IBM Data Science Capstone

SpaceX Falcon 9 Launch Project

ADAM ABDI

April 24th, 2022

OUTLINE

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

Executive Summary

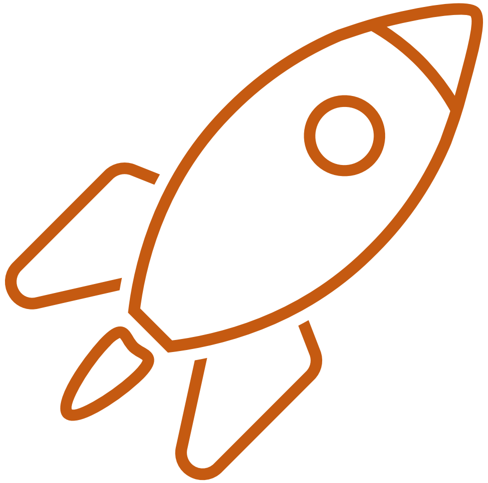
Methodologies

- Data collection
- Data Wrangling
- Exploratory Data Analysis (EDA) using Data Visualization
- Exploratory Data Analysis (EDA) using SQL
- Creating Interactive Map using Folium
- Dashboard with Plotly Dash
- Predictive analysis using Classification

Summarizing Results

- Exploratory Data Analysis (EDA) results
- Interactive analysis
- Predictive analysis

Introduction



Project background and context

As advertised in SpaceX website, Falcon 9 rocket launches costs of 62 million dollars while other competitors would cost them more than 165 million dollars each time. Most of the saving is coming from the fact SpaceX reuses the first stage which makes the rocket launches relatively inexpensive. In this project we would like to determine if first stage will land successfully, knowing this will help us determine the launch of the rocket. Even though most of the work is done by first stage and SpaceX's Falcon 9 can recover the first stage, sometimes first stage doesn't land as it will crash.

For this project our job is to determine the price of each launch. We will gather information about SpaceX and create a dashboard for our team. We will so determine if SpaceX will reuse the first stage. In this project we will also train a machine learning model and use public information or data to predict if SpaceX will reuse the first stage.

Problems to find solutions

- Determine if Falcon 9 first stage will land successfully so that SpaceX will reuse it.
- What are the rochet variables that will determine high success landing rate for the first stage of Falcon 9 rocket?
- Extract exciting insights from the collected data.

Section 1

Methodology

Methodologies

- **Data collection**
  - SpaceX launch data was collected using SpaceX REST API.
  - This API will provide us data about launches, rocket used, payload delivered, launch specifications, landing specifications, and landing outcome.
  - Our main goal is to predict whether SpaceX will attempt to land a rocket or not. We will work the SpaceX REST API with URL api.spacexdata.com/v4/launches/past
  - We also do Web scraping Falcon 9 launch records from Wiki pages using Python BeautifulSoup where we parse the data and convert them Pandas data frame.

- **Perform data wrangling**

  - The collected raw data is transformed to clean dataset by wrangling data using API, Sampling Data, and Dealing with Nulls.

- **Perform exploratory data analysis (EDA) using visualization and SQL**

- **Perform interactive visual analytics using Folium and Plotly Dash**

- **Perform predictive analysis using classification models**

  How to build, tune, evaluate classification models

Methodologies

- **Web Scrapping**
  - Obtain HTML response from HTML
  - Extract and parse data using Python BeautifulSoup
  - Normalize data and covert to csv file format.

- **SpaceX REST API steps**
  - Use SpaceX REST API.
  - API will return SpaceX data in JSON format.
  - JSON data is normalized by converting to csv data

Data Collection – SpaceX

- Using SpaceX REST API
- Filtering Falcon 9 Data Frame and Data Cleaning
- API Response SpaceX Data in JSON file
- Normalize SpaceX Data in .csv file format

GitHub URL Jupyter Notebook

1. **Request rocket launch data from SpaceX API**

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

2. **Decode the response content as Json and then Normalize**

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

3. **To clean data custom functions were applied**

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

4. **Construct Dataset using the data obtained**

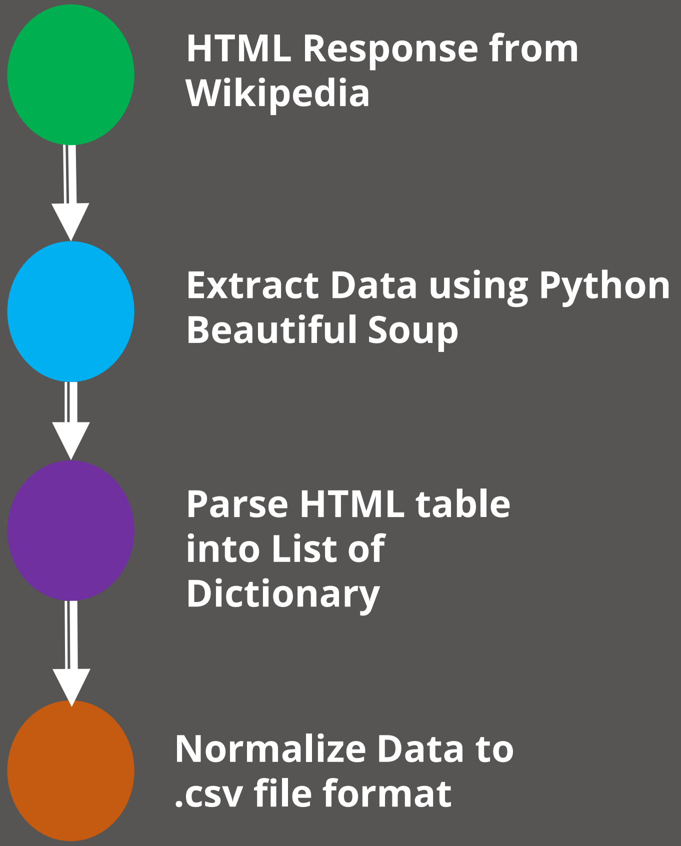
```
launch_dict = {'FlightNumber': list(data['flight_number']),
'Date': list(data['date']),
'BoosterVersion':BoosterVersion,
'PayloadMass':PayloadMass,
'Orbit':Orbit,
'LaunchSite':LaunchSite,
'Outcome':Outcome,
'Flights':Flights,
'GridFins':GridFins,
'Reused':Reused,
'Legs':Legs,
'LandingPad':LandingPad,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

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

5. **Data filtering and exporting to csv file format**

```
data_falcon9 =launch_data[launch_data['BoosterVersion'] ==   'Falcon 9']
data_falcon9.to_csv('dataset_part_1.csv', index=False)
```

Data Collection – Web Scrapping



GitHub URL Jupyter Notebook

1. **Perform HTTP GET to get Response from Falcon9 HTML page**

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

2. **BeautifulSoup Object**

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

3. **Find all tables elements**

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

4. **Obtain 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. **Create Dictionary**

```
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']=[]
```

6. **Appending data to keys**
   - Please refer to the block 20 of the Jupyter Notebook
7. **Convert dictionary to DataFrame and Save Dataframe to CSV file**

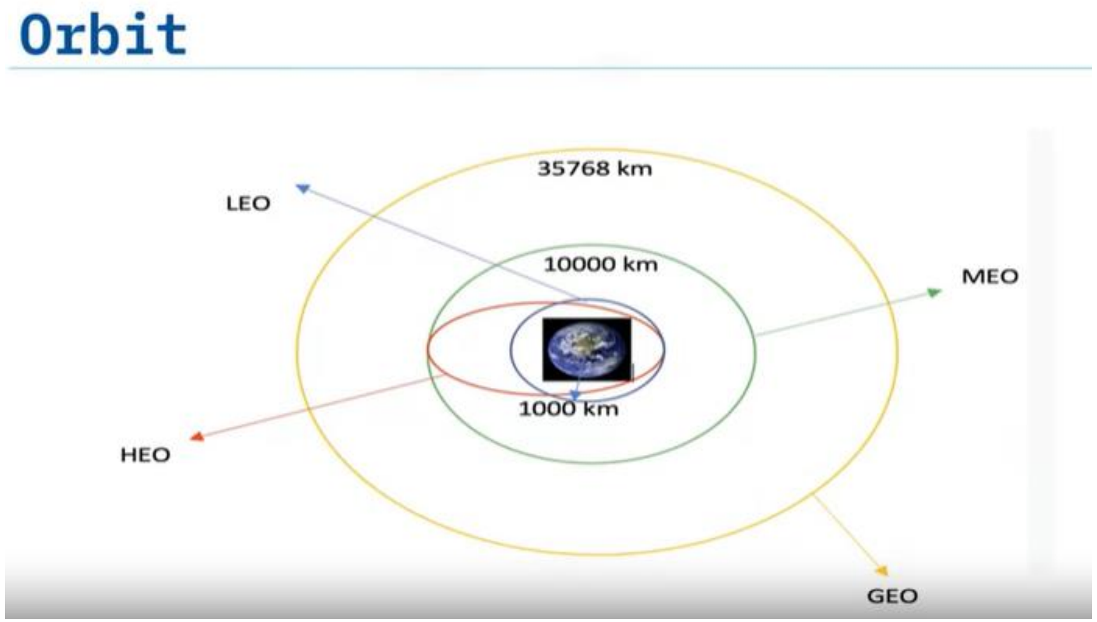
```
df = pd.DataFrame.from_dict(launch_dict)
df.to_csv('spacex_web_scraped.csv', index=False)
```

Data Wrangling

After raw data has been collected, you will need to improve the quality of the dataset by performing data wrangling or data munging. This is the process of transforming and mapping data from one raw data form into another format appropriate purpose such as analytics.
In this process we will perform Exploratory Data Analysis to figure out the label used for training supervised models.

In the data set, there are several different cases where the booster did not land successfully. Sometimes a landing was attempted but failed due to an accident.
Finally, we will convert those outcomes into Training Labels with 1 means the booster successfully landed 0 means it was unsuccessful.

Different Orbits of the payload



Data Wrangling

Exploratory Data Analysis (EDA) Performed

The number of launches on each site

The number and occurrence of each orbit

The number and occurence of mission outcome per orbit type

Create a landing outcome label from Outcome column

GitHub URL Jupyter Notebook

```
LaunchSite
CCAFS SLC 40     55
KSC LC 39A       22
VAFB SLC 4E      13
dtype: int64
```

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

```
Orbit
GTO      27
ISS      21
VLEO     14
PO        9
LEO       7
SSO       5
MEO       3
ES-L1     1
GEO       1
HEO       1
SO        1
dtype: int64
```

EDA with Data Visualization

Scatter plot

- 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

GitHub URL Jupyter Notebook

Bar Plot

- Mean vs. Orbit

Line Plot

- Success Rate vs. Year

EDA with SQL

SQL Queries performed to the Dataset Stored in Database table in DB2

- **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**

GitHub URL Jupyter Notebook

Build an Interactive Map with Folium

- Launch Site locations were analysis building Interactive Map with Folium since the launch success rate may depend also the location and proximities of a launch site. From the dataset we obtained the latitude and longitude coordinates of different launch site and included circle markers and name label on the launch site location.

- Marked the success/failed launches for each site on the map. Color was assigned for launch outcomes such as 1 for 'green' and 0 for 'red'
- Finally, we explored and analyzed the proximities of launch sites.
- After plotting the distance lines to the proximities, we find out the launch sites are not near railways, highways. However, the launch sites are near coastline and cities.

GitHub URL Jupyter Notebook

Build a Dashboard with Plotly Dash

- Dashboard application was built using Plotly Dash to perform interactive visual analytics on SpaceX launch data in real-time
- The Dashboard app contains input components such as dropdown list and range slider to interact with a pie chart and a scatter point chart.
- Interactive Pie Chart showing the Total success launches by site was created.
- Interactive scatter plot was created to show the correlation between Payload and success for all Sites for the different Booster versions

GitHub URL Jupyter Notebook

Predictive Analysis (Classification)

- Summarize how you built, evaluated, improved, and found the best performing classification model
- You need present your model development process using key phrases and flowchart
- Add the GitHub URL of your completed predictive analysis lab, as an external reference and peer-review purpose

Predictive Analysis (Classification) ...continued

Model Building

- Import Libraries and Define Auxiliary Functions (see Jupyter Notebook for details)
- Load Data frame
- Data Transforming
- Data was split into training and test datasets
- Machine algorithms appropriate to the dataset was used
- Created a GridSearchCV object using parameters and algorithms.
- Objects Created logistic regression, support vector Machine, decision tree classifier, K nearest neighbors

Model Evaluation

- Model accuracy
- Hyperparameters tuned for all algorithms
- Confusion Matrix Plotted

GitHub URL Jupyter Notebook

Predictive Analysis (Classification)

Model Improvement

- Perform Feature Engineering
- Fine Tune Algorithms

Best Classification Model

- The Best Performance model has best accuracy among the models.

GitHub URL Jupyter Notebook

Results

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

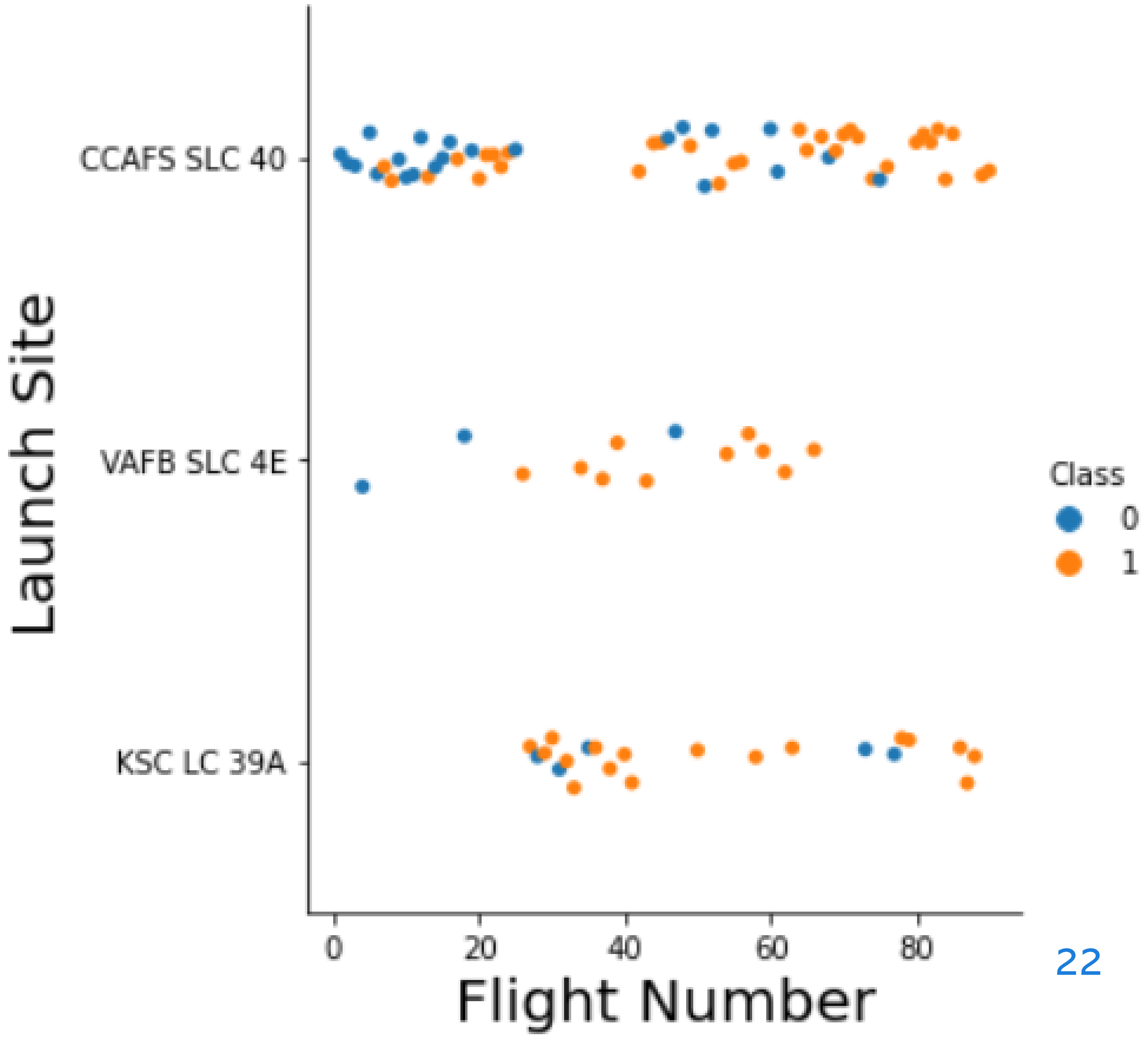
Section 2

Insights Drawn from EDA

EDA with Visualization

Flight Number vs. Launch Site

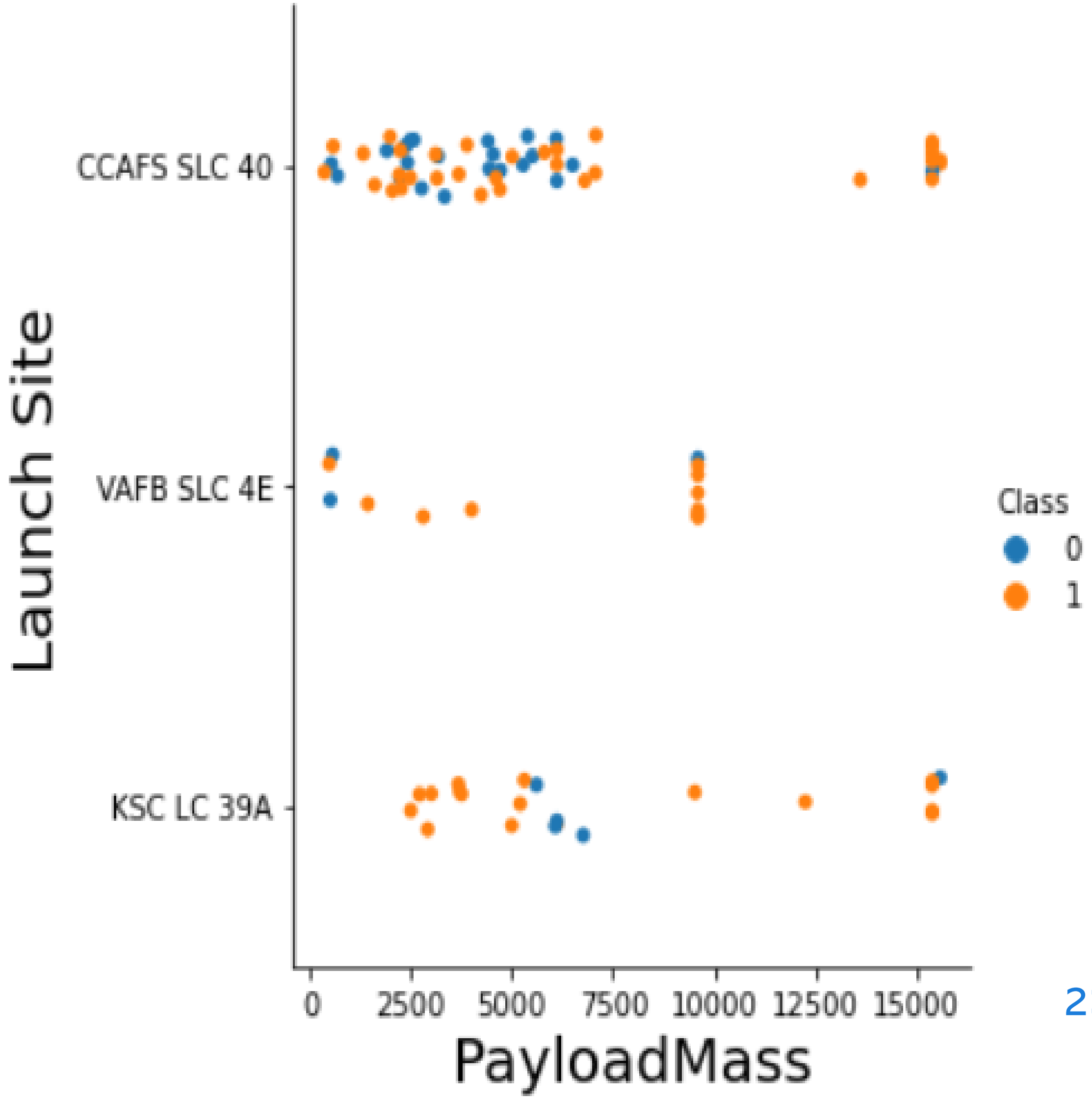
Flight Numbers greater 25 have higher success rate for all three launch sites.



Payload Mass vs. Launch Site

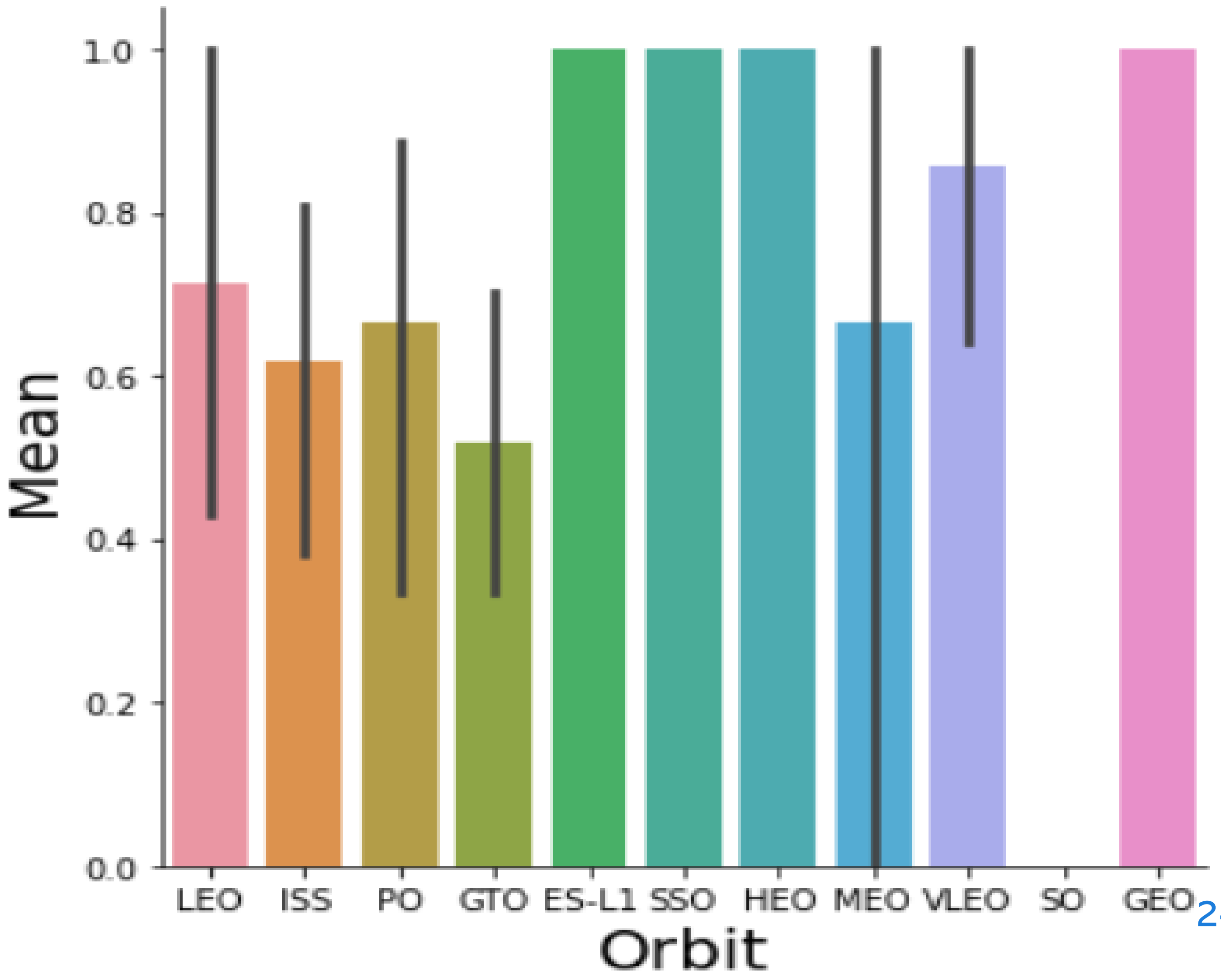
- The rockets launched VAFB-SLC 4E site the heavy payload Mass was not greater than 10000Kg.

- For launch Site CCAFS SLC 40 The greater payload mass the higher the success rate for the rocket



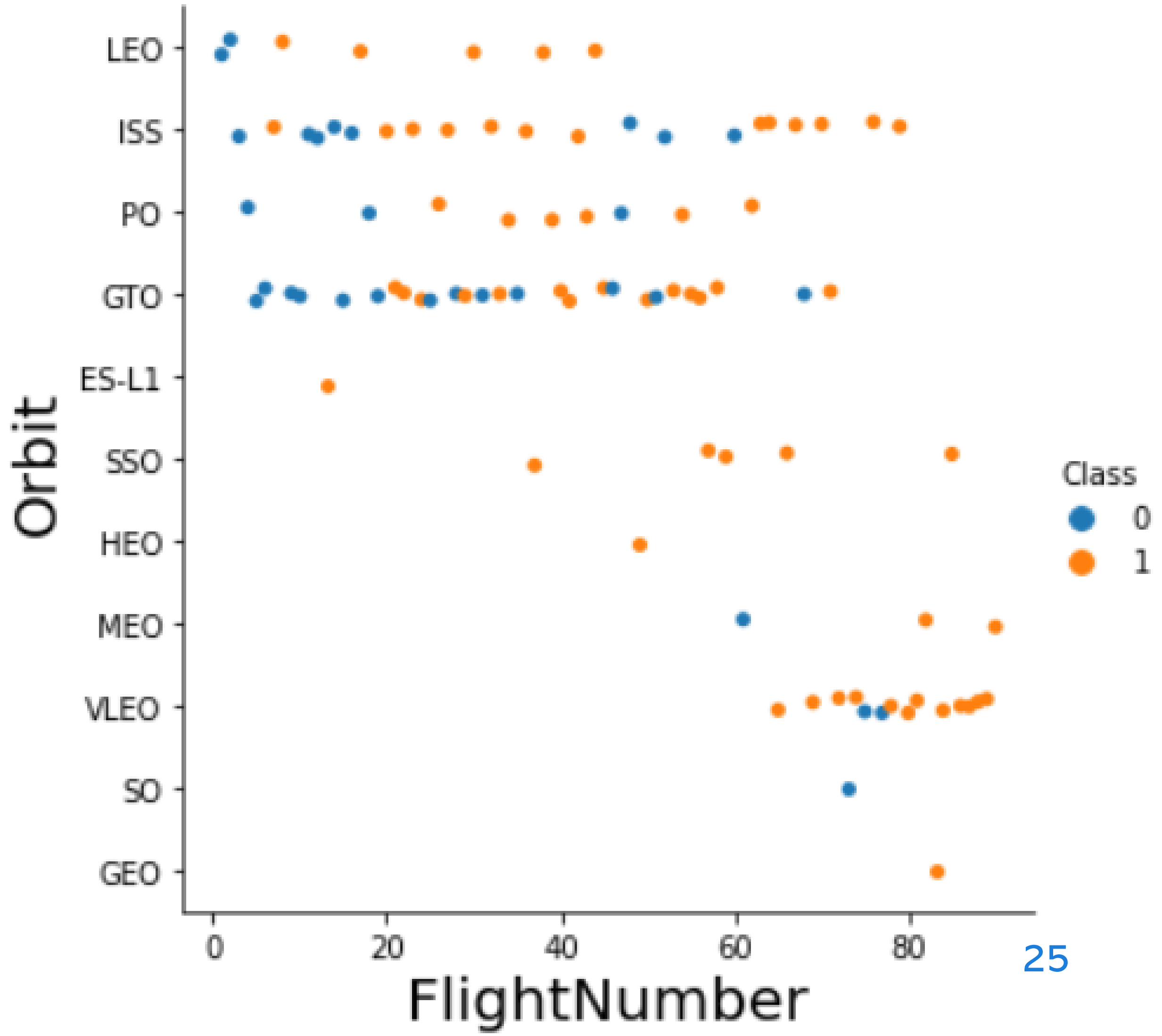
Success rate vs. Orbit type

Orbits of HEO, SSO, ESL1 GEO have the high success rate compared to rest of orbits



Flight Number vs. Orbit type

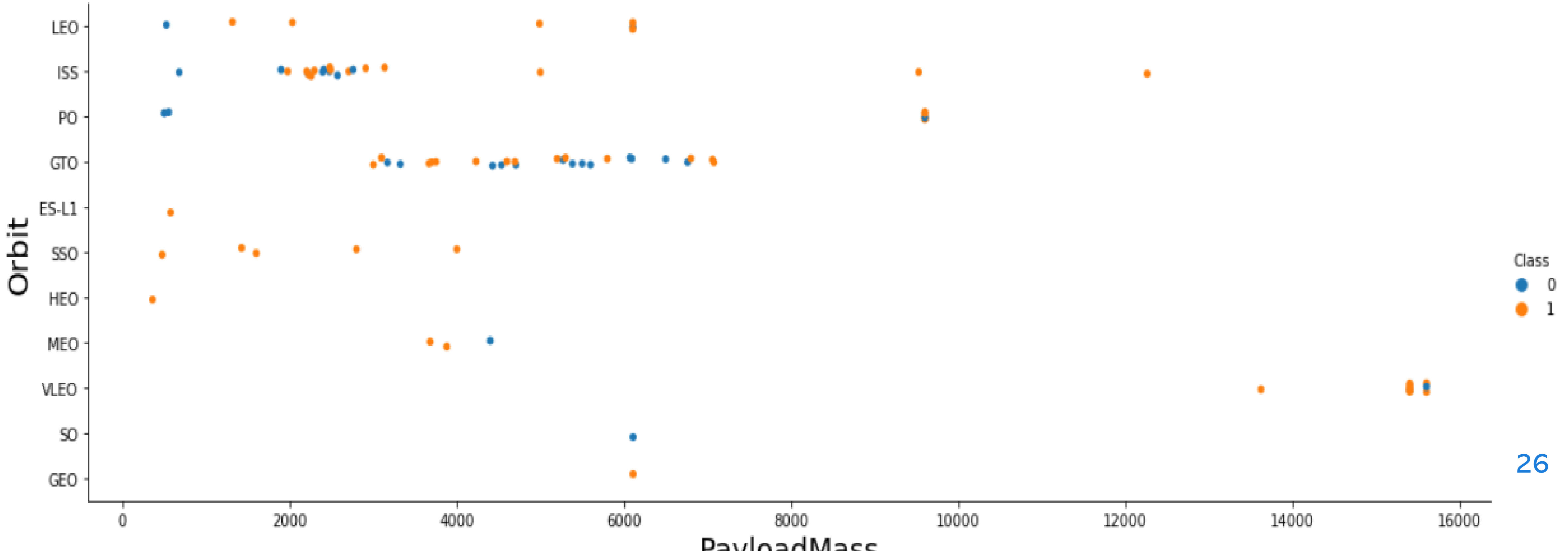
In the LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.



Payload vs. Orbit type

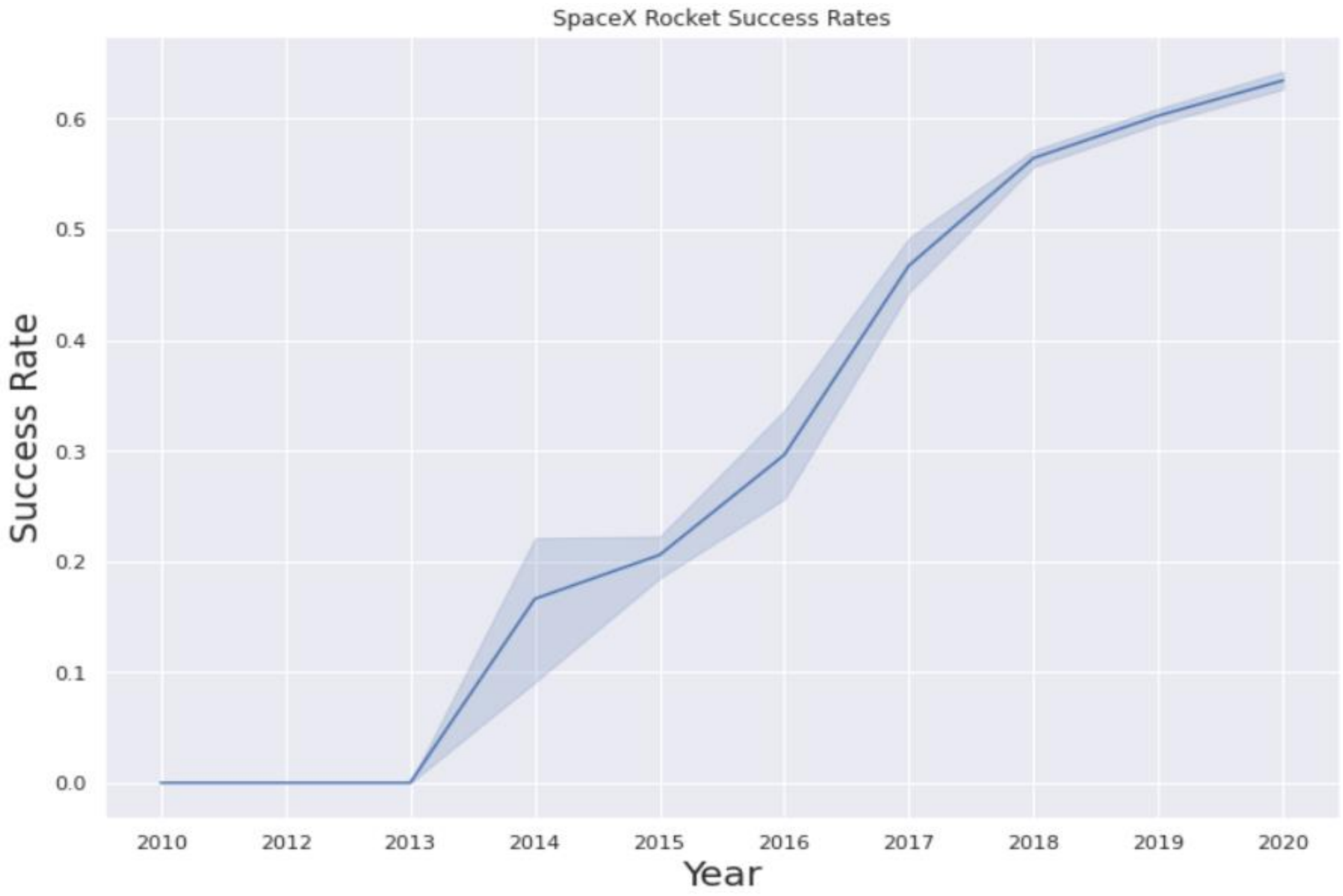
- With heavy payloads, the positive rate are more on orbits Polar, LEO and ISS

- GTO cannot be distinguished well as both positive and negative landing observed in heavy payloads.



Flight Number vs. Orbit type

The success rate increased drastically from the year 2013 and continued increasing up until 2020



Exploratory data Analysis (EDA) Using SQL

All Launch Site Names

SQL QUERY

```
SELECT DISTINCT launch_site FROM SPACEXTBL;
```

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

This SQL query displays unique names of the launch_site column in the SPACEXTBL table

Launch Site Names Begin with 'CCA'

SQL QUERY

```
SELECT * FROM SPACEXTBL WHERE launch_site LIKE 'CCA%' LIMIT 5;
```

This Query displays records where launch_site column like ‘CCA%’ and limits to 5 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

Total Payload Mass NASA(CRS)

```
SELECT SUM(payload_mass__kg_) as "Total payload mass of
NASA (CRS)" FROM SPACEXTBL WHERE customer = 'NASA
(CRS)';
```

SQL QUERY

This SQL query displays total payload mass carried by boosters launched by NSA(CRS) FROM PAYLOAD_MASS_KG_ column of the SPACEXTBL table

Total payload mass of NASA (CRS)
45596

Average Payload Mass by F9 v1.1

```
SELECT AVG(payload_mass__kg_) AS "AVG. payload mass"
FROM SPACEXTBL WHERE booster_version = 'F9 v1.1';
```

SQL QUERY

This SQL query displays average payload mass carried by boosters versionF9 v1.1 FROM PAYLOAD_MASS_KG_ column of the SPACEXTBL table

AVG. payload mass
2928

First Successful Ground Landing Date

```
SELECT MIN(DATE) FROM SPACEXTBL WHERE landing__outcome
LIKE 'Success%ground pad%';
```

SQL QUERY

This SQL query displays the date when the first successful landing outcome in ground pad was achieved

	First Successful Ground Landing Date
0	2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

SQL QUERY

```
SELECT booster_version, payload_mass__kg_,
landing__outcome FROM SPACEXTBL WHERE landing__outcome
LIKE 'Success%(drone ship)' AND (payload_mass__kg_
BETWEEN 4000 AND 6000);
```

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

This SQL query displays the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

Total Number of Successful and Failure Mission Outcomes

SQL QUERY

```
SELECT COUNT(mission_outcome) as Success_Count FROM
SPACEXTBL WHERE mission_outcome = 'Success' or
mission_outcome LIKE 'Failure (in flight)';
```

This SQL query displays the total number of successful and failure mission outcomes

success_count
100

Boosters Carried Maximum Payload

SQL QUERY

```
SELECT booster_version, payload_mass__kg_ FROM
SPACEXTBL WHERE payload_mass__kg_ = (SELECT
MAX(payload_mass__kg_) FROM SPACEXTBL)
```

This SQL query displays the names of the booster_version which have carried the maximum payload mass.

booster_version	payload_mass__kg_
F9 B5 B1048.4	15600
F9 B5 B1049.4	15600
F9 B5 B1051.3	15600
F9 B5 B1056.4	15600
F9 B5 B1048.5	15600
F9 B5 B1051.4	15600
F9 B5 B1049.5	15600
F9 B5 B1060.2	15600
F9 B5 B1058.3	15600
F9 B5 B1051.6	15600
F9 B5 B1060.3	15600
F9 B5 B1049.7	15600

2015 Launch Records

SQL QUERY

```
SELECT  DATE, booster_version, launch_site,
landing__outcome FROM SPACEXTBL WHERE landing__outcome
LIKE 'Failure%' AND booster_version IN (SELECT
booster_version FROM SPACEXTBL WHERE DATE LIKE
'2015%');
```

This SQL query displays the list of failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015.

DATE	booster_version	launch_site	landing__outcome
2015-01-10	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
2015-04-14	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

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

SQL QUERY

```
SELECT landing__outcome, COUNT(landing__outcome) AS
COUNT FROM SPACEXTBL WHERE (landing__outcome LIKE
'Success (ground pad)'  OR landing__outcome LIKE
'Failure (drone ship)') AND (DATE BETWEEN '2010-06-04'
AND '2017-03-20') GROUP BY landing__outcome ;
```

This SQL query Ranks 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

landing__outcome	COUNT
Failure (drone ship)	5
Success (ground pad)	3

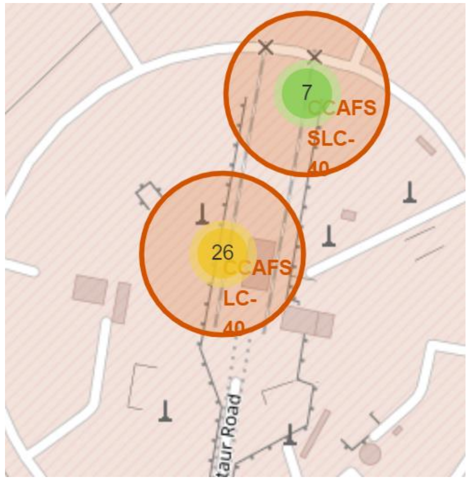
Section 3

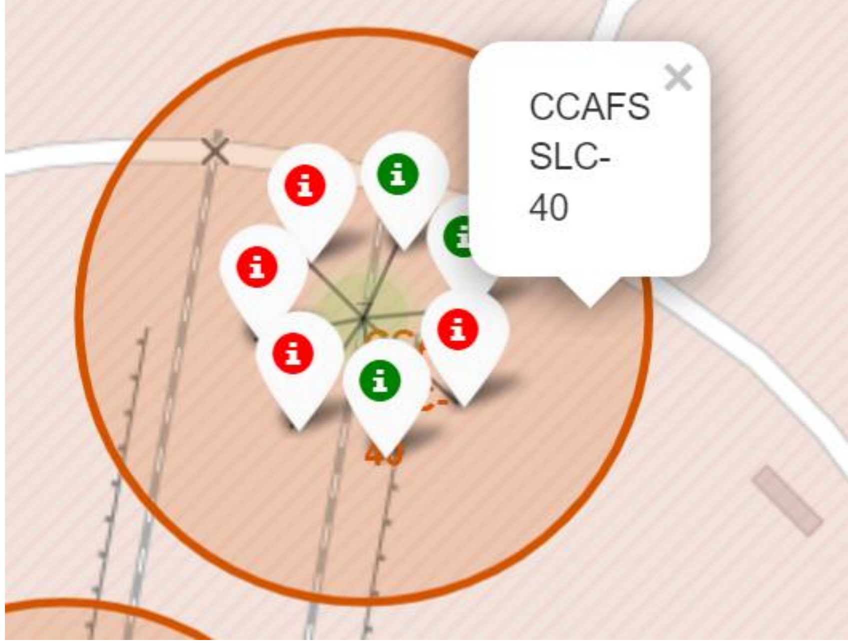
Launch Sites Proximities Analysis

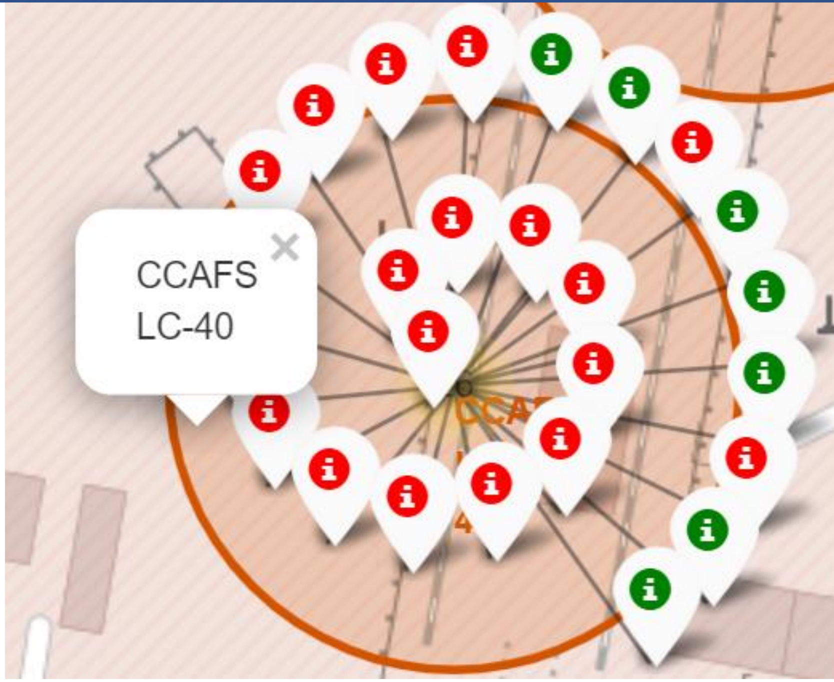
All launch sites' location markers on a global map

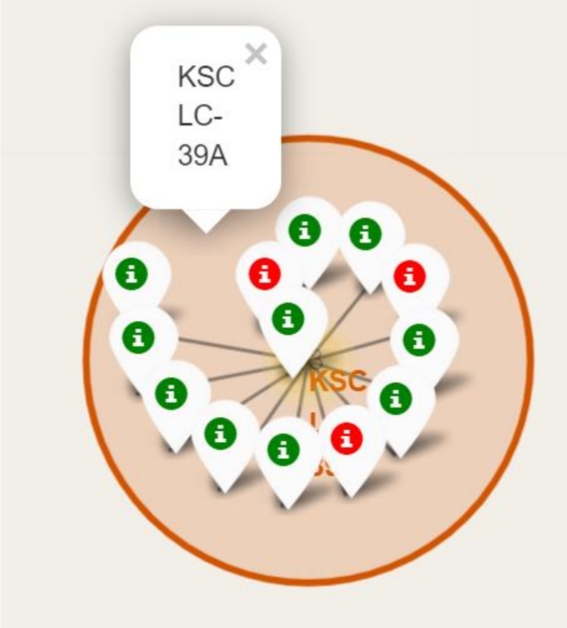
SpaceX launch sites are in the United States costal states of Florida and California.

Florida launch sites (cont.)



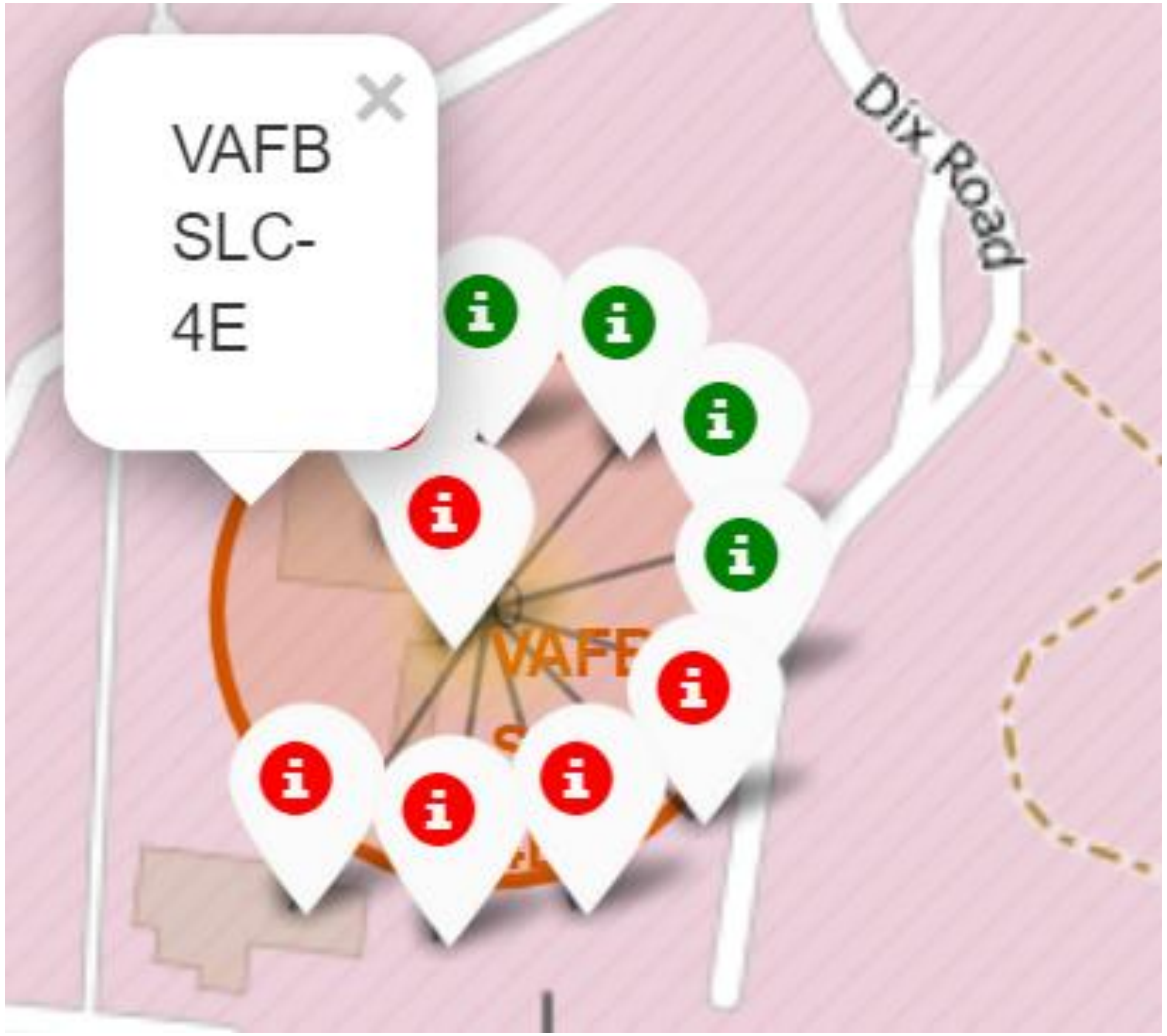




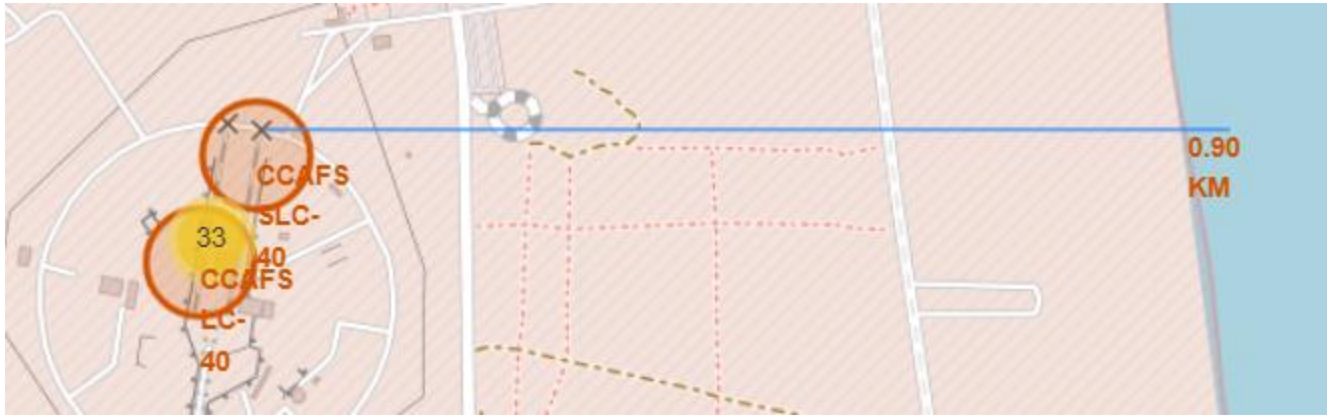


Successful launches in Florida launch sites are marker in **Green** while failure launches are marked in **red**

California launch sites



Successful launches in California launch sites are marker in **Green** while failure launches are marked in **red.**



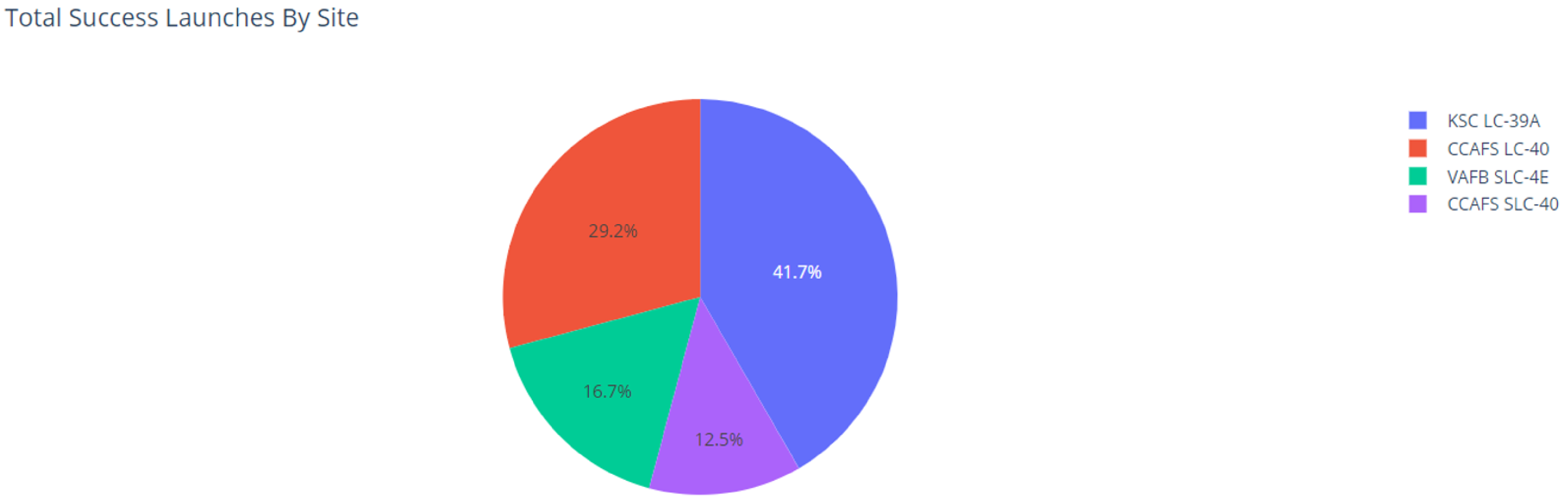
This shows distance to coastline
CCAFS-SLC-40

This shows distance to Highway
CCAFS-SLC-40

Section 4

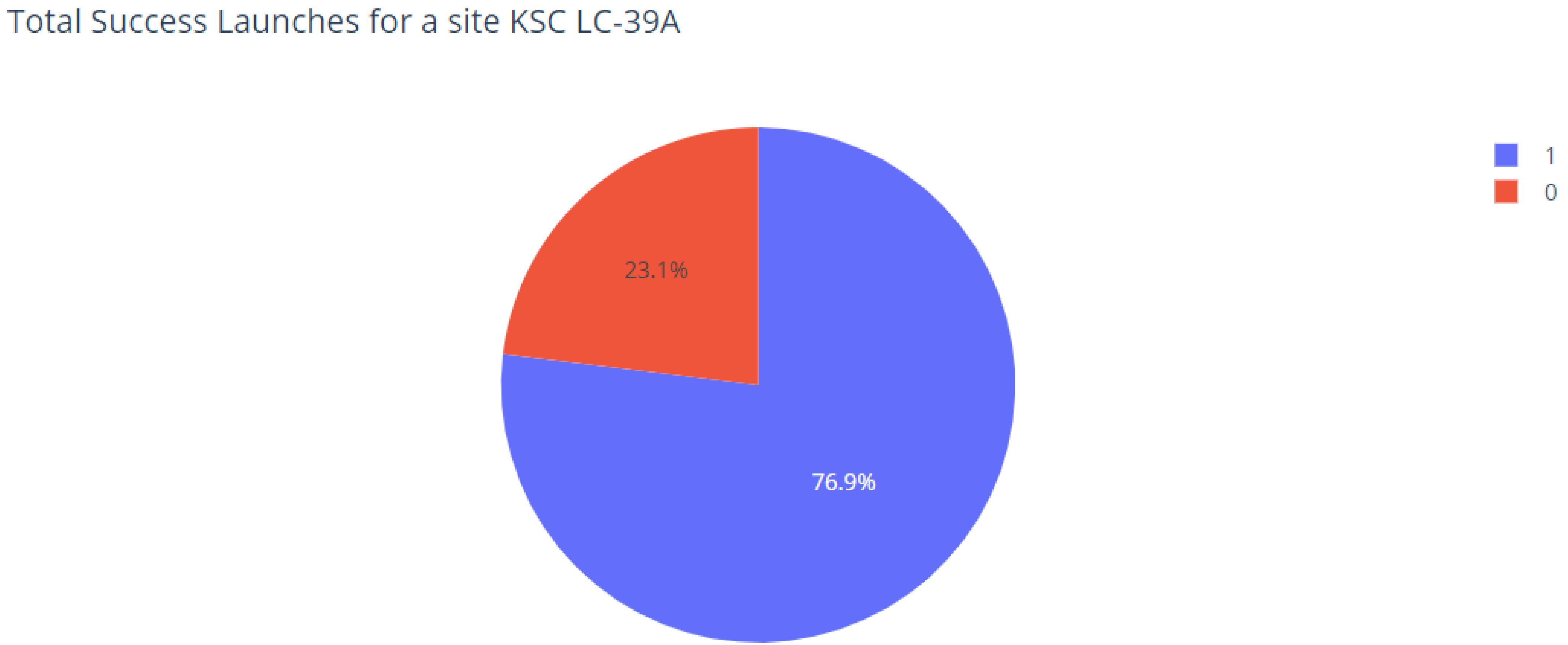
Dashboard with Plotly Dash

Total Success Launches by all sites



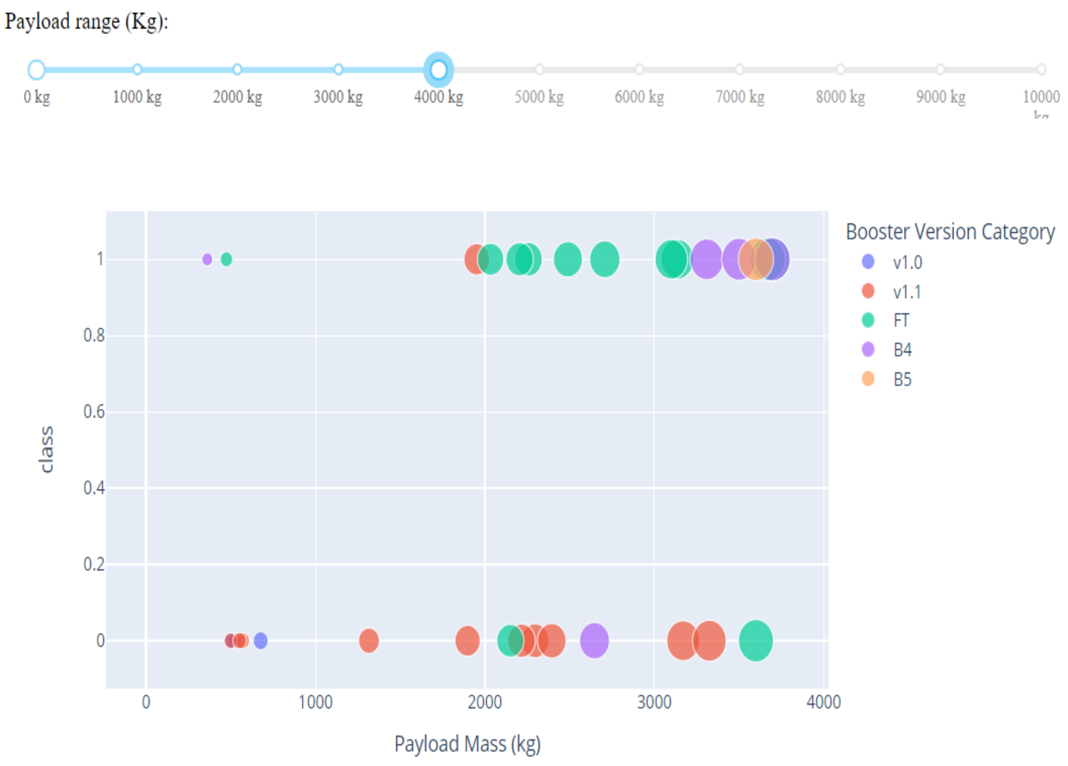
KSC LC-39A has the most successful launches when launch sites are compared

Launch site with Highest success rate

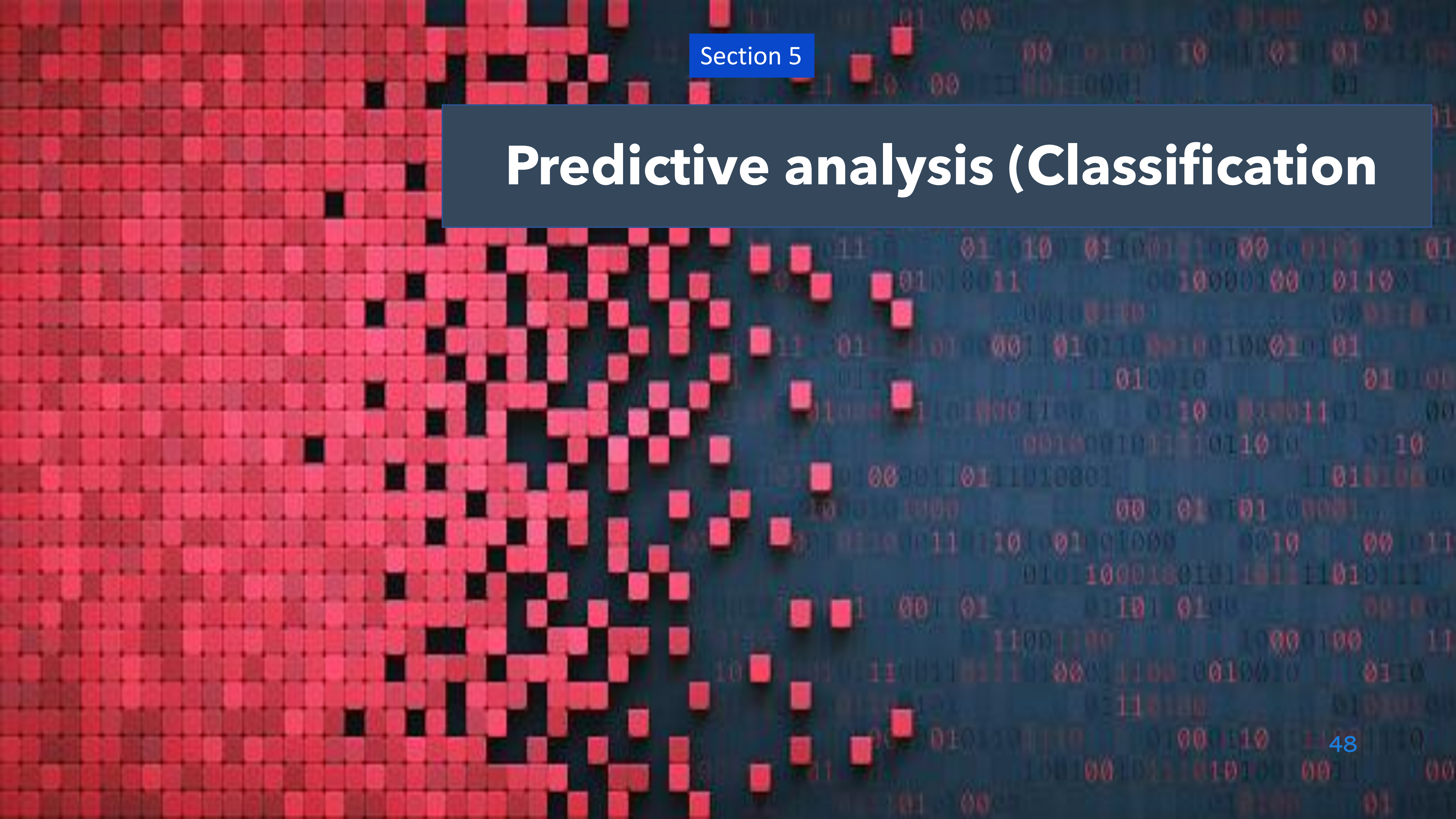


KSC LC-39A has successful rate launches of 76.9% and 23.1% of failure rate.

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



Lower weight payloads have higher success rate compared to heavy weight payloads.

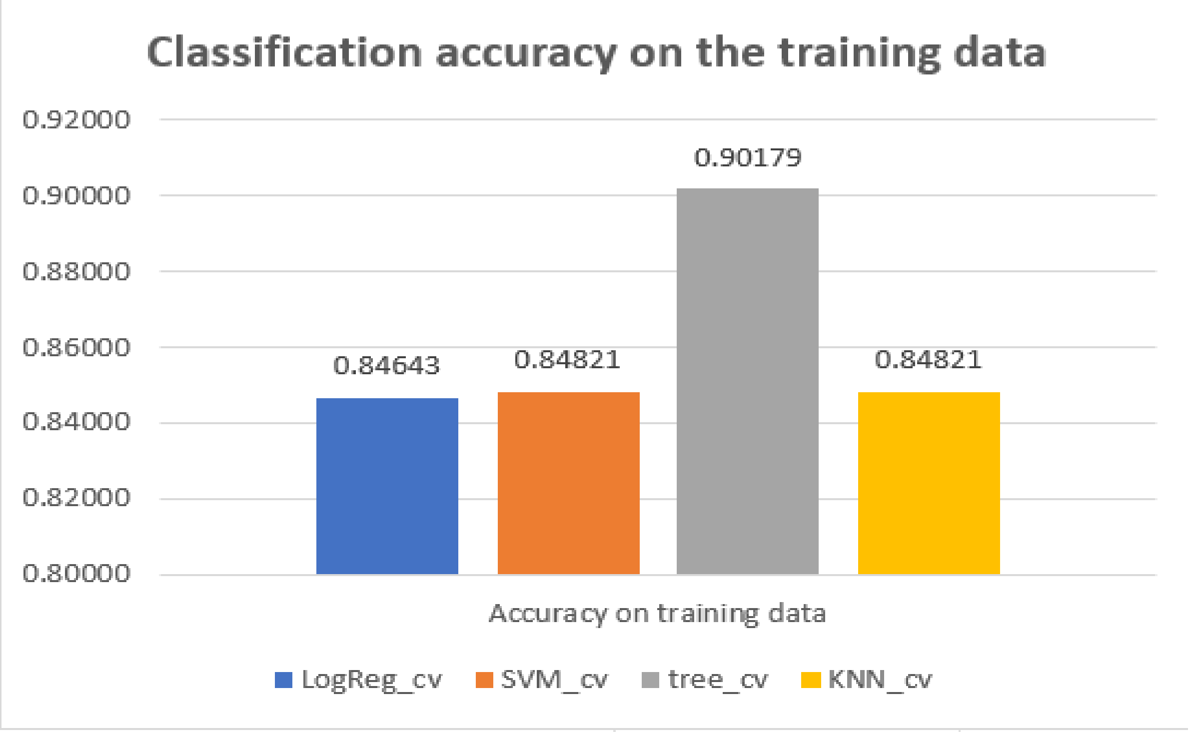


Section 5

Predictive analysis (Classification

Classification Accuracy on training data

Algorithm	Accuracy on training data
LogReg_cv	0.84643
SVM_cv	0.84821
tree_cv	0.90179
KNN_cv	0.84821

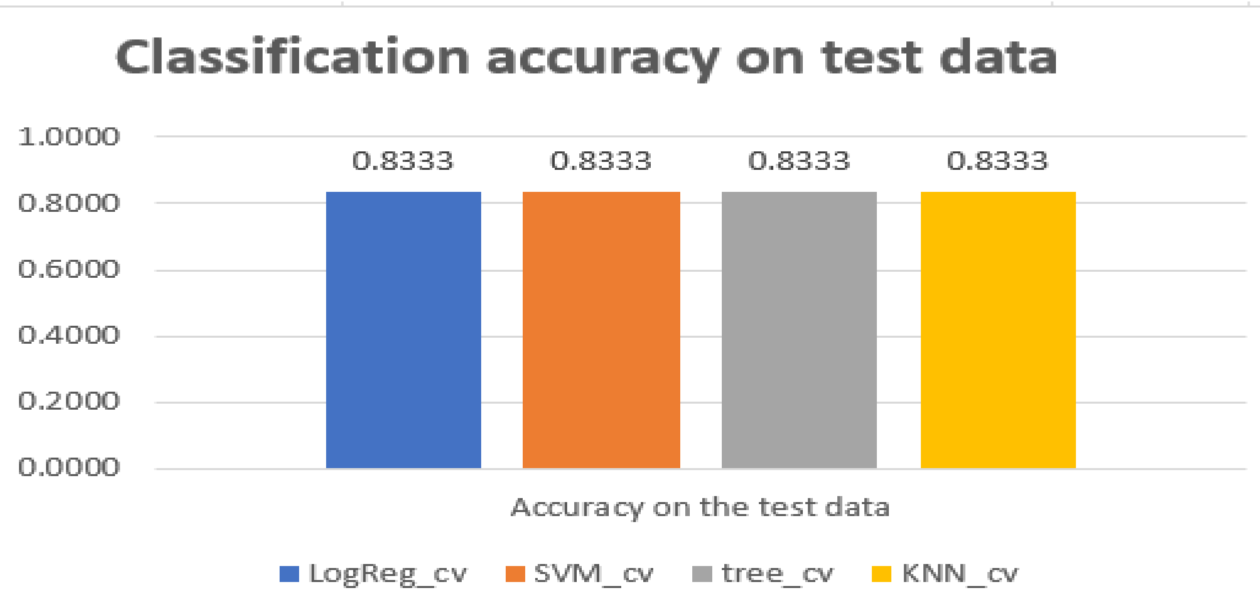


The Decision Tree Classifier wins in accuracy on training data.

```
tuned hpyerparameters :(best parameters)  {'criterion': 'entropy', 'max_depth': 4, 'max_features': 'auto', 'min_samples_leaf': 2, 'min_samples_split': 10, 'splitter': 'random'}
accuracy : 0.9017857142857144
```

Classification Accuracy on test data

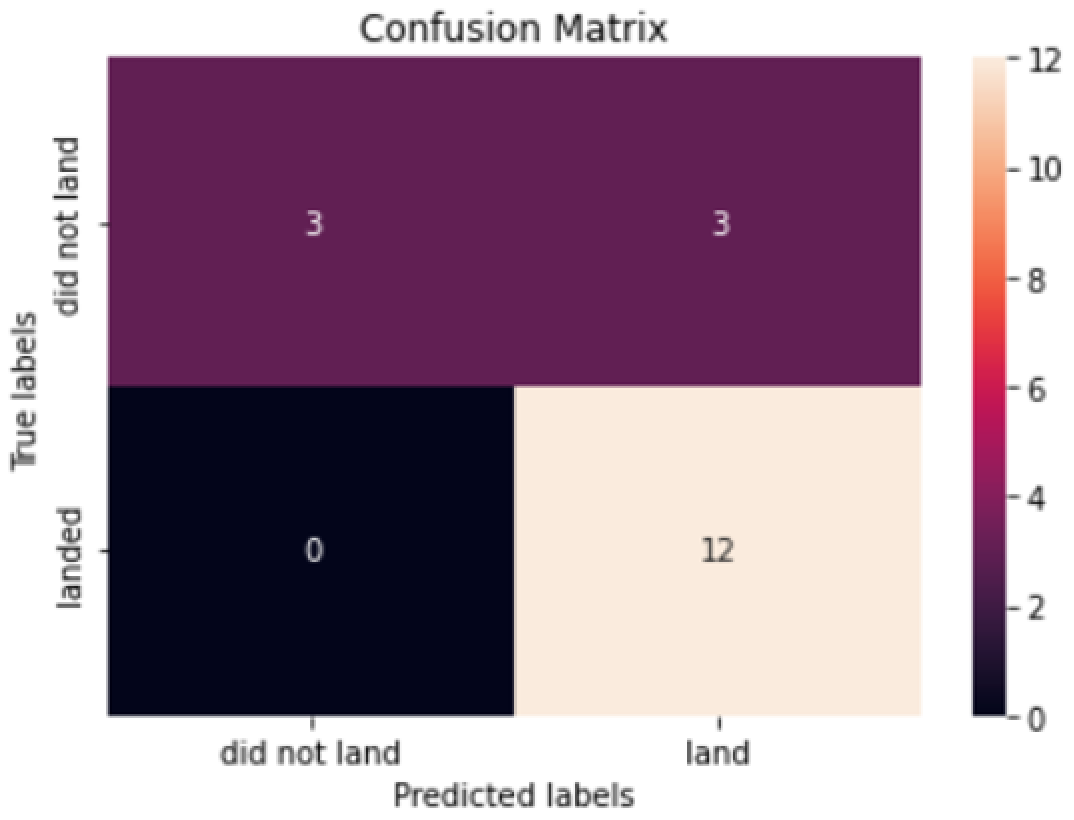
Algorithm	Accuracy on the test data
LogReg_cv	0.8333
SVM_cv	0.8333
tree_cv	0.8333
KNN_cv	0.8333



Using test data to evaluate all classification have same accuracy. However, we can say Decision Tree classifier is the best since it had higher score on the trained data as well.

Confusion matrix for Decision Tree Classifier

```
Confusion matrix for DecisionTreeClassifier
```



The Decision Tree Classifier predicted land correctly, but it predicted 3 as landed while they didn't land in true labels.

However, the confusion matrix of the other three classifiers showed same prediction as Tree Classifier.

0.8333333

Conclusions

- We conclude that HEO,GEO, SSO, ES-L1 orbits have best success rate
- Light payloads are better success rate than the heavier payloads

- When launch sites considered, it has been concluded that KSC LC-39A have the highest success rate of 41.73% compared to all sites.
- All algorithms had the same accuracy score on the test data which was 0.8333, but decision Decision Tree Classifier accuracy on the training data was 0.90179

Thank you!