Applied Data Science Capstone Project

Paraskevi-Spyridoula KITSAKI

09/2021

OUTLINE

1 Executive Summary

Scope of the project

Train a machine learning model and use public information to predict if the Falcon 9 first stage will land successfully and SpaceX will reuse the first stage .Determine the price of each launch.

Methodology

Data Collection	Gather relevant raw data using SpaceX REST API and Web scraping wiki pages
Data Wrangling	Process data to improve quality
Exploratory Data Analysis	Explore the processed data using SQL, Pandas and Matplotlib
Data Visualization	Build an interactive map and dashboard using with Folium and Plotly Dash to perform interactive visual analytics
Predictive Analysis	Build a machine learning pipeline to predict if the first stage of the Falcon 9 lands successfully

Results

- ❑ Exploratory data analysis results.
- ❑ Interactive Map
- ❑ Dashboard
- ❑ Predictive analysis results

2 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.

Space Y rocket company would like to compete with SpaceX founded by Billionaire industrialist Allon Musk.

Gathering information about Space X and creating dashboards we will predict if the Falcon 9 first stage will land successfully and determine the cost of a launch.

the first stage will land, we can determine the cost of a launch.

This information can be used if Space Y company wants to bid against SpaceX for a rocket launch.

Problems we want to find answers

- What features influence the successful landing of the first stage and to what extend?
- Can we automatically predict if
- the first stage can be reused with a high success rate?
- What conditions determine the best results and how to choose an optimal launch site?

3 Methodology

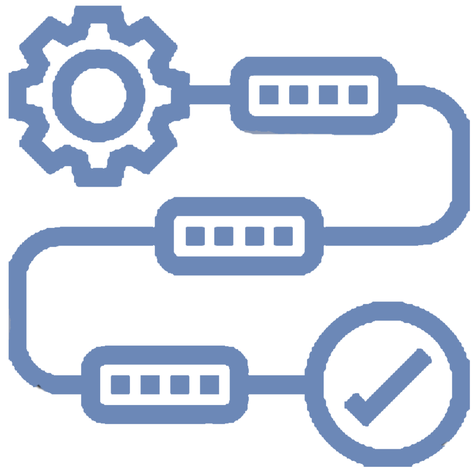
- **Data collection methodology:**
  - Request to the SpaceX API, to gather SpaceX launch data
  - web scraping related Wiki pages collect Falcon 9 historical launch records from a Wikipedia page titled List of Falcon 9 and Falcon Heavy launches

- **Perform data wrangling** (transform raw data into a clean dataset which provides meaningful data on the situation we are trying to address)
  - Remove irrelevant columns
  - Deal with missing values

- **Perform exploratory data analysis** (EDA) using visualization and SQL
  - Create Scatter Plots, Bar Chart and Line Chart to find some patterns in the data and show relationships between variables
  - Perform SQL queries to explore data

- **Perform interactive visual analytics using Folium and Plotly Dash**

- **Perform predictive analysis using classification models**
  - How to build, tune, evaluate classification models



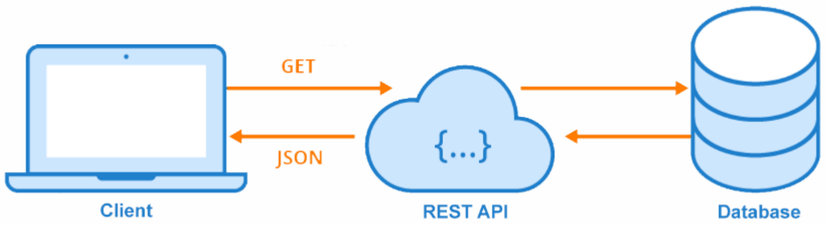
Methodology

3

Data collection

- We will be working with SpaceX launch data that is gathered from an API, specifically the **SpaceX REST API** (https://api.spacexdata.com/v4/).This API will give us data about launches, including information about the rocket used, payload delivered, launch specifications, landing specifications, and landing outcome. We will perform a get request to get the data from the API and then convert the response which will be in the form of a JSON, to a dataframe.

- Another data source for obtaining Falcon 9 Launch data is **web scraping related Wiki pages**. For this project, we will be performing web scraping to collect Falcon 9 historical launch records from a Wikipedia page titled "List of Falcon 9 and Falcon Heavy Launches" (https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches). Then we will be using the Python BeautifulSoup package to web scrape some HTML tables that contain valuable Falcon 9 launch records.



Data collection – SpaceX API (GitHub URL)

1 Request SpaceX launch data

requests.get

API endpoint

https://api.spacexdata.com/v4/launches/past

response

JSON result

2 Convert to data frame using json_normalize

Dataframe

3 Apply functions which use other end points to extract information

`getBoosterVersion(data)` `getLaunchSite(data)` `getPayloadData(data)` `getCoreData(data)`

requests.get

API endpoints

https://api.spacexdata.com/v4/rockets/	https://api.spacexdata.com/v4/launchpads/	https://api.spacexdata.com/v4/payloads/	https://api.spacexdata.com/v4/cores/

response

JSON results

Lists

4 Combine data from requests

Dictionary

Create

Pandas Dataframe

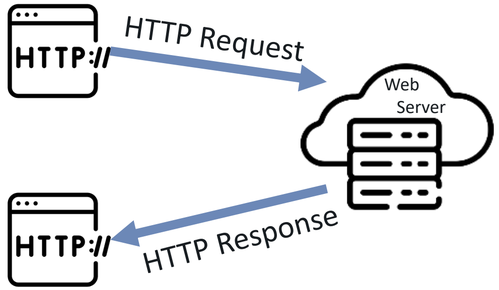
5 Clean Data

- Exclude Falcon 1 launches
- Deal with nulls

Data collection – Web scraping(GitHub URL)

1

Request the Falcon9 Launch Wiki page from its URL



2

Create a BeautifulSoup object

3

Extract all column/variable names from the HTML table header

4

Create a data frame by parsing the launch HTML tables

Data wrangling(GitHub URL)

Perform some Exploratory Data Analysis to find some patterns in the data and determine what would be the label for training supervised models.

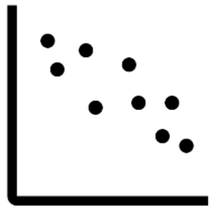
In the data set, there are several different cases where the booster did not land successfully. Sometimes a landing was attempted but failed due to an accident; for example, **True Ocean** means the mission outcome was successfully landed to a specific region of the ocean while **False Ocean** means the mission outcome was unsuccessfully landed to a specific region of the ocean. **True RTLS** means the mission outcome was successfully landed to a ground pad **False RTLS** means the mission outcome was unsuccessfully landed to a ground pad. **True ASDS** means the mission outcome was successfully landed on a drone ship **False ASDS** means the mission outcome was unsuccessfully landed on a drone ship.

We will mainly convert those outcomes into Training Labels with 1 means the booster successfully landed 0 means it was unsuccessful.

1. Calculate the number of launches at each site
2. Calculate the number and occurrence of each orbit
3. Calculate the number and occurrence of mission outcome per orbit type
4. Create a landing outcome label from Outcome column
5. Calculate the success rate

EDA with data visualization(GitHub URL)

Visualizations using Scatter Plots:



- Flight Number vs. Launch Site
- Payload vs. Launch Site
- Flight Number vs. Orbit type
- Payload vs. Orbit type

With this type of chart we can plot one numeric attribute against another numeric attribute and visualize the correlation (the relationship between two variables) between axes. Scatter plot provides a robust analysis of the correlation significance. We can estimate that the correlation relationship is stronger when the data points are concentrated on certain areas, whereas the relationship is weak if they are sparse.

Visualization using Bar Chart:

- Success rate vs. Orbit type

Bar chart compares the measure of categorical dimension. As we can see, comparing the height of each bar gives us a more intuitive perception than looking at the table alone.

Visualization using Line Chart:

- Launch success yearly trend

Line graph indicates trends and developments of numeric data over time. It is commonly used in time series analysis, by visualizing the fluctuation of a numeric variable against a date-type variable. Each line itself is a comparison between one historical time point and another.

EDA with SQL (GitHub URL)

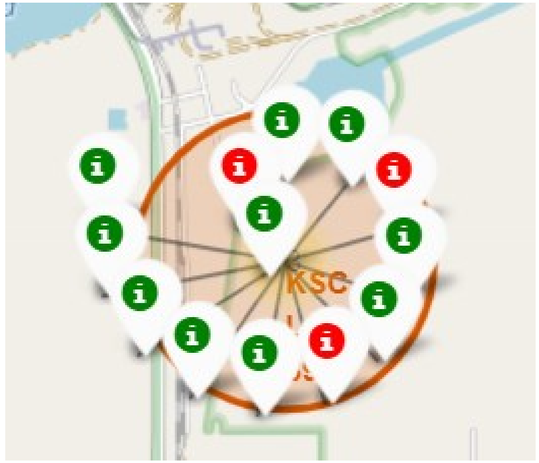
We store the dataset in a database table and perform SQL queries to solve the following tasks:

- Find the names of the unique launch sites in the space mission.
- Find all launch sites begin with `CCA`
- Calculate the total payload carried by boosters from NASA (CRS).
- Calculate the average payload mass carried by booster version F9 v1.1
- Find 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.
- Calculate 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 failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015.
- Rank the count of successful landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order.

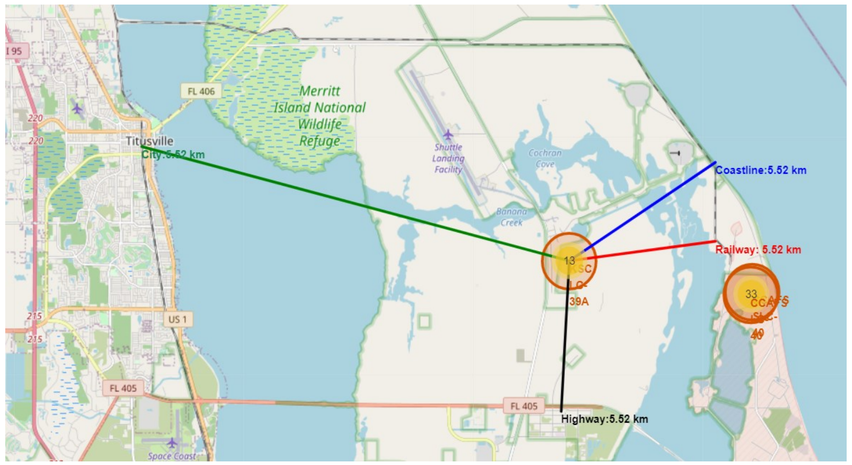
Build an interactive map with Folium (GitHub URL)

In order to visualize launch data on the folium map, circle markers were added at each site's location using site's latitude and longitude coordinates.

Then, the success/failed launches for each site were marked on the map and were created color-labeled markers in marker clusters, which made easy the exploration and identification of launch sites with relatively high success rates.

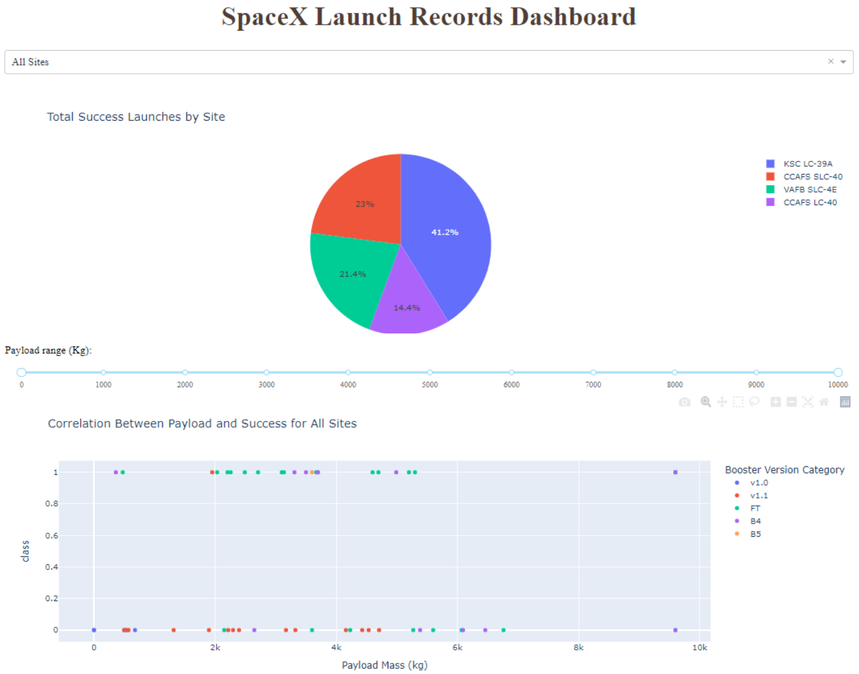


Next, we explored and analyzed the proximities of launch site KSC LC-39A. A MousePosition was added on the map to get coordinate for a mouse over a point on the map. Then folium.Marker objects were created and `folium.PolyLine` objects were used to draw lines between the launch site and its proximities (closest railway, coastline, highway,city)



Build a Dashboard with Plotly Dash (GitHub URL)

Plots/graphs and interactions of the Dashboard



- Launch Site Drop-down Input Component

We have four different launch sites and we would like to first see which one has the largest success count. Then, we would like to select one specific site and check its detailed success rate (class=0 vs. class=1). The dropdown menu let us select different launch sites.

- Callback function to render success-pie-chart based on selected site dropdown

It is used to get the selected launch site from site-dropdown and render a pie chart visualizing launch success counts.

- Range Slider to Select Payload

Used to find if variable payload is correlated to mission outcome. With the slider we can easily select different payload range and identify some visual patterns.

- Callback function to render the success-payload-scatter-chart scatter plot

It is used to plot a scatter plot with the x axis to be the payload and the y axis to be the launch outcome (i.e., class column). As such, we can visually observe how payload may be correlated with mission outcomes for selected site(s). In addition, color-labels were created for the Booster version on each scatter point so that we may observe mission outcomes with different boosters.

Predictive analysis (Classification) (GitHub URL)

Data Preparation

- Load dataset, using NumPy array and Pandas
- Standardize and transform the data

Building Model

- Split the data into training and testing data
- Machine Learning Algorithms:
  - Create logistic regression object
  - Create a support vector machine object
  - Create a decision tree classifier object
  - Create a k nearest neighbors object column

Evaluate Model Performance

- Hyperparameter Optimization
- Fit the object to find the best parameters
- Plot Confusion Matrix
- Calculate the accuracy of each model using the method score

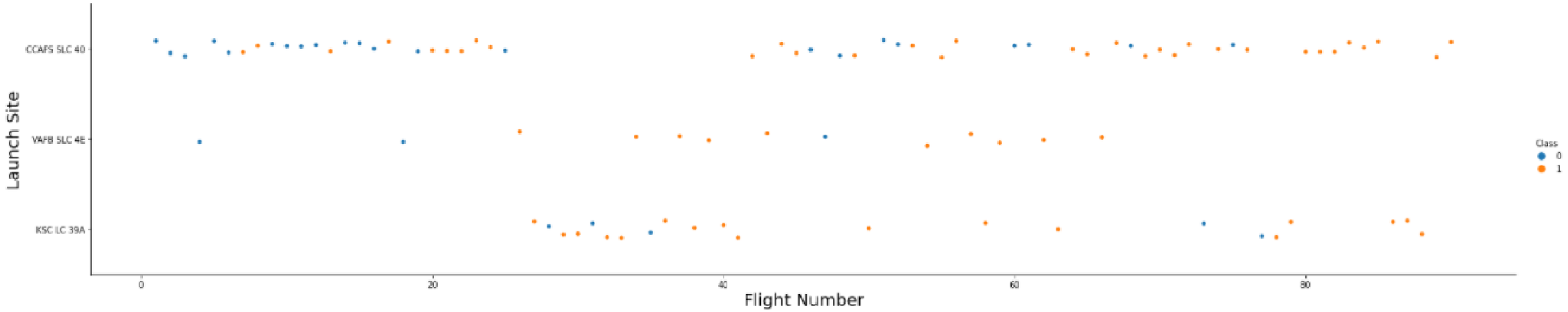
Results

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

EDA with Visualization

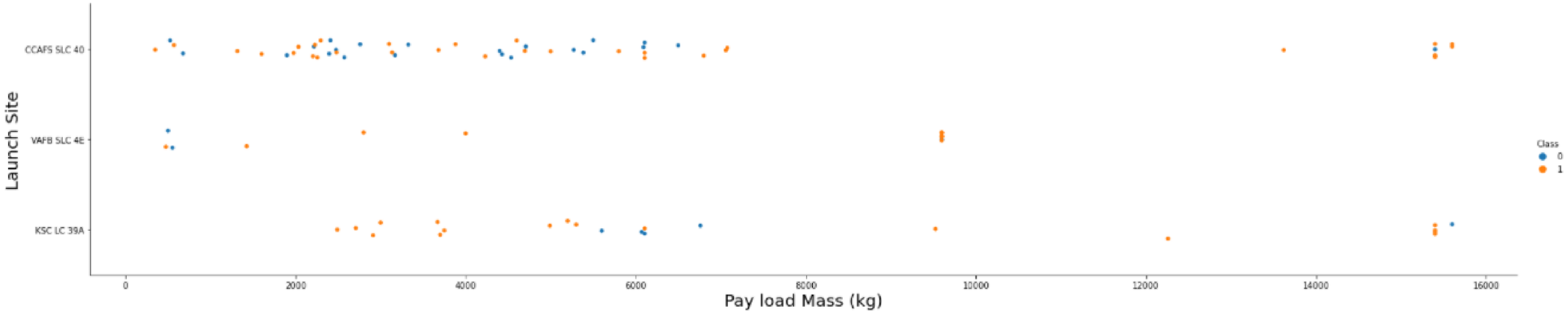
Flight Number vs. Launch Site

We observe that as the flight number (indicating the continuous launch attempts) increases, the first stage is more likely to land successfully at the different launch sites .



Payload vs. Launch Site

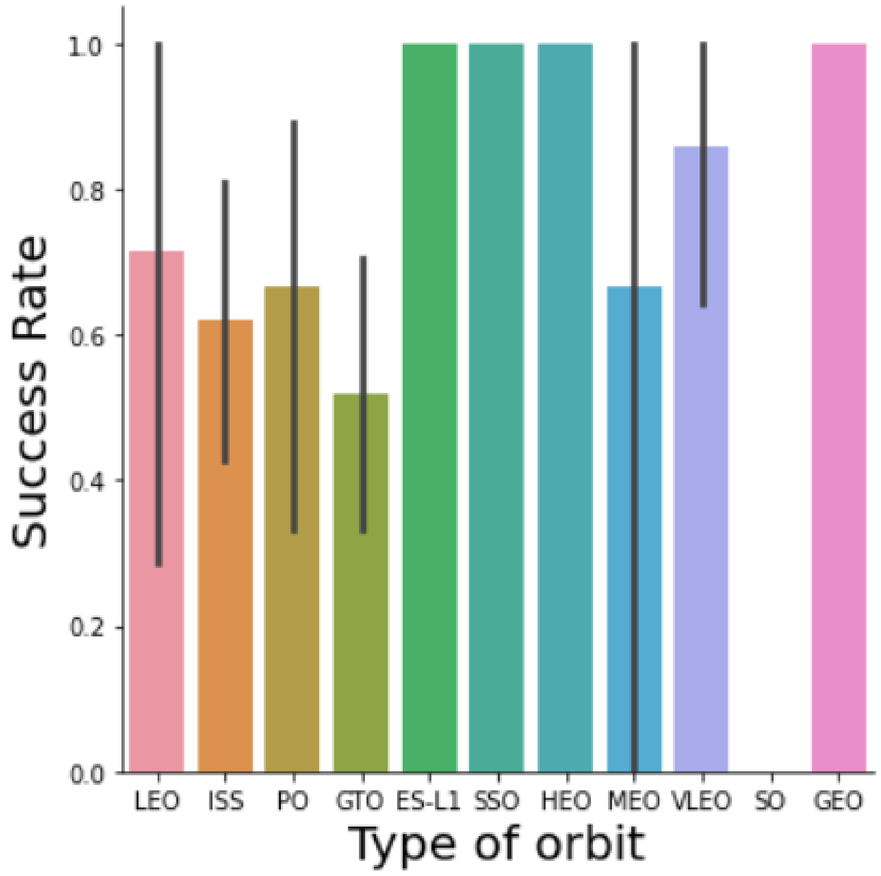
We observe that the more massive the payload, the higher the success rate at the launch site VAFB SLC 4E. Most of the launches with payload mass over 7000 kg were successfull
Launch site KSC LC 39A has a 100% success rate for payload mass under 5500 kg.



Success rate vs. Orbit type

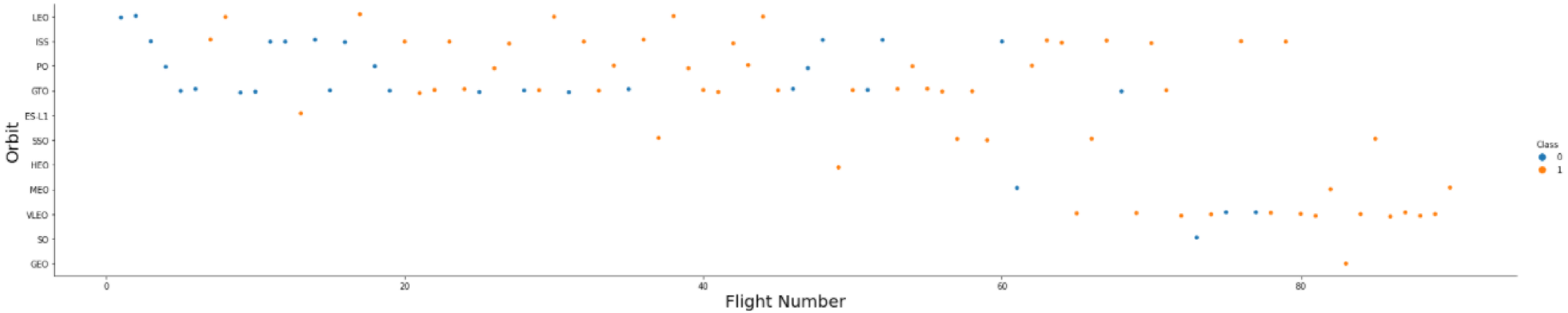
We observe that orbits ES-L1, SSO, HEO and GEO have the highest success rate (100%).

Orbit SO has zero success rate and the rest orbits have success rate between 50% and 85%.



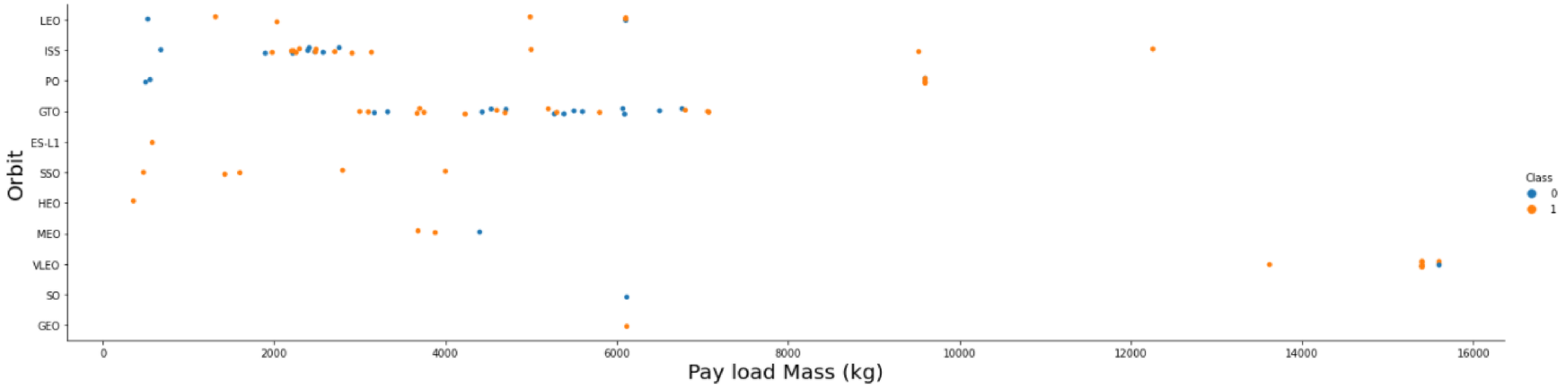
Flight Number vs. Orbit type

We observe 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.



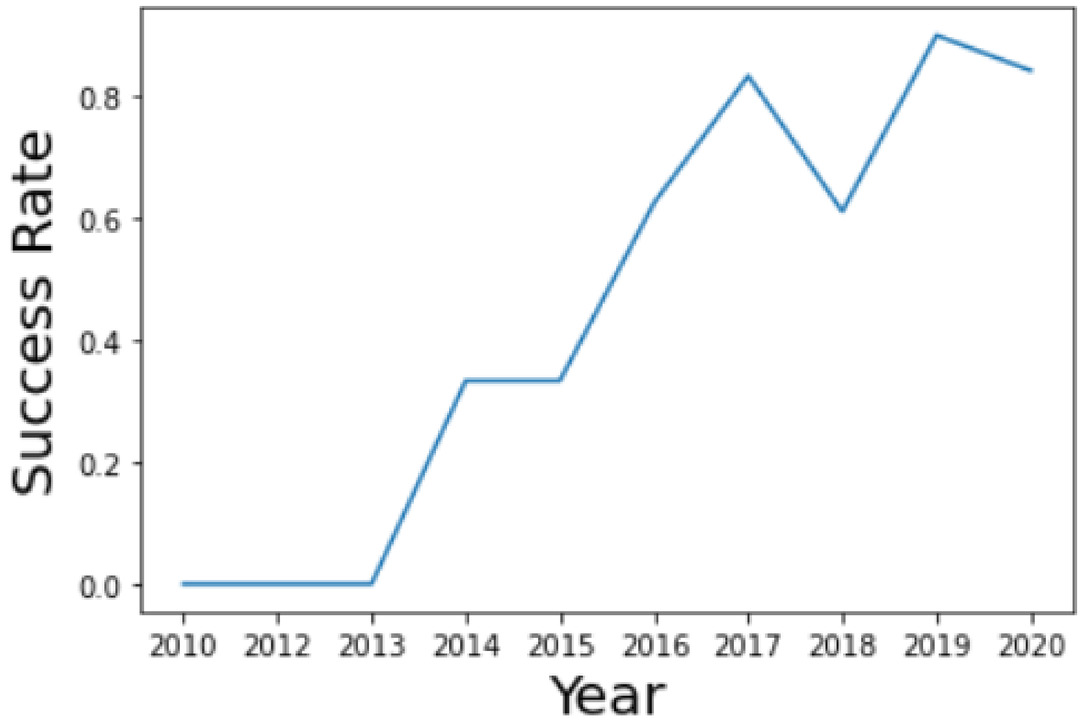
Payload vs. Orbit type

We observe that Heavy payloads have a negative influence on GTO orbits and positive on GTO and Polar LEO (ISS) orbits.



Launch success yearly trend

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



EDA with SQL

All launch site names

Find the names of the unique launch sites

SQL Query

```
select DISTINCT(LAUNCH_SITE)
from SPACEXDATASET
```

Result

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

Query Explanation

The **SELECT DISTINCT** statement in the query is used to return unique values of the Launch_Site column from table SPACEXDATASET

Launch site names begin with `CCA`

Find all launch sites which begin with `CCA`

SQL Query

```
select *
from SPACEXDATASET
where SUBSTRING(LAUNCH_SITE,1,3)='CCA'
LIMIT 5
```

Query Explanation

Using the **SUBSTRING()** function we extract the first three characters from the string in column to equal 'CCA'. The **LIMIT** keyword is used to fetch only 5 records from table SPACEXDATASET.

Result

DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-05-22	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total payload mass

Calculate the total payload carried by boosters from NASA

SQL Query

```
select sum(PAYLOAD_MASS__KG_)
from SPACEXDATASET
where customer='NASA (CRS)'
```

Result

1
45596

Query Explanation

The **SUM()** function returns the total of PAYLOAD_MASS__KG_ column from table SPACEXDATASET.

The **WHERE** clause specifies that the statement should only affect rows that meet specified criteria (boosters from NASA)

Average payload mass by F9 v1.1

Calculate the average payload mass carried by booster version F9 v1.1

SQL Query

```
select avg(PAYLOAD_MASS__KG_)
from SPACEXDATASET
where booster_version = 'F9 v1.1'
```

Result

1
2928

Query Explanation

The **AVG()** function returns the average value of the specified expression (PAYLOAD_MASS__KG_) column from table SPACEXDATASET.

The **WHERE** clause specifies that the statement should only affect rows that meet specified criteria (booster version F9 v1.1)

First successful ground landing date

Find the date when the first successful landing outcome in ground pad

SQL Query

```
select min(DATE)
from SPACEXDATASET
where landing__outcome = 'Success (ground pad)'
```

Result

1
2015-12-22

Query Explanation

The **MIN()** function returns the smallest value of the selected column (DATE) from table SPACEXDATASET.

The **WHERE** clause specifies that the statement should only affect rows that meet specified criteria (successful landing outcome in ground pad)

Successful drone ship landing with payload between 4000 and 6000

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

SQL Query

```
select booster_version
from SPACEXDATASET
where landing__outcome='Success (drone ship)'
and payload_mass__kg_ BETWEEN 4000 and 6000
```

Result

booster_version
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
F9 FT B1031.2

Query Explanation

The **WHERE** clause specifies that the statement should only affect rows that meet specified criteria, that is successful drone ship landing and payload mass greater than 4000 but less than 6000.

Total number of successful and failure mission outcomes

Calculate the total number of successful and failure mission outcomes

SQL Query

```
select mission_outcome, count(*)
from SPACEXDATASET
group by mission_outcome
```

Result

mission_outcome	2
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

Query Explanation

The **GROUP BY** statement specifies that the SQL SELECT statement partitions result rows into groups, based on their values in the column (mission_outcome) and is used to apply the aggregate function **count()** for each group.

Boosters carried maximum payload

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

SQL Query

```
select booster_version
from SPACEXDATASET
where PAYLOAD_MASS__KG_ =(select max(PAYLOAD_MASS__KG_)
                          from SPACEXDATASET)
```

Result

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

Query Explanation

The **subquery** returns the maximum value of column PAYLOAD_MASS__KG_ from table SPACEXDATASET.
Then we select only the boosters that have the maximum payload mass.

2015 launch records

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

SQL Query

```
select booster_version, landing__outcome, launch_site
from SPACEXDATASET
where landing__outcome = 'Failure (drone ship)'
and year(date)='2015'
```

Result

booster_version	landing__outcome	launch_site
F9 v1.1 B1012	Failure (drone ship)	CCAFS LC-40
F9 v1.1 B1015	Failure (drone ship)	CCAFS LC-40

Query Explanation

The **WHERE** clause specifies that the statement should only be applied to rows with year=2015 and failure in landing outcome.

Rank success count between 2010-06-04 and 2017-03-20

Rank the count of successful landing_outcomes between the date 2010-06-04 and 2017-03-20 in descending order.

SQL Query

```
select landing__outcome, count(*)
from SPACEXDATASET
where date between '2010-06-04' and '2017-03-20'
group by landing__outcome
order by count(*) DESC
```

Result

landing__outcome	2
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

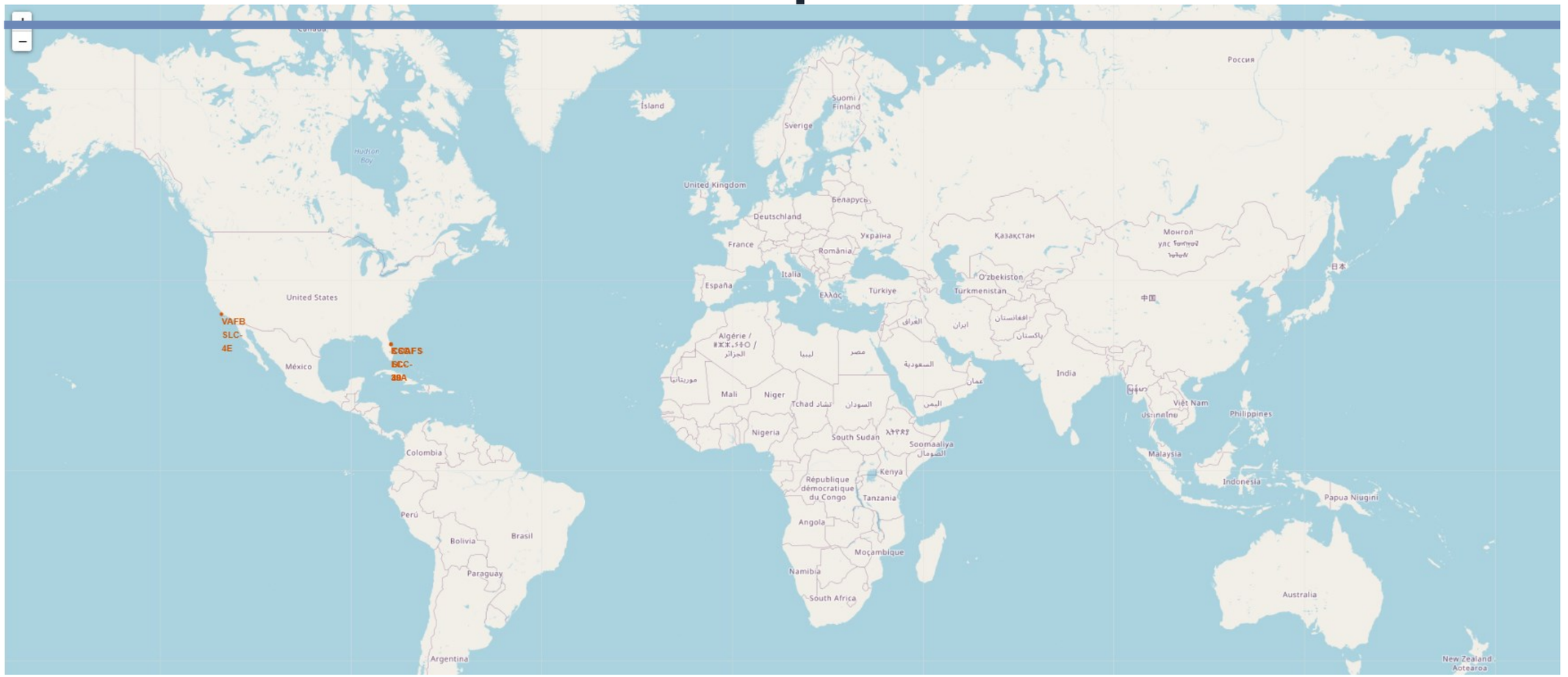
Query Explanation

The **WHERE** clause specifies that the statement is applied to rows that have successful landing outcomes and between the date 2010-06-04 and 2017-03-20.

The keyword **DESC** is used to sort the query result set in a descending order.

Interactive map with Folium

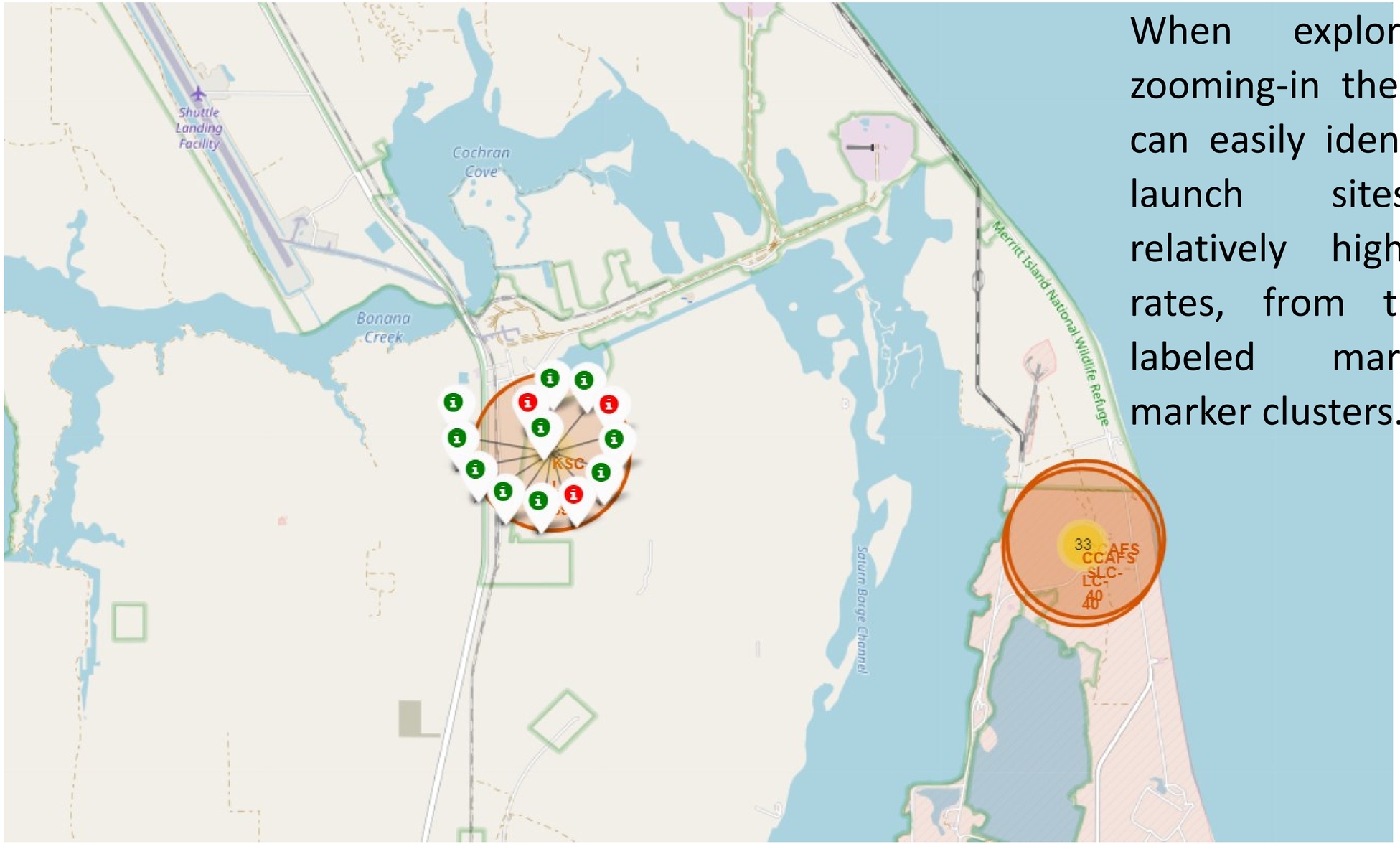
All launch sites location on a map



We observe that launch sites are in proximity to the Equator and most launches take place at the east coast, because a launch near the equator towards the east direction will get an initial boost equal to the velocity of Earth surface.

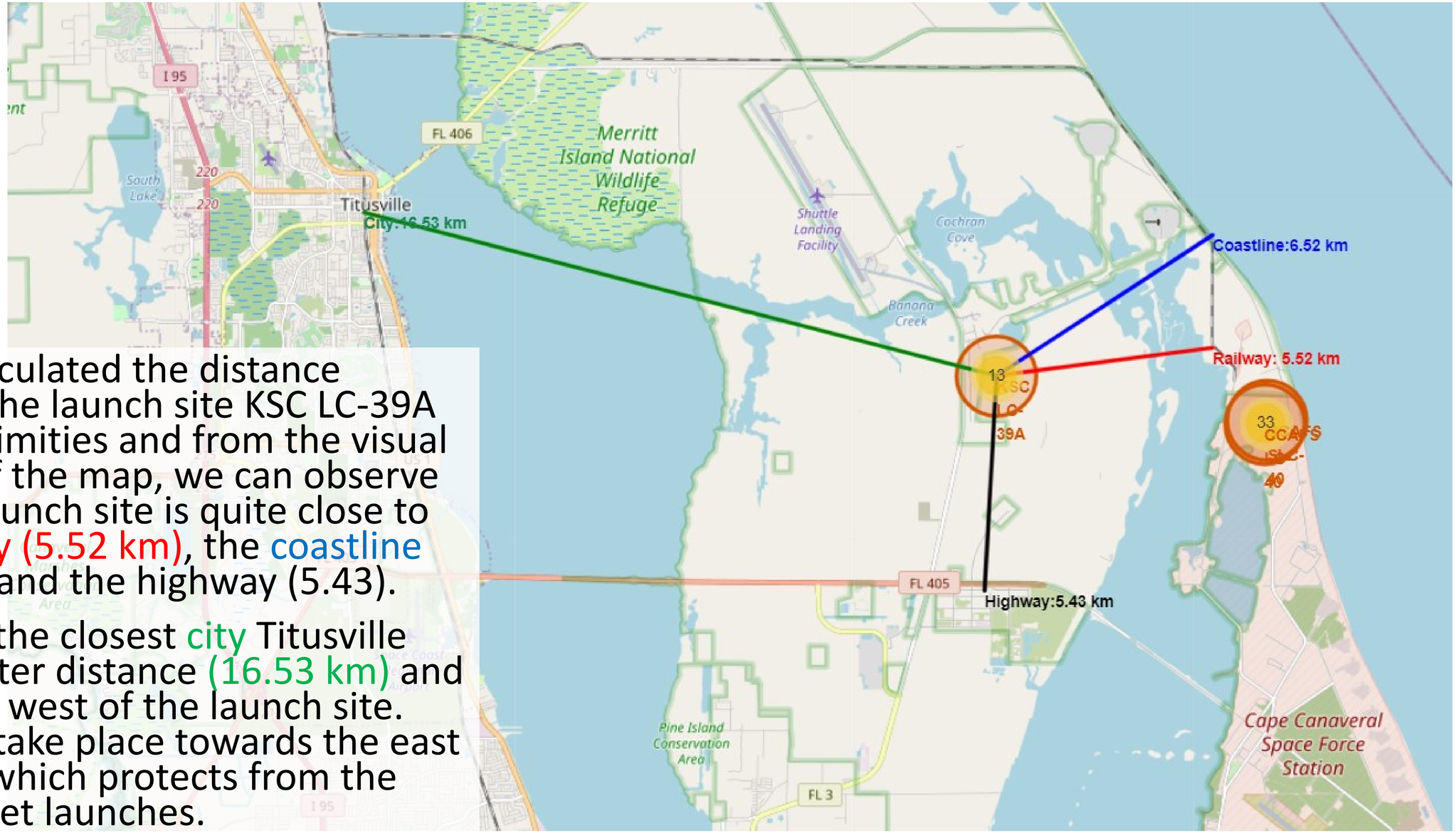
Moreover, all launch sites are in very close proximity to the coast (the air is thicker at sea level and less fuel needs to burn for rocket to take off and reach a desired acceleration).

Success/failed launches for each site on the map



When exploring and zooming-in the map, we can easily identify which launch sites have relatively high success rates, from the color-labeled markers in marker clusters.

Calculate the distances between a launch site to its proximities



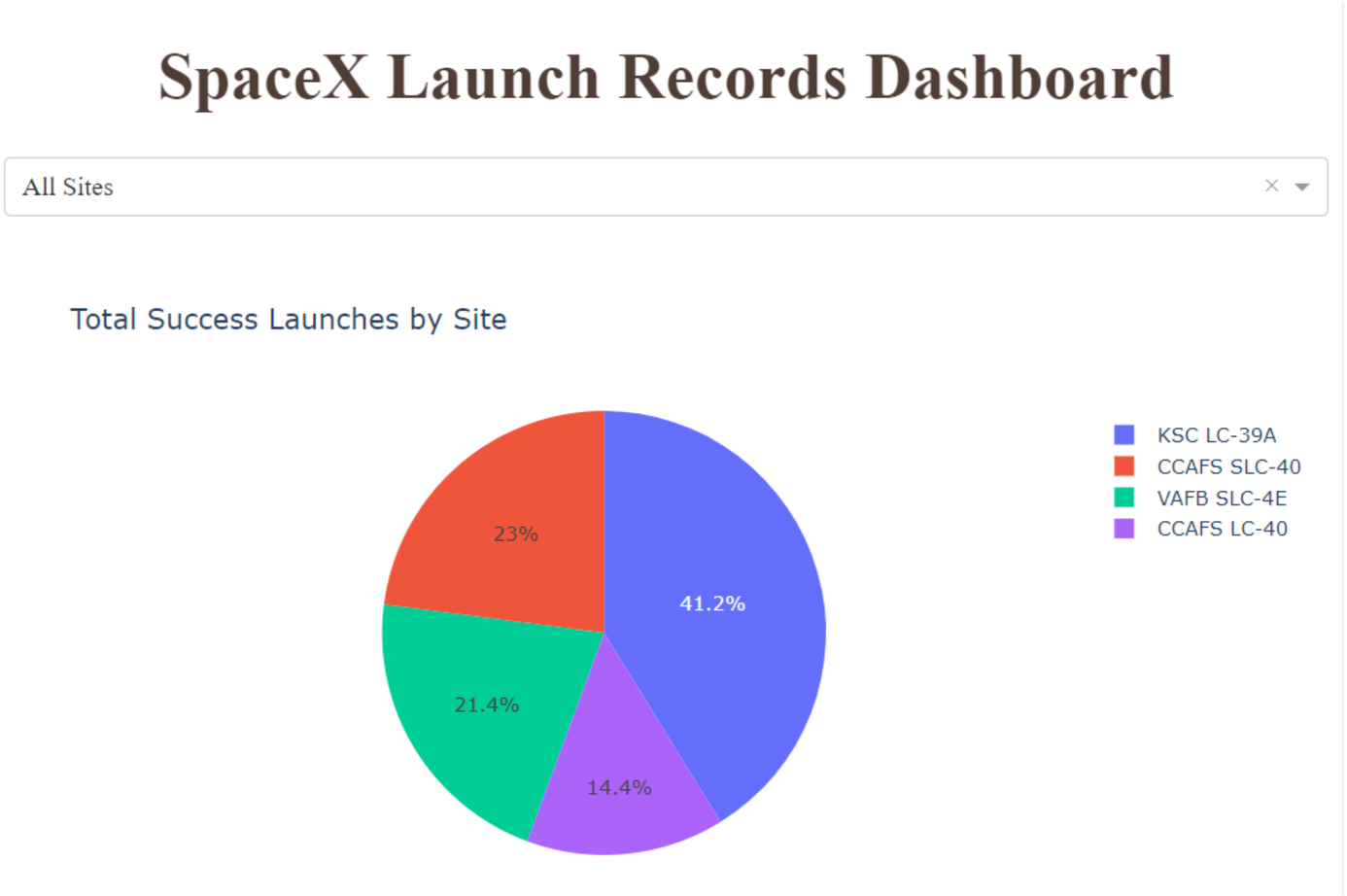
Having calculated the distance between the launch site KSC LC-39A to its proximities and from the visual analysis of the map, we can observe that the launch site is quite close to the railway (5.52 km), the coastline (6.52 km) and the highway (5.43).

However, the closest city Titusville has a greater distance (16.53 km) and is situated west of the launch site. Launches take place towards the east direction which protects from the failed rocket launches.

Build a Dashboard with Plotly Dash

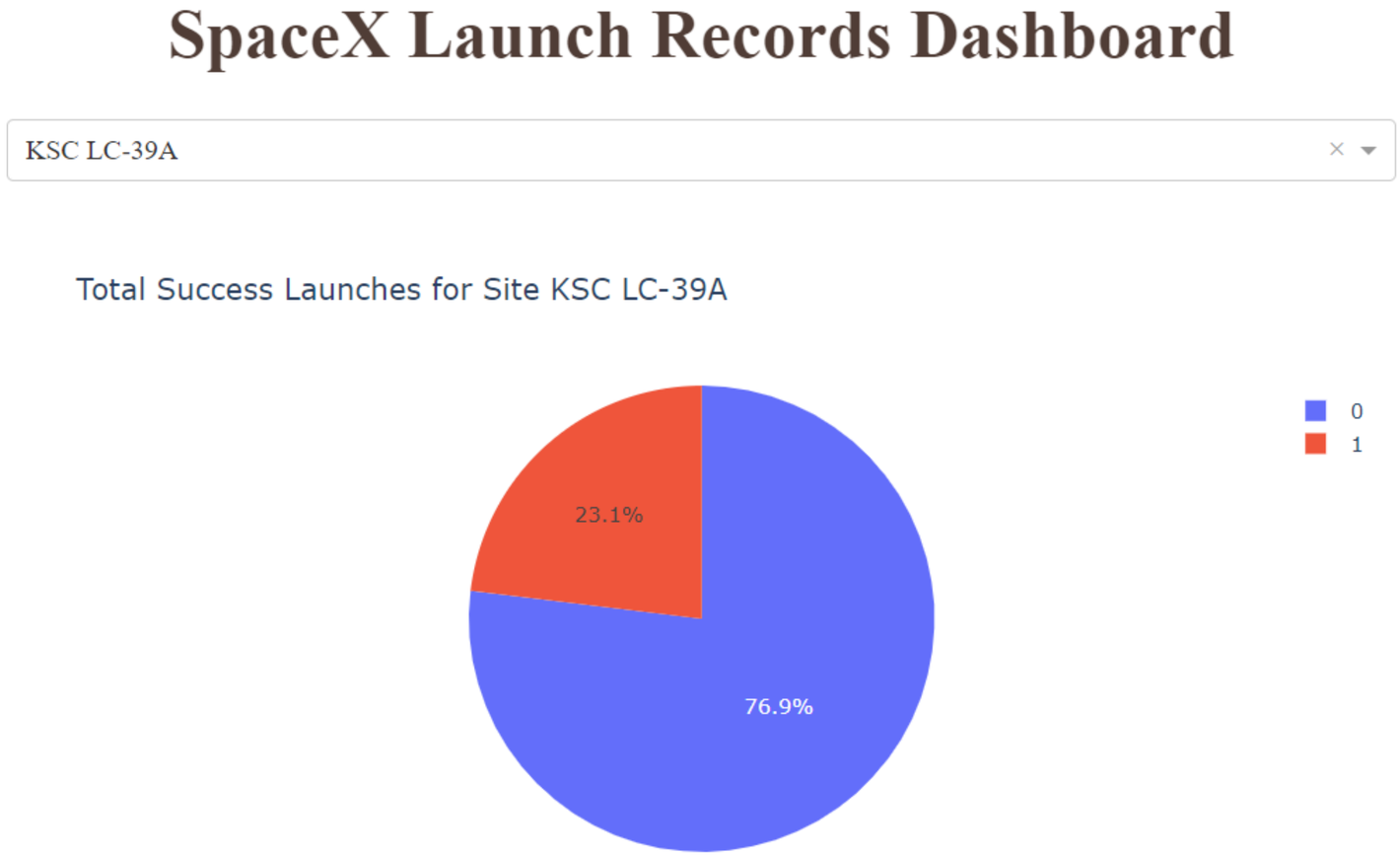
Total success launches by site pie chart

The pie chart visualizes the total launches by site and we observe that launch site KSC LC-39 has the largest successful launches (41.2%).



Launch site with highest launch success ratio pie chart

We observe that launch site KSC LC-39 which has the highest launch success ratio (41.2%) of all sites highest, has 76.9% launch success rate and only 23.1% failure rate.



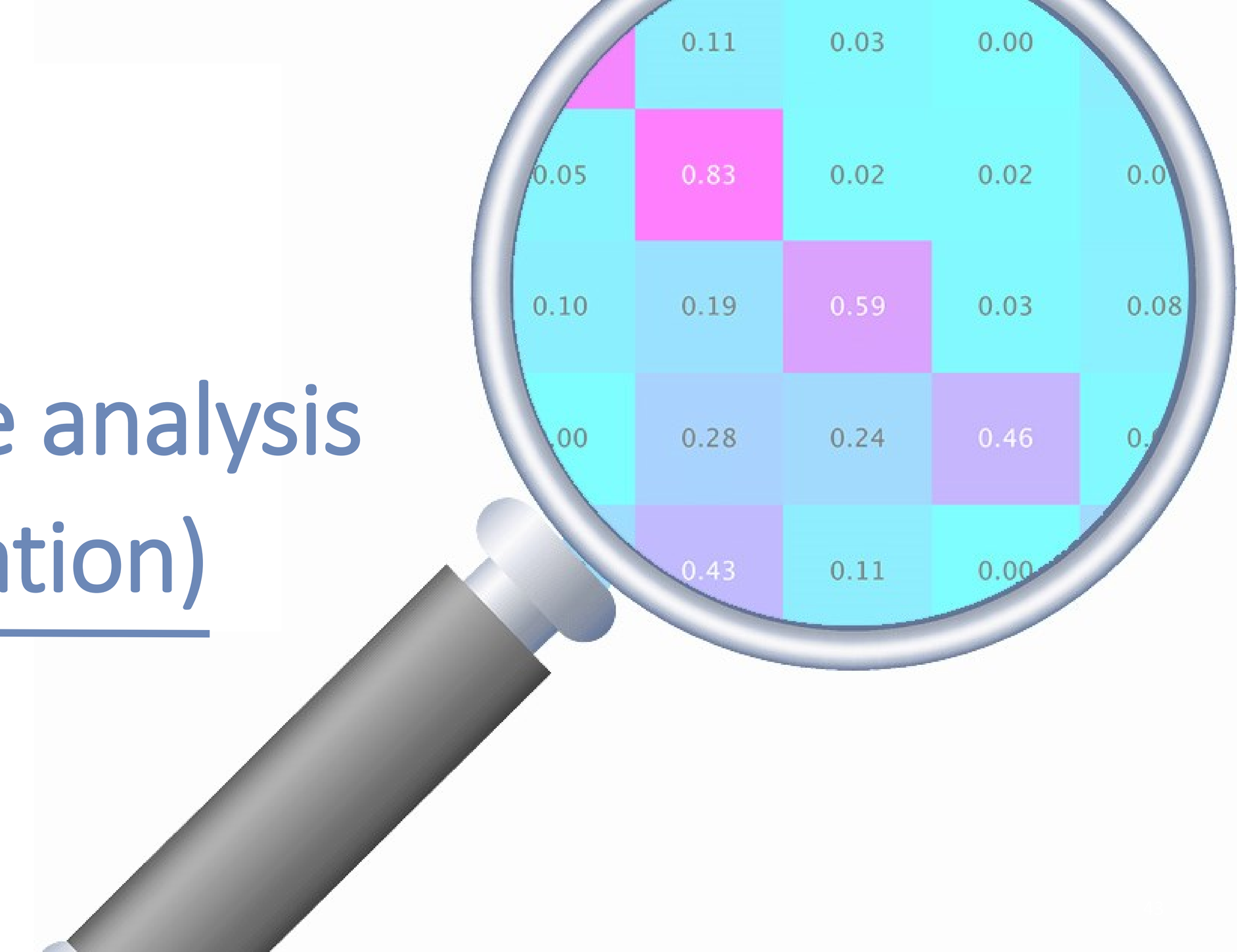
Payload vs. Launch Outcome scatter plot

We observe that for payload range 0-5k the launch success rate is higher than for heavier payload.

For payload between 6000 and 9500 all launches have failed.

Finally, booster version FT has the highest Launch success rate.

Predictive analysis (Classification)



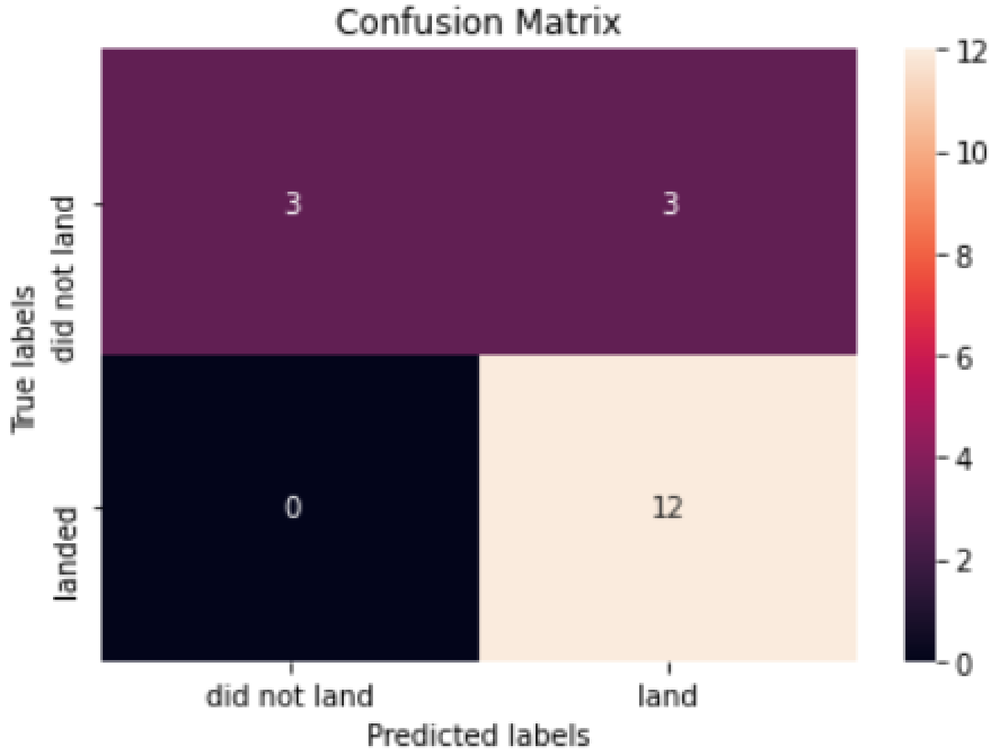
Confusion Matrix

	LogReg	SVM	Tree	KNN
Accuracy of Test Data	0.833333	0.833333	0.833333	0.833333
Accuracy of Validation Data	0.846429	0.848214	0.889286	0.848214

Based on the scores of the Test Set, we can not confirm which method performs best, which may be due to the small test sample size (18 samples).

Based on the scores of the Validation Set it is confirmed that the model with the highest accuracy is the Decision Tree Model.

Examining the confusion matrix, we see that the major problem is false positives.



CONCLUSION

- Orbits ES-L1, SSO, HEO and GEO have the highest success rate (100%).
- Success launch rate is increasing over the years.
- All launch sites are in proximity to the Equator ,very close to the coast and most launches take place at the east coast.
- Launch site KSC LC-39 has the largest successful launches (41.2%) ,with 76.9% launch success rate and only 23.1% failure rate
- Lower payload mass perform better than heavier payloads.
- The Decision Tree Classifier Algorithm has the highest accuracy from the machine learning algorithms that were used .