IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Sparsh Gautam
11/4/2021

Table of Contents

Executive Summary

- Summary of methodologies
  - Data Collection (API)
  - Data Collection (Web Scraping)
  - Data Wrangling using EDA
  - EDA with SQL
  - EDA with Data Visualization
  - Visual Analytics with Folium
  - Machine Learning Prediction
- Summary of all results
  - Used Visualization to Show Data
  - Analyzed trends of Data and provided justification conclusions
  - Machine Learning Prediction for SpaceX like operations

Introduction

- Project background and context
  - We predicted 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
  - Project seeks to aim cost, details and success rate of SpaceX’s Falcon9 to determine possibility of SpaceY to achieve low-cost launches based on orbits, payload mass, overall cost etc.
  - Task is to determine the cost of each launch depending on the specifics of the launch such as booster, payload mass, etc.
  - Also required to verify from SpaceX’s data whether first launch will be reused – using machine learning principles as opposed to rocket science

Section 1

Methodology

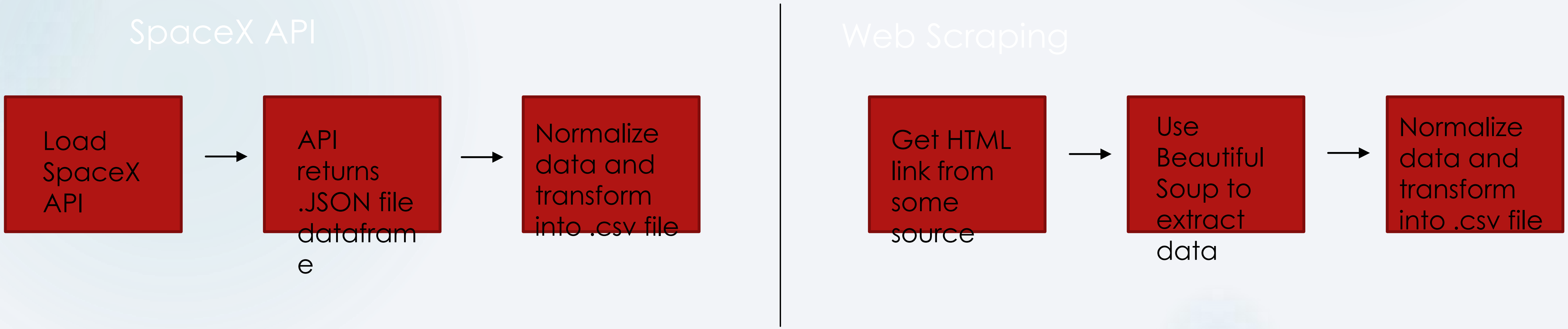
Methodology

Executive Summary

- Data collection methodology:
  - Web scraping (SpaceX Wikipedia Data)
  - SpaceX API Data collection and cleaning
- Perform data wrangling
  - One Hot encoding for Machine Learning setup
- Perform exploratory data analysis (EDA) using visualization and SQL
  - Plotting Bar graphs, scatter graphs for effective representation of data trends between a range of different variables (dependent & independent)
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection

- Started with Data collection on the provided SpaceX API to extract key information to generate trends and clean information where it was missing
- This API gave us data about launches, including information about the rocket used, payload delivered, launch specifications, landing specifications, and landing outcome.
- Using this data, the ML model was used to predict whether SpaceY's first launch would achieve success.
- Used Web scraping to obtain increased information of first launch of Falcon9 SpaceX launches from Wikipedia sources to build upon data sets which were initialized
- Built upon the web scraping method using Python library's BeautifulSoup



Data Collection – SpaceX API

- Data collection process with SpaceX REST calls using key phrases and flowcharts is shown on the right

- SpaceX API Github link here to get better understanding of calls and phrases

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

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

```
# 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

```
# Get the head of the dataframe
data.head()
```

	static_fire_date_utc	static_fire_date_unix	net	window	rocket

```
# Call getLaunchSite
getLaunchSite(data)
```

```
# Call getPayloadData
getPayloadData(data)
```

```
# Call getCoreData
getCoreData(data)
```

```
data_falcon9 = data[data['BoosterVersion'] != 'Falcon 1']
data_falcon9.head()
```

```
[91]: data_falcon9.loc[:,'FlightNumber'] = list(range(1, data_falcon9.shape[0]+1))
      data_falcon9.shape
```

We can now export it to a **CSV** for the next section,but to make the answers consistent, in the next lab we will provide data in a pre-selected date range.

```
data_falcon9.to_csv('dataset_part\_1.csv', index=False)
```

Data Collection - Scraping

- Data collection process with Webscraping using Wikipedia link on SpaceX. Key phrases and flowcharts are shown on the right

- [Github link for Webscraping]()

```
In [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

```
In [9]: # use requests.get() method with the provided static_url
        # assign the response to a object
        response = requests.get(static_url).text
        response
```

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

```
In [13]: # Use the find_all function in the BeautifulSoup object, with element type `table`
         # Assign the result to a list called `html_tables`

         html_tables = BeautifulSoup.find_all()
```

TASK 3: Create a data frame by parsing the launch HTML tables

We will create an empty dictionary with keys from the extracted column names in the previous task. Later, this dictionary will be converted into a Pandas dataframe

```
In [35]: launch_dict= dict.fromkeys(column_names)

         # Remove an irrelvant column
         #del launch_dict['Date and time ( )']

         # Let's initial the launch_dict with each value to be an empty list
         launch_dict['Flight No.'] = []
         launch_dict['Launch site'] = []
         launch_dict['Payload'] = []
         launch_dict['Payload mass'] = []
         launch_dict['Orbit'] = []
         launch_dict['Customer'] = []
         launch_dict['Launch outcome'] = []
         # Added some new columns
         launch_dict['Version Booster']=[]
         launch_dict['Booster landing']=[]
         launch_dict['Date']=[]
         launch_dict['Time']=[]
```

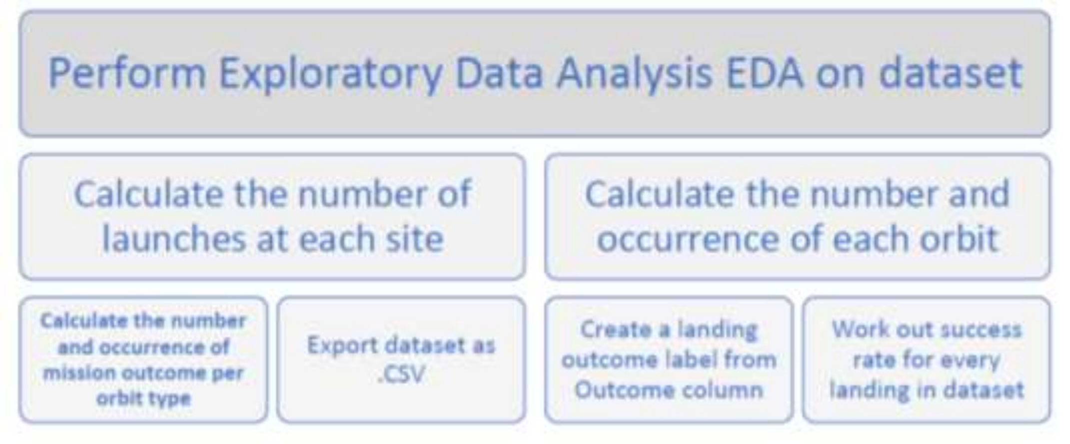
```
In [39]: df=pd.DataFrame(launch_dict)

In [40]: df
```

Data Wrangling

- Data Wrangling allows us to perform Exploratory Data Analysis (EDA) to find some patterns in the data and determine what would be the label for training supervised models. We mainly converted the outcomes into Training Labels with 1 means the booster successfully landed 0 means it was unsuccessful.

Flow chart steps



- [Data Wrangling Github Link here]

EDA with Data Visualization

- Graphs that were plotted were:
  - Scatter Plots: Great at establishing trends between numeric variables
    - Payload Mass vs Flight Number
    - Launch Site vs Flight Number
    - Payload Mass vs Launch Site
    - Payload Mass vs Orbit
  - Bar Plots: Effective at categorizing data; here success rate could be categorized well
    - Success Rate vs Orbit
  - Line Plots: Used to track changes over time thus explaining the use of line graphs when tracking years on x axis
    - Success Rate vs Year
- EDA Data Visualization Github link here

EDA with SQL

- These were the 10 important queries used 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 successful landing outcome in ground pad was achieved.
  - 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 the names of the booster_versions which have carried the maximum payload mass. Use a subquery
  - List the failed landing_outcomes in drone ship, their booster versions, and launch site names for 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
- EDA with SQL Github link here

Build an Interactive Map with Folium

- The launch success rate may depend on many factors such as payload mass, orbit type, and so on. It may also depend on the location and proximities of a launch site, i.e., the initial position of rocket trajectories. Finding an optimal location for building a launch site certainly involves many factors and hopefully we could discover some of the factors by analyzing the existing launch site locations.
- To visualize the Launch Data, took the Latitude and Longitude Coordinates at each launch site and added a Circle Marker around each launch site with a label of the name of the launch site. Then assigned the dataframe launch_outcomes(failures, successes) to classes with 0 and 1 with Green and Red markers on the map in a MarkerCluster()
- Folium Github link here

Build a Dashboard with Plotly Dash

- **The dashboard is built with Flask and Dash web framework.**
- **Main graph used were:**
  - **Pie charts: Effective in showing various proportions of different affecting parameters**
  - **Scatter Graph showing the relationship with Outcome and Payload Mass (Kg) for the different Booster Versions**
    - It shows the relationship between two variables.
    - It is the best method to show you a non-linear pattern.
    - Observation and reading are straightforward.
- Plotly Dash Lab here

Predictive Analysis (Classification)

- Space X 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 Space X 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 space X for a rocket launch. This explains the ML algorithm and model

- Steps Taken:

Create column for class → Standardize Data → Split data into train & test sets and analyse accuracy scores

- Machine Learning Github Link here

Results

These methods were used in results analysis:

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

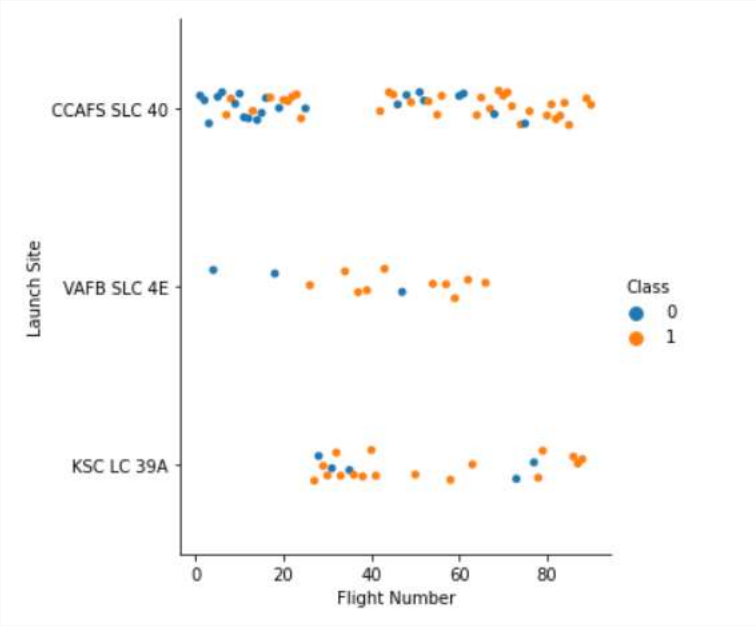
Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

- Scatter plot of Flight Number vs. Launch Site

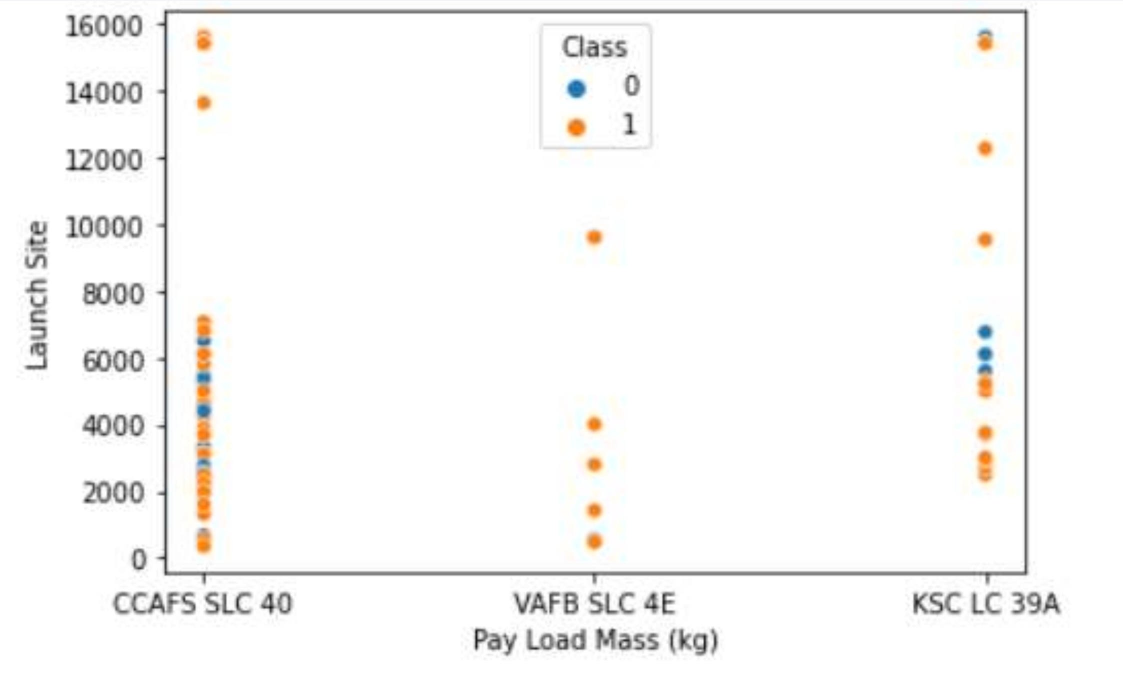
- The scatter plot shows that increased number of flights at a launch site increases success rate of the launch



Payload vs. Launch Site

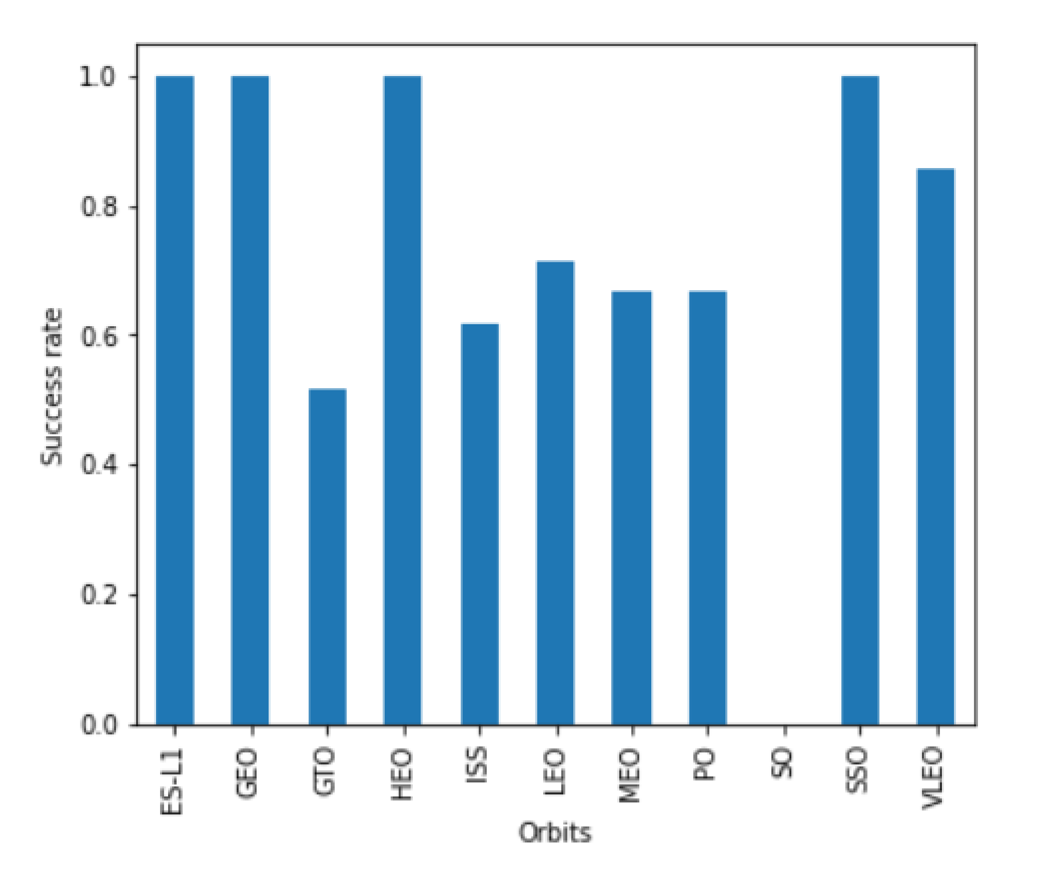
- Scatter plot of Payload vs. Launch Site

- On the whole more successes, but greater success rate for a higher payload mass



Success Rate vs. Orbit Type

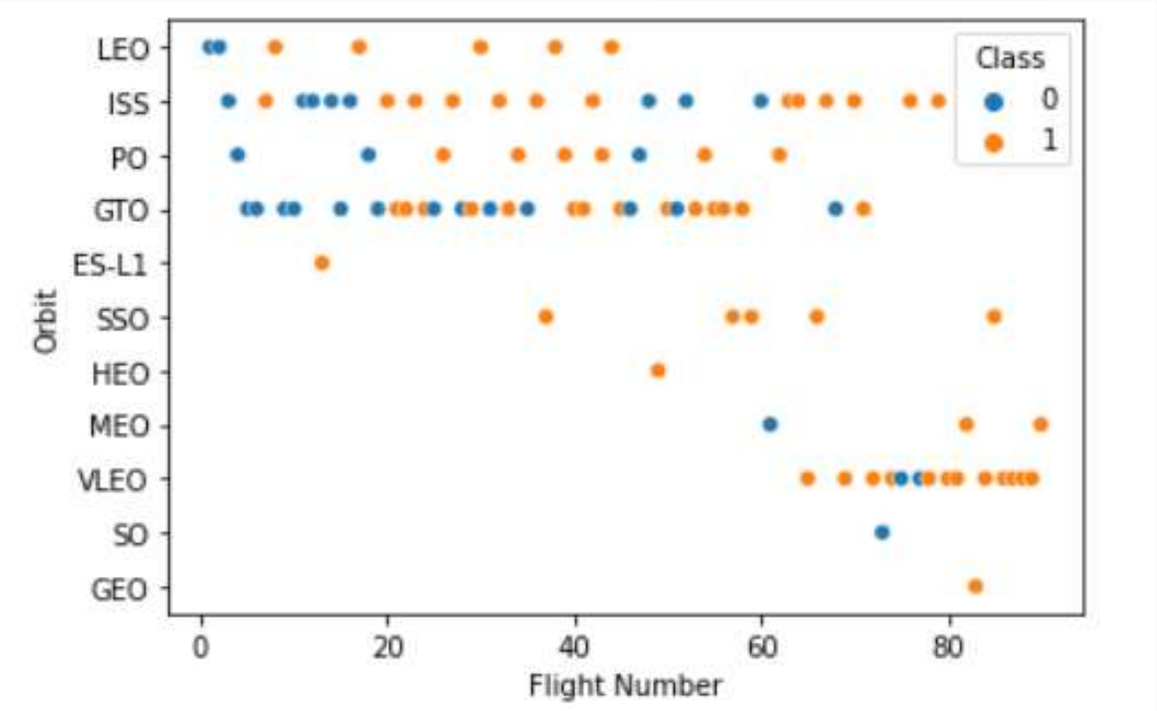
- Bar chart for the success rate of each orbit type
- ES-L1, GEO, HEO and SSO have best success rates for those specific orbits



Flight Number vs. Orbit Type

- Scatter plot of Flight number vs. Orbit type

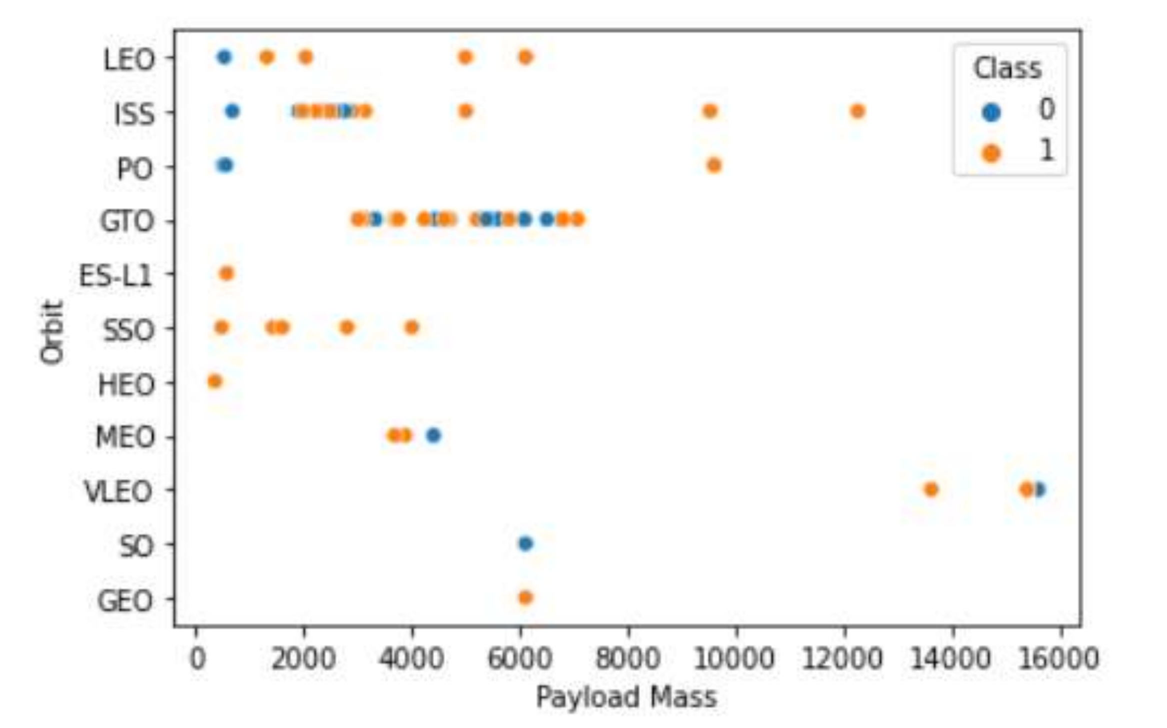
- Increased flight numbers does lead to better success rates.



Payload vs. Orbit Type

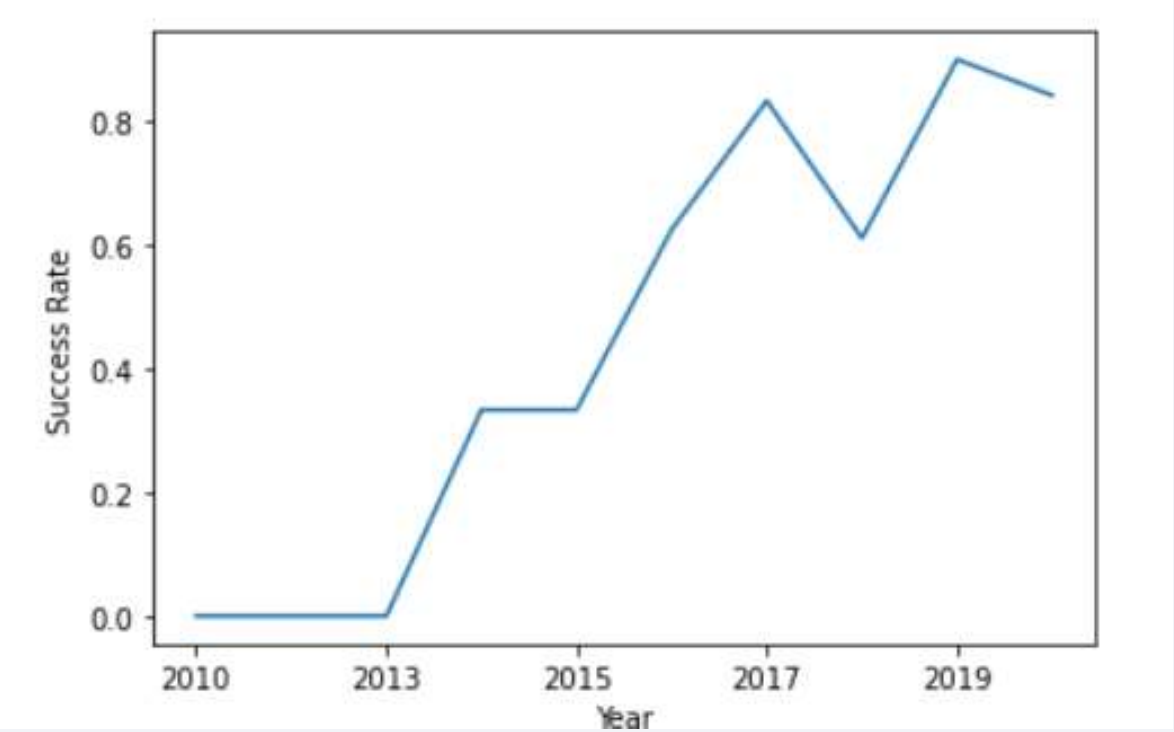
- Scatter point of payload vs. orbit type

- A balance of payload mass is in general the best approach when launching an orbit because excess mass can at times have a negative effect



Launch Success Yearly Trend

- Line chart of yearly average success rate
- In totality there has been a net increase in the success rate of launches starting from 2010 to 2020 (rising from 0.0 to now about 0.8-0.9)



All Launch Site Names

- Used query:
  - %sql SELECT DISTINCT launch_site FROM SPACEXTBL
  - This selects distinct values from launch_site column and SPACEXTBL table

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

Launch Site Names Begin with 'CCA'

- Used query:
  - %sql SELECT * FROM SPACEXTBL WHERE launch_site LIKE '%CCA%' LIMIT 5
  - Select all columns in the case where the values in launch site column are similar to or contain the string ‘CCA’ (limited to five)

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	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	00:35:00	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

- Used query:
  - %sql SELECT SUM(payload_mass__kg_) FROM SPACEXTBL WHERE customer LIKE '%NASA (CRS)%';
  - Sum up all values of mass where the launch site is from NASA CRS

1
48213

Average Payload Mass by F9 v1.1

- Used Query
  - %sql SELECT AVG(payload_mass__kg_) FROM SPACEXTBL WHERE booster_version LIKE '%F9 v1.%';
  - Average payload mass of boosters that are F9 v1

1
1986

First Successful Ground Landing Date

- Used query
  - %sql SELECT MIN(DATE) FROM SPACEXTBL WHERE landing__outcome LIKE '%Success (ground pad)%’
  - Select first date (min fnct) when landing outcome was a success

1
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- Used query
  - %sql SELECT booster_version FROM SPACEXTBL WHERE payload_mass__kg_>4000 AND payload_mass__kg_<6000
  - Select all booster versions where 4000 kg < payload mass < 6000 kg

booster_version
F9 v1.1
F9 v1.1 B1011
F9 v1.1 B1014
F9 v1.1 B1016
F9 FT B1020
F9 FT B1022
F9 FT B1026
F9 FT B1030
F9 FT B1021.2
F9 FT B1032.1
F9 B4 B1040.1
F9 FT B1031.2
F9 B4 B1043.1
F9 FT B1032.2
F9 B4 B1040.2
F9 B5 B1046.2
F9 B5 B1047.2
F9 B5B1054
F9 B5 B1048.3
F9 B5 B1051.2
F9 B5B1060.1
F9 B5 B1058.2
F9 B5B1062.1

Total Number of Successful and Failure Mission Outcomes

- Used Query:
  - %%sql
  - SELECT s.success, f.failure FROM
  - (SELECT COUNT(mission_outcome) as success FROM SPACEXTBL WHERE mission_outcome LIKE '%Success%') s,
  - (SELECT COUNT(mission_outcome) as failure FROM SPACEXTBL WHERE mission_outcome LIKE '%Failure%') f

success	failure
100	1

Boosters Carried Maximum Payload

- Used Query
  - %sql SELECT booster_version FROM SPACEXTBL WHERE (SELECT MAX(payload_mass__kg_) FROM SPACEXTBL)
  - Select booster_version which has max payload mass (and there were several)

booster_version
F9 v1.0 B0003
F9 v1.0 B0004
F9 v1.0 B0005
F9 v1.0 B0006
F9 v1.0 B0007
F9 v1.1 B1003
F9 v1.1
F9 v1.1
F9 v1.1
F9 v1.1
F9 v1.1
F9 v1.1 B1011
F9 v1.1 B1010
F9 v1.1 B1012
F9 v1.1 B1013
F9 v1.1 B1014

2015 Launch Records

- Used query
  - %%sql
  - SELECT landing__outcome, booster_version, launch_site, DATE from SPACEXTBL WHERE landing__outcome LIKE '%Failure (drone ship)%' AND DATE LIKE '%2015%'

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

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

- Used query
  - %%sql
  - SELECT landing__outcome, count(landing__outcome) as freq FROM SPACEXTBL WHERE DATE BETWEEN '2010-04-04' AND '2017-03-20' GROUP by landing__outcome ORDER by freq desc

landing__outcome	freq
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

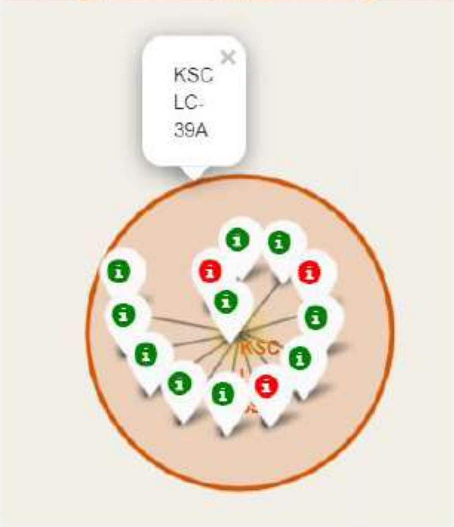
Section 4

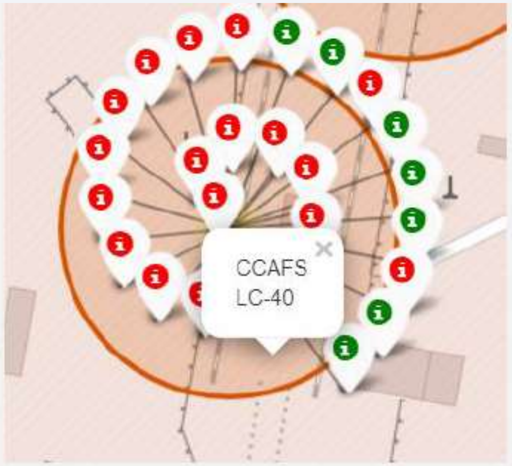
Launch Sites Proximities Analysis

All launch sites (global map)

We can see that the SpaceX launch sites are in the United States of America coasts. Florida and California

Color Labelled Launch Sites



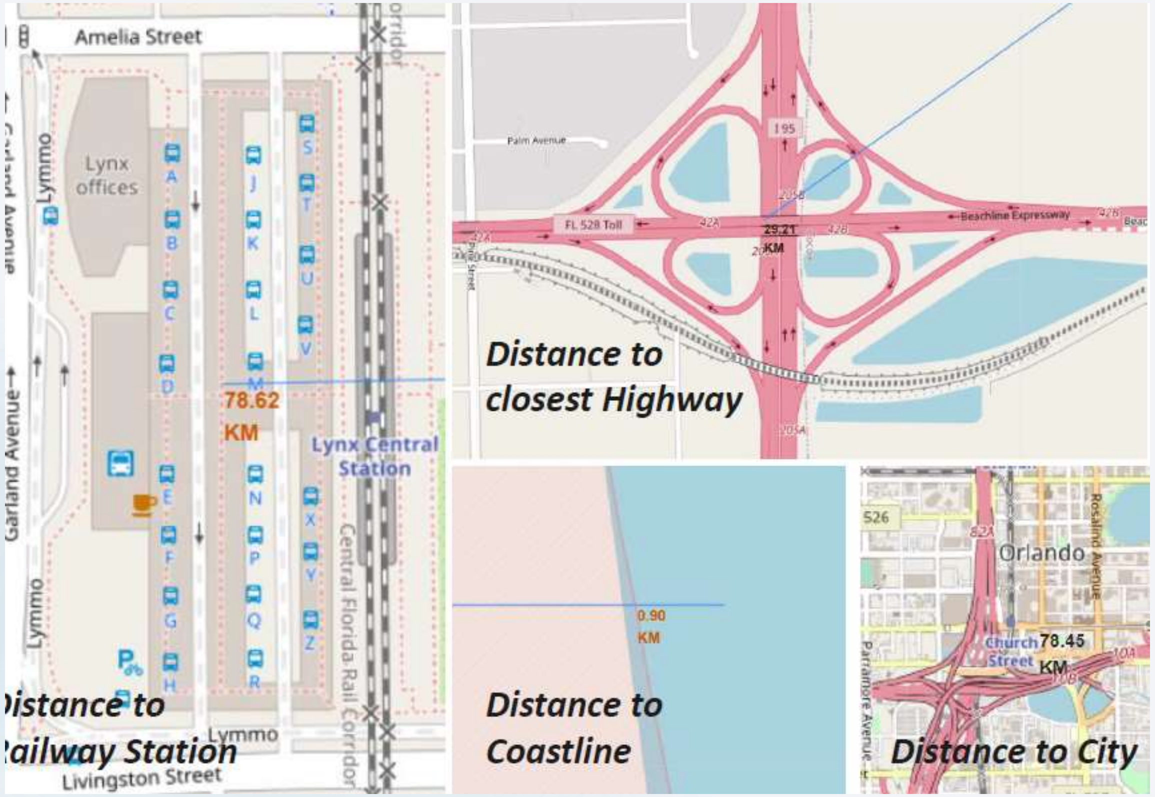


Florida Launch Sites

Green Marker shows successful Launches and Red Marker shows Failures

California Launch Site

Haversine Formula Map Plot

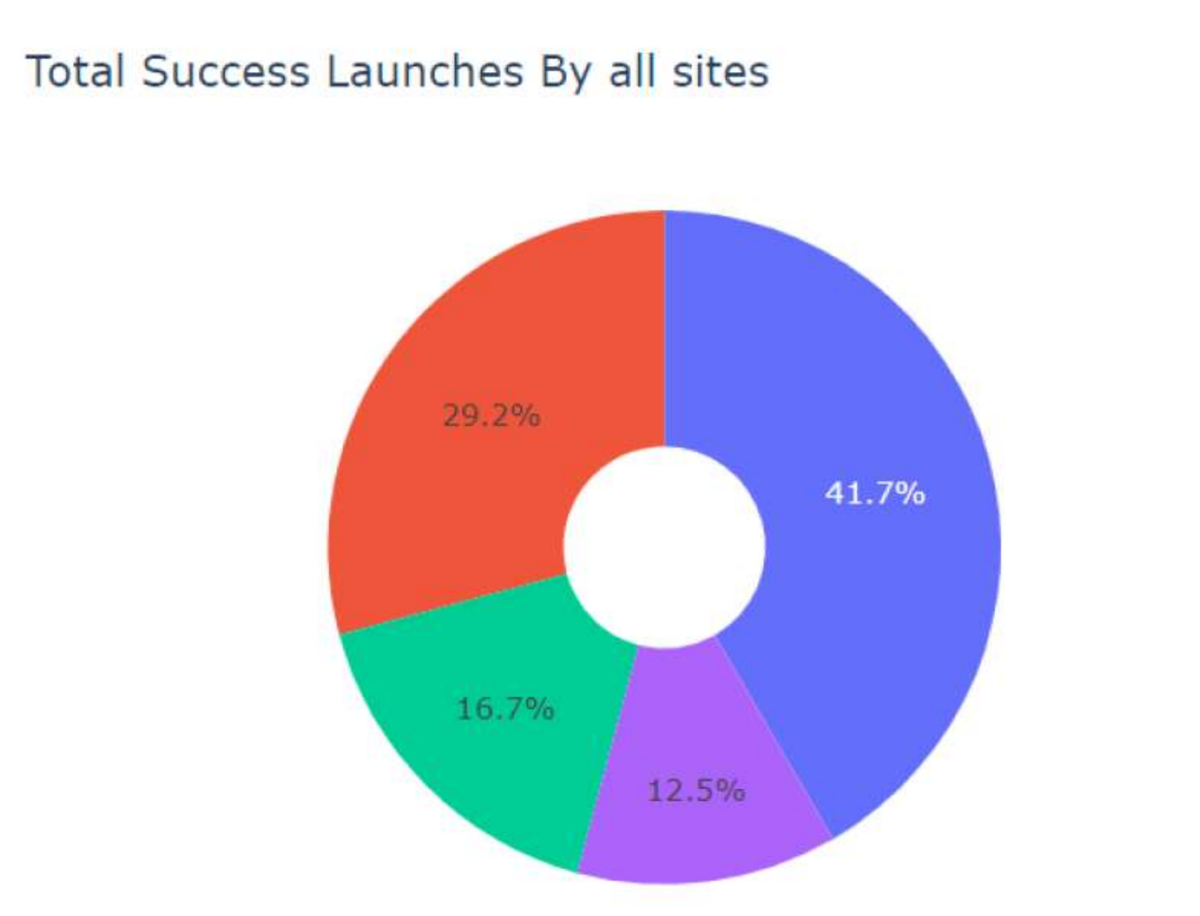


Are launch sites in close proximity to railways? No
Are launch sites in close proximity to highways? No
Are launch sites in close proximity to coastline? Yes
Do launch sites keep certain distance away from cities? Yes

Section 5

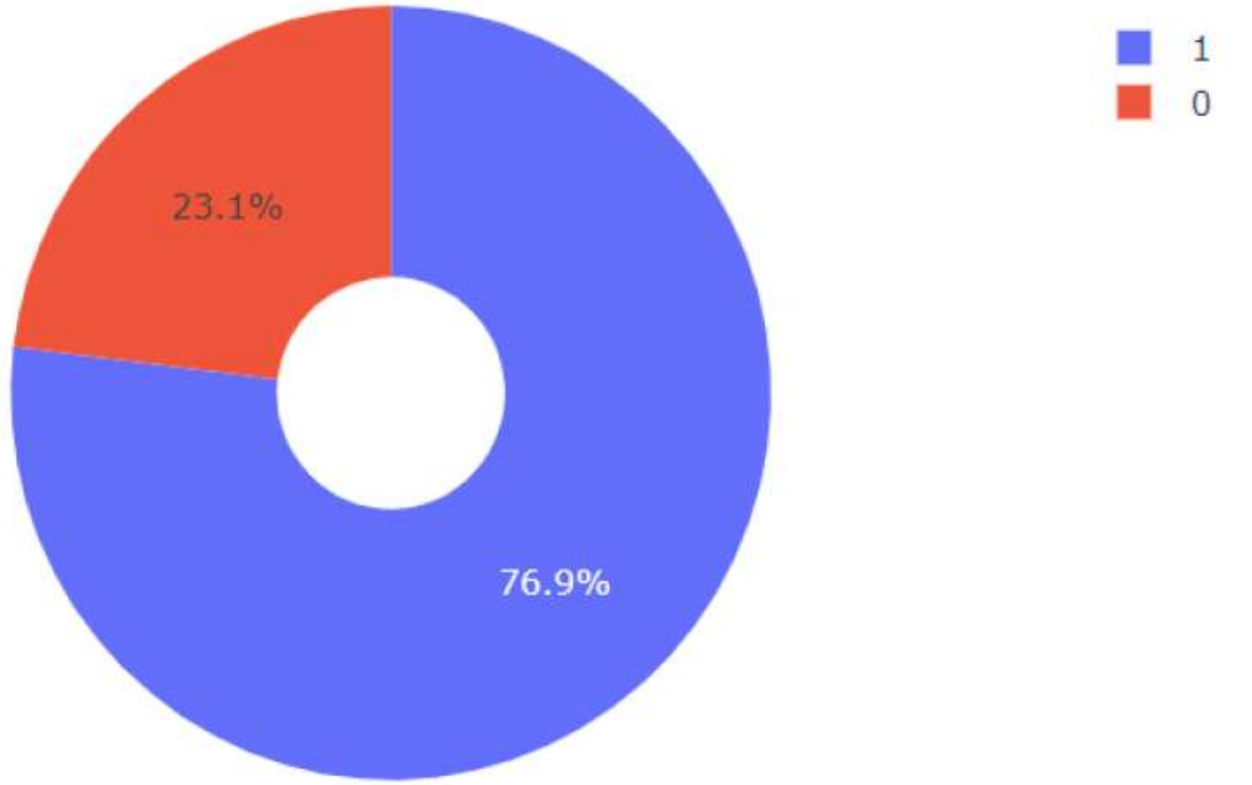
Build a Dashboard with Plotly Dash

Pie Chart Dashboard – Successful Launch Sites



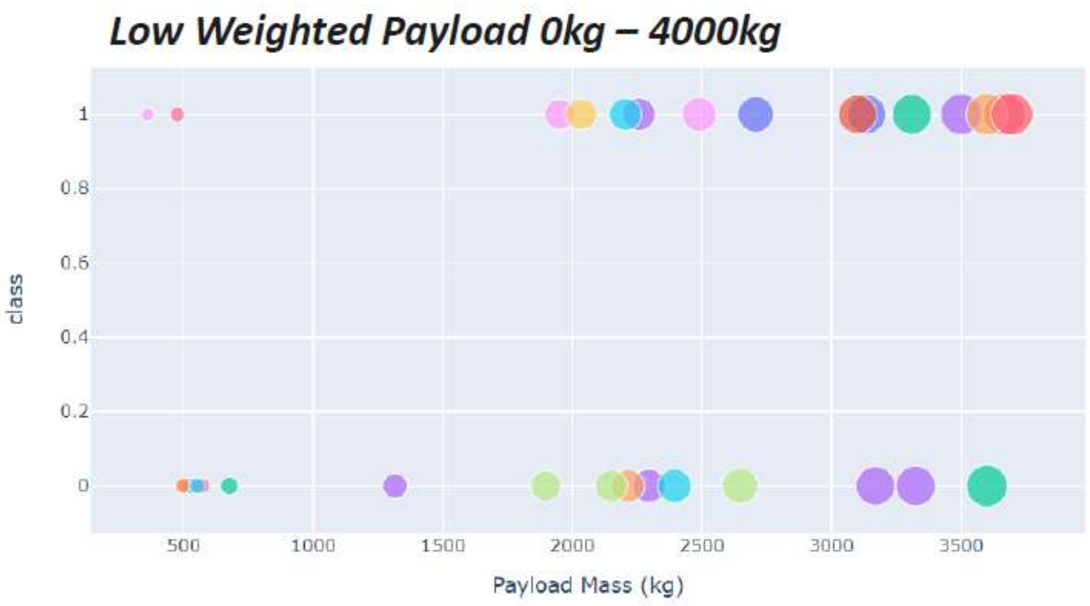
We can see that KSC LC-39A had the most successful launches from all the sites

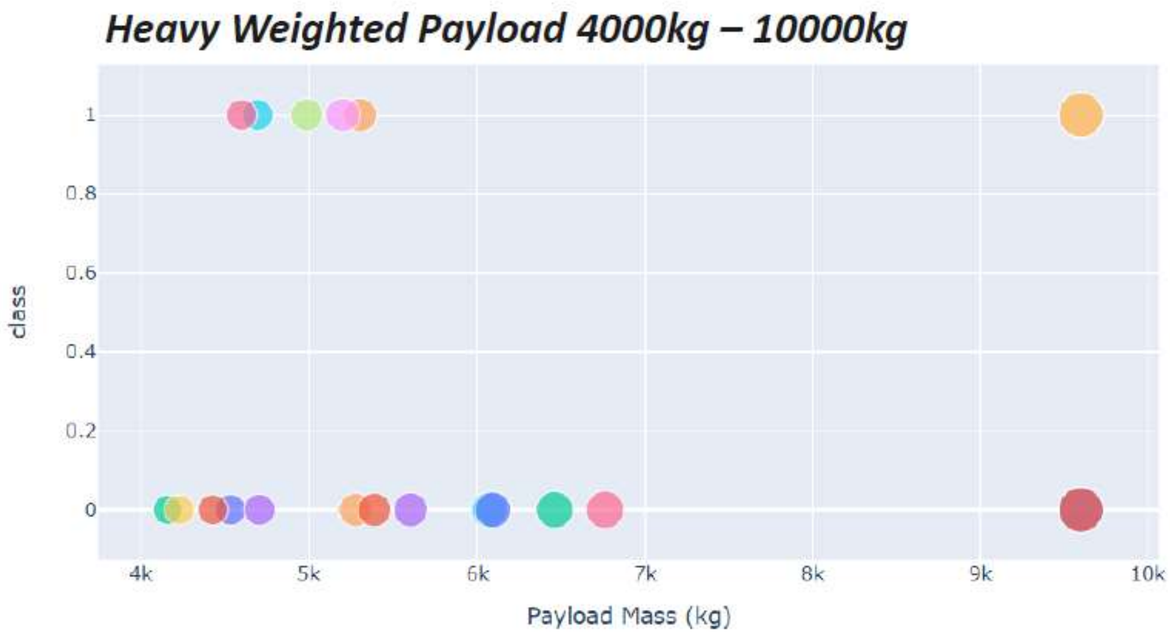
Pie Chart Dashboard – Launch Success Ratio



KSC LC-39A achieved a 76.9% success rate while getting a 23.1% failure rate

Scatter Plot Dashboard – Payload vs Launch Outcome





We can see the success rates for low weighted payloads is higher than the heavy weighted payloads

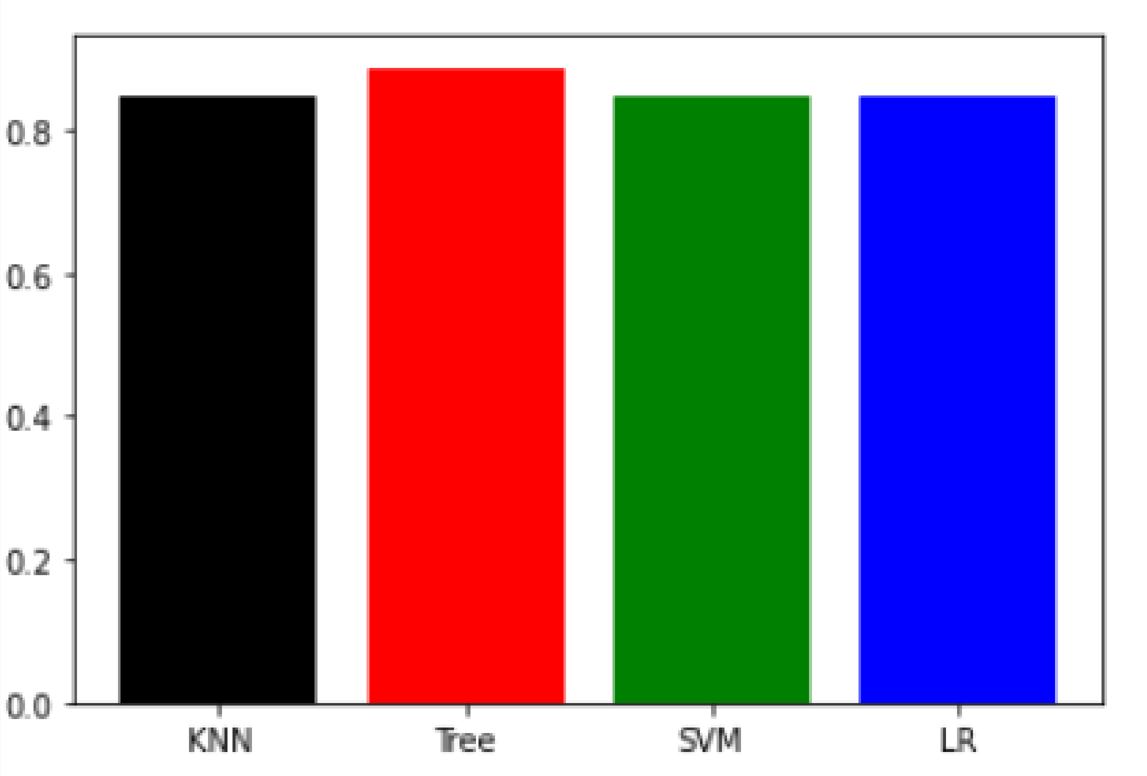
Section 6

Predictive Analysis (Classification)

Classification Accuracy

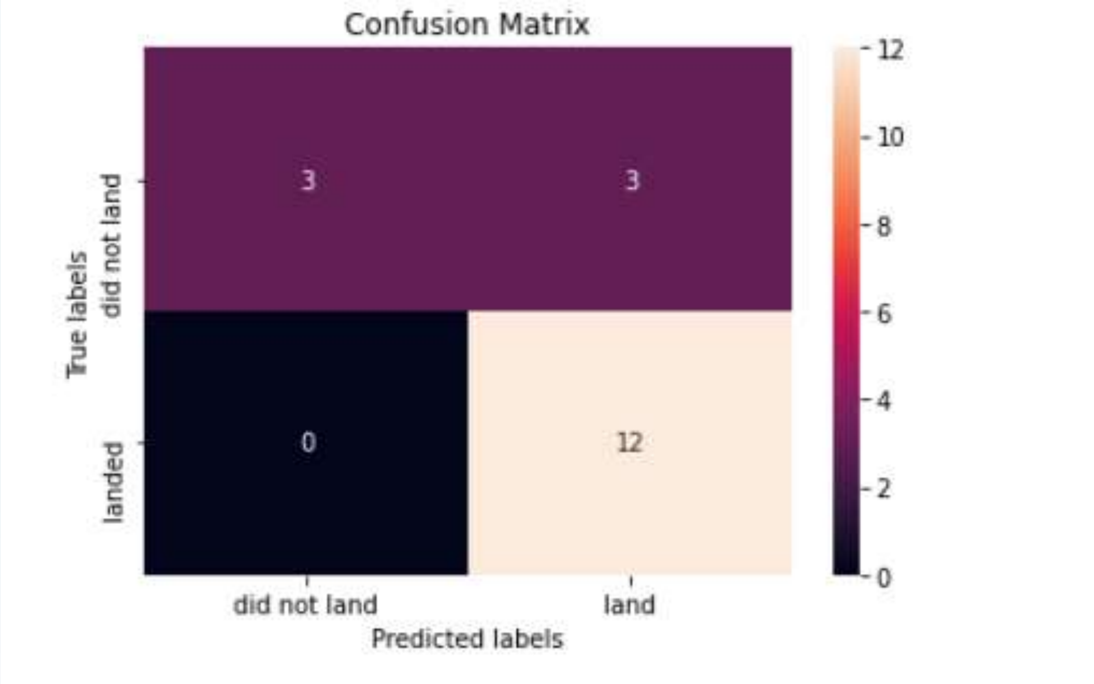
- The bar chart of all classification models showing accuracy

- Highest accuracy lies for tree_cv at 0.8875



Confusion Matrix

- Confusion matrix of tree_cv shown below as being highest accuracy scorer.
- Examining the confusion matrix, we see that Tree can distinguish between the different classes. We see that the major problem is false positives.



Conclusions

- Tree classification was best in Machine learning model
- Low weighted payload perform better to an extent but if higher payloads are consistently used, they perform better in the long run
- Success rate depends on time length – the longer such companies persists the higher chances of success they end up getting
- Orbit GEO,HEO,SSO,ES-L1 has the best Success Rate

Thank you!