IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Henry Norbert Alvarez
March 15, 2024

Outline

Executive Summary

- Summary of methodologies
  - Data Collection with SpaceX API
  - Data Collection with BeautifulSoup (WebScraping)
  - Data Wrangling
  - Exploratory Data Analysis using SQL calculating various statistics and visualizing this data using seaborn / matplotlib / plotly
  - Interactive analytics using Folium for geographical markers and a Dash Plotly web app
  - Machine Learnings predictions using logistic regression, SVM, decision trees and KNN
- Summary of all results
  - Launch success has improved 2013 onwards
  - Polar, LEO and ISS have better landing rate with heavy payloads
  - Most launch sites are near the coast
  - All models performed almost the same on the test set but SVM had higher scores in all metrics (Jaccard, F1 and Accuracy)

Introduction

- **Background**

  SpaceX is a leader in the space industry, with launch services that are less expensive that many of its competitors. It works closely with NASA to deliver supplies and astronauts to the International Space Station (ISS), as well to launch satellites into Earth orbit. It can send these missions because their launches are less expensive in comparison to NASA for example. In 2022, SpaceX charged around $62 million per launch, around $1,200 per pound of payload to reach low-Earth orbit, increasing the cost at $67 million the same year.

- Problems you want to find answers

  - How different variables (payload mass, number of flights, orbits and launch sites) affect landing success
  - What’s conditions does SpaceX need to have the highest success landing rate
  - What’s the best predictive model for successful landings

Section 1

Methodology

Methodology

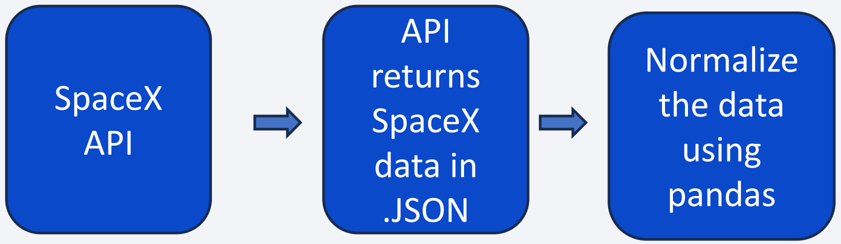
Executive Summary

- Data collection methodology:
  - I used the SpaceX API and web scraping techniques mainly using BeautifulSoup
- Perform data wrangling
  - I filtered the data, handling missing values to prepare the data for analysis and modeling
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - I used classification models, tuning and evaluating each model to find the best model and parameters

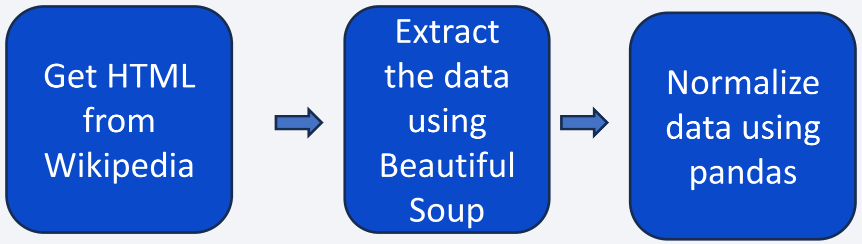
Data Collection

- Data Collection was done using the SpaceX API
- I used Pandas to normalize the data in form of a json result to a pandas dataframe using *pandas.json_normalize()*
- Complementary I used Web Scraping with BeautifulSoup, scraping the Wikipedia page for Falcon 9 data.

SpaceX API



Web Scraping

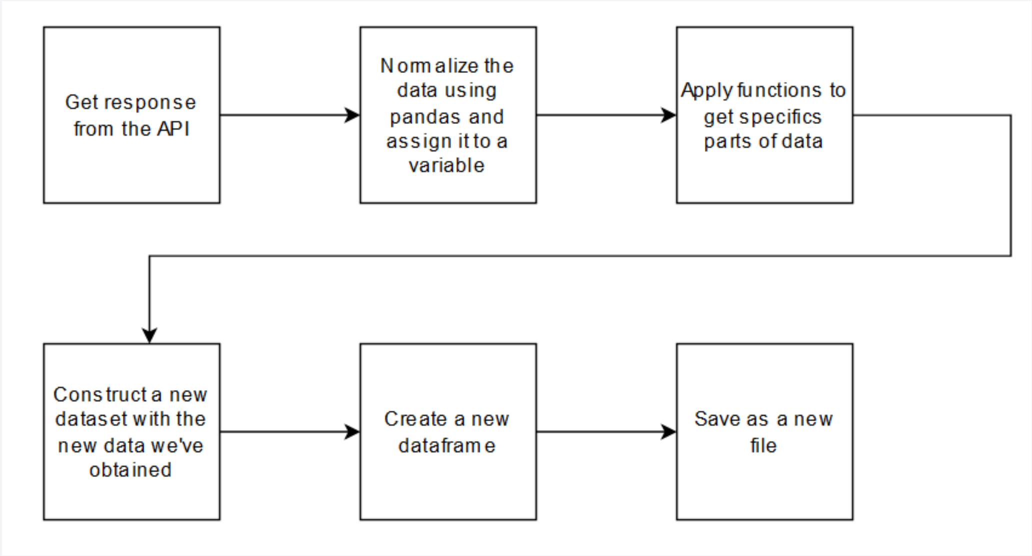


Data Collection – SpaceX API

- I used the SpaceX API to collect the data. With Pandas and some predefined function I created a new dataframe with clean data

- Github URL:

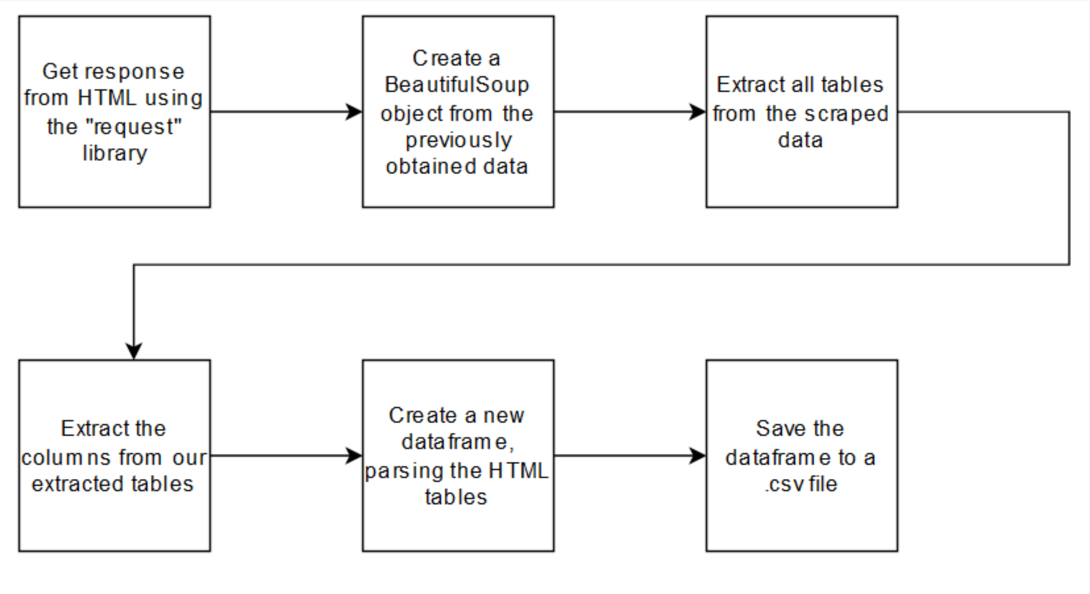
https://github.com/Soutert/IBM-DS-Capstone/blob/main/1-SpaceX_Data_Collection_API.ipynb



Data Collection - Scraping

- I used the Request library to get an HTML response, BeautifulSoup to convert this response to its object and scrape the data from tables to a dictionary then to a new dataframe
- Github URL:

https://github.com/Soutert/IBM-DS-Capstone/blob/main/2-SpaceX_WebScraping_SpaceX.ipynb

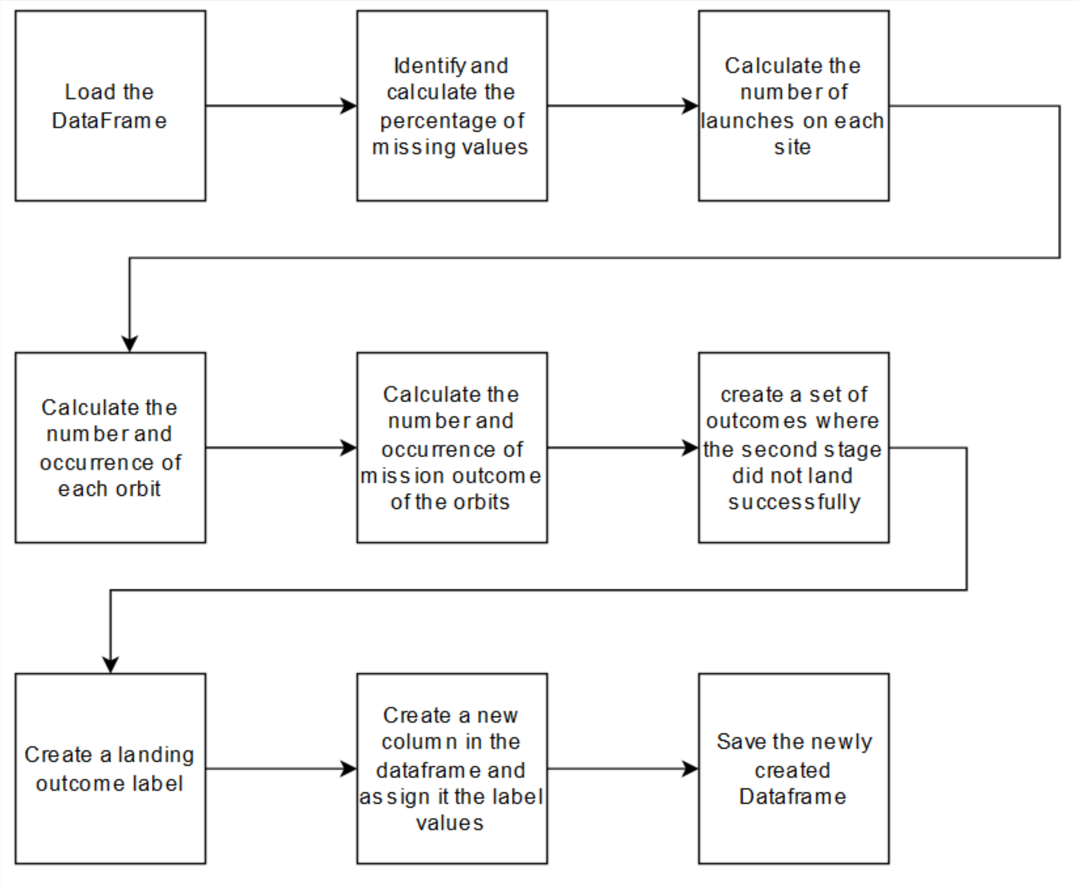


Data Wrangling

- Firstly I checked for missing values, calculated the number of launches at each site, the number and occurrence of each orbit and the number and occurrence of mission outcome per orbit type, then created a class label from the landing class and saved the new dataframe

- Github URL:

https://github.com/Soutert/IBM-DS-Capstone/blob/main/3-Data_Wrangling_SpaceX.ipynb



EDA with Data Visualization

Charts

- Flight Number vs Payload
- Flight Number vs Launch Site
- Payload Mass vs Launch Site
- Success Rate vs Orbit Type
- Flight Number vs Orbit Type
- Payload vs Orbit Type
- Success Rate vs Year

I used scatter plots to view relationship between the variables, showing comparisons using bar charts and finally a line chart to see the success rate across the years.

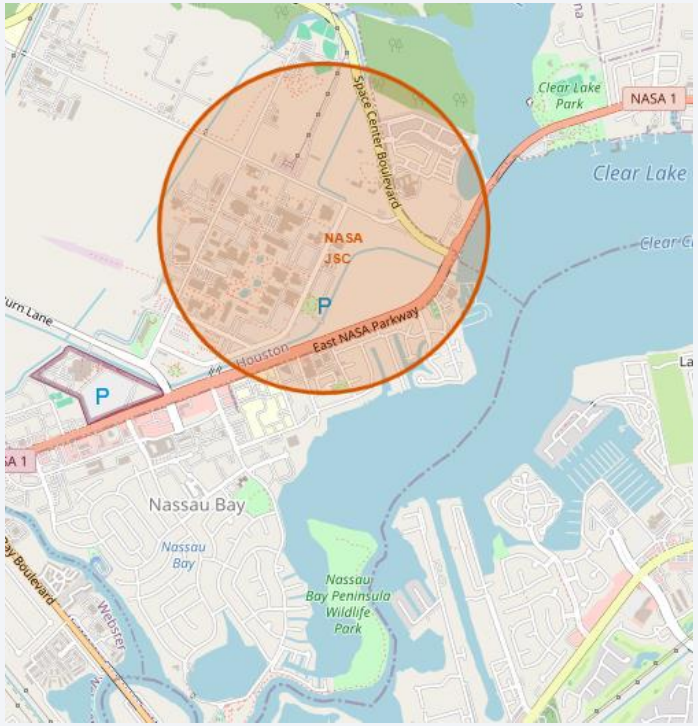
Github URL: https://github.com/Soutert/IBM-DS-Capstone/blob/main/5-EDA_Visualization.ipynb

EDA with SQL

- Performed these queries to gather information:
  - Created a new table from the file where the dates were not null
  - Selected the unique names of the launch sites
  - Displayed 5 records of launch sites which its name started with 'CCA'
  - Displayed the total payload mass carried by boosters launched by NASA
  - Displayed the average payload mass carried by booster version F9 v1.1
  - Listed the date when the first successful landing outcome in ground pad was achieved
  - Listed the name of the booster which had success in drone ship and a payload mass between 4000 and 6000 kg
  - Listed total number of successful and failed mission outcomes
  - Listed the booster version names which carried the maximum payload mass
  - Listed the records displaying month names, failure landing outcomes, booster versions, launch site for the months in 2015
  - Ranked the count of landing outcomes between 2010-06-04 and 2017-03-20
- Github URL: https://github.com/Soutert/IBM-DS-Capstone/blob/main/4-EDA_SQL.ipynb

Build an Interactive Map with Folium

- I created **circle markers** to visualize the coordinates at each launch site, each coordinate with its respective **label.**
- Using a **MarkerCluster** I was able to mark the success and failed launches for each site on the map.
- Github URL: https://github.com/Soutert/IBM-DS-Capstone/blob/main/6-Folium.ipynb

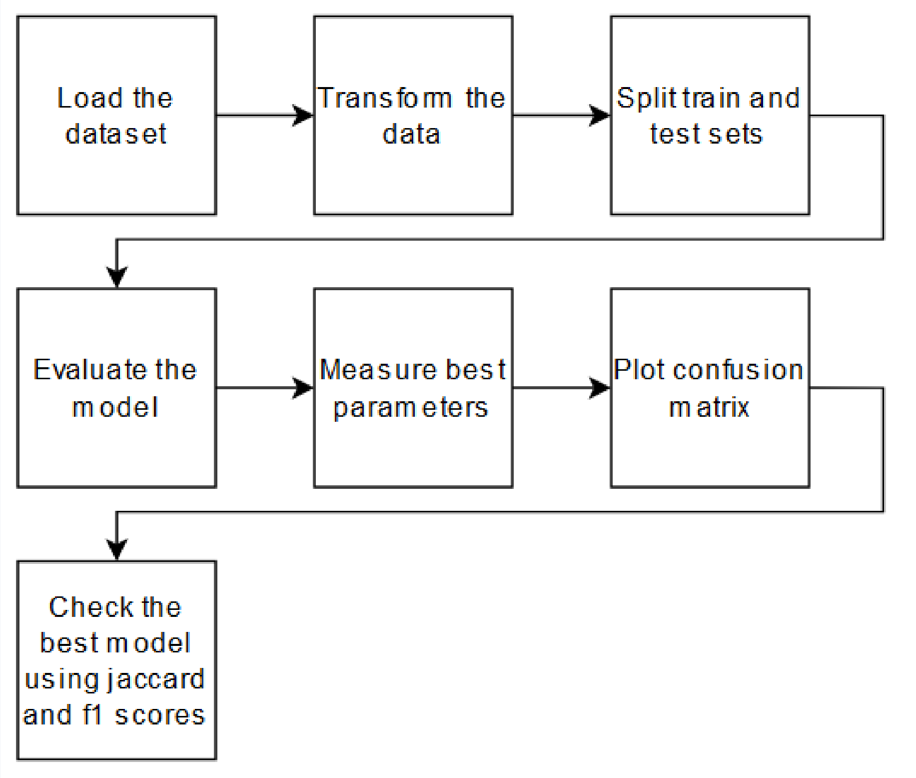


Build a Dashboard with Plotly Dash

- I added both Pie charts and Scatter plots to my Web App, making them interactive with the user where they can change the site from which they want to see the results, as well to selecting a minimum and maximum payload range for the scatter plot
- I added those plots to provide the user with a easy interface and a interactive way to see the launches per site and the correlation between failed / successful launches in a relation with the payload (kg)
- Github URL: https://github.com/Soutert/IBM-DS-Capstone/blob/main/7-spacex_dash_app.py

Predictive Analysis (Classification)

- Firstly I loaded the dataset, transformed the data and split it into training and test sets. Next I evaluated each model (Logistic Regression, Support Vector Machine), Decision Trees and K-Nearest Neighbors (KNN)), then I measured its best parameters and its accuracy, lastly plotting a confusion matrix for each model.
- Lastly I found the best method using jaccard and f1 scores, as well as measuring the accuracy of each one. Resulting in SVM being the best model with higher scores in the three categories.
- Github URL: https://github.com/Soutert/IBM-DS-Capstone/blob/main/8-SpaceX-ML.ipynb



Results

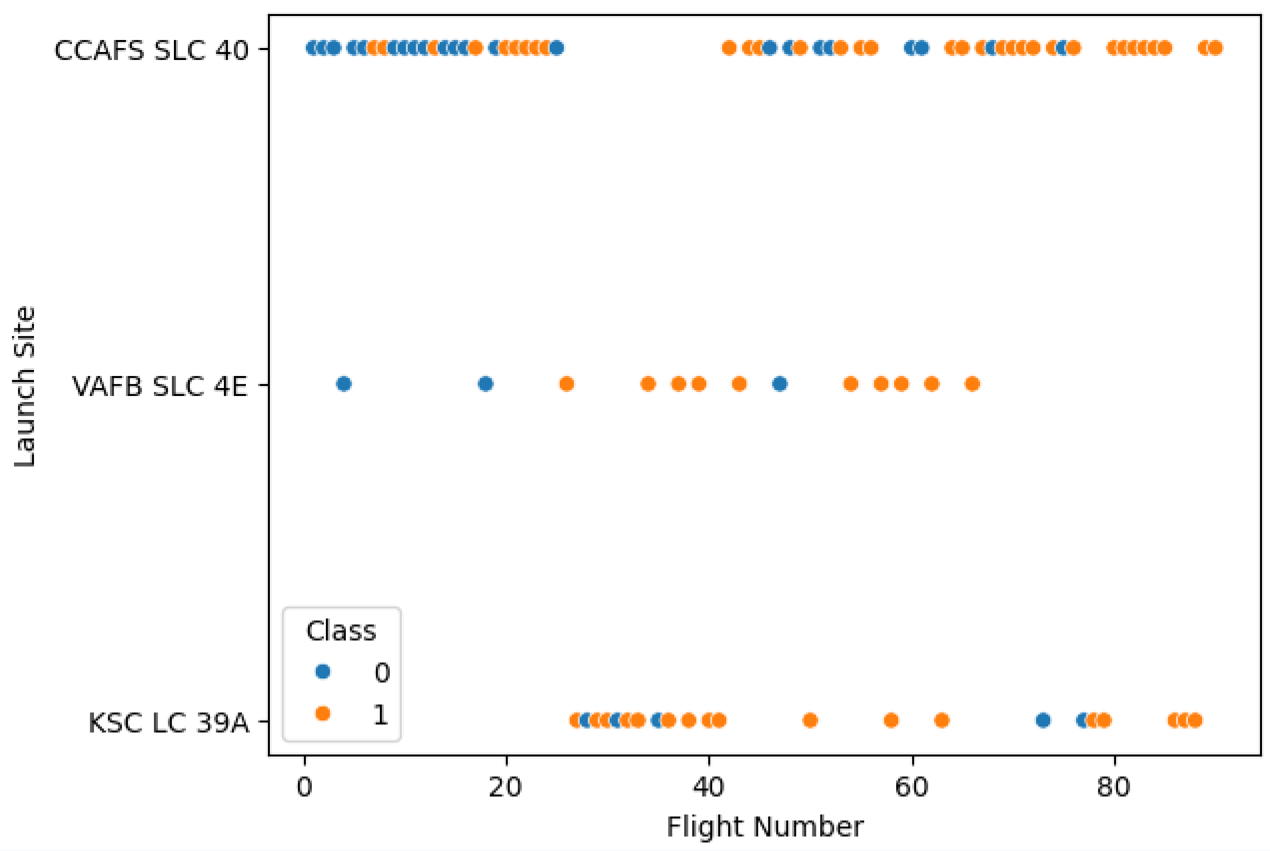
In the next slides I'll be showing the result in these categories:

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

Section 2

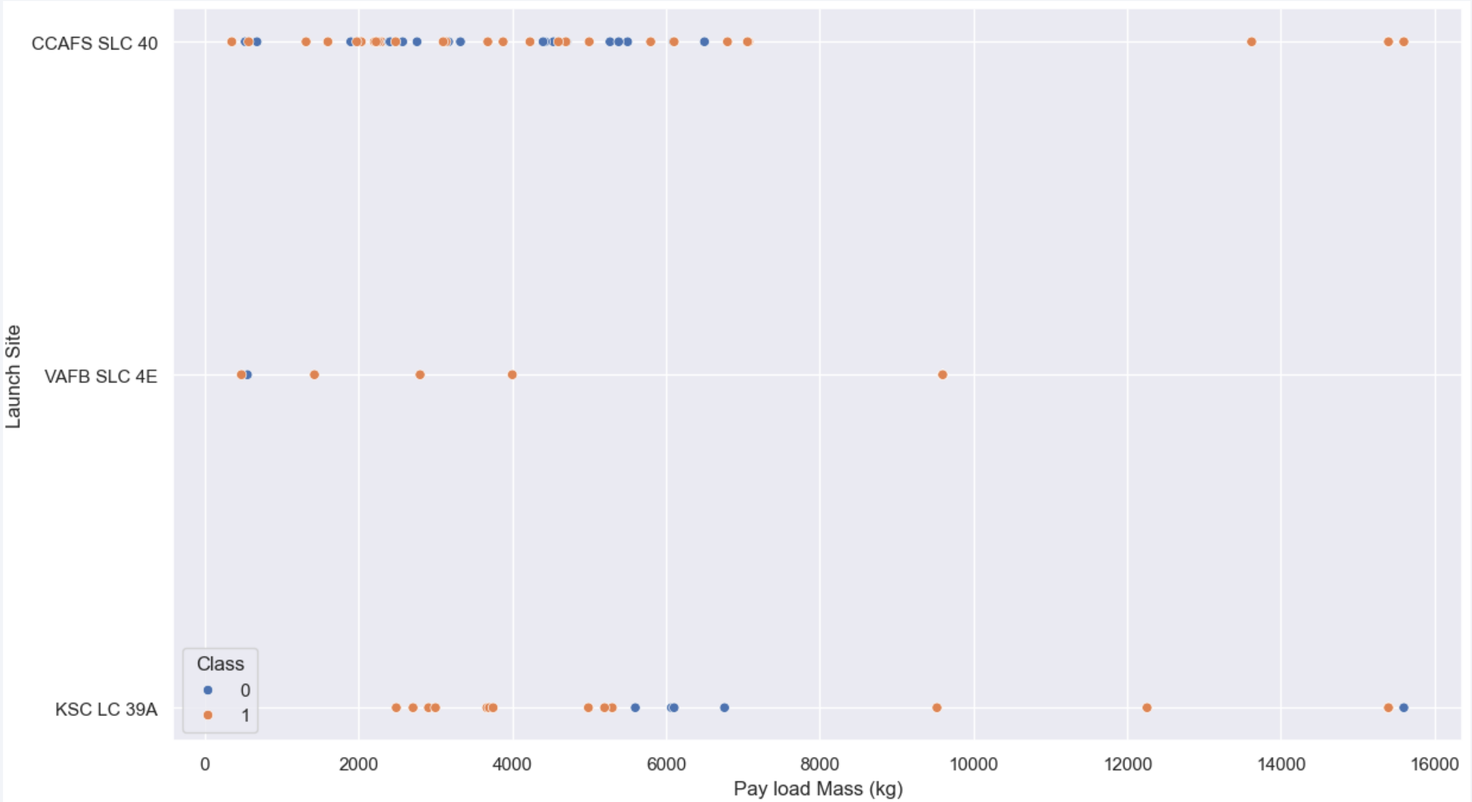
Insights drawn from EDA

Flight Number vs. Launch Site



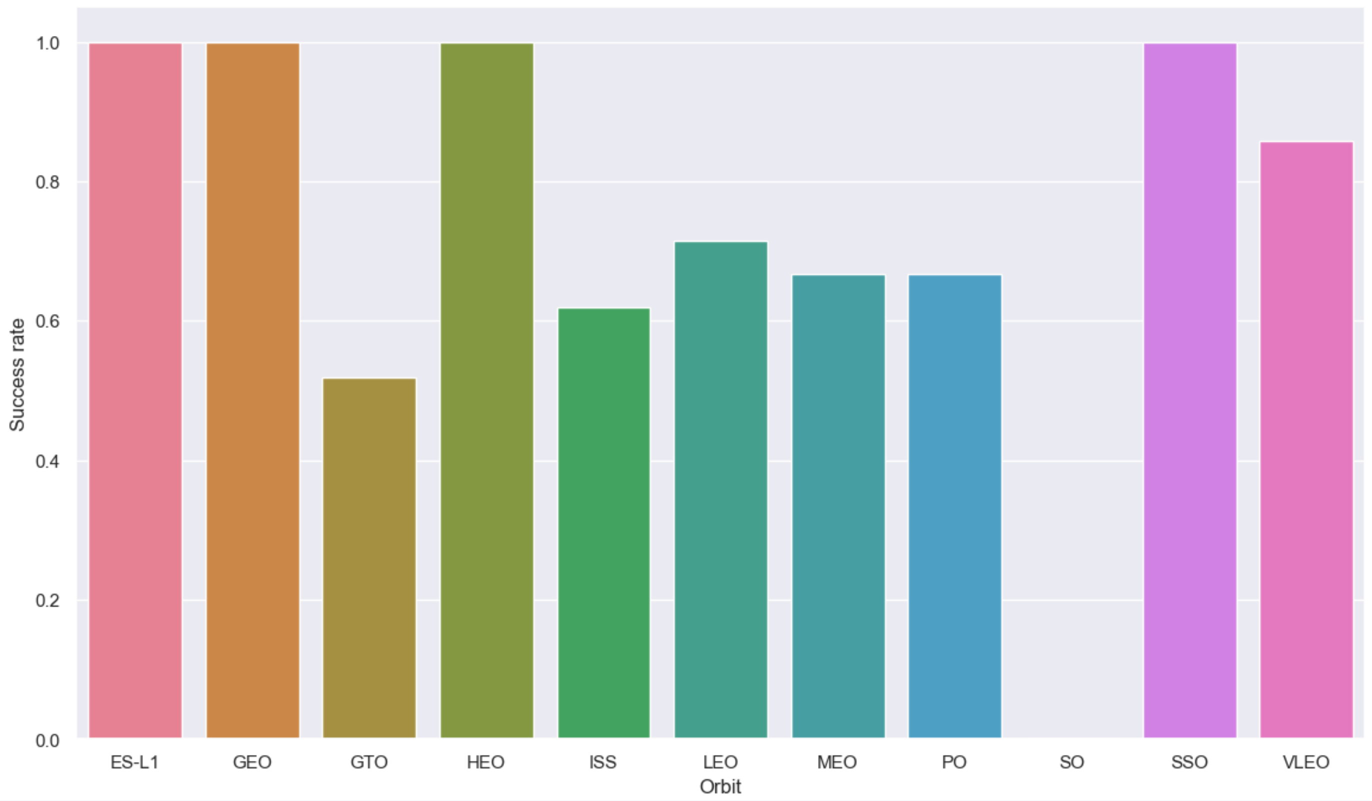
- This plot mainly shows that, apart from a few exceptions, the greater the amount of flights the greater the success rate will be. And we can see that VAFB SLC 4E didn’t send that many flights in comparison of the other two

Payload vs. Launch Site



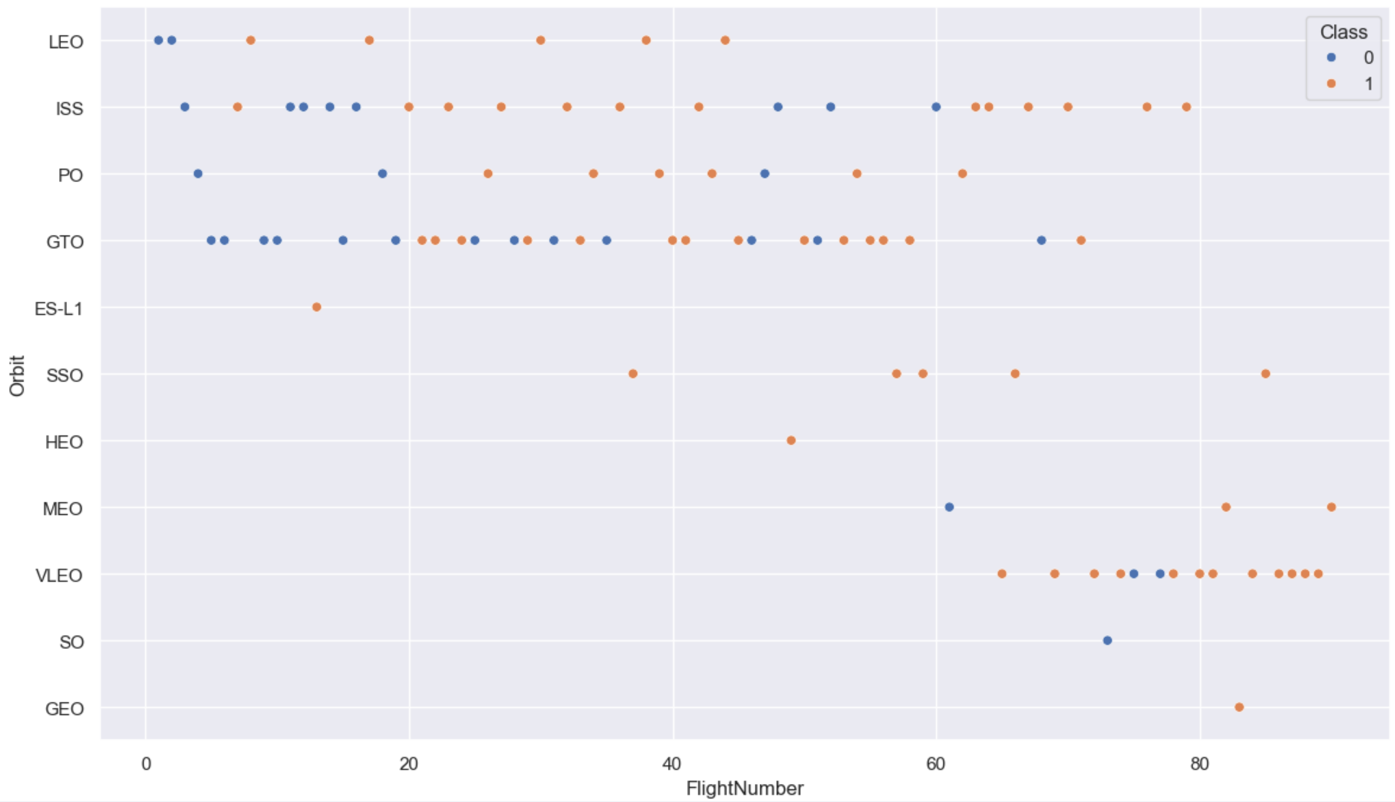
- We see that is common to send low payload flights that have mixed results. But when flights over 8000 kg of payload mass are sent, they have a better success rate. But this isn’t definitive because the lack of samples

Success Rate vs. Orbit Type



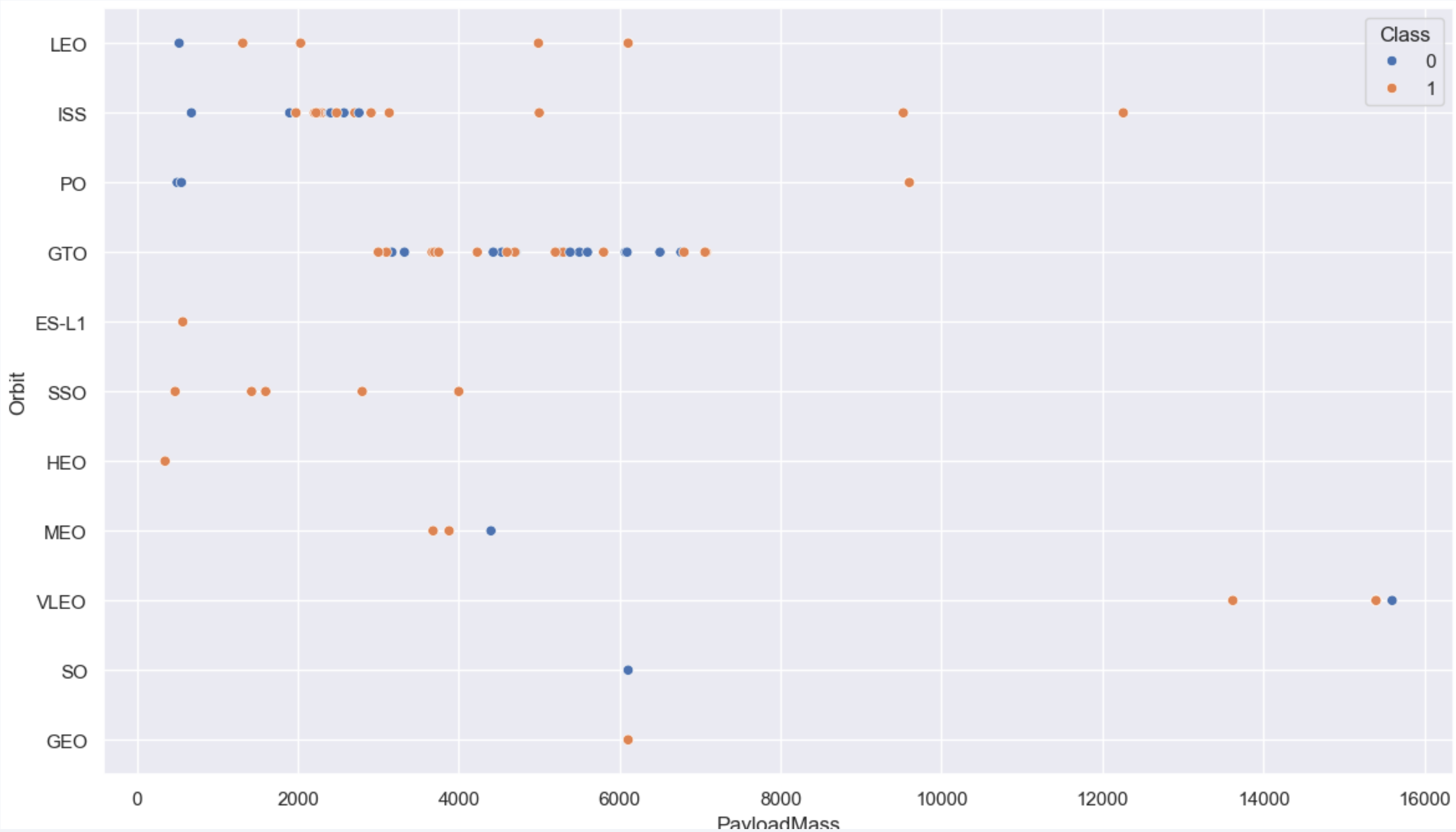
- This shows that ES-L1, GEO, HEO and SSO have a 100% success rate, SO has a 0% success rate and the other ones have mixed results, varying from 50% - 80% success rate

Flight Number vs. Orbit Type



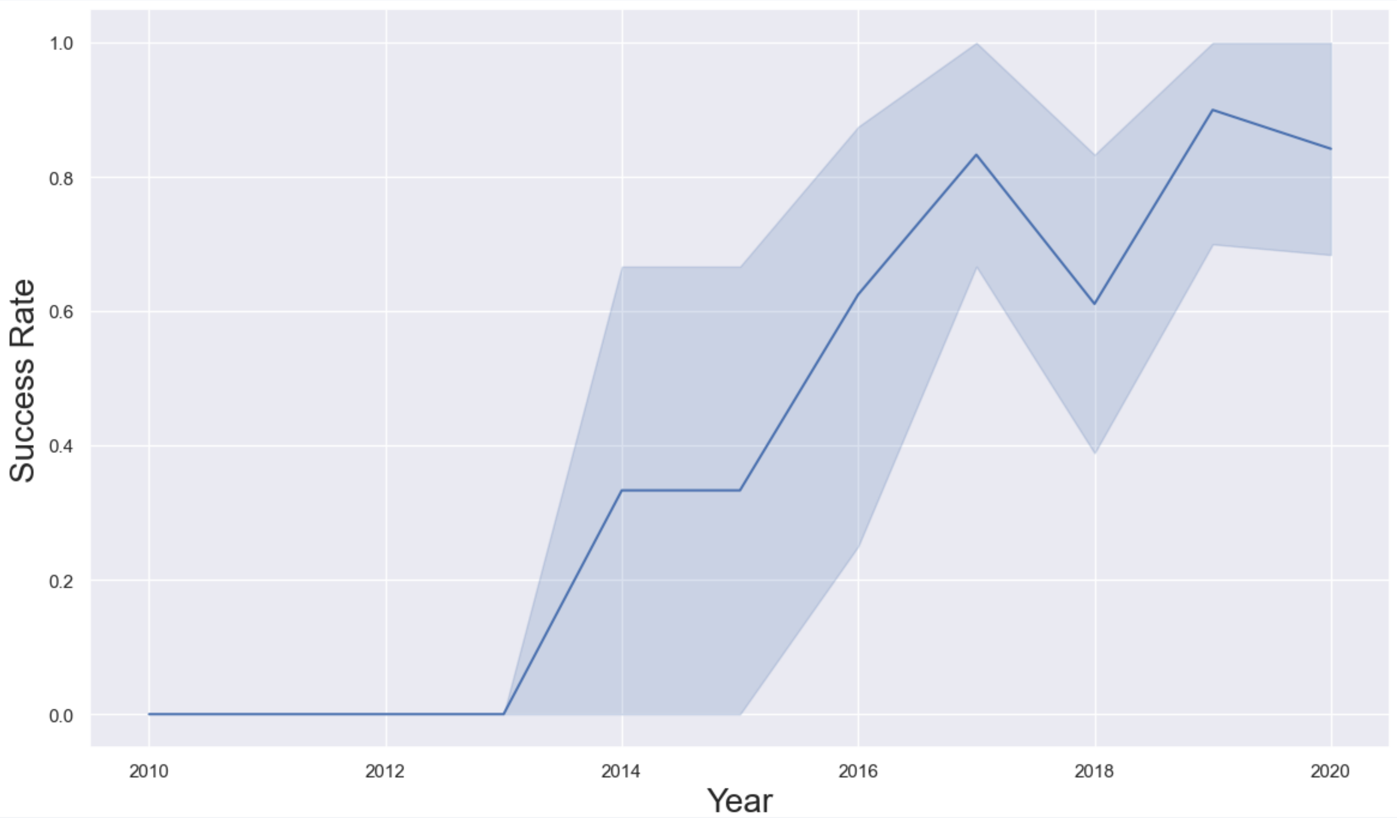
- We can see that more flights numbers on each orbit the better the success rate, we can see this specially on LEO, but it don’t apply to GTO and ISS found to have some failures in the middle ground.

Payload vs. Orbit Type



- Heavy payloads are better with some orbits like LEO, ISS, PO. We can also see that GTO have mixed results with heavier payloads and that SSO had positive results with all of the relatively lighter payloads

Launch Success Yearly Trend



- We can see that the success rate increased over the years, having some fluctuation in 2018 but peaking at 2019.

All Launch Site Names

- I used the SQL keyword **DISTINCT** to get all the unique launch sites in the space mission

```
%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

Launch Site Names Begin with 'CCA'

- I used the keyword LIKE to search records that started with the string "CCA", putting the '%' at the end to ask for results that start with that string. I also used the keyword LIMIT to limit the results that I get, in this case, five results.

Display 5 records where launch sites begin with the string 'CCA'

```
%sql SELECT * from SPACEXTABLE where Launch_Site LIKE 'CCA%' limit 5
```

✓ 0.0s

```
* 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

- I used the SUM function passing the column 'PAYLOAD_MASS__KG_' but specifying that I only wanted to check the results for 'NASA (CRS)' as the customer, for the last part I used the where clause

Display the total payload mass carried by boosters launched by NASA (CRS)

```
%sql select SUM(PAYLOAD_MASS__KG_) from SPACEXTABLE where Customer = 'NASA (CRS)'
```

✓ 0.0s

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

SUM(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

- I used the AVG function passing the ‘PAYLOAD_MASS__KG_’ column, using the where clause to specify which Booster Version I was going to use in this case.

Display average payload mass carried by booster version F9 v1.1

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) from SPACEXTABLE where Booster_Version = 'F9 v1.1'
```

✓ 0.0s

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

AVG(PAYLOAD_MASS__KG_)
2928.4

First Successful Ground Landing Date

- I used the MIN function, passing the ‘Date’ column as a parameter, to get the minimum date in all of the table.

List the date when the first succesful landing outcome in ground pad was acheived.

Hint:Use min function

```
1  %sql SELECT min(Date) from SPACEXTABLE
```

0.0s

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

min(Date)
2010-06-04

Successful Drone Ship Landing with Payload between 4000 and 6000

- I used a where clause with two conditions, first one is that the Mission Outcome was successful and the second one, with help of the BETWEEN clause, was to only select rows in which the PAYLOAD_MASS__KG_ was between 4000 and 6000

List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

```
%sql SELECT * from SPACEXTABLE where Mission_Outcome = 'Success' and PAYLOAD_MASS__KG_ BETWEEN 4000 and 6000
```

✓ 0.0s

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2014-08-05	8:00:00	F9 v1.1	CCAFS LC-40	AsiaSat 8	4535	GTO	AsiaSat	Success	No attempt
2014-09-07	5:00:00	F9 v1.1 B1011	CCAFS LC-40	AsiaSat 6	4428	GTO	AsiaSat	Success	No attempt
2015-03-02	3:50:00	F9 v1.1 B1014	CCAFS LC-40	ABS-3A Eutelsat 115 West B	4159	GTO	ABS Eutelsat	Success	No attempt
2015-04-27	23:03:00	F9 v1.1 B1016	CCAFS LC-40	Turkmen 52 / MonacoSAT	4707	GTO	Turkmenistan National Space Agency	Success	No attempt
2016-03-04	23:35:00	F9 FT B1020	CCAFS LC-40	SES-9	5271	GTO	SES	Success	Failure (drone ship)
2016-05-06	5:21:00	F9 FT B1022	CCAFS LC-40	JCSAT-14	4696	GTO	SKY Perfect JSAT Group	Success	Success (drone ship)
2016-08-14	5:26:00	F9 FT B1026	CCAFS LC-40	JCSAT-16	4600	GTO	SKY Perfect JSAT Group	Success	Success (drone ship)
2017-03-16	6:00:00	F9 FT B1030	KSC LC-39A	EchoStar 23	5600	GTO	EchoStar	Success	No attempt
2017-03-30	22:27:00	F9 FT B1021.2	KSC LC-39A	SES-10	5300	GTO	SES	Success	Success (drone ship)
2017-05-01	11:15:00	F9 FT B1032.1	KSC LC-39A	NROL-76	5300	LEO	NRO	Success	Success (ground pad)
2017-09-07	14:00:00	F9 B4 B1040.1	KSC LC-39A	Boeing X-37B OTV-5	4990	LEO	U.S. Air Force	Success	Success (ground pad)
2017-10-11	22:53:00	F9 FT B1031.2	KSC LC-39A	SES-11 / EchoStar 105	5200	GTO	SES EchoStar	Success	Success (drone ship)
2018-01-31	21:25:00	F9 FT B1032.2	CCAFS SLC-40	GovSat-1 / SES-16	4230	GTO	SES	Success	Controlled (ocean)
2018-06-04	4:45:00	F9 B4 B1040.2	CCAFS SLC-40	SES-12	5384	GTO	SES	Success	No attempt
2018-08-07	5:18:00	F9 B5 B1046.2	CCAFS SLC-40	Merah Putih	5800	GTO	Telkom Indonesia	Success	Success
2018-11-15	20:46:00	F9 B5 B1047.2	KSC LC-39A	Es hail 2	5300	GTO	Es hailSat	Success	Success
2018-12-03	18:34:05	F9 B5 B1046.3	VAFB SLC-4E	SSO-A	4000	SSO	Spaceflight Industries	Success	Success
2019-02-22	1:45:00	F9 B5 B1048.3	CCAFS SLC-40	Nusantara Satu, Beresheet Moon lander, S5	4850	GTO	PSN, SpaceIL / IAI	Success	Success
2019-06-12	14:17:00	F9 B5 B1051.2	VAFB SLC-4E	RADARSAT Constellation, SpaceX CRS-18	4200	SSO	Canadian Space Agency (CSA)	Success	Success
2020-06-30	20:10:46	F9 B5B1060.1	CCAFS SLC-40	GPS III-03, ANASIS-II	4311	MEO	U.S. Space Force	Success	Success
2020-07-20	21:30:00	F9 B5 B1058.2	CCAFS SLC-40	ANASIS-II, Starlink 9 v1.0	5500	GTO	Republic of Korea Army, Spaceflight Industries (BlackSky)	Success	Success
2020-11-05	23:24:23	F9 B5B1062.1	CCAFS SLC-40	GPS III-04 , Crew-1	4311	MEO	USSF	Success	Success

Total Number of Successful and Failure Mission Outcomes

- I selected the Mission Outcome and, with help of the COUNT function, a count of these outcomes. Grouping them (using the GROUP BY clause) by Mission Outcomes

List the total number of successful and failure mission outcomes

```
%sql SELECT Mission_Outcome, Count(Mission_Outcome) as 'Total' from SPACEXTABLE group by Mission_Outcome
```

✓ 0.0s

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

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

Boosters Carried Maximum Payload

- I selected only the booster version, using a WHERE clause with a subquery. The where clause only selected values in which the PAYLOAD MASS was equal to the maximum PAYLOAD MASS in the table.

List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

```
1  %sql SELECT Booster_Version from SPACEXTABLE where PAYLOAD_MASS__KG_ = (Select Max(PAYLOAD_MASS__KG_) from SPACEXTABLE)
```

✓ 0.0s

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

2015 Launch Records

- I used the SUBSTR function, passing the Date, 6 and 2 for the months and a WHERE clause with two parameters. First one was that the Landing Outcome was a failure (drone ship) and second one was that the year was equal to 2015

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, 6,2) as month to get the months and substr(Date,0,5)='2015' for year.

```
%sql SELECT substr(Date, 6,2) as month, Date, Booster_Version, Launch_Site, Landing_Outcome  FROM SPACEXTABLE where Landing_Outcome = 'Failure (drone ship)' and substr(Date,0,5) = '2015'
```

✓ 0.0s

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

month	Date	Booster_Version	Launch_Site	Landing_Outcome
01	2015-01-10	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
04	2015-04-14	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

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

- I selected the Landing Outcome and a count of these values, using a WHERE clause where the Date had to be between 2010-06-04 and 2017-03-20, also I grouped the values by Landing Outcome in descending order.

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.

```
%sql Select Landing_Outcome, count(*) as total from SPACEXTABLE where Date BETWEEN '2010-06-04' and '2017-03-20' group by Landing_Outcome order by total DESC
```

[19] ✓ 0.0s

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

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

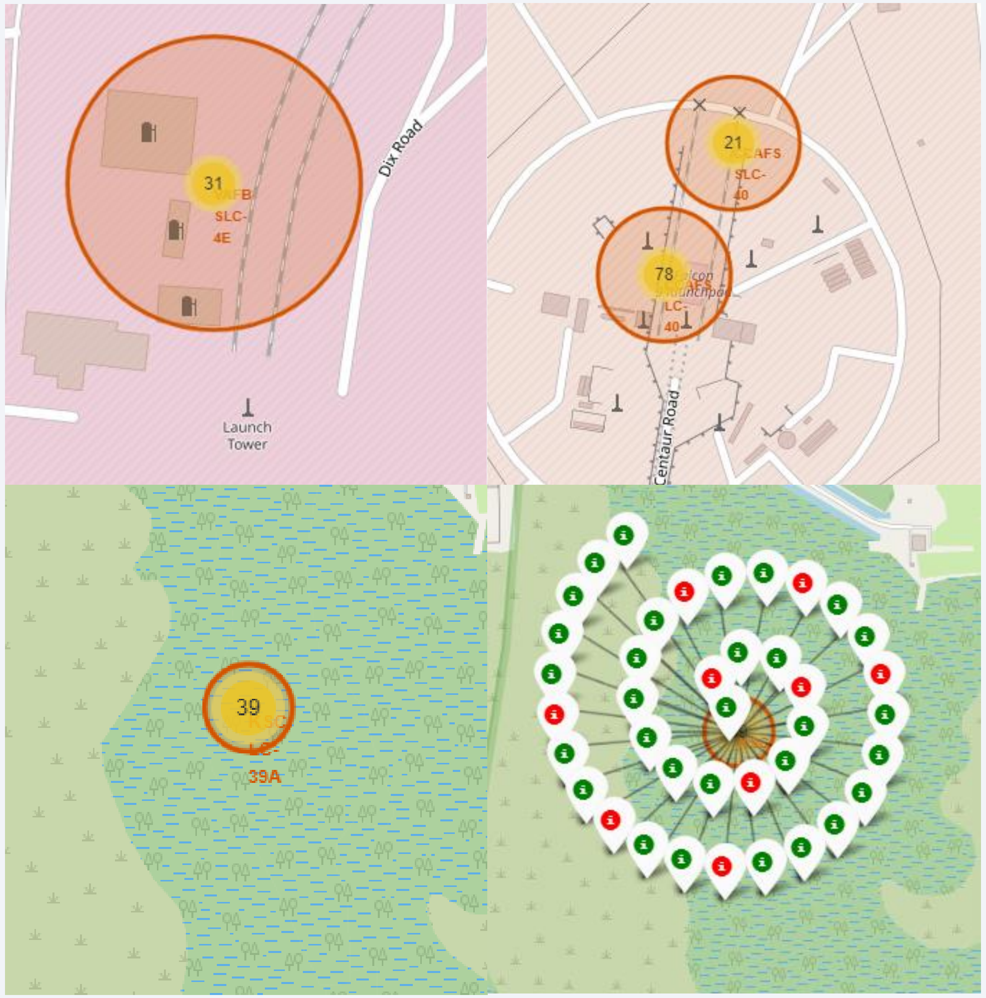
Section 3

Launch Sites Proximities Analysis

SpaceX Launch Sites

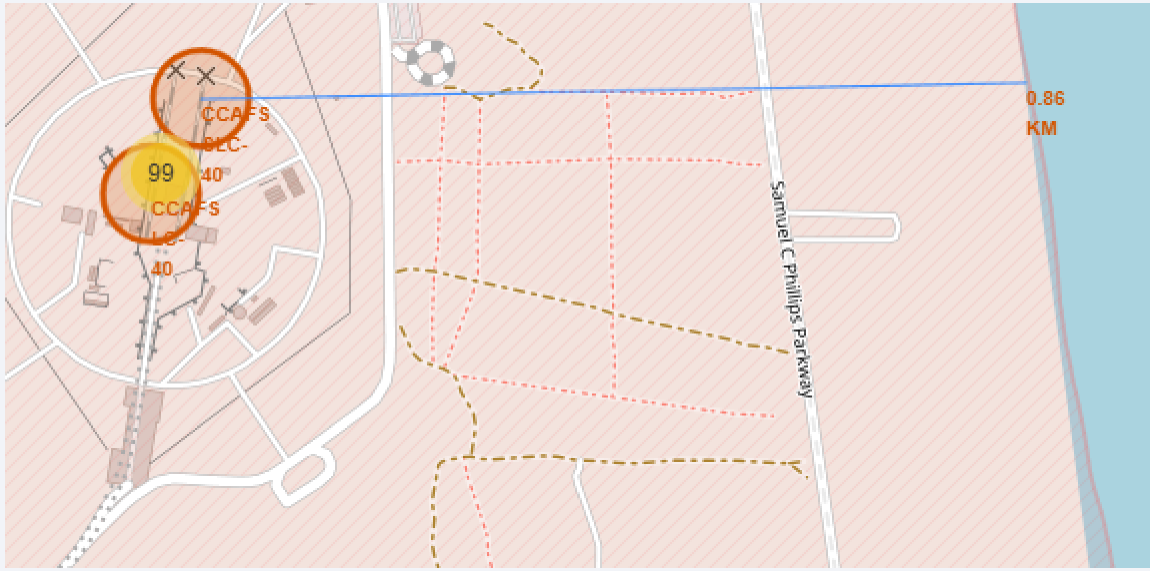
- This maps shows all of the SpaceX Launch Sites located inside the United States

Markers showing Launch Sites with color labels



- This map shows the successful and failed launches in every launch site. Where a green marker shows the successful launches and the red marker shows the failed ones.

Distance from Launch Site

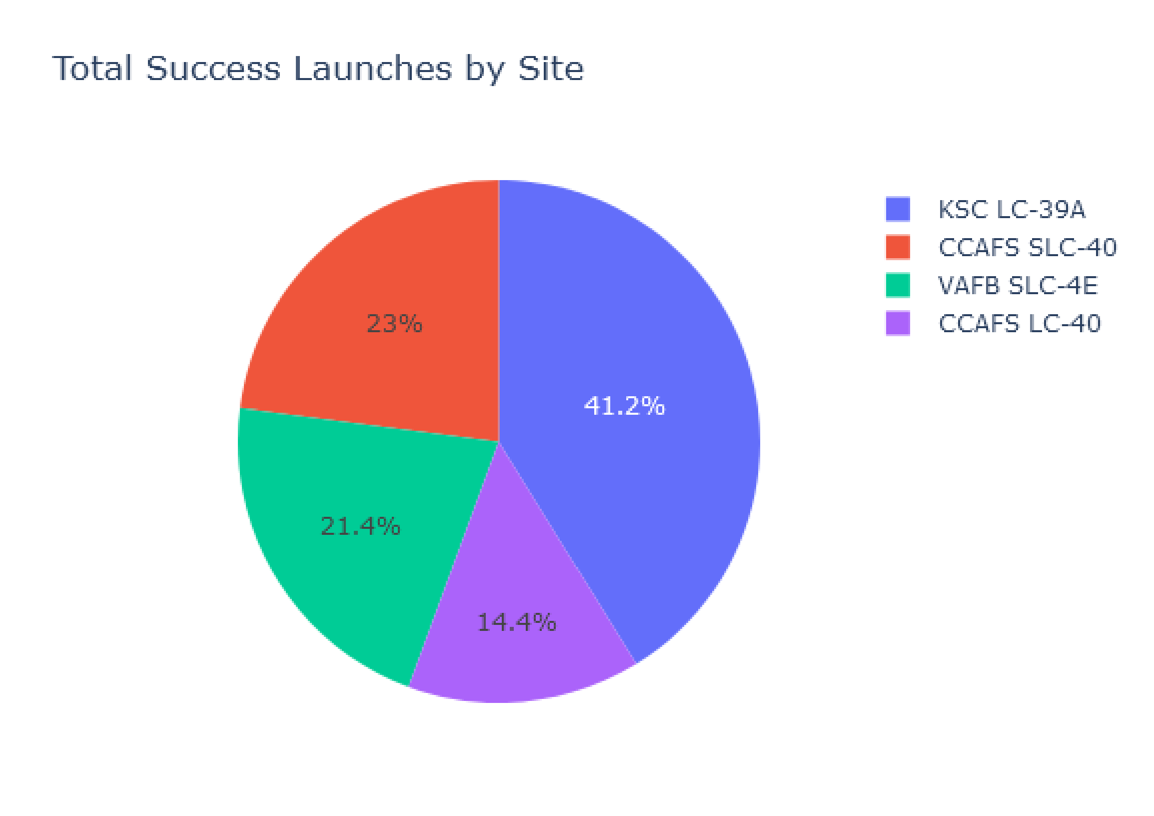


- Using MousePosition and drawing PolyLines I could find distance between one of the LaunchSites and a City and coastline

Section 4

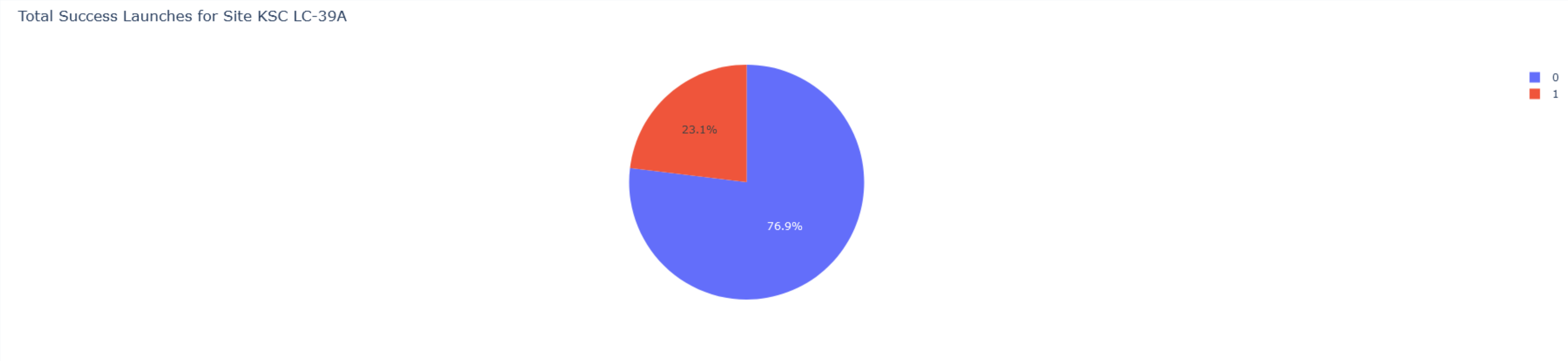
Build a Dashboard with Plotly Dash

Total successful launches in all launch sites



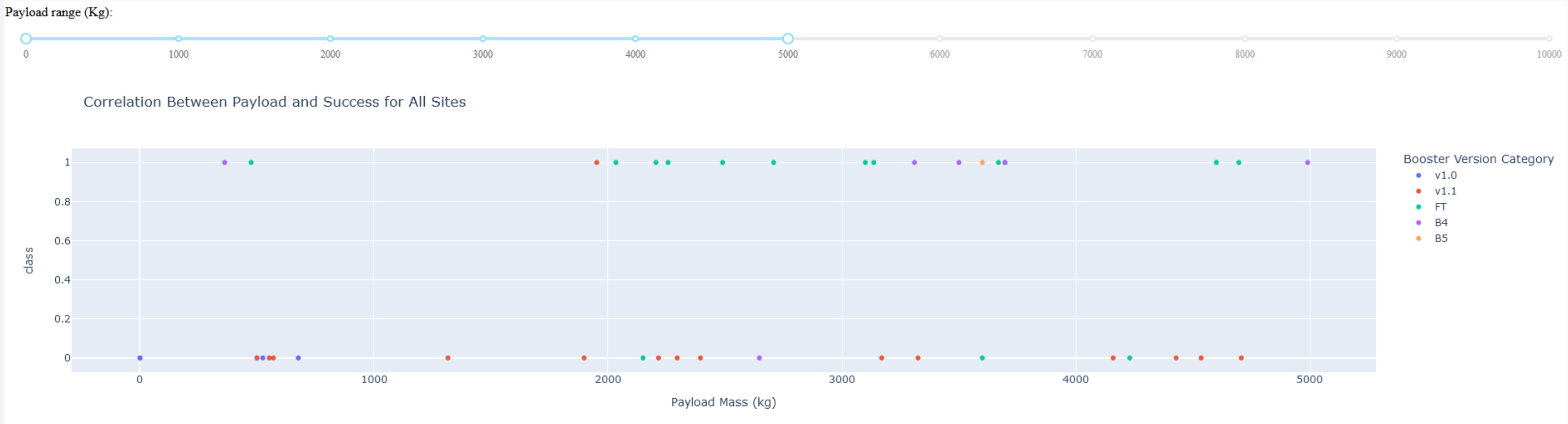
- I found that KSC LC-39A had the most successful launches by a great margin, almost doubling in successful launches the second place in this list, CCAFS SLC-40

Highest success launching site, KSC LC-39A



- KSC LC-39A was the Launch Site with the most successful launches

Payload mass and success

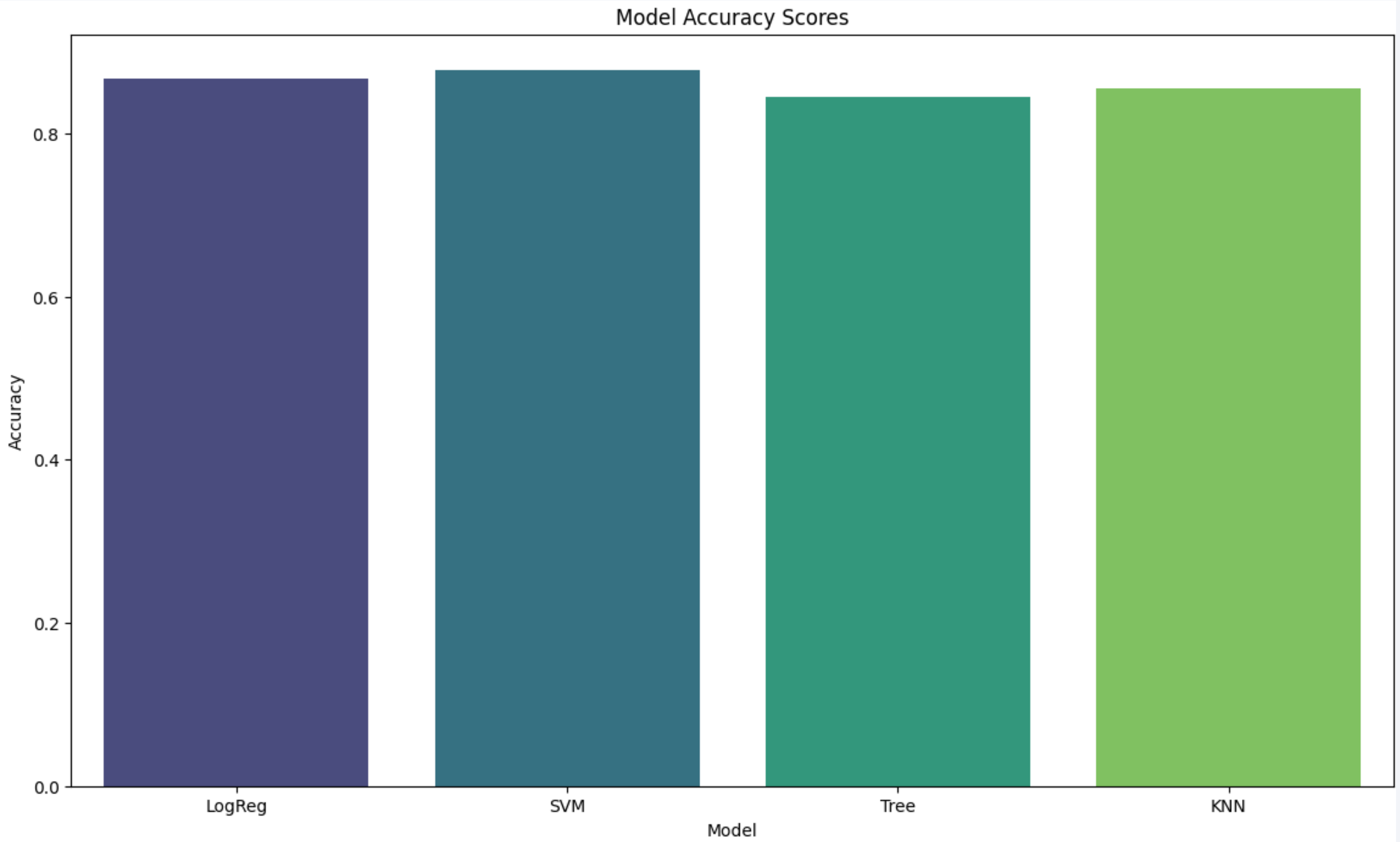


- I could see that only two booster versions were used to do launchings where the payload is greater than 5000 kg. But resulted in less success in all sites

Section 5

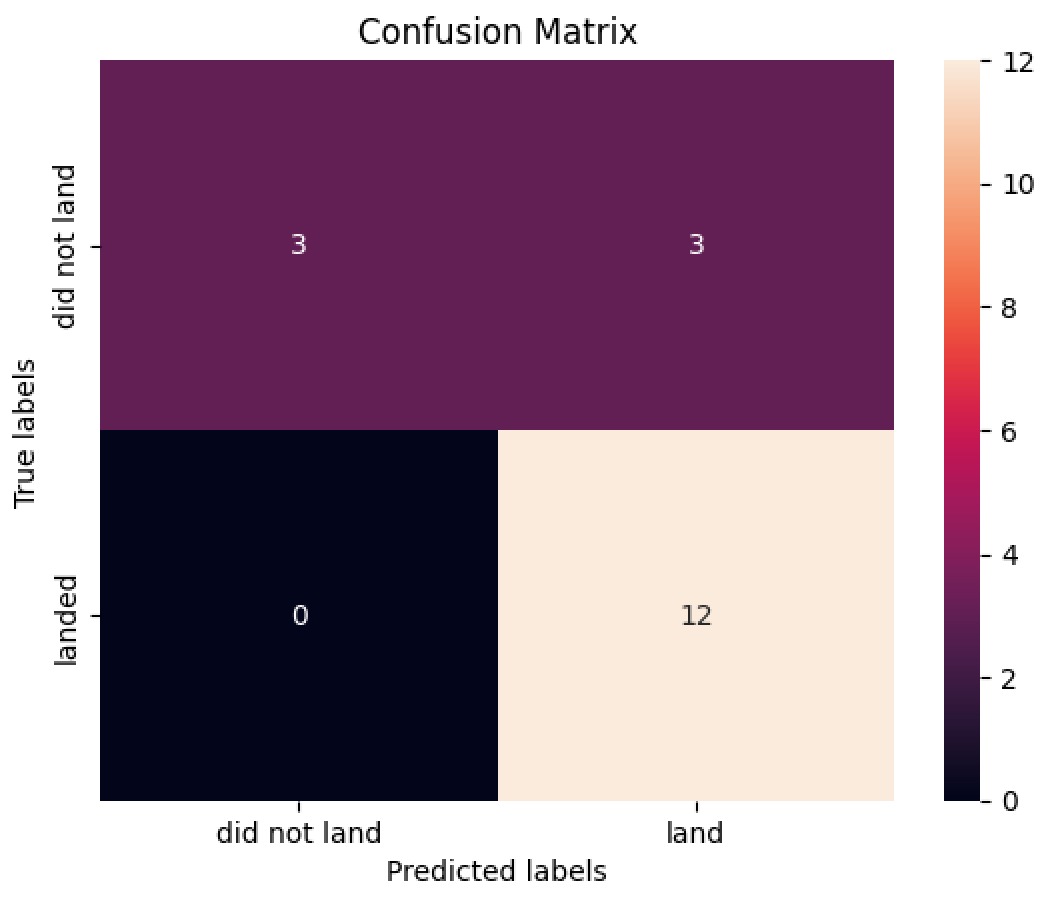
Predictive Analysis (Classification)

Classification Accuracy



- I could find that, even though all the models have closer accuracy scores. SVM was the best in this case, with an accuracy of the 87.7%

Confusion Matrix - SVM



- This is the Confusion Matrix of the SVM model. As we can see our biggest problem was that we got three false positives, but we didn't got any false negatives. So our precision was 80% and our recall 100%

Conclusions

- Launch successes increased over the years
- The models used in this project performed similarly on the test set
- SVM was the better model in this project
- Orbits ES-L1, GEO, HEO and SSO have a 100% success rate

Thank you!