IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

HaiLv
2024/08/15

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection through API
  - Data Collection with Web Scraping
  - Data Wrangling
  - Exploratory Data Analysis with SQL
  - Exploratory Data Analysis with Data Visualization
  - Interactive Visual Analytics with Folium
  - Machine Learning Prediction

- Summary of all results
  - Exploratory Data Analysis result
  - Interactive analytics in screenshots
  - Predictive Analytics result

Introduction

- **Project background and context**

    SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because Space X can reuse the first stage. Therefore, if we can determine if the first stage will land, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch. This goal of the project is to create a machine learning pipeline to predict if the first stage will land successfully.

- **Problems you want to find answers**
    - What factors determine if the rocket will land successfully?
    - The interaction amongst various features that determine the success rate of a successful landing.
    - What operating conditions needs to be in place to ensure a successful landing program.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data from SpaceX was obtained from 2 sources:
    - SpaceX API(https://api.spacexdata.com/v4/rockets/)
    - Web Scraping (https://en.wikipedia.org/wiki/List_of_Falcon/_9/_and_Falcon_Heavy_l aunches)
- Perform data wrangling
  - Collected data was enriched by creating a landing outcome label based on outcome data after summarizing and analyzing features

Methodology

Executive Summary

- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Data that was collected until this step were normalized, divided in training and test data sets and evaluated by four different classification models, being the accuracy of each model evaluated using different combinations of parameters

Data Collection

- **Describe how data sets were collected.**
  - Datasets were collected from SpaceX API (https://api.spacexdata.com/v4/rockets/) and from Wikipedia (https://en.wikipedia.org/wiki/List_of_Falcon/_9/_and_Falcon_Heavy_launches), using web scraping techniques

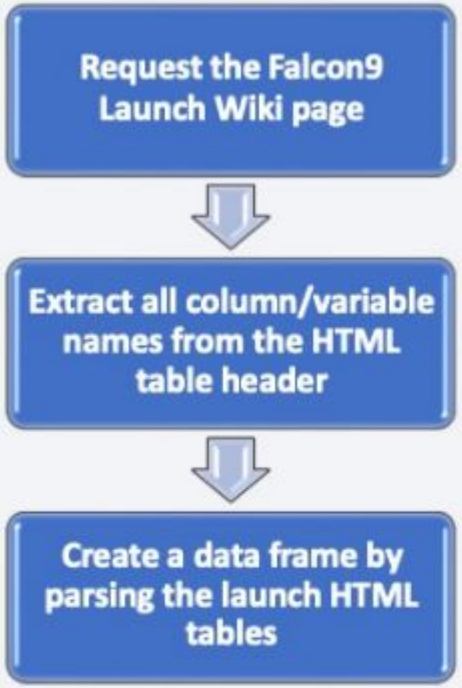
Data Collection – SpaceX API

- SpaceX offers a public API from where data can be obtained and then used;
- This API was used according to the flowchart beside and then data is persisted.
- Source: https://github.com/Lucky-Lee-029/DataScience/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

Data Collection - Scraping

- Data from SpaceX launches can also be obtained from Wikipedia;
- Data are downloaded from Wikipedia according to the flowchart and then persisted.

- Source: https://github.com/Lucky-Lee-029/DataScience/blob/main/jupyter-labs-webscraping.ipynb



Data Wrangling

- Initially some Exploratory Data Analysis (EDA) was performed on the dataset.
- Then the summary launches per site, occurrences of each orbit and occurrences of mission outcome per orbit type were calculated.
- Finally, the landing outcome label was created from Outcome column.

- Source: https://github.com/Lucky-Lee-029/DataScience/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

- The following SQL queries were performed:
  - Names of the unique launch sites in the space mission;
  - Top 5 launch sites whose name begins with the string 'CCA';
  - Total payload mass carried by boosters launched by NASA (CRS);
  - Average payload mass carried by booster version F9 v1.1;
  - Date when the first successful landing outcome in ground pad was achieved;
  - Names of the boosters which have success in drone ship and have payload mass between 4000 and 6000 kg;
  - Total number of successful and failure mission outcomes;
  - Names of the booster versions which have carried the maximum payload mass;
  - Failed landing out comes in droneship, their booster versions, and launch site names for in year 2015;
  - Rank of the count of landing outcomes (such as Failure (droneship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20.

- Source: https://github.com/Lucky-Lee-029/DataScience/blob/main/edadataviz.ipynb

EDA with SQL

- To explore data, scatterplots and bar plots were used to visualize the relationship between pair of features:
  - Payload Mass X Flight Number, Launch Site X Flight Number, Launch Site X Payload Mass, Orbit and Flight Number, Payload and Orbit
- Source: https://github.com/Lucky-Lee-029/DataScience/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- Markers, circles, lines and marker clusters were used with Folium Maps
- Markers indicate points like launch sites;
- Circles indicate highlighted areas around specific coordinates, like NASA Johnson Space Center;
- Marker clusters indicates groups of events in each coordinate, like launches in a launch site; and
- Lines are used to indicate distances between two coordinates.
- Source: https://github.com/Lucky-Lee-029/DataScience/blob/main/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

- We built an interactive dashboard with Plotly dash
- We plotted pie charts showing the total launches by a certain sites
- We plotted scatter graph showing the relationship with Outcome and Payload Mass (Kg) for the different booster version
- Source: https://github.com/Lucky-Lee-029/DataScience/blob/main/spacex_dash_app.py

Predictive Analysis (Classification)

- We loaded the data using numpy and pandas, transformed the data, split our data into training and testing.
- We built different machine learning models and tune different hyperparameters using GridSearchCV.
- We used accuracy as the metric for our model, improved the model using feature engineering and algorithm tuning.
- We found the best performing classification model.
- Source: https://github.com/Lucky-Lee-029/DataScience/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results

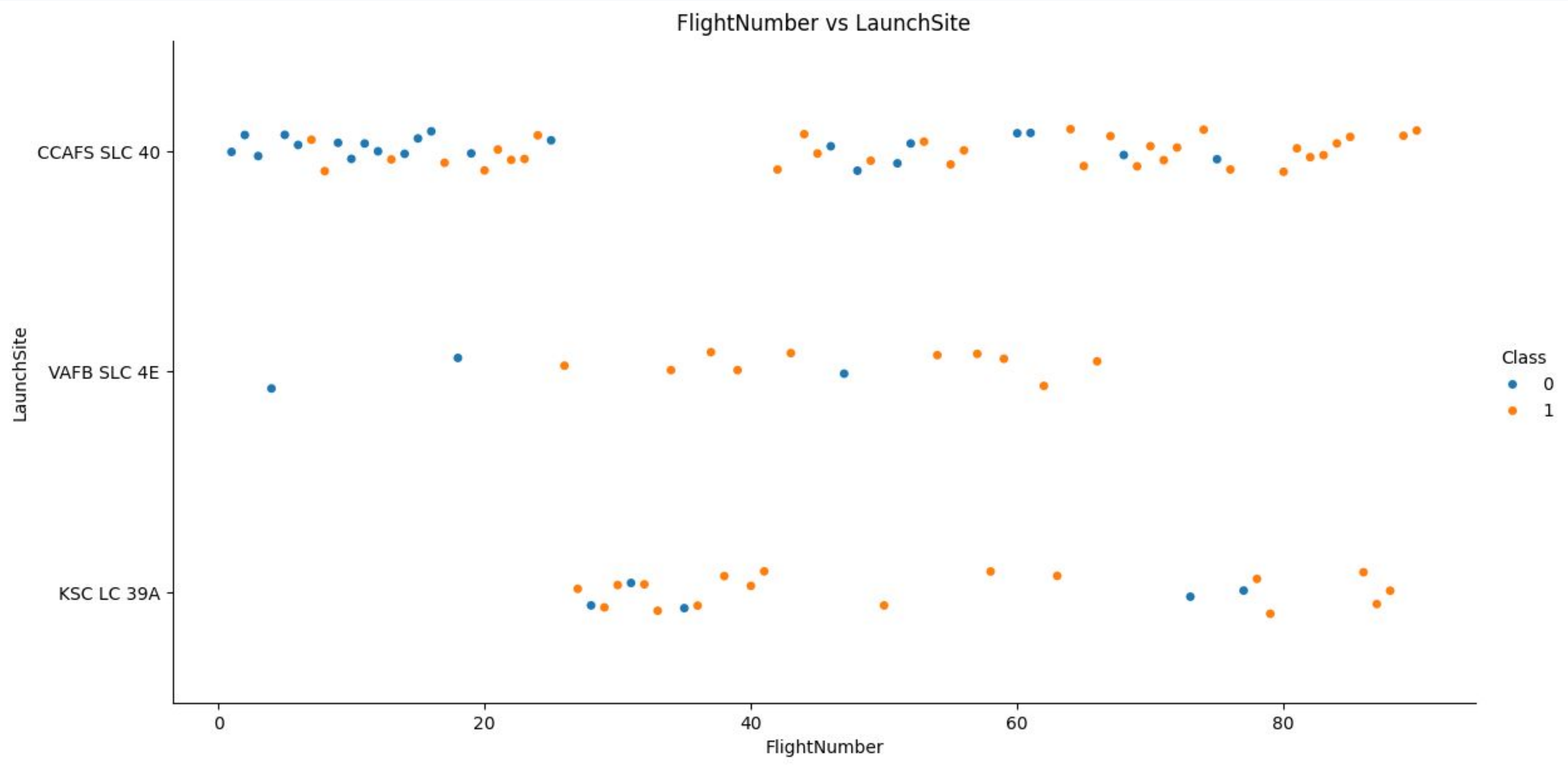
- SpaceX uses 4 different launch sites;
- The first launches were done to SpaceX itself and NASA;
- The average payload of F9 v1.1 booster is 2,928 kg;
- The first success landing outcome happened in 2015 fiver year after the first launch;
- Many Falcon 9 booster versions were successful at landing in drone ships having payload above the average; Almost 100% of mission outcomes were successful;
- Two booster versions failed at landing in drone ships in 2015: F9 v1.1 B1012 and F9 v1.1 B1015;
- The number of landing outcomes became as better as years passed.

Section 2

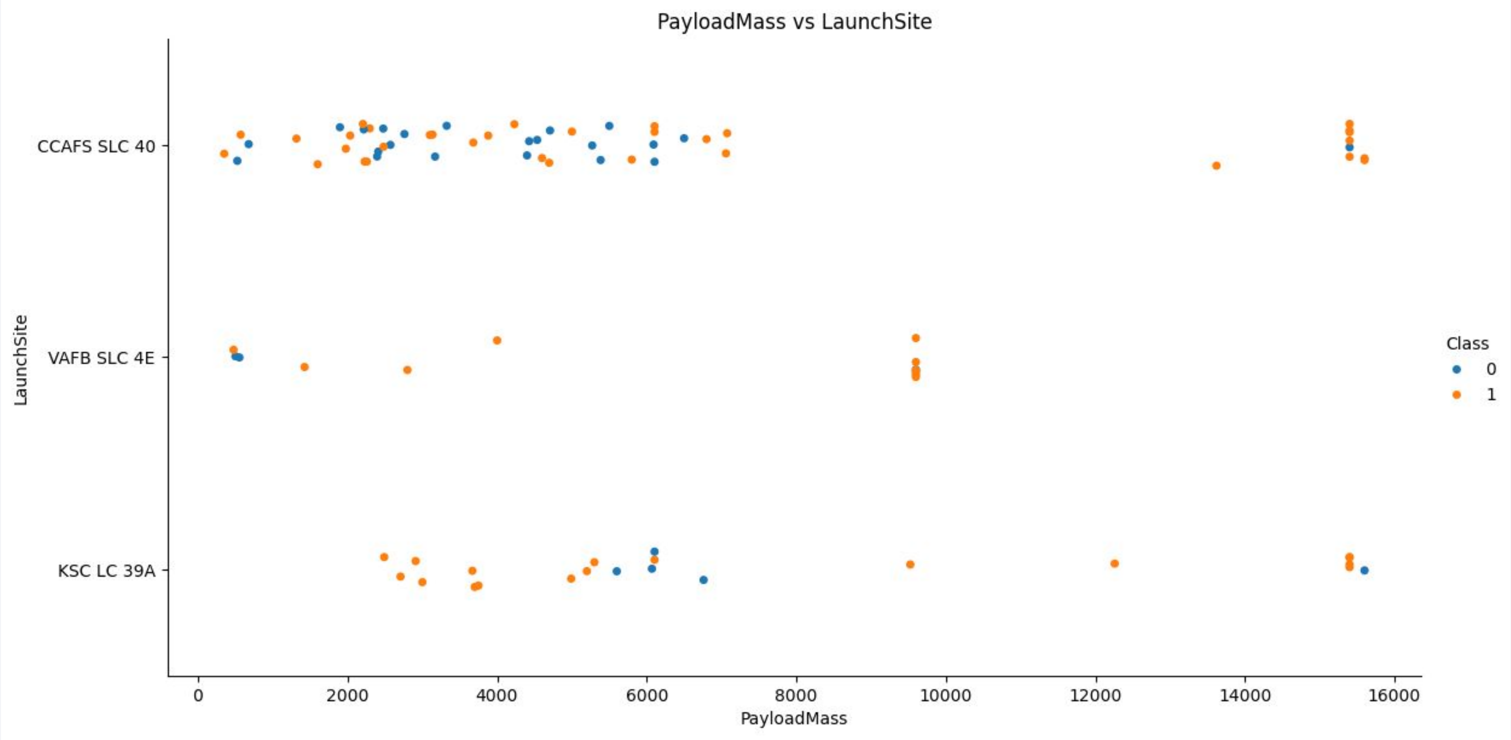
Insights drawn from EDA

Flight Number vs. Launch Site

- From the plot, we found that the larger the flight amount at a launch site, the greater the success rate at a launch site.

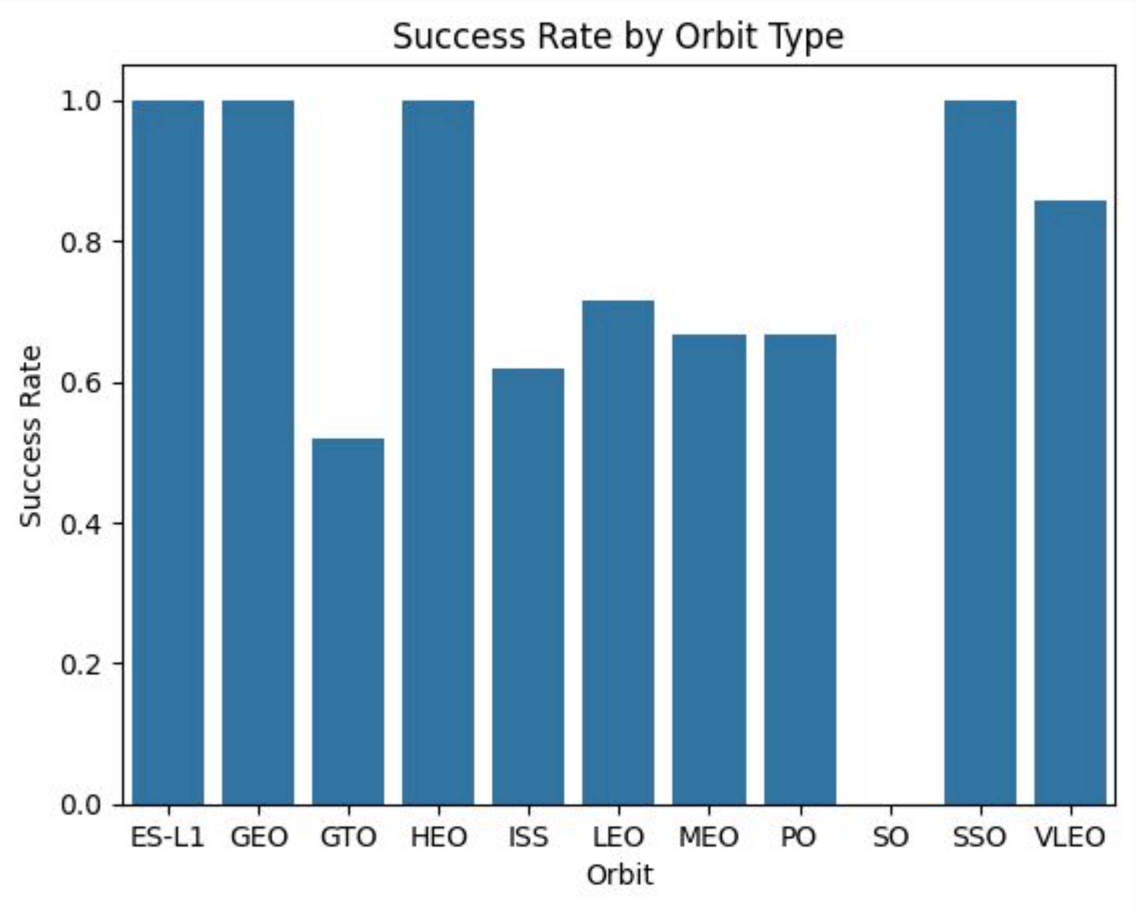


Payload vs. Launch Site



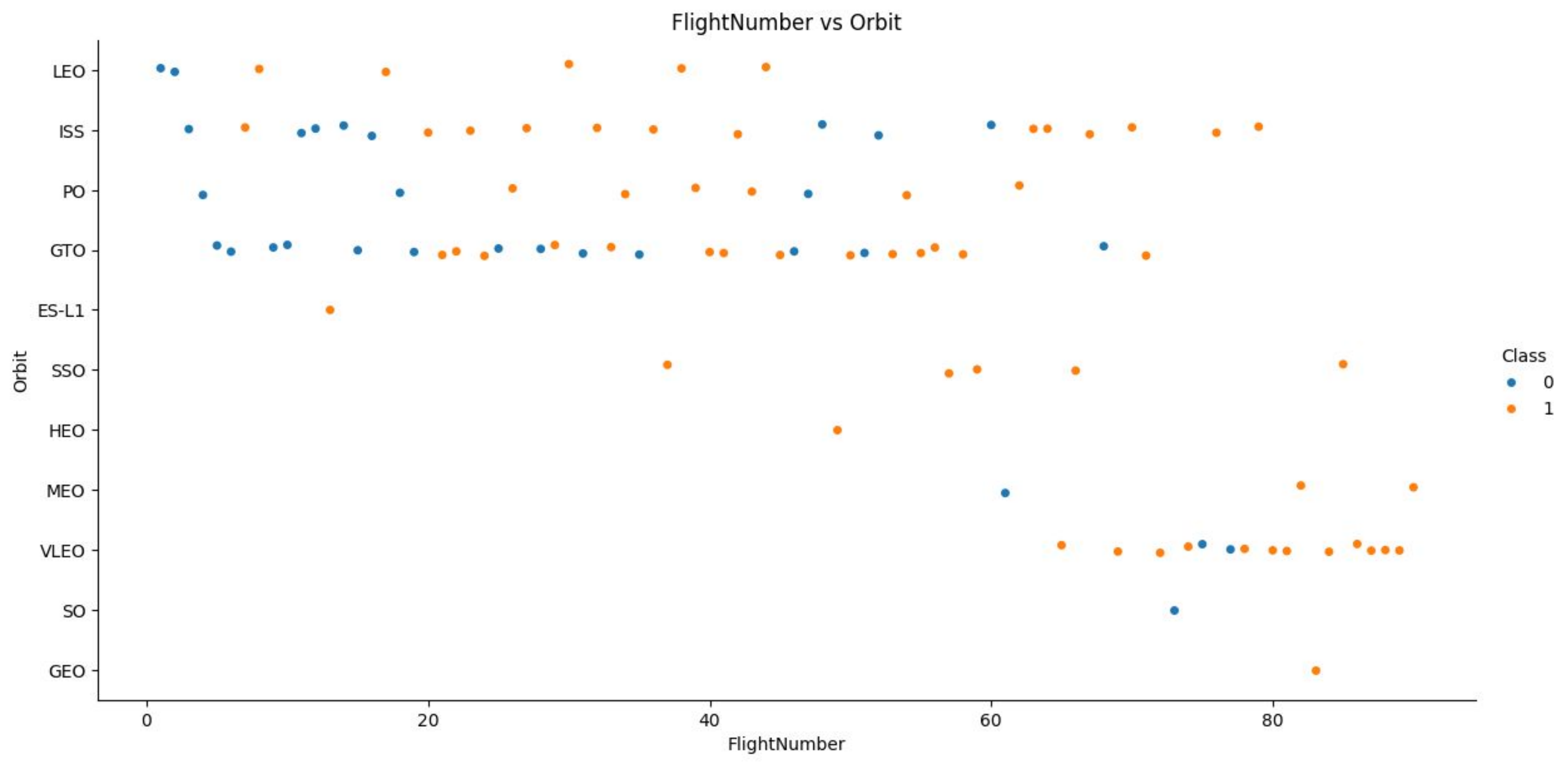
The higher payload mass, the higher successful rate for the rocket

Success Rate vs. Orbit Type



ES-L1, GEO, HEO, SSO, VLEO had the most success rate

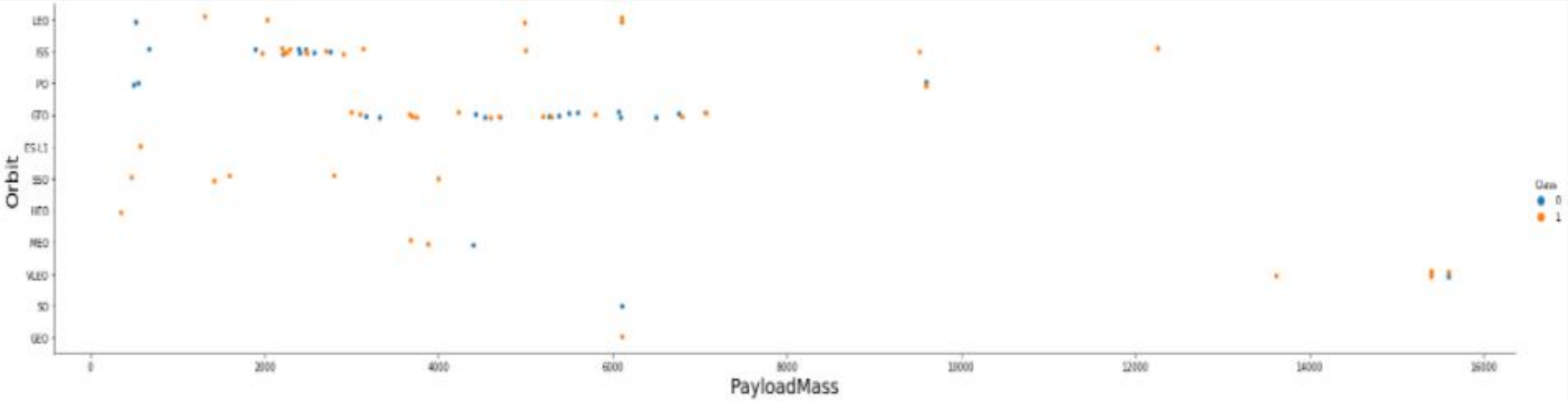
Flight Number vs. Orbit Type



We observe that in the LEO orbit, success is related to the number of flights whereas in the GTO orbit, there is no relationship between flight number and the orbit.

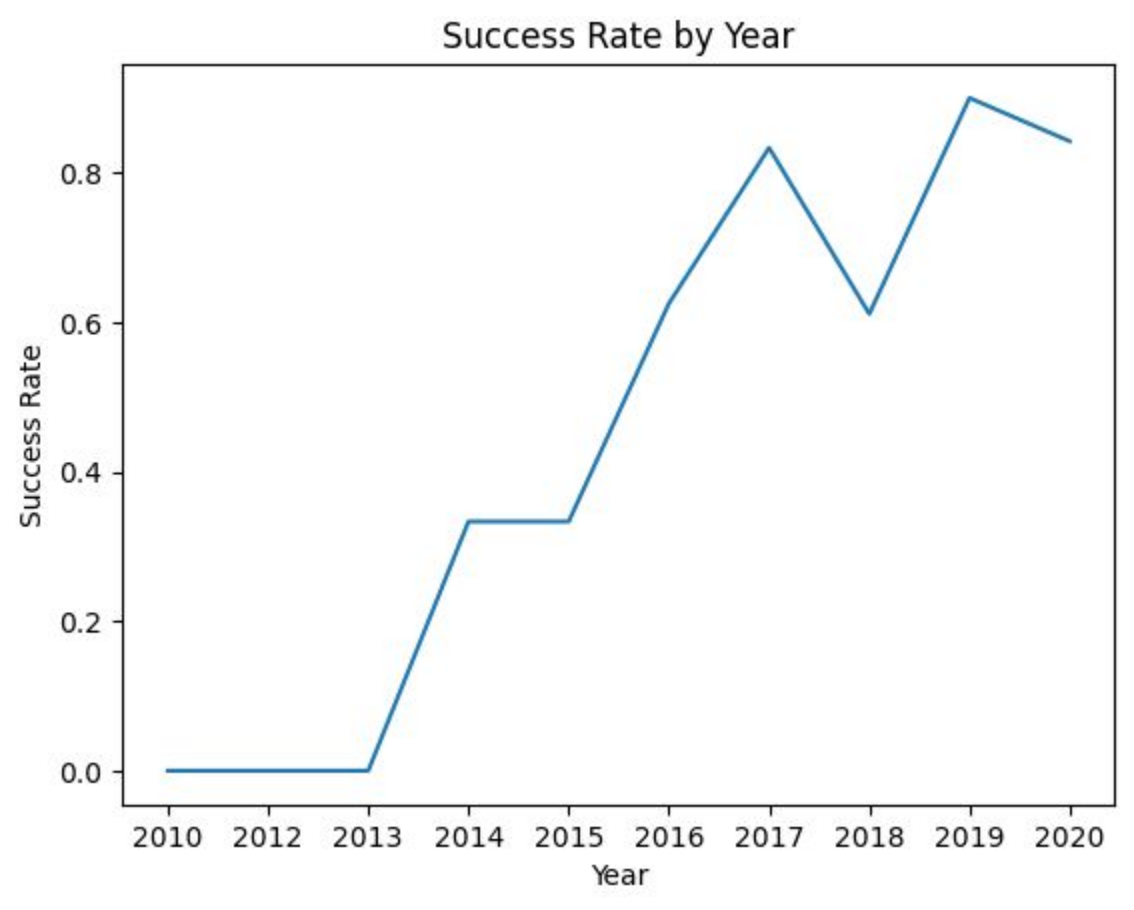
Payload vs. Orbit Type

- We can observe that with heavy payloads, the successful landing are more for PO, LEO and ISS orbits.



Launch Success Yearly Trend

From the plot, we can observe that success rate since 2013 kept on increasing till 2020.



All Launch Site Names

- Use DISTINCT to find to unique name of launch site:

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'

- LIMIT 5 records that names 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

Total Payload Mass

- We calculated the total payload carried by boosters from NASA as 45596 using the query below

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

- We calculated the total payload carried by boosters F9 v1.1 using the query below

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

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

avg(PAYLOAD_MASS__KG_)
2534.6666666666665

First Successful Ground Landing Date

- We observed that the dates of the first successful landing outcome on ground pad was 22nd December 2015

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

Hint:Use min function

```
%sql select * from SPACEXTABLE where Landing_Outcome = "Success (ground pad)" order by Date limit 1
```

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2015-12-22	1:29:00	F9 FT B1019	CCAFS LC-40	OG2 Mission 2 11 Orbcomm-OG2 satellites	2034	LEO	Orbcomm	Success	Success (ground pad)

Task 6

Successful Drone Ship Landing with Payload between 4000 and 6000

- We used the WHERE clause to filter for boosters which have successfully landed on drone ship and applied the AND condition to determine successful landing with payload mass greater than 4000 but less than 6000

Task 6

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

```
%sql select Booster_Version from SPACEXTABLE where Landing_Outcome = "Success (drone ship)" and PAYLOAD_MASS__KG_>4000 and
```

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

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

Task 7

Total Number of Successful and Failure Mission Outcomes

- We used wildcard like ‘%’ to filter for WHERE MissionOutcome was a success or a failure.

List the total number of successful and failure mission outcomes

```
countSuccess = %sql select count(*) from SPACEXTABLE WHERE Landing_Outcome like "Success%"

countFail = %sql select count(*) from SPACEXTABLE WHERE Landing_Outcome like "Failure%"

print(countSuccess)
print(countFail)
```

```
 * sqlite:///my_data1.db
Done.
 * sqlite:///my_data1.db
Done.
+----------+
| count(*) |
+----------+
|    61    |
+----------+
+----------+
| count(*) |
+----------+
|    10    |
+----------+
```

Boosters Carried Maximum Payload

- We determined the booster that have carried the maximum payload using a subquery in the WHERE clause and the MAX() function.

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

- We used a combinations of the WHERE clause, LIKE, AND, and BETWEEN conditions to filter for failed landing outcomes in drone ship, their booster versions, and launch site names for year 2015e

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.

```
%sql select substr(Date, 6,2), Landing_Outcome, Booster_Version, Launch_Site from SPACEXTABLE WHERE Landing_Outcome = "Failu
```

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

substr(Date, 6,2)	Landing_Outcome	Booster_Version	Launch_Site
01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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

- We selected Landing outcomes and the COUNT of landing outcomes from the data and used the WHERE clause to filter for landing outcomes BETWEEN 2010-06-04 to 2010-03-20. • We applied the GROUP BY clause to group the landing outcomes and the ORDER BY clause to order the grouped landing outcome 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.

```
%sql select Landing_Outcome, count(*) from SPACEXTABLE WHERE Date>='2010-06-04' and Date<='2017-03-20' Group by Landing_Out
```

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

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

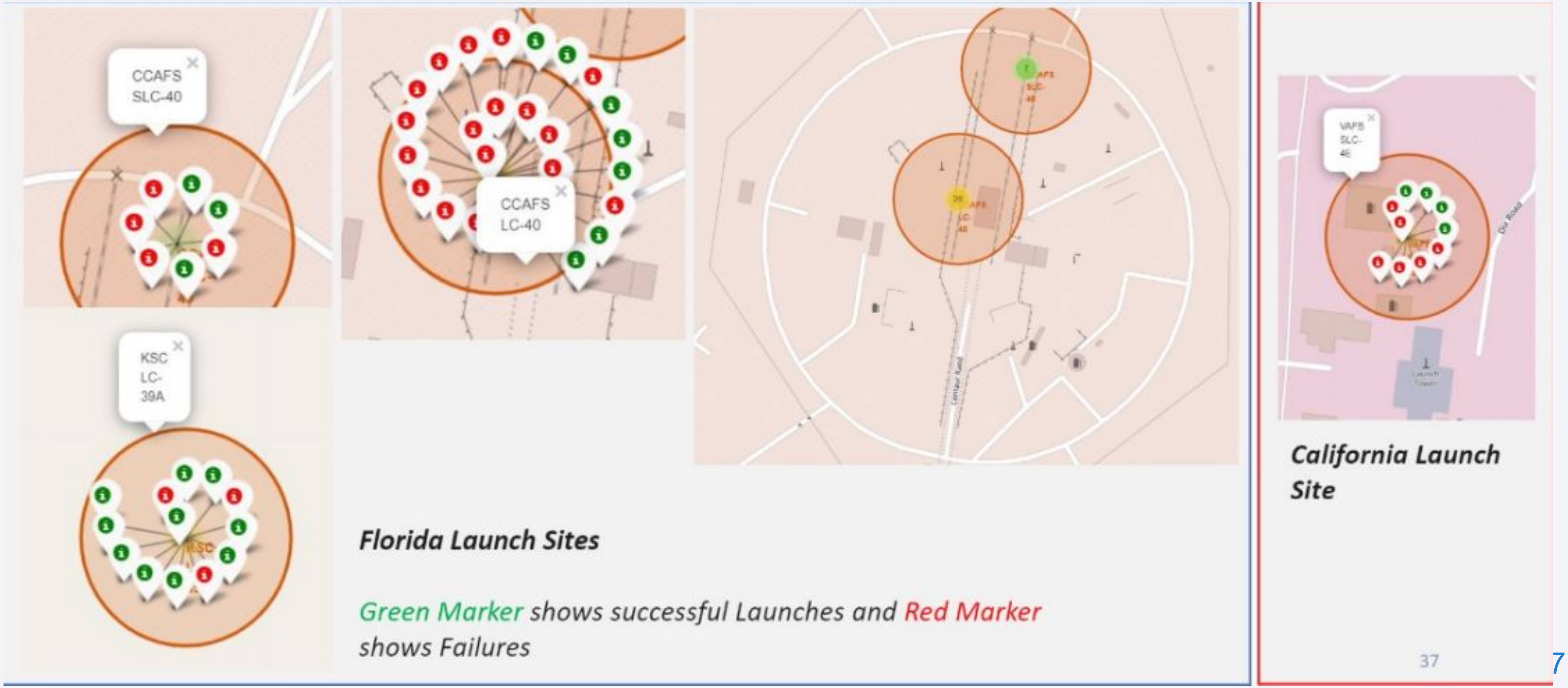
Reference Links

Section 3

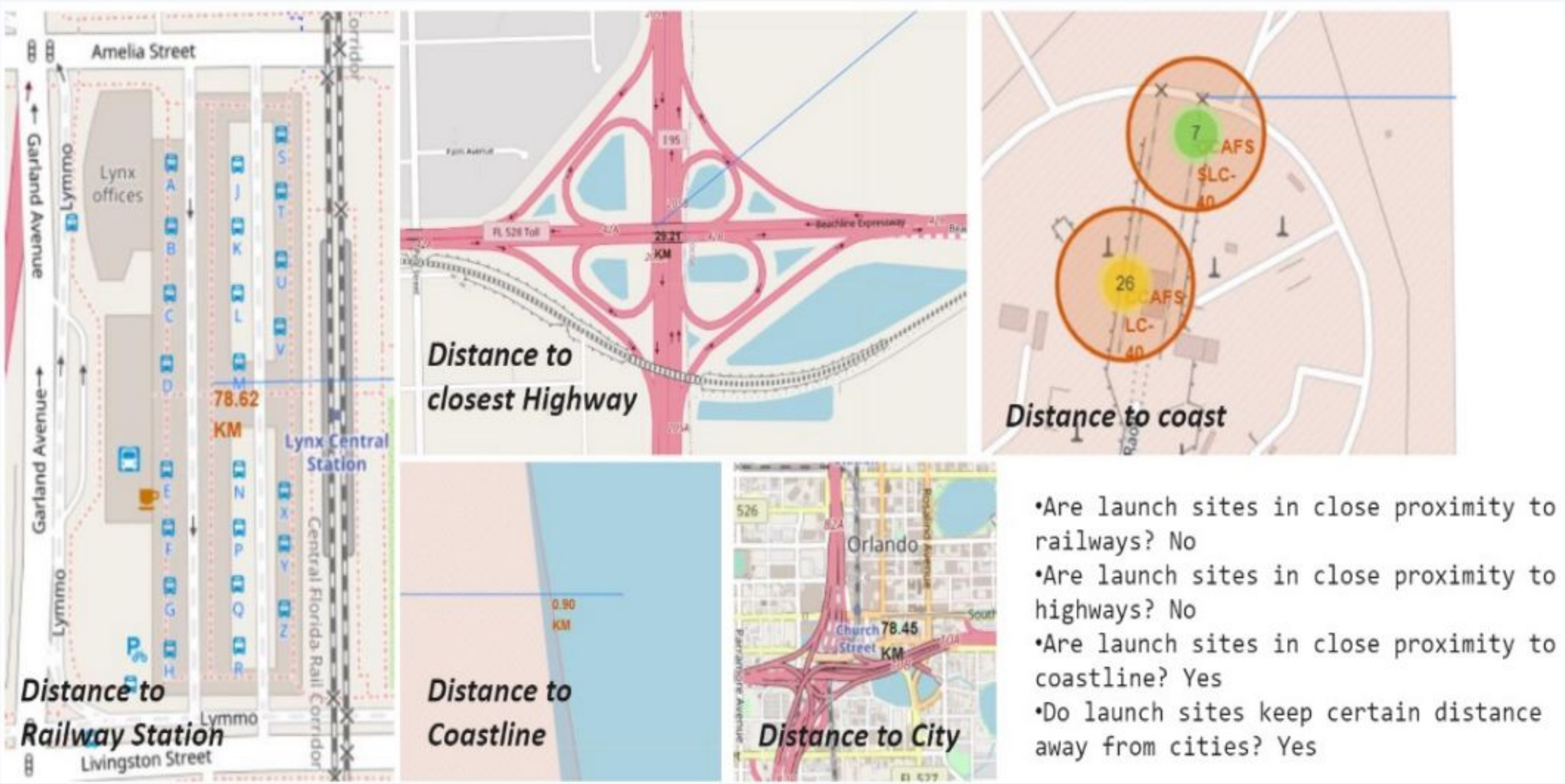
Launch Sites Proximities Analysis

All launch sites global map markers

Markers showing launch sites with color labels



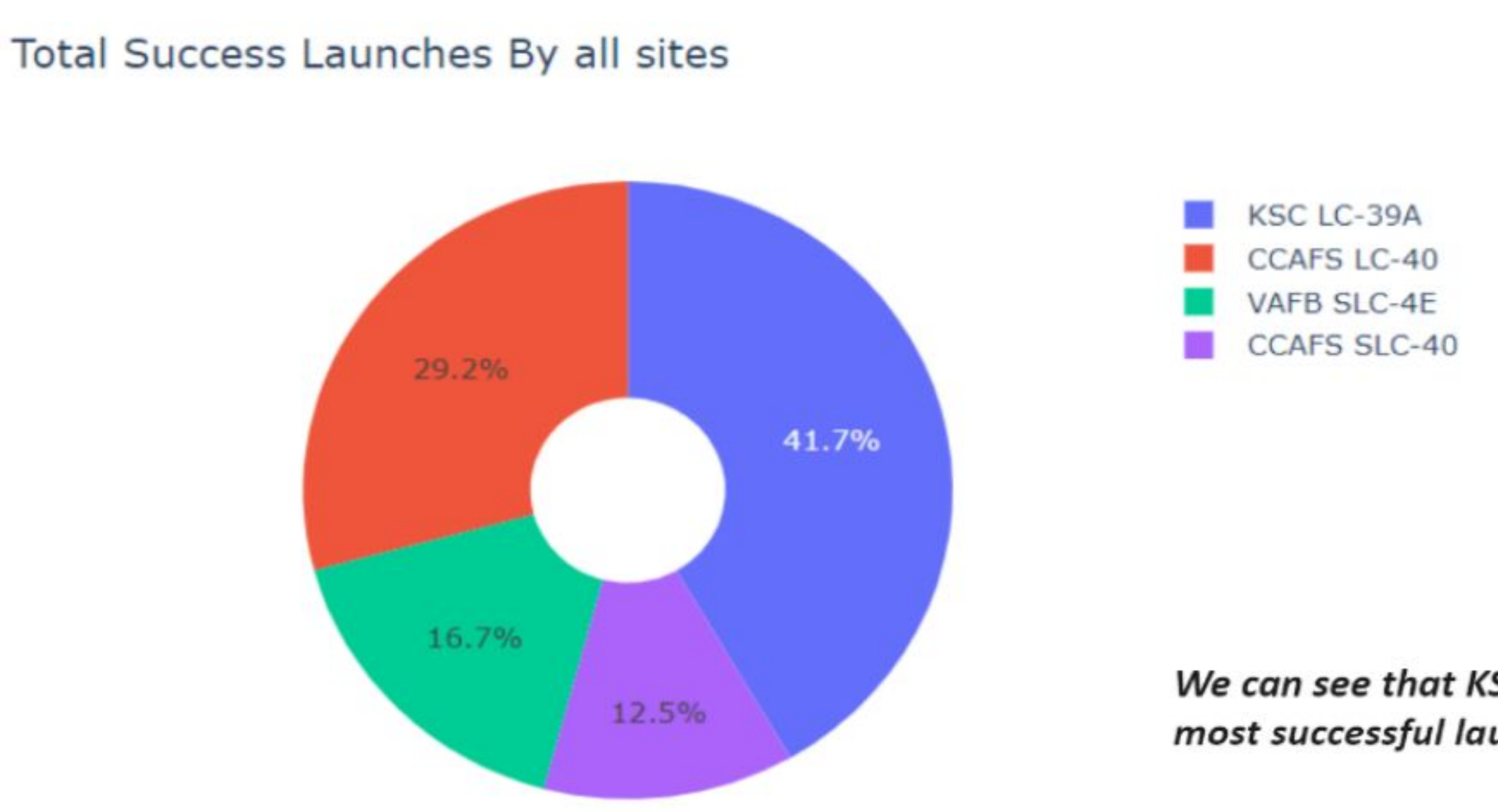
Launch Site distance to landmarks



Section 4

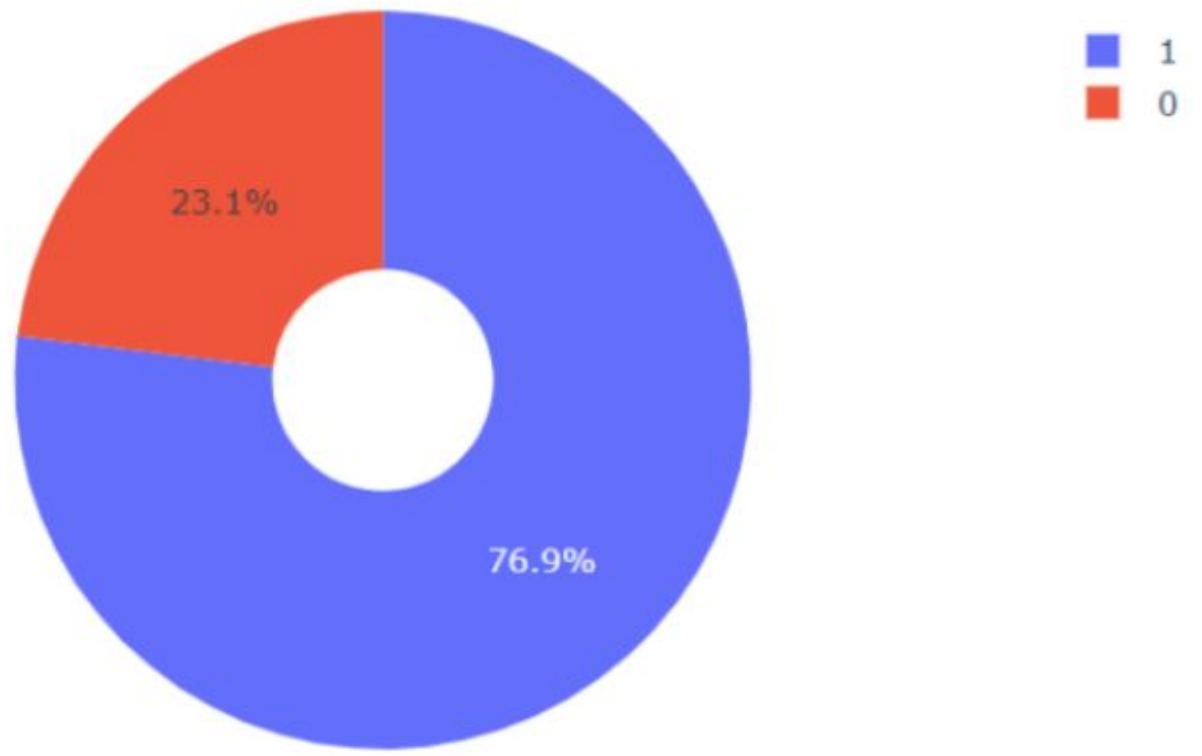
Build a Dashboard with Plotly Dash

Pie chart showing the success percentage achieved by each launch site



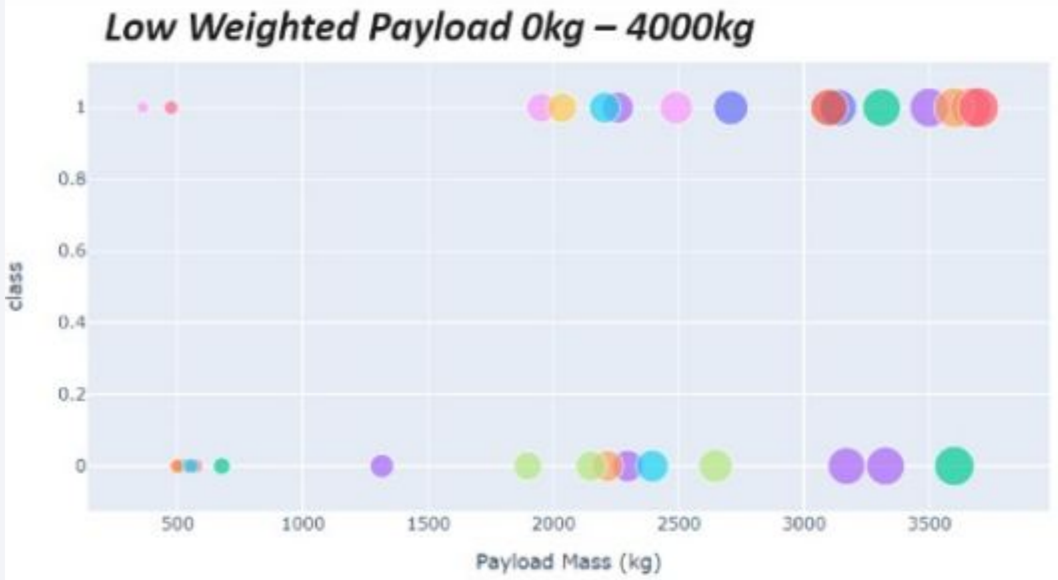
We can see that KSC LC-39A had the most successful launches from all the sites

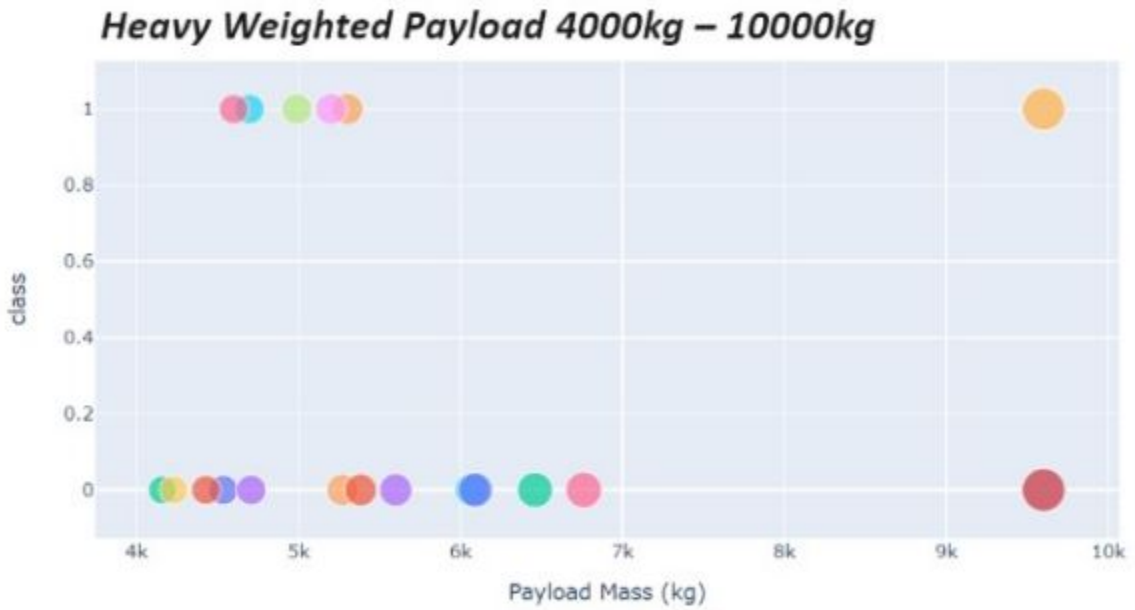
Pie chart showing the Launch site with the highest launch success ratio



KSC LC-39A achieved a 76.9% success rate while getting a 23.1% failure rate

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





We can see the success rates for low weighted payloads is higher than the heavy weighted payloads

Section 5

Predictive Analysis (Classification)

Classification Accuracy

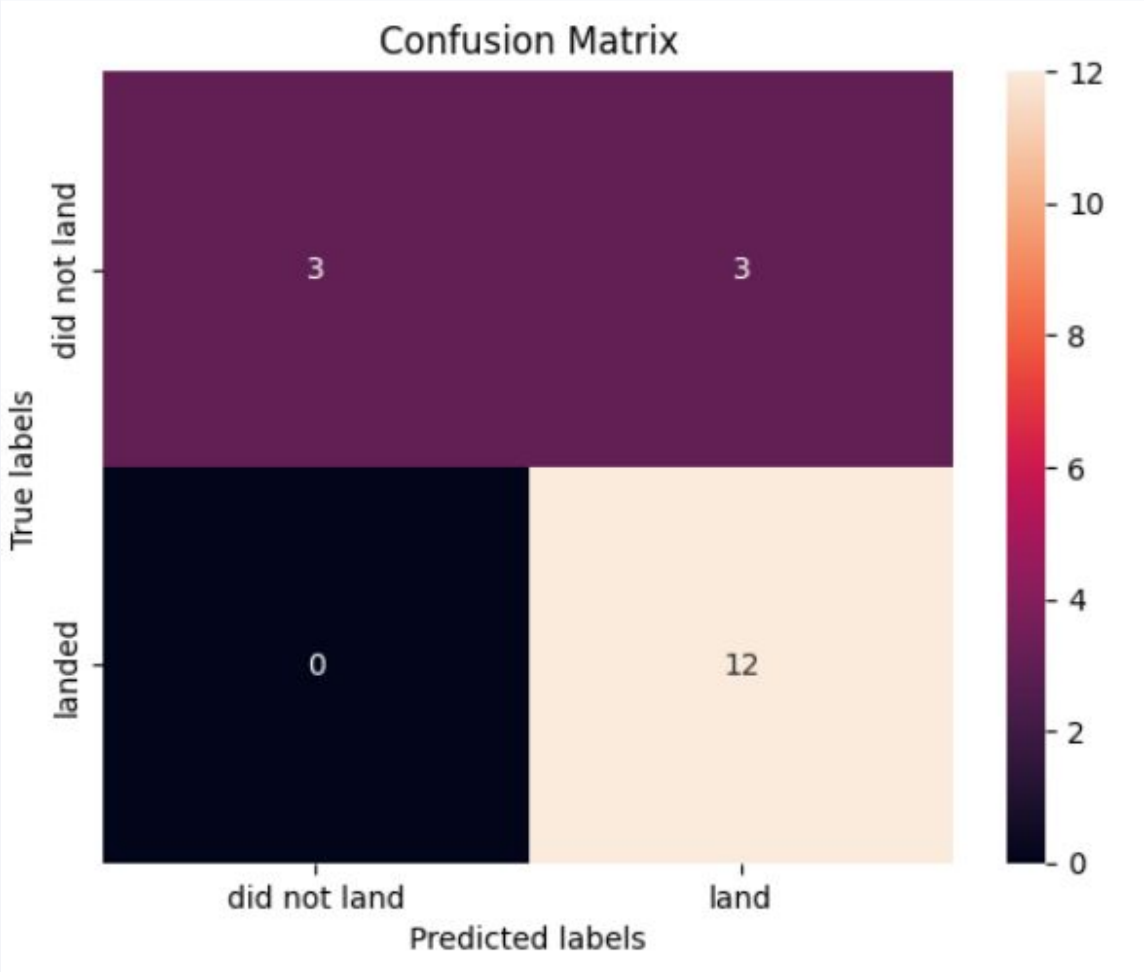
- The decision tree classifier is the model with the highest classification accuracy

```
print("tuned hpyerparameters :(best parameters) ",tree_cv.best_params_)
print("accuracy :",tree_cv.best_score_)
```

```
tuned hpyerparameters :(best parameters)  {'criterion': 'entropy', 'max_depth': 8, 'max_features': 'sqrt', 'min_samples_lea
f': 4, 'min_samples_split': 2, 'splitter': 'random'}
accuracy : 0.8767857142857143
```

Confusion Matrix

- The confusion matrix for the decision tree classifier shows that the classifier can distinguish between the different classes. The major problem is the false positives .i.e., unsuccessful landing marked as successful landing by the 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.
- Orbits ES-L1, GEO, HEO, SSO, VLEO had the most success rate.
- KSC LC-39A had the most successful launches of any sites.
- The Decision tree classifier is the best machine learning algorithm for this task.

Thank you!