Data Science Capstone Project

Javier Frade

18/12/2023

Outline

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

Executive Summary

- I gathered data from both the public SpaceX API and the SpaceX Wikipedia page, creating a 'class' column to categorize successful landings. Using SQL, visualization techniques, Folium maps, and dashboards, I delved into the data. After selecting relevant columns as features, I converted categorical variables to binary through one-hot encoding. Standardizing the data, I employed GridSearchCV to optimize machine learning models' parameters and visualized their accuracy scores.
- Four machine learning models—Logistic Regression, Support Vector Machine, Decision Tree Classifier, and K Nearest Neighbors—were developed, yielding similar results with an accuracy rate of approximately 83.33%. Notably, all models tended to overpredict successful landings. To enhance model accuracy and determination, additional data is essential.

Introduction

SpaceX Falcon 9 Rocket – The Verge

Background:

The commercial space age is in full swing, with SpaceX leading the charge by offering the best pricing at $62 million compared to $165 million USD from competitors. SpaceX's cost advantage is largely attributed to its capability to recover a significant portion of the rocket, specifically Stage 1. In a bid to compete with SpaceX, Space Y has enlisted our help to develop a machine learning model that can predict the success of Stage 1 recovery.

Problem:

Our mission is to train a machine learning model that can accurately forecast the outcome of Stage 1 recovery for Space Y. This involves leveraging data and insights to build a predictive model that aligns with Space Y's goal of enhancing their competitiveness in the evolving landscape of commercial space exploration.

Methodology

- Data collection methodology:
  - Combined data from SpaceX public API and SpaceX Wikipedia page
- Perform data wrangling
  - Classifying true landings as successful and unsuccessful otherwise
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Tuned models using GridSearchCV

Methodology

OVERVIEW OF DATACOLLECTION, WRANGLING, VISUALIZATION, DASHBOARD, AND MODEL METHODS

Data Collection Overview

The process of gathering data involved a dual approach, combining API requests from SpaceX's public API with the extraction of information through web scraping from a designated table within SpaceX's Wikipedia entry.

The upcoming slide will present the flowchart detailing the data collection process from the API, followed by another slide illustrating the flowchart specific to web scraping.

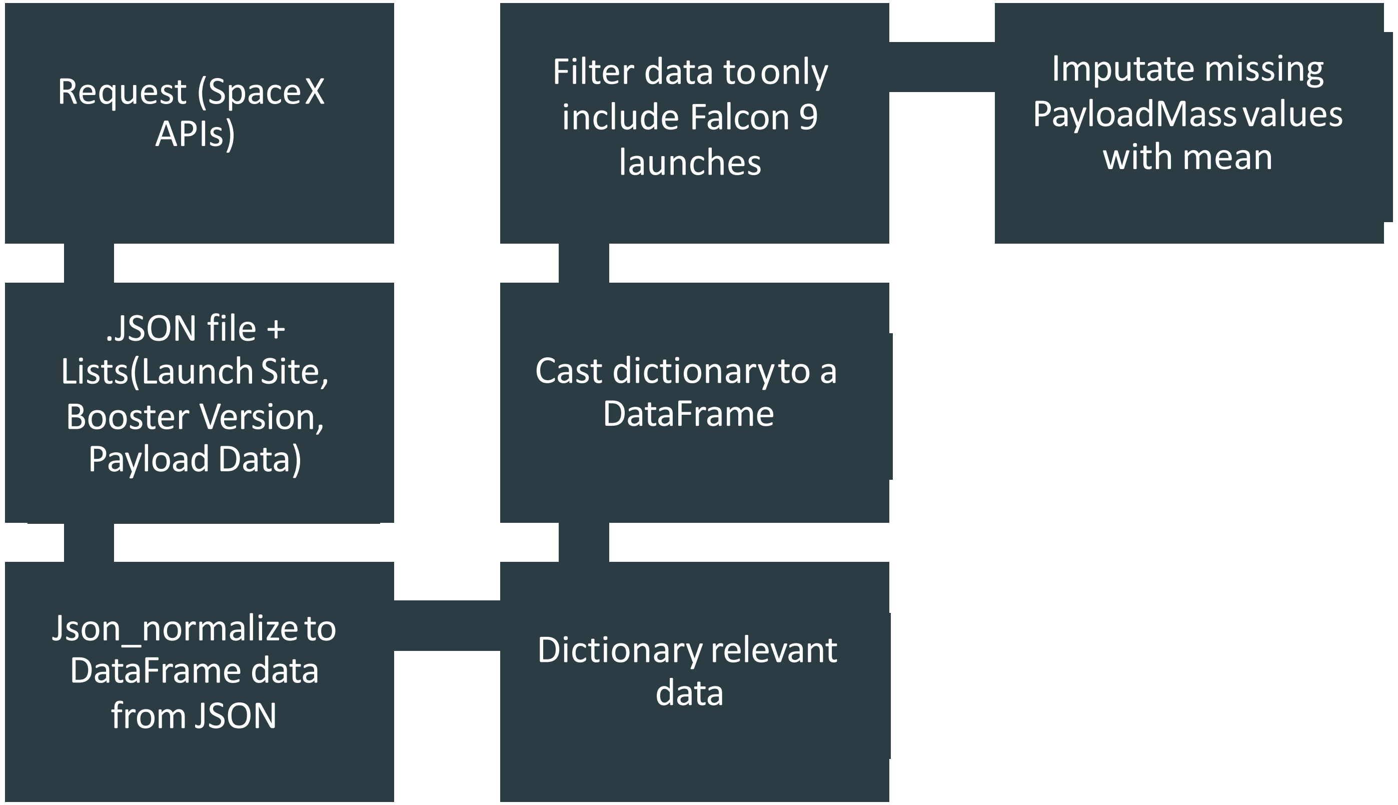
Space X API Data Columns:

Key columns extracted from the SpaceX API encompass FlightNumber, Date, BoosterVersion, PayloadMass, Orbit, LaunchSite, Outcome, Flights, GridFins, Reused, Legs, LandingPad, Block, ReusedCount, Serial, Longitude, and Latitude.

Wikipedia Webscrape Data Columns:

Simultaneously, data obtained through Wikipedia web scraping includes columns such as Flight No., Launch site, Payload, PayloadMass, Orbit, Customer, Launch outcome, Version Booster, Booster landing, Date, and Time.

Data Collection –SpaceX API



Request (SpaceX APIs)

.JSON file + Lists(Launch Site, Booster Version, Payload Data)

Json_normalize to DataFrame data from JSON

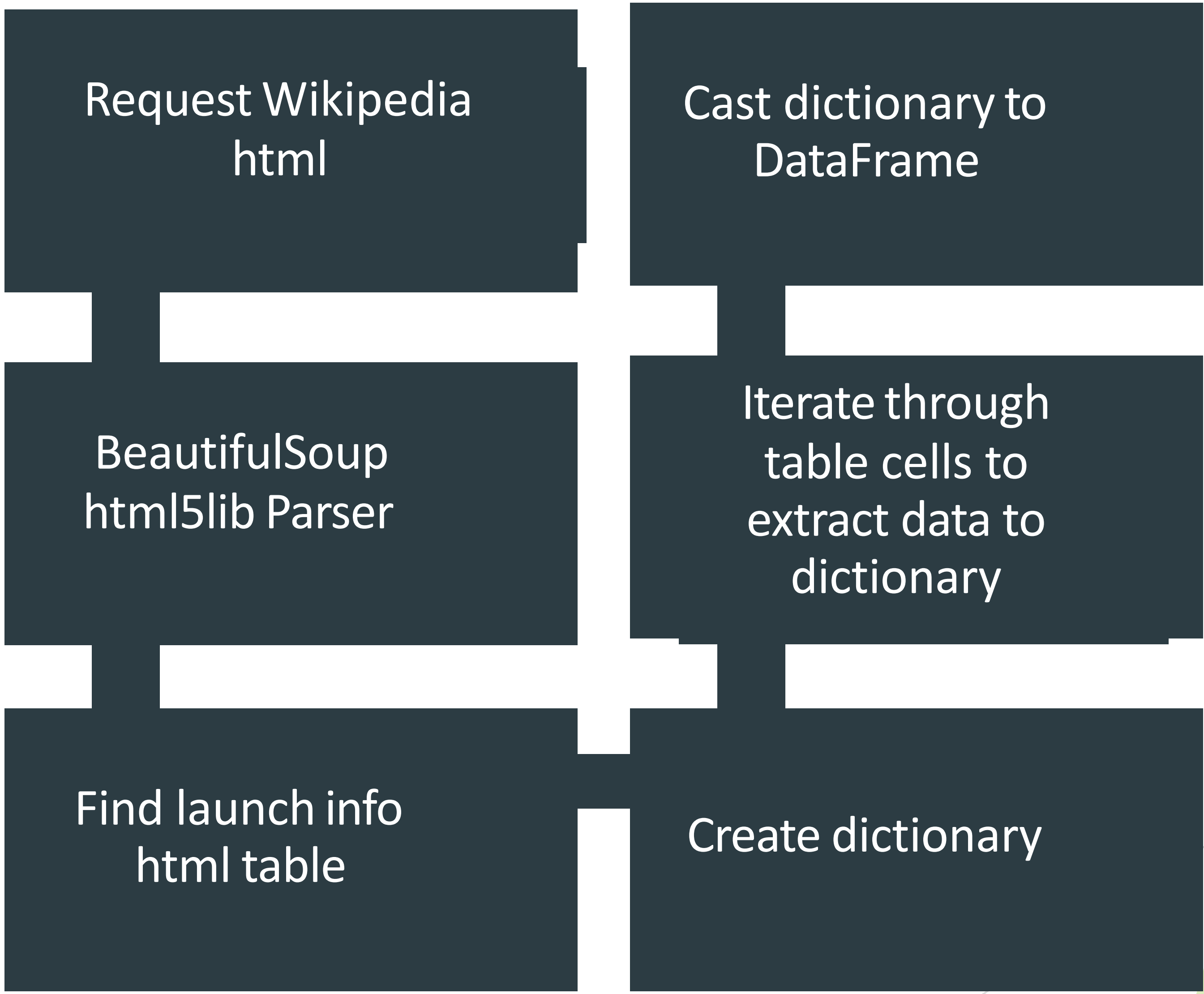
Dictionary relevant data

Cast dictionary to a DataFrame

Filter data to only include Falcon 9 launches

Imputate missing PayloadMass values with mean

Data Collection –Web Scraping



Request Wikipedia html

BeautifulSoup html5lib Parser

Find launch info html table

Create dictionary

Iterate through table cells to extract data to dictionary

Cast dictionary to DataFrame

Data Wrangling

- Create a training label with landing outcomes where successful = 1 & failure = 0.
- Outcome column has two components: ‘Mission Outcome’ ‘Landing Location’
- New training label column ‘class’ with a value of 1 if ‘Mission Outcome’ is True and 0 otherwise. Value

Mapping:

- True ASDS, True RTLS, & True Ocean – set to -> 1
- None None, False ASDS, None ASDS, False Ocean, False RTLS – set to -> 0

EDAwith Data Visualization

Exploratory Data Analysis performed on variables Flight Number, Payload Mass, Launch Site, Orbit, Class and Year.

<u>Plots Used:</u>

Flight Number vs. Payload Mass, Flight Number vs. Launch Site, Payload Mass vs. Launch Site, Orbit vs. Success Rate, Flight Number vs. Orbit, Payload vs Orbit, and Success Yearly Trend

Scatter plots, line charts, and bar plots were used to compare relationships between variables to decide if a relationship exists so that they could be used in training the machine learning model

EDA with SQL

Loaded data set into IBM DB2 Database.

Queried using SQL Python integration.

Queries were made to get a better understanding of the dataset.

Queried information about launch site names, mission outcomes, various pay load sizes of customers and booster versions, and landing outcomes

Build an interactive map with Folium

Folium maps mark Launch Sites, successful and unsuccessful landings, and a proximity example to key locations: Railway, Highway, Coast, and City.

This allows us to understand why launch sites may be located where they are. Also visualizes successful landings relative to location.

Build a Dashboard with Plotly Dash

Dashboard includes a pie chart and a scatter plot.

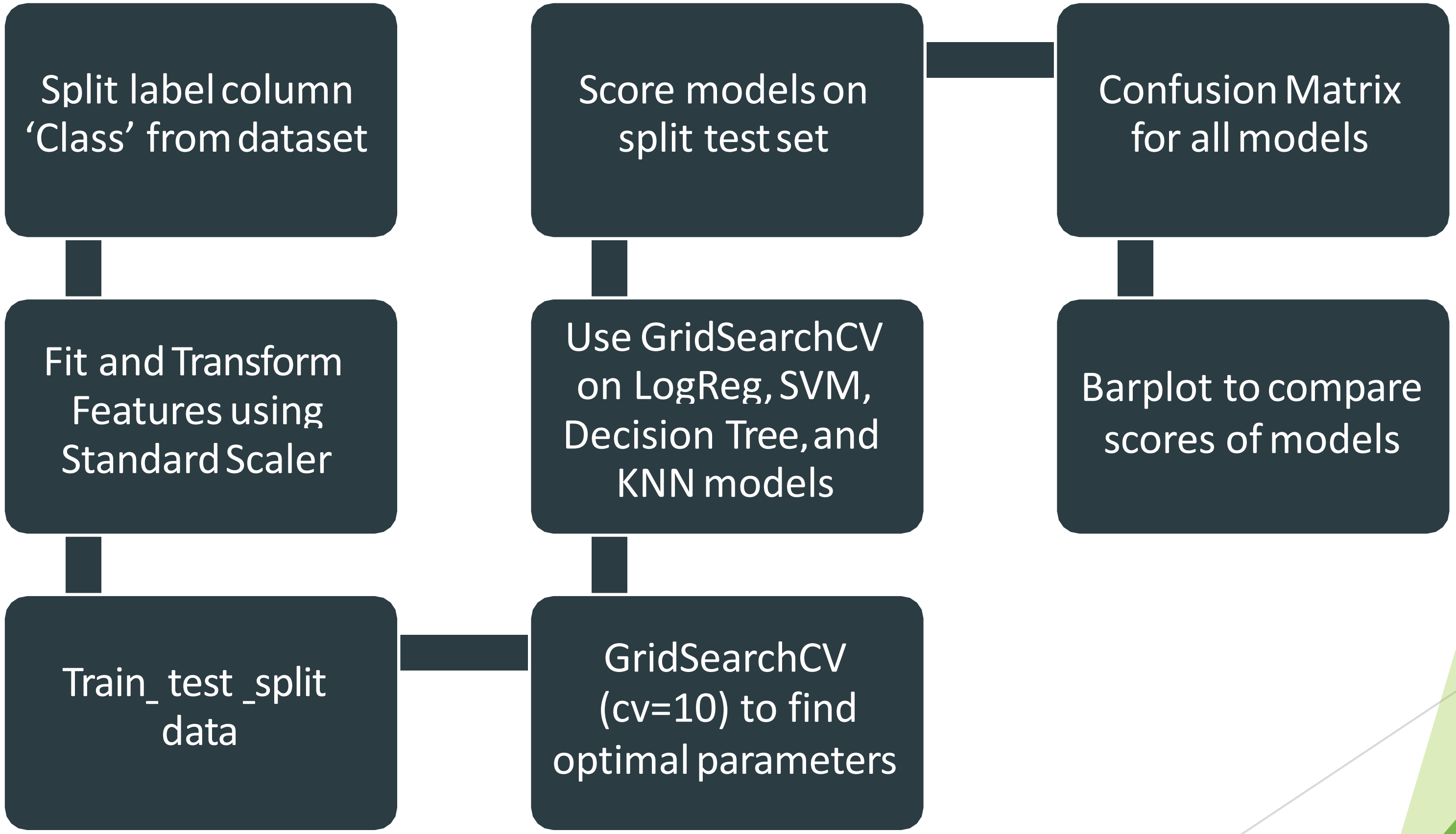
Pie chart can be selected to show distribution of successful landings across all launch sites and can be selected to show individual launch site success rates.

Scatter plot takes two inputs: All sites or individual site and payload mass on a slider between 0 and 10000 kg.

The pie chart is used to visualize launch site success rate.

The scatter plot can help us see how success varies across launch sites, payload mass, and booster version category.

Predictive analysis (Classification)



Split label column ‘Class’ from dataset

Fit and Transform Features using Standard Scaler

Train_ test _split data

GridSearchCV (cv=10) to find optimal parameters

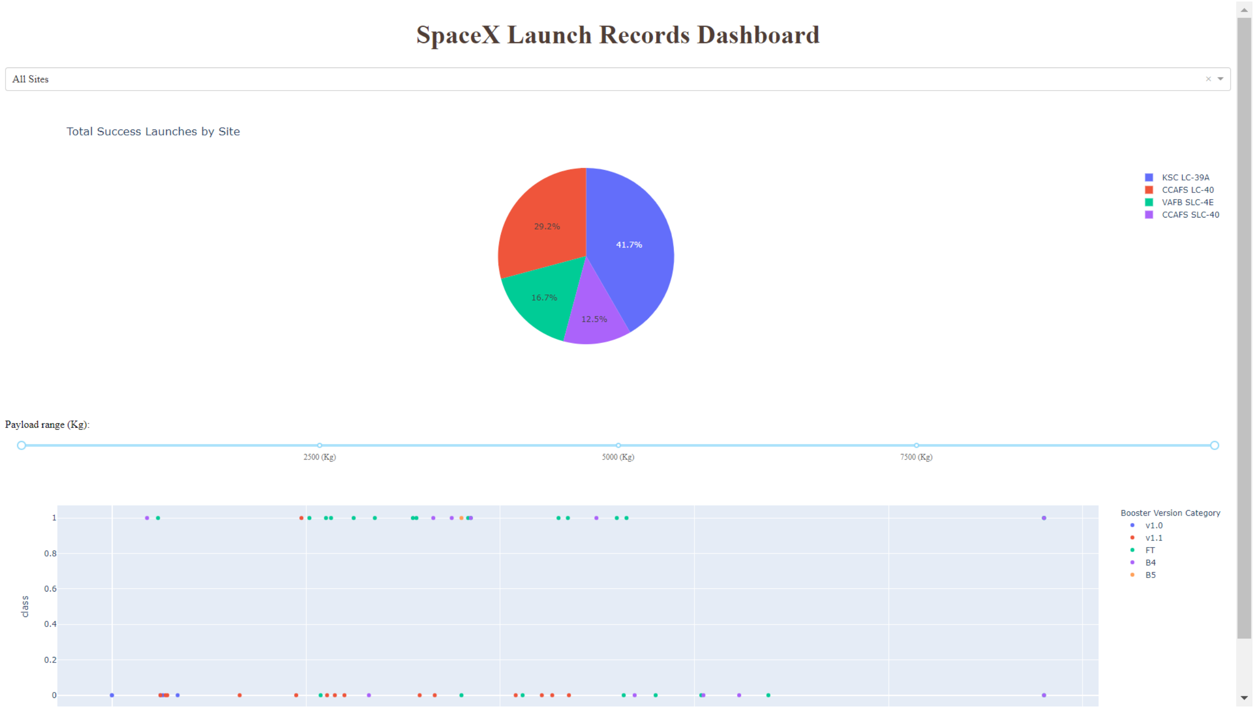
Use GridSearchCV on LogReg, SVM, Decision Tree, and KNN models

Score models on split test set

Confusion Matrix for all models

Barplot to compare scores of models

Results



SpaceX Launch Records Dashboard

All Sites

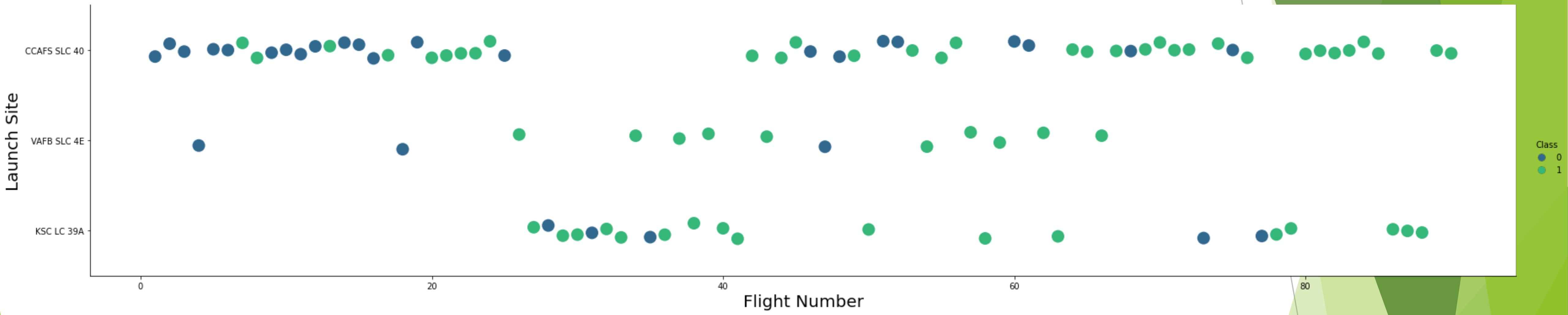
Payload range (Kg):

This is a preview of the Plotly dashboard. The following sides will show the results of EDA with visualization, EDA with SQL, Interactive Map with Folium, and finally the results of our model with about 83% accuracy.

EDA with Visualization

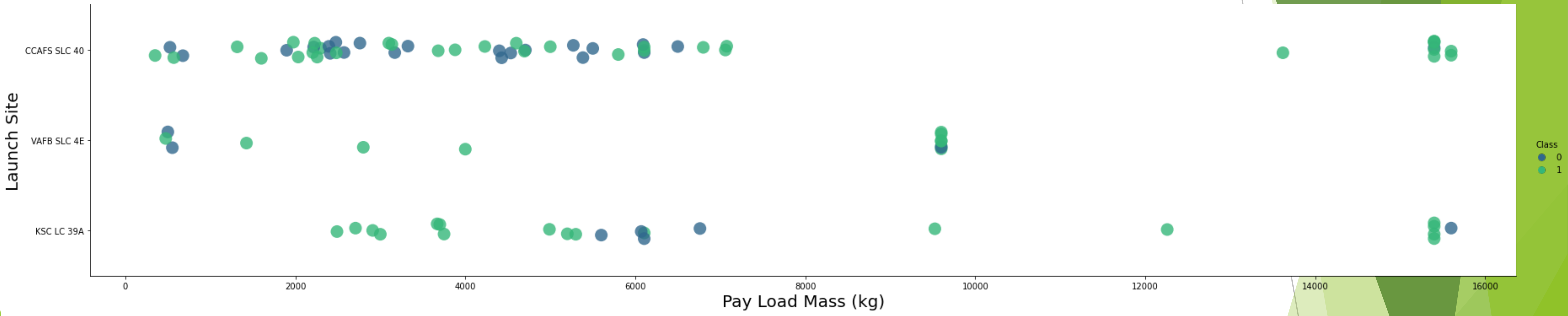
EXPLORATORY DATA ANALYSIS WITH SEABORN PLOTS

Flight Number vs. Launch Site



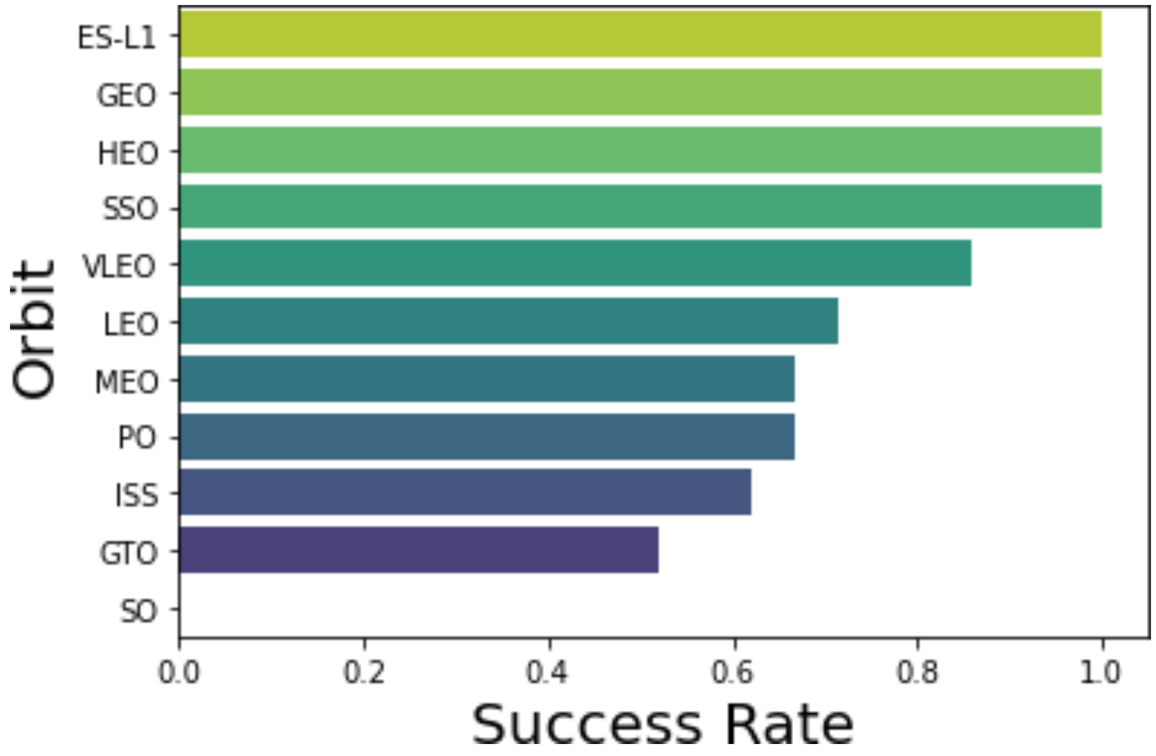
Green indicates successful launch; Purple indicates unsuccessful launch.

Payload vs. Launch Site



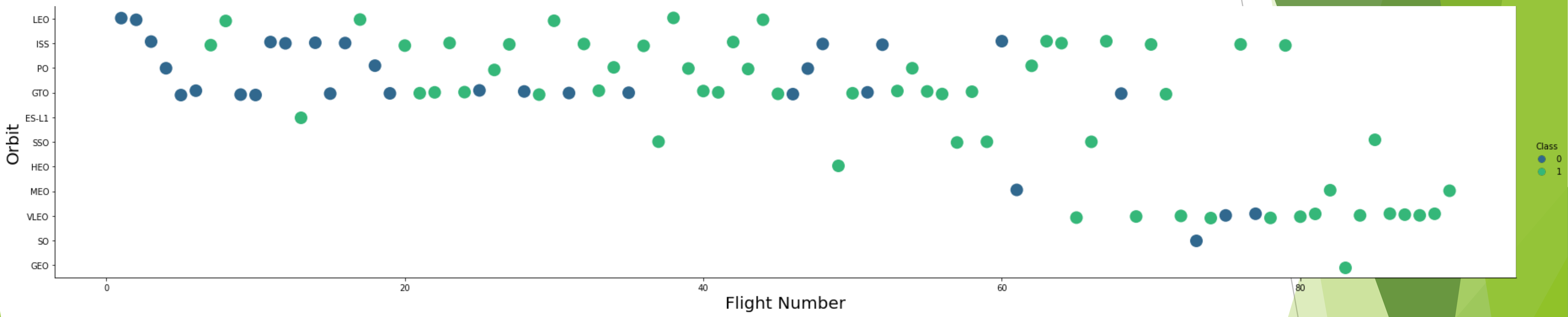
Green indicates successful launch; Purple indicates unsuccessful launch.

Success rate vs. Orbit type



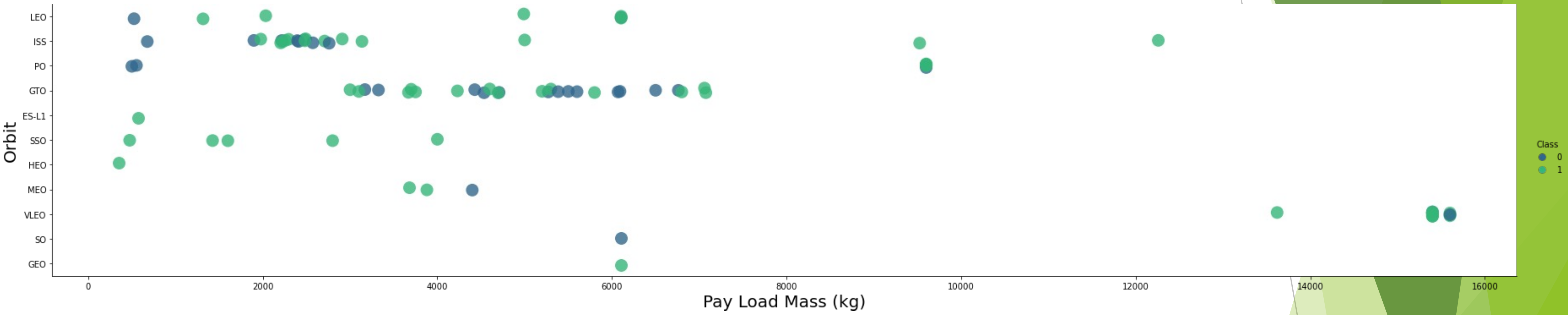
Success Rate Scale with
0 as 0%
0.6 as 60%
1 as 100%

Flight Number vs. Orbit type



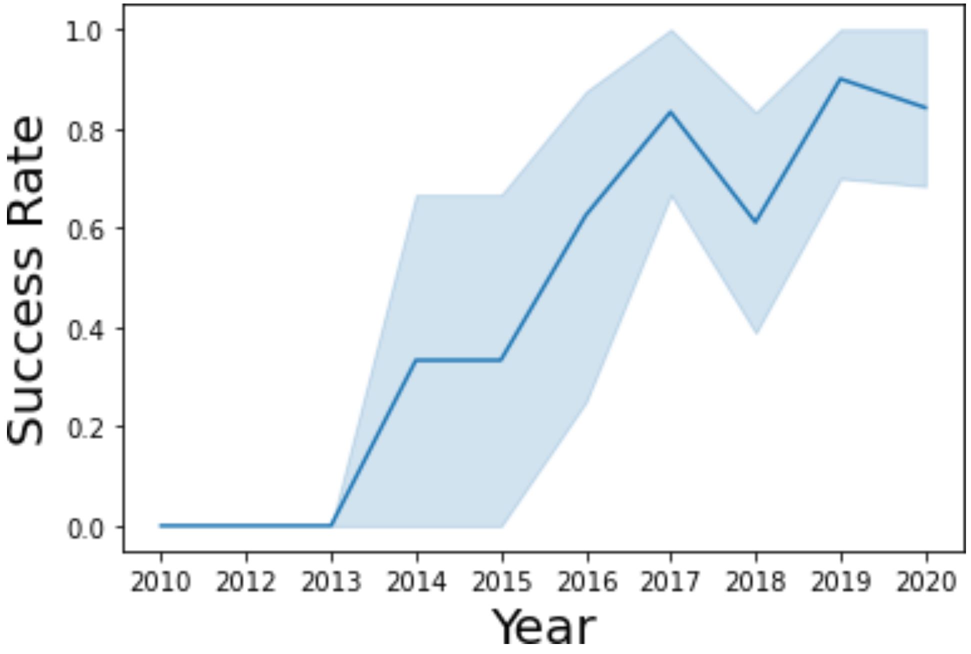
Green indicates successful launch; Purple indicates unsuccessful launch.

Payload vs. Orbit type



Green indicates successful launch; Purple indicates unsuccessful launch.

Launch Success Yearly Trend



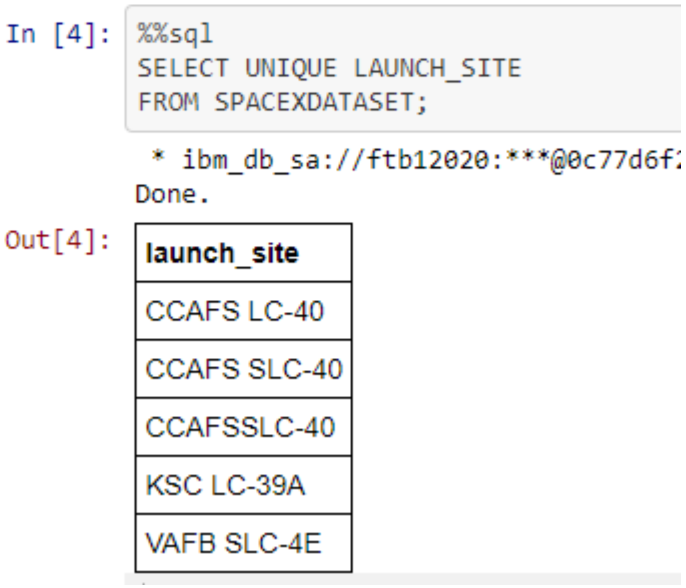
95% confidence interval
(light blue shading)

EDA with SQL

EXPLORATORY DATA ANALYSIS WITH SQL DB2

INTEGRATED IN PYTHON WITH SQLALCHEMY

All Launch Site Names



In [4]:

```
%%sql
SELECT UNIQUE LAUNCH_SITE
FROM SPACEXDATASET;
```

```
 * ibm_db_sa://ftb12020:***@0c77d6f
Done.
```

Out[4]:

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

Query unique launch site names from database.

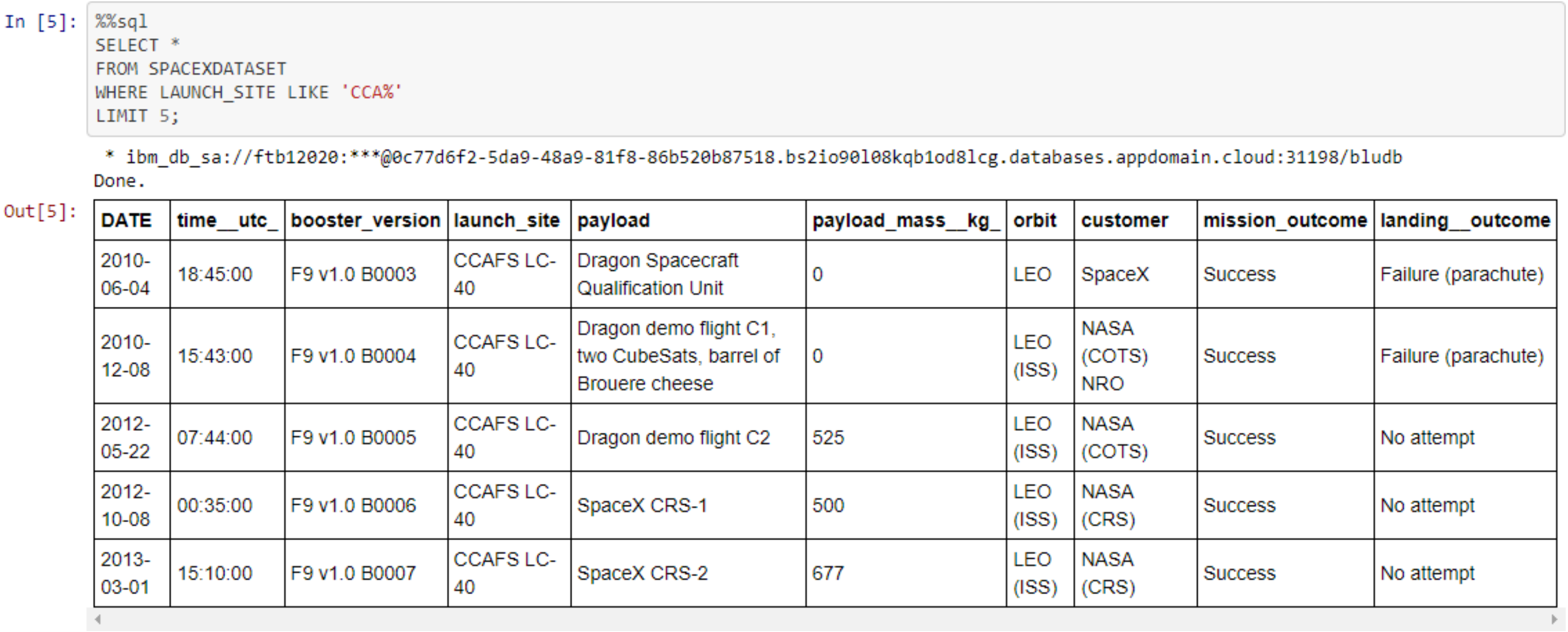
CCAFS SLC-40 and CCAFSSLC-40 likely all represent the same launch site with data entry errors.

CCAFS LC-40 was the previous name.

Likely only 3 unique launch_site values:

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

Launch Site Names Beginning with `CCA`



```
In [5]: %%sql
        SELECT *
        FROM SPACEXDATASET
        WHERE LAUNCH_SITE LIKE 'CCA%'
        LIMIT 5;
```

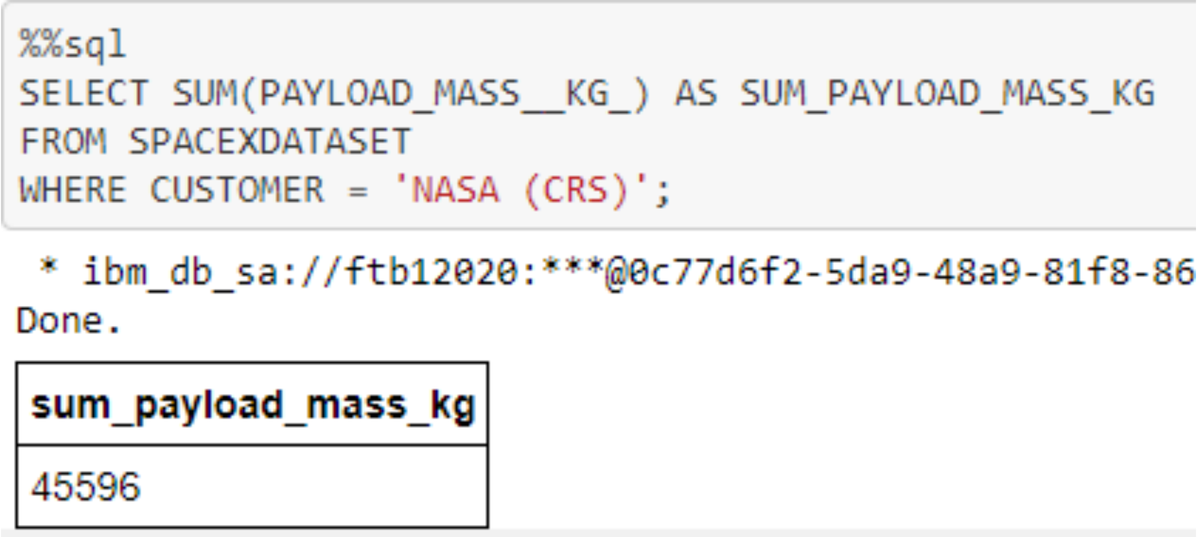
```
 * ibm_db_sa://ftb12020:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/bludb
Done.
```

Out[5]:

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	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

First five entries in database with Launch Site name beginning with CCA.

Total Payload Mass from NASA



```
%%sql
SELECT SUM(PAYLOAD_MASS__KG_) AS SUM_PAYLOAD_MASS_KG
FROM SPACEXDATASET
WHERE CUSTOMER = 'NASA (CRS)';
```

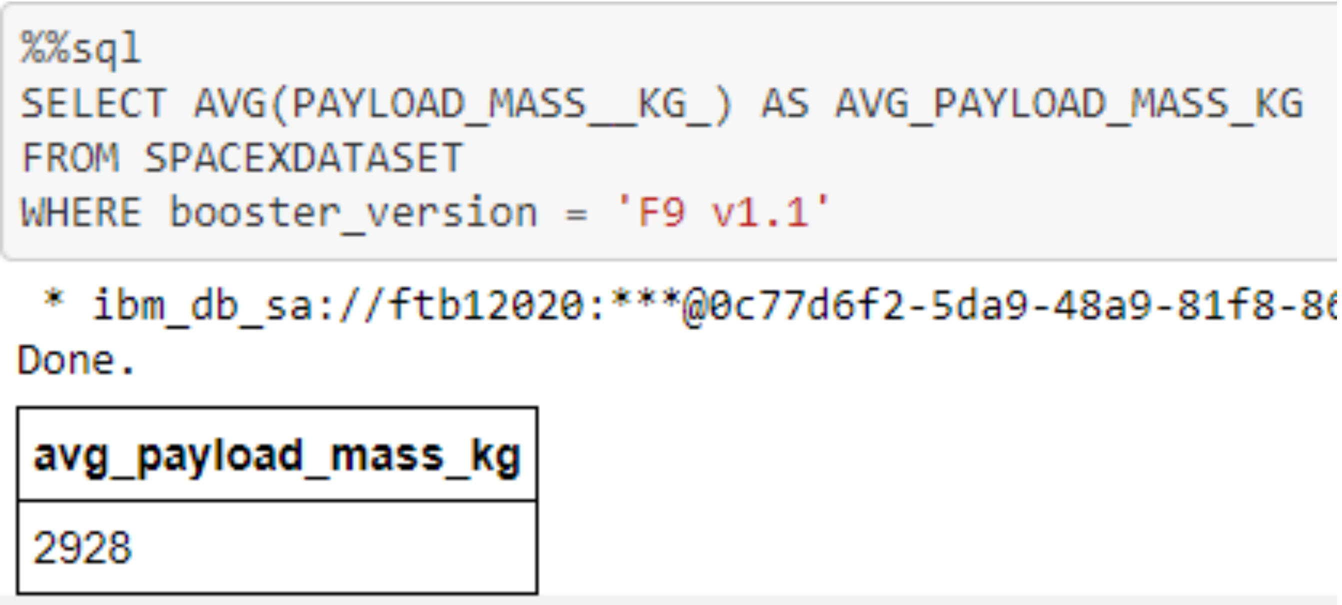
```
 * ibm_db_sa://ftb12020:***@0c77d6f2-5da9-48a9-81f8-86
Done.
```

sum_payload_mass_kg
45596

This query sums the total payload mass in kg where NASA was the customer.

CRS stands for Commercial Resupply Services which indicates that these payloads were sent to the International Space Station (ISS).

Average Payload Mass by F9v1.1



```
%%sql
SELECT AVG(PAYLOAD_MASS__KG_) AS AVG_PAYLOAD_MASS_KG
FROM SPACEXDATASET
WHERE booster_version = 'F9 v1.1'
```

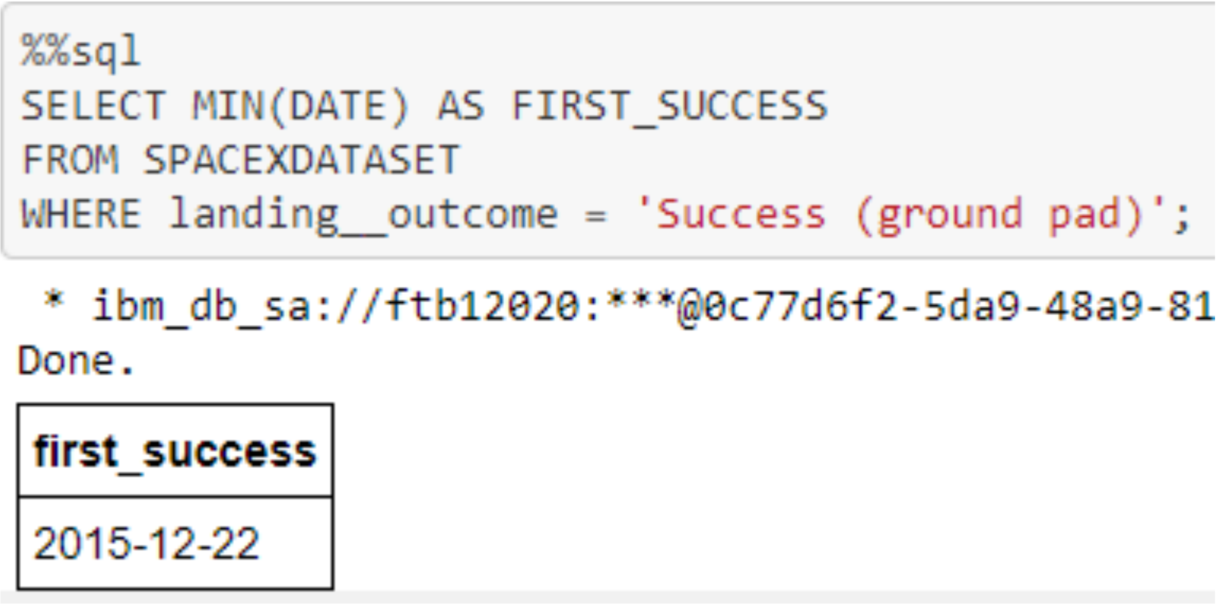
```
 * ibm_db_sa://ftb12020:***@0c77d6f2-5da9-48a9-81f8-86
Done.
```

avg_payload_mass_kg
2928

This query calculates the average payload mass or launches which used booster version F9 v1.1

Average payload mass of F9 1.1 is on the low end of our payload mass range

First Successful Ground Pad Landing Date



```
%%sql
SELECT MIN(DATE) AS FIRST_SUCCESS
FROM SPACEXDATASET
WHERE landing__outcome = 'Success (ground pad)';
```

```
 * ibm_db_sa://ftb12020:***@0c77d6f2-5da9-48a9-81
Done.
```

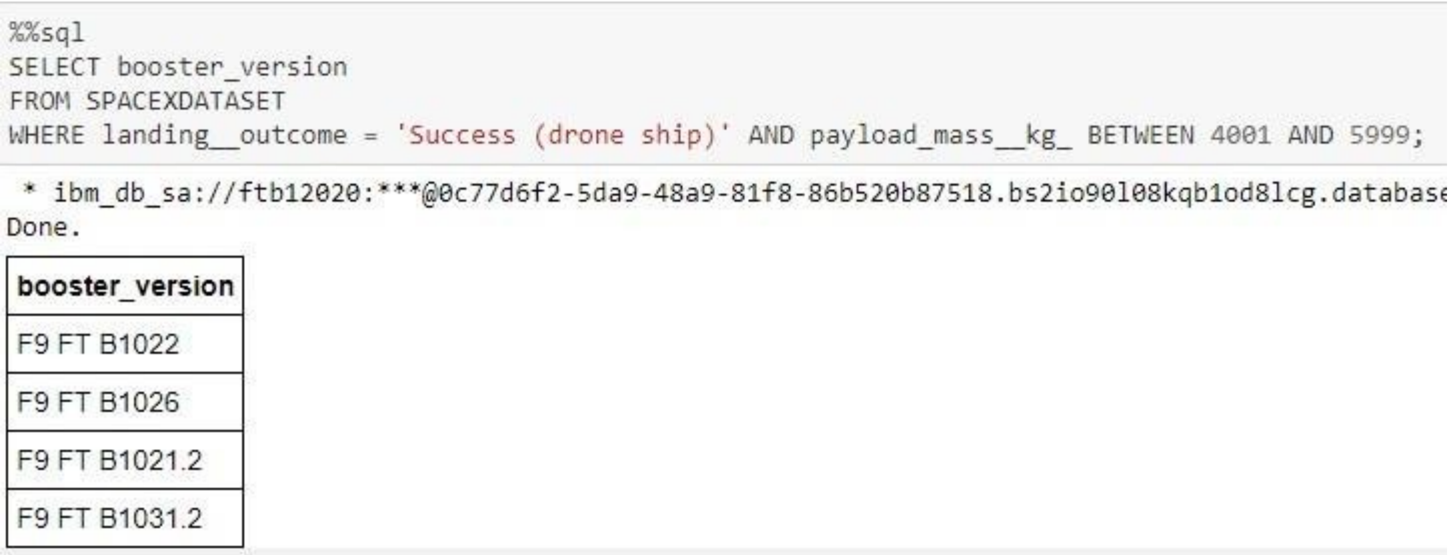
first_success
2015-12-22

This query returns the first successful ground pad landing date.

First ground pad landing wasn't until the end of 2015.

Successful landings in general appear starting 2014.

Successful Drone Ship Landing with Payload Between 4000 and 6000



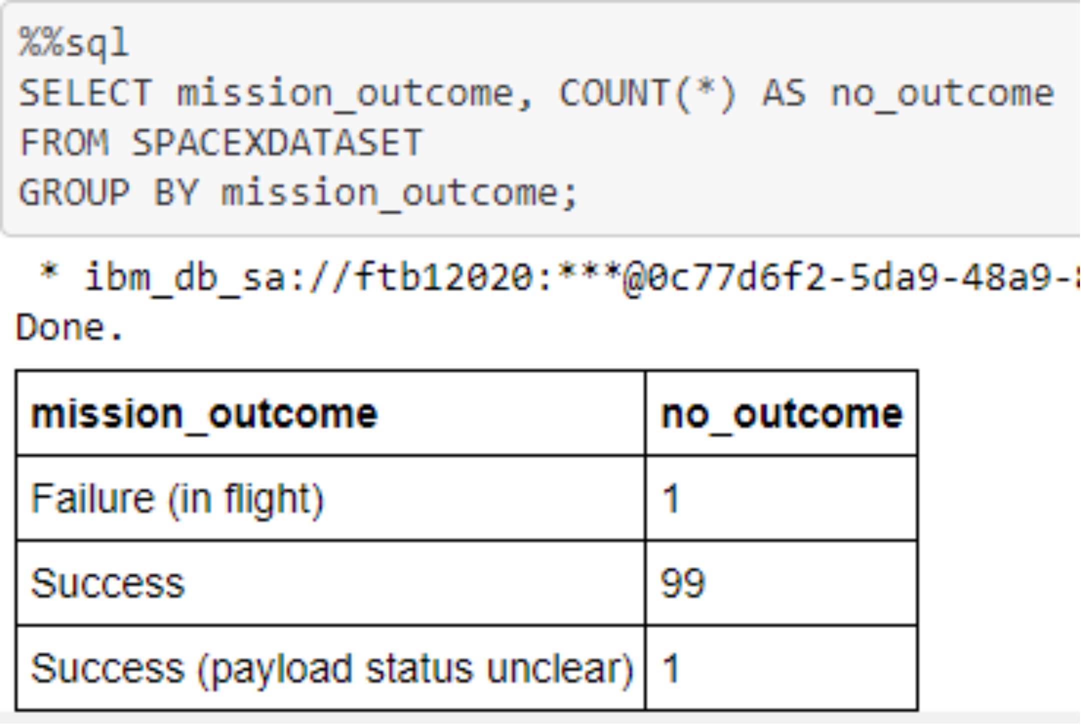
```
%%sql
SELECT booster_version
FROM SPACEXDATASET
WHERE landing__outcome = 'Success (drone ship)' AND payload_mass__kg_ BETWEEN 4001 AND 5999;
```

```
 * ibm_db_sa://ftb12020:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.database
Done.
```

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

This query returns the four booster versions that had successful drone ship landings and a payload mass between 4000 and 6000 noninclusively.

Total Number of Each Mission Outcome



```
%%sql
SELECT mission_outcome, COUNT(*) AS no_outcome
FROM SPACEXDATASET
GROUP BY mission_outcome;
```

```
 * ibm_db_sa://ftb12020:***@0c77d6f2-5da9-48a9-
Done.
```

mission_outcome	no_outcome
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

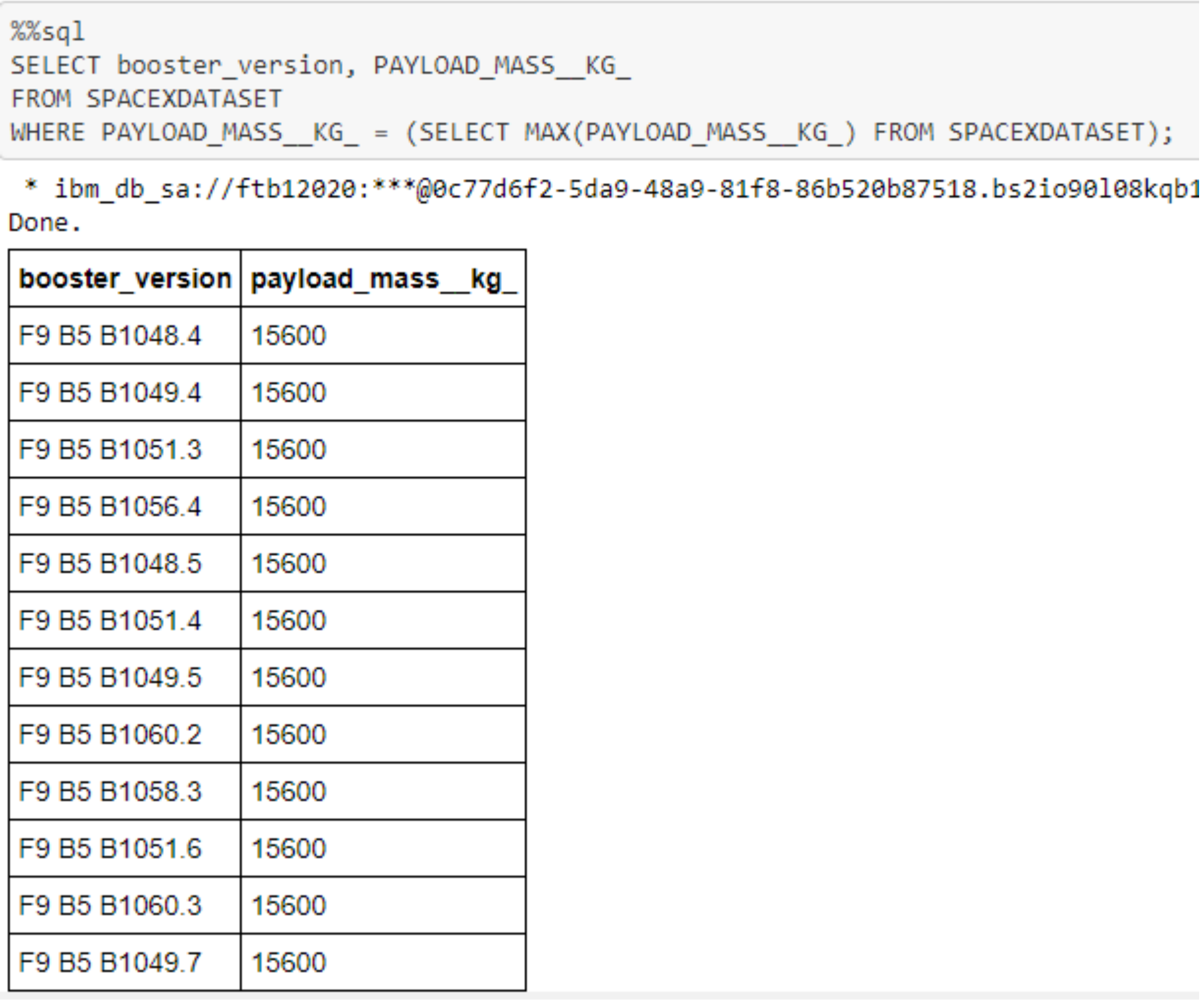
This query returns a count of each mission outcome.

SpaceX appears to achieve its mission outcome nearly 99% of the time.

This means that most of the landing failures are intended.

Interestingly, one launch has an unclear payload status and unfortunately one failed in flight.

Boosters that Carried Maximum Payload



```
%%sql
SELECT booster_version, PAYLOAD_MASS__KG_
FROM SPACEXDATASET
WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXDATASET);
```

```
 * ibm_db_sa://ftb12020:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1
Done.
```

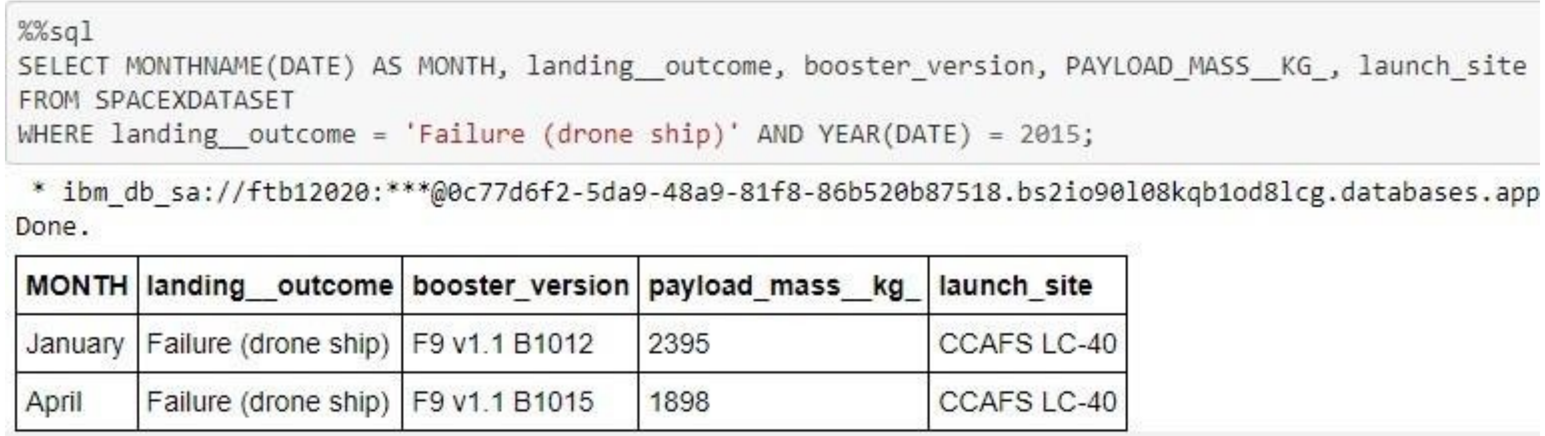
booster_version	payload_mass__kg_
F9 B5 B1048.4	15600
F9 B5 B1049.4	15600
F9 B5 B1051.3	15600
F9 B5 B1056.4	15600
F9 B5 B1048.5	15600
F9 B5 B1051.4	15600
F9 B5 B1049.5	15600
F9 B5 B1060.2	15600
F9 B5 B1058.3	15600
F9 B5 B1051.6	15600
F9 B5 B1060.3	15600
F9 B5 B1049.7	15600

This query returns the booster versions that carried the highest payload mass of 15600 kg.

These booster versions are very similar and all are of the F9 B5 B10xx.x variety.

This likely indicates payload mass correlates with the booster version that is used.

2015 Failed Drone Ship Landing Records



```
%%sql
SELECT MONTHNAME(DATE) AS MONTH, landing__outcome, booster_version, PAYLOAD_MASS__KG_, launch_site
FROM SPACEXDATASET
WHERE landing__outcome = 'Failure (drone ship)' AND YEAR(DATE) = 2015;
```

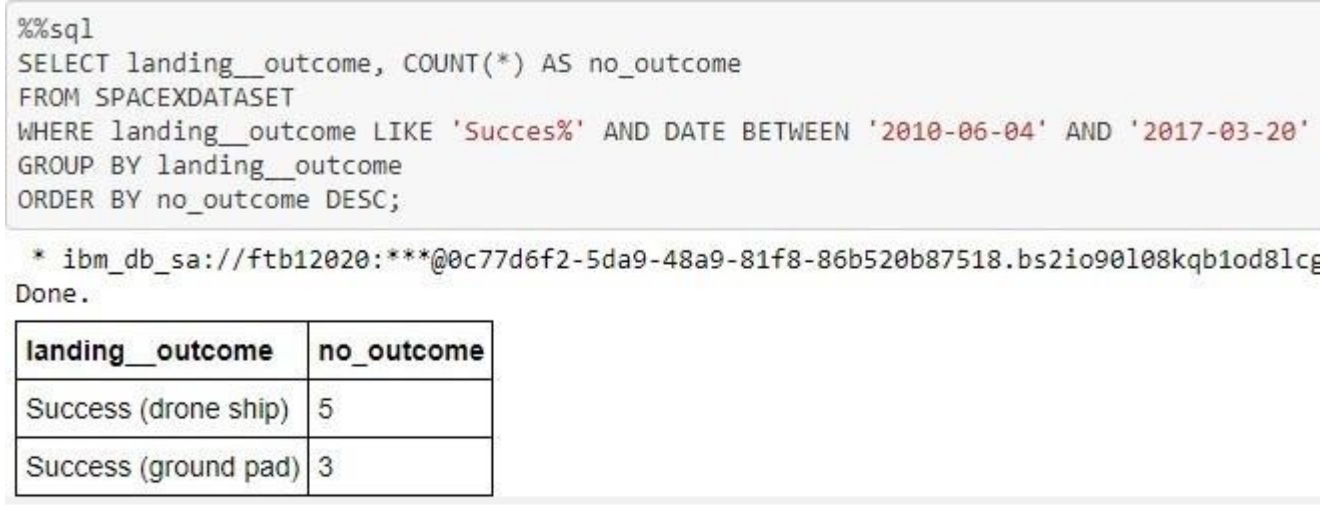
```
 * ibm_db_sa://ftb12020:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.app
Done.
```

MONTH	landing__outcome	booster_version	payload_mass__kg_	launch_site
January	Failure (drone ship)	F9 v1.1 B1012	2395	CCAFS LC-40
April	Failure (drone ship)	F9 v1.1 B1015	1898	CCAFS LC-40

This query returns the Month, Landing Outcome, Booster Version, Payload Mass (kg), and Launch site of 2015 launches where stage 1 failed to land on a drone ship.

There were two such occurrences.

Ranking Counts of Successful Landings Between 2010-06-04 and 2017-03-20



```
%%sql
SELECT landing__outcome, COUNT(*) AS no_outcome
FROM SPACEXDATASET
WHERE landing__outcome LIKE 'Succes%' AND DATE BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY landing__outcome
ORDER BY no_outcome DESC;
```

```
 * ibm_db_sa://ftb12020:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg
Done.
```

landing__outcome	no_outcome
Success (drone ship)	5
Success (ground pad)	3

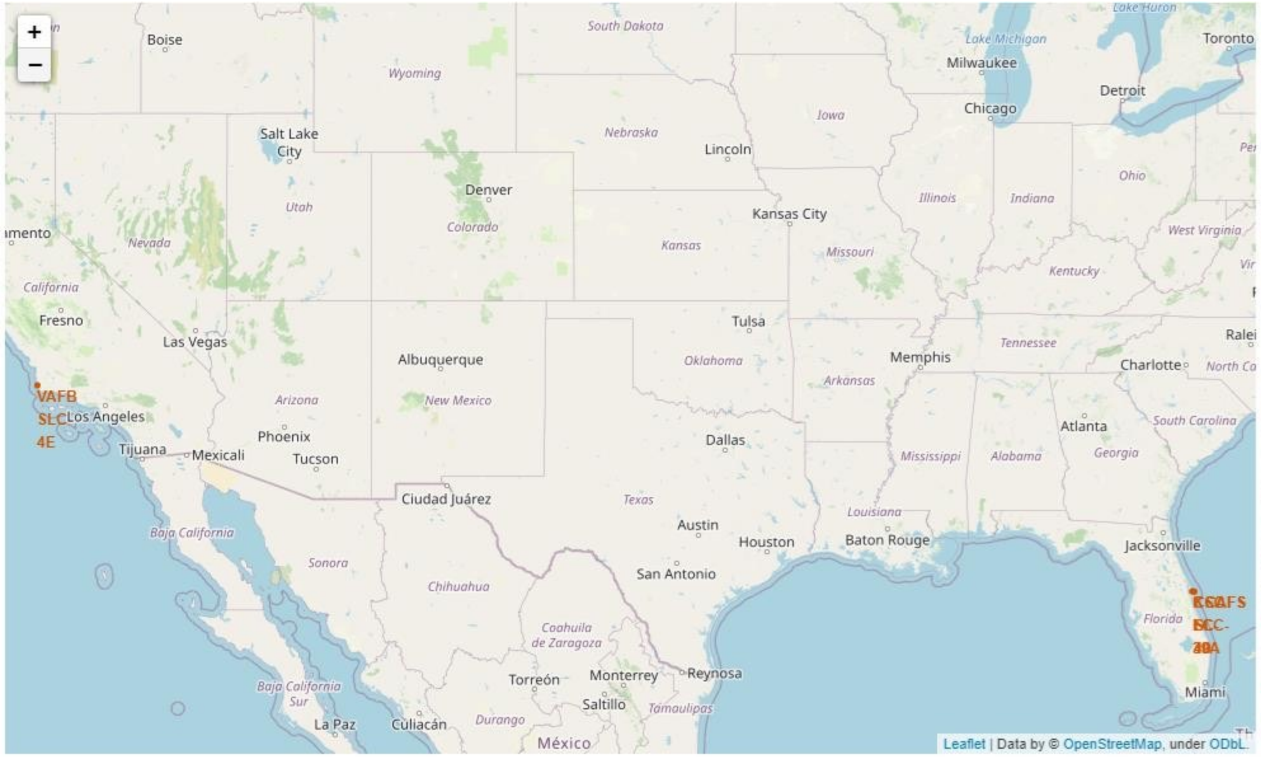
This query returns a list of successful landings and between 2010-06-04 and 2017-03-20 inclusively.

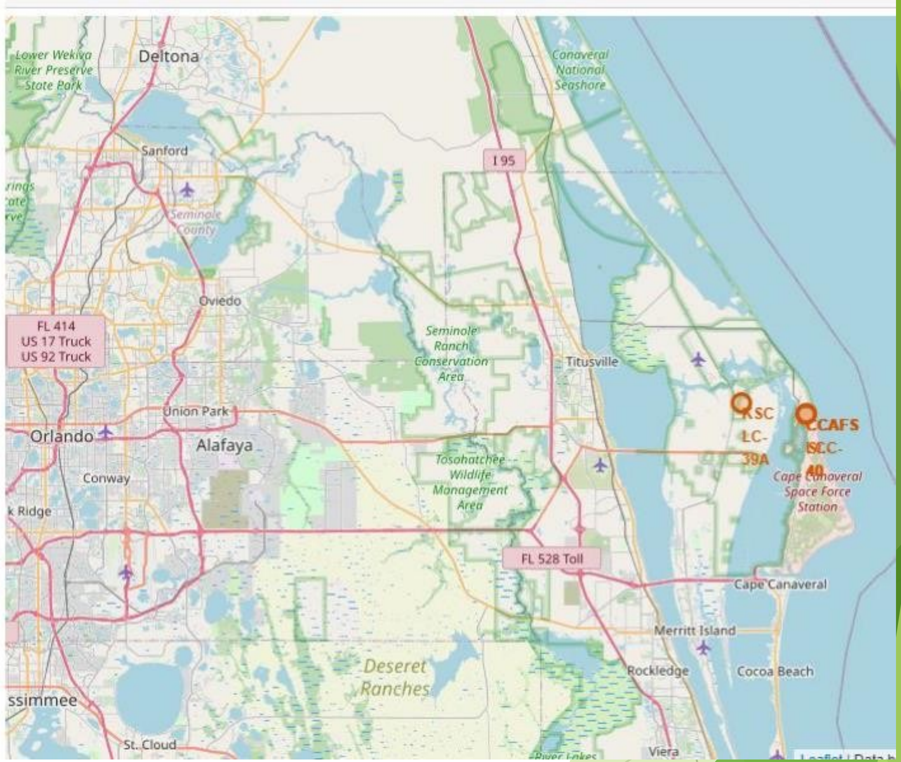
There are two types of successful landing outcomes: drone ship and ground pad landings.

There were 8 successful landings in total during this time period

Interactive Map with Folium

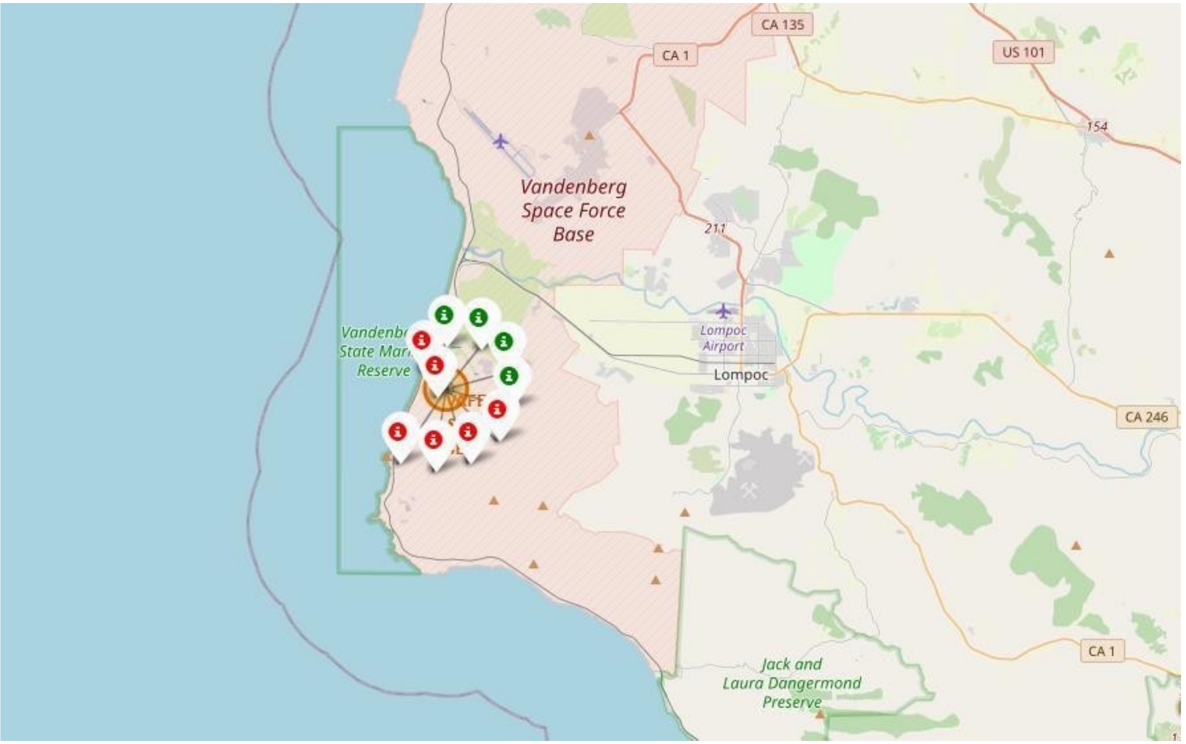
Launch Site Locations





The left map shows all launch sites relative US map. The right map shows the two Florida launch sites since they are very close to each other. All launch sites are near the ocean.

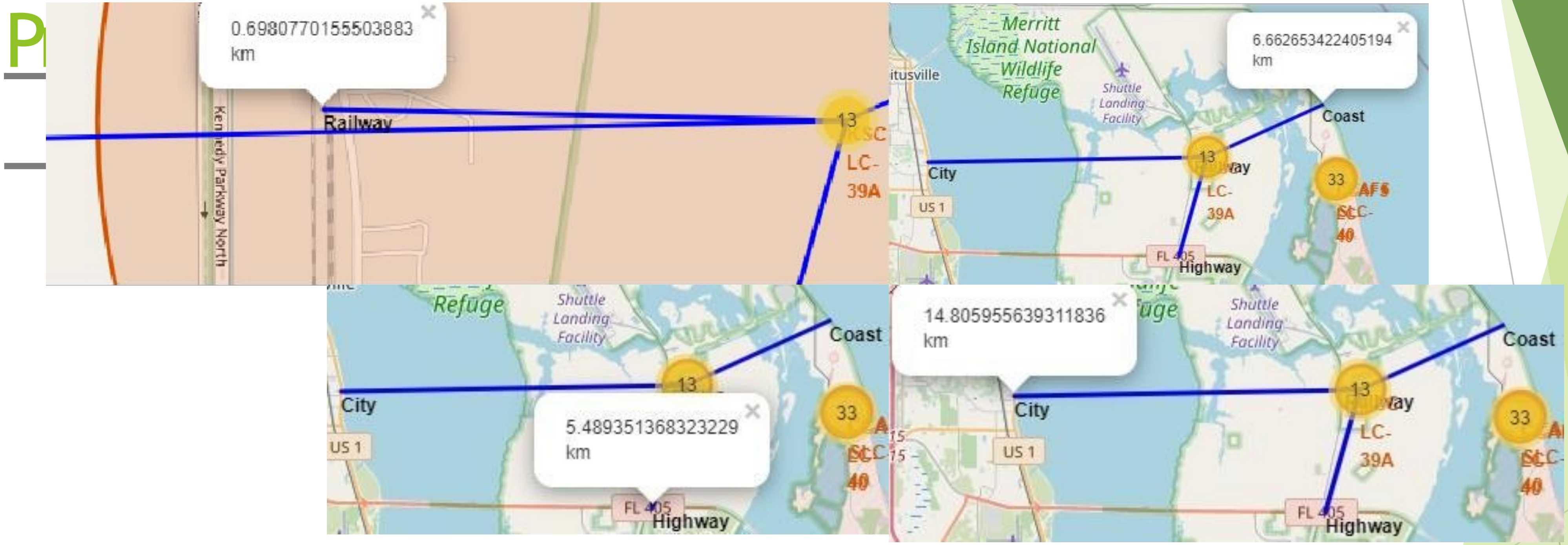
Color-Coded Launch Markers



CA 135
CA 1
US 101
154
Vandenberg Space Force Base
211
Lompoc Airport
Lompoc
CA 246
CA 1
Jack and Laura Dangermond Preserve

Clusters on Folium map can be clicked on to display each successful landing (green icon) and failed landing (red icon). In this example VAFB SLC-4E shows 4 successful landings and 6 failed landings.

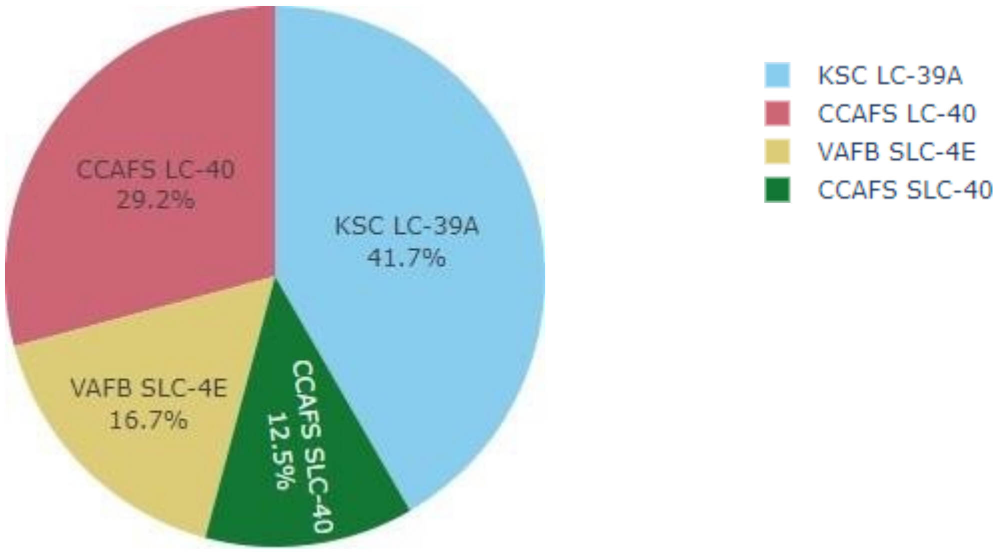
Key Location Pr



Using KSC LC-39A as an example, launch sites are very close to railways for large part and supply transportation. Launch sites are close to highways for human and supply transport. Launch sites are also close to coasts and relatively far from cities so that launch failures can land in the sea to avoid rockets falling on densely populated areas.

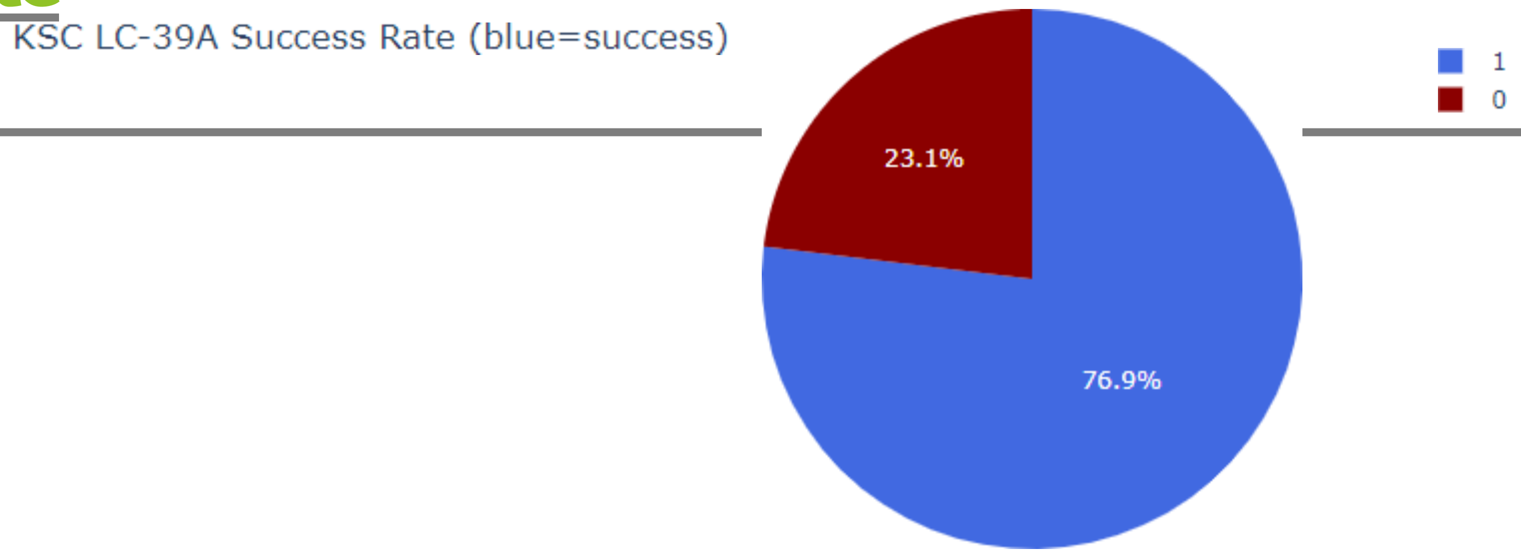
Build a Dashboard with Plotly Dash

Successful Launches Across Launch Sites



This is the distribution of successful landings across all launch sites. CCAFS LC-40 is the old name of CCAFS SLC-40 so CCAFS and KSC have the same amount of successful landings, but a majority of the successful landings where performed before the name change. VAFB has the smallest share of successful landings. This may be due to smaller sample and increase in difficulty of launching in the west coast.

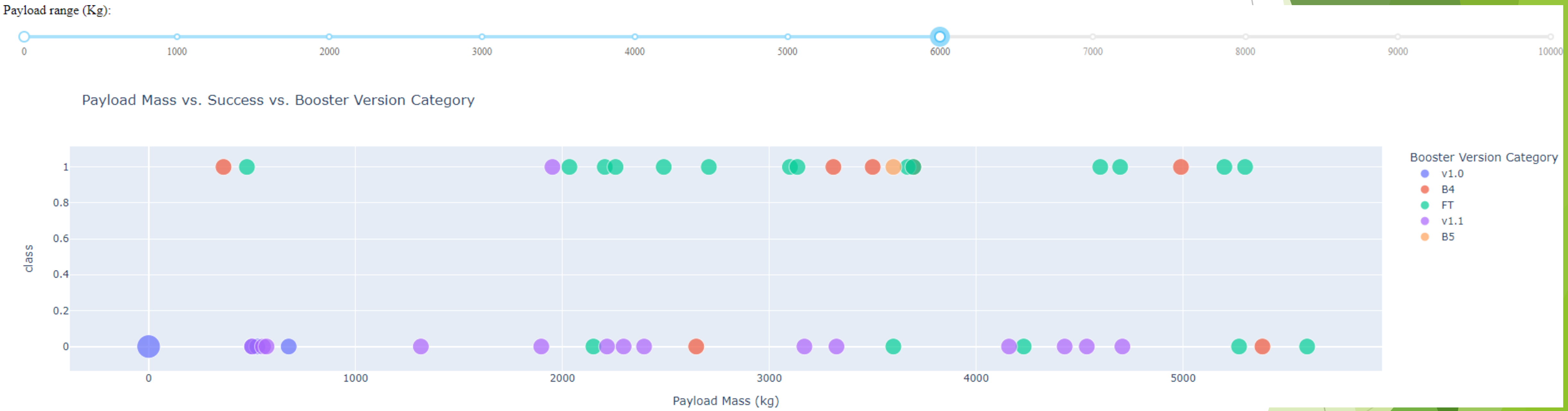
Highest Success Rate Launch Site



KSC LC-39A Success Rate (blue=success)

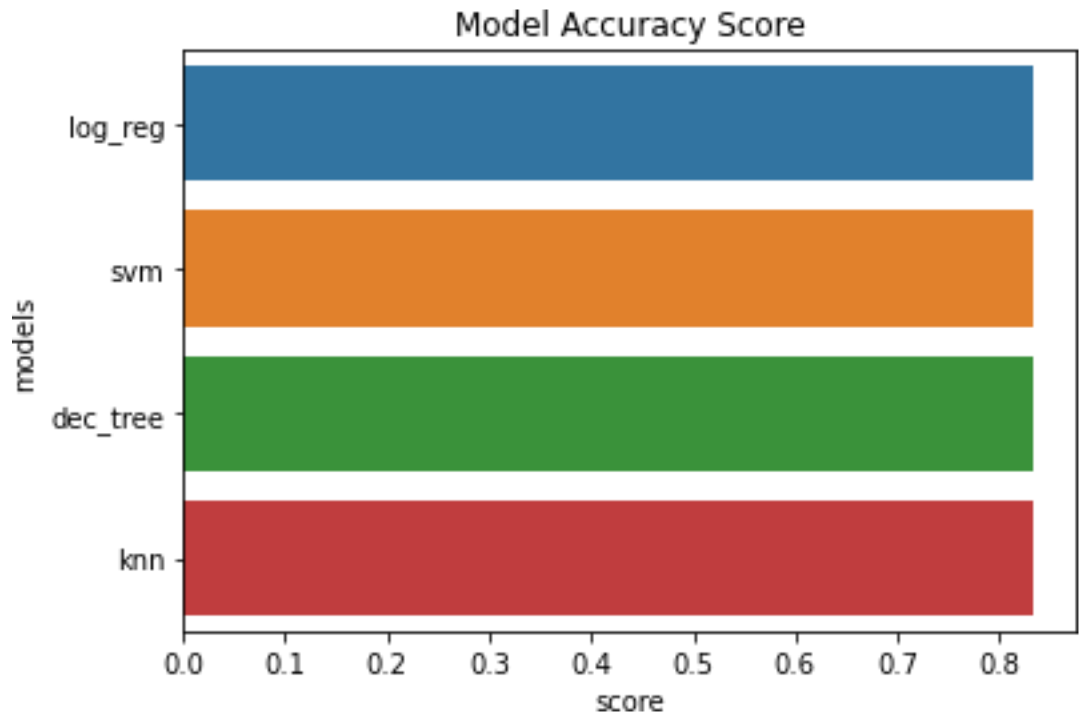
KSC LC-39A has the highest success rate with 10 successful landings and 3 failed landings.

Payload Mass vs. Success vs. Booster Version Category


Payload range (Kg):

Plotly dashboard has a Payload range selector. However, this is set from 0-10000 instead of the max Payload of 15600. Class indicates 1 for successful landing and 0 for failure. Scatter plot also accounts for booster version category in color and number of launches in point size. In this particular range of 0-6000, interestingly there are two failed landings with payloads of zero kg.

Classification Accuracy



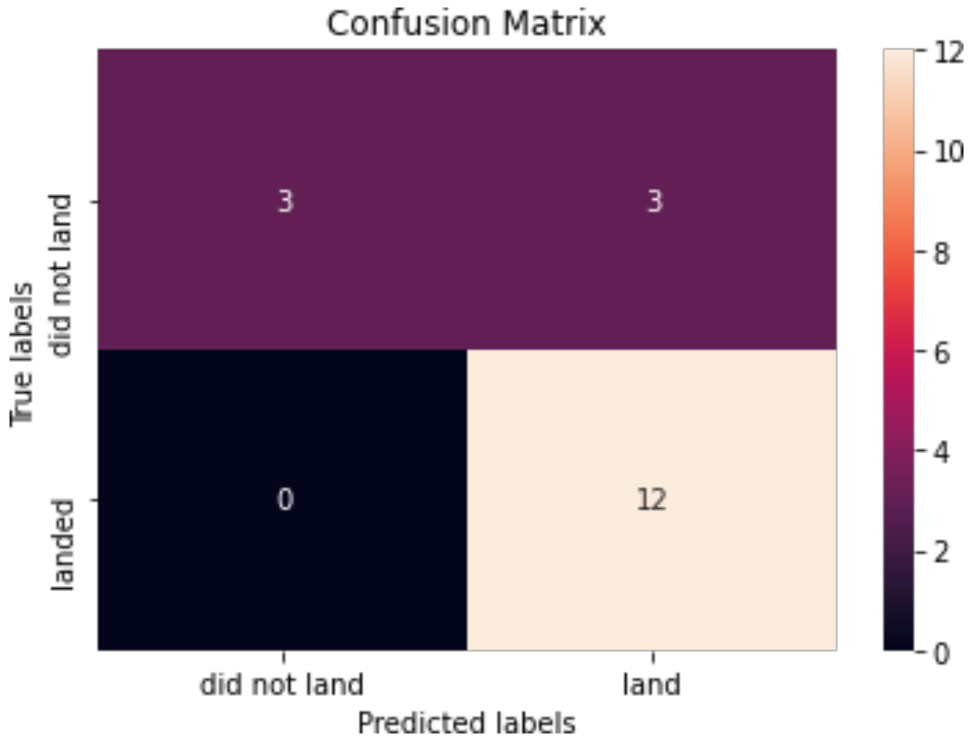
All models had virtually the same accuracy on the test set at 83.33% accuracy.

It should be noted that test size is small at only sample size of 18.

This can cause large variance in accuracy results, such as those in Decision Tree Classifier model in repeated runs.

We likely need more data to determine the best model.

Confusion Matrix



Correct predictions are on a diagonal from top left to bottom right.

Since all models performed the same for the test set, the confusion matrix is the same across all models.

The models predicted 12 successful landings when the true label was successful landing.

The models predicted 3 unsuccessful landings when the true label was unsuccessful landing.

The models predicted 3 successful landings when the true label was unsuccessful landings (false positives).

Our models over predict successful landings.

CONCLUSION

- Our task: to develop a machine learning model for Space Y who wants to bid against SpaceX
- The goal of model is to predict when Stage 1 will successfully land to save ~$100 million USD
- Used data from a public SpaceX API and web scraping SpaceX Wikipedia page
- Created data labels and stored data into a DB2 SQL database
- Created a dashboard for visualization
- We created a machine learning model with an accuracy of 83%
- Allon Mask of SpaceY can use this model to predict with relatively high accuracy whether a launch will have a successful Stage 1 landing before launch to determine whether the launch should be made or not
- If possible, more data should be collected to better determine the best machine learning model and improve accuracy