Data Science Capstone Project

Ünal Ferhat

10/09/2023

https://github.com/Ferhat-unal/IBM-Data-Science-Professional-Certification.git

OUTLINE

- Executive Summary
- Introduction
- Methodology
- Results
  - Visualization – Charts
  - Dashboard
- Discussion
  - Findings & Implications
- Conclusion
- Appendix

Executive Summary

- **Methodology Summary**
- Data was extracted from the SpaceX Wikipedia page and public SpaceX API. The column ‘Class' was created to categorize true successful landings. Utilized SQL, data visualization, folium maps, and plotly dashboards to explore the data. Compiled important columns for use as features for predictive analysis. Used one hot encoding to convert all categorical variables to binary values and all numeric columns to float64 data type. GridSearchCV was used to determine the ideal parameters for machine learning models using standardized data. Displayed the accuracy rating for each model and confusion matrix for best performing model(s).

- **Results Summary**
- Four machine learning models were used for predictive analysis: Logistic Regression, Support Vector Machines, Decision Tree Classifier, and K Nearest Neighbors (KNN). All models (except for Decision Tree Classifier – 88,88%) produced similar results around 83.33% accuracy rate. All models over predicted true successful landings and more data will be required for better model determination. The dataset we(ve used had only 90 records.

INTRODUCTION

- **Project Background**

→ During the existing commercial space age, companies are making space travel affordable for everyone

→ Space X conducts most inexpensive launches ($62 million vs. $165 million)

→ Due to recovery of rocket parts (Stage One)

→ Space Y wants to compete with Space X

- **Problem**

- → we’re trying to predict successful stage one recovery by creating a machine learning modelt for Space Y

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Collected and combined data utilizing public Space X API and scraping Space X Wikipedia page
- Perform data wrangling
  - Classified true landings as successful (class=1) and unsuccessful (class=0)
- 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 standardized data and finding the best parameters using GridSearchCV

Data Collection

- Data Collection process involved a combination of API requests from Space X public API and webscraping data from a table in SpaceX's Wikipedia entry

The next two slides shows flowcharts of data collection from public Space X API and data collection from webscraping

Space X Data Columns:
FlightNumber, Date, BoosterVersion, PayloadMass, Orbit, LaunchSite, Outcome, Flights, GridFins,
Reused, Legs, LandingPad, Block, ReusedCount, Serial, Longitude, Latitude

Wikipedia Webscrape Data Columns:
Flight No., Launch site, Payload, PayloadMass, Orbit, Customer, Launch outcome, Version Booster,
Booster landing, Date, Time

Data Collection – SpaceX API

- → Request (Space X API)
- → JSON file + Lists(Launch site, Booster Version, Payload Data, Cores
- → json_normalize to convert JSON file to DataFrame format
- → Combined data columns into dictionary data type
- → Cast Dictionary format into DataFrame
- → Filter data to only Flacon 9 launches
- → Impute Payload data with it’s average value

GitHub URL

IBM-Data-Science-Professional-Certification/test2 (1).ipynb at main · Ferhat-unal/IBM-Data-Science-Professional-Certification (github.com)

Data Collection - Scraping

- Request Wikipedia html
- BeautifulSoup html5lib parser
- Find launch info html table
- Create dictionary
- iterate through table cells to extract data to dictionary
- convert dictionary to DataFrame

GitHub URL

IBM-Data-Science-Professional-Certification/data collection with web scrapping.ipynb at main · Ferhat-unal/IBM-Data-Science-Professional-Certification (github.com)

Data Wrangling

- Enumerate all successful and unsuccessful mission outcomes
- Create set of bad outcomes where mission outcome is ‘False’ or ‘None’
- Create training label column ‘Class’ with landing outcomes where landing_class = 0 if bad_outcome and landing_class = 1 if successful outcome

Value mapping:

Class = 1

True ASDS, True RTLS, and True Ocean

Class = 0

None None, False ASDS, None ASDS, False Ocean, False RTLS

GitHub URL

IBM-Data-Science-Professional-Certification/data wrangling.ipynb at main · Ferhat-unal/IBM-Data-Science-Professional-Certification (github.com)

EDA with SQL

- **SQL Queries Performed**

→ List of Launch Site Names (Task 1)

→ First five Launch sites beginning with ‘CCA’ (Task 2)

→ Total and Average Payload Mass for Customers and Booster Versions (Tasks 3 and 4)

→ Year first successful landing outcome pad was achieved (Task 5)

- → List of Booster Names with successful drone ship and payload mass of 4000 to 6000 kg (Task 6)

→ Total number of successful and unsuccessful mission outcomes (Task 7)

→ Booster versions carried max payload mass (Task 8)

→ Month, Failure Landing Outcomes, Booster Versions, and Launch Site for Drone Ship (Task 9)

→ Count and rank of successful landing outcomes (Task 10)

GitHub URL

IBM-Data-Science-Professional-Certification/EDA with SQL lab.ipynb at main · Ferhat-unal/IBM-Data-Science-Professional-Certification (github.com)

EDA with Data Visualization

Variables used for plots:

➢ → Flight Number, Payload Mass, Launch Site, Orbit, Class, Year

Plots created:

➢ → **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**

➢ → **Success Yearly Trend**

Plot types

→ Scatter plots

→ Line charts

→ Bar plots

Used to compare relationships between variable to decide if the relationships between each variable exist so that the variables are used for training the machine learning models used for predictive analytics

GitHub URL

BM-Data-Science-Professional-Certification/module_2_Data visualization.ipynb at main · Ferhat-unal/IBM-Data-Science-Professional-Certification (github.com)

Build an Interactive Map with Folium

- Map objects in Folium such as markers, circles, and lines locates the launch sites, highlights successful and unsuccessful landings, and proximity to the following key locations: Railway, Coastline, City, Highway
- Purpose of having those map objects helps us understand why the launch sites are located in those specific locations and visualizes all successful landings relative to key locations

GitHub URL

IBM-Data-Science-Professional-Certification/module_3_lab_jupyter_launch_site_location.jupyterlite.ipynb at main · Ferhat-unal/IBM-Data-Science-Professional-Certification (github.com)

Build a Dashboard with Plotly Dash

Dashboard includes a pie chart, range slider, and a scatter plot

Pie chart shows the distribution of successful landings across all launch sites and can be selected to visualize individual launch site success rates

Two inputs: All sites or individual launch sites

Scatter plot shows how success varies across launch sites, payload mass, and booster version category

Two inputs: All sites or individual sites and payload mass on interactive slider between 0 and 10,000 kg

GitHub URL

IBM-Data-Science-Professional-Certification/spacex_dash_app.py at main · Ferhat-unal/IBM-Data-Science-Professional-Certification (github.com)

Predictive Analysis(Classification)

- Split label column 'Class' from dataset
- Fit and Transform features using StandardScalar()
- Train test split data
- GridSearchCV (cv=10) to find optimal parameters
- GridSearcHCV on LogReg, SVM, Decision Tree and KNN models
- Score models on test dataset
- Plot confusion matrix for all models
- Plot accuracy scores for all models

GitHub URL

IBM-Data-Science-Professional-Certification/IBM-DS0321EN-SkillsNetwork_labs_module_4_SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb at main · Ferhat-unal/IBM-Data-Science-Professional-Certification (github.com)

Results

- Exploratory data analysis results
  - Orbits ES-L1, GEO, HEO, and SSO have the highest average success rates
  - There are no rockets launched for heavy payload mass (greater than 10,000 kg) for the VAFB-SLC launch site
- Interactive analytics demo in screenshots
  - KSC LC-39A launch site has highest success rate of 41.7%
  - FT Booster has the most successful landings (13 of 15 with payload mass range of 2,000 to 5,500 kg
- Predictive analysis results
  - Three models: Logistic Regression, Decision Tree, and Support Vector Machines have test accuracy rate of 83.33%
  - K Nearest Neighbors had test accuracy rate of 77.78%

Section 2

insight drawn from EDA

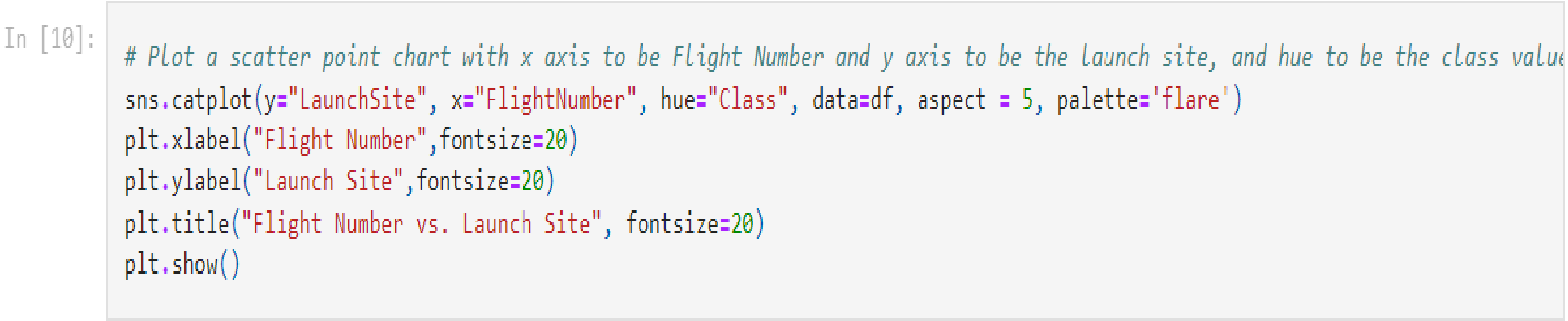
Flight Number vs. Launch Site

Purple indicates successful launches, pink indicates unsuccessful launces

Plot suggests an increase in success rate over time

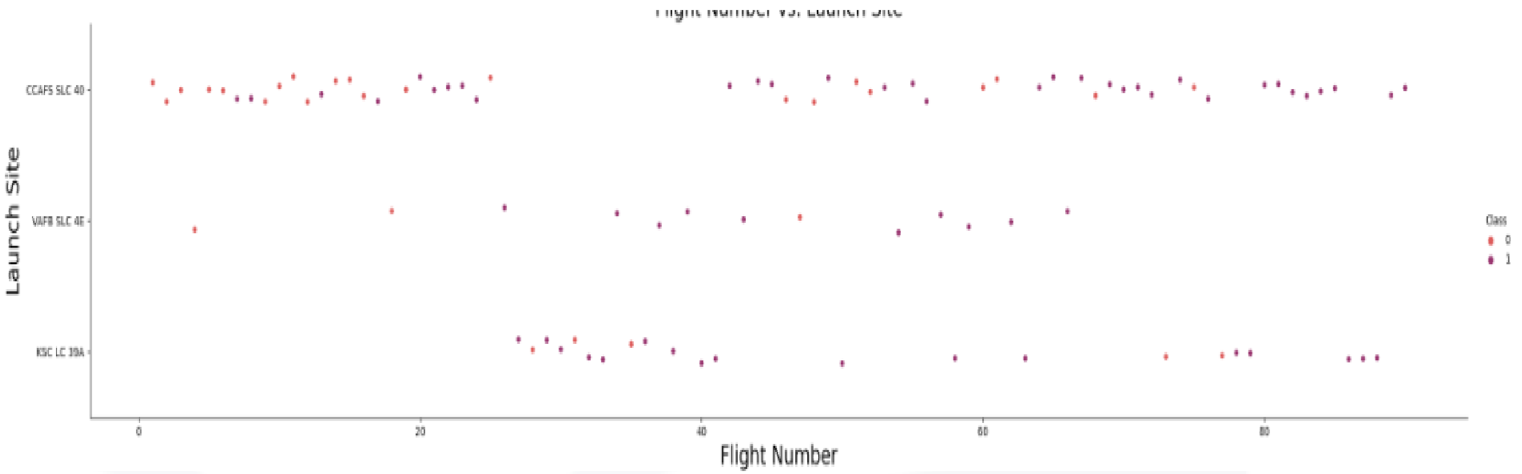
Possible breakthrough around Flight Number 20 where there's an increase in successful launces

CCAFS SLC 40 has the most launches and has the most volume of launches



In [10]:

```
# Plot a scatter point chart with x axis to be Flight Number and y axis to be the launch site, and hue to be the class value
sns.catplot(y="LaunchSite", x="FlightNumber", hue="Class", data=df, aspect = 5, palette='flare')
plt.xlabel("Flight Number",fontsize=20)
plt.ylabel("Launch Site",fontsize=20)
plt.title("Flight Number vs. Launch Site", fontsize=20)
plt.show()
```



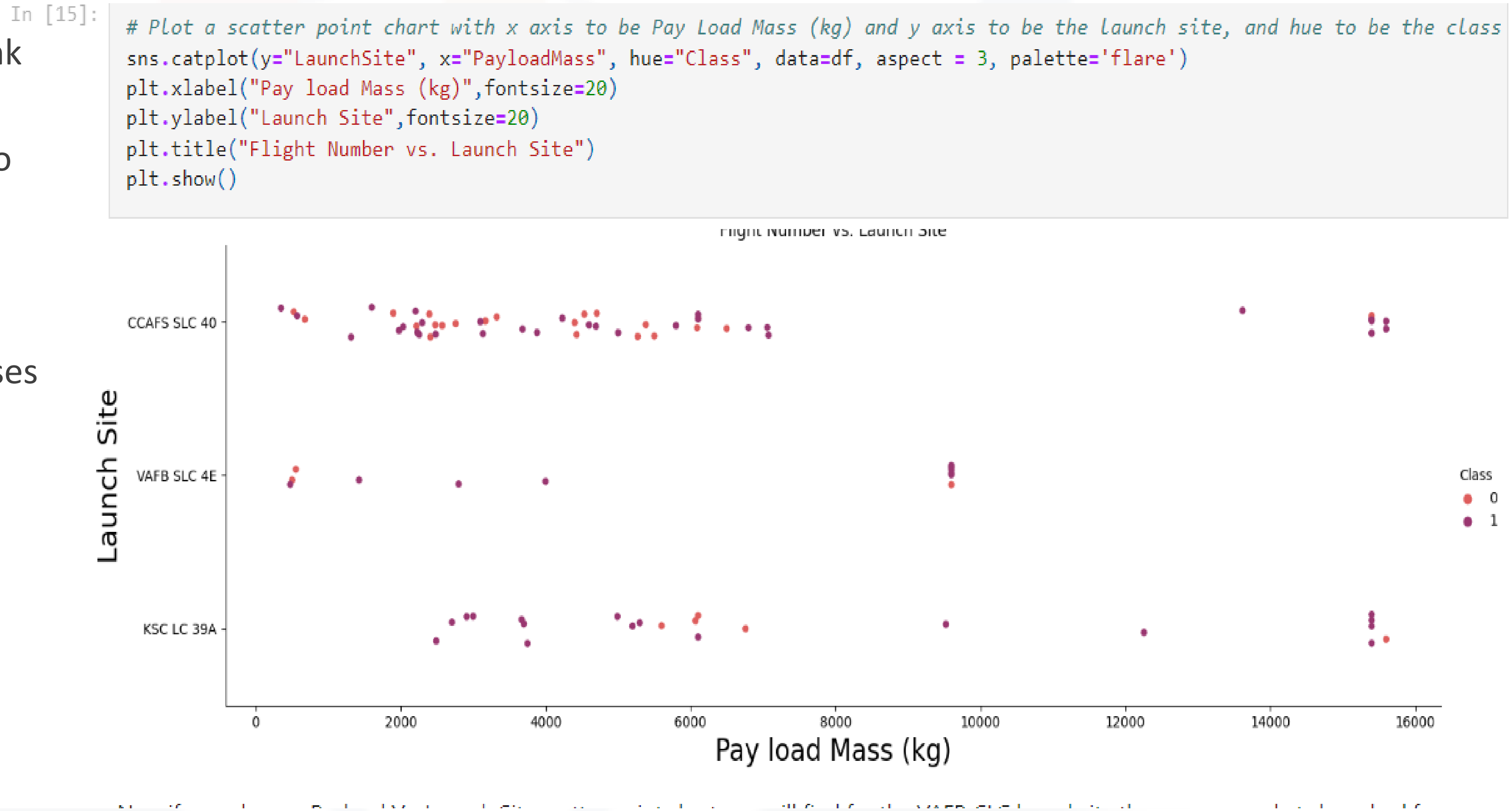
Payload vs. Launch Site

Purple indicates successful launches, pink indicates unsuccessful launces

Payload mass mostly appears from   0 to 7,500 kg

Each of the launch sites have different payload masses

VAFB SLC 4E doesn’t have payload masses greater than 10,000 kg



In [15]:

```
# Plot a scatter point chart with x axis to be Pay Load Mass (kg) and y axis to be the launch site, and hue to be the class
sns.catplot(y="LaunchSite", x="PayloadMass", hue="Class", data=df, aspect = 3, palette='flare')
plt.xlabel("Pay load Mass (kg)",fontsize=20)
plt.ylabel("Launch Site",fontsize=20)
plt.title("Flight Number vs. Launch Site")
plt.show()
```

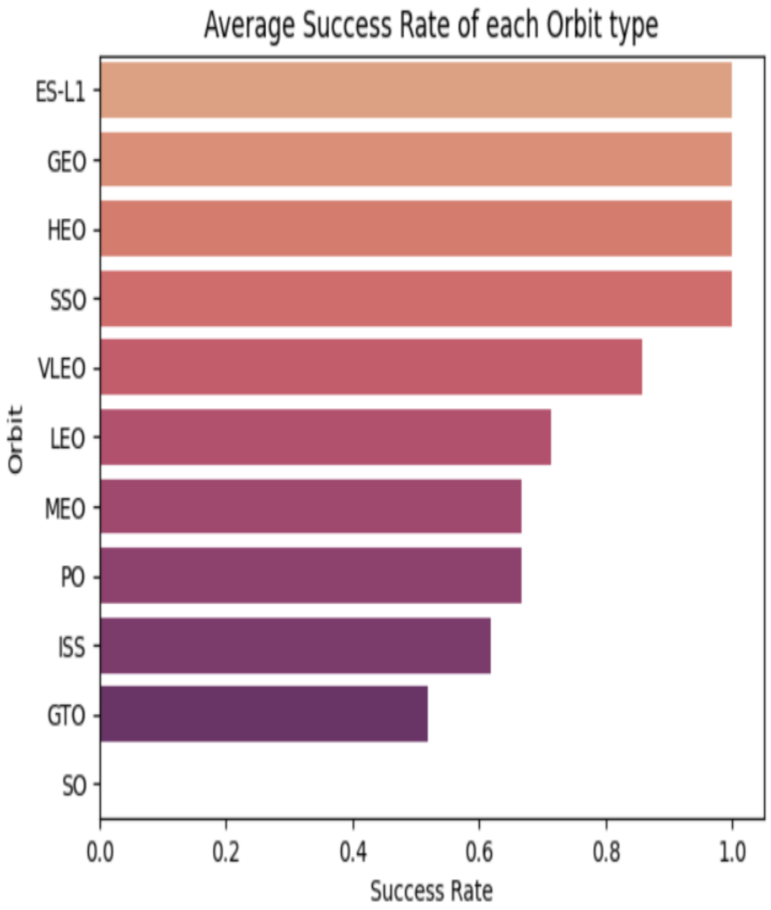
Success Rate vs. Orbit Type

ES-L1 (1), GEO (1), HEO(1), and SSO (5) all have success rate of 1.0 (100%).

MEO (3) and PO (9) have success rate of 0.67 (67%)

GTO (27) has success rate of 0.5 (50%)

SO (1) had a success rate of 0 (0%)

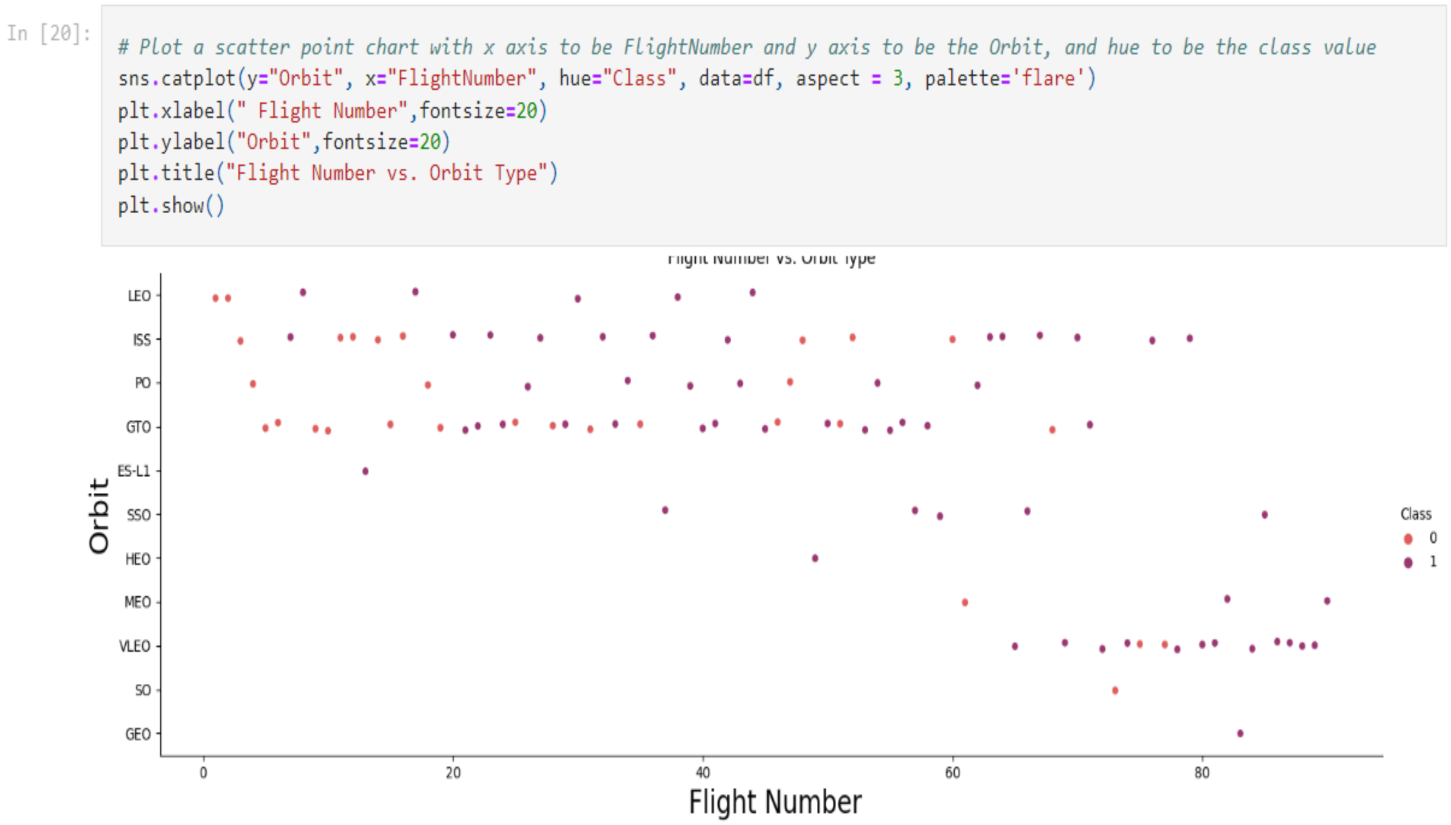


Flight Number vs. Orbit Type

Purple indicates successful launches, pink indicates unsuccessful launces

There seems to be more successful launches for Fligh numbers greater than 60

X appears to have successful launches in lower orbits or Sun-synchronous orbits



```
In [20]:
# Plot a scatter point chart with x axis to be FlightNumber and y axis to be the Orbit, and hue to be the class value
sns.catplot(y="Orbit", x="FlightNumber", hue="Class", data=df, aspect = 3, palette='flare')
plt.xlabel(" Flight Number",fontsize=20)
plt.ylabel("Orbit",fontsize=20)
plt.title("Flight Number vs. Orbit Type")
plt.show()
```

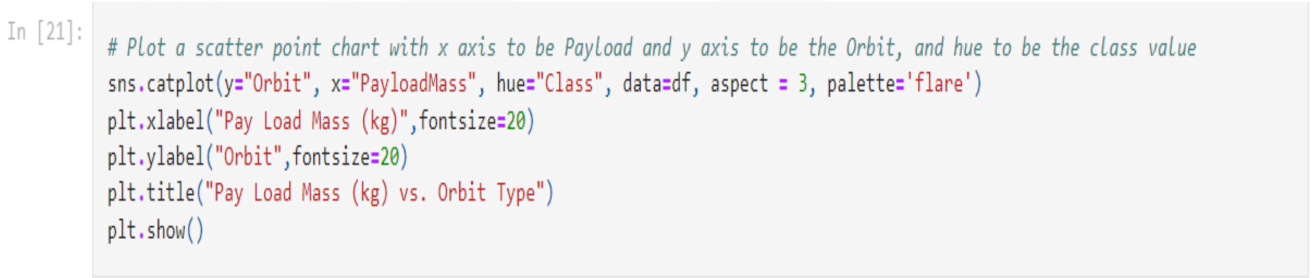
Payload vs. Orbit Type

Purple indicates successful launches, pink indicates unsuccessful launces

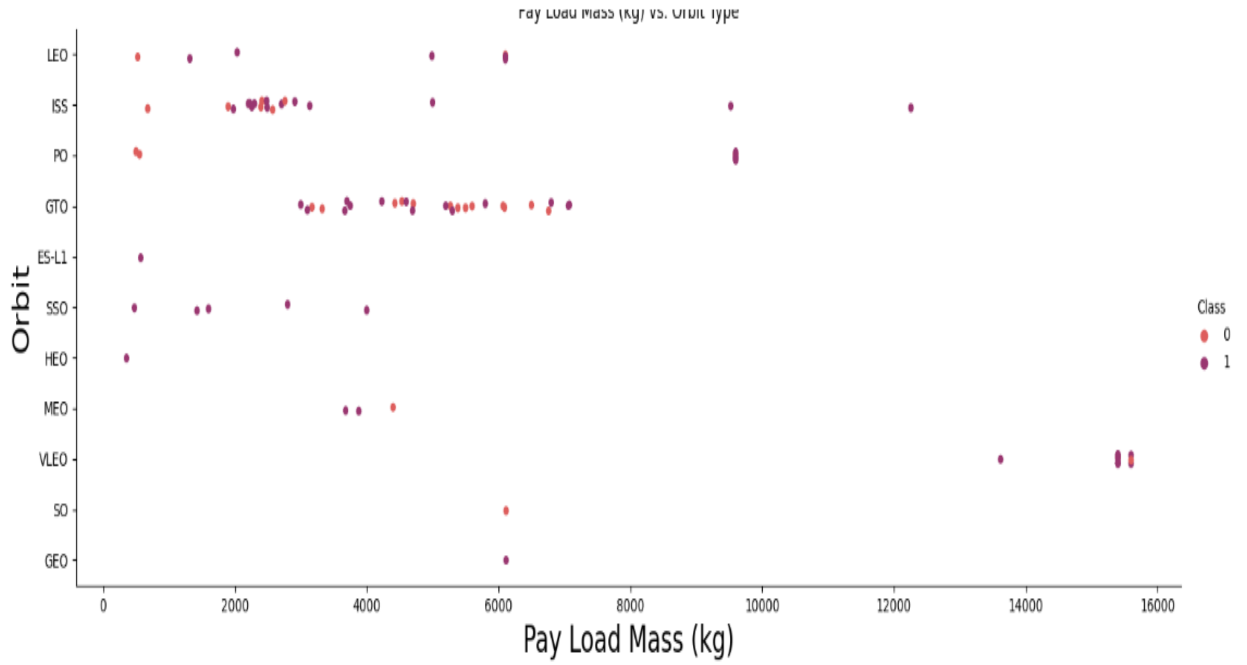
LEO and SSO appear to have low payload masses

VLEO only has playload masses at the higher range

GTO has a higher concentration of launches with payload masses between 2,500 to 7,500 kg



In [21]:
```
# Plot a scatter point chart with x axis to be Payload and y axis to be the Orbit, and hue to be the class value
sns.catplot(y="Orbit", x="PayloadMass", hue="Class", data=df, aspect = 3, palette='flare')
plt.xlabel("Pay Load Mass (kg)",fontsize=20)
plt.ylabel("Orbit",fontsize=20)
plt.title("Pay Load Mass (kg) vs. Orbit Type")
plt.show()
```

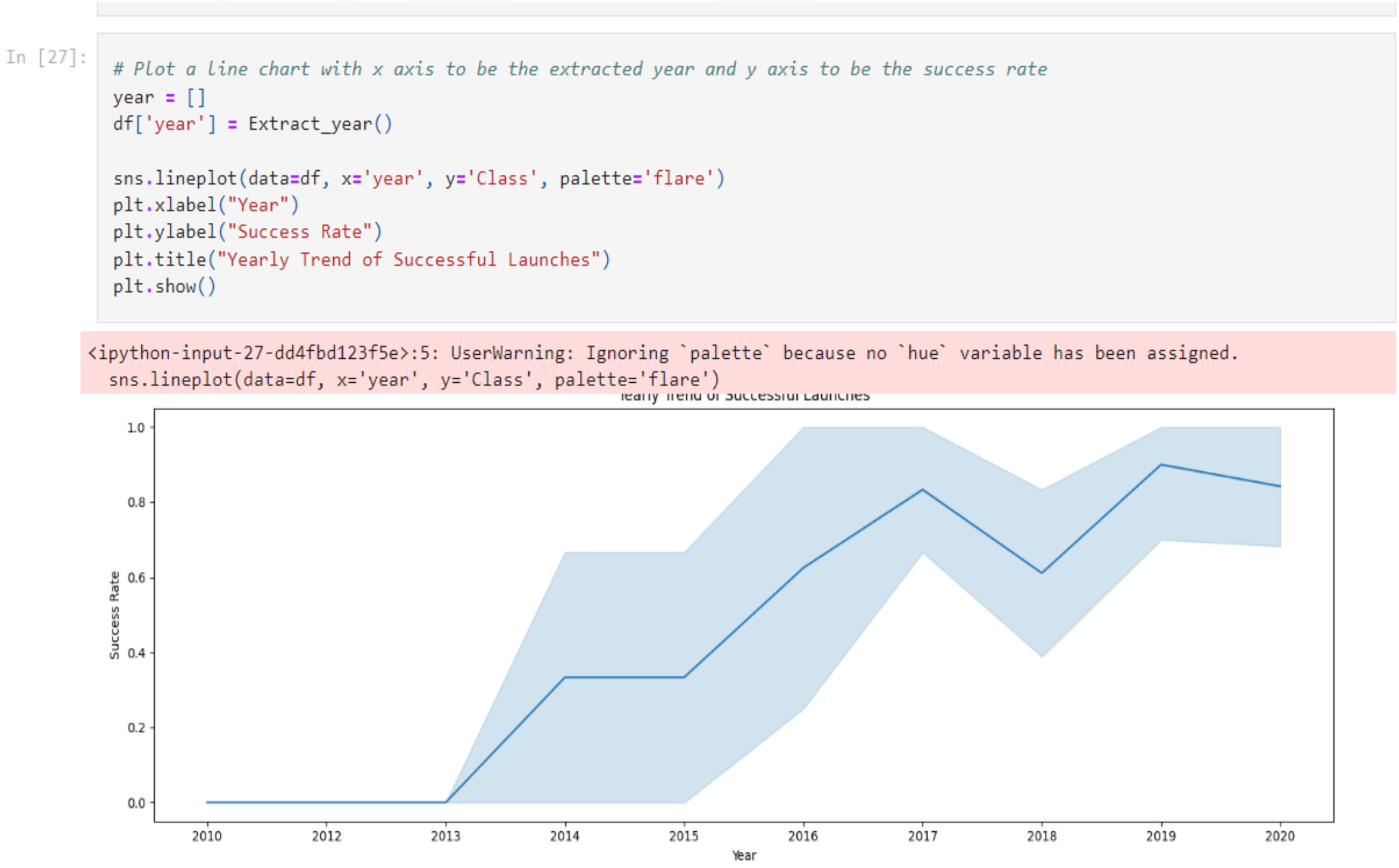


Launch Success Yearly Trend

Light blue shading indicates 95% confidence interval

Number of successful launches increases over time starting in year 2013, with slight dip in 2018

Most recent years has success rates around 80%

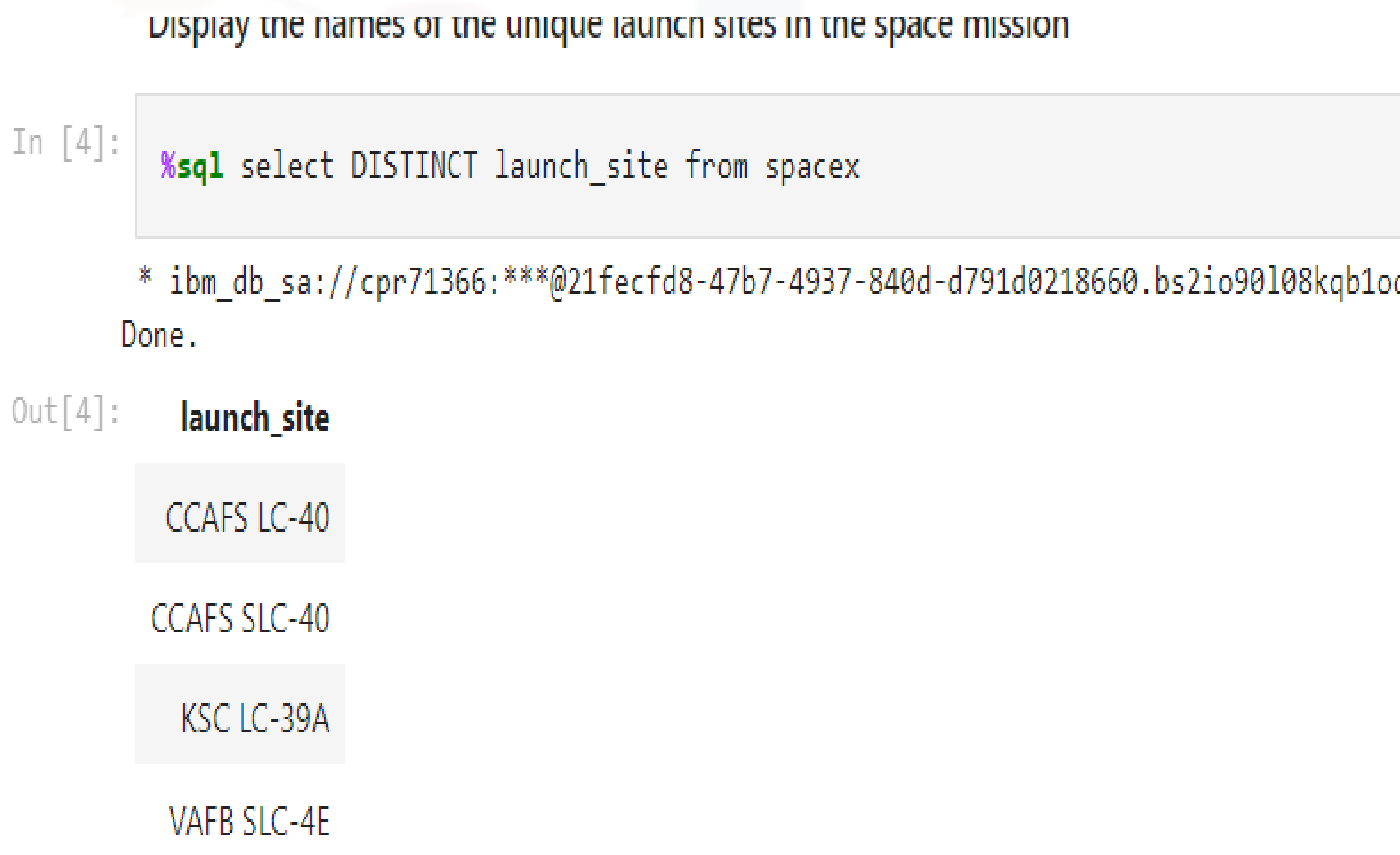


```
In [27]:
# Plot a line chart with x axis to be the extracted year and y axis to be the success rate
year = []
df['year'] = Extract_year()

sns.lineplot(data=df, x='year', y='Class', palette='flare')
plt.xlabel("Year")
plt.ylabel("Success Rate")
plt.title("Yearly Trend of Successful Launches")
plt.show()
```

```
<ipython-input-27-dd4fbd123f5e>:5: UserWarning: Ignoring `palette` because no `hue` variable has been assigned.
  sns.lineplot(data=df, x='year', y='Class', palette='flare')
```

All Launch Site Names

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



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

In [4]:

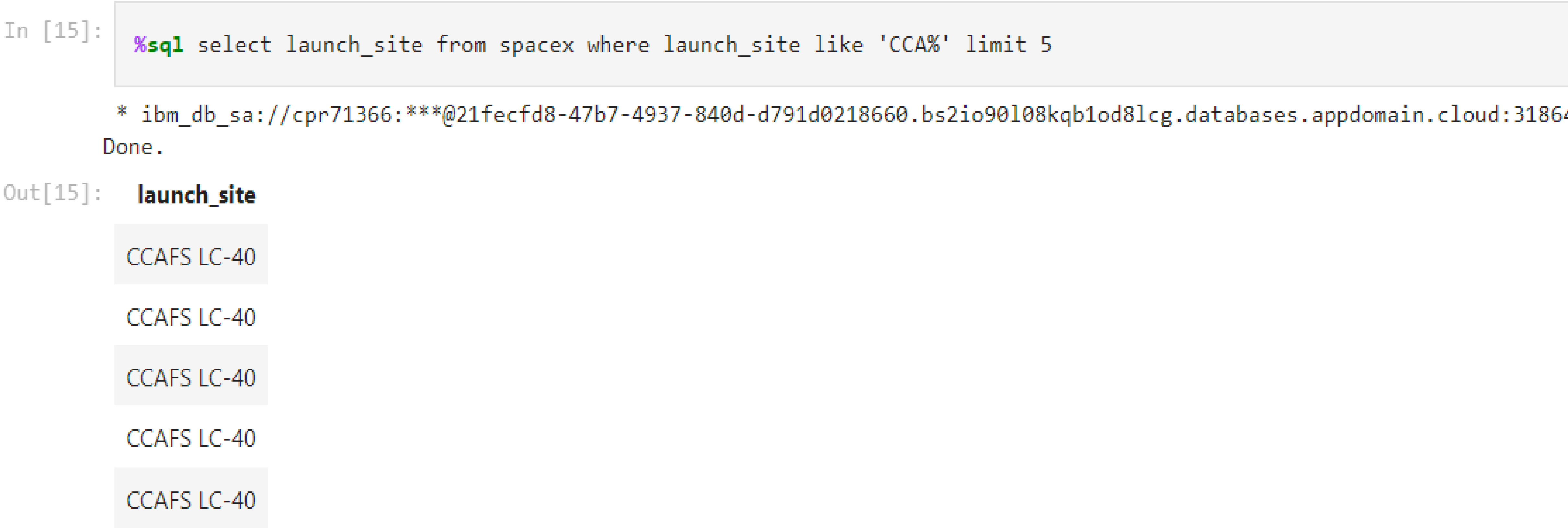
```
%sql select DISTINCT launch_site from spacex
```

```
 * ibm_db_sa://cpr71366:***@21fecfd8-47b7-4937-840d-d791d0218660.bs2io90l08kqb1od
Done.
```

Out[4]:

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

Launch Site Names Begin with 'CCA'



In [15]:

```
%sql select launch_site from spacex where launch_site like 'CCA%' limit 5
```

```
 * ibm_db_sa://cpr71366:***@21fecfd8-47b7-4937-840d-d791d0218660.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:3186
Done.
```

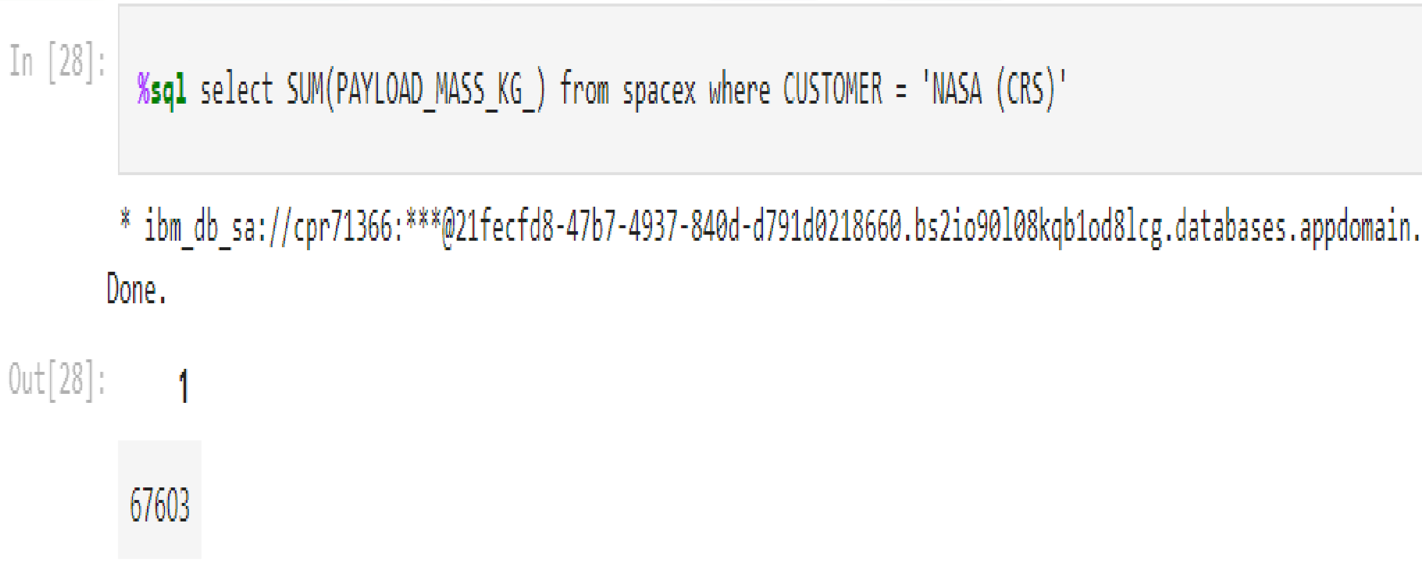
Out[15]:

launch_site
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40

Total Payload Mass

The query below sums all the payload mass (in kg) values where NASA was the customer

CRS strands for Commercial Resupply Services indicating that the payloads were sent to the International Space Station (ISS)

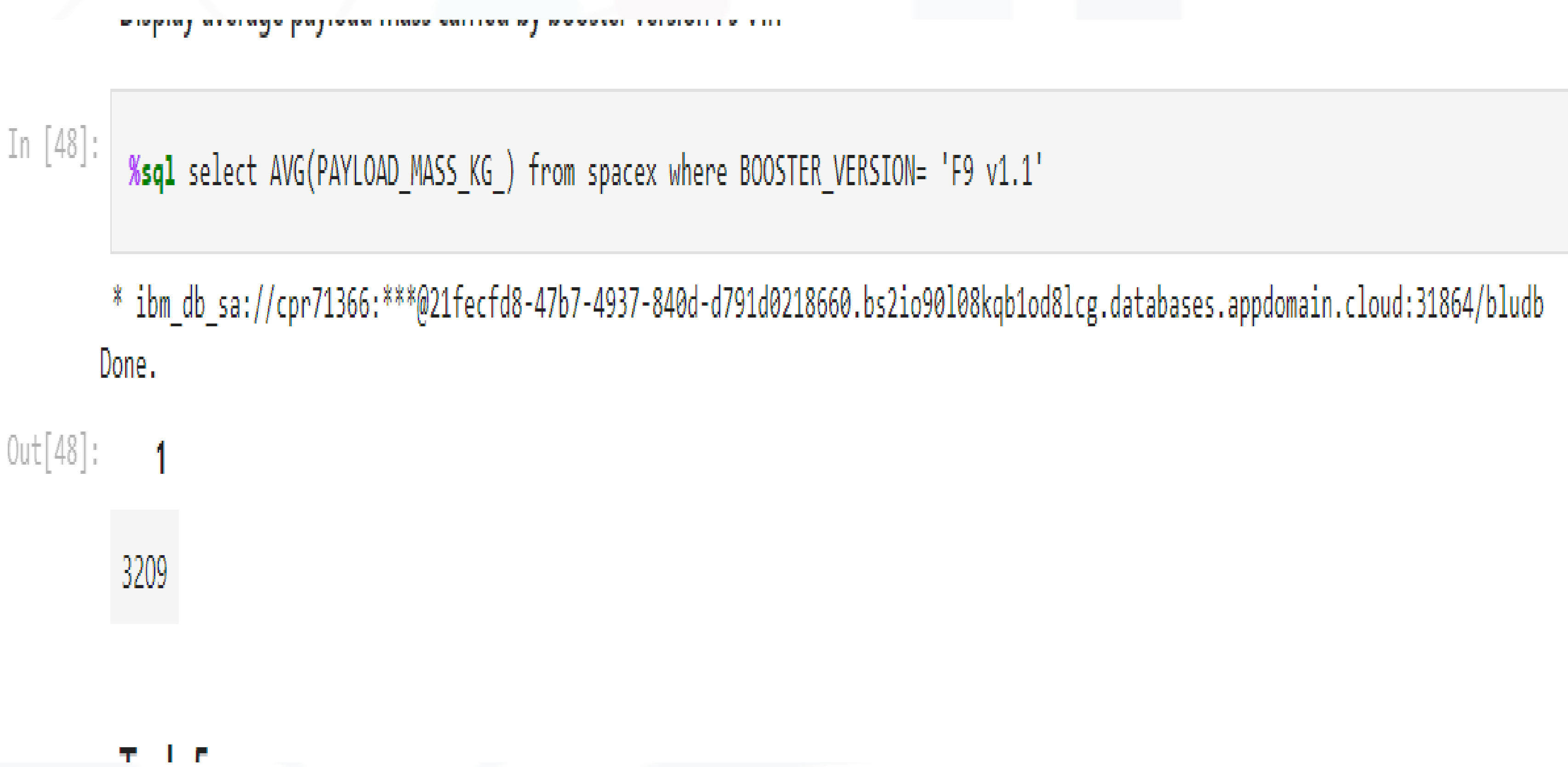


```
In [28]:  %sql select SUM(PAYLOAD_MASS_KG_) from spacex where CUSTOMER = 'NASA (CRS)'

 * ibm_db_sa://cpr71366:***@21fecfd8-47b7-4937-840d-d791d0218660.bs2io90l08kqb1od8lcg.databases.appdomain.
Done.

Out[28]:   1
         67603
```

Average Payload Mass by F9 v1.1

The query to the right calculates the average payload mass carried by booster version F9 v1.1

Average payload mass of F9 v1.1 is on the low end of the payload mass range



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

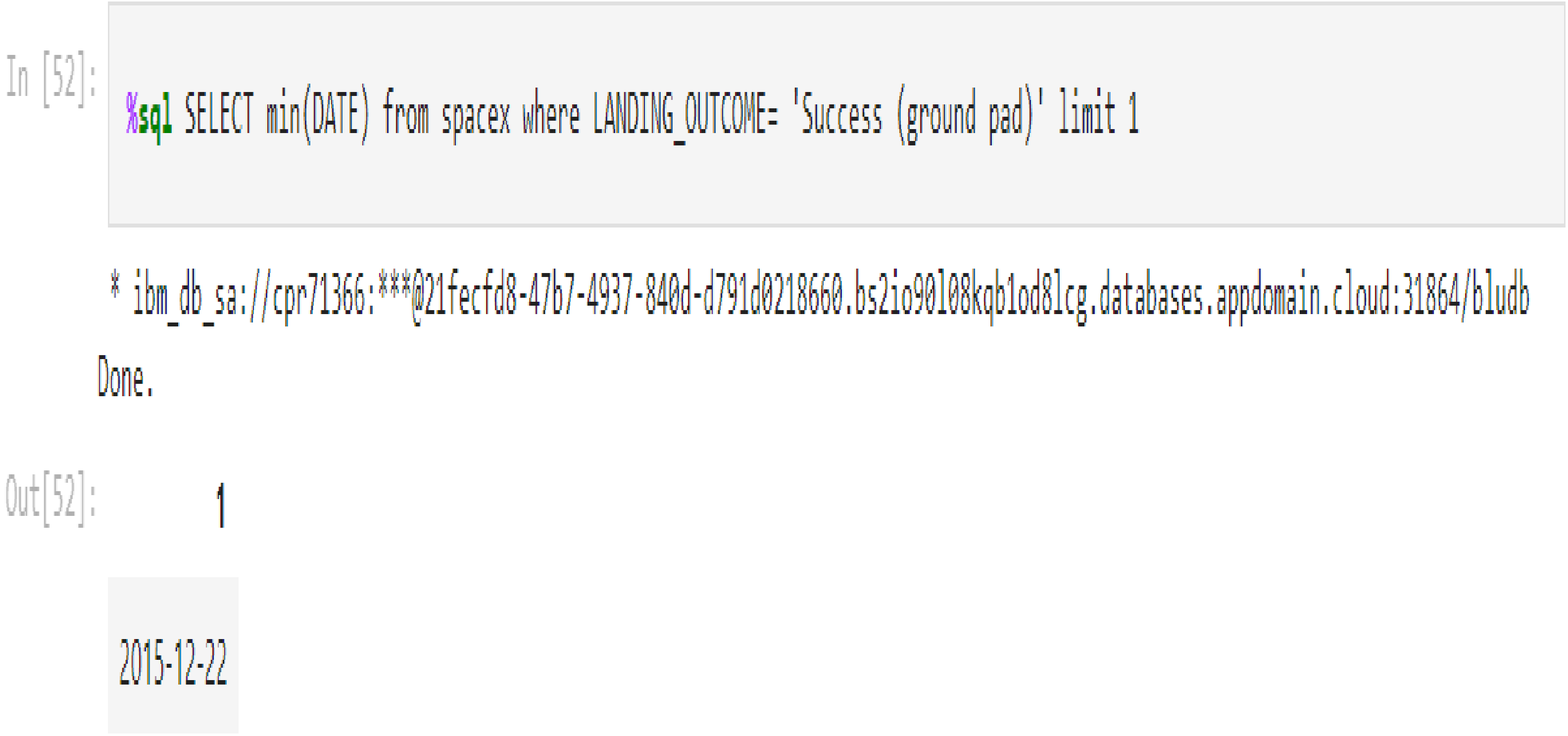
```
In [48]: %sql select AVG(PAYLOAD_MASS_KG_) from spacex where BOOSTER_VERSION= 'F9 v1.1'

 * ibm_db_sa://cpr71366:***@21fecfd8-47b7-4937-840d-d791d0218660.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31864/bludb
Done.

Out[48]:  1
       3209
```

First Successful Ground Landing Date

The query returns the first successful ground pad landing date

First grounding pad launch didn’t appear until the end of 2015

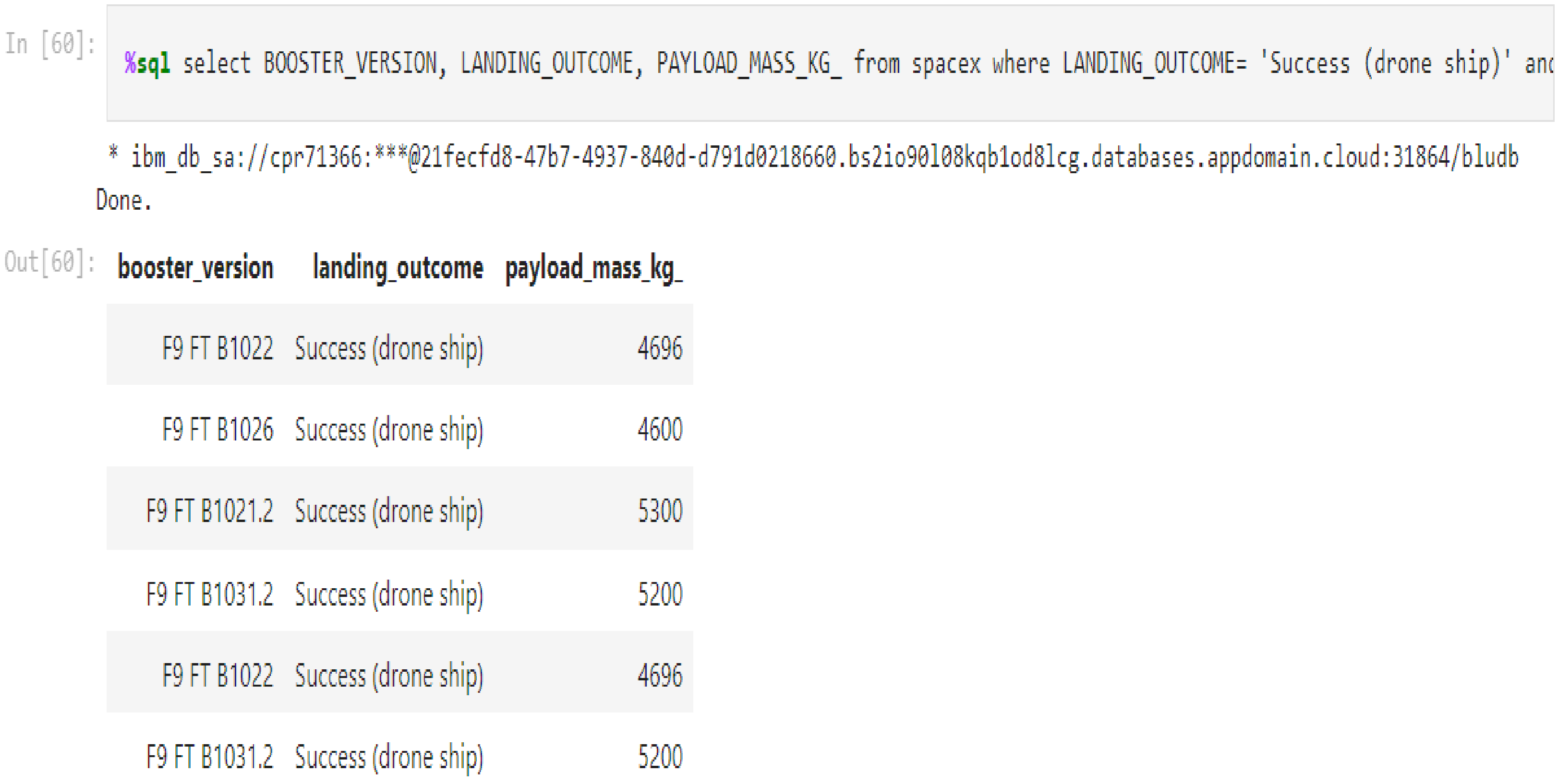
Successful launches didn’t appear until 2014



```
In [52]: %sql SELECT min(DATE) from spacex where LANDING_OUTCOME= 'Success (ground pad)' limit 1

 * ibm_db_sa://cpr71366:***@21fecfd8-47b7-4937-840d-d791d0218660.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31864/bludb
Done.

Out[52]:      1
         2015-12-22
```

Successful Drone Ship Landing with Payload between 4000 and 6000

This query returns the four booster versions that had successful drone ship landings and a payload mass between 4000 and 6000 kg non-inclusively



```
In [60]:  %sql select BOOSTER_VERSION, LANDING_OUTCOME, PAYLOAD_MASS_KG_ from spacex where LANDING_OUTCOME= 'Success (drone ship)' and

 * ibm_db_sa://cpr71366:***@21fecfd8-47b7-4937-840d-d791d0218660.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31864/bludb
Done.
```

Out[60]:

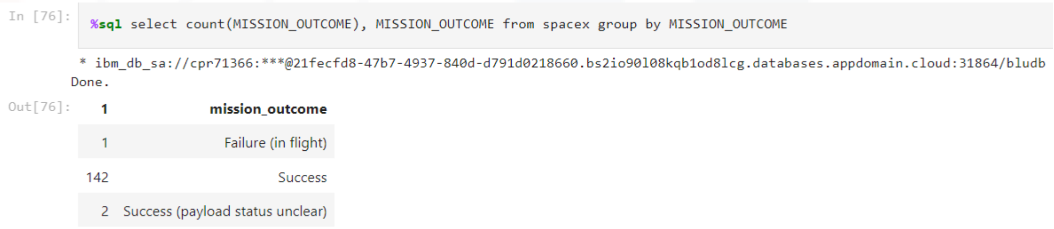
booster_version	landing_outcome	payload_mass_kg_
F9 FT B1022	Success (drone ship)	4696
F9 FT B1026	Success (drone ship)	4600
F9 FT B1021.2	Success (drone ship)	5300
F9 FT B1031.2	Success (drone ship)	5200
F9 FT B1022	Success (drone ship)	4696
F9 FT B1031.2	Success (drone ship)	5200

Total Number of Successful and Failure Mission Outcomes

This query calculates the total number of successful and failure mission outcomes

There’s 2 success with a payload status unclear , but in total there should be a total of 144 Success Missions

Only one launch failed in flight



```
In [76]: %sql select count(MISSION_OUTCOME), MISSION_OUTCOME from spacex group by MISSION_OUTCOME

 * ibm_db_sa://cpr71366:***@21fecfd8-47b7-4937-840d-d791d0218660.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31864/bludb
Done.
Out[76]:
```

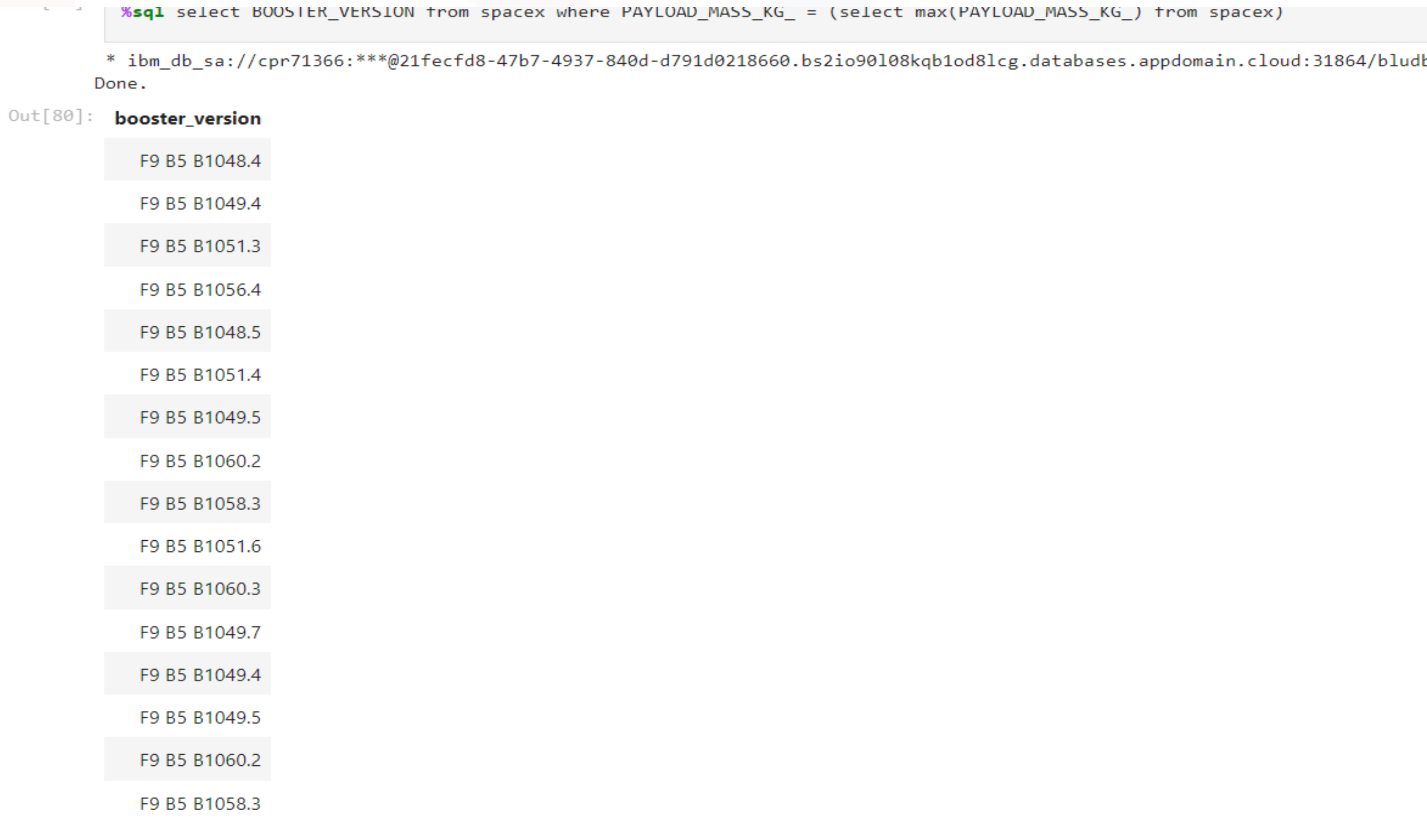
1	mission_outcome
1	Failure (in flight)
142	Success
2	Success (payload status unclear)

Boosters Carried Maximum Payload

The query returns the booster versions that carry a maximum payload mass of 15,600 kg

All boosters fall in the “F9 B5 B10xx.x” variety

There appears that the payload mass correlates with the booster version used.



```
%sql select BOOSTER_VERSION from spacex where PAYLOAD_MASS_KG_ = (select max(PAYLOAD_MASS_KG_) from spacex)
```

```
 * ibm_db_sa://cpr71366:***@21fecfd8-47b7-4937-840d-d791d0218660.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31864/bludb
Done.
```

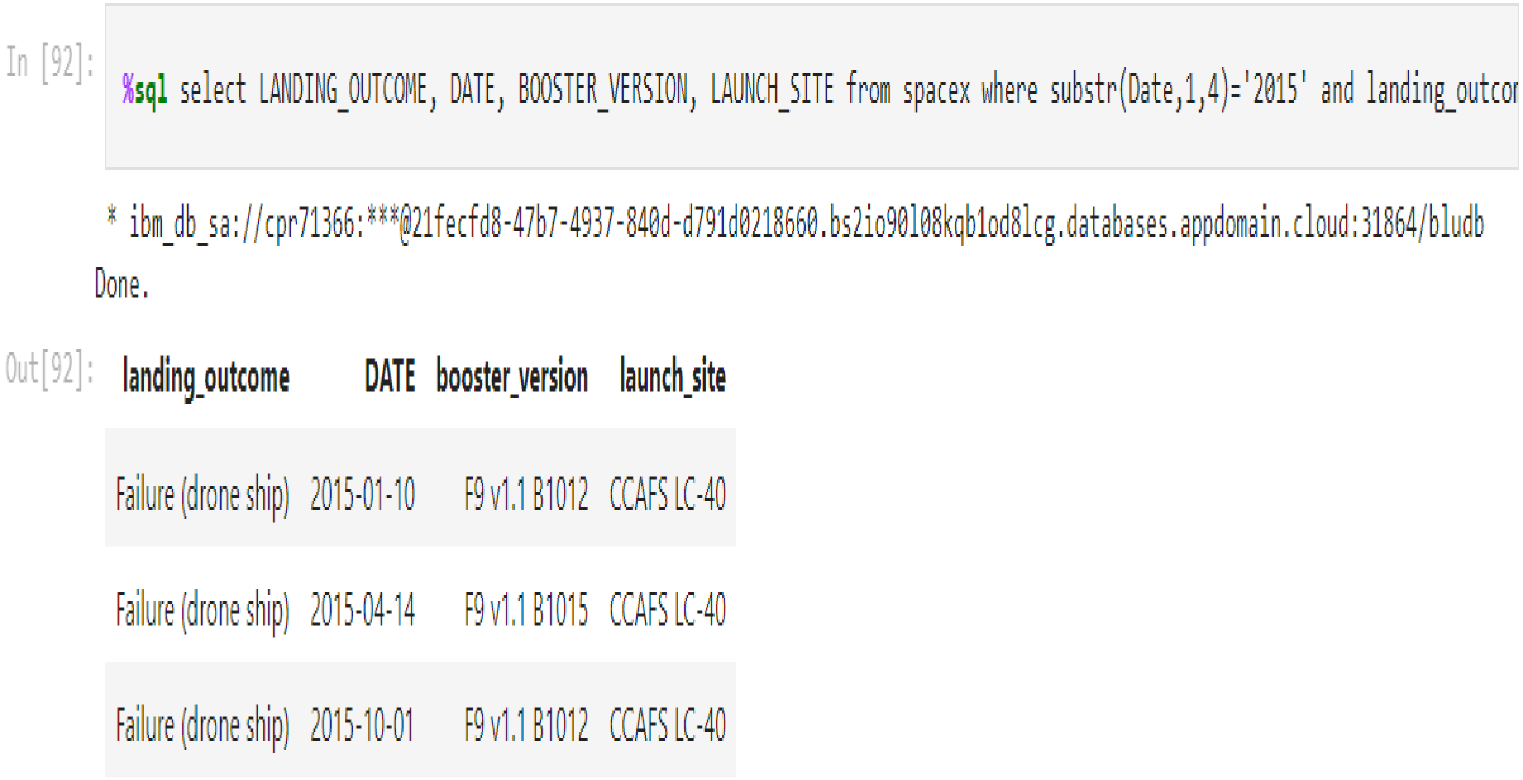
Out[80]:

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
F9 B5 B1049.4
F9 B5 B1049.5
F9 B5 B1060.2
F9 B5 B1058.3

2015 Launch Records

This query returns the Month number, landing outcome, booster version, and launch site for the 2015 launches where Stage One failed to land on a drone ship

Two of those occurrences were at the same launch site



In [92]:

```
%sql select LANDING_OUTCOME, DATE, BOOSTER_VERSION, LAUNCH_SITE from spacex where substr(Date,1,4)='2015' and landing_outcor
```

```
 * ibm_db_sa://cpr71366:***@21fecfd8-47b7-4937-840d-d791d0218660.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31864/bludb
Done.
```

Out[92]:

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

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

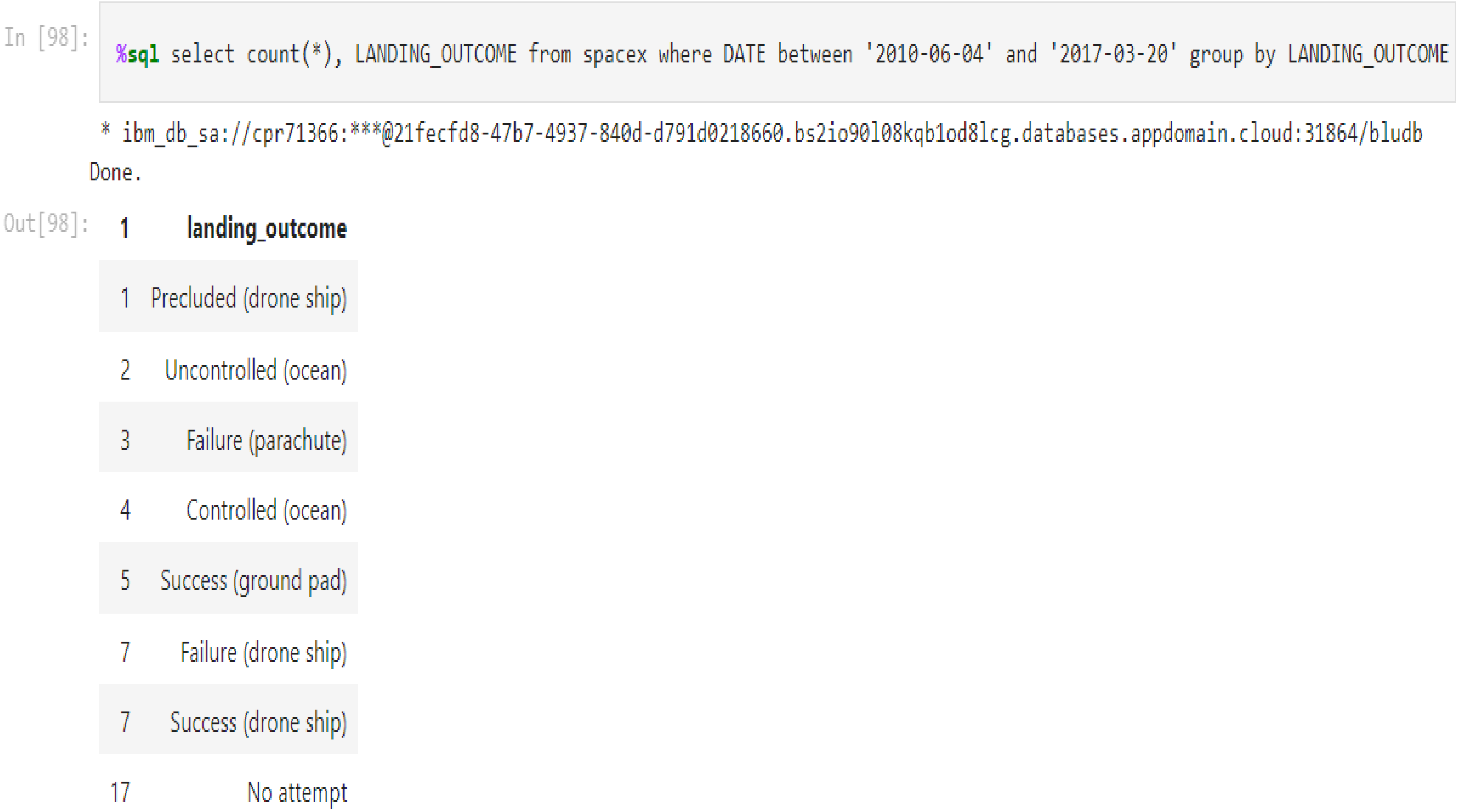
The query returns a list of successful landings between 2010-06-04 and 2017-03-20

Two types of successful landing outcomes

Ground pad

Drone ship

There are a total of 8 successful landings during that time period



```
In [98]: %sql select count(*), LANDING_OUTCOME from spacex where DATE between '2010-06-04' and '2017-03-20' group by LANDING_OUTCOME

 * ibm_db_sa://cpr71366:***@21fecfd8-47b7-4937-840d-d791d0218660.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31864/bludb
Done.
```

Out[98]:

1	landing_outcome
1	Precluded (drone ship)
2	Uncontrolled (ocean)
3	Failure (parachute)
4	Controlled (ocean)
5	Success (ground pad)
7	Failure (drone ship)
7	Success (drone ship)
17	No attempt

Section 3

Launch Site proximities analysis

Location marker on a global map

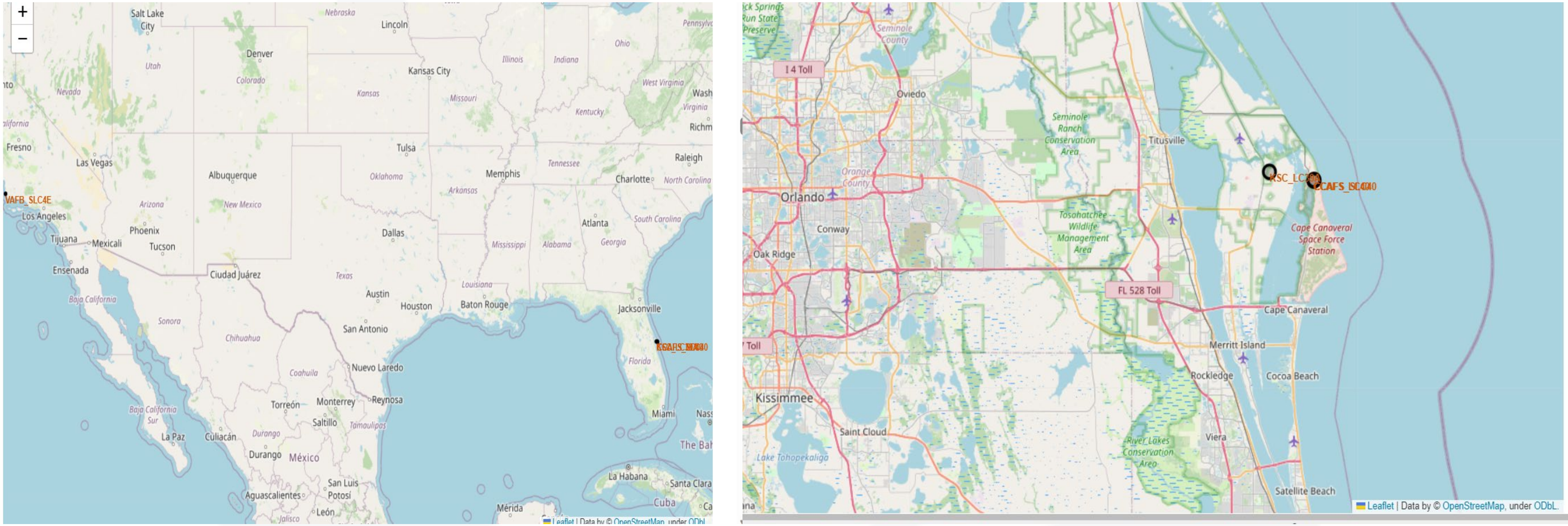
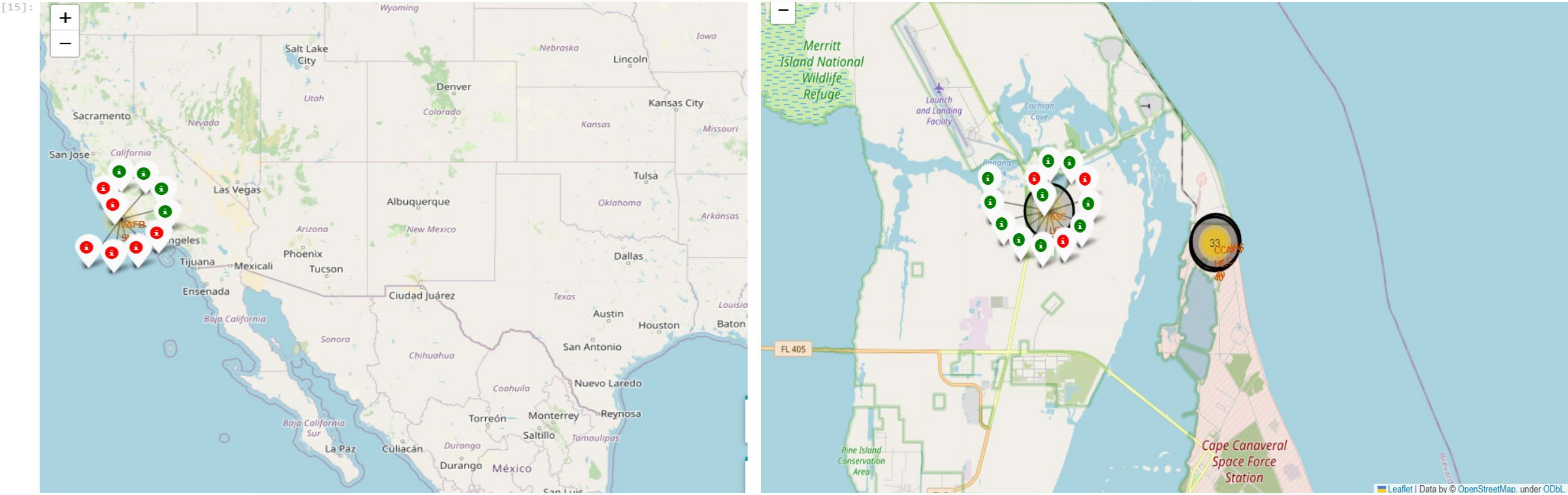


Image on the left shows all the launch site locations on the U.S. map. The image on the right shows the two launch site locations in Florida. All of the launch sites are near the ocean and far from cities.

Color labeled lauch marker

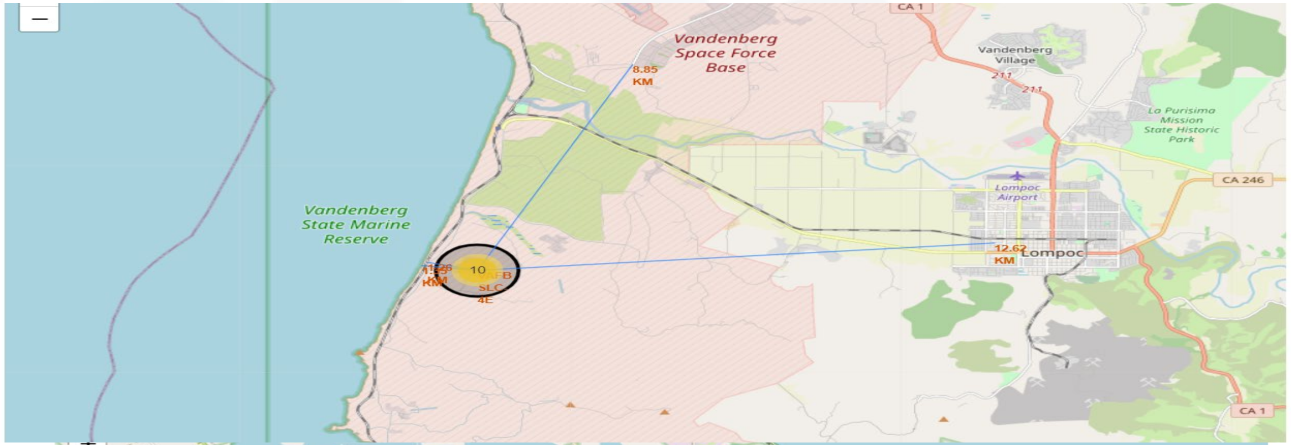


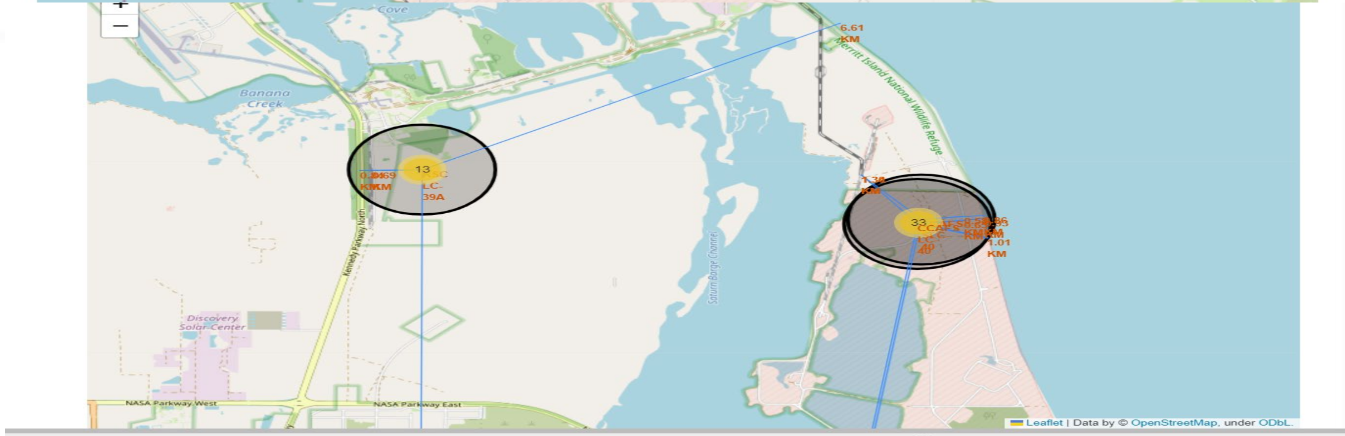
Marker clusters on Folium map are clickable and are displayed as successful landing (green icon) and failed landing (red icon). Image on the left (VAFB-SLC-4E) shows 4 successful and 6 failed landings while image on right (KSC-LC-39A) shows 10 successful and 3 unsuccessful landings.

Proximite to lauch site

- Image on the top shows the distances from CCFAS-SLC-40 launch site to the following key locations: Highway, Railway, and Coastline

  - •Distance to nearest coastline: 0.90 km
  - •Distance to nearest railway: 0.98 km
  - •Distance to nearest highway: 0.58 km


Vandenberg Space Force Base
8.85 KM
Vandenberg Village
La Purisima Mission State Historic Park
Lompoc Airport
CA 246
CA 1
12.62 KM
Lompoc
Vandenberg State Marine Reserve
10
SLC 4E

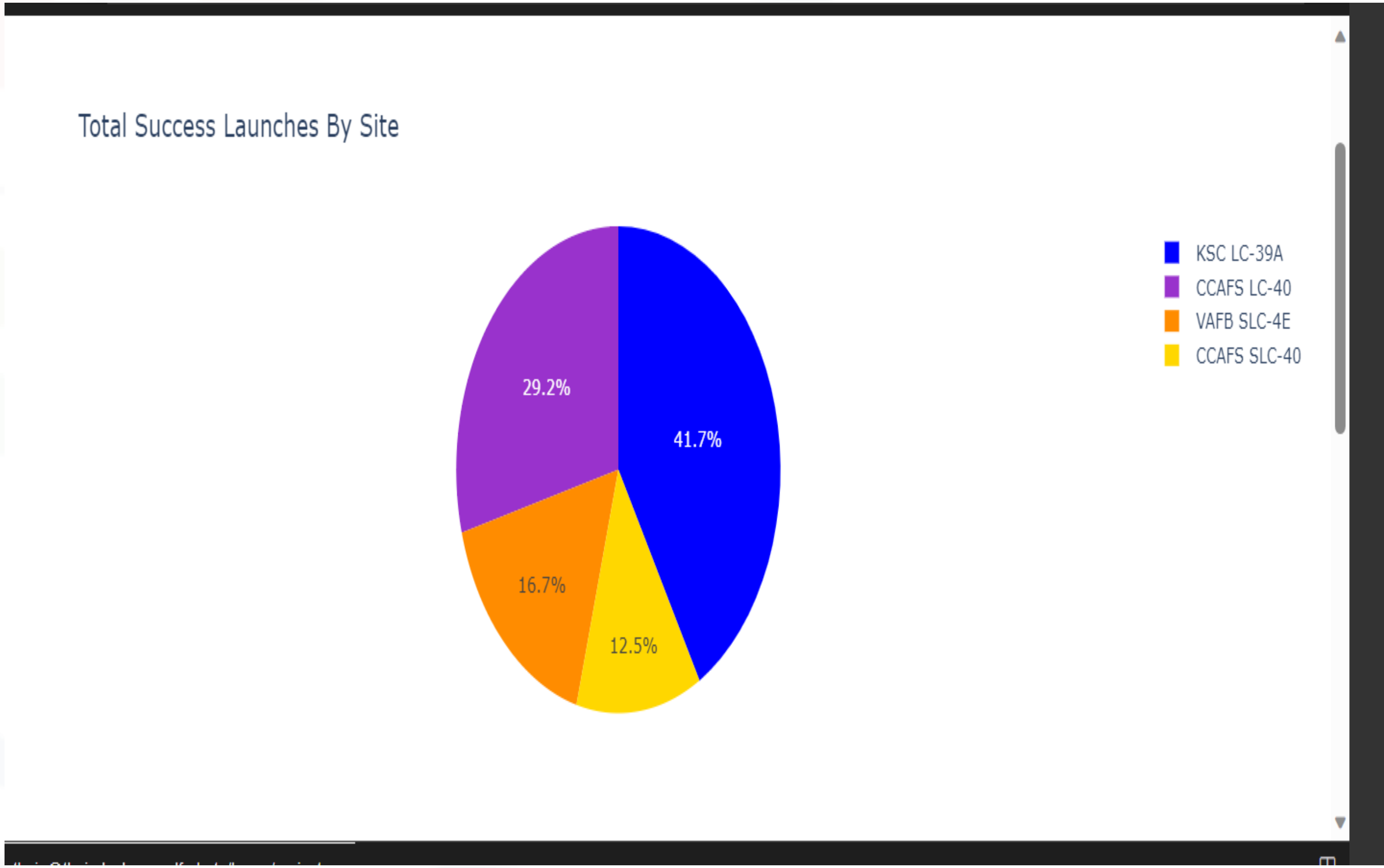


Section 4

Build a Dashboard with Plotly Dash

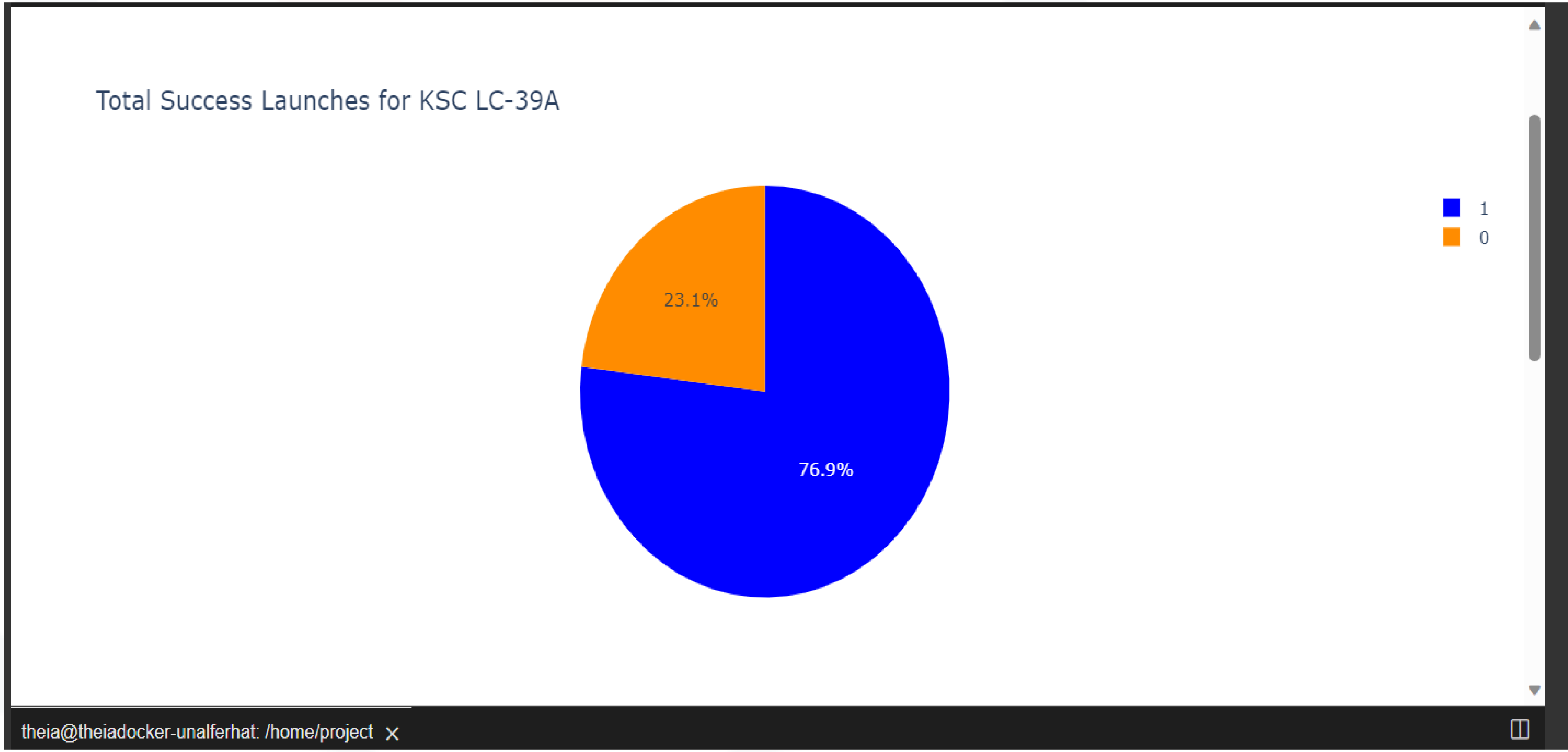
Successful Landings Across Launch Sites

Above is the distribution of successful landings across launch sites. Since CCAFS LC-40 is the older name for CCAFS SLC-40, the number of successful landings for CCAFS SLC-40 (total of 41.7%) and KSC LC-39A (41.7%) have roughly the same amount of successful landings. VAFB SLC-4E has the smallest area of successful landings of 16.7%. This small amount of successful landings in the west coast may be due to smaller sample size and difficulty of launching rockets.



Launch Site with Highest Success Rate

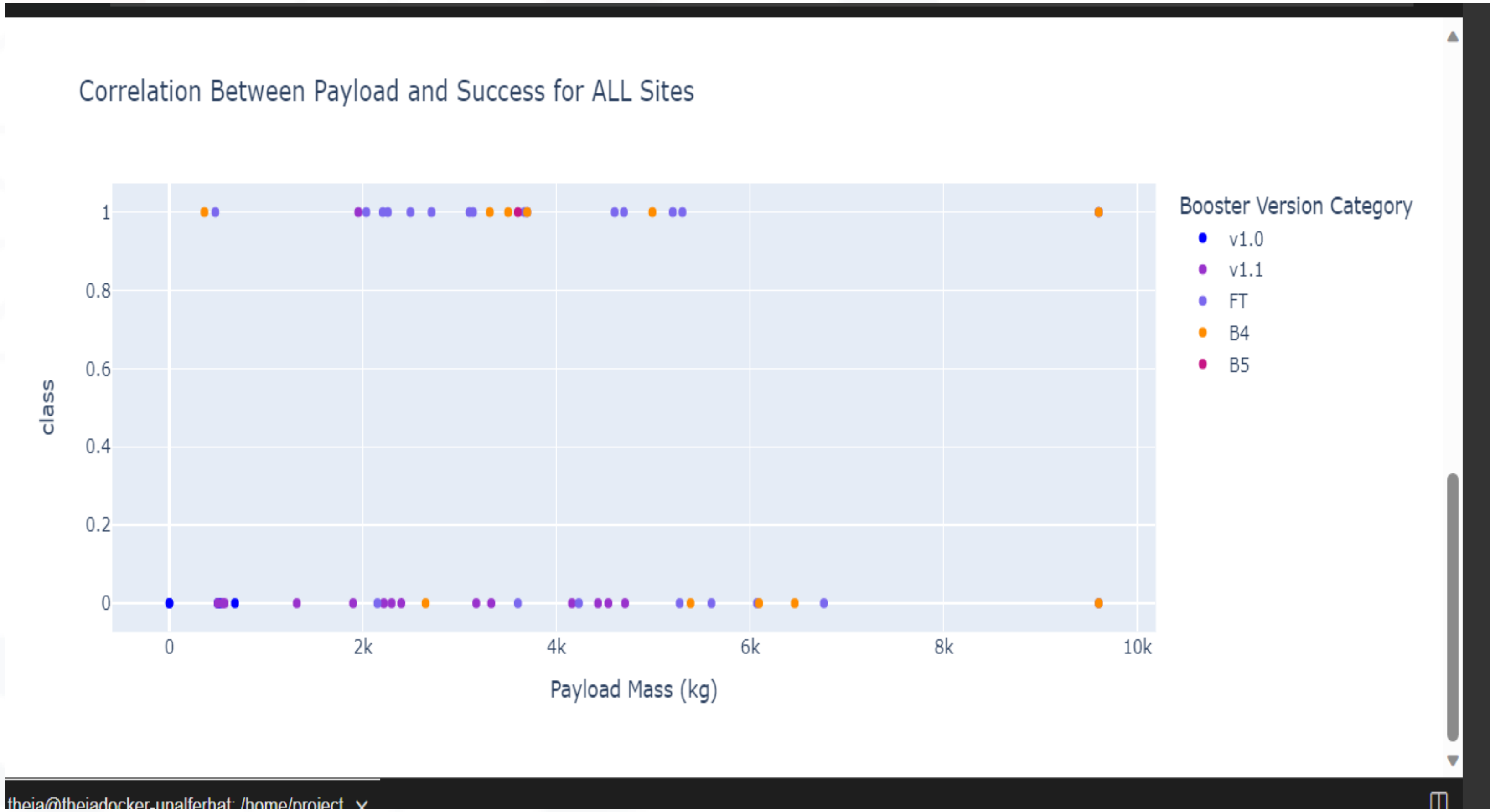
KSC SLC-39A has the highest success rate with 76.9% successful landings and 23.1& failed landings

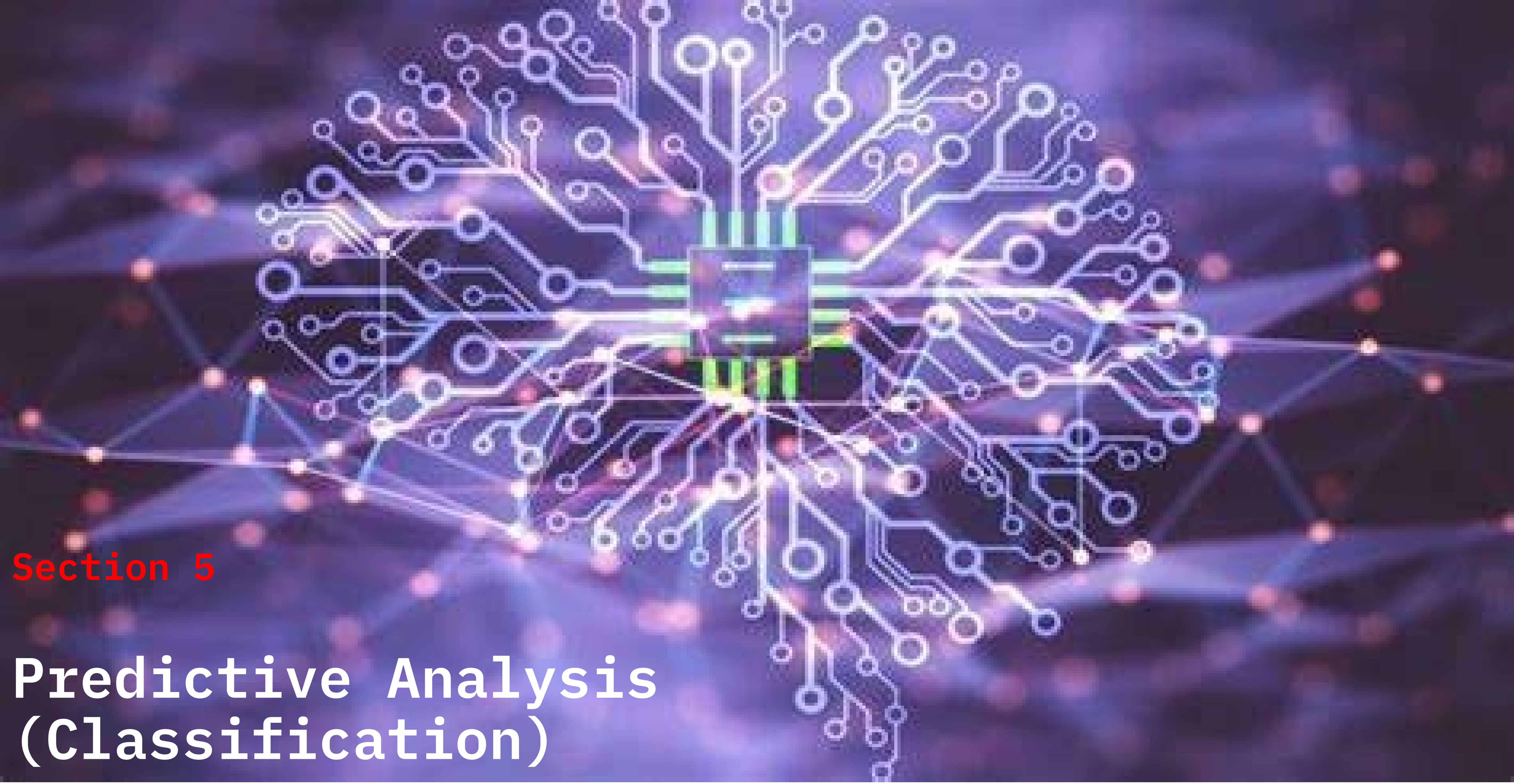


Total Success Launches for KSC LC-39A
23.1%
76.9%
🟦 1
🟧 0
theia@theiadocker-unalferhat: /home/project

Payload vs. Launch Outcome

The Payload range selector is only set from 0 to 10,000 kg, not including payloads greater than 10,000 kg. class 1 indicates successful landing and 0 for failure. The FT booster version accounts for majority of successful landings within the range 0-7,500 kg. The v1.1 booster version accounts for majority of failed landings within the range highlighted below.





Section 5

Predictive Analysis (Classification)

IBM Developer

SKILLS NETWORK

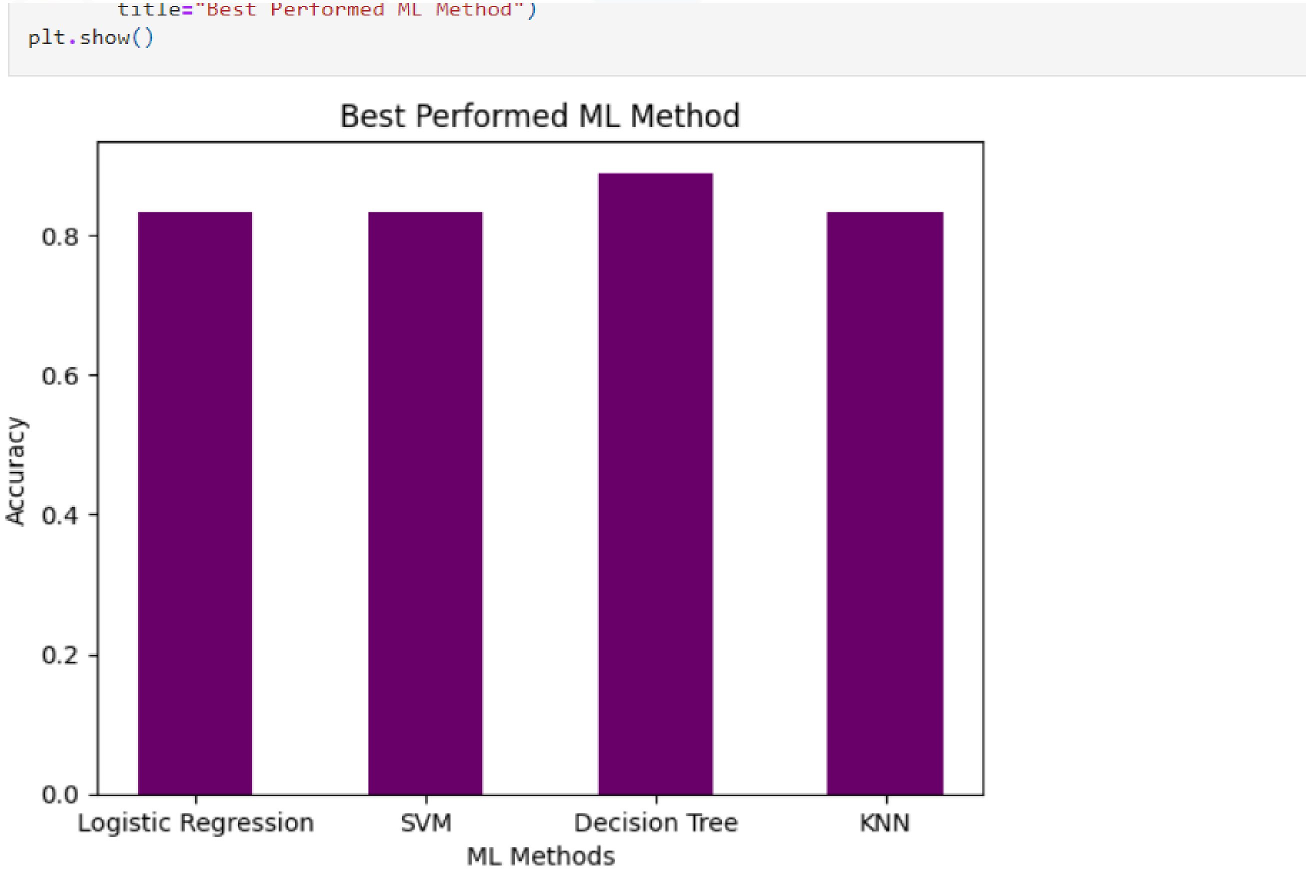
Classification Accuracy

All (except for Decision Tree) have relatively the same accuracy on the test set of 83.33%

Decision Tree’s test accuracy: 88,88%

Test size only has 18 samples, which can cause large variance in accuracy result

More data is more likely needed to determine the most accurate model



```
        title="Best Performed ML Method")
plt.show()
```

Confusion Matrix

This confusion matrix is the same across for Logistic Regression and Support Vector Machines since the same test set is used to evaluate those models

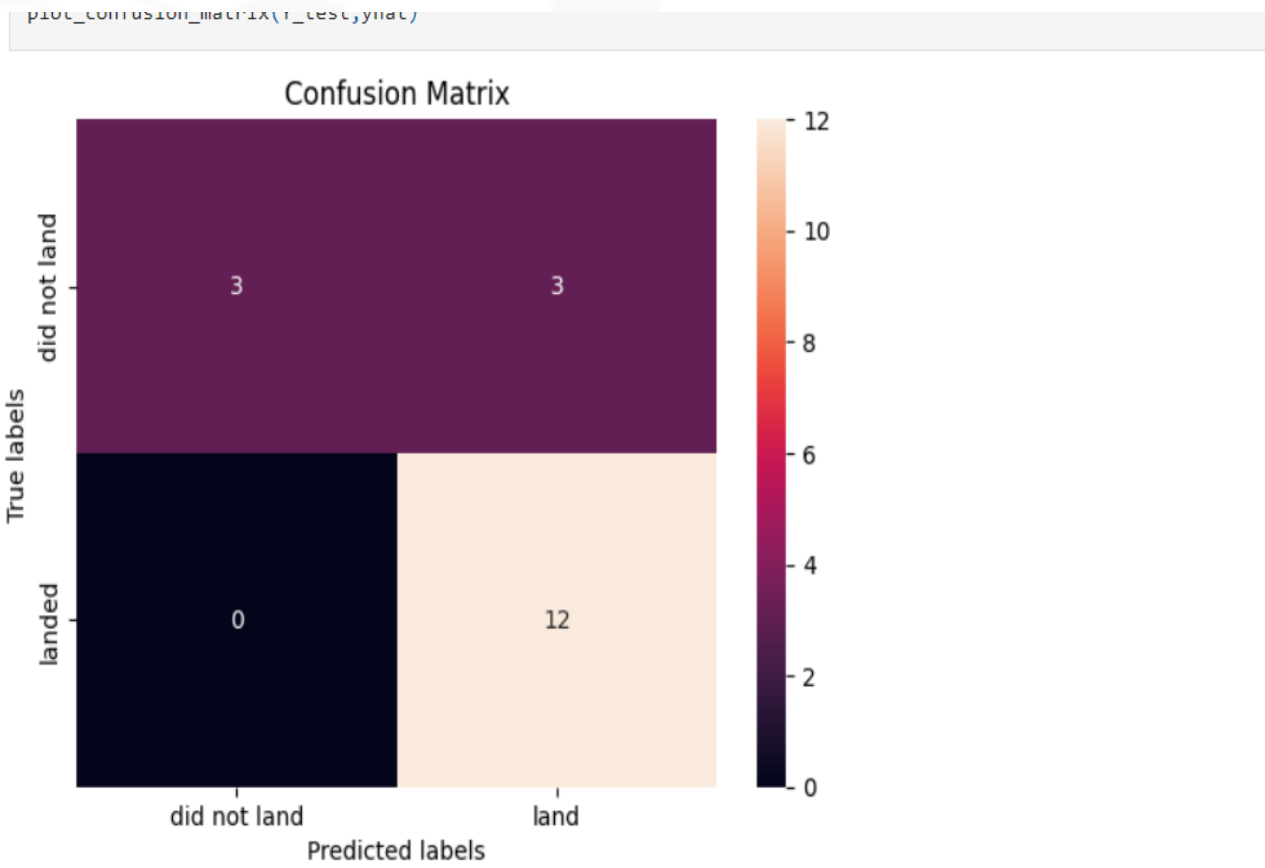
All correct predictions are on the diagonal of the matrix starting from top left to bottom right.

The models predicted 12 successful landings when the true label = landed

The models predicted 3 unsuccessful landings when the true label = did not land

The models predicted 3 successful landings when the true label = did not land

Successful landings were over predicted



```
plot_confusion_matrix(Y_test,yhat)
```

Conclusions

Task: Develop a machine learning model for Space Y who wants to bid against Space X

A machine model was created with an accuracy of 83.33%

The goal of the model is to predict when Stage One will successfully land to save ~$100 million USD

Space Y can implement this model to predict whether a launch will have successful Stage One landing before determining whether a launch should be made or not

Suggestion: Collect more data , if possible, to better determine the best machine learning model and improve model accuracy

Thank you!