# Reuse Capability

*A Data analytics Study for SpaceY*

BY SANGEETH KUNNATH

# Outline

- Executive Summary
- Introduction
- Methodology and Results
  - Data Collection and Wrangling
  - EDA with Visualization
  - EDA with SQL
  - Folium and Plotly
  - Prediction Models and Insights
- Conclusion

# Executive Summary

## Methodology

Data was collected for rocket launches using REST API from SpaceX and publicly available data from Wikipedia. Data wrangling was performed to extract launch outcomes and prepare it for Machine Learning models

Once the data was ready, exploratory data analytics was done through visualizations like plots, interactive maps and dashboards. Factors like critical payload, optimal launch sites, flight numbers and yearly trends were explored. Geographic visualizations using Folium was also utilized to understand proximity to Geographic markers.

Data was further analysed using SQL queries to support/augment findings observed though data visualization

We then pursued and built machine learning models to assess which one is the best for predicting landing outcomes

## Results

Over time, Launch success has vastly improved especially at KSC LC 39A has the highest success rate

Specific orbits have 100% success rate (Orbits ES L1, GEO, HEO, and SSO have

The launch sites that have the most favorable results are close to the coast

Except the decision tree model, all the other selected models performed similarly and very well for that matter

# Introduction

## Background

- SpaceX is aspires to be an affordable space travel and transportation option in the complex world of space industry. for everyone.
- Its accomplishments include sending spacecraft to the international space station, launching a satellite constellation that provides internet access and sending manned missions to space. SpaceX can do this
- Falcon 9 is a reusable, two-stage rocket designed and manufactured by SpaceX for the reliable and safe transport of people and payloads into Earth orbit and beyond.
- Falcon 9 is the world’s first orbital class reusable rocket. Reusability allows SpaceX to ‘refly’ the most expensive parts of the rocket, which in turn drives down the cost of space access.

Source

## Objectives

- Falcon 9 launches cost an average of ~$62 Million while other providers cost upward of $165 Million
- ***How is this possible?*** – Reusability of the first stage. We need to explore the data and understand what happens
- ***What factors make launches successful?*** – There are multiple Features/Conditions that determine the success rate. We need to understand which make more sense.
- ***What is the best predictive approach towards making this successful?*** – A predictive model is required to ascertain the best success rate

# Data Collection

## Methodology

- To understand how this all works, we need to first look at past launch data and understand facts basis  successful launches.
- In order to that, the fist step is to collect available launch data
- ***We used the GET request  method to collect the raw data and did some basic data wrangling / formatting***

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

[6] Python

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

[7] Python

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

Python

- Below is a summary of preprocessing tasks done to get the final data frame post using GET Request method
  - ***Decode response using .json() and convert to dataframe (see above)***
  - ***Create custom function to request information about launches***
  - ***Create dictionary and a data frame from the dictionary and load data***

## Results

```python
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}
```

21] Python

Then, we need to create a Pandas data frame from the dictionary launch_dict.

```python
# Create a data from launch_dict
launch_data = pd.DataFrame(launch_dict)
```

22] Python

Show the summary of the dataframe

```python
# Show the head of the dataframe
launch_data.head()
```

23] Python

| | FlightNumber | Date | BoosterVersion | PayloadMass | Orbit | LaunchSite | Outcome | Flights | GridFins | Reused | Legs |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2006-03-24 | Falcon 1 | 20.0 | LEO | Kwajalein Atoll | None None | 1 | False | False | False |
| 1 | 2 | 2007- | Falcon 1 | NaN | LEO | Kwajalein | None | 1 | False | False | Fals |

IBM Developer SKILLS NETWORK

# Data Collection - Contd

## Methodology

- ***Filter the data to Falcon9*** rockets only as that is the scope of our analysis
- The next step was to deal with missing values. ***We replaced missing values of Payload Mass with calculated .mean()***
- ***Export data to csv file***

```python
# Hint data['BoosterVersion']!='Falcon 1'
data_falcon9 = launch_data[launch_data['BoosterVersion']!='Falcon 1
data_falcon9.head()
```

✓ 0.0s Python

```python
# Calculate the mean value of PayloadMass column
mean=data_falcon9['PayloadMass'].mean()
# Replace the np.nan values with its mean value
data_falcon9['PayloadMass'].replace(np.nan,mean, inplace=True)

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

Python

## Results

**Falcon 9 filtered Data Frame →**

| | FlightNumber | Date | BoosterVersion | PayloadMass | Orbit | LaunchSite |
|---|---|---|---|---|---|---|
| 4 | 6 | 2010-06-04 | Falcon 9 | NaN | LEO | CCSFS SLC 40 |
| 5 | 8 | 2012-05-22 | Falcon 9 | 525.0 | LEO | CCSFS SLC 40 |
| 6 | 10 | 2013-03-01 | Falcon 9 | 677.0 | ISS | CCSFS SLC 40 |
| 7 | 11 | 2013-09-29 | Falcon 9 | 500.0 | PO | VAFB SLC 4E |
| 8 | 12 | 2013-12-03 | Falcon 9 | 3170.0 | GTO | CCSFS SLC 40 |

**Before Replacing Missing Values**

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

```
FlightNumber       0
Date               0
BoosterVersion     0
PayloadMass        5
Orbit              0
LaunchSite         0
Outcome            0
Flights            0
GridFins           0
Reused             0
Legs               0
LandingPad        26
Block              0
ReusedCount        0
Serial             0
Longitude          0
Latitude           0
dtype: int64
```

**After Replacing Missing Values**

```python
data_falcon9['PayloadMass'].re
```

```
FlightNumber       0
Date               0
BoosterVersion     0
PayloadMass        0
Orbit              0
LaunchSite         0
Outcome            0
Flights            0
GridFins           0
Reused             0
Legs               0
LandingPad        26
Block              0
ReusedCount        0
Serial             0
Longitude          0
Latitude           0
dtype: int64
```

# Data Web Scraping

## Methodology

- As the next source of data, Wikipedia was consulted and publicly available data therein was scraped. A permanently linked Wikipedia page with launch data in HTML tables was found
- ***Falcon 9 launch data was retrieved using HTTP GET method and Beautiful Soup***
- All columns names were extracted and parsed
- Once again a dictionary was created for the columns and data was loaded into the data frame

```python
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Fa
```
Python

```python
# use requests.get() method with the provided static_url
# assign the response to a object
html_data = requests.get(static_url)
html_data.status_code
```
Python

```python
# Use BeautifulSoup() to create a BeautifulSoup object from a respo
soup = BeautifulSoup(html_data.text)
```
Python

## Results

```python
# Use soup.title attribute
soup.title
```
Python

```
<title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>
```

```python
df=pd.DataFrame(launch_dict)
df.head()
```
Python

| | Flight No. | Launch site | Payload | Payload mass | Orbit | Customer | Launch outcome | |
|---|---|---|---|---|---|---|---|---|
| 0 | 1 | CCAFS | Dragon Spacecraft Qualification Unit | 0 | LEO | SpaceX | Success\n | v1.0 |
| 1 | 2 | CCAFS | Dragon | 0 | LEO | NASA | Success | v1.0 |
| 2 | 3 | CCAFS | Dragon | 525 kg | LEO | NASA | Success | v1.0 |
| 3 | 4 | CCAFS | SpaceX CRS-1 | 4,700 kg | LEO | NASA | Success\n | v1.0 |
| 4 | 5 | CCAFS | SpaceX CRS-2 | 4,877 kg | LEO | NASA | Success\n | v1.0 |

# Data Wrangling

## Methodology

- The next step is to do some wrangling and make sure the data is ready.
  - The first task was to ***determine the data labels***
  - We then calculated ***count of launches for each site and the occurrences of orbit and mission outcome per orbit***
  - ***Converted landing outcome to binary***
- It was notable that landing was not always successful and there were combination of scenarios. Successful outcomes were converted to 1 and 0 for unsuccessful ones. The scenarios were as follows
  - **True Ocean:** Successful landing to a specific region of the ocean
  - **False Ocean:** Unsuccessful landing to a specific region of ocean
  - **True RTLS:** Successful landing on a landing pad on ground
  - **False RTLS:** Unsuccessful landing on a landing pad on ground
  - **True ASDS:** Successful landing on a drone ship
  - **False ASDS:** Unsuccessful landing on a drone ship

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

```
# landing_class = 0 if bad_outcome
# landing_class = 1 otherwise
landing_class = []

for i in df['Outcome']:
    if i in set(bad_outcomes):
        landing_class.append(0)
    else:
        landing_class.append(1)
```

## Results

```
# Apply value_counts() on column LaunchSite
df['LaunchSite'].value_counts()
```

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

```
# landing_outcomes = values on Outcome column
landing_outcomes = df['Outcome'].value_counts()
landing_outcomes
```

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

```
# Apply value_counts on Orbit column
df['Orbit'].value_counts()
```

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

```
df['Class']=landing_class
df[['Class']].head(8)
```

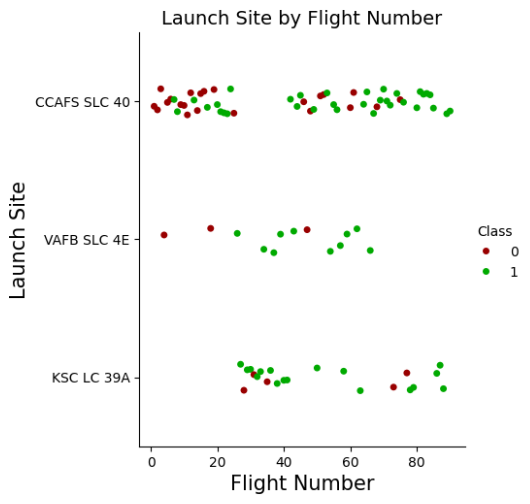
| | Class |
|---|---|
| 0 | 0 |
| 1 | 0 |
| 2 | 0 |
| 3 | 0 |
| 4 | 0 |
| 5 | 0 |
| 6 | 1 |
| 7 | 1 |

# Exploratory Data Analytics with Visualization

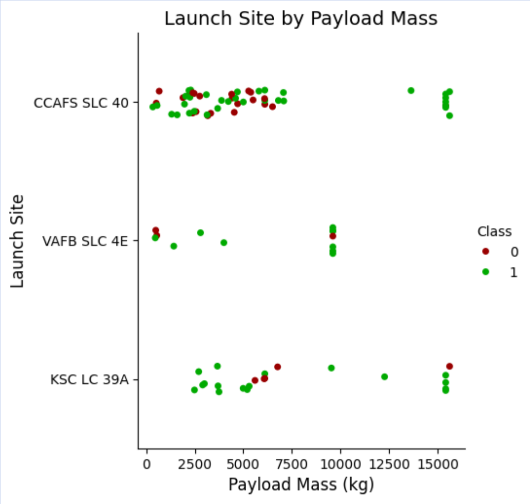
## Methodology

- Exploratory data analytics (EDA) was carried out by plotting the following
  - Mission outcome – Flight Number Vs Payload
  - Mission outcome – Launch site Vs Payload
  - Success Rate by Orbit Type
  - Success Rate Trend Over Years

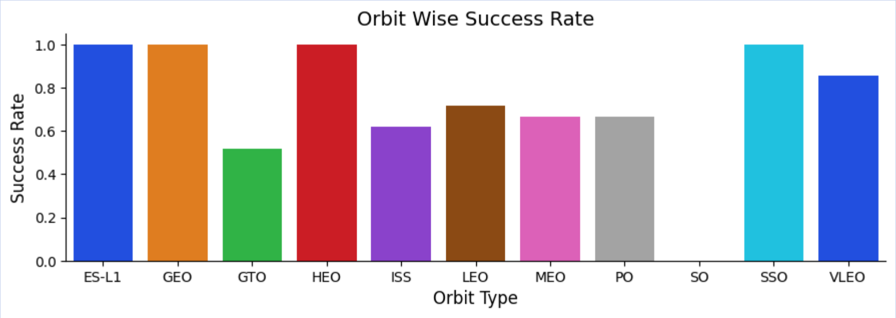
## Results



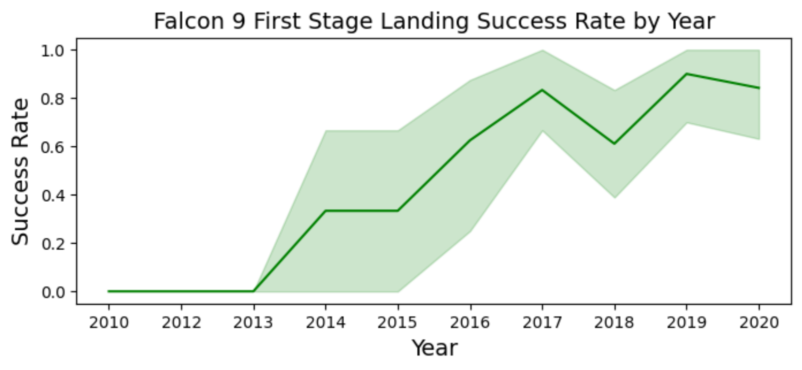

*As the flight number increases, the first stage is more likely to land successfully*




*KSC LC 39A has the highest success rate for launches with less than 5000 kg payload*




*Orbits ES L1, GEO, HEO and SSO have a 100% success rate*




*Overall Success rate has improved over the years since 2013*

# Exploratory Data Analytics with SQL

## Methodology

We continue with our EDA by querying the data to find the following information. Answers to the following aspects queried on the database is available to the right under results:

1. Unique Launch sites
2. Total payload carried by NASA (CRS)
3. Average payload mass carried by booster version F9 v1.1
4. Date of first successful ground pad landing
5. 5 Launch sites starting with CCS
6. Names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
7. Successful and failed mission outcomes count
8. Names of the booster versions that have carried out the maximum payload mass
9. Failed outcome by booster version and launch site by month in 2015
10. Successful landing outcomes between the date 04-06-2010 and 20-03-2017 in descending order

## Results

1

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

2

| SUM(PAYLOAD_MASS__KG_) |
|---|
| 48213 |

3

| AVG(PAYLOAD_MASS__KG_) |
|---|
| 2928.4 |

4

| MIN(DATE) |
|---|
| 2015-12-22 |

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 | 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 |

6

| Payload |
|---|
| JCSAT-14 |
| JCSAT-16 |
| SES-10 |
| SES-11 / EchoStar 105 |

7

| Success | Failure |
|---|---|
| 61 | 40 |

8

| 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 |

9

| Month | landing__outcome | booster_version | launch_site |
|---|---|---|---|
| January | Failure (drone ship) | F9 v1.1 B1012 | CCAFS LC-40 |
| April | Failure (drone ship) | F9 v1.1 B1015 | CCAFS LC-40 |

10

| 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 |

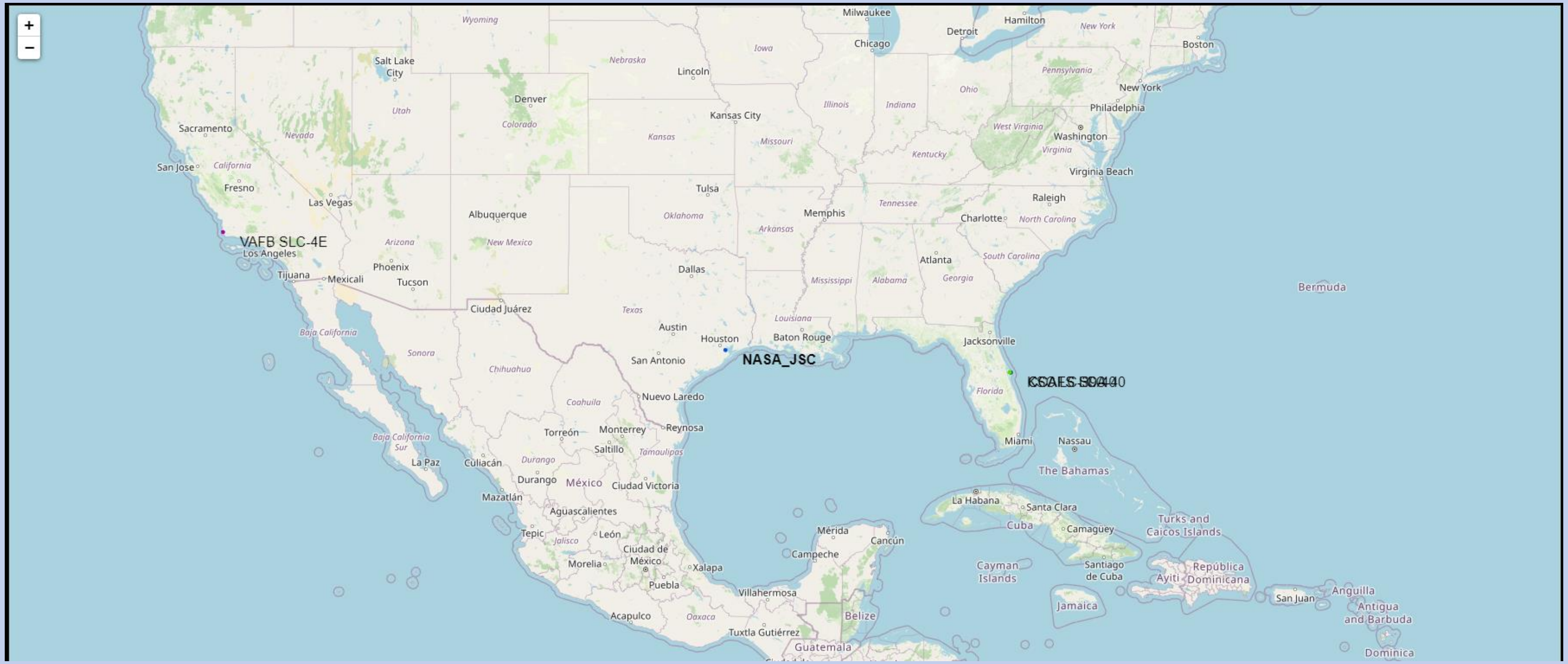
# Geographic Markers (Methodology and Results)

## Methodology

1. We have marked the launch sites and added markers to mark the success or failure for each site using Folium
2. We assigned the feature launch outcomes (failure or success) to class 0 and 1.i.e., 0 for failure, and 1 for success.
3. Using the color labeled marker clusters, we identified which launch sites have relatively high success rate.
4. We calculated the distances between a launch site to its proximities. We answered
5. some question for instance:
   - Are launch sites near railways, highways and coastlines.
   - Do launch sites keep certain distance away from cities.

## Results

***Launch Sites →***

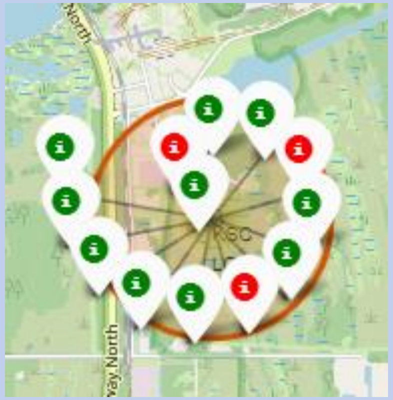



- VAFB SLC-4E (California, USA)
  - Vandenberg Air Force Base Space Launch Complex 4E
- KSC LC-39A (Florida, USA)
  - Kennedy Space Center Launch Complex 39A
- CCAFS LC-40 (Florida, USA)
  - Cape Canaveral Air Force Station Launch Complex 40
- CCAFS SLC-40 (Florida, USA)
  - Cape Canaveral Air Force Station Space Launch Complex 40

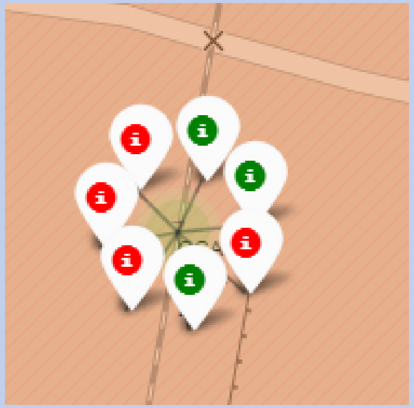
IBM Developer
SKILLS NETWORK

# Geographic Markers (Results Continued)

## Results

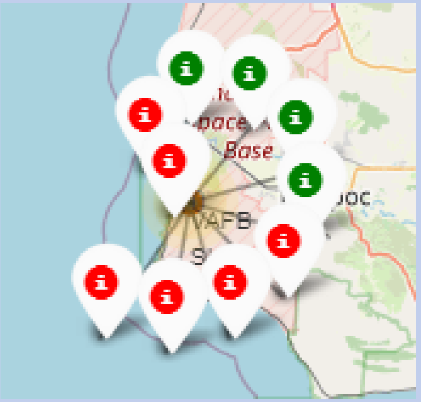


KSC LC-39A

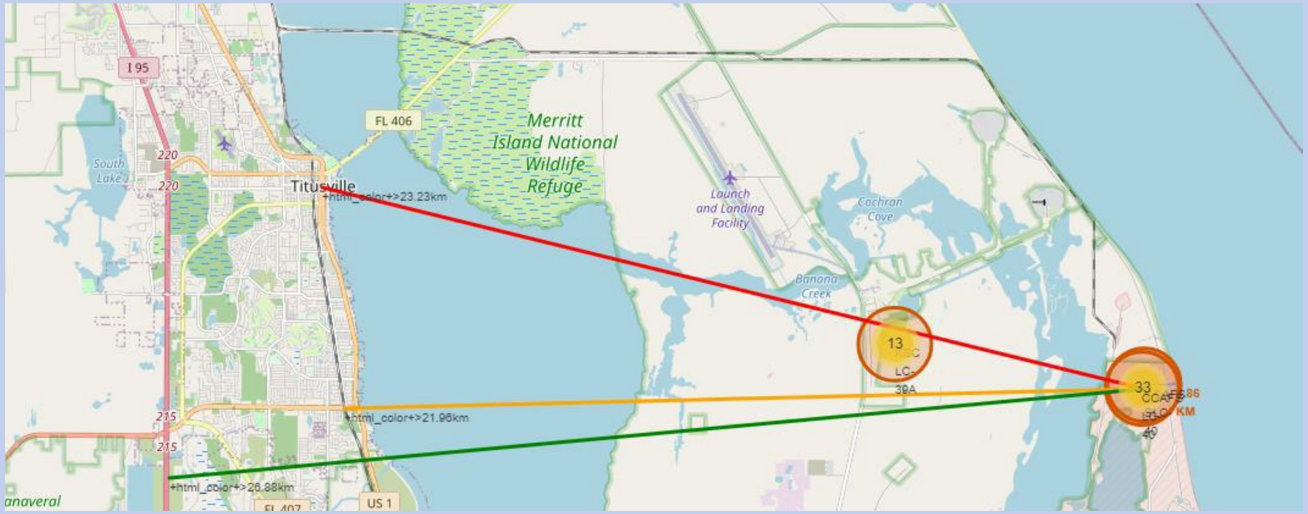


CCAFS LC-40

CCAFS SLC-40



VAFB SLC-4E




- CCAFS LC-40 and CCAFS SLC-40 are both almost together. If we look at the distance to proximities from these launch sites, they are as follows:
  - **.86 km** from nearest coastline
  - **21.96 km** from nearest railway
  - **23.23 km** from nearest city
  - **26.88 km** from nearest highway
- *Proximity to coasts is favorable. Spent stages or failed launches fall into the ocean and don't fall on property.*
- *Proximity to cities is unfavorable. Launch sites should be away from to avoid damages to property and lives*
- *Proximity to infrastructure is good as long as it is not too close to people but close enough to rail/docks/roads to move material and men as required to the launch site*

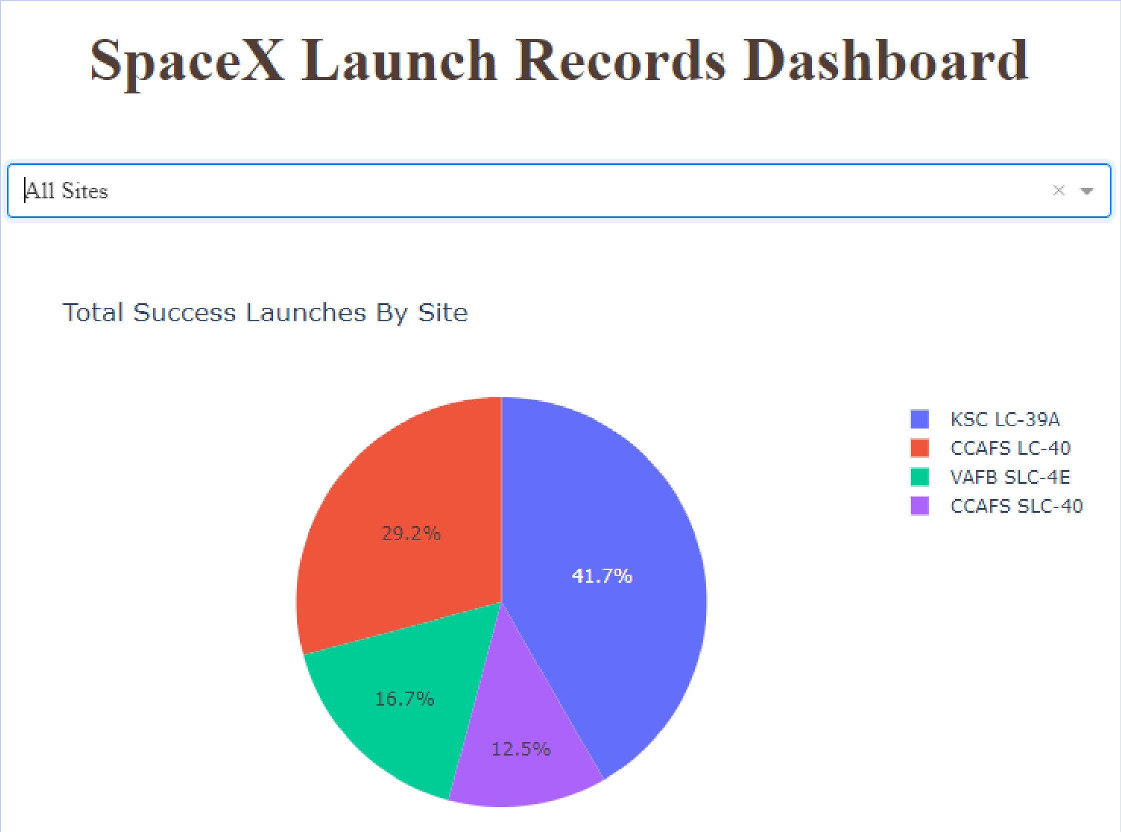
- **Green** markers show successful launches. **Red** show failed launches
- From the marker spread KSC LC-39A appears to be more successful

# Plotly Dashboard (Methodology and Results)

## Methodology

- The input dropdown is used to select one / all launch sites for the pie chart and scatterplot.
- The pie chart displays one of two things:
    - ***For All Sites*** –the distribution of successful Falcon 9 first stage landings between the sites
    - ***For One Site*** –the distribution of successful and failed Falcon 9 first stage landings for that site (0 = Failure, 1 = Success)
- The input slider is used to filter the ***payload masses for the scatterplot.***
- The scatterplot displays the distribution of Falcon 9 ***first stage landings split by payload mass, mission outcome and by booster version category***

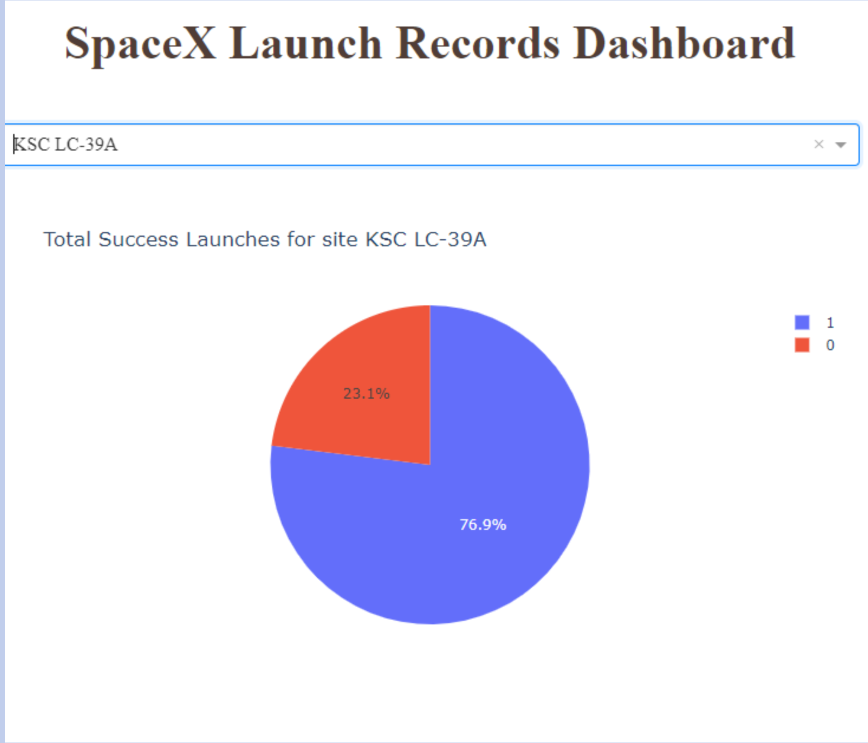
## Results



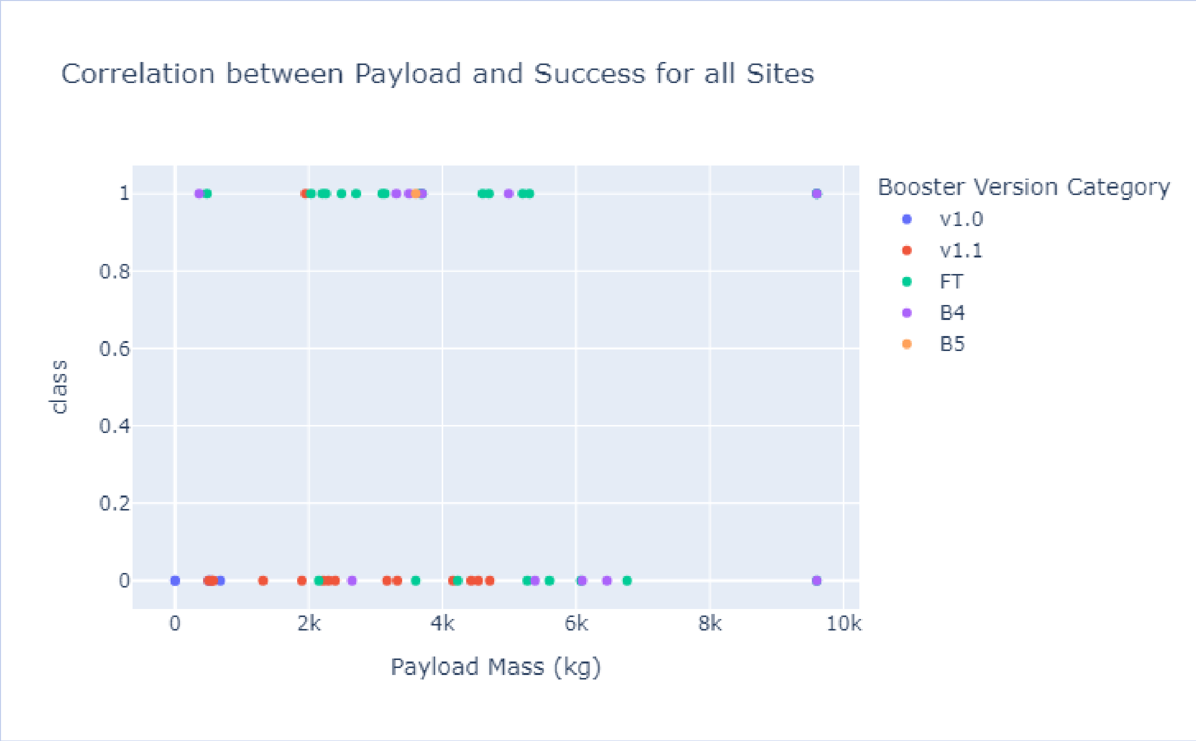

**The greatest share of successful Falcon 9 first stage landing outcomes (at 41.7% of the total) occurred at KSC LC-39A**

# Plotly Dashboard (Results continued)

## Results




**KSC LC 39A** has the highest success rate amongst launch sites **(76.9%)**



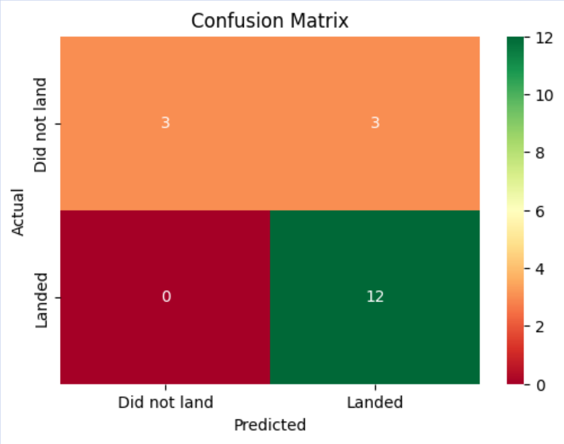

Payloads between **2,000 kg and 5,000 kg** have the highest success rate (**FT Booster** category has the most)

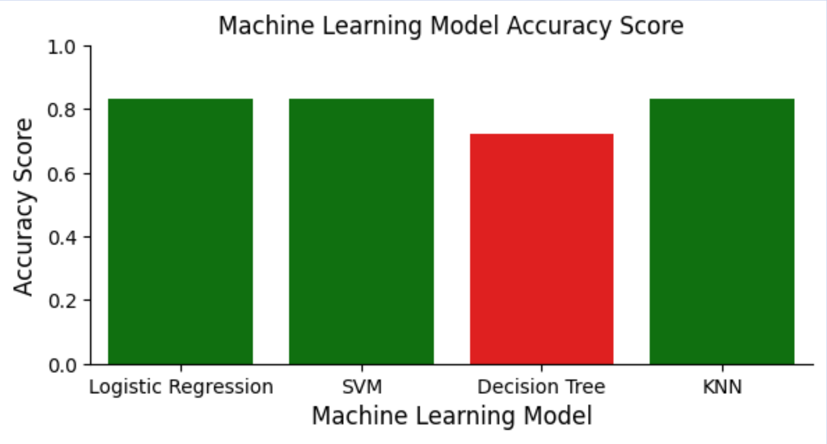
# Prediction Models and Insights

## Methodology

- Create a ***NumPy*** array from the class column and standardize using ***StandardScaler***. Fitted and transformed data
- Data was split into 2 buckets 1) Training Data Set 2) Testing Data Set
- The training data set was utilized to train on ***Logistic Regression, Support Vector Machine and K-Nearest Neighbours*** approaches for modelling
- Furthermore hyperparameters were evaluated using ***GridSearchCV().*** The best option was selected using '.best_params_'.
- Based on the above hyperparameters, each of the ***above models were scored on accuracy*** using the testing data set and the best performing model was finalized

## Results




- ***Precision*** = True Positive / (True Positive + False Positive)
- ➢ 12 / 15 = .80

- ***Recall*** = True Positive / (True Positive + False Negative)
- ➢ 12 / 12 = 1

- ***F1 Score*** = 2 * (Precision * Recall) / (Precision +
- ➢ 2 * (.8 * 1) / (.8 + 1) = .89

- ***Accuracy*** = (True Positive + TN) / (True Positive + True Negative + False Positive + False Negative) = .833

- There are ***12 True Positives*** and 3 True Negatives
- There are also ***3 False Positives*** and 0 False Negatives
- The fact that there are ***false positives (Type 1 error) is not good***
- All models performed equally well except for the ***Decision Tree model which performed poorly*** relative to the other models

# Conclusion

- The falcon 9 rocket’s first stage landing has been looking better and more successful as more of these are launched
- While multiple machine learning models proved to be successful, a larger data set will cement the predictive models. Except the decision tree model, the logistic regression, SVM and KNN approach all were equally successful
- The optimal scenario featured payloads between 2,000 kg and 5,000 kg in the FT booster category. Keeping the payload mass to the as high as possible within these thresholds have proved the most successful

# Thank You