Winning Space Race with Data Science

Teófilo Percy Landa
June 2022

Outline

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

Executive Summary

Summary of methodologies

Data collection using API and Web Scraping

Data wrangling (data improvement)

Exploratory Data Analysis with SQL

Exploratory Data Analysis using Pandas (statistical analysis) and Matplotlib (visualization)

Interactive map to analyze the launch site proximity data with Folium

Interactive dashboard to analyze launch records with Plotly Dash (Terminal)

Machine Learning Prediction: Perform exploratory Data Analysis and determined Training Labels to find best Hyperparameter for SVM, Classification Trees, Logistic Regression and KNN

Summary of all results

Collected data from SpaceX and Wikipedia using both API and Web Scraping respectively

Improved the quality of the data by performing data wrangling

Explored the processed data with SQL queries to gather insights

Gained further insights into the data by applying some basic statistical analysis (using Pandas) and data visualization (using Matplotlib)

Analyzed the launch site proximity with Folium (interactive map)

Analyzed launch records interactively with Plotly Dash (interactive dashboard)

Machine Learning Prediction: found the method that performs best

Introduction

Project background and context

The commercial space age is here and companies are making space travel affordable for everyone. Perhaps SpaceX is the most successful because their rocket launches are relatively inexpensive (they can reuse the first stage)

Therefore, determining if the first stage will land will provide the cost of a launch.

Unlike other rocket providers, SpaceX's Falcon 9 Can recover the first stage. Sometimes the first stage does not land. Sometimes it will crash. Other times, Space X will sacrifice the first stage due to the mission parameters like payload, orbit, and customer.

Problems you want to find answers

As a data scientist working for a new rocket company (Space Y) that would like to compete with SpaceX, my job is to determine the price of each launch.

I will do this by gathering information about Space X and creating dashboards for your team. I will also determine if SpaceX will reuse the first stage. Instead of using rocket science to determine if the first stage will land successfully, I will train a machine learning model and use public information to predict if SpaceX will reuse the first stage.

Section 1

Methodology

Methodology

Executive Summary

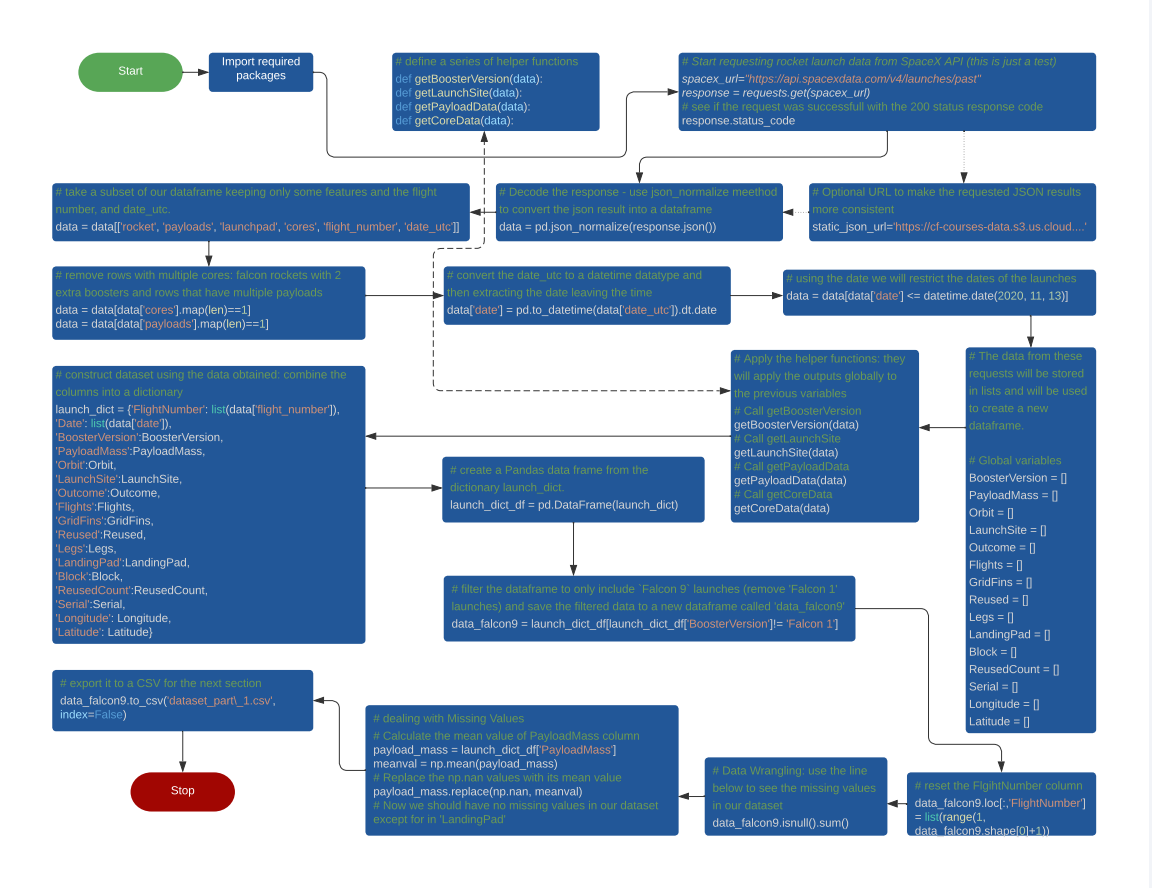
- Data collection methodology:
  - Collected data from SpaceX and Wikipedia using both API and Web Scraping
- Perform data wrangling
  - Improved the quality of the data by performing data wrangling
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection

- Data was collected from two main sources using the ‘requests.get()’ method from Python’s Requests library
  - **SpaceX API**: getting data from this Open Source REST API
    ```
    # Sample code:
    spacex_url = "https://api.spacexdata.com/v4/launches/past"
    response = requests.get(spacex_url)
    ```
  - **A Wikipedia page**: getting data from a static URL in Wikipedia; the page was titled *“List of Falcon 9 and Falcon Heavy launches”* and was updated on 9th June 2021
    ```
    # Sample code:
    static_url = "https://en.wikipedia.org/w/index.php?
    title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"
    # use requests.get() method with the provided static_url
    # assign the response to a object
    data = requests.get(static_url).text
    ```

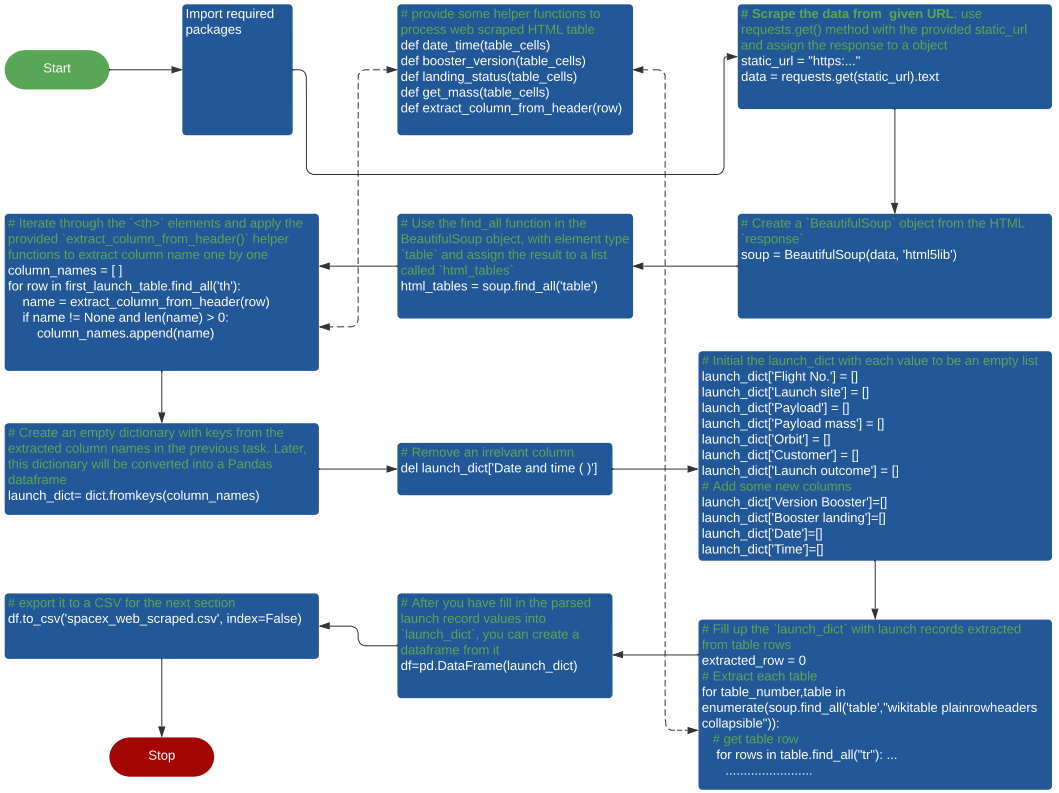
Data Collection – SpaceX API

- Used the 'get' request to collect data from SpaceX API. Then did data cleaning, basic data wrangling and formatting. See flowchart on the right for step-by-step details.
- Here is the link to the completed SpaceX API calls notebook in GitHub



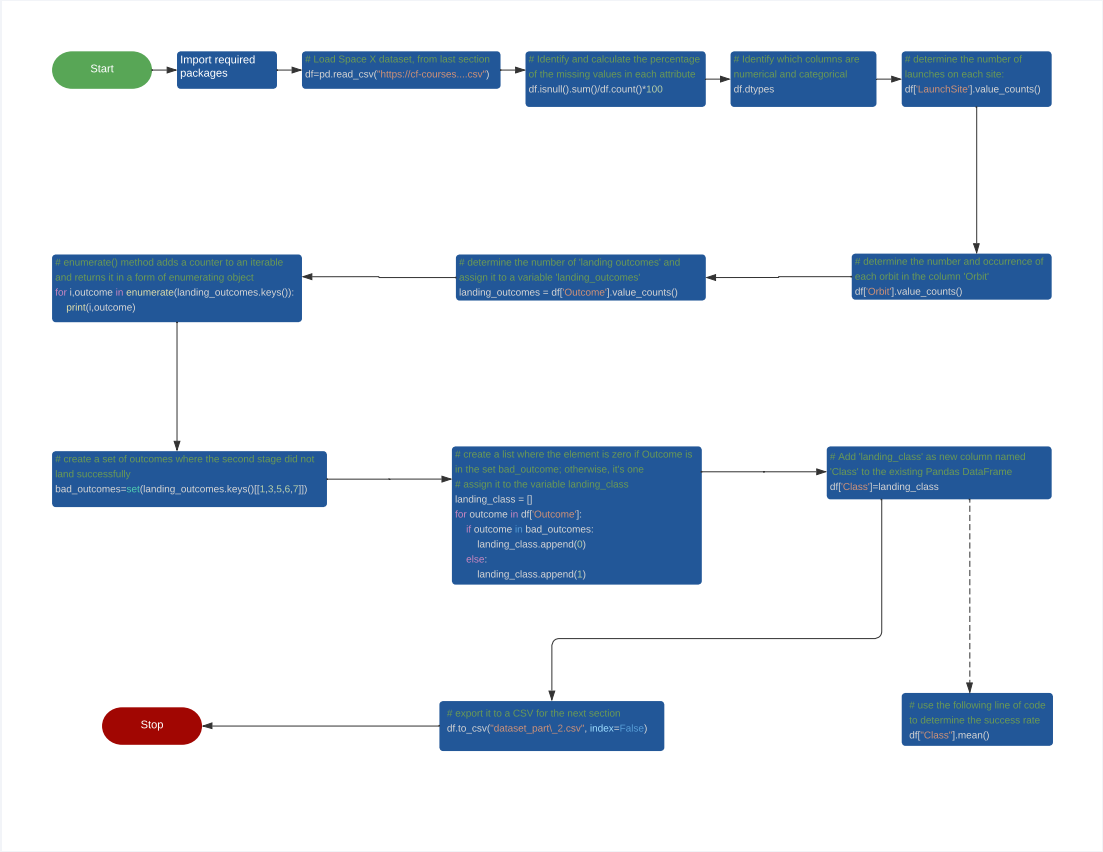
Data Collection - Scraping

- Extracted Falcon 9 launch records HTML table from Wikipedia, parse the table, and convert it into a Pandas data frame. See flowchart on the right for step-by-step details.
- Here is the link to the completed web scraping notebook in GitHub



Data Wrangling

- Performed Exploratory Data Analysis (EDA) to find patterns in the data and determine what would be the label for training supervised models. In the data set, there are several different cases where the booster did not land successfully. Sometimes a landing was attempted but failed due to an accident. Converted those outcomes into Training Labels with `1` means the booster successfully landed `0` means it was unsuccessful. See flowchart on the right for step-by-step details.
- [Here is the link to completed data wrangling notebook in Github]



EDA with Data Visualization

The following charts were plotted:

- **FlightNumber (continuous launch attempts) vs. PayloadMass and overlaid the outcome of the launch**: as the flight number increases, the first stage is more likely to land successfully. The payload mass is also important as the more massive the payload, the less likely the first stage will return.
- **FlightNumber vs LaunchSite and overlaid the outcome of the launch**: as the flight number increases, the first stage is more likely to land successfully
- **Payload and LaunchSite**: in the VAFB-SLC launchsite there are no rockets launched for heavypayload mass(greater than 10000).
- **Success rate of each orbit type**: ES-L1, GEO, HEO, SSO, and VLEO are orbits that have high success rate
- **FlightNumber and Orbit type**: in the LEO orbit the Success appears related to the number of flights. There seems to be no relationship between flight number when in GTO orbit.
- **Payload and Orbit type**: with heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS. For GTO we cannot distinguish this well as both positive and negative landing rate are there.
- **Launch success yearly trend**: the success rate since 2013 kept increasing until 2020

Here is the link to the completed EDA with data visualization notebook in Github

EDA with SQL

SQL queries performed:

- Display the names of the unique launch sites in the space mission: select distinct launch_site from spacextbl
- Display 5 records where launch sites begin with the string 'CCA': SELECT * FROM spacextbl WHERE launch_site LIKE 'CCA%' LIMIT 5
- Display the total payload mass carried by boosters launched by NASA (CRS): SELECT SUM(payload_mass__kg_) totalpayloadnasacrs FROM spacextbl WHERE customer LIKE 'NASA (CRS)'
- Display average payload mass carried by booster version F9 v1.1: SELECT avg(payload_mass__kg_) as average_payload from SPACEXTBL where (booster_version) = 'F9 v1.1'
- List the date when the first succesful landing outcome in ground pad was acheived. Because the date as formatted (dd-mm-yyyy) is not correct, we have to reformat it as yyyy-mm-dd:
  SELECT min(substr(Date, 7, 4) || '-' || substr(Date, 4, 2) || '-' || substr(Date, 1, 2)) as NewDate
  FROM SPACEXTBL
  where "Landing _Outcome" = 'Success (ground pad)'
- List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000:
  SELECT BOOSTER_VERSION AS BoosterName, Payload, "Landing _Outcome", PAYLOAD_MASS__KG_
  FROM SPACEXTBL
  WHERE "Landing _Outcome" ='Success (drone ship)'
  AND PAYLOAD_MASS__KG_ between 4000 and 6000

More the next page

EDA with SQL

SQL queries performed:

- List the total number of successful and failure mission outcomes: SELECT mission_outcome, COUNT(mission_outcome) Count from SPACEXTBL GROUP BY mission_outcome
- List the names of the booster_versions which have carried the maximum payload mass. Use a subquery:

```
SELECT booster_version
FROM spacextbl
WHERE payload_mass__kg_ = (SELECT MAX(payload_mass__kg_) FROM spacextbl)
```

- List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015. Note: SQLLite does not support monthnames. So you need to use substr(Date, 4, 2) as month to get the months and substr(Date,7,4)='2015' for year:

```
SELECT substr(Date, 4, 2) as month, "Landing _Outcome" , BOOSTER_VERSION, LAUNCH_SITE
FROM SPACEXTBL
WHERE "Landing _Outcome" = 'Failure (drone ship)'
AND substr(Date,7,4)='2015'
```

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

```
SELECT "Landing _Outcome", COUNT("Landing _Outcome") AS COUNT_LAUNCHES, (substr(Date, 7, 4) || '-' || substr(Date, 4, 2) || '-' || substr(Date,
1, 2)) as myDATE
FROM SPACEXTBL
WHERE myDATE BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY "Landing _Outcome"
ORDER BY COUNT_LAUNCHES DESC;
```

[Here is the link to the completed EDA SQL/SQLite notebook in GitHub]

Build an Interactive Map with Folium

Created map objects (see below) and added them to a Folium map:

- Created **markers** to indicate points like launch sites
- Created **circles** to highlight areas around a coordinates such as NASA Johnson Space Center
- Used **lines** to indicate distances between two coordinates
- Created **marker clusters** to indicate group of events in each coordinate such as launches in a launch site

Here is the link to the completed interactive map with Folium map notebook in GitHub

Build a Dashboard with Plotly Dash

This dashboard application contains input components such as a dropdown list and a range slider to interact with a pie chart and a scatter point chart

- **Launch Site Drop-down Input Component to let us select different launch sites**: It shows which one has the largest success count. If selecting one specific site, it shows detailed success rate (class=0 vs. class=1)
- **Callback function**: to render success-pie-chart based on selected site dropdown; it will get the selected launch site from site-dropdown and render a pie chart visualizing launch success counts
- **Range Slider to Select Payload**: to easily select different payload range and see if we can identify some visual patterns (if variable payload is correlated to mission outcome)
- **Callback function to render the success-payload-scatter-chart scatter plot**: it shows how payload may be correlated with mission outcomes for selected site(s). Color-label the Booster version on each scatter point to observe mission outcomes with different boosters

Here is the link to the completed Plotly Dash lab in GitHub

Here is the link to the spaceX launch data in GitHub

Predictive Analysis (Classification)

Summary:

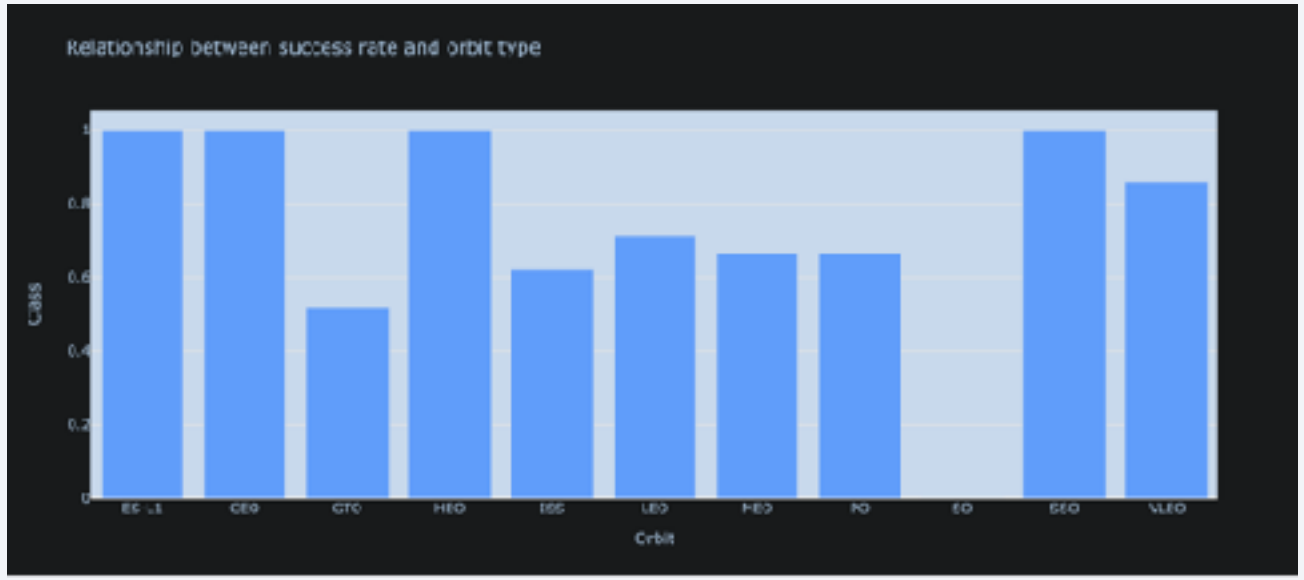
1. Imported libraries, created a function to plot the confusion matrix, and loaded the data frame
2. Created a NumPy array (from column Class in data) and assigned it to the variable Y
3. Standardize the data in X and reassigned it to the variable X
4. Used the function train_test_split to split the data X and Y into training and test(Settted the parameter test_size to 0.2 and random_state to 2)
5. **Created a logistic regression object**
6. **Create a GridSearchCV object logreg_cv with cv = 10.**
7. **Fit the object to find the best parameters from the dictionary parameters**
8. **Outputted the GridSearchCV object for logistic regression: displayed the best parameters using the data attribute best_params_ and the accuracy on the validation data using the data attribute best_score_**
9. **Calculated the accuracy on the test data using the method score: false positives was a major problem**
10. **Repeated steps 5 through 9 for SVM, Decision Tree Classifier and KNN**
11. Found the method that performed best: The best model was DecisionTree with a score of 0.8875. Its best parameters were : {'criterion': 'gini', 'max_depth': 14, 'max_features': 'auto', 'min_samples_leaf': 1, 'min_samples_split': 2, 'splitter': 'best'}. Its TestAccuracy was 0.88889

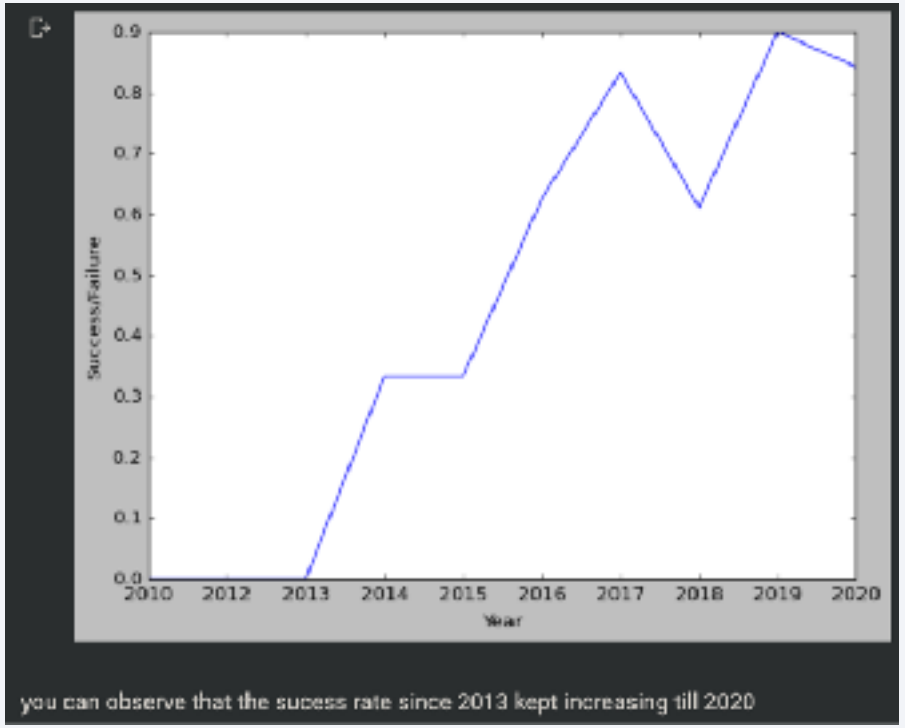
Here is the link to the completed predictive analysis lab in GitHub

Results - Exploratory data analysis results

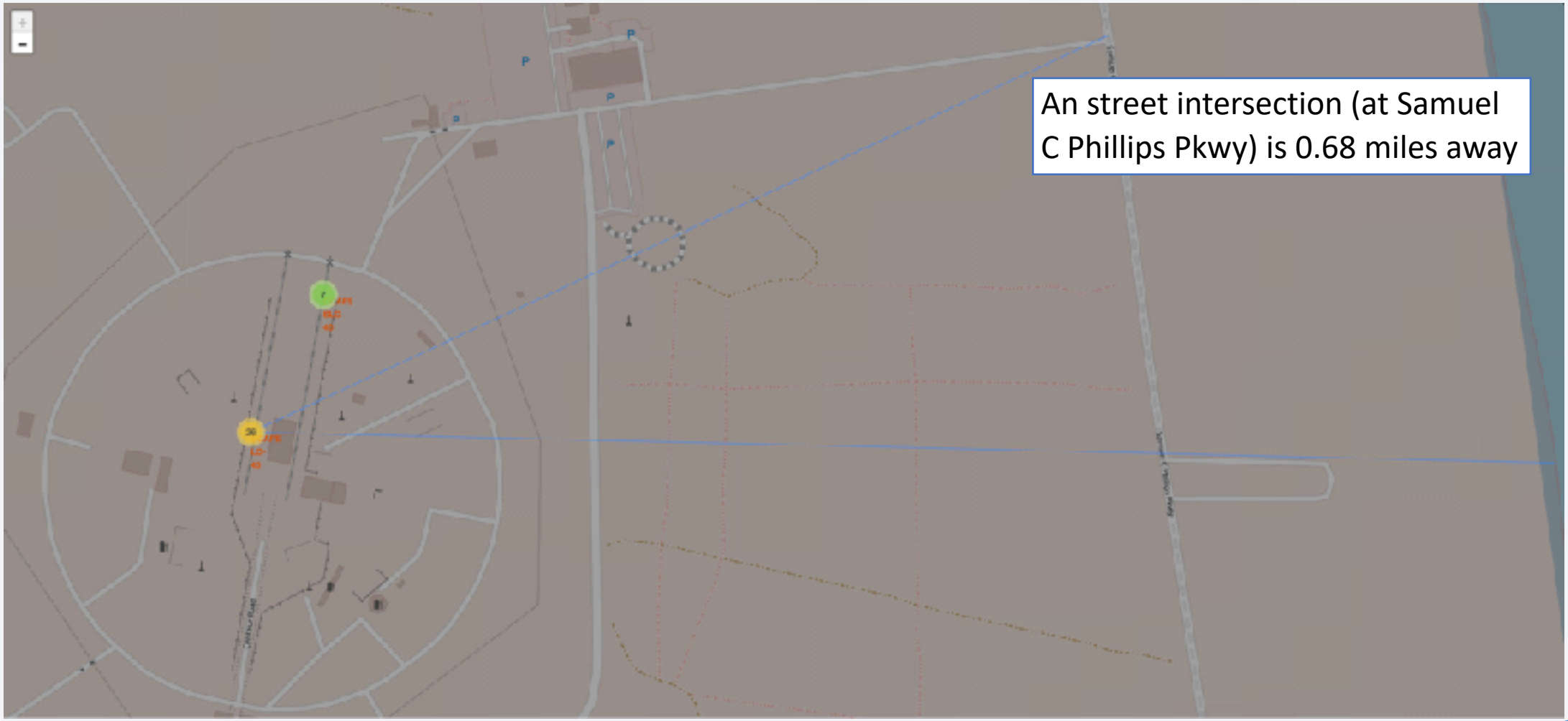
Using SQL, this is the count of successful landing_outcomes between 04-06-2010 and 20-03-2017 in descending order

Landing _Outcome	COUNT_LAUNCHES	DATE
No attempt	10	2017-03-16
Failure (drone ship)	5	2016-06-15
Success (drone ship)	5	2017-01-14
Controlled (ocean)	3	2015-02-11
Success (ground pad)	3	2017-02-19
Failure (parachute)	2	2010-12-08
Uncontrolled (ocean)	2	2014-09-21
Precluded (drone ship)	1	2015-06-28



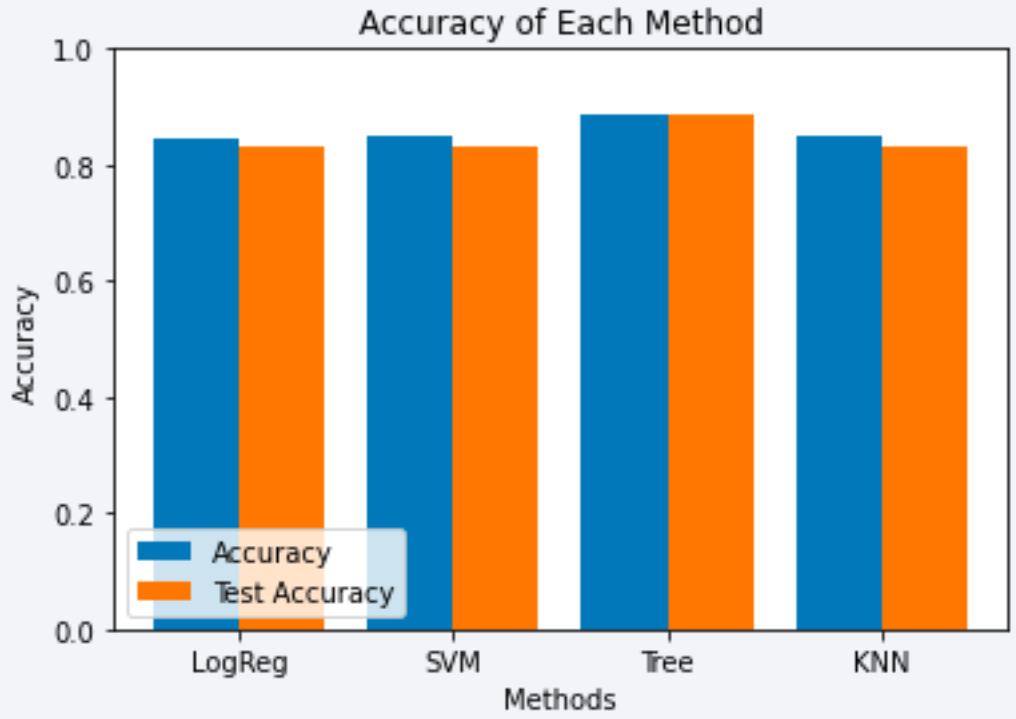


Results - Interactive analytics demo (in screenshot)



Results - Predictive analysis results

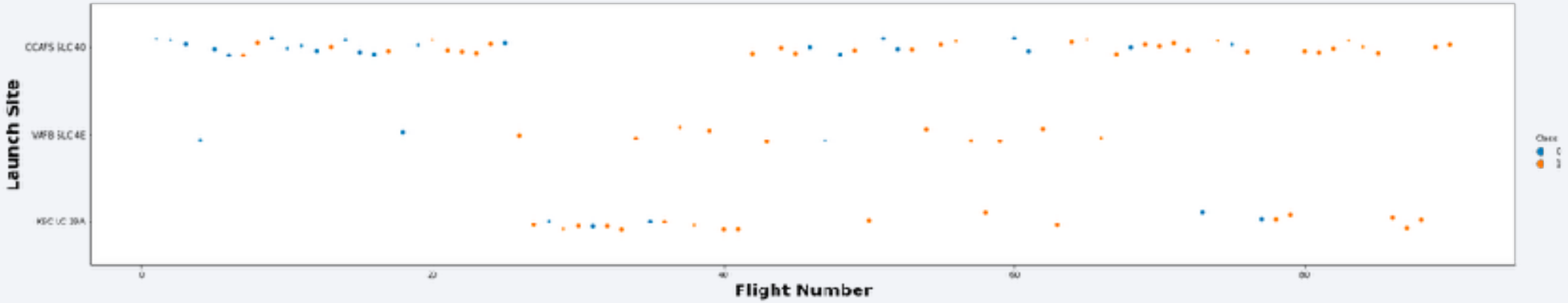
- The best model was the Decision Tree with a score of 0.8875, and an accuracy of over 88.89% for test data



Section 2

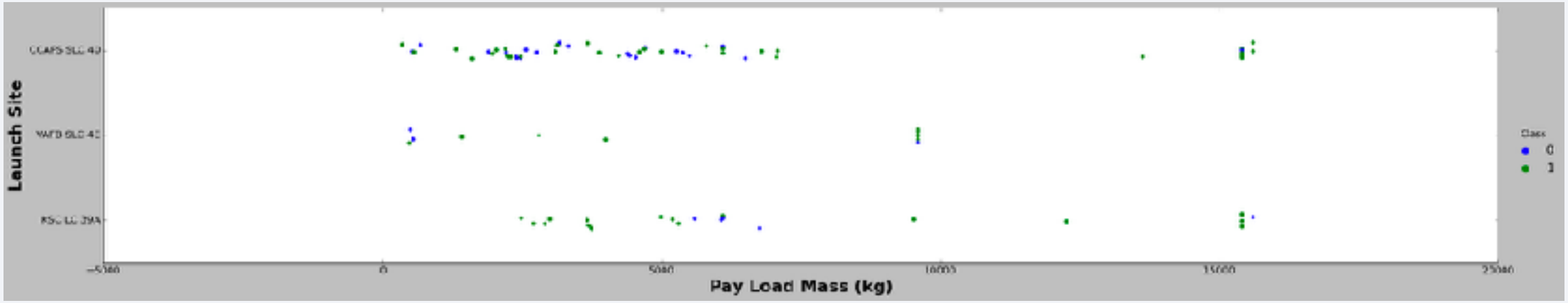
Insights drawn from EDA

Flight Number vs. Launch Site



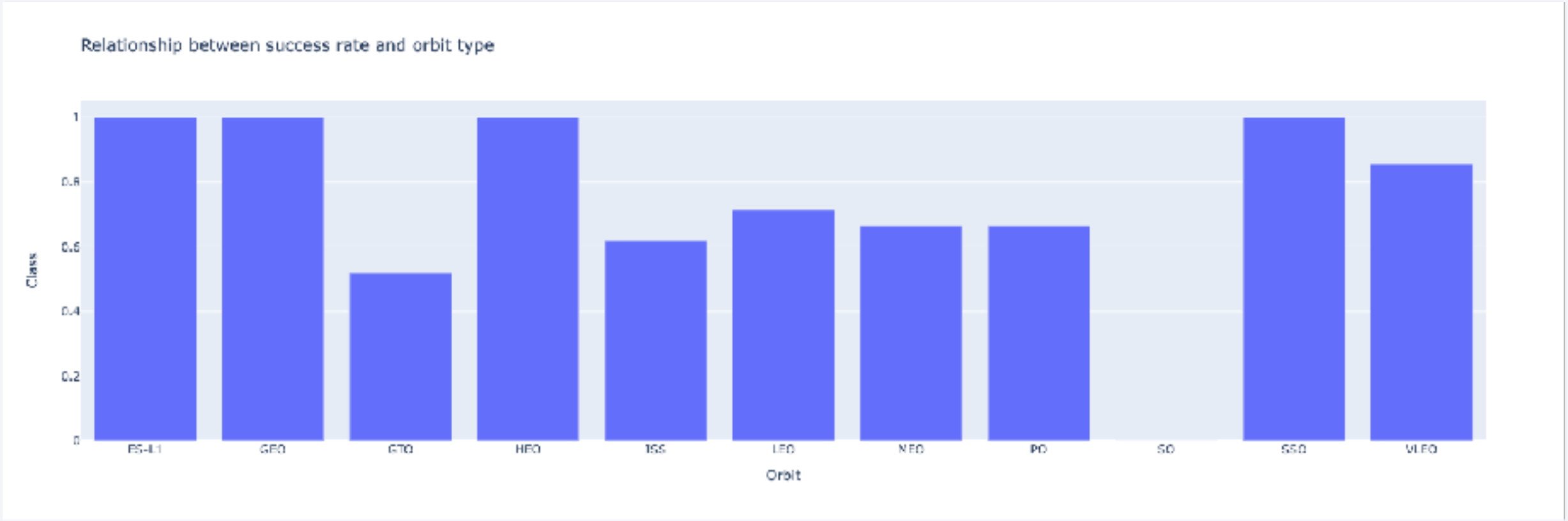
Explanation: as the number of flights increase on each launch sites, the first stage is more likely to land successfully

Payload vs. Launch Site



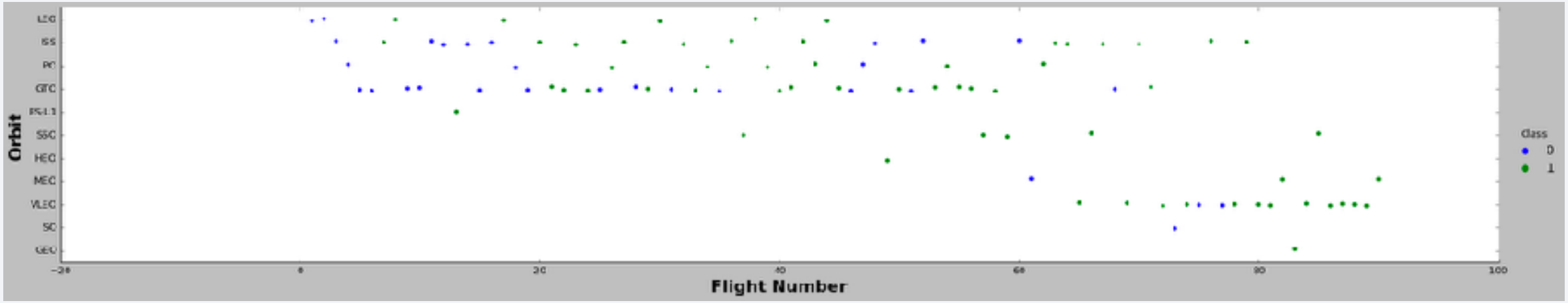
Explanation: for the VAFB-SLC launch site, there are no rockets launched for heavy payload mass (greater than 10000)

Success Rate vs. Orbit Type



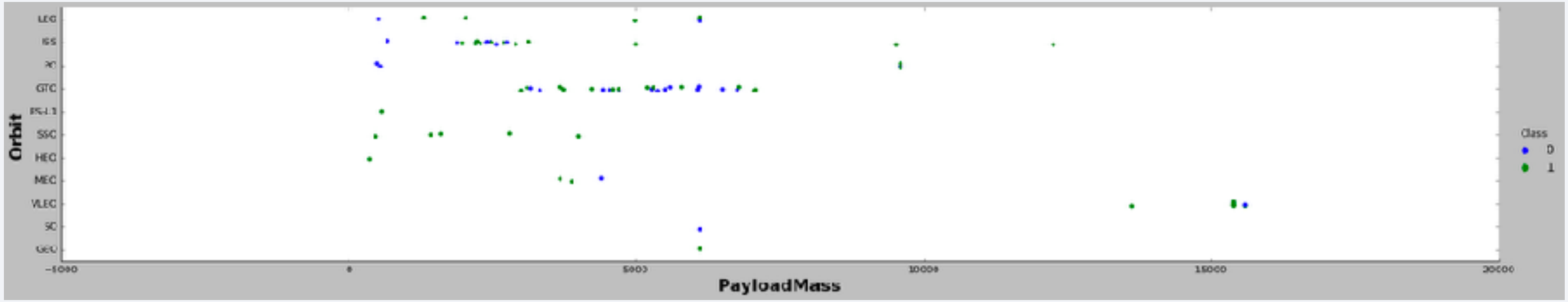
- The orbits that have high success rate are ES-L1, GEO, HEO, SSO, and VLEO

Flight Number vs. Orbit Type



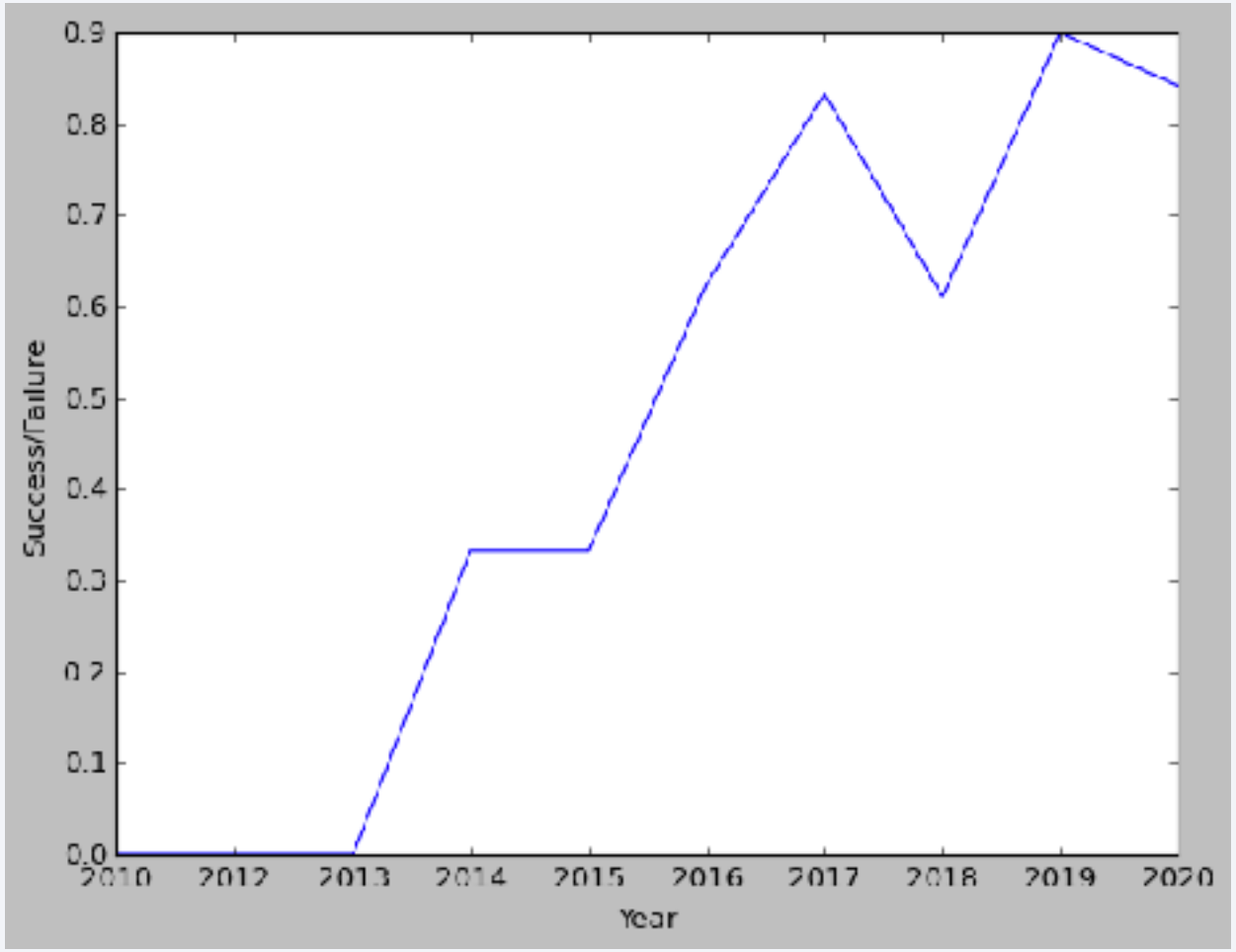
- FlightNumber and Orbit type: in the LEO orbit the Success appears related to the number of flights. There seems to be no relationship between flight number when in GTO orbit.

Payload vs. Orbit Type



- Payload and Orbit type: with heavy payloads the successful landing or positive landing rate are more for Polar, LEO and ISS. For GTO we cannot distinguish this well as both positive and negative landing rate are there.

Launch Success Yearly Trend



- Launch success yearly trend: the success rate since 2013 kept increasing until 2020

All Launch Site Names

Find the names of the unique launch sites

```
SELECT DISTINCT LAUNCH_SITE FROM SPACEXTBL
```

Query result:

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

There are four unique launch sites in the database:

Launch Site Names Begin with 'CCA'

Find 5 records where launch sites begin with `CCA`

```
SELECT *
FROM SPACEXTBL
WHERE LAUNCH_SITE LIKE 'CCA%'
LIMIT 5
```

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
08-12-2010	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)
22-05-2012	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
01-03-2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

This query will stop running when it finds 5 results of launch sites that begin with `CCA`

Total Payload Mass

Calculate the total payload carried by boosters from NASA (CRS)

```
SELECT SUM(payload_mass__kg_) totalpayloadnasacrs
FROM spacextbl
WHERE customer LIKE 'NASA (CRS)'
```

Result:

totalpayloadnasacrs
45596

Explanation:

This query will use the ‘SQL SUM aggregate function’ to sum all the values in column ‘payload_mass__kg_’ which is located (FROM) in the table ‘spacextbl’. In addition the WHERE function will filter in the values WHERE the column ‘customer’ is LIKE 'NASA (CRS)’. The SQL LIKE clause was used to compare and filter values.

Average Payload Mass by F9 v1.1

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

```
SELECT avg(payload_mass__kg_) as average_payload
FROM SPACEXTBL
WHERE (booster_version) = 'F9 v1.1'
```

Result:

average_payload
2928.4

Explanation: The AVG() function returned the average value of column payload_mass__kg_ which was located in the table (FROM) SPACEXTBL. The WHERE clause used the equal operator (=) to check and filter in the values in the “booster_version” column that were equal to 'F9 v1.1'

First Successful Ground Landing Date

Find the dates of the first successful landing outcome on ground pad
In SQLite, because the date as formatted (dd-mm-yyyy) was not correct, I had to reformat it as yyyy-mm-dd

```
SELECT min(substr(Date, 7, 4) || '-' || substr(Date, 4, 2) || '-' || substr(Date, 1, 2)) as NewDate
FROM SPACEXTBL
WHERE "Landing _Outcome" = 'Success (ground pad)'
```

Result: NewDate ==> this is the minimum date
2015-12-22

Explanation: after reformatting the date (see orange text), SQLite was able to find the smallest value in the 'Date' column with the MIN() function. The WHERE clause used the equal operator (=) to check and filter in the values in the "Landing _Outcome" column that were equal to 'Success (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

```
SELECT BOOSTER_VERSION AS BoosterName, Payload, "Landing _Outcome”, PAYLOAD_MASS__KG_
FROM SPACEXTBL
WHERE  "Landing _Outcome" = 'Success (drone ship)'
AND PAYLOAD_MASS__KG_ between 4000 and 6000
```

Result:

BoosterName	Payload	Landing _Outcome	PAYLOAD_MASS__KG_
F9 FT B1022	JCSAT-14	Success (drone ship)	4696
F9 FT B1026	JCSAT-16	Success (drone ship)	4600
F9 FT B1021.2	SES-10	Success (drone ship)	5300
F9 FT B1031.2	SES-11 / EchoStar 105	Success (drone ship)	5200

Explanation: selected 4 columns FROM the table ‘SPACEXTBL’, and changed the name of the first column to ‘BoosterName’. Using the WHERE clause, filtered in those values WHERE "Landing _Outcome" = 'Success (drone ship)’ AND The PAYLOAD_MASS__KG_ was **between** 4000 **and** 6000.

Total Number of Successful and Failure Mission Outcomes

Calculate the total number of successful and failure mission outcomes

```
SELECT mission_outcome, COUNT(mission_outcome) Count
FROM SPACEXTBL
GROUP BY mission_outcome
```

Result:

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

Explanation: this query counted the mission outcome and grouped it by its values. The mission outcome had 4 values: 3 were success and 1 was failure. The count of all success values was 100, and the count of failure values was 1. This simple query answers the question.

Boosters Carried Maximum Payload

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

```
SELECT booster_version
FROM spacextbl
WHERE payload_mass__kg_ =
  (SELECT MAX(payload_mass__kg_) FROM spacextbl)
```

Explanation: this query and subquery (also known as inner query or nested query) performed two operations. First, the subquery found the MAX payload mass FROM spacextbl. Then, the query used the result of the subquery to get the list of booster version with the MAX payload

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

2015 Launch Records

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

Note: SQLite does not support monthnames. So you need to use substr(Date, 4, 2) as month to get the months and substr(Date,7,4)='2015' for year.

```
SELECT "Landing _Outcome" , BOOSTER_VERSION, LAUNCH_SITE
FROM SPACEXTBL
WHERE "Landing _Outcome" = 'Failure (drone ship)'
AND substr(Date,7,4) = '2015'
```

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

Result: After selecting three columns from the SPACEXTBL table, we filtered in those that met the given criteria in the WHERE clause. The SUBSTR() function extracted the year substring from the string in date column. This extraction started at position 7 and continued until it reached a length of 4). Unlike Python, in SQL the first character has an index of 1

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

```
SELECT "Landing _Outcome”
    , COUNT("Landing _Outcome") AS COUNT_LAUNCHES
    , (substr(Date, 7, 4) || '-' || substr(Date, 4, 2) || '-' || substr(Date, 1, 2)) as myDATE
FROM SPACEXTBL
WHERE myDATE BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY "Landing _Outcome"
ORDER BY COUNT_LAUNCHES DESC;
```

Landing _Outcome	COUNT_LAUNCHES	myDATE
No attempt	10	2017-03-16
Failure (drone ship)	5	2016-06-15
Success (drone ship)	5	2017-01-14
Controlled (ocean)	3	2015-02-11
Success (ground pad)	3	2017-02-19
Failure (parachute)	2	2010-12-08
Uncontrolled (ocean)	2	2014-09-21
Precluded (drone ship)	1	2016-06-28

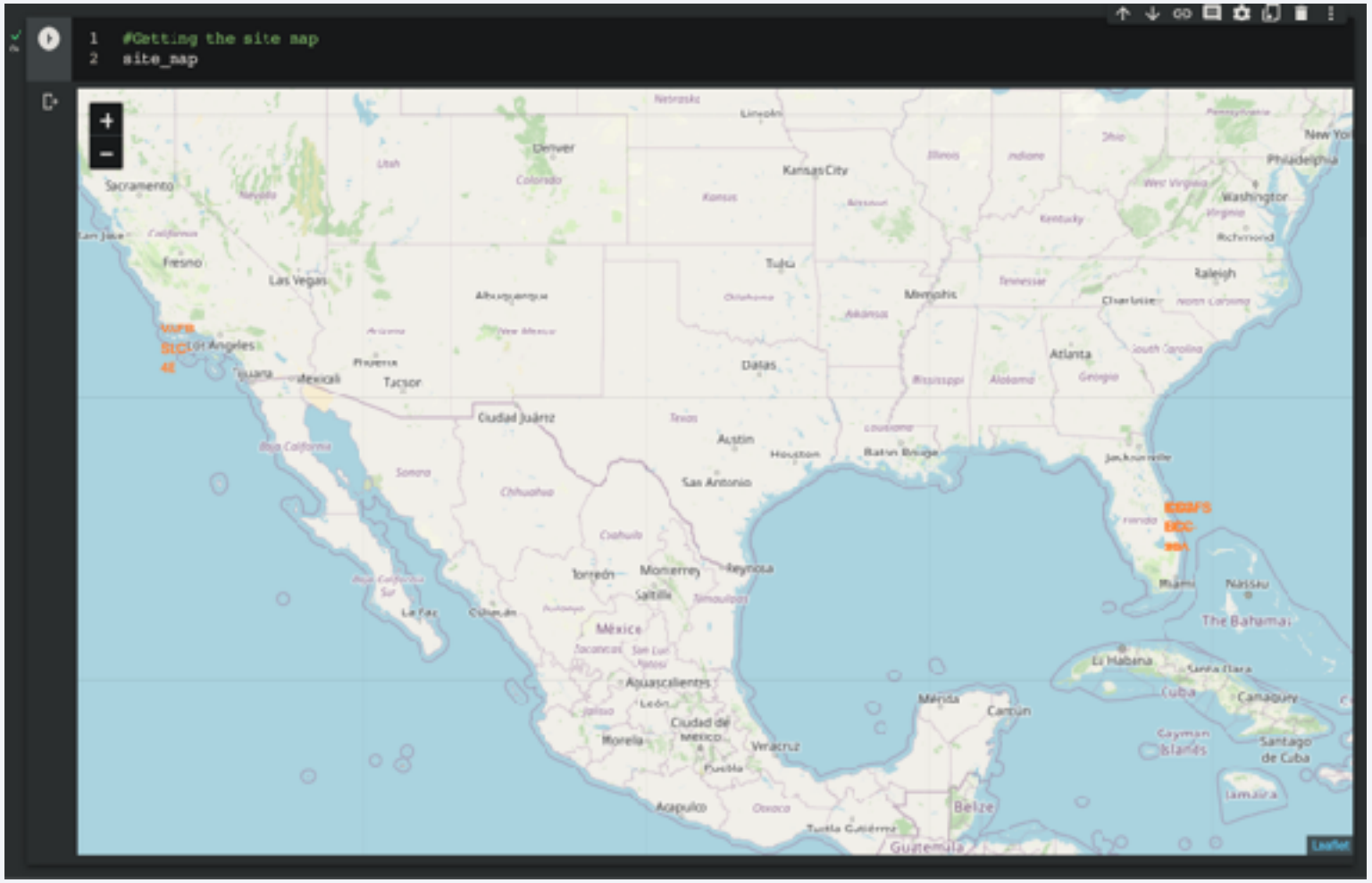
Explanation: Selected and counting the landing outcome from the table, created a column myDate with the dates in the format YYYY-MM-DD. We will use this column to filter in the requested dates. To count the landing outcomes by actual outcomes, we used the GROUP BY statement to group rows that have the same values. Then we ordered the count in descending order

Section 3

Launch Sites Proximities Analysis

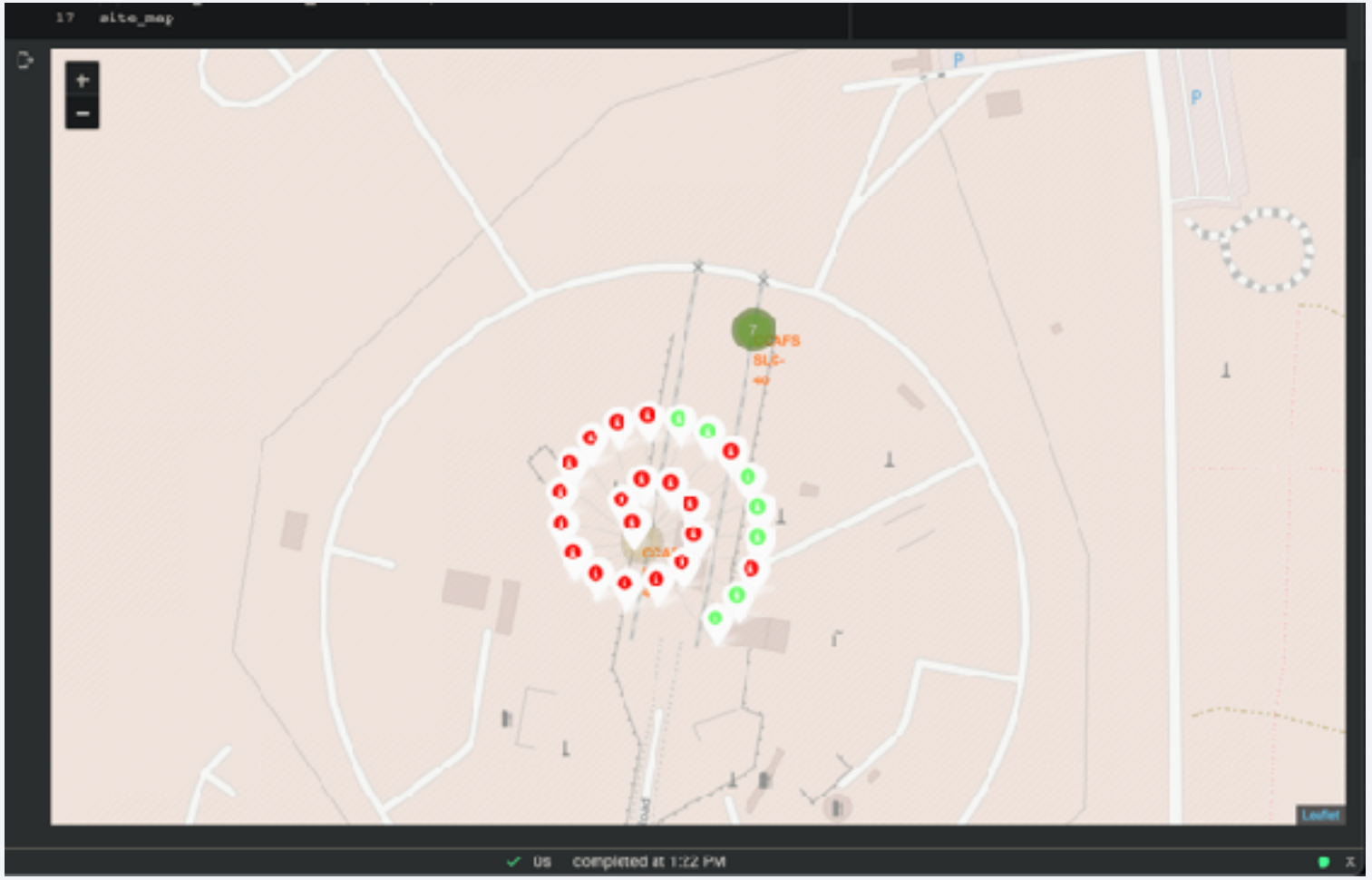
Location of all launch sites

We can see all the launch sites with this screenshot. There was no need to create a global map since all launch sites are in the USA. In fact, they are in both coasts (in the states of California and Florida)



Color-labeled launch outcomes

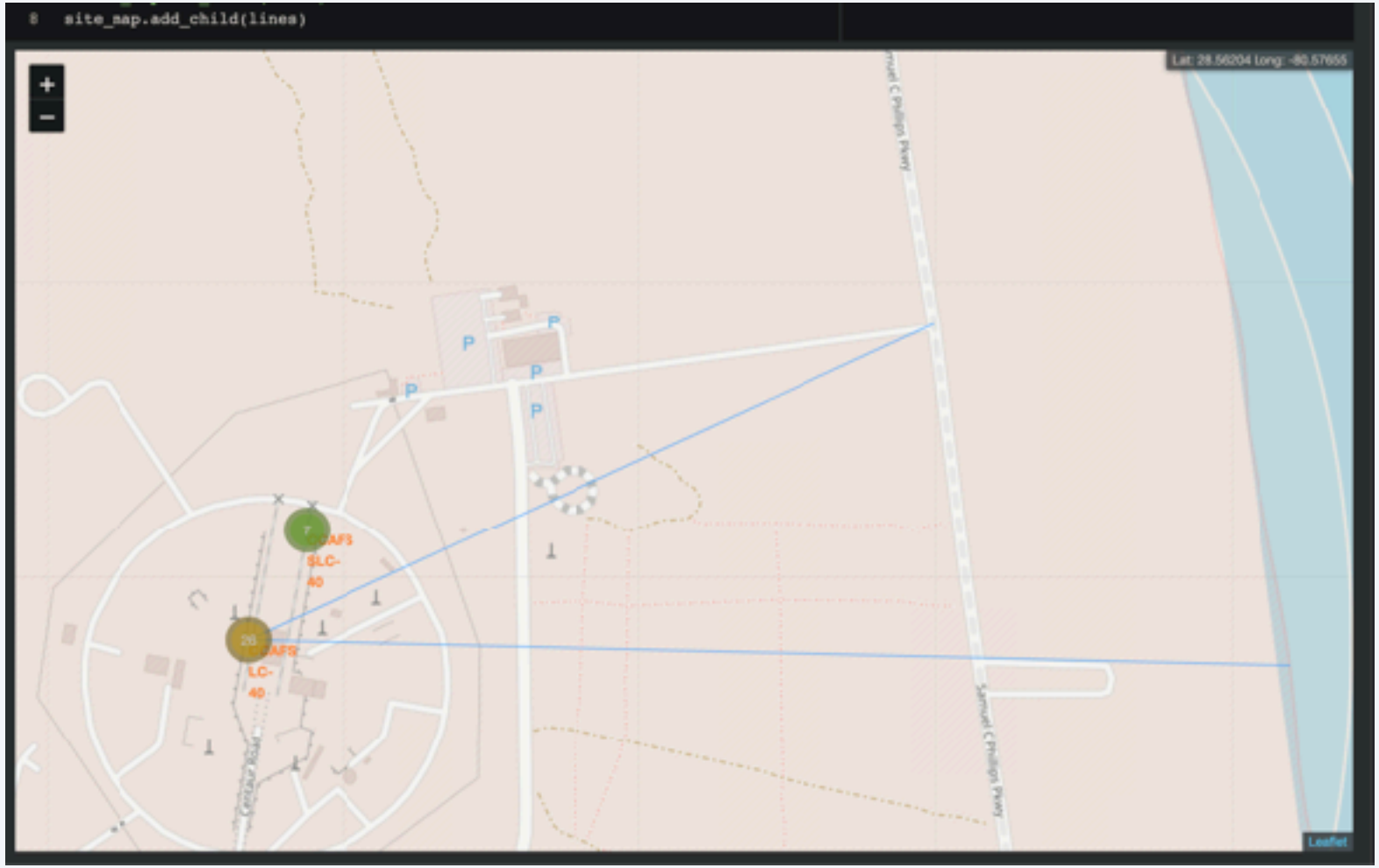
The color-labeled launch outcomes on the Folium map shows a green icon if the launch succeeded; and a red icon if it failed



Folium Map - launch site and its proximities to railway, highway and coastline

This launch site is close to highways and the coastline.

An street intersection (at Samuel C Phillips Pkwy) is 0.68 miles away.

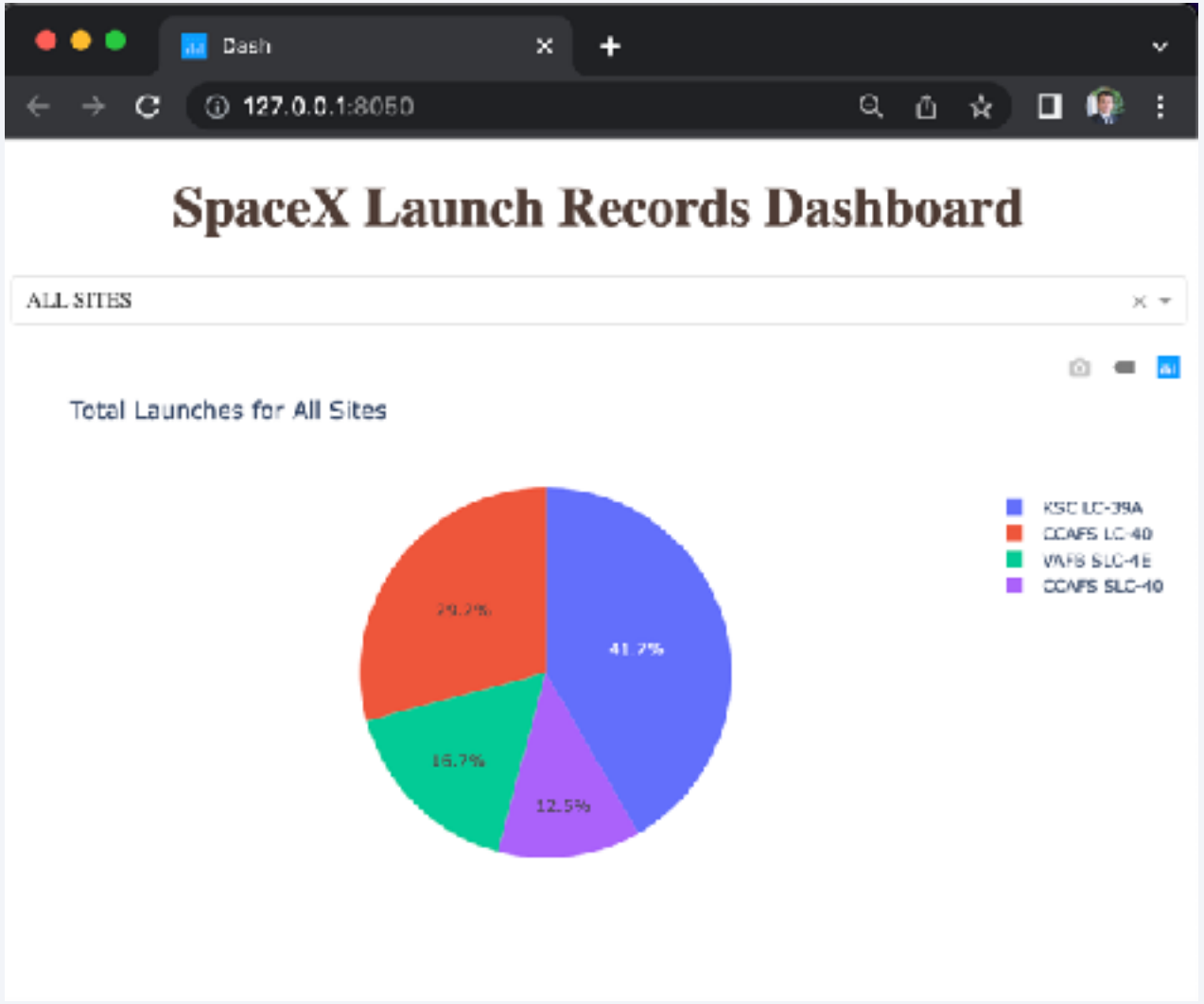


Section 4

Build a Dashboard with Plotly Dash

Dashboard - launch success count for all sites, in a piechart

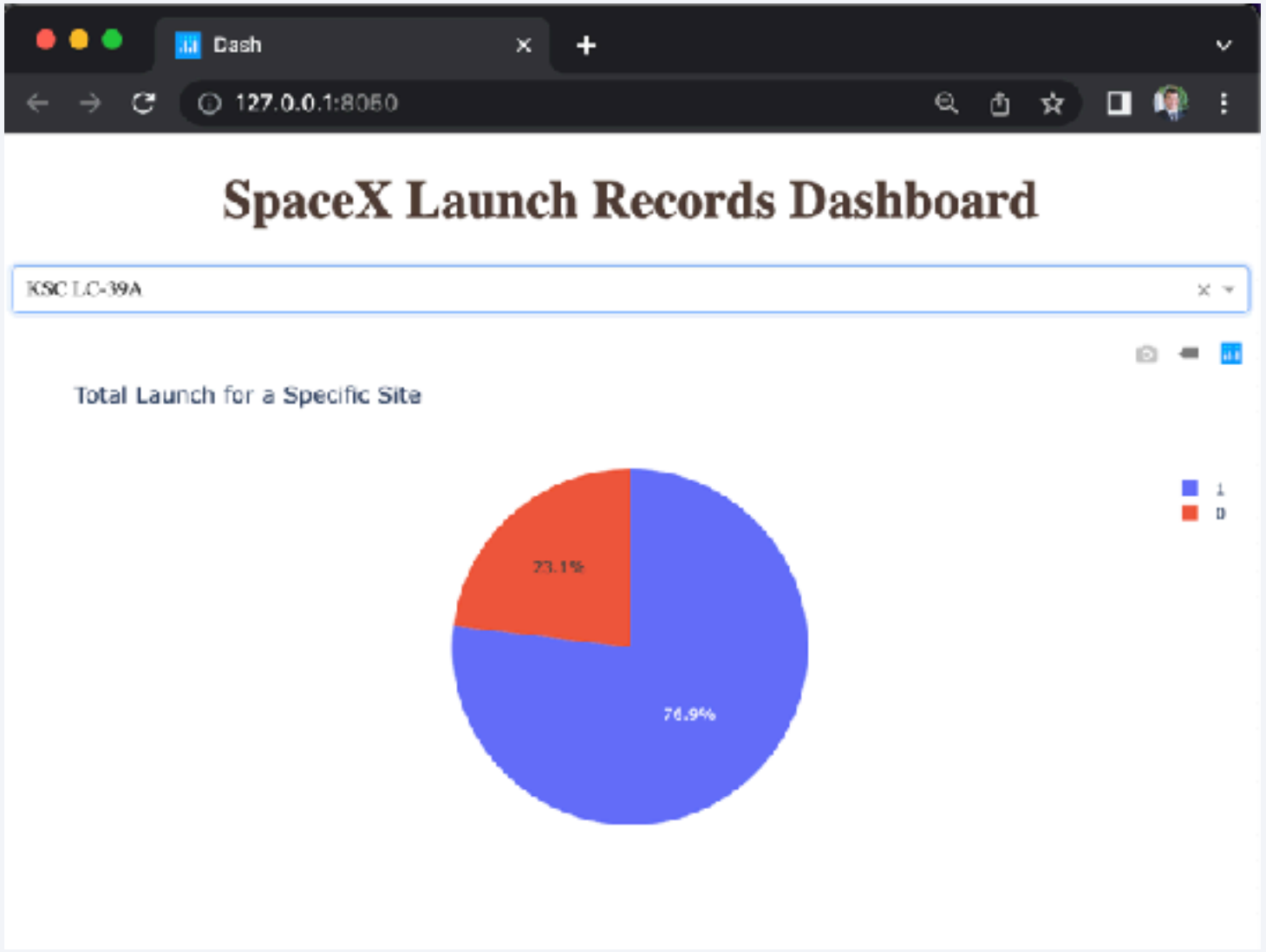
- The launch site KSC LC-39A had the most successful launches across all four launch sites. 41.7% of all the successful launches were launch from here
- The next one was CCAFS LC-40 with 29.2%



Dashboard - piechart for the launch site with highest launch success ratio

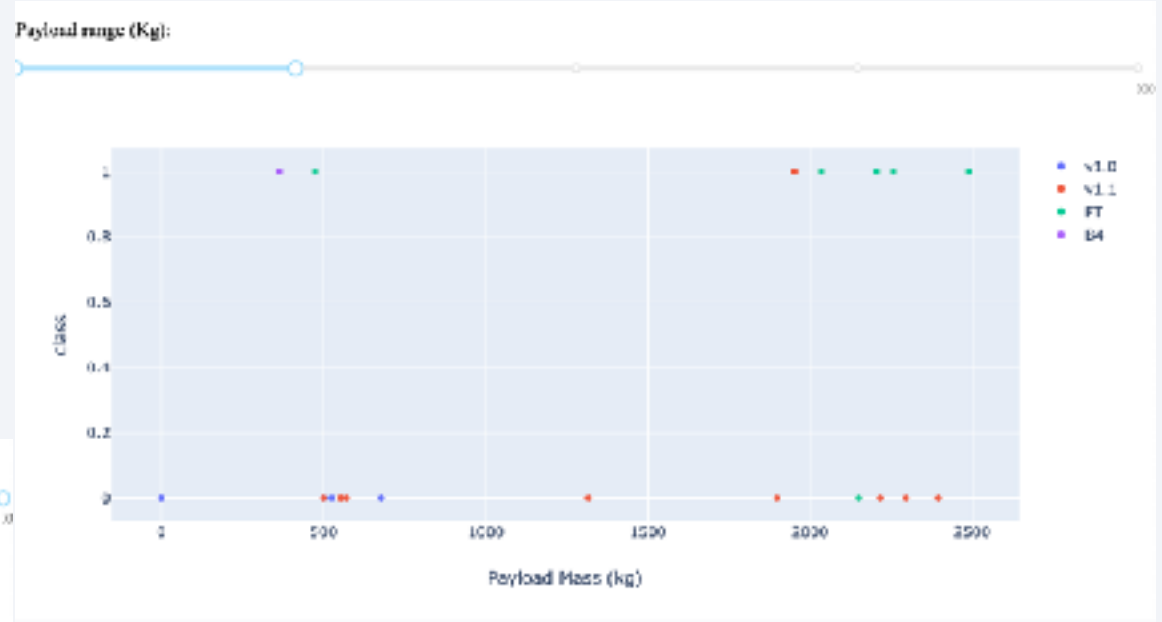
Launch Site KSC LC-39A

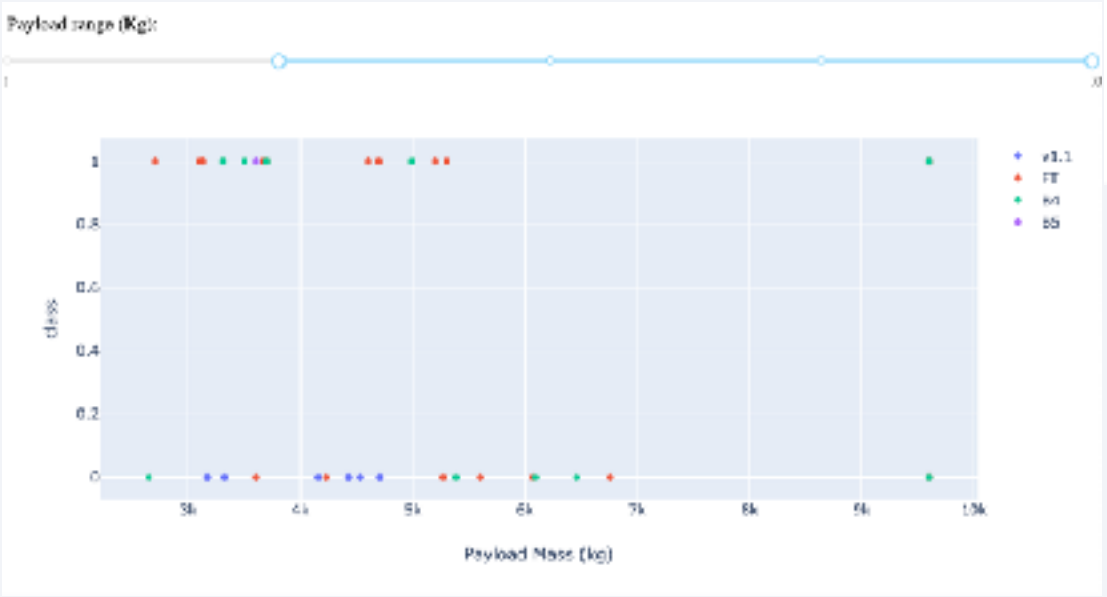
- 76.9% of al the launches form this site were successful
- This launch site (KSC LC-39A) had the most successful launches across all four launch sites with 41.7% of all the successful launches



Dashboard - Payload vs. Launch Outcome scatter plot for all sites

In the payload range of over 2,500 kilograms for all sites, almost 1/2 of all launches failed and thus almost 1/2 of all succeeded (see below)





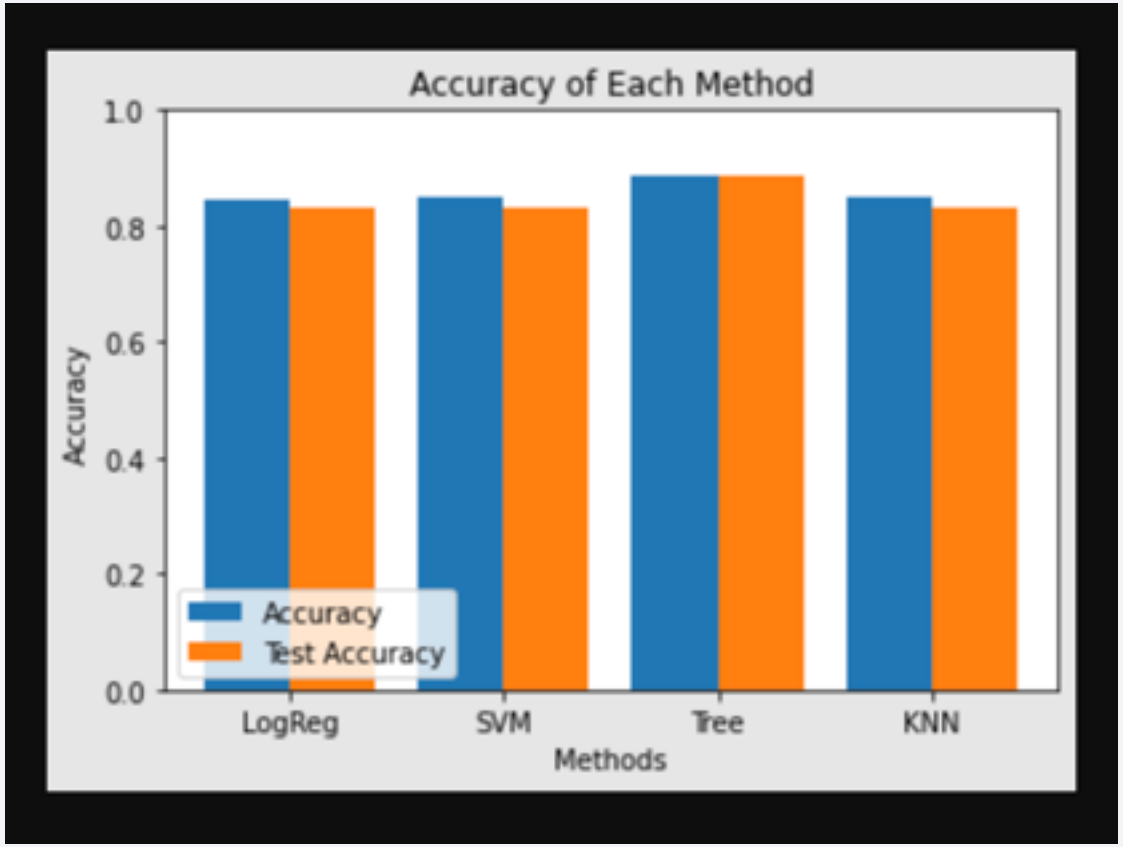
In the payload range of under 2,500 kilograms for all sites, 2/3 of all launches failed and thus 1/3 succeeded (see above)

Section 5

Predictive Analysis (Classification)

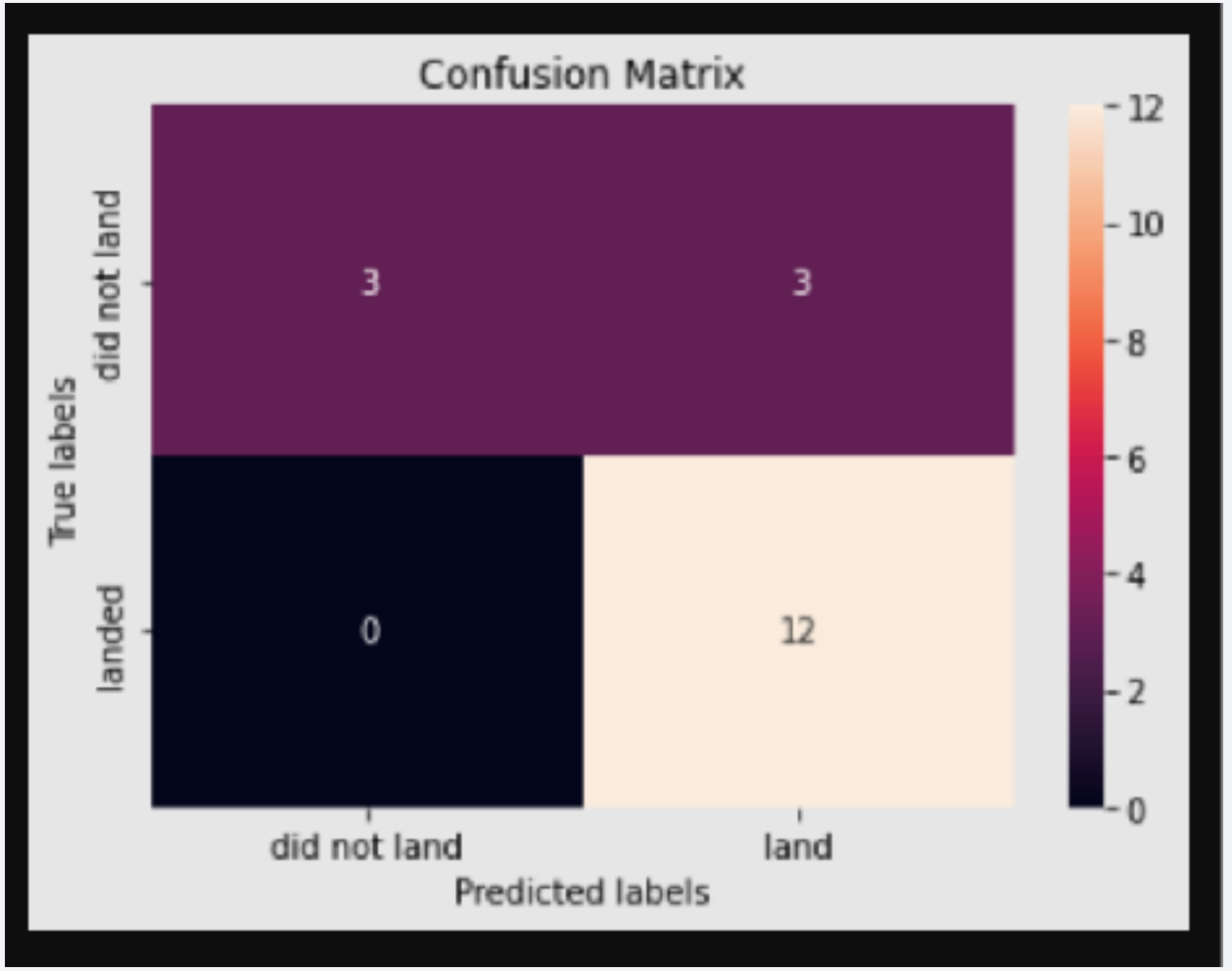
Classification Accuracy

- Four classification models were tested
- The model with the highest classification accuracy was the Decision Tree Classifier with an Accuracy of 0.8875 and a TestAccuracy of 0.88889



Confusion Matrix

- This is the Confusion Matrix for the Decision Tree Classifier
- It shows that the classifier can distinguish between the different classes
- A major problem is false positives: 3 predicted labels classified as landed successfully, did not land as such

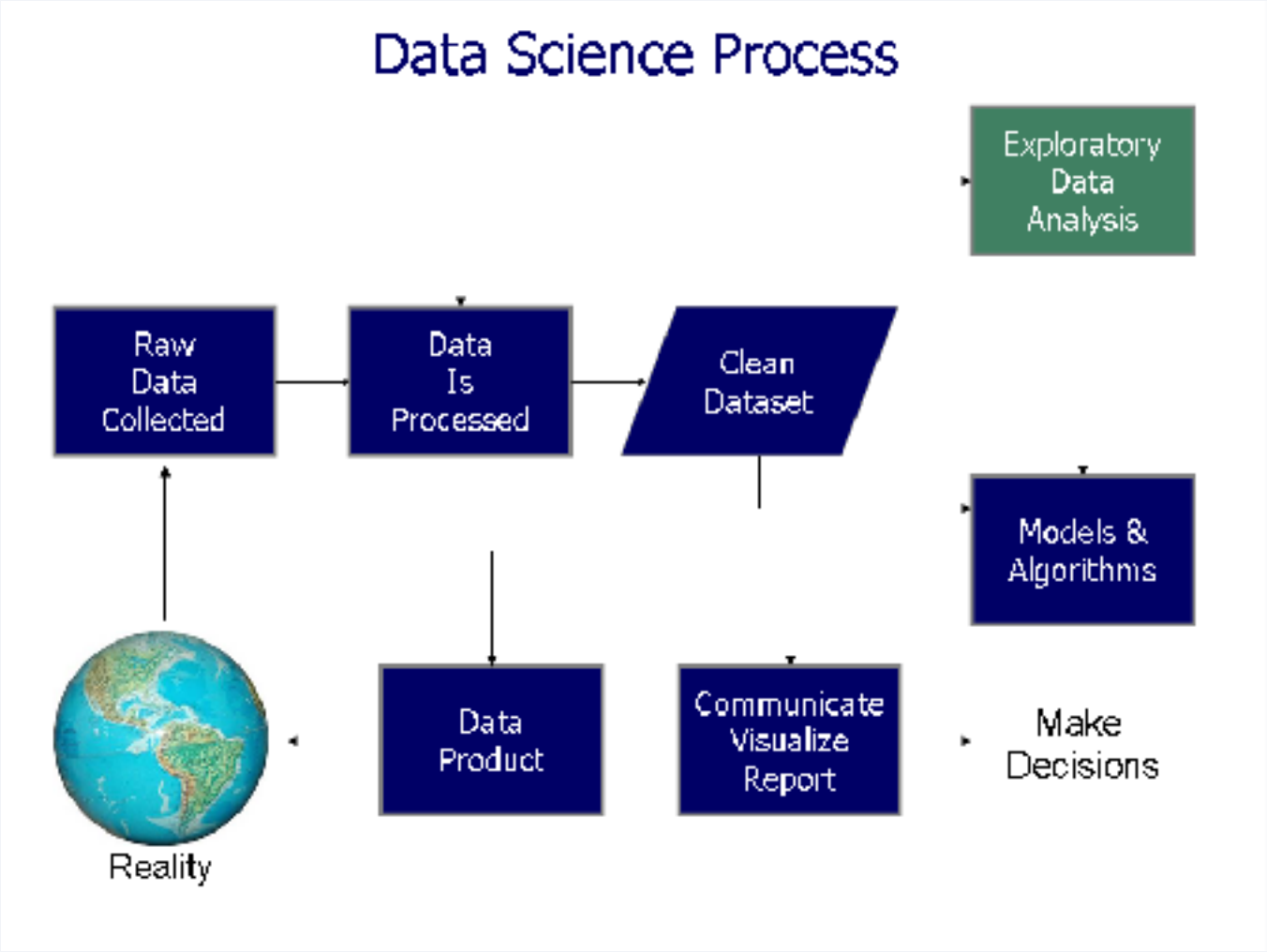


Conclusions

- The launch success since 2013 kept increasing until 2020
- The best model was the Decision Tree with a score of 0.8875, and an accuracy of over 88.89% for test data
- As the number of flights increase on each launch sites, the first stage is more likely to land successfully
- For the VAFB-SLC launch site, there are no rockets launched for heavy payload mass (greater than 10000)
- The orbits that have high success rate are ES-L1, GEO, HEO, SSO, and VLEO
- FlightNumber and Orbit type: in the LEO orbit the success appears related to the number of flights. In GTO orbit, there seems to be no relationship between flight number
- Payload and Orbit type: with heavy payloads the successful landing or positive landing rate are more for Polar, LEO and ISS. For GTO we cannot distinguish this well as both positive and negative landing rate are there
- The launch site KSC LC-39A had the most successful launches across all four launch sites. 41.7% of all the successful launches were launch from here. The next one was CCAFS LC-40 with 29.2%
- Launch Site KSC LC-39A: 76.9% of al the launches form this site were successful

Appendix

[Link to wikipedia]



Thank you!