



IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Oscar Ayala
Oct 2024



Outline

- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion

Executive Summary

- SpaceX differentiating factor is their ability to re-land the first stage of rockets.
- Some data about historical launches was accessed via their API. Then, exploratory data analysis was performed looking at any correlations between pairs of variables, including “Payload Mass (kg)”, “Booster Version”, “Orbit”, “Date” and others.
- The available data was split to train four different machine learning classifier models. Then, the other portion of the data was used to test whether the models were able to predict if a given launch (e.g. with a given set of values for the features) could be landed successfully. The models are great at predicting which launches would land, but they are not perfect at identifying those that would not land (provided a set of features for a given mission).
- The best model at predicting is a decision tree classifier, with 0.94 accuracy on the test data, **zero** false negatives and one **false** positive. Links are included to the tuned hyperparameters for all models.
- Overall, there is a positive trend for the percentage of successfully re-landed missions with the pass of time over the course of ten years.

Introduction

- Until recently rocket launches to space were reserved to missions backed by publicly funded agencies like NASA. SpaceX is an emerging private company that aims at significantly reducing the costs to put an object in orbit, by re-using the First stage of the rocket.
- The success of this company lies on the ability to land the first stage of the rocket after a launch. The goal of this work is to:
 - Gather and analyze SpaceX launch data
 - Find any preliminary correlations in the features (e.g. Payload mass (kg), Orbit)
 - Perform predictions of whether a launch would successfully land (1) or not (0) based on the features of a launch.

Section 1

Methodology

Methodology

Executive Summary

- **Data collection methodology:**
 - The data was obtained via get Request calls to the SpaceXdata API¹.
 - A backbone with the key IDs per launch was used as a starting point, to consistently call the API and append the additional columns.
- **Perform data wrangling:**
 - The % of NA values were estimated per column. Then, some basic statistics were run per column to determine the counts per outcome of each variable.
 - The Outcome of the mission column was transformed from string to numeric (0, 1) Response of the dataset.

Methodology

Executive Summary

- **Perform exploratory data analysis (EDA) using visualization and SQL**
 - The data is analyzed using Sqlite. Some basic descriptors were estimated for a few columns, such as: the count of Launch Sites and the date of the first successful launch.
 - Then, some subsets of data were queried to assess the success impact based on some the variables alone or combined (e.g Payload Mass (kg) and Booster Version).
- **Perform interactive visual analytics using Folium and Plotly Dash**
 - A visualization tool was built to enable quick insight into the data.
 - Two graphs were added:
 - A pie chart to count the successful missions with a menu to select one Launch Site or All.
 - A scattered plot to analyze the effect of the Booster version on the Success rate with the ability to graphically filter the data by a range of Payloads.

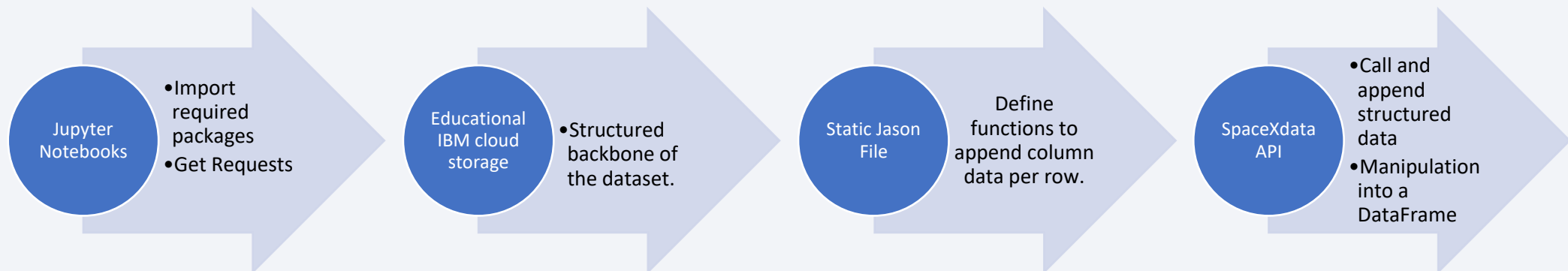
Methodology

Executive Summary

- Perform predictive analysis using classification models
 - The data was split into sets used for training (80% of observations) and testing (20%), of both the “Class” variable (Y) and the group of chosen features (X) for the model.
 - Then, data is used to train some models: Logistic Regression, Support Vector Machines, Tree Classifier and Nearest Neighbors.
 - To estimate the accuracy, the models were ran with the Test data and the output prediction value was compared to the true Class.
 - For each model, the confusion matrix was plotted and the best parameters were estimated and compared.

Data Collection

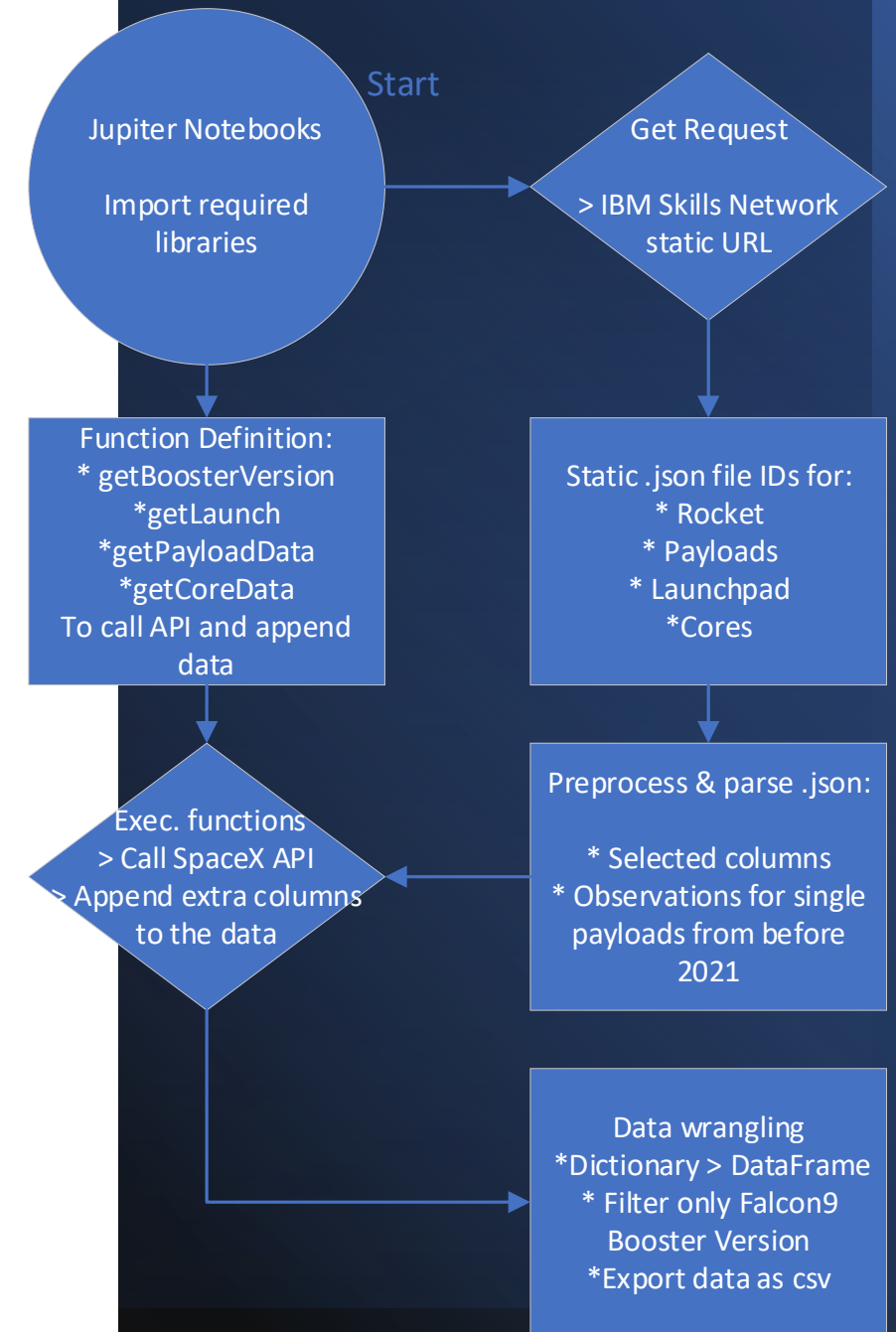
- Auxiliary functions were defined to call the SpaceX API and append column values per row.
- The data is obtained via a Get Request to Educational IBM cloud storage. Then, the json data is converted to a python pandas dataframe object and used as a backbone.
- Pre-processing of the data is performed: columns are selected, rows with multiple values for a column are dropped, date values are converted to UTC. Last, values are limited to launches prior to 2020-11-13. The result is a standard dataframe backbone.
- The auxiliary functions above are used to iterate through each flight, call the API, and append columns values per row.



Data Collection – SpaceX API

In the schematic:

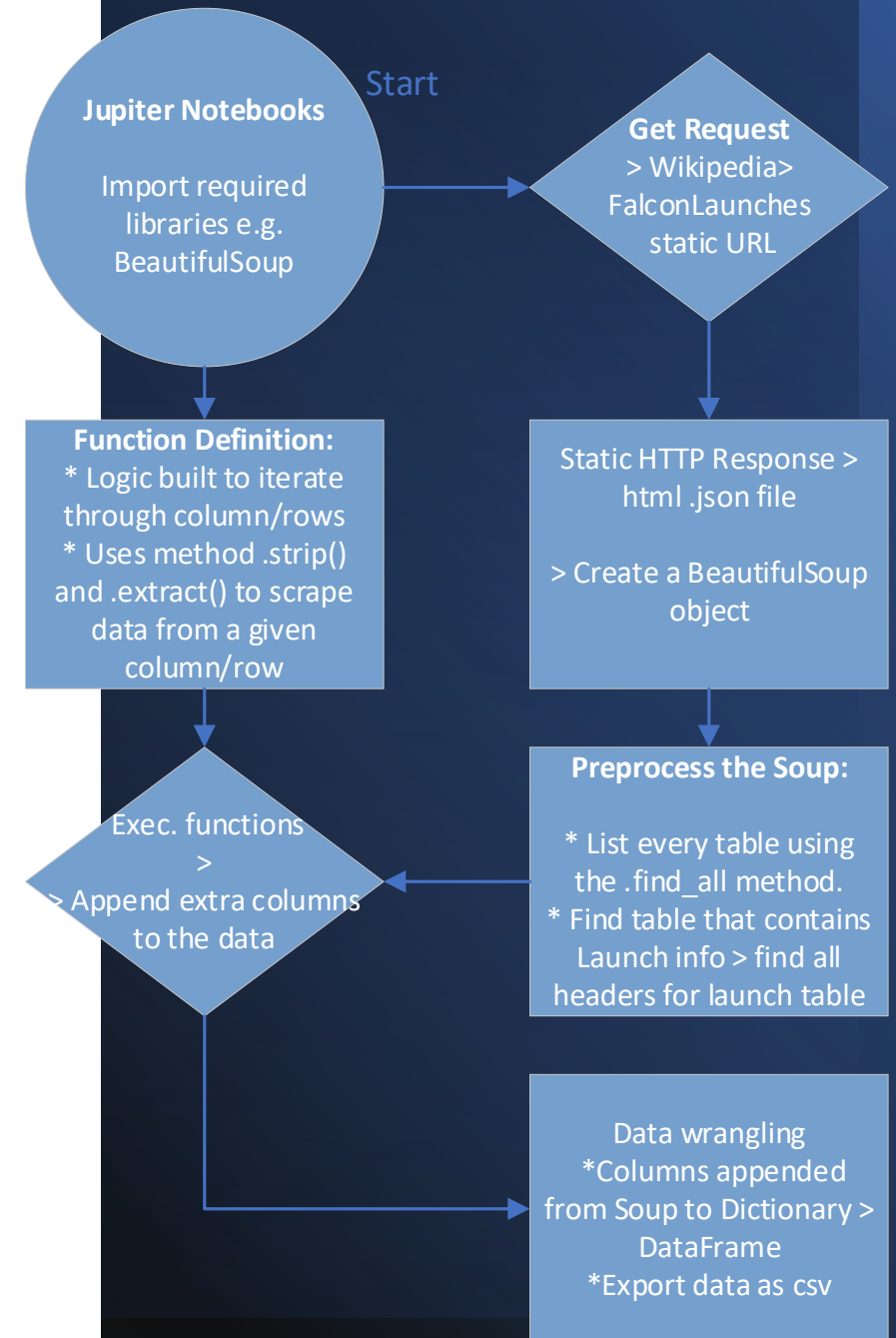
- The circle represents the platform used.
- Rectangles are portions of code that perform an action.
- Diamonds are API calls, the second one iterates through the rows of data to append more columns.
- This process and execution can be found in the Jupyter Notebook on [GitHub](#).



Data Collection - Scraping

In the schematic:

- The circle represents the platform used.
- Rectangles are portions of code that perform an action.
- Diamonds represent the use of a function to convert data
- This process and execution can be found in the Jupyter Notebook on [GitHub](#).



EDA with Data Visualization

- Next, exploratory analysis was conducted to identify correlations between the variables.
- The Payload Mass (kg), the Launch Site, and the Orbit are plotted individually versus the “Flight Number”. This is an increasing counter, so higher values provide an idea of how each of the mentioned variables behaved as the SpaceX team gathered more experience. Also, in each chart the “Class” or outcome variable is used to color each observation differently for the success vs fail landing missions.
- Other exploratory charts are also built for other variables such as Orbit vs Class and Orbit vs Payload Mass. Last, the “Average Success Rate” is calculated per year, and plotted, demonstrating a clear trend as the years progressed.
- All the exploratory work can be found in the [Jupyter Notebook](#) on GitHub.

EDA with SQL

- With the use of the “magic” commands of the sqlalchemy library, the dataset is loaded into a table and queried. Some insightful queries were:
- The average payload can be queried per booster version, for example for F9 v1.1.
- The first successful mission to land in a ground pad.
- A list of the boosters that successfully landed on drone ship, and had carried payloads between 4000-6000 kg.
- A list of the boosters that carried the maximum payload (15600kg).
- A count of the number of missions per possible landing outcome (e.g. Success drone ship, failure parachute, etc)
- The [Jupyter Notebook](#) on GitHub contains the details of these queries and more!

Build an Interactive Map with Folium

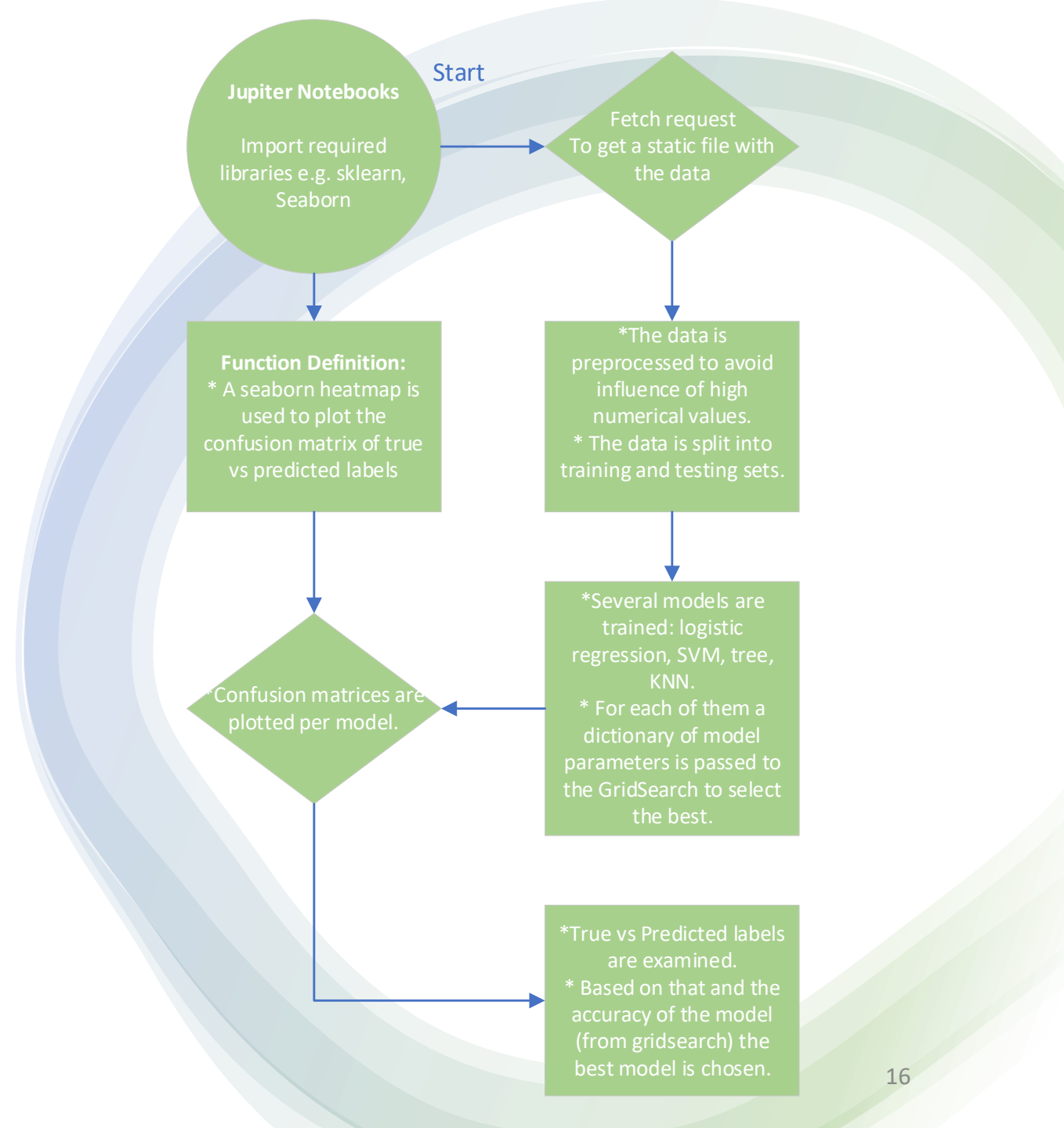
- The folium library was used to build interactive maps with the geographical coordinates of the launch sites.
- To do so, the Circle and Marker objects are used to flag the places in the map. Then, it is possible to visualize all the launching sites in a single map.
- Next a “Color_Marker” column was appended to the data to differentiate the success vs fail landing (i.e. “Class”) in the map. The marker cluster feature is used, since there are many more observations than launch sites, it is necessary to cluster in the map, by providing the number of success attempts. When the clustered number is clicked on the map, a schematic of red (fail) vs green (success) is represented.
- Last, a few coordinates to landmarks (roads, coastline, city, railway) are added and the distance to those is estimated.
- A walk through this process can be found in this [Jupyter notebook](#) on GitHub.

Build a Dashboard with Plotly Dash

- A dashboard was built to enable easy interaction with the data. For this purpose a dropdown menu was added to be able to select all or one of the Launch Sites. Then, a pie chart displays the number of successful launches for the selection.
- In addition, a range slider is added to allow users to select a range of Payload Mass (kg) to filter the data for. Then, a scatter plot displays success or fail attempts (Y) vs the selected payload range. The data can be further drilled down by selecting a desired site. Last, the booster version is added as the “hue” so the boosters show in the scatter plot as a different color.
- The extend of this work can be found in this [python script on GitHub](#). However, to be able to run it users must have a functional way to use Dash locally or in the cloud. This task was initially run in the IDE cloud environment of IBM skills network.

Predictive Analysis - Classification

- The data was split into training and testing set. Several models from the scikit-learn library were trained with the same data, and their prediction was tested and contrasted versus the true output of the test data set.
- This process and execution can be found in the [Jupyter Notebook](#) on GitHub.



Results

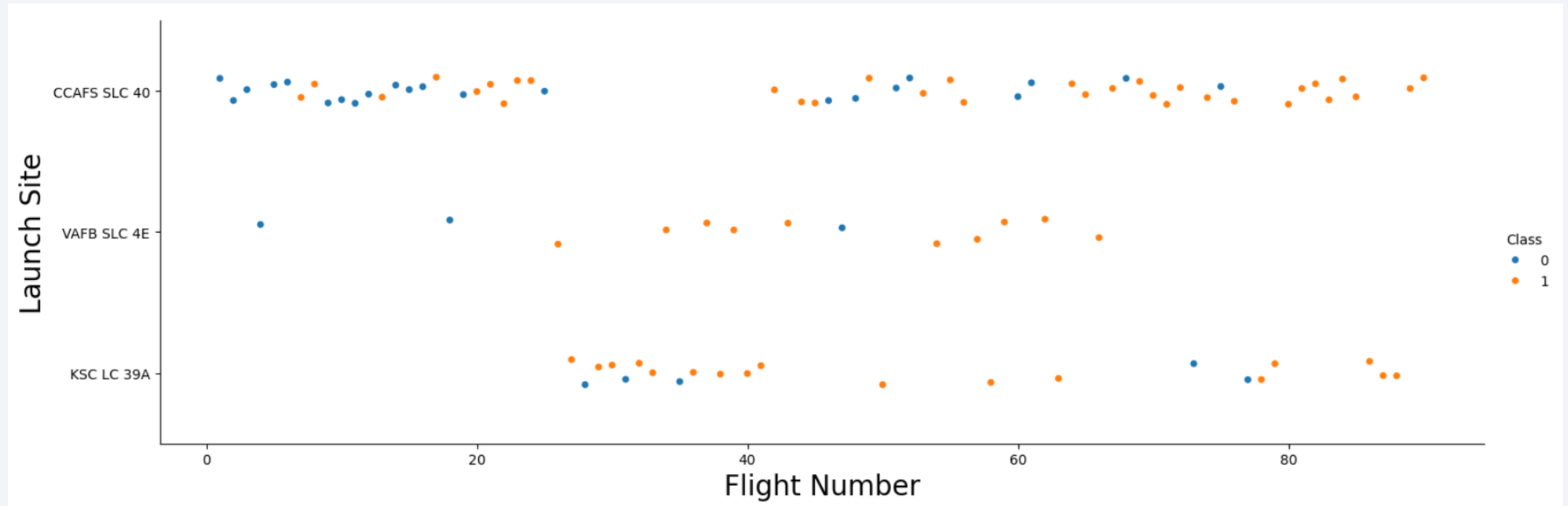
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

The background of the slide is an abstract composition. It features a dark blue base color. Overlaid on this are numerous diagonal streaks in shades of red and cyan. A faint, light blue grid pattern is also visible, particularly in the lower half of the image. The overall effect is dynamic and technological.

Section 2

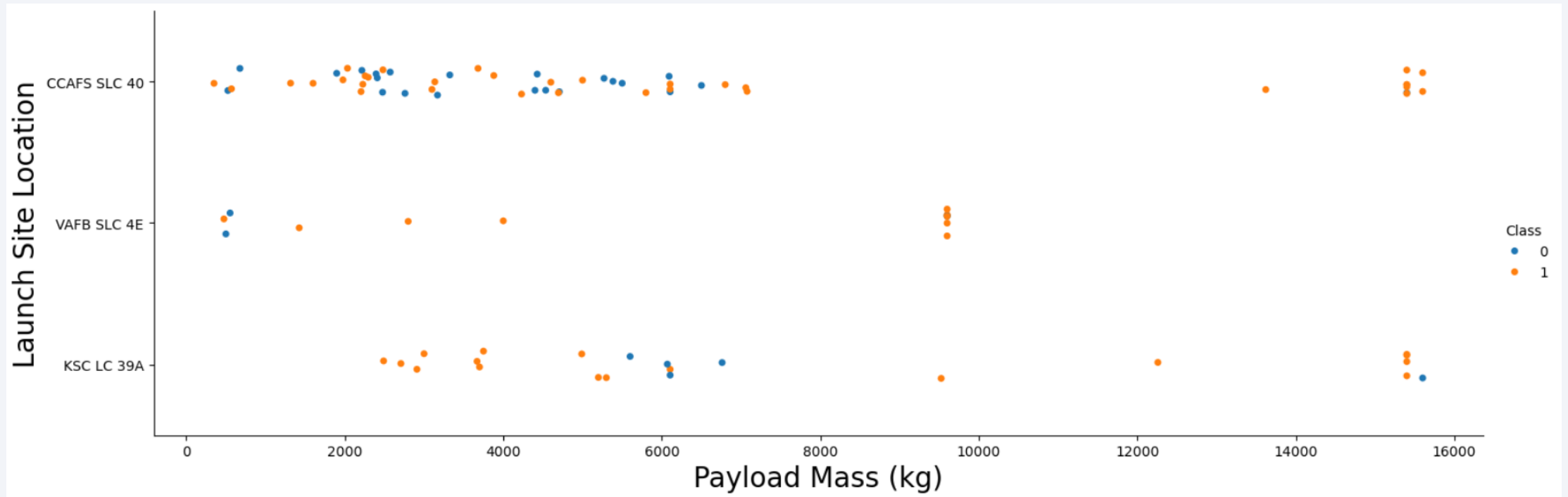
Insights drawn from EDA

Flight Number vs. Launch Site



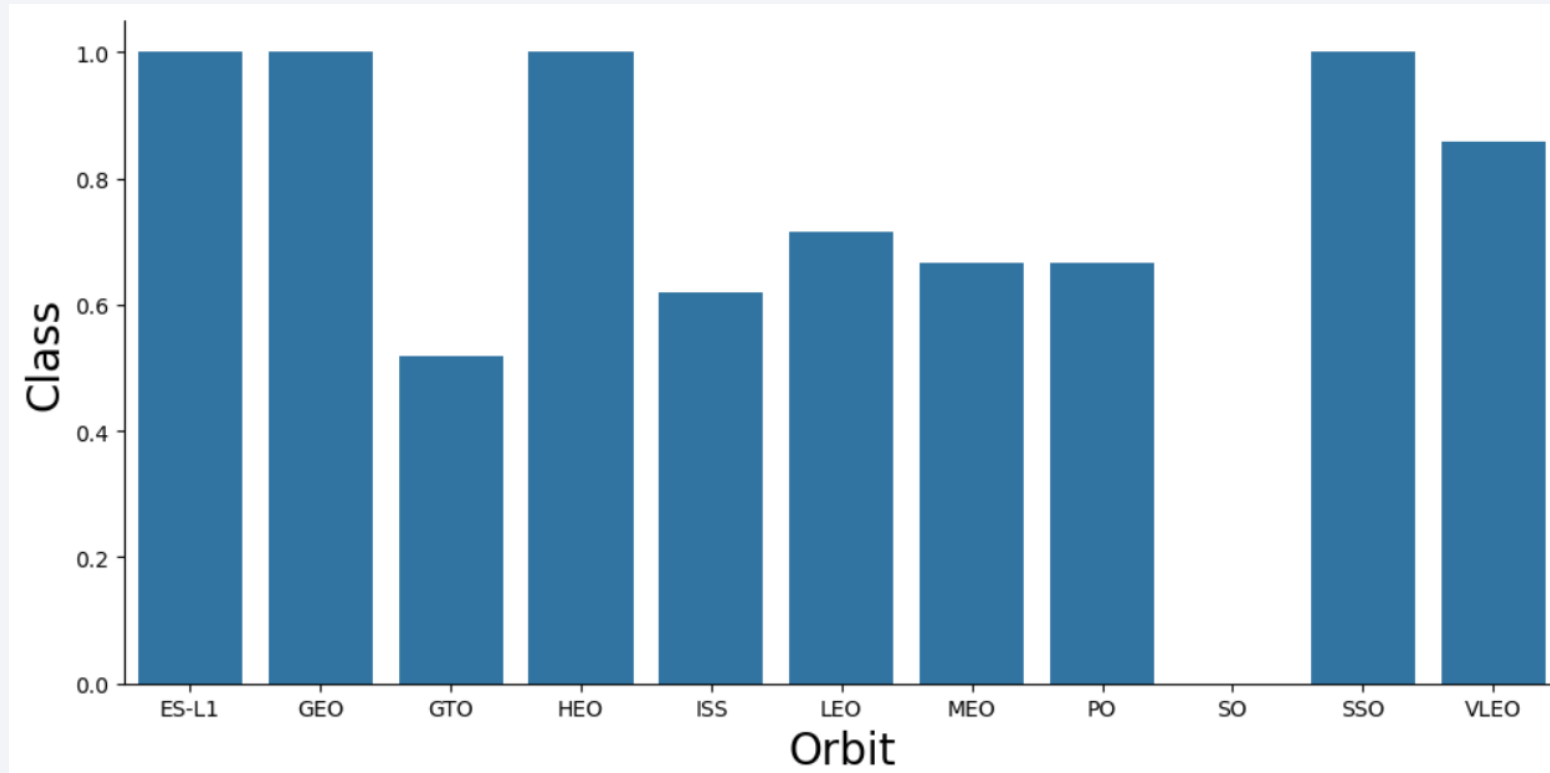
- Some observations: After the launch 60 the number of flights out of VAFB SLC 4E decreases. Also, the success portion of landings increases. Last, the site with highest number of launches to date in that dataset was CCAFS SLC 40.

Payload vs. Launch Site



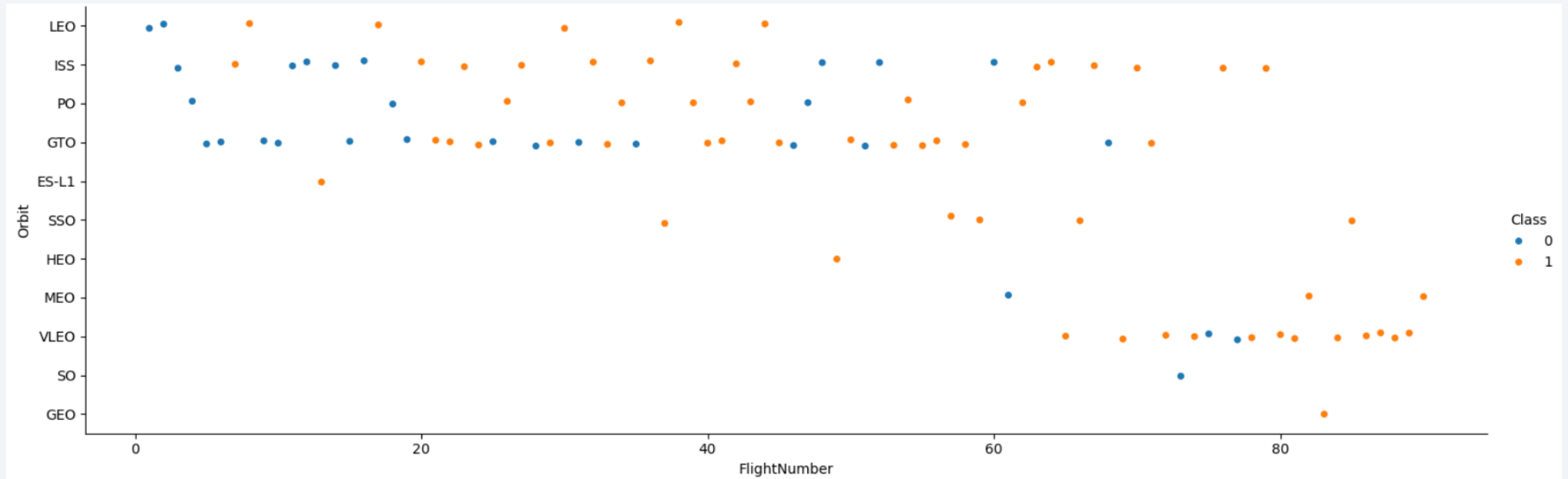
- Relatively small payloads launch from any site, but for heavier ones only CCAFS SLC 40 and KSC LC 39A were chosen. Also, there seems to be a higher success rate with heavier payloads.

Success Rate vs. Orbit Type



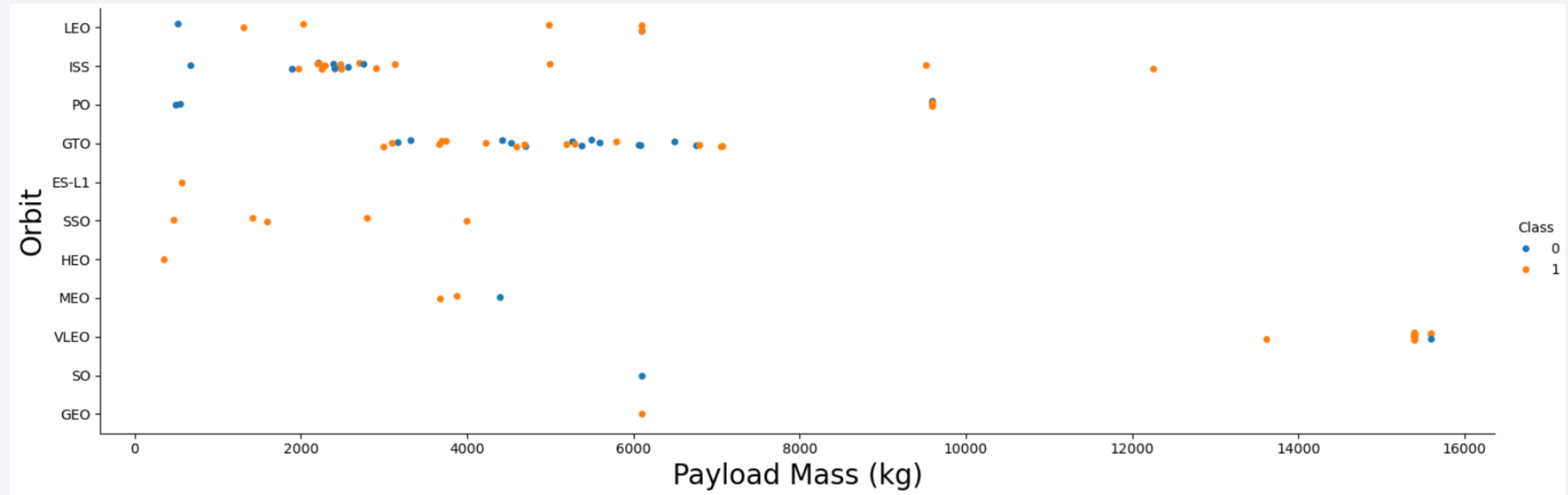
- For missions to orbits ES-L1, GEO, HEO and SSO, every attempt was successful. There were no missions to the SO orbit with successful landing. For other orbits the success launches were 50% of the time or higher.

Flight Number vs. Orbit Type



- A different representation shows the number of flights per Orbit. Single attempts were made to HEO and SO, with only the first having a successful landing. The most number of flight attempts were made to the LEO, ISS, PO and GTO orbits.

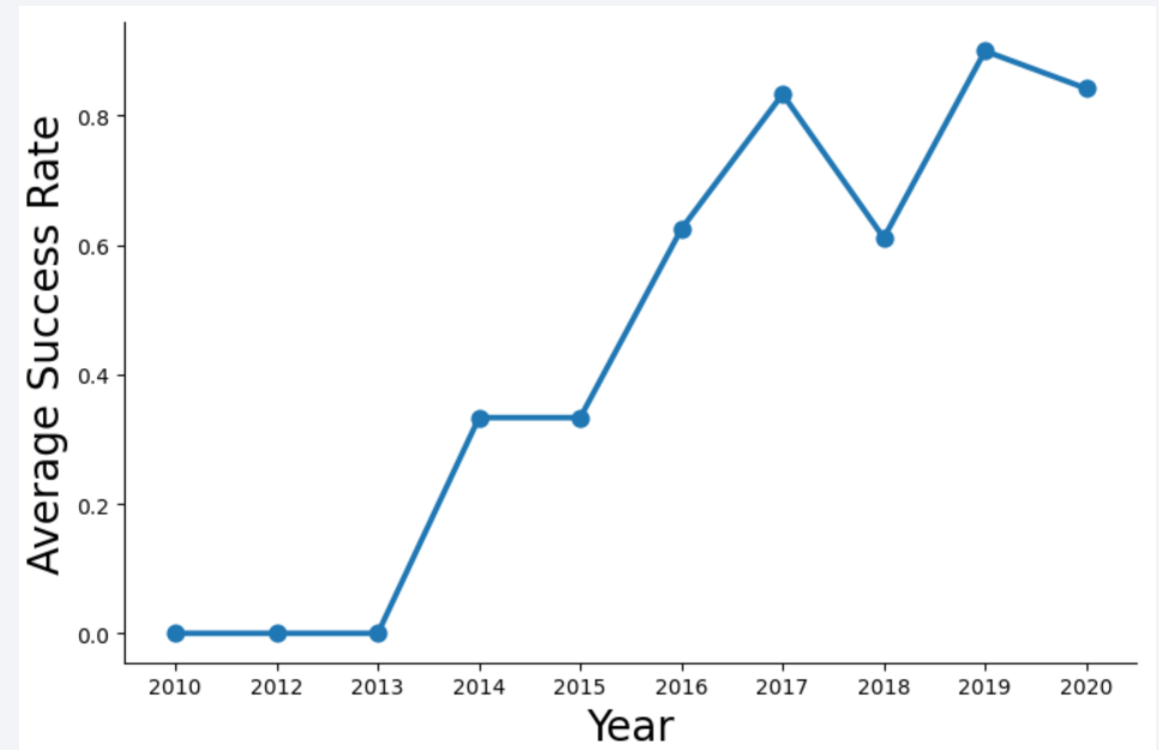
Payload vs. Orbit Type



- The heaviest payloads were taken to the VLEO orbit, with only one unsuccessful landing. Most of the flights to other orbits were taken payloads between 0 and 8000kg.

Launch Success Yearly Trend

- There is a strong increasing tendency to success flights as the years passed by towards 2020.
- For at least 5 years since 2010 the success rate was below 50%.



All Launch Site Names

- The names of the launch sites were obtained from a magic sql query by selecting the unique values for Launch Site within the data set:

```
[ ] %sql select distinct Launch_Site from SPACEXTABLE
```

```
↳ * sqlite:///my_data1.db  
Done.
```

Launch_Site

CCAFS LC-40

VAFB SLC-4E

KSC LC-39A


CCAFS SLC-40

- These are all located across the US.

Launch Site Names Begin with 'CCA'

- Below are 5 records of launches where the Launch site begins with CCA:

```
[ ] %sql select * from SPACEXTABLE where Launch_site like "CCA%" limit 5
```

 * sqlite:///my_data1.db
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	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

Total Payload Mass

- The total payload mass to the “customer” NASA can be counted from a query to this dataset and found to be:

```
[ ] %sql select total(Payload_Mass__Kg_) from SPACEXTABLE where Customer like "%NASA%"
```



```
* sqlite:///my_data1.db
```

```
Done.
```

```
total(Payload_Mass__Kg_)
```

```
107010.0
```

Average Payload Mass by F9 v1.1

- The average payload carried by the Booster version F9 v1.1 is:

```
[ ] %sql select AVG(Payload_Mass__kg_) from SPACEXTABLE where Booster_Version like '%F9 v1.1%'
```



```
* sqlite:///my_data1.db
```

```
Done.
```

```
AVG(Payload_Mass__kg_)
```

```
2534.6666666666665
```


First Successful Ground Landing Date

- The first successful mission to land in a ground pad was in:

```
[ ] %sql select Landing_Outcome, min(Date) from SPACEXTABLE where Landing_Outcome is 'Success (ground pad)'
```



```
* sqlite:///my_data1.db
```

```
Done.
```

```
  Landing_Outcome  min(Date)
```

```
Success (ground pad) 2015-12-22
```

Successful Drone Ship Landing with Payload between 4000 and 6000

```
[65] %%sql select Booster_Version, Payload_Mass__kg_, Landing_Outcome
      from SPACEXTABLE
      where Landing_Outcome is "Success (drone ship)" and Payload_Mass__kg_>4000 and Payload_Mass__kg_<6000
```



```
* sqlite:///my_data1.db
```

```
Done.
```

Booster_Version	PAYLOAD_MASS__KG_	Landing_Outcome
F9 FT B1022	4696	Success (drone ship)
F9 FT B1026	4600	Success (drone ship)
F9 FT B1021.2	5300	Success (drone ship)
F9 FT B1031.2	5200	Success (drone ship)

Notice that there is only one successful attempt for each Booster, and each of them seem to have a slight variation from the previous model.

Total Number of Successful and Failure Mission Outcomes

```
%sql select Mission_Outcome, Count(Mission_Outcome) from SPACEXTABLE group by Mission_Outcome
```

```
* sqlite:///my_data1.db
```

```
Done.
```

Mission_Outcome	Count(Mission_Outcome)
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

- The count of successful and failure missions is presented. It can be noted that there are different types for each. Also, the variable “Mission Outcome” should not be confused with “Landing Outcome”!

Boosters Carried Maximum Payload

```
%sql select Booster_Version, Payload_Mass__kg_ from SPACEXTABLE where (Payload_Mass__kg_ in (select max(Payload_Mass__kg_) from SPACEXTABLE ))
```


* sqlite:///my_data1.db
Done.

Booster_Version	PAYLOAD_MASS__KG_
F9 B5 B1048.4	15600
F9 B5 B1049.4	15600
F9 B5 B1051.3	15600
F9 B5 B1056.4	15600
F9 B5 B1048.5	15600
F9 B5 B1051.4	15600
F9 B5 B1049.5	15600
F9 B5 B1060.2	15600
F9 B5 B1058.3	15600
F9 B5 B1051.6	15600
F9 B5 B1060.3	15600
F9 B5 B1049.7	15600

- The heaviest payloads carried ever were of 15600kg per mission. The query above shows all the Booster Versions that have been used to carry these payloads.

2015 Launch Records

```
[64] %%sql select Year,
CASE
  when Month is "01" then "January"
  when Month is "02" then "February"
  when Month is "03" then "March"
  when Month is "04" then "April"
  when Month is "05" then "May"
  when Month is "06" then "June"
  when Month is "07" then "July"
  when Month is "08" then "August"
  when Month is "09" then "September"
  when Month is "10" then "October"
  when Month is "11" then "November"
  when Month is "12" then "December"
end as Month,
Landing_Outcome, Booster_Version, Launch_Site
from (select substr(Date, 6,2) as Month, substr (Date,0,5) as Year, Landing_Outcome, Booster_Version, Launch_Site from SPACEXTABLE)
where Landing_Outcome is "Failure (drone ship)" and Year is '2015'
```

 * sqlite:///my_data1.db
Done.

Year	Month	Landing_Outcome	Booster_Version	Launch_Site
2015	January	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
2015	April	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

- There were only two records in 2015 where the outcome of the mission was Failure to land in drone ship.

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

```
%%sql select Landing_Outcome, count(Landing_Outcome) as Number_of_times
from (select * from SPACEXTABLE
      where CAST(strftime('%s', Date) as INT) > CAST(strftime('%s', "2010-06-04") as INT)
      and CAST(strftime('%s', Date) as INT) < CAST(strftime('%s', "2017-03-20") as INT)
      )
group by Landing_outcome order by count(Landing_Outcome) desc
```

```
* sqlite:///my_data1.db
Done.
```

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

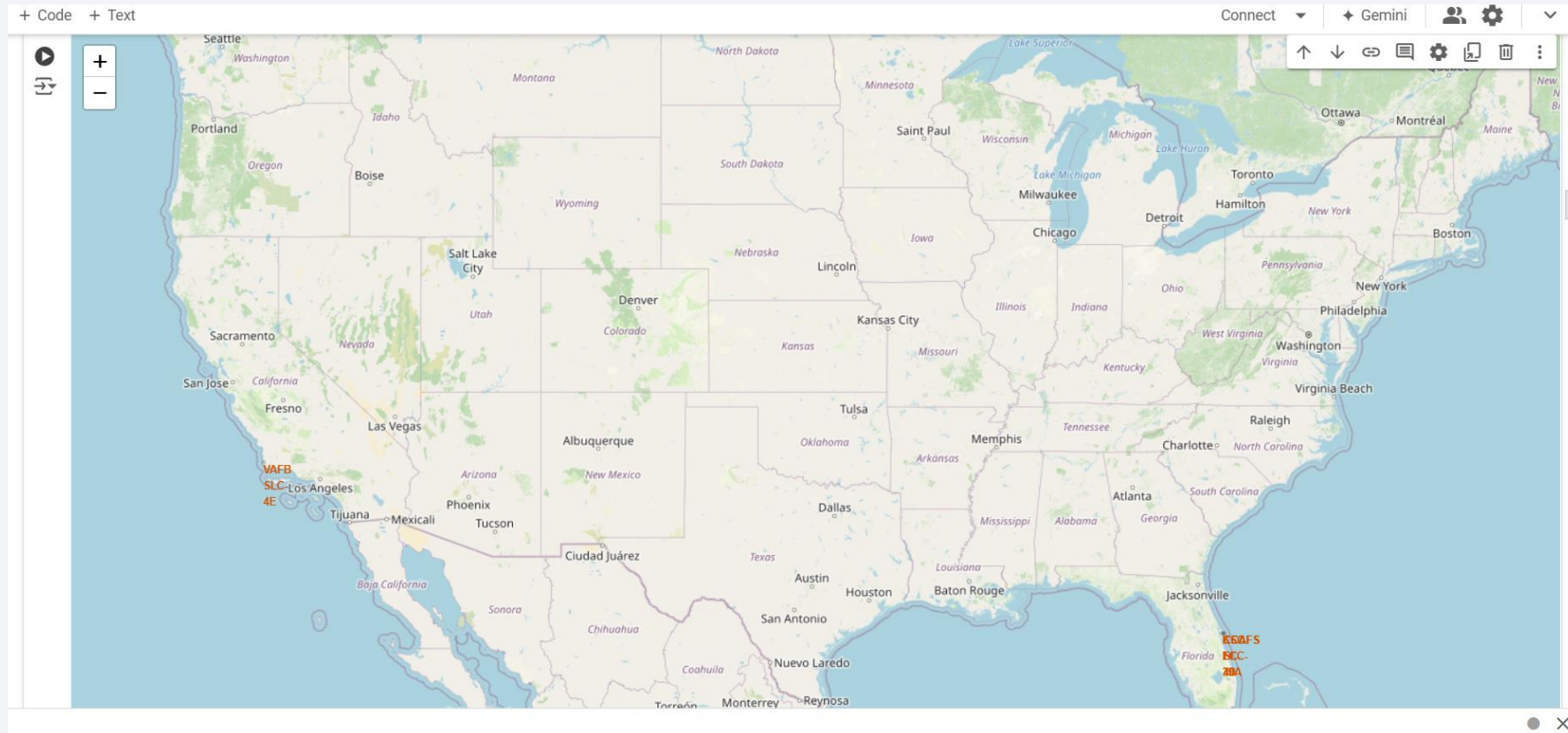
- The Landing Outcomes were categorized based on the number of times they occurred for all the missions on record within the given dates. Then, they are presented from highest to lowest occurrence.
- It can be noticed that most times there was no landing attempt.

A satellite view of Earth from space, showing the curvature of the planet and city lights at night. The background is a deep blue gradient.

Section 3

Launch Sites Proximities Analysis

Folium Maps for python: All Launch Sites



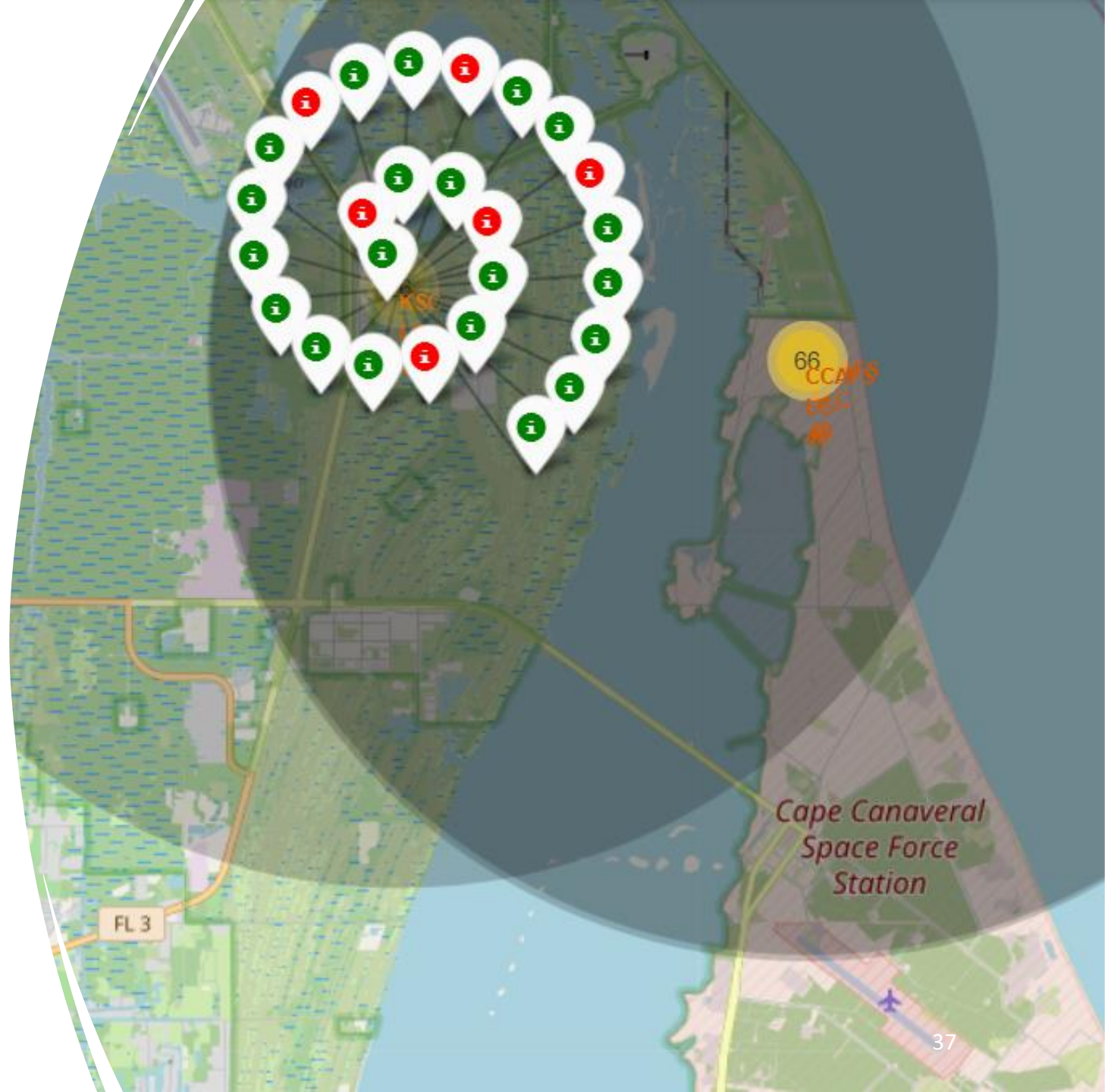
- The 4 Launch sites used by SpaceX are depicted in the map, using their coordinates.
- Notice that at this level of zoom, the three launch sites in Florida seem like one.

Folium map

Clustered markers :

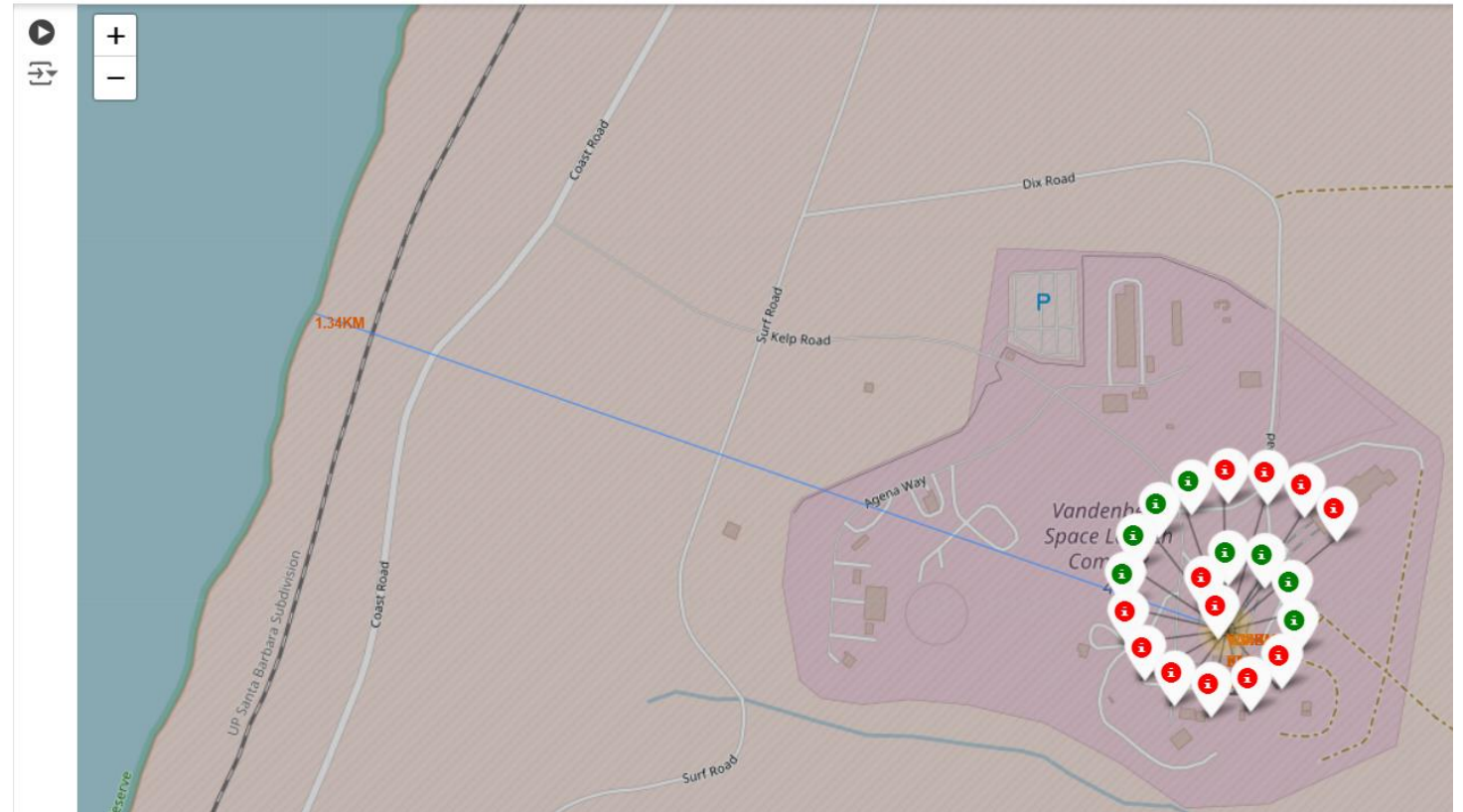
Success vs Fail

- In this map, the success landing missions are depicted with a Green marker, and the fail ones with Red.
- Since groups of markers would land in the same coordinates, it is necessary to use clusters.
- These improve navigability, since all of the launches from a given coordinate show as a counter inside the yellow circle. Then, when clicked it expands to show each individual Green or Red marker.



Folium Map : Distance to Landmarks

- It is also possible to calculate the distance between the launch sites and any geographical landmark.
- In the current image, the distance between the Vandenberg Space Launch Complex and the costal line is depicted.
- This calculated distance is based on the coordinates of two different points.





Section 4

Build a Dashboard with Plotly Dash

Dashboard Successful Launches by Site

- This dashboard allows for an interactive way of filtering data. This pie chart represents the total successful launches for all sites. When a specific site is selected, it depicts the proportion of fail vs success landings for the given site.
- The site with the highest number of successful launches is “CCAFS LC-40”

SpaceX Launch Records Dashboard

All Sites

×

Total successful launches by Site



SpaceX Launch Records Dashboard

CCAFS SLC-40

×

Success vs Failure count for CCAFS SLC-40



Highest success rate site

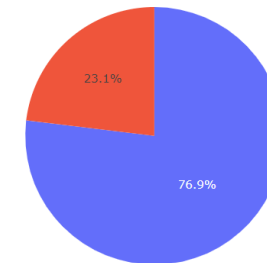
- In this case, the site with the highest success rate is “KSC LC-39A”
- Notice it is not the same as the site with the highest number of successful launches (“CCAFS LC-40”)

SpaceX Launch Records Dashboard

KSC LC-39A

×

Success vs Failure count for KSC LC-39A



■ 1
■ 0

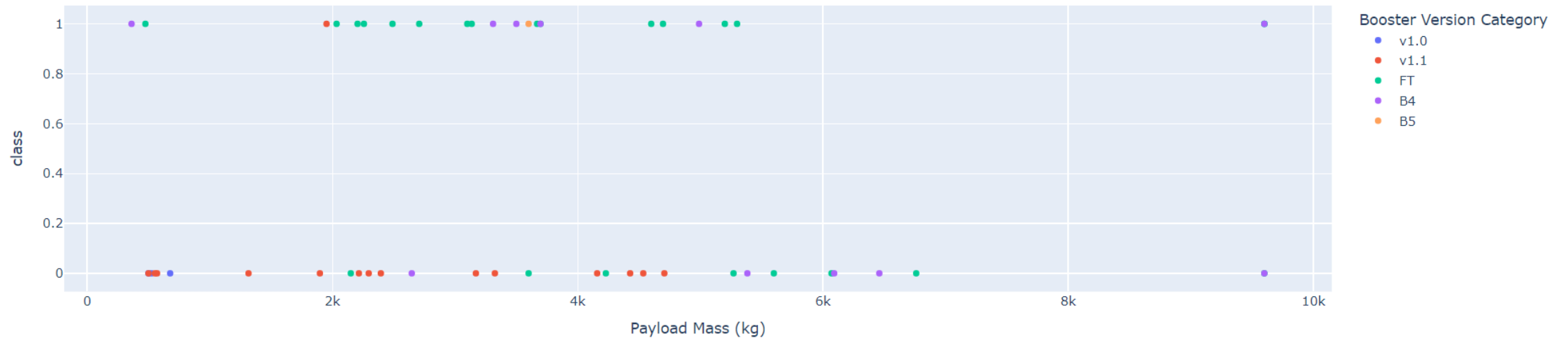
Dashboard with Payload vs Launch Outcome

- The Booster version with the highest number of success landings is “FT” in Green.
- The Payload with the highest success rate is in the range of 2000 to 4000 kg

Payload range (Kg):



Success vs Fail launches by Payload for All Sites

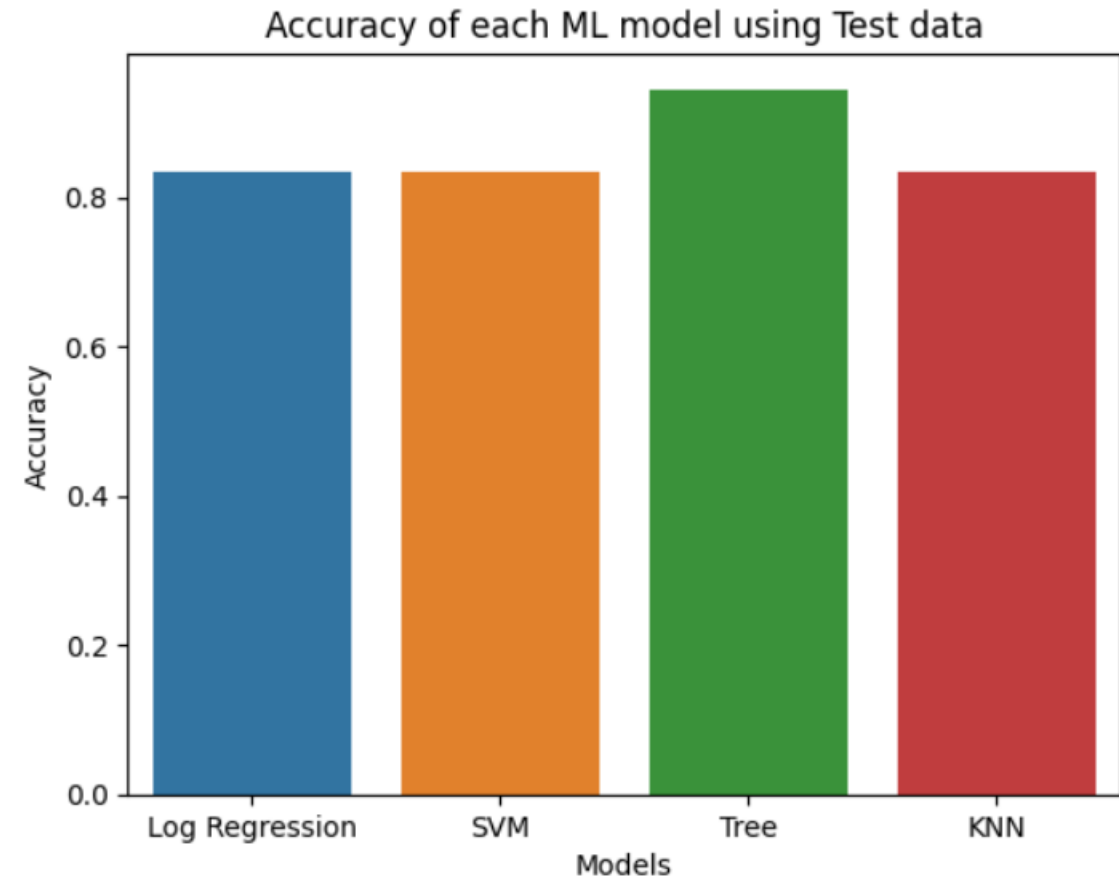


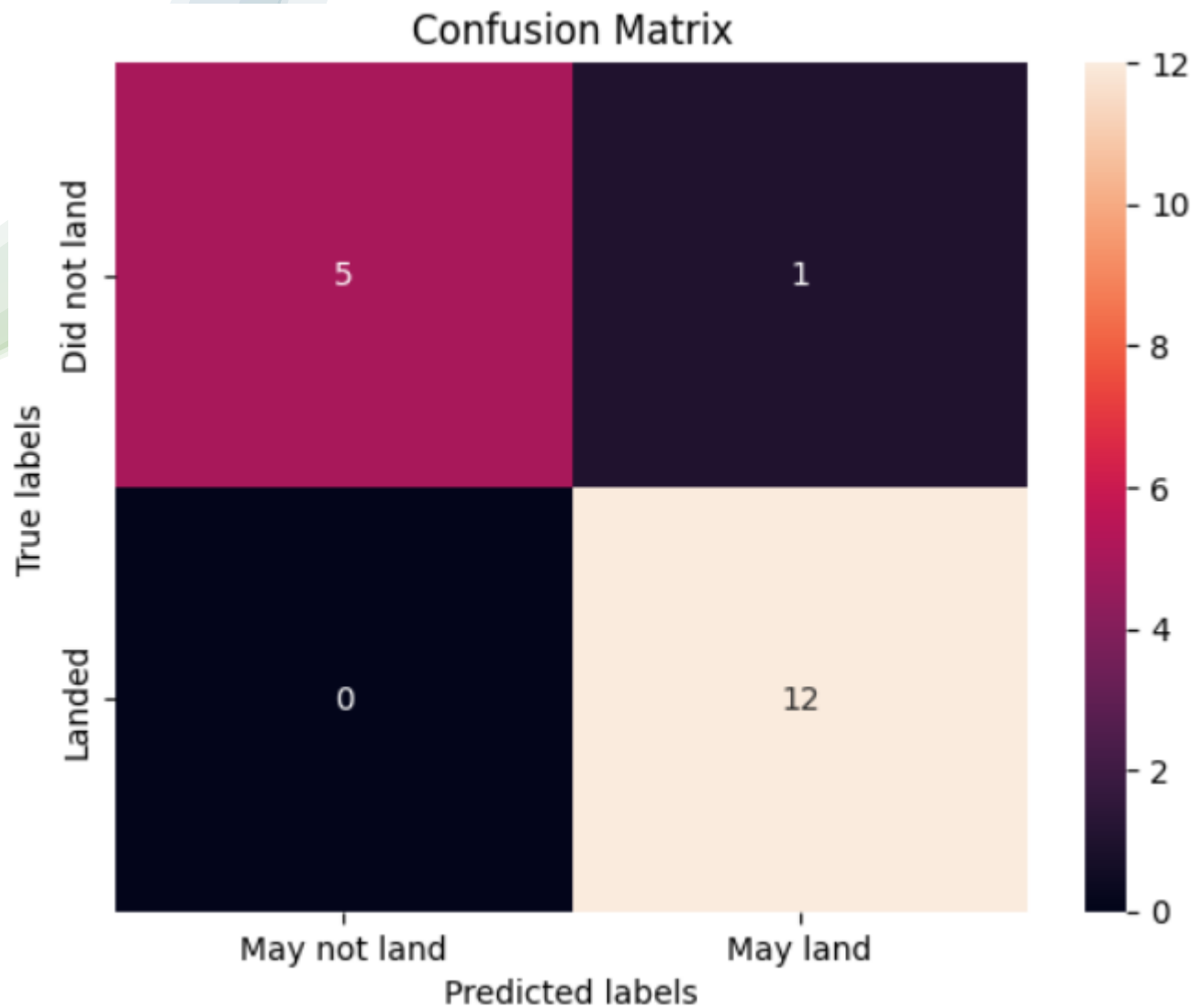
Section 5

Predictive Analysis (Classification)

Classification Accuracy

- The data was split in Train and Test data. Four machine learning models classifiers were trained and tested.
- The prediction accuracy is calculated using the test data, and it is presented in the chart.
- The tree classifier had the highest accuracy of 0.9017. Other models had around the same accuracy of 0.8333.





Confusion Matrix

For the Decision Tree model all the launches that landed are predicted by the model (12/12), so the **false negatives** are zero.

The launches that did not land are 6 in total, and 5 are accurately predicted as “May not land”. This model incorrectly predicts that 1 of those launches “May land” based on feature values of that launch.

In other words, there is 1 **false positive** with the decision tree classifier.

Conclusions

- SpaceX launch data can be gathered and analyzed to produce insights.
- As the years pass the SpaceX team became better and better at re-landing their rockets.
- Machine learning models can be built to predict the ability to land (or not) a rocket based on parameters associated with the flight, such as the payload, the orbit, the booster version and so on.
- After tuning the hyperparameters, the model that best predicts the ability to land a rocket (based on a set of values for the features) is the decision tree.

Thank you!

