Winning Space Race with Data Science *And Style*

Kai Karius

30.06.2022

Outline

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

Executive Summary

- Data was collected using web scraping a the SpaceX REST API. Both data sources were cleaned, processed and combined. EDA through visual means and SQL was performed. Once promising features were identified, several classifiers were trained, tuned and compared. A dashboard using the plotly library was employed.
- Data collection yielded a set of X observations and EDA provided several promising features for ML models. The most high quality model was identified providing a predictive tools. An interactive dashboard for user site data exploration was created.

Introduction

- Space X advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because Space X can reuse the first stage. Therefore if we can determine if the first stage will land, we can determine the cost of a launch.
- Is there enough data with good features to train a classifier that predicts successful first stage landings?

Section 1

Methodology

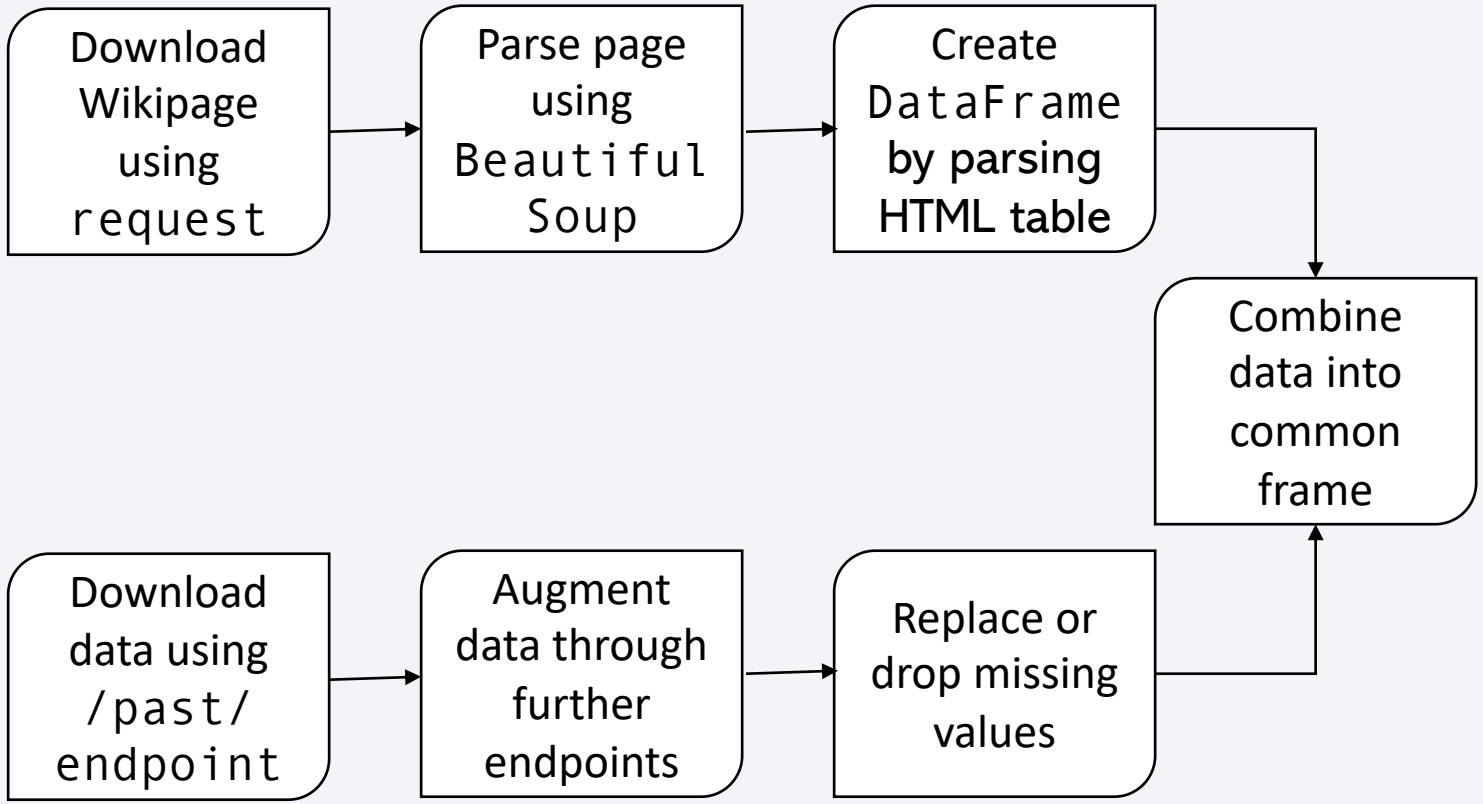
Methodology

Executive Summary

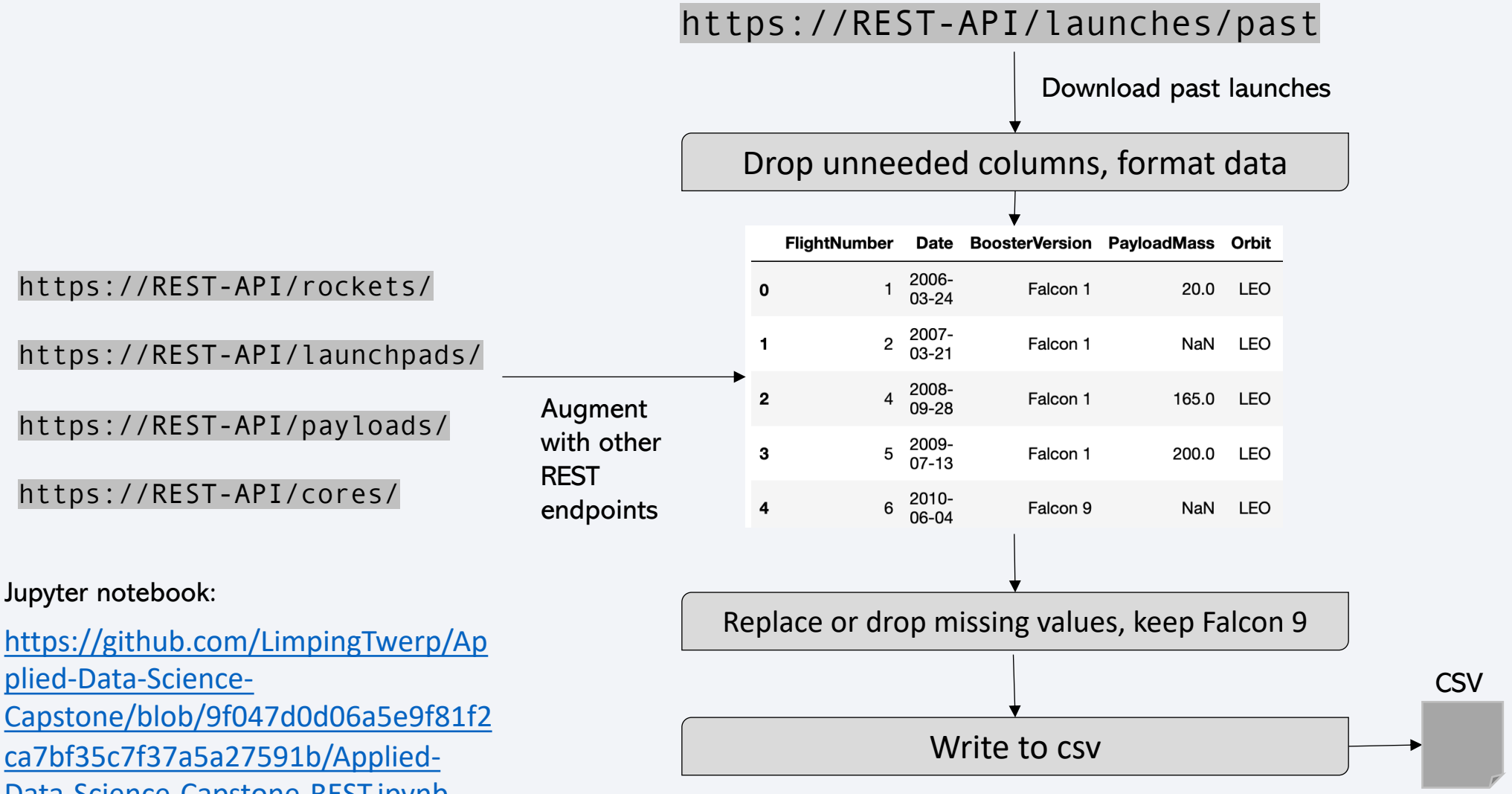
- Data collection methodology:
  - A REST API provided by SpaceX to gather data about launches
  - Web scraping on Wikipedia provided further data
- Data wrangling steps:
  - Missing data was replaced by its means, one hot encoding for categorical data
  - Only Falcon 9 launches were considered
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Trained, scored and tuned SVM, kNN, logistic regression and decision tree classifies

Data Collection

- A double approach was pursued, using web scraping on Wikipedia and the REST API of SpaceX
- Using web scraping from Wikipedia and BeautifulSoup, multiple observations of launches and their dates, engines, payloads asf. could be collected
- The REST apis of SpaceX provided launch data and augmentative data



Data Collection – SpaceX API



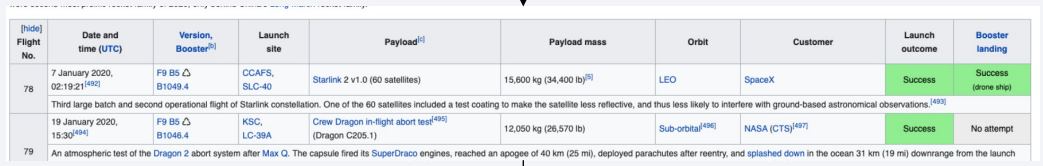
Jupyter notebook:

https://github.com/LimpingTwerp/Applied-Data-Science-Capstone/blob/9f047d0d06a5e9f81f2ca7bf35c7f37a5a27591b/Applied-Data-Science-Capstone-REST.ipynb

Data Collection - Scraping

- Get the Wikipedia html page content using `request`
- parse it using `BeautifulSoup`
- Find tables, use header element to create `dictionary` with column headers as keys
- Iterate through all rows to fill `dictionary` with values, then create `DataFrame`
- Get `DataFrame`

```
response = requests.get(static_url)
```



[hide] Flight No.	Date and time (UTC)	Version, Booster[b]	Launch site	Payload[c]	Payload mass	Orbit	Customer	Launch outcome	Booster landing
78	7 January 2020, 02:19:21[492]	F9 B5 △ B1049.4	CCAFS, SLC-40	Starlink 2 v1.0 (60 satellites)	15,600 kg (34,400 lb)[5]	LEO	SpaceX	Success	Success (drone ship)
	Third large batch and second operational flight of Starlink constellation. One of the 60 satellites included a test coating to make the satellite less reflective, and thus less likely to interfere with ground-based astronomical observations.[493]								
79	19 January 2020, 15:30[494]	F9 B5 △ B1046.4	KSC, LC-39A	Crew Dragon in-flight abort test[495] (Dragon C205.1)	12,050 kg (26,570 lb)	Sub-orbital[496]	NASA (CTS)[497]	Success	No attempt
	An atmospheric test of the Dragon 2 abort system after Max Q. The capsule fired its SuperDraco engines, reached an apogee of 40 km (25 mi), deployed parachutes after reentry, and splashed down in the ocean 31 km (19 mi) downrange from the launch								

```
soup = BeautifulSoup(response.content)
```

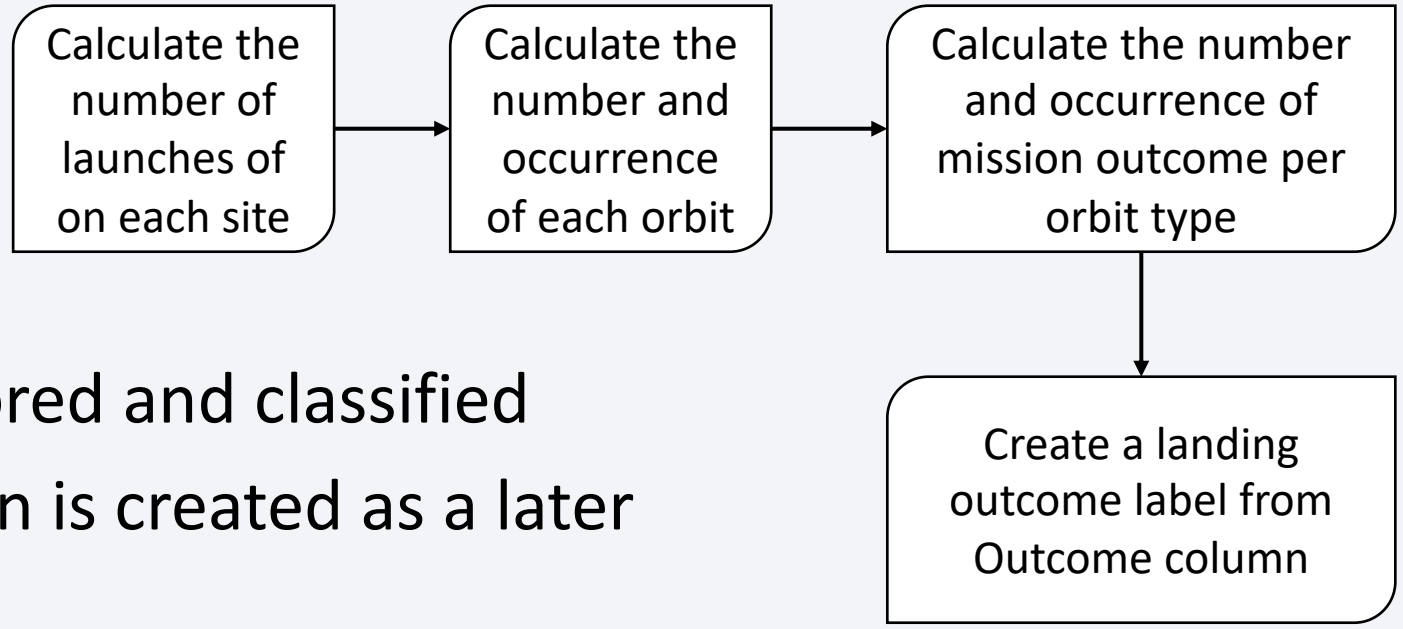
```
launch_dict= dict.fromkeys(column_names)

# Remove an irrelevant column
del launch_dict['Date and time ( )']

# Let's initial the launch_dict with each
launch_dict['Flight No.'] = []
launch_dict['Launch site'] = []
launch_dict['Payload'] = []
launch_dict['Payload mass'] = []
launch_dict['Orbit'] = []
launch_dict['Customer'] = []
launch_dict['Launch outcome'] = []
# Added some new columns
launch_dict['Version Booster']=[]
launch_dict['Booster landing']=[]
launch_dict['Date']=[]
launch_dict['Time']=[]
```

Flight No.	Launch site	Payload	Payload mass	Orbit
1	CCAFS	Dragon Spacecraft Qualification Unit	0	LEO
2	CCAFS	Dragon	0	LEO
3	CCAFS	Dragon	525 kg	LEO
4	CCAFS	SpaceX CRS-1	4,700 kg	LEO
5	CCAFS	SpaceX CRS-2	4,877 kg	LEO

Jupyter notebook:

https://github.com/LimpingTwerp/Applied-Data-Science-Capstone/blob/9f047d0d06a5e9f81f2ca7bf35c7f37a5a27591b/Applied-Data-Science-Capstone-Webscraping.ipynb

Data Wrangling



- The outcomes were explored and classified
- Using this, a target column is created as a later training target

Jupyter notebook:

https://github.com/LimpingTwerp/Applied-Data-Science-Capstone/blob/9f047d0d06a5e9f81f2ca7bf35c7f37a5a27591b/Applied-Data-Science-Capstone-Wrangling.ipynb

EDA with Data Visualization

- Columns were plotted against each other to see correlations among the features, such as shown above and recognize possible pattern. Among them are:
  - Flight Number vs Launch Success using a scatter plot
  - Payload Mass vs Launch Site using a scatter plot
  - Launch Success per Orbit type as a bar chart
  - Flight Number vs Orbit type as scatter plot
  - Payload Mass vs Orbit type as scatter plot
  - Years vs success rate

Jupyter notebook:

https://github.com/LimpingTwerp/Applied-Data-Science-Capstone/blob/9f047d0d06a5e9f81f2ca7bf35c7f37a5a27591b/Applied-Data-Science-Capstone-dataviz.ipynb

EDA with SQL

- the names of the unique launch sites in the space mission
- 5 records where launch sites begin with the string 'CCA'
- total payload mass carried by boosters launched by NASA (CRS)
- average payload mass carried by booster version F9 v1.1
- date when the first succesful landing outcome in ground pad was acheived
- names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
- total number of successful and failure mission outcomes
- names of the booster_versions which have carried the maximum payload mass
- count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017

Jupyter notebook:

https://github.com/LimpingTwerp/Applied-Data-Science-Capstone/blob/9f047d0d06a5e9f81f2ca7bf35c7f37a5a27591b/Applied-Data-Science-Capstone-SQL.ipynb

Build an Interactive Map with Folium

- Folium Circles were added at the coordinates of the launch sites to give a geographical overview
- Folium Markers and MarkerClusters were added to signify launches and encode their success as a color and further information as pop up
- Line segments to close by infrastructure were added to highlight geographical properties of launch sites

Jupyter notebook:

https://github.com/LimpingTwerp/Applied-Data-Science-Capstone/blob/9f047d0d06a5e9f81f2ca7bf35c7f37a5a27591b/Applied-Data-Science-Capstone-Folium.ipynb

Build a Dashboard with Plotly Dash

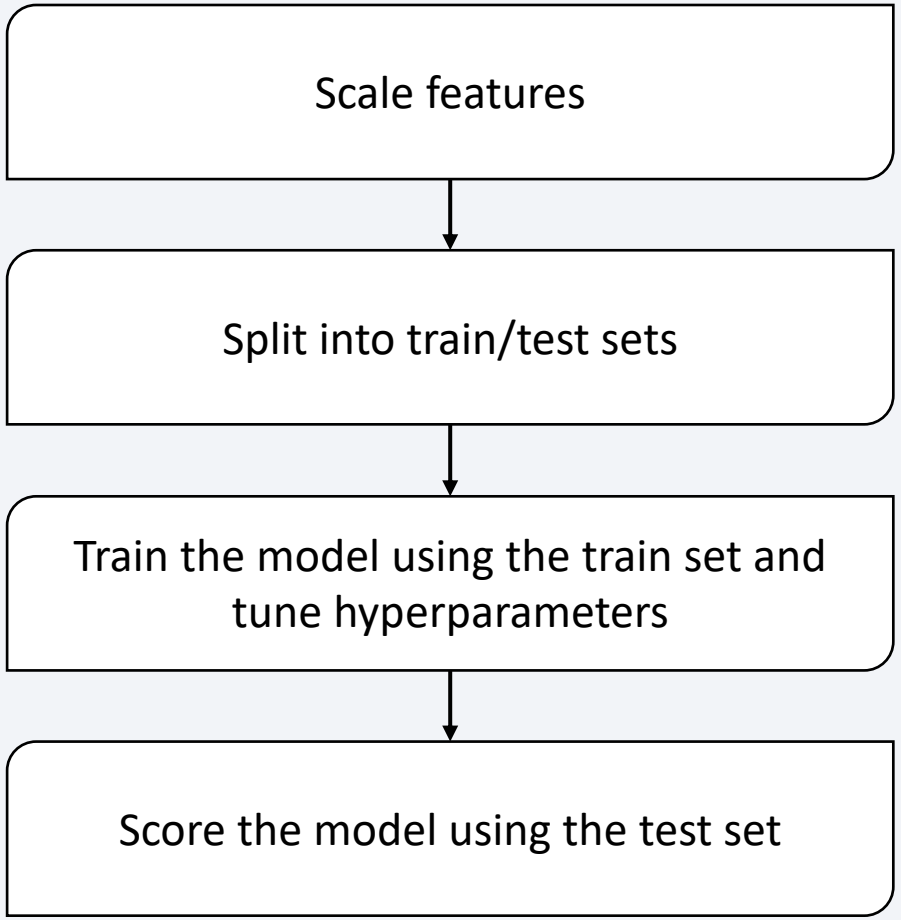
source:

https://github.com/LimpingTwerp/Applied-Data-Science-Capstone/blob/9f047d0d06a5e9f81f2ca7bf35c7f37a5a27591b/spacex_dash_app.py

- The dashboards consist of the following graphs:
  - A pie chart with success/failure for each launch site
  - A pie chart comparing the success of all sites
  - A Dropdown menu to switch between sites
  - A scatter plot of Payload (x) vs Success (y) as scatter plot with booster version as color coding
  - The Payload range of the scatter plot is controlled by a double slider
- The pie chart gives an overall overview of launch sites and their success rate and the payload vs success scatter plot gives an overview over the success rate as influenced by payload mass and booster version

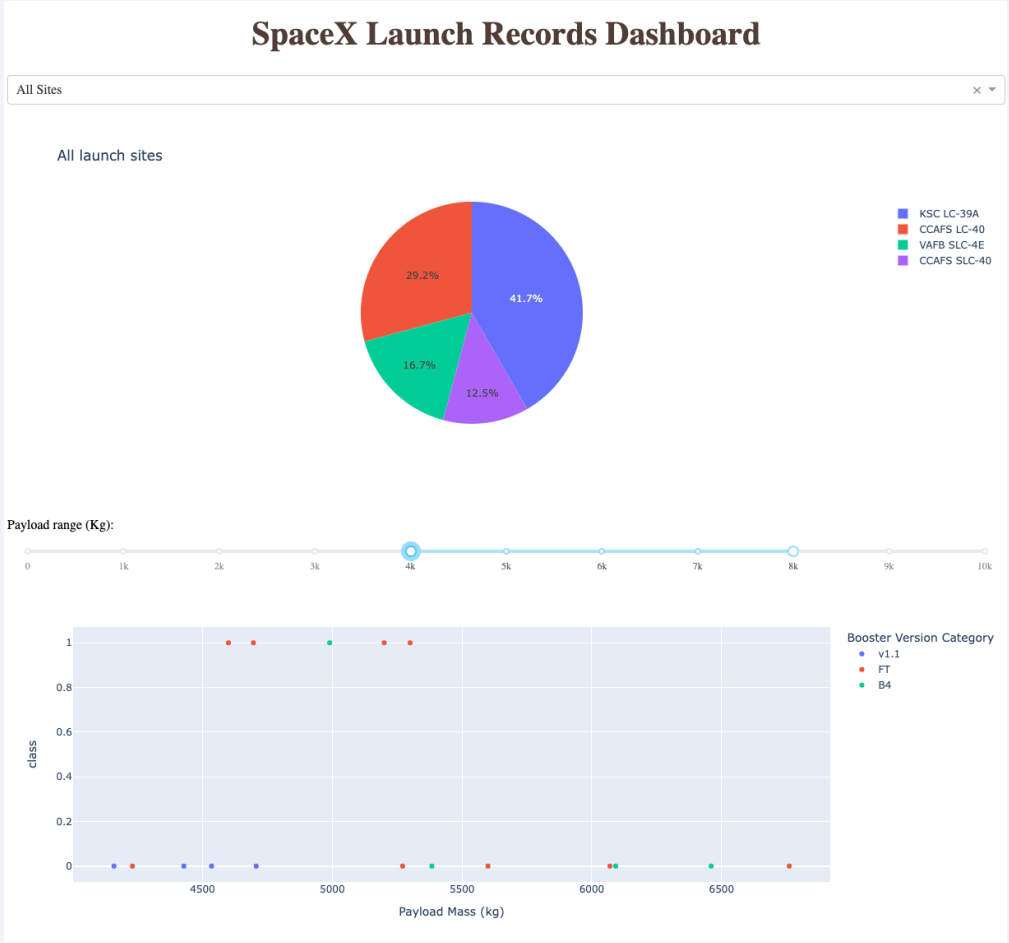
Predictive Analysis (Classification)

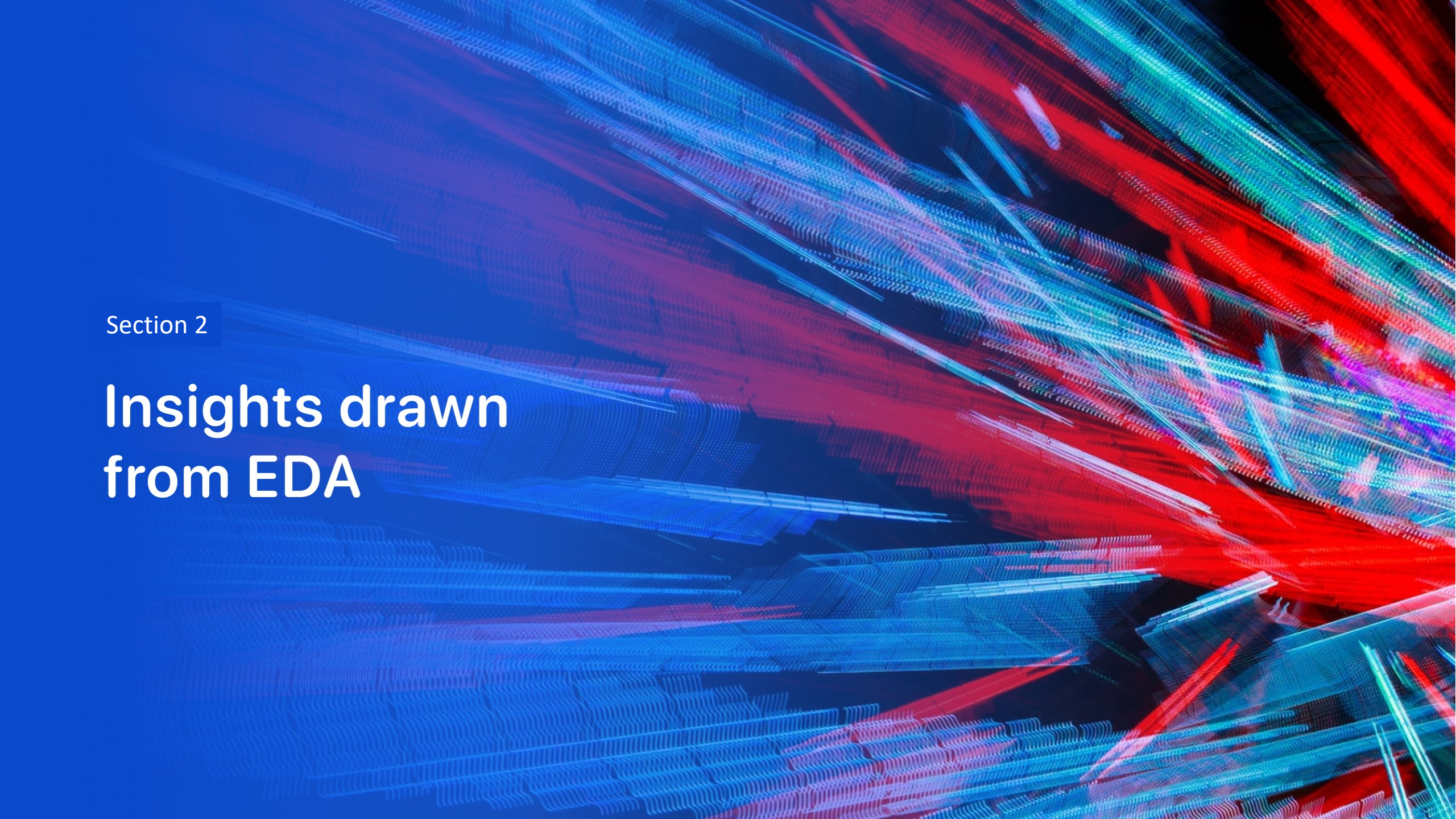
- After scaling and splitting the observations, the following models were training, tuned and scored:
  - kNN
  - Logistic regression
  - Decision Tree
  - SVM
- The accuracy and scores of each model were assessed



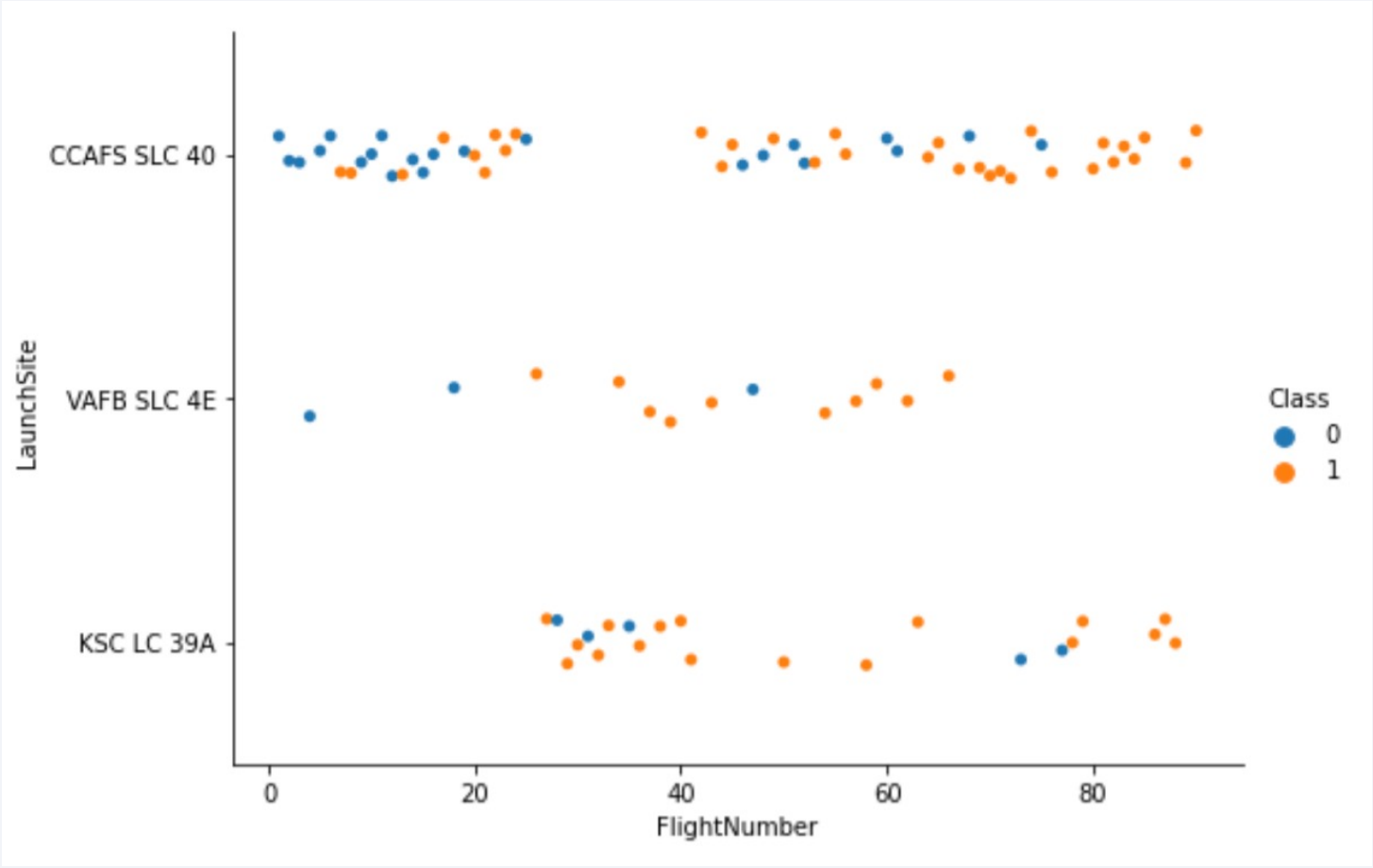
Results

- kNN, SVM and log-reg models perform the best in prediction
- EDA analysis results:
  - Lower payloads perform better
  - There is an increase of launch success over time
  - KSC LC 39A performs the most successful launches compared to other sites
  - Orbits GEO, HEO, SSO, ES L1 have the best success rate among orbits
- An interactive dashboard was created:



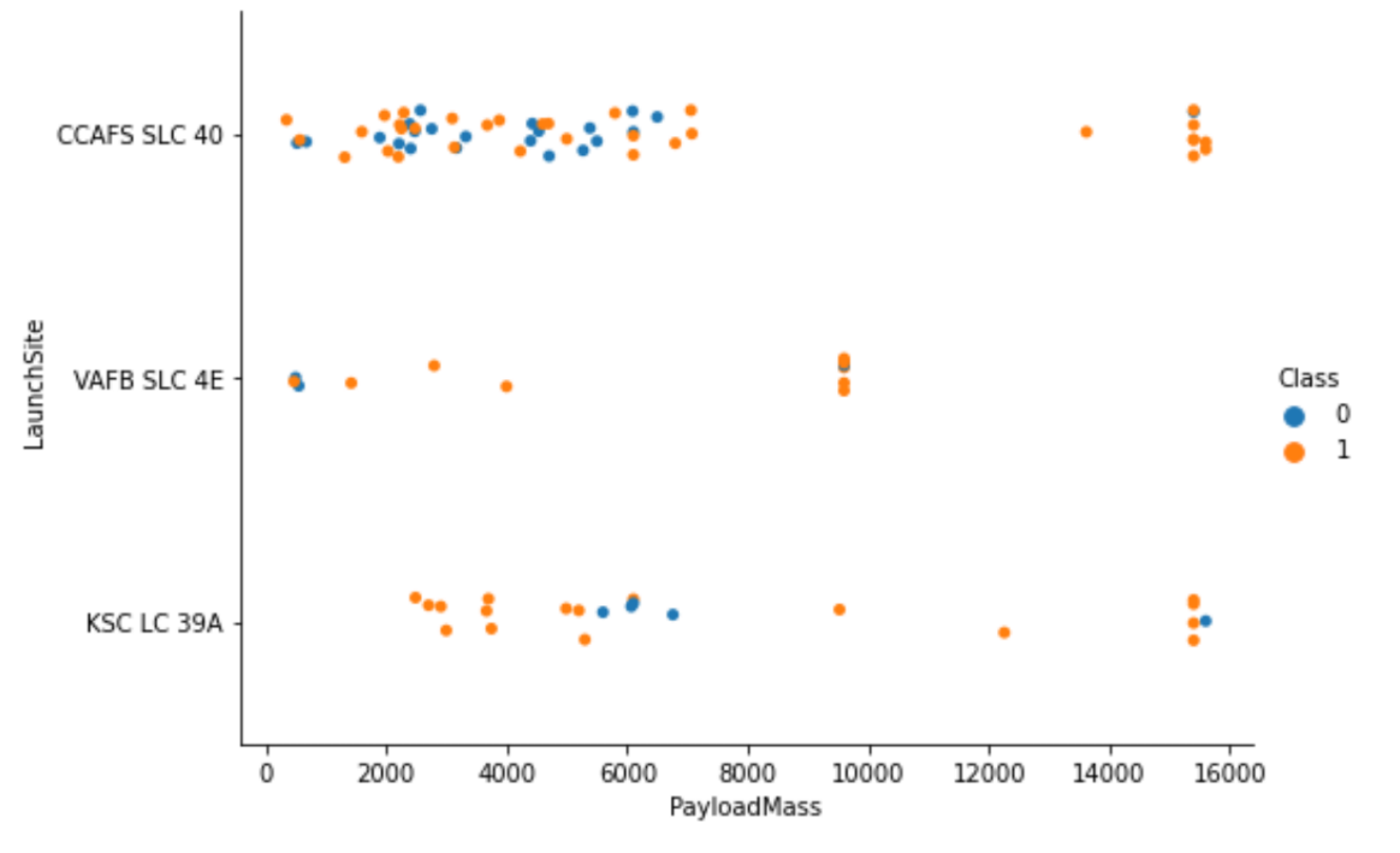

Section 2
Insights drawn from EDA

Flight Number vs. Launch Site



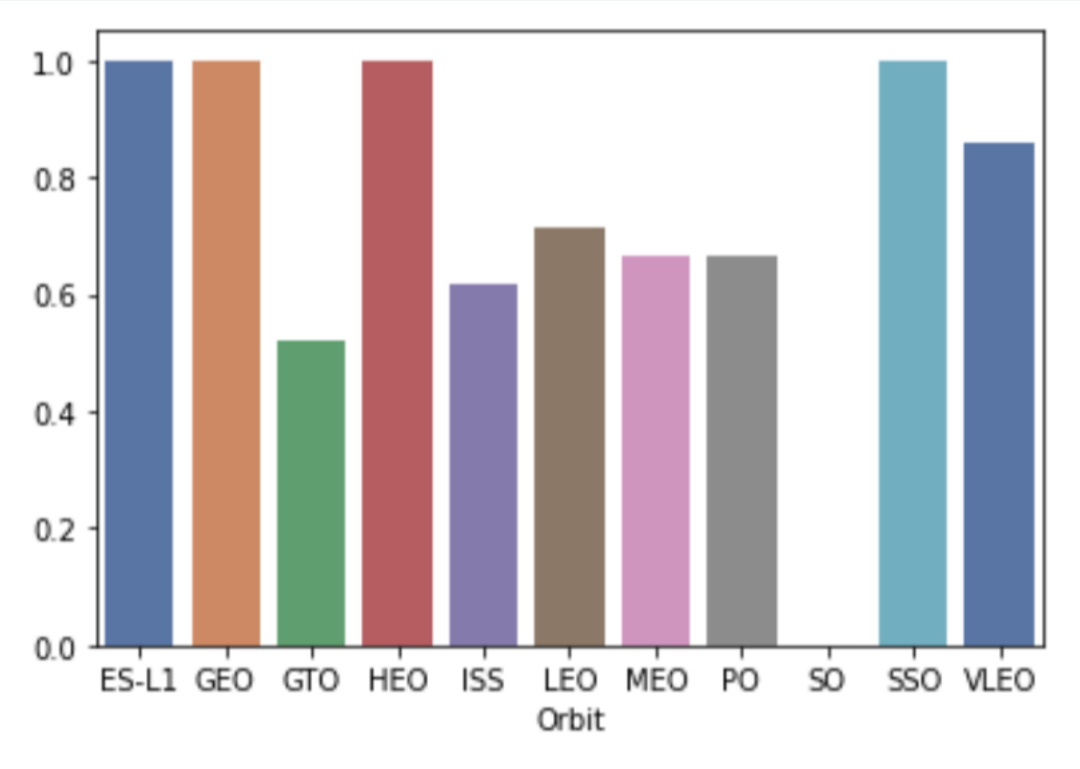
- CCAFS SLC 40 seems to be the oldest launch site and launches seem to have gotten more successful over the years.

Payload vs. Launch Site



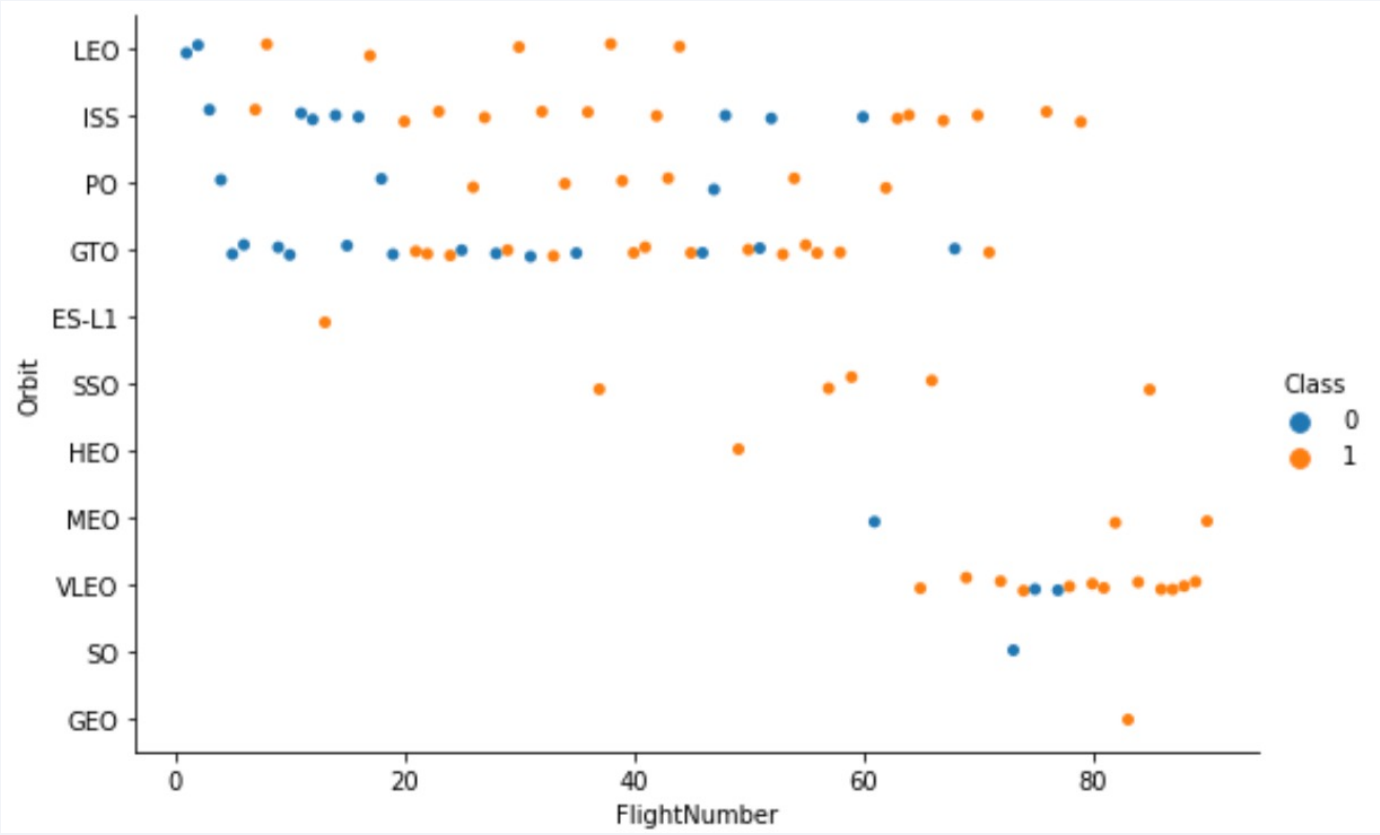
- CCAFS SLC 40 and KSC LC 39A service the heavy payloads, VAFB SLC 4E the mid range. Heavier payload == more success it seems

Success Rate vs. Orbit Type



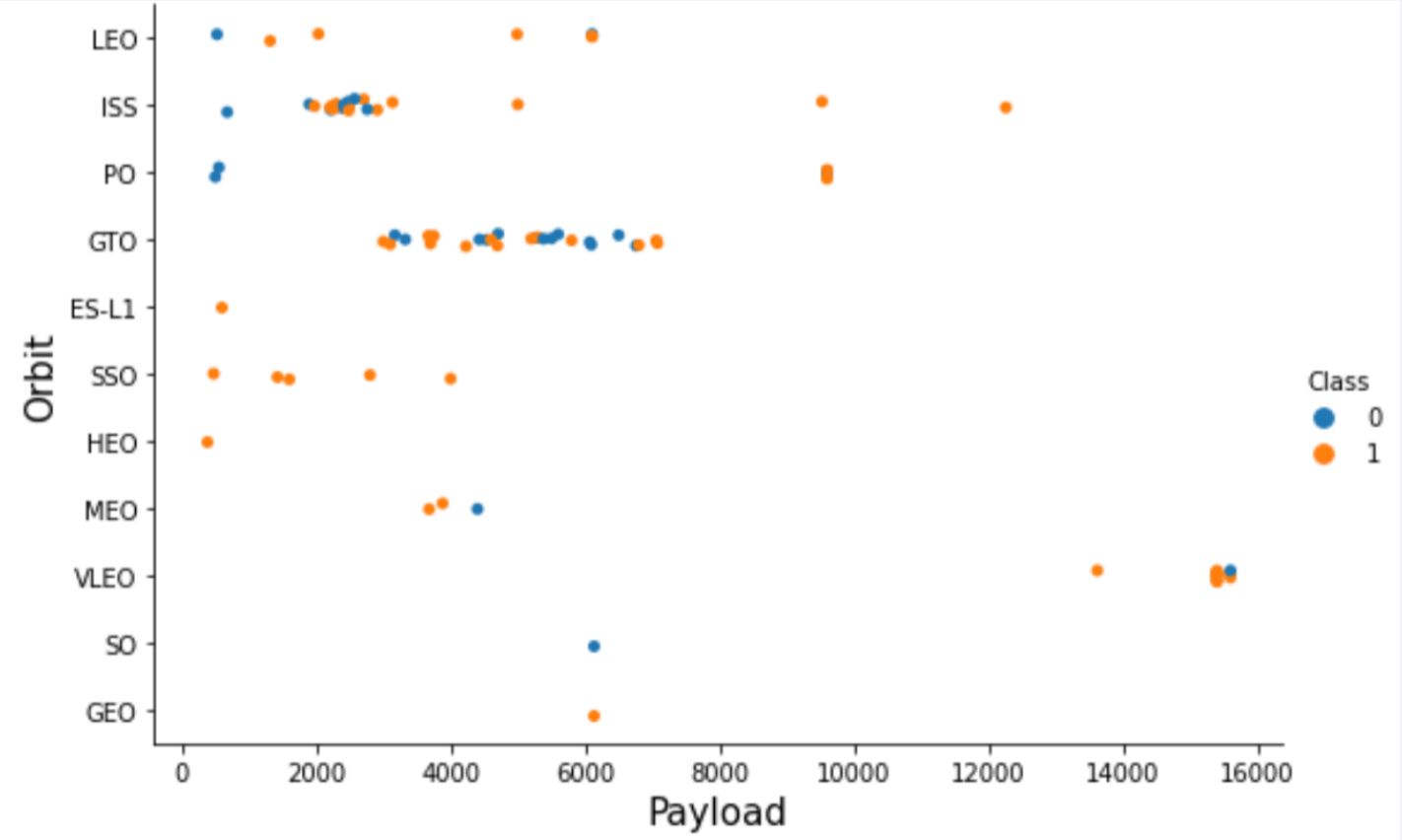
- ES-L1, GEO, HEO, SSO seem to be sae bets, don't attempt GTO
- Would have to check SO numbers to see if success rate == 0 or occurance

Flight Number vs. Orbit Type



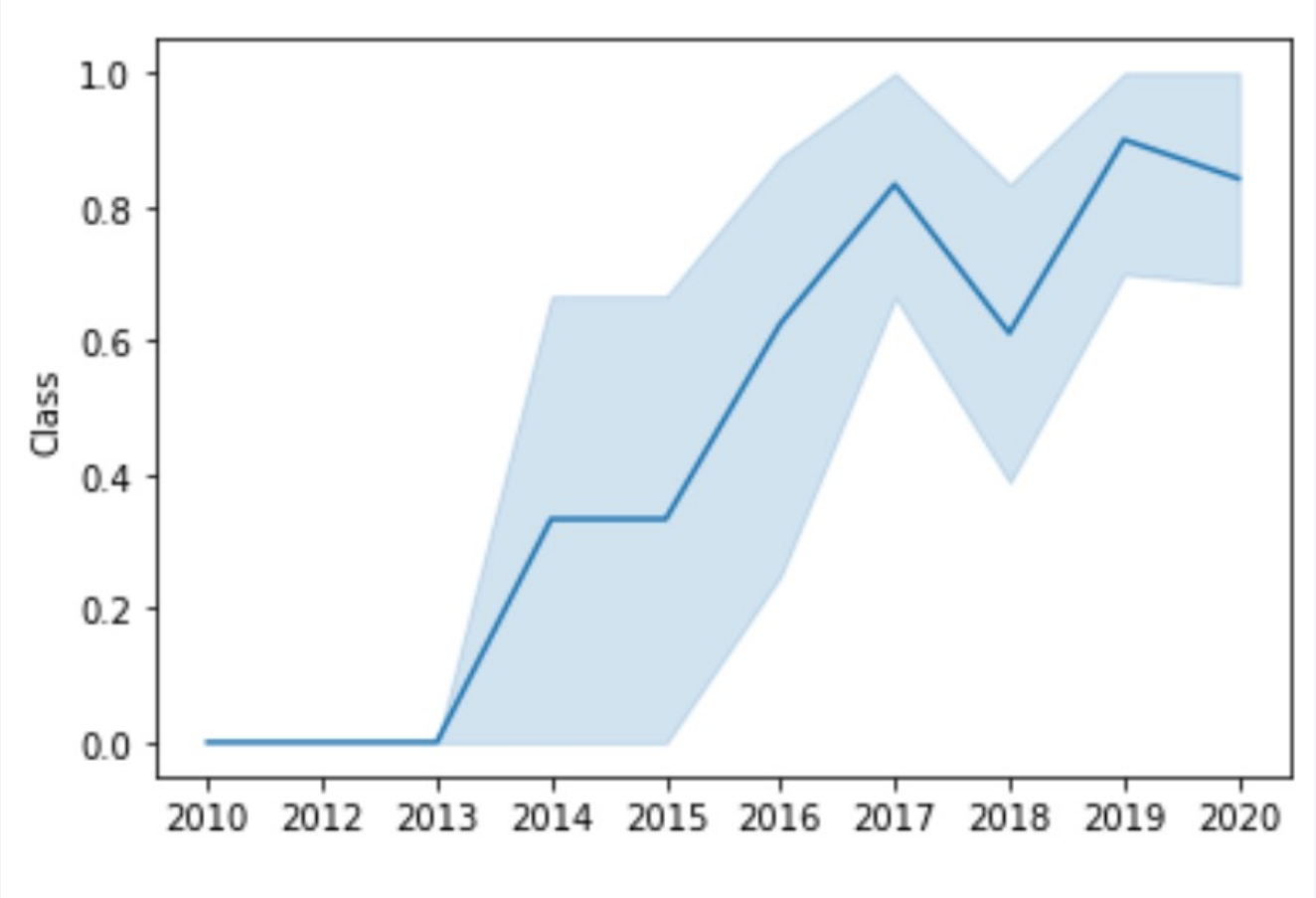
- VLEO more popular with high FlightNumbers, LEO seems to be working better
- GTO seems unrelated to FlightNumber

Payload vs. Orbit Type



- ISS seems to link to smaller payloads, mid payloads go to GTO, heavy payloads seem to go to VLEO

Launch Success Yearly Trend



Clear increase from 2015 onwards!

All Launch Site Names

```
%sql SELECT DISTINCT(Launch_Site) FROM SPACEXTBL;
```

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

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

- Listed: The unique names of all launch site, retrieved from the SQL database using the `DISTINCT` command

Launch Site Names Begin with 'CCA'

```
%sql SELECT * FROM SPACEXTBL 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
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

- 5 Launch sites instances beginning with 'CCA', using the `LIMIT` command for obtaining 5 instances and the `LIKE` command for string matching

Total Payload Mass

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL;
```

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

SUM(PAYLOAD_MASS__KG_)
619967

Calculating the total payload mass using the `SUM` function

Average Payload Mass by F9 v1.1

```
%%sql
SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTBL
WHERE "Booster_Version" LIKE "F9 v1.1%";
```

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

AVG(PAYLOAD_MASS__KG_)
2534.6666666666665

- Calculating the average payload mass carried by booster version F9 v1.1 using the AVG function, a WHERE clause and the LIKE function

First Successful Ground Landing Date

```
%%sql
SELECT MIN(Date) FROM SPACEXTBL
WHERE "Landing _Outcome" LIKE '%Success (ground pad)%';
```

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

MIN(Date)
01-05-2017

Finding the dates of the first successful landing outcome on ground pad using the `MIN` command, a `WHERE` clause with the `LIKE` command

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%%sql
SELECT "Booster_Version" FROM SPACEXTBL
WHERE "Landing _Outcome" LIKE '%Success (drone ship)%'
AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000;
```

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

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

- The names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000, using a `WHERE` clause, the `LIKE` and `BETWEEN` command

Total Number of Successful and Failure Mission Outcomes

```
%%sql
SELECT COUNT(Mission_Outcome) FROM SPACEXTBL WHERE Mission_Outcome LIKE '%Success%';
```

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

COUNT(Mission_Outcome)
100

```
%%sql
SELECT COUNT(Mission_Outcome) FROM SPACEXTBL WHERE Mission_Outcome NOT LIKE '%Success%';
```

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

COUNT(Mission_Outcome)
1

TOP: Successes, BOTTOM: Failures. Prominent use of the `NOT` operator.

Boosters Carried Maximum Payload

```
%%sql
SELECT DISTINCT(Booster_Version) FROM SPACEXTBL
WHERE PAYLOAD_MASS__KG_=(SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL);
```

```
 * 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

Booster Versions that carried maximum payload, using a subquery

2015 Launch Records

```
%%sql
SELECT substr(Date,4,2), "Landing _Outcome", Booster_Version, Launch_Site FROM SPACEXTBL
WHERE substr(Date,7,4)='2015'
AND "Landing _Outcome" LIKE 'Failure (drone ship)';
```

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

substr(Date,4,2)	Landing _Outcome	Booster_Version	Launch_Site
01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

- Listed: the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015, using the substr command

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

```
%%sql
SELECT "Landing _Outcome",COUNT(*) as _count FROM SPACEXTBL
WHERE Date BETWEEN '04-06-2010' AND '20-03-2017'
AND "Landing _Outcome" LIKE '%Success%'
GROUP BY "Landing _Outcome"
ORDER BY _count DESC;
```

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

Landing _Outcome	_count
Success	20
Success (drone ship)	8
Success (ground pad)	6

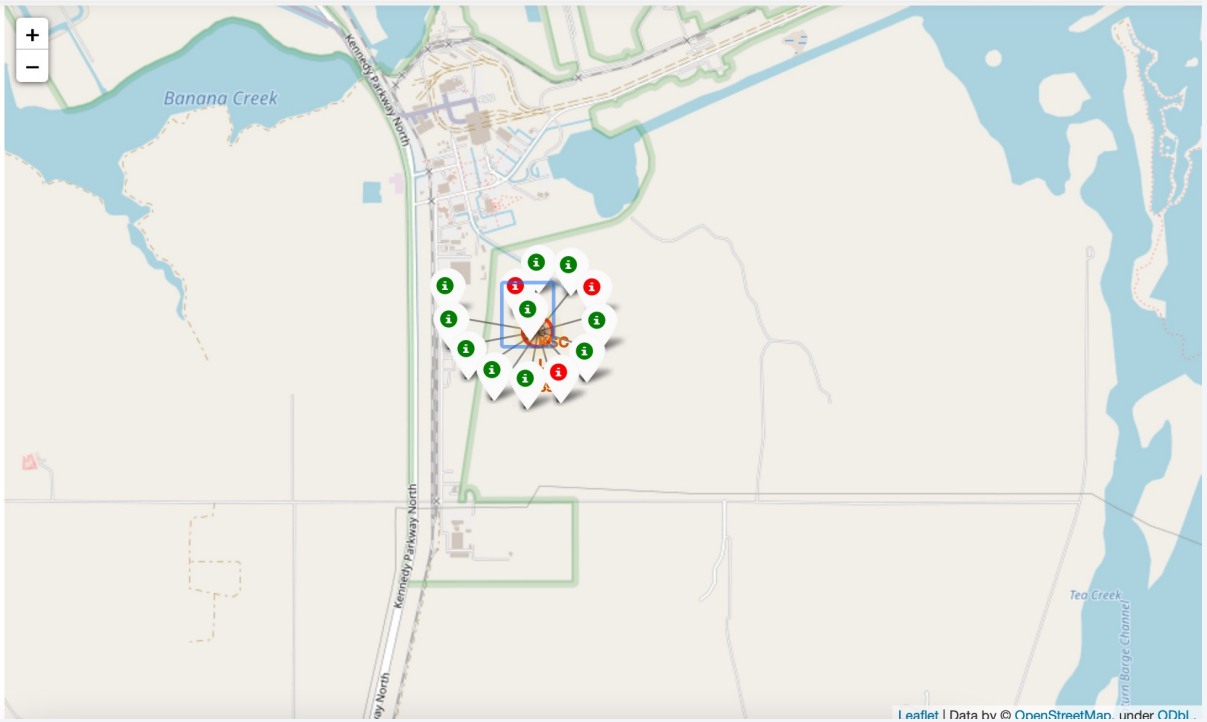
- Ranking the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order using a WHERE clause, the GROUP BY and ORDER BY statements


Section 3
Launch Sites
Proximities Analysis

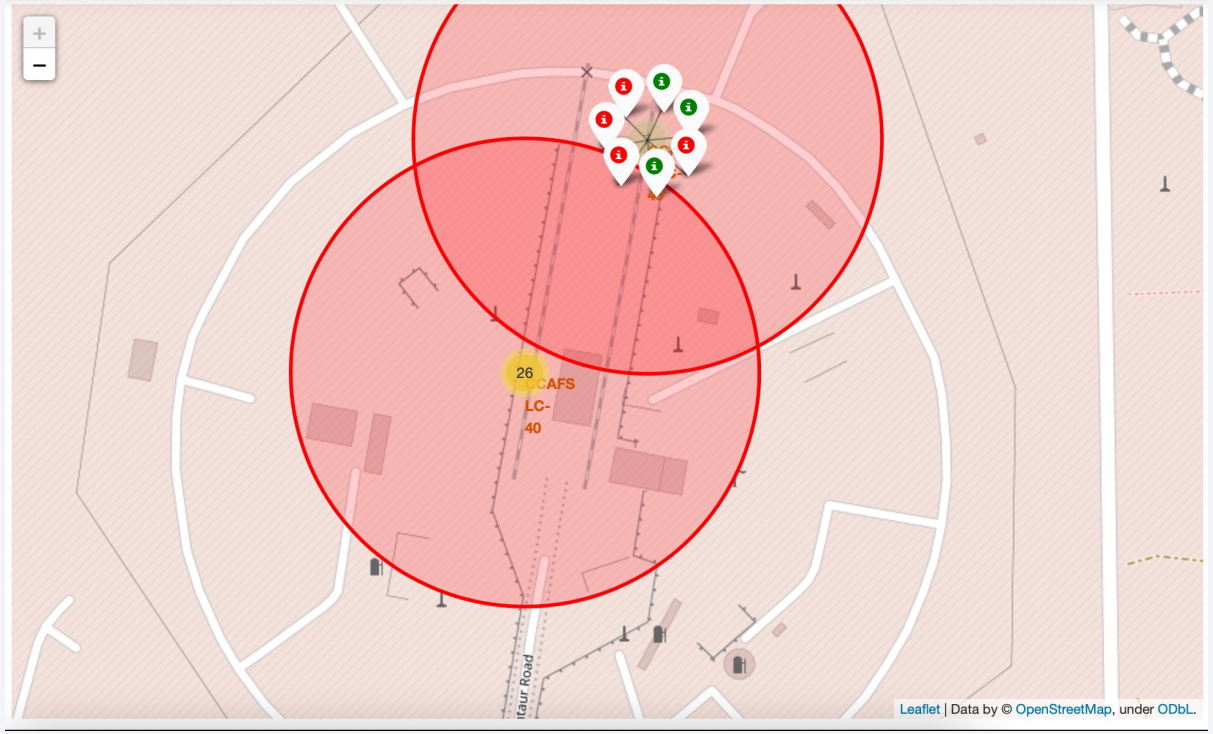
Distribution of launch sites within the US

- CCAFS LC-40,CCAFS SLC-40,KSC LC-39A are *really* close together, VAFB SLC-4E is somewhat more lonely
- The former are closer to the equator, giving more thrust
- Proximity to the coast for security reasons

Success of different launch sites



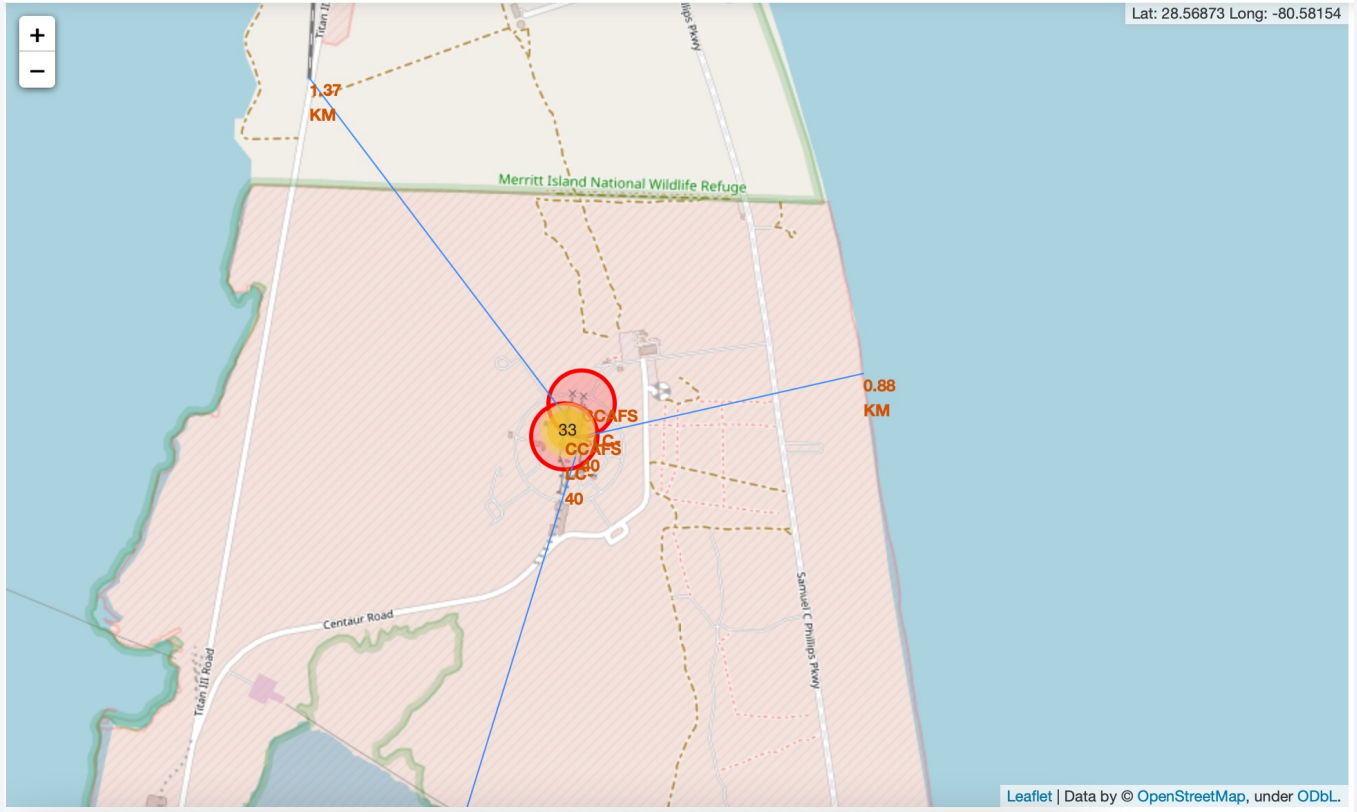
KSC LSC-39A



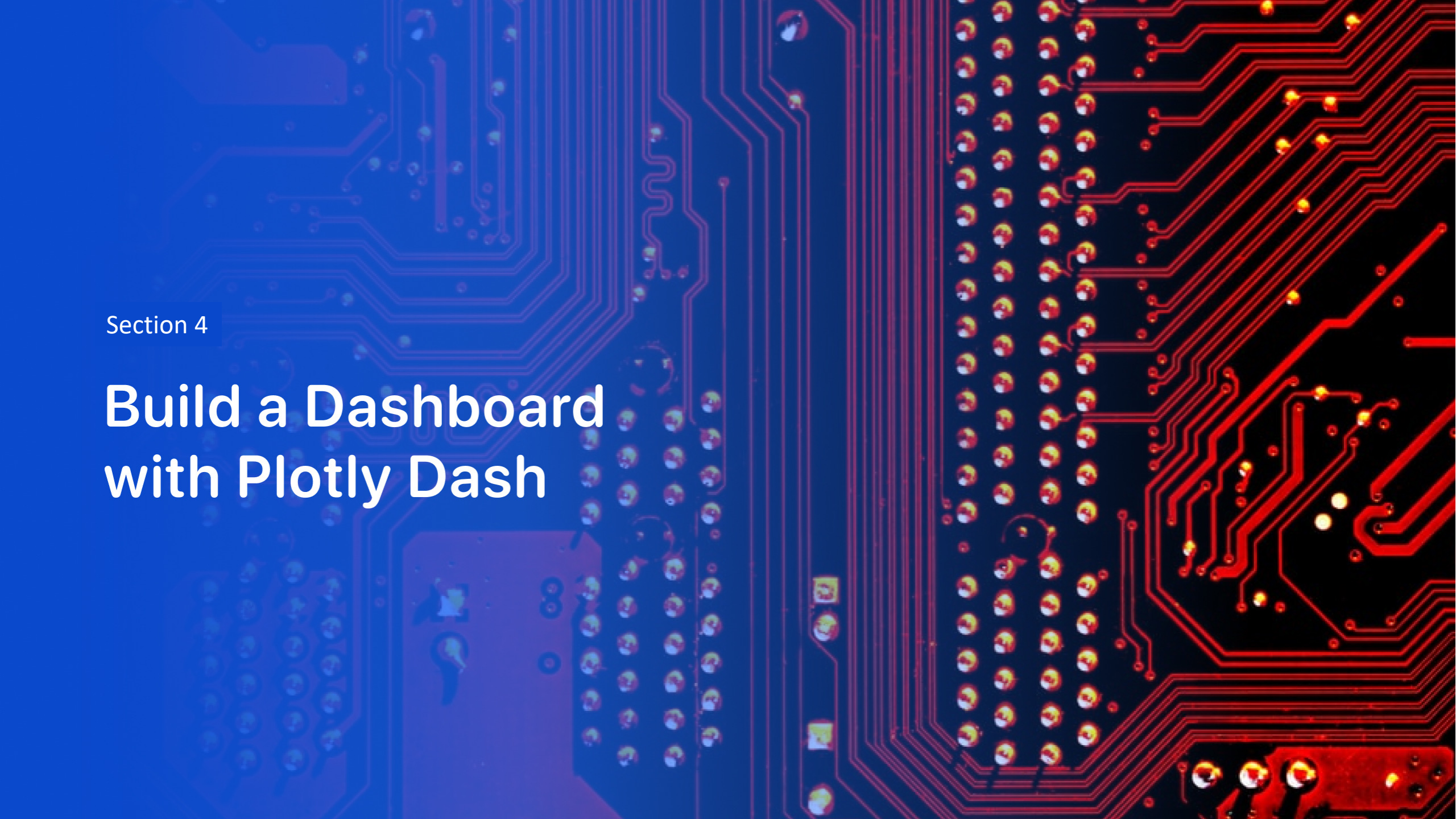
CCAFS SLC-40

- KSC LSC-39A very successful
- CCAFS SLC-40 kind of successful, CCAFS LC-40 not so much

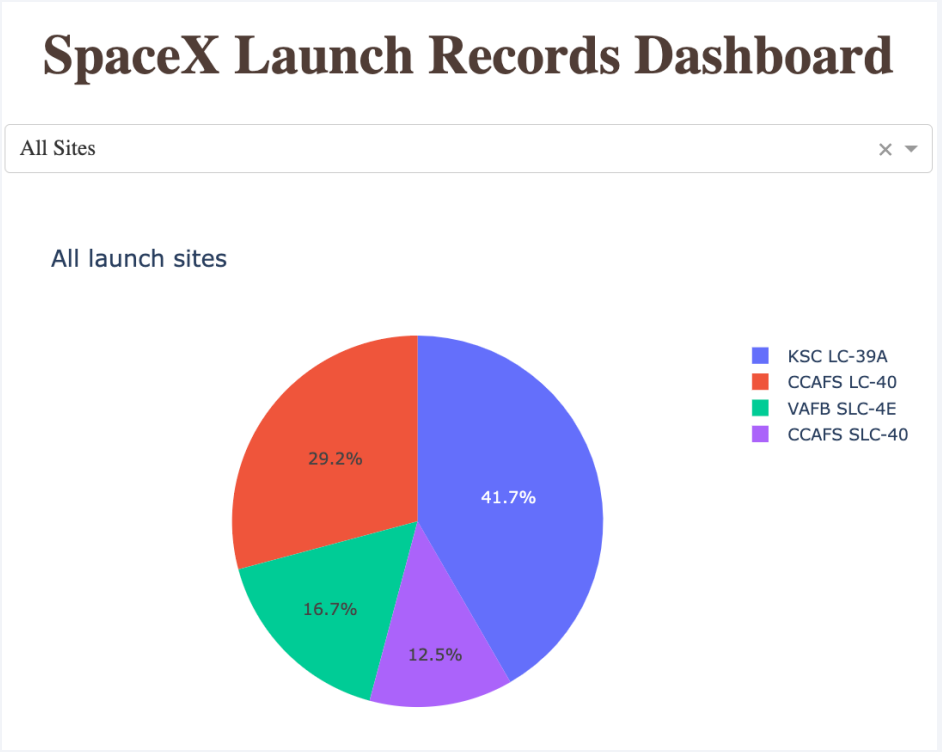
Distance to sites of interest



- Coastal proximity for security
- Railway and highway proximity for transport logistics
- Far away from towns and cities

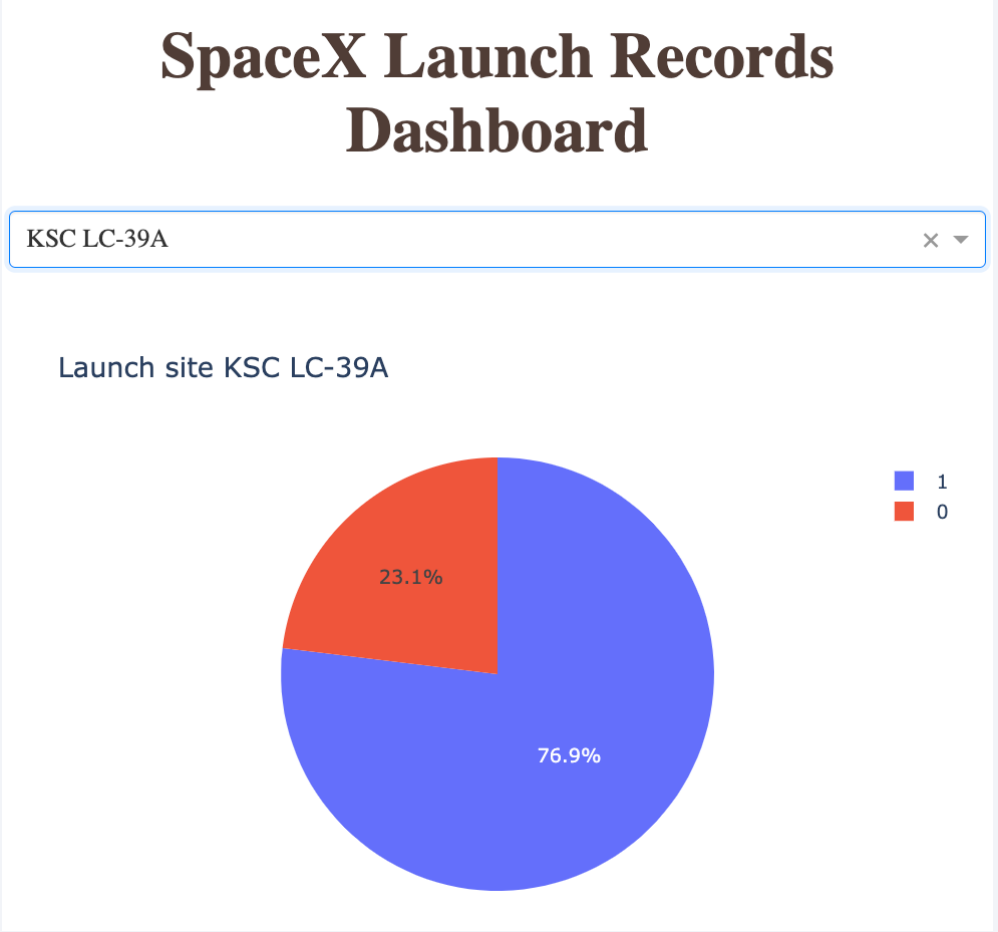

Section 4
Build a Dashboard with Plotly Dash

Launch success pie chart of all sites



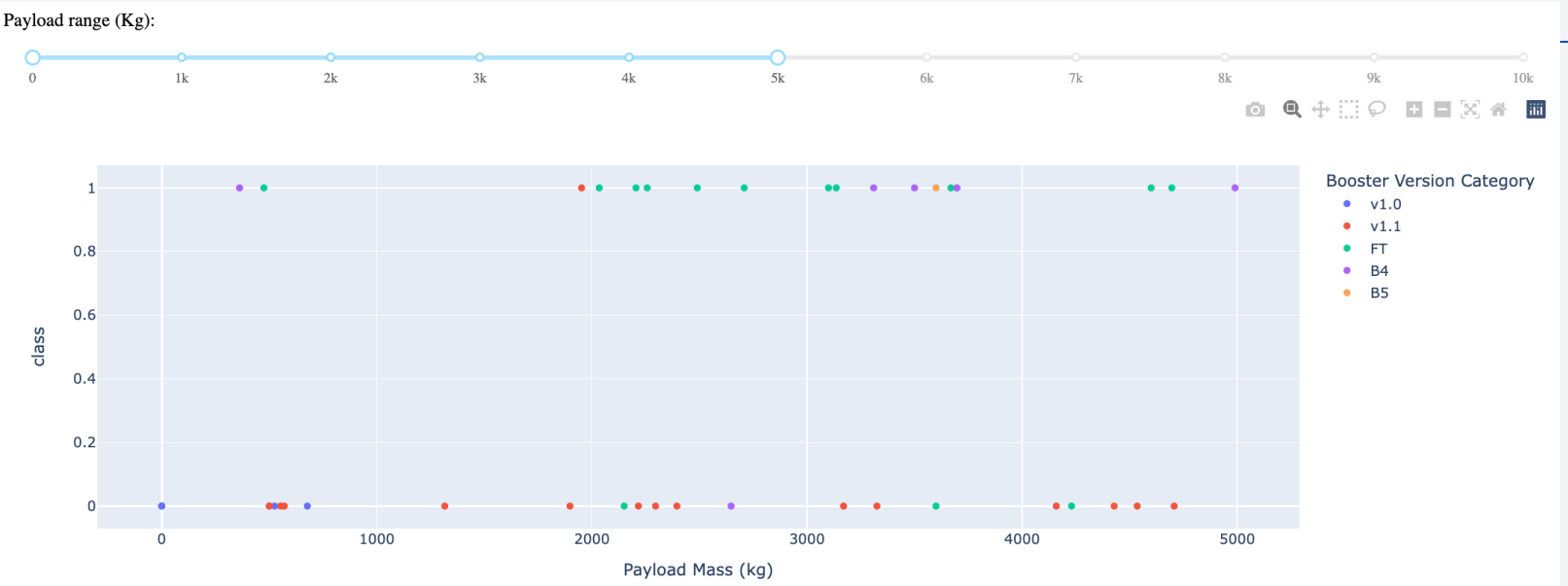
- Most successes are achieved by KSC LC-39, the least number of successes is achieved by CCAFS SLC-40

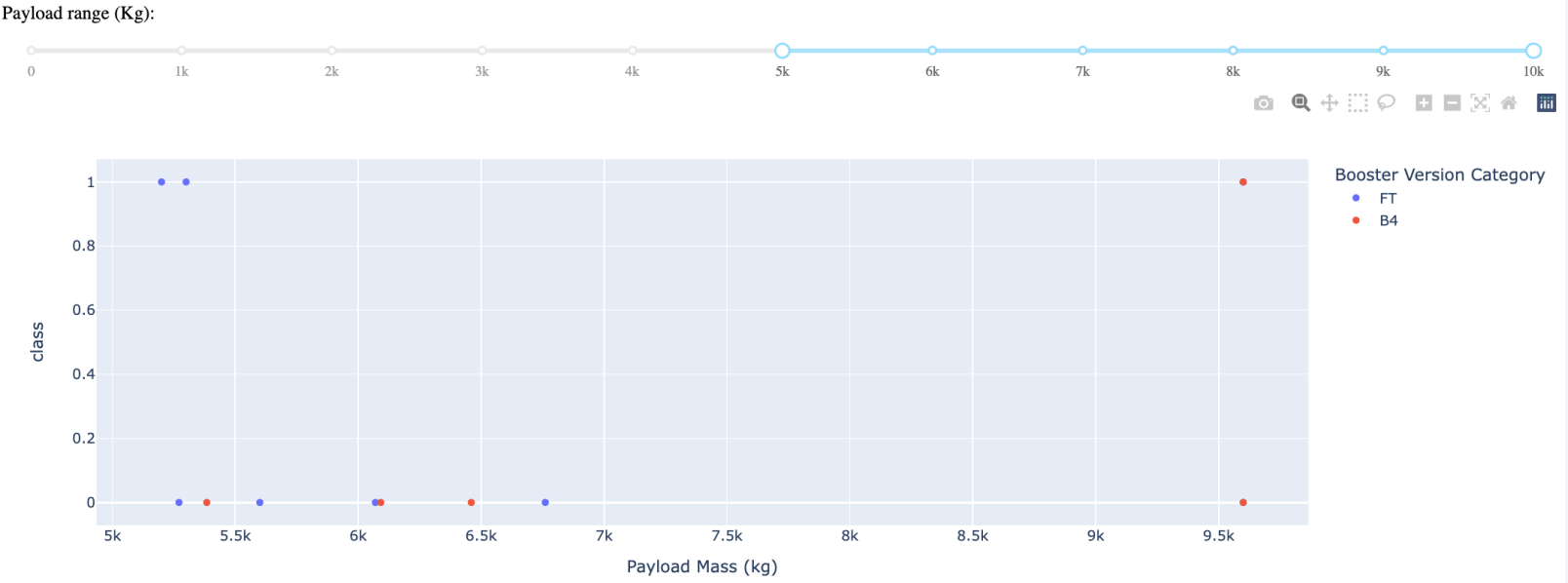
Success rate of the KSC LC-39A



Blue signifies success the percentage of launch success

Launch successes over time and booster versions





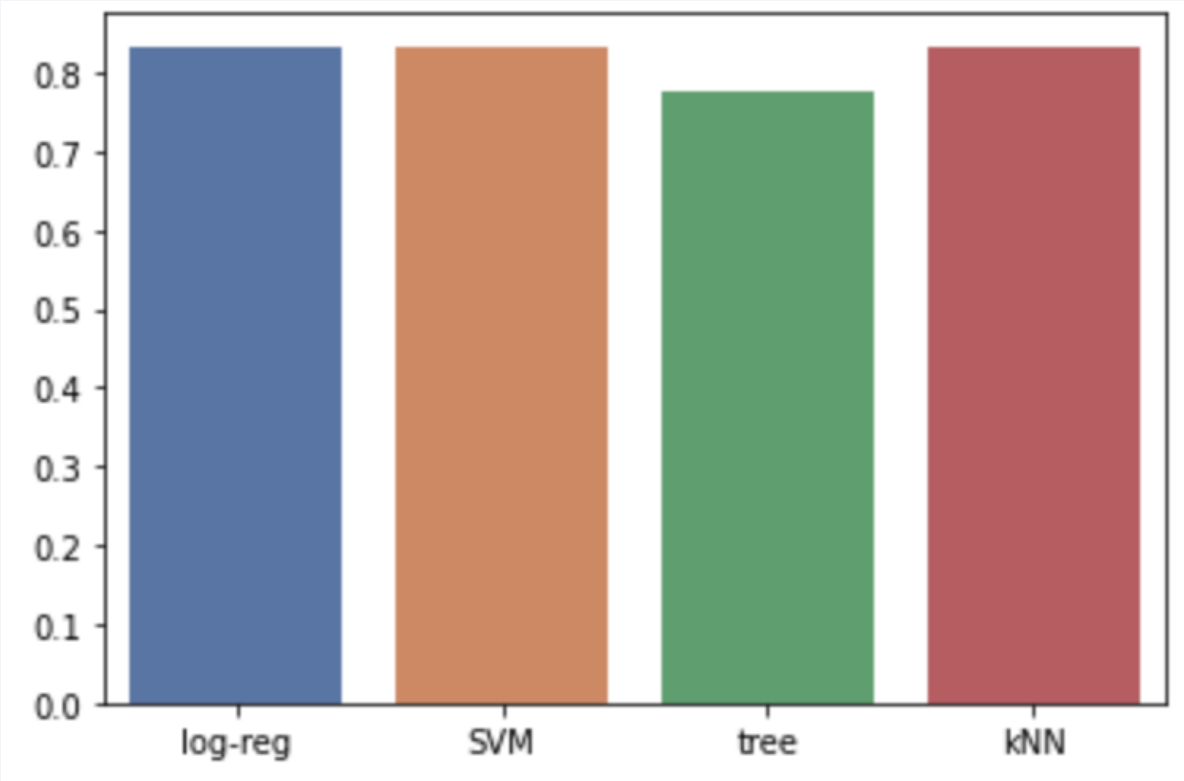
- Lower payloads (0-5000): v1.0, v1.1 boosters dominate

- Lower payloads (5000-10000): FT, B4 booster dominate

Section 5

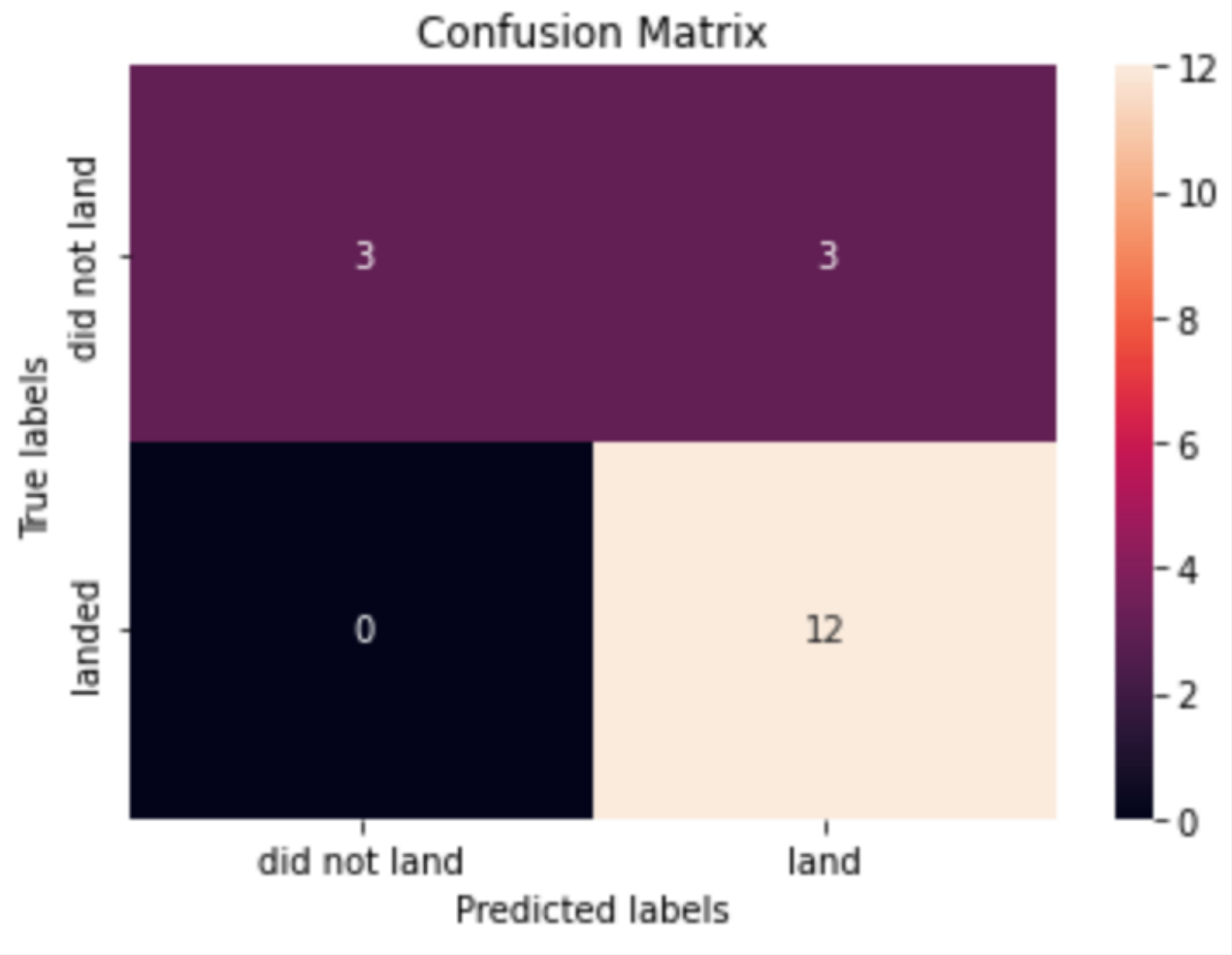
Predictive Analysis (Classification)

Classification Accuracy



- Log-reg, SVM and kNN perform in best after test score.

Confusion Matrix



- The confusion matrix of the kNN-model
- There is a problem with the false positives
- True negatives and false negatives can be distinguished very well

Conclusions

- Data is available via REST API and web scraping
- Payload mass, launch site, launch date and orbit are useful features for prediction
- A geographical analysis of launch sites and an interactive dashboard was provided
- kNN, SVM and logistic regression are trained and tuned models that score best against test data
- Further test cases and features are needed for higher precision predictions

Thank you!