IBM Developer
SKILLS NETWORK

# Winning Space Race with Data Science

Yeleussizov Arslan
26.03.2024

# Outline

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

# Executive Summary

- Summary of methodologies

In our report we used next methodologies:

- *Data Collection through API and Web Scraping*
- *Data Wrangling*
- *Exploratory Data Analysis with SQL and Data Visualization*
- *Interactive Visual Analytics with Folium*
- *Machine Learning Prediction*

- Summary of all results

- *Exploratory Data Analysis result*
- *Interactive analytics in screenshots*
- *Predictive Analytics result*

# Introduction

- Project background and context

*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. This goal of the project is to create a machine learning pipeline to predict if the first stage will land successfully.*

- Problems you want to find answers

1. *What factors determine if the rocket will land successfully?*
2. *The interaction amongst various features that determine the success rate of a successful landing.*
3. *What operating conditions needs to be in place to ensure a successful landing program.*

Section 1

# Methodology

# Methodology

## Executive Summary

- Data collection methodology:
  - Data was collected using SpaceX API and web scraping from Wikipedia.
- Perform data wrangling
  - One-hot encoding was applied to categorical features
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

# Data Collection

- The data was collected using various methods

- Data collection was done using get request to the SpaceX API
- Then, we decoded the response content as a Json using .json() function call and turn it into a pandas dataframe using .json_normalize().
- In next step we cleaned the data, checked for missing values and fill in missing values where necessary.
- In addition, we performed web scraping from Wikipedia for Falcon 9 launch records with BeautifulSoup.
- The objective was to extract the launch records as HTML table, parse the table and convert it to a pandas dataframe for future analysis.

# Data Collection – SpaceX API

- We used the get request to the SpaceX API to collect data, clean the requested data and did some basic data wrangling and formatting.

- Click here for the notebook is

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

In [9]:
```
static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_
```

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

In [10]:
```
response.status_code
```

Out[10]: 200

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

In [11]:
```
# Use json_normalize meethod to convert the json result into a dataframe
data = pd.json_normalize(response.json())
```

# Data Collection - Scraping

- We applied web scrapping to webscrap Falcon 9 launch records with BeautifulSoup
- We parsed the table and converted it into a pandas dataframe.
- Click here for the notebook

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

In [6]:
```
# use requests.get() method with the provided static_url
# assign the response to a object
response = requests.get(static_url).text
```

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

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

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

In [8]:
```
# Use soup.title attribute
print(soup.title)
```

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

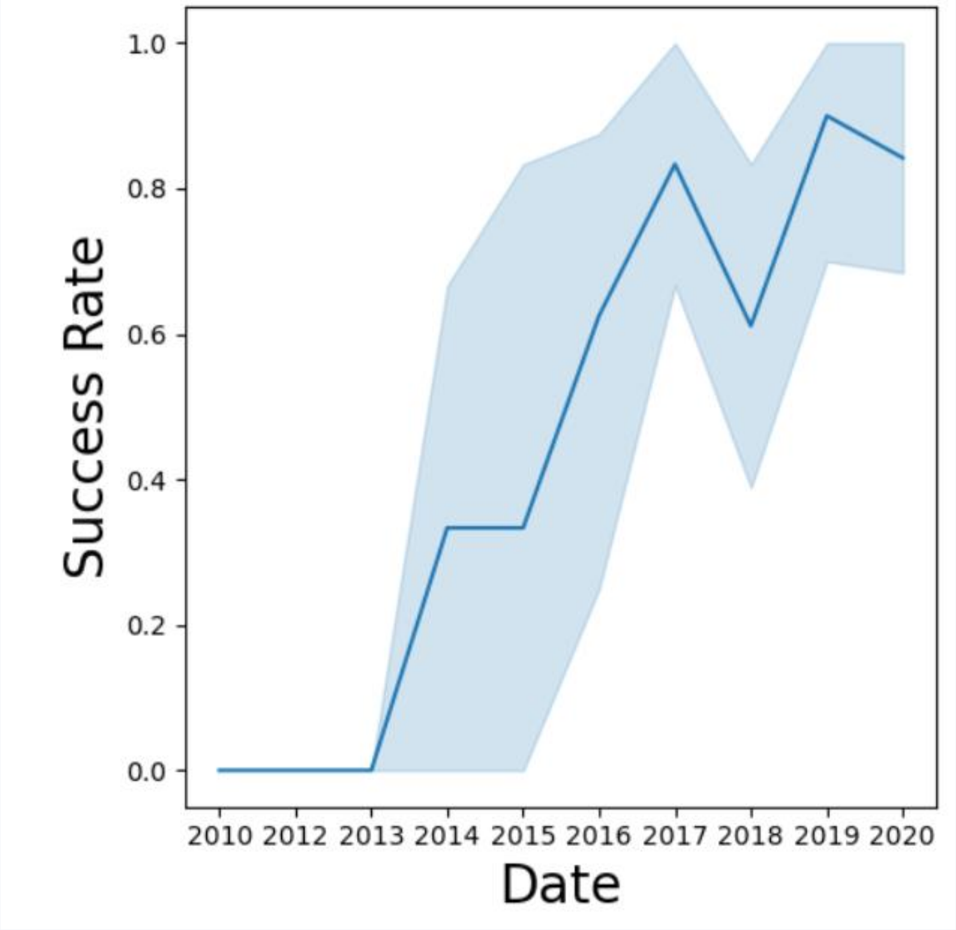
# Data Wrangling

- We performed exploratory data analysis and determined the training labels.
- We calculated the number of launches at each site, and the number and occurrence of each orbits
- We created landing outcome label from outcome column and exported the results to csv.
- Click here for the full notebook

# EDA with Data Visualization

- We explored the data by visualizing the relationship between flight number and launch Site, payload and launch site, success rate of each orbit type, flight number and orbit type, the launch success yearly trend.

- Click here for the notebook



# EDA with SQL

- We loaded the SpaceX dataset into a PostgreSQL database without leaving the jupyter notebook.
- We applied EDA with SQL to get insight from the data. We wrote queries to find out for instance:

- *The names of unique launch sites in the space mission.*
- *The total payload mass carried by boosters launched by NASA (CRS)*
- *The average payload mass carried by booster version F9 v1.1*
- *The total number of successful and failure mission outcomes*
- *The failed landing outcomes in drone ship, their booster version and launch site names.*

- Click here for the notebook

# Build an Interactive Map with Folium

- We marked all launch sites, and added map objects such as markers, circles, lines to mark the success or failure of launches for each site on the folium map.
- We assigned the feature launch outcomes (failure or success) to class 0 and 1.i.e., 0 for failure, and 1 for success.
- Using the color-labeled marker clusters, we identified which launch sites have relatively high success rate.
- We calculated the distances between a launch site to its proximities. We answered some question for instance:

- Are launch sites near railways, highways and coastlines.
- Do launch sites keep certain distance away from cities.

- Click here for the notebook

# Build a Dashboard with Plotly Dash

- We built an interactive dashboard with Plotly dash
- We plotted pie charts showing the total launches by a certain sites
- We plotted scatter graph showing the relationship with Outcome and Payload Mass (Kg) for the different booster version.
- Click here for the notebook

# Predictive Analysis (Classification)

- We loaded the data using numpy and pandas, transformed the data, split our data into training and testing.
- We built different machine learning models and tune different hyperparameters using GridSearchCV.
- We used accuracy as the metric for our model, improved the model using feature engineering and algorithm tuning.
- We found the best performing classification model.
- Click here for the jupyter Notebook

# Results

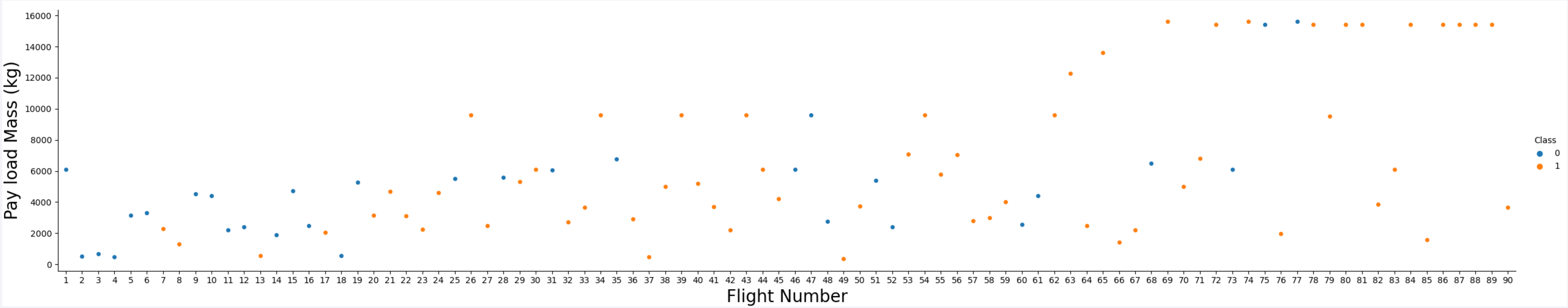
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

# Insights drawn from EDA

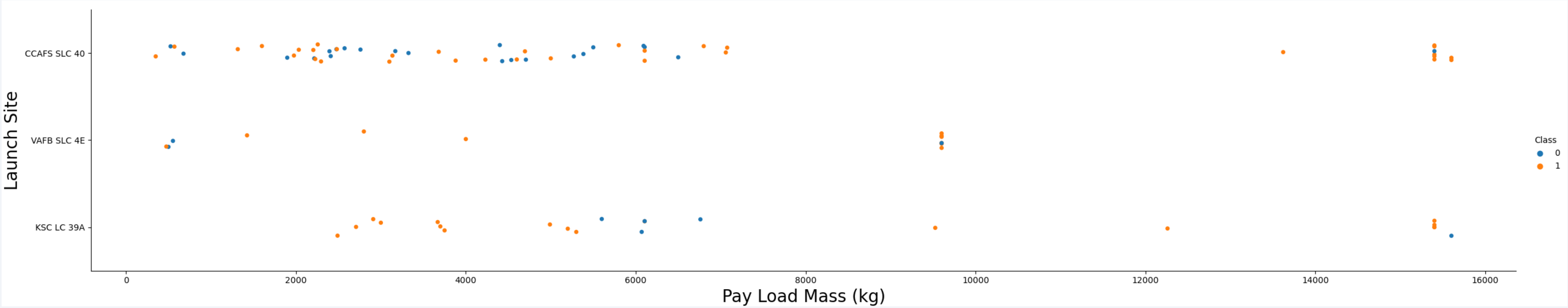
# Flight Number vs. Launch Site

- This picture show us a scatter plot of Flight Number vs. Launch Site



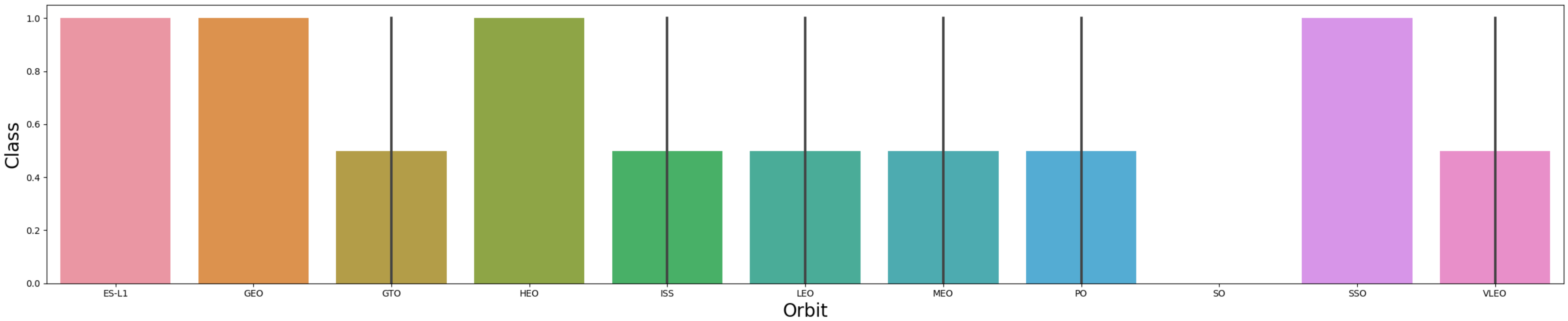
# Payload vs. Launch Site

- Show a scatter plot of Payload vs. Launch Site



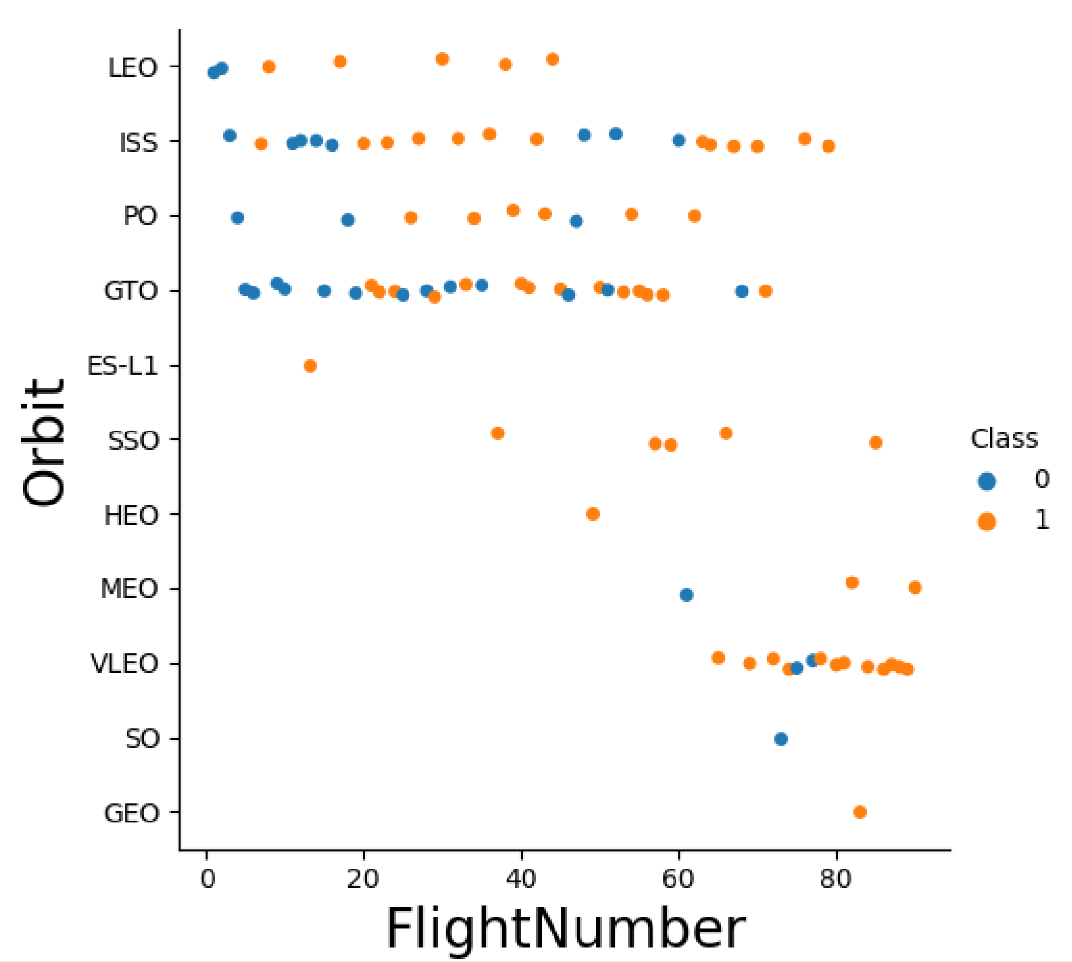
# Success Rate vs. Orbit Type

- Show a bar chart for the success rate of each orbit type



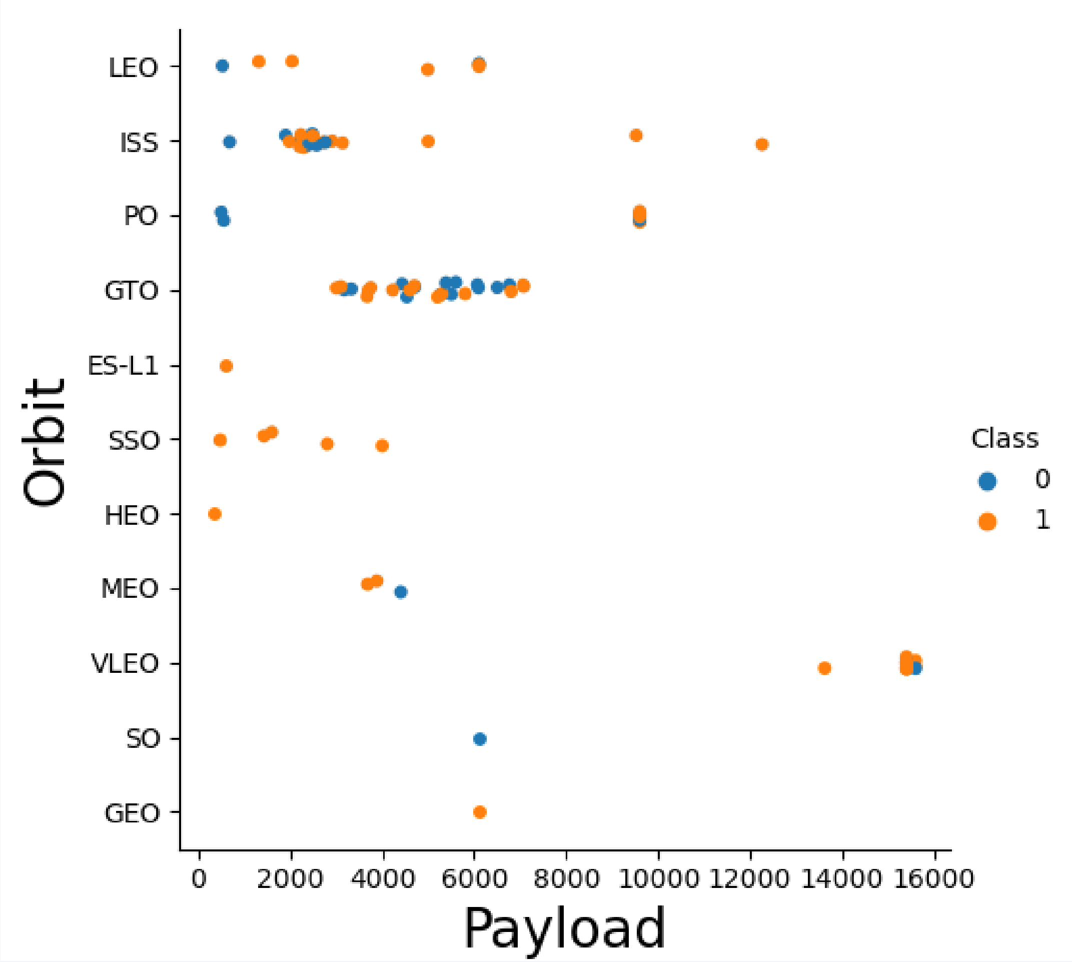
# Flight Number vs. Orbit Type

- Show a scatter point of Flight number vs. Orbit type



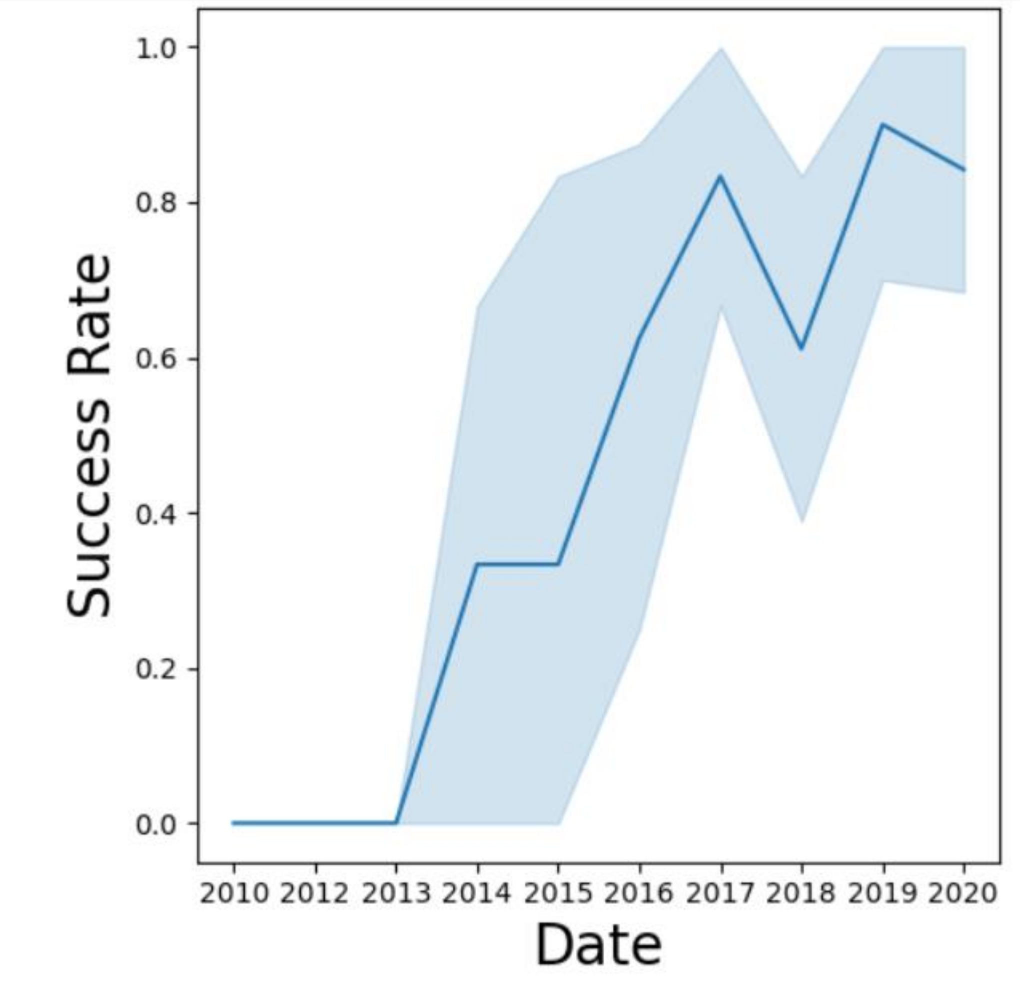
# Payload vs. Orbit Type

- Show a scatter point of payload vs. orbit type



# Launch Success Yearly Trend

- Show a line chart of yearly average success rate
- From the plot, we can observe that success rate since 2013 kept on increasing till 2020.



# All Launch Site Names

- Find the names of the unique launch sites
- Present your query result with a short explanation here

### Task 1

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

In [20]:
```
%sql select distinct(Launch_Site) from SPACEXTBL
```

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

Out[20]:

| 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`
- Present your query result with a short explanation here

## Task 2

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

In [26]:

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

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

Out[26]:

| Date | Time (UTC) | Booster_Version | Launch_Site | Payload | PAYLOAD_MASS__KG_ | Orbit | Customer |
|---|---|---|---|---|---|---|---|
| 2010-06-04 | 18:45:00 | F9 v1.0 B0003 | CCAFS LC-40 | Dragon Spacecraft Qualification Unit | 0 | LEO | SpaceX |
| 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 |
| 2012-05-22 | 7:44:00 | F9 v1.0 B0005 | CCAFS LC-40 | Dragon demo flight C2 | 525 | LEO (ISS) | NASA (COTS) |
| 2012-10-08 | 0:35:00 | F9 v1.0 B0006 | CCAFS LC-40 | SpaceX CRS-1 | 500 | LEO (ISS) | NASA (CRS) |
| 2013-03-01 | 15:10:00 | F9 v1.0 B0007 | CCAFS LC-40 | SpaceX CRS-2 | 677 | LEO (ISS) | NASA (CRS) |

# Total Payload Mass

- Calculate the total payload carried by boosters from NASA
- Present your query result with a short explanation here

## Task 3

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

In [27]:
```
%sql select sum(PAYLOAD_MASS__KG_) from SPACEXTBL where CUSTOMER = 'NASA (CRS)'
```

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

Out[27]:

| sum(PAYLOAD_MASS__KG_) |
|---|
| 45596 |

# Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1
- Present your query result with a short explanation here

## Task 4

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

```
In [28]: %sql select avg(PAYLOAD_MASS__KG_) from SPACEXTBL where BOOSTER_VERSION = 'F9 v1.1'

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

Out[28]:

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

# First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad
- Present your query result with a short explanation here

## Task 5

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

*Hint:Use min function*

In [29]:
```
%sql select min(DATE) from SPACEXTBL where Landing_Outcome = 'Success (ground pad)'
```

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

Out[29]:

| 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
- Present your query result with a short explanation here

## 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

In [30]:
```
%sql select Booster_Version from SPACEXTBL WHERE Landing_Outcome = 'Success (drone ship)' and PAYLOAD_MASS__KG_ > 4000 and P
```

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

Out[30]:

| 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
- Present your query result with a short explanation here

## Task 7

List the total number of successful and failure mission outcomes

In [31]:
```
%sql select count(Mission_Outcome) from SPACEXTBL WHERE Mission_Outcome = 'Success' or Mission_Outcome = 'Failure (in flight
```

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

Out[31]:

| count(Mission_Outcome) |
| --- |
| 99 |

# Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass
- Present your query result with a short explanation here

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

In [33]:
```
%sql select Booster_Version from SPACEXTBL where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEXTBL)
```

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

Out[33]:

| 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 |

# 2015 Launch Records

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

- Present your query result with a short explanation here

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

In [18]:

```
task_9 = '''
        SELECT BoosterVersion, LaunchSite, LandingOutcome
        FROM SpaceX
        WHERE LandingOutcome LIKE 'Failure (drone ship)'
            AND Date BETWEEN '2015-01-01' AND '2015-12-31'
        '''
create_pandas_df(task_9, database=conn)
```

Out[18]:

| | boosterversion | launchsite | landingoutcome |
|---|---|---|---|
| 0 | F9 v1.1 B1012 | CCAFS LC-40 | Failure (drone ship) |
| 1 | F9 v1.1 B1015 | CCAFS LC-40 | Failure (drone ship) |

# 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

- Present your query result with a short explanation here

Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad))

In [19]:

```
task_10 = '''
        SELECT LandingOutcome, COUNT(LandingOutcome)
        FROM SpaceX
        WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20'
        GROUP BY LandingOutcome
        ORDER BY COUNT(LandingOutcome) DESC
        '''
create_pandas_df(task_10, database=conn)
```

Out[19]:

| | landingoutcome | count |
|---|---|---|
| 0 | No attempt | 10 |
| 1 | Success (drone ship) | 6 |
| 2 | Failure (drone ship) | 5 |
| 3 | Success (ground pad) | 5 |
| 4 | Controlled (ocean) | 3 |
| 5 | Uncontrolled (ocean) | 2 |
| 6 | Precluded (drone ship) | 1 |
| 7 | Failure (parachute) | 1 |

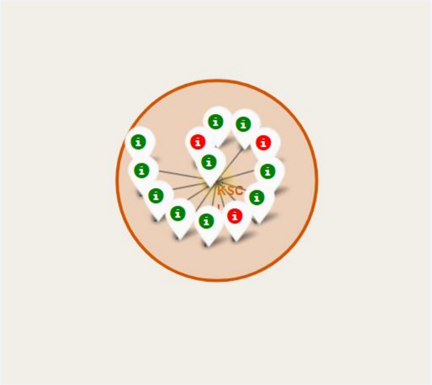
Section 3

# Launch Sites Proximities Analysis

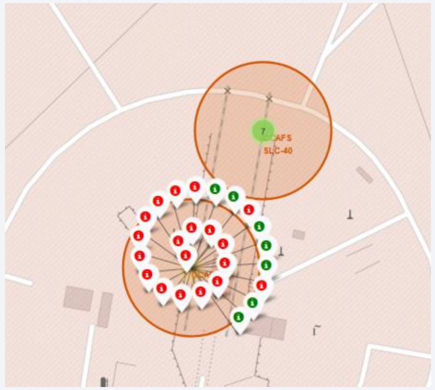
# <Folium Map Screenshot 1>

# <Folium Map Screenshot 2>


Vandenberg Space Launch Complex 4 (CA)
VAFB SLC-4E




Kennedy Space Center (FL)
KSC LC 39A



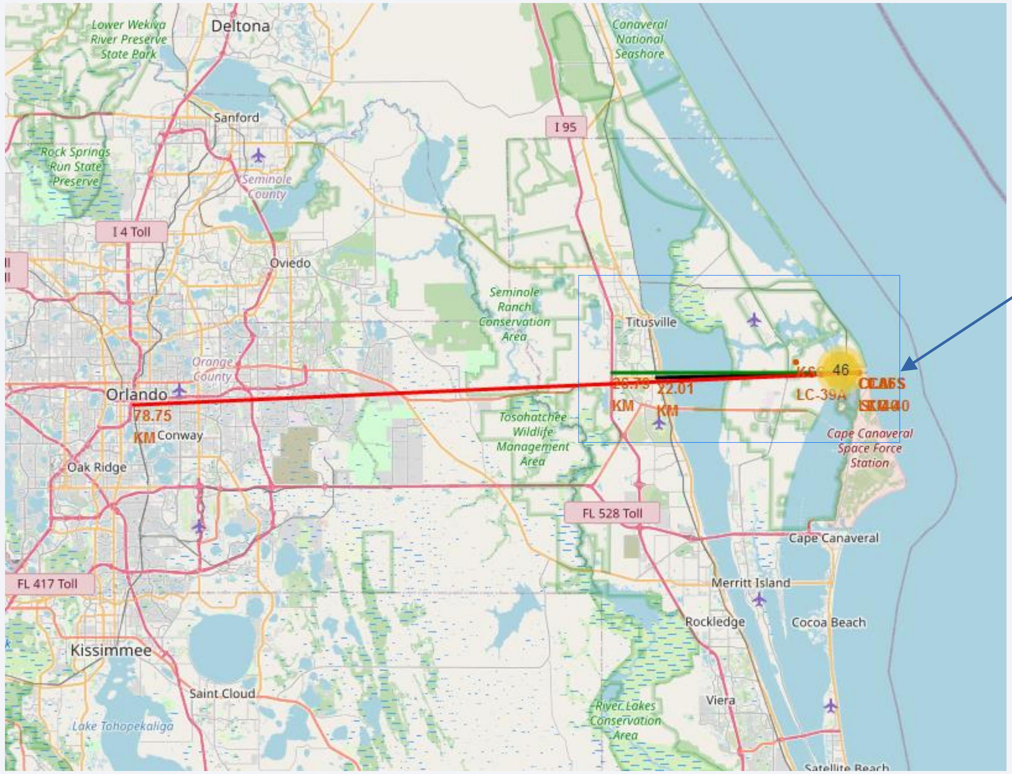
Cape Canaveral (FL)
CCAFS-LC40

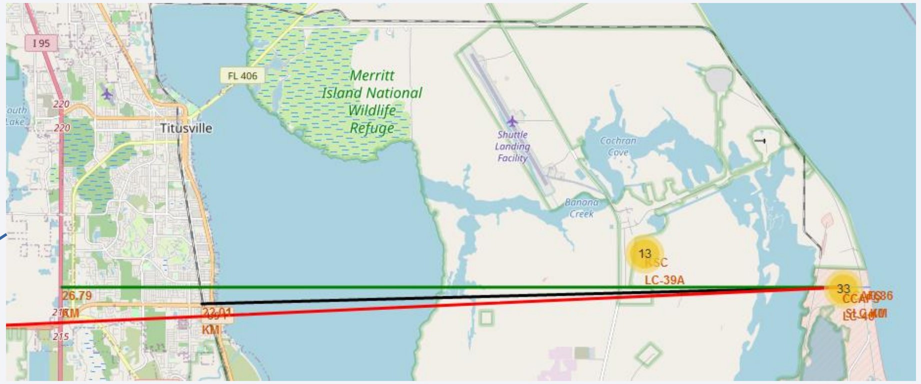
Cape Canaveral (FL)
CCAFS-SLC40

| Launch Site | class | |
|---|---|---|
| CCAFS LC-40 | 0 | 19 |
| | 1 | 7 |
| CCAFS SLC-40 | 0 | 4 |
| | 1 | 3 |
| KSC LC-39A | 0 | 3 |
| | 1 | 10 |
| VAFB SLC-4E | 0 | 6 |
| | 1 | 4 |

**Table: Synthesis of launches outcomes**
Class 0= failure
Class 1= success

# <Folium Map Screenshot 3>




**Distance from CCAFS_SLC40 to:**

- **Closest coast: ~900 m**
- **Florida East Coast Railway: 22.0 km**
- **Highway I 95: 26.8 km**
- **Orlando: 78.75 km**

Launch sites are close to coasts. For safety issues if launcher is lost in the early stage of the flight.

Rockets are launched:

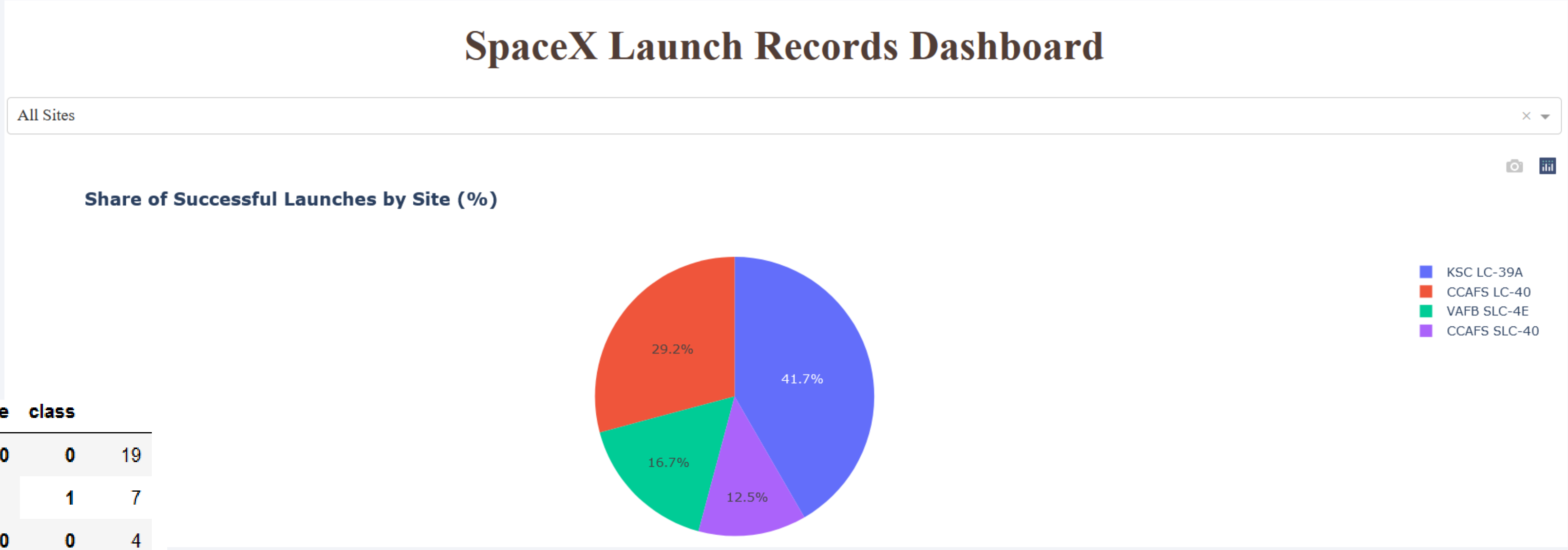
- From West to East over the ocean in Florida.
- North or South bound over the ocean in California. (Polar orbits only)

Launch sites are relatively far from populated areas for protecting population from serious incidents at lift off: explosion on the launch pad.

Section 4

# Build a Dashboard with Plotly Dash

# <Dashboard Screenshot 1>



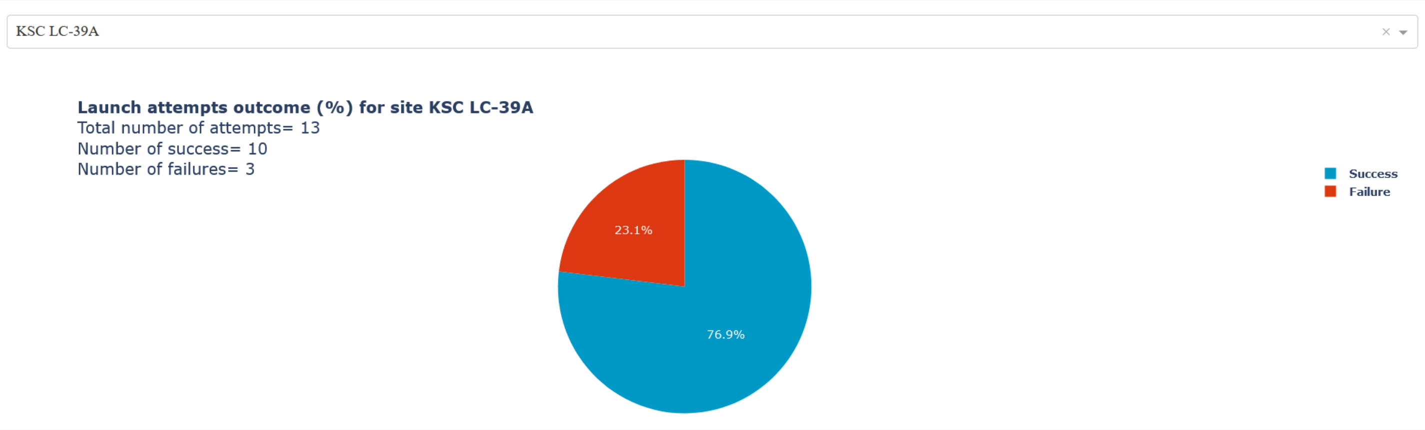

| Launch Site | class | |
|---|---|---|
| CCAFS LC-40 | 0 | 19 |
| | 1 | 7 |
| CCAFS SLC-40 | 0 | 4 |
| | 1 | 3 |
| KSC LC-39A | 0 | 3 |
| | 1 | 10 |
| VAFB SLC-4E | 0 | 6 |
| | 1 | 4 |

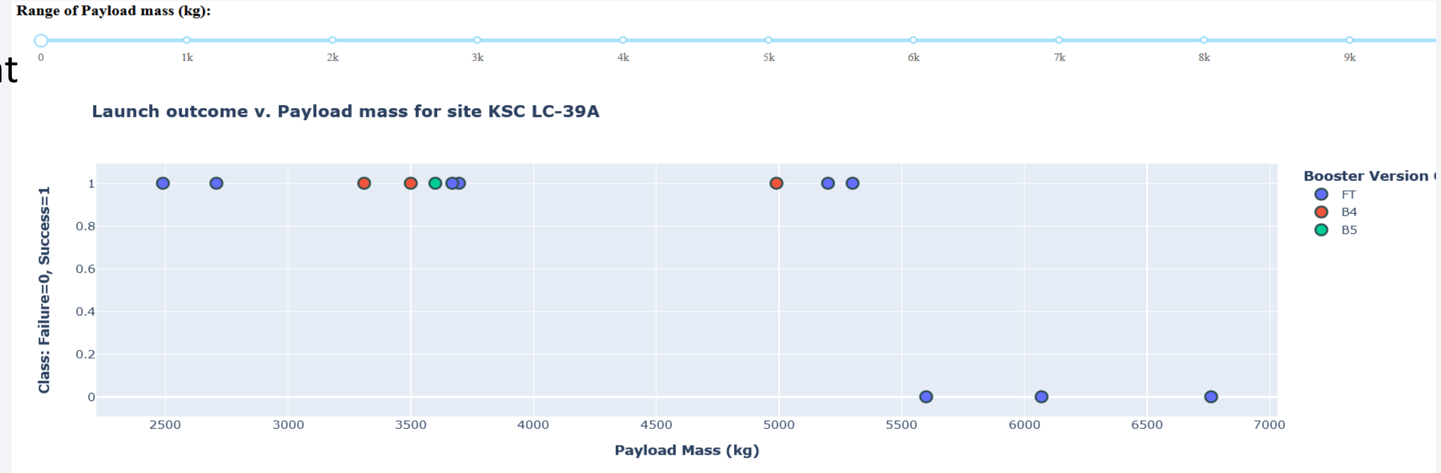
The dahsboard allows an interactive visualization and analysis of Falcon successful launches.
It completes scattered charts.
Here for all sites, we can identify the launch site with highest success rate.
Kennedy Space Center in Florida.

# <Dashboard Screenshot 2>

**KSC LC-39A**
Kennedy Space Center in Florida.
13 flights, 10 successful missions.

- Heavy payload are “high risk”
- Success does not seem to depend upon boosters versions with low mass payload <5500kg.
- B5 and FT are the most reused launchers. Data is not sufficient, but may indicates that they are as reliable as 1 time launchers.



# <Dashboard Screenshot 3>

- V1.0 and v1.1 are early launchers with low reliability.
  Landing legs, were pioneered on the Falcon 9 v1.1 version, but that version never landed intact.
  They were phased out in 2015.
- FT: “Full Thrust” is the next generation and has the highest success rate for payload mass under 6 tons. Including with “drone landing” (see details in next slide).
- Many FT flights are done with reused launchers. And show good reliability.
- Heavy payload are “high risk”.

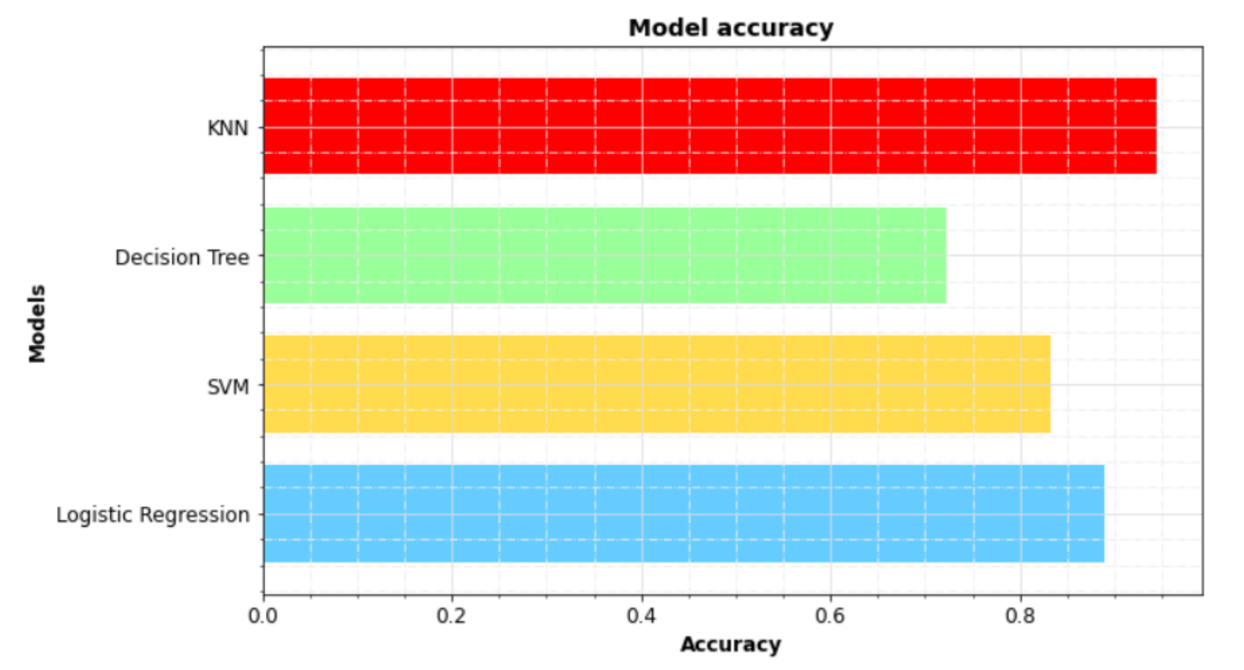
Section 5

# Predictive Analysis (Classification)

# Classification Accuracy

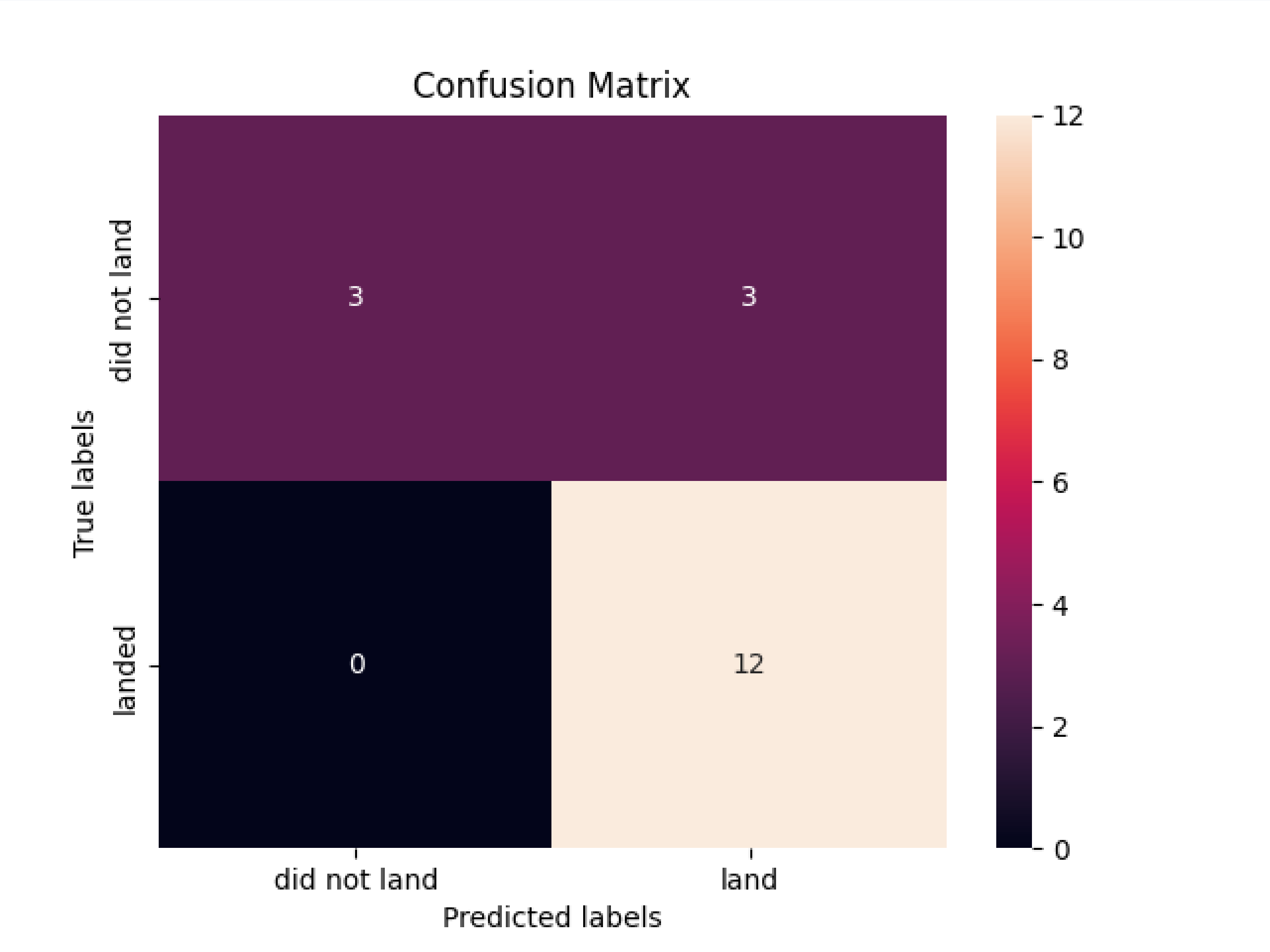
**Classification Accuracy with test set.**
Results with “train test split” random_state=3

- Optimization of SVM and LR hyper-parameters was refined for increasing accuracy with train set.
- 
- It did not necessarily improved accuracy with test set.
- Test set is too small.
- In our case, KNN exhibits the best accuracy: ~94%



# Confusion Matrix

- Show the confusion matrix of the best performing model with an explanation



# Conclusions

We can conclude that:

- The larger the flight amount at a launch site, the greater the success rate at a launch site.
- Launch success rate started to increase in 2013 till 2020.
- Orbits ES-L1, GEO, HEO, SSO, VLEO had the most success rate.
- KSC LC-39A had the most successful launches of any sites.
- The Decision tree classifier is the best machine learning algorithm for this task

# Appendix


[1] https://arstechnica.com/science/2021/03/european-leaders-say-an-immediate-response-needed-to-the-rise-of-spacex/

[2] https://www.arianespace.com/vehicle/ariane-5/

[3] https://www.arianespace.com/wp-content/uploads/2020/06/Arianespace_Brochure_Ariane5_Sept2019.pdf

[4] https://www.arianespace.com/ariane-6/

[5] https://www.arianespace.com/wp-content/uploads/2020/06/Arianespace_Brochure_Ariane6_Sept2019.pdf

[6] https://www.inverse.com/innovation/ariane-6-vs-spacex

[7] https://adsabs.harvard.edu/full/1996ESASP.386..237F

# Thank you!