IBM Developer
SKILLS NETWORK

# Winning Space Race with Data Science

<Name>
<Date>

# Outline

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

# Executive Summary

- Summary of methodologies
- Summary of all results

# Introduction

- Project background and context
- Problems you want to find answers

Section 1

# Methodology

# Methodology

## Executive Summary

Data collection methodology:

- Data was collected through a get request to the SpaceX Rest API
  - Data was scraped from the Falcon 9 historical launch records from wikipedia

Perform data wrangling

- Filtered data for only Falcon 9 rockets and missing values were replaced with mean

Perform exploratory data analysis (EDA) using visualization and SQL

- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
- Built multiple models and evaluated them based on accuracy and visualization

# Data Collection

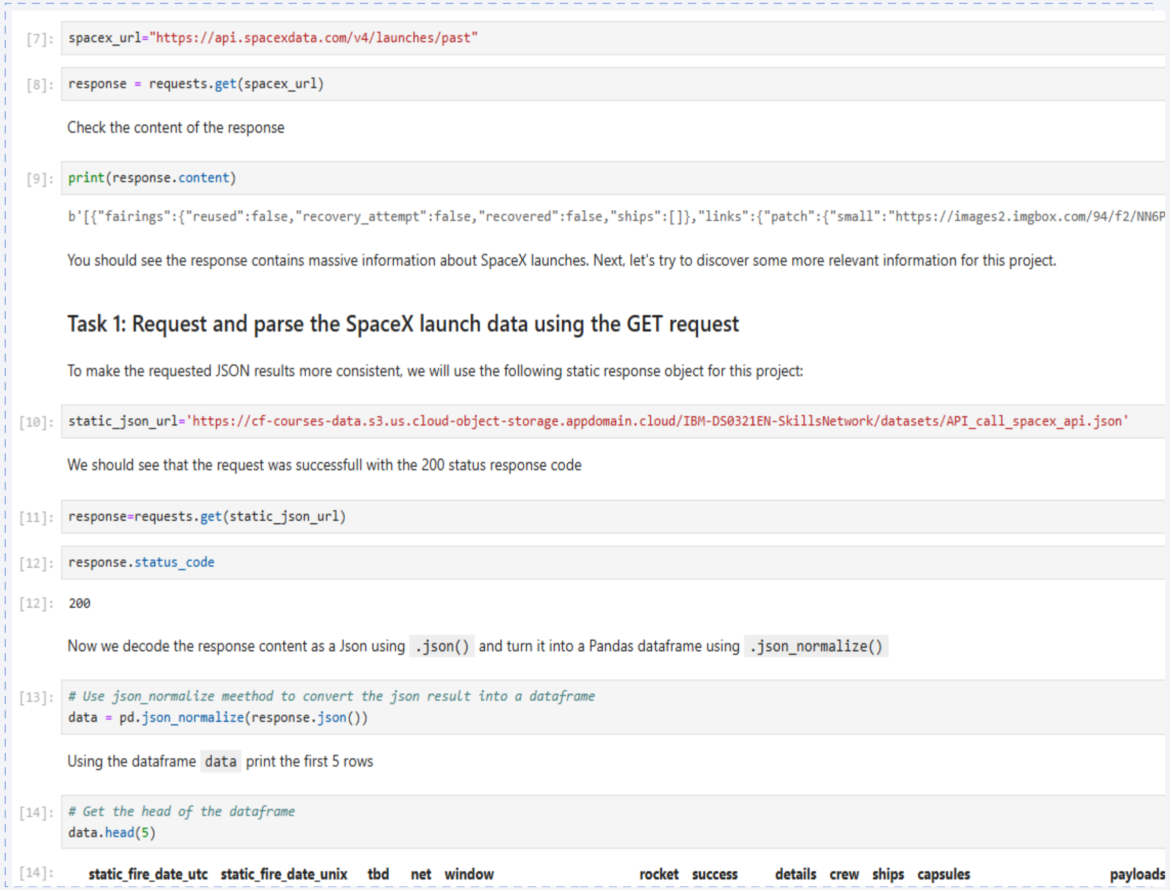
Data collection was done by two methods:

1. Collected data through the SpaceX Rest API using get requests to the following API:
   - https://api.spacexdata.com/v4/rockets/
2. Collected data by web scraping the Wikipedia page for historical Falcon 9 launches.
   - https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches

# Data Collection – SpaceX API

- A get request was used on the SpaceX Rest API to retrieve data and decode it into a json format. This was then transformed into a pandas dataframe for easier manipulation and analysis.

- https://github.com/jdowling23/Applied-Data-Science-SpaceX-Launch/blob/main/jupyter-labs-spacex-data-collection-api.ipynb



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

Check the content of the response

```
[9]: print(response.content)
```

b'[{"fairings":{"reused":false,"recovery_attempt":false,"recovered":false,"ships":[]},"links":{"patch":{"small":"https://images2.imgbox.com/94/f2/NN6P

You should see the response contains massive information about SpaceX launches. Next, let's try to discover some more relevant information for this project.

### Task 1: Request and parse the SpaceX launch data using the GET request

To make the requested JSON results more consistent, we will use the following static response object for this project:

```
[10]: static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_call_spacex_api.json'
```

We should see that the request was successfull with the 200 status response code

```
[11]: response=requests.get(static_json_url)
[12]: response.status_code
[12]: 200
```

Now we decode the response content as a Json using `.json()` and turn it into a Pandas dataframe using `.json_normalize()`

```
[13]: # Use json_normalize meethod to convert the json result into a dataframe
      data = pd.json_normalize(response.json())
```

Using the dataframe `data` print the first 5 rows

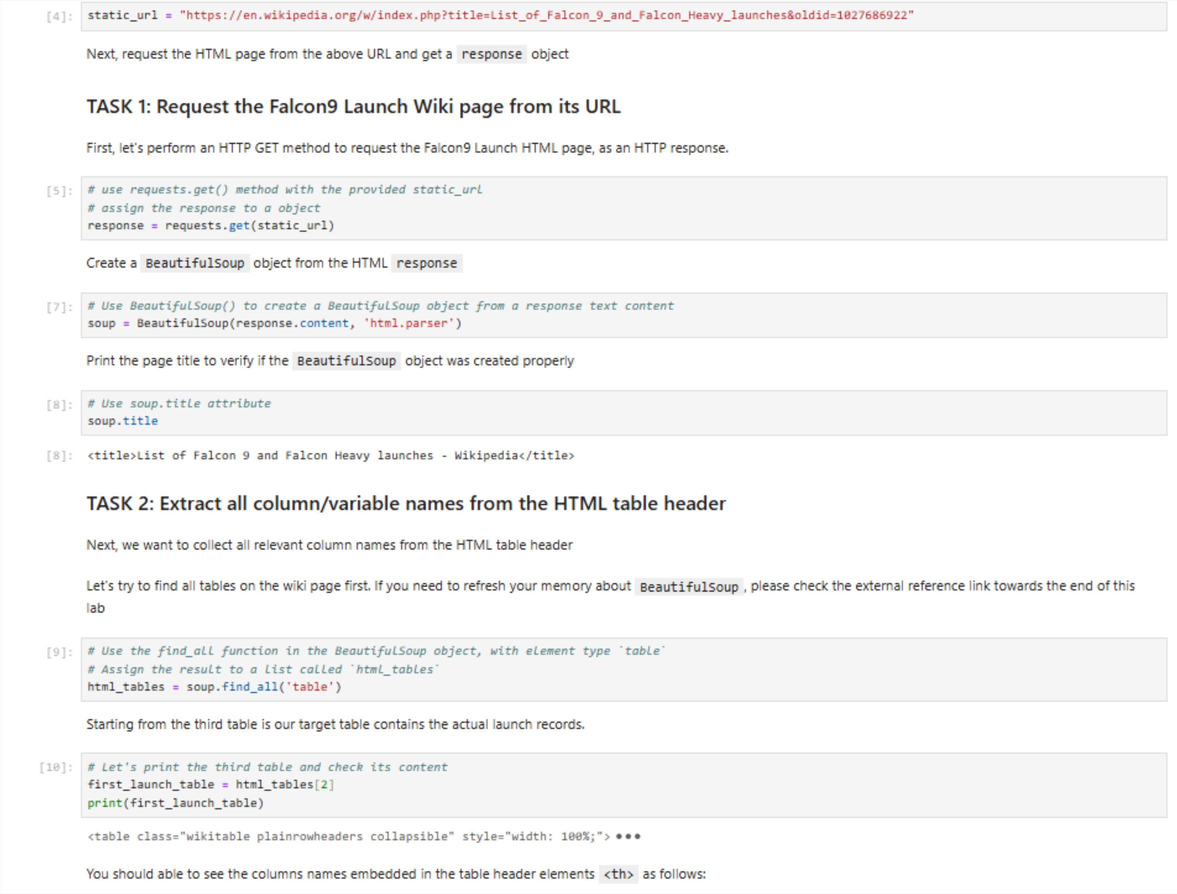
```
[14]: # Get the head of the dataframe
      data.head(5)
```

| [14]: | static_fire_date_utc | static_fire_date_unix | tbd | net | window | rocket | success | details | crew | ships | capsules | payloads |
|---|---|---|---|---|---|---|---|---|---|---|---|---|

# Data Collection - Scraping

- Performed web scraping on the historical falcon 9 launches from the table on the wikipedia page. This was done using requests and BeautifulSoup libraries. Then created a dataframe from the html table.

- https://github.com/jdowling23/Applied-Data-Science-SpaceX-Launch/blob/main/jupyter-labs-webscraping.ipynb



[4]:
```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"
```

Next, request the HTML page from the above URL and get a `response` object

### TASK 1: Request the Falcon9 Launch Wiki page from its URL

First, let's perform an HTTP GET method to request the Falcon9 Launch HTML page, as an HTTP response.

[5]:
```
# use requests.get() method with the provided static_url
# assign the response to a object
response = requests.get(static_url)
```

Create a `BeautifulSoup` object from the HTML `response`

[7]:
```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(response.content, 'html.parser')
```

Print the page title to verify if the `BeautifulSoup` object was created properly

[8]:
```
# Use soup.title attribute
soup.title
```

[8]:
```
<title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>
```

### TASK 2: Extract all column/variable names from the HTML table header

Next, we want to collect all relevant column names from the HTML table header

Let's try to find all tables on the wiki page first. If you need to refresh your memory about `BeautifulSoup`, please check the external reference link towards the end of this lab

[9]:
```
# Use the find_all function in the BeautifulSoup object, with element type `table`
# Assign the result to a list called `html_tables`
html_tables = soup.find_all('table')
```

Starting from the third table is our target table contains the actual launch records.

[10]:
```
# Let's print the third table and check its content
first_launch_table = html_tables[2]
print(first_launch_table)
```

```
<table class="wikitable plainrowheaders collapsible" style="width: 100%;"> •••
```

You should able to see the columns names embedded in the table header elements `<th>` as follows:

# Data Wrangling

After the data collection process was complete and data was in a pandas dataframe, it was filtered on the ‘booster-version’ to get only the falcon 9 rocket data.

The Payload Mass column had missing values so the mean value of that column was used to replace them.

https://github.com/jdowling23/Applied-Data-Science-SpaceX-Launch/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

# EDA with Data Visualization

Scatter plots were used to visualize data relationships

- Launch Site vs Flight Number
- Launch Site vs Payload Mass
- Orbit vs Flight Number
- Orbit vs Payload Mass

Bar Chart was used to visualize Success vs Orbit type

Line Chart was used to visualize success rate vs year

https://github.com/jdowling23/Applied-Data-Science-SpaceX-Launch/blob/main/EDA_feature_engineering.ipynb

# EDA with SQL

- **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 acheived**
- **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 all the booster_versions that have carried the maximum payload mass, using a subquery with a suitable aggregate function.**
- **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 landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order.**

**https://github.com/jdowling23/Applied-Data-Science-SpaceX-Launch/blob/main/SpaceX%20EDA%20using%20SQL.ipynb**

# Build an Interactive Map with Folium

A map was created using Folium to show the locations of launch sites. These launch sites were marked with circles and labeled. Lines were used to visualize distance from nearest coast lines and roads. The launch success/failures were also added to each launch site.

These markers were added to visualize the number of success/failures for each launch site on a map of the US.

https://github.com/jdowling23/Applied-Data-Science-SpaceX-Launch/blob/main/Data%20Analysis%20of%20Launch%20Sites%20with%20Folium.ipynb

# Interactive Map with Folium

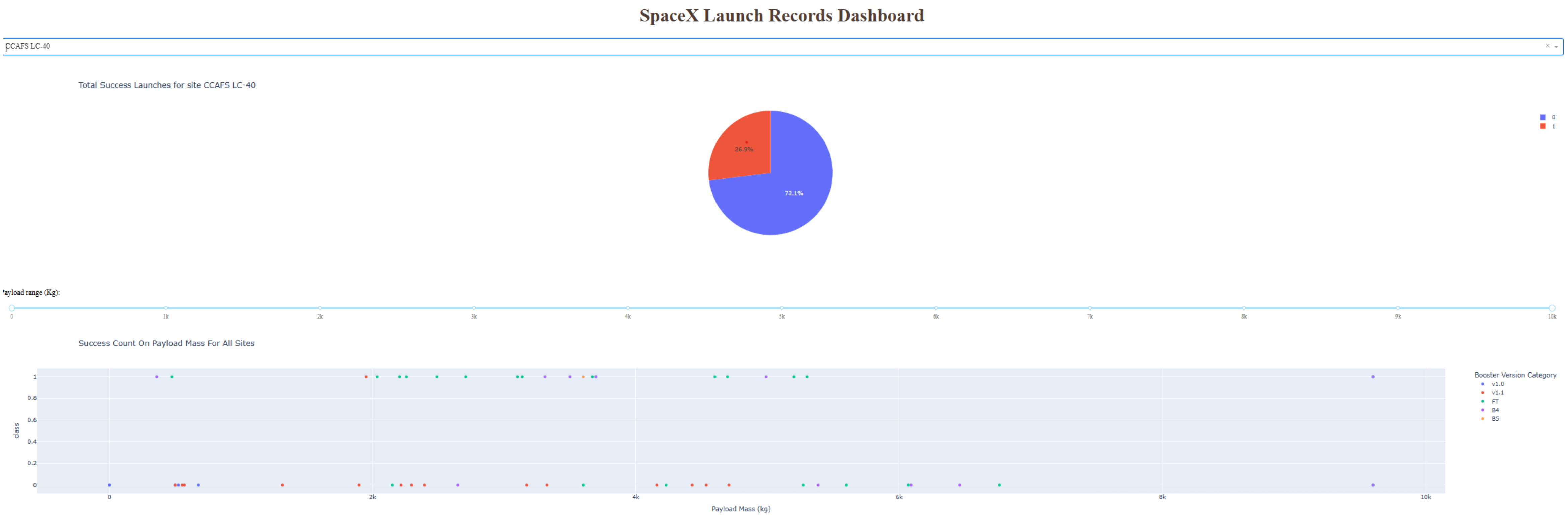
# Build a Dashboard with Plotly Dash

A plotly dashboard was created to interact with. A drop down menu contains the different launch sites where when chosen a pie chart and scatter plot as updated with the data for that particular locations launch success and failure.

This dashboard is user friendly and easy to interact with while providing informative and easy to read data points about the launch success rate for each location.

https://github.com/jdowling23/Applied-Data-Science-SpaceX-Launch/tree/main/SpaceX-Dashboard

# Dashboard with Plotly Dash



# Predictive Analysis (Classification)

Models used for predictive analysis

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

Used train_test_split function to gather training and testing data for models above. Measured the accuracy score and visualized accuracy in a confusion matrix.

https://github.com/jdowling23/Applied-Data-Science-SpaceX-Launch/blob/main/SpaceX_Machine%20Learning%20Prediction.ipynb

# Predictive Analysis (Classification)

Comparing the accuracy of the different models:

[42]:

| | 0 |
|---|---|
| **Method** | Test Data Accuracy |
| **Logistic_Reg** | 0.833333 |
| **SVM** | 0.833333 |
| **Decision Tree** | 0.888889 |
| **KNN** | 0.833333 |

# Results

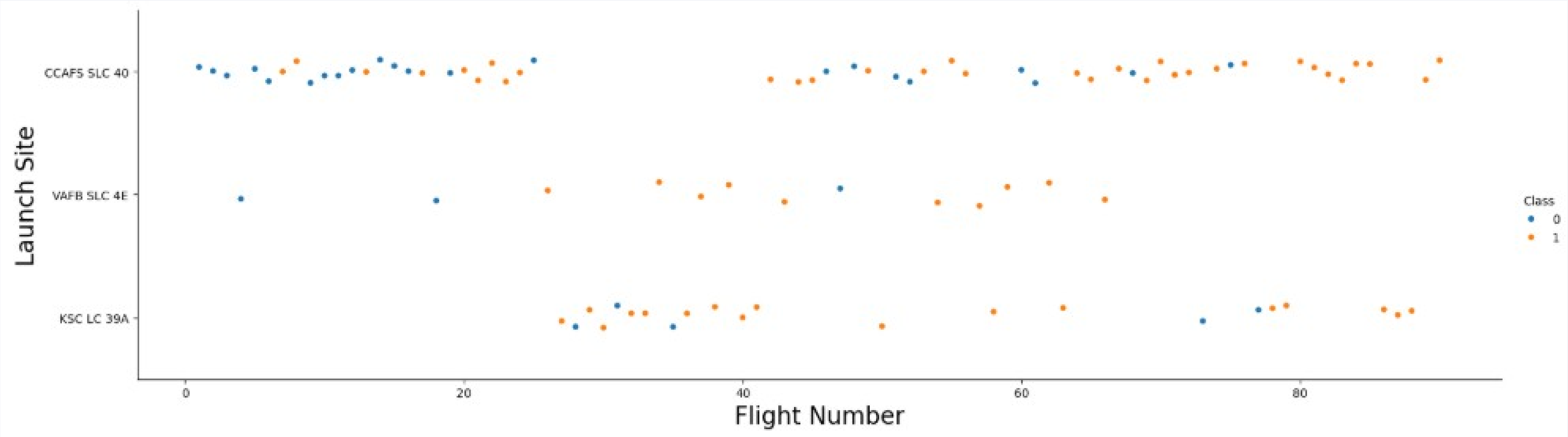
- Exploratory data analysis results
  - Success rate has increased over time since 2014
  - Heavy payloads are more successful for Polar, LEO and ISS orbits
- Interactive analytics demo in screenshots
- Predictive analysis results
  - The decision tree provided the most accurate score among the models used, at 88% when predicting launch success rates.

Section 2

# Insights drawn from EDA

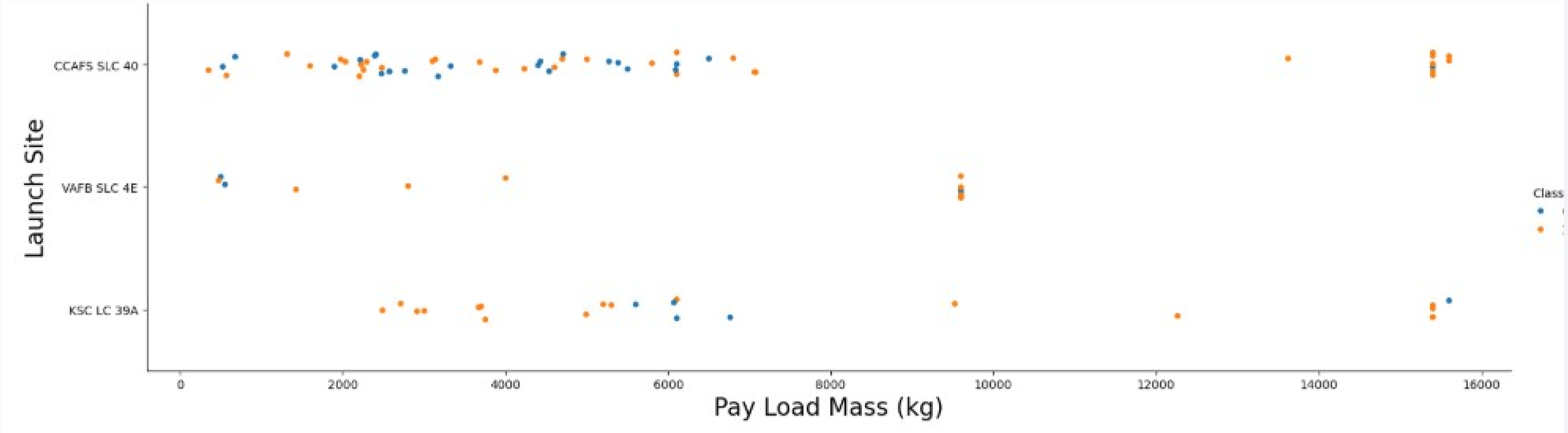
# Flight Number vs. Launch Site

The majority of the flights occurred at the CCA launch site and we saw success rate increase with increase in flight number.



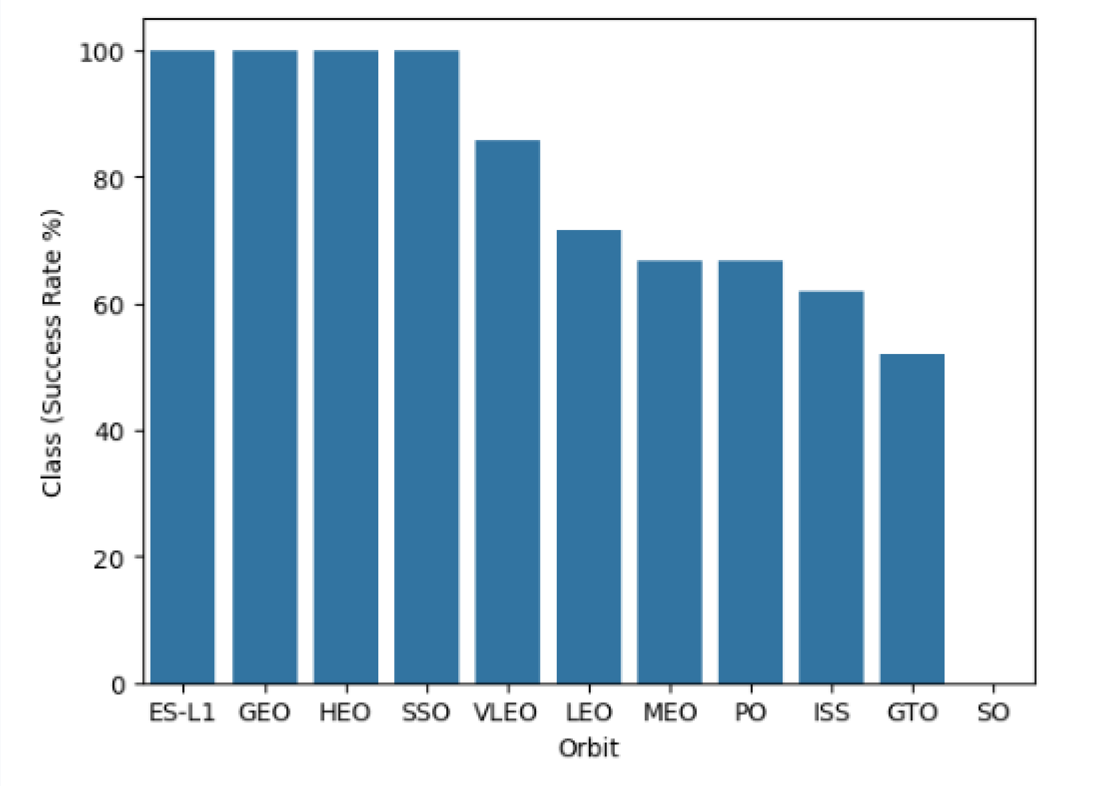
# Payload vs. Launch Site

There were no heavy payloads launched from the VAFB site. While the success rate for heavy payloads at the other 2 sites had a high success rate.



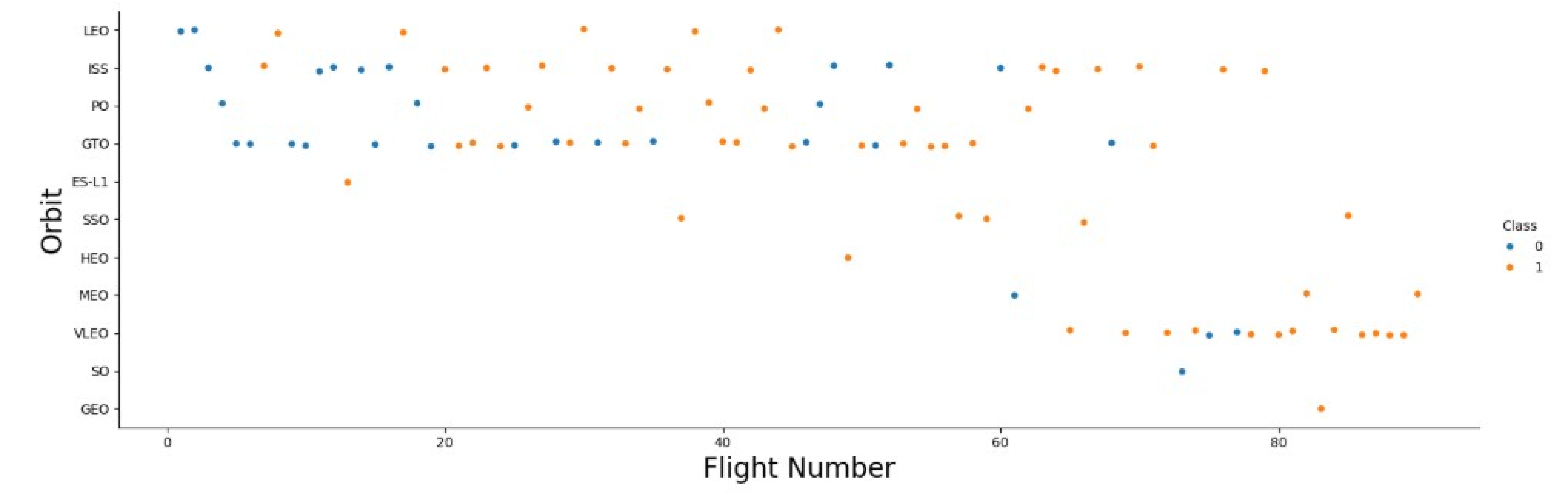
# Success Rate vs. Orbit Type

- Bar chart for the success rate of each orbit type
- ES-L1, GEO, HEO, and SSO orbits had a 100% success rate.
- SO orbits had either zero launches or zero success



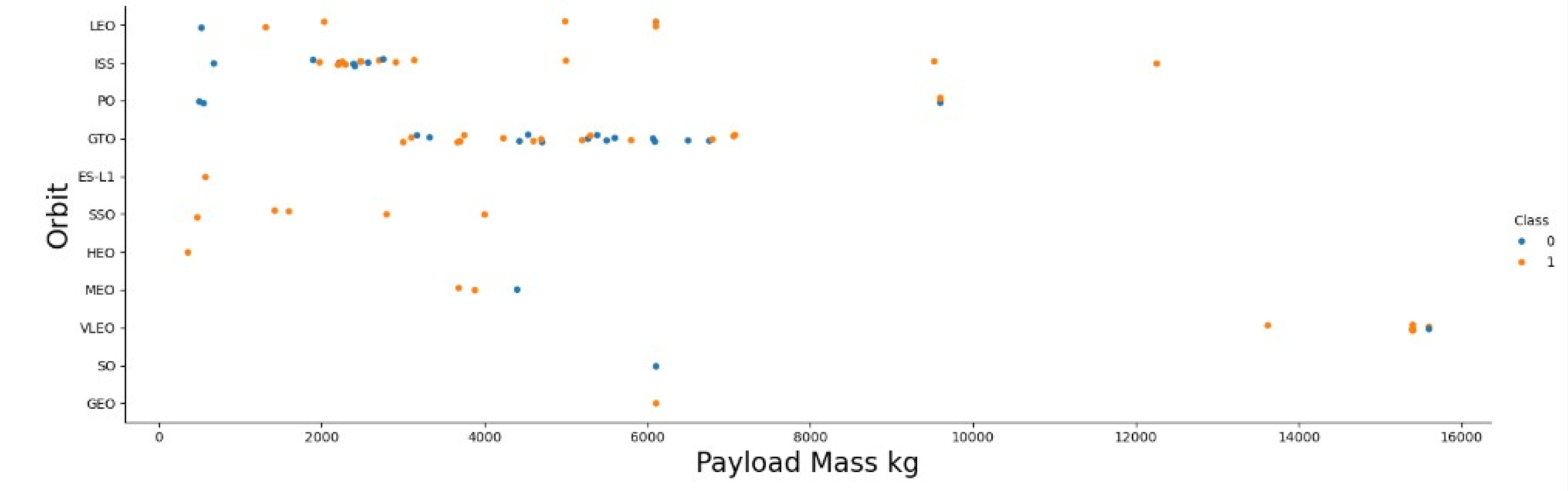
# Flight Number vs. Orbit Type

LEO orbits seem to be more successful with increase in flight number. Where GTO orbits seem to have no correlation.



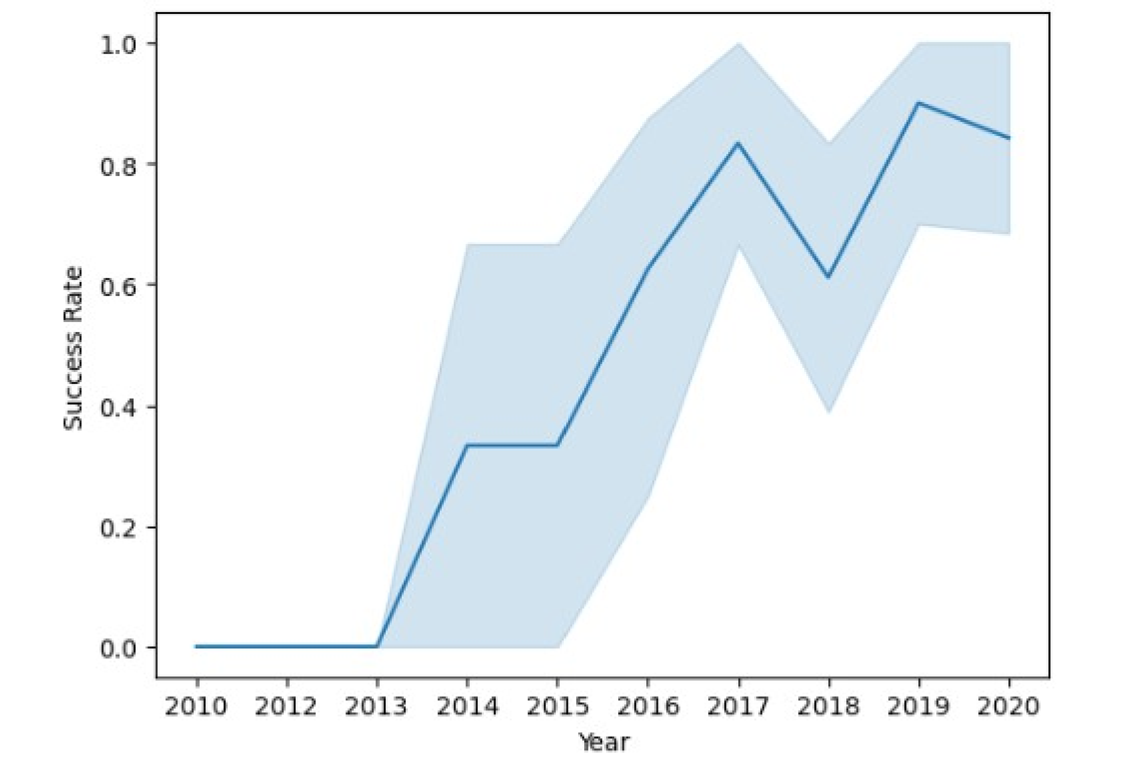
# Payload vs. Orbit Type

- Heavy payloads have a higher success rate for LEO, ISS, and Polar orbits
- GTO orbits seem to be unrelated as both success and failed launches are present



# Launch Success Yearly Trend

- The success rate has continued to increase since 2013 with the exception to a slight dip in 2018.



# All Launch Site Names

Find the names of the unique launch sites

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'

Find 5 records where launch sites begin with `CCA`

Use the ‘like’ keyword and ‘%’ in the string to find sites 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

Calculate the total payload carried by boosters from NASA

Used the ‘sum()’ function to find the total payload mass from all launches.

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

```
%sql select sum(PAYLOAD_MASS__KG_) as payload_mass from SPACEXTABLE;
```

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

| payload_mass |
|---|
| 619967 |

# Average Payload Mass by F9 v1.1

Calculate the average payload mass carried by booster version F9 v1.1

Used the ‘avg()’ function and filtered on ‘Booster_Version’ for desired results of 2928.4 KG

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

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

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

| avg(PAYLOAD_MASS__KG_) |
|---|
| 2928.4 |

# First Successful Ground Landing Date

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

Used the ‘min’ function on the Date column and filtered on the Landing_Outcome column to get the first successful ground landing date.

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

*Hint:Use min function*

```
%sql select min("Date") from SPACEXTABLE where "Landing_Outcome" = "Success (ground pad)";
```

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

| min("Date") |
|---|
| 2015-12-22 |

# 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

Used Distinct to filter Booster_Version and filtered on Landing_Outcome column for “Success (drone ship)” as well as desired payload mass values

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 Distinct("Booster_Version") from SPACEXTABLE where "Landing_Outcome" = "Success (drone ship)" and PAYLOAD_MASS__KG_ > 4000 and PAYLOAD_MASS__
```

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

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

# Total Number of Successful and Failure Mission Outcomes

Calculate the total number of successful and failure mission outcomes

Used count function on Mission_Outcome column and grouped by same column to get counts of success and failed missions.

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

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

| Mission_Outcome | missionoutcomes |
|---|---|
| Failure (in flight) | 1 |
| Success | 98 |
| Success | 1 |
| Success (payload status unclear) | 1 |

# Boosters Carried Maximum Payload

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

Used a subquery for PAYLOAD_MASS__KG_ column and the 'max' aggregate function.

List all the booster_versions that have carried the maximum payload mass, using a subquery with a suitable aggregate function.

```
%%sql select "Booster_Version", PAYLOAD_MASS__KG_ from SPACEXTABLE
where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEXTABLE);
```

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

| Booster_Version | PAYLOAD_MASS__KG_ |
|---|---|
| F9 B5 B1048.4 | 15600 |
| F9 B5 B1049.4 | 15600 |
| F9 B5 B1051.3 | 15600 |
| F9 B5 B1056.4 | 15600 |
| F9 B5 B1048.5 | 15600 |
| F9 B5 B1051.4 | 15600 |
| F9 B5 B1049.5 | 15600 |
| F9 B5 B1060.2 | 15600 |
| F9 B5 B1058.3 | 15600 |
| F9 B5 B1051.6 | 15600 |
| F9 B5 B1060.3 | 15600 |
| F9 B5 B1049.7 | 15600 |

# 2015 Launch Records

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

Used the substr function on Date column to get the required month in the year 2015. Also filtered on Landing_Outcome for “failure (drone ship)”

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 substr(Date, 0,5) = '2015' and "Landing_Outcome" = "Failure (drone ship)";
```

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

| 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

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

Used aggregate count function and filtered between the desired dates. Grouped by Landing_Outcomes column and ordered by date descending.

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" BETWEEN '2010-06-04' AND '2017-03-20'
group by "Landing_Outcomes"
ORDER BY "DATE" DESC;
```

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

| Landing_Outcome | count(*) |
|---|---|
| Failure (parachute) | 31 |

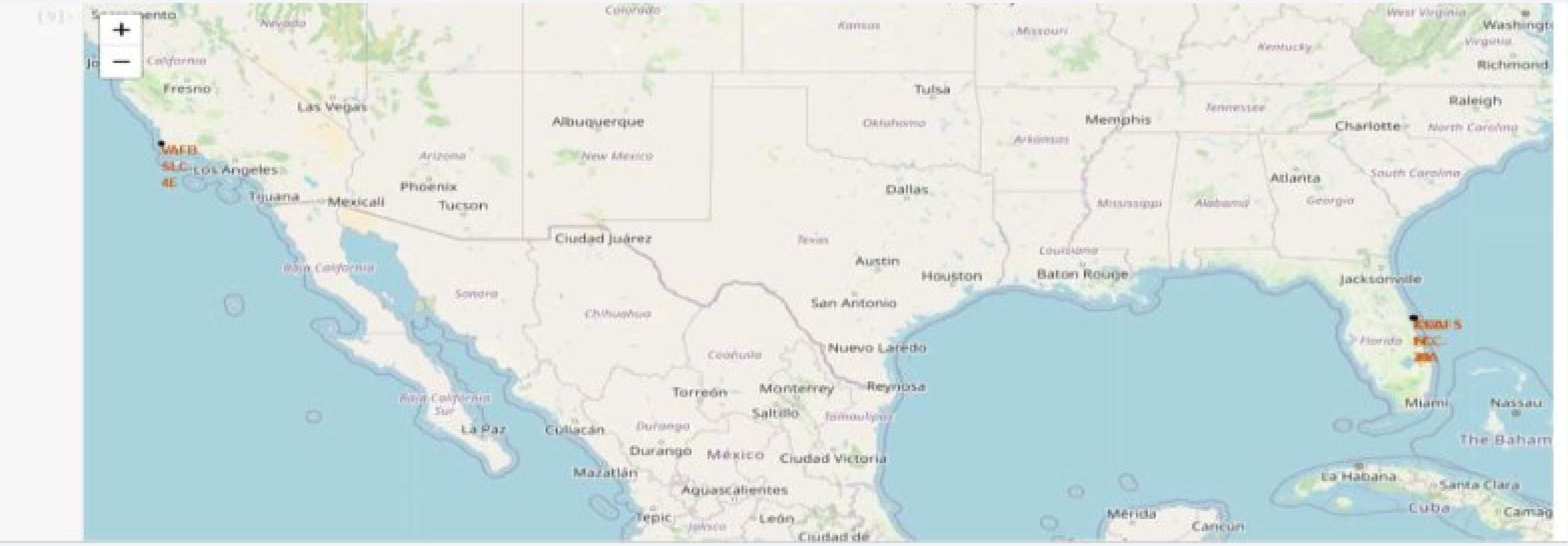
Section 3

# Launch Sites Proximities Analysis

# SpaceX Launch Sites – Folium Map

**There are a total of 4 launch sites shown on this map of the US labeled in red text.**

- **VAFB SLC-4E**
- **KSC-LC29A**
- **CCAFS-LC40**
- **CCAF-SLC40**



# Launch Site Outcomes

Kennedy Space Center (FL) – KSC-LC29A

This shows the successful launch outcomes in green and failed launch outcomes in red for this particular launch site.

10 successful, 3 failed

# Launch Site proximity

This is a Launch Site in Cape Canaveral (CCAFS-SLC40)

This graphic shows the launch site with blue lines demonstrating the distance between the closest highway at .58 KM.

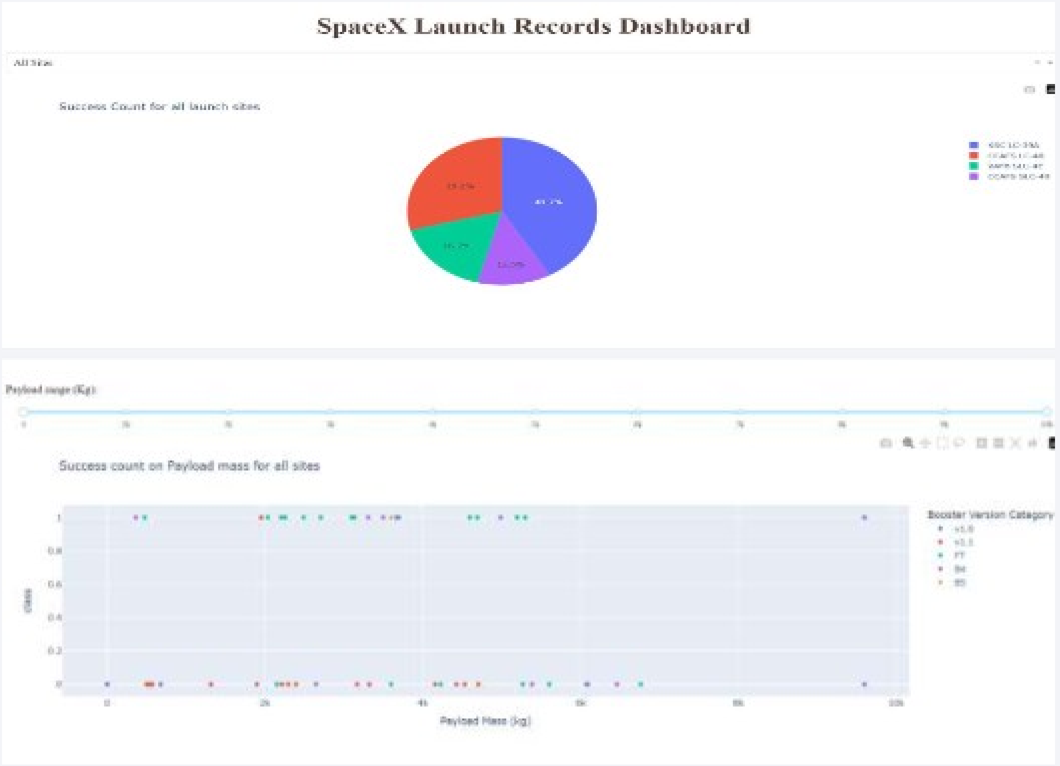
Section 4

# Build a Dashboard with Plotly Dash

# Plotly Dashboard All Sites

This Dashboard has a drop down menu to choose the launch site. As well as a pie chart and scatter plot to demonstrate the data for the chosen launch site.

This dashboard shows the Success rate for all launch sites as default since there is no launch site chosen.



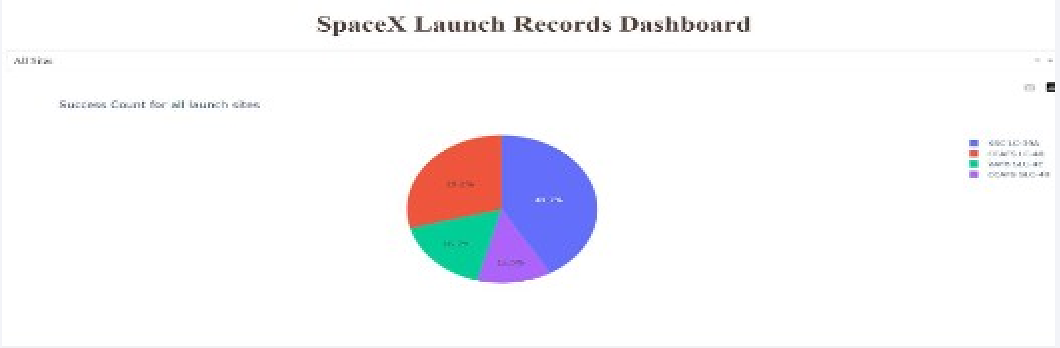
# KSC-LC29A Launch Site

- KSC-LC29A is the launch site with the highest success rate as shown in the pie chart.
- Payload ranges with highest and lowest success rate
- F9 booster version (v1.0, v1.1, FT, B4, B5, etc) with highest launch success rate.

# Payload vs Launch Outcome

Show screenshots of Payload vs. Launch Outcome scatter plot for all sites, with different payload selected in the range slider

- F9 booster version (v1.0, v1.1, FT, B4, B5, etc) with highest launch success rate.




Section 5

# Predictive Analysis (Classification)

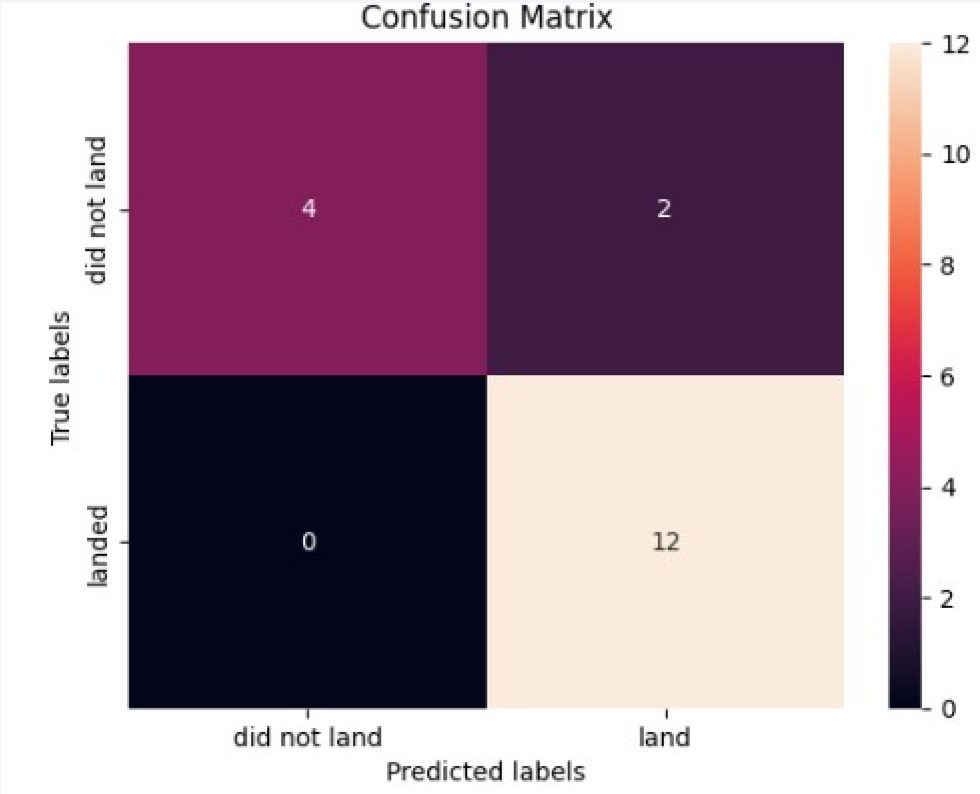
# Classification Accuracy

The different models presented similar accuracy results with the Decision Tree being slightly better at 89% accuracy.

| | 0 |
|---|---|
| **Method** | Test Data Accuracy |
| **Logistic_Reg** | 0.833333 |
| **SVM** | 0.833333 |
| **Decision Tree** | 0.888889 |
| **KNN** | 0.833333 |

# Confusion Matrix

- Confusion Matrix for the Decision Tree which was the most accurate.
- 4 True positives (top left)
- 2 False positives (top right)
- 0 False Negatives (bottom left)
- 12 True Negatives (bottom right)



# Conclusions

- Different launch sites have different success rates with CCAFS LC-40 success rate at 60% while KSC LC-39A and VAFB SLC4E have 77% success rate.
- The data shows that the success rate increases with the increase in number of flights for each site.
- The ES-L1, GEO, HEO and SSO orbits have the highest success rates at 100%
- LEO orbit succes rate seems to be correlated to flight number while GTO orbit has no relation.
- Polar, ISS and LEO are more successful with heavy payloads than other orbits.
- Success rates have continued to increase since 2013

# Appendix

Github project link which includes all jupyter notebooks:

https://github.com/jdowling23/Applied-Data-Science-SpaceX-Launch/tree/main

# Thank you!