IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Adam Green
09/10/2022

Outline

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

Executive Summary

Summary of methodologies

- Data Collection – API
- Data Collection – Scraping
- Data Wrangling
- EDA with data visualisation
- EDA with SQL
- Interactive Maps with Folium
- Dashboards with Ploty Dash
- Machine learning classifications

Summary of all results

- Graphical results of EDA
- Screen captures of dashboard and folium

Introduction

- We are predicting if the Falcon 9 first stage will land successfully. SpaceX has the Falcon 9 rocket launch as costing 62 million dollars where as other providers are upwards of 165 million. The saving is due to SpaceX being able to reuse the first stage. has the Falcon 9 rocket launch as costing 62 million dollars where as other providers are upwards of 165 million. The saving is due to SpaceX being able to reuse the first stage. has the Falcon 9 rocket launch as costing 62 million dollars where as other providers are upwards of 165 million. The saving is due to SpaceX being able to reuse the first stage.
- We want to predict If the first stage will land to determine the cost of a launch

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was collected using SpaceX API and web scraping a static Wikipedia URL
- Perform data wrangling
  - Data was processed by creating data frames and analysing the data types
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Split and trained data models and compared the accuracy of different parameters

Data Collection

- Collected data using get request on the SpaceX API
- Data is decoded as a normalised json file, and is converted to a pandas dataframe
- A data dictionary is created
- Data cleaning is performed, such as replacing NaN with mean values

Import Packages

Decode response

Create data dictionary

Perform data cleaning

Data Collection – SpaceX API

- Data was collected from SpaceX and then cleaned to be usable

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

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

Request data from SpaceX API

```
data = pd.json_normalize(response.json())
```

Decode the response
Turn it into a Pandas dataframe

```
launch_dict = {'FlightNumber': list(data['flight_number']),
'Date': list(data['date']),
'BoosterVersion':BoosterVersion,
'PayloadMass':PayloadMass,
'Orbit':Orbit,
'LaunchSite':LaunchSite,
'Outcome':Outcome,
'Flights':Flights,
'GridFins':GridFins,
'Reused':Reused,
'Legs':Legs,
'LandingPad':LandingPad,
'Block':Block,
'ReusedCount':ReusedCount,
'Serial':Serial,
'Longitude': Longitude,
'Latitude': Latitude}
```

```
data_launch = pd.DataFrame.from_dict(launch_dict)
```

Create a data dictionary

```
data_falcon9['PayloadMass'].replace(np.nan, PayloadMassMEAN, inplace=True)
```

```
PayloadMassMEAN = data_falcon9['PayloadMass'].mean()
```

Perform data cleaning

GitHub
URL: https://github.com/ajg3/IBM_DATA_SCIENCE_CAPSTONE/blob/main/jupyter-labs-spacex-data-collection-api%20(1).ipynb

Data Collection - Scraping

- During this section I performed web scraping to collect Falcon 9 historical launch records from a Wikipedia page

Import Packages

Request page from URL
Create a beautiful soup object

Extract all column/variable names from the HTML table header

Create a data frame by parsing the launch HTML tables

- GitHub: https://github.com/ajg3/IBM_DATA_SCIENCE_CAPSTONE/blob/main/jupyter-labs-webscraping.ipynb

Data Wrangling

- Processed data and performed analysis such as:
  - Calculate the number of launches on each site
  - Calculate the number and occurrence of each orbit
  - Calculate the number and occurrence of mission outcome per orbit type
  - Create a landing outcome label from Outcome column

GitHub: https://github.com/ajg3/IBM_DATA_SCIENCE_CAPSTONE/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

Import Packages

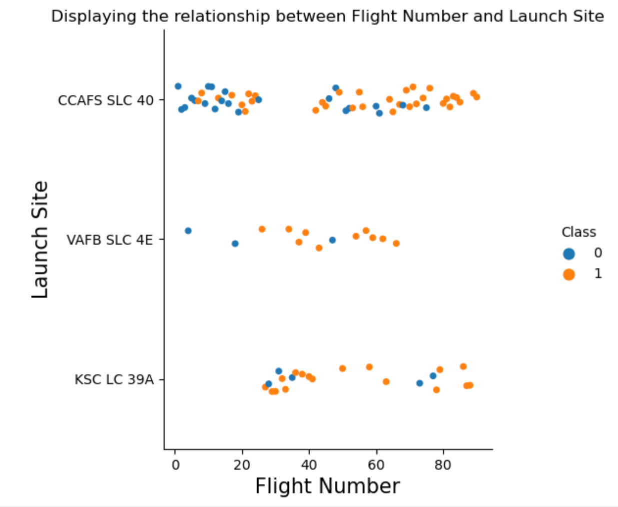
Load in datasets
Create pandas data frame

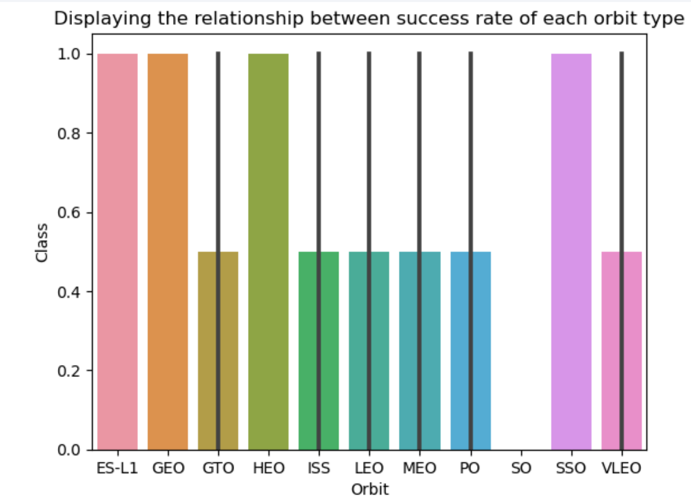
Identify data types and process data

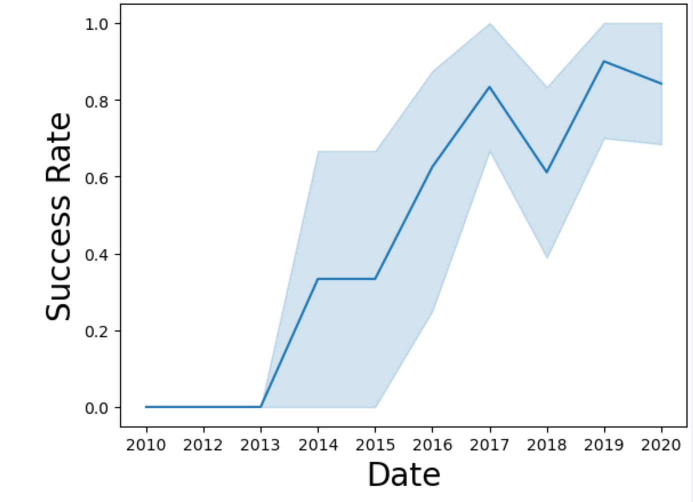
Perform Exploratory Data Analysis

EDA with Data Visualization

- Scatter graph
  - Visualize the relationship between:
    - Flight Number and Launch Site
    - Payload and Launch Site
    - Flight Number and Orbit type
    - Payload and Orbit type
- Bar graph
  - Visualize the relationship between success rate of each orbit type
- Line chart
  - Visualize the launch success yearly trend







GitHub: https://github.com/ajg3/IBM_DATA_SCIENCE_CAPSTONE/blob/main/jupyter-labs-eda-dataviz.ipynb

EDA with SQL

List of tasks performed

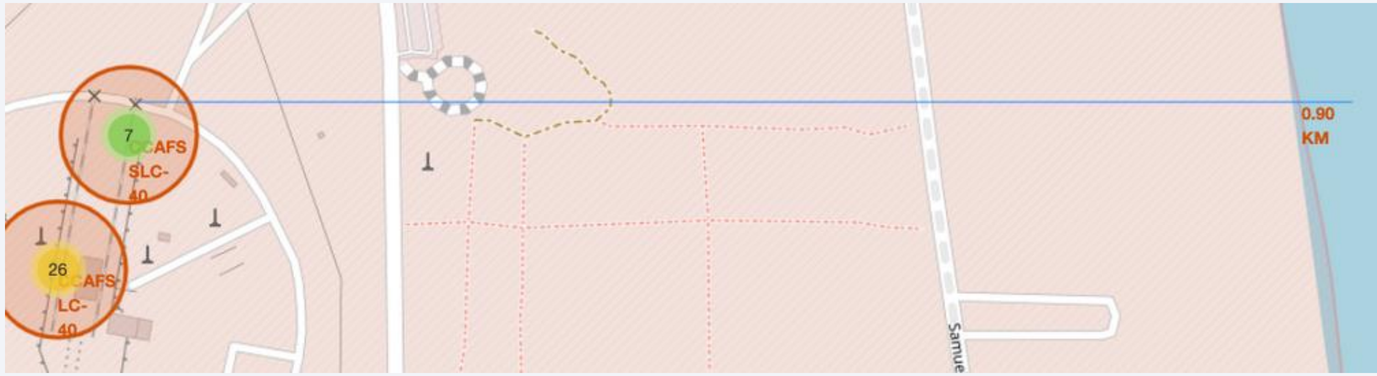
- Using bullet point format, summarize the SQL queries you performed
- Display the names of the unique launch sites in the space mission
- Display 5 records where launch sites begin with the string 'CCA'
- Display the total payload mass carried by boosters launched by NASA (CRS)
- Display average payload mass carried by booster version F9 v1.1
- List the date when the first succesful landing outcome in ground pad was achieved
- List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
- List the total number of successful and failure mission outcomes
- List the names of the booster_versions which have carried the maximum payload mass. Use a subquery
- 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.
- Rank the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.

GitHub: https://github.com/ajg3/IBM_DATA_SCIENCE_CAPSTONE/blob/main/lab_jupyter_launch_site_location(3).ipynb

Build an Interactive Map with Folium

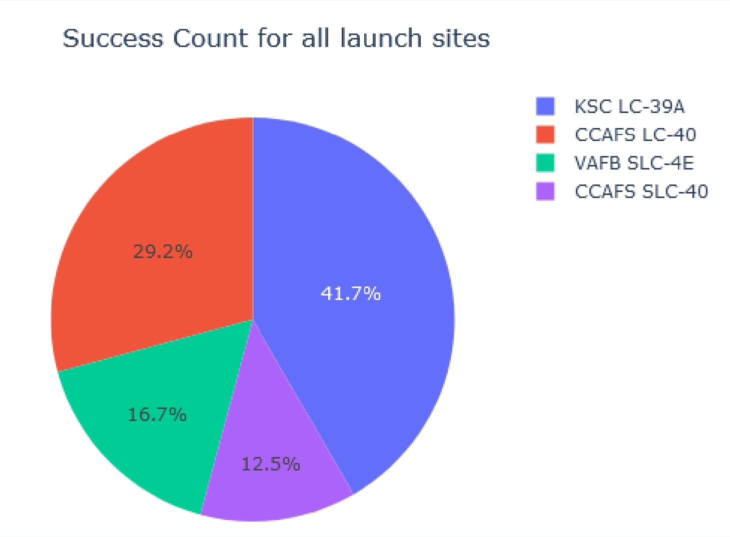
- Marked all locations on the map using clusters and markers with categories: successful launch and failed launch
- Marked the distance between a site location and airport, and site location and coastline

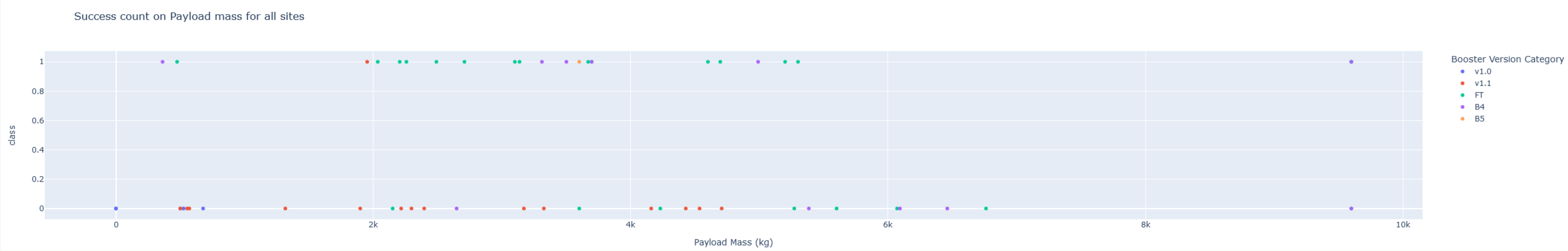
GitHub:
https://github.com/ajg3/IBM_DATA_SCIENCE_CAPSTONE/blob/main/lab_jupyter_launch_site_location(3).ipynb



Build a Dashboard with Plotly Dash

- A dashboard was created to show all launch sites in one visual, and also the option to view each launch site individually.
- These plots and visualisations allow you to find:
  - Which site has the largest successful launches?
  - Which site has the highest launch success rate?
  - Which payload range(s) has the highest launch success rate?
  - Which payload range(s) has the lowest launch success rate?
  - Which F9 Booster version has the highest launch success rate?





https://github.com/ajg3/IBM_DATA_SCIENCE_CAPSTONE/blob/main/spacex_dash_app.py

Predictive Analysis (Classification)

- The dataset was taken and split into test and train
- A confusion matrix was made and a for loop method was used to identify which paramter (knn_cv, svm_cv, logreg_cv,tree_cv) was best, and which hyperparamter (SVM, Classification Trees and Logistic Regression) was best

Import Packages

Load in datasets
Create pandas data frame

Split test and train data

Identify the best method and hyperparameter

GitHub: https://github.com/ajg3/IBM_DATA_SCIENCE_CAPSTONE/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5%20(1).ipynb

Results

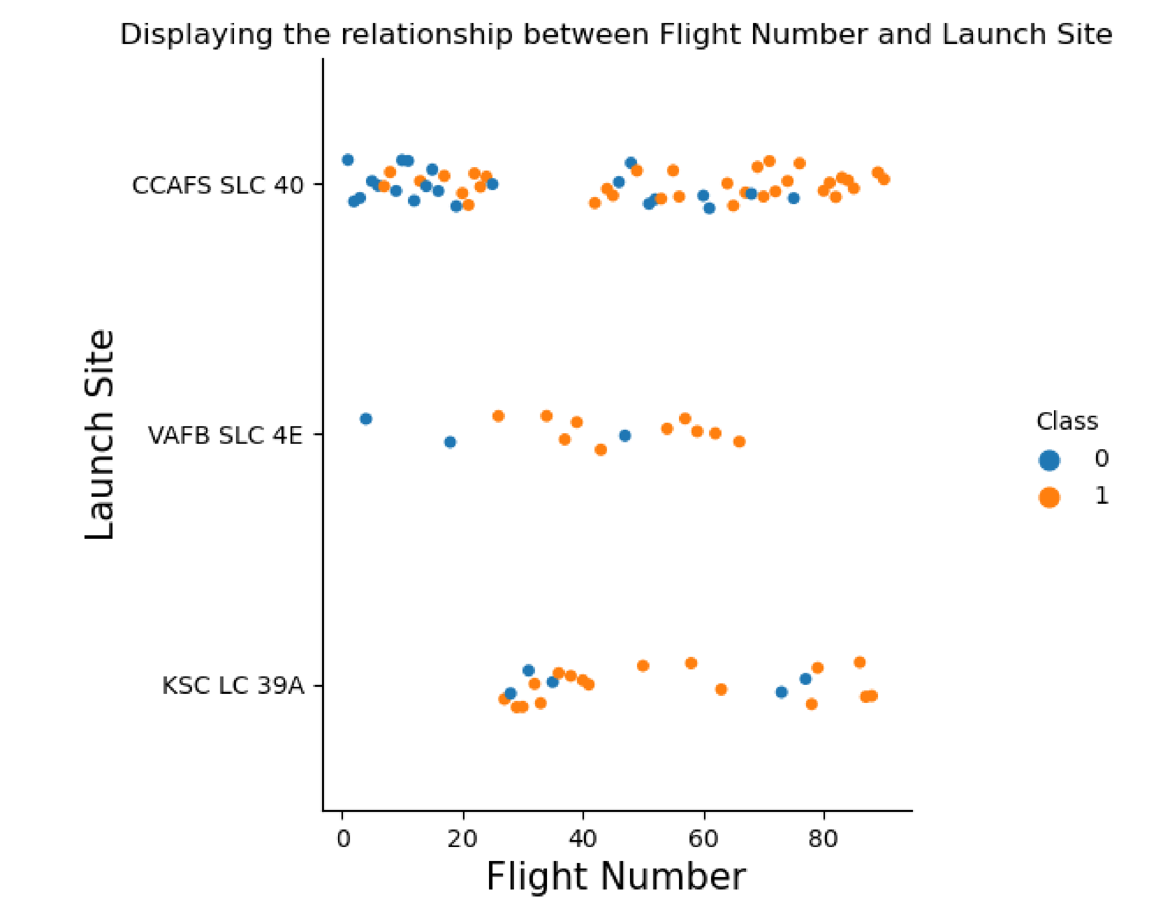
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

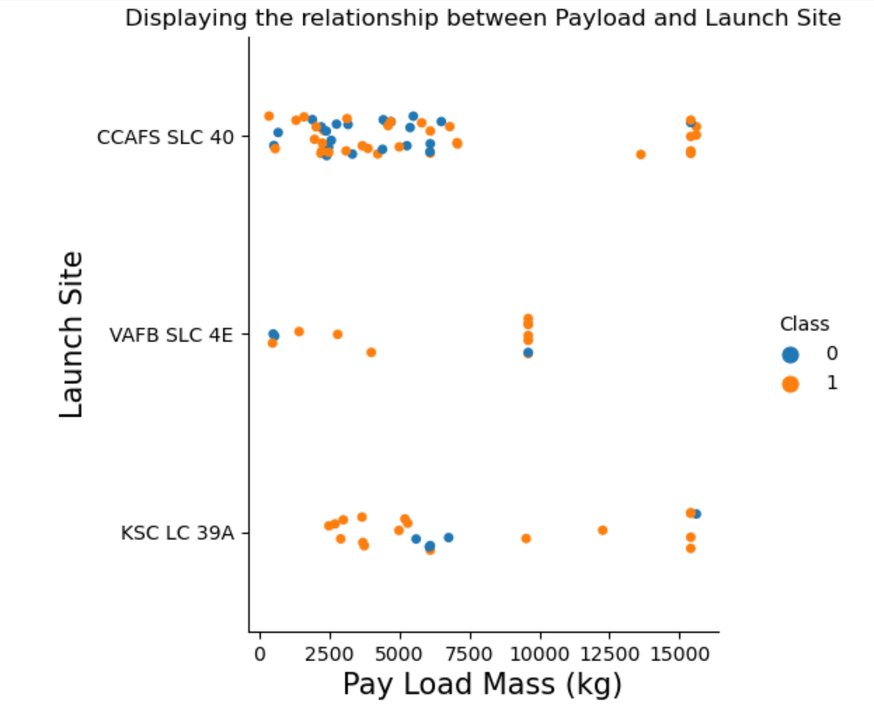
Flight Number vs. Launch Site

- You can see the launch site against the flight number with the colour indicating a fail (blue) and success (orange)



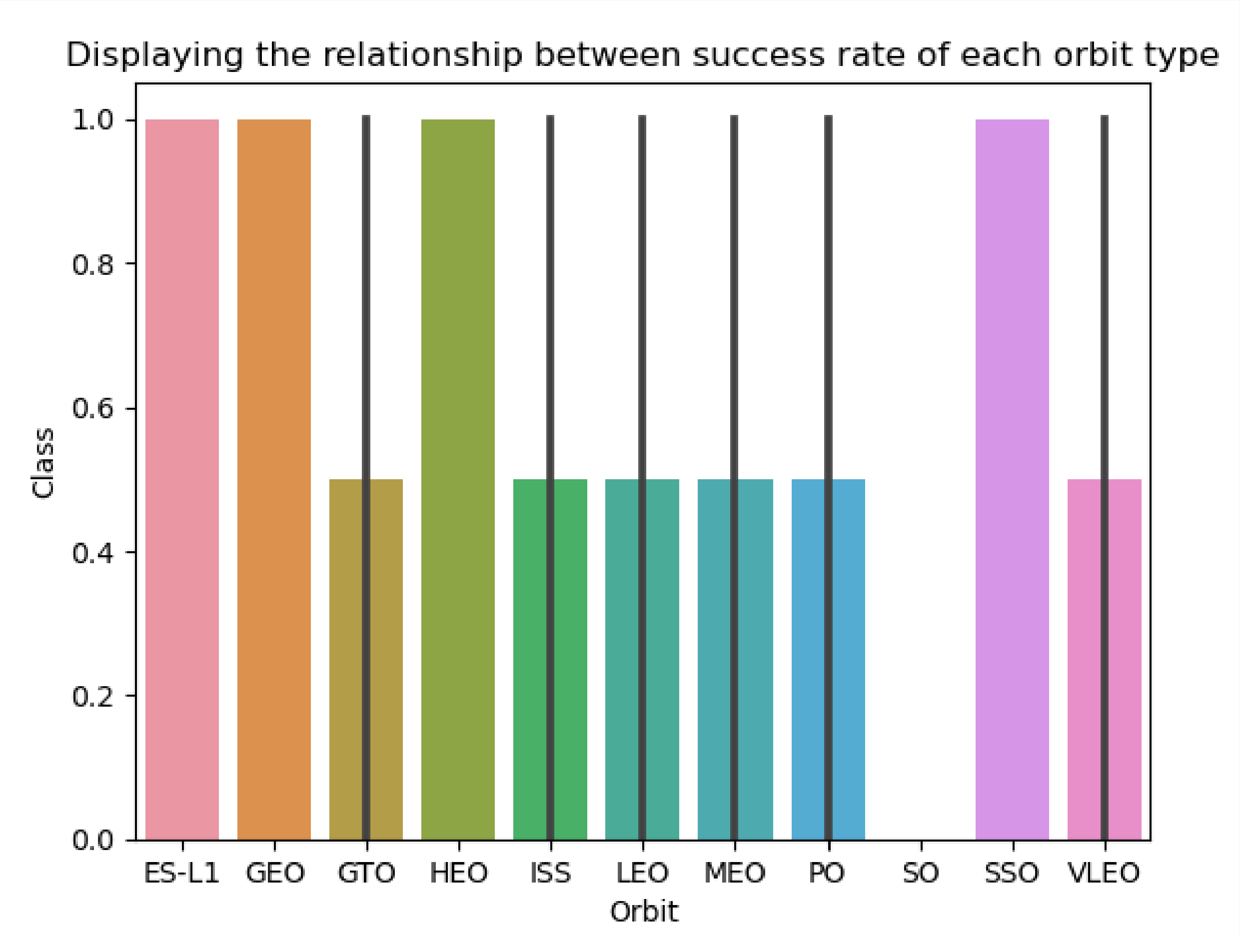
Payload vs. Launch Site

- You can see the launch site against the pay load mass with the colour indicating a fail (blue) and success (orange)



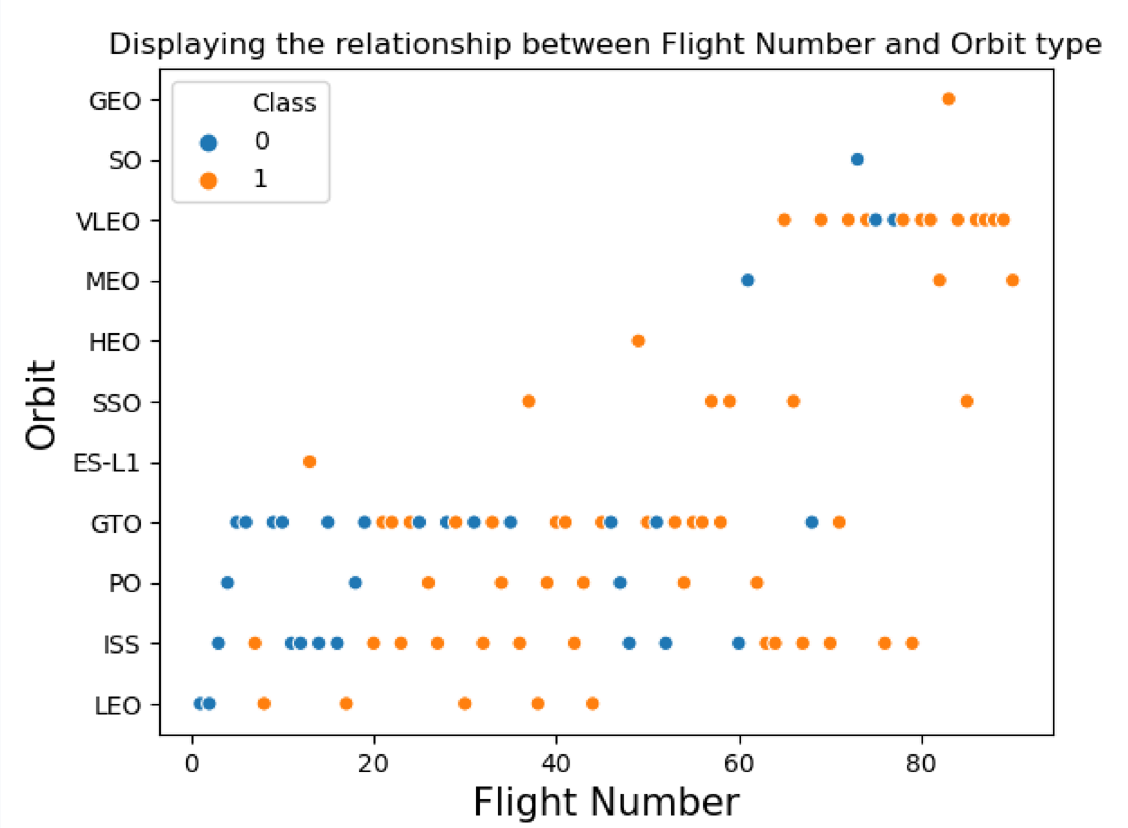
Success Rate vs. Orbit Type

- We compare the success rate of each orbit type, 1 being success and 0 being fail.



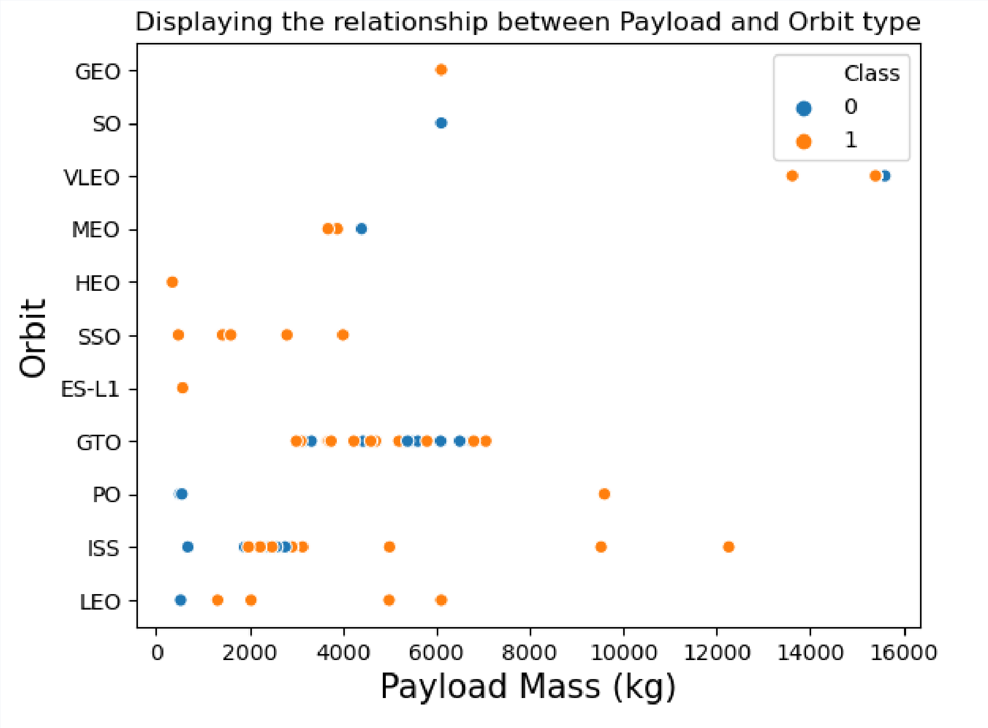
Flight Number vs. Orbit Type

- You can see the flight number against the orbit with the colour indicating a fail (blue) and success (orange)
- There is no relationship



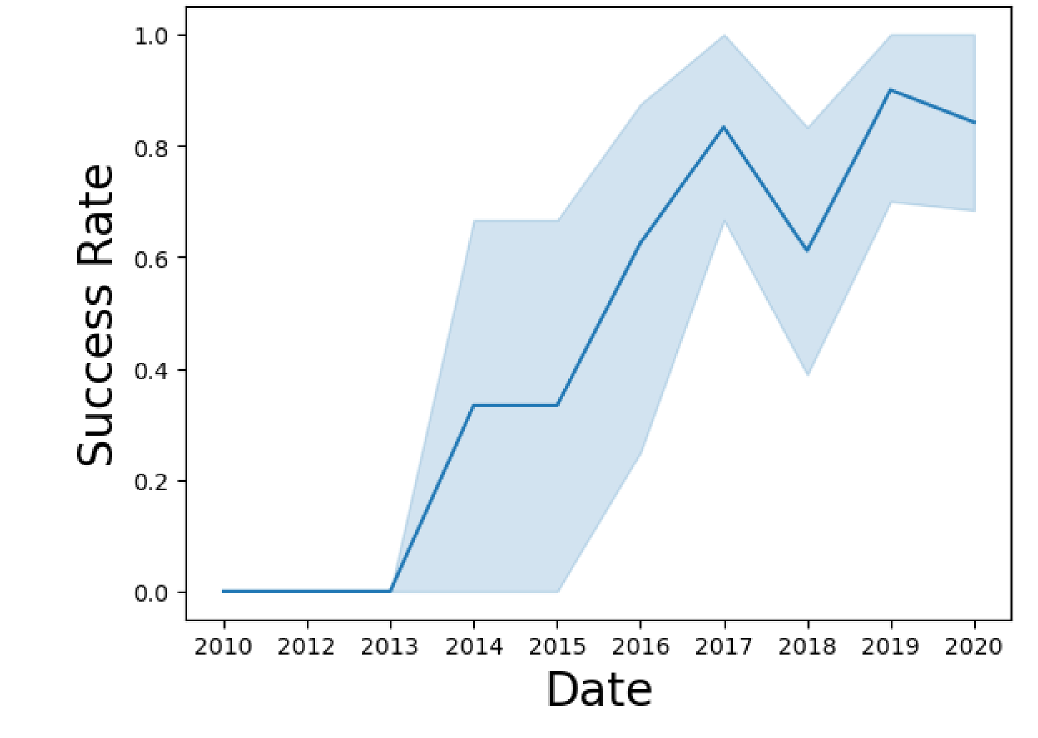
Payload vs. Orbit Type

- You can see the flight number against the orbit with the colour indicating a fail (blue) and success (orange)



Launch Success Yearly Trend

- There is a positive trend of the success rate as time progresses



All Launch Site Names

- Select Distinct is used to find unique values

```
%sql SELECT DISTINCT LAUNCH_SITE FROM SPACEXTBL
```

launch_site
CCAFS LC-40
CCAFS SLC-40
KSC LC-39A
VAFB SLC-4E

Launch Site Names Begin with 'CCA'

- LIKE '%CCA%' is used to find CCA in Launch Site
- LIMIT 5 is used to limit the output to 5

```
%sql SELECT * FROM SPACEXTBL WHERE LAUNCH_SITE LIKE '%CCA%' LIMIT 5
```

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	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	00:35:00	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

- This time we specific an exact value EG where customer = 'NASA (CRS)'

```
%sql SELECT SUM(payload_mass__kg_) AS "PAY LOAD MASS TOTAL (kg)",customer FROM SPACEXTBL where customer = 'NASA (CRS)' GROUP BY customer
```

PAY LOAD MASS TOTAL (kg)	customer
45596	NASA (CRS)

Average Payload Mass by F9 v1.1

- We used SELECT AVG(name) to get the average, and use AS "name2" to label the table
- We use the familiar where x='y' for an exact find

```
%sql SELECT AVG(payload_mass__kg_) AS "PAY LOAD MASS AVG (kg)",booster_version FROM SPACEXTBL where booster_version = 'F9 v1.1' GROUP BY booster_version
```

PAY LOAD MASS AVG (kg)	booster_version
2928	F9 v1.1

First Successful Ground Landing Date

- We use SELECT MIN(date) to find the first value, again using AS to rename the table
- We use the familiar where x='y' for an exact find

```
%sql SELECT MIN(date) AS "First successful landing date",landing__outcome FROM SPACEXTBL where landing__outcome = 'Success (ground pad)' GROUP BY landing
```

First successful landing date	landing__outcome
2015-12-22	Success (ground pad)

Successful Drone Ship Landing with Payload between 4000 and 6000

- We use the BETWEEN, AND function to specify it is between 4000 and 6000
- We use the familiar where x='y' for an exact find

```
%sql SELECT BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000 AND LANDING__OUTCOME='Success (drone ship)'
```

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

Total Number of Successful and Failure Mission Outcomes

- We use "Select count" this time to count the total of the input, eg mission_outcomes
- We use a LIKE statement to find where it is successful and fail

```
%sql SELECT count (mission_outcome) AS "Total number of successful and failure mission outcomes" FROM SPACEXTBL \
WHERE mission_outcome LIKE '%Success%' OR MISSION_OUTCOME LIKE '%Fail%'
```

Total number of successful and failure mission outcomes
101

Boosters Carried Maximum Payload

- We use SELECT DISTINCT to specify a unique value
- We use a subquery to specify the max

```
%sql SELECT DISTINCT BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL)
```

booster_version
F9 B5 B1048.4
F9 B5 B1048.5
F9 B5 B1049.4
F9 B5 B1049.5
F9 B5 B1049.7
F9 B5 B1051.3
F9 B5 B1051.4
F9 B5 B1051.6
F9 B5 B1056.4
F9 B5 B1058.3
F9 B5 B1060.2
F9 B5 B1060.3

2015 Launch Records

- We use SELECT TO_CHAR to change the date format to output word value months as opposed to 8 for August
- We specify the year is 2015

```
%sql SELECT TO_CHAR(TO_DATE(MONTH("DATE"), 'MM'), 'MONTH') AS MONTH_NAME, BOOSTER_VERSION, LAUNCH_SITE, LANDING__OUTCOME FROM SPACEXTBL \
WHERE LANDING__OUTCOME = 'Failure (drone ship)' AND YEAR(DATE) = 2015 ORDER BY MONTH(DATE)
```

month_name	booster_version	launch_site	landing__outcome
JANUARY	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
APRIL	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

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

- We use SELECT to specify which tables to index,
- COUNT to count the ranking and
- BETWEEN AND to specify the date period
- LIKE to specify Success

```
%sql SELECT LANDING__OUTCOME,COUNT(LANDING__OUTCOME) AS "Ranking" FROM SPACEXTBL WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' AND LANDING__OUTCOME LIKE 'Success%'GROUP BY
```

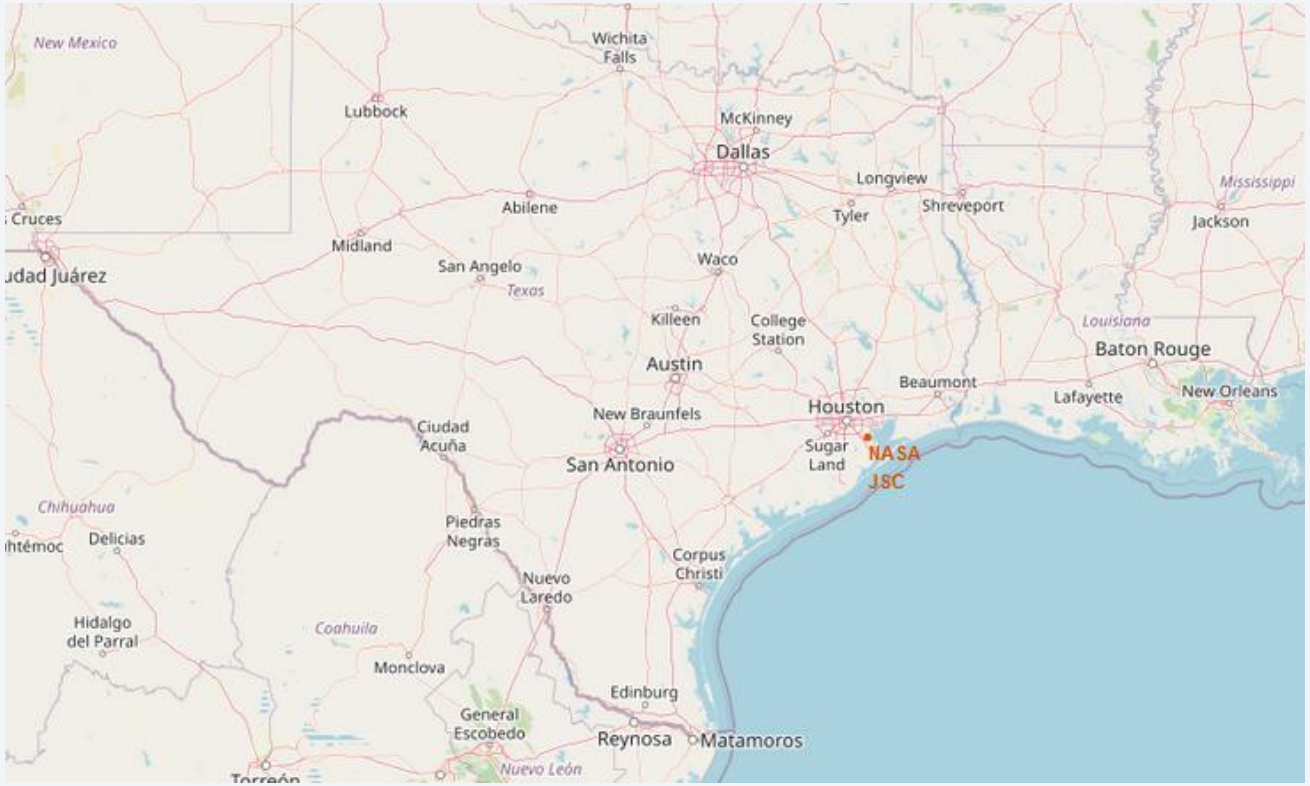
landing__outcome	Ranking
Success (drone ship)	5
Success (ground pad)	3

Section 3

Launch Sites Proximities Analysis

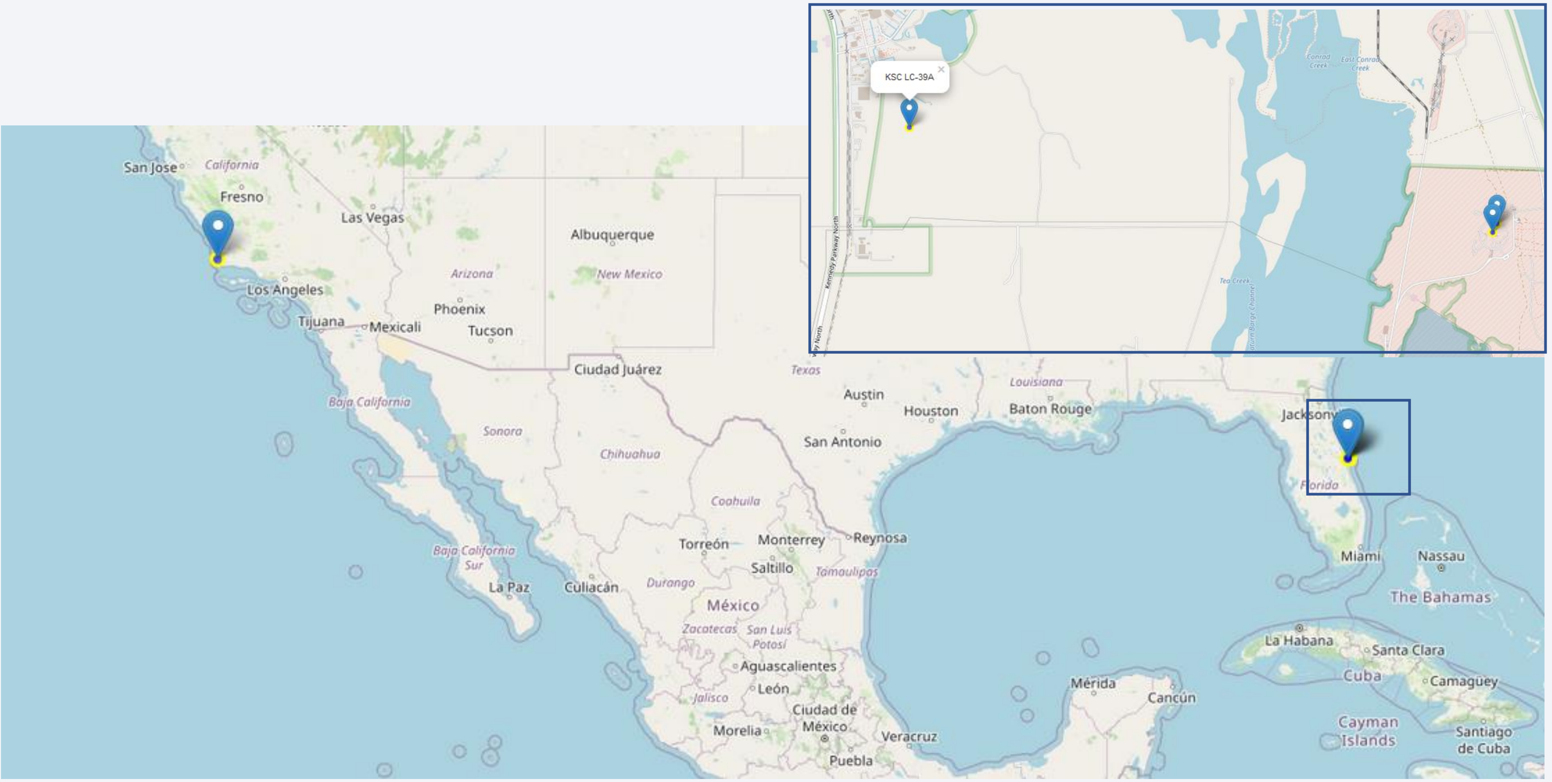
Mark All Sites

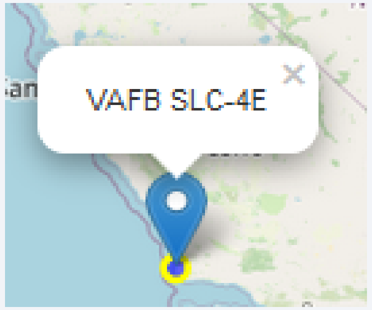
- Mark all sites on the map

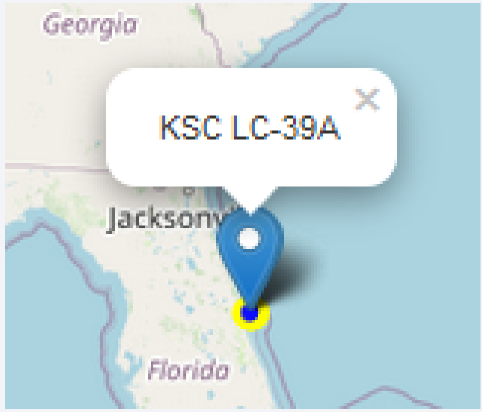


Mark successful and failed launches for each site

- Successful and failed launches marked – We are able to click to learn the name of the site

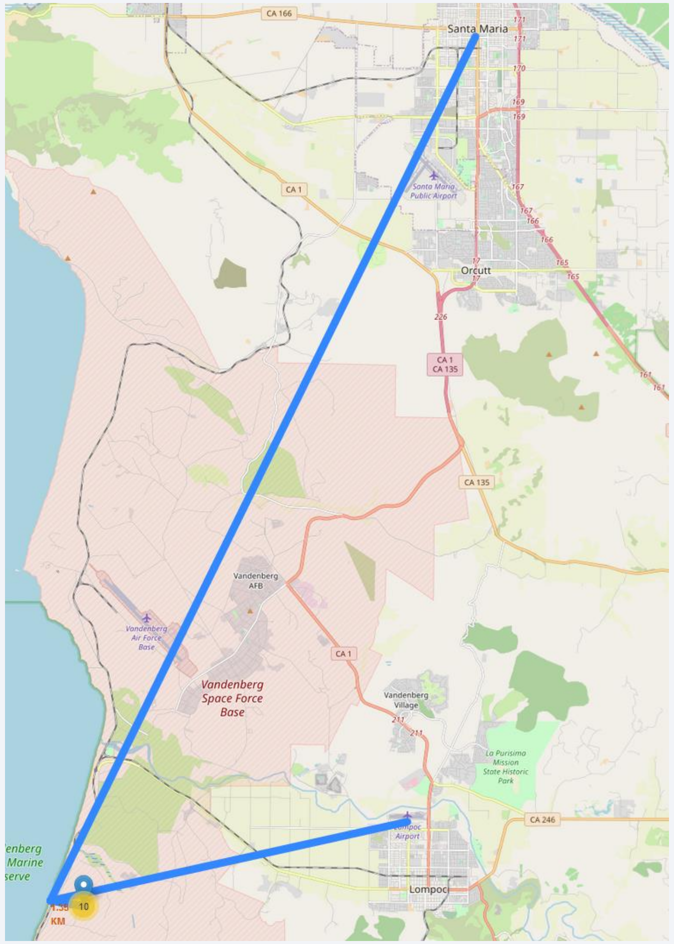






Import distance markings

- Map showing the distance from launch site to coastline to airport

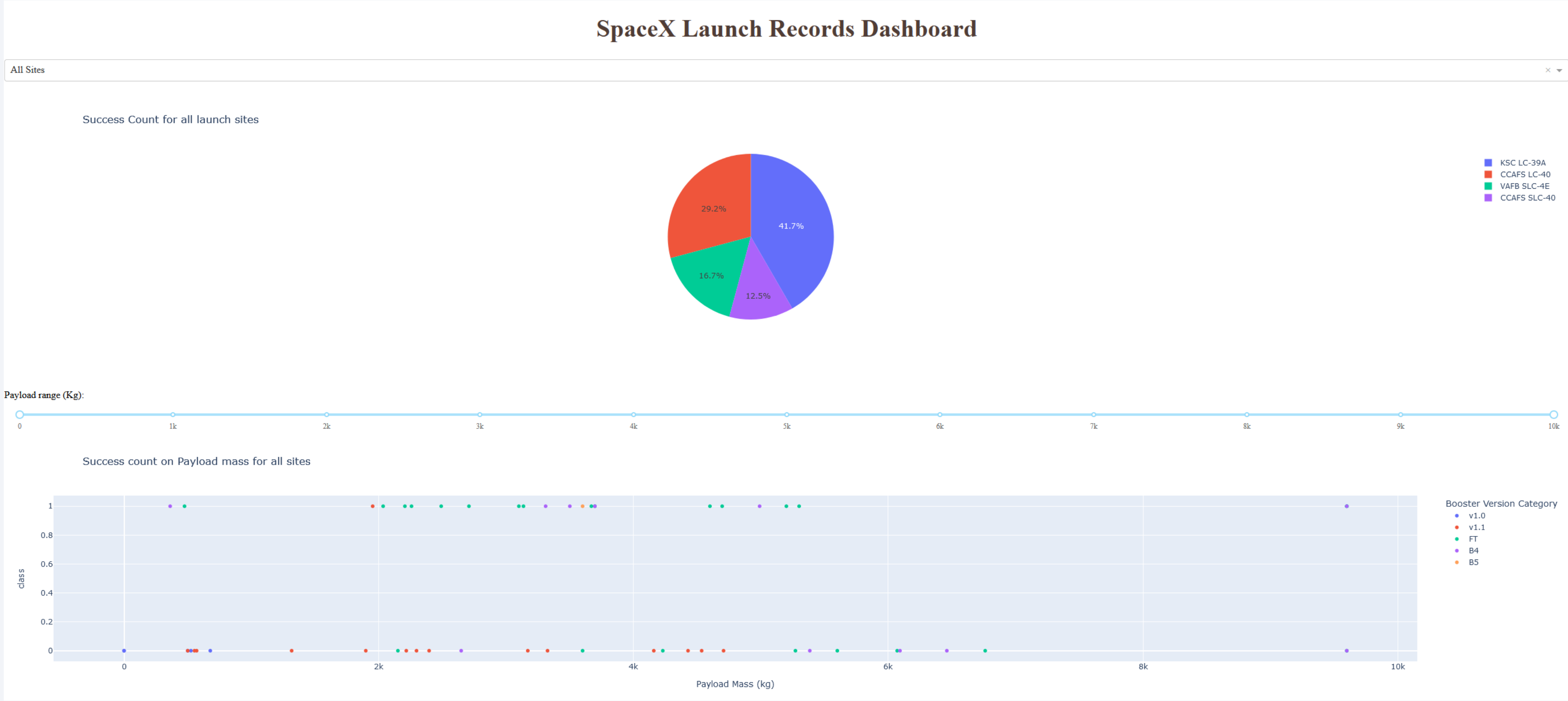


Section 4

Build a Dashboard with Plotly Dash

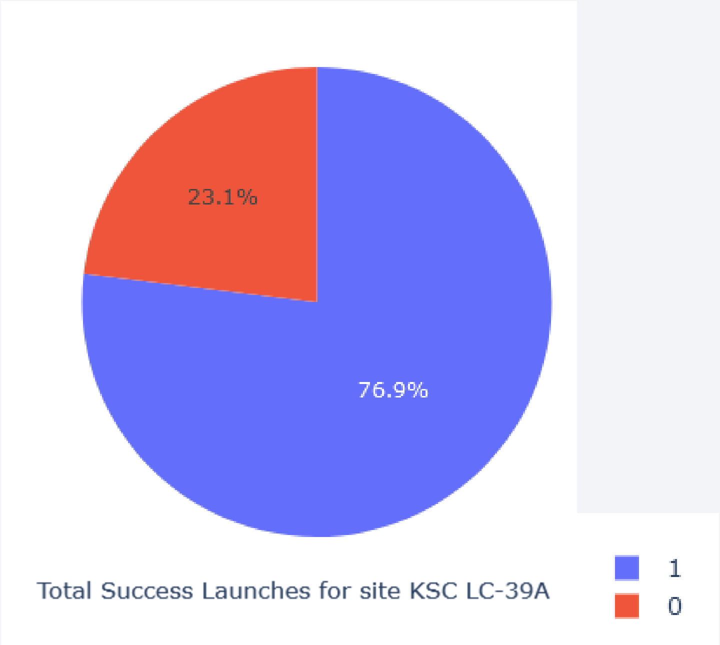
All sites dashboard

- Showing all launch sites on a pie chart success count



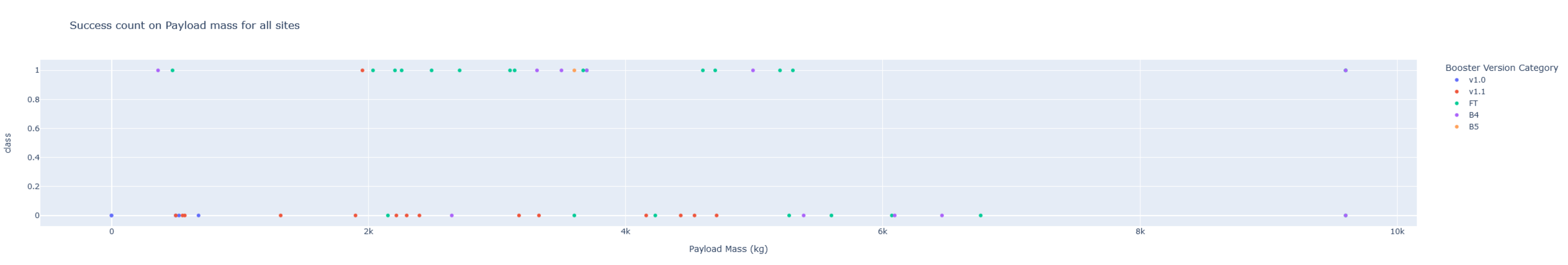
Highest launch site success ratio pie chart

- Pie chart for launch site KSC LC-39A



Payload vs Launch outcome scatter plot for all sites

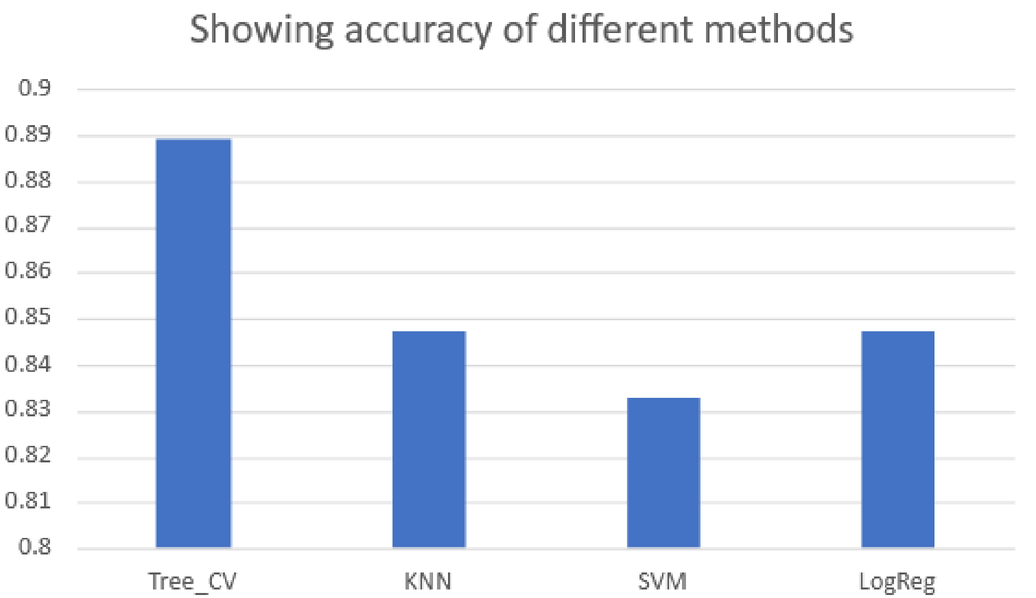
- Max payload is 9600kg which had one success and one fail
- The success rate is higher the lower the payload



Section 5

Predictive Analysis (Classification)

Classification Accuracy



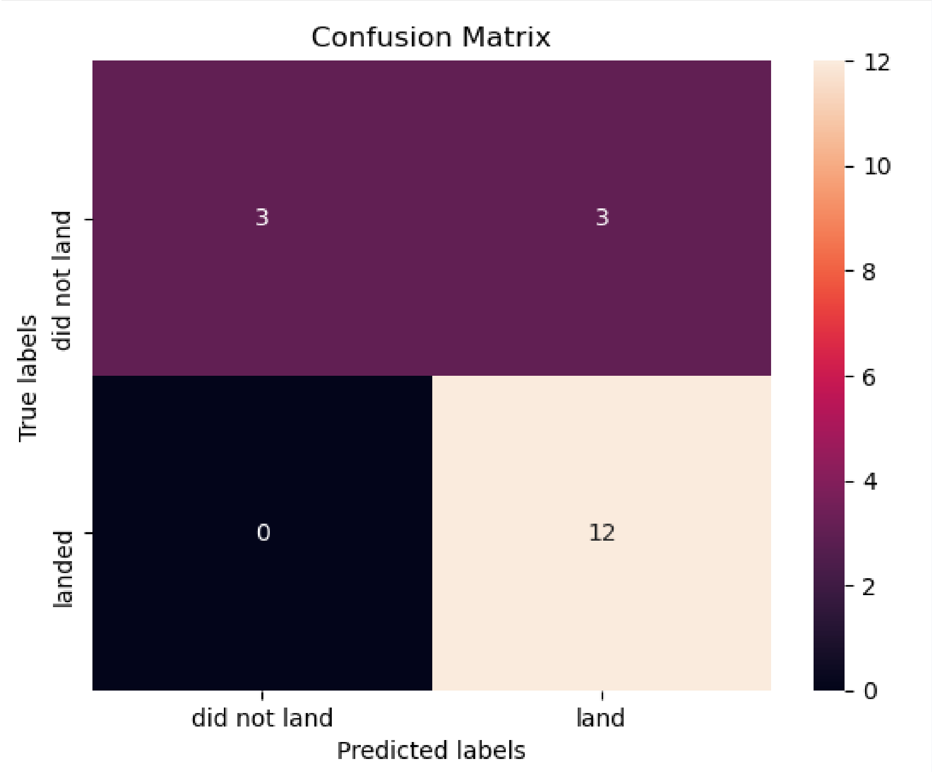

```
BestMethod=[]
for k in (knn_cv, svm_cv, logreg_cv,tree_cv):
    BestMethod.append(k.best_score_)

print( "The best method accuracy is", "{:.3f}".format(max(BestMethod)), "using Tree_CV" )
```

```
The best method accuracy is 0.889 using Tree_CV
```

Confusion Matrix

- Confusion matrix showing the true positives and negatives, and the false positives and negatives for different prediction labels



Conclusions

- The lower the payload the higher the success rate
- Success rate increases over time
- ES-L1, GEO, HEO and SSO are the most successful orbit types

Thank you!