Winning Space Race with Data Science

Anish Rodrigues
12.July.2022

Outline

- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion

Executive Summary

- In this project, I have studied SpaceX's Falcon 9 past rocket launches in order to understand which factors determined their success or failure, and predict the outcome of future launches.
- To achieve this goal, I have employed many of the usual data analysis and visualization techniques, while several machine learning classification algorithms have been used to build predictive models.

Introduction

- As a data scientist working for the new rocket company SpaceY, an essential part of my job is studying SpaceX, our direct competitor.
- One of the major competitive advantage of SpaceX is due to the fact that they can reuse the first stage of a launch. Therefore, I am interested in data from SpaceX's past launches, in order to understand when their launches succeed and why.

Section 1

Methodology

Methodology

Executive Summary

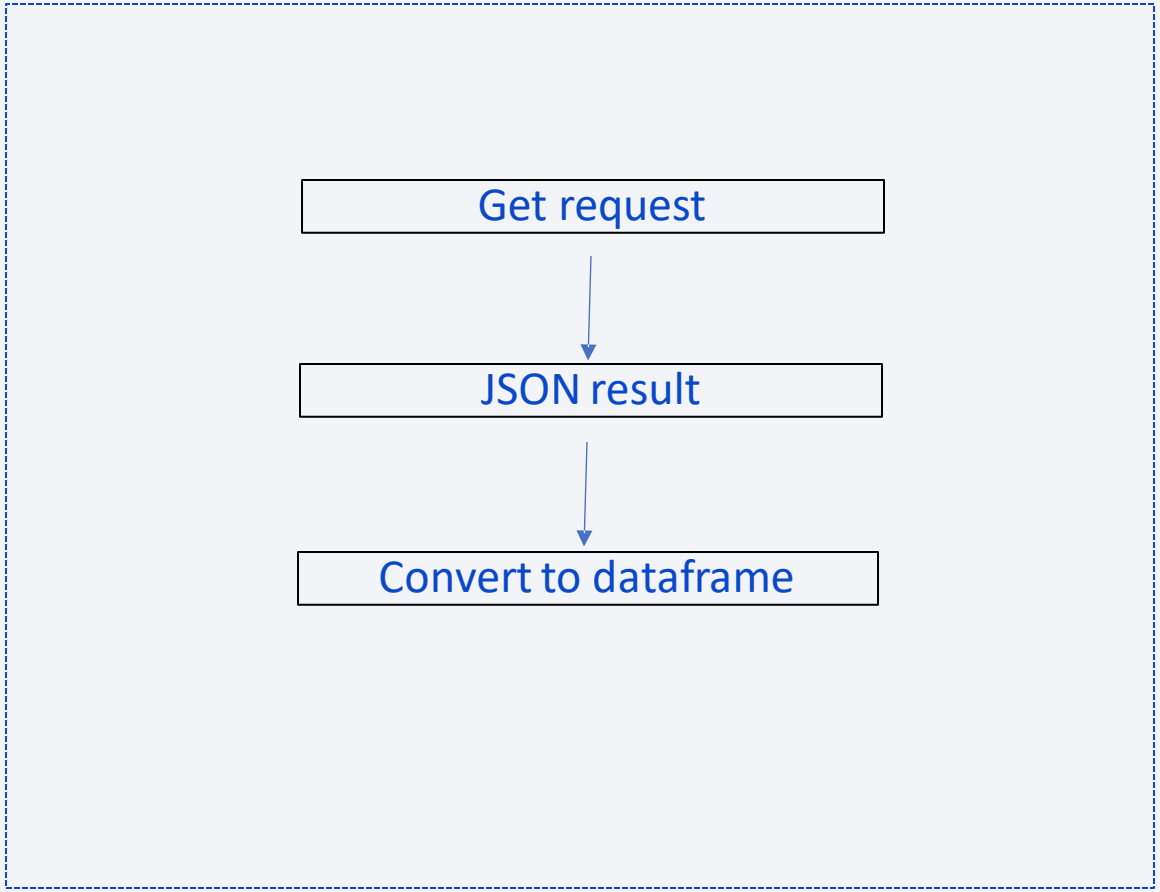
- Data collection methodology:
  - Data were collected partly from the SpaceX REST API, partly from Wikipedia.
- Perform data wrangling
  - The first part of analysis was devoted to computing some immediate statistics, like the number of launches on each site, the number and occurrence of each orbit, and creating the target variable that indicates success or failure of each launch.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - I have tested several algorithms, namely KNN, logistic regression, decision tree and support vector machine, finding the best hyperparameters via grid search.

Data Collection

- Data were collected from two main sources, the SpaceX REST API and the Wikipedia dedicated pages.
- The SpaceX REST API is an API that collects publicly available data about SpaceX's past launches.

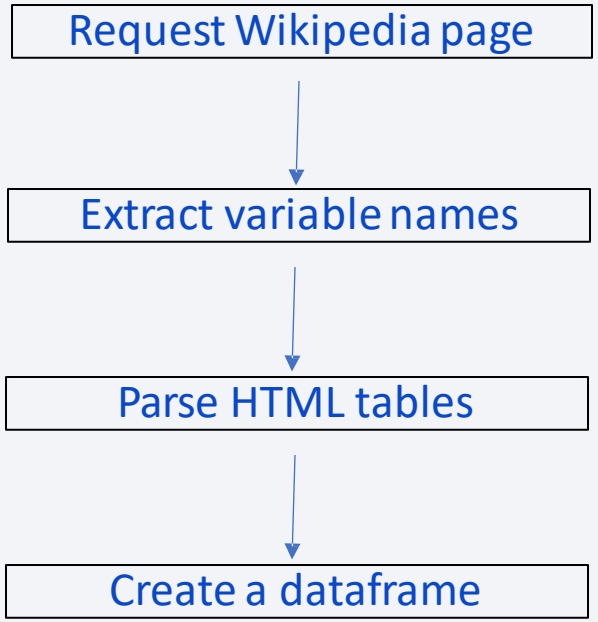
Data Collection - SpaceX API

- First I requested rocket launch data from SpaceX API: they come in the JSON form. Then I used the method .json_normalize() to convert them into a dataframe.
- The results are in the Jupyter Notebook Data Collection.



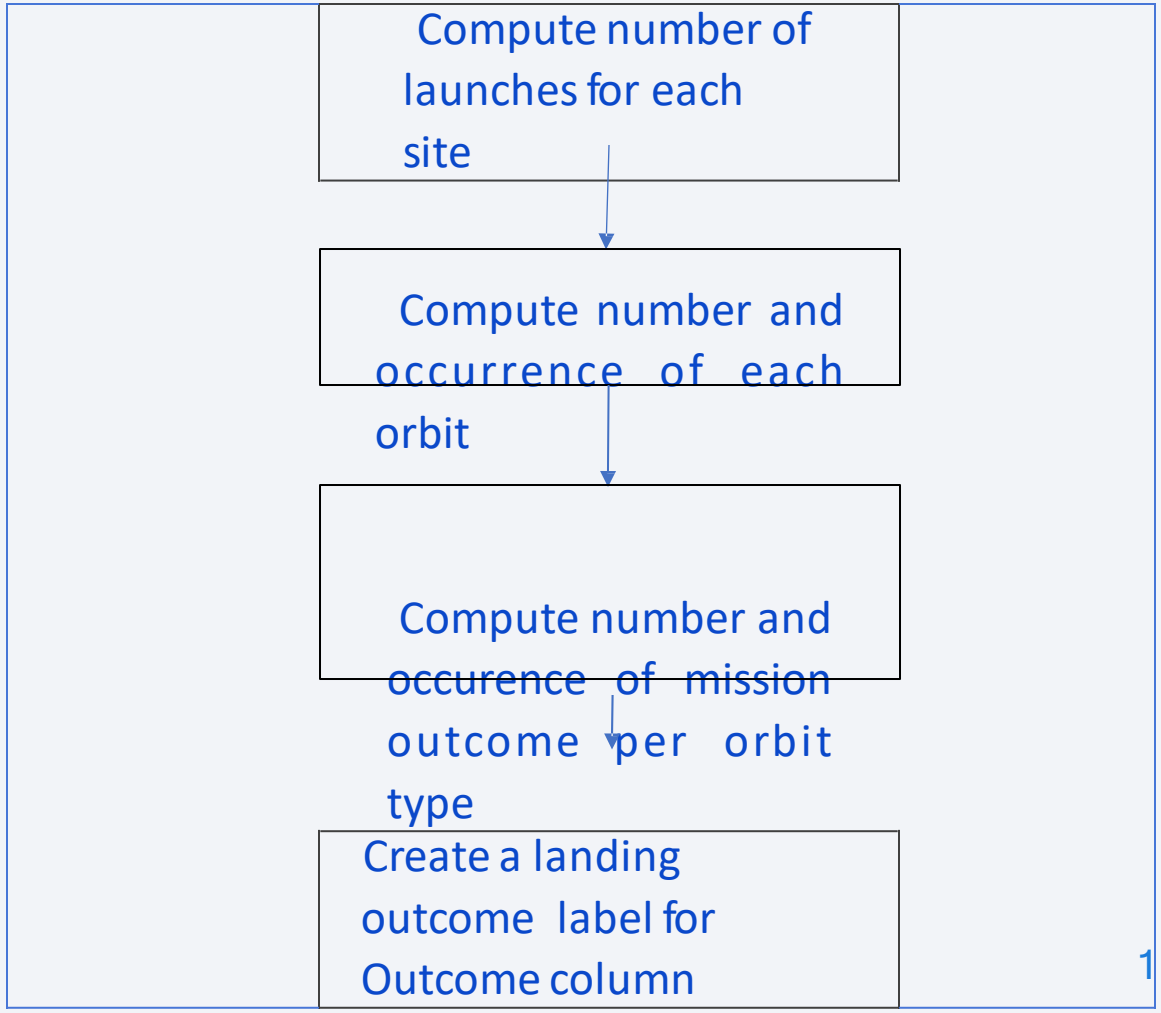
Data Collection - Scraping

- Webscraping begins with a HTTP request. This returns a BeautifulSoup object in Python: in order to obtain a dataframe several manipulations are needed, as you can see in the flowchart on the right.
- The results are in the Jupyter Notebook Data Collection with Webscraping.



Data Wrangling

- After collecting data, I performed some simple exploratory data analysis (EDA) in order to have a first understanding. For details, see the flowchart on the right.
- The results are in the Jupyter Notebook Data Wrangling.



EDA with Data Visualization

An important part of EDA is plotting some charts, in order to get an intuitive idea of the relationships between variables involved. In particular, I plotted:

- Flight number vs Launch site;
- Payload vs Launch site;
- Success rate vs Orbit type;
- Flight number vs Orbit type;
- Payload vs Orbit type;
- Launch success yearly trend.

The results are in the Jupyter Notebook EDA with Visualization.

EDA with SQL

The last part of EDA consisted of performing some SQL queries, asking for:

- Names of unique launch sites;
- 5 records where launch sites begins with the string 'CCA';
- total payload mass carried by boosters launched by NASA;
- average payload mass carried by booster version F9 v1.1;
- date of the first successful landing outcome in ground pad;
- names of the boosters which have success in drone ship and have payload mass between 4000 and 6000;
- total number of successful and failure mission outcomes;
- names of the booster versions which have carried the maximum payload mass;
- failed landing outcomes in drone ship, their booster versions, and launch site names in year 2015;
- count of landing outcomes between the date 2010-06-04 and 2017-03-20, in descending order.

The results are in the Jupyter Notebook EDA with SQL.

Build an Interactive Map with Folium

I inserted the following objects on a Folium map:

- A circle and a marker to highlight launch sites;
- A marker cluster for each launch site that counts success/failed launches;
- A mouse position object to get coordinates for a mouse over a point on the map;
- Some lines from the CCAFS LC-40 launch site and the closest coastline, railway, highway and city, along with markers carrying the distance values.

The results are in the Jupyter Notebook Interactive Visual Analytics with Folium. Important. Folium maps do not show up natively on GitHub, please use https://nbviewer.org/ to see the correct result.

Build a Dashboard with Plotly Dash

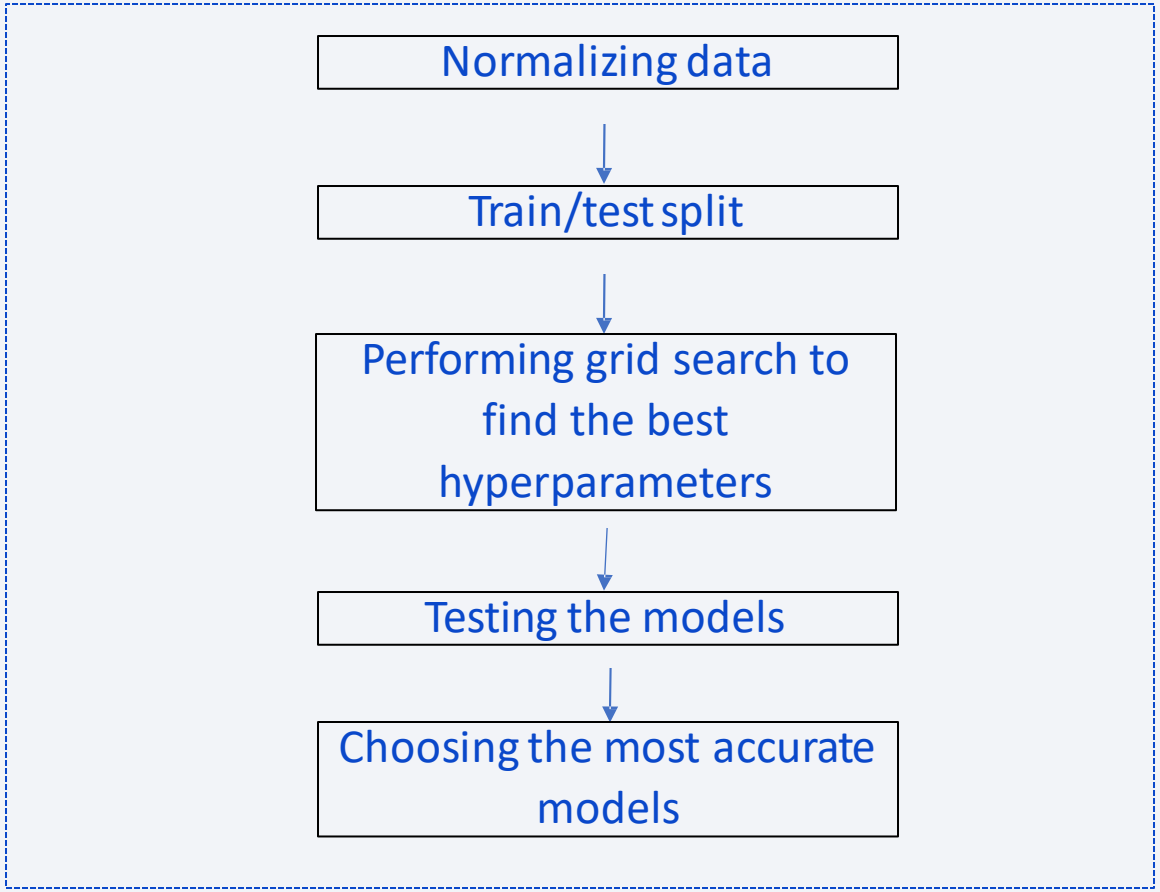
The last step before training a prediction model was building a dashboard with Plotly Dash. I included:

- A dropdown menu where one can select the launch site to analyze, together with an 'all' option;
- A pie chart that represents percentage of total success launches for each site, if the option 'all' is select, or percentage of success/failed launches for the site if a specific site is selected;
- A range slider to select payload mass range for the last chart;
- A scatter plot payload mass vs. success/failed in the selected range, with the colors of the points representing booster version.

The app is available on GitHub here.

Predictive Analysis (Classification)

- Finally, I built predictive models using different algorithms: logistic regression, support vector machine, decision tree and KNN.
- The flowcharts shows the modelling process in more details. In the end, all algorithms had the same performance expect KNN that performed worse.
- The results are in the Jupyter Notebook Machine Learning Prediction.

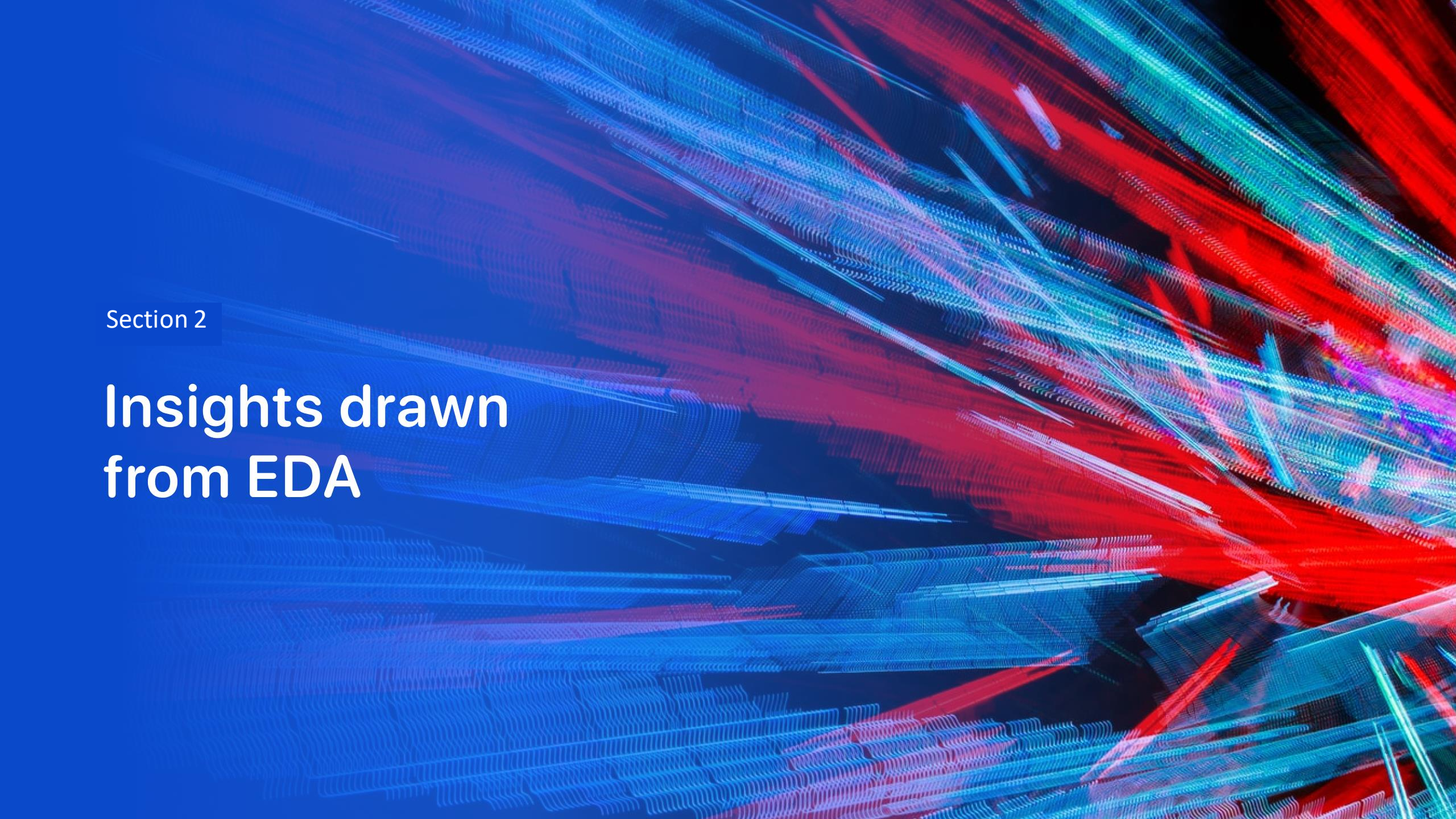


Results

We can draw some conclusions from EDA with data visualization:

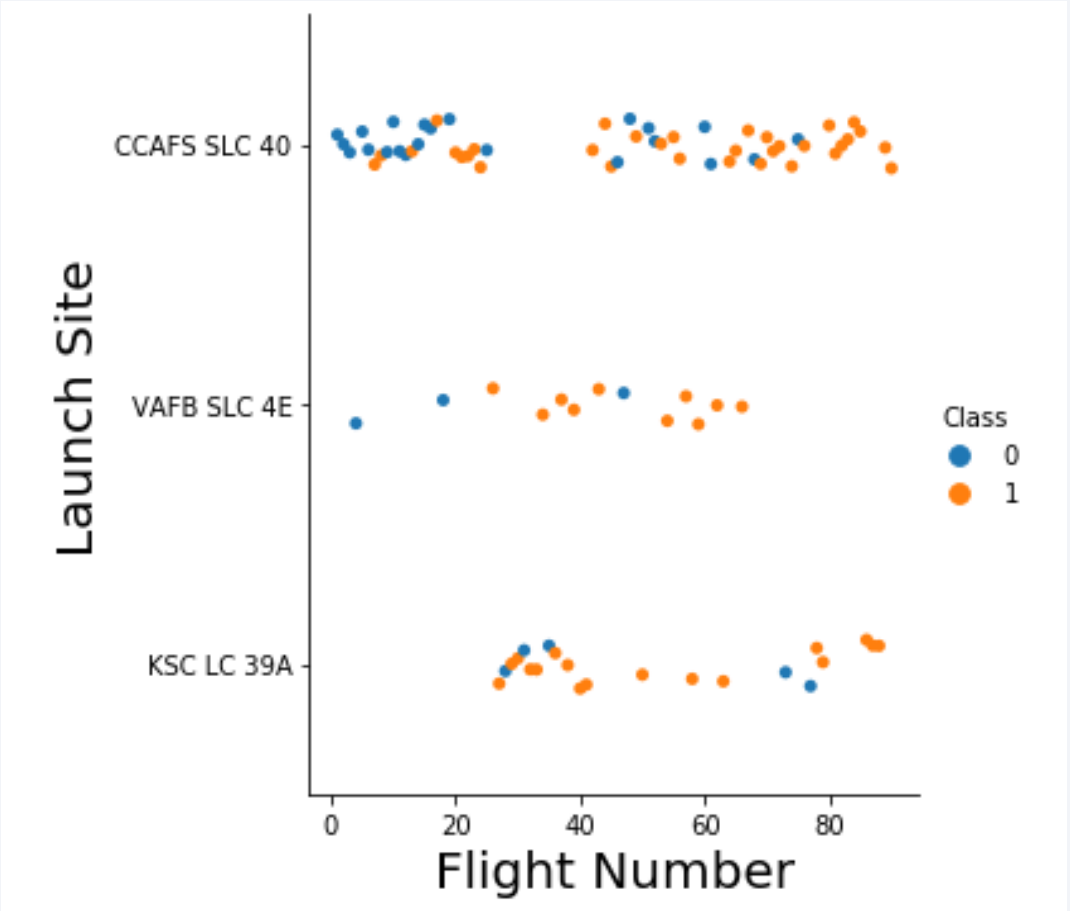
- For all launch sites, the success rate improves with time and payload mass.
- The orbits ES-L1, GEO, HEO and SSO show a 100% success rate.
- The latest launches are concentrated in the VLEO orbit.
- With heavy payloads, orbits Polar, LEO and ISS show high success rate.

- For what concerns predictive analysis, I have discovered that logistic regression, support vector machine and decision tree all show an accuracy score of 0.83 on my test set, when employed with the optimal hyperparameters found with grid search. In order to better assess accuracy, they may be tested further using cross validation.


Section 2
Insights drawn from EDA

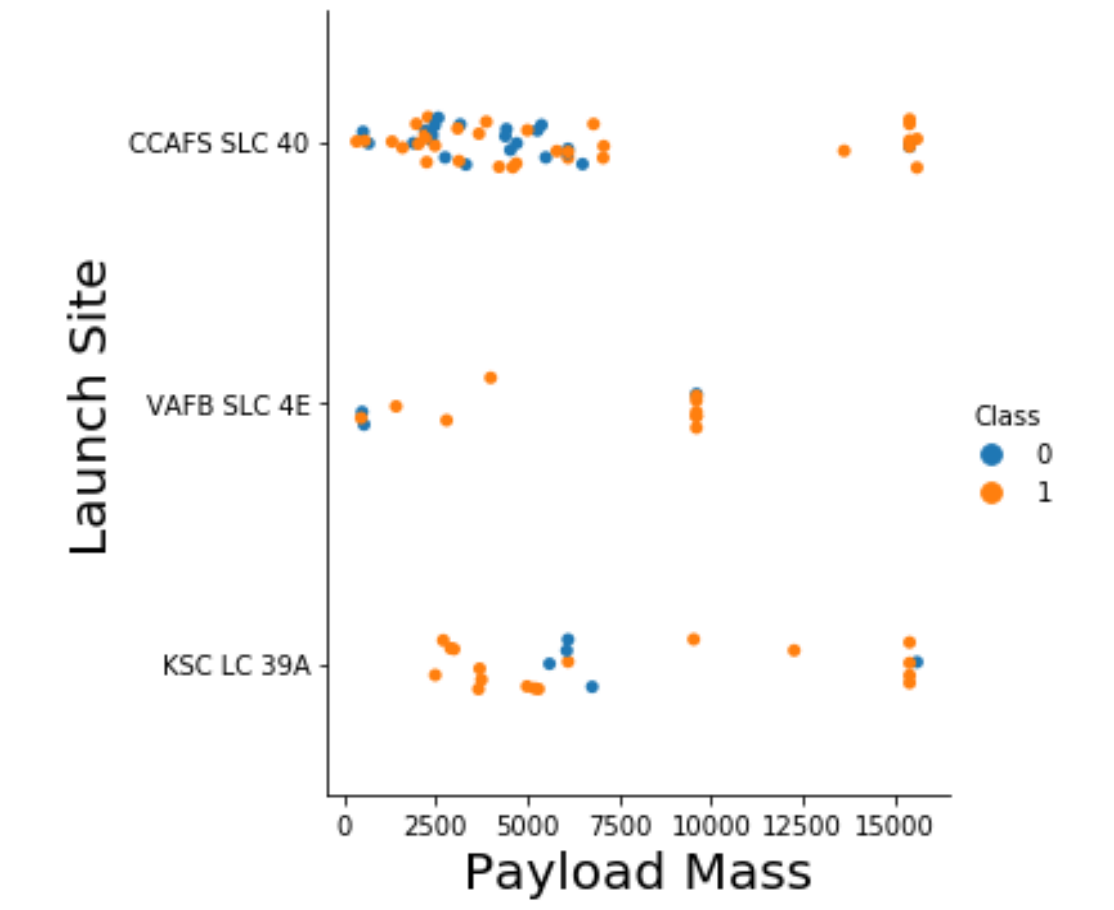
Flight Number vs. Launch Site

This scatter plot shows the flight number against launch site, while the point colors represent success (1) or failure (0). We can clearly see that the success rate increases with the flight number, so with time. Moreover, the launch site CCAFS SLC 40 shows the greatest number of launches, while no launch started from the VAFB SLC 4E site after the 65th approximately.



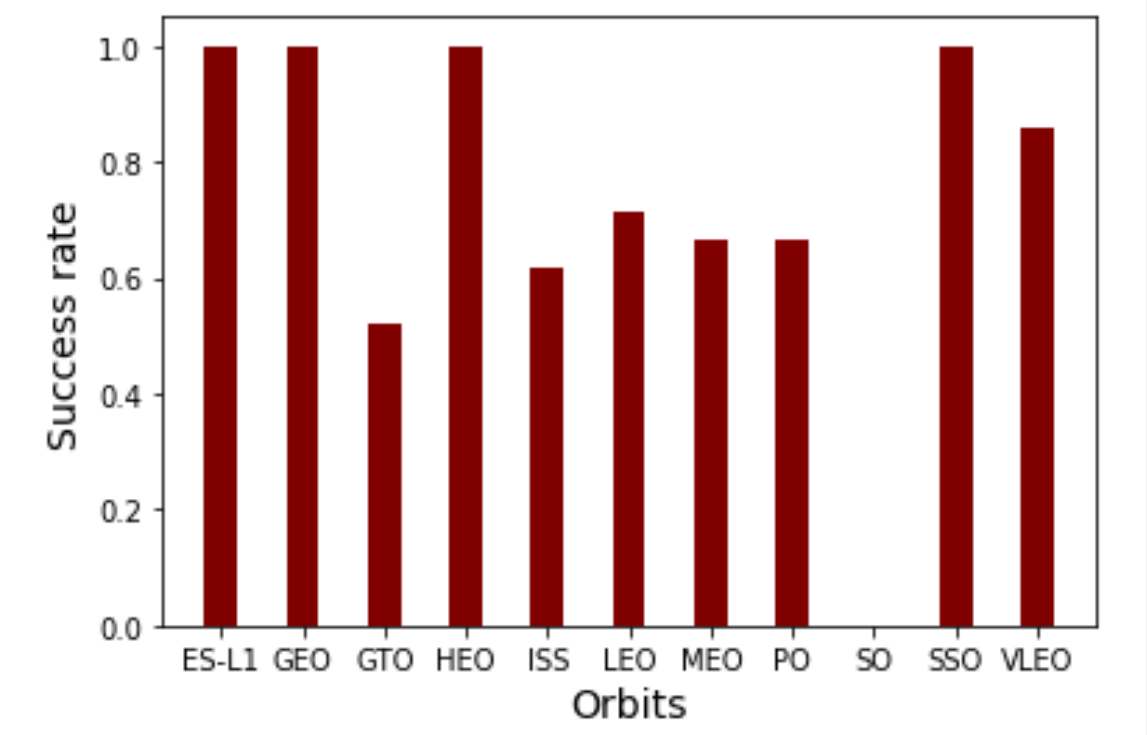
Payload vs. Launch Site

This scatter plot shows the payload mass against launch site, while the point colors represent again success/ failure. We can see that no high payload mass launches departed from VAFB SLC 4E site, while most low payload mass departed from CCAFS SLC40 site. Also, heavy payload mass launches are more likely to succeed.



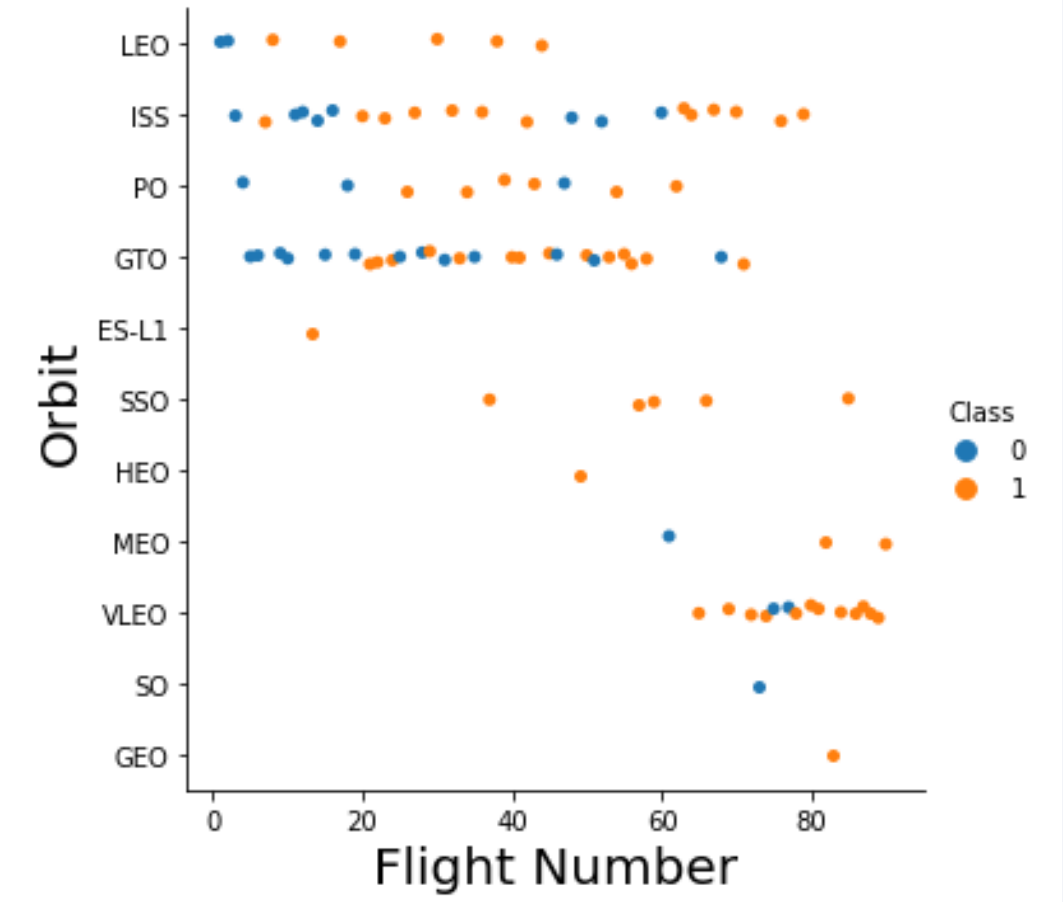
Success Rate vs. Orbit Type

This bar chart shows the success rate for each orbit. We can see that there are four orbits that exhibit 100% success rate, namely ES-L1, GEO, HEO and SSO. On the contrary, the SO has 0 success rate, while the other ones have success rate between 50% and 100%.



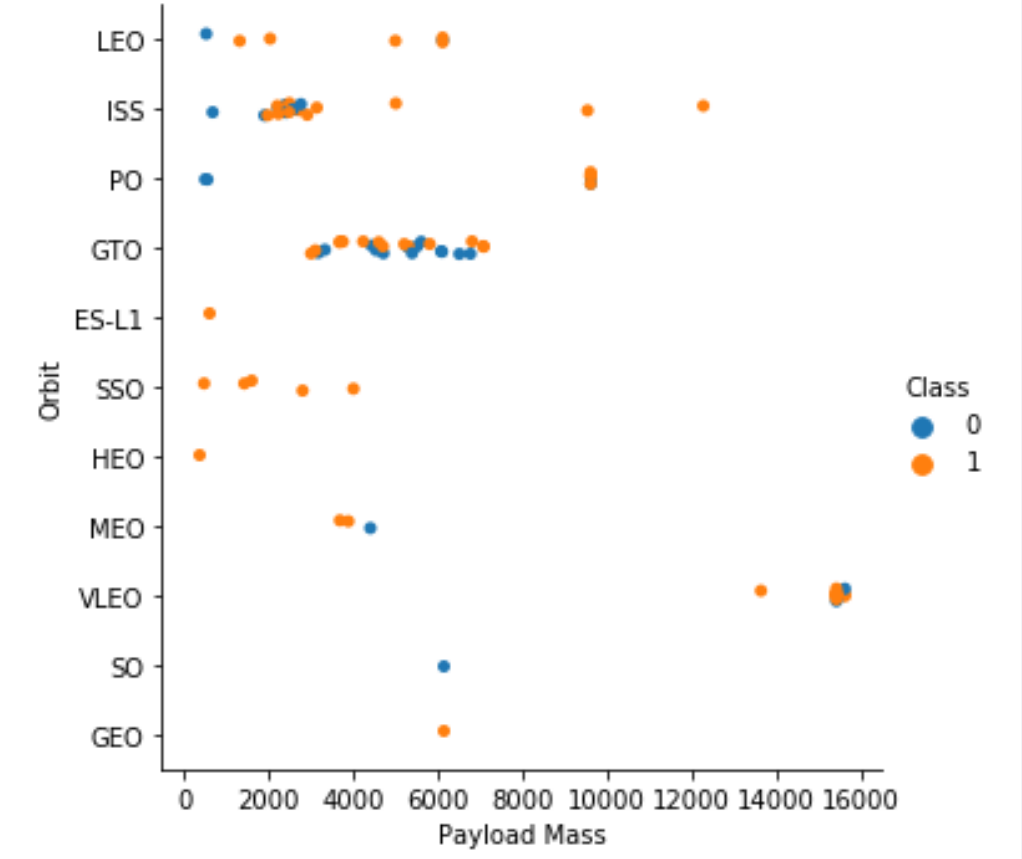
Flight Number vs. Orbit Type

This scatter plot shows the flight number against orbit, with colors representing success/failure as usual. Here we can see that the first flights are concentrated in the orbits LEO, ISS, PO and GTO and the latest in the orbit VLEO. While usually success is positively correlated with the flight number, this is not the case for the orbit GTO.



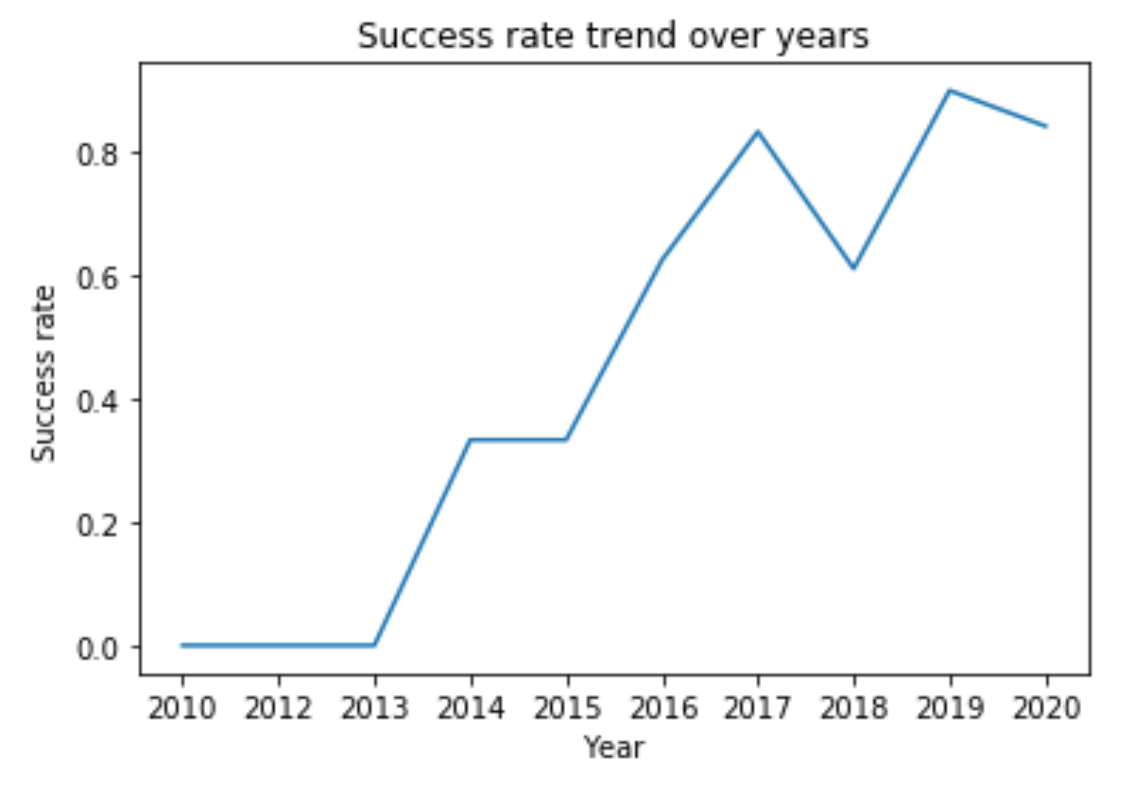
Payload vs. Orbit Type

This scatter plot shows the payload mass against orbit type, while colors represent success/failure. Here we can notice that mid-high payload mass (9k-13k) launches are concentrated in the orbits ISS and PO and high payload mass (>13k) launches are concentrated in the VLEO orbit. They both show very high success rate, while the low payload mass launches show lower success rate.



Launch Success Yearly Trend

This line chart shows success rate trend over years. The trend is clearly increasing, starting from 0% in the years 2010-2013 and exceeding 80% in 2019-2020. The only exception in the trend is year 2018, where success rate was only slightly above 60%.



All Launch Site Names

Here I show a SQL query that displays the name of the unique launch sites. Unique results are retrieved by employing the keyword distinct.

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

```
%sql select distinct launch_site from spacexdataset
```

```
 * ibm_db_sa://xrw98732:***@9938aec0-8105-433e-8bf9-0fbb7e483086.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:3245
9/bludb
Done.
```

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

Launch Site Names Begin with 'CCA'

Here I show a SQL query that displays 5 records where launch sites begin with 'CCA'. I ask for exactly 5 records via the command limit 5, while the string 'CCA%' indicates a generic word beginning with 'CCA'.

```
%%sql

select * from spacexdataset
where launch_site like 'CCA%'
limit 5
```

```
 * ibm_db_sa://xrw98732:***@9938aec0-8105-433e-8bf9-0fbb7e483086.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:3245
9/bludb
Done.
```

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

This is a SQL query that computes the total payload mass carrier by boosters launched by NASA: the sum function adds all records, while the where clause filters the database.

```
%%sql

select sum(payload_mass__kg_) from spacexdataset
where customer = 'NASA (CRS)'
```

```
 * ibm_db_sa://xrw98732:***@9938aec0-8105-433e-8bf9-0fbb7e483086.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32459/bludb
Done.
```

1
45596

Average Payload Mass by F9 v1.1

This SQL query computes the average payload mass carried by booster version F9 v1.1: the avg function computes the average and the where clause filters the database.

```
%%sql

select avg(payload_mass__kg_) from spacexdataset
where booster_version = 'F9 v1.1'
```

```
 * ibm_db_sa://xrw98732:***@9938aec0-8105-433e-8bf9-0fbb7e483086.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:3245
9/bludb
Done.
```

1
2928

First Successful Ground Landing Date

This SQL query computes the average payload mass carried by booster version F9 v1.1: the where clause filters the database, while the combination of order by and limit 1 gives the first record satisfying the condition.

```
%%sql

select DATE from spacexdataset
where landing__outcome = 'Success (ground pad)'
order by DATE
limit 1
```

```
 * ibm_db_sa://xrw98732:***@9938aec0-8105-433e-8bf9-0fbb7e483086.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:3245
9/bludb
Done.
```

DATE
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

This SQL query retrieves the names of the boosters which have success in drone ship and have payload mass between 4000 and 6000. The clauses are similar to previous queries, except for the between keyword that introduces a numeric range.

```
%%sql

select distinct booster_version from spacexdataset
where landing__outcome = 'Success (drone ship)' and payload_mass__kg_ between 4000 and 6000
```

```
 * ibm_db_sa://xrw98732:***@9938aec0-8105-433e-8bf9-0fbb7e483086.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:3245
9/bludb
Done.
```

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

Total Number of Successful and Failure Mission Outcomes

This SQL query counts the number of different possible success and failure mission outcomes. The distinct possible outcomes are retrieved by a subquery, then the main query counts the possible outcomes.

```
%%sql

select count(*) from
(select distinct landing__outcome from spacexdataset
where landing__outcome like 'Failure%' or landing__outcome like 'Success%')
```

```
 * ibm_db_sa://xrw98732:***@9938aec0-8105-433e-8bf9-0fbb7e483086.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32459/bludb
Done.
```

1
6

Boosters Carried Maximum Payload

This SQL query retrieves the names of the boosters which carried the maximum payload mass. This maximum is computed by a subquery via a max function.

```
%%sql

select distinct booster_version from spacexdataset
where payload_mass__kg_ = (select max(payload_mass__kg_) from spacexdataset)
```

```
 * ibm_db_sa://xrw98732:***@9938aec0-8105-433e-8bf9-0fbb7e483086.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32459/bludb
Done.
```

2015 Launch Records

This SQL query retrieves date, time, booster version and launch site of the failed landing outcomes in drone ship in year 2015. The query employs constructs already found in previous queries.

```
%%sql

select DATE, time__utc_, booster_version, launch_site from spacexdataset
where landing__outcome = 'Failure (drone ship)' and year(DATE) = 2015
```

```
 * ibm_db_sa://xrw98732:***@9938aec0-8105-433e-8bf9-0fbb7e483086.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:3245
9/bludb
Done.
```

DATE	time__utc_	booster_version	launch_site
2015-01-10	09:47:00	F9 v1.1 B1012	CCAFS LC-40
2015-04-14	20:10:00	F9 v1.1 B1015	CCAFS LC-40

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

This SQL ranks the count of landing outcomes between 2010-06-04 and 2017-03-20 in descending order. The count is based on a group by clause, while descending ranking is obtained via an order by ... desc clause.

```
%%sql

select landing__outcome, count(*) nr_landing_outcome from spacexdataset
where DATE between '2010-06-04' and '2017-03-20'
group by landing__outcome
order by nr_landing_outcome desc
```

```
 * ibm_db_sa://xrw98732:***@9938aec0-8105-433e-8bf9-0fbb7e483086.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:3245
9/bludb
Done.
```

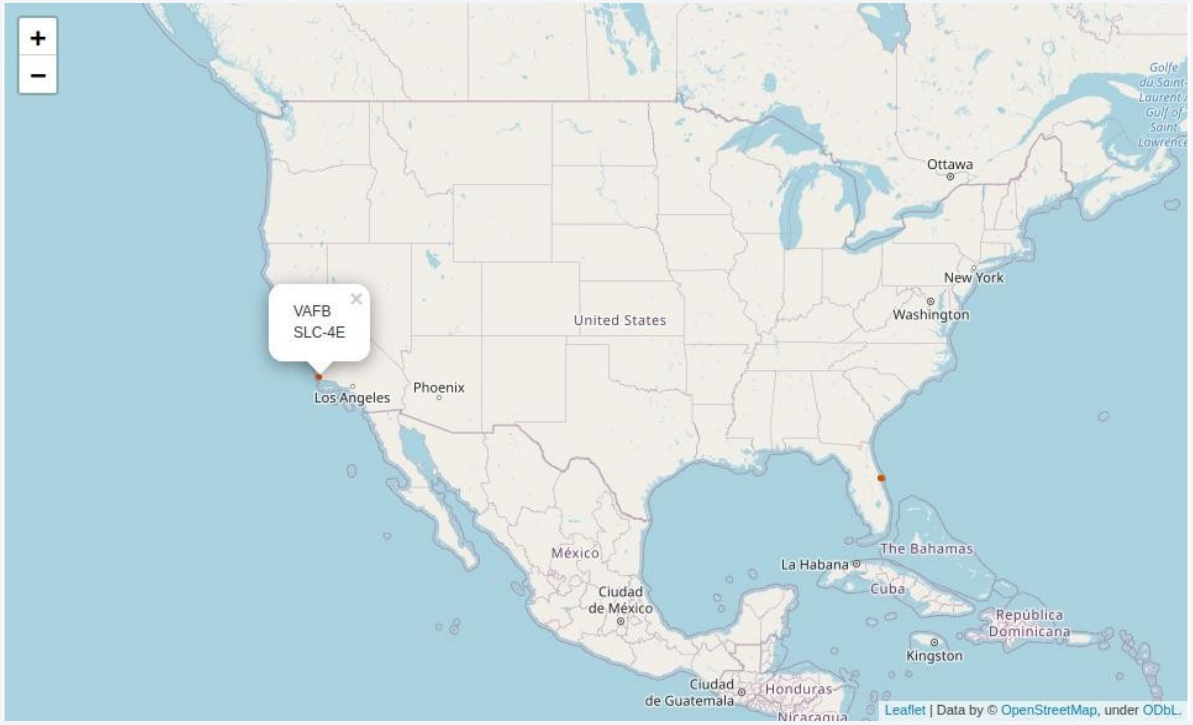
landing__outcome	nr_landing_outcome
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

Section 3

Launch Sites Proximities Analysis

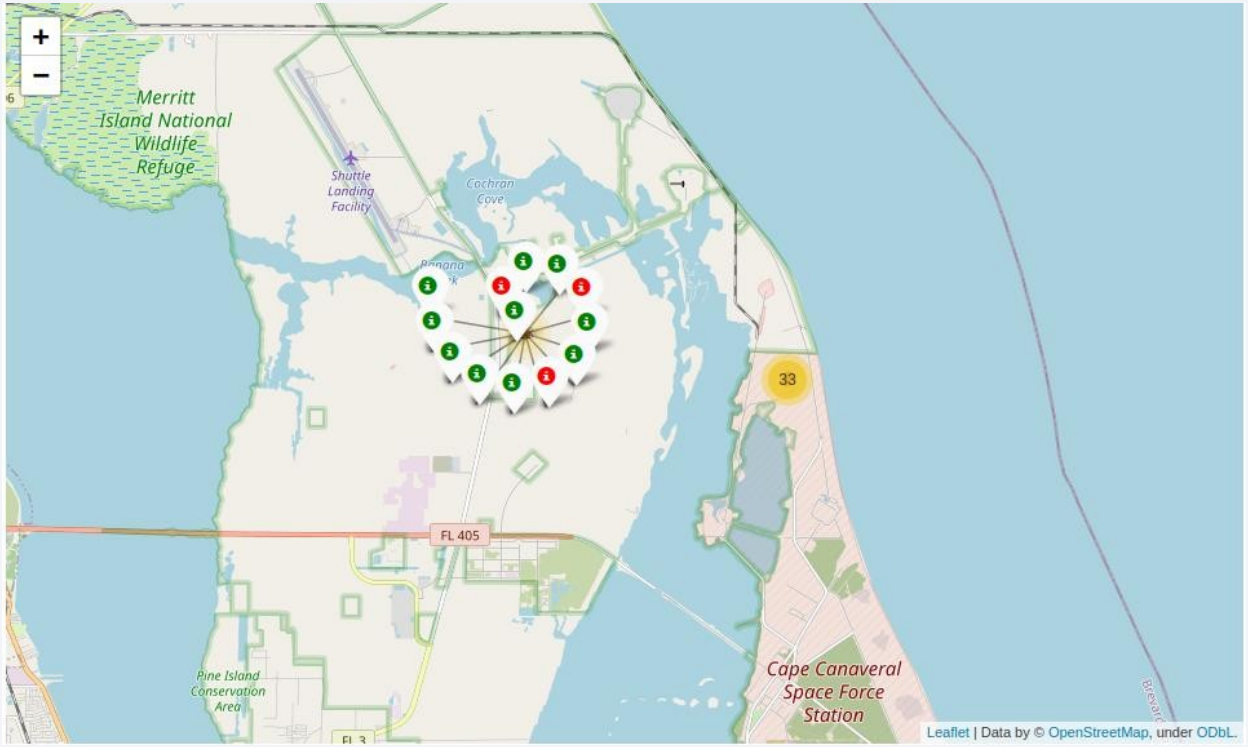
Launch sites' locations

In this folium map the 4 launch sites are displayed and a popup appears if you click on one of them. Zooming in the map, one can see that 3 launch sites are on the East Coast and one on the West Coast.



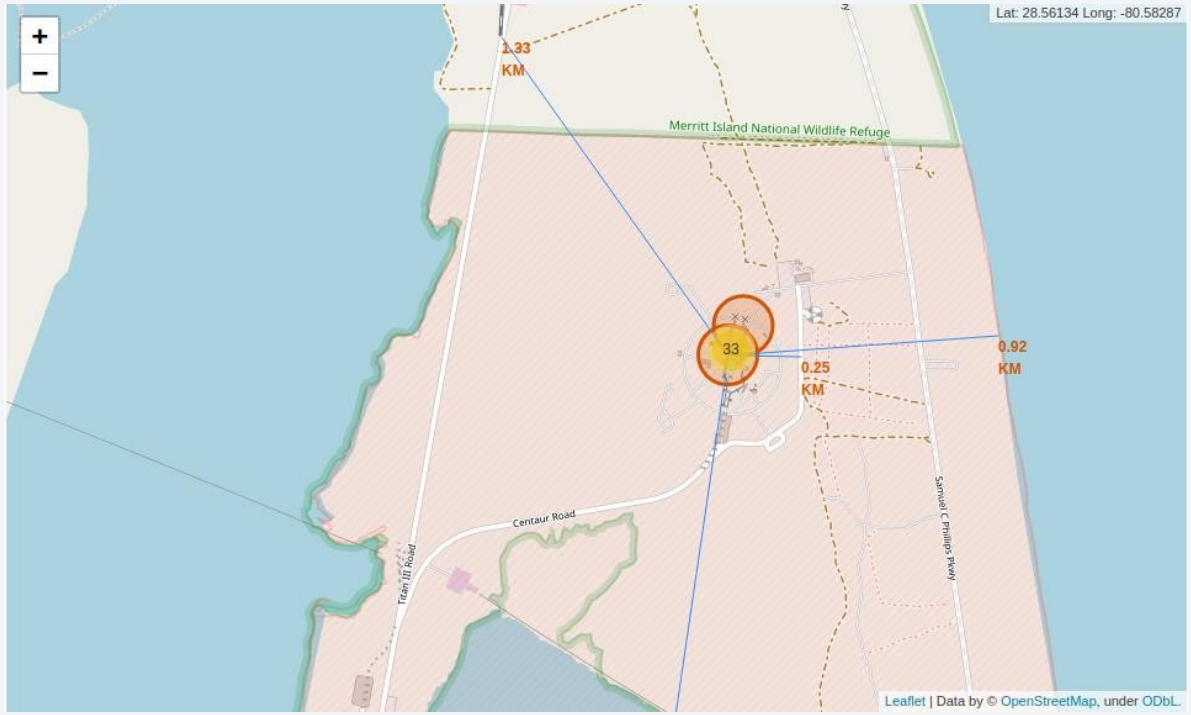
Launch outcomes for each site

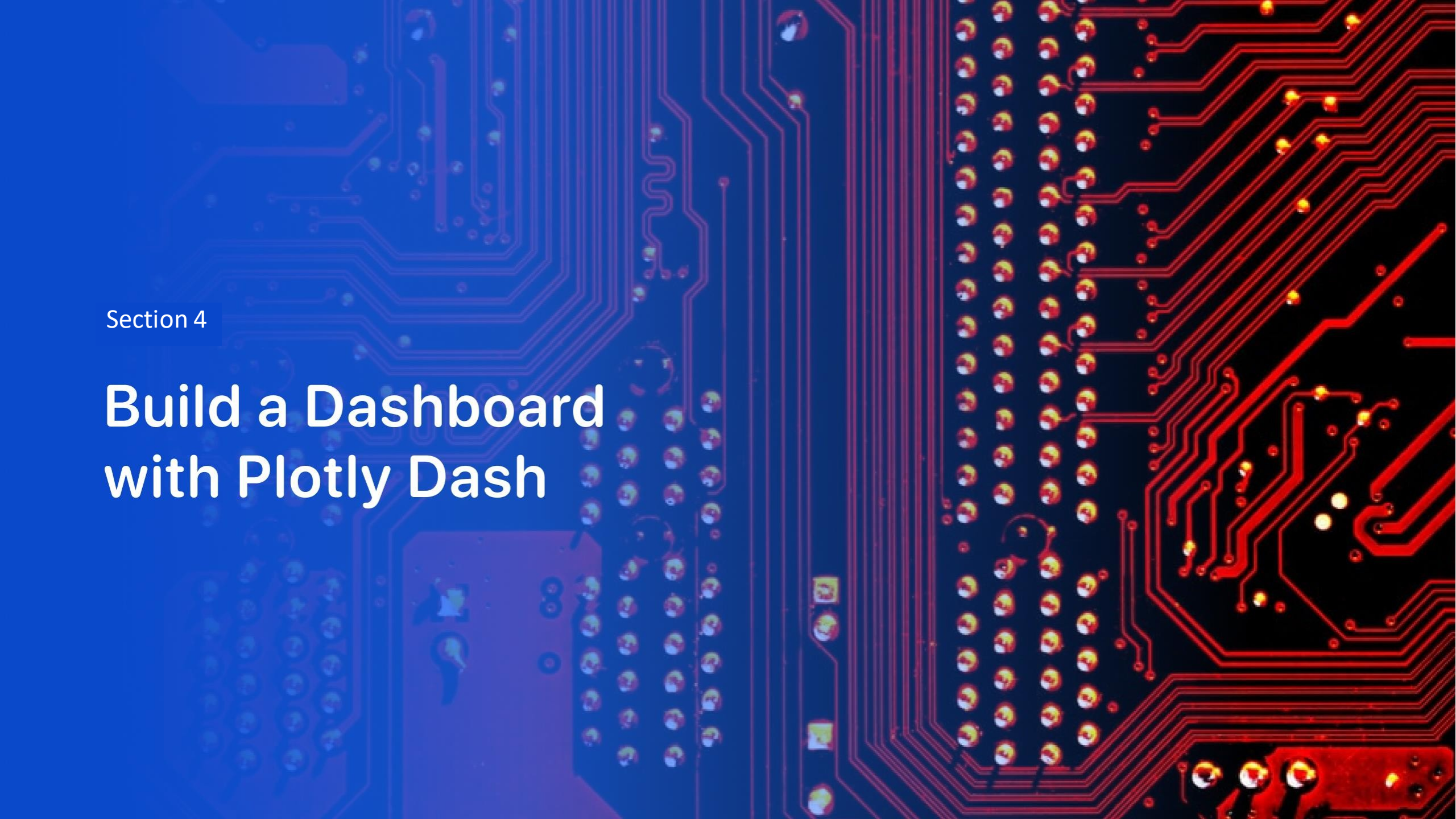
In the second folium map, I added marker clusters that represent launch outcomes for each site. In the image, we can see the marker cluster for KSC LC-39A site: fortunately, in this case most of the launches succeeded.



Distance from relevant structures

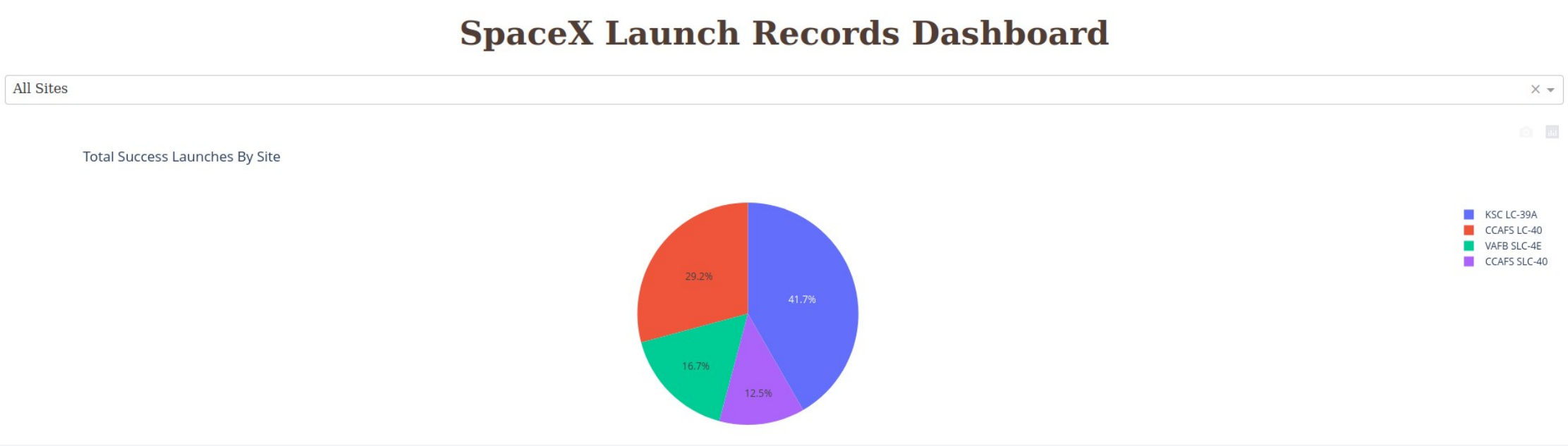
The last map shows the distance of the CCAFS LC-40 launch site from the nearest coastline, railway, highway and city. While the first three are within a 2 km distance, the nearest city is farther (18 km, not shown on the map), arguably for security reasons.




Section 4
Build a Dashboard with Plotly Dash

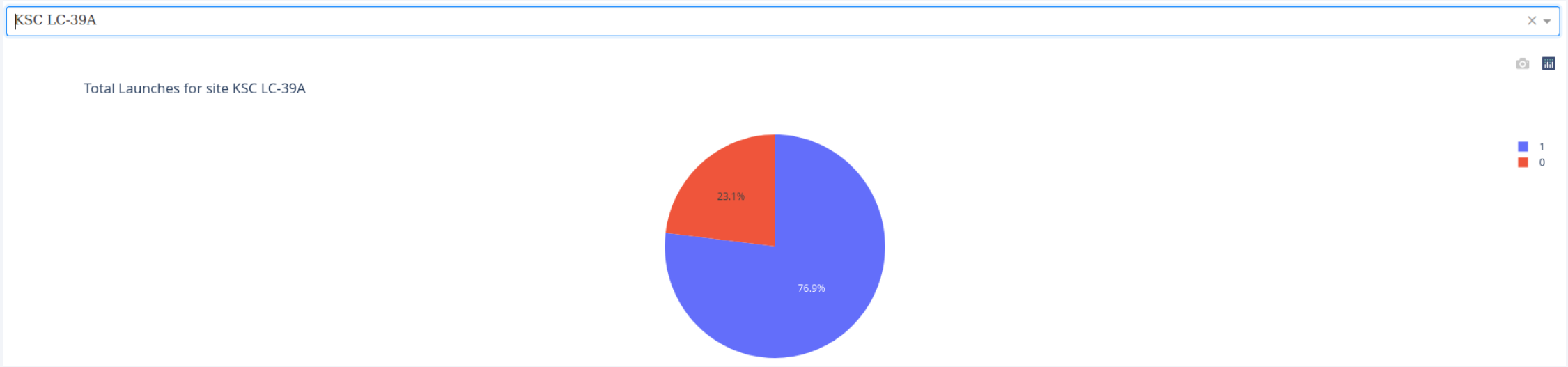
Launch success count for all sites

The first screenshot from the dashboard is a pie chart showing the contribution of each site to successful launches. We can see that the best contribution comes from the KSC LC-39A site.



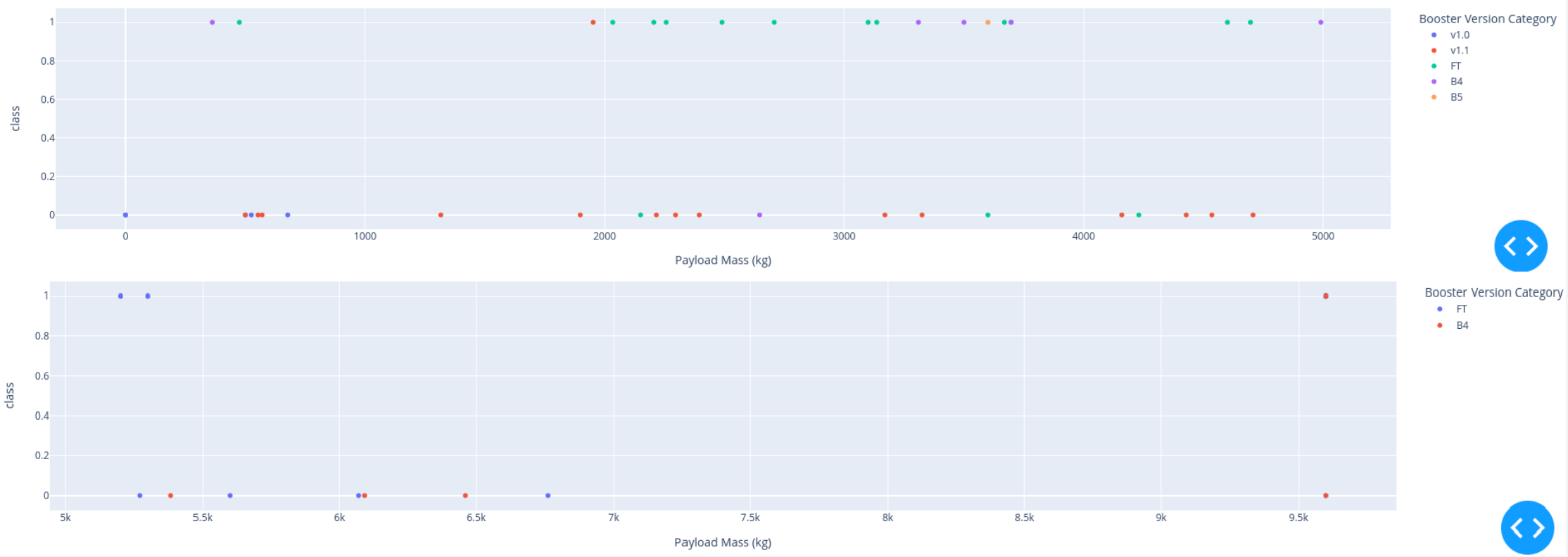
The most successful launch site

Here we show the pie chart for the most successful launch site: the site KSC LC-39A, that exhibits a success rate of 76.9%.



Payload mass vs. Launch outcome

Here we show 2 scatter plots Payload mass vs. Launch outcome, where the ranges for Payload mass are 0-5k and 5-10k respectively and colors represent different booster versions. I comment them in the next slide.



Payload mass vs. Launch outcome (2)

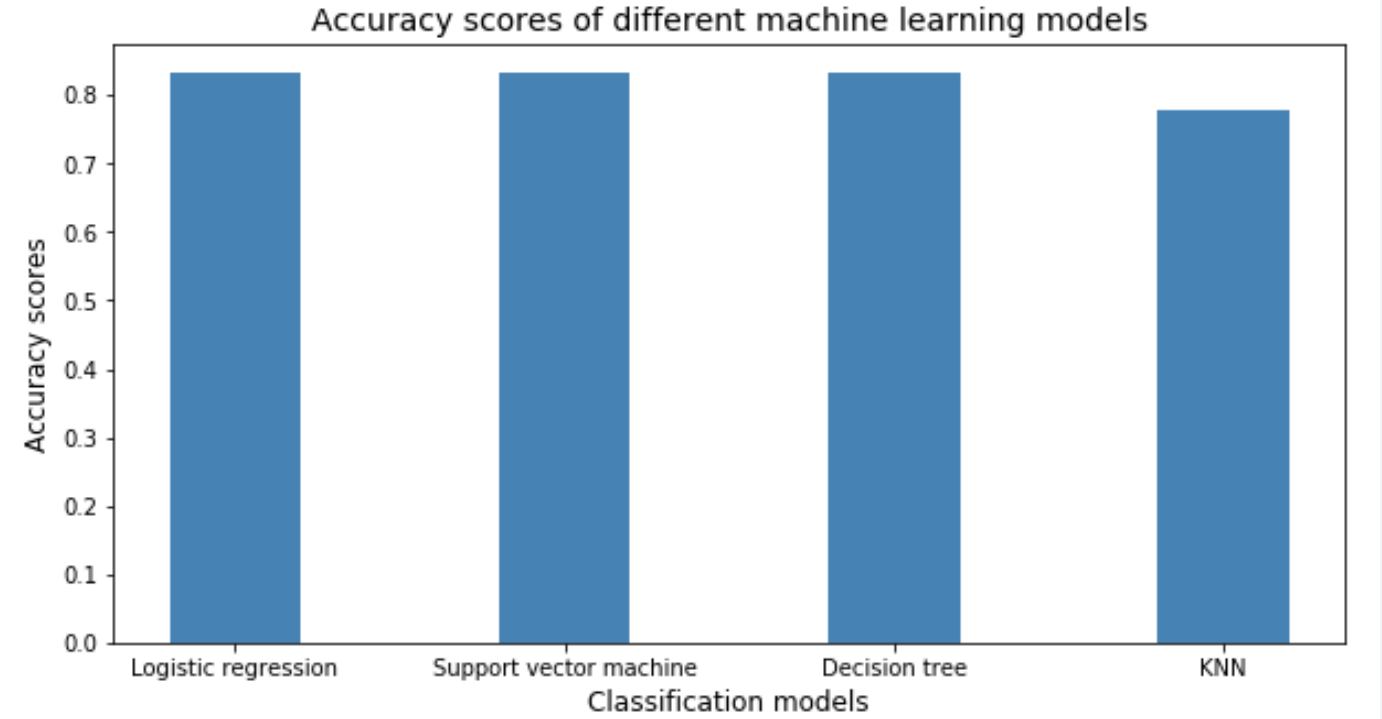
First of all, we can notice that in the lower range there are 5 booster versions (v1.0, v1.1, FT, B4, B5), while in the higher range only FT and B4 booster versions appear. In both ranges, the booster version FT has the highest number of successful launches. In the 0-5k range, the booster version B4 also shows good results, while in the 5-10k range B4 launches are mostly unsuccessful.

Section 5

Predictive Analysis (Classification)

Classification Accuracy

This bar chart shows accuracy scores of the different machine learning models I have built. It is clear that logistic regression, support vector machine and decision tree all achieved the same score (0.83), while KNN achieved a slightly smaller score.



Confusion Matrix

- As an example, here we can see the confusion matrix of the decision tree model.
- The black cells represent wrong predictions: they are clearly very few compared to correct predictions.

Conclusions

In this project, we have developed some tools that can be very useful in order to understand and predict launch outcomes of our competitor SpaceX. In particular, after exploring data asking SQL queries and drawing charts, we have built:

- a Folium map that shows results of SpaceX launch sites on a geographical map;
- a Dash dashboard that allows for interactive visual analytics;
- several machine learning models that predict launch outcomes with high accuracy.

With these tools, we will be able to extract a lot of information about SpaceX's success and take inspiration from it.

Thank you!