IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

<Thomas>
<23 January 2026>

Outline

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

Executive Summary

- **Objective:** We want to enter the competition with SpaceX by determining optimal launch pricing and assessing first-stage reusability.
- **Methods :**
  - Gather and analyze public data on SpaceX's launch costs, pricing, and operational metrics.
  - Develop interactive dashboards to visualize key insights for strategic decision-making.
- **Innovation:**
  - Replace traditional rocket science with a **machine learning model** to predict first-stage reusability, using publicly available data.
- **Deliverables:**
  - Data-driven launch price recommendations.
  - Predictive model to inform reusability strategy, reducing costs and improving competitiveness.

- **Findings:**

Introduction

- Project background and context
- Problems you want to find answers

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Webscraping of the Space X Wiki page (using Beautiful Soup library) and using Space X API using GET command
- Perform data wrangling
  - Filtering of the dataframe to only include Falcon 9 launches
  - Correction of missing values for Payload Mass (imputation with the mean)
- 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

- Describe how data sets were collected.
- You need to present your data collection process use key phrases and flowcharts

Data Collection – SpaceX API

- We first retrieved Falcon 9 data using the Space X API using GET requests, predefined functions to point at the variables we wanted, turned the JSON object into a dataframe then filtered out all non-Falcon 9 data.
- We corrected missing values on Payload Mass (imputation with the mean)
- GitHub link:

https://github.com/Heenok93/Coursera/blob/6ee962beacffe5af62e3c6e8636715dae522ad22/jupyter-labs-spacex-data-collection-api.ipynb

Place your flowchart of SpaceX API calls here

Data Collection - Scraping

- We then scraped the Space X Wikipedia page to turn a table into a dataframe (Falcon 9 Launches), using the BeautifulSoup library
- We extracted automatically headers and rows for each header
- Turned the beautiful soup object into a dataframe and then csv file

GitHub link:

https://github.com/Heenok93/Coursera/blob/6ee962beacffe5af62e3c6e8636715dae522ad22/jupyter-labs-webscraping.ipynb

Place your flowchart of web scraping here

Data Wrangling

- Wee imported the csv file into a Pandas object and start exploring, create useful new variables and format existing ones
- We first check variable types and distribution with dtypes, value_counts and describe methods
- We calculated the number and occurence of mission outcome of the orbits by creating the landing_outcomes variable
- We finally created a binary label called landing_class to exclude bad outcomes

- Github link:

https://github.com/Heenok93/Coursera/blob/6ee962beacffe5af62e3c6e8636715dae522ad22/labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

- We then used matplotlib and seaborn to visually understand patterns in launches and landings
- We created scatterplots between Flight Number and Launch Site, Flight Number and Payload, Payload and Launch Site, all colored based on success/failure
- We then created scatterplots and lineplots to see the correlation with Orbit

Github link:
https://github.com/Heenok93/Coursera/blob/6ee962beacffe5af62e3c6e8636715dae522ad22/edadataviz.ipynb

EDA with SQL (1)

- First, we displayed the names of the unique launch sites (simple DISTINCT launch_sites query) -> we could see only 4 sites were used
- We used LIMIT 5 to retrieve only the first 5 launching sites starting by 'CCA'
- We computed the sum of payload mass carried by boosters launched by NASA (CRS) (48213)
- We calculated the average payload mass carried by booster version F9 v1.1 (2534.6)
- WE then look for the first successful landing outcome in ground pad (min date) -> '2015-12-22'
- We Listed the names of the boosters which have success in drone ship and have payload mass greater between 4000 and 6000

EDA with SQL (2)

- We computed the total number of successful and failure mission outcomes (79 total, excluding No attempts)
- We then used a subquery to find all the booster_versions that have carried the maximum payload mass
- We then filtered the data to display info about the launches done in year 2015 (only 2)
- Finally we ranked the count of landing outcomes between the date 2010-06-04 and 2017-03-20, in descending order

- Github link:

https://github.com/Heenok93/Coursera/blob/6ee962beacffe5af62e3c6e8636715dae522ad22/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- We created a Folium map to mark the NASA Base, the launching sites and draw some lines to illustrate distances from sites to key features (coastlines, roads, railways and closest city)
- We used circles, basic markers and Cluster of markers to ease the use's life while browsing the map
- We added those objects to understand why location of launch sites is key to the success/failure of launches, and what would happen in case of failures: being as close to the equator as possible is mandatory for success, and in case of failure, we want the debris to fall in the ocean, not on houses/cars or train !
- GitHub link: https://github.com/Heenok93/Coursera/blob/6ee962beacffe5af62e3c6e8636715dae522ad22/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

- We created a dashboard including a Pie Chart for total launches and scatter plots to explore further for each launching site
- We created a dropdown menu to select the launching site, and then created the scatter plots with Plotly to allow for interactivity (popup label, sliders to focus on specific ranges for the variables)
- This allows the used to easily pick what are the most successful launch sites, according to payload mass etc

Github Link:

https://github.com/Heenok93/Coursera/blob/6ee962beacffe5af62e3c6e8636715dae522ad22/spacex-dash-app.py

Predictive Analysis (Classification)

- We then used Scikit-Learn library to setup and compare different classification models (Logistic Regression, SVM, Decision Trees and K-Nearest Neighbours)
- We split the database into Train/Test sets, and setup a cross-validation (10 folds) grid search to fine tune hyperparameters for each models (we did not use a pipeline, but we could have)
- We then computed global accuracy on the train set, the test set and final confusion matrices to be able to compare models
- Github link:

https://github.com/Heenok93/Coursera/blob/6ee962beacffe5af62e3c6e8636715dae522ad22/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

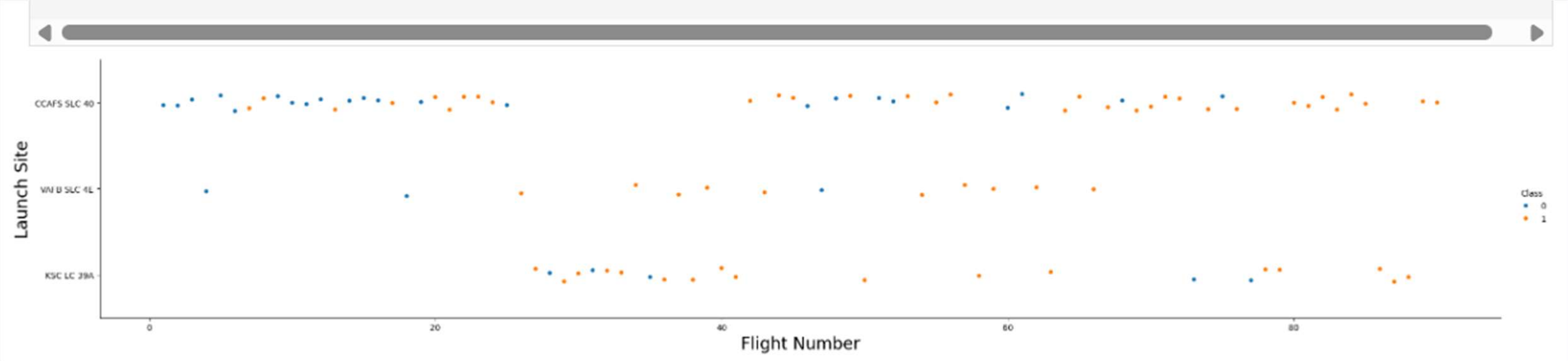
Results

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

Section 2

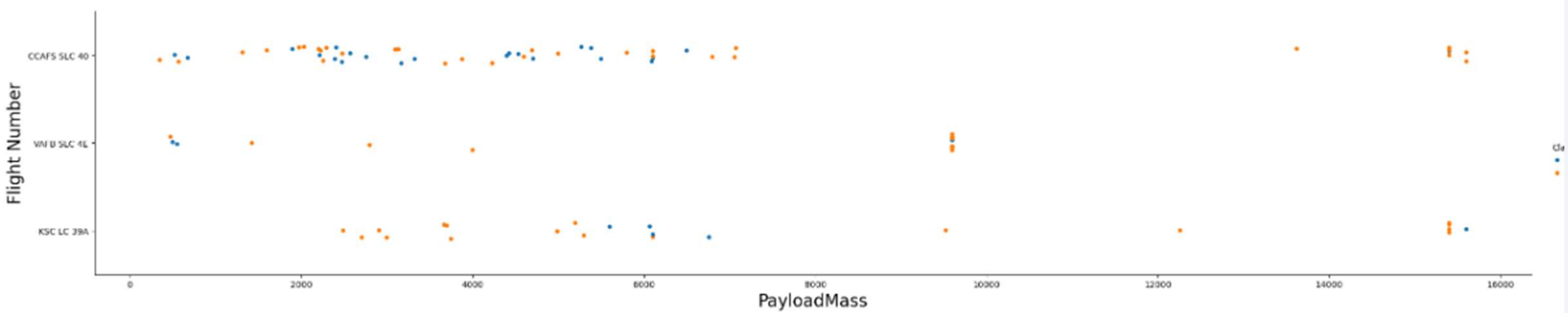
Insights drawn from EDA

Flight Number vs. Launch Site



- CCAS had mixed results in the beginning, but after about 60 flights, their launches were almost always successful
- WFAB did not launch many ships, but most of their were successful
- KSC is the most recent, with a bit more launches than WFAB but with good success as well

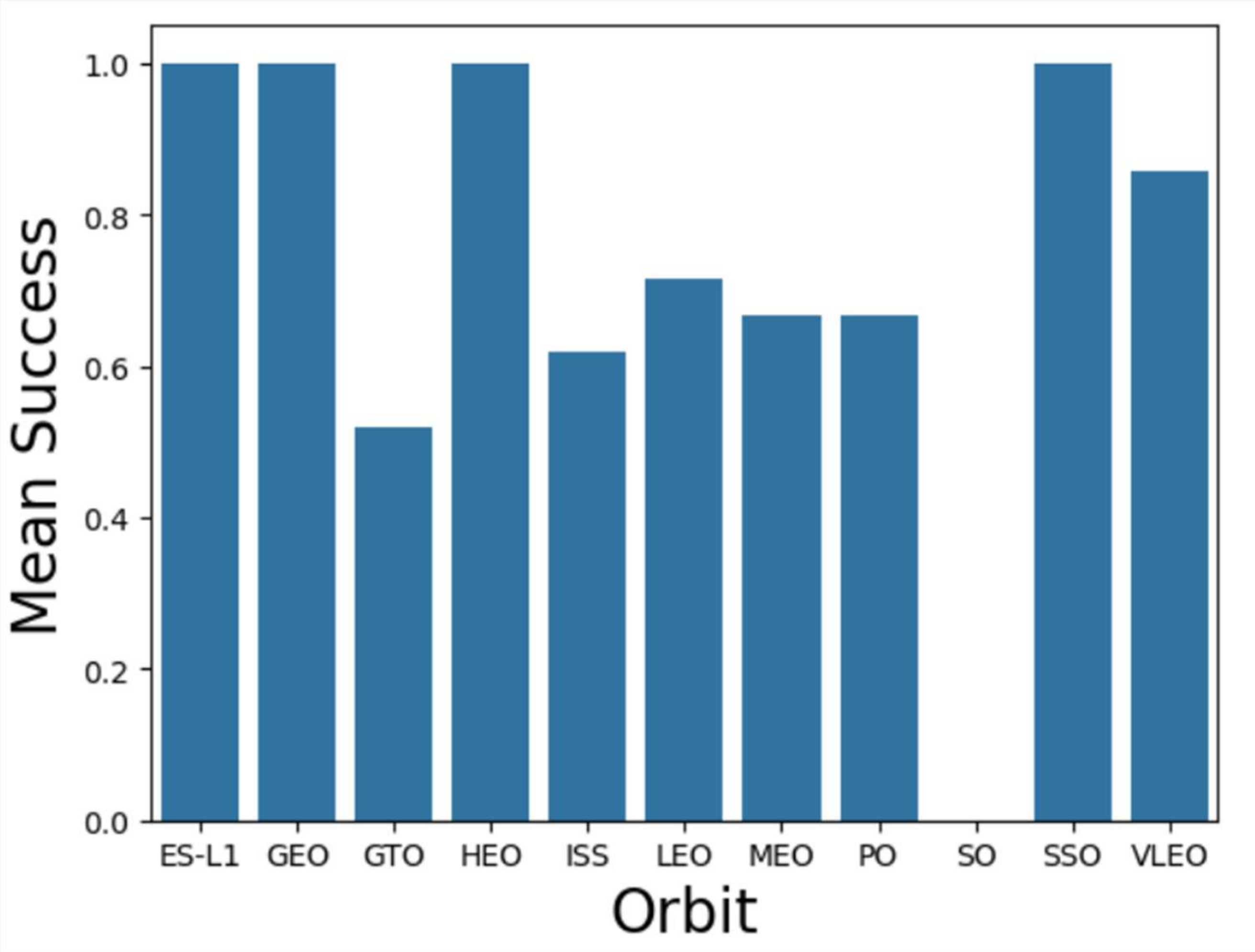
Payload vs. Launch Site



Now if you observe Payload Mass Vs. Launch Site scatter point chart you will find for the VAFB-SLC launchsite there are no rockets launched for heavypayload mass(greater than 10000).

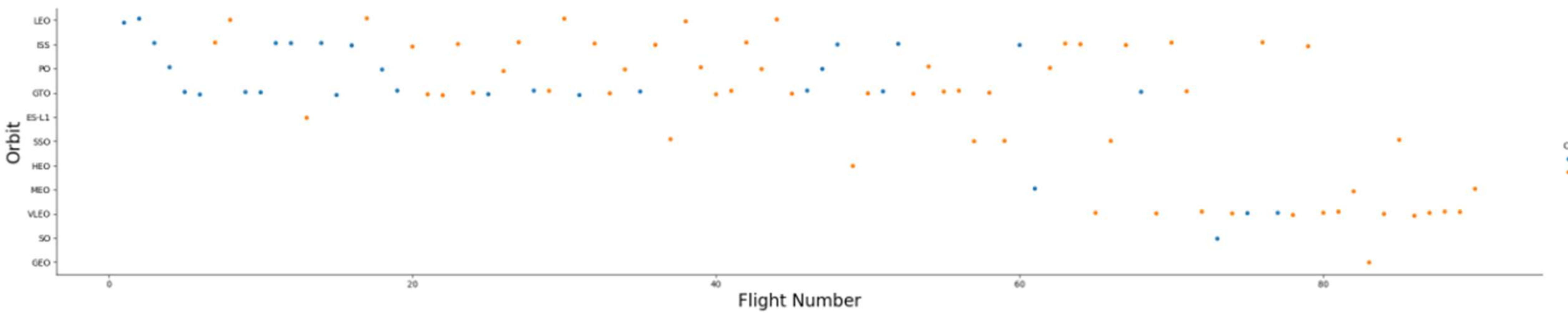
- We can also say than most of CCAS laucnhes were done with less than 8000
- KSC launches had the most variance in payload

Success Rate vs. Orbit Type



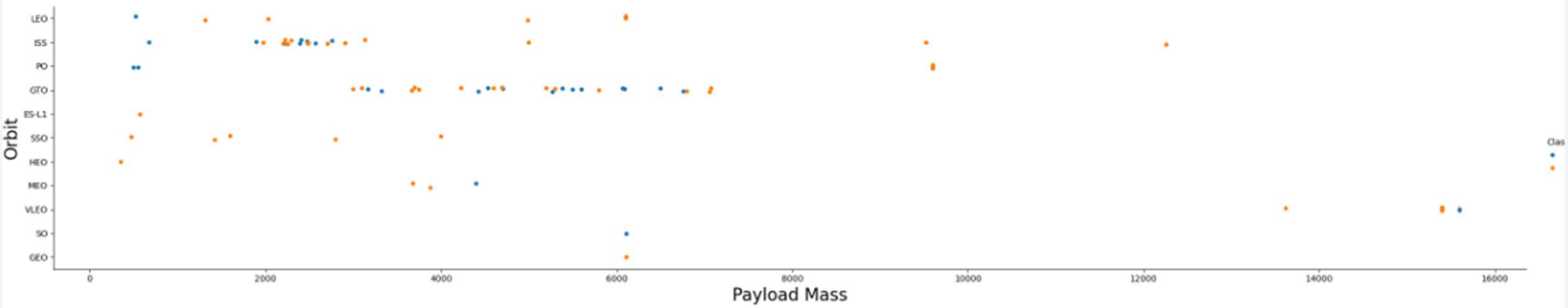
- Most successful are to ESL1, GEO, HEO and SSO
- Least successful are to GTO and ISS
- LEO MEO and PO about 60% success only
- SO no landing ?

Flight Number vs. Orbit Type



You can observe that in the LEO orbit, success seems to be related to the number of flights. Conversely, in the GTO orbit, there appears to be no relationship between flight number and success.

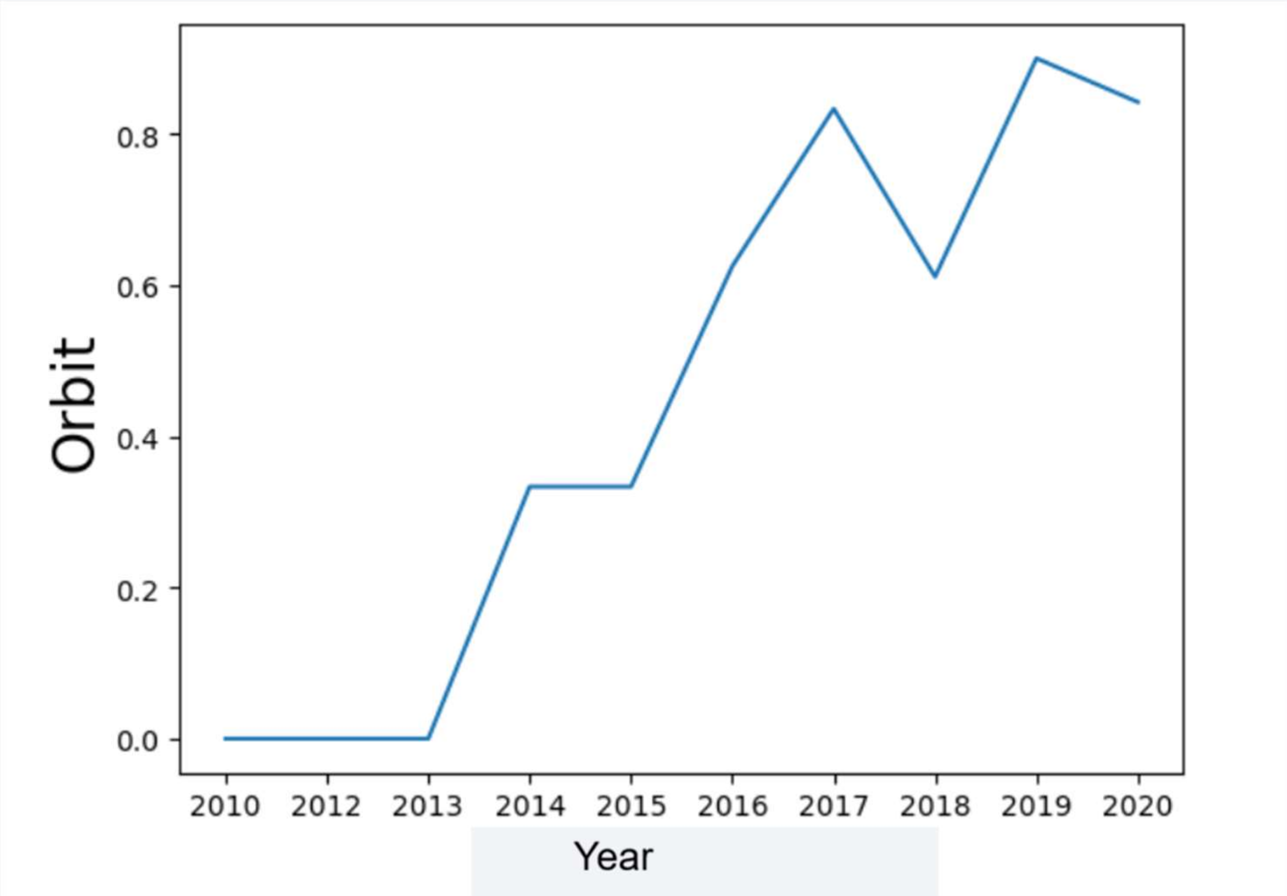
Payload vs. Orbit Type



With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS.

However, for GTO, it's difficult to distinguish between successful and unsuccessful landings as both outcomes are present.

Launch Success Yearly Trend



- Clear increase from 2013 to 2020
- Success seems to stabilize around 80% since 2017

All Launch Site Names

Task 1

Display the names of 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

We have to use distinct to list the sites. We would also have used a GROUP BY and a count.

Launch Site Names Begin with 'CCA'

Task 2

Display 5 records where launch sites begin with the string '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

- We use LIMIT 5 to show only the 5 first ones, and we use a LIKE “CCA%” condition to select only the one starting by CCA.

Total Payload Mass

Task 3

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

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) AS total_payload FROM SPACEXTABLE WHERE Customer LIKE '%NASA (CRS)%';
```

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

total_payload
48213

- We use the sum function and a where condition to select only NASA CRS (with a LIKE statement just in case some are spelled differently)

Average Payload Mass by F9 v1.1

Task 4

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

```
%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

- Same as above but with an AVG function

First Successful Ground Landing Date

Task 5

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

Hint:Use min function

```
%sql SELECT Min(Date) first_success FROM SPACEXTABLE WHERE Landing_Outcome = 'Success (ground pad)';
```

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

first_success
2015-12-22

- Use of min function with a where condition

Successful Drone Ship Landing with Payload between 4000 and 6000

Task 6

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 Booster_Version FROM SPACEXTABLE WHERE ( Landing_Outcome LIKE '%Success %drone ship%' AND ( PAYLOAD_MASS__KG_ BE
```

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

Booster_Version
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
F9 FT B1031.2

Total Number of Successful and Failure Mission Outcomes

List the total number of successful and failure mission outcomes

```
%sql SELECT Landing_Outcome,COUNT(*) as n FROM SPACEXTABLE GROUP BY Landing_Outcome ORDER BY n DESC;
```

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

Landing_Outcome	n
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

```
#if the question is only failure or sucess, here is the total
%sql SELECT COUNT(*) as n FROM SPACEXTABLE WHERE Landing_Outcome NOT LIKE "%attempt%" ORDER BY n DESC;
```

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

n
79

- Excluding ‘no attempts’, there were 79 landings
- 38+14+9 successes (61/79=77% successes)

Boosters Carried Maximum Payload

List all the booster_versions that have carried the maximum payload mass, using a subquery with a suitable aggregate function.

```
%sql SELECT Booster_Version FROM SPACEXTABLE WHERE PAYLOAD_MASS__KG_= (select MAX(PAYLOAD_MASS__KG_) from SPACEXTABLE)
```

```
 * 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

Task 9

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, Booster_Version, Launch_Site, Landing_Outcome FROM SPACEXTABLE WHERE substr(Date, 6
```

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

Month	Booster_Version	Launch_Site	Landing_Outcome
01	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
04	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

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

```
%%sql SELECT
    Landing_Outcome,
    COUNT(*) AS N_landing
FROM
    SPACEXTABLE
WHERE
    Date BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY
    Landing_Outcome
ORDER BY
    N_landing DESC;
```

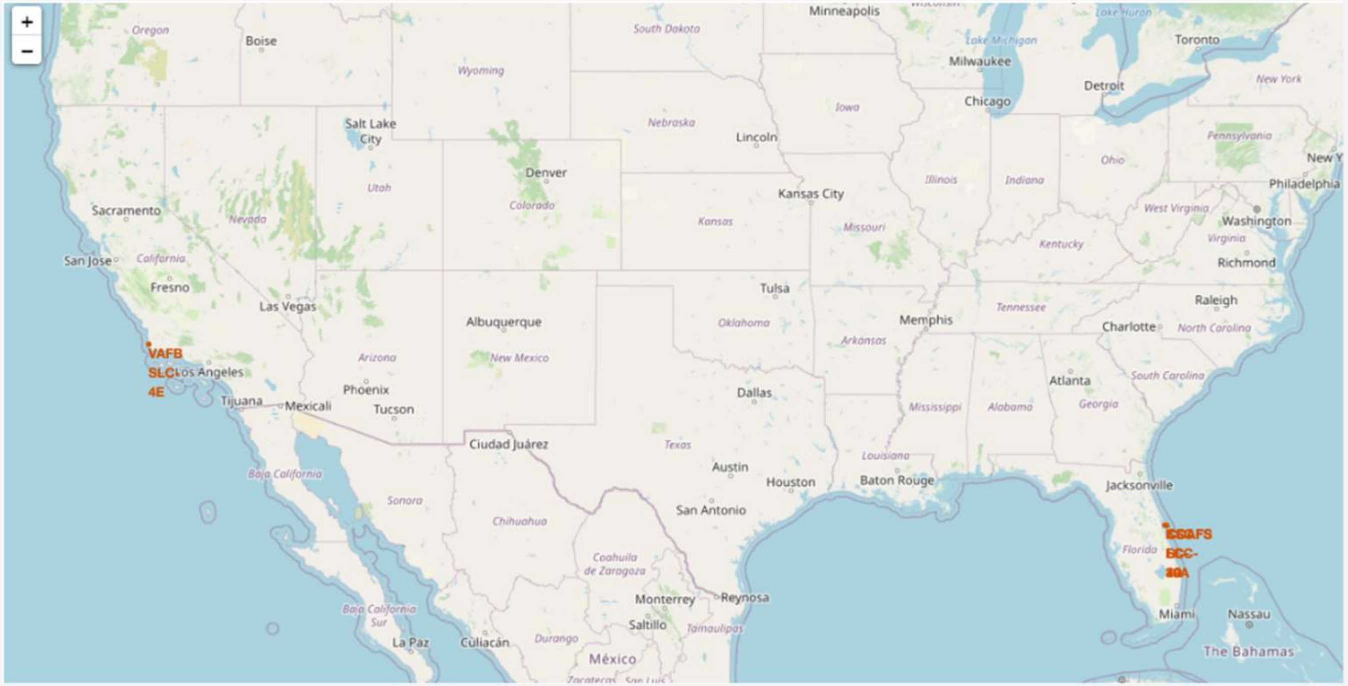
```
 * sqlite:///my_data1.db
Done.
```

Landing_Outcome	N_landing
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

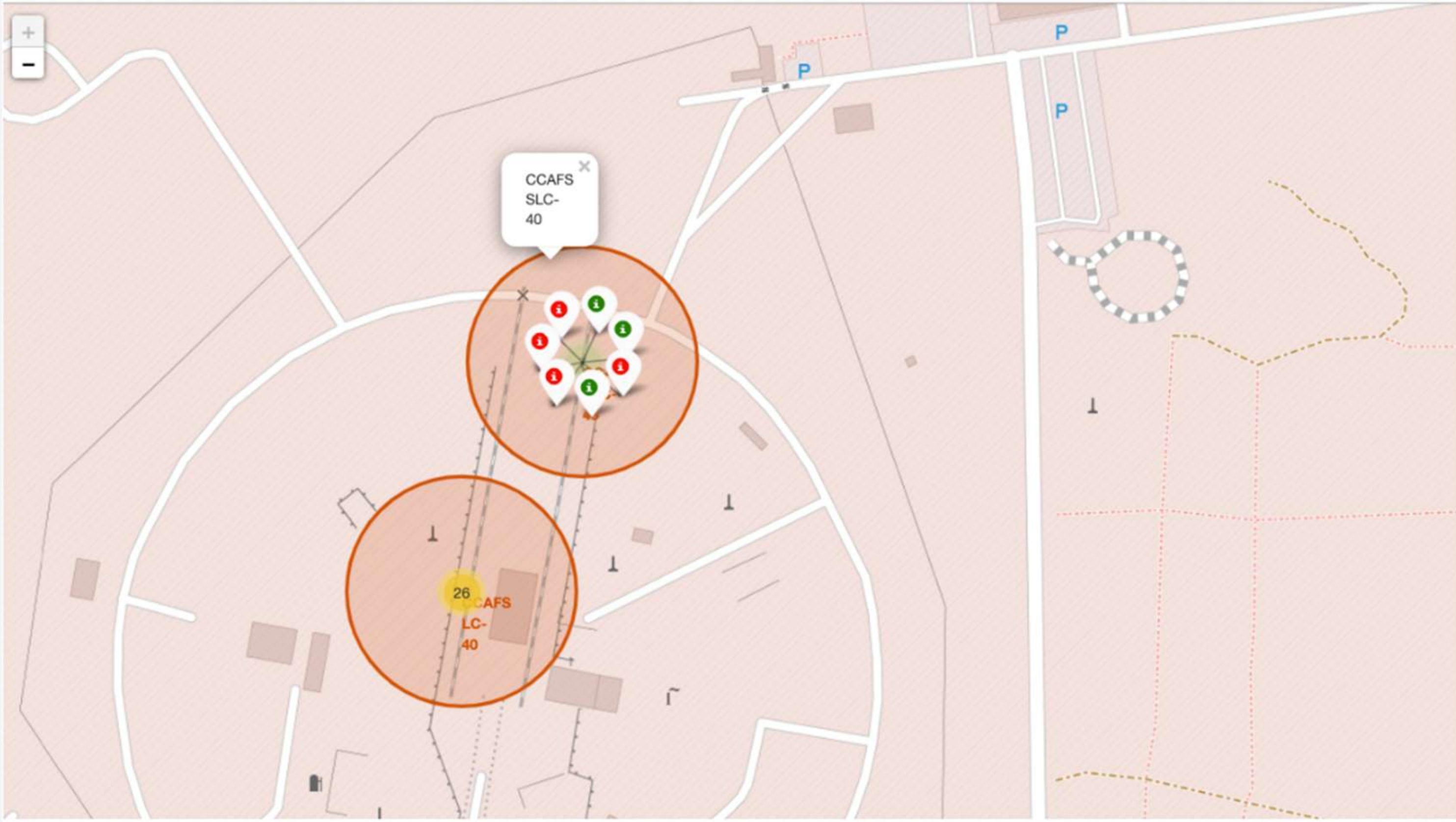
Launch Sites Proximities Analysis

SpaceX Falcon 9 launching sites



- NASA launching site and the 4 Space X launching sites are marked on the map
- In Floriada, Space X built its launching sites close to NASE ones
- All sites are close to a coastline, and as southern as possible (to be as close to the equator as possible, maximizing succesful landings)

Landing outcomes for each SpaceX launching site



- We grouped markers by launching site to illustrate success/failure of the landing
- User can easily zoom in/out to see outcomes for each site
- Most successful site was KSC

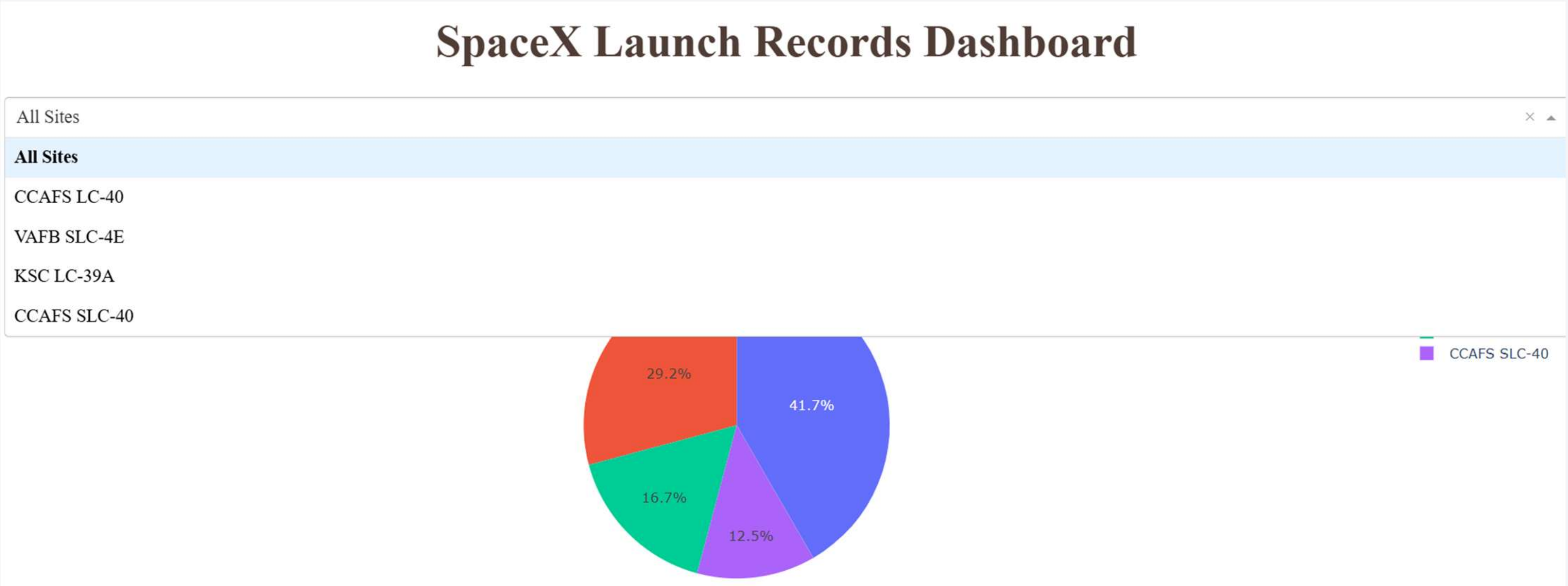
Proximity to crucial points in case of failure

- We showed distances to coastline, road, railway or city with an adhoc function and by drawing polylines
- We can see tha, in case of failure, launching sites are close to a coastline and far from roads/railways/cities

Section 4

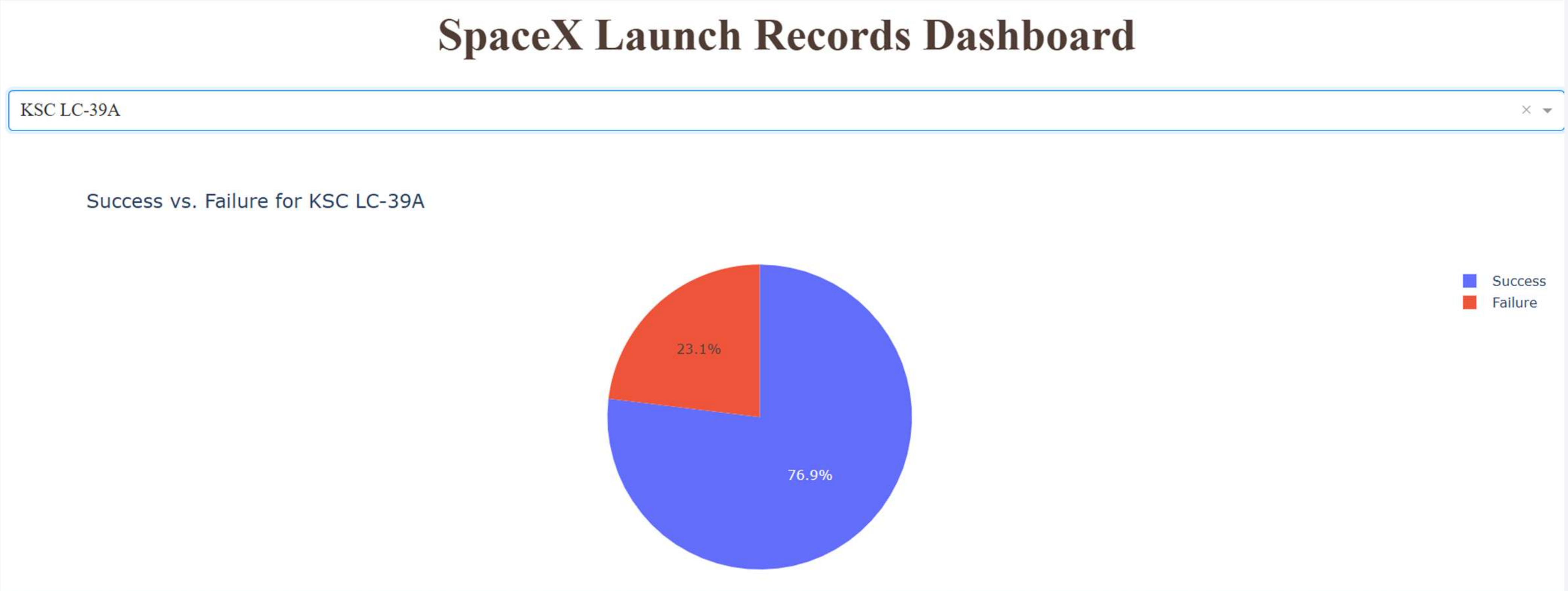
Build a Dashboard with Plotly Dash

Launch success count for all sites



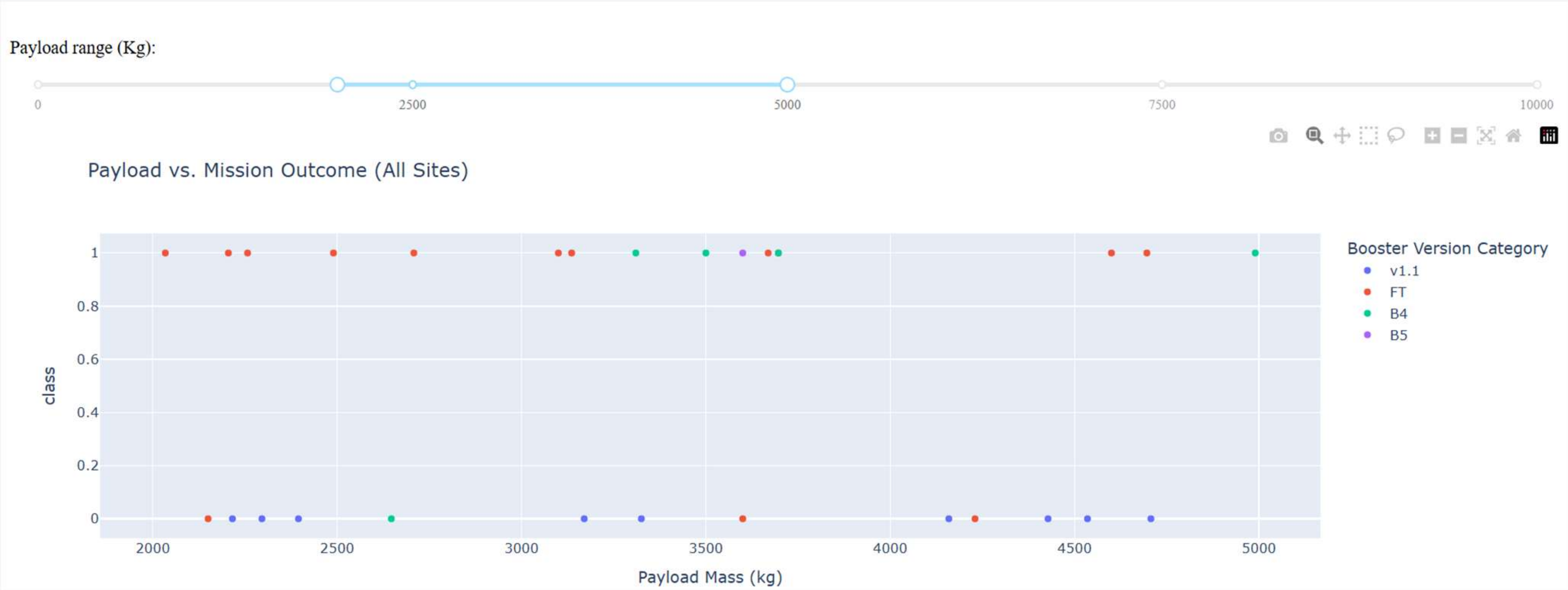
- CCAFS LC40 registered the most successful launches (in total)

Launching Site with the highest success ratio



- KSC LC 39A was the most successful of the sites

Success according to Payload Mass, all site combined



- The most successful Payload Range was 2800-3800 with 66%

Section 5

Predictive Analysis (Classification)

Classification Accuracy

```
logreg_cv.score(X_test, Y_test)
```

```
0.8333333333333334
```

```
svm_cv.score(X_test, Y_test)
```

```
0.8333333333333334
```

```
tree_cv.score(X_test, Y_test)
```

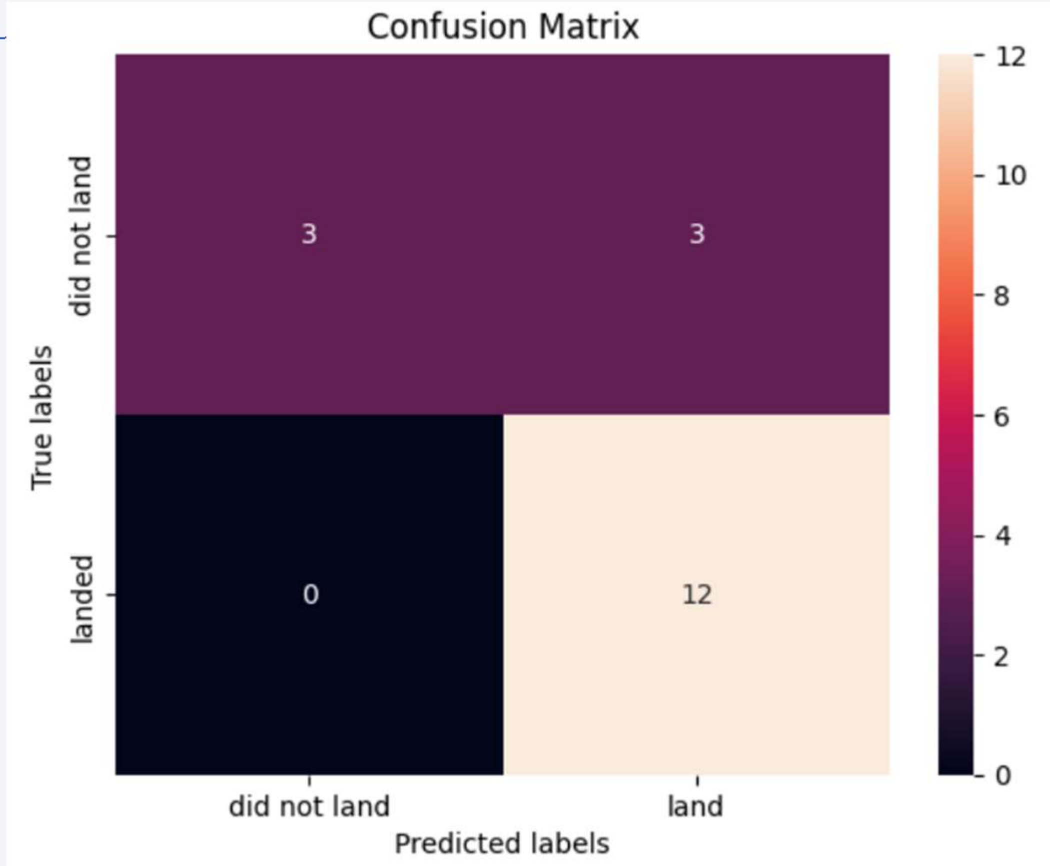
```
0.7222222222222222
```

```
knn_cv.score(X_test, Y_test)
```

```
0.8333333333333334
```

- Best models are Logistic Regression, SVM and KNN

Confusion Matrix



- All 3 models had the same confusion matrix

Conclusions

- Space X data were available for scraping and through API, allowing for exploratory and predictive analyses
- We collected pretty insightful variables to predict successful landings, based on plots and analyses
- Space X launching sites look pretty well located (near the NASA one, which is smart)
- Our machine learning models all predicted successful landing with a pretty good accuracy. But we would not need way more data to conclude, the balance costs vs risks being super high here

Appendix

- Include any relevant assets like Python code snippets, SQL queries, charts, Notebook outputs, or data sets that you may have created during this project

Thank you!