



IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Brian Fontana
December 16, 2024



Outline

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

Executive Summary

- Data detailing the history of SpaceX Falcon9 launch outcomes was analyzed in-depth in search of relationships between key launch variables
- Predictive models with identified optimal hyperparameters were generated utilizing numerous algorithms to assess predictive ability
- Results indicate predictive accuracy scores ranging from ~77-83% across models

Introduction

- SpaceX Falcon9 rocket launches cost ~\$60 million while other providers cost upward of \$165 million - much of the savings is because SpaceX can reuse the first stage. If we can determine the success of the first stage, we can determine the cost of a launch. This information can be valuable for a competitor company to bid against SpaceX for a potential rocket launch opportunity
- We seek to predict the success of first stage landings utilizing known independent launch variables

Section 1

Methodology

Methodology

Executive Summary

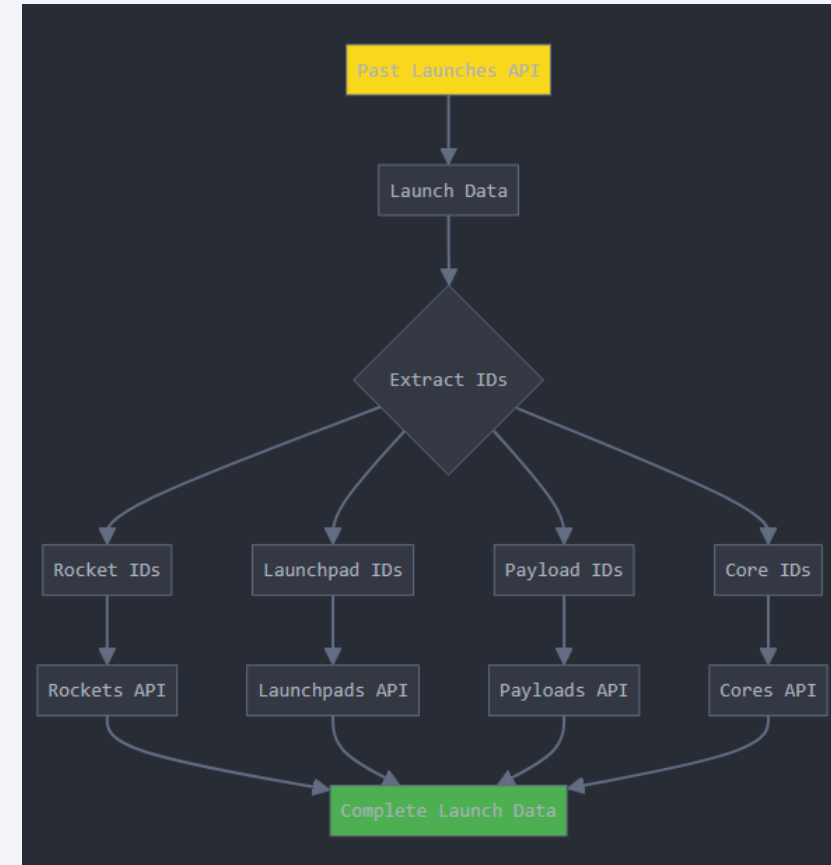
- Data collection methodology:
 - Data was obtained via SpaceX's API and webscraping Wikipedia's page on Falcon9 launch history
- Perform data wrangling
 - Data was pre-processed to address incomplete records and engineer numerical features for categorical variables
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
 - LR, SVM, Decision Tree, and KNN models were fit with optimal parameters (via GridSearch) and training data, then tested for prediction accuracy on unseen test data

Data Collection

- SpaceX's API provides several endpoints enabling access to detailed data including:
 - *Past (launches)*
 - *Rockets*
 - *Launchpads*
 - *Payloads*
 - *Cores*
- Data was first obtained from past launches then additional endpoints were used to lookup referenced variables of interest to form the new dataset that served as the basis for our analysis
- Finally, a webscraping exercise was conducted to extract launch history data from the SpaceX Falcon9 Wikipedia webpage

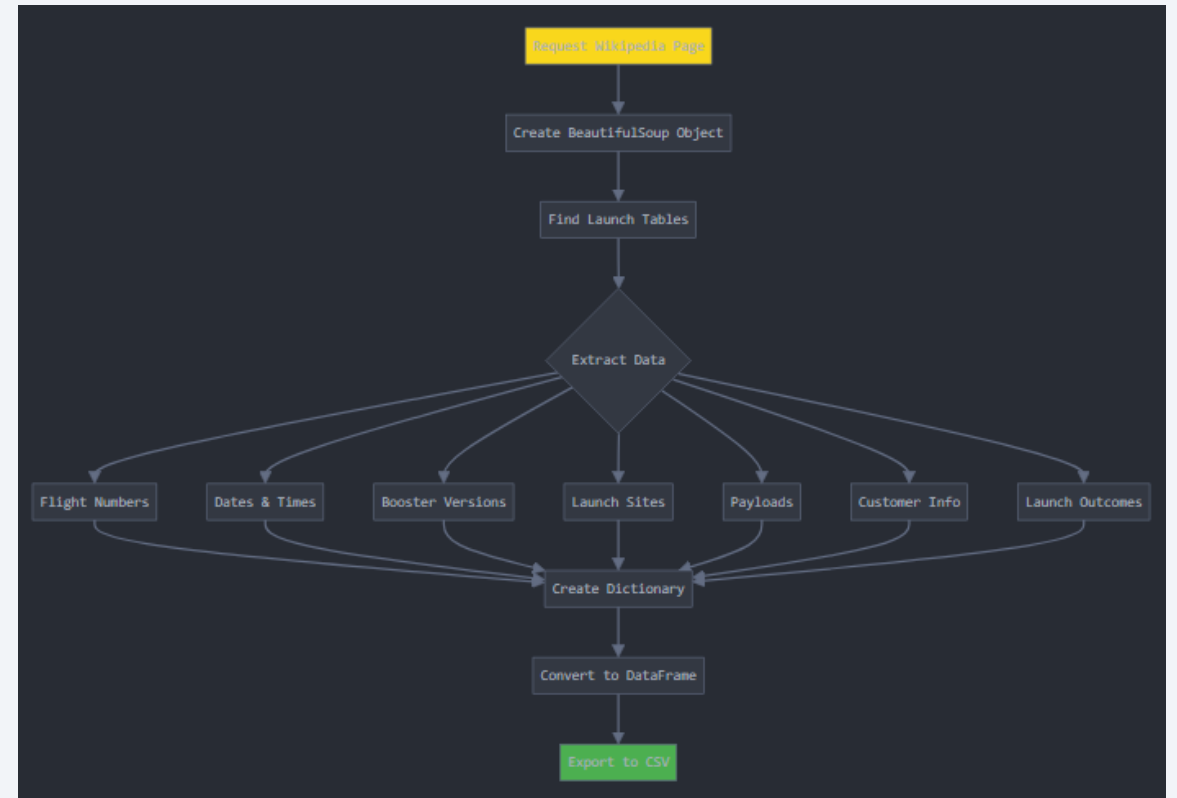
Data Collection – SpaceX API

- The following endpoints were used for data collection:
 - *Past* – primary launch data
 - *Rocket* – booster name
 - *Payload* – payload mass and target orbit
 - *Launchpad* – launch site name and coordinates
 - *Cores* – landing outcome and type, number of flights per core, additional core usage history and descriptors
- Data Collection - API



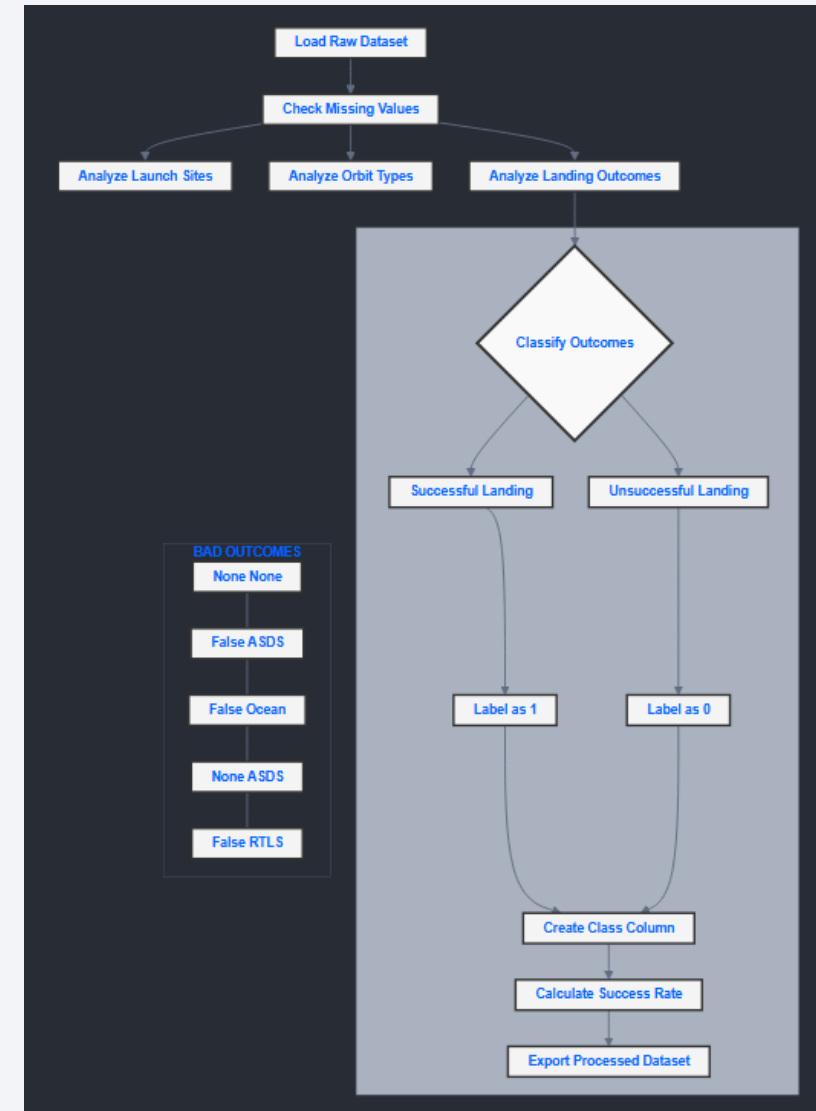
Data Collection - Scraping

- SpaceX's Falcon9 Wikipedia page was scraped of its tables, targeted variable were extracted into a python dictionary, then converted to a pandas data frame for analysis
- [Data Collection - Webscraping](#)



Data Wrangling

- Raw data was cleaned and processed utilizing feature engineering to create the target binary 'Class' series for landing outcome:
 - 1.0 – *successful landing*
 - 0.0 – *unsuccessful landing*
- One hot encoding was utilized to create dummy variables for valued categorical features
 - 'Orbit', 'LaunchSite', 'LandingPad', 'Serial'
- [Data Wrangling](#)



EDA with Data Visualization

- **Relationship Analysis Charts:** *Examined flight experience (number) and payload mass against success to understand how operational experience and mission load affect landing outcomes*
- **Launch Site Analysis Charts:** *Analyzed the distribution and performance across different launch facilities to identify site-specific patterns that might influence mission success*
- **Orbit Analysis Charts:** *Compared success rates across different orbit types to determine if certain orbital destinations present more challenges for successful landings*
- **Year-over-Year Analysis:** *Tracked landing success rates over time to assess the overall progress and maturation of SpaceX's landing technology and procedures*
- **Data Visualization**

EDA with SQL

- **Basic Data Exploration:** *Queries to identify unique launch sites and specific launch locations to understand the geographic distribution of SpaceX operations*
- **Payload Analysis:** *Calculations of total and average payload mass for specific missions and booster types to assess launch capacity*
- **Landing Success Analysis:** *Queries to track landing outcomes, including first successful ground landing date and successful drone ship landings within specific payload ranges*
- **Mission Success Metrics:** *Count of mission outcomes and identification of boosters carrying maximum payload to evaluate operational performance*
- **Time-Based Analysis:** *Examination of specific time periods to understand landing outcome distributions and historical performance patterns*
- [EDA with SQL](#)

Build an Interactive Map with Folium

- **Base Map Objects:** *Folium Map centered on NASA JSC, MousePosition tool to capture coordinates, MarkerCluster to handle overlapping markers*
- **Site Identification Objects:** *Circles around each launch site for clear visual boundaries, Text markers with site names for easy identification, Distance markers showing proximity measurements*
- **Launch Outcome Visualization:** *Color-coded markers (green/red) to show success/failure of launches, Marker clusters to handle multiple launches at same coordinates, Popup information showing launch details*
- **Distance Analysis Objects:** *PolyLines connecting sites to key points of interest, Distance markers showing calculated distances, Custom markers for geographical features*
- Each object was added to help analyze geographical patterns in SpaceX launch operations and success rates
- [Launch Sites Locations Analysis with Folium](#)

Build a Dashboard with Plotly Dash

- **Interactive Selection Components:**
 - **Site Dropdown:** *Allows users to select specific launch sites or view all sites*
 - **Payload Range Slider:** *Enables filtering of data by payload mass range*
 - *Both controls affect multiple visualizations for dynamic analysis*
- **Visualization Components:**
 - **Success Pie Chart:** *Shows success rate distribution across all sites or selected site*
 - **Scatter Plot:** *Visualizes payload-success relationship by booster version and site selection*
- The dashboard was designed to allow users to explore relationships between launch site, payload mass, and mission success rates through interactive filtering and multiple complementary views of the data
- [SpaceX Plotly Dash App](#)

Predictive Analysis (Classification)

Data Preparation & Feature Engineering:

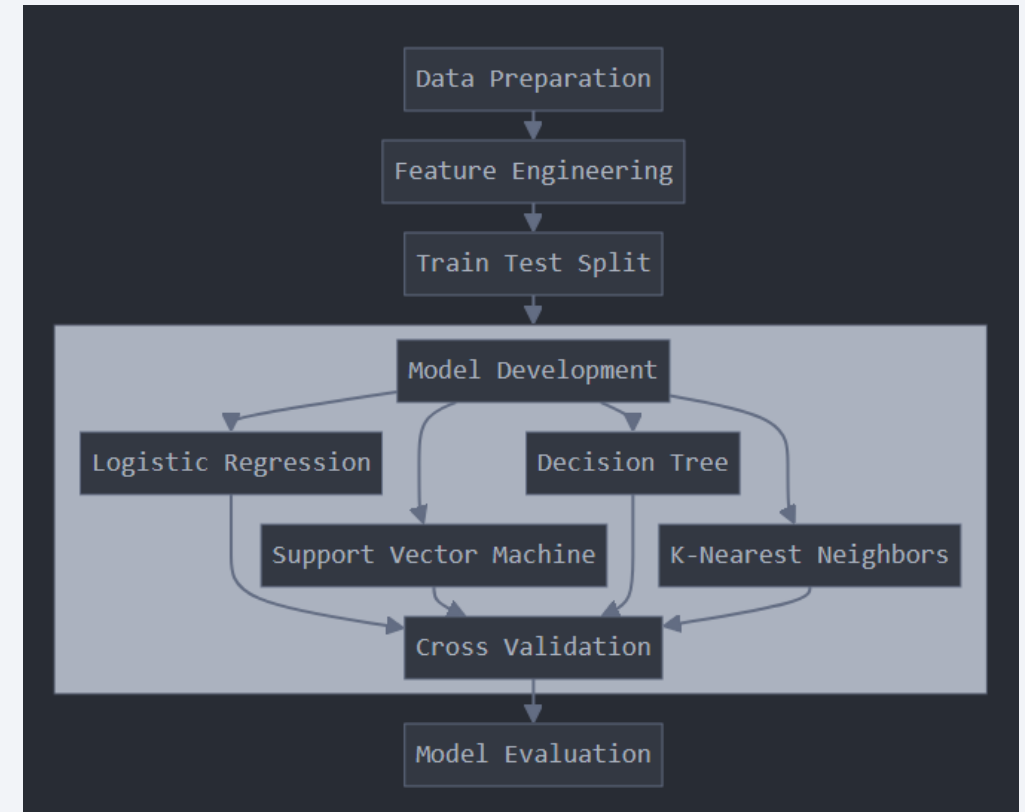
- Created target variable 'Class' indicating landing success
- Standardized features using StandardScaler
- Split data: 80% training, 20% test using TrainTestSplit

Model Development & Results (Accuracy):

- Logistic Regression - Training: 84.64% / Test: 83.33%
- Support Vector Machine - Training: 84.82% / Test: 83.33%
- Decision Tree - Training: 87.50% / Test: 77.78%
- K-Nearest Neighbors - Training: 84.82% / Test: 83.33%

Model Evaluation & Selection:

- All models tuned using GridSearchCV with 10-fold CV
- LR, SVM, and KNN showed consistent performance
- Decision Tree showed signs of overfitting
- Confusion matrices were used to evaluate prediction patterns



SpaceX Falcon9 First Stage Landing Prediction

Results

EDA Results

Launch Success Trends:

- Success rates improved 2013-2017
- Higher flight numbers = better success rates

Payload & Orbit:

- Heavy payloads (>10,000 kg) absent from VAFB-SLC
- LEO missions correlated with flight number
- GTO showed no clear patterns

Launch Sites:

- Different sites specialized in specific orbits/payloads
- Success rates varied by location

Payload Mass Impact:

- Heavier loads generally meant lower success
- Exception: Polar, LEO, and ISS orbits maintained success with heavy loads

Predictive Results

Logistic Regression:

- Training accuracy: 0.846 / Test accuracy: 0.833
- Best params: C=0.01, L2 penalty, lbfgs solver

SVM:

- Training accuracy: 0.848 / Test accuracy: 0.833
- Best params: C=1.0, gamma=0.032, sigmoid kernel

Decision Tree:

- Training accuracy: 0.875 / Test accuracy: 0.778
- Best params: entropy criterion, max_depth=8, random splitter

KNN:

- Training accuracy: 0.848 / Test accuracy: 0.833
- Best params: n_neighbors=10, auto algorithm, p=1

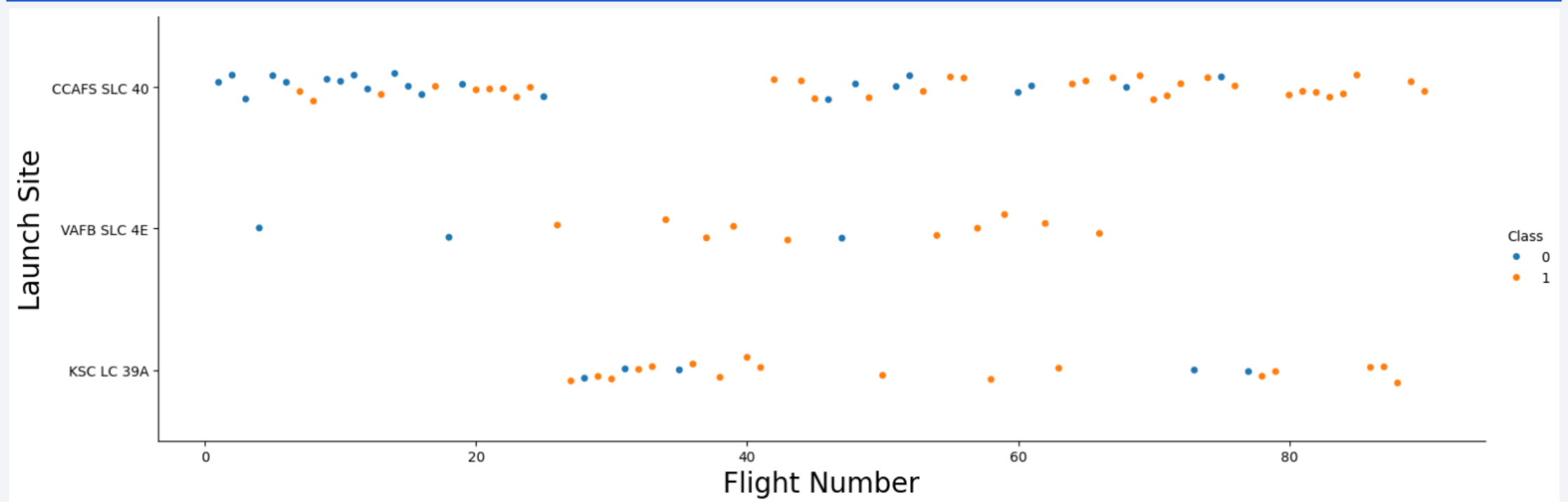
***Best Performance:** Decision Tree had the highest training accuracy but showed signs of overfitting with lower test accuracy. LR, SVM, and KNN performed equally well on test data, with simpler models and better generalization*

The background of the slide is an abstract composition. It features a dark blue base color. Overlaid on this are numerous diagonal streaks in shades of red and cyan. A faint, light blue grid pattern is also visible, particularly in the lower half of the image. The overall effect is dynamic and technological.

Section 2

Insights drawn from EDA

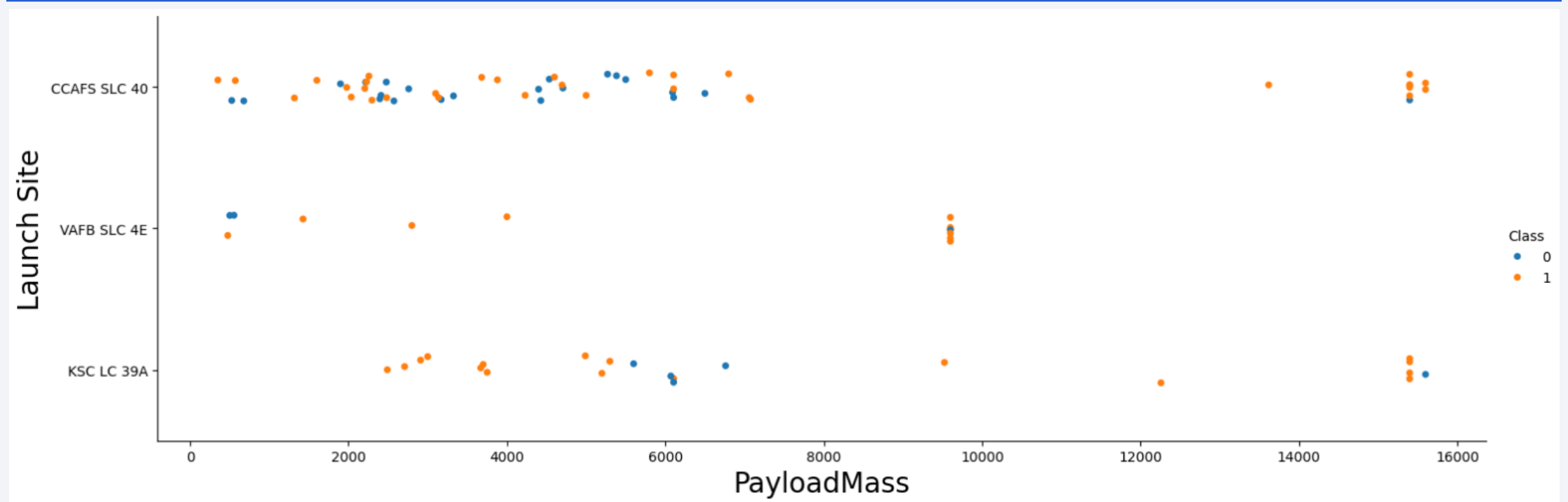
Flight Number vs. Launch Site



The Launch Site does not appear to directly influence the landing outcome

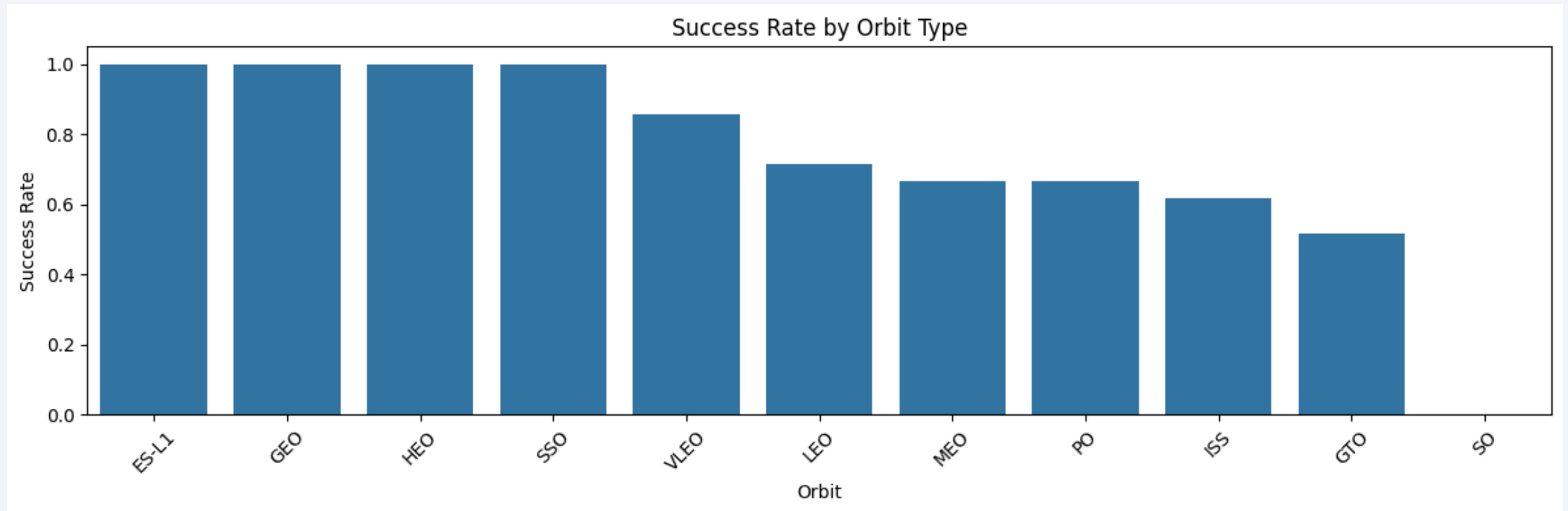
The more significant driver of outcome appears to be Flight Number where the likelihood of success increased over time with more flight experience

Payload vs. Launch Site



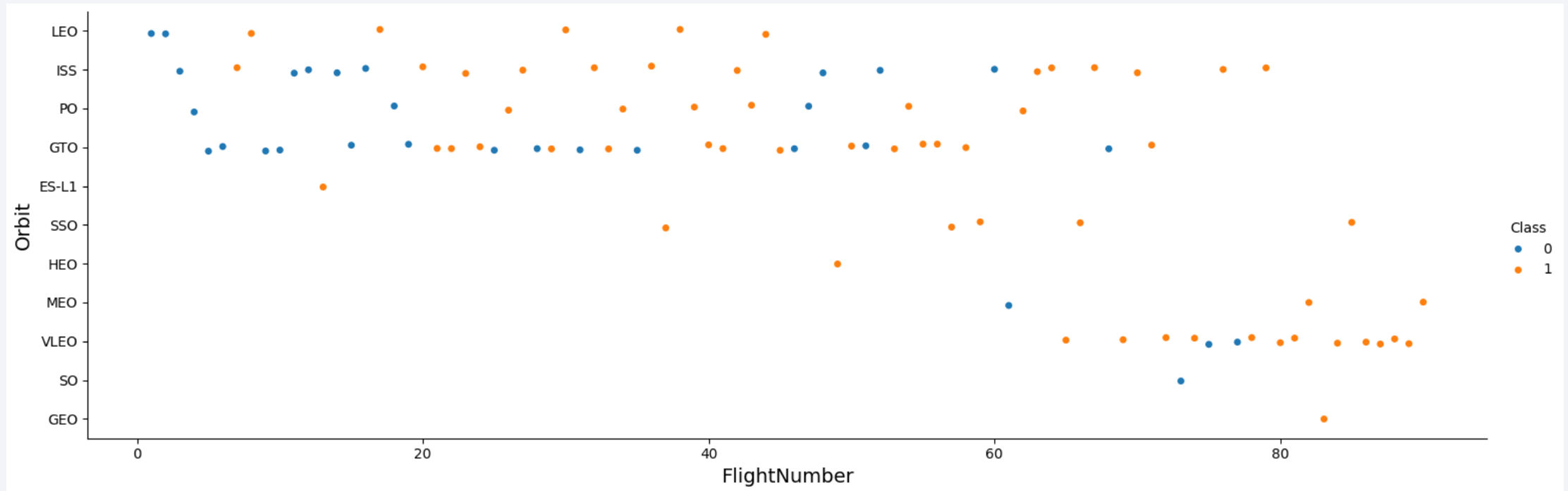
Payloads larger than 10,000 kg were not launched from site VAFB SLC 4E

Success Rate vs. Orbit Type



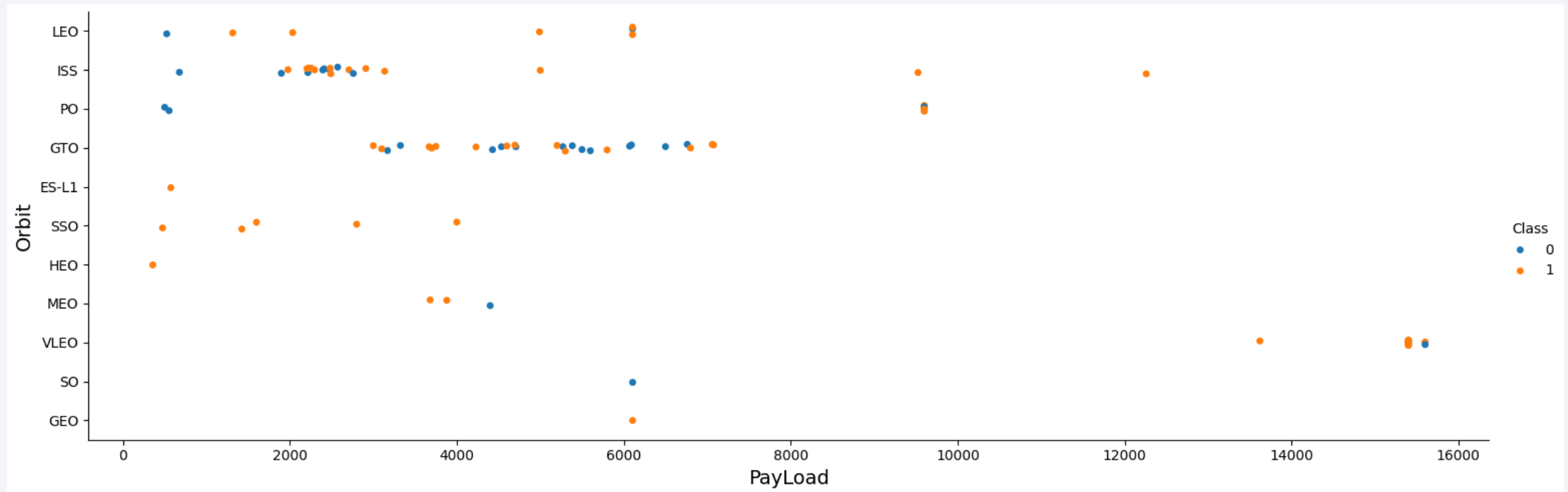
GTO orbit appears to be the most difficult to achieve a successful outcome

Flight Number vs. Orbit Type



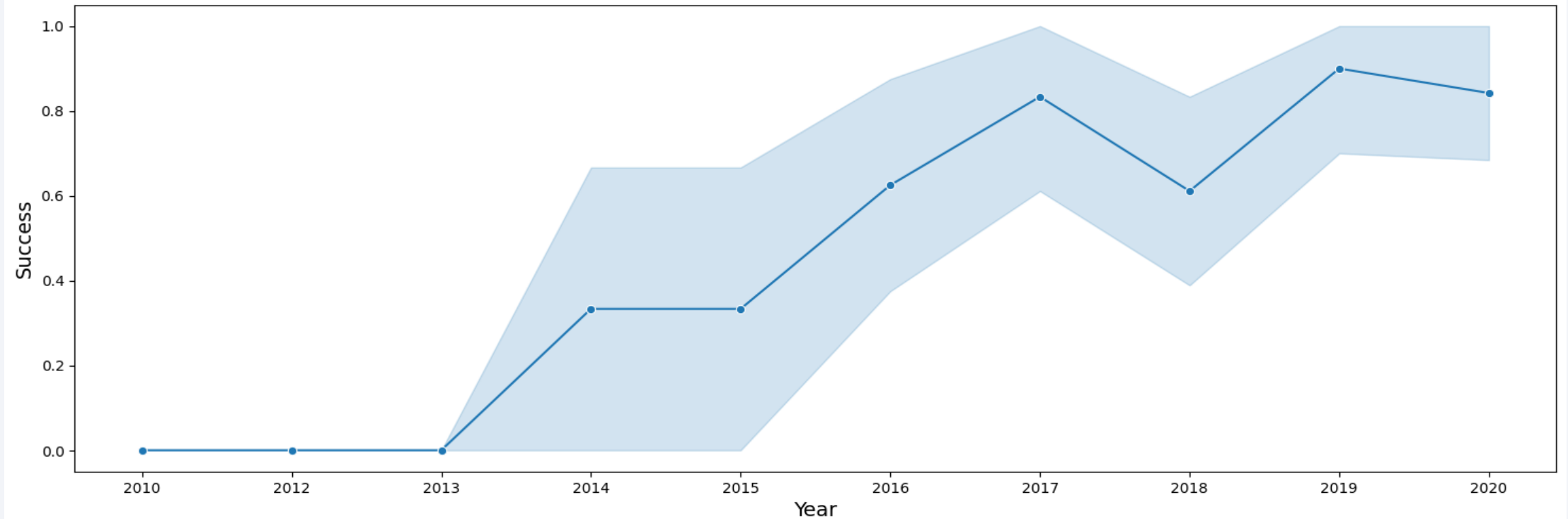
Most earlier flights targeted the LEO, ISS, PO, and GTO orbits, while later flights targeted VLEO

Payload vs. Orbit Type



Most payloads were less than 10,000 kg for all orbits apart from VLEO

Launch Success Yearly Trend



It took a few years! ... but SpaceX success grew significantly in 2014, then again in 2016-2017 before plateauing around 80%

All Launch Site Names

```
1 %sql SELECT DISTINCT Launch_Site FROM SPACEXTABLE ORDER BY Launch_Site;
```

Python

```
* sqlite:///my\_data1.db  
Done.
```

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

Unique Launch Site names were obtained using SQL's SELECT DISTINCT command

Launch Site Names Begin with 'CCA'

```
1 %sql SELECT * FROM SPACEXTABLE WHERE Launch_Site LIKE 'CCA%' LIMIT 5;
```

Python

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

Done.

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

We can SELECT 5 records from a table where the Launch Site name starts with 'CCA' by utilizing the LIKE and LIMIT commands

Total Payload Mass

```
1 %sql SELECT SUM(PAYLOAD_MASS_KG_) as 'Total Payload Mass' FROM SPACEXTABLE WHERE Customer LIKE 'NASA (CRS)'
```

Python

```
* sqlite:///my\_data1.db
```

Done.

Total Payload Mass

45596

We can find the Total Payload Mass for NASA by utilizing the SUM function on the PAYLOAD_MASS_KG column filtered by Customer 'NASA (CRS)'

Average Payload Mass by F9 v1.1

```
1 %sql SELECT AVG(PAYLOAD_MASS_KG_) as 'Average Payload Mass' FROM SPACEXTABLE WHERE Booster_Version LIKE 'F9 v1.1'
```

Python

```
* sqlite:///my\_data1.db
```

Done.

Average Payload Mass

2928.4

We can find the Average Payload Mass for Booster F9 v1.1 by utilizing the AVG function on the PAYLOAD_MASS_KG column filtered by Booster 'F9 v1.1'

First Successful Ground Landing Date

```
1 %sql SELECT MIN(Date) as 'First Success' FROM SPACEXTABLE WHERE Landing_Outcome LIKE 'Success (ground pad)'
```

Python

```
* sqlite:///my\_data1.db
```

```
Done.
```

```
First Success
```

```
2015-12-22
```

We can find the First Successful Ground Landing Date by utilizing the MIN function on the Date column filtered by Landing Outcome 'Success (ground pad)'

Successful Drone Ship Landing with Payload between 4000 and 6000

```
1 %sql SELECT Booster_Version FROM SPACEXTABLE WHERE Landing_outcome = 'Success (drone ship)' AND PAYLOAD_MASS_KG_ > 4000 AND PAYLOAD_MASS_KG_ < 6000
* sqlite:///my_data1.db
Done.
```

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

We can find the Boosters with successful drone ship landings and payloads between 4,000 to 6,000 kg by utilizing the SELECT command on the Booster_Version column filtered by Landing Outcome 'Success (drone ship)' and PAYLOAD_MASS_KG filtered between our targeted sizes

Total Number of Successful and Failure Mission Outcomes

```
1 %sql SELECT Mission_Outcome, Count(*) FROM SPACEXTABLE GROUP BY Mission_outcome
```

Python

* [sqlite:///my_data1.db](#)
Done.

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

We can find the Total Number of Successful and Failed Mission Outcomes by utilizing the GROUP BY function on the Mission_outcome column and displaying the COUNT of each outcome

Boosters Carried Maximum Payload

```
1 %sql SELECT DISTINCT Booster_version, PAYLOAD_MASS_KG_ FROM SPACEXTABLE WHERE PAYLOAD_MASS_KG_ = (SELECT MAX(PAYLOAD_MASS_KG_) FROM SPACEXTABLE);
```

Python


* [sqlite:///my_data1.db](#)
Done.

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

*We can identify all Boosters that carried the maximum payload by utilizing the **SELECT DISTINCT** command on column **Booster_version** and filtering **PAYLOAD_MASS_KG** to the highest observed value by utilizing the **MAX** function on that column*

2015 Launch Records

```
1 %%sql
2 SELECT CASE substr(Date, 6, 2) WHEN '01' THEN 'January' WHEN '02' THEN 'February' WHEN '03' THEN 'March' WHEN '04' THEN 'April' END as Month,
3 Booster_Version, Launch_Site, Landing_Outcome FROM SPACEXTABLE
4 WHERE substr(Date, 0, 5) = '2015' AND Landing_Outcome LIKE '%failure%drone ship%'
5 ORDER BY substr(Date, 6, 2);
```

 Python

* [sqlite:///my_data1.db](#)

Done.

Month	Booster_Version	Launch_Site	Landing_Outcome
January	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
April	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

We can identify all failed drone ship landing outcomes in 2015 as seen in the query above

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

```
1 %%sql
2 SELECT Landing_outcome, COUNT(*) as Count FROM SPACEXTABLE WHERE Date > '2010-06-04' and Date < '2017-03-20'
3 GROUP BY Landing_outcome ORDER BY Count DESC
```

Python

* [sqlite:///my_data1.db](#)

Done.

Landing_Outcome	Count
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

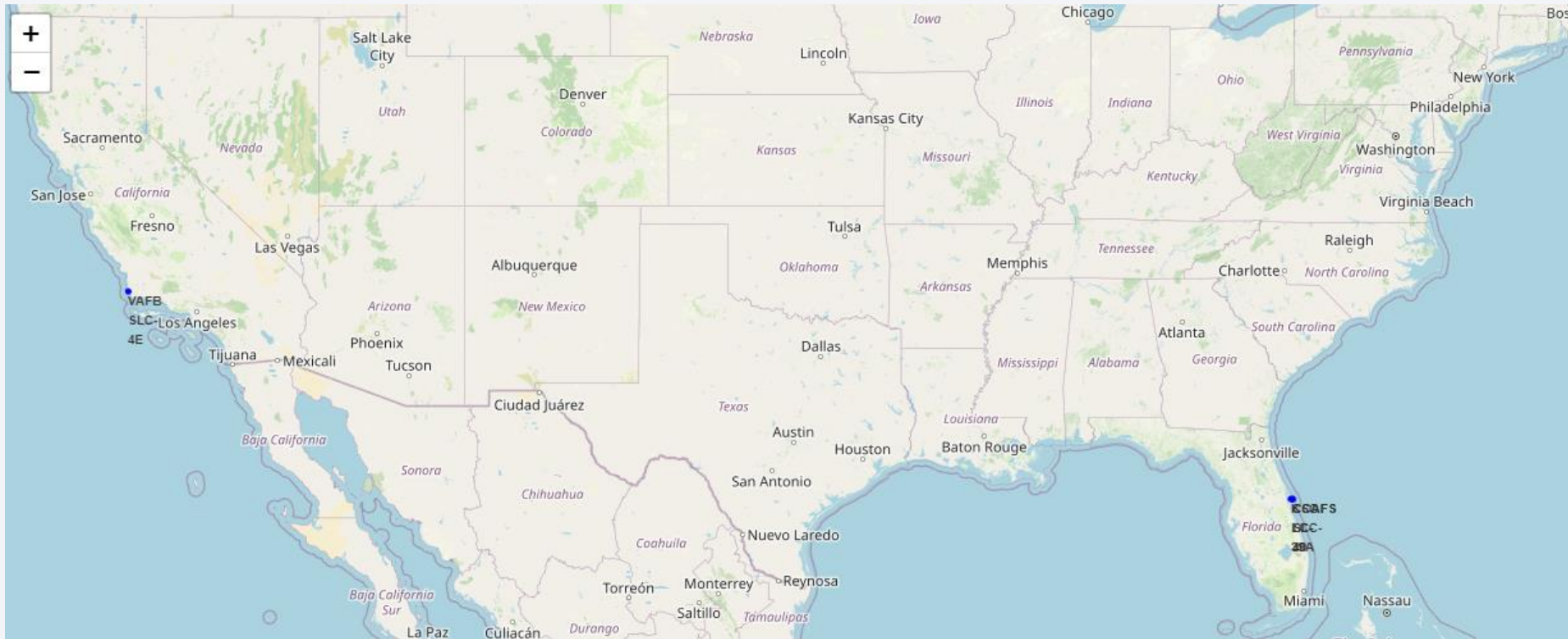
We can identify the number of all landing outcome types between specific dates as seen in the query above

A satellite view of Earth from space, showing the curvature of the planet and the glowing lights of cities at night. The background is a deep blue gradient.

Section 3

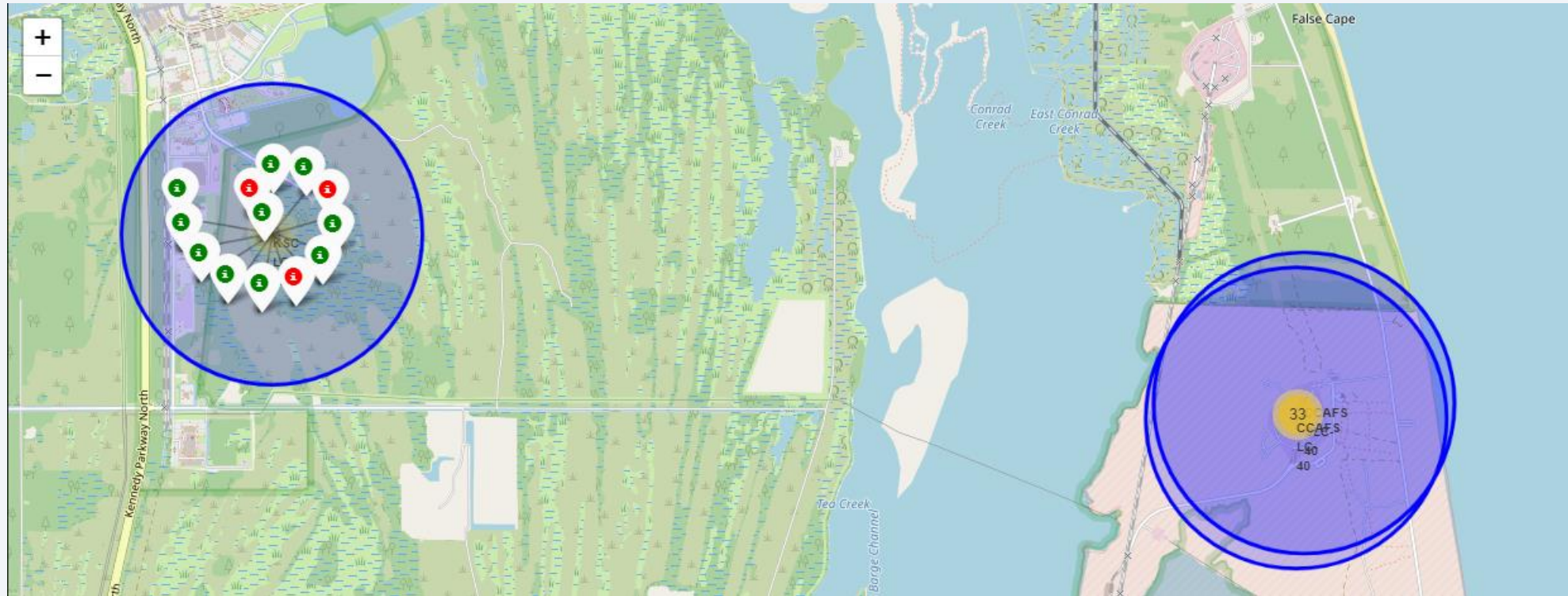
Launch Sites Proximities Analysis

SpaceX Launch Sites



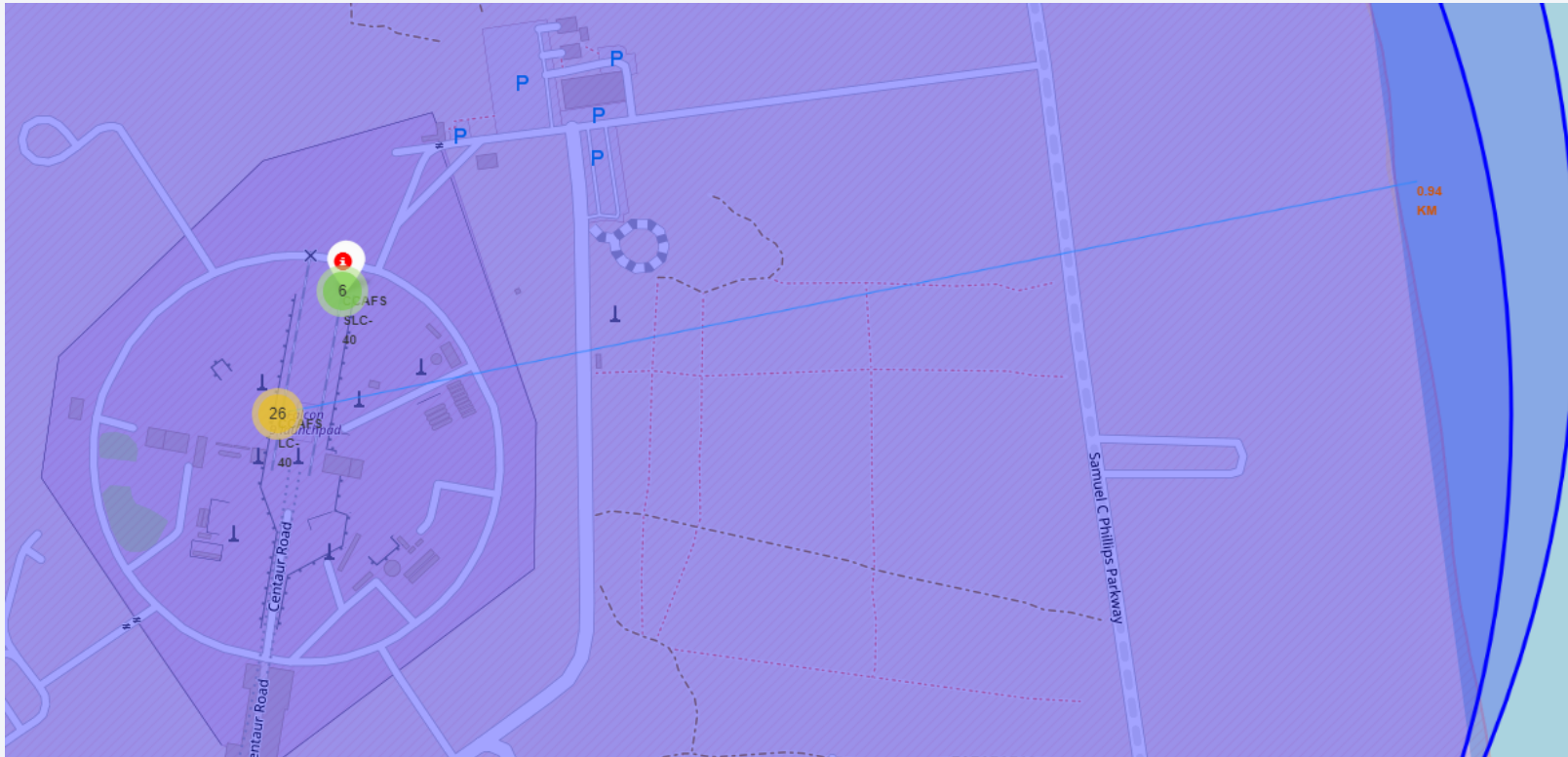
We can locate and label all SpaceX Launch Sites utilizing Folium and the latitude and longitude coordinates for each site

Successful and Failed Landing Markers by Site



We can create colored markers by engineering a new column 'marker_color' as a function of 'Class' where green = successful and red = failure ... the markers can be placed on the Folium map as shown above

Distance between CCAFS SLC-40 and the coastline



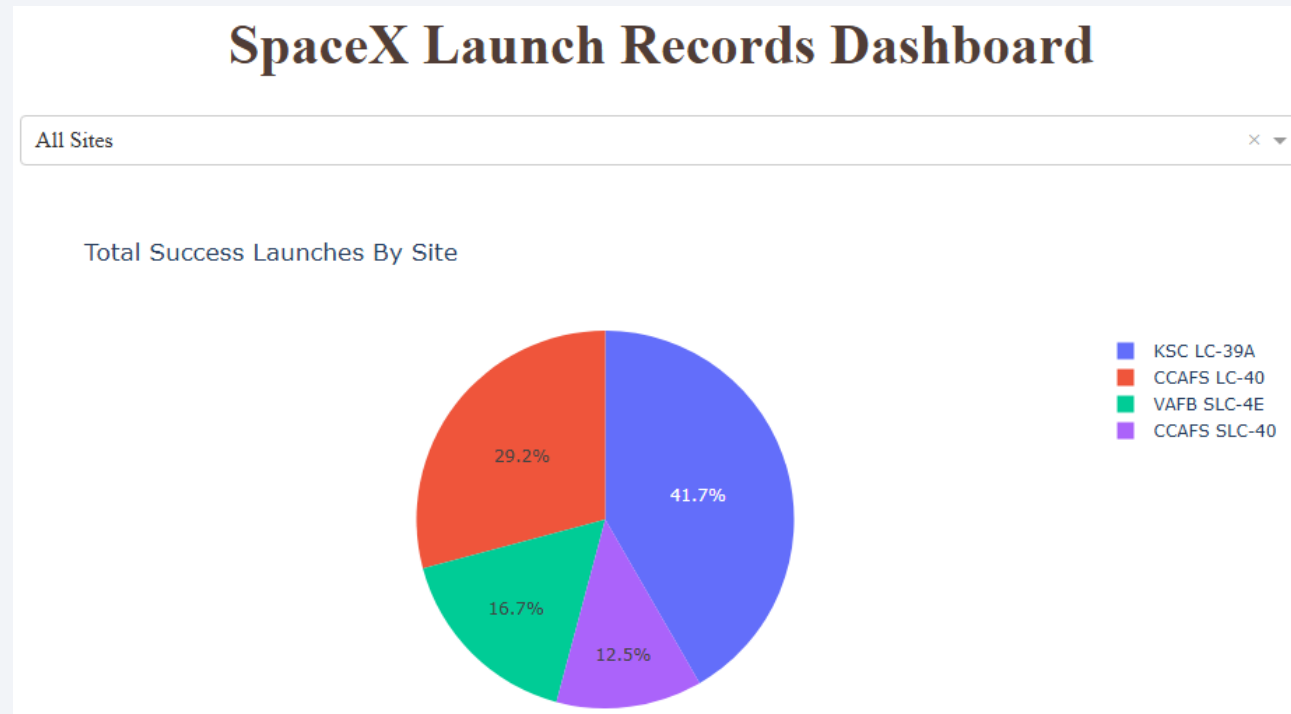
Folium allows for the functionality to draw lines between locations such as that between a Launch Site and the coastline



Section 4

Build a Dashboard with Plotly Dash

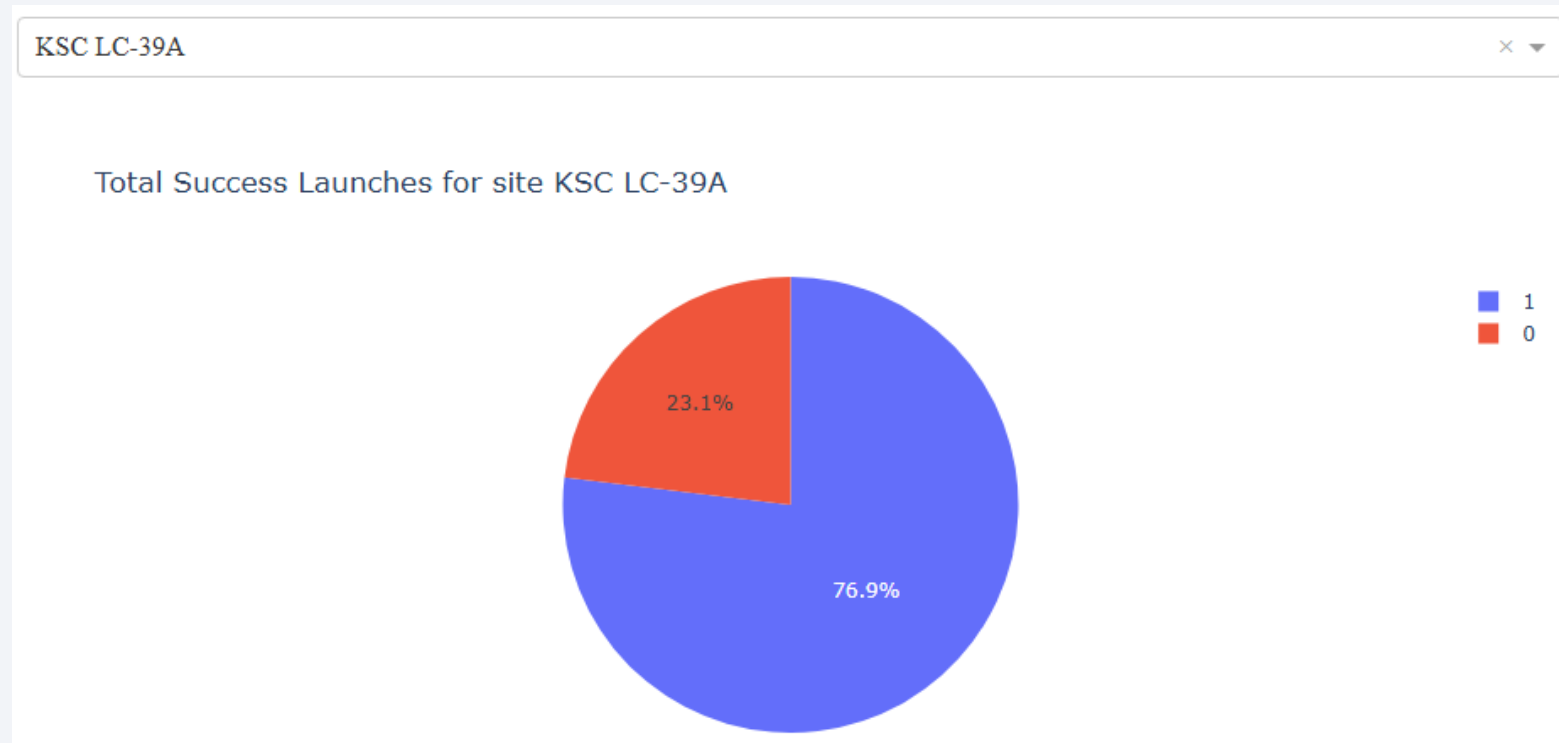
Total Successful Landings for All Sites



Plotly Dash dashboard showing overall successful landings across all sites

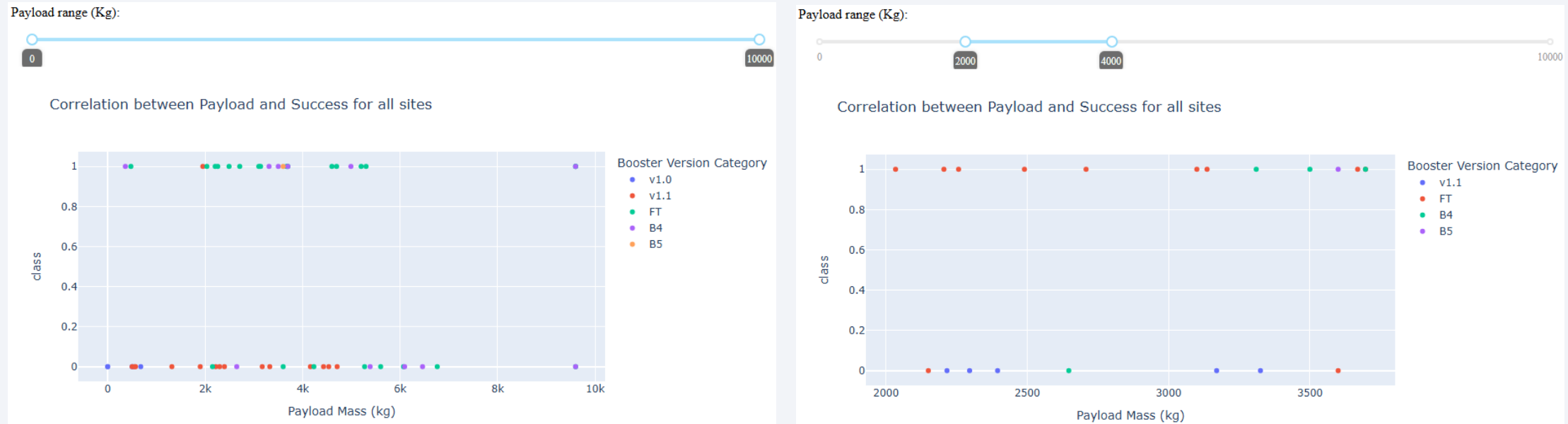
We can see that ~70% of successful landings have come from two sites: KSC LC-39A and CCAFS LC-40

KSC LC-39A has been the most successful site ...



Highest percentage of successful landings have come from the KSC LC-39A site

Relationship between Payload, Booster, and Success



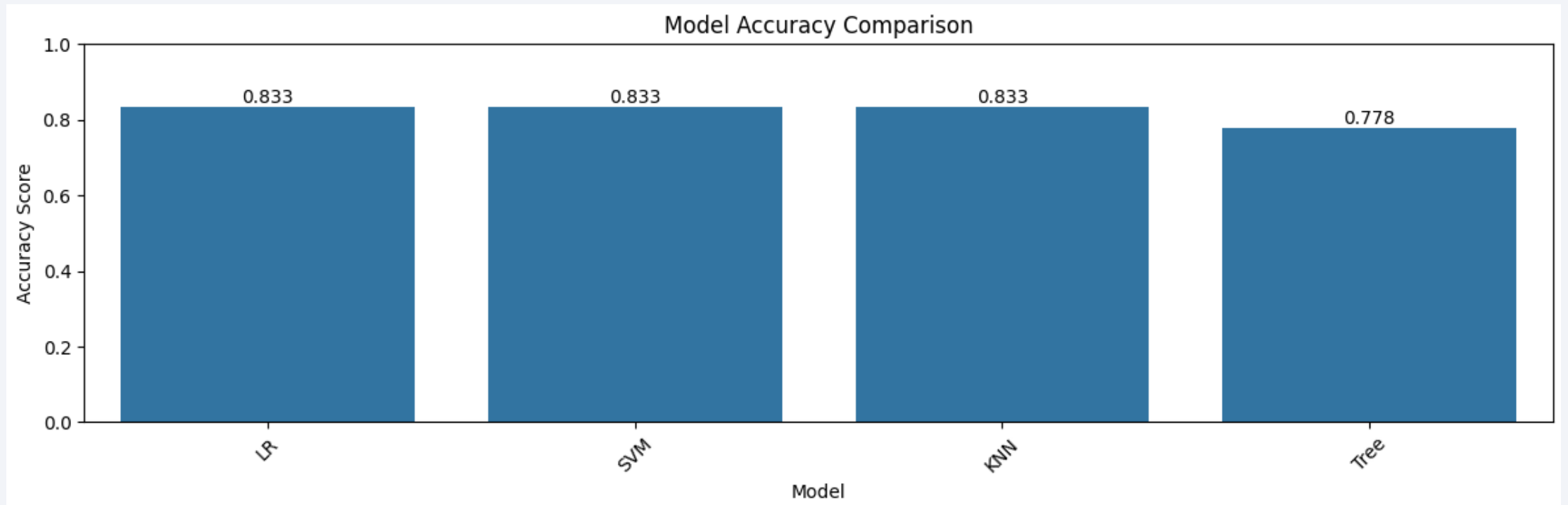
We can see that most successful landings have been with a payload mass between ~2,000-4,000 kg

Narrowing our focus on that range, we see that most of those launches have been with Booster Version FT

Section 5

Predictive Analysis (Classification)

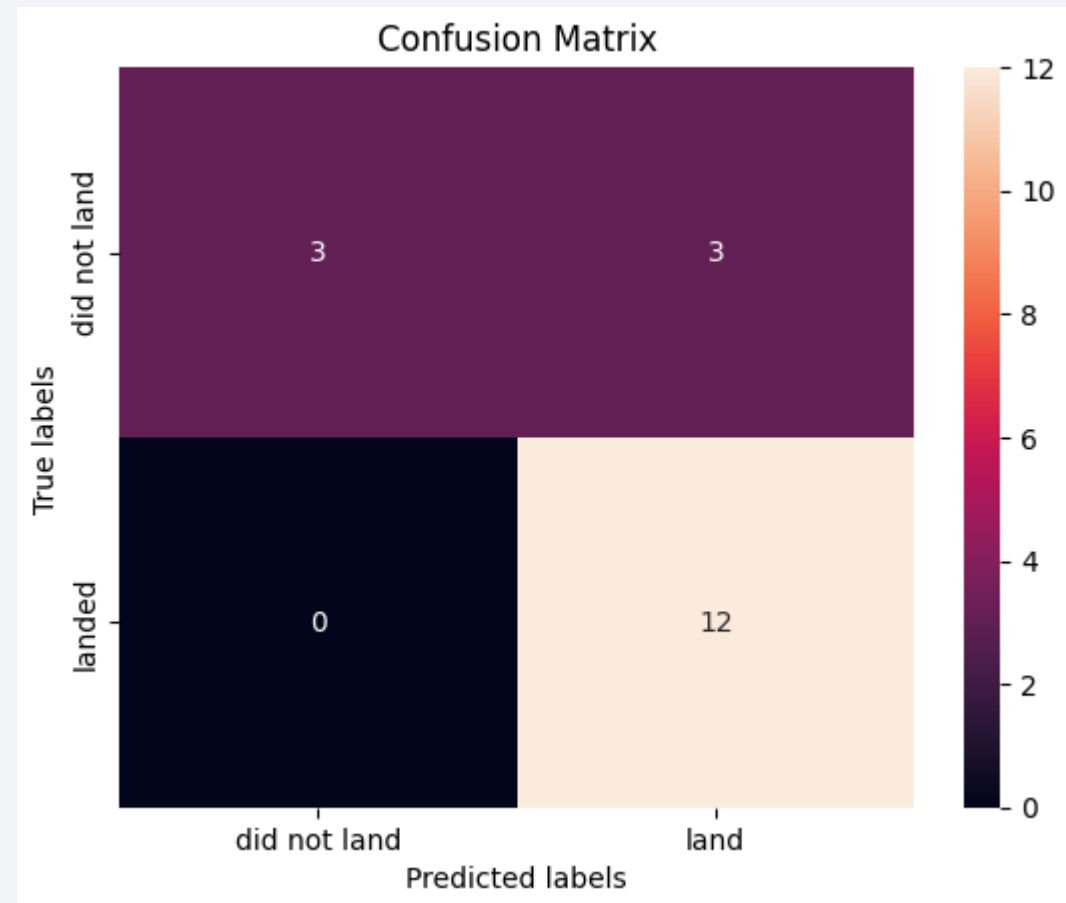
Classification Accuracy



LR, SVM, and KNN all have the same accuracy on the Test set

Confusion Matrix

- *LR, SVM, and KNN all have the same prediction accuracy and confusion matrix*
- *The models were able to predict every successful landing (12/12) ...*
- *However, each was only able to predict half of the failed landings correctly (3/6)*



Conclusions

- SpaceX has significantly improved their rate of successful Falcon9 first stage landings from 0% in the early years of 2010-2013 to ~80% at the conclusion of this analysis around 2020
- SpaceX has been able to achieve successful landings at multiple launch sites, though ~70% of success has come from two sites - both in Florida
- While initial success was focused on the LEO orbit, more recent success has been achieved in the VLEO orbit
- Similarly, initial attempted payloads were smaller and generally around 5,000 kg and below; but with experience, SpaceX has been regularly successful delivering payloads of more than 15,000 kg

Thank you!

