IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Ethan Schroder
4th August 2024

Outline

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

Executive Summary

- Summary of methodologies
  - Data collection using APIs
  - Data collection with Web Scrapping
  - Data Wrangling
  - Exploratory Analysis with SQL
  - Exploratory Analysis with Data Visualisation
  - Interactive Visual Analytics with Folium
  - Interactive Dashboard with Ploty
  - Predictive Analysis using Machine Learning Classification
- Summary of all results
  - Exploratory Data Analysis result
  - Interactive Analytics result
  - Predictive Analysis result

Introduction

- Project background and context

In this project, we predict whether the Falcon 9 first stage will land successfully. For context, SpaceX advertises Falcon 9 rocket launches on their website, with a cost of 62 million dollars, with some products costing upwards of 165 million dollars, much of the savings is because SpaceX can reuse the first stage. As a result, we can determine if the first stage will land successfully. Hence, this information can be used if an alternate company wants to bid against SpaceX for a rocket launch.

- Problems you want to find answers
  - What factors determine if the rocket will land successfully.
  - How each relationship of the rocket's variables effect it's outcome.
  - Conditions which will aid SpaceX have to achieve the best outcome.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was collected by using SpaceX API and web scraping from Wikipedia
- Perform data wrangling
  - One-hot encoding data fields for machine learning and dropping irrelevant columns.
- 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

- We begin with the data collection using Web scraping by importing requests, BeautifulSoup and Pandas library, request data from Wikipedia URL using requests.get() method, using BeautifulSoup() constructor to create a beautiful soup object on the HTML response data, extracting all tables from the Soup object using soup.find_all(), retrieving the third table and Iterate through to extract header/column names and, creating a data frame by iterating through each of the HTML tables to extract 'Flight No’, 'Launch site’, 'Payload', 'Payload mass', 'Orbit', 'Customer', 'Launch outcome', 'Version Booster', 'Booster landing', 'Date', 'Time’.
- The URL below is the GIT repository containing the Jupyter Notebook: https://github.com/Ethan-Schroder/IBM-Data-Science-Capstone-Project-SpaceX/blob/0fc69dee46d94e583df81ce051daabcb26d3ee4f/Juypter%20Notebooks/jupyter-labs-spacex-data-collection-api.ipynb

Data Collection – SpaceX API

- We used the get request to the SpaceX API to collect data, clean the requested data and did some basic data wrangling and formatting.
- The URL below is the GIT repository containing the Jupyter Notebook: https://github.com/Ethan-Schroder/IBM-Data-Science-Capstone-Project-SpaceX/blob/add2592c42cce1683c70b5c9d8f528616f12a9c8/Juypter%20Notebooks/jupyter-labs-spacex-data-collection-api.ipynb

Data Collection - Scraping

- We applied web scrapping to webscrap Falcon 9 launch records with BeautifulSoup
- We parsed the table and converted it into a pandas dataframe.
- The URL below is the GIT repository containing the Jupyter Notebook: https://github.com/Ethan-Schroder/IBM-Data-Science-Capstone-Project-SpaceX/blob/0fc69dee46d94e583df81ce051daabcb26d3ee4f/Juypter%20Notebooks/jupyter-labs-webscraping.ipynb

Data Wrangling

- We begin by performing exploratory data analysis and determined the training labels.
- We calculated the number of launches at each site, and the number and occurrence of each orbits
- We created landing outcome label from outcome column and exported the results to csv.
- The URL below is the GIT repository containing the Jupyter Notebook: https://github.com/Ethan-Schroder/IBM-Data-Science-Capstone-Project-SpaceX/blob/0fc69dee46d94e583df81ce051daabcb26d3ee4f/Juypter%20Notebooks/labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

- In the exploratory data analysis, we use scatter plots to visualize how some variables would affect the launch outcome and also the relationships between these variables e.g. Payload mass vs flight number, flight number vs launch site, and payload vs launch site.
- We also use a bar chart to visualize the success rate of each orbit type and scatter plots to visualize the relationship between flight number and orbit type, payload, and orbit type
- Lastly we use a line plot to visualize the yearly trend of the launch successes.
- The URL below is the GIT repository containing the Jupyter Notebook: https://github.com/Ethan-Schroder/IBM-Data-Science-Capstone-Project-SpaceX/blob/0fc69dee46d94e583df81ce051daabcb26d3ee4f/Juypter%20Notebooks/edadataviz.ipynb

EDA with SQL

- We begin with loading the SpaceX dataset into a PostgreSQL database without leaving the Jupyter notebook, we then applied EDA with SQL to get insight from the data.
- Display names of the booster versions that have carried the maximum payload mass, the total payload mass carried by booster launched by NASDA (CRS), average payload mass carried by booster version F9 v1.1.
- Display month names, failure landing outcomes in drone ship, booster versions, and launch sites for the months in the year 2015.
- Rank the count of successful landing outcomes between the dates 04-06-2010 and 20-03-2017 in descending order.
- The URL below is the GIT repository containing the Jupyter notebook for EDA with SQL: https://github.com/Ethan-Schroder/IBM-Data-Science-Capstone-Project-SpaceX/blob/0fc69dee46d94e583df81ce051daabcb26d3ee4f/Juypter%20Notebooks/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- We used functions from the Folium API, these includes, folium.Circle to add a highlighted circle area of NASA JSC as an initial centre location, Folium.map.Marker to create a marker at a specific launch location on the map, MarkerCluster() to create cluster markers of successful and failed launches for a particular site, MousePosition() to provide a way to display the latitude and longitude coordinates of the mouse cursor's position on a map and Folium.PolyLine() to create a series of connected line segments on the map to mark the distance of the launch sites to the coast, railways, highways, and major cities
- We marked all launch sites, and added map objects such as markers, circles, lines to mark the success or failure of launches for each site on the folium map.
- We assigned the feature launch outcomes (failure or success) to class 0 and 1, where 0 was for failure, and 1 was for success.
- The URL below is the GIT repository containing the Jupyter Notebook: https://github.com/Ethan-Schroder/IBM-Data-Science-Capstone-Project-SpaceX/blob/0fc69dee46d94e583df81ce051daabcb26d3ee4f/Juypter%20Noteboks/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

- We built an interactive dashboard with Plotly dash, where we plot pie charts showing the total launches by a certain sites and plots of scatter graphs showing the relationship with Outcome and Payload Mass (Kg) for the different booster version.
- The plots and graphs added to the dashboard include a drop-down input containing all the launch sites, a range slider to select the payload mass and a scatter plot showing the correlations between the payload mass and the success for the launch sites selected.
- The URL below is the GIT repository containing the Python file: https://github.com/Ethan-Schroder/IBM-Data-Science-Capstone-Project-SpaceX/blob/9998220faffe1ee9617e9291bd314c0abad9b7db/spacex_dash_app.py

Predictive Analysis (Classification)

- We begin with loaded the data set using NumPy and Pandas, transformed the data, split our data into training and testing.
- Then we built different machine learning models those are logistic regression, support vector , tree classifier, and k nearest neighbours. Afterwards we then tune different hyperparameters using GridSearchCV.
- We used accuracy as the metric for our model, improved the model using feature engineering and algorithm tuning, where we were able to find the best performing classification model.
- The URL below is the GIT repository containing the Jupyter Notebook: https://github.com/Ethan-Schroder/IBM-Data-Science-Capstone-Project-SpaceX/blob/add2592c42cce1683c70b5c9d8f528616f12a9c8/Juypter%20Notebooks/SpaceX_Machine%20Learning%20Prediction.ipynb

Results

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

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

The figure below shows that the greater the flight number, the more successful launches are at each launch site. Note as well CCAFS SLC-40 has launched the highest number of flights.

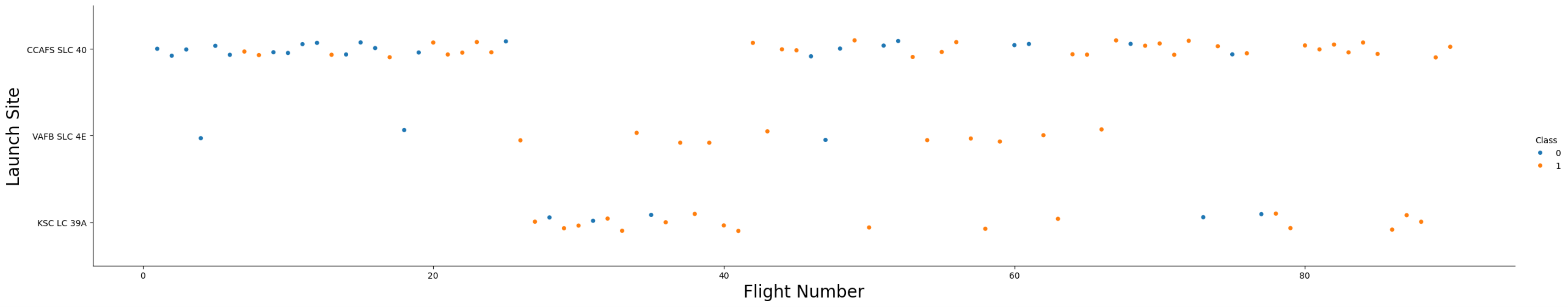


Figure 1. Scatter plot of flight number vs Launch Site.

Payload vs. Launch Site

The figure below, it is observed that VAFB-SLC 4E launch site has no rockets launched for heavy payload mass greater than 10000kg, most of the rockets launched in all launch sites have a payload mass of less than 9000kg and compared to VAFB-SLC 4E and KSC LC 39A, CCAFS SLC 40 has a higher success rate for rockets launched with a heavy payload mass of 14000kg and 16000kg.

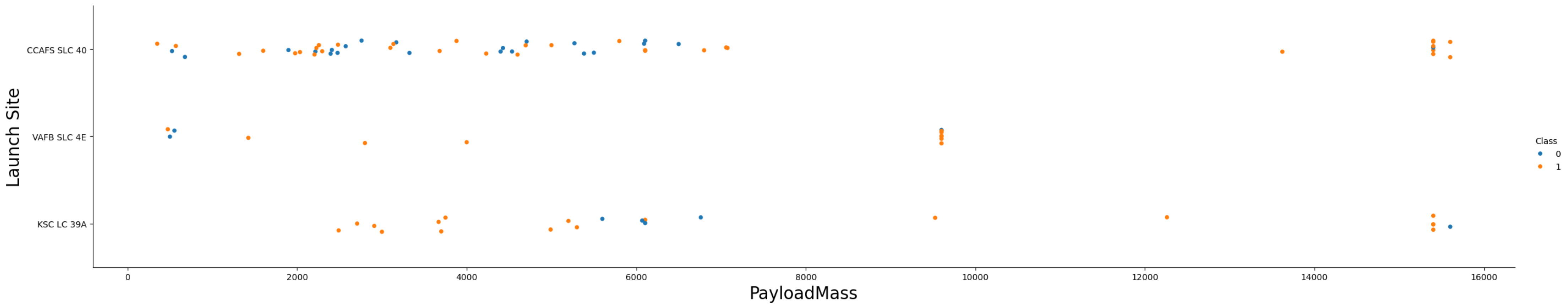


Figure 2. Scatter plot of Payload vs Launch Site.

Success Rate vs. Orbit Type

From the figure, it is observed that orbits ES-L1, GEO and HEO, have the highest success rates compared to the other orbit types, note as well it is observed that orbit SO has the least success rate.

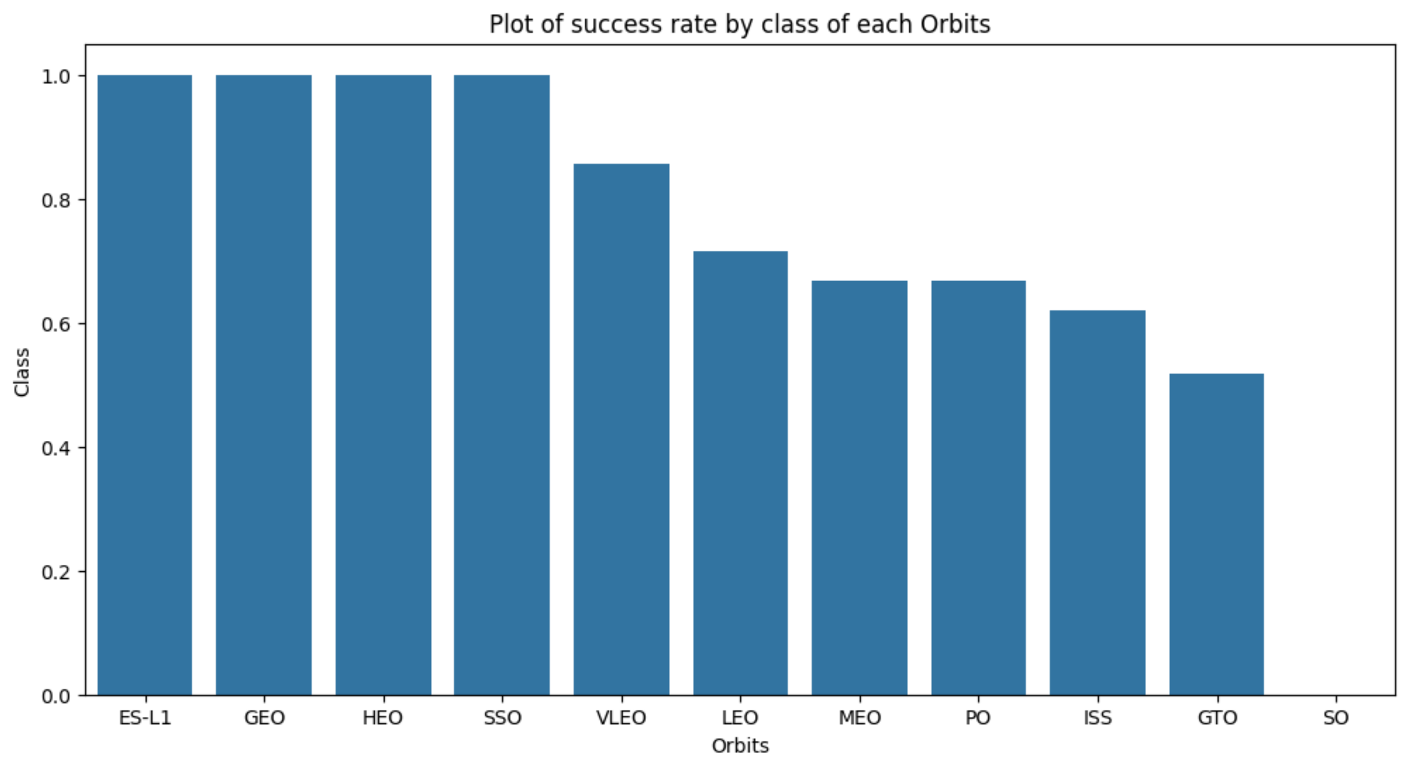


Figure 3. Bar Chart of the success rate by each type of orbit.

Flight Number vs. Orbit Type

The figure below shows that there are more rockets were launched in LES ISS PO GTO and VLEO. It is also observed that in the LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number and other orbit.

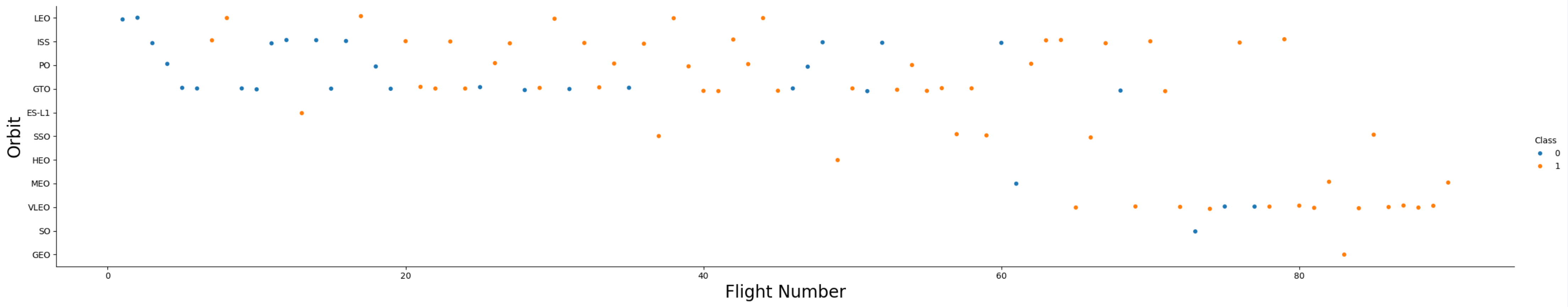


Figure 4. Scatter plot of flight number vs orbit type.

Payload vs. Orbit Type

There is a higher success for rockets with heavy payloads launched in PO, LEO, and ISS, where rockets launched in SSO and MEO orbits on the other hand have a high success rate with lighter payloads. Rockets launched in GTO have both positive landing rates and negative landing rates regardless of the size of the payload.

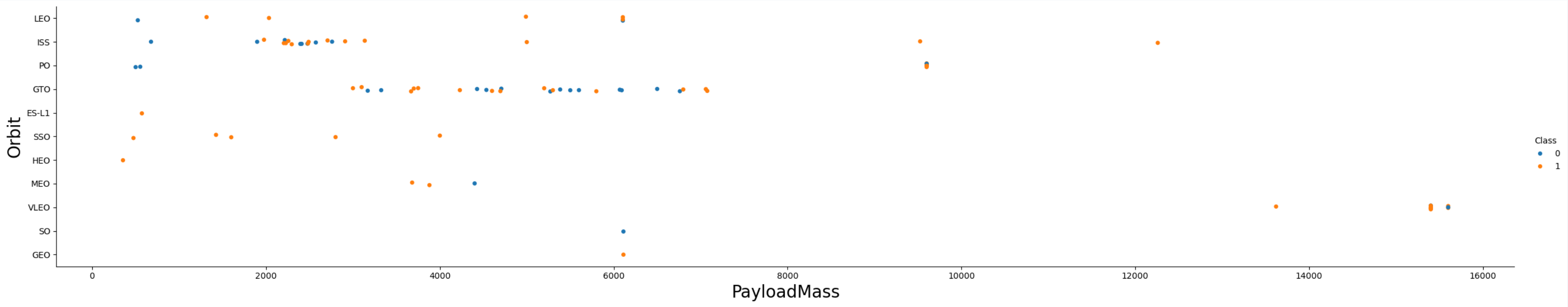


Figure 5. Scatter plot of payload vs orbit type.

Launch Success Yearly Trend

From the figure we can observe that the success rate since 2013 kept increasing till 2020

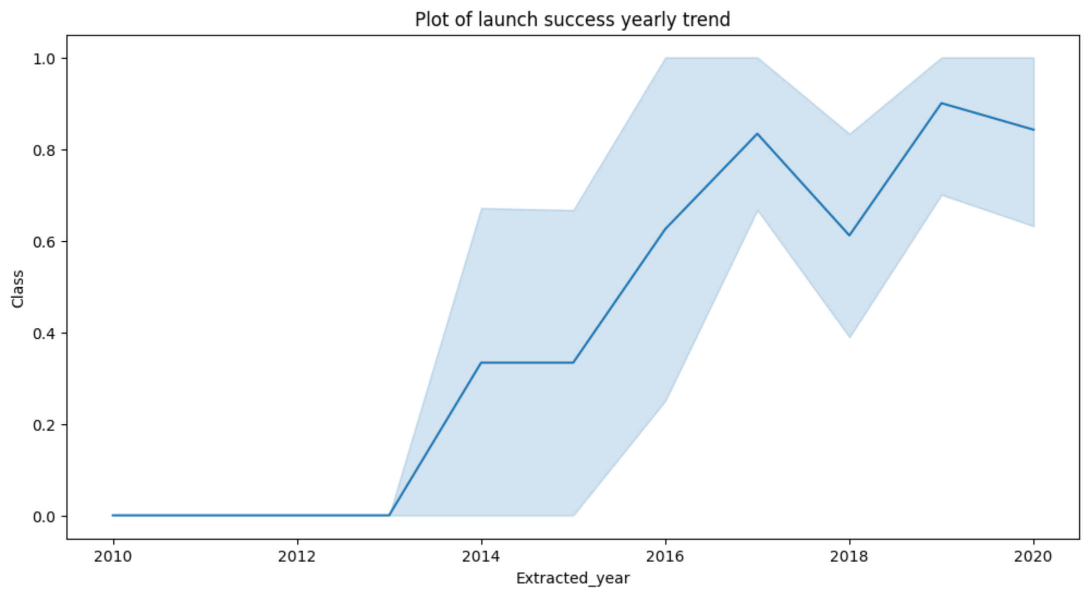
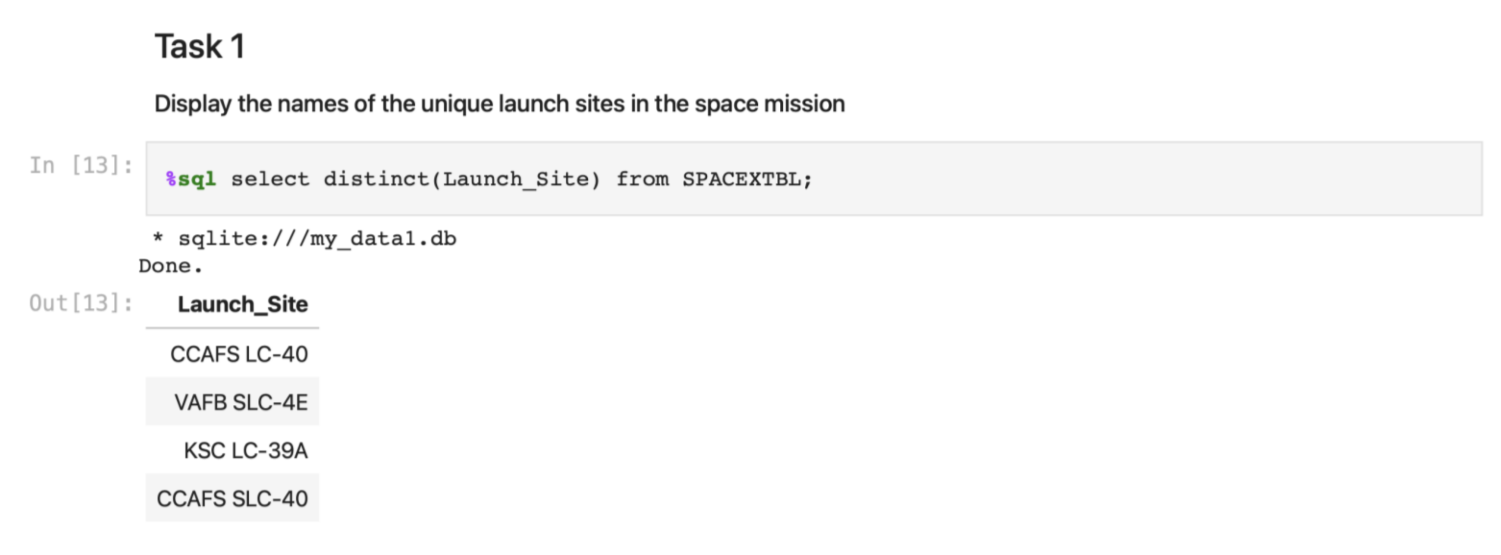


Figure 6. Regression plot of launch success yearly trend

All Launch Site Names

For this data we used the SPACEXTBL data frame. Then we used the DISTINCT key word to show the names of each unique launch site from the SpaceX



Task 1

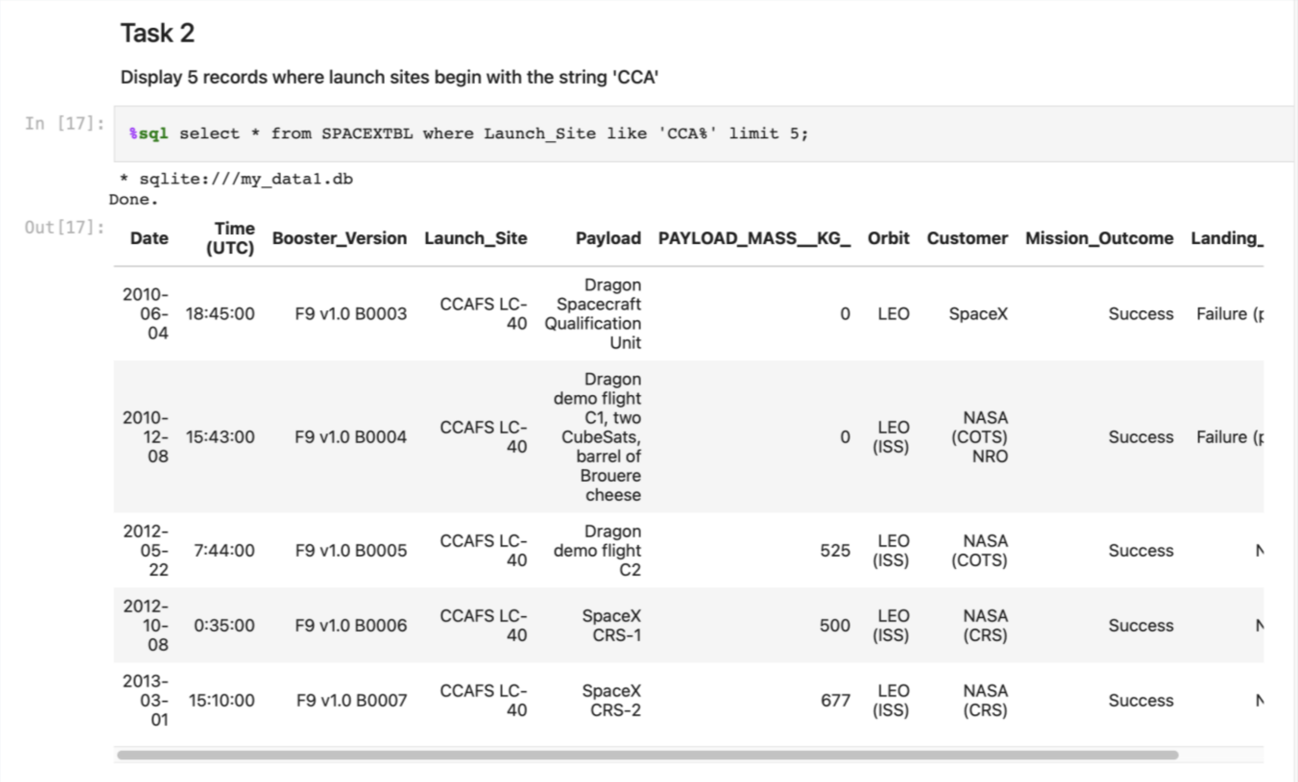
Display the names of the unique launch sites in the space mission

```
In [13]: %sql select distinct(Launch_Site) from SPACEXTBL;

 * sqlite:///my_data1.db
Done.

Out[13]:
 Launch_Site
CCAFS LC-40
VAFB SLC-4E
 KSC LC-39A
CCAFS SLC-40
```

Figure 7. SQL query for unique launch site names.

Launch Site Names Begin with 'CCA'

Here we use the keyword LIKE ‘CCA%’ to get the launch site names that begin with CCA. We also use the LIMIT 5 keyword to get only 5 records.



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;

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

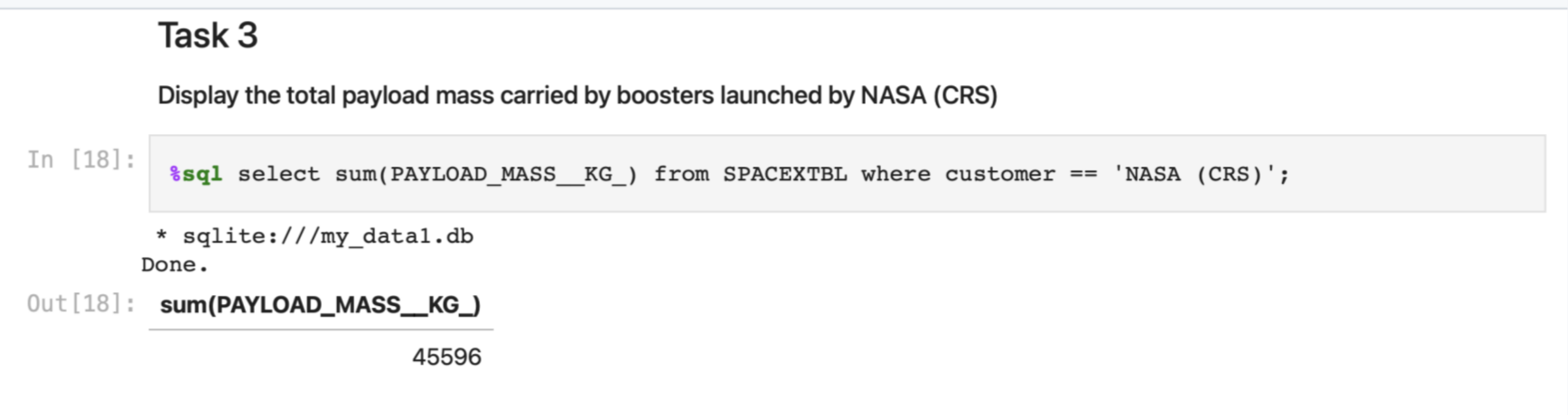
Out[17]:

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

Figure 8. SQL query for launch site names that begin with CCA.

Total Payload Mass

Here we use SUM keyword to obtain the total payload mass alongside using the WHERE keyword to ensure the total payload mass is from the customer NASA CRS.



Task 3

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

```
In [18]: %sql select sum(PAYLOAD_MASS__KG_) from SPACEXTBL where customer == 'NASA (CRS)';

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

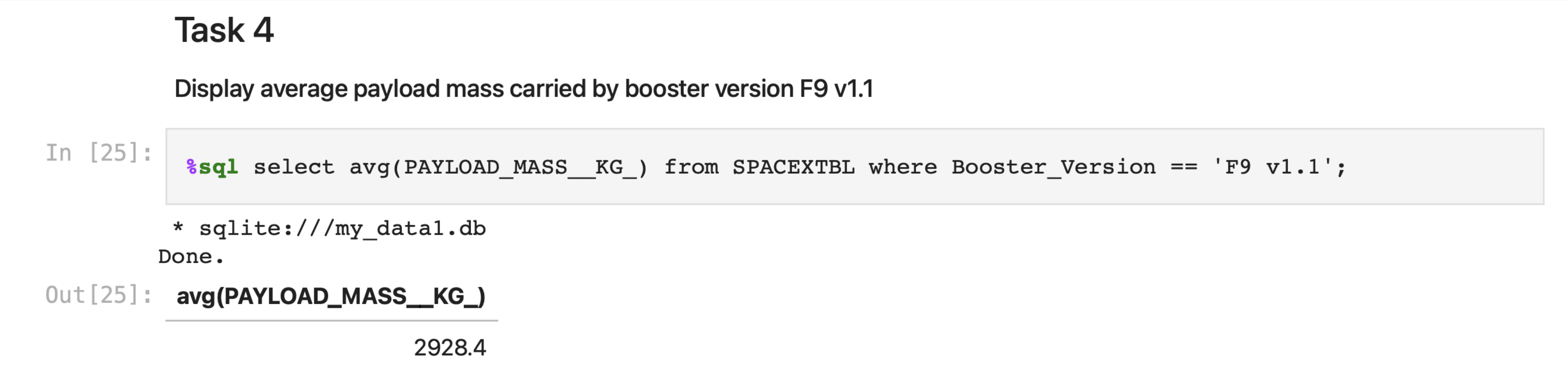
Out[18]:

sum(PAYLOAD_MASS__KG_)
45596

Figure x. SQL query for Total Payload Mass from NASA CRS.

Average Payload Mass by F9 v1.1

Here we use the AVG keyword to get the average payload mass, not we use the WHERE keyword to select the Booster Version F9 v1.1.



Task 4

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

In [25]:
```
%sql select avg(PAYLOAD_MASS__KG_) from SPACEXTBL where Booster_Version == 'F9 v1.1';
```

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

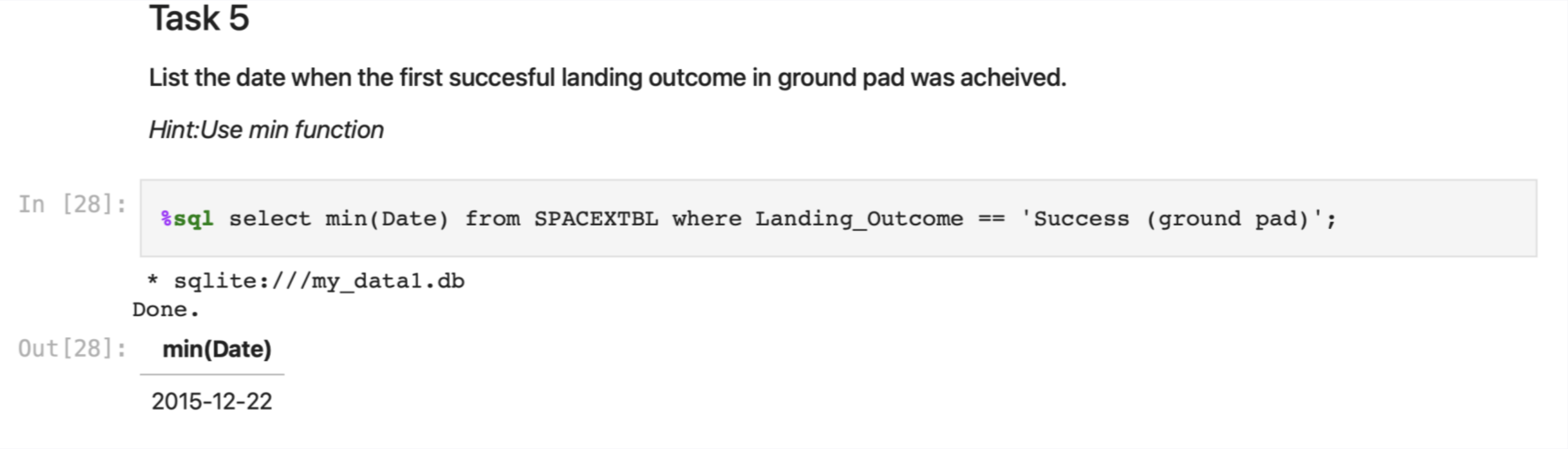
Out[25]:

avg(PAYLOAD_MASS__KG_)
2928.4

Figure 9. SQL query for average payload mass by F9 v1.1

First Successful Ground Landing Date

Here we use the MIN keyword to select the earliest date of any launch, then we use the WHERE keyword to select a ground landing outcome that was successful, so we obtain the earliest date of a successful ground landing.



Task 5

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

Hint:Use min function

```
In [28]: %sql select min(Date) from SPACEXTBL where Landing_Outcome == 'Success (ground pad)';
```

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

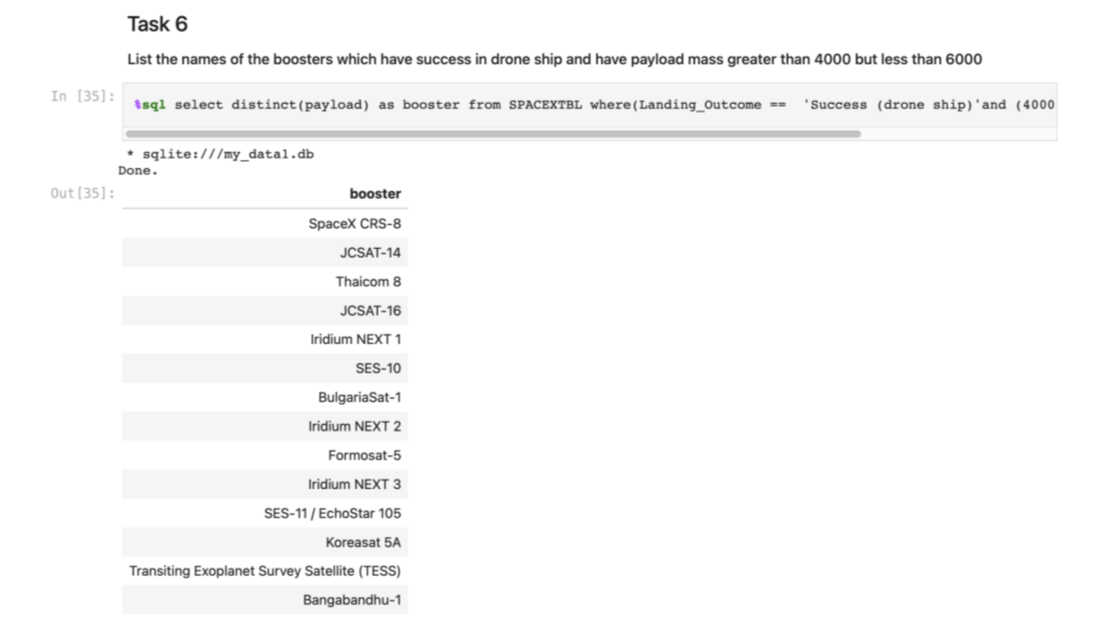
Out[28]:

min(Date)
2015-12-22

Figure 10. SQL query for the date of the first successful ground launch.

Successful Drone Ship Landing with Payload between 4000 and 6000

We use the DISTINCT keyword to select each unique payload that we relabel as the booster column, then we use a WHERE keyword and a sub query to add a condition where the drone ship was successful and had a payload mass between 4000kg and 6000kg.



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 [35]: %sql select distinct(payload) as booster from SPACEXTBL where(Landing_Outcome ==  'Success (drone ship)'and (4000
```

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

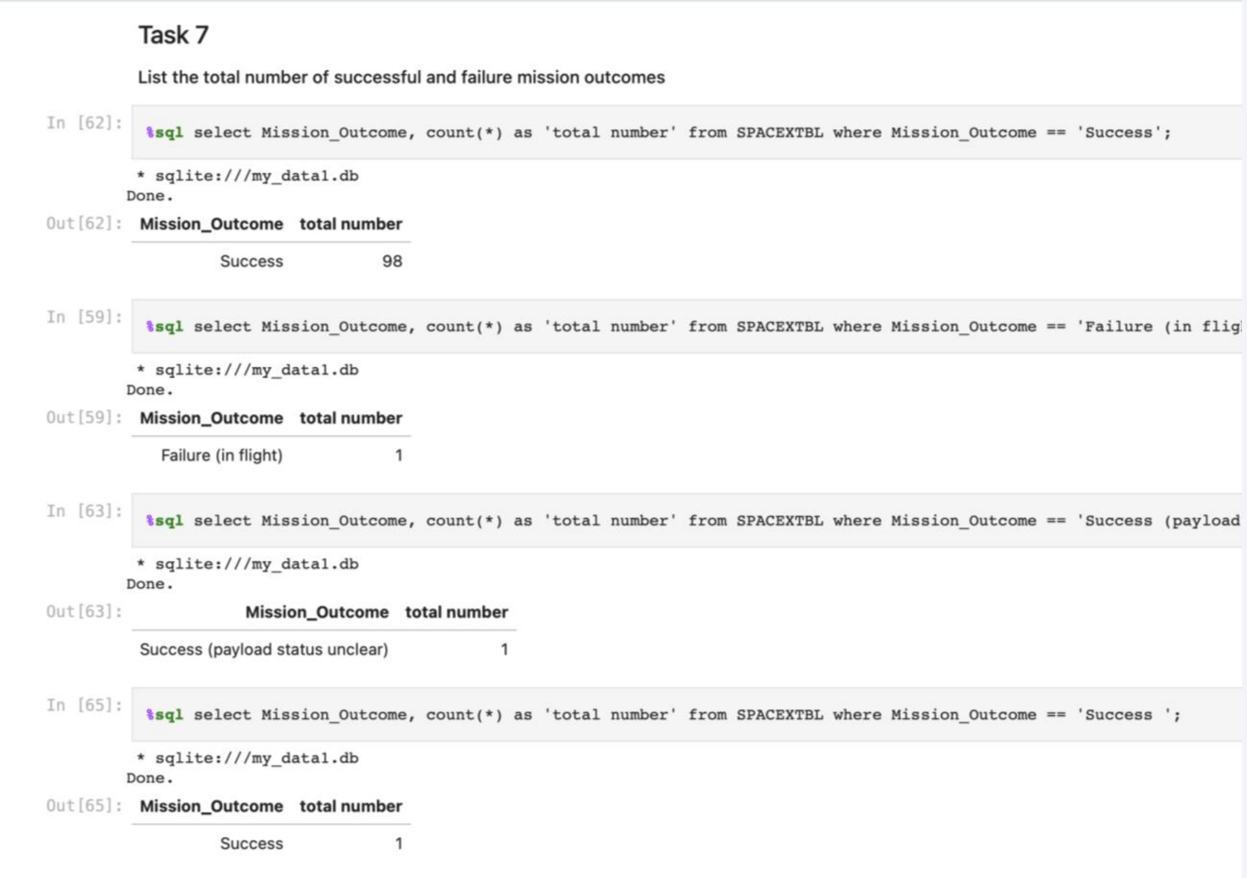
Out[35]:

booster
SpaceX CRS-8
JCSAT-14
Thaicom 8
JCSAT-16
Iridium NEXT 1
SES-10
BulgariaSat-1
Iridium NEXT 2
Formosat-5
Iridium NEXT 3
SES-11 / EchoStar 105
Koreasat 5A
Transiting Exoplanet Survey Satellite (TESS)
Bangabandhu-1

Figure 11. SQL query for successful drone ship landing with payload between 4000 and 6000.

Total Number of Successful and Failure Mission Outcomes

Note as before we use the WHERE keyword to select rows where the mission was successful or not, we use the COUNT keyword to count the number of rows that fall under this condition, note there are two rows that are successful but named differently as shown in the figure.



Task 7

List the total number of successful and failure mission outcomes

```
In [62]:  %sql select Mission_Outcome, count(*) as 'total number' from SPACEXTBL where Mission_Outcome == 'Success';

           * sqlite:///my_data1.db
          Done.
Out[62]:  Mission_Outcome  total number
                 Success            98
```

```
In [59]:  %sql select Mission_Outcome, count(*) as 'total number' from SPACEXTBL where Mission_Outcome == 'Failure (in flig

           * sqlite:///my_data1.db
          Done.
Out[59]:  Mission_Outcome  total number
          Failure (in flight)        1
```

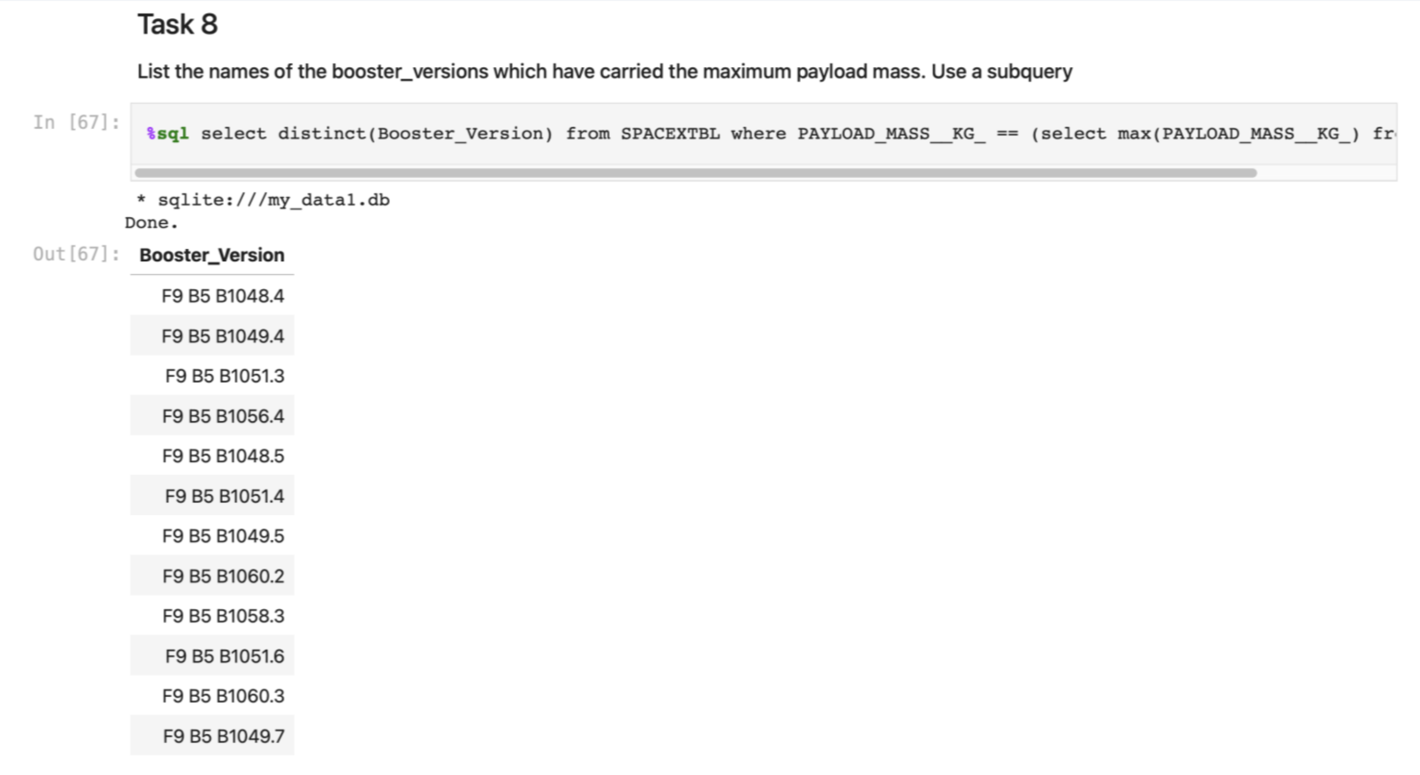
```
In [63]:  %sql select Mission_Outcome, count(*) as 'total number' from SPACEXTBL where Mission_Outcome == 'Success (payload

           * sqlite:///my_data1.db
          Done.
Out[63]:                Mission_Outcome  total number
          Success (payload status unclear)          1
```

```
In [65]:  %sql select Mission_Outcome, count(*) as 'total number' from SPACEXTBL where Mission_Outcome == 'Success ';

           * sqlite:///my_data1.db
          Done.
Out[65]:  Mission_Outcome  total number
                 Success             1
```

Figure 12. SQL query for total number of successful and failure mission outcomes.

Boosters Carried Maximum Payload

Here we use the WHERE keyword to use a sub query to select the rows that contain the maximum payload mass, where the sub query uses the MAX keyword to do this, we then use the DISTINCT keyword to select each unique Booster Version.



Task 8

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

```
In [67]: %sql select distinct(Booster_Version) from SPACEXTBL where PAYLOAD_MASS__KG_ == (select max(PAYLOAD_MASS__KG_) fr
```

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

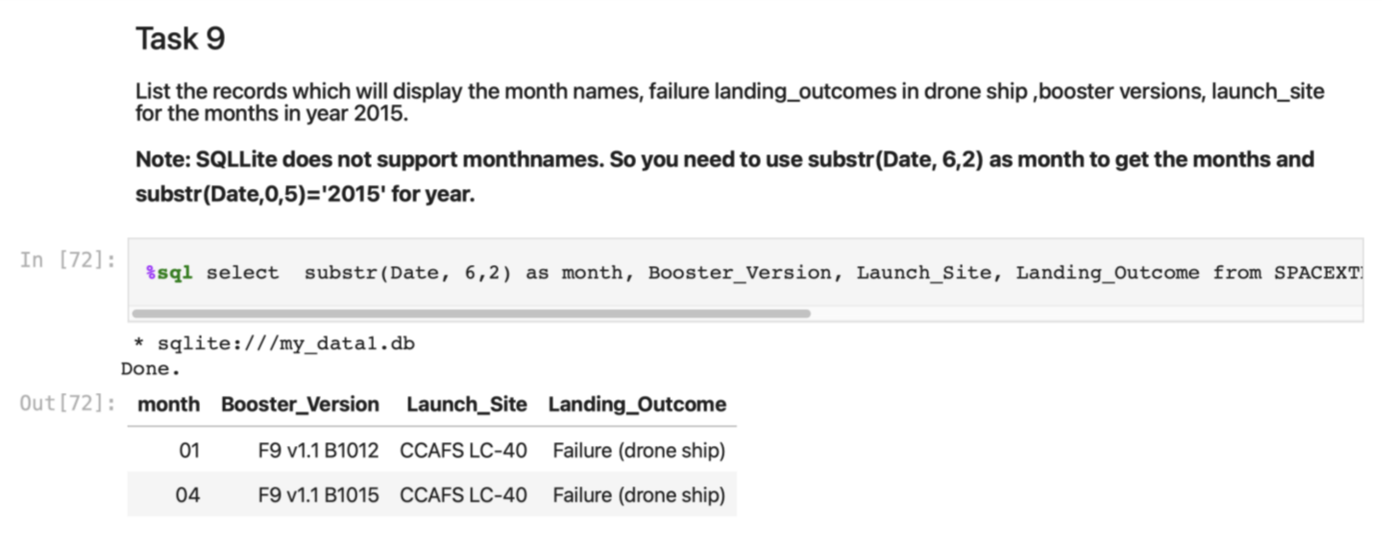
Out[67]:

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

Figure 13. SQL query for which booster carried the maximum payload.

2015 Launch Records

We begin with using the function SUBSTR to obtain the month number from our rows and name this column as month, then we use the WHERE keyword to select the failure landing outcomes in drone ships.



Task 9

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, 6,2) as month to get the months and substr(Date,0,5)='2015' for year.

```
In [72]: %sql select  substr(Date, 6,2) as month, Booster_Version, Launch_Site, Landing_Outcome from SPACEXT
```

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

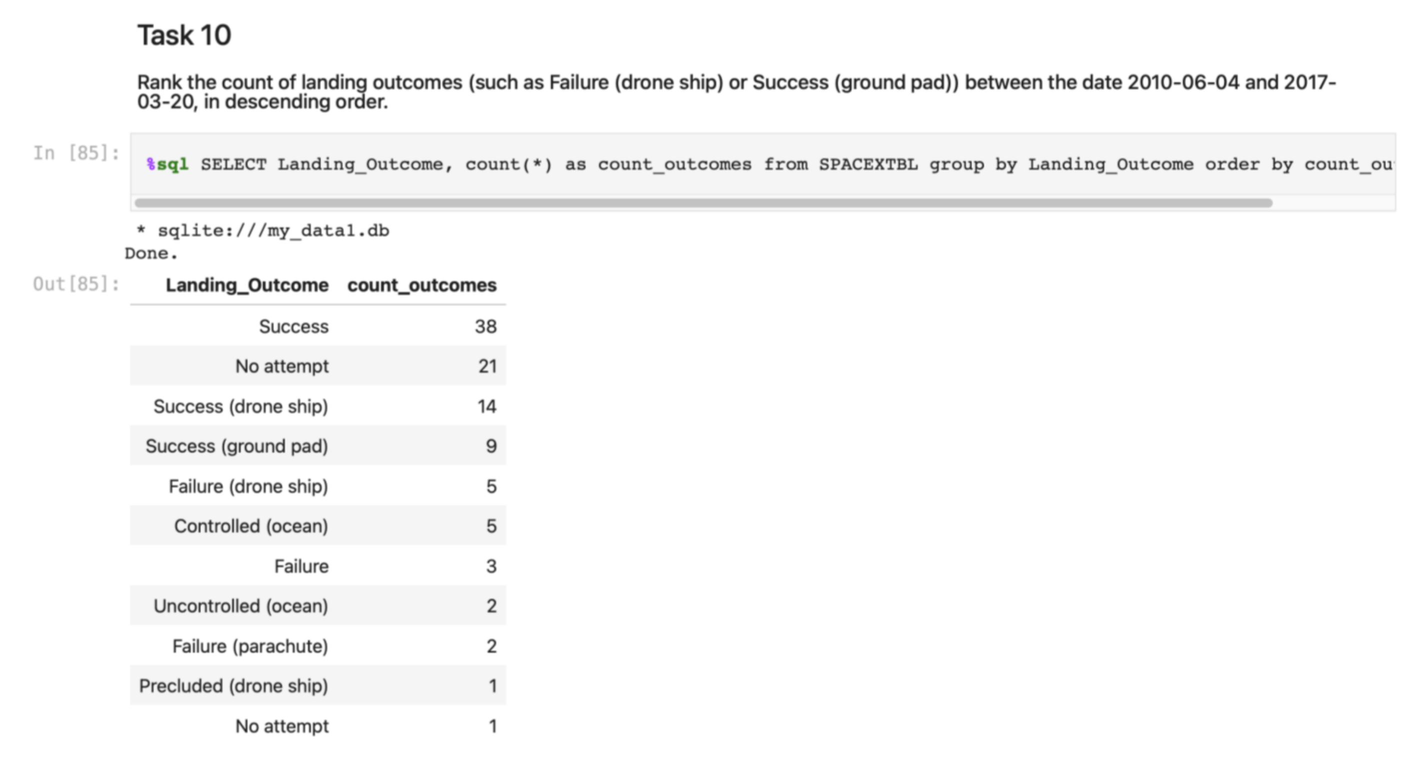
Out[72]:

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

Figure 14. SQL query for 2015 launch records.

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

Here we use keywords such as GROUP BY, ORDER BY, and DESC as well as functions like SUBSTR() and COUNT() were used to rank the count landing outcomes between 2010-06-04 and 2017-03-20, in descending order.



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 [85]: %sql SELECT Landing_Outcome, count(*) as count_outcomes from SPACEXTBL group by Landing_Outcome order by count_ou
```

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

Out[85]:

Landing_Outcome	count_outcomes
Success	38
No attempt	21
Success (drone ship)	14
Success (ground pad)	9
Failure (drone ship)	5
Controlled (ocean)	5
Failure	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1
No attempt	1

Figure 15. SQL query for rank landing outcomes between 2010-06-04 and 2017-03-20.

Section 3

Launch Sites Proximities Analysis

Launch Site Locations for Space X Falcon 9

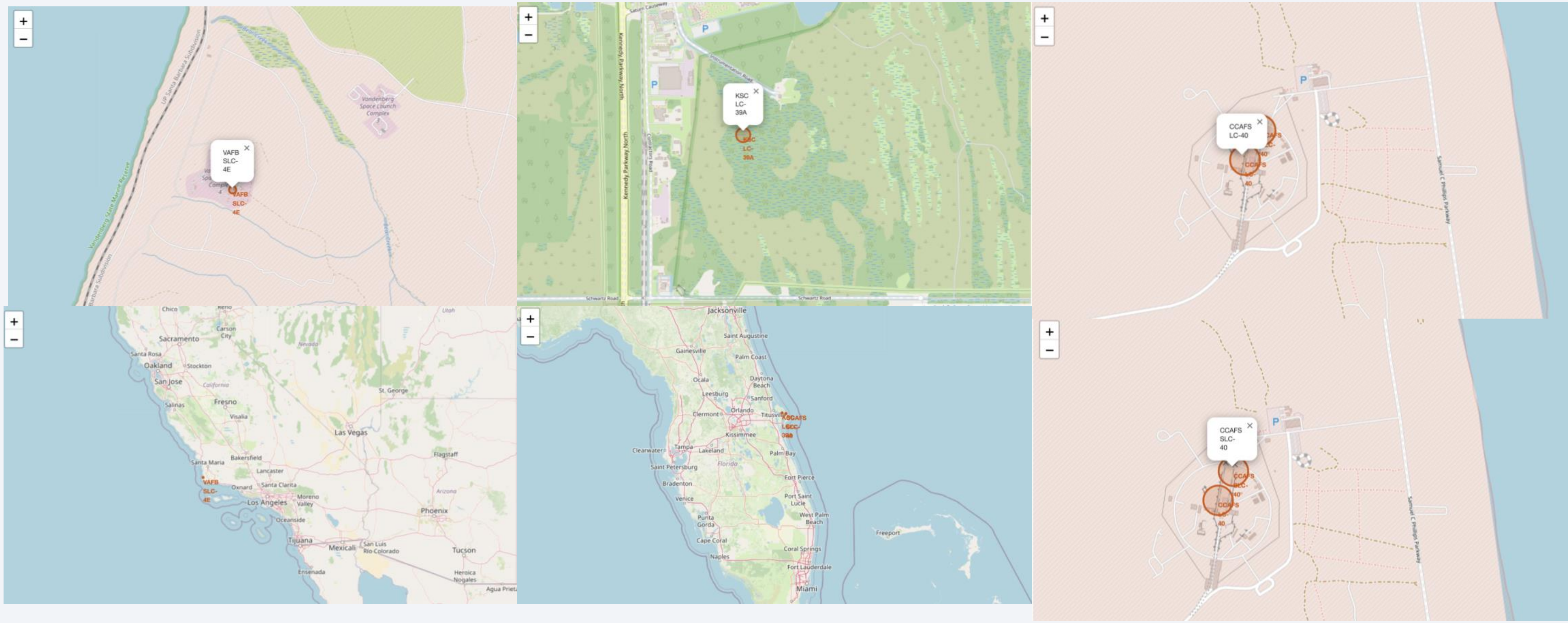


Figure 16. Note CCAFS LC-40, CCAFS SLC-40 and KSC LC-39A are in Florida USA, and VAFB SLC-4E is in California USA.

Launch Outcomes for Space X Falcon 9

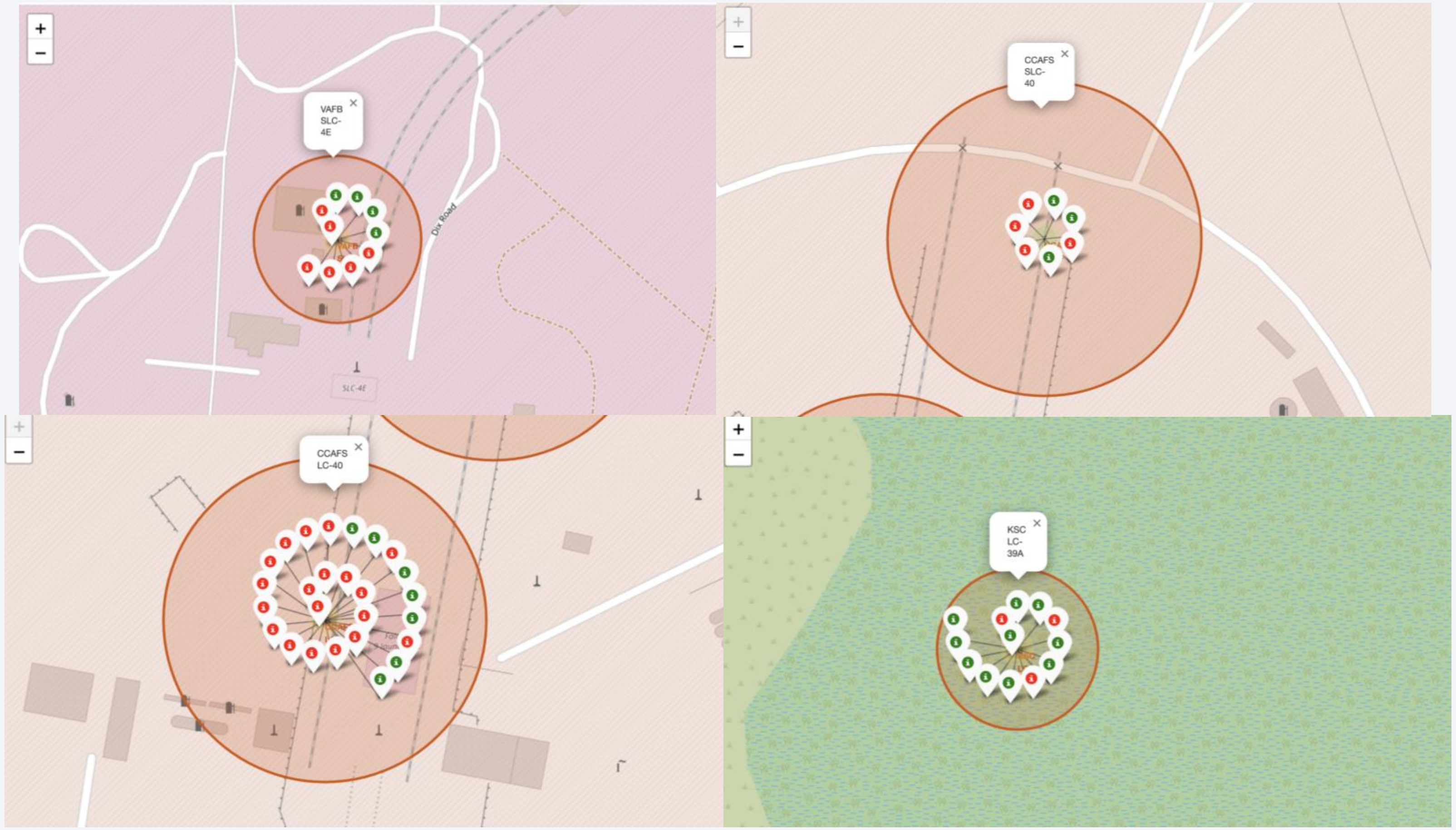


Figure 17. Launch outcomes for each launch site, green successful and red unsuccessful.

Launch Site CCAFS SLC-40 distance to Landmarks

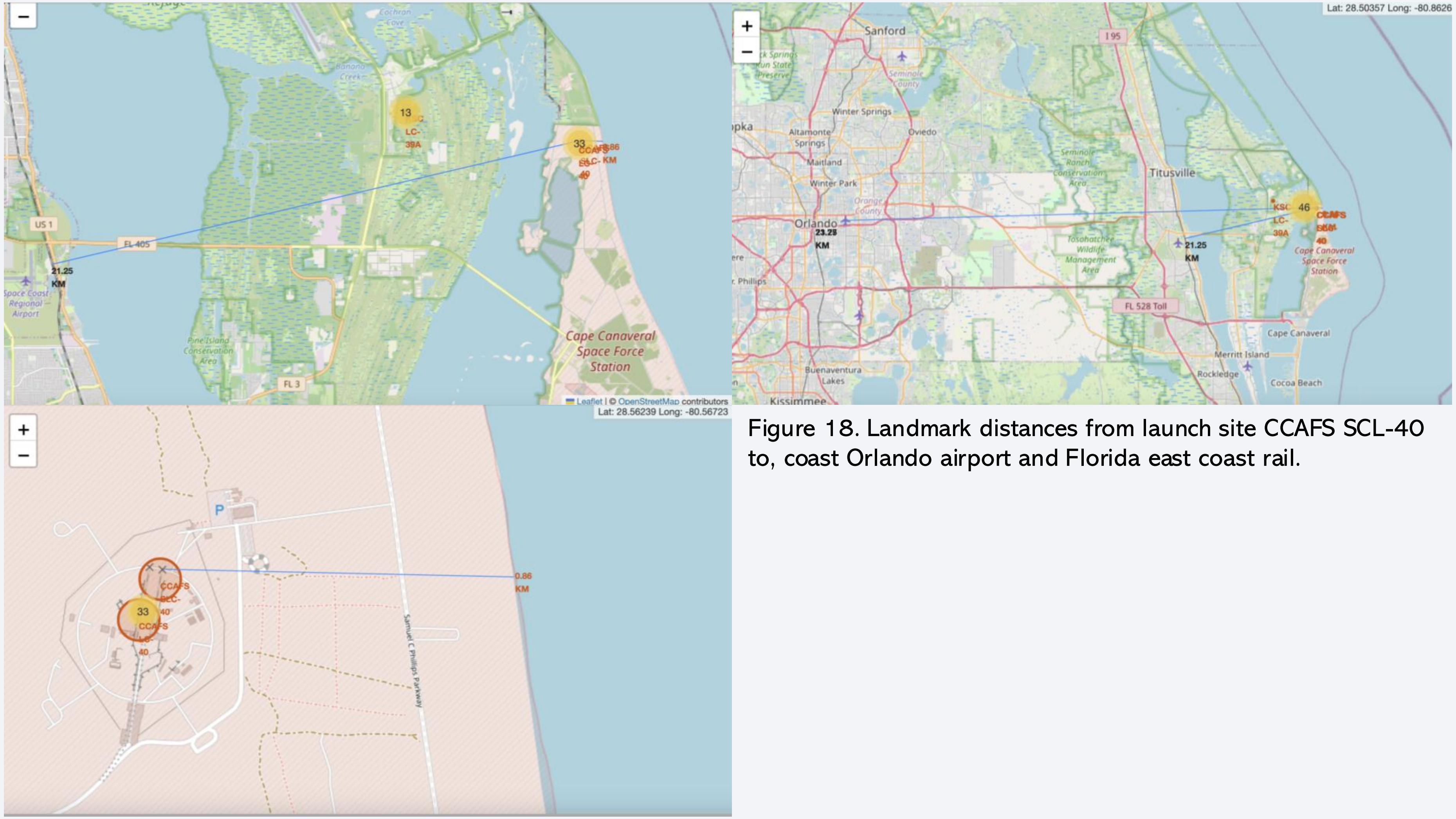
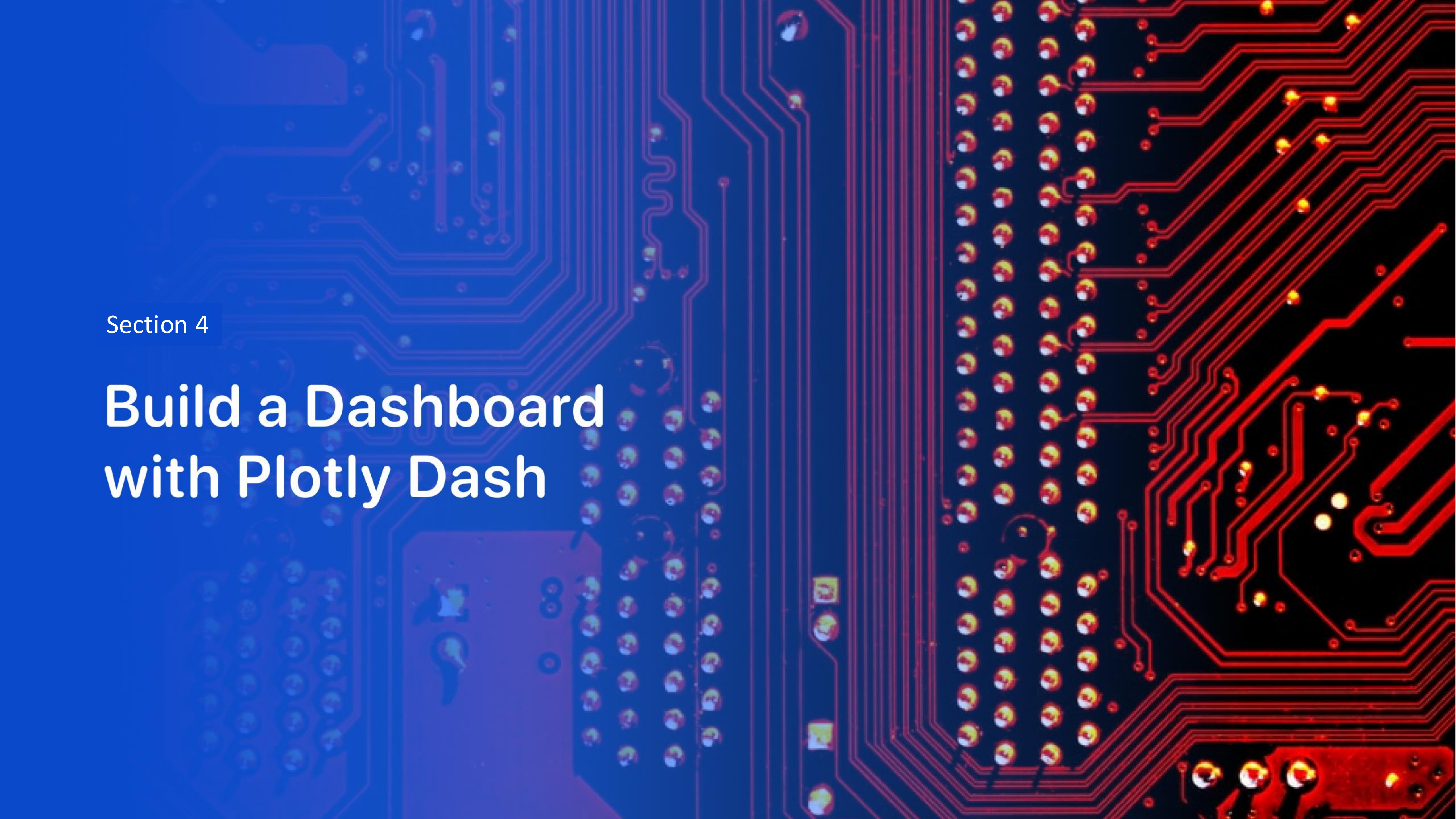


Figure 18. Landmark distances from launch site CCAFS SCL-40 to, coast Orlando airport and Florida east coast rail.

Section 4

Build a Dashboard with Plotly Dash



Pie Chart showing the success percentage achieved by each launch site

Figure x illustrates the KSC LC-39A had the largest success rate ratio at 41.7%, while the CCAFS SLC-40 had the lowest success rate ratio at 12.5%.

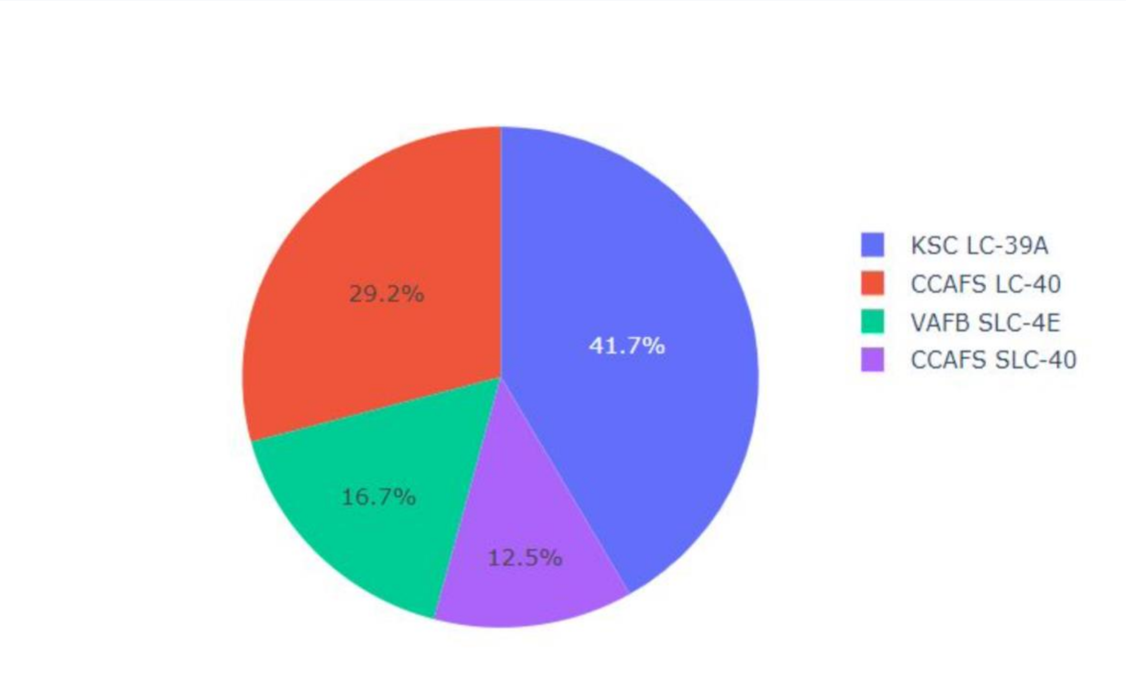


Figure 19. Pie Chart showing the success rate of all Launch sites.

Pie Chart showing the launch site with the highest launch success

Figure x. illustrates that the KSC LC-39A has the highest success ratio with about 76.9%, compared to the other sites. 73.1% for CCAFS LC-40, 60% for VAFB SLC-4E and 57.1% for CCAFS SLC-40

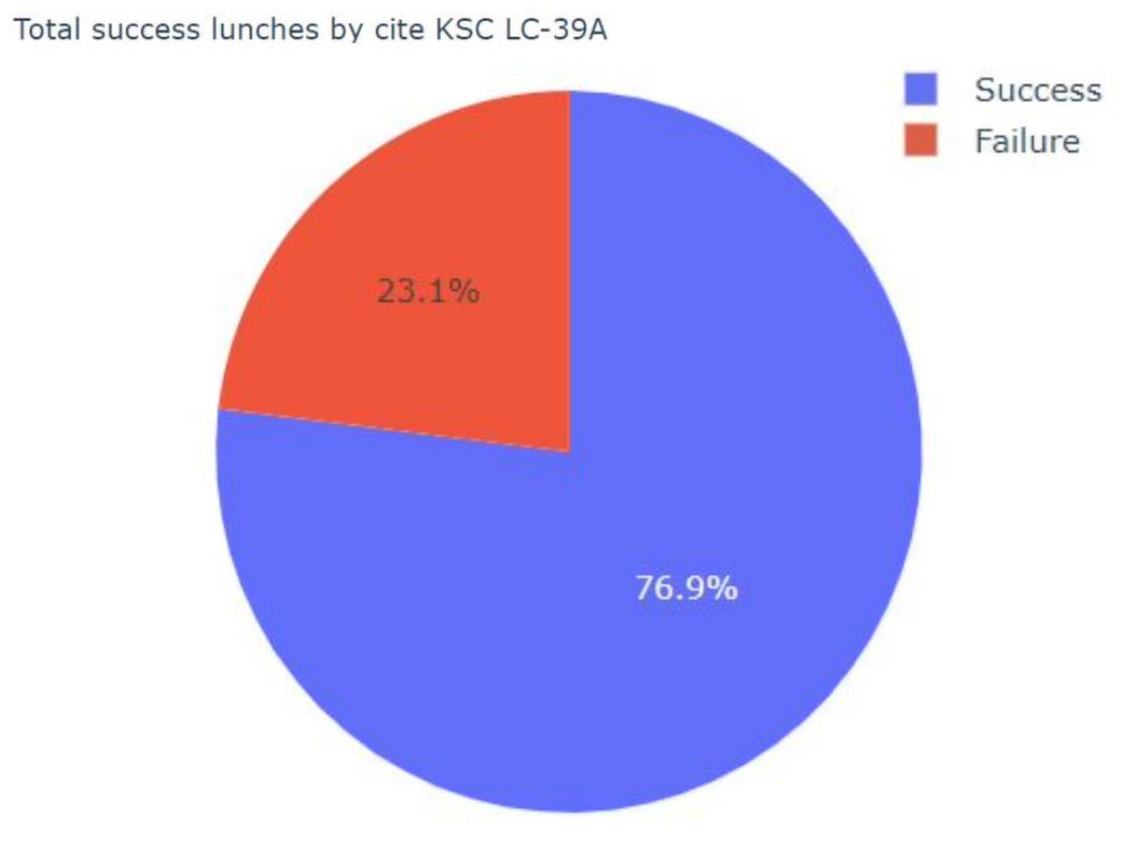


Figure 20. Pie Chart showing launch site with highest success ratio.

Scatter plot of Payload vs Launch Outcome for all sites, with different payload selected in the range slider

From the figures below, Booster version FT has the highest success rate with its payload mass of about between 700kg to 5,500kg.

It is also shown that rockets with payload mass above 5,500kg have a lower success rate, which means the heavier the payload, the slimmer the chance of a successful outcome.

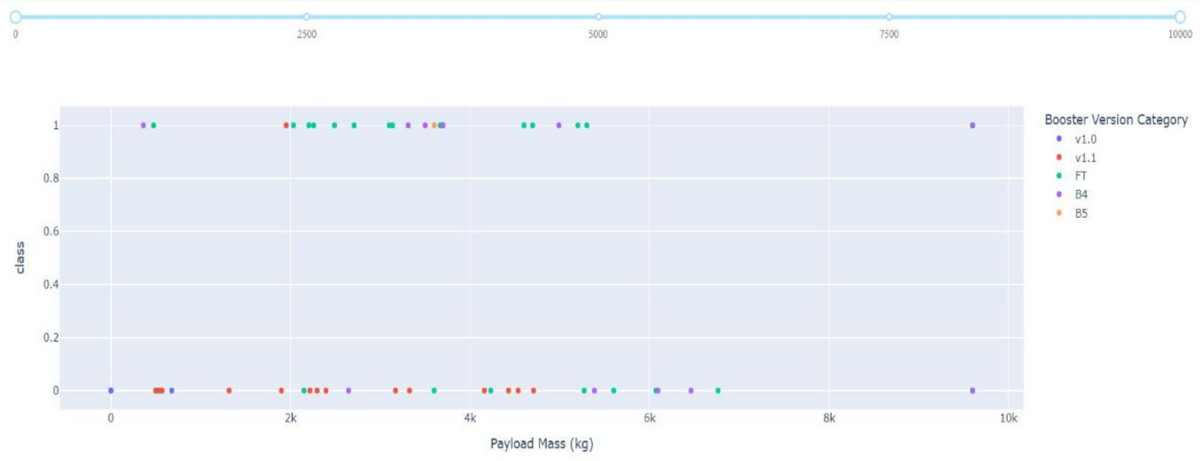


Figure 21. Scatter plot showing the booster versions with different payload mass for all launch sites.

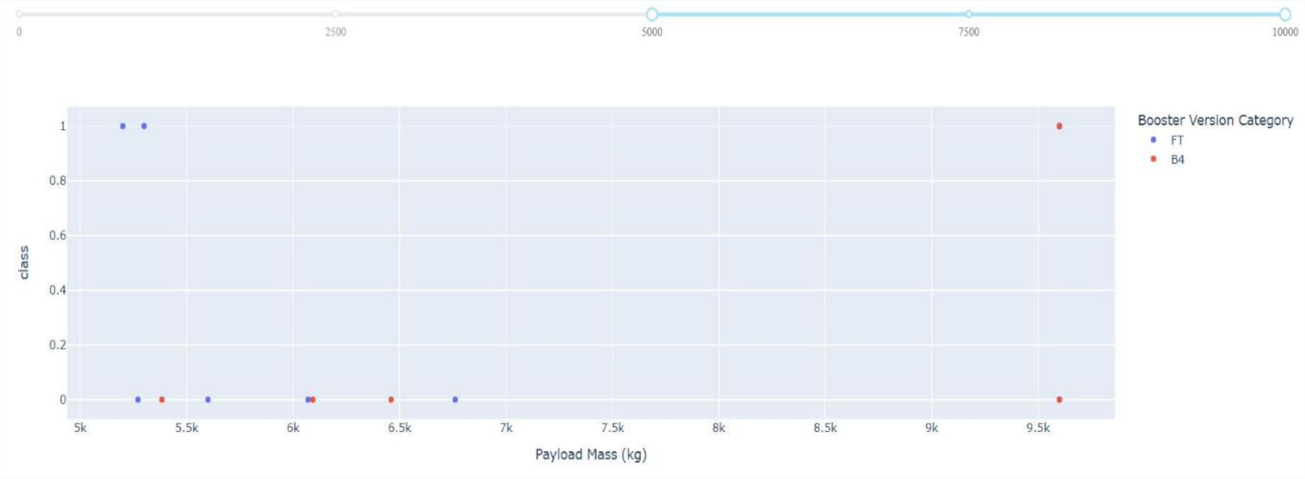


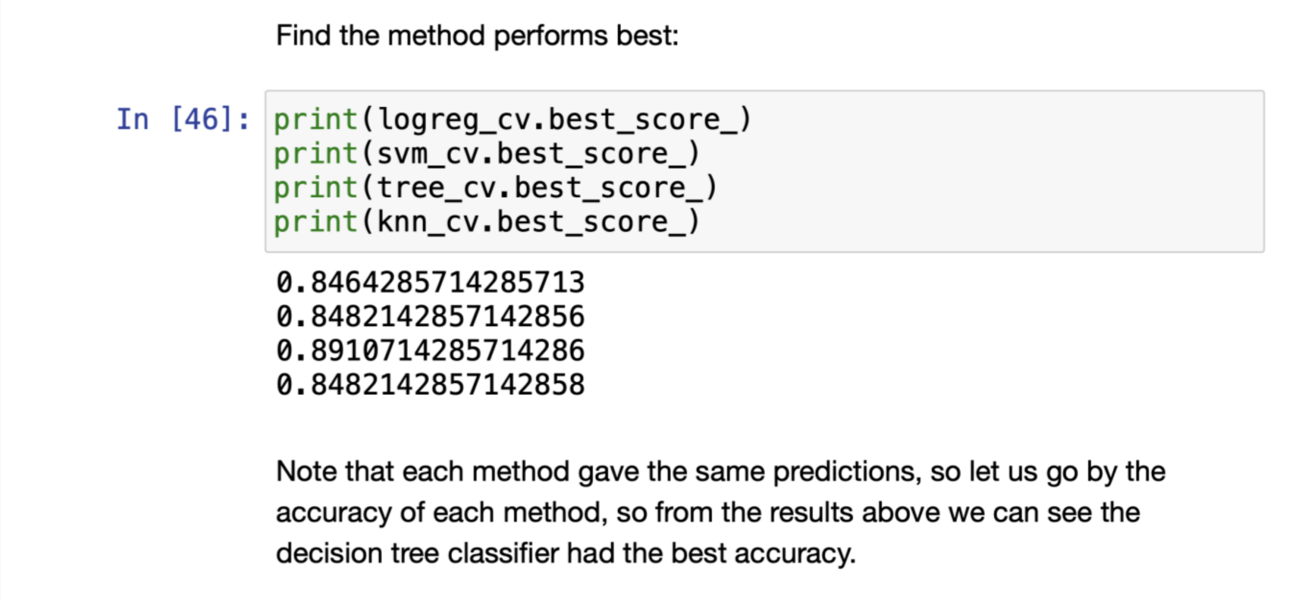
Figure 22. Scatter plot showing the booster versions of different payload mass greater than 5,500kg.

Section 5

Predictive Analysis (Classification)

Classification Accuracy

After performing each of our tests, we can see that we have obtained an accuracy score for each of our tests, the scores of each of our tests are the following: logistic regression of 0.846, support vector of 0.848, tree classifier of 0.891, and k nearest neighbours of 0.848, all rounded to the three decimal places. Hence, we can see that the tree classifier was the most accurate.



Find the method performs best:

```
In [46]: print(logreg_cv.best_score_)
         print(svm_cv.best_score_)
         print(tree_cv.best_score_)
         print(knn_cv.best_score_)
```

```
0.8464285714285713
0.8482142857142856
0.8910714285714286
0.8482142857142858
```

Note that each method gave the same predictions, so let us go by the accuracy of each method, so from the results above we can see the decision tree classifier had the best accuracy.

Figure 23. Accuracy result of each test.

Confusion Matrix

Note the dataset is spilt into training and test set, were we ended up with only 18 test samples. Then from the test set, we performed different types of tests and needed with decision tree classifier being the most accurate. From this we've plotted a confusion matrix where it has correctly predicted 12 observations that landed (12 True positives), 3 observations that did not land (3 True Negatives), 0 false negatives observations as it did not wrongly predict any successful landings, and it did have 3 false positives as it predicted wrongly for 3 observations that the outcome was successful.

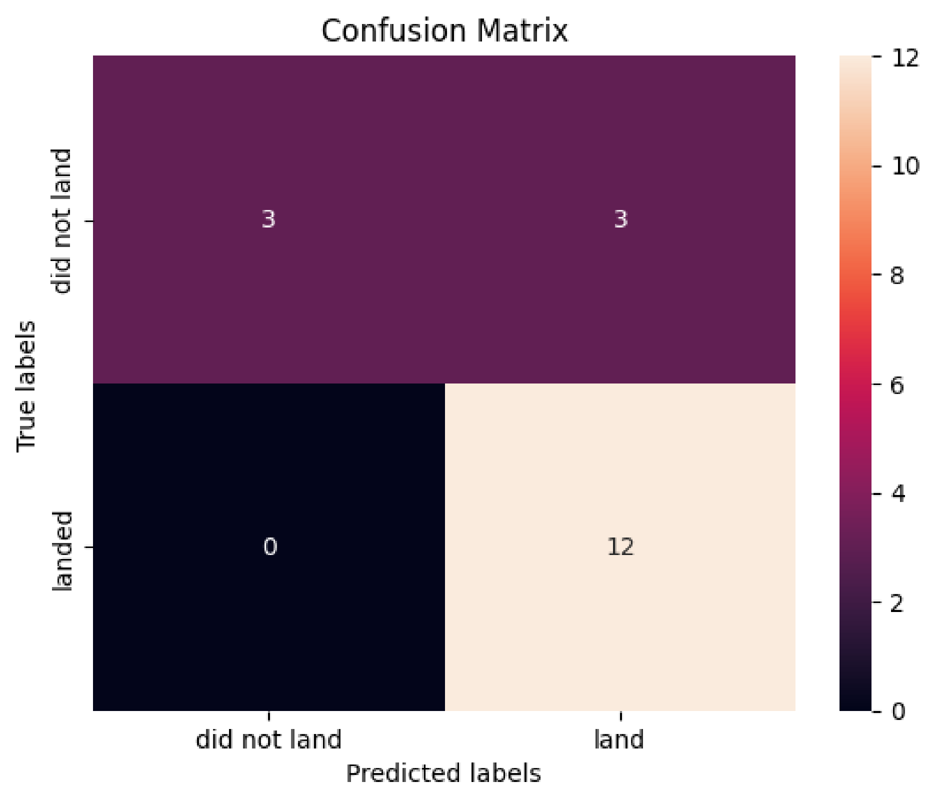


Figure 24. Confusion matrix of decision tree classifier.

Conclusions

- The larger the flight amount at a launch site, the greater the success rate at a launch site.
- Launch success rate started to increase in 2013 till 2020.
- KSC LC-39A had the most successful launches of any sites.
- For a successful mission, the mass of the payload should be considered as rockets with smaller payload had a higher success rate.
- Orbit type should also be considered because rockets launched to certain orbits, those being VLEO, ES-L1, GEO, HEO, and SSO, had different success rates compared to others.
- Launch sites that are in coastal cities for easy retrieval/recovery and far from busy areas like major highways and cities to minimize casualties in the event of a failure.
- Compared to other classification algorithms, decision tree classifiers had the best performance of approximately 87% making it a good model for landing outcome prediction.

Thank you!