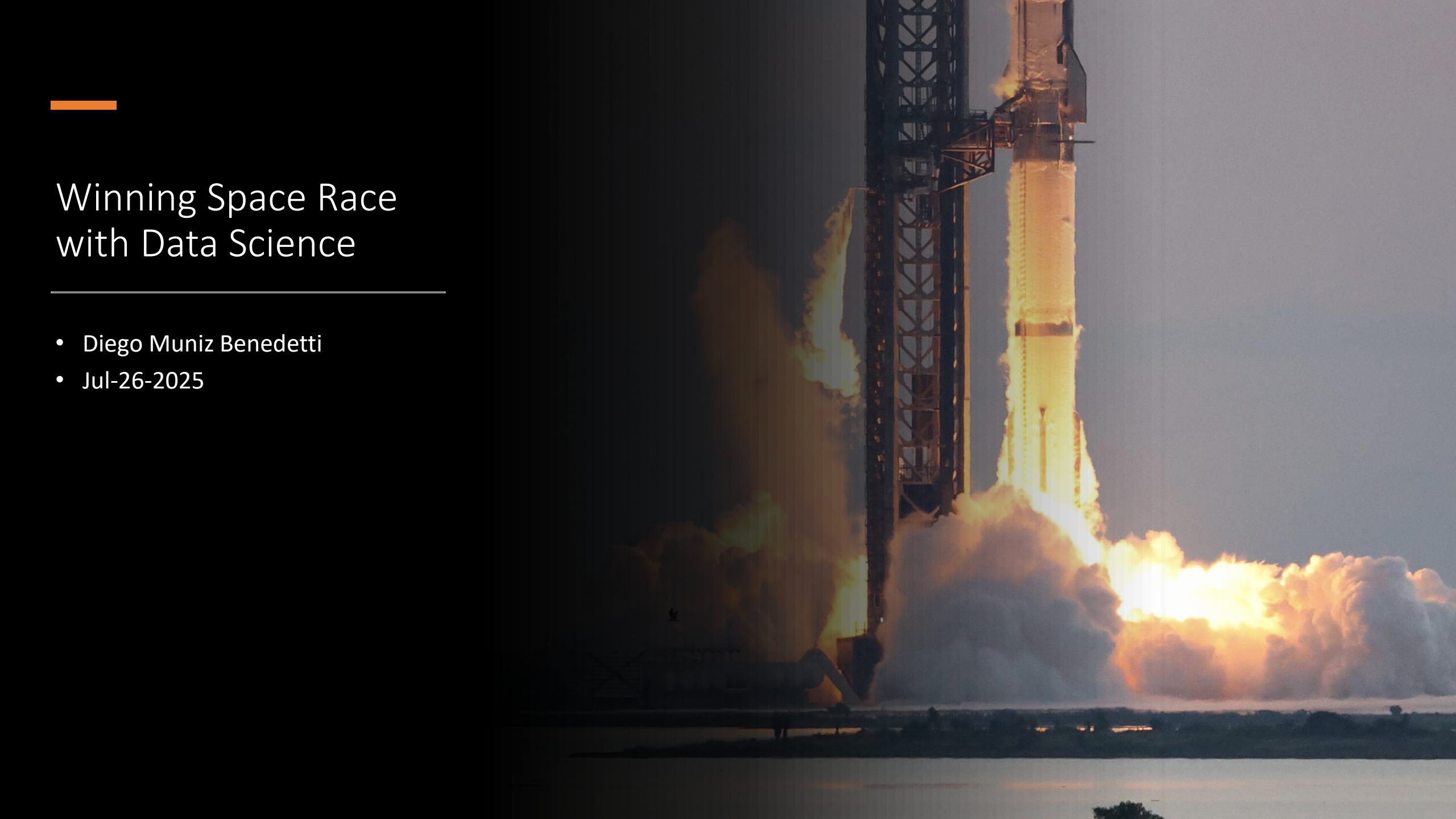
Winning Space Race with Data Science

- Diego Muniz Benedetti

- Jul-26-2025



Outline

- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion

Executive Summary

- This project explores the full cycle of data analysis using Space-X launching data and applies following methodologies:
  - ❑ Data collection
  - ❑ Data wrangling
  - ❑ Exploratory data analysis
  - ❑ Descriptive data analysis
  - ❑ Interactive visual analytics
  - ❑ Predictive data analysis

- Results show a comprehensive insight on:
  - ❑ Nature and statistics of the raw data
  - ❑ Positional distribution of the data
  - ❑ Selective classification of the data
  - ❑ Predictions using classification models

Introduction

- SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage.
- Therefore, if we can determine if the first stage will land, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch.

- In this project we mainly investigate factors determining if the launching will be successful.

Methodology

- Data collection methodology:
  - Making GET requests to SpaceX REST API
  - Web Scrapping Wikipedia
- Perform data wrangling
  - Exploring the data by checking types and counting values
  - Defining new attributes for classification
- Perform interactive visual analytics using Folium and Plotly Dash
  - Plotting different types of charts and graphics using user friendly frameworks
- Perform exploratory data analysis (EDA) using visualization and SQL
  - Plotting different types of charts and graphics for data analysis
  - Running queries to explore relations and singularities
- Perform predictive analysis using classification models
  - Using grid search to evaluate different classification models to predict success.

Data Collection – SpaceX API

- Requesting and parsing the SpaceX launch data:

```
static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_call_spacex_api.json'
response=requests.get(static_json_url)
data=pd.json_normalize(response.json(), sep=',')
data.head(2)
```

```
#Global variables
BoosterVersion = []
PayloadMass = []
Orbit = []
LaunchSite = []
Outcome = []
Flights = []
GridFins = []
Reused = []
Legs = []
LandingPad = []
Block = []
ReusedCount = []
Serial = []
Longitude = []
Latitude = []
```

```
#separation functions
getBoosterVersion(data)
BoosterVersion[0:5]
getLaunchSite(data)
getPayloadData(data)
getCoreData(data)
```

```
#Dataframe structure
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}
```

```
#DataFrame creation:
df=pd.DataFrame(launch_dict)
df.head(3)
```

Data Collection - Scraping

- Web scraping Wikipedia:

```
# use requests.get() and BeautifulSoup
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"
response=requests.get(static_url)
bs_obj=BeautifulSoup(response.text)
bs_obj.title
html_tables=bs_obj.find_all('table')
```

```
# Assign the result to tables
html_tables=bs_obj.find_all('table')
first_launch_table = html_tables[2]
column_names = []
for i in list(first_launch_table.find_all('th')):
    column_names.append(extract_column_from_header(i))
column_names =[x for x in column_names if x is not None]
column_names =[x for x in column_names if x is not '']
print(first_launch_table)
```

```
#Creating the dataframe
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'] = []
# Added some new columns
launch_dict['Version Booster']=[]
launch_dict['Booster landing']=[]
launch_dict['Date']=[]
launch_dict['Time']=[]
```

```
soup=bs_obj
extracted_row = 0
for table_number,table in enumerate(soup.find_all('table',"wikitable plainrowheaders collapsible")):
    for rows in table.find_all("tr"):
        if rows.th:
            if rows.th.string:
                flight_number=rows.th.string.strip()
                flag=flight_number.isdigit()
        else:
            flag=False
        row=rows.find_all('td')
        if flag:
            extracted_row += 1
            launch_dict['Flight No.']=flight_number
            datatimelist=date_time(row[0])
            date = datatimelist[0].strip(',')
            launch_dict['Date']=date
            time = datatimelist[1]
            launch_dict['Time']=time
            bv=booster_version(row[1])
            if not(bv):
                bv=row[1].a.string
            launch_dict['Version Booster']=bv
            launch_site = row[2].a.string
            launch_dict['Launch site'] = launch_site
            payload = row[3].a.string
            launch_dict['Payload'] = payload
            payload_mass = get_mass(row[4])
            launch_dict['Payload mass'] = payload_mass
            orbit = row[5].a.string
            launch_dict['Orbit'] = orbit
            try:
                customer = row[6].a.string
            except:
                customer ="Not found"
            launch_dict['Customer'] = customer
            launch_outcome = list(row[7].strings)[0]
            launch_dict['Launch outcome'] = launch_outcome
            booster_landing = landing_status(row[8])
            launch_dict['Booster landing']=booster_landing
            dict_list.append(launch_dict)
```

Data Wrangling

- Data exploration, analysis and modification:

Data exploration → Data analysis → Features engineering

```
df.head(10)
```

```
df.isnull().sum()/len(df)*100
```

```
df.dtypes
```

```
# number of launches on each site
df['LaunchSite'].value_counts()
```

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

```
# number and occurrence of each orbit
df['Orbit'].value_counts()
```

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

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

```
Outcome
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: count, dtype: int64
```

```
for i,outcome in enumerate(landing_outcomes.keys()):
    print(i,outcome)
```

```
0 True ASDS
1 None None
2 True RTLS
3 False ASDS
4 True Ocean
5 False Ocean
6 None ASDS
7 False RTLS
```

We create a set of outcomes where the second stage did not land successfully:

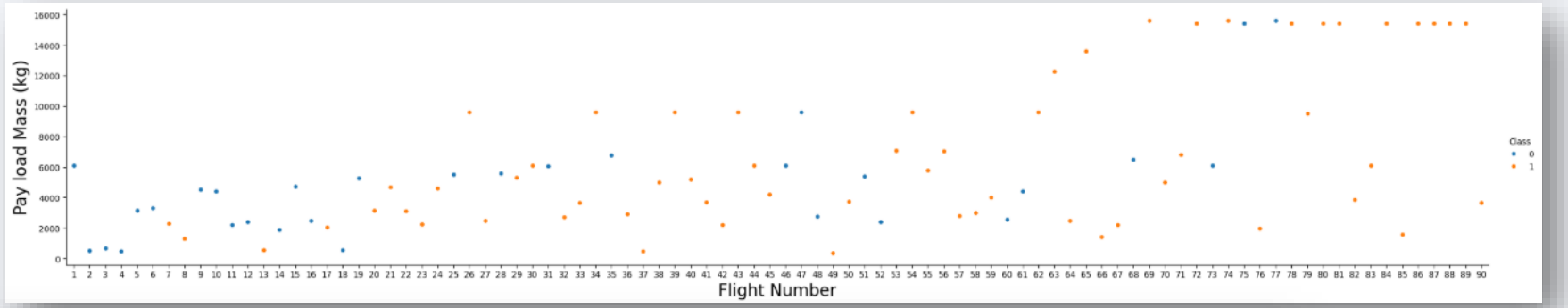
```
bad_outcomes=set(landing_outcomes.keys()[[1,3,5,6,7]])
bad_outcomes
```

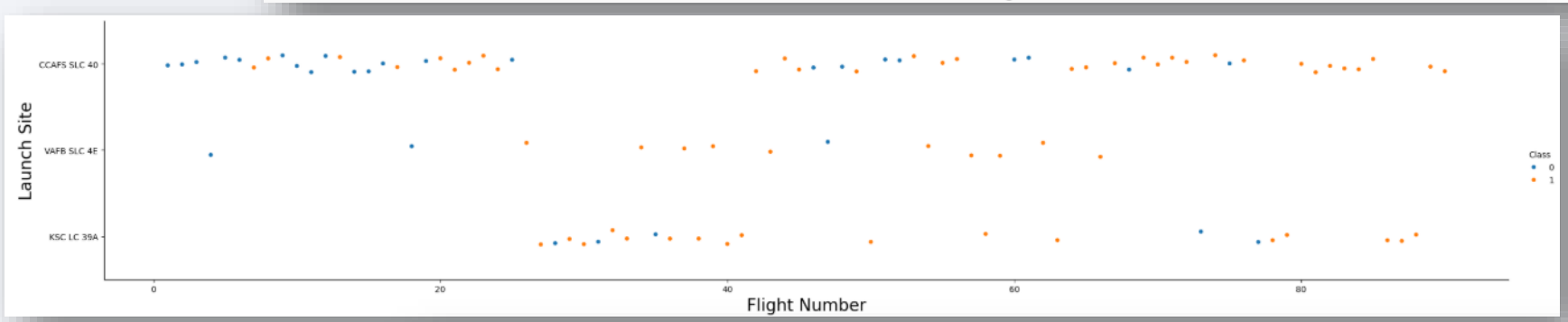
```
{'False ASDS', 'False Ocean', 'False RTLS', 'None ASDS', 'None None'}
```

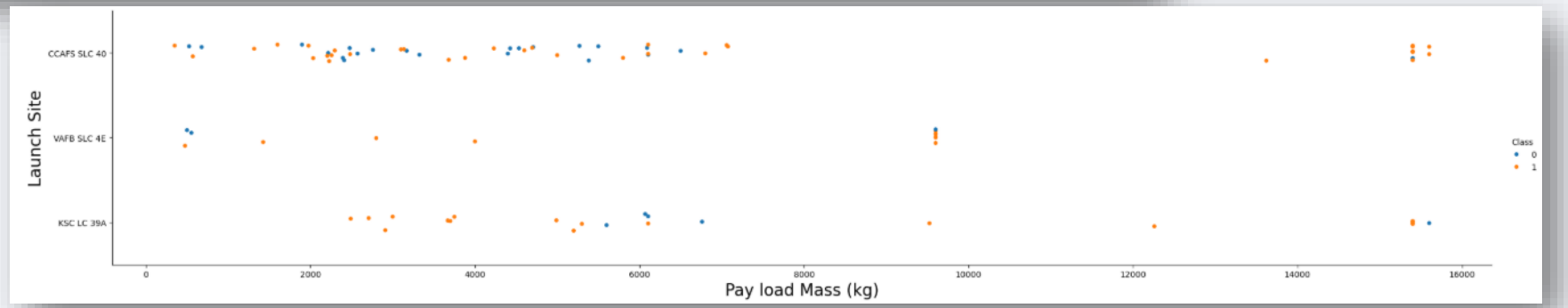
```
# landing_class = 0 if bad_outcome
# landing_class = 1 otherwise
df['Class'] = df.apply(lambda x: 1 if str(x['Outcome'])[:4] == 'True' else 0, axis=1)
```

EDA with Data Visualization (1/3)

- Scatter charts:

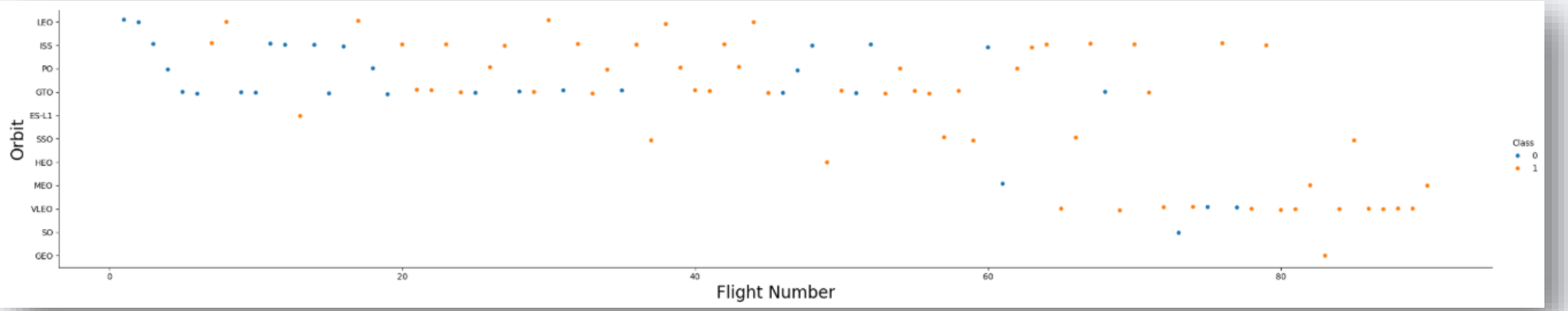


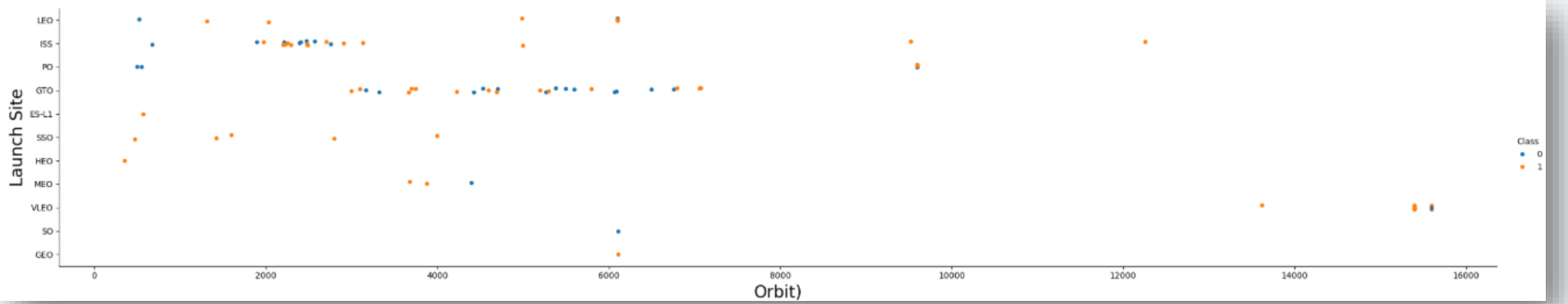




EDA with Data Visualization (2/3)

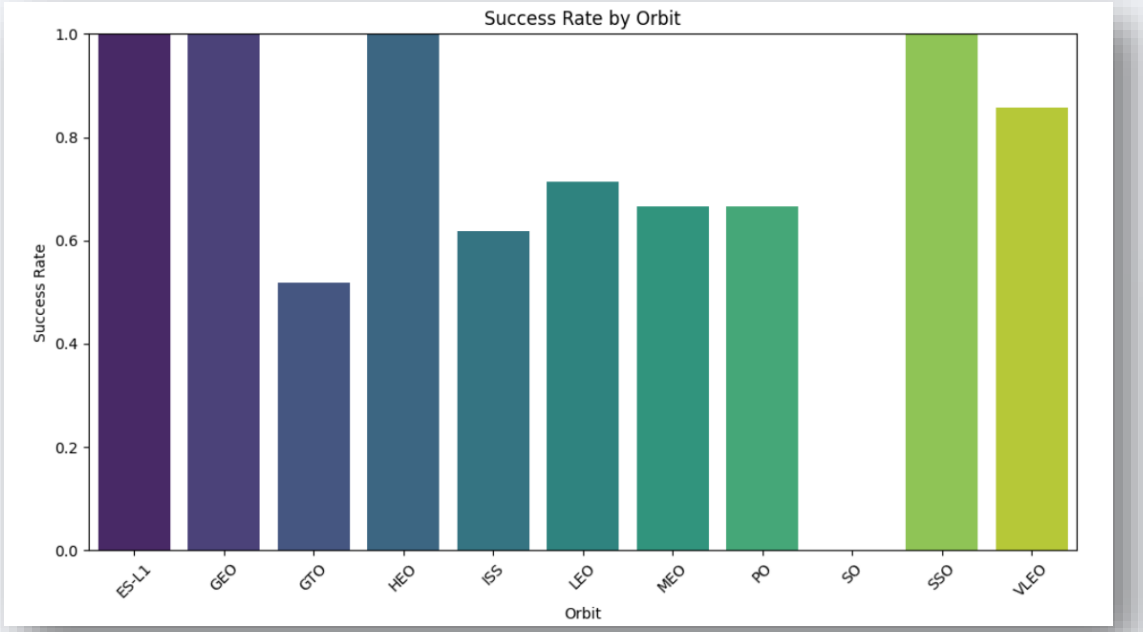
- Scatter charts:

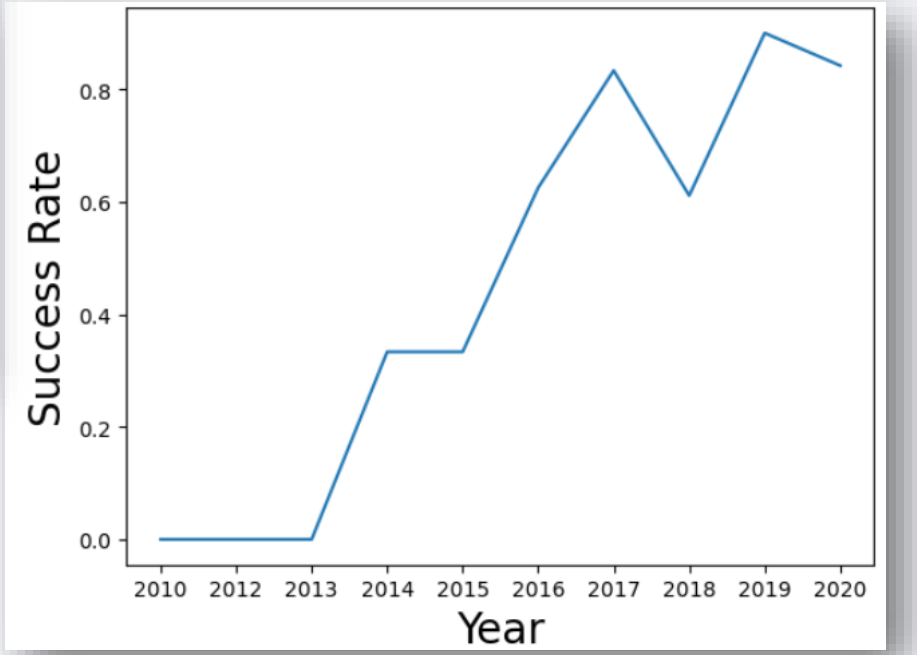




EDA with Data Visualization (3/3)

- Bars and lines:





EDA with SQL (1/2)

- SQL queries:

```
%sql SELECT DISTINCT Launch_Site FROM SPACEXTBL;
```

```
 * sqlite:///my_data1.db
Done.
```

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

```
%sql SELECT * FROM SPACEXTBL WHERE Launch_Site LIKE 'CCA%';
```

```
 * sqlite:///my_data1.db
Done.
```

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)

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE Customer LIKE 'NASA (CRS)';
```

```
 * sqlite:///my_data1.db
Done.
```

SUM(PAYLOAD_MASS__KG_)
45596

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE Booster_Version LIKE 'F9 v1.1';
```

```
 * sqlite:///my_data1.db
Done.
```

AVG(PAYLOAD_MASS__KG_)
2928.4

```
%sql SELECT MIN(Date) FROM SPACEXTBL WHERE Landing_Outcome LIKE 'Success (ground pad)';
```

```
 * sqlite:///my_data1.db
Done.
```

MIN(Date)
2015-12-22

```
%sql SELECT Booster_Version  FROM SPACEXTBL WHERE (Landing_Outcome LIKE 'Success (drone ship)') AND (PAYLOAD_MASS__KG_ > 4000)AND (PAYLOAD_MASS__KG_ < 6000);
```

```
 * sqlite:///my_data1.db
Done.
```

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

```
%sql SELECT COUNT(*) FROM SPACEXTBL WHERE (Mission_Outcome LIKE 'Success');
```

```
 * sqlite:///my_data1.db
Done.
```

COUNT(*)
98

```
%sql SELECT COUNT(*) FROM SPACEXTBL WHERE (Mission_Outcome NOT LIKE 'Success');
```

```
 * sqlite:///my_data1.db
Done.
```

COUNT(*)
3

EDA with SQL (2/2)

- SQL queries:

```
%sql SELECT Booster_Version, PAYLOAD_MASS__KG_ FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL);
```

```
 * sqlite:///my_data1.db
Done.
```

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

```
%%sql
SELECT
  CASE SUBSTR(Date, 6, 2)
    WHEN '01' THEN 'January'
    WHEN '02' THEN 'February'
    WHEN '03' THEN 'March'
    WHEN '04' THEN 'April'
    WHEN '05' THEN 'May'
    WHEN '06' THEN 'June'
    WHEN '07' THEN 'July'
    WHEN '08' THEN 'August'
    WHEN '09' THEN 'September'
    WHEN '10' THEN 'October'
    WHEN '11' THEN 'November'
    WHEN '12' THEN 'December'
  END AS Month_Name,
  Landing_Outcome,
  Booster_Version,
  Launch_Site
FROM SPACEXTBL
WHERE
  SUBSTR(Date, 1, 4) = '2015'
  AND LOWER(Landing_Outcome) LIKE '%failure%'
  AND LOWER(Landing_Outcome) LIKE '%drone ship%';
```

```
 * sqlite:///my_data1.db
Done.
```

Month_Name	Landing_Outcome	Booster_Version	Launch_Site
January	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
April	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

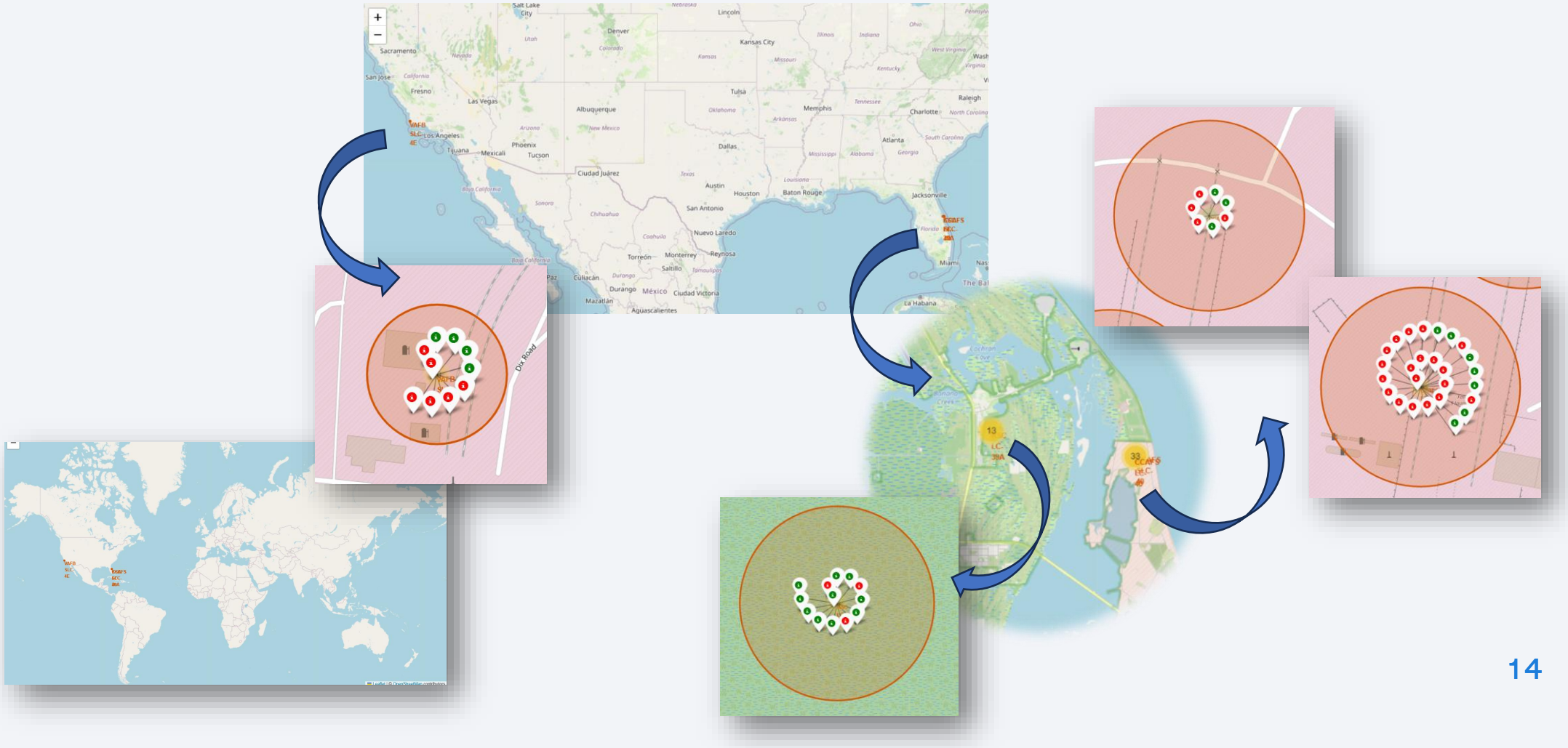
```
%%sql
SELECT
  Landing_Outcome,
  COUNT(*) AS Outcome_Count
FROM SPACEXTBL
WHERE Date >= '2010-06-04' AND Date <= '2017-03-20'
GROUP BY Landing_Outcome
ORDER BY Outcome_Count DESC;
```

```
 * sqlite:///my_data1.db
Done.
```

Landing_Outcome	Outcome_Count
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

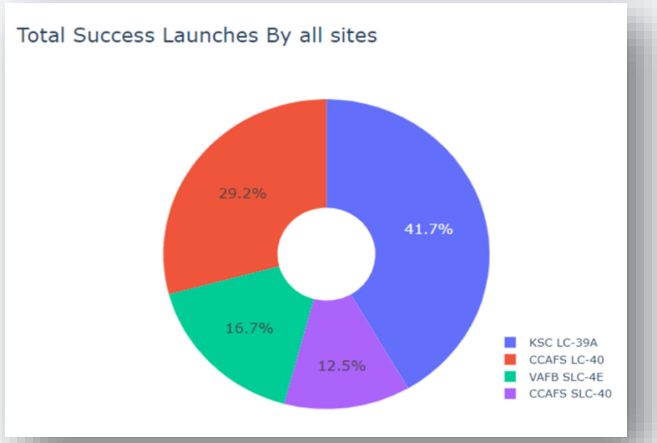
Interactive Map with Folium

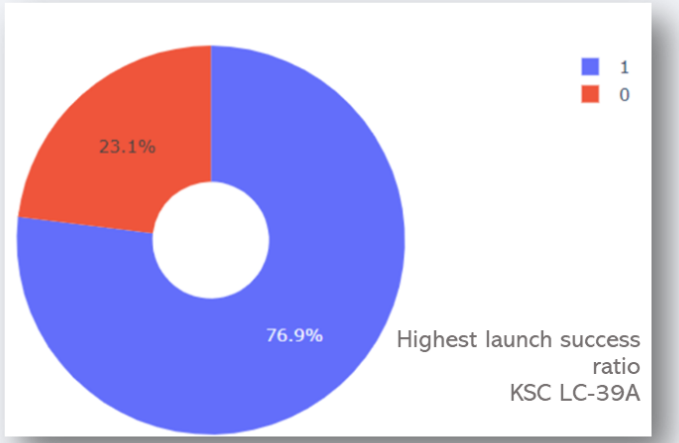
- Positional map of launching sites

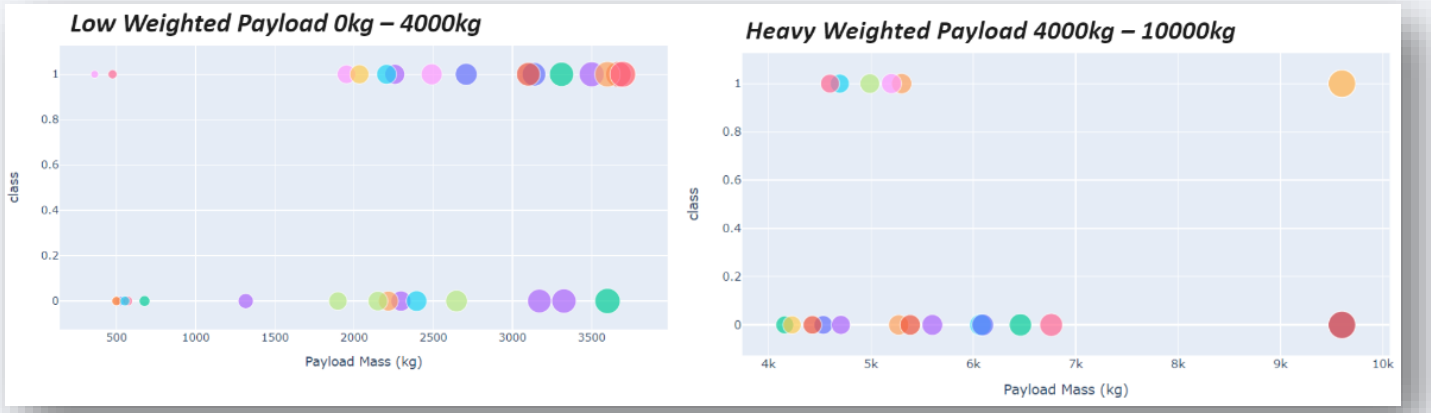


Dashboard (Plotly Dash)

- Outputs from interactive Dashboard

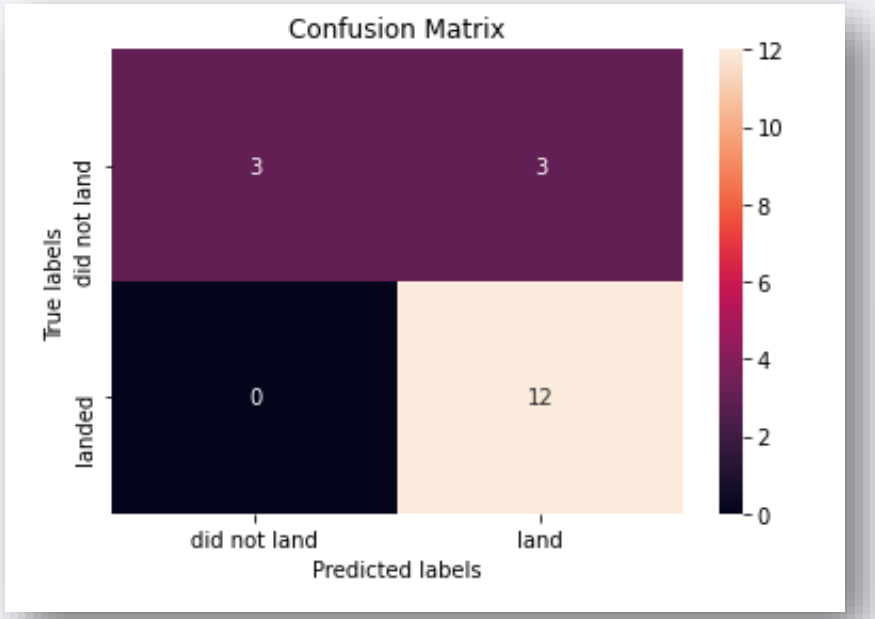


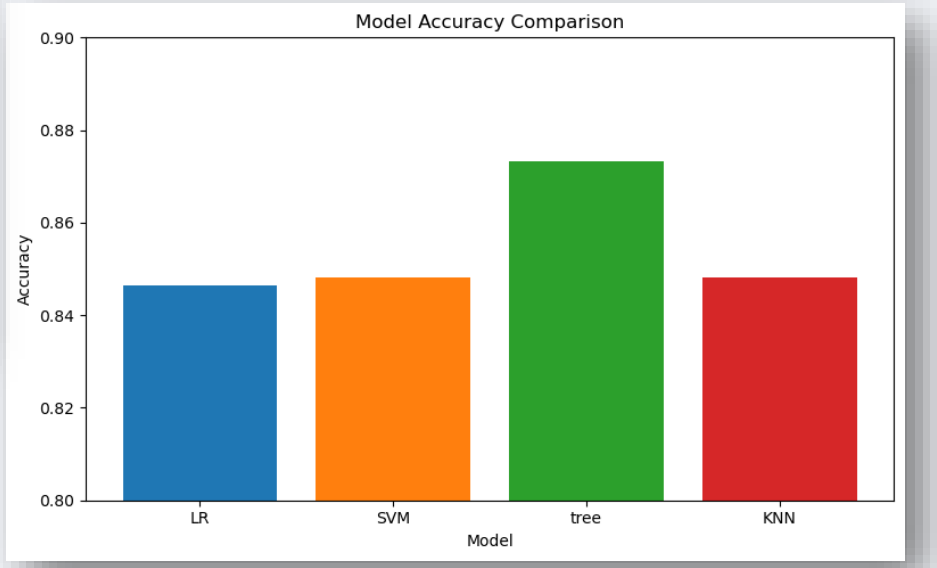




Predictive Analysis (Classification)

- GridSearch using different models resulting different levels of accuracy

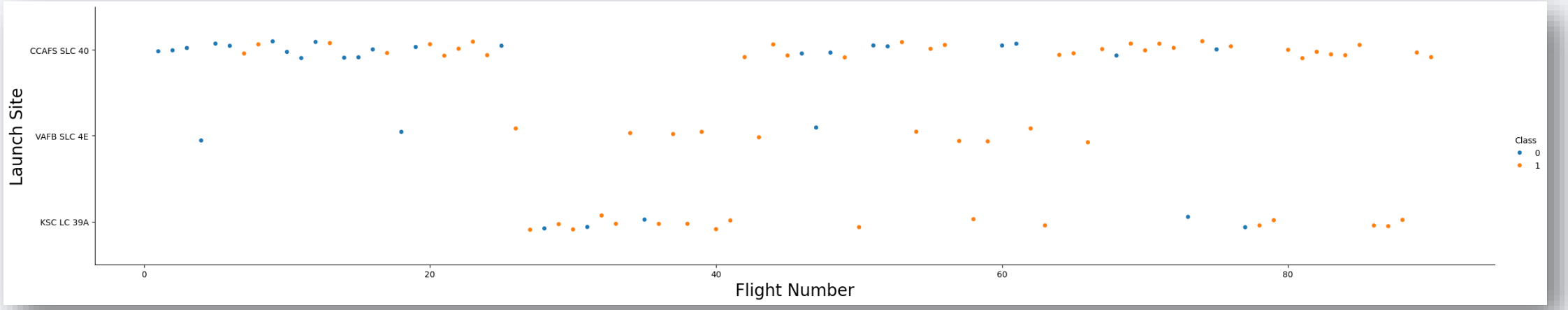




Results

Flight Number vs. Launch Site

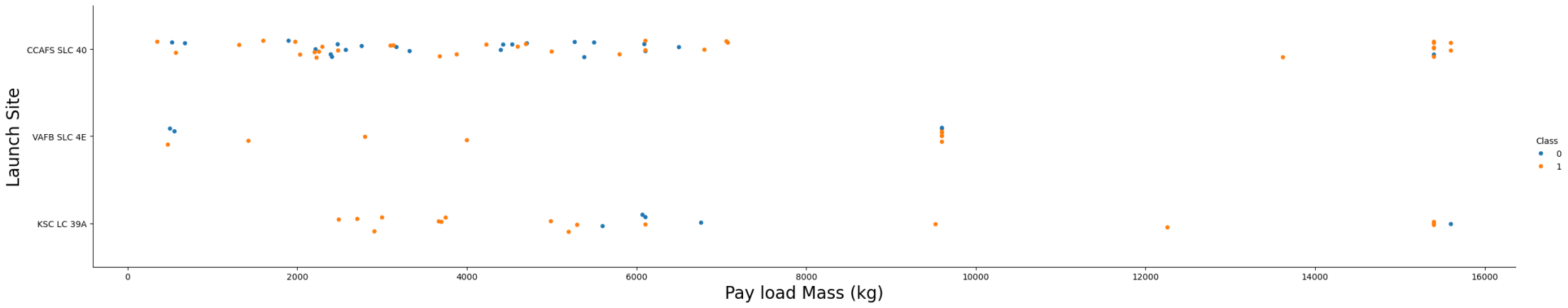
- Flight Number vs. Launch Site



It gives an idea of the landing success rate per site, as well as the evolution of sites' use in time (denoted by the ordered flight number). It reveals that site CCAPS is the most used and that all of them have successful and unsuccessful landing records.

Payload vs. Launch Site

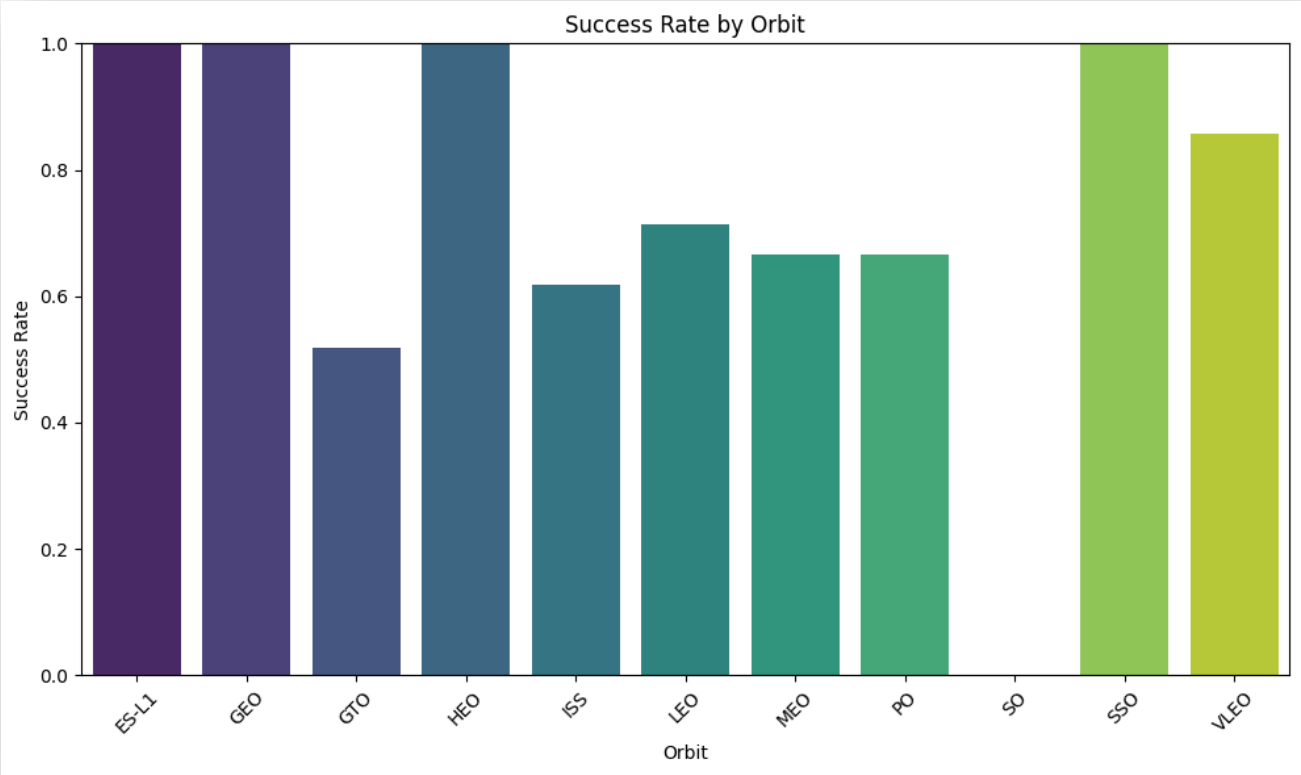
- Payload vs. Launch Site



- It shows that payload mass does not seem limited by launching site. It also shows that launching with payload under 7000Kg are more common. It also shows that no relation between payload and success of landing is visible.

Success Rate vs. Orbit Type

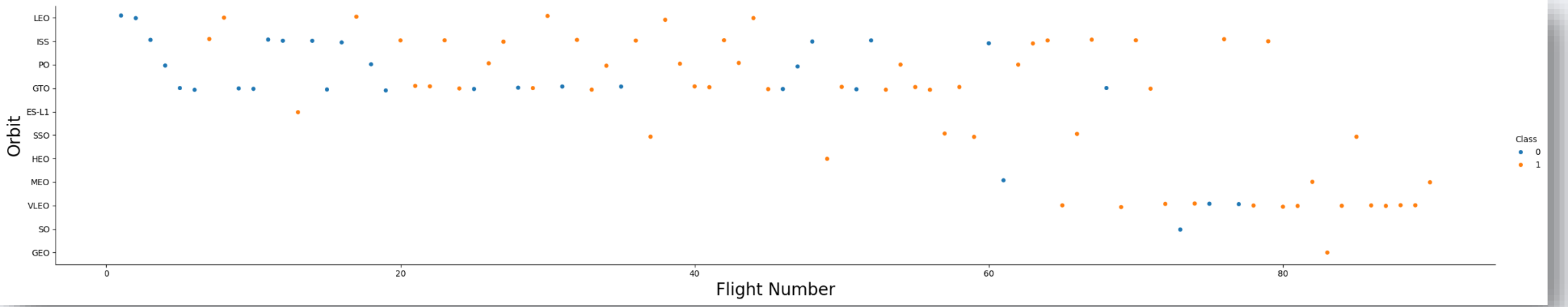
- Success rate of each orbit type



- It shows 3 orbits with 100% success rate and an average value near 60% for the others

Flight Number vs. Orbit Type

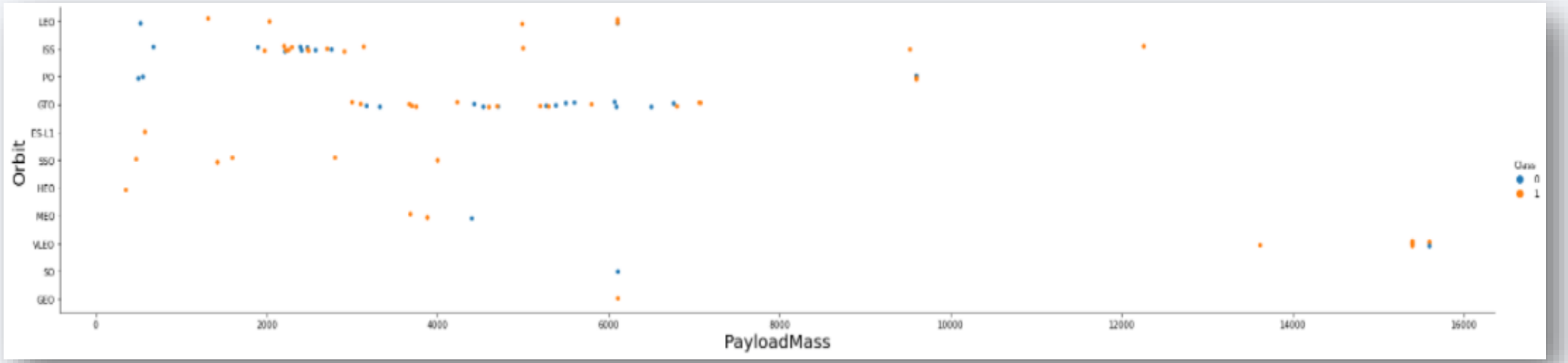
- Flight number vs. Orbit type



- It shows that the preferred orbit is the GTO, however last flights have chosen VLEO predominantly.

Payload vs. Orbit Type

- Payload vs. orbit type

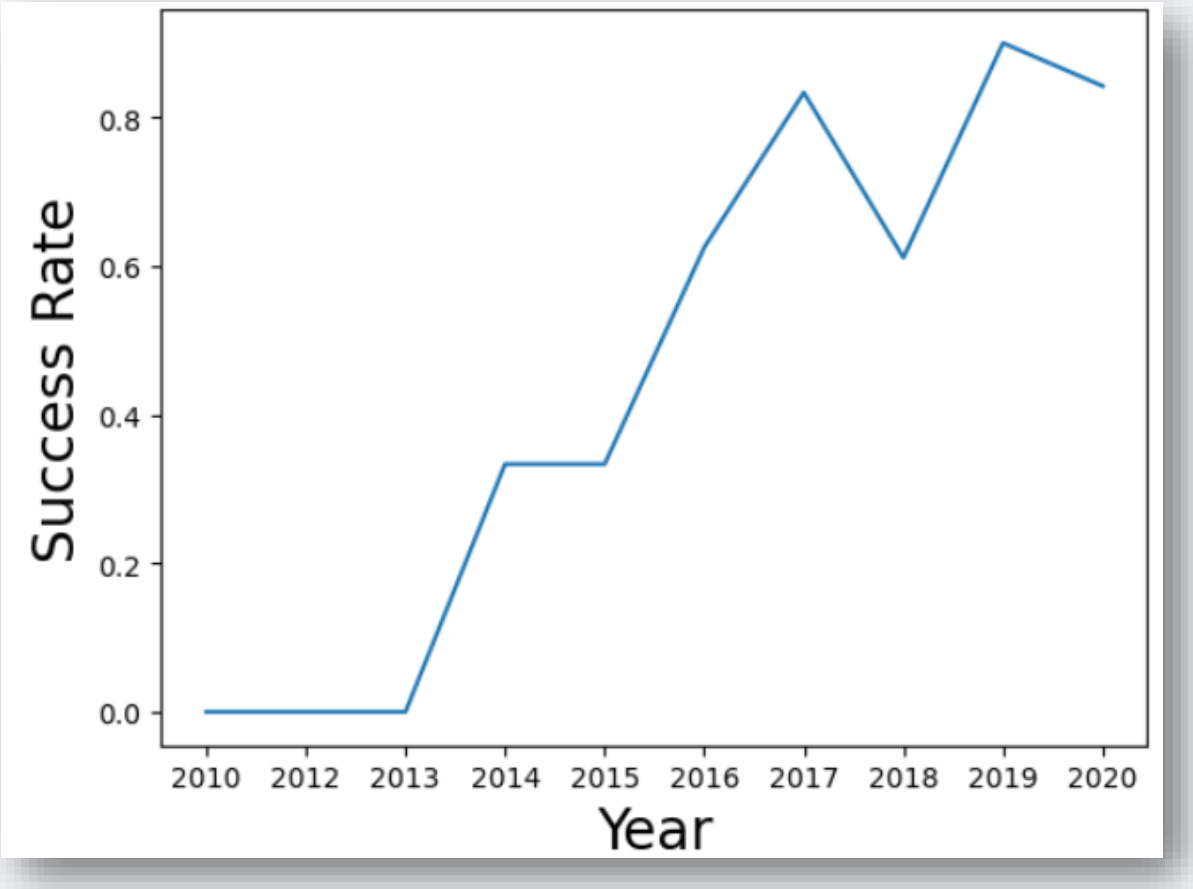


- It shows a wide range of payloads assigned to each orbit and a concentration of high payload launchings using VLEO

Launch Success Yearly Trend

- Yearly average success rate

It shows that success rate has climbed during the last decade and remains stable around 70% nowadays



All Launch Site Names

- Distinct selection shows the different launch sites in the database.

```
%sql SELECT DISTINCT Launch_Site FROM SPACEXTBL;
```

```
 * sqlite:///my_data1.db
Done.
```

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

Launch Site Names Begin with 'CCA'

```
%sql SELECT * FROM SPACEXTBL WHERE Launch_Site LIKE 'CCA%';
```

```
 * sqlite:///my_data1.db
Done.
```

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	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	0:35:00	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

- These are the first 5 records with launch sites beginning with ‘CCA’

Total Payload Mass

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE Customer LIKE 'NASA (CRS)';
```

```
 * sqlite:///my_data1.db
Done.
```

SUM(PAYLOAD_MASS__KG_)
45596

- Total payload carried by all boosters from NASA reaches 45,596 kg

Average Payload Mass by F9 v1.1

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE Booster_Version LIKE 'F9 v1.1';
```

```
 * sqlite:///my_data1.db
Done.
```

AVG(PAYLOAD_MASS__KG_)
2928.4

- payload carried by all boosters from NASA averages 2,928.4 Kg.

First Successful Ground Landing Date

```
%sql SELECT MIN(Date) FROM SPACEXTBL WHERE Landing_Outcome LIKE 'Success (ground pad)';
```

```
 * sqlite:///my_data1.db
Done.
```

MIN(Date)
2015-12-22

- Fisrt successful landing happened in December 22, 2015.

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%sql SELECT Booster_Version  FROM SPACEXTBL WHERE (Landing_Outcome LIKE 'Success (drone ship)') AND (PAYLOAD_MASS__KG_ > 4000)AND (PAYLOAD_MASS__KG_ < 6000);
```

```
 * sqlite:///my_data1.db
Done.
```

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

- These 4 boosters have successfully landed on drone ship and had payload between 4000 and 6000.

Total Number of Successful and Failure Mission Outcomes

```
%sql SELECT COUNT(*) FROM SPACEXTBL WHERE (Mission_Outcome LIKE 'Success');
```

```
 * sqlite:///my_data1.db
Done.
```

COUNT(*)
98

```
%sql SELECT COUNT(*) FROM SPACEXTBL WHERE (Mission_Outcome NOT LIKE 'Success');
```

```
 * sqlite:///my_data1.db
Done.
```

COUNT(*)
3

- Dataset shows 98 successful missions and 3 failures.

Boosters Carried Maximum Payload

```
%sql SELECT Booster_Version, PAYLOAD_MASS__KG_ FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL);
```

```
 * sqlite:///my_data1.db
Done.
```

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

- These are the 12 boosters which have carried the maximum payload mass of 15,600 Kg

2015 Launch Records

```
%%sql
SELECT
  CASE SUBSTR(Date, 6, 2)
    WHEN '01' THEN 'January'
    WHEN '02' THEN 'February'
    WHEN '03' THEN 'March'
    WHEN '04' THEN 'April'
    WHEN '05' THEN 'May'
    WHEN '06' THEN 'June'
    WHEN '07' THEN 'July'
    WHEN '08' THEN 'August'
    WHEN '09' THEN 'September'
    WHEN '10' THEN 'October'
    WHEN '11' THEN 'November'
    WHEN '12' THEN 'December'
  END AS Month_Name,
  Landing_Outcome,
  Booster_Version,
  Launch_Site
FROM SPACEXTBL
WHERE
  SUBSTR(Date, 1, 4) = '2015'
  AND LOWER(Landing_Outcome) LIKE '%failure%'
  AND LOWER(Landing_Outcome) LIKE '%drone ship%';
```

- These are the two 2015 failed landings in drone ship, with their respective booster versions, and launch site names.

```
 * sqlite:///my_data1.db
Done.
```

Month_Name	Landing_Outcome	Booster_Version	Launch_Site
January	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
April	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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

```
%%sql
SELECT
  Landing_Outcome,
  COUNT(*) AS Outcome_Count
FROM SPACEXTBL
WHERE Date >= '2010-06-04' AND Date <= '2017-03-20'
GROUP BY Landing_Outcome
ORDER BY Outcome_Count DESC;
```

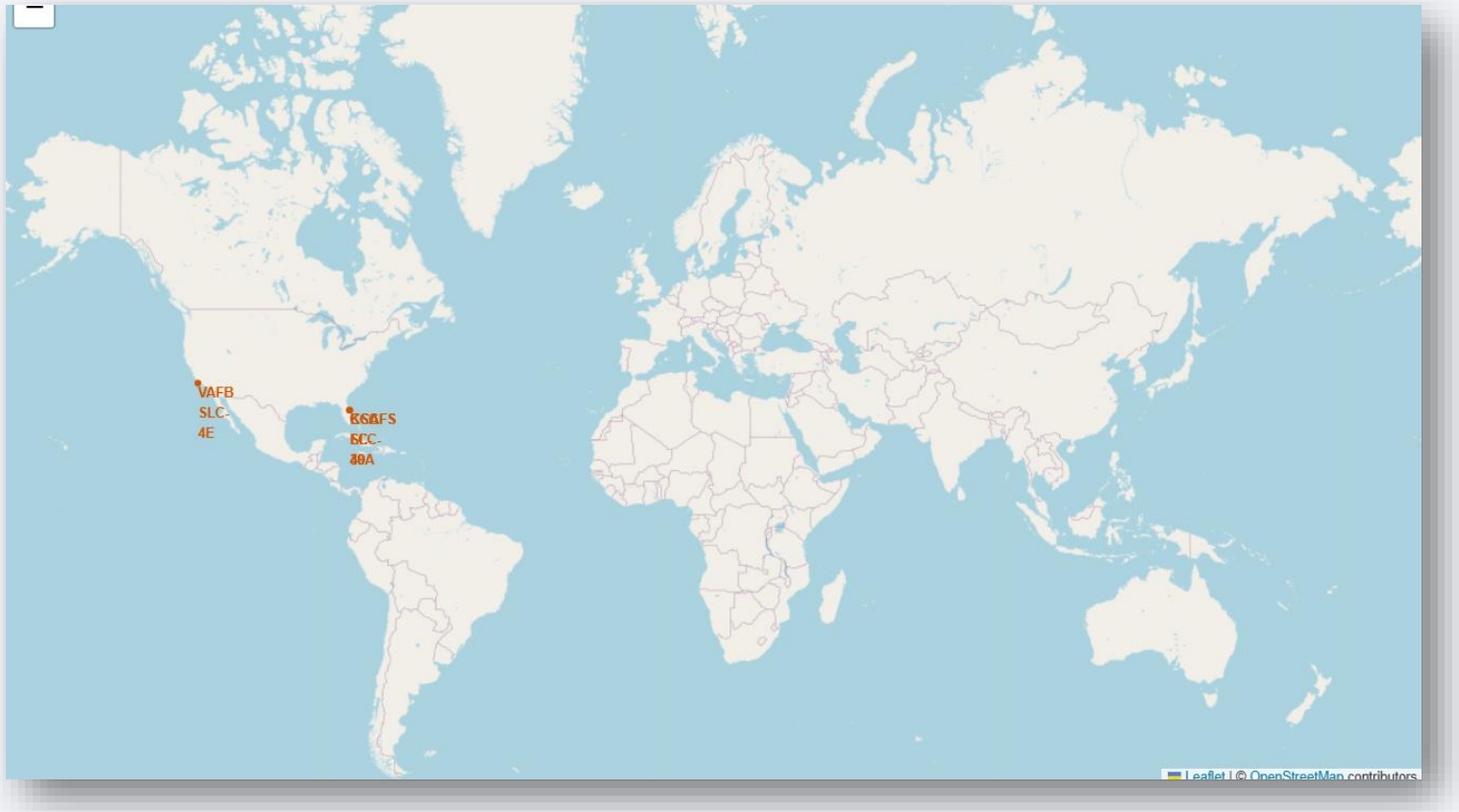
- This is the ranking of landing outcomes between 2010-06-04 and 2017-03-20, in descending order.
- It shows a predominance of drone ship landings and it shows that the rate of success per type of landing is constant

```
 * sqlite:///my_data1.db
Done.
```

Landing_Outcome	Outcome_Count
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

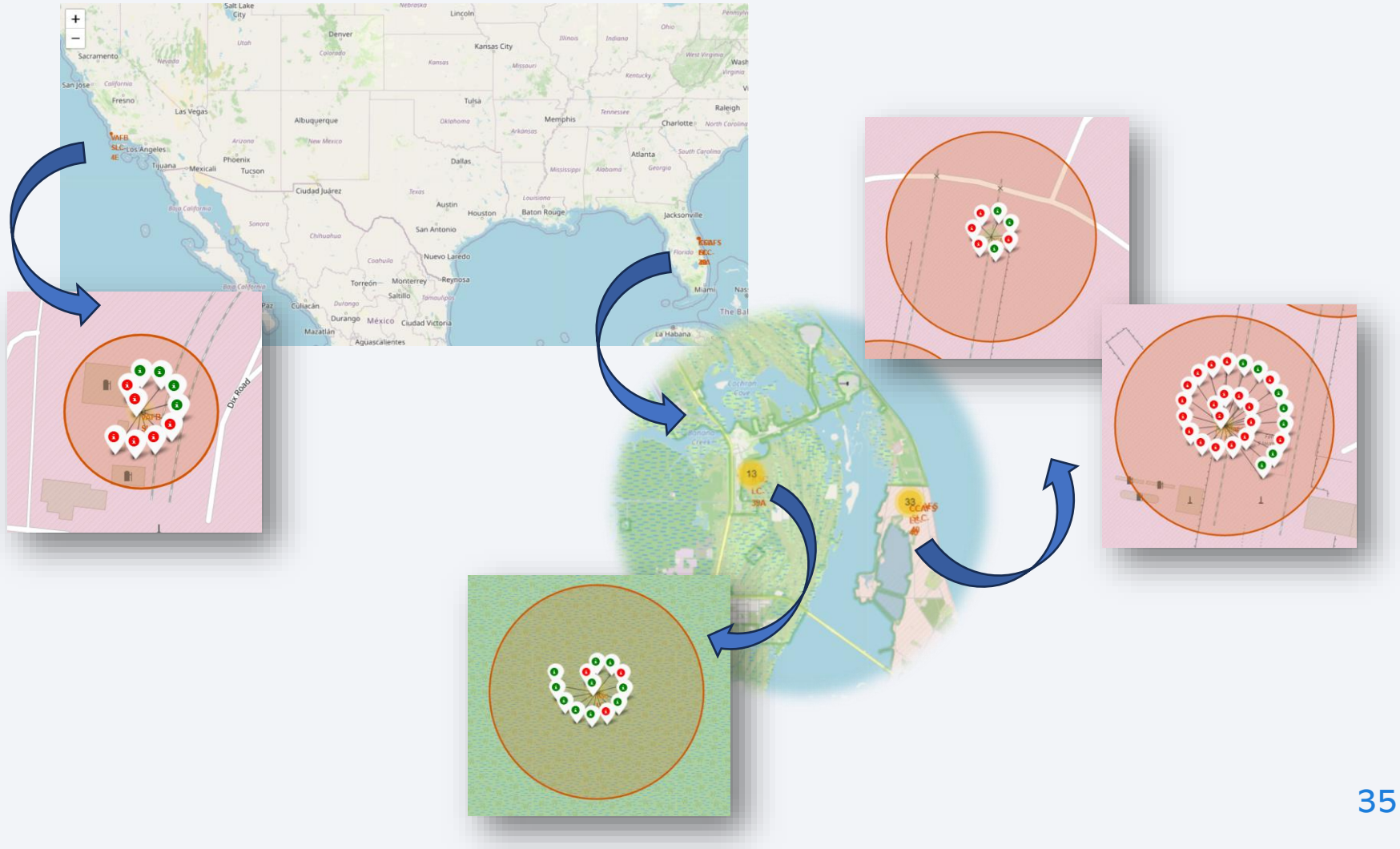
Space X launchings global positioning

- The sites are concentrated in two locations one in each coast of the USA.



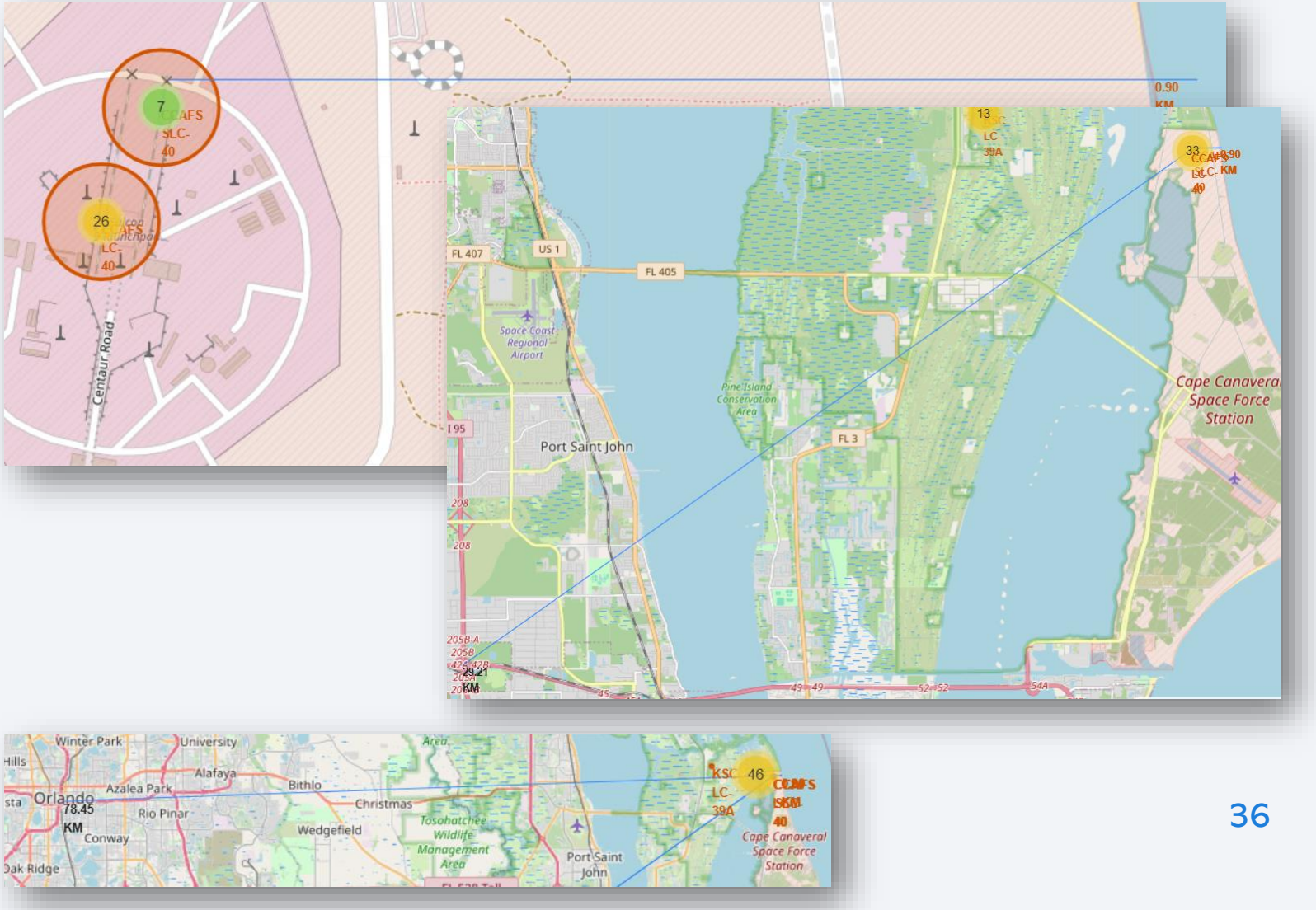
Exploring details of un/successful landings per site

- 4 main locations presenting successful and unsuccessful landings.
- The interactive map allows drilling down to the details of each location.



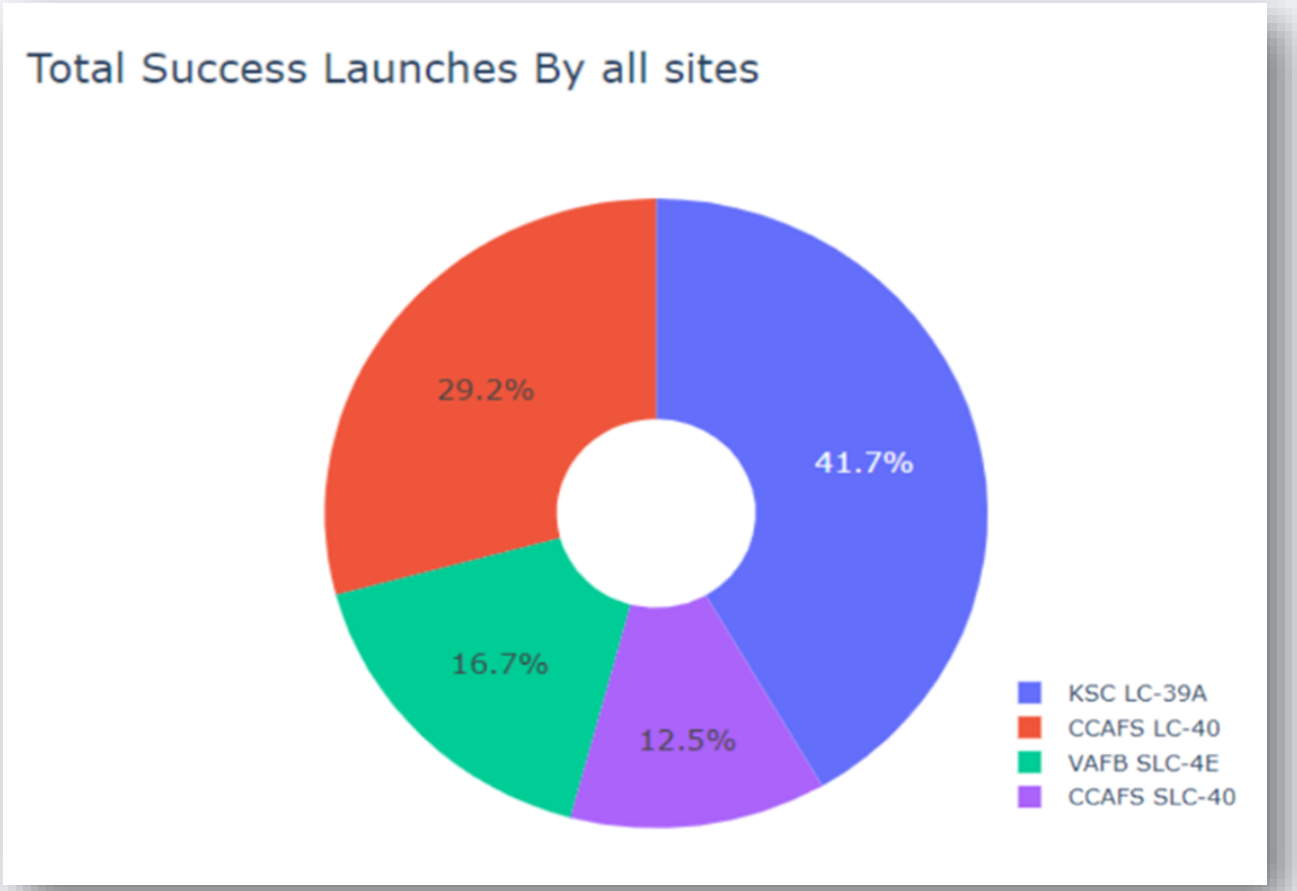
Launch sites distances to POI

- Distances from Florida's sites to coastline (0.9km), highway (29.21km), and cities (78.45km) shows very different levels of proximity.



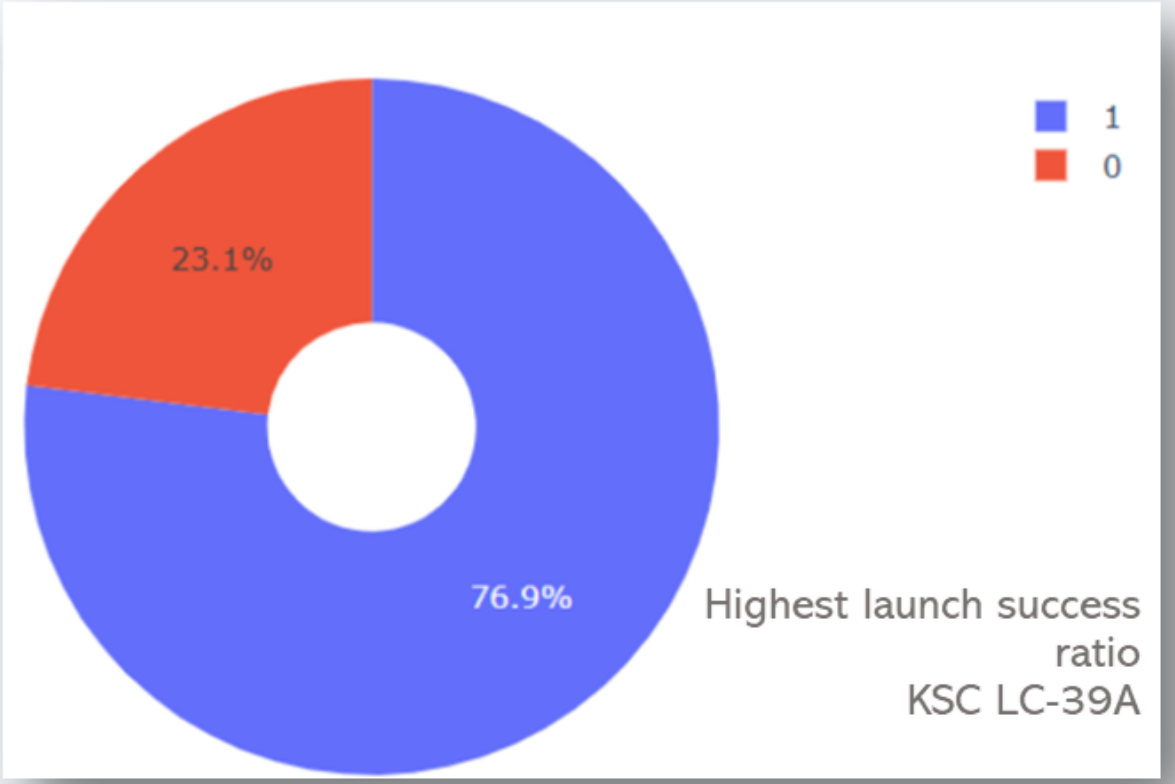
Success percentage achieved by each launch site

- KSC LC leads the ranking of successful launches followed by CCAFS.

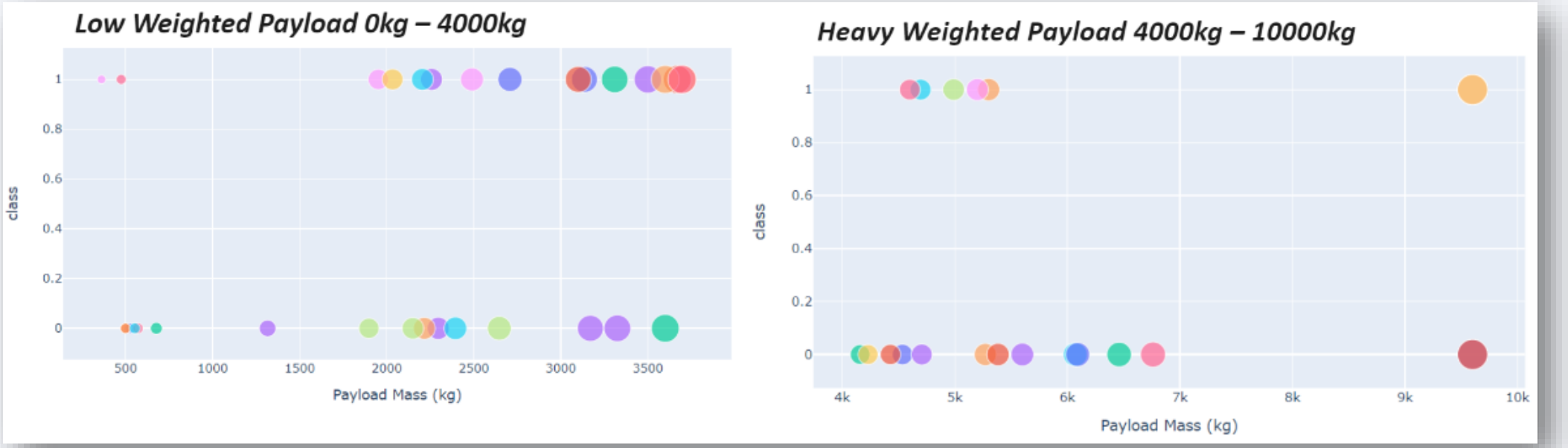


Highest launch success ratio site

- The KSC LC-39A has the highest launch success ratio (76.9%)



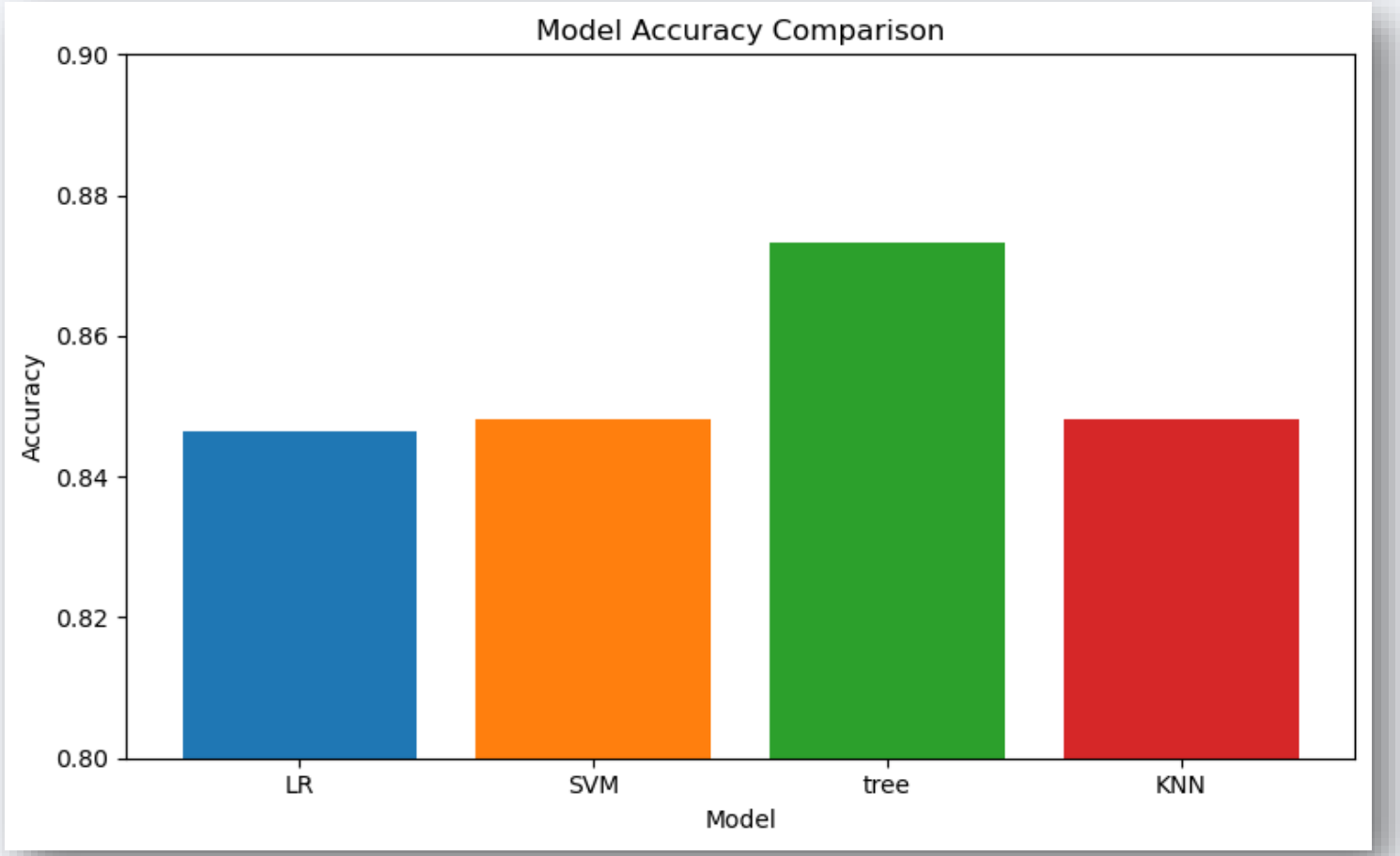
Payload vs Launch Outcome



- Payload range under 4000kg have the largest success rate.

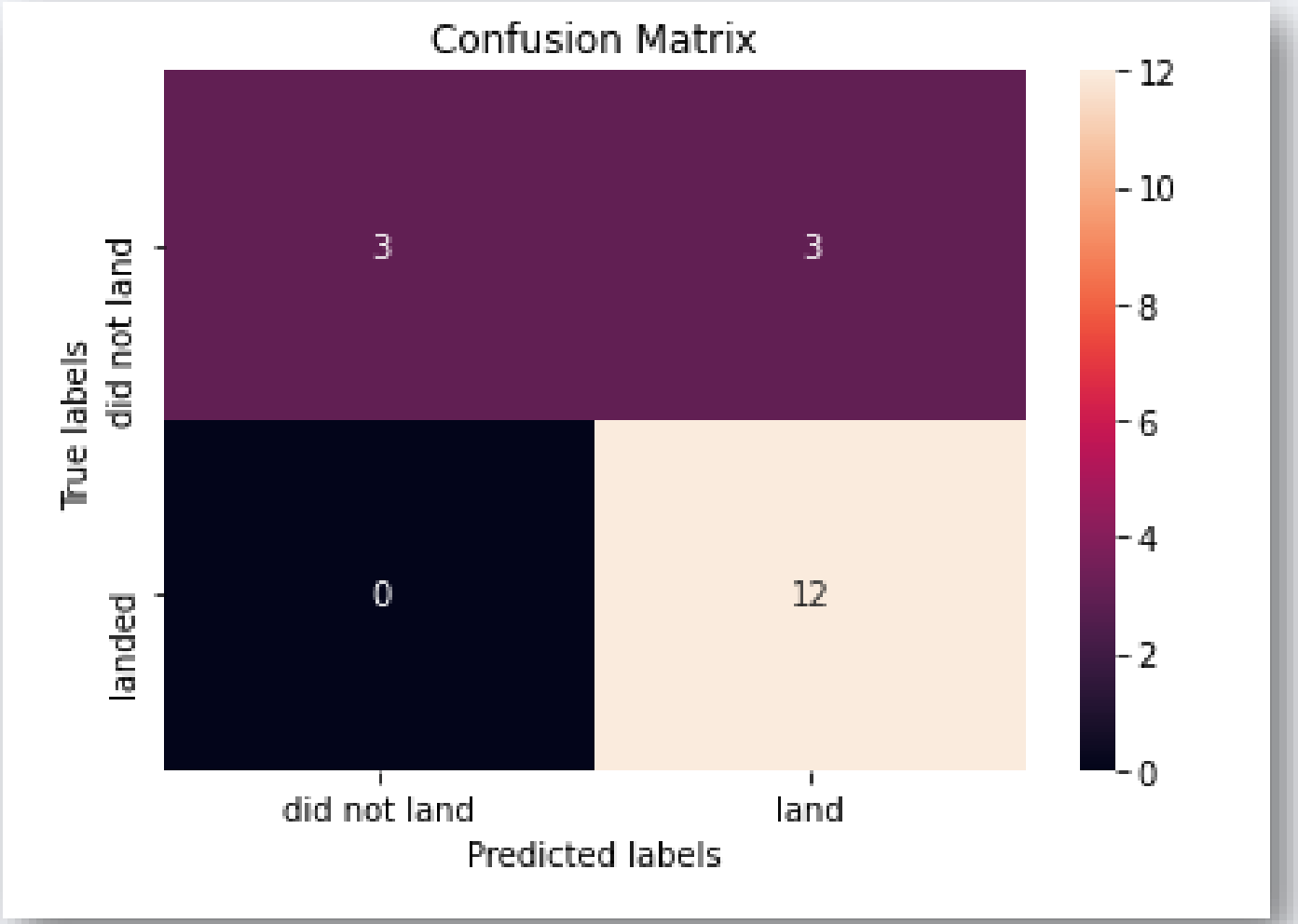
Classification Accuracy

- The decision tree model shows a better accuracy (around 87%) while the average accuracy is around 85%.



Confusion Matrix

- Confusion matrix for the decision tree model which is the best performing model, demonstrated by maximum accuracy



Conclusions

Conclusions

The data about SpaceX launchings allow us to conclude that:

- After 2013 the rate of successful launchings rose dramatically.
- Orbits GEO and VLEO are the most successful.
- KSC LC-39A is the most successful site.
- The Decision tree classifier presented the highest accuracy to predict success of launchings

Appendix

- Codes for reproducing the results shown in this presentation are available at:

https://github.com/Diegobenedetti/IBM_DS_course_capstone_project


Thank you