IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Daohua Huang
12/07/2021

Outline

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

Executive Summary

- **Summary of methodologies**
  - Data collection
  - Perform 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 data analysis results
  - Interactive analytics demo in screenshots
  - Predictive analysis results

Introduction

Project background and context

- In this project, we will predict whether the Falcon 9 first stage will land successfully or not. 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 to be solved

- What influences if the rocket will land successfully?
- The effect each relationship with certain rocket variables will impact in determining the success rate of a successful landing
- What conditions does SpaceX have to achieve to get the best results and ensure the best rocket success landing rate

Section 1

Methodology

Methodology

Executive Summary

Data collection methodology

- Request to the SpaceX API and clean the requested data
- Extract a Falcon 9 launch records HTML table from Wikipedia and parse the table and convert it into a Pandas data frame

Perform data wrangling

- Exploratory Data Analysis and determine Training Labels

Perform exploratory data analysis (EDA) using visualization and SQL

Perform interactive visual analytics using Folium and Plotly Dash

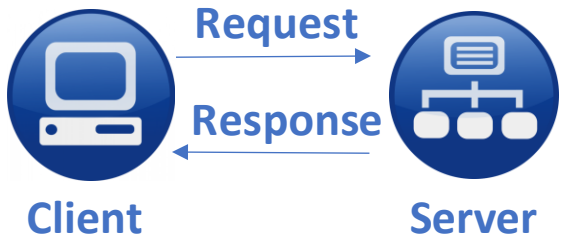
Perform predictive analysis using classification models

- How to build, tune, evaluate classification models

Data Collection

- **The SpaceX launch data**

  The data is gathered from the SpaceX REST API, the process shows below. Then we clean the date including drop some columns, and deal with missing values

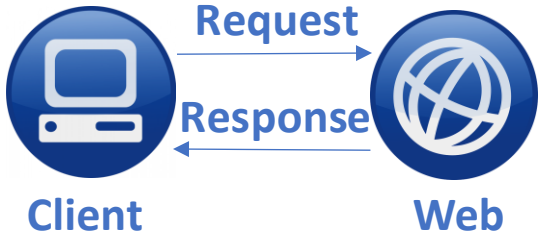


- **The flowchart of whole process**

Continue …

- **Falcon 9 Launch Records Date**

  Web scraping to collect Falcon 9 historical launch records with BeautifulSoup from a Wikipedia page titled List of Falcon 9 and Falcon Heavy launches.



- **The flowchart of whole process**

Data Collection – SpaceX API

[SpaceX API Notebook]

1. Call API and Get Response

```
spacex_url="https://api.spacexdata.com/v4/launches/past"
data=pd.json_normalize(response.json())
```

2. Decode content as Json file and turn it into PD

```
data=pd.json_normalize(response.json())
```

3. Apply functions to convert value to readable

```
getBoosterVersion(data):
    for x in data['rocket']:
        response = requests.get("https:..."+str(x)).json()
        BoosterVersion.append(response['name'])
getBoosterVersion(data)
```

4. Create a dictionary

```
launch_dict = {'FlightNumber': list(data['flight_number']),...}
```

5. Create a Pandas data frame from the dictionary

```
launch_pf = pd.DataFrame.from_dict(launch_dict)
```

Data Collection - Scraping

Scraping data from Wiki

1. Get response from URL: List of Falcon 9 and Falcon Heavy launches)

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

2. Create a BeautifulSoup object from the HTML response

```
soup = BeautifulSoup(response.text, 'html.parser')
```

3. Extract all column name from the chosen table

```
first_launch_table = html_tables[2]
column_names = []
temp = first_launch_table.find_all('th')
for x in range(len(temp)):
    try:
        name = extract_column_from_header(temp[x])
        if (name is not None and len(name) > 0):
            column_names.append(name)
    except:
        pass
```

4. Create a dictionary which column is the keys and get the values

```
launch_dict= dict.fromkeys(column_names)
```

5. Create a Dataframe from the dictionary

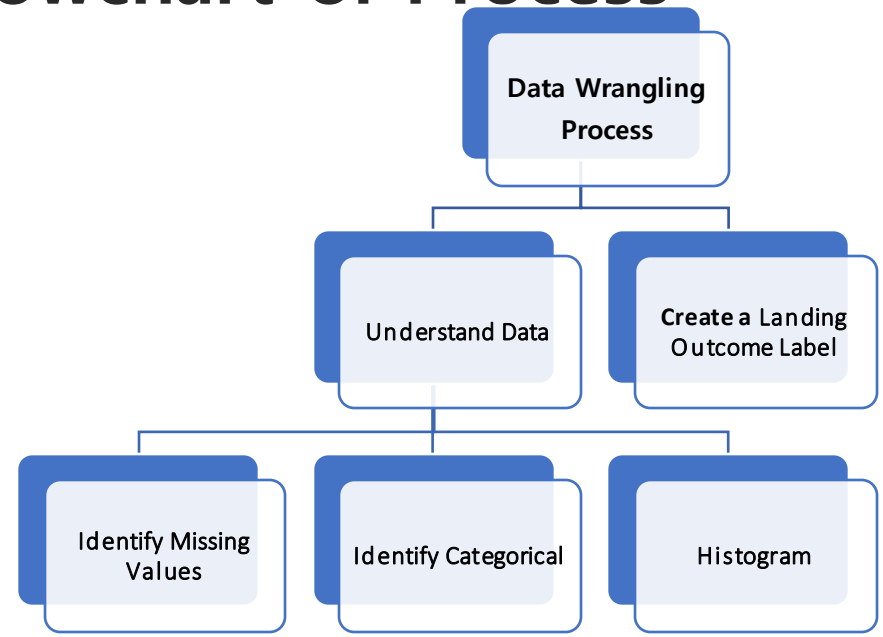
```
df=pd.DataFrame(launch_dict)
```

Data Wrangling

Introduction

Perform some Exploratory Data Analysis (EDA) to find some patterns in this data and determine what would be lable for training supervised models.

Flowchart of Process



Data Wrangling

- **Identify Missing Value**

```
df.isnull().sum()/df.count()*100
```

- **Identify Categorical Variables**

```
df.dtypes
```

- **Histogram**

```
df['LaunchSite'].value_counts()
df['Orbit'].value_counts()
df['Outcome'].value_counts()
```

- **Create a Landing Outcome Label**

```
landing_outcomes=df['Outcome'].value_counts()
bad_outcomes=set(landing_outcomes.keys()[[1,3,5,6,7]])
landing_class=[]
for outcome in df['Outcome']:
    if outcome in bad_outcomes:
        landing_class.append(0)
    else:
        landing_class.append(1)
```

EDA with Data Visualization

SUMMARIZE

- The goal is that predict if the Falcon 9 first stage will land successfully. First, check the relationship of variables by using scatter graphs
  - **FlightNumber vs PayloadMass**
  - **`FlightNumber` vs `LaunchSite`**
  - **PayloadMass vs `LaunchSite`**
  - **FlightNumber vs Orbit**
  - **PayloadMass vs `Orbit`**

  The above five graphs show the correlation of the two variables, that shows how much the one is affected by the other

- The success rate is the main concern in this project, the two graph below show us the detail.
  - **Bar Graph: Average Success Rate vs Orbit**

  A bar diagram is easy to compare the sets of data between different groups at a glance.
  - **Line Graph: Average Success Rate vs Year**

  Line graphs is very clear to show the trends by time or other variable which you chosen. And it also can help us to make predictions.

EDA with Data Visuzlization

EDA with SQL

Performed SQL queries to understand the dataset and gather information

- 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

Build an Interactive Map with Folium

The launch success rate may depend on many factors which you chosen, and it may also depend on the location and proximities of a launch site. Building an interactive map will supply more information to get the answer.

- Create a folium map object with an initial center location
- Add circles which add a highlighted circle with a text label on a specific coordinate
- Add markers which show the icon in the specific coordinate
- Add marker_cluster to add multiple marker in the specific coordinate
- Calculate the distances between a launch site to its proximities
  - Add a MousePositon on the mao to get coordinate on the map
  - Add distance marker to show the value of distance
  - Draw a PloyLine between a launch site to the selected point

Interactive Map with Folium

Build a Dashboard with Plotly Dash

Building a Dashboard is perfect way to perform interactive visual analytics on SpaceX launch data in real-time. This dashboard application contains several aspects below:

- A launch site drop-down input component
- A callback function to render success-pie-chart based on selected site dropdown
- A range slider to select payload
- A callback function to render the success-payload-scatter-chart scatter plot

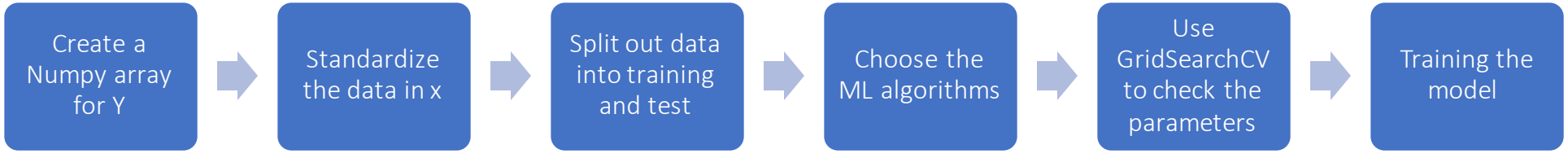
After visual analysis using the dashboard, some insights can be obtained:

- Which site has the largest successful launches
- Which site has the highest launch success rate, et...

Build a Dashboard with Plotly Dash

Predictive Analysis (Classification)

Building the Models



Evaluating Models

- Check accuracy for each model
- Get tuned hyperparameters for each type of algorithms
- Plot Confusion Matrix

Improving Models

- Feature Engineering
- Algorithm tuning

Finding the Best Performing Classification Model

- The model with the best accuracy score

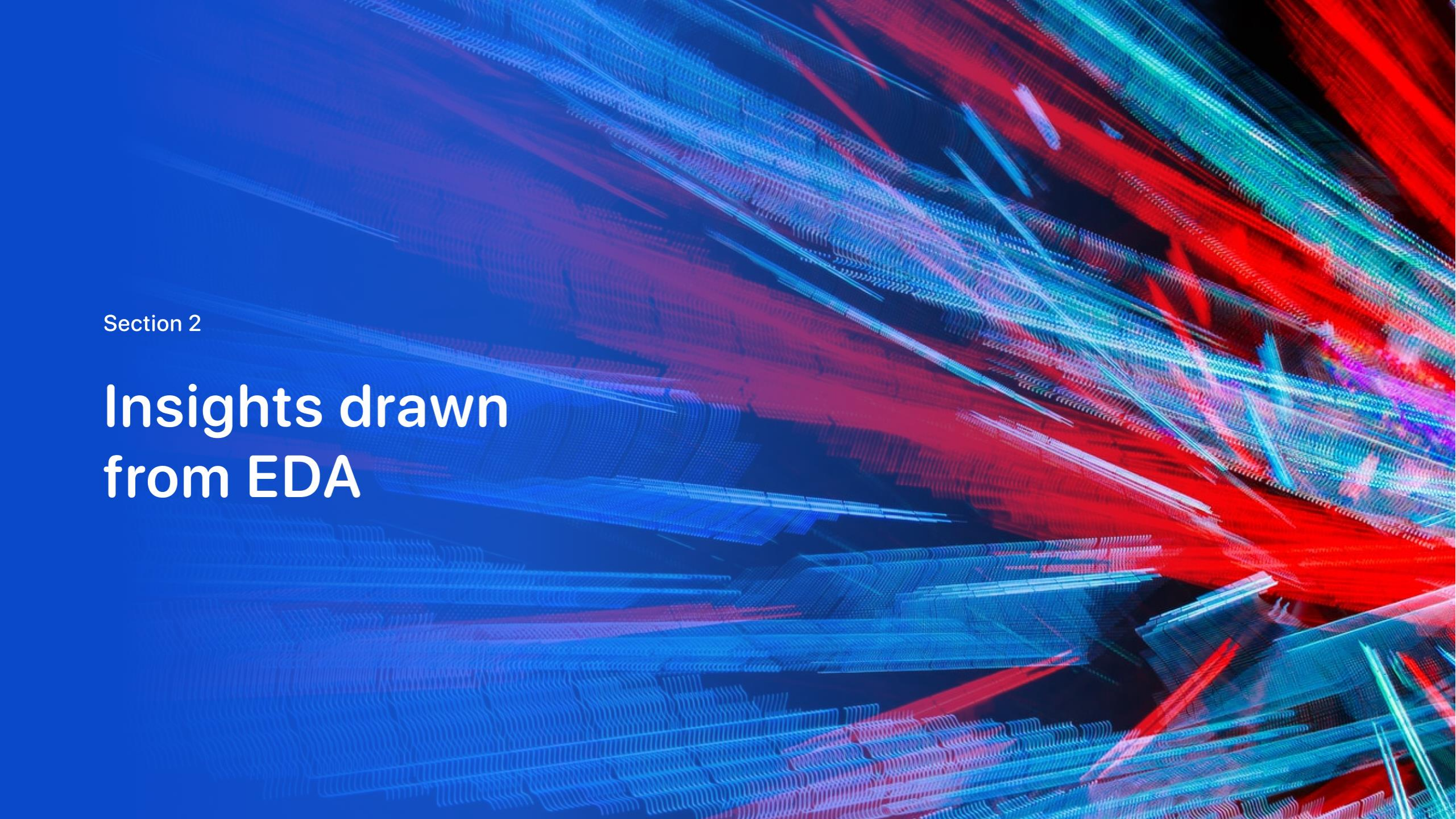
Predictive Analysis (Classification)

Results

Exploratory data analysis results

Interactive analytics demo in screenshots

Predictive analysis results

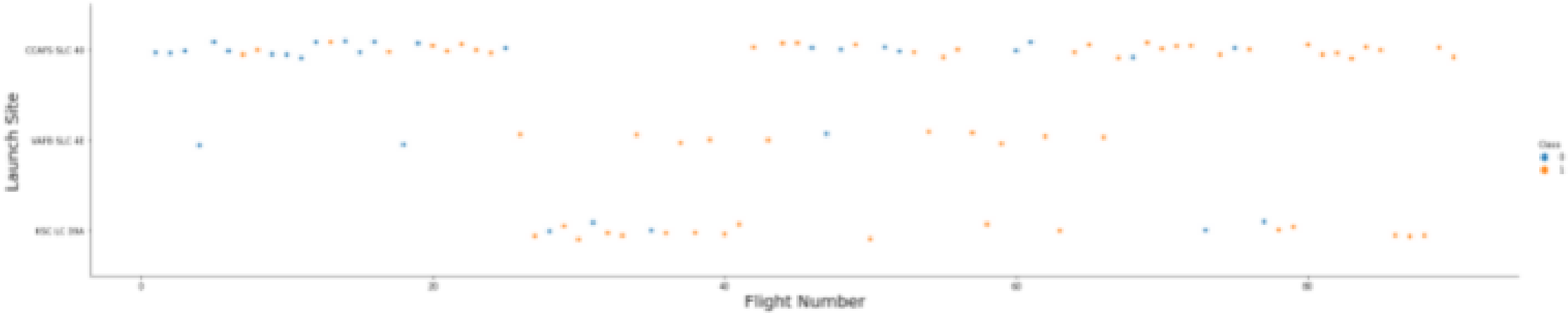


Section 2

Insights drawn from EDA

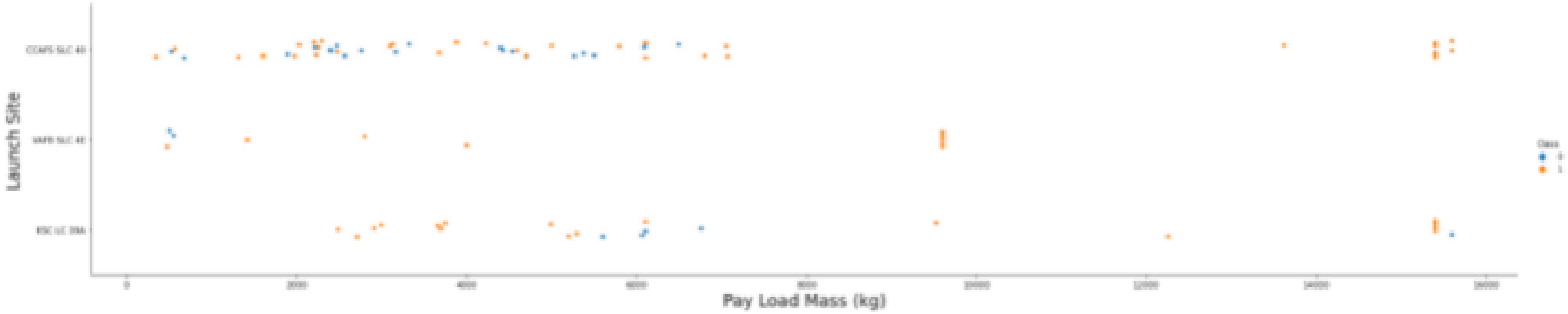
Flight Number vs. Launch Site

From the below diagram, the first stage is more likely to land successfully as the flight number increases. The launch sites seem the location doesn't affect the success rate of first stage returning.



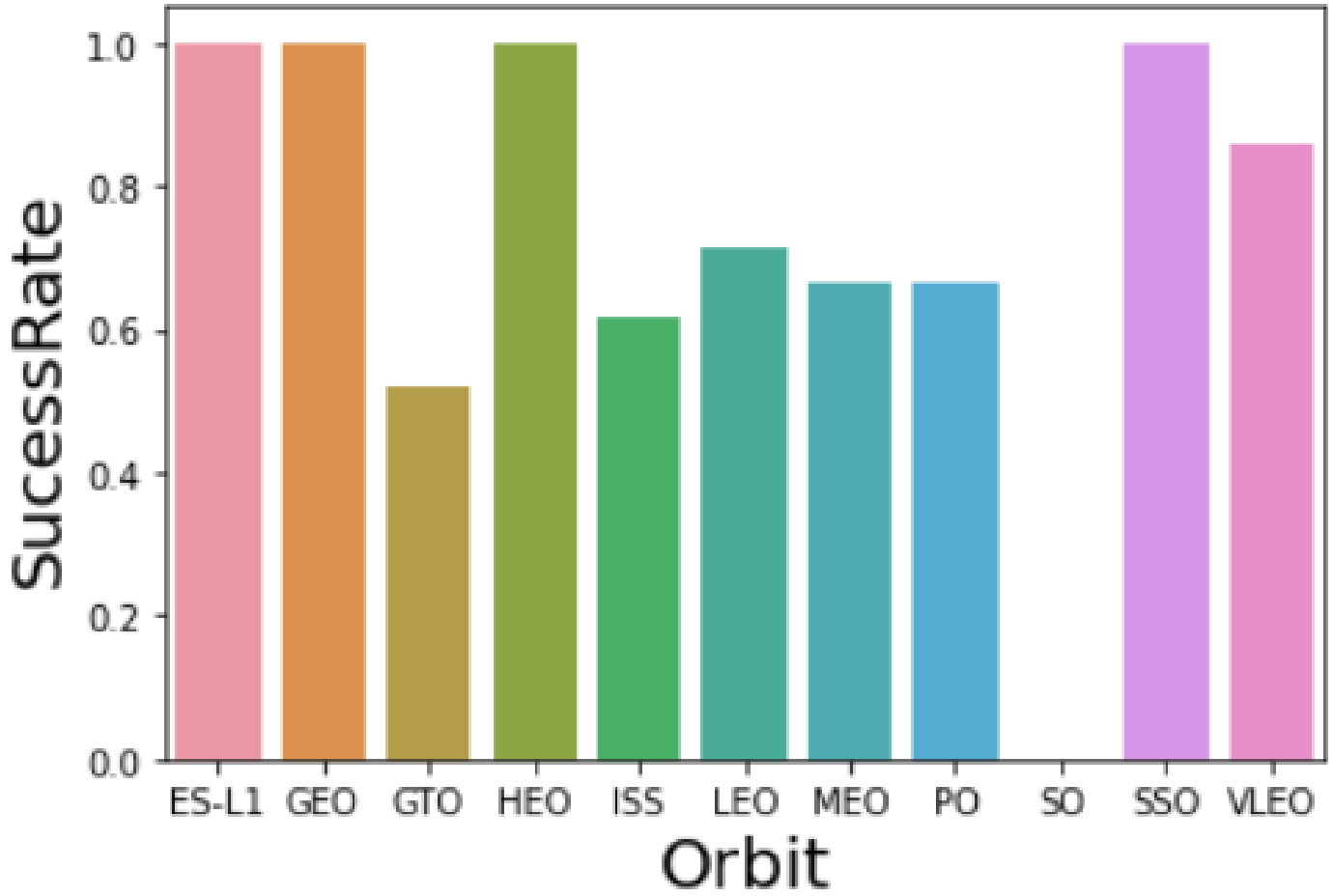
Payload vs. Launch Site

The scatter point chart seems there are no rockets launched for heavypayload mass (greater than 10000) the VAFB-SLC launchsite.



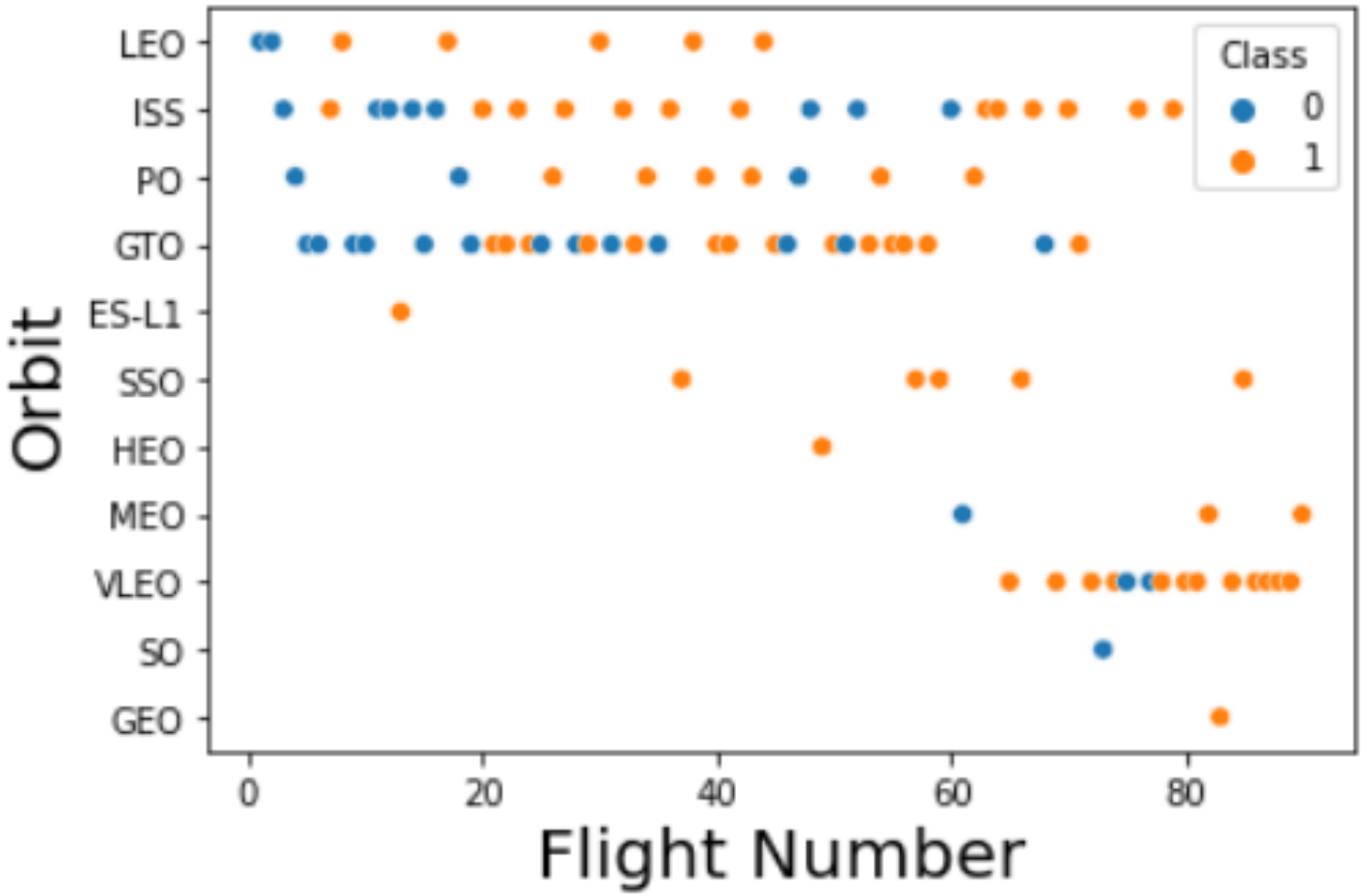
Success Rate vs. Orbit Type

- First stage returning was all successful for the Orbit ES-L1, GEO, HEO, SSO.



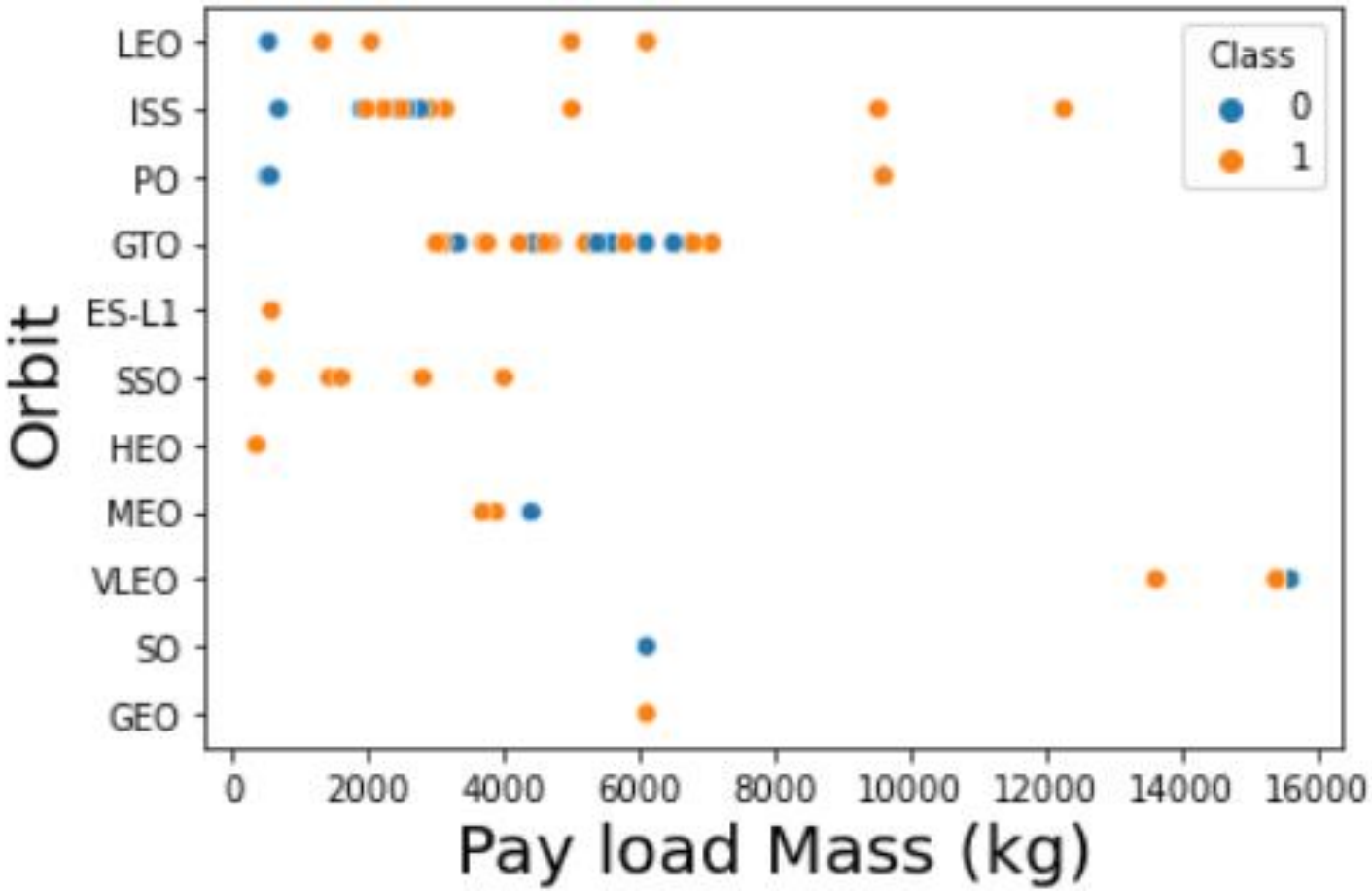
Flight Number vs. Orbit Type

- In the LEO orbit the Success appears related to the number of flights;
- there seems to be no relationship between flight number in GTO orbit, ISS orbit, etc...



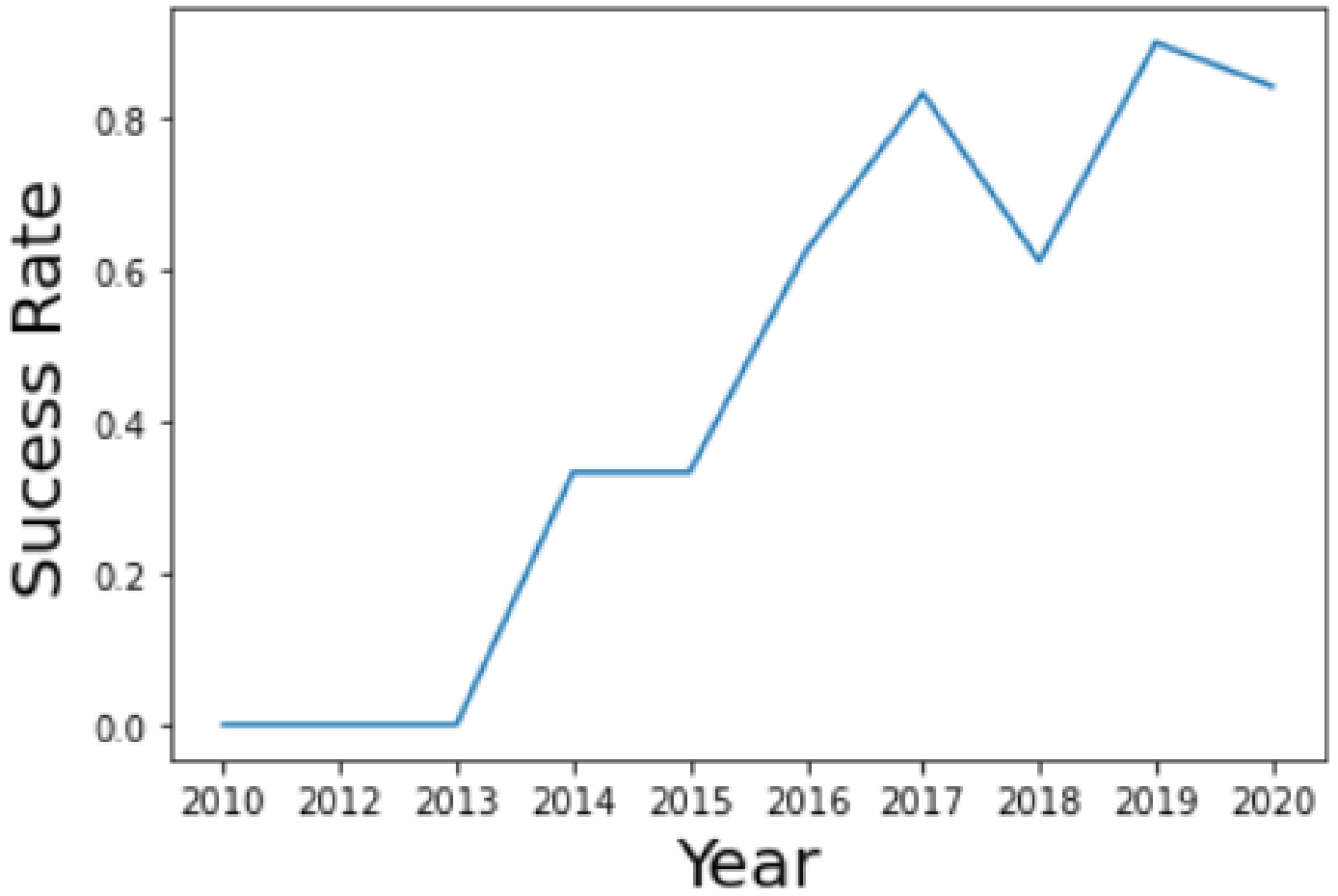
Payload vs. Orbit Type

- With heavy payloads the successful landing rate are more for Polar, LEO, and ISS
- It is not different between both positive landing rate and negative landing in GTO orbit



Launch Success Yearly Trend

- The success rate since 2013 kept increasing till 2017.



All Launch Site Names

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

- The SQL Query Code:

```
%sql select DISTINCT LAUNCH_SITE from SPACEXTBL
```

- Use DISTINCT to delete the duplicate name

Launch Site Names Begin with 'CCA'

- The SQL Query Code:

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

- Use limit or top to control the number of records

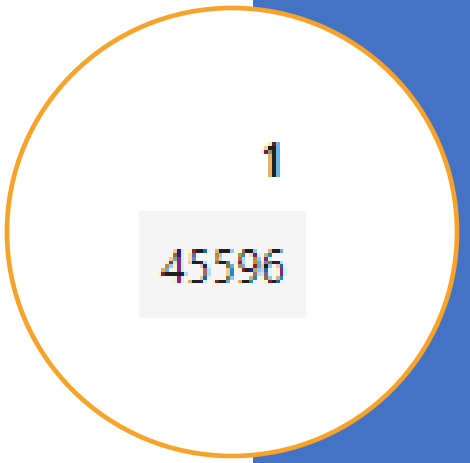
DATE	time_utc_	booster_version	launch_site	payload	payload_mass_kg_	orbit	customer	mission_outcome	landing_outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight 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 SQL Query Code:

```
%sql select sum(payload_mass__kg_) from SPACEXTBL where CUSTOMER = 'NASA (CRS)'
```

- Using sum( ) function to get the total value and the where statement to select the rows



Average Payload Mass by F9 v1.1

- SQL Query Code

```
%sql select avg(PAYLOAD_MASS__KG_) from SPACEXTBL where BOOSTER_VERSION = 'F9 v1.1'
```

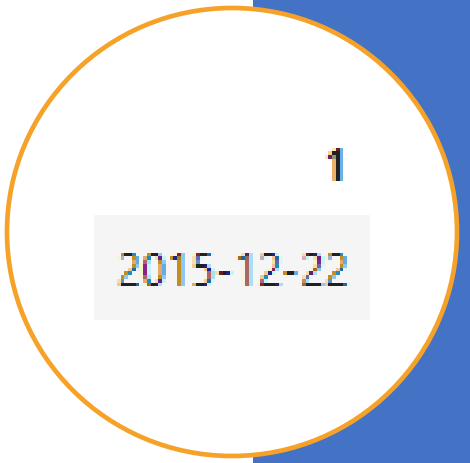
- Avg( ) function to get mean and where statement select the rows

First Successful Ground Landing Date

- SQL Query Code:

```
%sql select min(DATE) from SPACEXTBL where Landing__Outcome = 'Success (ground pad)'
```

- The min( ) function find the earliest date successful ground landing.



Successful Drone Ship Landing with Payload between 4000 and 6000

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

- SQL Query Code:

```
%%sql
select BOOSTER_VERSION from SPACEXTBL
where LANDING__OUTCOME='Success (drone ship)'
    and (PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000)
```

- Where statement selects the rows, and use AND to take more than one conditions.

Total Number of Successful and Failure Mission Outcomes

- SQL Query Code:

```
%%sql
select MISSION_OUTCOME, count(MISSION_OUTCOME) as Num from SPACEXTBL
group by MISSION_OUTCOME;
```

- The group by statement to get frequency of the variable. And must figure out the two different success value.

mission_outcome	num
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

Boosters Carried Maximum Payload

- SQL Query Code:

```
%%sql
select booster_version from SPACEXTBL
where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEXTBL)
```

- The condition of where statement can be a subquery which is very efficient method to achieve the goal

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

- SQL Query Code:

```
%%sql
select BOOSTER_VERSION, LAUNCH_SITE, LANDING__OUTCOME from SPACEXTBL
where LANDING__OUTCOME = 'Failure (drone ship)'
    and YEAR(DATE) = 2015
```

- The year( ) function extract the year from the date.

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

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

- SQL Query Code:

```
%%sql
select LANDING__OUTCOME, count(LANDING__OUTCOME) as types from SPACEXTBL
where DATE BETWEEN '2010-06-04' AND '2017-03-20'
group by LANDING__OUTCOME
order by count(LANDING__OUTCOME) desc
```

- The order by statement sorts the dataset and the desc function display data in descending order

landing__outcome	types
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

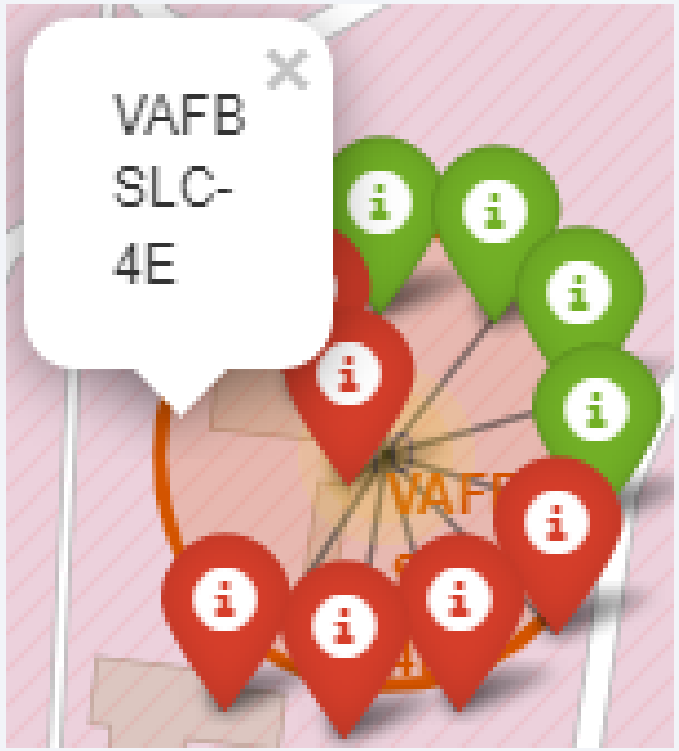
Section 4

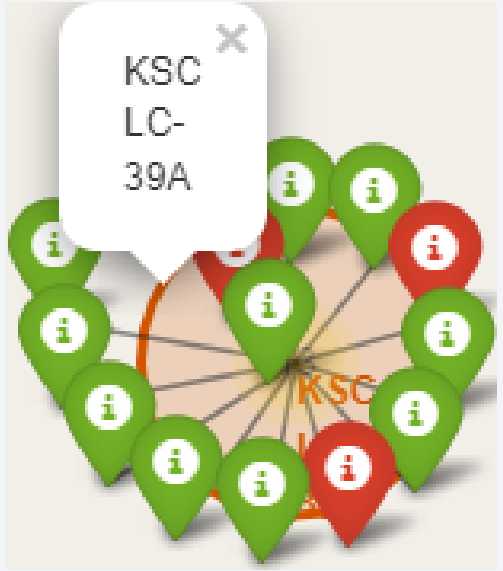
Launch Sites Proximities Analysis

All Launch Sites Markers on Global Map

- The global map shows SpaceX launch sites are in the south America coasts. One is in the Florida and the other three are in California

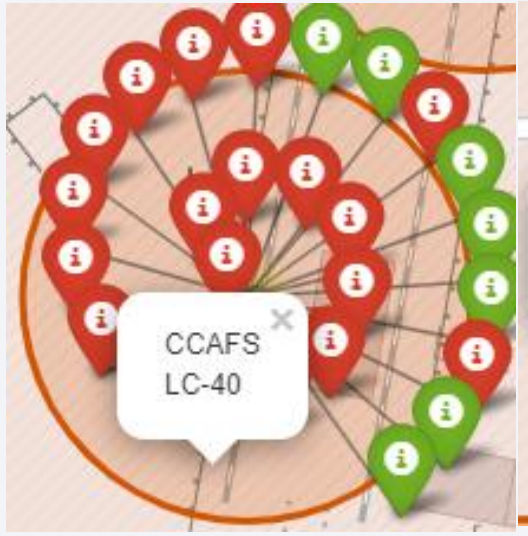
Enhance Map by Adding Cluster of Marker



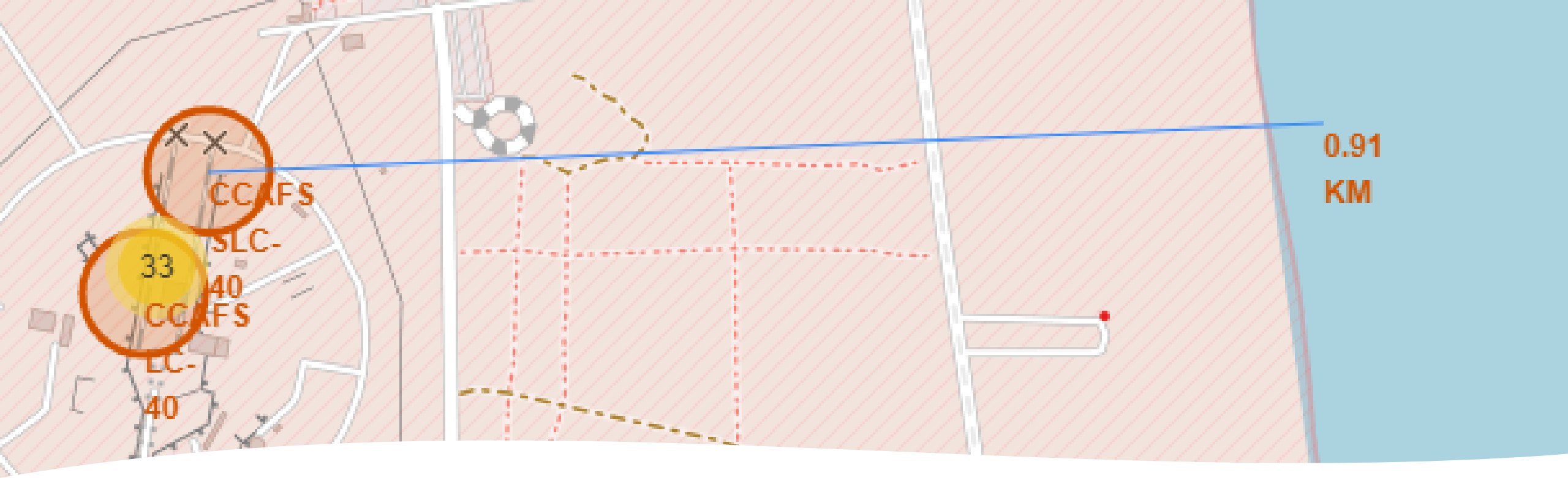


California Launch Site

Florida Launch Sites (Right three pictures)

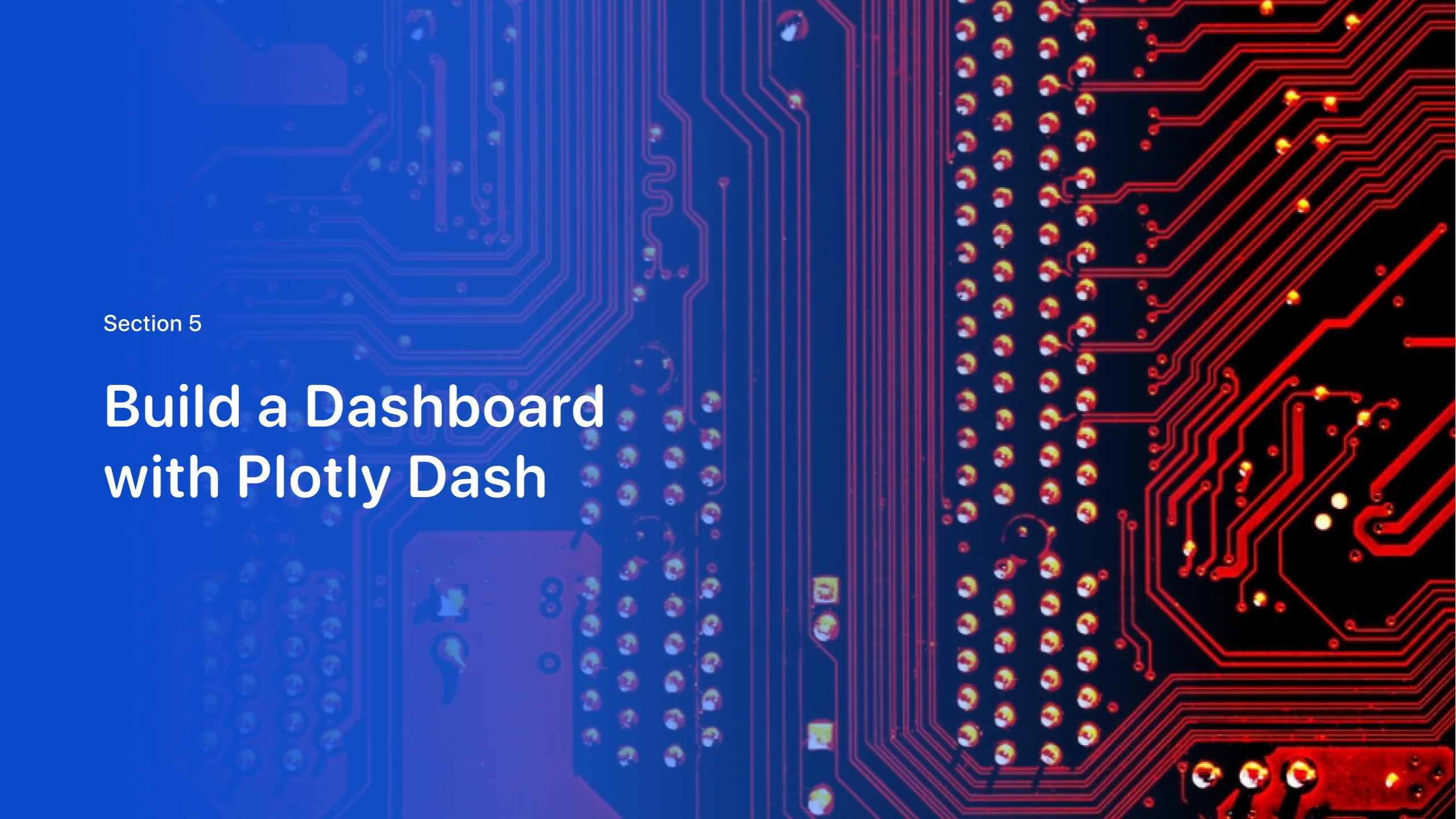


Green Marker is successful and red is Failure



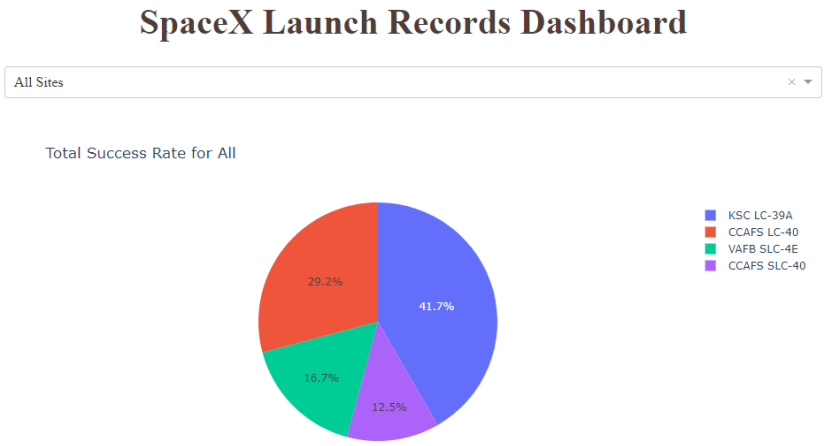
Calculate Distances between Launch Site to its Proximities

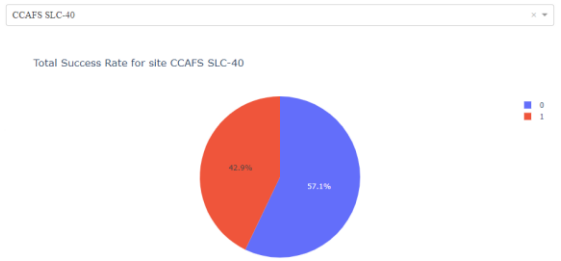
- The concerned Problem:
  - Are launch sites in close proximity to railways? Yes
  - Are launch sites in close proximity to highways? Yes
  - Are launch sites in close proximity to coastline? Yes
  - Do launch sites keep certain distance away from cities? No
- The transportation is the most important for a launch site and it is also important to keep a high quality living environment to residents.

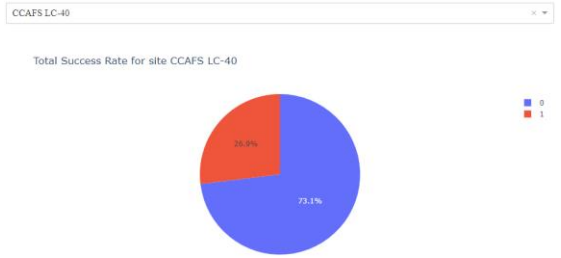


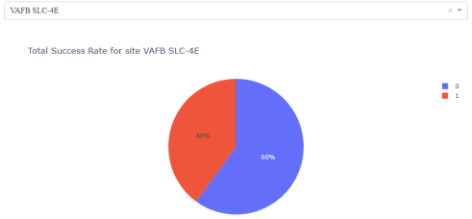
Section 5

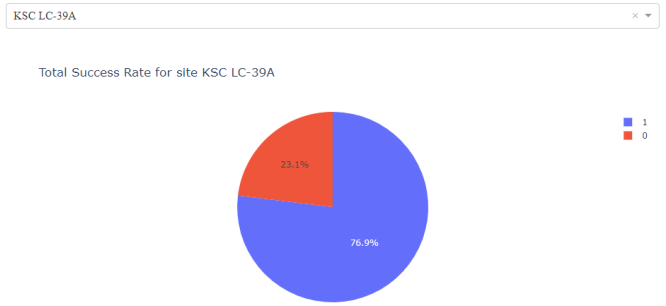
Build a Dashboard with Plotly Dash









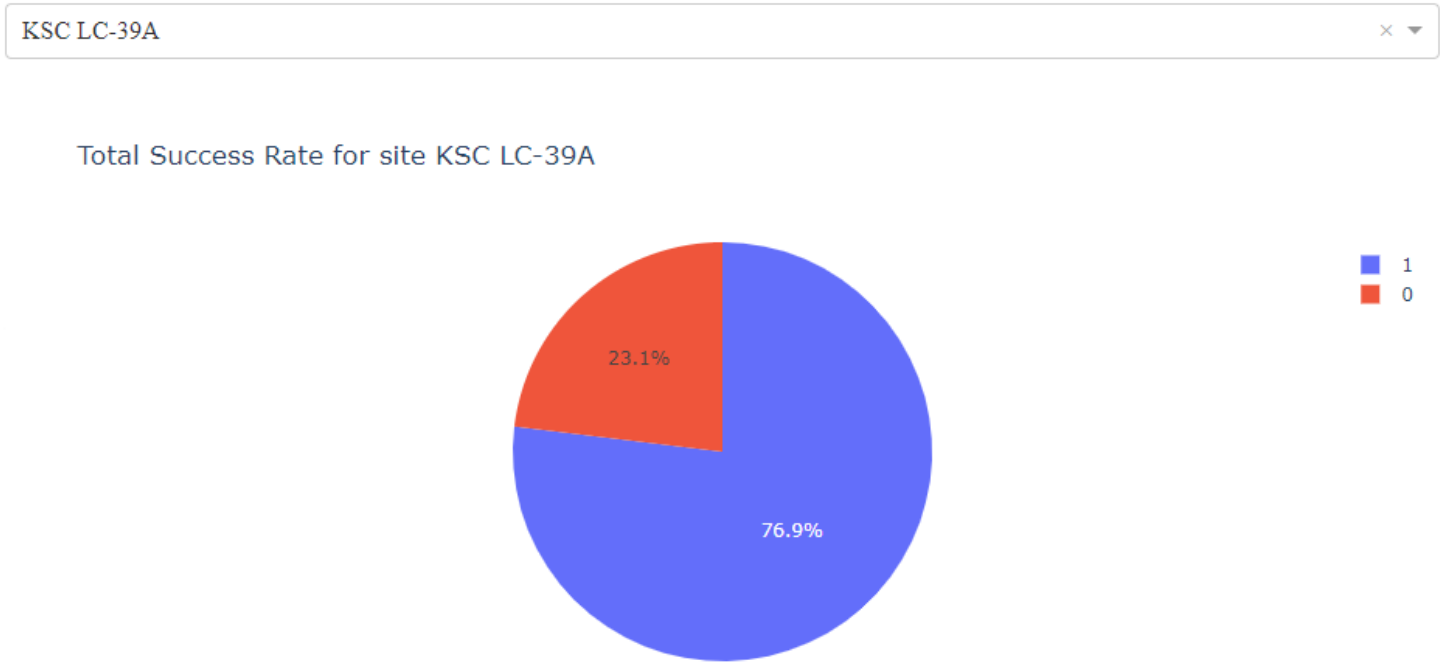


Success Rate Pie Chart

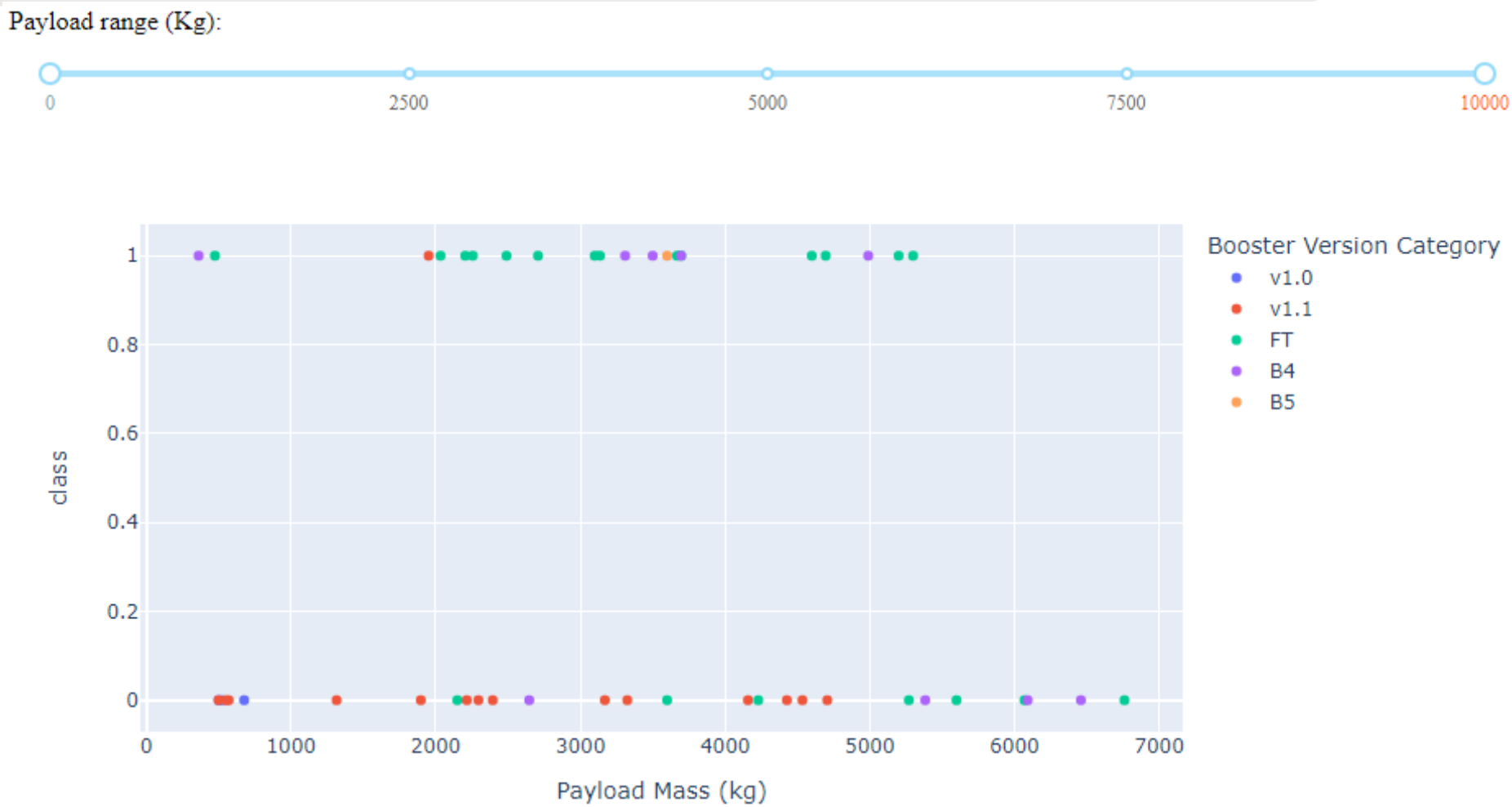
- The first pie chart shows the success rate for all the launch sites
- The other four are CCAFS LC-40, VAFB SLC-4E, KSC LC-39A, and CCAFS SLC-40.
- "1" is success and "0" is failure.

KSC LC-39A

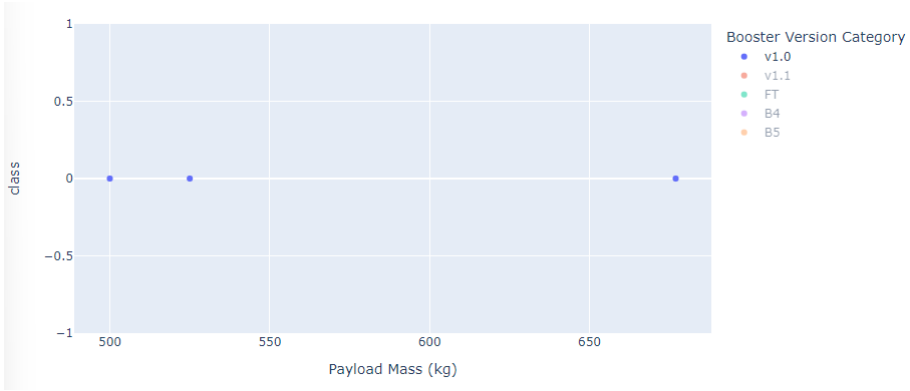
- The pie chart for KSC LC-39A with the highest launch success ratio
- Frow the pie chart, we can see the launch success rate is 76.9%



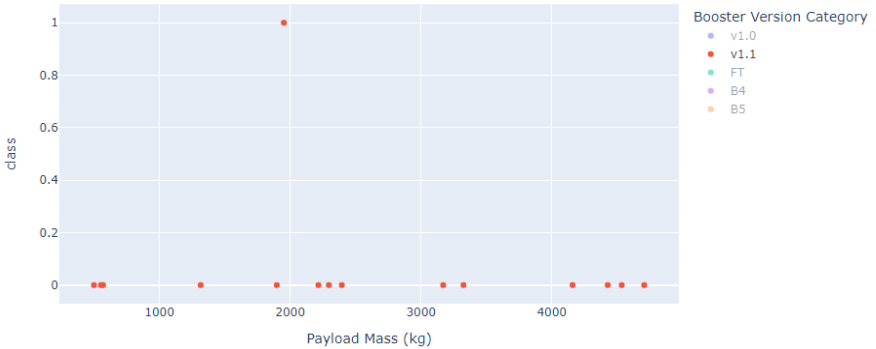
Payload vs. Launch Outcome Scatter Plot for All Sites

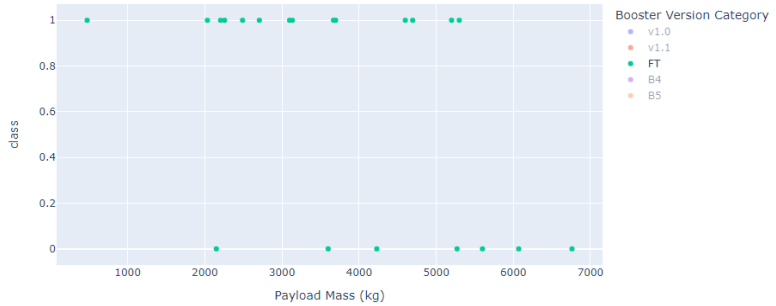


Payload vs. Launch Outcome Scatter Plot

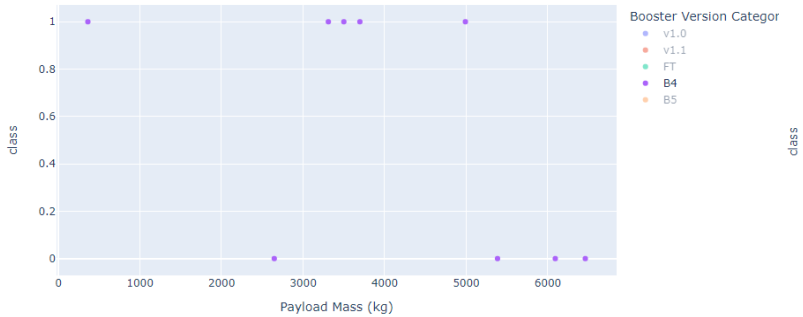


- V1.0: 100% Failure (3)
- V1.1: 93% Failure (14)
  7% Success (1)

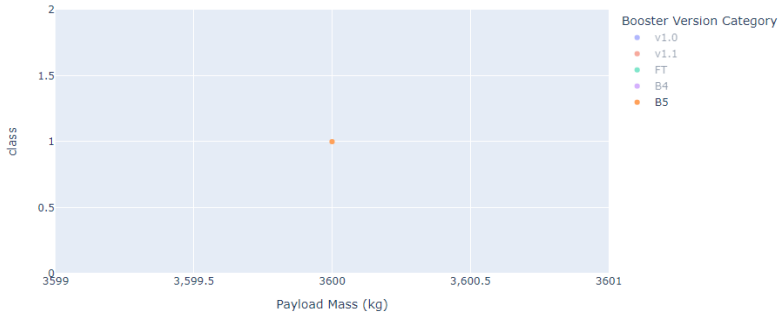




- FT: 67% Success (14)
  33% Failure (7)



- B4: 56% Success (5)
  44% Failure (4)



- B5: 100% Success (1)
  0.00% Failure (0)

Section 6

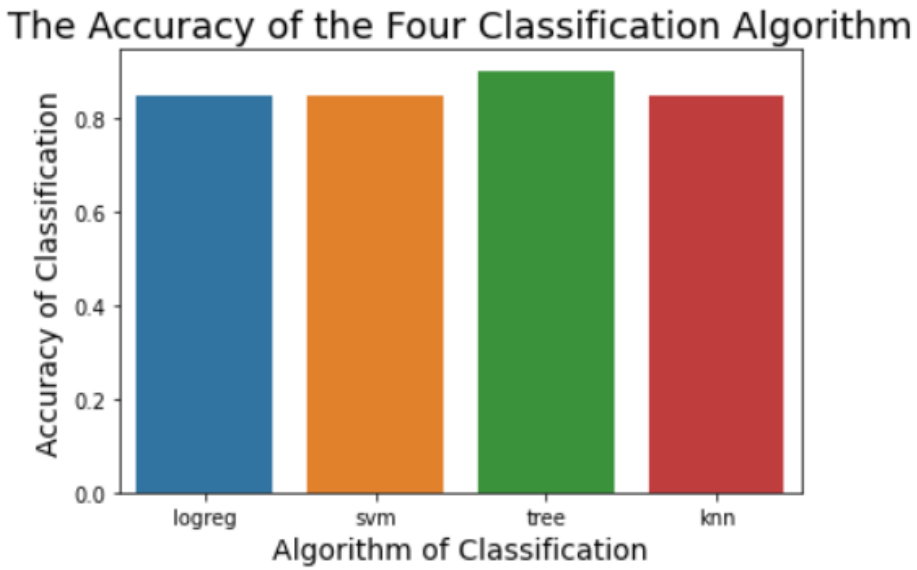
Predictive Analysis (Classification)

Classification Accuracy

- From the bar chart and the date frame at the left side, the decision tree classification algorithm got the highest accuracy, 0.901786.

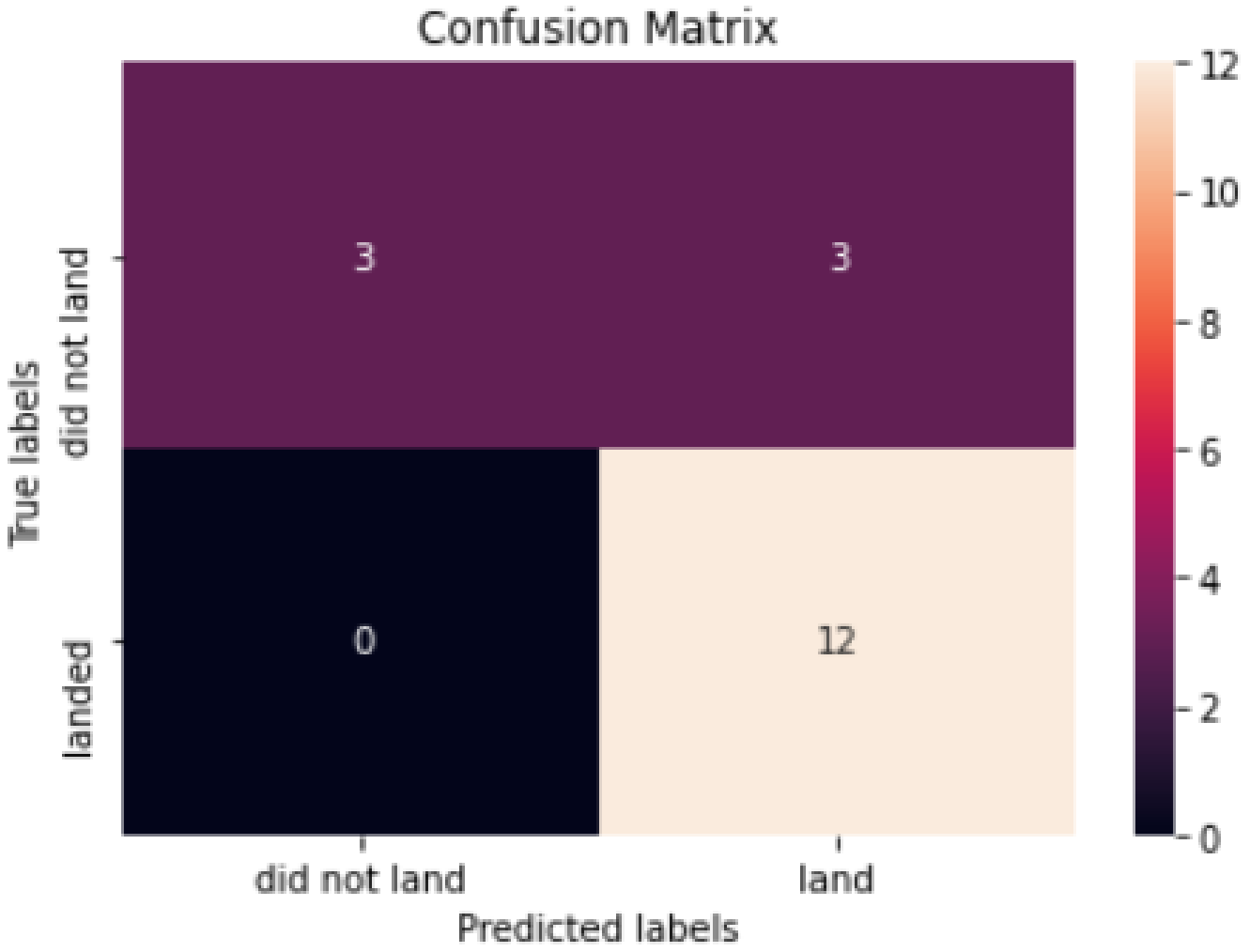
- The decision tree is the best algorithm for this training data frame depend on the accuracy.

	Algorithm	Accuracy
0	logreg	0.846429
1	svm	0.848214
2	tree	0.901786
3	knn	0.848214



Confusion Matrix

- Check the confusion matrix, the decision tree model is successful to predict the landed part, on the other hand, there are 3 false-true error.
- For solve this problem, we need to analyze the variables, add some new variables which is more correlation with target variable.



Conclusions

- The Tree classifier Algorithm is the best in the choosing four machine learning algorithm for this data frame.
- Low weighted payloads perform better than the heavier payloads
- The success rates for SpaceX launches is directly proportional time in years they will eventually perfect the launches
- KSC LC-39A has the most successful return landing rate in all four sites
- Orbit GEO, HEO, SSO, and ES-L1 have the best success return landing rate
- For the next launch, we can use the decision tree classification to predict on the collecting data

Appendix

- SpaceX API Notebook
- Scraping data from Wiki
- Data Wrangling
- EDA with Data Visuzlization
- EDA with SQL
- Interactive Map with Folium
- **Build a Dashboard with Plotly Dash**
- Predictive Analysis (Classification)

Thank you!