IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

<Shamila Habib>
<12/21/2023>

Outline

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

Executive Summary

- In this capstone project, we will predict if the SpaceX Falcon 9 first stage will land successfully using several machine learning classification algorithms. The main steps in this project include:
- Data collection, wrangling, and formatting
- Exploratory data analysis
- Interactive data visualization
- Machine learning prediction
- Our graphs show that some features of the rocket launches have a correlation with the outcome of the launches, i.e., success or failure. It is also concluded that decision tree may be the best machine learning algorithm to predict if the Falcon 9 first stage will land successfully.

Introduction

- In this capstone, we will predict if the Falcon 9 first stage will land successfully. SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage. Therefore if we can determine if the first stage will land, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch.
- Most unsuccessful landings are planned. Sometimes, SpaceX will perform a controlled landing in the ocean. The main question that we are trying to answer is, for a given set of features about a Falcon 9 rocket launch which include its payload mass, orbit type, launch site, and so on, will the first stage of the rocket land successfully?

Section 1

Methodology

Methodology

- The overall methodology includes:

1. Data collection, wrangling, and formatting, using:

- SpaceX API
- Web scraping

2. Exploratory data analysis (EDA), using:

- Pandas and NumPy
- SQL

3. Data visualization, using:

- Matplotlib and Seaborn
- Folium
- Dash

4. Machine learning prediction, using

- Logistic regression
- Support vector machine (SVM)
- Decision tree
- K-nearest neighbors (KNN)

Data Collection

- Describe how data sets were collected.
- You need to present your data collection process use key phrases and flowcharts

Data Collection – SpaceX API

- 1_Data Collection API.ipynb
- Libraries or modules used: requests, pandas, numpy, datetime
- The API used is here.
- The API provides data about many types of rocket launches done by SpaceX, the data is therefore filtered to include only Falcon 9 launches.
- The API is accessed using requests.get().
- The json result is converted to a dataframe using the json_normalize() function from pandas.
- Every missing value in the data is replaced the mean the column that the missing value belongs to.
- We end up with 90 rows or instances and 17 columns or features.

Data Collection - Scraping

- 2_ Data Collection with Web Scraping.ipynb
- Libraries or modules used: sys, requests, BeautifulSoup from bs4, re, unicodedata, pandas
- The data is scraped from List of Falcon 9 and Falcon Heavy launches.
- The website contains only the data about Falcon 9 launches.
- First, the Falcon9 Launch Wiki page is requested from the url and a BeautifulSoup object is created from response of requests.get().
- Next, all column/variable names are extracted from the HTML table header by using the find_all() function from BeautifulSoup.
- A dataframe is then created with the extracted column names and entries filled with launch records extracted from table rows.
- We end up with 121 rows or instances and 11 columns or features.

zero if the corresponding row in Outcome is in the set bad_outcome; otherwise, it's one

Data Wrangling

2_Data wrangling.ipynb

- Libraries or modules used: pandas, numpy
- SpaceX dataset was loaded from here.
- First, we identified and calculated the percentage of the missing values in each attribute.
- We calculated the number of launches on each site, number and occurrence of each orbit, number and occurrence of mission outcome of the orbits.
- Then we created a landing outcome label from Outcome Column. Using the Outcome, we created a list where the element is zero if the corresponding row in Outcome is in the set bad_outcome; otherwise, it's one.
- Lastly we calculated the success rate.

EDA with Data Visualization

- 3_EDA.ipynb
- Exploratory Data Analysis performed on variables Flight Number, Payload Mass, Launch Site, Orbit, Class and Year.

Plots Used:

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

- 4_EDA with SQL.ipynb
- Framework used: IBM DB2
- Libraries or modules used: ibm_db
- The data is queried using SQL to answer several questions about the data such as:
- The names of the unique launch sites in the space mission
- The total payload mass carried by boosters launched by NASA (CRS)
- The average payload mass carried by booster version F9 v1.1
- The SQL statements or functions used include SELECT, DISTINCT, AS, FROM, WHERE, LIMIT, LIKE, SUM(), AVG(), MIN(), BETWEEN, COUNT(), and YEAR().

Build an Interactive Map with Folium

- 6_Interactive Visual Analytics with Folium lab.ipynb
- Libraries or modules used: folium, wget, pandas, math
- Functions from the Folium libraries are used to visualize the data through interactive maps. The Folium library is used to:
- Mark all launch sites on a map
- Mark the succeeded launches and failed launches for each site on the map
- Mark the distances between a launch site to its proximities such as the nearest city, railway, or highway
- These are done using functions from folium such as add_child() and folium plugins which include MarkerCluster, MousePosition, and DivIcon.
-

  Example: A folium map showing the succeeded launches and failed launches for a specific launch site. If we zoom in on one of the launch site, we can see green and red tags. Each green tag represents a successful launch while each red tag represents a failed launch.

Build a Dashboard with Plotly Dash

- 7_spacex_dash_app.py
- Libraries or modules used: pandas, dash, dash_html_components, dash_core_components, Input and Output from dash.dependencies, plotly.express
- Functions from Dash are used to generate an interactive site where we can toggle the input using a dropdown menu and a range slider. Using a pie chart and a scatterplot, the interactive site shows:
- The total success launches from each launch site
- The correlation between payload mass and mission outcome (success or failure) for each launch site
- The application is launched on a terminal on the IBM Skills Network website.
-

  The picture below shows a pie chart when launch site CCAFS LC-40 is chosen in the dropdown menu on the website. 0 represents failed launches while 1 represents successful launches. We can see that 73.1% of launches done at CCAFS LC-40 are failed launches.

Predictive Analysis (Classification)

- 8_Machine Learning Prediction.ipynb
- Libraries or modules used: pandas, numpy, matplotlib.pyplot, seaborn, sklearn
- Functions from the Scikit-learn library are used to create our machine learning models. The machine learning prediction phase include the following steps:

1. Standardizing the data using the preprocessing.StandardScaler() function from sklearn
2. Splitting the data into training and test data using the train_test_split function from sklearn.model_selection
3. Creating machine learning models, which include:

- Logistic regression using LogisticRegression from sklearn.linear_model
- Support vector machine (SVM) using SVC from sklearn.svm
- Decision tree using DecisionTreeClassifier from sklearn.tree
- K nearest neighbors (KNN) using KNeighborsClassifier from sklearn.neighbors

4. Fit the models on the training set
5. Find the best combination of hyperparameters for each model using GridSearchCV from sklearn.model_selection
6. Evaluate the models based on their accuracy scores and confusion matrix using the score() function and confusion_matrix from sklearn.metrics

- Putting the results of all 4 models side by side, we can see that they all share the same accuracy score and confusion matrix when tested on the test set. Therefore, their GridSearchCV best scores are used to rank them instead. Based on the GridSearchCV best scores, the models are ranked in the following order with the first being the best and the last one being the worst:
- Decision tree (GridSearchCV best score: 0.8892857142857142)
- K nearest neighbors, KNN (GridSearchCV best score: 0.8482142857142858)
- Support vector machine, SVM (GridSearchCV best score: 0.8482142857142856)
- Logistic regression (GridSearchCV best score: 0.8464285714285713)
- The picture below shows the confusion matrix when the Decision Tree model is tested on the test data.

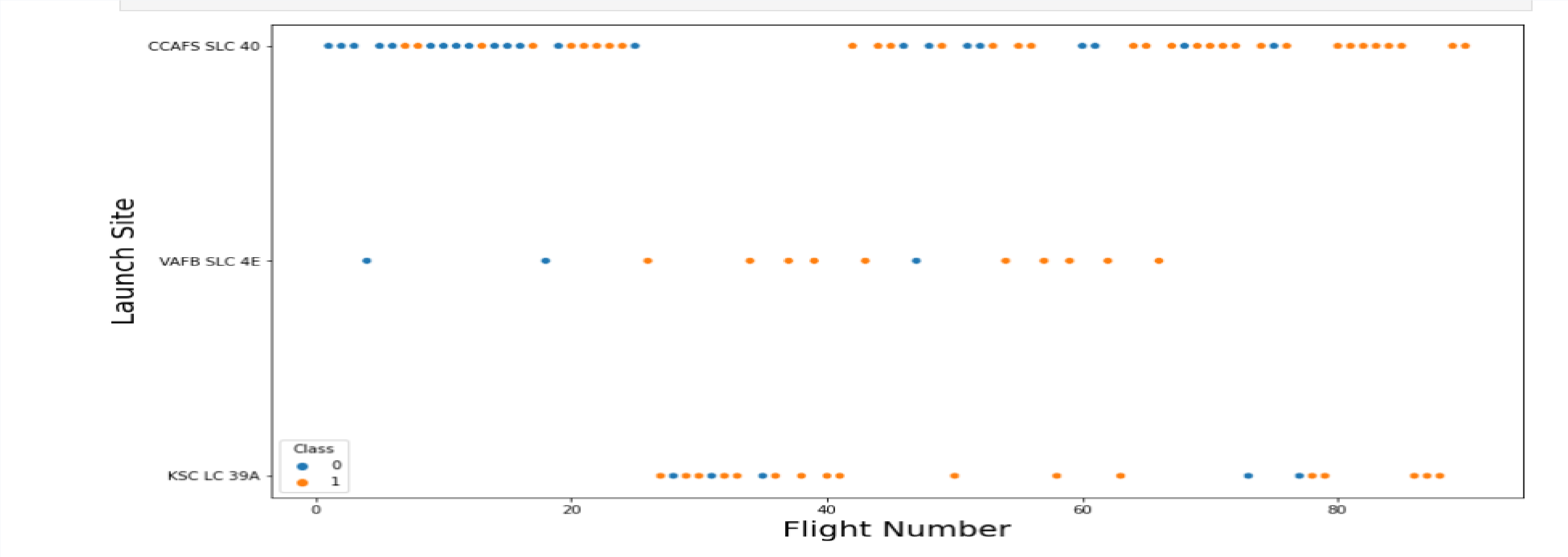
Results

- From the data visualization section, we can see that some features may have correlation with the mission outcome in several ways. For example, with heavy payloads the successful landing or positive landing rate are more for orbit types Polar, LEO and ISS. However, for GTO, we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission) are both there here.
- Therefore, each feature may have a certain impact on the final mission outcome. The exact ways of how each of these features impact the mission outcome are difficult to decipher. However, we can use some machine learning algorithms to learn the pattern of the past data and predict whether a mission will be successful or not based on the given features.

Section 2

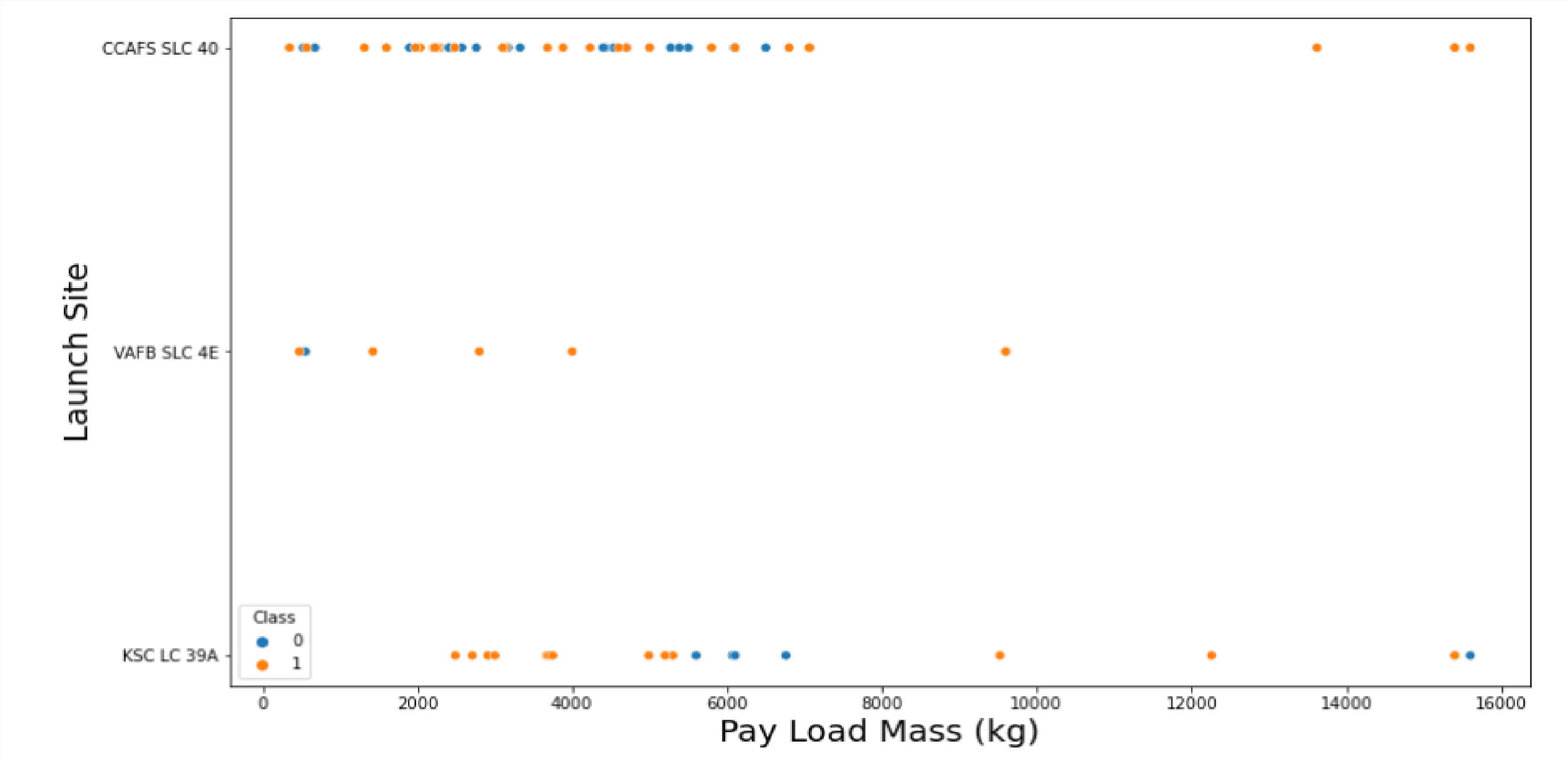
Insights drawn from EDA

Flight Number vs. Launch Site



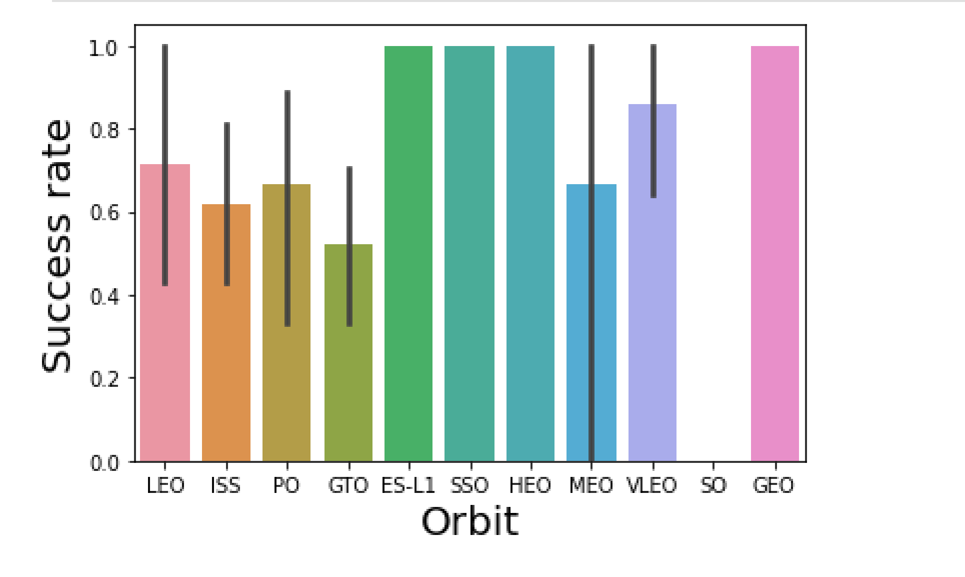
- Graphic suggests an increase in success rate over time (indicated in Flight Number). Likely a big breakthrough around flight 20 which significantly increased success rate. CCAFS appears to be the main launch site as it has the most volume.

Payload vs. Launch Site



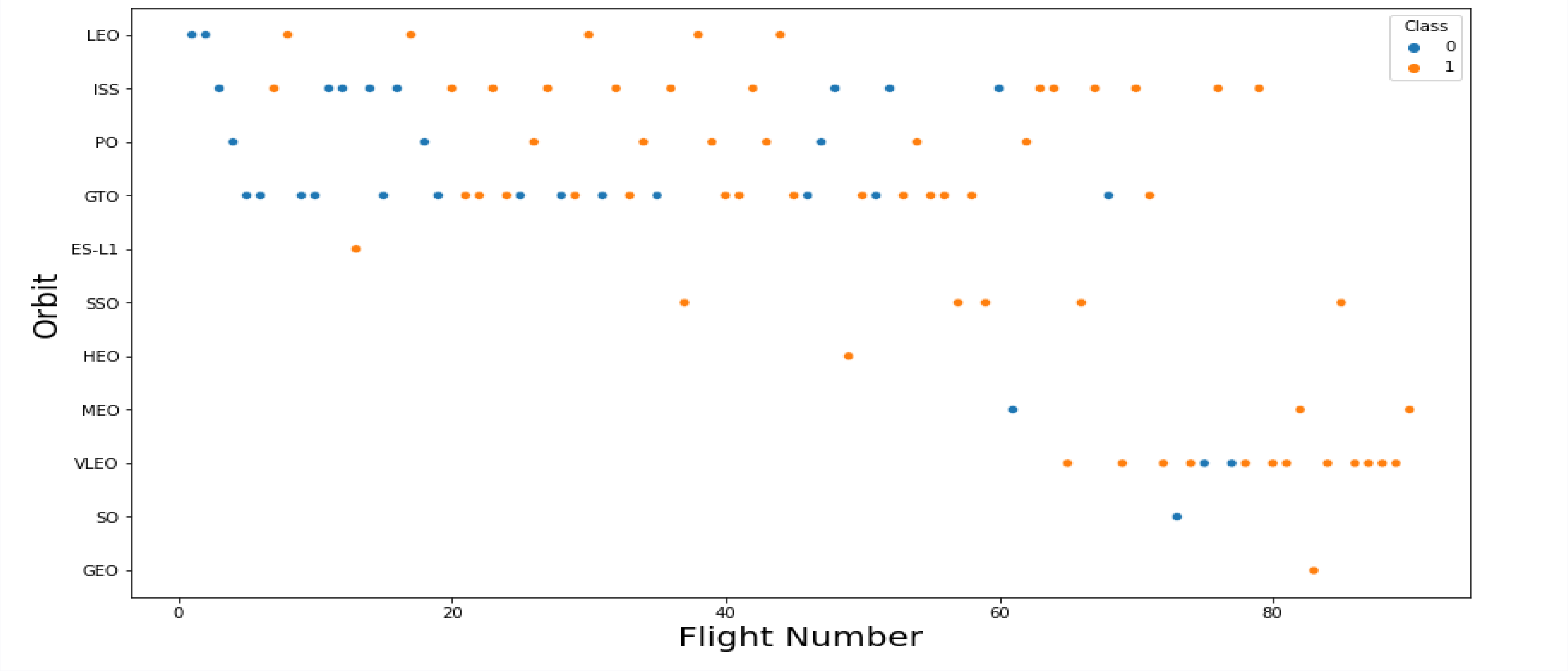
Payload mass appears to fall mostly between 0-7000 kg. Difference launch sites also seem to use different payload mass.

Success Rate vs. Orbit Type



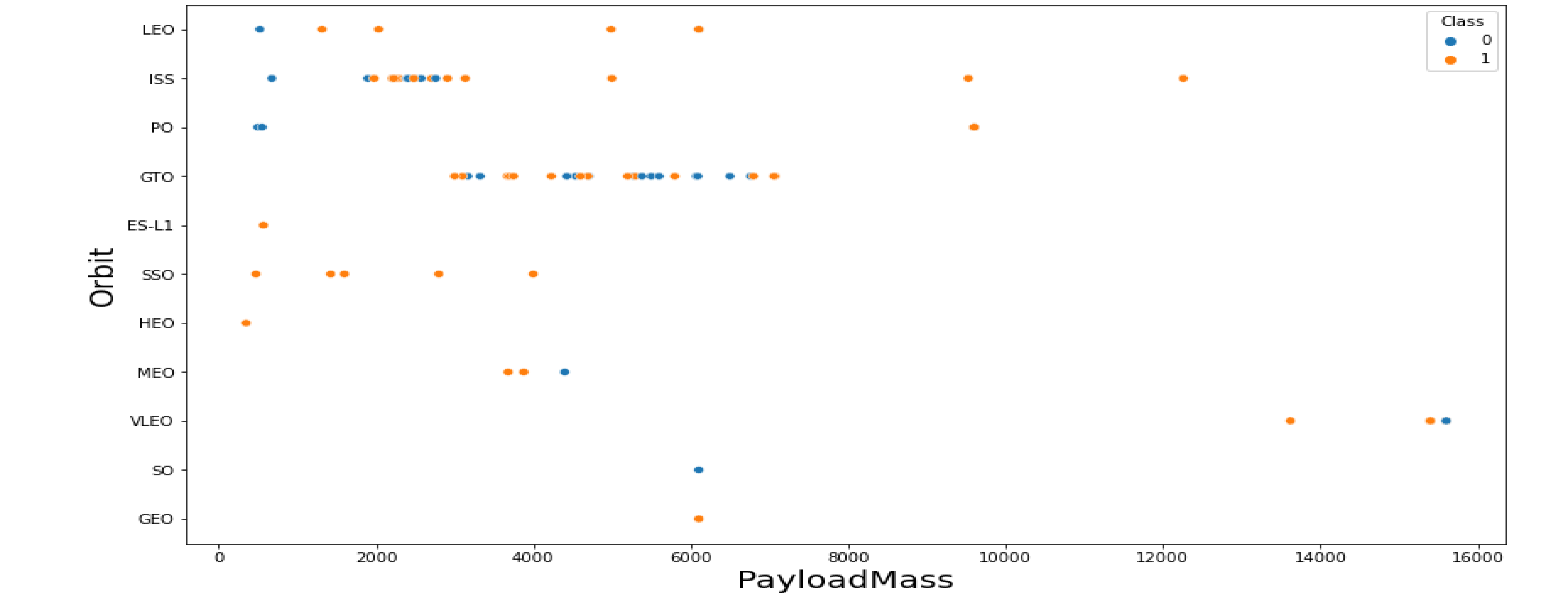
- ES-L1(1),GEO(1), HEO(1)have100% success rate(sample sizes in parenthesis)
- SSO(5)has 100% success rate
- VLEO(14) has decent success rate and attempts
- SO(1) has 0% success rate
- GTO(27) has around 50% success rate but largest sample

Flight Number vs. Orbit Type



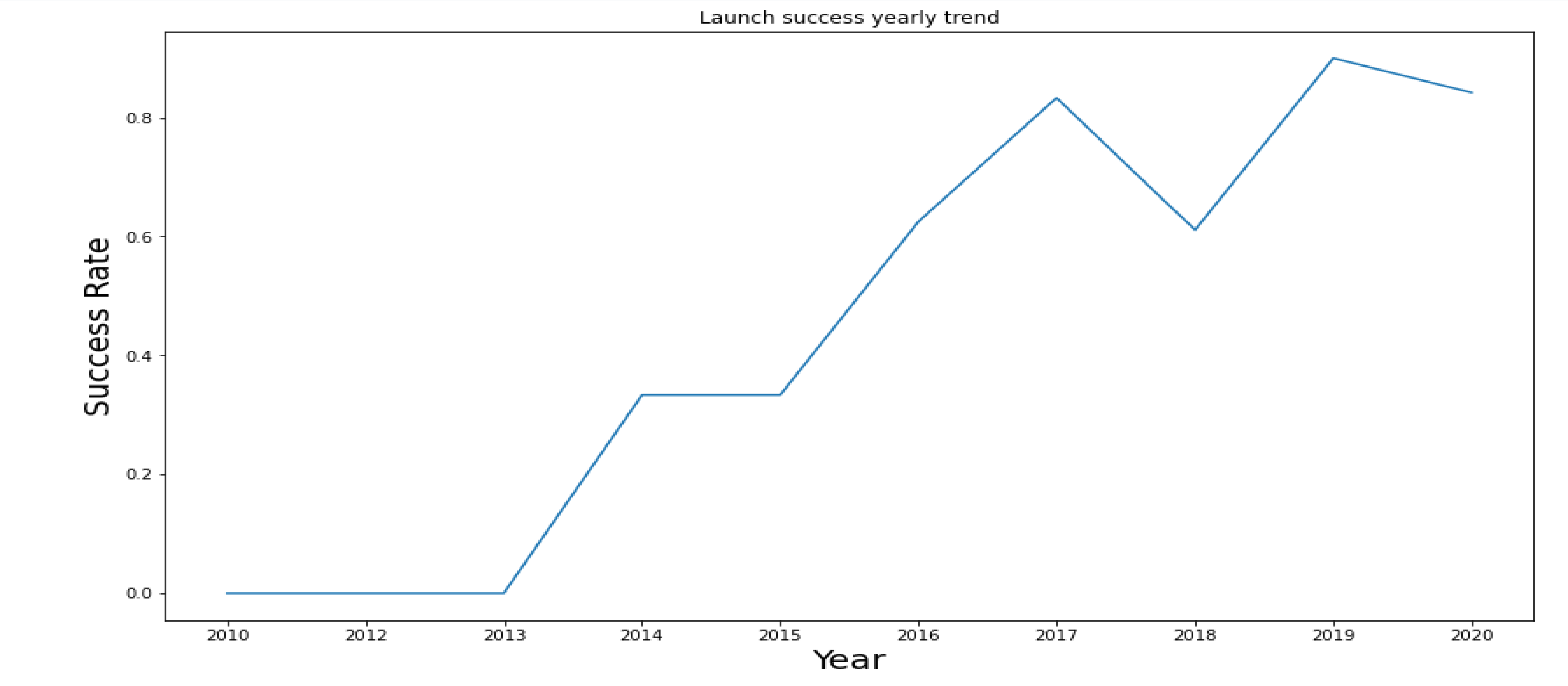
- Launch Orbit preferences changed over Flight Number.
- Launch Outcome seems to correlate with this preference.
- SpaceX started with LEO orbits which saw moderate success LEO and returned to VLEO in recent launches
- SpaceX appears to perform better in lower orbits or Sun-synchronous orbits

Payload vs. Orbit Type



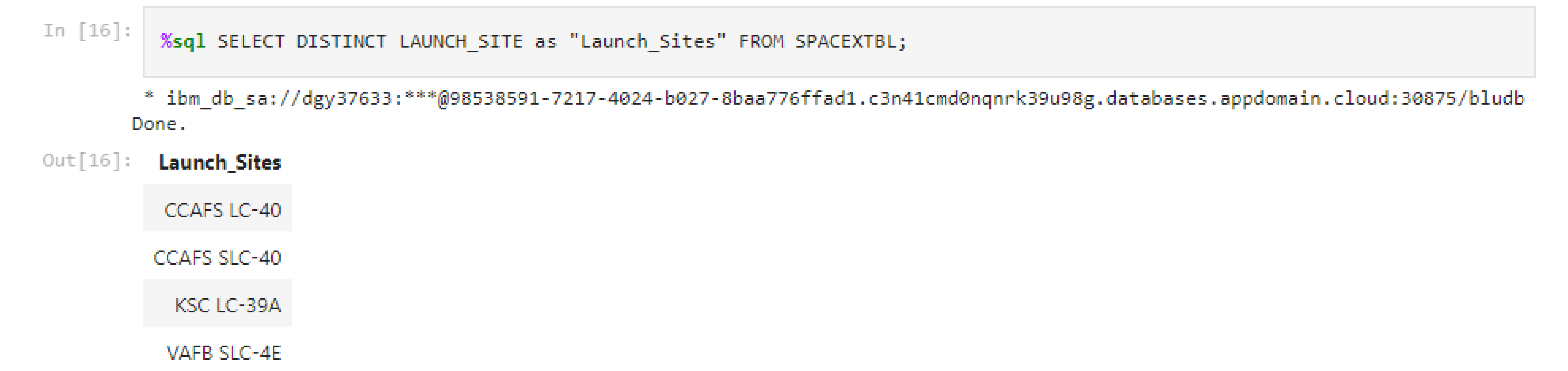
- Payload mass seems to correlate with orbit
- LEO and SSO seem to have relatively low payloadmass
- The other most successful orbit VLEO only has payloadmass values in the higher end of the range

Launch Success Yearly Trend



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

All Launch Site Names



In [16]:

```
%sql SELECT DISTINCT LAUNCH_SITE as "Launch_Sites" FROM SPACEXTBL;
```

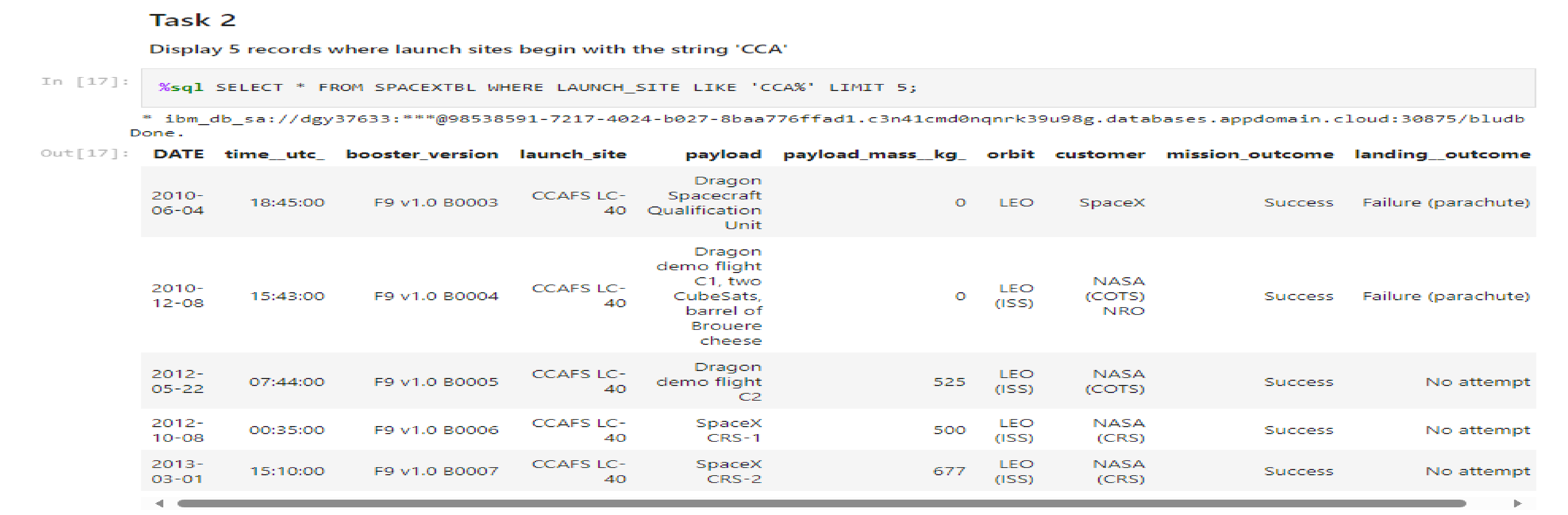
```
 * ibm_db_sa://dgy37633:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30875/bludb
Done.
```

Out[16]:

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

- 4 unique launch sites were found from database using the sql command Distinct.

Launch Site Names Begin with 'CCA'



Task 2

Display 5 records where launch sites begin with the string 'CCA'

```
In [17]:  %sql SELECT * FROM SPACEXTBL WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5;
```

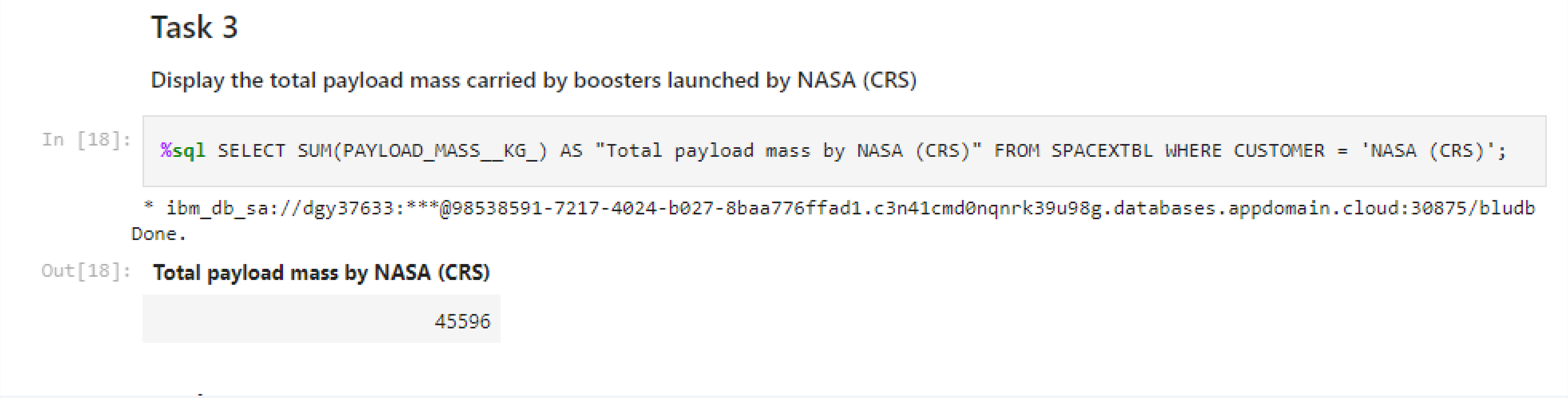
```
 * ibm_db_sa://dgy37633:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30875/bludb
Done.
```

Out[17]:

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



Task 3

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

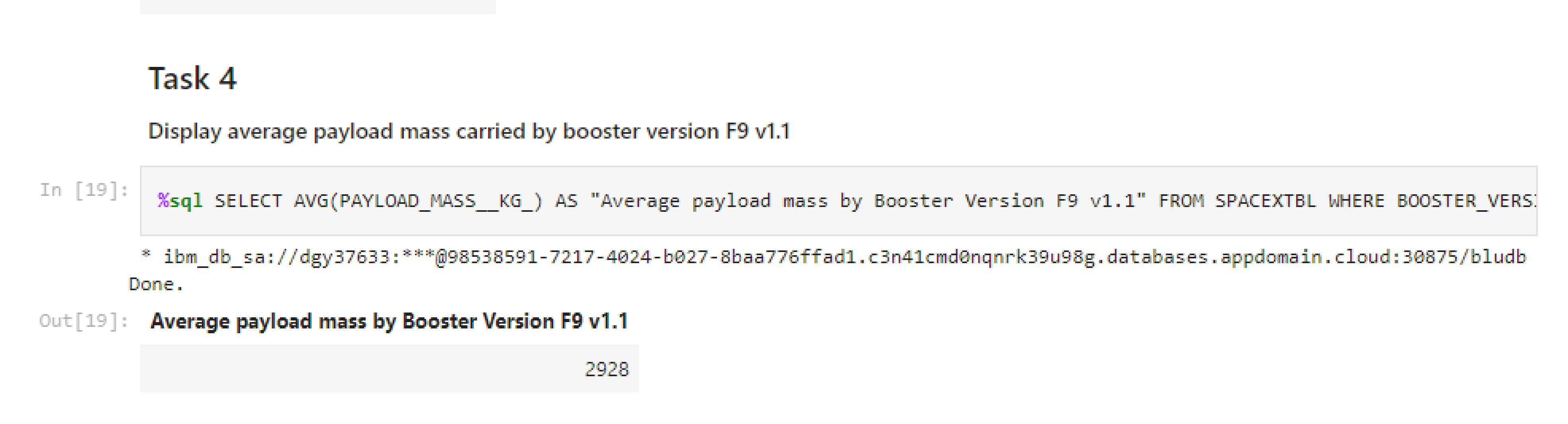
```
In [18]: %sql SELECT SUM(PAYLOAD_MASS__KG_) AS "Total payload mass by NASA (CRS)" FROM SPACEXTBL WHERE CUSTOMER = 'NASA (CRS)';

 * ibm_db_sa://dgy37633:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30875/bludb
Done.

Out[18]: Total payload mass by NASA (CRS)
                                   45596
```

- The query sums the total payload mass in kg where NASA was the customer

Average Payload Mass by F9 v1.1



Task 4

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

In [19]:
```
%sql SELECT AVG(PAYLOAD_MASS__KG_) AS "Average payload mass by Booster Version F9 v1.1" FROM SPACEXTBL WHERE BOOSTER_VERS
```

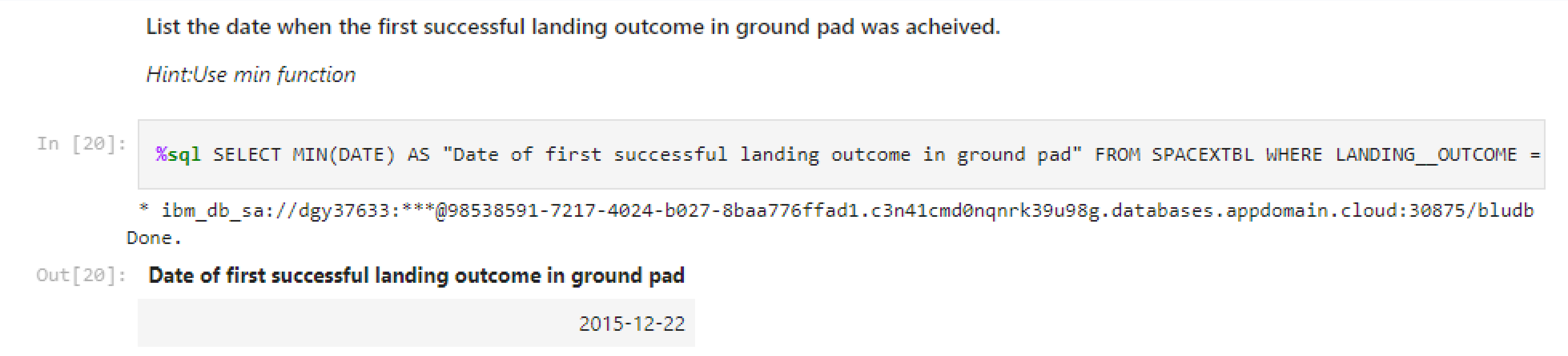
```
 * ibm_db_sa://dgy37633:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30875/bludb
Done.
```

Out[19]:

Average payload mass by Booster Version F9 v1.1
2928

- This query calculates the average payload mass or launches which used booster version F9v1.1
- Average payload mass of F9 1.1 is on the lower end of our payload mass range

First Successful Ground Landing Date



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

Hint:Use min function

In [20]:

```
%sql SELECT MIN(DATE) AS "Date of first successful landing outcome in ground pad" FROM SPACEXTBL WHERE LANDING__OUTCOME =
```

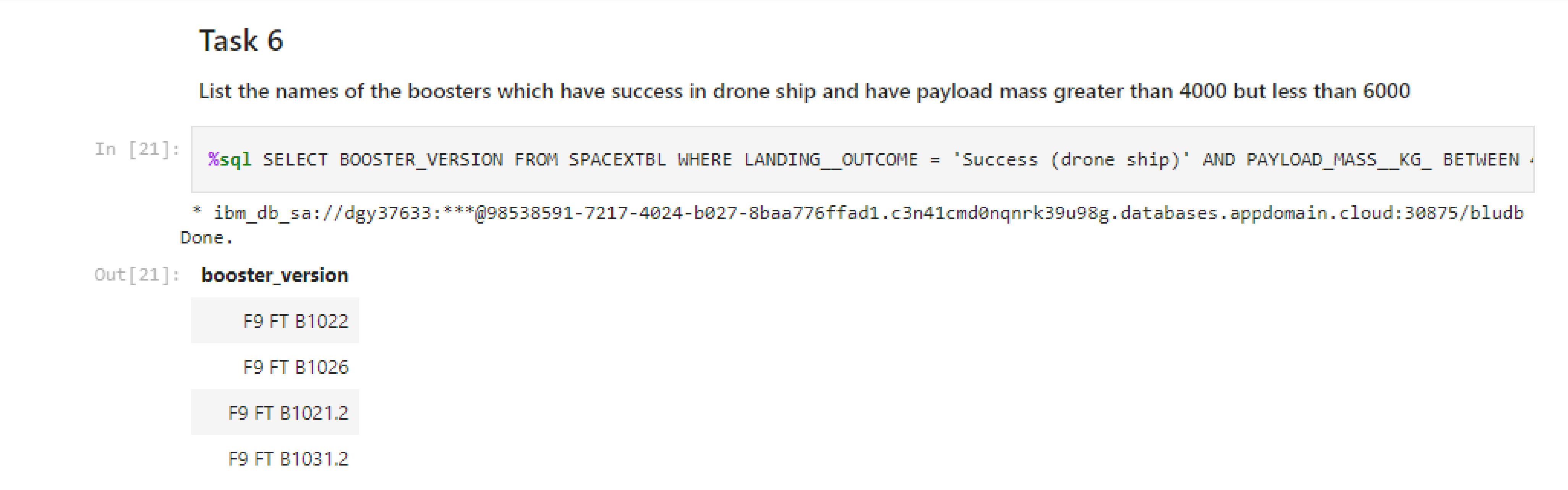
```
 * ibm_db_sa://dgy37633:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30875/bludb
Done.
```

Out[20]:

Date of first successful landing outcome in ground pad
2015-12-22

- The query indicates that the first successful landing outcome happened on 22nd December, 2015

Successful Drone Ship Landing with Payload between 4000 and 6000



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 [21]:
```
%sql SELECT BOOSTER_VERSION FROM SPACEXTBL WHERE LANDING__OUTCOME = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ BETWEEN
```

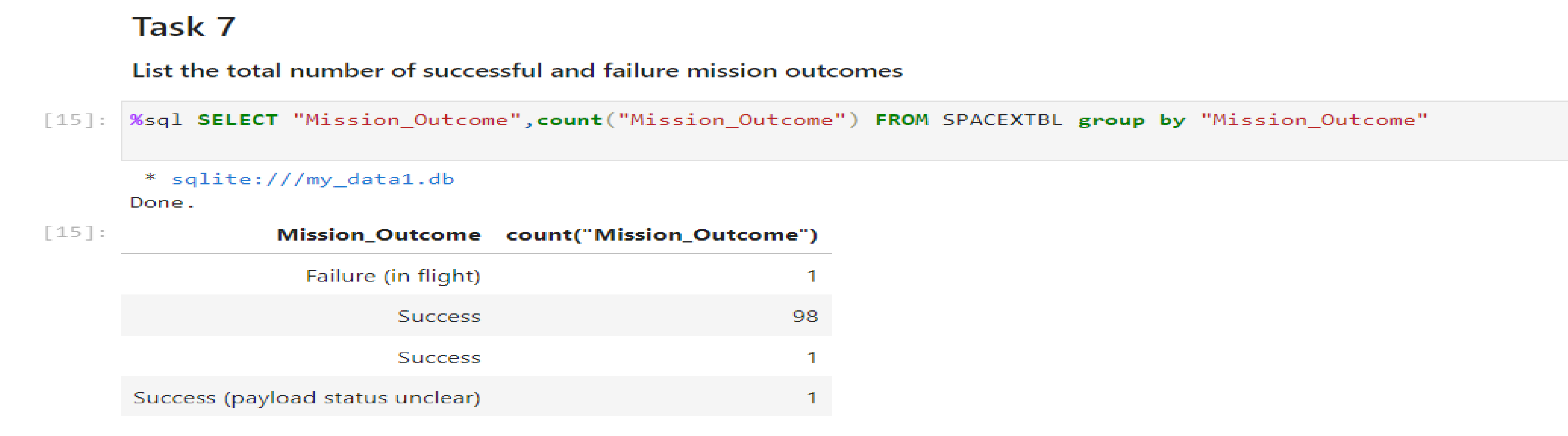
```
 * ibm_db_sa://dgy37633:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30875/bludb
Done.
```

Out[21]:

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 no inclusively.

Total Number of Successful and Failure Mission Outcomes



Task 7

List the total number of successful and failure mission outcomes

```
[15]: %sql SELECT "Mission_Outcome",count("Mission_Outcome") FROM SPACEXTBL group by "Mission_Outcome"
```

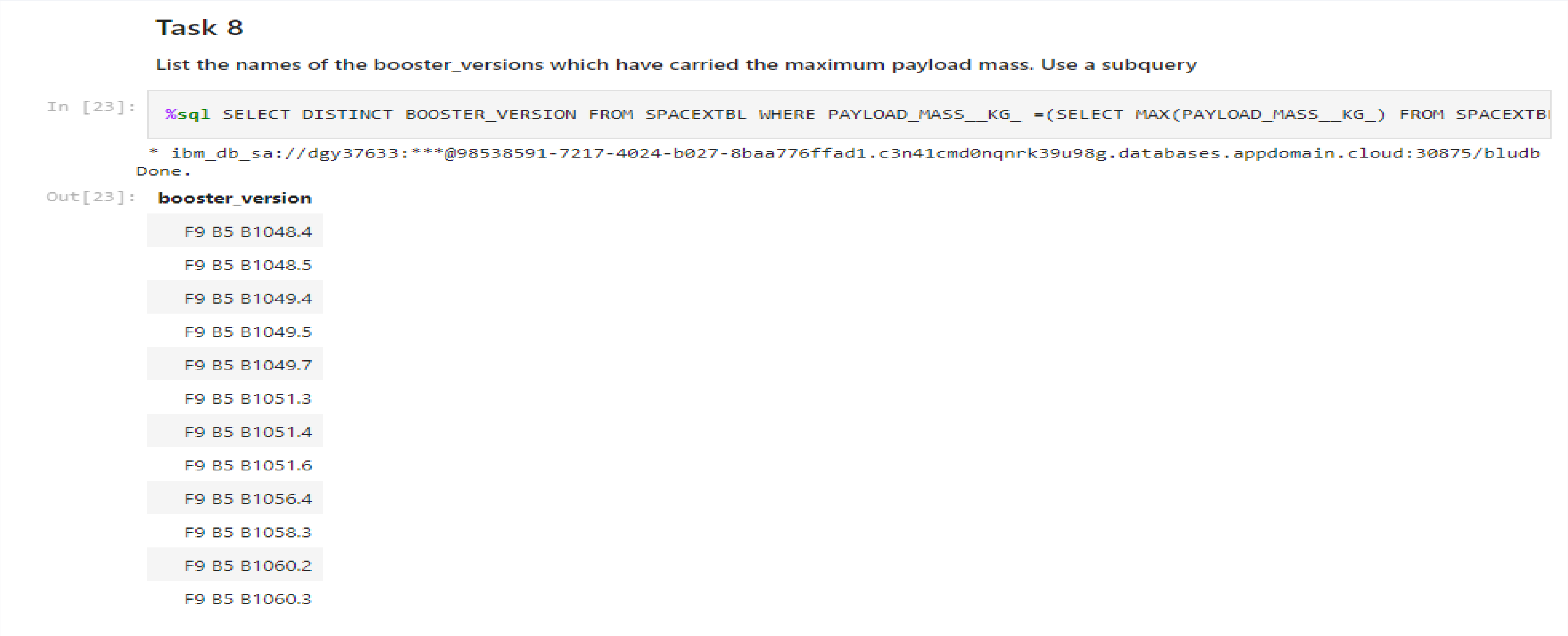
```
 * sqlite:///my_data1.db
Done.
```

[15]:

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

- SpaceX appears to achieve its mission outcome nearly 99% of the time. One launch has an unclear payload status and unfortunately one failed in flight.

Boosters Carried Maximum Payload



Task 8

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

```
In [23]:  %sql SELECT DISTINCT BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ =(SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTB
```

```
 * ibm_db_sa://dgy37633:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30875/bludb
Done.
```

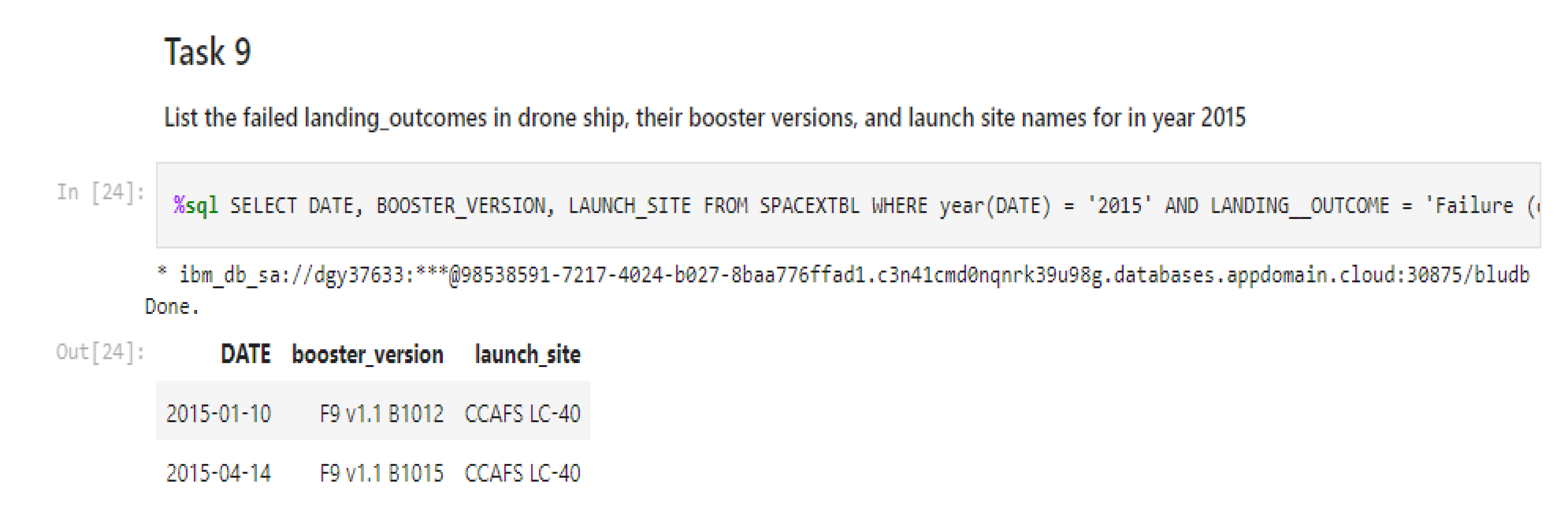
Out[23]:

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

- This query returns the booster versions that carried the highest paload mass of 156000 kg. These booster versions are very similar and all are of the F9 B5 B10xx.x variety.

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

2015 Launch Records



Task 9

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

In [24]:

```
%sql SELECT DATE, BOOSTER_VERSION, LAUNCH_SITE FROM SPACEXTBL WHERE year(DATE) = '2015' AND LANDING__OUTCOME = 'Failure (
```

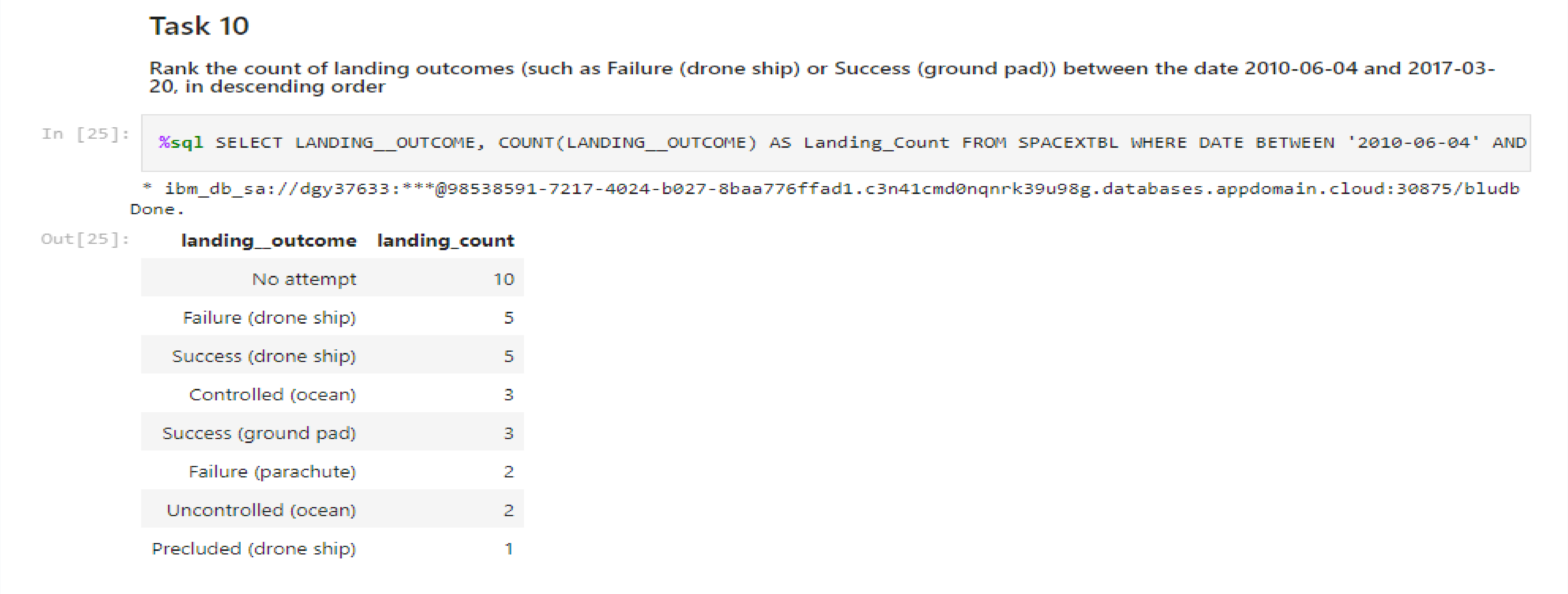
```
 * ibm_db_sa://dgy37633:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30875/bludb
Done.
```

Out[24]:

DATE	booster_version	launch_site
2015-01-10	F9 v1.1 B1012	CCAFS LC-40
2015-04-14	F9 v1.1 B1015	CCAFS LC-40

- This query shows the failed landing ourcomes in drone ship, their booster versions and launch site names for 2015

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



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 [25]: %sql SELECT LANDING__OUTCOME, COUNT(LANDING__OUTCOME) AS Landing_Count FROM SPACEXTBL WHERE DATE BETWEEN '2010-06-04' AND
```

```
 * ibm_db_sa://dgy37633:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30875/bludb
Done.
```

Out[25]:

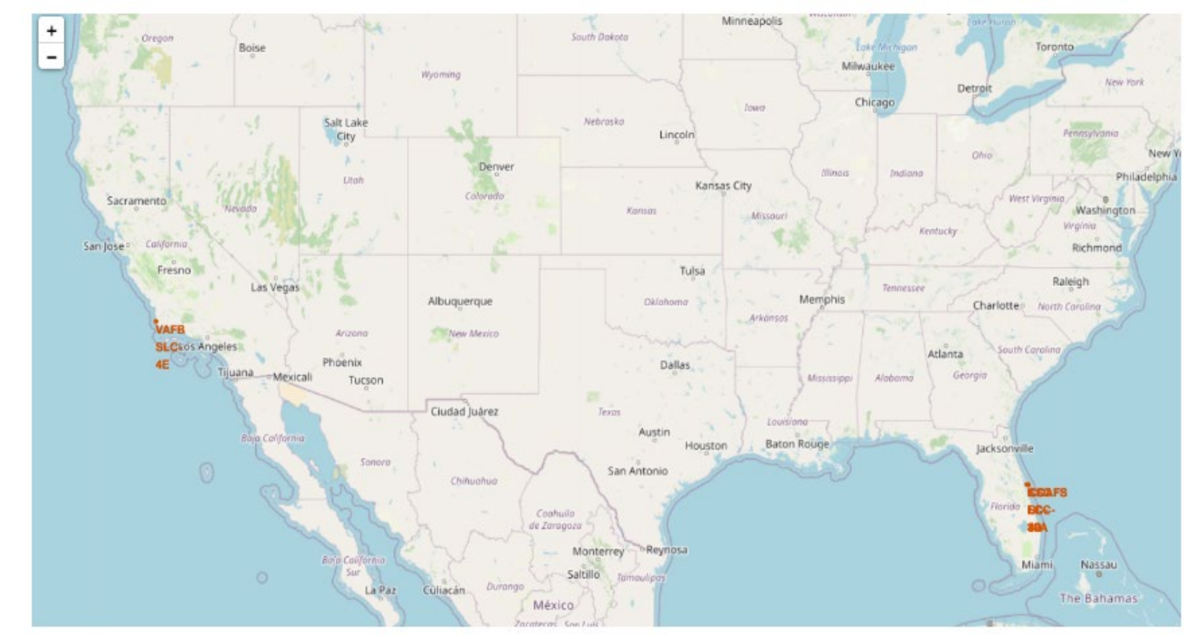
landing__outcome	landing_count
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

- This query ranks 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

Section 3

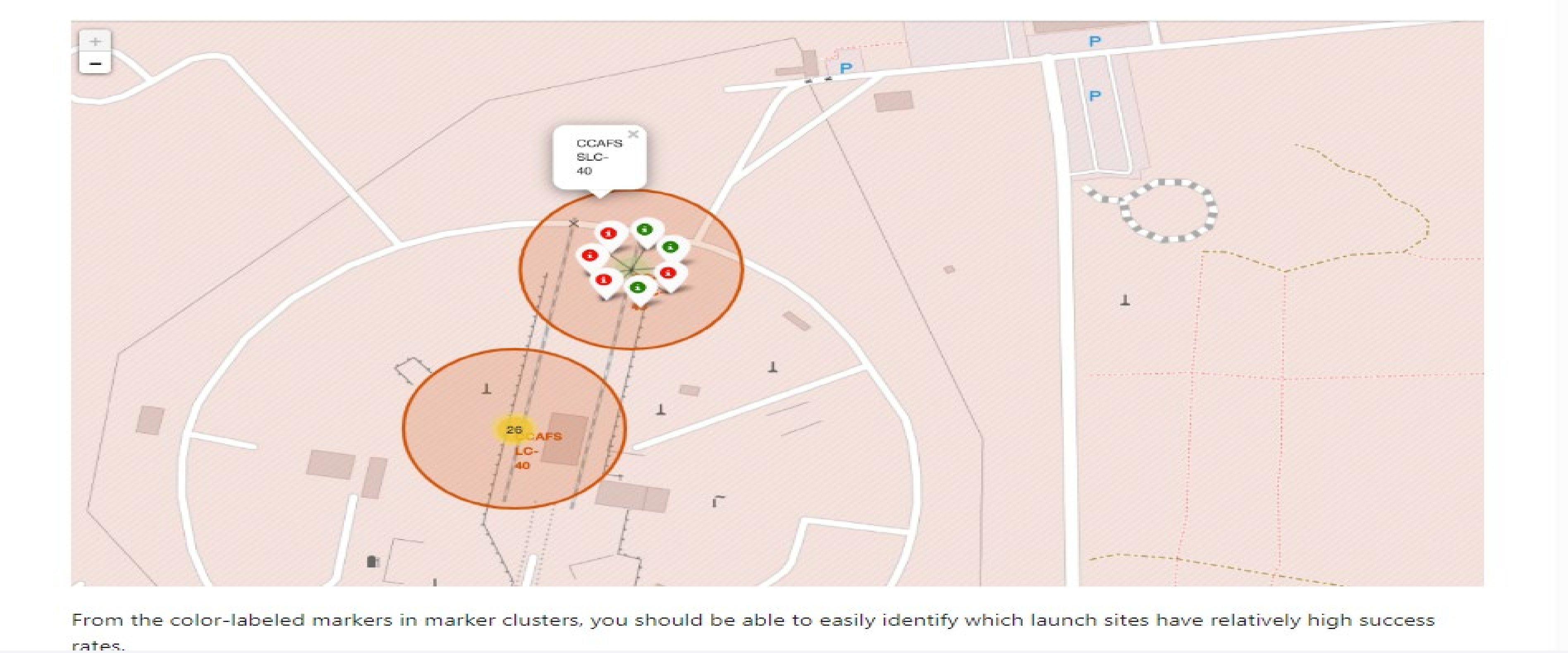
Launch Sites Proximities Analysis

Launch Site Locations



- All launch sites are near the ocean.

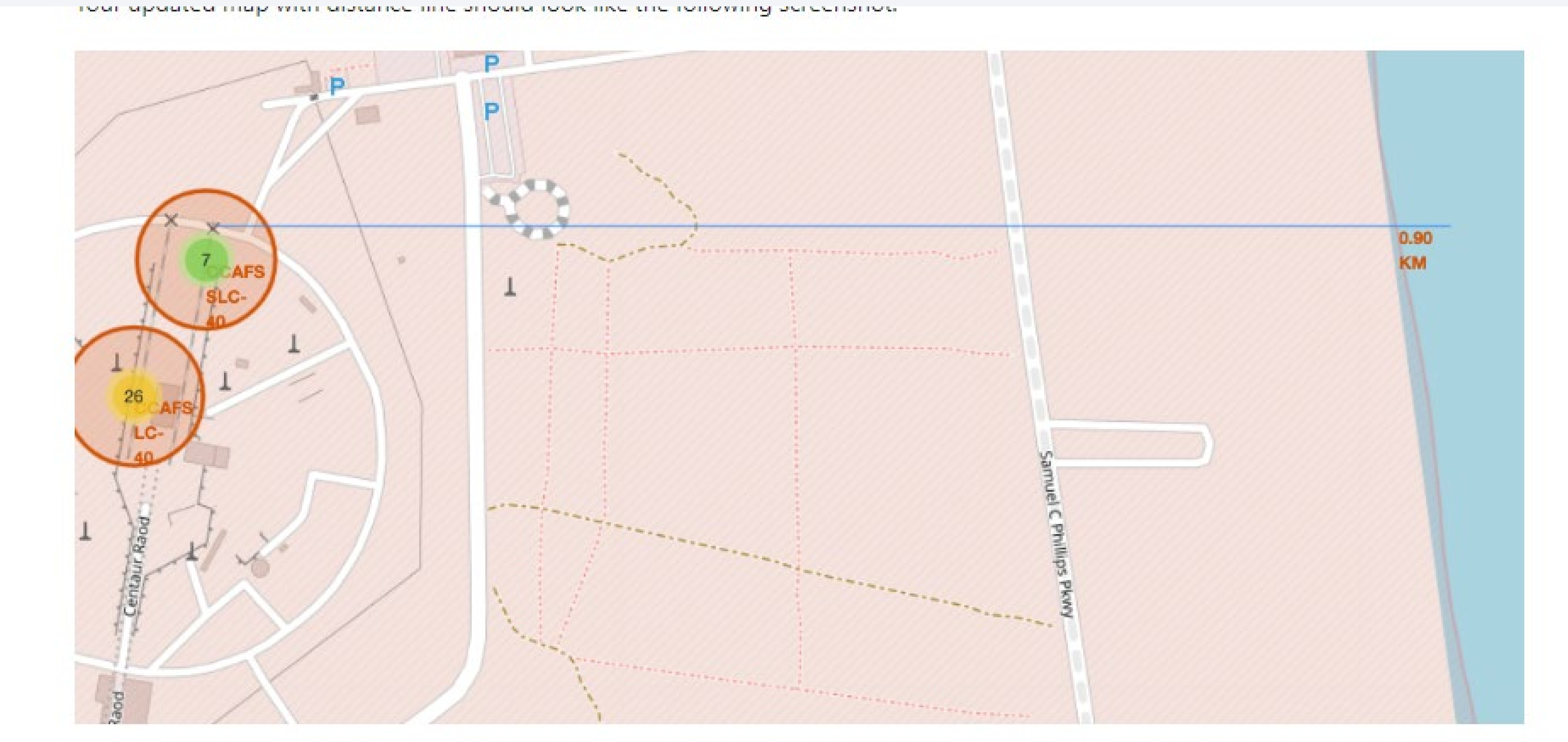
Color Coded Launch Markers



From the color-labeled markers in marker clusters, you should be able to easily identify which launch sites have relatively high success rates.

- Clusters on Folium map can be clicked on to display each successful landing (green icon) and failed landing(red icon).InthisexampleVAFBSLC-4Eshows4successfullandingsand6failedlandings.

Key Location Proximities

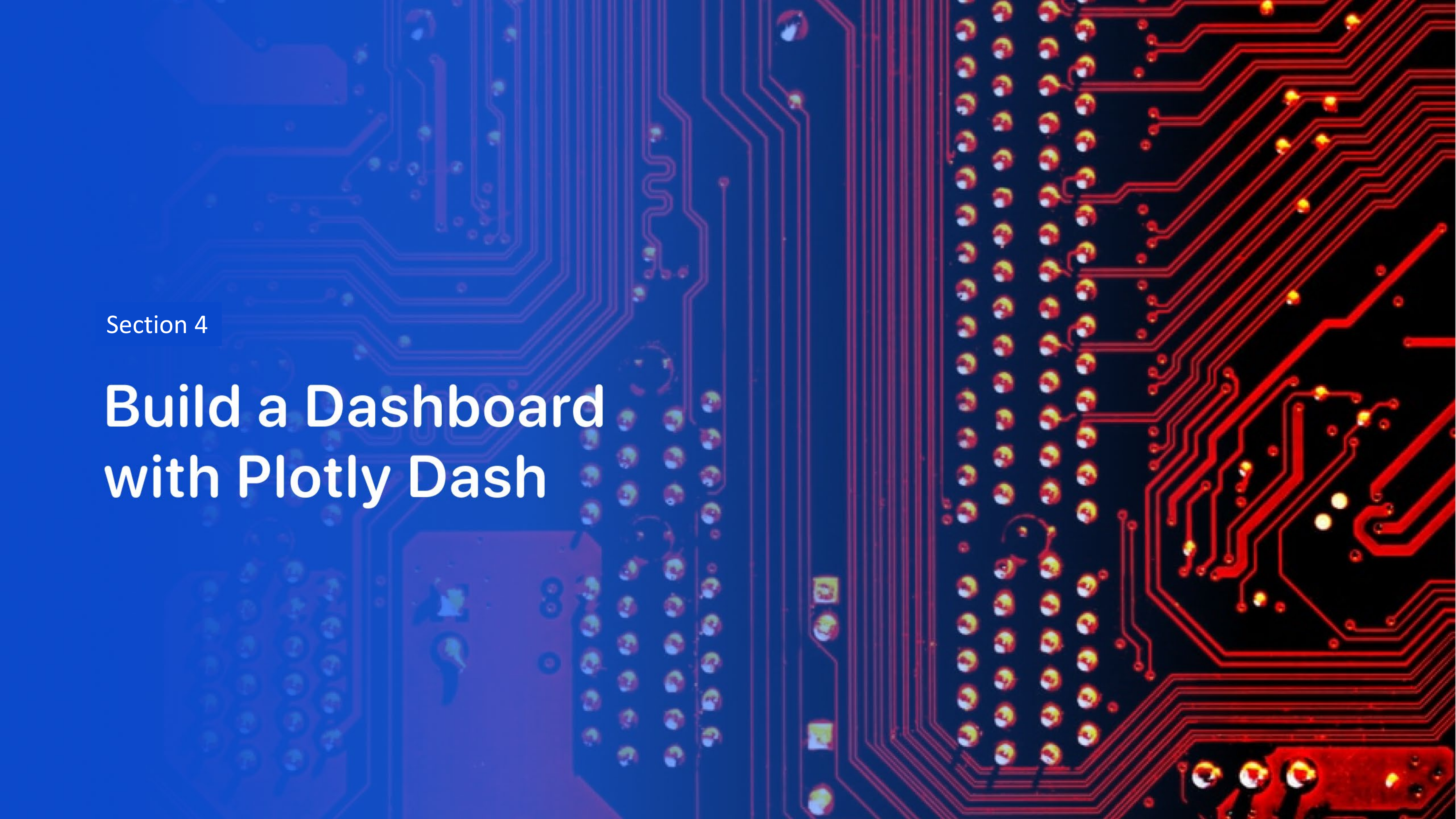


Your updated map with distance line should look like the following screenshot.

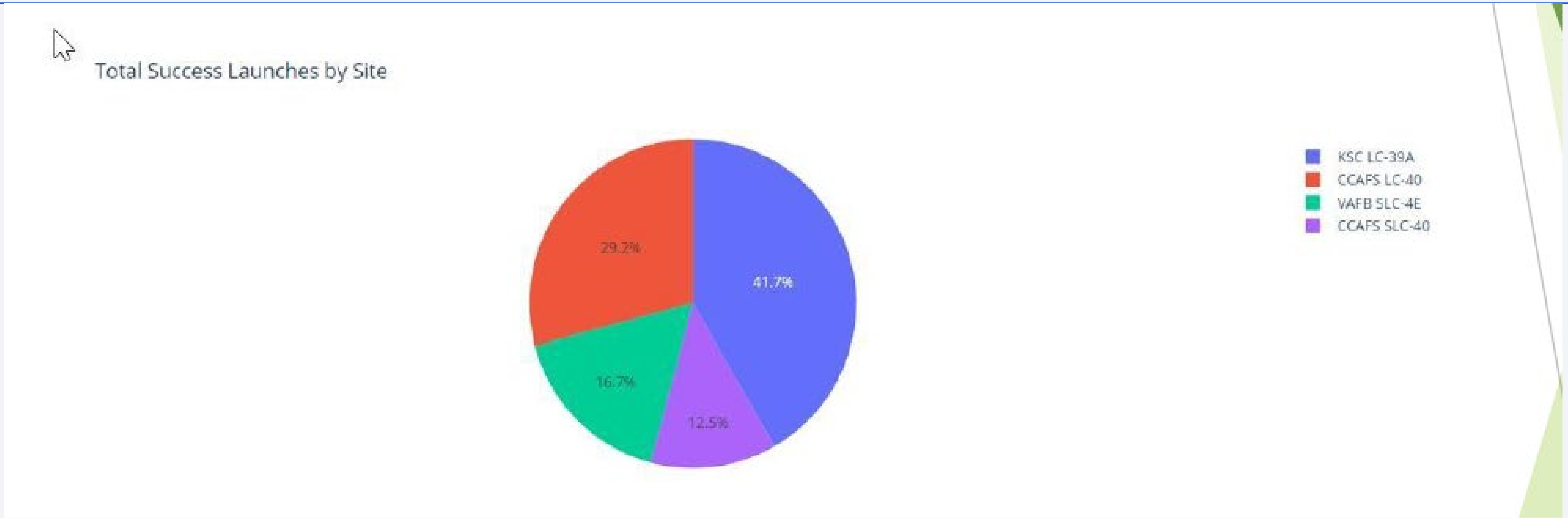
- Launch sites are very close to railways for large part and supply transportation. Launch sites are close to highways for human and supply transport. Launch sites are also close to coasts and relatively far from cities to that launch failures can land in the sea to avoid rockets falling on densely populated areas.

Section 4

Build a Dashboard with Plotly Dash

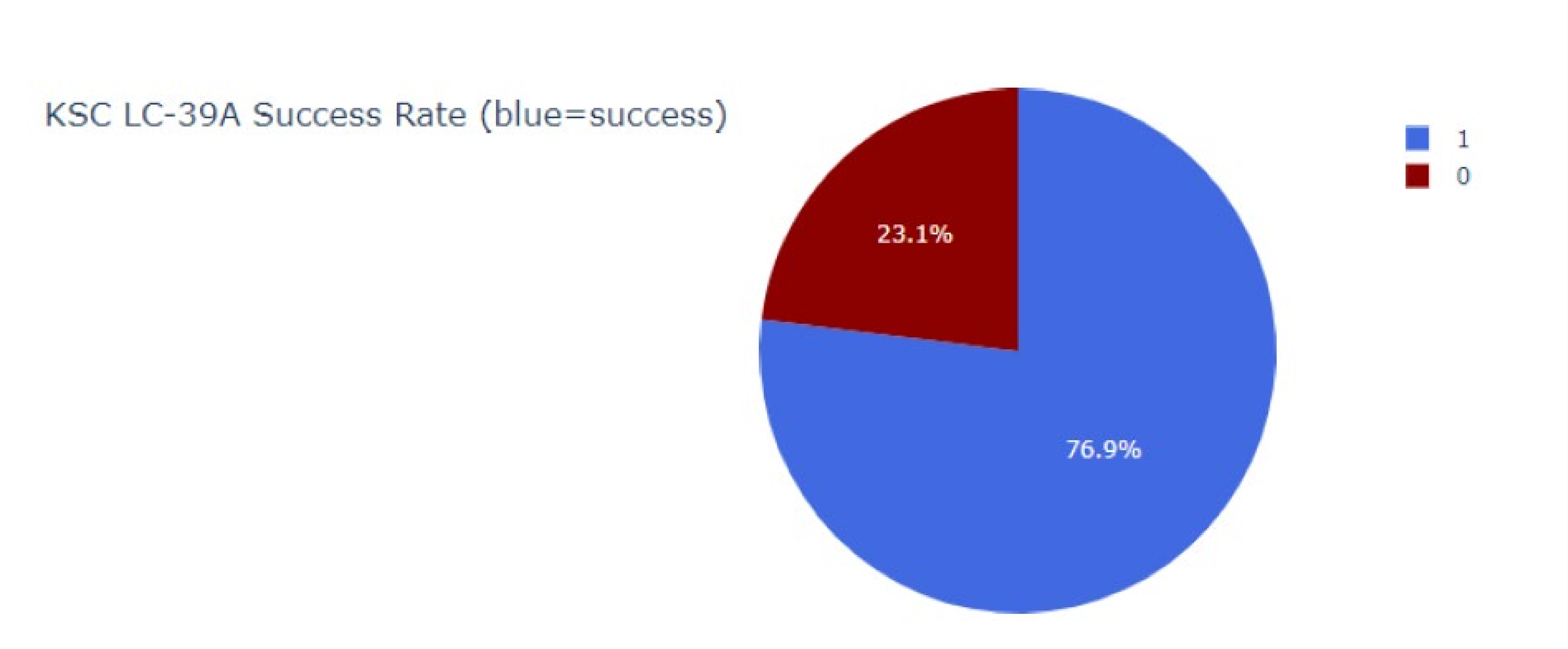


Successful Launches Across Launch Sites



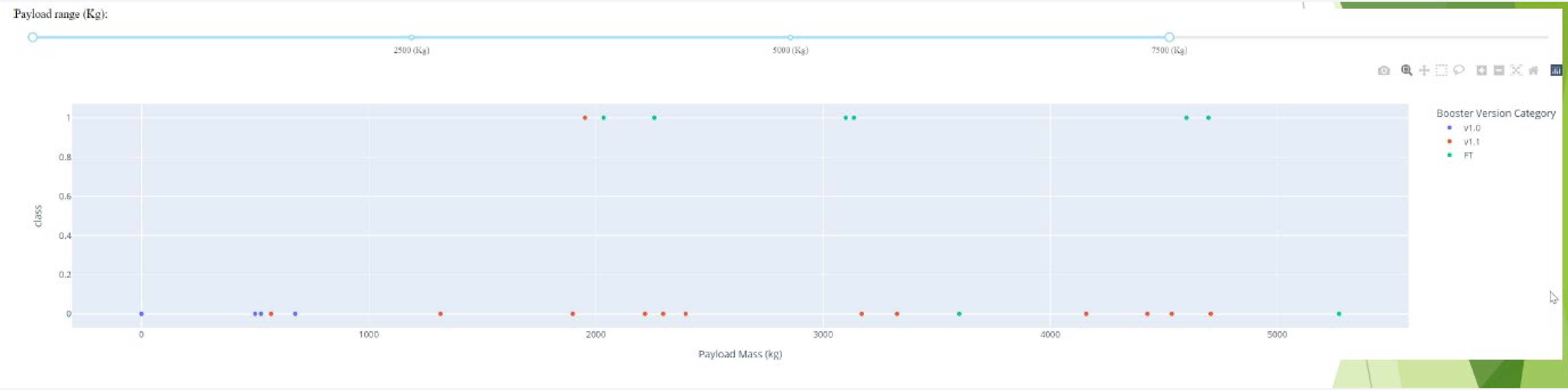
- This is the distribution of successful landings across all launch sites. CCAFS LC-40 is the old name of CCAFS SLC-40 so CCAFS and KSC have the same amount of successful landings, but a majority of the successful landings where performed before the name change. VAFB has the smallest share of successful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

Highest Success Rate Launch Site



- KSCLC-39Ahasthehighestsuccessratewith10 successfullandingsand3failedlandings.

Payload Mass vs. Success vs. Booster Version Category

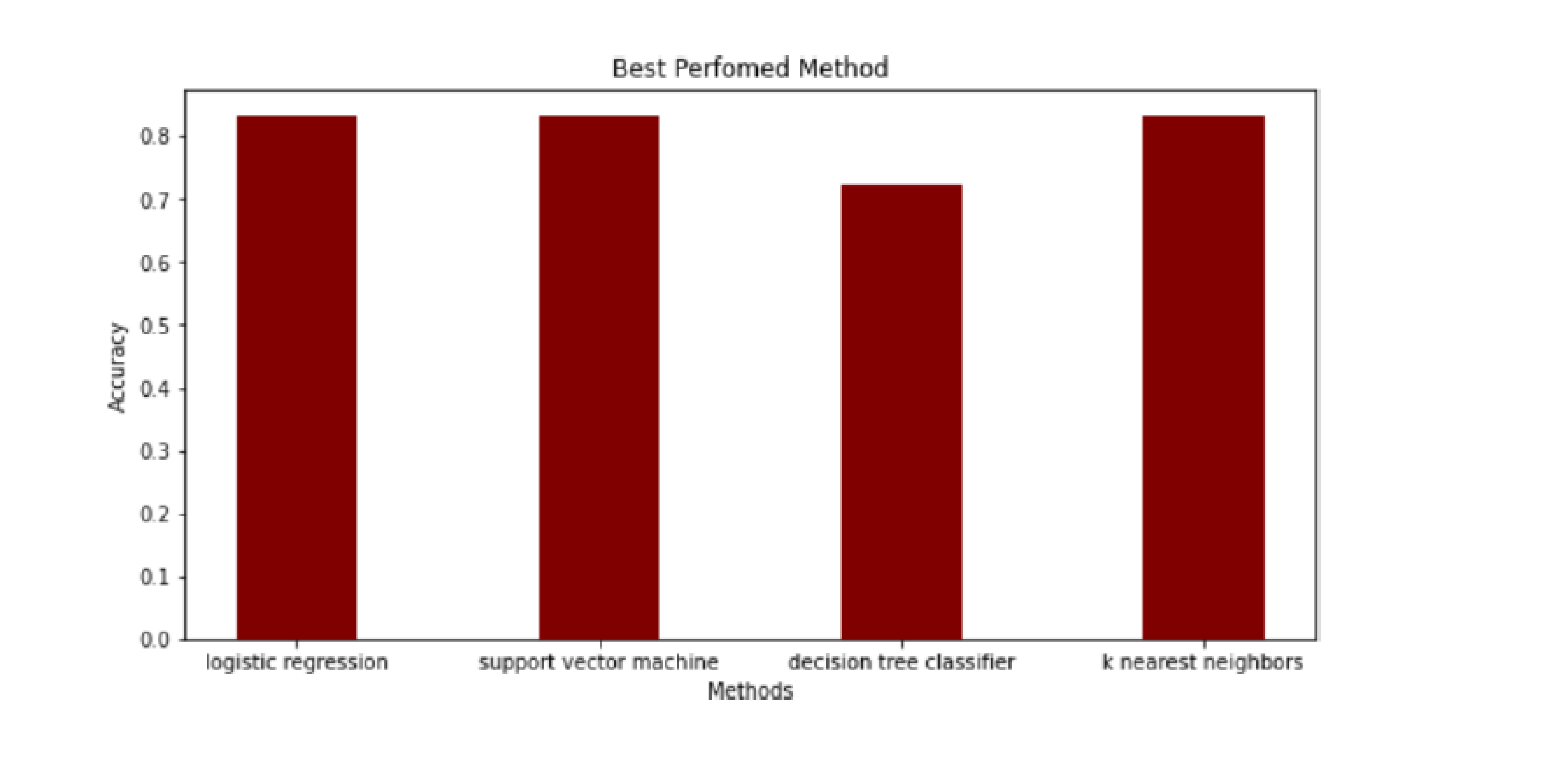


- Plotly dashboard has a Payload range selector. However, this is set from 0-10000 instead of the max Payload of 15600. Class indicates 1 for successful landing and 0 for failure. Scatter plot also accounts for booster version category in color and number of launches in point size. In this particularrangeof0-7500,interestinglytherearetwofailedlandings with payloads of zero kg.

Section 5

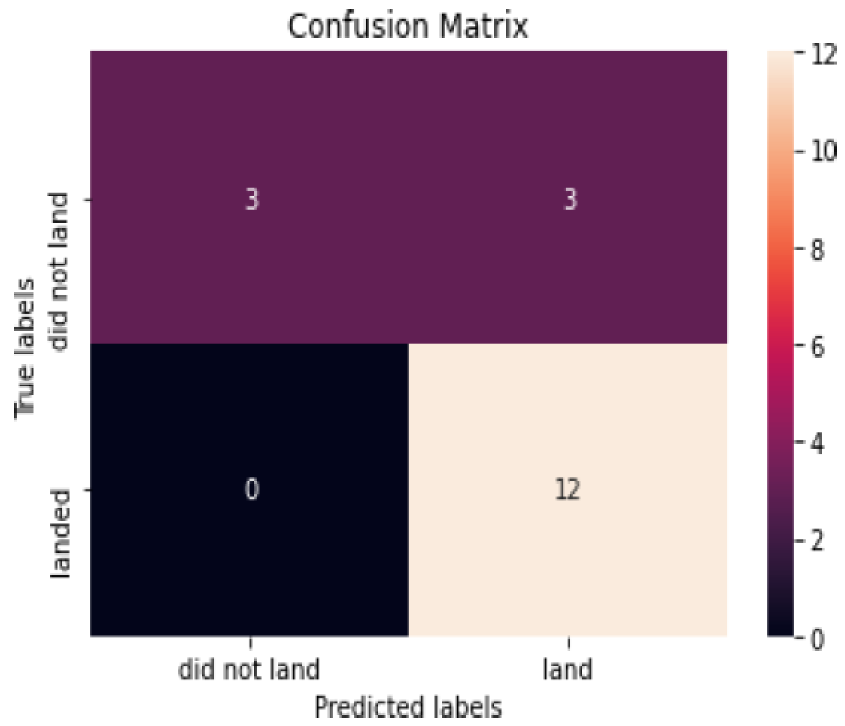
Predictive Analysis (Classification)

Classification Accuracy



- The models had virtually the same accuracy on the test set at 83.33%accuracy, except the decisiontreeclassifierwith72.23%.
- 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.

- 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.

Conclusions

- ◦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 Stage1 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 in to a DB2 SQL database
- ◦Created a dashboard for visualization
- ◦We created a machine learning model with an accuracy of 83%
- ◦Allon Mask of Space Y 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
- ◦More data should be collected to better determine the best machine learning model and improve the accuracy

Appendix

- GitHub repository URL:

AnthonyWen/Capstone (github.com)

Thank you!