IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Jo-El Aramis Graulau
January 18, 2023

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection
  - Data Wrangling
  - EDA with SQL
  - EDA with Data Viz
  - Dashboard with Plotly Dash
  - Predictive Analysis
- Summary of all results
  - Data Analysis Results
  - Interactive Analytics
  - Predictive Analysis

Introduction

- Commercial space is here and multiple companies are in a competition to be the biggest space travel “airlines”. SpaceX is known as the most popular at the moment due to their affordability. The cost of admission for SpaceX’s Falcon 9 rocket is $62 million though it doesn’t seem affordable it is in comparison to the competitions which advertises for roughly $165 million. Since SpaceX can use the first stage of the Falcon 9 rocket multiple times they are able to save money thus the price difference. If another company wants to truly be a competitor to SpaceX they will have to use similar models and be able to reuse the first stage, we predict the first stage of the Falcon 9 is successful and we can find the cost of launch for the other company.

- Problems you want to find answers
    - How to get the best results and be successful in reusing the first stage via landing.
    - If there are any correlations between certain variables of the rocket and landing successfully.

Section 1

Methodology

Methodology

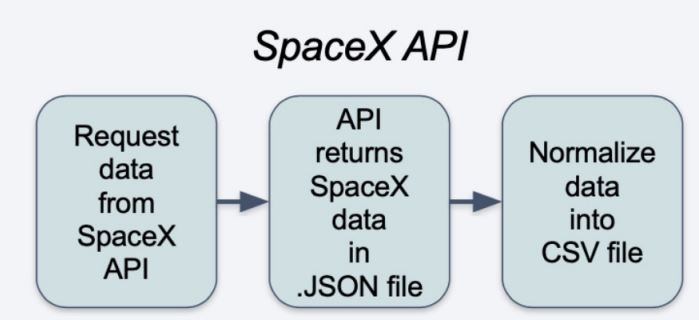
- Executive Summary
- Data collection methodology:
  - Web Scraping of the Wikipedia section Falcon 9 and Falcon Heavy Launch Records and the SpaceX API
- Perform data wrangling
  - Converting the outcomes into Training Label with whether or not the booster was successful in landing.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Classification Trees, Logistic Regression and SVM.

Data Collection

- The data was collected via web scraping of Wikipedia page of SpaceX and Falcon 9 and Falcon Heavy Launches, also the SpaceX API.
  - Wikipedia Web Scrape Attributes are Flight No., Launch Site, Payload Mass, Payload, Launch Outcome, Booster landing, Time, Date, Customer, Orbit, Payload Mass
  - SpaceX API Attributes are Reused Count, Longitude, Latitude, Landing Pad, Reused, Legs, Block, Flights, Flight Number, Date, Booster Version, Orbit, Serial, Grid Fins, Payload Mass, Launch Site,
- Flow Chart inserted below

```
launch_dict = {'FlightNumber': list(data['flight_number']),
'Date': list(data['date']),
'BoosterVersion':BoosterVersion,
'PayloadMass':PayloadMass,
'Orbit':Orbit,
'LaunchSite':LaunchSite,
'Outcome':Outcome,
'Flights':Flights,
'GridFins':GridFins,
'Reused':Reused,
'Legs':Legs,
'LandingPad':LandingPad,
'Block':Block,
'ReusedCount':ReusedCount,
'Serial':Serial,
'Longitude': Longitude,
'Latitude': Latitude}
```

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




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

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

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

Data Collection – SpaceX API

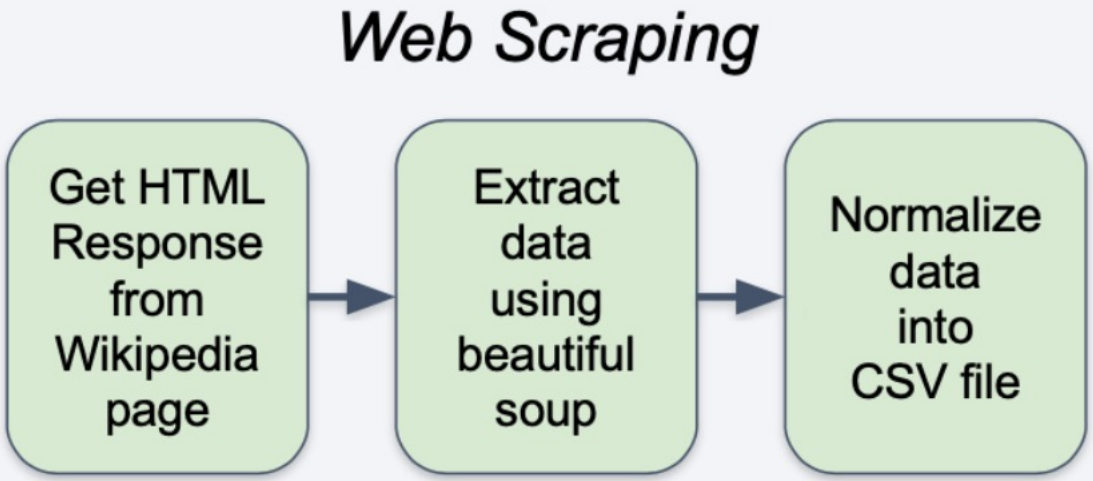
Data Collection - Scraping

```
html_data = requests.get(static_url).text
```

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

```
html_tables = soup.find_all('table')
```

```
for row in first_launch_table.find_all('th')
    name = extract_column_from_header(row)
    if(name != None and len(name) > 0):
        column_names.append(name)
```




```
launch_dict= dict.fromkeys(column_names)

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

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']=[]
```

Data Wrangling

- There were multiple cases where they were unsuccessful in landing either via accident or failure of the booster.
  - True RTLS: successfully landed on the ground pad
  - True Ocean: successfully landed in the ocean area
  - True ASDS: successfully landed on drone ship
  - False RTLS: failed to land on ground pad
  - False Ocean: failed to land in ocean area
  - False ASDS: failed to land on drone ship
- 0 was failed while 1 is successful

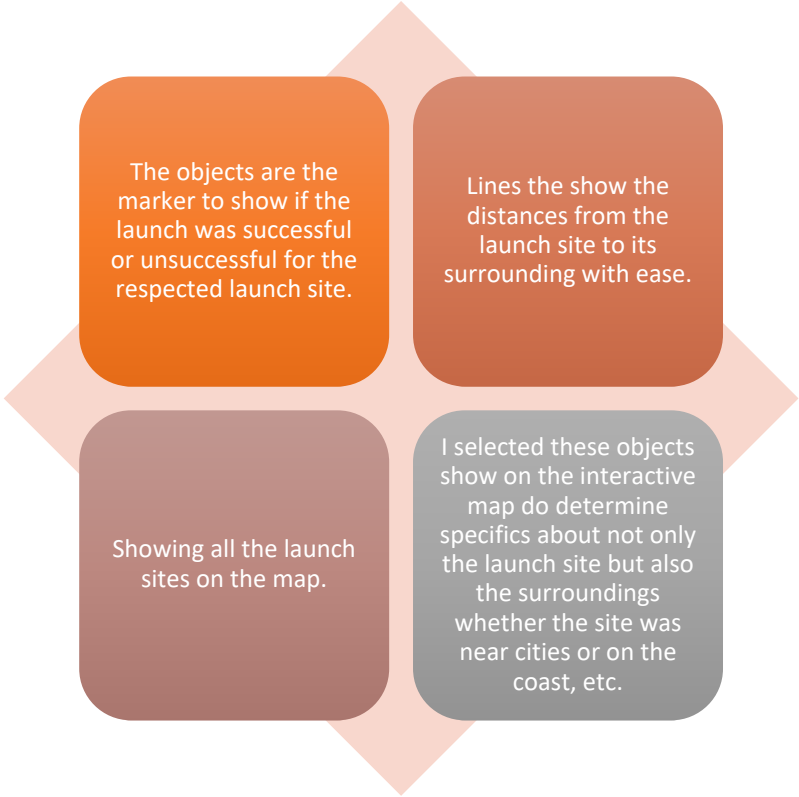
EDA with Data Visualization

- Bar Chart
  - I prefer using bar charts when I need to compare 2 different data points in this case it was between Success Rate and the type of orbit.
- Line Chart
  - I chose to use a line chart due to it being ideal to view trends clearly, where it was showing the rate of success and the years of each data point.
- Scatter Chart
  - The scatter chart shows correlation extremely well and we'll be able to see exactly how much a certain variable affected or lack there of to another. Here it would include the payload and the launch site, payload and the type of orbit, the flight number and the launch site and the flight number and the type of orbit.

EDA with SQL

- The queries I performed for this dataset included
  - The dates when there was a successful landing from each of the landing sites
  - The total of successful and failed missions.
  - Names of the boosters with the payload more than 4000 but less than 6000
  - Names of the booster with the max payload
  - The names of the distinct launch sites
  - The average payload mass by the F9 version 1
  - The first 5 launch sites where the site began with CCA
  - In descending order the outcomes of the landing between June 2010 and March 2017

Build an Interactive Map with Folium



Build a Dashboard with Plotly Dash

Scatter Chart

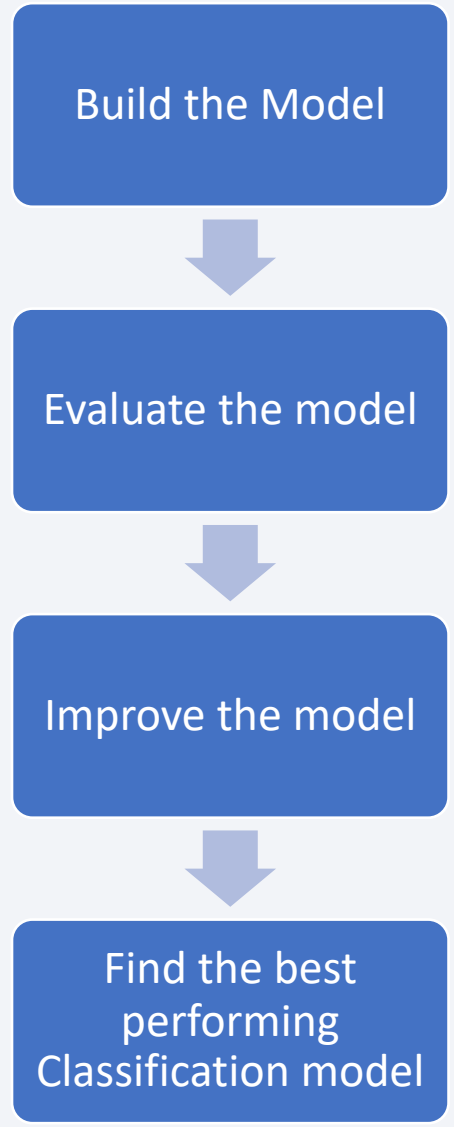
- The only 2 inputs are the Payload mass with a slider up to 10000kg and all of the launch sites.
- To show the relationships between the payload mass in kg and outcomes for the different boosters.

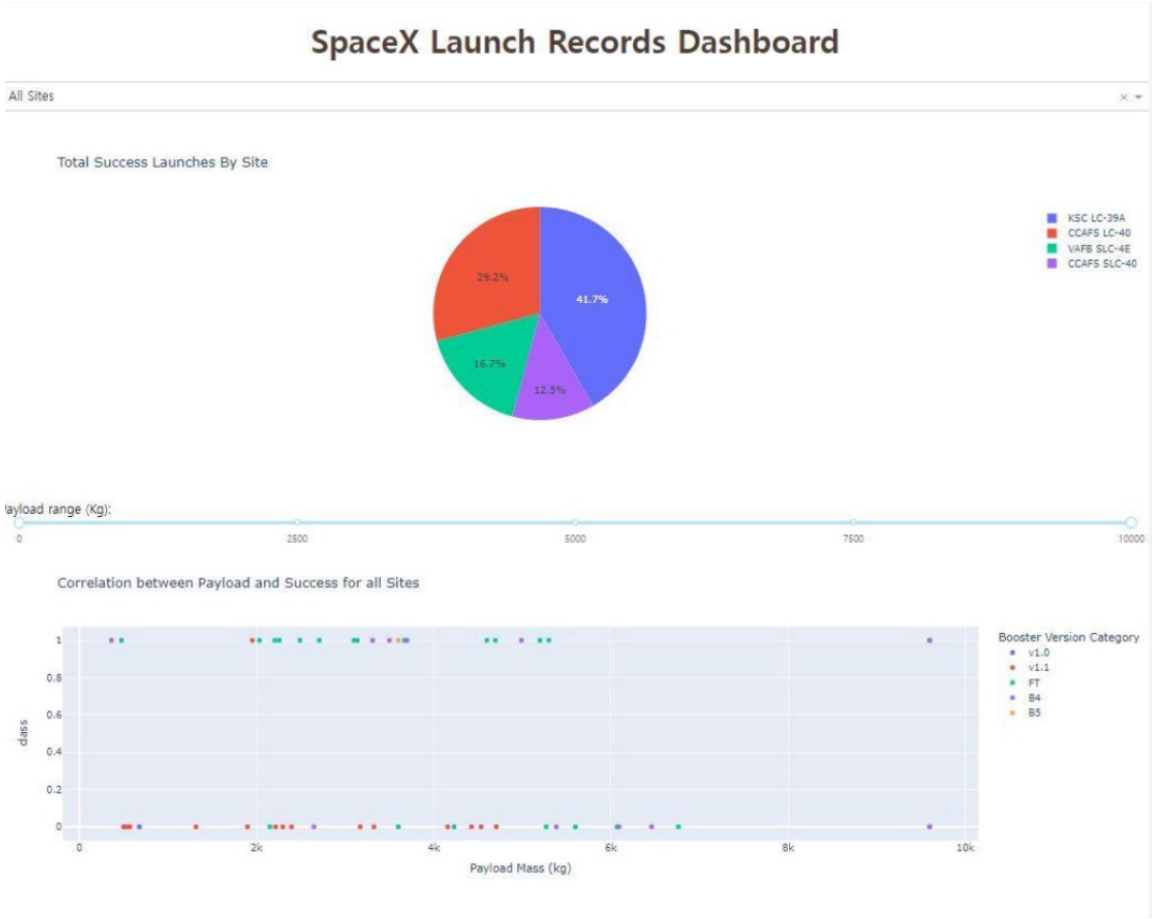
Pie Chart

- To show whether the launch sites were successful and the locale for the successful launches.
- Show the successes of the different launches by site.

Predictive Analysis (Classification)

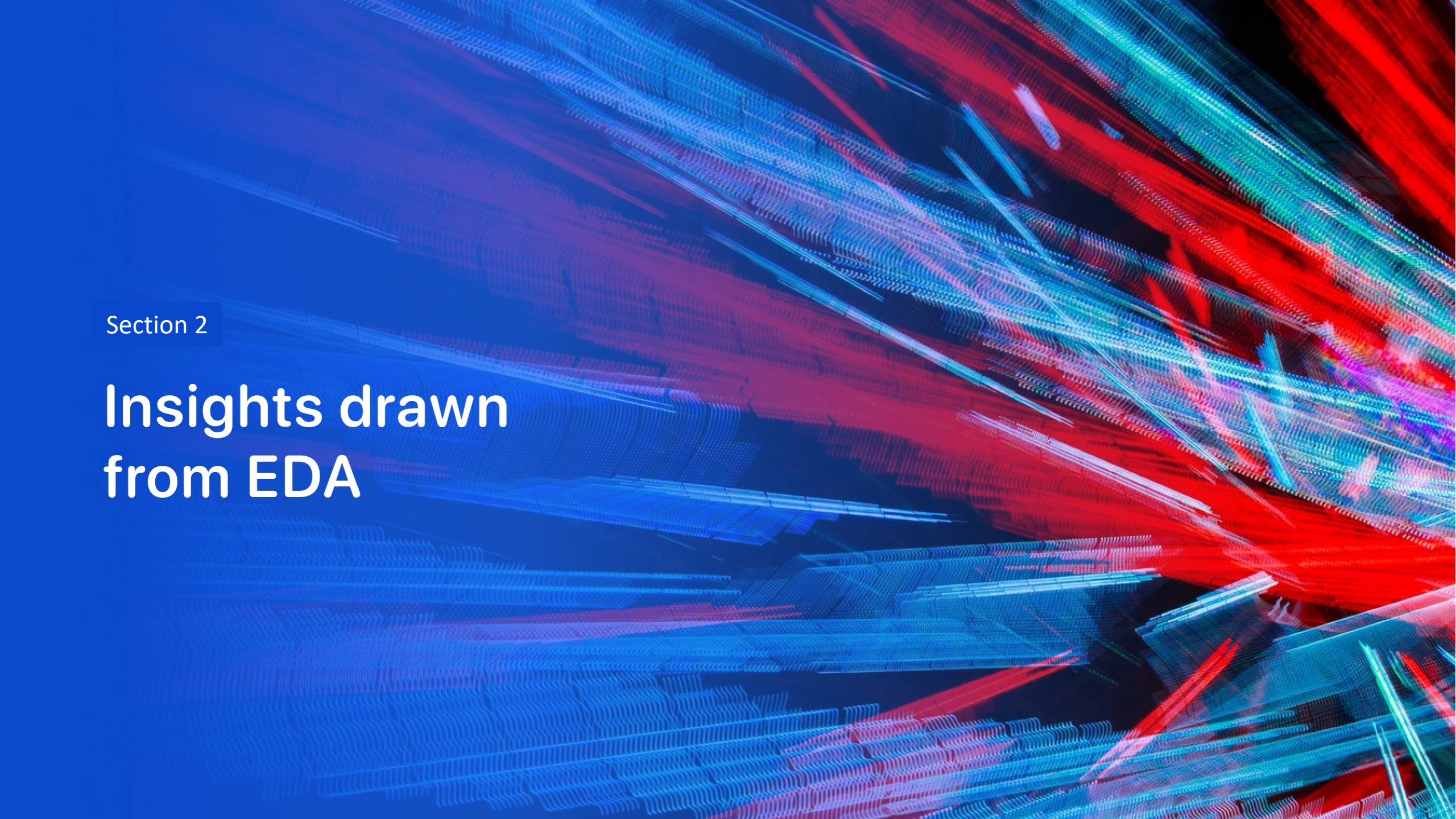
- Performing the analysis while determining the training labels
  - Standardizing the data from the dataset
  - Splitting the data between training or test
  - Creating the attribute for the respective class.
- Select the optimal Classification Trees, Logistic Regression and Hyperparameter for the SVM.





Results

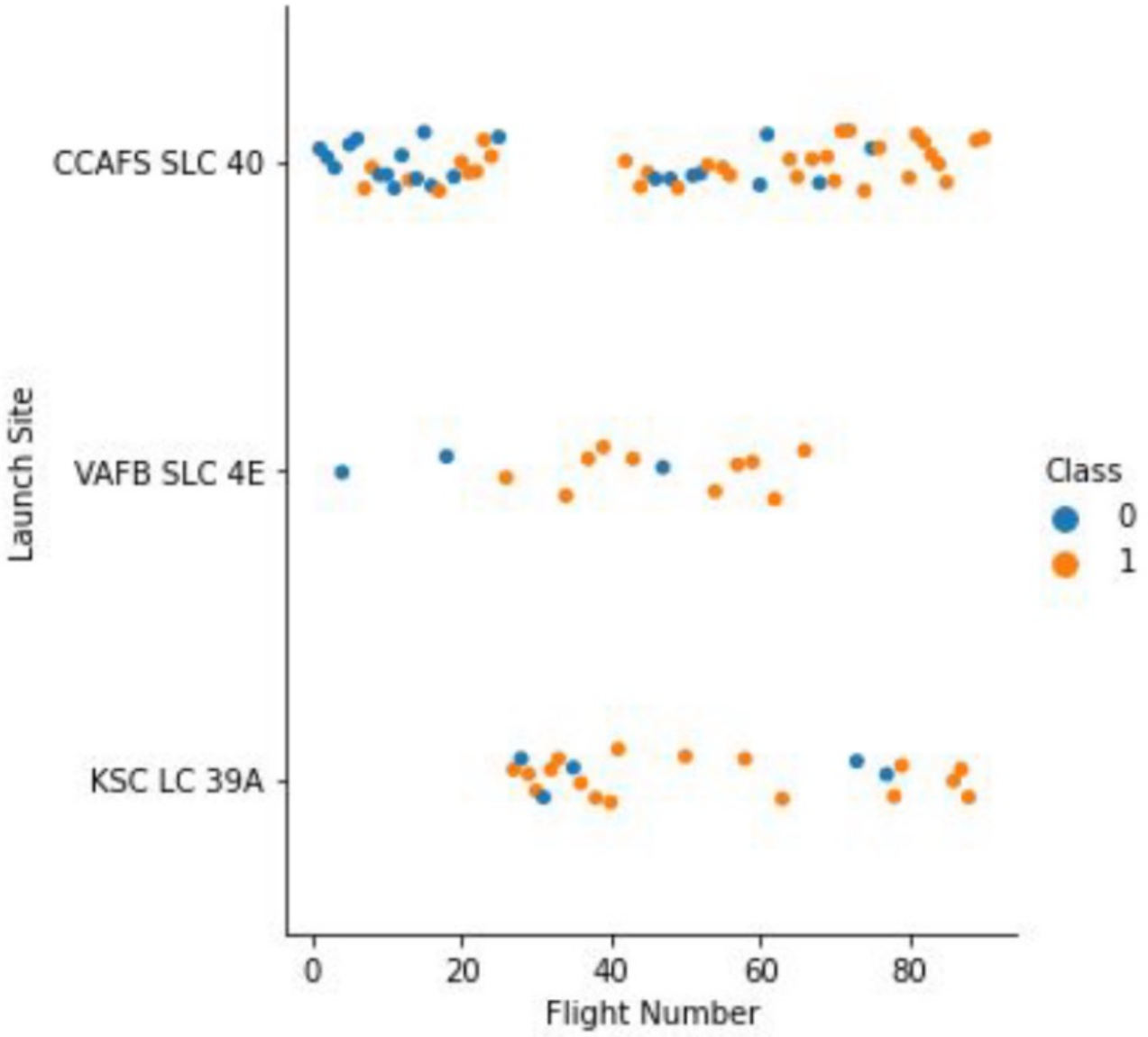
- The preview of the Dashboard is presented here.
- The accuracy between all the methods is 83% for the test data.


Section 2
Insights drawn from EDA

Flight Number vs. Launch Site

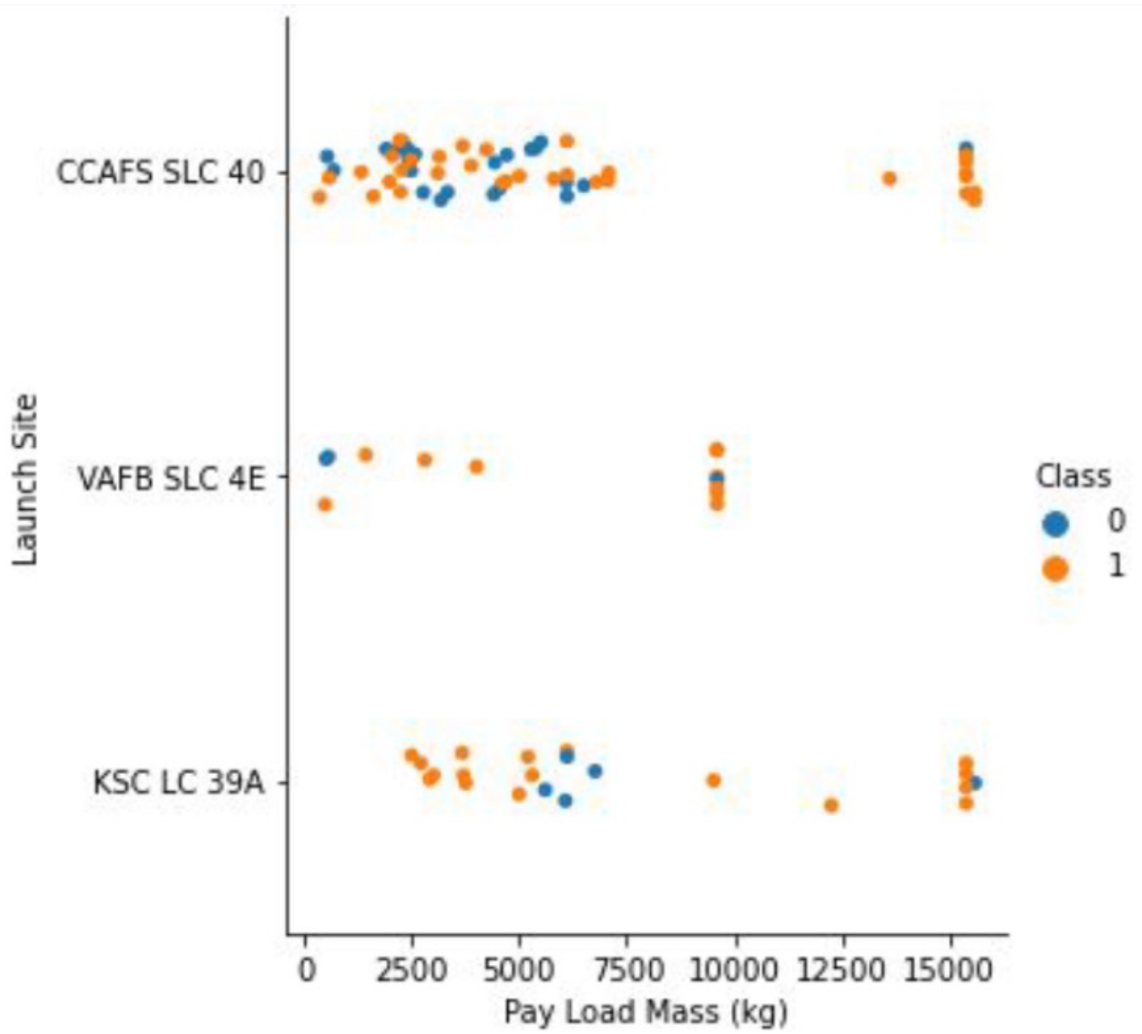
- Here is shown the success rates increasing as the flights also increased.

- Post the 20th flight there is a considerable jump in success.



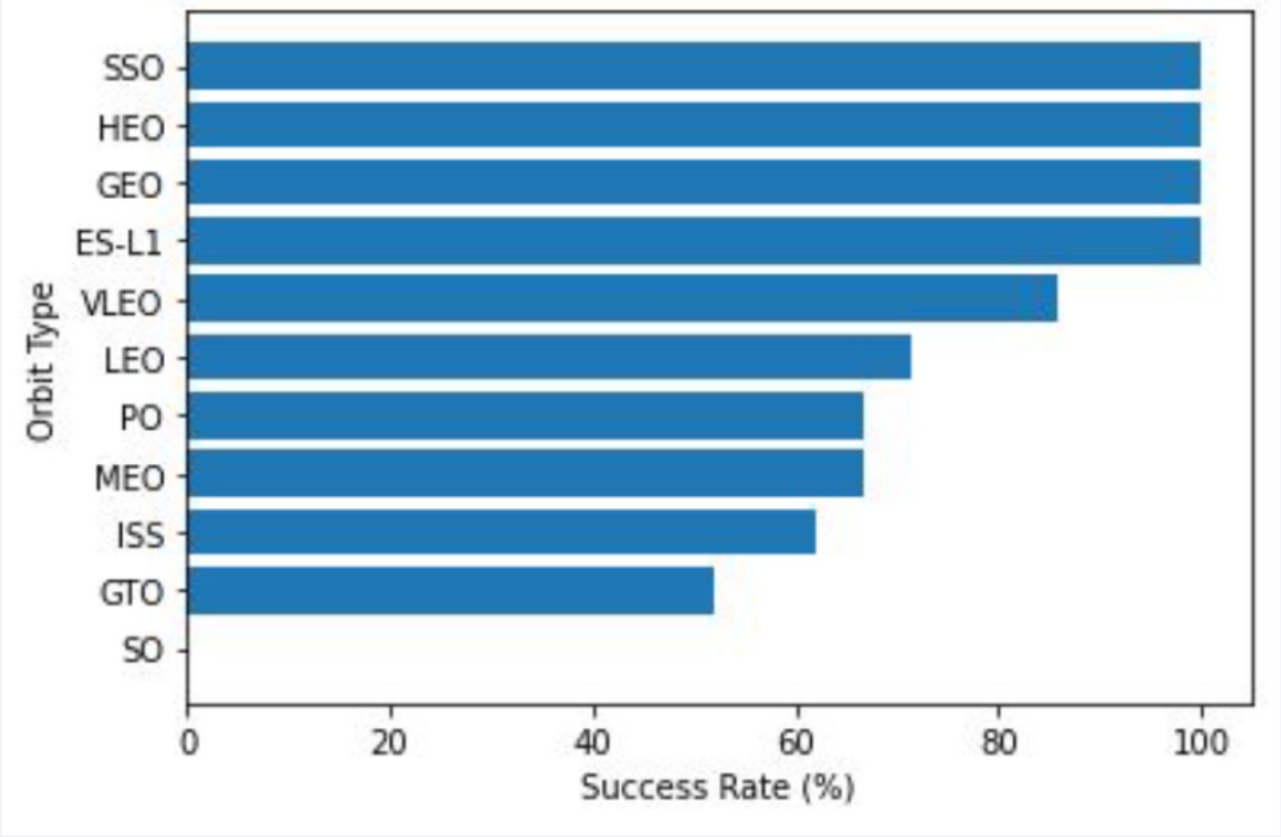
Payload vs. Launch Site

- There really isn’t an obvious deduction made with this scatter chart as there isn’t a clear correlation between the two.



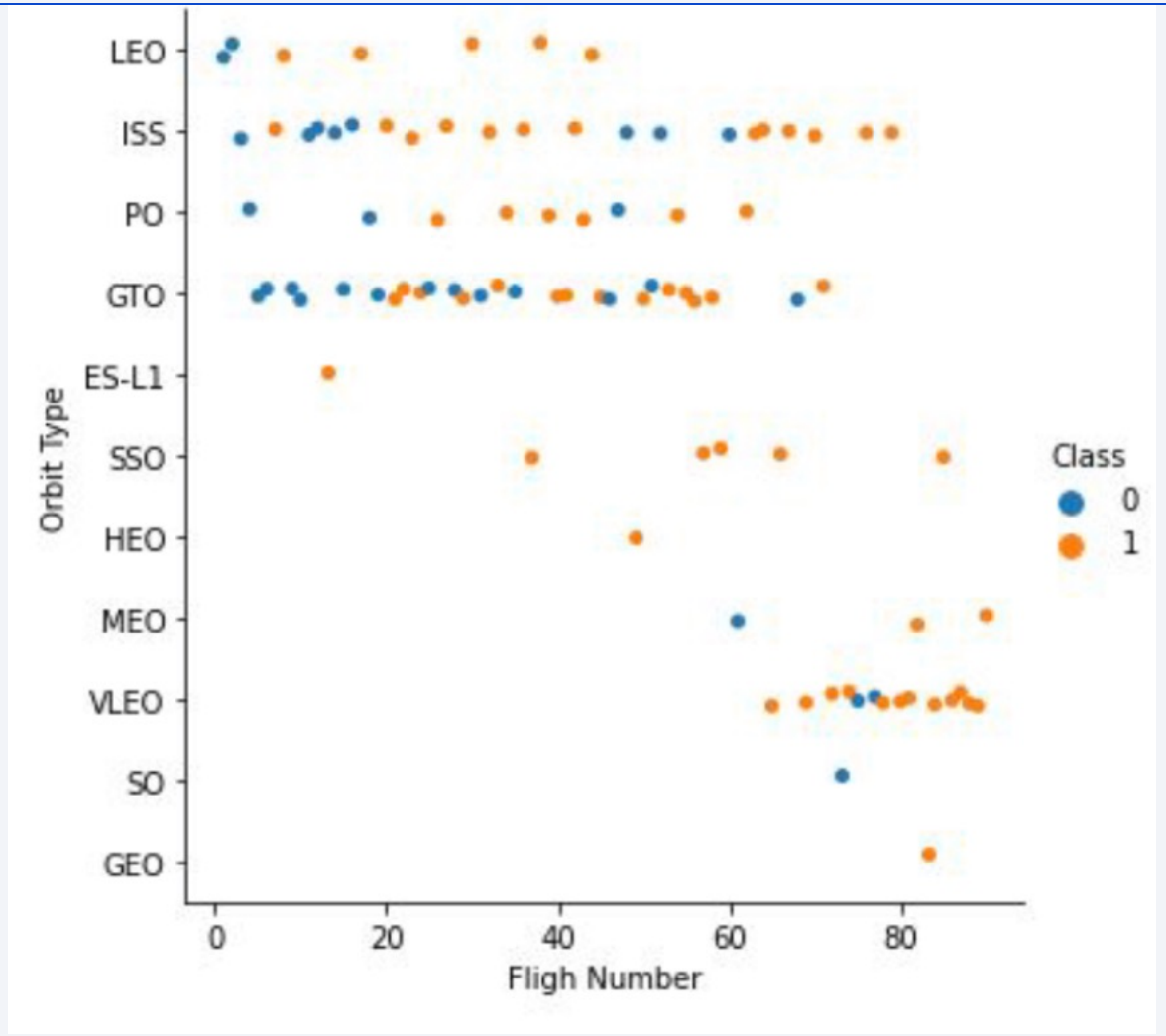
Success Rate vs. Orbit Type

- ES-L1, SSO, HEO, and GEO are the orbit types with the highest success rates with those being flawless at the moment. While GTO and SO are the lowest though the SO is somewhat of an outlier with its one attempt being a failure and later scrapped.



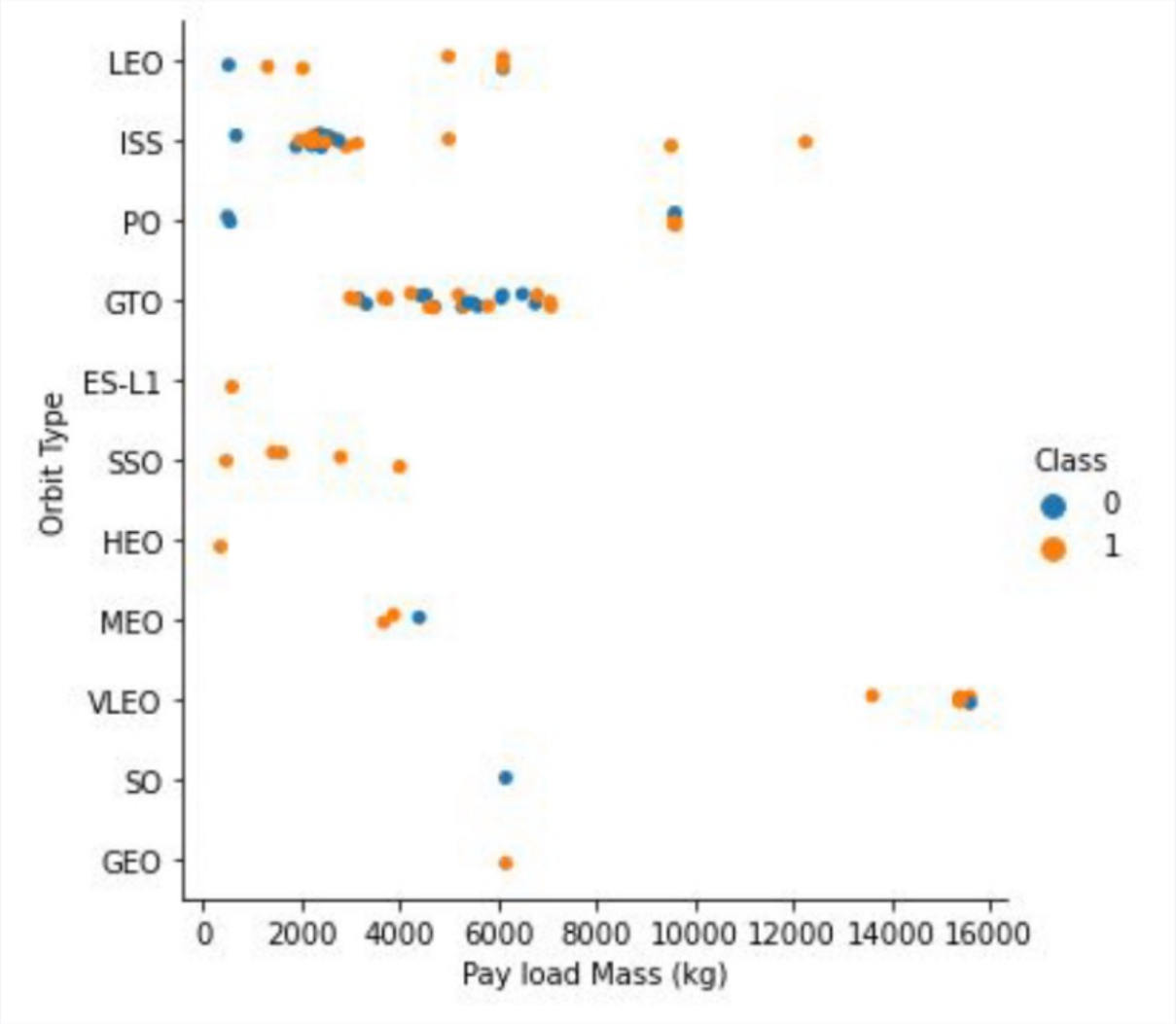
Flight Number vs. Orbit Type

- It seems that there is a correlation between the flight number.
- Beginning with LEO which is down the middle of the pack and VLEO which is the highest success rate is being used the most.



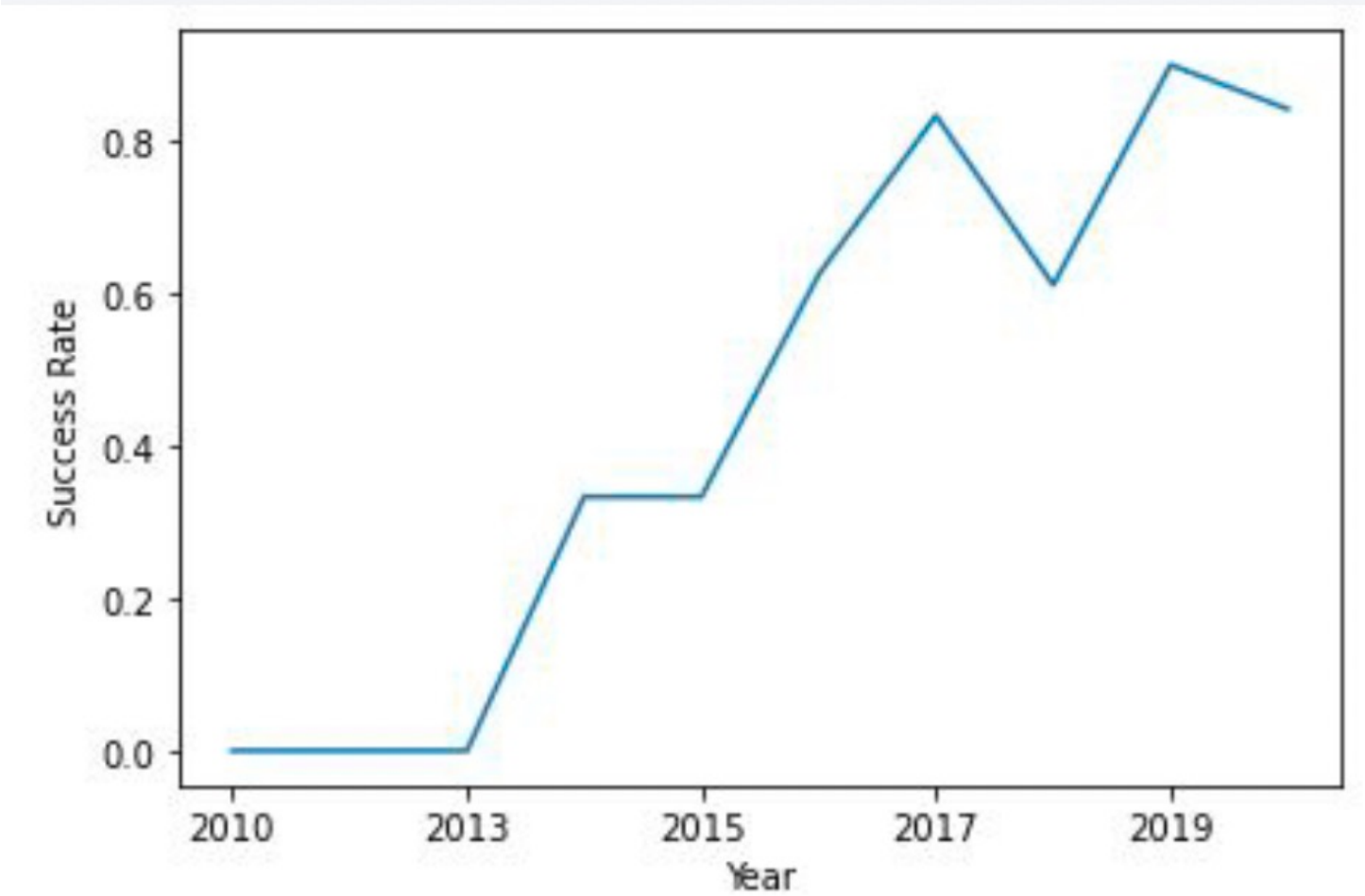
Payload vs. Orbit Type

- GTO cannot be differentiated as all of the results are clustered on top of one another.
- VLEO seems suitable for the heavier payloads with ISS and there aren't as many failures with the heavier payload compared to the lighter though there aren't nearly as many attempts.



Launch Success Yearly Trend

- 2013 was the year the success rate jumped until 2018 where there was a down year and increased slightly at the end of the year where it has hovered around 0.8 or 80%.



All Launch Site Names

```
SELECT DISTINCT LAUNCH_SITE
FROM SPACEXTBL
```

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

- Using the DISTINCT clause to find unique values in the column of Launch_site from the SpaceX table.
- The four unique sites are presented here.

```
SELECT DISTINCT LAUNCH_SITE
FROM SPACEXTBL
```

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

Launch Site Names Begin with 'CCA'

- Using the LIMIT clause and setting it to 5 to find the 5 which begin with ‘CCA’ using the LIKE clause with the % to pinpoint it from the SpaceX table.

Total Payload Mass

- I used the SUM function and used an alias to name it the Total Payload Mass in kg from the payload_mass_kg and used the WHERE clause to specify the customer as NASA.

```
SELECT SUM(PAYLOAD_MASS__KG_)
      AS total_payload_mass_kg
FROM SPACEXTBL
WHERE CUSTOMER = 'NASA (CRS)'
```

total_payload_mass_kg
45596

Average Payload Mass by F9 v1.1

- To find the average I used the AVG function from the payload_mass_kg column and set the alias as avg_payload_mass_kg, using the WHERE clause to selected values where the booster version is strictly F9 v1.1.

avg_payload_mass_kg
2928

```
SELECT AVG(PAYLOAD_MASS__KG_)
       AS avg_payload_mass_kg
FROM SPACEXTBL
WHERE BOOSTER_VERSION = 'F9 v1.1'
```

First Successful Ground Landing Date

- Using the MIN function to find the earliest DATE with the alias as first_successful_landing_date using the WHERE clause to find the successful landing outcomes on the ground pad.

```
SELECT MIN(DATE)
  AS first_successful_landing_date
FROM SPACEXTBL
WHERE LANDING__OUTCOME
      = 'Success (ground pad)'
```

first_successful_landing_date
2015-12-22

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

- I use the WHERE clause to receive the landing outcomes as a success specifically on the drone ship and using the AND function to specify the payload_mass_kg between 4000kg and 6000kg

```
SELECT BOOSTER_VERSION
FROM SPACEXTBL
WHERE LANDING__OUTCOME = 'Success (drone ship)'
    AND (PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000)
```

Total Number of Successful and Failure Mission Outcomes

- Using the COUNT function to count the amount the outcome from the SpaceX table and using the GROUP BY clause to group up the mission_outcome the results and have the amount next to it.

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

```
SELECT MISSION_OUTCOME,
       COUNT(*) AS total_number
FROM SPACEXTBL
GROUP BY MISSION_OUTCOME
```

Boosters Carried Maximum Payload

booster_version	payload_mass__kg_
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
F9 B5 B1051.3	15600
F9 B5 B1051.4	15600
F9 B5 B1051.6	15600
F9 B5 B1056.4	15600
F9 B5 B1058.3	15600
F9 B5 B1060.2	15600
F9 B5 B1060.3	15600

```
SELECT DISTINCT BOOSTER_VERSION,
       PAYLOAD_MASS__KG_
FROM SPACEXTBL
WHERE PAYLOAD_MASS__KG_ = (
       SELECT MAX(PAYLOAD_MASS__KG_)
       FROM SPACEXTBL)
```

- Using the MAX function in a subquery using the WHERE clause to search for the payload_mass_kg which would return the heaviest payload. With the DISTINCT clause to return unique booster_versions. Where the heaviest booster is F9 B5 B10 with its multiple versions weighing 15600 kg.

2015 Launch Records

- Using the WHERE clause to search strictly for the failed launches for the drone ship, using the AND function to show the year to be specifically 2015 where there were only 2 failures for the respective query,

```
SELECT LANDING__OUTCOME,
       BOOSTER_VERSION,
       LAUNCH_SITE
FROM SPACEXTBL
WHERE LANDING__OUTCOME
          = 'Failure (drone ship)'
       AND YEAR(DATE) = '2015'
```

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

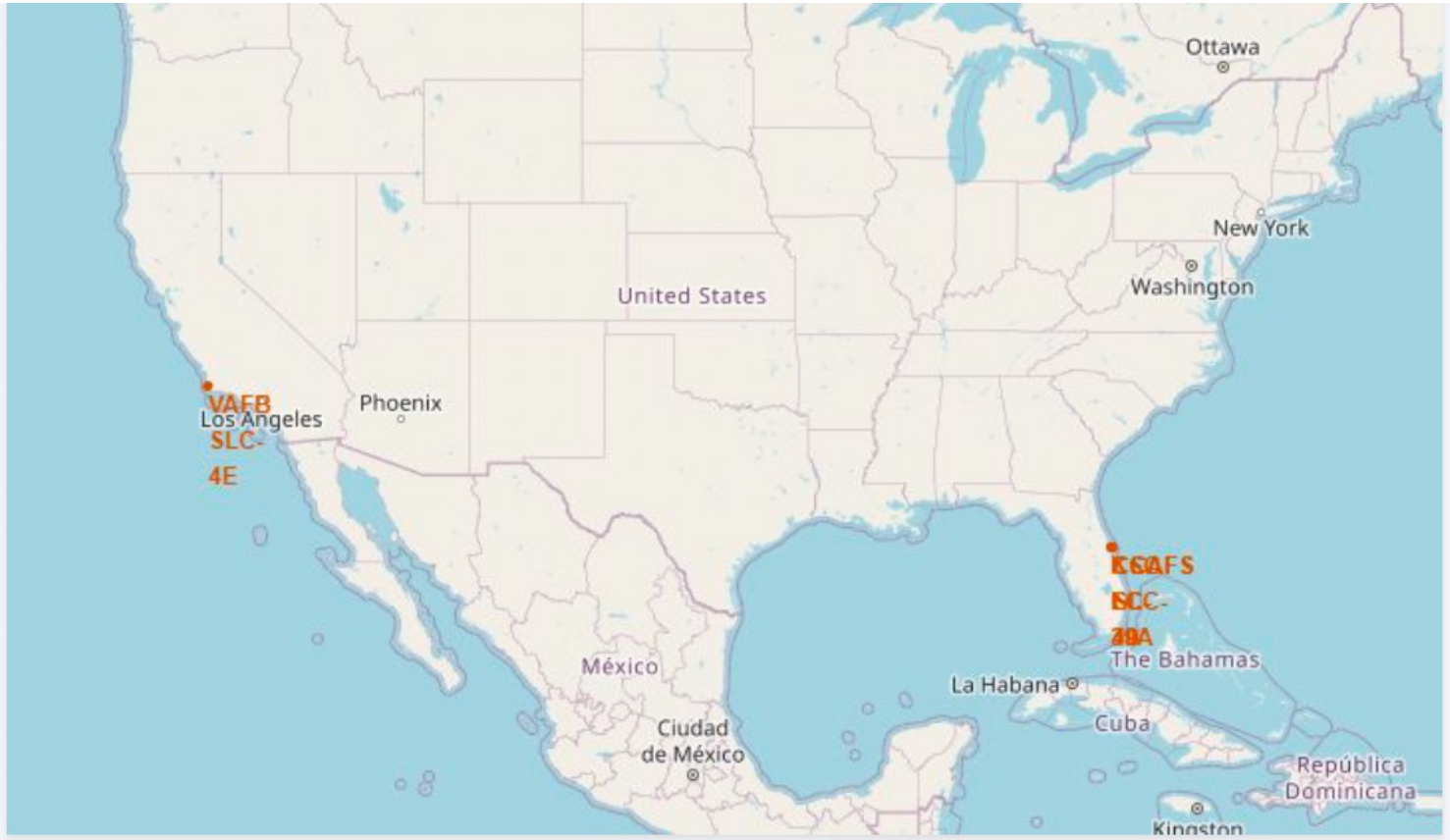
- Using the WHERE clause to filter the dates between June 2010 and March 2017. With the ORDER BY clause to order the total_number in descending order, group up by the outcome of landing and counting the outcome with its total_number alias.

```
SELECT LANDING__OUTCOME,
       COUNT(LANDING__OUTCOME) AS total_number
FROM SPACEXTBL
WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY LANDING__OUTCOME
ORDER BY total_number DESC
```

landing__outcome	total_number
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

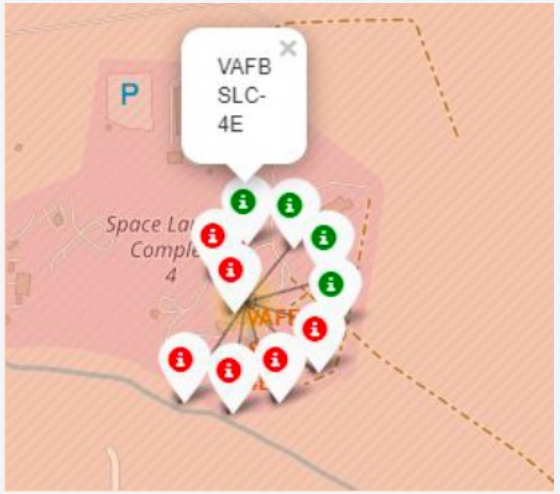
Launch Locations

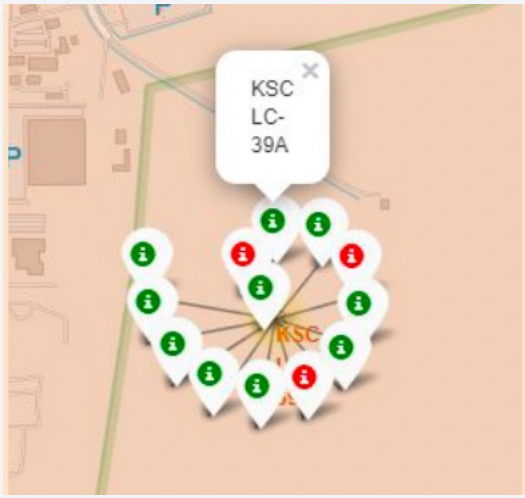


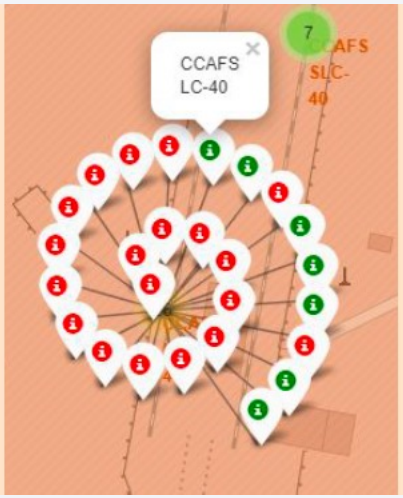
- Here is the map with the launch locations of SpaceX are currently in the US and near the coast in the south mainly in space coast FL.

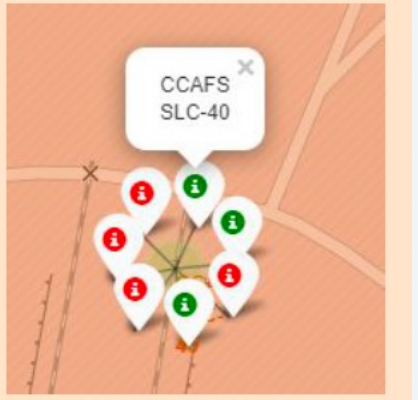
Launch Successes/Failures

- Here there are the marker clusters with green determining success and red determining failure.

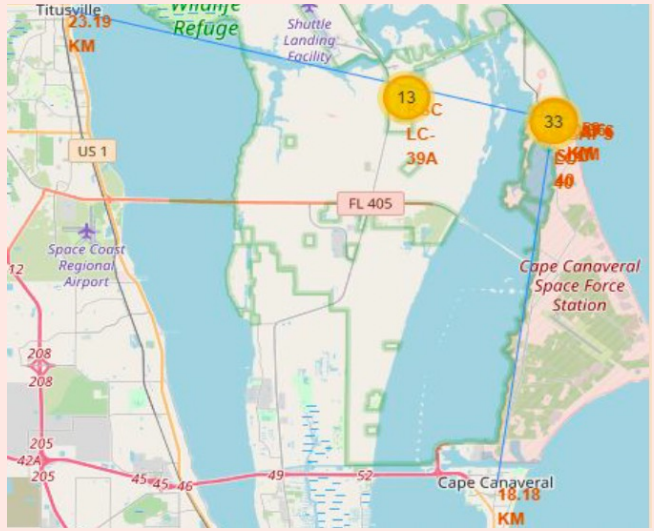




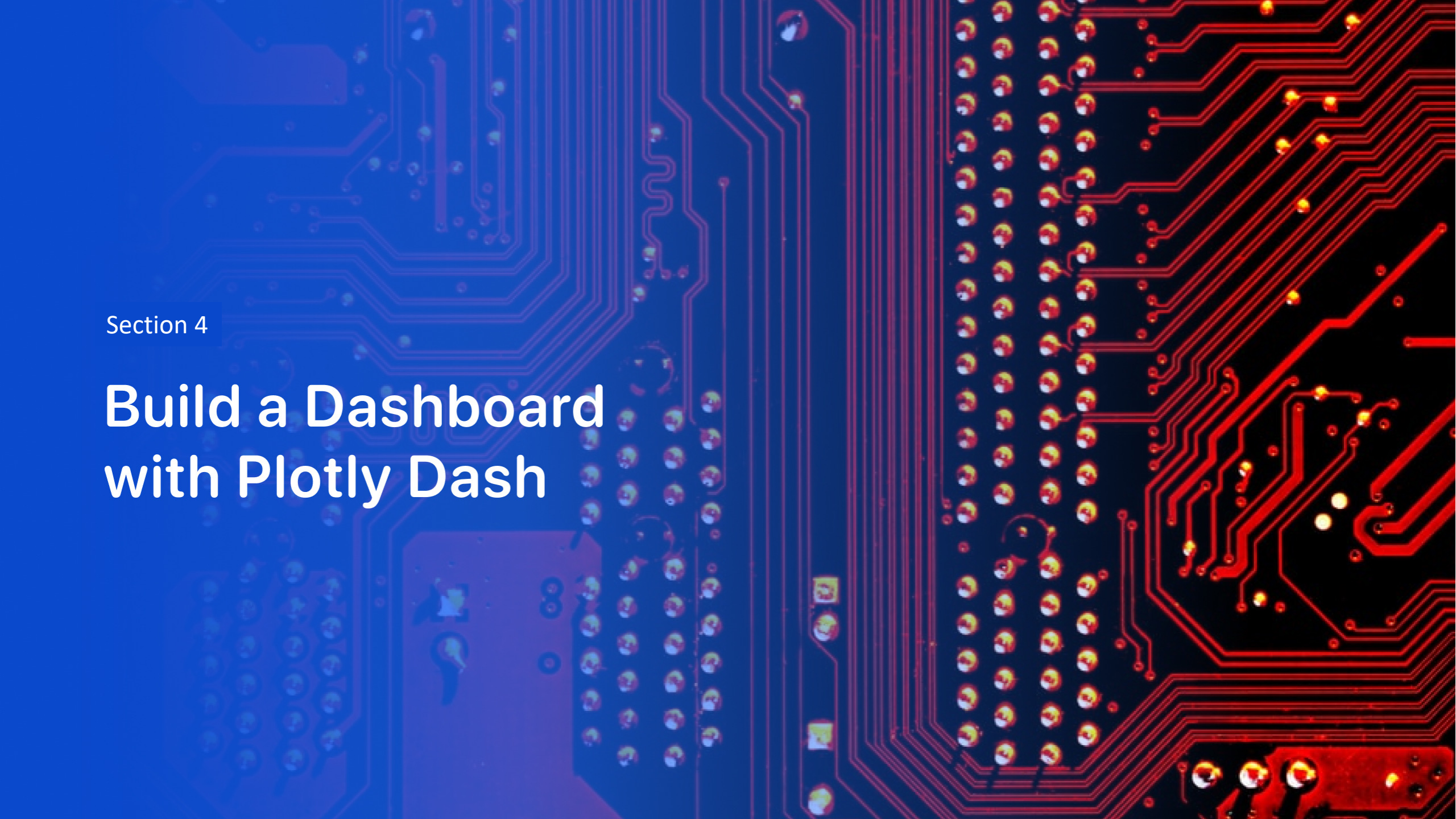




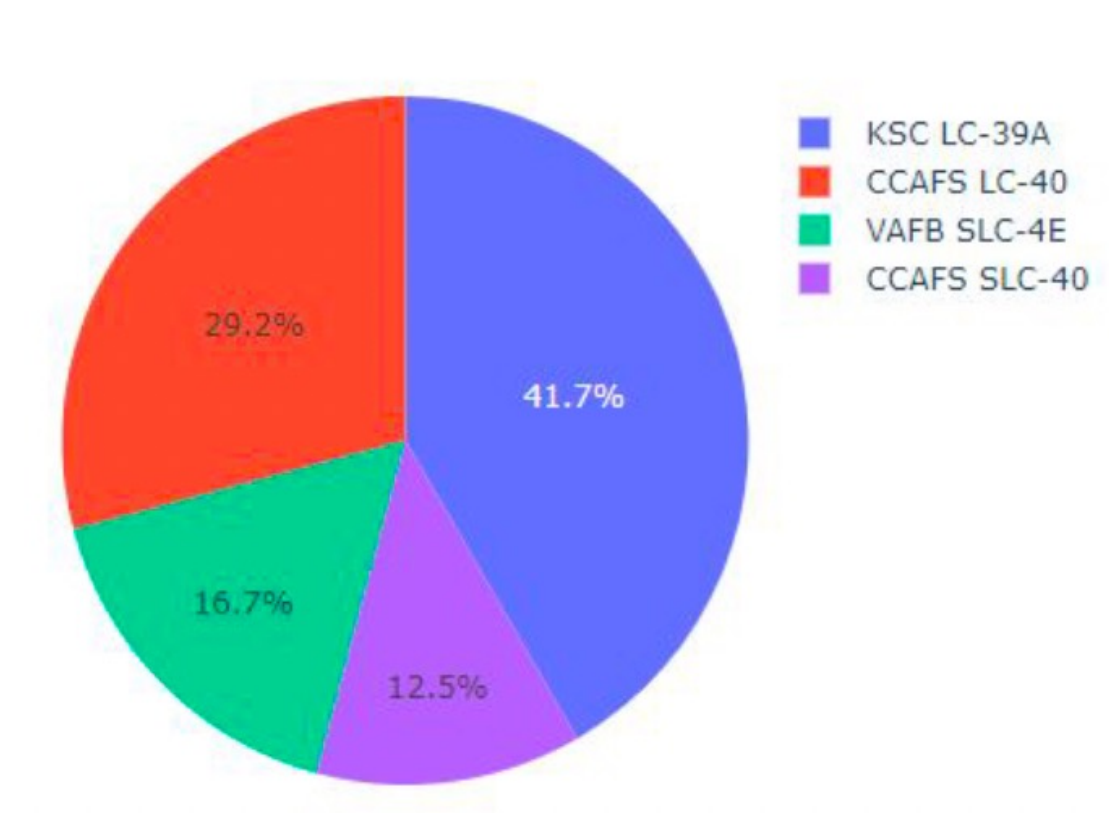
Areas surrounding the Launch Sites



- The launch sites are strictly on the coast, away from cities with an acceptable amount of land around it purposefully established for this..


Section 4
Build a Dashboard with Plotly Dash

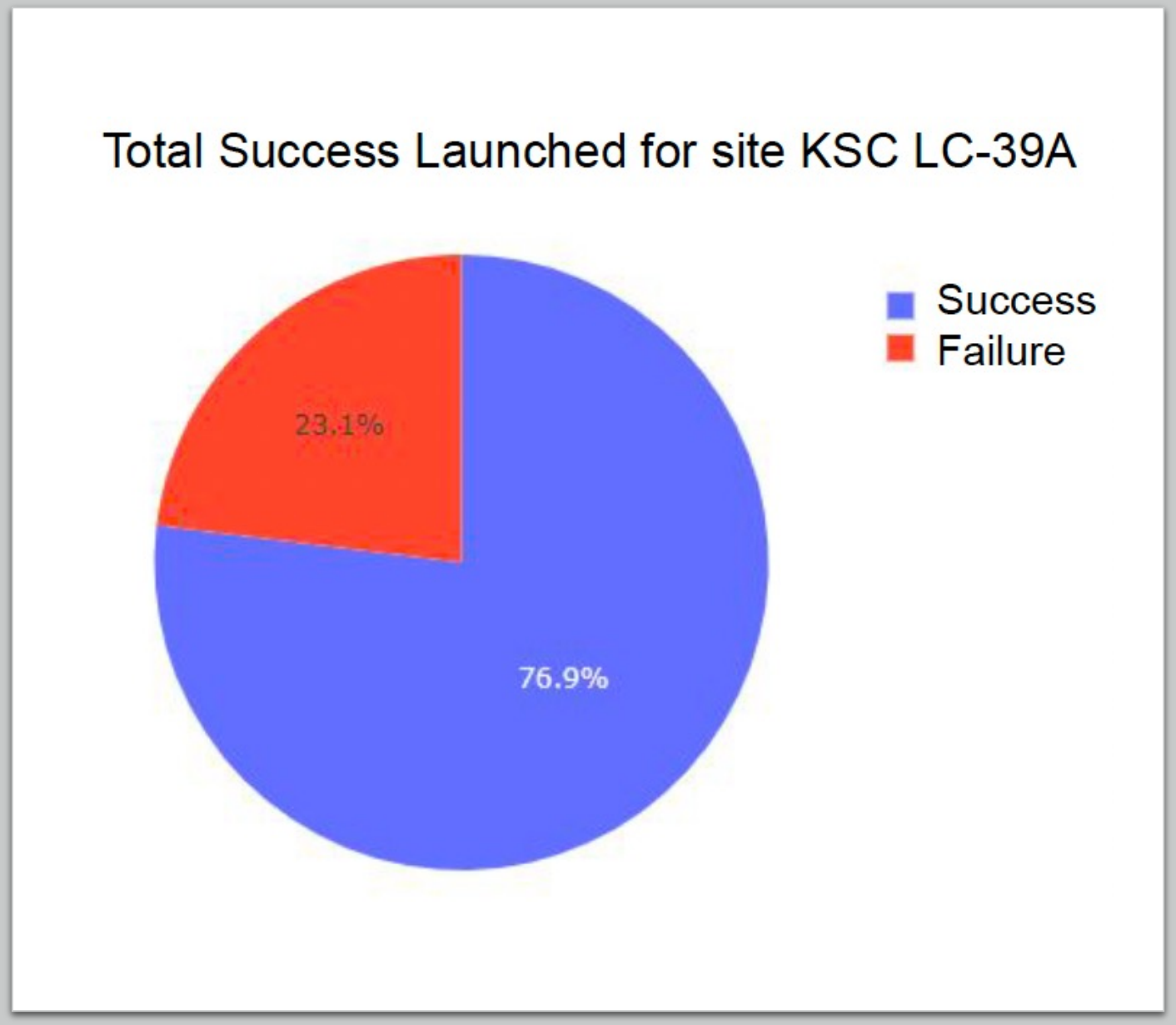
Percentage of Launches by sites



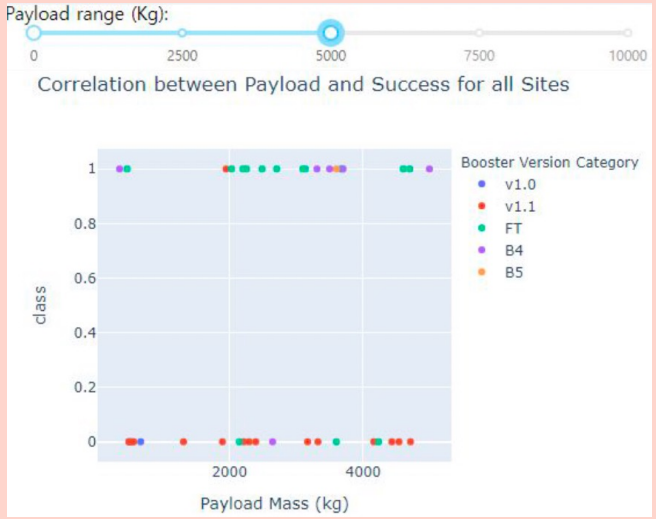
- The most successful site is KSC LC-39A while VAFB SLC-4E is the least successful, though there aren't as many launches compared to the rest which are in Space Coast, FL.

KSC LC-39A Outcome Ratio

- This site has a total of 13 launches with 10 being successful and 3 being failures
- A 10:3 ratio.
- The pie chart is color coordinated with red being failures and blue being successes.



Launch vs Payload Outcome Scatter Plot



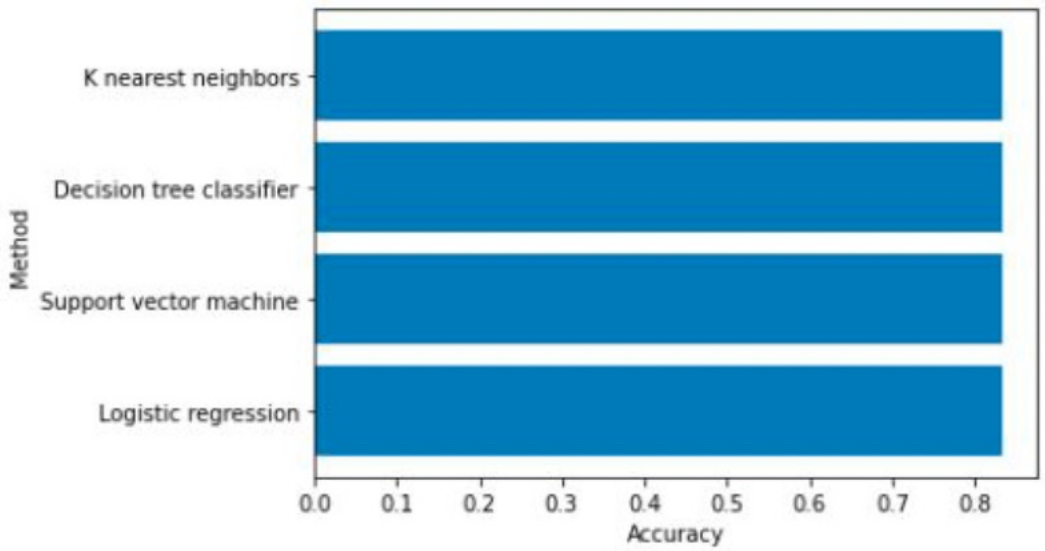
- This shows the success rate for the lighter weighted payloads and the heavier payloads.
- Shown here the lower weighted payloads are much more successful compared to the heavier payloads.

Section 5

Predictive Analysis (Classification)

Classification Accuracy

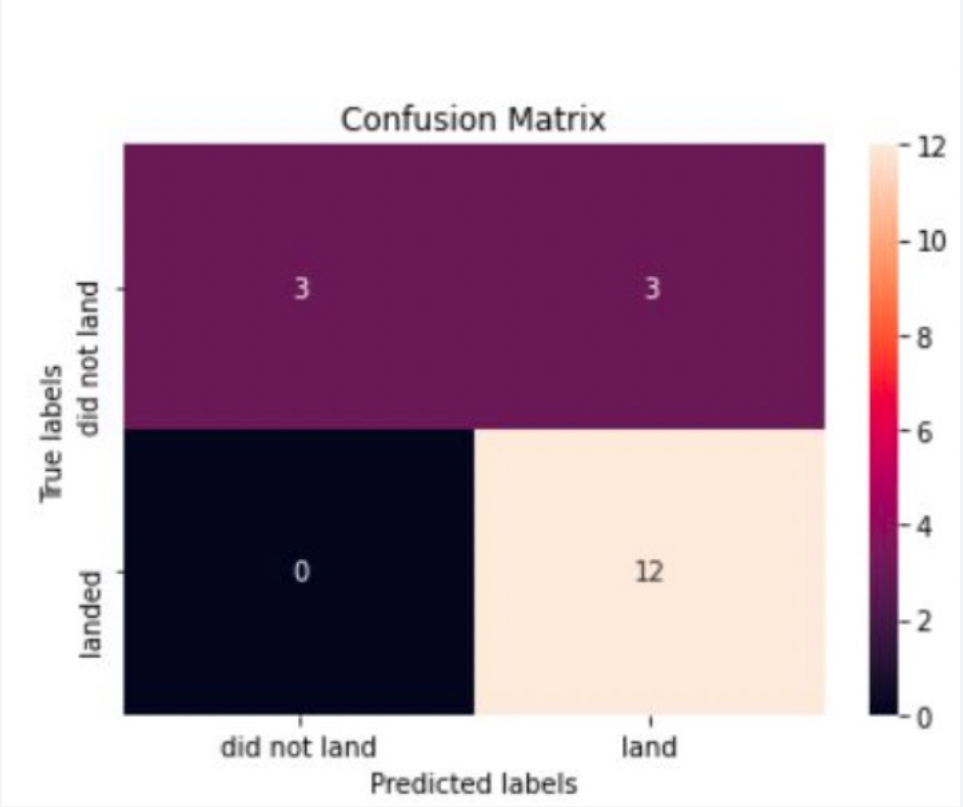
- The test set shows the accuracy is roughly the same with 0.833333 which is 83.3333%. Though more data is needed as the test size wasn't large enough to truly get a sample of the set.



	Method	Accuracy
0	Logistic regression	0.833333
1	Support vector machine	0.833333
2	Decision tree classifier	0.833333
3	K nearest neighbors	0.833333

Confusion Matrix

- This model shows the prediction of 12 successful landings with 3 failed landings and 3 false positives were reported.



Conclusions

- The Orbital types HEO, GEO, ES-L1 and SSO have the highest success rate being flawless thus far.
- KSC LC-39 is the site with the most successful launch site.
- Those payloads with the lower weight are more successful compared to the highest weight.
- When the number of flights increase the success rate also increases,

Appendix

- https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/labs/module_4/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb
- https://labs.cognitiveclass.ai/v2/tools/jupyterlite?ulid=ulid-9c5bf3da6f5875a7ffd486a5fc20554afdd15c90
- https://labs.cognitiveclass.ai/v2/tools/cloud-ide-kubernetes?ulid=ulid-86c283a8ce5e0f718bd45fee78022802a4ed2b95
- https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/labs/module_2/jupyter-labs-eda-sql-coursera.ipynb
- https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/labs/module_1_L3/labs-jupyter-spacex-Data%20wrangling.ipynb
- https://labs.cognitiveclass.ai/v2/tools/jupyterlite?ulid=ulid-ef9fb2b72b064bb54e71608e8221f0d1620788fe

Thank you!