Winning Space Race with Data Science

Juan Esteban Moreno Agudelo
November 6, 2022

Outline

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

Executive Summary

Rocket launches are quite expensive due to the high level of technology that is required. On the one hand, there is the need of experts in many engineering and sciences areas, and on the other hand the specialized raw materials needed. However, what if there was an alternative to reduce expenses and make rocket launches affordable for companies with less purchasing power?. When it comes to saving money, engineers must redirect their efforts on all of those key parts which make the process expensive (e.g. first stage components, fuels, logistic). One example is reusing the first stage in rockets as a crucial approach to minimize expenses.

During the past 12 years the company has carried out more than 90 launches and after collecting information of them all, company has available around 50-GB database which allow them to have a full traceability of every launch. Greats insights has been obtained, showing patterns and some issues that need to be taken into account to figure out a good strategy to reduce the cost of every rocket launch.

This report contains useful information about how the company SpaceXB addresses the issue of predicting if the first stage landing could be successful or unsuccessful.

Introduction

Companies involved in the area of rocket launches make huge investments in improving launch performances and at the same time trying to create a product affordable for more people. There are mainly two stages in rockets each of with has motors, fuel tanks and other key components for their operation. Technically there are aspects which are extremely difficult to improve due to the lack of innovative materials (), more powerful fuels to propel the rockets or simply because the implementation of more innovative components would drastically rise the launches prices. As a response to such issues, The integration of the know-how of the company, statistical information and deep learning models to come up with the best approach to predict landing success. The current tendency is to reuse the first stage of the rocket, because it seems to be a quite promising approach when it comes to implementing reduction-cost strategies.

The first stage is highly expensive; it is said that if it could be reused the total rocket launch cost would reduce between 10 and 100 times. Currently companies do test to observe crucial aspects in the rocket launches, collecting and store information to appeal to analysts and involve them within the design and financial stages of the rocket's production.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:

  Collecting data was carried out using an API available for users on the internet. Other approach to collect data was the use of Web Scraping. The previous tasks were carried out through Python employing the appropriate packages.

- Data collection methodology:

  With the collected data, the next step was observed how variables were distributed, the number of null data, amount of features and its type. Identify the variable to predict and transform it in a more useful way to analyze.

- Perform exploratory data analysis (EDA) using visualization and SQL:

  Exploration of data using SQL in order to get some insights of them and employing graphing packages to show the results.

Methodology

Executive Summary

- Perform interactive visual analytics using Folium and Plotly Dash:

  Creation of interactive plots in order to have an outlook more precise. Plotly to set up more dynamics figures and Folium to create maps.

- Perform predictive analysis using classification models:

  Knowing in advance what is output variable its type and already formatted, the implementation of some Machine learning models were done aimed In predict the output variable. Models were tuned using Python packages and tested using unseen data previously. The performance was checked using metrics for categorical variables.

Data Collection

- Describe how data sets were collected.

The first strategy in the collecting data process was carried out using an API in which there are information available about previous rocket launches. The Python packages employed were (Requests, NumPy, Pandas, datetime, Json, Beautifulsoup4). With the URL of the endpoints, the request package is useful to send a request and get the content. The Json package to convert the response content into pandas data frame to work with. The other strategy to collect the data was based on Web Scraping. The request package to get the content and then transform it in a more usable way through Beautifulsoup4. It's important to know some concepts about HTML in order to manipulate the content and extract tables, titles, etc.

Data Collection – SpaceX API

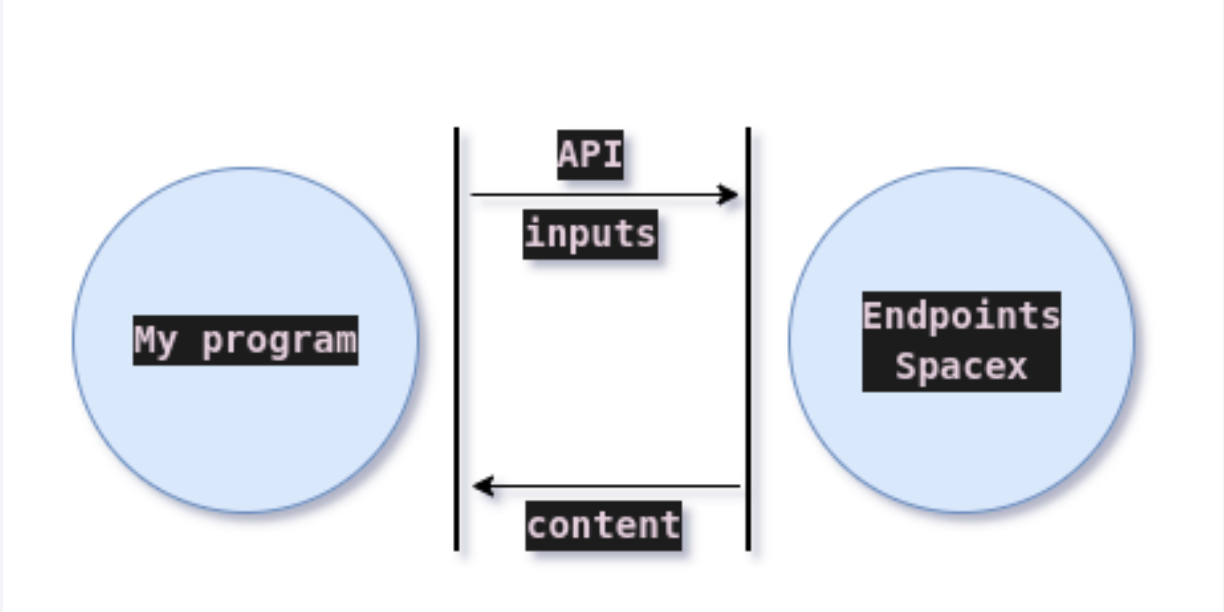
There is a Python code to request the endpoints and save the collected data.

Example of an Endpoint:

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

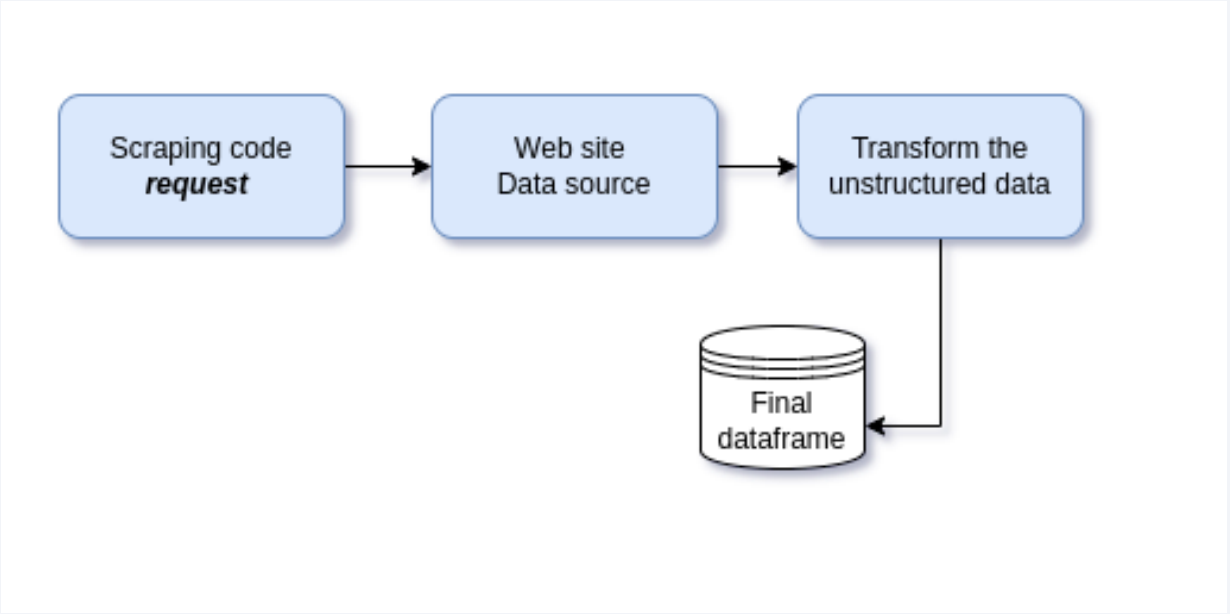
GitHub URL

https://github.com/esteban-93/Data-science-exercises/blob/main/data_collection/jupyter-labs-spacex-data-collection-api.ipynb



Data Collection - Scraping

Request to get the content response, after that the use of beautifulsoup4 to have a more usable content, then the use of pandas, Json and some HTML tags to get the right information and finally store it in a data frame.



Git Hub URL: https://github.com/esteban-93/Data-science-exercises/blob/main/data_collection/jupyter-labs-webscraping.ipynb

Data Wrangling

Describe how data were processed

The main purpose in this point is get some insights of the data. It is important to observe some patterns, check the null data and evaluate the data types. The features and the output variables have been selected in order to continue the data wrangling. In this step the relationship between features and the outputs is evaluated, the types of orbits and its respective outcomes have been counted to see how the relationship between landing outcomes and orbits is.

Finally the output variable was formatted to a categorical variable.

Git Hub link:

https://github.com/esteban-93/Data-science-exercises/blob/main/EDA/labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

Summarize what charts were plotted and why you used those charts

Having the data formatted the plots employed were 1) *catplot* to observe the relationship between two continuous variables with the output class. 2) *scatter plot* to visualize how categorical and continuous variables. 3) *bar plots* to observe the categorical and continuous variables, count and group them.

These plots were implemented because they have the capacity to show or reveal unseen patterns and also the ability to combine continuous and categorical variables.

Git Hub link:

https://github.com/esteban-93/Data-science-exercises/blob/main/EDA/jupyter-labs-eda-dataviz.ipynb

EDA with SQL

Performed queries.

1. select DISTINCT launch_site from SPACEXTABLE
2. select * from SPACEXTABLE

   WHERE launch_site like 'CCA%' LIMIT 5

3. select SUM(payload_mass__kg_) as Total_mass from SPACEXTABLE

   WHERE customer like 'NASA%'

4. select AVG(payload_mass__kg_) as Average_pay_load_mass from SPACEXTABLE

   WHERE BOOSTER_VERSION like 'F9 v1.1'

EDA with SQL

Performed queries.

5. SELECT MIN(DATE) as Date_min from SPACEXTABLE

   WHERE LANDING__OUTCOME='Success (ground pad)'

6. SELECT BOOSTER_VERSION as Booster_version from SPACEXTABLE

   WHERE LANDING__OUTCOME='Success (drone ship)'
   AND payload_mass__kg_ >= 4000

7. SELECT COUNT(MISSION_OUTCOME) from SPACEXTABLE

   WHERE MISSION_OUTCOME like 'Success%' OR

   MISSION_OUTCOME like 'Fail%'

EDA with SQL

Performed queries.

8. SELECT BOOSTER_VERSION from SPACEXTABLE

```
    WHERE payload_mass__kg_ = (SELECT MAX(payload_mass__kg_)
from SPACEXTABLE)
```

9. SELECT LANDING__OUTCOME, BOOSTER_VERSION, LAUNCH_SITE from SPACEXTABLE

```
    WHERE LANDING__OUTCOME = 'Failure (drone ship)' AND YEAR(DATE) = 2015
```

EDA with SQL

Performed queries.

10. SELECT LANDING__OUTCOME from SPACEXTABLE

```
WHERE LANDING__OUTCOME = 'Failure (drone ship)' OR
LANDING__OUTCOME = 'Success (ground pad)' AND
DATE > '2010-06-04' AND
DATE < '2017-03-20'
```

Git Hub link to check the outputs:

https://github.com/esteban-93/Data-science-exercises/blob/main/EDA/jupyter-labs-eda-sql-coursera.ipynb

Build an Interactive Map with Folium

A Folium map was implemented in order to show important points for rocket launches. Having the points identified, A map object was created and the next elements were used:

- ***Circles*** located using coordinates
- ***DivIcon*** to add a test showing the name of the place
- ***MarkerCluster*** to mark the success and unsuccess launches for each site
- ***MousePosition*** to get the coordinates for the mouse over a point on the map.
- ***Polyline*** to draw the lines between two points and measure the distances.

Git Hub link:

https://github.com/esteban-93/Data-science-exercises/blob/main/launch_sites/lab_jupyter_launch_site_location.ipynb

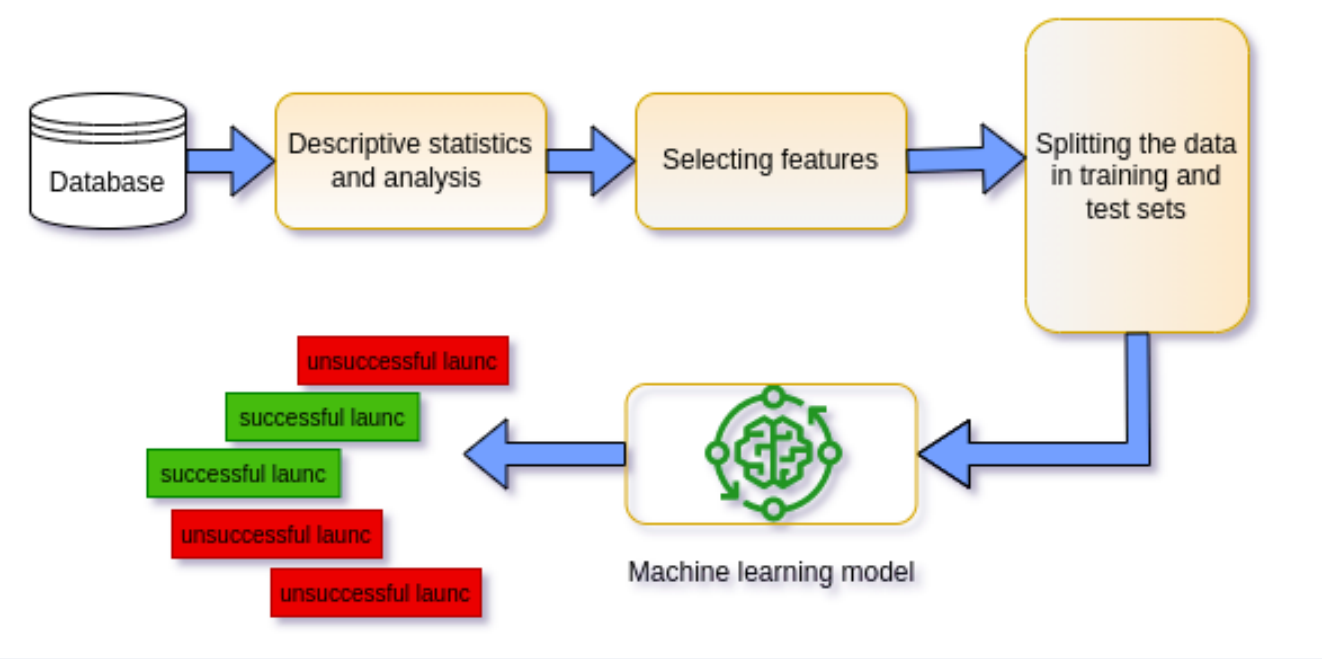
Build a Dashboard with Plotly Dash

- Plots employed where the ***Pie chart*** to show the percentage of success for every launch site. The dynamism was incorporated using a ***DropDown*** component to plot according to the launch site.
- A ***RangeSlider*** with the variation of the Payload mass was used together with a scatter plot to show the relationship with the Booster version category.
- The ***pie chart*** is able to show the all sites together or just show the plot by ***launch site***.
- Finally, HTML components were employed to add titles, labels and locate each component.

Git Hub link:
https://github.com/esteban-93/Data-science-exercises/blob/main/dashboard/spacex_dash_app.py

Predictive Analysis (Classification)

Several classification model were built using Scikit-learn python package and the dataset but divided into training and test sets. As for the evaluation process is important to define the metric, it has been used the confusion matrix to verify how well the model predict the categorical output. Finally, unseen data has been used to verify the predictive power of the models and checking their performances.

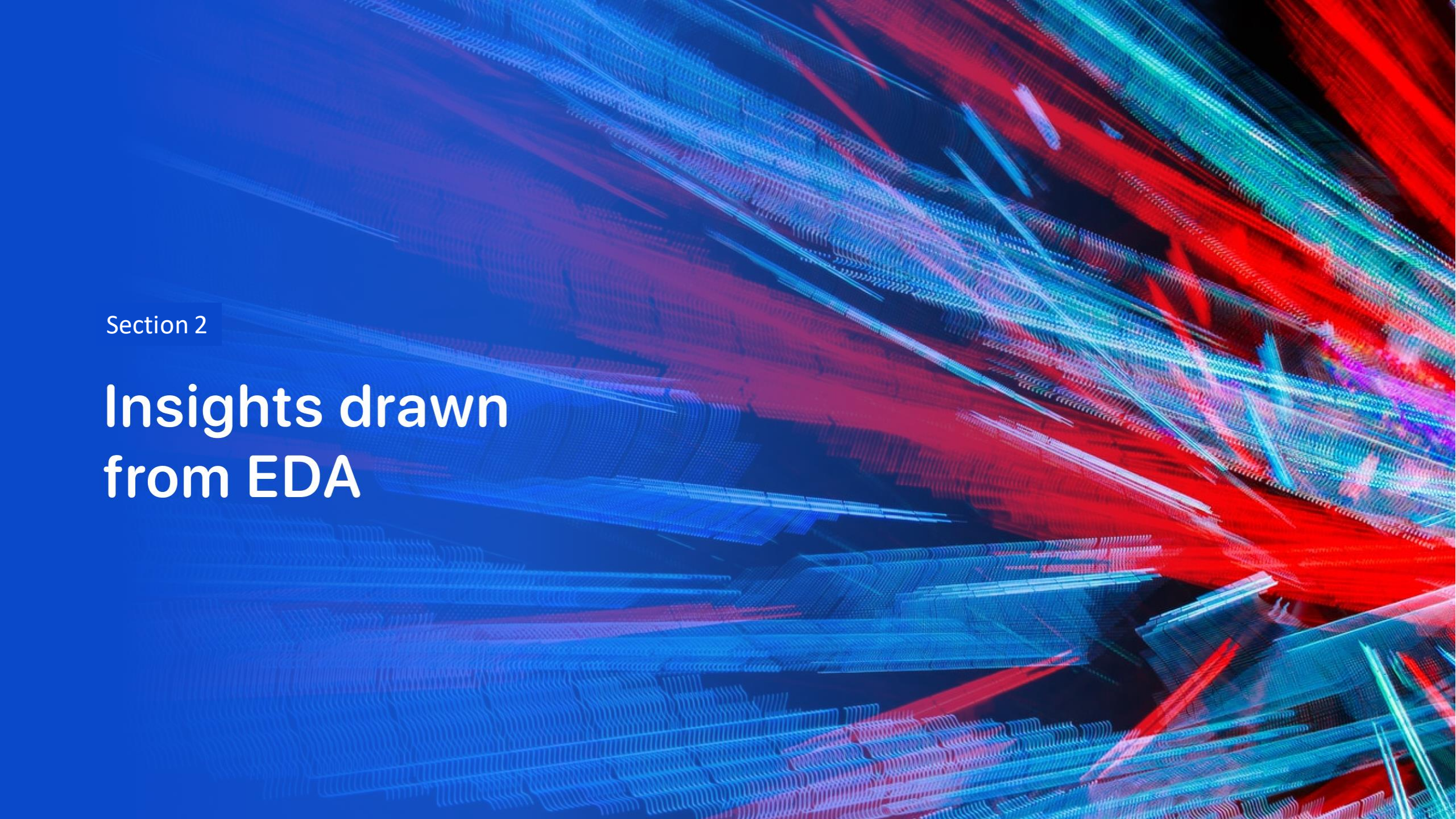


Results

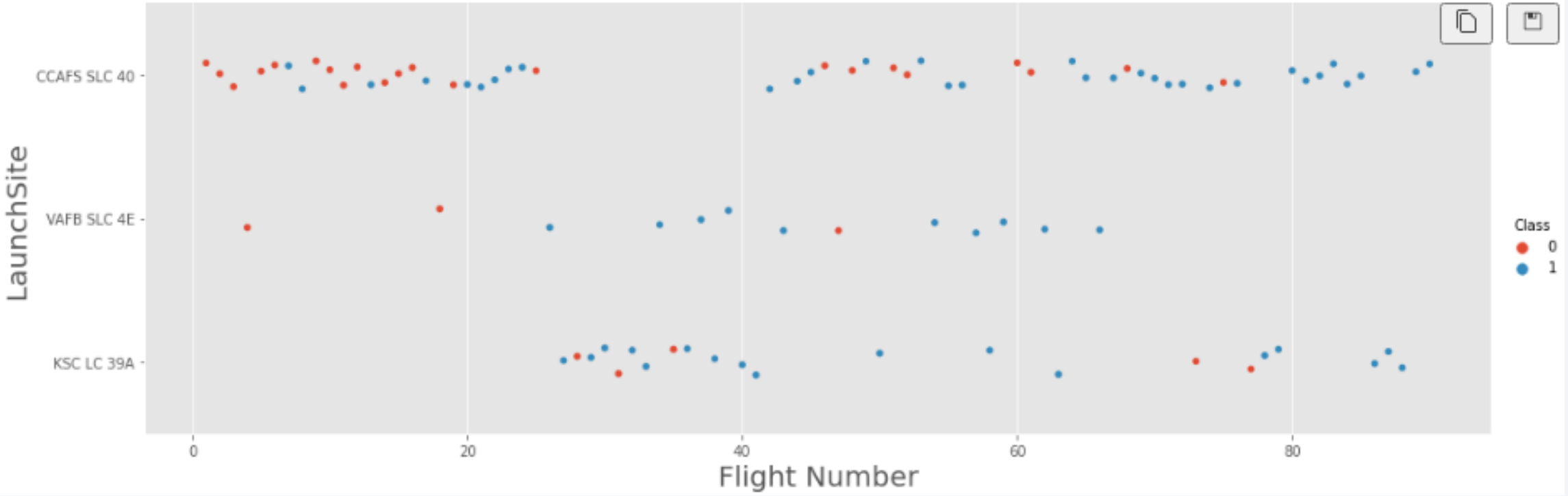
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

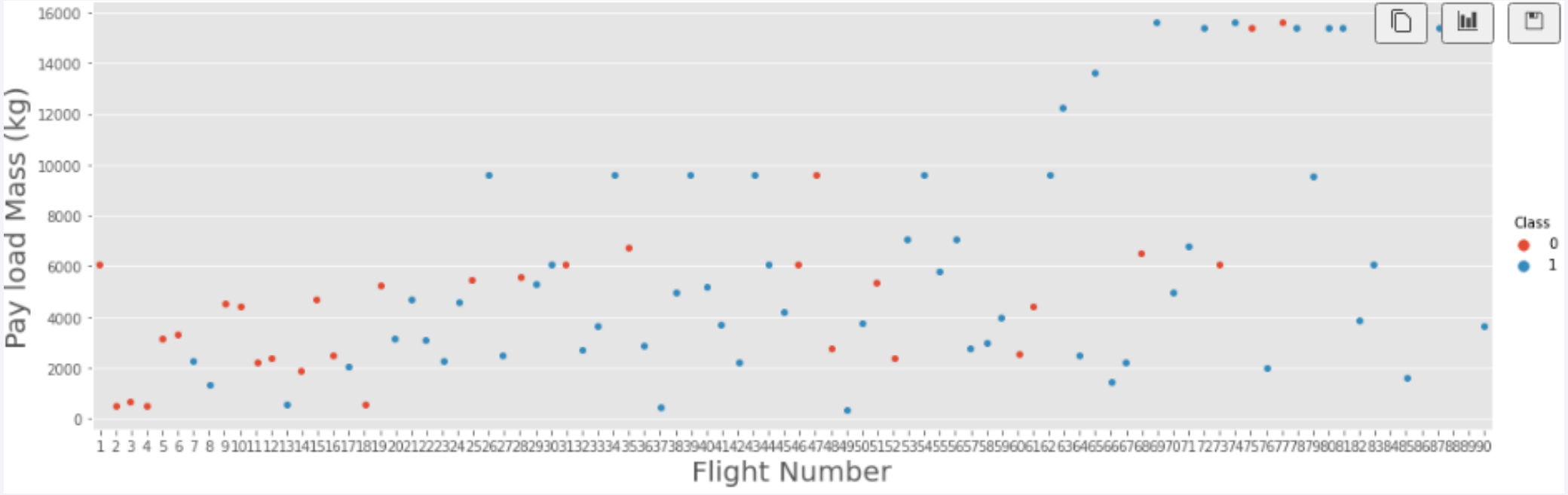


Flight Number vs. Launch Site



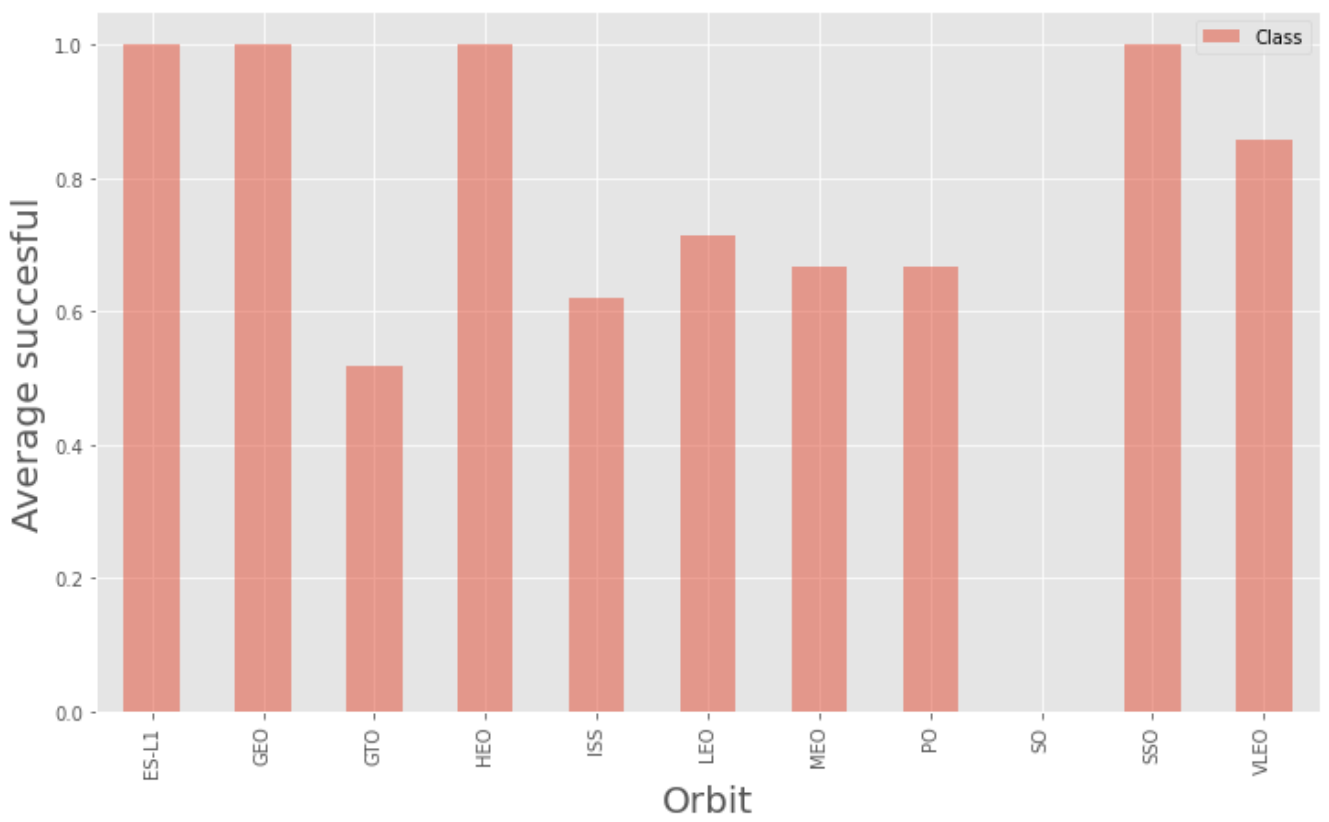
Launches sites *CCAFS SLC 40, KSC LC 39A* shows that as the *Flight Number* increases the probability of having successful landing Is higher. There are fewer rocket launches I the *VAFB SLC 4E* but the probability of have a successful landing is high.

Payload vs. Launch Site



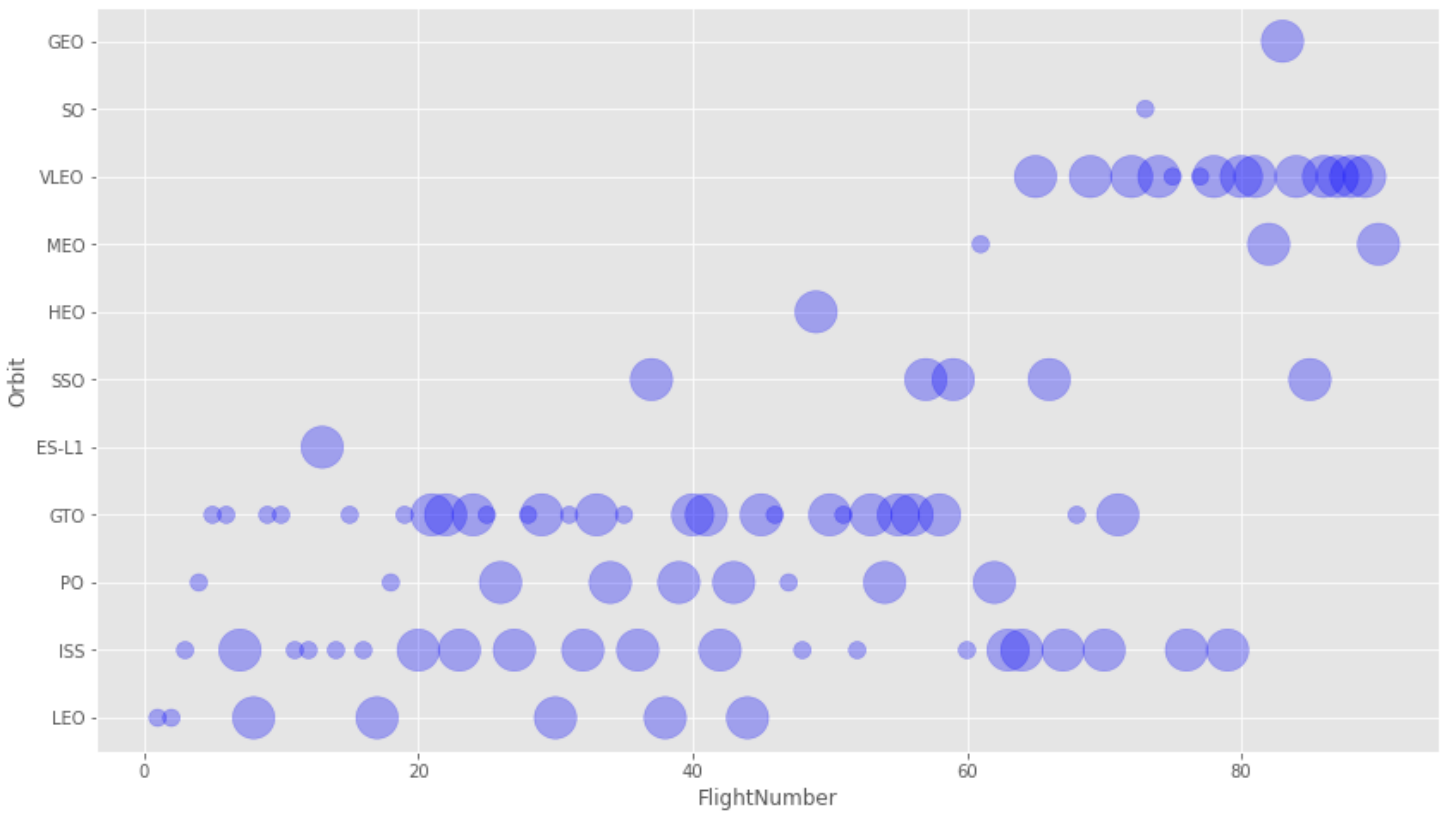
We can plot out the *Flight Number* vs. *PayloadMass* overlay the outcome of the launch. We see that as the *Flight Number* increase the first stage is more likely to land successfully. The *PayloadMass* is also important; it seems the more massive the *PayloadMass*, the less likely the first stage will return

Success Rate vs. Orbit Type



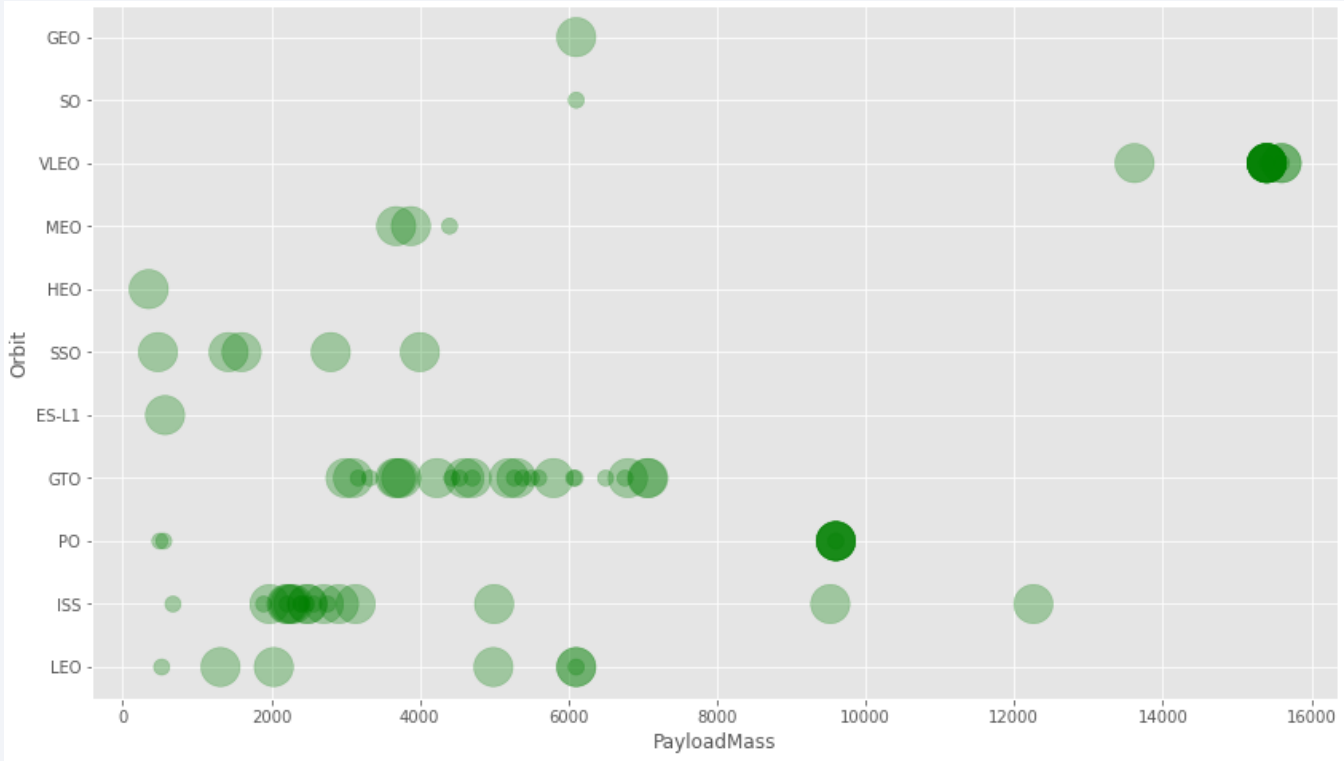
On one hand, The launches in the site VLEO, has the highest probability of successful rate with 0.8571%, 12 out of 14 launches. On the other hand, the sites: GEO, ES-L1, HEO, SSO have successful rate of 100% but it's not possible make conclusions due to the lack of more information they are just, 1,1,1,5 launches respectively. Finally, the least successful rate is in the site: ISS with 0.619%, 13 out of 21 launches were successful.

Flight Number vs. Orbit Type



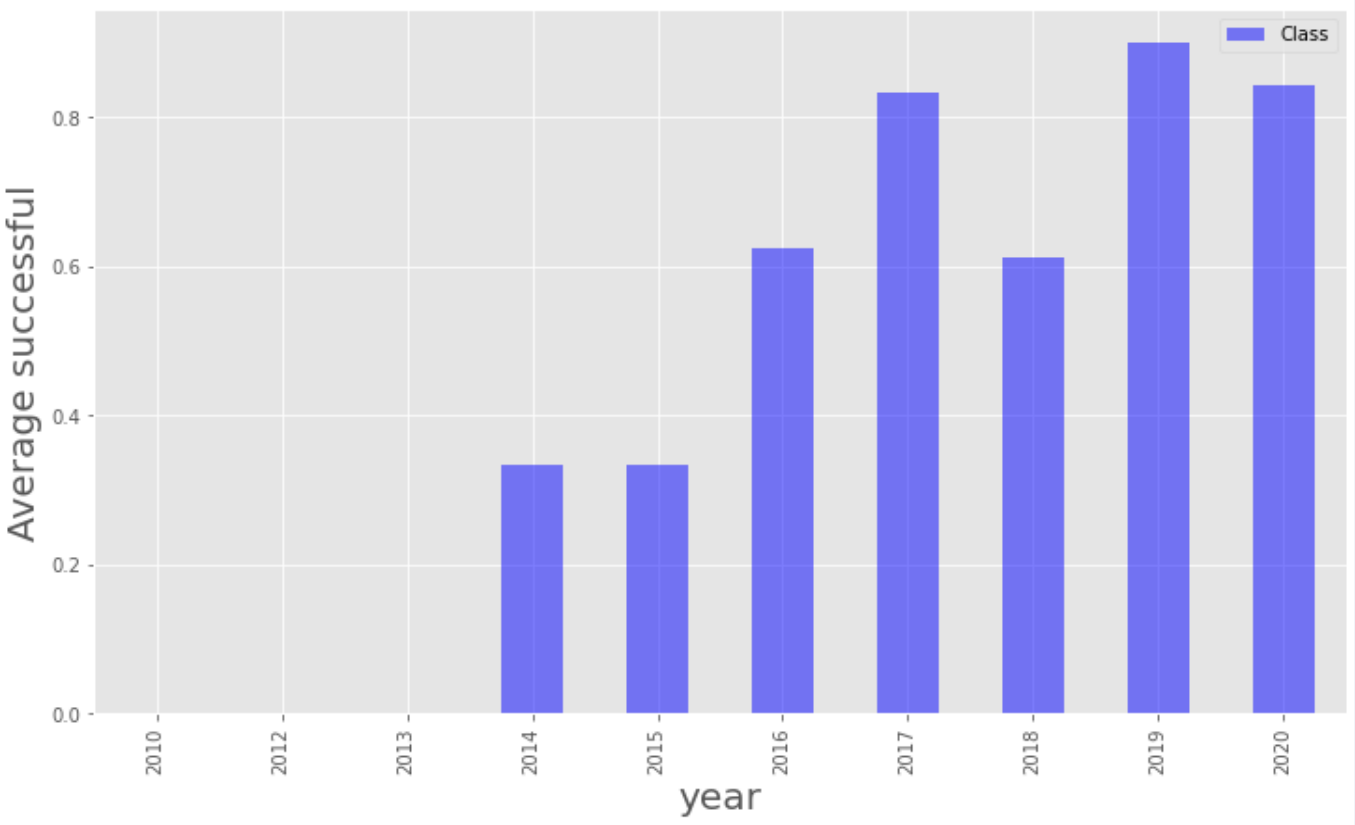
You should see that in the LEO *orbit* the success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO *orbit*. For ES-L1, SSO, HEO, MEO, SO and GEO *orbits* the lack of information does not allow to determine if they are good or bad sites to execute launches. Finally, in VLEO, ISS and GTO as the *Flight Number* increase the successful rate increase as well.

Payload vs. Orbit Type



With the heavy payloads the successful landing rate correspond to Polar, LEO and ISS. However, for GTO we cannot distinguish this well as both positive or negative landings are there in the same proportion.

Launch Success Yearly Trend



You can observe that the success rate since 2013 kept increasing until 2020. The successful rate in 2020 compared with 2021 is fewer apparently due to the pandemic

All Launch Site Names

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

```
CCSFS SLC 40    55
KSC LC 39A      22
VAFB SLC 4E     13
Name: LaunchSite, dtype: int64
```

There are 90 launches registered, most of the launches has been done in CCSFS SLC 40, and the site with fewer launches is VAFB SLC 4E.

Launch Site Names Begin with 'CCA'

```
%%sql
select * from SPACEXTABLE
    WHERE launch_site like 'CCA%' LIMIT 5
```

```
 * ibm_db_sa://whd33722:***@b0aebb68-94fa-46ec-a1fc-1c999edb6187.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:31249/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

The five launches were sent to the LEO orbit where most of the manmade object are. The site where the five launches took place is the same: CCAFS –LC40 and apparently the Payload Mass increased as the amount of launches increased. Finally the landing was failed for two of them and the others weren't attempt.

Total Payload Mass

```
%%sql
select SUM(payload_mass__kg_) as Total_mass from SPACEXTABLE
    WHERE customer like 'NASA%'
```

```
 * ibm_db_sa://whd33722:***@b0aebb68-94fa-46ec-a1fc-1c999edb6187.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:31249/bludb
Done.
```

total_mass
99980

It has been almost 100 tons of mass sent to the outer space using rockets.

Average Payload Mass by F9 v1.1

```
%%sql
select AVG(payload_mass__kg_) as Average_pay_load_mass from SPACEXTABLE
    WHERE BOOSTER_VERSION like 'F9 v1.1'
```

```
 * ibm_db_sa://whd33722:***@b0aebb68-94fa-46ec-a1fc-1c999edb6187.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:31249/bludb
Done.
```

average_pay_load_mass
2928

2928 kg of mass have been transported with booster version F9 v1.1.

First Successful Ground Landing Date

```
%%sql

SELECT MIN(DATE) as Date_min from SPACEXTABLE
    WHERE LANDING__OUTCOME='Success (ground pad)'
```

```
 * ibm_db_sa://whd33722:***@b0aebb68-94fa-46ec-a1fc-1c999edb6187.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:31249/bludb
Done.
```

date_min
2015-12-22

The first successful landing outcome was in 2015-12-22 in (ground pad)

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%%sql

SELECT BOOSTER_VERSION as Booster_version from SPACEXTABLE
    WHERE LANDING__OUTCOME='Success (drone ship)' AND payload_mass__kg_ >= 4000
```

```
 * ibm_db_sa://whd33722:***@b0aebb68-94fa-46ec-a1fc-1c999edb6187.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:31249/bludb
Done.
```

booster_version
F9 FT B1022
F9 FT B1026
F9 FT B1029.1
F9 FT B1021.2
F9 FT B1036.1
F9 B4 B1041.1
F9 FT B1031.2

There are seven booster version which are successful landing in "Success (drone ship)" and the Payload mass higher than 4000kg and fewer than 6000kg.

Total Number of Successful and Failure Mission Outcomes

```
%%sql
SELECT COUNT(MISSION_OUTCOME) from SPACEXTABLE
    WHERE MISSION_OUTCOME like 'Success%' OR
          MISSION_OUTCOME like 'Fail%'
```

```
 * ibm_db_sa://whd33722:***@b0aebb68-94fa-46ec-a1fc-1c999edb6187.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:31249/bludb
Done.
```

1
101

There are a total of 101 mission outcomes, considering successful and unsuccessful missions.

Boosters Carried Maximum Payload

```
%%sql
SELECT BOOSTER_VERSION from SPACEXTABLE
    WHERE payload_mass__kg_ = (SELECT MAX(payload_mass__kg_) from SPACEXTABLE)
```

```
 * ibm_db_sa://whd33722:***@b0aebb68-94fa-46ec-a1fc-1c999edb6187.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:31249/bludb
Done.
```

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

They correspond to Falcon 9 Block 5 and their main configurations have transported the biggest Payload mass on board.

2015 Launch Records

```
%%sql
SELECT LANDING__OUTCOME, BOOSTER_VERSION, LAUNCH_SITE from SPACEXTABLE
    WHERE LANDING__OUTCOME = 'Failure (drone ship)' AND
          YEAR(DATE) = 2015
```

```
 * ibm_db_sa://whd33722:***@b0aebb68-94fa-46ec-a1fc-1c999edb6187.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:31249/bludb
Done.
```

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

The failed landing_outcomes they took place in CCAFS LC-40. The Falcon 9 v1.0, v1.1 were both retired.

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

```
%%sql
SELECT LANDING__OUTCOME from SPACEXTABLE
    WHERE LANDING__OUTCOME = 'Failure (drone ship)' OR
          LANDING__OUTCOME = 'Success (ground pad)' AND
          DATE > '2010-06-04' AND
          DATE < '2017-03-20'
```

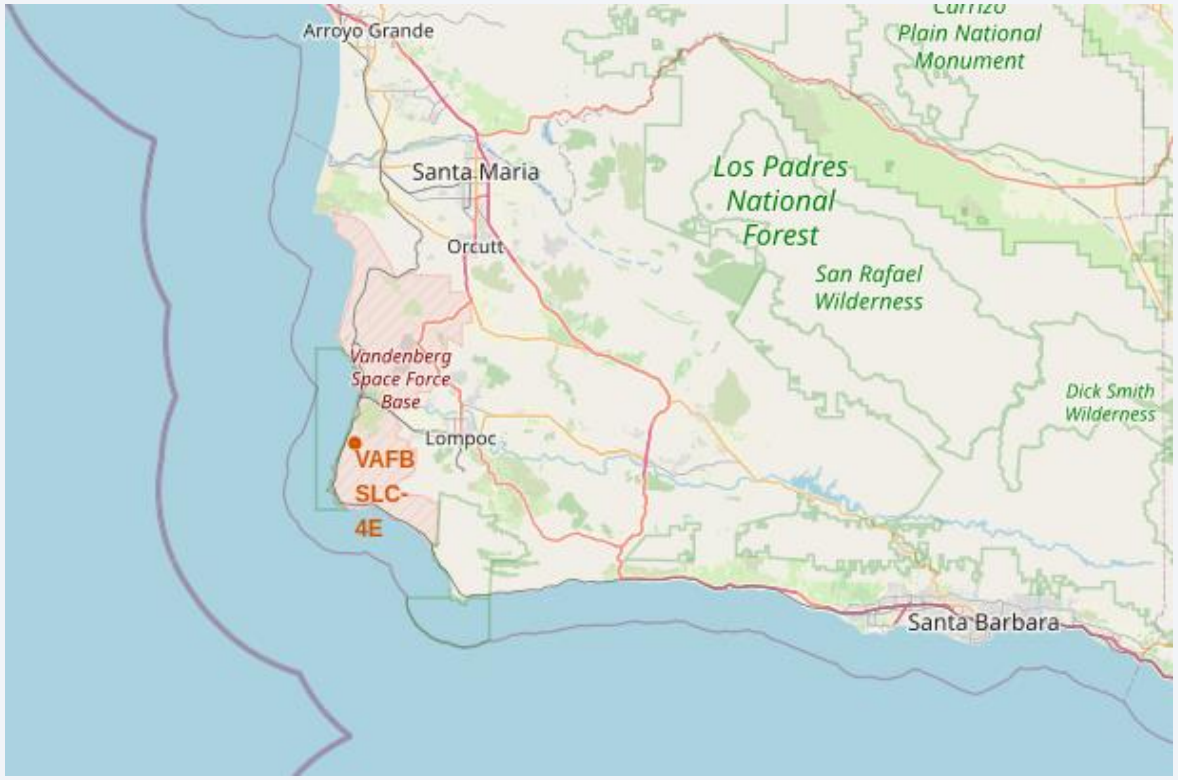
```
 * ibm_db_sa://whd33722:***@b0aebb68-94fa-46ec-a1fc-1c999edb6187.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:31249/bludb
Done.
```

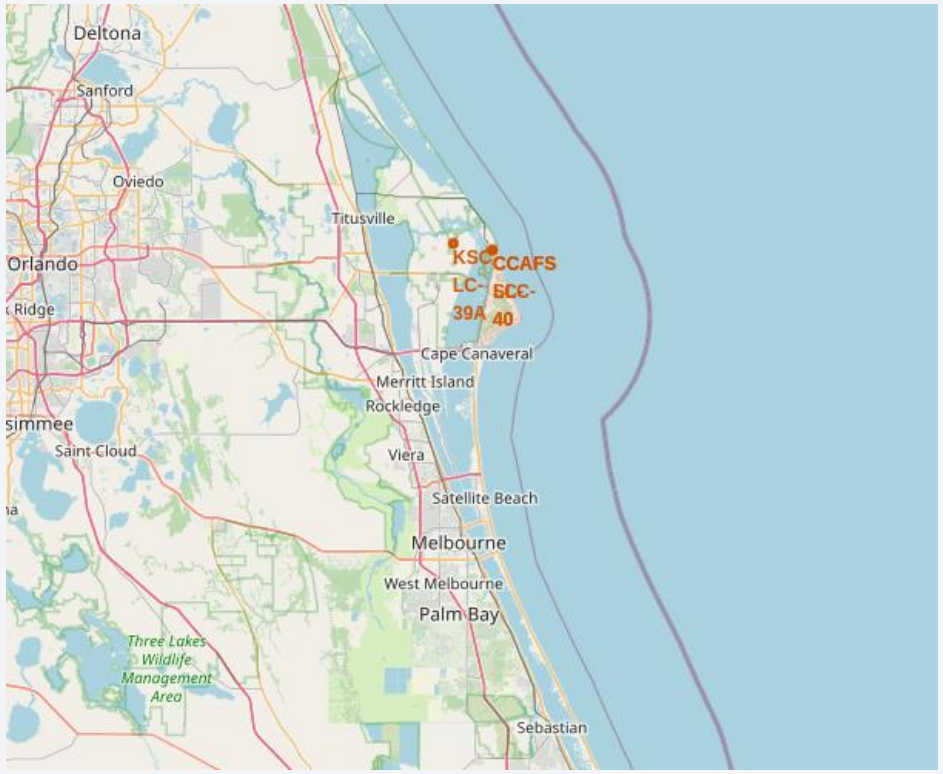
landing__outcome
Failure (drone ship)
Failure (drone ship)
Success (ground pad)
Failure (drone ship)
Failure (drone ship)
Failure (drone ship)
Success (ground pad)
Success (ground pad)

In 6 years there was just three successful (ground pad) and five unsuccessful (drone ship)


Section 3
Launch Sites
Proximities Analysis

Launch sites coordinates





There are three launches sites located by the coast. The importance of that is due to the fact that if somethings wrong happen, the rockets could land in the sea.

Launch outcomes in every site

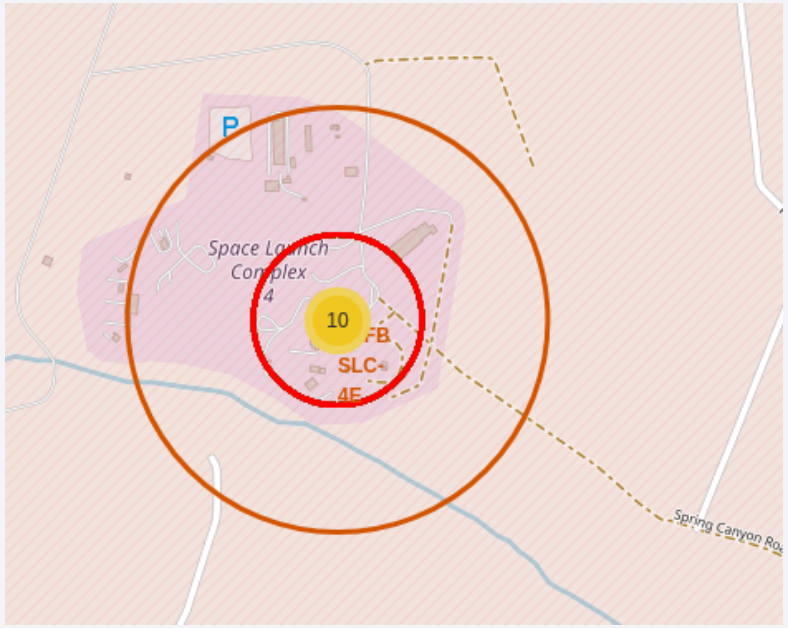


Figura 1 – VAFB SLC-4E

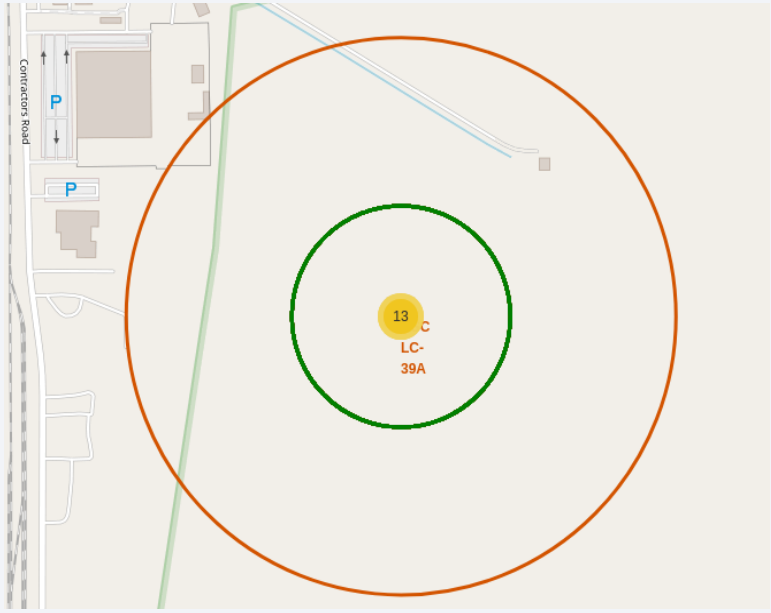


Figura 2 – KSC LC-39A

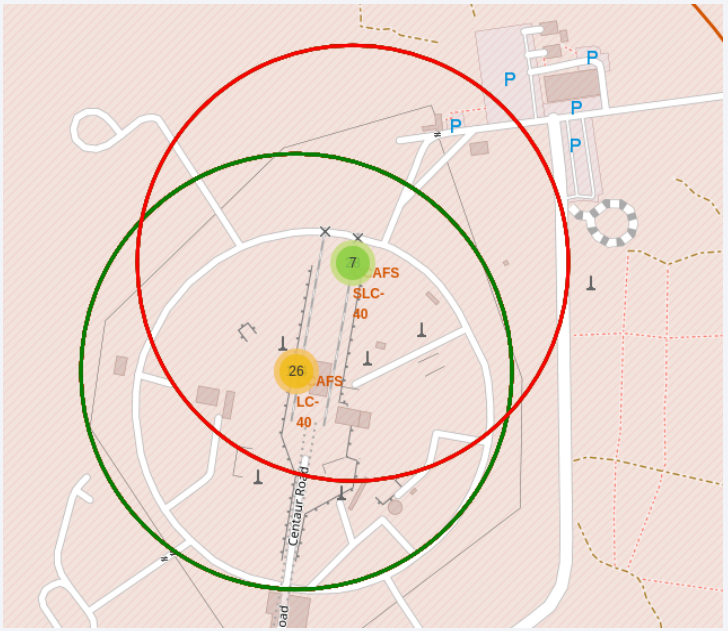
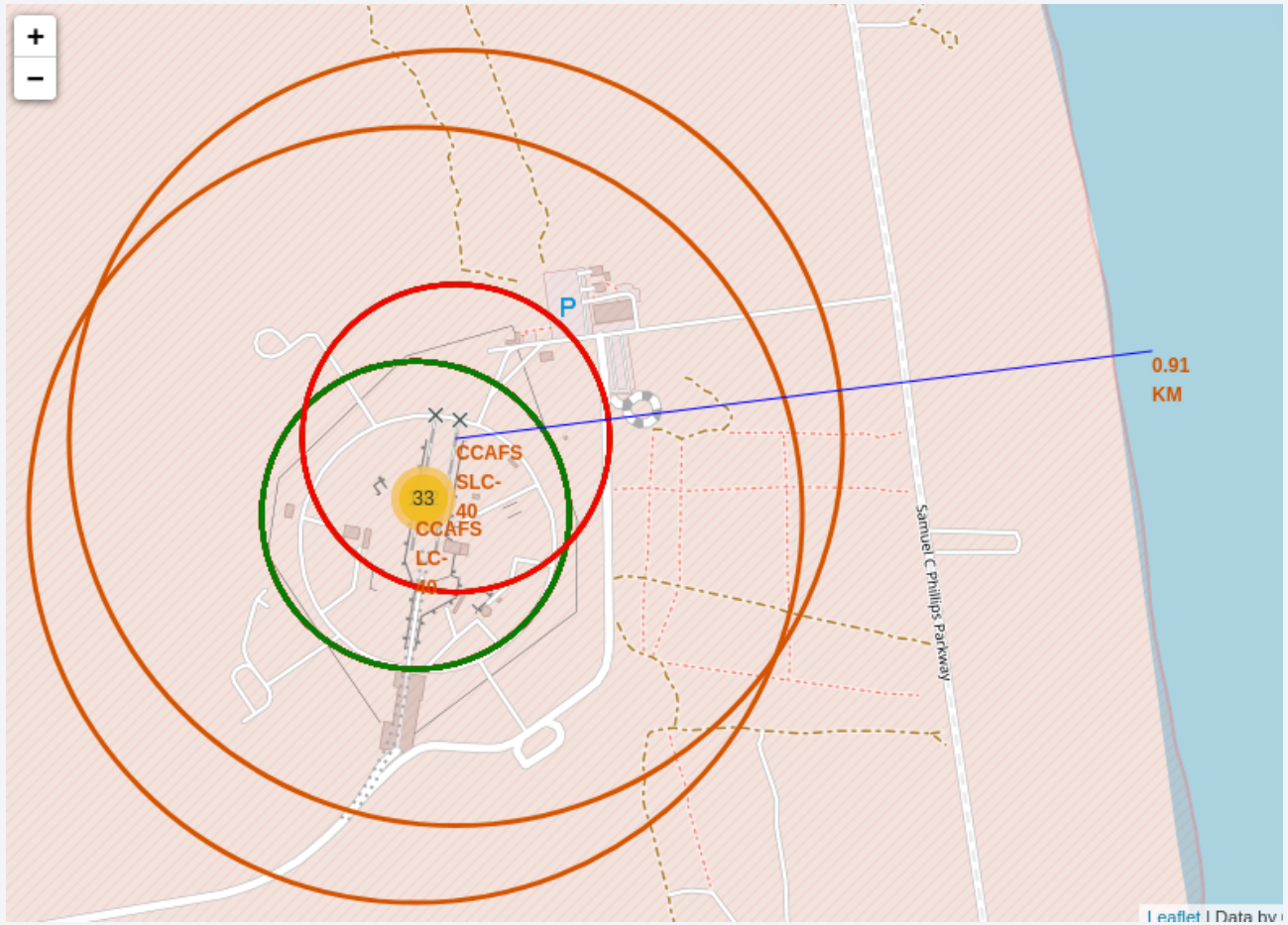


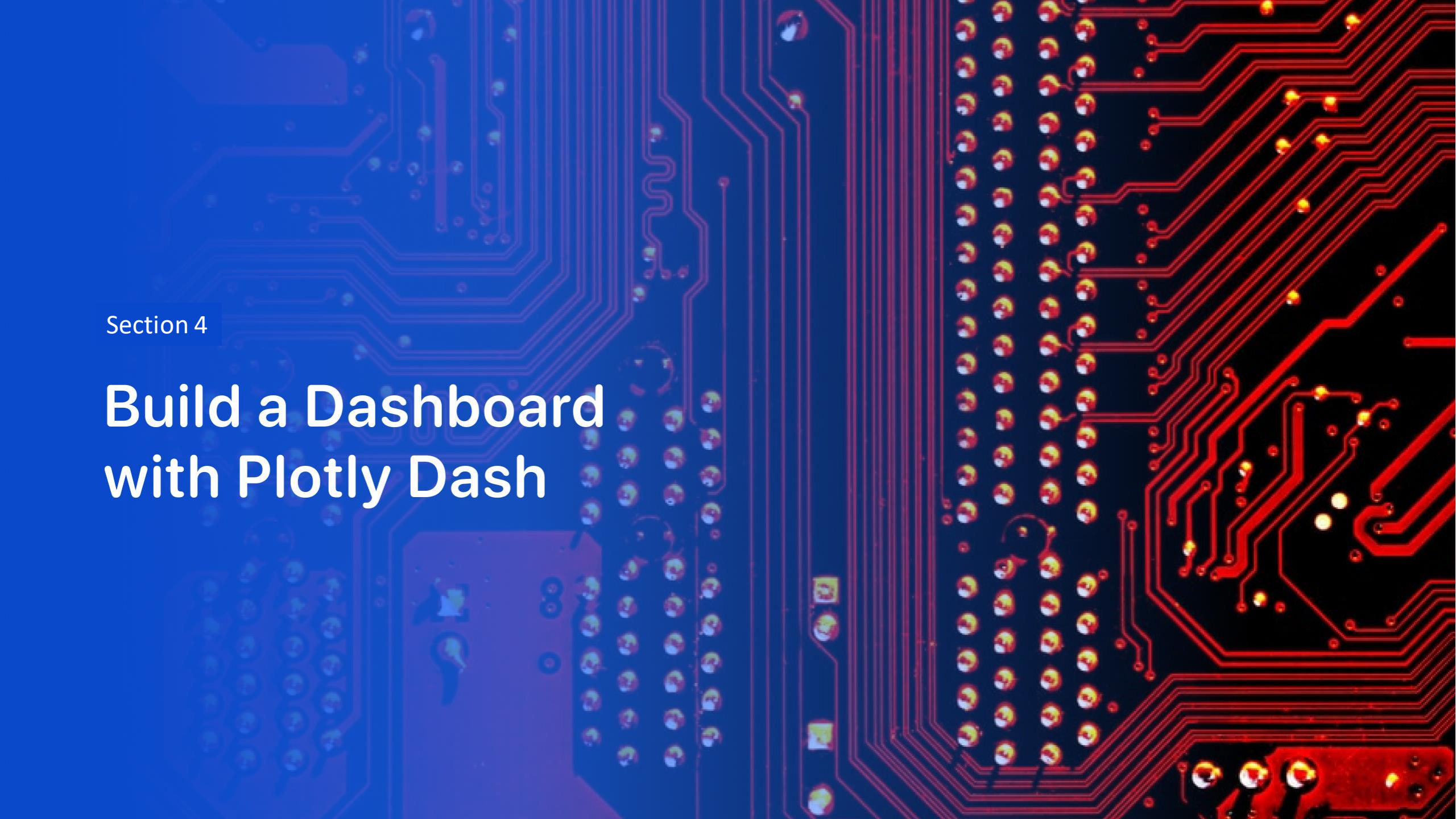
Figura 3 – CCAFS LC-40

In VAFB SLC-4E (fig 1) there were 6 unsuccessful landing and 4 successful ones. In the western coast in KSC LC-39 (fig 2) there were 10 successful landings and 3 unsuccessful ones. In the CCAFS SLC-40 (fig 3) there were 3 successful landings and 4 unsuccessful ones, finally in the site CCAFS LC-40 (fig 3) there was 19 unsuccess landings and 7 success ones. There are in total 56 instances.

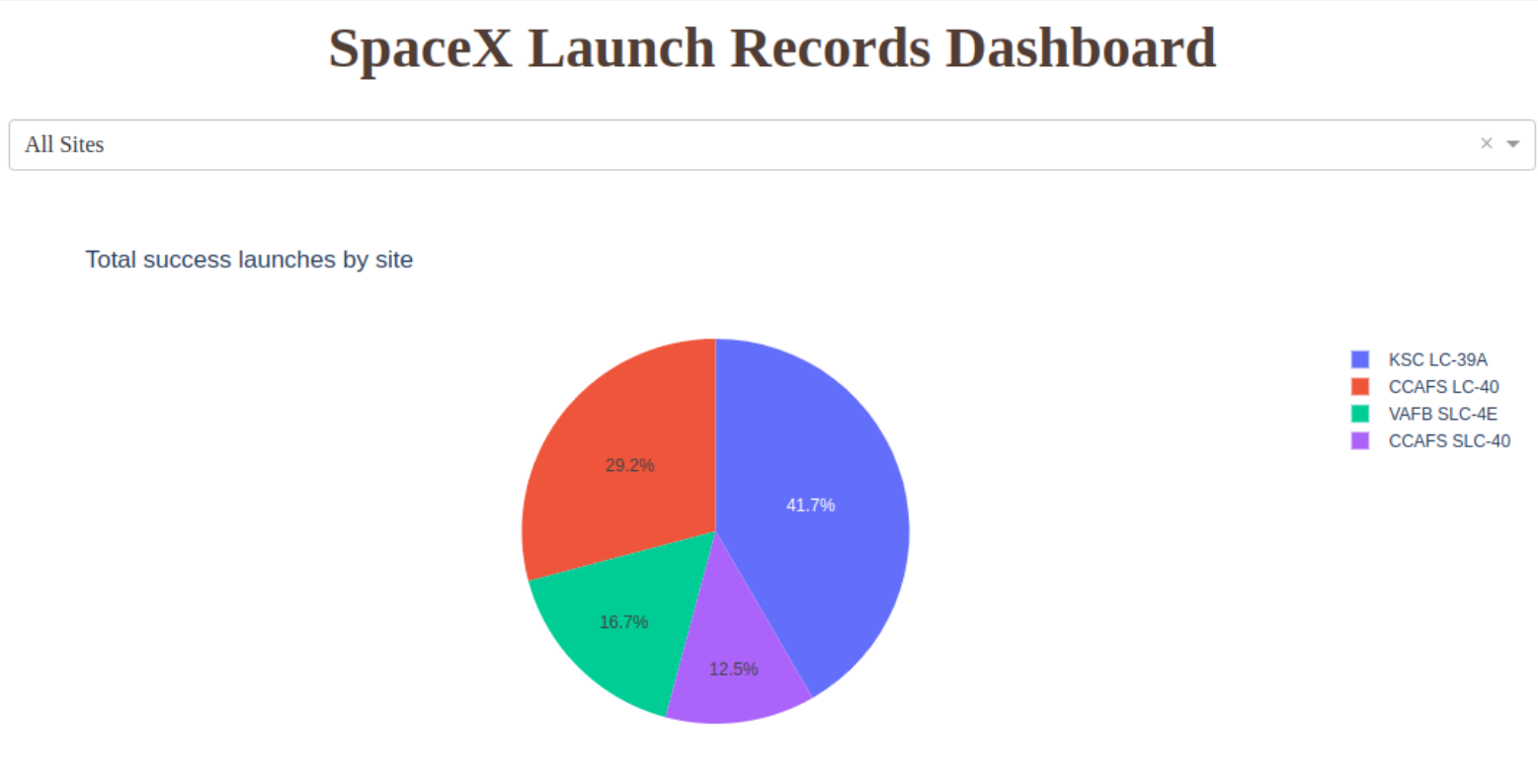
Proximities for each launch site.



CCAFS SLC-40 is about 0.91km away from the coastline. This information is important to estimate external effects or possible situations the rocket could suffer.

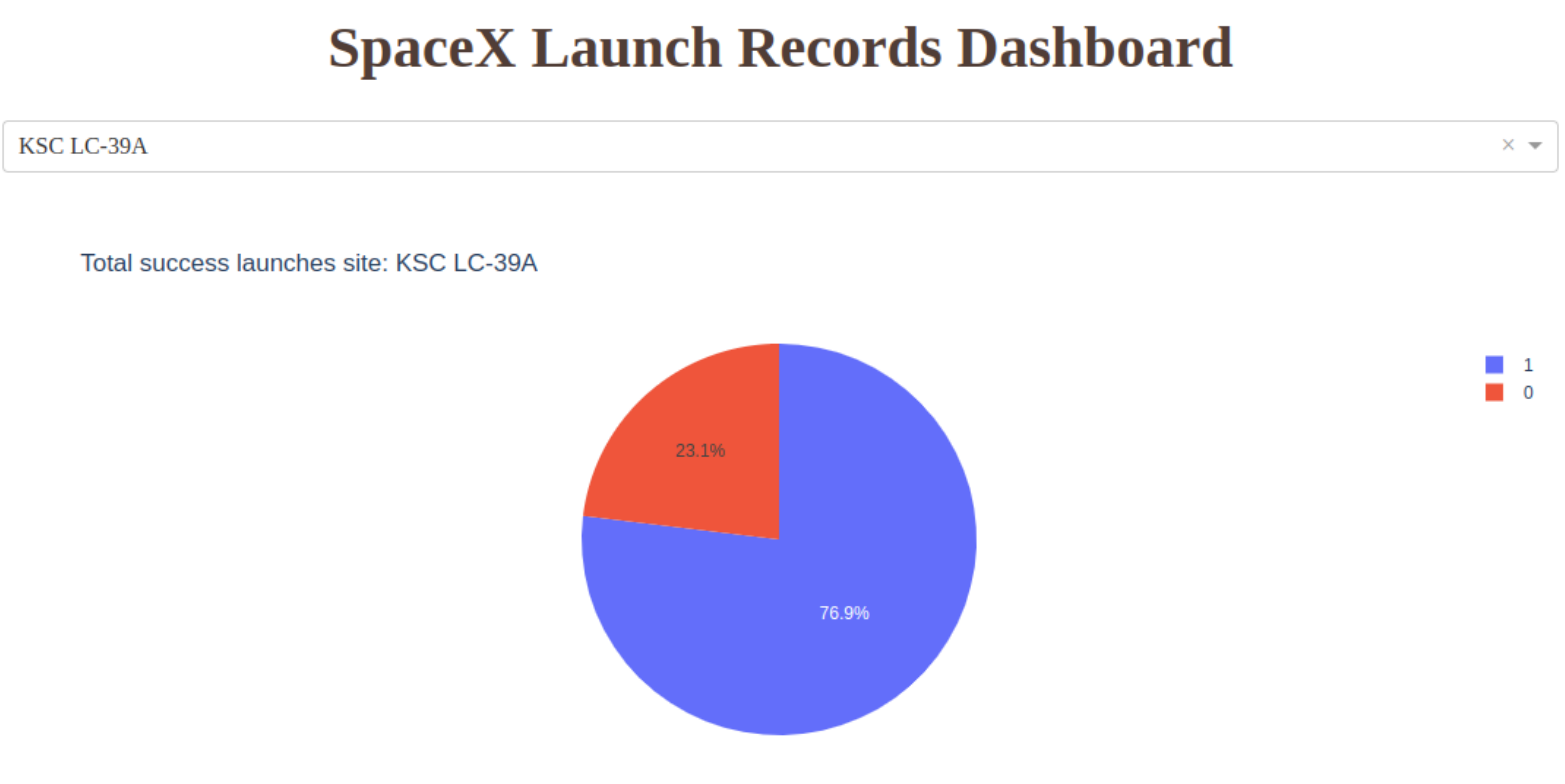

Section 4
Build a Dashboard with Plotly Dash

Successful rate for every launch site



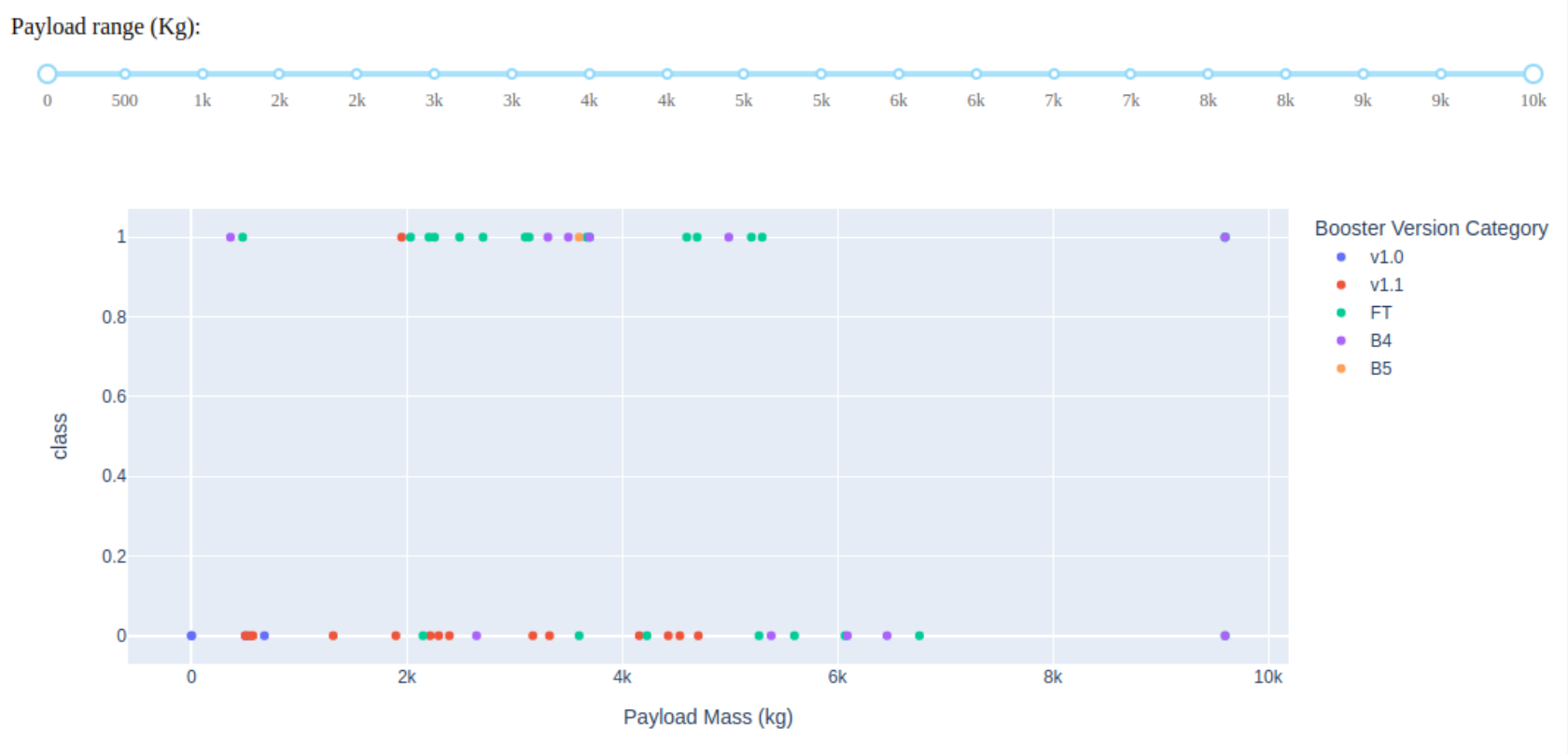
This plot show for the 4 launches sites how good is that place for execute a launch and how the first stage landing turn out. The KSC LC-39A has the highest success and the VAFB SLC-4E the lowest.

Site with the highest successful rate



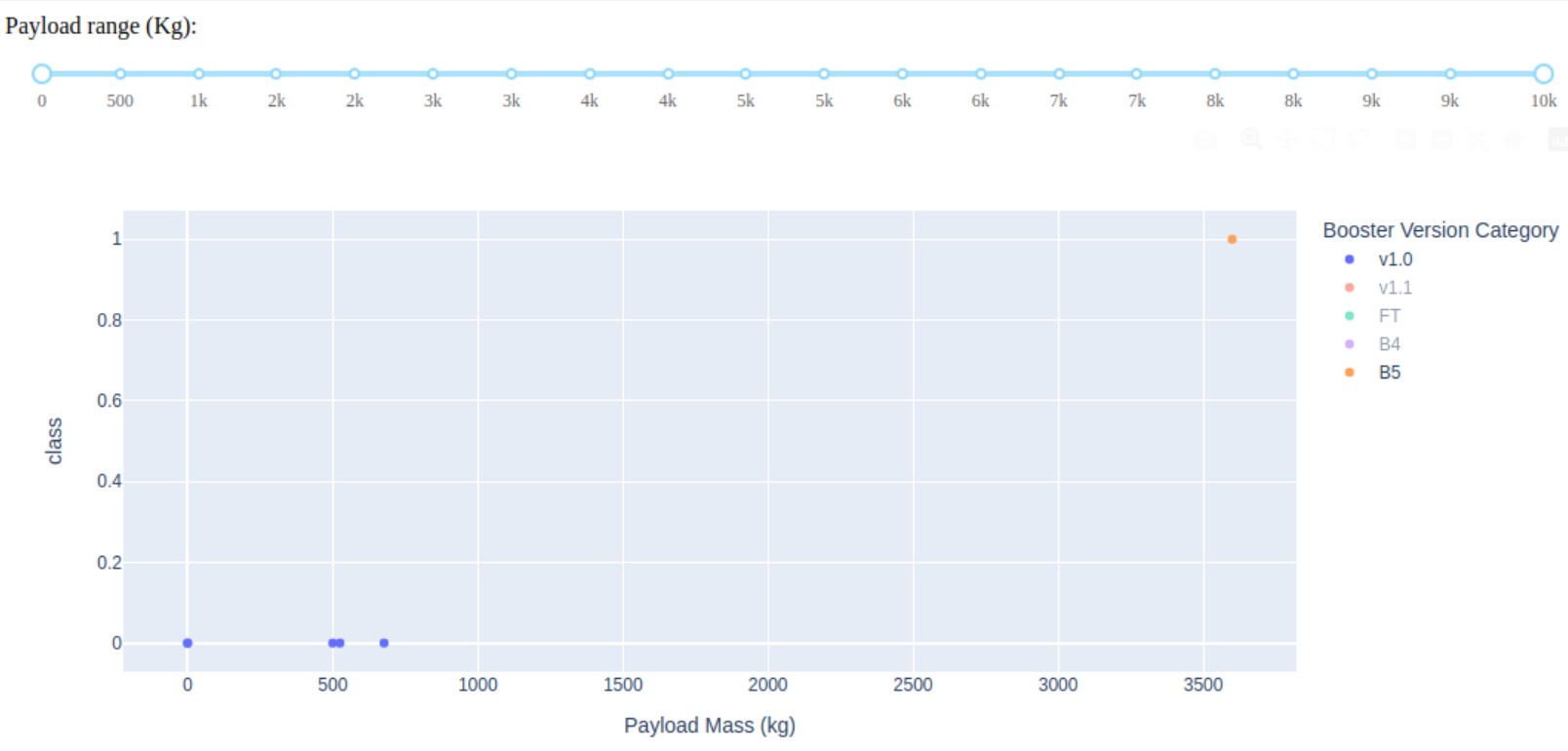
KSC LC-39A is the best place for execute launches due to the high probability of successful landings. 10 out of 13 landings turned out successful.

PayLoad vs. class (landing outcome)
For every Booster version



For the *FT* booster and Payload mass range between (2000 – 5300kg) the successful rate is high. The opposite happens for the *v.1.1* booster in the same range, that is, unsuccessful rate is high. For the *B4* booster is difficult to conclude since the same proportion is both successful and not in the whole range of Payload mass.

PayLoad vs. class (landing outcome)
For every Booster version

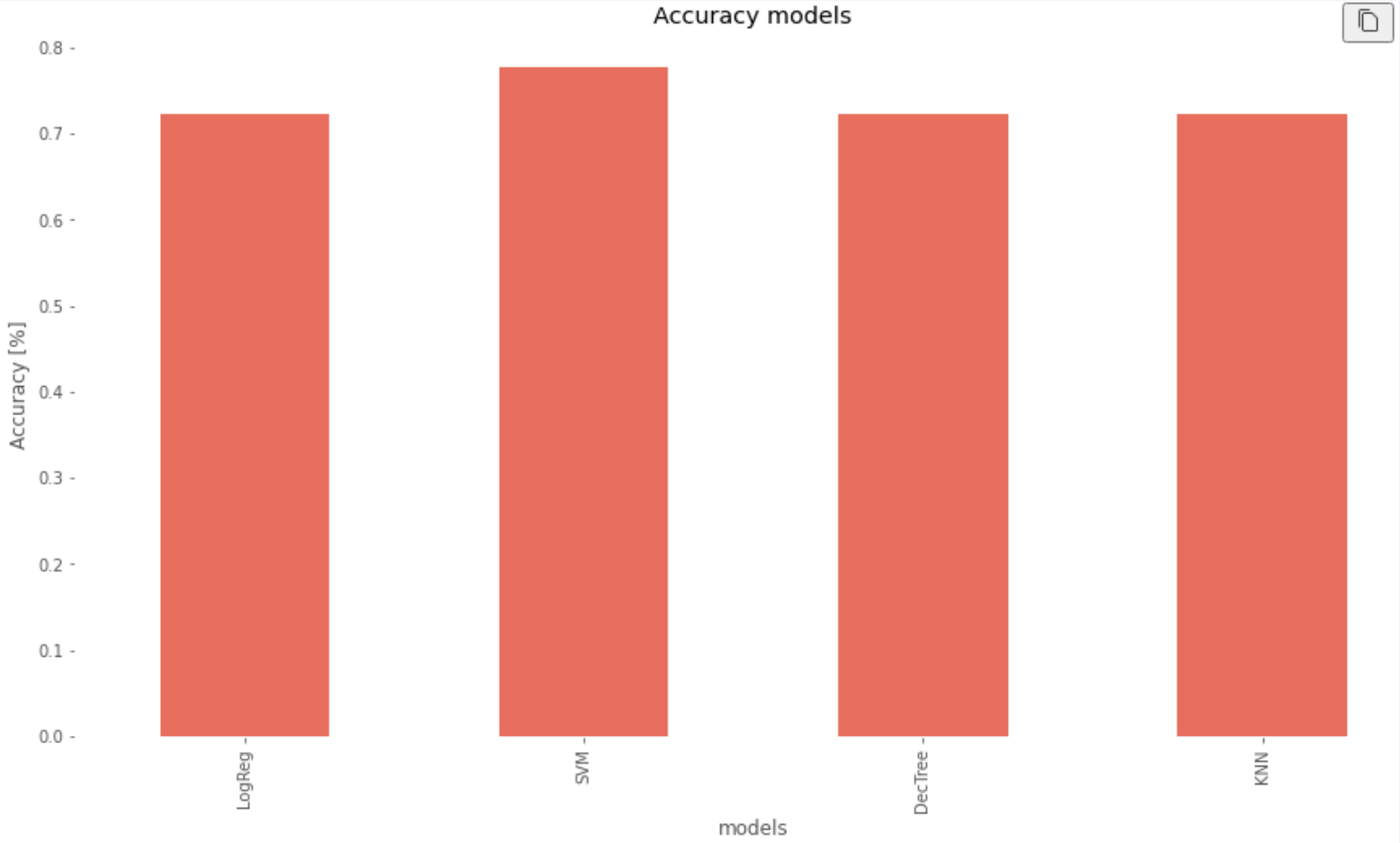


For the *B5* booster there are just one launch and it was successful. For the *v.1.0* there 4 launches and they were unsuccessful.

Section 5

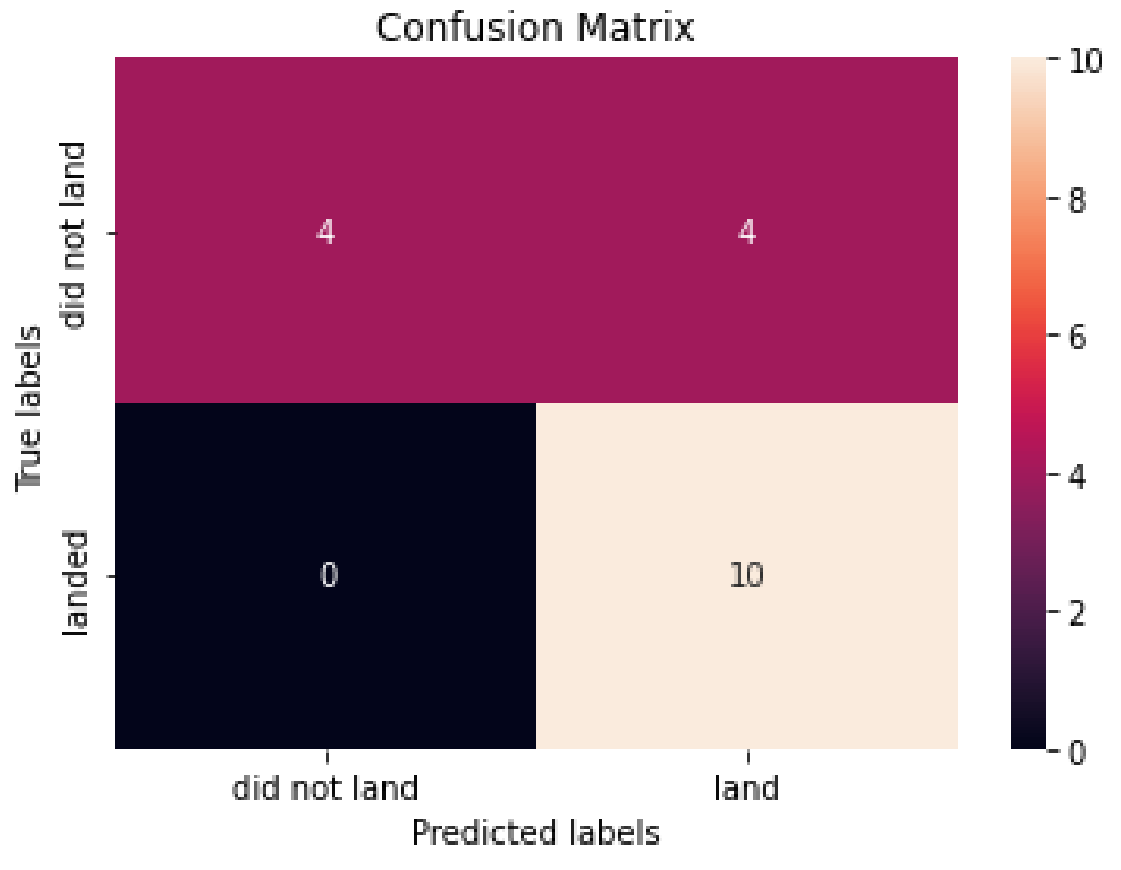
Predictive Analysis (Classification)

Classification Accuracy



The SVM is the best model it has 0.777% of accuracy. The other ones have similar accuracy. Models were trained with 72 samples and tested using 18 samples.

Confusion Matrix



This plot tell us about the true or false values well predicted. That is, 4 samples did not landed and effectively the model predict them well. On the other side, the model predict 10 samples that effectively landed but it went wrong with 4.

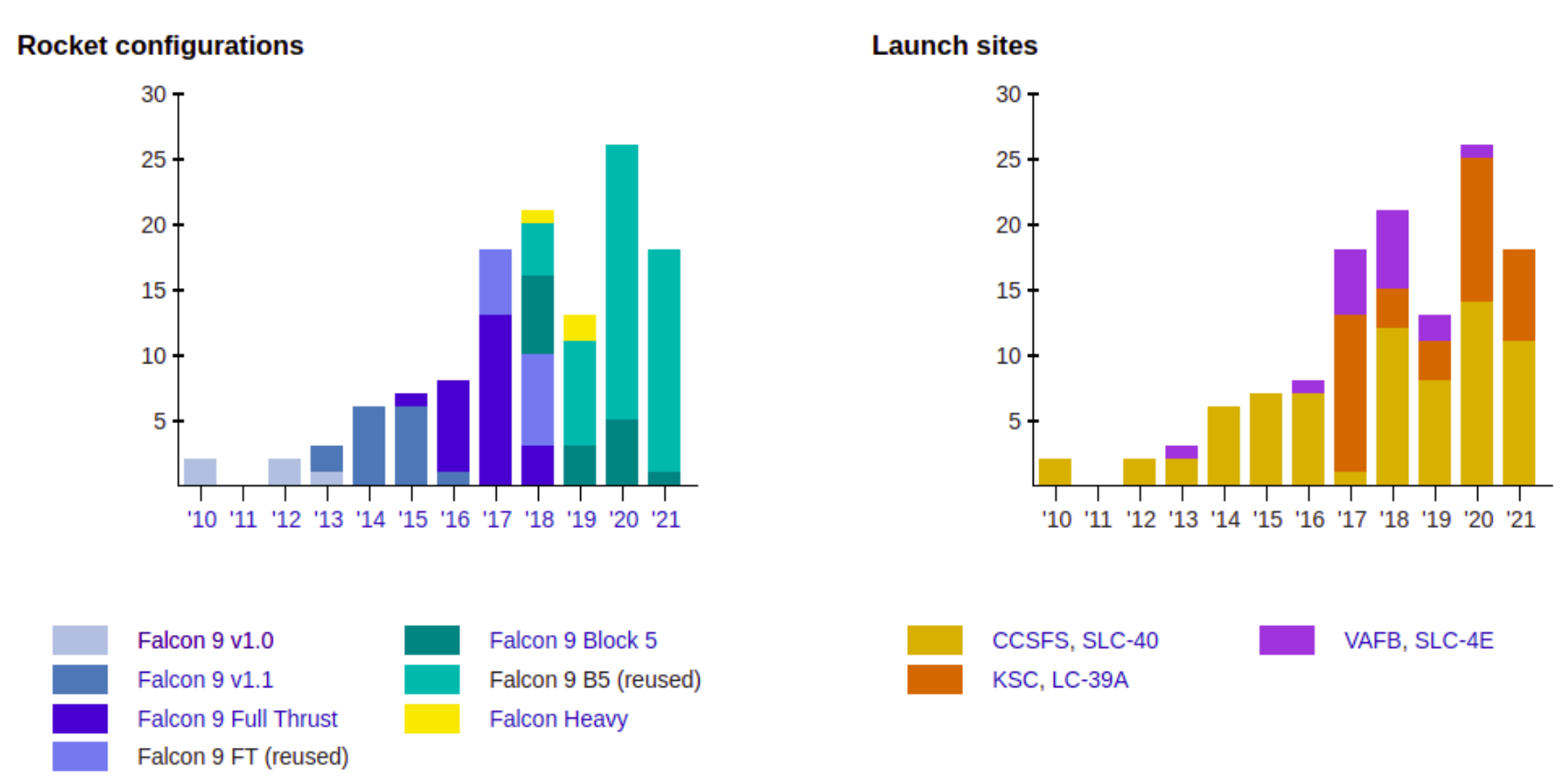
Git Hub link:

Conclusions

- The success of the launches is highly dependent upon the site and the orbit to which the rocket is directed.
- To evaluate in a more rigorous way how great is a site in terms of successful launches is important to collect much more information since by now with the available data is impossible for some sites draw conclusions.
- In general, as the Flight number increases the amount of successful missions increased due to the experienced obtained in the previous launches and also because the efforts were redirected to sites where the probability of success was high.
- The Payload mass is a key factor in the success of missions to support the previous conclusion. With the pass of time this component was increased and the success rate increased.

Appendix (1)

Rockets configuration and launch sites



Appendix (2)

Code to create the Payload range and show the ranges in the plot

```
@app.callback(Output(component_id='success-payload-scatter-chart',
                     component_property='figure'),
              [Input(component_id='site-dropdown', component_property='value'),
               Input(component_id='payload-slider', component_property='value')])
def get_scatter_chart(input_site, pay_load):
    print(pay_load)
    if input_site == "ALL":
        fig = px.scatter(spacex_df, x="Payload Mass (kg)",
                         y="class",
                         color="Booster Version Category"
                         )
        return fig
    else:
        filtered_df = spacex_df[(spacex_df["Launch Site"] == input_site) &
                                (spacex_df["Payload Mass (kg)"] >= pay_load[0]) &
                                (spacex_df["Payload Mass (kg)"] <= pay_load[1])]
        print(filtered_df)

        fig = px.scatter(filtered_df, x="Payload Mass (kg)",
                         y="class",
                         color="Booster Version Category"
                         )

        return fig
```

Thank you!