IBM
Data Science
Capstone Project
SPACEX Data

Niranjhan Palanikumar

May 02, 2022

Presentation Outline

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

Executive Summary

The main aim of this project was to determine if the first stage of the rocket launch will successfully land.

Summary of methodologies

-The dataset used was SpaceX's previous Falcon 9 launches

-Data collection and data wrangling

-Interactive maps

-Building models for predictive analysis

Summary of all results

-Interactive dashboard for visual analytics.

-Comparing accuracy to choose the best model performance.

Introduction

Project background and context

We will predict if the Falcon 9 first stage will land successfully. 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 SpaceX 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.

Problems you want to find answers

- Factors for a successful landing of stage one

- Impact/effect of other factors on the landing outcome

Methodology

Methodology

Data collection methodology:

- Using SpaceX REST APIs
- Web Scrapping from Wikipedia

Data wrangling:

- Missing values, One Hot Encoding the categorical fields

Perform exploratory data analysis (EDA) using visualization and SQL

- Scatter plots and bar graphs to identify patterns and trends

Perform interactive visual analytics using Folium and Plotly Dash

- To help in quick visual analysis of major parameters like launch site and payload mass.

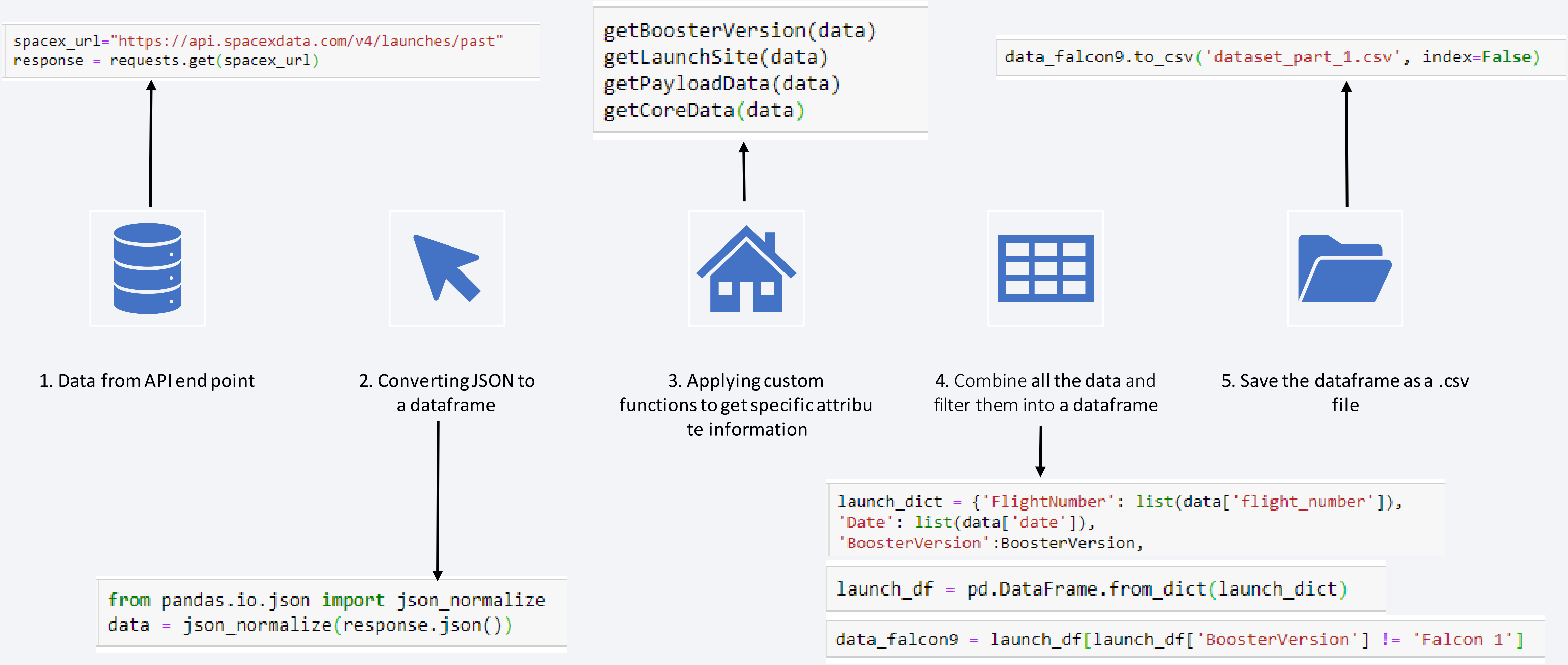
Perform predictive analysis using classification models

- Building the model with multiple algorithms and comparing their performance

Data Collection

- The dataset used for this analysis was collected from a combination of SPACEX REST APIs.
- SPACEX REST API endpoint -> https://api.spacexdata.com/v4/
- Different REST API urls contain their specific information, eg.
  - https://api.spacexdata.com/v4/cores contains information on the type of "cores" used.
  - https://api.spacexdata.com/v4/capsules contains information on the type of "capsules" used.
- Custom functions were written and applied to obtain specific information from their respective API endpoints.
- The information were put together and converted into .csv file for storage, and into a dataframe for initial exploratory analysis.

Data Collection from API Flowchart



1. Data from API end point

```
spacex_url="https://api.spacexdata.com/v4/launches/past"
response = requests.get(spacex_url)
```

2. Converting JSON to a dataframe

```
from pandas.io.json import json_normalize
data = json_normalize(response.json())
```

3. Applying custom functions to get specific attribute information

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

4. Combine all the data and filter them into a dataframe

```
launch_dict = {'FlightNumber': list(data['flight_number']),
'Date': list(data['date']),
'BoosterVersion':BoosterVersion,
```

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

```
data_falcon9 = launch_df[launch_df['BoosterVersion'] != 'Falcon 1']
```

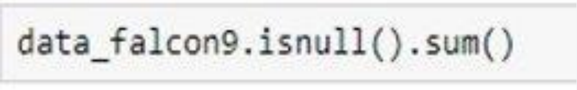
5. Save the dataframe as a .csv file

```
data_falcon9.to_csv('dataset_part_1.csv', index=False)
```

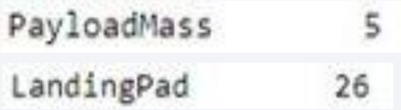
GitHub URL

Finalizing Data Collection

- The last step in data collection and preparing the data for analysis is identifying and dealing with missing values
  - isnull() - returns value '1' for 'True' case and value '0' for 'False' cases
  - sum() - performs simple addition on the given set of numbers



```
data_falcon9.isnull().sum()
```

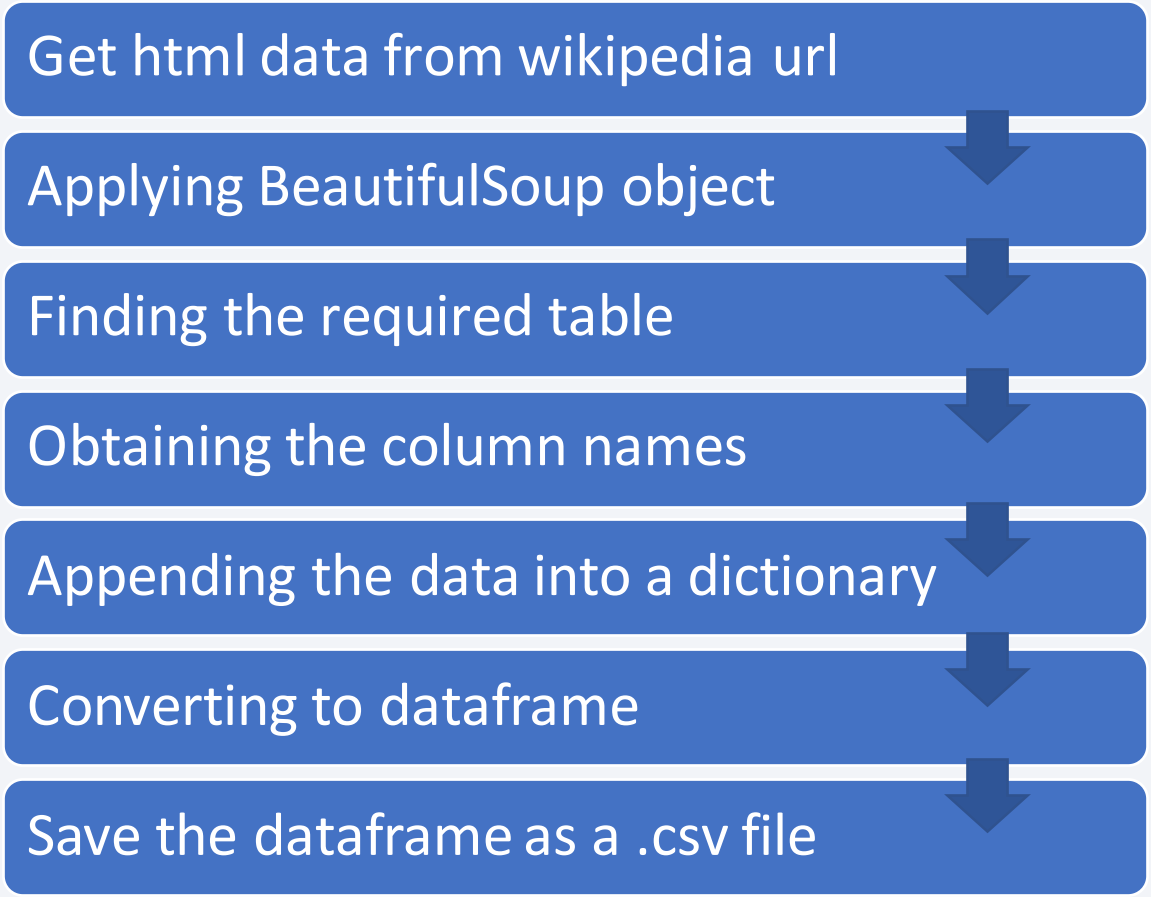


```
PayloadMass      5
LandingPad      26
```

- Here the missing PayloadMass was replaced with the mean of the column, and the missing values in LandingPad were retained to signify that landing pads were "not used" in these launches.

[GitHub URL]

Data Collection Webscrapping Flowchart



Get html data from wikipedia url

Applying BeautifulSoup object

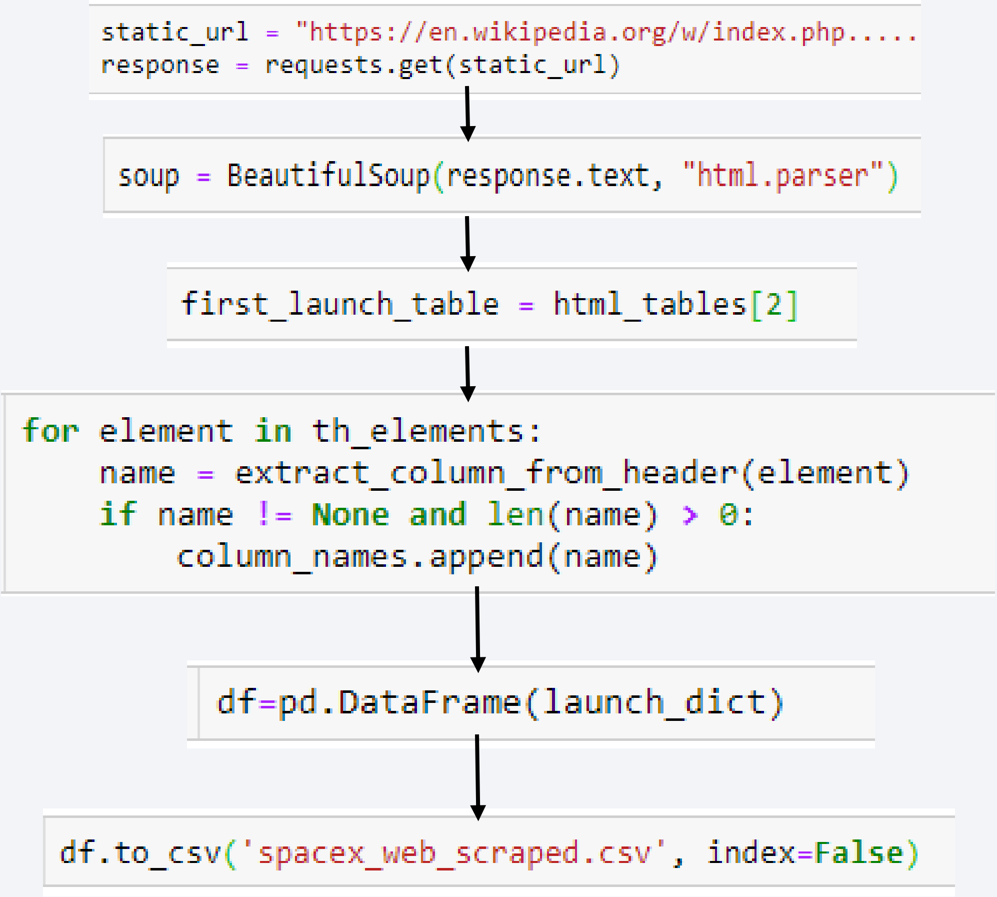
Finding the required table

Obtaining the column names

Appending the data into a dictionary

Converting to dataframe

Save the dataframe as a .csv file



```
static_url = "https://en.wikipedia.org/w/index.php.....
response = requests.get(static_url)
```

```
soup = BeautifulSoup(response.text, "html.parser")
```

```
first_launch_table = html_tables[2]
```

```
for element in th_elements:
    name = extract_column_from_header(element)
    if name != None and len(name) > 0:
        column_names.append(name)
```

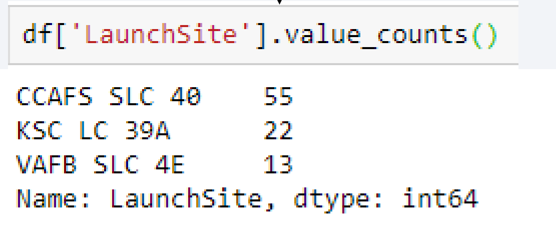
```
df=pd.DataFrame(launch_dict)
```

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

GitHub URL

Data Wrangling steps

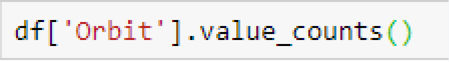
Calculate number of launches at each site



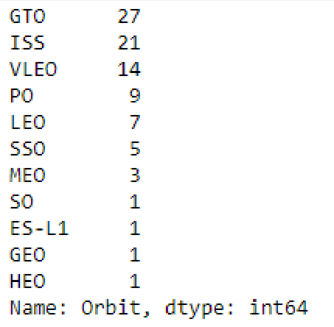
```
df['LaunchSite'].value_counts()
```

```
CCAFS SLC 40    55
KSC LC 39A      22
VAFB SLC 4E     13
Name: LaunchSite, dtype: int64
```

Calculate the number and occurrence of each orbit

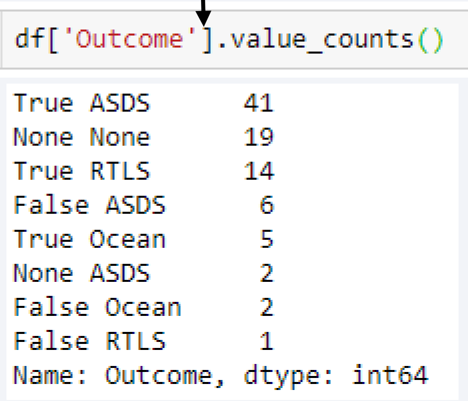


```
df['Orbit'].value_counts()
```



```
GTO      27
ISS      21
VLEO     14
PO        9
LEO       7
SSO       5
MEO       3
SO        1
ES-L1     1
GEO       1
HEO       1
Name: Orbit, dtype: int64
```

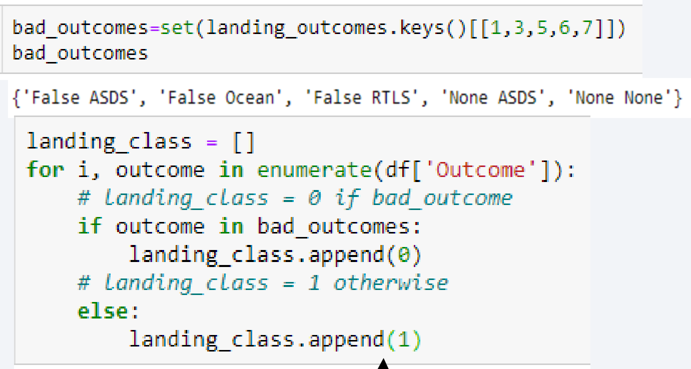
Calculate the number of occurrence and mission outcome per orbit



```
df['Outcome'].value_counts()
```

```
True ASDS      41
None None      19
True RTLS      14
False ASDS      6
True Ocean      5
None ASDS       2
False Ocean     2
False RTLS      1
Name: Outcome, dtype: int64
```

Create a new landing outcome label



```
bad_outcomes=set(landing_outcomes.keys()[[1,3,5,6,7]])
bad_outcomes
```

```
{'False ASDS', 'False Ocean', 'False RTLS', 'None ASDS', 'None None'}
```

```
landing_class = []
for i, outcome in enumerate(df['Outcome']):
    # landing_class = 0 if bad_outcome
    if outcome in bad_outcomes:
        landing_class.append(0)
    # landing_class = 1 otherwise
    else:
        landing_class.append(1)
```

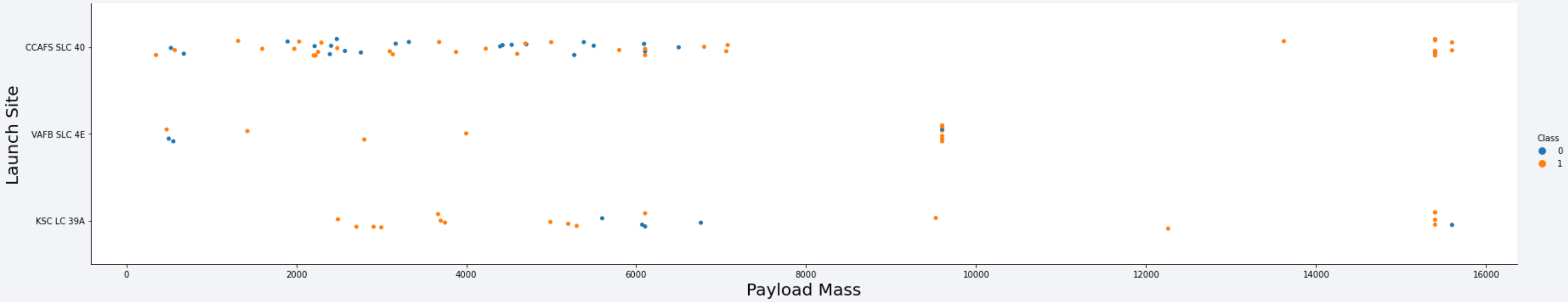
-Value = '1' for successful landing

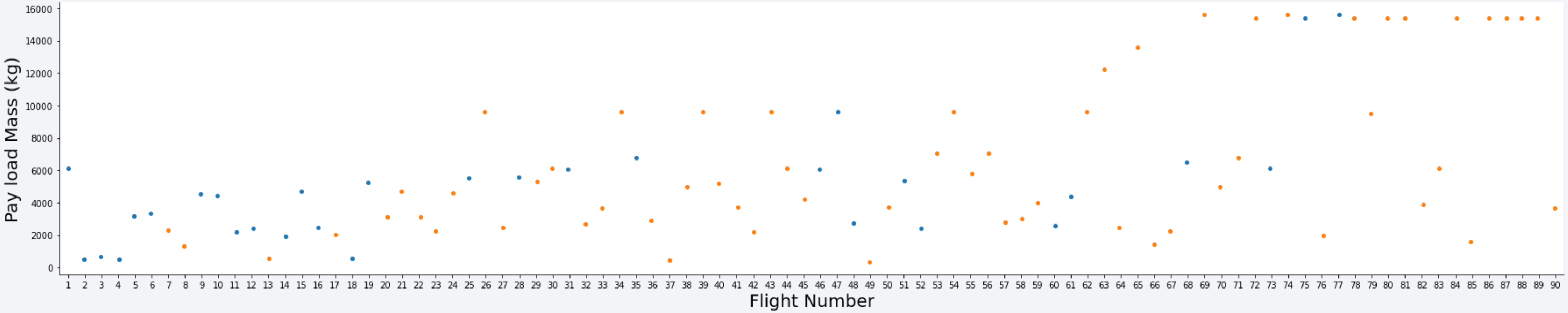
-Value = '0' for unsuccessful landing

GitHub URL

EDA with Data Visualization

- Scatter plot indicating their respective outcome class were drawn for:
  - •Flight Number vs Pay Load Mass
  - •Flight Number vs Launch Site
  - •Launch Site vs Payload Mass
  - •Flight Nuber vs Orbit
  - •Orbit vs Pay Load Mass

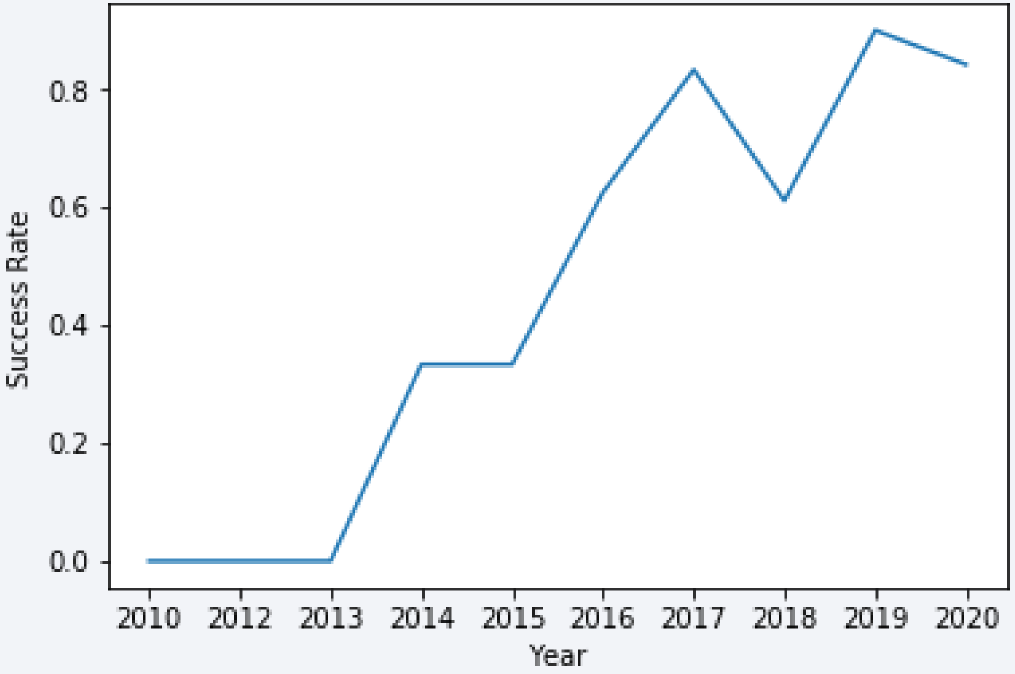


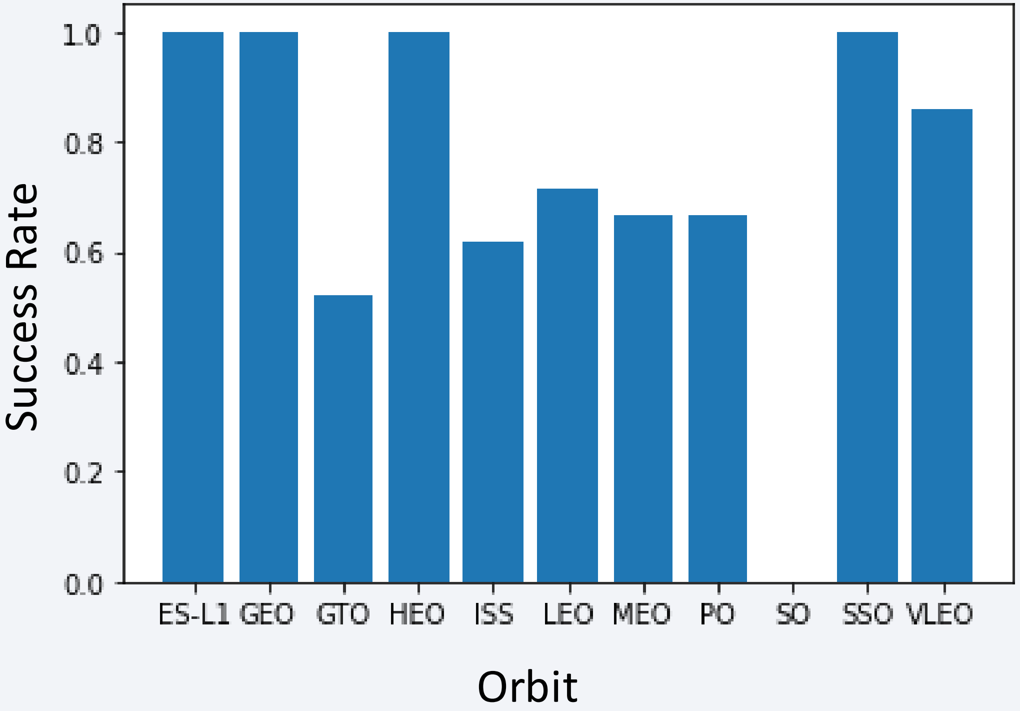


GitHub URL

EDA with Data Visualization

- Bar chart on the success rate of each orbit type
- Line plot on the yearly success trend





GitHub URL

EDA with SQL

The SQL queries performed to extract:

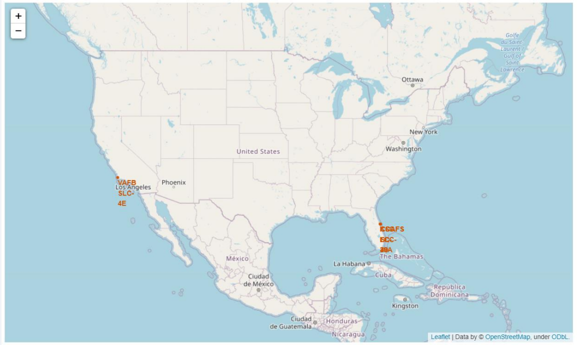
- Names of unique launch sites
- Records where launch sites begin with string 'CCA'
- The total payload mass carried by boosters launched by NASA(CRS)
- Average payload mass carried by booster version F9 v1.1
- Date when the first successful landing outcome in ground pad was achieved
- Names of boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
- Total number of successful and failure mission outcomes
- Names of the booster versions which have carried the maximum payload mass
- Booster versions and launch site for failed landing outcomes on drone ship in 2015
- Rank the count of landing outcomes (eg Failure (drone) or Success (ground pad)) between 2010-06-04 and 2017-03-20 in descending order

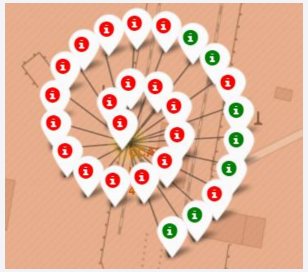
GitHub URL

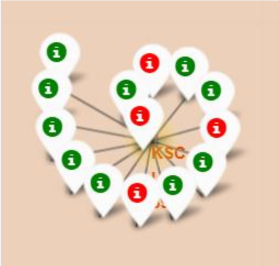
Build an Interactive Map with Folium

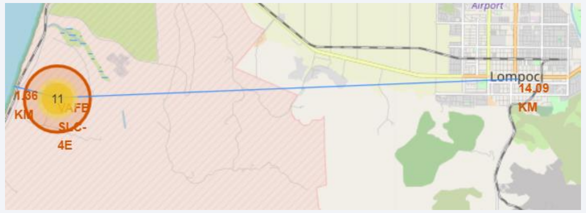
- The latitude and longitude coordinates of each launch site were marked with a circle and a label.
- Depending on the outcome of each launch an icon_marker was used in their respective color, "Green" for "Success" and "Red" for "Failure".
- The distance between different landmarks or entities were from each launch sites eg. Coastline, Railway, City.

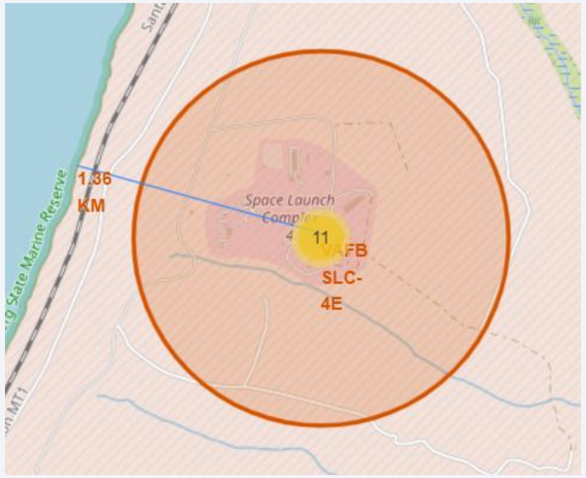
A major trend observed here was that every launch site is situated very close to a coastline and far away from the cities.











GitHub URL

Build a Dashboard with Plotly Dash

The plotly dashboard built has two main categories for visual analytics:

- Selection of launch site
- Weight (kg) parameter for payload mass

The two graphs for analysis:

- A pie-chart depicting the success and failure rate of a selected launch site
- A scatter plot showing the correlation between the payload mass and success of the launch, for a selected launch site with a modifiable weight parameter for payload

GitHub URL

Predictive Analysis (Classification)

Splitting the dataset into feature set and target set.

Feature engineering and standardizing the dataset

Splitting the dataset into training and test set

Building multiple classification model, and choosing the best **parameters using** Grid Search method:

- Logistic Regression
- Support Vector Machine
- Decision Tree
- K-Nearest Neighbors

GitHub URL

Results

EXPLORATORY DATA ANALYSIS

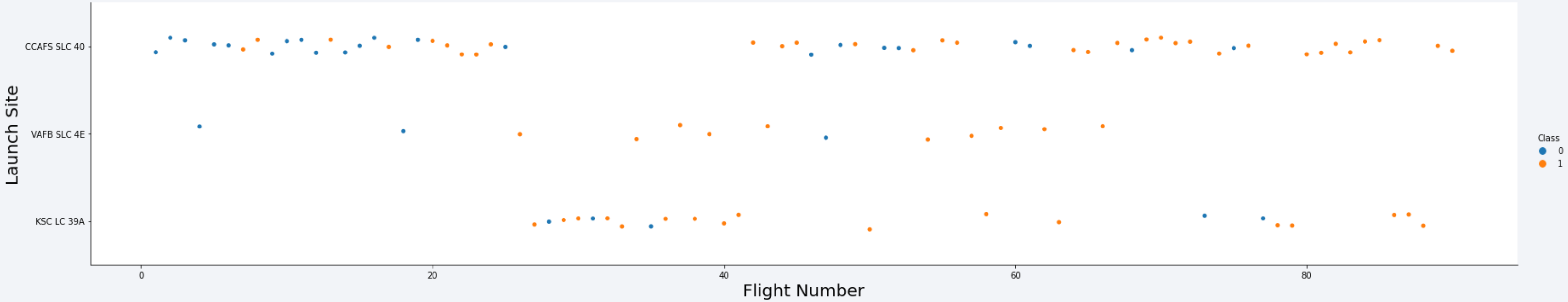
INTERACTIVE VISUAL ANALYTICS

PREDICTIVE ANALYSIS USING MACHINE LEARNING

Insights from Exploratory Data Analysis

Flight Number vs. Launch Site

- Scatter plot of Flight Number vs. Launch Site

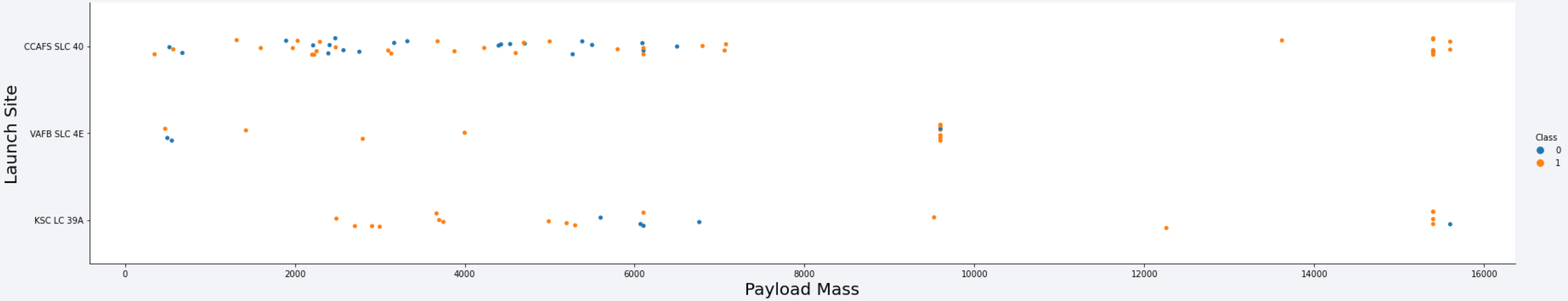


Explanation:

- Launch Site VAFB SLC-4E has the highest success rate (10/13) and the last five flights have all been success.
- All flights after 77 have been a success.
- CCAFC-SLC-40 has the highest number of launches with steadily improving success rate.

Payload vs. Launch Site

Scatter plot of Payload vs. Launch Site

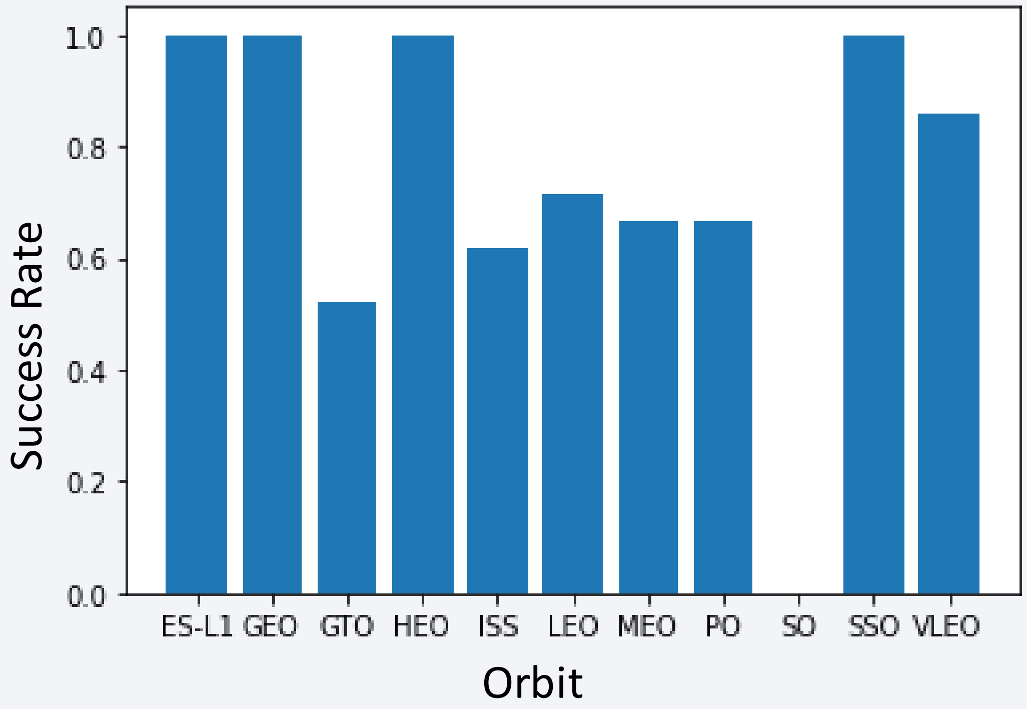


Explanation:

- All flights launched from KSC-LC-39A with a lower payload have a successful outcome.
- Rockets with low payload mass have been mostly launched from site CCAFS-SLC-40.
- Almost all rockets having a higher payload have seen a successful outcome.
- Site VAFB-SLC-4E has the lowest number of launches

Orbit Type vs. Success Rate

Bar chart for the success rate of each orbit type

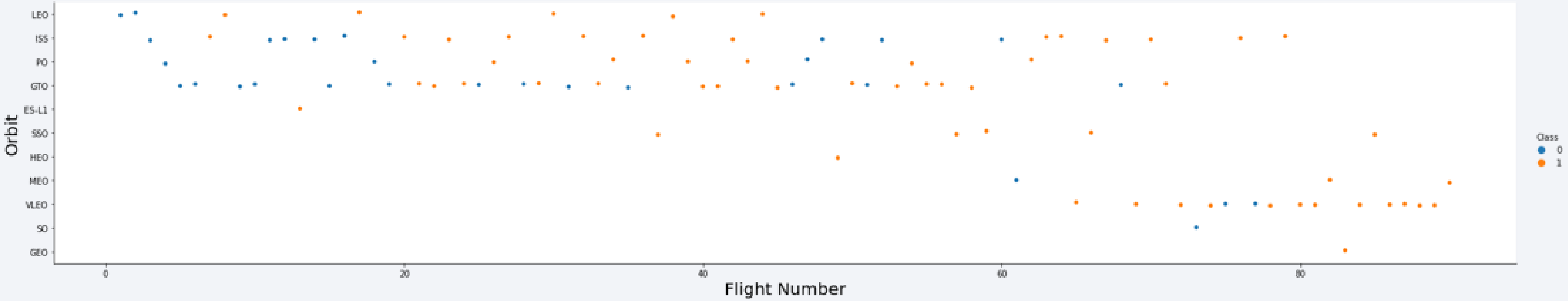


Explanation:

- Orbits ES-L1, GEO, HEO, SSO have the highest success, followed by VLEO close to 90% success.
- Orbit SO is the only orbit which has no successful outcome yet.

Flight Number vs. Orbit Type

Scatter point of Flight number vs. Orbit type

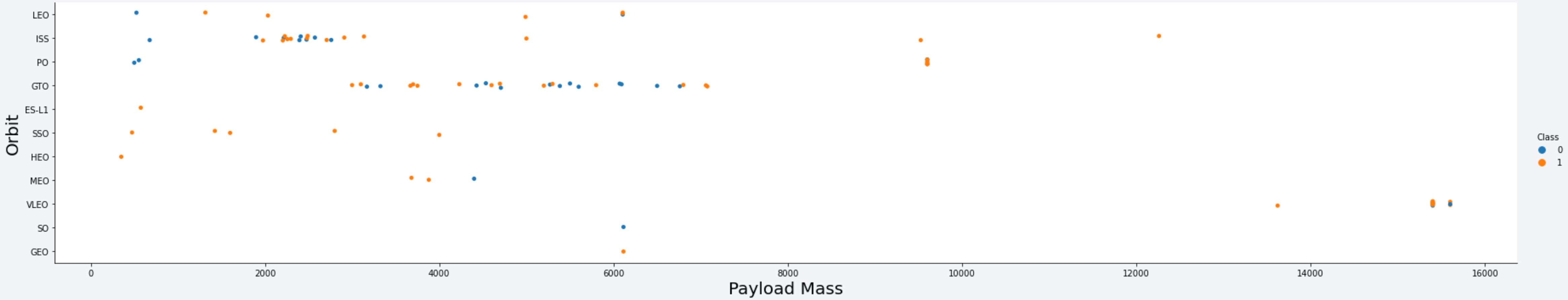


Explanation:

- Orbits LEO, ISS, PO, GTO have the highest number missions.
- Latest missions targeting orbits LEO, ISS, and VLEO have all had a successful outcomes.
- There have been only 1 mission targeting orbits ES-L1, HEO, GEO, SO.

Payload vs. Orbit Type

Scatter point of payload vs. orbit type

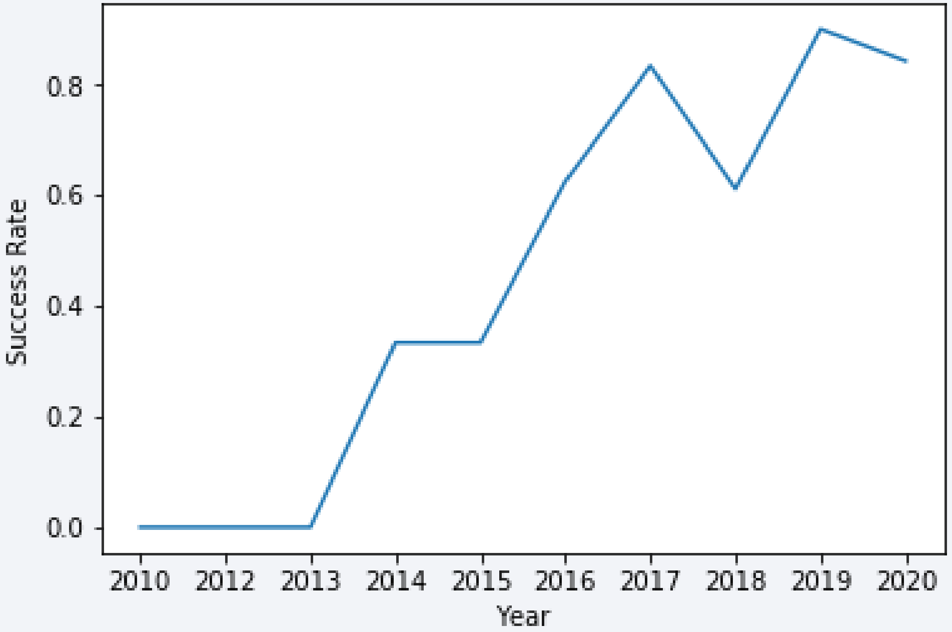


Explanation:

- Higher payload have a significant success rate compared to lower payload missions.
- SSO orbit missions are the only one to have 100% success rate across a payload range.
- Orbits LEO, ISS, and PO have failure outcomes with payload on the lower scale, whereas have a successful outcome with higher payloads.

Launch Success Yearly Trend

Line chart of yearly average success rate

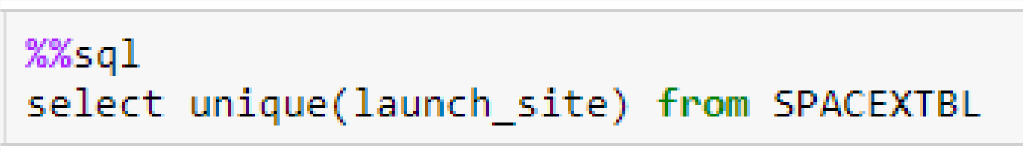


Explanation:

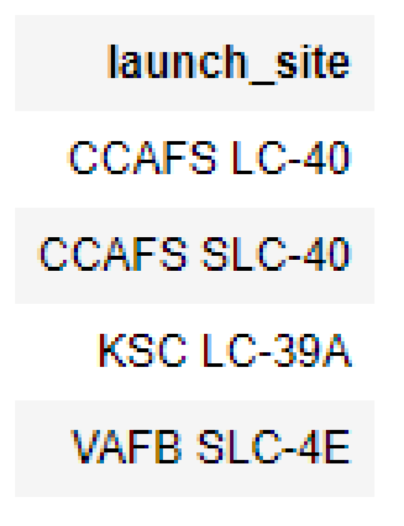
- There is an overall steady increase in the success rate.
- The success increased tremendously after 2015

All Launch Site Names

SQL command for finding the names of unique launch sites



```
%%sql
select unique(launch_site) from SPACEXTBL
```

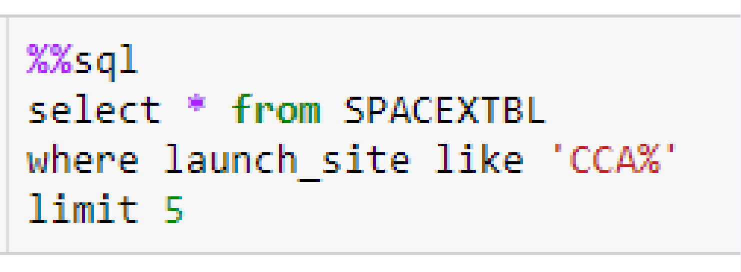


launch_site
CCAFS LC-40
CCAFS SLC-40
KSC LC-39A
VAFB SLC-4E

Query Explanation:
Unique() function selects all the distinct categories

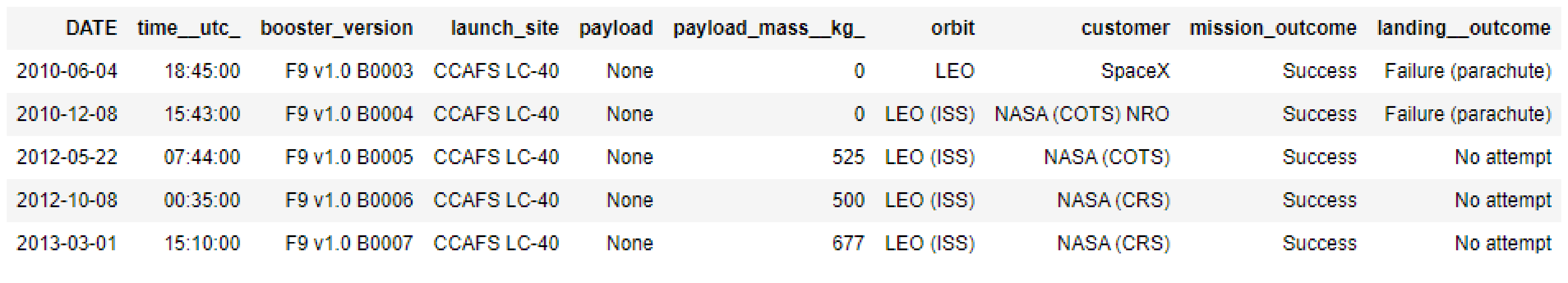
Launch Site Names Begin with 'CCA'

- SQL command to find 5 records where launch sites begin with `CCA`



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

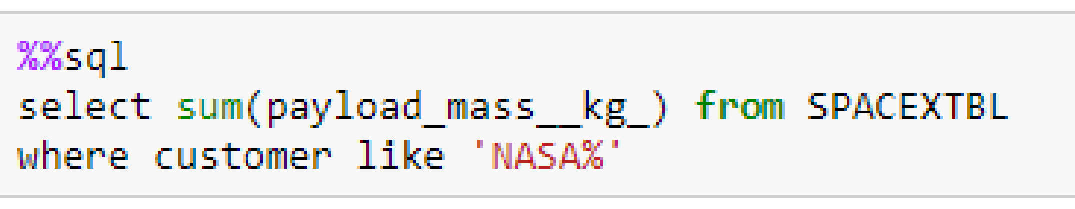
Query Explanation:
'like' command searches for the specified string pattern, in this case 'CCA'.



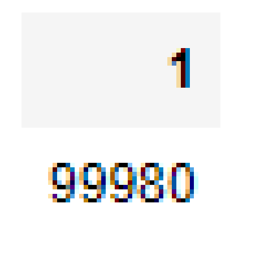
DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	None	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	None	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-05-22	07:44:00	F9 v1.0 B0005	CCAFS LC-40	None	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	00:35:00	F9 v1.0 B0006	CCAFS LC-40	None	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	None	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

- Calculate the total payload carried by boosters from NASA



```
%%sql
select sum(payload_mass__kg_) from SPACEXTBL
where customer like 'NASA%'
```

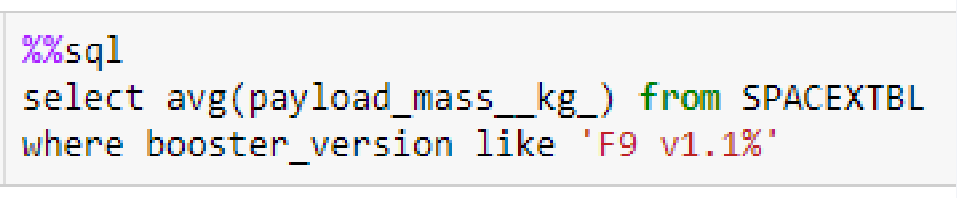


```
    1
99980
```

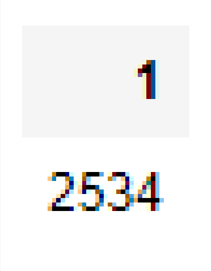
Query Explanation:
Sun() add the values in the selection, and 'like' command searches for the specified string pattern, in this case 'NASA'.

Average Payload Mass by F9 v1.1

- Calculate the 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%'
```

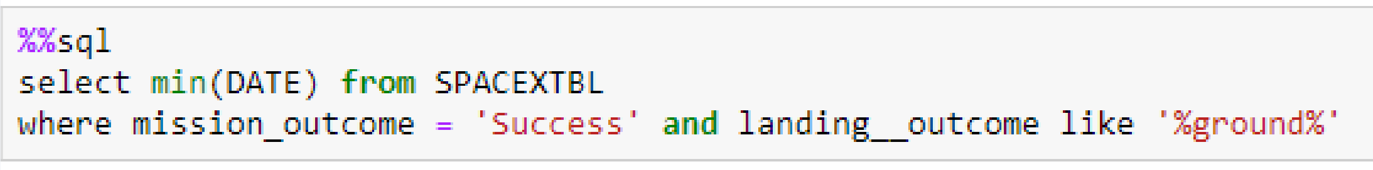


1
2534

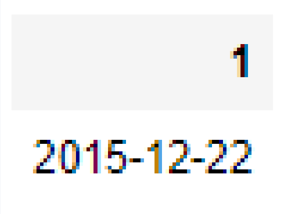
Query Explanation:
Avg() calculates the average value of the selected values, and the comparing the booster string

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad



```
%%sql
select min(DATE) from SPACEXTBL
where mission_outcome = 'Success' and landing__outcome like '%ground%'
```

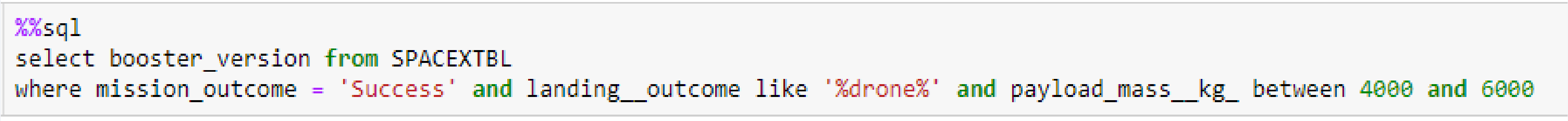


1
2015-12-22

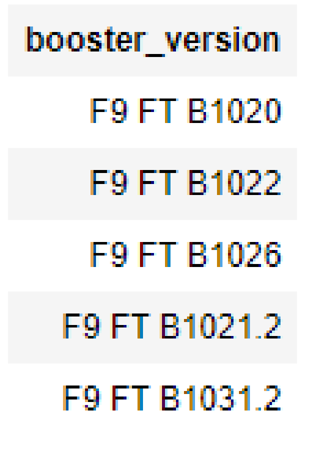
Query Explanation:
Min() selects the minimum value, and the where statement selects the specific cases

Successful Drone Ship Landing with Payload between 4000 and 6000

- List the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000



```
%%sql
select booster_version from SPACEXTBL
where mission_outcome = 'Success' and landing__outcome like '%drone%' and payload_mass__kg_ between 4000 and 6000
```

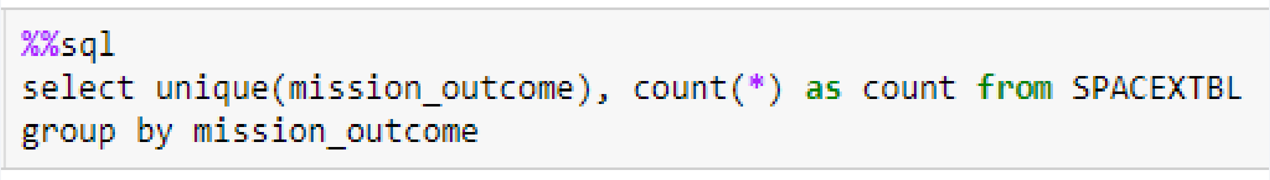


booster_version
F9 FT B1020
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
F9 FT B1031.2

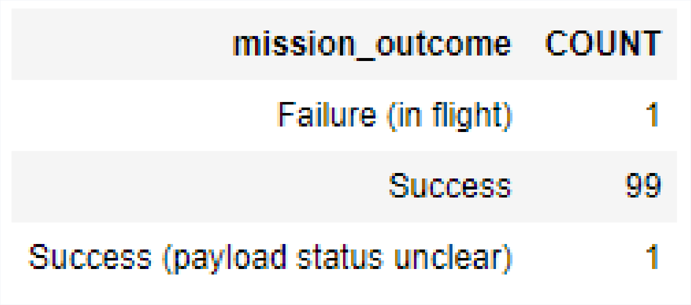
Query Explanation:
Selecting booster_version where certain specific cases are satisfied using the "where" clause.

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes



```
%%sql
select unique(mission_outcome), count(*) as count from SPACEXTBL
group by mission_outcome
```

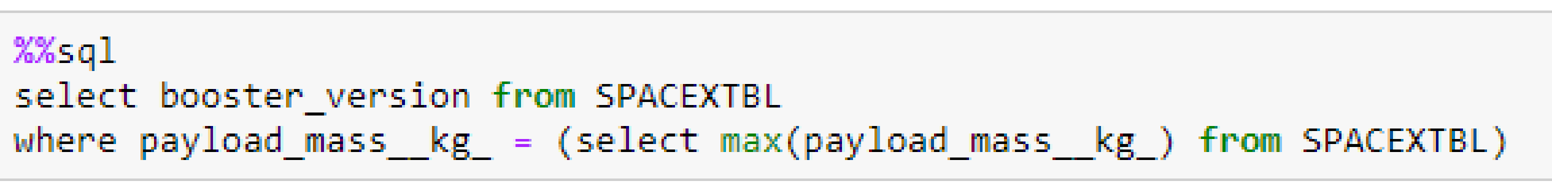


mission_outcome	COUNT
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

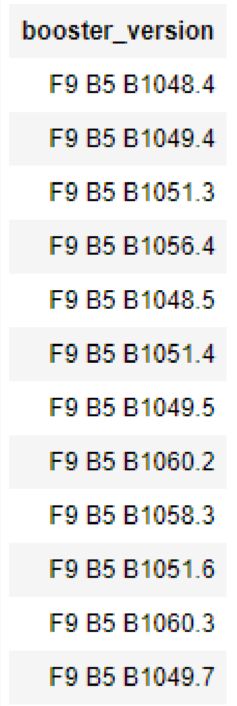
Query Explanation:
Count() function counts the number of occurrence, in this case the occurrences of the mission_outcomes.

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass



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

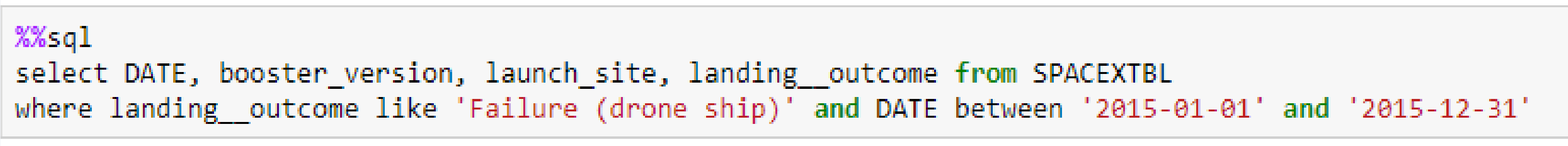


booster_version
F9 B5 B1048.4
F9 B5 B1049.4
F9 B5 B1051.3
F9 B5 B1056.4
F9 B5 B1048.5
F9 B5 B1051.4
F9 B5 B1049.5
F9 B5 B1060.2
F9 B5 B1058.3
F9 B5 B1051.6
F9 B5 B1060.3
F9 B5 B1049.7

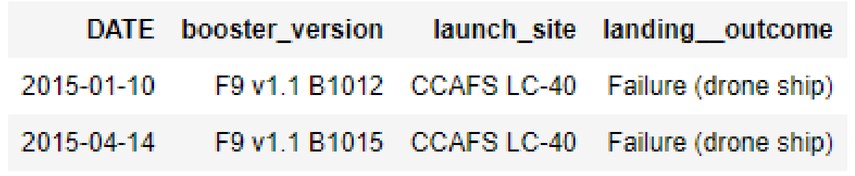
Query Explanation:
A sub-query is used here to select the maximum value of the payload

2015 Launch Records

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



```
%%sql
select DATE, booster_version, launch_site, landing__outcome from SPACEXTBL
where landing__outcome like 'Failure (drone ship)' and DATE between '2015-01-01' and '2015-12-31'
```

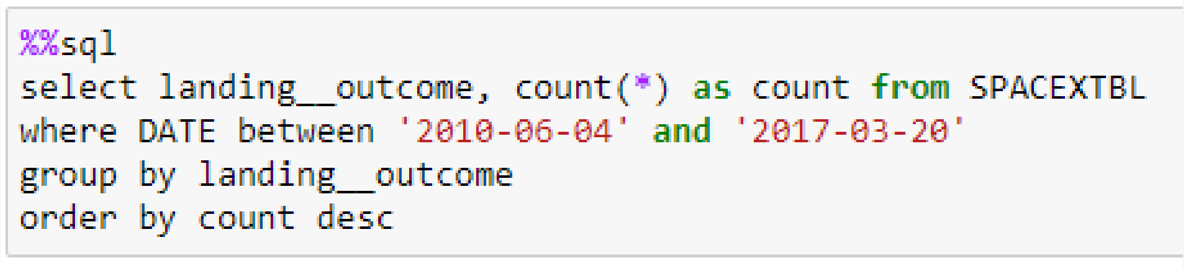


DATE	booster_version	launch_site	landing__outcome
2015-01-10	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
2015-04-14	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

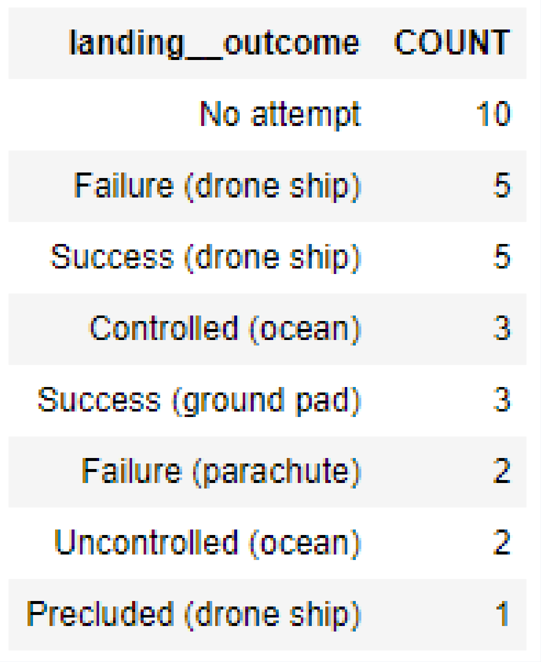
Query Explanation:
Selecting the failed landing outcomes for the specified scenarios

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

- Ranking 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(*) as count from SPACEXTBL
where DATE between '2010-06-04' and '2017-03-20'
group by landing__outcome
order by count desc
```



landing__outcome	COUNT
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

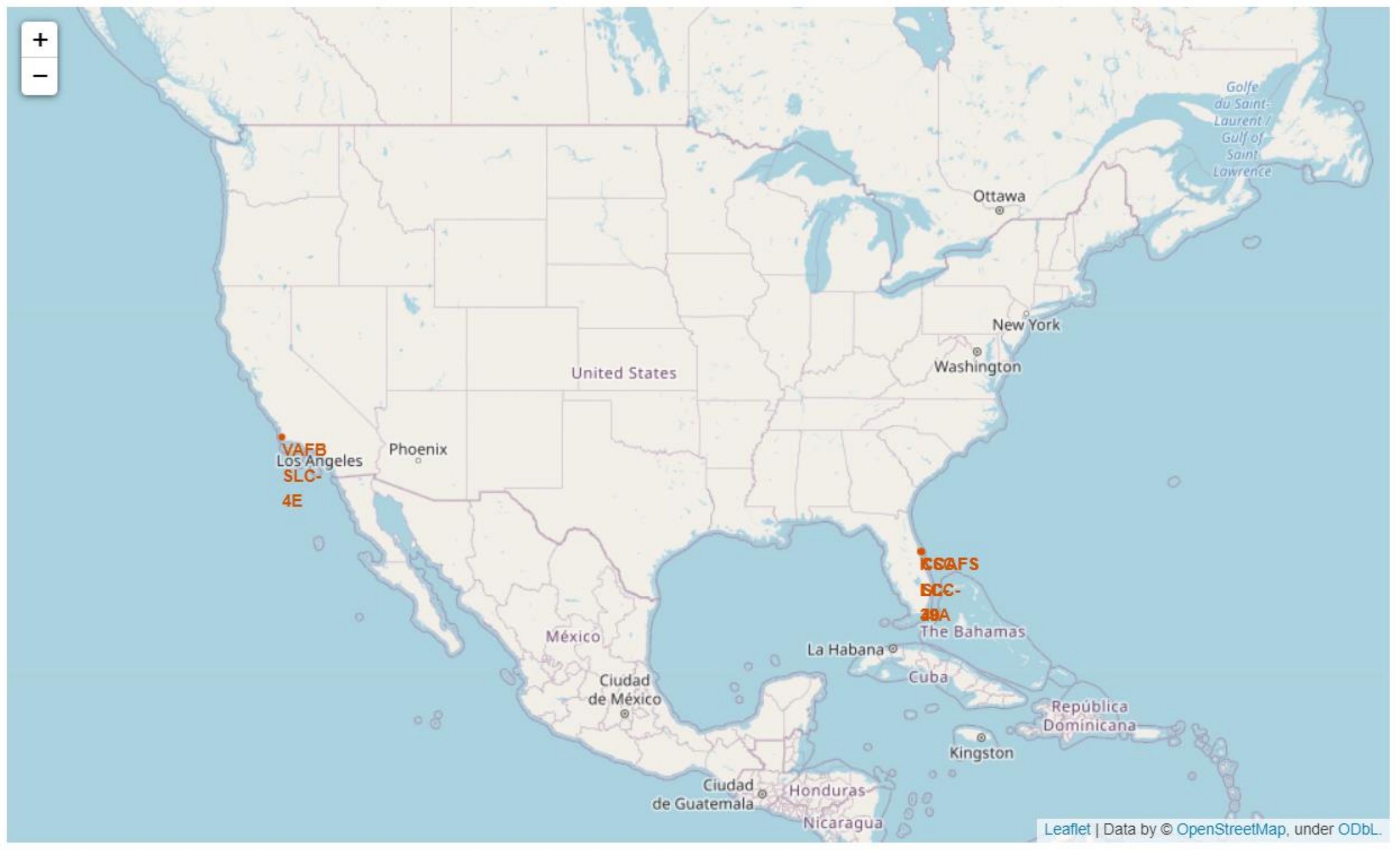
Query Explanation:

Grouping the count of different landing_coutcomes between a range of dates using "group by" function.
"Order by" function orders the attribute in specified order

Launch Sites Proximity Analysis

Locations of the launch sites

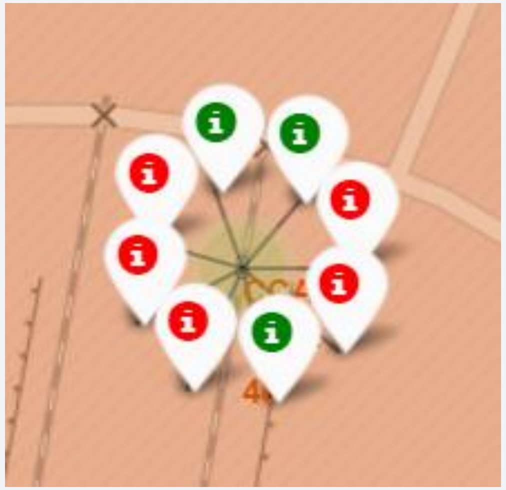
- The launch sites are marked on the world map with a circular marker.
- It can be seen that all the launch sites are located on the coastal sides



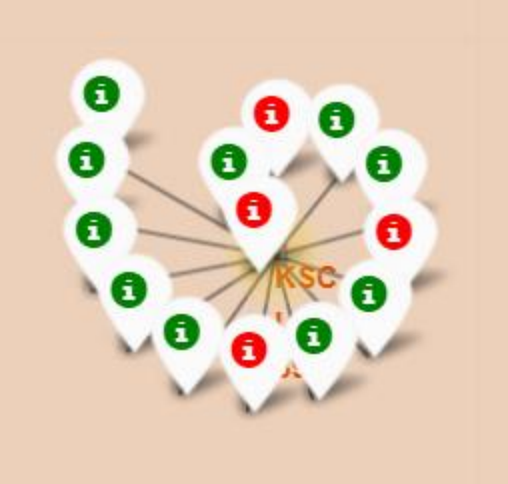
The Success and Failures at launch sites

- A cluster of markers are used here to show multiple launches from a single launch site.
- The "green" color denotes successful outcome and "red" denotes failure outcome.
- It can be seen site "CCAFS LC-40" has the most number of launches, and site "CCAFS SLC-40" has the least.

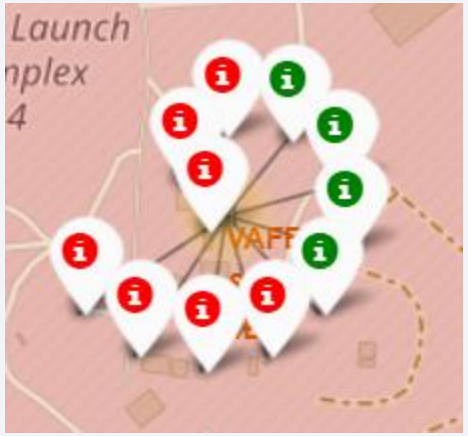
CCAFS SLC-40



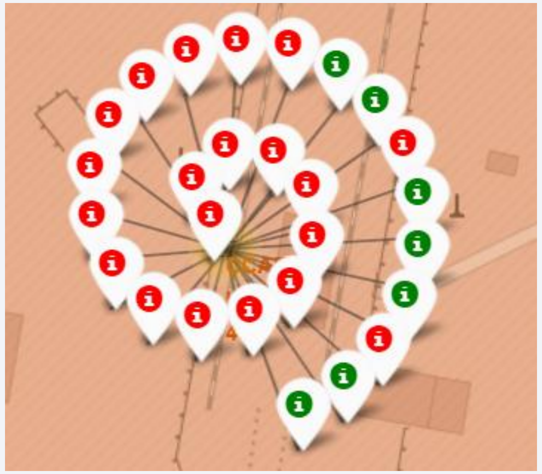
KSC LC-39A



VAFB SLC-4E



CCAFS LC-40



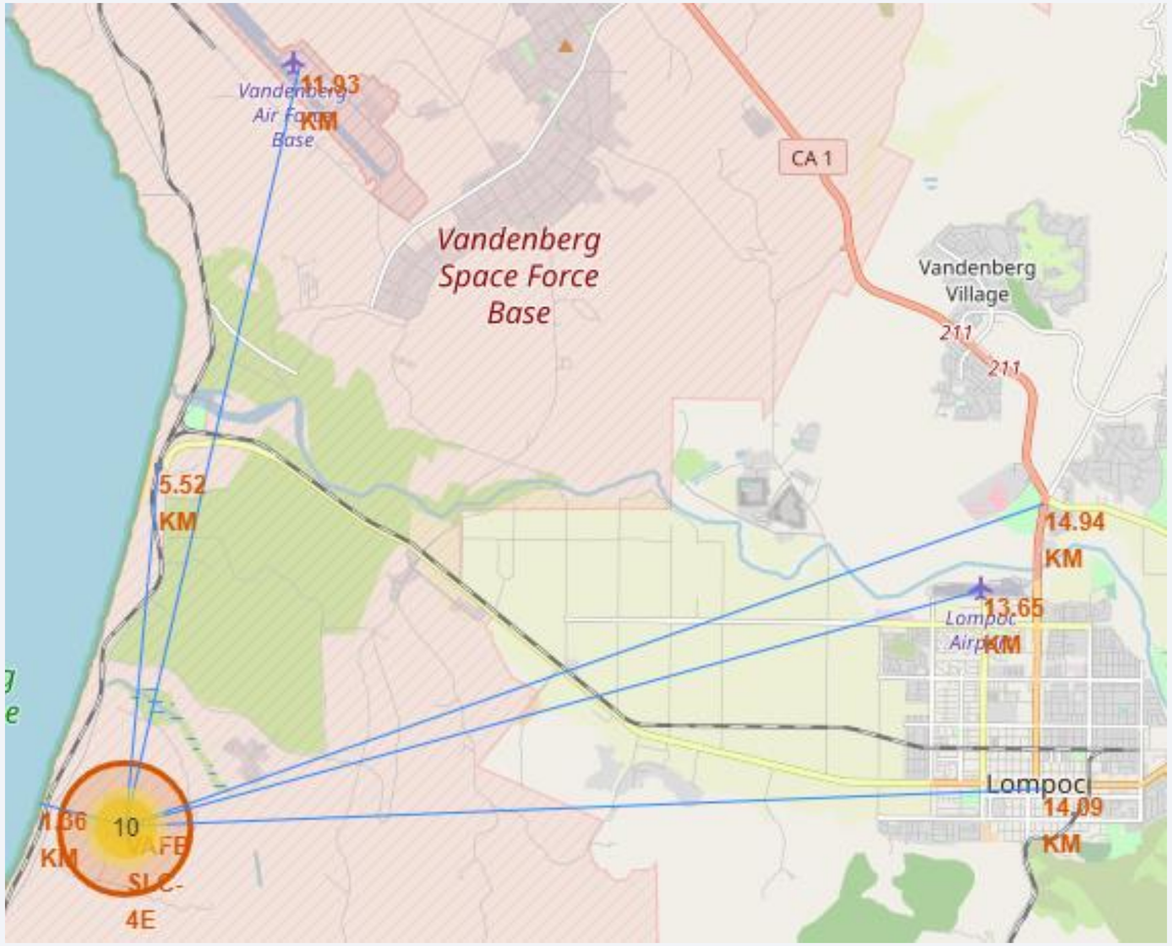
Location proximities from launch site

- Proximities of various landmarks are shown for the launch site "VAFB SLC-4E".

Landmarks:

- ➢Coast
- ➢Nearest City
- ➢Highway
- ➢Railway station
- ➢Nearest Airports

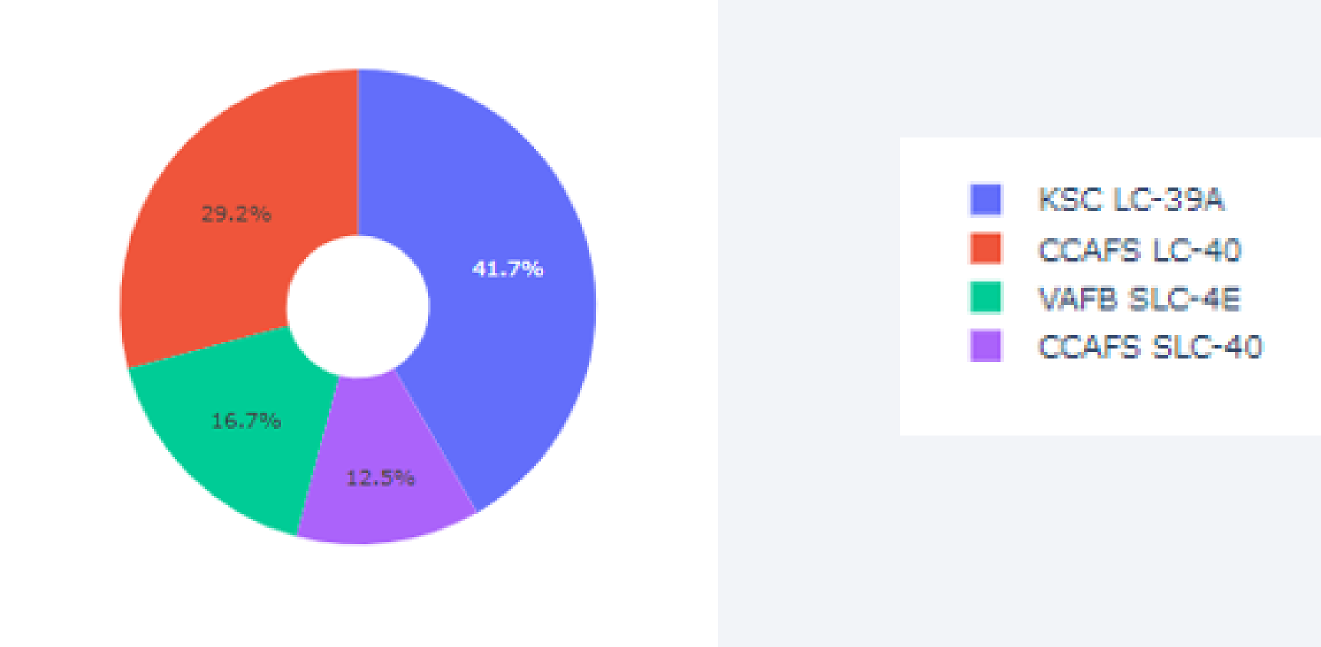
- It can be seen that the launch site is chosen such that it is far away from populated location, and close to coast.



Build a Dashboard with Plotly Dash

Successful launches for all sites

- The successful outcomes are shown for all the sites in a pie chart for comparison.
- It can be seen that KSC LC-39A has the highest success rate with 41.7%, followed by CCAFS LC-40 with 29.2%.

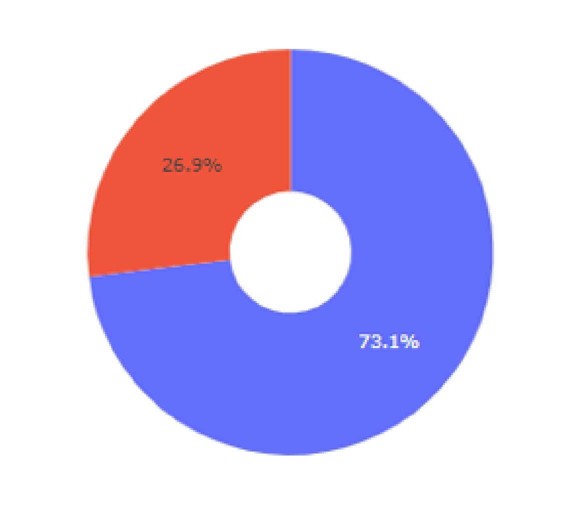


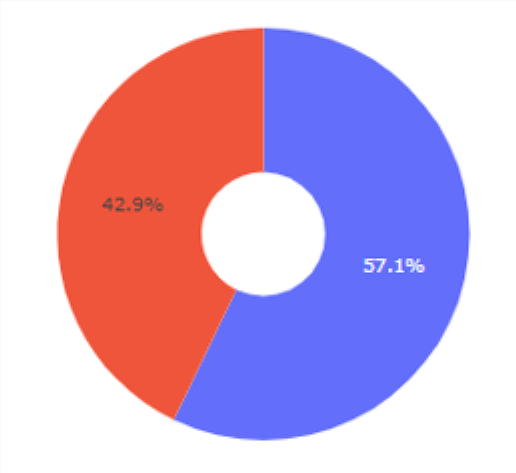
Highest successful launches

Ranking each launch site according to their successful outcomes

1	26.9%
0	73.1%

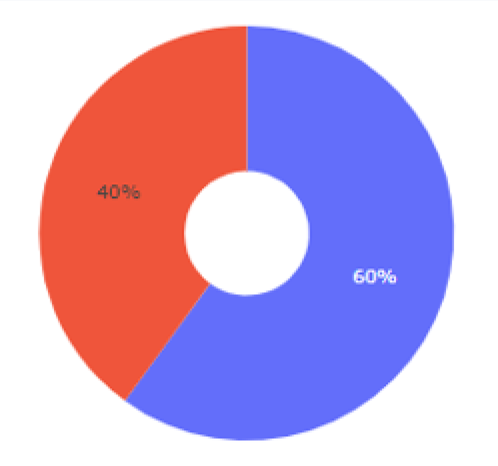
CCAFS LC-40
[3]





CCAFS SLC-40
[1]

1	42.9%
0	57.1%



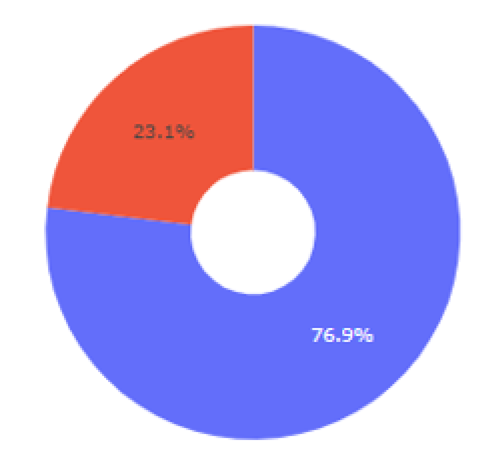
VAFB SLC-4E
[2]

1	40%
0	60%

🟦 0
🟥 1

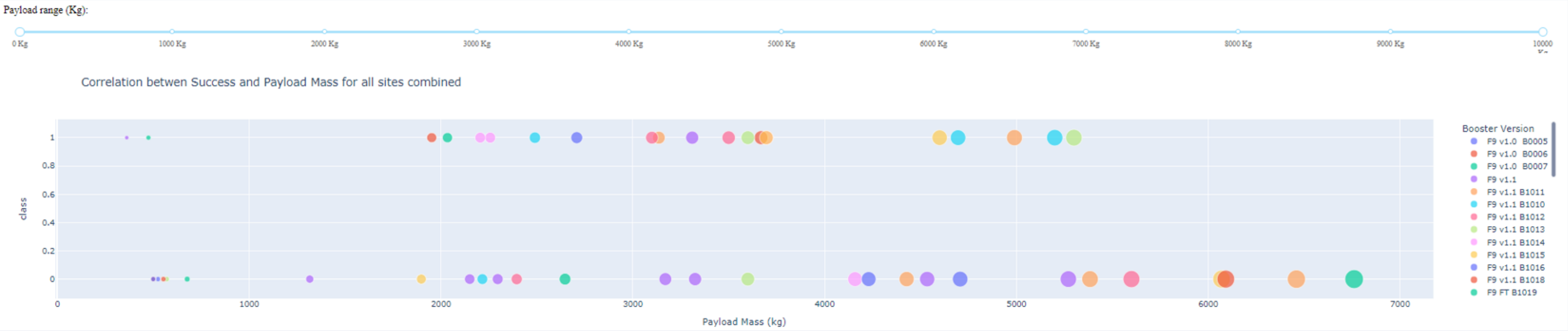
1	23.1%
0	76.9%

KSC LC-39A
[4]

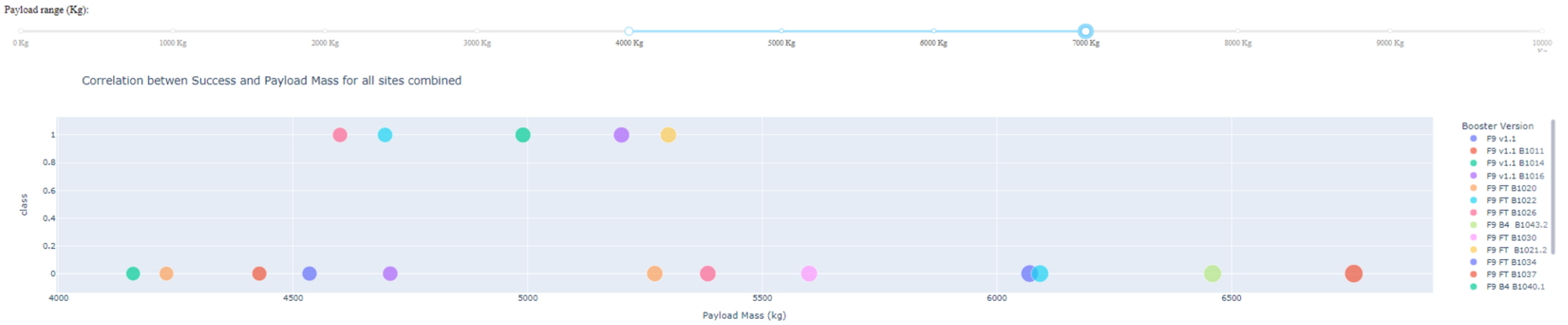


Scatter plot for different payloads

Scatter plot showing launches for all payload types

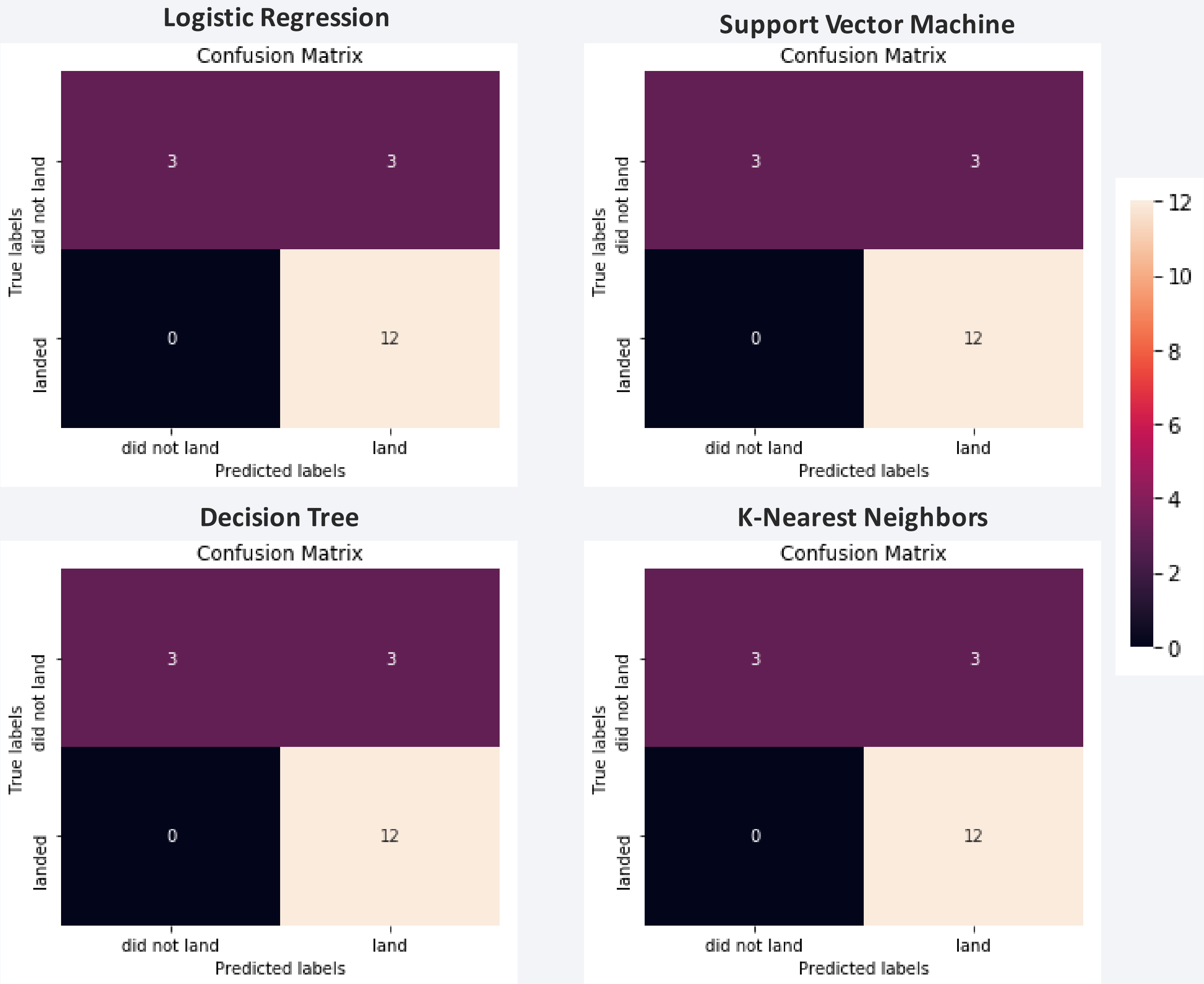


Scatter plot showing launches for payload in range 4000Kg and 7000Kg



Prediction Analysis [Classification]

Confusion Matrix



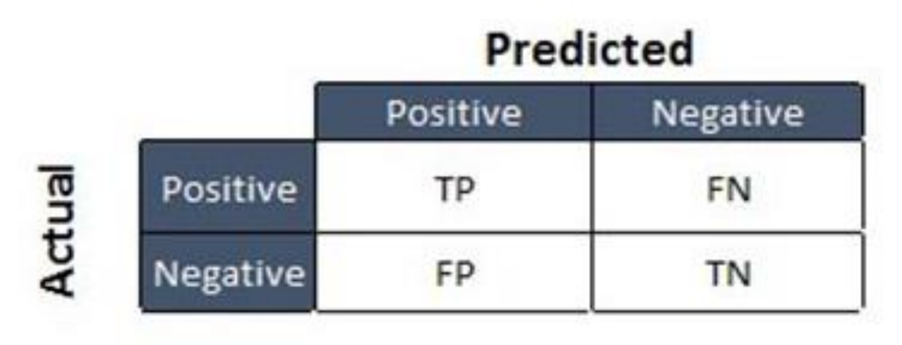
Logistic Regression

Support Vector Machine

Decision Tree

K-Nearest Neighbors

Confusion matrix labels:



		Predicted	
		Positive	Negative
Actual	Positive	TP	FN
	Negative	FP	TN

TP – True Positive

FN – False Negative

FP – False Positive

TN – True Negative

Accuracy Scores

The accuracy and test accuracy scores for all the algorithms are shown in the following table:

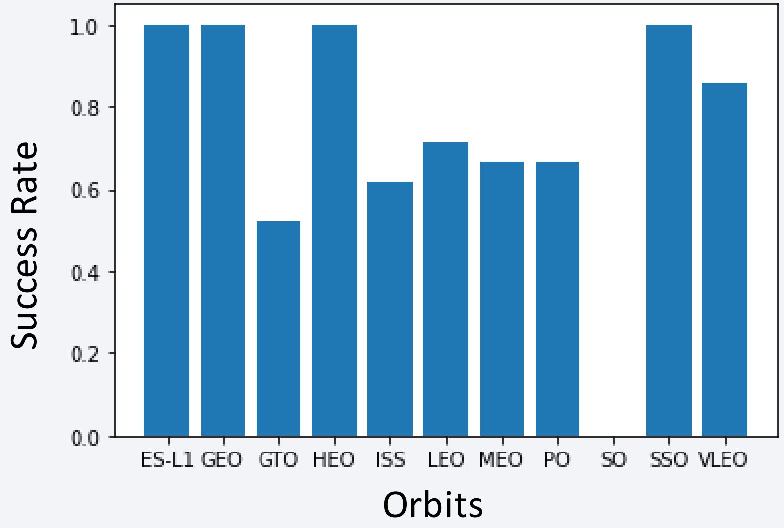
Algorithm	Accuracy	Test set score
Logistic Regression	0.8476	0.8333
Support Vector Machine	0.8482	0.8333
Decision Tree	0.8875	0.8333
K-Nearest Neighbors	0.8472	0.8333

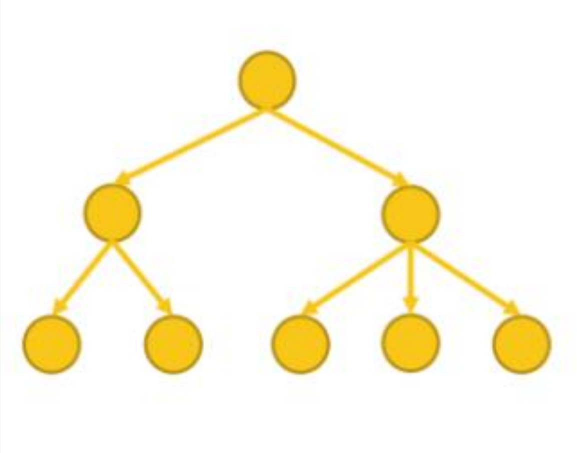
The best parameters found through GridSearch:

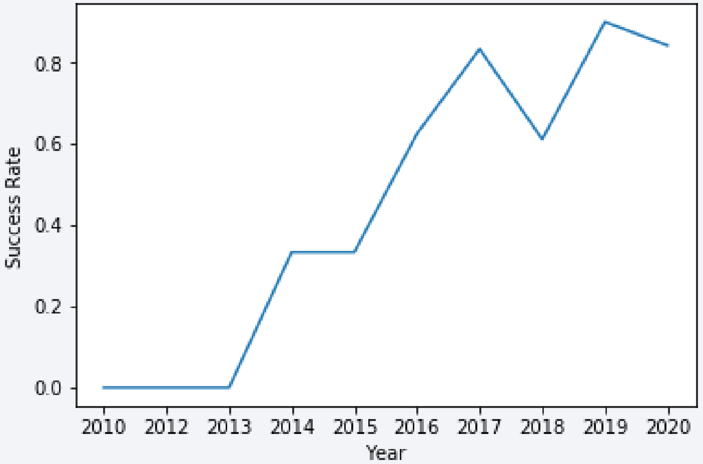
Algorithm	Best Parameters
Logistic Regression	c= 0.01
Support Vector Machine	c= 1.0, gamma = '0.0316', kernel = 'sigmoid'
Decision tree	criterion = 'entropy, max_depth = 6, max_features='sqrt', min_samples_leaf = 2, min_samples_split = 5, splitter = 'random,
K-Nearest Neighbors	algorithm = 'auto', n_neighbors = 10, p = 1

Conclusions

- The orbits ES-L1, GEO, HEO, SSO have the highest success rate.
- The overall success has been increasing since 2015.
- Decision Tree classifier performs the best in predicting a successful landing







Thank you!