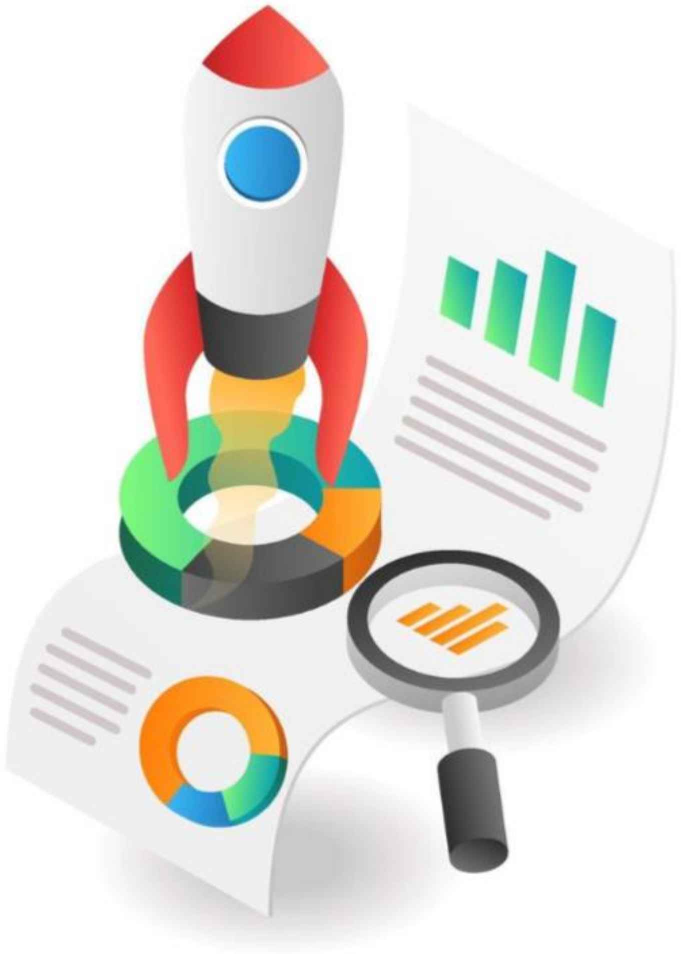
A Data Science Approach to Winning the Space Race

Ann Gline Gomez

https://github.com/AnnGlineGomez/IBM_Applied_Capstone_Project_by_Ann_Gline_Gomez

13/06/2023

Outline



- Executive Summary (3)
- Introduction (4)
- Methodology (6)
- Results (16)
- Conclusion (44)
- Appendix (45)

Executive Summary

Summary of methodologies

- Data Collection through API
- Data Collection with Web Scraping
- Data Wrangling
- Exploratory Data Analysis with SQL
- Exploratory Data Analysis with Data Visualization
- Interactive Visual Analytics with Folium
- Machine Learning Prediction

•Summary of all results

- Exploratory Data Analysis result
- Interactive analytics in screenshots
- Predictive Analytics result

Introduction

SpaceX Falcon 9 Rocket – The Verge

Context:

- We are currently in the era of commercial space exploration, and SpaceX is leading the way.
- SpaceX stands out with its competitive pricing of $62 million USD compared to the industry average of $165 million USD.
- One of the key factors contributing to this advantage is SpaceX's ability to recover a significant portion of their rockets, specifically Stage 1.
- In this scenario, Space Y aims to rival SpaceX's achievements.

Problem Statement:

- Space Y has approached us with a specific task: to develop a machine-learning model that can accurately predict the success of Stage 1 rocket recovery.

Methodology

Data Collection Approach:

- The data for this project was gathered by combining information from the SpaceX public API and the SpaceX Wikipedia page.
- A comprehensive data wrangling process was carried out to ensure the data was in a suitable format for analysis.
- The collected data was then classified, distinguishing between true landings categorized as successful and those categorized as unsuccessful.

Exploratory Data Analysis (EDA) Process:

- To gain insights from the collected data, exploratory data analysis techniques were employed.
- Visualization methods and SQL queries were utilized to delve into the data and extract meaningful patterns.
- Additionally, interactive visual analytics tools such as Folium and Plotly Dash were employed to enhance the exploration process and provide interactive visual representations of the data.

Predictive Analysis Approach:

- To predict the success of Stage 1 rocket recovery, various classification models were employed.
- These models were trained using the collected data and subsequently fine-tuned using the GridSearchCV technique to optimize their performance.

METHODOLOGY

OVERVIEW OF DATA COLLECTION, WRANGLING, VISUALIZATION, DASHBOARD, AND MODEL METHODS

Summary of Data Collection Process

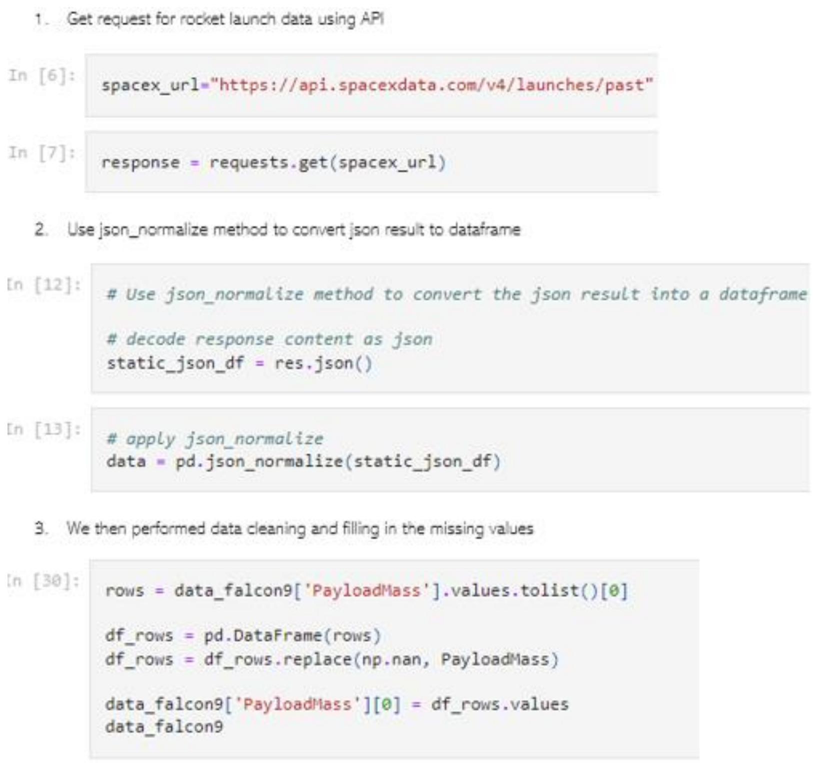
- The data collection process involved a combination of obtaining data through API requests from the Space X public API and extracting information from a table in the Space X Wikipedia entry using web scraping.

- The next slide will showcase the flowchart illustrating the data collection from the API, followed by another slide depicting the flowchart for data collection through web scraping.

- For the Space X API data, the collected columns include FlightNumber, Date, BoosterVersion, PayloadMass, Orbit, LaunchSite, Outcome, Flights, GridFins, Reused, Legs, LandingPad, Block, ReusedCount, Serial, Longitude, and Latitude.

- Regarding the Wikipedia web scrape data, the columns obtained consist of Flight No., Launch site, Payload, PayloadMass, Orbit, Customer, Launch outcome, Version Booster, Booster landing, Date, and Time.

Data Collection – SpaceX API



1. Get request for rocket launch data using API

```
In [6]:  spacex_url="https://api.spacexdata.com/v4/launches/past"
```

```
In [7]:  response = requests.get(spacex_url)
```

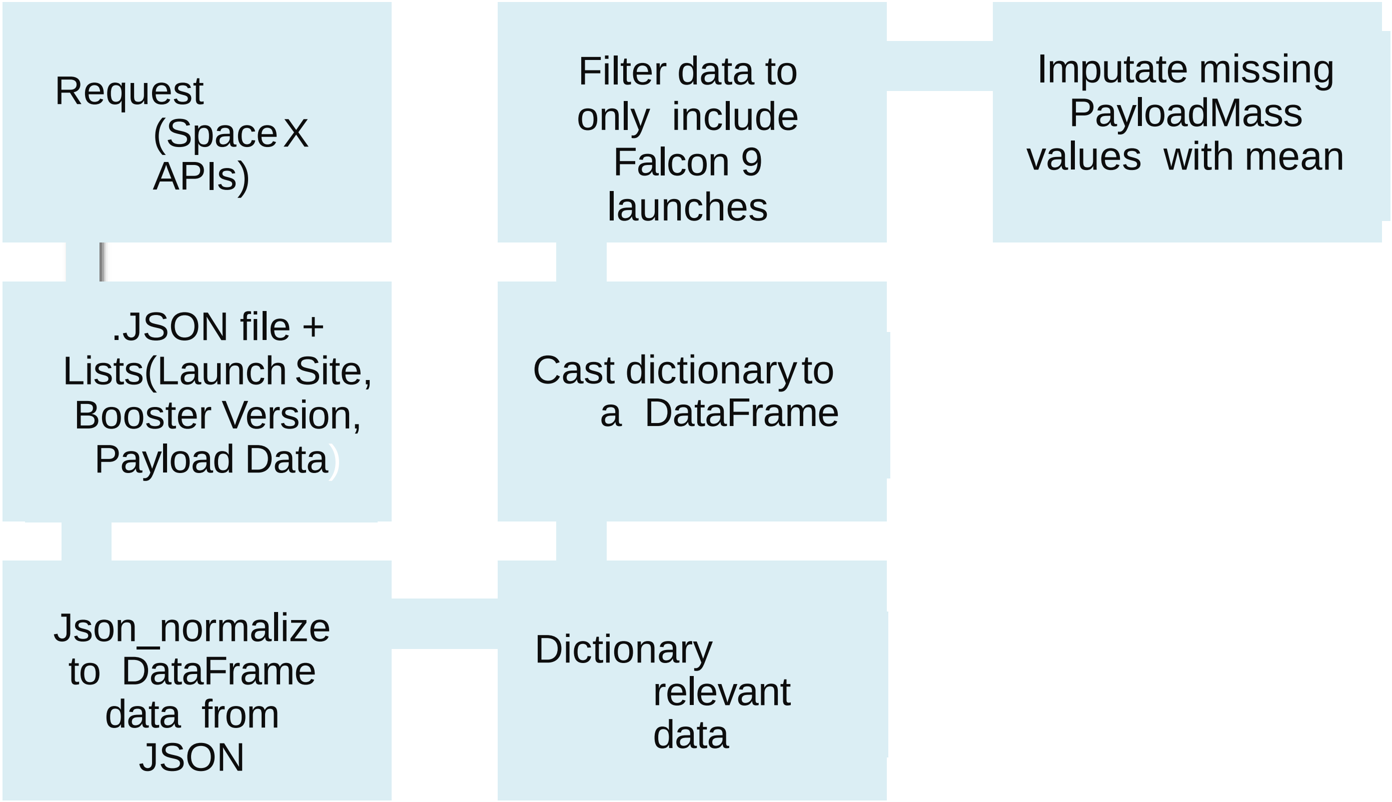
2. Use json_normalize method to convert json result to dataframe

```
In [12]: # Use json_normalize method to convert the json result into a dataframe

         # decode response content as json
         static_json_df = res.json()
```

```
In [13]: # apply json_normalize
         data = pd.json_normalize(static_json_df)
```

3. We then performed data cleaning and filling in the missing values

```
In [30]: rows = data_falcon9['PayloadMass'].values.tolist()[0]

         df_rows = pd.DataFrame(rows)
         df_rows = df_rows.replace(np.nan, PayloadMass)

         data_falcon9['PayloadMass'][0] = df_rows.values
         data_falcon9
```



Request (Space X 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

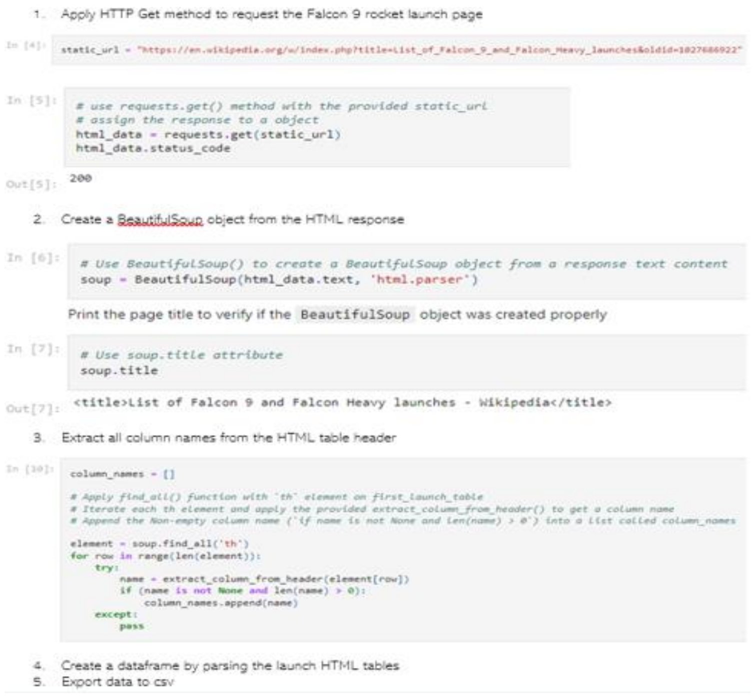
Filter data to only include Falcon 9 launches

Imputate missing PayloadMass values with mean

GitHub url:

https://github.com/AnnGlineGomez/IBM_Applied_Capstone_Project_by_Ann_Gline_Gomez/blob/main/DataCollectionAPI.ipynb

Data Collection – Web Scraping



1. Apply HTTP Get method to request the Falcon 9 rocket launch page

```
In [4]: static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"
```

```
In [5]: # use requests.get() method with the provided static_url
        # assign the response to a object
        html_data = requests.get(static_url)
        html_data.status_code

Out[5]: 200
```

2. Create a BeautifulSoup object from the HTML response

```
In [6]: # Use BeautifulSoup() to create a BeautifulSoup object from a response text content
        soup = BeautifulSoup(html_data.text, 'html.parser')
```

Print the page title to verify if the `BeautifulSoup` object was created properly

```
In [7]: # Use soup.title attribute
        soup.title

Out[7]: <title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>
```

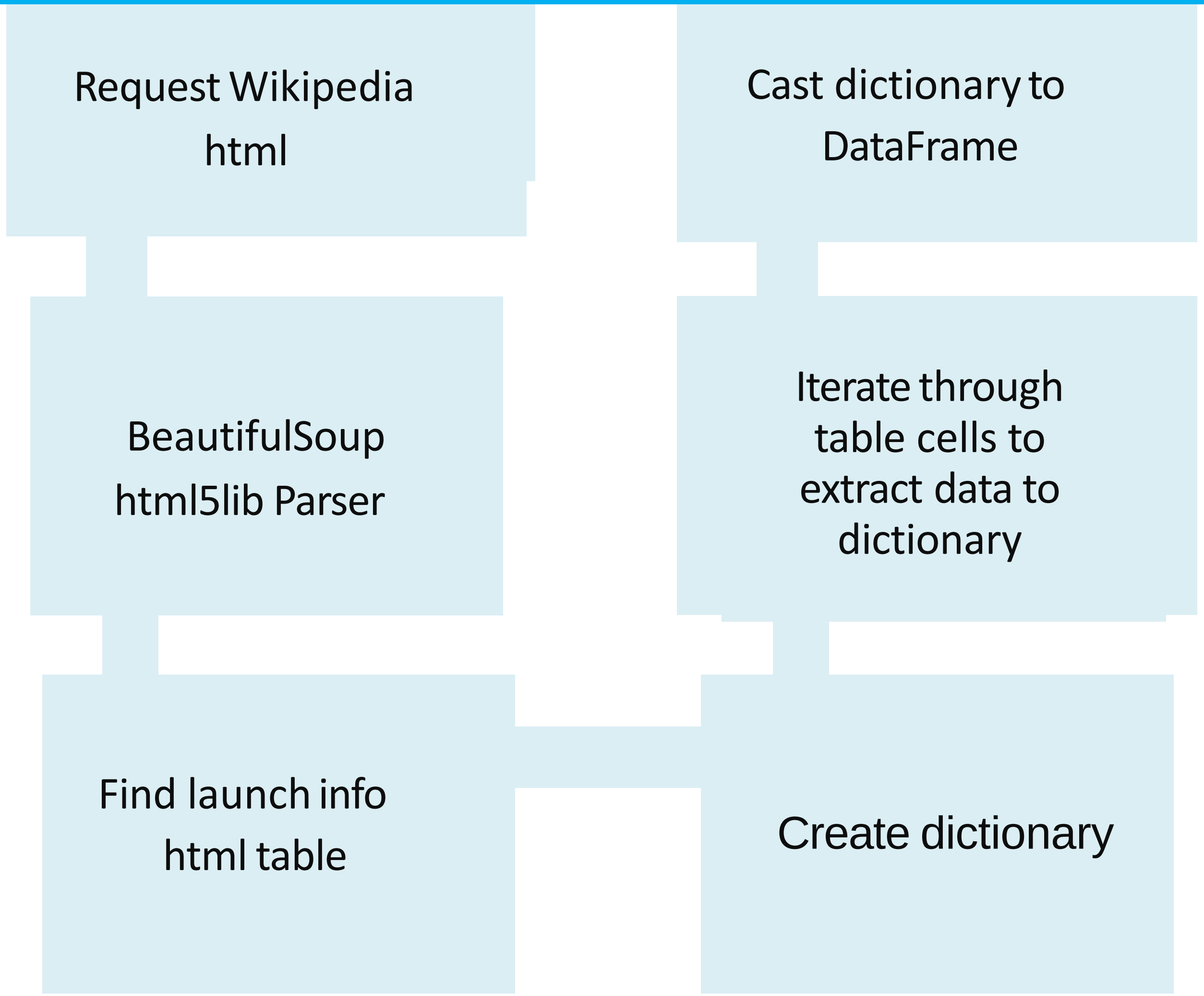
3. Extract all column names from the HTML table header

```
In [10]: column_names = []

         # Apply find_all() function with `th` element on first_launch_table
         # Iterate each th element and apply the provided extract_column_from_header() to get a column name
         # Append the Non-empty column name (`if name is not None and len(name) > 0`) into a list called column_names

         element = soup.find_all('th')
         for row in range(len(element)):
             try:
                 name = extract_column_from_header(element[row])
                 if (name is not None and len(name) > 0):
                     column_names.append(name)
             except:
                 pass
```

4. Create a dataframe by parsing the launch HTML tables
5. Export data to csv

GitHub url:

https://github.com/AnnGlineGomez/IBM_Applied_Capstone_Project_by_Ann_Gline_Gomez/blob/main/DataCollectionWithWebScrapping.ipynb



Request Wikipedia html

BeautifulSoup html5lib Parser

Find launch info html table

Create dictionary

Iterate through table cells to extract data to dictionary

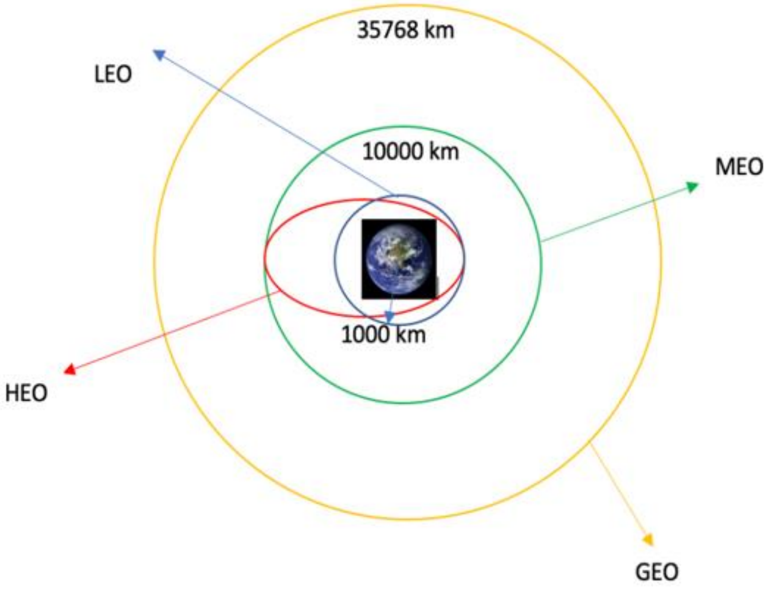
Cast dictionary to DataFrame

Data Wrangling

A training label was generated based on the landing outcomes, assigning a value of 1 for successful landings and 0 for failures. The 'Outcome' column comprises two components: 'Mission Outcome' and 'Landing Location'. A new column named 'class' was created as the training label, where a value of 1 is assigned if the 'Mission Outcome' is true, and 0 otherwise. The value mapping is as follows:

'True ASDS', 'True RTLS', and 'True Ocean' are set to 1.

'None ', 'False ASDS', 'None ASDS', 'False Ocean', and 'False RTLS' are set to 0.



GitHub url:

https://github.com/AnnGlineGomez/IBM_Applied_Capstone_Project_by_Ann_Gline_Gomez/blob/main/DataWrangling.ipynb

EDA with Data Visualization

To gain insights into the data, an exploratory data analysis was carried out on several variables, namely Flight Number, Payload Mass, Launch Site, Orbit, Class, and Year. Various types of plots, including scatter plots, line charts, and bar plots, were employed to investigate the relationships between these variables.

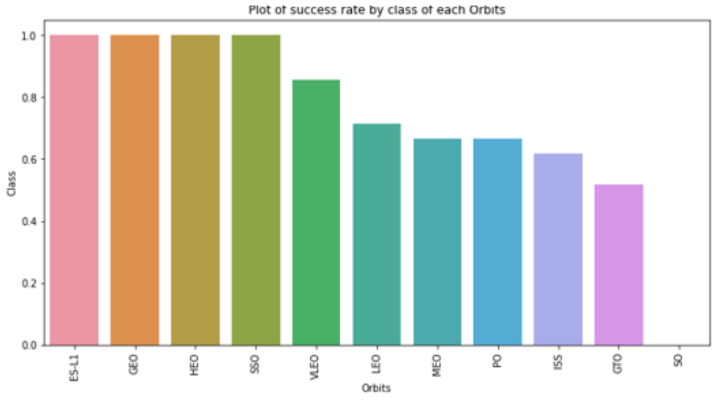
In particular, the following plots were utilized in the analysis: Flight Number vs. Payload Mass, Flight Number vs. Launch Site, Payload Mass vs. Launch Site, Orbit vs. Success Rate, Flight Number vs. Orbit, Payload vs. Orbit, and Success Yearly Trend. By visualizing these relationships, we aimed to uncover any potential connections and dependencies between the variables.

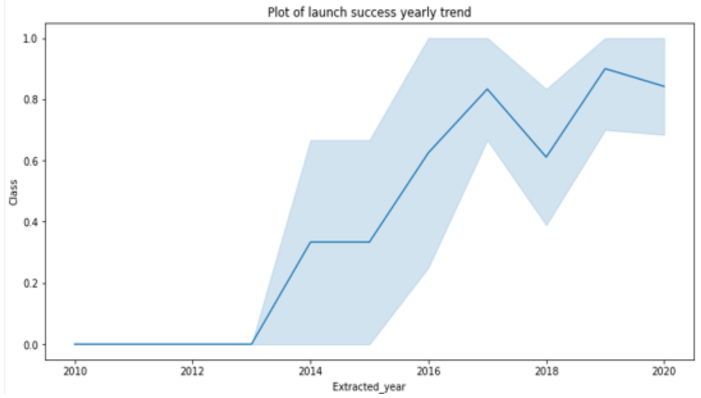
The exploration of the data involved examining the association between flight number and launch site, payload and launch site, success rate of each orbit type, flight number and orbit type, as well as the yearly trend of launch success. These visualizations allowed us to assess if any significant relationships existed among the variables.

The ultimate goal of these visualizations was to identify suitable variables that could be used to train a machine learning model. By discovering patterns and correlations through the analysis, we aimed to determine which variables would be most relevant and informative for the model's training process.

GitHub url:

https://github.com/AnnGlineGomez/IBM_Applied_Capstone_Project_by_Ann_Gline_Gomez/blob/main/EDAwithVisualization.ipynb





EDA with SQL

The dataset was loaded into an IBM DB2 Database, and SQL Python integration was utilized to execute queries.

These queries were performed to gain a deeper understanding of the dataset, extracting information such as launch site names, mission outcomes, payload sizes of various customers, booster versions, and landing outcomes.

GitHub url:

https://github.com/AnnGlineGomez/IBM_Applied_Capstone_Project_by_Ann_Gline_Gomez/blob/main/EDAwithSQL.ipynb

Build an interactive map with Folium

- Folium maps were utilized to mark Launch Sites, successful and unsuccessful landings, as well as provide an example of proximity to key locations such as Railway, Highway, Coast, and City.

- This visual representation aids in understanding the rationale behind the selection of launch site locations and provides a visualization of successful landings in relation to their respective locations.

- We marked all launch sites, and added map objects such as markers, circles, lines to mark the success or failure of launches for each site on the folium map.

- We assigned the feature launch outcomes (failure or success) to class 0 and 1.i.e., 0 for failure, and 1 for success. Using the color-labeled marker clusters, we identified which launch sites have relatively high success rate.

- We calculated the distances between a launch site to its proximities. We answered some question for instance:

  -Are launch sites near railways, highways and coastlines.

  -Do launch sites keep certain distance away from cities.

GitHub url:

https://github.com/AnnGlineGomez/IBM_Applied_Capstone_Project_by_Ann_Gline_Gomez/blob/main/Interactive%20Visual%20Analytics%20with%20Folium.ipynb

Predictive analysis (Classification)

Predictive analysis involves using historical data and statistical techniques to make predictions or forecasts about future outcomes.

It utilizes machine learning algorithms and models to identify patterns, trends, and relationships within the data, and then applies these insights to predict future outcomes or make informed decisions.

GitHub url:

https://github.com/AnnGlineGomez/IBM_Applied_Capstone_Project_by_Ann_Gline_Gomez/blob/main/MachineLearningPrediction.ipynb

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.

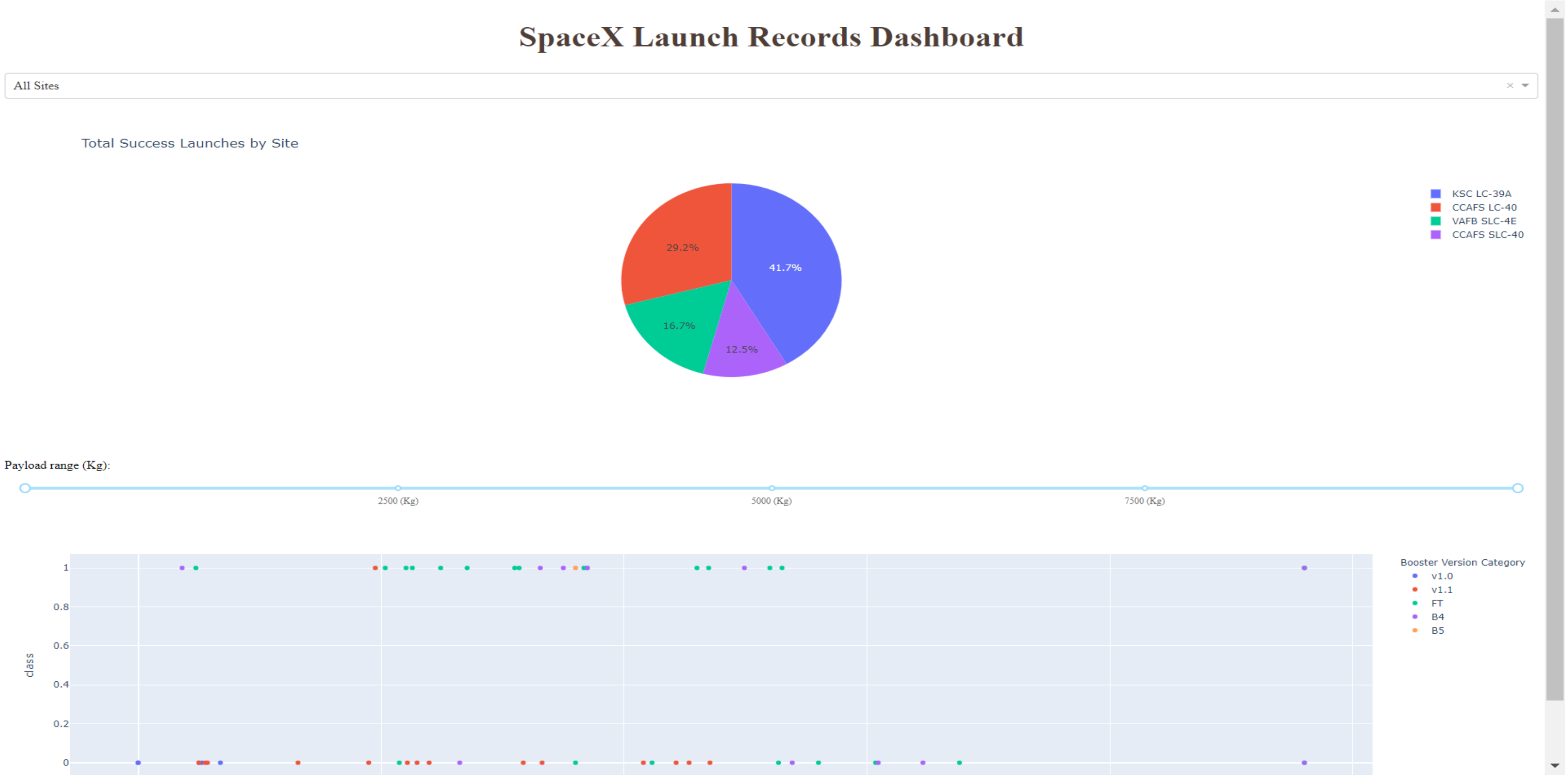
Scatter plot takes two inputs: All sites or individual site and payload mass on a slider between 0 and 10000 kg.

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.

GitHub url:
https://github.com/AnnGlineGomez/IBM_Applied_Capstone_Project_by_Ann_Gline_Gomez/blob/main/DashboardwithPlotyDash.ipynb

Results



SpaceX Launch Records Dashboard

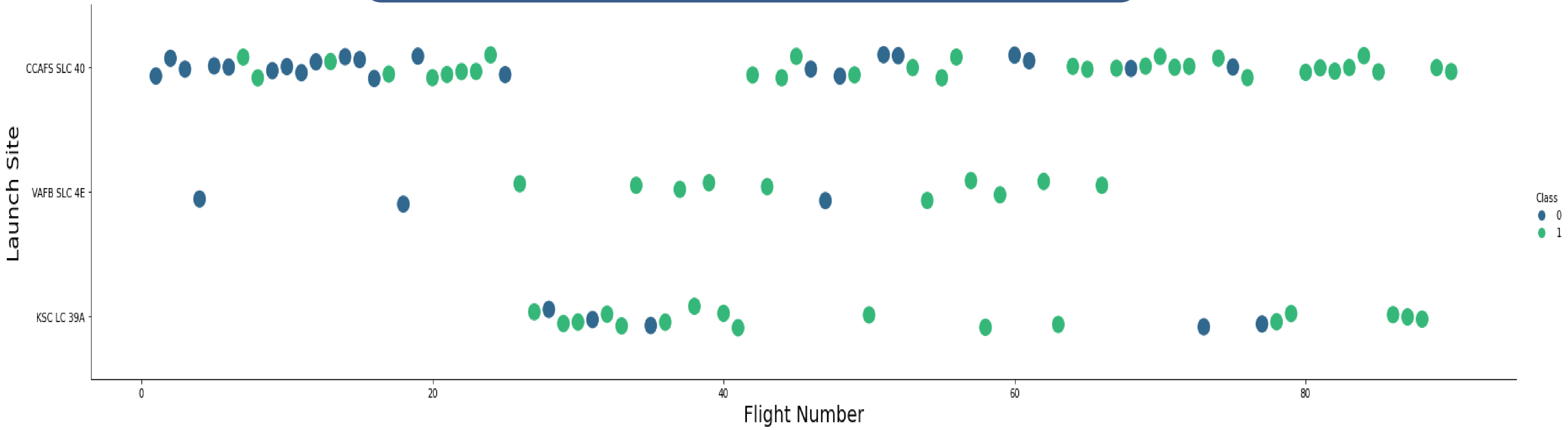
All Sites

Payload range (Kg):

Here's a glimpse of the Plotly dashboard preview. The upcoming slides will showcase the results of exploratory data analysis (EDA) through visualizations, EDA with SQL, an interactive map using Folium, and the final outcome of our model, demonstrating an accuracy rate of approximately 83%.

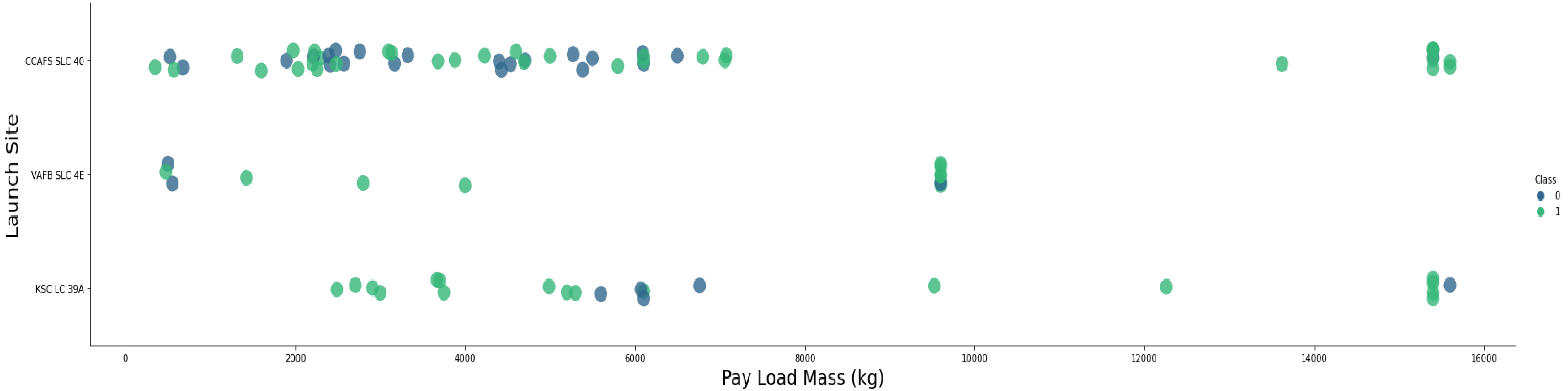
EDA with Visualization

Flight Number vs. Launch Site



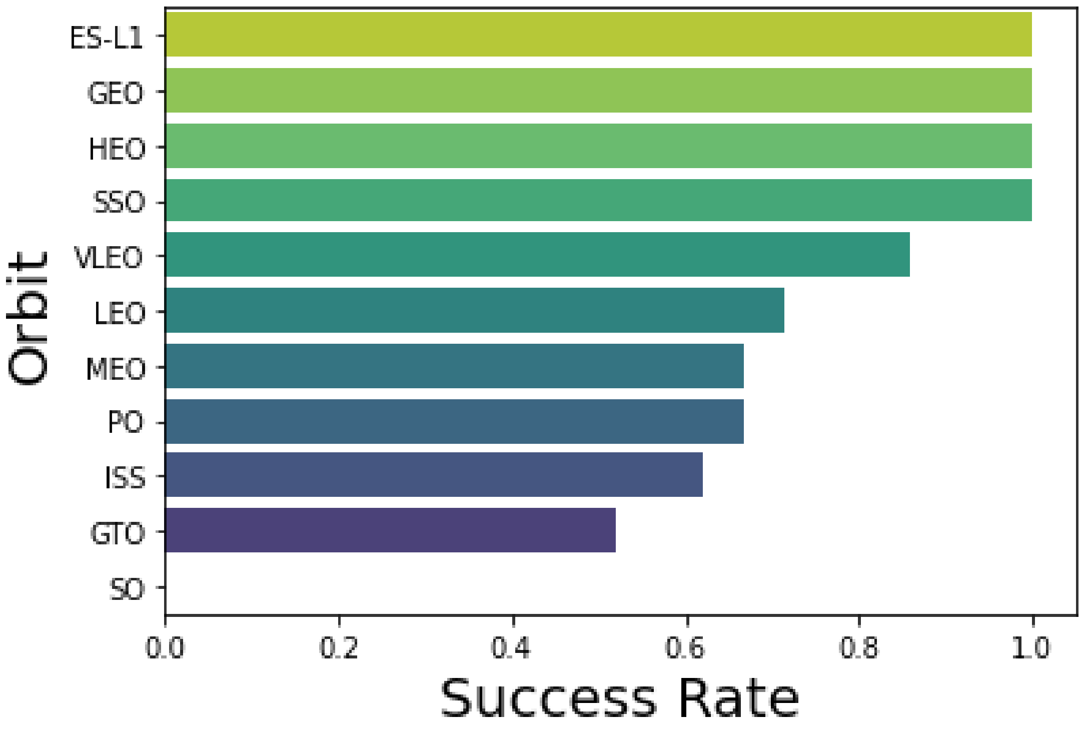
- Green represents successful launches, while purple represents unsuccessful launches.
- The graphic indicates an increasing success rate over time, with a significant breakthrough observed around flight 20.
- CCAFS stands out as the primary launch site with the highest volume of launches.

Payload vs. Launch Site



- Successful launches are depicted in green, while unsuccessful launches are represented in purple.
- The analysis of payload mass reveals that it predominantly falls within the range of 0-6000 kg.
- Additionally, it is observed that different launch sites have varying payload mass preferences.

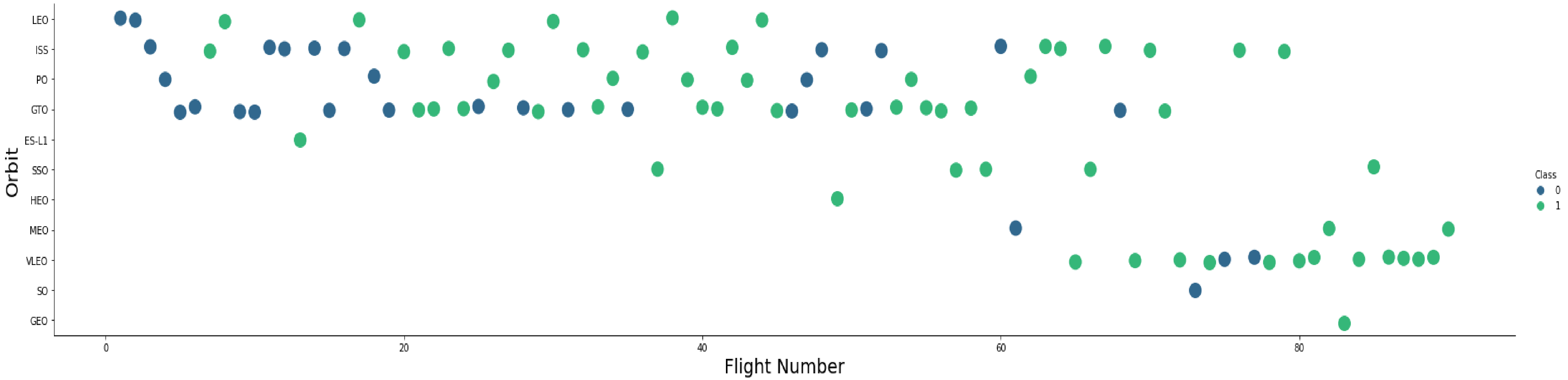
Success rate vs. Orbit type



- ES-L1 (1), GEO (1), HEO (1) have 100% success rate (sample sizes in parenthesis) SSO (5) has 100% success rate
- VLEO (14) has decent success rate and attempts
- SO (1) has 0% success rate
- GTO (27) has the around 50% success rate but largest sample

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

Flight Number vs. Orbittype

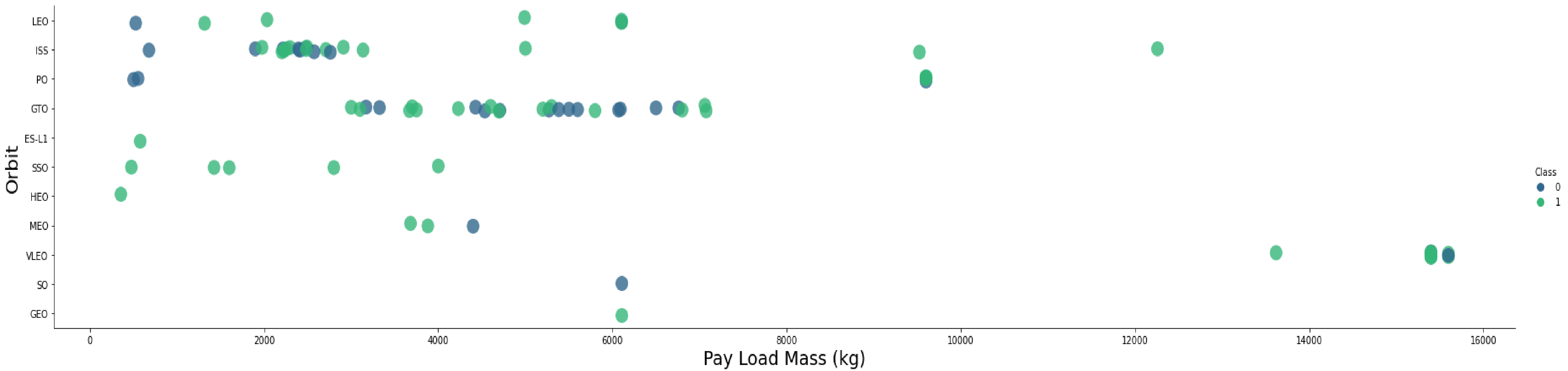


- Green indicates successful launch; Purple indicates unsuccessful launch.
- Launch Orbit preferences changed over Flight Number.

  Launch Outcome seems to correlate with this

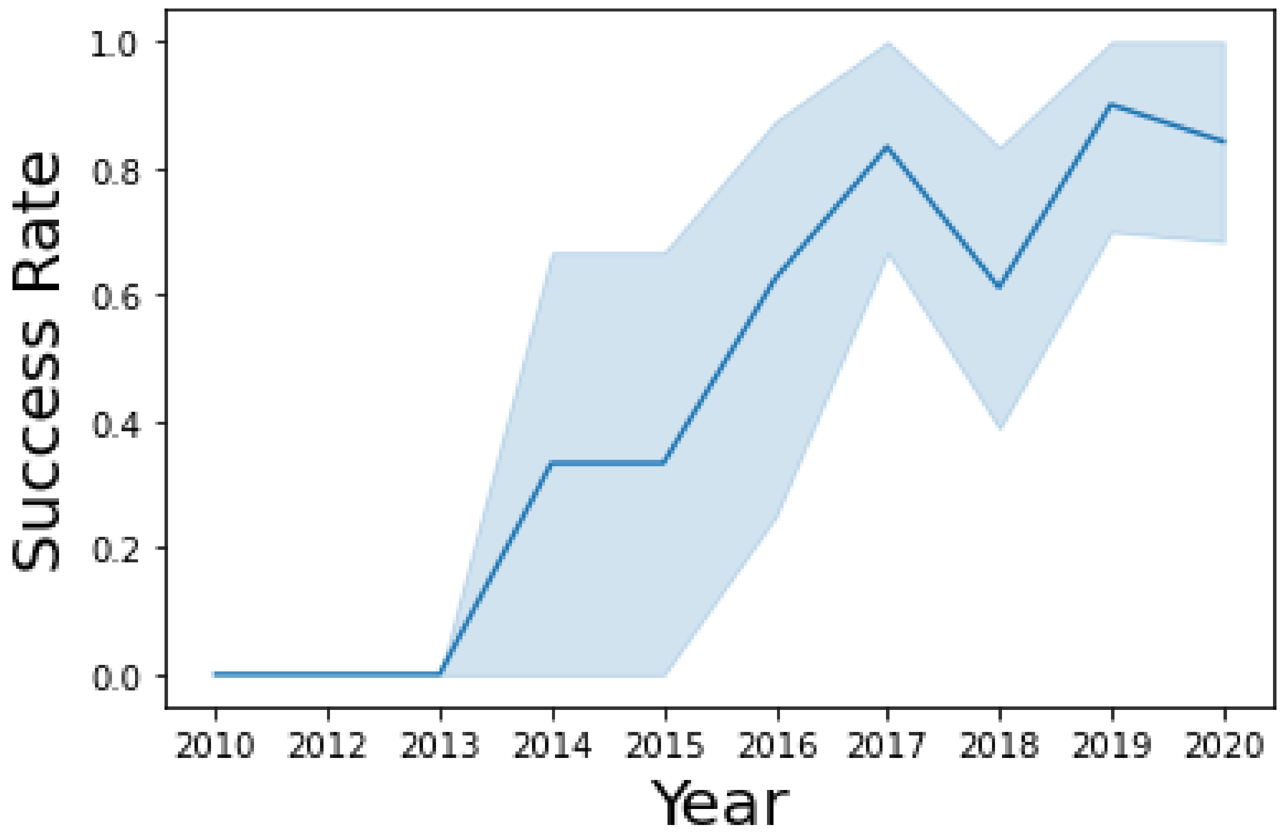
  preference.
- SpaceX started with LEO orbits which saw moderate success LEO and returned to VLEO in recent launches SpaceX appears to perform better in lower orbits or Sun-synchronous orbits

Payload vs. Orbit type



- Green indicates successful launch; Purple indicates unsuccessful launch.
- Payload mass seems to correlate with orbit
- LEO and SSO seem to have relatively low payload mass
- The other most successful orbit VLEO only has payload mass values in the higher end of the range

Launch Success Yearly Trend



95% confidence interval (light blue shading)

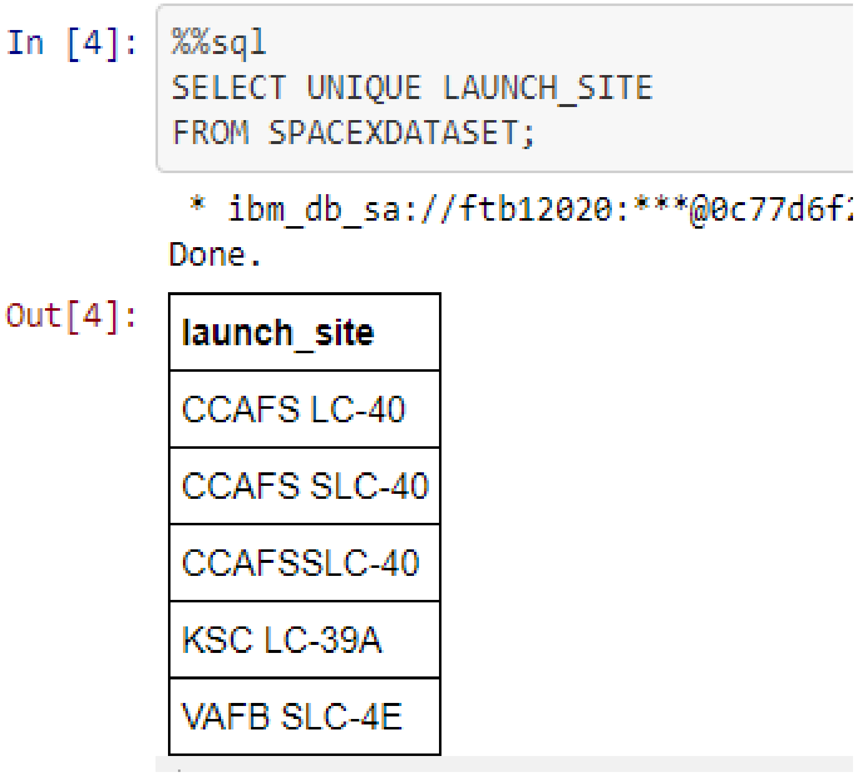
- Success generally increases over time since 2013 with a slight dip in 2018
- Success in recent years at around 80%

EDA with SQL

EXPLORATORY DATA ANALYSIS WITH SQL DB2

INTEGRATED IN PYTHON WITH SQLALCHEMY

All Launch Site Names



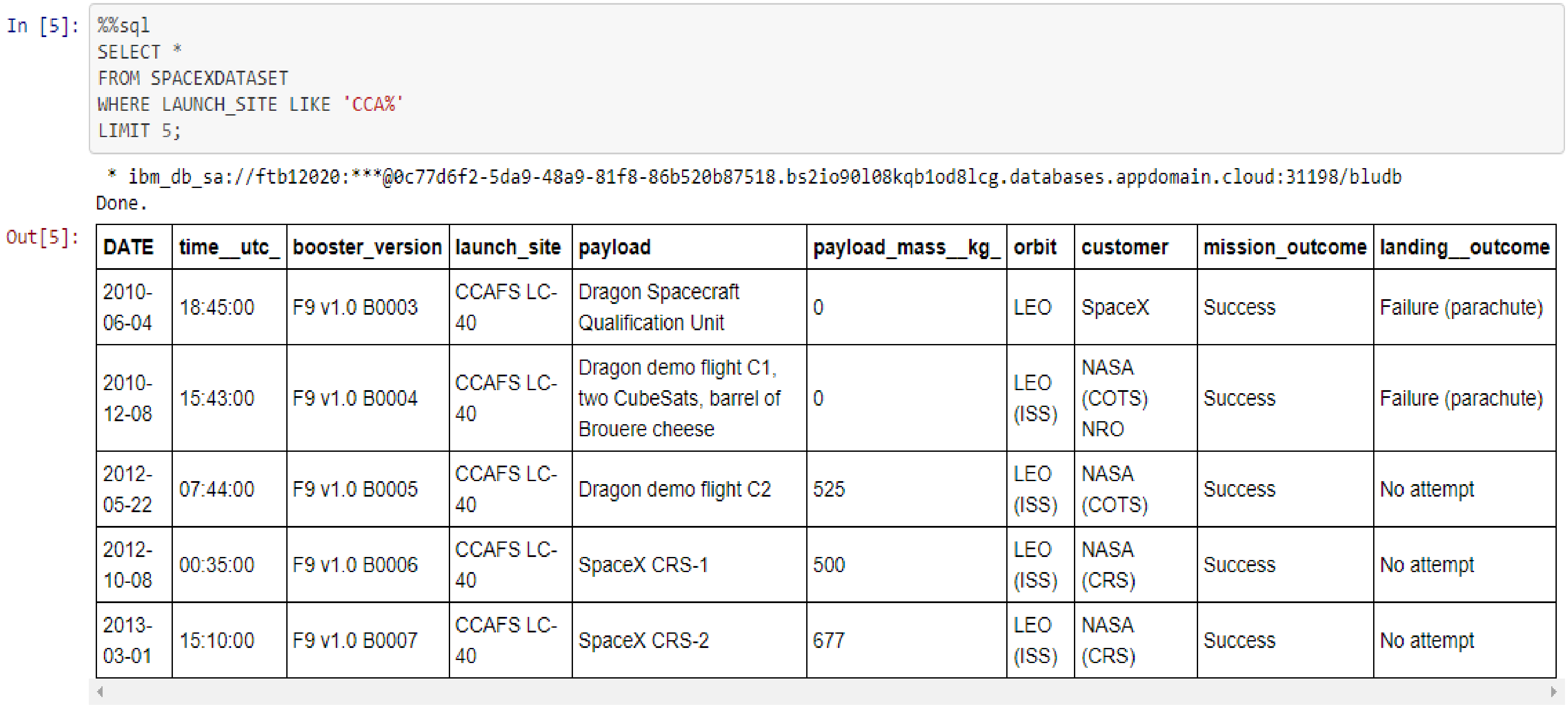
```
In [4]: %%sql
        SELECT UNIQUE LAUNCH_SITE
        FROM SPACEXDATASET;

         * 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;
```

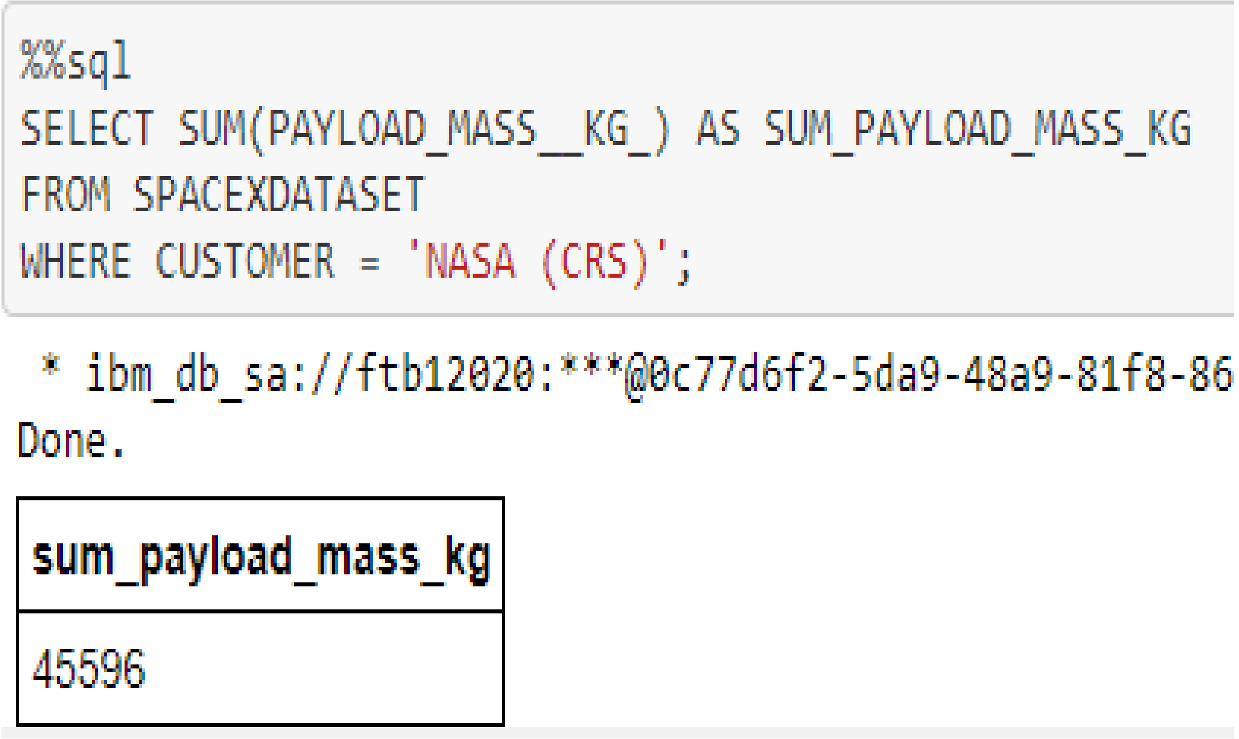
```
 * ibm_db_sa://ftb12020:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/bludb
Done.
```

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



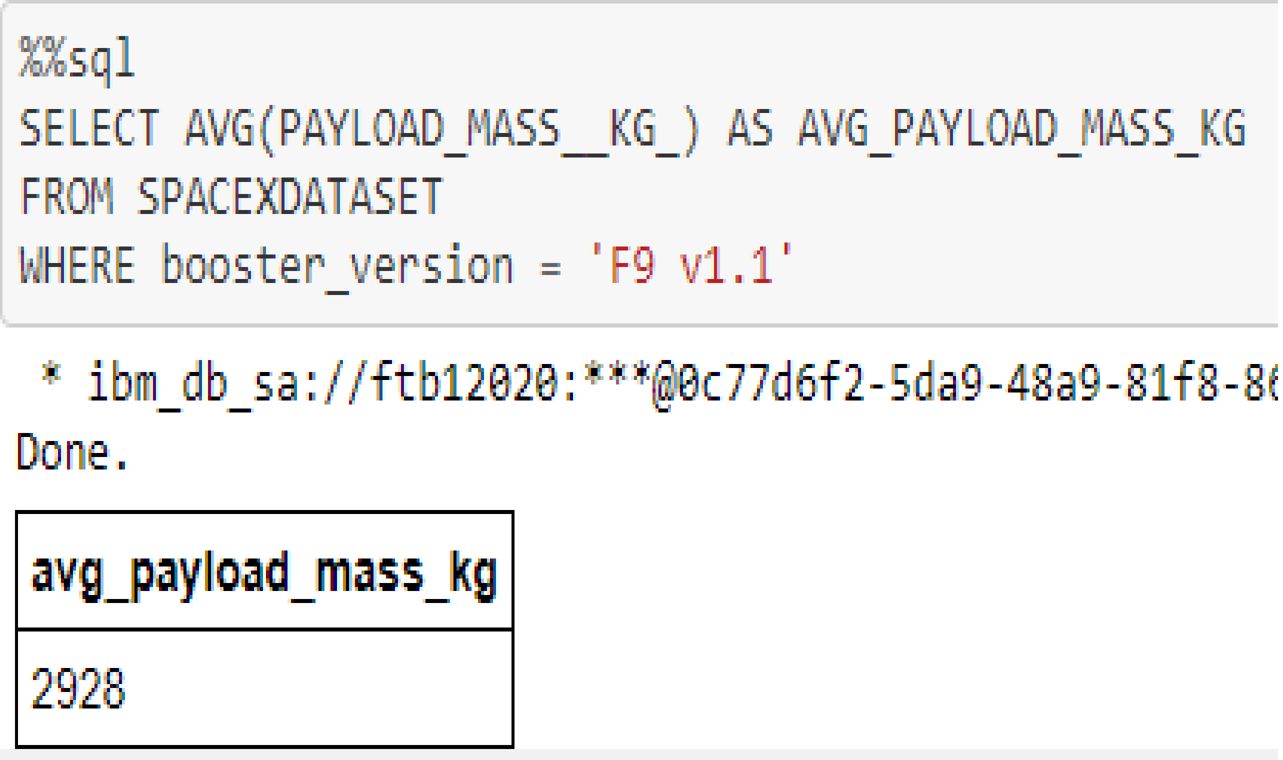
```
%%sql
SELECT SUM(PAYLOAD_MASS__KG_) AS SUM_PAYLOAD_MASS_KG
FROM SPACEXDATASET
WHERE CUSTOMER = 'NASA (CRS)';
```

```
 * 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



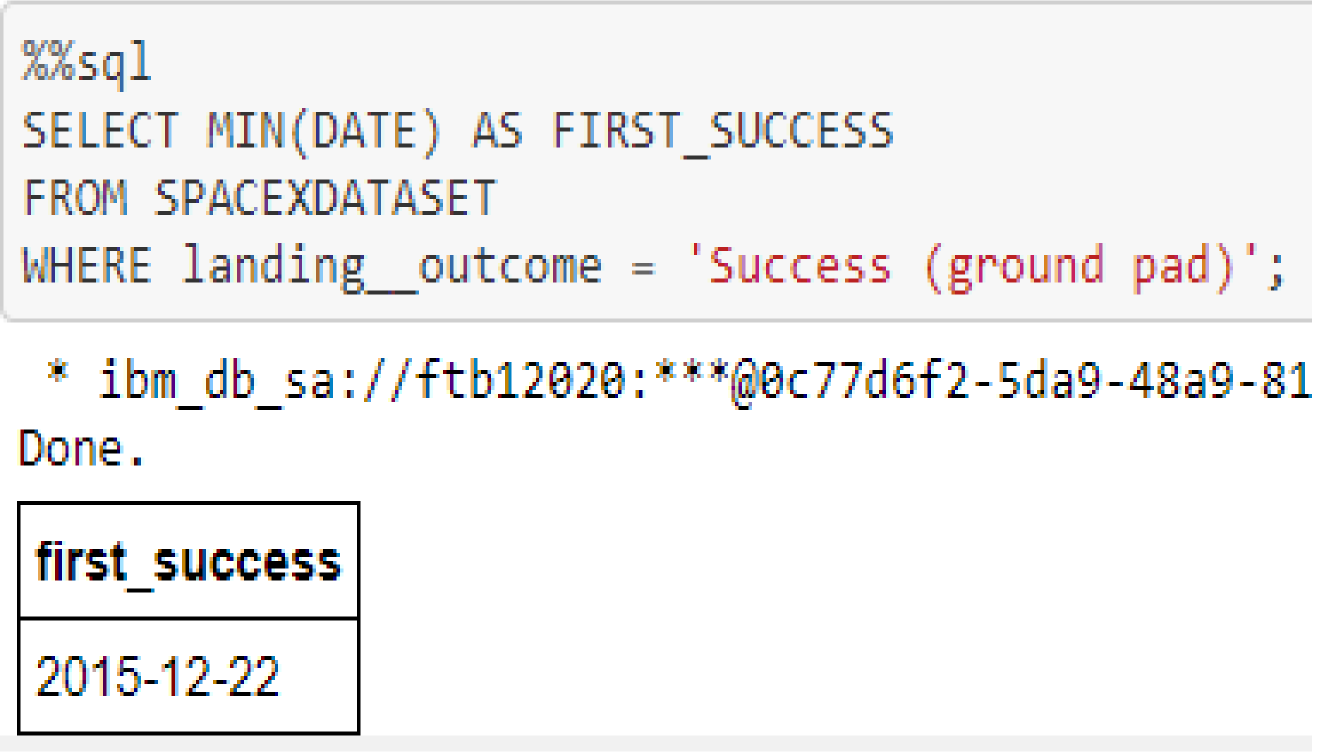
```
%%sql
SELECT AVG(PAYLOAD_MASS__KG_) AS AVG_PAYLOAD_MASS_KG
FROM SPACEXDATASET
WHERE booster_version = 'F9 v1.1'
```

```
 * ibm_db_sa://ftb12020:***@0c77d6f2-5da9-48a9-81f8-8
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



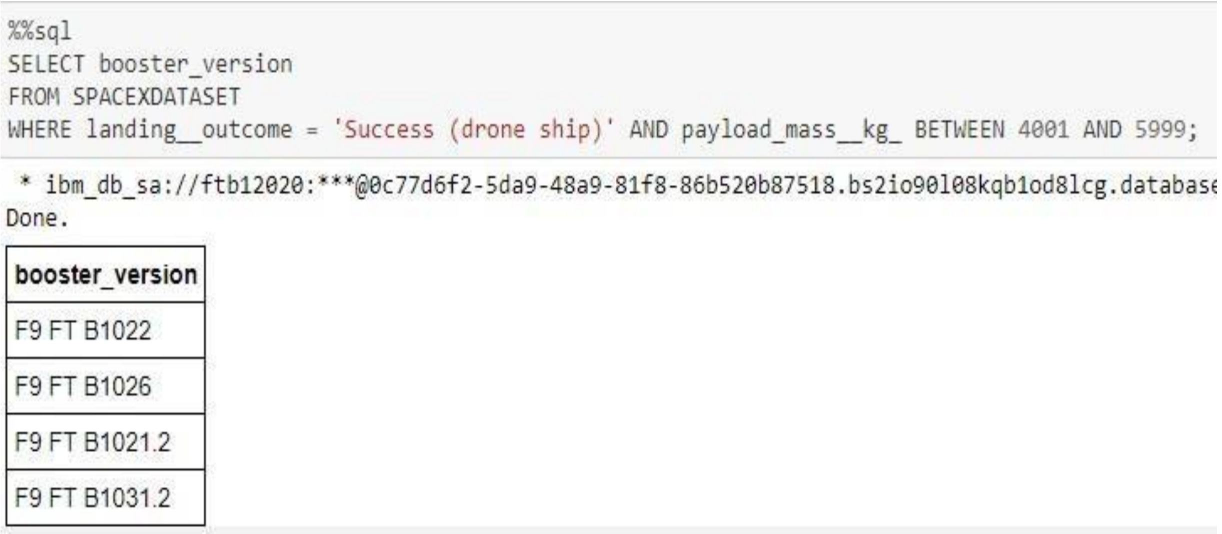
```
%%sql
SELECT MIN(DATE) AS FIRST_SUCCESS
FROM SPACEXDATASET
WHERE landing__outcome = 'Success (ground pad)';
```

```
 * 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



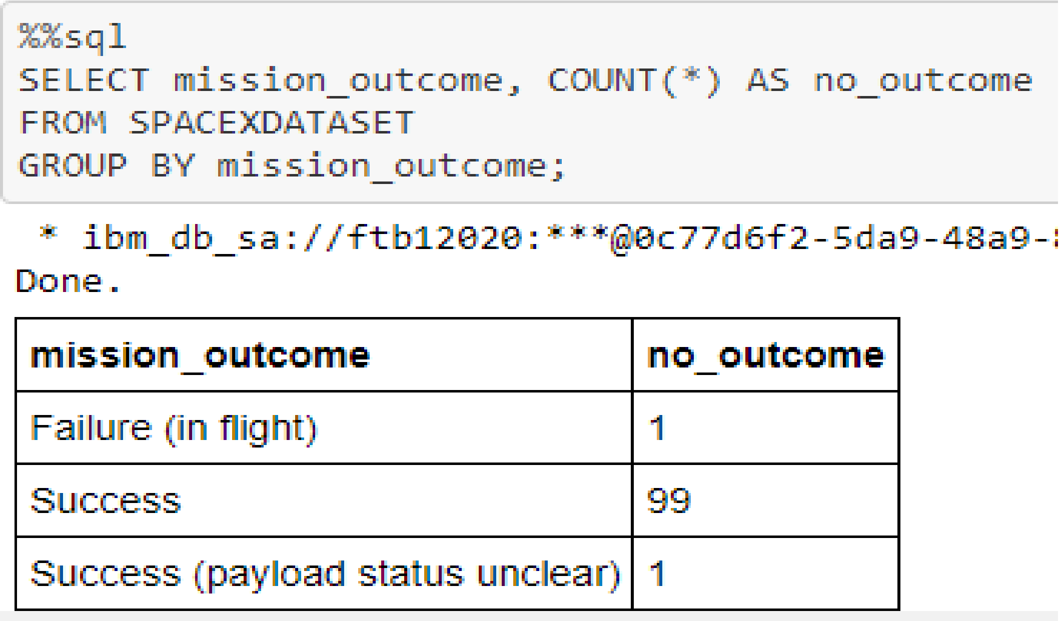
```
%%sql
SELECT booster_version
FROM SPACEXDATASET
WHERE landing__outcome = 'Success (drone ship)' AND payload_mass__kg_ BETWEEN 4001 AND 5999;
```

```
 * 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



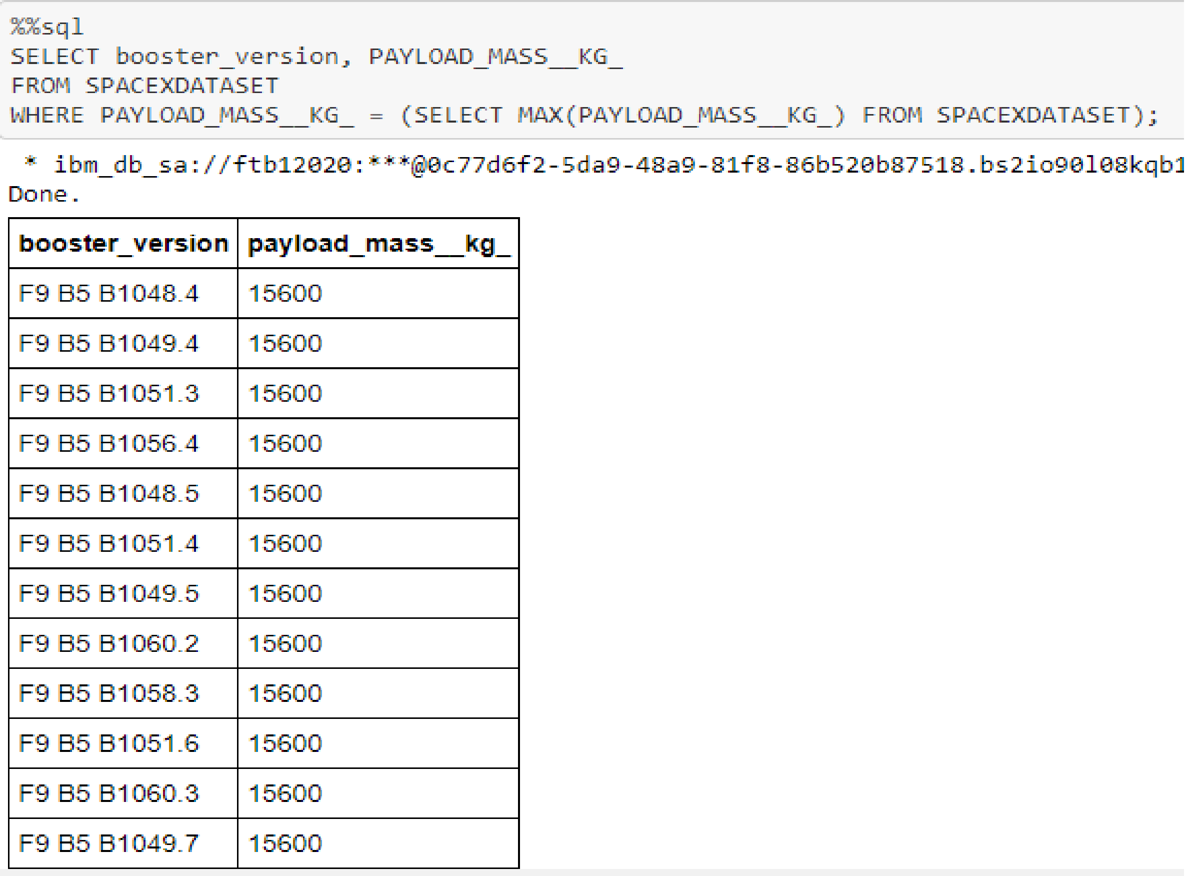
```
%%sql
SELECT mission_outcome, COUNT(*) AS no_outcome
FROM SPACEXDATASET
GROUP BY mission_outcome;
```

```
 * 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



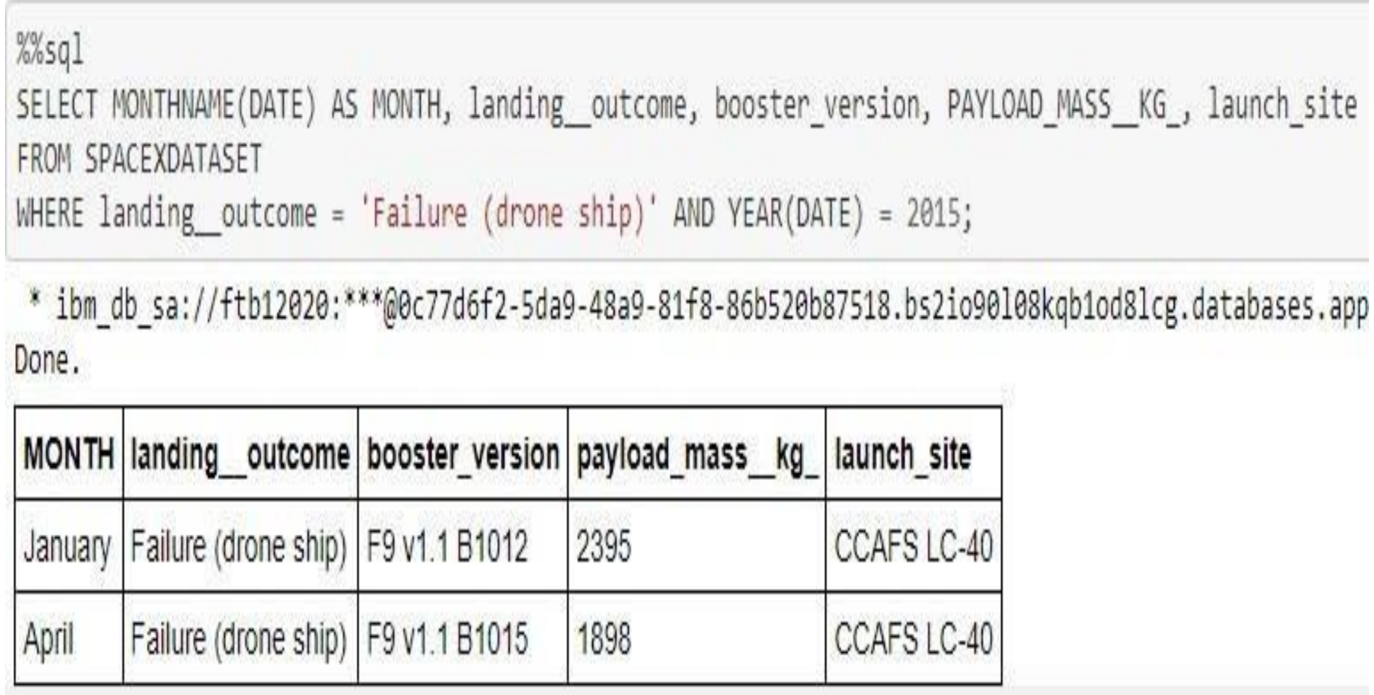
```
%%sql
SELECT booster_version, PAYLOAD_MASS__KG_
FROM SPACEXDATASET
WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXDATASET);
```

```
 * 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



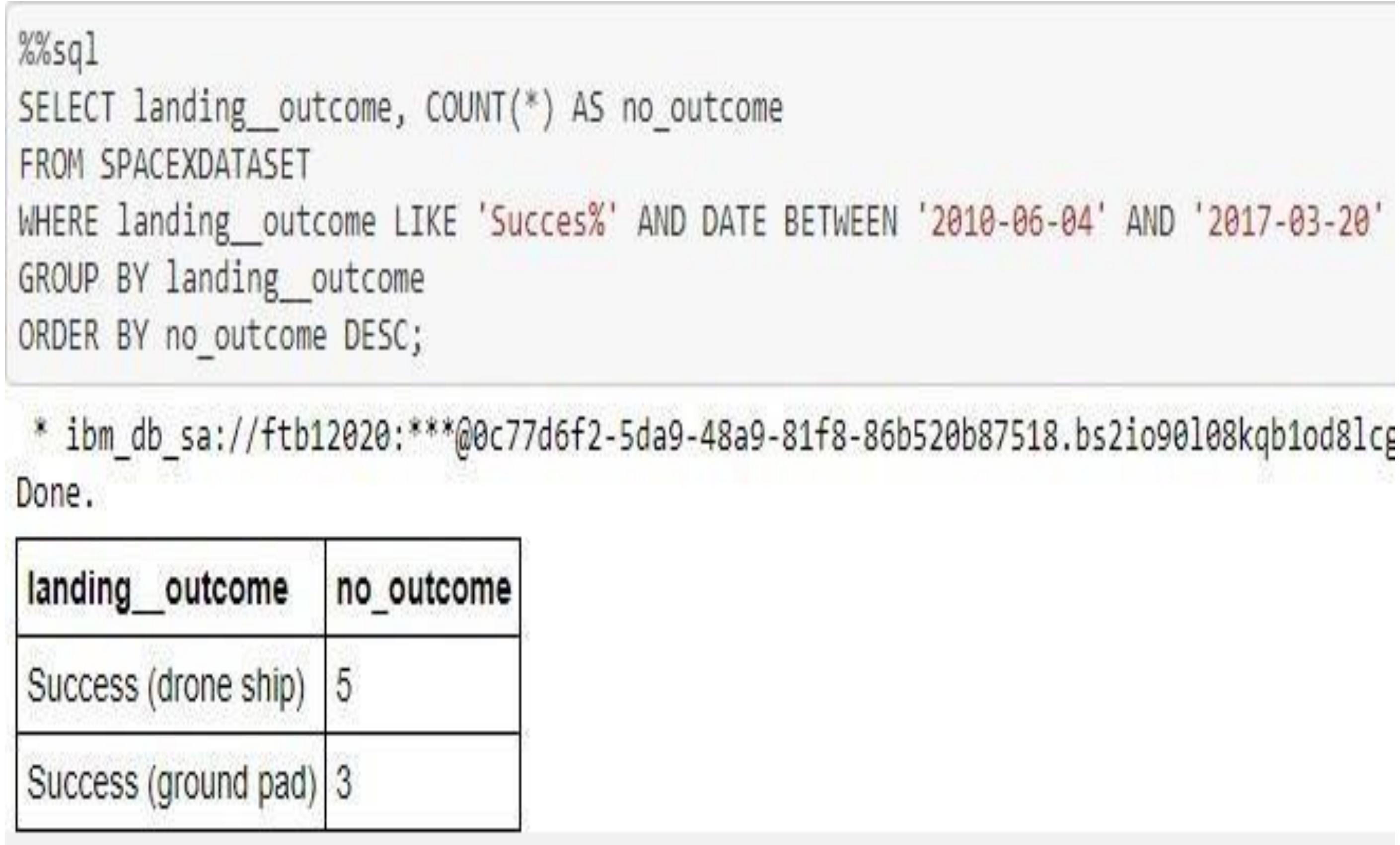
```
%%sql
SELECT MONTHNAME(DATE) AS MONTH, landing__outcome, booster_version, PAYLOAD_MASS__KG_, launch_site
FROM SPACEXDATASET
WHERE landing__outcome = 'Failure (drone ship)' AND YEAR(DATE) = 2015;
```

```
 * 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 SuccessfulLandings Between 2010-06-04 and2017-03-20



```
%%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;
```

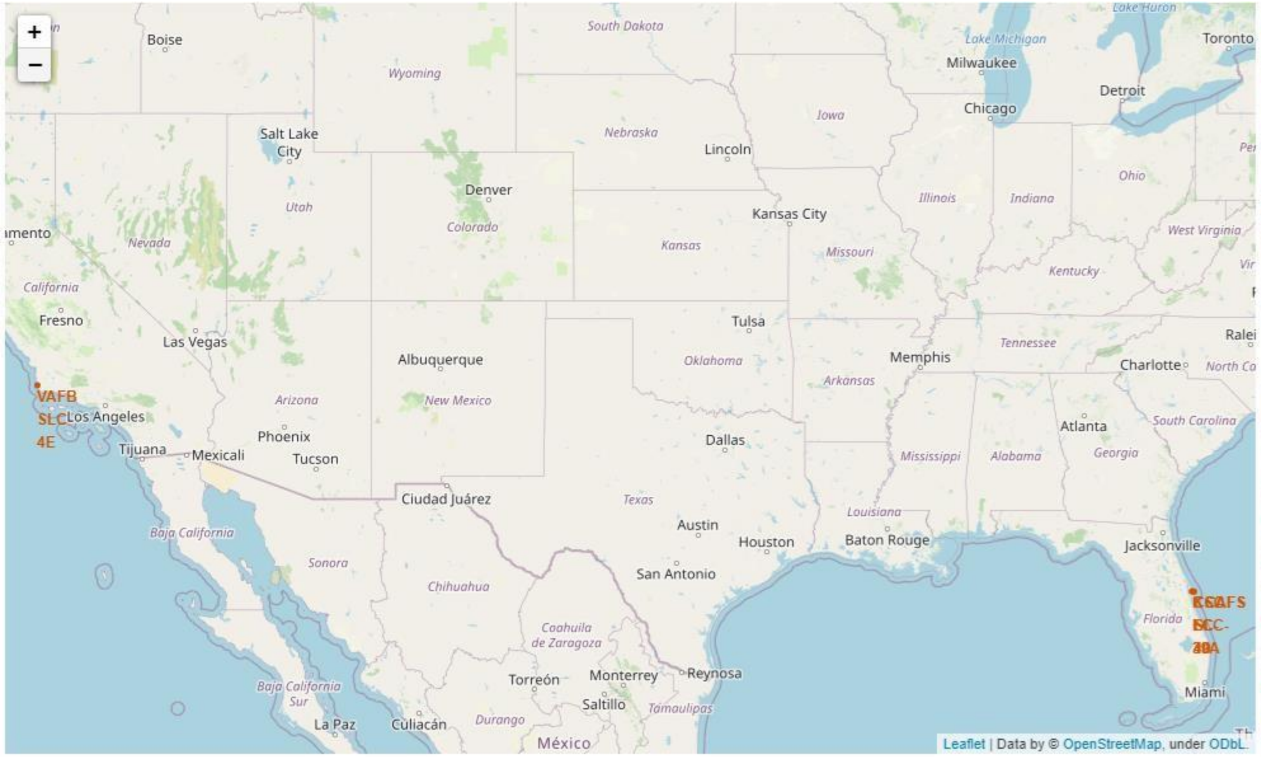
```
 * ibm_db_sa://ftb12020:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg
Done.
```

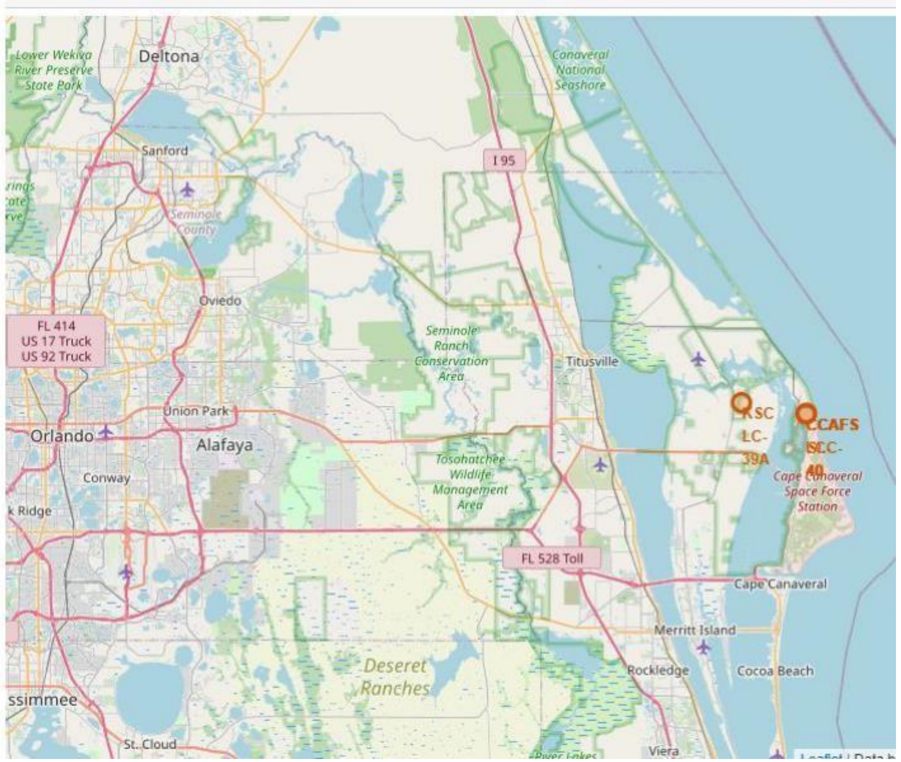
landing__outcome	no_outcome
Success (drone ship)	5
Success (ground pad)	3

- 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

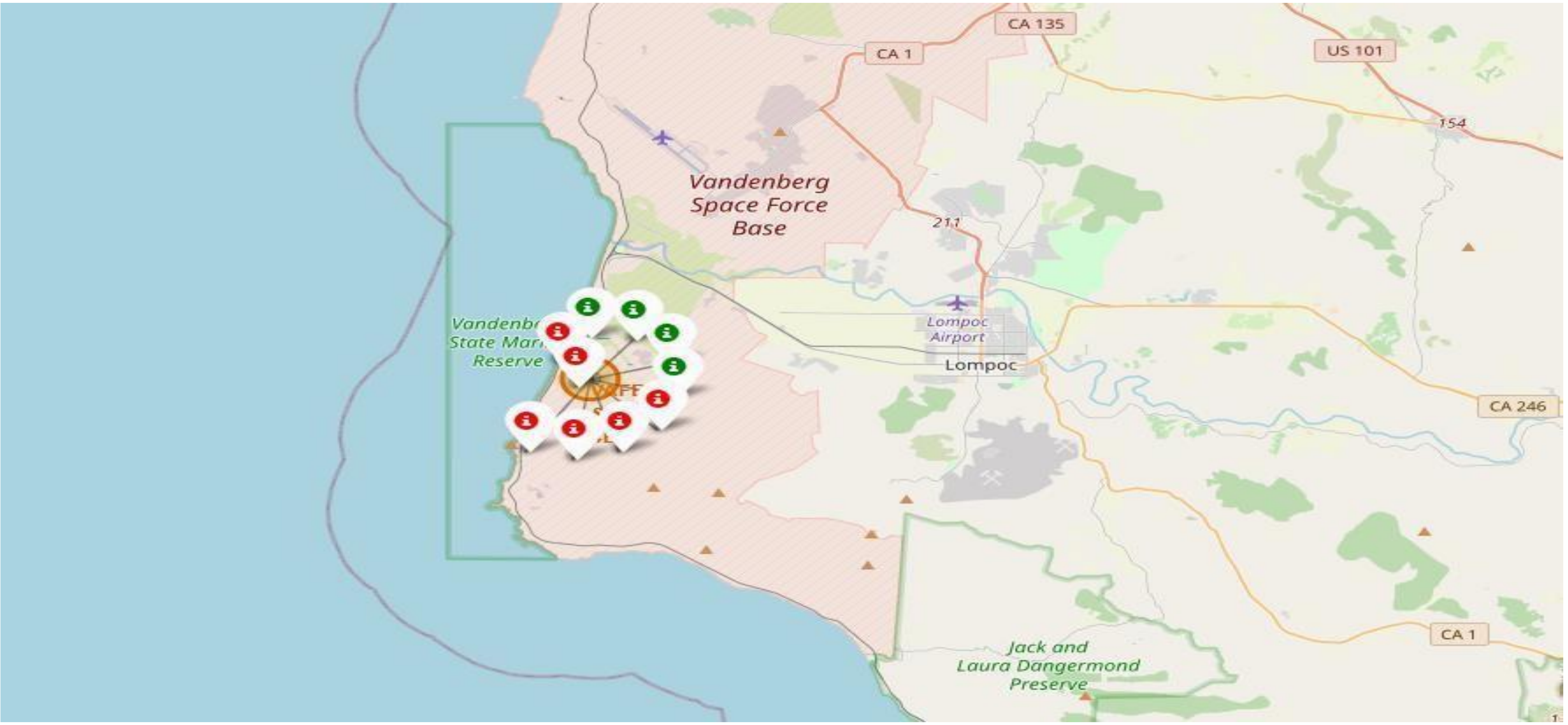
Launch Site Locations





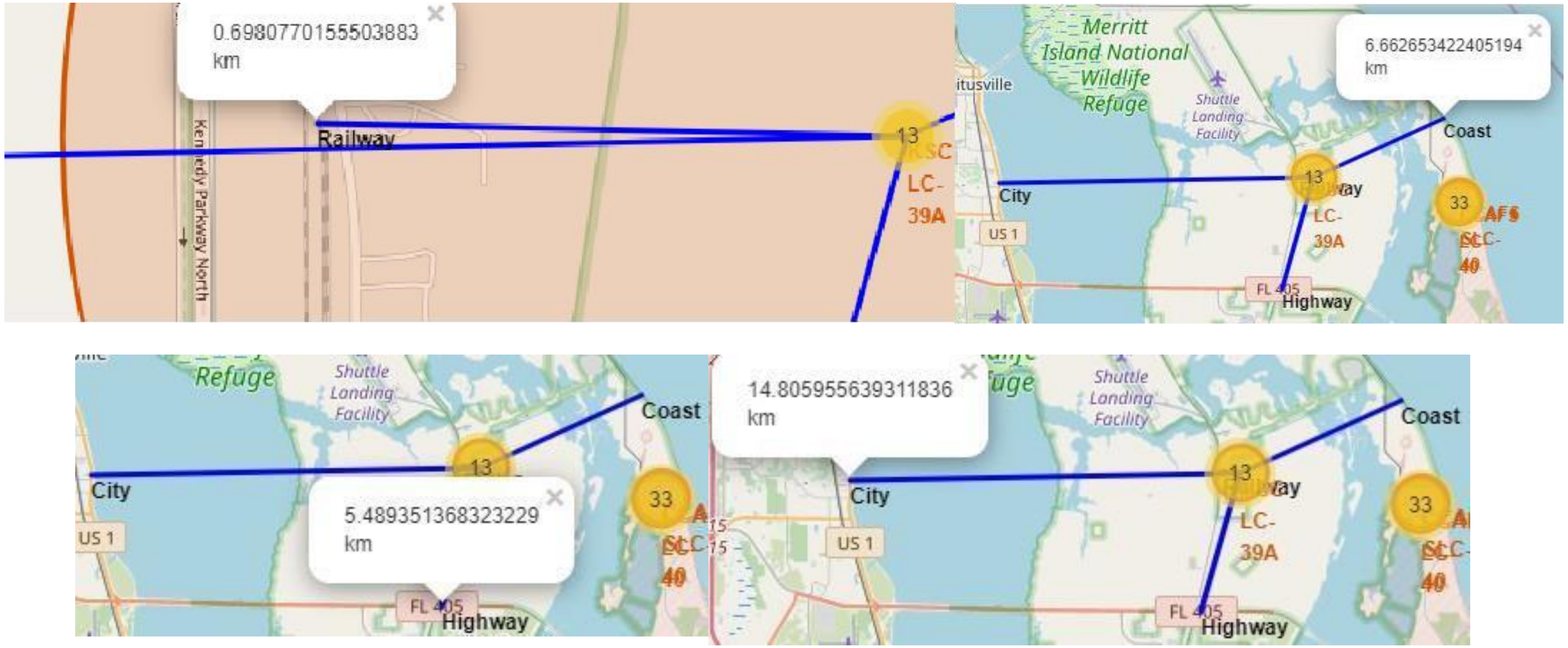
- The left map shows all launch sites relative US map.
- The right map shows the two Florida launch sites since they are very close to each other.
- All launch sites are near the ocean.

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
Vandenb State Mar Reserve

- 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 Proximities



- 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.

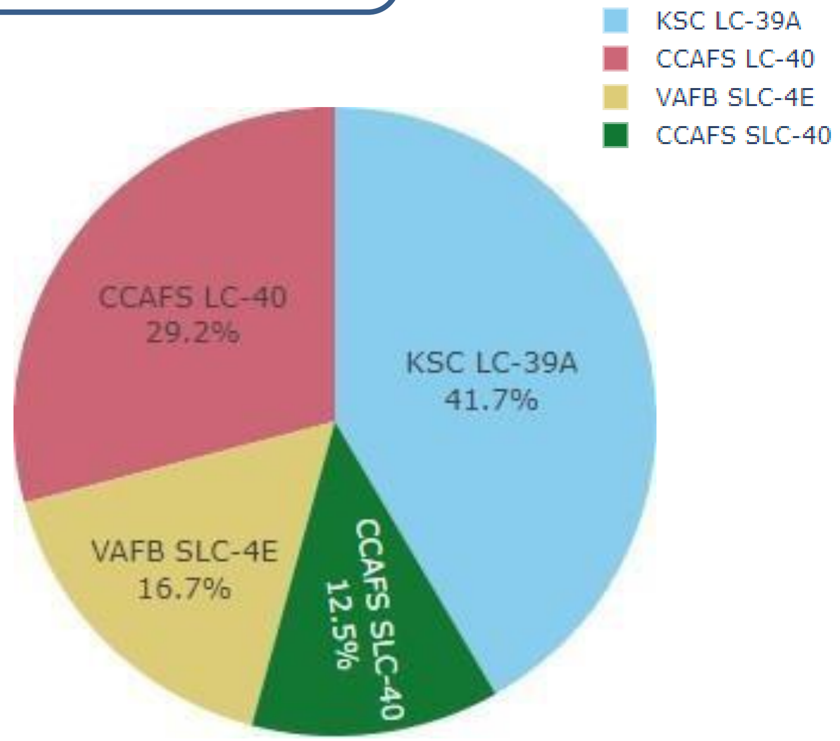
Build a Dashboard with Plotly Dash

Build a Dashboard with Plotly Dash

Successful Launches Across Launch Sites

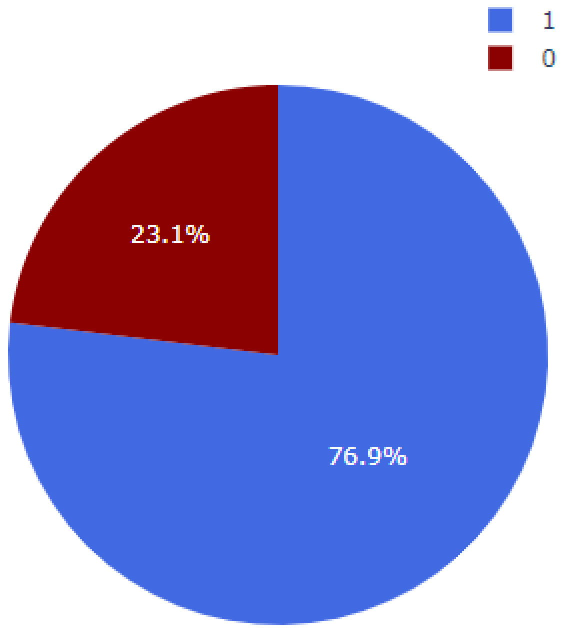
- This is the distribution of successful landings across all launch sites.
- CCAFS LC-40 is the old name of CCAFS SLC-40 so CCAFS and KSC have the same amount of successful landings, but a majority of the successful landings where performed before the name change.

- 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.



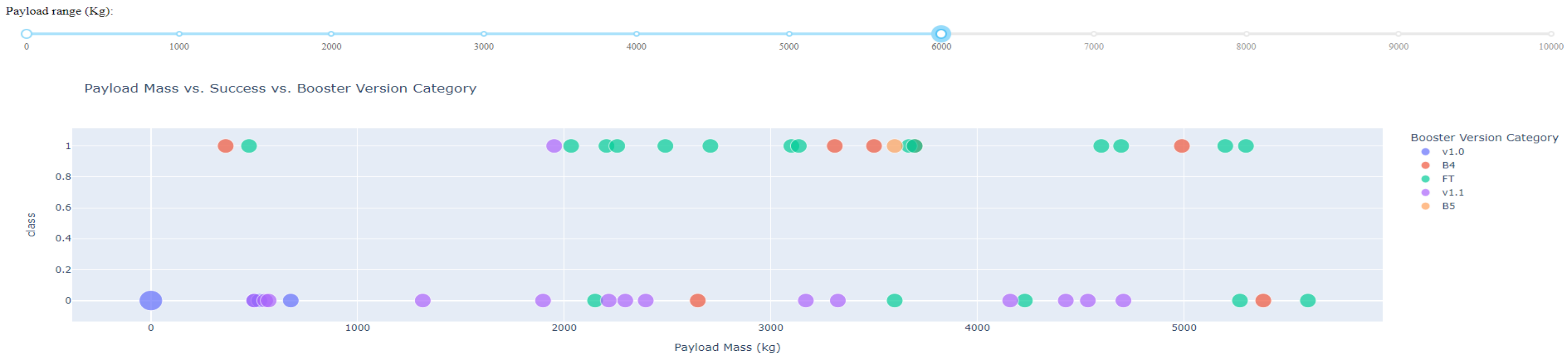
Highest Success Rate Launch Site

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



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

Payload Mass vs. Success vs. Booster Version Category



Payload range (Kg):

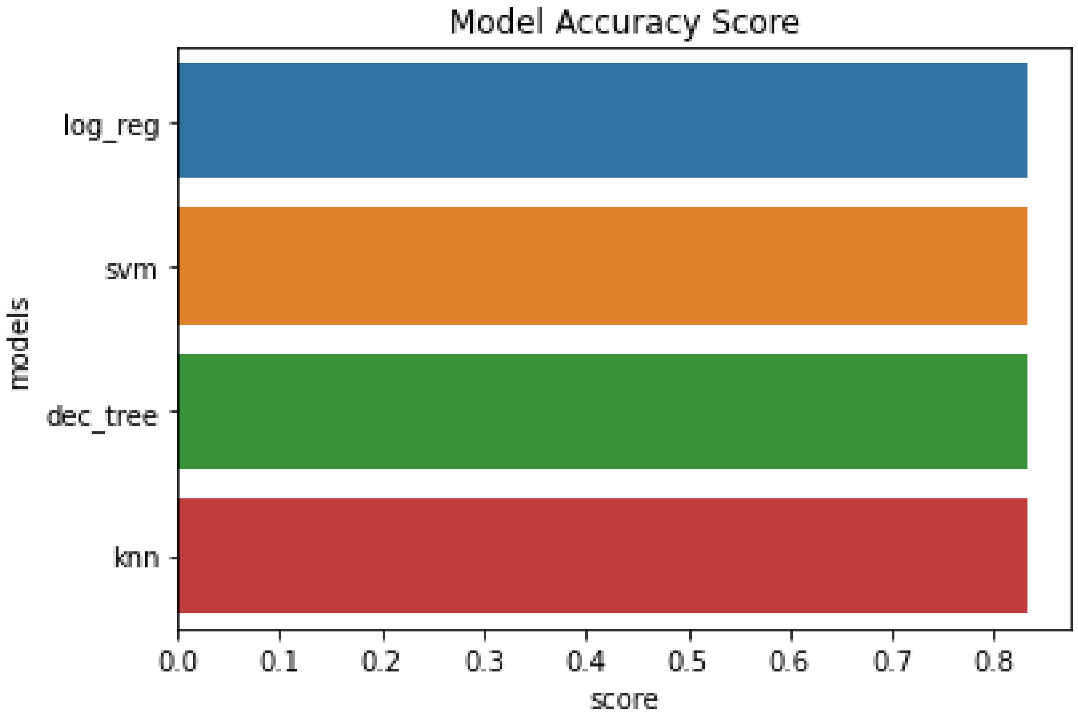
- Plotly dashboard has a Payload range selector. However, this is set from 0-10000 instead of the max Payload of 15600.
- Class indicates 1 for successful landing and 0 for failure. Scatter plot also accounts for booster version category in color and number of launches in point size.
- In this particular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

Predictive Analysis (Classification)

GRIDSEARCHCV(CV=10) ON LOGISTIC REGRESSION, SVM, DECISION TREE, AND KNN

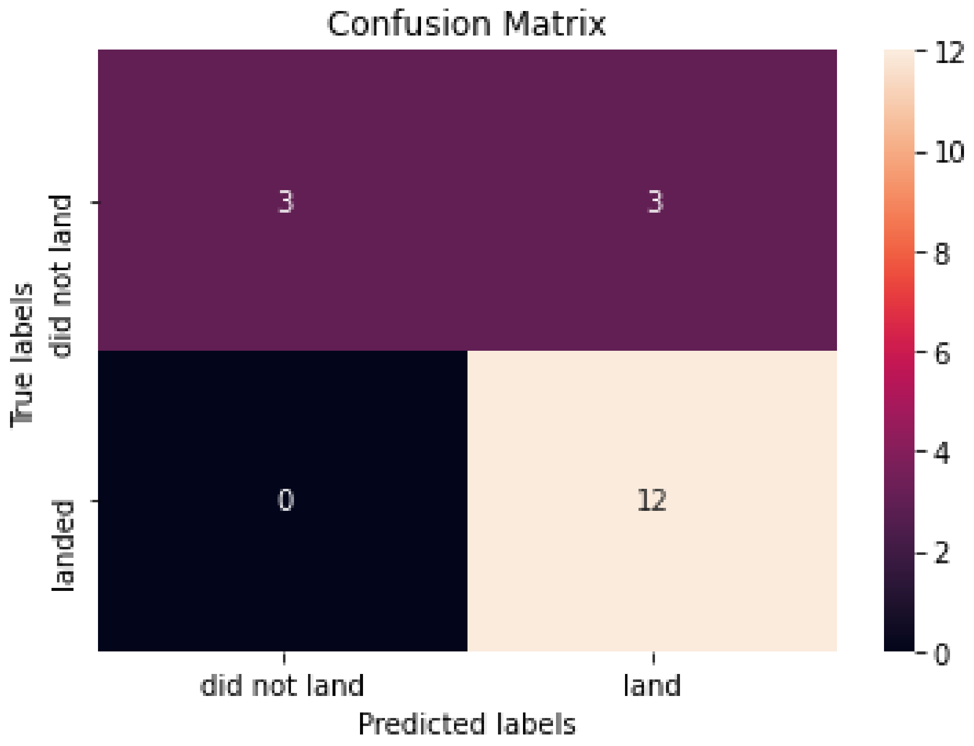
Classification Accuracy

- 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

- 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.



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

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

Appendix

GitHub repository url:

https://github.com/AnnGlineGomez/IBM_Applied_Capstone_Project_by_Ann_Gline_Gomez

Instructors:
Instructors: Rav Ahuja, Alex Aklson, Aije Egwaikhide, Svetlana Levitan, Romeo Kienzler, Polong Lin, Joseph Santarcangelo, Azim Hirjani, Hima Vasudevan, Saishruthi Swaminathan, Saeed Aghabozorgi, Yan Luo

Special Thanks to All Instructors:

https://www.coursera.org/professional-certificates/ibm-data-science?#instructors