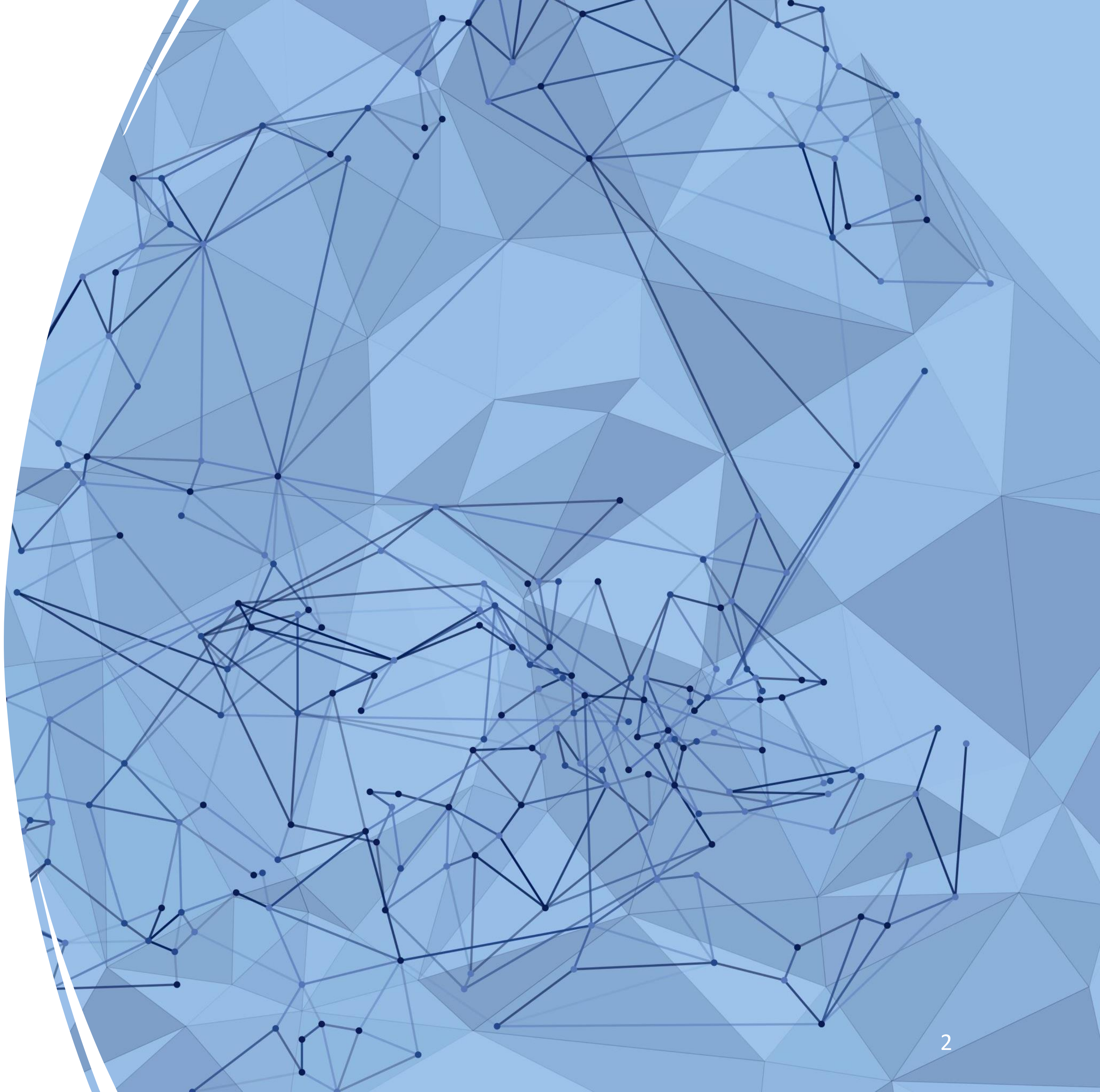
IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Matthew H Thomas
March 10, 2024

Outline

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



Executive Summary – Methodology

Data collection methodology

- Collected using SPACEX API and web-scraping.

Perform data wrangling

- Used APIs, filtering, and data mean to prepare data.

Perform exploratory data analysis (EDA) using visualization and SQL

Perform interactive visual analytics using Folium and Plotly Dash

Perform predictive analysis using classification models

- Used grid search to tune parameters and confusion matrices to evaluate models.

Executive Summary - Results

In exploratory analysis, we found:

- The success rate has improved for the most recent flight numbers for all orbits.
- Year after year, the success rate of the launches has shown a steady improvement.

Using interactive analysis, we found:

- Almost 50% of the successful launches occurred at one site, the *CCAFS LC-40* site.
- However, upon further analysis, we found only 26.9% of its launches were successful.

The best performing predictive model was the Decision Tree model, which achieved an accuracy score of 88.9%.

Introduction

- In this capstone, we took the role of a data scientist working for a new rocket company, Space Y.
- Our job was to determine the price of each launch.
- We did this by gathering information about SpaceX and creating dashboards for our team.
- We also determined if SpaceX would be able to reuse the first stage of their rockets.
- Instead of using rocket science to determine if the first stage will land successfully, we trained a machine learning model and used public information to make predictions about their successful launches.

Section 1

Methodology

Methodology

Executive Summary	
Data collection methodology	• Collected using SPACEX API and web-scraping.
Perform data wrangling	• Used APIs, filtering, and data mean to prepare data.
Perform exploratory data analysis (EDA) using visualization and SQL	
Perform interactive visual analytics using Folium and Plotly Dash	
Perform predictive analysis using classification models	• Used grid search to tune parameters and confusion matrices to evaluate models.

Data Collection

For this capstone, we gathered most of the SpaceX launch data using the SpaceX REST API.

Using this API, we were able to download launch data including the rocket used, payload delivered, launch specifications, landing specifications, and landing outcome.

For additional launch data, we web scraped the Wiki pages related to the launches.

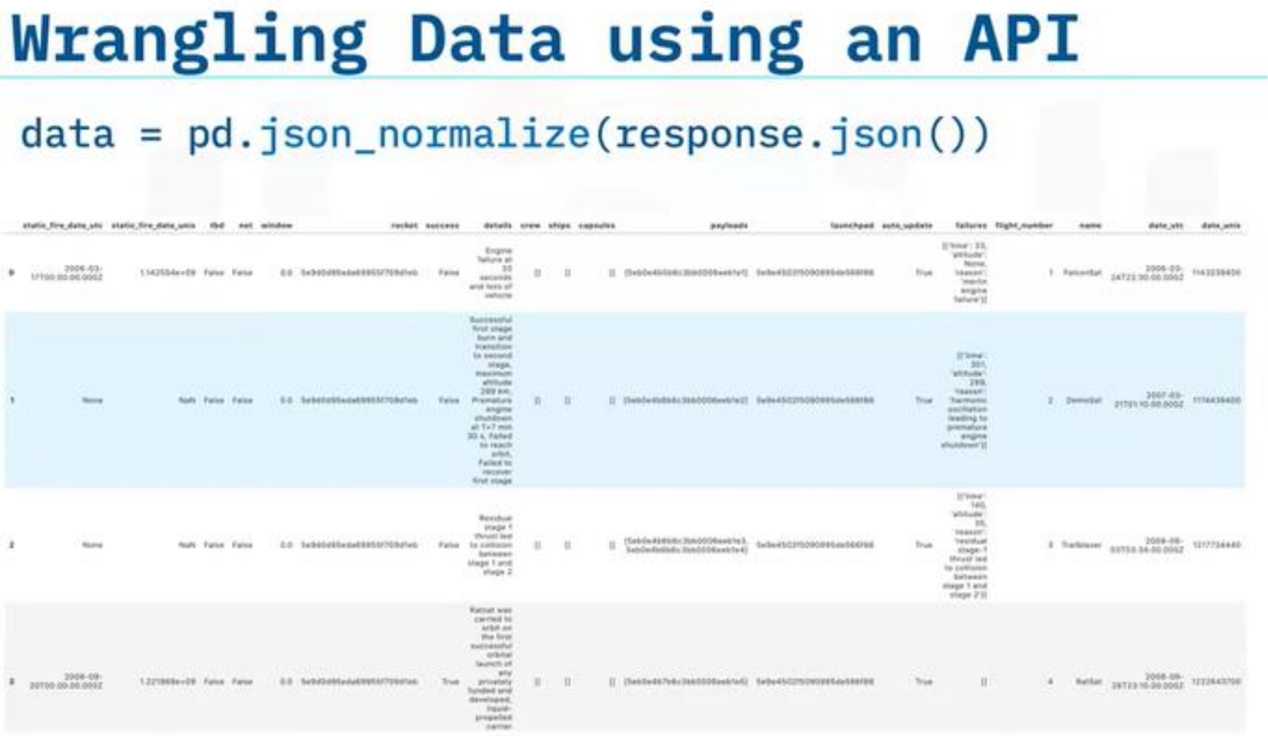
From these web pages, we obtained other launch data like the date and time of the launches, launch site, payload masses, target orbit, the customer and launch outcome.

Data Collection – SpaceX API

The SpaceX REST API includes data about the rocket used, payload delivered, launch specifications, landing specifications, and landing outcome.

The launch data is located at the URL: **api.spacexdata.com/v4/launches/past**.

Because the data received via the API was in the form of a JSON object, we used the **pandas** *json_normalize* function to convert it into a data frame for exploratory analysis.



[Link to notebook on Github]

Data Collection – Web Scraping

To web scrape the Wiki HTML pages related to the launches, we used the Python **BeautifulSoup** package.
The downloaded data was formatted as tables.
We then parsed the data from those tables and converted them into a **pandas** data frame for further visualization and analysis.

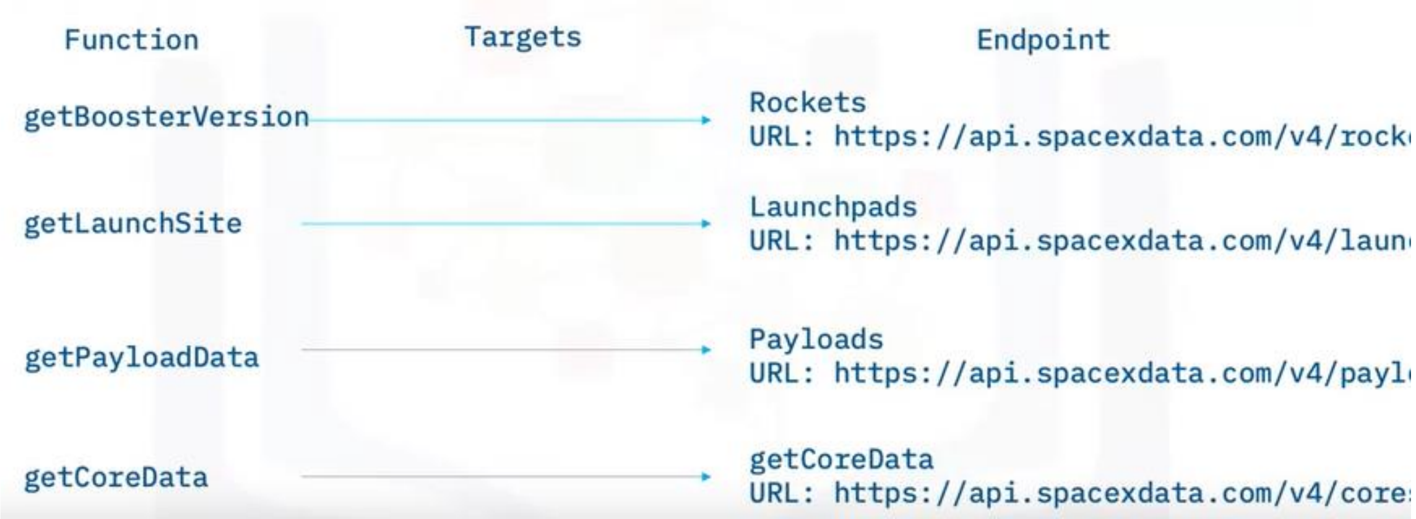
Web scraping Falcon 9 Launch records

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	Longitude	Latitude
0	1	2006-03-24	Falcon 1	20.0	LEO	Kwajalein Atoll	None None	1	False	False	False	None	NaN	0	Merlin1A	167.743129	9.047721
1	2	2007-03-21	Falcon 1	NaN	LEO	Kwajalein Atoll	None None	1	False	False	False	None	NaN	0	Merlin2A	167.743129	9.047721
2	4	2008-09-28	Falcon 1	165.0	LEO	Kwajalein Atoll	None None	1	False	False	False	None	NaN	0	Merlin2C	167.743129	9.047721
3	5	2009-07-13	Falcon 1	200.0	LEO	Kwajalein Atoll	None None	1	False	False	False	None	NaN	0	Merlin3C	167.743129	9.047721
4	6	2010-06-04	Falcon 9	NaN	LEO	CCAFS SLC 40	None None	1	False	False	False	None	1.0	0	B0003	-80.577366	28.561857

[Link to notebook on Github]()

Data Wrangling

- To transform this raw data into a clean dataset, we must address three issues:
  - Replacing id numbers with data.
  - Filtering Falcon 1 launch data.
  - Dealing with null values.
- For some the columns, there is just an identification number, not actual data.
  - This means we will need to use the API again to target other endpoint data.
  - These endpoints include information about the booster, launchpad, payload, and core.



Data Wrangling

The raw data includes launch data about both Falcon 1 and Falcon 9 boosters.

Since we only want data about the Falcon 9 booster, we filtered the data to remove the Falcon 1 launches.

Finally, some of the data contains NULL values.

For the NULLS in payload mass data, we replaced missing values with the mean of the payload masses.

In the column with the landing pad data, the NULL values were retained, because they represent when a landing pad is not used.

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused
0	1	2006-03-24	Falcon 1	20.0	LEO	Kwajalein Atoll	None None	1	False	False
1	2	2007-03-21	Falcon 1	NaN	LEO	Kwajalein Atoll	None None	1	False	False
2	4	2008-09-28	Falcon 1	165.0	LEO	Kwajalein Atoll	None None	1	False	False
3	5	2009-07-13	Falcon 1	200.0	LEO	Kwajalein Atoll	None None	1	False	False
4	6	2010-06-04	Falcon 9	NaN	LEO	CCAFS SLC 40	None None	1	False	False

```
data_falcon9.isnull().sum()

FlightNumber       0
Date               0
BoosterVersion     0
PayloadMass        5
Orbit              0
LaunchSite         0
Outcome            0
Flights            0
GridFins           0
Reused             0
Legs               0
LandingPad        26
Block              0
ReusedCount        0
Serial             0
Longitude          0
Latitude           0
dtype: int64
```

```
data_falcon9.isnull().sum()

FlightNumber       0
Date               0
BoosterVersion     0
PayloadMass        0
Orbit              0
LaunchSite         0
Outcome            0
Flights            0
GridFins           0
Reused             0
Legs               0
LandingPad        26
Block              0
ReusedCount        0
Serial             0
Longitude          0
Latitude           0
dtype: int64
```


Link to notebook on Github

EDA with Data Visualization

In the exploratory data analysis, we used three visualization types to identify success rate patterns in the data.

In two scatter plots, we examined the relationship between flight numbers and launch sites as well as to payload orbits, to identify any success rate patterns related to flight number.

In other scatter plots, we also analyzed the relationship between payload mass and launch sites and also to orbits, looking for success patterns related to payload mass.

To determine whether the orbit affects the success rate, we compared the success rates of orbits in a bar chart.

And finally, we used a line plot to identify any patterns in the success rate over time.

[Link to notebook on Github]

EDA with SQL

SQL Commands Used:

- CREATE TABLE
  - Table Creation
- SELECT
  - Selecting table columns
- WHERE
  - Conditionals
- LIKE
  - Contains specific sub-strings
- GROUP BY
  - Divide into groups
- ORDER BY
  - Sort rows
- AVG
  - Find average value
- SUM
  - Add values together
- COUNT
  - Count rows
- LIMIT
  - Restrict number of rows

Link to notebook on Github

Build an Interactive Map with Folium

Map Objects Used:

- **Marker**
  - Used to mark and label each launch site.
- **Circle**
  - Used to fill markers with color and pop-up identifying each site.

[Link to notebook on Github]

- **MarkerCluster**
  - Used to group Markers identifying successful and failed launches at each site.
- **PolyLine**
  - Used to mark proximity of railroads, highways, coast, and cities to sites.
- **MousePosition**
  - Used to get latitude and longitude of proximity objects.

Build a Dashboard with Plotly Dash

The dashboard is divided into two main areas: the pie chart and the scatter plot.

For the pie chart, the user can select all the sites, or just one of the launch sites.

- If all sites are selected, the pie chart compares the number of successful launches at all sites as a percentage.
- If only one is selected, the pie chart will display the percentage of successful and failed launches at that site.

Given the launch site selection above, the scatter plot will display each of the successful and failed launches for each type of booster rocket at a given payload mass range.

- The scrollbar allows you to change the range of payloads.

[Link to Python file on Github]()

Predictive Analysis (Classification)

Link to notebook on Github

For the predictive analysis, we created four models with the data: Logistic Regression, SVM, Decision Tree, and k-Nearest Neighbor models.

The first strategy to improve results was to normalize the attributes used in the models.

To ensure my results overfit the data, we then divided the attributes and target data into training and test data with an 80/20 split.

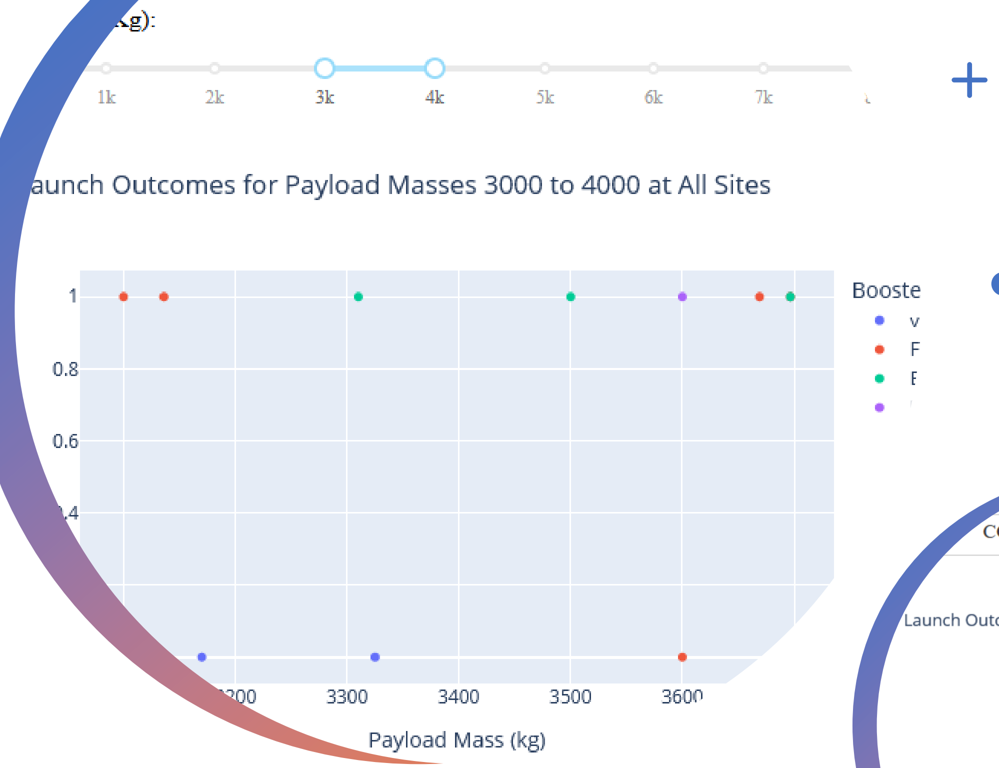
For each model, we used grid search to identify the best hyper-parameters for the models with the training data.

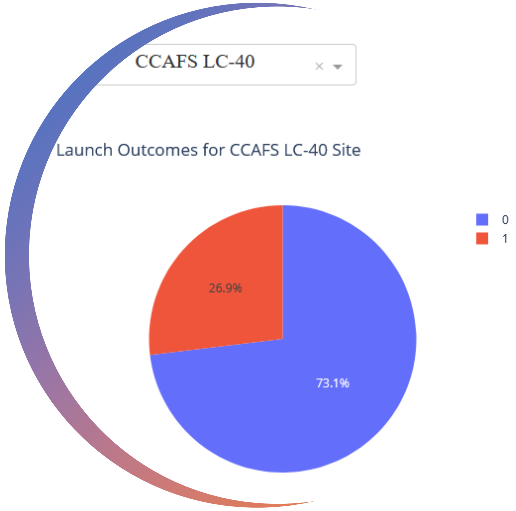
Having identified the best estimator for each model, we then found the accuracy of the models on the test data.

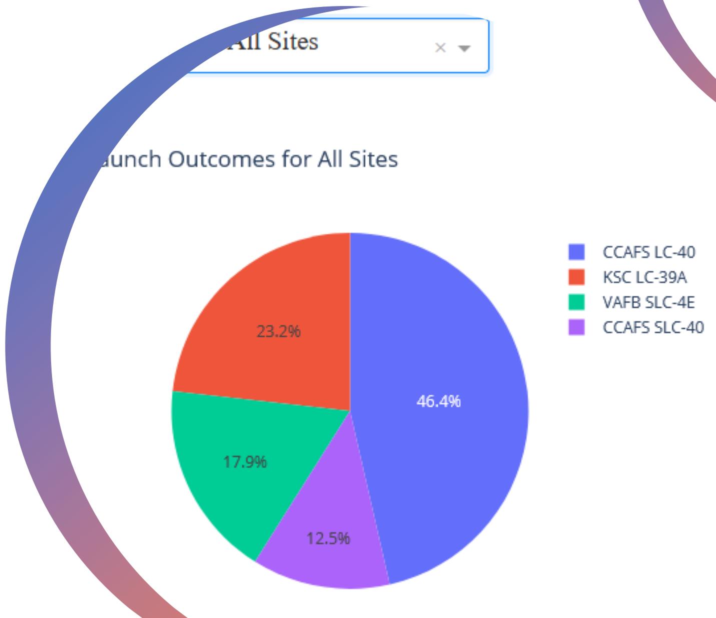
We also reviewed the Confusion Matrices for each model to evaluate them.

Results – Exploratory Analysis

- The *VAFB SLC 4E* and *KSC LC 39A* sites had the best success rate.
- The *KSC LC 39A* site has the best success rate for all payload sizes.
- The *ES-L1*, *GEO*, *HEO*, and *SSO* orbits all have 100% success rates.
- The success rate has improved for the most recent flight numbers for all orbits.
- For the heavy payloads, the *Polar*, *LEO* and *ISS* orbits were the most successful orbits.
- The *ES-L1*, *SSO*, *HEO*, and *MEO* orbits were most successful for the smaller payloads.
- Year after year, the success rate of the launches has shown a steady improvement.







Results – Interactive Analytics

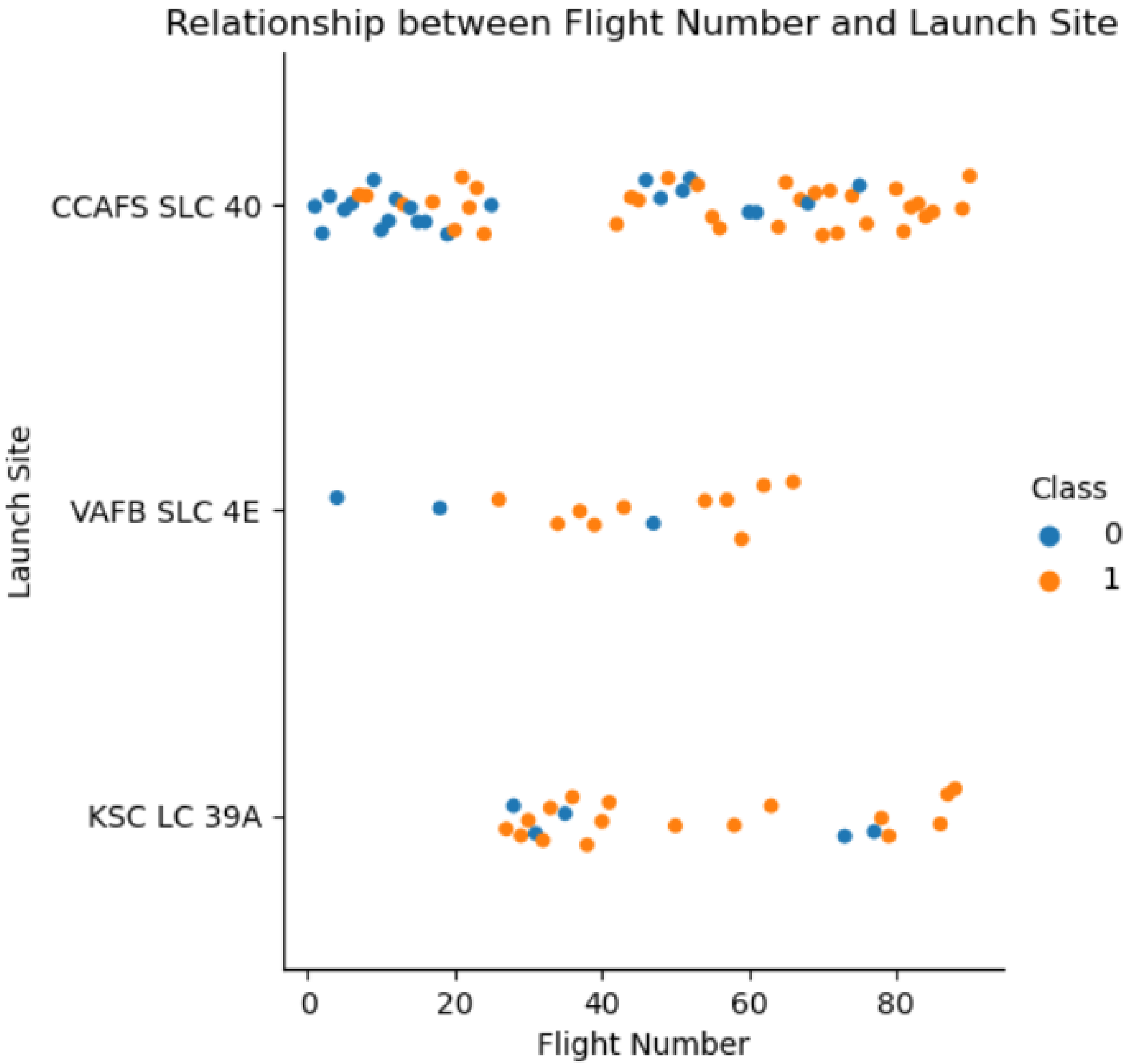
- Using interactive analytics, we found almost 50% of the successful launches occurred at one site, the *CCAFS LC-40* site.
- However, upon further analysis, we found only 26.9% of its launches were successful.
- So, to achieve a low success rate but still have the highest number of successes, it must have also had a high number of total launches.
- In the payload range of *3,000* to *4,000* kg, there were more than twice as many successful launches as failures.
- With this payload mass range, the most successes were evenly divided between the *FT* and *B4* booster versions.

Results – Predictive Analysis

- The best performing predictive model was the Decision Tree model.
- Its accuracy score was **88.9%**, more than 5% better than all of the other models.
- On the test data, it was also the only model with one false positive and one false negative.
- All of the other models had at least five false positives.

Section 2

Insights drawn from EDA

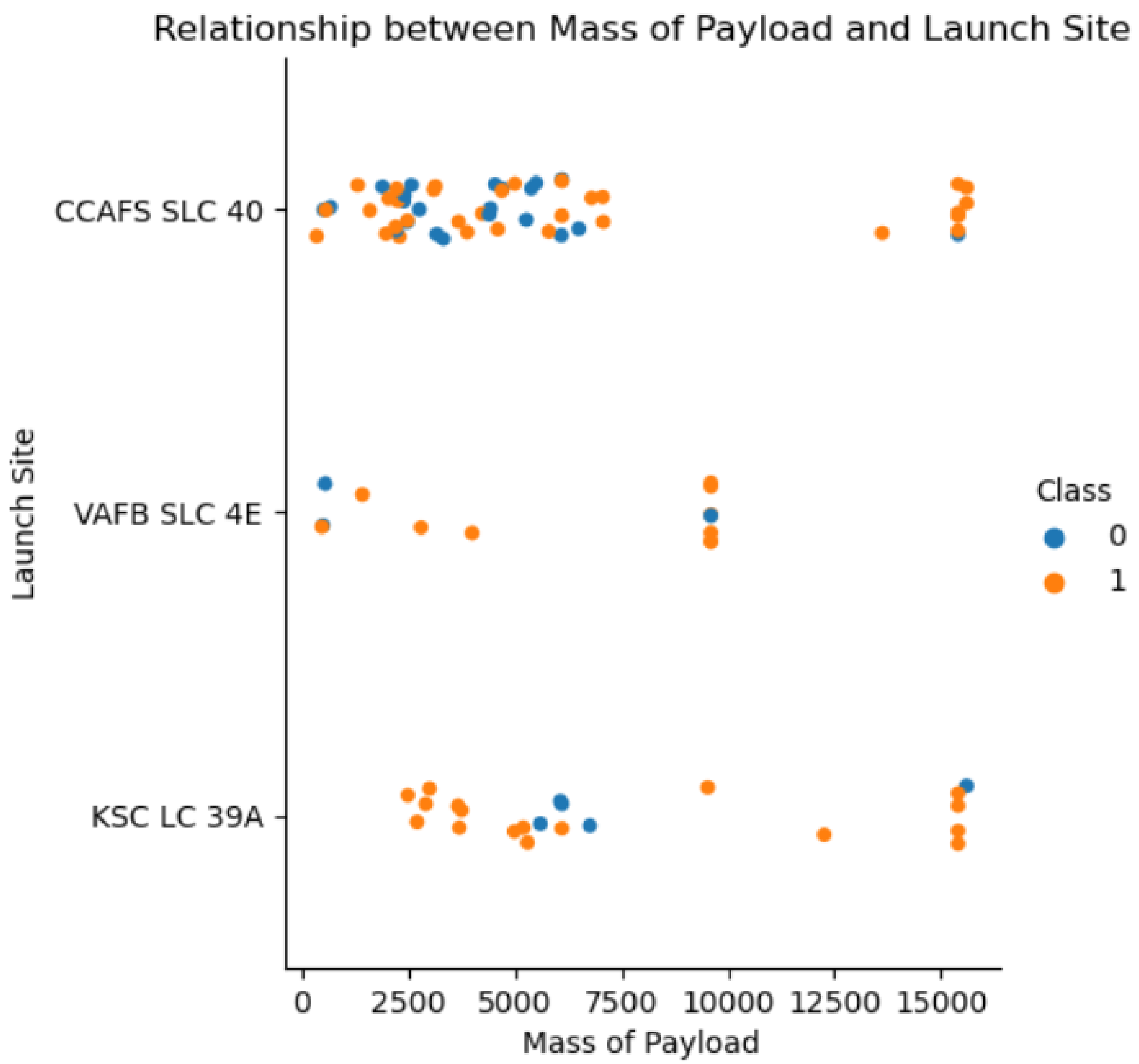


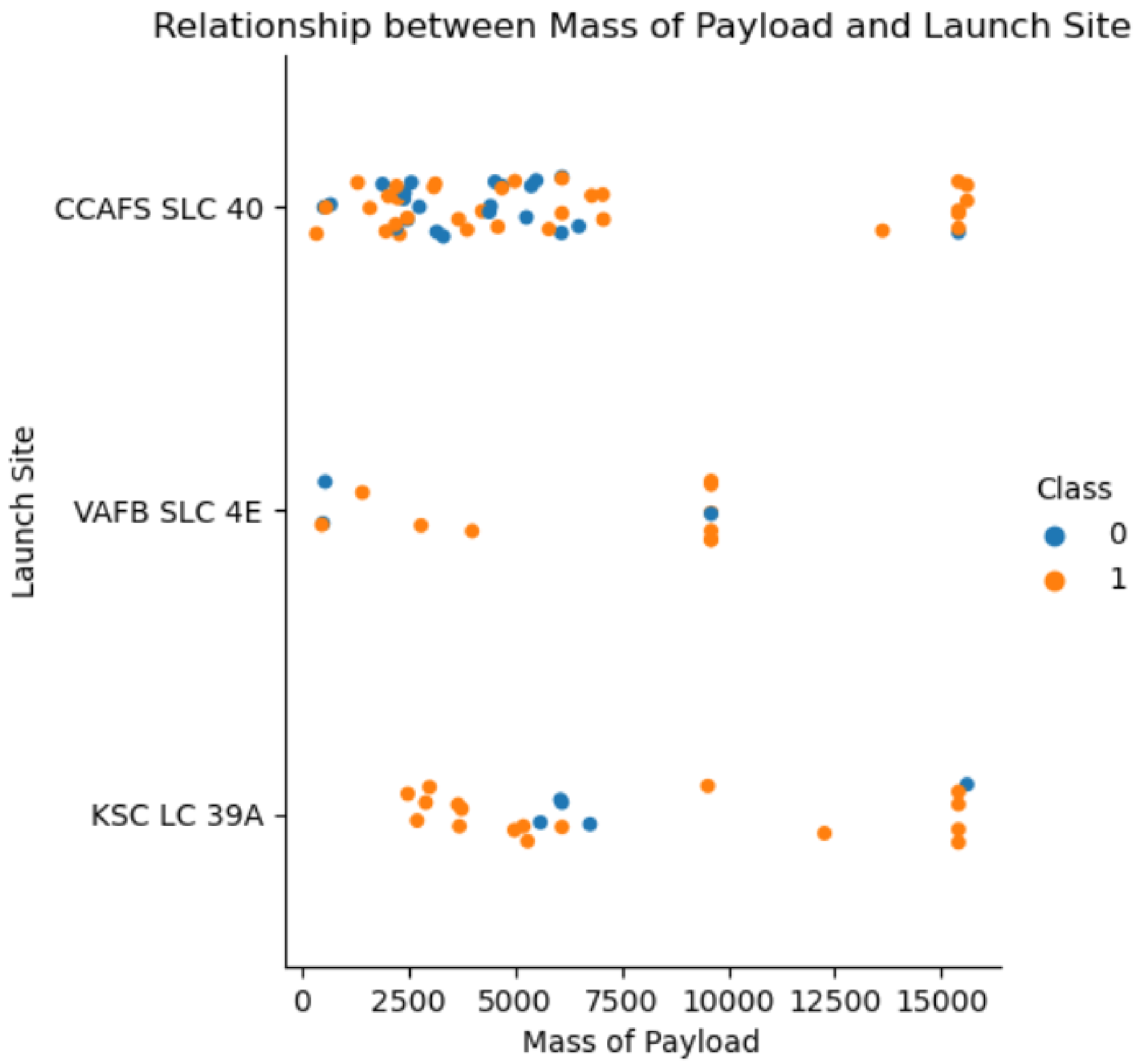
Flight Number vs. Launch Site

- Overall, the *VAFB SLC 4E* and *KSC LC 39A* sites had the best success rate.
- However, most of the launches have been from the *CCAFS SLC 40* site, and its success rate has recently improved.

Payload vs. Launch Site

- Overall, the *KSC LC 39A* site has the best success rate for all payload sizes.
- The *VAFB SLC 4E* site also did well, but it had the fewest launches. And none of its launches had a higher-payload.
- The *CCAFS SLC 40* site did well for the higher payloads but had mixed results for the smaller payloads.



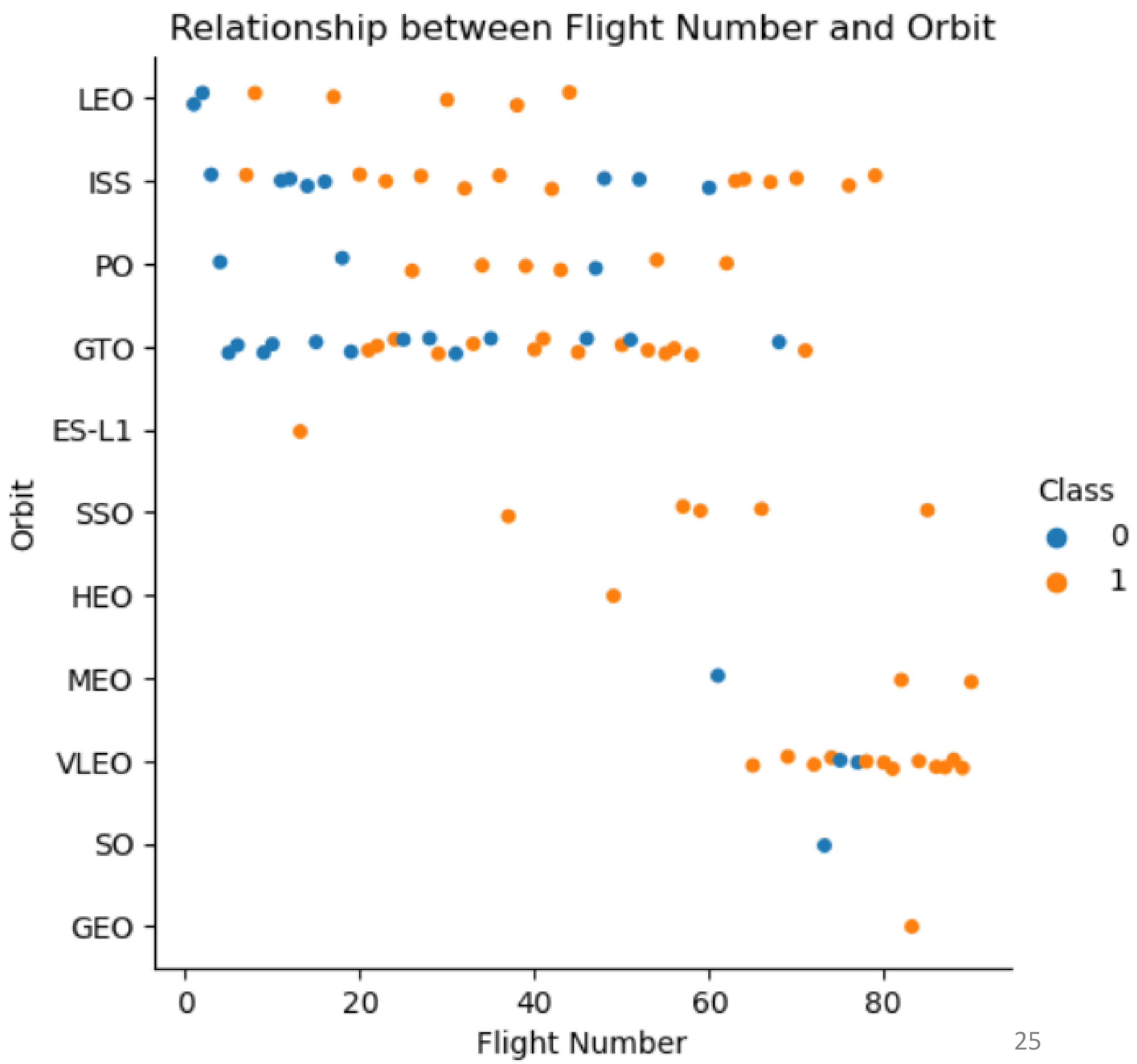


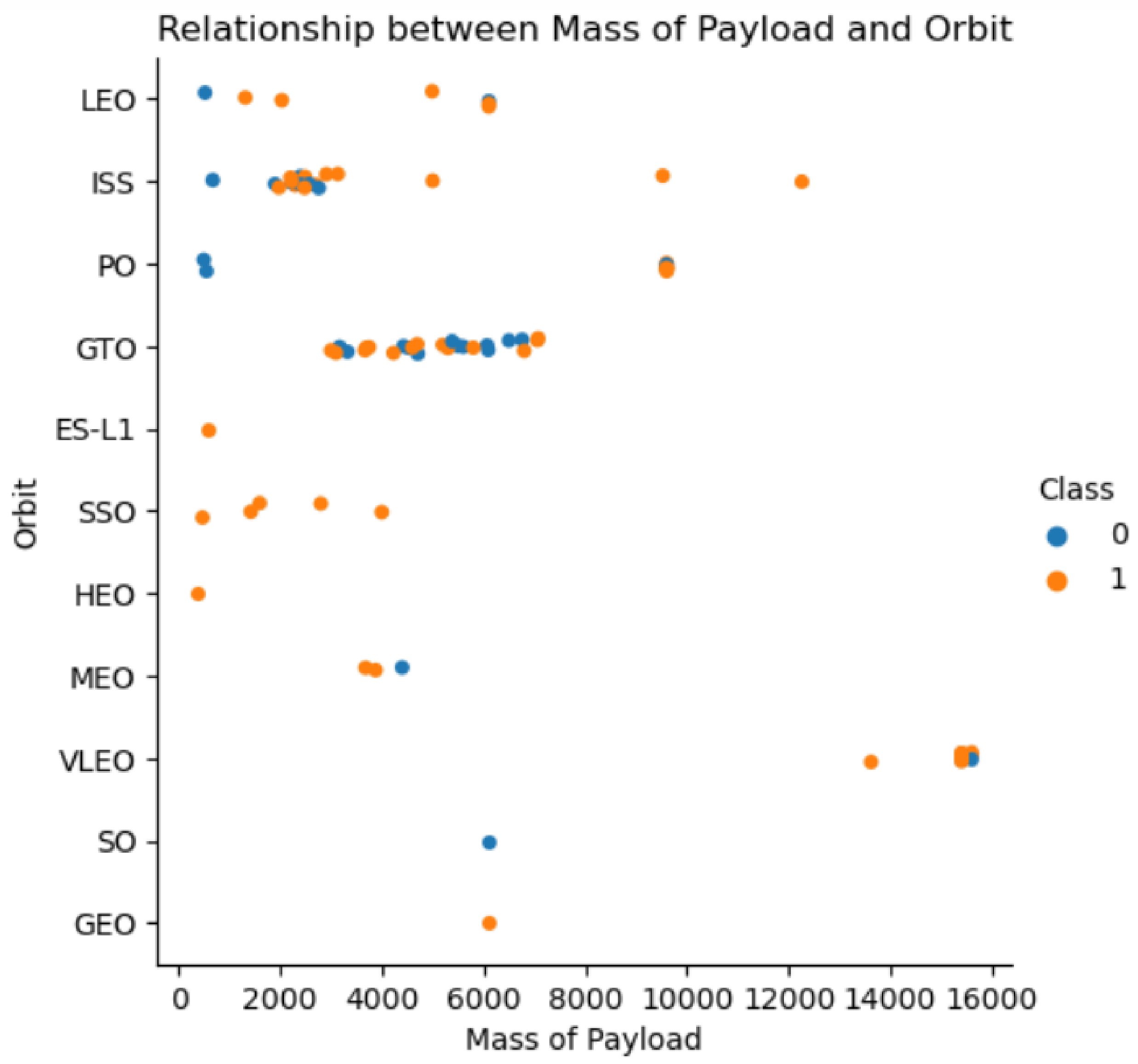
Success Rate vs. Orbit Type

- The *ES-L1*, *GEO*, *HEO*, and *SSO* orbits all have 100% success rates.
- The only orbit with no successes is the *SO* orbit.
- The rest of the orbits have success rates between 52% and 86%.

Flight Number vs. Orbit Type

- For the *LEO* orbit, the success rate improved as the number of flights increase.
- However, the success rate for the *GTO* orbit was mixed for all flight numbers.
- Overall, the success rate has improved for the most recent flight numbers.

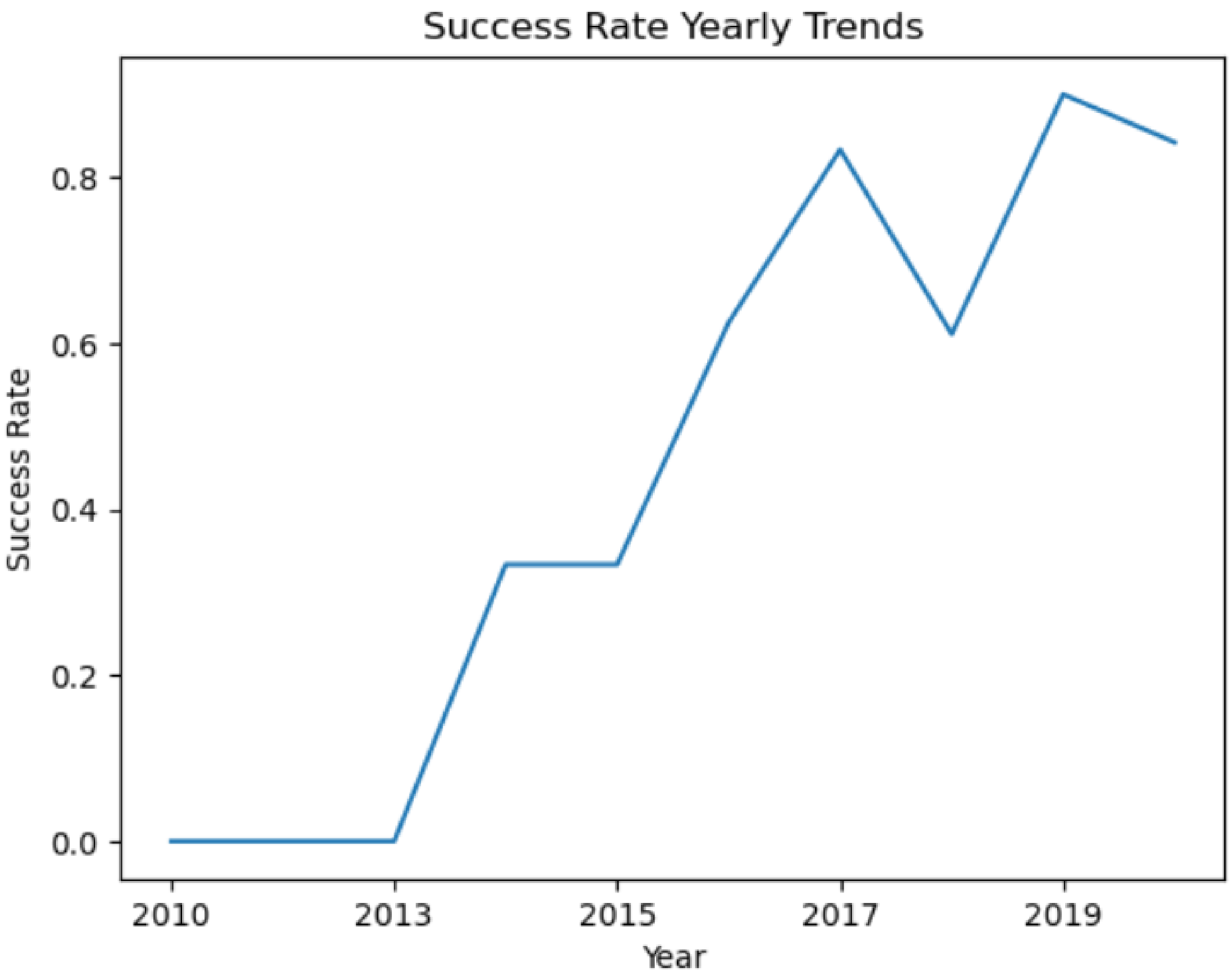




Payload vs. Orbit Type

- For the heavy payloads, the Polar, LEO and ISS orbits were the most successful orbits.
- The ES-L1, SSO, HEO, and MEO orbits were most successful for the smaller payloads.
- The GTO results had mixed results for the low to medium payloads.

Launch Success Yearly Trend



- In general, the success rate of the launches has shown a steady improvement.
- One notable exception was in 2018. The success rate dropped from 83% in 2017 to 61% that year.
- However, the success rate jumped up to 90% the next year.

All Launch Site Names

- There are four unique launch site names:
  - CCAFS LC-40
  - CCAFS SLC-40
  - KSC LC-39A
  - VAFB SLC-40
- The SQL **GROUP BY** command was used on the **Launch_Site** column to find the names.

```
%%sql SELECT Launch_Site AS 'Launch Site'
    FROM SPACEXTABLE
    GROUP BY Launch_Site
```

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

Launch Site
CCAFS LC-40
CCAFS SLC-40
KSC LC-39A
VAFB SLC-4E

Launch Site Names Begin with 'CCA'

```
%%sql SELECT * FROM SPACEXTABLE
    WHERE Launch_Site LIKE 'CCA%'
    LIMIT 5
```

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOA
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	

- The SQL **LIKE** command was used in the **WHERE** command on the string **'CCA%'** to find records for launch sites beginning with *'CCA'*.
- The **LIMIT** command was used to display the first five records meeting this condition.

Total Payload Mass

- The SQL **WHERE** command was used to find the records for the **Customer = 'NASA (CRS)'**
- To find the total payload of *45,596 kg*, the **SUM** command was used.

```
%%sql SELECT SUM(PAYLOAD_MASS__KG_) AS 'Total Payload for NASA (CRS)'
        FROM SPACEXTABLE
        WHERE Customer = 'NASA (CRS)'
```

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

Total Payload for NASA (CRS)
45596

Average Payload Mass by F9 v1.1

- The SQL **LIKE** command was used with the **WHERE** command to find launches using the *F9 v1.1* boosters.
- The **AVG** command was used to find the average payload mass of these launches *2534.67 kg*.

```
%%sql SELECT AVG(PAYLOAD_MASS__KG_) AS 'Average Payload for F9 v1.1 Booster'
        FROM SPACEXTABLE
        WHERE Booster_Version LIKE 'F9 v1.1%'
```

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

Average Payload for F9 v1.1 Booster
2534.6666666666665

First Successful Ground Landing Date

- The SQL **WHERE** command was used to find the records for the successful landing outcomes on a ground pad.
- The **MIN** command was used to find the date of the first of these successful landing outcomes on *December 22, 2015*.

```
%%sql SELECT MIN(Date) AS 'Date First Successful Ground Pad Landing'
        FROM SPACEXTABLE
        WHERE Landing_Outcome = 'Success (ground pad)'
```

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

Date First Successful Ground Pad Landing
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- The SQL **WHERE** command was used to find the launches which successfully landed on drone ship and had payload mass greater than 4000 but less than 6000.
- The **GROUP BY** command was used to find the booster versions used for these launches.

```
%%sql SELECT Booster_Version AS 'Booster'
    FROM SPACEXTABLE
    WHERE PAYLOAD_MASS__KG_ > 4000 and
          PAYLOAD_MASS__KG_ < 6000 and
          Landing_Outcome = 'Success (drone ship)'
    GROUP BY Booster_Version
```

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

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

Total Number of Successful and Failure Mission Outcomes

```
%%sql SELECT Mission_Outcome AS 'Mission Outcome',
      COUNT(*) AS 'Total'
      FROM SPACEXTABLE
      GROUP BY Mission_Outcome
```

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

Mission Outcome	Total
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

- The SQL **GROUP BY** and **COUNT** commands were used to calculate the total number of successful and failure mission outcomes.
- Because one of the successful outcomes has a trailing space, it was considered another type of outcome.
- Overall, there were *100 successes* and only *1 failure*.

Boosters Carried Maximum Payload

- To find the names of the boosters which have carried the maximum payload mass, a sub-query using the SQL **MAX** command was used within the **WHERE** command.

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

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

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

2015 Launch Records

```
%%sql SELECT SUBSTR(Date, 6,2) AS 'Month',
             Landing_Outcome AS 'Landing Outcome',
             Booster_Version AS 'Booster',
             Launch_Site AS 'Launch Site'
    FROM SPACEXTABLE
    WHERE Landing_Outcome = 'Failure (drone ship)' and
          Date LIKE '2015%'
```

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

Month	Landing Outcome	Booster	Launch Site
01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

- The SQL **LIKE** command within a **WHERE** command was used to find the failed landings on a drone ship in 2015.
- Their landing outcomes, months, booster versions, and launch site names were returned in a **SELECT** statement.
- To return the month, a substring command, **SUBSTR**, was used.

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

- The SQL **WHERE** command was used to find the launches in this date range.
- To organize the records by their landing outcomes, the **GROUP BY** command was used.
- The **COUNT** command with the descending order option, **DESC**, was used to rank the count of landing outcomes.

```
%%sql SELECT Landing_Outcome AS 'Landing Outcome',
      COUNT(*) AS 'Total Outcomes'
      FROM SPACEXTABLE
      WHERE Date > '2010-06-04' and Date < '2017-03-20'
      GROUP BY Landing_Outcome
      ORDER BY COUNT(*) DESC
```

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

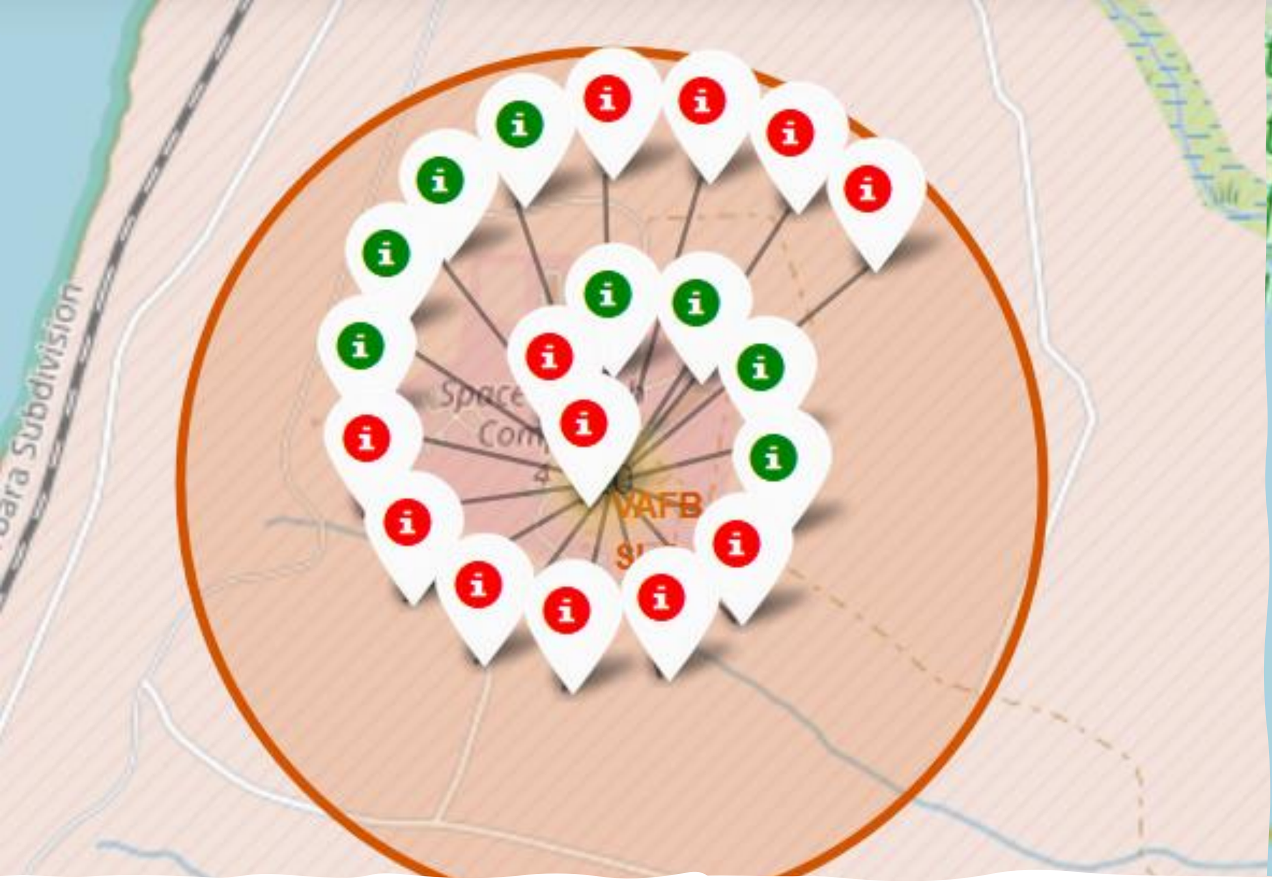
Landing Outcome	Total Outcomes
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Precluded (drone ship)	1
Failure (parachute)	1

Section 3

Launch Sites Proximities Analysis

Location of Launch Sites

- Using the Folium module, we were able to mark all four launch sites on a global map.
- The two sites at Cape Canaveral are located close together and are hard to distinguish at this zoom level.



Launch Outcomes at Each Site

- By marking successful launches in green and the failures in red, we can quickly compare the success rates of different sites.
- Above, we can see that the VAFB SLC-4E site wasn't as successful as the KSC LC-39A site.

Launch Site Proximities to Resources and Populated Areas

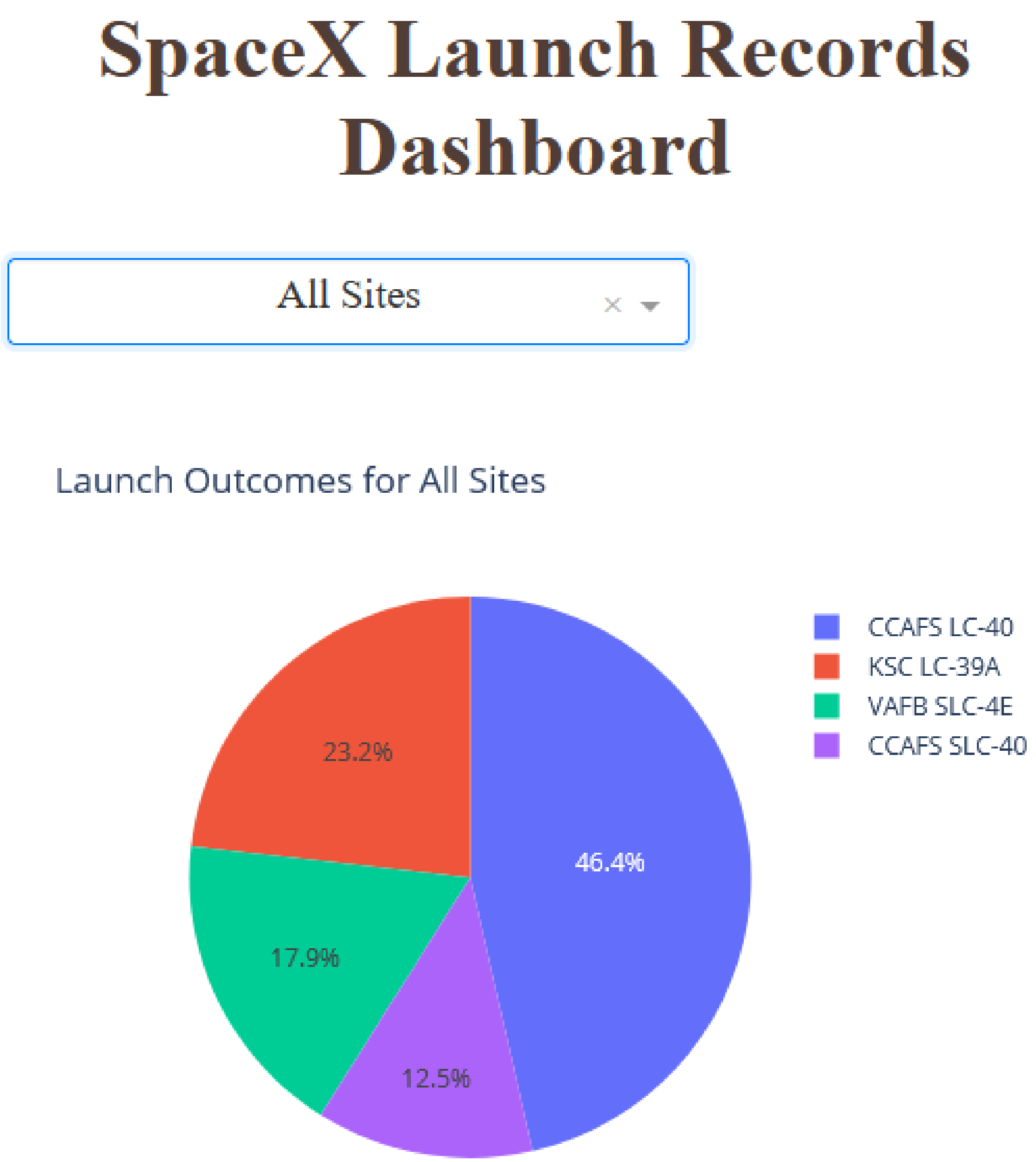
- All the launch sites are located close to railroads and highways, as well as the ocean. And they are located far from heavily populated areas.
- For instance, VAFB SLC-4E is a little more than one kilometer from a highway, railroad and the ocean. However, it's over 14 kilometers from the nearest city.

Section 4

Build a Dashboard with Plotly Dash

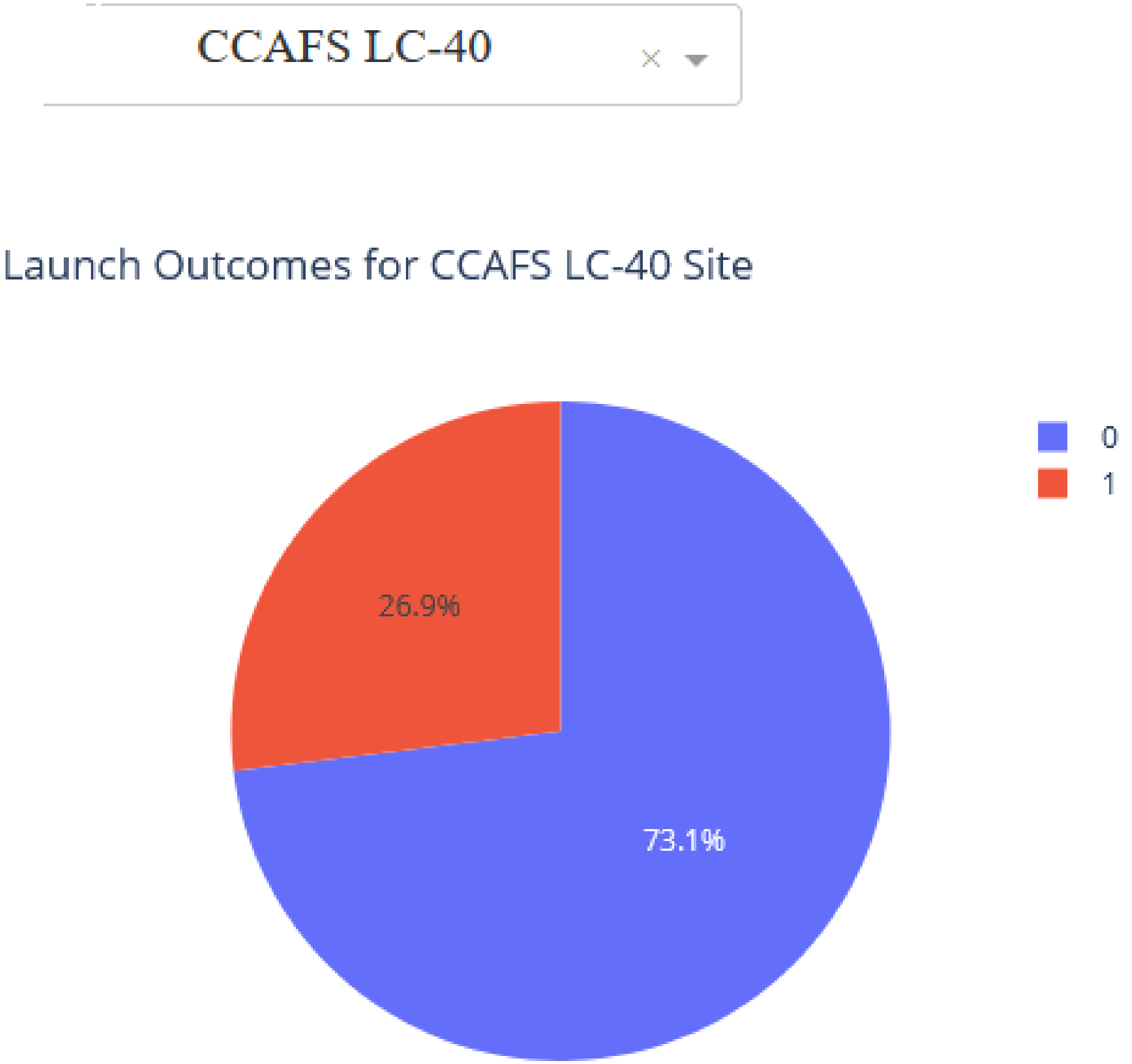
Number of Successful Launches for All Sites

- Almost 50% of the successful launches occurred at one site.
- The *CCAFS LC-40* site had twice as many successful launches compared to all the other sites.
- The site with the fewest number of successes was the *CCAFS SLC-40* site.



Success Rate for CCAFS LC-40 Site

- For the launch site with the highest number of successes, *CCAFS LC-40*, only 26.9% of its launches were successful.
- This indicates the site had more launches than the other sites, because only about a quarter were successful.



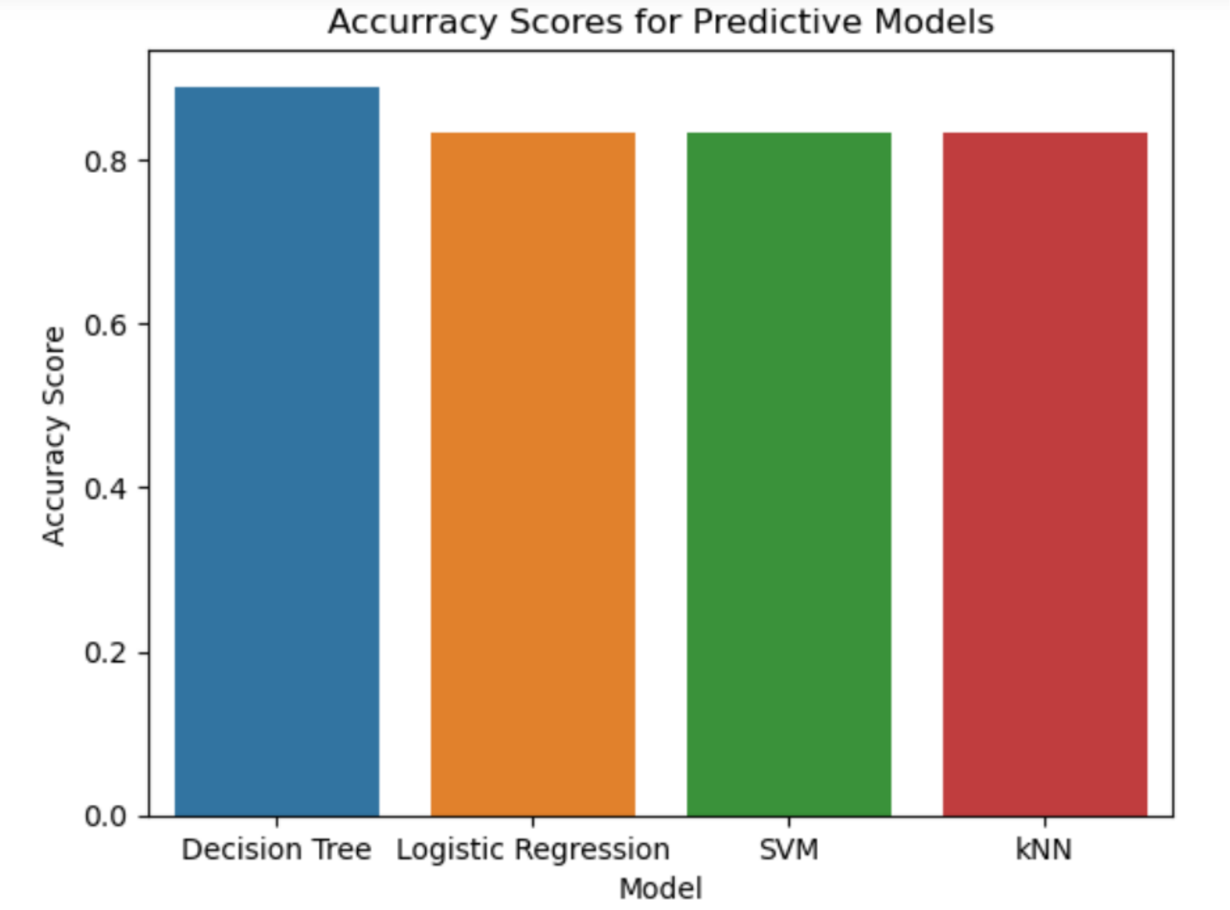
Success Launches for 3k - 4k Payloads by Booster

- In the payload range of *3,000* to *4,000* kg, there were more than twice as many successful launches as failures.
- The most successes were evenly divided between the *FT* and *B4* booster versions.

Section 5

Predictive Analysis (Classification)

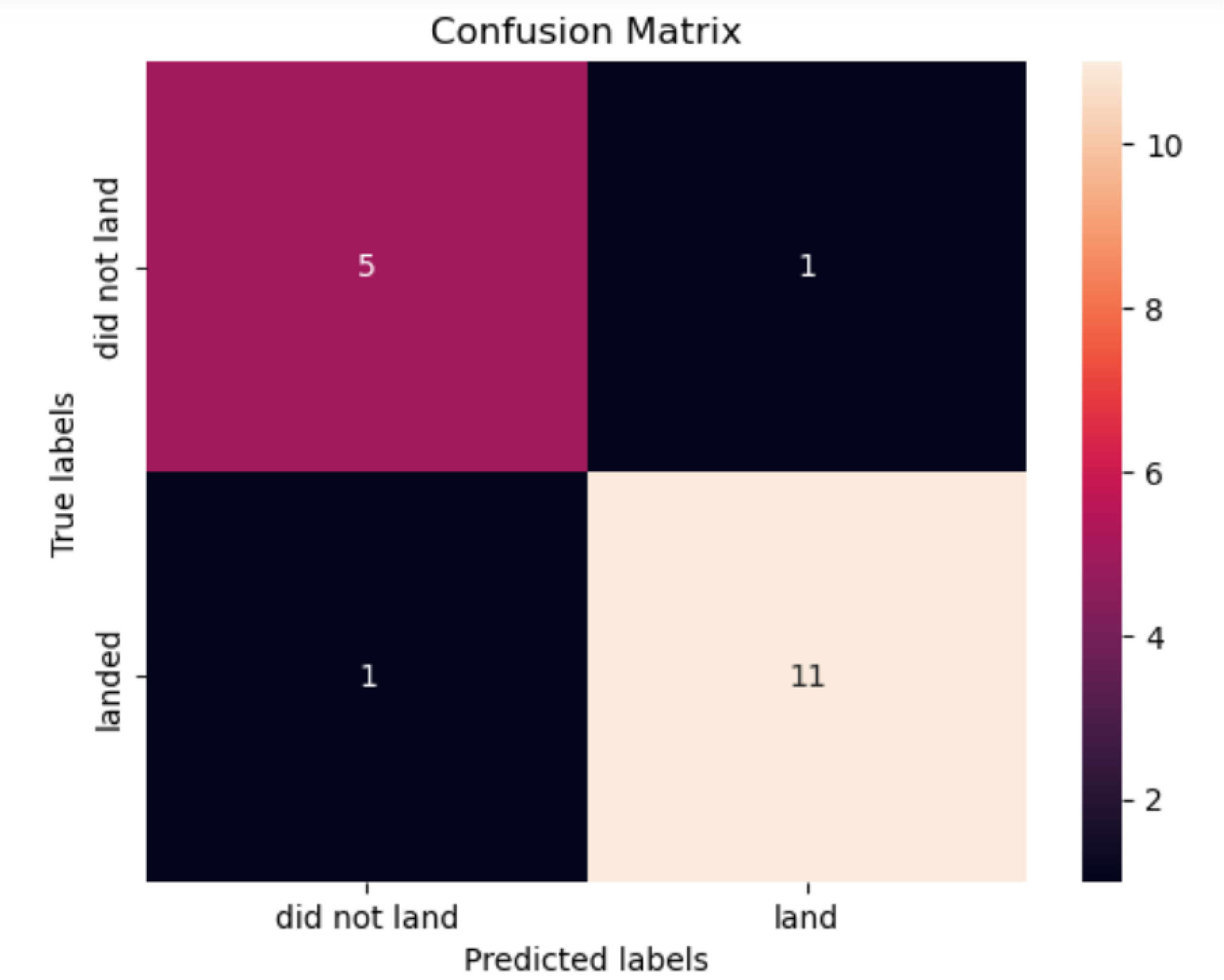
Classification Accuracy



- Although the *k-Nearest Neighbor*, *Support Vector Machine*, and *Logistic Regression* models all scored **83.3%** on the test data, the *Decision Tree* model had the highest accuracy score of **88.9%**.

Decision Tree Confusion Matrix

- On the test data, the *Decision Tree* classifier only had 1 *false positive* and 1 *false negative*.
- However, the sample size is small, and further testing should be done to validate the model.



Conclusions

Use the KSC LC 39A Launch Site on Merritt Island, Florida

Target ES-L1, GEO, HEO, and SSO Orbits

Focus on Payload Masses between 4k and 6k kg

Use One of These Boosters: F9 FT B1021.2, B1031.2, B1022, or B1026

Appendix

- [Link to database used for SQL queries]()

Thank you!