IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Gustavo Guidini
2023/08/01

Outline

Executive Summary

- This study encompassed a comprehensive range of methodologies, including data collection, wrangling, exploration, and feature engineering. Additionally, it involved interactive visual analysis and predictive analysis to gain valuable insights from the data.
- Our model achieved an accuracy of 83.4% in predicting the success of a launch.

Introduction

- Our aim is to establish a robust competition with SpaceX, fostering profitability and a significant competitive presence. We seek to explore the feasibility of payload reuse, determine the success rate of targeted launches, and devise strategies for cost reduction.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - involved web scraping using BeautifulSoup.
- Perform data wrangling
  - was conducted using Pandas and Numpy.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - First split the data into training and test sets. Next, train the model using the training data and tune its hyperparameters using techniques like cross-validation. Finally, evaluate the model's predictive accuracy by assessing its performance on the test data.

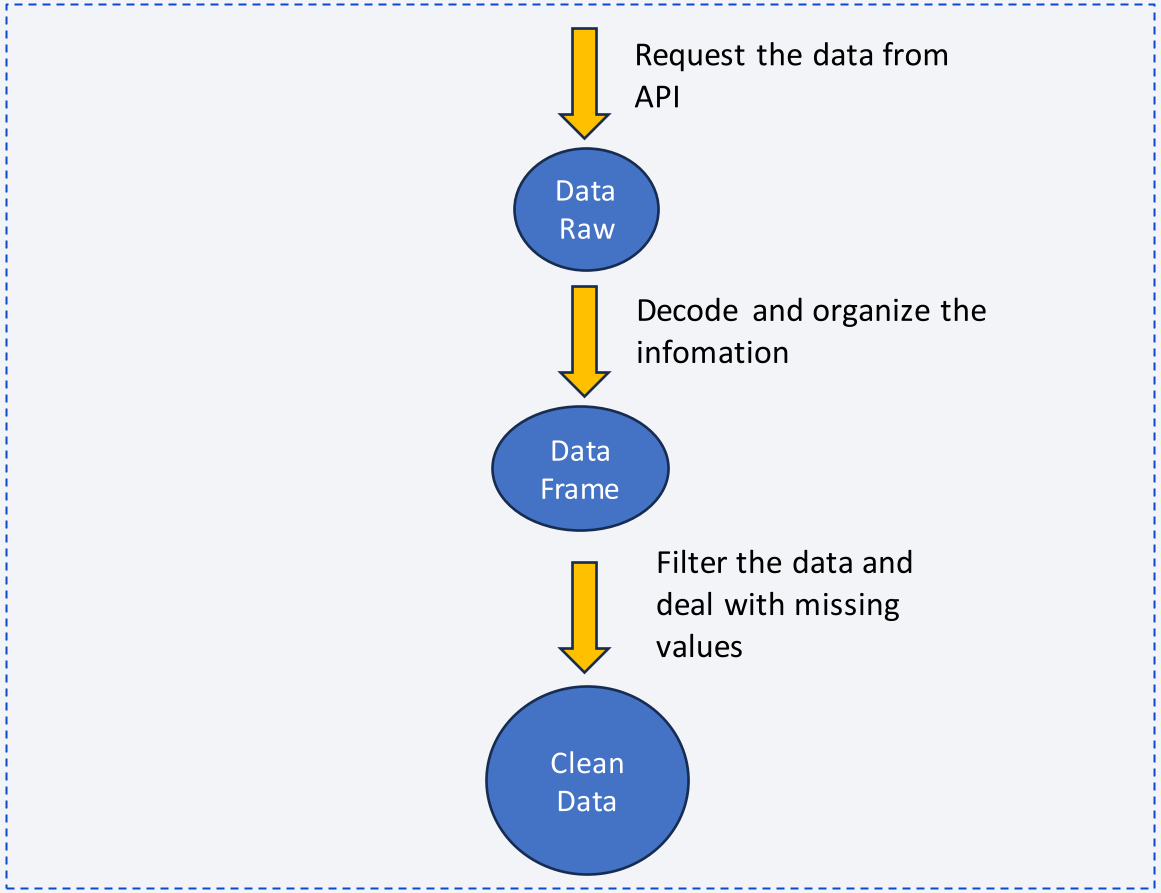
Data Collection

- The data sets were collected by initially retrieving raw data from the API or URL source. Subsequently, the data was transformed into an understandable format and structured into a data frame to facilitate cleaning and filtering operations. This approach allowed for efficient organization and preparation of the data for further analysis and processing.

Data Collection – SpaceX API

- Data Collection Flowchart

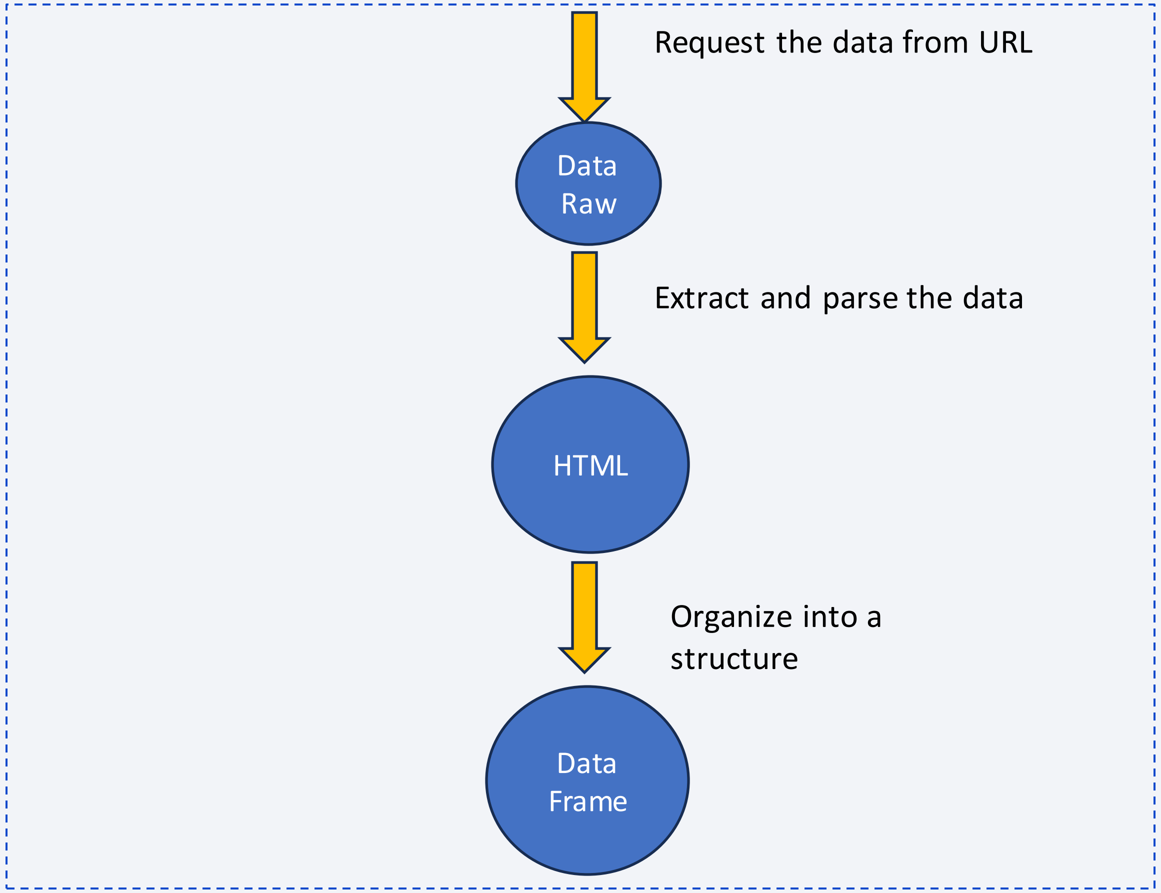
- https://github.com/Guipini/IBM/blob/main/Collecting%20Data/jupyter-labs-spacex-data-collection-api.ipynb


Request the data from API
Data Raw
Decode and organize the infomation
Data Frame
Filter the data and deal with missing values
Clean Data

Data Collection - Scraping

- WebScraping FlowChart

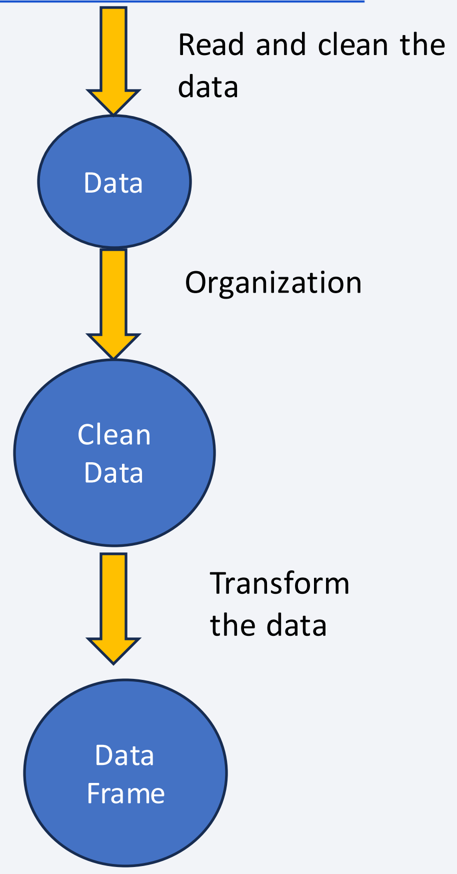
- https://github.com/Guipini/IBM/blob/main/Collecting%20Data/jupyter-labs-webscraping.ipynb


Request the data from URL
Data Raw
Extract and parse the data
HTML
Organize into a structure
Data Frame

Data Wrangling

- The data processing involved reading the data and conducting a thorough cleaning process, which included identifying and handling any missing values or null entries. Subsequently, an examination of the data was performed to determine the number of unique values and their respective frequencies. The next crucial step was to integrate these unique values with the relevant data information and organize it into a structured Data Frame, enabling efficient data management and analysis.

- https://github.com/Guipini/IBM/blob/main/Data%20Wrangling/labs-jupyter-spacex-Data%20wrangling(1).ipynb



EDA with Data Visualization

- A series of charts were plotted to observe and analyze various relationships within the data. The first chart focused on the correlation between Flight Number and Payload Mass, revealing a positive relationship. To expand the scope, another chart was created to examine the connection between Flight Number and Launch Site, demonstrating a correlation between higher flight numbers and success rates. Further insights were sought by plotting Launch Site against Payload Mass, which highlighted variations in success rates based on different launch sites and payload masses. A more intuitive visualization was presented, illustrating the success rates for each orbit group. Next, the relationship between Flight Number and Orbit type was explored, revealing a single notable correlation. Additionally, two more charts were generated to investigate the relationships between Payload Mass and Orbit type, displaying two distinct correlations. Lastly, a trending plot showcased the yearly success rate, illustrating a clear upward trend over time.
- https://github.com/Guipini/IBM/blob/main/Exploratory%20Data%20Analysis%20Using%20Pandas%20and%20Matplotlib/jupyter-labs-eda-dataviz.ipynb

EDA with SQL

- Display unique names of launch sites and after records where launch site begin with 'CCA'
- Present the sum of mass carried by boosters launched by NASA
- illustrate the average payload mass carried by booster version F v1.1
- List the date when the first successful landing outcome in ground pad was achieved.
- Query the booster which have success in drone ship and mass between 4000 and 6000
- Display total number of successful and failure mission outcomes
- Present the names of the booster version which have carried max payload mass
- List records with month, names, outcomes, version, site for the months in year 2015
- Rank the count of landing outcomes between 2010-06-04 and 2017-03-20 in descending order
- https://github.com/Guipini/IBM/blob/main/Exploratory%20Analysis%20using%20SQL/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- On the initial Folium map, we incorporated a yellow circle to pinpoint the NASA Space Center in Houston, and subsequently, we added red circles to denote the launch site locations. To illustrate the launch outcomes, we utilized green markers for successful launches and red markers for failed ones. In order to provide a clearer understanding of the locations, we included lines that indicated the distances between the various points. Additionally, we enhanced the map by adding roads, railroads, and other landmarks to provide contextual information and aid in visualizing the spatial relationships between the launch sites and other key features. These map objects were added to facilitate effective visualization, analysis, and interpretation of the data and its geographical context.
- https://nbviewer.org/github/Guipini/IBM/blob/main/Interactive%20Visual%20Analytics%20and%20Dashboard/lab_jupyter_launch_site_location.ipynb

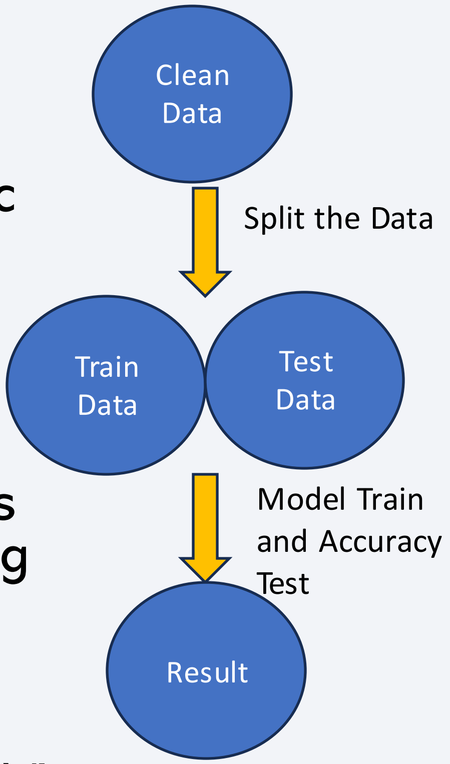
Build a Dashboard with Plotly Dash

- In the dashboard, we have included two crucial plots for data visualization. The first plot is a pie chart that effectively illustrates the success rates across all Launch Sites, providing a concise overview of mission outcomes at different locations. This chart enables quick comparisons and insights into any variations or patterns in success rates among launch sites. The second plot is a scatter plot, displaying the relationship between Payload Mass and the Booster Version, allowing us to explore the correlation between these variables. The inclusion of these plots serves to present essential information in an easily digestible format, enhancing data exploration and facilitating informed decision-making within the dashboard.
- https://github.com/Guipini/IBM/blob/main/Interactive%20Visual%20Analytics%20and%20Dashboard/spacex_dash_app.py

Predictive Analysis (Classification)

- The process of building, evaluating, and finding the best performing classification model began with obtaining cleaned and standardized data. Subsequently, the data was split into training and test sets to enable model training and evaluation. Logistic Regression, Support Vector Machine, Decision Tree Classifier, and KNN models were tested, and their respective accuracies were assessed. While the Logistic Regression and Support Vector Machine models showed promising results, they had some false positives. The Decision Tree Classifier demonstrated better accuracy in the training data but performed worse in the test data. The KNN model exhibited similar performance in accuracy and score. Finally, a thorough comparison of all models was conducted, and the Decision Tree Classifier emerged as the best performer, achieving the highest accuracy and score among all the tested models.

- https://github.com/Guipini/IBM/blob/main/Predictive%20Analysis%20(Classification)/IBM-DS0321EN-SkillsNetwork_labs_module_4_SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb



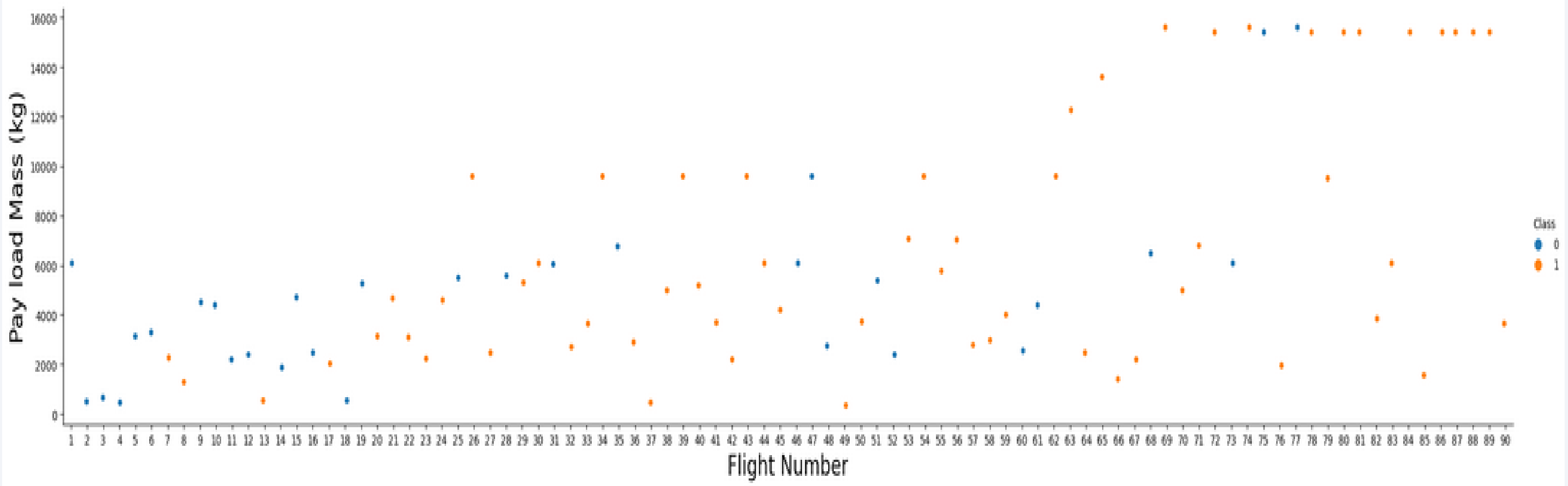
Results

- Certain factors, such as payload mass range and orbit type, have an impact on the success rate.
- Our findings emphasize the crucial role of launch location in ensuring a successful launch.
- We have developed a reliable model capable of predicting launch success with accuracy.

Section 2

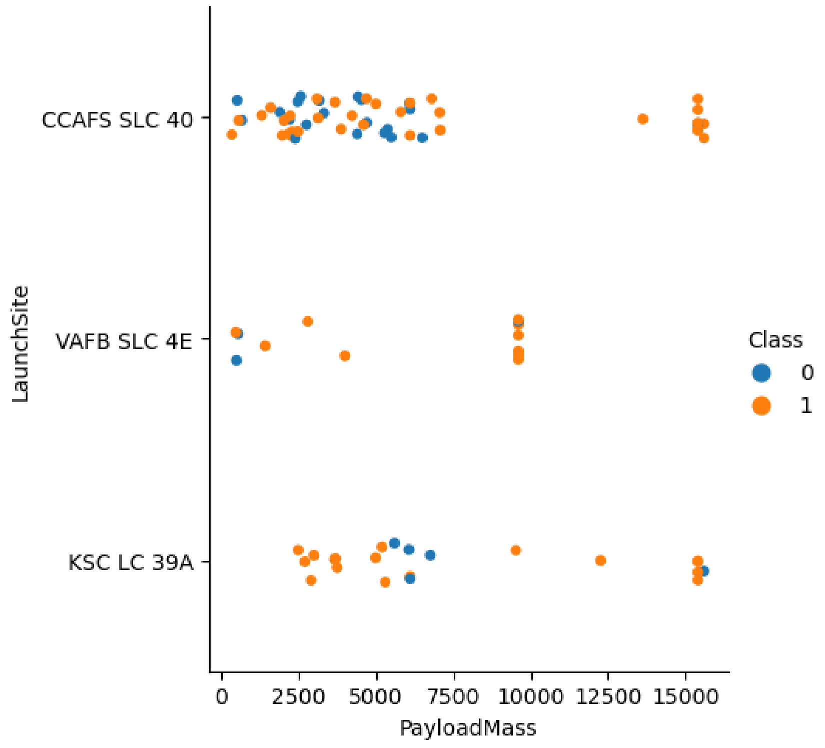
Insights drawn from EDA

Flight Number vs. Launch Site



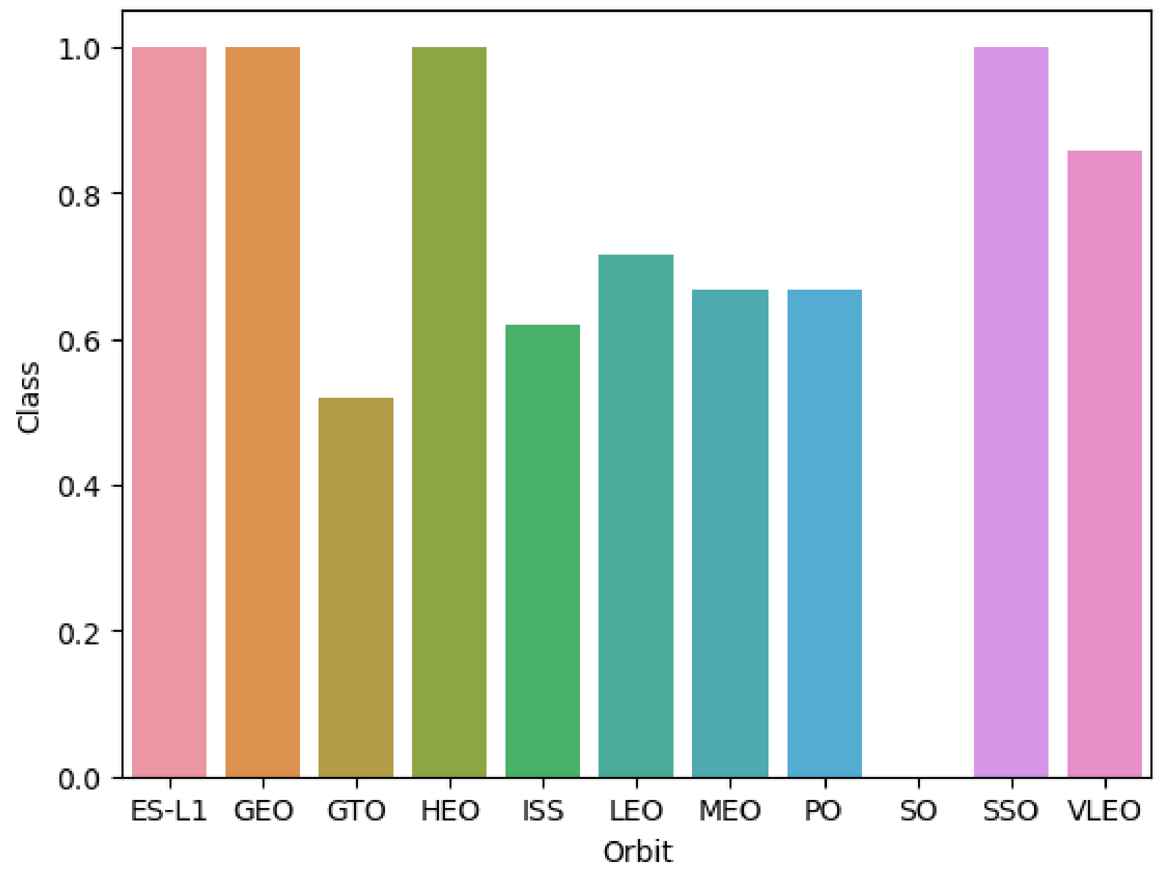
We see that different launch sites have different success rates. `CCAFS LC-40`, has a success rate of 60 %, while `KSC LC-39A` and `VAFB SLC 4E` has a success rate of 77%.

Payload vs. Launch Site



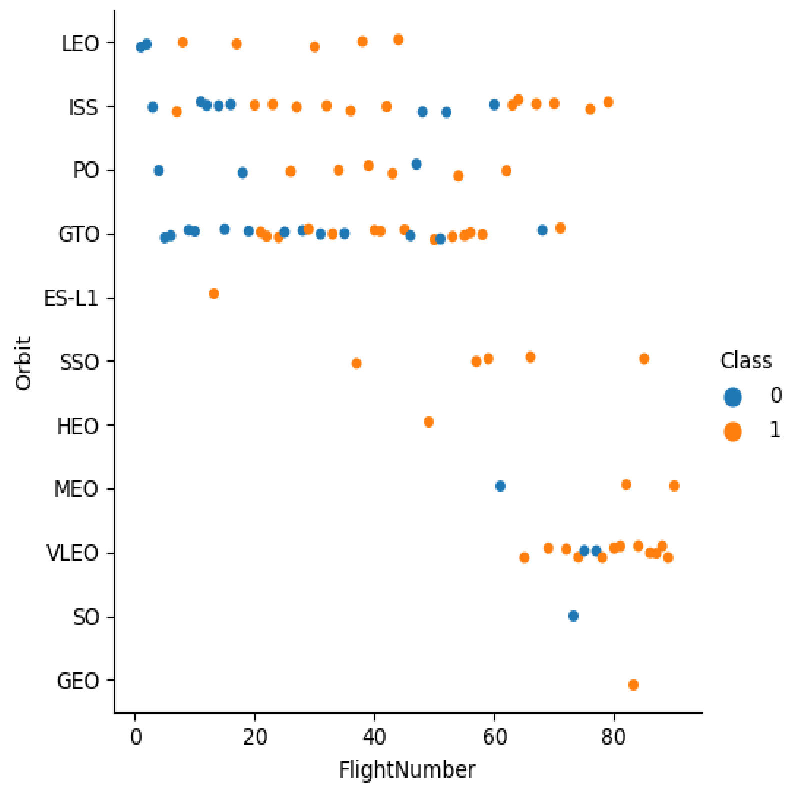
Now if you observe Payload Vs. Launch Site scatter point chart you will find for the VAFB-SLC launchsite there are no rockets launched for heavypayload mass(greater than 10000).

Success Rate vs. Orbit Type



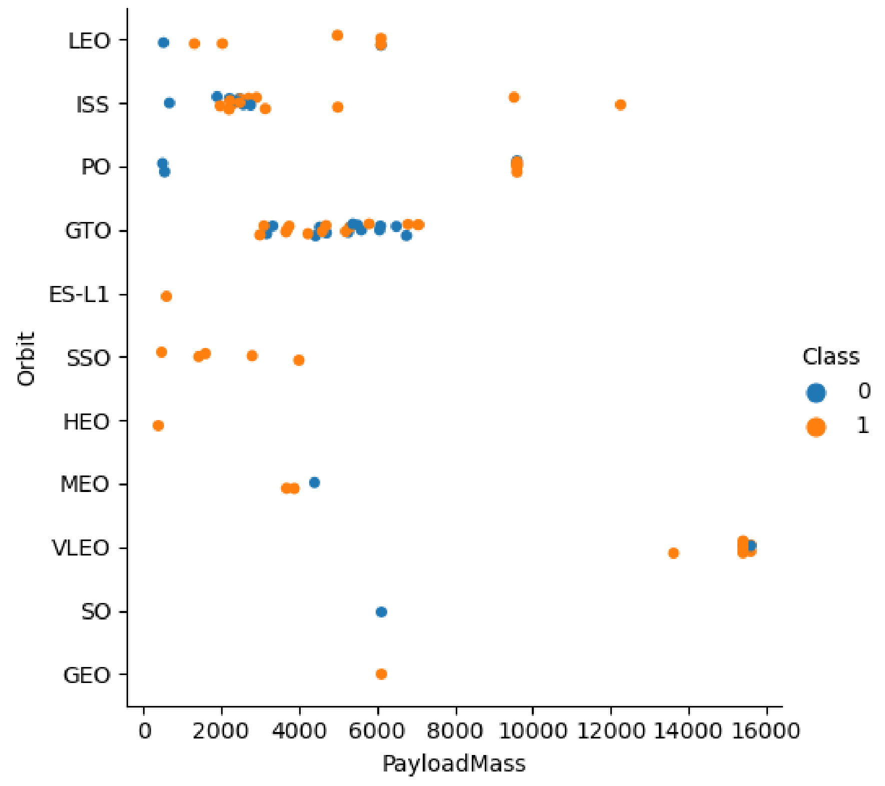
Analyze the ploted bar chart try to find which orbits have high sucess rate.

Flight Number vs. Orbit Type



You should see 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 when in GTO orbit.

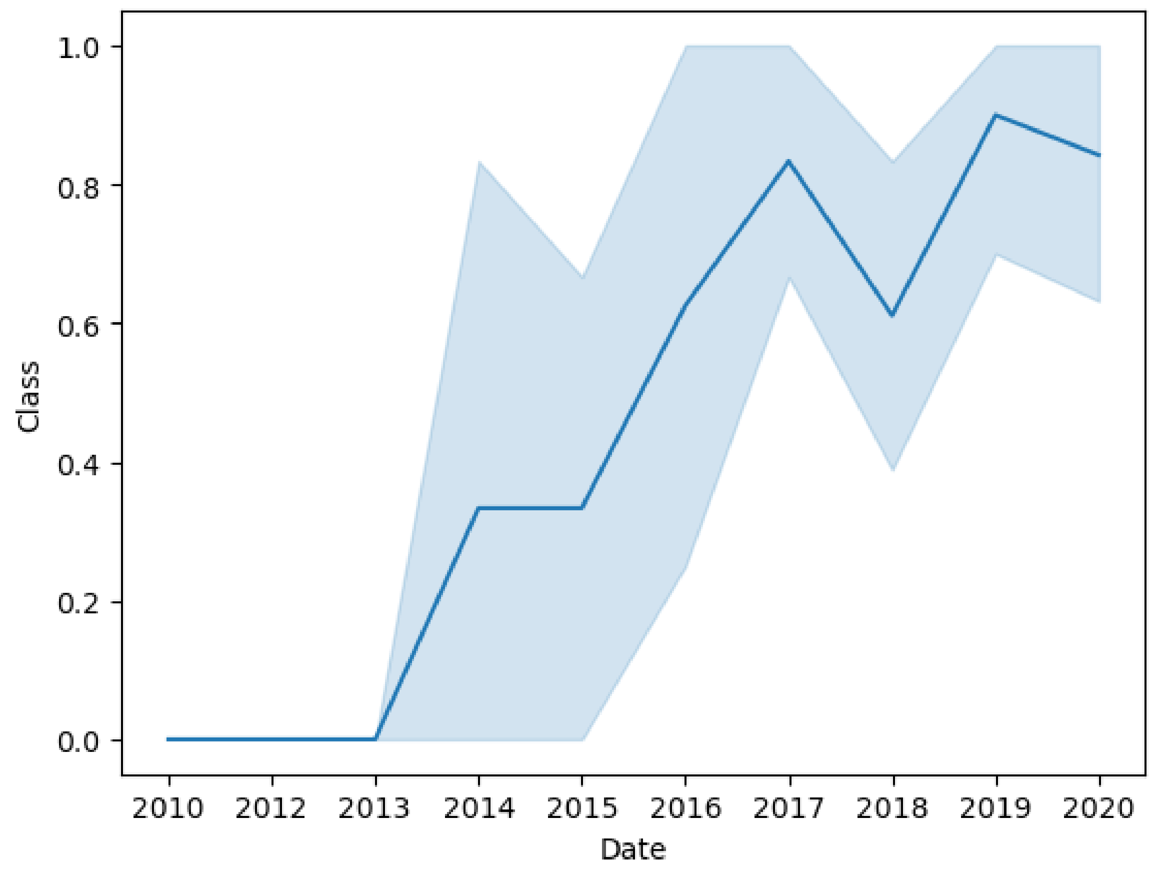
Payload vs. Orbit Type



With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS.

However for GTO we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission) are both there here.

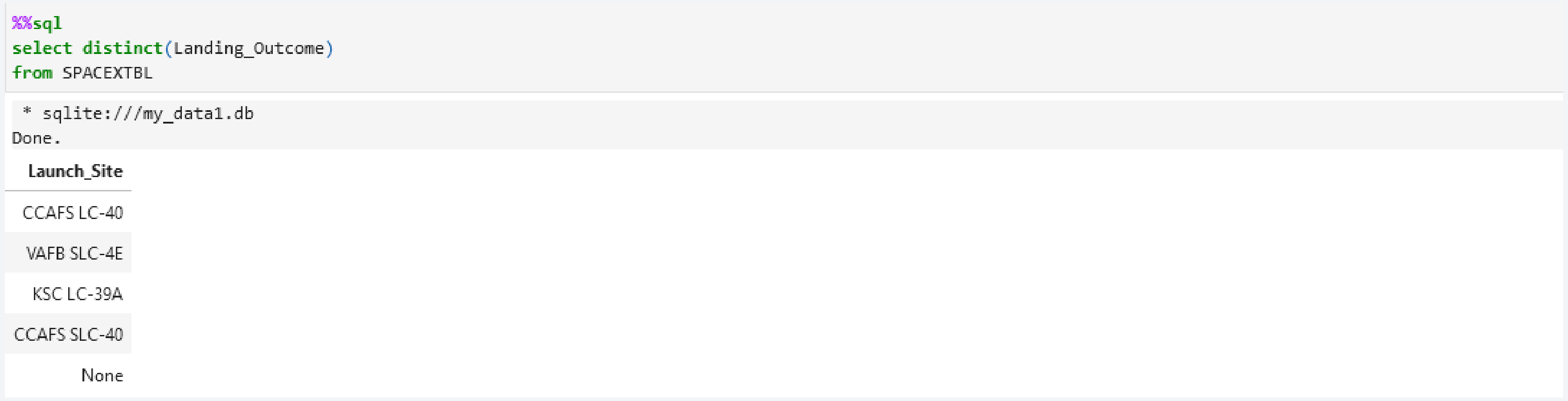
Launch Success Yearly Trend



you can observe that the sucess rate since 2013 kept increasing till 2020

All Launch Site Names

- You should see all the names of launch sites in the space mission.



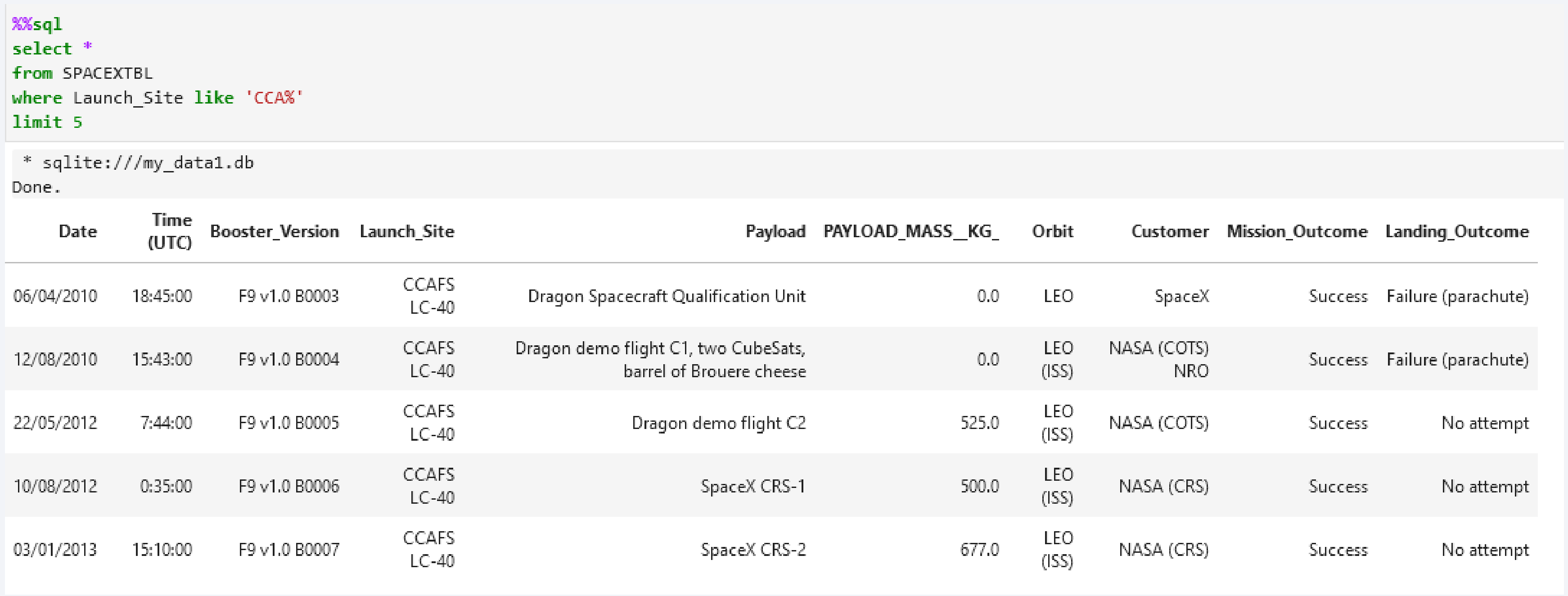
```
%%sql
select distinct(Landing_Outcome)
from SPACEXTBL
```

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

Launch_Site
CCAFS LC-40
VAFB SLC-4E
KSC LC-39A
CCAFS SLC-40
None

Launch Site Names Begin with 'CCA'

- Results of 5 records where launch sites begin with `CCA`



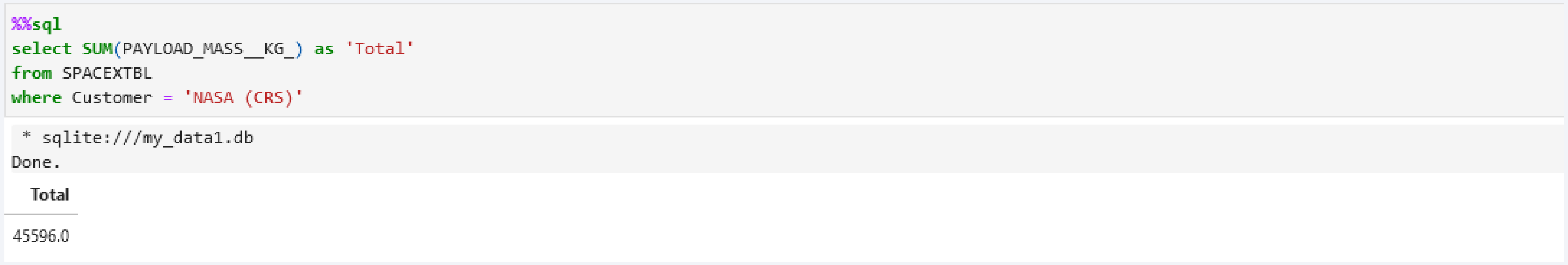
```
%%sql
select *
from SPACEXTBL
where Launch_Site like 'CCA%'
limit 5
```

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
06/04/2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0.0	LEO	SpaceX	Success	Failure (parachute)
12/08/2010	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0.0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
22/05/2012	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525.0	LEO (ISS)	NASA (COTS)	Success	No attempt
10/08/2012	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500.0	LEO (ISS)	NASA (CRS)	Success	No attempt
03/01/2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677.0	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

- Calculation of the total payload mass carried by boosters from NASA



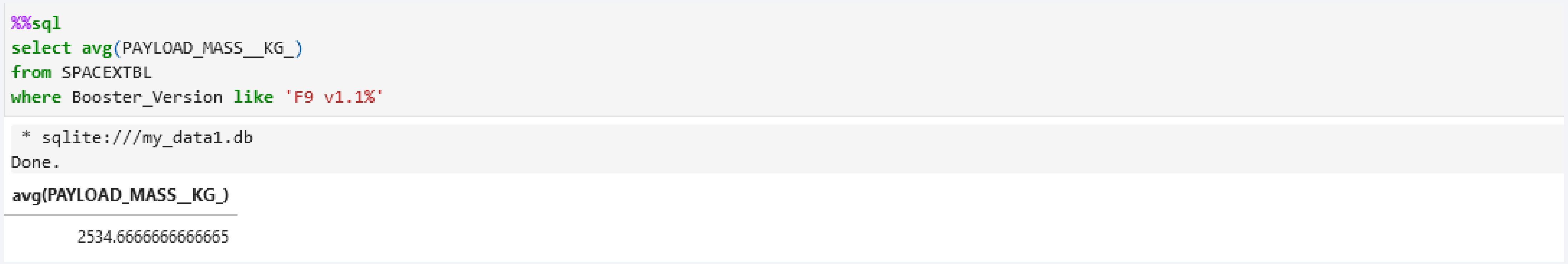
```
%%sql
select SUM(PAYLOAD_MASS__KG_) as 'Total'
from SPACEXTBL
where Customer = 'NASA (CRS)'
```

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

Total
45596.0

Average Payload Mass by F9 v1.1

- Result of the average payload mass carried by booster version F9 v1.1



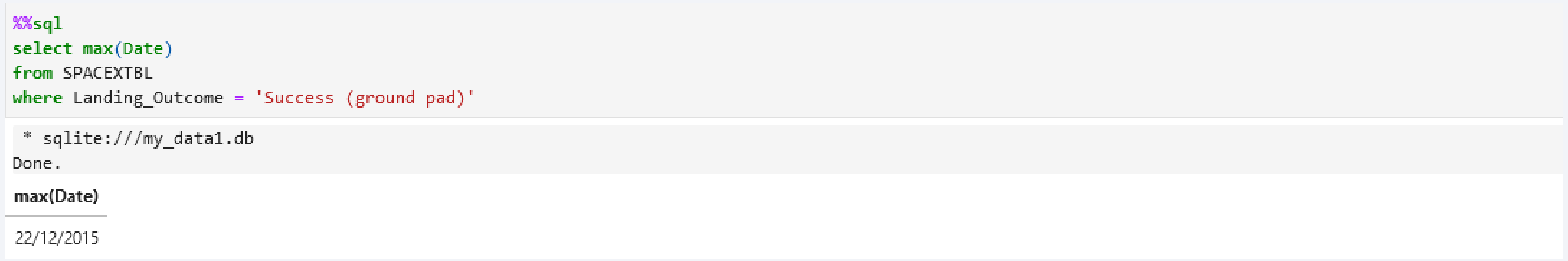
```
%%sql
select avg(PAYLOAD_MASS__KG_)
from SPACEXTBL
where Booster_Version like 'F9 v1.1%'
```

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

avg(PAYLOAD_MASS__KG_)
2534.6666666666665

First Successful Ground Landing Date

- Presenting the dates of the first successful landing outcome on ground pad



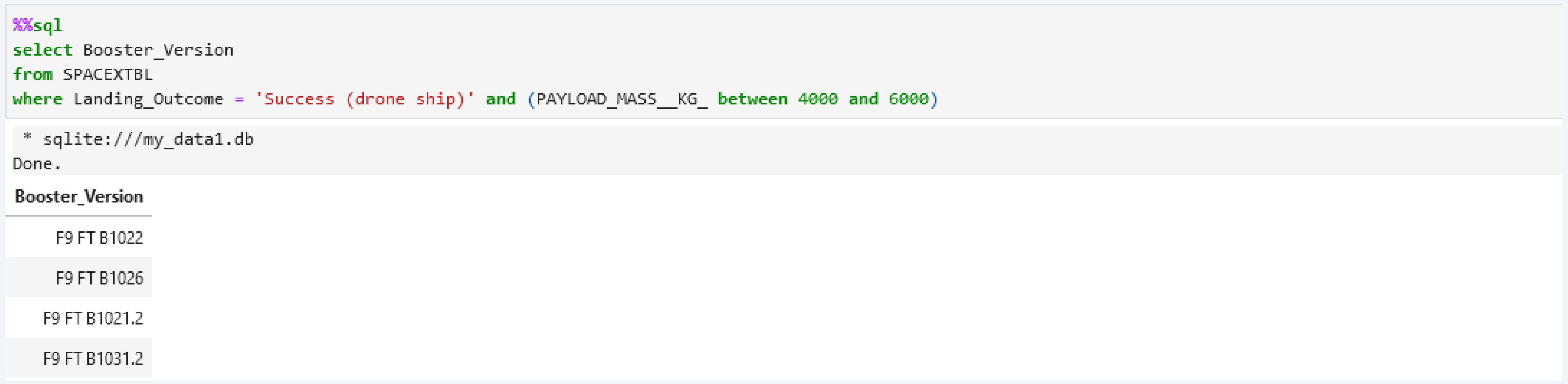
```
%%sql
select max(Date)
from SPACEXTBL
where Landing_Outcome = 'Success (ground pad)'
```

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

```
max(Date)
22/12/2015
```

Successful Drone Ship Landing with Payload between 4000 and 6000

- List the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000



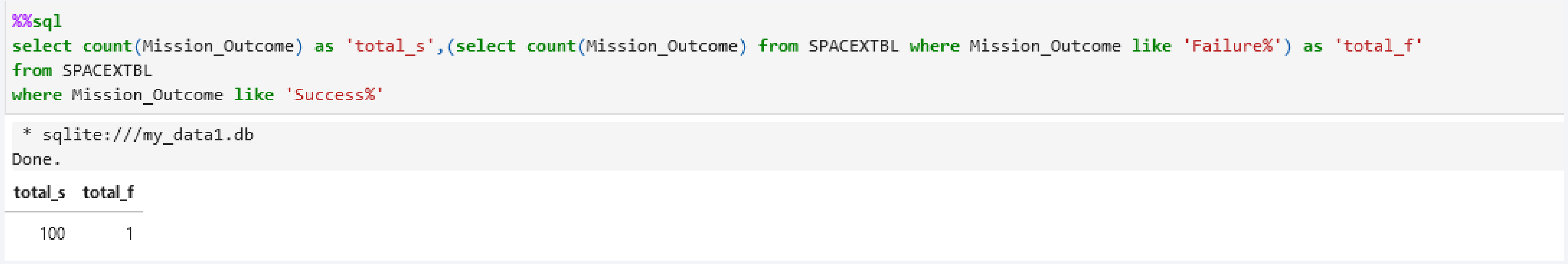
```
%%sql
select Booster_Version
from SPACEXTBL
where Landing_Outcome = 'Success (drone ship)' and (PAYLOAD_MASS__KG_ between 4000 and 6000)
```

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

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

Total Number of Successful and Failure Mission Outcomes

- Result of the total number of successful and failure mission outcomes



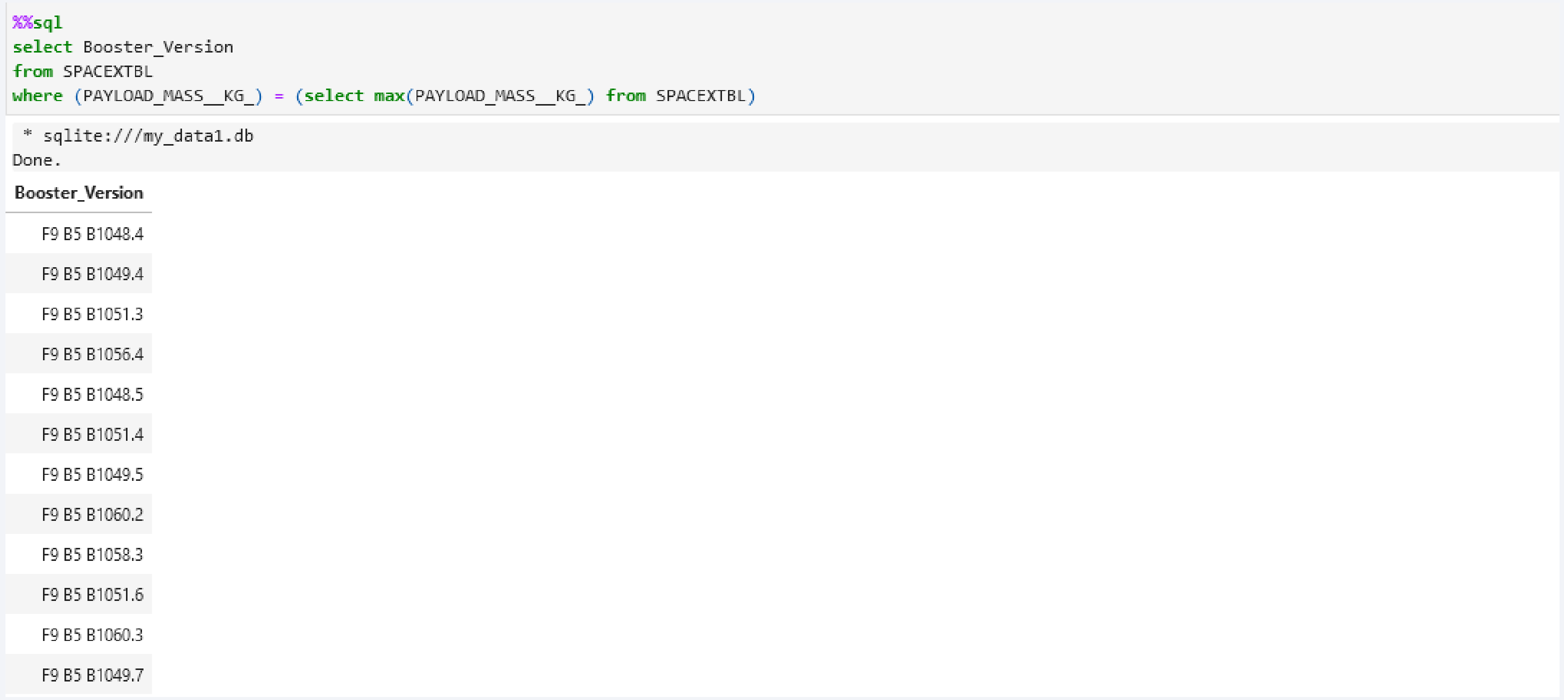
```
%%sql
select count(Mission_Outcome) as 'total_s',(select count(Mission_Outcome) from SPACEXTBL where Mission_Outcome like 'Failure%') as 'total_f'
from SPACEXTBL
where Mission_Outcome like 'Success%'
```

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

total_s	total_f
100	1

Boosters Carried Maximum Payload

- Listing the names of the booster which have carried the maximum payload mass



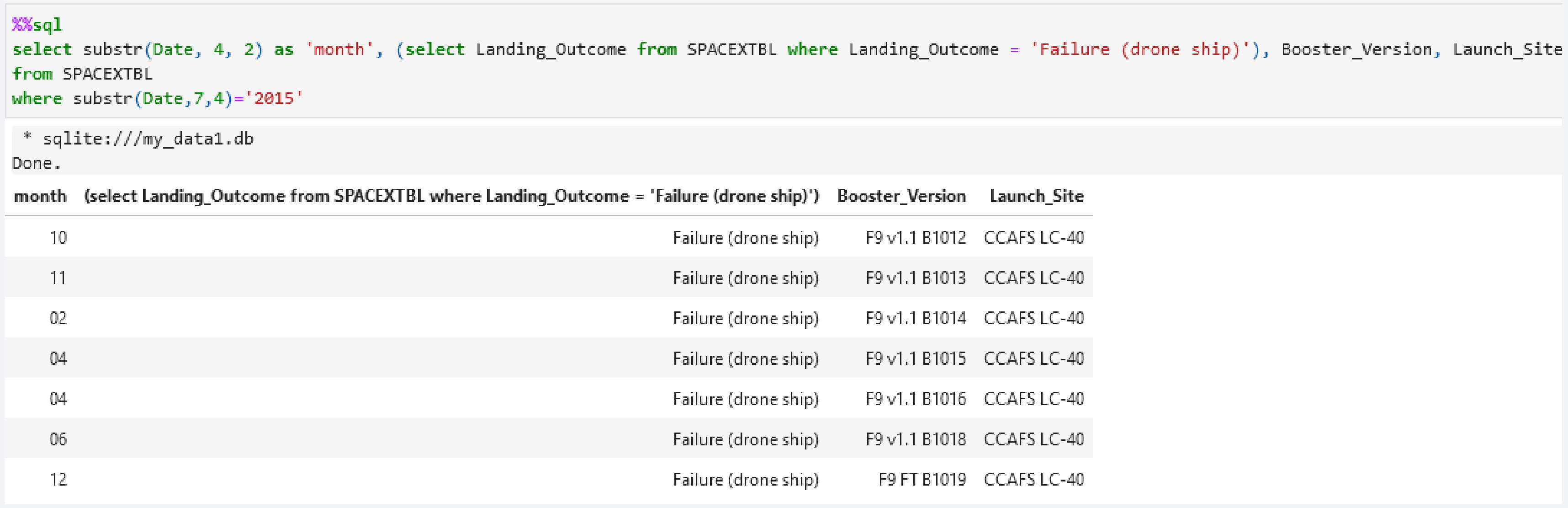
```
%%sql
select Booster_Version
from SPACEXTBL
where (PAYLOAD_MASS__KG_) = (select max(PAYLOAD_MASS__KG_) from SPACEXTBL)
```

```
 * 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 of failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015



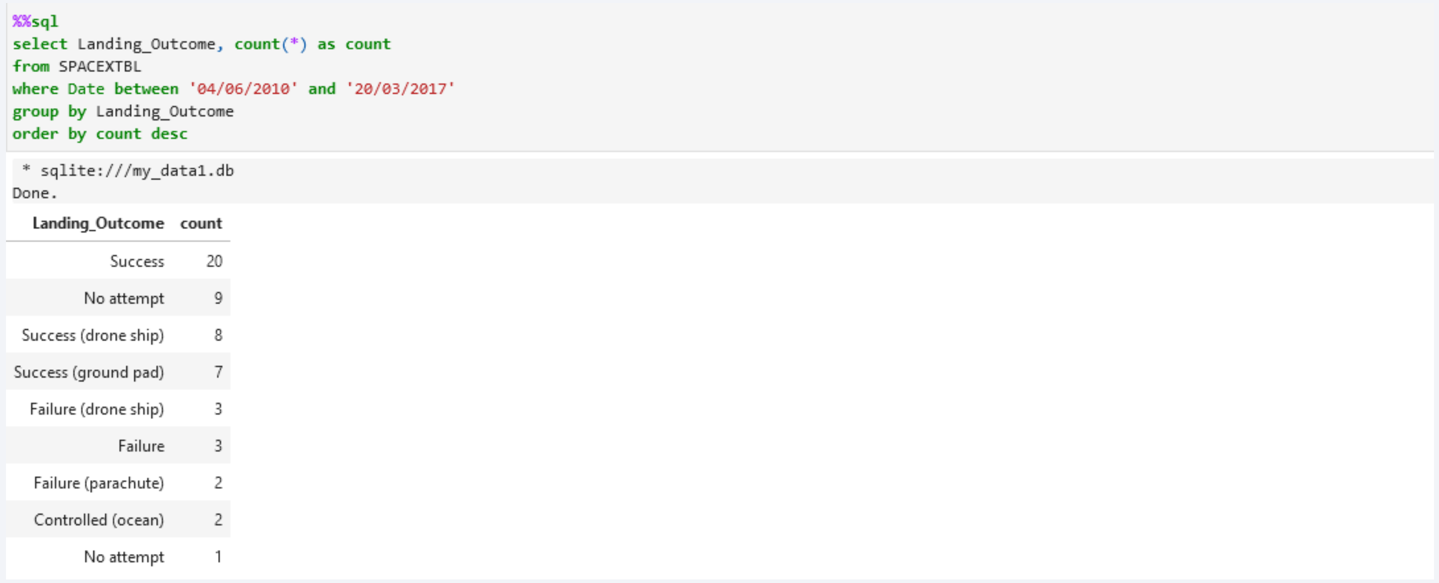
```
%%sql
select substr(Date, 4, 2) as 'month', (select Landing_Outcome from SPACEXTBL where Landing_Outcome = 'Failure (drone ship)'), Booster_Version, Launch_Site
from SPACEXTBL
where substr(Date,7,4)='2015'
```

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

month	(select Landing_Outcome from SPACEXTBL where Landing_Outcome = 'Failure (drone ship)')	Booster_Version	Launch_Site
10	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
11	Failure (drone ship)	F9 v1.1 B1013	CCAFS LC-40
02	Failure (drone ship)	F9 v1.1 B1014	CCAFS LC-40
04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40
04	Failure (drone ship)	F9 v1.1 B1016	CCAFS LC-40
06	Failure (drone ship)	F9 v1.1 B1018	CCAFS LC-40
12	Failure (drone ship)	F9 FT B1019	CCAFS LC-40

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

- Ranking 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



```
%%sql
select Landing_Outcome, count(*) as count
from SPACEXTBL
where Date between '04/06/2010' and '20/03/2017'
group by Landing_Outcome
order by count desc
```

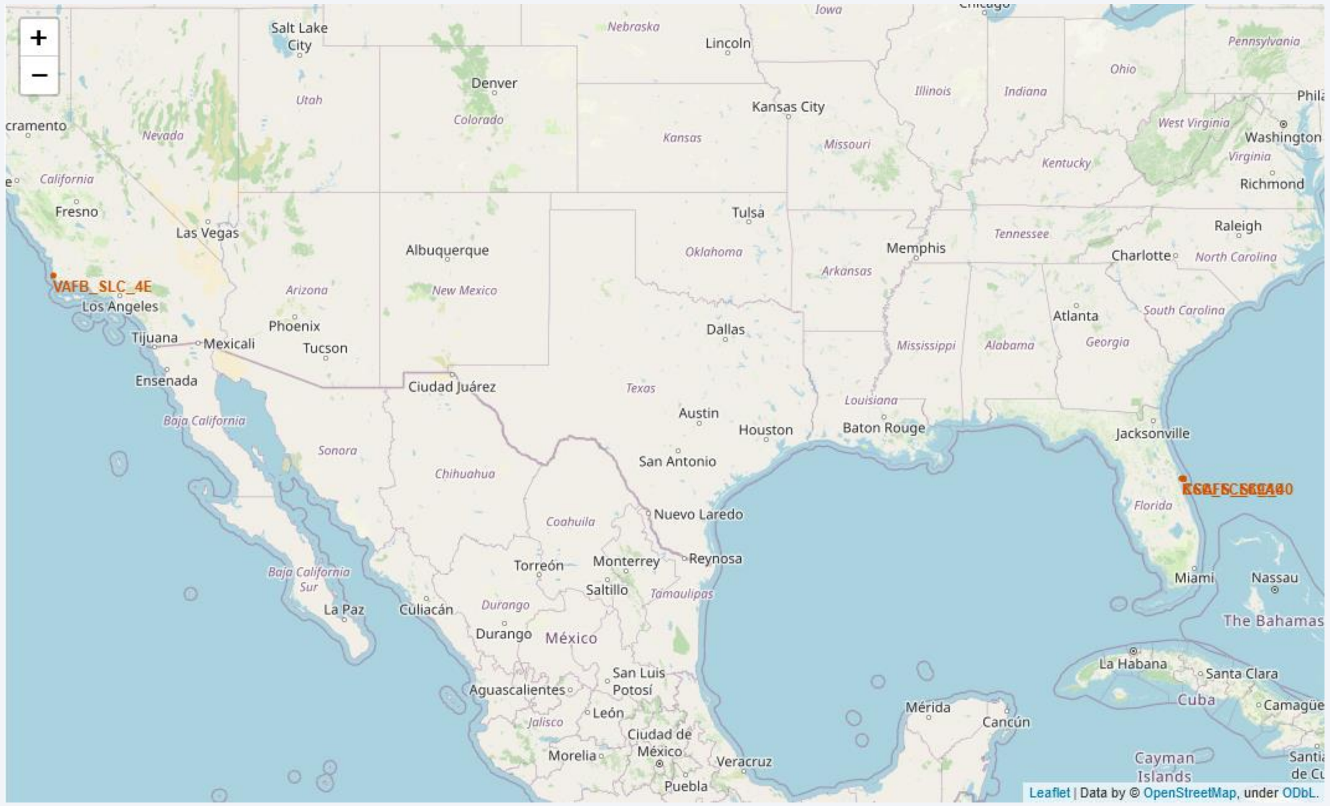
```
 * sqlite:///my_data1.db
Done.
```

Landing_Outcome	count
Success	20
No attempt	9
Success (drone ship)	8
Success (ground pad)	7
Failure (drone ship)	3
Failure	3
Failure (parachute)	2
Controlled (ocean)	2
No attempt	1

Section 3

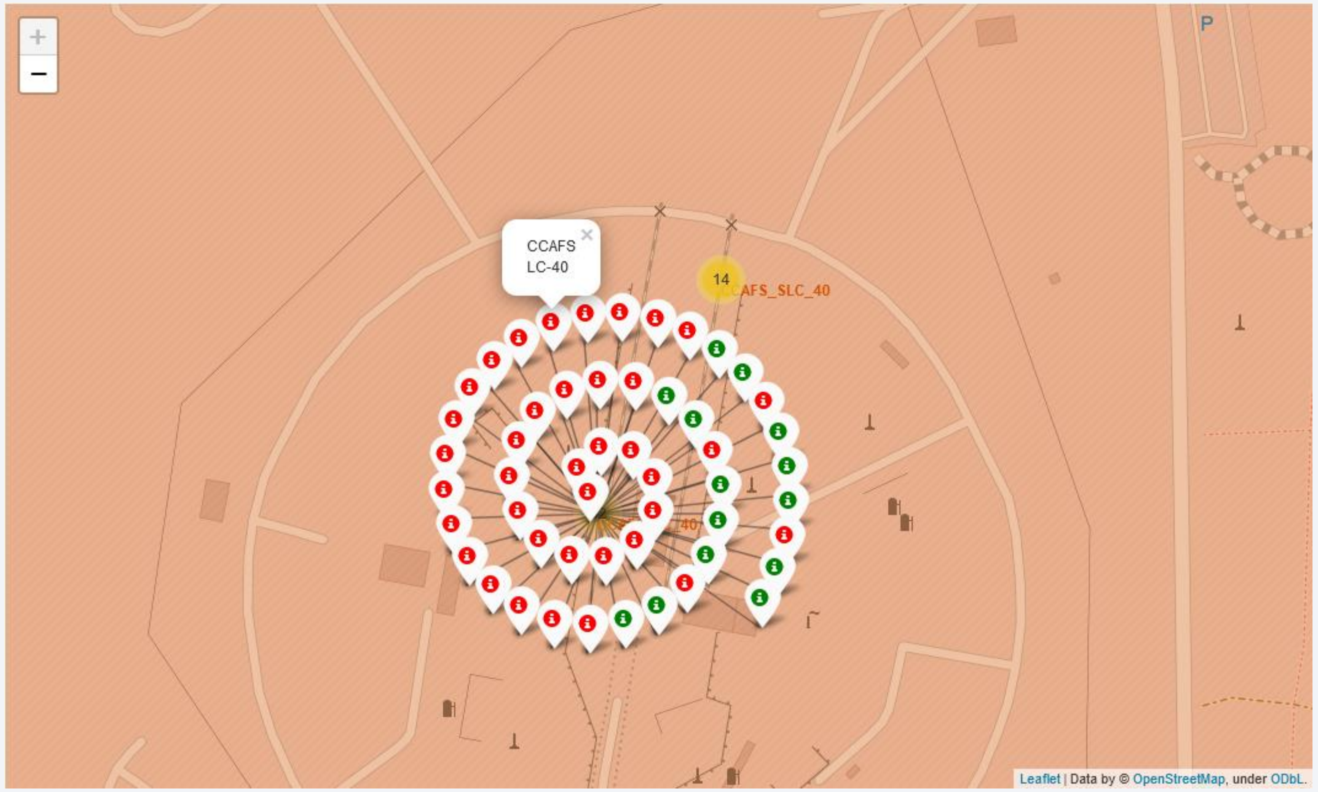
Launch Sites Proximities Analysis

All launch Sites in a map



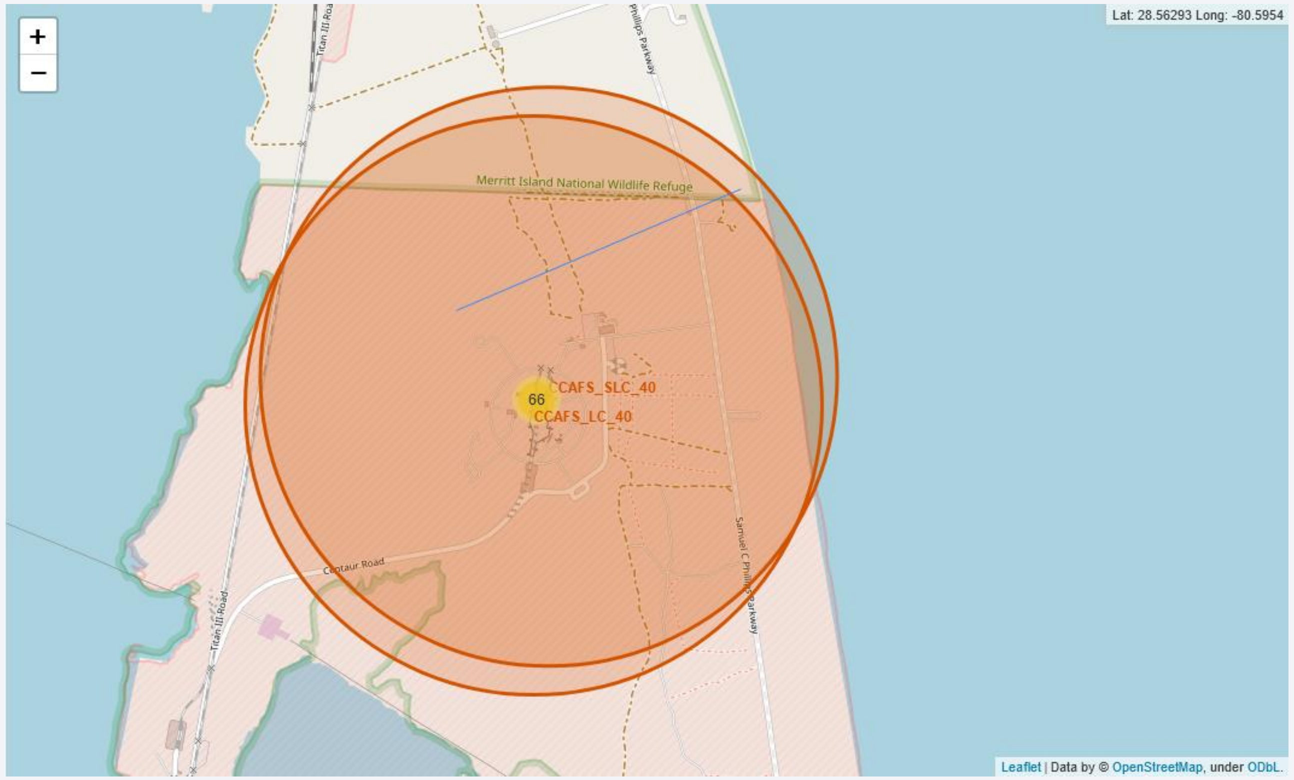
- Three significant launch sites are indicated.

Launch Outcomes

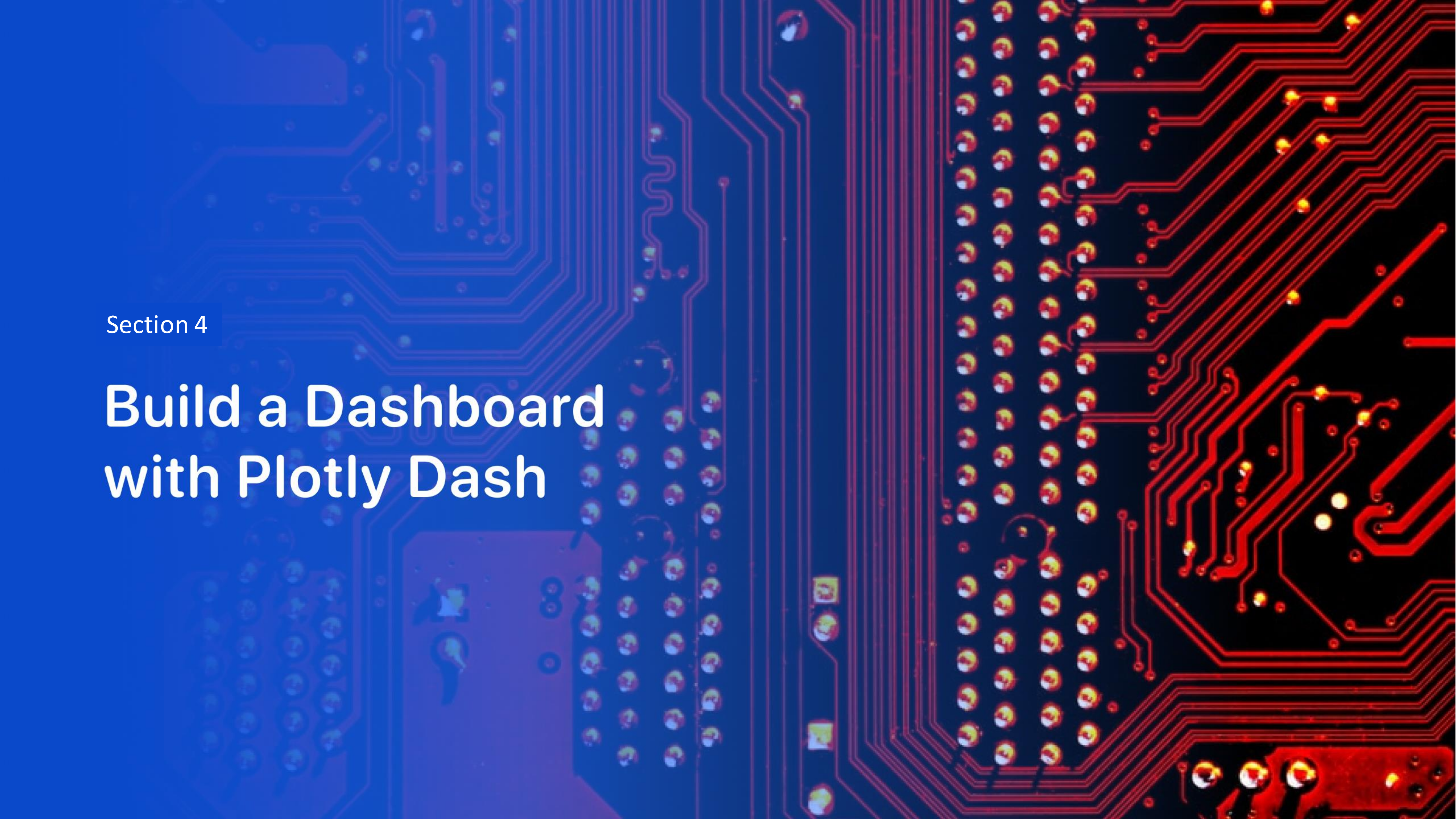


- We are showing a selected launch site, the green mark is a successful launch and in red is the failure.

Distance between Coast line and Launch site


Lat: 28.56293 Long: -80.5954
Merritt Island National Wildlife Refuge
CCAFS_SLC_40
66
CCAFS_LC_40
Centaur Road
Titan III Road
Phillips Parkway
Samuel C Phillips Parkway
Leaflet | Data by © OpenStreetMap, under ODbL.

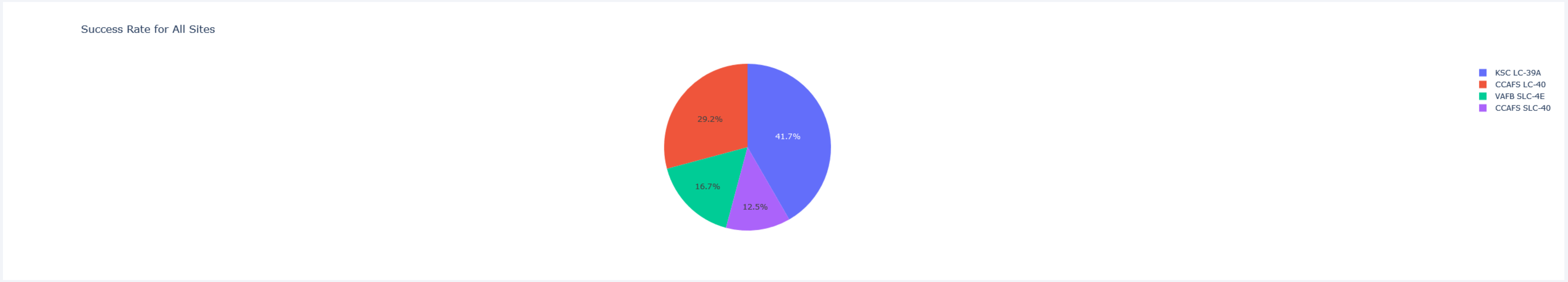
- The blue line indicates the distance, with a railway and highway running between them.



Section 4

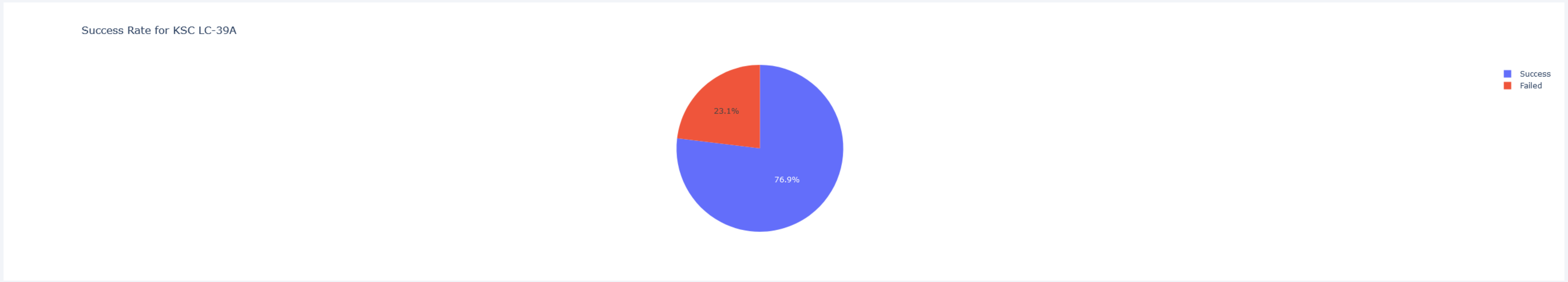
Build a Dashboard with Plotly Dash

SpaceX Launch Records in Pie chart



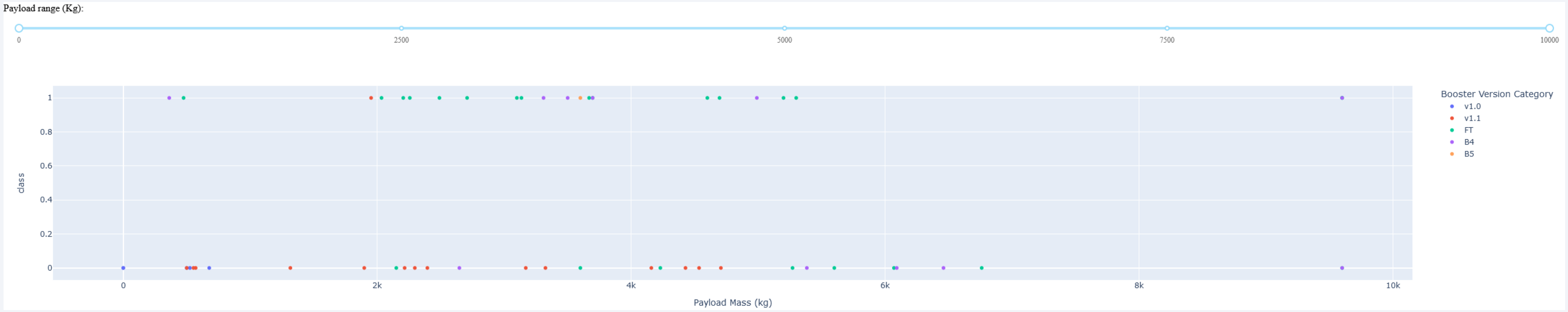
- The disparity in launch site success rates is evident, with a notably higher rate observed in comparison to other cases.

SpaceX Success Rate for KSC Launch Site



- It is important to note a higher success rate in this launch site

SpaceX Success Rate in relation to Payload Mass



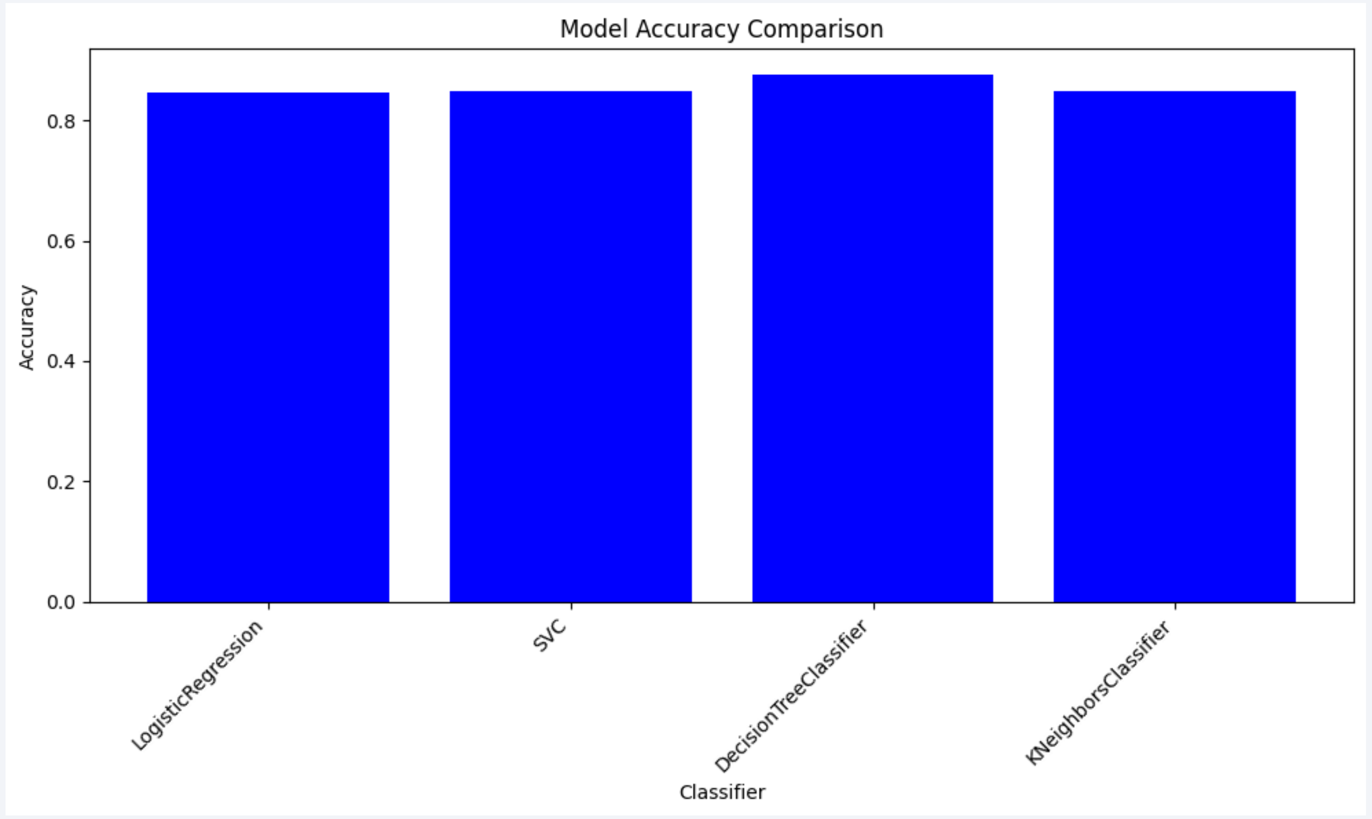
- A heightened rate of success is observed within the payload mass range of 2000 to 4000 units.

Section 5

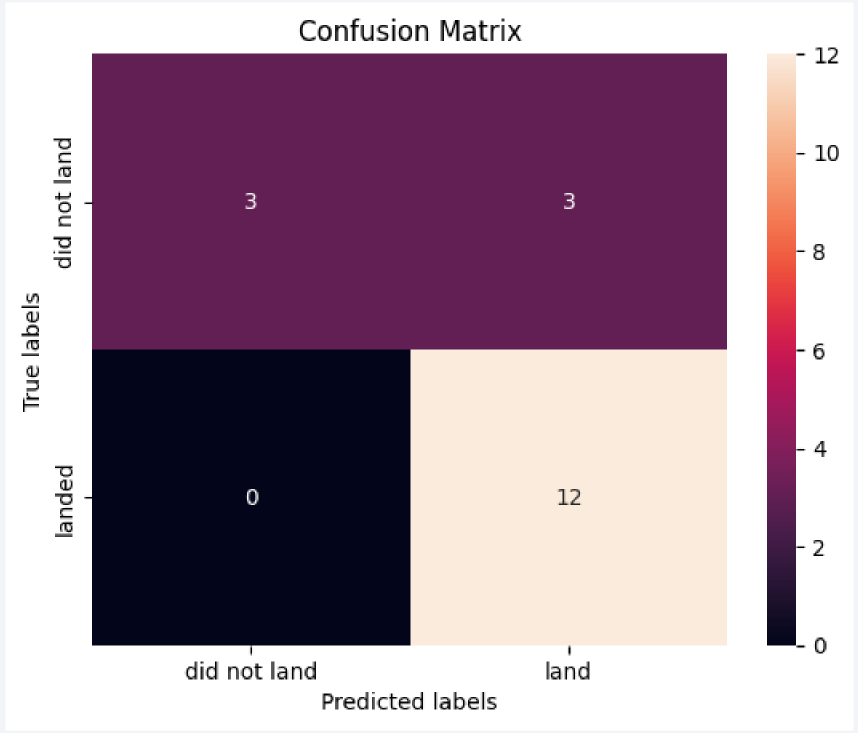
Predictive Analysis (Classification)

Classification Accuracy

- The Decision Tree Classifier exhibits a slightly higher accuracy, as evident from the results.



Confusion Matrix



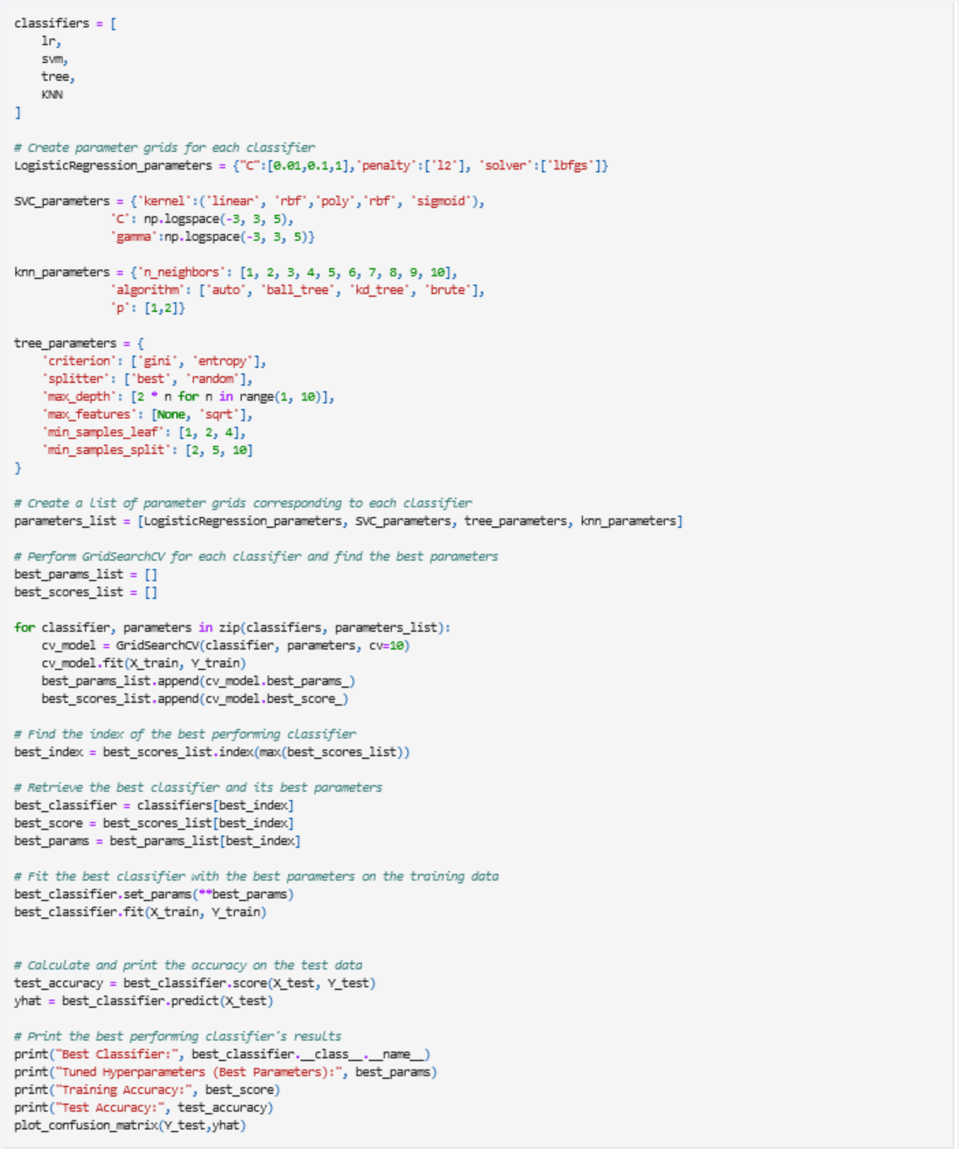
- We have three false positive in this graph and all the others are correct.

Conclusions

- It is conceivable to engage in healthy competition with SpaceX by meticulous planning and precise specification of launch location and payload mass.
- Employing the predictive model provides a deeper understanding of the likelihood of success.
- When feasible, focus on a particular launch site that exhibits a higher success rate.

Thank you!

Appendix



```python
classifiers = [
    lr,
    svm,
    tree,
    KNN
]

# Create parameter grids for each classifier
LogisticRegression_parameters = {"C":[0.01,0.1,1],'penalty':['l2'], 'solver':['lbfgs']}

SVC_parameters = {'kernel':('linear', 'rbf','poly','rbf', 'sigmoid'),
              'C': np.logspace(-3, 3, 5),
              'gamma':np.logspace(-3, 3, 5)}

knn_parameters = {'n_neighbors': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10],
              'algorithm': ['auto', 'ball_tree', 'kd_tree', 'brute'],
              'p': [1,2]}

tree_parameters = {
    'criterion': ['gini', 'entropy'],
    'splitter': ['best', 'random'],
    'max_depth': [2 * n for n in range(1, 10)],
    'max_features': [None, 'sqrt'],
    'min_samples_leaf': [1, 2, 4],
    'min_samples_split': [2, 5, 10]
}

# Create a list of parameter grids corresponding to each classifier
parameters_list = [LogisticRegression_parameters, SVC_parameters, tree_parameters, knn_parameters]

# Perform GridSearchCV for each classifier and find the best parameters
best_params_list = []
best_scores_list = []

for classifier, parameters in zip(classifiers, parameters_list):
    cv_model = GridSearchCV(classifier, parameters, cv=10)
    cv_model.fit(X_train, Y_train)
    best_params_list.append(cv_model.best_params_)
    best_scores_list.append(cv_model.best_score_)

# Find the index of the best performing classifier
best_index = best_scores_list.index(max(best_scores_list))

# Retrieve the best classifier and its best parameters
best_classifier = classifiers[best_index]
best_score = best_scores_list[best_index]
best_params = best_params_list[best_index]

# Fit the best classifier with the best parameters on the training data
best_classifier.set_params(**best_params)
best_classifier.fit(X_train, Y_train)


# Calculate and print the accuracy on the test data
test_accuracy = best_classifier.score(X_test, Y_test)
yhat = best_classifier.predict(X_test)

# Print the best performing classifier's results
print("Best Classifier:", best_classifier.__class__.__name__)
print("Tuned Hyperparameters (Best Parameters):", best_params)
print("Training Accuracy:", best_score)
print("Test Accuracy:", test_accuracy)
plot_confusion_matrix(Y_test,yhat)
```

- Loop to find the best prediction model accuracy and the parameters.
- https://github.com/Guipini/IBM/blob/main/Predictive%20Analysis%20(Classification)/IBM-DS0321EN-SkillsNetwork_labs_module_4_SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite(2).ipynb