Winning Space Race with Data Science

Richard Knights
18/5/2022
rknightscode/DS_Captstone [Github]

Outline

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

Executive Summary

- Collected data from the public SpaceX API as well as the Wikipedia page for Space X. Labels were created to classify successful landings (by class 1).
- Subsequently explored and analyzed data using SQL. Incorporating visualization, folium maps and various dashboards.
- Categorical fields, containing features, were converted to binary via one hot encoding.
- Standardized data. Then utilized GridSearchVB to establish the optimum parameters for machine learning mods.
- Machine Learning models used were: Logistic Regression, Support Vector Machine, Decision Tree Classifier and K Nearest Neighbours.
- Scores are visualized but were relatively similar accuracy across all models giving high accuracy for each at ~ 83.33%.
- Recommend further data is needed to better train our models from the larger sample size.

Introduction

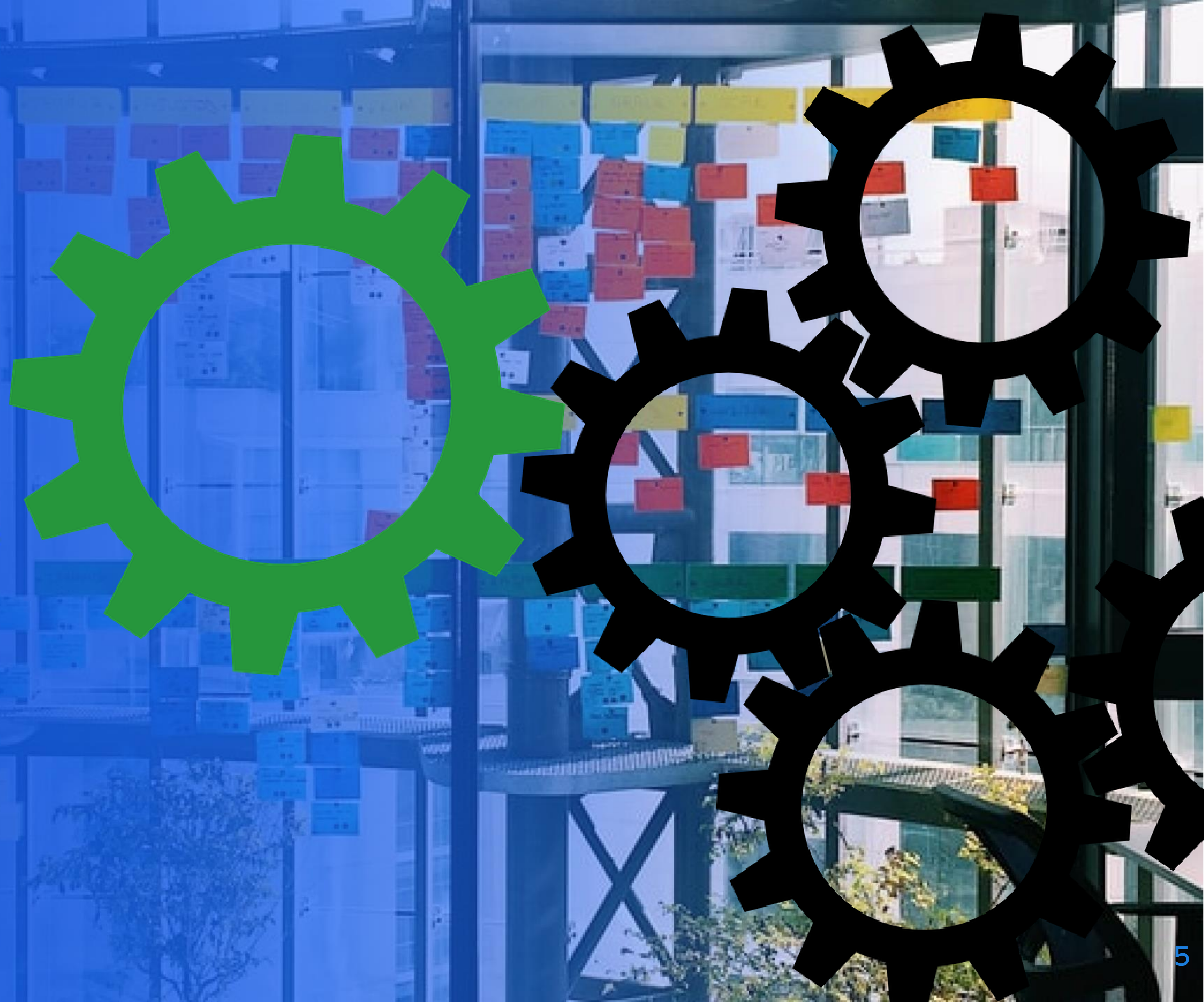
- The commercial space age is here.
- Companies are making space travel affordable for everyone
- Space X is one of the most successful.
- Space X produces relatively inexpensive rockets.
  This is as a result of reusing the first stage.
- Space Y wants to compete with Space X.

Problem:

- Objective from Space Y: Utilize machine learning to predict successful Stage 1 recovery.

Section 1

Methodology

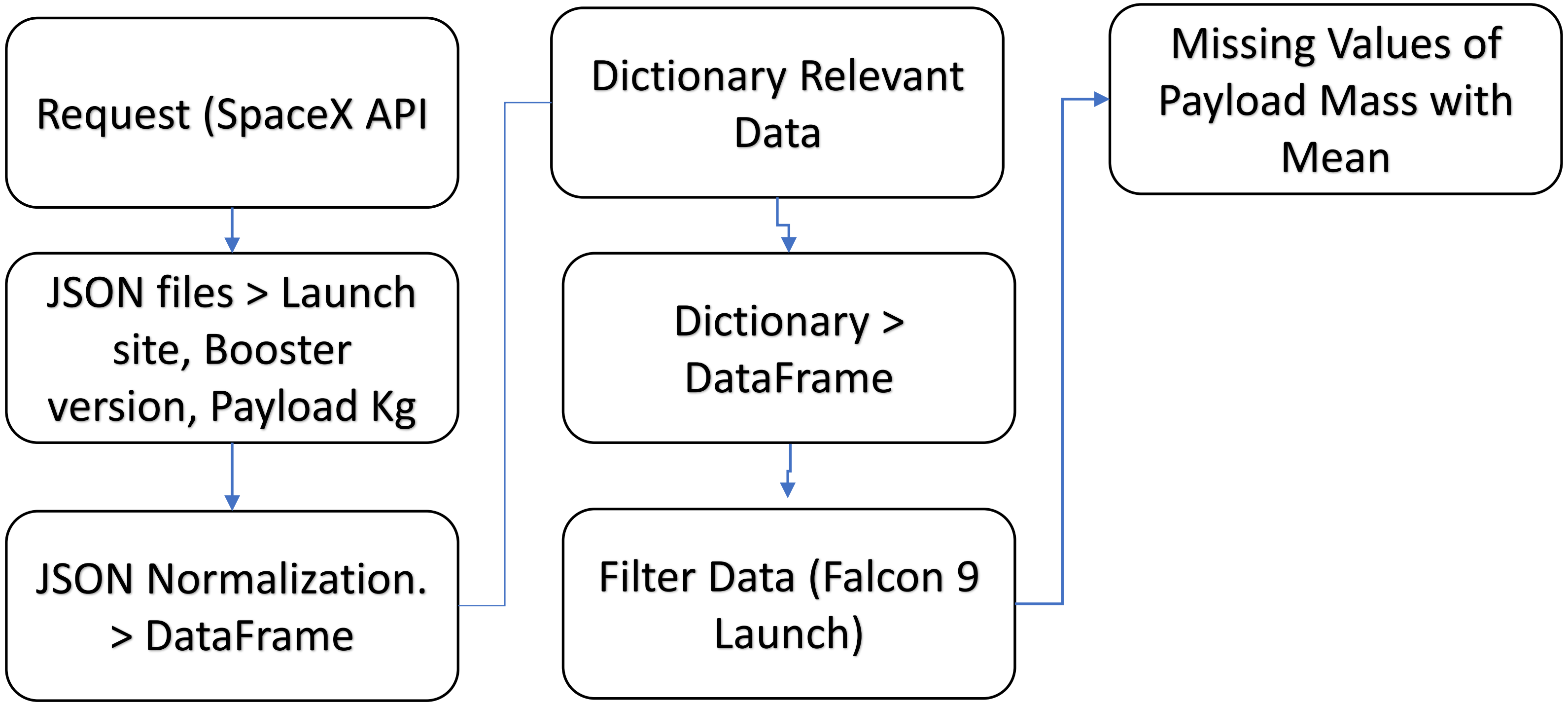


Methodology

Summary of Methodolgy

- Data collection methodology:
    - Combined data from both:
        - Space X using the public API.
        - Space X Wikipedia page using web scraping and the beautiful soup python package
- Perform data wrangling
    - Classified landings, as successful versus unsuccessful.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models (Logistic Regression, SVM, Decision Tree, and K Nearest Neighbor.)

Data Collection: SpaceX API



Request (SpaceX API

JSON files > Launch site, Booster version, Payload Kg

JSON Normalization. > DataFrame

Dictionary Relevant Data

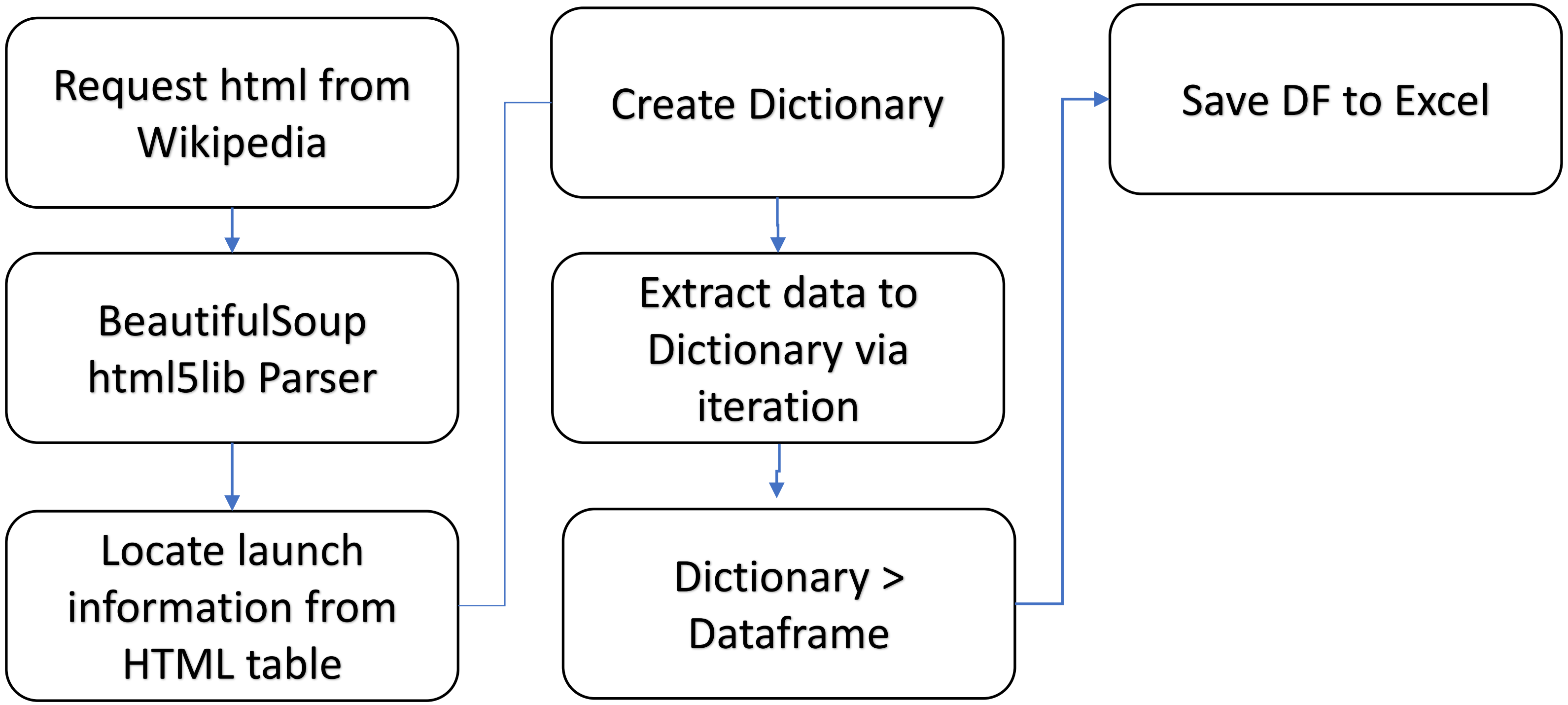
Dictionary > DataFrame

Filter Data (Falcon 9 Launch)

Missing Values of Payload Mass with Mean

- DS_Captstone/Data Collection API.ipynb at main · rknightscode/DS_Captstone (github.com)

Data Collection: WebScraping



Request html from Wikipedia

BeautifulSoup html5lib Parser

Locate launch information from HTML table

Create Dictionary

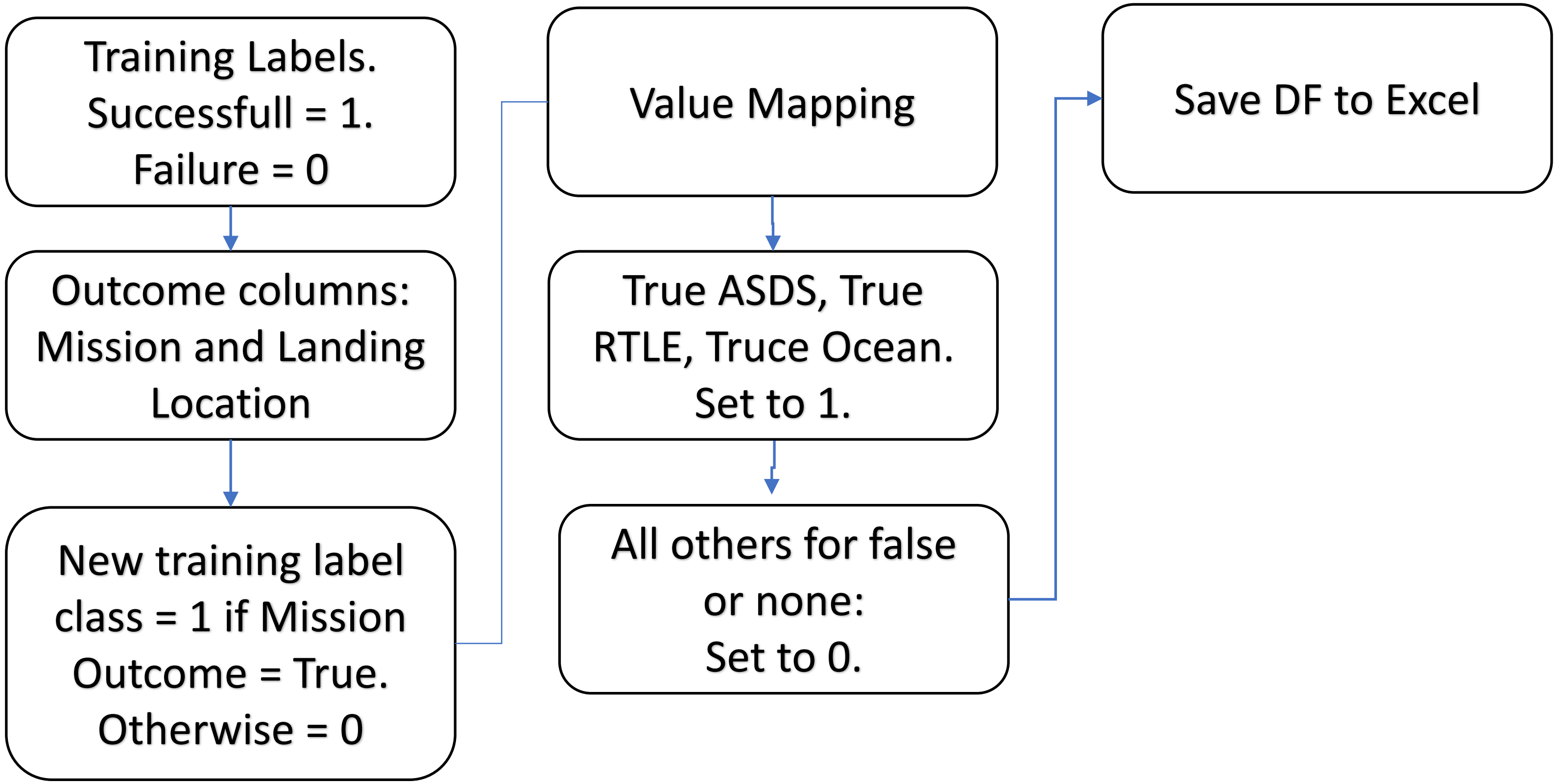
Extract data to Dictionary via iteration

Dictionary > Dataframe

Save DF to Excel

- DS_Captstone/Data Collection with Web Scraping.ipynb at main · rknightscode/DS_Captstone (github.com)

Data Collection: Data Wrangling



Training Labels. Successfull = 1. Failure = 0

Outcome columns: Mission and Landing Location

New training label class = 1 if Mission Outcome = True. Otherwise = 0

Value Mapping

True ASDS, True RTLE, Truce Ocean. Set to 1.

All others for false or none: Set to 0.

Save DF to Excel

- DS_Captstone/EDA.ipynb at main · rknightscode/DS_Captstone (github.com)

EDA with Data Visualization

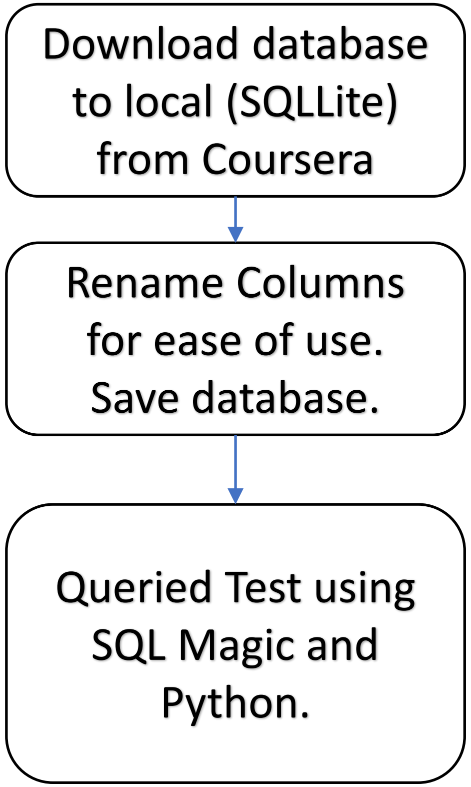
- EDA performed on the following variables: Flight Number, Payload Mass, Launch Site, Orbit, Class, and Year

Plots Utilized:

- Flight Number versus Payload Mass
- Flight Number versus Launch Site
- Payload Mass versus Launch Site
- Orbit versus Success rate.
- Flight Number versus Orbit
- Payload versus Orbit
- Success Yearly Trend

- DS_Captstone/eda-dataviz.ipynb at main · rknightscode/DS_Captstone (github.com)

Data Collection: EDA with SQL


Download database to local (SQLLite) from Coursera
Rename Columns for ease of use. Save database.
Queried Test using SQL Magic and Python.

Queries Made:

- Filter unique launch sites names.
- Total payload mass carried by boosters by NASA.
- Average payload mass carried by booster F9 v1.1
- Names of boosters for a drone ship at various payloads.
- Total numbers for Mission outcomes
- Rank of Successful landing outcomes in a date range.

- DS_Captstone/eda-sql-coursera_sqllite.ipynb at main · rknightscode/DS_Captstone (github.com)

Interactive Map with Folium

Queries Made:

- Maps created to mark Launch sites used.
- Successful and Unsuccessful landings.

This was to determine, and filter for:

  - Proximity to important infrastructure as well as key locations including coastline and nearest large city.
  - The aim of this was to understand why launch sites are located. E.g Safety. Large population to attract key workers.

- DS_Captstone/folium_launch_site_location.ipynb at main · rknightscode/DS_Captstone (github.com)

Interactive Map with: Plotly Dash

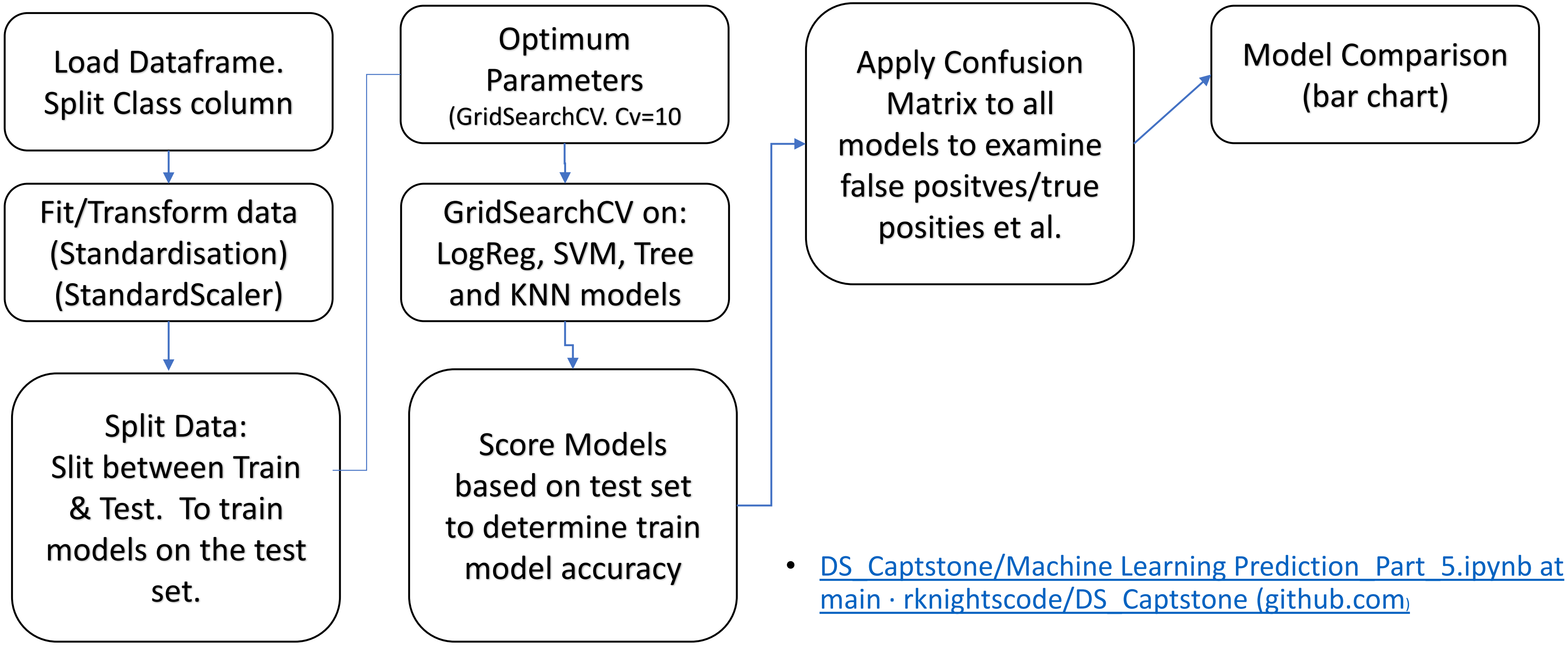
Dashboard created for:

- Pie Chart.
  Launch site success rate.
- Scatter Plot
  Two inputs include:
  Launch site selection.
  Payload Mass slider(0-10k kg).

Purpose:
To visualize success filtering against launch sites, payload mass, and booster type.

- DS_Captstone/Dash_Space.py at main · rknightscode/DS_Captstone (github.com)

Classification (Predictive Analysis)



Load Dataframe. Split Class column

Fit/Transform data (Standardisation) (StandardScaler)

Split Data: Slit between Train & Test. To train models on the test set.

Optimum Parameters (GridSearchCV. Cv=10

GridSearchCV on: LogReg, SVM, Tree and KNN models

Score Models based on test set to determine train model accuracy

Apply Confusion Matrix to all models to examine false positves/true posities et al.

Model Comparison (bar chart)

- DS_Captstone/Machine Learning Prediction_Part_5.ipynb at main · rknightscode/DS_Captstone (github.com)

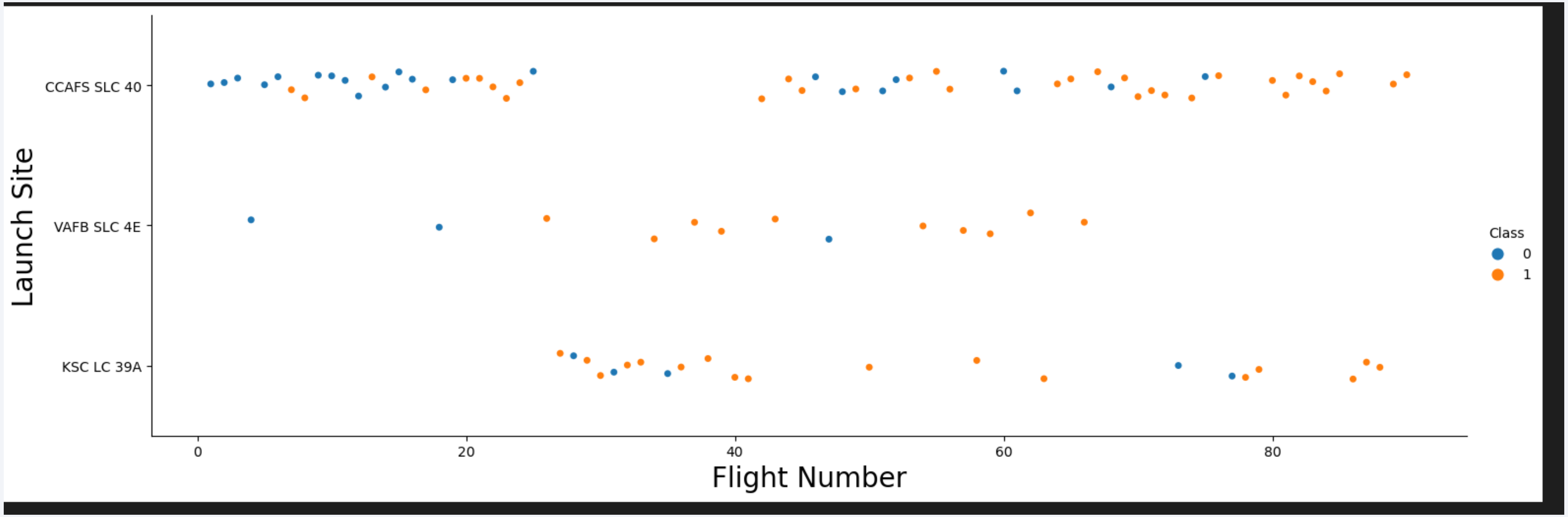
Results

- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

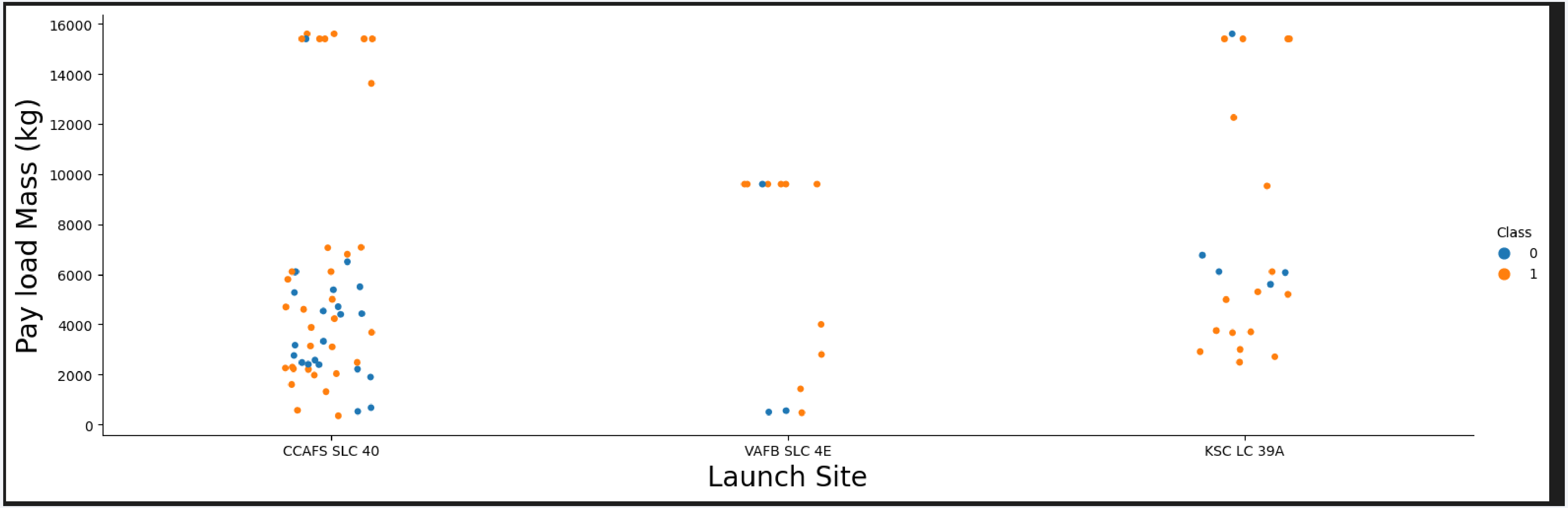
Flight Number vs. Launch Site



- Legend:
  Class 1 = Success. Class 2 = Unsuccessful.

- Graphic suggests an increase over time in the number of launches as the 2nd and 3rd sites become active at ~20 flight number.
  CCAFS appears to be the favoured launch site in frequency of launches. Possibly from access to key inferstructure.

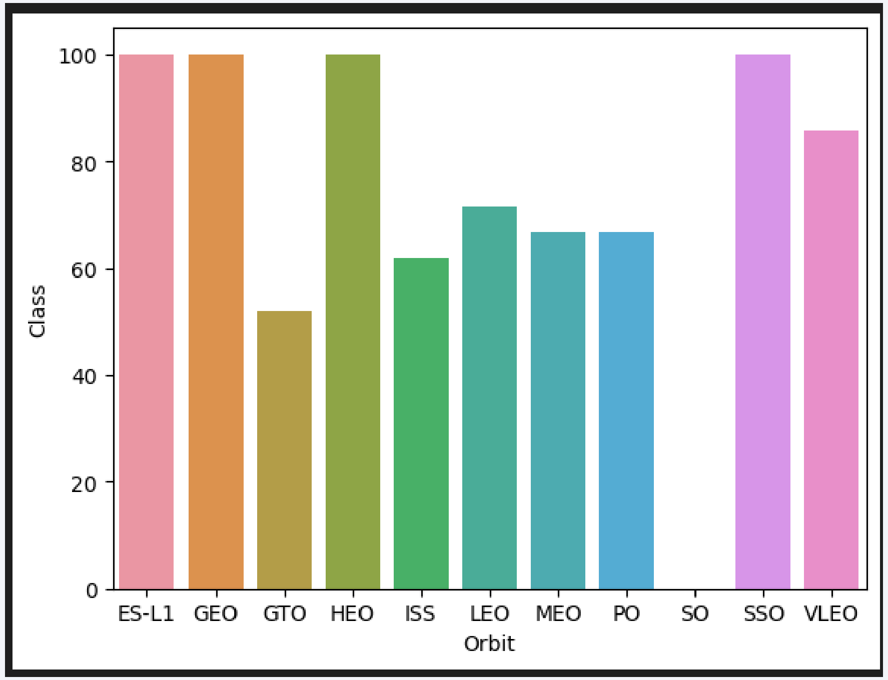
Payload vs. Launch Site



- Legend:
  Class 1 = Success. Class 2 = Unsuccessful.

- Payload mass mostly between 0-8000kg.
  Heavier launches most often at CCAFS, then at KSC LC29A. Possibly due to limitations of infrastructure.

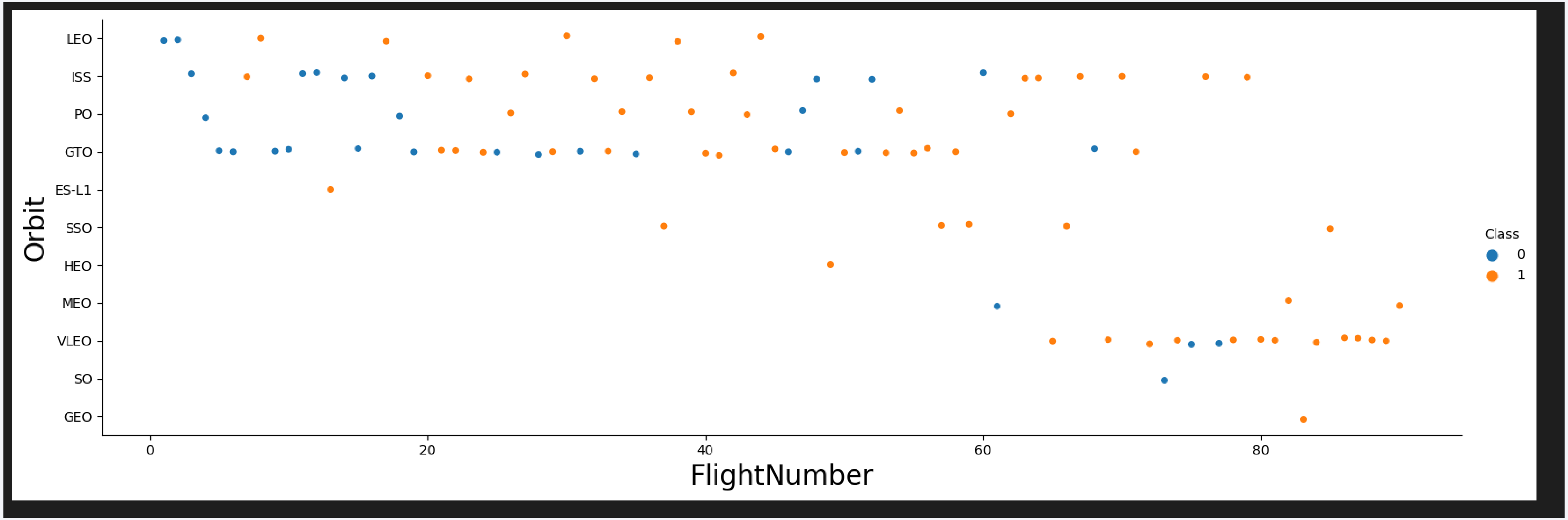
Success Rate vs. Orbit Type



- Legend:
  Class 100 = Most Successful.

  SO: 0% success rate.
  ES-L1, GEO, HEO, SSO, VLEO most successful.
  GTO, ISS, LEO, MEO, PO. Variable, partial success.

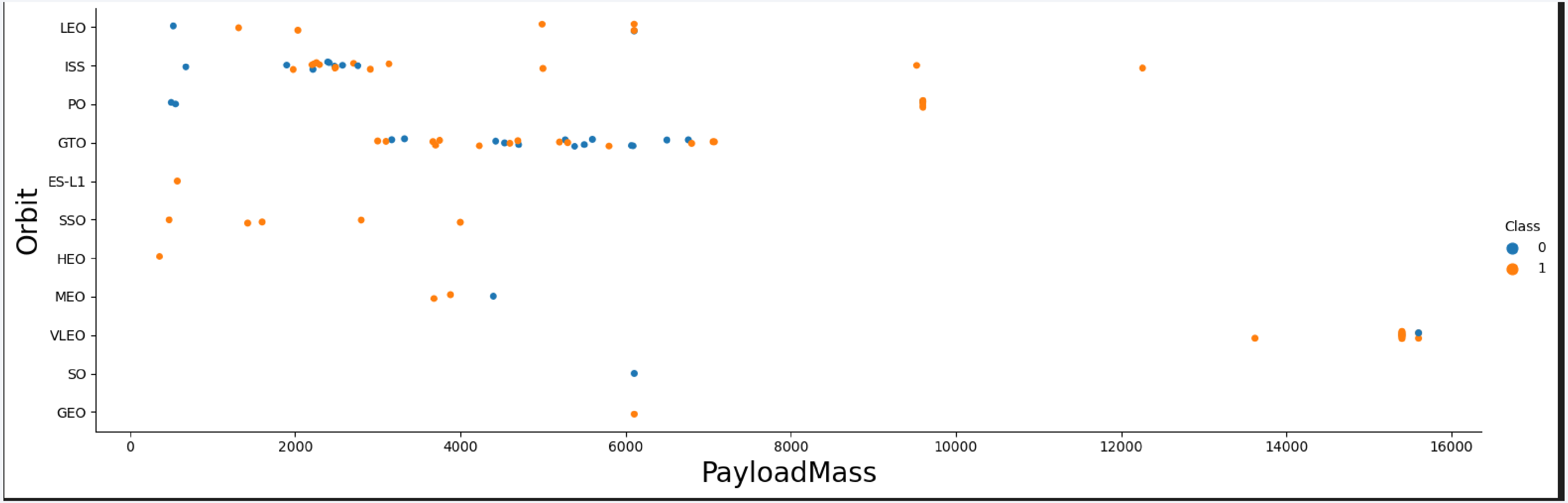
Flight Number vs. Orbit Type



- Legend:
  Class 1 = Success. Class 2 = Unsuccessful.

  Trend of orbit preference changing as flight numbers progressed.
  Space X appears to perform better in lower, or sun synchronous orbits.
  Mixed results from 0-20 results. Mostly failures.

Payload vs. Orbit Type

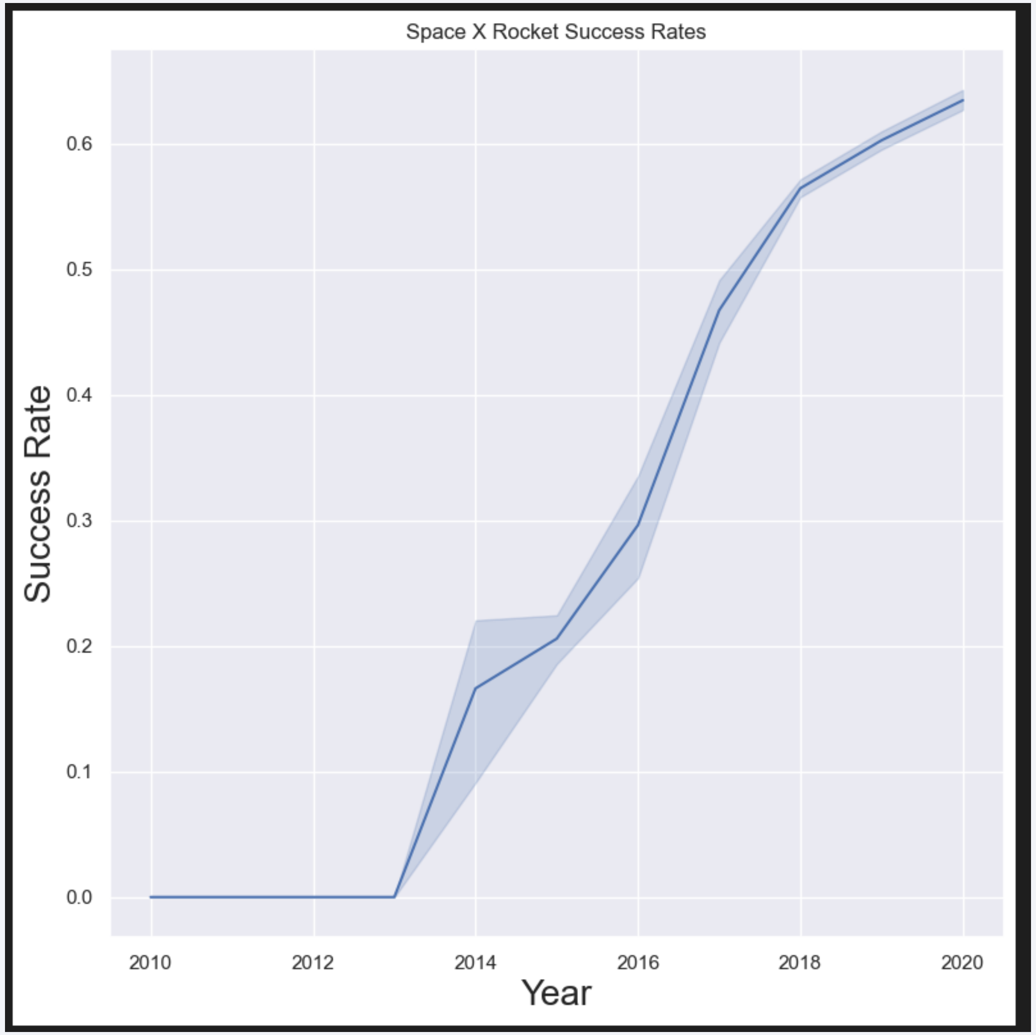


- Legend:
  Class 1 = Success. Class 2 = Unsuccessful.

  Unsuccessful payload mass seems to be frequent ~2000kg on ISS missions, and at 4-7000 Kg-GTO missions.
  This may also be due to frequency of those missions.
  ES-LQ, SSO and HEO appear to be most successful.
  Majority of missions have a Payload Mass up to ~7000kg

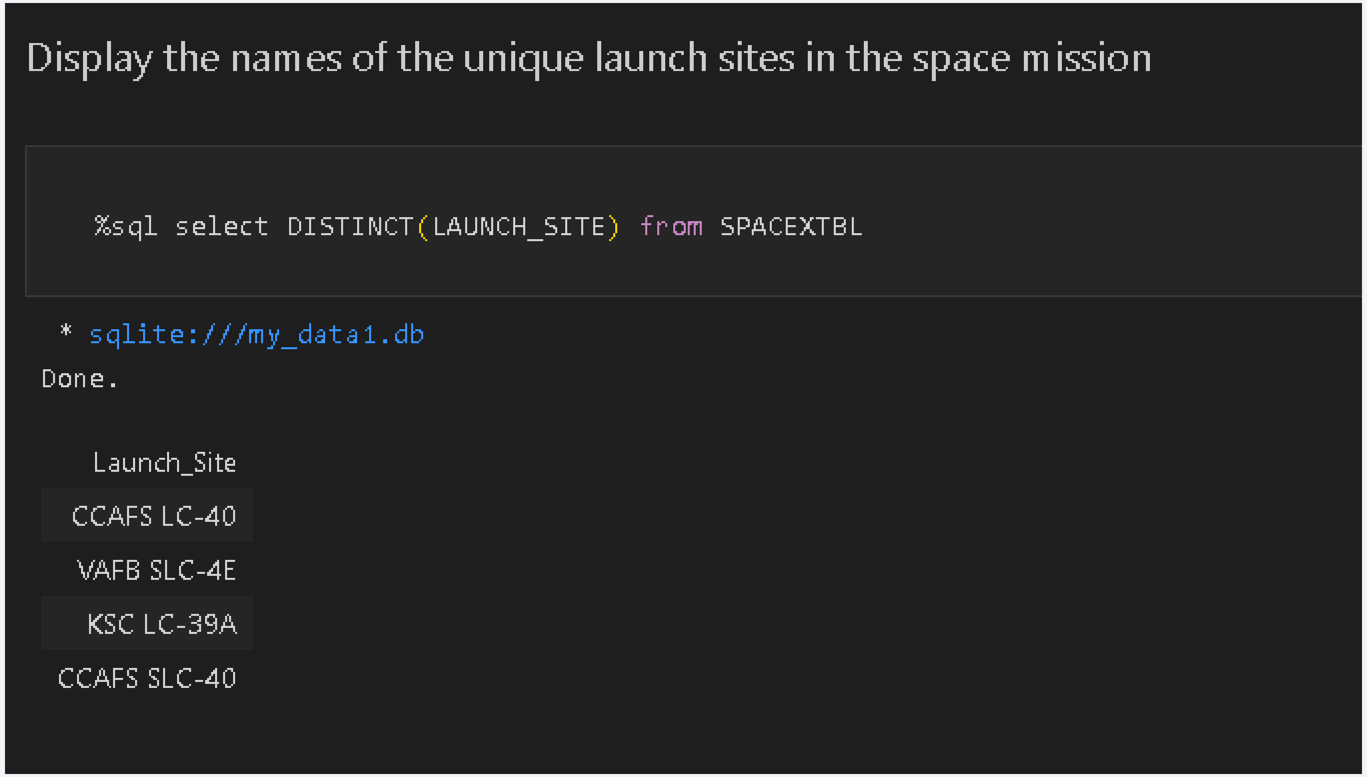
Launch Success Yearly Trend

- Success rate improves over time.
- Success rate relatively constant (angle of the line).
- Rate of success improvement seems to be declining somewhat towards 0.7 success.
- Suggest further, mode recent data is collected.



All Launch Site Names

- Query the unique launch sites form the database
- Variants of CCAFS may be from the same launch site but variants of the launch pad.
- Therefore, there may be only 3 unique names, CCAFS, VAFB and KSC.

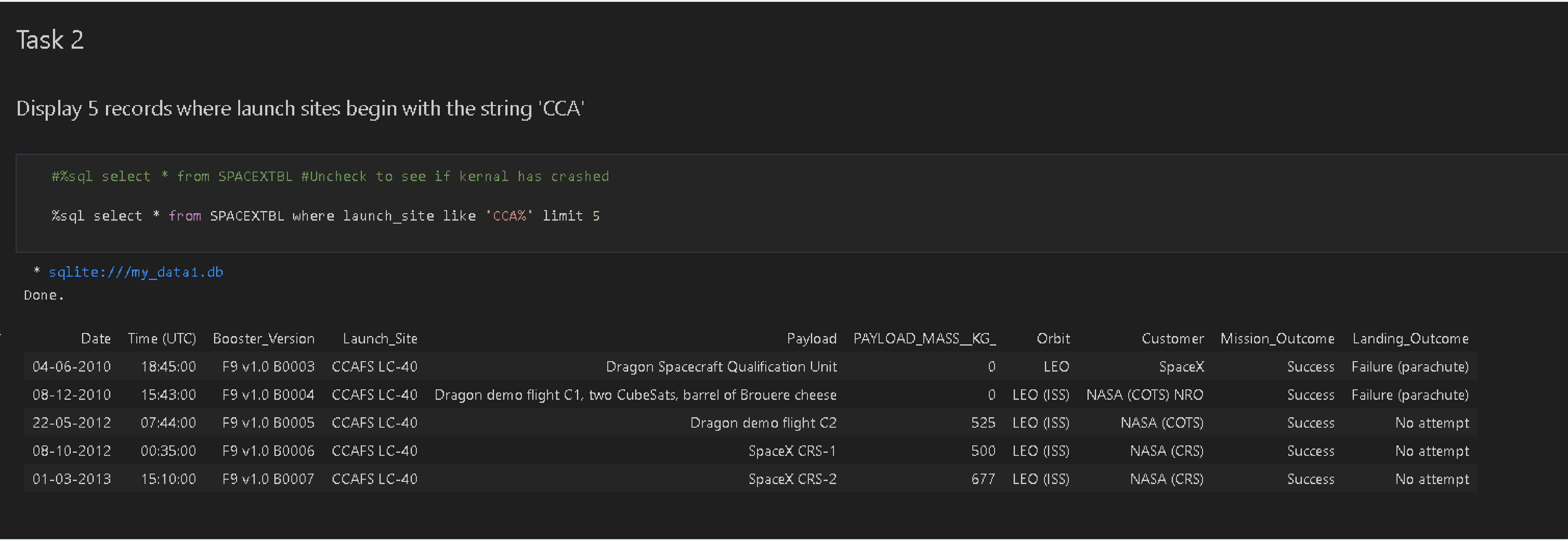


Display the names of the unique launch sites in the space mission

```
%sql select DISTINCT(LAUNCH_SITE) from SPACEXTBL
```

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

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

Launch Site Names Begin with 'CCA'



Task 2

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

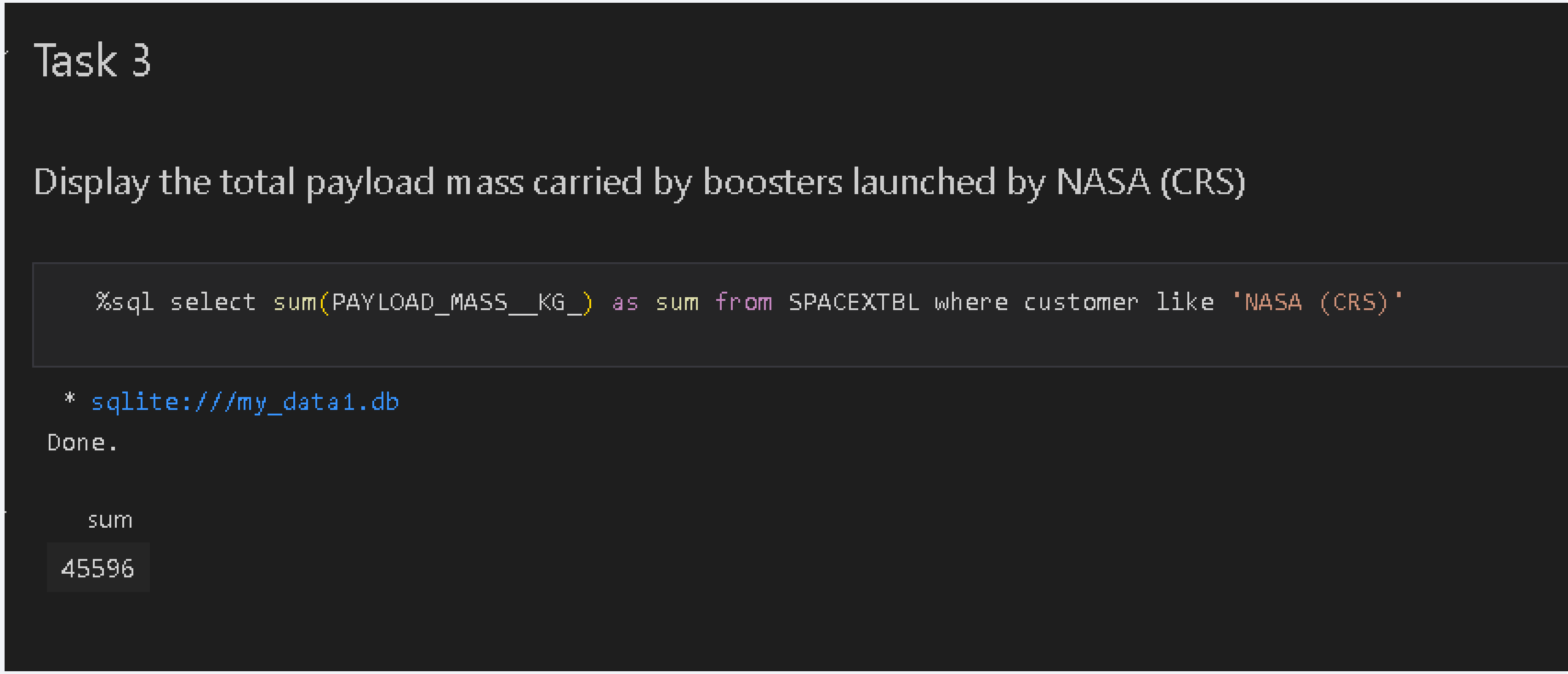
```
#%sql select * from SPACEXTBL #Uncheck to see if kernal has crashed

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

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
08-12-2010	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)
22-05-2012	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
01-03-2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

- Query sums total payload mass launched by NASA.



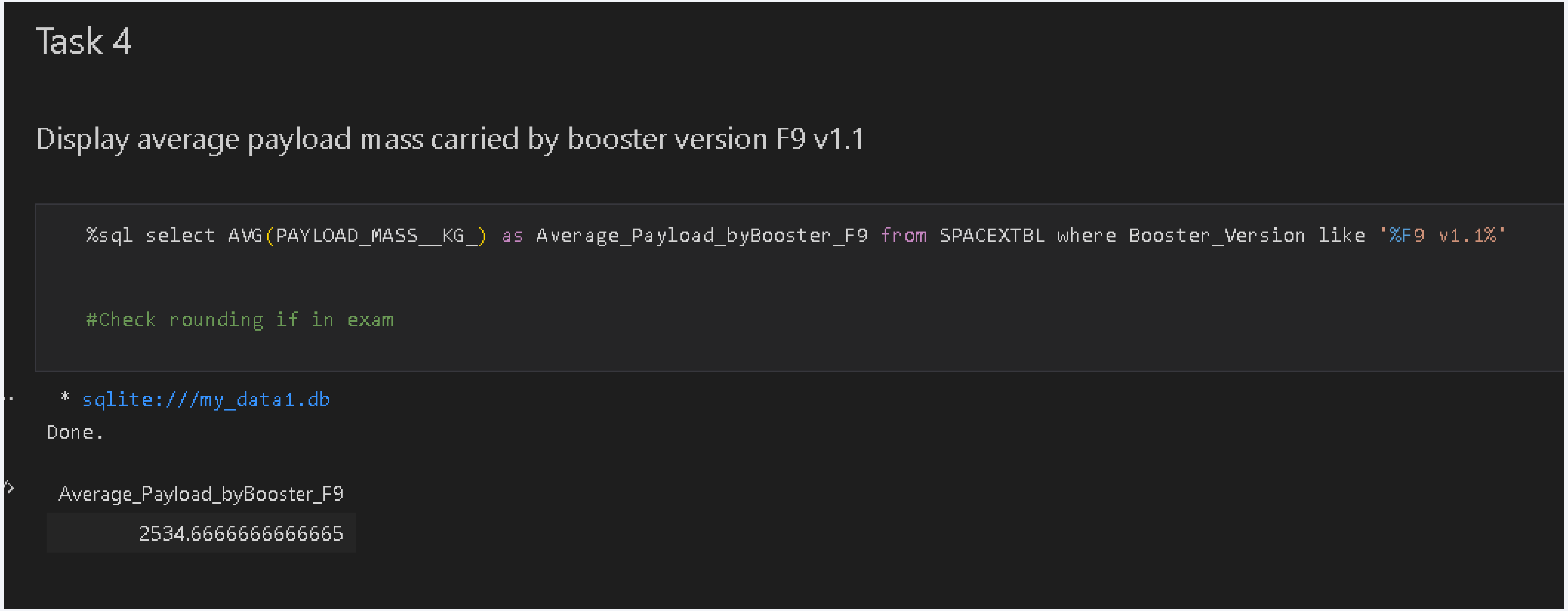
Task 3

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

```
%sql select sum(PAYLOAD_MASS__KG_) as sum from SPACEXTBL where customer like 'NASA (CRS)'
```

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

   sum
 45596
```

Average Payload Mass by F9 v1.1



Task 4

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

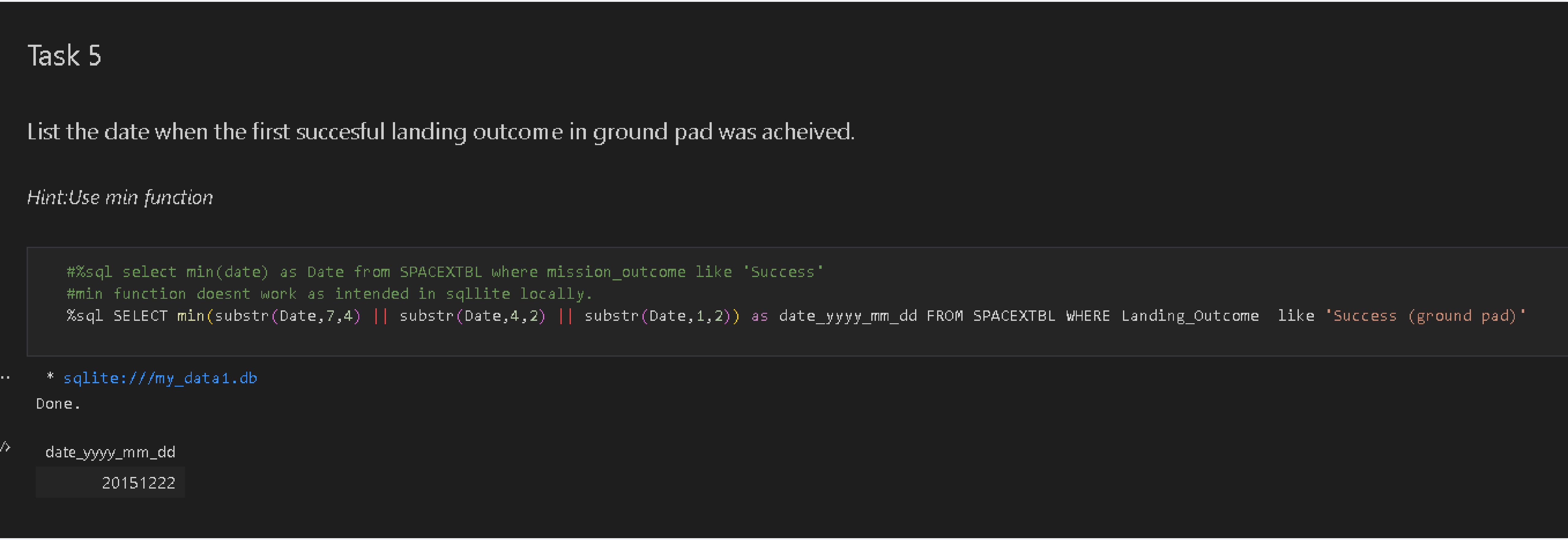
```
%sql select AVG(PAYLOAD_MASS__KG_) as Average_Payload_byBooster_F9 from SPACEXTBL where Booster_Version like '%F9 v1.1%'

#Check rounding if in exam
```

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

Average_Payload_byBooster_F9
2534.6666666666665

Average payload is towards the lower end of the data collected.

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

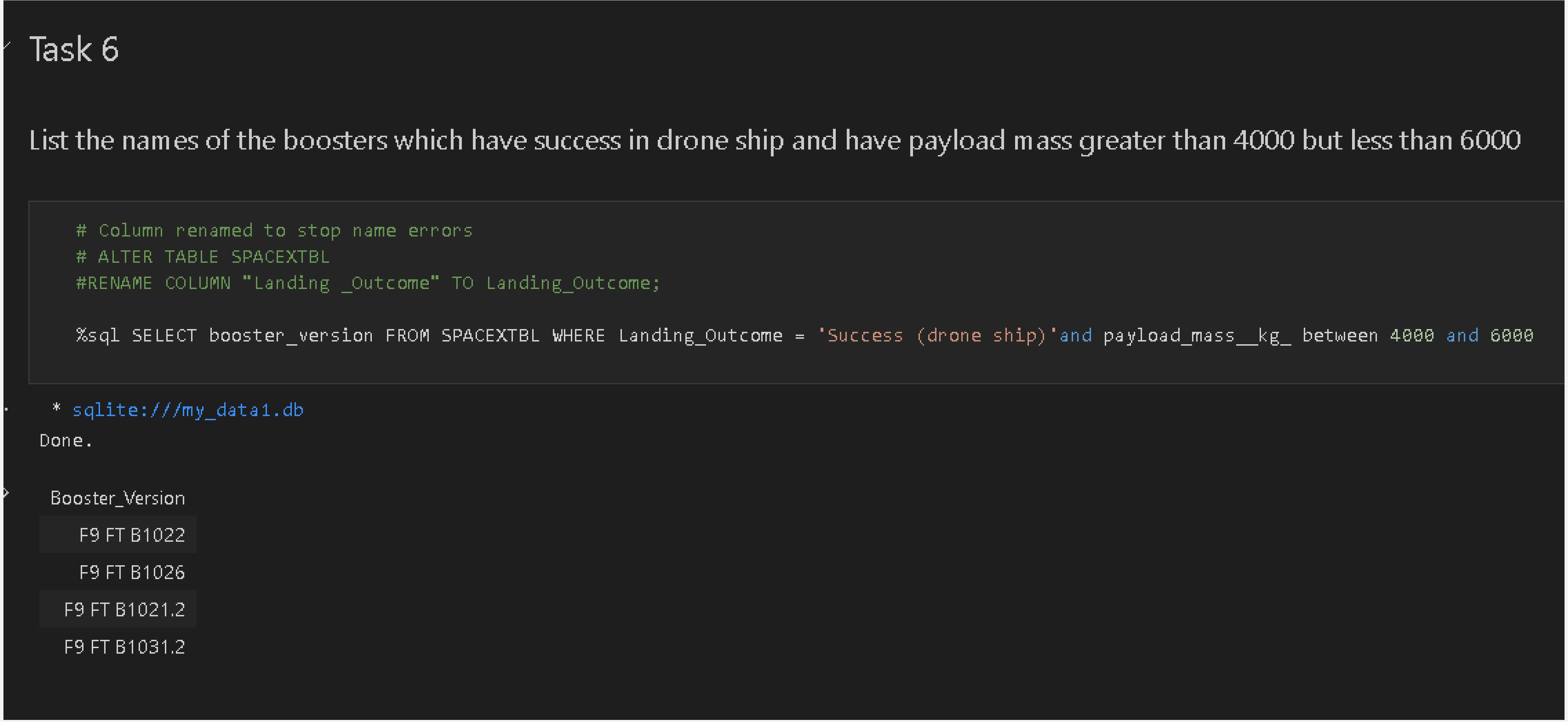
```
#%sql select min(date) as Date from SPACEXTBL where mission_outcome like 'Success'
#min function doesnt work as intended in sqllite locally.
%sql SELECT min(substr(Date,7,4) || substr(Date,4,2) || substr(Date,1,2)) as date_yyyy_mm_dd FROM SPACEXTBL WHERE Landing_Outcome  like 'Success (ground pad)'
```

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

date_yyyy_mm_dd
20151222

First launch did not appear until around ~2014. First ground pad landing wasn't until the end of ~2015. It took possibly up to 1-2 years to achieve a successful landing.

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

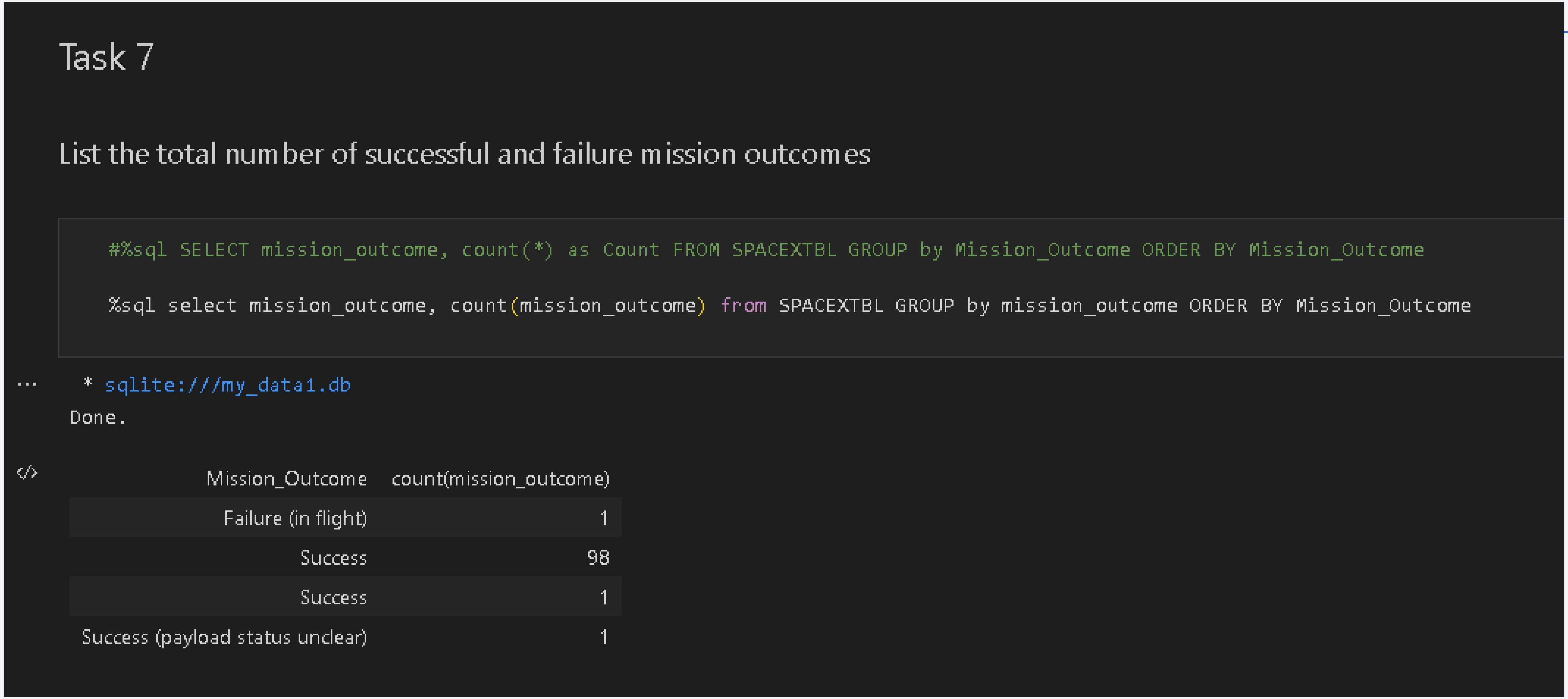
```
# Column renamed to stop name errors
# ALTER TABLE SPACEXTBL
#RENAME COLUMN "Landing _Outcome" TO Landing_Outcome;

%sql SELECT booster_version FROM SPACEXTBL WHERE Landing_Outcome = 'Success (drone ship)'and payload_mass__kg_ between 4000 and 6000
```

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

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

Booster version type that are both successful and withing the stated mass.

Total Number of Successful and Failure Mission Outcomes



Task 7

List the total number of successful and failure mission outcomes

```
#%sql SELECT mission_outcome, count(*) as Count FROM SPACEXTBL GROUP by Mission_Outcome ORDER BY Mission_Outcome

%sql select mission_outcome, count(mission_outcome) from SPACEXTBL GROUP by mission_outcome ORDER BY Mission_Outcome
```

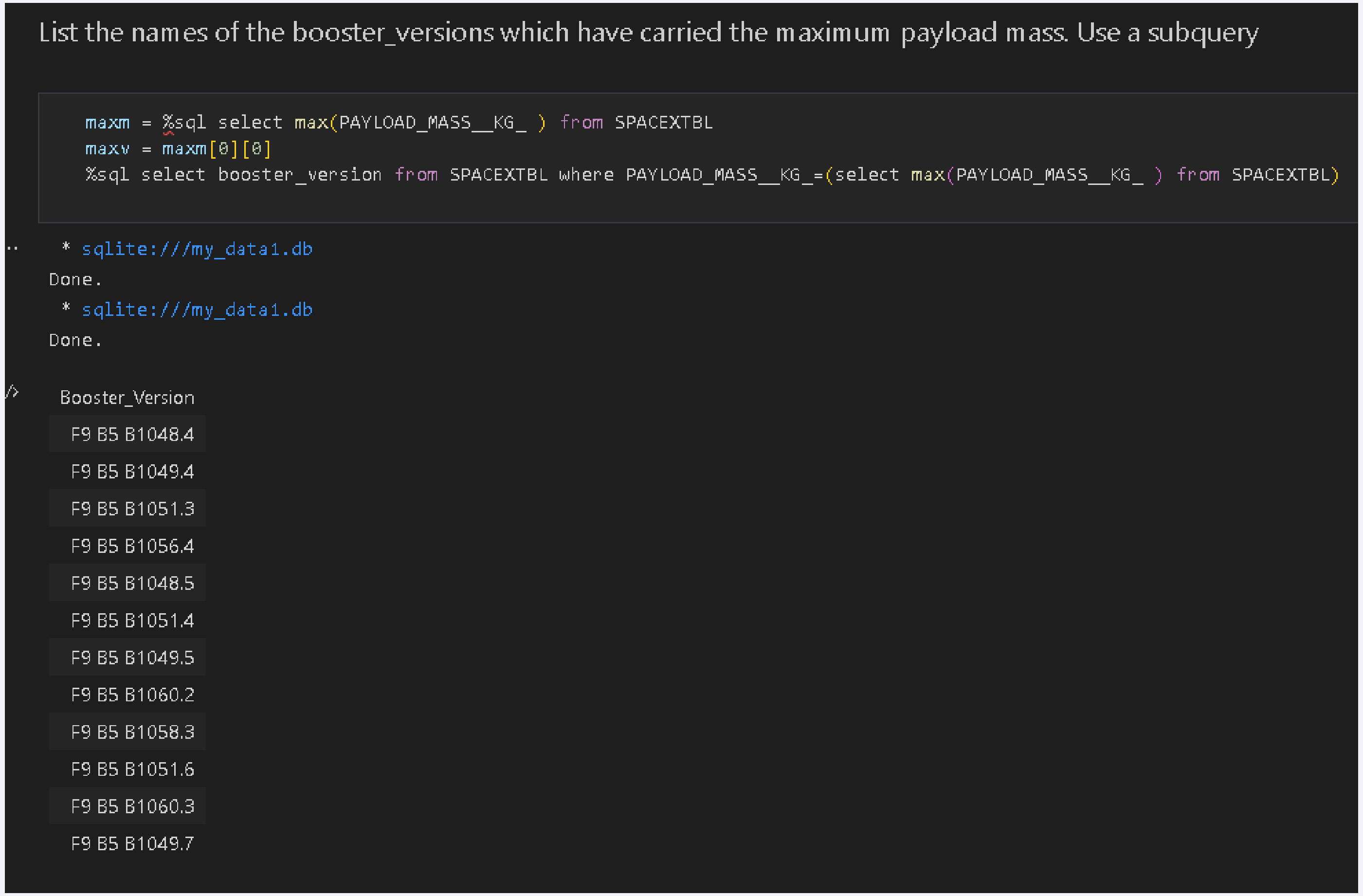
```
 * sqlite:///my_data1.db
Done.
```

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

- Success type may have a naming error if split between two mission outcomes.
- 99% rate . May suggest landing failure number manipulation.

Boosters Carried Maximum Payload

- Booster variants that carried the largest payload mass (15600kg).
- All appear to be variants, or evolution of the F9 B5 type.
- Correlates payload mass and booster version.



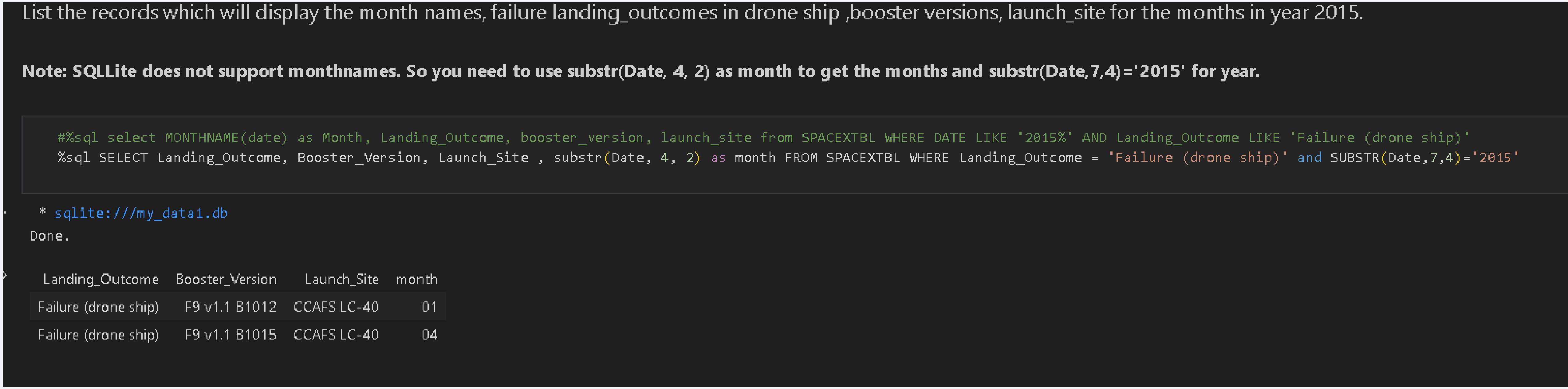
List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

```
maxm = %sql select max(PAYLOAD_MASS__KG_ ) from SPACEXTBL
maxv = maxm[0][0]
%sql select booster_version from SPACEXTBL where PAYLOAD_MASS__KG_=(select max(PAYLOAD_MASS__KG_ ) from SPACEXTBL)
```

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

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



List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.

Note: SQLLite does not support monthnames. So you need to use substr(Date, 4, 2) as month to get the months and substr(Date,7,4)='2015' for year.

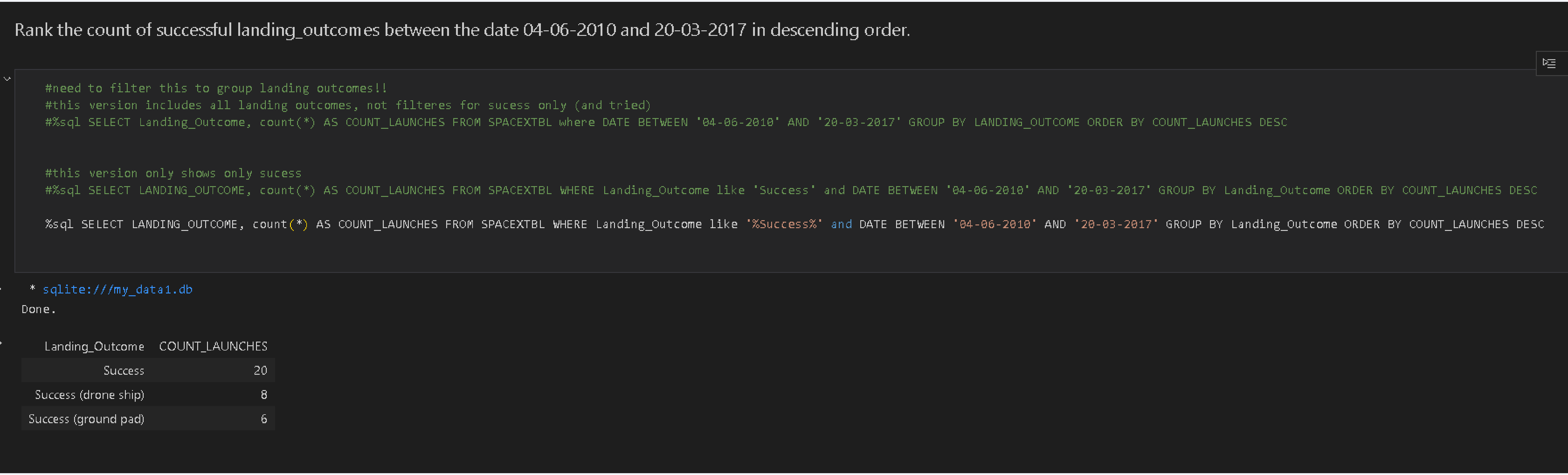
```
#%sql select MONTHNAME(date) as Month, Landing_Outcome, booster_version, launch_site from SPACEXTBL WHERE DATE LIKE '2015%' AND Landing_Outcome LIKE 'Failure (drone ship)'
%sql SELECT Landing_Outcome, Booster_Version, Launch_Site , substr(Date, 4, 2) as month FROM SPACEXTBL WHERE Landing_Outcome = 'Failure (drone ship)' and SUBSTR(Date,7,4)='2015'
```

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

Landing_Outcome	Booster_Version	Launch_Site	month
Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40	01
Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40	04

- Query returns month, landing outcome, booster version, and launch site. Filtered to 2015 where stage 1 failed to land. There were two variants – both from the same booster variant (F9 v1.1), and launch site.

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



Rank the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.

```
#need to filter this to group landing outcomes!!
#this version includes all landing outcomes, not filteres for sucess only (and tried)
#%sql SELECT Landing_Outcome, count(*) AS COUNT_LAUNCHES FROM SPACEXTBL where DATE BETWEEN '04-06-2010' AND '20-03-2017' GROUP BY LANDING_OUTCOME ORDER BY COUNT_LAUNCHES DESC


#this version only shows only sucess
#%sql SELECT LANDING_OUTCOME, count(*) AS COUNT_LAUNCHES FROM SPACEXTBL WHERE Landing_Outcome like 'Success' and DATE BETWEEN '04-06-2010' AND '20-03-2017' GROUP BY Landing_Outcome ORDER BY COUNT_LAUNCHES DESC

%sql SELECT LANDING_OUTCOME, count(*) AS COUNT_LAUNCHES FROM SPACEXTBL WHERE Landing_Outcome like '%Success%' and DATE BETWEEN '04-06-2010' AND '20-03-2017' GROUP BY Landing_Outcome ORDER BY COUNT_LAUNCHES DESC
```

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

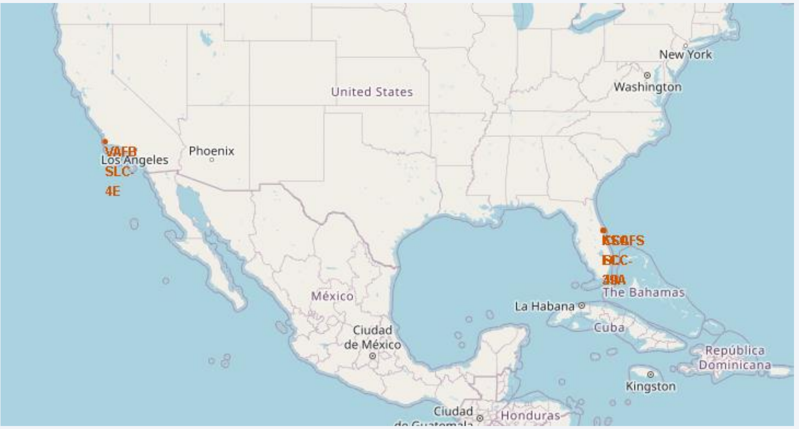
Landing_Outcome	COUNT_LAUNCHES
Success	20
Success (drone ship)	8
Success (ground pad)	6

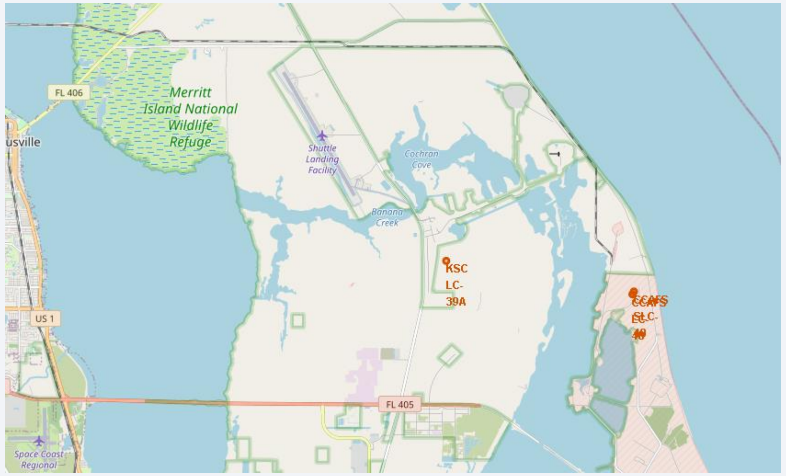
- Query returns a list of successful landings between the listed dates.
- There are three different outcomes, split between the drone and ground pad.

Section 3

Launch Sites Proximities Analysis

Launch Site Locations

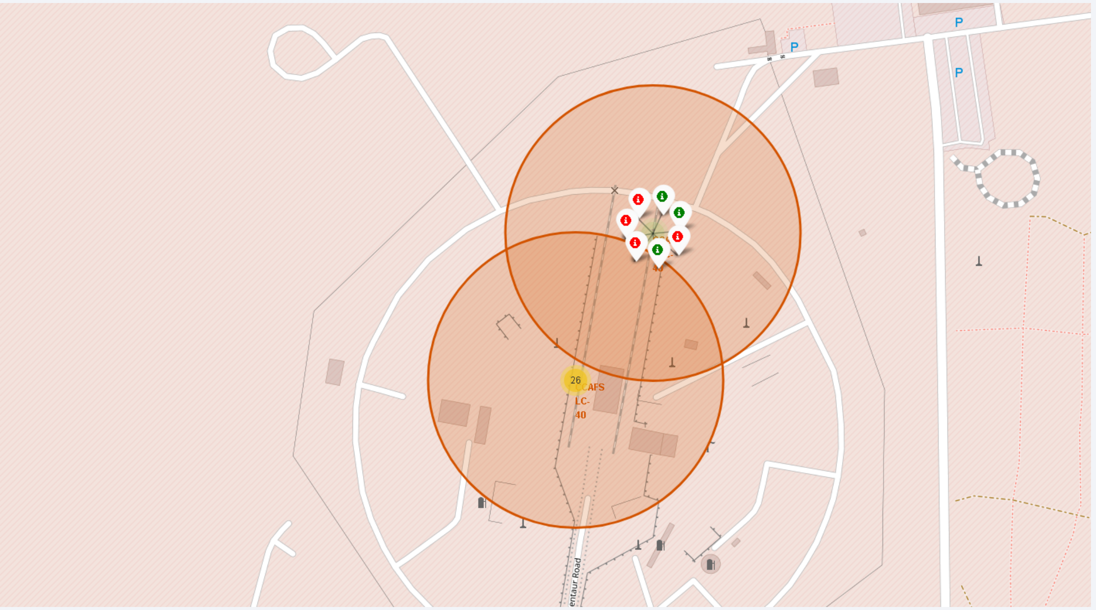




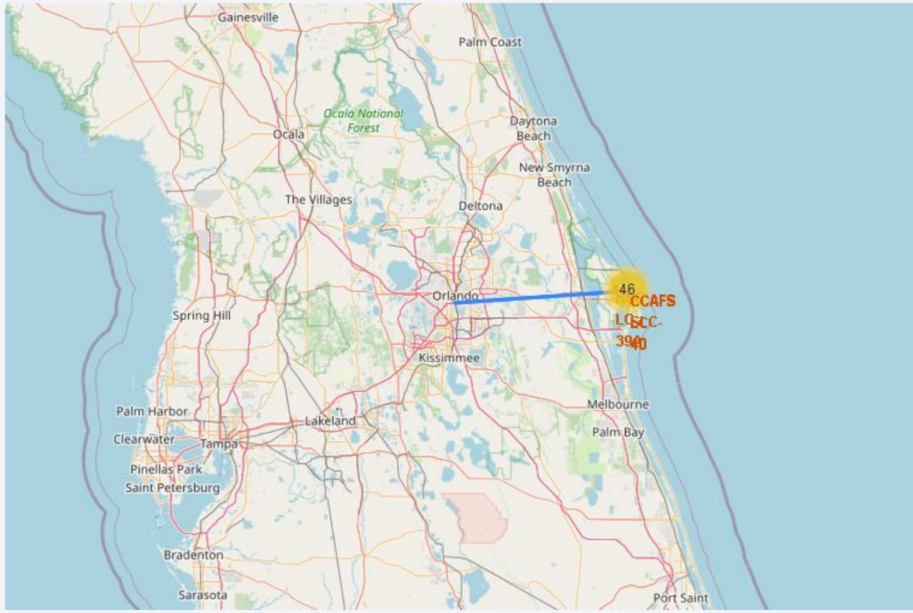
- West and East Coast Launch sites in the USA.
  The East Coast has two separate sites, located relatively close to each other.

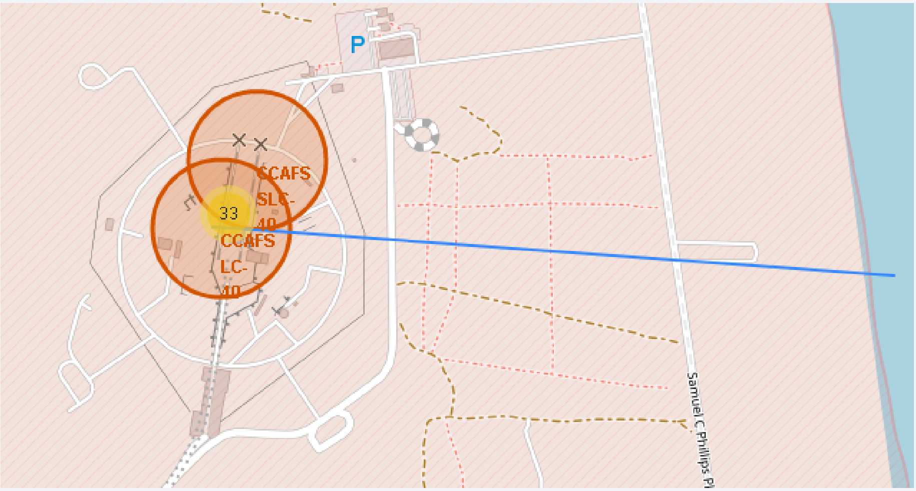
Color Coded Launch Markers

Color coded Cluster map of the launch site.
Green = Successful, and failed, unsuccessful = Red.
At this site, there are 3 successful and 4 failed missions.



Key Locations Map

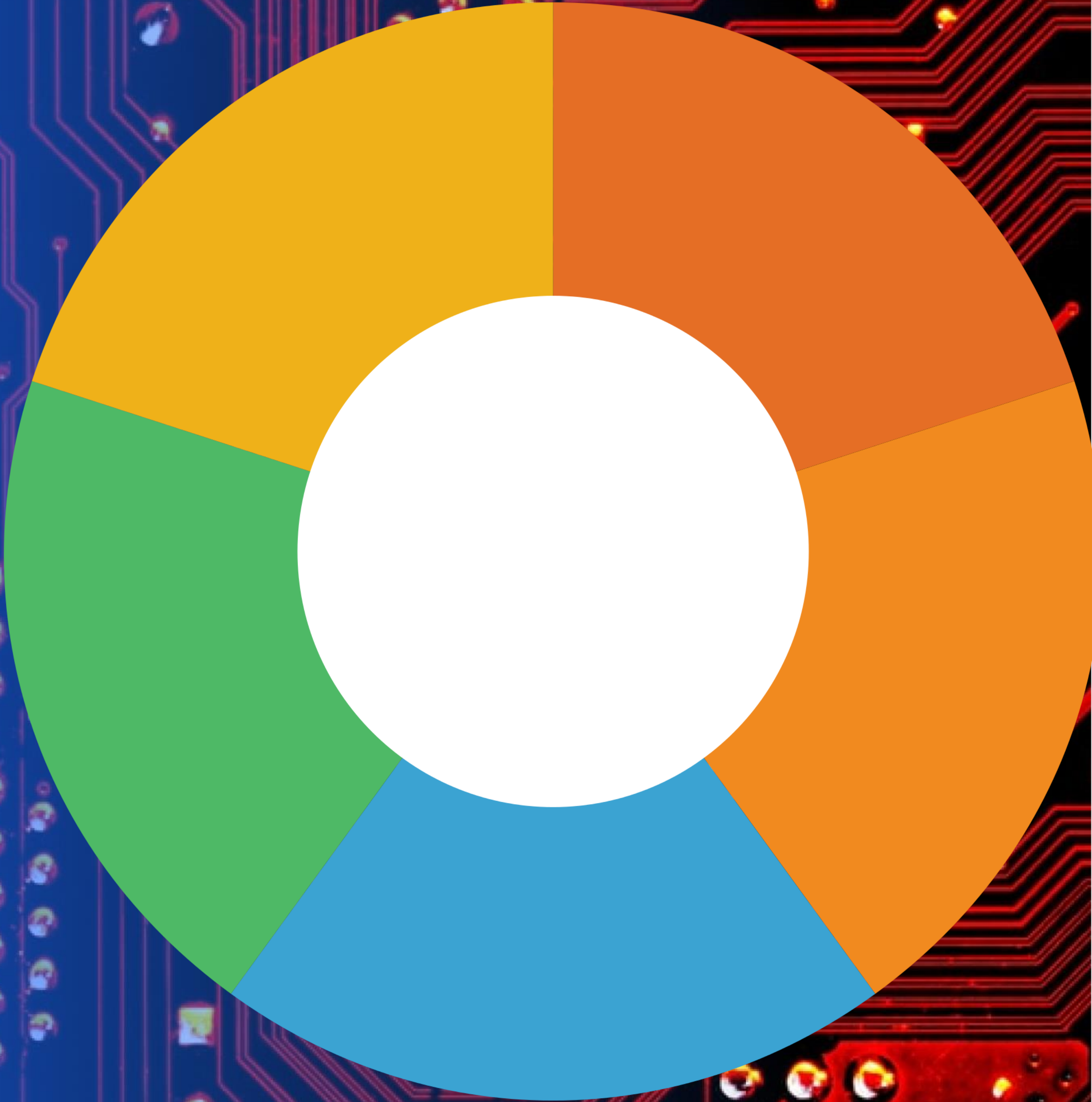




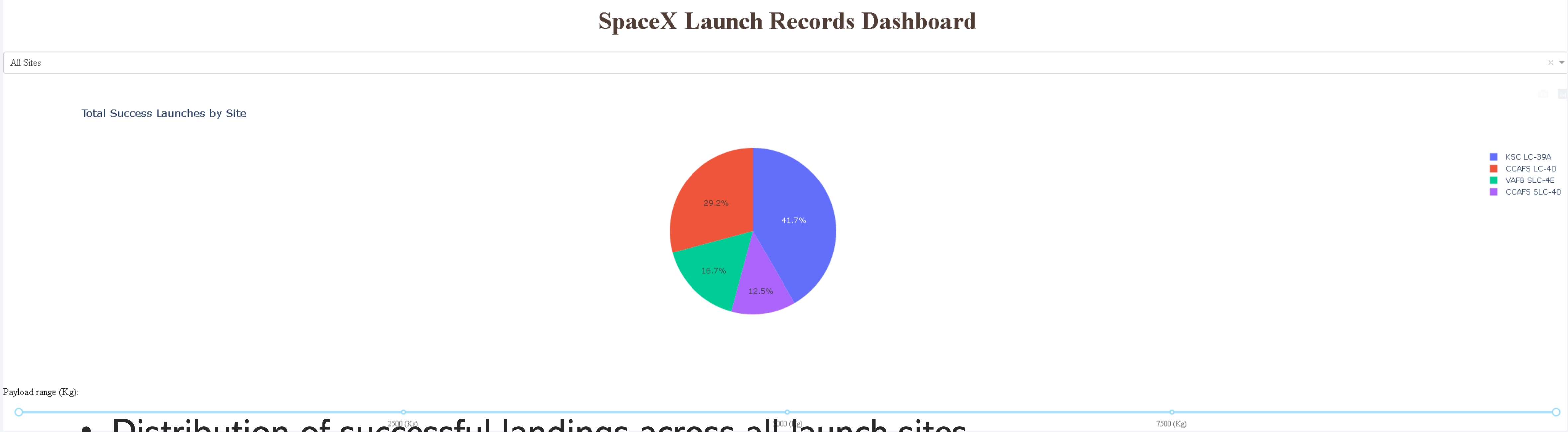
- Distance from the East Code launch site to the closest, largest city/metro area. This may be an advantage for key workers to be located in the general, commuting area.
- The image on the right displays the distance to the coastline in Florida from a launch site. This may be a safety precaution against the risk of failures and danger to the local population.

Section 4

Build a Dashboard with Plotly Dash



Launch success count for all sites:



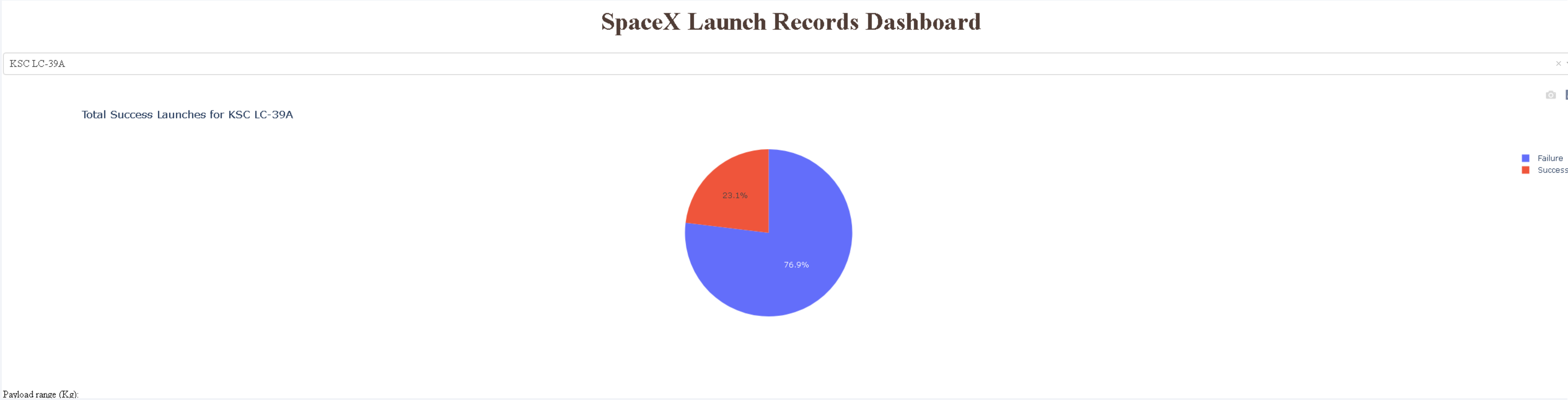
SpaceX Launch Records Dashboard

All Sites

Payload range (Kg):

- Distribution of successful landings across all launch sites.
- KSC appears to be the most successful.
- VAFS (after the CCAFS sites are amalgamated if they are variations of the same/naming error) has the smallest share of successful landings. Possibly due to difficulty landing in the area compared to others.

Highest Success Rate for a launch site.



SpaceX Launch Records Dashboard

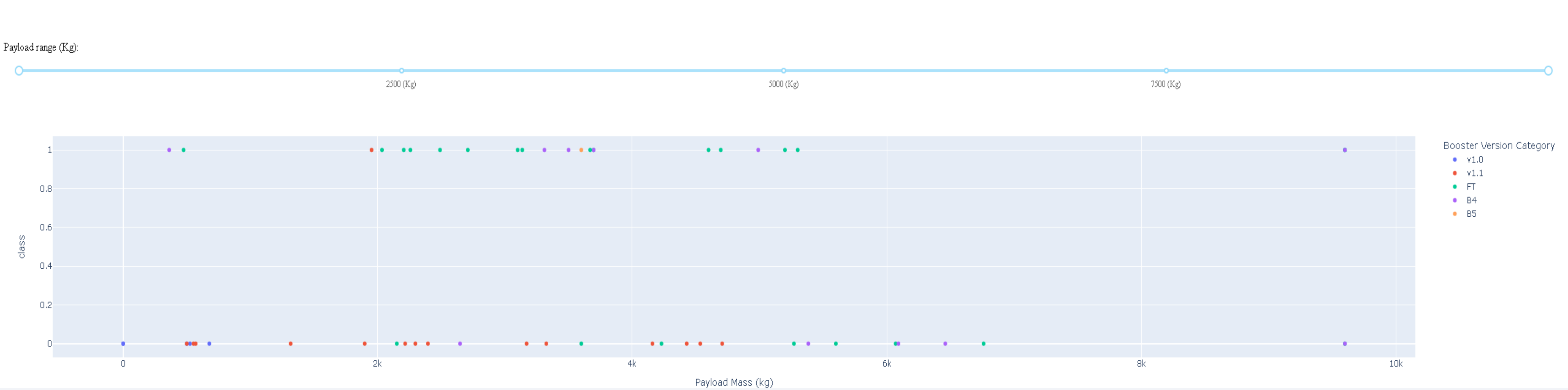
KSC LC-39A

Total Success Launches for KSC LC-39A

Payload range (Kg):

- KSC LC-39A has the highest success rate.

Payload Mass vs Success vs Booster Versions.



- Class 1 = Successful. Class 0 = Unsuccessful.
- Option to slide to filter against desired payload range.
- Payload range between 2-6k kg has the highest success rate.
- Payload range between 0-7 has the lowest launch success rate.
- F9 Booster is most common at Class 1.

Section 5

Predictive Analysis (Classification)

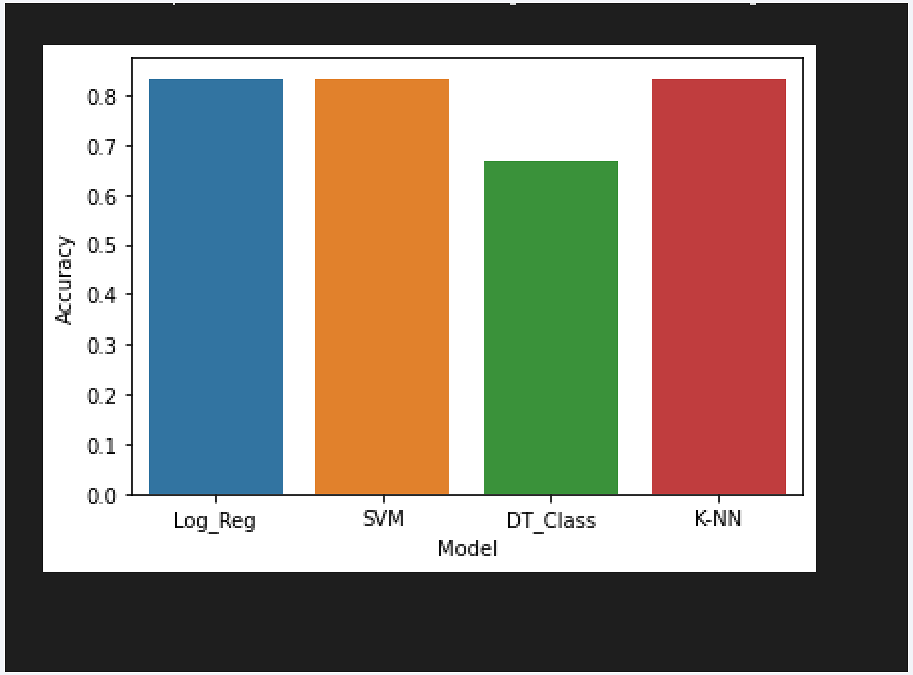
Classification Accuracy

- GridSearch (CV(10) Accuracy for each of the models:
  Logistic Regression, SVM, Decision Tree, and K Nearest Neighbor.

- All models except the Decision tree had roughly the same accuracy at 83.33%.

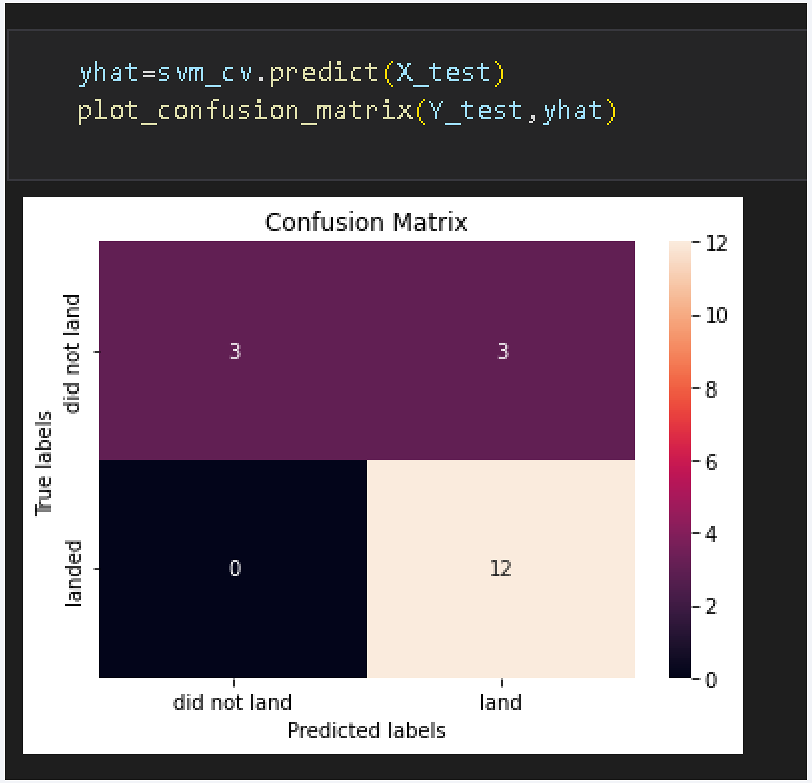
- The decision tree accuracy would vary between ~60% and 83.33% when run multiple occasions. This may be due to how the train data was split. The bar chart shown is when the DT test was at its lowest to be constructive.

- A larger sample size may remove that occurring.



Confusion Matrix

- Correct predictions, are diagonally from top left to bottom right. (true negative and true positive).
- The model did not have any false positives.
  The model had 3 false negatives.
- Models over predict successful landings. Collecting further data may minimize any discrepancies.



```
yhat=svm_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
```

Conclusions

- Objective was to develop machine learning models on behalf of Space Y. They wish to compete against Space X who are now established in the sector.
- Data has been collected from public sources, processed using various techniques.
- A user-friendly dashboard has been created which may be helpful to non-technical decision makers.
- The machine learning model created has an accuracy of 83.33%.
- Mr Mask of Space Y, can therefore, confidently use this model, with a relatively high degree of certainty, as to whether a stage 1 landing should be made or not.
- The model may also be used to plan for budgeting and cash flow forecasting to allow for crash landings, failures, and non recovered rockets.
- Further data should be collected to better improve the model accuracy and the level of confidence placed in it.

Appendix

- Thank you to all of the instructors, and to anyone who has contributed, or given advice to the online forums to other students.

Link to Instructors:

- Instructors for: IBM Data Science Professional Certificate | Coursera

SPACE Y

Allon Mask

Thank you!