Winning The New Space Race With Data Science

Gerardo Rivera

20/01/2024

IBM Developer SKILLS NETWORK

Outline

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

Executive Summary

Summary of Methodologies

Data Collection

Data Wrangling

Exploratory Data Analysis (EDA) With Data Visualization

EDA With SQL

Interactive Visual Analytics With Folium

Machine Learning Predictive Analysis

Results Summary

EDA Results

Interactive Analytics Screenshots

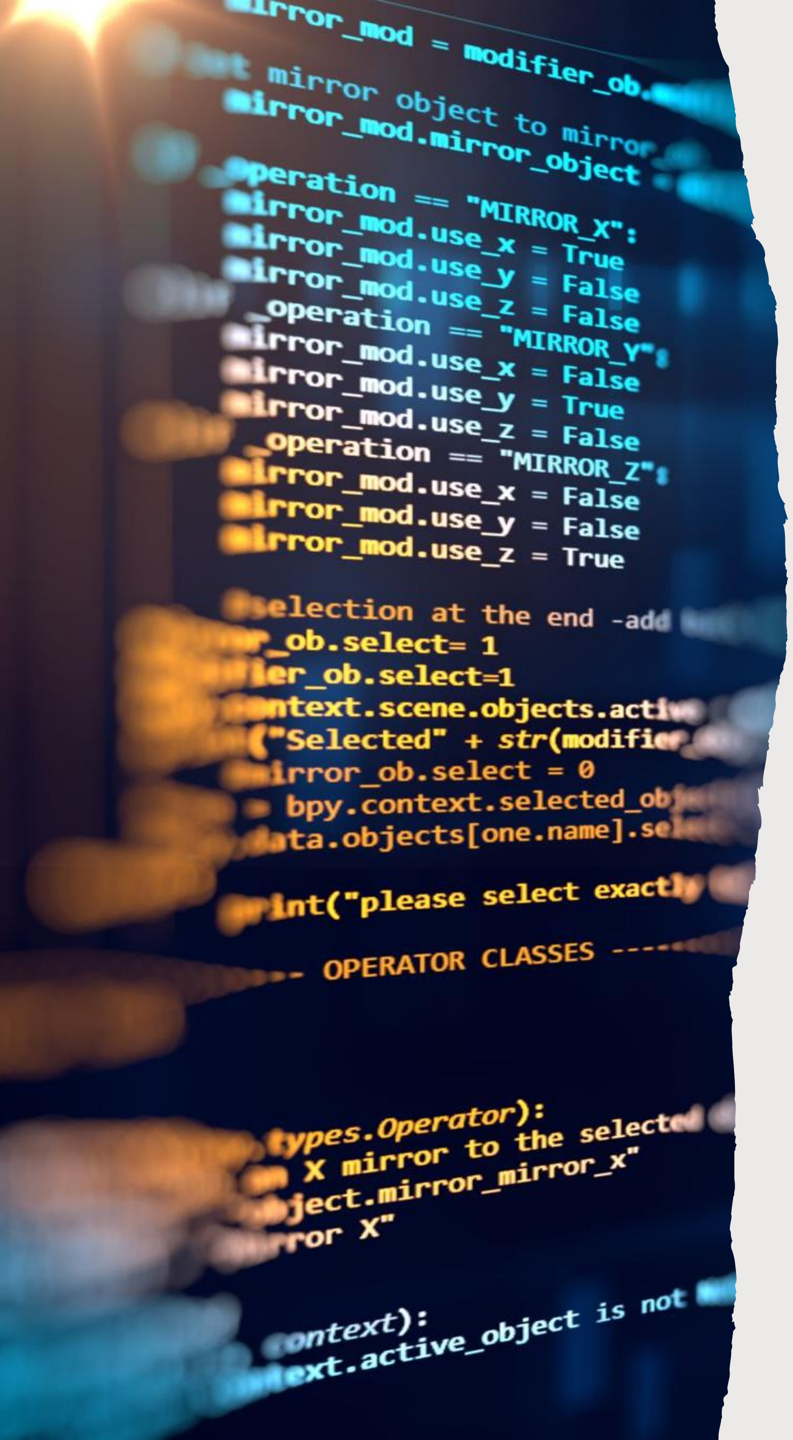
Predictive Analysis Results

Introduction

We are in the beginnings of a New Space Race with SpaceX making major strides in reducing launch costs. SpaceX's Reusable Launch System or RLS technology has dramatically reduced their launch costs. Advertising a Falcon 9 launch for as low as 62 million dollars at a time competitors are charging over 165 million. This price is contingent that the first stage lands safely and is recoverable. Utilizing public records, we will analyze the data using machine learning models to predict if SpaceX will reuse a RLS rocket.

We will set out to answer the following

- Identify what variables influence landing outcomes
- Relationship between variables and landing outcomes
- Best conditions for a successful landing outcome



Methodology

Data Collecting Methodology

- Using SpaceX Rest API
- Using Web Scrapping from Wikipedia’s List of Falcon 9 & Heavy Launches

Performed Data Wrangling

- Filtering data
- Cleaning data
- Using One Hot Encoding for categorizing data

Performed Exploratory Data Analysis using visualization and SQL

Performed Interactive Visual Analytics using Folium and Ploty Dash

Performed Predictive Analysis using classification models

- Build, Tune, and Evaluate classification models

Data Collection

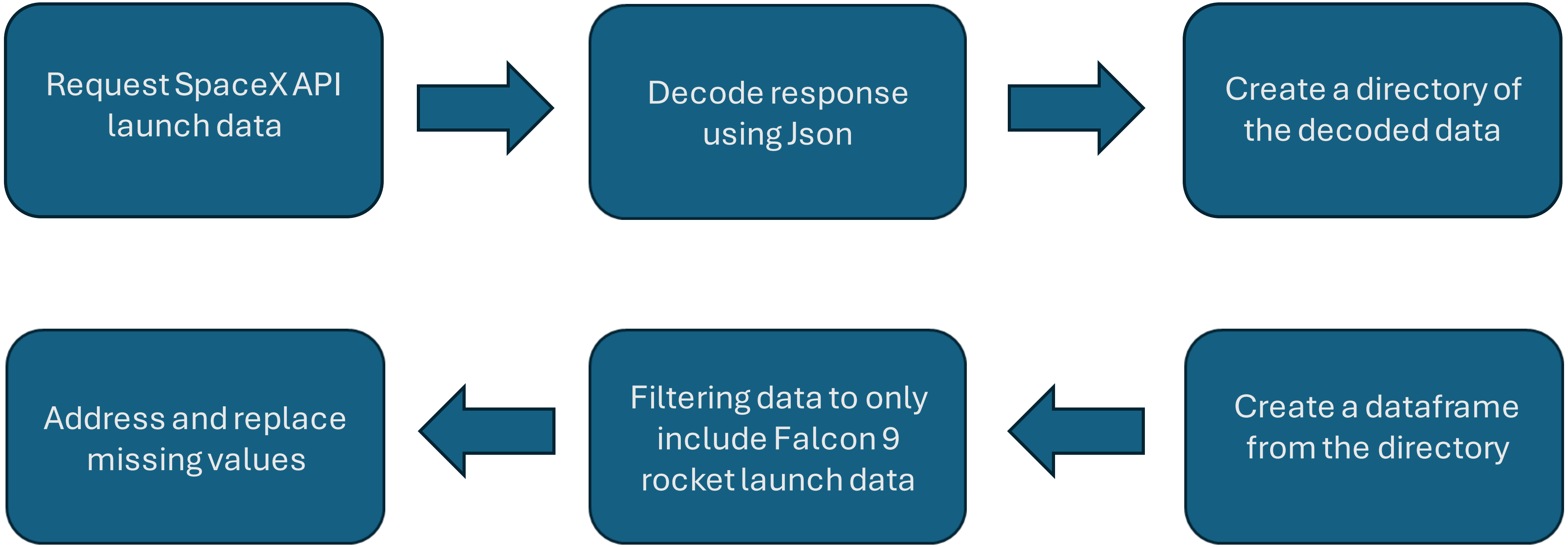
Data collection was done utilizing API requests from SpaceX Rest API and Web Scrapping from Wikipedia's entry on SpaceX Falcon 9 launches. Using both sources we were able to build a more complete data set to work from.

SpaceX Rest API provided us data such as FlightNumber, LaunchSite, LaunchDate, BoosterVersion, Payload, etc. This was done through a get request, using Json to decode the raw data and fit into a Pandas Dataframe. The data was then cleaned, and missing values handled appropriately.

Wikipedia also provided data such as Flight No, Launch Site, Launch Date, Booster Version, Payload, etc. The data was extracted using BeautifulSoup and then placed into a Pandas dataframe for analysis.

Data Collection – SpaceX API

Utilizing SpaceX's public API we followed the data collection process outlined below



Request SpaceX API launch data

Decode response using Json

Create a directory of the decoded data

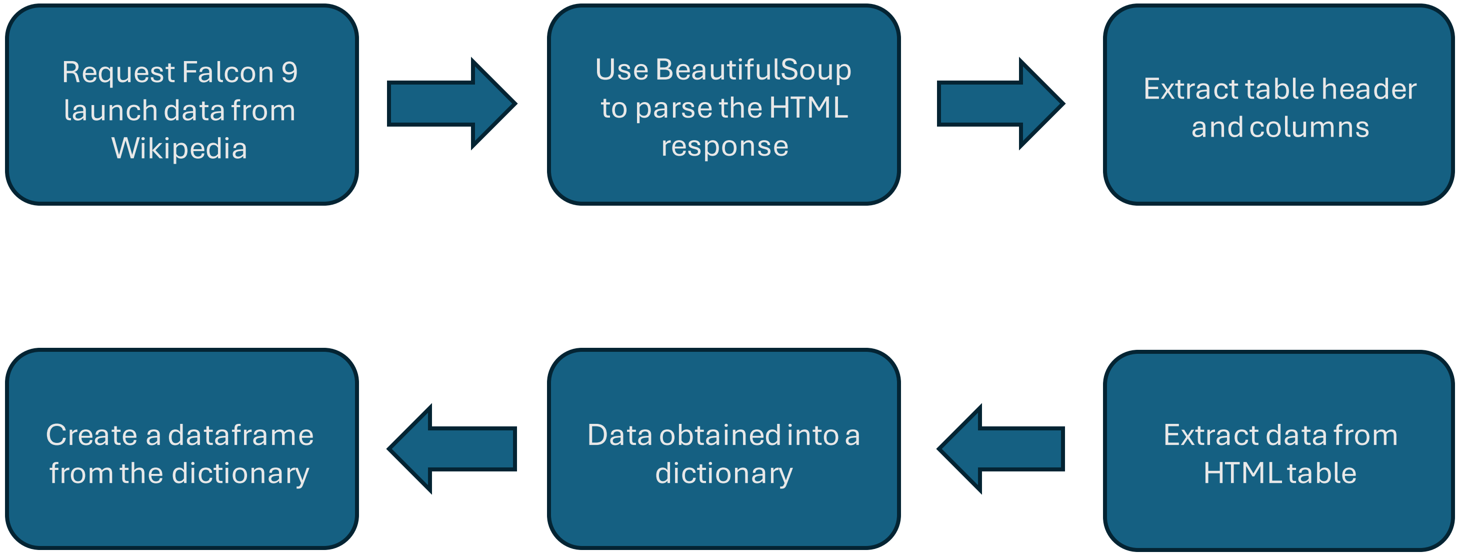
Create a dataframe from the directory

Filtering data to only include Falcon 9 rocket launch data

Address and replace missing values

https://github.com/NativeLag/Capstone-Project-IMBD/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

Data Collection - Scraping

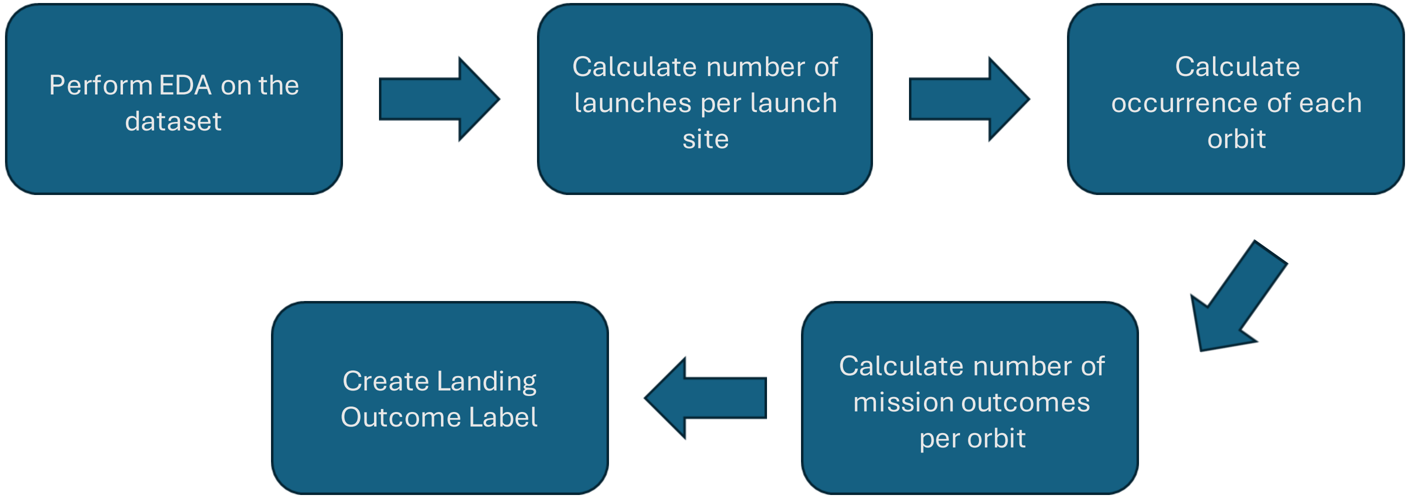

Request Falcon 9 launch data from Wikipedia
Use BeautifulSoup to parse the HTML response
Extract table header and columns
Extract data from HTML table
Data obtained into a dictionary
Create a dataframe from the dictionary

Data was also gathered from Wikipedia which contains a detailed list of Falcon 9 launches. The processing of which is outlined above

https://github.com/NativeLag/Capstone-Project-IMBD/blob/main/jupyter-labs-webscraping.ipynb

Data Wrangling

Exploratory Data Analysis (EDA) was performed on the datasets. Launches per site, occurrence of each orbit, and number of missions and outcomes were all calculated. This was used to create a Landing Outcome Label. The process is outlined below.



Perform EDA on the dataset

Calculate number of launches per launch site

Calculate occurrence of each orbit

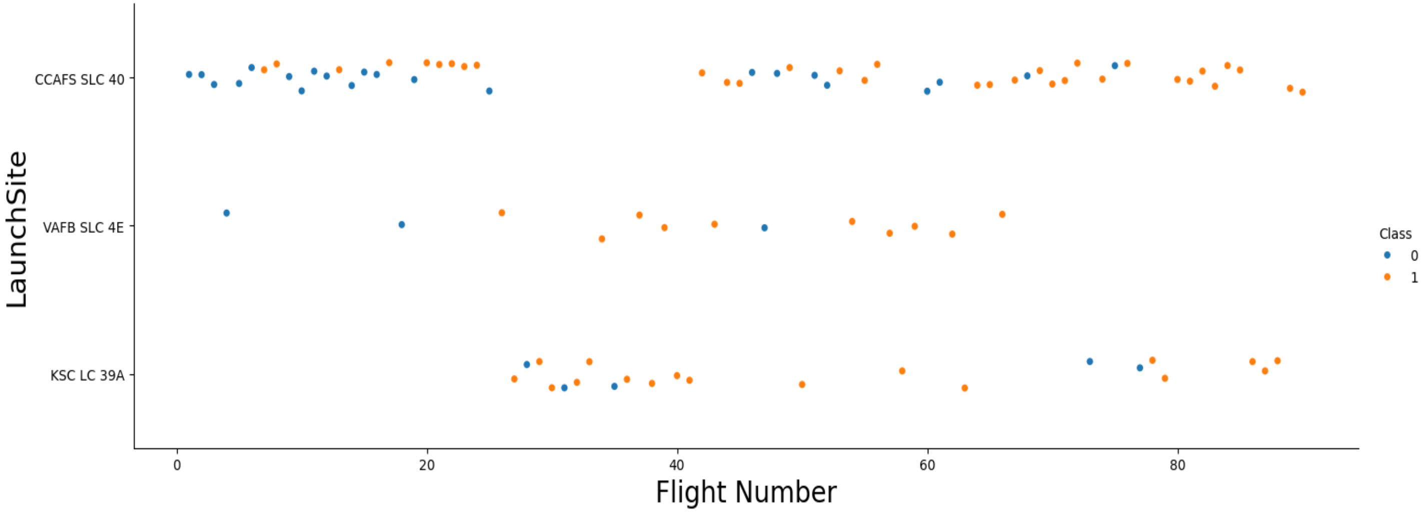
Calculate number of mission outcomes per orbit

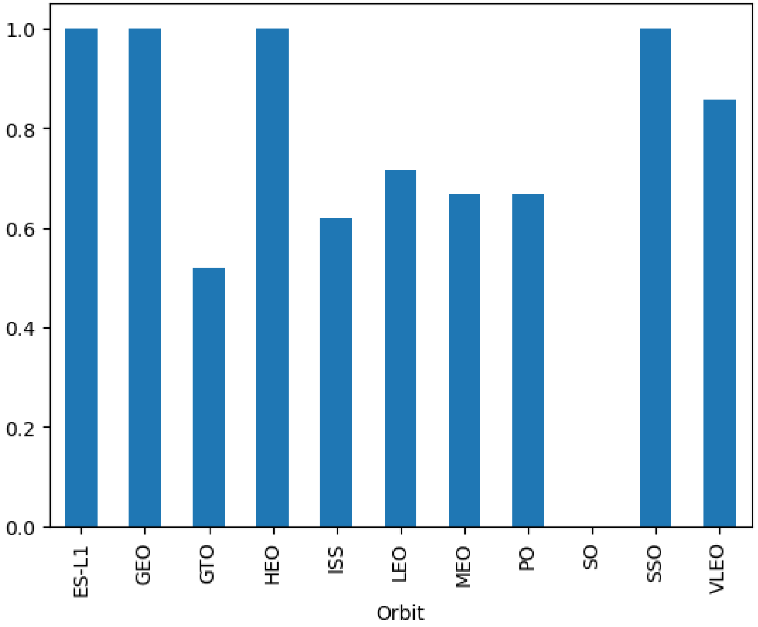
Create Landing Outcome Label

https://github.com/NativeLag/Capstone-Project-IMBD/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

To explore the data, scatterplots, a bar chart, and line graph were used to find the relationship between variables. As an example, the scatterplot below shows the relationship between Flight Number vs Launch Site. Likewise, the bar chart shows Orbit Type vs Success Rate.





https://github.com/NativeLag/Capstone-Project-IMBD/blob/main/jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb

EDA with SQL

Performed the following SQL queries:

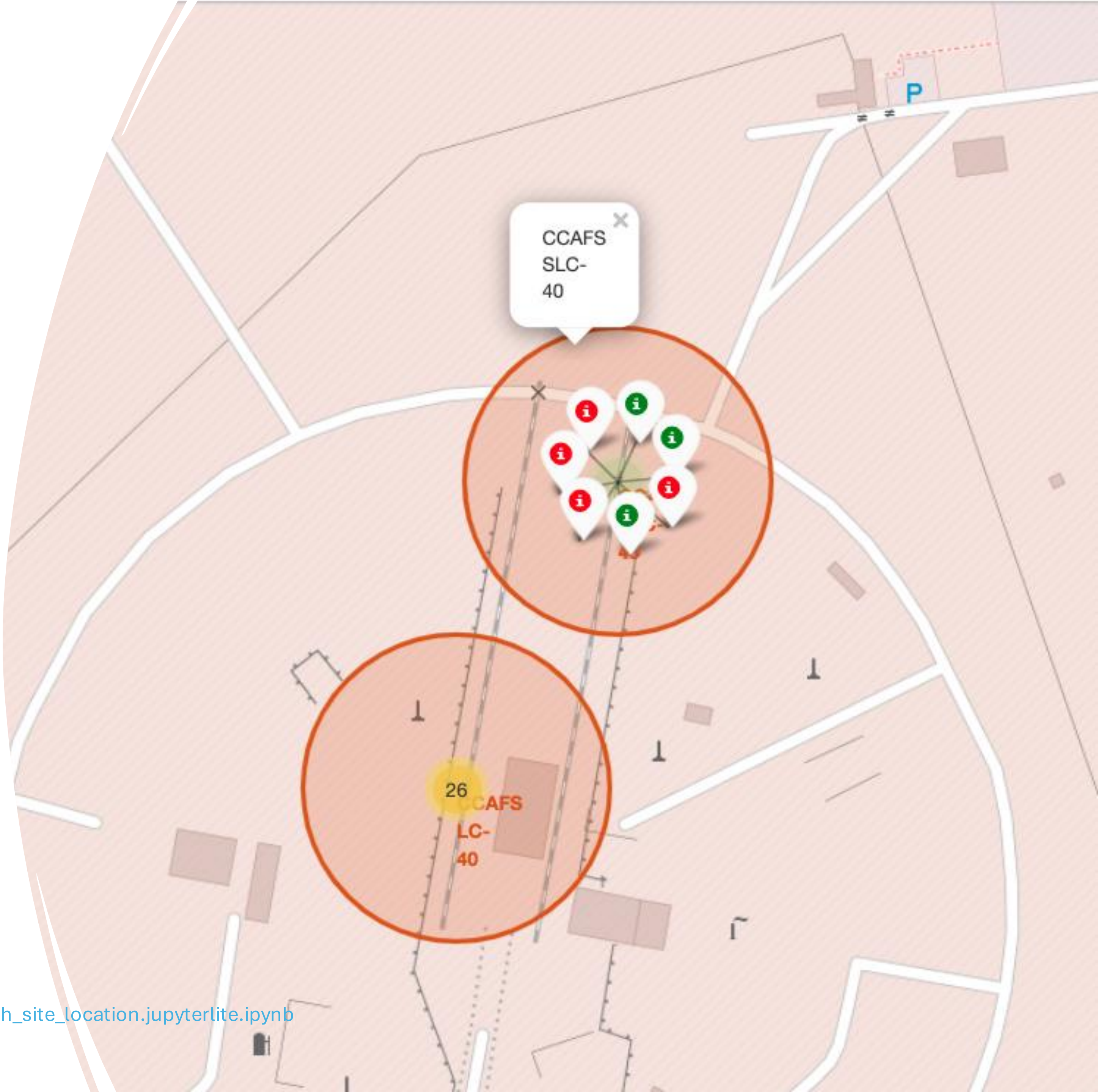
- Display names of the unique launch sites in the space mission
- Display 5 records where launch sites begin with the string 'CCA’
- Display total payload mass carried by boosters launched by NASA (CRS)
- Display average payload mass carried by booster version F9 v1.1
- Display date when the first successful landing outcome in ground pad was achieved
- Display names of boosters which have success in drone ship with payload mass greater than 4000 but less than 6000
- Display total number of successful and failure mission outcomes
- Display names of booster versions which have carried maximum payload mass
- Display failed landing outcomes in drone ship, booster versions, and launch site for the in year 2015
- 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

https://github.com/NativeLag/Capstone-Project-IMBD/blob/main/jupyter-labs-eda-sql-coursera-sqllite.ipynb

Build an Interactive Map with Follium

Launch Site Markers

- Markers indicate Launch Site
- Circles indicate areas around specific locations
- Marker Clusters indicate events at location such as launches
- Lines indicate distances between coordinates



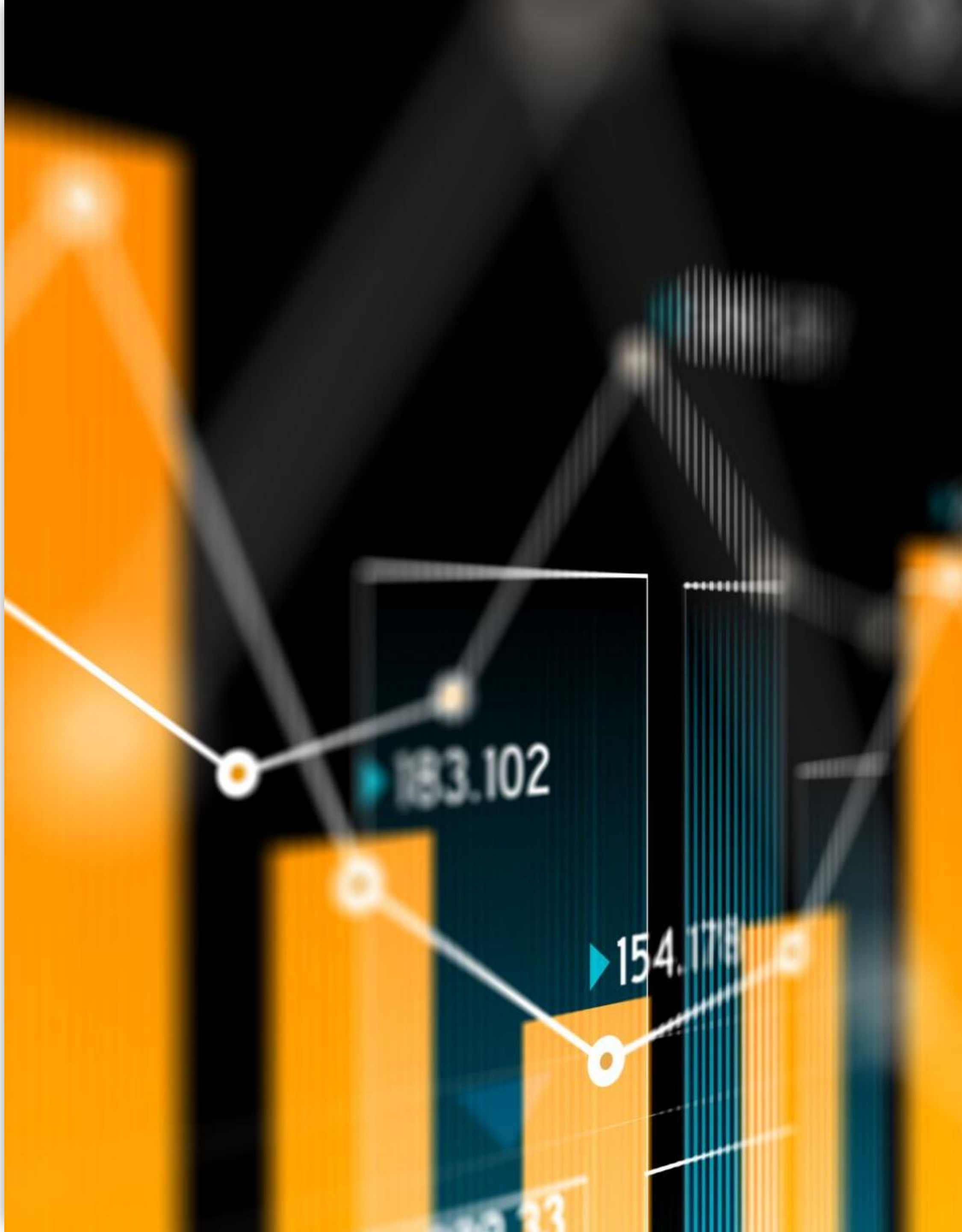
https://github.com/NativeLag/Capstone-Project-IMBD/blob/main/lab_jupyter_launch_site_location.jupyterlite.ipynb

Build a Dashboard with Plotly Dash

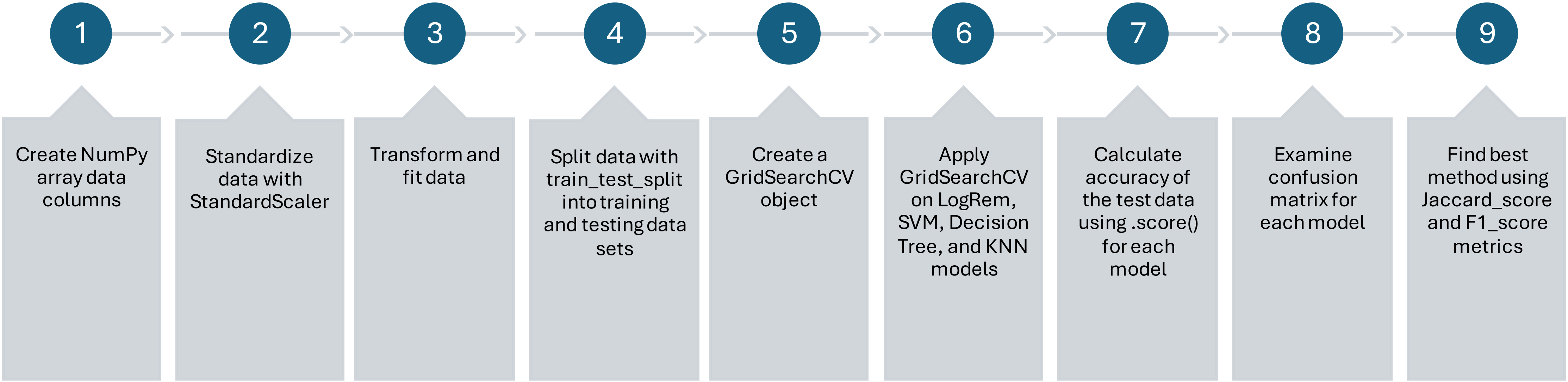
The Dashboard was created with the following:

- Launch Site dropdown selection
- Pie Chart visually representing Success vs Failed launches for each site
- Mass Payload Slider
- Scatter Chart Mass vs Success Rate for different booster types

https://github.com/NativeLag/Capstone-Project-IMBD/blob/main/spacex_dash_app.py



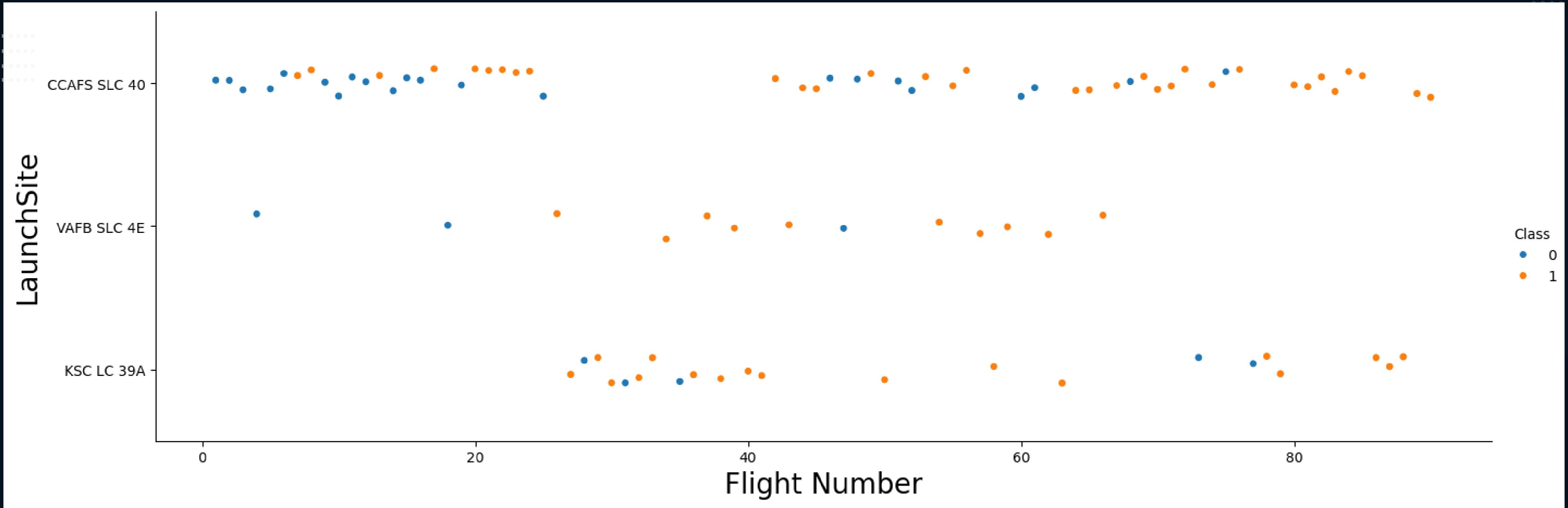
Predictive Analysis (Classification)



1. Create NumPy array data columns
2. Standardize data with StandardScaler
3. Transform and fit data
4. Split data with train_test_split into training and testing data sets
5. Create a GridSearchCV object
6. Apply GridSearchCV on LogRem, SVM, Decision Tree, and KNN models
7. Calculate accuracy of the test data using .score() for each model
8. Examine confusion matrix for each model
9. Find best method using Jaccard_score and F1_score metrics

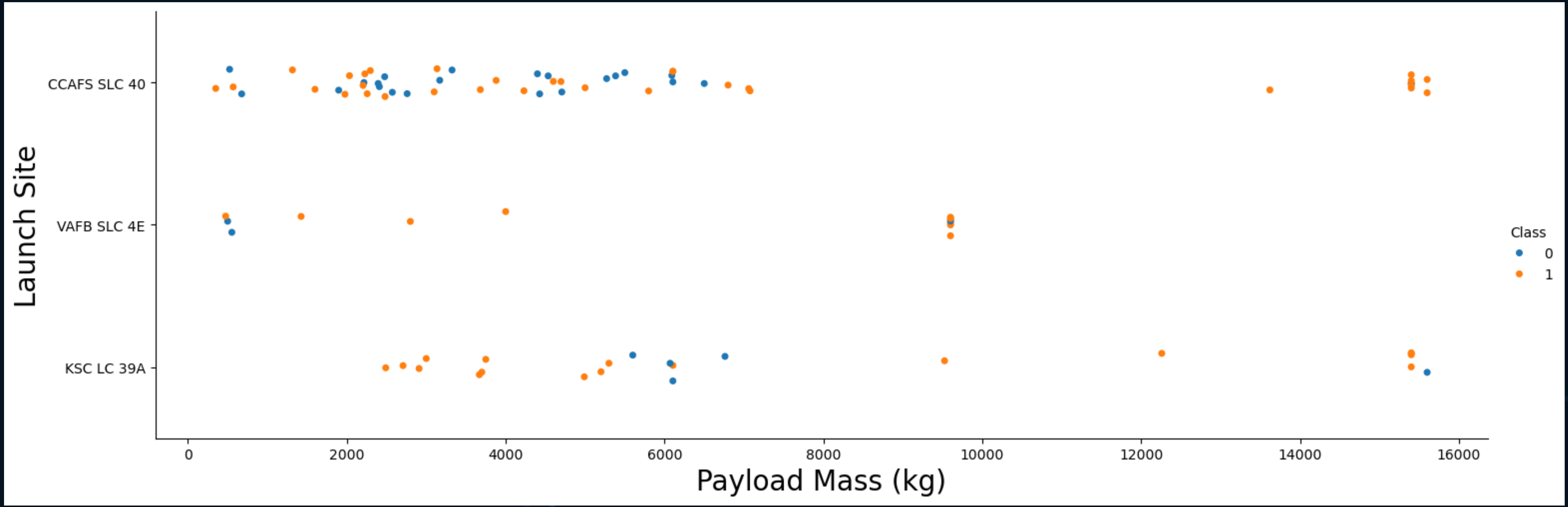
Results and Initial Insights

IBM Developer
SKILLS NETWORK



Flight Number vs Launch Site

- Early flights had a low success rate
- Improvement has been continuous
- CCAFS SLC 40 makes 50% of all launches
- VAFB SLC 4E and KSC LC 39A have high success rates

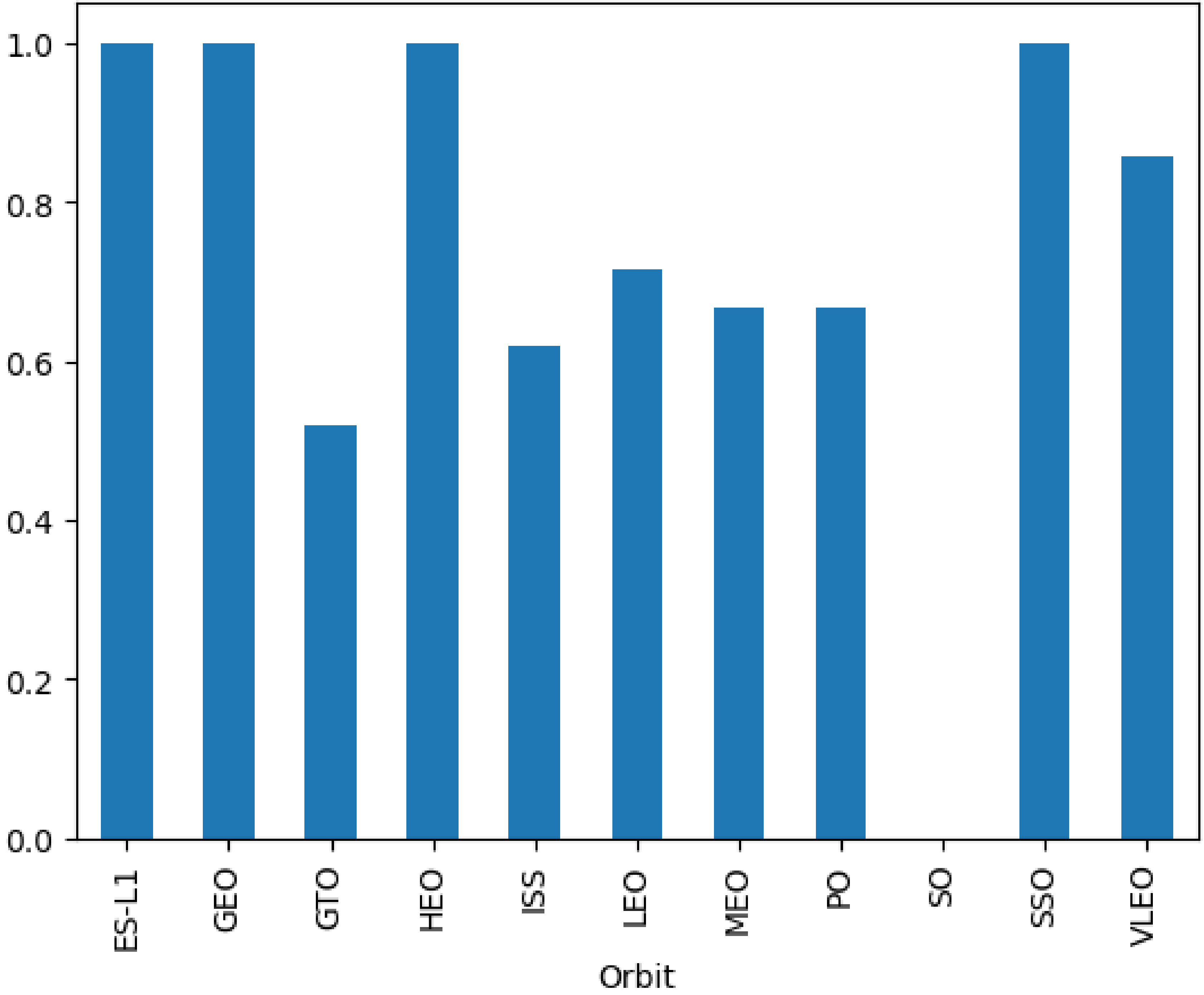


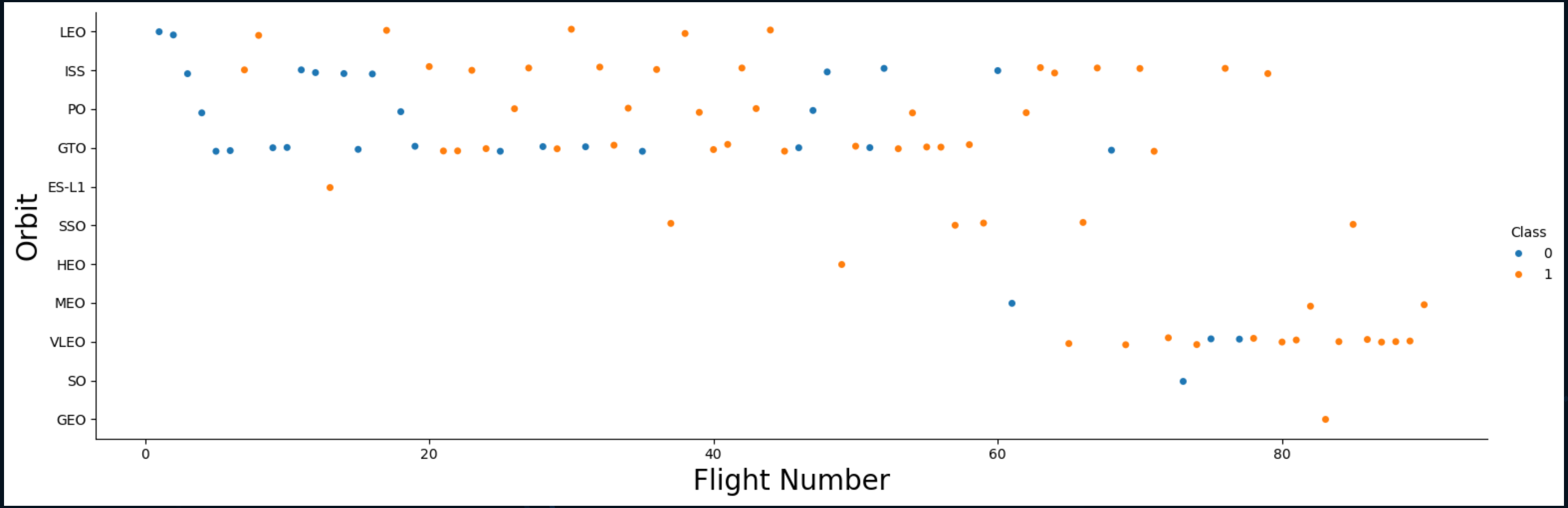
Payload vs Launch Site

- Payloads above 9000kg have a dramatically higher success rate
- Payloads above 12000kg seem to only be possible at KSC LC 39A and CCAFS SLC 40
- KSC LC 39A has the highest success rate for payloads under 6000kg
- CCAFS SLC 40 has the highest success rate for payloads above 13000kg

Success Rate vs Orbit

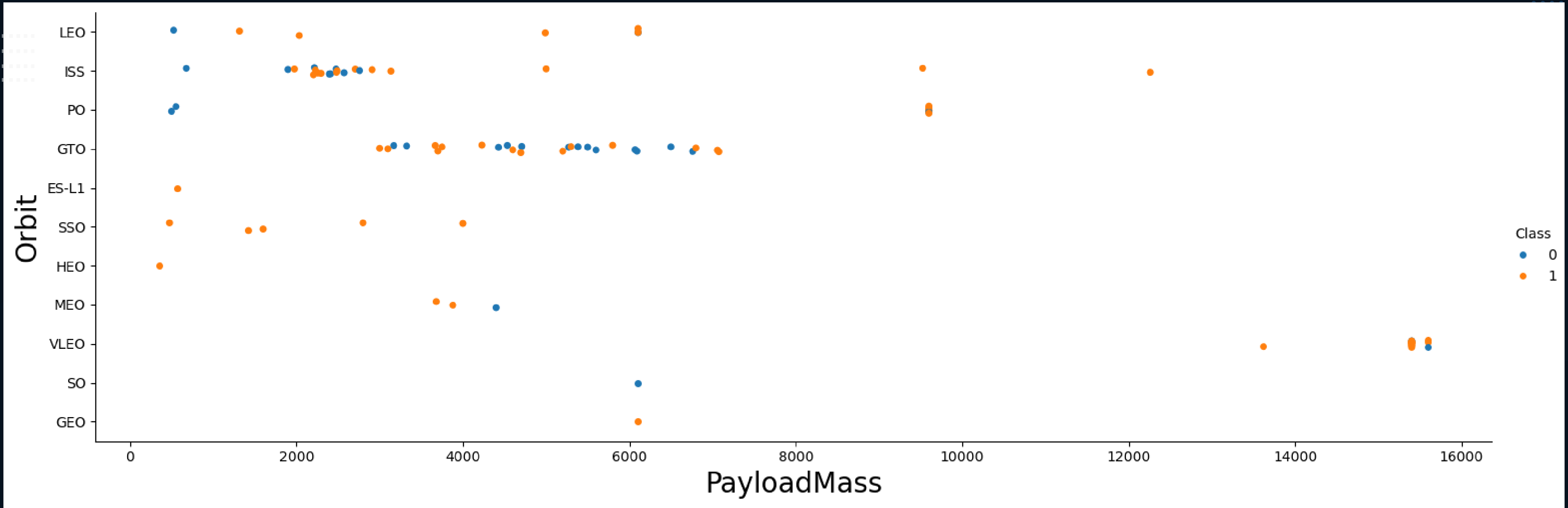
- ES-L1, GEO, HEO, and SSO have a 100% success rate
- VLEO has a success rate above 80%
- LEO has a success rate above 70%
- MEO, PO, ISS, and GTO have success rates 50% and above
- SO has a 0% success rate





Flight Number vs Orbit

- Success rates improved across the board over time
- LEO had high success but few launches before seemingly being discontinued
- GTO had most granular improvement over time
- ISS and GTO seem to be some of the most common
- VLEO appears to be an emerging market with high success rates

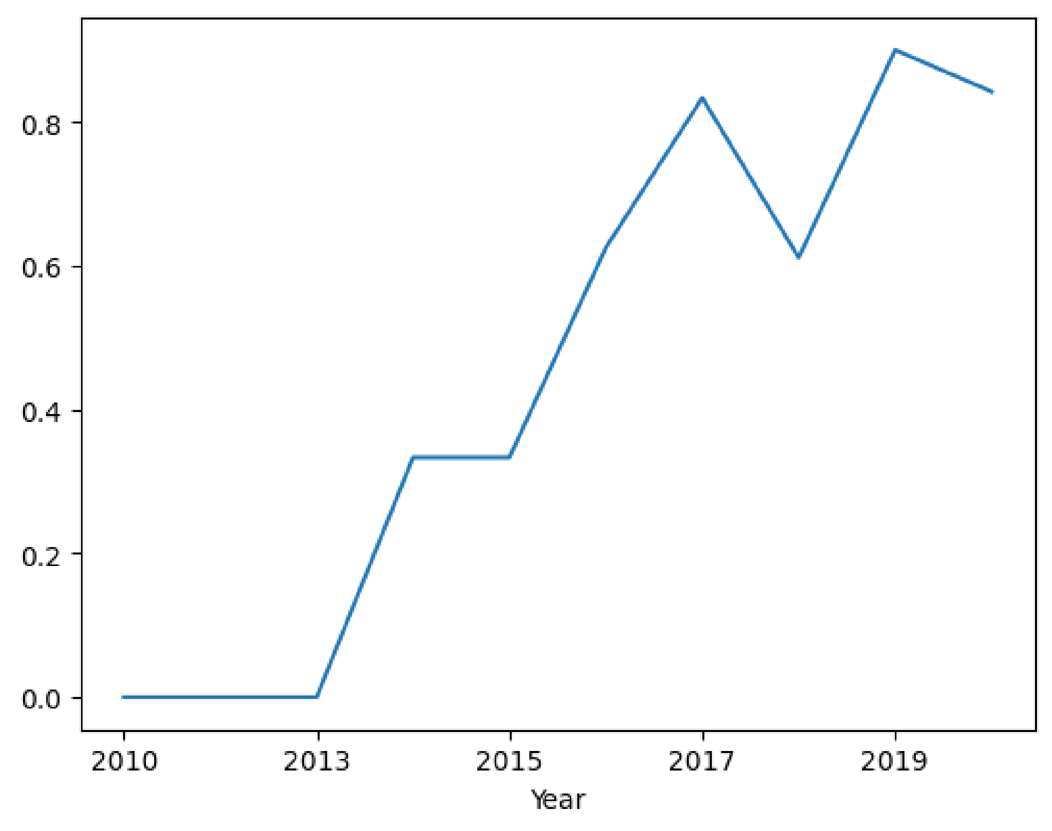


Payload Mass vs Orbit Type

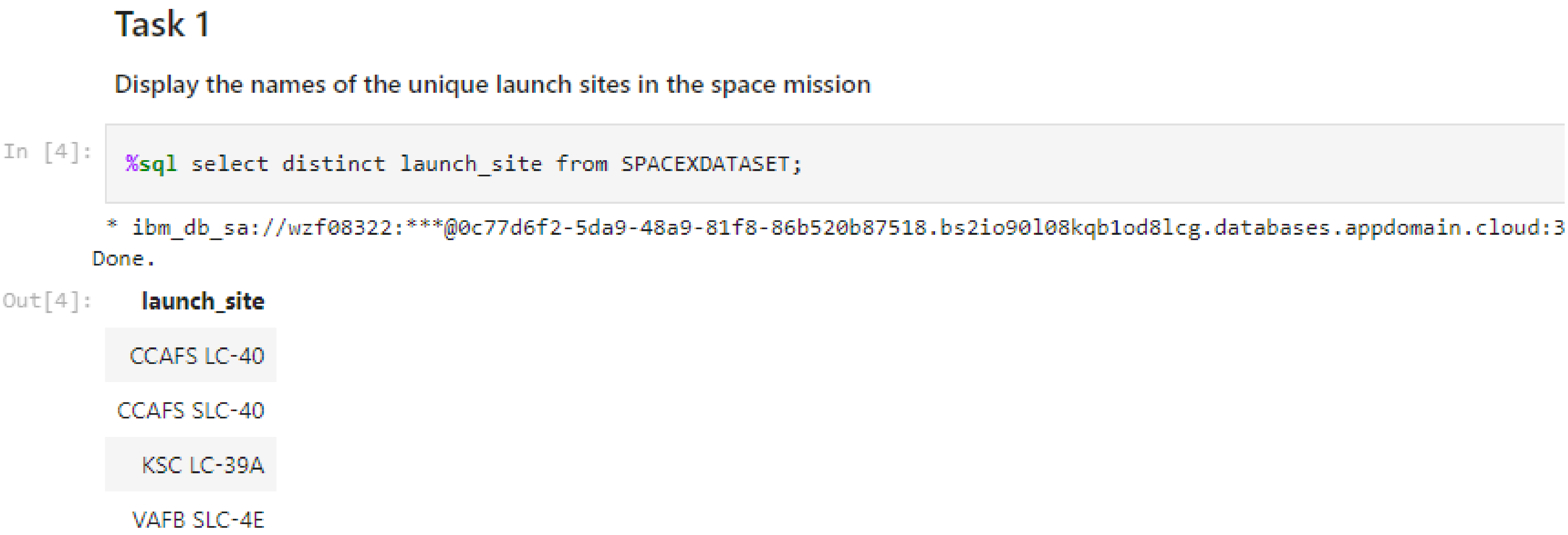
- LEO, ISS, and PO orbit types seem to handle sub 1000kg payloads poorly. Handling plus 4000kg best
- GTO show no correlation between payload mass and success rate between 3000kg and 7000kg
- SSO has 100% success rate with payloads below 4000kg
- VLEO only handles payloads with masses above 13000kg

Launch Success Yearly Trend

- Success rate has been on an upward trend since 2013
- The rate of increase seems to have slowed down between 2017 and 2020



Insights Drawn From EDA



Task 1

Display the names of the unique launch sites in the space mission

```
In [4]: %sql select distinct launch_site from SPACEXDATASET;
```

```
 * ibm_db_sa://wzf08322:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:3
Done.
```

Out[4]:

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

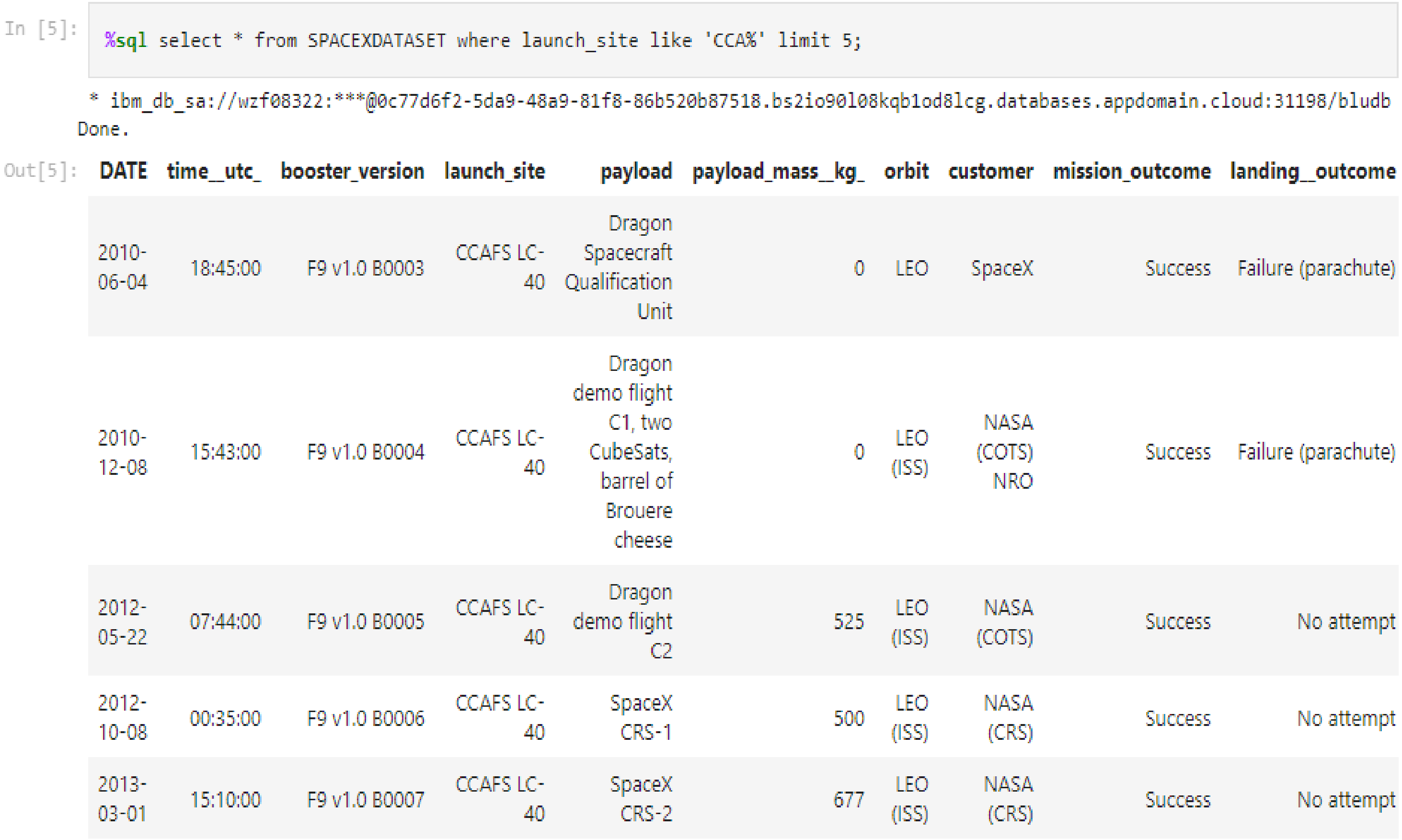
All Launch Site Names

Names of each launch site used by SpaceX

Launch Sites Beginning with ‘CCA’

Using an SQL query, we called up the first five launch sites starting with the ‘CCA’ designation

Here we can see the dates, times, booster version, payload, mass, orbit type, mission outcome, and landing outcome of all five launches



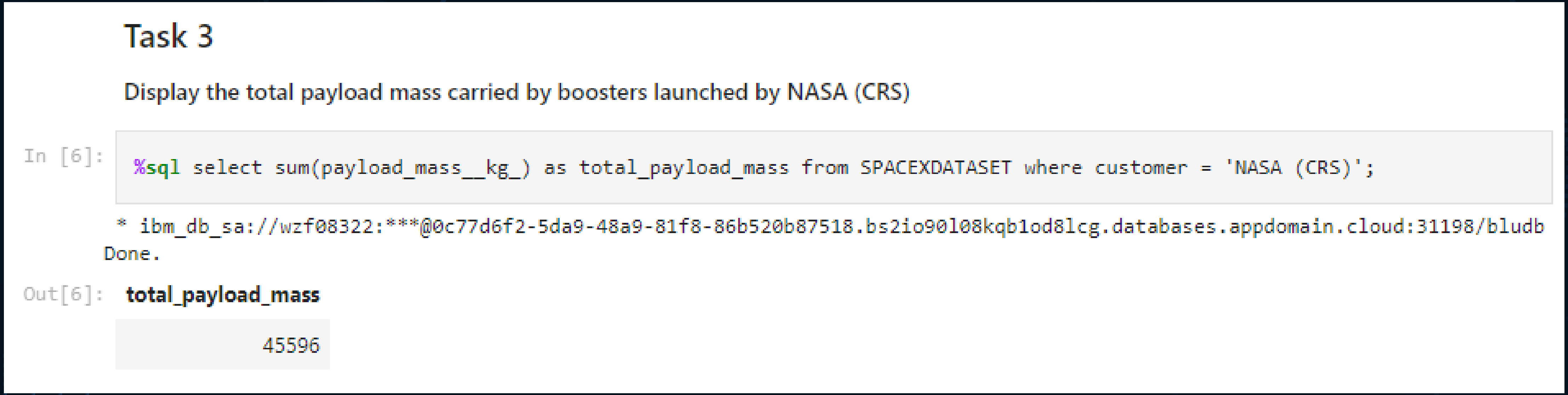
```
In [5]: %sql select * from SPACEXDATASET where launch_site like 'CCA%' limit 5;

 * ibm_db_sa://wzf08322:***@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

Total Payload Mass

Using the ‘sum’ command for total payload mass of all NASA launches we’re able to pull up a figure of 45596kg as illustrated below



Task 3

Display the total payload mass carried by boosters launched by NASA (CRS)

```
In [6]: %sql select sum(payload_mass__kg_) as total_payload_mass from SPACEXDATASET where customer = 'NASA (CRS)';
```

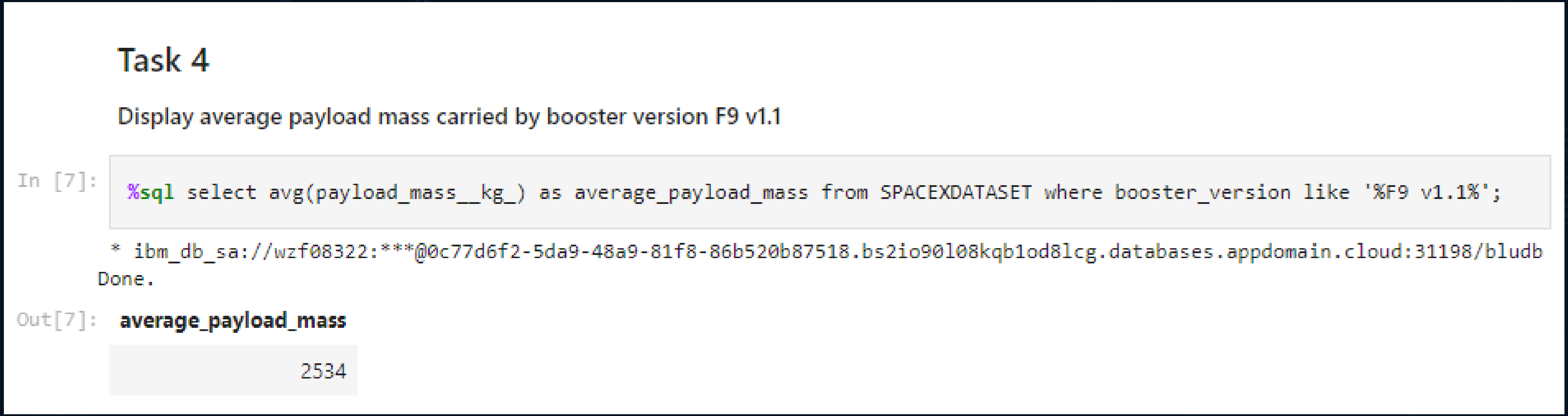
```
 * ibm_db_sa://wzf08322:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/bludb
Done.
```

Out[6]:

total_payload_mass
45596

Average Payload Mass by F9 v1.1

Using ‘avg’ command in SQL we’re able to pull up the average payload mass of Falcon 9 version 1.1 rocket boosters. Across all F9 v1.1 launches the average payload mass comes out to 2534 as shown in the below query



Task 4

Display average payload mass carried by booster version F9 v1.1

```
In [7]: %sql select avg(payload_mass__kg_) as average_payload_mass from SPACEXDATASET where booster_version like '%F9 v1.1%';
```

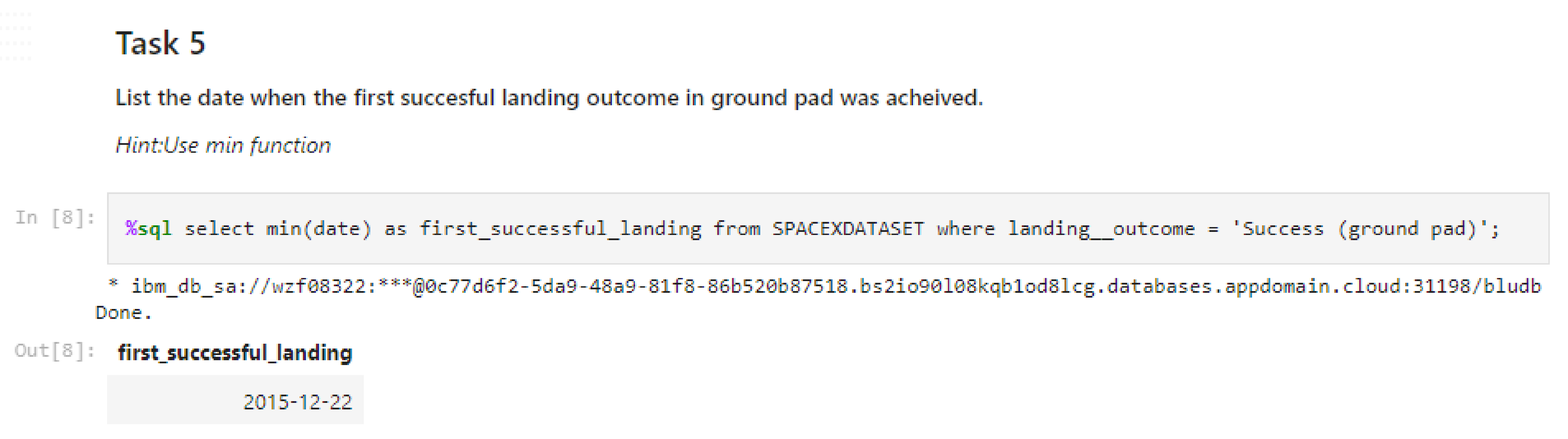
```
 * ibm_db_sa://wzf08322:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/bludb
Done.
```

Out[7]:

average_payload_mass
2534

First Successful Ground Landing Date

Using the ‘min’ command on landing outcomes that were successful we can pull up the exact date of when SpaceX managed a successful ground landing. 22nd of December 2015



Task 5

List the date when the first succesful landing outcome in ground pad was acheived.

Hint:Use min function

In [8]:

```
%sql select min(date) as first_successful_landing from SPACEXDATASET where landing__outcome = 'Success (ground pad)';
```

```
 * ibm_db_sa://wzf08322:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/bludb
Done.
```

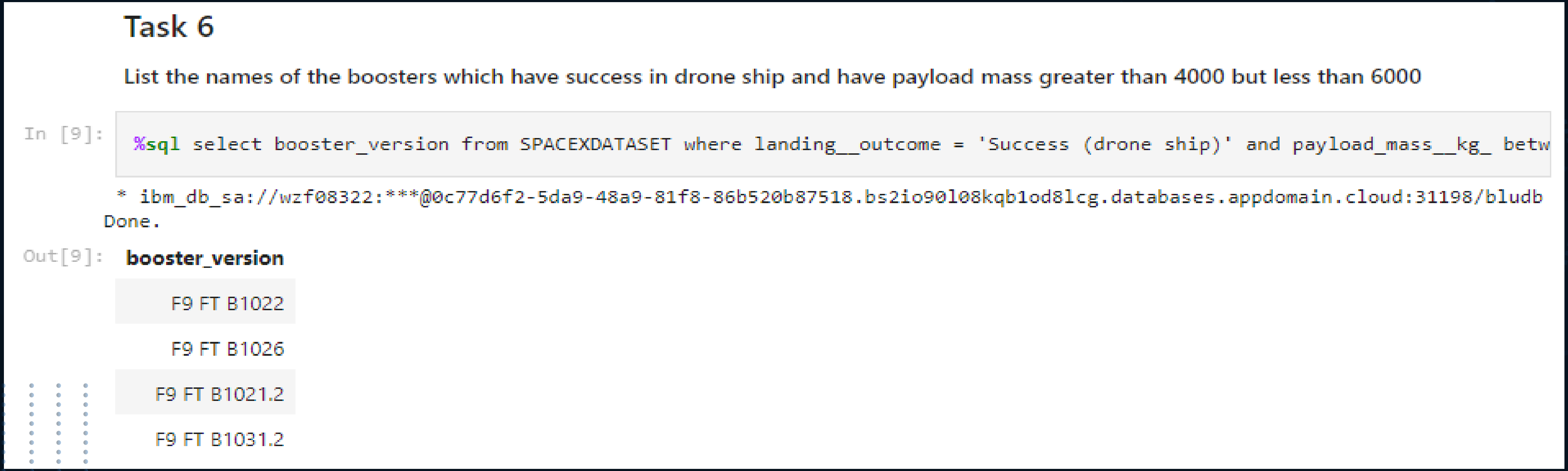
Out[8]:

first_successful_landing
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

Using SQL we pulled up a list of boosters that have successfully landed on drone ships carrying a payload mass between 4000 to 6000kg

Shown below we can see the B1022, B1026, B1021.2, and B1031.2 have all managed successful drone ship landings



Task 6

List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

```
In [9]:  %sql select booster_version from SPACEXDATASET where landing__outcome = 'Success (drone ship)' and payload_mass__kg_ betw
```

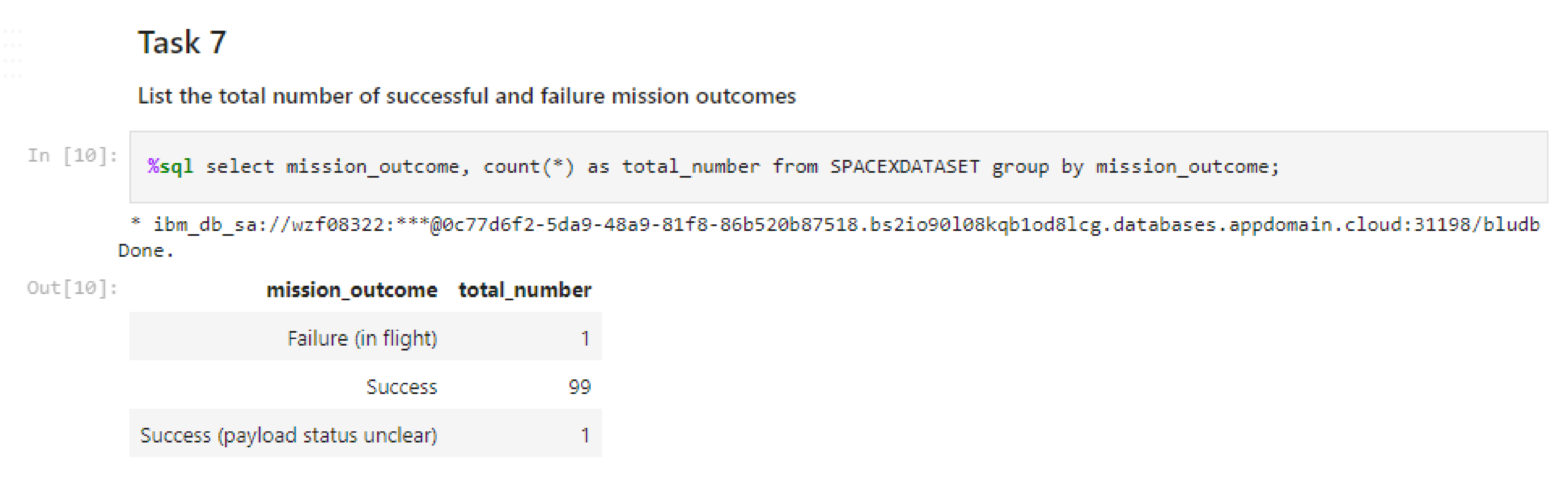
```
 * ibm_db_sa://wzf08322:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/bludb
Done.
```

Out[9]:

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

Total Number of Successful and Failure Mission Outcomes

Selecting mission outcomes and having SQL count the total number then group them by said outcomes we can see clearly there has been 1 Failure, 99 Successes, and 1 success with unclear payload status.



Task 7

List the total number of successful and failure mission outcomes

In [10]:

```
%sql select mission_outcome, count(*) as total_number from SPACEXDATASET group by mission_outcome;
```

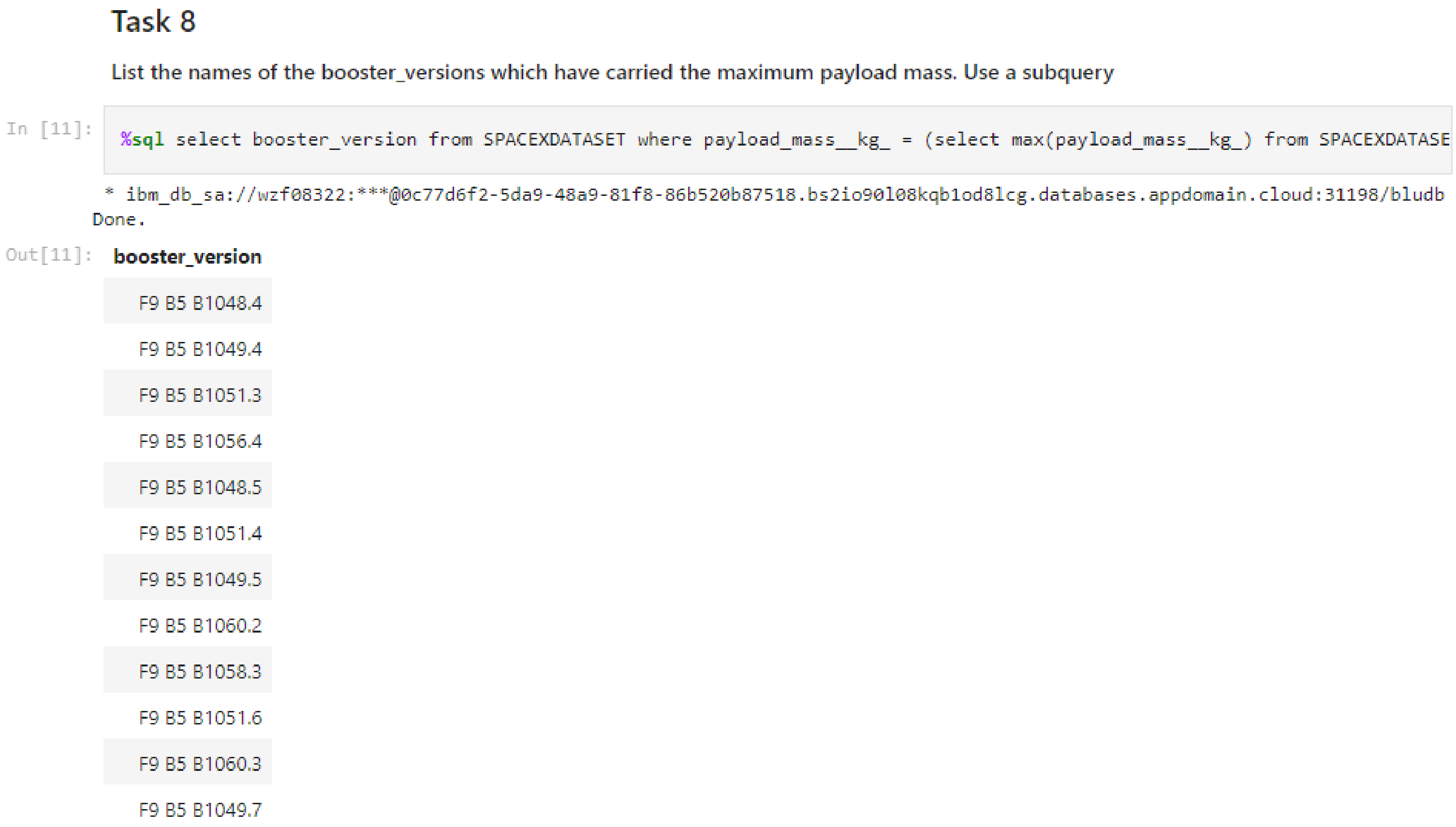
```
 * ibm_db_sa://wzf08322:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/bludb
Done.
```

Out[10]:

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

Booster Carried Maximum Payload

Selecting booster version, we launch a sub-query where payload mass has been max. Listed are all booster versions that have carried the maximum payload mass



Task 8

List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

```
In [11]: %sql select booster_version from SPACEXDATASET where payload_mass__kg_ = (select max(payload_mass__kg_) from SPACEXDATASE
```

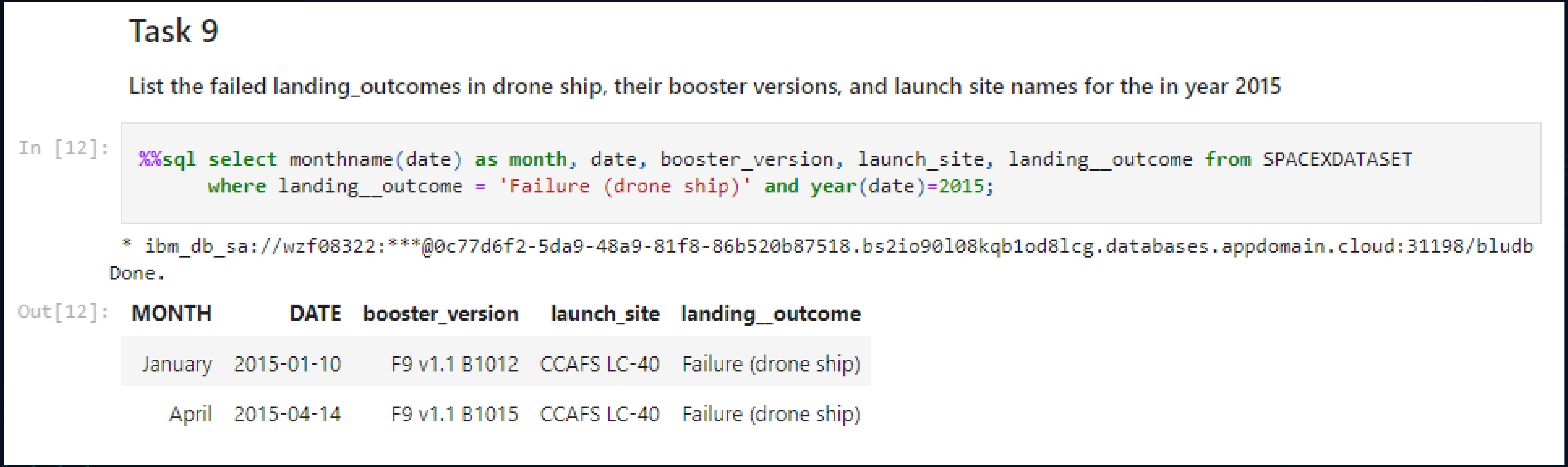
```
 * ibm_db_sa://wzf08322:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/bludb
Done.
```

Out[11]:

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

2015 Launch Records

Selecting relevant information to be displayed we set ‘WHERE’ as “landing_outcome = ‘Failure (drone ship)’” and use an ‘AND’ command to specifically pull for the year 2015



Task 9

List the failed landing_outcomes in drone ship, their booster versions, and launch site names for the in year 2015

In [12]:

```
%%sql select monthname(date) as month, date, booster_version, launch_site, landing__outcome from SPACEXDATASET
      where landing__outcome = 'Failure (drone ship)' and year(date)=2015;
```

```
 * ibm_db_sa://wzf08322:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/bludb
Done.
```

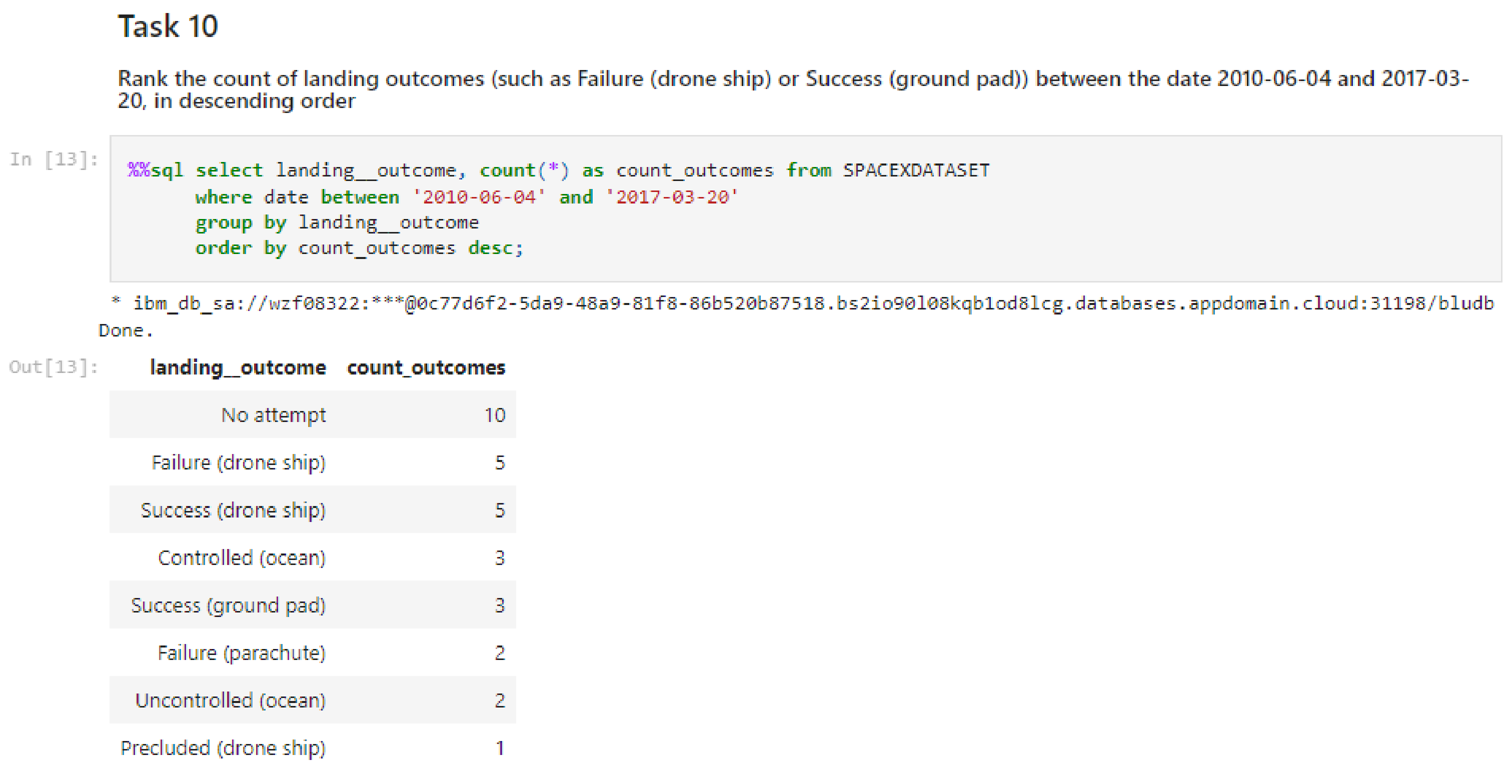
Out[12]:

MONTH	DATE	booster_version	launch_site	landing__outcome
January	2015-01-10	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
April	2015-04-14	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

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

Selecting for landing outcome we grouped them by type of outcome, counted, and ranked them accordingly.

As shown, there were 10 ‘No Attempt’s, 5 drone ship failures, 5 drone ship success, 3 controlled ocean landings, 3 successful ground pad landings, 2 failed parachute landings, 2 uncontrolled ocean landings, and 1 precluded drone ship landing.



Task 10

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

In [13]:

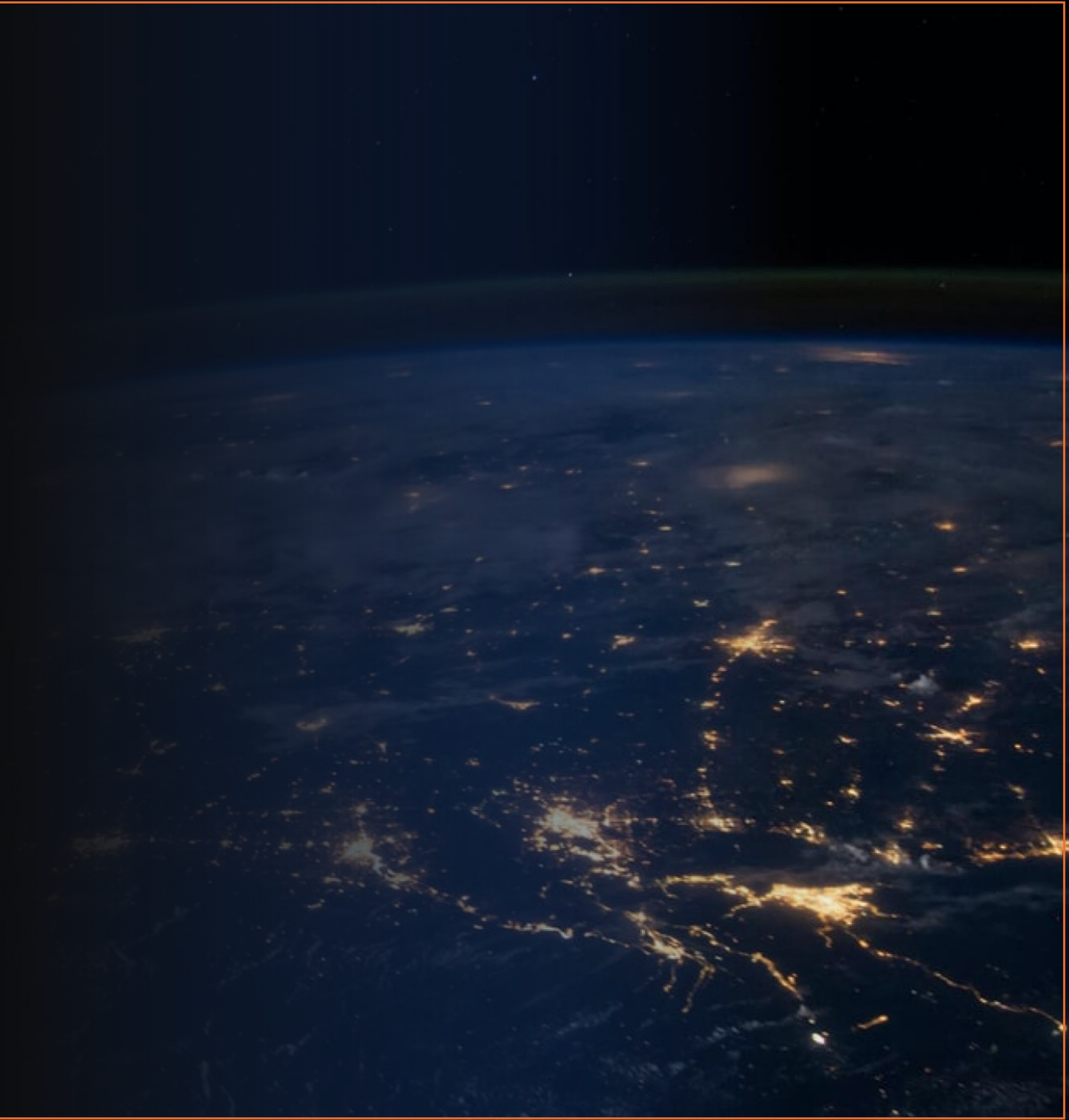
```
%%sql select landing__outcome, count(*) as count_outcomes from SPACEXDATASET
    where date between '2010-06-04' and '2017-03-20'
    group by landing__outcome
    order by count_outcomes desc;
```

```
 * ibm_db_sa://wzf08322:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/bludb
Done.
```

Out[13]:

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

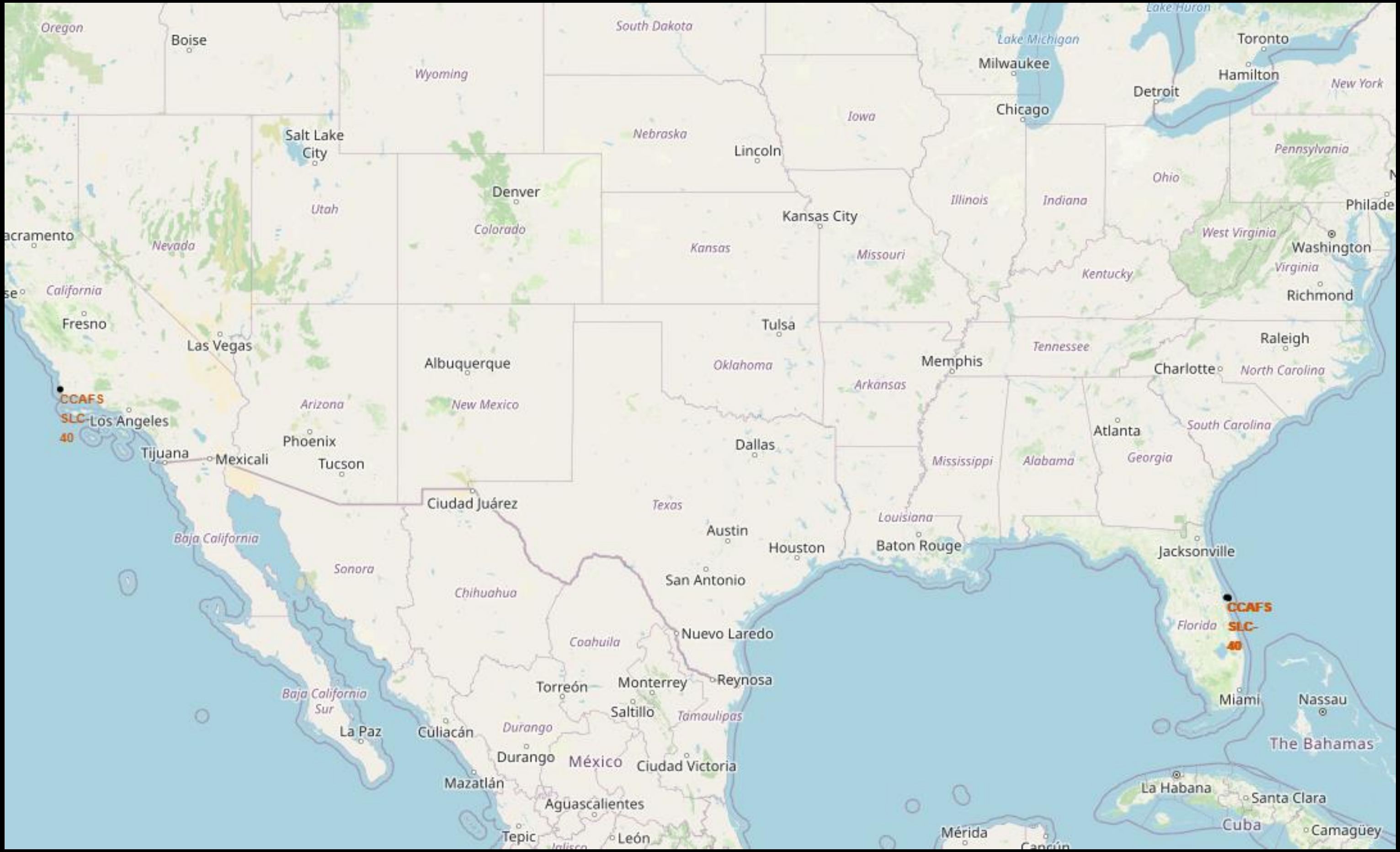
Launch Site Proximities Analysis



Launch Site Locations Map

Through Folium an interactive map was created listing all launch locations.

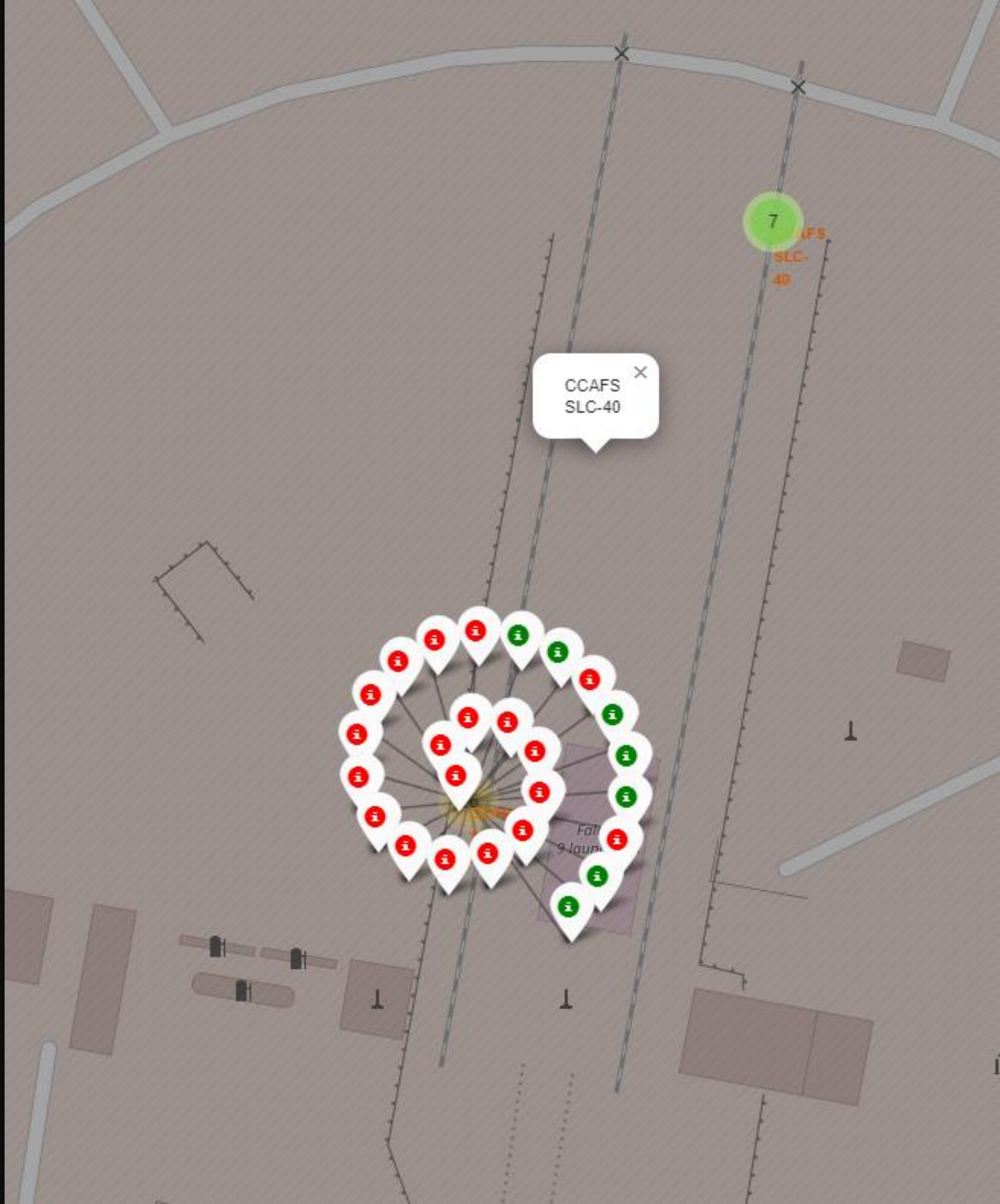
CCAFS SLC-40 in California and the remaining three locations in Florida.



Launch Outcomes Map

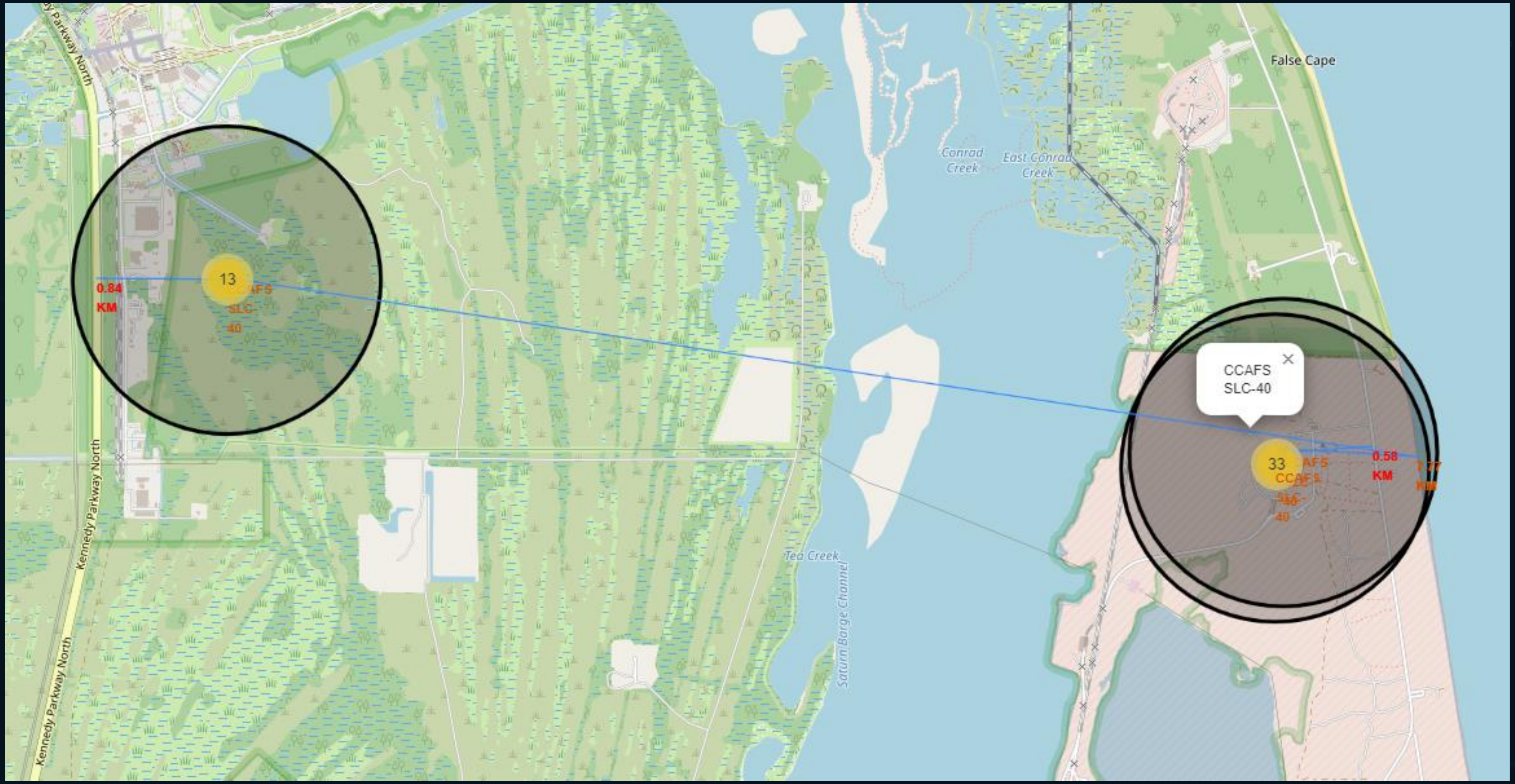
Zooming into our map we can see a comprehensive overview of SpaceX's launch outcomes from each launch location.

As an example, displayed are 33 launch outcomes with 27 on display from the CCAFS SLC-40 Launch Site.



Launch Site Proximity and Distances

Calculating distances to major locations such as highway, railway, and coastline we can see the need for a balance between easy access to transportation infostructure while being located away from major populated areas.

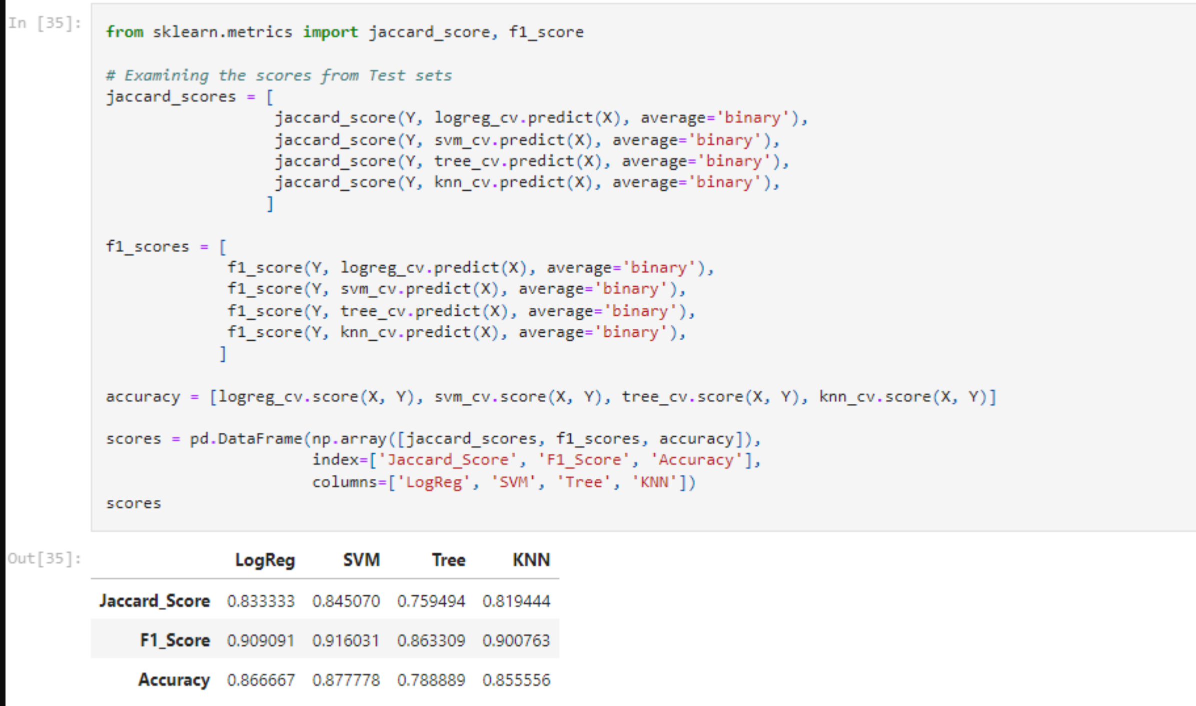


Predictive Analysis (Classification)

Classification Accuracy

Examining the Machine Learning Predictions we’re able to see that of the four models the Decision Tree performed the poorest while the other three averaged out to be nearly identical.

Looking at the raw numbers though we can see that SVM edges out KNN and Logistic Regression.

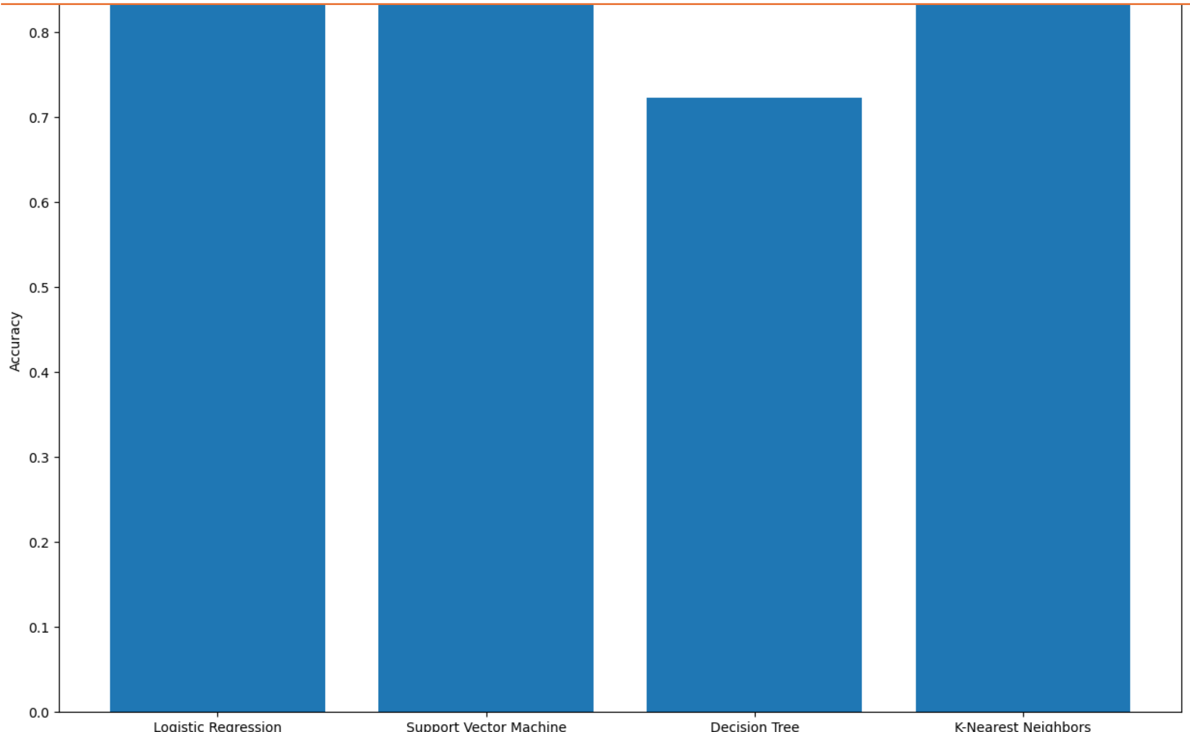


In [35]:

```
from sklearn.metrics import jaccard_score, f1_score

# Examining the scores from Test sets
jaccard_scores = [
                    jaccard_score(Y, logreg_cv.predict(X), average='binary'),
                    jaccard_score(Y, svm_cv.predict(X), average='binary'),
                    jaccard_score(Y, tree_cv.predict(X), average='binary'),
                    jaccard_score(Y, knn_cv.predict(X), average='binary'),
                   ]

f1_scores = [
               f1_score(Y, logreg_cv.predict(X), average='binary'),
               f1_score(Y, svm_cv.predict(X), average='binary'),
               f1_score(Y, tree_cv.predict(X), average='binary'),
               f1_score(Y, knn_cv.predict(X), average='binary'),
              ]

accuracy = [logreg_cv.score(X, Y), svm_cv.score(X, Y), tree_cv.score(X, Y), knn_cv.score(X, Y)]

scores = pd.DataFrame(np.array([jaccard_scores, f1_scores, accuracy]),
                         index=['Jaccard_Score', 'F1_Score', 'Accuracy'],
                         columns=['LogReg', 'SVM', 'Tree', 'KNN'])
scores
```

Out[35]:

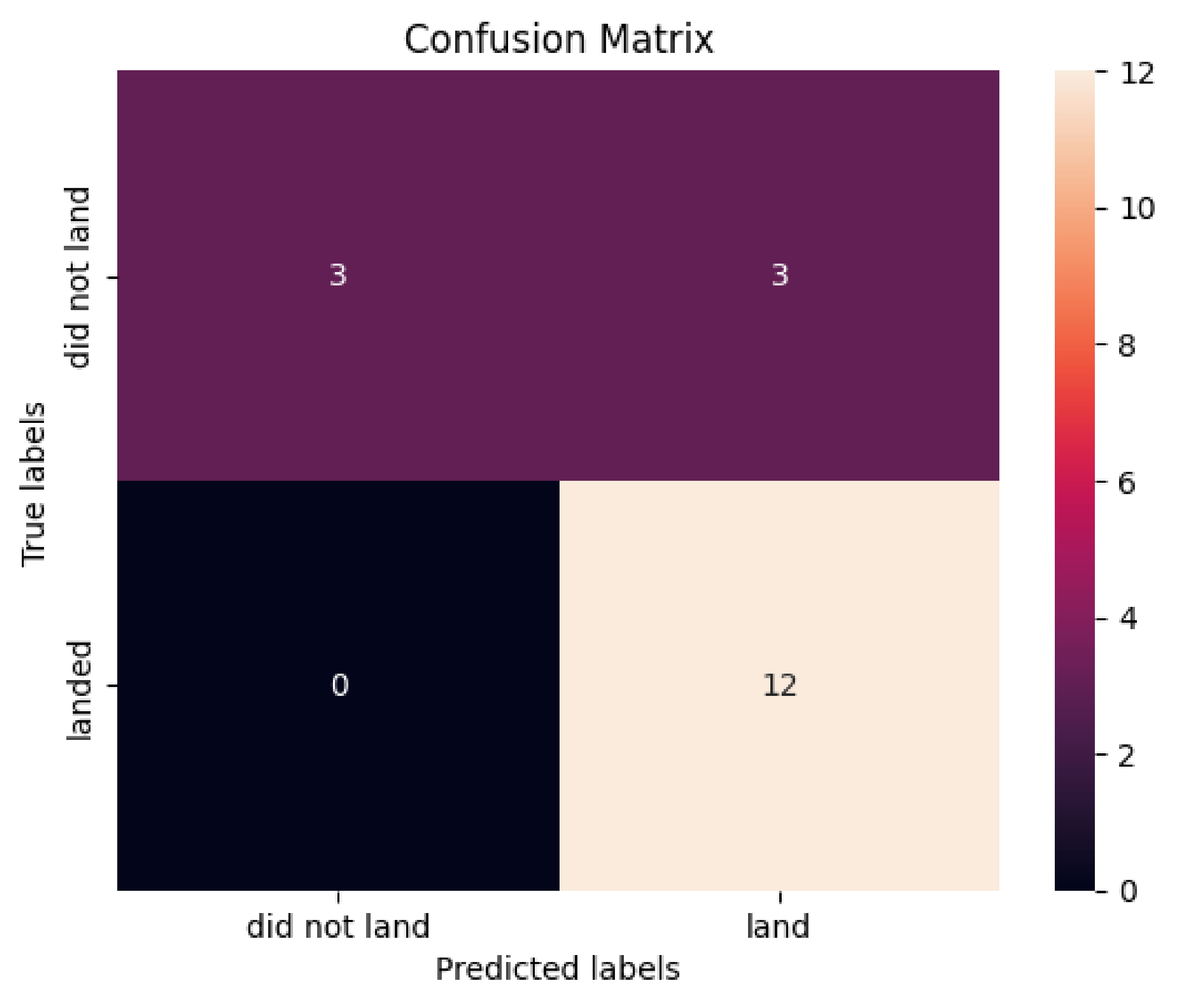
	LogReg	SVM	Tree	KNN
Jaccard_Score	0.833333	0.845070	0.759494	0.819444
F1_Score	0.909091	0.916031	0.863309	0.900763
Accuracy	0.866667	0.877778	0.788889	0.855556



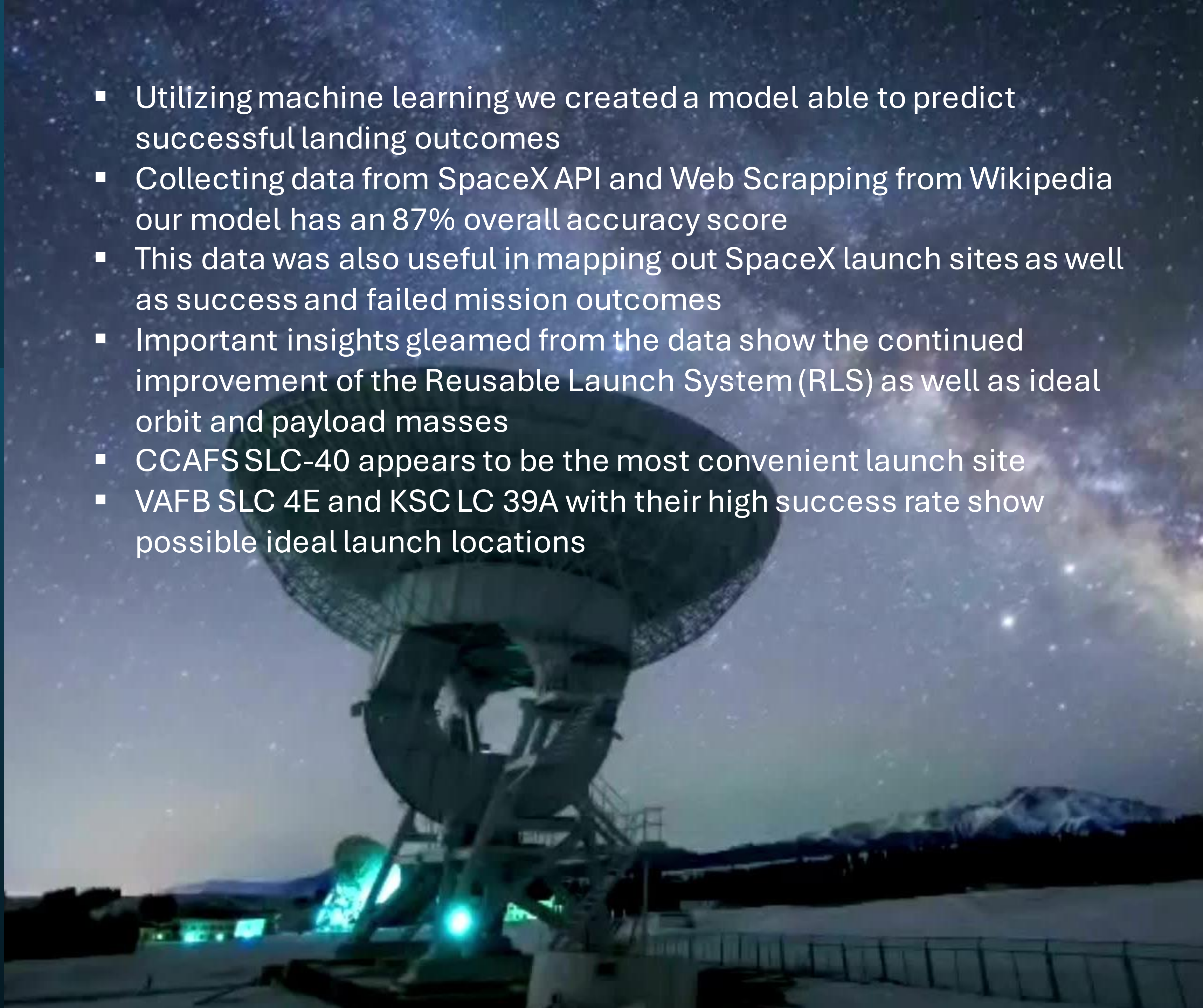
SVM Confusion Matrix

Using our most accurate model we can see that how well it was able to predict landed or ‘True Positive’ outcomes and 0 False Negatives.

It does appear to have a small room for error with False Positive.



Conclusion



- Utilizing machine learning we created a model able to predict successful landing outcomes
- Collecting data from SpaceX API and Web Scrapping from Wikipedia our model has an 87% overall accuracy score
- This data was also useful in mapping out SpaceX launch sites as well as success and failed mission outcomes
- Important insights gleamed from the data show the continued improvement of the Reusable Launch System (RLS) as well as ideal orbit and payload masses
- CCAFS SLC-40 appears to be the most convenient launch site
- VAFB SLC 4E and KSC LC 39A with their high success rate show possible ideal launch locations

Appendix

Sources:

SpaceX API

Wikipedia

GitHub URL:
https://github.com/NativeLag/Capstone-Project-IMBD

Special Thanks to All Instructors:
https://www.coursera.org/professional-certificates/ibm-data-science