IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Mahmoud ALI
20-04-2022

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection using API and Web Scraping
  - Exploratory Data Analysis using SQL
  - Exploratory Data Analysis using Data Visualization
  - Interactive Visual Analytics with Folium
  - Interactive Visual Analytics with Dashboard
  - Machine Learning Prediction

- Summary of all results
  - Exploratory Data Analysis
  - Interactive analytics in Dashboard
  - Predictive Analytics using different algorithms

Introduction

- Project background and context

  Space X 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 Space X can reuse the first stage. Therefore, if we can determine if the first stage will land, we can determine the cost of a launch. The goal of the project is to create a machine learning pipeline to predict if the first stage will land successfully.

- Problems you want to find answers
  - The factors affect the successful landing of the rocket
  - The interaction of various features that determine the success rate of a successful landing.
  - The operating conditions assist the successful landing

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was collected using SpaceX API and web scraping from Wikipedia
- Perform data wrangling
  - Clean data to be suitable for exploratory data analysis and machine learning.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection

- The data was collected using various methods
  - Data was collected using get request to the SpaceX API.
  - using .json() and .json_normalize(), the response content is turned to a pandas dataframe.
  - The missing values are examined and replace with the appropriate values
  - A web scraping from Wikipedia for Falcon 9 launch records is performed with BeautifulSoup.
  - Using the BeautifulSoup, the HTML tables are parsed and converted to pandas dataframe.

Data Collection – SpaceX API

- get request to the SpaceX API is used to collect data, clean the requested data and did some data wrangling and formatting

https://github.com/mahmoud-E-ALI/Capstone-IBM-Coursera/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

In [11]:
```
static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_call_
```

We should see that the request was successfull with the 200 status response code

In [12]:
```
response.status_code
```

Out[12]: 200

Now we decode the response content as a Json using `.json()` and turn it into a Pandas dataframe using `.json_normalize()`

In [13]:
```
# Use json_normalize meethod to convert the json result into a dataframe
response.json()
data = pd.json_normalize(response.json())
```

Using the dataframe `data` print the first 5 rows

In [24]:
```
# Get the head of the dataframe
data.head();
```

Data Collection - Scraping

- A web scraping from Wikipedia for Falcon 9 launch records is performed with BeautifulSoup.
- Using the BeautifulSoup, the HTML tables are parsed and converted to pandas dataframe.

https://github.com/mahmoud-E-ALI/Capstone-IBM-Coursera/blob/main/jupyter-labs-webscraping.ipynb

TASK 1: Request the Falcon9 Launch Wiki page from its URL

First, let's perform an HTTP GET method to request the Falcon9 Launch HTML page, as an HTTP response.

In [8]:
```
# use requests.get() method with the provided static_url
req = requests.get(static_url)
# assign the response to a object
html=req.text
```

Create a `BeautifulSoup` object from the HTML `response`

In [12]:
```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(html, 'html5lib')
```

Print the page title to verify if the `BeautifulSoup` object was created properly

In [15]:
```
# Use soup.title attribute
soup.title
soup.title.text
```

Out[15]:
```
'List of Falcon 9 and Falcon Heavy launches - Wikipedia'
```

TASK 2: Extract all column/variable names from the HTML table header

Data Wrangling

- The data is cleaned and duplication and missing values are examined
- Missing values are replaced with the appropriate values

https://github.com/mahmoud-E-ALI/Capstone-IBM-Coursera/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

TASK 2: Calculate the number and occurrence of each orbit

Use the method `.value_counts()` to determine the number and occurrence of each orbit in the column `Orbit`

In [6]:
```
# Apply value_counts on Orbit column
df.Orbit.value_counts()
```

Out[6]:
```
GTO      27
ISS      21
VLEO     14
PO        9
LEO       7
SSO       5
MEO       3
ES-L1     1
HEO       1
SO        1
GEO       1
Name: Orbit, dtype: int64
```

TASK 3: Calculate the number and occurence of mission outcome per orbit type

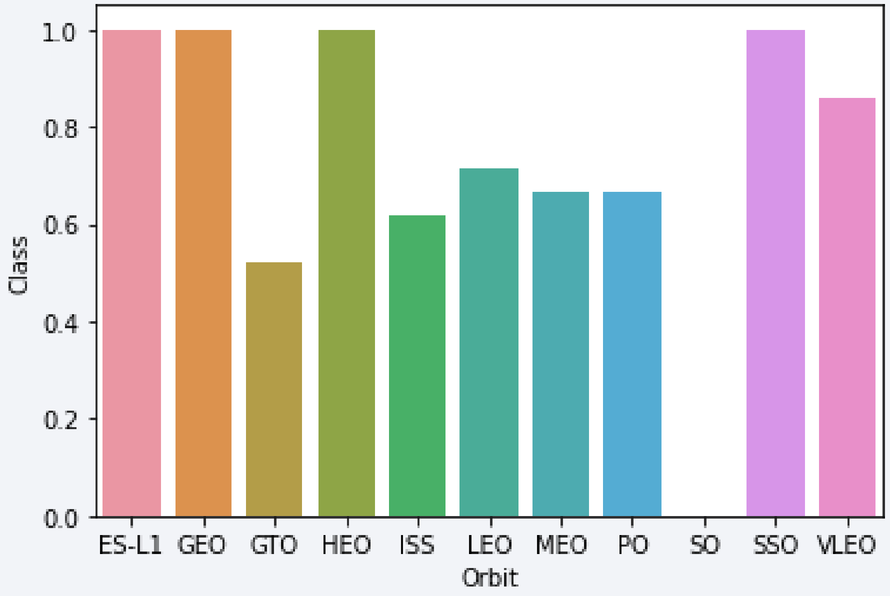
Use the method `.value_counts()` on the column `Outcome` to determine the number of `landing_outcomes` .Then assign it to a variable landing_outcomes.

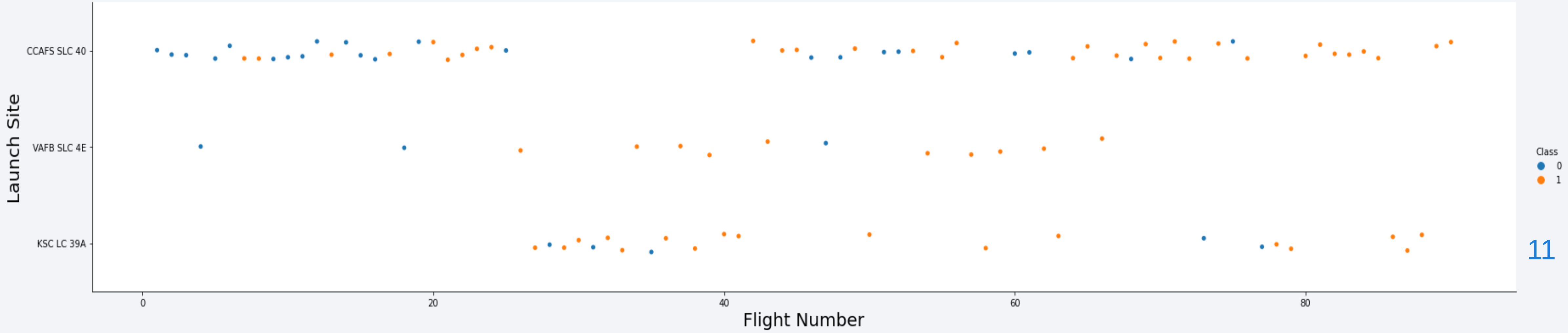
In [13]:
```
# landing_outcomes = values on Outcome column
landing_outcomes = df.Outcome.value_counts()
landing_outcomes
```

EDA with Data Visualization

- The data is explored by visualizing the relationship between flight number and launch Site, payload and launch site, success rate of each orbit type, flight number and orbit type, the launch success yearly

https://github.com/mahmoud-E-ALI/Capstone-IBM-Coursera/blob/main/jupyter-labs-eda-dataviz.ipynb





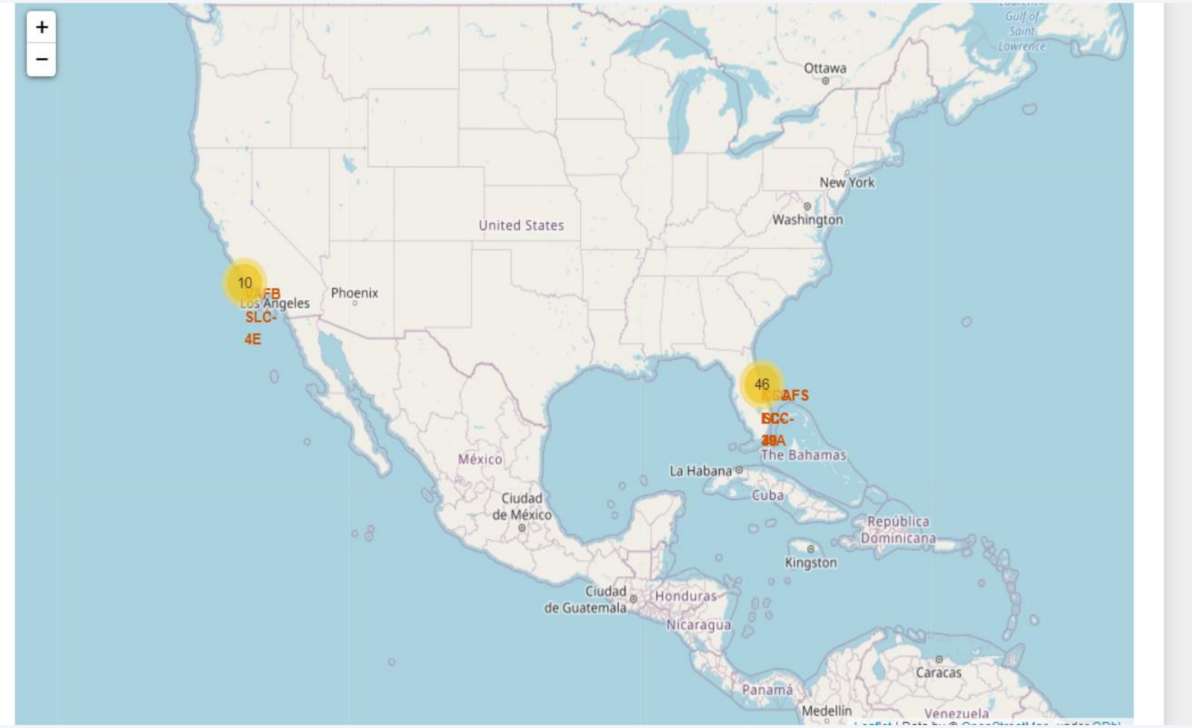
EDA with SQL

- The SpaceX dataset is used with MYSQL database
- Different queries are examined using SQL magic function in Jupyter.
- The queries include:
  - The names of unique launch sites in the space mission.
  - The total payload mass carried by boosters launched by NASA (CRS)
  - The average payload mass carried by booster version F9 v1.1
  - The total number of successful and failure mission outcomes
  - The failed landing outcomes in drone ship, their booster version and launch site names.

https://github.com/mahmoud-E-ALI/Capstone-IBM-Coursera/blob/main/jupyter-labs-eda-sql-coursera.ipynb

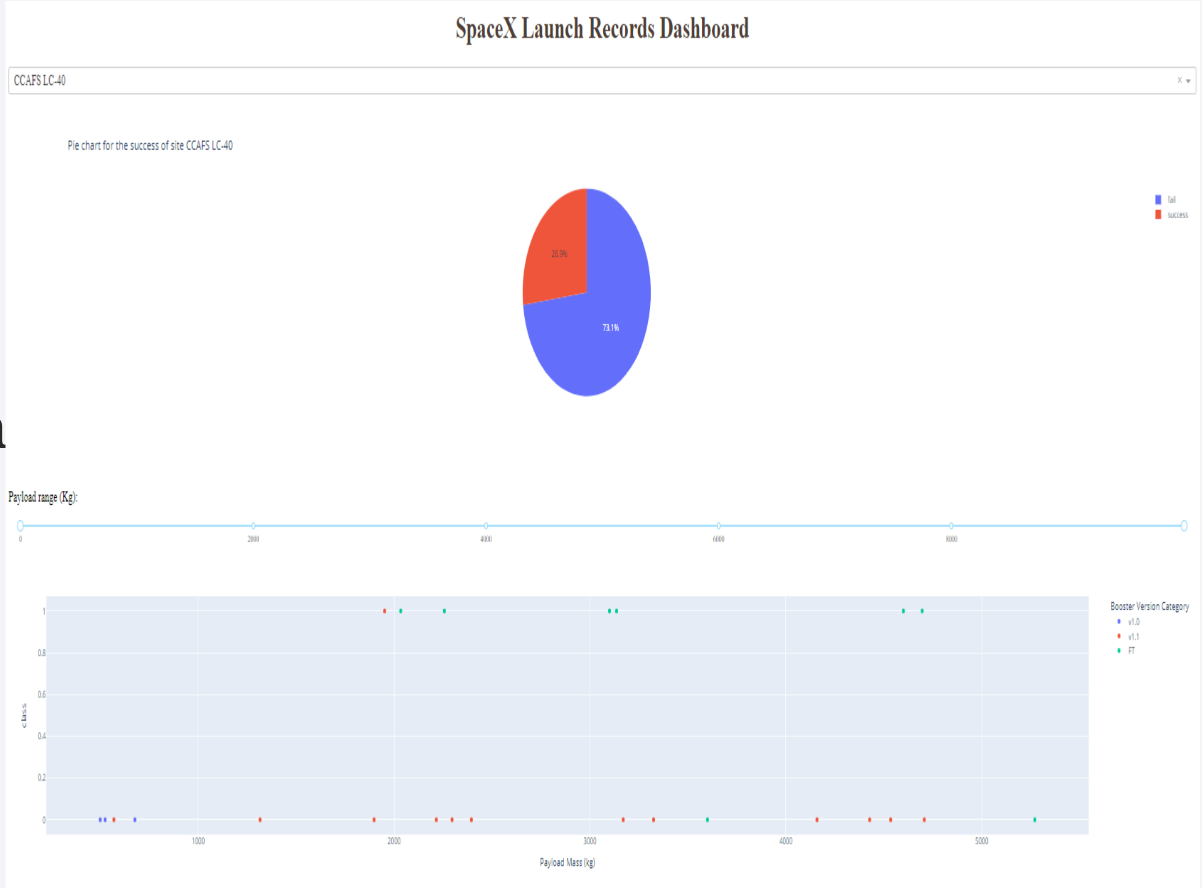
Build an Interactive Map with Folium

- All launch sites are explained on map with adding objects such as markers, circles, lines to mark the success or failure of launches for each site.
- An assignment of failure or success on each site is differentiate by color .
- The distances between a launch site to its proximities to railway, high way and coasts are explained.
- https://github.com/mahmoud-E-ALI/Capstone-IBM-Coursera/blob/main/lab_jupyter_launch_site_location.ipynb



Build a Dashboard with Plotly Dash

- a Plotly Dash application for users to perform interactive visual analytics on SpaceX launch data in real-time.
- This dashboard application contains input components such as a dropdown list and a range slider to interact with a pie chart and a scatter point chart.
- https://github.com/mahmoud-E-ALI/Capstone-IBM-Coursera/blob/main/spacex_dash_app.py



Predictive Analysis (Classification)

- The predictive analysis is started by importing the required packages
- Indicating the input data and the output data
- Splitting the data into train and test data
- Try multiple algorithm to get the best results for the prediction
- These algorithms are :
  - Logistic Regression classification algorithm
  - Support Vector Machine classification algorithm
  - Decision Tree classification algorithm
  - K Nearest Neighbors classification algorithm

https://github.com/mahmoud-E-ALI/Capstone-IBM-Coursera/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results

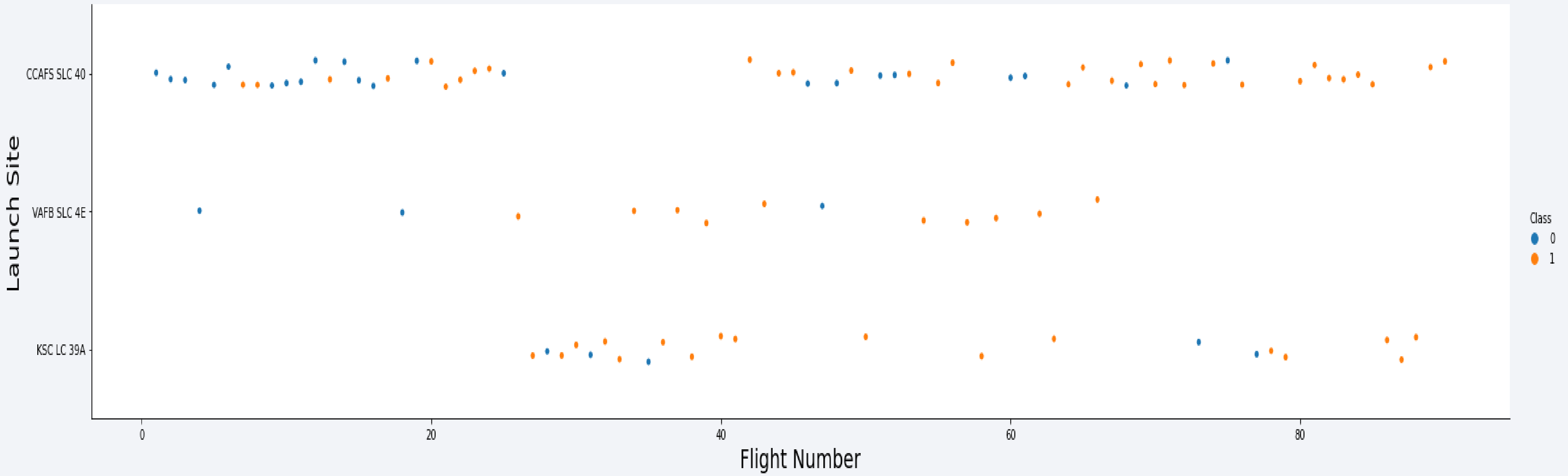
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

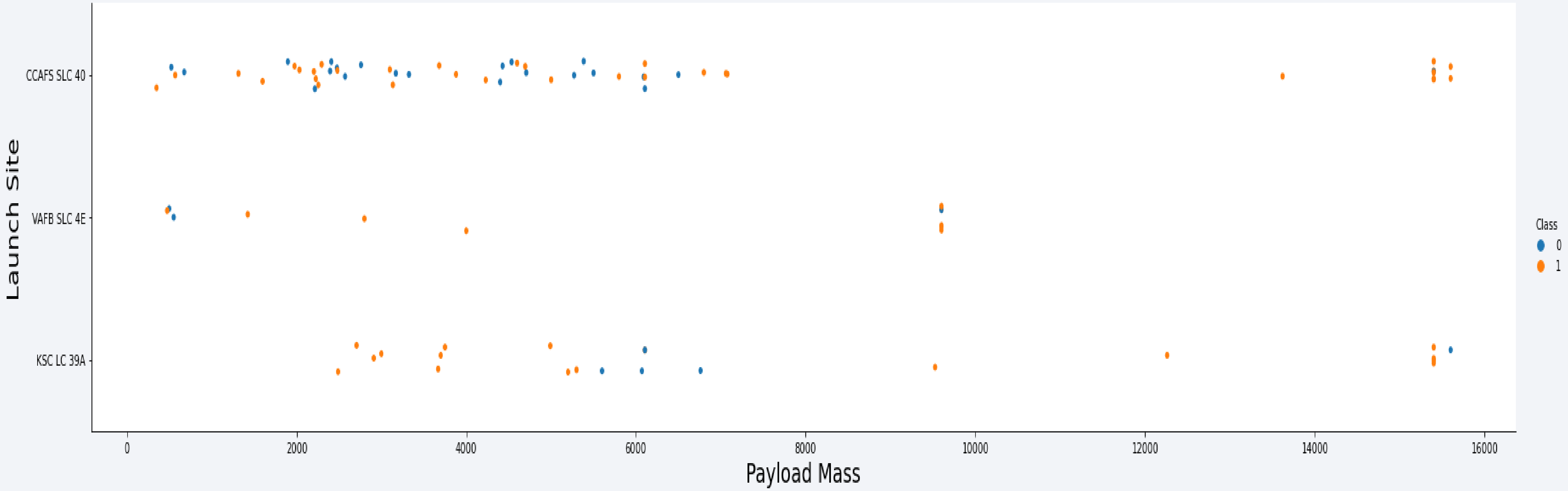
Flight Number vs. Launch Site

- This Figure shows that as the flight number increases, the success rate increases at all launch sites.



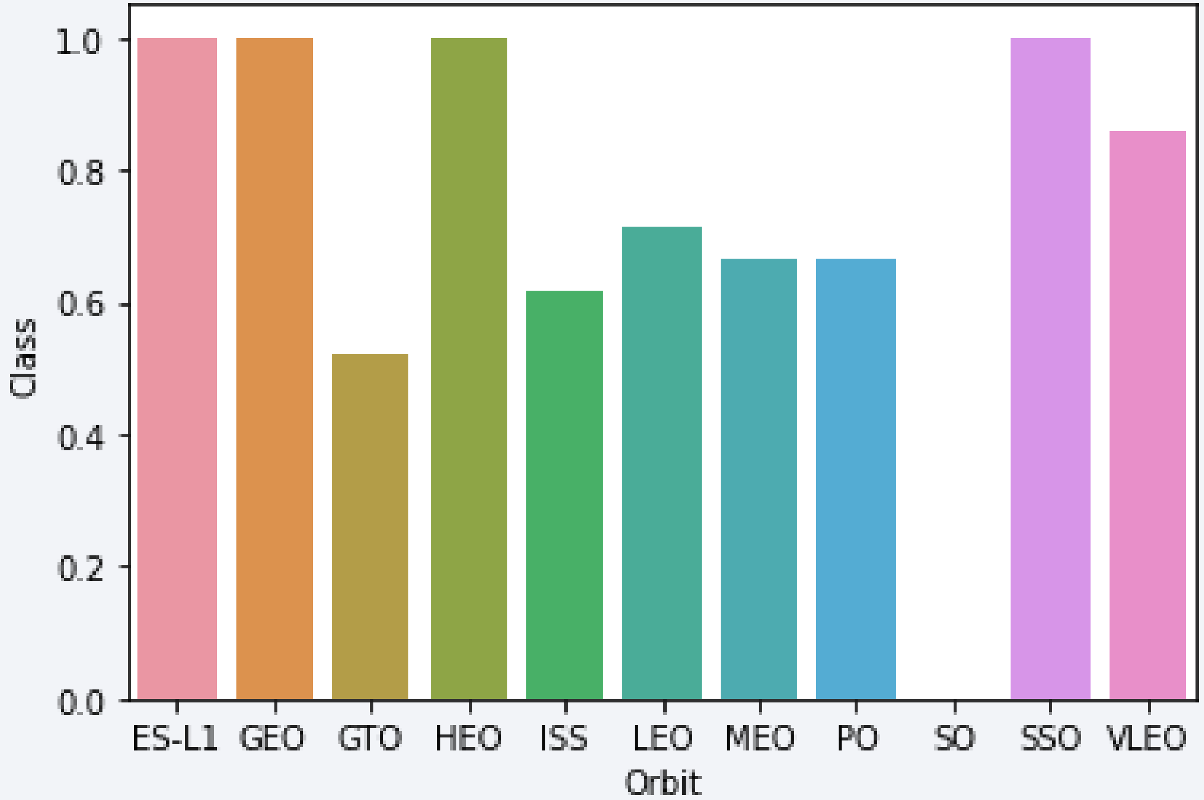
Payload vs. Launch Site

- Now if you observe Payload Vs. Launch Site scatter point chart you will find for the VAFB-SLC launchsite there are no rockets launched for heavypayload mass(greater than 10000)



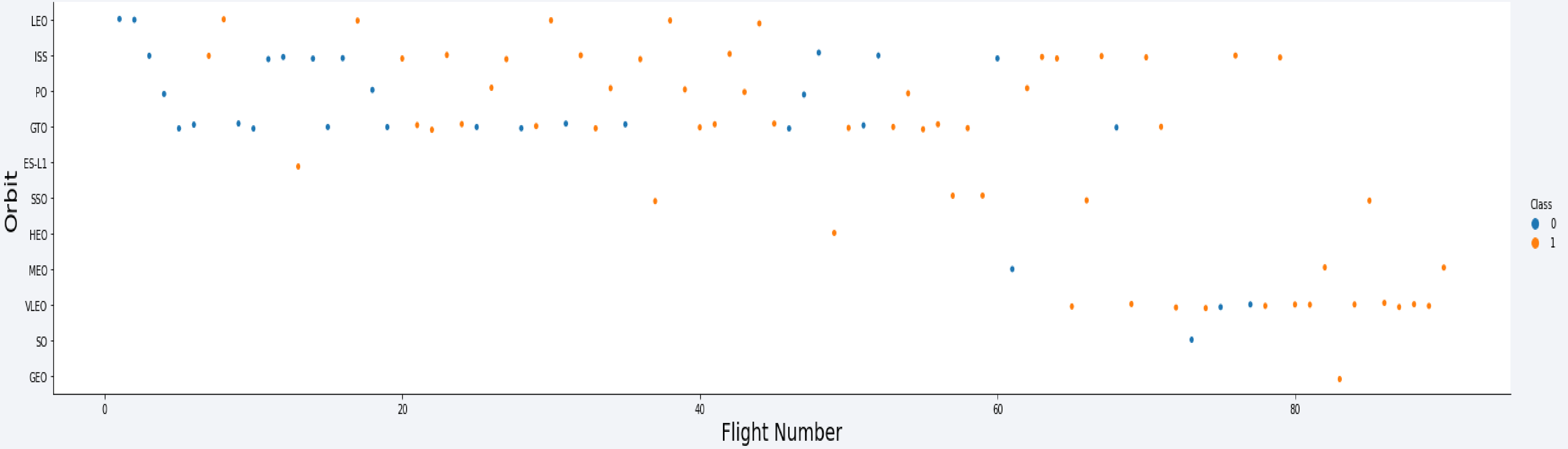
Success Rate vs. Orbit Type

- This Figure shows a bar chart for the success rate of each orbit type, in which the high success rate at ES-L1, GEO, HEO and SSO



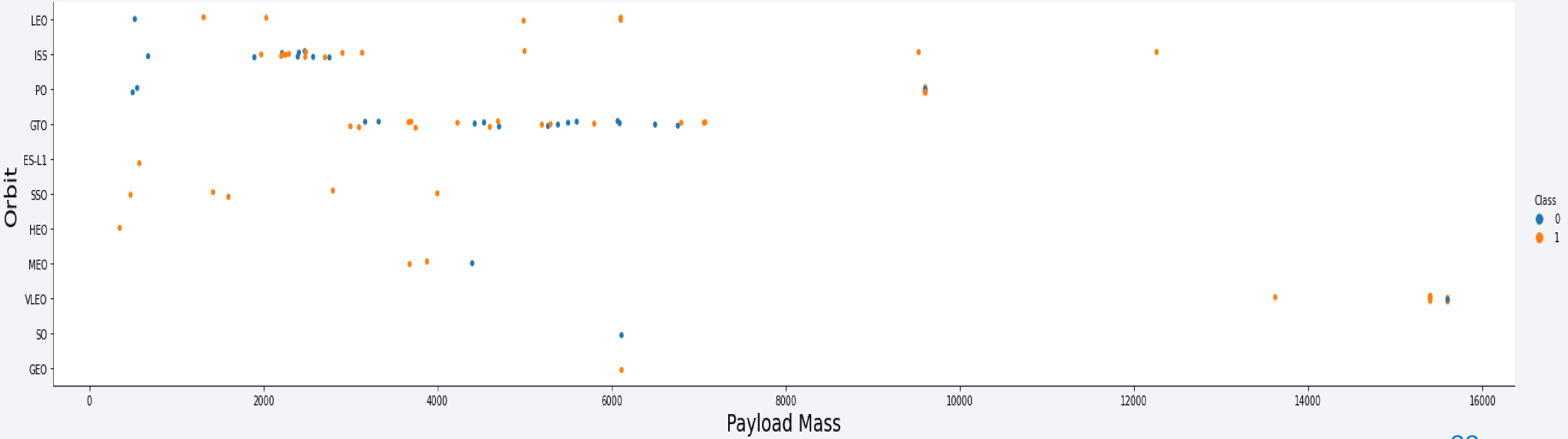
Flight Number vs. Orbit Type

- It is shown that in the LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit



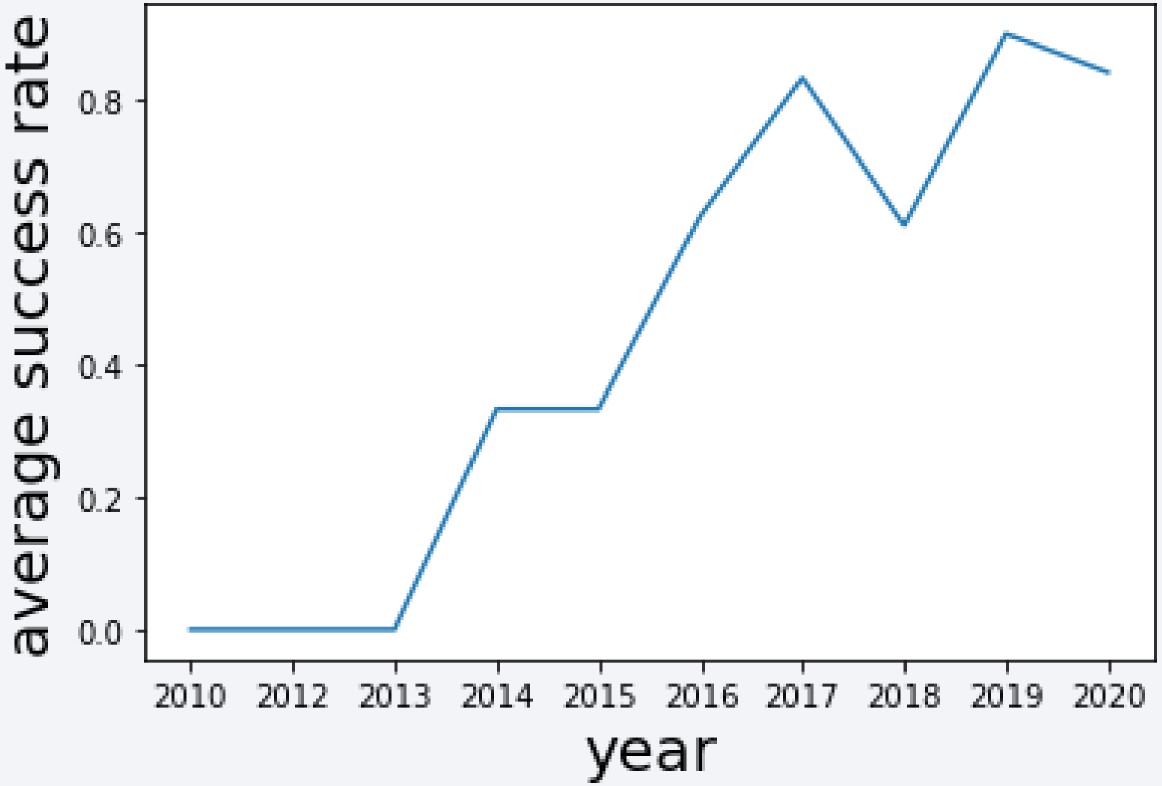
Payload vs. Orbit Type

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



Launch Success Yearly Trend

- you can observe that the sucess rate since 2013 kept increasing till 2020



All Launch Site Names

The querry is used as shown with the DISTINCT keyword

```
In [6]: %%sql
select DISTINCT Launch_Site from spacex

 * mysql+mysqlconnector://root:***@localhost:3306/mydb
4 rows affected.
```

Out[6]:

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

Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with `CCA`

In [25]:
```
%%sql
select *
from spacex
where Launch_Site like "CCA%" limit 5
```

```
 * mysql+mysqlconnector://root:***@localhost:3306/mydb
5 rows affected.
```

Out[25]:

My_date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing _Outcome
04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
08-12-2010	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)
22-05-2012	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
01-03-2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

- Calculate the total payload carried by boosters from NASA

In [26]:
```
%%sql
select Booster_Version, sum(PAYLOAD_MASS__KG_) from spacex where Customer like "NASA%" group by Booster_Version
```

```
 * mysql+mysqlconnector://root:***@localhost:3306/mydb
30 rows affected.
```

Out[26]:

Booster_Version	sum(PAYLOAD_MASS__KG_)
F9 v1.0 B0004	0
F9 v1.0 B0005	525
F9 v1.0 B0006	500
F9 v1.0 B0007	677
F9 v1.1	2296
F9 v1.1 B1010	2216
F9 v1.1 B1012	2395
F9 v1.1 B1015	1898
F9 v1.1 B1018	1952
F9 v1.1 B1017	553
F9 FT B1021.1	3136
F9 FT B1025.1	2257
F9 FT B1031.1	2490
F9 FT B1035.1	2708

In [27]:
```
%%sql
select sum(PAYLOAD_MASS__KG_) from spacex where Customer like "NASA%"
```

```
 * mysql+mysqlconnector://root:***@localhost:3306/mydb
1 rows affected.
```

Out[27]:

sum(PAYLOAD_MASS__KG_)
99980

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1

```
In [28]: %%sql
         select AVG(PAYLOAD_MASS__KG_) from spacex where Booster_Version like  "F9 v1.1%"

          * mysql+mysqlconnector://root:***@localhost:3306/mydb
         1 rows affected.

Out[28]:
```

AVG(PAYLOAD_MASS__KG_)
2534.6667

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad

In [57]:
```
%%sql
SELECT  Landing_Outcome, STR_TO_DATE(My_date, '%d-%m-%Y') as DD
FROM    spacex
where Landing_Outcome like "Success%" limit 10
```

```
 * mysql+mysqlconnector://root:***@localhost:3306/mydb
10 rows affected.
```

Out[57]:

Landing_Outcome	DD
Success (ground pad)	2015-12-22
Success (drone ship)	2016-04-08
Success (drone ship)	2016-05-06
Success (drone ship)	2016-05-27
Success (ground pad)	2016-07-18
Success (drone ship)	2016-08-14
Success (drone ship)	2017-01-14
Success (ground pad)	2017-02-19
Success (drone ship)	2017-03-30
Success (ground pad)	2017-05-01

Successful Drone Ship Landing with Payload between 4000 and 6000

- List the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000

In [63]:
```
%%sql
select Booster_Version
from spacex where Landing_Outcome like 'Success%' and  PAYLOAD_MASS__KG_ > 4000 and PAYLOAD_MASS__KG_ < 6000
```

```
 * mysql+mysqlconnector://root:***@localhost:3306/mydb
14 rows affected.
```

Out[63]:

Booster_Version
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
F9 FT B1032.1
F9 B4 B1040.1
F9 FT B1031.2
F9 B4 B1043.1
F9 B5 B1046.2
F9 B5 B1047.2
F9 B5 B1048.3
F9 B5 B1051.2
F9 B5B1060.1
F9 B5 B1058.2
F9 B5B1062.1

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes

In [65]:
```
%%sql
select Mission_Outcome, count(Mission_Outcome) from spacex group by Mission_Outcome
```

```
 * mysql+mysqlconnector://root:***@localhost:3306/mydb
4 rows affected.
```

Out[65]:

Mission_Outcome	count(Mission_Outcome)
Success	98
Failure (in flight)	1
Success (payload status unclear)	1
Success	1

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass

```
In [74]: %%sql
select Booster_Version, PAYLOAD_MASS__KG_ from spacex where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from spacex)

 * mysql+mysqlconnector://root:***@localhost:3306/mydb
12 rows affected.
```

Out[74]:

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

- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

```
In [81]: %%sql
select Landing_Outcome, Booster_Version, Launch_Site, my_date
from spacex where Landing_Outcome = 'Failure (drone ship)' and my_date like '%2015'

 * mysql+mysqlconnector://root:***@localhost:3306/mydb
2 rows affected.
```

Out[81]:

Landing_Outcome	Booster_Version	Launch_Site	my_date
Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40	10-01-2015
Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40	14-04-2015

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

- 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

```
In [90]: %%sql
select count(*), Landing_Outcome, STR_TO_DATE(My_date, '%d-%m-%Y') as DD from spacex group by Landing_Outcome  '

 * mysql+mysqlconnector://root:***@localhost:3306/mydb
11 rows affected.
```

Out[90]:

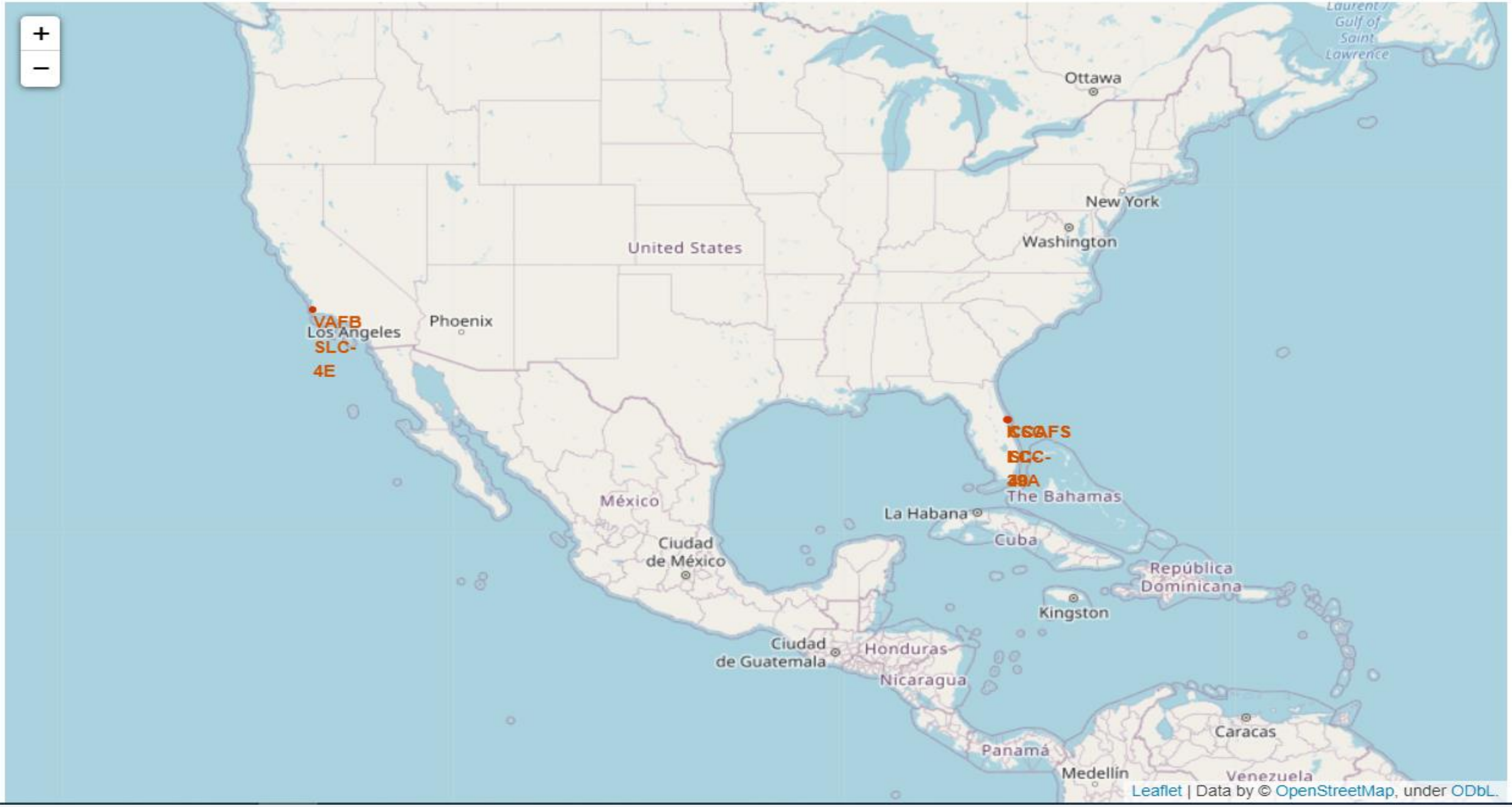
count(*)	Landing_Outcome	DD
2	Failure (parachute)	2010-06-04
21	No attempt	2012-05-22
2	Uncontrolled (ocean)	2013-09-29
5	Controlled (ocean)	2014-04-18
5	Failure (drone ship)	2015-01-10
1	Precluded (drone ship)	2015-06-28
9	Success (ground pad)	2015-12-22
14	Success (drone ship)	2016-04-08
38	Success	2018-07-22
3	Failure	2018-12-05
1	No attempt	2019-08-06

Section 3

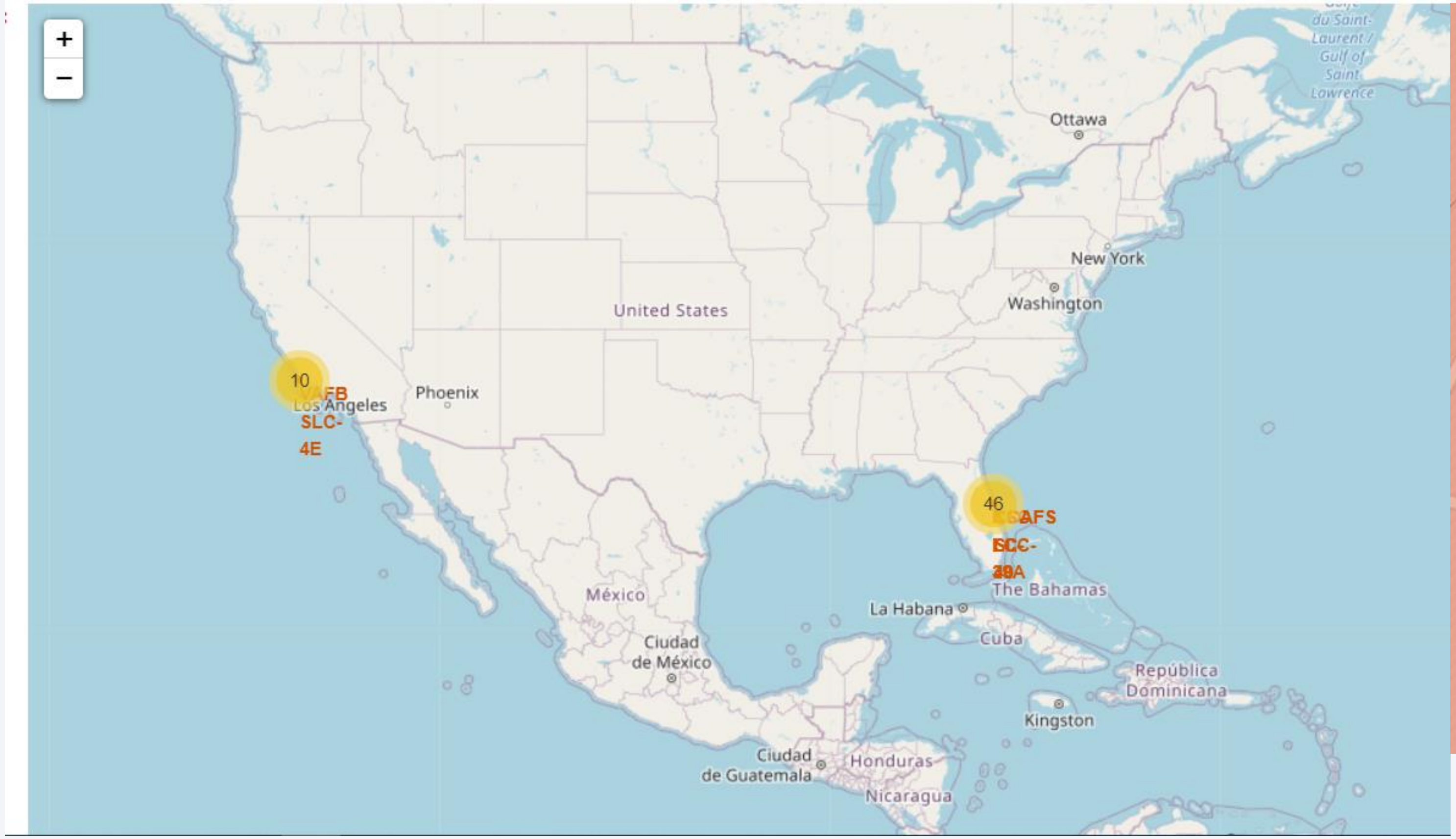
Launch Sites Proximities Analysis

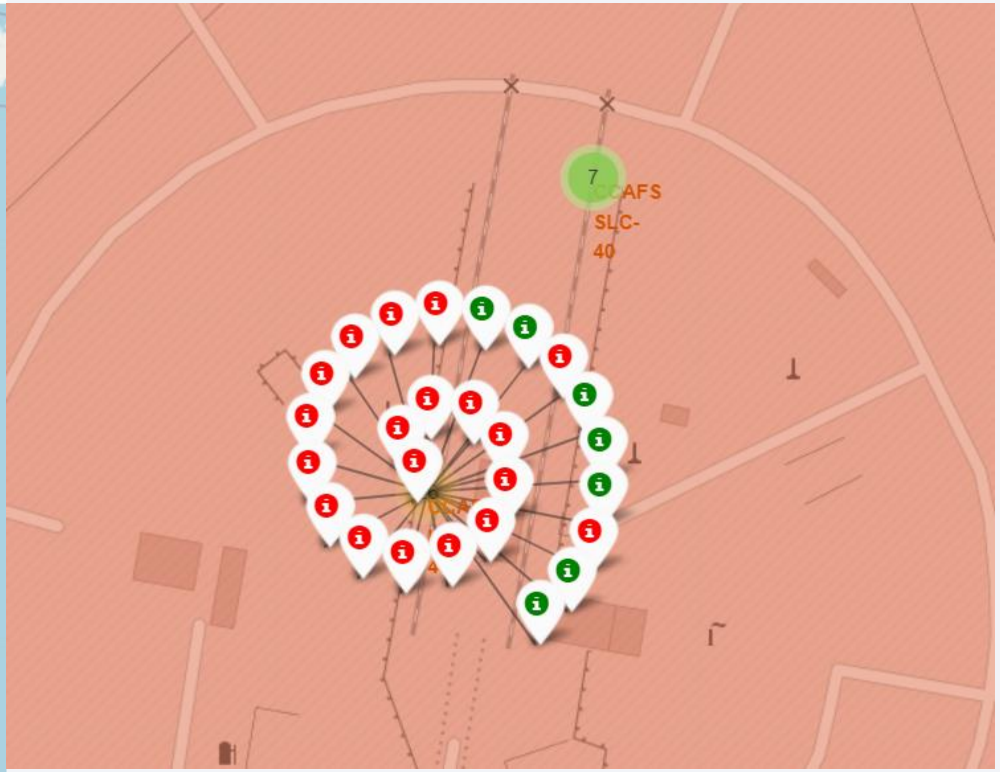
all launch sites on a map

Out[10]:

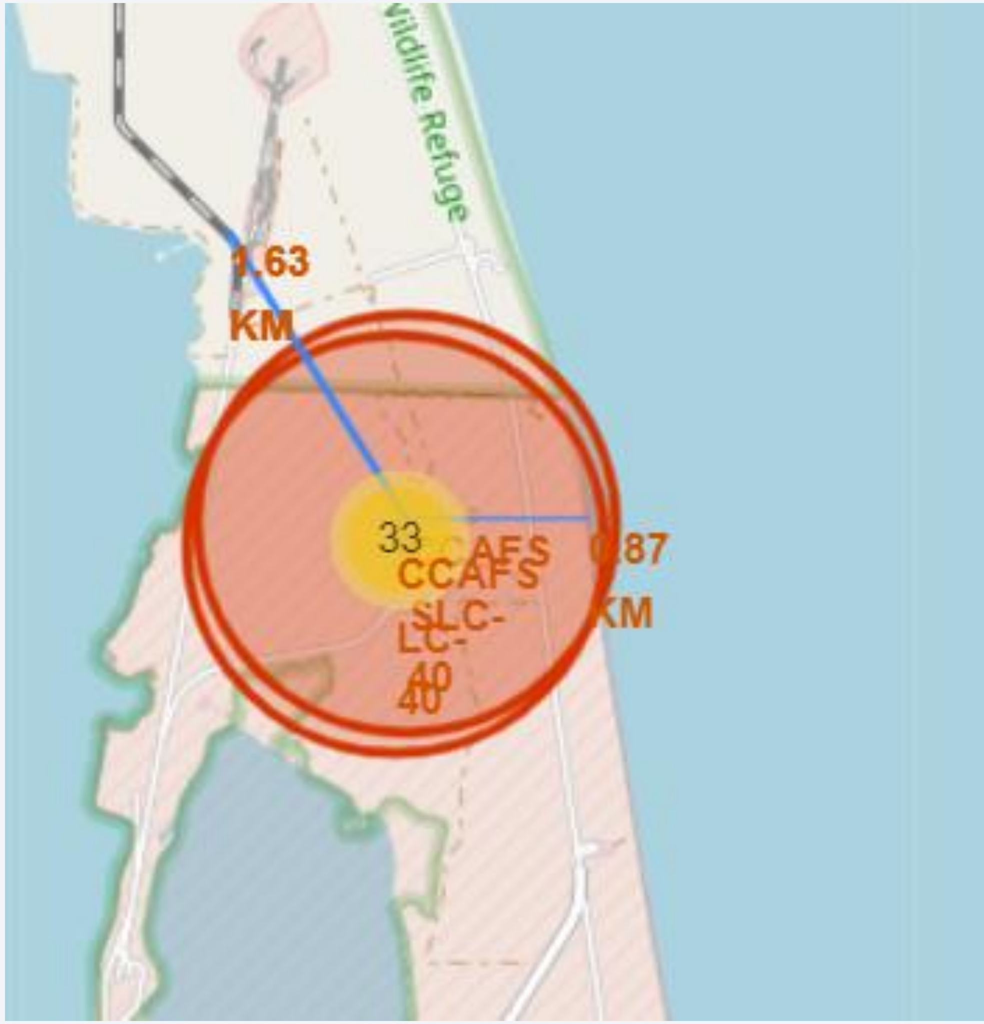


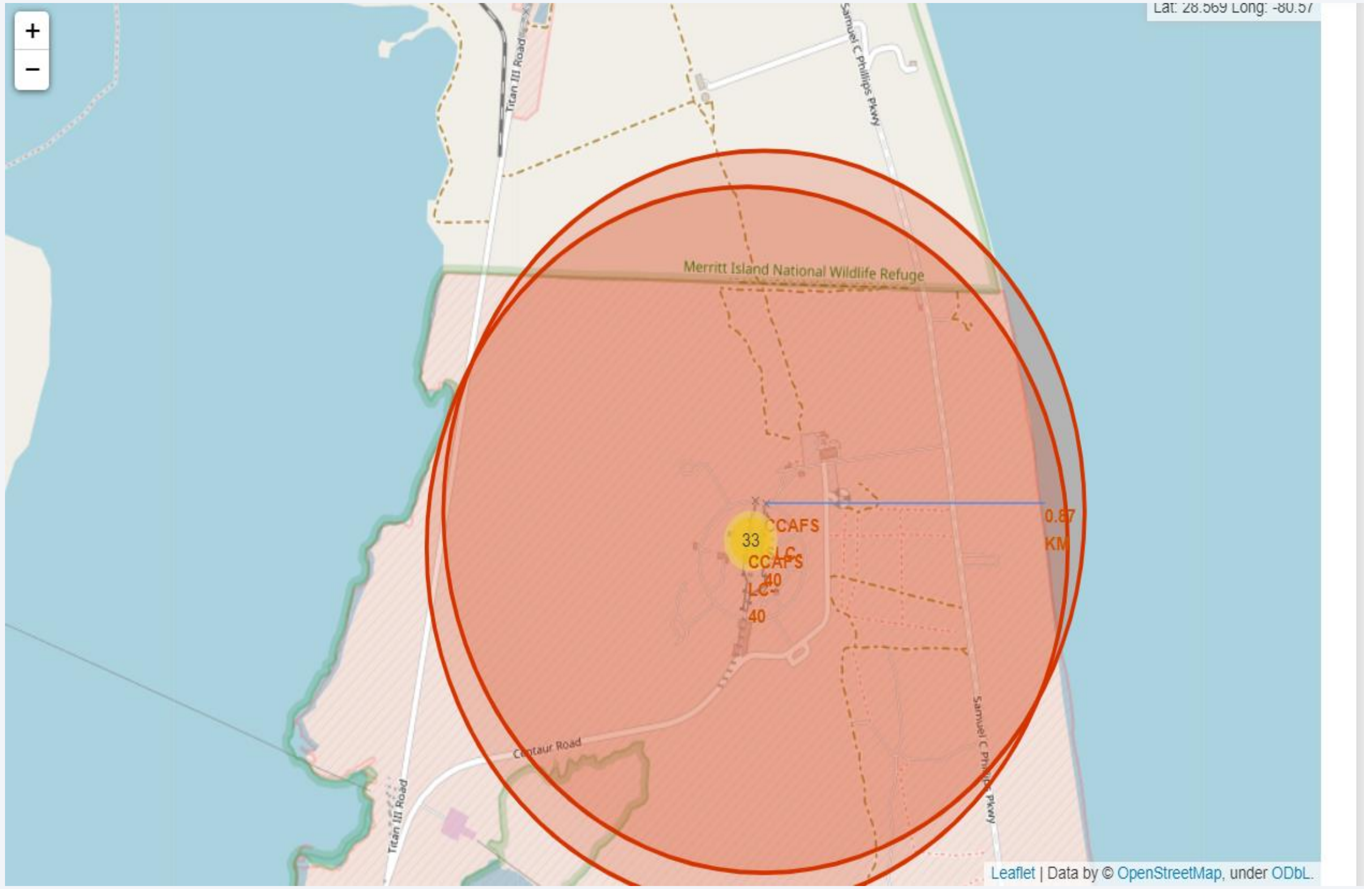
The success/failed launches for each site on the map





the distances between a launch site to its proximities

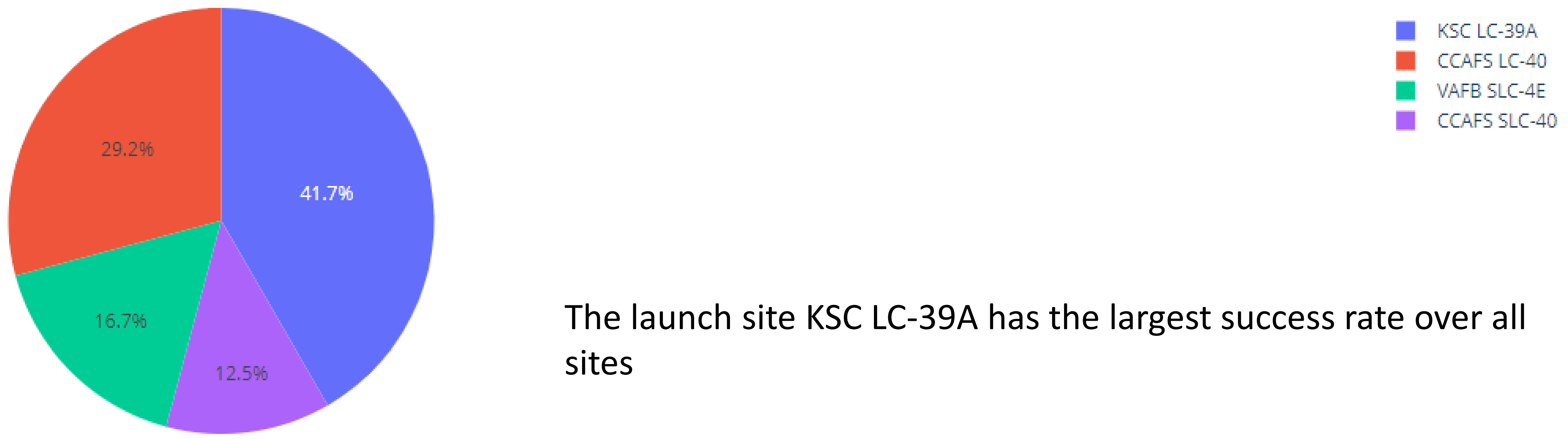




Section 4

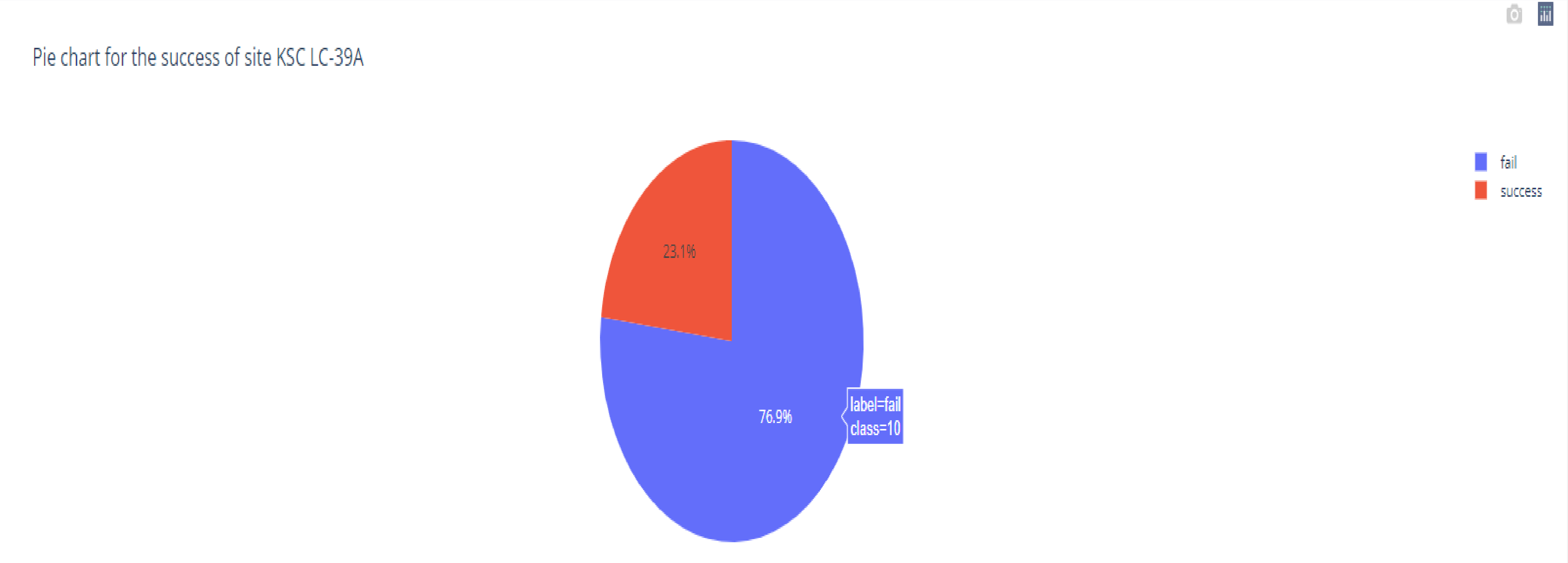
Build a Dashboard with Plotly Dash

Pie chart showing the success rate of all launch site



The launch site KSC LC-39A has the largest success rate over all sites

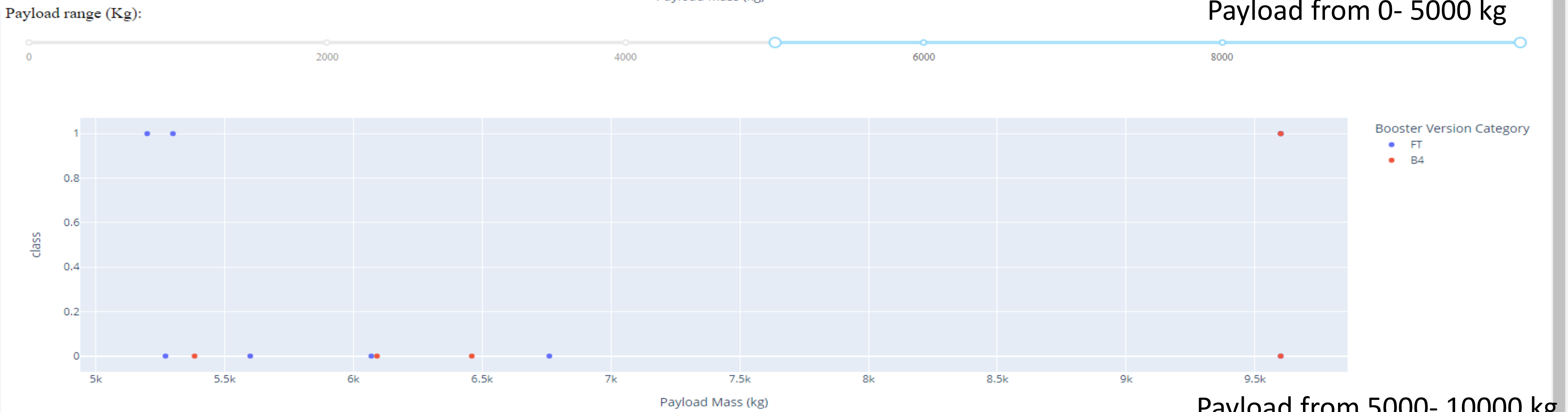
Pie chart showing the Launch site with the highest launch success ratio



The largest success is at launch site KSC LC-39A, where 23.1% has success over are tries

Payload vs. Launch Outcome scatter plot for all sites with different payload

Payload from 0- 5000 kg



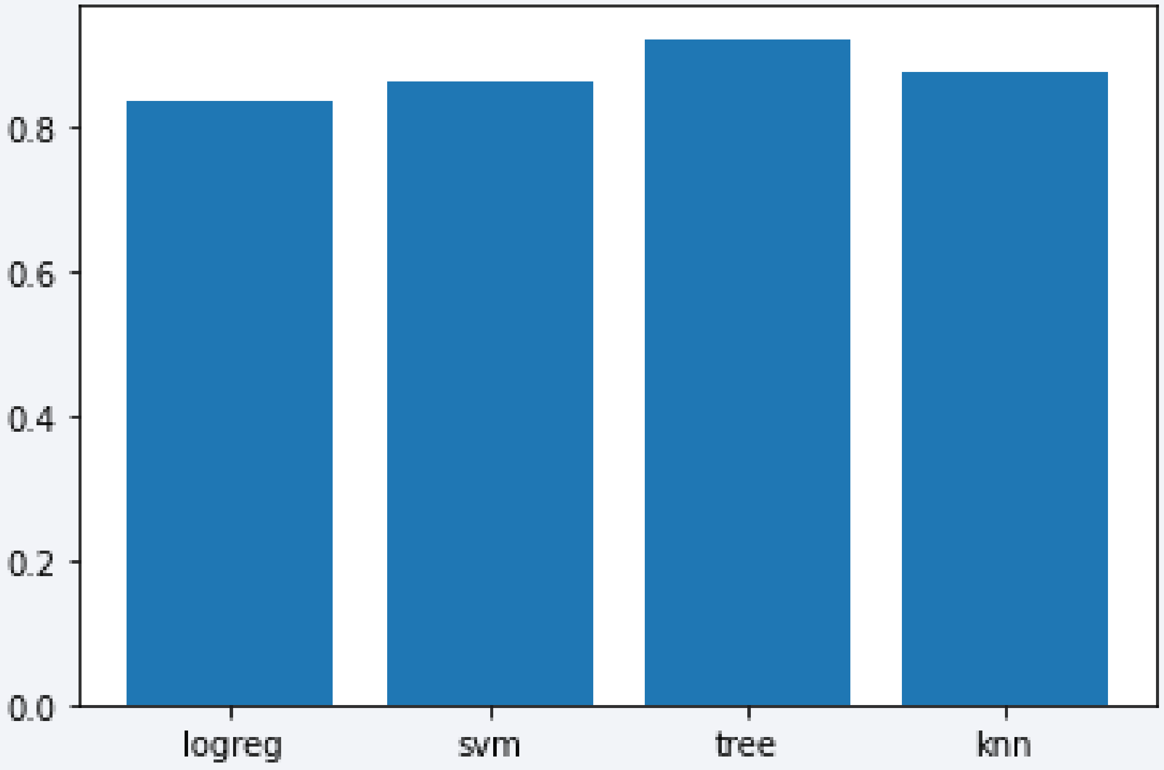
Payload from 5000- 10000 kg

Section 5

Predictive Analysis (Classification)

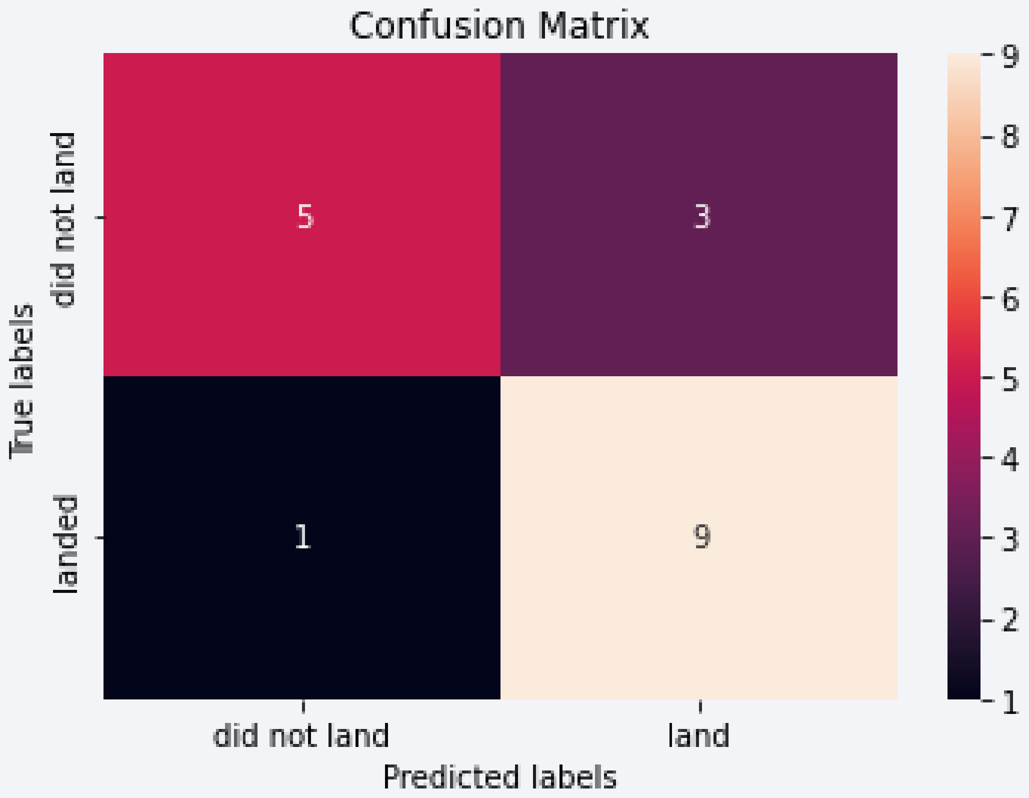
Classification Accuracy

- The accuracy of models as follows:
- 'logreg': 0.84,
- 'svm': 0.86,
- 'tree': 0.92,
- 'knn': 0.88
- It can be shown that the Decision tree has the best accuracy



Confusion Matrix

- the Decision tree has the best accuracy



Conclusions

- Launch success rate started to increase in 2013 till 2020.
- Orbits ES-L1, GEO, HEO and SSO have the most success rate.
- KSC LC-39A had the most successful launches of any sites.
- The Decision tree classifier is the best machine learning algorithm for this task.

Thank you!