IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Donnoban Maldonado
11/01/23

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection via API
  - Data Collection via web scraping
  - Data Wrangling
  - Exploratory Data Analysis via SQL and visualizations
  - Interactive Geospatial Analysis with Folium
  - Interactive Visual Analytics Dashboard with Dash
  - Predictive analysis using classification models
- Summary of results
  - Exploratory Data Analysis findings
  - Interactive Analytics findings
  - Predictive Analytics model

Introduction

- Project background and context

  Space exploration is a costly venture, or, to put it more intriguingly, it is a highly profitable pursuit. Typically, rocket launches can cost upwards of a whopping 165 million dollars. However, Space X boasts Falcon 9 rocket launches on its website at a cost of 62 million dollars. This is due to an innovative approach of reusing the first stage of a rocket. If we can accurately predict if the first stage will land, we can determine the cost of the launch. Since many of these launches are government contracted, this information could be used by a competing company to bid against Space X. Our goal in this project is to build a machine learning pipeline that can predict if the first stage of the Falcon 9 rocket will land successfully.

- Problems you want to find answers

  1. What factors determine if the first stage will land successfully?
  2. How accurately can we predict if the first stage will land?

Section 1

Methodology

Methodology

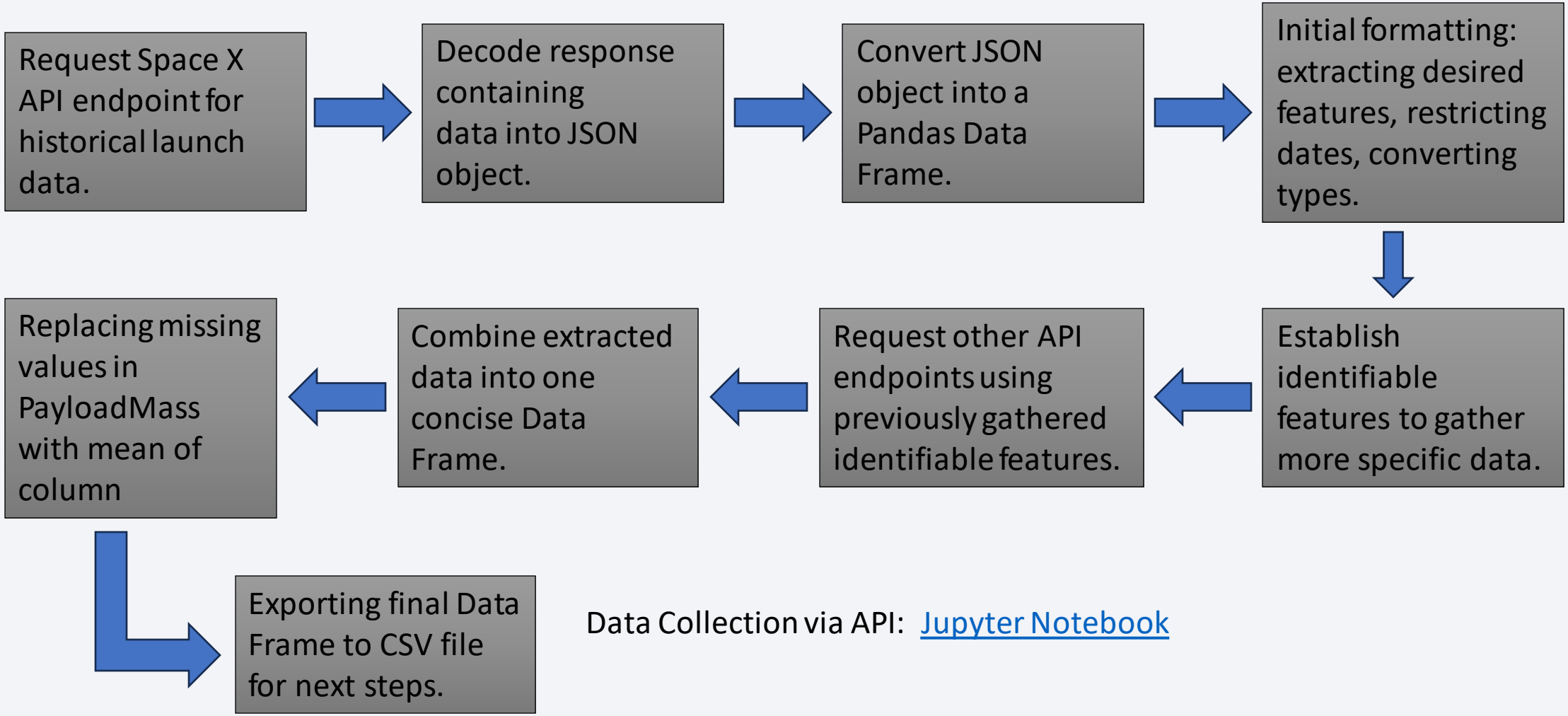
Executive Summary

- Data collection methodology:
  - Collected data through publicly available Space X REST API as well as through scraping historical data from relevant Wikipedia pages.
- Perform data wrangling
  - Process data to address missing values and generalizing outcomes into binary labels to represent successful or failed landing.
- Perform exploratory data analysis (EDA) using visualization and SQL
  - Explored relationships between features by querying and plotting data.
- Perform interactive visual analytics using Folium, Plotly, and Dash
  - Explored data via marked interactive maps and plots.
- Perform predictive analysis using classification models
  - Built, tuned, and evaluated a number of models to find the most accurate method of predicting landing outcomes.

Data Collection

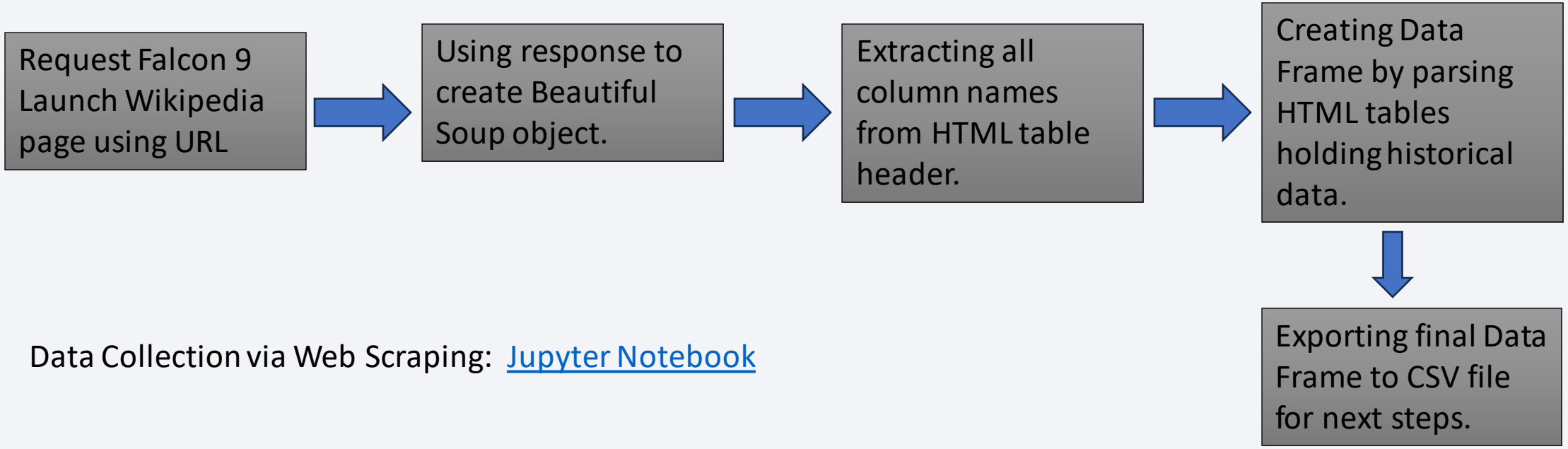
- Data collection via the Space X API involved requesting multiple endpoints as well as manipulation in order to get our data set into its most useful state.
  - This process was done using the following Python libraries: Requests, Datetime, Pandas, and NumPy.
- Data collection via web scraping involved requesting a Wikipedia page in order to extract it's HTML tables. These tables were then traversed and parsed in order to format them into a useful DataFrame.
  - This process was done using the following Python libraries: Requests, Unicode Data, Pandas, and BeautifulSoup.

Data Collection – SpaceX API



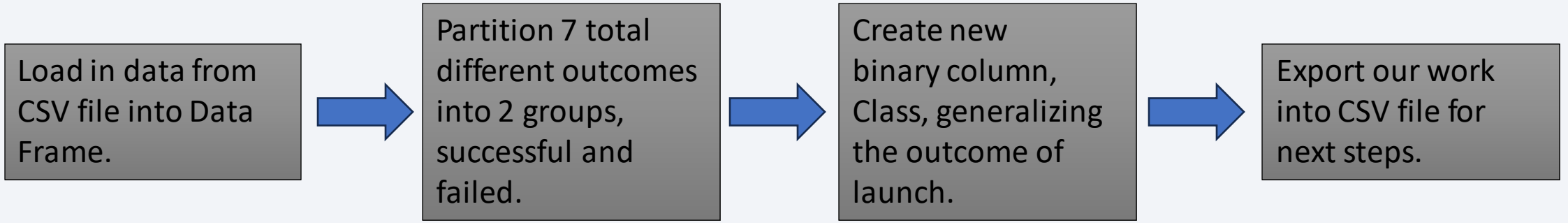
Data Collection via API: Jupyter Notebook

Data Collection – Scraping



Data Collection via Web Scraping: Jupyter Notebook

Data Wrangling



Data Wrangling : Jupyter Notebook

EDA with Data Visualization

- Summary of charts that were plotted:
  - Catplot to illustrate the relationship between Flight Number, Payload, and Outcome (Class).
  - Catplot to illustrate the relationship between Flight Number, Launch Site, and Outcome (Class).
  - Catplot to illustrate the relationship between Payload Mass, Launch Site, and Outcome (Class).
  - Bar Chart to illustrate the relationship between Orbit and average Outcome (Class).
  - Catplot to illustrate the relationship between Flight Number, Orbit, and Outcome (Class).
  - Catplot to illustrate the relationship between Payload Mass, Orbit, and Outcome (Class).
  - Line Chart to illustrate the relationship between Year and average Outcome (Class)

Exploratory Data Analysis with Data Visualization : Jupyter Notebook

EDA with SQL

Summary of SQL queries performed:

- Names of unique launch sites:
  - *SELECT DISTINCT "Launch_Site" FROM SPACEXTABLE*
- Records where launch sites begin with 'CCA':
  - *SELECT * FROM SPACEXTABLE WHERE "Launch_Site" LIKE 'CCA%' LIMIT 5*
- Total payload mass carried by boosters launched by NASA (CRS):
  - *SELECT SUM(PAYLOAD_MASS__KG_) AS TOTALPAYLOAD FROM SPACEXTABLE WHERE "Customer" like '%NASA%CRS%'*
- Average payload mass carried by booster F9 v1.1:
  - *SELECT AVG(PAYLOAD_MASS__KG_) AS AVG_PAYLOAD_MASS FROM SPACEXTABLE WHERE "Booster_Version" like '%F9 V1.1%'*
- Date of the first successful landing outcome in a ground pad:
  - *SELECT MIN("Date") AS FIRST_SUCCESFUL_LANDING FROM SPACEXTABLE WHERE "Landing_Outcome"="Success (ground pad)"*

Exploratory Data Analysis with SQL : Jupyter Notebook

EDA with SQL

Summary of SQL queries performed (cont.):

- Boosters with success in drone ship landing and have a payload mass greater than 4000 kg but less than 600 kg:
  - *SELECT "Booster_Version", PAYLOAD_MASS__KG FROM SPACEXTABLE WHERE "Landing_Outcome"="Success (drone ship)" AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000*
- Number of successful and failure mission outcomes:
  - *SELECT COUNT(CASE WHEN "Mission_Outcome"="Success" THEN "Mission_Outcome" END) AS SUCCESS, COUNT(CASE WHEN "Mission_Outcome"="Failure (in flight)" THEN "Mission_Outcome" END) AS FAILURE FROM SPACEXTABLE*
- Boosters with maximum payload mass:
  - *SELECT "Booster_Version" FROM SPACEXTABLE WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE)*
- Record for failed landing outcomes in drone ship during 2015:
  - *SELECT SUBSTR(Date,6,2) AS MONTH, SUBSTR(Date,0,5) AS YEAR, "Booster_Version", "Launch_Site", "Landing_Outcome" FROM SPACEXTABLE WHERE SUBSTR(Date,0,5)='2015' AND "Landing_Outcome"="Failure (drone ship)"*
- Count of landing outcomes between 2010-06-02 and 2017-03-20, ranked in descending order.
  - *SELECT "Landing_Outcome", COUNT(*) AS COUNT FROM SPACEXTABLE WHERE "Date" BETWEEN "2010-06-04" AND "2017-03-20" GROUP BY "Landing_Outcome" ORDER BY COUNT DESC*

Exploratory Data Analysis with SQL : Jupyter Notebook

Build an Interactive Map with Folium

Summary of map objects that were created and added to Folium map

- *Circle* and *Marker* objects to highlight a circular area with text label, marking each unique launch site.
- *MarkerCluster* to mark multiple launch attempts with the same coordinates.
- *MousePosition* to get coordinates for point on map on which the mouse is positioned.
- *PolyLine* to draw a line between launch site and it's proximities, such as closest coastline or city.

Interactive Location Analysis with Folium : Jupyter Notebook

Build a Dashboard with Plotly Dash

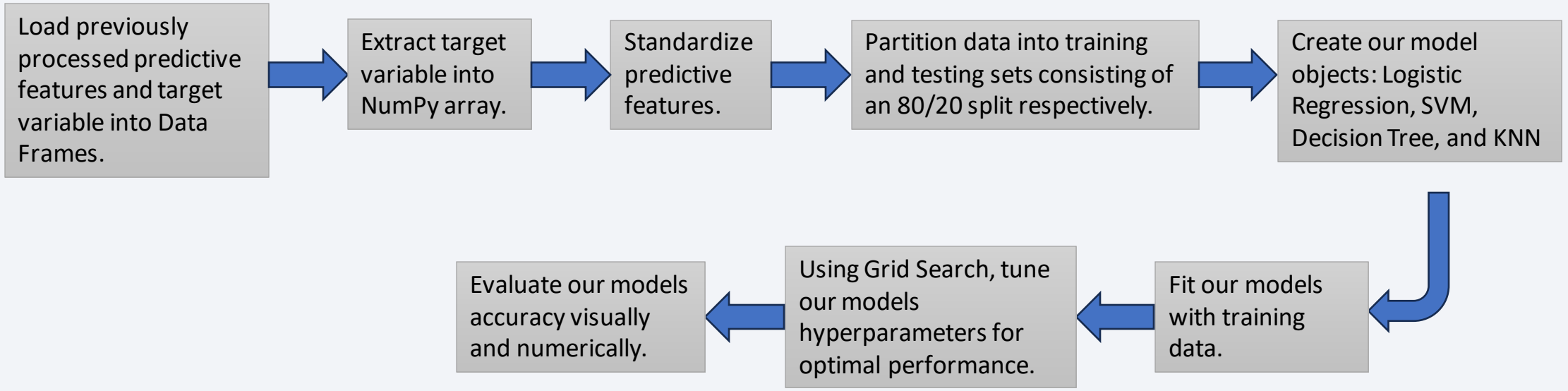
The dashboard was created with Dash and uses Plotly to visualize our data,

- Our dashboard was built to perform interactive visual analytics on Space X launch data in real-time, it does this through a pie chart and a scatter plot.
- The pie chart can be interacted with by using the drop down menu to filter which launch sites we want to be visualizing.
- The scatter plot can be interacted with by both the drop down menu and the range slider. The range slider allows us to further filter the data by Payload Mass.

Interactive Dashboard : GitHub

Predictive Analysis (Classification)

Model development process used to predict if the first stage of the Falcon 9 rocket.



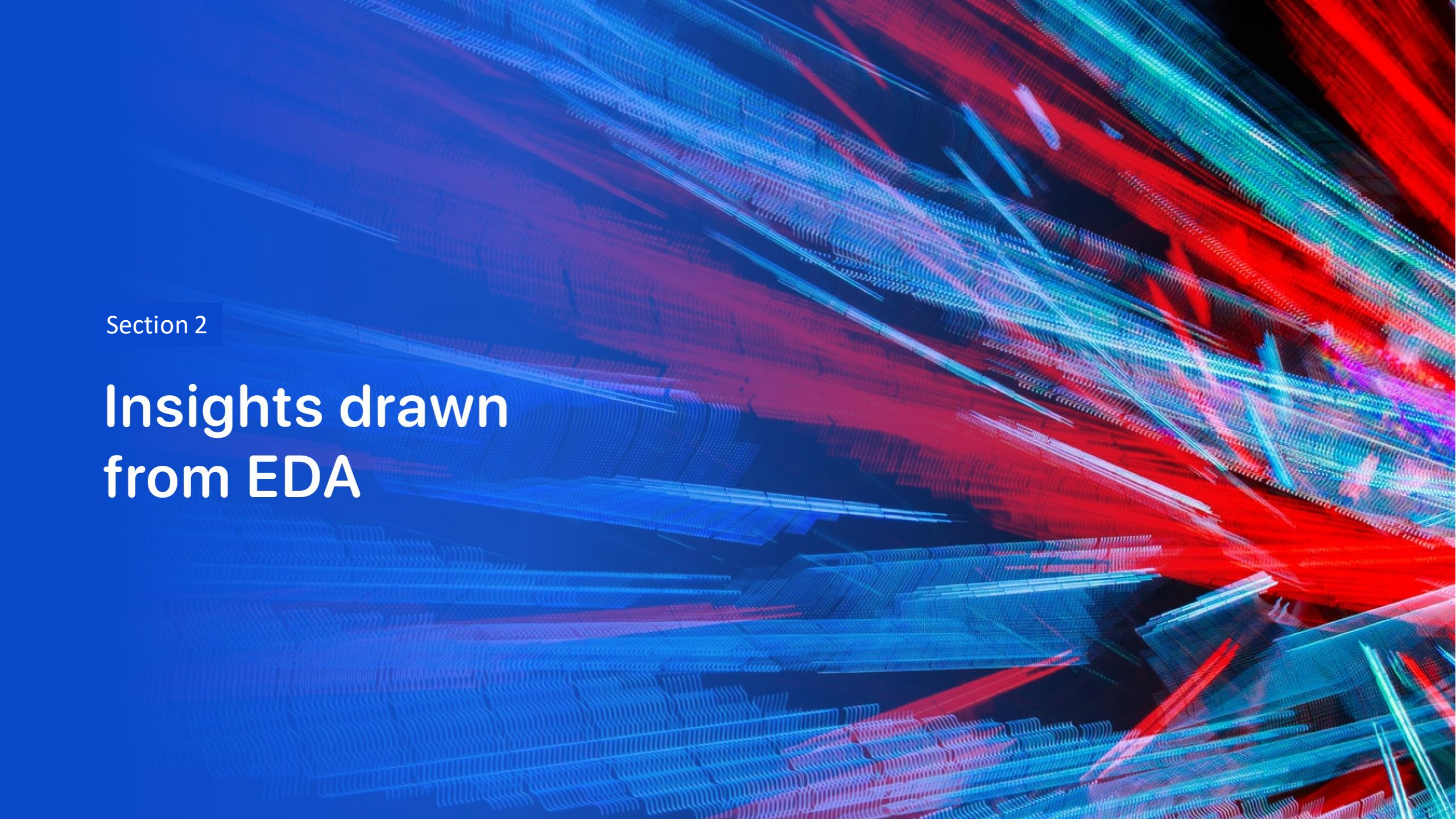
Predictive Analysis Development : Jupyter Notebook

Results

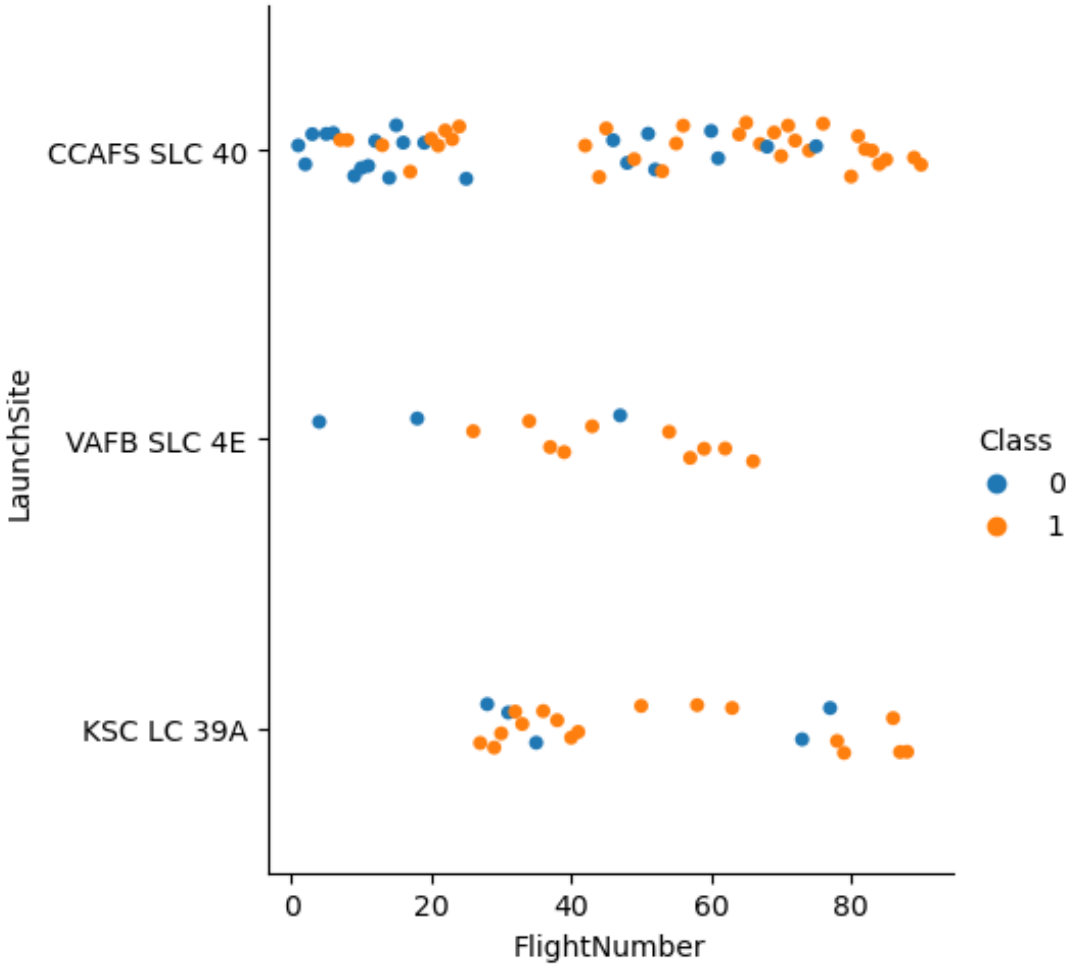
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA



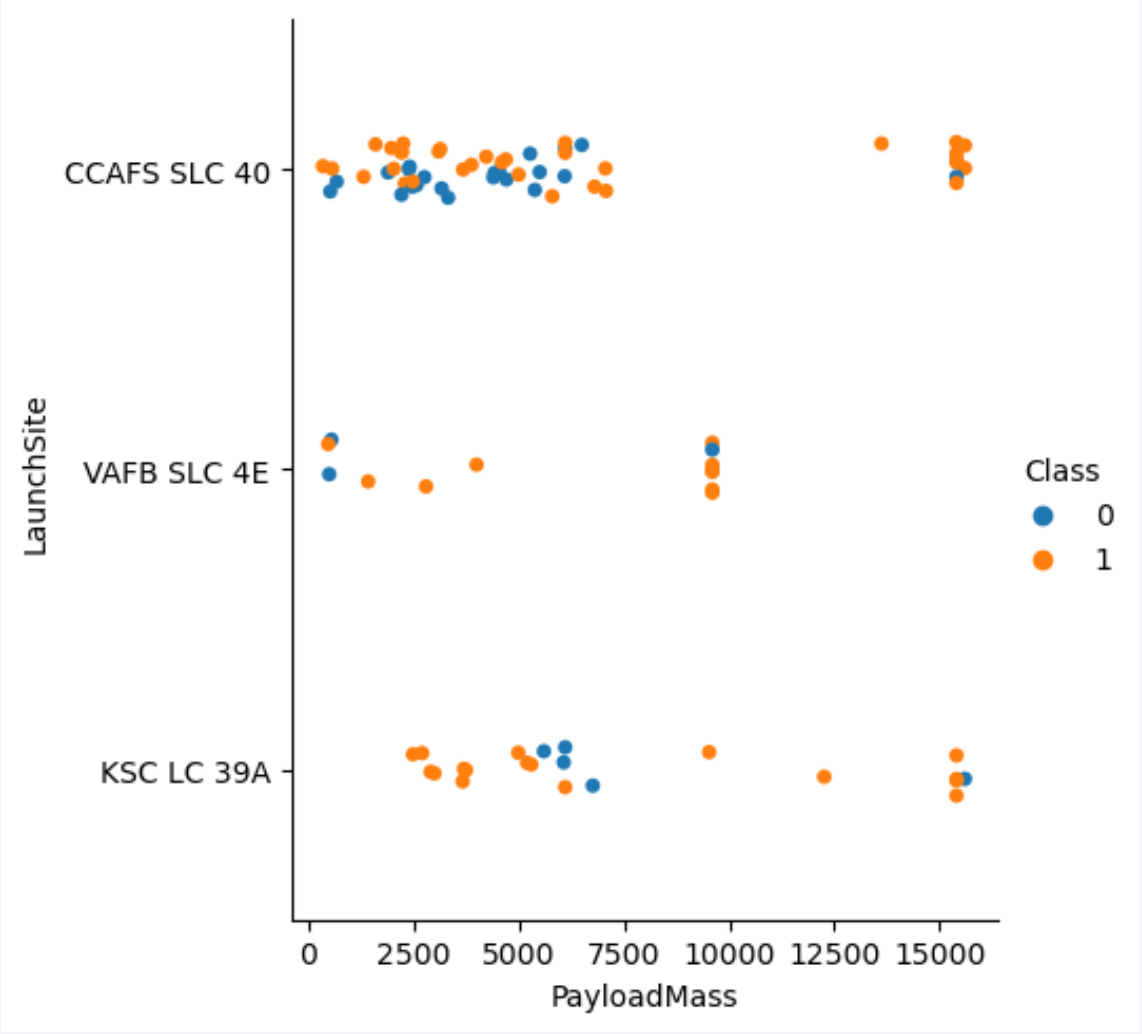
Flight Number vs. Launch Site



Insights:

1. As flight attempts continued, overall success rate increased.
2. Flights launched from VAFB SLC 4E had the highest amount of successful landing outcomes, however they only accounted for a third of total flight launch attempts.

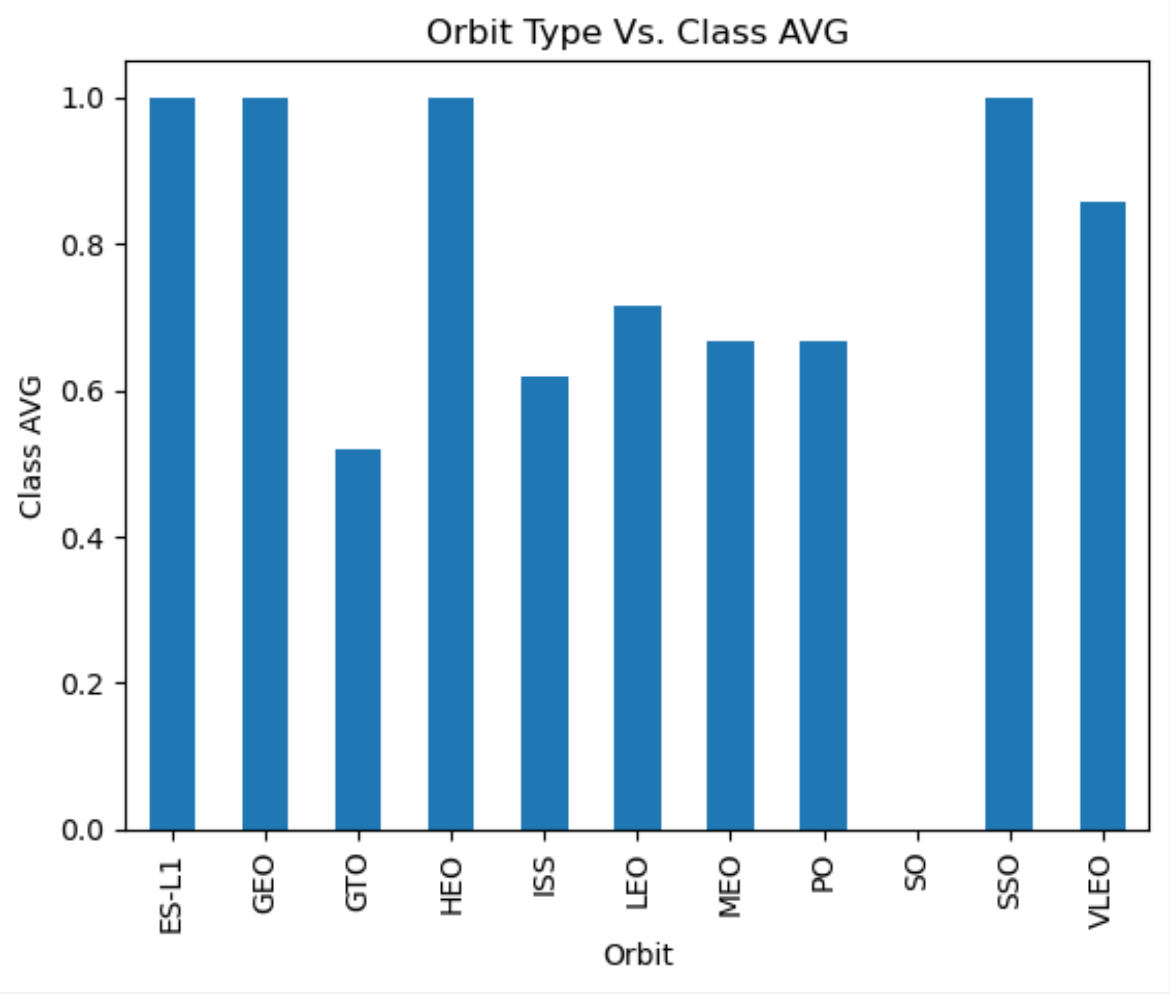
Payload vs. Launch Site



Insights:

1. As Payload Mass increases, we are more likely to have a successful landing outcome.
2. There are no flights with payloads higher than 10000 kg for VAFB SLC 4E

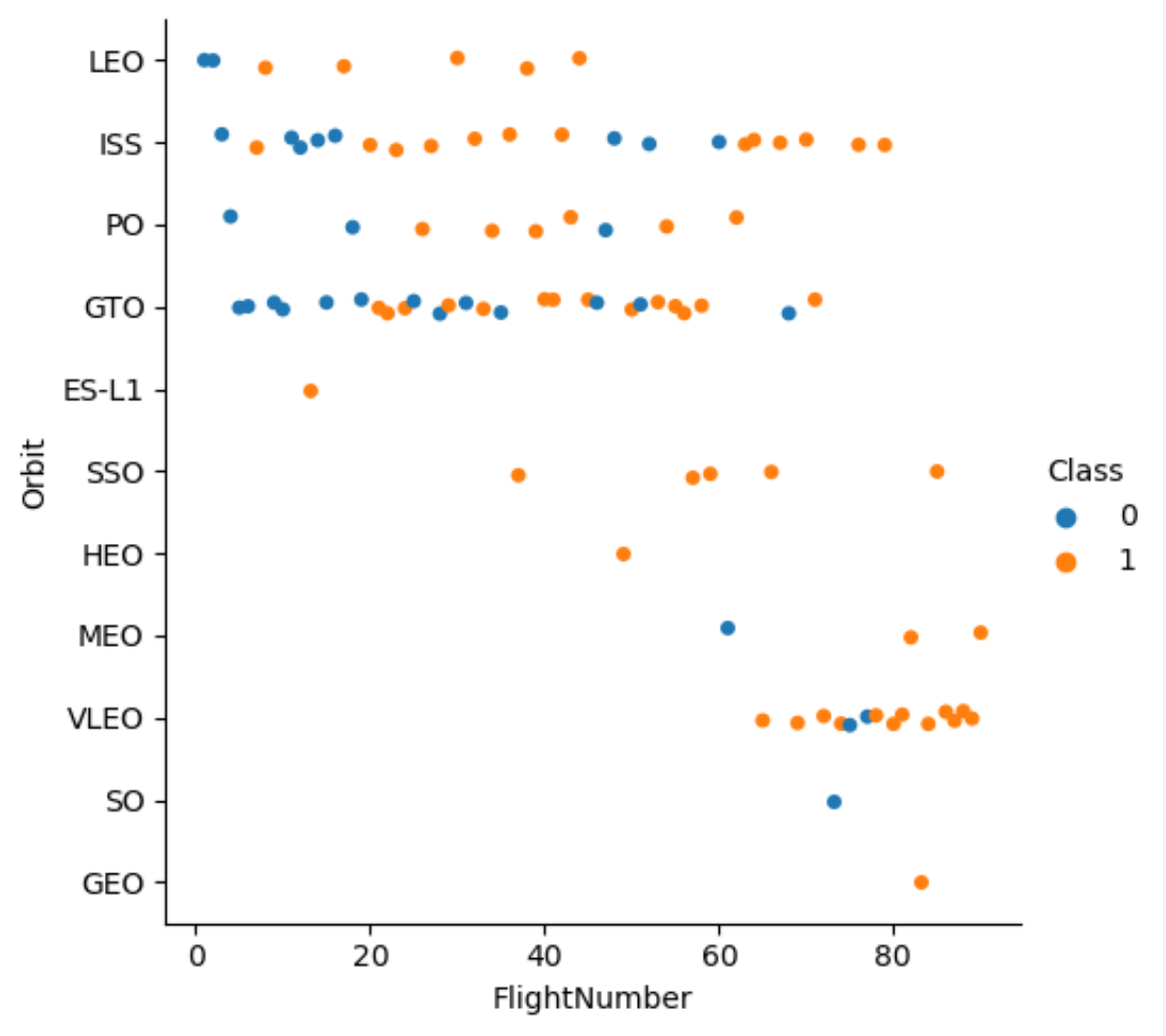
Success Rate vs. Orbit Type



Insights:

1. Flights launched at ES-L1, GEO, HEO, and SSO have the best highest success rate.

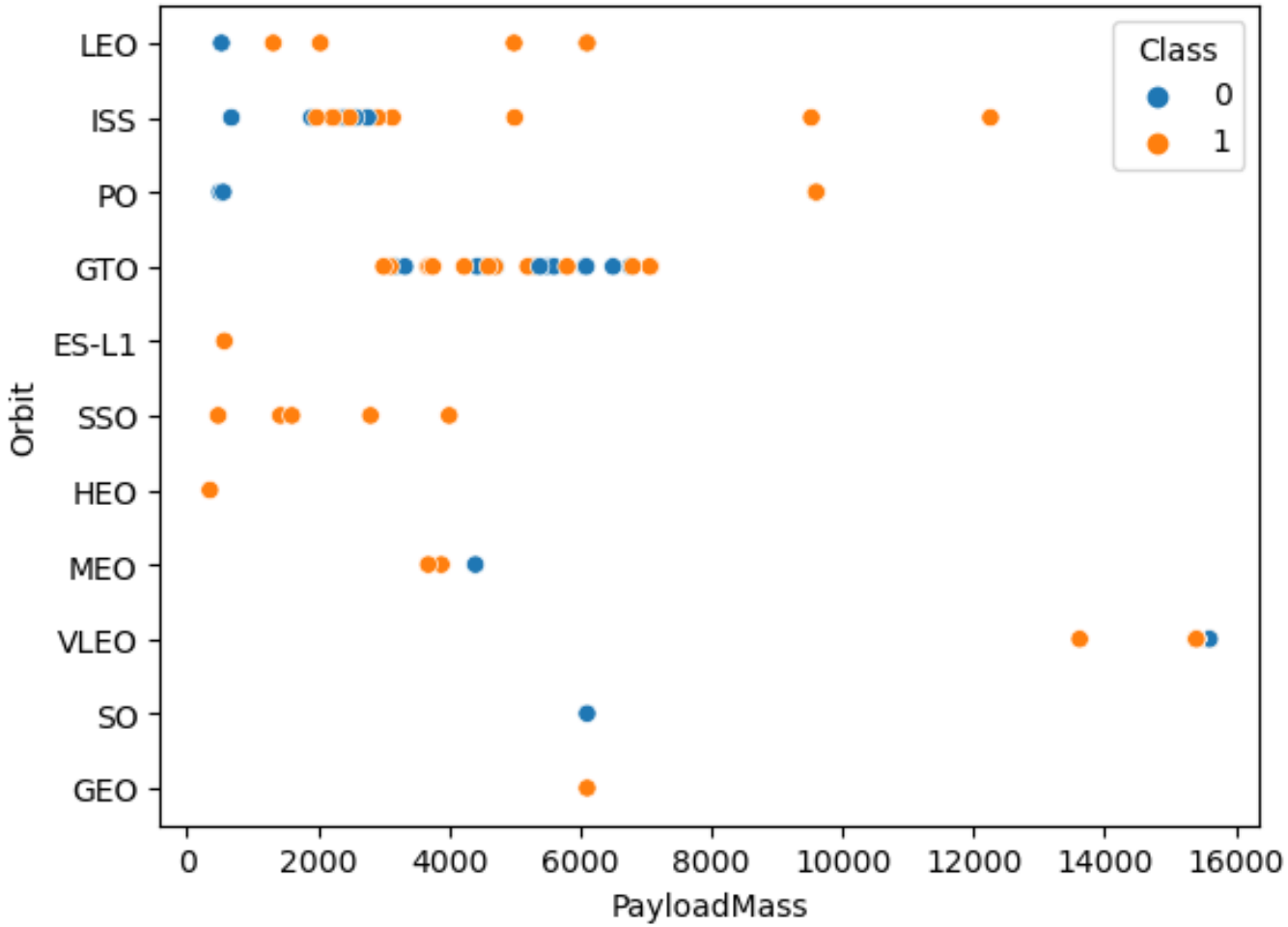
Flight Number vs. Orbit Type



Insights:

1. As flight attempts progressed, we became less likely to see a failed class assignment, specially notable after flight number 80.

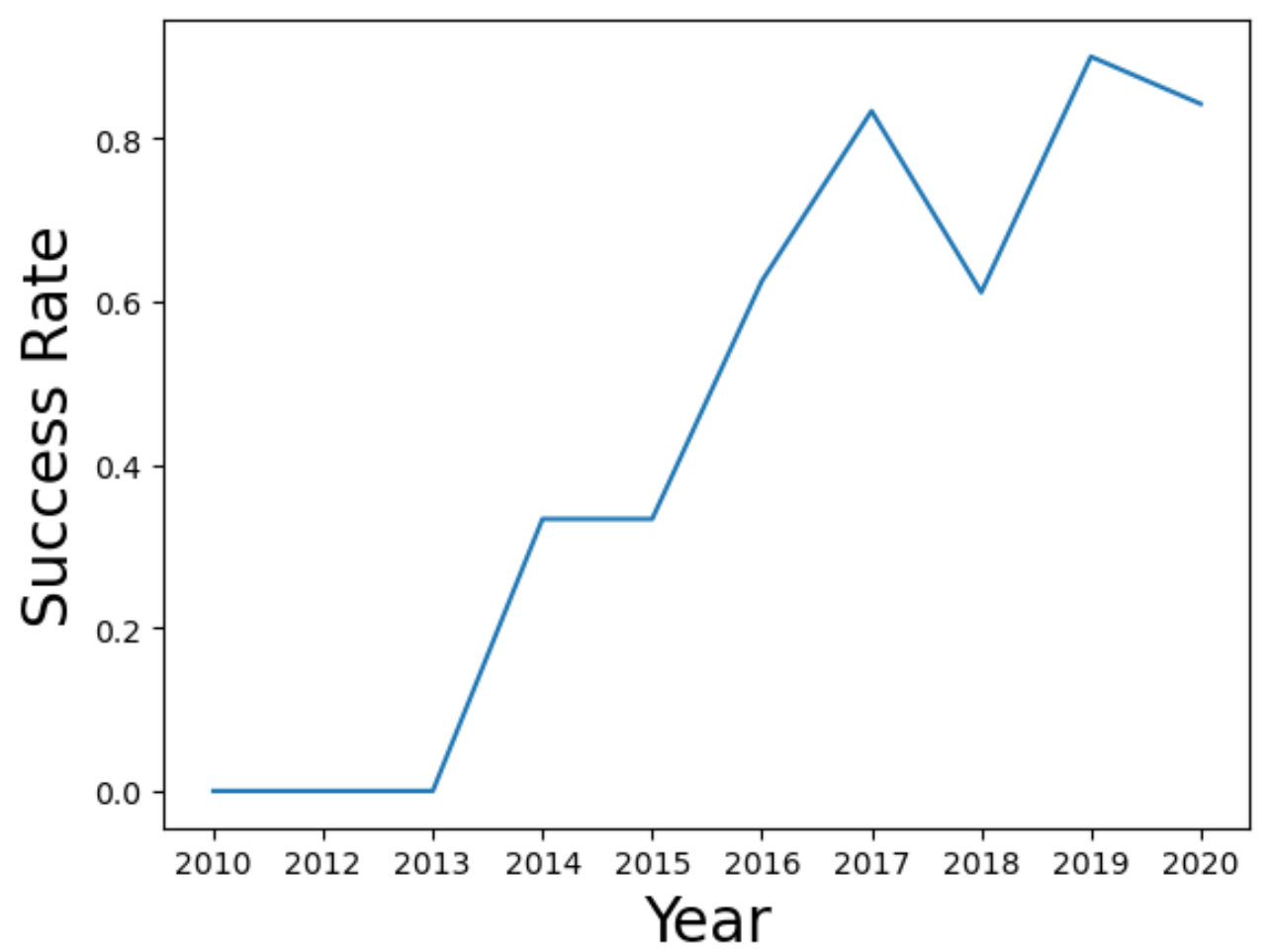
Payload vs. Orbit Type



Insights:

1. LEO, ISS, and PO are more likely to have a successful landing outcome with a heavy payload.

Launch Success Yearly Trend



Insights:

1. We can observe that success rate maintained an uptrend after 2013 until 2017 where we took a dip until 2018, but then recontinued up to 2020

All Launch Site Names

```
%sql select distinct "Launch_Site" from SPACEXTABLE
 * sqlite:///my_data.db
Done.
```

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

1. Using the DISTINCT statement we can see that we have 4 launch sites included in our dataset.

Launch Site Names Begin with 'CCA'

```
%sql select * from SPACEXTABLE where "Launch_Site" like 'CCA%' limit 5
```

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

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

1. Using the query above, we displayed 5 record where the launch site begins with "CCA".

Total Payload Mass

```
%sql select sum(PAYLOAD_MASS__KG_) as TOTALPAYLOAD from SPACEXTABLE where "Customer" like '%NASA%CRS%'
 * sqlite:///my_data.db
Done.
```

TOTALPAYLOAD
48213

1. The total payload mass carried by boosters launched by NASA (CRS) using the SUM function and WHERE clause.

Average Payload Mass by F9 v1.1

```
%sql select avg(PAYLOAD_MASS__KG_) as AVG_PAYLOAD_MASS from SPACEXTABLE where "Booster_Version" like '%F9 v1.1%'
```

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

AVG_PAYLOAD_MASS
2534.6666666666665

1. The average payload mass carried by booster version F9 V1.1 using the AVG function and WHERE clause.

First Successful Ground Landing Date

```
%%sql
select min("Date") as FIRST_SUCCESSFUL_LANDING
from SPACEXTABLE where "Landing_Outcome"="Success (ground pad)"
```

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

FIRST_SUCCESSFUL_LANDING
2015-12-22

1. The date when the first successful landing outcome in ground pad was achieved using the MIN function.

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%%sql
select "Booster_Version", PAYLOAD_MASS__KG_ from SPACEXTABLE
where "Landing_Outcome"="Success (drone ship)" and PAYLOAD_MASS__KG_ between 4000 and 6000
```

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

Booster_Version	PAYLOAD_MASS__KG_
F9 FT B1022	4696
F9 FT B1026	4600
F9 FT B1021.2	5300
F9 FT B1031.2	5200

1. The names of the boosters which have had success landing in drone ship and have a payload mass greater than 4000 but less than 6000, combining the WHERE clause and the AND operator.

Total Number of Successful and Failure Mission Outcomes

```
%%sql
select
count(case when "Mission_Outcome"="Success" then "Mission_Outcome" END) as SUCCESS,
count(case when "Mission_Outcome"="Failure (in flight)" then "Mission_Outcome" END) as FAILURE
from SPACEXTABLE
```

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

SUCCESS	FAILURE
98	1

1. The total number of successful and failed mission outcomes. This query was performed using the COUNT function and a GROUP BY clause.

Boosters Carried Maximum Payload

```
%%sql
select "Booster_Version" from SPACEXTABLE
where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEXTABLE)
```

```
 * sqlite:///my_data.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

1. The names of booster version which have carried the maximum payload mass, performed using a subquery.

2015 Launch Records

```
%%sql
select
substr(Date,6,2) as MONTH,
substr(Date,0,5) as YEAR,
"Booster_Version",
"Launch_Site",
"Landing_Outcome"
from SPACEXTABLE
where
substr(Date,0,5)='2015'
and
"Landing_Outcome"="Failure (drone ship)"
```

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

MONTH	YEAR	Booster_Version	Launch_Site	Landing_Outcome
10	2015	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
04	2015	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

1. Records of failed outcomes in drone ship for the year 2015.

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

```
%%sql
select
"Landing_Outcome",
count(*) as COUNT
from SPACEXTABLE
where "Date" between "2010-06-04" and "2017-03-20"
group by "Landing_Outcome"
order by COUNT desc
```

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

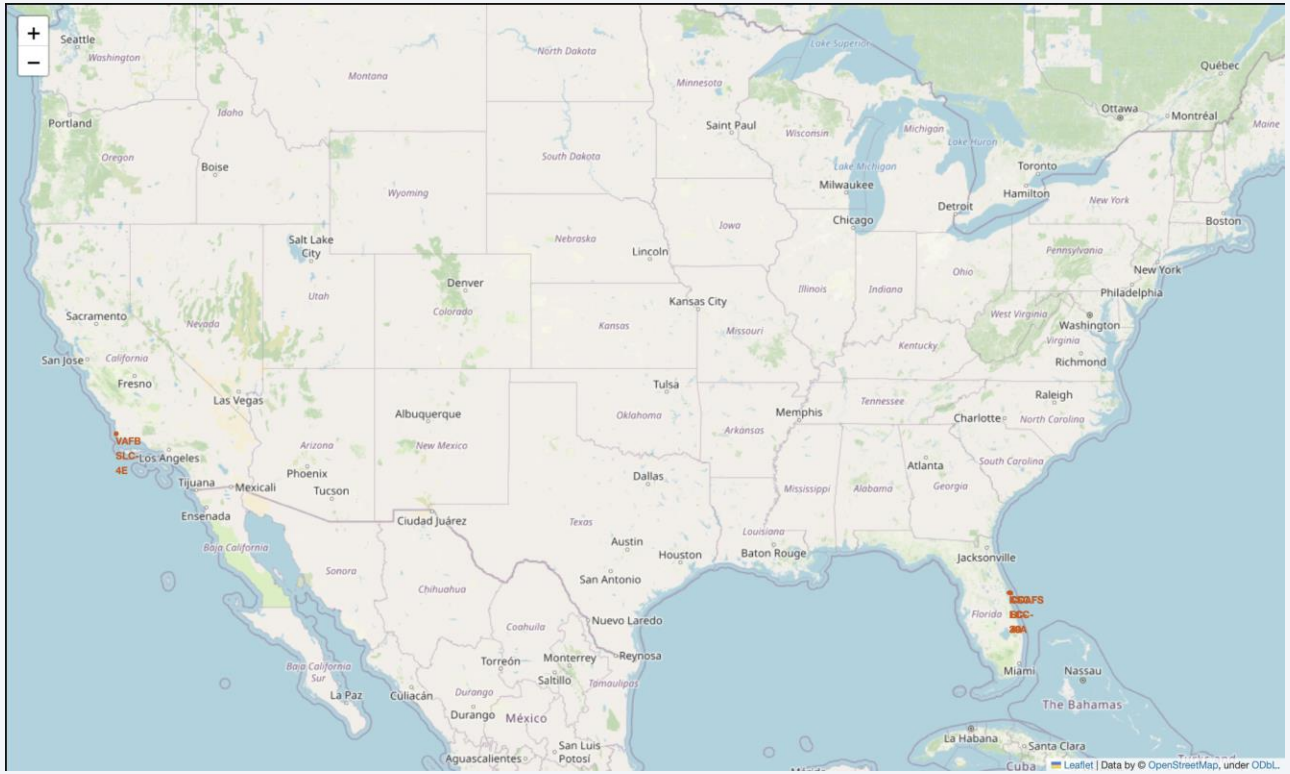
Landing_Outcome	COUNT
No attempt	10
Success (ground pad)	5
Success (drone ship)	5
Failure (drone ship)	5
Controlled (ocean)	3
Uncontrolled (ocean)	2
Precluded (drone ship)	1
Failure (parachute)	1

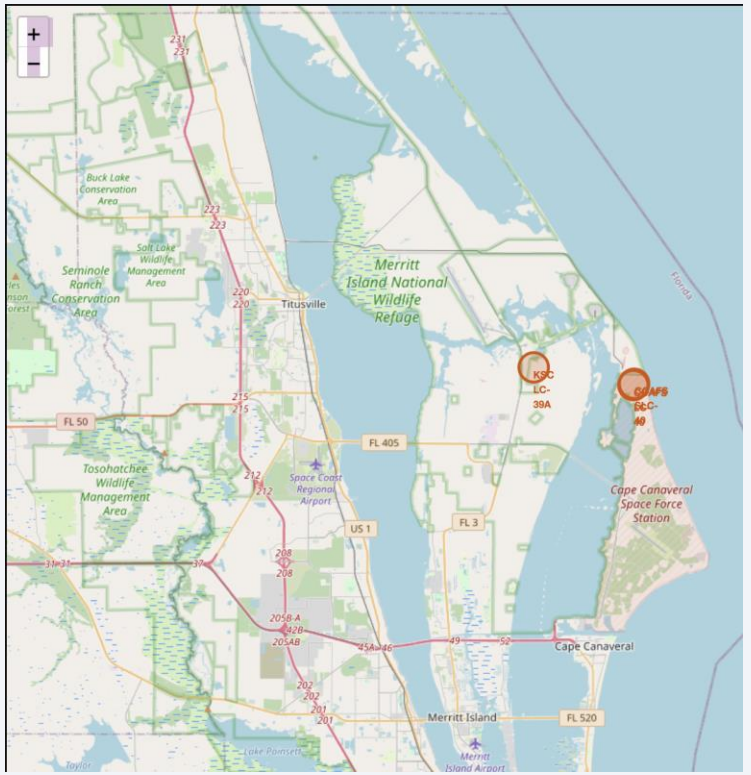
1. The count of landing outcomes between date range, ranked in descending order.


Section 3
Launch Sites
Proximities Analysis

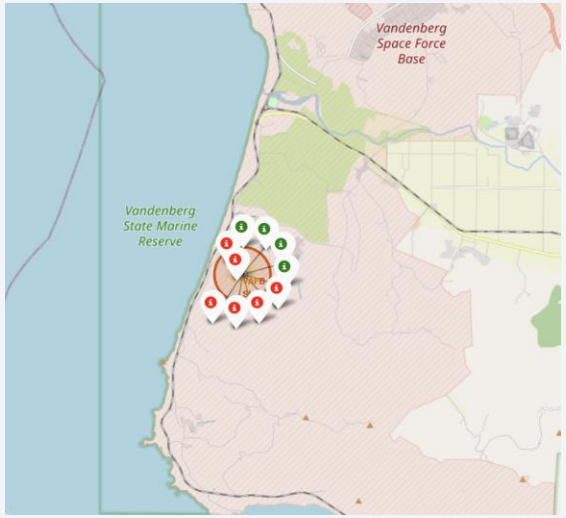
All Launch Sites

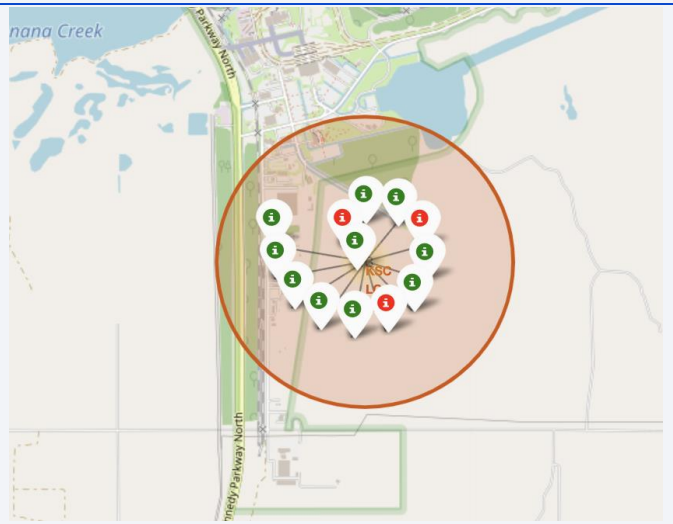
- All launch sites are in close proximity to their corresponding coast.

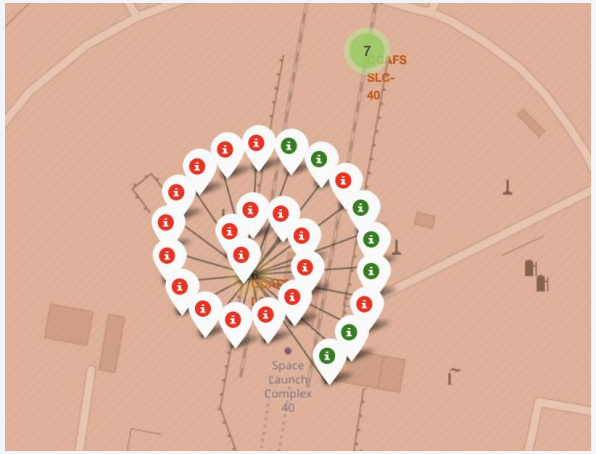


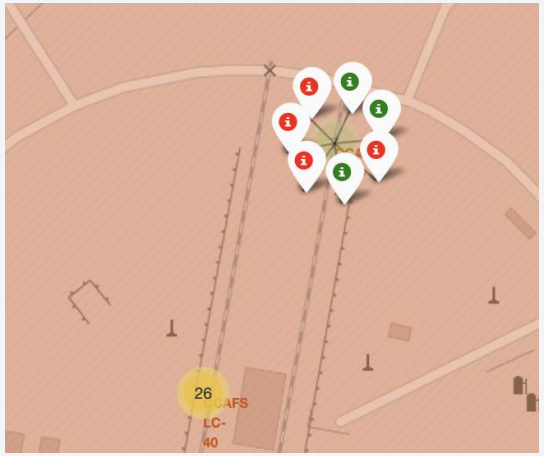


Successful and Failed Launches



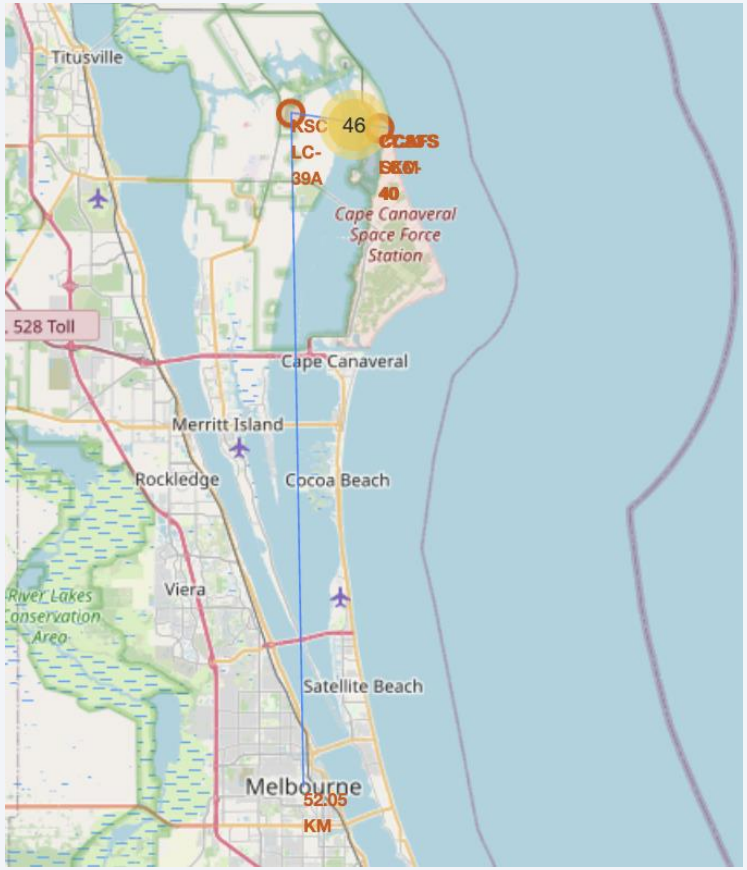


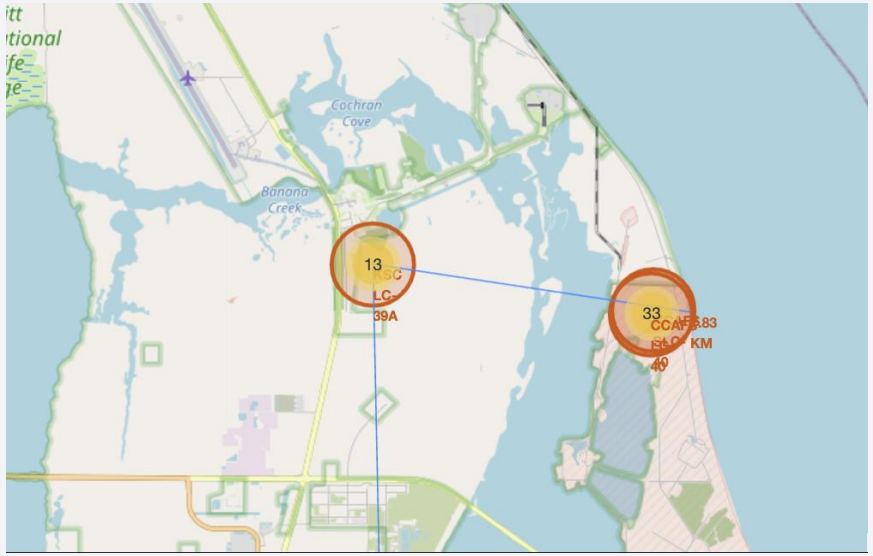




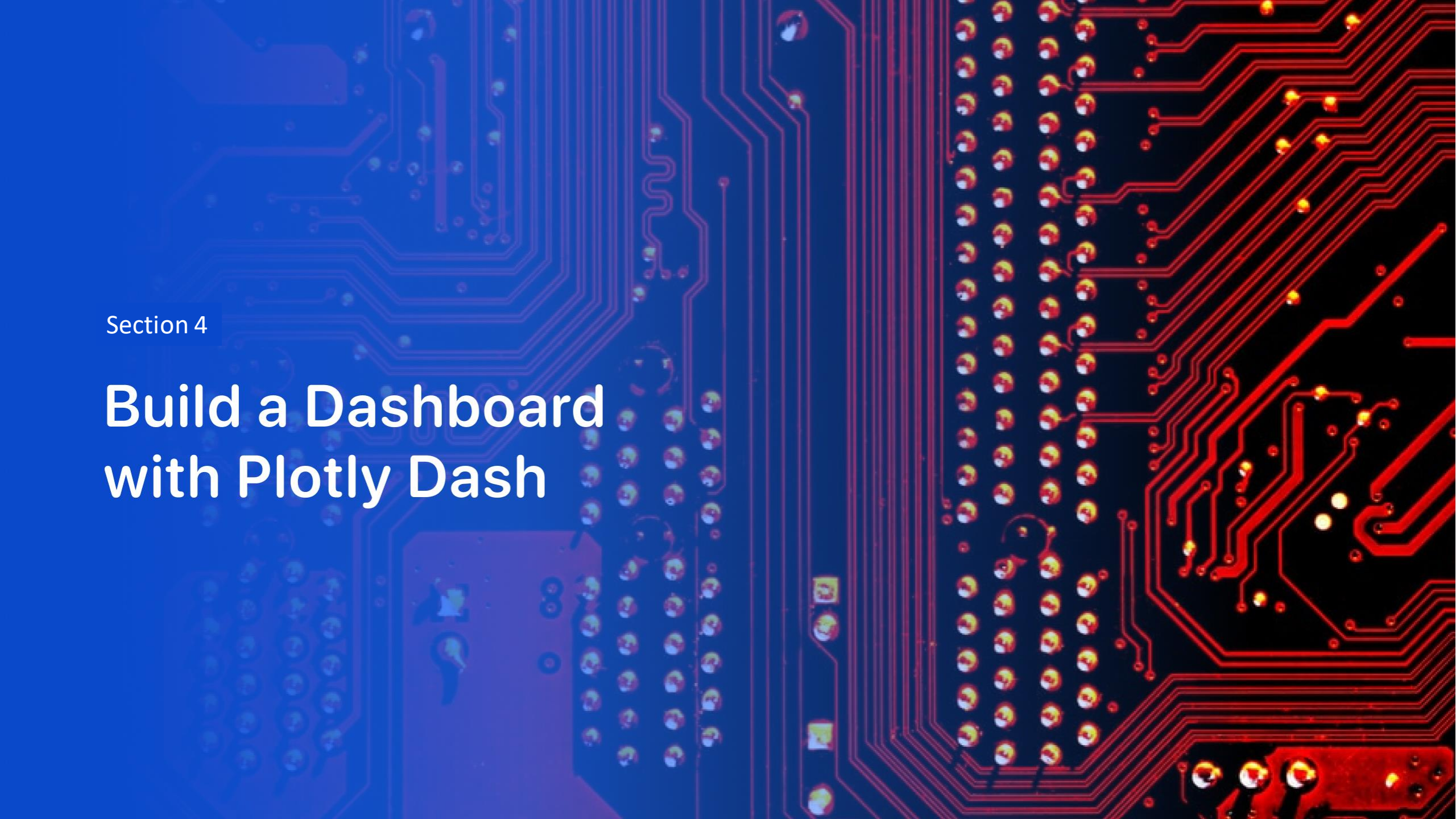
1. Markers on these screenshots show the outcome of launches at each site, green if successful or red if failure.

Launch Site and Proximities

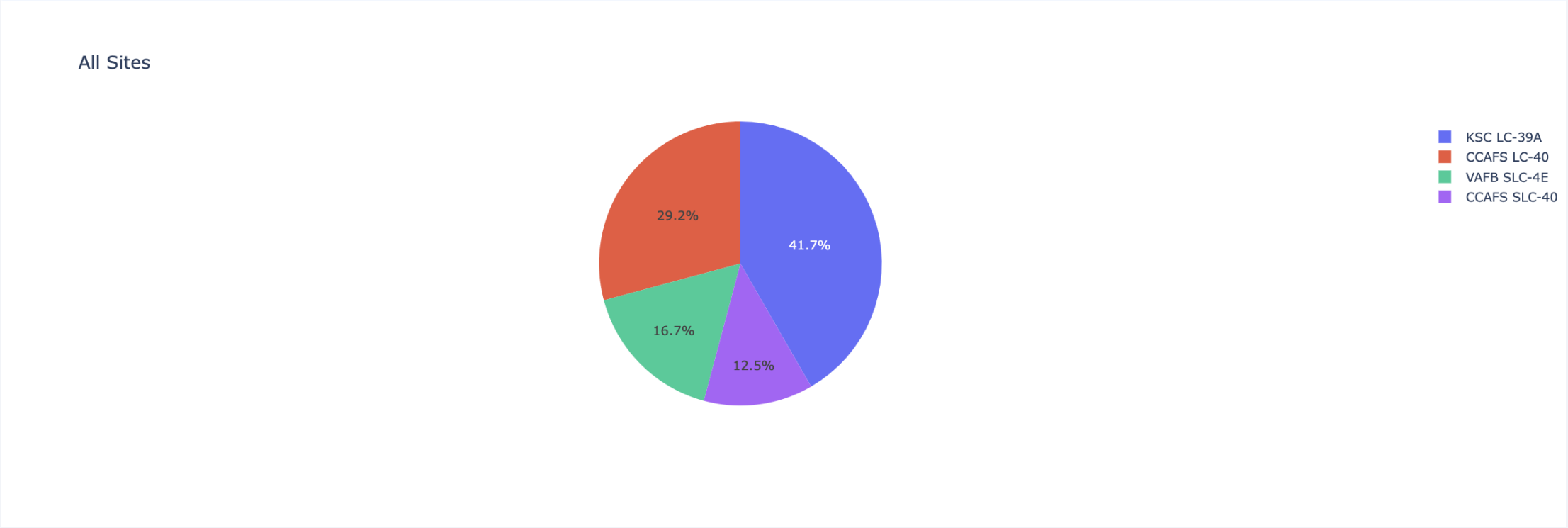




1. Launch sites are always coastlines, railways, and highways.
2. Launch sites also always maintain a safe distance from major cities.

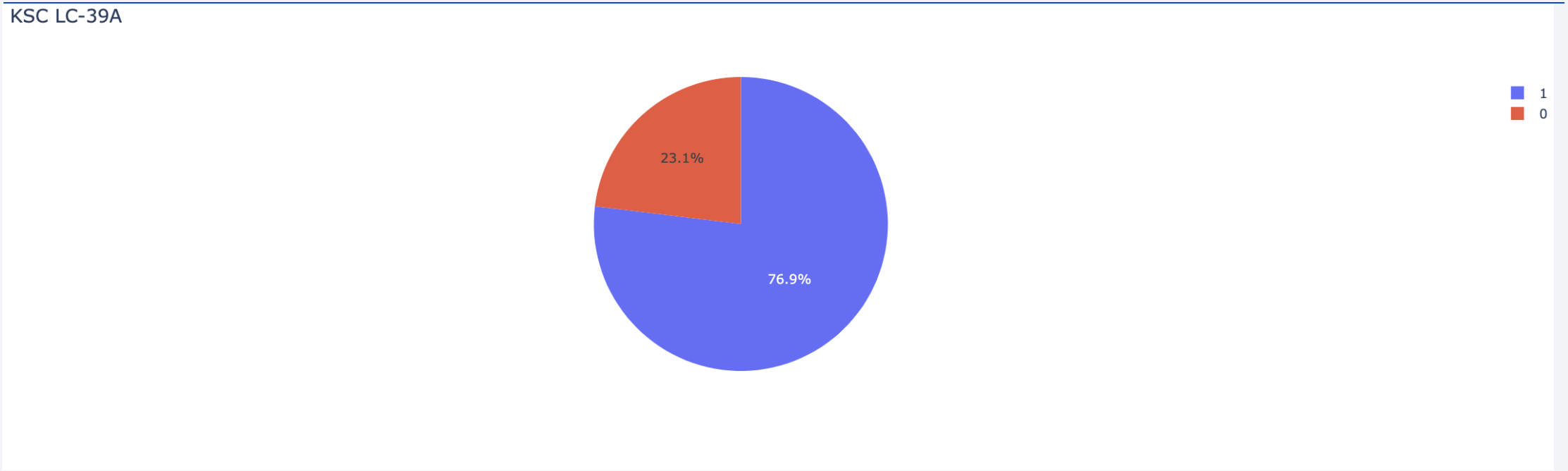

Section 4
Build a Dashboard with Plotly Dash

Successful Outcomes at All Sites



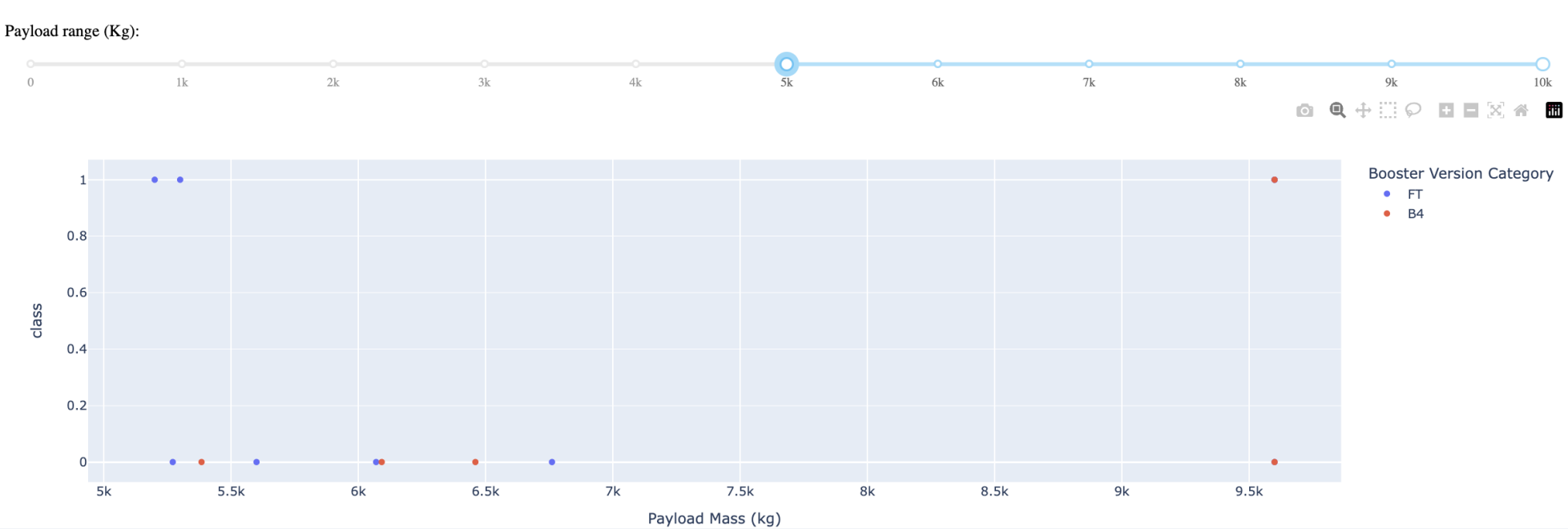
1. Distribution of successful outcomes throughout all launch sites.
2. Notably, KSC LC-39 has the highest percentage of successful outcomes out of all launch sites.

KSC LC-39A



1. KSC LC-39A holds the highest piece of the pie when it comes to successful outcomes.
2. This outcome is due to a 76.9% success rate.

All Sites: Payload vs Class

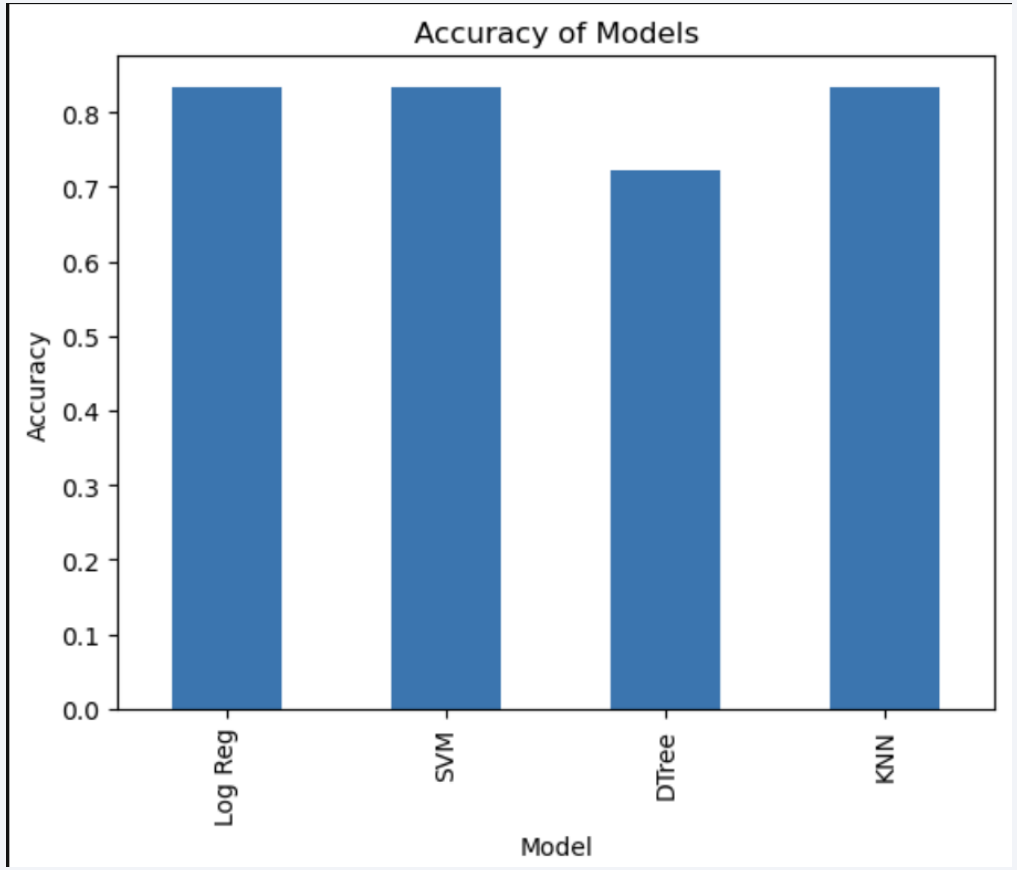


1. Considering all sites, launches with a payload between 5000 kg and 10000 kg have had only three successful outcomes.
    1. Successful outcomes being represented with a 1, failed outcomes with a 0.

Section 5

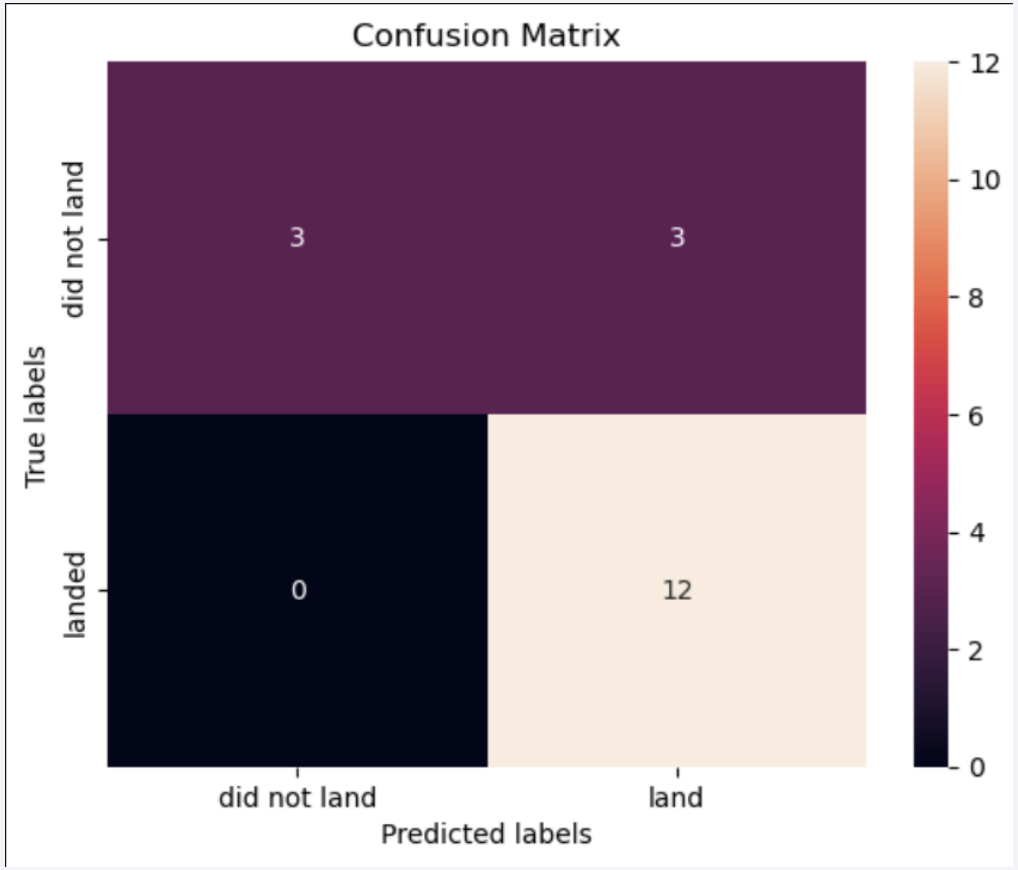
Predictive Analysis (Classification)

Classification Accuracy



1. All of our models have the same or comparable accuracy.

KNN: Confusion Matrix



1. As visible in our confusion matrix, our KNN model is more prone to type 1 errors than type 2.
2. Type one being false positives, and type two being false negatives.

Conclusions

- All of our models using different methods of machine learning were approximately of same accuracy.
- We are able to use data to predict if the first stage of the Falcon 9 rocket will land, to a statistically significant accuracy of 83%.

Appendix

- For notebooks, scripts, and further documentation attached below is the repository link: GitHub project repository.

Thank you!