IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Satwika Purusottama
April 2022

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection through API
  - Data Collection with Web Scraping
  - Data Wrangling
  - Exploratory Data Analysis with SQL
  - Exploratory Data Analysis with 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

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

Section 1

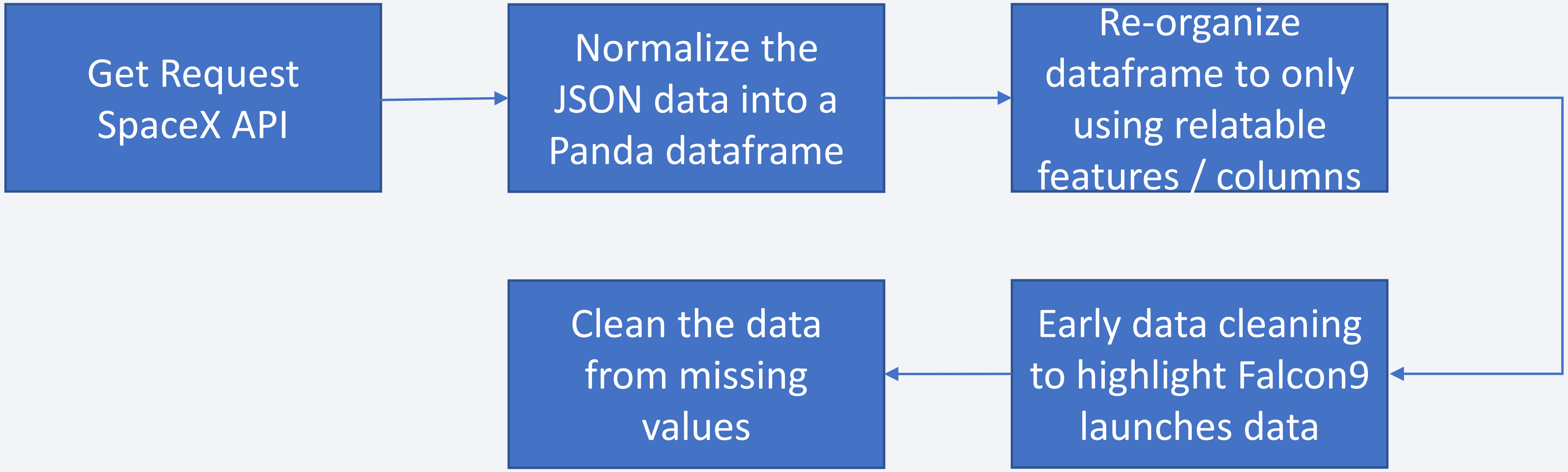
Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was collected using given SpaceX API and web scraping from given URL HTML
- Perform data wrangling
  - Using features engineering to highlight/focus the analysis and one-hot encoding to categorize the 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
  - Evaluate all the possible classification models to best accurate predict the data
  - Final verdict based on the F1 score and accuracy with checking the confusion matrix

Data Collection



Data Collection – SpaceX API

- To get and request to the SpaceX collect data, we utilize the API given by the url. Despite it is given, we still need to normalize and do some features engineering to select the columns we want to highlight.

- https://github.com/spythontest/SpaceY/blob/main/Space%20Y%20Capstone%20Collecting%20Data.ipynb

```
In [6]: spacex_url="https://api.spacexdata.com/v4/launches/past"

In [7]: response = requests.get(spacex_url)
```

```
In [9]: 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

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

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

- Applying the Get Method first for the Falcon9 HTML page as response text. Then using BS to parse the text. From there we can extract/scrape the tables rows and columns. Eventually, we convert and save the to a Dataframe pandas.

- https://github.com/spythontest/SpaceY/blob/main/Space%20Y%20Capstone%20Web%20Scraping.ipynb


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

```
In [8]: # use requests.get() method with the provided static_url
        # assign the response to a object

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

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

```
In [9]: # Use BeautifulSoup() to create a BeautifulSoup object from a response text content

        soup = BeautifulSoup(html, 'html5lib')
```

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

```
In [10]: # Use soup.title attribute

         print(soup.title)

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

```
In [11]: # 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(name='table')
         print(html_tables)
```

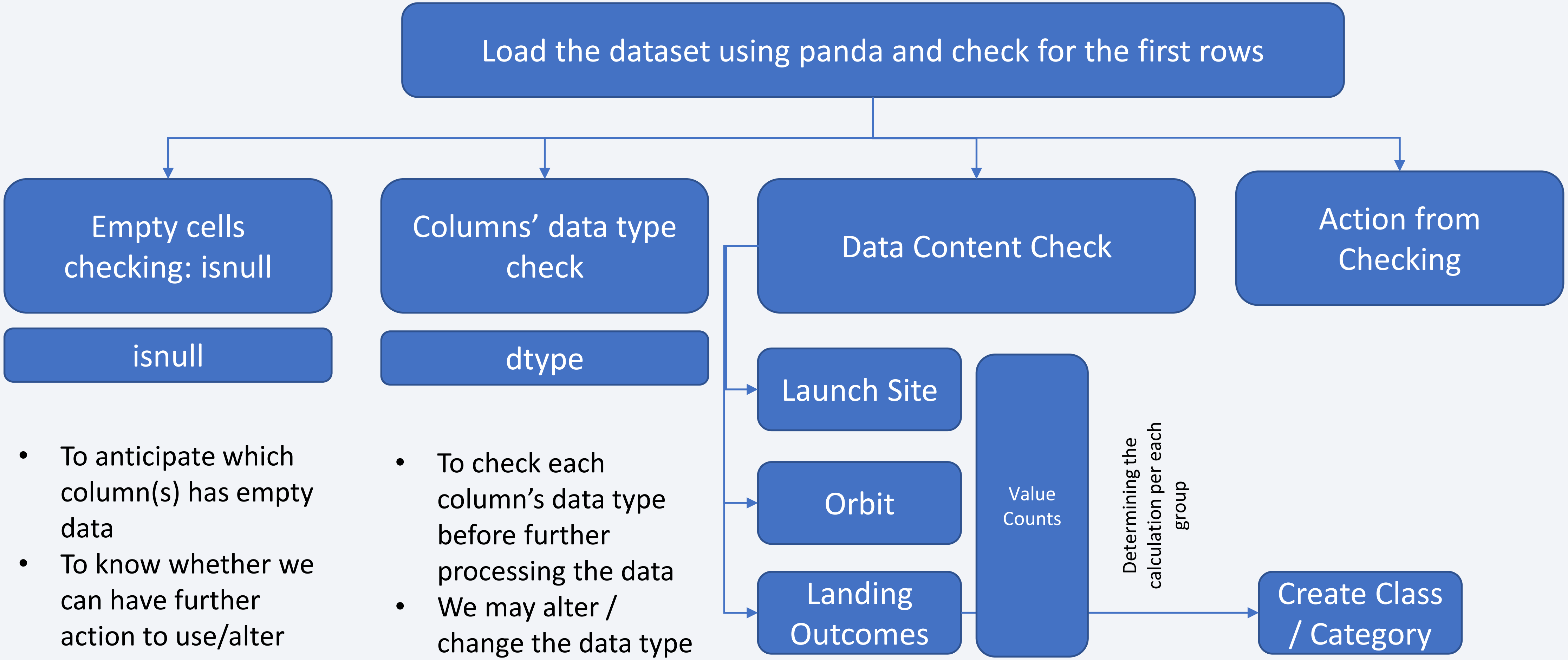
Next, we just need to iterate through the `<th>` elements and apply the provided `extract_column_from_header()` to extract column name one by one

```
In [11]: column_names = []

         # Apply find_all() function with `th` element on first_launch_table
         # Iterate each th element and apply the provided extract_column_from_header() to get a column name
         # Append the Non-empty column name (`if name is not None and len(name) > 0`) into a list called column_names

         element = soup.find_all('th')
         for row in range (len(element)):
             try :
                 name = extract_column_from_header(element[row])
                 if (name is not None and len (name) >0):
                     column_names.append(name)
             except:
                 pass
```

Data Wrangling



- To anticipate which column(s) has empty data
- To know whether we can have further action to use/alter those columns

- To check each column’s data type before further processing the data
- We may alter / change the data type if necessary

Created category / class new column as the future dependent variable from landing outcomes

LINK: https://github.com/spythontest/SpaceY/blob/main/Space%20Y%20Capstone%20EDA.ipynb

EDA with Data Visualization

EDA with Visualization

Early analysis and grab the feel of the data through exploratory data analysis with visualization

R'ship Flightnumber - LaunchSite

" Did any launch site have more success/rates and conducted more flight numbers? "

R'ship Payload - LaunchSite

" How about the payload came into play? Did it have influence in each launch site? "

R'ship Success Rate Orbit Type

" In which orbit type that we had more success rate ? "

R'ship Fligthnumber – Orbit Type

" In which orbit type that we had more success rate, specifically its relationship with flight number ? "

R'ship Payload – Orbit Type

" As previously payload played a factor in launch site, how about its relationship with orbit type? "

R'ship Yearly Success Trend

" Did Space Y project have made progression through the years? "

R'ship Dummy Feature Engineering

Having all factors considered, we can use them as features for the launch pad and launch site categories.

LINK: https://github.com/spythontest/SpaceY/blob/main/Space%20Y%20Capstone%20Pandas%20Matplotlib%20EDA%20Visual.ipynb

EDA with SQL

- For this one, we use the DB2 instead of notebook to perform the SQL queries
- We applied SQL queries to do EDA to find out:
  - The names of unique launch sites in the space mission
  - The totally paload 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.

- Link: https://github.com/spythontest/SpaceY/blob/main/Space%20Y%20Capstone%20EDA%20with%20SQL%20Lab.ipynb

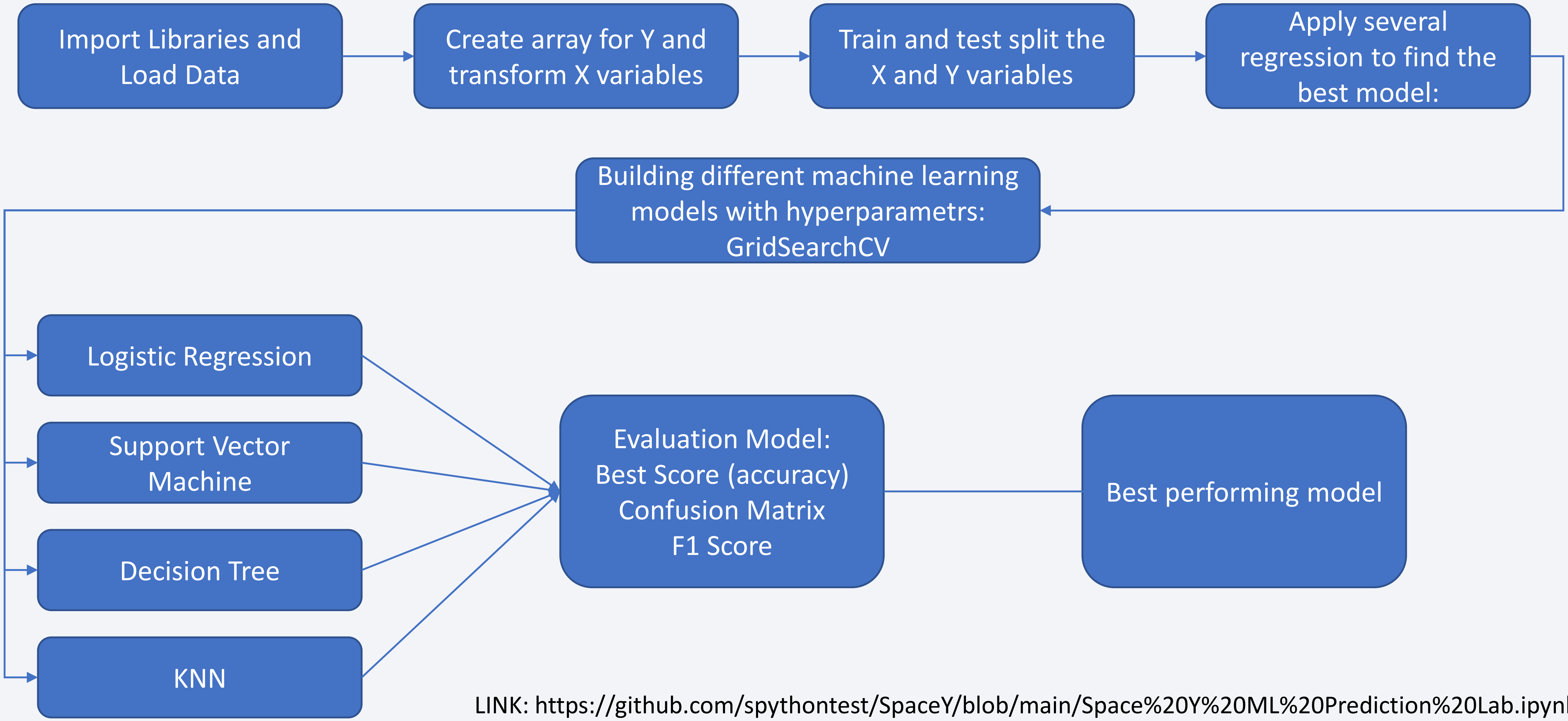
Build an Interactive Map with Folium

- We marked all of the launch sites on a map given their latitude and longitude coordinates with folium
- We assigned the markings for success/failed launches for each site on the map, 0 for failure and 1 for success. This can be done by adding color for failure and success and also adding the zoom/cluster feature
- We then calculated the distance between launch sites to other objects such as coast line, sea, railway, highway .etc in order to know important nearest site features that can support the launch program
- Link: https://github.com/spythontest/SpaceY/blob/main/Space%20Y%20Capstone%20Interactive%20Viz%20Analytics%20Dashboa.ipynb

Build a Dashboard with Plotly Dash

- We built an interactive dashboard with Plotly Dash
- Given the instructions, we built the pie charts that is interacting with given input of certain launch sites
- In addition, the input also interacts with scatter plot showing the relationship between the launch outcome and payload for different booster version.
- Link: https://github.com/spythontest/SpaceY/blob/main/Plotly%20Dash.ipynb

Predictive Analysis (Classification)



LINK: https://github.com/spythontest/SpaceY/blob/main/Space%20Y%20ML%20Prediction%20Lab.ipynb

Results

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

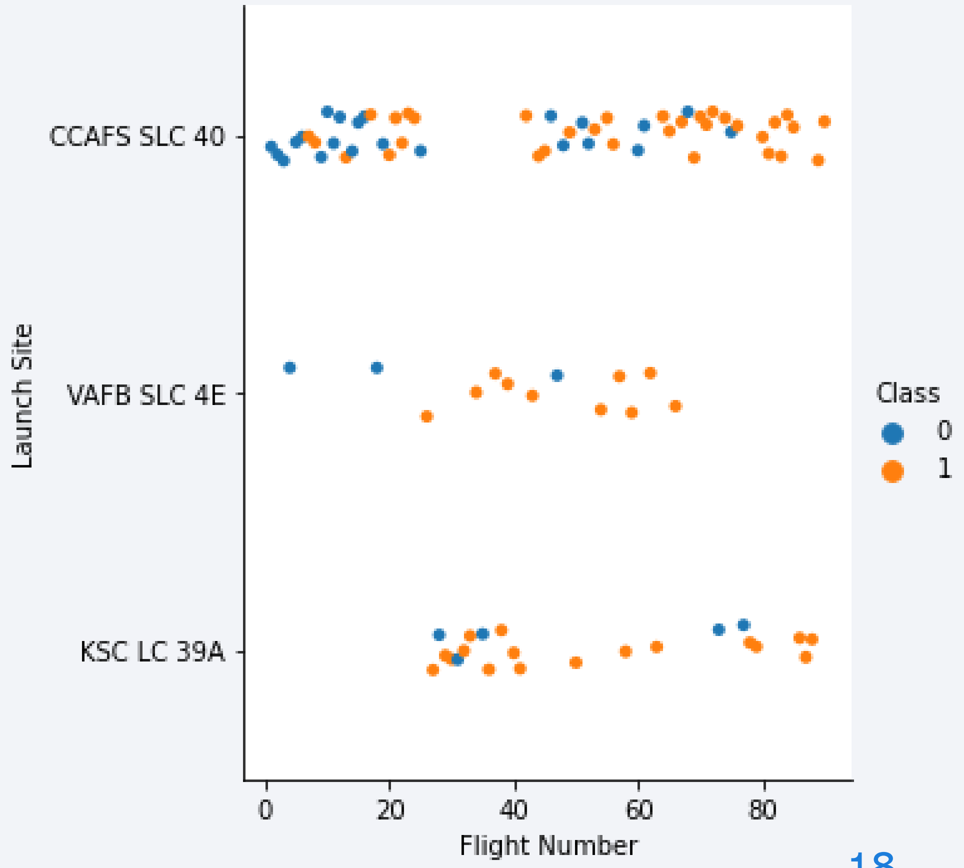
Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

- It is still inconclusive that the more flight numbers, the more success the launch is across all of the three launch sites

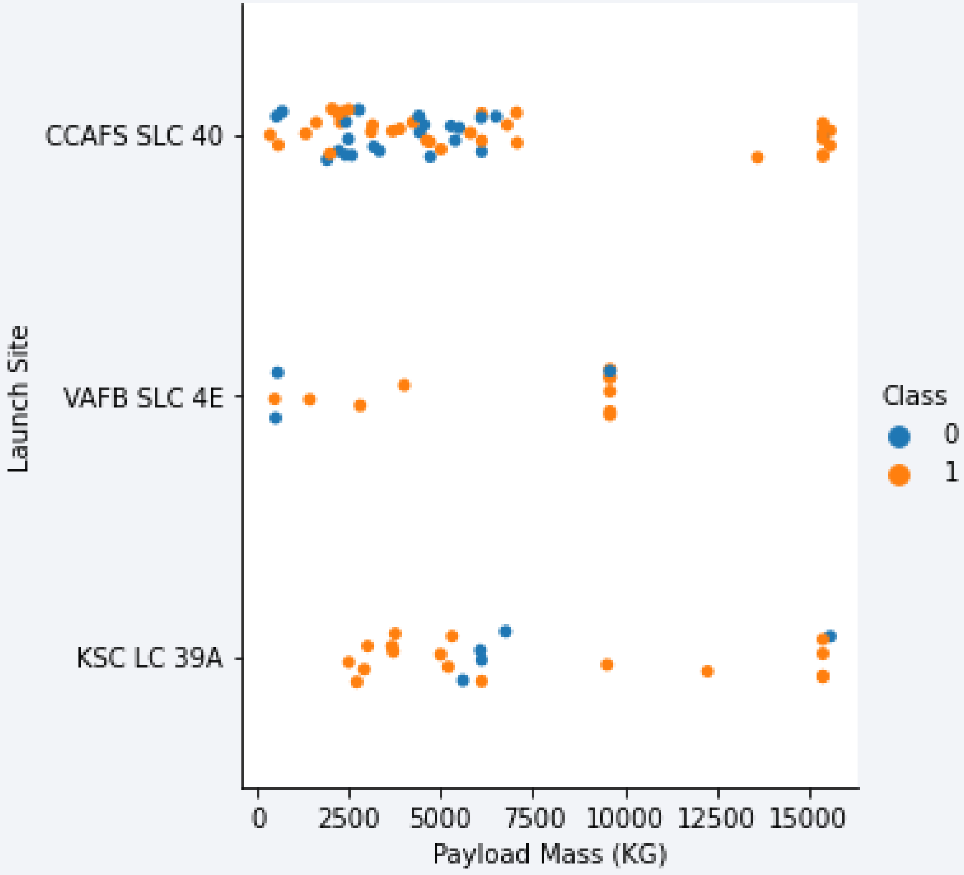
- VAFB SLCE 4E launch site have more successful launch percentage, nevertheless have less flight numbers compared to KSC LC39A



Payload vs. Launch Site

- Payload plays essential factor for a successful launch

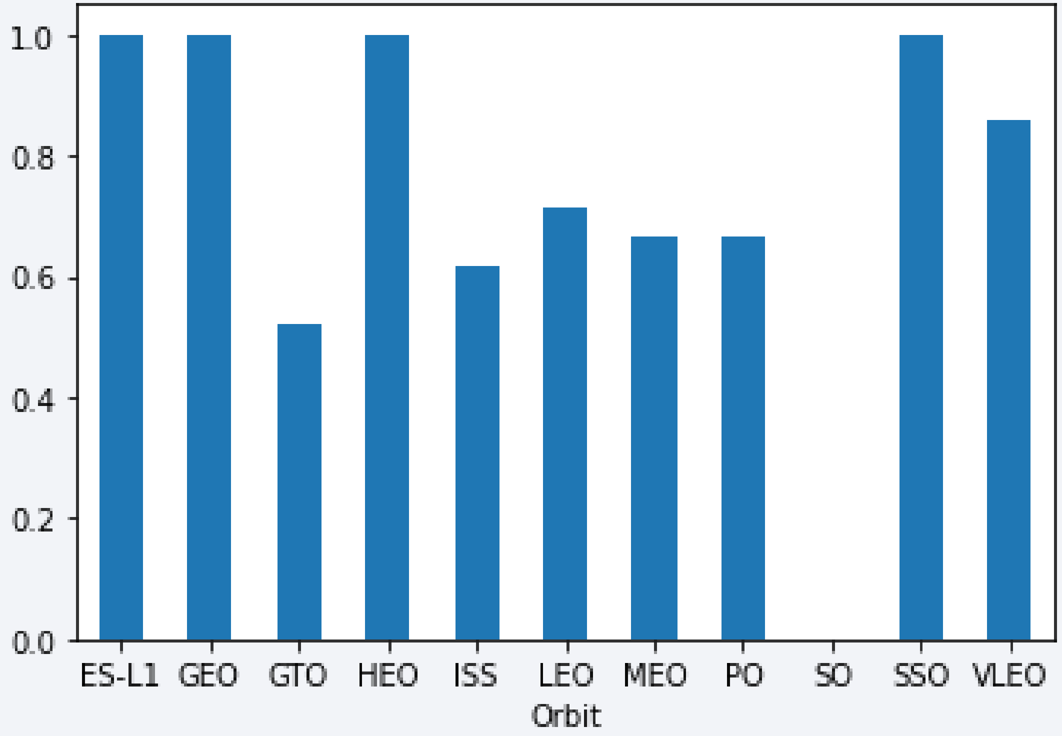
- It is shown that for payload that has more than 7500 KG has more successful launches compared less than that. Nevertheless, worth to note that launch site KSC LC 39A has more successful launches with lighter payload



Success Rate vs. Orbit Type

- Based on orbit type, there are several orbits that have successful launch rates

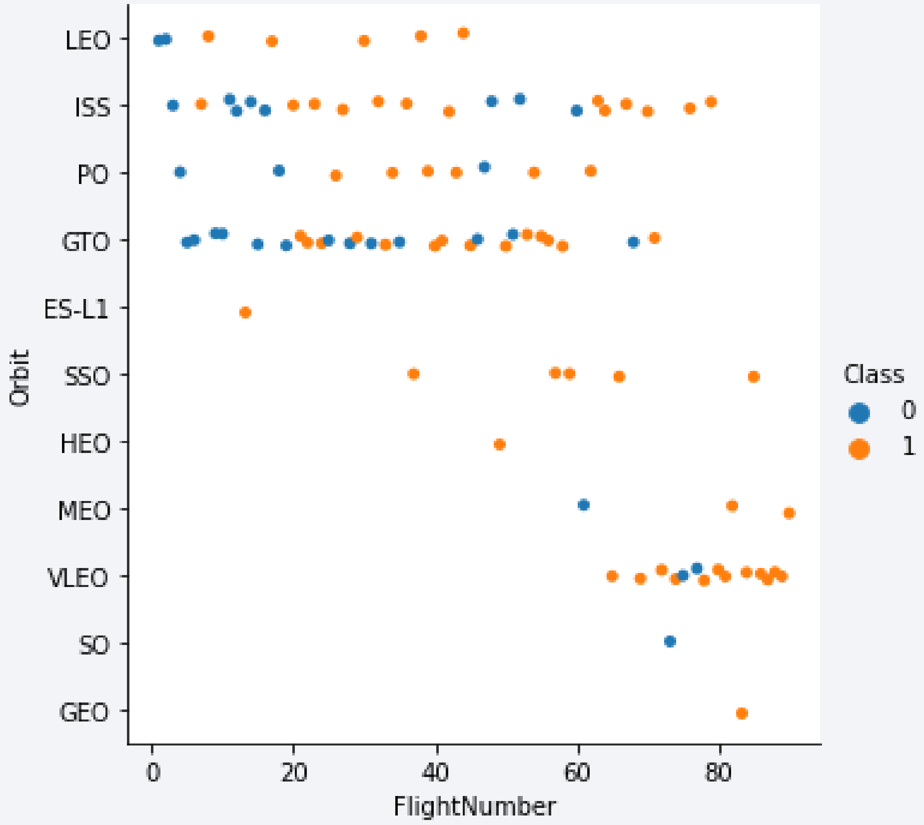
- Nevertheless, we still need to see the frequency per orbit to gain more confidence for the overall success rates



Flight Number vs. Orbit Type

- Subsequently, from the previous high success rates, it is shown here that the flights/flight numbers might not adequate as overall success rates

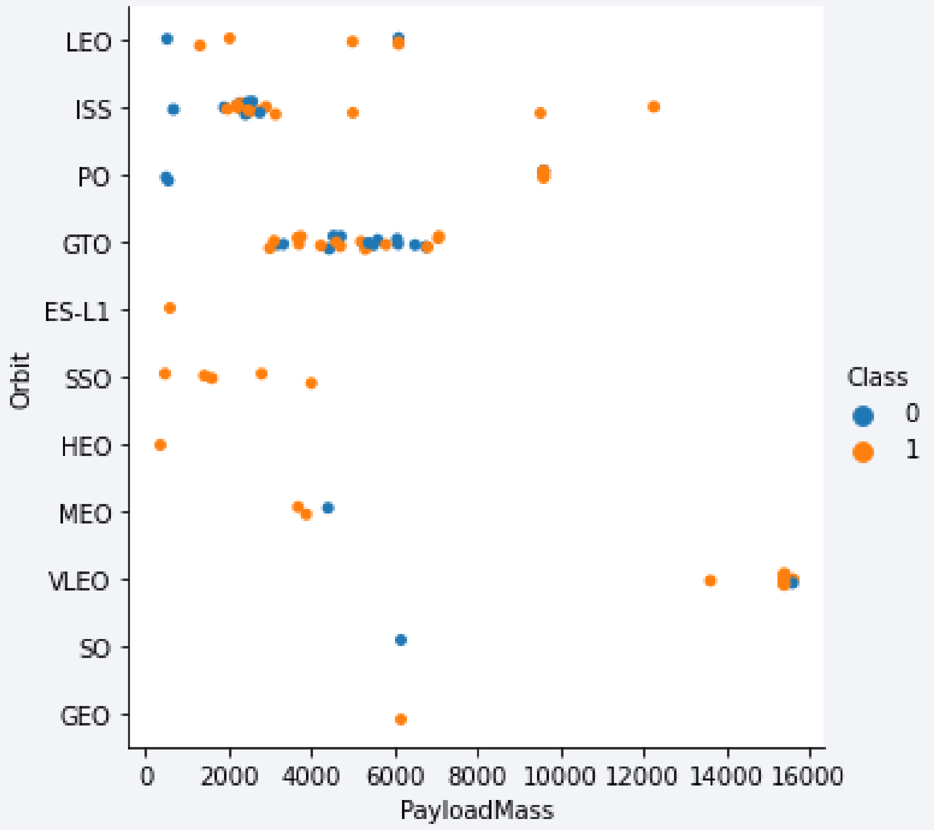
- We can see that there is a clear linear relationship for LEO orbit that the more flight numbers, the more success rates achieved. Meanwhile, others are still inconclusive



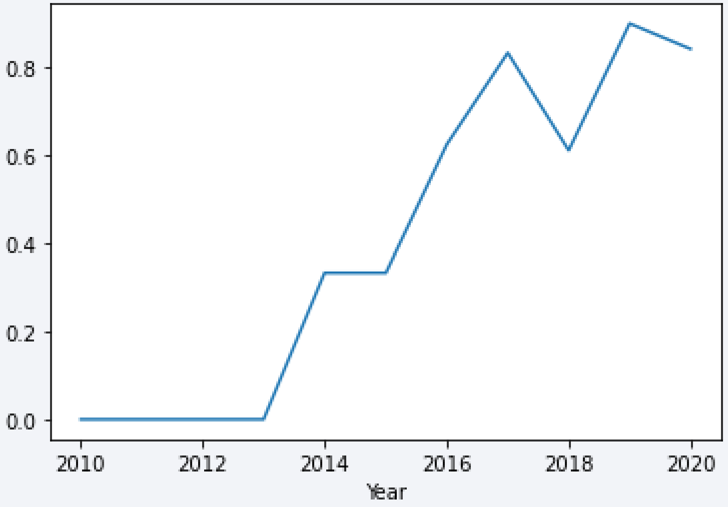
Payload vs. Orbit Type

- As previously mentioned, payload plays a role for the success rate based on orbit type as well

- We can see clearly that ISS and PO orbit type have positive landing rate for higher payload for more than 6000 KG



Launch Success Yearly Trend



- It is shown that the team on yearly basis since 2013, there has been improvement on the success landing rate percentage.

- Nevertheless, significant dip in 2018 should be analyzed for future learnings

All Launch Site Names

- There are four unique launch sites given their code names
- Using ‘DISTINCT’ feature on SELECT of SQL from the given table name

```
SELECT
        DISTINCT LAUNCH_SITE
FROM
        KQL46318.SPACEXTBL
```

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

Launch Site Names Begin with 'CCA'

- Using the ‘like’ feature for the characters that starts with CCA in LAUNCH SITE, we shall find the first five entries (LIMIT 5) details

DATE	TIME__UTC_	BOOSTER_VERSION	LAUNCH_SITE	PAYLOAD	PAYLOAD_MASS__KG_	ORBIT	CUSTOMER	MISSION_OUTCOME	LANDING__OUTCOME
04/06/2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
08/12/2010	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)
22/05/2012	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
08/10/2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
01/03/2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

```
SELECT
        *
FROM
        KQL46318.SPACEXTBL
WHERE
        LAUNCH_SITE like 'CCA%' LIMIT 5
```

Total Payload Mass

- Using the Sum feature and its alias, we can calculate the total payload in KG with certain filter in WHERE that the customer is NASA
- The total payload is at 45,596 KG

```
SELECT
        SUM (PAYLOAD_MASS__KG_) as PAYLOAD_KG
FROM
        KQL46318.SPACEXTBL
WHERE
        CUSTOMER = 'NASA (CRS)’
```

PAYLOAD_KG
45596

Average Payload Mass by F9 v1.1

- Space Y has many versions of boosters
- Using the AVG feature on SELECT with its alias and filter on WHERE utilizing ‘like’ for the characters on boosters F9 v1.1, we have the average payload for the version is 2,534 KG

```
SELECT
        AVG (PAYLOAD_MASS__KG_) as AVG_KG
FROM
        KQL46318.SPACEXTBL
WHERE
        BOOSTER_VERSION like 'F9 v1.1%'
```

AVG_KG
2534

First Successful Ground Landing Date

- Using the MIN feature on SELECT we can have the least value of the DATE column so that it indicates the very first successful ground landing date (filter on WHERE)

```
SELECT
        MIN (DATE)
FROM
        KQL46318.SPACEXTBL
WHERE
        LANDING__OUTCOME = 'Success (ground pad)'
```

1
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- As per visual analysis, the payload plays crucial role with its given load. Here we can have the success landing outcome on filter WHERE and which booster version that the rocket was using

```
SELECT
        BOOSTER_VERSION
FROM
        KQL46318.SPACEXTBL
WHERE
        PAYLOAD_MASS__KG_ between 4000 and 6000 AND
        LANDING__OUTCOME = 'Success (drone ship)’
```

BOOSTER_VERSION
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
F9 FT B1031.2

Total Number of Successful and Failure Mission Outcomes

- Using the MISSION_OUTCOME column, we can count/calculate how many successful missions based on the outcome. Here we find there are 99 successful missions while there was only one failure and one successful mission without clear payload status

```
SELECT
        MISSION_OUTCOME,
        COUNT (MISSION_OUTCOME) as Total
FROM
        KQL46318.SPACEXTBL
GROUP BY
        MISSION_OUTCOME
```

MISSION_OUTCOME	TOTAL
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

Boosters Carried Maximum Payload

- Here we can enlist the booster versions based on the its maximum payload and we list them using descending order.

```
SELECT
        DISTINCT BOOSTER_VERSION,
        MAX (PAYLOAD_MASS__KG_) as maximum
FROM
        KQL46318.SPACEXTBL
GROUP BY
        BOOSTER_VERSION
ORDER BY
        maximum DESC;
```

BOOSTER_VERSION	MAXIMUM
F9 B5 B1048.4	15600
F9 B5 B1048.5	15600
F9 B5 B1049.4	15600
F9 B5 B1049.5	15600
F9 B5 B1049.7	15600

2015 Launch Records

- In 2015 there were only two failed landing outcome from two booster version B1012 and B1015 at the same launch site
- The result derived from the specific filter features (landing outcome equals to failures and also ‘like’ feature to capture the year 2015)

```
SELECT
        BOOSTER_VERSION,
        LAUNCH_SITE
FROM
        KQL46318.SPACEXTBL
WHERE
        LANDING__OUTCOME = 'Failure (drone ship)' AND
        DATE like '2015%'
```

BOOSTER_VERSION	LAUNCH_SITE
F9 v1.1 B1012	CCAFS LC-40
F9 v1.1 B1015	CCAFS LC-40

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

- With WHERE filter on the date given between the date 2010-06-04 and 2017-03-20 by the landing outcome result in descending order, we can count specifically in each category/group how many the outcome. It seems that ‘no attempt’ was the most frequent on that date range.

```
SELECT
        LANDING__OUTCOME,
        COUNT (LANDING__OUTCOME) as Total
FROM
        KQL46318.SPACEXTBL
WHERE
        DATE between '2010-06-04' and '2017-03-20'
GROUP BY
        LANDING__OUTCOME
ORDER BY
        Total DESC
```

LANDING__OUTCOME	TOTAL
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

Section 3

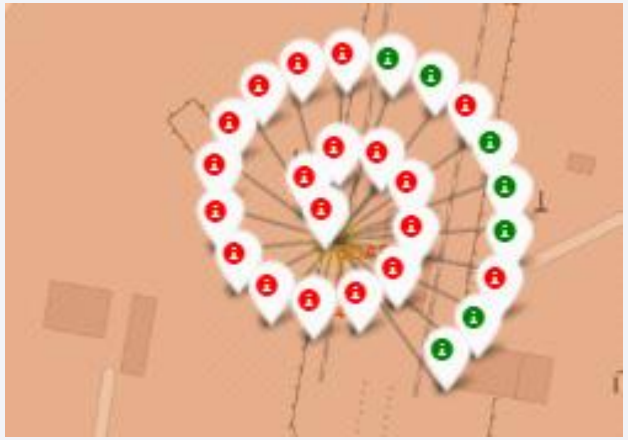
Launch Sites Proximities Analysis

Launches Sites Dominated in Florida, Near NASA HQ

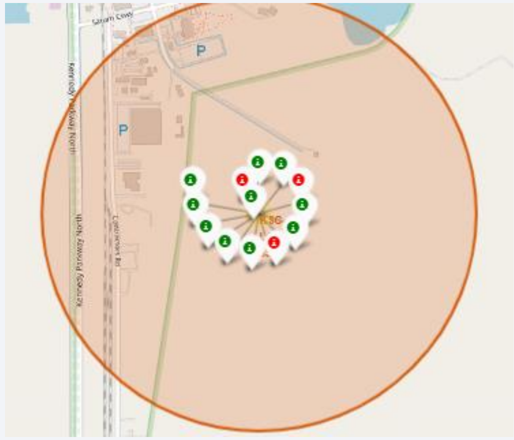
- Most of the launches are held near to the NASA headquarters in Florida
- Only at the other side of the country (West) KSC LC-39A was held in California for only 13 launches
- We shall look the impact on this different launch sites to the different success rates

Launch Site KSC LC-39A had the highest success rate

CCAFS SLC-40
From 7 attemps,
only 3 successful



CCAFS LC-40
From 26 attempts,
only 7 successful



KSC LC-39A
From 13 Attempts,
only 3 were
unsuccessful

KSC LC-39A
From 13 Attempts,
only 3 were
unsuccessful

- Of all of the four sites for launches, KSC LC-39A have the highest success rate
- This need to be further analyzed since KSC LC-39A location is in CA, not in NASA HQ Florida. Also, further factors that determine the higher success rate for this site compared to others.

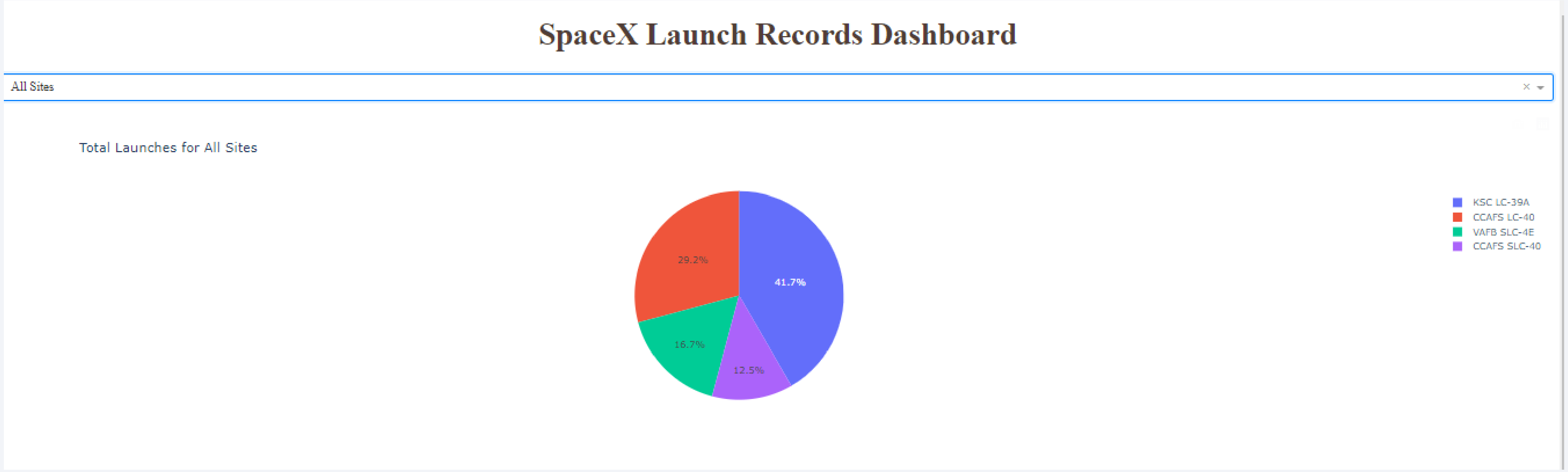
Coast Lines are the Common Nearest Object

- From the given calculation and pictures below, these are the two nearest object which are the coast line to the launch sites
- Note that the highway, roads and railway are also to be noted for the access/logistics to the launch sites
- Meanwhile, the coast lines are common object for the launch sites as the nearest so that the organizer have contingency plan for unsuccessful landings

Section 4

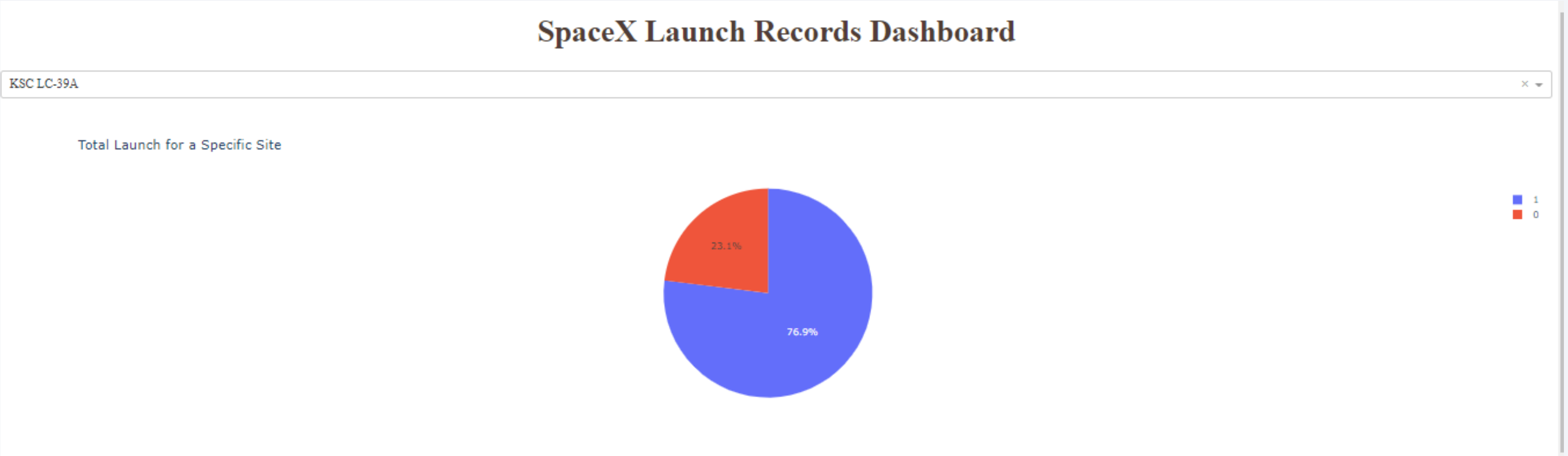
Build a Dashboard with Plotly Dash

Total Launches for All Sites



- Of all the successful launches, the larger part of them fall in to site KSC LC-39A about 41%.
- From previous findings, this is align that the aforementioned site have higher rate successful launches due to it also have positive rate on lighter payload launches.

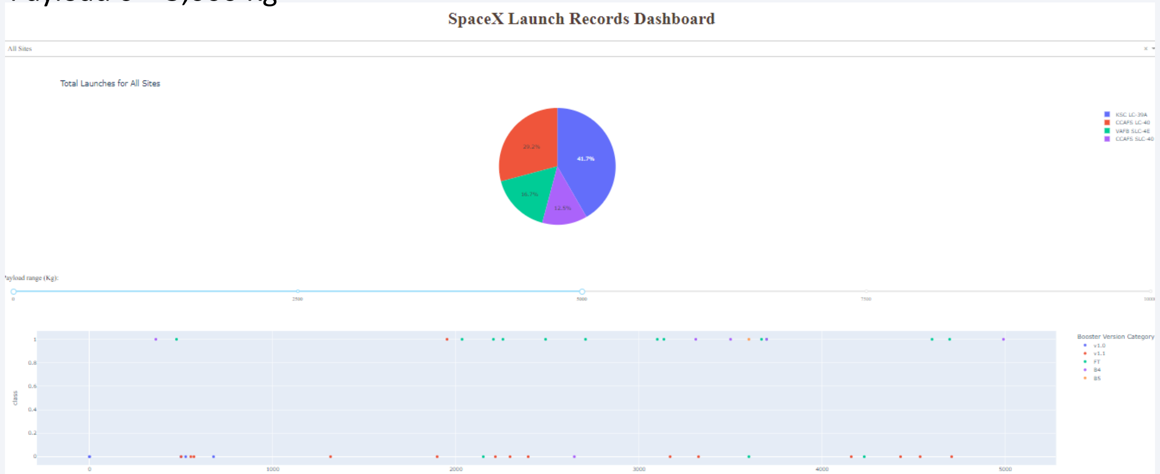
Highest Successful Launch Site: KSC LC-39A



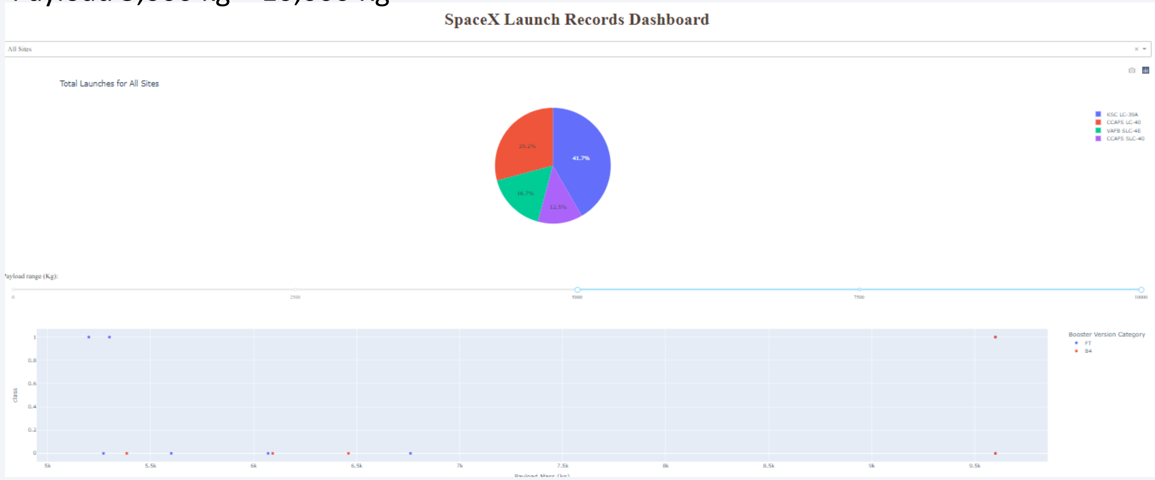
- KSC LC-39A has the highest successful launch rate at 76% due to several reasons:
  - Despite higher payload contribute to higher success rate, the site can accommodate the lighter payload to be successful
  - The site started with higher flight number, thus, more lesson learned from earlier flight numbers from previous sites to be successful

Payload Does Play The Role for Success Rate

Payload 0 – 5,000 Kg



Payload 5,000 kg – 10,000 Kg



- For lighter payload (<5000 KG), it is inconclusive that certain payload could affect the success rate for the launch. Meanwhile, for payload ranges above 5000 KG and below 10,000 KG, it is found that all of the launches are successful for payload near 10,000 KG (9,500 KG)
- For the current state, ceteris paribus, it would be ideal to have higher success rate, the payload needs to be put at near 10,000 KG

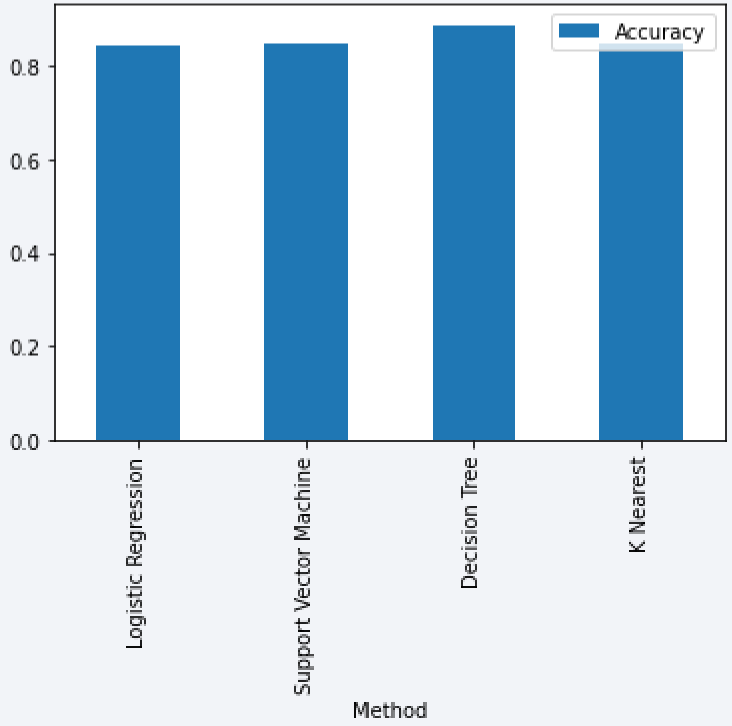
Section 5

Predictive Analysis (Classification)

Classification Accuracy

- Of all the model (Train), the Decision Tree has the highest accuracy rate at 89%.

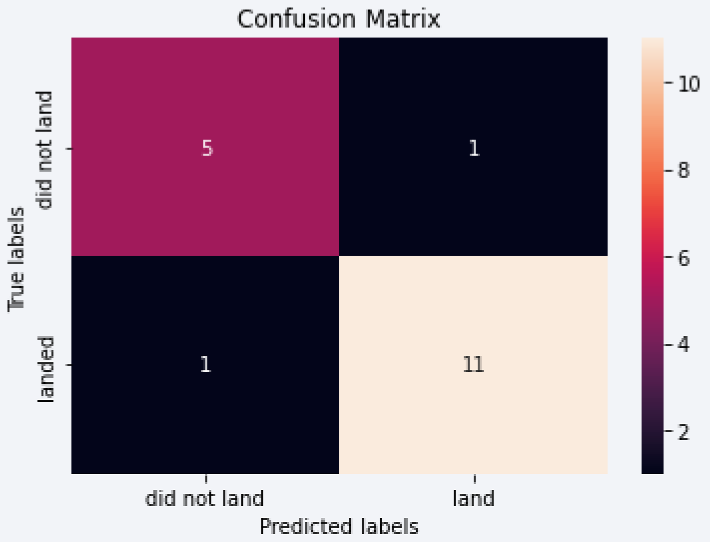
Method	Accuracy
Logistic Regression	0.846429
Support Vector Machine	0.848214
Decision Tree	0.889286
K Nearest	0.848214



Confusion Matrix

- The decision tree model has the accuracy of 89% and has f1-score of 83% on modelling the failure rate and 92% for success rate
- Overall, we can conclude that we can do good prediction on successful rate in order to further the analysis for cost computation

	precision	recall	f1-score	support
0	0.83	0.83	0.83	6
1	0.92	0.92	0.92	12
accuracy			0.89	18
macro avg	0.88	0.88	0.88	18
weighted avg	0.89	0.89	0.89	18



Conclusions

- As the company launches more flights each year, they learn to improve the success rate with valuable lessons and inputs from 2013 to 2020
- There are several factors that correlate or in line with successful launch rate:
  - Location : KSC LC-39A has highest success rate, after many flight numbers conducted from other launch sites
  - Orbits : LEO and VLEO orbits have positive learnings on success rate as with more flight numbers conducted and payload increased
  - Payload : Plays an important role for the success rate whereas the more payload at above 7500 KG, the more success rate can be achieved. Nevertheless, on a certain site, there is still lighter payload that can achieve positive success rate.
- We can predict the success rate classification well with Decision Tree Model Classifier for the best machine learning algorithm/model for this task

Thank you!