IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Stuart McIntosh
21-Feb-2023

Outline

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

Executive Summary

The question is “Will the Falcon 9 first stage land successfully”.

- Summary of methodologies

Data was obtained directly from SpaceX using a REST API, then the data was cleaned. Next we performed exploratory data analysis (EDA) using visualisation and SQL, followed by interactive visual analytics using Folium and Plotly Dash. Finally we did predictive analysis testing various classification models.

- Summary of all results

We created a model that can predict if a falcon 9 first stage will land successfully with 83% accuracy. We can use that model to estimate the cost of a launch.

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.

- Problem to solve

If we can determine if the first stage will land, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch. We will predict if the Falcon 9 first stage will land successfully.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was collected using the SpaceX Rest API.
- Perform data wrangling
  - The data was filtered to only include Falcon 9 launches.
  - Missing values in ‘Payload Mass’ were replaced by the mean value.
  - The outcome for each flight was labelled either as ‘1’ (successful) or ‘0’ (unsuccessful).
- Perform exploratory data analysis (EDA) using visualization and SQL
  - The data was loaded into a SQL database and queries were run to investigate various scenarios. The data was also plotted to highlight any patterns in the data.
- Perform interactive visual analytics using Folium and Plotly Dash
  - Folium was used to show the location of the launch-sites, and the distance from each launch-site to other features such as roads and cities.
  - Plotly was used to develop interactive graphs to explore the relationships between different launch-sites, payload sizes and launch success.
- Perform predictive analysis using classification models
  - Various classification models were tested to find the model that best predicted launch success, trained on the SpaceX data.

Data Collection

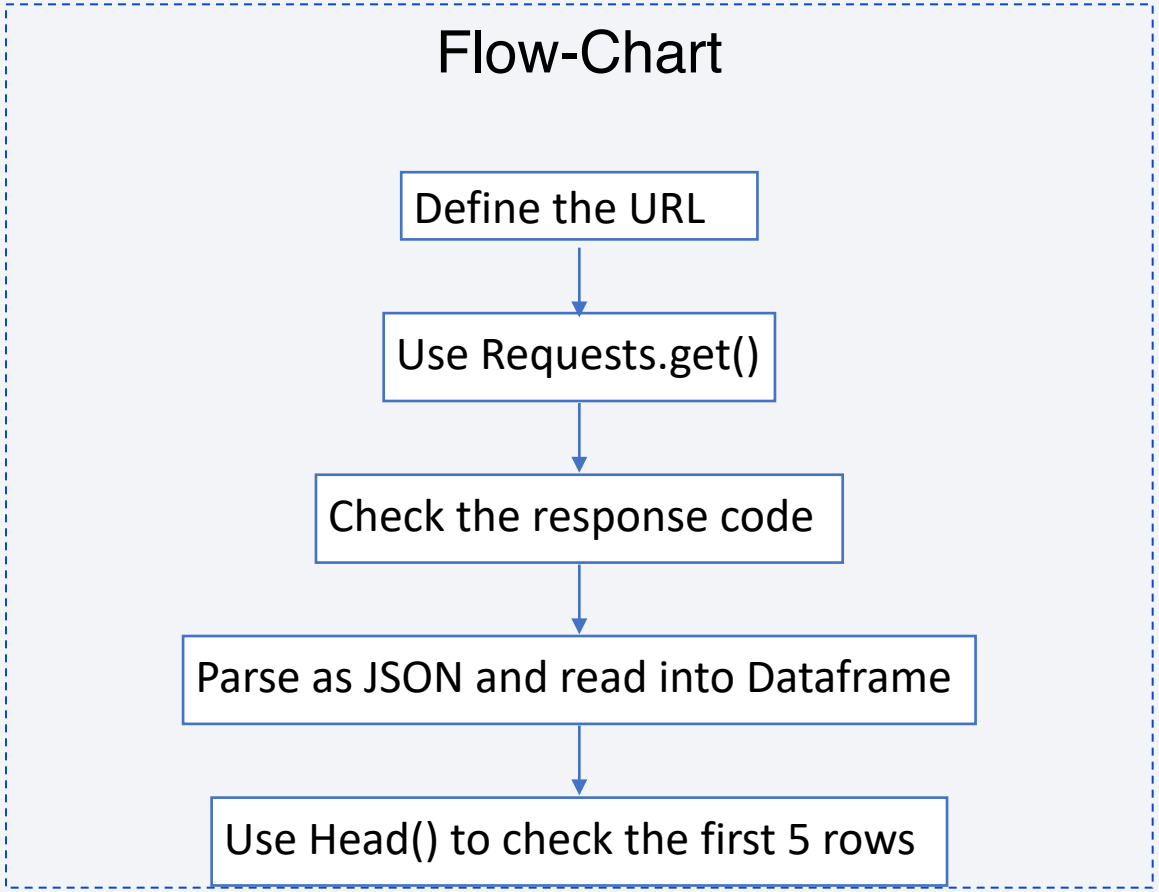
- Two methods of data collection were used:
  - Using the SpaceX API, and Webscraping historical data from an html page.

- Steps required to collect data via the SpaceX API:
  1. Define the URL spacex_url=“https://api.spacexdata.com/v4/launches/past”,
  2. Get the data using “response = requests.get(spacex_url)”
  3. Check the response code to confirm that the data loaded (response code 200)
  4. Parse the data as JSON and read into a Pandas data-frame “data = pd.read_json(spacex_url)”
  5. Check the first 5 rows of data using “data.head()”

- Steps required to collect data via Webscraping:
  1. Define the static URL that holds the data.
  2. Get the data using “response = requests.get(url)
  3. Check the response code to confirm that the data loaded (response code 200)
  4. Create a BeautifulSoup object from the response
  5. Find all the table objects using html_tables = soup.find_all(‘table’). The third table has the data we want.
  6. Get all the column names (the <th> elements
  7. Create a list object for each column and iterate through each table row and append to each list object
  8. Finally create a Dataframe from all the list objects and save as a CSV file.

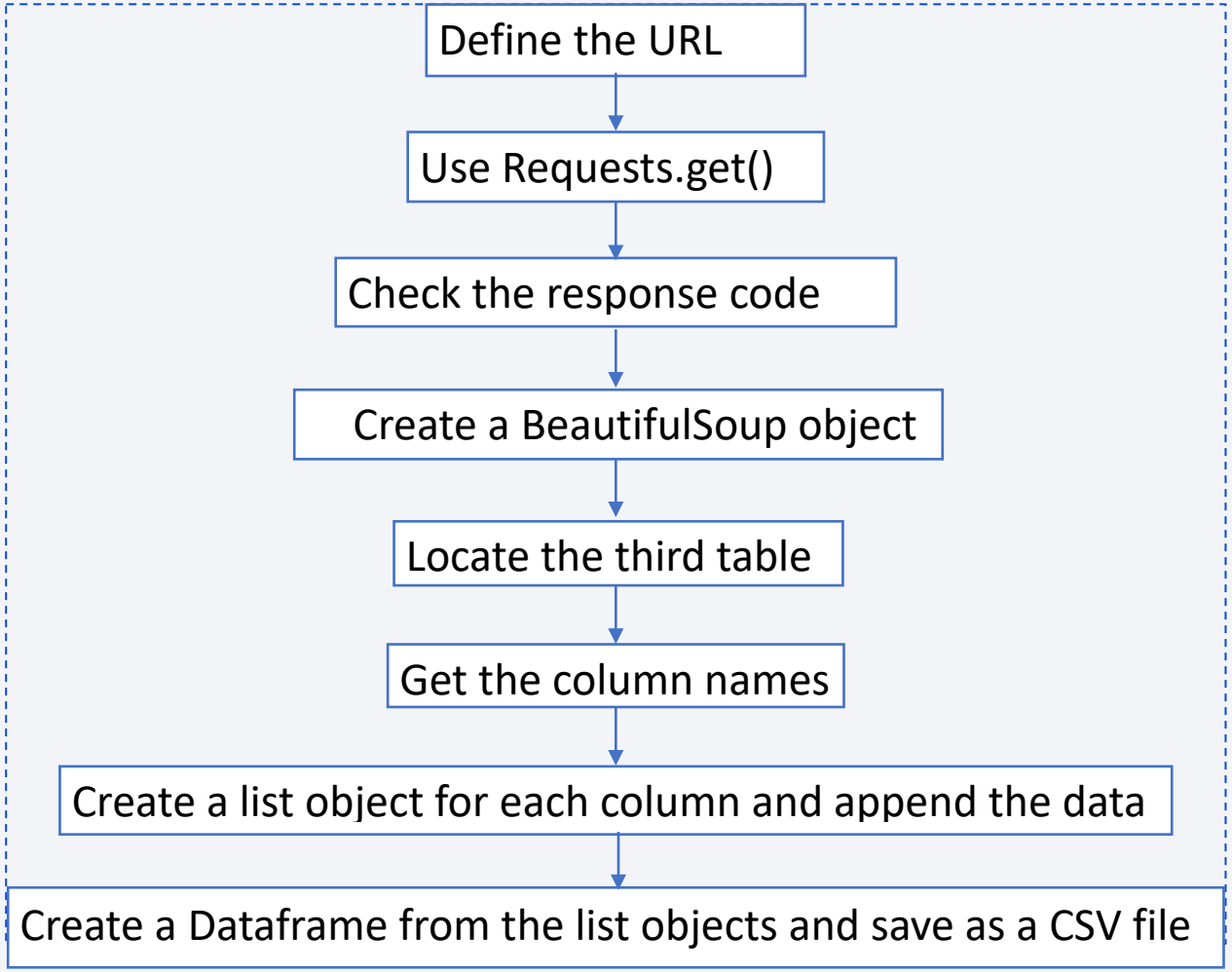
Data Collection – SpaceX API

- GitHub URL of the completed SpaceX API calls notebook: https://github.com/mcintst2/capstone/blob/0a32d6aec64d9a74c9a49e627f0cc610a5dc9cd9/jupyter-labs-spacex-data-collection-api.ipynb



Data Collection - Scraping

- GitHub URL of the completed web scraping notebook: https://github.com/mcintst2/capstone/blob/0a32d6aec64d9a74c9a49e627f0cc610a5dc9cd9/jupyter-labs-webscraping.ipynb



Data Wrangling

Once the data was obtained via the SpaceX API and web scraping, we needed to clean the data, using the following steps:

- The data was filtered to only include Falcon 9 launches,
- Missing values in ‘Payload Mass’ were replaced by the mean value, and
- The outcome for each flight was labelled either as ‘1’ (successful) or ‘0’ (unsuccessful).

- GitHub URL of the completed data wrangling related notebook: https://github.com/mcintst2/capstone/blob/0a32d6aec64d9a74c9a49e627f0cc610a5dc9cd9/Data%20Wrangling.ipynb

EDA with Data Visualization

The following charts were plotted in order to explore various parameters related to launch success.
For the scatter-plots success was coloured red and failure coloured blue.

- Flight number vs payload mass
- Flight number vs launch site
- Payload mass vs launch site
- Orbit vs success rate
- Flight number vs orbit
- Payload mass vs orbit
- Date vs success rate

- GitHub URL of the completed EDA with data visualization notebook: https://github.com/mcintst2/capstone/blob/0a32d6aec64d9a74c9a49e627f0cc610a5dc9cd9/jupyter-labs-eda-dataviz.ipynb.jupyterlite-3.ipynb

EDA with SQL

- With the data loaded in a mysql table, the following queries were run to explore the data:

- `SELECT * FROM SpaceX.spacex` - returning 101 rows
- `SELECT DISTINCT Launch_Site FROM SpaceX.spacex` - get the unique launch site names)
- `SELECT * FROM SpaceX.spacex WHERE Launch_Site LIKE %s LIMIT 5` - with parameter CCA% to only get data for launch sites starting with 'CCA'
- `SELECT SUM(PAYLOAD_MASS__KG_) FROM SpaceX.spacex WHERE Customer = 'NASA (CRS)'` - Display average payload mass carried by booster version F9 v1.1
- `SELECT MIN(STR_TO_DATE(Date,%s)), `Landing _Outcome` FROM SpaceX.spacex WHERE `Landing _Outcome` = 'Success (ground pad)` - when the first successful landing outcome in ground pad was achieved.
- `SELECT Booster_Version FROM SpaceX.spacex WHERE `Landing _Outcome` = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 6000` - the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
- `SELECT COUNT(*),Mission_Outcome FROM SpaceX.spacex GROUP BY Mission_Outcome` - the total number of successful and failure mission outcomes
- `SELECT Booster_Version FROM SpaceX.spacex WHERE PAYLOAD_MASS__KG_ IN (SELECT MAX(PAYLOAD_MASS__KG_) FROM SpaceX.spacex)` - the names of the booster_versions which have carried the maximum payload mass.
- `SELECT `Landing _Outcome`, Booster_Version, Launch_Site, Date FROM SpaceX.spacex WHERE `Landing _Outcome` = 'Failure (drone ship)' AND Date LIKE %s` - the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015 (parameter %s = '%2015')
- `SELECT COUNT(*), `Landing _Outcome` FROM SpaceX.spacex WHERE STR_TO_DATE(Date,%s)> STR_TO_Date('2010-06-04',%s) AND STR_TO_DATE(Date,%s)< STR_TO_Date('2017-03-20',%s) GROUP BY `Landing _Outcome` ORDER BY COUNT(*) DESC` - 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, using parameters '%d-%m-%Y','%Y-%m-%d','%d-%m-%Y','%Y-%m-%d'.

- GitHub URL of the completed EDA with SQL notebook: https://github.com/mcintst2/capstone/blob/0a32d6aec64d9a74c9a49e627f0cc610a5dc9cd9/jupyter-labs-eda-sql-coursera.ipynb

Build an Interactive Map with Folium

- Map objects were created in order to explore the location of the various launch sites and their proximities:
  - The launch site locations on a map
  - The success/failed launches for each site
  - The distances between a launch site to its proximities
- GitHub URL of your completed interactive map with Folium map: https://github.com/mcintst2/capstone/blob/0a32d6aec64d9a74c9a49e627f0cc610a5dc9cd9/lab_jupyter_launch_site_location.jupyterlite-5.ipynb

Build a Dashboard with Plotly Dash

- The following interactive charts were added:
  - Pie chart of launch location vs success and failure
  - Scatter chart to show the correlation between payload and launch success

  Using interactive charts allows you to compare the results for each site.
- GitHub URL of the completed Plotly Dash lab:https://github.com/mcintst2/capstone/blob/0a32d6aec64d9a74c9a49e627f0cc610a5dc9cd9/spacex_dash_app-4.py

Predictive Analysis (Classification)

- In order to create a model to predict the success of future launches based on the previously gathered data the following steps were taken:

1. Perform exploratory Data Analysis and determine Training Labels

   - create a column for the class
   - Standardize the data
   - Split the data into training data and test data

2. Find best Hyper-parameter for the following models: SVM, Classification Trees and Logistic Regression.

   - Find the method which performs best using test data

- GitHub URL of the completed predictive analysis lab: https://github.com/mcintst2/capstone/blob/0a32d6aec64d9a74c9a49e627f0cc610a5dc9cd9/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

Results

- Exploratory data analysis results

- KSLC-39A has a success rate of 77%.
- Success gets better as the number of launches increases.
- for the VAFB-SLC launchsite there are no rockets launched for heavypayload mass(greater than 10000).
- ESL1, GEO, HEO and SSO are the most successful orbits.
- 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.
- With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS.
- the sucess rate since 2013 keeps increasing.
- The following features will be used to predict the success of future launches:

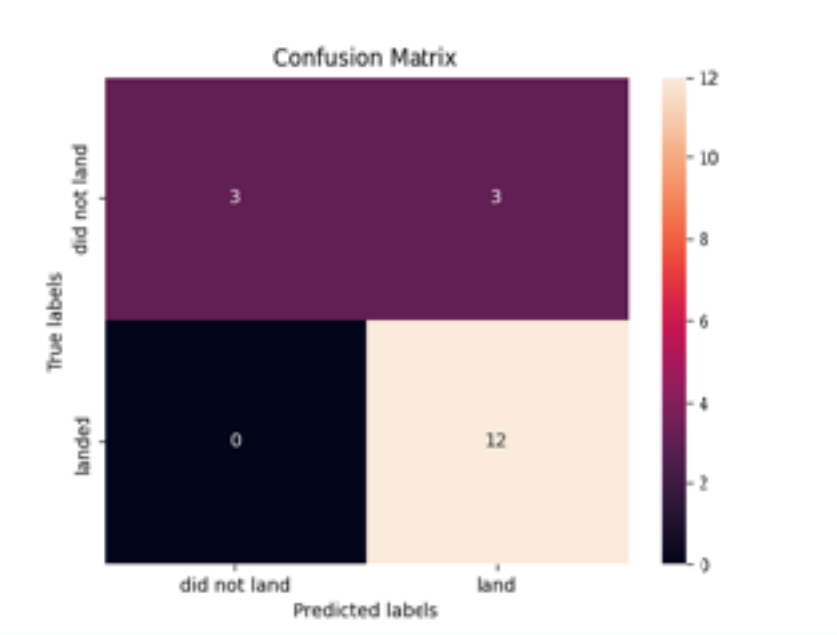
FlightNumber, PayloadMass, Orbit, LaunchSite, Flights, GridFins, Reused, Legs, LandingPad, Block, ReusedCount and Serial

- Interactive analytics demo in screenshots

- Predictive analysis results

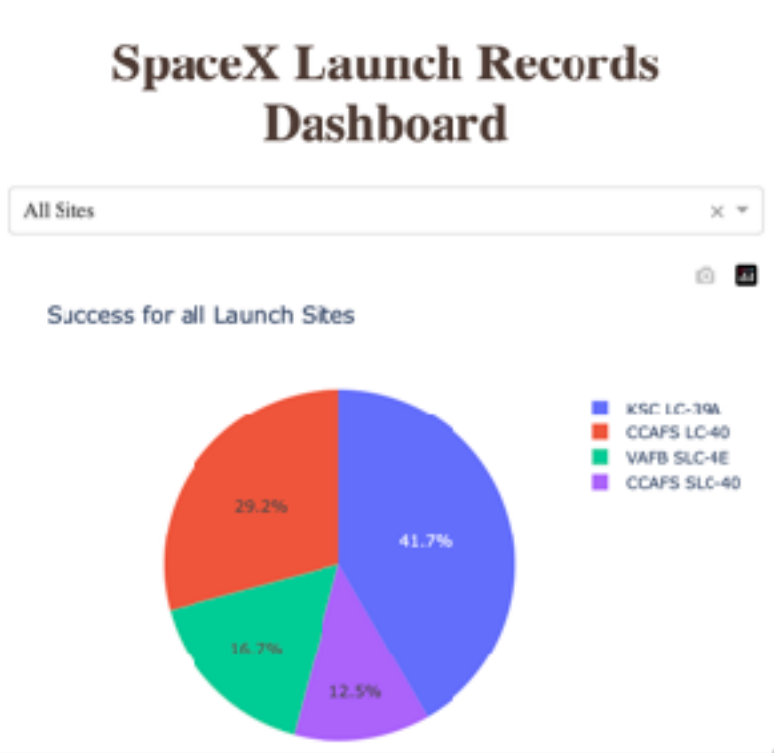
Three models (logistic regression, support vector machine and k nearest neighbours) had the highest accuracy at 83%, with only 3 false positives

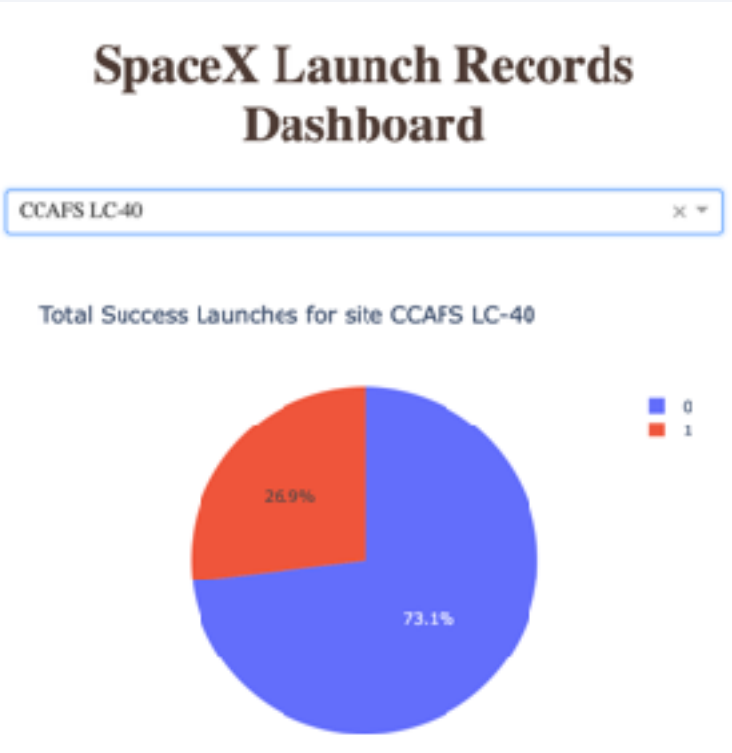
Confusion Matrix for k nearest neighbour model

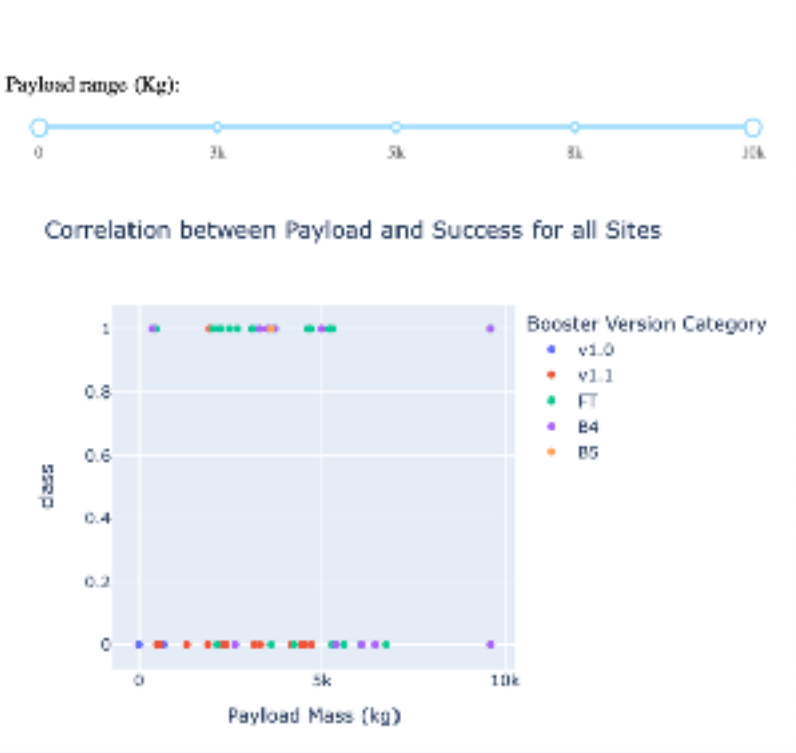


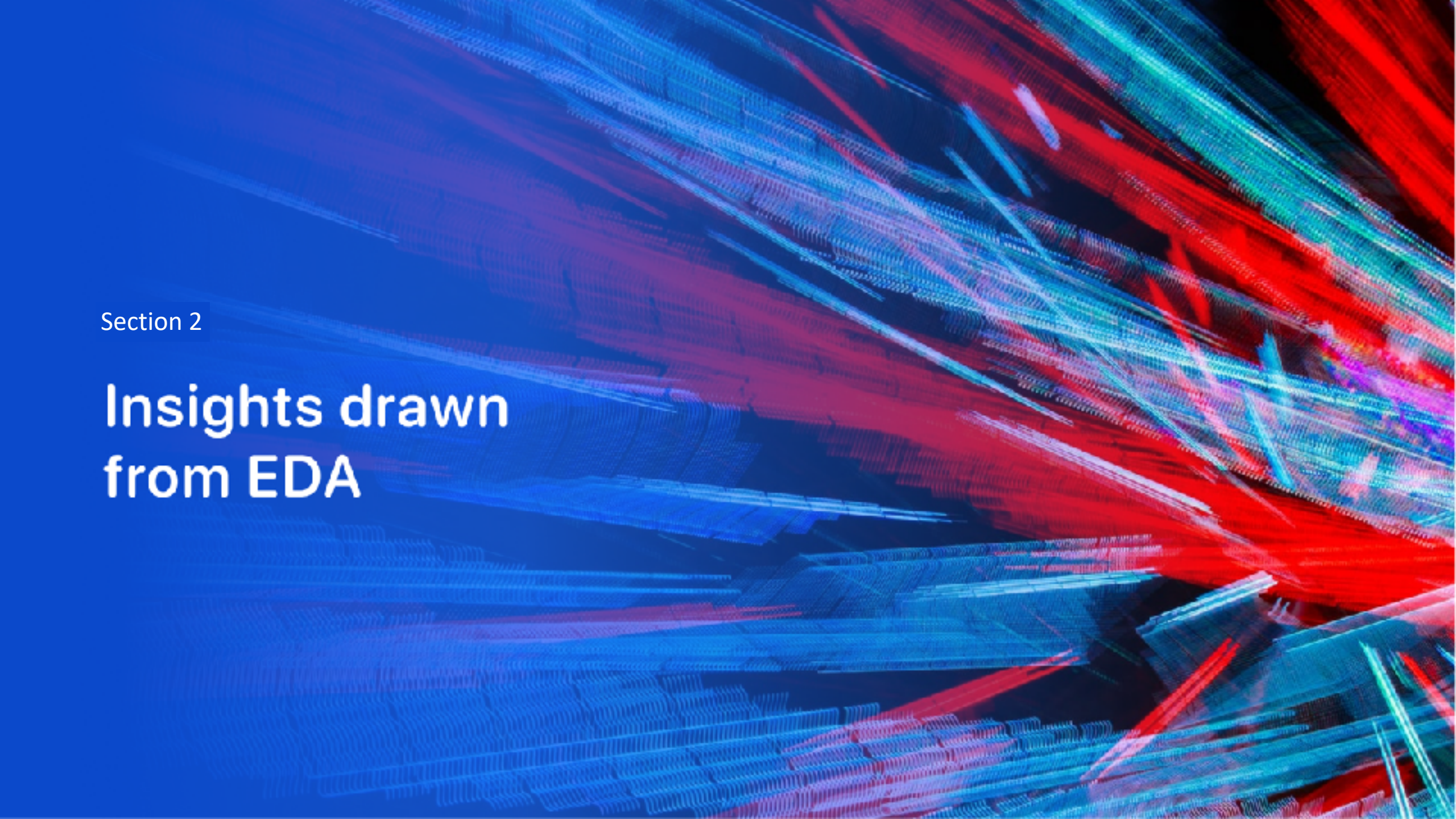
Results (continued)

- Interactive analytics demo in screenshots



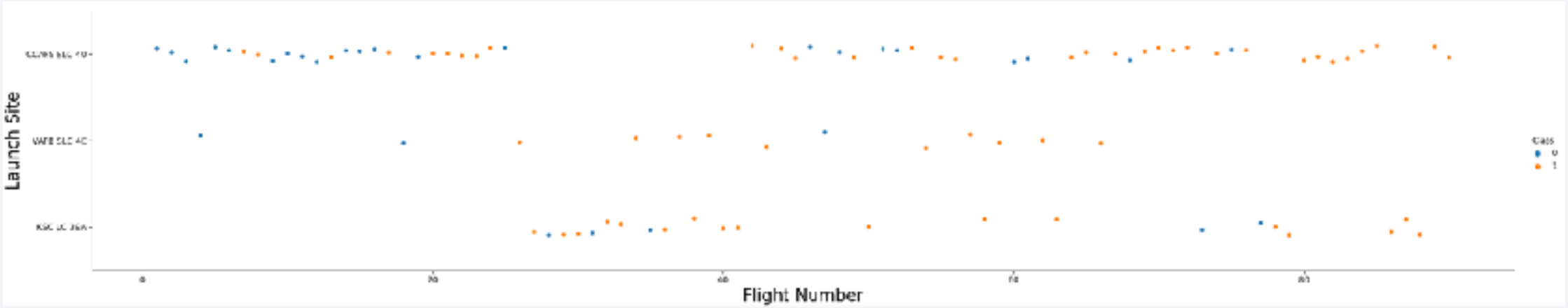





Section 2
Insights drawn
from EDA

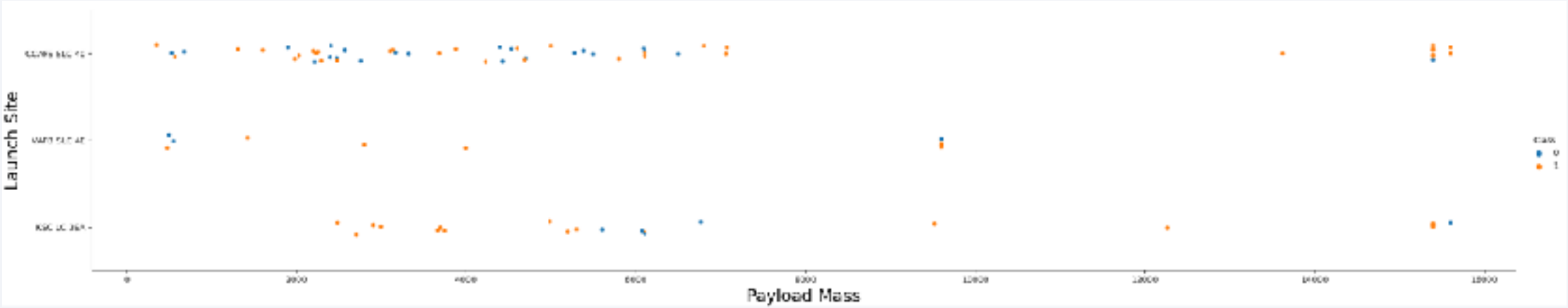
Flight Number vs. Launch Site

- The following scatter plot shows the relative success rate per site. As the number of flights increases, so does the success rate (class = 1)



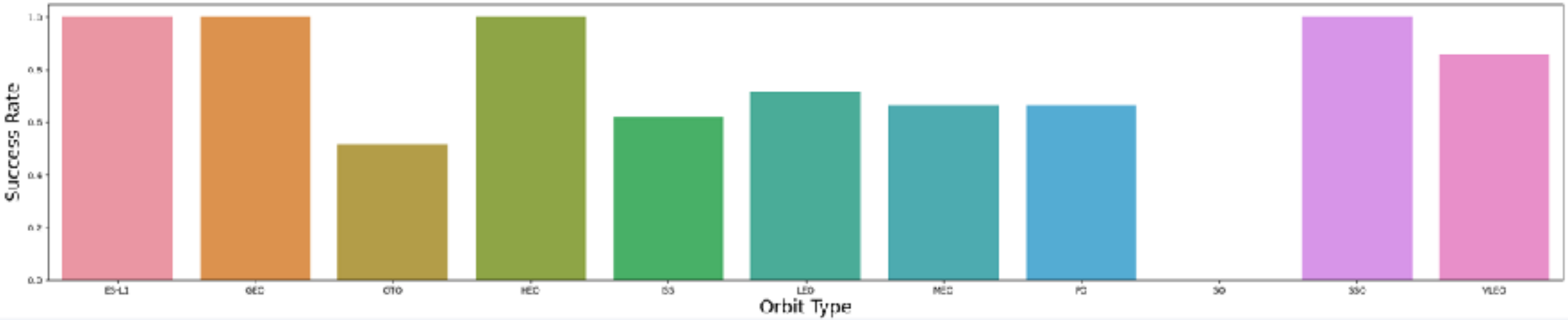
Payload vs. Launch Site

- The following scatter plot shows that for the VAFB-SLC launch-site there are no rockets launched for heavy payload mass (greater than 10000 kg).



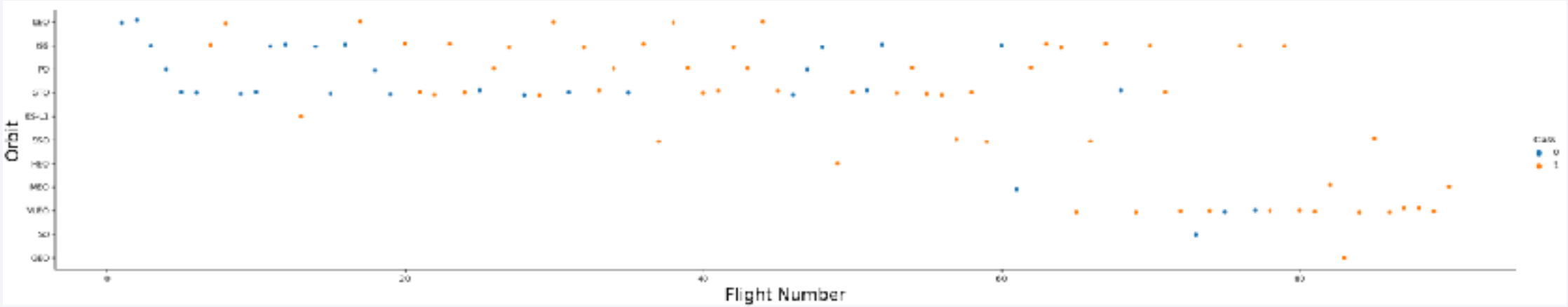
Success Rate vs. Orbit Type

- The following bar chart shows that the orbits ES-L1, GEO, HEO and SSO have a higher success rate



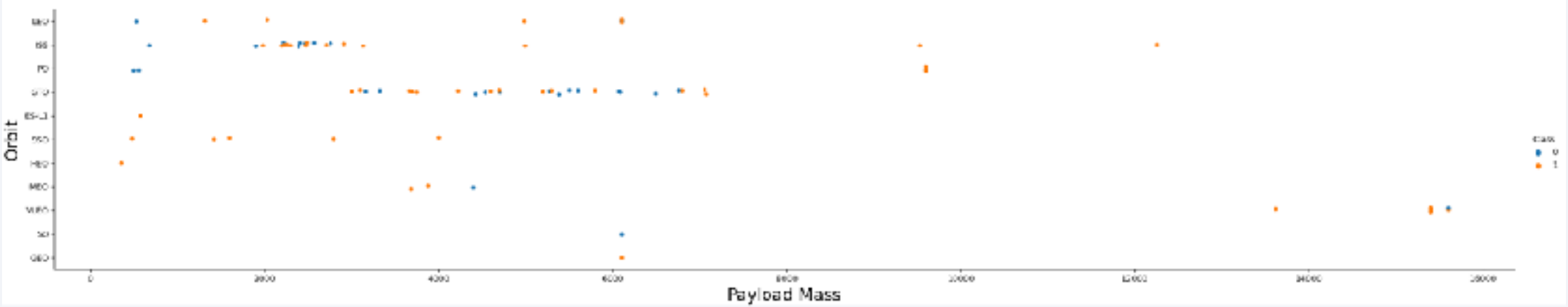
Flight Number vs. Orbit Type

- The following scatter plot shows 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.



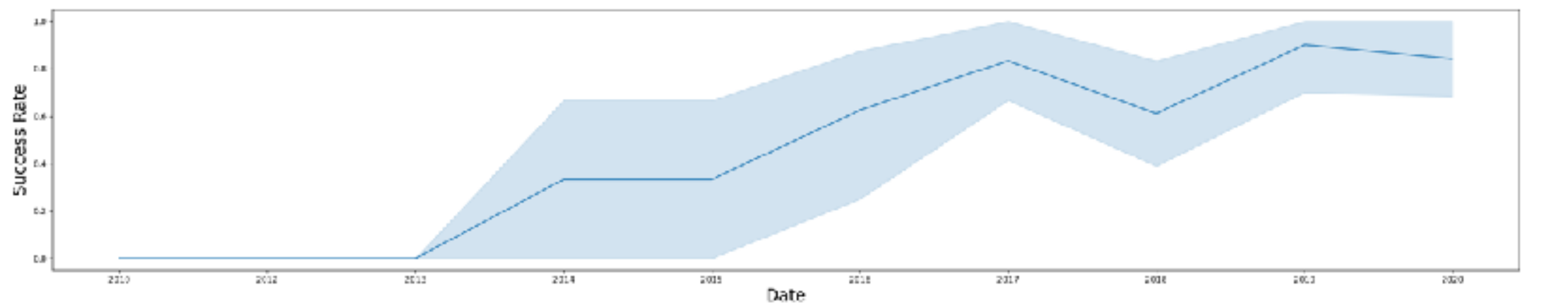
Payload vs. Orbit Type

- The following scatter plot shows that with heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS.



Launch Success Yearly Trend

- The following line chart shows that success has been increasing since 2013



All Launch Site Names

- Use an SQL query to find out the unique launch site names

```
pd.read_sql('SELECT DISTINCT Launch_Site FROM SpaceX.spacex', cnx)
```

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

Launch Site Names Begin with 'CCA'

- The following SQL query was used to find 5 records where the launch sites begin with `CCA`

```
sql_string = """SELECT * FROM SpaceX.spacex WHERE Launch_Site LIKE %s LIMIT 5"""
pd.read_sql(sql_string, con=cnx, params=("CCA%",))
```

- Results

	Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing _Outcome
0	04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
1	08-12-2010	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of...	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2	22-05-2012	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
3	08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
4	01-03-2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

- The following SQL query was used to calculate the total payload carried by boosters from NASA

```
sql_string = """SELECT SUM(PAYLOAD_MASS__KG_) FROM SpaceX.spacex WHERE Customer = 'NASA (CRS)'"""
pd.read_sql(sql_string, con=cnx)
```

- The total payload was 45,596 kg

Average Payload Mass by F9 v1.1

- The following SQL query was used to calculate the average payload mass carried by booster version F9 v1.1

```
sql_string = """SELECT AVG(PAYLOAD_MASS__KG_) FROM SpaceX.spacex WHERE Booster_Version LIKE %s"""
pd.read_sql(sql_string, con=cnx, params=("F9 v1.1%",))
```

- The average payload mass was 2535 kg

First Successful Ground Landing Date

- The following SQL query was used to find the dates of the first successful landing outcome on ground pad

```
sql_string = """SELECT MIN(STR_TO_DATE(Date,%s)), `Landing _Outcome` FROM SpaceX.spacex WHERE
`Landing _Outcome` = 'Success (ground pad)'"""
pd.read_sql(sql_string, con=cnx, params=('%d-%m-%Y',))
```

- The first successful landing outcome on ground pad was 2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- The following SQL query was used to find the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000

```
sql_string = """SELECT Booster_Version FROM SpaceX.spacex WHERE `Landing _Outcome` = 'Success (drone
ship)' AND PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 6000"""
pd.read_sql(sql_string, con=cnx)
```

- The booster names were as follows:
- F9 FT B1022
- F9 FT B1026
- F9 FT B1021.2
- F9 FT B1031.2

Total Number of Successful and Failure Mission Outcomes

- The following query was used to find the total number of successful and failure mission outcomes

```
sql_string = """SELECT COUNT(*),Mission_Outcome FROM SpaceX.spacex GROUP BY Mission_Outcome"""
pd.read_sql(sql_string, con=cnx)
```

- Results were as follows:

	COUNT(*)	Mission_Outcome
0	98	Success
1	1	Failure (in flight)
2	1	Success (payload status unclear)
3	1	Success

Boosters Carried Maximum Payload

- The following SQL query was used to find the names of the booster which have carried the maximum payload mass

```
sql_string = """SELECT Booster_Version FROM SpaceX.spacex WHERE PAYLOAD_MASS__KG_ IN (SELECT
MAX(PAYLOAD_MASS__KG_) FROM SpaceX.spacex)"""
pd.read_sql(sql_string, con=cnx)
```

- Results were as follows:

  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

- The following SQL query was used to find the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

```
sql_string = """SELECT `Landing _Outcome`, Booster_Version, Launch_Site, Date FROM SpaceX.spacex
WHERE `Landing _Outcome` = 'Failure (drone ship)' AND Date LIKE %s"""
pd.read_sql(sql_string, con=cnx, params=('%2015',))
```

- Results were as follows:

	Landing _Outcome	Booster_Version	Launch_Site	Date
0	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40	10-01-2015
1	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40	14-04-2015

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

- The following SQL query was used to 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_string = """SELECT COUNT(*), `Landing _Outcome` FROM SpaceX.spacex WHERE STR_TO_DATE(Date,%s)>
STR_TO_Date('2010-06-04',%s) AND STR_TO_DATE(Date,%s)< STR_TO_Date('2017-03-20',%s) GROUP BY `Landing
_Outcome` ORDER BY COUNT(*) DESC """
pd.read_sql(sql_string, con=cnx, params=('%d-%m-%Y','%Y-%m-%d','%d-%m-%Y','%Y-%m-%d',))
```

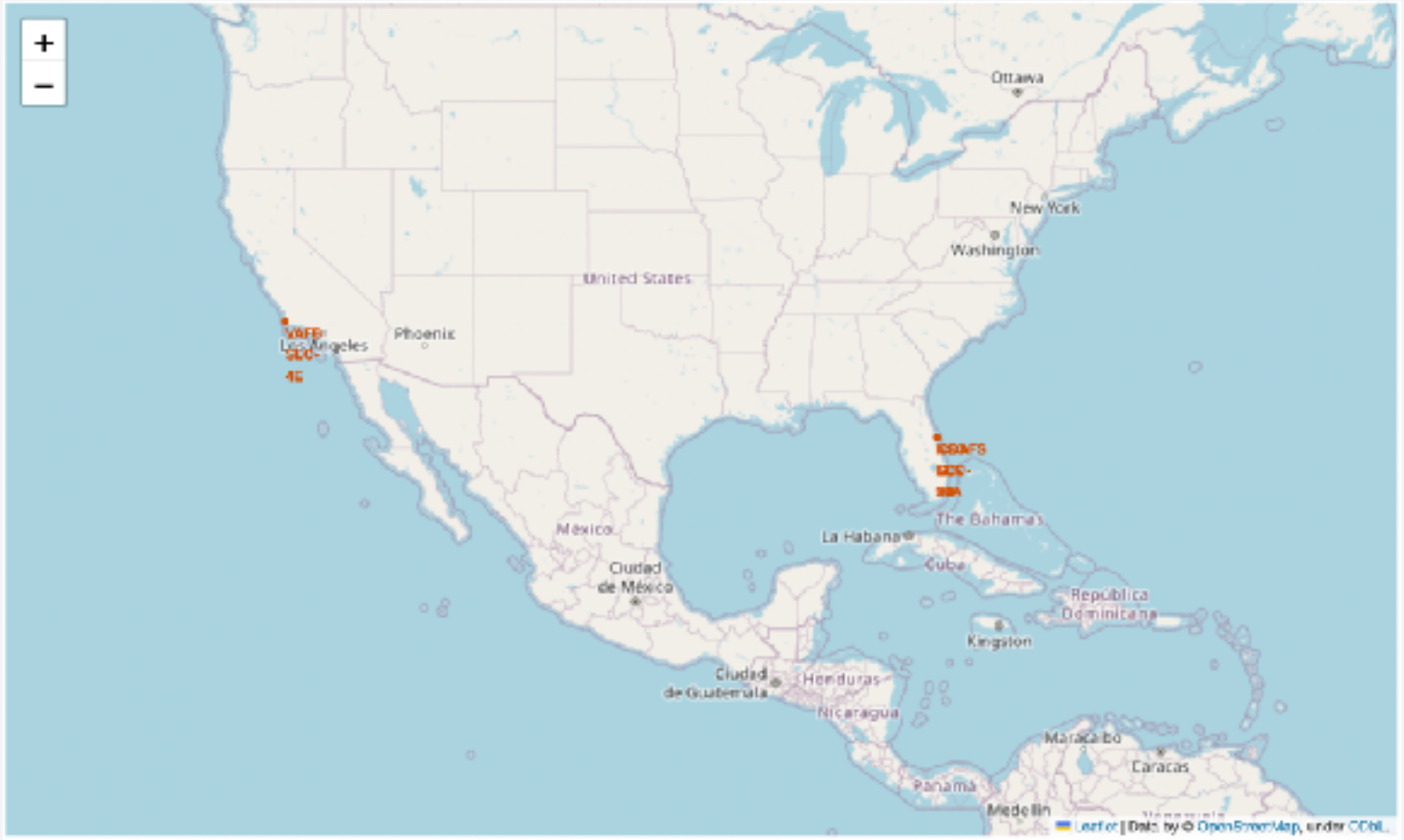
- The results were as follows:

	COUNT(*)	Landing _Outcome
0	10	No attempt
1	5	Failure (drone ship)
2	5	Success (drone ship)
3	3	Controlled (ocean)
4	3	Success (ground pad)
5	2	Uncontrolled (ocean)
6	1	Failure (parachute)
7	1	Precluded (drone ship)

Section 3

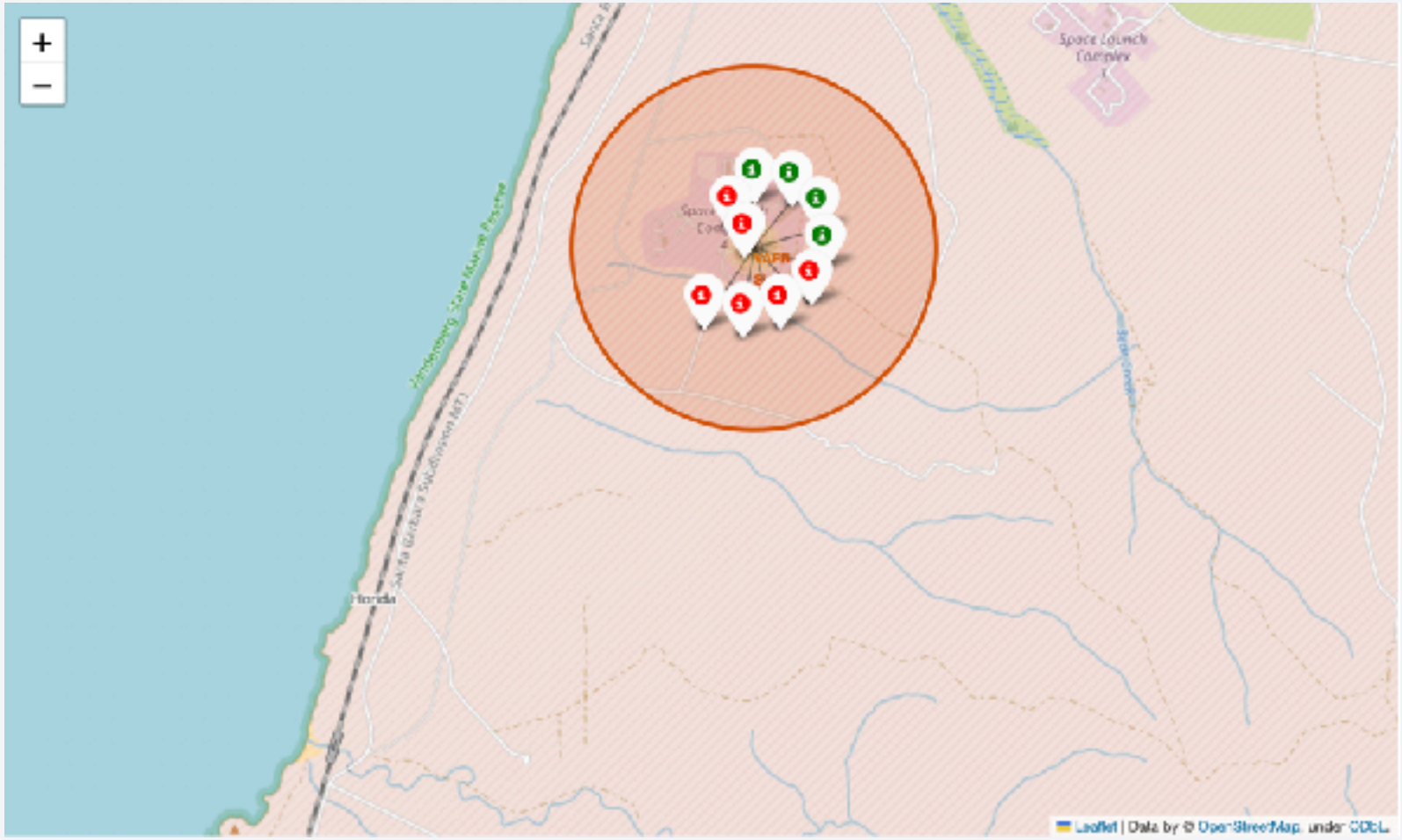
Launch Sites Proximities Analysis

Launch Site Locations



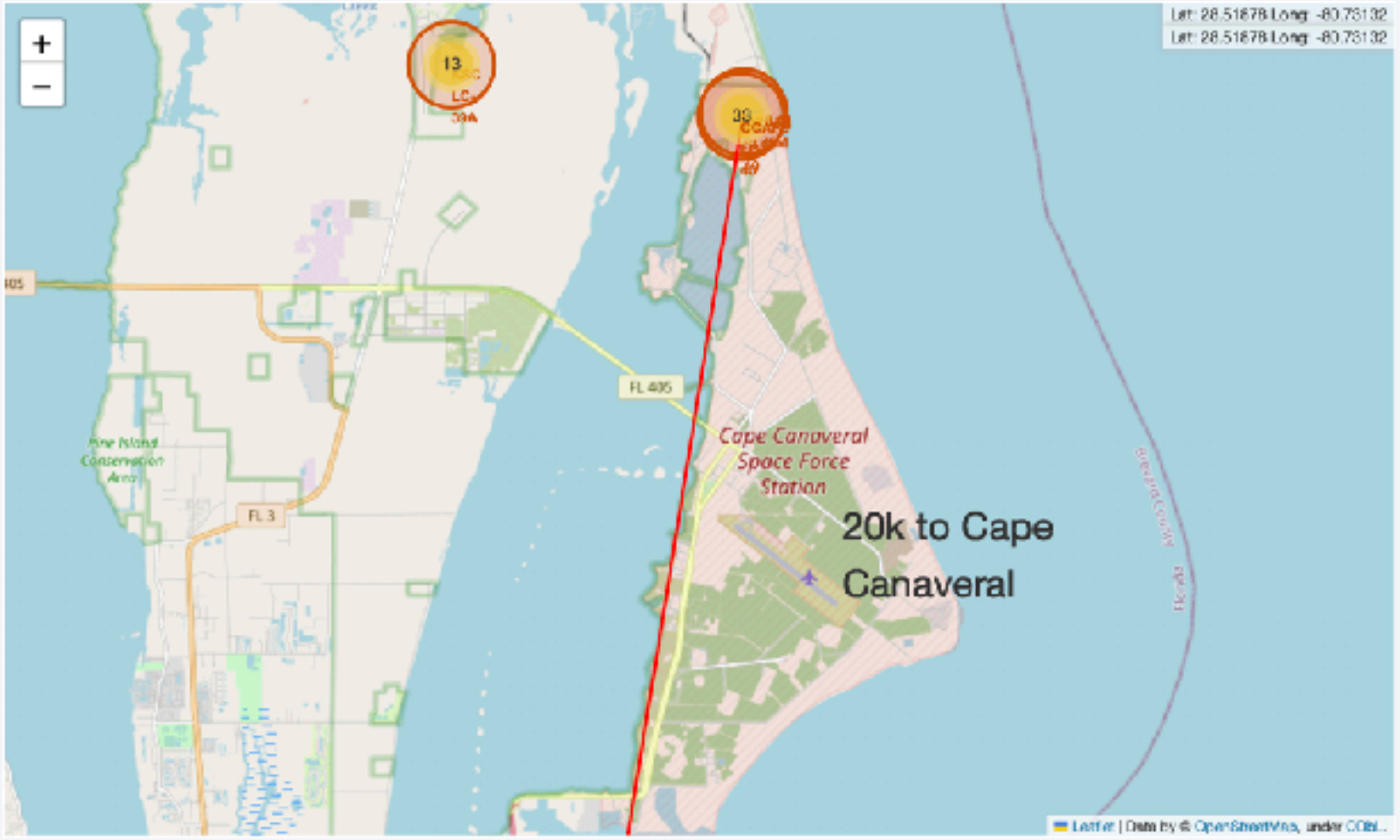
- All launch sites are on the coast.
- These are not located on the Equator, which would actually give the rockets additional speed.

Launch Outcomes for VAFB SLC-4E



- For the VAFB SLC-4E launch-site, the map shows the number of successful launches in green, versus the number of failed launches in red.

Launch Site Proximities

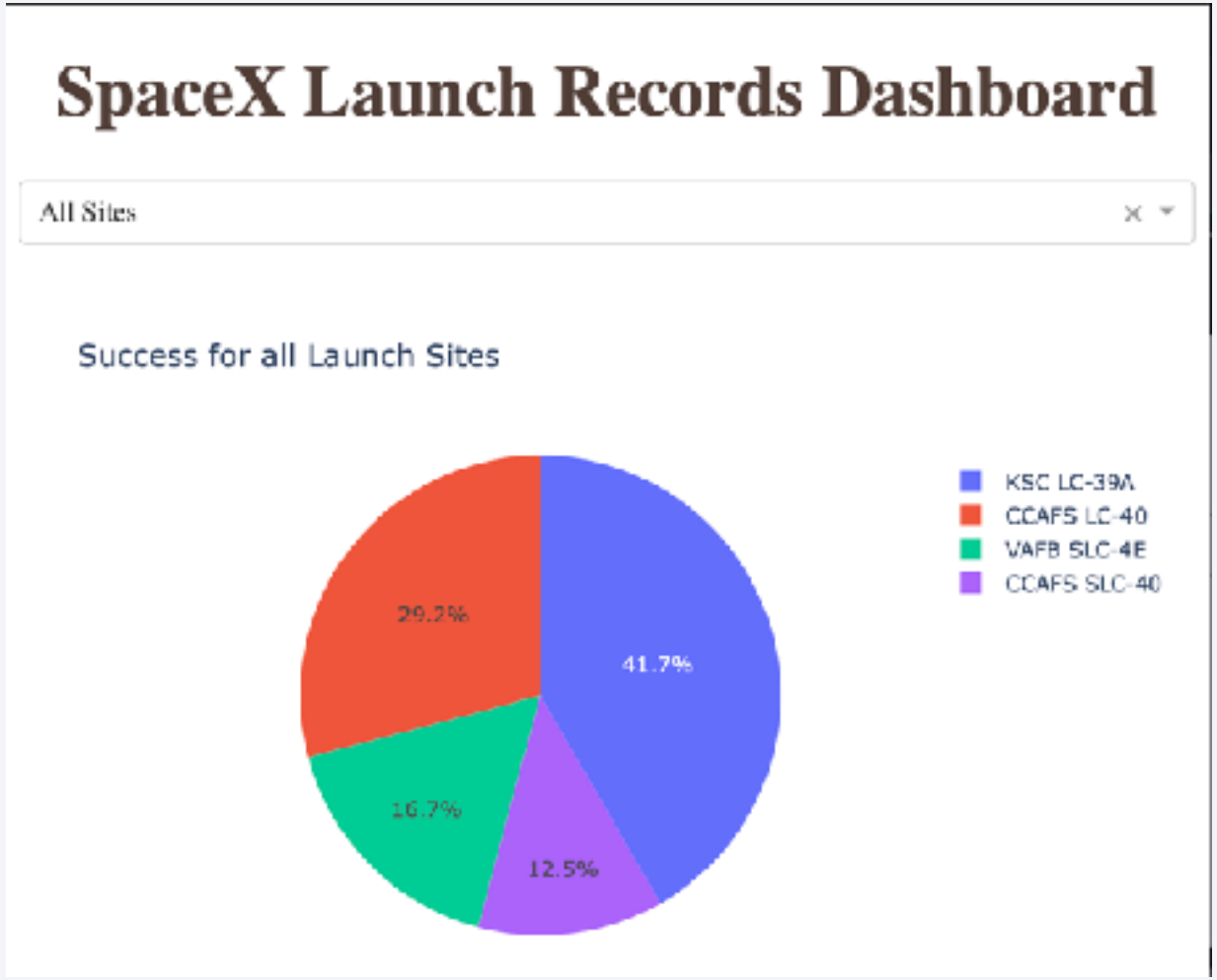


- The launch-sites are either close to the East Coast, or have uninhabited areas to the East. This is because rocket launches go East to save fuel.
- The sites in Florida have Cape Canaveral to the South where many of their workers will be based.

Section 4

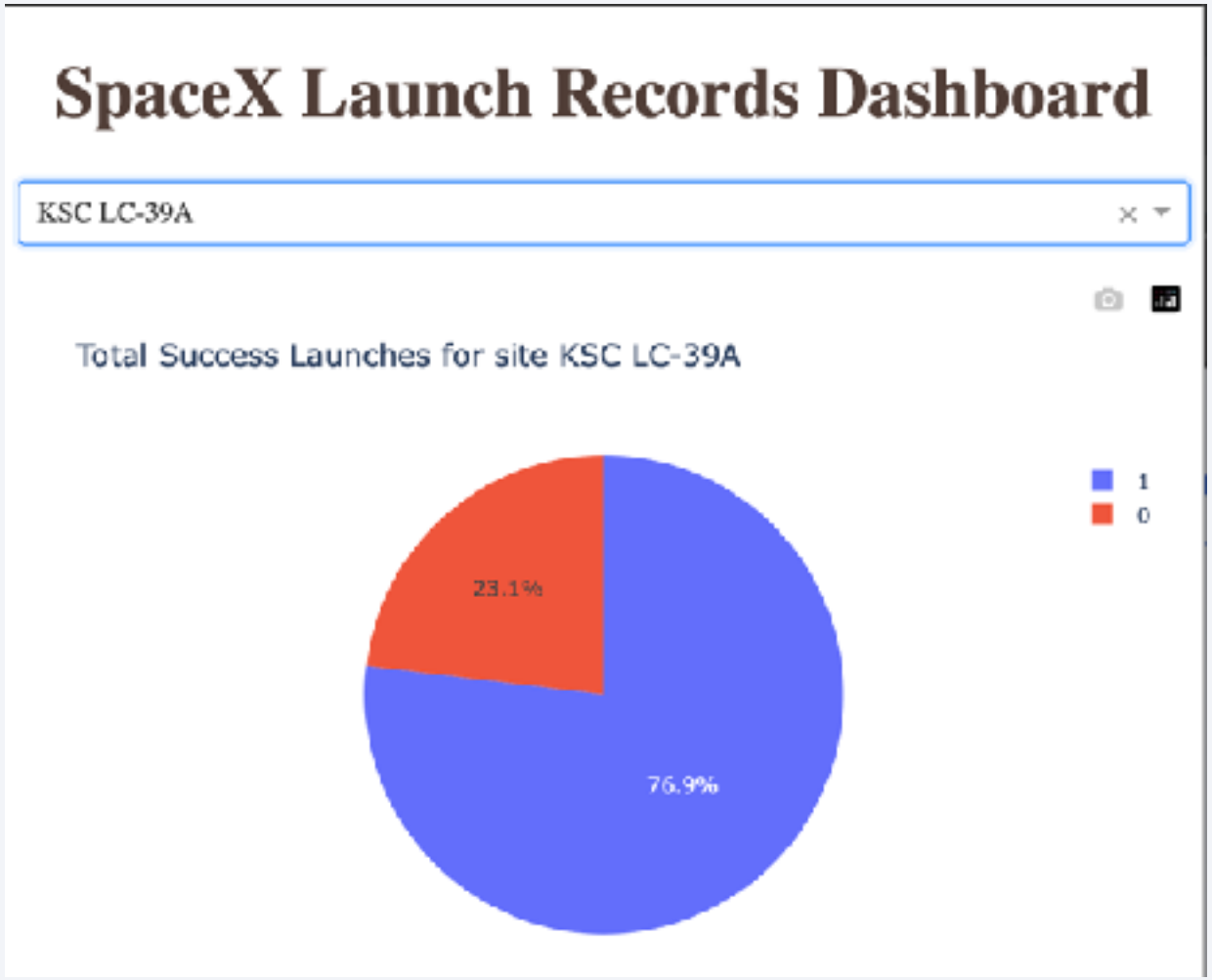
Build a Dashboard with Plotly Dash

Launch Success Counts for All Sites



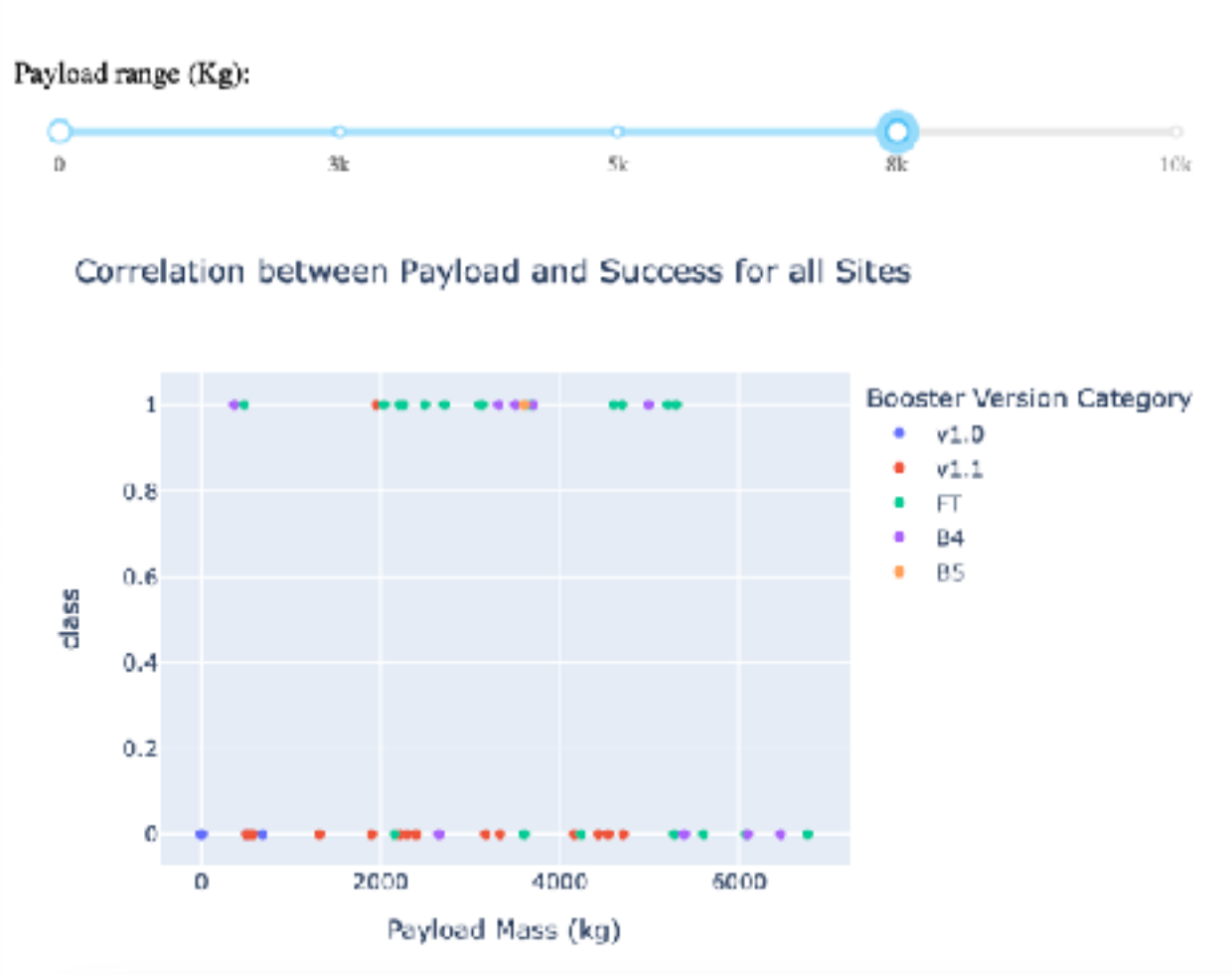
- For the four launch-sites, KSC LC-39A has the highest launch success ratio

Launch Success Ratio for the Most Successful Site



- The launch-site KSC LC-39A had successful launches 76.9% of the time.

Booster Version and Payload

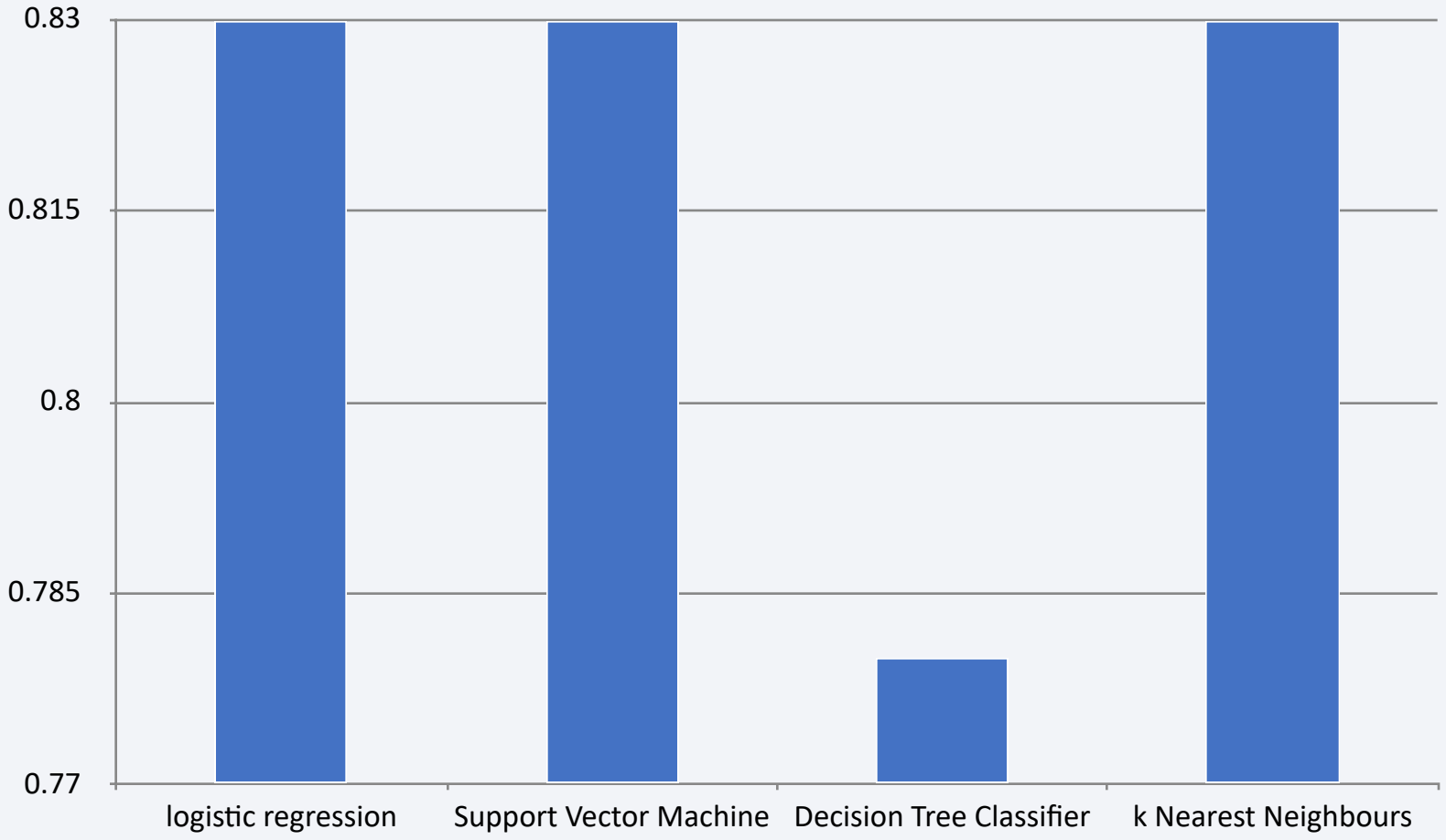


- The FT Booster version showed to be successful with payloads between 2000 kg and 6000 kg

Section 5

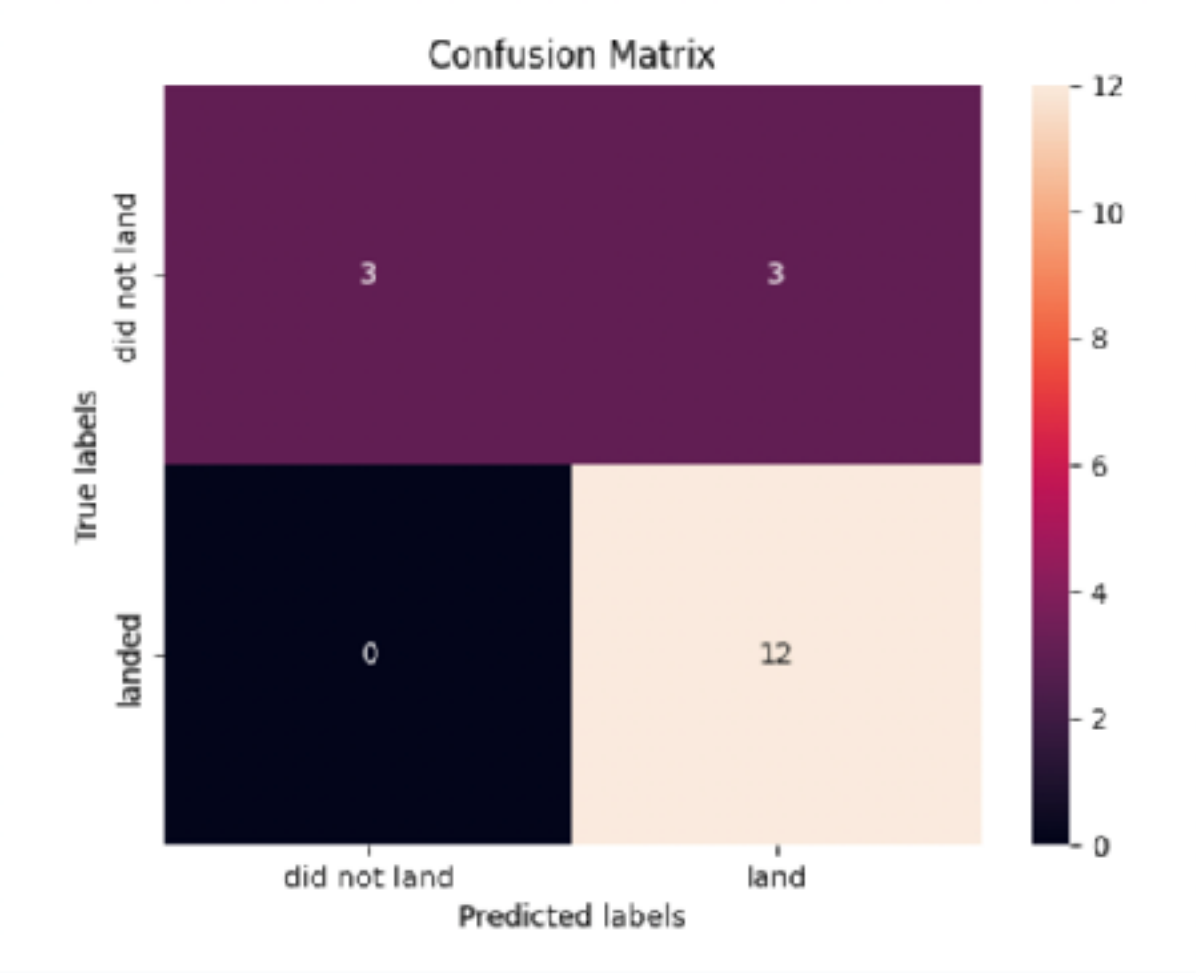
Predictive Analysis (Classification)

Classification Accuracy



- Three of the models had the same accuracy

Confusion Matrix



- All three models (logistic regression, support vector machine and k nearest neighbours) had only 3 false positives

Conclusions

- Launch-site KSLC-39A has the highest success rate at 77%.
- Success gets better as the number of launches increases.
- ESL1, GEO, HEO and SSO are the most successful orbits.
- 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.
- With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS.
- the success rate since 2013 keeps increasing.

Appendix

- Link to Github for project: https://github.com/mcintst2/capstone

Thank you!