
IBM Developer
SKILLS NETWORK
Winning Space Race
with Data Science
EPHRON MARTIN
SPACEX

Outline

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

Executive Summary

- Summary of methodologies
  - ✓ Data Collection
  - ✓ Data Wrargling
  - ✓ EDA with data visualization
  - ✓ EDA with SQL
  - ✓ Building an interactive map with Folium
  - ✓ Building a Dashboard with Plotly Dash
  - ✓ Predictive analysis (Classification)
- Summary of all results
  - ❖ Exploratory data analysis results
  - ❖ Interactive analytics demo in screenshots
  - ❖ Predictive analysis results

Introduction

- **Project background and context**

We predicted if the Falcon9 first stage will land successfully. SpaceX advertises Falcon9 rocket launches on its website, showcasing with a cost of 62M dollars, where other providers cost around 165M dollars each. The great diffrence is due to the fact that, spaceX can reuse the first stage. If we could determine if the first stage will land, we can therefore determine the cost of a launch. This infomation can be used by any alternative company who wants to bid against SpaceX for rocket launch.

- **Problems which need to be solved**

- What influences if the rocket will land successfully?
- Will relationship with certain rocket variables impact in determining the success rate?
- What should spaceX focus on inorder to achieve best results to ensure success landing rate?

Section 1

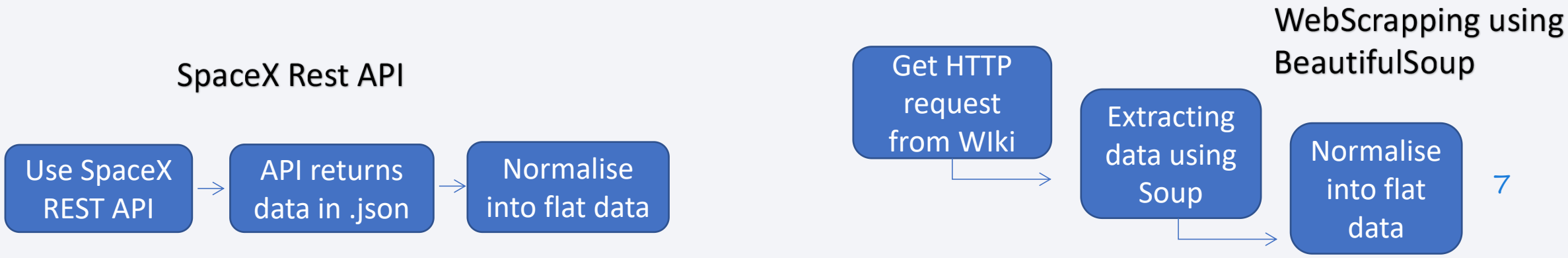
Methodology

Methodology

- Data collection methodology:
  - SpaceX Rest API
  - WebScrapping from Wiki
- Perform data wrangling { Transforming data for Machine Learning }
  - One Hot Encoding data and desired feature selection
- Perform Exploratory Data Analysis (EDA) using visualization and SQL
  - Scatter graphs, Bar graphs to show relationships
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Building, Tuning, evaluating using classification models

Data Collection

- The Datasets was collected by
  - Using SpaceX REST API, the launch data was gathered
  - It gives data related to rocket used, payload delivered, launch specification, landing specifications and landing outcomes.
  - We needed these data to predict whether spaceX will land a rocket or not
  - https://api.spacexdata.com/v4/launches/past -- API endpoints or URL
  - The alternative way for obtaining Falcon9 launch data is webScraping Wikipedia using BeautifulSoup.



Data Collection - SpaceX API

1. Response from API

```
spacex_url="https://api.spacexdata.com/v4/launches/past"
response = requests.get(spacex_url)
```

2. Converting Response to .json file

```
response = requests.get(static_json_url).json()
data = pd.json_normalize(response)
```

3. Apply custom functions to clean data

```
getBoosterVersion(data)
getLaunchSite(data)
getPayloadData(data)
getCoreData(data)
```

4. Apply list to dictionary then dataframe

```
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_data = pd.DataFrame(launch_dict,index=None)
```

GitHub URL to Notebook

5. Filter dataframe and export to flat file (.csv)

```
data_falcon9 = launch_data[launch_data['BoosterVersion']!='Falcon 1']
```

```
data_falcon9.to_csv('dataset_part_1.csv', index=False)
```

Data Collection - Scraping

1. Response from HTML

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

2. Creating BeautifulSoup Object

```
soup = BeautifulSoup(response.content, "html.parser")
```

3. Finding tables

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

4. Getting Column names

```
column_names = []
temp = soup.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
```

5. Creating dictionary

```
launch_dict= dict.fromkeys(column_names)

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'] = []
launch_dict['Version Booster']=[]
launch_dict['Booster landing']=[]
launch_dict['Date']=[]
launch_dict['Time']=[]
```

GitHub URL to Notebook

6. Appending data to Keys (Refer Notebook)

```
extracted_row = 0
#Extract each table
for table_number,table in enumerate(soup.find_all('table'
   # get table row
    for rows in table.find_all("tr"):
        #check to see if first table heading is as number
        if rows.th:
            if rows.th.string:
                flight_number=rows.th.string.strip()
                flag=flight_number.isdigit()
        else:
            flag=False
```

7. Converting dictionary to dataframe

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

8. Dataframe to .csv

```
df.to_csv('SpaceX webscapped', index=False)
```

Data Wrangling

In the dataset, there serveral different cases where the booster failed to land due to accidents and many more. Like, True ocean means, the mission was successfully landed to a specific region of ocean where false Ocean means vice versa. True RTLS means the mission was successfully landed to a ground pad, Flase RTLS the vice versa. Similarly True ASDS means, successfully landed om a drone ship.

We should convert those outcomes to labels with 1 (booster successfully landed), 0 (unsuccessfull)

GitHub URL to Notebook

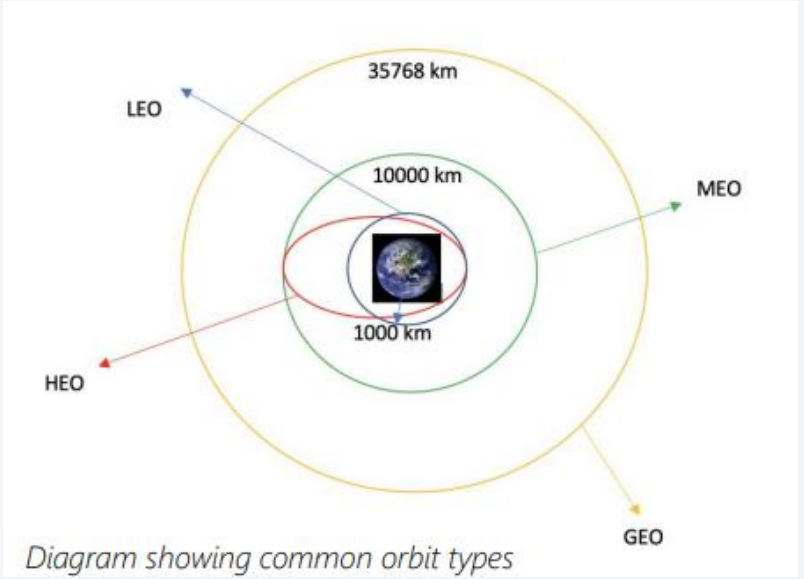


Diagram showing common orbit types

Exploratory Data Analysis EDA on Dataset

Calculate number of launches at each site

Calculate the number and occurence of each orbit

Calculate the number and occurrence of mission outcome per orbit type

Export data as .csv

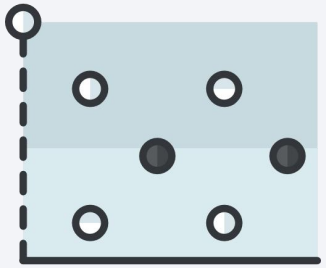
Create a landing Outcome label from outcome column

Work out success rate for every landing dataset

EDA with Data Visualization

Scatter Graphs drawn for:

- Flight Number VS. Payload Mass
- Flight Number VS. Launch Site
- Payload VS. Launch Site
- Orbit VS. Flight Number
- Payload VS. Orbit Type
- Orbit VS. Payload Mass



Scatter plots show how much one variable is affected by another. The relationship between two variables is called their correlation .

Bar Graph drawn for:

- Mean vs Orbit

Bar plot is easier to compare between different groups at a glance. Could show big changes in data over time

Line Graph drawn for:

- Success Rate vs Year

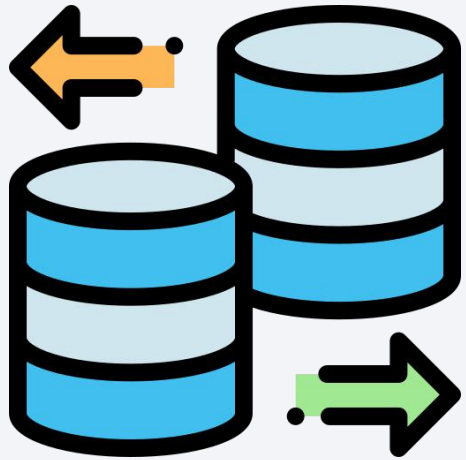
Line Graph shows data variables and trends very clearly, makes it easlier to predict for future

GitHub URL to Notebook

EDA with SQL

SQL queries were used to gather information about the dataset.

Following are few Questions which are answered using SQL queries from dataset:



- ✓ Displaying the names of the unique launch sites in the space mission
- ✓ Displaying 5 records where launch sites begin with the string 'KSC'
- ✓ Displaying the total payload mass carried by boosters launched by NASA (CRS)
- ✓ Displaying average payload mass carried by booster version F9 v1.1
- ✓ Listing the date where the successful landing outcome in drone ship was achieved.
- ✓ Listing the names of the boosters which have success in ground pad and have payload mass greater than 4000 but less than 6000
- ✓ Listing the total number of successful and failure mission outcomes
- ✓ Listing the names of the booster_versions which have carried the maximum payload mass.
- ✓ List the failed landing outcomes in drone ship, their booster versions, and launch site names for in year 2015
- ✓ Ranking the count of successful landing_outcomes between the date 2010-06-04 and 2017-03-20 in descending order.

GitHub URL to Notebook

Building an Interactive Map with Folium

Visualizing the Data in an interactive map

The Latitude and Longitude Coordinates at each launch site is marked in circles and labeled the name of the corresponding launch site.

We created a MarkerCluster() and assigned the classes launch_outcomes(failures, successes) 0 and 1 with **Green** and **Red** markers on the map

We also calculated the distance from the Launch Site to various landmarks to find various trends about what is around the Launch Site.

Insights like:

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

[GitHub URL to Notebook]

Build a Dashboard with Plotly Dash

The dashboard is built with Flask and Dash web framework.

Graphs

- Pie Chart showing the total launches by a certain site/all sites
- display relative proportions of multiple classes of data.
- size of the circle is made proportional to the total quantity it represents

Scatter Graph showing the relationship with Outcome and Payload Mass (Kg) for the different Booster Versions

- It indicates a non-linear pattern.
- The range of data flow, i.e. maximum and minimum value, can be determined.
- Observation and reading are straightforward.

Predictive Analysis (Classification)

BUILDING MODEL

- Loading dataset into NumPy and Pandas
- Transform Data
- Train - Test - split
- Try building multiple machine learning algorithms
- Set parameters and algorithms to GridSearchCV
- Fit into the GridSearchCV objects and train our dataset.

EVALUATING MODEL

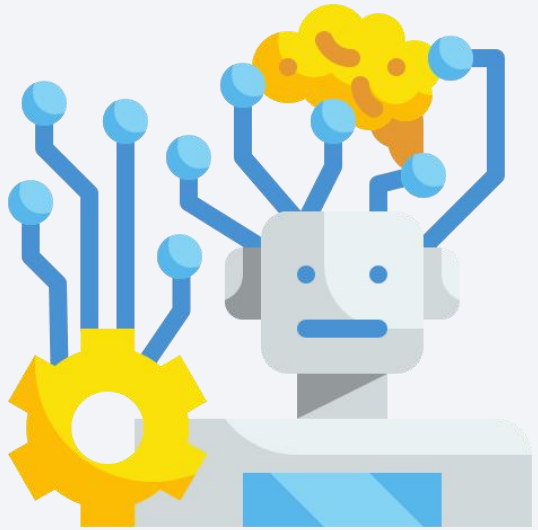
GitHub URL to Notebook

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

THE BEST PERFORMING CLASSIFICATION MODEL

- The model with the best accuracy score wins

In the notebook there is a table of algorithms with scores at the bottom

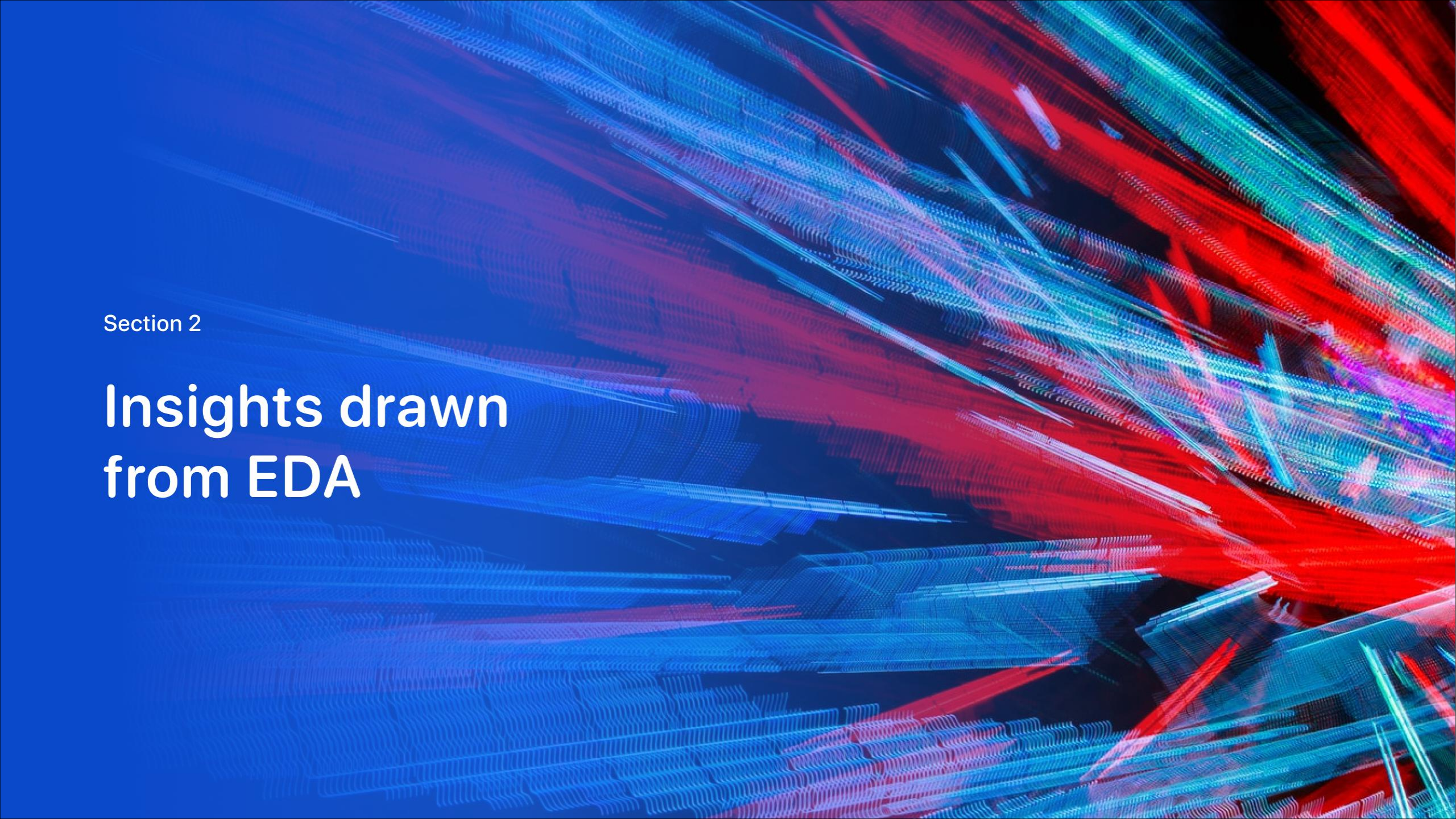


Results

- Exploratory Data Analysis results
- Interactive Dashboards in screenshots
- Predictive Analysis results

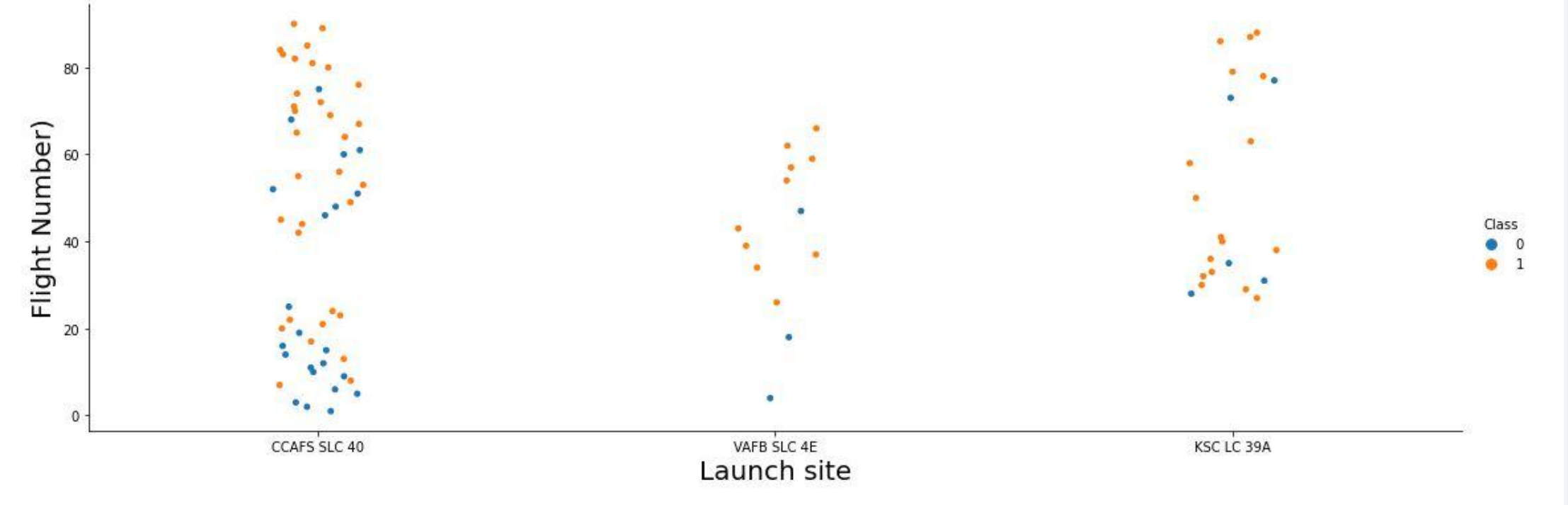
Section 2

Insights drawn from EDA



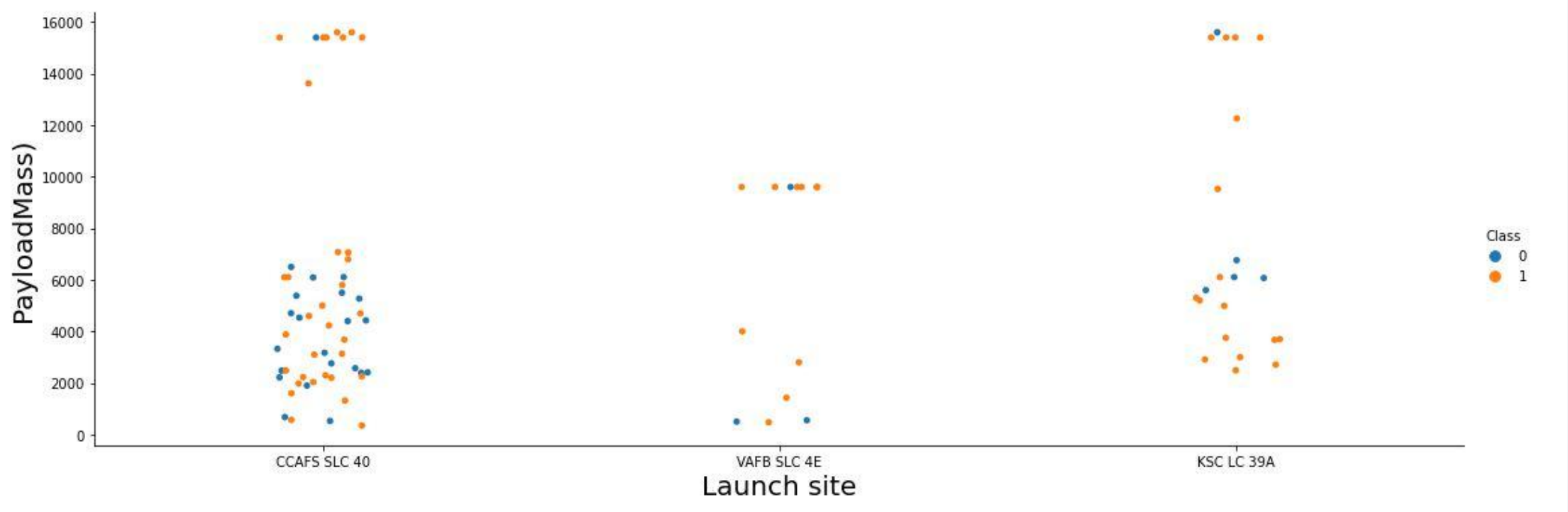
Flight Number vs. Launch Site

More the number of Flights at a launchsite, greater the success rate



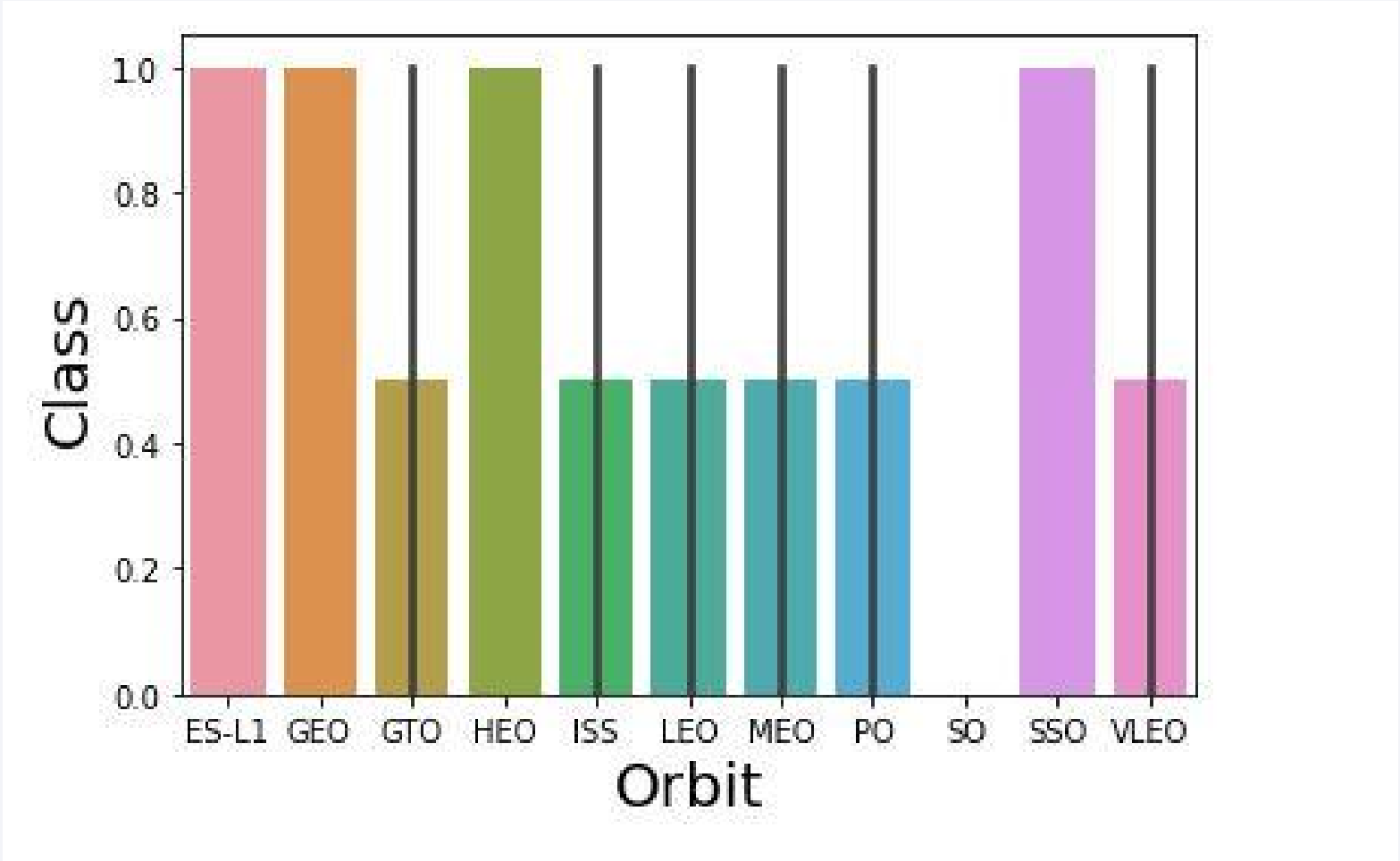
Payload vs. Launch Site

In case of CCAFS SLC 40, when payload mass is greater, higher is the success rate



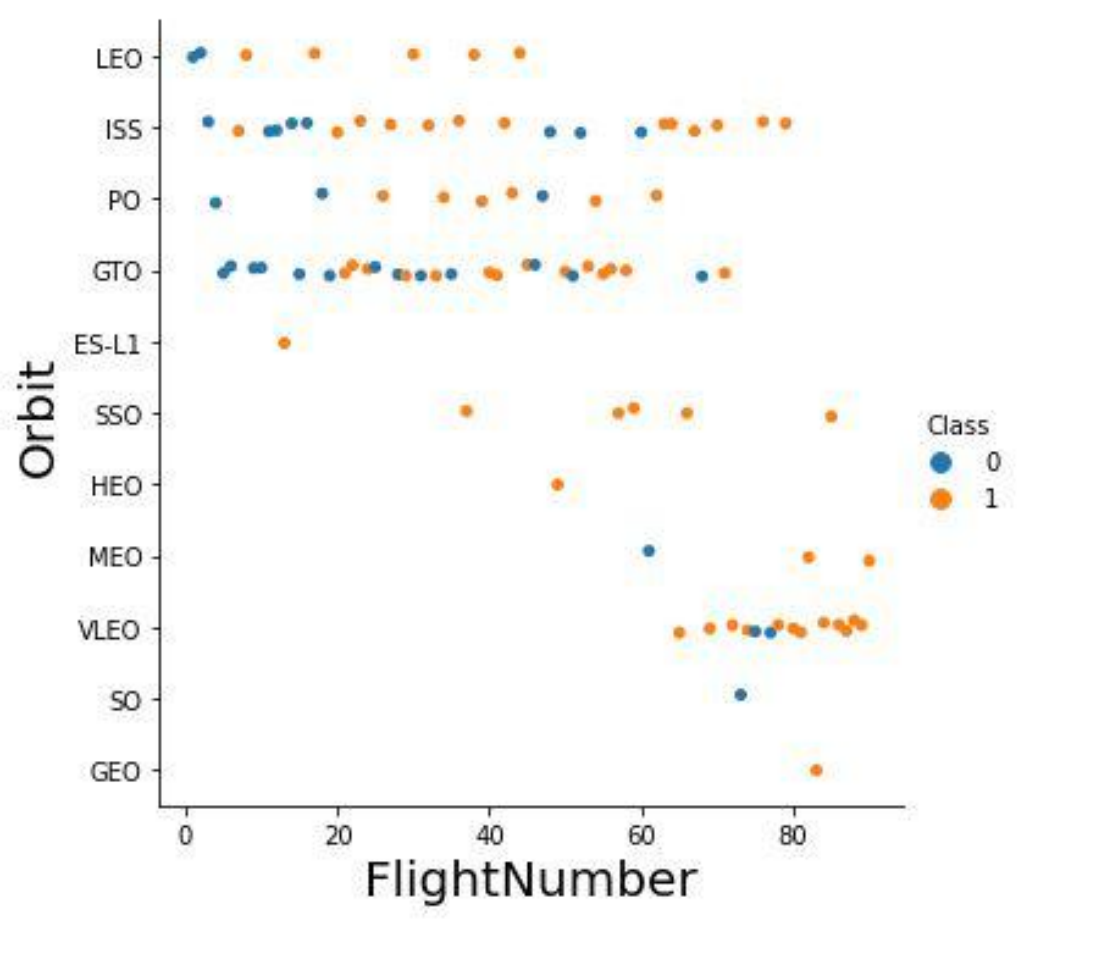
payload vs launch site plot does not give much insights to decide, launch site is depended on payload

Success Rate vs. Orbit Type



Orbits ES-L1, GEO, HEO, SSO has better success rates.

Flight Number vs. Orbit Type

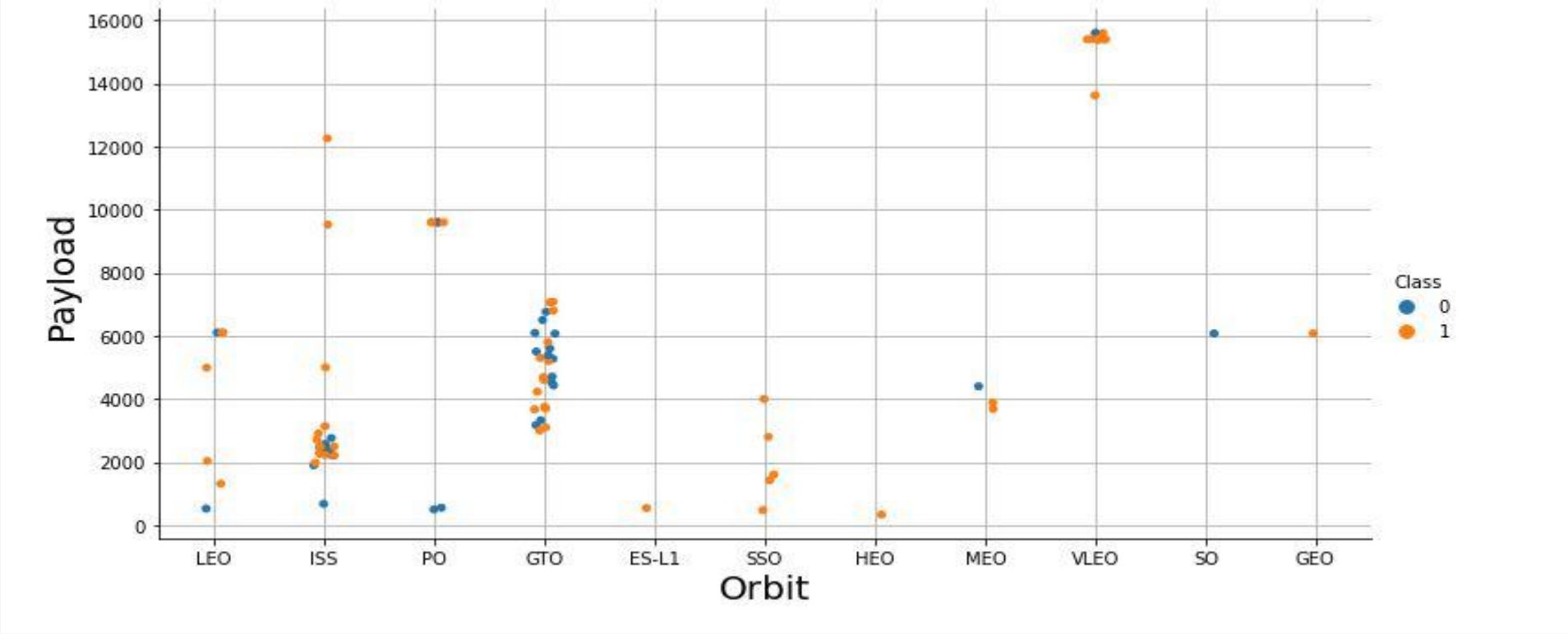


In LEO orbit, success appears related to the number of flights

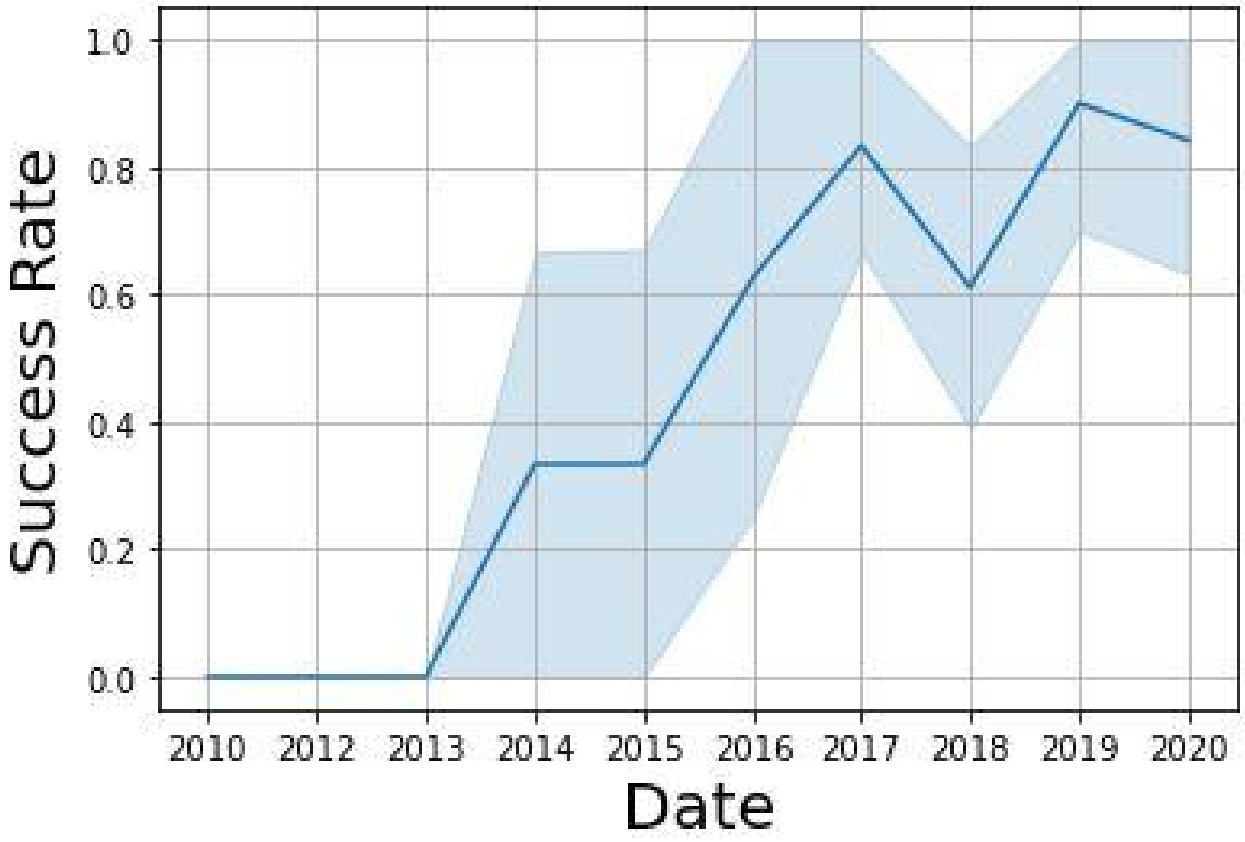
Also, there seems to be no relationship between flight number when in GTO orbit.

Payload vs. Orbit Type

Heavy payloads have a negative influence on GTO orbits and positive on GTO and Polar LEO (ISS) orbits.



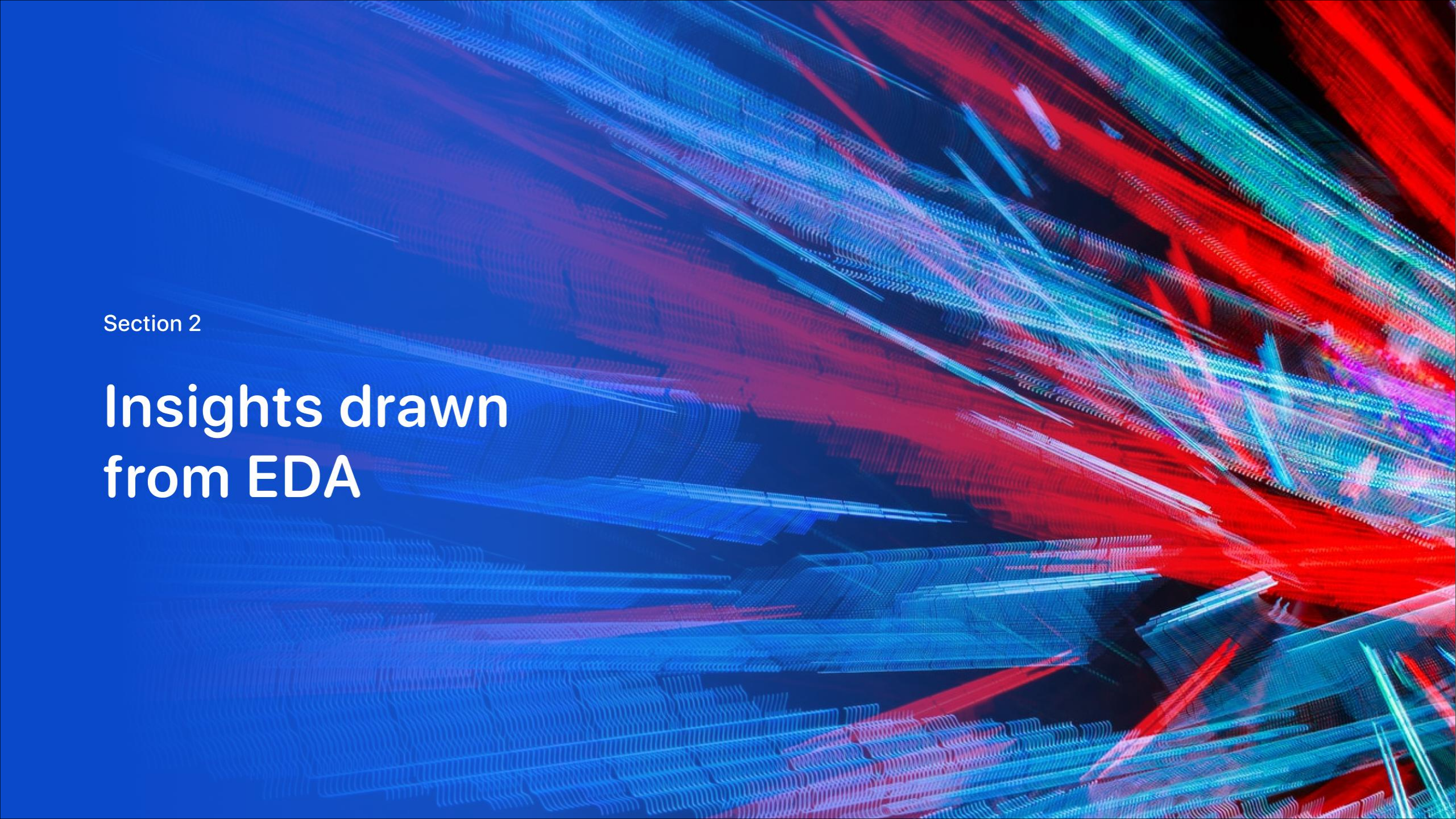
Launch Success Yearly Trend



The success rate since 2013 kept increasing till 2020 with a small down fall in the year 2018

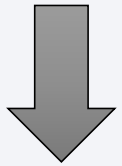
Section 2

Insights drawn from EDA



All Launch Site Names

```
select DISTINCT launch_site from SPACEXTBL
```



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

QUERY EXPLAINATION

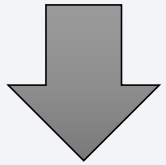
DISTINCT means that it will only show Unique values in the Launch_Site column from SpaceXtbl

Launch Site Names Begin with 'CCA'

QUERY EXPLAINATION

TOP 5 means, it will only show 5 records from tblSpaceX and LIKE search for the words ‘KSC%’ and the percentage in the end suggests that the Launch_Site name must start with KSC.

```
SELECT * FROM SPACEXTBL WHERE lower(launch_site) LIKE 'cca%' or lower(launch_site) LIKE 'CCA%' limit 5
```



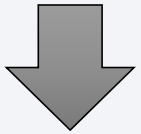
DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight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

QUERY EXPLAINATION

SUM shows the total in the column PAYLOAD_MASS_KG_ and WHERE clause filters the dataset to only perform calculations on Customer NASA (CRS)

```
select sum(payload_mass__kg_) as  total_payload_mass from spacextbl where customer = 'NASA (CRS)'
```



total_payload_mass
45596

Average Payload Mass by F9 v1.1

QUERY EXPLAINATION

AVG gives the average in the column PAYLOAD_MASS_KG_ and WHERE clause filters the dataset to only perform calculations on Booster_version F9 v1.1

```
select avg(PAYLOAD_MASS__KG_) as  Avg_payload_mass from spacextbl where BOOSTER_VERSION = 'F9 v1.1'
```

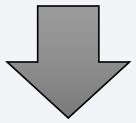
avg_payload_mass
2928

First Successful Ground Landing Date

QUERY EXPLAINATION

MIN brings out the minimum date in the column Date and WHERE clause filters the dataset to only perform calculations on Landing_Outcome Success (drone ship)

```
select min(DATE) AS DATE from SPACEXTBL where landing__outcome like ('%Success%')
```



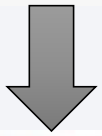
DATE
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

QUERY EXPLAINATION

Selecting only Booster_Version and then the WHERE clause filters the dataset to Landing_Outcome =Success (drone ship), The AND clause specifies additional filter conditions Payload_MASS_KG_ > 4000 AND Payload_MASS_KG_ < 6000

```
SELECT booster_version FROM SPACEXTBL WHERE landing__outcome = 'Success (drone ship)' and (payload_mass__kg_ between 4000 and 6000)
```



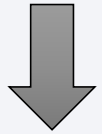
booster_version
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
F9 FT B1031.2

Total Number of Successful and Failure Mission Outcomes

QUERY EXPLAINATION

subqueries are used to produce the results. LIKE helps to filter out the word using '%Success%' included it in the dataset and same case for the word failure using '%Failure%'. Percentage on both side filters the words starting or ending or anywhere in between.

```
SELECT(SELECT Count(Mission_Outcome) from SPACEXTBL where Mission_Outcome LIKE '%Success%') as Successful_Mission_Outcomes,
(SELECT Count(Mission_Outcome) from SPACEXTBL where Mission_Outcome LIKE '%Failure%') as Failure_Mission_Outcomes
```



	Successful_Mission_Outcomes	Failure_Mission_Outcomes
0	100	1

Boosters Carried Maximum Payload

QUERY EXPLAINATION

DISTINCTmeans that it will only show Unique values in the Booster_Version column from SpaceXtbl , and GROUP BY puts the list in order set to a certain condition. DESC arrangs the dataset into descending order

```
select booster_version from SPACEXTBL where payload_mass__kg_ = (select max(payload_mass__kg_) from SPACEXTBL)
```

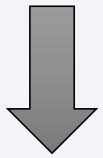
booster_version
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
F9 FT B1031.2

2015 Failed landing outcomes in drone ship

QUERY EXPLAINATION

selecting the columns, landing_outcome, booster version, launch site from spacextbl table and then filtering out the date using LIKE ‘%2015%’ , then checking the word ‘%drone%’ in landing outcome column

```
select landing__outcome, booster_version, launch_site from SPACEXTBL WHERE landing__outcome LIKE '%drone%' and DATE LIKE '%2015%'
```



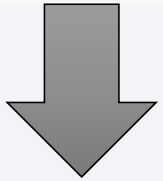
landing__outcome	booster_version	launch_site
Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40
Precluded (drone ship)	F9 v1.1 B1018	CCAFS LC-40

Rank success count between 2010-06-04 and 2017-03-20

QUERY EXPLAINATION

Function COUNT counts records in column, WHERE filters the data , using AND as condition

```
SELECT COUNT(Landing_Outcome) AS sl FROM SPACExTBL WHERE (Landing_Outcome LIKE '%Success%') AND (Date >'04-06-2010') AND (Date < '20-03-2017')
```



	Successful Landing Outcomes Between 2010-06-04 and 2017-03-20
0	34

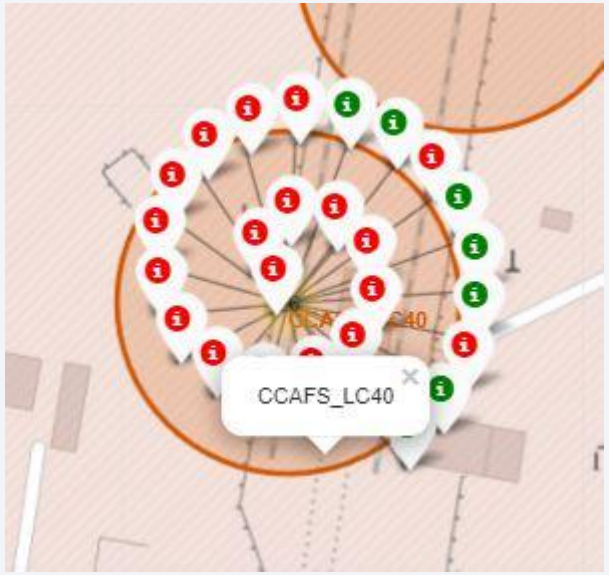
Section 4

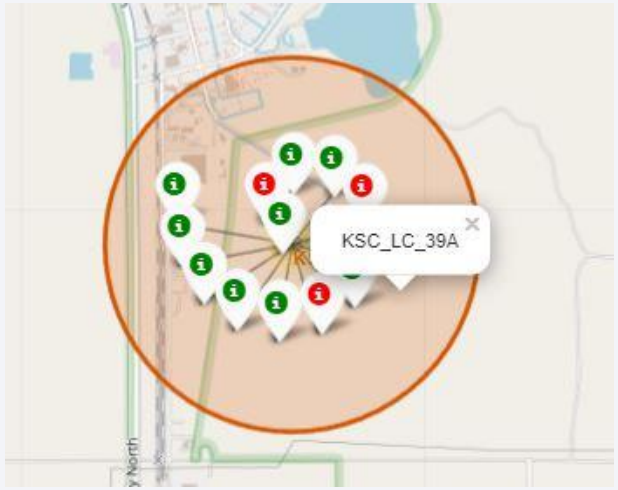
Launch Sites Proximities Analysis

All launch sites marked global map

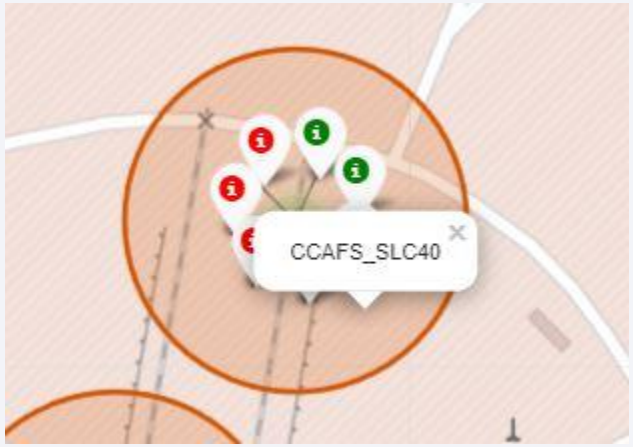
The SpaceX launch sites are in the United States of America coasts.

Labelled Markers





Florida Launch Sites

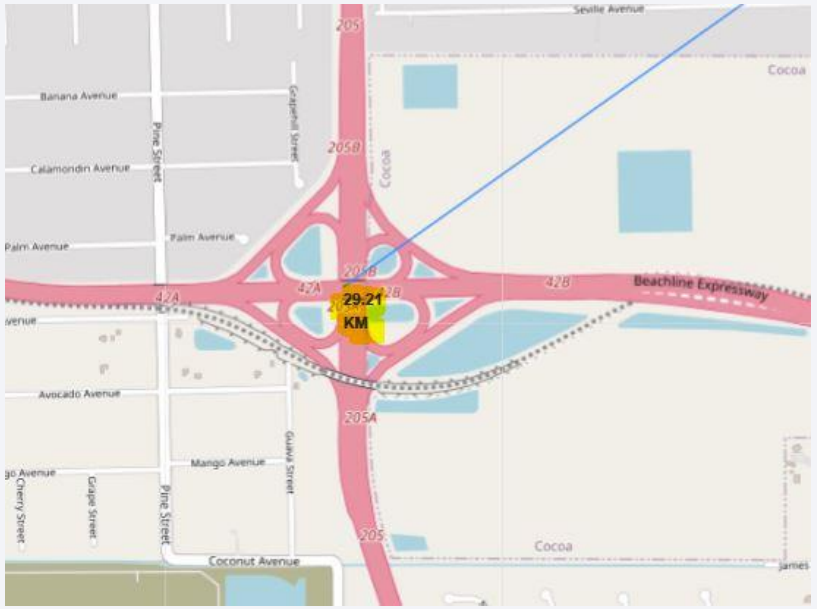


Green Marker - successful Launches

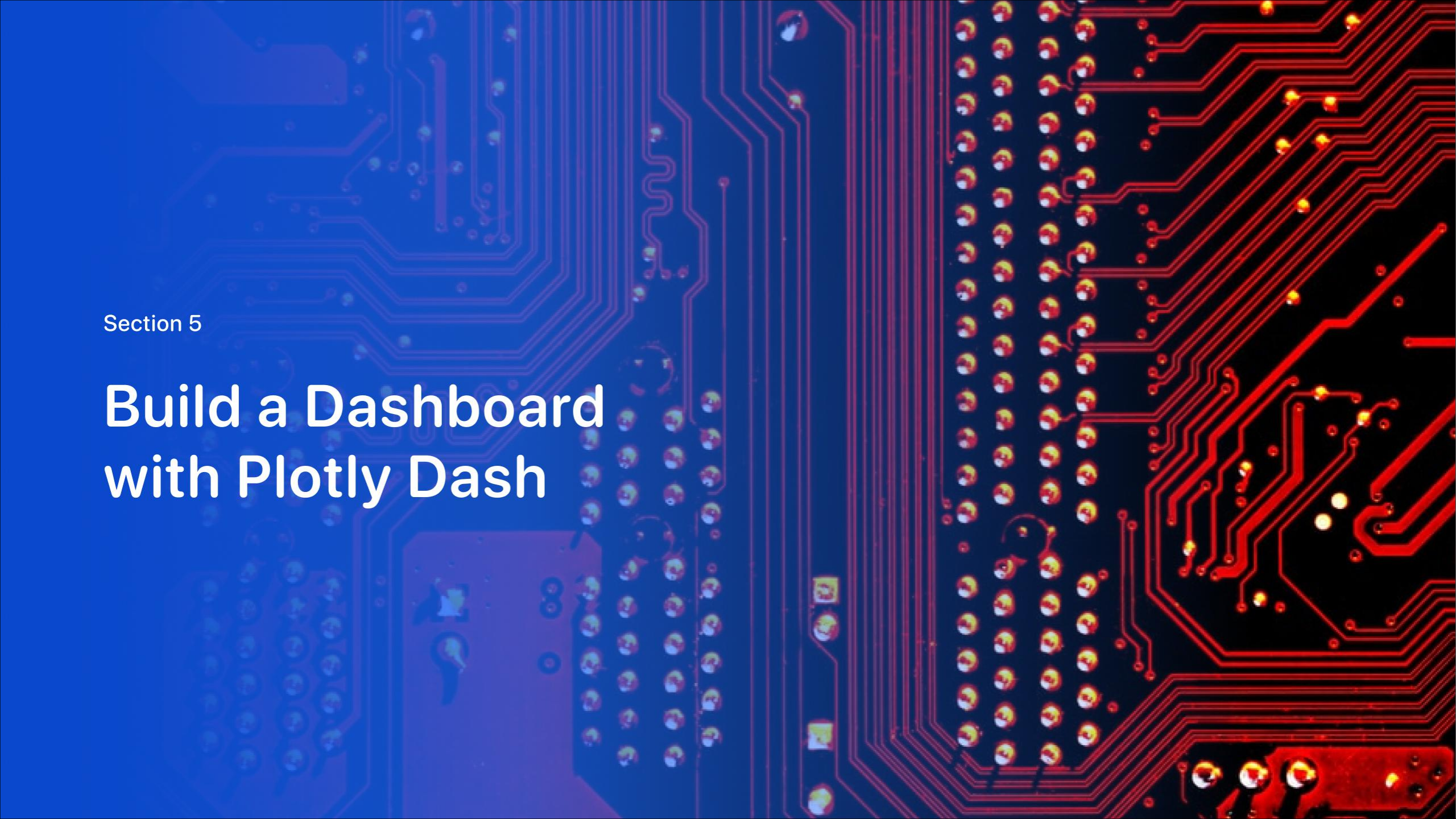
Red Marker - Failures Launches

California Launch Site

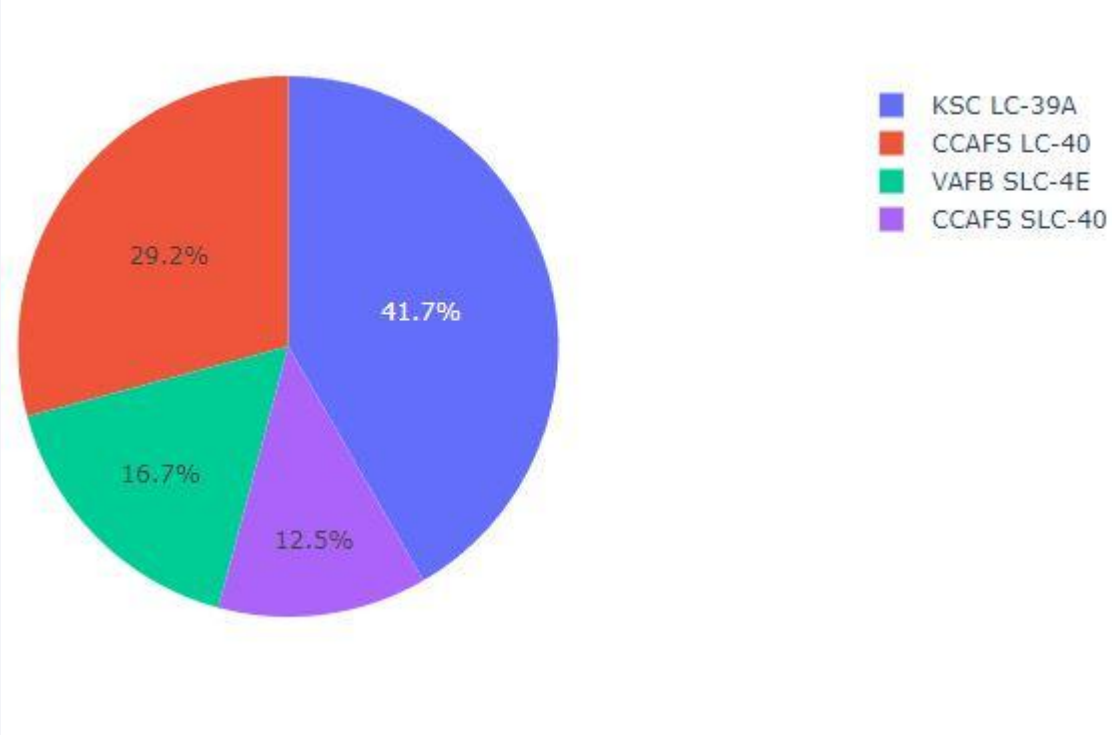
Launch Sites distance to landmarks with CCAFS-SLC-40 as a reference



Launch sites are not close proximity to railways,Highways, cities and is in close proximity to coastline

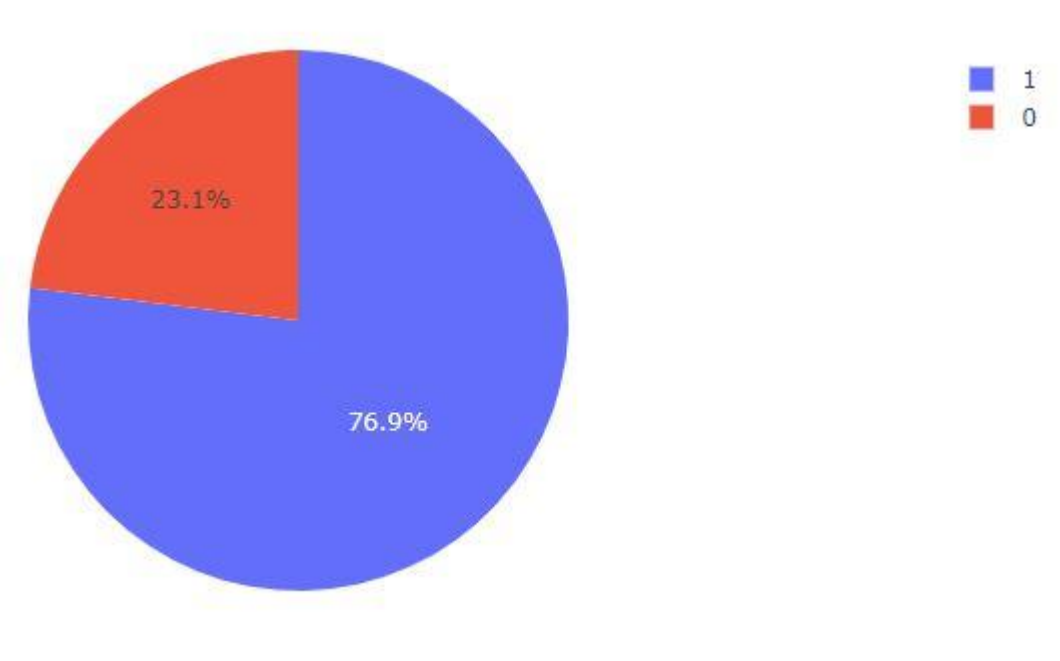

Section 5
Build a Dashboard with Plotly Dash

DASHBOARD – Pie chart showing the success percentage achieved by each launch site



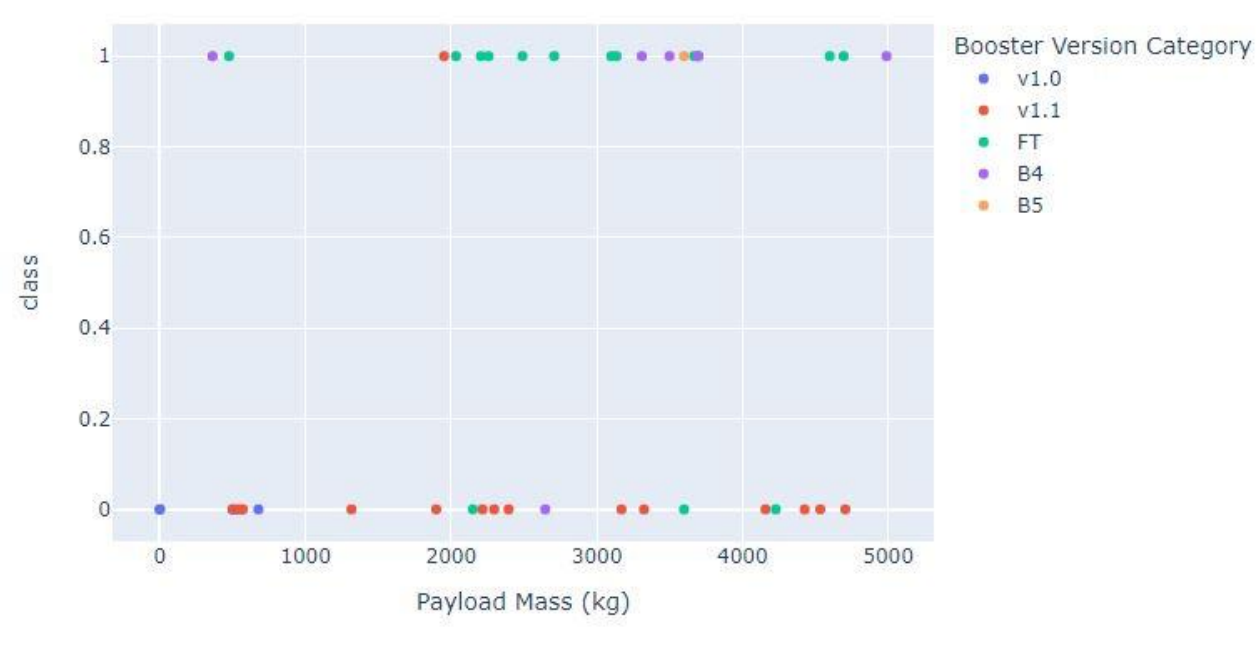
KSC LC-39A had the most successful launches 41.7%

DASHBOARD Pie chart for the launch site with highest launch success ratio

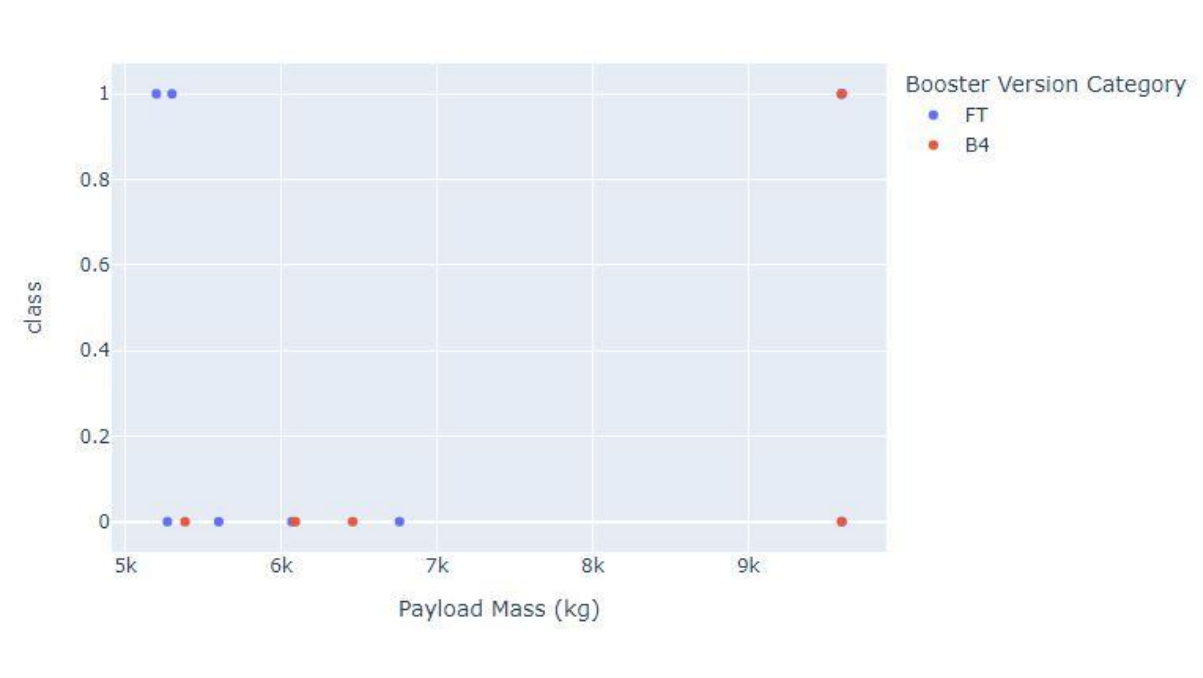


KSC LC-39A achieved a 76.9% success rate with 23.1% failure rate

DASHBOARD – Payload vs. Launch Outcome scatter plot for all sites, with different payload selected in the range slider



Low Weighted Payload 0kg – 5000kg



Heavy Weighted Payload 5000kg – 10000kg

The success rates for low weighted payloads is higher when compared

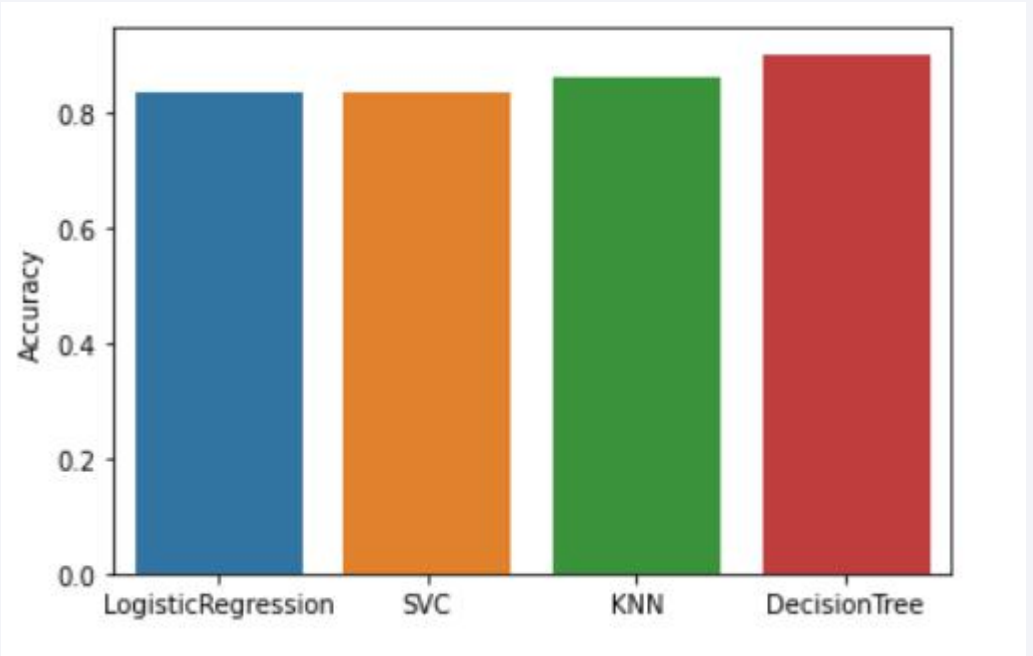
Section 6

Predictive Analysis (Classification)

Classification Accuracy

The accuracy is close to each other.
Best Model and its parameters are,

	Accuracy	Algorithm
0	0.834	LogisticRegression()
1	0.834	SVC()
2	0.861	KNeighborsClassifier()
3	0.902	DecisionTreeClassifier()

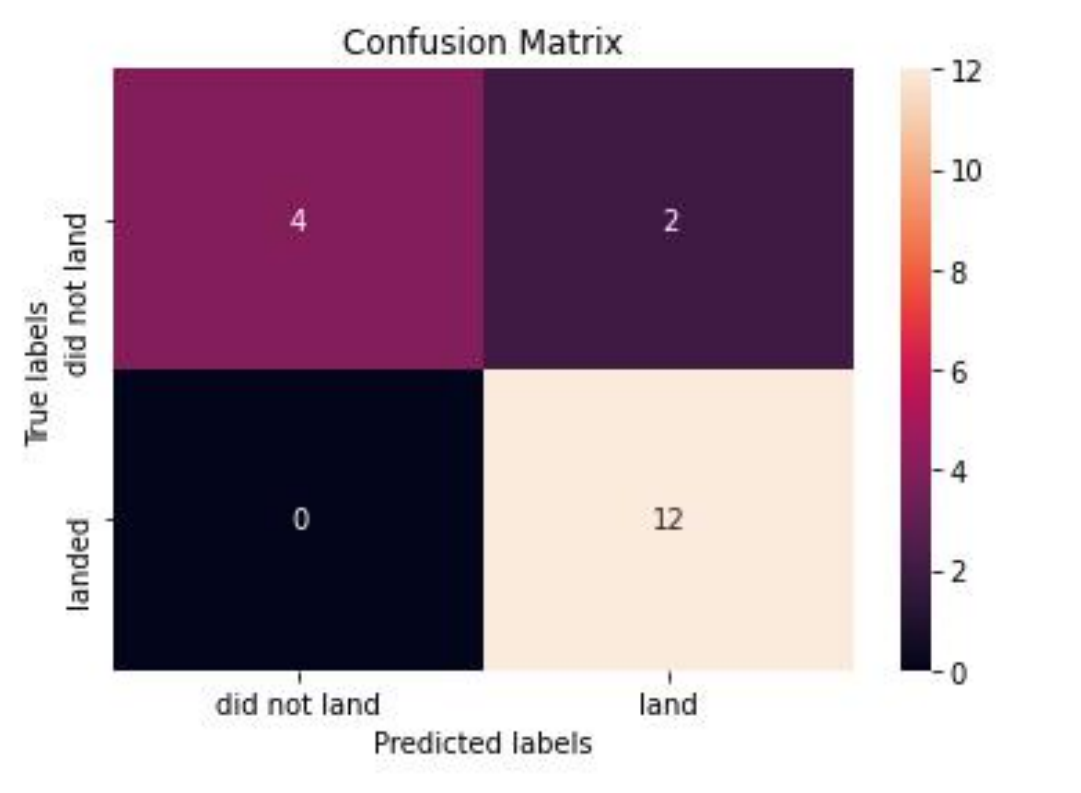


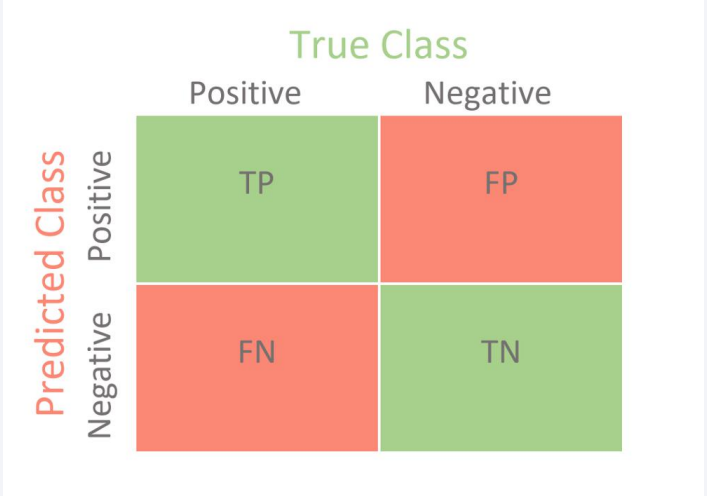
Graph between Model Names and Accuracy

Decision Tree Classifier has won with 90% Accuracy. The best hyperparameters for the tree are,

```
Best parameters:  {'criterion': 'entropy', 'max_depth': 8,
'max_features': 'auto', 'min_samples_leaf': 2, 'min_samples
_split': 2, 'splitter': 'random'}
accuracy : 0.9019047619047619
```

Confusion Matrix





The Tree can distinguish between the different classes. We see that the major problem is false positives.

Conclusions

- ✓ Orbit GEO,HEO,SSO,ES-L1 has the best Success Rate
- ✓ KSC LC-39A had the most successful launches from all the sites
- ✓ The success rates for SpaceX launches is directly proportional time in years, shows a bright future
- ✓ The Tree Classifier Algorithm is the best for Machine Learning for this dataset
- ✓ Low weighted payloads perform better than the heavier payloads

Appendix

- IBM watson studio (JupyterNotebook)
- MySQL (DB6) (SQL Queries)
- IBM cloud (Storage)

Thank you!