IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Veronika Imrik
01.11.2024

Presentation contents

Introduction

- **Background**

  The commercial space age has arrived, making space travel more accessible. Companies like Virgin Galactic, Rocket Lab, and Blue Origin are pioneering various aspects of space travel and satellite services. Among these, SpaceX stands out with significant achievements such as sending spacecraft to the International Space Station, launching the Starlink satellite internet constellation, and conducting manned space missions. A key factor in SpaceX's success is the relatively low cost of its Falcon 9 rocket launches, priced at $62 million compared to other providers' $165 million. This cost efficiency is largely due to SpaceX's ability to reuse the first stage of its rockets.

- **Business problem**

  Space Y, a new rocket company and aims to compete with SpaceX. However, Space Y faces a significant challenge: determining the cost of each rocket launch. Unlike SpaceX, which has mastered the art of reusing the first stage of its rockets, Space Y needs to predict whether the first stage of their rockets will land successfully. This prediction is crucial because it directly impacts the overall cost of the launch.

Executive Summary

- Summary of methodologies
    - Data collection
    - Data wrangling
    - EDA with data visualisation
    - EDA with SQL
    - Intecartive map building with Folium
    - Dashboard building with Plotly Dash
    - Predictive analysis with classification
- Summary of all results
    - EDA result
    - Interactive analytics
    - Predictive analytisis

Section 1

Methodology

Methodology

- Collect data using SpaceX Rest API
- Wrangle data – filtering data, handling misisng values
- Explore data via EDA with SQL and data visulization techniques
- Visualize the data using Folium and Plotly Dash
- Build models to predict landing outcomes using classification models.

Data Collection

Data sets are collecting used 2 methods.

- 1. API
  - Collected data from SpaceX's open API
  - Retrived and processed the data with GET request
  - Ensure that only Falcon9 lanunches will be download
  - Filled missing payload values with avarage values
- 2. Web Scrapping
  - Requested Falcon9 and Falcon Heavy launch data from Wikipedia
  - Accessed to the Falcon9 launch pages via a direct link
  - Extracted column names from the HTML table
  - Parsed and transformed data into a Pandas data frame

Data Collection – SpaceX API

Request and parse the SpaceX launch data using the GET request

Filter the data frame to include Falcon 9 launches

Dealing with missing values

Source: https://github.com/VeronikaImrik/Applied-data-science-capstone/blob/main/Lab1%20-%20collecting%20data%20-%20jupyter-labs-spacex-data-collection-api.ipynb

Data Collection - Scraping

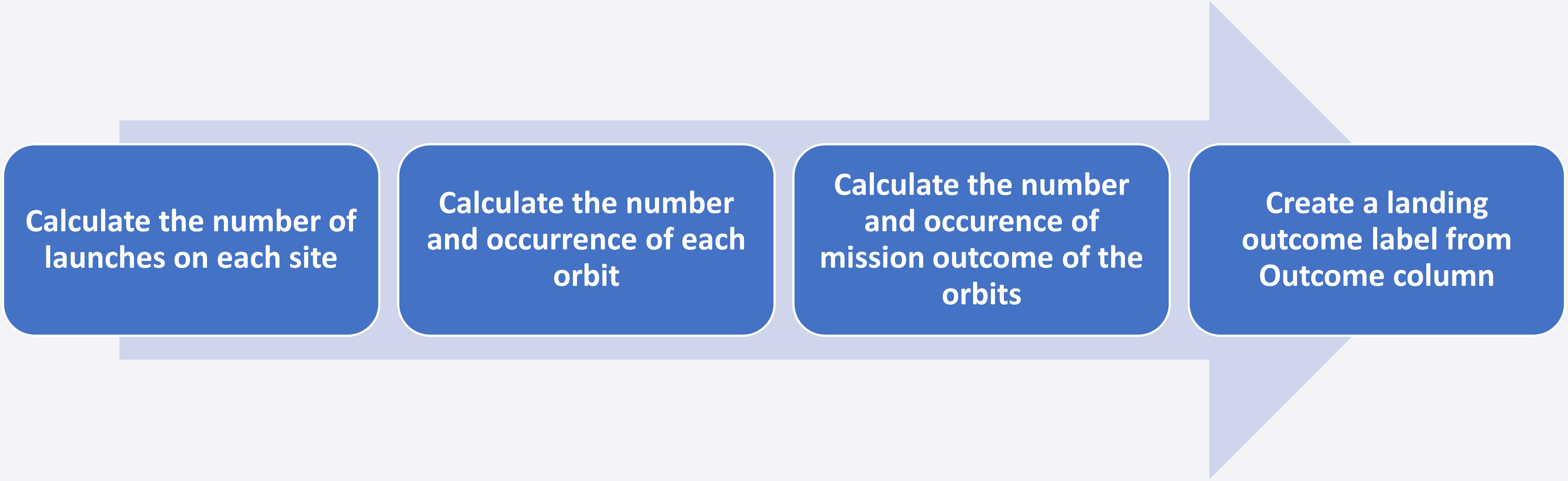
Request the Falcon9 Launch Wiki page from its URL

Extract all column/variable names from the HTML table header

Create a data frame by parsing the launch HTML tables

Source: https://github.com/VeronikaImrik/Applied-data-science-capstone/blob/main/Lab1.2%20-Web%20scraping%20Falcon%209%20and%20Falcon%20Heavy%20Launches%20Records%20from%20Wikipedia%20-jupyter-labs-webscraping.ipynb

Data Wrangling



Source: https://github.com/VeronikaImrik/Applied-data-science-capstone/blob/main/Lab2%20Data%20wrangling%20-%20labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

- Bar chart
  - To compare the success rate of each orbit
- Scatter plot chart
  - Identify the correlation between:
    - Launch site vs Flight number
    - Payload mass vs Launch site
    - Flight number vs Orbit type
    - Payload mass vs Orbit type
- Line chart
  - Visualize the launch success yearly trend

Source: https://github.com/VeronikaImrik/Applied-data-science-capstone/blob/main/Lab4%20-%20Assignment%20Exploring%20and%20Preparing%20Data%20-%20edadataviz.ipynb

EDA with SQL

1. %sql select distinct(LAUNCH_SITE) from SPACEXTBL;
2. %sql select * from SPACEXTBL where LAUNCH_SITE like 'CCA%' limit 5;
3. %sql select sum(PAYLOAD_MASS__KG_) as payloadmass from SPACEXTBL;
4. %sql select avg(PAYLOAD_MASS__KG_) as payloadmass from SPACEXTBL;
5. %sql select min(DATE) from SPACEXTBL;
6. %sql select BOOSTER_VERSION from SPACEXTBL where LANDING_OUTCOME='Success (drone ship)' and
7. %sql select count(MISSION_OUTCOME) as missionoutcomes from SPACEXTBL GROUP BY MISSION_OUTCOME;
8. %sql select BOOSTER_VERSION as boosterversion from SPACEXTBL where PAYLOAD_MASS__KG_=(select max(PAYLOAD_MASS__KG_) from SPACEXTBL);
9. %sql SELECT substr(Date, 6, 2) as month, MISSION_OUTCOME, BOOSTER_VERSION, LAUNCH_SITE FROM SPACEXTBL WHERE [Landing_Outcome] = 'Failure (drone ship)' AND substr(Date, 1, 4) = '2015';
10. %sql SELECT LANDING_OUTCOME FROM SPACEXTBL WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' ORDER BY DATE DESC;

Source: https://github.com/VeronikaImrik/Applied-data-science-capstone/blob/main/Lab3%20-%20jupyter-labs-eda-sql-coursera_sqllite.ipynb

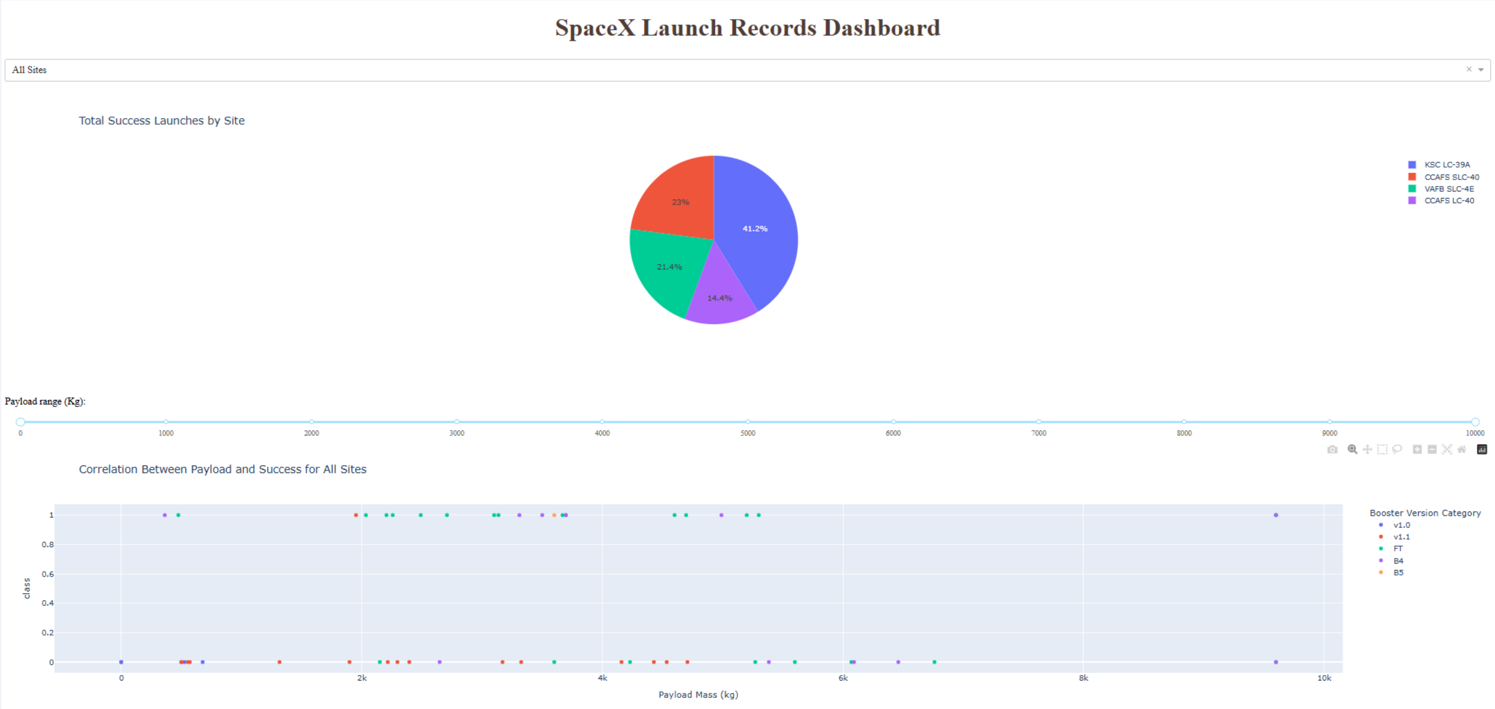
Build an Interactive Map with Folium

- folium.Circle
    - to add a highlighted circle area with a text label on a specific coordinate
- folium.Marker
    - to marker_cluster
- folium.Popup
    - is used to display additional information when a user clicks on a marker on a map
- folium.Map
    - It allows you to set the initial location, zoom level, and other map settings.

Source: https://github.com/VeronikaImrik/Applied-data-science-capstone/blob/main/Lab5%20-%20Interactive%20Visual%20Analytics%20with%20Folium%20-%20lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

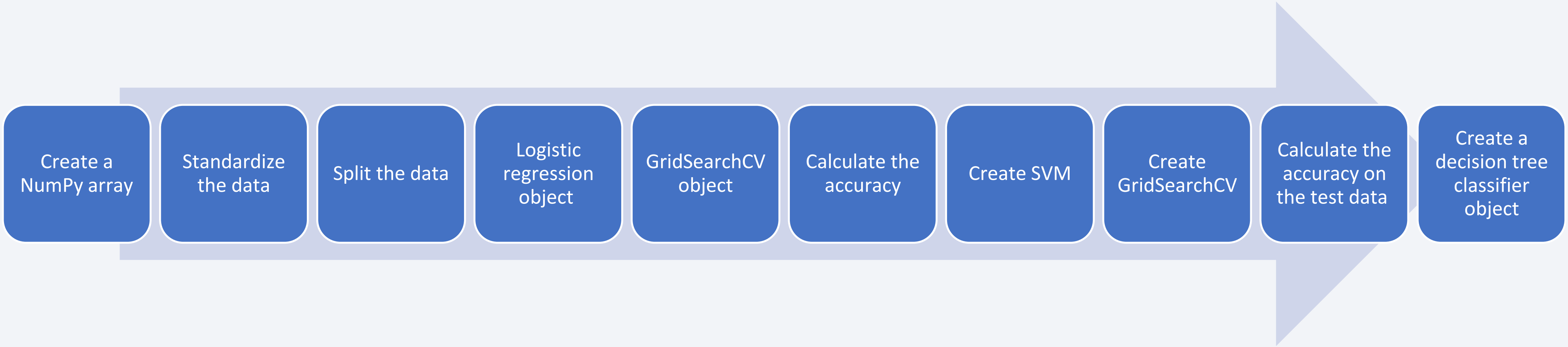
Pie chart to show the total successful launches count for all sites. Also added a slider to select the payload range.
Scatter chart to show the correlation between payload and launch success



Source: https://github.com/VeronikaImrik/Applied-data-science-capstone/blob/main/Lab6%20-%20spacex_dash_app.py

Predictive Analysis (Classification)

- Perform exploratory Data Analysis and determine Training Labels
  - create a column for the class
  - Standardize the data
  - Split into training data and test data
- Find best Hyperparameter for SVM, Classification Trees and Logistic Regression
  - Find the method performs best using test data



Source: https://github.com/VeronikaImrik/Applied-data-science-capstone/blob/main/Lab7-SpaceX_Machine_Learning_Prediction_Part_5.ipynb

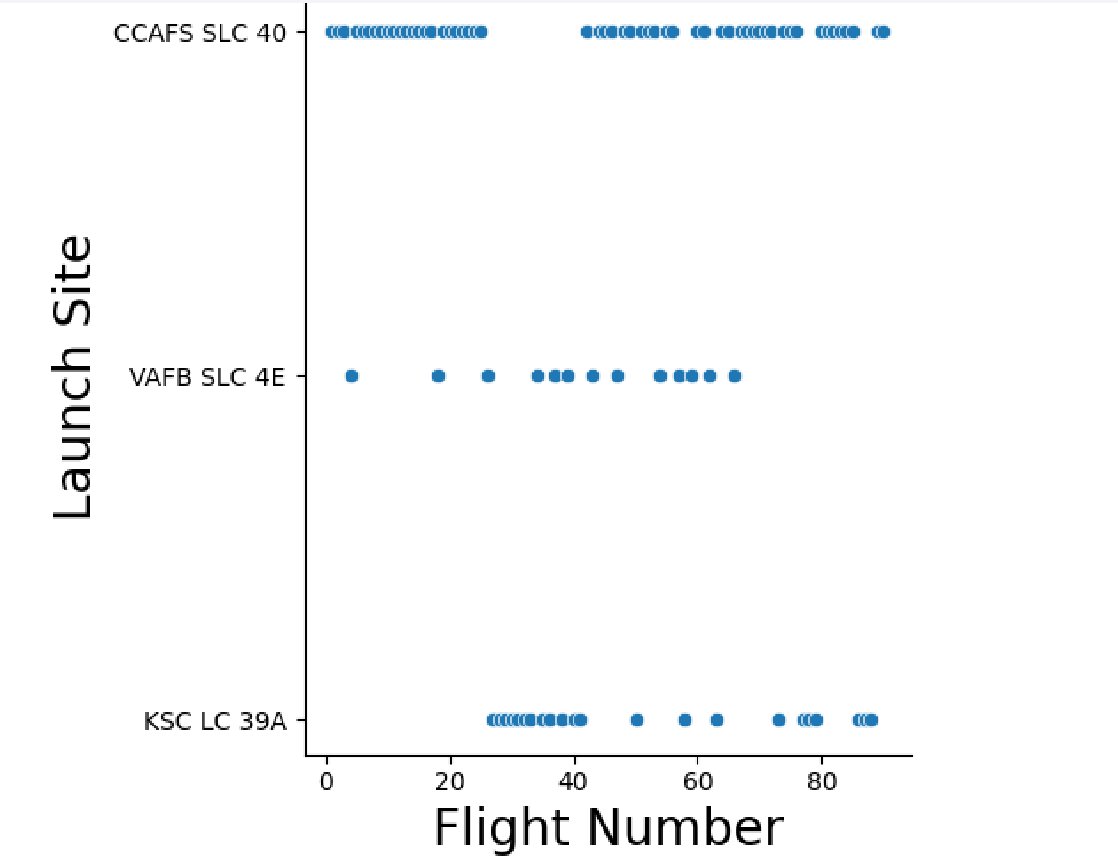
Results

- Exploratory data analysis:
  - Launch success has improved
  - KSC LC-39A has the highest success rate among landing sites
  - Orbits ES-L1, GEO, HEO, and SSO have a 100% cusses rate
- Interactive analytics
  - Launch sites are fare engough to make demage
- Predictive analysis results
  - Decision Tree model is the best predictive model for the dataset

Section 2

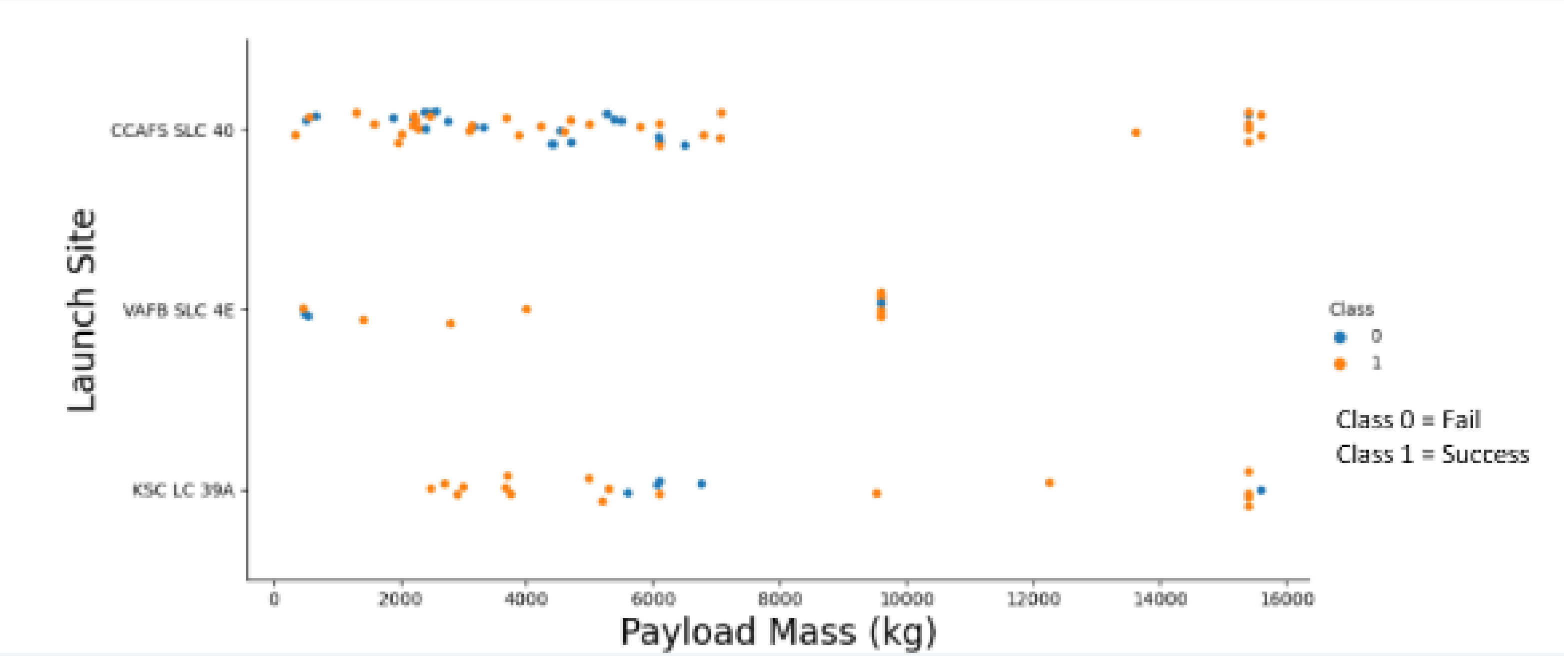
Insights drawn from EDA

Flight Number vs. Launch Site



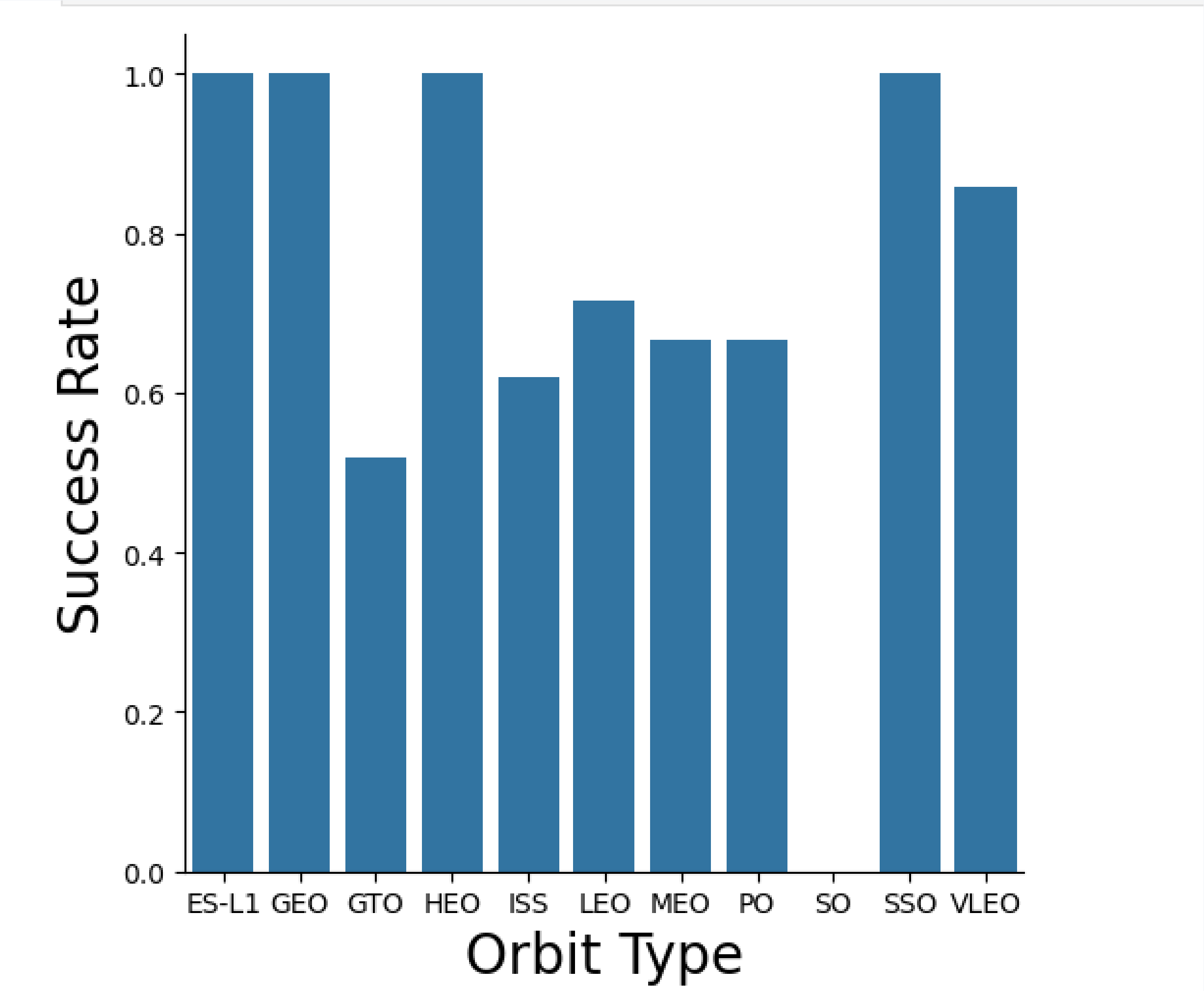
Earlier flights have lower succes rate and later flights have the opposite.

Payload vs. Launch Site



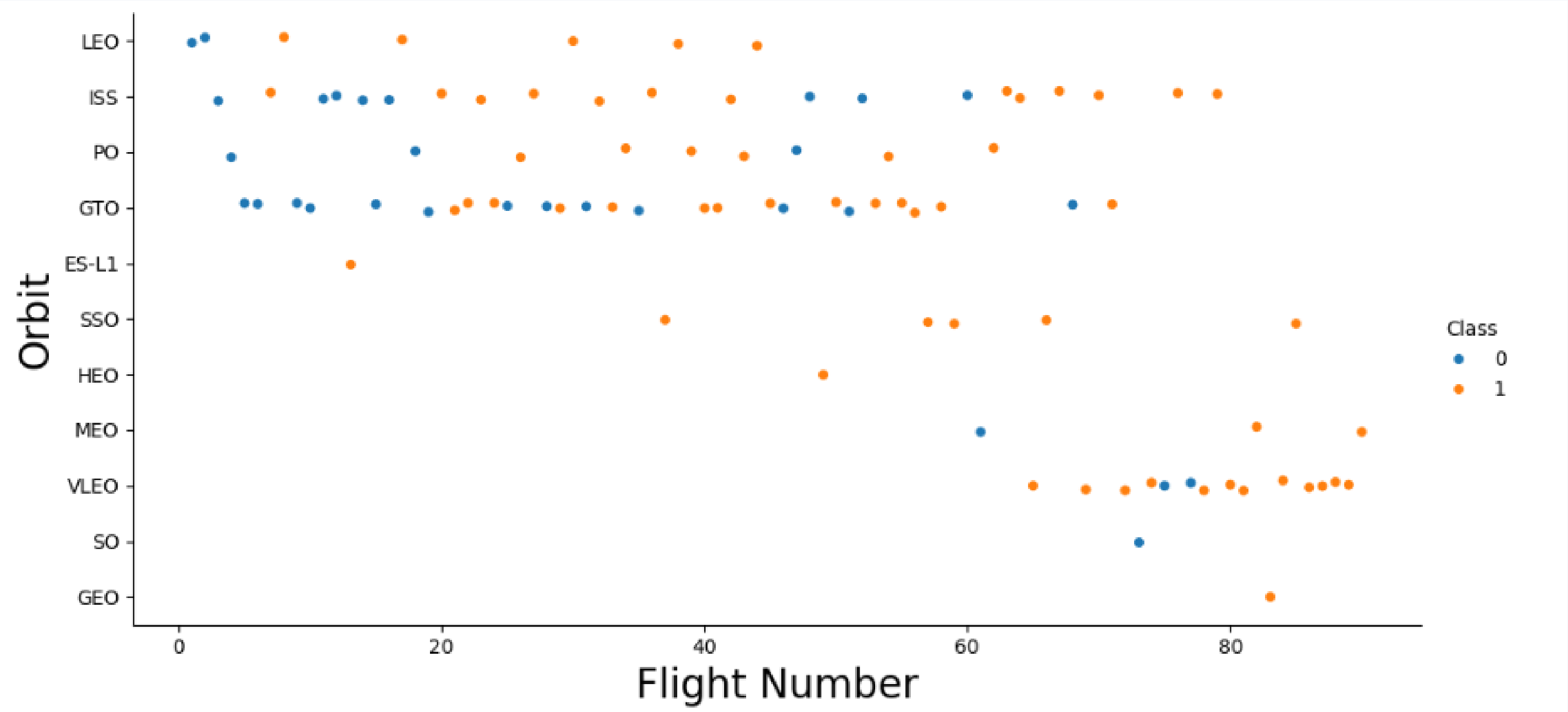
If the payload is higher, the succes rate is also higher.

Success Rate vs. Orbit Type



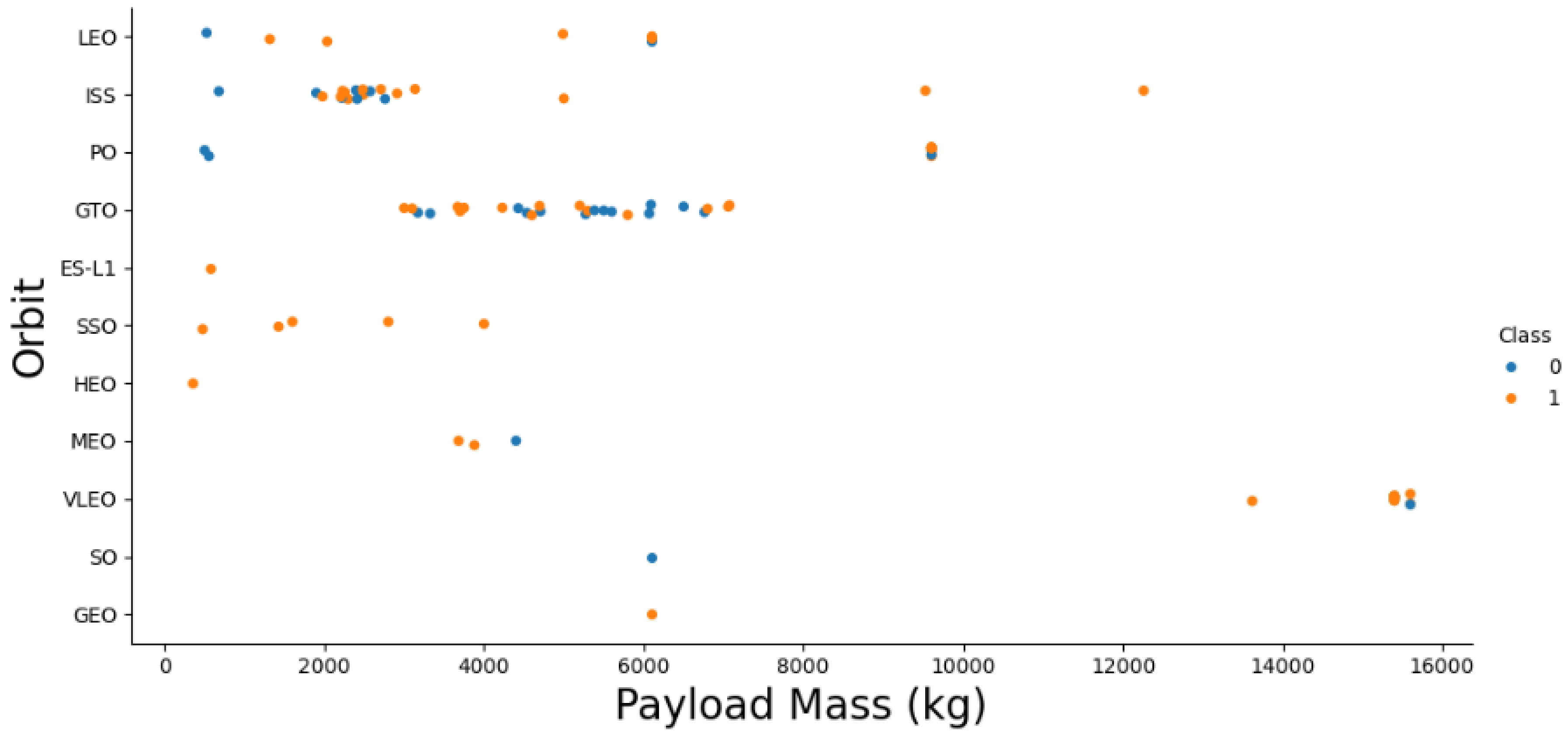
ES-L1, GEO, HEO and SSO has the highest success rate

Flight Number vs. Orbit Type



In the LEO orbit, success seems to be related to the number of flights. Conversely, in the GTO orbit, there appears to be no relationship between flight number and success.

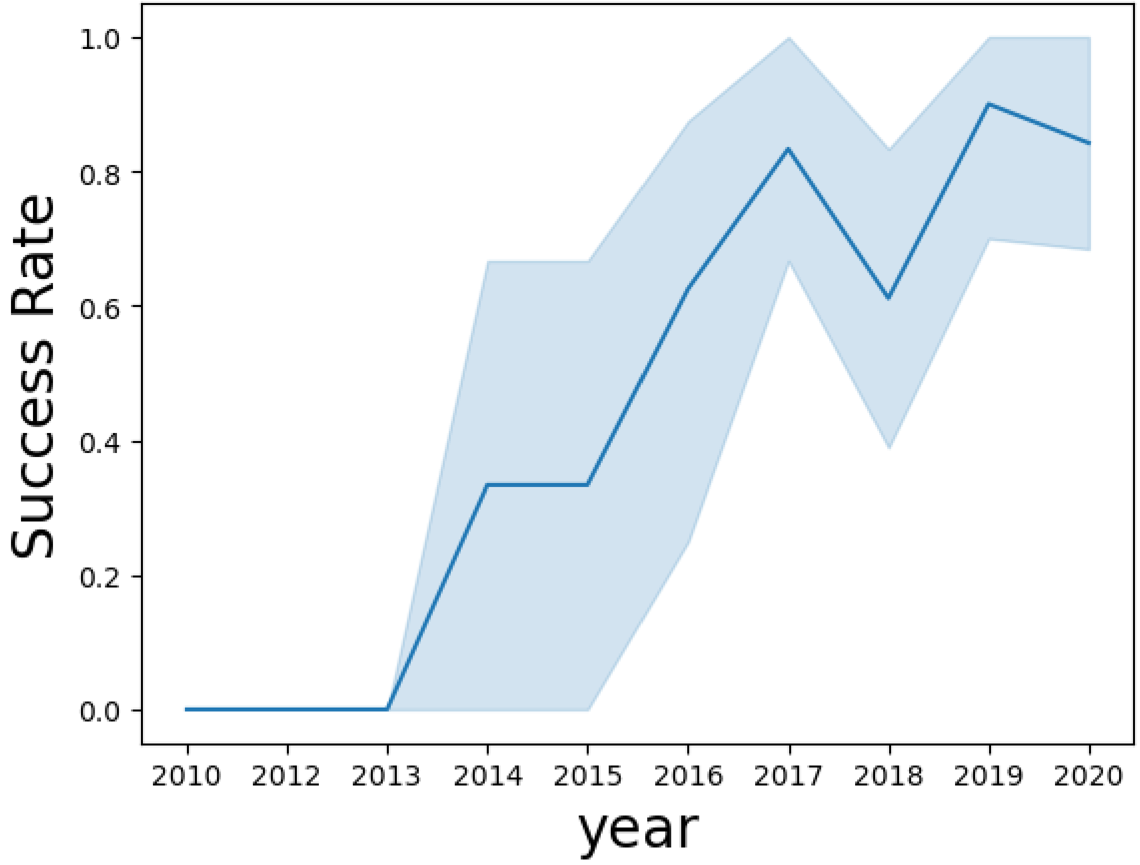
Payload vs. Orbit Type



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

However, for GTO, it's difficult to distinguish between successful and unsuccessful landings as both outcomes are present.

Launch Success Yearly Trend



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

All Launch Site Names

Task 1

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 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
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success

Total Payload Mass

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

```
]: %sql select sum(PAYLOAD_MASS__KG_) as payloadmass from SPACEXTBL;
```

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

]:

payloadmass
619967

Average Payload Mass by F9 v1.1

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

```
%sql select avg(PAYLOAD_MASS__KG_) as payloadmass from SPACEXTBL;
```

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

payloadmass
6138.287128712871

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) from SPACEXTBL;
```

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

min(DATE)
2010-06-04

Successful Drone Ship Landing with Payload between 4000 and 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 SPACEXTBL where LANDING_OUTCOME='Success (drone ship)' and PAYLOAD_MASS__KG
```

```
 * 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

List the total number of successful and failure mission outcomes

```
%sql select count(MISSION_OUTCOME) as missionoutcomes from SPACEXTBL GROUP BY MISSION_OUTCOME;
```

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

missionoutcomes
1
98
1
1

Boosters Carried Maximum Payload

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

```
]: %sql select BOOSTER_VERSION as boosterversion from SPACEXTBL where PAYLOAD_MASS__KG_=(select max(PAYLOAD_MAS
```

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

]:

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

```
%sql SELECT substr(Date, 6, 2) as month, MISSION_OUTCOME, BOOSTER_VERSION, LAUNCH_SITE FROM SPACEXTBL WHERE
```

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

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

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

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 FROM SPACEXTBL WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' ORDER BY DATE DESC;
```

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

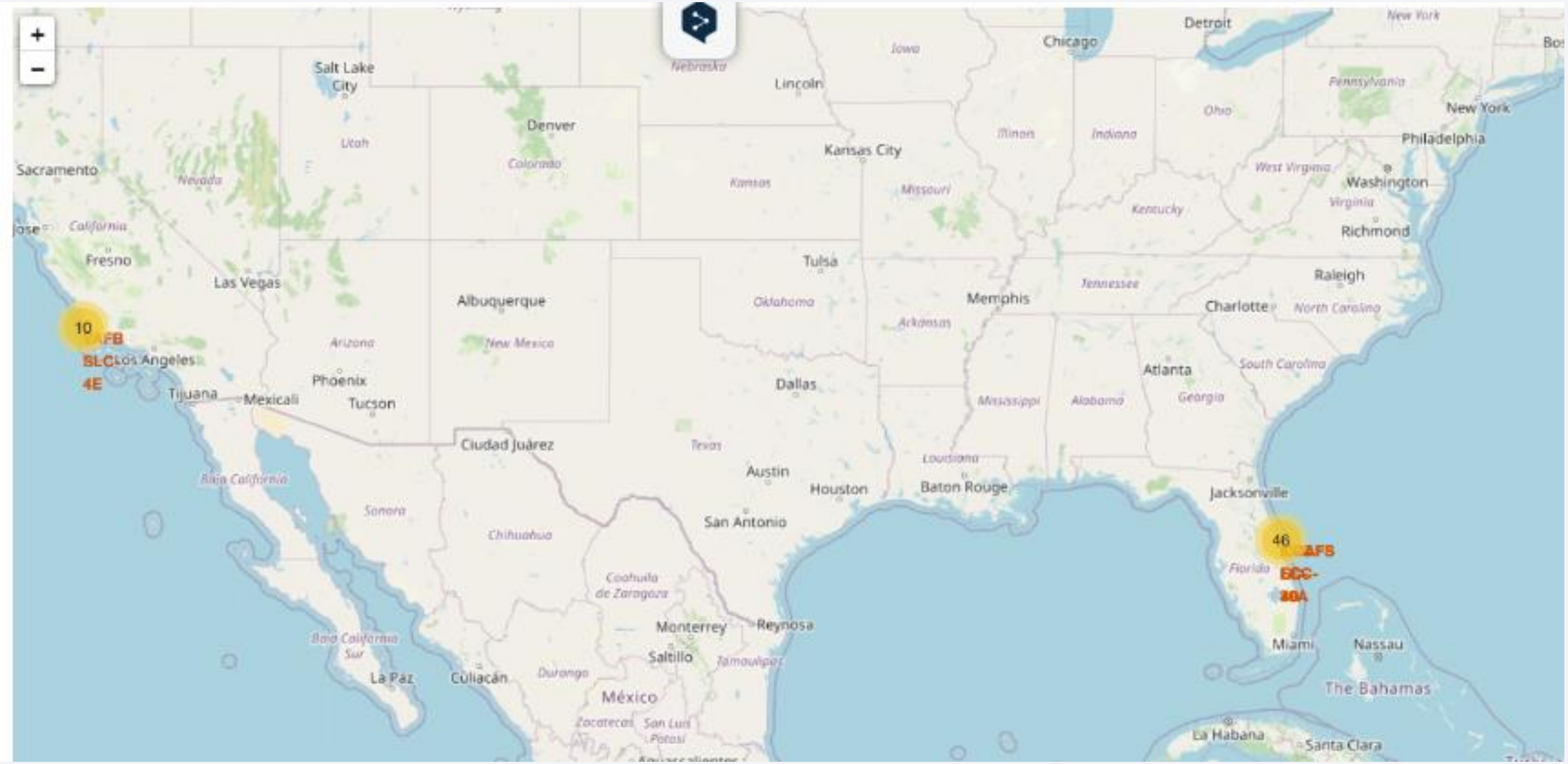
Landing_Outcome
No attempt
Success (ground pad)
Success (drone ship)
Success (drone ship)
Success (ground pad)
Failure (drone ship)
Success (drone ship)
Success (drone ship)
Success (drone ship)
Failure (drone ship)
Failure (drone ship)
Success (ground pad)
Precluded (drone ship)
No attempt
Failure (drone ship)
No attempt
Controlled (ocean)
Failure (drone ship)
Uncontrolled (ocean)
No attempt

Section 3

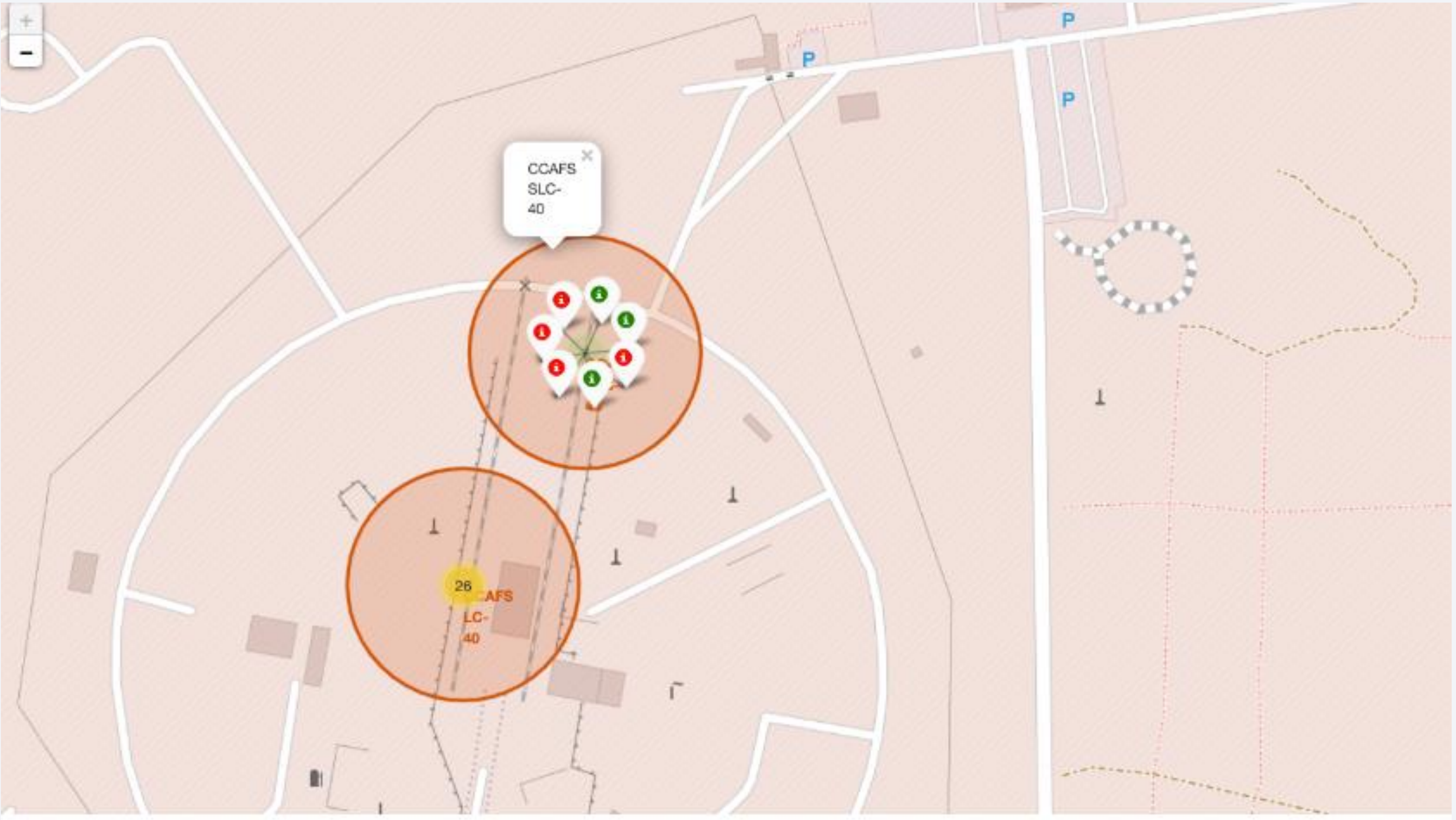
Launch Sites Proximities Analysis

Launch site map

Launch site map with markers



Launch site map with markers



Section 4

Build a Dashboard with Plotly Dash

SpaceX Launch Records Dashboard

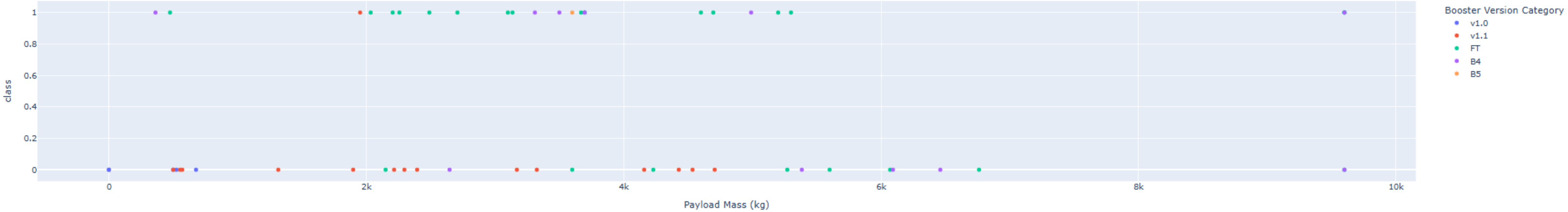
All Sites

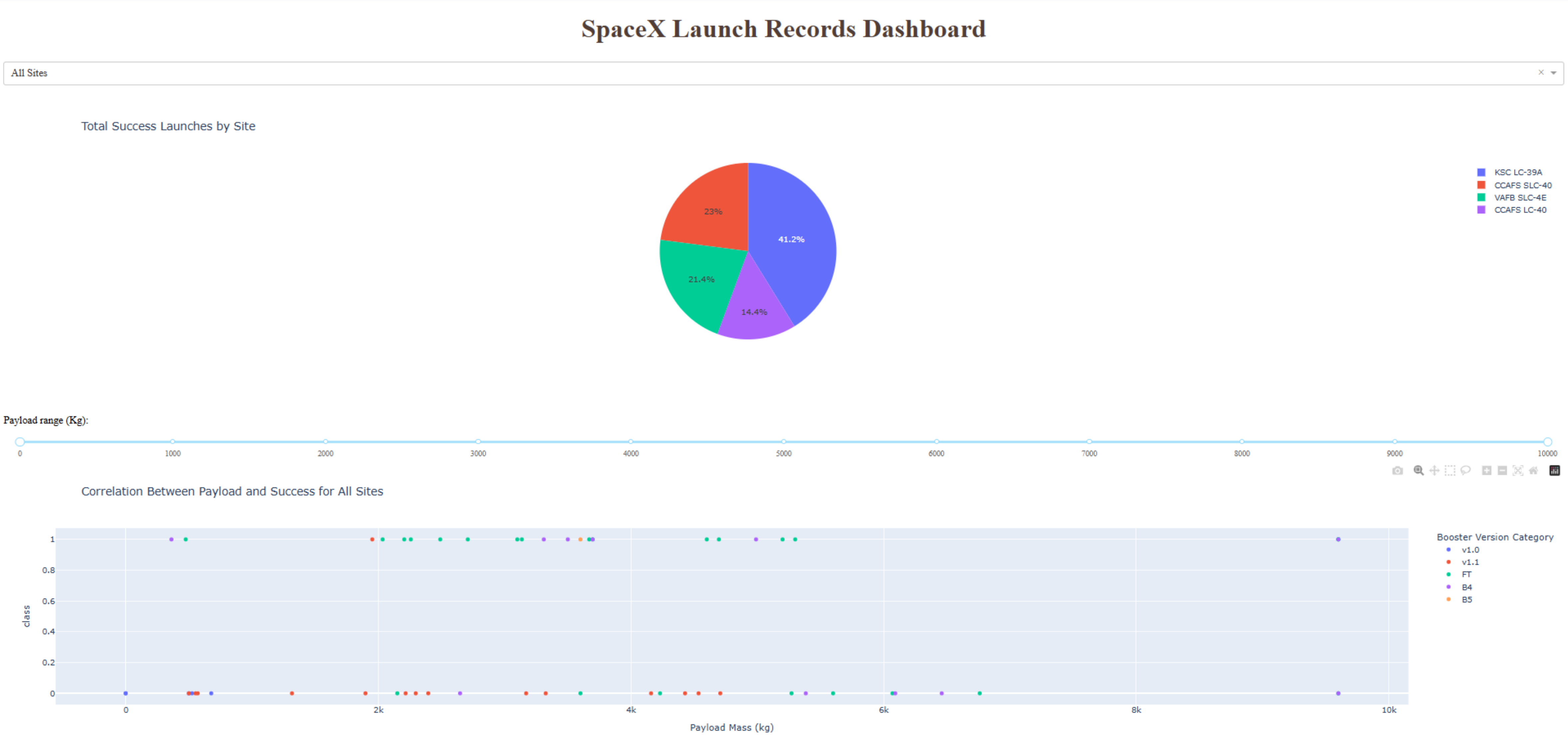
Total Success Launches by Site

Payload range (Kg):

0 1000 2000 3000 4000 5000 6000 7000 8000 9000 10000

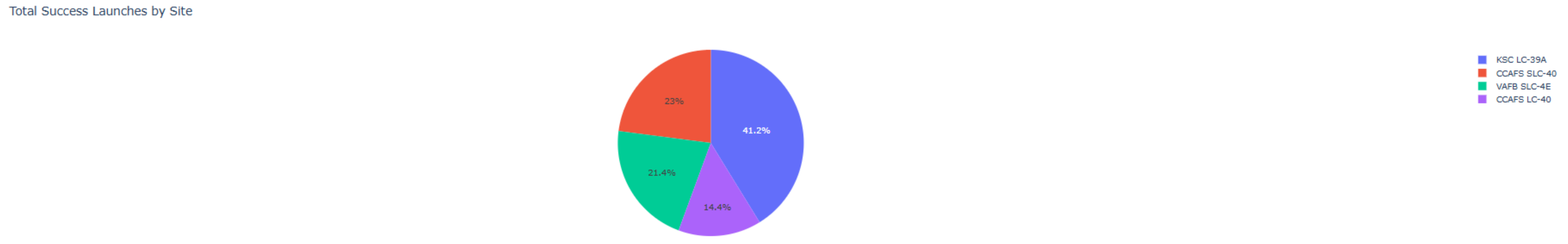
Correlation Between Payload and Success for All Sites




SpaceX Launch Records Dashboard
All Sites
Total Success Launches by Site
41.2%
23%
21.4%
14.4%
KSC LC-39A
CCAFS SLC-40
VAFB SLC-4E
CCAFS LC-40
Payload range (Kg):
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
10000
Correlation Between Payload and Success for All Sites
class
1
0.8
0.6
0.4
0.2
0
Payload Mass (kg)
0
2k
4k
6k
8k
10k
Booster Version Category
v1.0
v1.1
FT
B4
B5

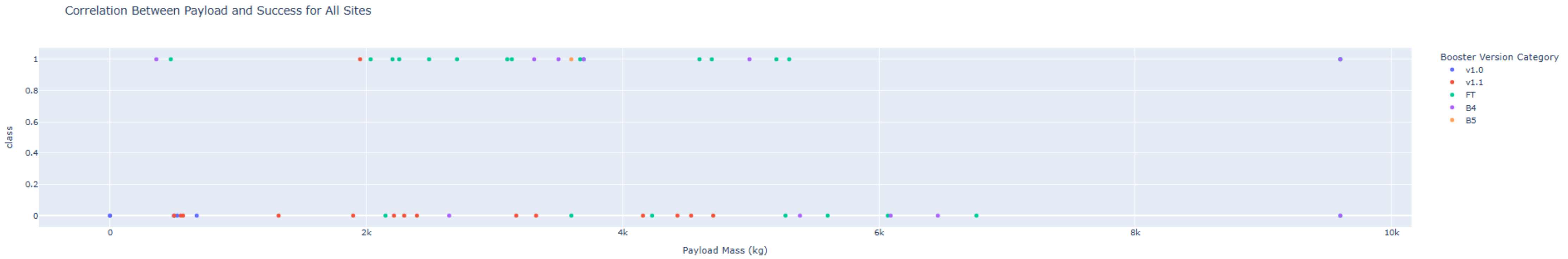
SpaceX Launch Records Dashboard

All Sites



Payload range (Kg):

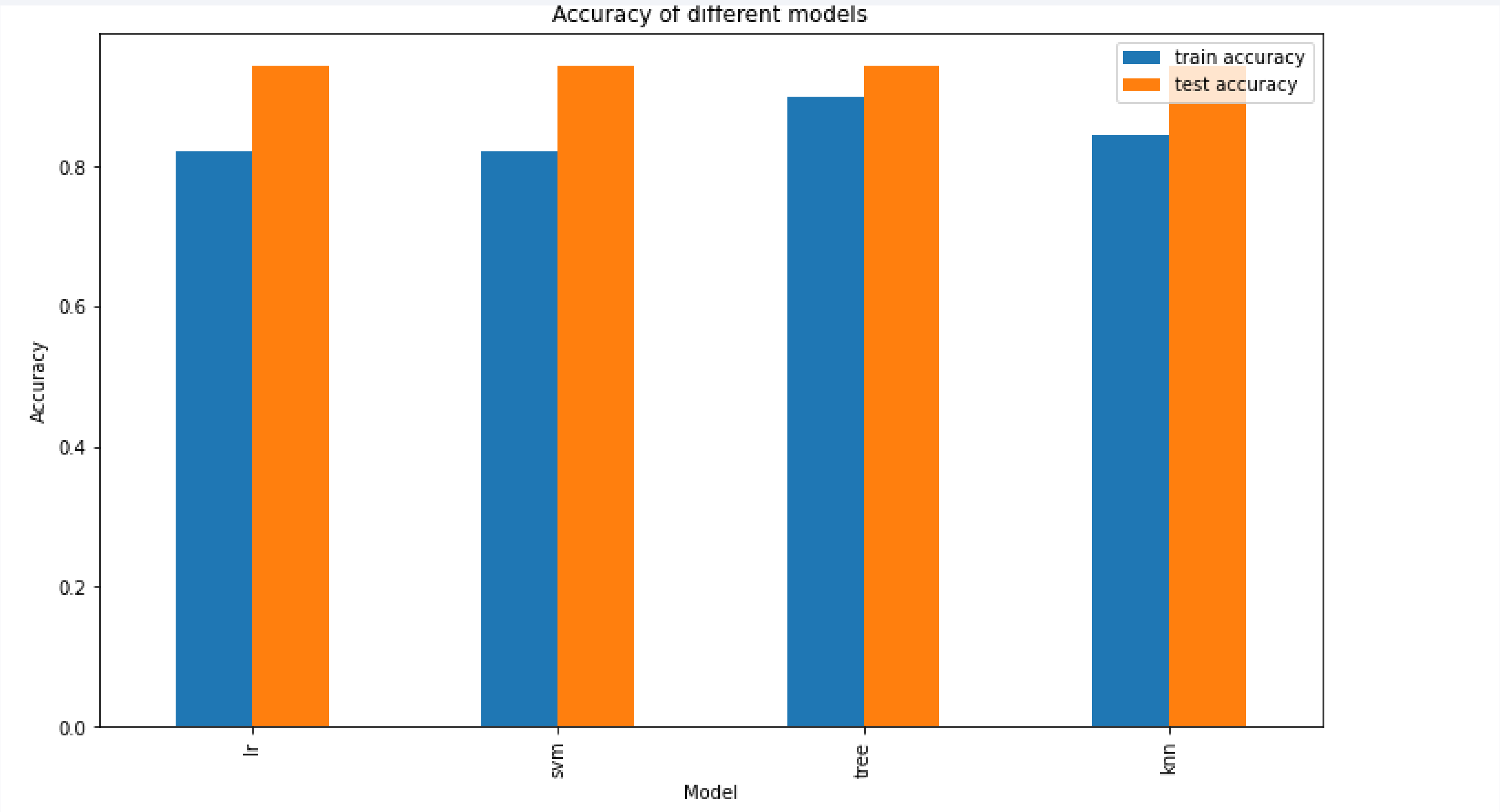
0 1000 2000 3000 4000 5000 6000 7000 8000 9000 10000



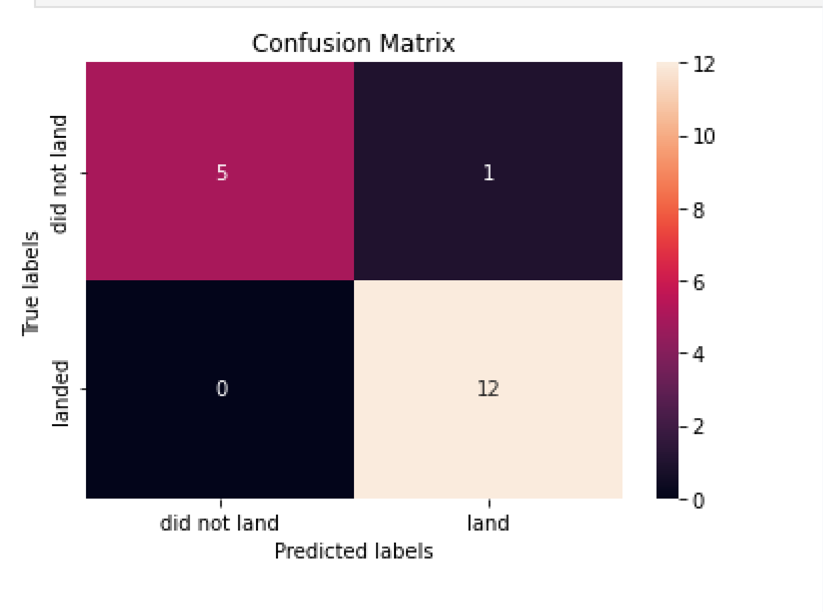
Section 5

Predictive Analysis (Classification)

Classification Accuracy



Confusion Matrix



TASK 11

Calculate the accuracy of tree_cv on the test data using the method `score` :

```
knn_score = knn_cv.score(X_test, Y_test)
knn_score
```

```
0.9444444444444444
```

We can plot the confusion matrix

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

Conclusions

- LR, SVM, KNN are top-performing models for forecasting outcomes in this data
- Lighter paylods have a higher perfomance
- Launch sites are fare engough to make demage

Appendix

- Python code snippets
- SQL queries, charts
- Notebook outputs

Thank you!