IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Tomasz Kostuch
6/10/2021

Outline

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

Executive Summary

- The Methodologies used include:
  - Data collection
  - Data Wrangling
  - EDA with Data Visualisation
  - EDA with SQL
  - Building an Interactive Map with Folium
  - Building an Interactive Dashboard with Plotly Dash
  - Predictive Analysis
- Summary of Results:
  - EDA of Results
  - Interactive Analysis Demo with Screenshots
  - Predictive Analysis

Introduction

Project background and context

The aim of the project was to predict if the Falcon 9 first stage would land successfully. Space X advertises Falcon 9 rocket launches on its website with a cost of $62 million; other providers cost upward of $165 million each, the reason for most of this saving is that 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 also be used by alternate companies that want to bid against Space X for a rocket launch.

Key Questions

- What factors influence the rocket landing successfully?
  - The impact of the different rocket variables on success rate of the rocket landing
  - What are the optimal conditions that ensure the SpaceX rocket has the best chance of landing?

Section 1

Methodology

Methodology

1. **Data collection:**
   - Data collection was performed using:
     - Space X Rest API
     - Web Scraping from Wikipedia
2. **Data Wrangling:**
   - The data was transformed to make it suitable for Machine learning:
     - Performing one Hot encoding on data fields to express each categoric variable as a binary vector
     - Dropping irrelevant columns and dealing with missing values appropriately
3. **Exploratory Data Analysis (EDA) using Visualization and SQL**
   - EDA was performed by plotting Bar and Scatter Graphs to visualize the relationships between the rocket variables, and identify any patterns in the data.
   - SQL queries were used to drill down into the relationships between different variables.
4. **Interactive Visual Analytics using Folium and Plotly Dash**
5. **Predictive Analysis using Classification Models**
   - How to build, tune, evaluate classification models

Data Collection

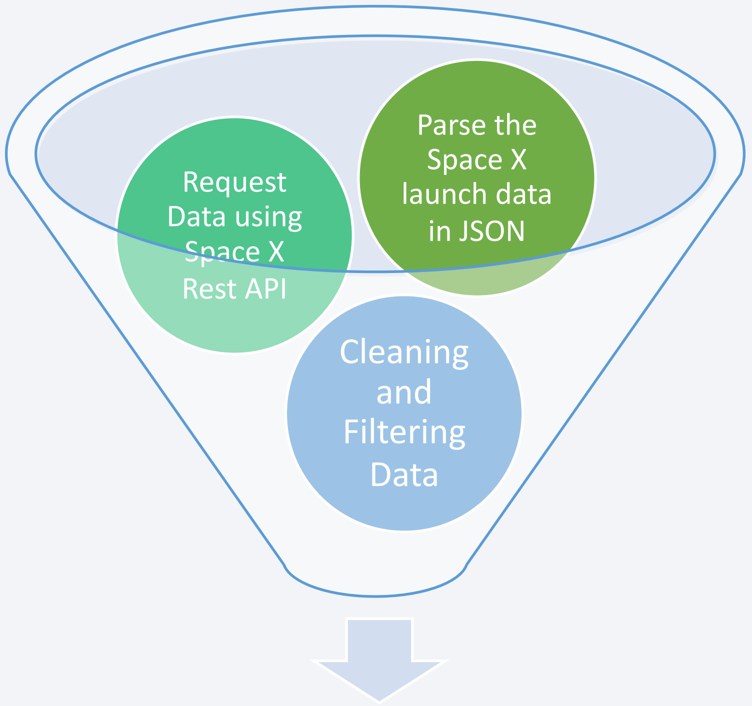
- The Space X launch data was gathered using the Space X Rest API and Web scraping of Wikipedia:
  - Space X Rest API:
    - The data contained information relating to; the type of rocket, the payload used, launch specifications, landing specifications and the outcome of the landing

Request Data using Space X Rest API → Parse the Space X launch data in JSON → Normalise the data into a flat csv file

  - Web Scraping:
    - The data obtained specifically focused on the Falcon 9 Launch and Landing Information

Request HTML response from Wikipedia → Create BeautifulSoup object from HTML response → Extract and parse relevant launch data from soup object to Dataframe

Data Collection – SpaceX API



Export Normalised data to a CSV file

[GITHUB URL to NOTEBOOK]

1. Get Response from API

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

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

2. Convert response to JSON

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

3. Clear

4. Create Data

```
getLaunchSite(data)
getPayloadData(data)
getCoreData(data)
getBoosterVersion(data)
```

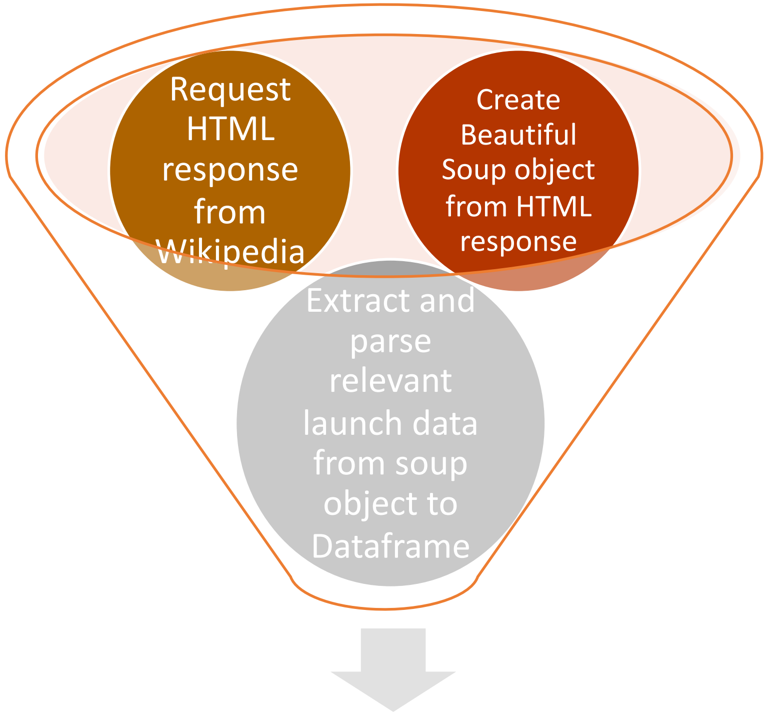
5. Filter Data

```
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}

launch_df = pd.DataFrame(launch_dict)
```

```
data_falcon9.loc[:,'FlightNumber'] = list(range(1, data_falcon9.shape[0]+1))
data_falcon9.fillna({'PayloadMass':plm_mean}, inplace=True)
data_falcon9.to_csv('dataset_part_1.csv', index=False)
```

Data Collection - Scraping



Export Normalised data to CSV File

[GITHUB URL to NOTEBOOK]

1. Get response from HTML

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

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

2. Create Beautiful Soup Object

```
soup = BeautifulSoup(response.text, 'html5lib')
```

3. Find Data using tables and obtain column names

```
html_tables= soup.find_all('table')
```

```
column_names = []
for i in (first_launch_table.find_all('th')):
    col_name = extract_column_from_header(i)
    print(col_name)
    if col_name is not None and len(col_name) > 0:
        column_names.append(col_name)
```

4. Extract Data by appending to dictionary (See in notebook by following Github URL)

```
launch_dict= dict.fromkeys(column_names)

# Remove an irrelvant column
del launch_dict['Date and time ( )']

# Let's initial the launch_dict with each value
launch_dict['Flight No.'] = []
launch_dict['Launch site'] = []
launch_dict['Payload'] = []
launch_dict['Payload mass'] = []
launch_dict['Orbit'] = []
launch_dict['Customer'] = []
launch_dict['Launch outcome'] = []
# Added some new columns
launch_dict['Version Booster']=[]
launch_dict['Booster landing']=[]
launch_dict['Date']=[]
launch_dict['Time']=[]
```

5. Create Dataframe using Dictionary

```
df = pd.DataFrame(launch_dict)
```

6. Export Dataframe to CSV file

```
df.to_csv('spacex_web_scraped.csv', index=False)
```

Data Wrangling

[GITHUB URL to NOTEBOOK]

Summary

During Data Wrangling Exploratory Data Analysis (EDA) identified patterns in the data and the necessity transforming certain variables to appropriate labels that could be used for training supervised models. For example, the dataset possessed several different cases where the booster did not land successfully. Sometimes a landing was attempted but failed due to an accident; for example, True Ocean means the mission outcome was successfully landed to a specific region of the ocean while False Ocean means the mission outcome was unsuccessfully landed to a specific region of the ocean. The main focus of the data transformation was to convert those outcomes into Training Labels with; 1 meaning the booster successfully landed, and 0 meaning it was unsuccessful.

Process

EDA on the Dataset

STAGE 1

Calculating the number of launches on every site

Calculating the occurrences of each orbit

STAGE 2

Calculating the number of different mission outcomes per orbit type

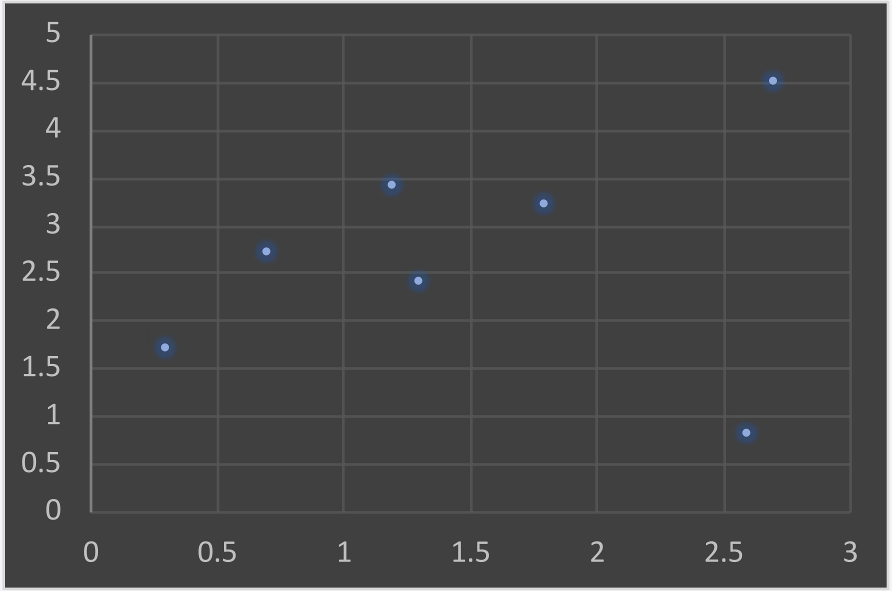
Creating landing outcome labels

Export Data as CSV

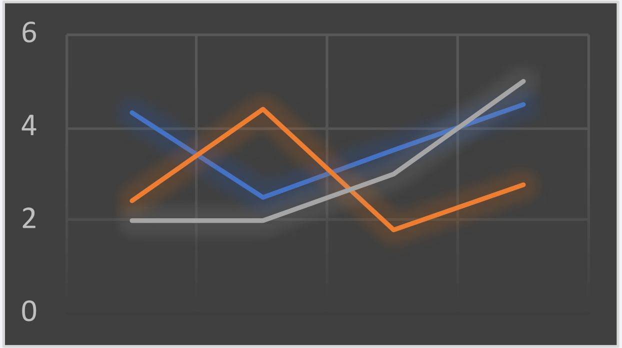
STAGE 3

EDA with Data Visualization

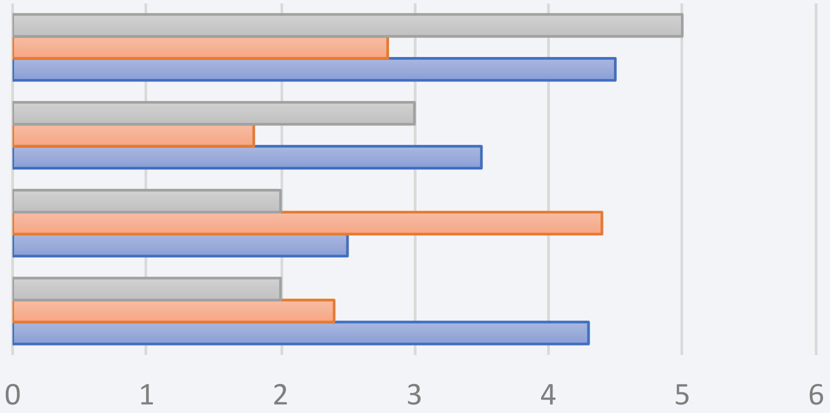
Several scatter graphs were plotted to visualise the correlation between the different variables, and further explore the relationship between them



A line graph was used to visualise the trend in the success rate of launching rockets over time



A bar graph was used to explore how the success rate changed with different orbit types.



[GITHUB URL to NOTEBOOK]()

EDA with SQL

- Various SQL queries were performed to gather information from the dataset
- In certain cases, the queries were used to drill down into the data, a comprehensive list of the queries that were performed is located below:
    - ***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 successful 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 failed landing outcomes in drone ship, their booster versions, and launch site names for in year 2015***
    - ***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***

GITHUB URL to NOTEBOOK

Build an Interactive Map with Folium

Visualising the Launch Data on an Interactive Map

The longitude and latitude was taken at each site and used with the *Circle object* to add a Circle Marker for each launch site to the map.

Assigning Launch Outcomes to Classes

Failures and Successes were mapped to 0 and 1 respectively, those classes were then used to colour the markers that were added to the interactive map by using a *MarkerCluster()*

Calculating the Distance from Various Landmarks to the Launch Sites

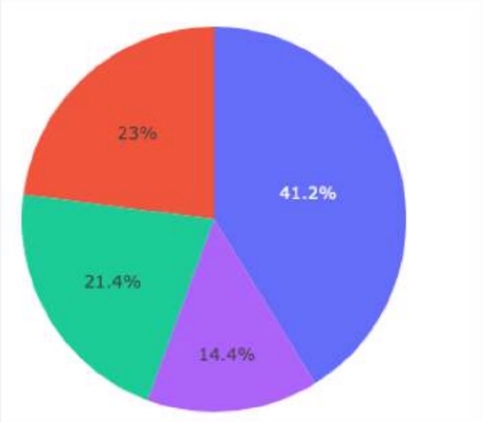
These distances were used to explore potential trends between the proximity of launch sites to landmarks such as; cities, railways or highways. Lines were drawn on the map to visualise the distance. The trends explored are detailed in the results.

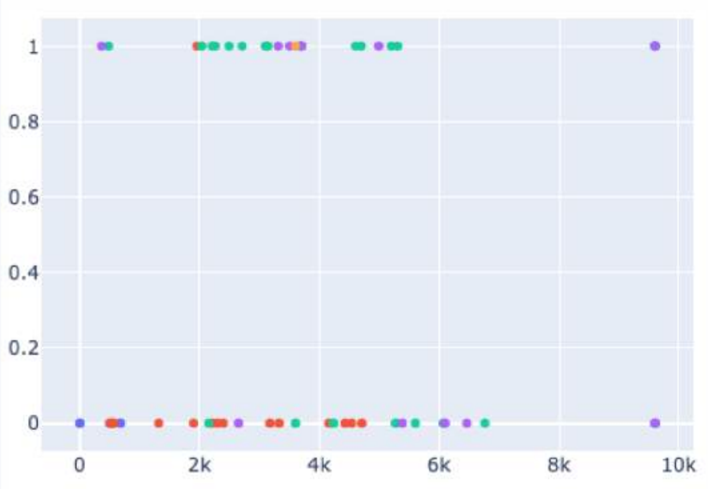
[GITHUB URL to NOTEBOOK]()

Build a Dashboard with Plotly Dash

- Dash was used to create an interactive dashboard containing a variety of plots exploring the relationship between; landing outcome, number of launches from the sites, booster version, and Payload Mass.
- **Graphs**
- A Pie chart is located at the top of the dashboard visualising the breakdown of the number of launches from each or every site.
- A scatter graph is located at the bottom of the dashboard displaying the relationship between landing outcome and payload mass for different booster versions. The range of the payload mass can be altered to explore how that particular variable affects landing outcome.

GITHUB URL to DASH SCRIPT





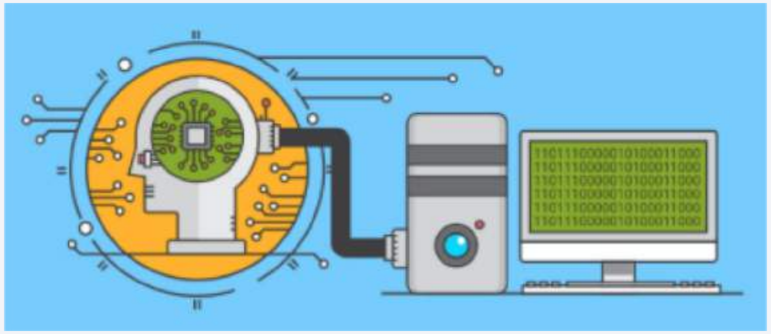
Predictive Analysis (Classification)

Building Model
- Loading Dataset into NumPy and Pandas
- Transforming Data
- Splitting data into training and test sets
- Select Machine Learning (ML) algorithm to use
- Set parameters to tune in GridSearchCV
- Fit ML algorithms and train them on training data

Evaluating Model
- Calculate accuracy of each ML Model
- Identify best parameters in GridSearchCV for each algorithm
- Plot Confusion Matrix

Improving Model
- Engineer features and tune algorithm

Finding Best Performing Model
- Calculate the F1 and Jaccard scores for each model
- Create table and graph comparing the metrics of each model



[GITHUB URL to NOTEBOOK]

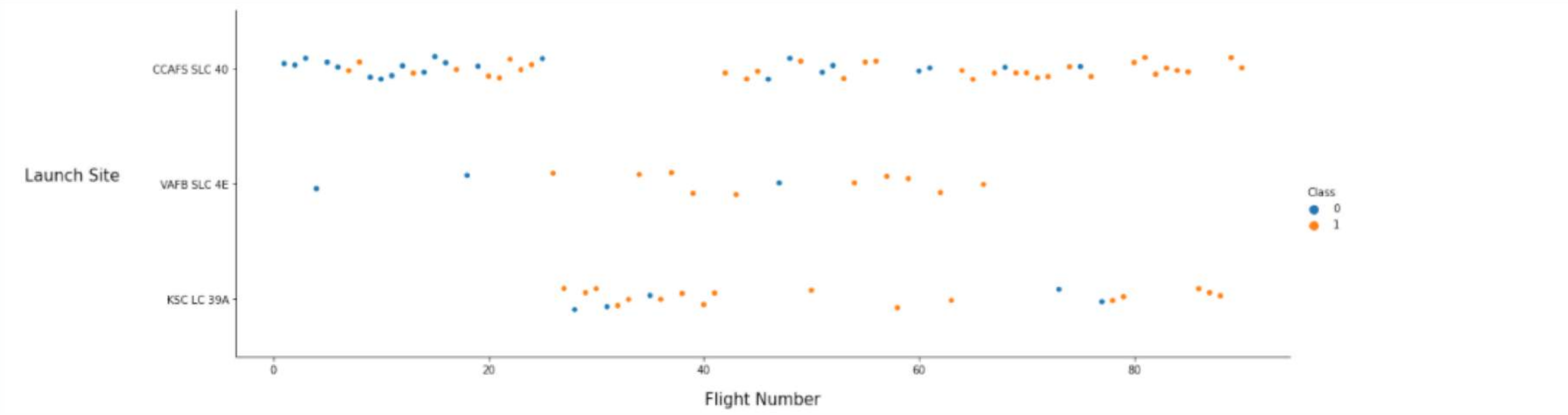
Results

- Exploratory Data Analysis Results: Visualisation, SQL
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site



Observations

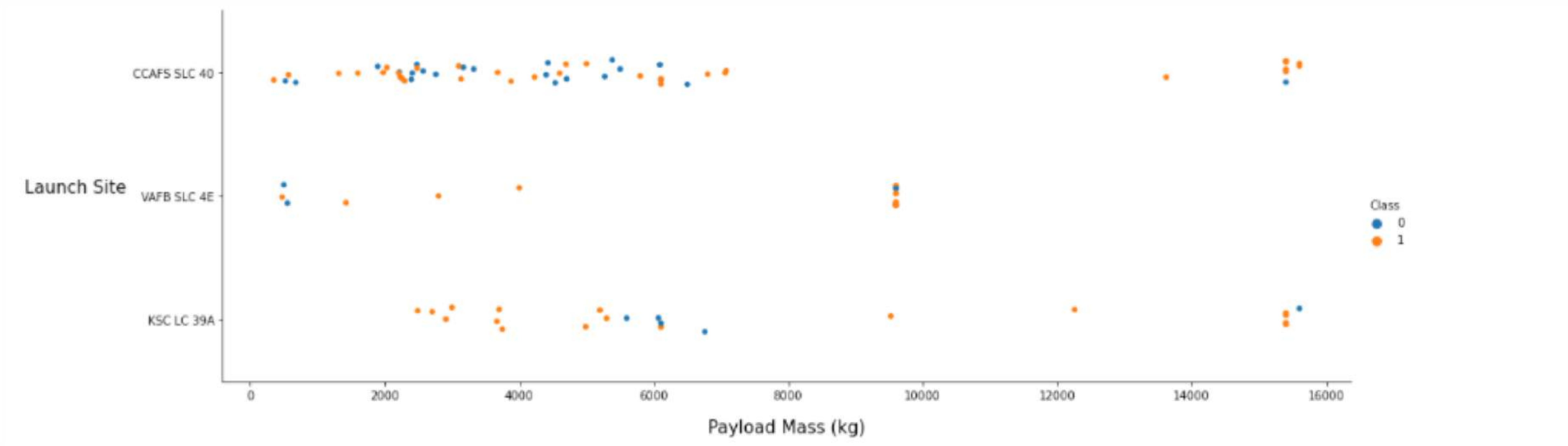
1. The first 6 flights all resulted in failures with the last 13 being successful. May suggest that as time has progressed, we have got better at achieving successful launches.

2. CCAFS SLC 40 launch site has the most launches when compared with the other two sites

3. Comparatively the VAFB SLC 4E and KSC LC 39A have a higher success rate then CCAFS SLC 40

4. During a period of no flights from CCAFS SLC 40, KSC LC 39A flights started being launched and more flights were also launched from VAFB SLC 4E

Payload vs. Launch Site



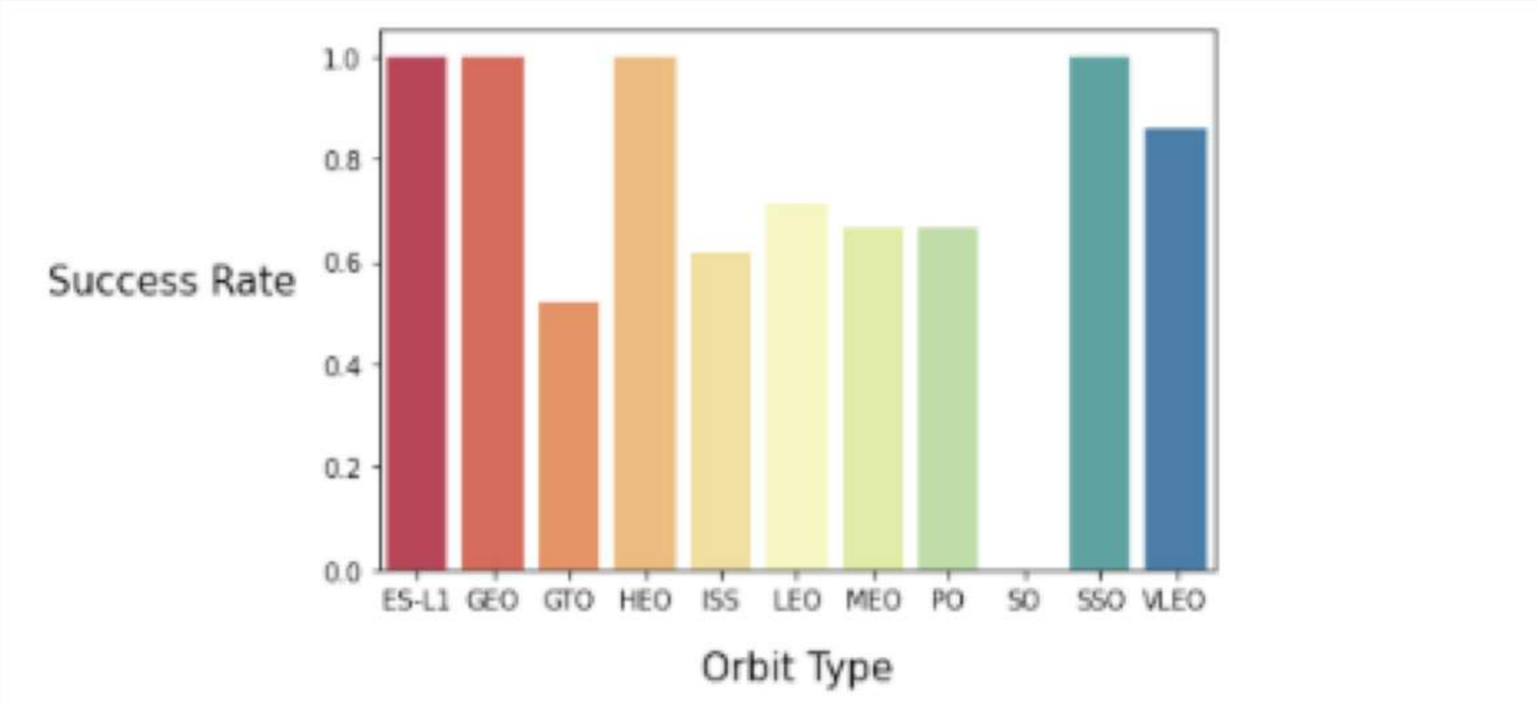
Observations

1. Higher payload masses have a higher rate of success then the low payload masses

2. Appears to be certain payload weights that have been used as a standard across multiple launches (especially around 10000 and 15000), they can be seen on the graph forming almost straight lines

3. The low payload masses launched from KSC LC 39A were successful

Success Rate vs. Orbit Type



	Orbit	Success Rate
0	ES-L1	1.000000
1	GEO	1.000000
3	HEO	1.000000
9	SSO	1.000000
10	VLEO	0.857143
5	LEO	0.714286
6	MEO	0.666667
7	PO	0.666667
4	ISS	0.619048
2	GTO	0.518519
8	SO	0.000000

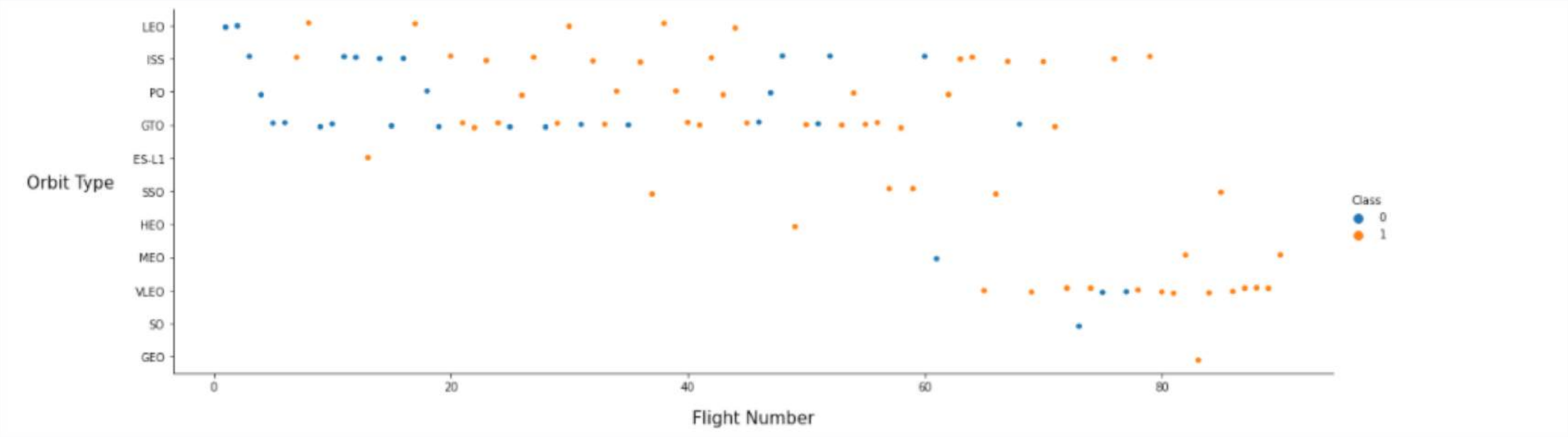
Observations

1. ES-L1, GEO, HEO AND SSO are the standout performers with a 100% success rate

2. Then; VLE0, LEO, MEO and PO,ISS, GTO

3. The worst performer with 0% success rate is SO

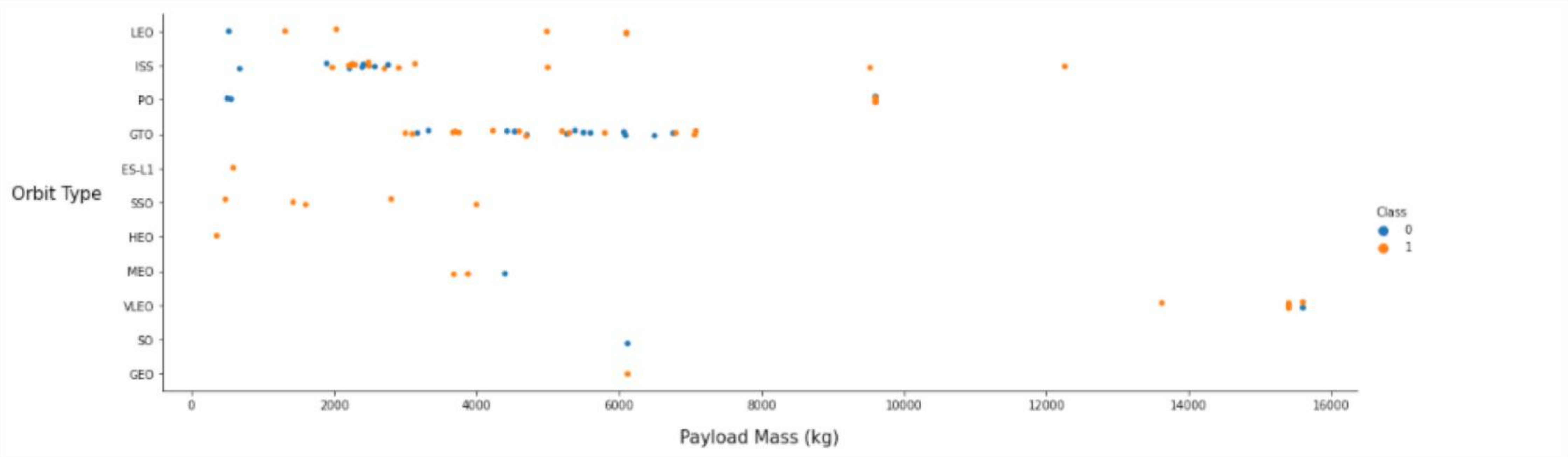
Flight Number vs. Orbit Type



Observations

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.

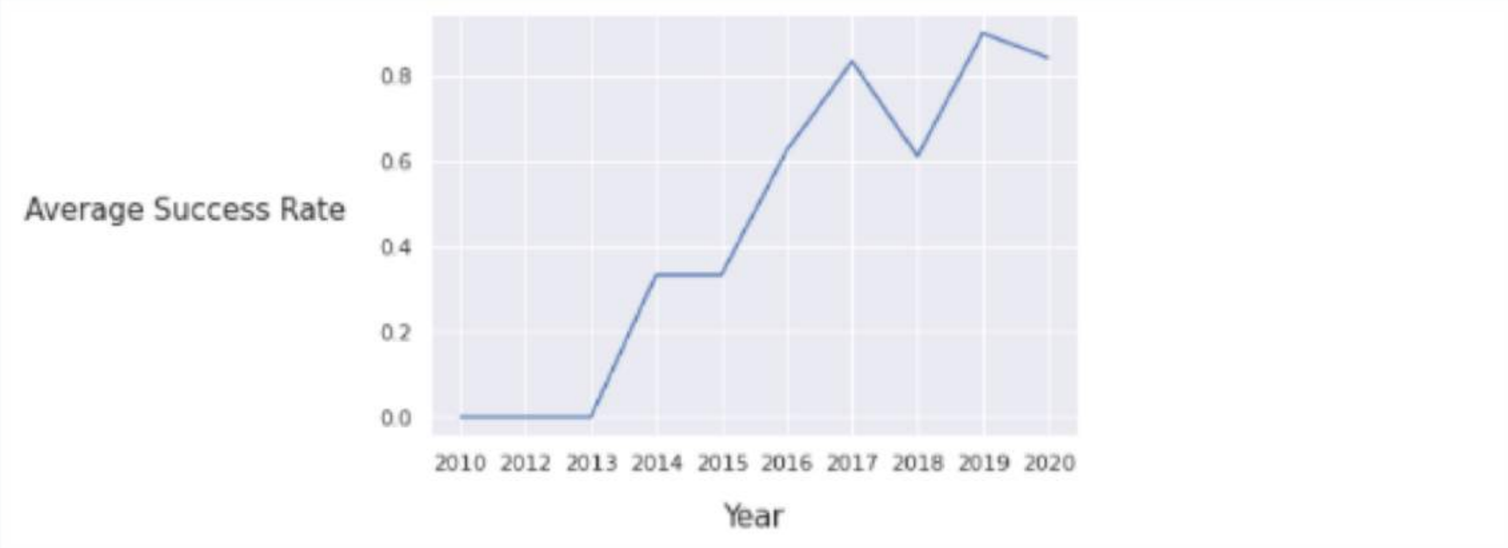
Payload vs. Orbit Type



Observations

Heavy payloads have a negative influence on GTO orbits and positive on GTO and Polar LEO (ISS) orbits.

Launch Success Yearly Trend



<u>Observations</u>

Average Success rate has kept increasing from 2013 to 2020, with a slight decrease during 2018.


EDA with
SQL

All Launch Site Names

SQL QUERY:

select DISTINCT(launch_site) as "Unique launch sites" from SPACEXTBL

Task 1

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

```
%sql select DISTINCT(launch_site) as "Unique launch sites" from SPACEXTBL
```

```
 * ibm_db_sa://qdg91234:***@21fecfd8-47b7-4937-840d-d791d0218660.bs2io90108kqblod8lcg.databases.appdomain.cloud
:31864/bludb
Done.
```

Unique launch sites
CCAFS LC-40
CCAFS SLC-40
KSC LC-39A
VAFB SLC-4E

Launch Site Names Begin with 'CCA'

SQL QUERY:

*select * from SPACEXTBL where launch_site like 'CCA%' limit 5*

Task 2

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

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

```
 * ibm_db_sa://qdg91234:***@21fecfd8-47b7-4937-840d-d791d0218660.bs2io90108kqblod8lcg.databases.appdomain.cloud
:31864/bludb
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	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

SQL QUERY:

select sum(payload_mass__kg_) as "Total payload mass (including NASA CRS, KACIFIC 1) " from SPACEXTBL where customer like 'NASA (CRS)%'

Or

select sum(payload_mass__kg_) as "Total payload mass (excluding NASA CRS, KACIFIC 1) " from SPACEXTBL where customer like 'NASA (CRS)'

Task 3

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

```
%sql select sum(payload_mass__kg_) as "Total payload mass (including NASA CRS, KACIFIC 1) " from SPACEXTBL where
customer like 'NASA (CRS)%'
```

```
 * ibm_db_sa://qdg91234:***@21fecfd8-47b7-4937-840d-d791d0218660.bs2io90108kqblod8lcg.databases.appdomain.cloud
:31864/bludb
Done.
```

Total payload mass(including NASA CRS, KACIFIC 1)
48213

```
%sql select sum(payload_mass__kg_) as "Total payload mass (excluding NASA CRS, KACIFIC 1) " from SPACEXTBL where
customer like 'NASA (CRS)'
```

```
 * ibm_db_sa://qdg91234:***@21fecfd8-47b7-4937-840d-d791d0218660.bs2io90108kqblod8lcg.databases.appdomain.cloud
:31864/bludb
Done.
```

Total payload mass (excluding NASA CRS, KACIFIC 1)
45596

Average Payload Mass by F9 v1.1

SQL QUERY:

select AVG(payload_mass__kg_) from SPACEXTBL where booster_version like '%F9 v1.1%'

Task 4

Display 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%'
```

```
 * ibm_db_sa://qdg91234:***@21fecfd8-47b7-4937-840d-d791d0218660.bs2io90l08kqblod8lcg.databases.appdomain.cloud
:31864/bludb
Done.
```

1
2534

First Successful Ground Landing Date

SQL QUERY:

select min(DATE) as "1st succesfsul landing" from SPACEXTBL where "Landing _Outcome" like 'Success (ground pad)'

Task 5

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

Hint:Use min function

```
%sql select min(DATE) as "1st succesfsul landing" from SPACEXTBL where "Landing _Outcome" like 'Success (ground pad)'
```

```
 * ibm_db_sa://qdg91234:***@21fecfd8-47b7-4937-840d-d791d0218660.bs2io90108kqblod8lcg.databases.appdomain.cloud:31864/bludb
Done.
```

1st succesfsul landing
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

SQL QUERY:

select booster_version from SPACEXTBL where "Landing _Outcome" like 'Success (drone ship)' and payload_mass__kg_ between 4000 and 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 SPACEXTBL where "Landing _Outcome" like 'Success (drone ship)' and payload_mass
__kg_ between 4000 and 6000
```

```
 * ibm_db_sa://qdg91234:***@21fecfd8-47b7-4937-840d-d791d0218660.bs2io90108kqblod8lcg.databases.appdomain.cloud
:31864/bludb
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

SQL QUERY:

select count(*) as "Sucess" from SPACEXTBL where mission_outcome like 'Success%’

Task 7

List the total number of successful and failure mission outcomes

```
%sql select count(*) as "Sucess" from SPACEXTBL where mission_outcome like 'Success%'
```

```
 * ibm_db_sa://qdg91234:***@21fecfd8-47b7-4937-840d-d791d0218660.bs2io90108kqblod8lcg.databases.appdomain.cloud
:31864/bludb
Done.
```

Sucess
100

select count(*) as "Failure" from SPACEXTBL where mission_outcome like 'Failure%'

```
%sql select count(*) as "Failure" from SPACEXTBL where mission_outcome like 'Failure%'
```

```
 * ibm_db_sa://qdg91234:***@21fecfd8-47b7-4937-840d-d791d0218660.bs2io90108kqblod8lcg.databases.appdomain.cloud
:31864/bludb
Done.
```

Failure
1

Boosters Carried Maximum Payload

SQL QUERY:

select DISTINCT(booster_version) from SPACEXTBL where payload_mass__kg_ = (select max(payload_mass__kg_) from SPACEXTBL)

Task 8

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

```
%sql select DISTINCT(booster_version) from SPACEXTBL where payload_mass__kg_ = (select max(payload_mass__kg_) fr
om SPACEXTBL)
```

```
 * ibm_db_sa://qdg91234:***@21fecfd8-47b7-4937-840d-d791d0218660.bs2io90108kqb1od8lcg.databases.appdomain.cloud
:31864/bludb
Done.
```

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

SQL QUERY:

select "Landing _Outcome", booster_version, launch_site from SPACEXTBL where "Landing _Outcome" like 'Failure (drone ship)' and DATE like '2015%'

Task 9

List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

```
%sql select "Landing _Outcome", booster_version, launch_site from SPACEXTBL where "Landing _Outcome" like 'Failu
re (drone ship)' and DATE like '2015%'
```

```
 * ibm_db_sa://qdg91234:***@21fecfd8-47b7-4937-840d-d791d0218660.bs2io90108kqblod8lcg.databases.appdomain.cloud
:31864/bludb
Done.
```

Landing _Outcome	booster_version	launch_site
Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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

SQL QUERY:

select "Landing _Outcome", COUNT("Landing _Outcome") as "No. of Outcomes" from SPACEXTBL where DATE between '2010-06-04' and '2017-03-20' group by "Landing _Outcome" order by "No. of Outcomes" DESC

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("Landing _Outcome") as "No. of Outcomes" from SPACEXTBL where DATE between
'2010-06-04' and '2017-03-20' group by "Landing _Outcome" order by "No. of Outcomes" DESC
```

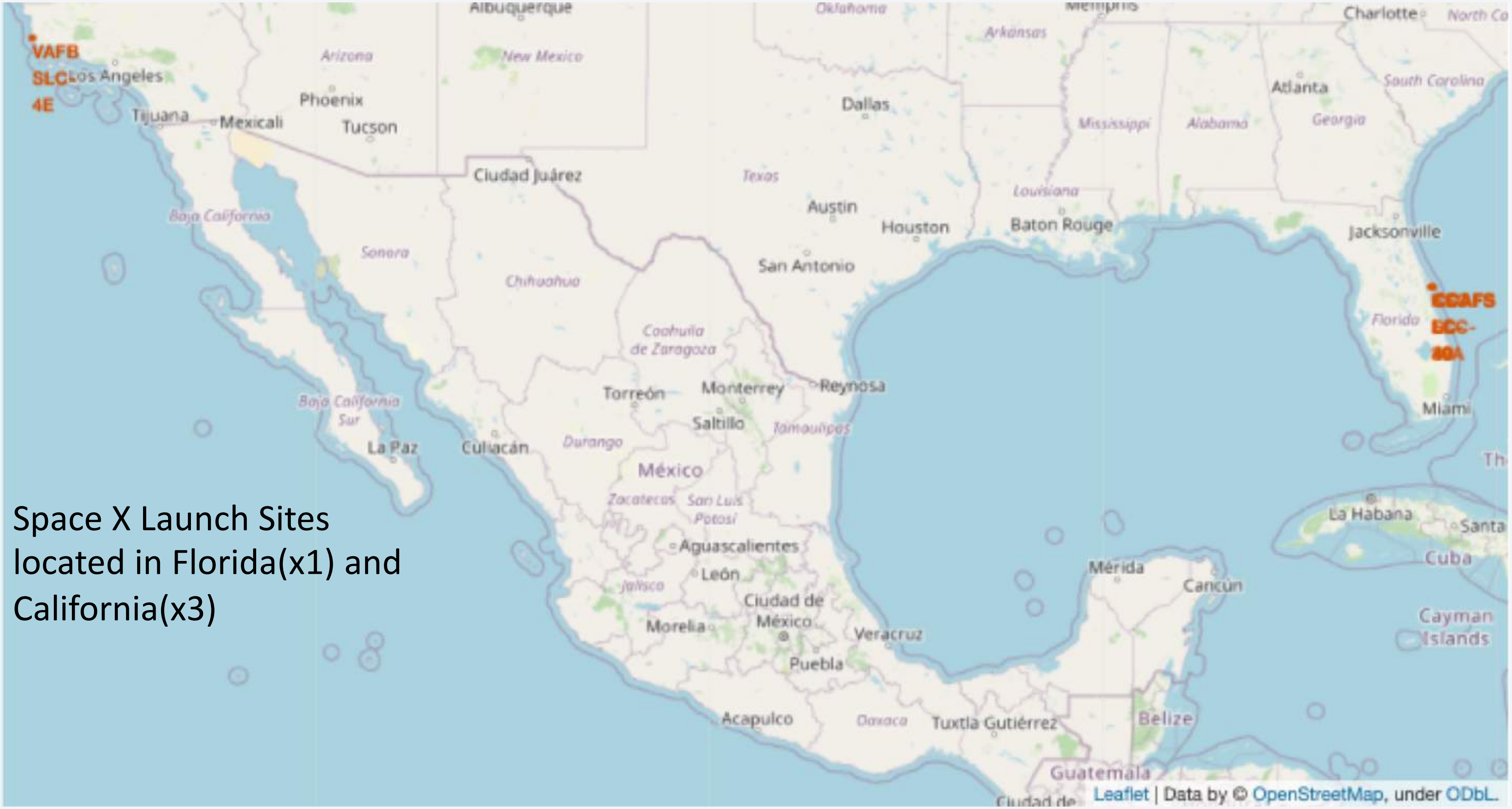
```
 * ibm_db_sa://qdg91234:***@21fecfd8-47b7-4937-840d-d791d0218660.bs2io90108kqblod8lcg.databases.appdomain.cloud
:31864/bludb
Done.
```

Landing _Outcome	No. of Outcomes
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

Section 4

Launch Sites Proximities Analysis

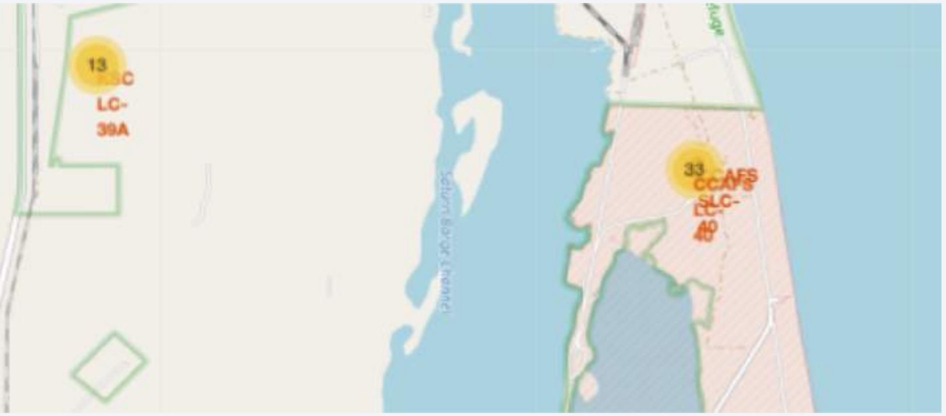
Launch Sites on Global Map

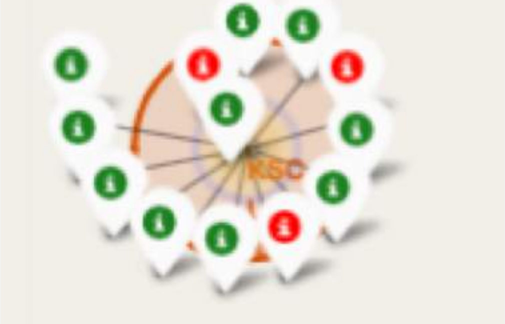


Space X Launch Sites located in Florida(x1) and California(x3)

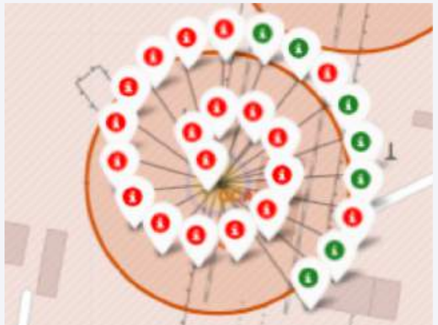
Coloured Labelled Markers: Success and Failure

Florida Sites: KSC LC-39A, CCAFS LC-40, CCAFS SLC-40

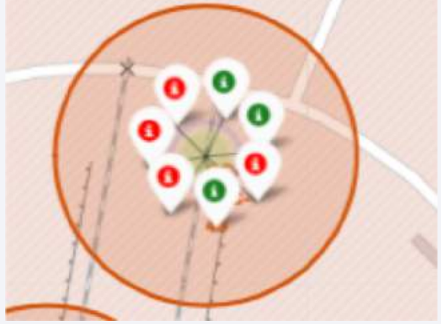




KSC LC-39A



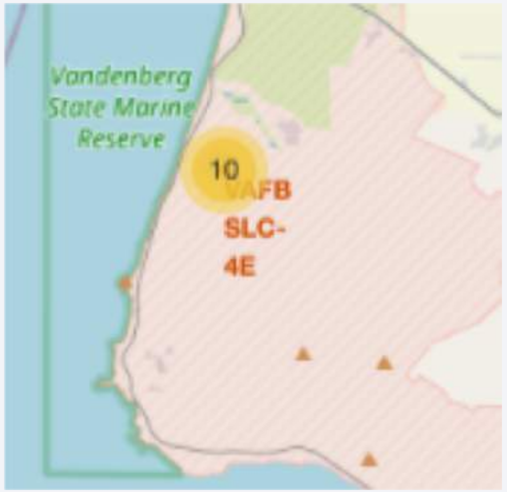
CCAFS LC-40

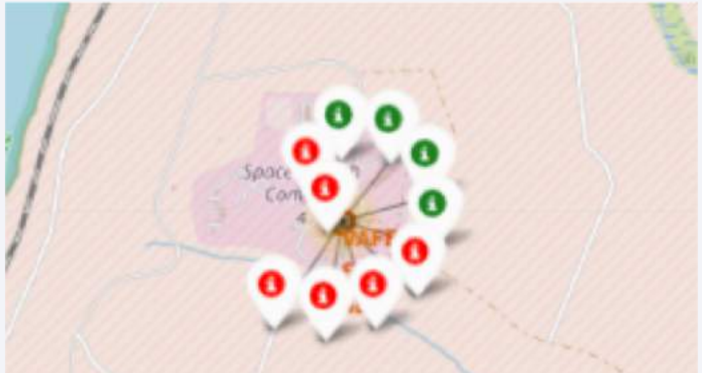


CCAFS SLC-40

Green Icon markers represent successful launches and Red Icon markers represent failed launches

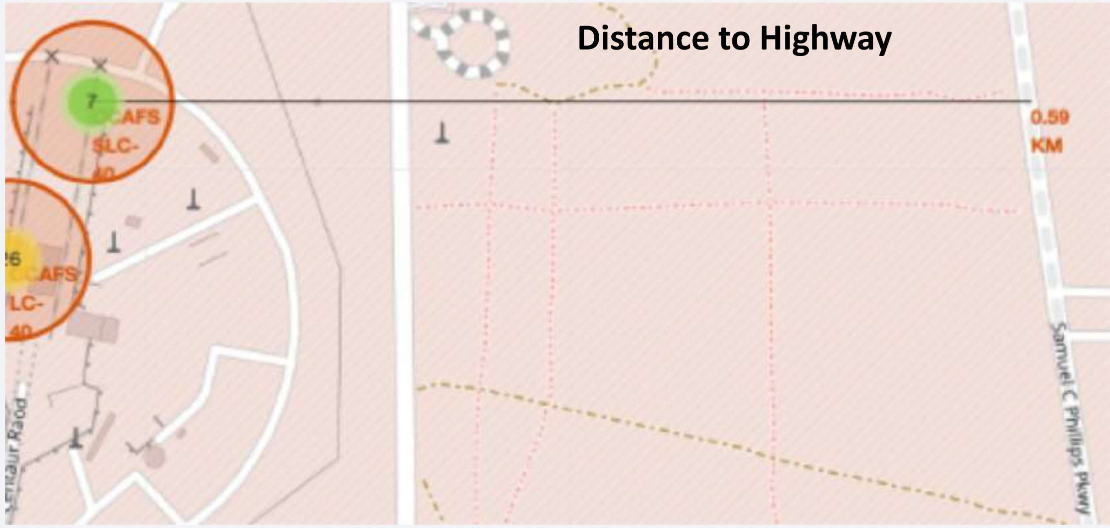
California Site: VAFB SLC-4E

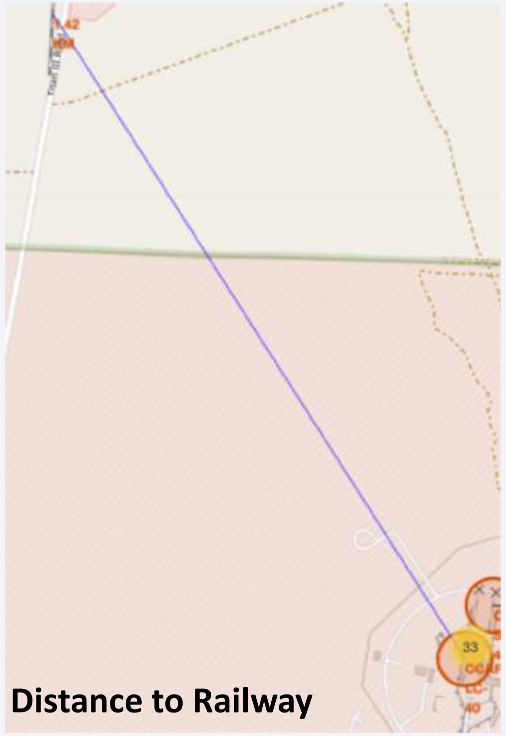


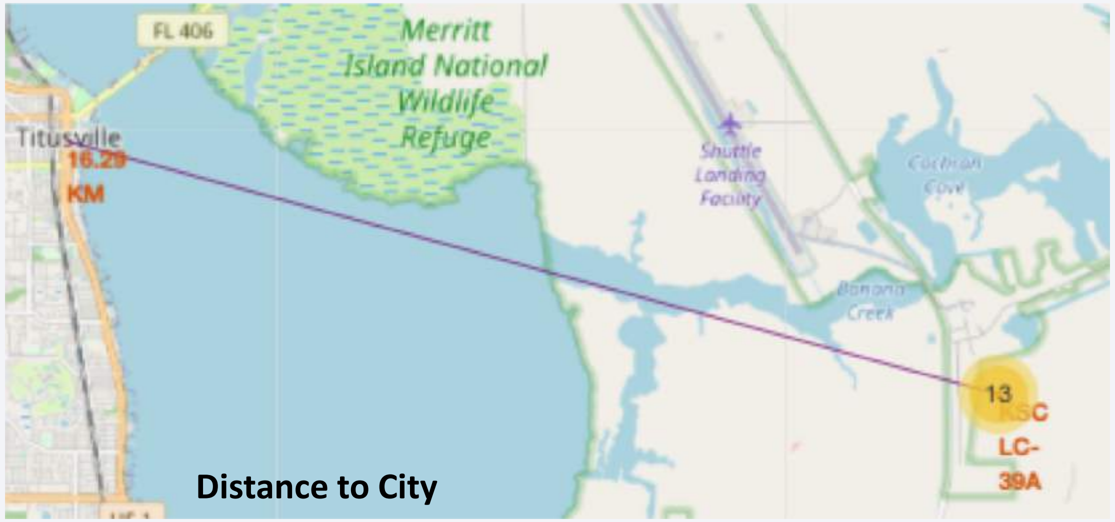


VAFB SLC-4E

Launch Sites Proximity to Cities, Railways, and Highways







<u>Observations</u>

Are launch sites in close proximity to; railways, highways, and coastlines?

They are in close proximity to all of those:

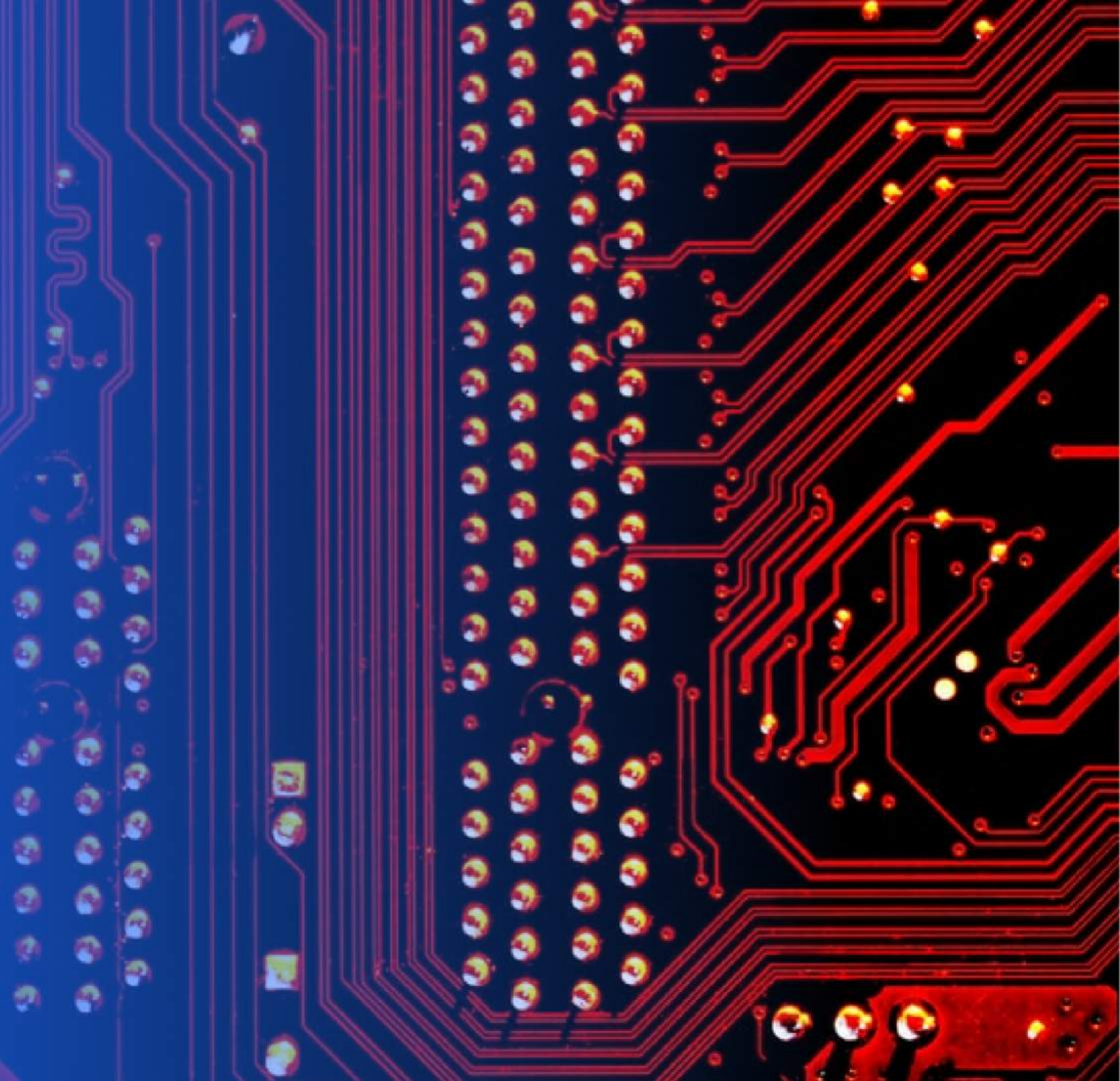
- To the coastline so they can launch rockets with the added layer of safety of the ocean
- They are relatively close to railways and highways as people need to get in and out of the launch sites for work, also if something wen wrong they may need an easy to to evacuate the sites

Do launch sites keep certain distance away from cities?

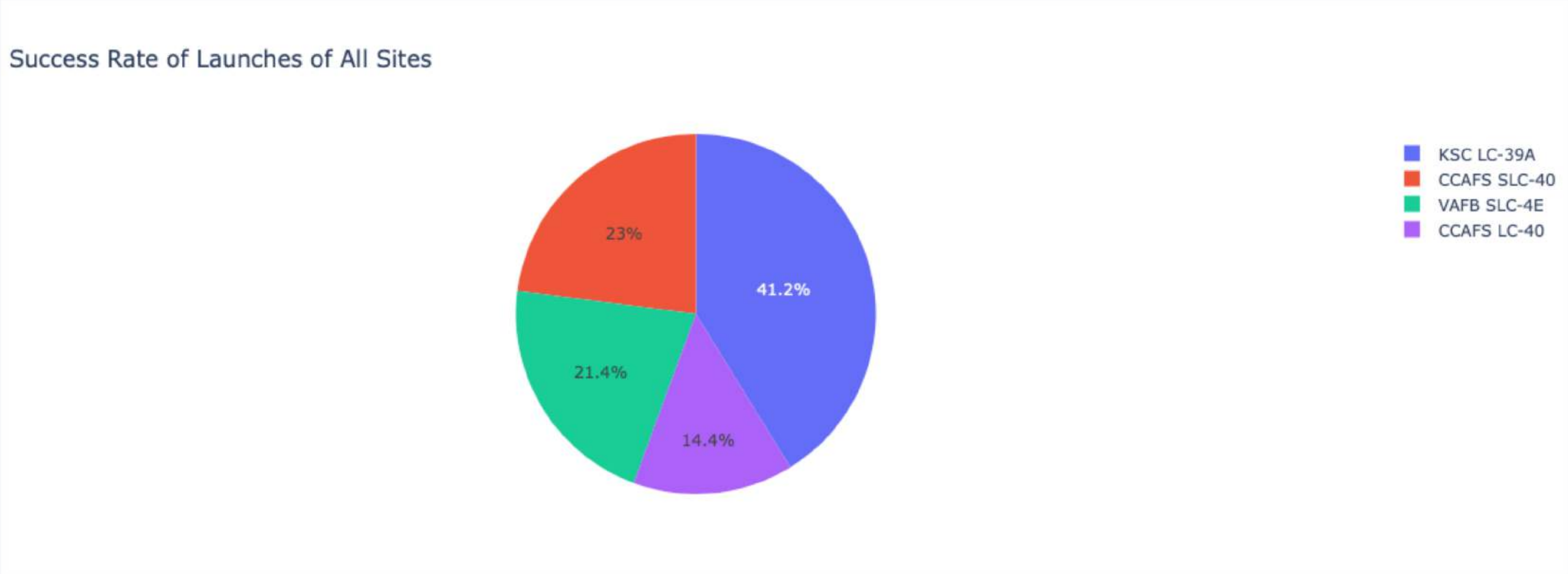
- They are likely situated away from cities to reduce the threat to populated areas and any noise disruption that may occur

Section 5

Build a Dashboard with Plotly Dash



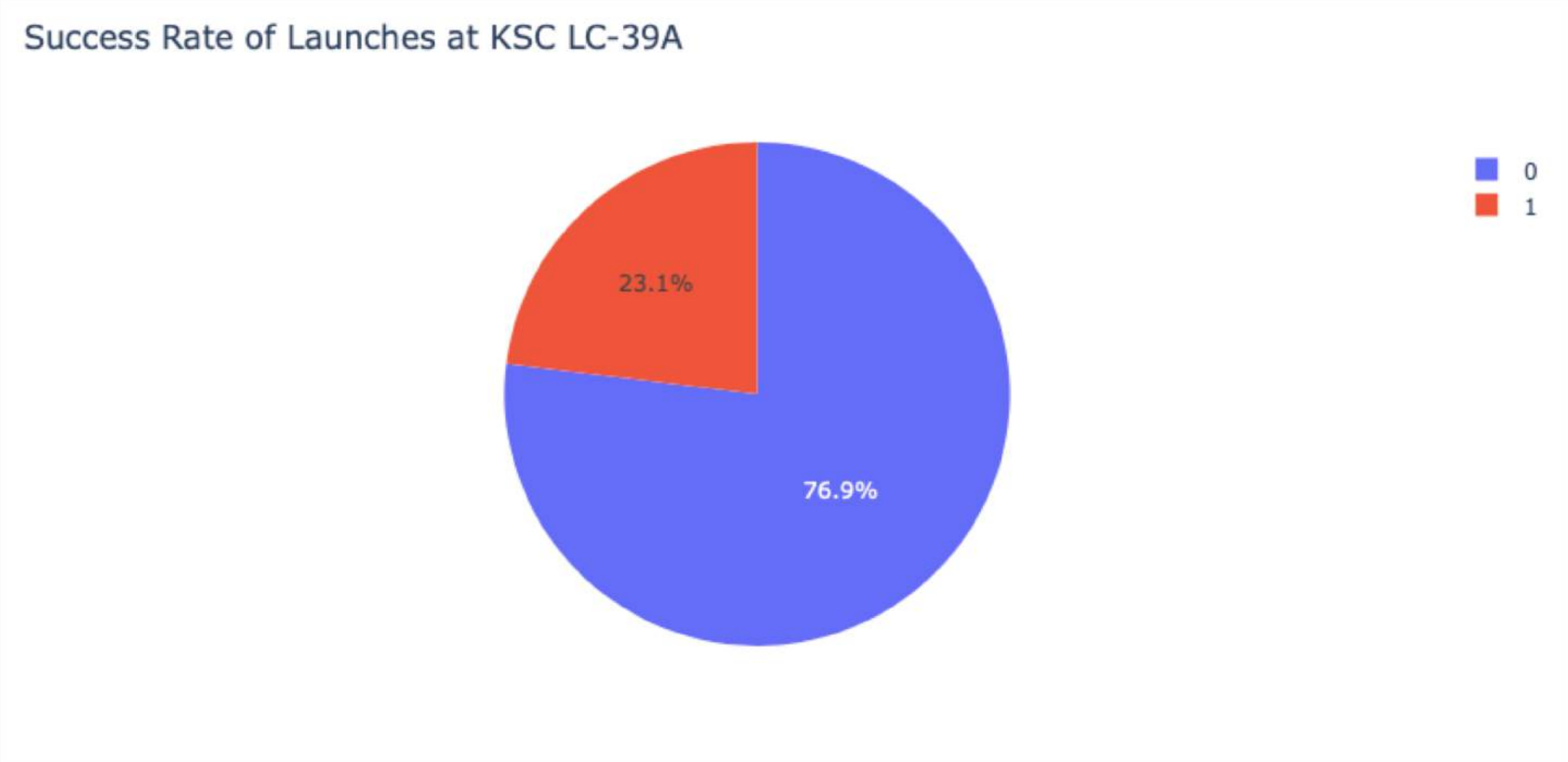
Dashboard: Success Rate Achieved by all Sites



Observations

KSC LC-394 had the highest successful rate when compared with the other sites.

Dashboard: Success Rate of Most Successful Launch Site

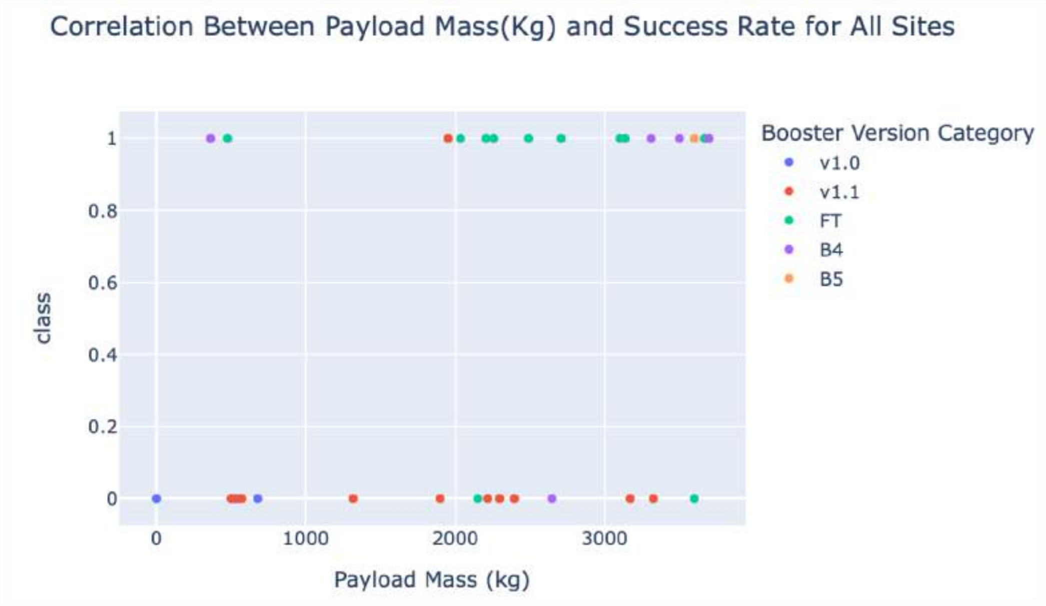


Observations

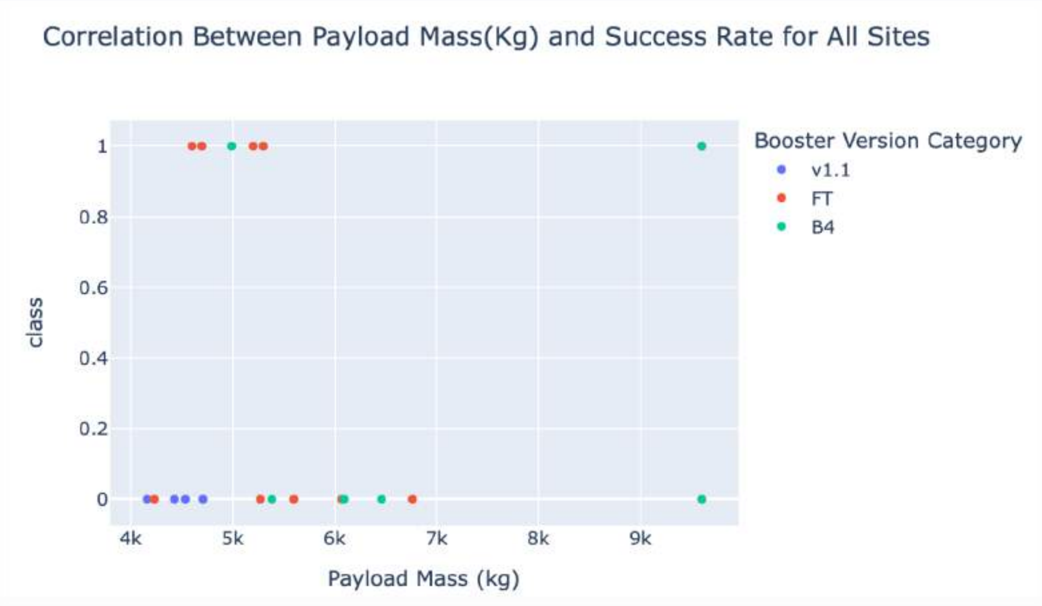
KSC LC-394 has a success rate of 76.9% for launches (0=Failures, 1=Successes).

<Dashboard Screenshot 3>

Where Payload Mass between: 0- 4000Kg



Where Payload Mass between: 4000- 10000Kg



<u>Observations</u>

There are more rockets with lower weighted payloads (0-4000Kg) who have successfully landed compared with rockets with a larger payload. This suggests that rockets with a lower payload mass are more successful than rockets with a higher payload mass

Section 6

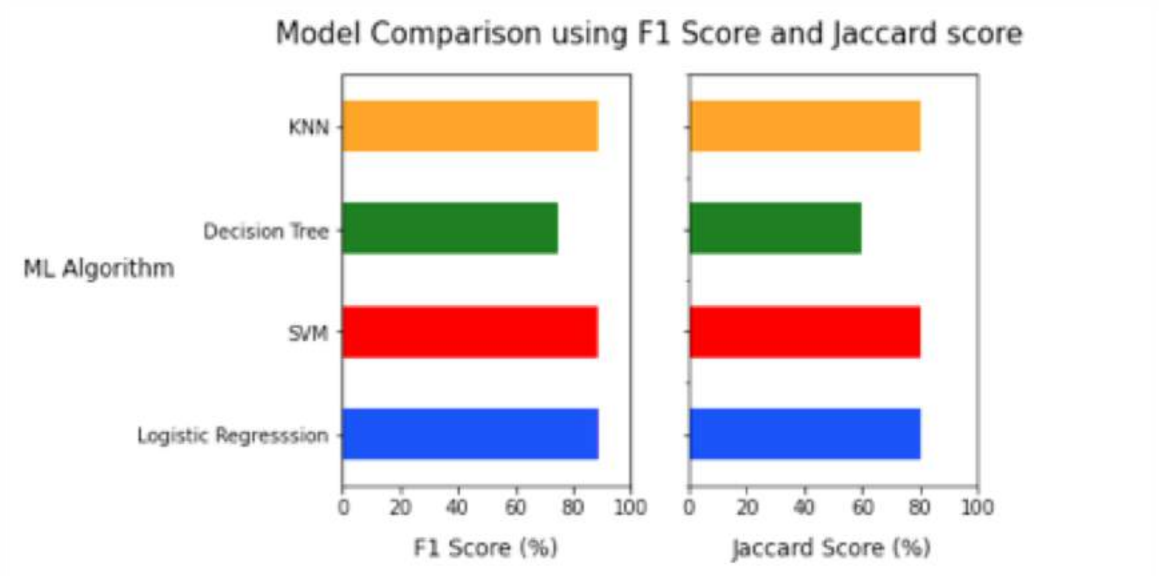
Predictive Analysis (Classification)

Classification Accuracy

The ML Models were evaluated on the test set using various metrics including; Accuracy Score, F1 score, Jaccard Score, and Log Loss.

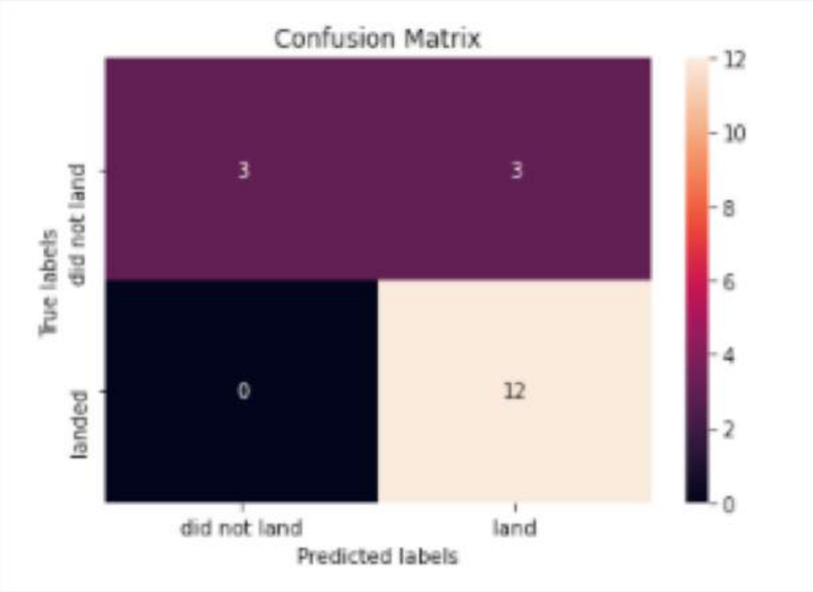
SVM, Logistic regression and KNN performed identically in terms of; Accuracy, F1 and Jaccard Score. While the models scored highly they were all susceptible to false positives. The decision tree algorithm performed worse than the others, with it being susceptible to both false positives and false negatives.

ML Algorithm	F1 Score(%)	Jaccard Score (%)	Log Loss (%)	Accuracy Score (%)
Logistic Regresssion	88.889	80.0	Na	83.333
SVM	88.889	80.0	Na	83.333
Decision Tree	75.000	60.0	Na	66.667
KNN	88.889	80.0	47.867	83.333

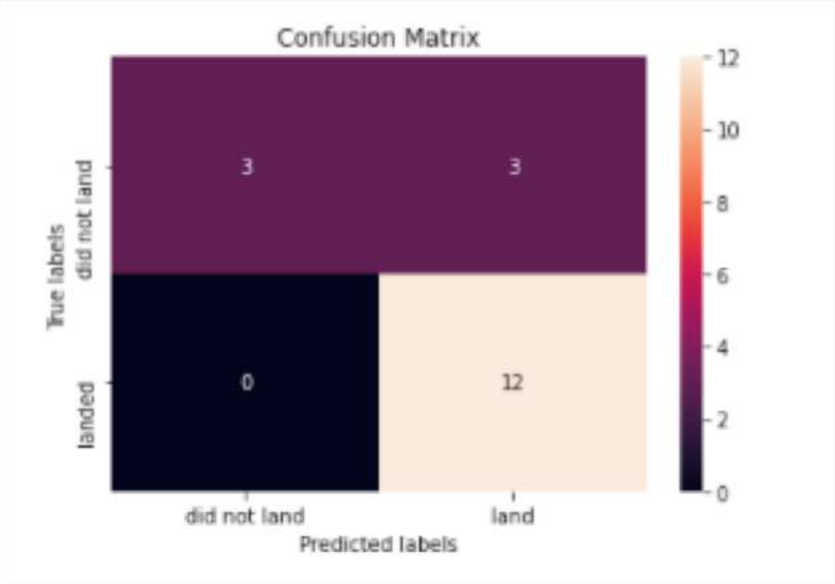


Confusion Matrix

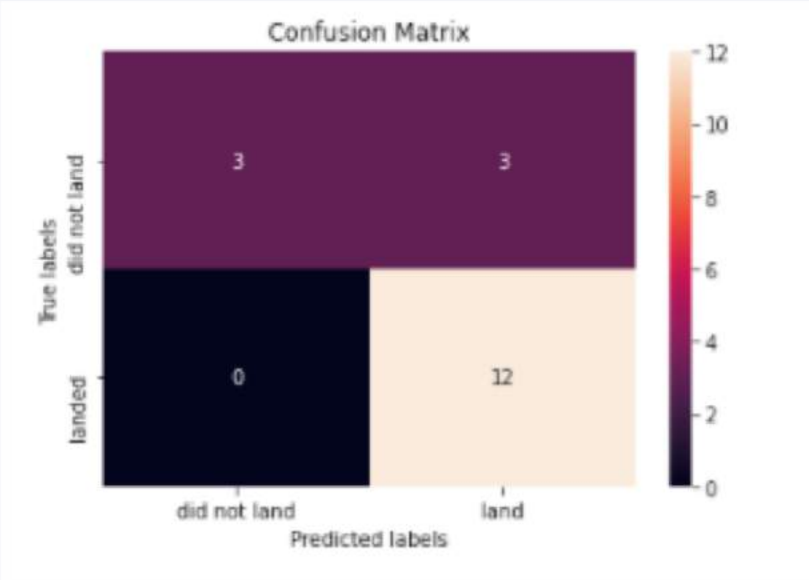
Logistic Regression



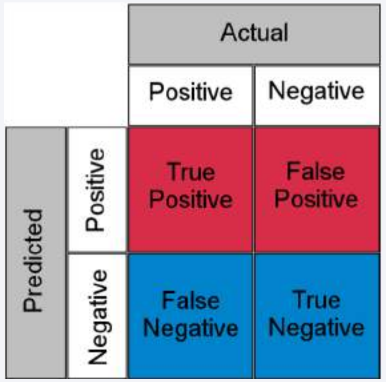
SVM



KNN



It's no surprise that the Logistic Regression, SVM and KNN models all performed identically when being evaluated as they had the exact same number of; True positive, True Negative, False Negatives and False Positives. The models all performed relatively the models main limitation was incorrectly predicting the positive class (because of the three occurrences of False Positives).



Conclusion

- Three ML algorithms performed identically; Logistic Regression, SVM, and KNN. They were equally the best choice for the dataset.
- Space X Success Rates:
    - Rockets with a lower payload mass perform better than rockets with a larger payload mass
    - Orbit types; GEO, SSO, ES-L1 and HEO are most successful for landing rockets.
    - KCS LC-39A is the most successful site for launching rockets
    - From 2013 to the present day, the success rates of Space X have steadily increased and continue to trend upwards.

Thank you!