IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

François Jansen
August 11, 2024

Outline

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

Executive Summary

Summary of methodologies

- Data Collection
- Data Wrangling
- Exploratory Data Analysis through Visualization
- Exploratory Data Analysis using SQL
- Interactive Mapping using Folium
- Dashboarding using Plotly Dash
- Predictive Analysis using Classification Methods

Summary of Results Provided

- Exploratory Data Analysis results
- Interactive Analytics using screenshots
- Predictive Analysis results

Introduction

Project background and context

SpaceX 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 SpaceX can reuse the first stage. Therefore if we can determine if the first stage will land, we can determine the cost of a launch. This information will be used to determine if we want to bid against SpaceX for a rocket launch.

Questions to answer

- What is the impact of payload mass, launch site, flight count and orbit on success of first stage landing?
- Does the frequency of landing success improve over time?
- What is the best methodology for predicting launch and first stage landing success?

Section 1

Methodology

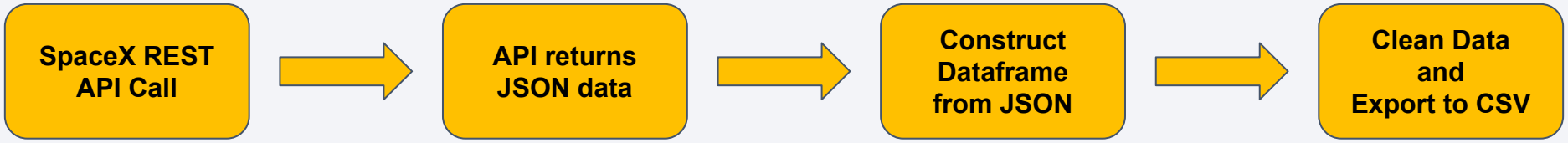
Methodology

Executive Summary

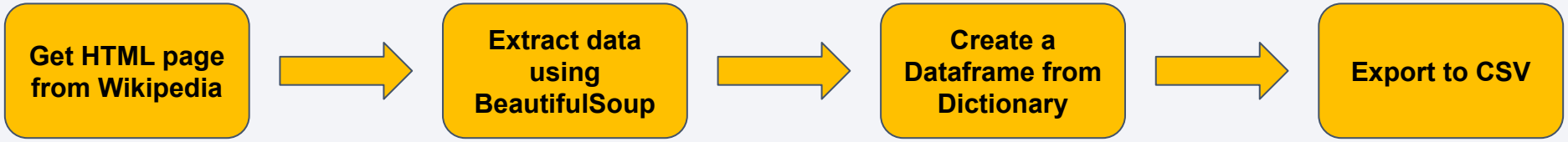
- Data collection methodology:
  - SpaceX data was collected using their public REST API
  - Additional launch information was Scraped from Wikipedia
- Perform data wrangling
  - Dropping unnecessary data columns to focus on relevant data points
  - One-Hot Encoding of classification odels
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Comparative analysis of various classification models to determine accuracy

Data Collection

- Datasets were collected using the SpaceX REST API and scraping Wikipedia pages
  - Information collected from SpaceX provided rocket, launch and payload information.
    (url: https://api.spacexdata.com/v4/)



  - Information collected from wikipedia through scraping provided launch, landing and payload information.
    (url: https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922)



Data Collection – SpaceX API

1. Call SpaceX API

```
spacex_url = "https://api.spacexdata.com/v4/launches/past"
response = requests.get(spacex_url)
```

2. Convert response to JSON file

```
response = requests.get(static_json_url).json()
data = pd.json_normalize(response)
```

3. Transform data

```
getBoosterVersion(data)
getLaunchSite(data)
getPayloadData(data)
getCoreData(data)
```

4. Create dictionary from data

```
launch_dict = {'FlightNumber': list(data['flight_number']),
               'Date': list(data['date']),
               'BoosterVersion':BoosterVersion,
                        ○
                        ○
                        ○
               'Longitude': Longitude,
               'Latitude': Latitude}
```

5. Create dataframe

```
df = pd.DataFrame.from_dict(launch_dict)
```

6. Filter dataframe

```
data_falcon9 = df[df['BoosterVersion']=='Falcon 9']
```

7. Export to file

```
data_falcon9.to_csv('../data/dataset_part_1.csv', index=False)
```


SpaceX API Data Collection notebook link

Data Collection – Scraping

1. Get HTML page and create BeautifulSoup object

```
response = requests.get(static url)
soup = BeautifulSoup(response.content, 'html.parser')
```

2. Find all tables in the page

```
html_tables = soup.find_all('table')
```

3. Get column names

```
column names = []
for header in first_launch_table.find_all('th'):
    column_name = extract_column_from_header(header)
    if column_name is not None and len(column_name) > 0:
        column_names.append(column_name)
```

4. Create dictionary

```
launch_dict= dict.fromkeys(column_names)
```

4. Populate dictionary with data

```
for table_number,table in enumerate(soup.find_all('table',"wikitable
plainrowheaders collapsible")):
    # get table row
    for rows in table.find_all("tr"):
```

   -
   -

5. Create dataframe from dictionary

```
df = pd.DataFrame.from_dict(launch_dict)
```

6. Export to file

```
df.to_csv('../data/dataset_part_2.csv', index=False)
```

[Wikipedia Web Scraping notebook link]

Data Wrangling

- Transform string variable describing landing outcome to categorical variable (1=Success, 0=Failure)
  - True Ocean, True RTLS, True ASDS describe successful missions
  - Falce Ocean, Falce RTLS, False ASDS describe unsuccessful outcomes

1. Calculate launches for each site

```
# Apply value_counts() on column LaunchSite
df['LaunchSite'].value_counts()
```

```
CCAFS SLC 40     55
KSC LC 39A       22
VAFB SLC 4E      13
Name: LaunchSite, dtype: int64
```

2. Calculate number of each orbit occurrences

```
# Apply value_counts on Orbit column
print(df['Orbit'].value_counts())
```

```
GTO      27
ISS      21
VLEO     14
PO        9
LEO       7
SSO       5
MEO       3
ES-L1     1
HEO       1
SO        1
GEO       1
Name: Orbit, dtype: int64
```

3. Calculate number of outcomes per orbit type

```
landing_outcomes = df['Outcome'].value_counts()
landing_outcomes
```

```
True ASDS      41
None None      19
True RTLS      14
False ASDS      6
True Ocean      5
None ASDS       2
False Ocean     2
False RTLS      1
Name: Outcome, dtype: int64
```

4. Create landing outcome label from Outcome column

```
# landing_class = 0 if bad_outcome
# landing_class = 1 otherwise
landing_class = []
for outcome in df['Outcome']:
    if outcome in bad_outcomes:
        landing_class.append(0)
    else:
        landing_class.append(1)
```

5. Export to file

```
df.to_csv("dataset_part_2.csv", index=False)
```

[Data Wrangling notebook link]

EDA with Data Visualization

- Scatter Graphs
  - Flight Number vs. Payload Mass
  - Flight Number vs. Launch Site
  - Payload vs. Launch Site
  - Flight Number vs. Orbit Type
  - Payload Mass vs. Orbit Type

These graphs show the correlation between the above sets of variables

- Bar Graph
  - Success rate vs. Orbit

Bar graphs show relationship between numeric and categorical variables

- Line Graph
  - Success vs Year of launch

Line graphs display the trend in data variables. Good for tracking the direction that launches are going in terms of success in the above graph.

EDA Data Visualization notebook link

EDA with SQL

- A set of SQL queries were performed to gain an understanding of the data, including:
  - Display names of unique launch sites
  - Display total payload mass carried by boosters launched by NASA (CRS_
  - Display average payload mass carried by booster version F9 v1.1
  - List date of first successful landing outcome on ground pad
  - List the names of boosters with success on drone ship and payload mass between 4000 and 6000 kg
  - List the total number of successful and failing mission outcomes
  - List the names of booster version which have carried maximum payload mass
  - List record of failed drone ship landing outcomes, booster versions and launch site by month for 2015
  - Rank the count of successful landing outcomes between 2010-06-04 and 2017-03-20 in descending order

EDA with SQL notebook link

Build an Interactive Map with Folium

- Used Folium map of USA and performed the following exercises for analysis:
  - Market NASA Johnson Space Center's location and labeled it
  - Marked each of the launch sites used by SpaceX and labeled their names
  - Created a MarkerCluster to display detailed information grouped at a coordinate
  - Used the MarkerCluster to mark Successful and Unsuccessful landings
  - Used cursor location to get coordinates of key nearby locations (railway, coastline, highway, city), calculate the distance to these locations and plotted a line to each of them labeled with the distance from the launch site
- Analysis showed launch sites tend to be in proximity of coastline, railway and highway, and a good distance from cities.

Launch Sites Analysis notebook link

Build a Dashboard with Plotly Dash

Create a dashboard with a drop-down, pie chart, range-slider and scatter-plot:

- Drop-down gives a choice of selecting an individual launch site or all sites. (*dash_core_components.Dropdown*)
- Pie chart indicates the number of successes and failures for the specific launch site (if all sites are chosen, the chart shows the percentages of successful launches for each site) (*plotly.express.pie*)
- Range-slider selects the payload mass between 0 and 10000 kg, with 1000 kg increments. (*dash_core_components.RangeSlider*)
- Scatter chart shows the correlation between launch successes and payload mass (*plotly.express.scatter*)

Launch Sites Analysis Dashboard code

Predictive Analysis (Classification)

- In order the following steps were performed:
  - Data Preparation
    - Dataset load
    - Data normalization
    - Splitting training and test data (80/20)
  - Model Preparation
    - Run through a selection of algorithms:
      - LogisticRegression
      - Support Vector Machine
      - DecisionTree
      - K-nearest Neighbor
    - Tune parameters using GridSearchCV
    - Train GridSearchModel
  - Model Evaluation
    - Get tuned hyperparameters
    - Compute accuracy
    - Plot Confusion Matrix
  - Model Comparison
    - Evaluate models based on accuracy
    - Choose model with highest accuracy

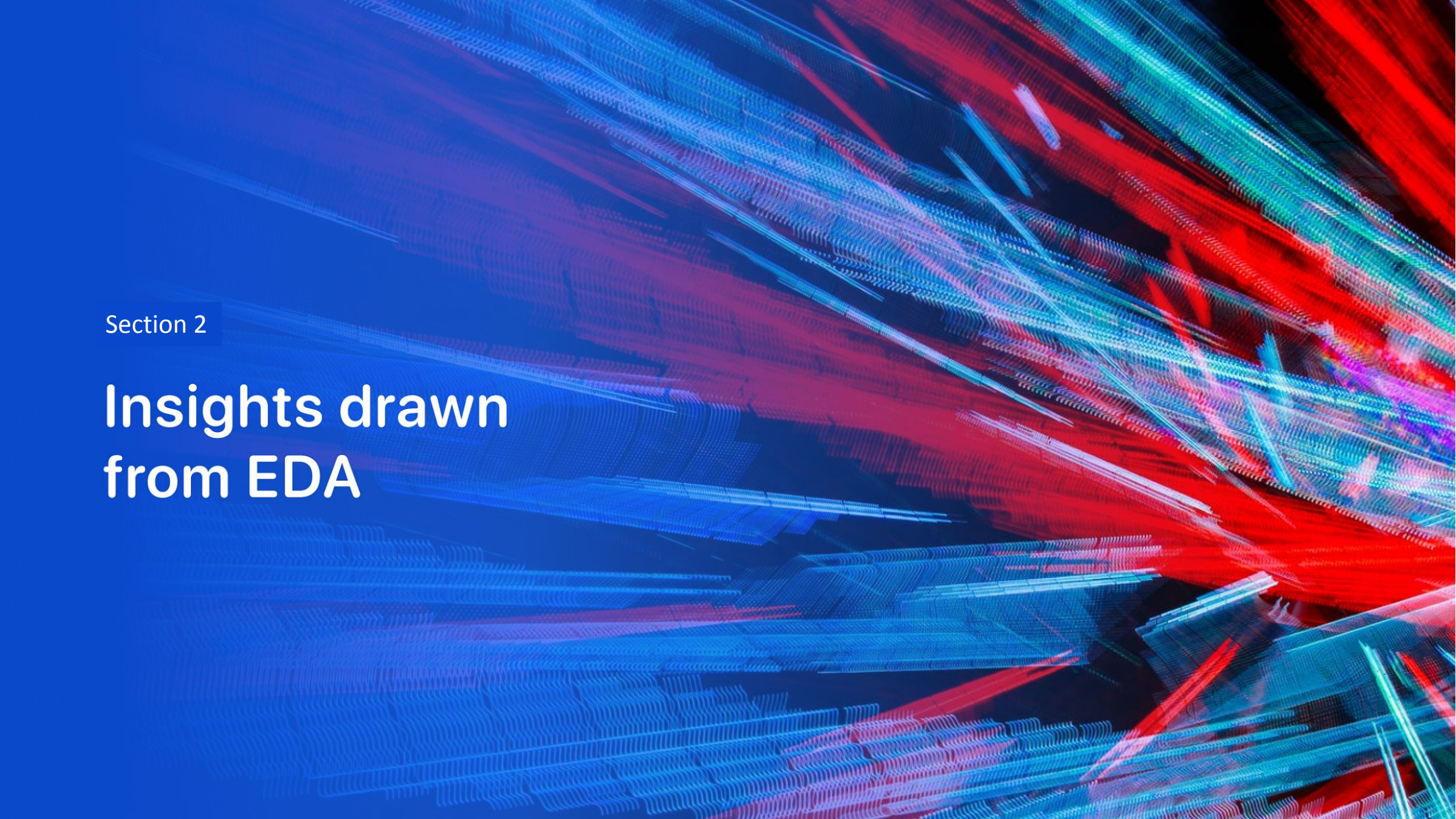
Machine Learning prediction notebook link

Results

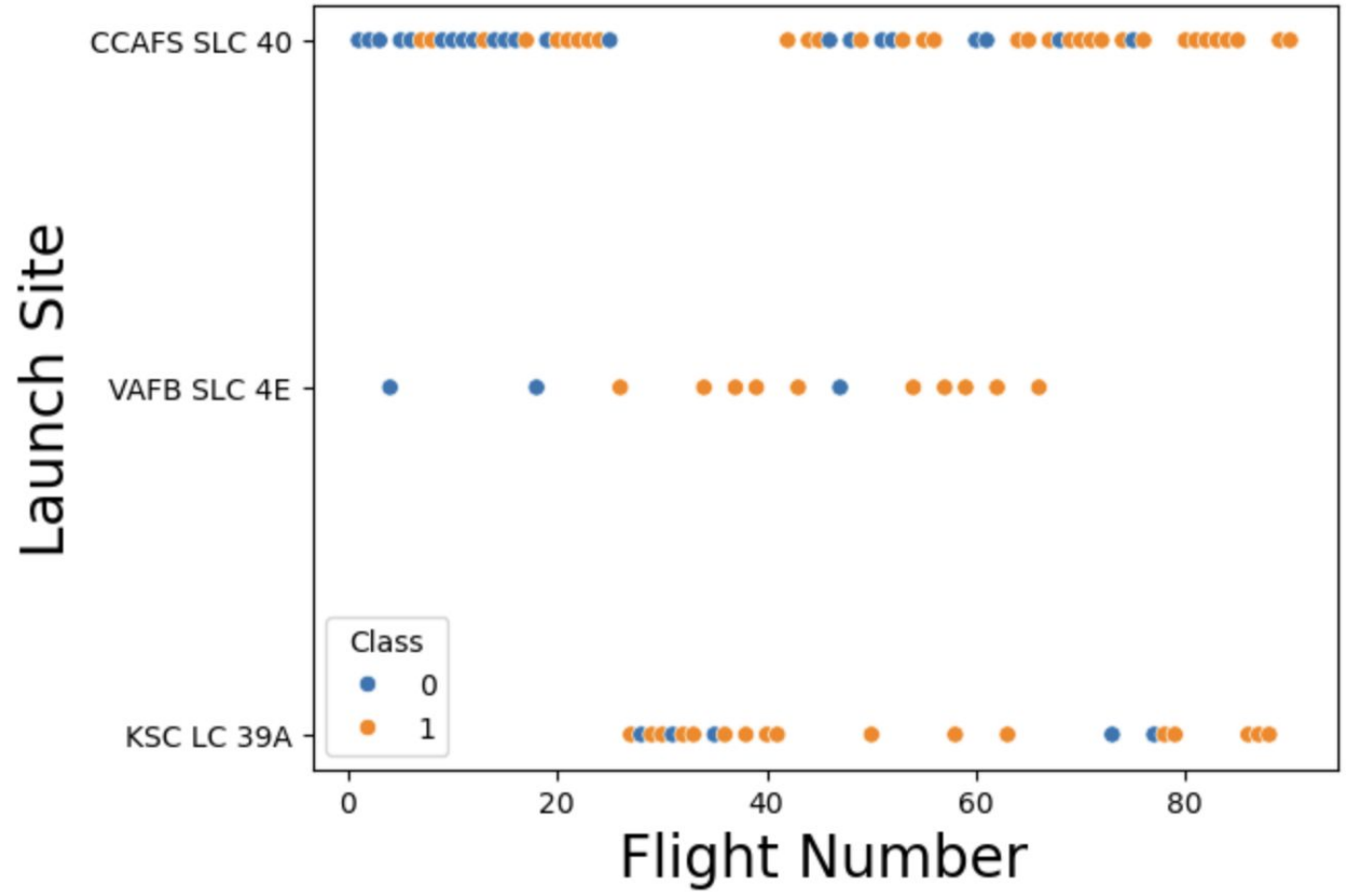
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

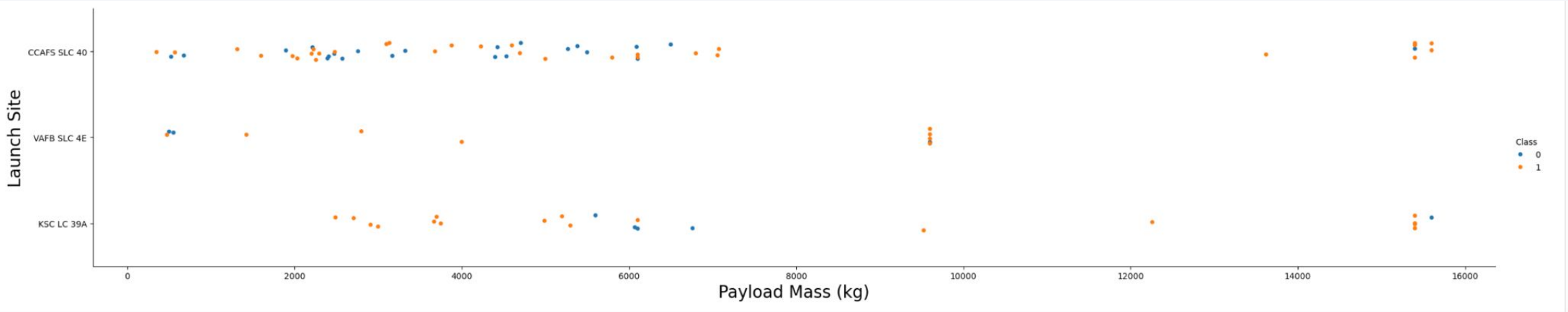


Flight Number vs. Launch Site



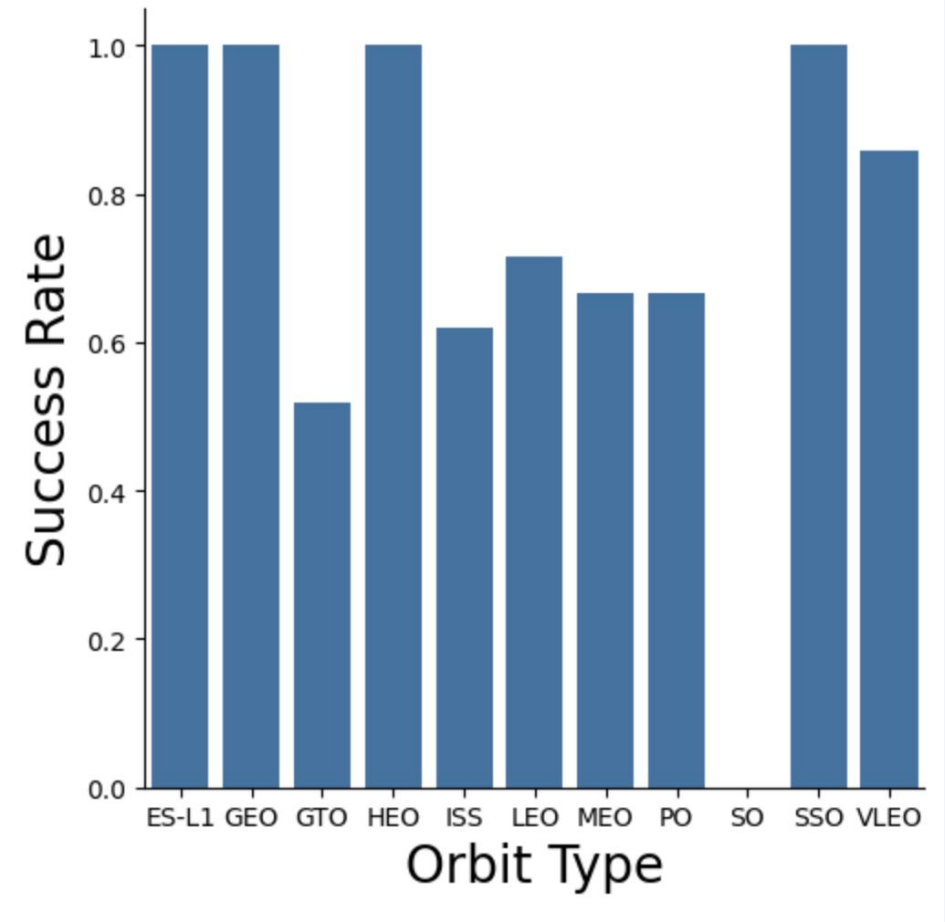
For each launch site, the success rate increases over time.

Payload vs. Launch Site



Payload size appears to have a slight correlation to landing success rate.

Success Rate vs. Orbit Type

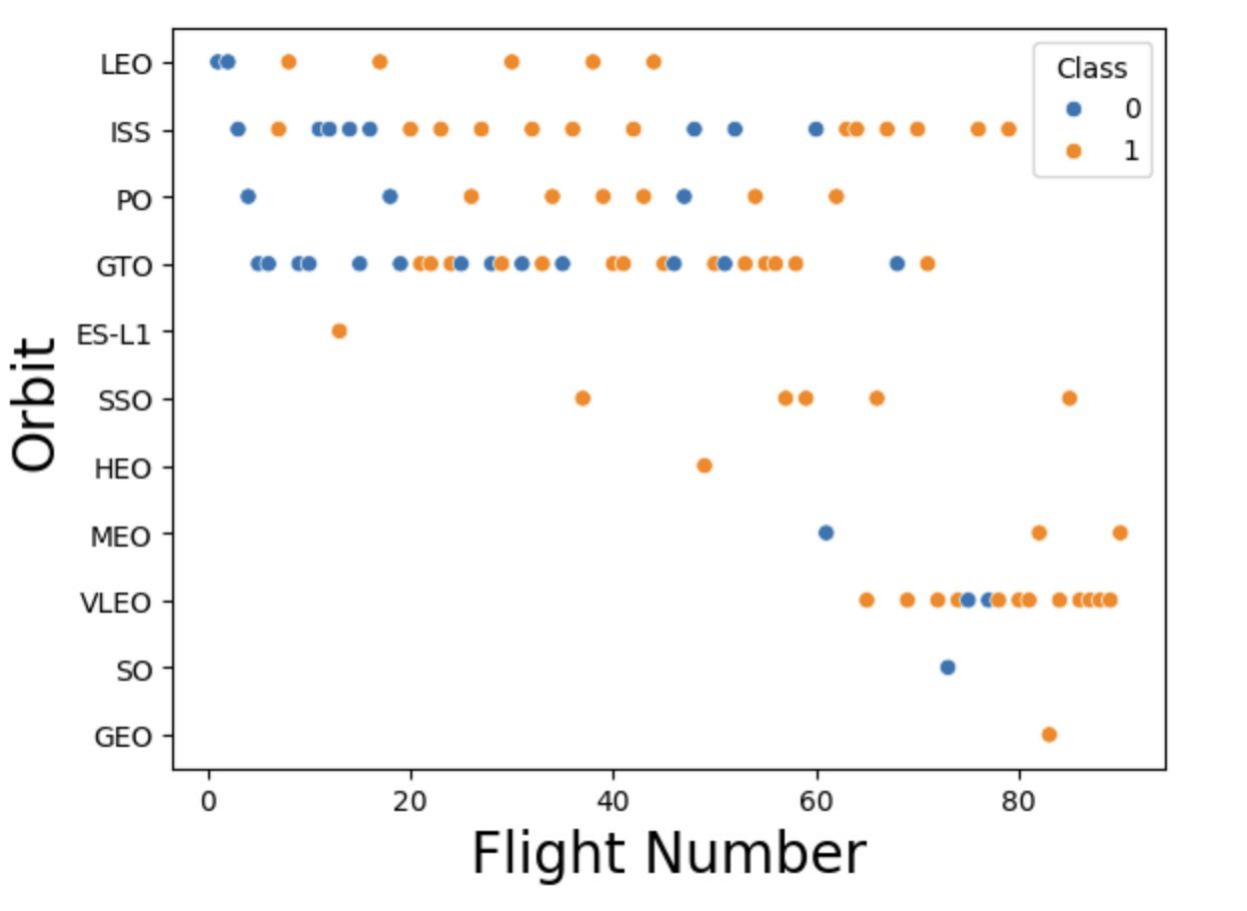


Four orbit types stand out with a great success rate:

- ES-L1
- GEO
- HEO
- SSO

On the other hand, GTO and ISS orbits have a much lower success rate.

Flight Number vs. Orbit Type

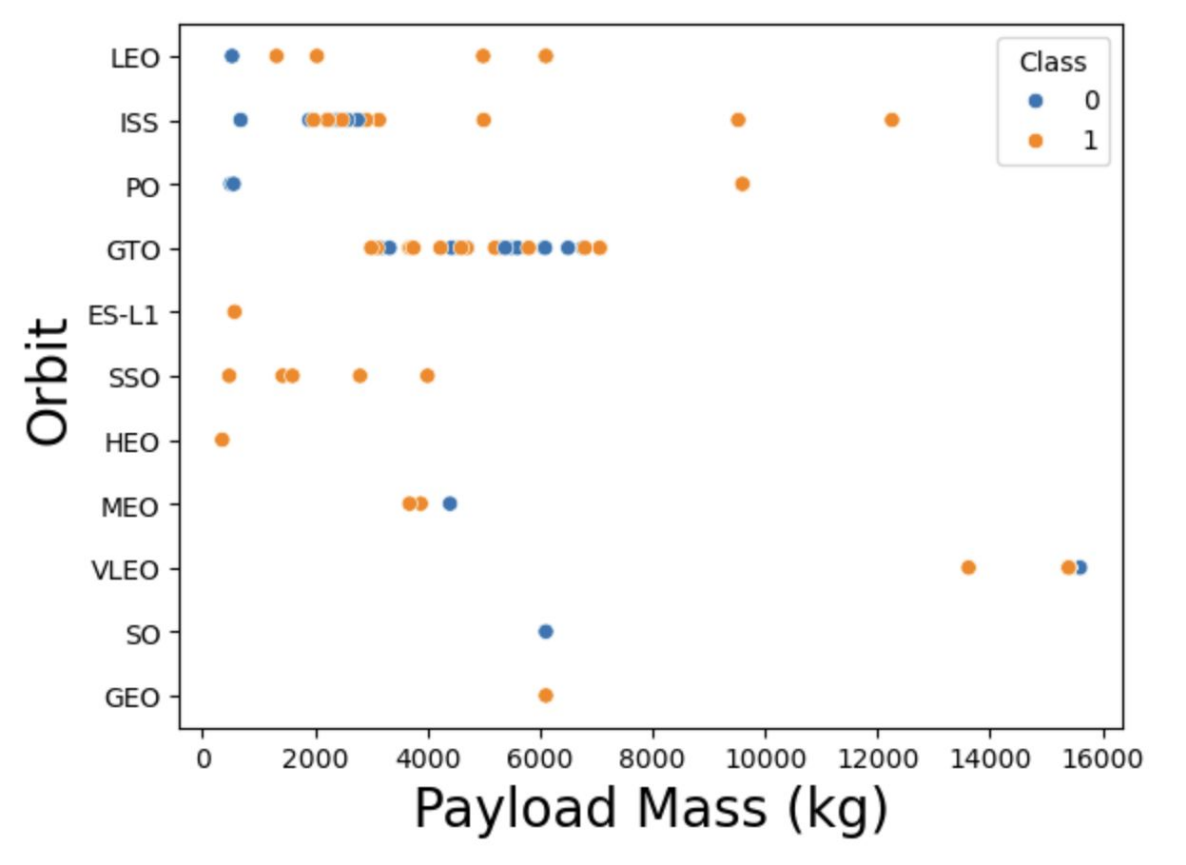


In general for each orbit type, landing success is improving over time.

LEO (Low-Earth Orbit) shows that progression very well.

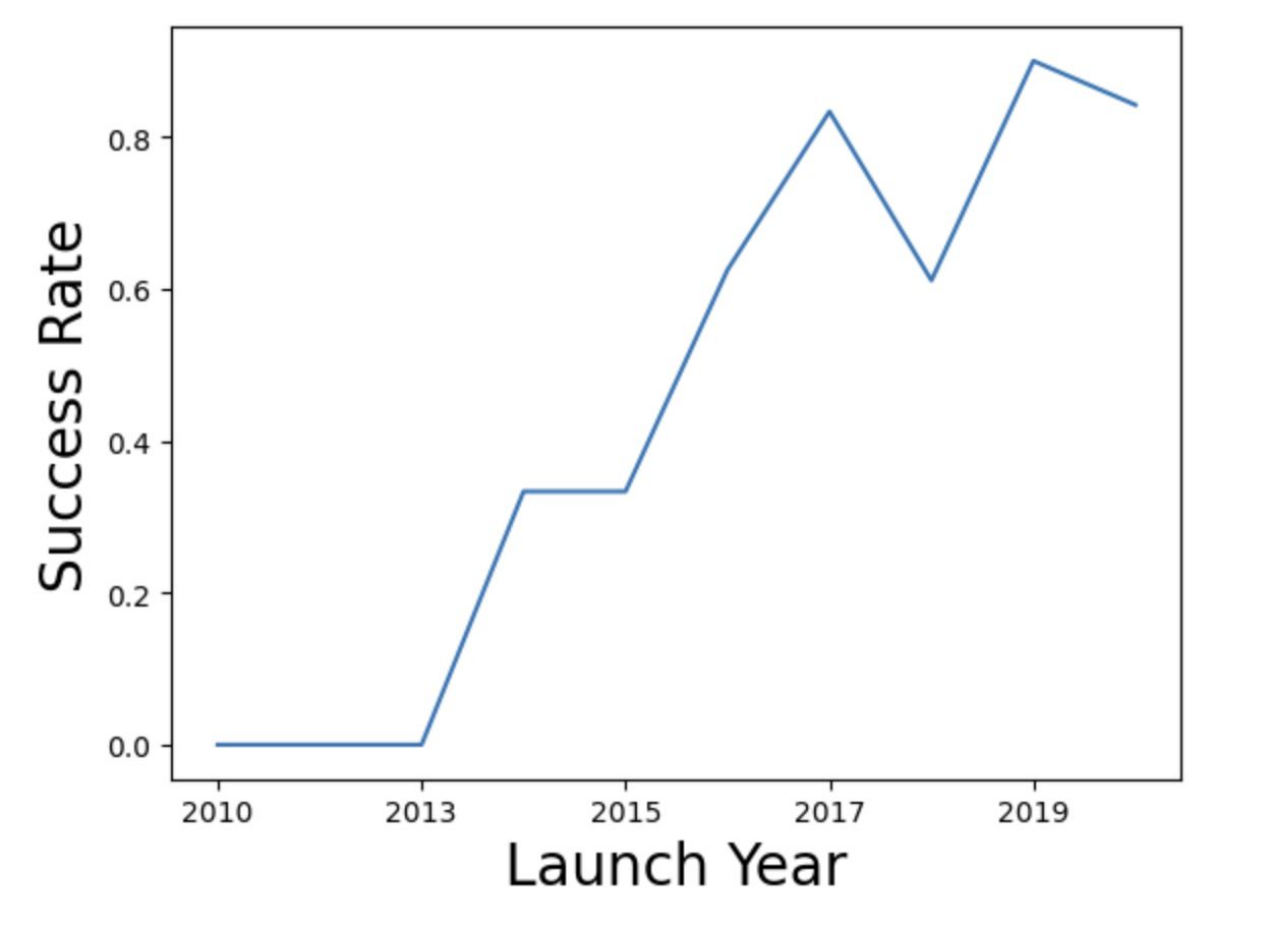
Notable is that ISS orbit launches had a spate of failures well into the program.

Payload vs. Orbit Type



While there is a correlation between payload size and orbit type, it doesn't appear from this plot that landing success rates are significantly affected.

Launch Success Yearly Trend



There is a clear trend toward improving launch success rate as more experience is gained.

The significant dip in 2018 should be examined for a root cause, so that it's a pattern that can be avoided in the future.

All Launch Site Names

SQL Query:

select distinct "Launch_Site" from SPACEXTABLE

Explanation:

‘distinct’ removes any duplicate Launch_Site from this query.

Result

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

Launch Site Names Begin with 'CCA'

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

SQL Query:

select * from SPACEXTABLE where launch_site like 'CCA%' limit 5

Explanation:

“launch_site like ‘CCA%’” ensures that strings starting with CCA match
‘limit 5’ shows the first 5 records found

Total Payload Mass

SUM(PAYLOAD_MASS__KG_)
45596

SQL Query:

SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE Customer = 'NASA (CRS)

Explanation:

For records where ‘Customer’ matches NASA (CRS) add the Payload_Mass in kg and return the total

Average Payload Mass by F9 v1.1

AVG(PAYLOAD_MASS__KG_)
2928.4

SQL Query:

SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE Booster_Version = 'F9 v1.1'

Explanation:

Select records where Booster_Version matches ‘F9 v1.1’ and calculate the average of the payload mass

First Successful Ground Landing Date

Date
2015-12-22

SQL Query:

SELECT Date FROM SPACEXTABLE WHERE Landing_Outcome = 'Success (ground pad)' ORDER BY Date ASC LIMIT 1

Explanation:

Select all records where ‘Landing_Outcome’ is ‘Success (ground pad)’ and sort them by ‘Date’ in ASCending order.

Return only 1 record, which will be the earliest successful ground landing date.

Successful Drone Ship Landing with Payload between 4000 and 6000

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

SQL Query:

```
SELECT Booster_Version FROM SPACEXTABLE WHERE Landing_Outcome = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000
```

Explanation:

Select records where ‘Landing_Outcome’ is a successful ship landing and whose payload is between 4000 and 6000 kg

Return the Booster_Version of these records.

Total Number of Successful and Failure Mission Outcomes

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

SQL Query:

SELECT Mission_Outcome, COUNT(*) FROM SPACEXTABLE GROUP BY Mission_Outcome

Explanation:

This query took the approach of detailing the Mission_Outcome and grouping them and getting the total count for each outcome set.

Boosters Carried Maximum Payload

SQL Query:

```
select Booster_Version from SPACEXTABLE where payload_mass__kg_ = \
        (select max(payload_mass__kg_) from SPACEXTABLE);
```

Explanation:

The subquery first finds the maximum payload mass in the database and then uses that mass to find which Booster_Version's have carried that payload.

Note that all are F9 B5

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

MONTH	Booster_Version	Launch_Site
01	F9 v1.1 B1012	CCAFS LC-40
04	F9 v1.1 B1015	CCAFS LC-40

SQL Query:

```
SELECT substr("DATE", 6, 2) AS MONTH, "BOOSTER_VERSION", "LAUNCH_SITE" FROM SPACEXTBL \
        WHERE "LANDING_OUTCOME" = 'Failure (drone ship)' and substr("DATE",0,5) = '2015'
```

Explanation:

Select all records where the year is 2015 and the ‘Landing_Outcome’ is a drone ship failed landing.

For these report the Month, Booster_Sersion and Launch_Site

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

Landing_Outcome	COUNT("LANDING_OUTCOME")
Success (drone ship)	5
Success (ground pad)	3

SQL Query:

```
SELECT "LANDING_OUTCOME", COUNT("LANDING_OUTCOME") FROM SPACEXTBL \
        WHERE "DATE" >= '2010-06-04' and "DATE" <= '2017-03-20' and \
        "LANDING_OUTCOME" LIKE '%Success%' GROUP BY "LANDING_OUTCOME" \
        ORDER BY COUNT("LANDING_OUTCOME") DESC ;
```

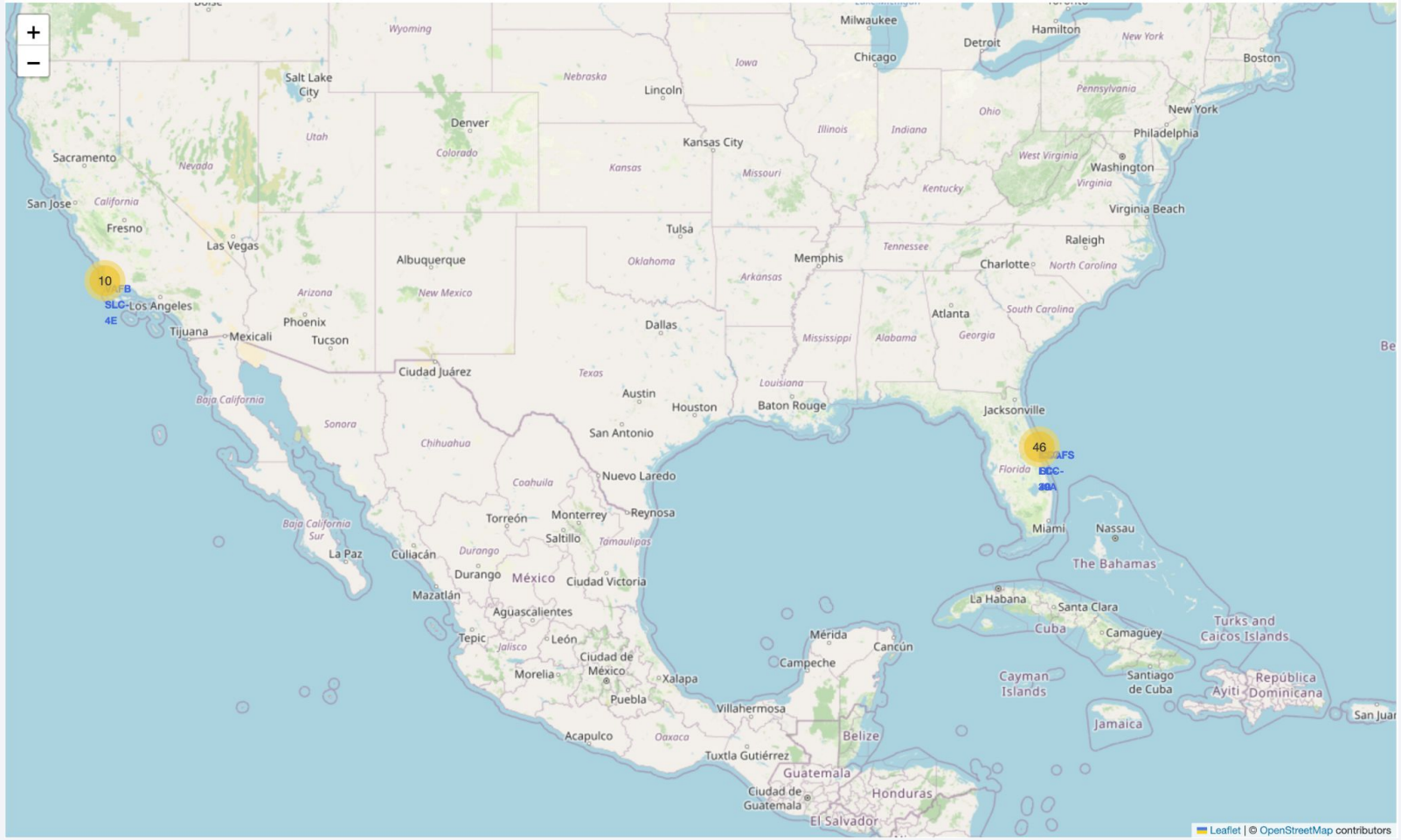
Explanation:

Select record where the date is between 6/4/2010 and 3/20/2017 and Landing_Outcome was success of some kind; organize these by Landing_Outcome and sort them by the number of these outcomes DESCending.

Report the ‘Landing_Outcome’ and Count

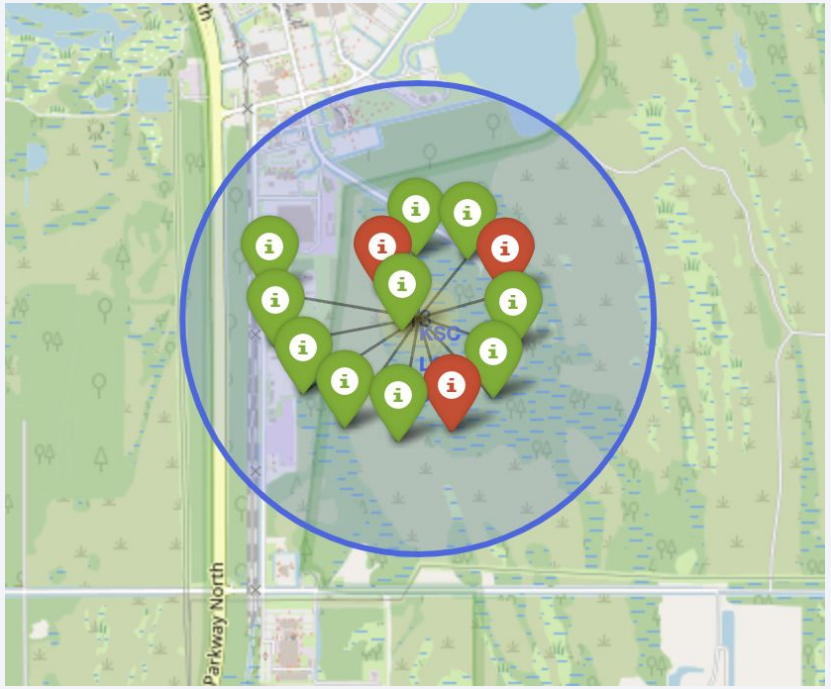

Section 3
Launch Sites
Proximities Analysis

Folium Map - Launch Sites

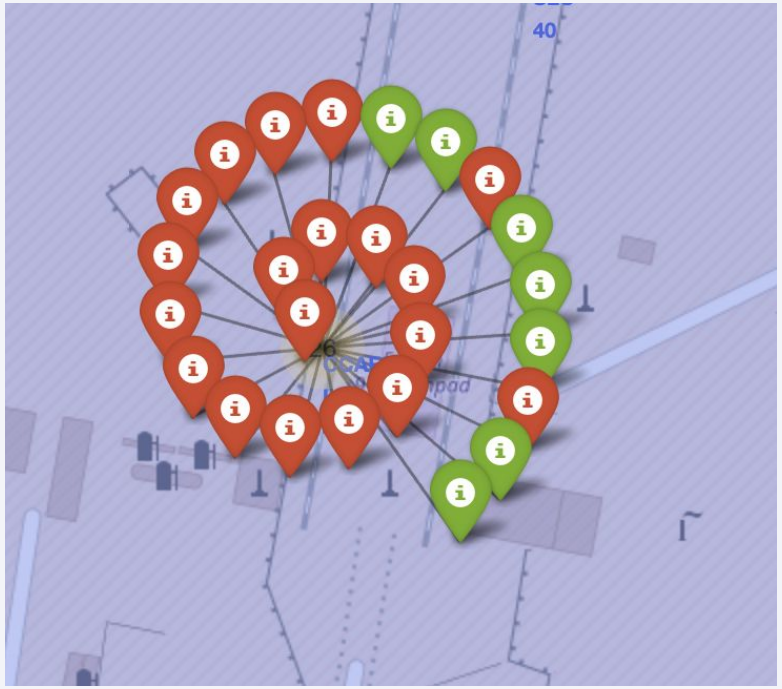


From this map, we can see that launch sites are near coastlines.

Folium Map - Marker Clusters



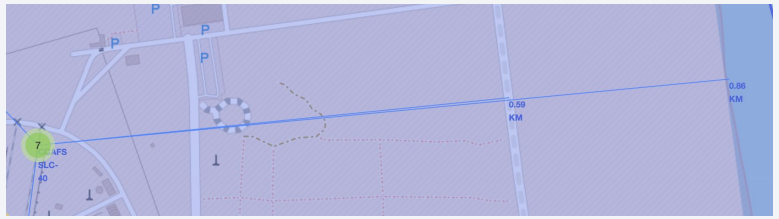
KSC LC-39A



CCAFS LC-40

Green markers indicate successful landings for the launch, and red ones indicate failure. The difference in success rate of KSC LC-39A and CCAF LC-40 is quite remarkable.

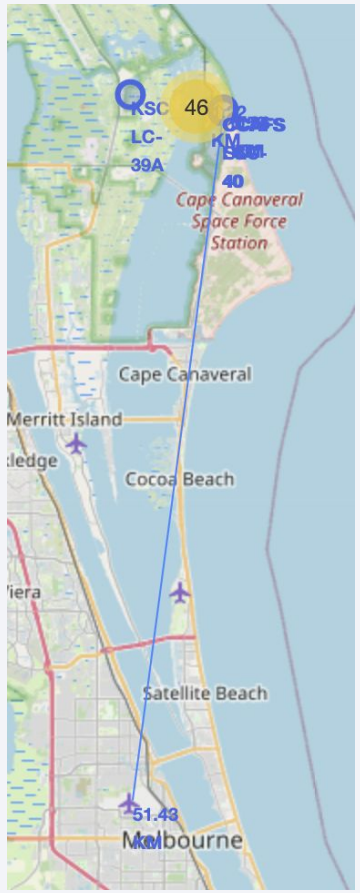
Folium Map - Distances



Coast - 0.86 km

Highway - 0.58 km

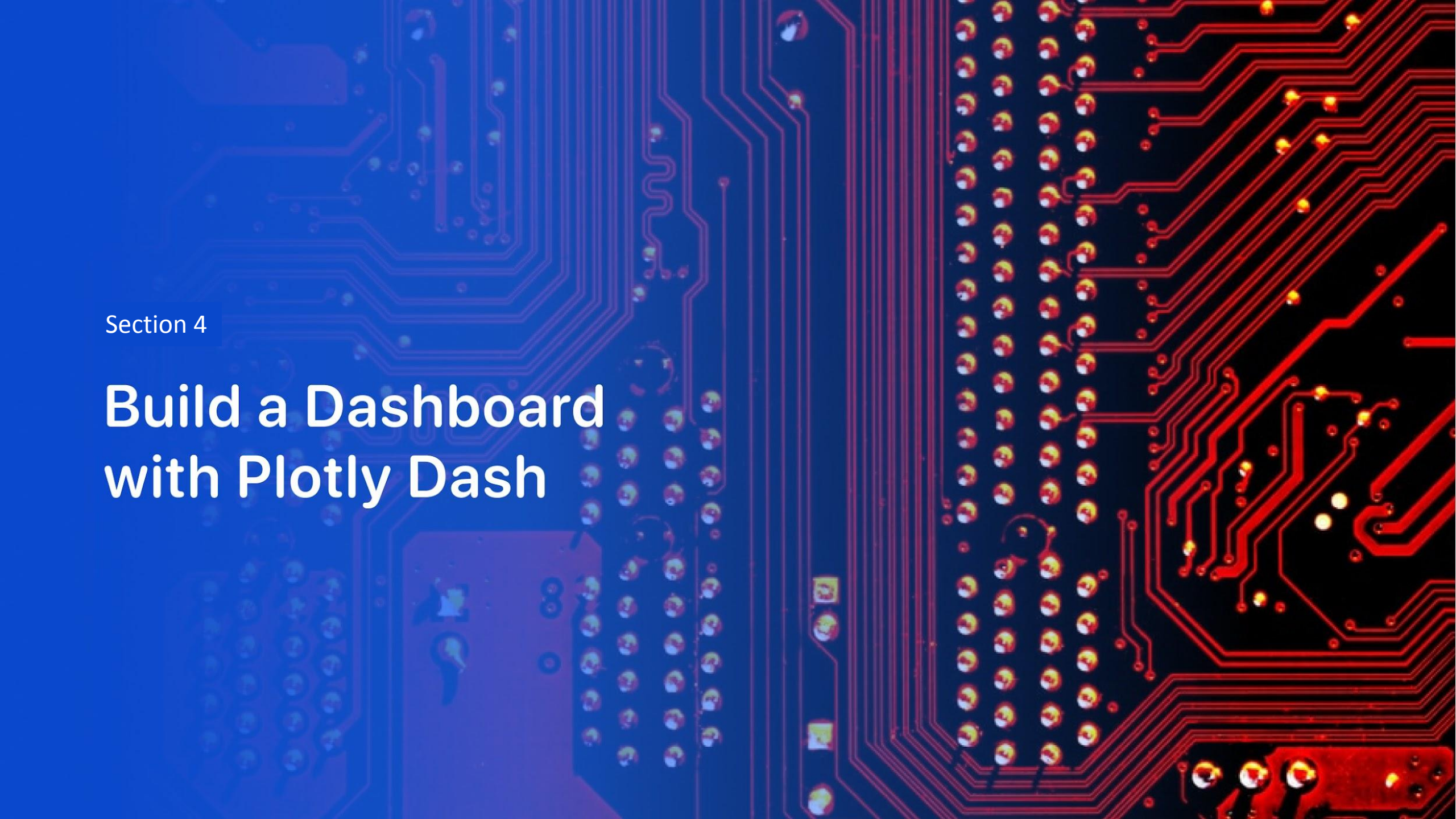
Railroad - 1.22 km



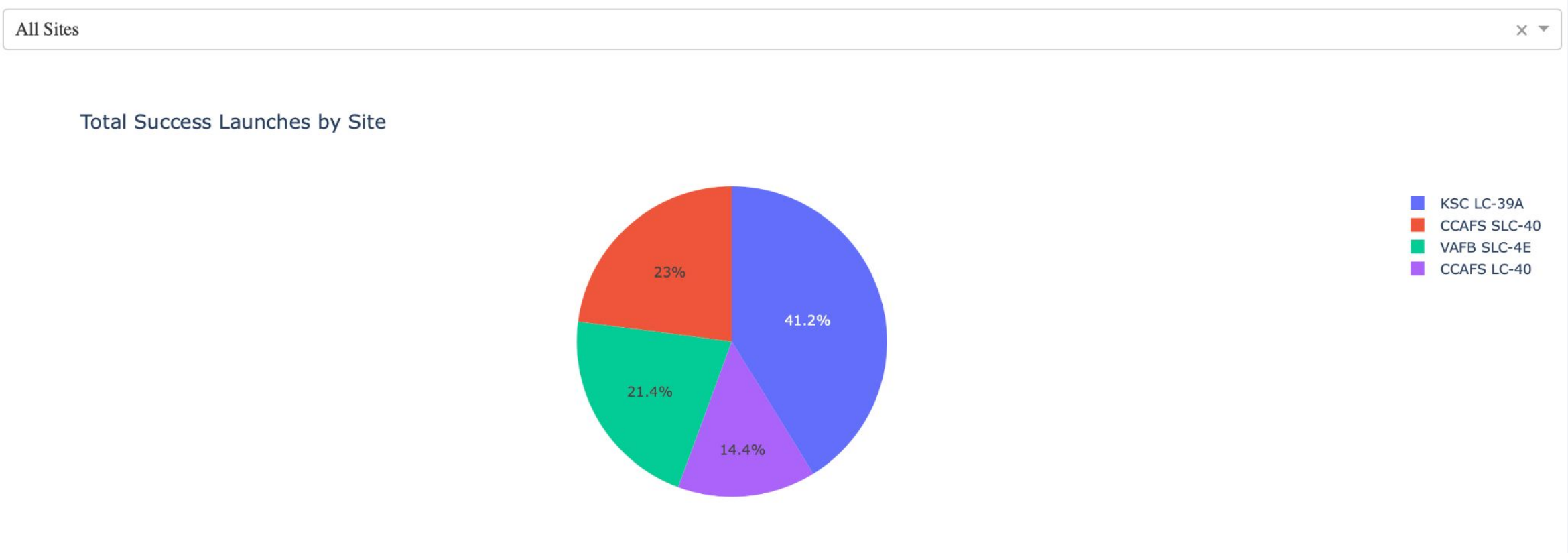
City - 51.49 km

Conclusions:

- Close proximity to supply lines, such as highway and railroads
- Close proximity to coastline in case of catastrophic failure
- Significant distance to city to ensure safety

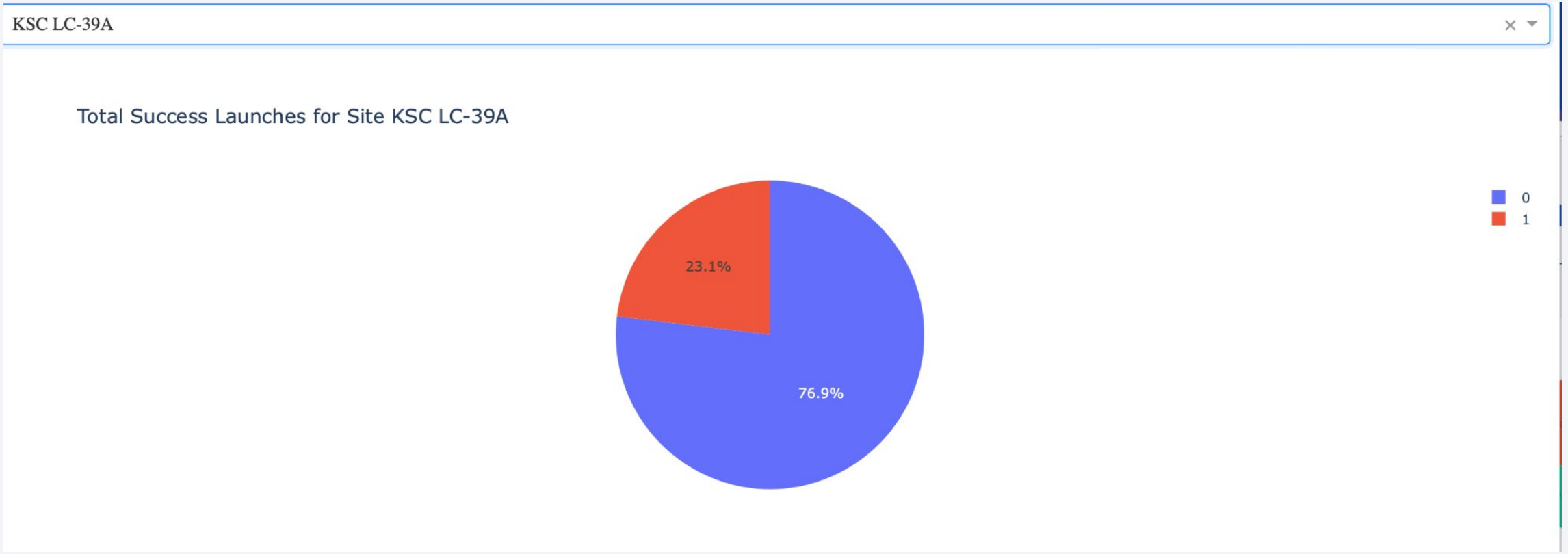

Section 4
Build a Dashboard with Plotly Dash

Dashboard - Total Launch Success by Site



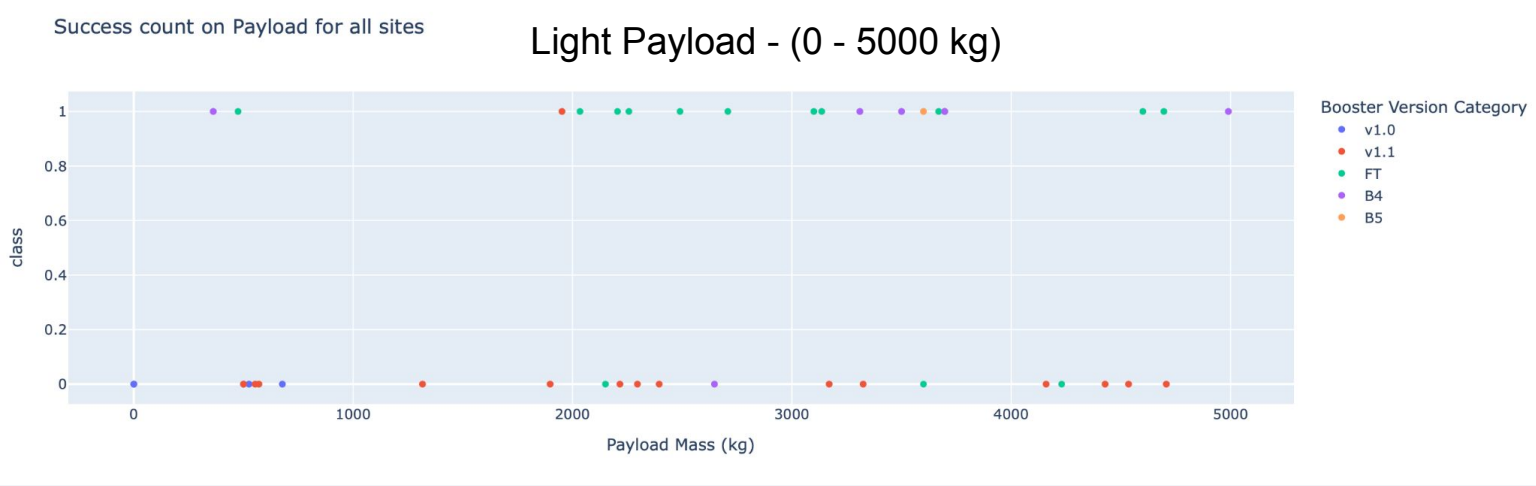
Key observation is the highest number of successful launches from KSC LC-39A.

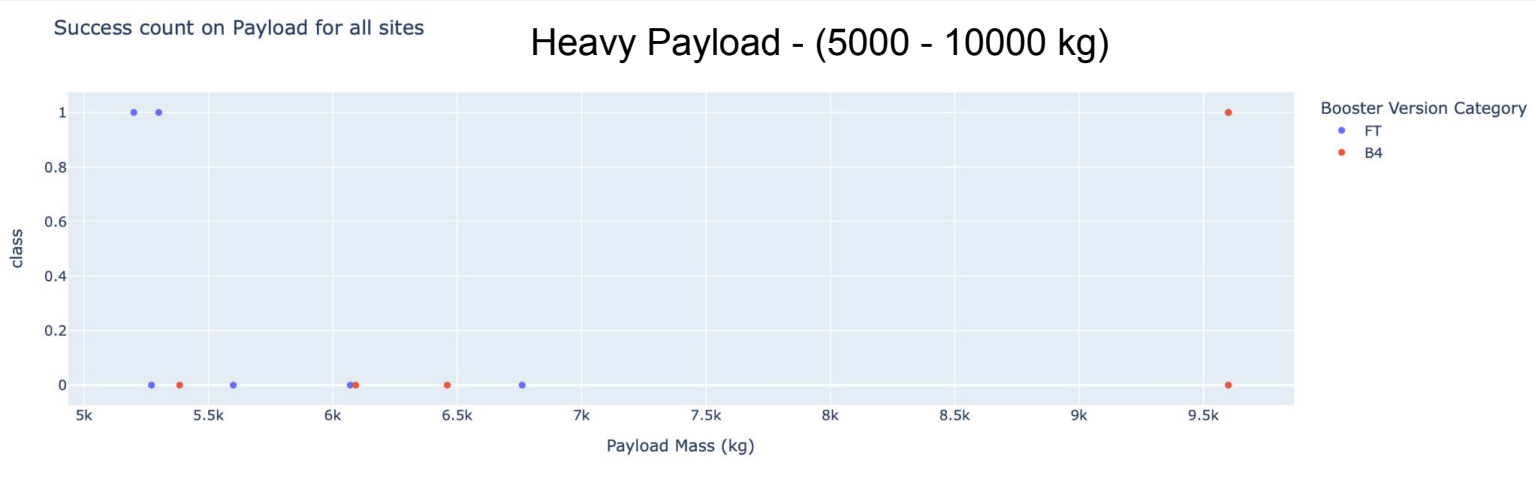
Dashboard - KSC LC-39A Success Rate



Launch site KSC 39-A comes in best with a 76.9% successful landing of first stage.

<Dashboard Screenshot 3>



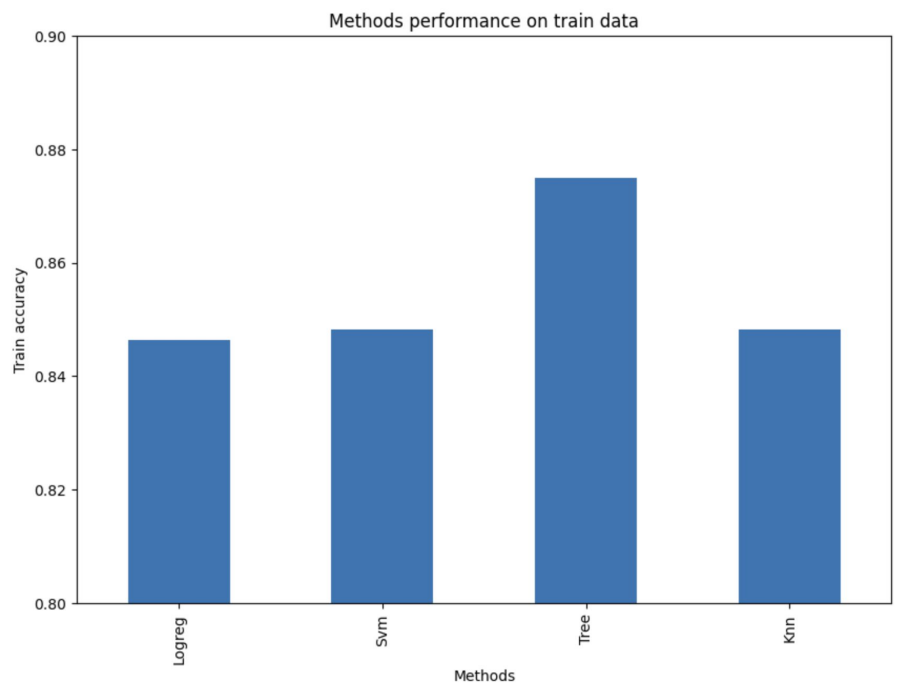


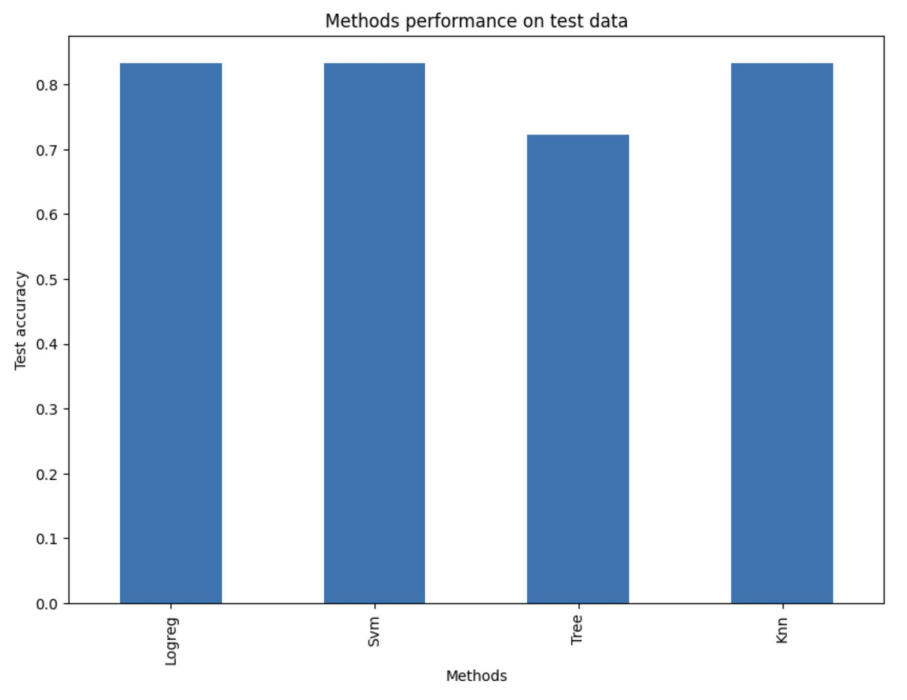
The success rate for first stage landing and recovery is significantly higher for lighter payloads than heavy ones.

Section 5

Predictive Analysis (Classification)

Classification Accuracy



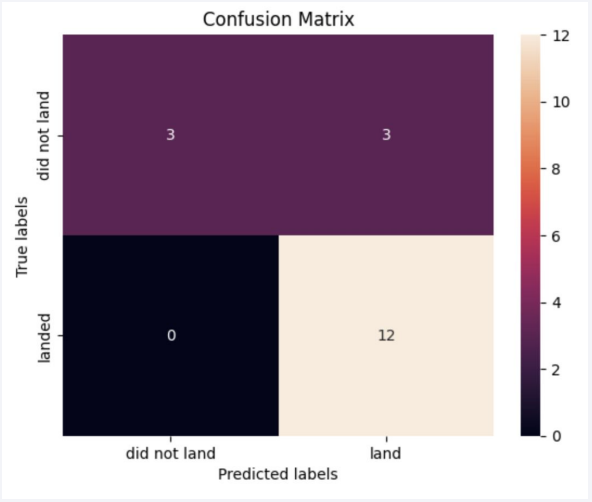


DecisionTree performance is better on test data, but lacking a bit on training data; given its propensity for further improvement, DecisionTree with an expanded training set might be interesting to explore further.

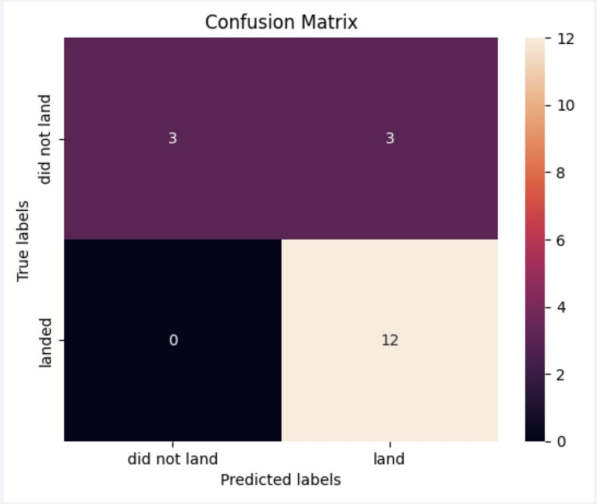
	Accuracy Train	Accuracy Test
Logreg	0.846429	0.833333
Svm	0.848214	0.833333
Tree	0.875000	0.722222
Knn	0.848214	0.833333

Confusion Matrix

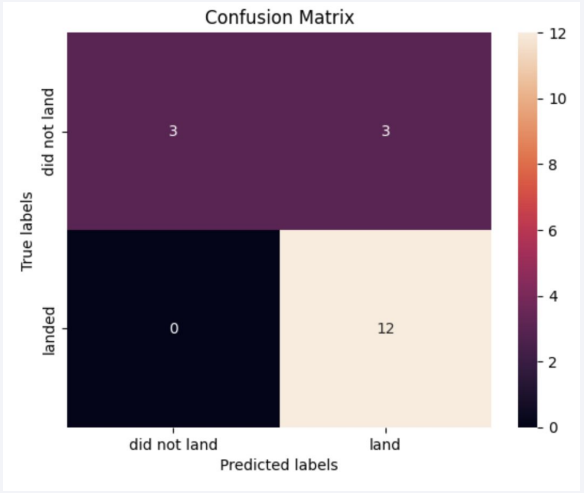
KNN



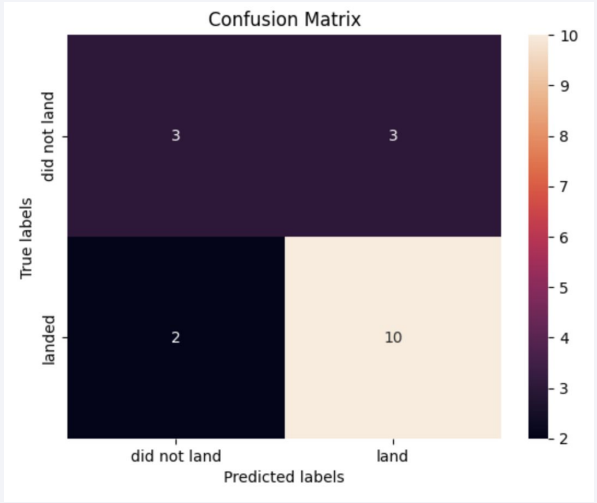
SVM



Logistic Regression



Decision Tree



Decision Tree underperforms a bit among these models.

Overall, there are still too many false positives, due to a rather small training set.

Further data should be gathered, so that models can be refined.

Conclusions

- Successful recovery of first stage vehicles by landing them has improved significantly over time. In particular, there is significant improvement with newer booster generations. Launch site influence and orbit type only have secondary influence on landing success.
- Orbit types with highest success rates thus far are GEO, HEO, SSO, ES-L1
- Payload mass appears to have an influence, although not decisively so. In general, lighter payloads perform better than heavy ones.
- Even though KSC LC-39A appears to be the most successful launch site, there is insufficient data to come to a clear conclusion on the factors that make it better than others.
- Machine Learning prediction is still generating too many false positives to be of great use at this time; it should be revisited as more data becomes available.

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!