Winning the Space Race with Data Science

Will Nobles

11/26/21

Outline




Executive Summary

METHODOLOGIES USED

- Data collection and wrangling
- EDA with Visualization, SQL
- Data visualization with Folium, Plotly Dash app
- Machine learning classification

RESULTS SUMMARY

- EDA results from graphs, SQL queries
- Visualization of launch site data on maps, graphs in screenshots
- ML model accuracy, confusion matrix



Introduction

PROJECT BACKGROUND

SpaceX can launch Falcon9 rockets at a fraction of the cost of its competitors. This project will help SpaceY by investigating the cost of each stage 1 launch and if it will be successful.

QUESTIONS

- What is the average success rate for a launch?
- Is there a relationship between payload and launch outcome?
- Can the landing outcome be predicted based on the available data?

Methodology

Methodology

DATA COLLECTION

- Describe how data was collected

DATA WRANGLING

- Describe how data was processed

EXPLORATORY DATA ANALYSIS (EDA) WITH VISUALIZATION, SQL

INTERACTIVE VISUAL ANALYTICS WITH FOLIUM, PLOTLY DASH

PREDICTIVE ANALYSIS WITH CLASSIFICATION MODELS

- How to build, tune, evaluate classification models

Data Collection

SpaceX API

PROCESS

1. Request launch data from API
2. Normalize JSON response, convert to DataFrame
3. Use functions to retrieve request data, add to new DataFrame
4. Filter and clean data
5. Export new DataFrame to CSV

GITHUB LINK TO JUPYTER NOTEBOOK

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

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

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

launch_dict = {'FlightNumber': list(data['flight_number']),
               'Date': list(data['date']), 'BoosterVersion':BoosterVersion,
               'PayloadMass':PayloadMass, 'Orbit':Orbit, 'LaunchSite':LaunchSit
               'Outcome':Outcome, 'Flights':Flights, 'GridFins':GridFins,
               'Reused':Reused, 'Legs':Legs, 'LandingPad':LandingPad,
               'Block':Block, 'ReusedCount':ReusedCount, 'Serial':Serial,
               'Longitude': Longitude,'Latitude': Latitude}
df_launch = pd.DataFrame(launch_dict)

data_falcon9 = df_launch[df_launch['BoosterVersion'] !=
                         'Falcon 1']
data_falcon9['PayloadMass'].replace(
    {np.nan: data_falcon9['PayloadMass'].mean()},
    inplace=True)

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

Data Collection

Web Scraping

PROCESS

1. Get response from HTML request, create BeautifulSoup Object
2. Find HTML tables, extract columns from table header elements
3. Create dictionary to store table data
4. Extract HTML information, convert to DataFrame
5. Export new DataFrame to CSV

GITHUB LINK TO JUPYTER NOTEBOOK

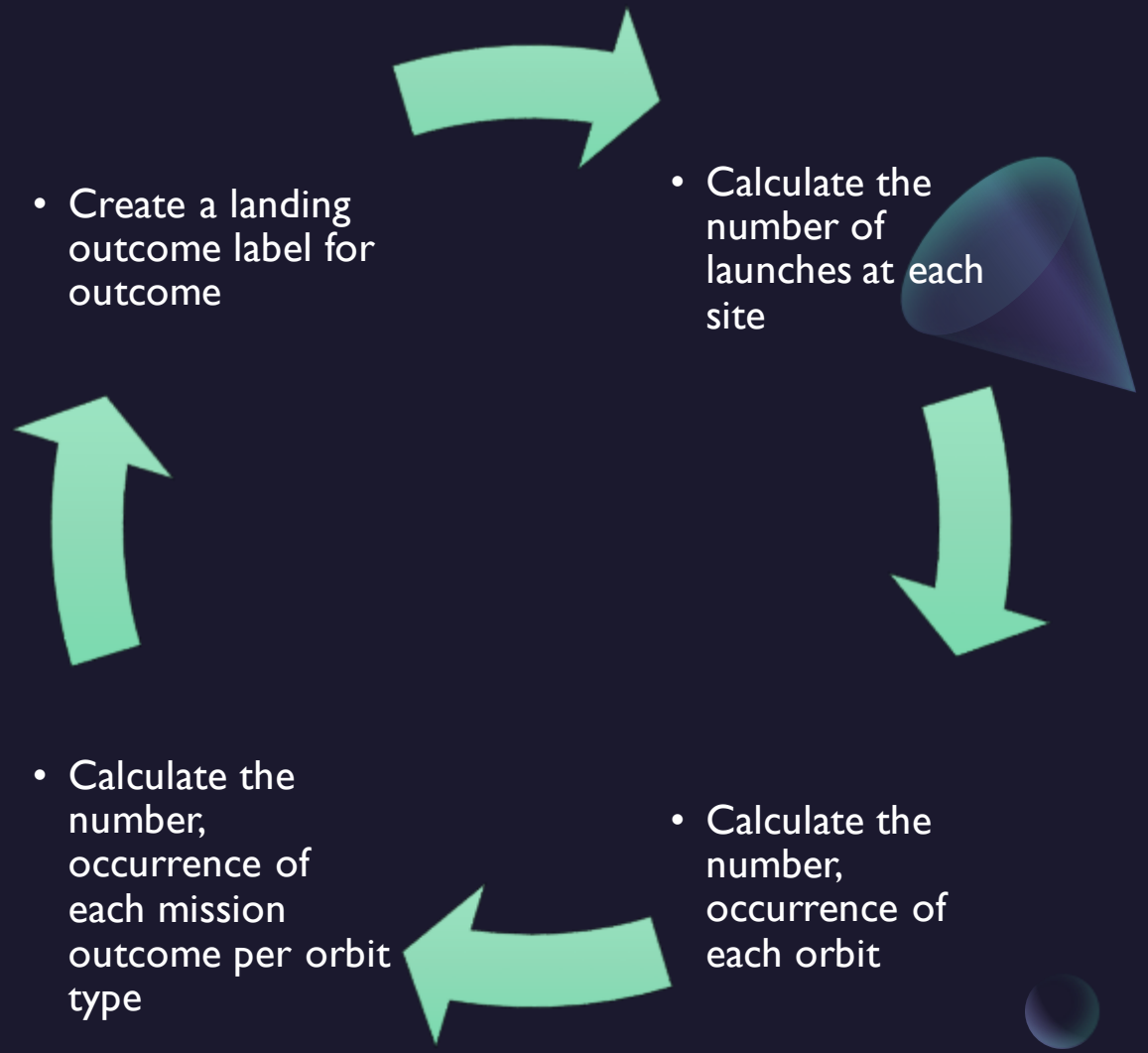
```
static_url = "https://en.wikipedia.org/w/index.php?title
response = requests.get(static_url)
soup = BeautifulSoup(response.content, 'html.parser')

html_tables = soup.find_all("table")
column_names = []
rows = first_launch_table.find_all('th')
for th in rows:
    name = extract_column_from_header(th)
    if name != None and len(name) > 0:
        column_names.append(name)

launch_dict = dict.fromkeys(column_names)
del launch_dict['Date and time ( )']
launch_dict['Flight No.'] = []
launch_dict['Launch site'] = []

extracted_row = 0
for table_number,table in enumerate(soup.find_all('table'
    for rows in table.find_all("tr"):
        if rows.th:
            if rows.th.string:
                flight_number=rows.th.string.strip()
                flag=flight_number.isdigit()
        else:
            flag=False
        row = rows.find_all('td')
        if flag:
            extracted_row += 1
            launch_dict['Flight No.'].append(flight_numbe
            print(flight_number)
            datatimelist=date_time(row[0])

df = pd.DataFrame(launch_dict)
df.to_csv('spacex_web_scraped.csv', index=False)
```

Data Wrangling

BACKGROUND

- There are several different cases in the data set where a booster did not land successfully. For example, True Ocean means the mission outcome was successfully landed to a specific region of the ocean while False Ocean means the mission outcome was unsuccessfully landed to a specific region of the ocean. The various True and False outcomes were processed to create a landing outcome label of 1 and 0, respectively, in a DataFrame.

GITHUB LINK TO JUPYTER NOTEBOOK



EDA with Visualization

SCATTER PLOTS

- Payload Mass vs Flight Number
- Launch Site vs Flight Number
- Payload Mass vs Launch Site
- Orbit Type vs Flight Number
- Orbit Type vs Payload Mass

Seaborn's catplot is a type of scatter plot that is useful for visualizing categorical data, or for visualizing numerical data with a category overlayed (I.e. hue). The above features are a combination of numerical and categorical data, which makes the catplot ideal.

BAR CHART

- Mean Class vs Orbit Type

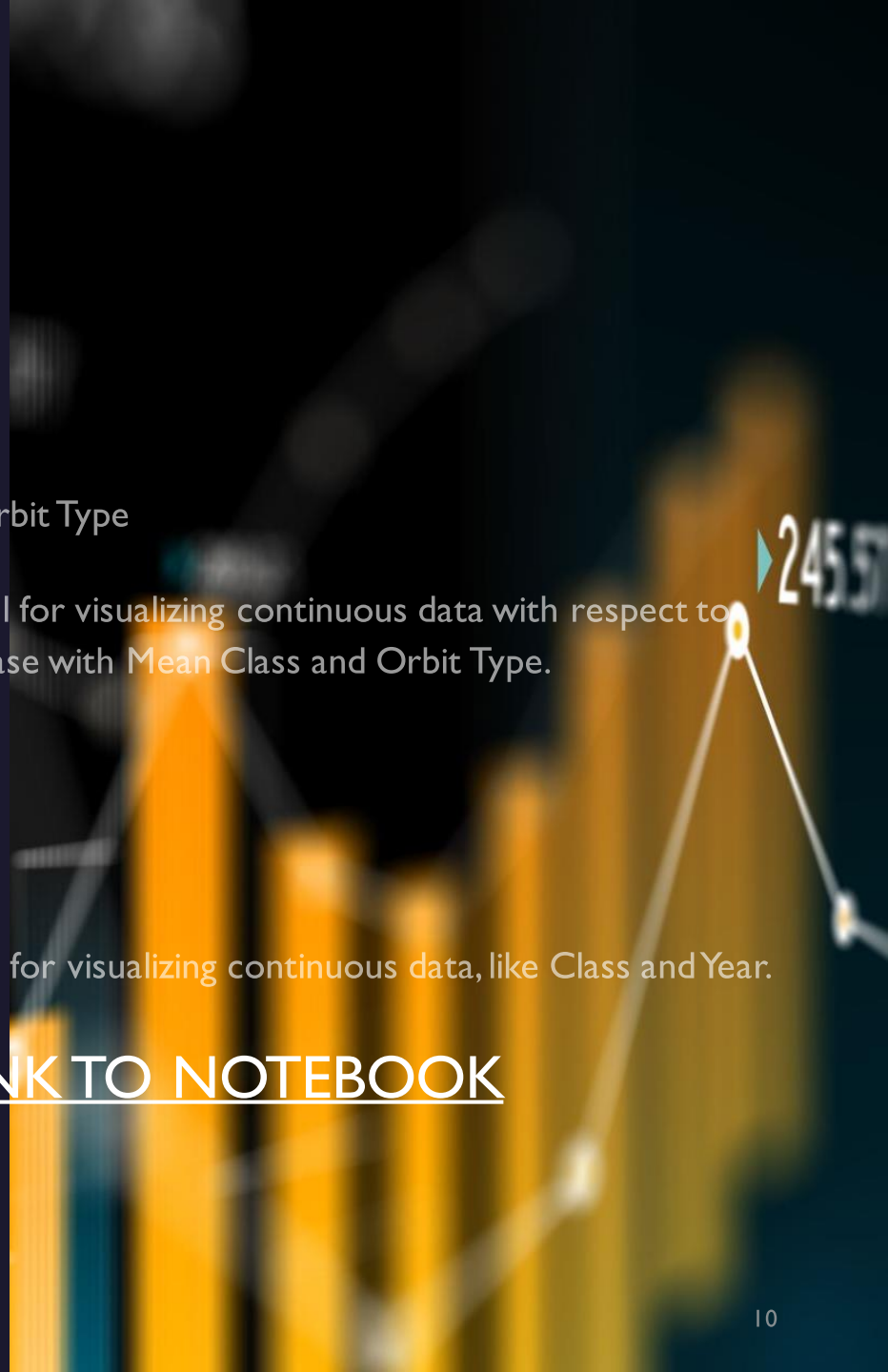
Bar charts are useful for visualizing continuous data with respect to a category, as is the case with Mean Class and Orbit Type.

LINE PLOT

- Class vs Year

Line plots are useful for visualizing continuous data, like Class and Year.

GITHUB LINK TO NOTEBOOK



EDA with SQL

SUMMARY OF QUERIES

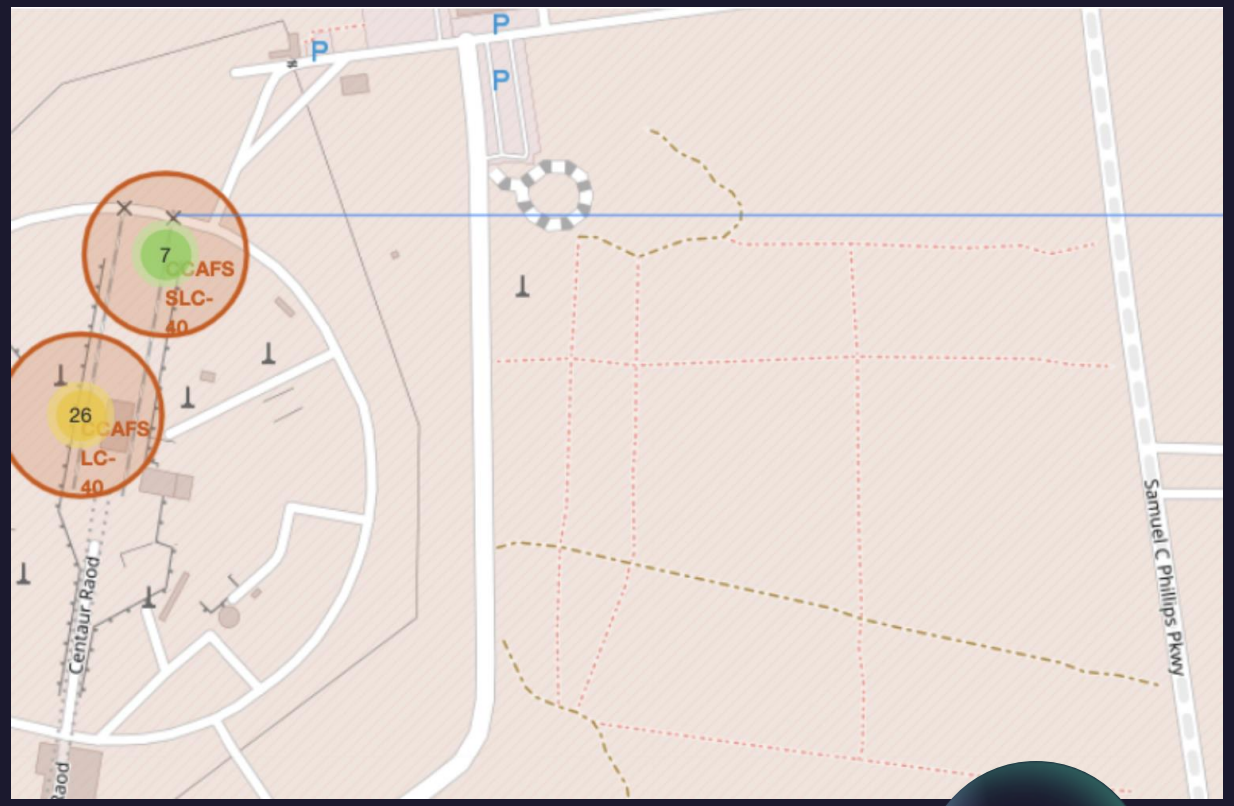
- Display the names of the unique launch sites in the space mission.
- Display 5 records where launch sites begin with the string 'CCA'.
- Display the total payload mass carried by boosters launched by NASA (CRS).
- Display average payload mass carried by booster version F9 v1.1.
- List the date when the first successful landing outcome in ground pad was achieved.
- List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000.

SUMMARY OF QUERIES

- List the total number of successful and failure mission outcomes.
- List the names of the booster_versions which have carried the maximum payload mass. Use a subquery.
- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015.
- 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.

GITHUB LINK TO NOTEBOOK

Build an Interactive Map with Folium

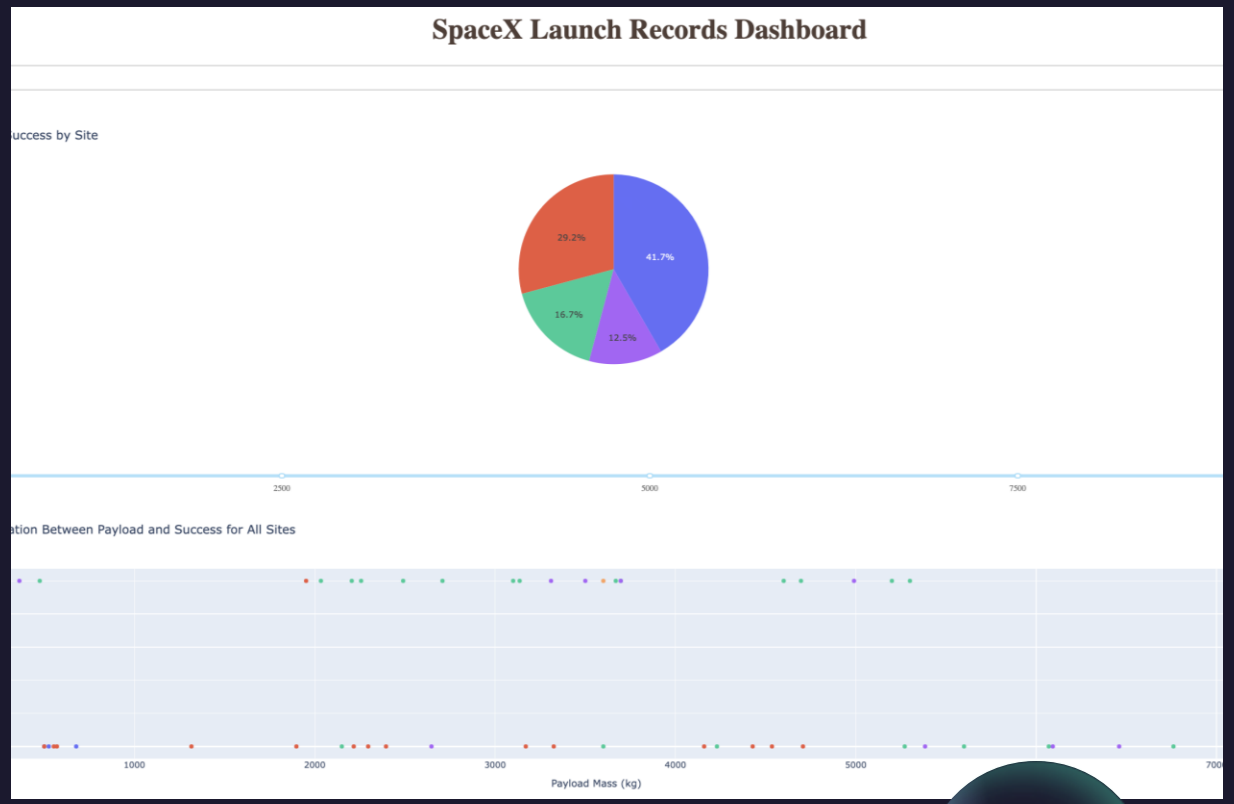


OBJECTS ADDED TO MAP

- Each launch site's location was added to the map using Circle and Marker objects to highlight the area and add text.
- The success/fail of launches for each site were added using green and red (1 and 0) Markers and MarkerClusters.
- A MousePosition object was added to get the coordinates of a point on the map, and then a Marker was added to show the distance between a launch site and its closest coastline.
- A PolyLine was drawn to show the straight line distance, and then a second Marker and Polyline were drawn to show the distance between a launch site and its closest city.

GITHUB LINK TO NOTEBOOK

Build a Dashboard using Plotly Dash



OBJECTS ADDED TO DASHBOARD

- A site dropdown menu was added, along with a pie chart showing the total number of successful launches for each site, or the percentage of successful and unsuccessful launches for a single site.
- A range slider was added to select a range of payload masses in kilograms.
- A scatter chart was added to illustrate the correlation between payload mass and class for each booster version.

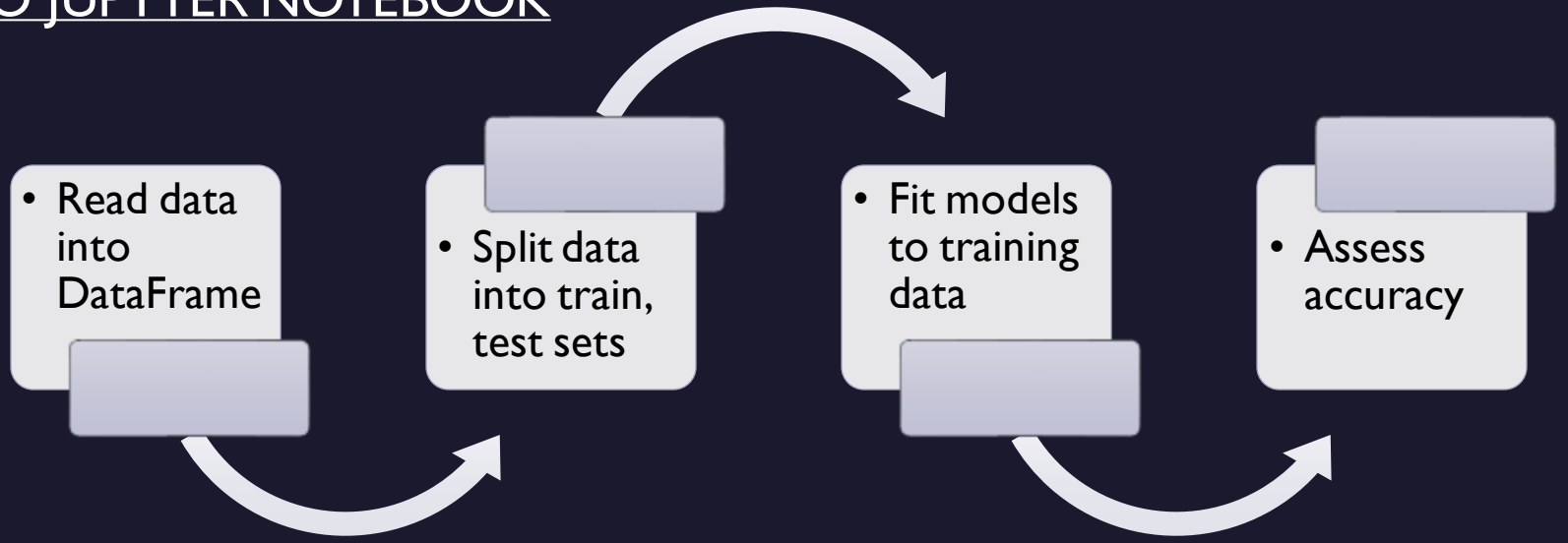
GITHUB LINK TO SOURCE CODE

Predictive Analysis using Classification

SUMMARY

- The data was read from a CSV file to a Pandas DataFrame, where it was transformed using NumPy and scikit-learn. Various machine learning models were created using GridSearchCV such as a linear model, support vector machine, decision tree classifier, and k nearest neighbors classifier. The best_params_ method was used to determine the best parameters for each model. Lastly, a confusion matrix and the score method were used to determine the accuracy for each model.

GITHUB LINK TO JUPYTER NOTEBOOK

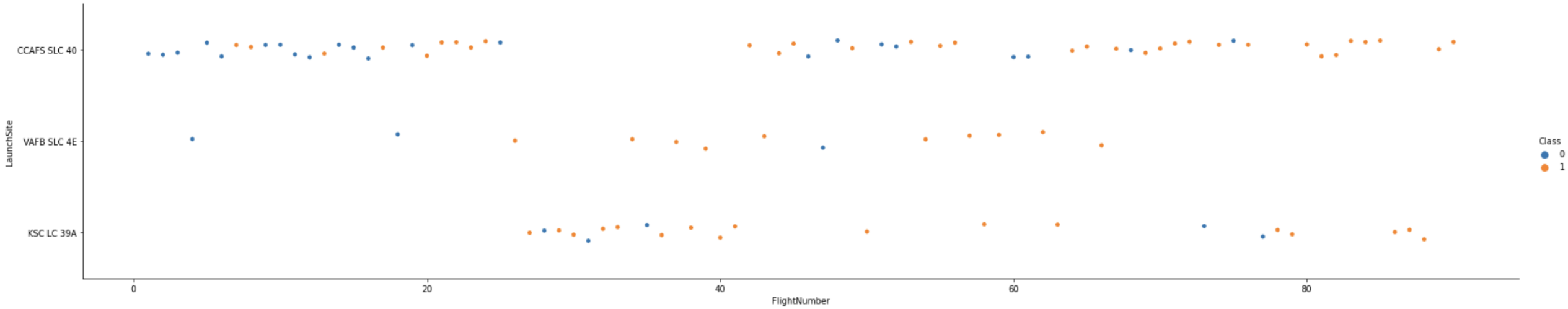




Results

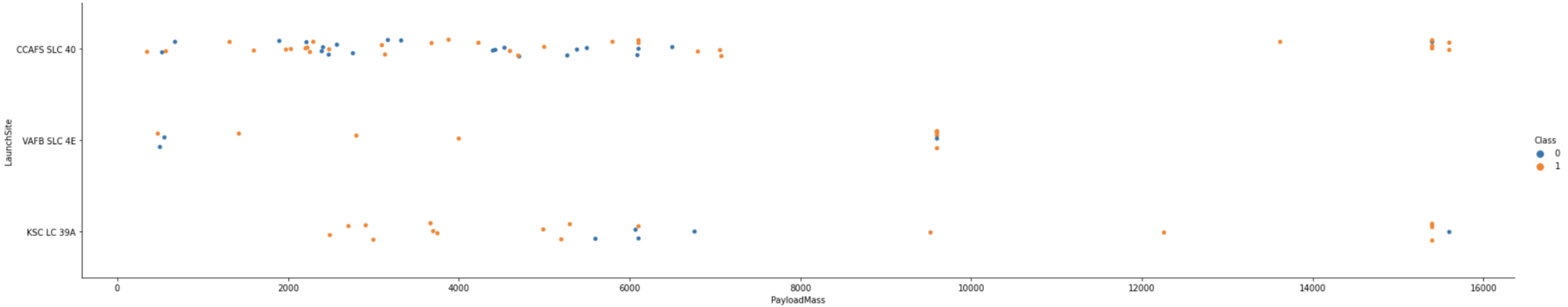
- Exploratory data analysis results
- Interactive analytics demo
- Predictive analysis results

Insights Drawn from EDA



Flight Number vs Launch Site

Earlier flights are mostly unsuccessful (blue), but as the flight number, characterized by continuous launch attempts, increases, the number of successful launches (orange) at each site appears to increase.

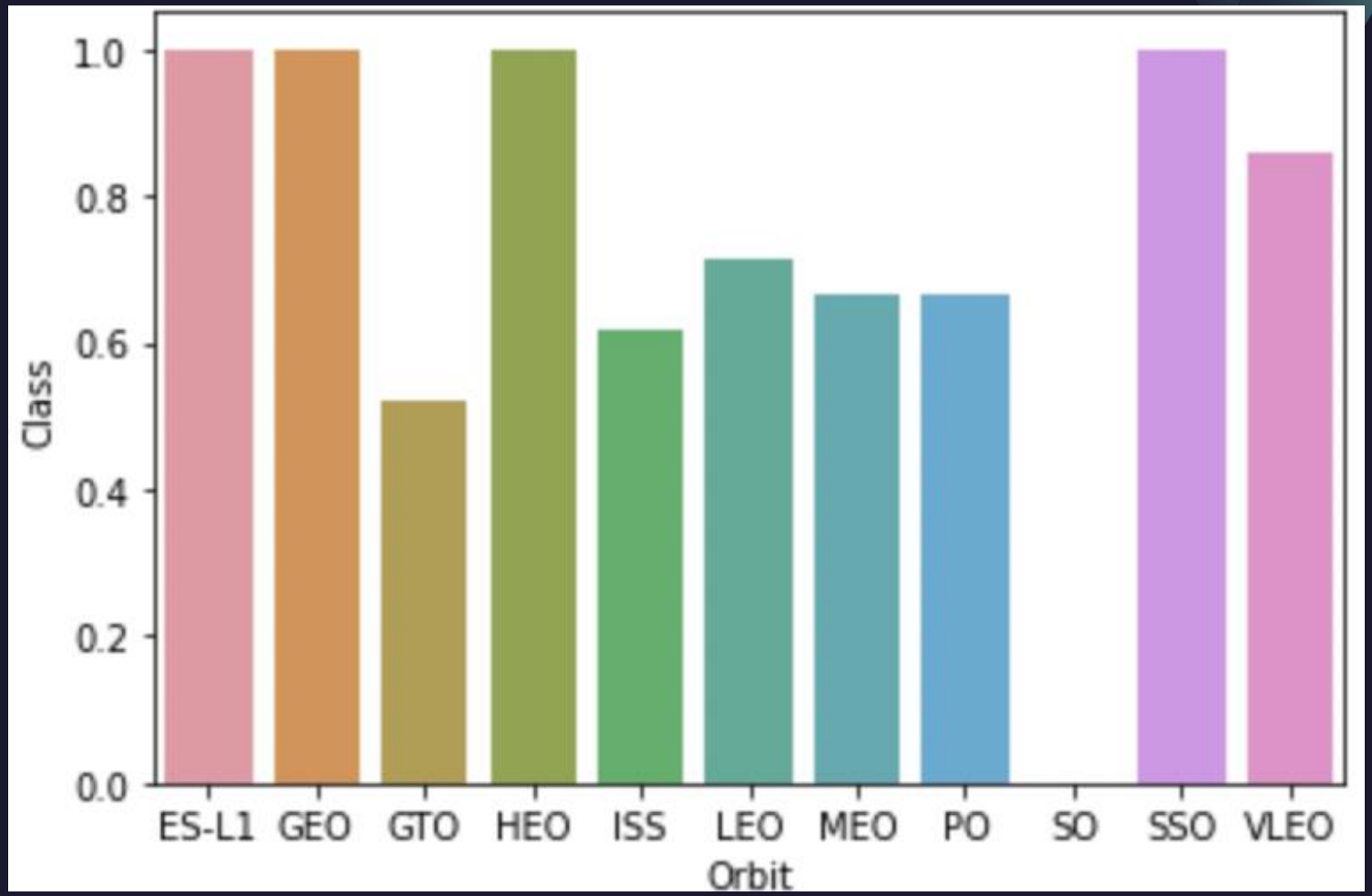


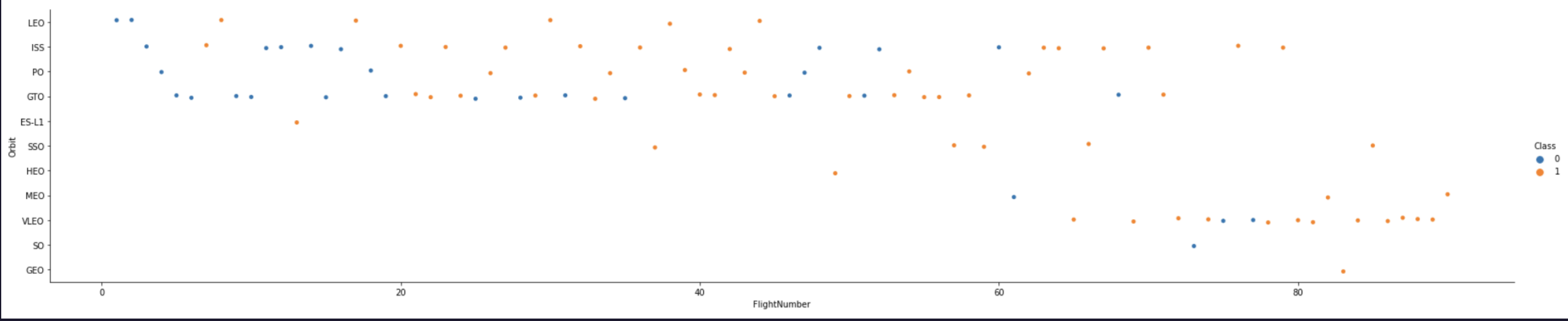
Payload vs Launch Site

Most of the launches, regardless of site, occur for payload masses less than 8000 kg. Additionally, most launches occur at site CCAFS SLC 40. A mix of successful and unsuccessful launches for payloads less than 8000 kg occur, but for those above 8000 kg, the majority of launches are unsuccessful.

Success Rate vs Orbit Type

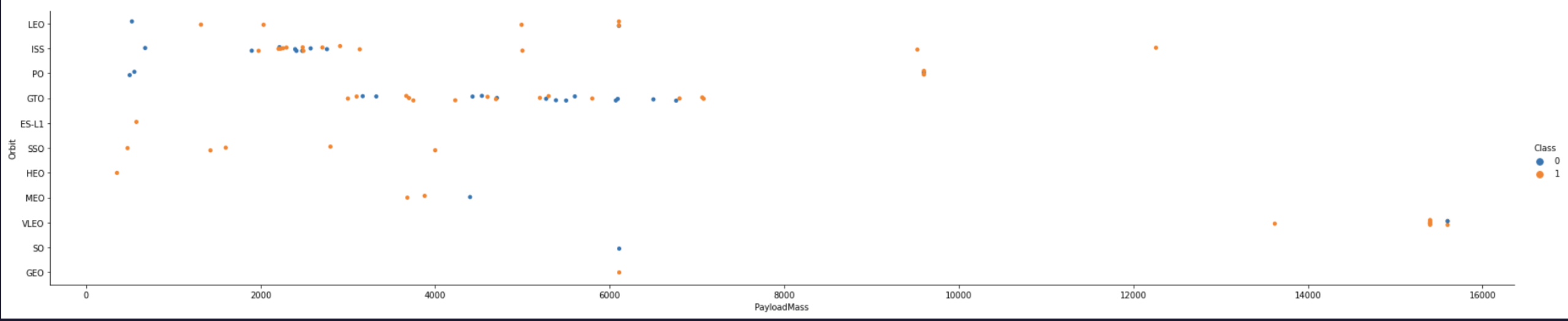
Orbits ES-L1, GEO, HEO, and SSO have perfect success rates while others hover around 50% to 60%, and SO has zero success.





Flight Number vs Orbit Type

in the LEO orbit, successful landings appear to be related to the number of flights. On the other hand, there seems to be no relationship between flight number and successful landings in the GTO orbit.

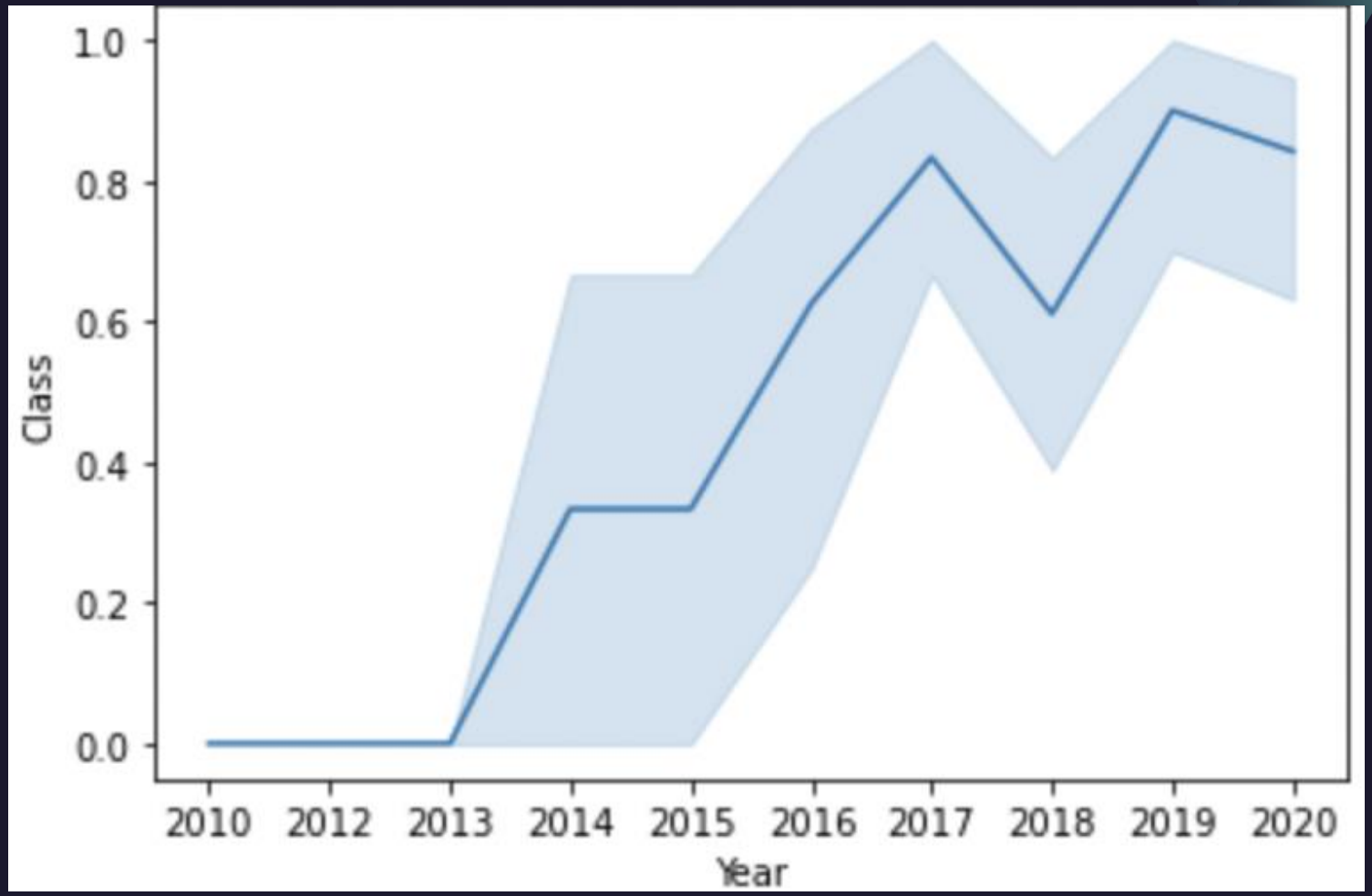


Payload vs Orbit Type

With heavy payloads, successful landings, or positive landing rate, increase with Polar, LEO, and ISS. For GTO, however, the number of successful and unsuccessful landings is similar and hard to distinguish which is more likely for that orbit.

Launch Success Yearly Trend

The success rate increases from 2013 to 2020.




Insights
Drawn from
SQL Queries

All Launch Site Names

SQL QUERY

- `SELECT DISTINCT LAUNCH_SITE FROM SPACEXTBL;`

EXPLANATION

- Only unique values for launch sites in the SpaceX table are displayed.

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

Launch Site Names Begin with 'CCA'

DATE	time_utc	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__outcomes
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-18	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
2012-10-08	00:35:00	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 SPACEXTBL WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5;`

EXPLANATION

- The asterisk displays all records in the table, and the percent wildcard character allows a search for launch sites beginning with CCA.

Total Payload Mass

SQL QUERY

- ```
  SELECT SUM(PAYLOAD_MASS__KG_)
  AS TOTAL_PAYLOAD_MASS FROM
  SPACEXTBL WHERE CUSTOMER =
  'NASA (CRS)';
  ```

EXPLANATION

- The SUM function is used to calculate the total payload mass by summing each entry in the payload_mass__kg_ column and a WHERE clause is used to specify that the customer is NASA (CRS).

Total Payload Mass
45596
  ```

Average Payload Mass by F9 v1.1

SQL QUERY

- ```
  SELECT AVG(PAYLOAD_MASS__KG_)
  AS AVERAGE_PAYLOAD_MASS FROM
  SPACEXTBL WHERE
  BOOSTER_VERSION = 'F9 v1.1';
  ```

Average Payload Mass
2928

EXPLANATION

- The AVG function is used to calculate the average payload mass, and a WHERE clause is used to specify that the booster version is F9 v1.1.
  ```

First Successful Ground Landing Date

SQL QUERY

- `SELECT MIN(DATE) AS MIN_DATE FROM SPACEXTBL WHERE LANDING__OUTCOME = 'Success (ground pad)';`

Date
2015-12-22

EXPLANATION

- The MIN funtion retrieves the minimum value from the Date column, and the WHERE clause specifies a successful ground landing.

Successful Drone Ship Landing with Payload between 4000 kg and 6000 kg

SQL QUERY

- SELECT BOOSTER_VERSION FROM SPACEXTBL WHERE LANDING__OUTCOME = 'Success (drone ship)' AND (PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000);

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

EXPLANATION

- A successful landing outcome from a drone ship is specified in the where clause, along with a BETWEEN clause to specify ranges for the payload mass.

Total Number of Success and Fail Mission Outcomes

SQL QUERY

```
SELECT COUNT(MISSION_OUTCOME)
AS "Total Outcomes" FROM
SPACEXTBL WHERE
(MISSION_OUTCOME LIKE
'Success%' OR MISSION_OUTCOME
LIKE 'Failure%');
```

EXPLANATION

- A count of the mission outcomes for specific instances is retrieved by using the LIKE clause to specify whether the outcome was a success or failure.

Total Outcomes
101

Boosters Carried Maximum Payload

SQL QUERY

- `SELECT DISTINCT BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL);`

EXPLANATION

- The distinct booster versions are selected from the SpaceX table where the payload mass is equal to a subquery that selects the maximum payload mass from the table.

booster_version
F9 B5 B1048.4
F9 B5 B1048.5
F9 B5 B1049.4
F9 B5 B1049.5
F9 B5 B1049.7
F9 B5 B1051.3
F9 B5 B1051.4
F9 B5 B1051.6
F9 B5 B1056.4
F9 B5 B1058.3
F9 B5 B1060.2
F9 B5 B1060.3

2015 Launch Records

SQL QUERY

- SELECT LANDING__OUTCOME, BOOSTER_VERSION, LAUNCH_SITE FROM SPACEXTBL WHERE LANDING__OUTCOME = 'Failure (drone ship)' AND DATE LIKE '2015%';

EXPLANATION

- Landing outome, booster version, and launch site are selected from the SpaceX table for failed landing outcomes in the year 2015.

landing_outcome	booster_version	launch_site
Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

Rank Landing Outcomes Between 2010-06-4 and 2017-03-20

SQL QUERY

- SELECT LANDING__OUTCOME, COUNT(LANDING__OUTCOME) AS "Count" FROM SPACEXTBL WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY LANDING__OUTCOME ORDER BY COUNT(LANDING__OUTCOME) DESC;

EXPLANATION

- A count of various landing outcomes is retrieved from the SpaceX table for the specified dates listed above and are ranked by grouping the counts for each outcome and sorting the counts in descending order.

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

Launch Sites Proximities Analysis

Launch Sites Marked Globally

The four SpaceX launch sites located in California and Florida are shown on a global map.

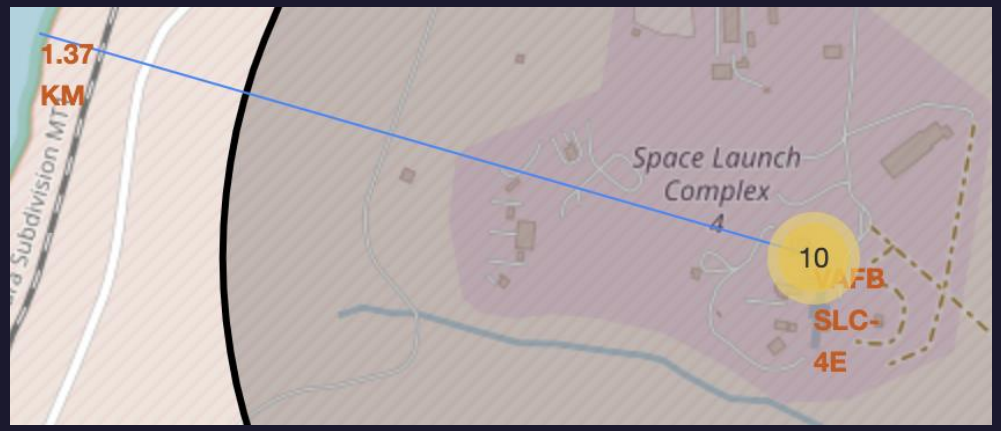
Launch Site Outcomes

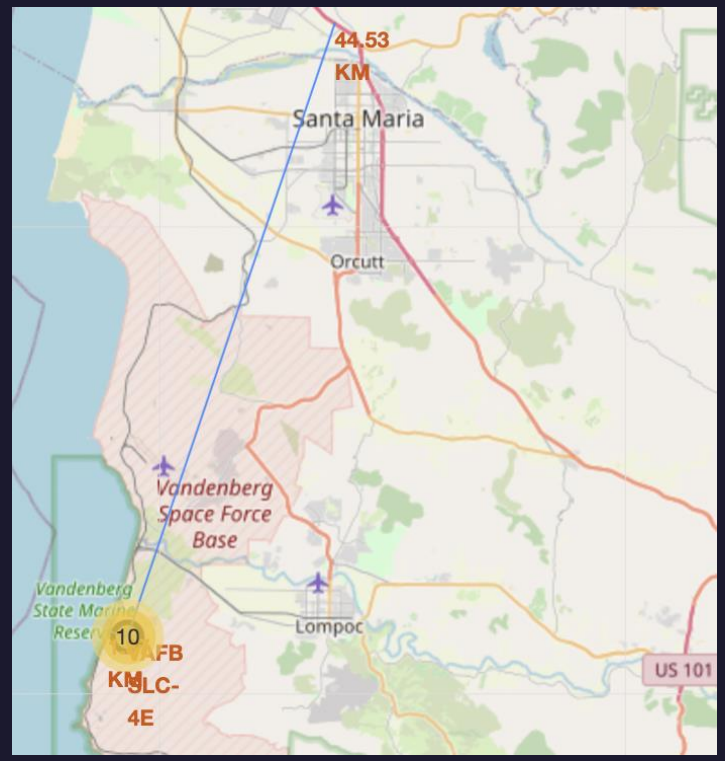
- Shown to the right are color-coded launch outcomes in green and red for the three sites in located in Florida (top left, top right, bottom left) and the site located in Calfornia (bottom right).

- **Green** markers indicate successful launches and **red** markers indicate failed failed launches.

Distances to Major Landmarks

Text shown in red indicates the distance in km from the base to the nearest marker, along with a blue line to mark the straight line distance. Distances are shown from the SpaceX launch site in CA to the nearest coastline (below) and to the nearest city, Santa Barbara (right).

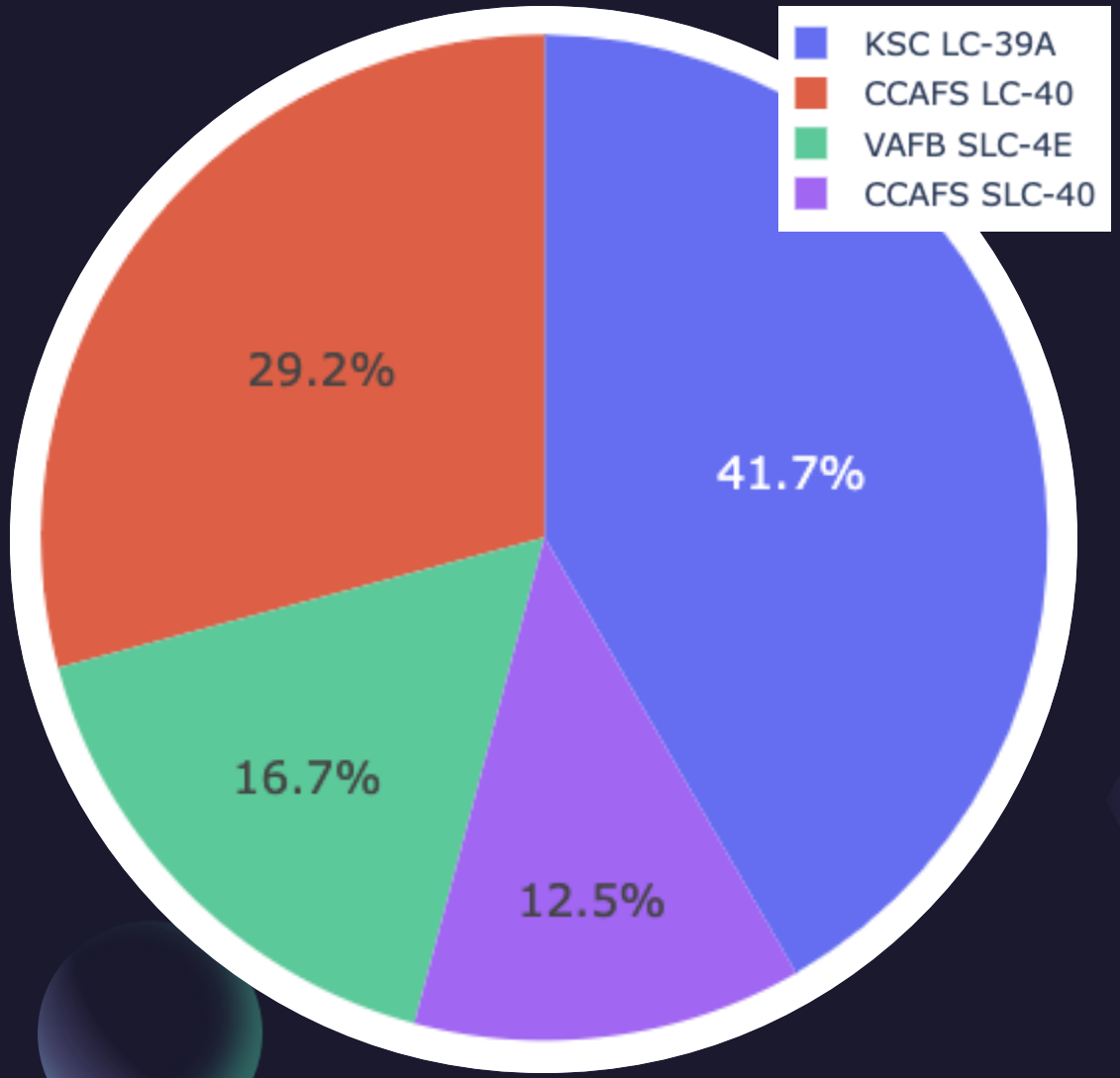




Build a Dashboard with Plotly Dash

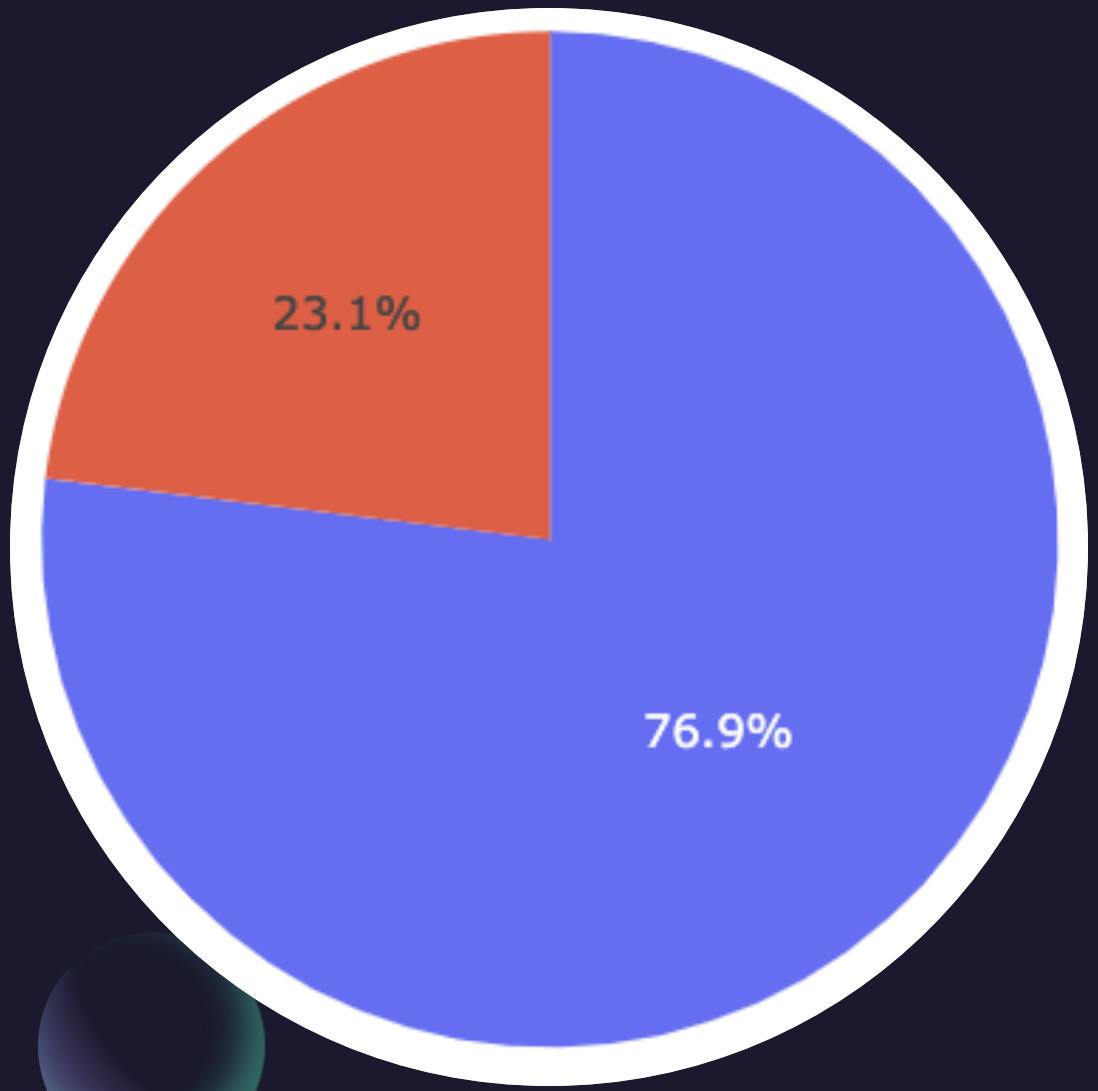
Launch Site Success

Success rates for each of the four launch sites are presented in a pie chart to the right. KSC LC-39A (blue) was the most successful.



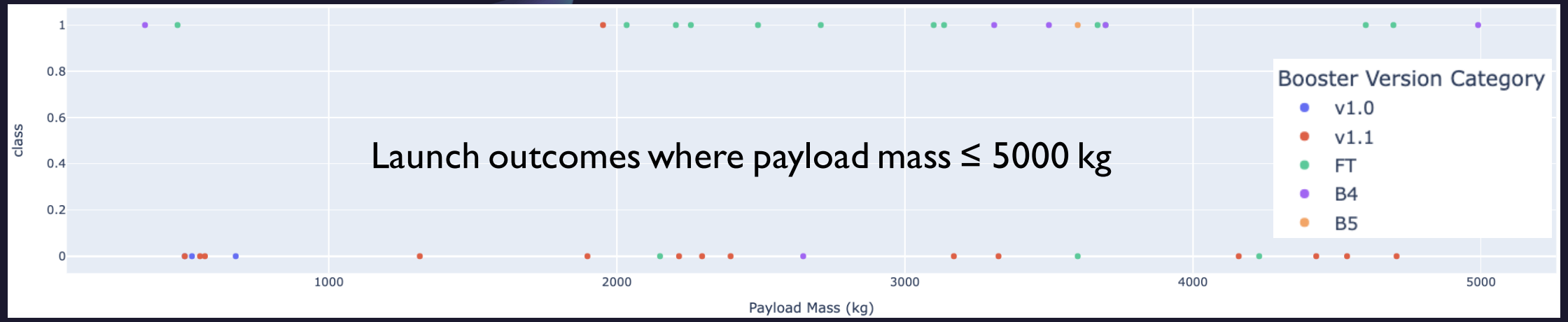
Site with Highest Launch Success

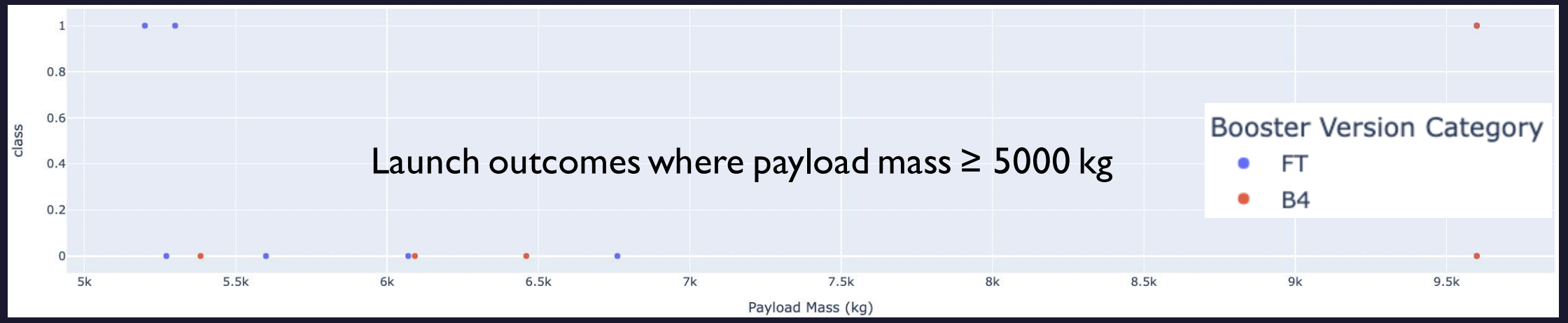
The KSC LC-39A site has the highest success rate at 76.9% while just 23.1% of launches resulted in failure.



Payload vs Launch Outcome

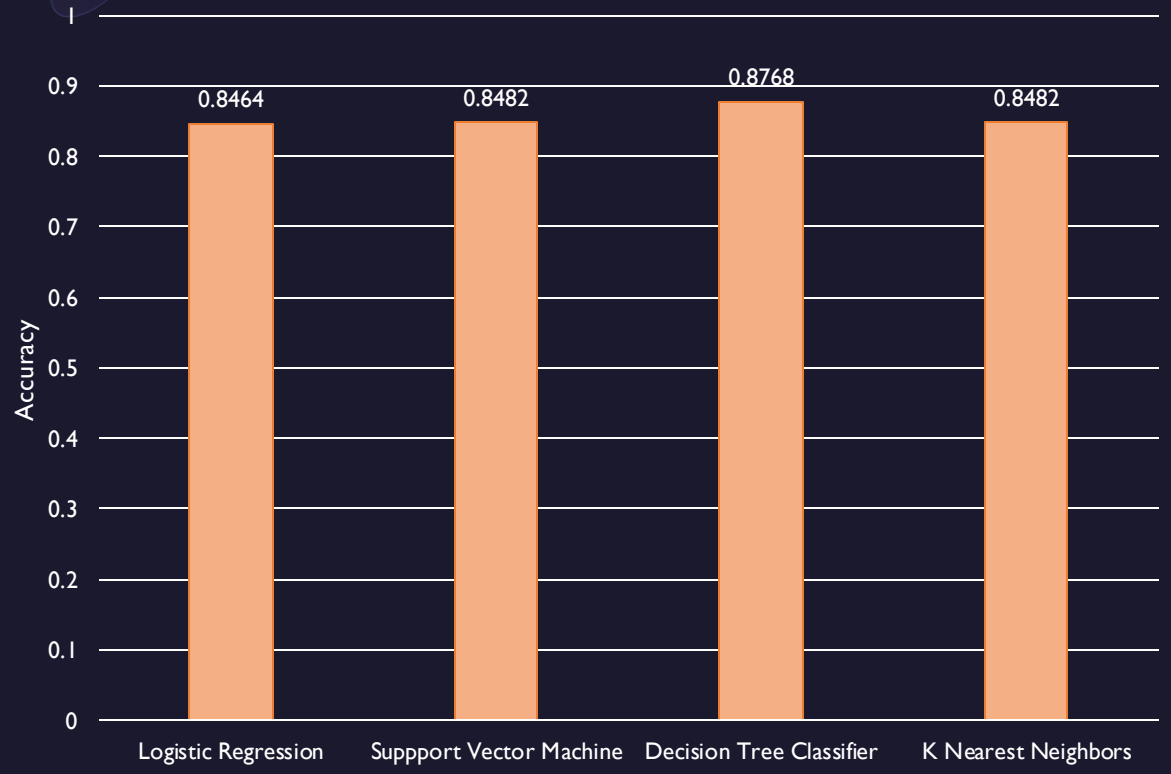
Scatter plots from the dashboard app show launch outcomes for various booster versions. It is clear from the two plots that launch outcomes are less successful for heavier payloads. Additionally, the FT booster appears to have successful outcomes for most launches.



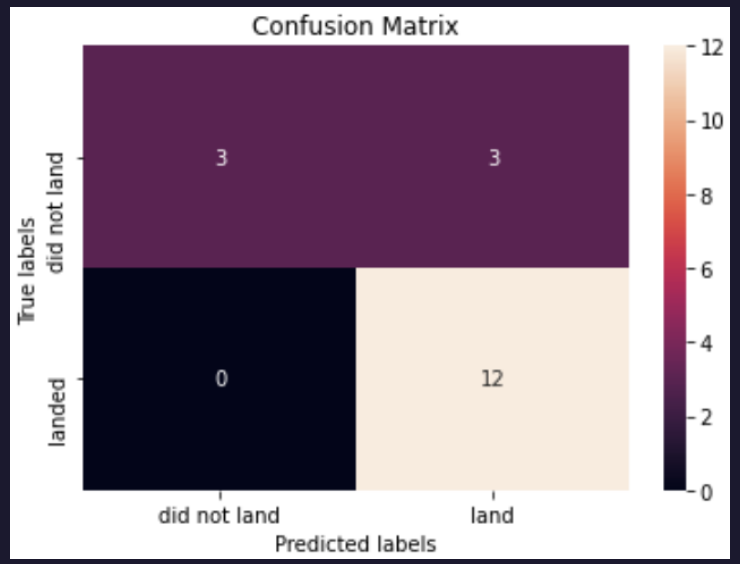



Predictive
Analysis
(Classification)

Classification Accuracy



The decision tree classifer performs best with just over 87% accuracy on the validation data. All models perform similarly on the test set.



PREDICTED VALUES

PREDICTED VALUES		
	TRUE POSITIVE	FALSE NEGATIVE
	FALSE POSITIVE	TRUE NEGATIVE

Confusion Matrix

The confusion matrix to the left shows the accuracy, precision, and recall of a decision tree classifier, which performed the best on the data out of the models tested. The matrix shows the number of correctly predicted values and incorrectly predicted values for the Class column, which are related to a landing being successful or not.

Conclusions

- Overall, the success rate has increased over time.
- Heavier payloads are less successful.
- KS LC-39A has the most successful launches of any site.
- The Decision Tree Classifier model performs best on the data.


Appendix

67%

Average launch success rate

Mission Outcomes

Mission Outcome	Count
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

Thank You