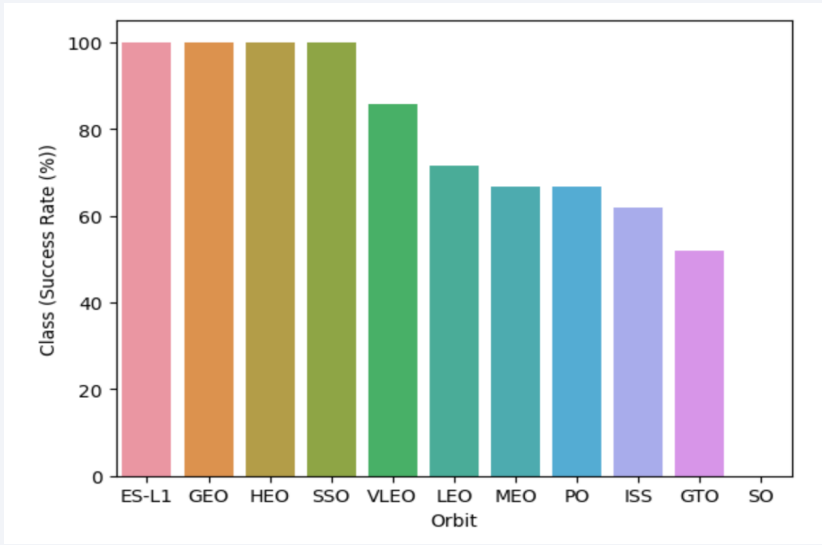
EDA with Data Visualization

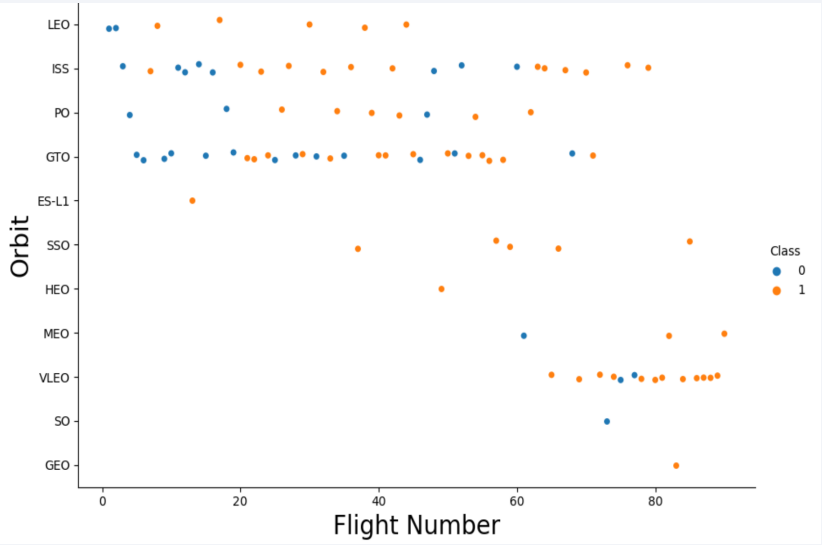
- Conducted data analysis and feature engineering using Pandas and Matplotlib
  - Exploratory Data Analysis
  - Preparing Data Feature Engineering
- Employed scatter plots to illustrate relationships between Flight Number and Launch Site, Payload and Launch Site, Flight Number and Orbit Type, as well as Payload and Orbit Type
- Utilizing bar charts to visualize the success rates for each orbit type
- Implementing line plots to depict the annual trend in launch success
- Here is the GitHub URL of the completed EDA with data visualization notebook

EDA with Data Visualization (Plots)









EDA with SQL

- The following SQL queries were performed for EDA:
  - **Display the names of the unique launch sites in the space mission**

```
%sql SELECT DISTINCT LAUNCH_SITE AS "Launch_Sites" FROM SPACEXTBL;
```

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

```
%sql select * from SPACEXTBL WHERE Launch_Site LIKE 'CCA%' LIMIT 5;
```

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

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

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

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) as "Payload Mass Kgs", Customer, Booster_Version FROM 'SPACEXTBL' WHERE Booster_Version LIKE 'F9 v1.1%';
```

EDA with SQL

- The following SQL queries were performed for EDA (cont'd):
  - **List the date when the first successful landing outcome in ground pad was achieved.**

```
%sql select min(DATE) from 'SPACEXTBL' where Landing_Outcome = "Success (ground pad)";
```

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

%sql SELECT DISTINCT Booster_Version, Payload FROM SPACEXTBL WHERE Landing_Outcome = "Success (drone ship)" AND PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 6000;

  - **List the total number of successful and failure mission outcomes**

```
%sql SELECT "Mission_Outcome", COUNT("Mission_Outcome") as Total FROM SPACEXTBL GROUP BY "Mission_Outcome";
```

  - **List the names of the booster versions which have carried the maximum payload mass. Use a subquery**

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

- Here is the GitHub URL of the completed EDA with SQL notebook

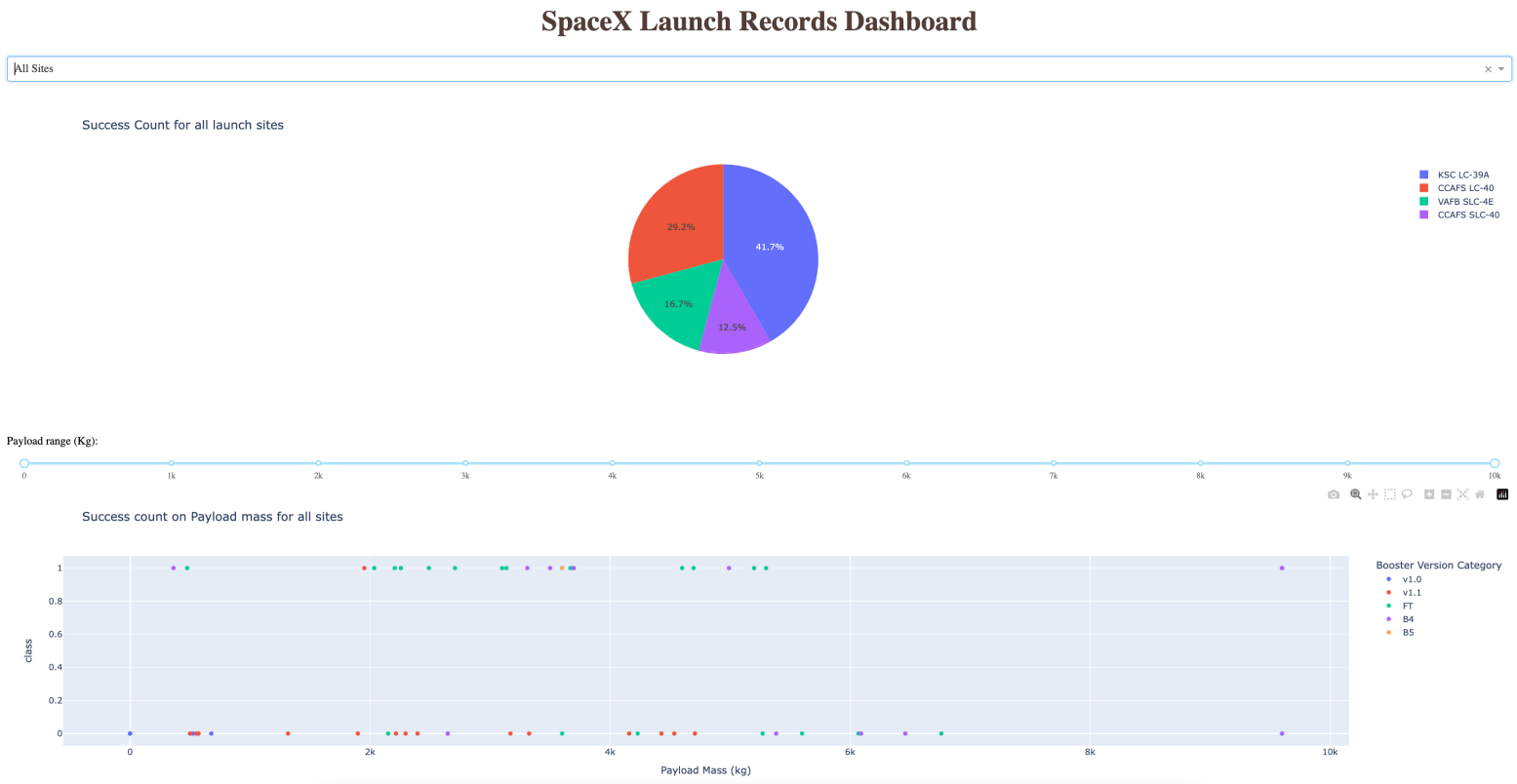
Build an Interactive Map with Folium

- Created folium map to mark all the launch sites, and created map objects such as markers, circles, lines to mark the success or failure of launches for each launch site
- Created a launch set outcomes (failure=0 or success=1)
- Here is the GitHub URL of the completed interactive map with Folium map

Build a Dashboard with Plotly Dash

- Developed an interactive dashboard application using Plotly Dash, incorporating:
  - Integration of a Launch Site Drop-down input component
  - Implementation of a callback function to dynamically render the success-pie-chart based on the selected launch site from the dropdown
  - Inclusion of a range slider for payload selection
  - Implementation of another callback function to dynamically render the success-payload-scatter-chart based on the selected payload range
- Here is the GitHub URL of the completed Plotly Dash project.

SpaceX Dashboard Application



Predictive Analysis (Classification)

- To determine the optimal ML model among SVM, Classification Trees, k-nearest neighbors, and Logistic Regression for the test data:
    - Firstly, individual objects were created for each algorithm, followed by the instantiation of GridSearchCV objects. Each model was assigned specific parameters for evaluation.
    - For each model, GridSearchCV objects were created with a cross-validation value of 10. The training data was fitted into these GridSearch objects to identify the best hyperparameters.
    - Post-training, the GridSearchCV objects were outputted for each model, displaying the best parameters via the 'best_params_' attribute and the validation accuracy using the 'best_score_' attribute.
    - Finally, the 'score' method was employed to calculate accuracy on the test data for each model. Confusion matrices were plotted for each, comparing the test and predicted outcomes.

Predictive Analysis (Classification)

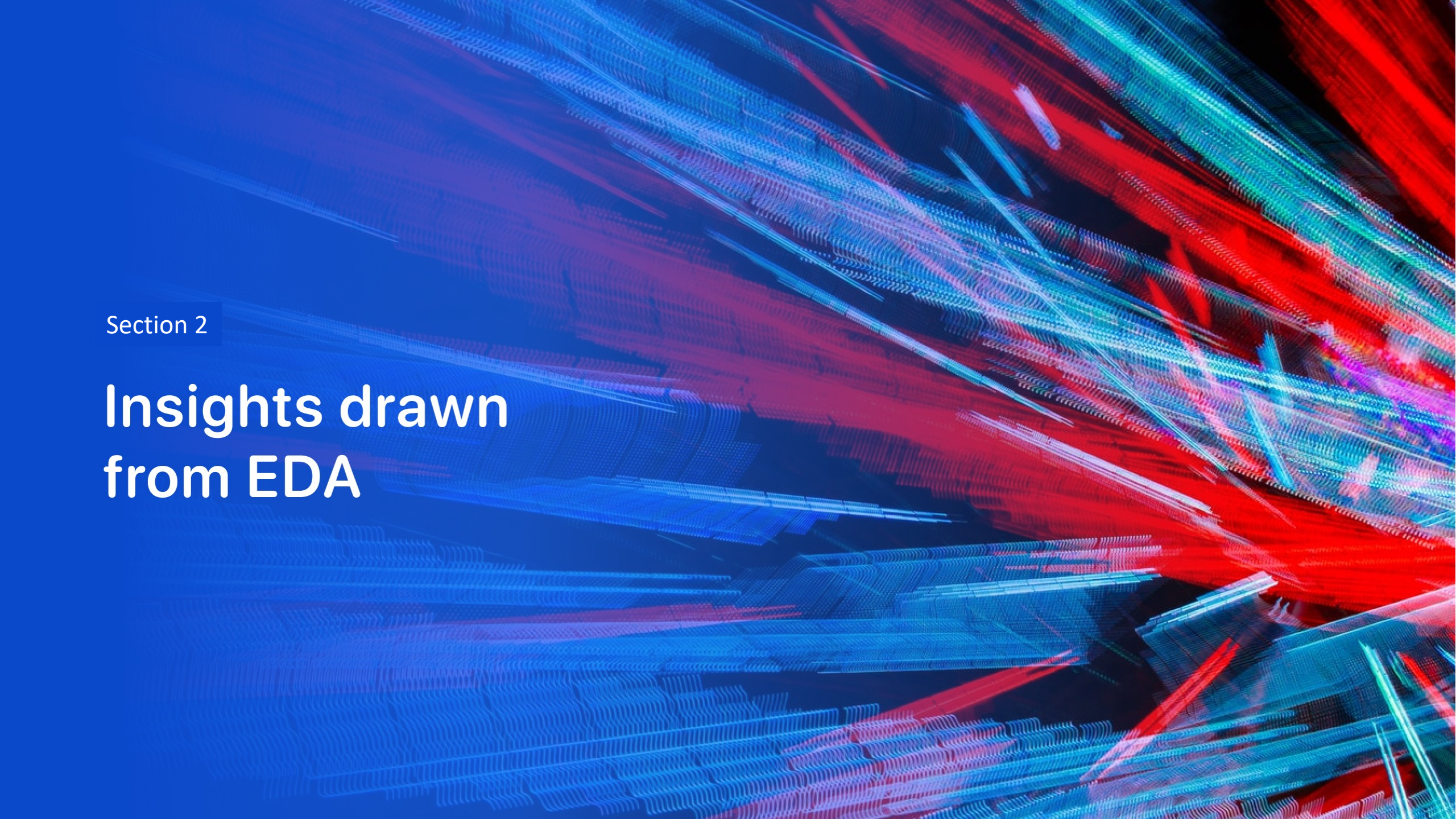
- The table below shows the test data accuracy score for each of the methods comparing them to show which performed best using the test data between SVM, Classification Trees, k nearest neighbors and Logistic Regression

	0
Method	Test Data Accuracy
Logistic_Reg	0.833333
SVM	0.833333
Decision Tree	0.777778
KNN	0.833333

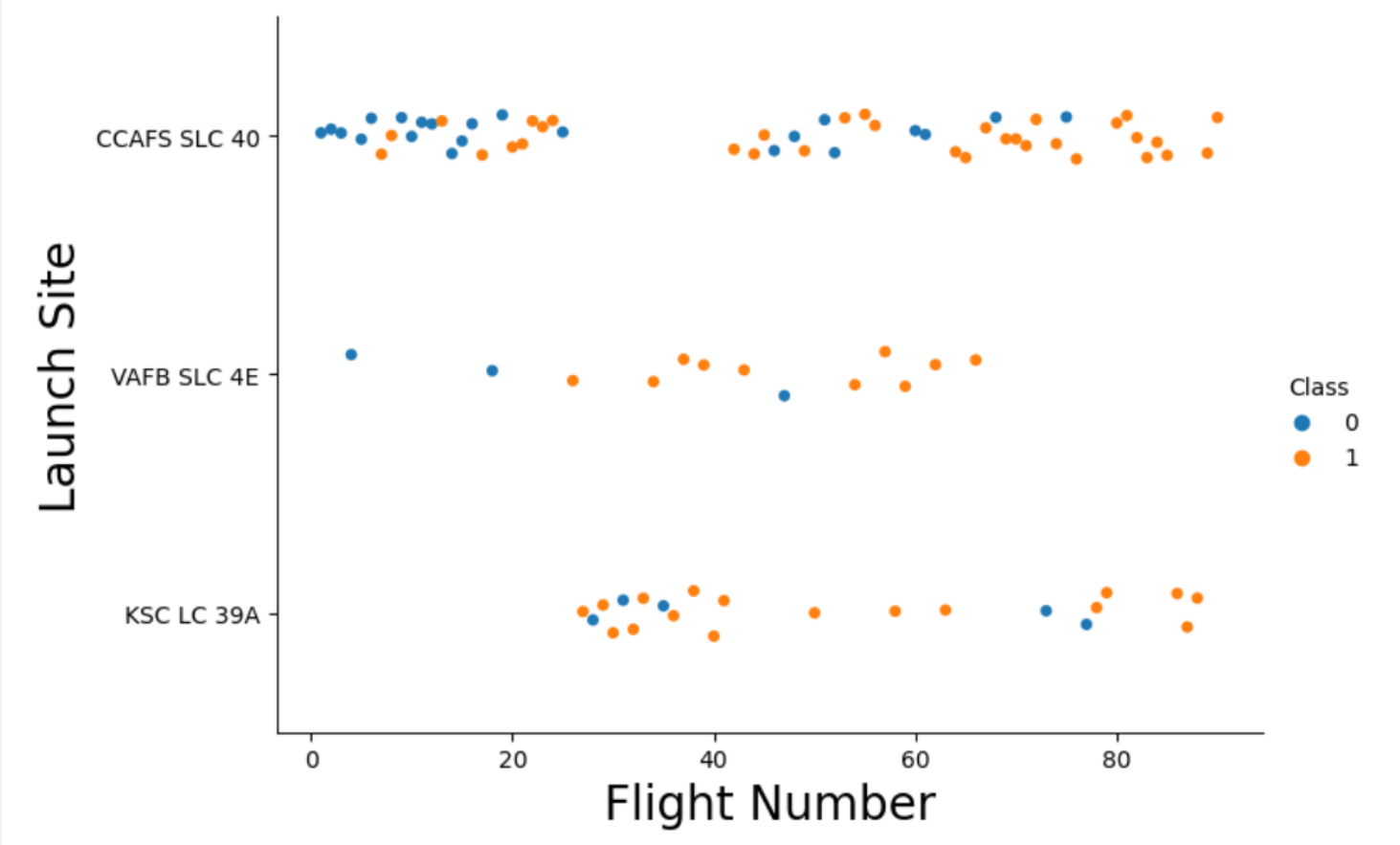
- Here is the GitHub URL of the completed predictive analysis

Results

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


Section 2
Insights drawn from EDA

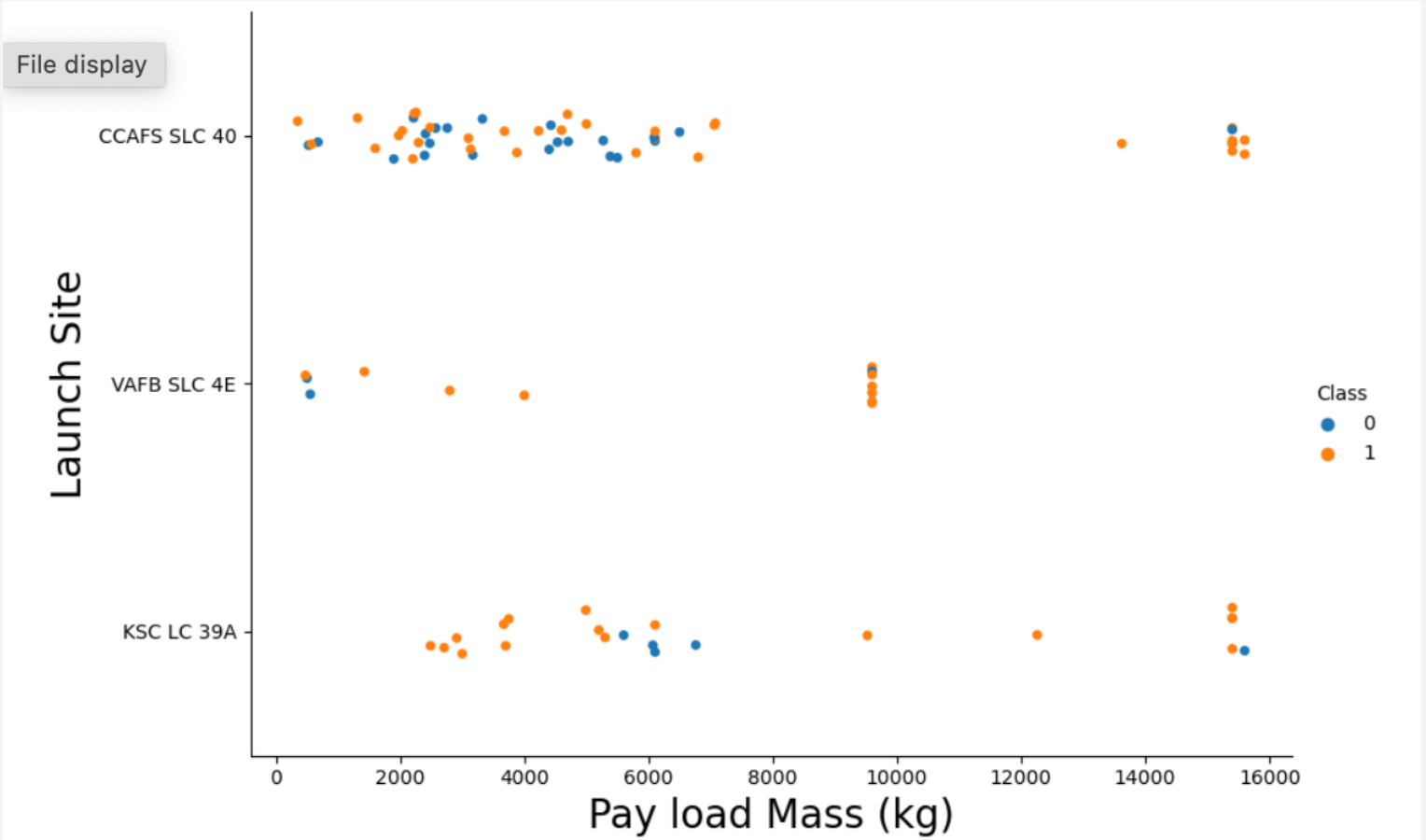
Flight Number vs. Launch Site



Now try to explain the patterns you found in the Flight Number vs. Launch Site scatter point plots.

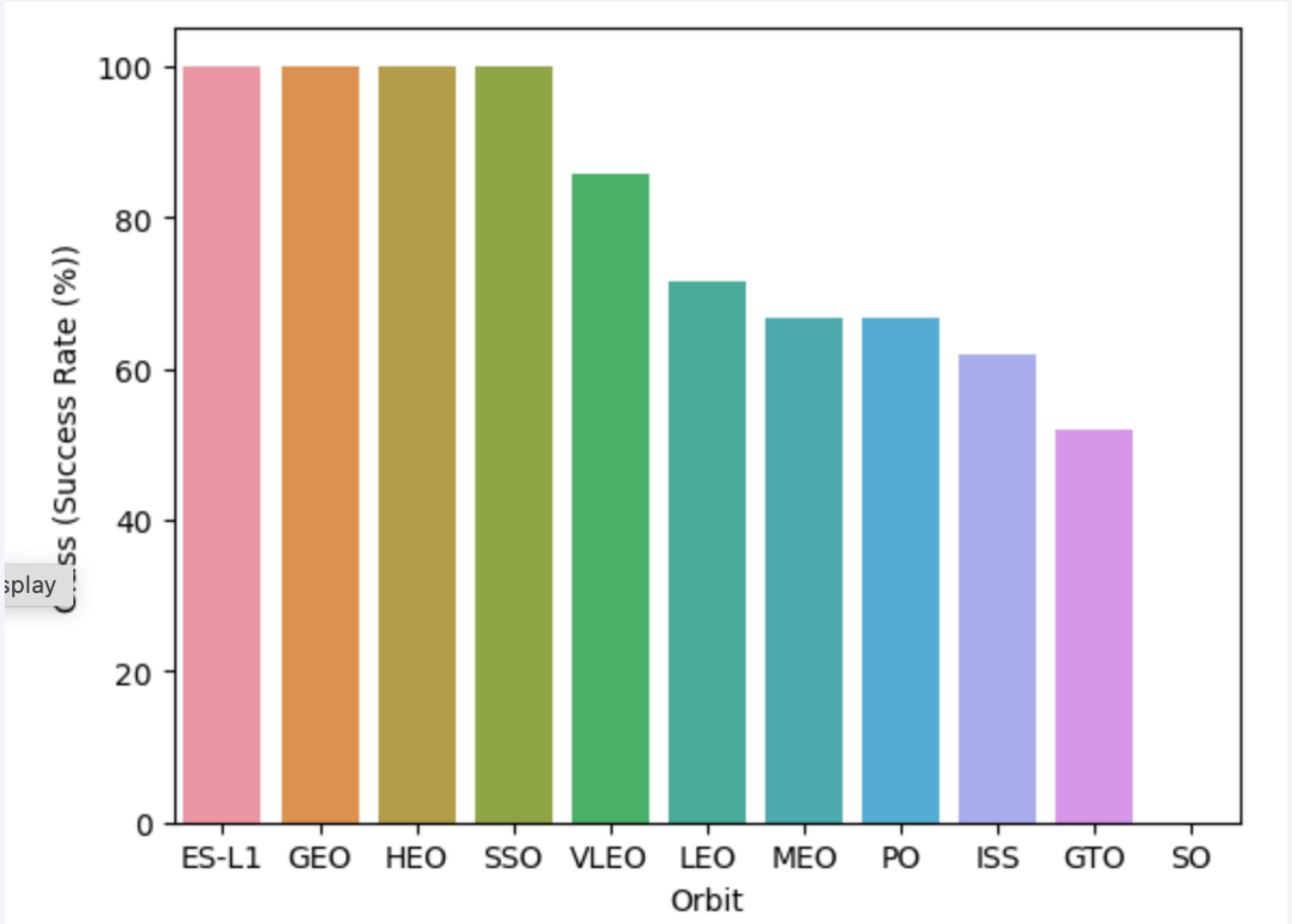
We can deduce that, as the flight number increases in each of the 3 launcg sites, so does the success rate. For instance, the success rate for the VAFB SLC 4E launch site is 100% after the Flight number 50. Both KSC LC 39A and CCAFS SLC 40 have a 100% success rates after 80th flight.

Payload vs. Launch Site



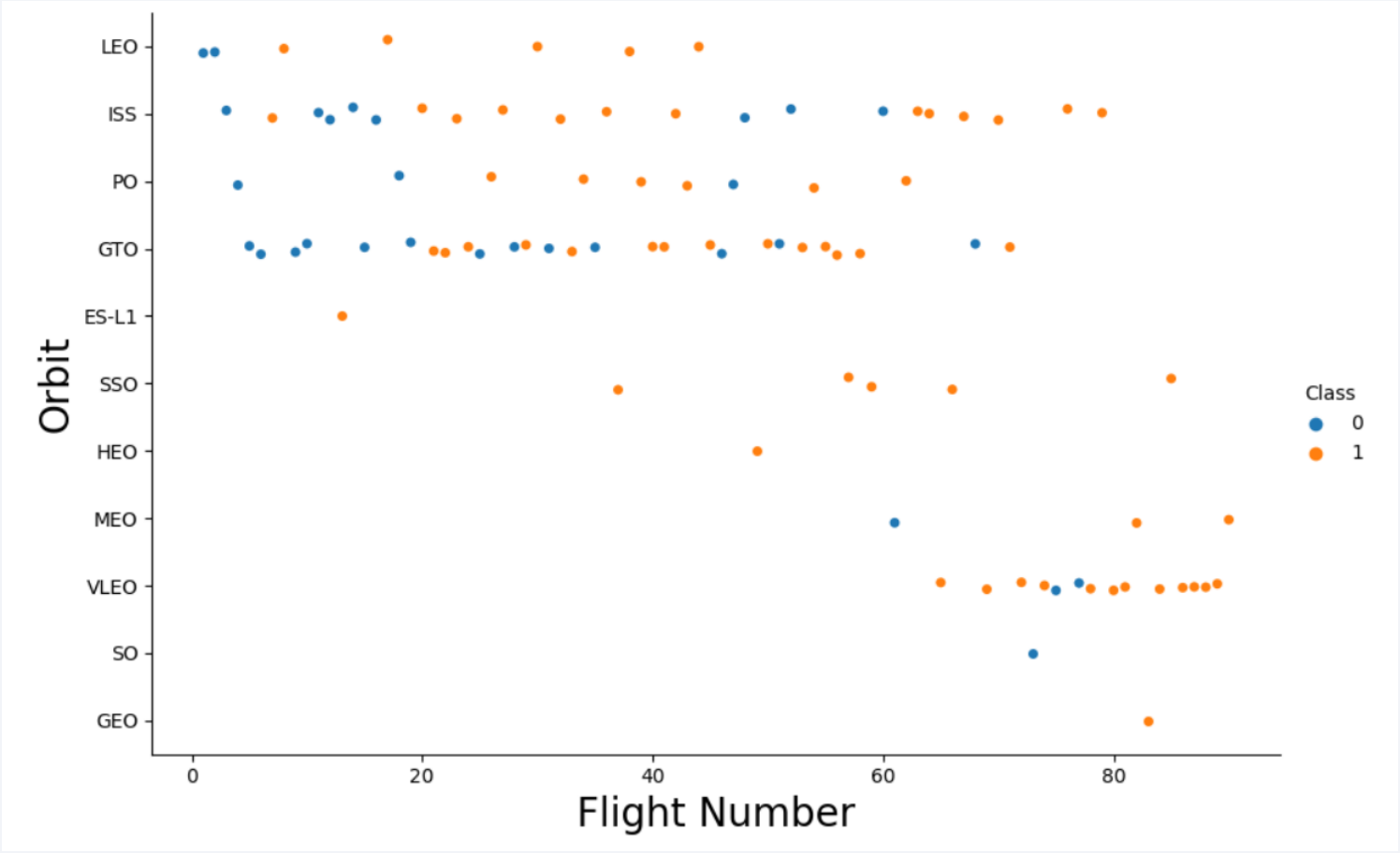
Now if you observe Payload Vs. Launch Site scatter point chart you will find for the VAFB-SLC launchsite there are no rockets launched for heavypayload mass(greater than 10000).

Success Rate vs. Orbit Type



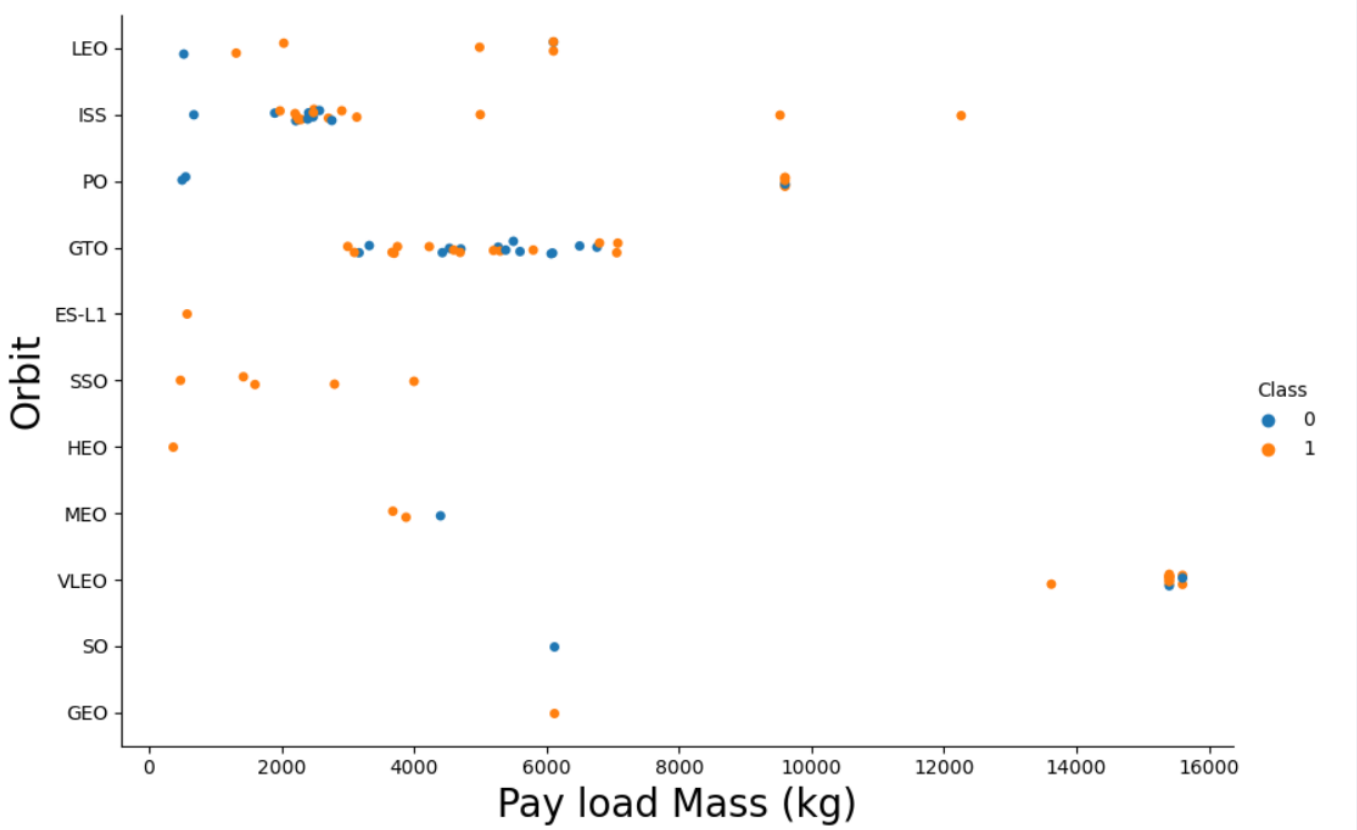
Analyze the ploted bar chart try to find which orbits have high sucess rate.

Flight Number vs. Orbit Type



You should see that in the 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.

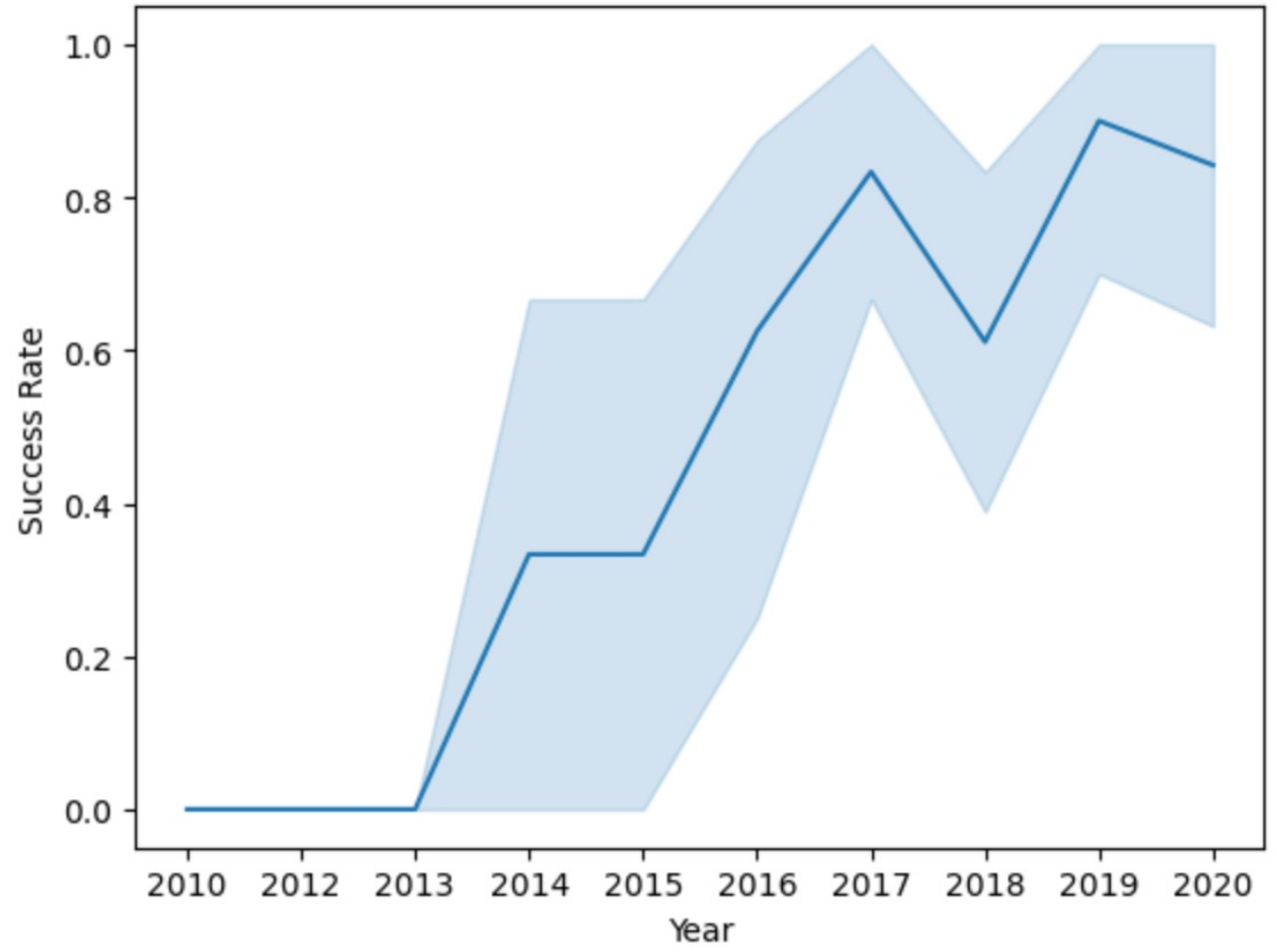
Payload vs. Orbit Type



With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS.

However 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



you can observe that the sucess rate since 2013 kept increasing till 2020

All Launch Site Names

- A SQL query was used to retrieve launch site names. The 'SELECT DISTINCT' statement was used to return only the unique launch sites from the 'LAUNCH_SITE' column of the SPACEXTBL table

```
%sql SELECT DISTINCT LAUNCH_SITE AS "Launch_Sites" FROM SPACEXTBL;
```

```
 * sqlite:///my_data1.db
Done.
```

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

Launch Site Names Begin with 'CCA'

- The command 'LIKE' was used with '%' wildcard in 'WHERE' clause to select and display a table of all records with launch site names that begin with the string 'CCA'

```
%sql select * from SPACEXTBL WHERE Launch_Site LIKE 'CCA%' LIMIT 5;
```

```
 * sqlite:///my_data1.db
Done.
```

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_
2010-04-06	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (I
2010-08-12	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 (I
2012-05-22	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	N
2012-08-10	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	N
2013-01-03	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	N

Total Payload Mass

- To calculate the total payload carried by boosters from NASA, the ‘SUM()’ function was used to return and display the total sum of ‘PAYLOAD_MASS_KG’ column for customer ‘NASA (CRS)’

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

```
 * sqlite:///my_data1.db
Done.
```

Total Payload Mass(Kgs)	Customer
45596	NASA (CRS)

Average Payload Mass by F9 v1.1

- To calculate the average payload mass carried by booster version F9 v1.1, the 'AVG()' function was used to return and display the average payload mass carried by booster version F9 v1.1

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

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) as "Payload Mass Kgs", Customer, Booster_Version FROM 'SPACEXTBL' WHERE Booster_Version LIKE 'F9 v1.1%';
```

```
 * sqlite:///my_data1.db
Done.
```

Payload Mass Kgs	Customer	Booster_Version
2534.6666666666665	MDA	F9 v1.1 B1003

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad
- The 'MIN()' function was used to return the first (oldest) date of successful landing outcome on ground pad.

```
%sql select min(DATE) from 'SPACEXTBL' where Landing_Outcome = "Success (ground pad)";
```

```
 * sqlite:///my_data1.db
Done.
```

min(DATE)
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- List the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000
- The 'SELECT DISTINCT' function as used to return unique names of booster with operators >4000 and <6000 to only list booster with payloads between 4000-6000 with landing outcome of 'Success (drone ship)'

```
%sql SELECT DISTINCT Booster_Version, Payload FROM SPACEXTBL WHERE Landing_Outcome = "Success (drone ship)" AND P
```

```
 sqlite:///my_data1.db
one.
```

Booster_Version	Payload
F9 FT B1022	JCSAT-14
F9 FT B1026	JCSAT-16
F9 FT B1021.2	SES-10
F9 FT B1031.2	SES-11 / EchoStar 105

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes
- Used the ‘COUNT()’ together with the ‘GROUP BY’ statement to return total number of mission outcomes

```
%sql SELECT "Mission_Outcome", COUNT("Mission_Outcome") as Total FROM SPACEXTBL GROUP BY "Mission_Outcome";
```

```
 * sqlite:///my_data1.db
Done.
```

Mission_Outcome	Total
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass
- Using a subquery to return and pass the max payload and used it to list the boosters that have carried the Max payload of 15600 kgs

```
%sql SELECT "Booster_Version",Payload, "PAYLOAD_MASS__KG_" FROM SPACEXTBL WHERE "PAYLOAD_MASS__KG_" = (SELECT MAX
```

```
* sqlite:///my_data1.db
one.
```

Booster_Version	Payload	PAYLOAD_MASS__KG_
F9 B5 B1048.4	Starlink 1 v1.0, SpaceX CRS-19	15600
F9 B5 B1049.4	Starlink 2 v1.0, Crew Dragon in-flight abort test	15600
F9 B5 B1051.3	Starlink 3 v1.0, Starlink 4 v1.0	15600
F9 B5 B1056.4	Starlink 4 v1.0, SpaceX CRS-20	15600
F9 B5 B1048.5	Starlink 5 v1.0, Starlink 6 v1.0	15600
F9 B5 B1051.4	Starlink 6 v1.0, Crew Dragon Demo-2	15600
F9 B5 B1049.5	Starlink 7 v1.0, Starlink 8 v1.0	15600
F9 B5 B1060.2	Starlink 11 v1.0, Starlink 12 v1.0	15600
F9 B5 B1058.3	Starlink 12 v1.0, Starlink 13 v1.0	15600
F9 B5 B1051.6	Starlink 13 v1.0, Starlink 14 v1.0	15600
F9 B5 B1060.3	Starlink 14 v1.0, GPS III-04	15600
F9 B5 B1049.7	Starlink 15 v1.0, SpaceX CRS-21	15600

2015 Launch Records

- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
- Used the 'substr()' in the select statement to get the month and year from the date column where substr(Date,7,4)='2015' for year and Landing_outcome was 'Failure (drone ship)' and return the records matching the filter

File display

```
CT DATE, strftime('%Y', DATE) AS Year, strftime('%m', DATE) AS Month,"Booster_Version", "Launch_Site", "
```

```
 * sqlite:///my_data1.db
Done.
```

Date	Year	Month	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Mission_Outcome	Landing_Outcome
2015-10-01	2015	10	F9 v1.1 B1012	CCAFS LC-40	SpaceX CRS-5	2395	Success	Failure (drone ship)
2015-04-14	2015	04	F9 v1.1 B1015	CCAFS LC-40	SpaceX CRS-6	1898	Success	Failure (drone ship)

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

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

```
%sql SELECT * FROM SPACEXTBL WHERE "Landing_Outcome" LIKE 'Success%' AND (DATE BETWEEN '2010-06-04' AND '2017-03-
 sqlite:///my_data1.db
one.
```

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Lan
2017-03-06	21:07:00	F9 FT B1035.1	KSC LC-39A	SpaceX CRS-11	2708	LEO (ISS)	NASA (CRS)	Success	S
2017-02-19	14:39:00	F9 FT B1031.1	KSC LC-39A	SpaceX CRS-10	2490	LEO (ISS)	NASA (CRS)	Success	S
2017-01-14	17:54:00	F9 FT B1029.1	VAFB SLC-4E	Iridium NEXT 1	9600	Polar LEO	Iridium Communications	Success	
2017-01-05	11:15:00	F9 FT B1032.1	KSC LC-39A	NROL-76	5300	LEO	NRO	Success	S
2016-08-14	05:26:00	F9 FT B1026	CCAFS LC-40	JCSAT-16	4600	GTO	SKY Perfect JSAT Group	Success	
2016-08-04	20:43:00	F9 FT B1021.1	CCAFS LC-40	SpaceX CRS-8	3136	LEO (ISS)	NASA (CRS)	Success	
2016-07-18	04:45:00	F9 FT B1025.1	CCAFS LC-40	SpaceX CRS-9	2257	LEO (ISS)	NASA (CRS)	Success	S
2016-06-05	05:21:00	F9 FT B1022	CCAFS LC-40	JCSAT-14	4696	GTO	SKY Perfect JSAT Group	Success	
2016-05-27	21:39:00	F9 FT B1023.1	CCAFS LC-40	Thaicom 8	3100	GTO	Thaicom	Success	
2015-12-22	01:29:00	F9 FT B1019	CCAFS LC-40	OG2 Mission 2 11 Orbcomm-OG2 satellites	2034	LEO	Orbcomm	Success	S

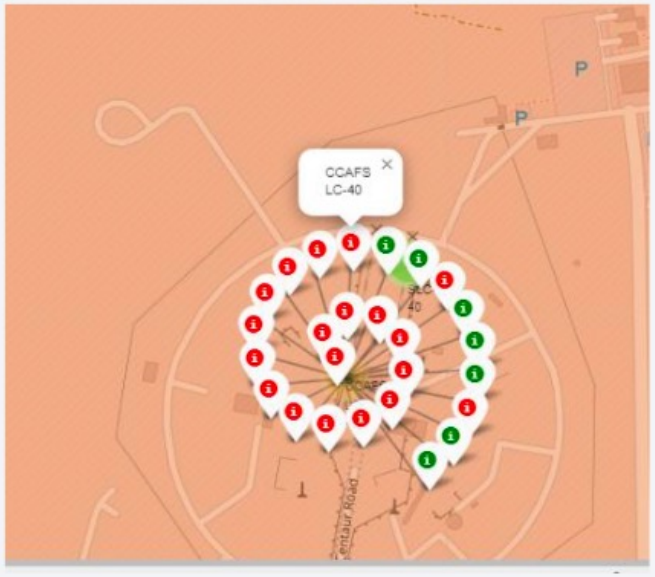

Section 3
Launch Sites
Proximities Analysis

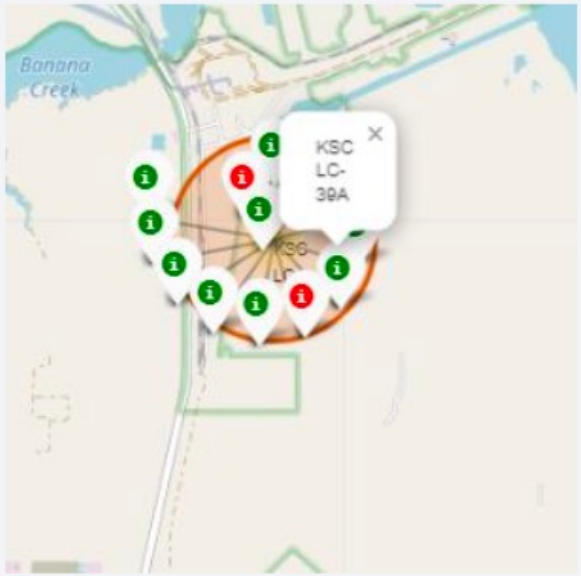
Markers of all launch sites on global Map

- All launch sites are in proximity to the Equator (located southwards of the US map). Also all the launch sites are in very close proximity to the coast

Launch outcomes for each site on map with color markers

- In the Eastern coast (Florida) launch site KSC LC-39A has high success rates compared to CCAFS SCL-40 and CCAFS LC-40





Launch outcomes for each site on map with color markers

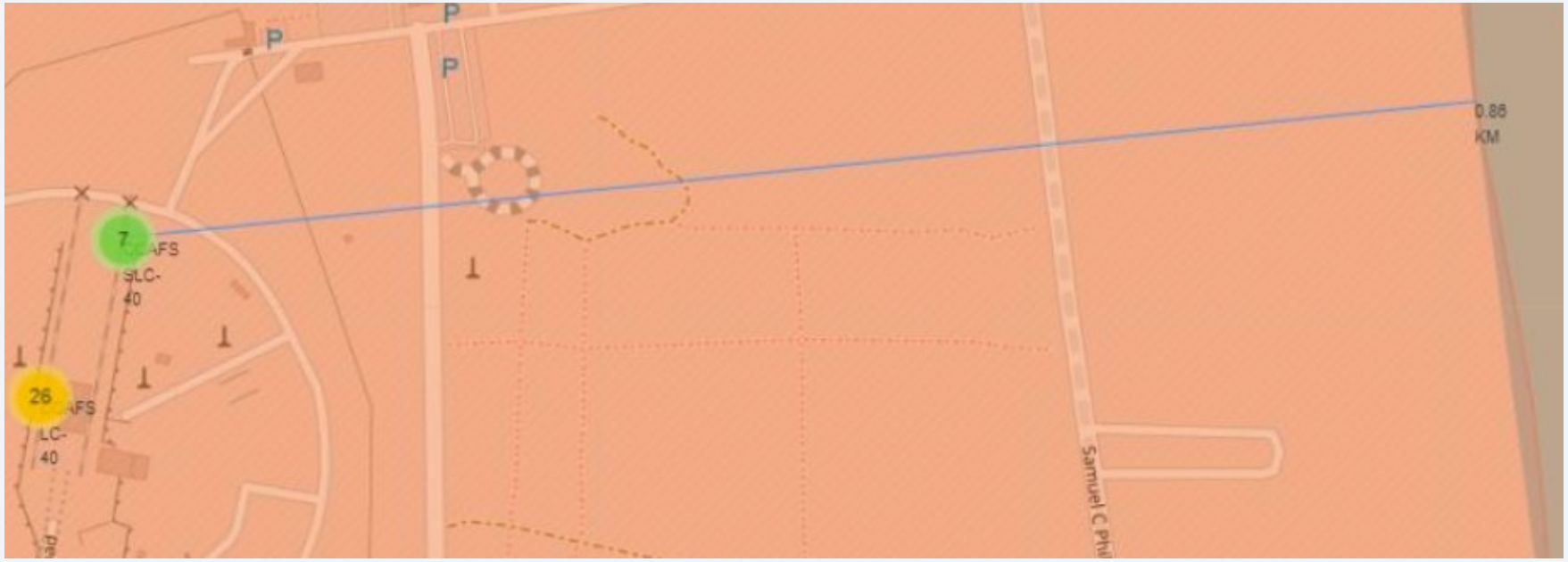
- In the West coast (California) launch site VAFB SLC-4E has low success rates compared to KSC LC-39A in the eastern coast of Florida

Launch outcomes for each site on map with color markers

- In the West coast (California) launch site VAFB SLC-4E has low success rates compared to KSC LC-39A in the eastern coast of Florida

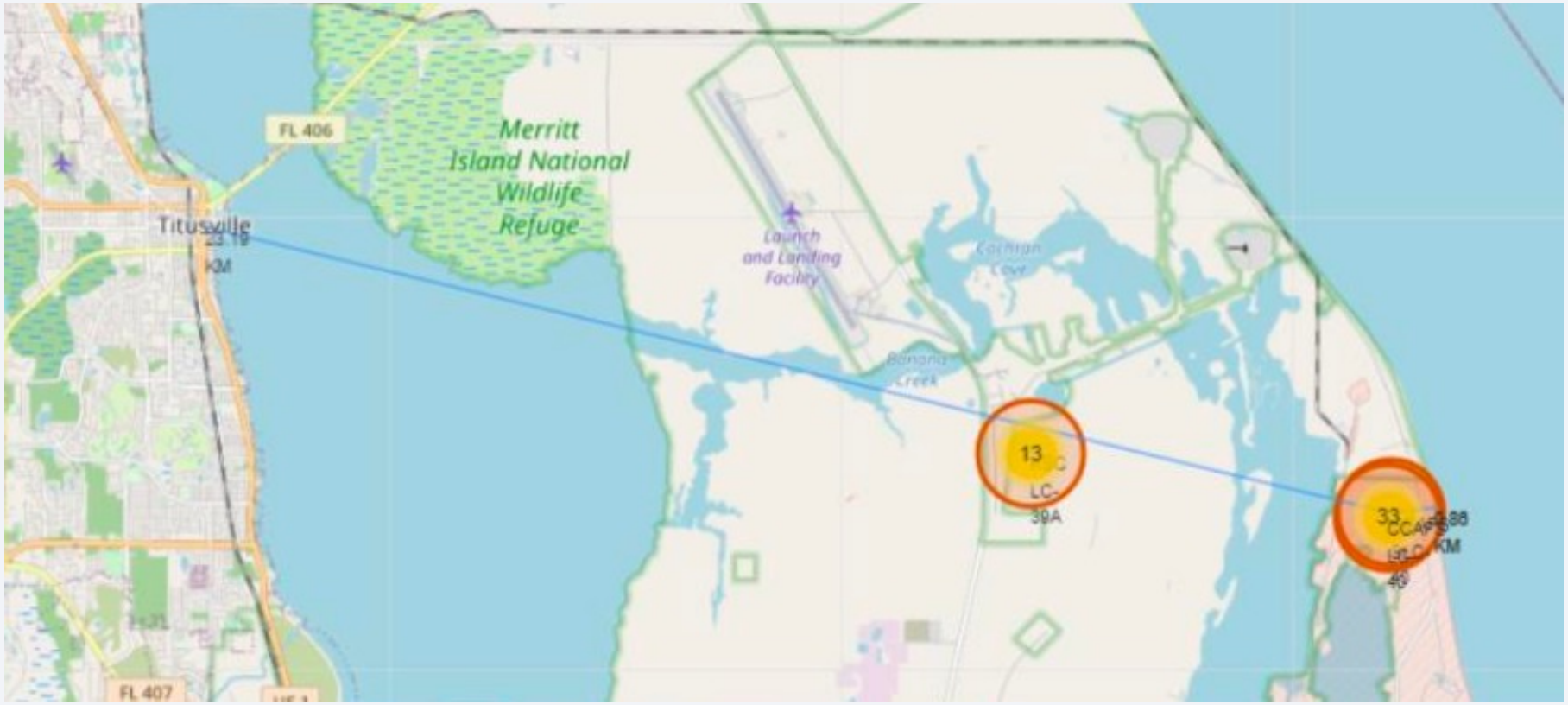
Distances between a launch site to its proximity

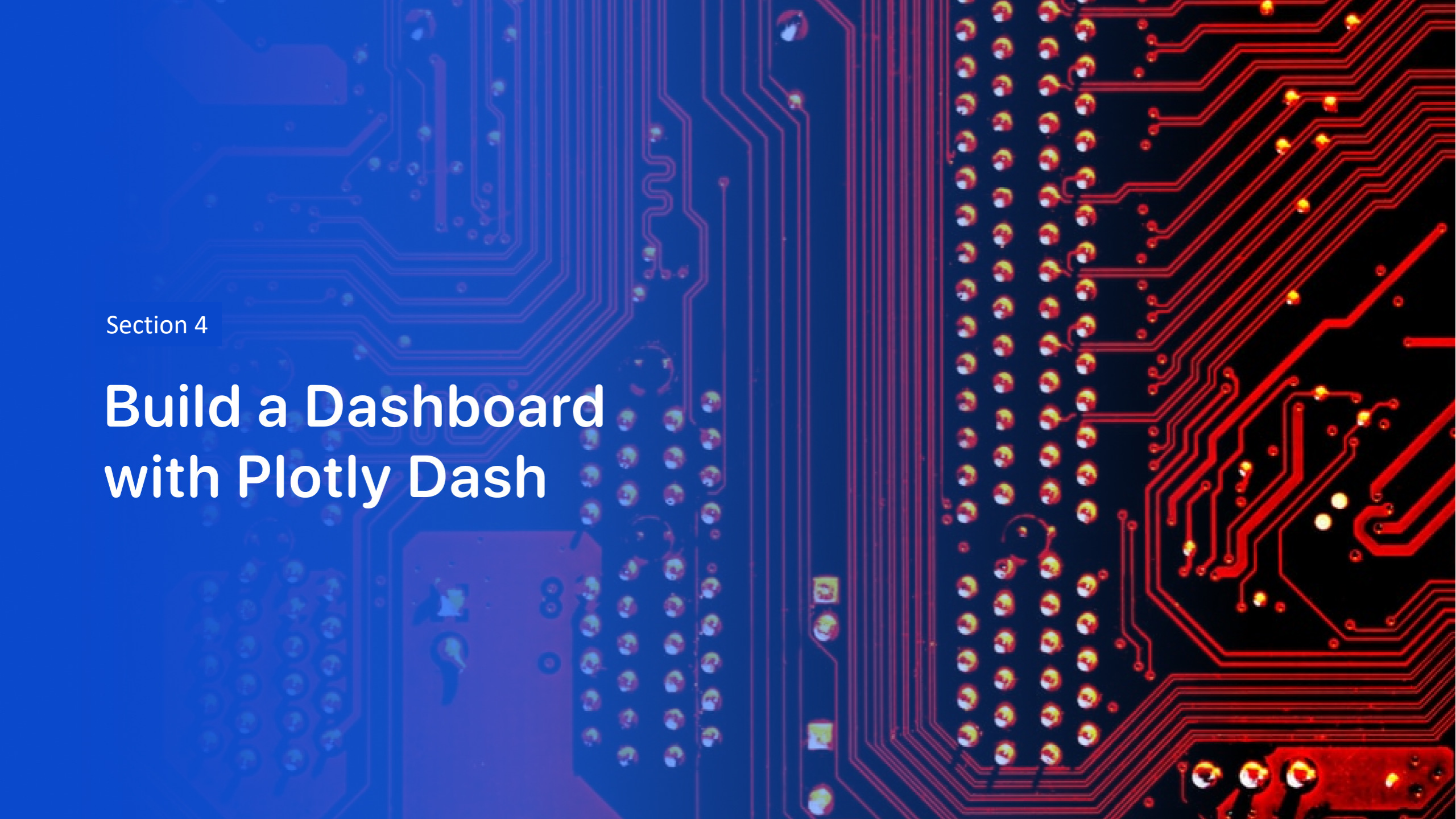
- Launch site CCAFS SLC-40 proximity to coastline is 0.86km



Distances between a launch site to its proximity

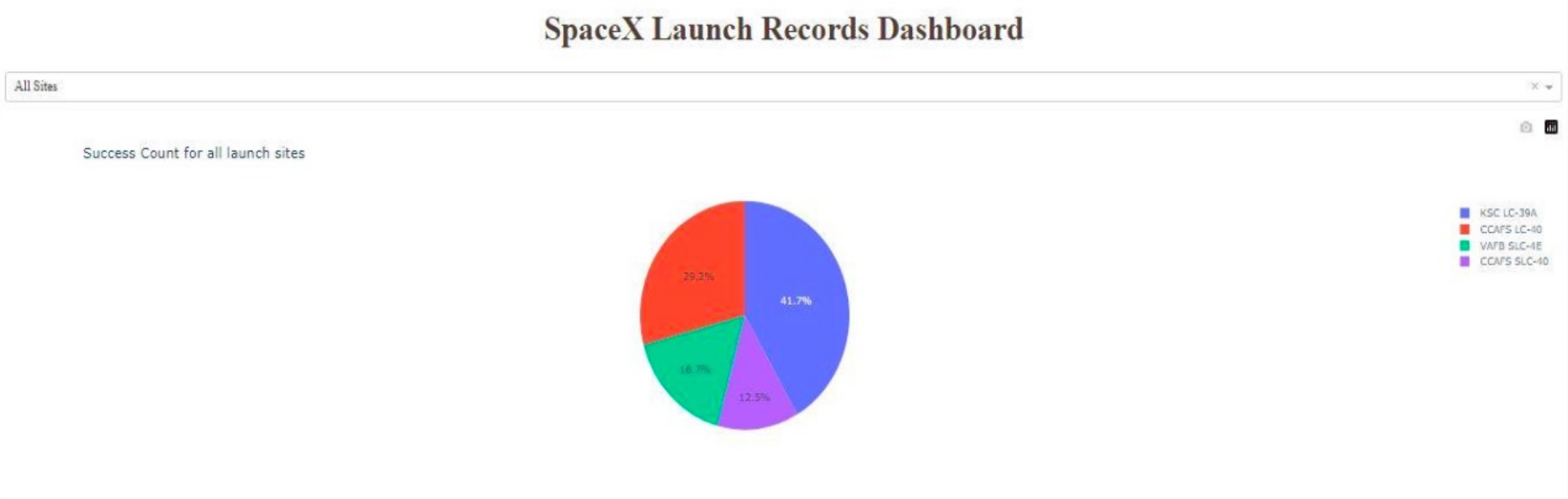
- Launch site CCAFS SLC-40 is 23.19 km away from closest highway (Washington Avenue)




Section 4
Build a Dashboard with Plotly Dash

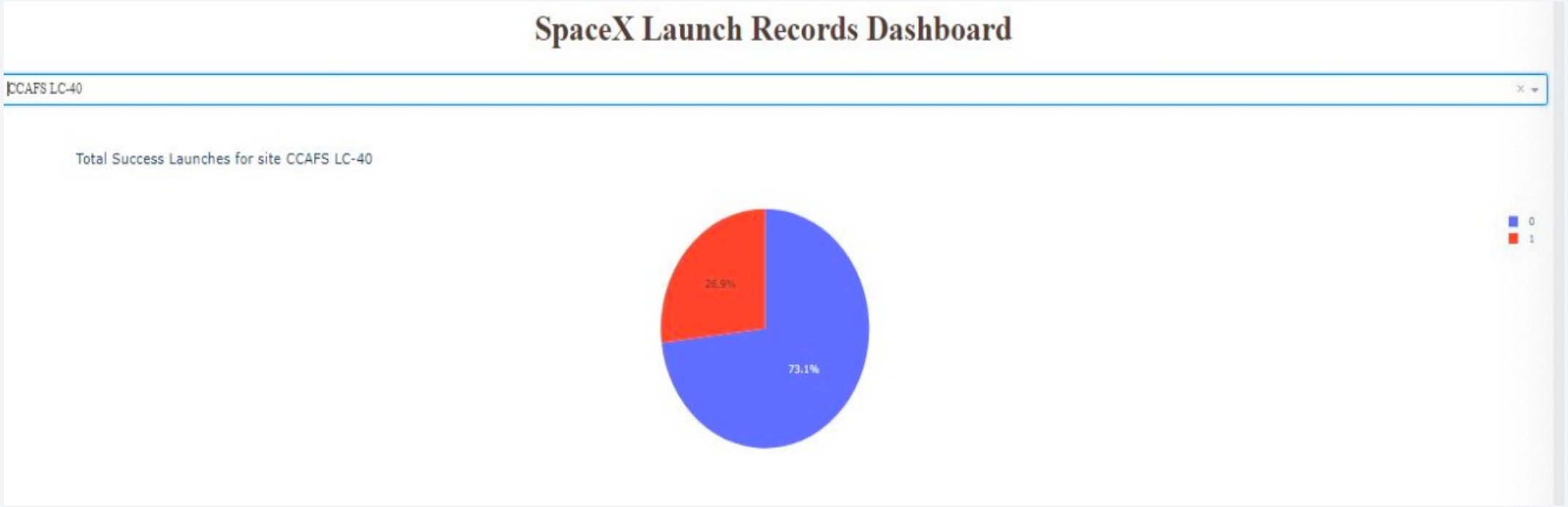
Pie-chart for launch success count for all sites

- KSC LC-39A boasts the highest launch success rate at 42%, followed by CCAFS LC-40 at 29%. VAFB SLC-4E trails at 17%, while CCAFS SLC-40 lags behind with a success rate of 13%.



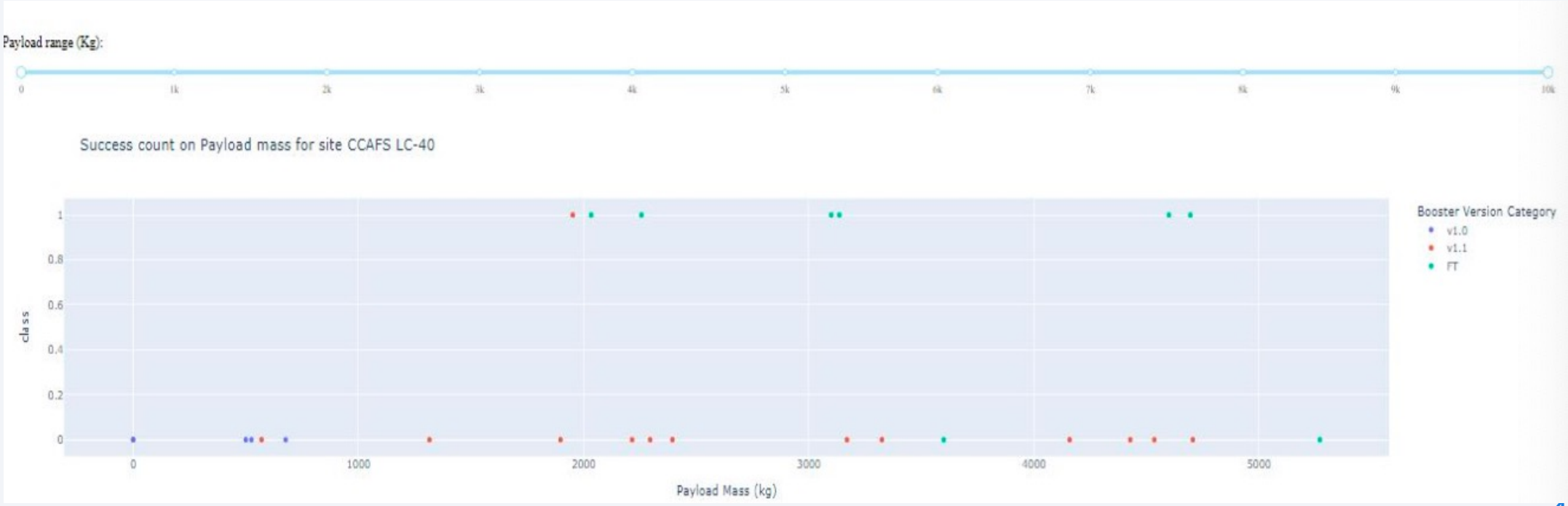
Pie chart for launch site with highest launch success ratio

- Launch site CCAFS LC-40 had the highest success ratio of 73% success against 27% failed launches



Payload vs. Launch Outcome scatter plot for all sites

- For launch site CCAFS LC-40, the booster version FT has the largest success rate from a payload mass of >2000 kg



Section 5

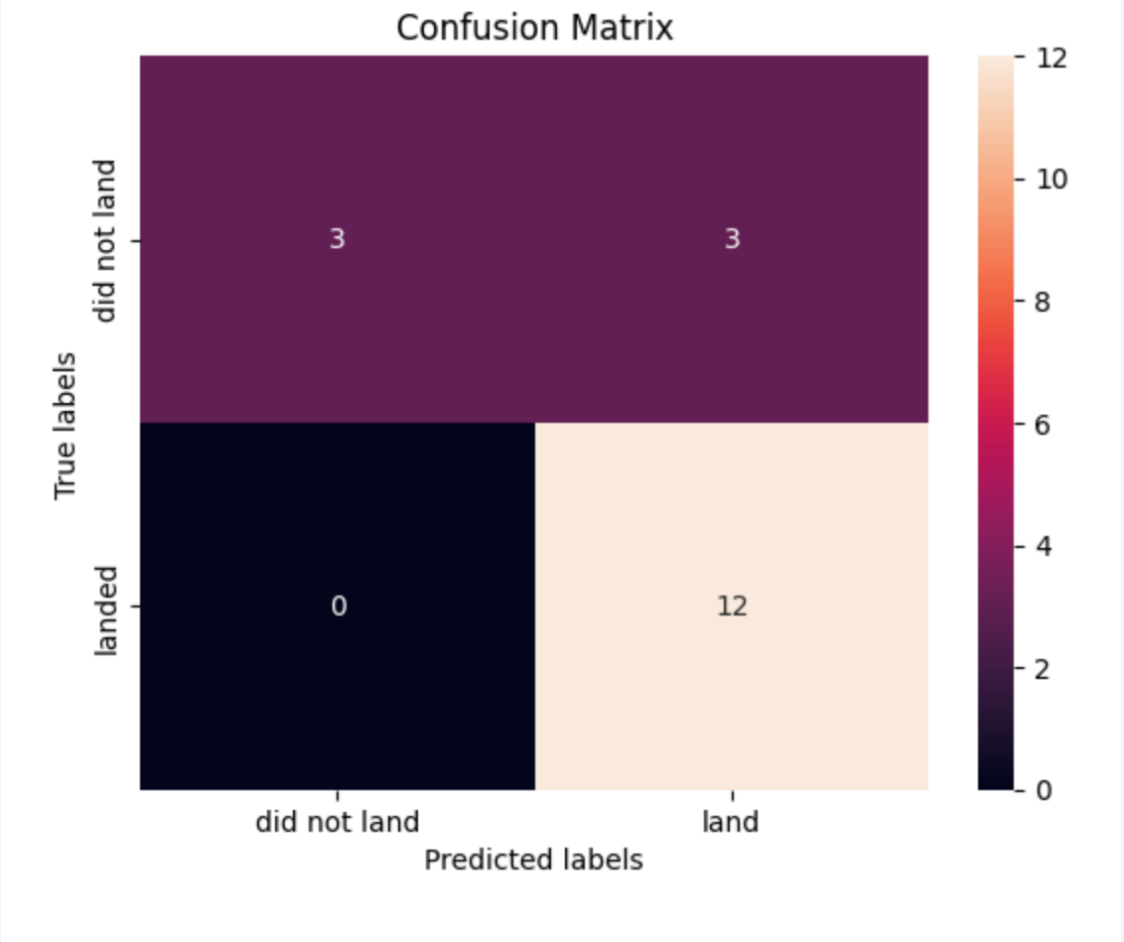
Predictive Analysis (Classification)

Classification Accuracy

	0
Method	Test Data Accuracy
Logistic_Reg	0.833333
SVM	0.833333
Decision Tree	0.777778
KNN	0.833333

Confusion Matrix

- The four classification models exhibited identical confusion matrices and demonstrated equal proficiency in distinguishing between various classes.
- The predominant challenge encountered across all models was the occurrence of false positives.



Conclusions

- Different launch sites exhibit varying success rates. CCAFS LC-40 achieves a 60% success rate, while KSC LC-39A and VAFB SLC 4E boast success rates of 77%.
- Additionally, the success rate tends to increase with the flight number at these launch sites. For instance, after the 50th flight, VAFB SLC 4E achieves a 100% success rate, and both KSC LC 39A and CCAFS SLC 40 maintain a 100% success rate after the 80th flight.
- In the Payload Vs. Launch Site scatter plot, it's noticeable that VAFB-SLC has no rockets launched with heavy payload mass (greater than 10000).
- Concerning orbits, ES-L1, GEO, HEO, and SSO demonstrate perfect success rates at 100%, while the SO orbit exhibits the lowest success rate at approximately 50%. Notably, Orbit SO shows a 0% success rate.
- In the LEO orbit, success seems correlated with the number of flights. However, there appears to be no discernible relationship between flight number and success in the GTO orbit.

Conclusions (cont'd)

- When dealing with heavy payloads, the positive landing rate is higher for Polar, LEO, and ISS orbits. However, distinguishing this trend for GTO is challenging because both positive and negative landing outcomes (unsuccessful missions) coexist in this category.
- Lastly, the success rate has steadily risen since 2013, reaching its peak in 2020.

Thank you!