Data Science Capstone project

Mhd Haikal

9/12/2021

Outline

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

Executive Summary

- **Summary of methodologies:**
  - Data collection
  - Data wrangling
  - EDA with visualization
  - EDA with SQL
  - Interactive amp with folium
  - Dashboard with plotly dash
  - Predictive analysis
- **Summary of all results:**
  - EDA results
  - Prediction results

Introduction

- **Project background and context:**

In this project we are trying to find, clean and process falcon9 data, in order to predict if the Falcon 9 first stage will land successfully. This information could be useful in bidding on SpaceX rocket launch.

- **Problems you want to find answers:**

What variables influence the rocket landing the most.

The impact of each variable on the success rate of landing.

Methodology

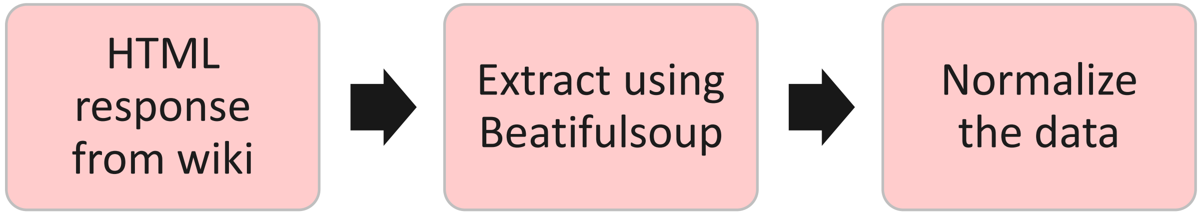
- **Data collection methodology:**
  - SpaceX Rest API
  - Web Scrapping
- **Perform data wrangling:**
  - Dropping irrelevant columns.
  - Calculating important variables.
  - One hot encoding.
- **Perform exploratory data analysis (EDA) using visualization and SQL:**
- **Perform interactive visual analytics using Folium and Plotly Dash:**
- **Perform predictive analysis using classification models:**
  - How to build, tune, evaluate classification models

Methodology

Data collection

- **We gathered SpaceX launch data from the SpaceX Rest API.**

- **We also used web scraping to collect Falcon9 and Falcon heavy launches records from Wikipedia using BeautifulSoup.**



Data collection – SpaceX API

SpaceX Rest API

Get SpaceX data in .JSON

Normalize the data

Getting response from API

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

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

Convert to Json

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

Custom functions to clean

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

Assign to dict then Df

```
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}
lauchDF = pd.DataFrame(launch_dict)
```

Filter the data

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

```
mean = data_falcon9['PayloadMass'].mean()

data_falcon9['PayloadMass'].replace(to_replace =np.nan, value = mean, inplace =True)
```

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

Save data to csv

Github

Data collection – Web scraping

HTML response from wiki

Extract using Beatifulsoup

Normalize the data

Getting response from HTML

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

Beautifulsoup obj

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

Find tables

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

Columns names

```
column_names = []
first_launch_table = first_launch_table.find_all('th')
for row in first_launch_table:
    column = extract_column_from_header(row)
    if column is not None and len(column) >0:
        column_names.append(column)
```

Create dict

```
del launch_dict['Date and time ( )']
launch_dict= dict.fromkeys(column_names)
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']=[]
```

Append data to keys

```
for table_number,table in enumerate(soup.find_all('table',"wikitable plainrowheaders collapsible")):
    # get table row
    for rows in table.find_all("tr"):
```

Convert dict to df

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

Save data to csv

```
df.to_csv('spacex_web_scraped.csv', index=False)
```

Github

Data wrangling

- 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; for example, 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.

- **Data wrangling process using flowcharts:**

Calculate the number of launches on each site → Calculate the number and occurrence of each orbit → Calculate the number and occurrence of mission outcome per orbit type → Create a landing outcome label from Outcome column → Save dataset as CSV

Github

EDA with data visualization

- **The Plotted Chart:**

1. **Scatter Graphs:** Scatter plots show how much one variable is affected by another
   - Flight Number and Payload Mass
   - Class and Launch Site
   - Flight Number and Launch Site
   - Payload and Launch Site
   - FlightNumber and Orbit type
   - Payload and Orbit type
2. **Bar Graph:** A bar diagram makes it easy to compare sets of data between different groups at a glance
   - the relationship between success rate of each orbit type.
3. **Line Graph:** Line graphs are useful in that they show data variables and trends very clearly and can help to make predictions about the results of data not yet recorded
   - the launch success yearly trend.

Github

EDA with SQL

- Display the names of the unique launch sites in the space mission
- Display 5 records where launch sites begin with the string 'KSC'
- 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 where the succesful landing outcome in drone ship was acheived.
- List the names of the boosters which have success in ground pad 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 records which will display the month names, succesful landing_outcomes in ground pad ,booster versions, launch_site for the months in year 2017
- Rank the count of successful landing_outcomes between the date 2010-06-04 and 2017-03-20 in descending order.

Build an interactive map with Folium

- a circle at NASA Johnson Space Center's coordinate with a popup label showing its name and an icon as text label
- A circle object on each Launch site coordinates with a popup label with its name
- We assigned to each launch in the df a marker (Green if it was successful and Red if not) on the map in a MarkerCluster()
- We calculated the distance from the Launch Site to other landmarks Lines are drawn between the launch site and a landmark

Github

Build a Dashboard with Plotly Dash

- **a Launch Site Drop-down Input Component**
- **a callback function to render success-pie-chart based on selected site dropdown**

  - display relative proportions of multiple classes of data

- **a Range Slider to Select Payload**
- **a callback function to render the success-payload-scatter-chart scatter plot**

  - It shows the relationship between two variables.
  - Observation and reading are straightforward.

Github

Predictive analysis (Classification)

- **Data Preparation:**
  - Load the dataframe into Numpy
  - Standardize the data
  - Split the data to test/train
- **Building Models:**
  - Logistic regression
  - Support vector machine object
  - decision tree classifier
  - k nearest neighbors
- **Evaluating Models:**
  - Choosing the best hyperparameters
  - Calculating the accuracy for each model
  - Confusion matrix

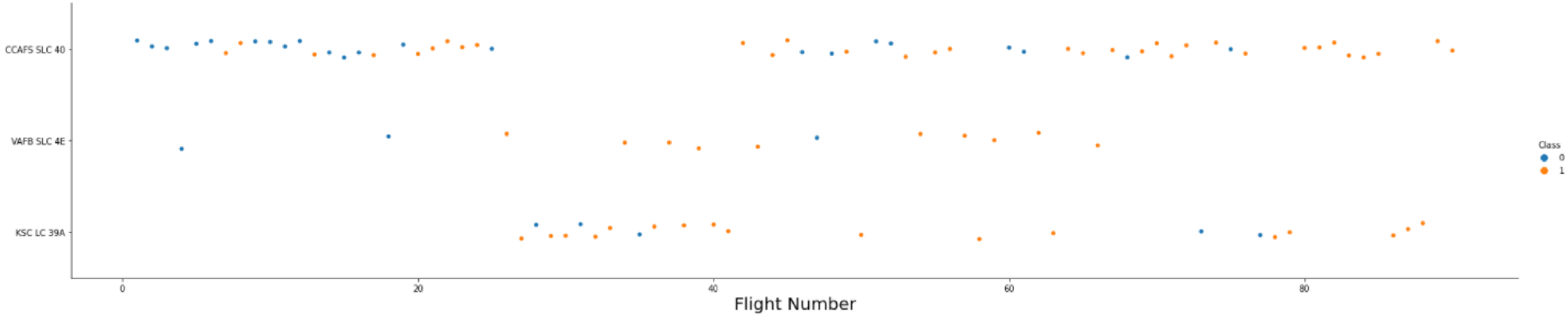
Github

Results

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

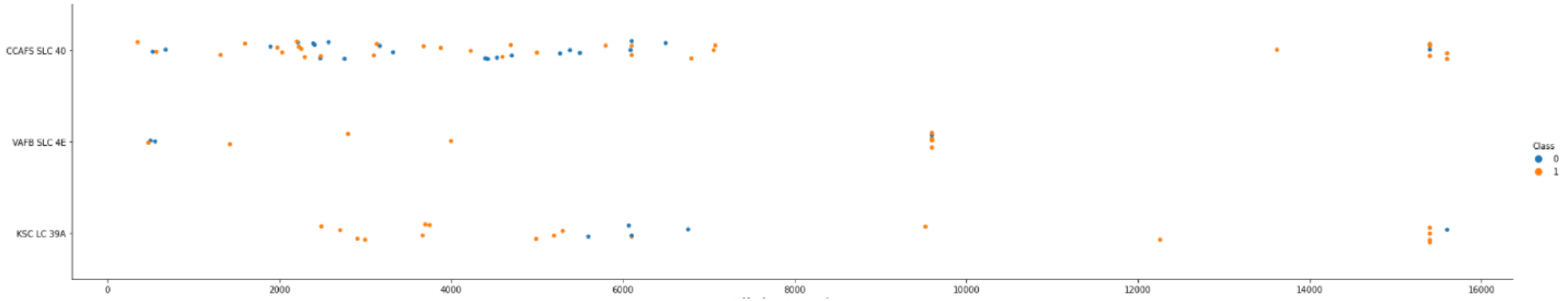
EDA with Visualization

Flight Number vs. Launch Site



The success rate in a launch site is higher with a higher flights number

Payload vs. Launch Site

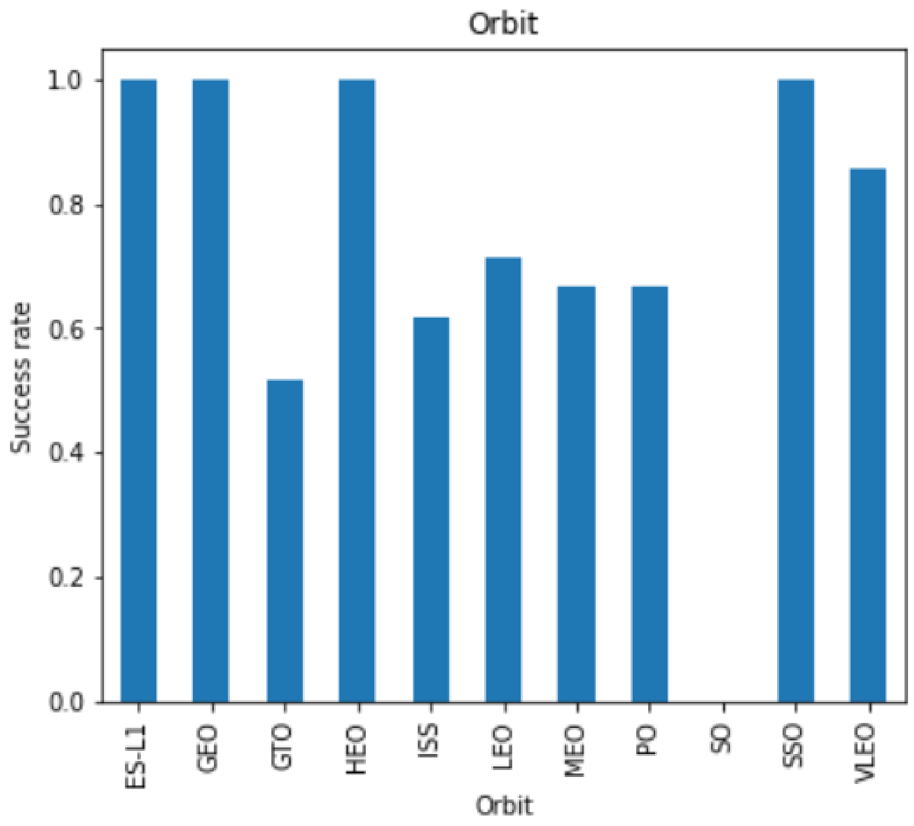


The success rate for CCAFS SLC 40 is higher with a greater payload.

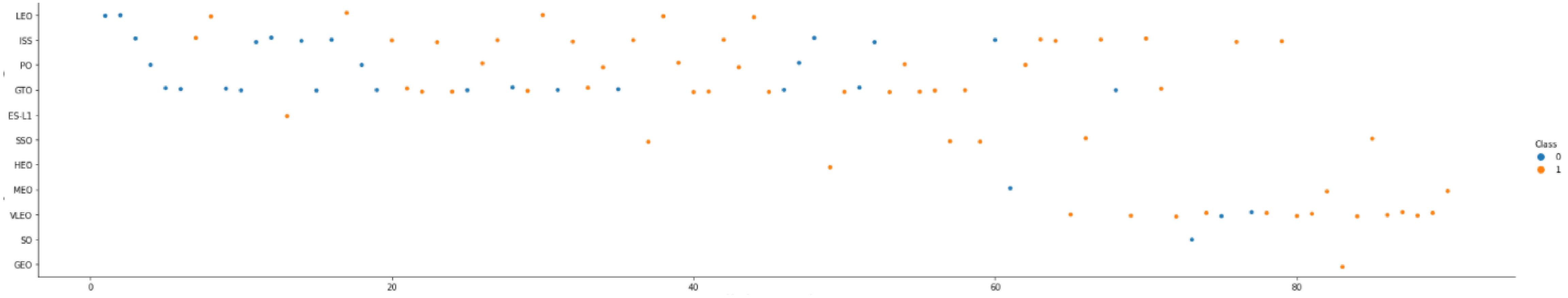
No pattern for other sites.

Success rate vs. Orbit type

ES-L1, Geo, HEO and SSO orbits have the highest success rate

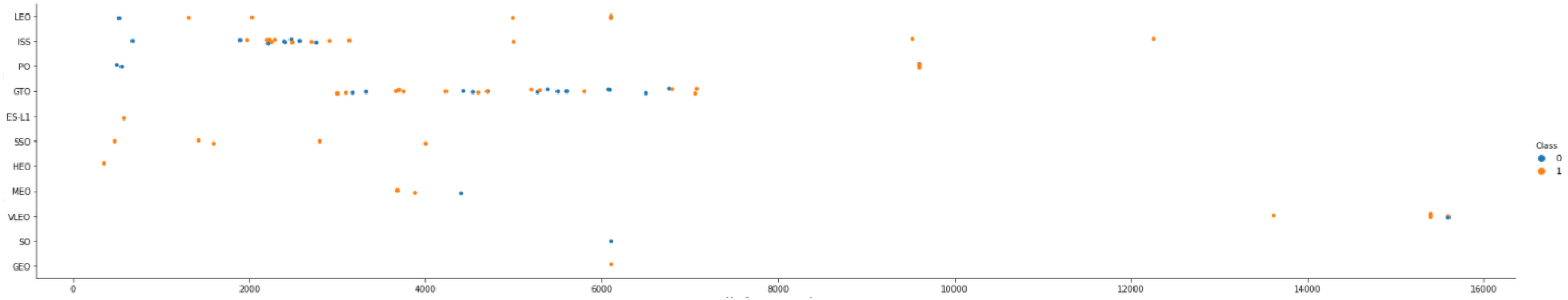


Flight Number vs. Orbit type



We can see that with most of the orbits the higher the flight number the better the success rate especially for LEO

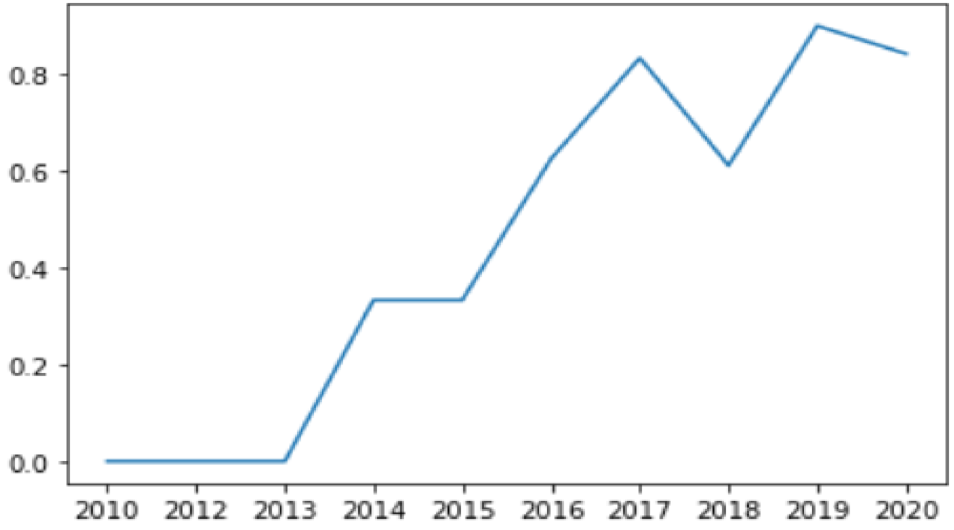
Payload vs. Orbit type



We can see that GTO suffers with heavy payload but for LEO, ISS and PO heavy payload is good.

Launch success yearly trend

The success rate has been increasing since 2013



EDA with SQL

All launch site names

Query:

```
SELECT DISTINCT(launch_site) FROM SPACEX;
```

Result:

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

Explanation:

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

Launch site names begin with `CCA`

Query:

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

Explanation:

TOP 5 means that it will only show 5 records from tblSpaceX and LIKE keyword with the words ‘KSC%’ suggests that the Launch_Site name must start with KSC.

Result:

DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__outcome
2010-04-06	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-08-12	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)
2012-08-10	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-01-03	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-12	22:41:00	F9 v1.1	CCAFS LC-40	SES-8	3170	GTO	SES	Success	No attempt

Total payload mass

Query:

```
SELECT SUM(payload_mass__kg_) FROM SPACEX
WHERE customer = 'NASA (CRS)';
```

Explanation:

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

Result:

1
22007

Average payload mass by F9 v1.1

Query:

```
SELECT AVG(payload_mass__kg_) FROM SPACEX
WHERE booster_version = 'F9 v1.1';
```

Result:

1
3676.666666

Explanation:

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

First successful ground landing date

Query:

```
select min(DATE) from SPACEX
where landing__outcome ='Success (ground pad)';
```

Result:

1
2017-01-05

Explanation:

the function MIN gives the earliest date in the column Date
The WHERE clause filters the dataset to only perform calculations on Landing_Outcome ‘Success (ground pad)’

Successful drone ship landing with payload between 4000 and 6000

Query:

```
select booster_version from SPACEX
where landing__outcome ='Success (drone ship)' and payload_mass__kg_ between 4001 and 5999;
```

Explanation:

Selecting only Booster_Version The WHERE clause filters the dataset to Landing_Outcome = Success (drone ship) The AND clause specifies additional filter conditions The between key words select the values between 2 numbers, in this case 4001 and 5999

Result:

booster_version
F9 FT B1022
F9 FT B1031.2

Total number of successful and failure mission outcomes

Query:

```
select landing__outcome, count(landing__outcome) as count from SPACEX
group by landing__outcome;
```

Result:

landing__outcome	COUNT
Controlled (ocean)	1
Failure	1
Failure (drone ship)	2
Failure (parachute)	2
No attempt	12
Success	18
Success (drone ship)	5
Success (ground pad)	4

Explanation:

the group by for ‘Landing_outcome’ combined with the count keyword gives us the total for each unique landing outcome

Boosters carried maximum payload

Query:

```
select booster_version from spacex
where payload_mass__kg_ = (select max(payload_mass__kg_) from spacex);
```

Explanation:

The where keyword specifies a condition, the max gives the maximum value of the Payload_mass_kg_ column.

Result:

booster_version
F9 B5 B1048.4
F9 B5 B1049.4
F9 B5 B1049.5
F9 B5 B1060.2
F9 B5 B1058.3

2015 launch records

Query:

```
select month(DATE), landing__outcome, booster_version, launch_site from spacex
where landing__outcome = 'Failure (drone ship)' and year(DATE) = 2015;
```

Result:

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

Explanation:

The where keyword specifies a condition, the and keyword to add another condition, the year keyword gives the year from the date

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

Query:

```
select landing__outcome, count(landing__outcome) as count from spacex
where DATE between '2010-06-04' and '2017-03-20'
group by landing__outcome
having landing__outcome like 'Success%'
order by count desc;
```

Result:

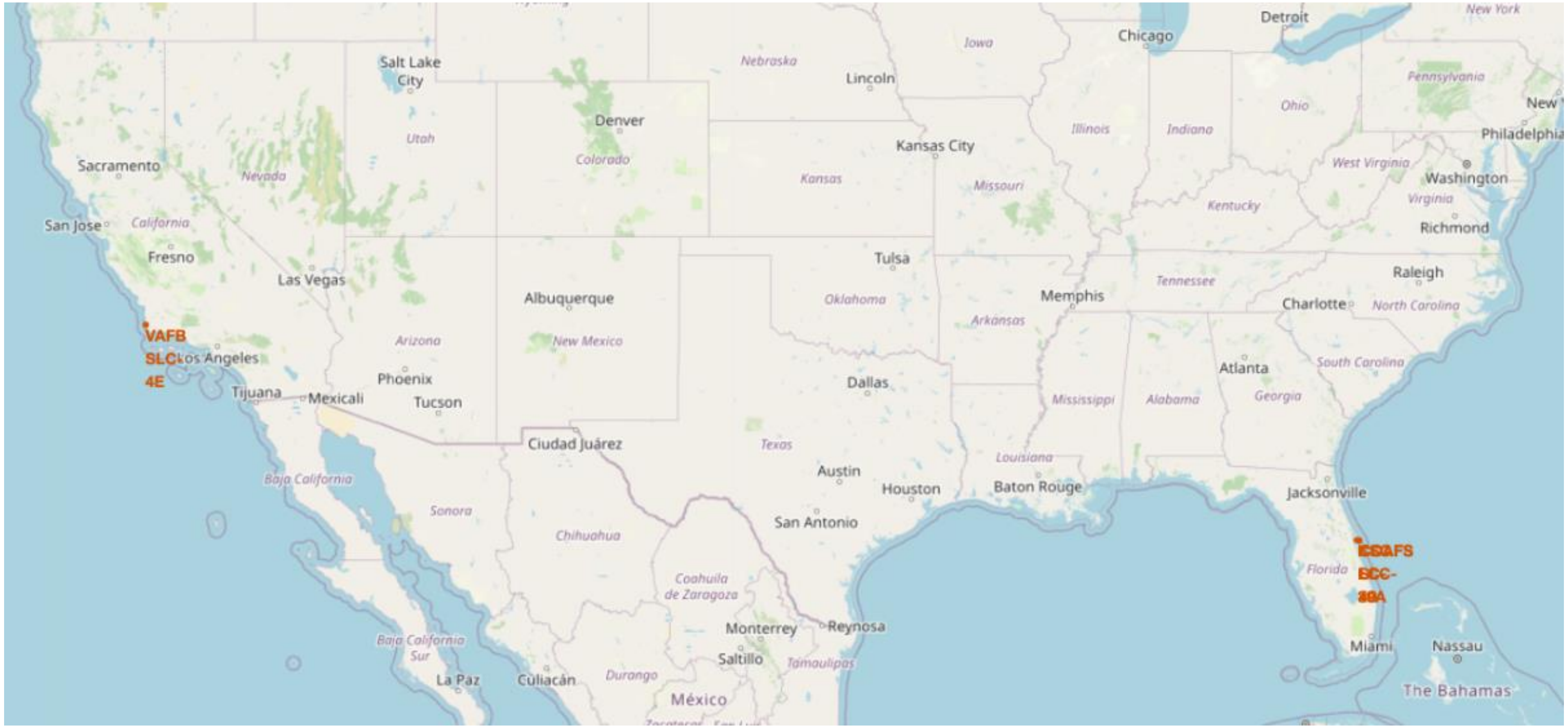
landing__outcome	COUNT
Success (drone ship)	2
Success (ground pad)	2

Explanation:

Function COUNT counts records in column WHERE filters data, order by keyword orders the data in either descending or ascending, in our case desc

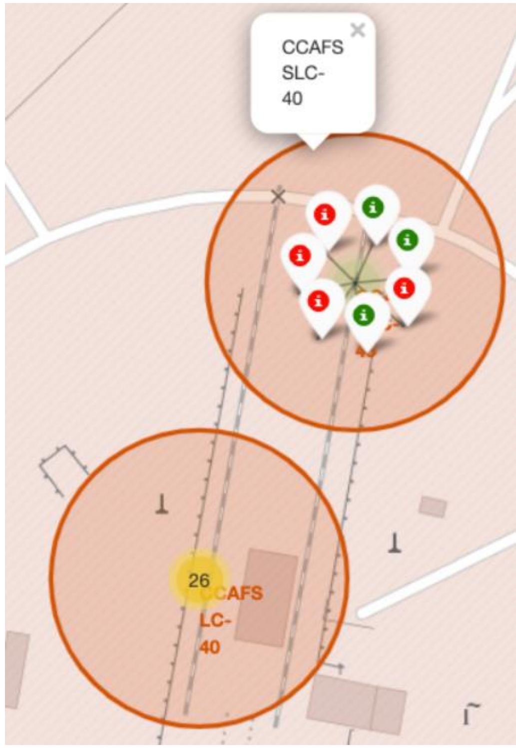
Interactive map with Folium

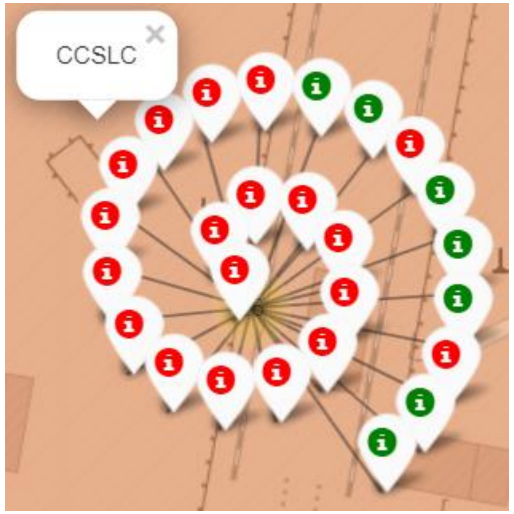
all launch sites' location markers on a global map

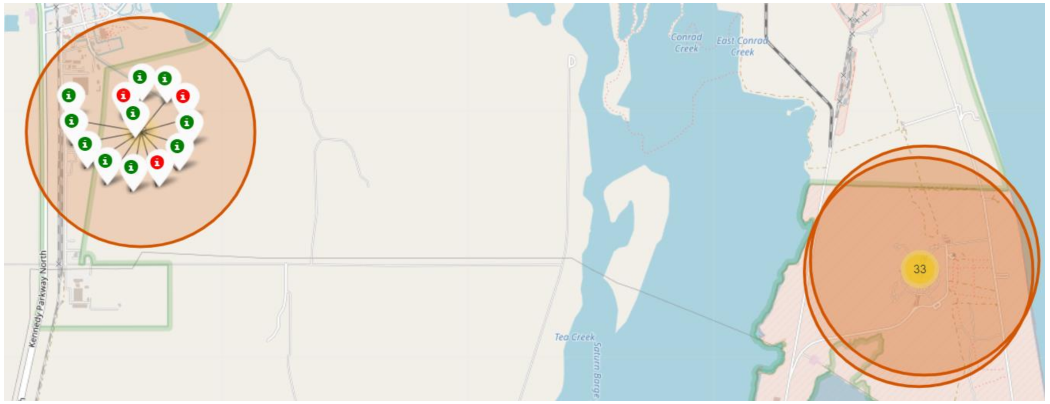


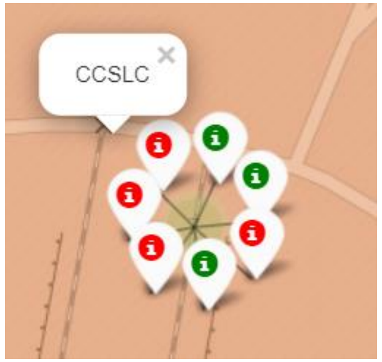
We can see that the SpaceX launch sites are in the USA near the coasts.

color-labeled launch records on the map

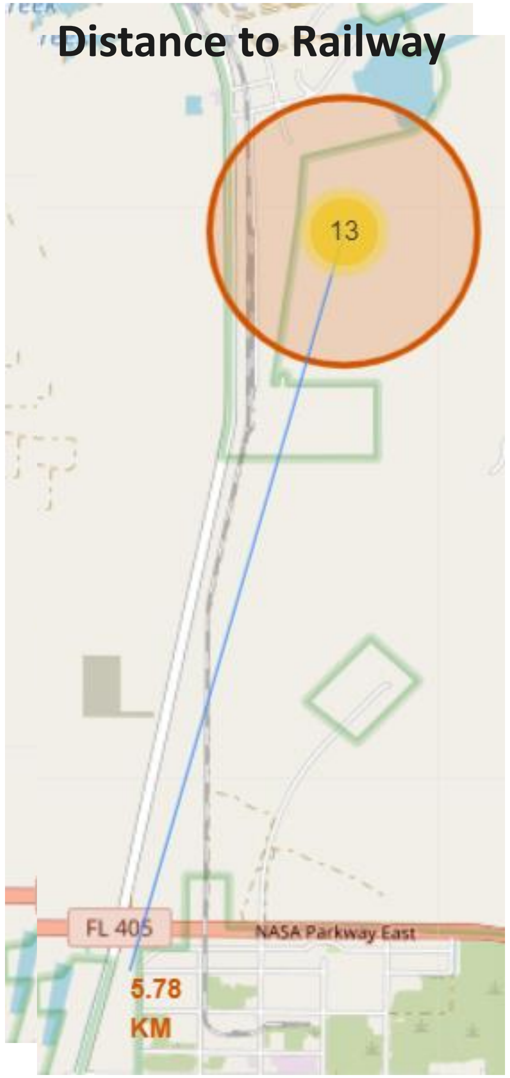


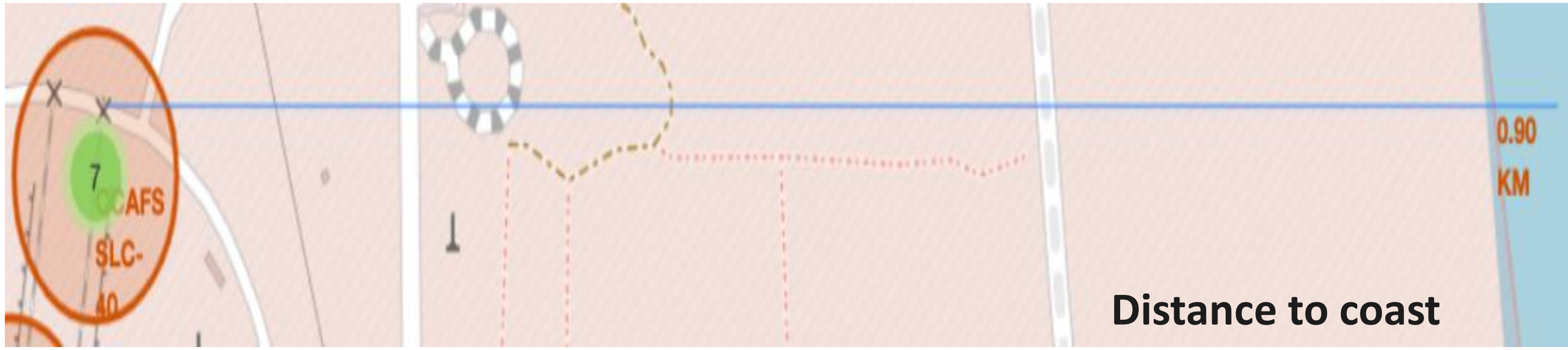






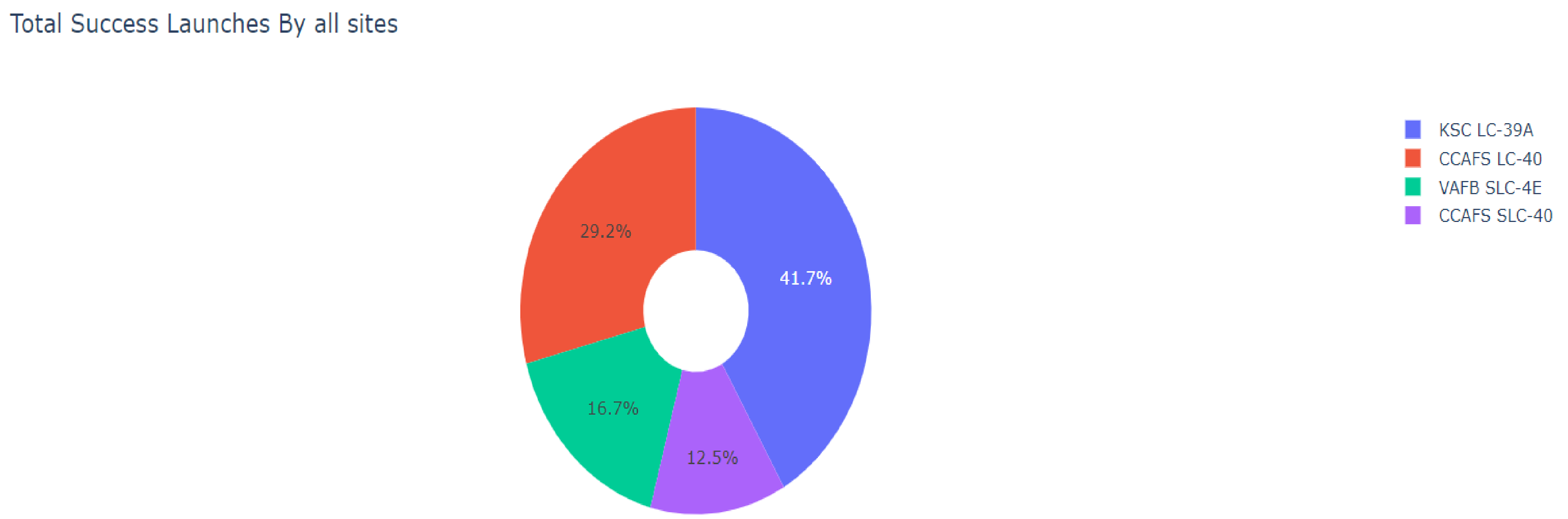
a selected launch site to its proximities





Build a Dashboard with Plotly Dash

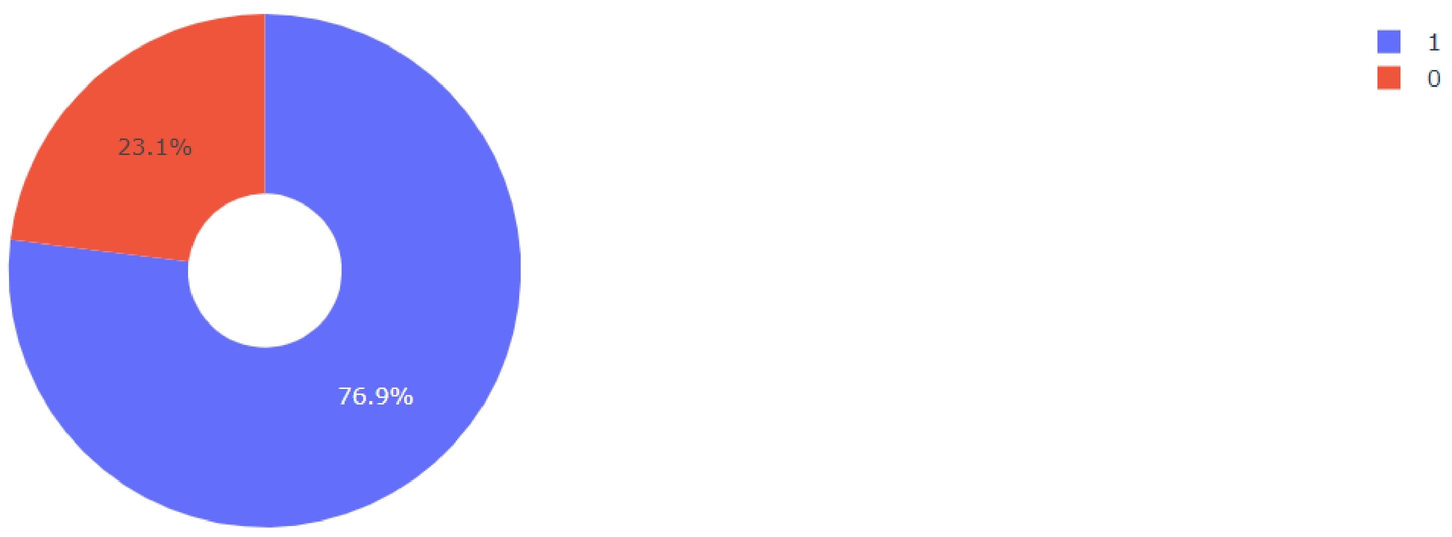
Success rates for all launch sites



We can see that KSC LC-39A had the most successful launches

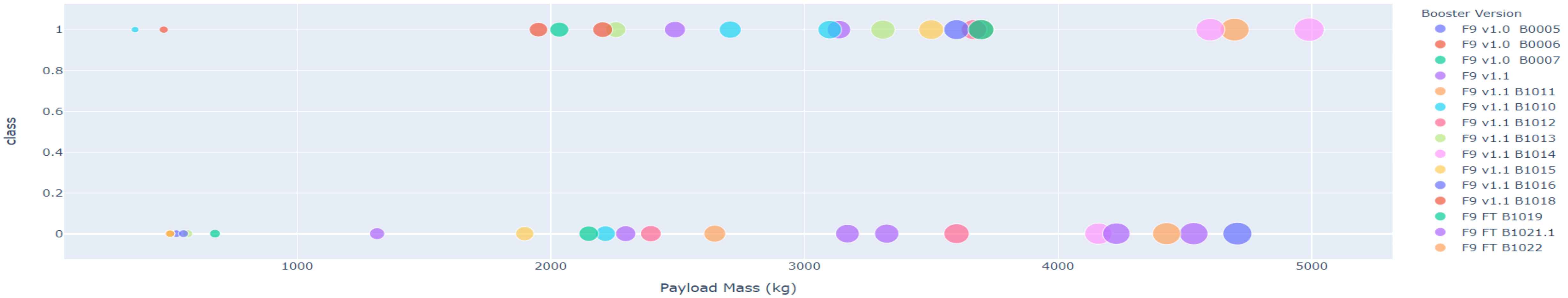
The highest success rate Launch site

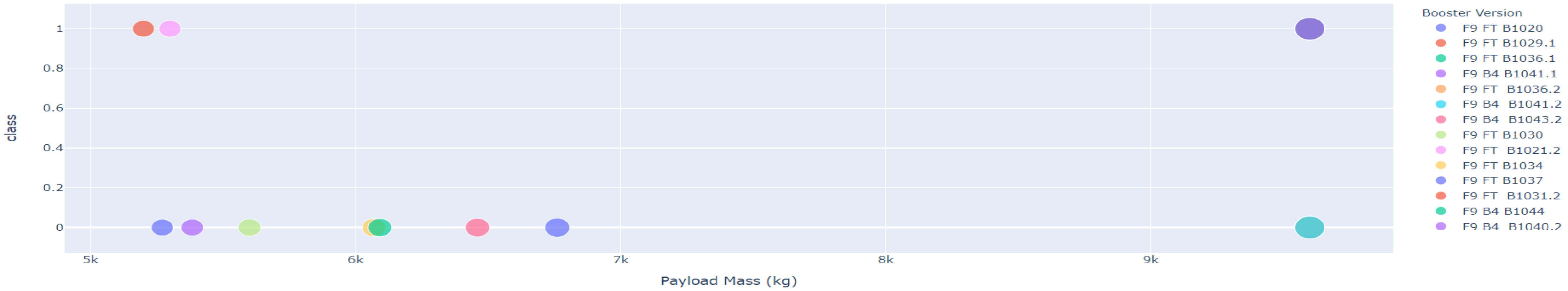
Total Success Launches for site KSC LC-39A



KSC LC-39A has a 76.9% success rate

Payload vs. Launch Outcome scatter plot





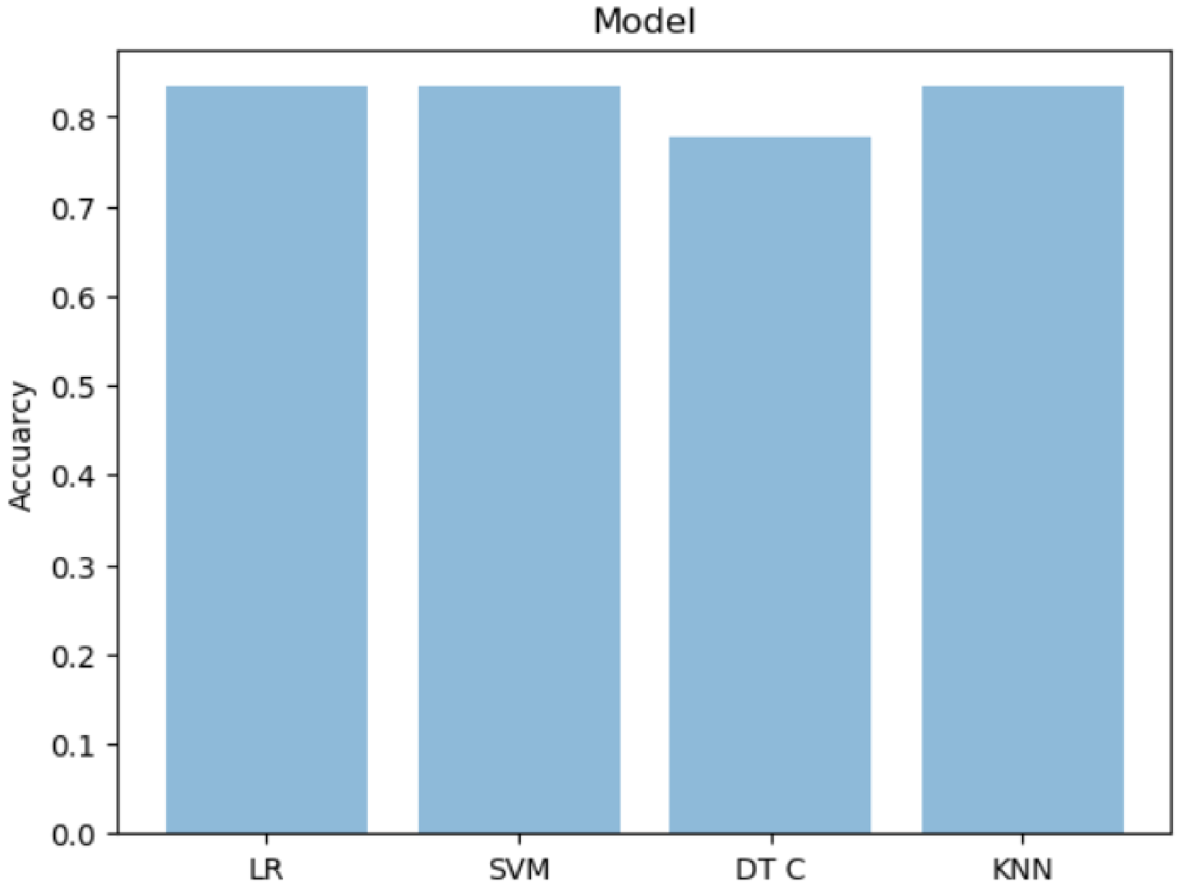
the success rates for low weighted payloads is higher

Predictive analysis (Classification)

Classification Accuracy

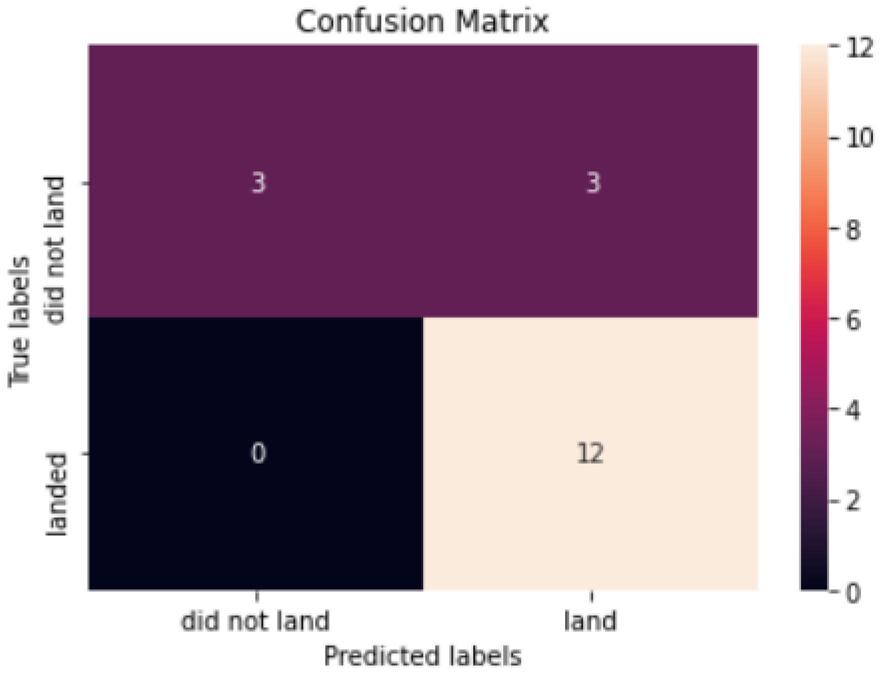
All the methods had similar results.

decision tree classifier had the lowest accuracy of 0.777



Confusion Matrix

Examining the confusion matrix, we see that Tree can distinguish between the different classes. We see that the major problem is false positives.



CONCLUSION

- The Decision Tree Classifier Algorithm had the worst accuracy.
- KSC LC-39A had the most successful launches
- ES-L1, Geo, HEO and SSO orbits have the highest success rate
- The success rate is increasing each year.