IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Hector Oliva

12 May 2025

Outline

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

Executive Summary

- Summary of methodologies
- Summary of all results

Introduction

- Project background and context
- Problems you want to find answers

Section 1

Methodology

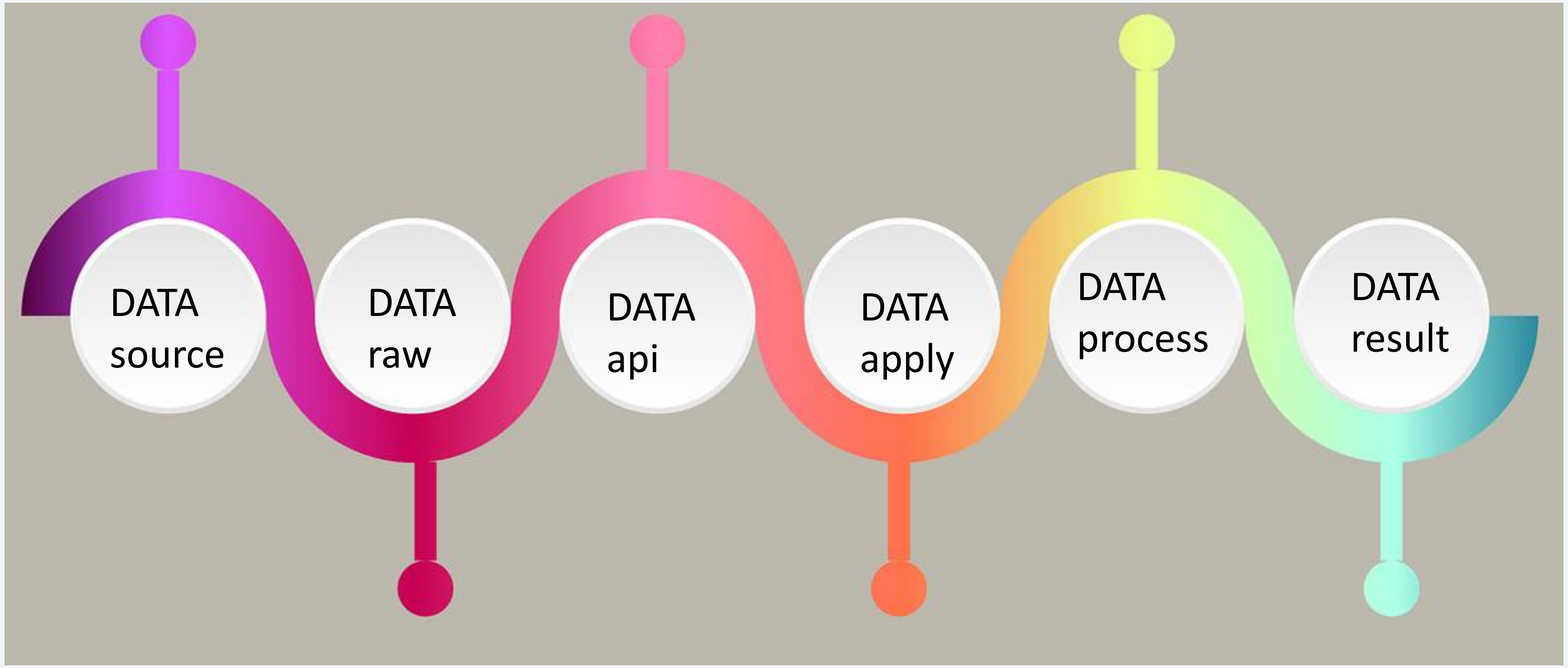
Methodology

Executive Summary

- Data collection methodology:
  - By API Lab Ungraded App
  - By Web Scraping
- Perform data wrangling
  - Perform some Exploratory Data Analysis (EDA) to find some patterns in the data and determine what would be the label for training supervised models.Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection

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



DATA source

DATA raw

DATA api

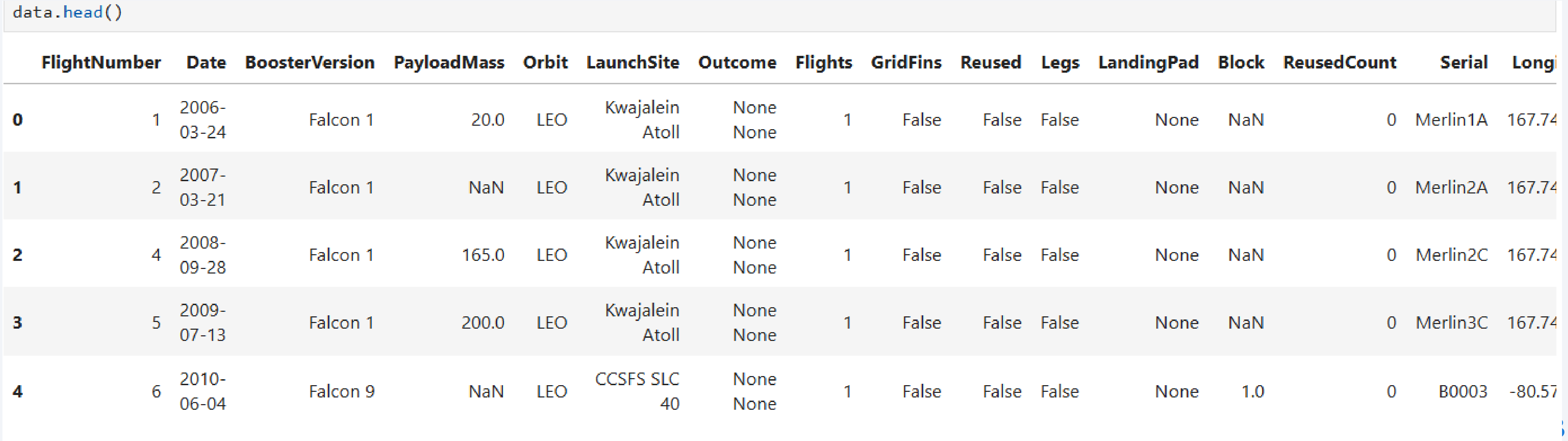
DATA apply

DATA process

DATA result

Data Collection – SpaceX API

- Present your data collection with SpaceX REST calls using key phrases and flowcharts
- Add the GitHub URL of the completed SpaceX API calls notebook (must include completed code cell and outcome cell), as an external reference and peer-review purpose

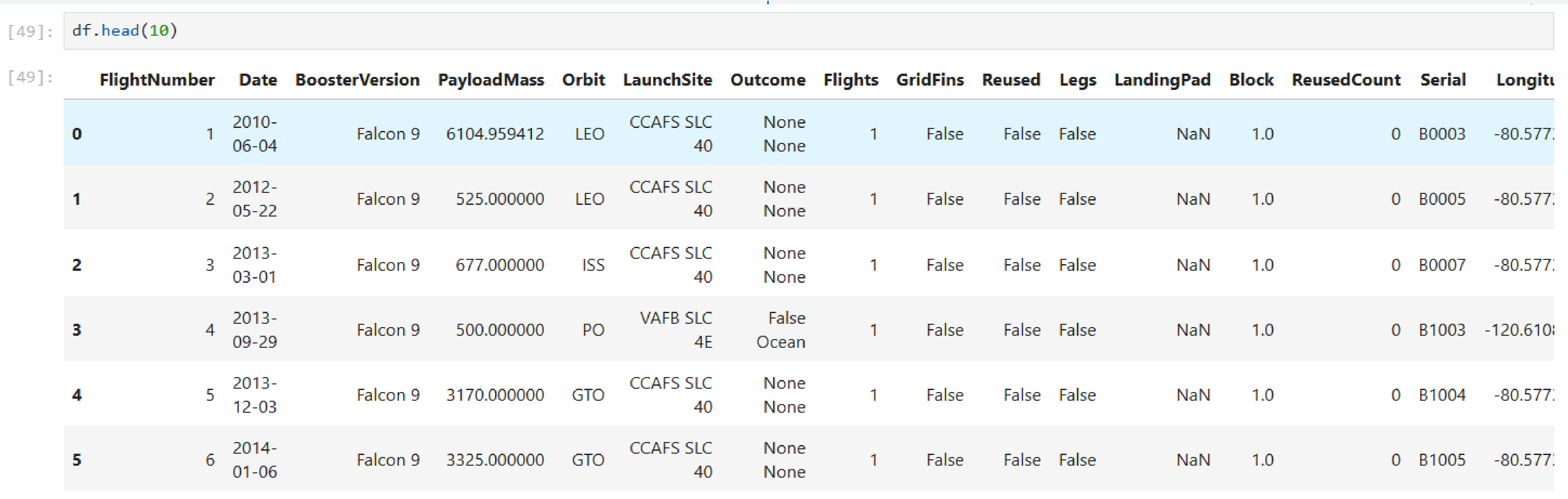


```
data.head()
```

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	Longi
0	1	2006-03-24	Falcon 1	20.0	LEO	Kwajalein Atoll	None None	1	False	False	False	None	NaN	0	Merlin1A	167.74
1	2	2007-03-21	Falcon 1	NaN	LEO	Kwajalein Atoll	None None	1	False	False	False	None	NaN	0	Merlin2A	167.74
2	4	2008-09-28	Falcon 1	165.0	LEO	Kwajalein Atoll	None None	1	False	False	False	None	NaN	0	Merlin2C	167.74
3	5	2009-07-13	Falcon 1	200.0	LEO	Kwajalein Atoll	None None	1	False	False	False	None	NaN	0	Merlin3C	167.74
4	6	2010-06-04	Falcon 9	NaN	LEO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0003	-80.57

Data Collection - Scraping

- Present your web scraping process using key phrases and flowcharts
- Add the GitHub URL of the completed web scraping notebook, as an external reference and peer-review purpose



[49]:
```
df.head(10)
```

[49]:

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	Longitu
0	1	2010-06-04	Falcon 9	6104.959412	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0003	-80.577
1	2	2012-05-22	Falcon 9	525.000000	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0005	-80.577
2	3	2013-03-01	Falcon 9	677.000000	ISS	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0007	-80.577
3	4	2013-09-29	Falcon 9	500.000000	PO	VAFB SLC 4E	False Ocean	1	False	False	False	NaN	1.0	0	B1003	-120.610
4	5	2013-12-03	Falcon 9	3170.000000	GTO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B1004	-80.577
5	6	2014-01-06	Falcon 9	3325.000000	GTO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B1005	-80.577

EDA with Data Visualization

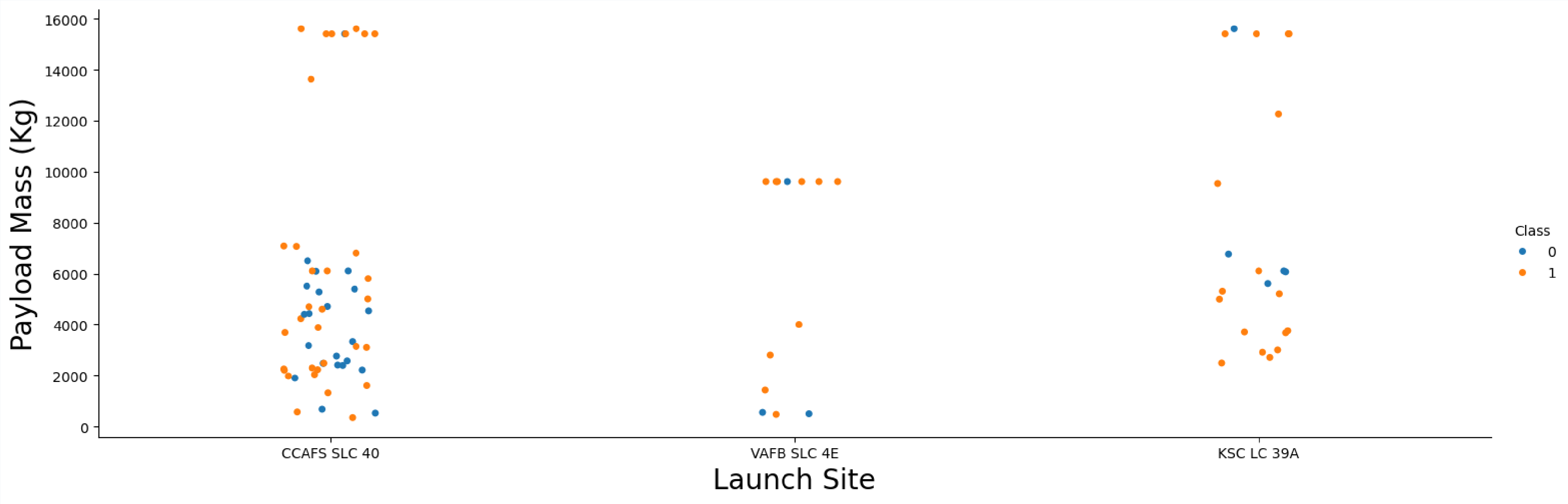
- Exploratory Data Analysis and Feature Engineering
- Create scatter plots and bar charts by writing Python code to analyze data in a Pandas data frame
- Write Python code to conduct exploratory data analysis by manipulating data in a Pandas data frame
- Create and execute SQL queries to select and sort data
- Utilize your data visualization skills to visualize the data and extract meaningful patterns to guide the modeling process

EDA with SQL

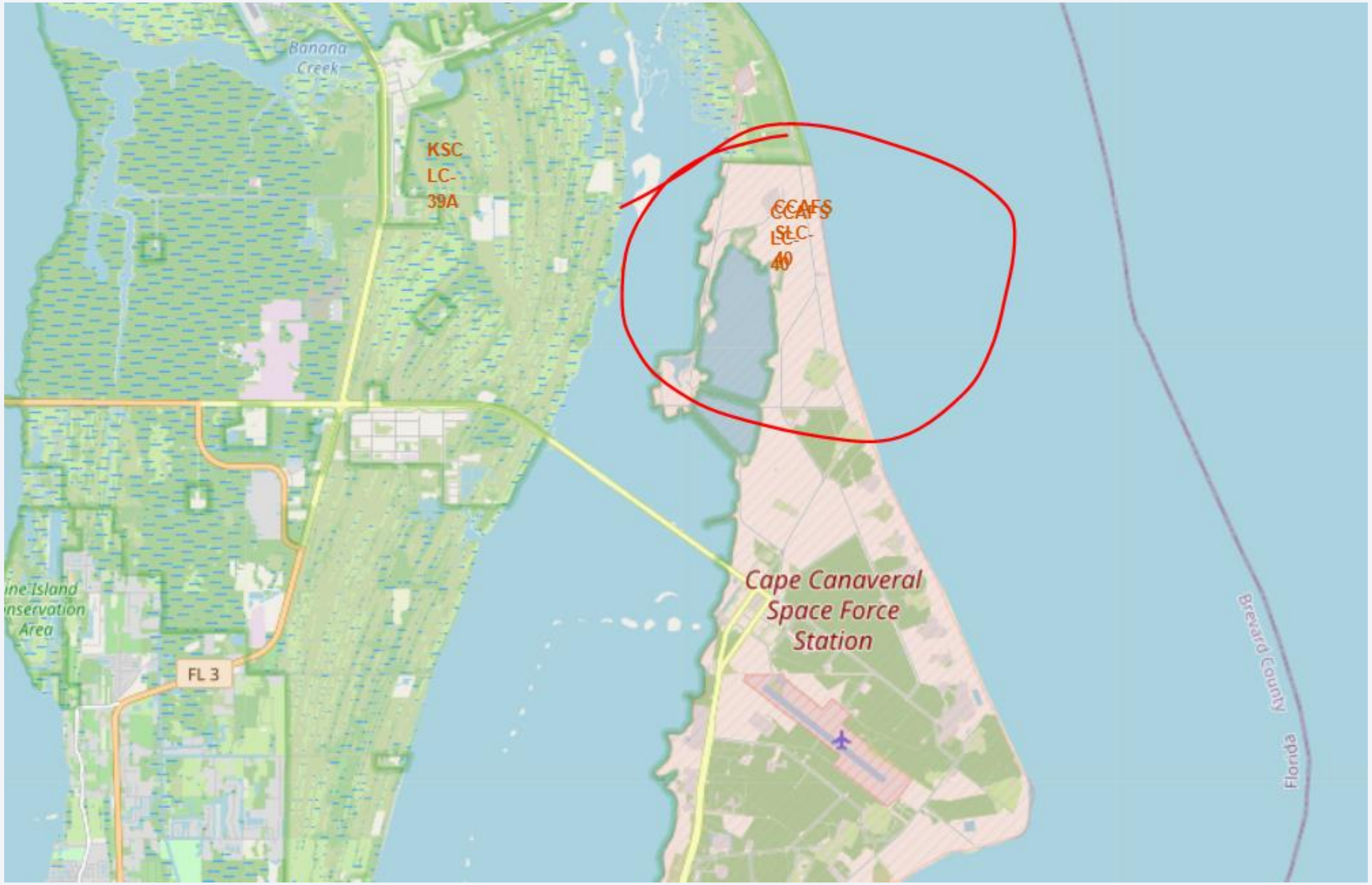
- Using bullet point format, summarize the SQL queries you performed
- Somenting…. For example:

  - %sql SELECT DISTINCT "Launch_Site" FROM SPACEXTABLE
  - %%sql SELECT * FROM SPACEXTABLE WHERE "Launch_Site" LIKE 'CCA%' LIMIT 5;
  - %%sql SELECT Date FROM SPACEXTABLEWHERE Landing_Outcome LIKE '%Success (ground pad)%'ORDER BY Date LIMIT 1
  - …. Others apply.

EDA with SQL

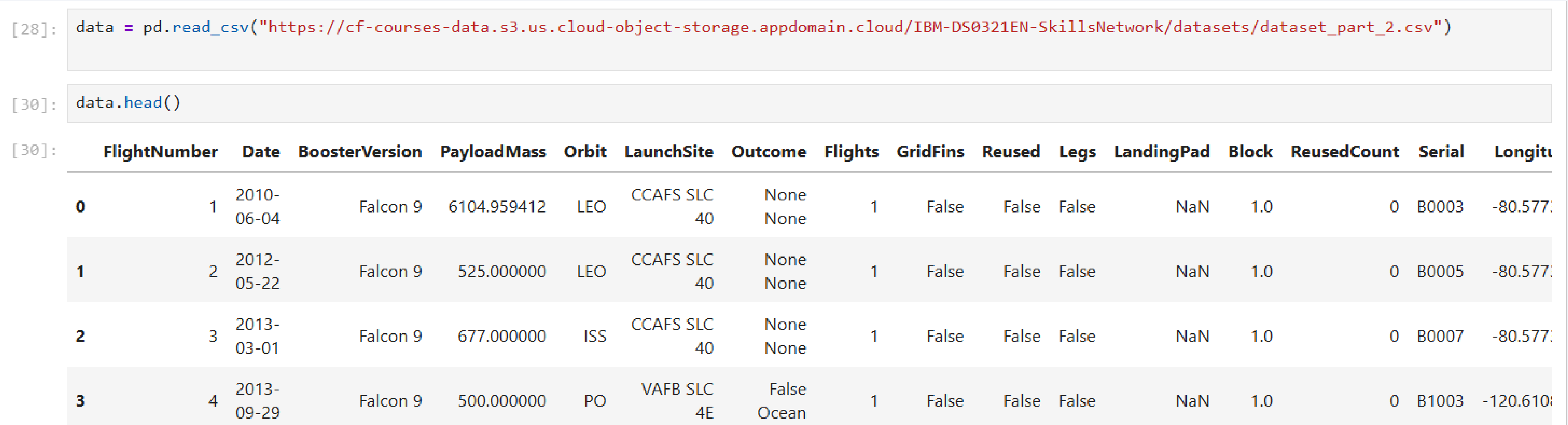

Payload Mass (Kg)
16000
14000
12000
10000
8000
6000
4000
2000
0
CCAFS SLC 40
VAFB SLC 4E
KSC LC 39A
Launch Site
Class
0
1

Build an Interactive Map with Folium



Predictive Analysis (Classification)

Find the best Hyperparameter for SVM, Classification Trees, and Logistic Regression.



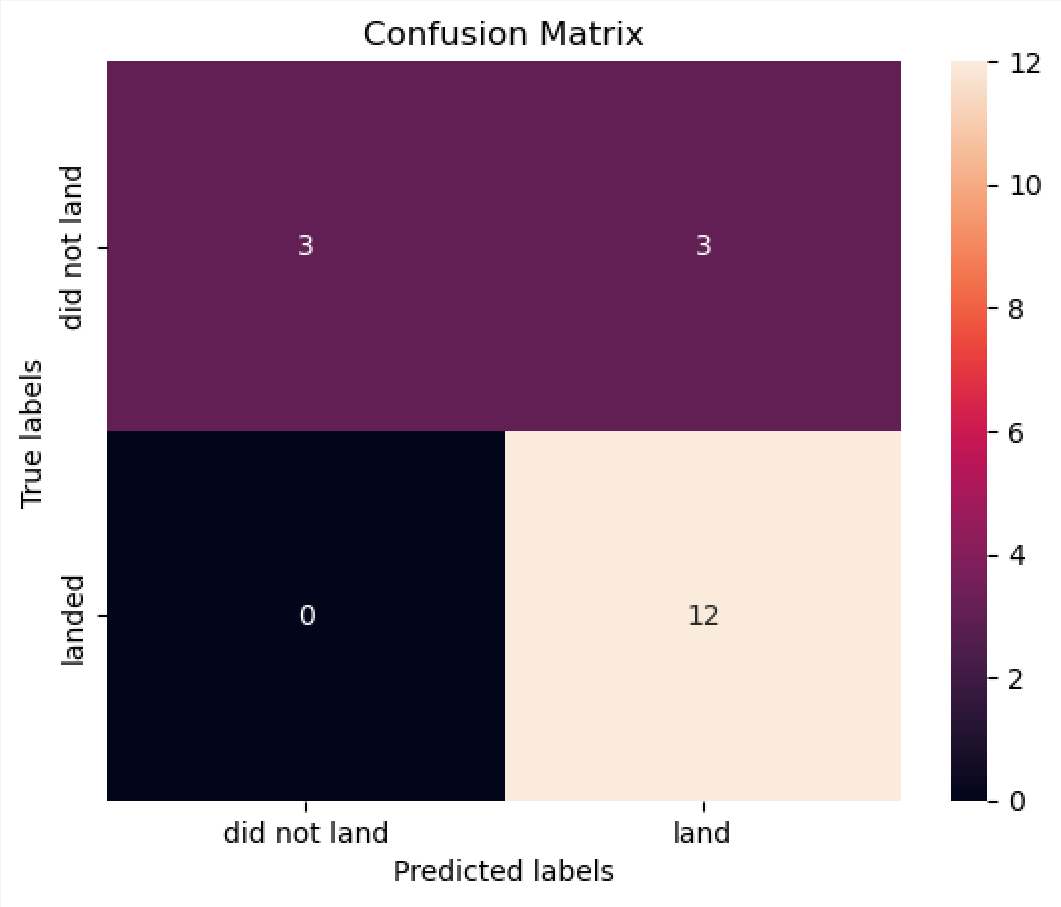
```
[28]: data = pd.read_csv("https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/dataset_part_2.csv")
```

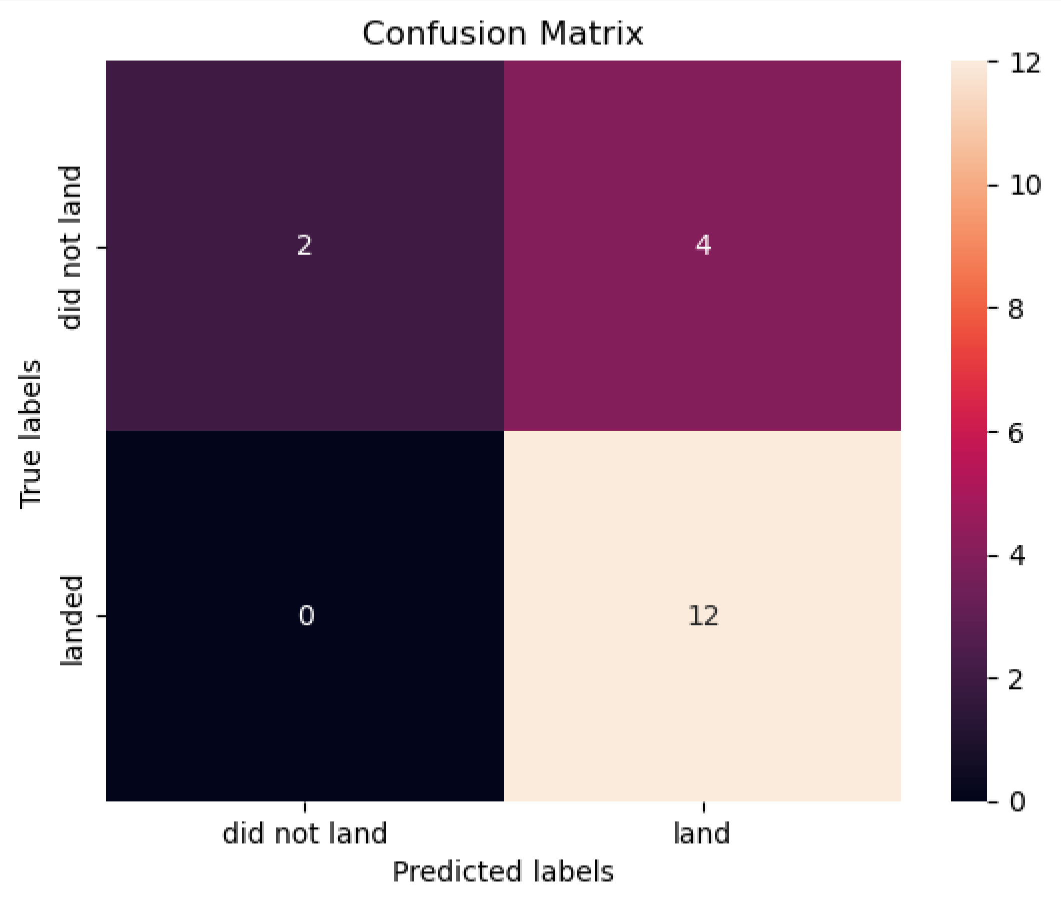
```
[30]: data.head()
```

[30]:

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	Longitu
0	1	2010-06-04	Falcon 9	6104.959412	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0003	-80.577
1	2	2012-05-22	Falcon 9	525.000000	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0005	-80.577
2	3	2013-03-01	Falcon 9	677.000000	ISS	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0007	-80.577
3	4	2013-09-29	Falcon 9	500.000000	PO	VAFB SLC 4E	False Ocean	1	False	False	False	NaN	1.0	0	B1003	-120.610

Results

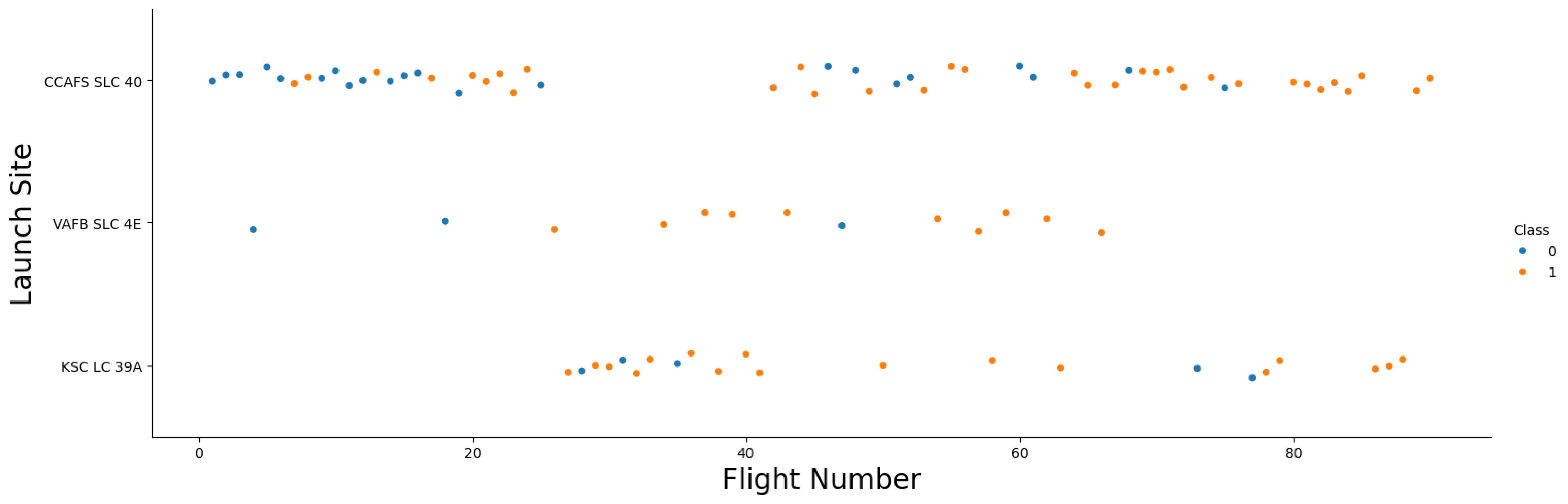

Confusion Matrix
True labels
did not land
landed
3
3
0
12
did not land
land
Predicted labels
12
10
8
6
4
2
0


Confusion Matrix
True labels
did not land
landed
2
4
0
12
did not land
land
Predicted labels
12
10
8
6
4
2
0

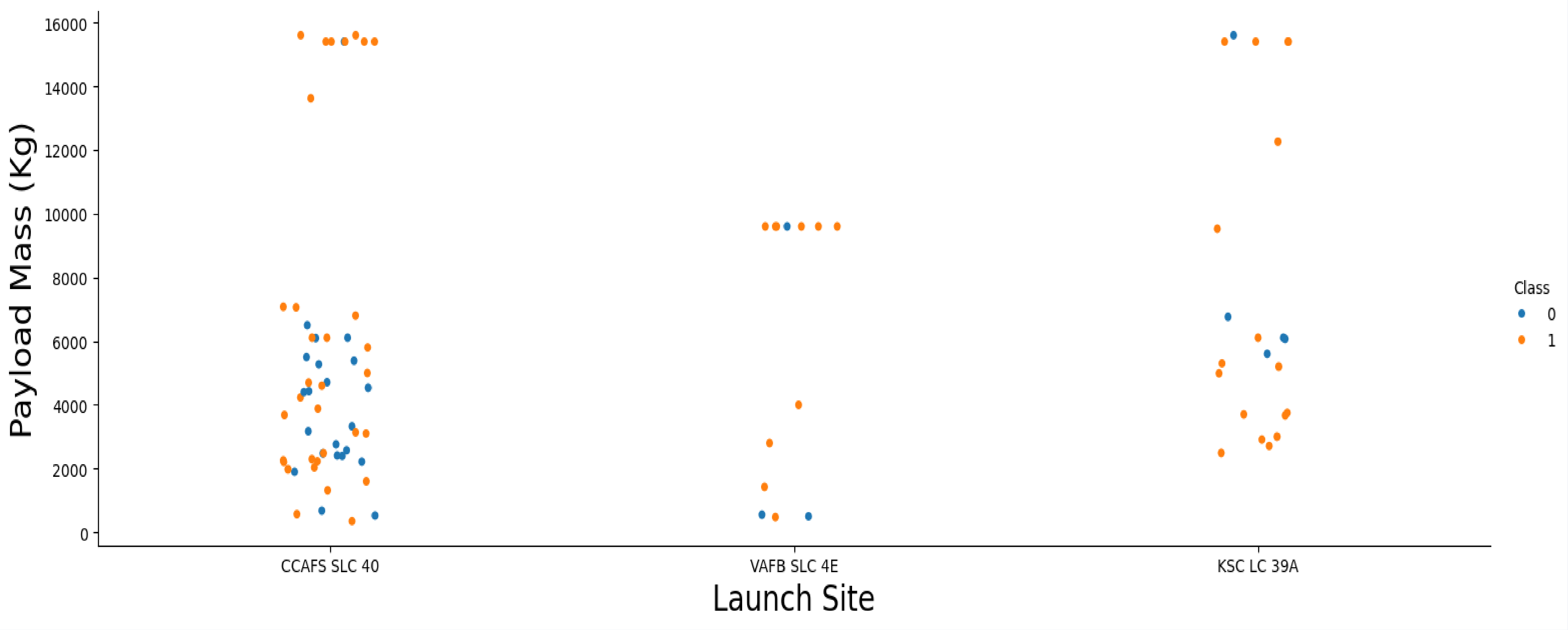
Section 2

Insights drawn from EDA

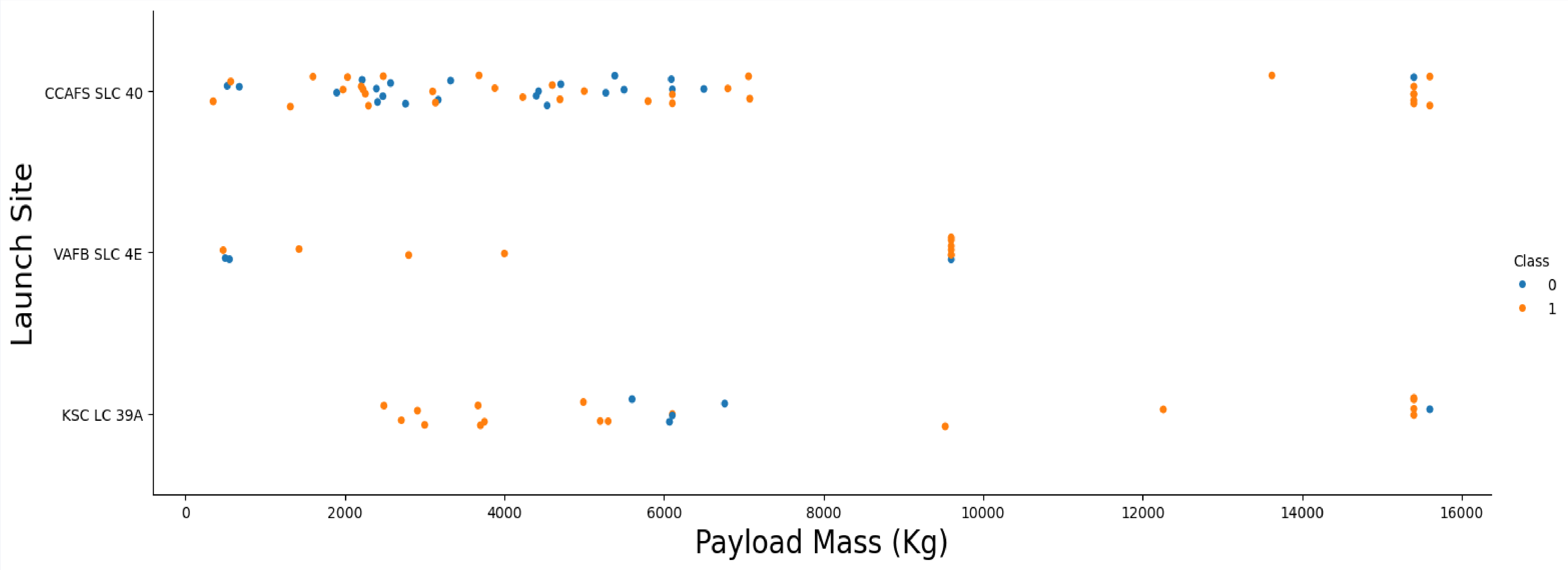
Flight Number vs. Launch Site



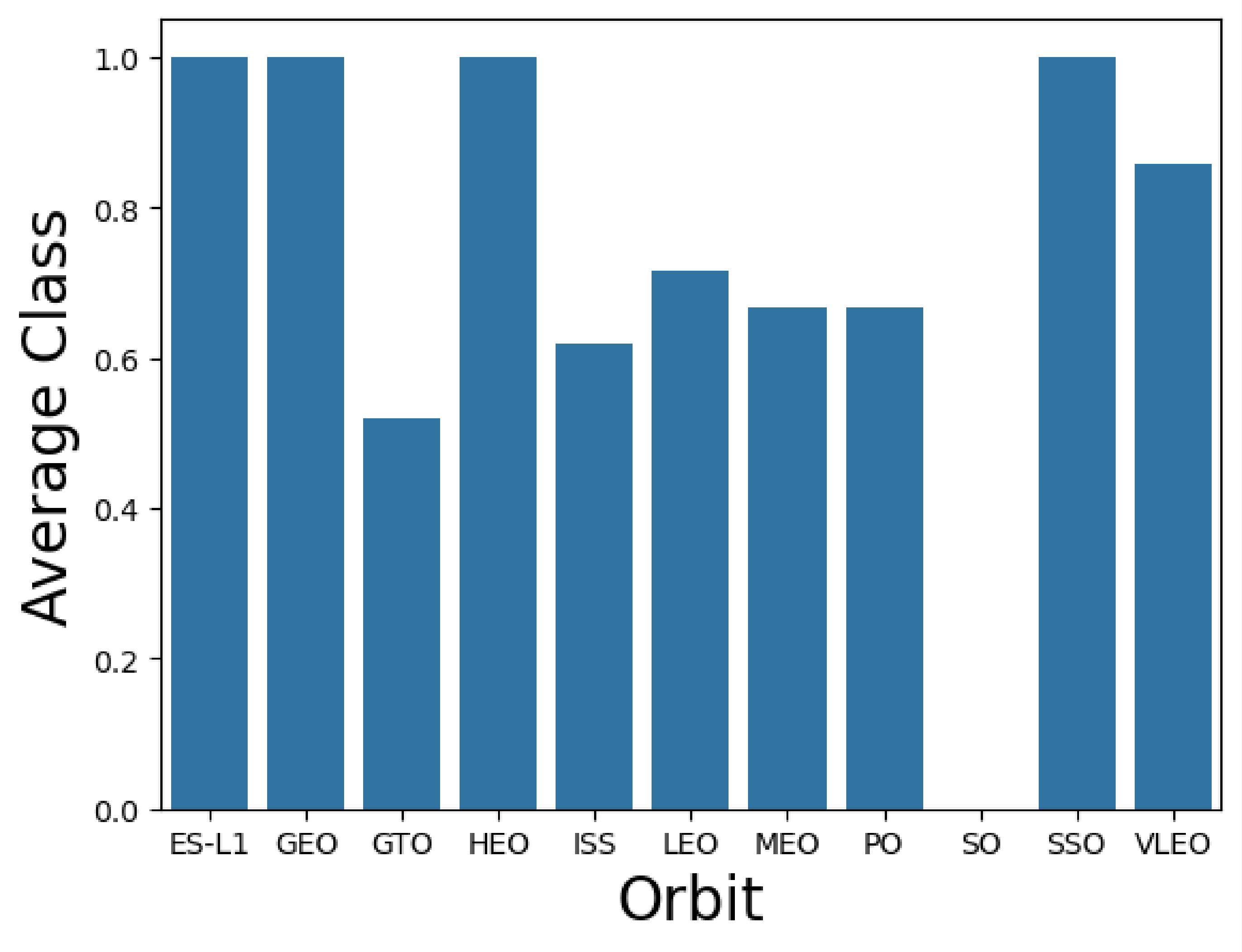
Payload vs. Launch Site



Payload vs. Launch Site

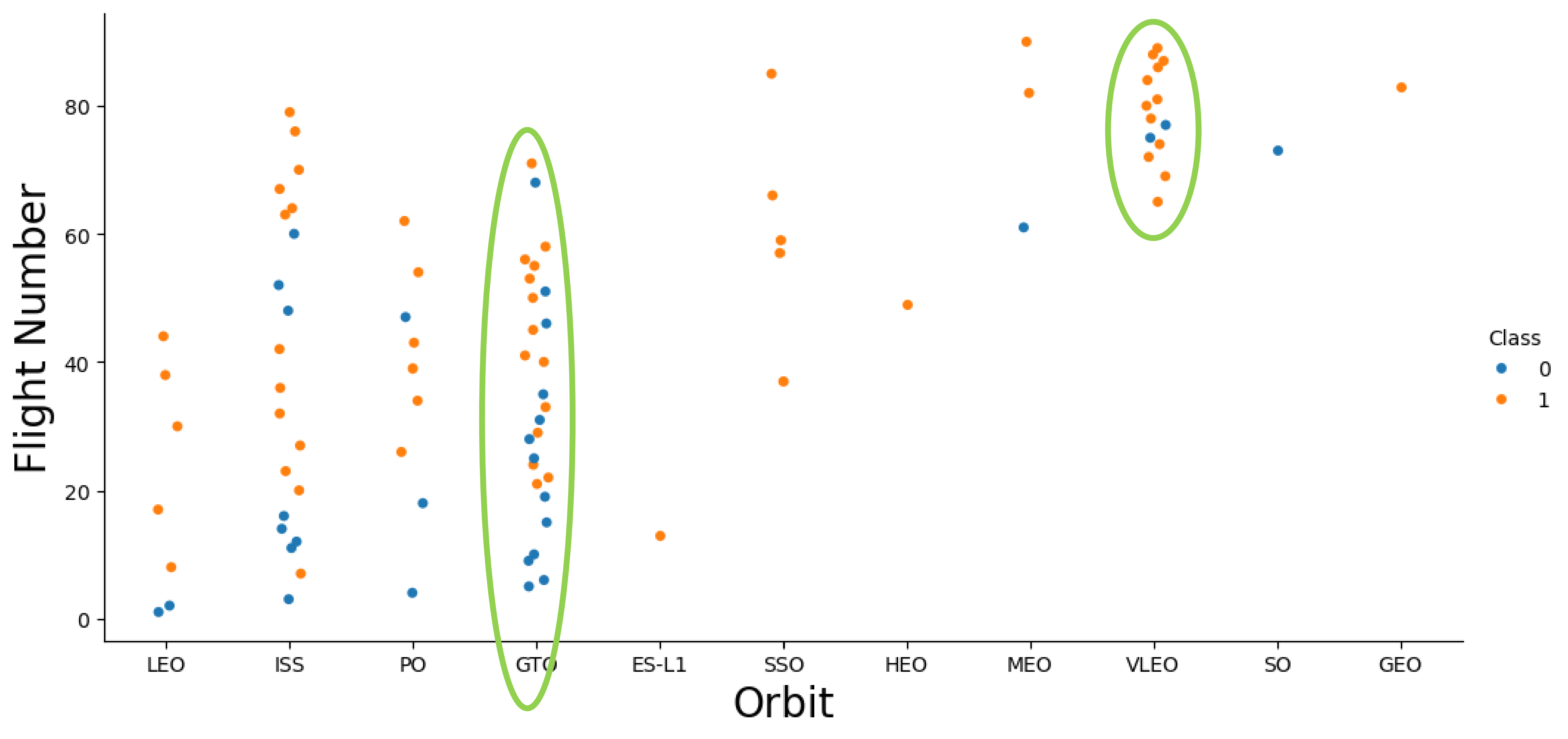


Success Rate vs. Orbit Type

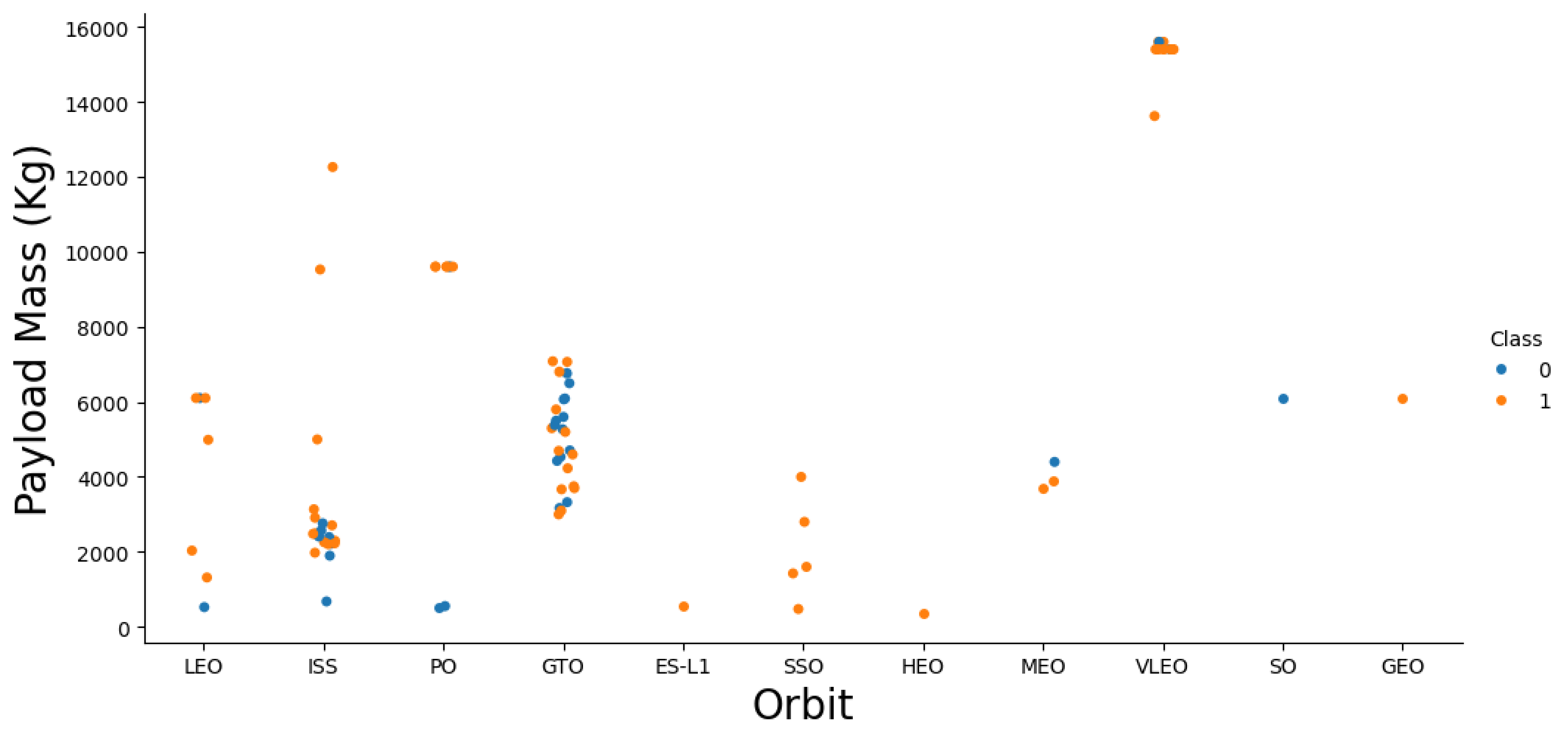


Most important:
ES – GEO – HEO SSO

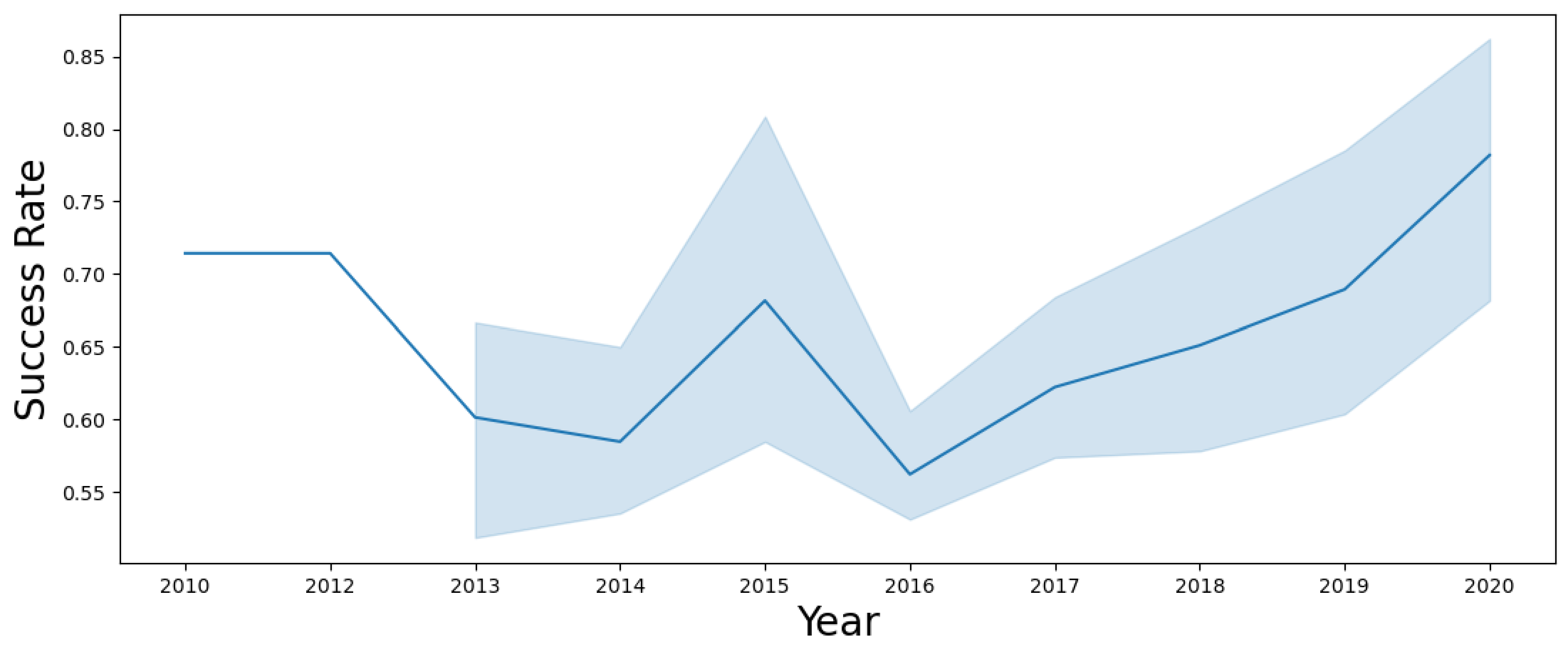
Flight Number vs. Orbit Type



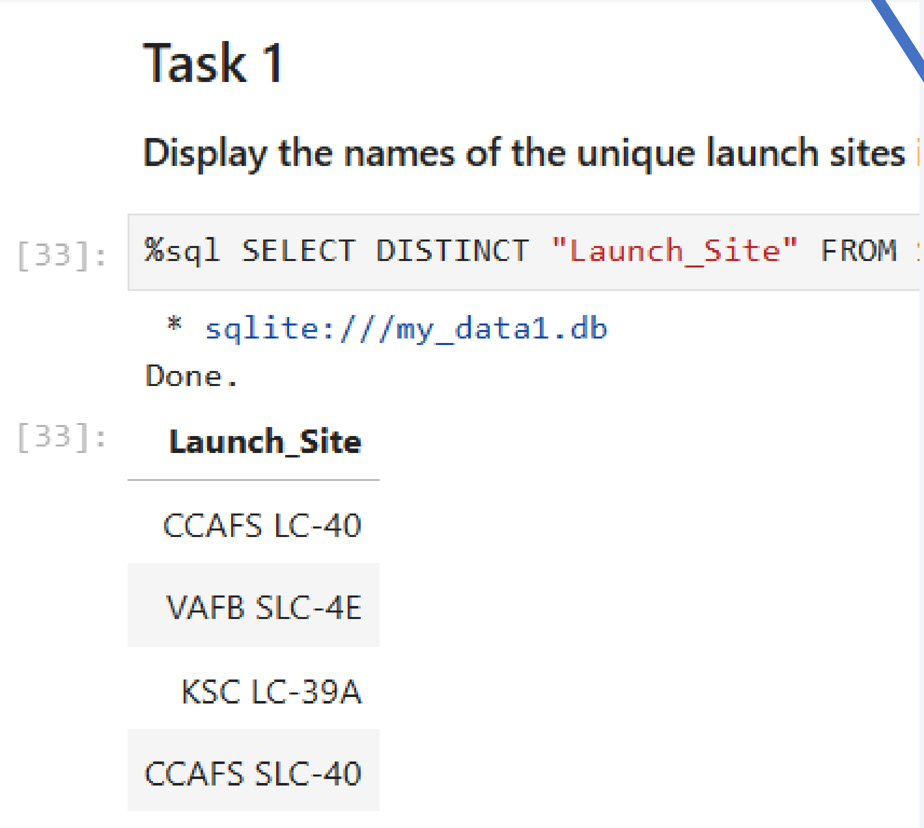
Payload vs. Orbit Type



Launch Success Yearly Trend



All Launch Site Names



Task 1

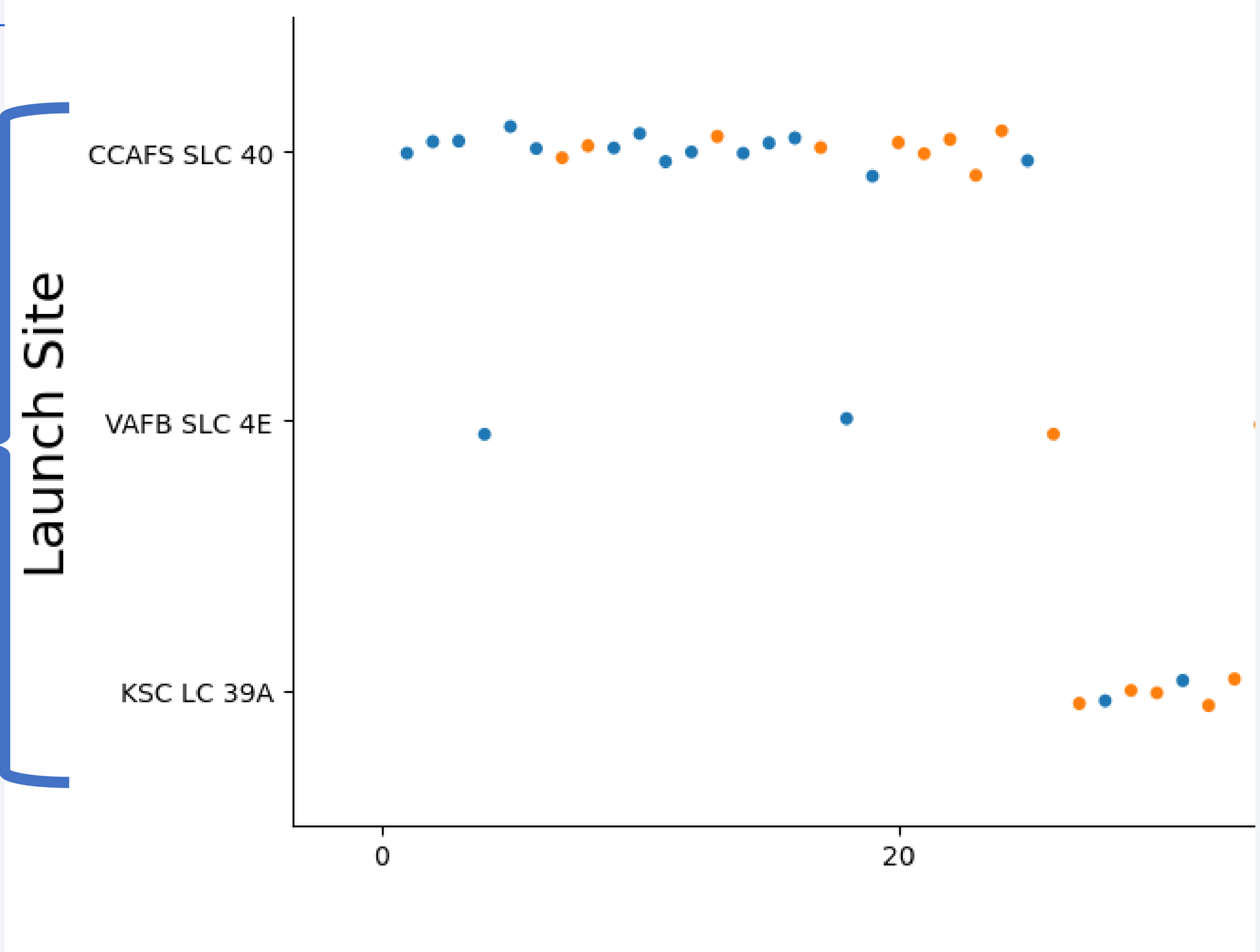
Display the names of the unique launch sites i

```
[33]: %sql SELECT DISTINCT "Launch_Site" FROM :
```

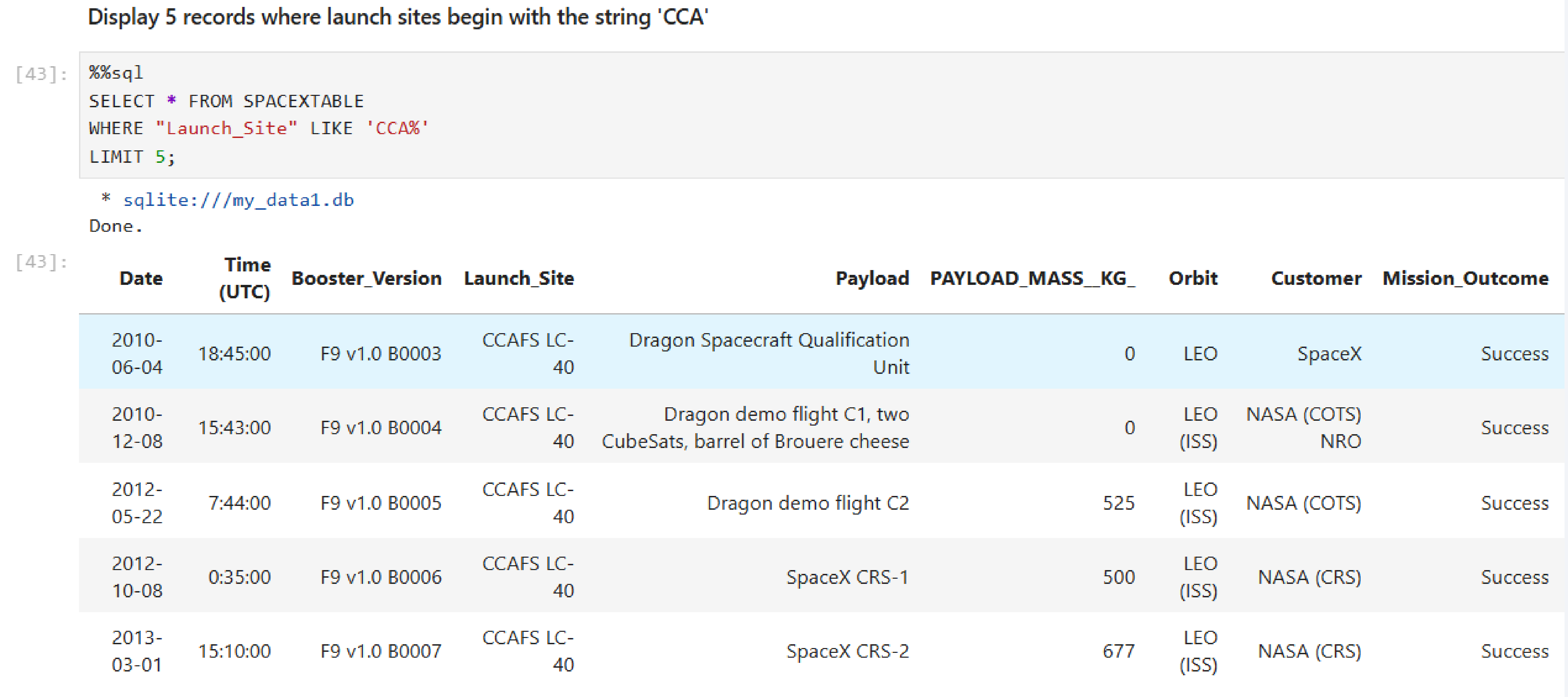
```
 * sqlite:///my_data1.db
Done.
```

[33]:

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



Launch Site Names Begin with 'CCA'



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

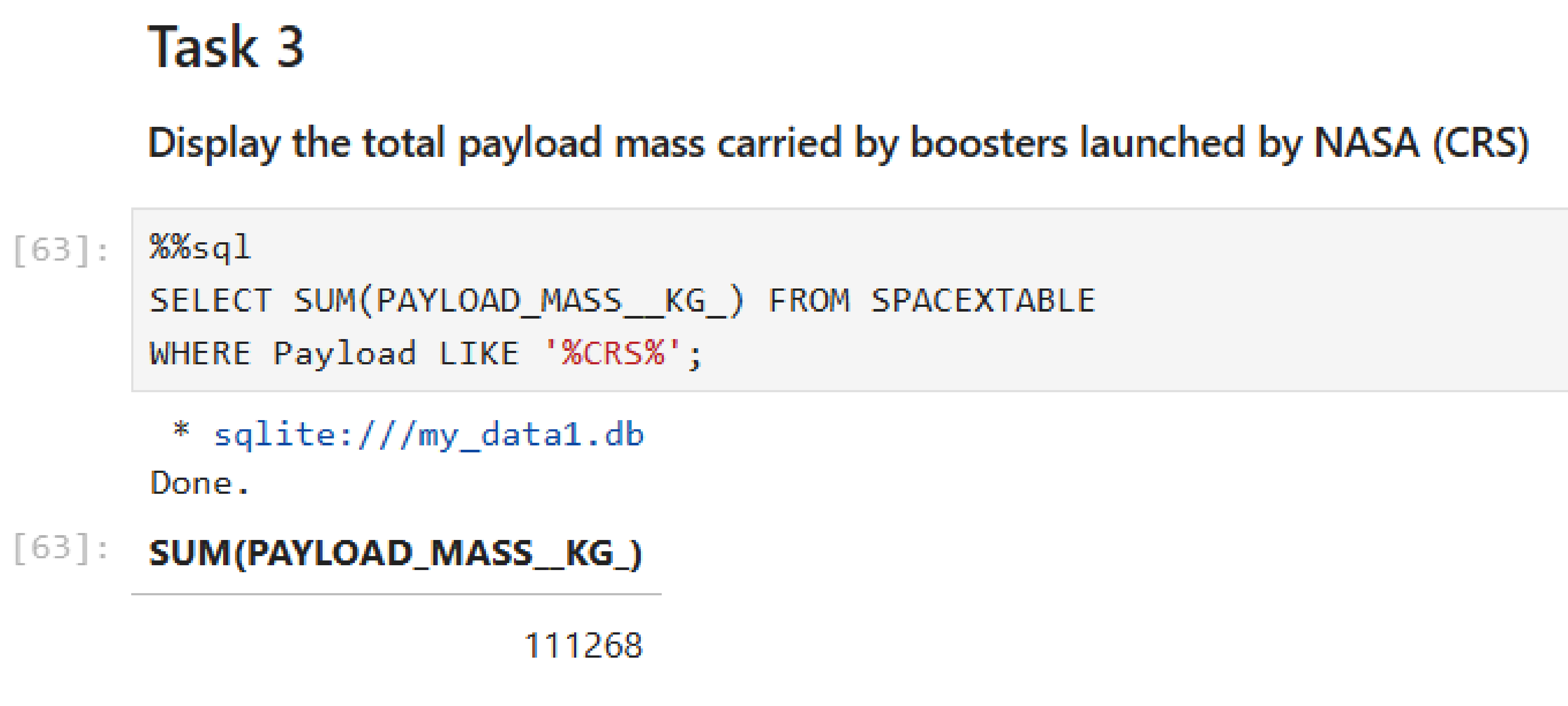
```
[43]: %%sql
      SELECT * FROM SPACEXTABLE
      WHERE "Launch_Site" LIKE 'CCA%'
      LIMIT 5;
```

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

[43]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success
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
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success

Total Payload Mass



Task 3

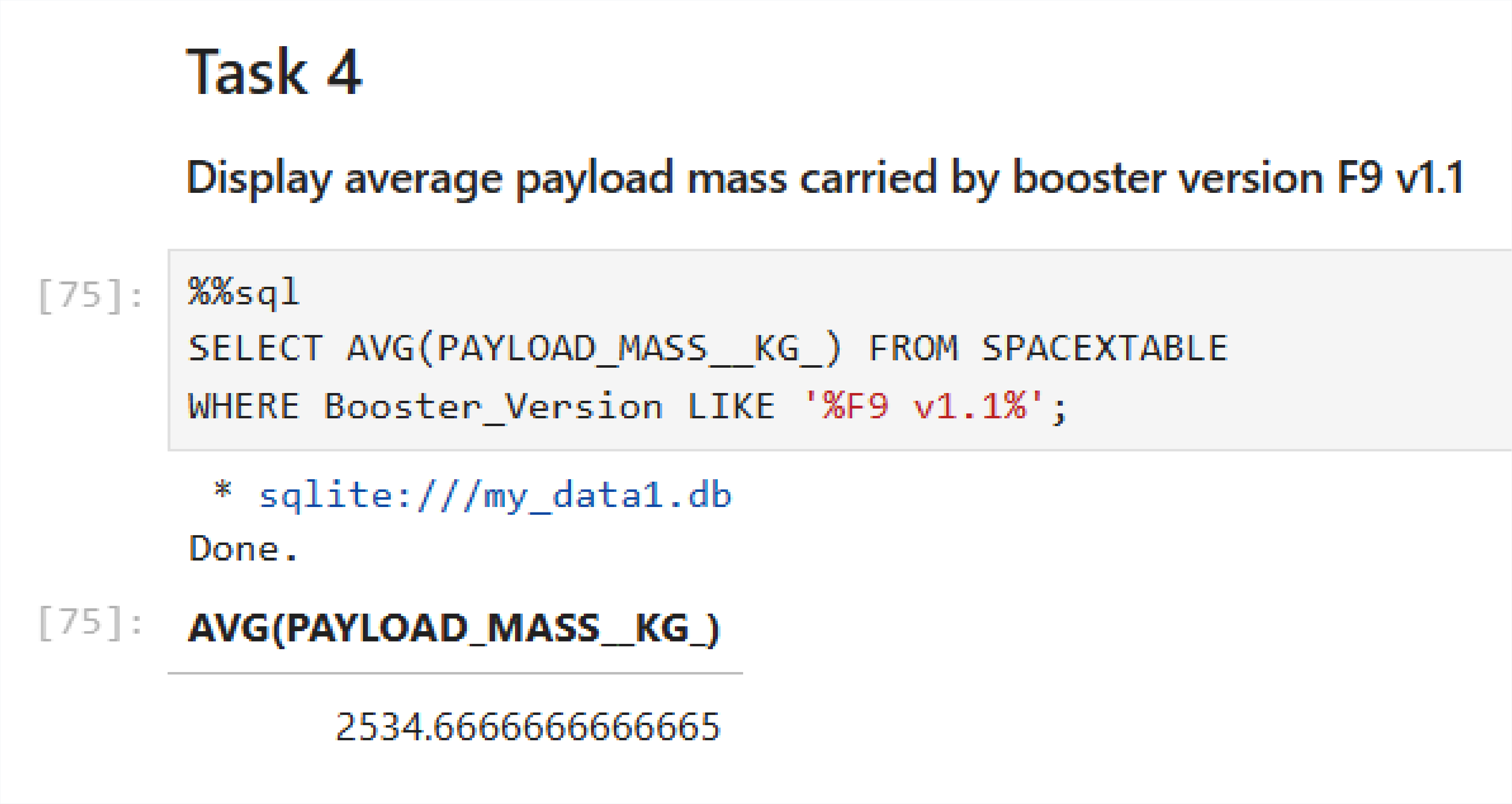
Display the total payload mass carried by boosters launched by NASA (CRS)

```
[63]: %%sql
      SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTABLE
      WHERE Payload LIKE '%CRS%';
```

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

```
[63]: SUM(PAYLOAD_MASS__KG_)
      ----------------------
                      111268
```

Average Payload Mass by F9 v1.1



Task 4

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

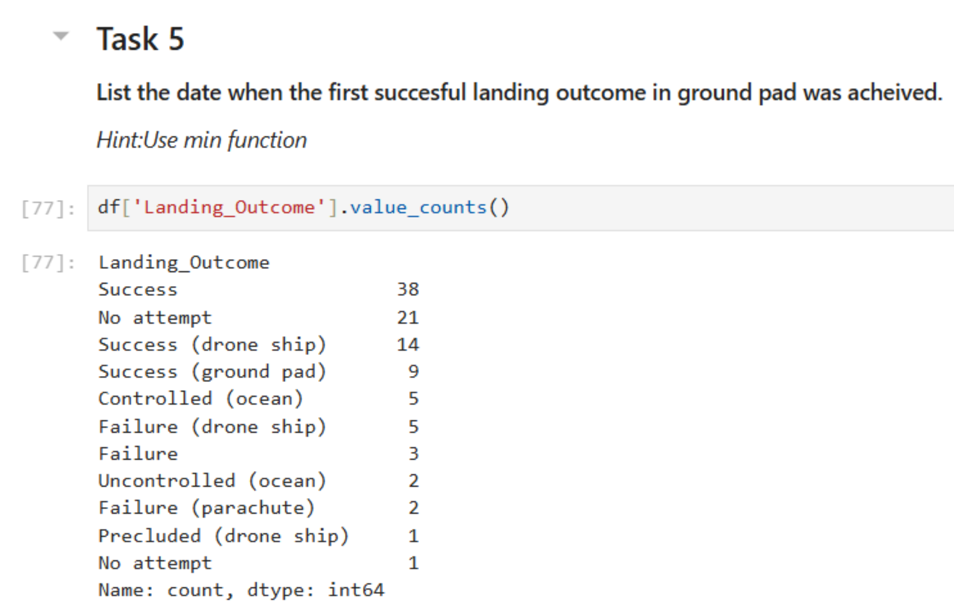
```
[75]: %%sql
SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTABLE
WHERE Booster_Version LIKE '%F9 v1.1%';
```

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

[75]:

AVG(PAYLOAD_MASS__KG_)
2534.6666666666665

First Successful Ground Landing Date



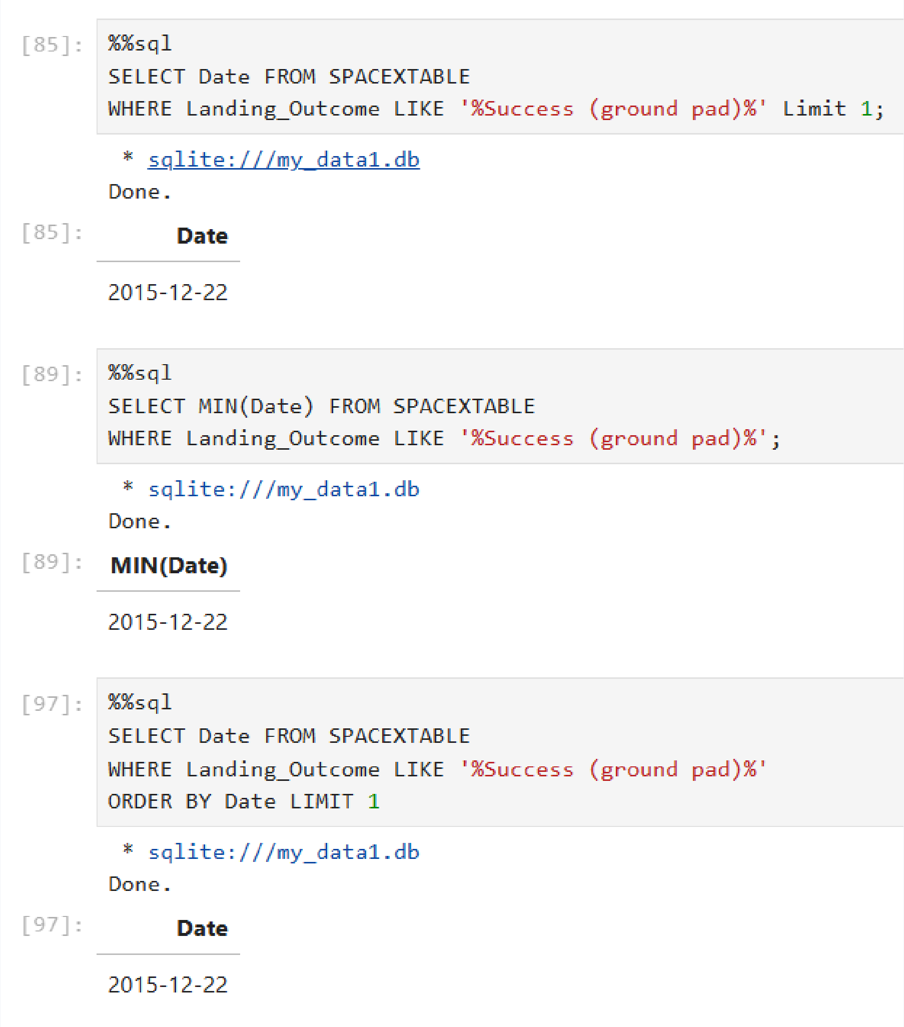
Task 5

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

Hint:Use min function

```
[77]: df['Landing_Outcome'].value_counts()
```

```
[77]: Landing_Outcome
      Success                   38
      No attempt                21
      Success (drone ship)      14
      Success (ground pad)       9
      Controlled (ocean)         5
      Failure (drone ship)       5
      Failure                    3
      Uncontrolled (ocean)       2
      Failure (parachute)        2
      Precluded (drone ship)     1
      No attempt                 1
      Name: count, dtype: int64
```



```
[85]: %%sql
      SELECT Date FROM SPACEXTABLE
      WHERE Landing_Outcome LIKE '%Success (ground pad)%' Limit 1;

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

[85]:

Date
2015-12-22

```
[89]: %%sql
      SELECT MIN(Date) FROM SPACEXTABLE
      WHERE Landing_Outcome LIKE '%Success (ground pad)%';

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

[89]:

MIN(Date)
2015-12-22

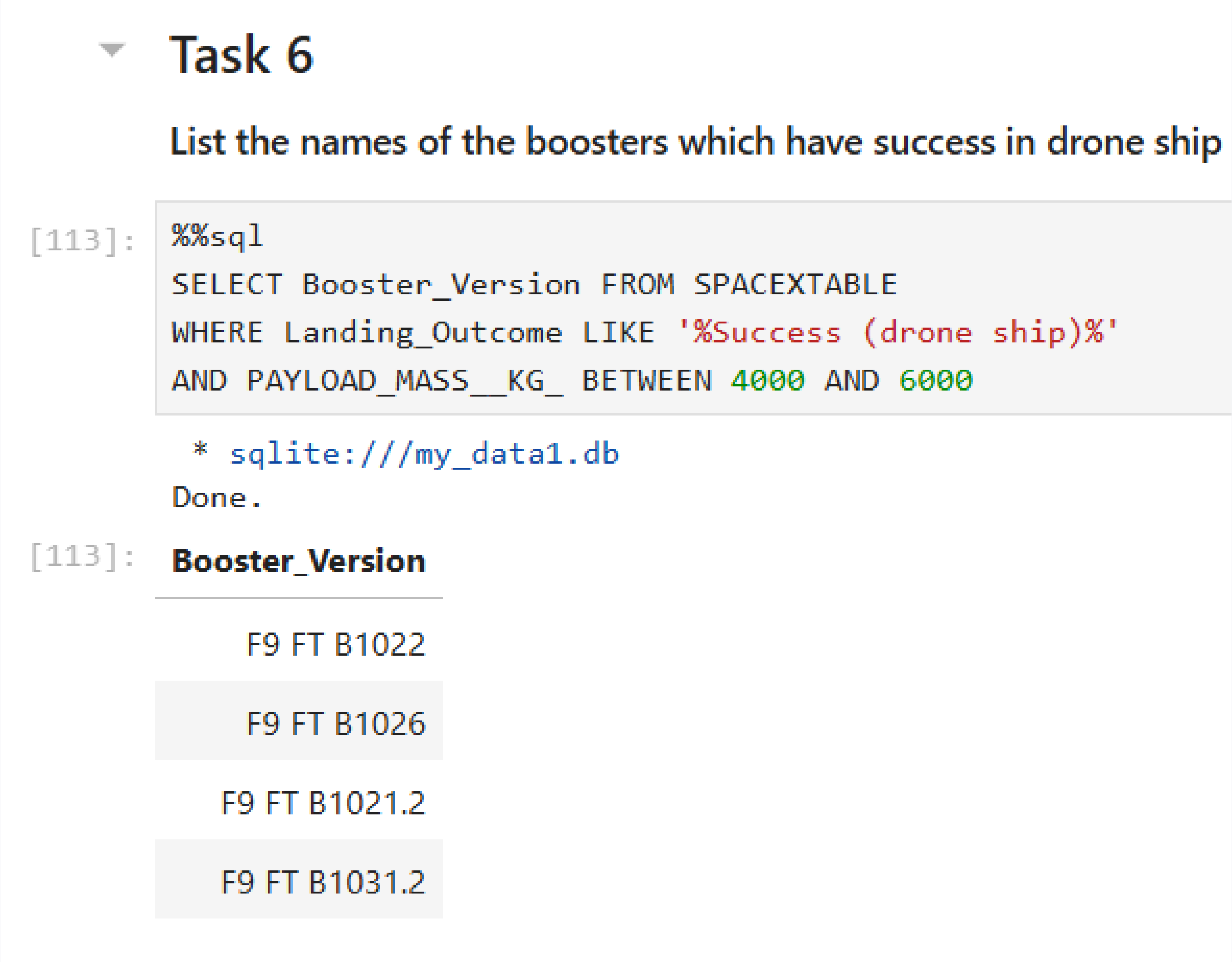
```
[97]: %%sql
      SELECT Date FROM SPACEXTABLE
      WHERE Landing_Outcome LIKE '%Success (ground pad)%'
      ORDER BY Date LIMIT 1

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

[97]:

Date
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000



Task 6

List the names of the boosters which have success in drone ship

[113]:

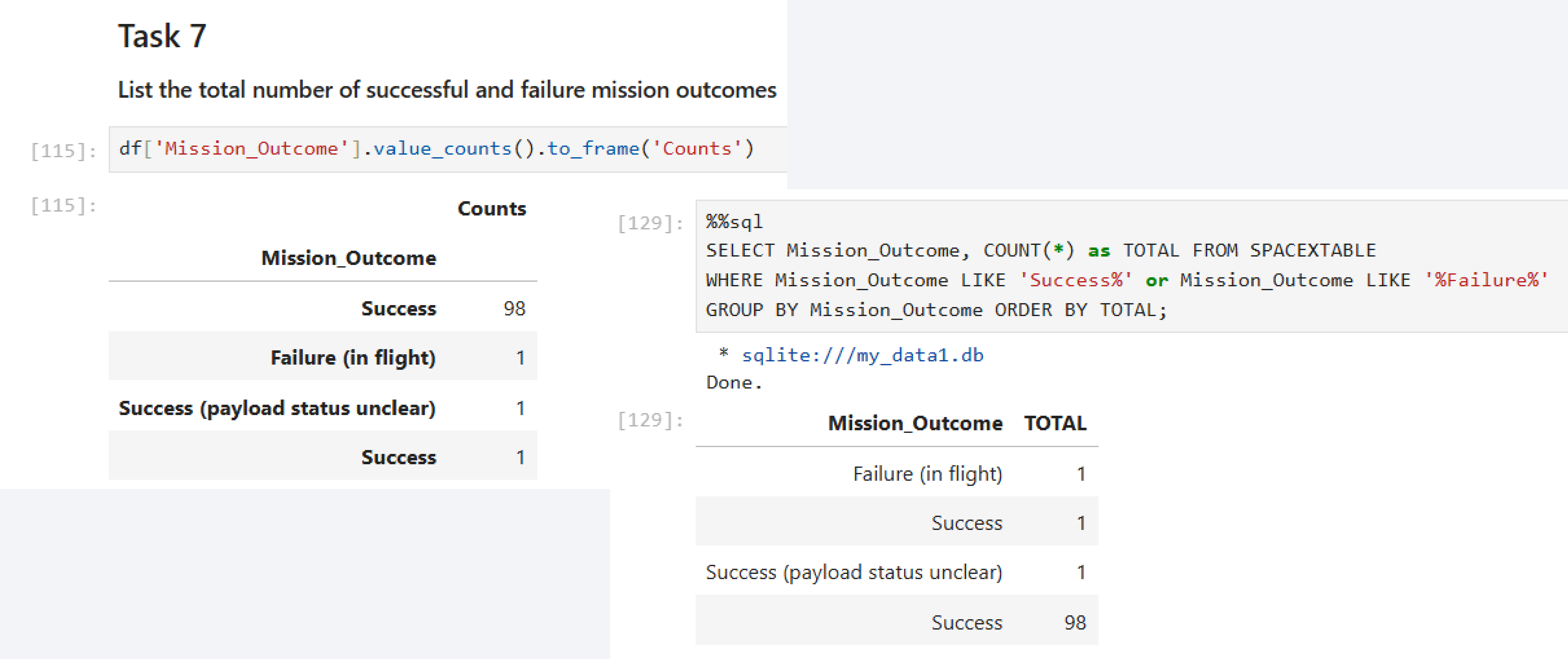
```
%%sql
SELECT Booster_Version FROM SPACEXTABLE
WHERE Landing_Outcome LIKE '%Success (drone ship)%'
AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000
```

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

[113]:

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

Total Number of Successful and Failure Mission Outcomes



Task 7

List the total number of successful and failure mission outcomes

```
[115]: df['Mission_Outcome'].value_counts().to_frame('Counts')
```

[115]:

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

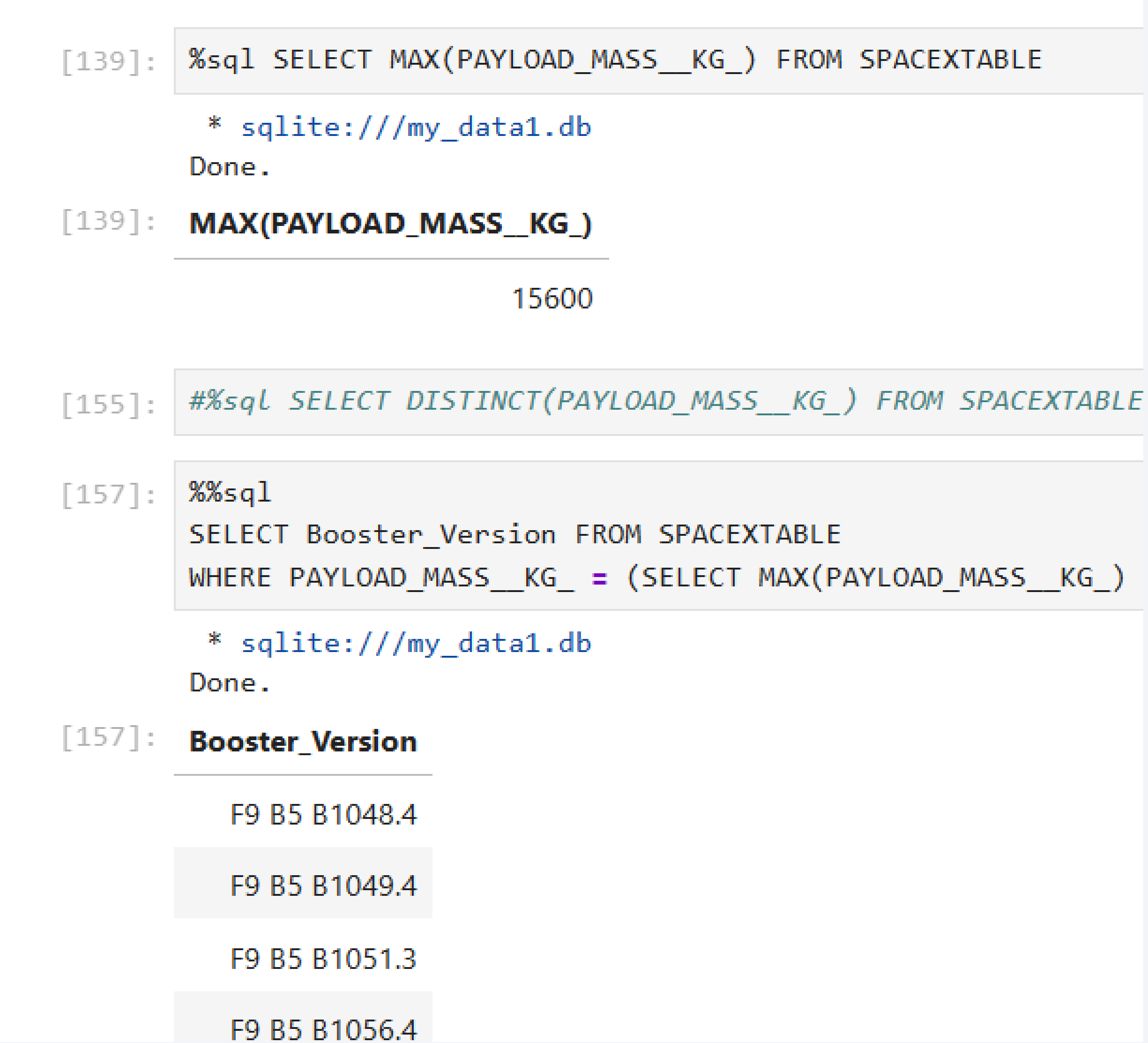
```
[129]: %%sql
SELECT Mission_Outcome, COUNT(*) as TOTAL FROM SPACEXTABLE
WHERE Mission_Outcome LIKE 'Success%' or Mission_Outcome LIKE '%Failure%'
GROUP BY Mission_Outcome ORDER BY TOTAL;
```

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

[129]:

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

Boosters Carried Maximum Payload



```
[139]: %sql SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE
```

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

[139]:

MAX(PAYLOAD_MASS__KG_)
15600

```
[155]: #%sql SELECT DISTINCT(PAYLOAD_MASS__KG_) FROM SPACEXTABLE
```

```
[157]: %%sql
       SELECT Booster_Version FROM SPACEXTABLE
       WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_)
```

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

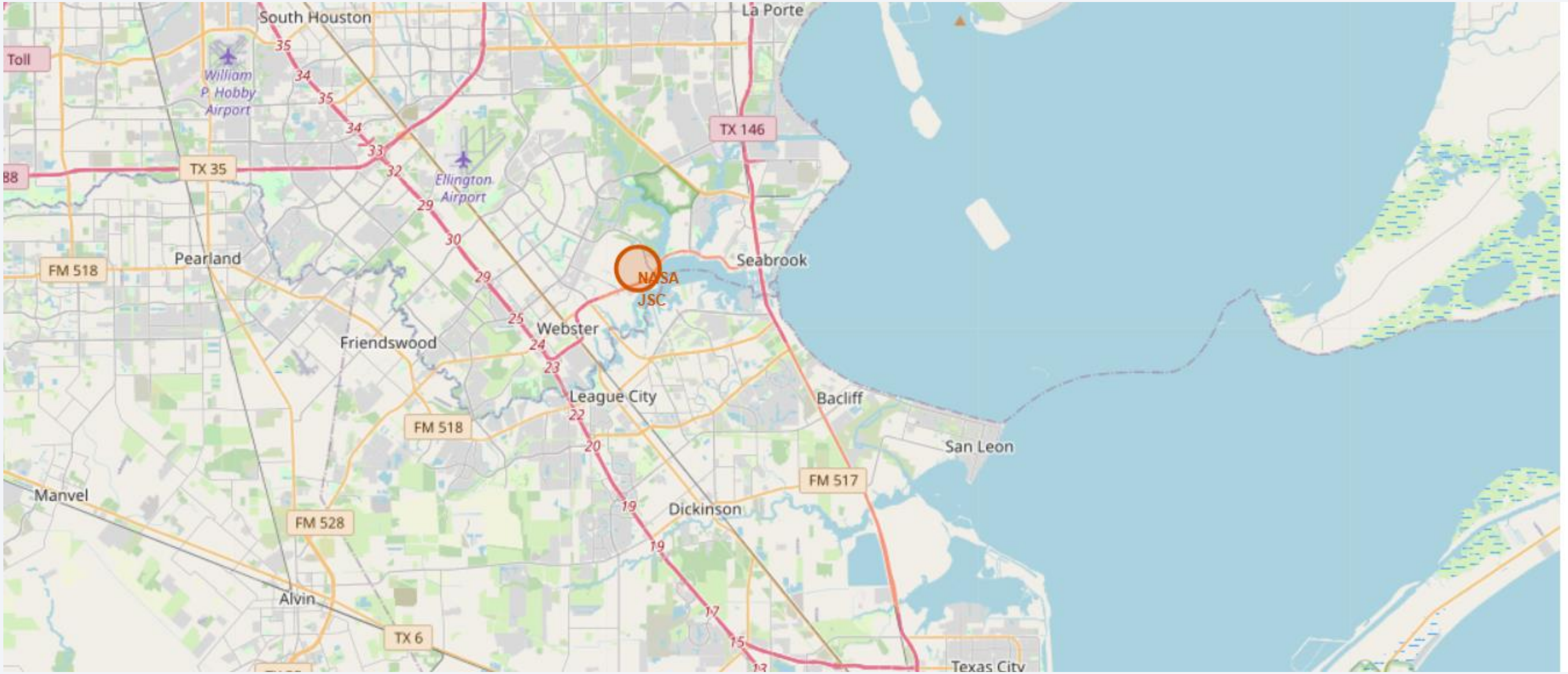
[157]:

Booster_Version
F9 B5 B1048.4
F9 B5 B1049.4
F9 B5 B1051.3
F9 B5 B1056.4

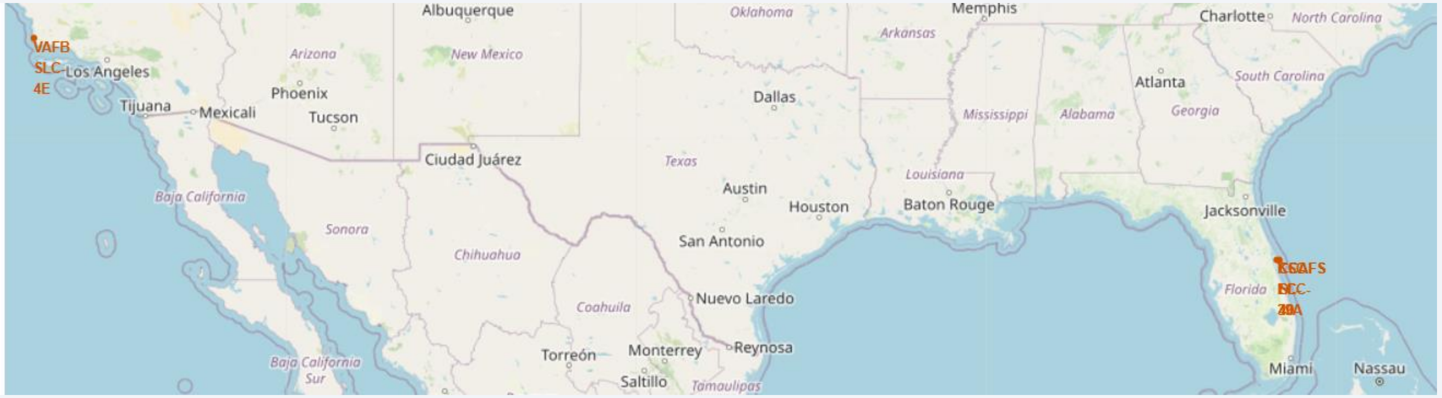
Section 3

Launch Sites Proximities Analysis

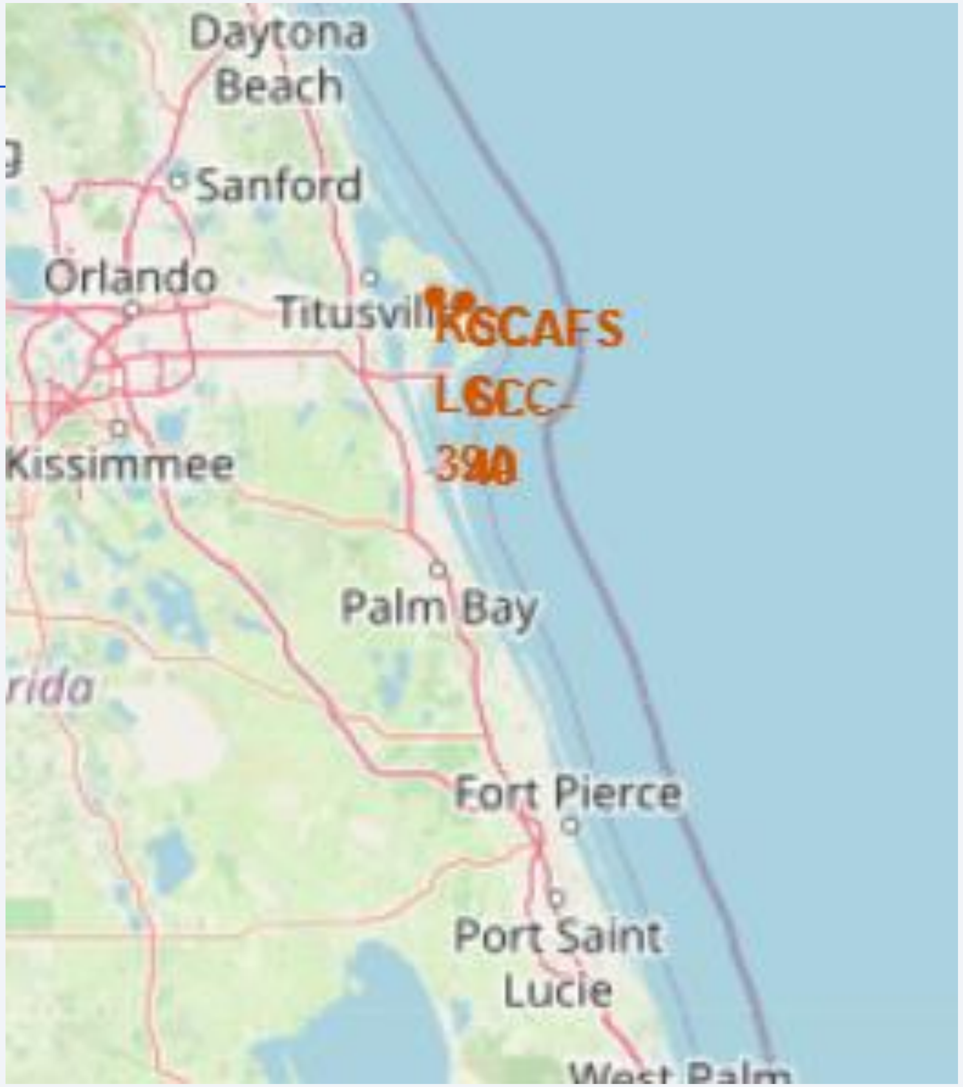
<Folium Map Screenshot 1>


South Houston
La Porte
Toll
William P. Hobby Airport
TX 146
TX 35
88
Ellington Airport
Seabrook
Pearland
FM 518
NASA JSC
Webster
Friendswood
League City
Bacliff
FM 518
San Leon
FM 517
Manvel
Dickinson
FM 528
Alvin
TX 6
Texas City

<Folium Map Screenshot 2>



VAFB
SLC-
4E
Los Angeles
Tijuana
Mexicali
Phoenix
Tucson
Arizona
Albuquerque
New Mexico
Oklahoma
Arkansas
Memphis
Dallas
Charlotte
North Carolina
South Carolina
Atlanta
Georgia
Mississippi
Alabama
Louisiana
Baton Rouge
Ciudad Juárez
Texas
Austin
Houston
San Antonio
Jacksonville
Baja California
Sonora
Chihuahua
Coahuila
Nuevo Laredo
Torreón
Monterrey
Saltillo
Reynosa
Tamaulipas
Baja California Sur
Florida
KSCAFS
LSCC-
39A
Miami
Nassau



Daytona Beach
Sanford
Orlando
Titusvil
KSCAFS
LSCC-
39A
Kissimmee
Palm Bay
rida
Fort Pierce
Port Saint Lucie
West Palm

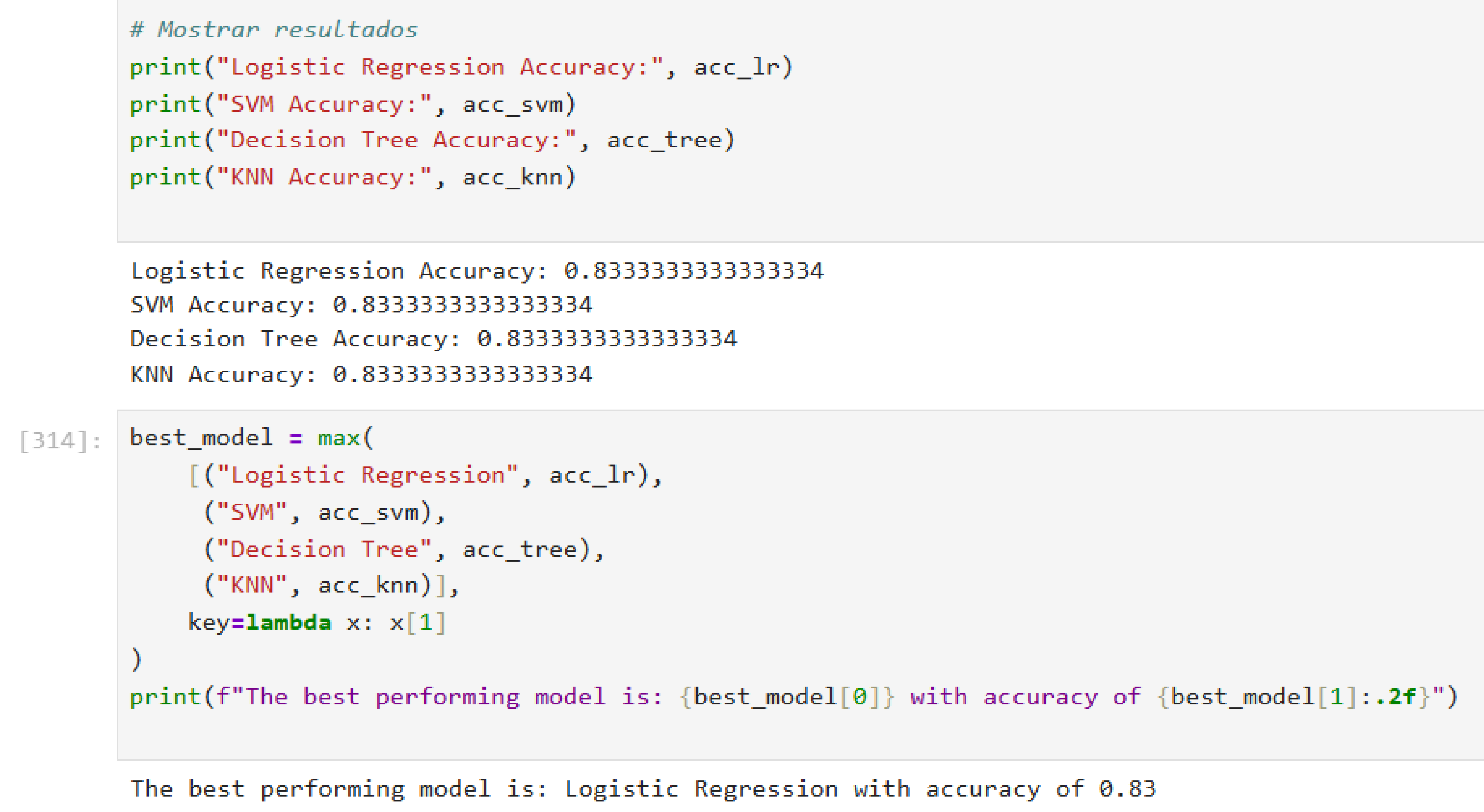
<Folium Map Screenshot 3>

CCAFS
SLC-
40
CCAFS
LC-
40
Space Launch
Complex 40
Centaur Road
Samuel C Phillip

Section 5

Predictive Analysis (Classification)

Classification Accuracy



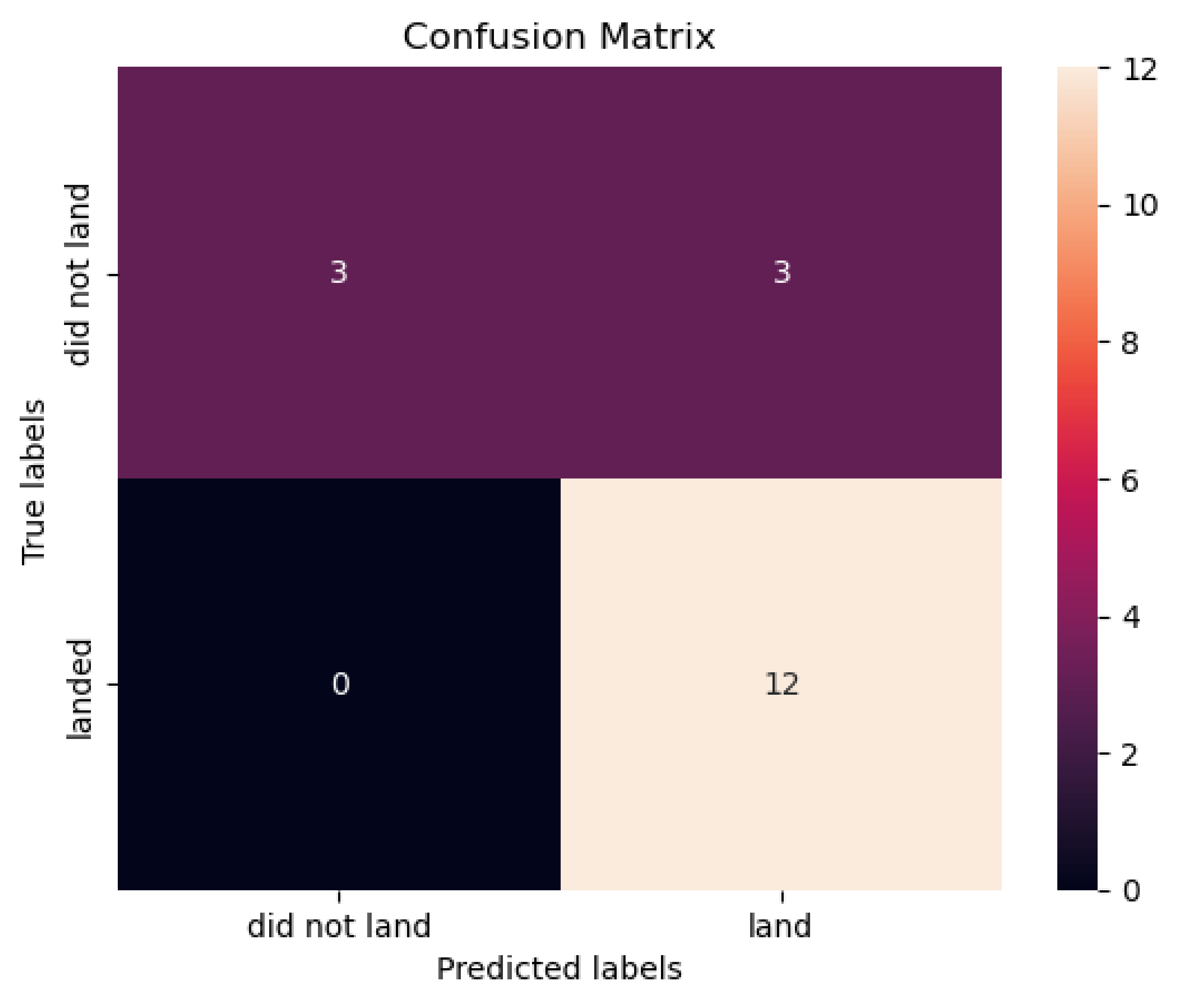
```
# Mostrar resultados
print("Logistic Regression Accuracy:", acc_lr)
print("SVM Accuracy:", acc_svm)
print("Decision Tree Accuracy:", acc_tree)
print("KNN Accuracy:", acc_knn)
```

```
Logistic Regression Accuracy: 0.8333333333333334
SVM Accuracy: 0.8333333333333334
Decision Tree Accuracy: 0.8333333333333334
KNN Accuracy: 0.8333333333333334
```

```
best_model = max(
    [("Logistic Regression", acc_lr),
     ("SVM", acc_svm),
     ("Decision Tree", acc_tree),
     ("KNN", acc_knn)],
    key=lambda x: x[1]
)
print(f"The best performing model is: {best_model[0]} with accuracy of {best_model[1]:.2f}")
```

```
The best performing model is: Logistic Regression with accuracy of 0.83
```

Confusion Matrix



Conclusions

- Launch Success Rate Increased Over TimeFrom 2013 onward, SpaceX significantly improved its launch success rate, with a notable upward trend in successful missions year by year.
- LEO Orbit Shows Highest Success CorrelationLaunches targeting Low Earth Orbit (LEO) demonstrate a higher success rate, possibly due to simpler mission profiles compared to more complex orbits like GTO.
- Launch Site Influences Success RateAmong the different launch sites, VAFB SLC-4E had a slightly lower success rate compared to KSC LC-39A and CCAFS LC-40, suggesting environmental or logistical differences may impact outcomes.
- Payload Mass Has Limited Impact on SuccessThe payload mass alone does not strongly determine mission success; both heavy and light payloads have shown successful and failed outcomes depending on other variables.

Thank you!