IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Innocent Enyekwe
20th April, 2022

Outline

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

Executive Summary

Summary of methodologies

- Data collection through API
- Data collection with Web Scrapping
- Data wrangling
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models

Summary of all results

- Exploratory analysis result
- Interactive visuals
- Predictive analysis result

Introduction

This project aims at gathering data on Falcon 9 rocket launches from SpaceX to try and get some insight on their success rate in landing the rocket's first stage.

Information gathered here, such as determining if the first stage will land and factors that contributes to successful first stage landing can be used by order rocket competitors to bid against SpaceX.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - The data for this project was gathered from SpaceX REST API and Wikipedia using a get request (requests.get()) and BeautifulSoup respectively.
- Perform data wrangling
  - .json_normalize() function was used to normalize the structured .json data into a flat table, followed by sampling of the data and dealing with the Nulls.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Various classification models were built, and tuned to get best parameters for them.

Data Collection

- The data collection flow is as follows

1. Get Request & WebScrapping
2. Normalize the structured JSON data into a Dataframe
3. Sort the Dataframe
4. **Clean the Dataframe** leaving only Falcon 9 data
5. Clean the Dataframe of missing values

Data Collection – SpaceX API

```
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
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 [28]: # Use json_normalize meethod to convert the json result into a dataframe
         # response1 = requests.get(static_json_url).json()
         # data = pd.json_normalize(response1)
         data = pd.json_normalize(requests.get(static_json_url).json())
         data
```

- Data was collected from SpaceX REST API using requests.get() and turned into Pandas dataframe using .json_normalize().

- [datascitest/Data_Collection_API.ipynb at master · enyekwe/datascitest (github.com)]()

Data Collection - Scraping

```
In [20]: # 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")
```

```
In [15]: # use requests.get() method with the provided static_url
         # assign the response to a object
         response = requests.get(static_url).text
         # response
```

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

```
In [16]: # Use BeautifulSoup() to create a BeautifulSoup object from a response text content
         soup = BeautifulSoup(response, "html.parser")
```

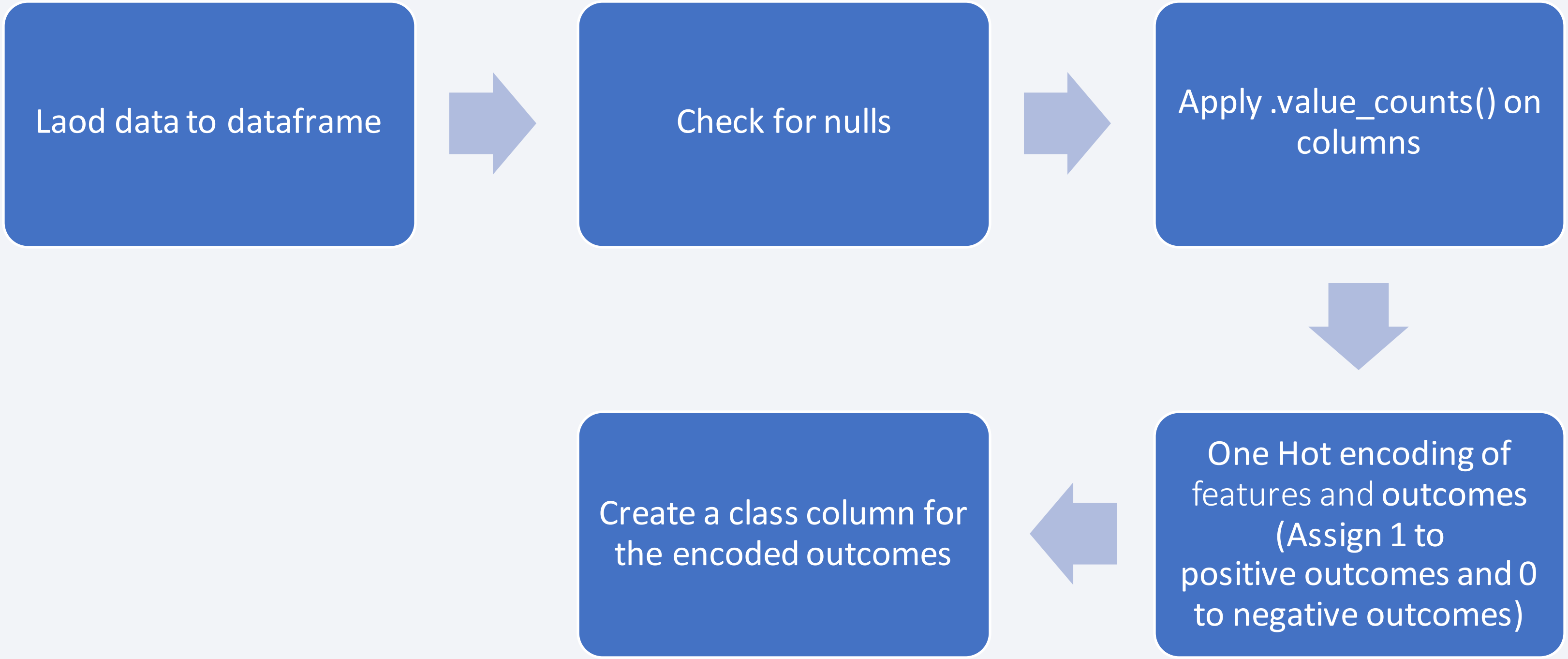
```
In [27]: column_names = []

         # 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
         for th in first_launch_table.find_all("th"):
             name = extract_column_from_header(th)
             if name is not None and len(name) > 0:
                 column_names.append(name)
```

- Using request.get(), Wikipedia was accessed. Then using the BeautifulSoup object, the table rows and column were extracted.

- datascitest/Data_Collection_Web_Scraping.ipynb at master · enyekwe/datascitest (github.com)

Data Wrangling



datascitest/Exploratory_Data_Analysis.ipynb at master · enyekwe/datascitest (github.com)

EDA with Data Visualization

- To Explore relationships between features, the following type of charts were utilized.
  - Scatter plot
    - The following feature pairs were visualized
      - FlightNumber vs PayloadMass
      - FlightNumber vs LaunchSite
      - Launch Sites vs PayloadMass
      - FlightNumber vs Orbit type
      - Payload vs Orbit type
  - Bar chart
    - Success rate of each orbit
  - Line plot
    - Year vs average success rate

datascitest/Exploratory_Data_Analysis_Pandas&Matplotlib.ipynb at master · enyekwe/datascitest (github.com)

EDA with SQL

- SQL queries were performed to acquire the following information
  - Names of unique launch sites
  - Records where launch sites begin with "CCA"
  - Total payload mass carried by boosters launched by NASA (CRS)
  - Average payload mass carried by booster version F9 v1.1
  - Total number of successful and failed mission outcomes
  - Booster versions that have carried the maximum payload

datascitest/Exploratory_Data_Analysis_SQL.ipynb at master · enyekwe/datascitest (github.com)

Build an Interactive Map with Folium

Map objects such as markers, circles were created to identify the various launch sites on the map using their latitude and longitude coordinates.

Using markers and line objects, nearest coastlines, highways, railways and cities to each launch site were identified and the distance shown.

Markings were also created to identify successful and failed launches on each launch sites.

datascitest/Interactive_Visual_Analytics_Folium.ipynb at master · enyekwe/datascitest (github.com)

Build a Dashboard with Plotly Dash

- An interactive dashboard was created to help with the following visualization
  - A pie chart that takes a users input choice of launch site and visualizes its success rate
  - A scatter plot that takes the users input choice of launch site and a range of pay load mass (Through a range slider) and plots the pay load mass range vs the various outcomes for the selected site and range.

- datascitest/spacex_dash_app at master · enyekwe/datascitest (github.com)

Predictive Analysis (Classification)

Load data

Create the target array (Y), and features array (X).

Standardize the data in X

Split X and Y to train and test data in the ratio of 8:2

Create the predictive models with the following objects

- Logistic Regression
- Support Vector Machine
- Decision Tree
- K nearest neighbour

The best parameters for theses models were determined using the GridSearchCV object

The performance of the best parameter was visualized with a confusion matrix.

[datascitest/Spacex_Falcon9_Landing_Machine_Learning_Prediction.ipynb at master · enyekwe/datascitest (github.com)]

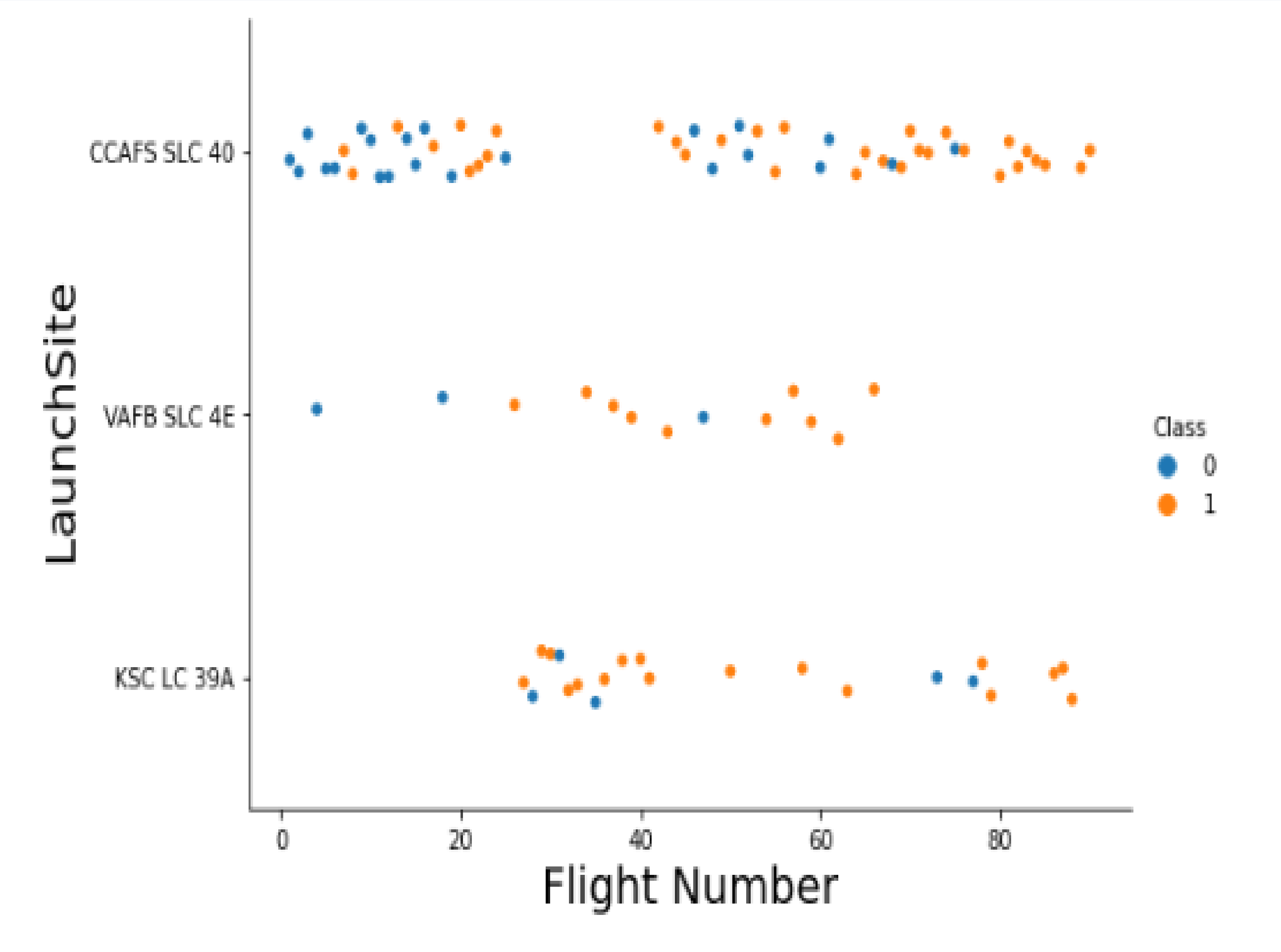
Results

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

Section 2

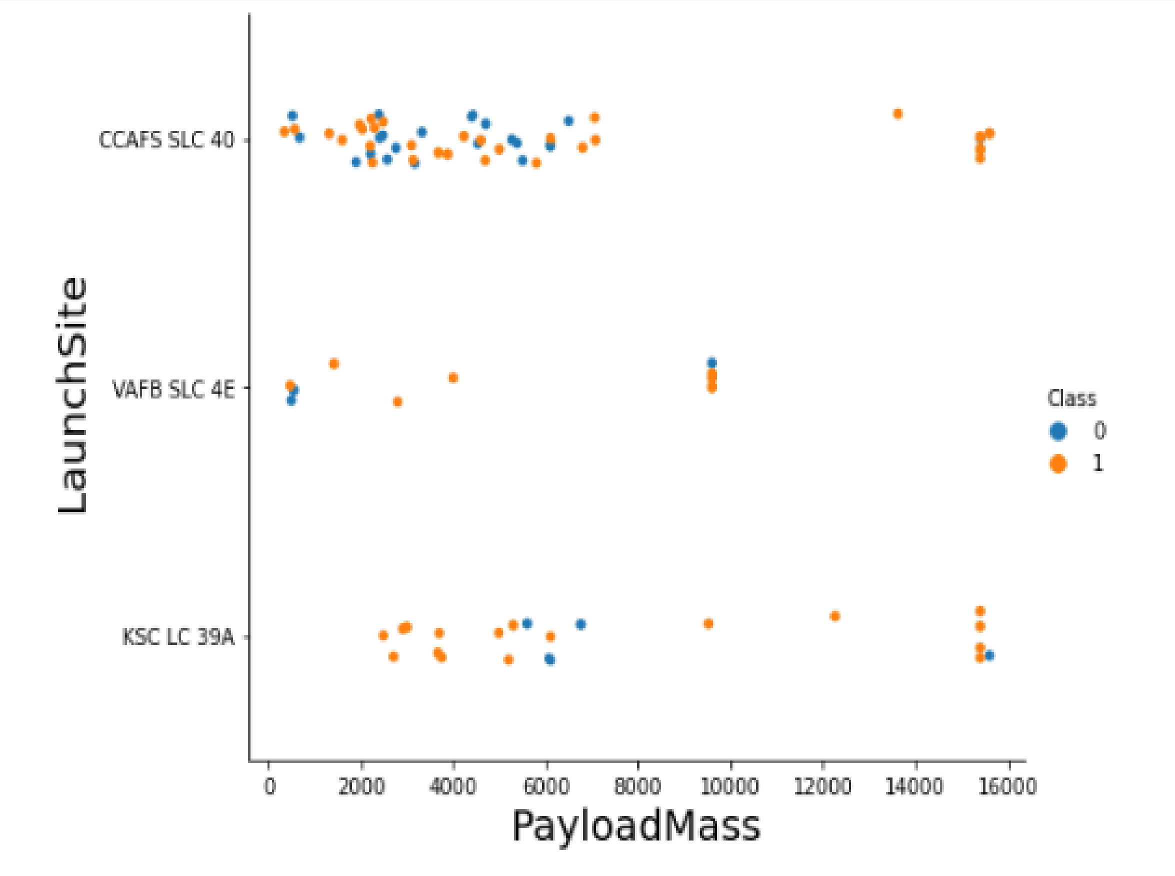
Insights drawn from EDA

Flight Number vs. Launch Site



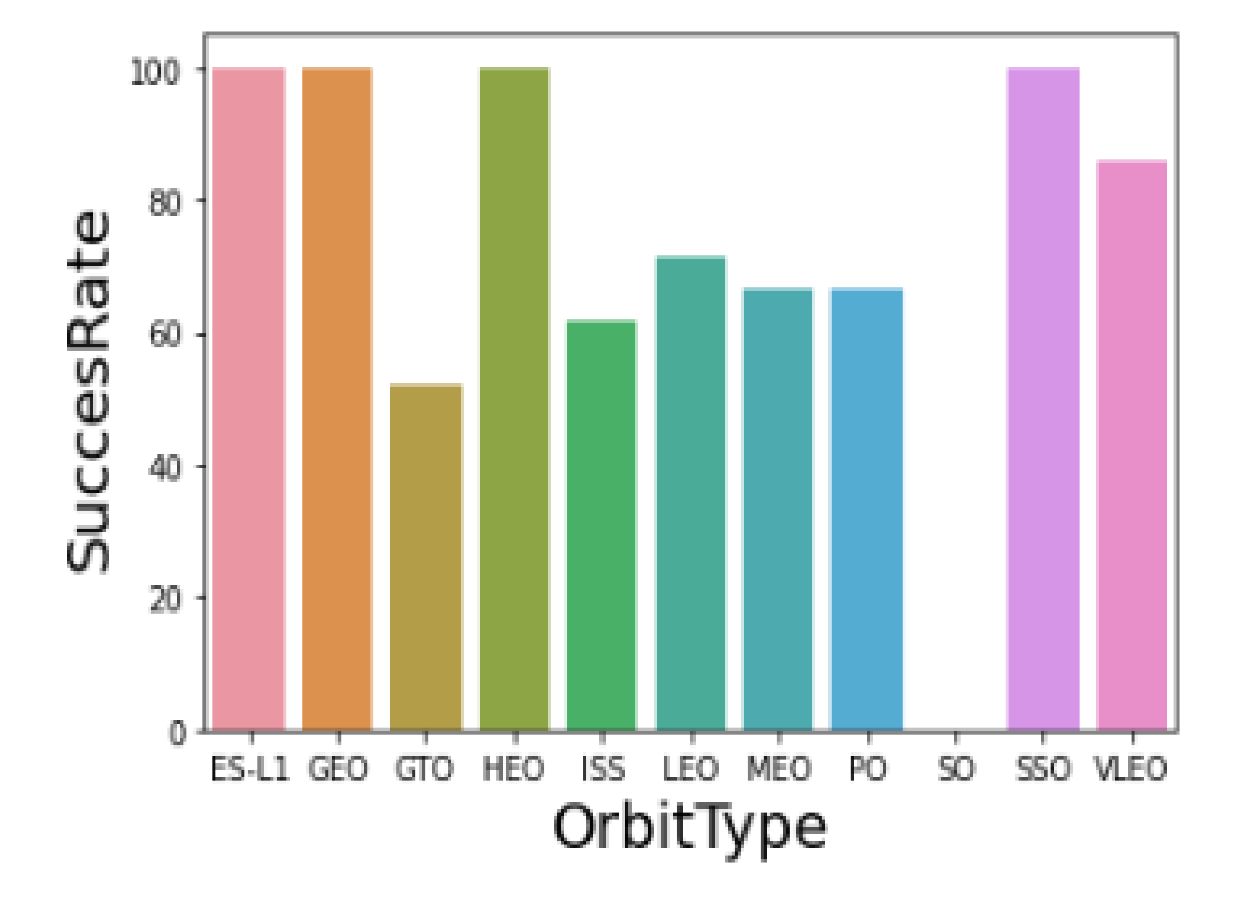
- There is no conclusive correlation between the flight number and Launch site.
- Comparing the three launch sites, VAFB SLC 4E has the highest success rate.
- Most of the first twenty launches where done from CCAFS SLC 40. Which also has the overall highest number of launch.

Payload vs. Launch Site



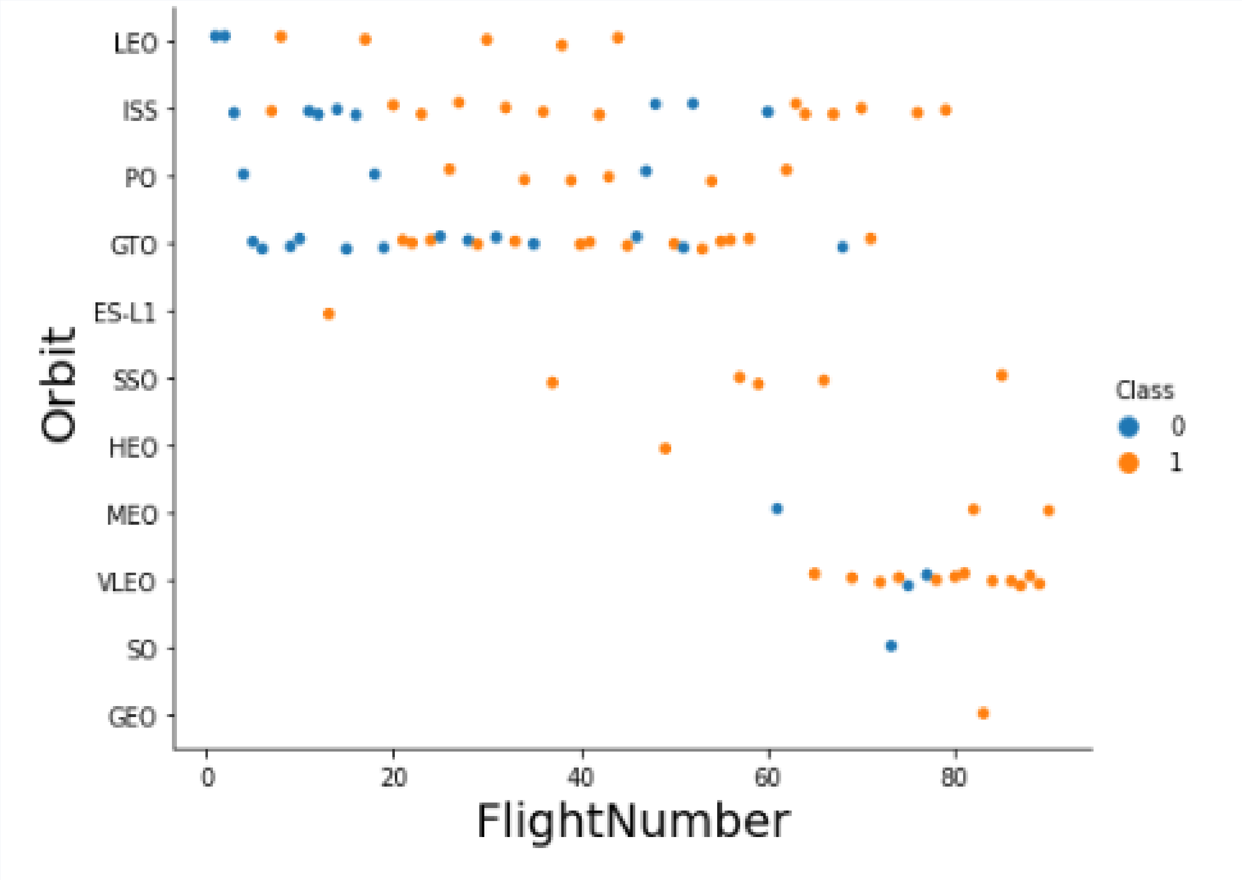
- Launches with payload of over 7500KG recorded a high success rate.
- For lower payloads below 7500KG, a high succeessful outcome can be seen for launch site KSC LC 39A

Success Rate vs. Orbit Type



- This shows the success rate of launches to various orbits. To justify this, we need to also put into consideration the frequency of launch to these orbits.

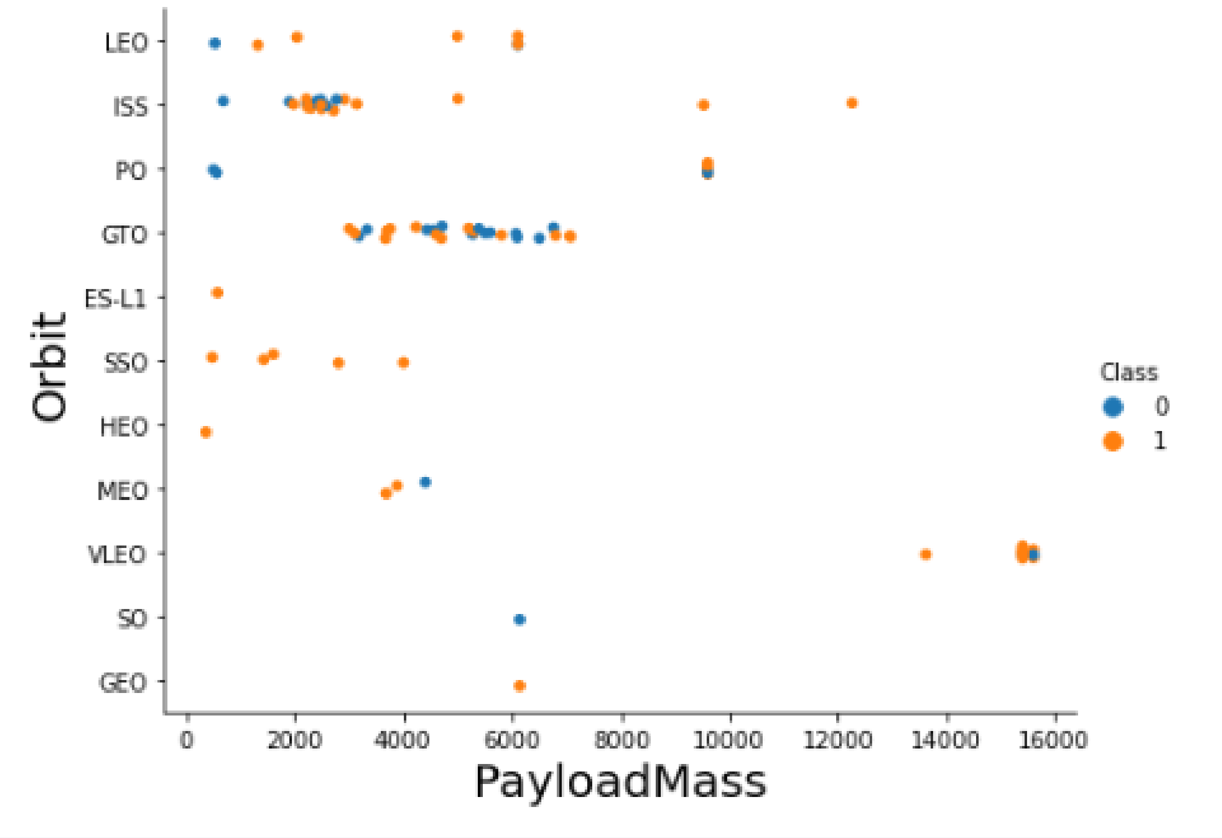
Flight Number vs. Orbit Type



- We can clearly see that the early launches (0 – 50) were done mostly to these orbits, LEO, ISS, PO, and GTO
- SSO has a total of 5 launches and zero record of failure
- LEO can be said to have a linear correlation to the flight number why the rest are not clear

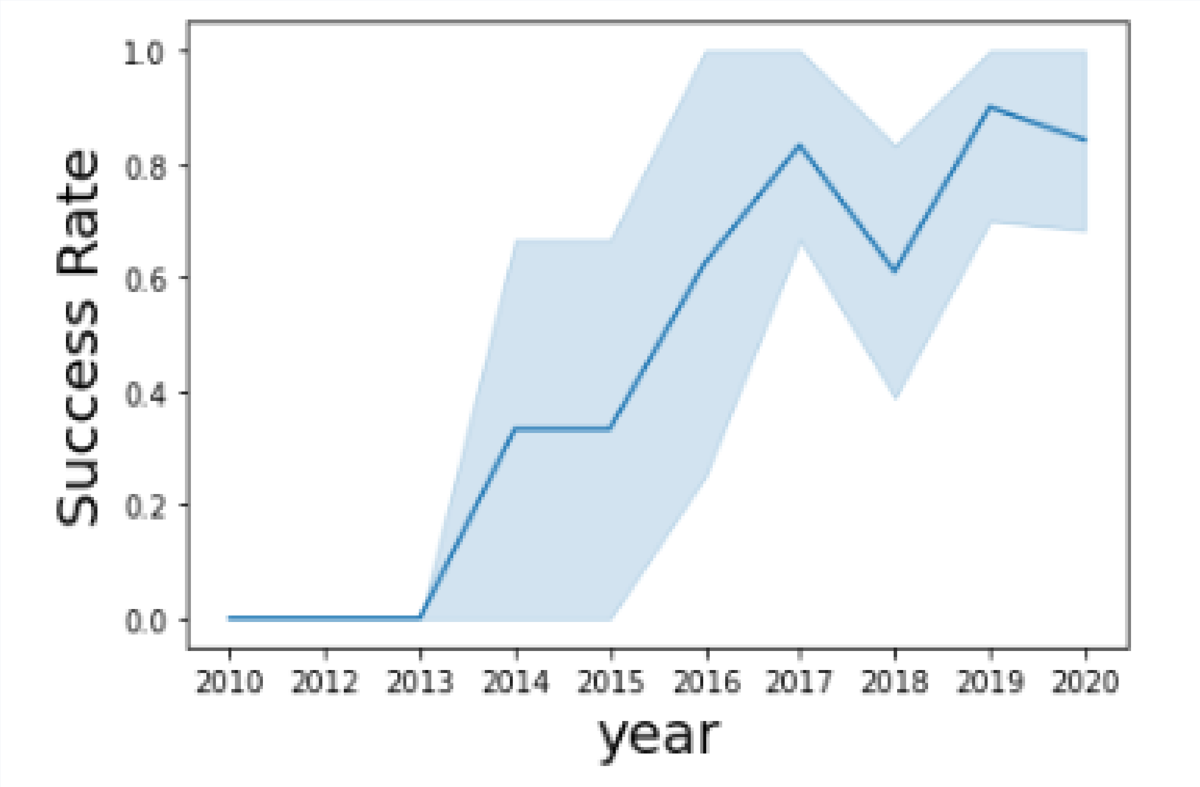
Payload vs. Orbit Type

- This plot tries to visualize any relationship between Payload and orbit type
- Most of the payloads below 7500KG were launched to these orbits LEO, ISS, PO, and GTO orbits



Launch Success Yearly Trend

- A line plot to understand the relationship between success rate and year
- A clear correlation can be seen between success rate and year, with a sudden rise in success rate from 2013



All Launch Site Names

- The launch sites are as shown below
- The database was querried using DISTINCT(). This goes through the launch site column and outputs all unique values in it.

```
In [22]: %sql select distinct(launch_site) from SPACEXTBL

   ibm_db_sa://ccl28209:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0
 * ibm_db_sa://mzm98143:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0
Done.

Out[22]:
```

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

Launch Site Names Begin with 'CCA'

- We select the SACEXTBL and use the where clause and like() to output rows that where item in launch_site column starts with 'CCA"

```
In [29]: %sql select * from SPACEXTBL where launch_site like ('CCA%') limit 5

   ibm_db_sa://ccl28209:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
 * ibm_db_sa://mzm98143:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:31321/bludb
Done.
```

Out[29]:

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

- The total payload carried by boosters from NASA
- With sum(), the sum of the payload mass for rows where the customer column is 'NASA (CRS)' was calculated.

```
In [37]: %sql select sum(payload_mass__kg_) from SPACEXTBL where customer = 'NASA (CRS)'

   ibm_db_sa://ccl28209:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databa
 * ibm_db_sa://mzm98143:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databa
Done.

Out[37]:
```

1
45596

Average Payload Mass by F9 v1.1

- The average payload mass carried by booster version F9 v1.1
- With avg(), the average payload mass for rows where the booster version is 'F9 v1.1' was calculated.

```
In [41]: %sql select avg(payload_mass__kg_) from SPACEXTBL where booster_version = 'F9 v1.1'

   ibm_db_sa://ccl28209:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.da
 * ibm_db_sa://mzm98143:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.da
Done.
```

Out[41]:

1
2928

First Successful Ground Landing Date

- The first successful landing outcome on ground pad
- Applying min() to the date column where landing outcome is success (ground pad) gives the first
- uccessful landing outcome on ground pad

```
In [44]: %sql select min(date) from SPACEXTBL where landing__outcome = 'Success (ground pad)'

   ibm_db_sa://ccl28209:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.da
 * ibm_db_sa://mzm98143:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.da
Done.

Out[44]:
```

1
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- Boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000
- We select the booster version applying the where clause for landing_outcome = 'success (drone ship)' and payload_mass_kg_ between 4000 and 6000

In [46]:
```
%sql select booster_version, payload_mass__kg_ from SPACEXTBL where landing__outcome = 'Success (drone ship)' and payload_mass__kg_ between 4001 and 6000

# >= 4000 and <=6000
```

```
   ibm_db_sa://ccl28209:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
 * ibm_db_sa://mzm98143:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:31321/bludb
Done.
```

Out[46]:

booster_version	payload_mass__kg_
F9 FT B1022	4696
F9 FT B1026	4600
F9 FT B1021.2	5300
F9 FT B1031.2	5200

Total Number of Successful and Failure Mission Outcomes

- COUNT() was applied to the mission_outcome column to get the total

```
In [47]: %sql select count(mission_outcome) from SPACEXTBL

   ibm_db_sa://ccl28209:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1
 * ibm_db_sa://mzm98143:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1
Done.
```

Out[47]:

1
101

Boosters Carried Maximum Payload

- Booster which have carried the maximum payload mass
- We select booster versions whose payload mass = maximum payload mass

```
In [49]: %sql select booster_version, payload_mass__kg_ from SPACEXTBL where payload_mass__kg_ = (select MAX(payload_mass__kg_) from SPAC
EXTBL)

   ibm_db_sa://ccl28209:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
 * ibm_db_sa://mzm98143:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:31321/bludb
Done.
```

Out[49]:

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

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

In [55]:
```
%%sql
select Landing__Outcome, booster_version, launch_site from SPACEXTBL
    where landing__outcome = ('Failure (drone ship)')
    and year(date) = 2015
```

```
   ibm_db_sa://cc128209:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0l
 * ibm_db_sa://mzm98143:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0l
Done.
```

Out[55]:

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

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

- 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

In [94]:
```
%%sql
select Landing__Outcome, count(Landing__outcome) as frequency from SPACEXTBL
where date between '2010-06-04' and '2017-03-20'
group by Landing__outcome
order by frequency desc
```

```
   ibm_db_sa://ccl28209:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.datal
 * ibm_db_sa://mzm98143:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.datal
Done.
```

Out[94]:

landing__outcome	frequency
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

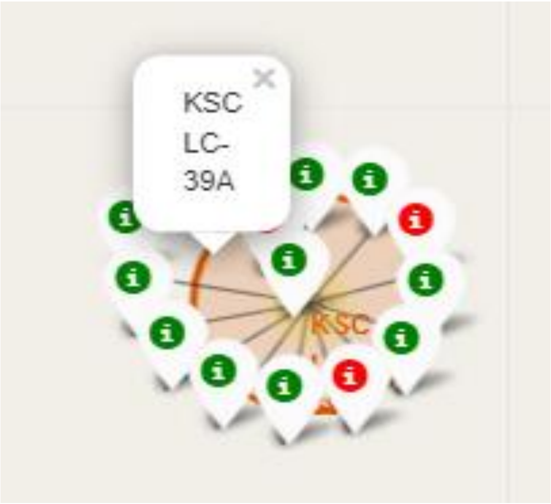
Launch Sites Proximities Analysis

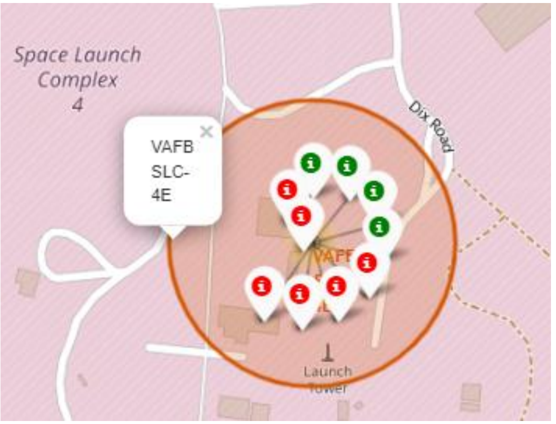
Launch Sites Map View

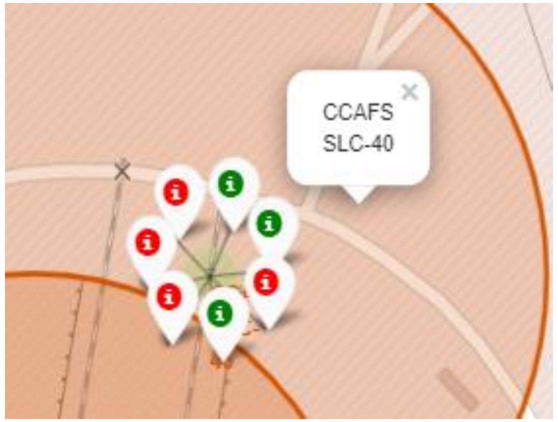
- Most of the launches were held near NASA headquater in Florida
- All launch sites has close proximity to coastal lines

Launch Outcomes on Various sites

- Green marker was used to identify successful launches and the red marker shows failed ones.
- KSC LC-39A has the higest success rate

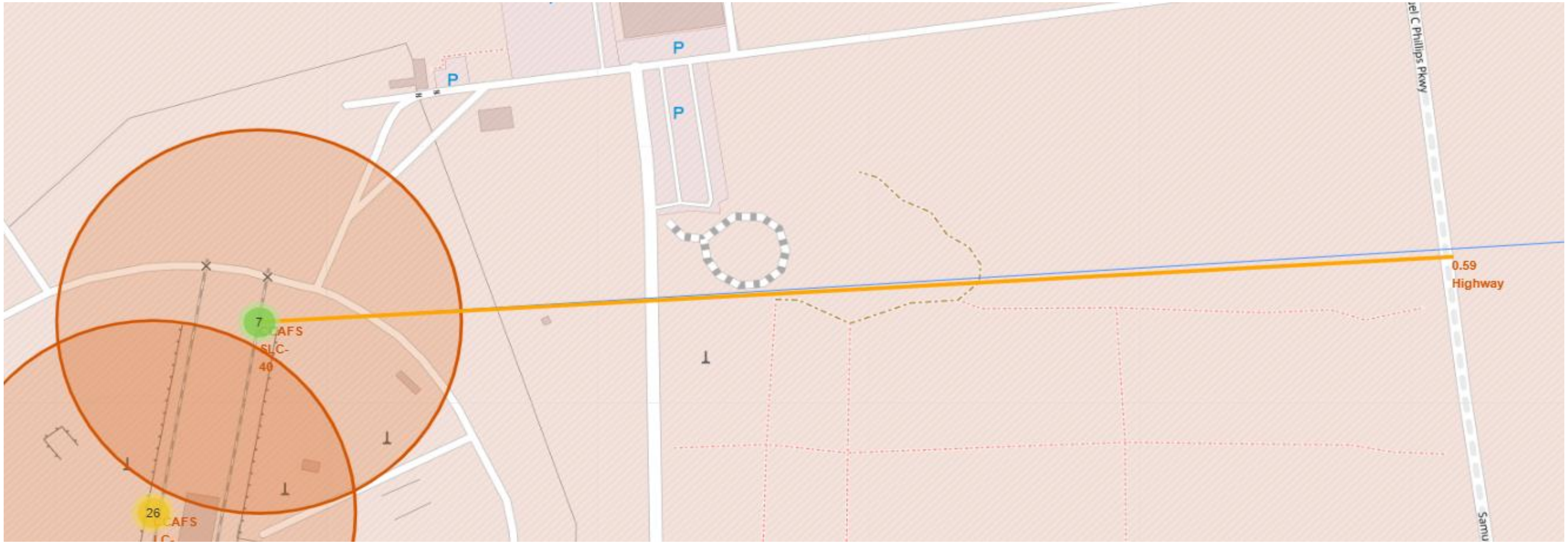






CCAFS SLC-40 distance from the highway

- It can be easily seen that this launch site is at close proximity to the highway (0.59KM). This makes it easy to handle logistics.

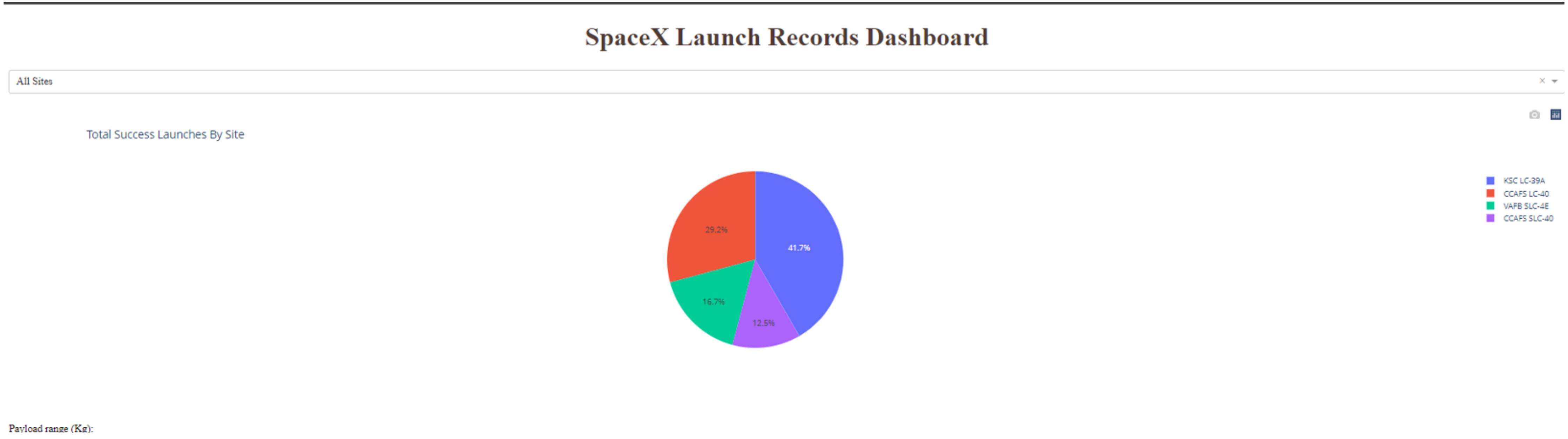


Section 4

Build a Dashboard with Plotly Dash

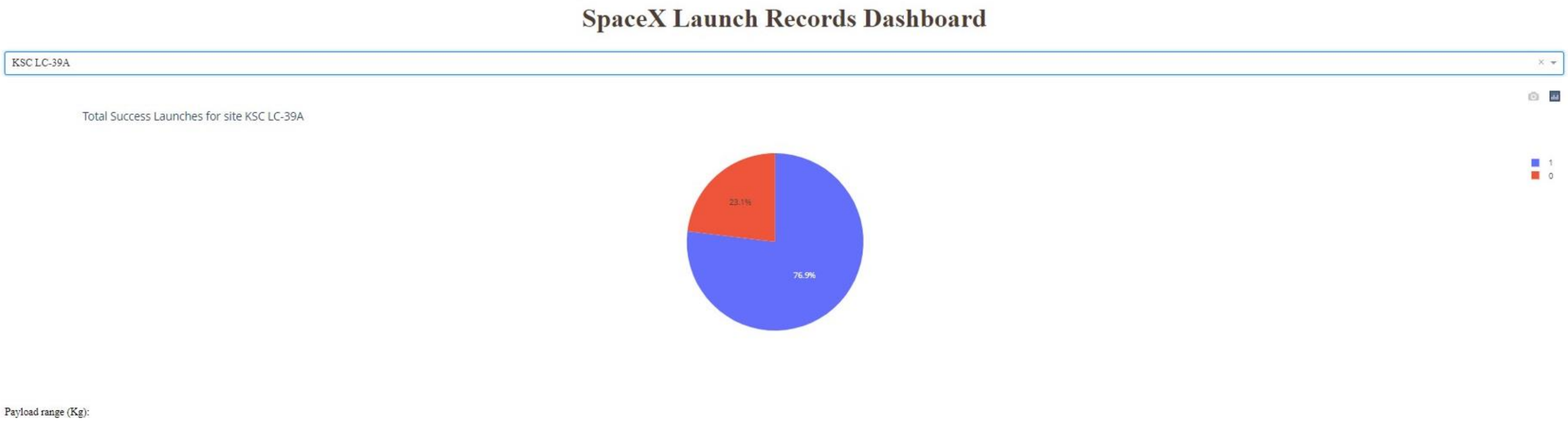
Success Rate for all Sites

- The pie chart clearly shows the success rate at each launch site compared to others, with KSC LC-39A having a 41.7% success rate while the lowest is CCAFS SLC-40 with 12.5% success rate.



KSC LC-39A Launch Outcomes

- KSC LC-39A records a success rate of 76.9%.



Payload vs Launch Outcome

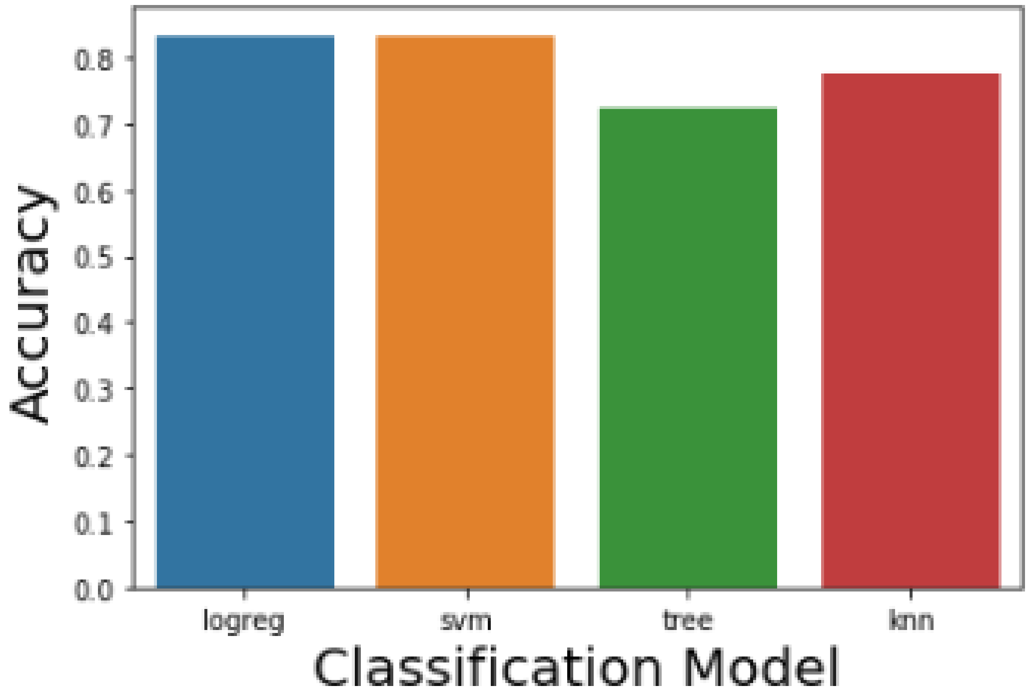
- It can be seen that lighter Payloads has low success rate
- Among other booster versions, FT records a higher success rate

Section 5

Predictive Analysis (Classification)

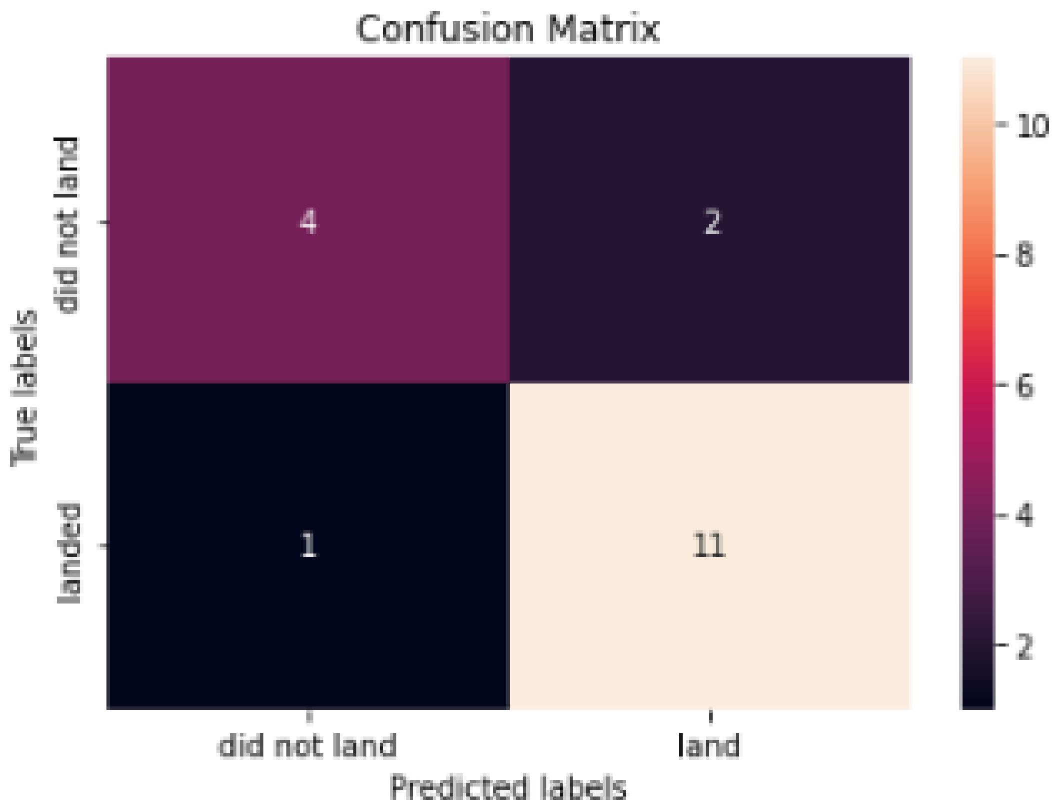
Classification Accuracy

- My logistic regression and Support vector machine recorded the highest accuracy of 83.34%



Confusion Matrix

- From the confusion matrix of my best model, 15 out of the 18 test data were classified correctly.
- Likewise, the model also produced 2 false positive and 1 false negative.



Conclusions

- Payload plays an important role on the success of a launch. Where payloads of above 7500KG can improve the chance of having a successful outcome
- To increase the chance of a successful outcome for lower payloads, it is recommednded to launch from KSC LC 39A
- LEO, SSO and VLEO appears to have improving success rate as the number of flight increases
- With the predictive model designed, we can predict if we can successfully land the first stage for various variables such as launch site, pay load mass, orbit, and booster version

Thank you!