IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Glen Brunke
August 9, 2023

Outline

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

Executive Summary

- **Problem and Business Context**:
  - SpaceX uses re-usable rocket boosters which has the potential to cut the cost of rocket launches in more than half from $165 million to as little as $62 million. This analysis seeks to determine if the first stage will land to understand the potential cost of a competitive bid to SpaceX.
- **Data Collection and Exploration**:
  - Data was collected from public sources including the SpaceX API and Wikipedia. The data was then cleaned and transformed using various methods including Pandas DataFrames and BeautifulSoup HTML parser. The data was then loaded into a SQL Lite database and explored via SQL queries. The data was processed into a dashboard using Plotly Dash in Python. Finally, an analysis was conducted using Machine Learning techniques to determine the best model to accurately predict launch success.
- **Summary of all results:**
  - The KNN model was the best overall machine learning model to predict whether a rocket booster would be able to land or not.
  - The KNN model can be used to estimate with an 84% accuracy the reusability of a rocket booster. The unknown 16% will need to be accounted for in budgeted for a competitive bid.

Introduction

- SpaceX uses re-usable rocket boosters which has the potential to cut the cost of rocket launches in more than half from $165 million to as little as $62 million.
- This analysis seeks to determine if the first stage will land to understand the potential cost of a competitive bid to SpaceX.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - The data was collected from two sources via Python GET commands: 1) The SpaceX API 2) Wikipedia
- Perform data wrangling
  - The data was processed using Beautiful Soup and Pandas DataFrames
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Multiple machine learning models used to uncover the best method and parameters using GridSearch
  - Linear and Logistic Regress, Decision Trees, K Nearest Neighbors

Data Collection

- The datasets were collected 1) Directly from SpaceX via their API and from a static Wikipedia page
- The API was requested via the Request object and processed using Panda DataFrames

- The Wikipedia page was processed with BeautifulSoup and loaded into Panda DataFrames

Data Collection – SpaceX API

- Data was collected via SpaceX API
- Data was filtered to include only Falcon 9 launches
- Data was cleaned and null values were processed.

https://github.com/glenbrunke/data_science_capstone/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

Data Collection - Scraping

- Data was captured using Beautiful Soup from a static Wikipedia page
  - Reference: https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922

GitHub https://github.com/glenbrunke/data_science_capstone/blob/main/jupyter-labs-webscraping.ipynb

Data Wrangling

- Data was loaded from a CSV file from previous data extraction methods
- The data was cleaned and prepared for analysis
- The data was finally stored in a CSV file for later analysis

GitHub https://github.com/glenbrunke/data_science_capstone/blob/main/labs-jupyter-spacex-data_wrangling_jupyterlite.jupyterlite.ipynb

EDA with Data Visualization

- Data visualized using pandas and seaborn:
  - Scatter plots
    - Payload Mass and Flight Number
    - Flight Number and Launch Site
    - Payload Mass and Launch Site
    - Flight Number and Orbit Type
    - Payload Mass and Orbit Type
  - Line Chart
    - Launch Success By Year
  - Bar Chart
    - Launch success by orbit type

https://github.com/glenbrunke/data_science_capstone/blob/main/jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb

EDA with SQL

- All unique launch sites
- All records where launch site began with ‘CCA’
- Total payload mass from NASA (CRS)
- Average payload for booster F9 v1.1
- First successful landing on ground pad
- Boosters with success landing on drone ship
- Looked at total successful and failed outcomes
- Names of booster that carried the largest payloads
- Dates for and landing sites in 2015
- Count of all landing outcomes from 2010 to 2017

https://github.com/glenbrunke/data_science_capstone/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- Added markers for each launch site with marker clusters indicating the success or failure of launches from each site.
- Added location data for coast lines, rail ways, and large cities to determine the ideal location for launch sites.
- The success/failure clusters allow you to easily see which sites have a high success rate.

https://github.com/glenbrunke/data_science_capstone/blob/main/lab_jupyter_launch_site_location.jupyterlite.ipynb

Build a Dashboard with Plotly Dash

- Added pie chart to show the total successful launches for all sites or the success/failure rate for a particular launch site if selected.
- Added scatter plot to show the success and failure of payload size and launches by site. This allows you to easily see which launch sites and payload sizes have the most success.

https://github.com/glenbrunke/data_science_capstone/blob/main/spacex_dash_app.py

Predictive Analysis (Classification)

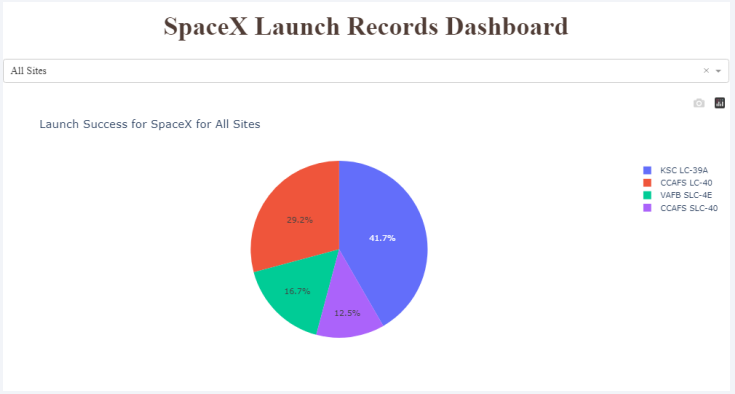
- I used several machine learning models to predict launch success with available launch data:
  - K-nearest neighbor
  - Logistic Regression
  - Decision Tree Classification
  - Support Vector Machines

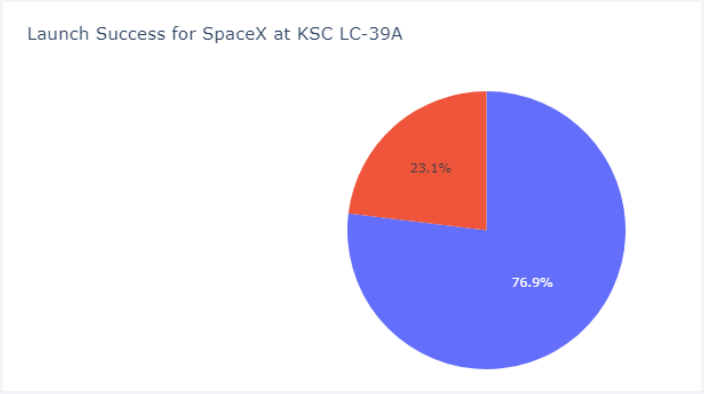
- The models were tuned using GridSearchCV to find the best parameters and test and training data scores were used to determine the best model.

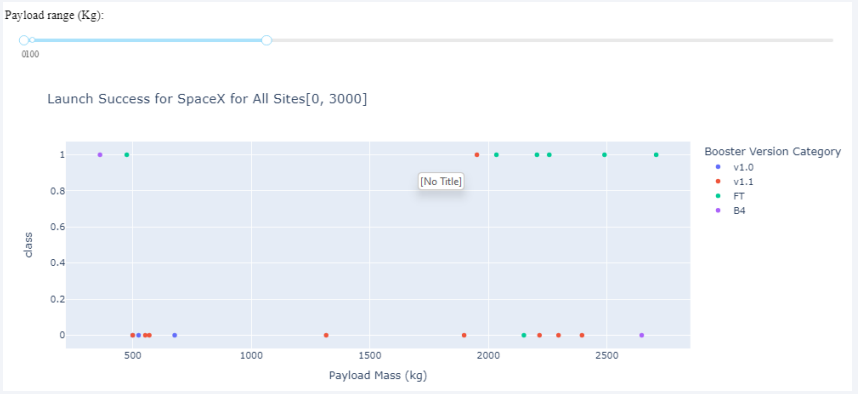
GitHub https://github.com/glenbrunke/data_science_capstone/blob/main/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

Results

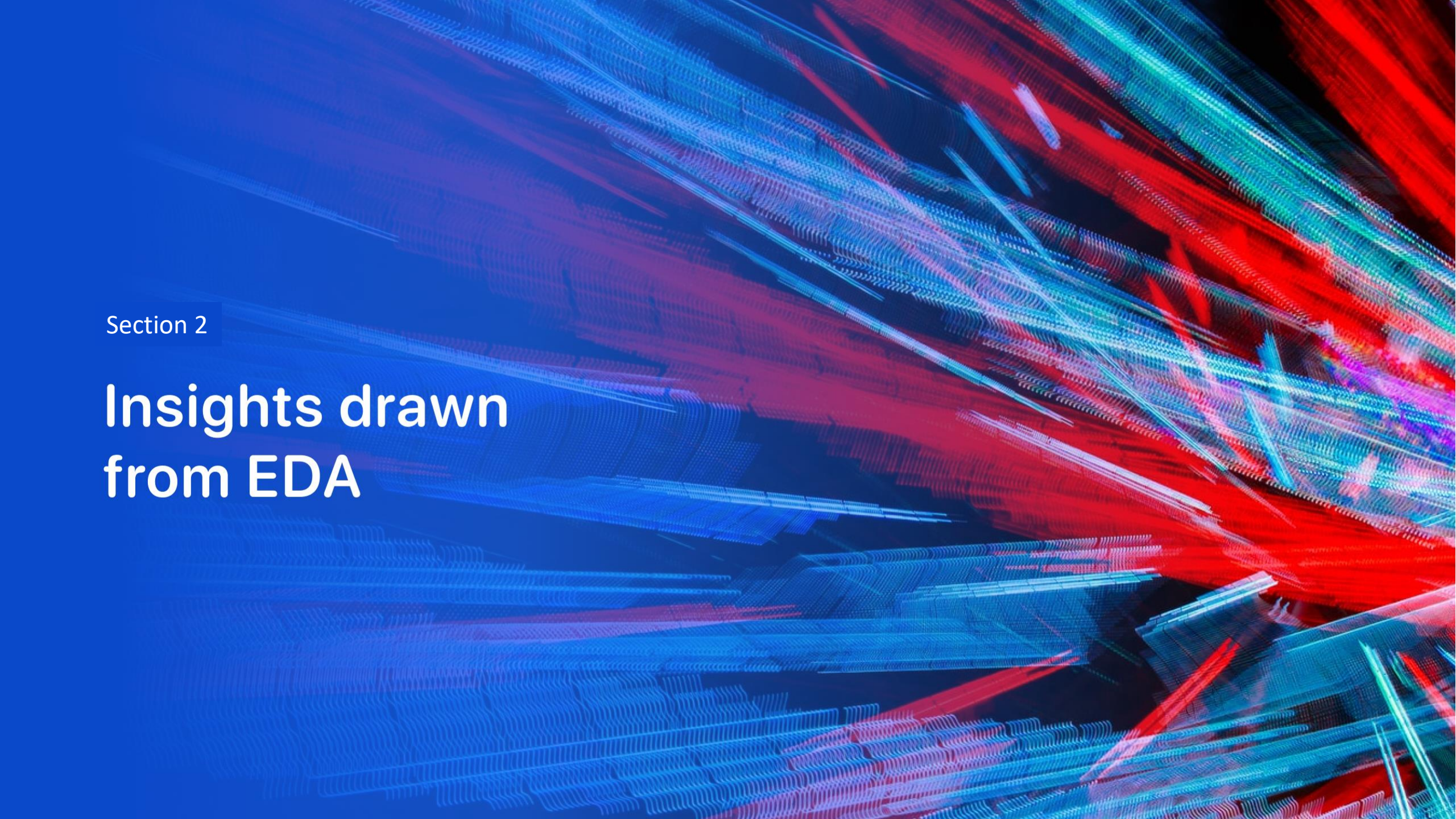
- Exploratory data analysis results
  - Deeper understanding of the data and potential relationships were observed in exploratory analysis to be confirmed with data models.
- Interactive analytics demo in screenshots





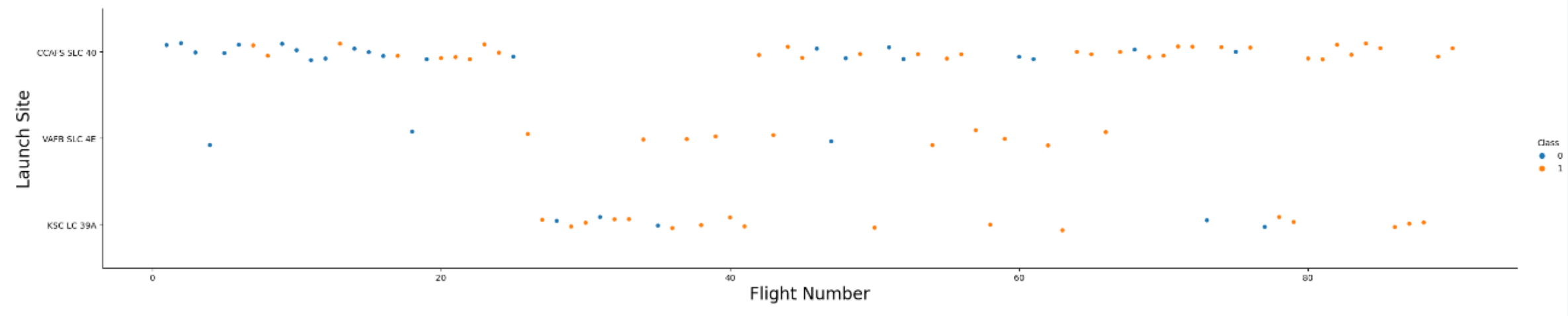


- Predictive analysis results
  - The DecisionTreeClassifier had the highest overall training data R^2 score, however, it performed the worst with the test datapoints.
  - The best overall model was the KNN which scored well for training and test data.


Section 2
Insights drawn from EDA

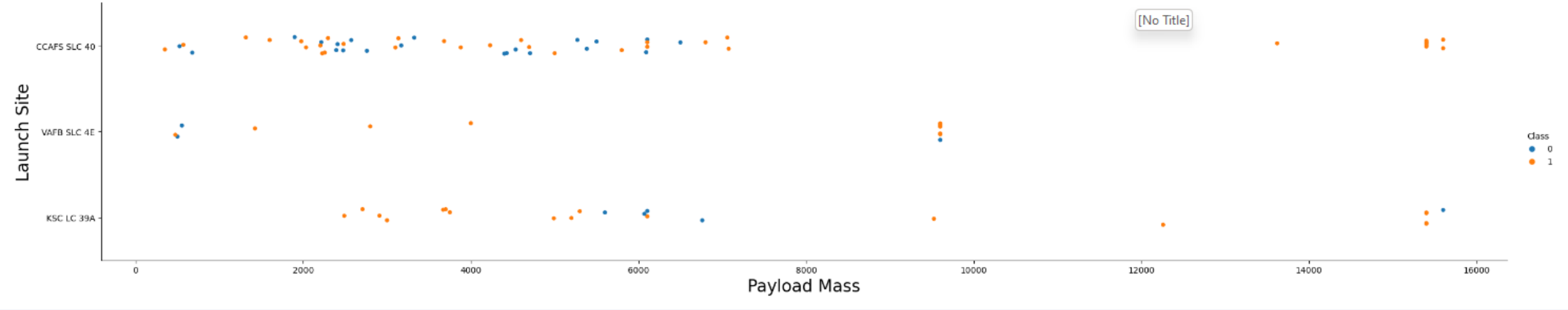
Flight Number vs. Launch Site

- Early flight numbers were concentrated in the Canaveral Space Center with low success and later transitioned to a mix of launch sites across the US with a much higher success rate.



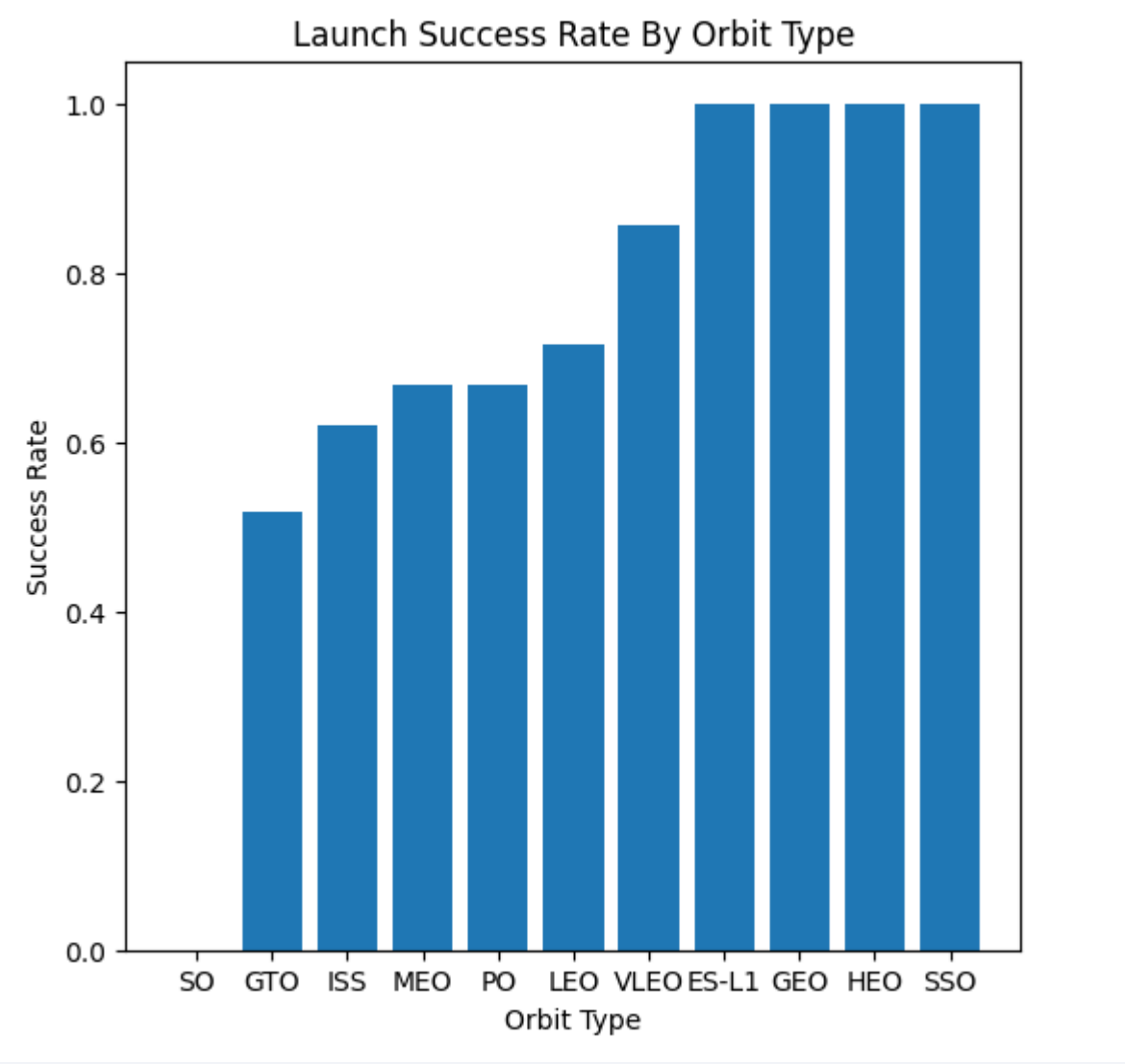
Payload vs. Launch Site

- Most payloads are in the range of 2000-6000kgs with a mixed success rate.
- Heavy payloads greater than 10,000kg are primarily launched from Florida launch sites (KSC and CCAFS) with a high success rate.



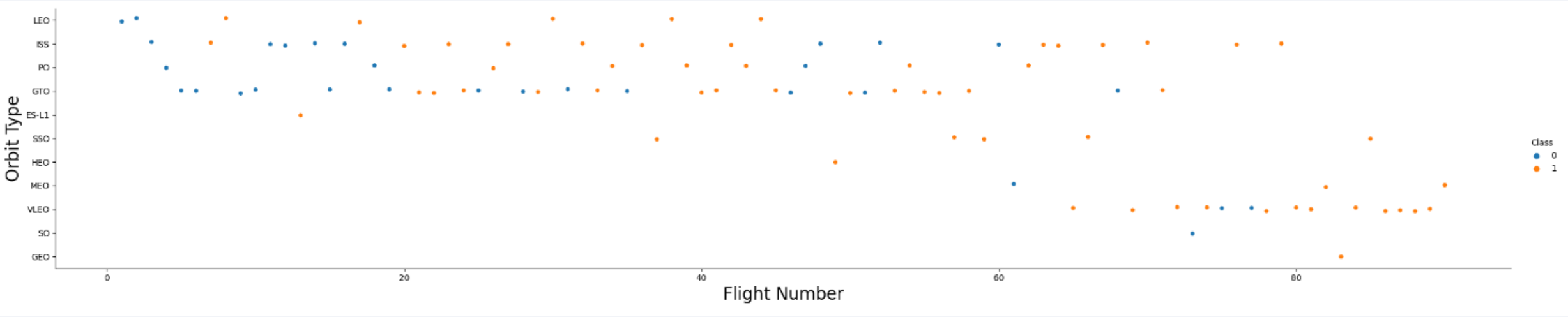
Success Rate vs. Orbit Type

- The most successful launches were for ES-L1, GEO, HEO, and SSO orbit types.
- The least successful orbit type was GTO.



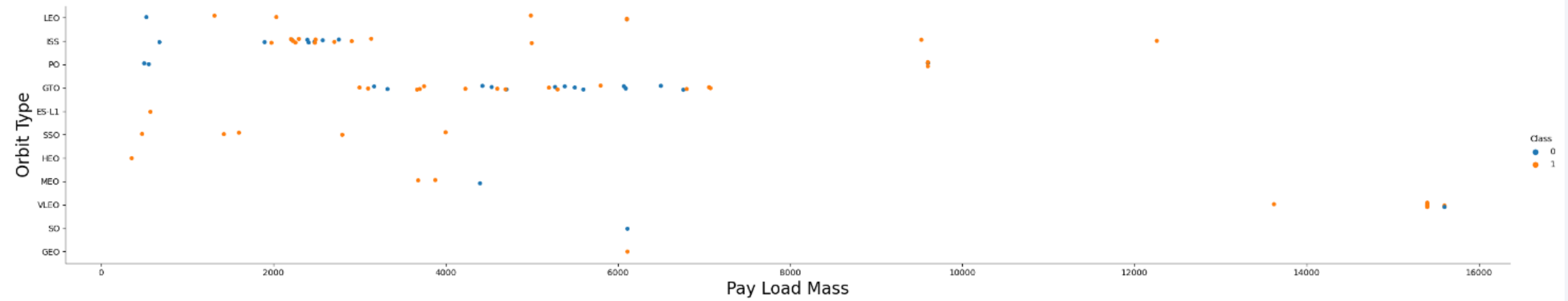
Flight Number vs. Orbit Type

- Early flight numbers concentrated on LEO, ISS, PO, and GTO orbit types with mixed success and many failures.
- Later flight numbers add many more VLEO orbits with a much higher success rate.



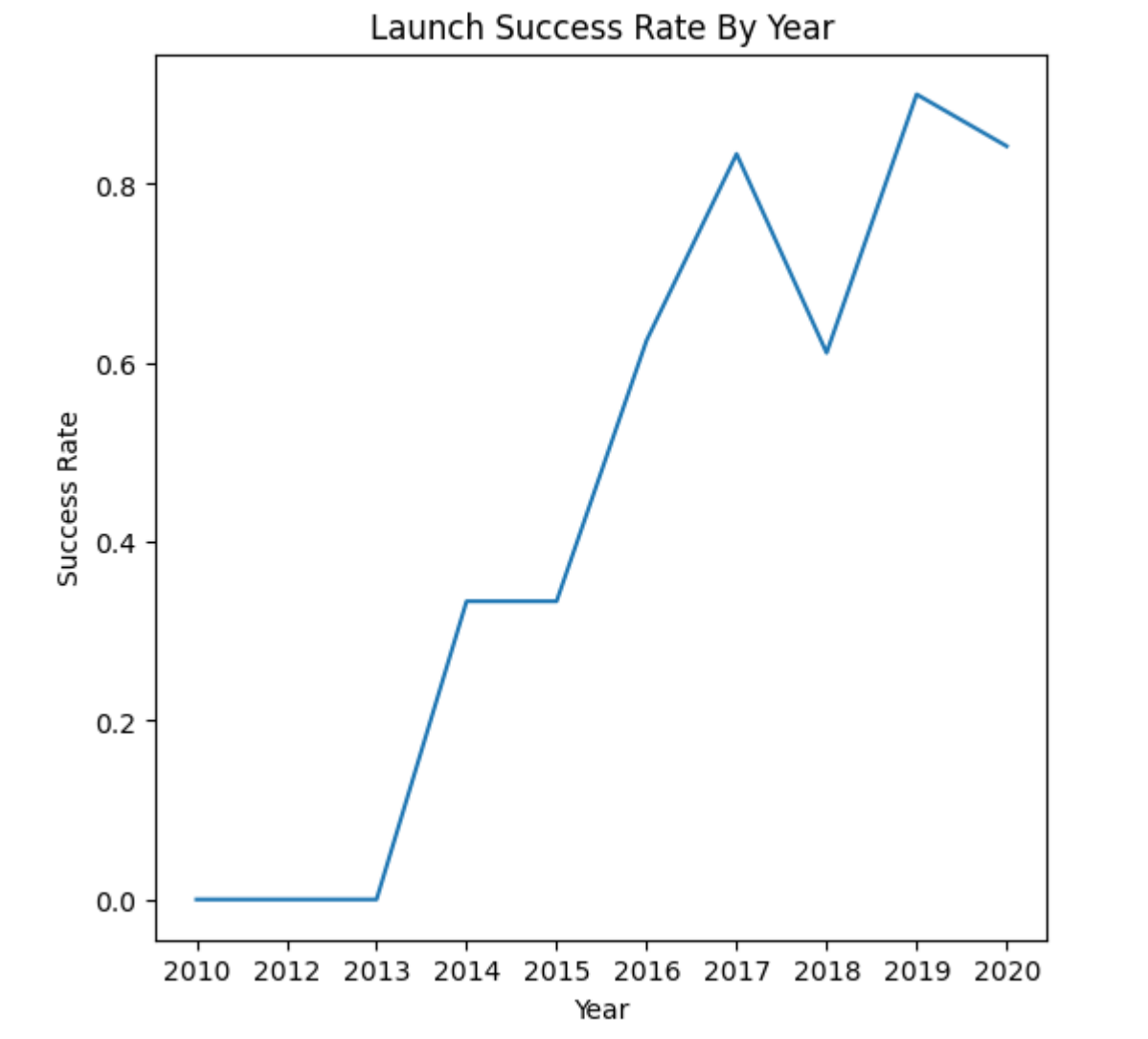
Payload vs. Orbit Type

- The heaviest payloads are of VLEO orbit type.



Launch Success Yearly Trend

- Success rates have improved dramatically since the beginning launches in 2013.
- Rates have dipped in 2018 and 2020 slightly, but not down to the levels seen in the early days.



All Launch Site Names

- To find all of the unique launch site names, I used the ‘DISTINCT()’ function in the SQL language.

```
[11]: %sql select DISTINCT(Launch_Site) from SPACEXTABLE
 * sqlite:///my_data1.db
Done.
[11]: ,,,,,,,,,,,,,,,,,,,,
```

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

Launch Site Names Begin with 'CCA'

- To search for 5 records with a launch site that begins with ‘CCA’, I used the ‘like “CCA%” to search for a string that begins with those letters.
- To select only 5 records, I used the “limit 5” in the SQL.

```
[12]: %sql select * from SPACEXTABLE WHERE Launch_Site like "CCA%" LIMIT 5
 * sqlite:///my_data1.db
Done.
[12]:
```

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

Total Payload Mass

- Using the SUM() function in SQL, I was able to find the total payload mass where the customer was ‘NASA (CRS)’

```
[15]: %sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE Customer='NASA (CRS)'
 * sqlite:///my_data1.db
Done.
[15]: ''''''''''
```

SUM(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

- To calculate the average payload mass carried by booster version F9 v1.1, I used the AVG() function in SQL.

```
[16]: %sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE Booster_Version ='F9 v1.1'
 * sqlite:///my_data1.db
Done.
[16]: ,,,,,,,,,,,
AVG(PAYLOAD_MASS__KG_)
2928.4
```

First Successful Ground Landing Date

- To find the date of the first successful landing outcome on a ground pad, I used the MIN() function with the date as my variable.

```
[17]: %sql SELECT MIN(Date) FROM SPACEXTABLE WHERE Landing_Outcome ='Success (ground pad)'
       * sqlite:///my_data1.db
      Done.
[17]: ,,,,,,,,,,,
      MIN(Date)
      2015-12-22
```

Successful Drone Ship Landing with Payload between 4000 and 6000

- To list the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000, I used the 'AND' statement in the 'WHERE' clause to filter the data.

```
[19]: %sql SELECT Booster_Version FROM SPACEXTABLE WHERE Landing_Outcome ='Success (drone ship)' AND PAYLOAD_MASS__KG_ between 4000 and 6000
 * sqlite:///my_data1.db
Done.
[19]: ''''''''''''''''''''
```

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

Total Number of Successful and Failure Mission Outcomes

- To calculate the total number of successful and failure mission outcomes, I used the ‘GROUP BY’ clause and the COUNT() function to total the data instances.

```
[41]: %sql SELECT Mission_Outcome, COUNT(Mission_Outcome) as Mission_Count from SPACEXTABLE GROUP BY Mission_Outcome
 * sqlite:///my_data1.db
Done.
[41]: ,,,,,,,,,,,,,,,,,,,,,,,,
```

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

Boosters Carried Maximum Payload

- To list the names of the boosters which have carried the maximum payload mass, I used a sub-query in the WHERE clause to find the highest payload.

```
[48]: %sql SELECT Booster_Version, PAYLOAD_MASS__KG_ from SPACEXTABLE WHERE PAYLOAD_MASS__KG_ IN (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE)

 * sqlite:///my_data1.db

Done.
```

[48]:

Booster_Version	PAYLOAD_MASS__KG_
F9 B5 B1048.4	15600
F9 B5 B1049.4	15600
F9 B5 B1051.3	15600
F9 B5 B1056.4	15600
F9 B5 B1048.5	15600
F9 B5 B1051.4	15600
F9 B5 B1049.5	15600
F9 B5 B1060.2	15600
F9 B5 B1058.3	15600
F9 B5 B1051.6	15600
F9 B5 B1060.3	15600
F9 B5 B1049.7	15600

2015 Launch Records

- To list the failed landing outcomes in drone ship, their booster versions, and launch site names for in year 2015, I used the SUBSTR() function to select the year and month.

```
[64]: %sql SELECT substr(Date, 6, 2) as month, substr(Date,1,4) as year, Launch_Site, Landing_Outcome from SPACEXTABLE where year='2015'

 * sqlite:///my_data1.db
Done.
[64]:
```

month	year	Launch_Site	Landing_Outcome
10	2015	CCAFS LC-40	Failure (drone ship)
11	2015	CCAFS LC-40	Controlled (ocean)
02	2015	CCAFS LC-40	No attempt
04	2015	CCAFS LC-40	Failure (drone ship)
04	2015	CCAFS LC-40	No attempt
06	2015	CCAFS LC-40	Precluded (drone ship)
12	2015	CCAFS LC-40	Success (ground pad)

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

- To 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, I used the COUNT() function, the ORDER BY clause, and a WHERE clause to to filter the correct date range.

```
[79]: %sql SELECT Landing_Outcome, count(Landing_Outcome) as NumOC from SPACEXTABLE WHERE Date < '2017-03-20' and Date > '2010-06-04' GROUP BY Landing_Outcome ORDER BY NumOC DESC

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

[79]:

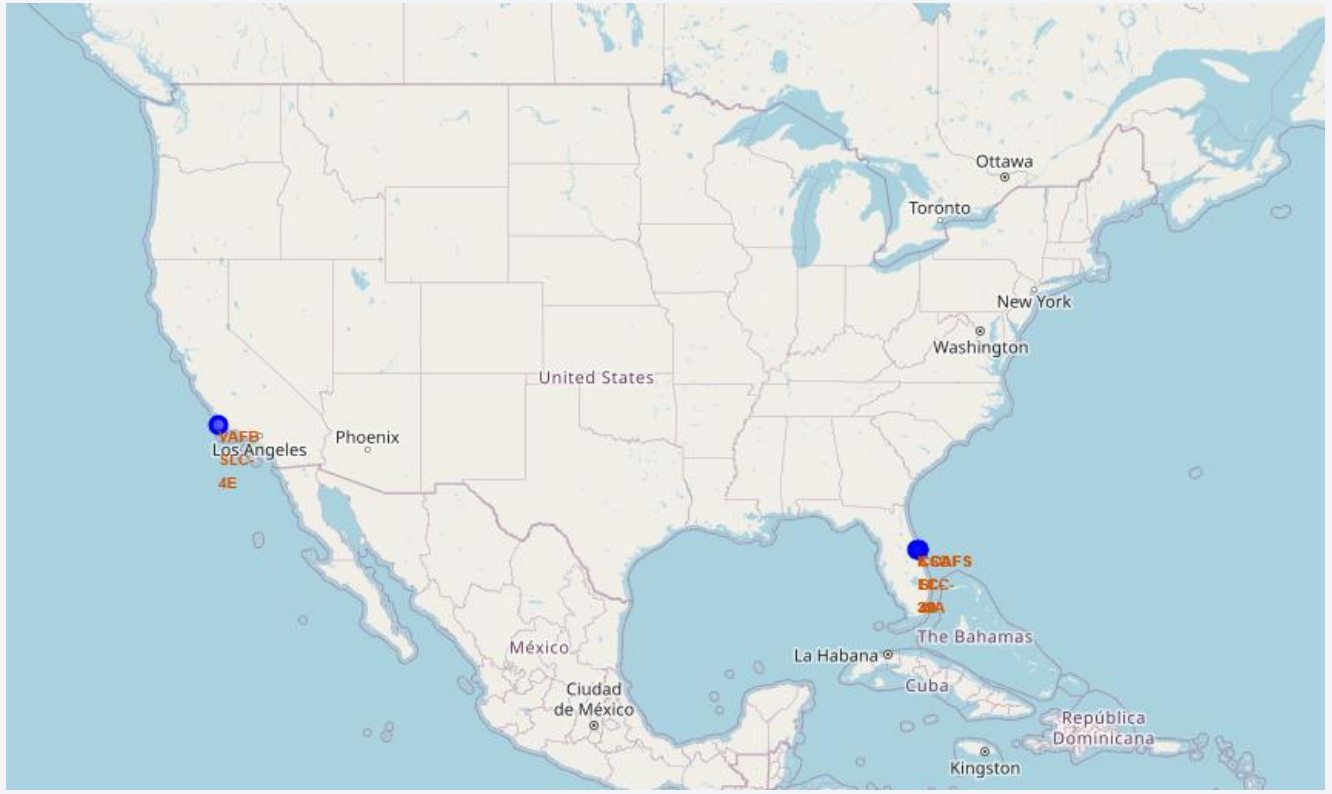
Landing_Outcome	NumOC
No attempt	10
Success (ground pad)	5
Success (drone ship)	5
Failure (drone ship)	5
Controlled (ocean)	3
Uncontrolled (ocean)	2
Precluded (drone ship)	1
Failure (parachute)	1

Section 3

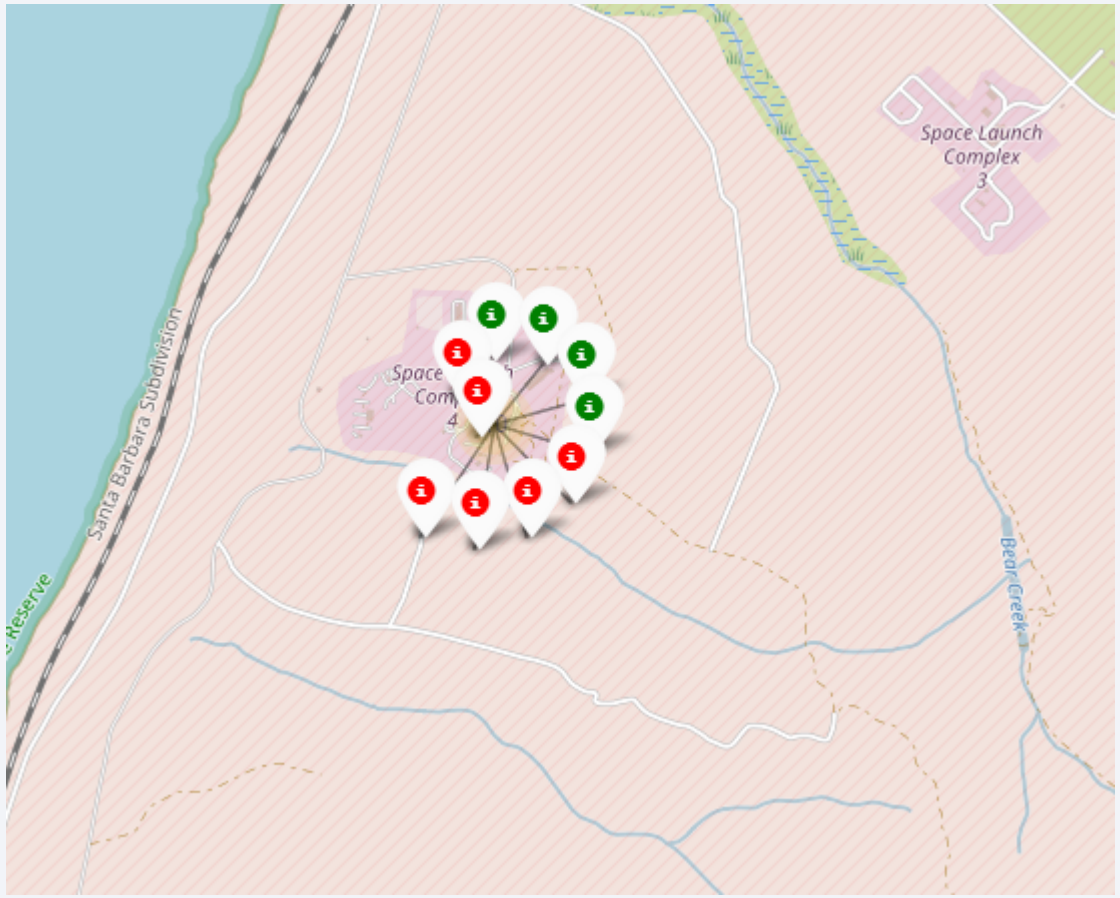
Launch Sites Proximities Analysis

Launch Sites Are Located in Warm, Coastal Regions

- Generally, launch sites are located in warmer, coastal regions within the United States.



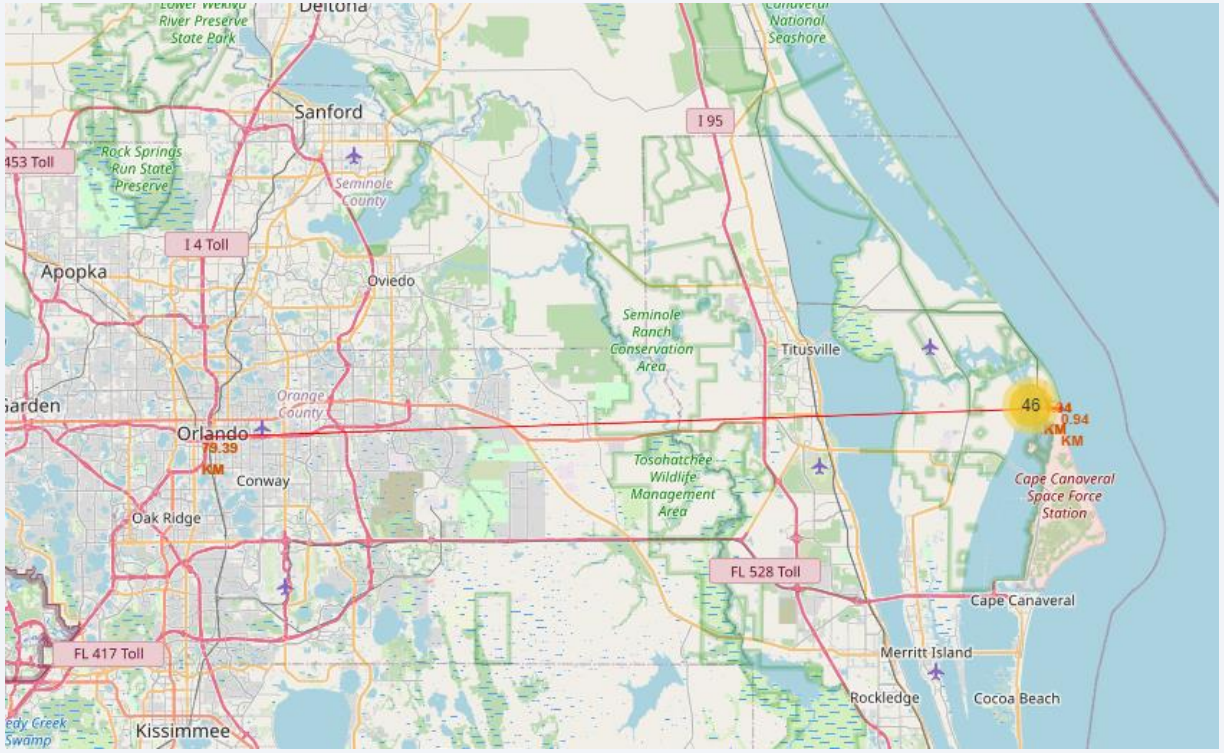
All Launch Sites Have A Mix Of Successful and Unsuccessful Launches

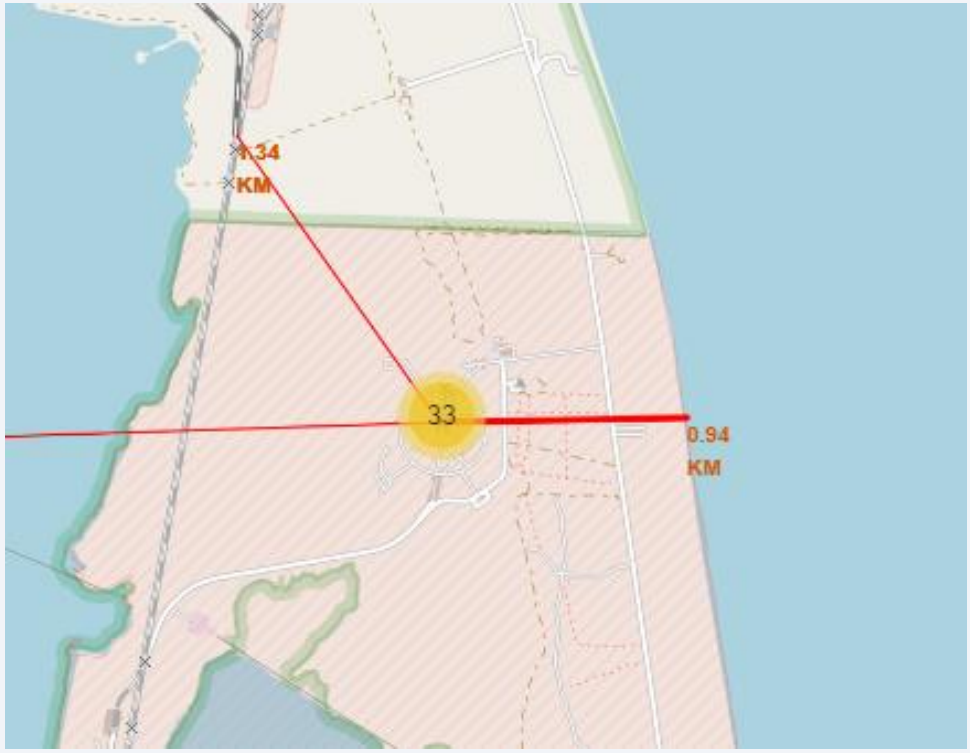


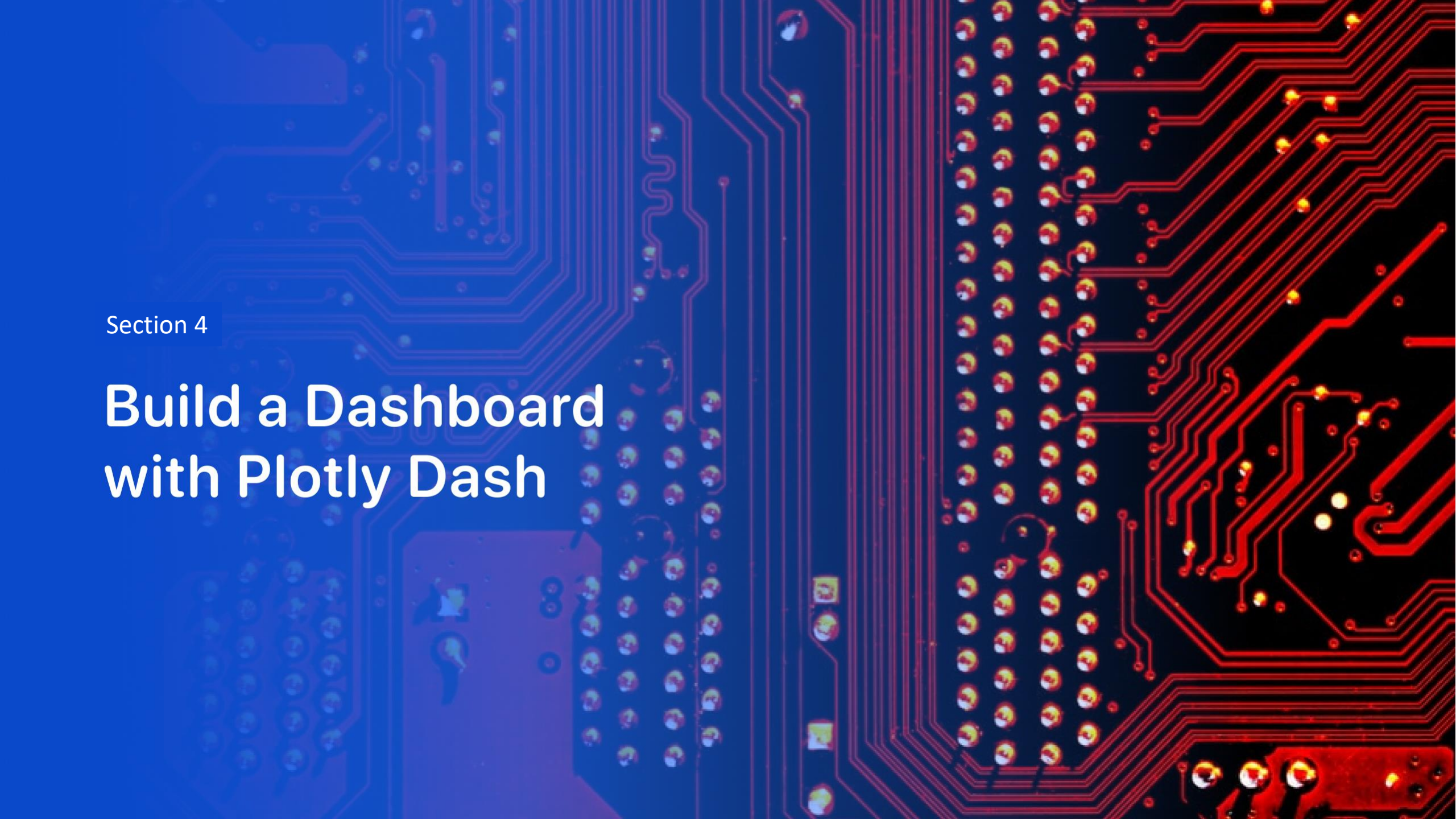
- The example launch site is Vandenberg Space Force Base where 6 of the 10 total launches were unsuccessful.
- This mixed success result is common across all launch sites.

Launch Sites Are Near Coasts And Railways

- Launch sites are chosen to be near the ocean and railways to support launch logistics, but away from major cities to avoid unnecessary risk.

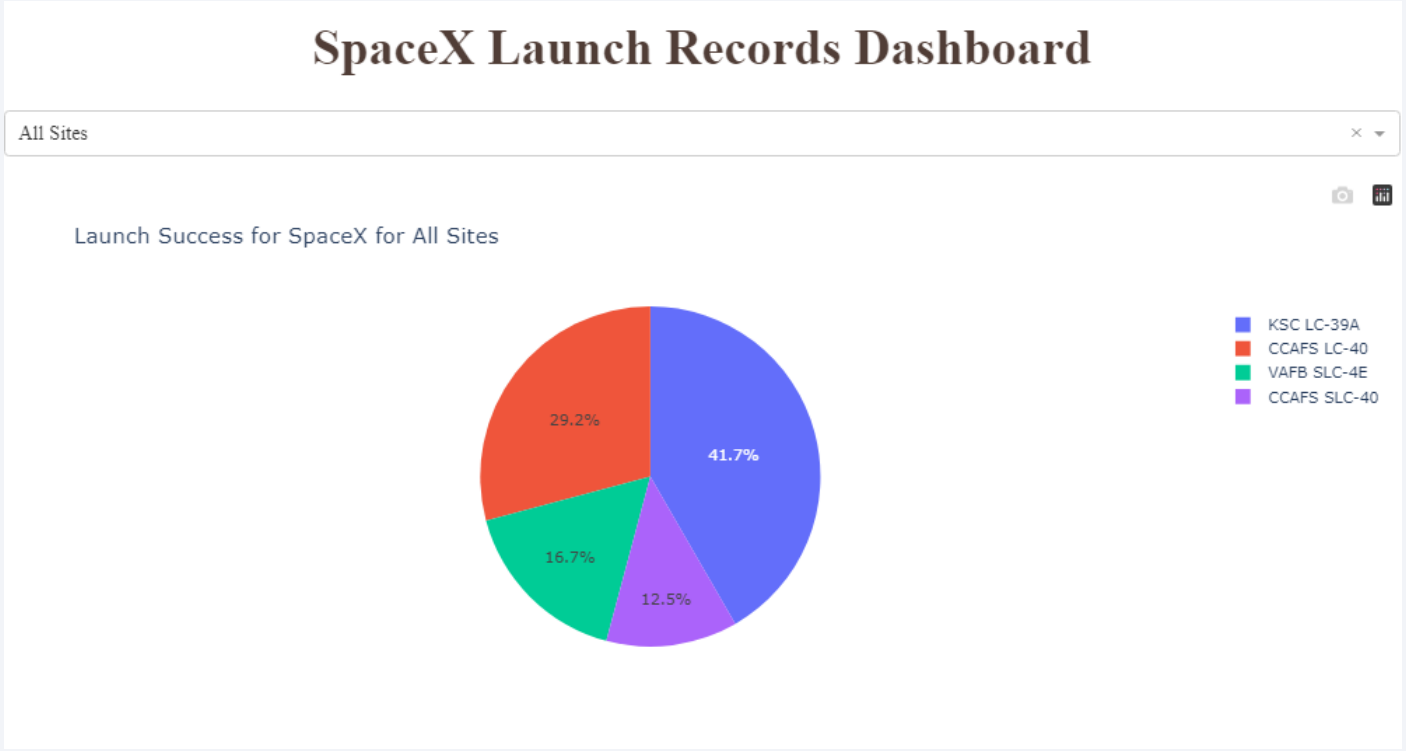





Section 4
Build a Dashboard with Plotly Dash

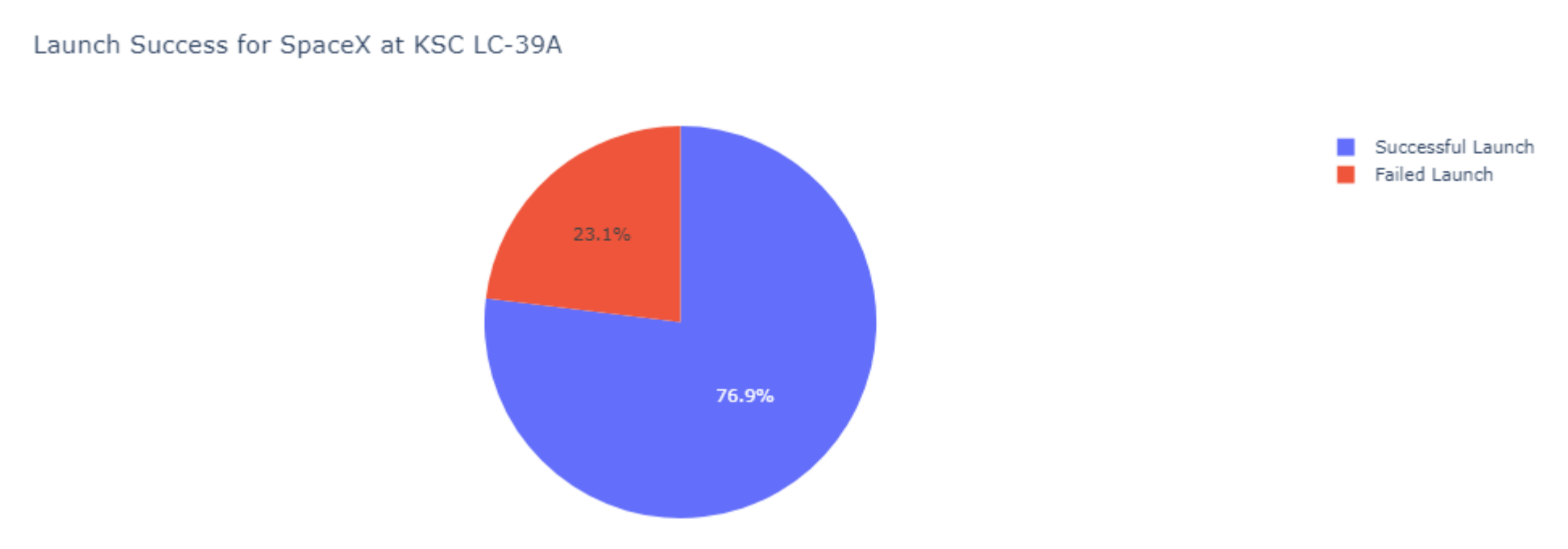
Most Successful Launches Are From Florida Sites

- Kennedy Space Center (KSC LC-39A) and Cape Canaveral Space Force Station (CCAFS LC-40) dominate the total successful launches with over 70% of total successful launches.



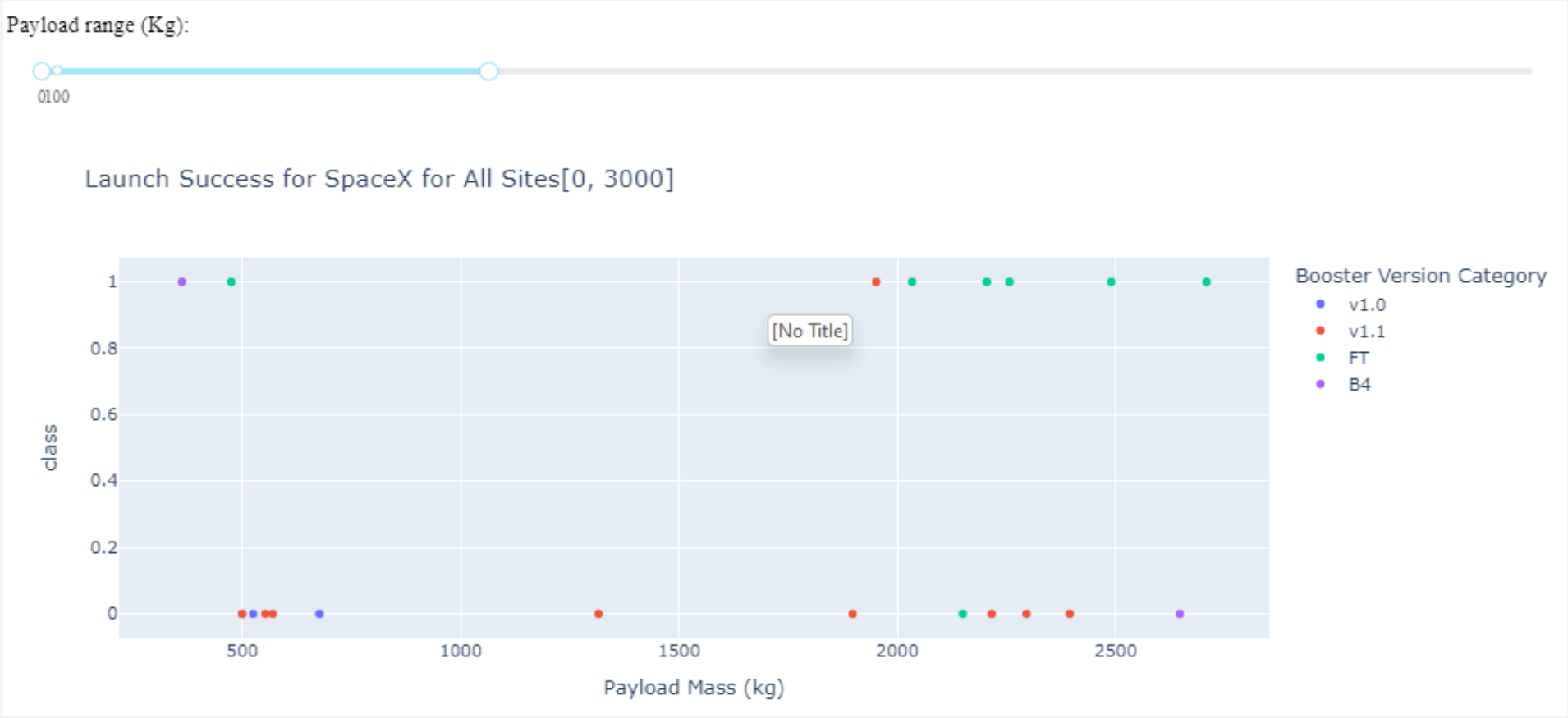
Kennedy Space Center Has The Best Success Record

- Kennedy Space Center (KSC LC-39A) has the highest ratio of successful launches with a 76.9% success rate.



Booster v1.1 Has A Poor Success Rate With Small Payloads

- With payloads under 3000kg, the v1.1 Booster has a very poor success rate with only one successful launch.

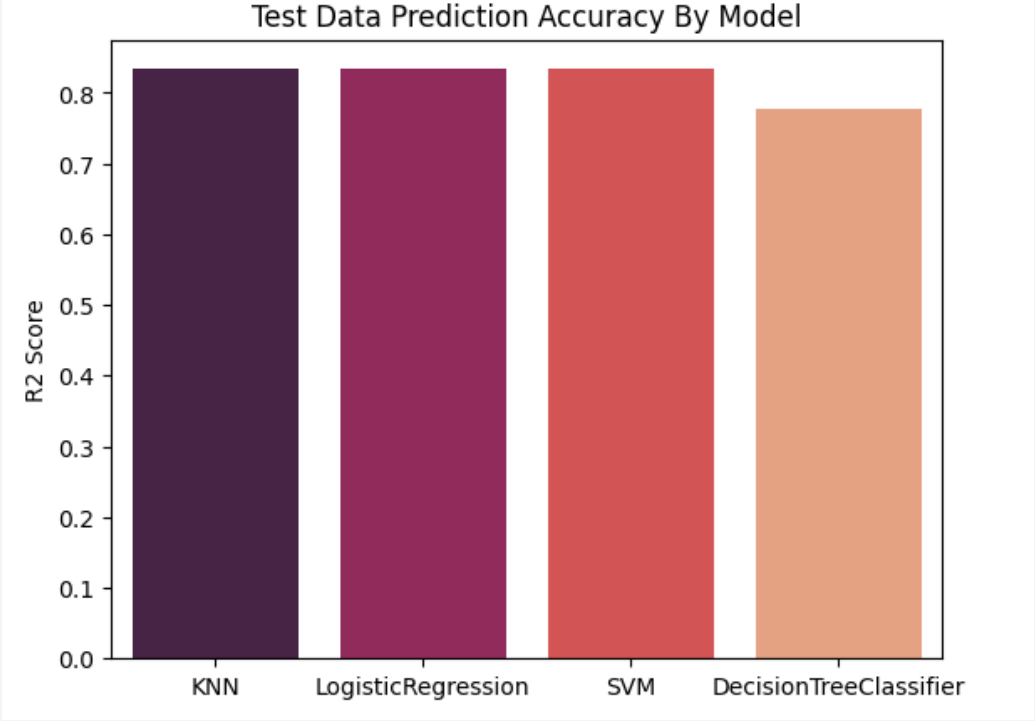


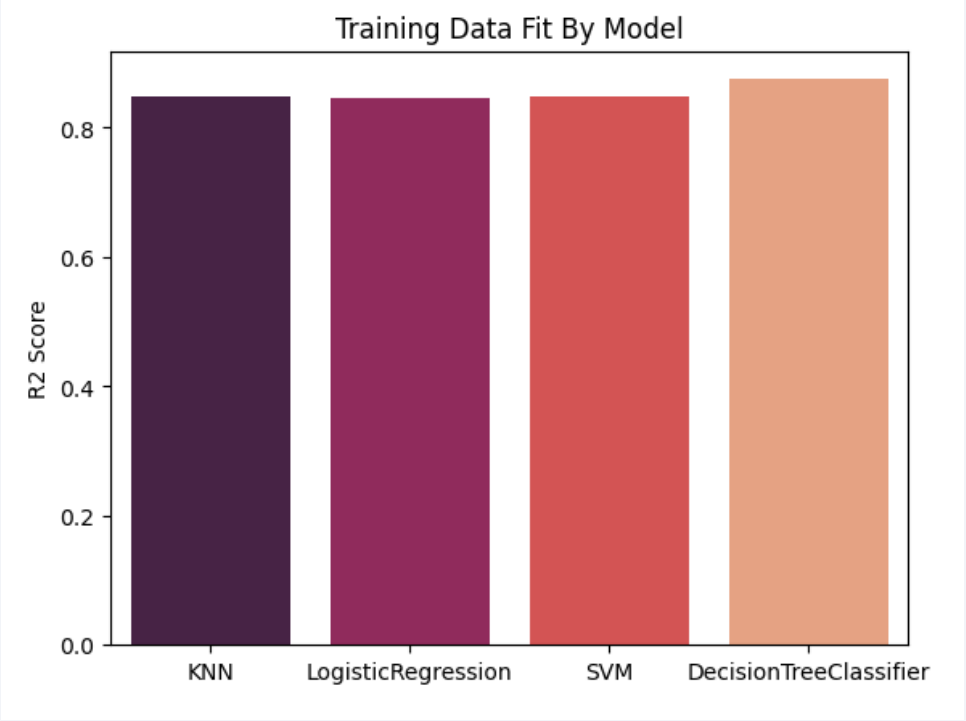
Section 5

Predictive Analysis (Classification)

Classification Accuracy

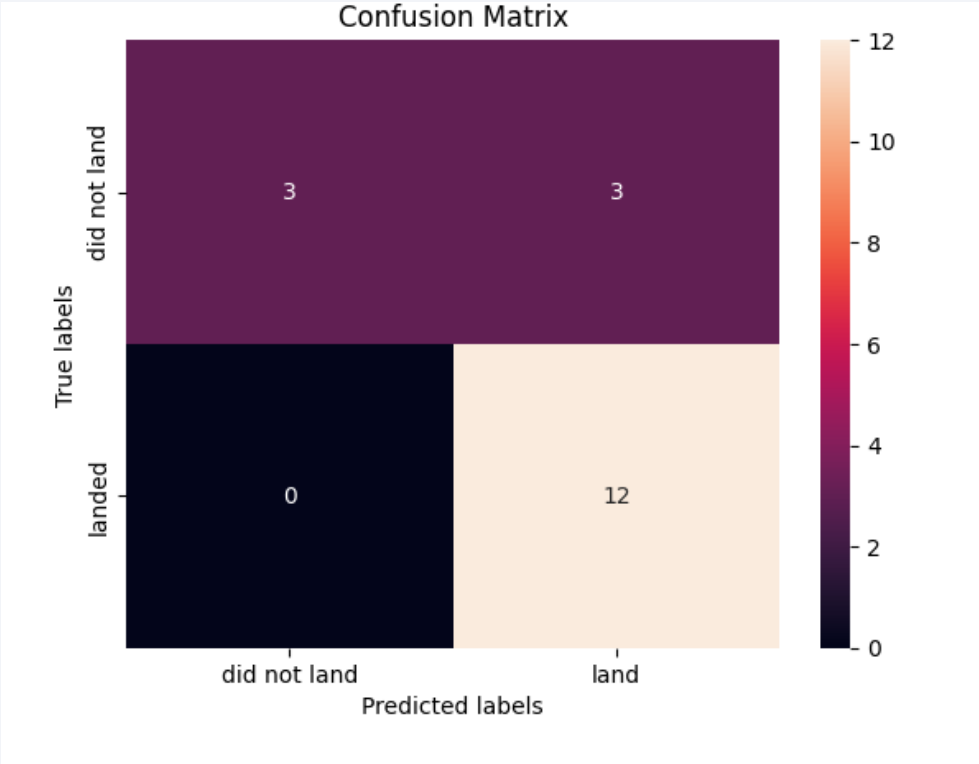
- The DecisionTreeClassifier had the highest overall training data R^2 score, however, it performed the worst with the test datapoints.
- The best overall model was the KNN which scored well for training and test data.





Confusion Matrix

- The KNN model was able to predict 12 landings with training data and did not predict any false failures with test data.



Conclusions

- The KNN model was the best overall machine learning model to predict whether a rocket booster would be able to land or not.
- The KNN model can be used to estimate with an 84% accuracy the reusability of a rocket booster. The unknown 16% will need to be accounted for in budgeted for a competitive bid.

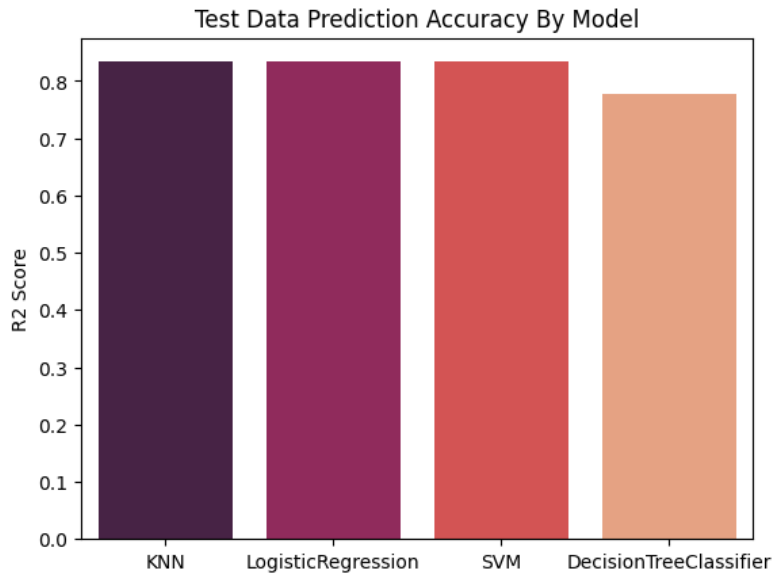
Appendix

- Code to plot test data scores:

```
[62]: test_data_scores = dict({'test_name': ['KNN', 'LogisticRegression', 'SVM', 'DecisionTreeClassifier'],
                               'test_data_score': [0.8333333333333334, 0.8333333333333334, 0.8333333333333334, 0.7777777777777778]})
```

```
[68]: results_plot = sns.barplot(data=test_data_scores, x='test_name', y='test_data_score', palette='rocket')
      results_plot.set_title("Test Data Prediction Accuracy By Model")
      results_plot.set_ylabel("R2 Score")
      results_plot
```

```
[68]: <AxesSubplot:title={'center':'Test Data Prediction Accuracy By Model'}, ylabel='R2 Score'>
```



Thank you!