IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Parul Verma
31-Jan-2025

Outline

Executive Summary

- Objective of this project is to analyze SpaceX Falcon 9 rocket launches and determine whether the landing of rocket's first stage will be successful or not.
- In this project, SpaceX data have been collected from various sources, cleansed, analyzed using visualization, SQL and interactive maps.
- As per EDA analysis, ES-L1, GEO, HEO and SSO orbits have high success rates while GTO have lowest success rate of launch.
- For heavy payloads, successful landing rate is more for Polar, LEO and ISS orbits.
- Launch site maps and plotly dashboards clearly state that there are high chances of successful launch if the selected launch site is KSC LC-39A.
- Further, created multiple ML models using different classification algorithms in order to predict the outcome of future launches.
- SVM, KNN and Logistic Regression models can be used to predict the outcome of future launches.

Introduction

- Objective of this project is to analyze SpaceX Falcon 9 rocket launches and determine whether the landing of rocket's first stage will be successful or not.
- The project involves collection, cleansing, exploratory data analysis, visualization and prediction of falcon 9 landing using various methodologies.

Section 1

Methodology

Methodology

Executive Summary

- Data collection:
  - SpaceX data collected from various API endpoints and web scraping. Further, data filtered to include only Falcon 9 launches.
- Data wrangling process:
  - After data collection, cleansing performed by replacing missing values in payload mass column with mean value and missing values in landing pad column kept as is.
  - New columns created for success and failure of landing based on landing outcome.
- Exploratory data analysis (EDA) using various visualization techniques and SQL
- Interactive visual analytics using Folium and Plotly Dash
- Predictive analysis using classification models
  - Built and evaluated models using logistic regression, decision tree, K-means and SVM.

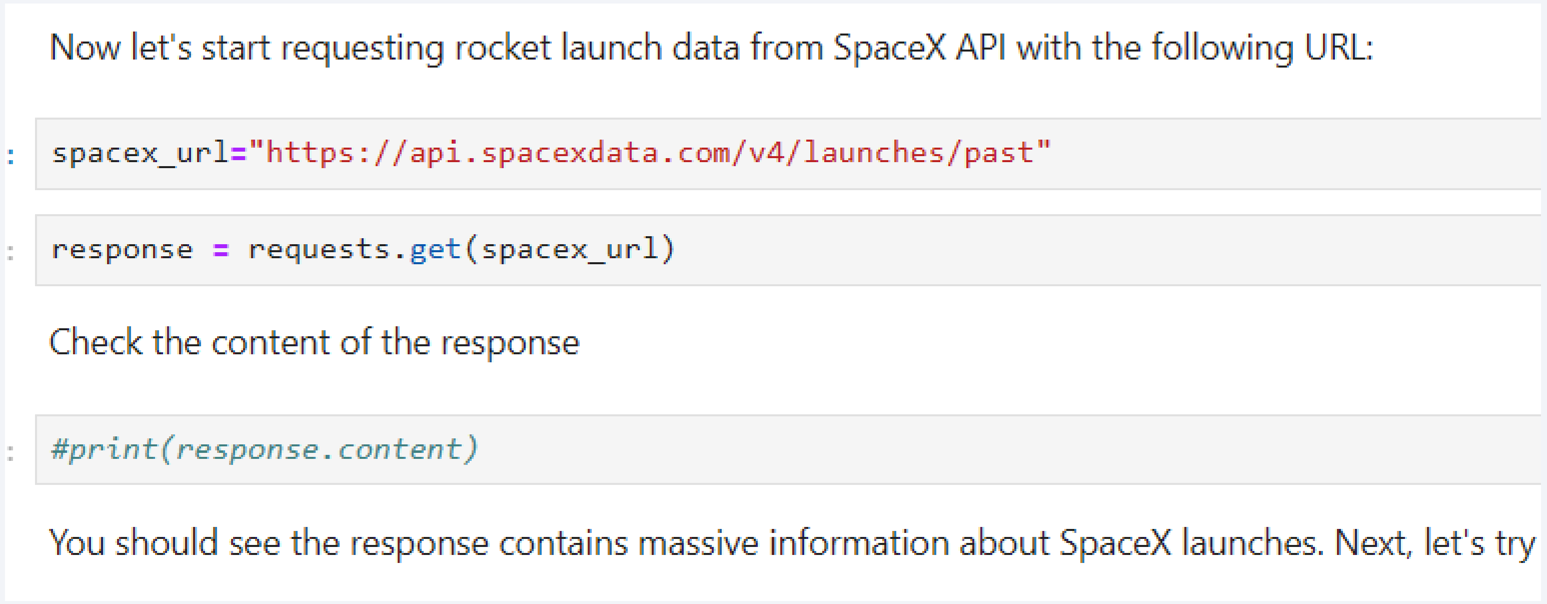
Data Collection

- Data collected using below API endpoints:
  - First SpaceX information is collected using below API endpoint into a pandas dataframe:

    https://api.spacexdata.com/v4/launches/past
  - Above data contained ids in rocket, payload, launchpad and cores columns which are further used to collect data for booster version, mass of the payload and orbit information, launch site name, and its longitude and latitude, outcome of landing, type of landing and various other information related to flights.
  - Booster Version information API - https://api.spacexdata.com/v4/rockets/<rocket-id>
  - Launch Site information API - https://api.spacexdata.com/v4/launchpads/<launchpad-id>
  - Payload and orbit information API - https://api.spacexdata.com/v4/payloads/<payload-id>
  - Flights and landing outcome information API - https://api.spacexdata.com/v4/cores/<core-id>

Data Collection – SpaceX API

- Below is the SpaceX API call screenshot.
- Reference: https://github.com/vermaparul85/IBM-Data-Science-Projects/blob/main/Capstone/spacex-data-collection-api.ipynb



Now let's start requesting rocket launch data from SpaceX API with the following URL:

```
spacex_url="https://api.spacexdata.com/v4/launches/past"
```

```
response = requests.get(spacex_url)
```

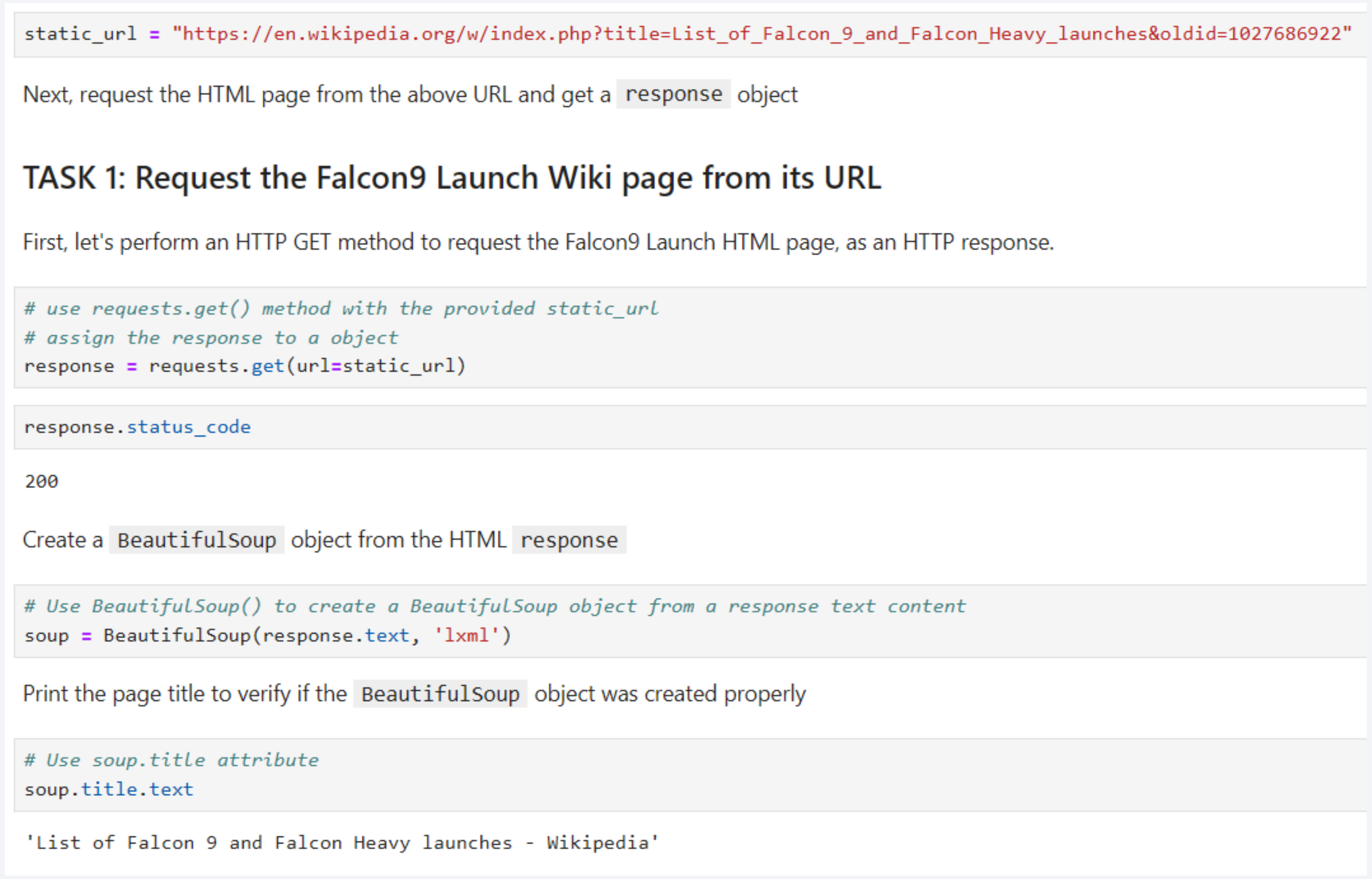
Check the content of the response

```
#print(response.content)
```

You should see the response contains massive information about SpaceX launches. Next, let's try

Data Collection - Scraping

- Attached SpaceX web scraping screenshot.
- Web scraping done using requests and BeautifulSoup python libraries.
- Reference: https://github.com/vermaparul85/IBM-Data-Science-Projects/blob/main/Capstone/spacex-webscraping.ipynb



```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"
```

Next, request the HTML page from the above URL and get a `response` object

TASK 1: Request the Falcon9 Launch Wiki page from its URL

First, let's perform an HTTP GET method to request the Falcon9 Launch HTML page, as an HTTP response.

```
# use requests.get() method with the provided static_url
# assign the response to a object
response = requests.get(url=static_url)
```

```
response.status_code
```

```
200
```

Create a `BeautifulSoup` object from the HTML `response`

```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(response.text, 'lxml')
```

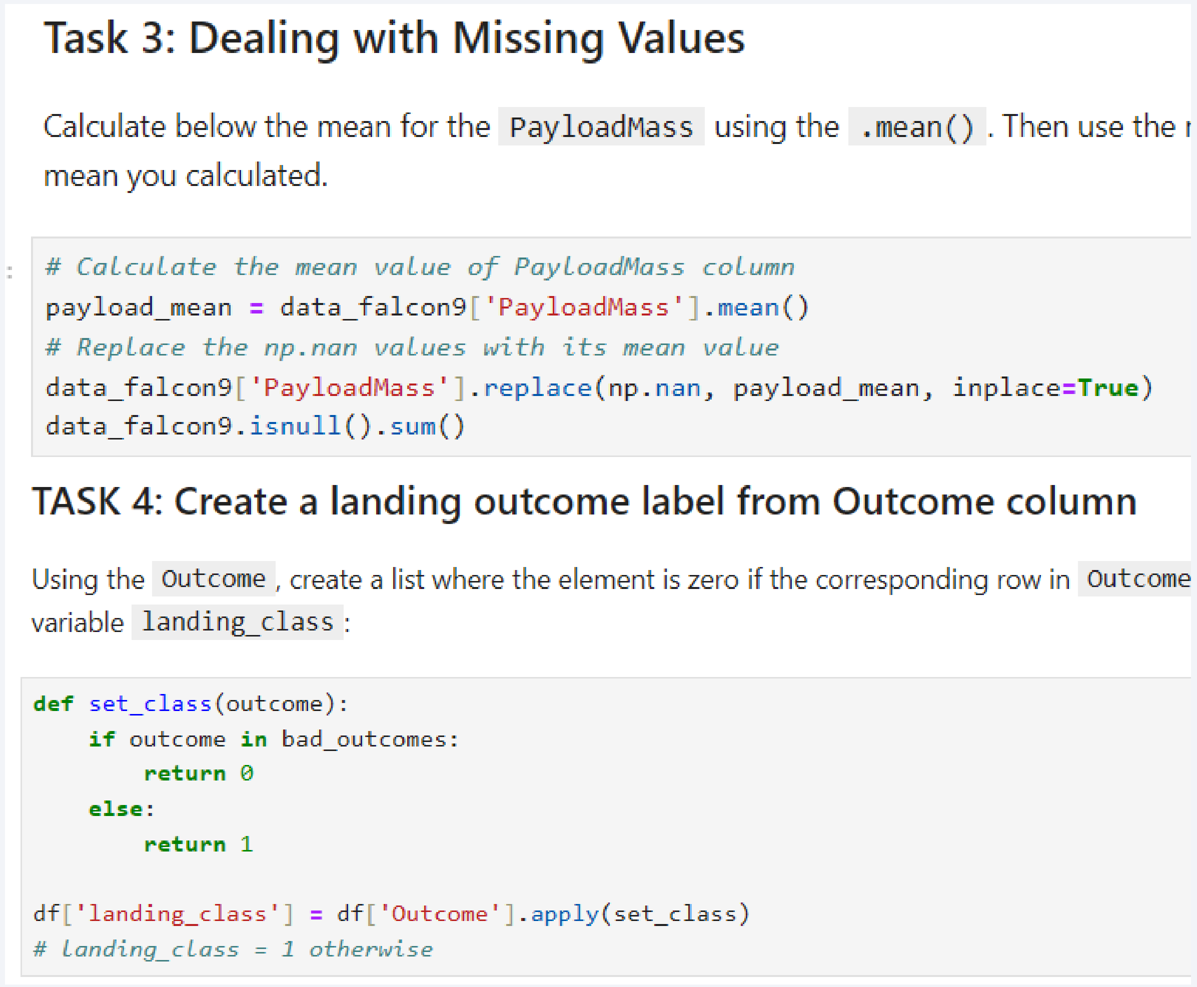
Print the page title to verify if the `BeautifulSoup` object was created properly

```
# Use soup.title attribute
soup.title.text
```

```
'List of Falcon 9 and Falcon Heavy launches - Wikipedia'
```

Data Wrangling

- After data collection, cleansing performed by replacing missing values in payload mass column with mean value and missing values in landing pad column kept as is.
- Reference: https://github.com/vermaparul85/IBM-Data-Science-Projects/blob/main/Capstone/spacex-data-collection-api.ipynb
- New columns created for success and failure of landing based on landing outcome.
- Reference: https://github.com/vermaparul85/IBM-Data-Science-Projects/blob/main/Capstone/spacex-Data-wrangling.ipynb



Task 3: Dealing with Missing Values

Calculate below the mean for the `PayloadMass` using the `.mean()`. Then use the
mean you calculated.

```
# Calculate the mean value of PayloadMass column
payload_mean = data_falcon9['PayloadMass'].mean()
# Replace the np.nan values with its mean value
data_falcon9['PayloadMass'].replace(np.nan, payload_mean, inplace=True)
data_falcon9.isnull().sum()
```

TASK 4: Create a landing outcome label from Outcome column

Using the `Outcome`, create a list where the element is zero if the corresponding row in `Outcome`
variable `landing_class`:

```
def set_class(outcome):
    if outcome in bad_outcomes:
        return 0
    else:
        return 1

df['landing_class'] = df['Outcome'].apply(set_class)
# landing_class = 1 otherwise
```

Created by Parul Verma

EDA with Data Visualization

- Visualized different types of plots for SpaceX falcon 9 launch data
  - Visualized relationship between Flight Number and Payload using scatter plot to determine outcome of launch.
  - Visualized relationship between Flight Number and Launch Site using scatter plot.
  - Visualized relationship between Payload and Launch Site using scatter plot.
  - Visualized relationship between success rate of each orbit type using bar chart.
  - Visualized relationship between Flight Number and Orbit type using scatter plot.
  - Visualized relationship between Payload and Orbit type using scatter plot.
  - Visualized the launch success yearly trend using line plot.
- Reference: https://github.com/vermaparul85/IBM-Data-Science-Projects/blob/main/Capstone/spacex-eda-dataviz.ipynb

EDA with SQL

- SpaceX data first loaded to sqlite database and then analyzed using SQL queries.
- Identified unique launch sites used for Falcon 9 launches.
- Analyzed payload mass for different customers and booster versions.
- Analyzed data for successful landing outcomes on land and by drone ship.
- Identified booster versions carried maximum payload mass.
- Fetched count of landing outcomes (success/fail) for a specific period.
- Reference: https://github.com/vermaparul85/IBM-Data-Science-Projects/blob/main/Capstone/spacex-eda-sql.ipynb

Build an Interactive Map with Folium

- Visualized Launch site longitude and latitude on world map using Folium library.
- Marked all sites using marker and circle map objects to pin each launch site location on map.
- Created group for similar coordinates using marker cluster to simplify the map.
- Marked each location on map based on launch success or failure using different colors.
- Calculated distance between launch sites to its proximities using map object PolyLine.
- Reference: https://github.com/vermaparul85/IBM-Data-Science-Projects/blob/main/Capstone/launch-site-location-map-interactive_dashboard.ipynb

Build a Dashboard with Plotly Dash

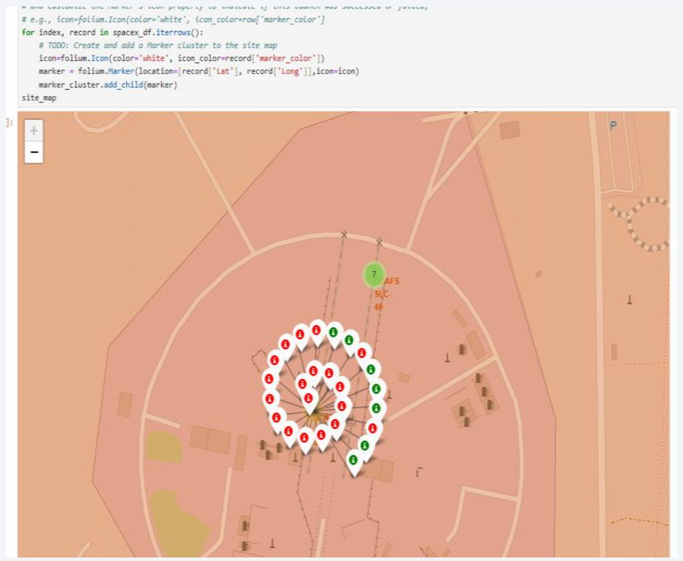
- Created pie chart to display launch success and failure outcome which can be visualized for all launch sites and for specific site as well using “Launch Site” dropdown.
- Created scatter plot to display relationship between payload mass and launch success/failure outcome which can be filtered using payload range slider.
- Reference: https://github.com/vermaparul85/IBM-Data-Science-Projects/blob/main/Capstone/spacex_dash_app.py

Predictive Analysis (Classification)

- First, created feature and target variables X and y.
- Standardized the feature variable using scikit-learn preprocessing library.
- Split the X and y dataset into training and test datasets.
- Built cross-validation models using scikit-learn model selection GridSerachCV.
- Found best Hyperparameter for SVM, Classification Trees and Logistic Regression and k-means Nearest Neighbours.
- Predicted yhat for test data and checked the accuracy score using score method and examined confusion matrix for each model.
- Reference: https://github.com/vermaparul85/IBM-Data-Science-Projects/blob/main/Capstone/SpaceX-Machine-Learning-Prediction.ipynb

Results

- Exploratory data analysis results:
  - As per scatter plots for launch site and payload mass, no rockets launched for heavy payload mass (greater than 10000) for VAFB-SLC site.
  - As per bar chart for orbit types, ES-L1, GEO, HEO and SSO have high success rates while GTO have lowest success rate of launch.
  - As per chart between payload and orbit type, successful landing rate is more for Polar, LEO and ISS with heavy payloads.
  - As per line chart, success rate increased from 2013 till 2017.
- Picture shows success/failure of landing for a launch site using interactive map.
- Predictive analysis results: Logistic regression, SVM and KNN performed best with 83.33% accuracy on test data.



```
# and customize the marker's icon property to indicate if this launch was successed or failed,
# e.g., icon=folium.Icon(color='white', icon_color=row['marker_color']
for index, record in spacex_df.iterrows():
    # TODO: Create and add a Marker cluster to the site map
    icon=folium.Icon(color='white', icon_color=record['marker_color'])
    marker = folium.Marker(location=[record['Lat'], record['Long']],icon=icon)
    marker_cluster.add_child(marker)
site_map
```

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

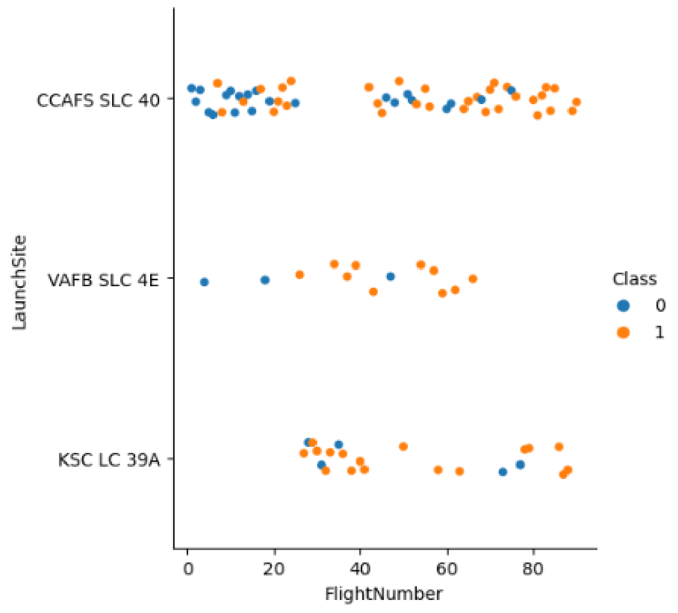
TASK 1: Visualize the relationship between Flight Number and Launch Site

Use the function `catplot` to plot `FlightNumber` vs `LaunchSite`, set the parameter `x` parameter to `FlightNumber`,set the `y` to `Launch Site` and set the parameter `hue` to `'class'`

```
# Plot a scatter point chart with x axis to be Flight Number and y axis to be the launch site, and hue to be the class value
sns.catplot(data=df, x='FlightNumber', y='LaunchSite', hue='Class')
```

```
C:\Users\verma\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a future
version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
C:\Users\verma\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a future
version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

```
<seaborn.axisgrid.FacetGrid at 0x1ecb244ec10>
```



Now try to explain the patterns you found in the Flight Number vs. Launch Site scatter point plots.

KSC LC 39A launch site have high success rate as per the scatter plot. CCAFS SLC 40 site had high rate of failture for initial flights.

Created by Parul Verma

Payload vs. Launch Site

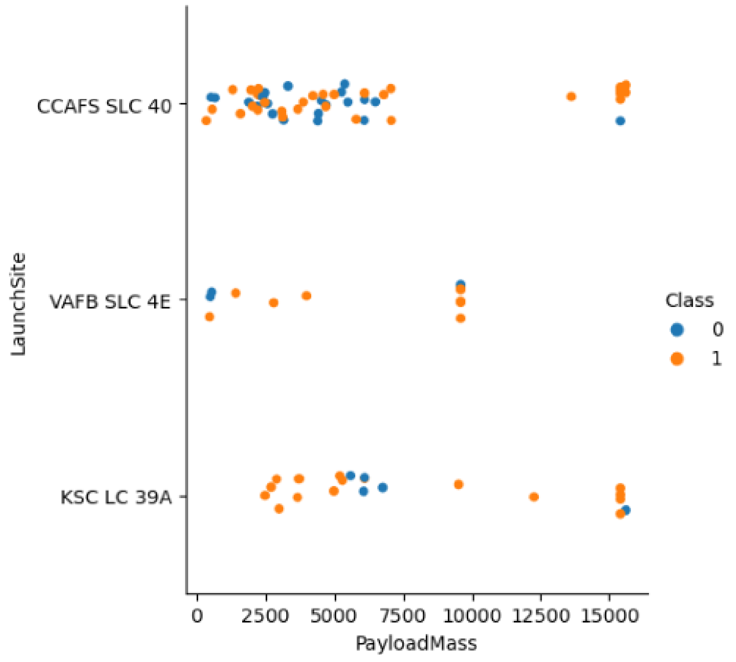
TASK 2: Visualize the relationship between Payload and Launch Site

We also want to observe if there is any relationship between launch sites and their payload mass.

```
# Plot a scatter point chart with x axis to be Pay Load Mass (kg) and y axis to be the launch site, and hue to be the class value
sns.catplot(data=df, x='PayloadMass', y='LaunchSite', hue='Class')
```

```
C:\Users\verma\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a future v
ersion. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
C:\Users\verma\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a future v
ersion. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

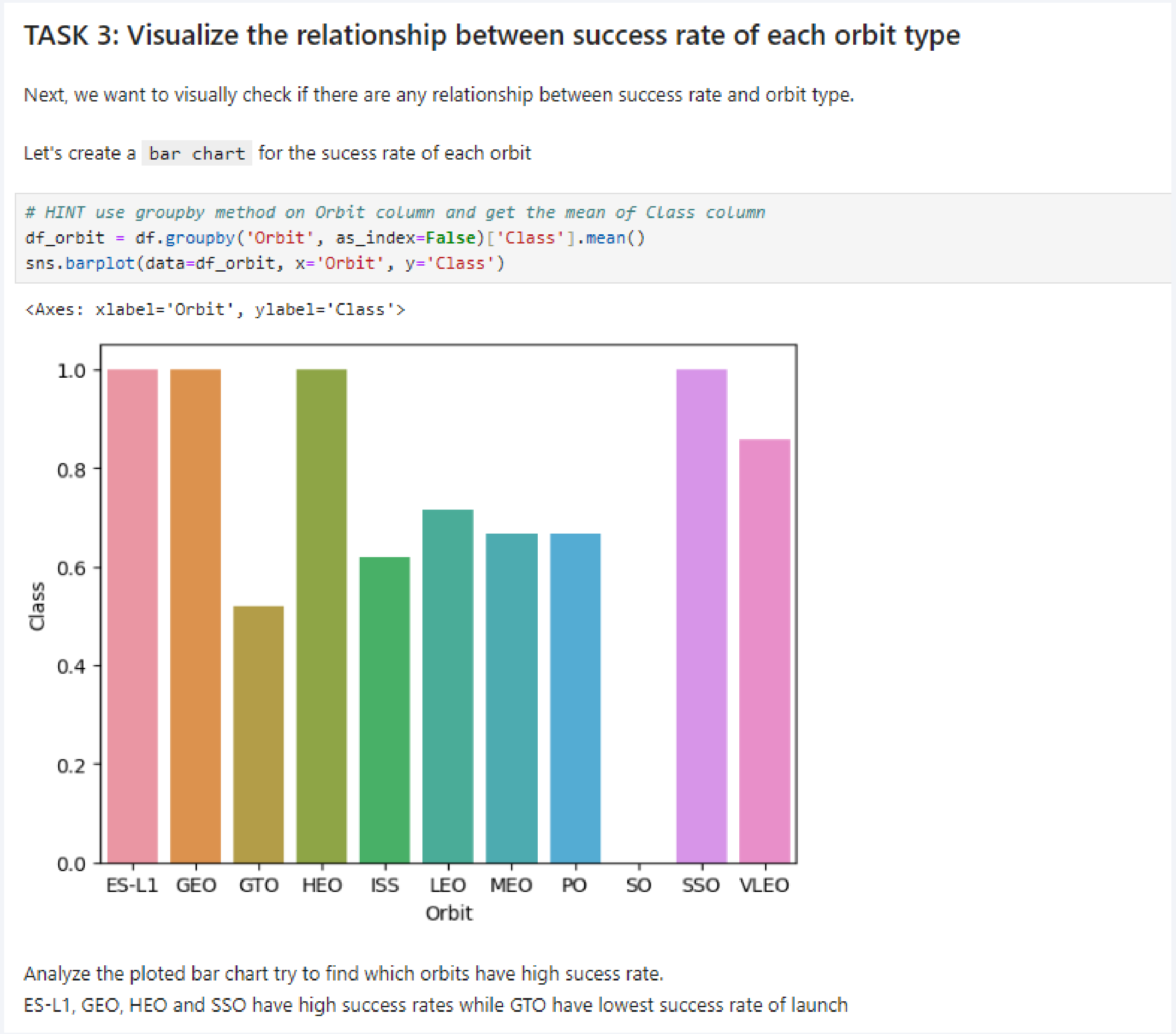
```
<seaborn.axisgrid.FacetGrid at 0x1ecb2697fd0>
```



Payload Vs. Launch Site scatter point chart shows that for VAFB-SLC launch site there are no rockets launched for heavy payload mass (greater than 10000). Also, CCAFS SLC site have more flights with heavy payload.

Created by Parul Verma

Success Rate vs. Orbit Type



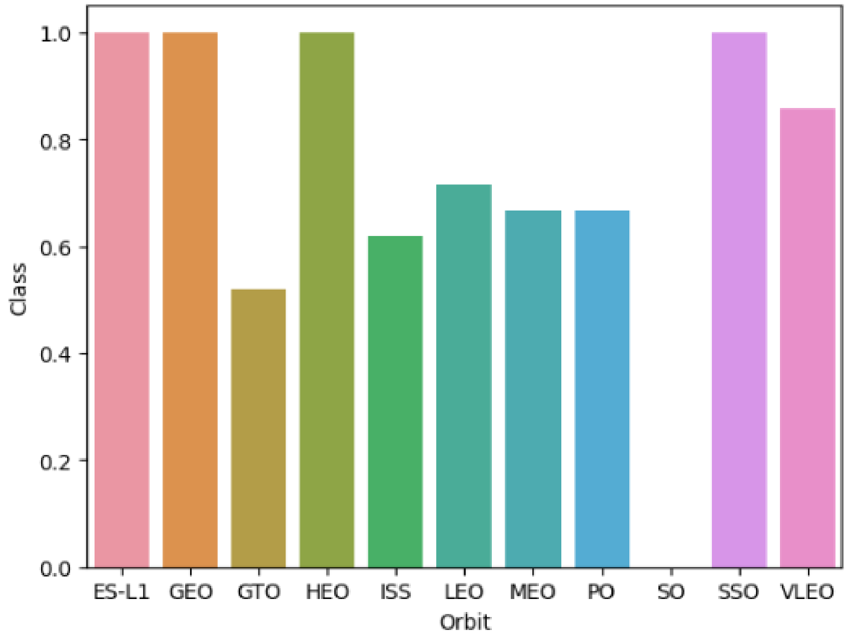
TASK 3: Visualize the relationship between success rate of each orbit type

Next, we want to visually check if there are any relationship between success rate and orbit type.

Let's create a `bar chart` for the sucess rate of each orbit

```
# HINT use groupby method on Orbit column and get the mean of Class column
df_orbit = df.groupby('Orbit', as_index=False)['Class'].mean()
sns.barplot(data=df_orbit, x='Orbit', y='Class')
```

```
<Axes: xlabel='Orbit', ylabel='Class'>
```



Analyze the ploted bar chart try to find which orbits have high sucess rate.
ES-L1, GEO, HEO and SSO have high success rates while GTO have lowest success rate of launch

Created by Parul Verma

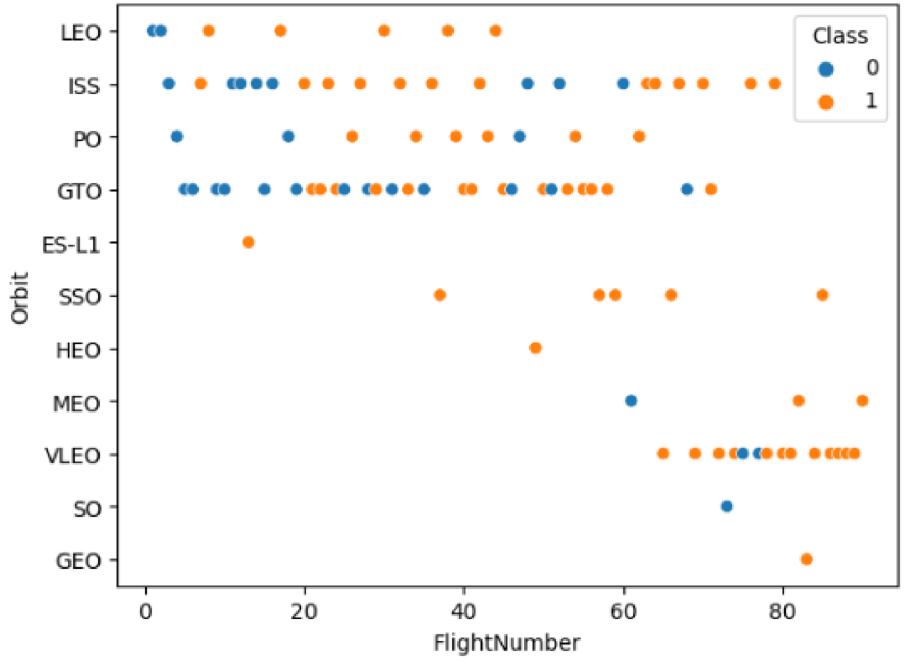
Flight Number vs. Orbit Type

TASK 4: Visualize the relationship between FlightNumber and Orbit type

For each orbit, we want to see if there is any relationship between FlightNumber and Orbit type.

```
# Plot a scatter point chart with x axis to be FlightNumber and y axis to be the Orbit, and hue to be the class value
sns.scatterplot(data=df, x='FlightNumber', y='Orbit', hue='Class')
```

```
<Axes: xlabel='FlightNumber', ylabel='Orbit'>
```



As per above chart, 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.

Created by Parul Verma

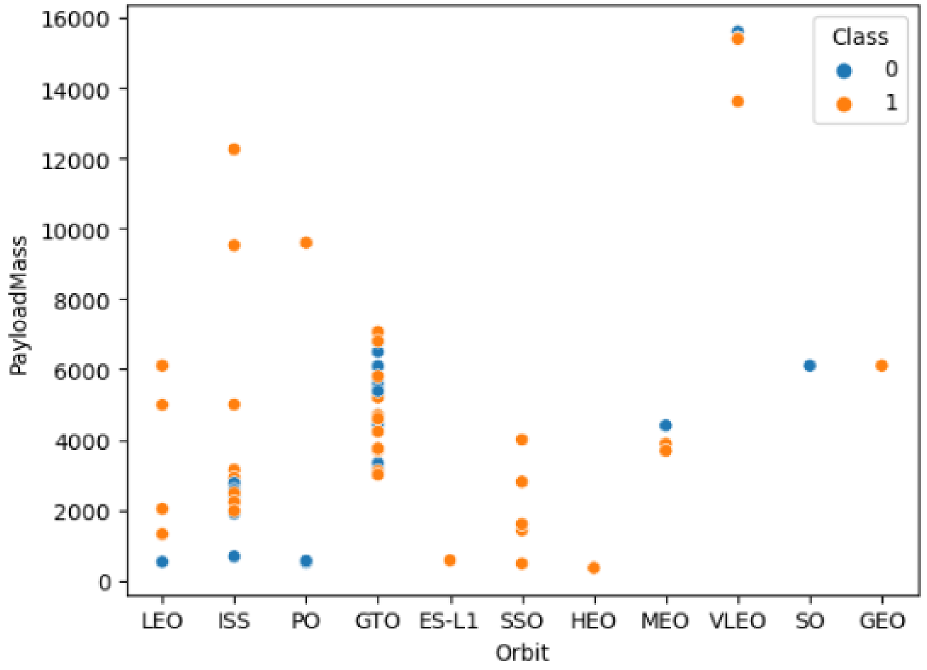
Payload vs. Orbit Type

TASK 5: Visualize the relationship between Payload and Orbit type

Similarly, we can plot the Payload vs. Orbit scatter point charts to reveal the relationship between Payload and Orbit type

```
# Plot a scatter point chart with x axis to be Payload and y axis to be the Orbit, and hue to be the class value
sns.scatterplot(data=df, y='PayloadMass', x='Orbit', hue='Class')
```

```
<Axes: xlabel='Orbit', ylabel='PayloadMass'>
```

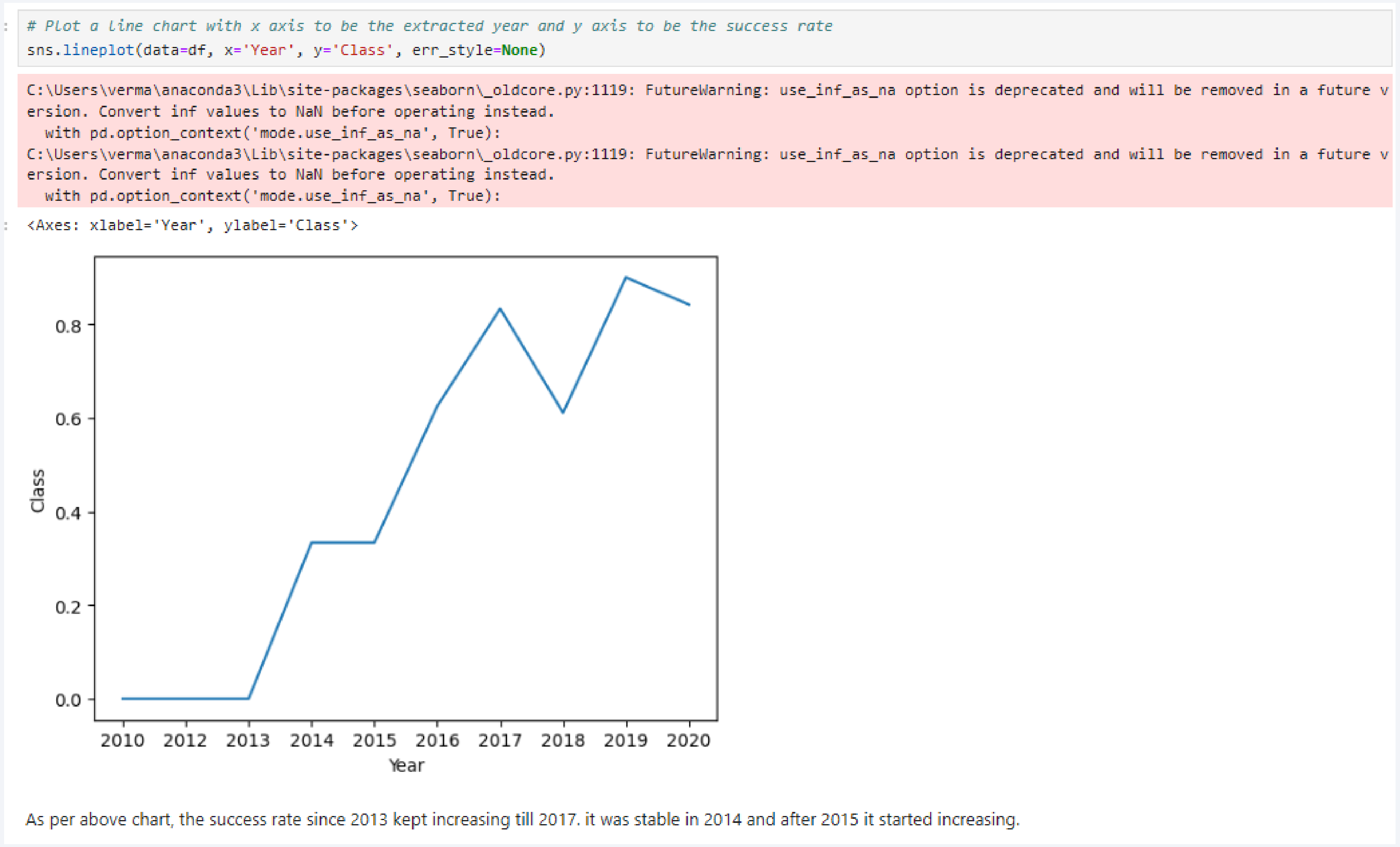


As per chart between payload and orbit type, successful landing rate is more for Polar, LEO and ISS with heavy payloads.

However for GTO, cannot distinguish this well as both positive landing rate and negative landing are there.

Created by Parul Verma

Launch Success Yearly Trend



```
# Plot a line chart with x axis to be the extracted year and y axis to be the success rate
sns.lineplot(data=df, x='Year', y='Class', err_style=None)
```

```
C:\Users\verma\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a future v
ersion. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
C:\Users\verma\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a future v
ersion. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

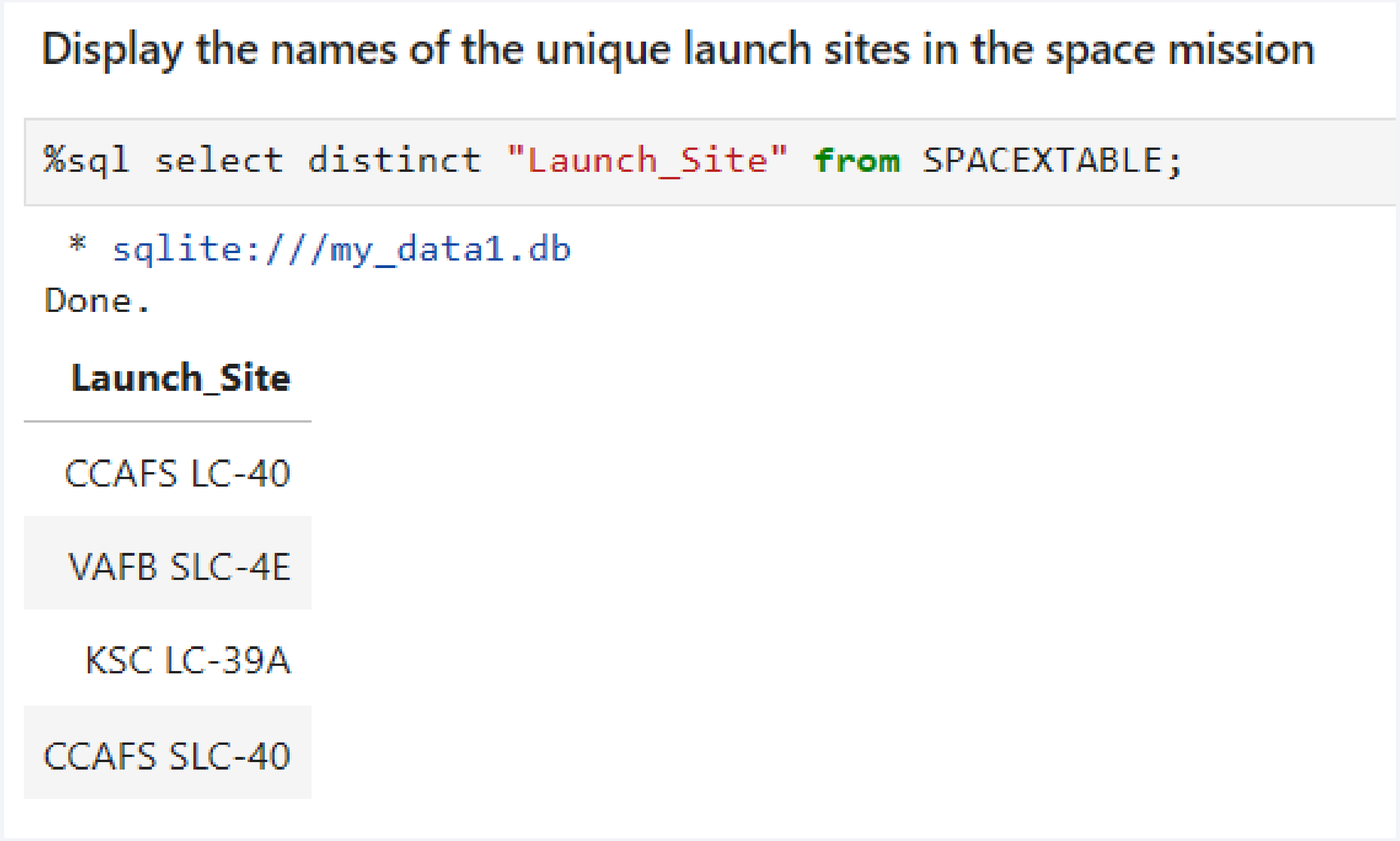
```
<Axes: xlabel='Year', ylabel='Class'>
```

As per above chart, the success rate since 2013 kept increasing till 2017. it was stable in 2014 and after 2015 it started increasing.

Created by Parul Verma

All Launch Site Names

There are 4 unique launch sites for Falcon 9 rockets.



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

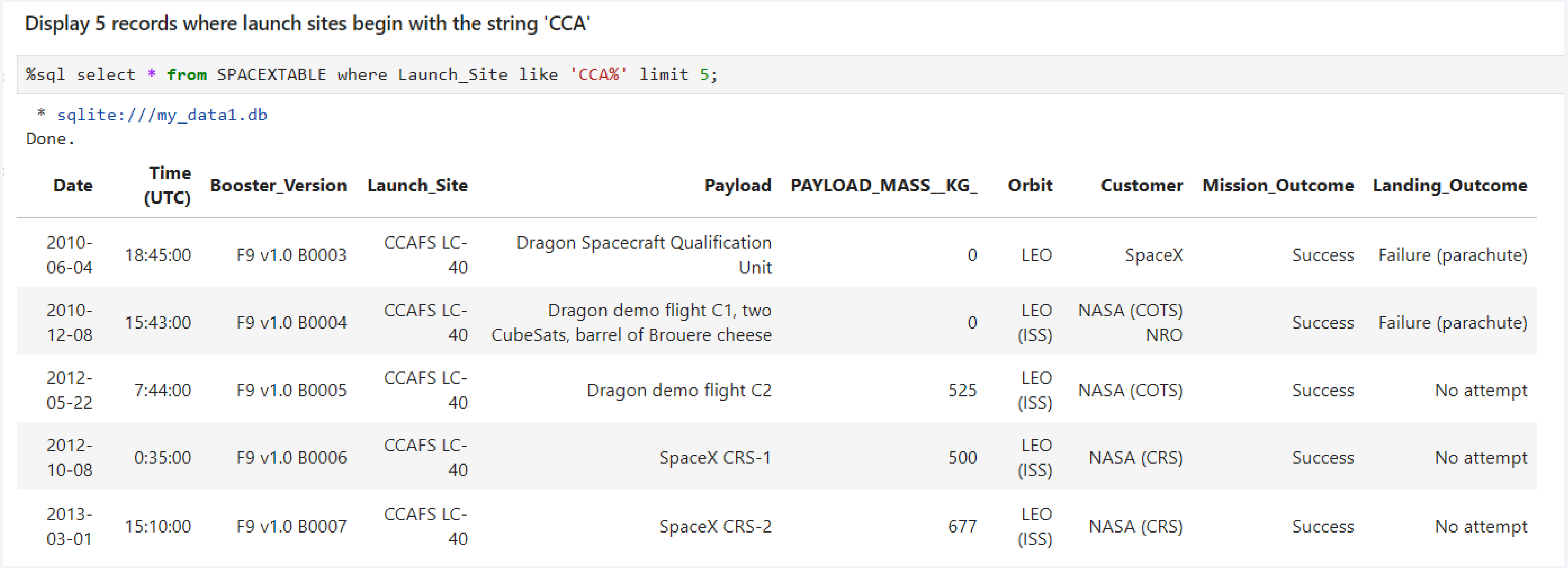
```
%sql select distinct "Launch_Site" from SPACEXTABLE;
```

```
 * 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'

5 records where launch sites begin with `CCA`



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

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

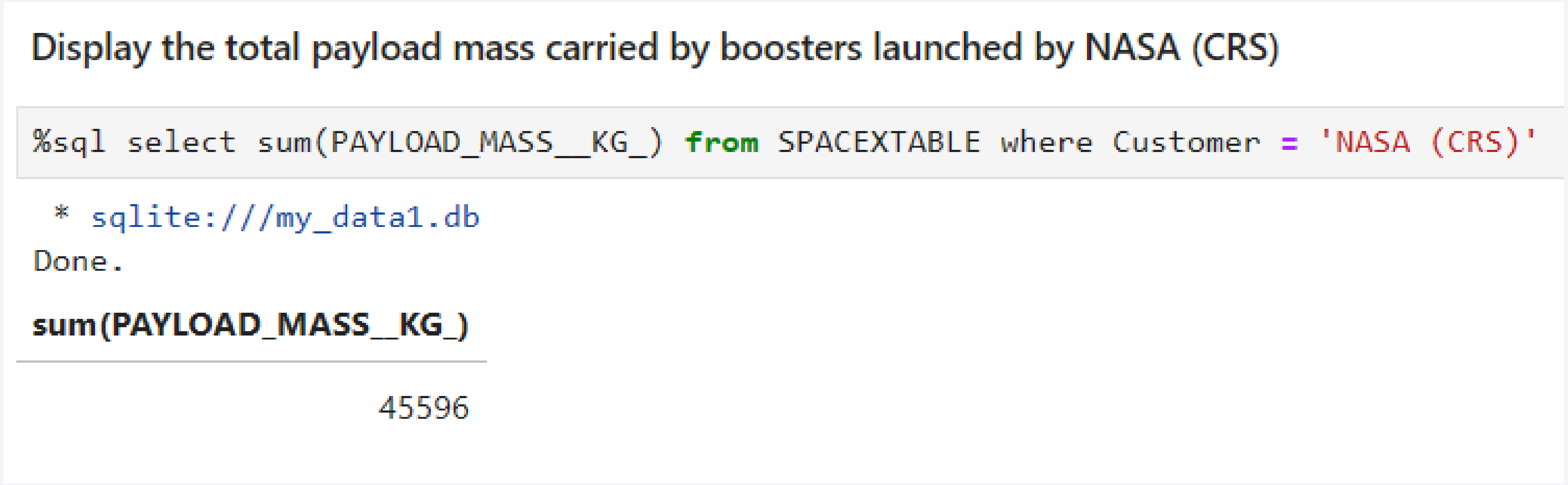
```
 * sqlite:///my_data1.db
Done.
```

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
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 (parachute)
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Created by Parul Verma

Total Payload Mass

Total payload mass carried by boosters from NASA is 45596 kg.



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

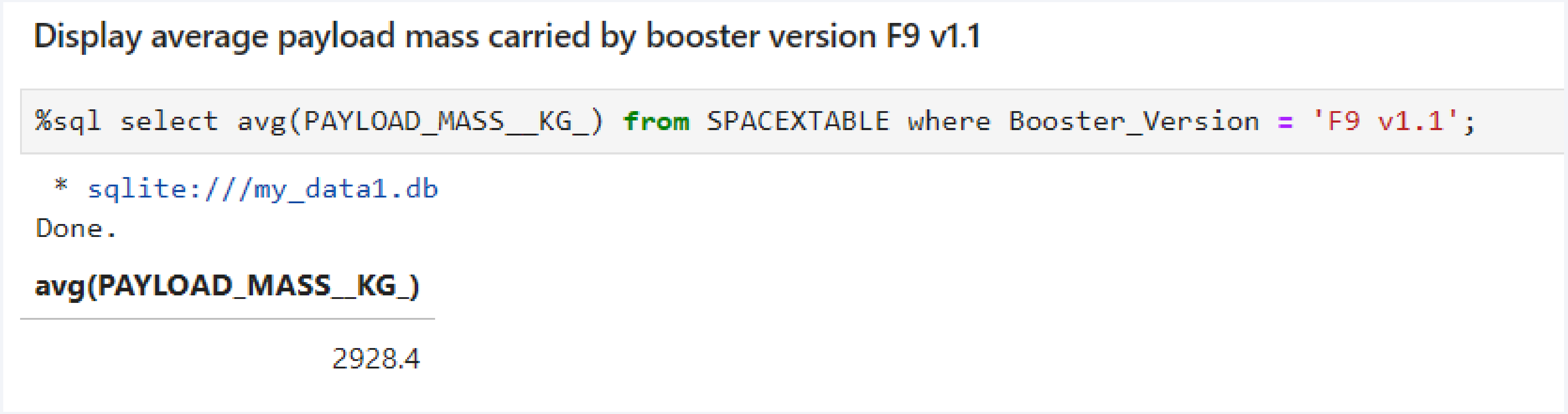
```
%sql select sum(PAYLOAD_MASS__KG_) from SPACEXTABLE where Customer = 'NASA (CRS)'
```

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

sum(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

Average payload mass carried by booster version F9 v1.1 was 2928.4 kg.



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

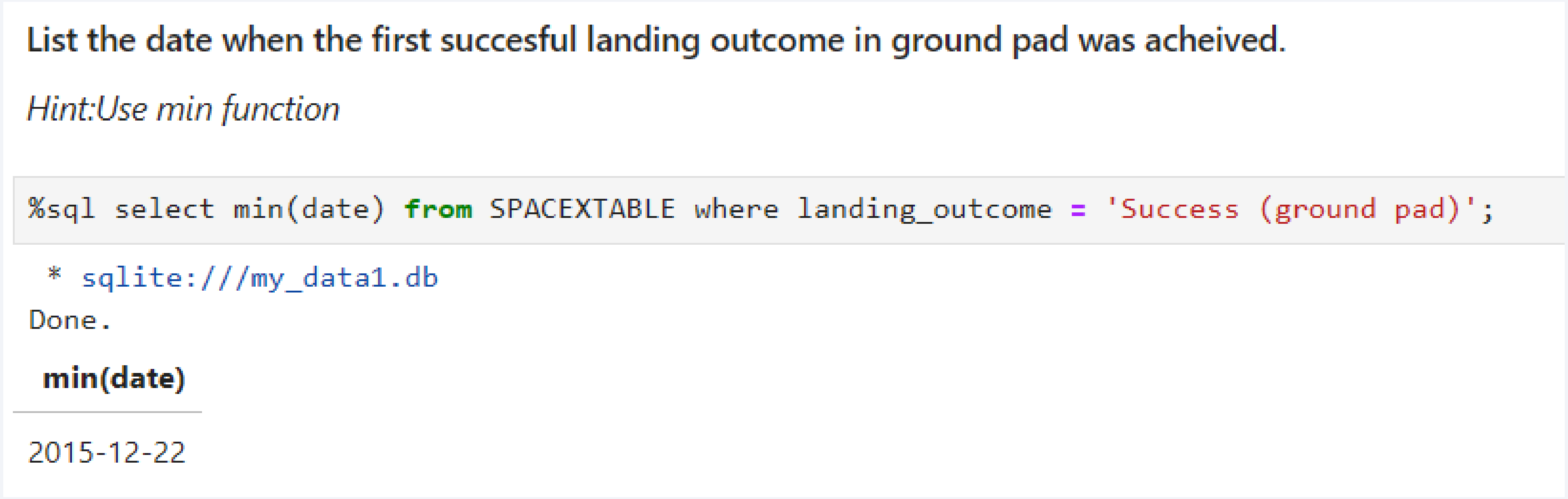
```
%sql select avg(PAYLOAD_MASS__KG_) from SPACEXTABLE where Booster_Version = 'F9 v1.1';
```

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

avg(PAYLOAD_MASS__KG_)
2928.4

First Successful Ground Landing Date

First successful landing on ground pad happened in 2015 on 22-Dec.



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

Hint:Use min function

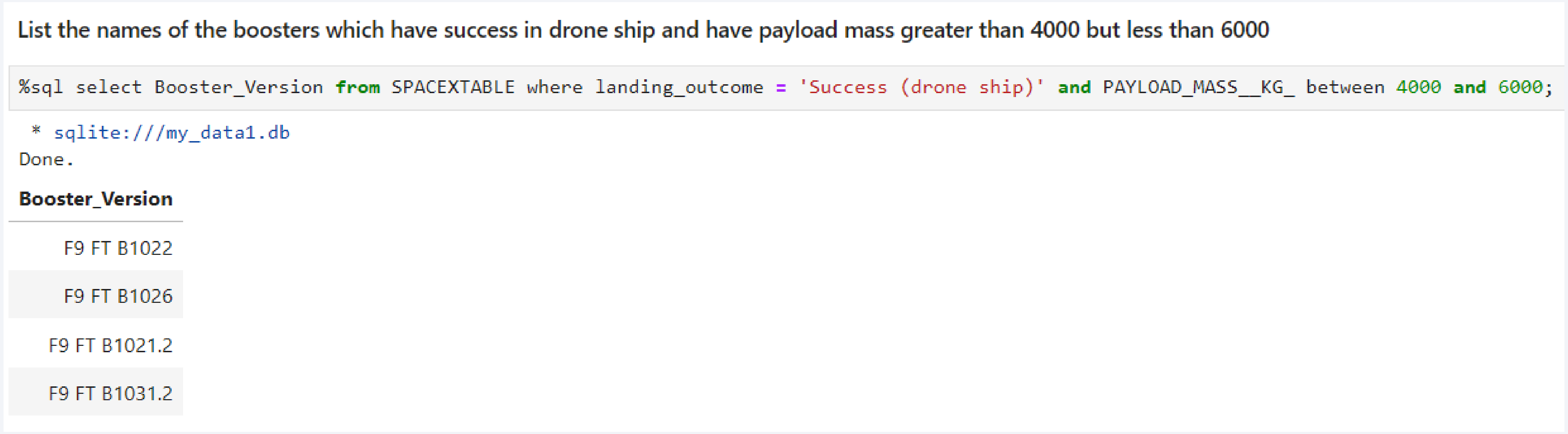
```
%sql select min(date) from SPACEXTABLE where landing_outcome = 'Success (ground pad)';
```

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

min(date)
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

There are 4 boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000.



List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

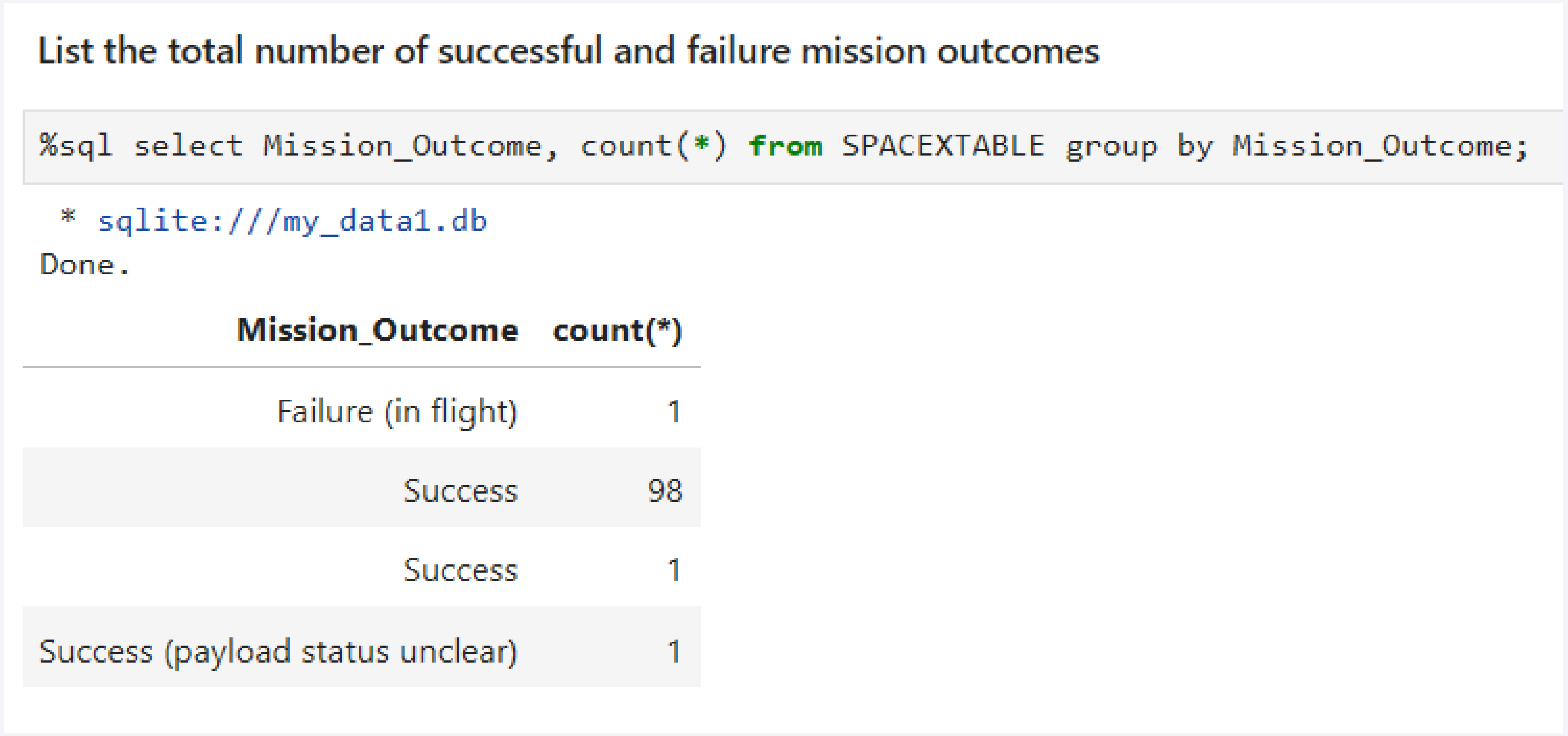
```
%sql select Booster_Version from SPACEXTABLE where landing_outcome = 'Success (drone ship)' and PAYLOAD_MASS__KG_ between 4000 and 6000;
```

```
 * 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

The total number of successful mission = 99 and failure mission = 1.



List the total number of successful and failure mission outcomes

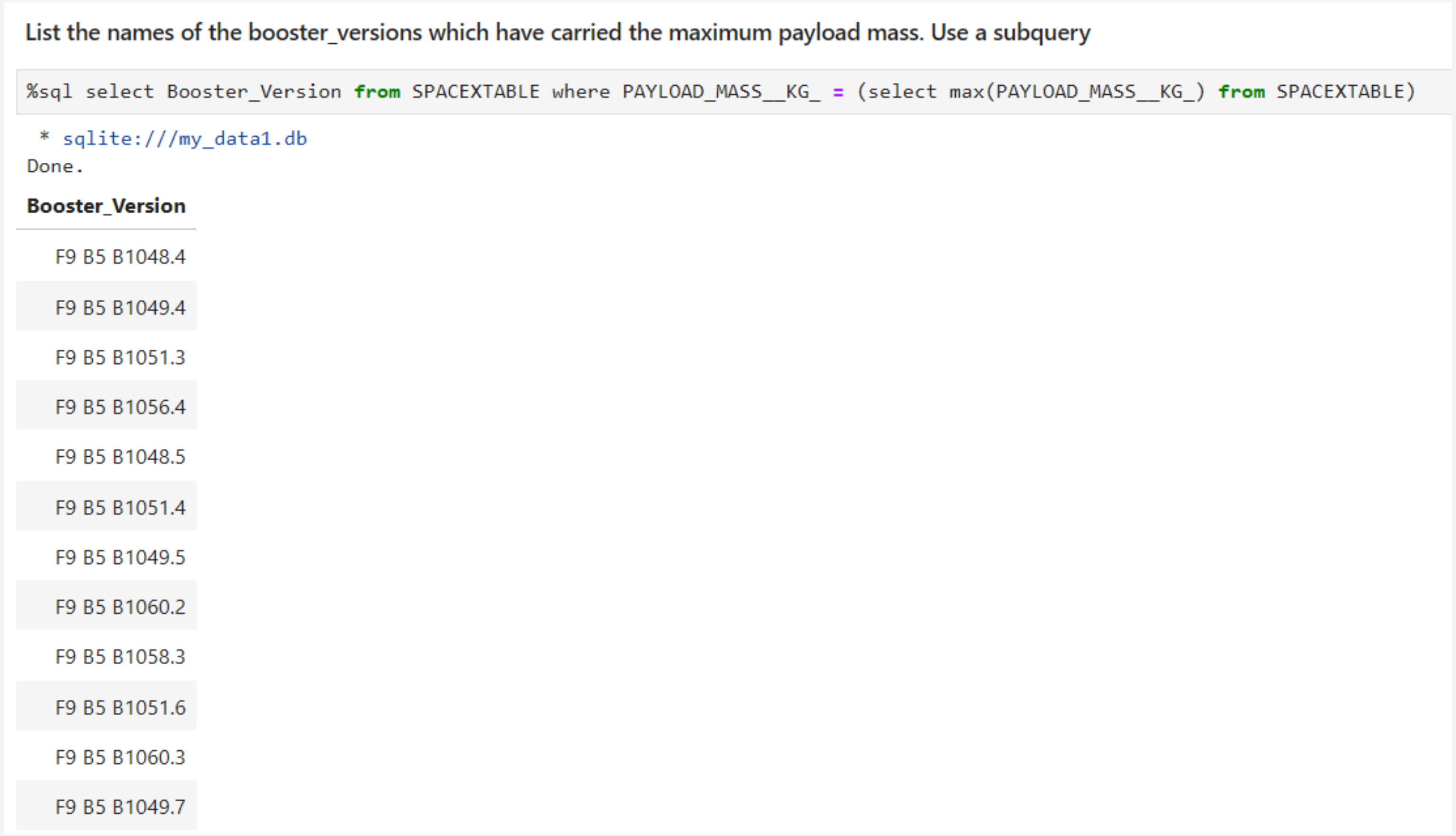
```
%sql select Mission_Outcome, count(*) from SPACEXTABLE group by Mission_Outcome;
```

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

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

Boosters Carried Maximum Payload

There are 12 booster which have carried the maximum payload mass.



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

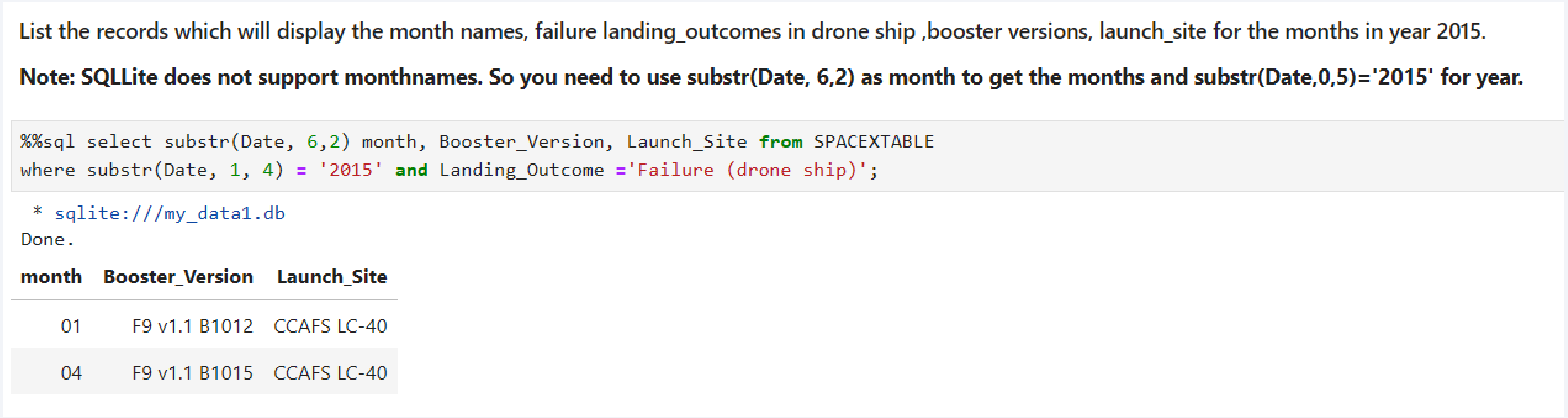
```
%sql select Booster_Version from SPACEXTABLE where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEXTABLE)
```

```
 * 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

There are only 2 failed landing outcomes in drone ship in year 2015.



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.

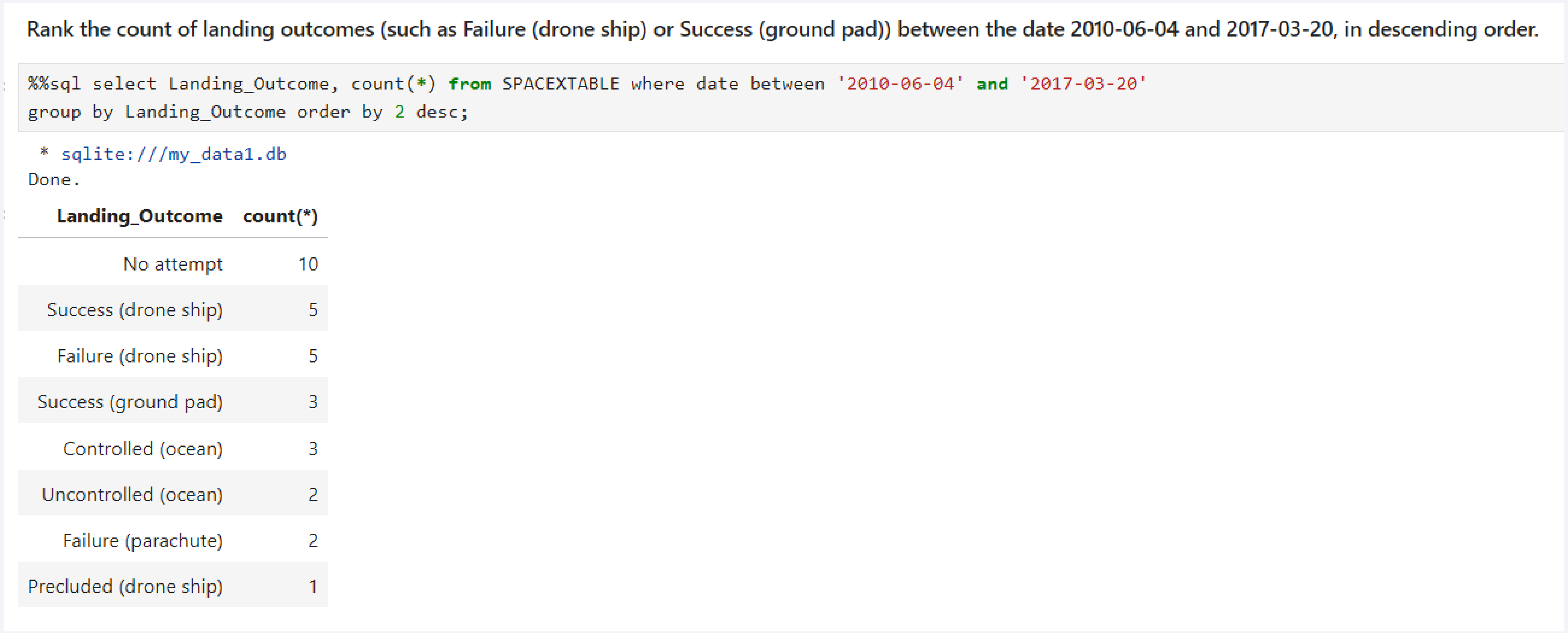
```
%%sql select substr(Date, 6,2) month, Booster_Version, Launch_Site from SPACEXTABLE
where substr(Date, 1, 4) = '2015' and Landing_Outcome ='Failure (drone ship)';
```

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

month	Booster_Version	Launch_Site
01	F9 v1.1 B1012	CCAFS LC-40
04	F9 v1.1 B1015	CCAFS LC-40

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

Below screenshot shows count for each landing outcome between the date 2010-06-04 and 2017-03-20 in descending order.



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.

```
%%sql select Landing_Outcome, count(*) from SPACEXTABLE where date between '2010-06-04' and '2017-03-20'
group by Landing_Outcome order by 2 desc;
```

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

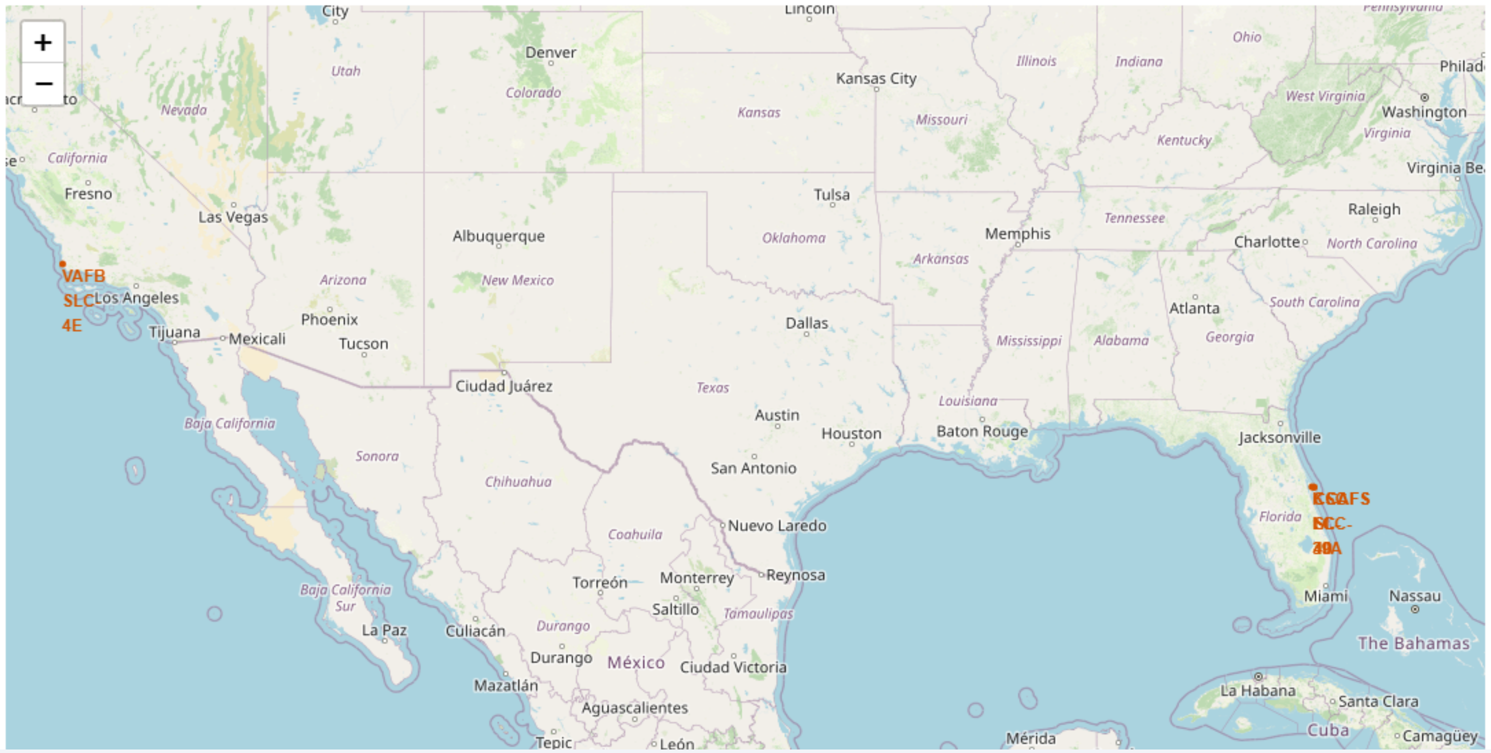
Landing_Outcome	count(*)
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

Section 3

Launch Sites Proximities Analysis

All Launch site locations on map

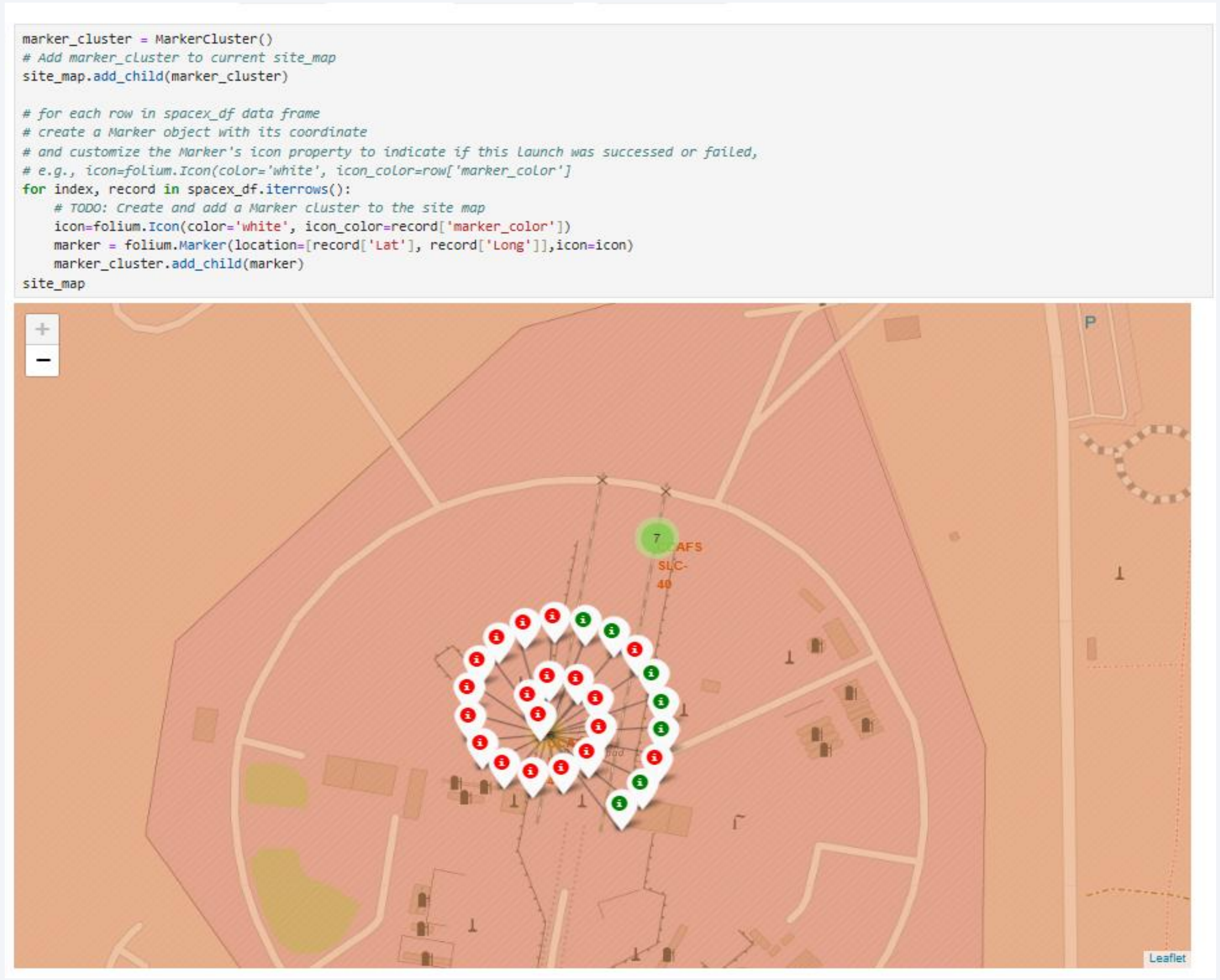
```
# Initial the map
site_map = folium.Map(location=nasa_coordinate, zoom_start=5)
# For each launch site, add a Circle object based on its coordinate (Lat, Long) values. In addition, add Launch site name as a popup label
for i, site in launch_sites_df.iterrows():
    circle = folium.Circle(location=[site['Lat'], site['Long']], radius=1000, color='#d35400', fill=True).add_child(folium.Popup(site['Launch Site']))
    marker = folium.map.Marker(location=[site['Lat'], site['Long']],
    # Create an icon as a text label
    icon=DivIcon(icon_size=(20,20),icon_anchor=(0,0),html='<div style="font-size: 12; color:#d35400;"><b>%s</b></div>' % site['Launch Site'],))
    site_map.add_child(circle)
    site_map.add_child(marker)
site_map
```



Visualize launch site locations on map

As per the locations on map, all launch sites are near coastlines/ocean.

Success/Failed Launches for each site



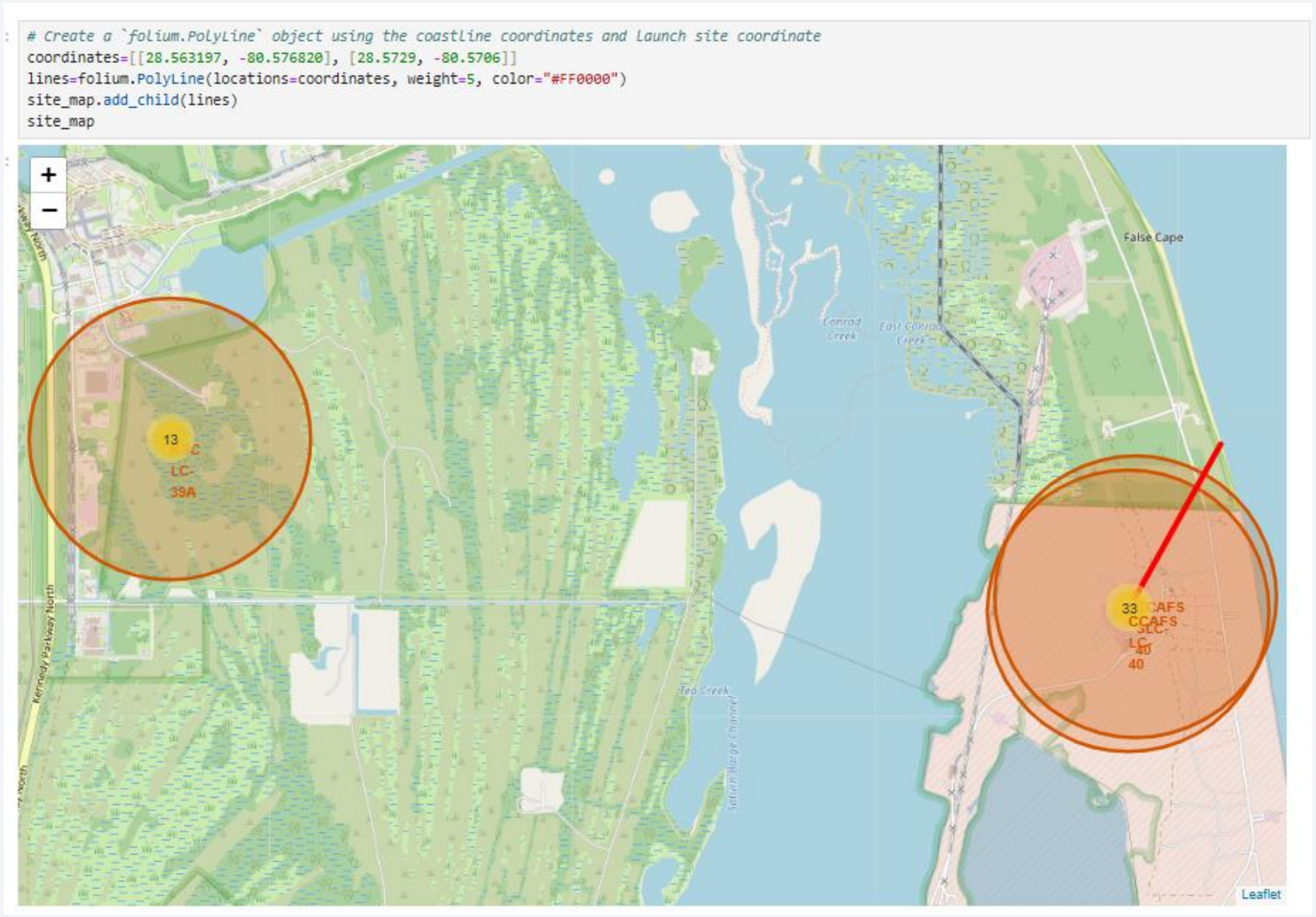
```
marker_cluster = MarkerCluster()
# Add marker_cluster to current site_map
site_map.add_child(marker_cluster)

# for each row in spacex_df data frame
# create a Marker object with its coordinate
# and customize the Marker's icon property to indicate if this launch was successed or failed,
# e.g., icon=folium.Icon(color='white', icon_color=row['marker_color']
for index, record in spacex_df.iterrows():
    # TODO: Create and add a Marker cluster to the site map
    icon=folium.Icon(color='white', icon_color=record['marker_color'])
    marker = folium.Marker(location=[record['Lat'], record['Long']],icon=icon)
    marker_cluster.add_child(marker)
site_map
```

Each launch site shows successful launches with green and failed launches with red color.

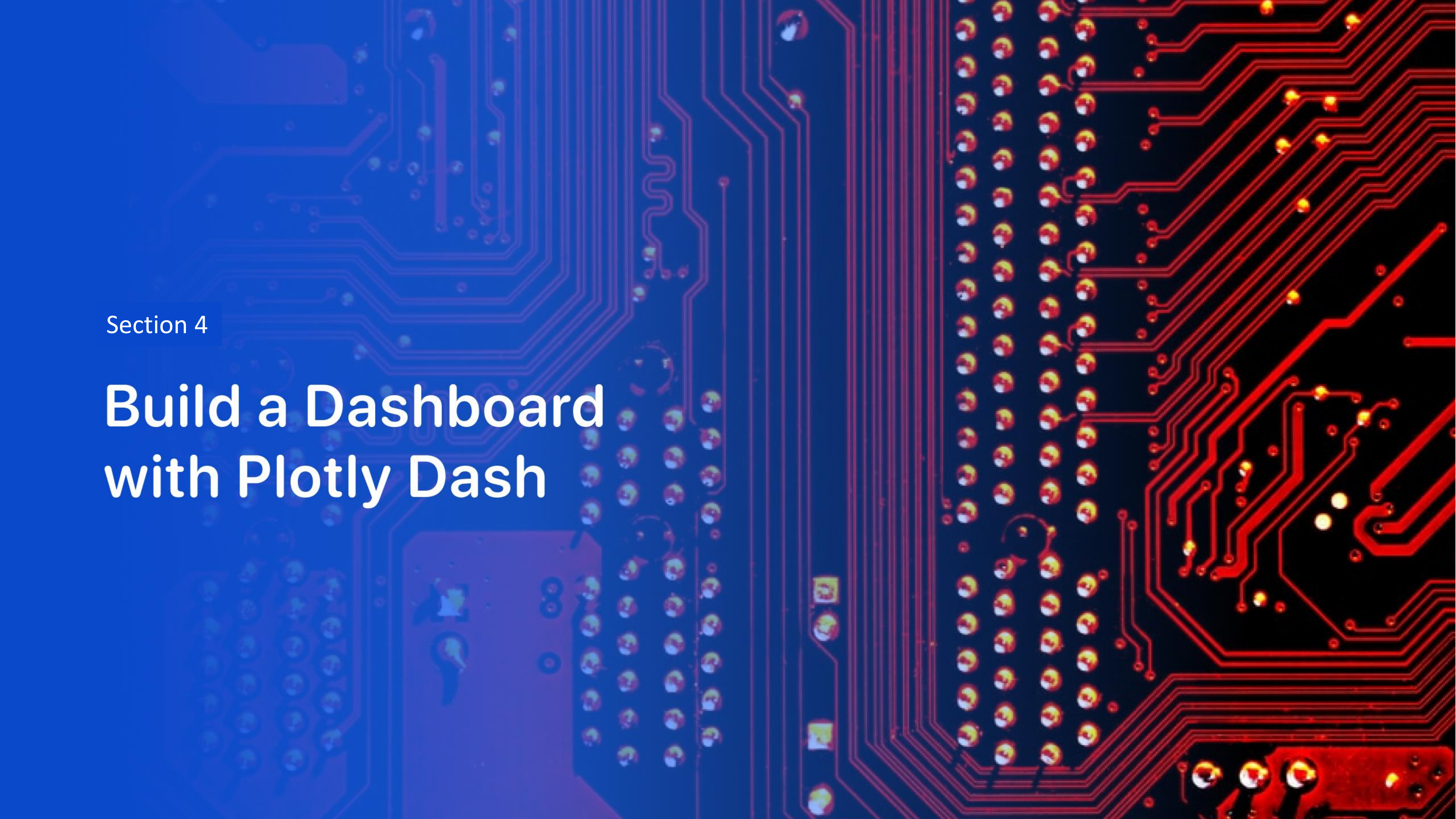
Used marker clusters to simplify the map as many markers were having same coordinates.

Mark distance between launch site and its proximities



```
# Create a `folium.PolyLine` object using the coastline coordinates and launch site coordinate
coordinates=[[28.563197, -80.576820], [28.5729, -80.5706]]
lines=folium.PolyLine(locations=coordinates, weight=5, color="#FF0000")
site_map.add_child(lines)
site_map
```

Displaying distance between launch site and coastline coordinates with thick red line using PolyLine method.

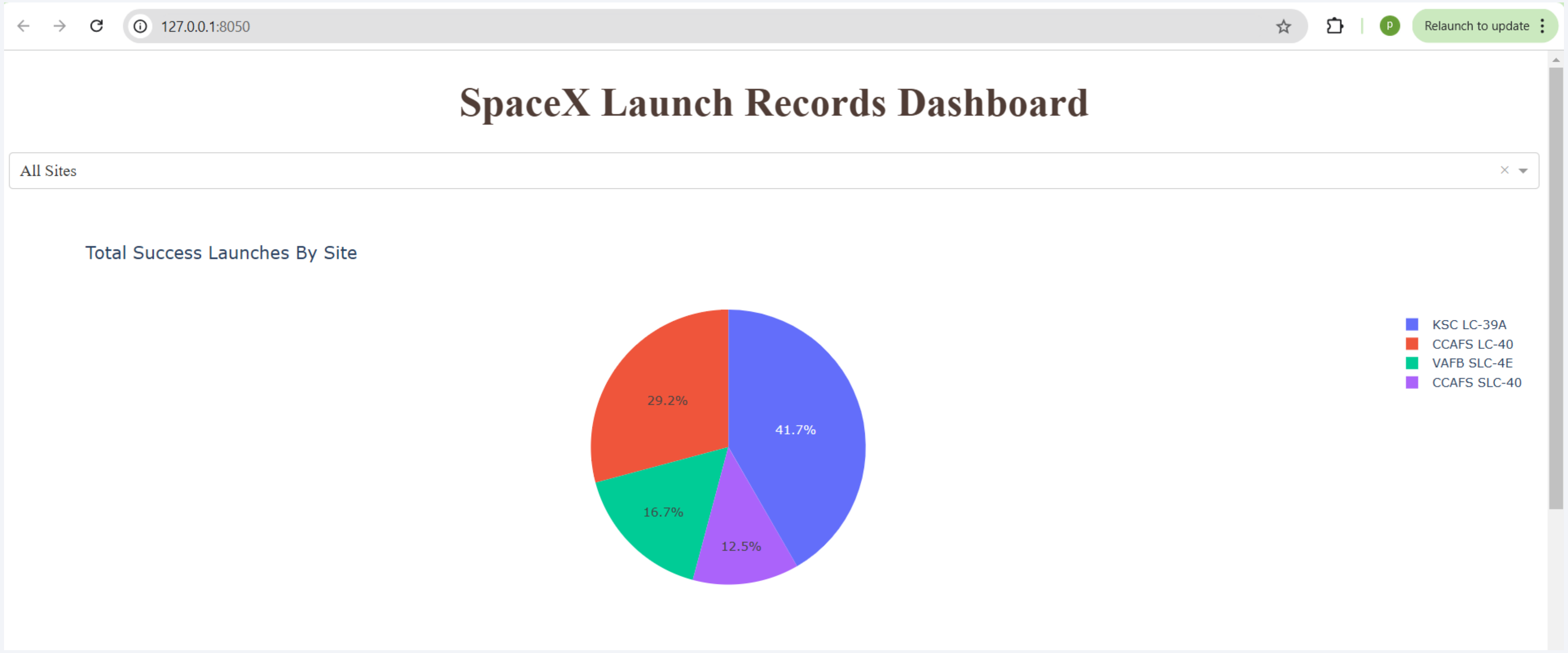


Section 4

Build a Dashboard with Plotly Dash

Total success launches by site

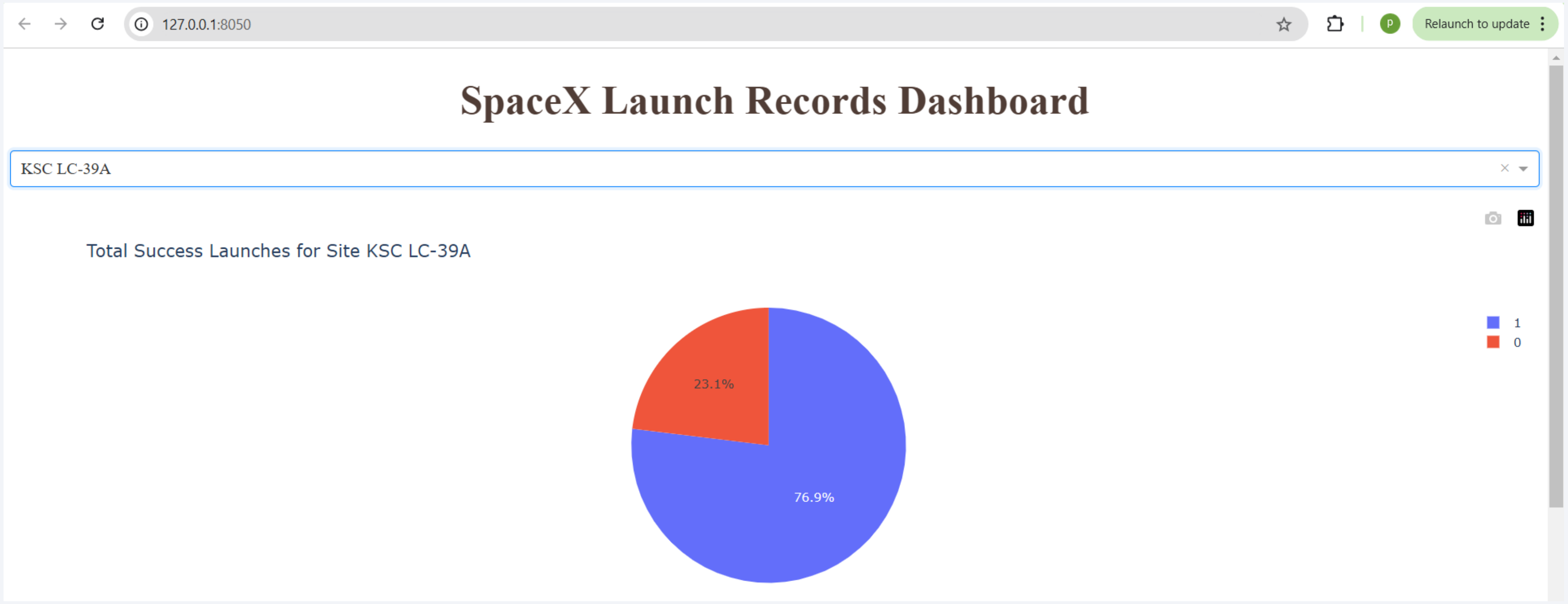
- KSC LC-39A Launch site have the highest success rate followed by CCAFS SLC-4E.
- CCAFS SLC-40 have the lowest success rate.



Created by Parul Verma

Total success launches for site KSC LC-39A

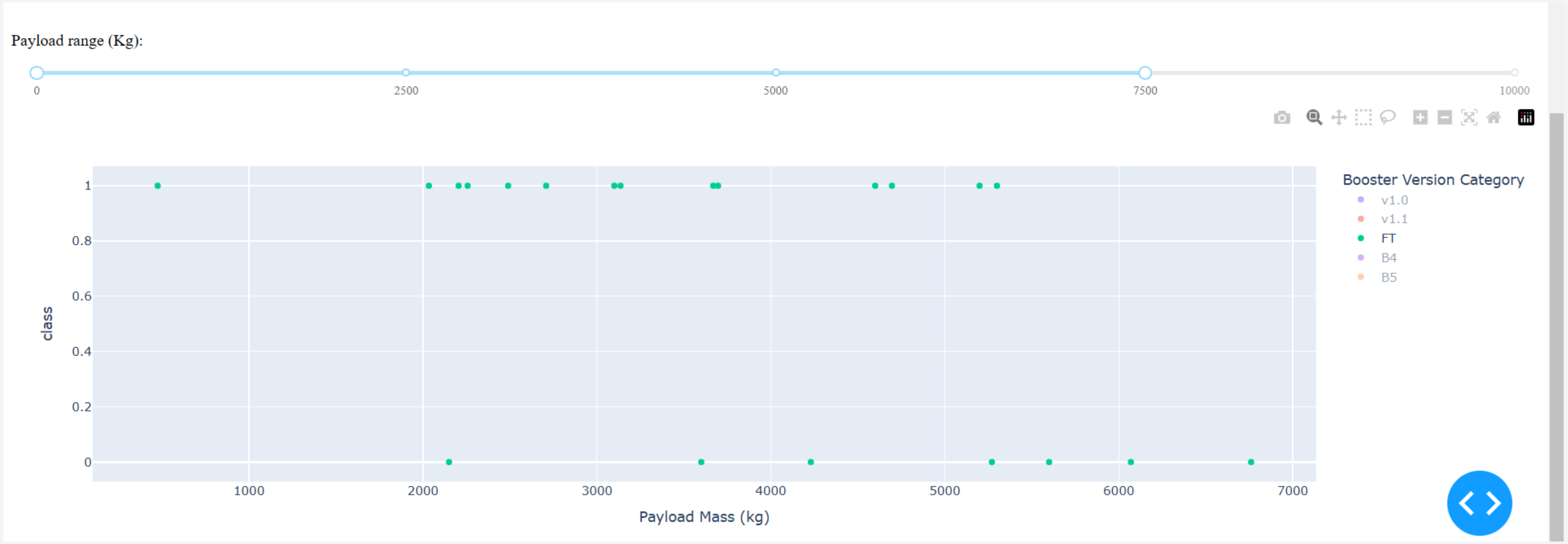
- KSC LC-39A Launch site have the highest success rate.


127.0.0.1:8050
Relaunch to update
SpaceX Launch Records Dashboard
KSC LC-39A
Total Success Launches for Site KSC LC-39A
23.1%
76.9%
1
0

Created by Parul Verma

Payload vs. Launch Outcome for all sites

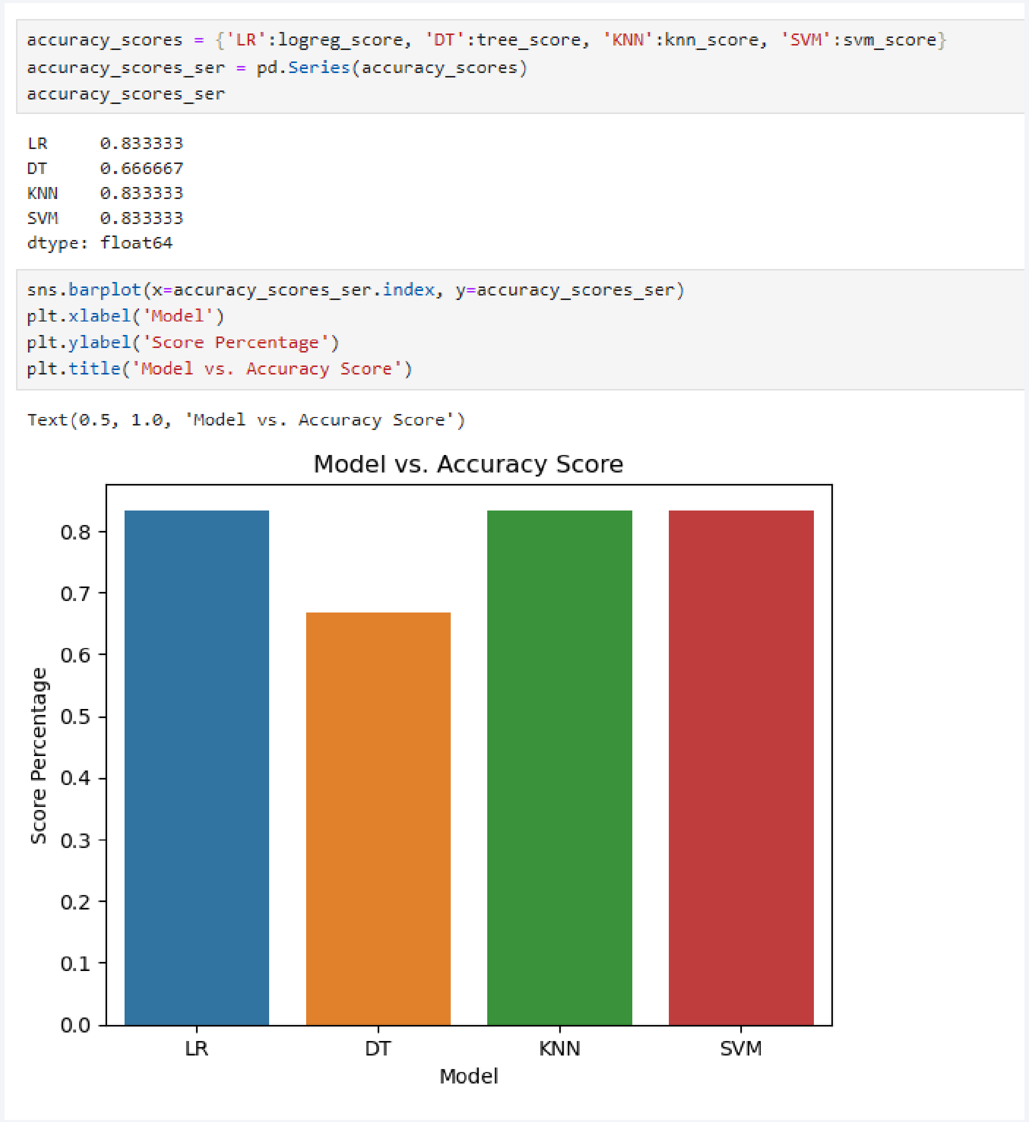
- FT and B4 booster version have high success rate with light payload mass.



Section 5

Predictive Analysis (Classification)

Classification Accuracy



```
accuracy_scores = {'LR':logreg_score, 'DT':tree_score, 'KNN':knn_score, 'SVM':svm_score}
accuracy_scores_ser = pd.Series(accuracy_scores)
accuracy_scores_ser
```

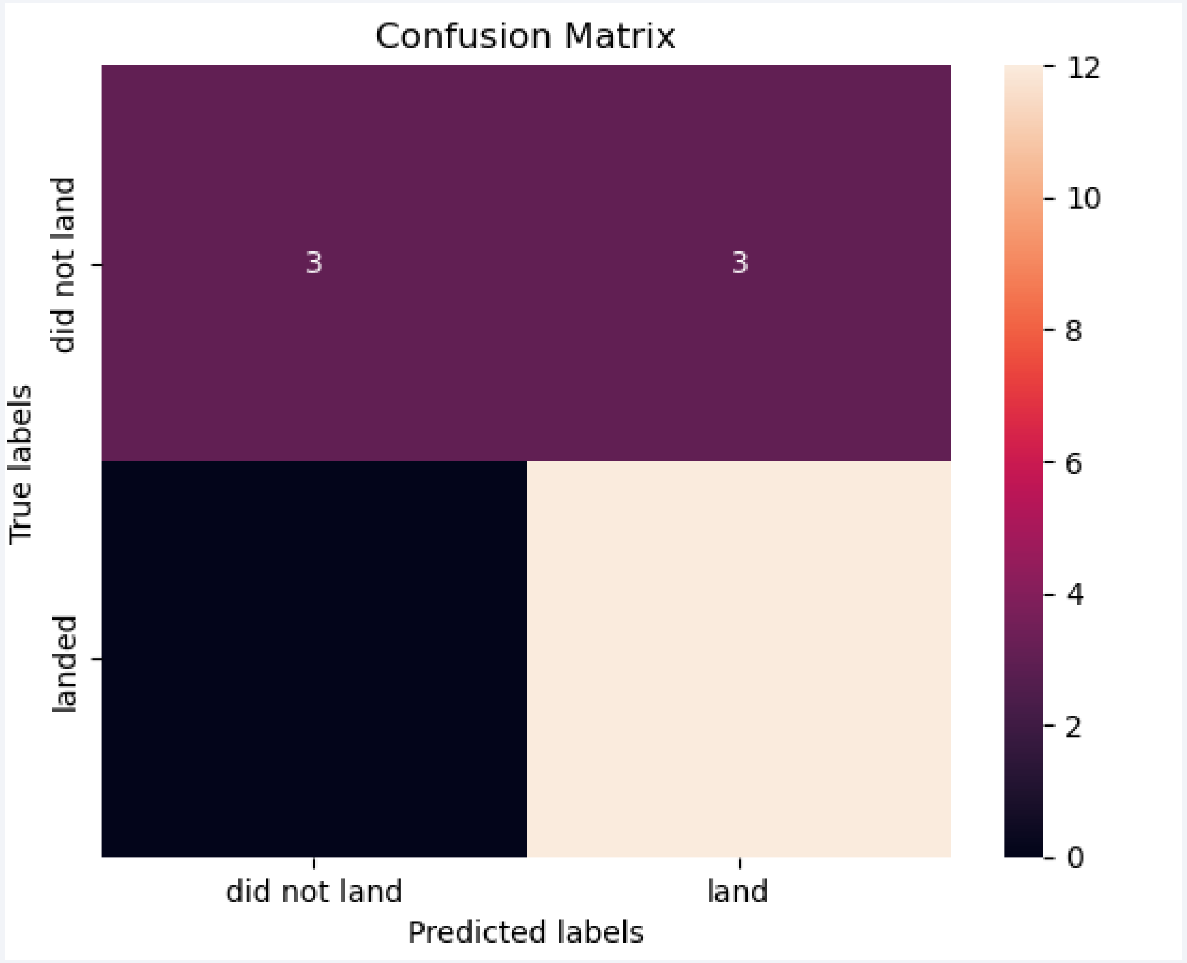
```
LR      0.833333
DT      0.666667
KNN     0.833333
SVM     0.833333
dtype: float64
```

```
sns.barplot(x=accuracy_scores_ser.index, y=accuracy_scores_ser)
plt.xlabel('Model')
plt.ylabel('Score Percentage')
plt.title('Model vs. Accuracy Score')
```

```
Text(0.5, 1.0, 'Model vs. Accuracy Score')
```

- All models performed well except Decision Tree.
- SVM, KNN and Logistic Regression models had 83.33% accuracy score on test data while Decision Tree had only 66.67%.

Confusion Matrix



- SVM, KNN and Logistic Regression models had same confusion matrix as shown in the picture.
- These models predicted 12 success launches and 3 failed launches correctly. However, 3 failed launches are predicted as successful launches.

Conclusions

- As per EDA analysis, ES-L1, GEO, HEO and SSO orbits have high success rates while GTO have lowest success rate of launch.
- For heavy payloads, successful landing rate is more for Polar, LEO and ISS orbits.
- Launch site maps clearly show that there are high chances of successful launch if the selected launch site is KSC LC-39A.
- Plotly dashboard also clearly state that there are high chances of successful launch for launch site KSC LC-39A.
- SVM, KNN and Logistic Regression models can be used to predict the outcome of future launches.

Appendix

- All references added to the respective slides.

Thank you!