IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Johan Forslund
2024-06-12

Outline

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

Executive Summary

We have performed Data Science Analysis following normal methodology steps to analyze launch data to answer the business question: "**What is the probability that the first segment of the Falcon 9 rocket will be able to land correctly after a launch and therefore be able to be reused.**"

- Data collection using SpaceX API REST-APIs
- Data collection using web Scraping techniques getting data from SpaceX website
- Data Wrangling to and cleaning to get the data prepared for analysis. Data is then stored in a SQL-database
- Exploratory Data Analysis using SQL
- Exploratory Data Analysis using Pandas, Matplotlib and Plotly Dash
- Exploratory Data Analysis of launch sites using Folium maps
- Create and Evaluate Machine Learning prediction model

Executive Summary

Summary of all results

- Best algorithm to use for predicting outcome of launches is the Decision Tree Algorithm, with 87,5 % success rate.
- Success rate has in general increased with the number of launches, which is normal due to that problems are fixed, and the organization are learning.
- Success rate seem to be higher for launches with low Payload Mass, than for higher Payload mass.
- Launch site KSX-LC-39A has the most successful launches.

Introduction

SpaceY is a new tough competitor to SpaceX. Its is owned by the multi billionaire Allon Mask. Allon has created a task force that will try and bring order to the costs that SpaceX has today for their launches, by creating a data model that can estimate the probability that a lauch will be a success and the first stage of the rocket safely will return to Earth to be reused in a coming launch.

The business questions that the data science group must answer is:

- "**What is the probability that the first segment of the Falcon 9 rocket will be able to land correctly after a launch and therefore be able to be reused.**"

This information will then be used by the finance department when a bidding war against SpaceX occurs, as we will better understand the costs SpaceX has for their operations.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was collected by accessing public SpaceX REST APIs and scraping data from their public web site. Python was used as only programming language.
- Perform data wrangling
  - Describe how data was processed
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection

Data introduction
SpaceX has presented a lot of information on their web site and on public APIs. All information used in this analysis is public information that has been found on their public web site and API REST-services. We will be using the following types of information all available in REST APIs: **Rocket data, Launch site data, Payload data and Launch data**.

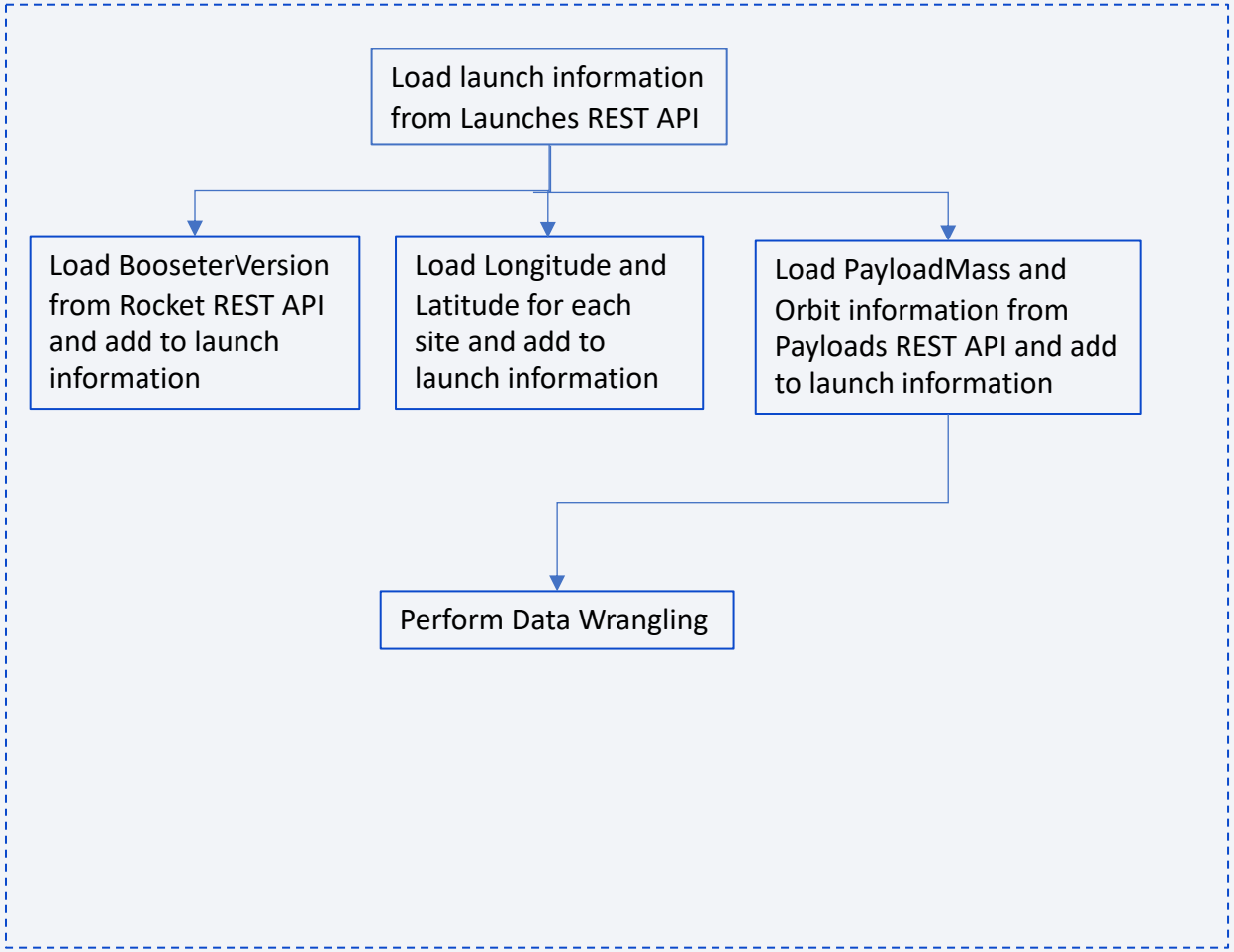
Data was imported using python scripts.
The **requests** library was used to import data in json format from the REST APIs.
For the web scraping, getting data from web pages, the **BeautifulSoup** library was used to access the HTML-code.

Data Collection – SpaceX API

Summary of the data flow for importing the Launch Information from REST APIs

The full code can be found here in a Jupyter Notebook presentation:

[01.jupyter-labs-spacex-data-collection-api.ipynb]



Data Collection - Scraping

Other way to find the needed data,than from a REST API, is to collect data from a web page, a.k.a. web scraping. The picture illustrates the high level steps done to try this technique. Python scripts has been used.

The detailed steps can be found in a Jupyter Notebook that you find here:

[02.jupyter-labs-webscraping.ipynb]

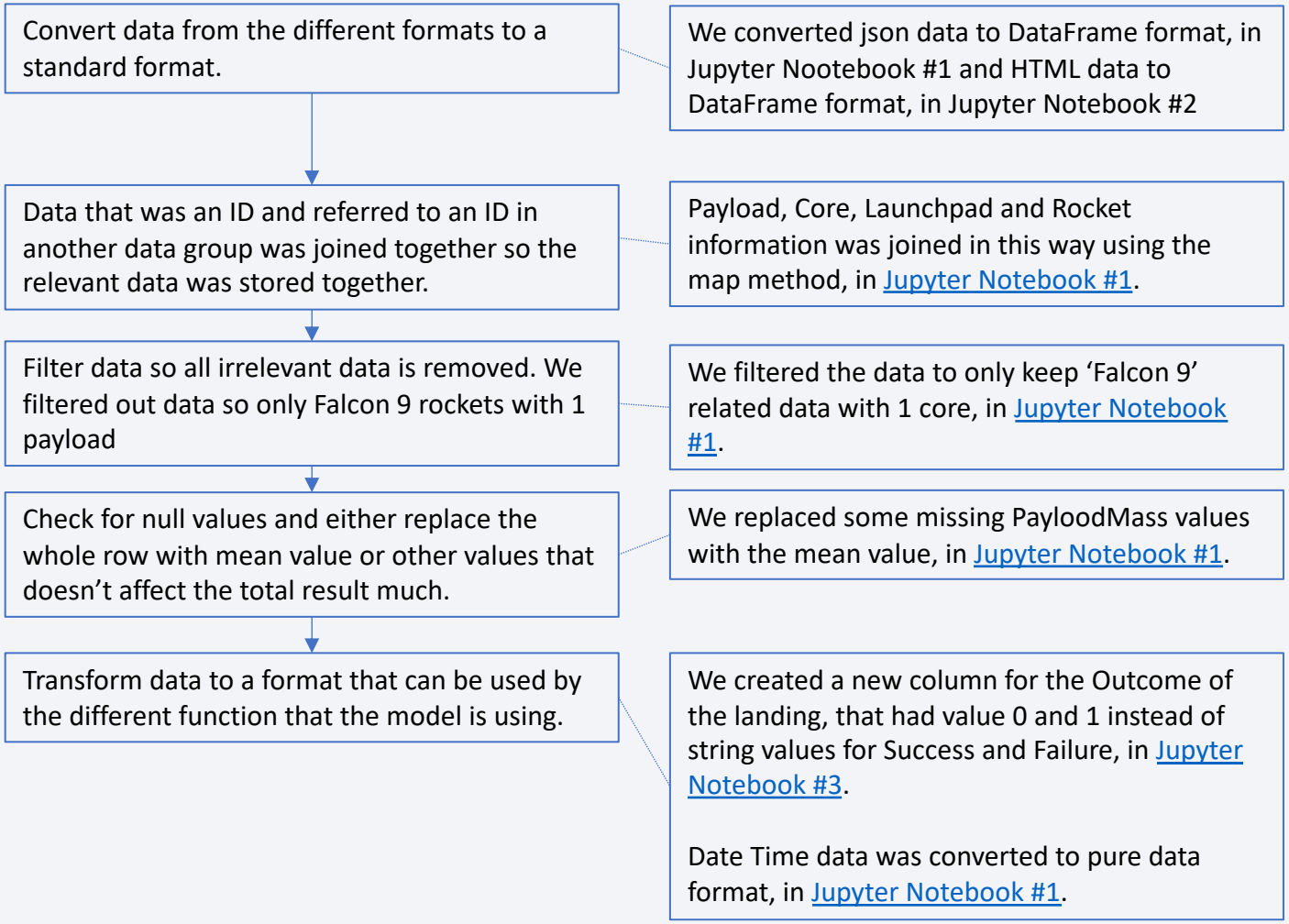
Data Wrangling

Data Wrangling is the step after data has been imported. It means that the data is cleaned and fixed-up so it is possible to use it in the models.

Data Wrangling is done by first examining the data manually in tables and also in graphs to find out what needs to be done with it.

The steps done to prepare our data are listed in high level in the flowchart. The implementation is found in the referenced Jupyter Notebooks in github.

Data Wrangling continued



EDA with Data Visualization – Scatter Plots

To be able to perform Exploratory Data Analysis we created a number of scatter graphs to put 2 or3 attributes against each other to enable finding a relationship by visually exploring the graphs. This is an important step in the process of understanding the data before starting to build a machine learning model for prediction. The following attributes were analyzed in graphs. The results are presented in detail later in this document.

- “Flight Number vs Launch Site” graph shows how successful and failed launches are divided by the different sites and on what order it was launched
- “Payload Mass vs Launch Site” graph shows how successful and failed launches are divided by the different sites and the Payload Mass.
- “Flight Number vs Orbit Type ” graph shows how successful and failed launches are divided by the Flight Number and the orbit in space the Payload are deployed for.
- “Payload Mass vs Orbit Type” graph shows how successful and failed launches are divided by the Payload Mass and the orbit in space the payload are deployed for.

Each graph can also be found in the Lab Notebook, 05.jupyter-labs-eda-dataviz.ipynb

EDA with Data Visualization – Bar Chart / Line graphs

The following attribute were analyzed in Bar Chart. The results are presented in detail later in this document.

- “Average Success rate vs Orbit type” graph shows how bars for each Orbit Type where the height is the average success rate.

The following attribute were analyzed in Line Chart. The results are presented in detail later in this document.

- “Average success rate per Year” graph shows how the average success rate fluctuates over time.

Each graph can also be found in the Lab Notebook, 05.jupyter-labs-eda-dataviz.ipynb

EDA with SQL

A number of SQL-commands have been run make it possible to analyze the following information about launches:

1. Show all Launch Sites
2. Show information about the first 5 lauches from a Launch Site starting with ‘CCA’
3. Show total Payload Mass that has SpaceX has delivered for NASA
4. Show average Payload Mass that has been carried by Booster Version F9 v.1.1.
5. Show date for the first successful landing in ground pad
6. Show all Booster_Versions that has been carrying a Payload Mass between 4000 and 6000
7. Show the number of successful outcomes and failed outcomes
8. Show the names of the booster_versions which have carried the maximum payload mass
9. Show Month Name, Landing Outcome, Booster Version and Launch Site information about the failed landings in drone shop during 2015
10. Show the count Launches per unique landing outcome between 2010-06-04 and 2017-03-20, in descending order. Sort on count in descending order.

The result from the analysis can bee seen later in this document.
The real SQL-code can also be seen here in this Lab-notebook: 04.jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

Folium is a python tool for displaying geographical maps with different information that can be added. We have used the following objects:

- Circles mark the position of each Launch Site, to make it clear where the sites we are working with are located.
- Markers show the name of each Launch Site
- Marker Clusters show the number of all individual Launches on the sites and the success/fail status for each individual Launch, to make it easy to get a big picture on how much the Lauch Sites are used.
- Lines and Markers witch distance information was added to show the distances between the Launch Sites and different objects of interest, like nearest city, cost line, railroad larger road whicj is of interests to find out what is important when selecting a location for a Launch Site.

The Folium maps can be found in this Lab-Notebook: 06.lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

Dash is a web-based dashboard tool build on the Plotly library for Python. It makes it possible to create smart dashboards that can be used to examine data in a flexible way using the web.

- We have added a Pie Chart to examine Launch status in total or per Site, by selecting values in a dropdown list.
- We also added a Scatter chart for Launch status and Payload Mass. The graph can be controlled to show all Payload Mass values or a range by using a Range Slider.

The Plotly Dash code for the dashboard can be found here:
spacex_dash_app.py

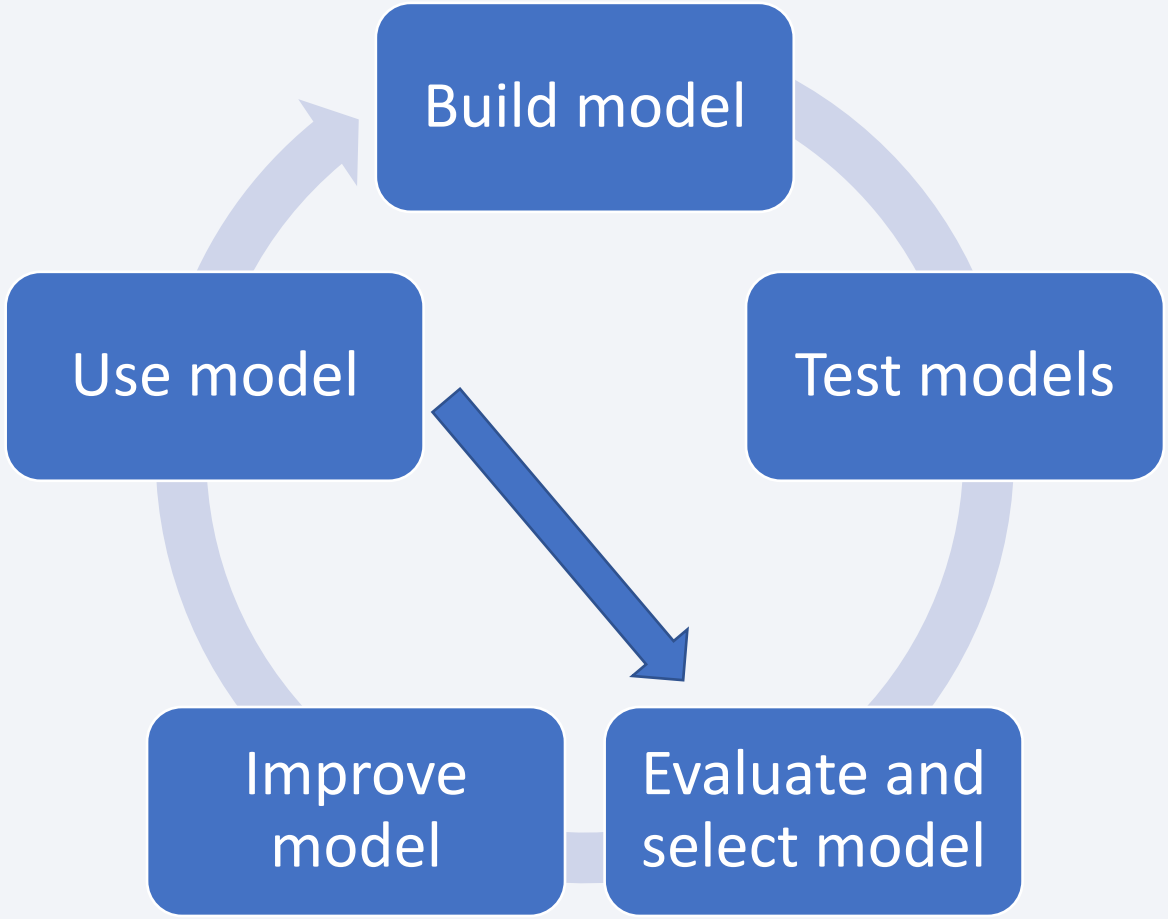
Predictive Analysis (Classification) (1/3)

We now start create a prediction model that will estimate the future outcomes of rocket launches.
We do this in an iterative way building, testing, evaluating, improving using and evaluating/selecting and improving the model.

A model performance can change over time due to new factors coming in so its important to monitor the model in production and do improvements when needed.

Detailed info about the result is presented later in the document.

The code can be found in this Lab-Notebook: [07.SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb]



Predictive Analysis (Classification) (2/3)

Building model

- Load data using Pandas and Numpy libraries in Python
- Create input data series X and output data series Y.
- Normalize the data (making the numeric values be in the same ranges so they affects the predictions evenly)
- Split data into a training set (X-train, Y-train) that we will train the model on, and a testing set (X-test, Y-test) that we then will run the model on to evaluate its performance.
- When we build the model we do it using GridSearchCV objects that allows us to run a model for a specific algorithm
  with different values on the parameters, to find the best parameters for that model.
- We prepare different version of the model for different algorithms also, to find out what algorithm is best suited and gives the most accurate results. The algorithms tested are: Logistic Regression, Support Vector Machine, Decision Tree Classification and K-nearest Neighbors

Testing model

- We run the model with different parameters and for different algorithms so we

Predictive Analysis (Classification) (3/3)

Evaluation and select model

- Evaluating the model requires that we go through the result from each of the test runs and evaluate what parameter is best for each algorithm and then compare the best results for each algorithm and selects the algorithm that berforms best and has the highest accuracy.
- The results from the test runs are documents in Confusion Matrixes that show how good the model performs

Improve model

- Based on real results from using the model we can tweak the model and try to optimize it.

Use model

- We use the model and do real prediction
- The result is monitored and evaluated and when needed we try to improve it.

Predictive Analysis (Classification)

- Summarize how you built, evaluated, improved, and found the best performing classification model

Next step in the the process is to start build a model that can be used to predict future outcomes launches based on the different attributes.

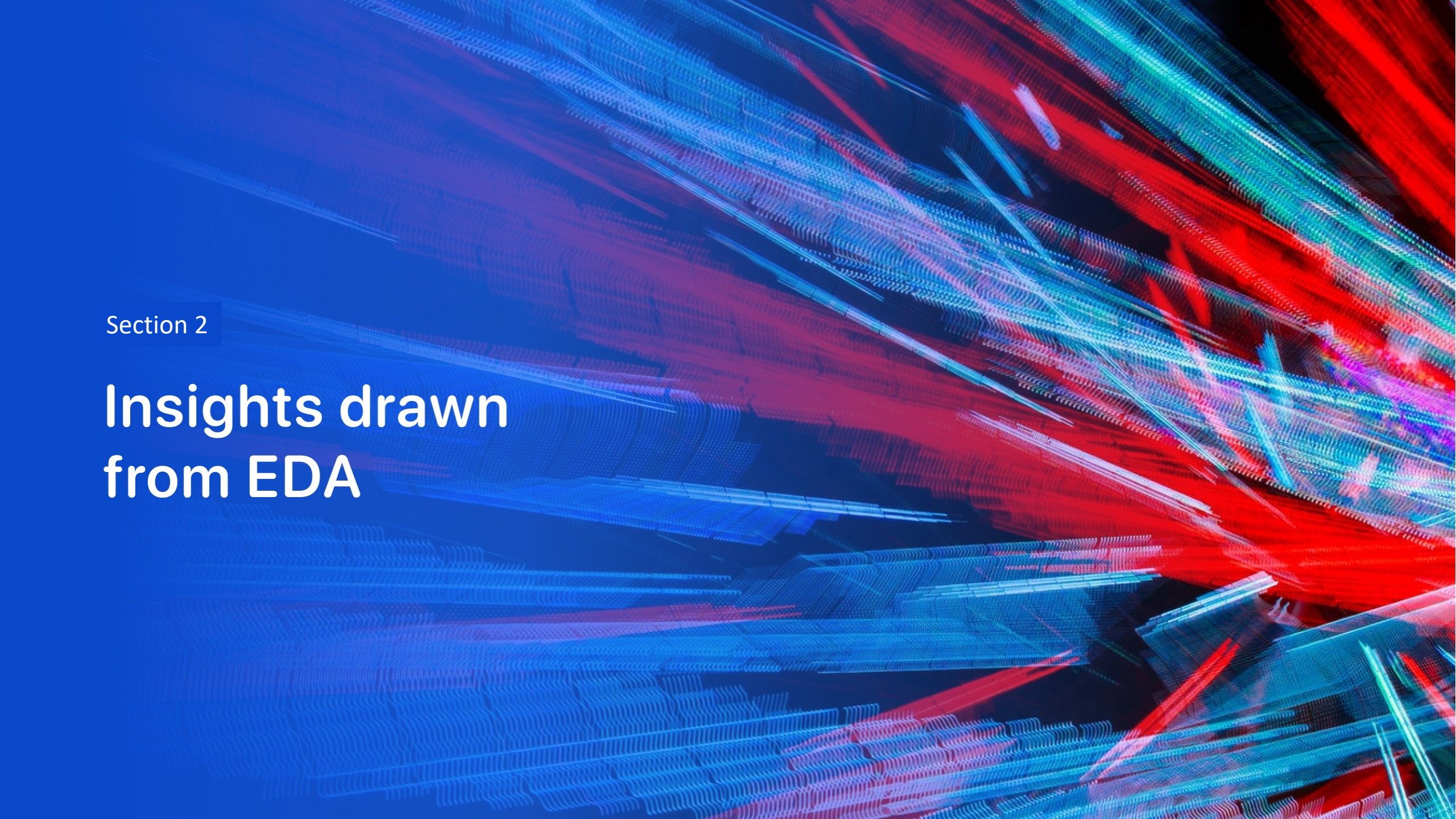
Building phase

- You need present your model development process using key phrases and flowchart
- Add the GitHub URL of your completed predictive analysis lab, as an external reference and peer-review purpose

Results

The findings and conclusions will be presented for each type of analysis:

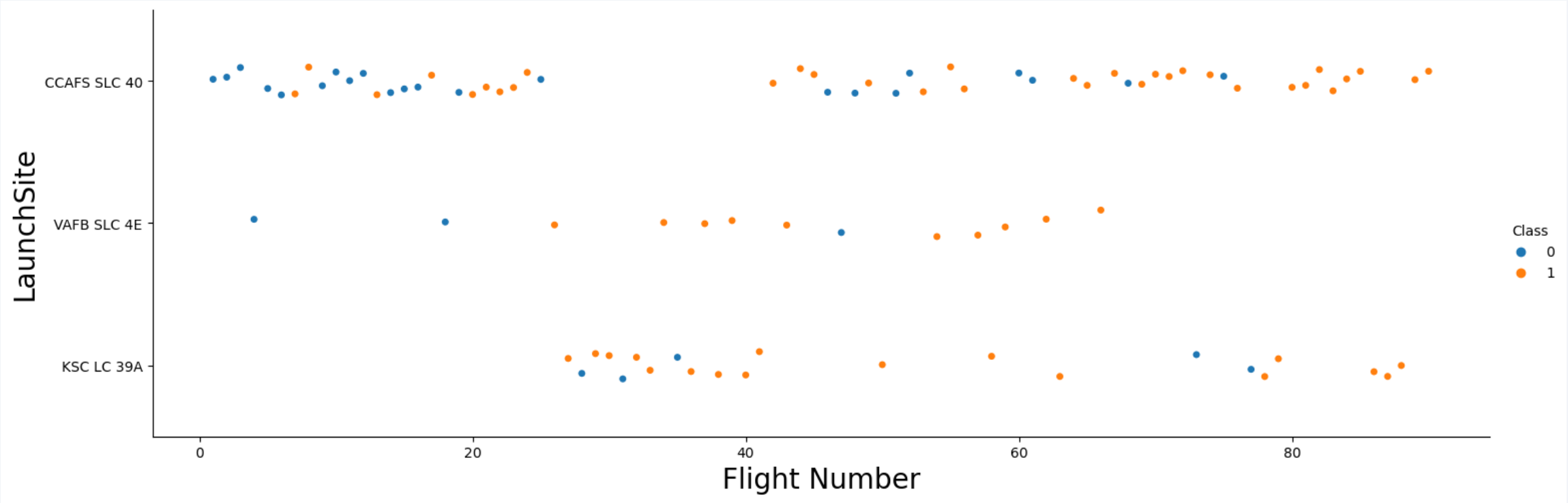
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results


Section 2
Insights drawn from EDA

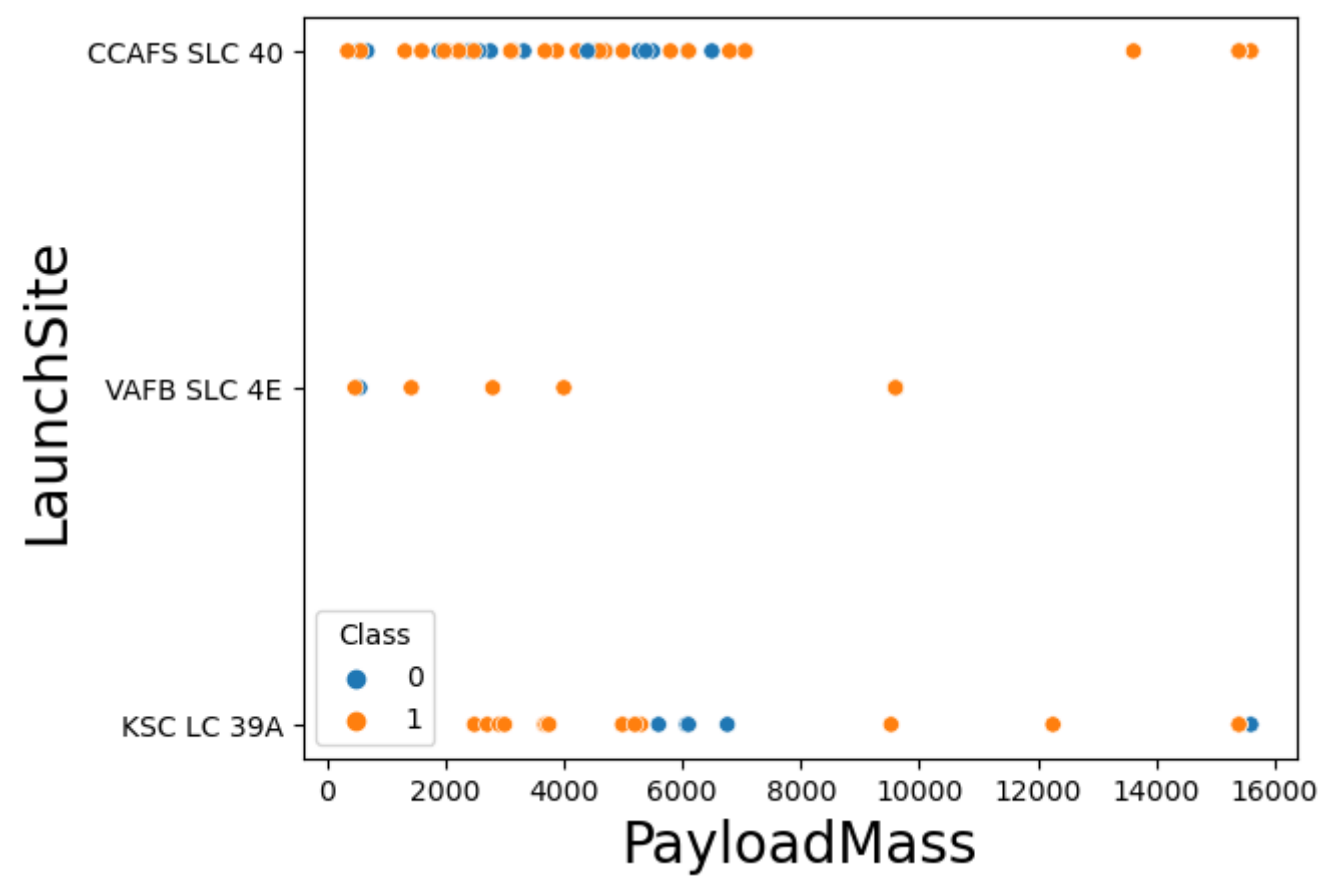
Flight Number vs. Launch Site

Here we see a cat plot graph displaying launch success / failure per site and flight number.

We can see that the number of successes increases drastically the more flights are launched, due to that the problems are solved.



Payload vs. Launch Site



Here we see a scatter plot showing success and failure for launches per Launch site and Payload Mass.

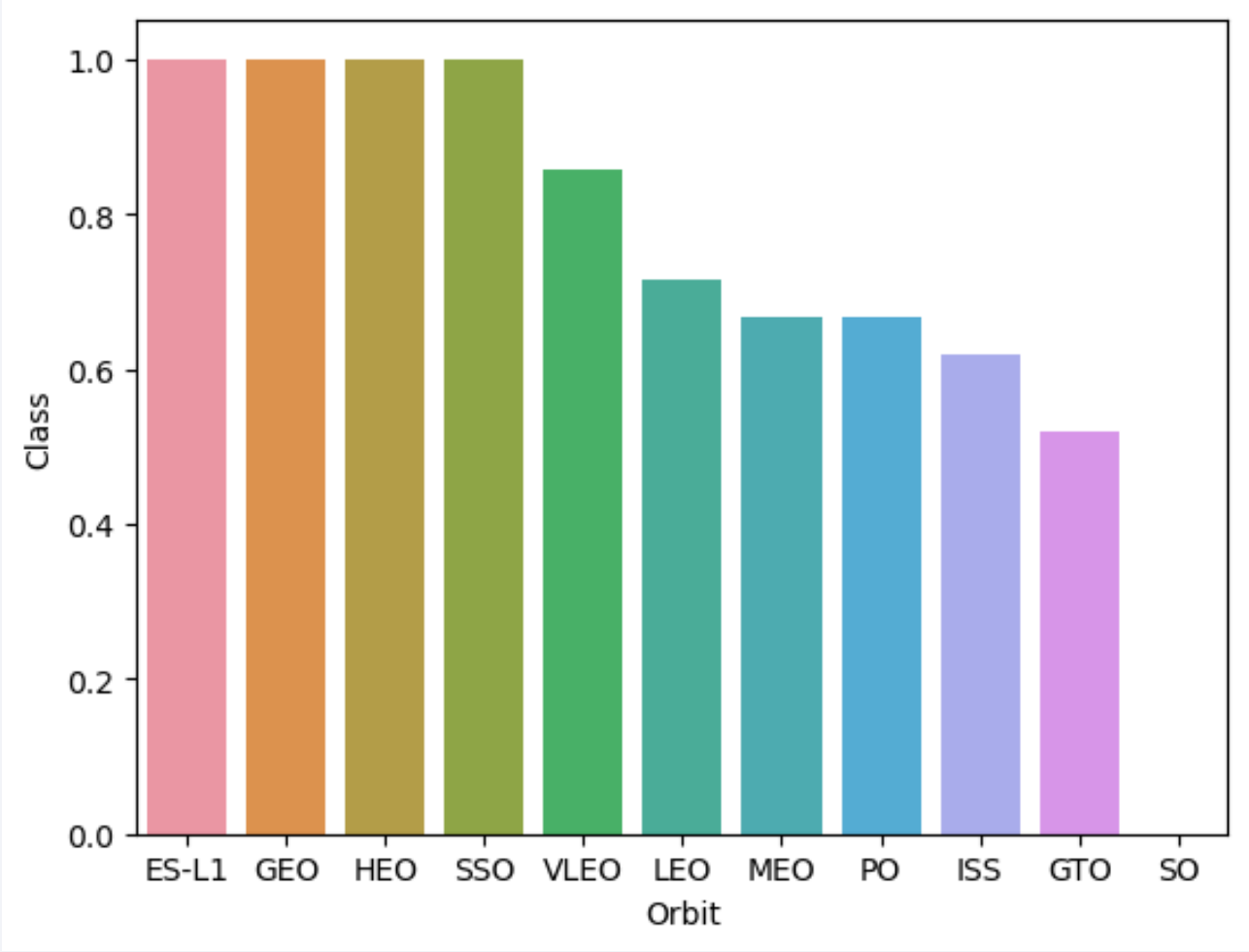
VAFB SLC 4E launch site has not sent up and payloads over 10 000.

Success rate seem to be higher for low Payload Masses also which make sense. This should be looked at more to find out if there is a correlation or not when more flights have been done.

Success Rate vs. Orbit Type

This bar chart shows the mean success rate per Orbit Type

- We see that flight with Orbit Type SO have never succeeded.
- ES-L1 , GEO, HEO and SSO has 100 % success rate.

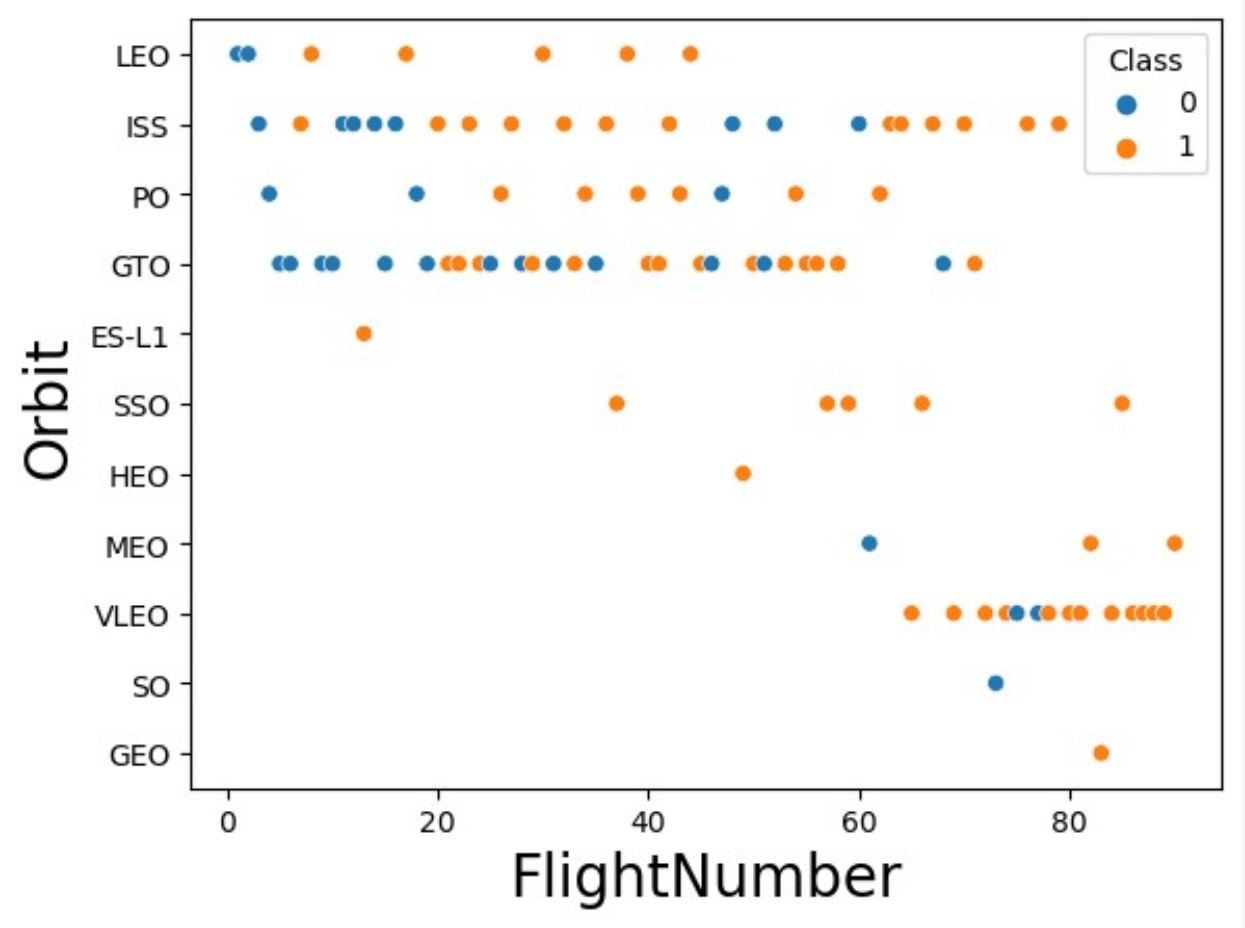


Flight Number vs. Orbit Type

Here we see a scatter plot showing success and failure for launches per Orbit Type site and Flight Number.

- Can see that for LEO orbit type the latest launches has all succeeded after first failing.
- For ES-L1, SSO, GEO and MEO orbit types all launches has succeeded

Success rate improves by flight number.



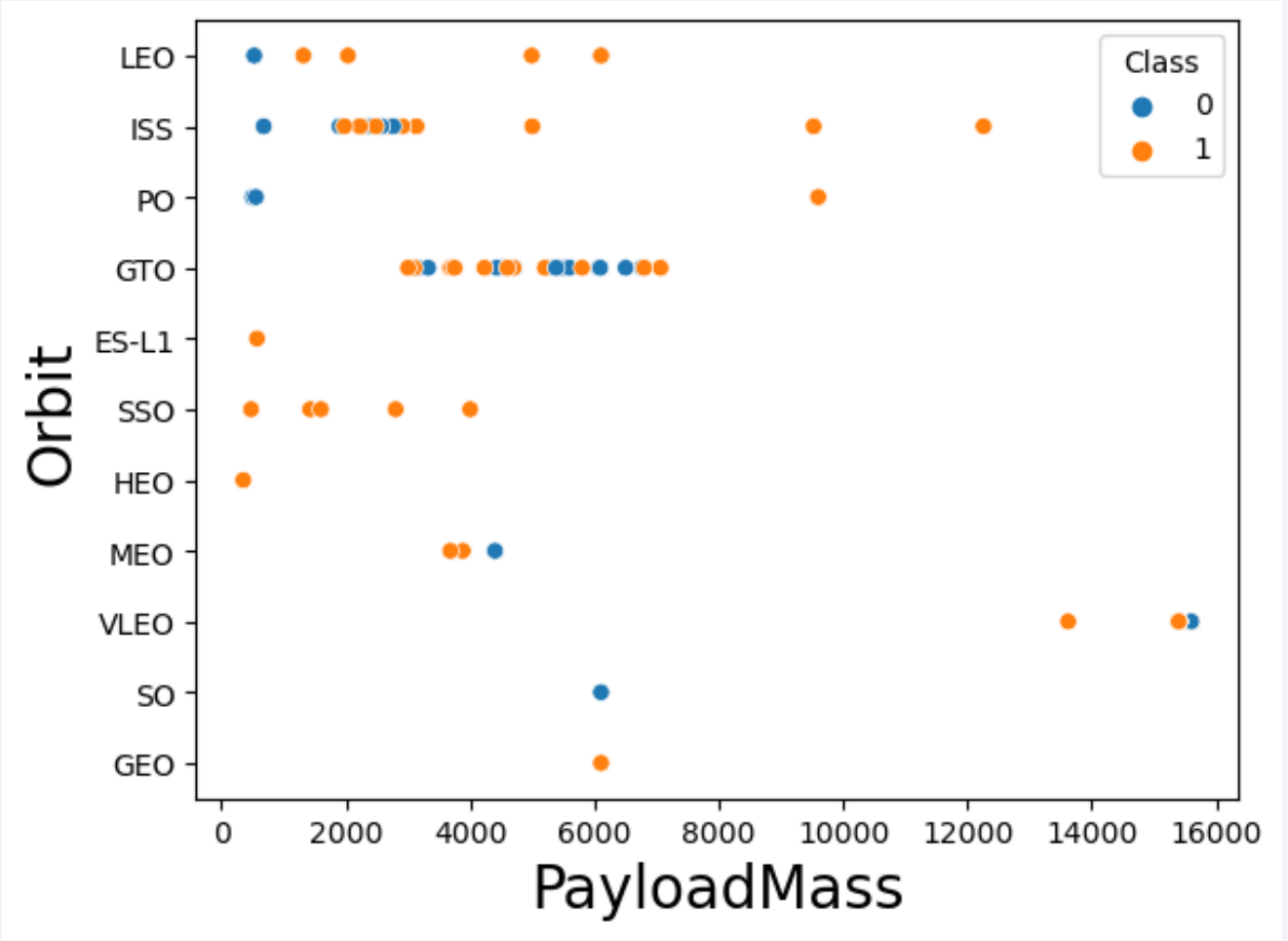
Payload vs. Orbit Type

Here we see a scatter plot showing success and failure for launches per Orbit Type site and Payload Mass.

LEO, ISS and PO all goth failed launches for their lowest Payloads.

GTO has got a mixed results. No high payloads aer sent there.

ES-L1, SSO, HEO, GEO has all successful launches.

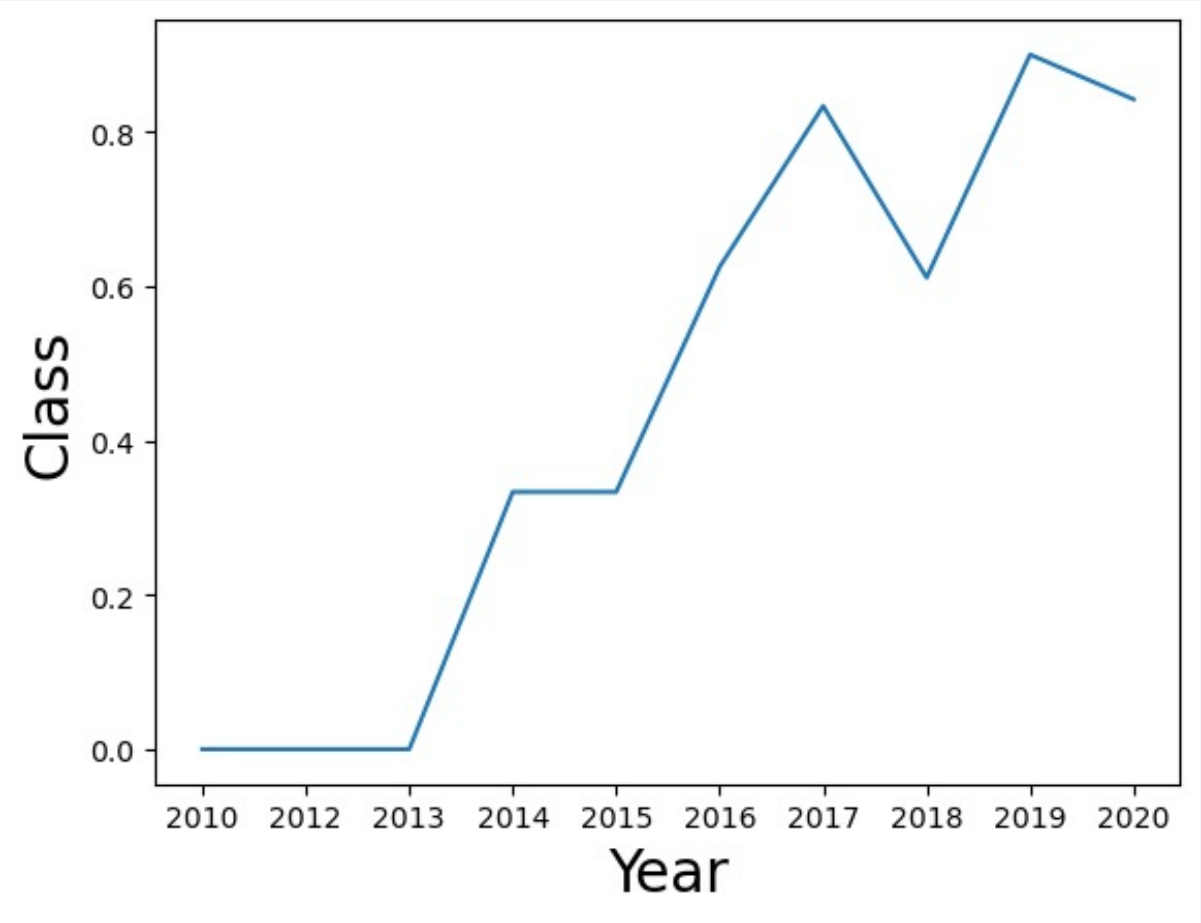


Launch Success Yearly Trend

Here we see a line graph showing how average success rate changes.

The conclusion is that after the first 3 years where no lauch succeeded it has gone better and better up to 2018 where we had a dip by 20%.

What happened 2018? Check other variables!



All Launch Site Names

SpaceX is using the following launch sites

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

How did we find this information?

By using the SQL-command DISTINCT against the database we created, containing all the launch information, it is possible to find the unique list of launch sites used by SpaceX.

> SELECT DISTINCT Launch_Site FROM SPACEXTABLE

Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with `CCA`

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	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	0:35:00	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

How did we find this information?

By using the SQL-command LIKE against the database we created, containing all the launch information, it is possible to find data that begins with specific characters. The LIMIT 5 command limits the number of rows.

> SELECT * FROM SPACEXTABLE WHERE Launch_Site like 'CCA%' LIMIT 5

Total Payload Mass

- Calculate the total payload carried by boosters from NASA

SpaceX has launched 107 010 kg of payload into space

This is about 17% of the total payload mass delivered.

How did we find this information?

By using the SQL-command SUM against the database we created, containing all the launch information, it is possible to summarize numeric values like the payload mass. With the WHERE command we limited the information to the NASA customer. To make sure that we found all customers containing NASA we also converted the name to Upper Case.

```
> SELECT SUM(PAYLOAD_MASS__KG_) AS TOTAL_PAYLOAD_MASS FROM SPACEXTABLE
WHERE UPPER(Customer) like '%NASA%'
```

Average Payload Mass by F9 v1.1

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

The average payload mass per launch is 2 928,4 kg

How did we find this information?

By using the SQL-command AVG it is possible to get the average value from all rows. With the WHERE command we limited the data to launches made with Booster ‘F9 v1.1’

```
> SELECT AVG(PAYLOAD_MASS__KG_) AS AVERAGE_PAYLOAD_MASS
FROM SPACEXTABLE WHERE Booster_Version = 'F9 v1.1'
```

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad

The first time a successful landing was done in a ground pad was 2015-12-22

How did we find this information?

By using the SQL-command MIN it is possible to get the lowest value value from all rows. With the WHERE command we limited the data to launches where the landing was done in a ground pad.

> SELECT MIN(Date) FROM SPACEXTABLE WHERE Landing_Outcome like '%ground pad%'

Successful Drone Ship Landing with Payload between 4000 and 6000

- List the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000

How did we find this information?

By using the SQL-command BETWEEN is it possible to limit the Payload Mass to a specific range. Notice that the BETWEEN command includes the ending values given. The DISTINCT command returned all unique Booster Versions from the matching launches. By using WHERE command we could limit the data to only successful launches.

> %sql SELECT DISTINCT (Boster_Version) FROM SPACEXTABLE WHERE Mission_Outcome = 'Success' AND PAYLOAD_MASS__KG_ BETWEEN 4001 AND 5999 ORDER BY Booster_Version

Booster_Version
F9 B4 B1040.2
F9 B4 B1040.1
F9 B5 B1046.2
F9 B5 B1047.2
F9 B5 B1048.3
F9 B5 B1051.2
F9 B5 B1058.2
F9 B5B1060.1
F9 B5B1062.1
F9 FT B1021.2
F9 FT B1031.2
F9 FT B1032.2
F9 FT B1020
F9 FT B1022
F9 FT B1026
F9 FT B1030
F9 FT B1032.1
F9 v1.1
F9 v1.1 B1011
F9 v1.1 B1014
F9 v1.1 B1016

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes

SUCCESSFUL_OUTCOMES	FAILED_OUTCOMES
98	3

How did we find this information?

We used COUNT(*) to count all successful outcomes and failure outcomes. To make it more interesting we used two subqueries to return the result to another query so we got the return values in two columns.

```
> SELECT * FROM (SELECT COUNT(*) AS SUCCESSFUL_OUTCOMES FROM
SPACEXTABLE WHERE Mission_Outcome = 'Success'), (SELECT COUNT(*) AS
FAILED_OUTCOMES FROM SPACEXTABLE WHERE Mission_Outcome <> 'Success')
```

Boosters Carried Maximum Payload

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

How did we find this information?

We could use a sub-query in the WHERE clause to return the highest payload mass and then use a main quesry to get the Booster_Versions that matches this payload mass.

> SELECT Booster_Version FROM SPACEXTABLE WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE) ORDER BY Booster_Version

Booster_Version
F9 B5 B1048.4
F9 B5 B1048.5
F9 B5 B1049.4
F9 B5 B1049.5
F9 B5 B1049.7
F9 B5 B1051.3
F9 B5 B1051.4
F9 B5 B1051.6
F9 B5 B1056.4
F9 B5 B1058.3
F9 B5 B1060.2
F9 B5 B1060.3

2015 Launch Records

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

How did we find this information?

We used CASE WHEN END command to substitute the month number with the month name, and then limited the launches by using WHERE on the year part of the date and the Landing_Outcome to get failed launches in drone_ship.

```
%% sql SELECT CASE WHEN (substr(Date, 6,2)) == '01' THEN "Januari"
                WHEN (substr(Date, 6,2)) == '02' THEN "February"
                WHEN (substr(Date, 6,2)) == '03' THEN "Mars"
                WHEN (substr(Date, 6,2)) == '04' THEN "April"
                WHEN (substr(Date, 6,2)) == '05' THEN "May"
                WHEN (substr(Date, 6,2)) == '06' THEN "June"
                WHEN (substr(Date, 6,2)) == '07' THEN "July"
                WHEN (substr(Date, 6,2)) == '08' THEN "August"
                WHEN (substr(Date, 6,2)) == '09' THEN "September"
                WHEN (substr(Date, 6,2)) == '10' THEN "October"
                WHEN (substr(Date, 6,2)) == '11' THEN "November"
                WHEN (substr(Date, 6,2)) == '12' THEN "December"
END AS Month,Landing_Outcome,
Booster_Version,Launch_Site FROM SPACEXTABLE WHERE substr(Date,0,5) = '2015' AND Landing_Outcome like 'Failure (drone ship)'
```

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

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

How did we find this information?

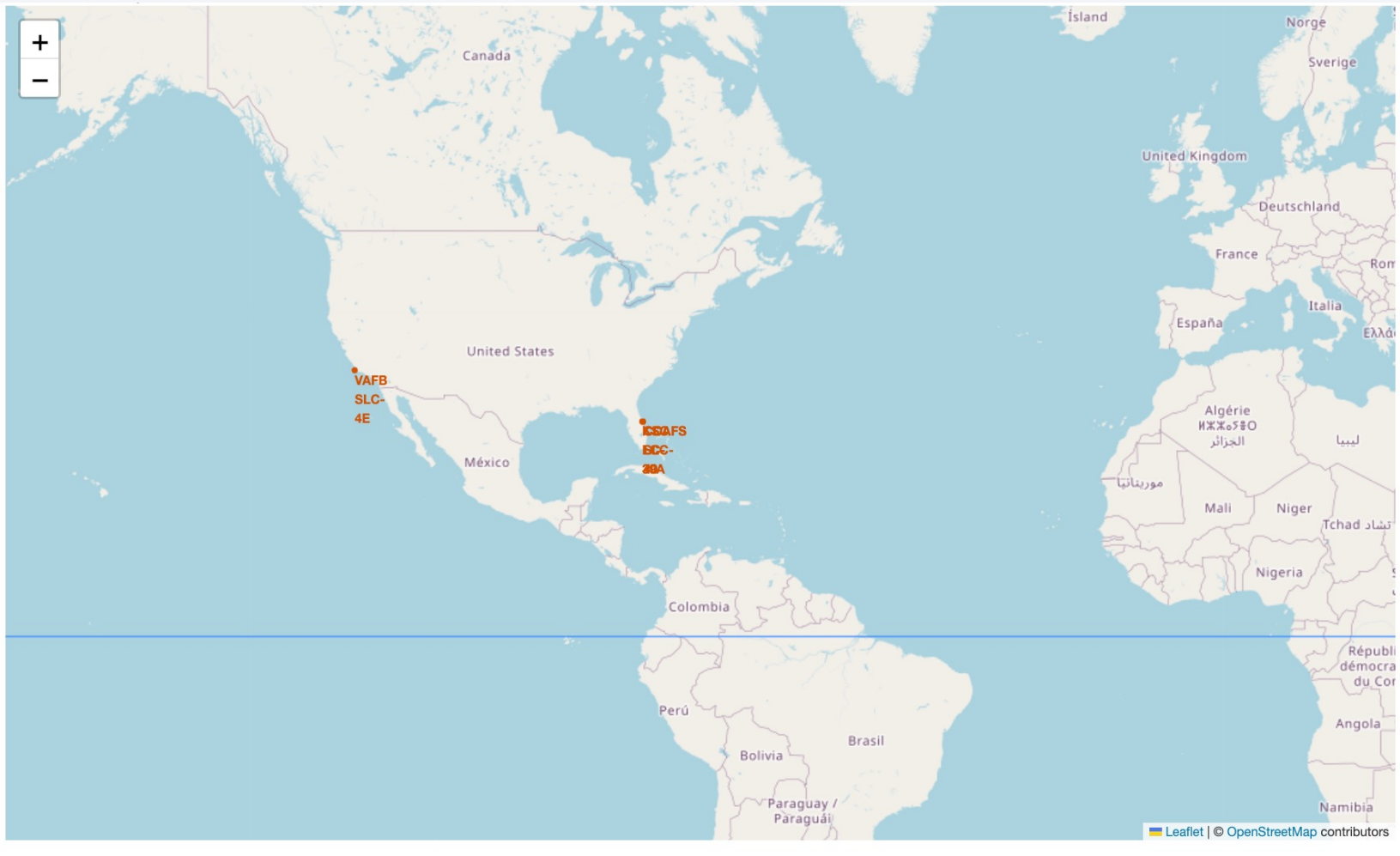
We used GROUP and COUNT command to get the list of unique landing outcomes, and then used ORDER DESC to sort the list in descending order on the count value

> SELECT Landing_Outcome, COUNT(Landing_Outcome) as LANDING_OUTCOME_COUNT FROM SPACEXTABLE GROUP BY Landing_Outcome ORDER BY LANDING_OUTCOME_COUNT DESC

Landing_Outcome	LANDING_OUTCOME_COUNT
Success	38
No attempt	21
Success (drone ship)	14
Success (ground pad)	9
Failure (drone ship)	5
Controlled (ocean)	5
Failure	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1
No attempt	1


Section 3
Launch Sites
Proximities Analysis

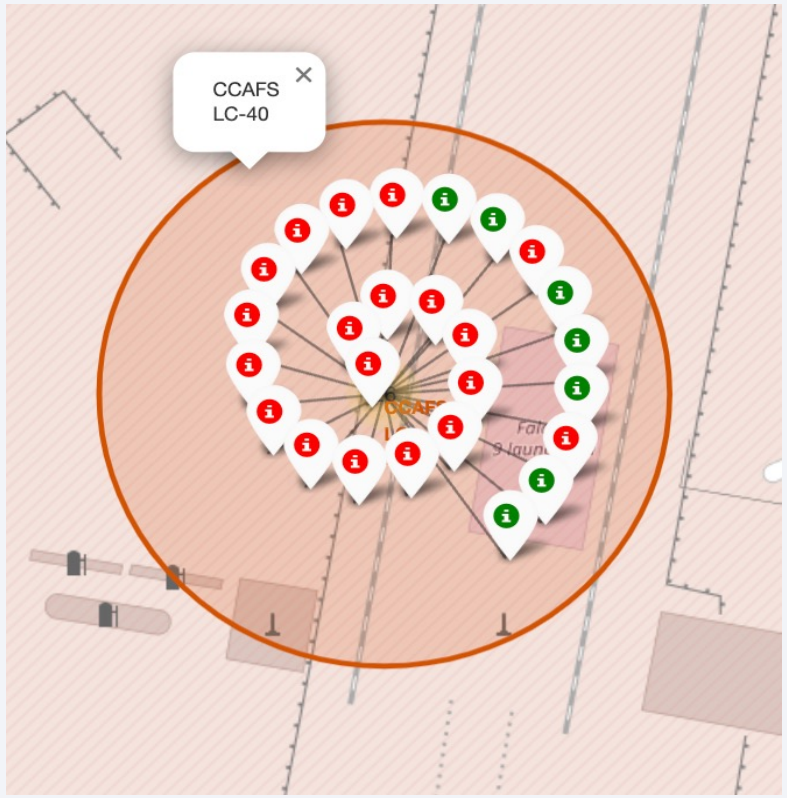
Launch site locations

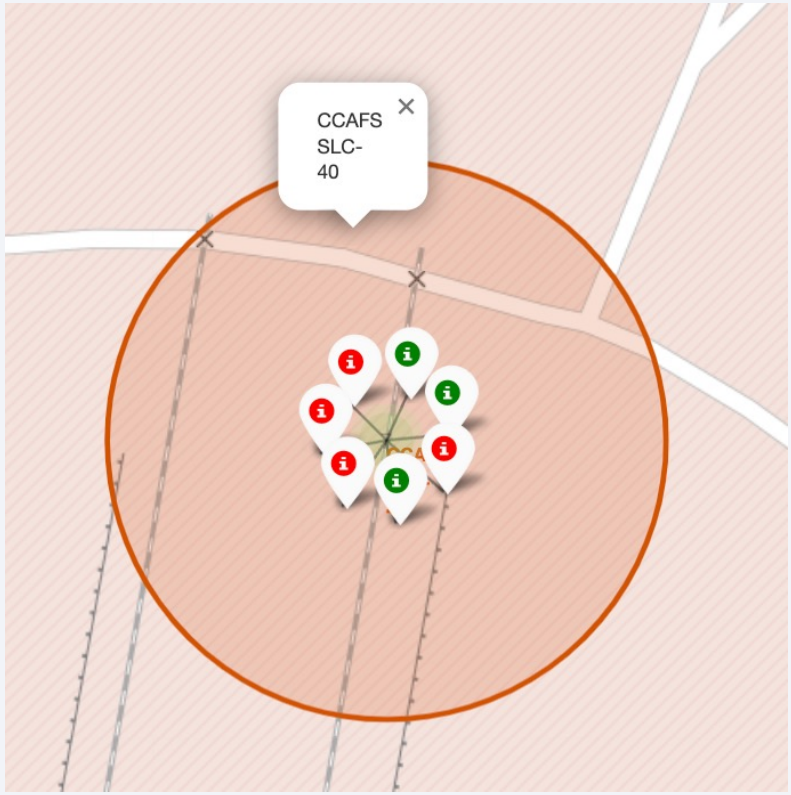


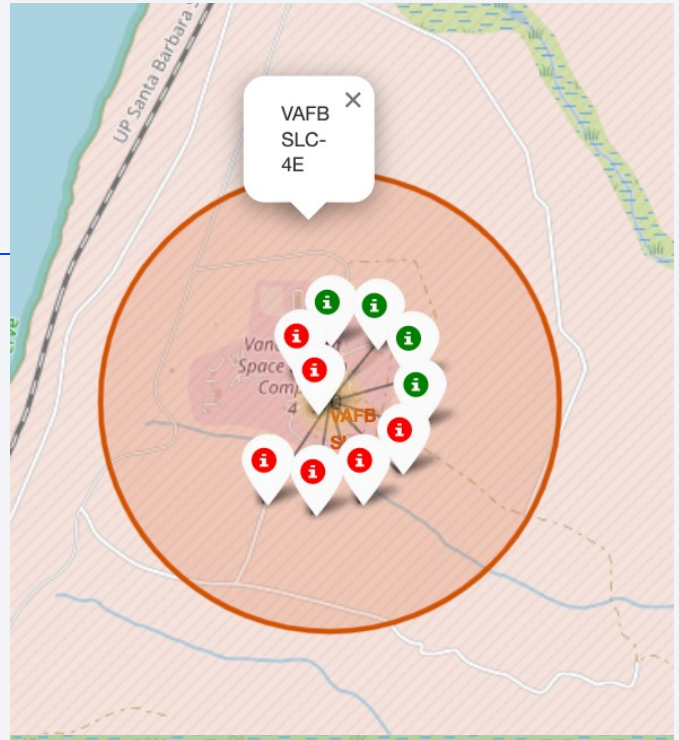
- To minimize the risk are the launch sites placed near large waters or areas without populations to the east.
- Launces are done to the east to take advantage of the boost from the earth rotation.
- All sites are also located more to the equator than to the north side to take use of the extra boost from the earth rotation, as it is highest at the equator.

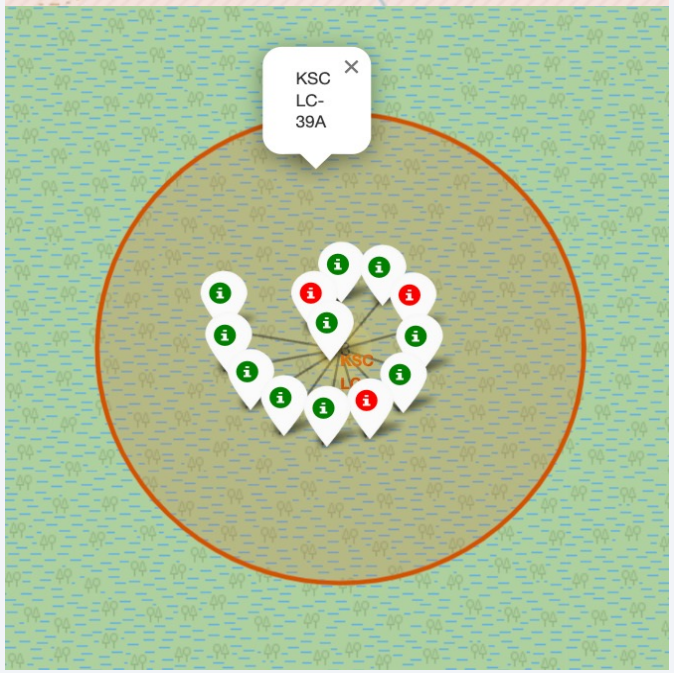
Launch outcomes

Pictures show status of launches done from the different launch sites. Green means that the first section of the rocket landed correctly and could be reused. Red means that something vent wring and the first section could not be used again.









Launch outcomes (continued)

Pictures show number of launches done from the different launch sites.
The oldest launches are in the inner part of the swirl.

The graphs shows the following:

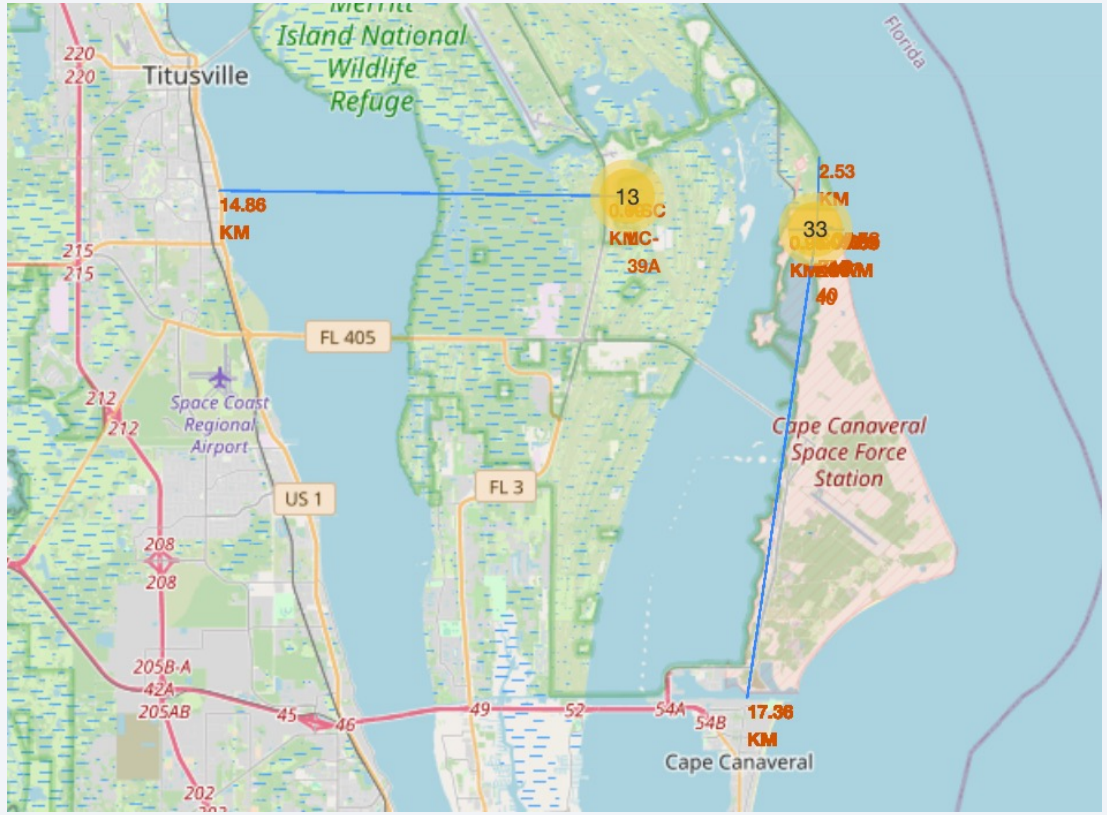
- **CCAFS LC-40** : 7 successful launches and 19 failed. Launch status increased over time from 0 % success rate to 7 out of 9 success (78%). Totally 7/26 = 27 %, success rate.
- **CCAFS SLC-40** : 3 success and 4 failed (7 total). To few launches and to mixed up results to draw any conclusions.
- KSC LC-39A : 13 successful launches and 3 failed (16 total). The 5 latest launches succeeded
- VAFB SLC-4E : 4 successful launches and 6 failed (10 total)

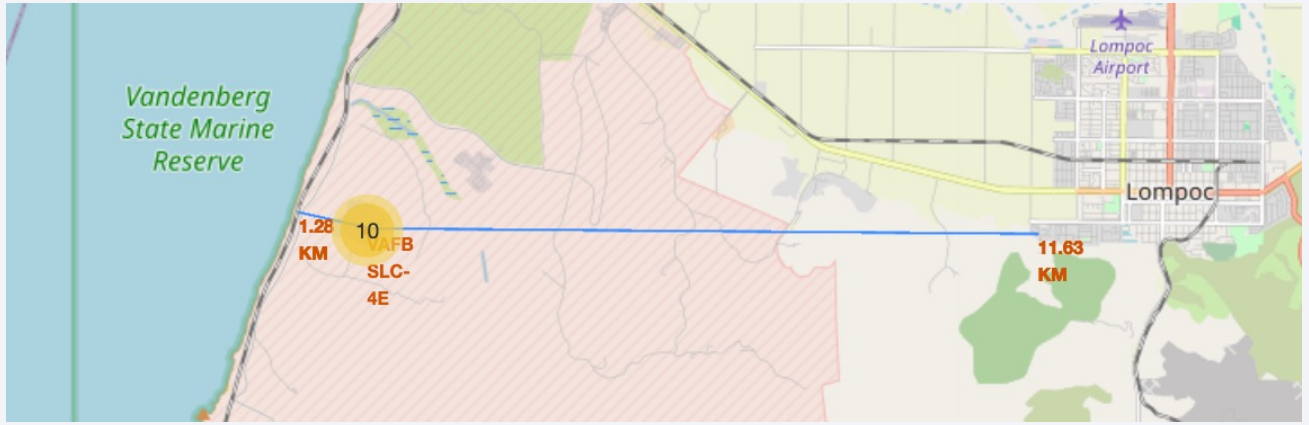
Conclusion:

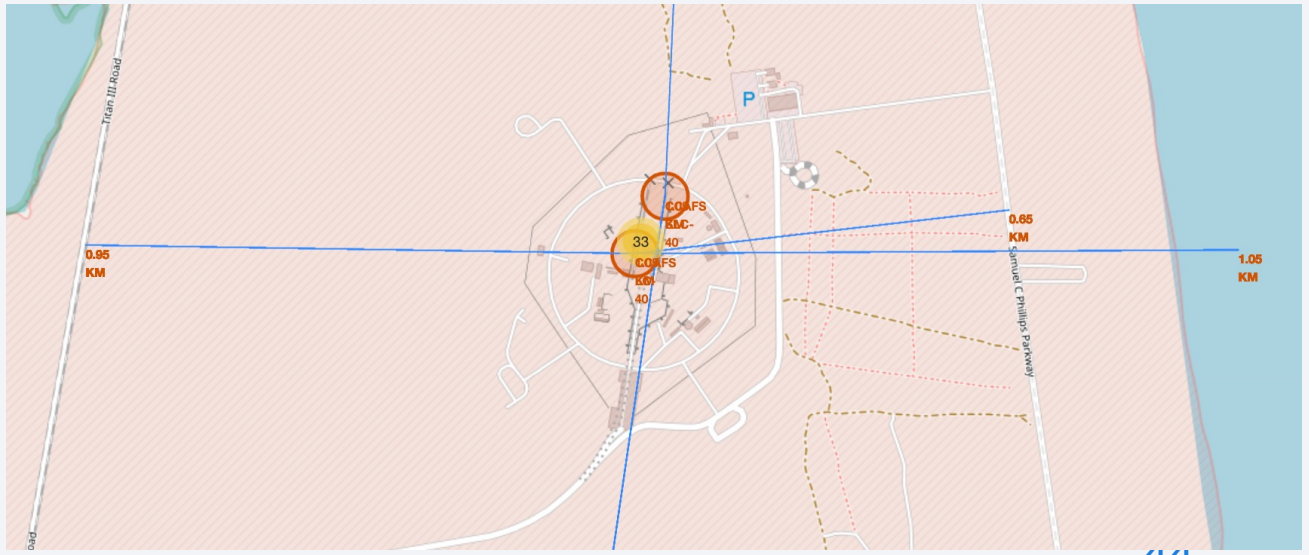
- The result increases over time at CCAFS- LC-40 probably due to that problems are fixed.
  KSC LC-39A has a high success rate in total and the rate also increased over time.

More investigation is needed looking at more parameters.

Distances to important areas from launch sites







Distances to important areas (continued)

Information from the maps by looking at the generated distance lines:

- Shortest distance to nearest town = 11 km
- Longest distance to railroad / highway = 1 km
- Distance to sea line or large free area = 1 km

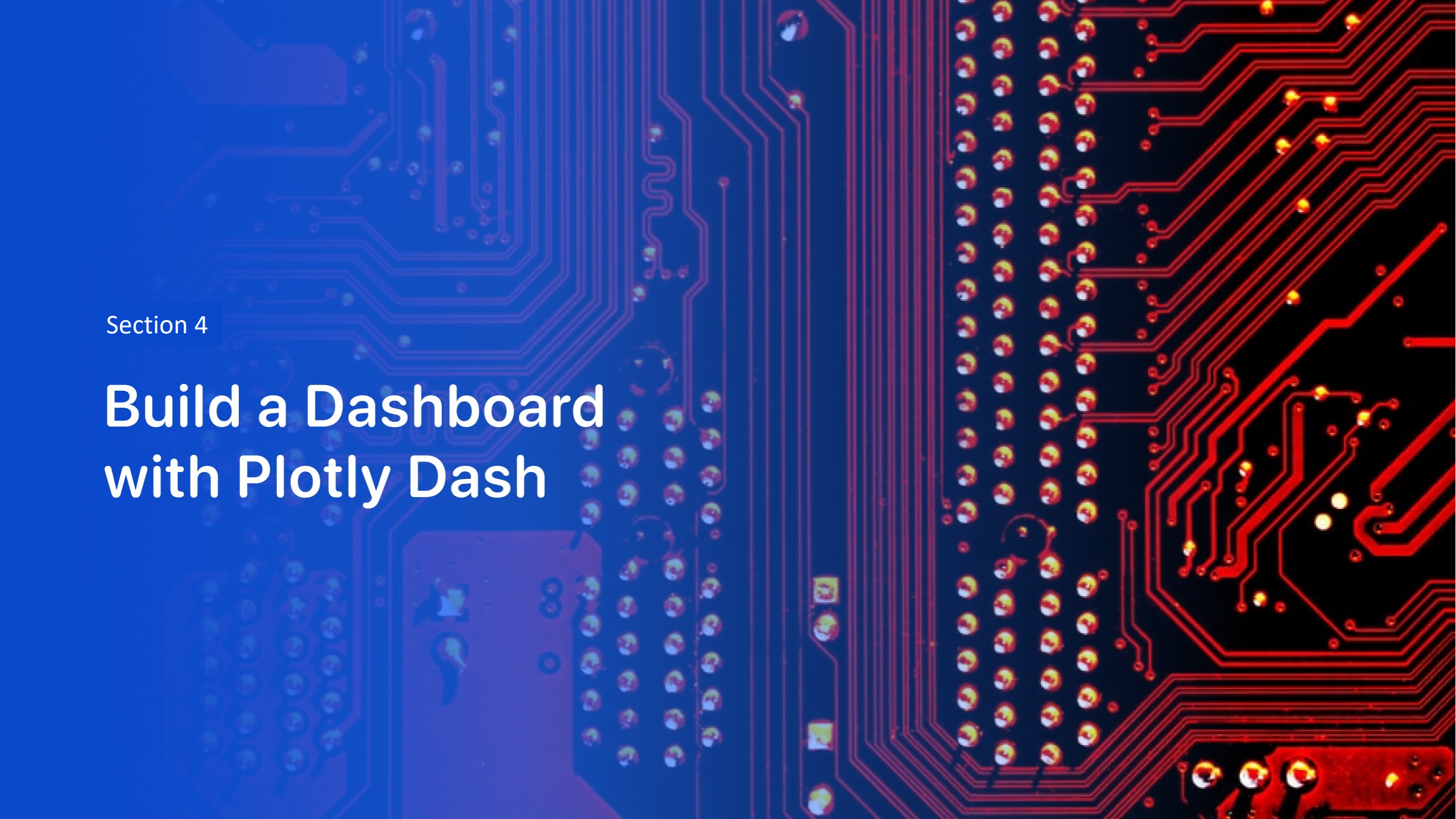
Findings and conclusions

A lot of equipments are needed for launches and they need to be able to be moved quickly and cheaply to and from the launch sites, so therefore are good transportation vital and needs to be available nearby.

Minimum distance to nearest town seems to be over 11 km. Its important to minimize the risk for people and buildings being hit when launches fail.

Distance to the sea or other large free areas are also important to minimize the risk. Extra important are that areas to the east of the lauch site are free, as launches are done to the east to gain extra speed from the earth rotation.

Launch sites are preferred to be near the equator, also to take maximum use of the extra speed from the earth rotation, plus that the number of possible orbit paths are bigger, according to information from NASA. (https://science.nasa.gov/learn/basics-of-space-flight/chapter14-1)

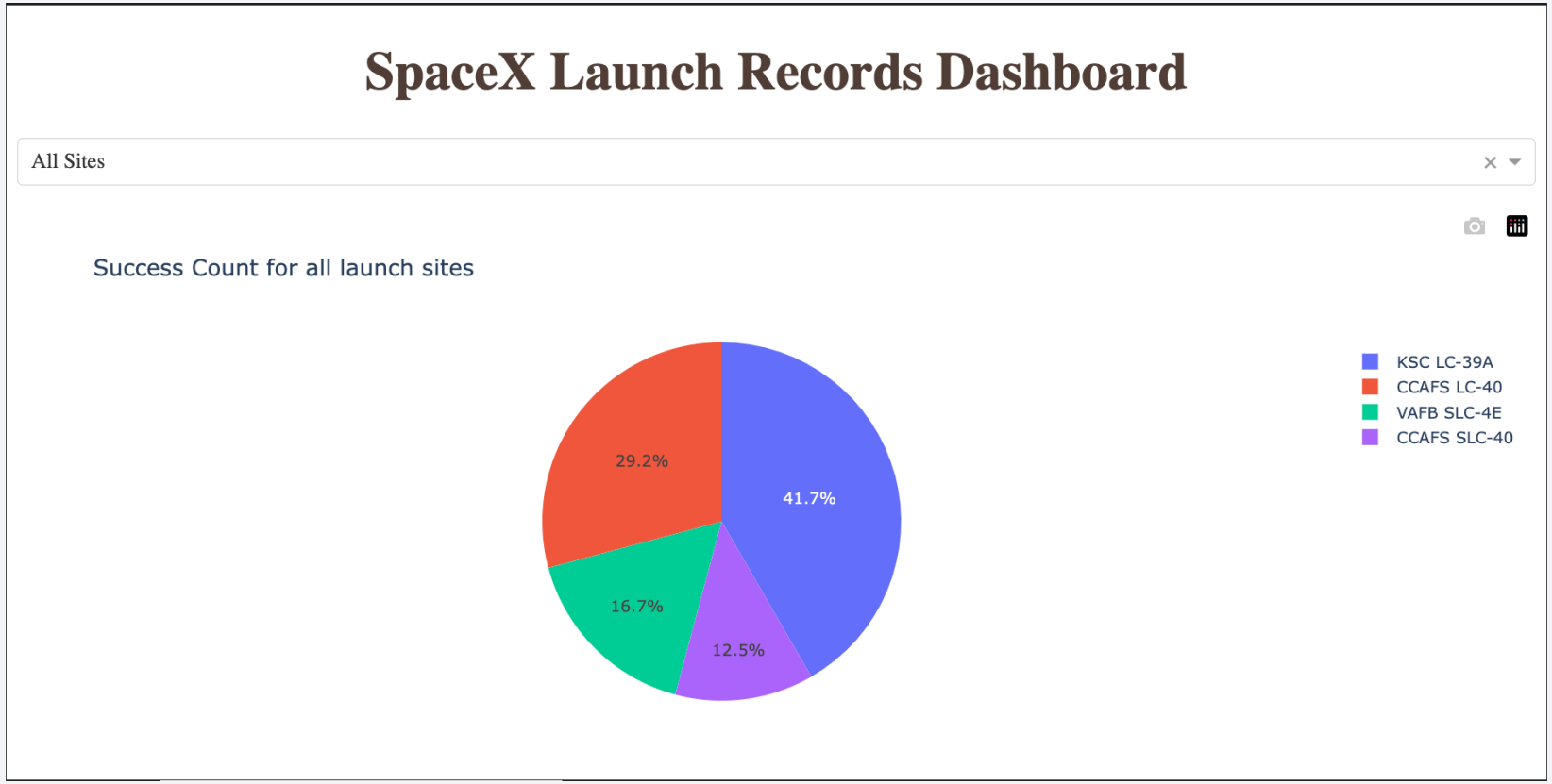


Section 4

Build a Dashboard with Plotly Dash

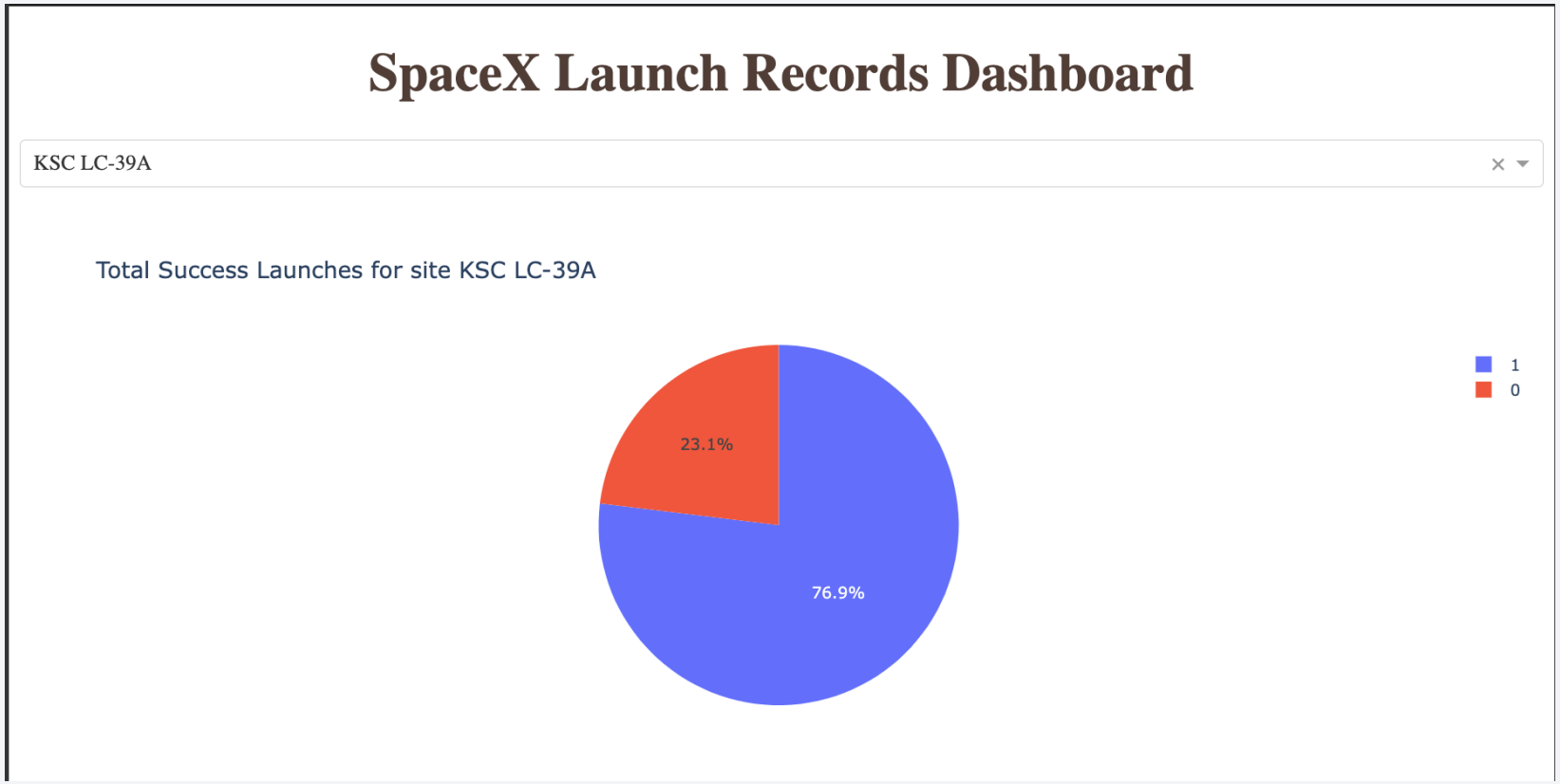
Successful Launches per Launch Site

- The pie chart shows the number of successful launches per Launch Site compared to the total number of successful lauches.
- KSC LC-39A have had the most successful launches of all, 41.7 %. CCAFS LC-40 comes next with 29.2% and then VAFB SLC-4E and CCAFS SLC-40



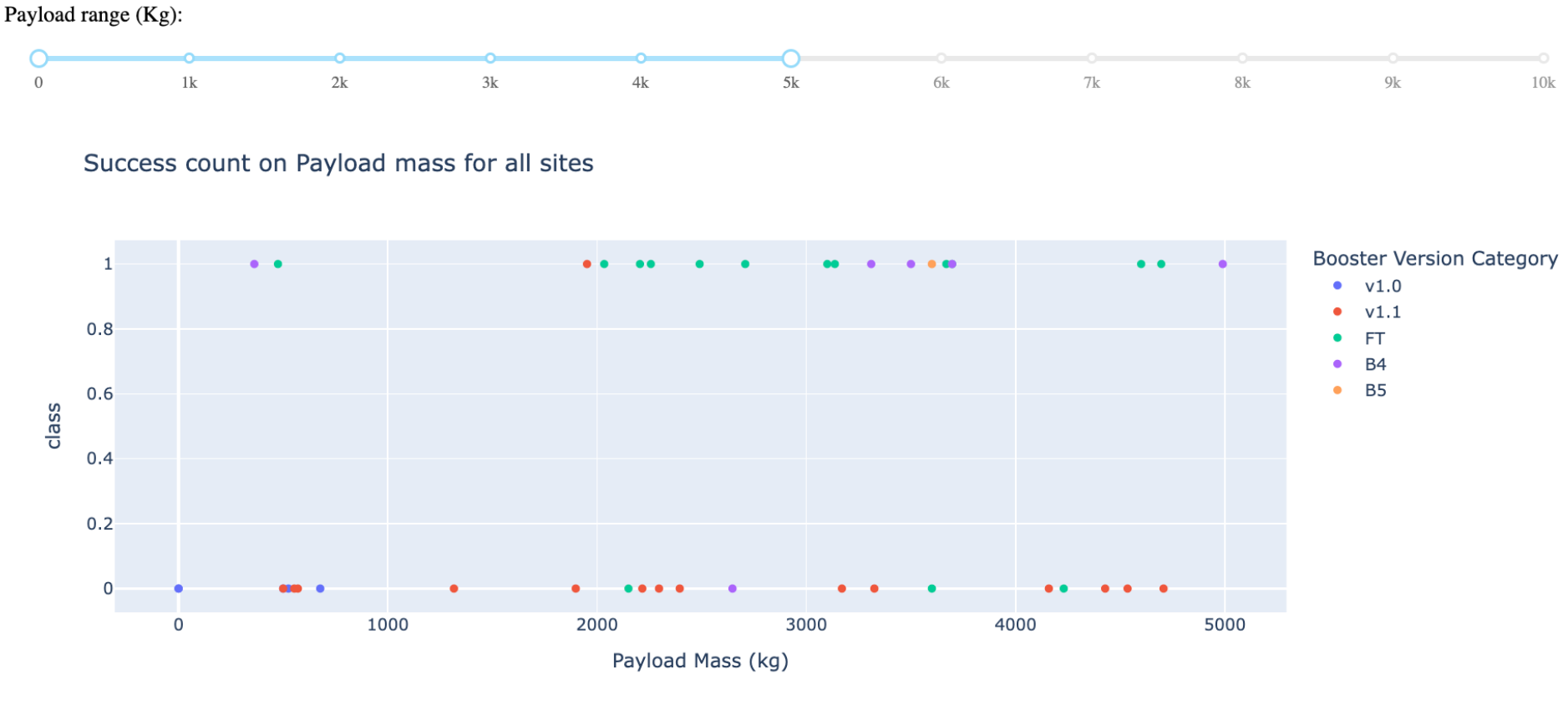
Lauch site with highest launch success rate

- The pie chart shows the success rate for site KSC LC-39A which has the highest success rate of all sites, 76,9 %.



<Dashboard Screenshot 3>

The scatter plot shows success/fail status per launch and Payload Mass in the range 0 - 5000.



<Dashboard Screenshot 3>

The scatter plot shows success/fail status per launch and Payload Mass in the range 5 000 – 10 000.

Conslusions

- Success rate is significantly higher for lauches with lower Payload Mass.

Section 5

Predictive Analysis (Classification)

Classification Accuracy

Accuracy for all algorithms

	Algorithm	Accuracy
0	Decision Tree	0.875000
1	K-Nearest Neighbours	0.848214
2	Support Vector Machine	0.848214
3	Logistic Regression	0.846429

Best algorithm and parameters

```
Best algorithm for predicting Launch status is:  Decision Tree
Best parameters are:  {'criterion': 'entropy', 'max_depth': 10, 'max_features': 'auto', 'min_samples_leaf': 4, 'min_samples_split': 5, 'split
ter': 'random'}
```

Conclusion: We will use Descision Tree algorithm for the prediction model, as it gives the highest accuracy for the predictions (87,5 %)

Code for calculating the accuracy can be found in this Lab-Notebook:
07.SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

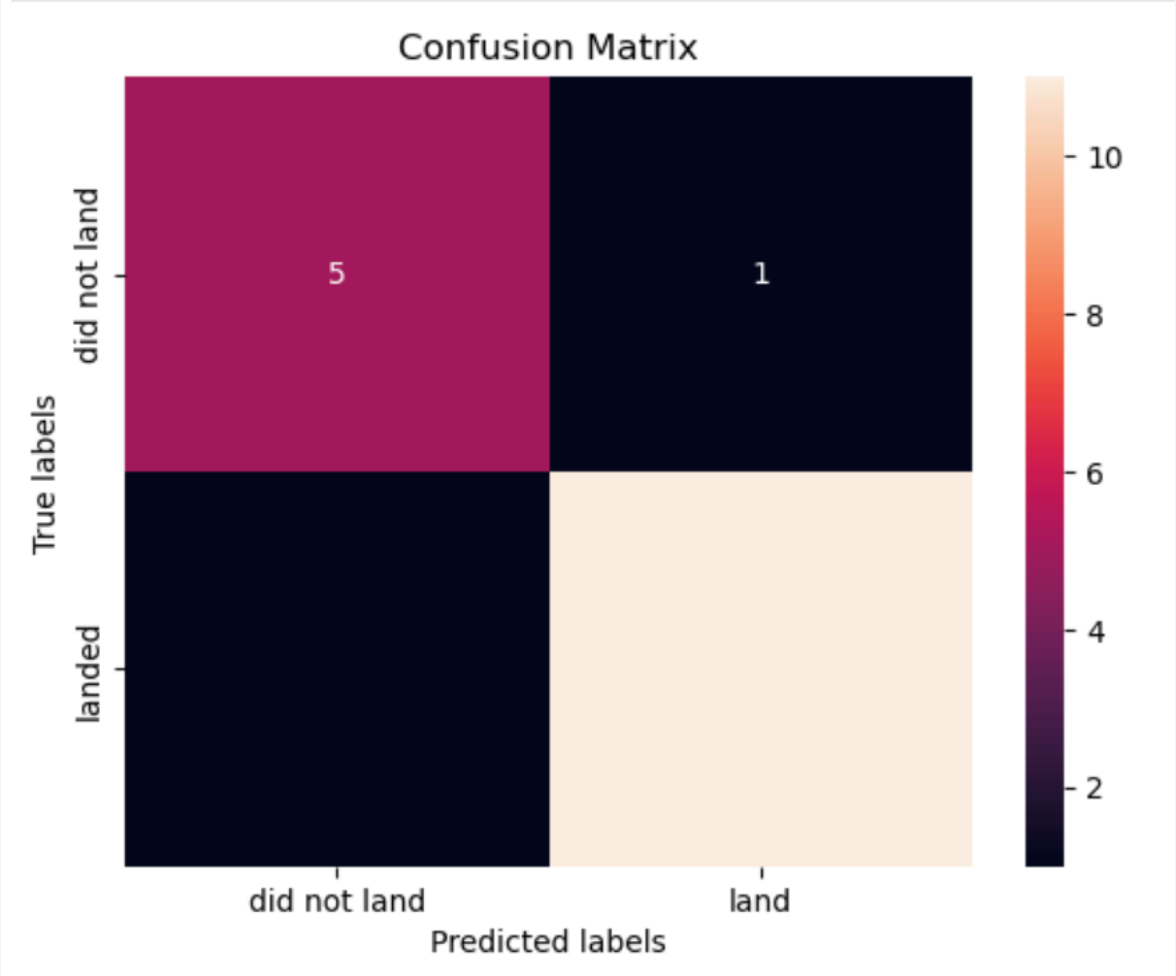
Confusion Matrix

- This is the confusion matrix for the best model, Descision Tree.

There are 2 type of false results:
False Positive meaning the prediction is that launch succeeds, but it failes.
False Negative, meaning the prediction is that launch fails, but it succeeds.

- The worst type of error is, as I see it, False Positive, as it will trigger a better result and potentially a lower calculated cost per launch, which can result that the company looses money in the long
run.

Note: There seem to be a bug in the function plot_confusion_matrix() that makes some values not show.



Conclusions

- Best algorithm to use for predicting outcome of launches is the Decision Tree Algorithm, with 87,5 % success rate.
- Success rate has in general increased with the number of launhces, which is normal due to that problems are fixed, and the organization are learning.
- Success rate seem to be higher for launches with low Payload Mass, than for higher Payload mass.
- Launch sites are located near large areas without people or buildings, like the sea or dessert and swamp, to minimize the risk of serios problems when the launches fail.
- Launch site KSX-LC-39A has the most successful launches.

Appendix

Thank you!