Winning Space Race with Data Science

Vallabhapurapu L Sai Ruthwik

09-02-2022

Outline

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

Executive Summary

- Summary of methodologies
  - Data collection
  - Data visualization
  - Interactive visualizations
  - Predictive analysis using 4 models

- Summary of all results
  - EDA outcomes
  - Prediction accuracy

Introduction

- Project background and context

  We can now predict if a falcon rocket first stage land back successfully or not. This information helps a lot when it comes to building new rocket.

- Problems you want to find answers

  Parameters that are affecting the landing of first stage of rocket.

  Suitable conditions that a rocket needs to meet to have a successful landing.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - SpaceX Rest API
  - Web scraping on Wikipedia (link)
- Perform data wrangling
  - Based on the mission outcomes a new column is created and labeled as 1 for successful outcome and 0 for an unsuccessful outcome.
- 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 – SpaceX API

- https://github.com/vlsruthwik/Data-Science-Capstone-Project/blob/master/Data_collection_using_API.ipynb

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

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

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

	static_fire_date_utc	static_fire_date_unix	net	window	rocket	success	failures	details	crew	ships	capsules	
0	2006-03-17T00:00:00.000Z	1.142554e+09	False	0.0	5e9d0d95eda69955f709d1eb	False	[{'time': 33, 'altitude': None, 'reason': 'merlin engine failure'}]	Engine failure at 33 seconds and loss of vehicle	[]	[]	[]	[5eb
1	None	NaN	False	0.0	5e9d0d95eda69955f709d1eb	False	[{'time': 301, 'altitude': 289, 'reason': 'harmonic oscillation leading to premature engine shutdown'}]	Successful first stage burn and transition to second stage, maximum altitude 289 km, Premature engine shutdown at T+7 min 30 s, Failed to reach orbit	[]	[]	[]	[5eb

Data Collection - Scraping

- https://github.com/vlsruthwik/Data-Science-Capstone-Project/blob/master/Data_collection_using_webscraping.ipynb

```
static_url=https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922
source=requests.get(static_url)
soup=BeautifulSoup(source.text)
```

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

	Flight No.	Launch site	Payload	Payload mass	Orbit	Customer	Launch outcome	NaN	Versio Booste
0	1 None	CCAFS	Dragon Spacecraft Qualification Unit	Dragon Spacecraft Qualification Unit	LEO	SpaceX	Success\n	None	F9 v1.0B0003.
1	2 None	CCAFS	Dragon	Dragon	LEO	NASA	Success	None	F9 v1.0B0004.
2	3 None	CCAFS	Dragon	Dragon	LEO	NASA	Success	None	F9 v1.0B0005.
3	4 None	CCAFS	SpaceX CRS-1	SpaceX CRS-1	LEO	NASA	Success\n	None	F9 v1.0B0006.
4	5 None	CCAFS	SpaceX CRS-2	SpaceX CRS-2	LEO	NASA	Success\n	None	F9 v1.0B0007.

```
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']=[]
```

Data Wrangling

- There are many outcomes of landing. Each case is mapped as 1 for successful landings and 0 for failed landings
  - True Ocean means the mission outcome was successfully landed to a specific region of the ocean while False Ocean means the mission outcome was unsuccessfully landed to a specific region of the ocean.
  - True RTLS means the mission outcome was successfully landed to a ground pad False RTLS means the mission outcome was unsuccessfully landed to a ground pad.
  - True ASDS means the mission outcome was successfully landed on a drone ship False ASDS means the mission outcome was unsuccessfully landed on a drone ship.
- https://github.com/vlsruthwik/Data-Science-Capstone-Project/blob/master/EDA.ipynb

```
# landing_outcomes = values on Outcome column
landing_outcomes = df['Outcome'].value_counts()
landing_outcomes
```

```
True ASDS      41
None None      19
True RTLS      14
False ASDS      6
True Ocean      5
None ASDS       2
False Ocean     2
False RTLS      1
Name: Outcome, dtype: int64
```

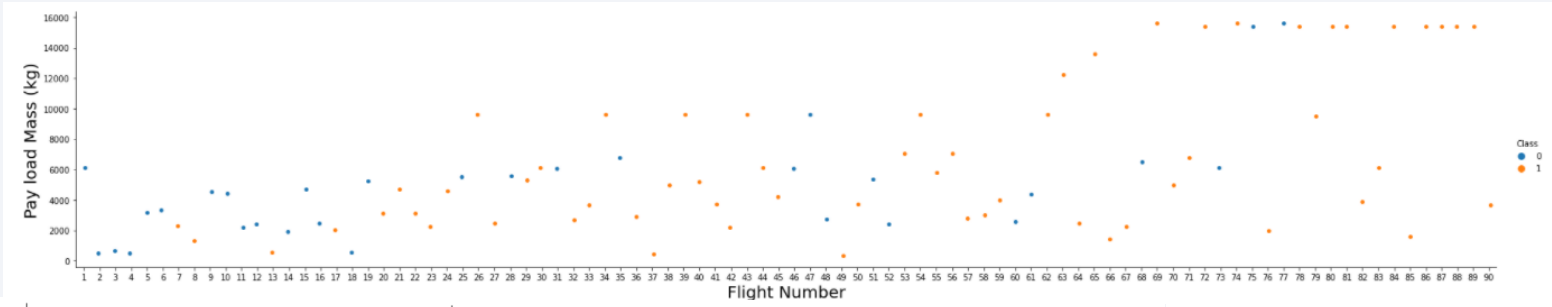
```
df['Class']=landing_class
df[['Class']].head(8)
```

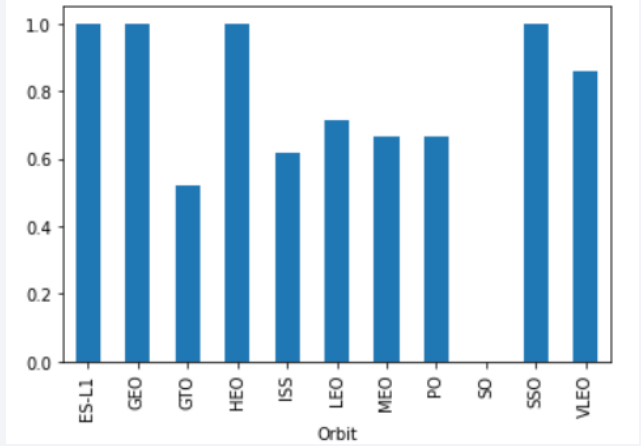
	Class
0	0
1	0
2	0
3	0
4	0
5	0
6	1
7	1

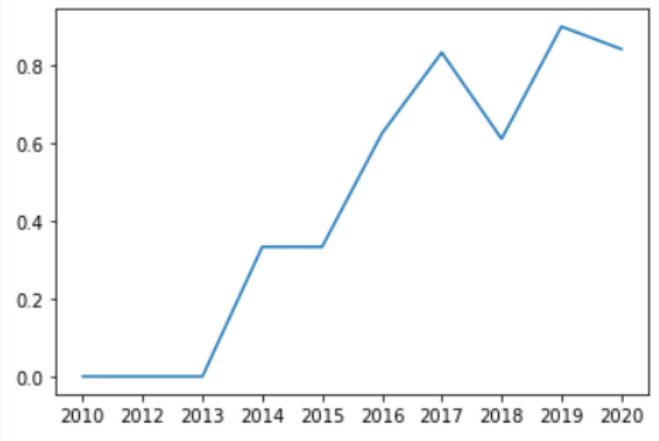
EDA with Data Visualization

- Scatter plots
  - Flight Number vs Payload Mass
  - Flight Number vs Launch Site
  - Payload Mass vs Launch Site
  - Flight Number vs Orbit
  - Payload Mass vs Orbit
- Bar graph
  - Orbit
- Line plot
  - Year vs Success rate

- https://github.com/vlsruthwik/Data-Science-Capstone-Project/blob/master/EDA_using_visualization.ipynb







EDA with SQL

- Using bullet point format, summarize the SQL queries you performed
  - 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 acheived.
  - 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

- https://github.com/vlsruthwik/Data-Science-Capstone-Project/blob/master/EDA_using_SQL.ipynb

Build an Interactive Map with Folium

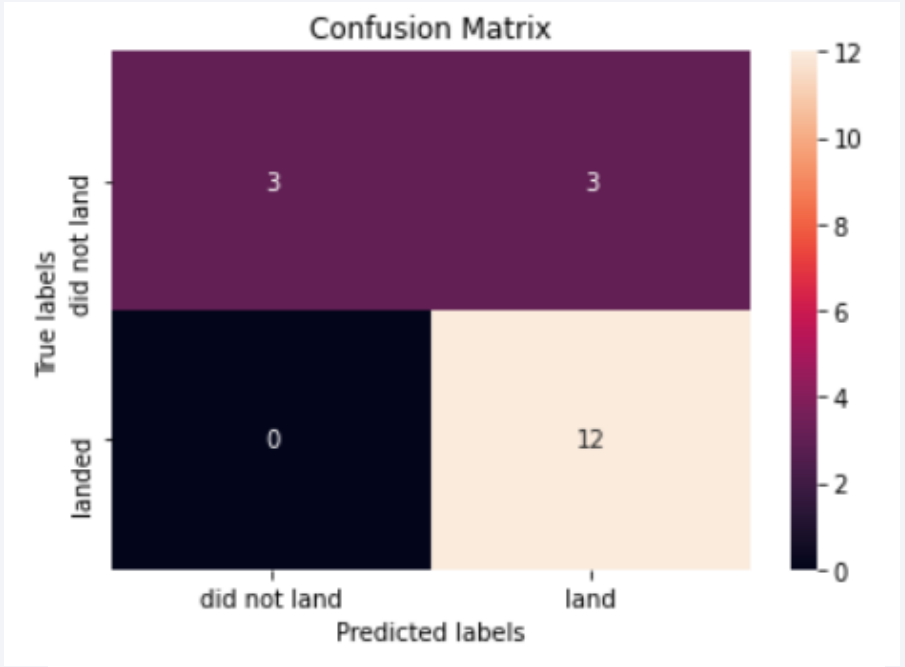
- Objects
  - Circles are created to mark launch sites using their coordinates.
  - No of successful and unsuccessful launches are marked for each launch site.
  - Lines are draw from launch sites to nearby railway, highway, coastline and city
- Using those objects, we found that
  - launch sites are not in close proximity to railways
  - launch sites are not in close proximity to highways
  - launch sites are in close proximity to coastline
  - launch sites keep certain distance away from cities
- https://github.com/vlsruthwik/Data-Science-Capstone-Project/blob/master/Interactive_visual_analysis_using_folium.ipynb

Build a Dashboard with Plotly Dash

- Graphs
  - Pie chart which shows success rate between launch sites and for individual launch site as well.
  - Scatter plot between success lands and Payload Mass.
- Pie chart useful to get relative proportions of multiple classes.
- Scatter plot is useful to show relation between two variables.
- https://github.com/vlsruthwik/Data-Science-Capstone-Project/blob/master/Interactive_visualization.py

Predictive Analysis (Classification)

- Model Building
  - Loading dataset as Pandas DataFrame
  - Transforming data
  - Splitting dataset into train set and test set
  - Training as model using GridSearchCV based on logistic regression, SVM and classification trees on train set.
  - Find the best parameters based on evaluation.
- Evaluating Model
  - Checking accuracy of each model
  - Plotting confusion matrix
- https://github.com/vlsruthwik/Data-Science-Capstone-Project/blob/master/Predictions.ipynb

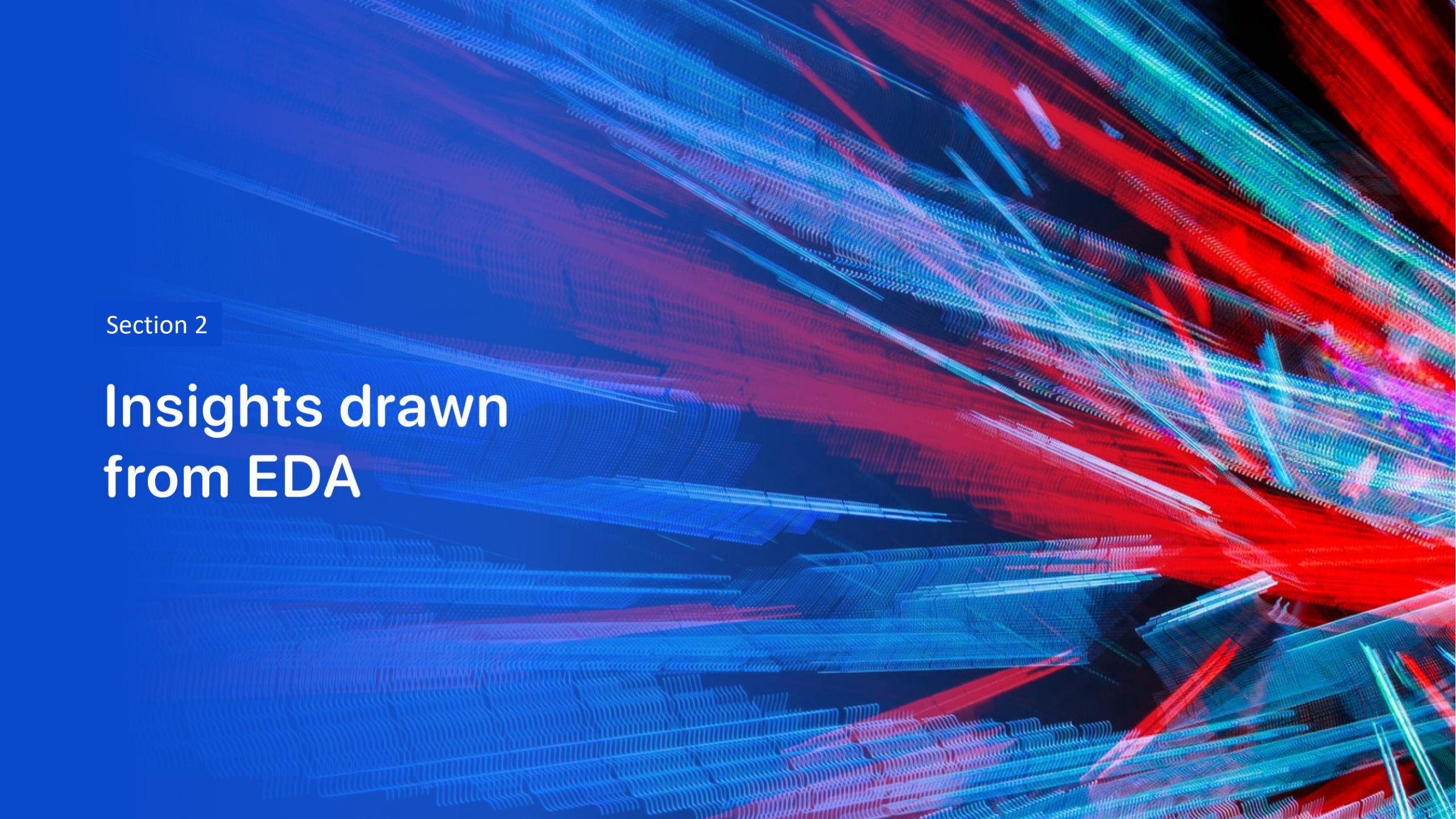



```
algorithms = {'KNN':knn_cv.best_score_,'Tree':tree_cv.b
bestalgorithm = max(algorithms, key=algorithms.get)
print('Best Algorithm is',bestalgorithm,'with a score o
if bestalgorithm == 'Tree':
    print('Best Params is :',tree_cv.best_params_)
if bestalgorithm == 'KNN':
    print('Best Params is :',knn_cv.best_params_)
if bestalgorithm == 'LogisticRegression':
    print('Best Params is :',logreg_cv.best_params_)
```

```
Best Algorithm is Tree with a score of 0.875
Best Params is : {'criterion': 'gini', 'max_depth': 4, '
```

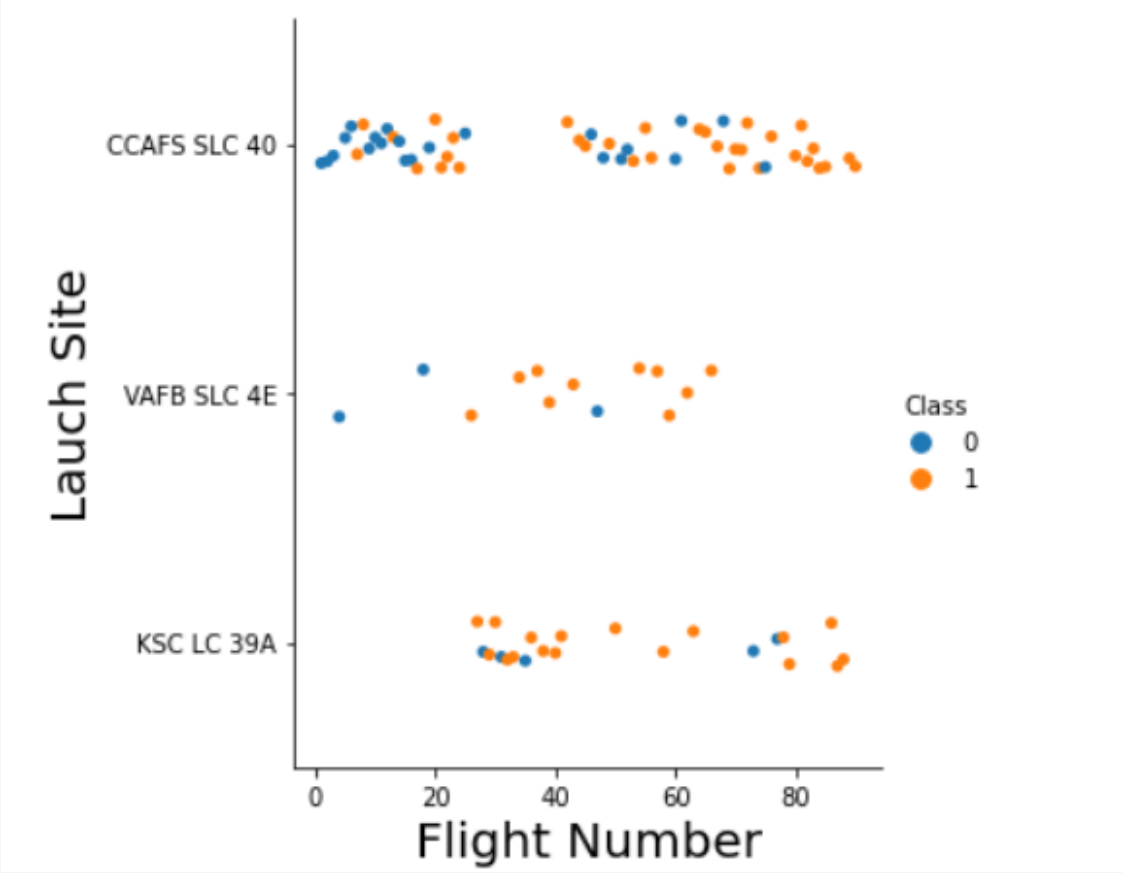
Results

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


Section 2
Insights drawn from EDA

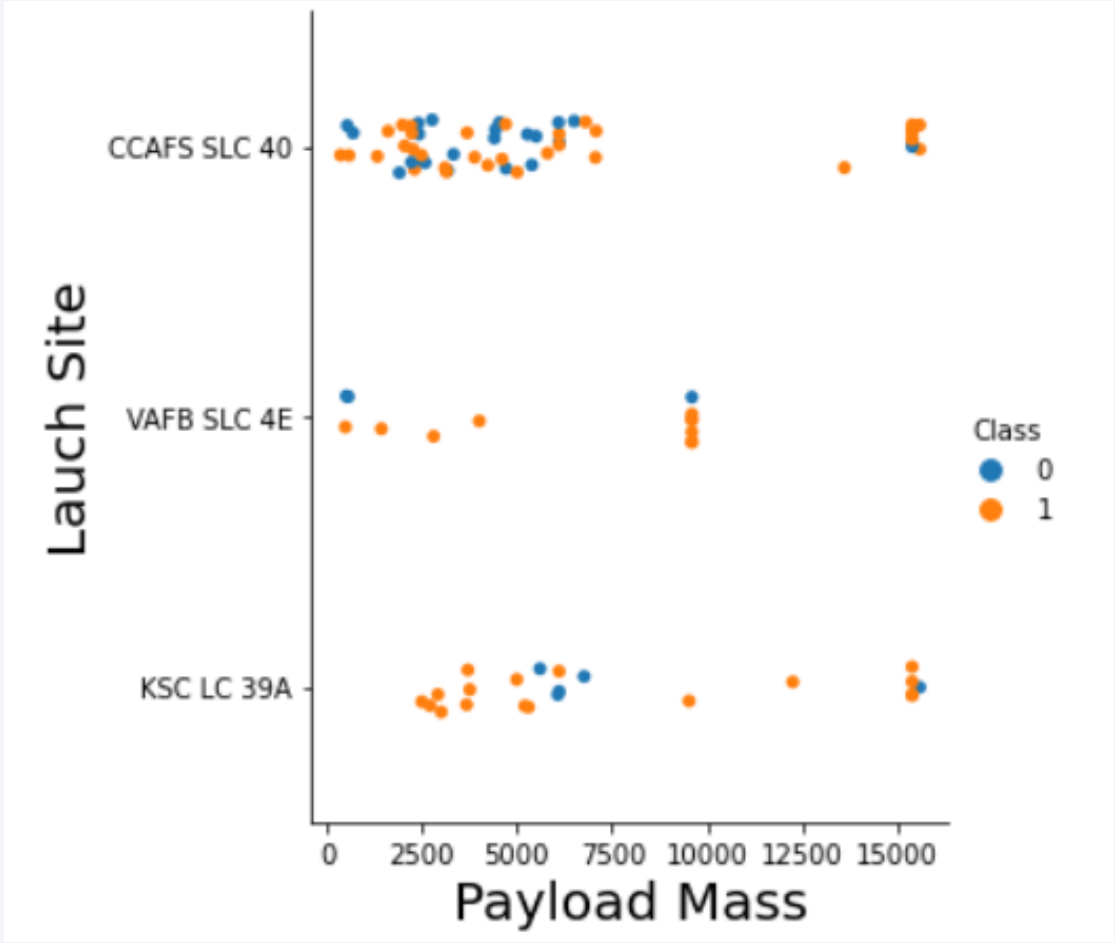
Flight Number vs. Launch Site

- You can observe that more the Flight number, the success rate is increasing for each site.



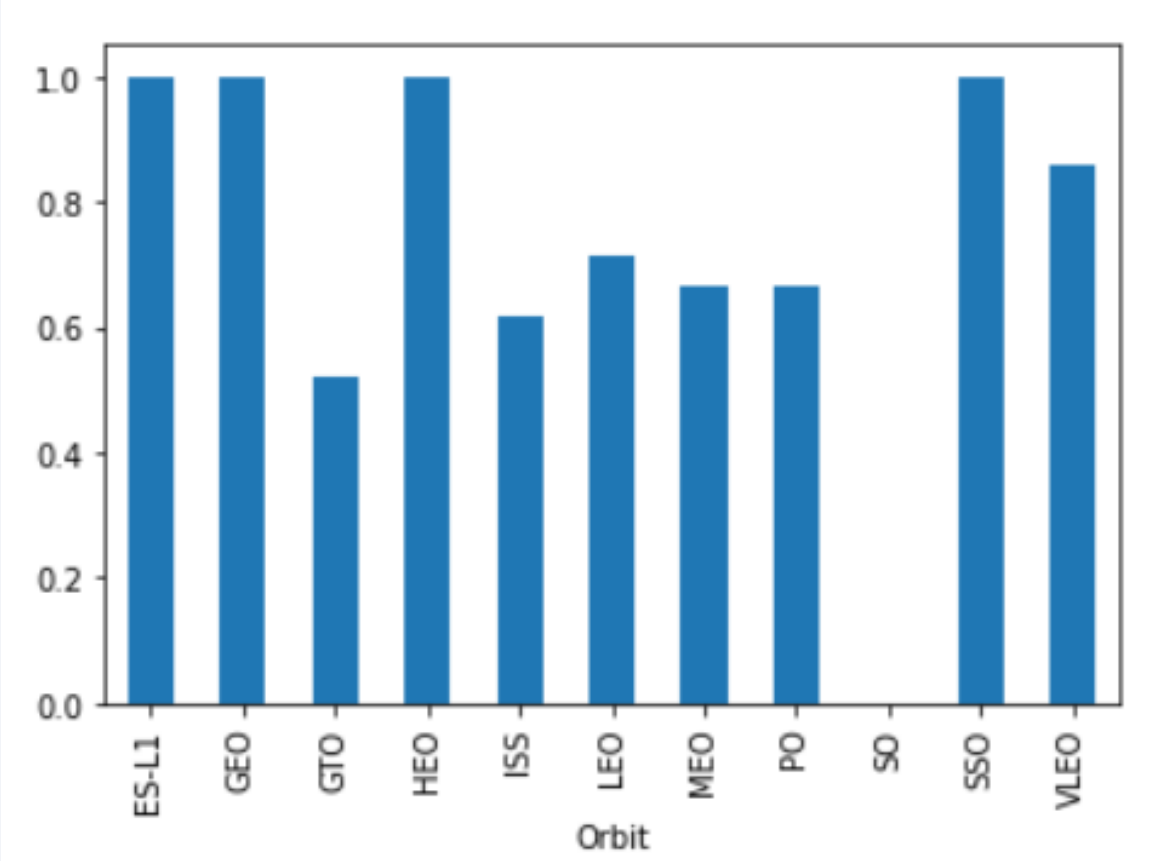
Payload vs. Launch Site

- the VAFB-SLC launch site there are no rockets launched for heavy payload mass(greater than 10000).



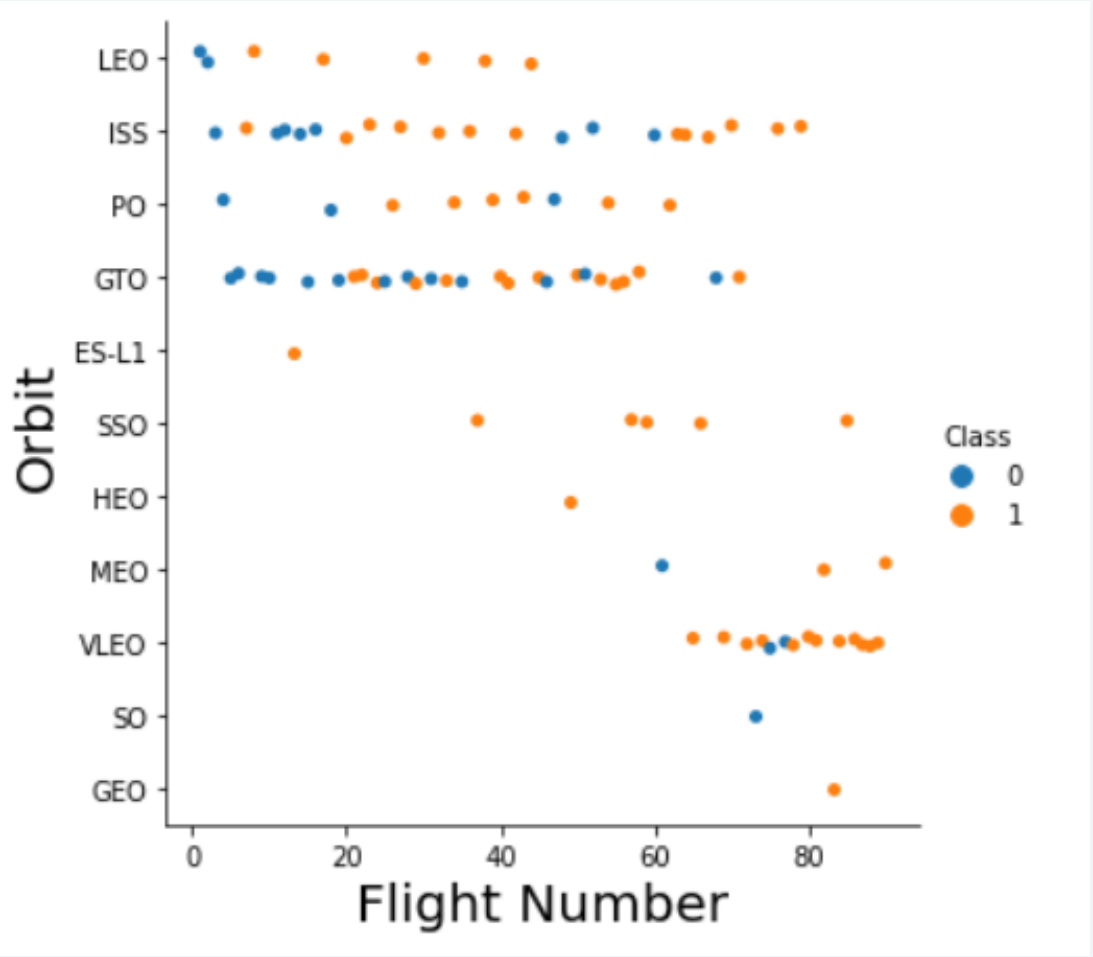
Success Rate vs. Orbit Type

- Orbit GEO,HEO,SSO,ES-L1 has the best and equal Success rate



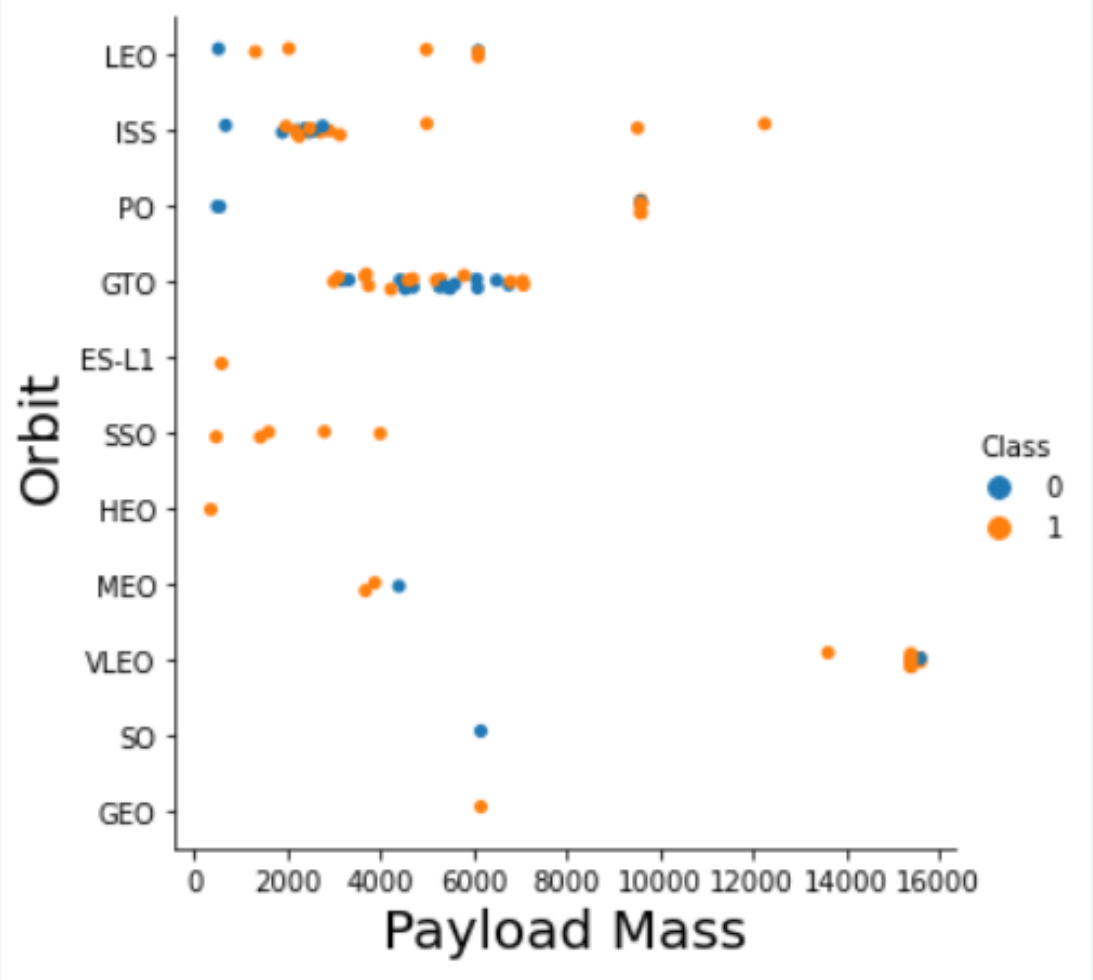
Flight Number vs. Orbit Type

- You should see that in the LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.



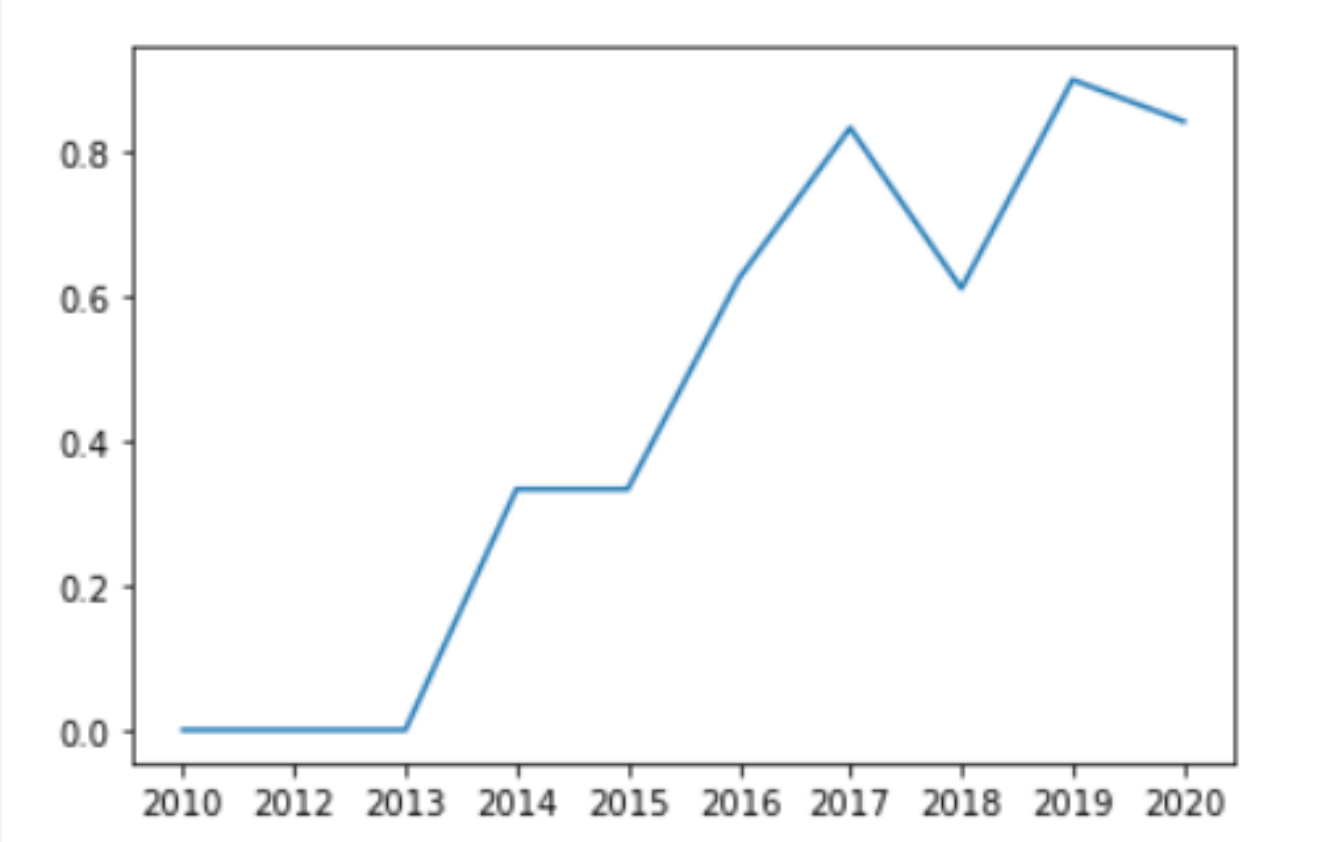
Payload vs. Orbit Type

- With heavy payloads the successful landing or positive landing rate are more for Polar, LEO and ISS.
- However, for GTO we cannot distinguish this well as both positive landing rate and unsuccessful landings both are there here.



Launch Success Yearly Trend

- You can observe that the success rate since 2013 kept increasing till 2020



All Launch Site Names

```
SELECT DISTINCT(LAUNCH_SITE) FROM SPACEXDATASET;
```

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

- Unique values of LAUNCH_SITE column from SPACEXDATASET table

Launch Site Names Begin with 'CCA'

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

```
SELECT * FROM SPACEXDATASET
WHERE LAUNCH_SITE LIKE 'CCA%'
LIMIT 5;
```

- 5 of all columns where launch site starts with CCA in SPACEXDATASET table

Total Payload Mass

```
SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXDATASET
WHERE CUSTOMER='NASA (CRS)';
```

1
45596

- Sum of Payload Mass where the customer is NASA (CRS) of SPACEXDATASET table

Average Payload Mass by F9 v1.1

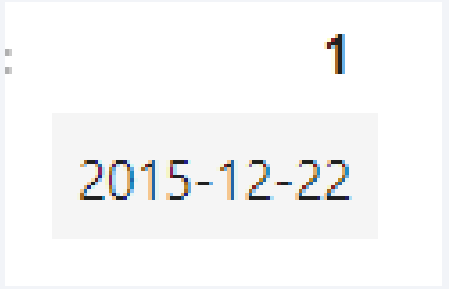
```
SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXDATASET
WHERE BOOSTER_VERSION LIKE '%F9 v1.1%';
```

1
2534

- Average of Payload Mass where the booster version is F9 v1.1 of SPACEXDATASET table

First Successful Ground Landing Date

```
SELECT MIN(DATE) FROM SPACEXDATASET
WHERE LANDING_OUTCOME='Success (ground pad)';
```



- MIN(DATE) give the first date of a set
- Minimum of date where landing outcome is successful on a ground pad of SPACEXDATASET table

Successful Drone Ship Landing with Payload between 4000 and 6000

booster_version	landing_outcome	payload_mass__kg_
F9 FT B1022	Success (drone ship)	4696
F9 FT B1026	Success (drone ship)	4600
F9 FT B1021.2	Success (drone ship)	5300
F9 FT B1031.2	Success (drone ship)	5200

```
SELECT BOOSTER_VERSION,LANDING_OUTCOME, PAYLOAD_MASS__KG_ FROM SPACEXDATASET

WHERE LANDING_OUTCOME='Success (drone ship)' AND (PAYLOAD_MASS__KG_>4000 AND
PAYLOAD_MASS__KG_<6000);
```

- Booster version, landing outcome, payload mass where landing outcome is a success drone ship and payload mass is between 4000 and 6000 of SPACEXDATASET table

Total Number of Successful and Failure Mission Outcomes

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

```
SELECT MISSION_OUTCOME,COUNT(MISSION_OUTCOME) AS COUNT
FROM SPACEXDATASET

GROUP BY MISSION_OUTCOME;
```

- Here we use AS to name new value(COUNT(MISSION_OUTCOME) named as COUNT)
- GROUP BY groups the columns based in MISSION_OUTCOME value

Boosters Carried Maximum Payload

```
SELECT DISTINCT(BOOSTER_VERSION) FROM SPACEXDATASET

WHERE PAYLOAD_MASS__KG_ = (SELECT
MAX(PAYLOAD_MASS__KG_) FROM SPACEXDATASET);
```

booster_version
F9 B5 B1048.4
F9 B5 B1048.5
F9 B5 B1049.4
F9 B5 B1049.5
F9 B5 B1049.7
F9 B5 B1051.3
F9 B5 B1051.4
F9 B5 B1051.6
F9 B5 B1056.4
F9 B5 B1058.3
F9 B5 B1060.2
F9 B5 B1060.3

- Here we use sub query to find maximum payload mass
- Unique values of booster version where the Payload mass is the highest(maximum of Payload mass of SPACEXDATASET table)

2015 Launch Records

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

```
SELECT LANDING_OUTCOME, BOOSTER_VERSION, LAUNCH_SITE FROM SPACEXDATASET

WHERE YEAR(DATE)='2015' AND LANDING_OUTCOME='Failure (drone ship)';
```

- YEAR(DATE) gives only the year of date object
- Landing outcome, booster version, launch site columns of SPACEXDATASET table where the year is 2015 and landing outcome is a failure of droneship

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

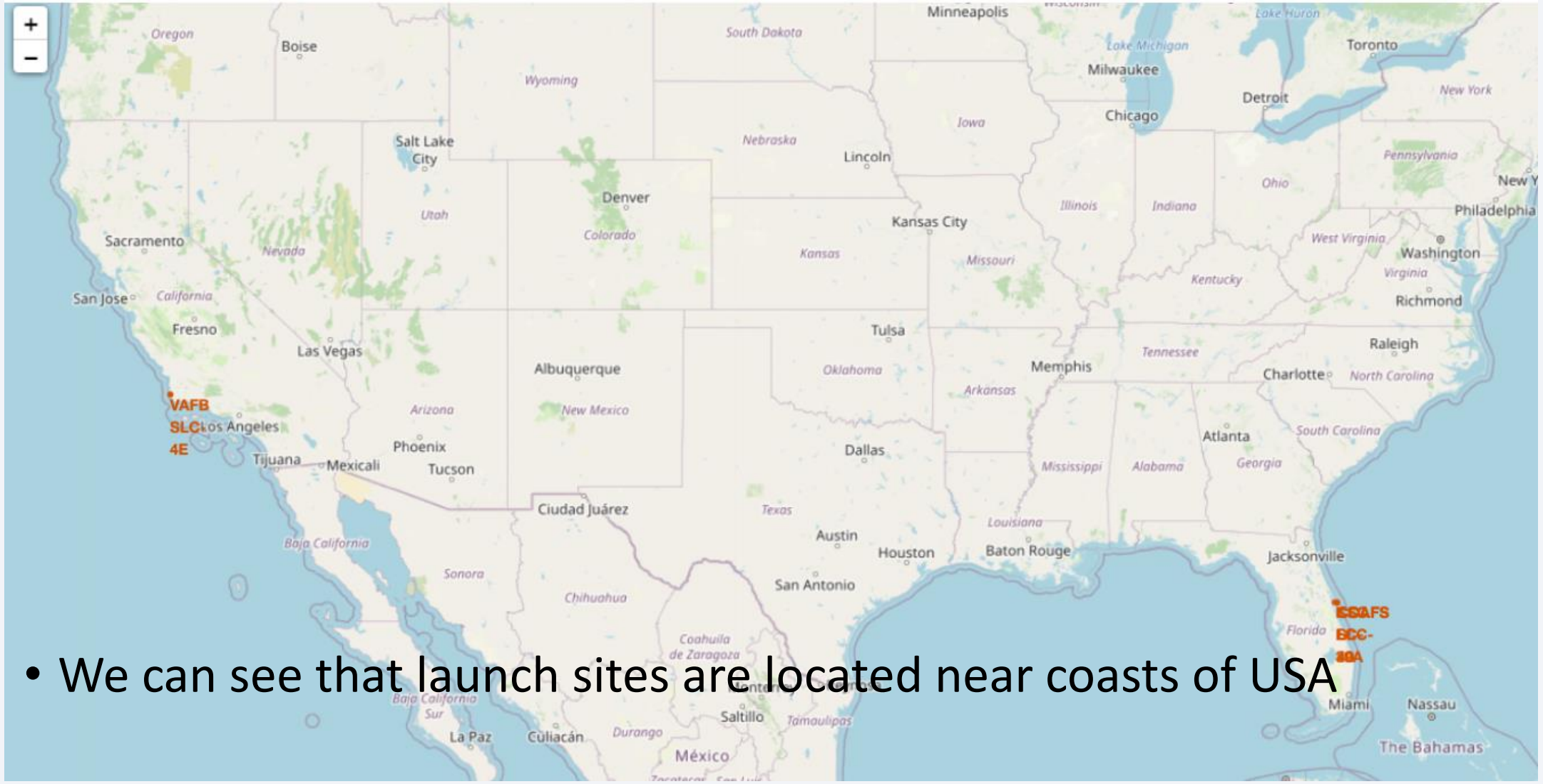
```
SELECT LANDING_OUTCOME,COUNT(LANDING_OUTCOME) AS
COUNT FROM SPACEXDATASET

WHERE DATE>='2010-06-04' AND DATE<='2017-03-20'

GROUP BY LANDING_OUTCOME

ORDER BY COUNT DESC;
```

landing_outcome	COUNT
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

- Landing outcome, COUNT of landing outcome of SPACEXDATASET table where the date range is 2010-06-04 to 2017-03-20 and the output is grouped by landing outcome and ordered in descending order of COUNT

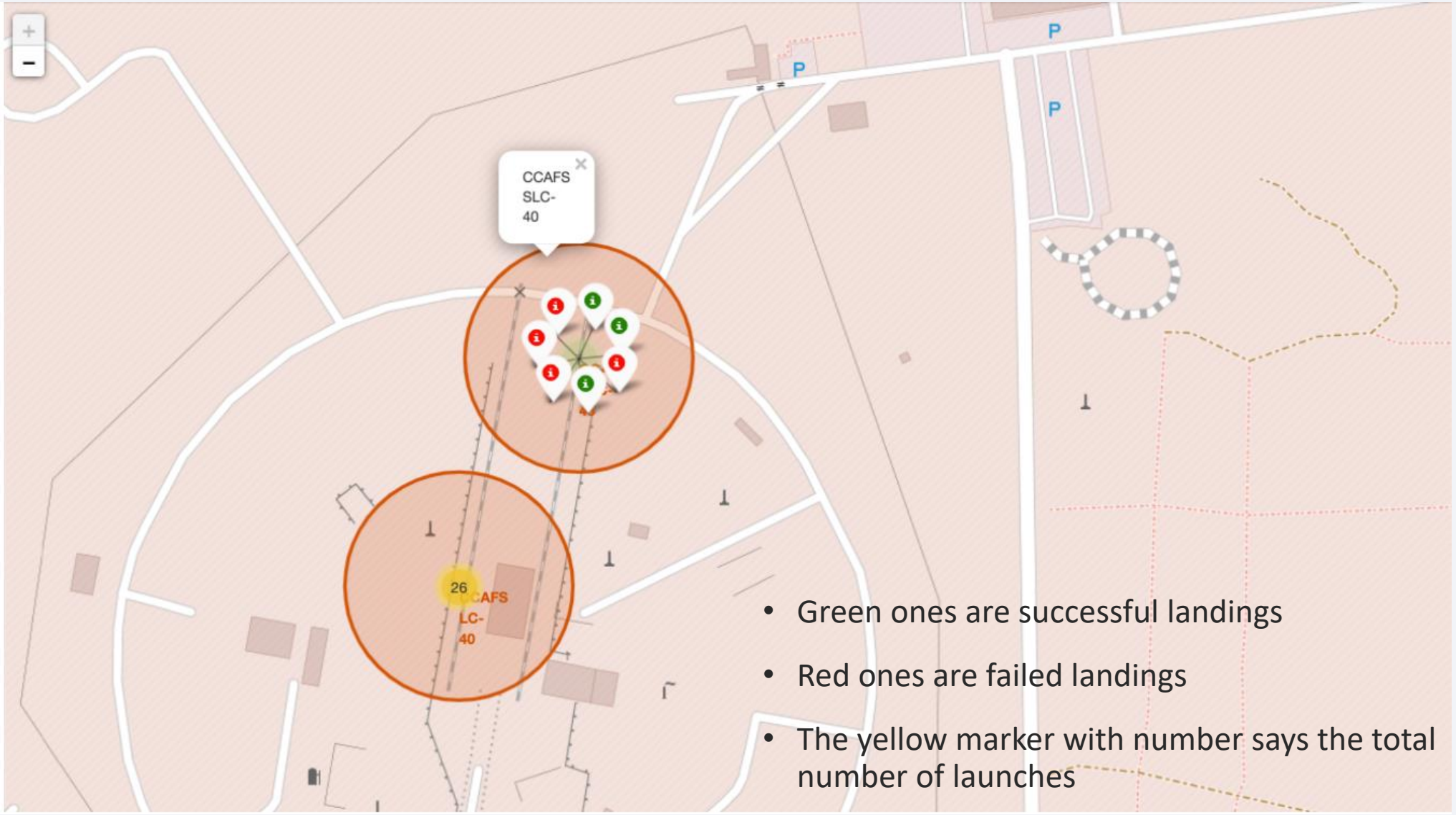

Section 3
Launch Sites
Proximities Analysis

Launch Sites on Global Map



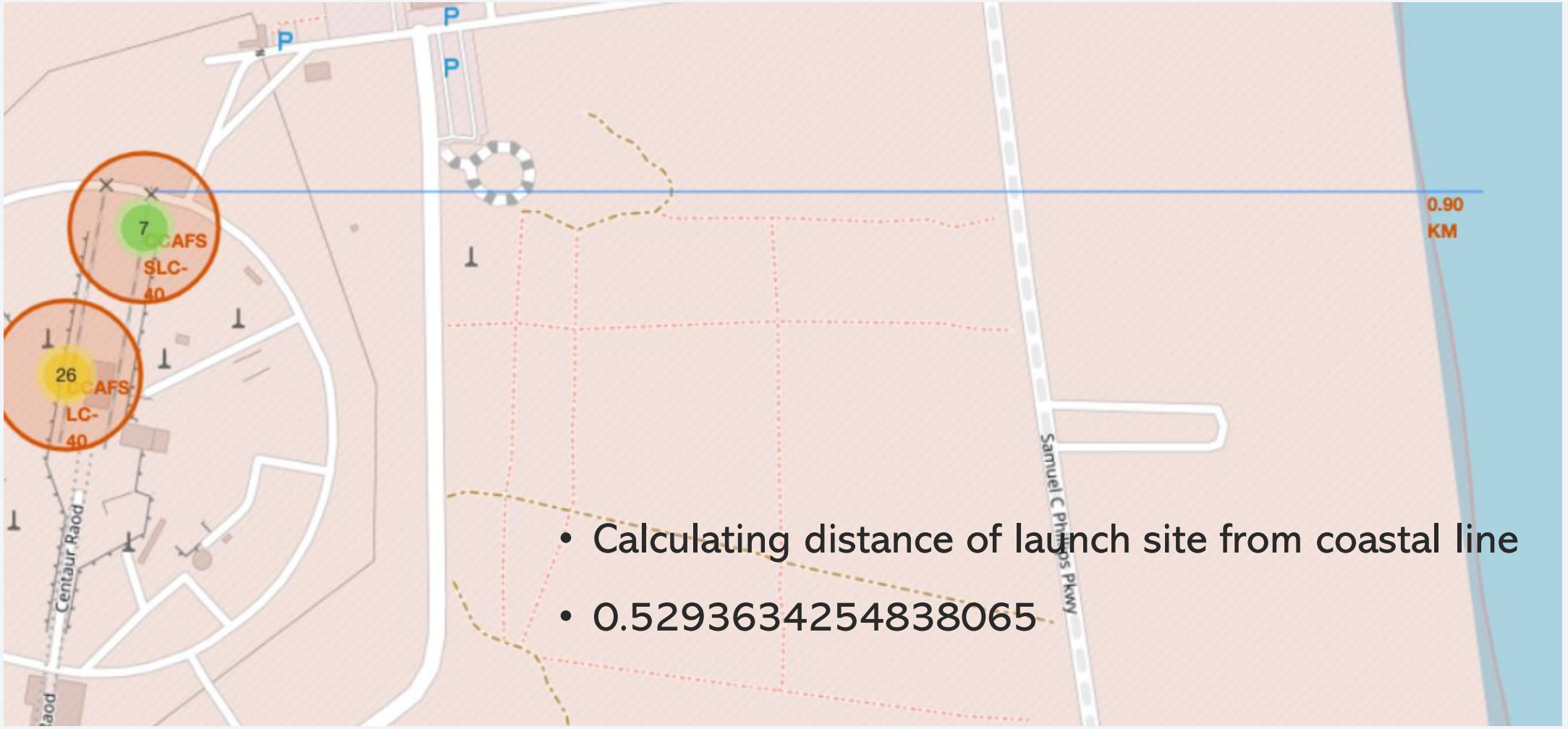
- We can see that launch sites are located near coasts of USA

Markers of landings for each launch site

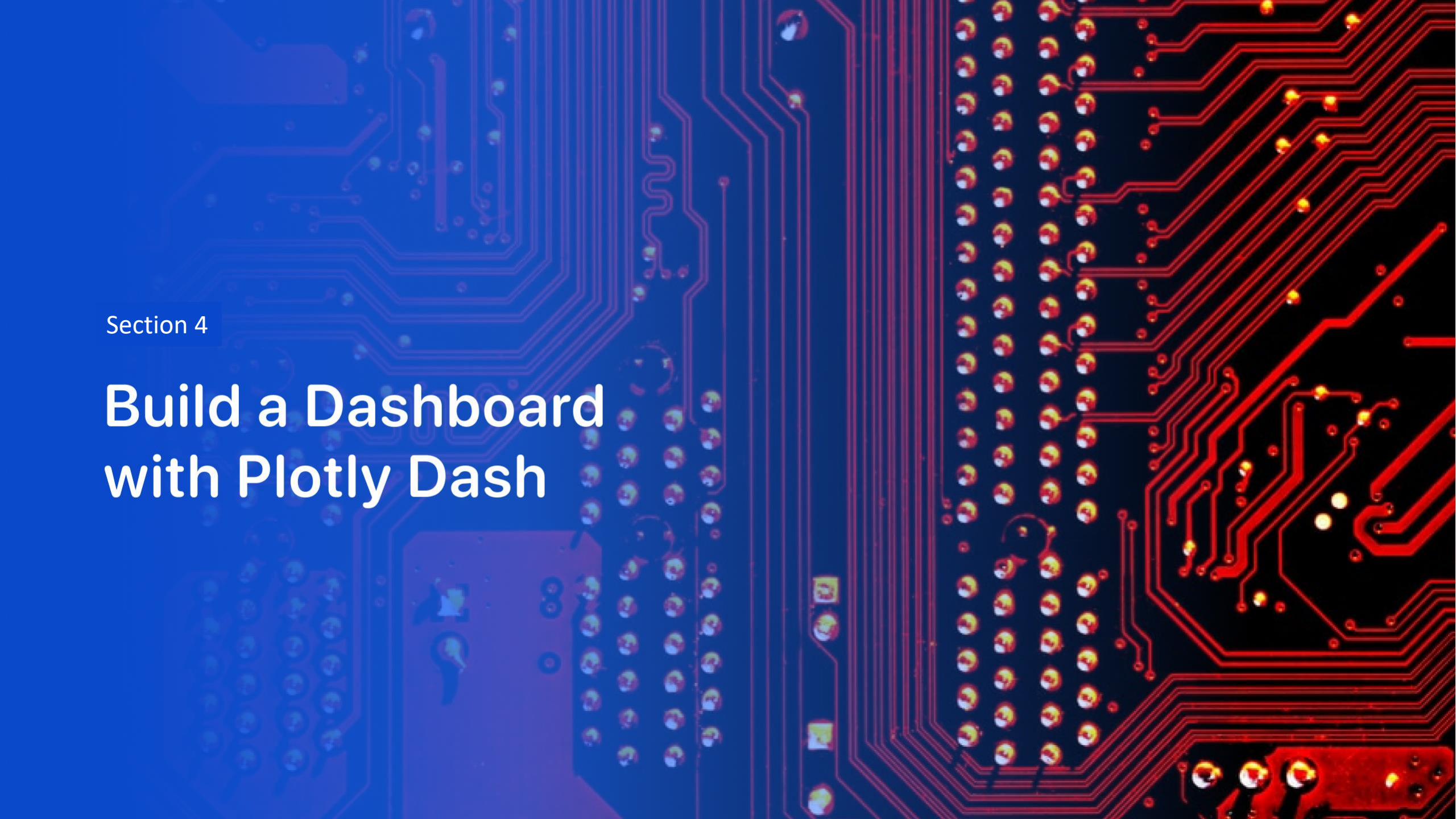


- Green ones are successful landings
- Red ones are failed landings
- The yellow marker with number says the total number of launches

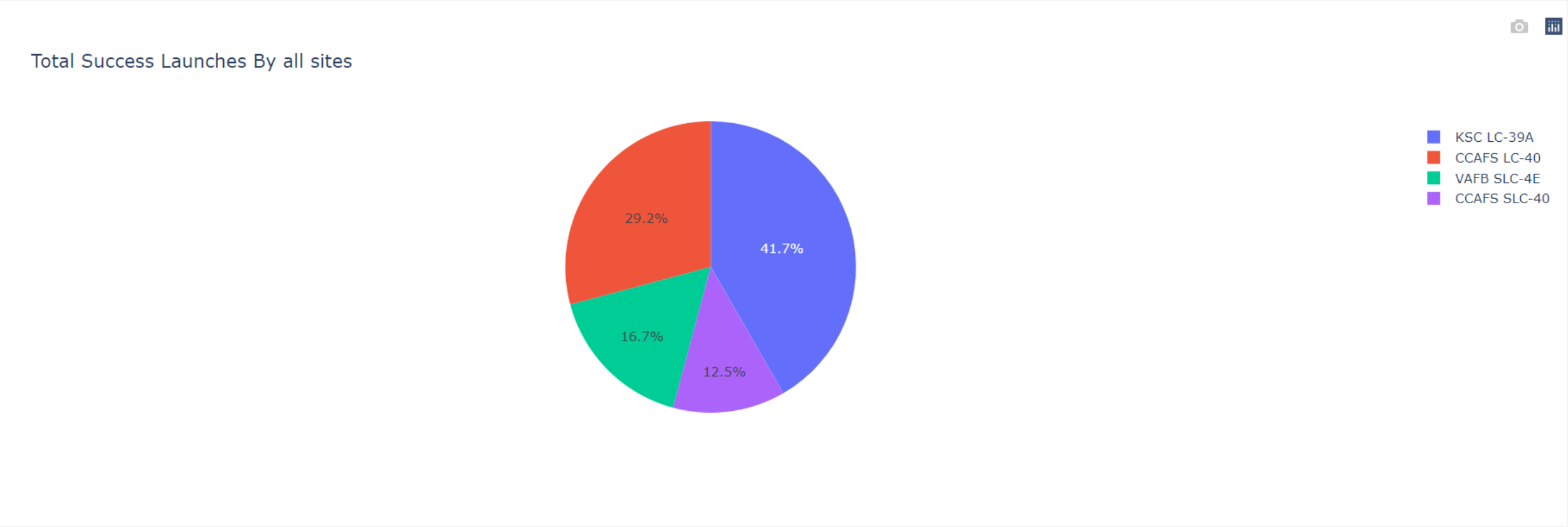
<Folium Map Screenshot 3>



- Calculating distance of launch site from coastal line
- 0.5293634254838065

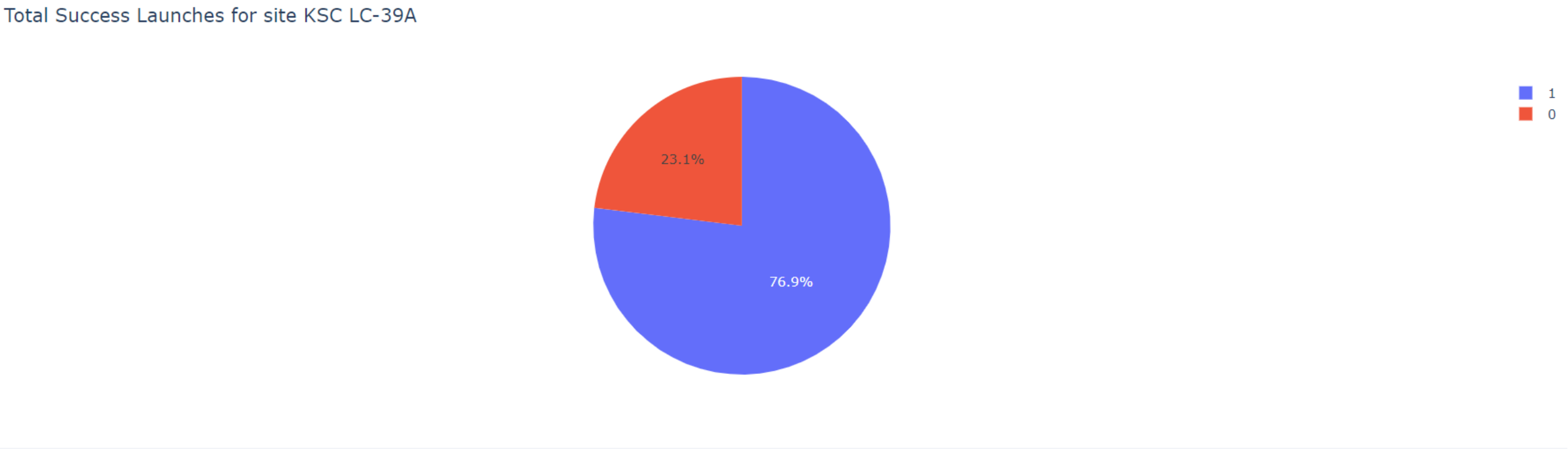

Section 4
Build a Dashboard with Plotly Dash

Success rate of landing by launch site proprtions



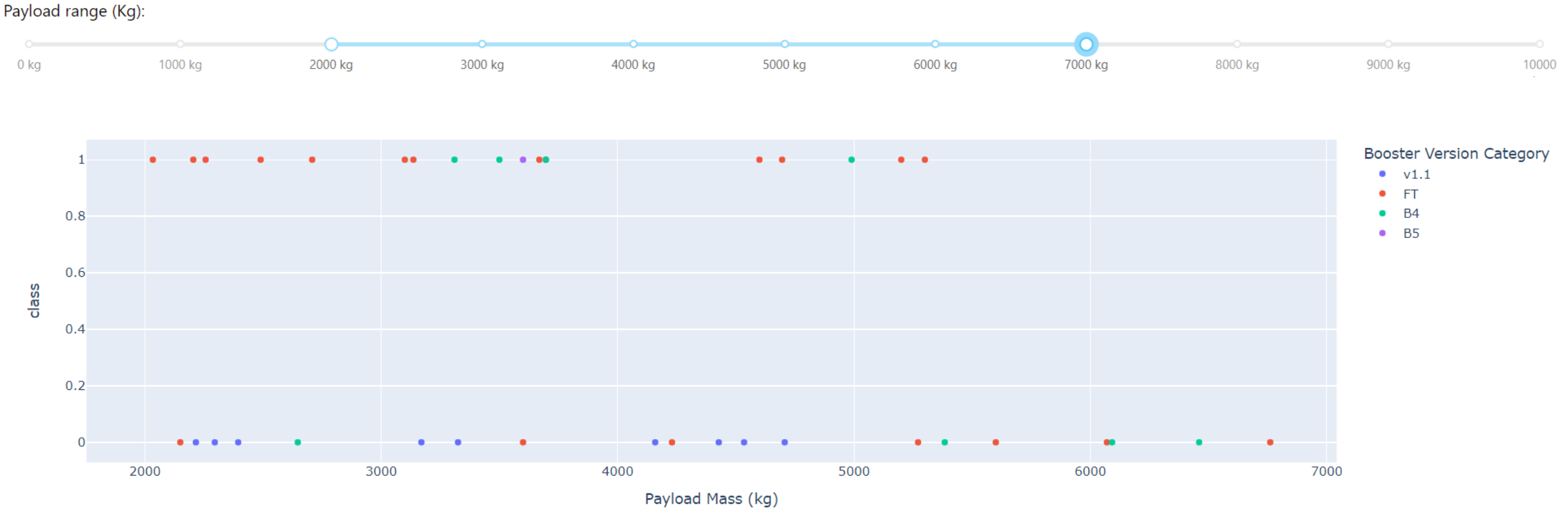
- We can see that KSC LC-39A launch site has highest success landing rate

Success vs Failed landings for each site



- For KSC LC-39A we can see that Success rate is 76.9%

Scatter plot between Payload Mass and Success rate



- We can get plot for any Payload mass range

Section 5

Predictive Analysis (Classification)

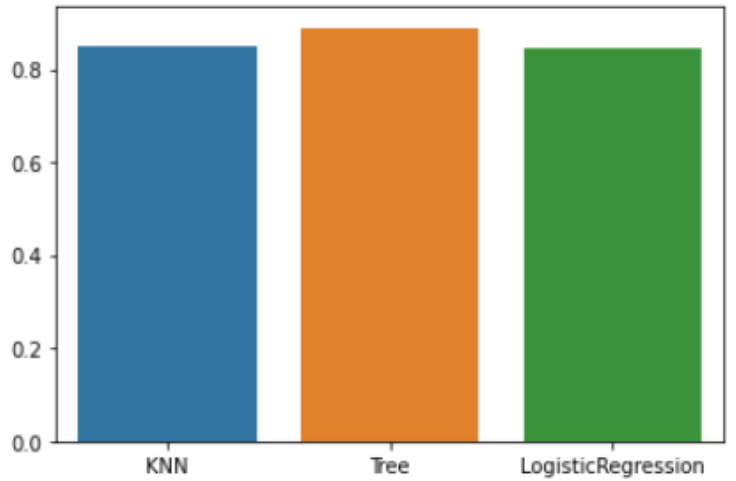
Classification Accuracy

```
Best Algorithm is Tree with a score of 0.8892857142857142
Best Params is : {'criterion': 'gini', 'max_depth': 14, 'ma
```

```
]: for i in algorithms:
       print(f'{i} : {algorithms[i]}')
```

```
KNN : 0.8482142857142858
Tree : 0.8892857142857142
LogisticRegression : 0.8464285714285713
```

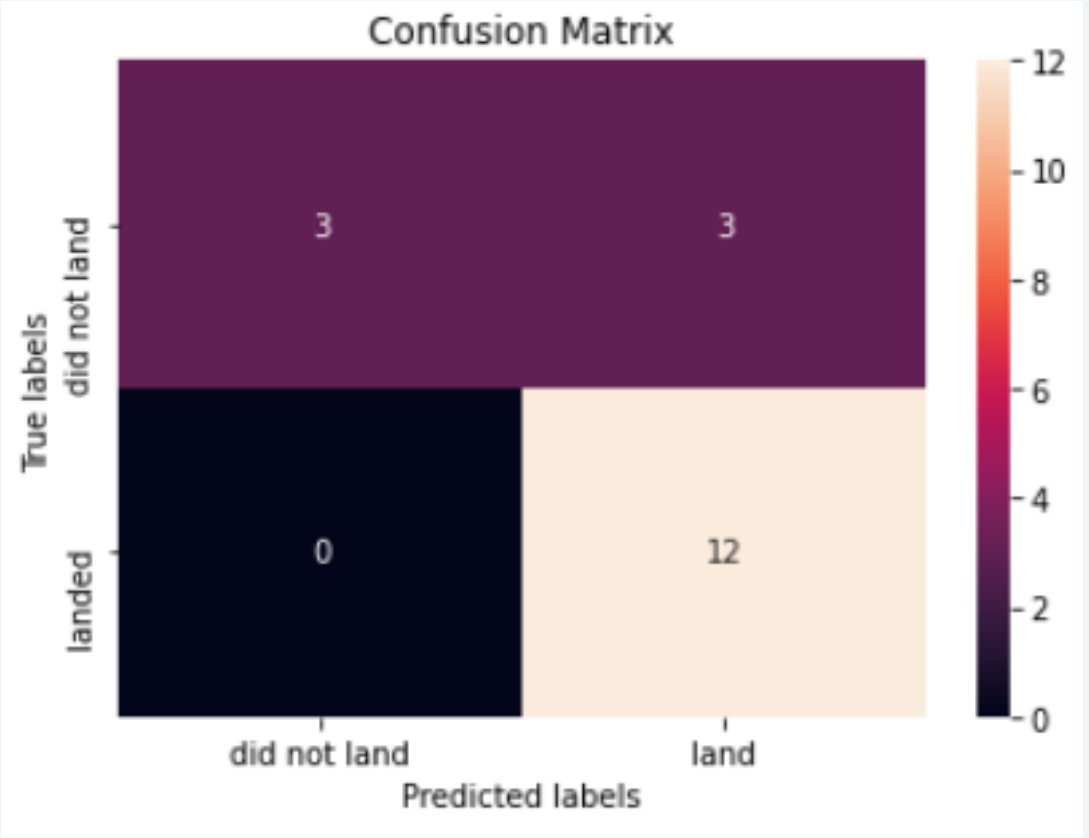
```
sns.barplot(data = bar_data,x='Model',y='Accuracy')
plt.show()
```



- After selecting the best hyperparameters for the decision tree classifier using the validation data, we achieved 88.93% accuracy on the test data.

Confusion Matrix

- We see that
  - 3 labels are as false positives.
  - The y axis labels are true labels and x axis labels are predicted outcomes.
  - Those labels which match on both side are correct predications and rest are not



Conclusions

- Low Payload Mass rockets have high success rate than high Payload Mass.
- As the time goes SpaceX success rate is increasing.
- KSC LC-39A had the most successful launches from all the sites
- GEO,HEO,SSO,ES-L1 has the best success rate
- Tree classification is best fit for this data with an accuracy of 88.93%

Appendix

- https://github.com/vlsruthwik/Data-Science-Capstone-Project

- The following repository have all the files and codes that are used in this project.

Thank you!