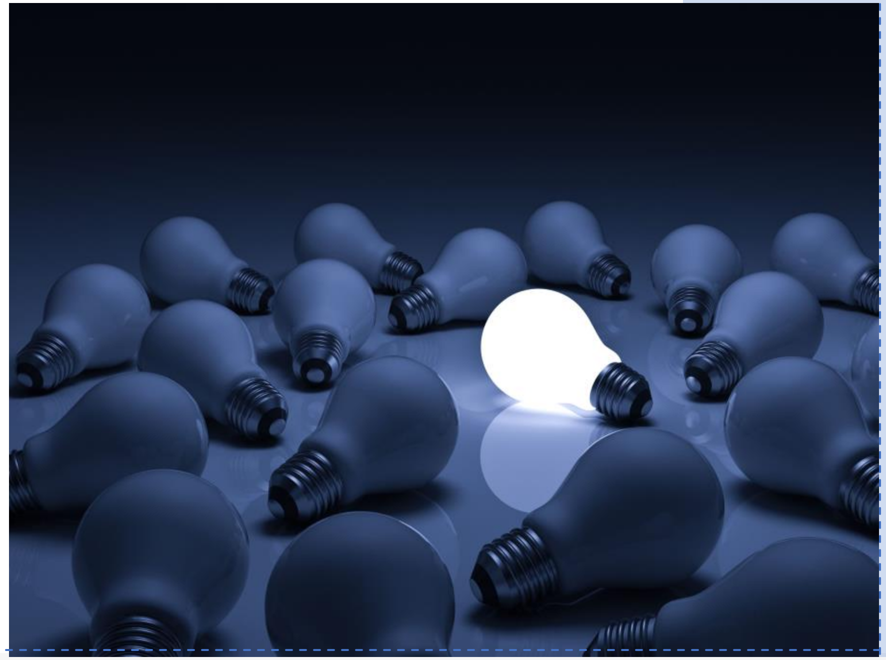
IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

ANNA LYAKHOVA
14 December 2021

Outline

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



Executive Summary

Summary of methodologies

Data Collection

Data Wrangling

EDA with SQL

EDA with data visualization

Building an Interactive map with Folium

Building an Interactive Dashboard
with Plotly Dash

Predictive Analysis (Classification)

Summary of all results

Exploratory Data Analysis results

Interactive Analytics (screenshots)

Predictive Analysis results

Introduction

Project background and context

We predicted whether the first stage of the SpaceX's Falcon 9 rocket would land successfully.

SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upwards 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.

Problems you want to find answers

- Relationship whit rockets variables outcome
- In what case the rocket will land successfully

Methodology

Data collection methodology:

- SpaceX REST API, Web scraping from [Wikipedia]

Perform data wrangling

- I converted landing outcomes to Classes (0 or 1) with using OneHotEncoder
- **Perform exploratory data analysis (EDA) using visualization and SQL**
- Plotting: scatter point chart, bar chart to show relationships between variables

Perform interactive visual analytics using Folium and Plotly Dash

- I built a dashboard to analyze launch records interactively with Plotly Dash and an interactive map to analyze the launch site proximity with Folium.

Perform predictive analysis using classification models

I split my data into training data and test data to find the best Hyperparameter for SVM, Classification Trees, and Logistic Regression. Then find the method that performs best using test data.

Data Collection

Getting Data from Api and Web page

Convert it to Dataframe

Clean and Filter Dataframe

Export it to a CSV file

Data Collection – SpaceX API

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

Getting response from API and **Converting response** to a .json file

Crate lists and Apply functions

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

Compose a dictionary from a list and create Dataframe

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

Filter Data and export to .csv file

```
data_falcon9 = df[df['BoosterVersion']!='Falcon 1']
data_falcon9.loc[:,'FlightNumber'] = list(range(1, data_falcon9.shape[0]+1))
data_falcon9.to_csv('dataset_part\_1.csv', index=False)
```

[GitHub]()

Data Collection - Scraping

Request the Falcon9 Launch Wiki page from its URL

```
static_url ='https://en.wikipedia.org ..'
response = requests.get(static_url)
```

Create a BeautifulSoup object from the HTML response

```
soup =
BeautifulSoup(response.content,
'html5lib')
```

Extract tables and get column names

```
html_tables = soup.find_all('tr')
first_launch_table = html_tables[2]
column_names = []
```

Create a data frame by parsing the launch HTML tables

```
launch_dict=
dict.fromkeys(column_names)
```

Convert Dictionary to Dataframe

```
df= pd.DataFrame({ key:pd.Series(val
ue) for key,
value in launch_dict.items()})
```

Export to .csv file

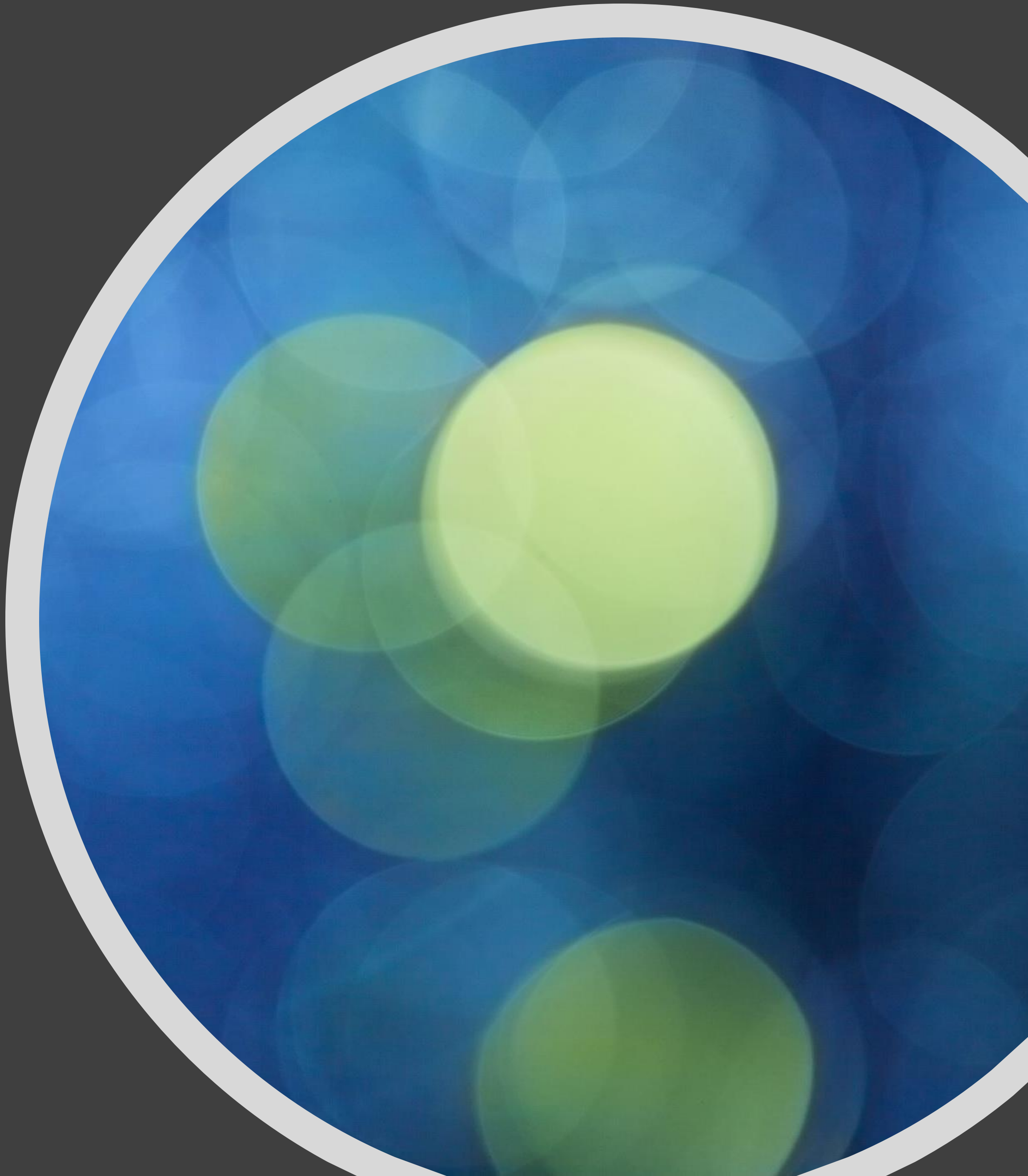
```
df.to_csv('spacex_web_scraped.csv',
index=False)
```

[GitHub]

Data Wrangling

Load	Calculate	Export	Create
Load Space X dataset	Calculate the number and occurence of mission outcome per orbit type	Export it to .csv file	Create a landing outcome label from Outcome column (boolean values)

GitHub



EDA with Data Visualization

Scatter point chart:

FlightNumber vs. PayloadMassand

FlightNumber vs LaunchSite

Payload vs. Launch Site

FlightNumber vs Orbit type

Payload vs. Orbit

Bar chart:

Success rate of each orbit type

Line chart

launch success yearly trend

GitHub

Link

EDA with SQL

1. *Display the names of the unique launch sites in the space mission*
2. *Display 5 records where launch sites begin with the string 'CCA'*
3. *Display the total payload mass carried by boosters launched by NASA (CRS)*
4. *Display average payload mass carried by booster version F9 v1.1*
5. *List the date when the first successful landing outcome in ground pad was acheived*
6. *List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000*
7. *List the total number of successful and failure mission outcomes*
8. *List the names of the booster_versions which have carried the maximum payload mass.*
9. *List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015*
10. *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]()

MAP Object	Code	Explanation
Circle	Folium.Circle()	Create circle area with a text label on a specific coordinate where Marker is being placed
Marker	folium.map.Marker()	To make a mark on Map
Marker Cluster	MarkerCluster()	Marker clusters can be a good way to simplify a map containing many markers having the same coordinate.
Mouse Position	MousePosition()	To get coordinate for a mouse over a point on the map
PolyLine	folium.PolyLine()	Create the line between points

- [GitHub]

Build an Interactive Map with Folium

Map Object	Code	Explanation
Dropdown	dcc.Dropdown()	Add a dropdown list to enable Launch Site selection
Range Slider	dcc.RangeSlider()	Add a slider to select payload range
Pie Chart	dcc.Graph(id='success-pie-chart')	Add a pie chart to show the total successful launches count for all sites
Scatter Chart	dcc.Graph(id='success-payload-scatter-chart')	Add a scatter chart to show the correlation between payload and launch success

- GitHub

Build a Dashboard with Plotly Dash

Predictive Analysis (Classification)

Building model

Load DataFrame

Crate a NumPy array from the column Class in data

Standardize and transform the data

Split the data into training and testing data

Create a logistic regression object then create a GridSearchCV object . Then fit the model

Evaluating Model

Find accuracy for each model

Find best parameters

Plot confusion matrixes

Find the method performs best

Finding the model with best accuracy using the method score

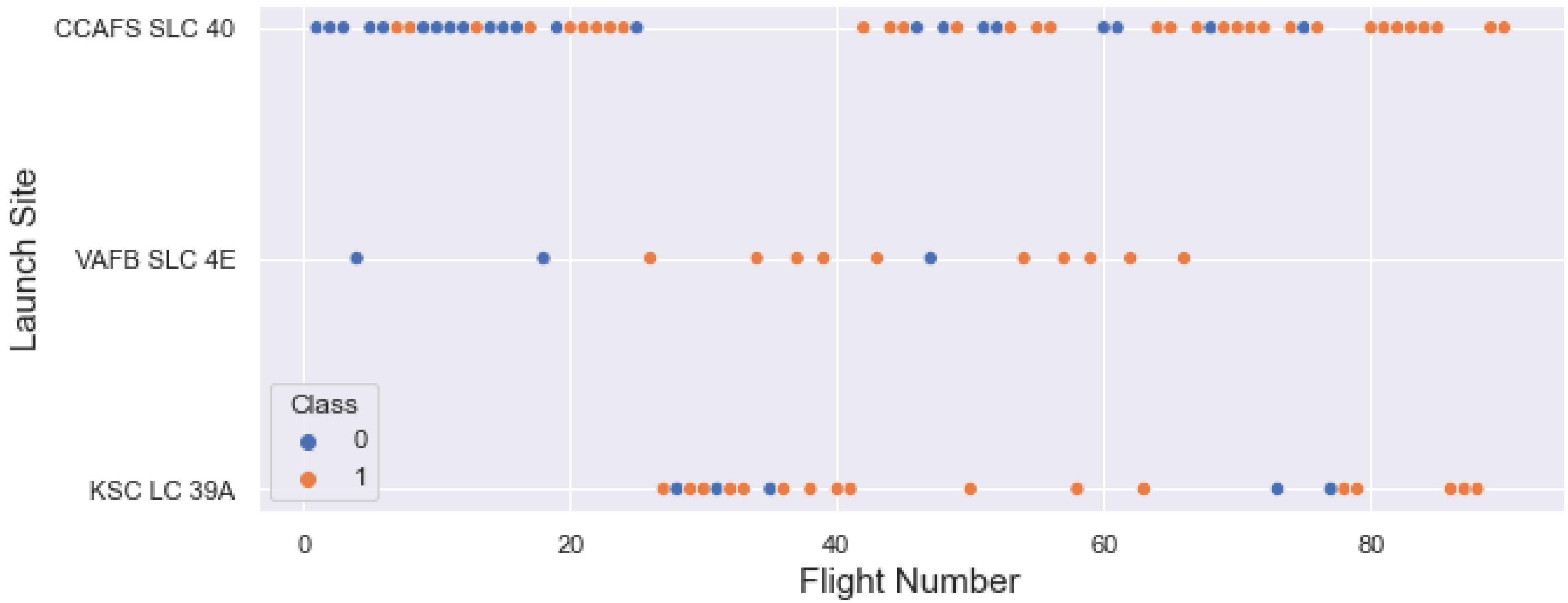
[GitHub]

Results

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

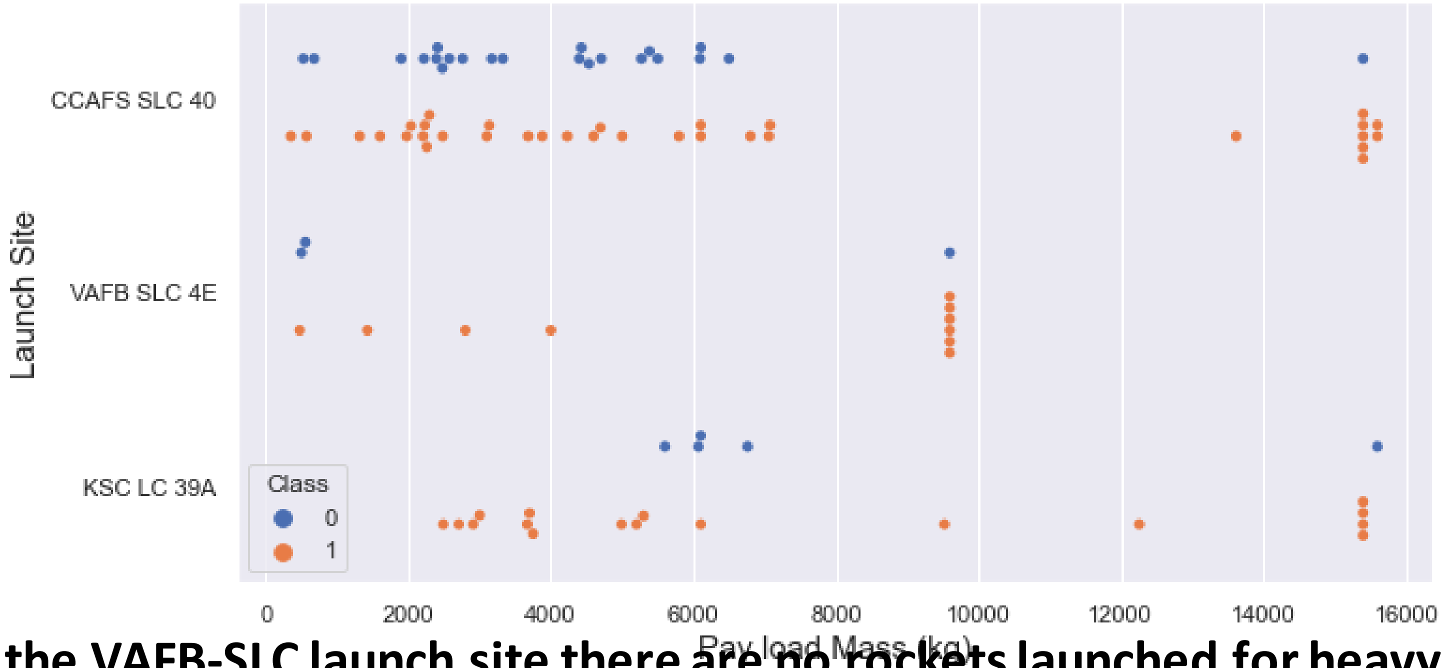
EDA with Visualization

Flight Number vs. Launch Site



- With higher Flight number the success Launch site is increasing

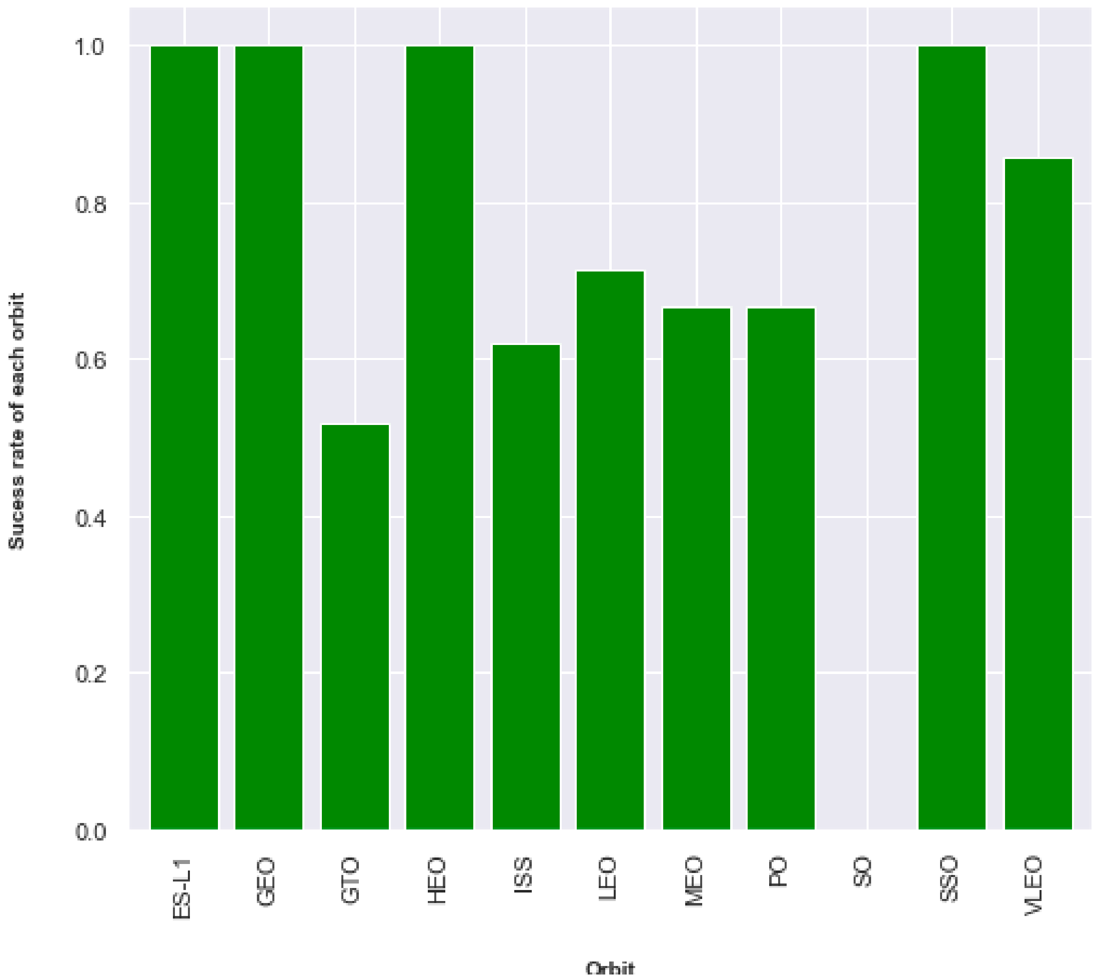
Payload vs. Launch Site

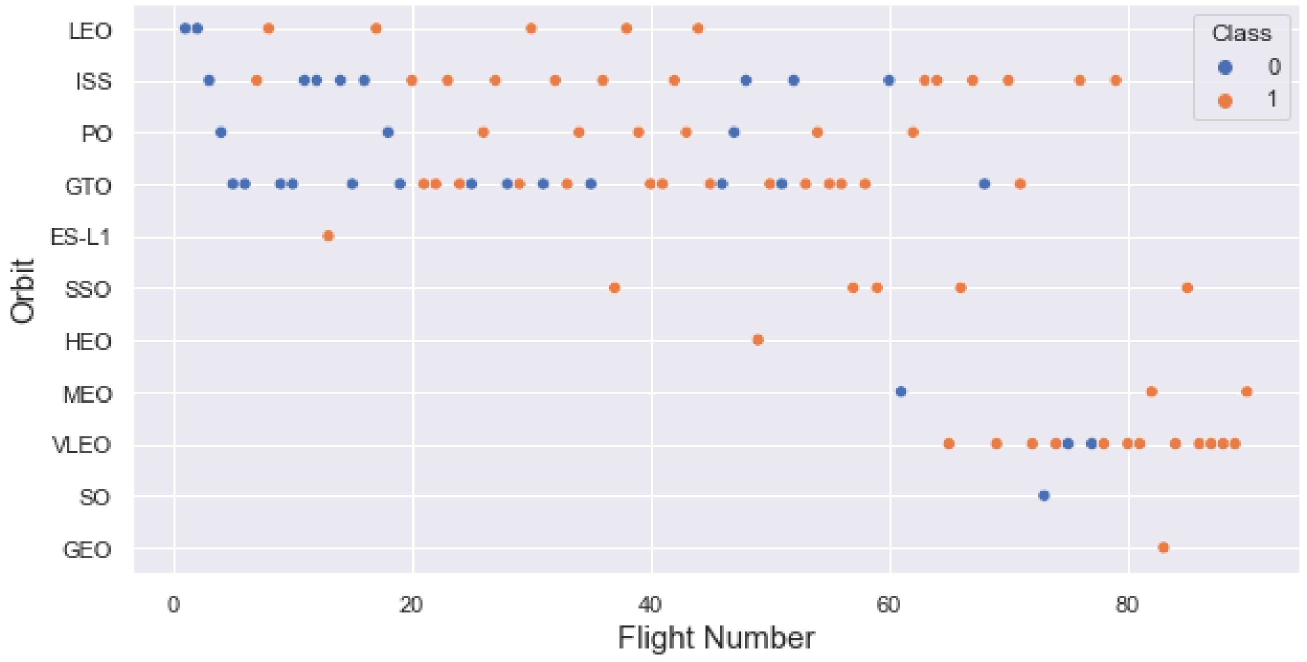


- **For the VAFB-SLC launch site there are no rockets launched for heavy payload mass(greater than 10000).**
- **And the greater the Payload mass, the higher the success rate**

Success Rate vs. Orbit Type

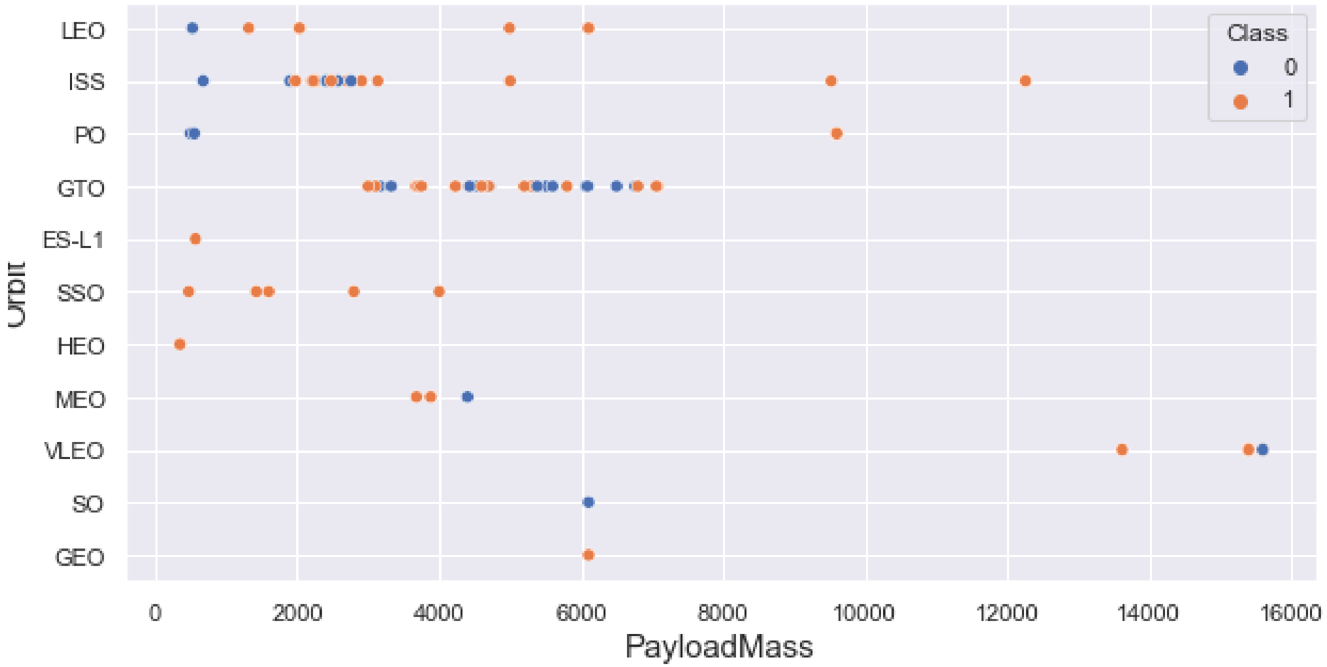
- ES-L1, GEO, HEO and SSO has highest success rates





- We see 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.

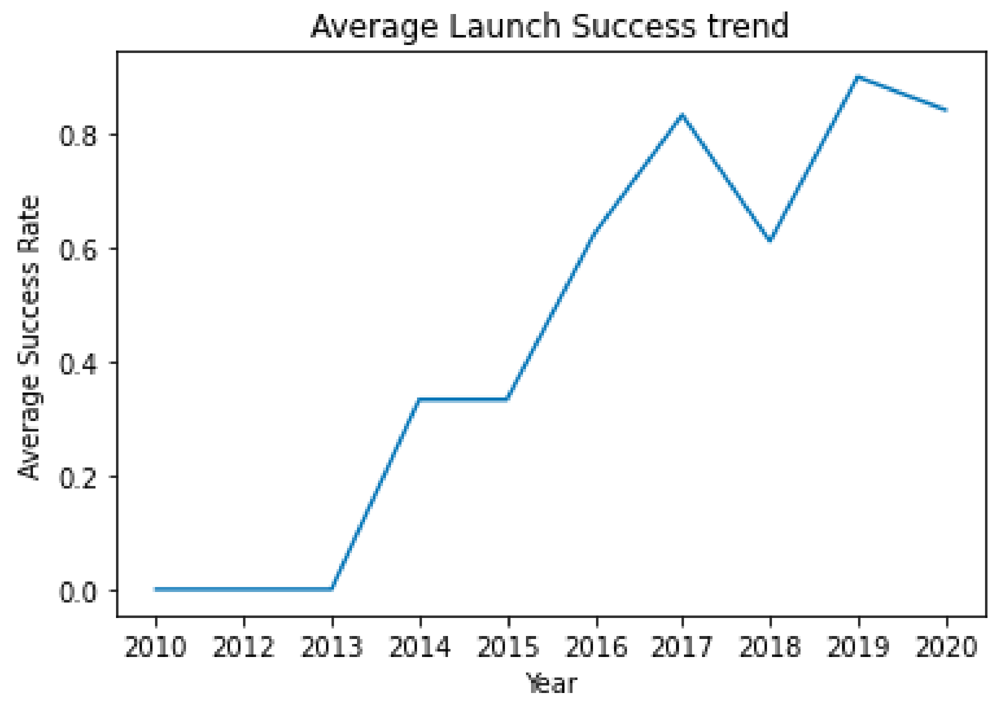
Flight Number vs. Orbit Type



- 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

Payload vs. Orbit Type

Launch Success Yearly Trend



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

EDA with SQL

All Launch Site Names

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

%sql select DISTINCT LAUNCH_SITE from SPACEXTBL

Using DISTINCT we pull unique values for LAUNCH_SITE from SPACEXTBL

Launch Site Names Begin with 'CCA'

%sql select * from SPACEXTBL where LAUNCH_SITE like 'CCA%' LIMIT 5

LIMIT is used for restricting the number of rows (5) retrieved from the SPACEXTBL

Keyword 'like' means that LAUNCH_SITE name must start with 'CCA

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

%sql select sum(PAYLOAD_MASS__KG_) from SPACEXTBL where CUSTOMER='NASA (CRS)'

- Sum calculated the total in (PAYLOAD_MASS__KG_)
- Where clause with the predicate CUSTOMER by name 'NASA (CRS)'

Total Payload Mass by NASA (CRS)
45596

Average Payload Mass by F9 v1.1

- %sql select AVG(PAYLOAD_MASS__KG_) from SPACEXTBL where BOOSTER_VERSION = 'F9 v1.1'
- AVG find the average in (PAYLOAD_MASS__KG_)

Average Payload Mass by Booster Version F9 v1.1
2928

First Successful Ground Landing Date

- %sql select MIN(DATE) from SPACEXTBL where LANDING__OUTCOME = 'Success (ground pad)'
- MIN find the minimum in Date

First Succesful Landing Outcome in Ground Pad
2015-12-22

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

Successful Drone Ship Landing with Payload between 4000 and 6000

- %sql select BOOSTER_VERSION from SPACEXTBL where LANDING_OUTCOME='Success (drone ship)' and PAYLOAD_MASS__KG_ between 4000 and 6000
- AND clause specifies additional filter conditions

Total Number of Successful and Failure Mission Outcomes

- **%sql** select COUNT(MISSION_OUTCOME) as "Total Number of Successful and Failure Mission" FROM SPACEXTBL \
- where MISSION_OUTCOME like 'Success%' or MISSION_OUTCOME like 'Failure%'
- I used sub query to get success and failure counts

Successful Mission	Failure Mission
100	1

Boosters Carried Maximum Payload

- %sql select DISTINCT BOOSTER_VERSION as "Booster Versions which carried the Maximum Payload Mass" from SPACEXTBL where PAYLOAD_MASS__KG_=(select MAX(PAYLOAD_MASS__KG_) from SPACEXTBL)
- MAX finding the maximum in (PAYLOAD_MASS__KG_)

Booster Versions which carried the Maximum Payload Mass
F9 B5 B1048.4
F9 B5 B1048.5
F9 B5 B1049.4
F9 B5 B1049.5
F9 B5 B1049.7
F9 B5 B1051.3
F9 B5 B1051.4
F9 B5 B1051.6
F9 B5 B1056.4
F9 B5 B1058.3
F9 B5 B1060.2
F9 B5 B1060.3

2015 Launch Records

- %sql select {fn MONTHNAME(DATE)} BOOSTER_VERSION, LAUNCH_SITE from SPACEXTBL where DATE like '2015%' and Landing_Outcome='Failure (drone ship)'

- Using {fn MONTHNAME(DATE)} we get the month name

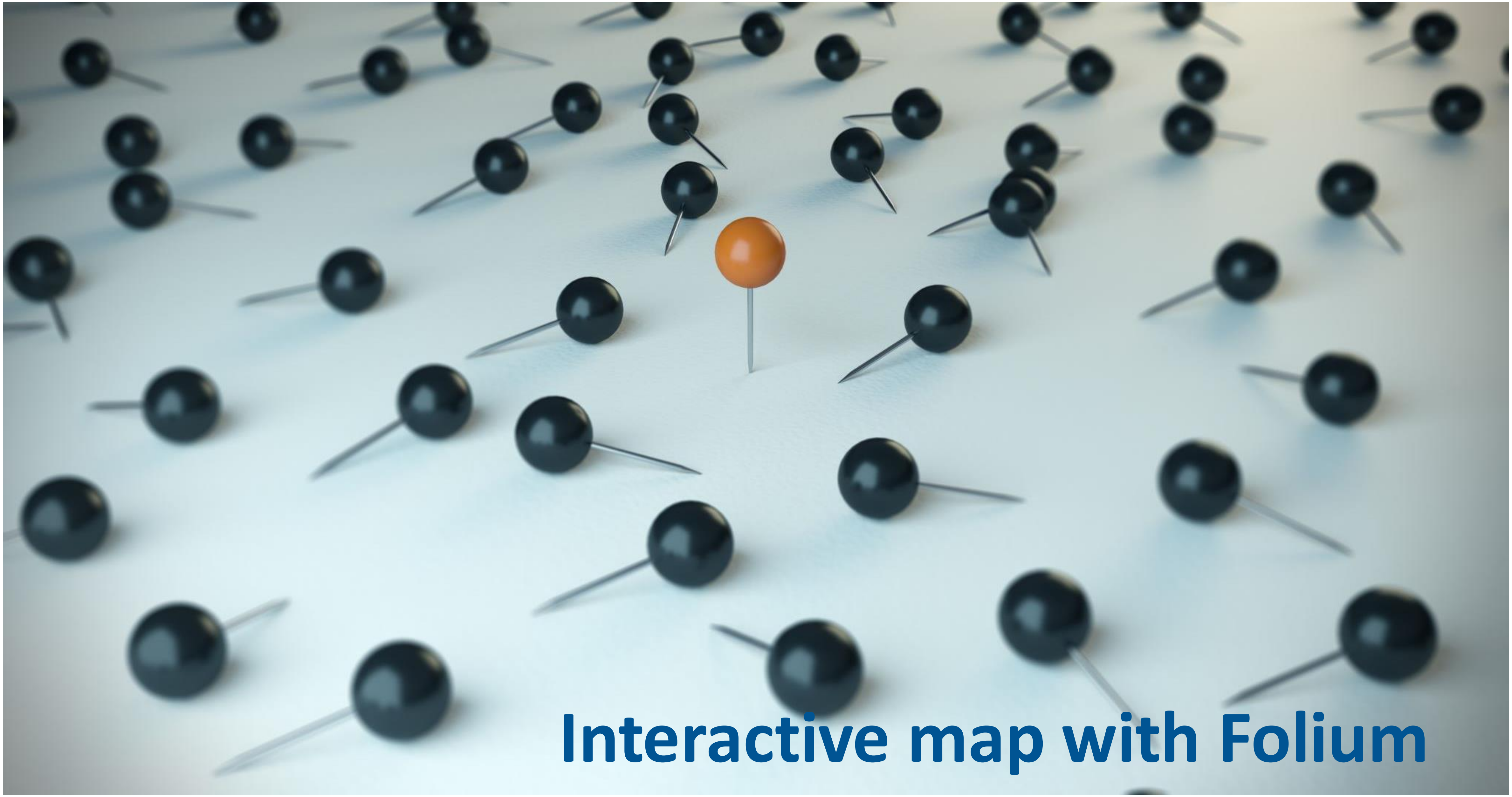
Month	booster_version	launch_site
January	F9 v1.1 B1012	CCAFS LC-40
April	F9 v1.1 B1015	CCAFS LC-40

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

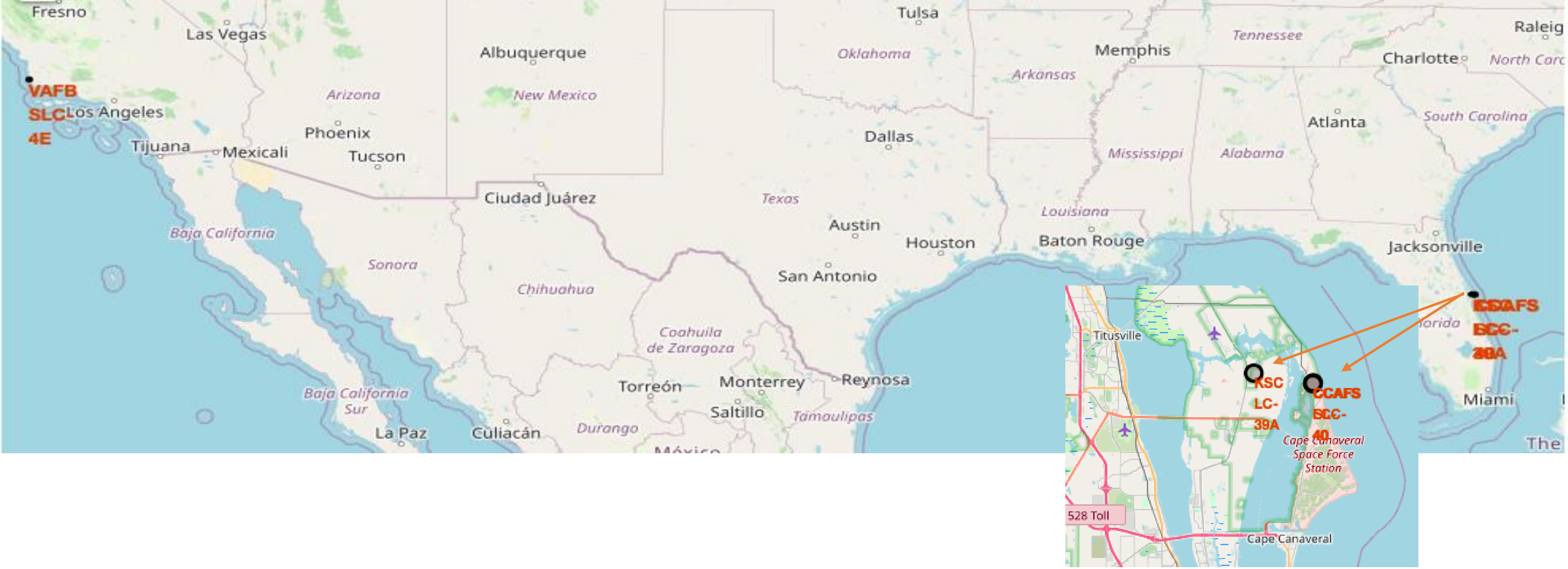
- %sql select Landing_Outcome as 'Landing Outcome', \COUNT(Landing_Outcome) as 'Total Count'from SPACEXTBL \where DATE between '2010-06-04' and '2017-03-20' \group by Landing_Outcome\order by COUNT(Landing_Outcome) DESC

- Grouping by Landing_Outcome and Ordering by COUNT(Landing_Outcome) in Descending order

Landing Outcome	Total Count
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1



Interactive map with Folium



We can see that SpaceX launch sites are in Florida and California regions (near the costs)

All launch sites on map

Color mark the success/failed launches for each site on the map

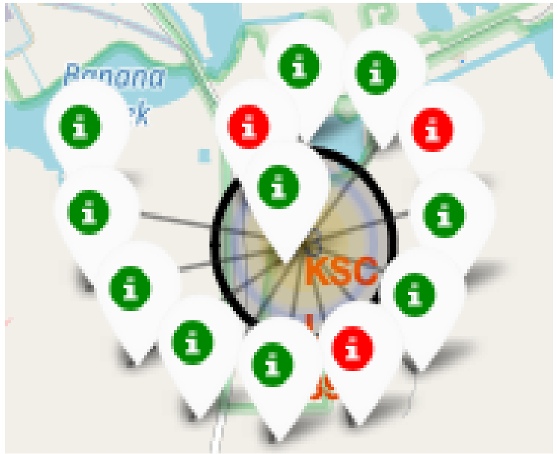
 Shows successful launches

 Shows failures

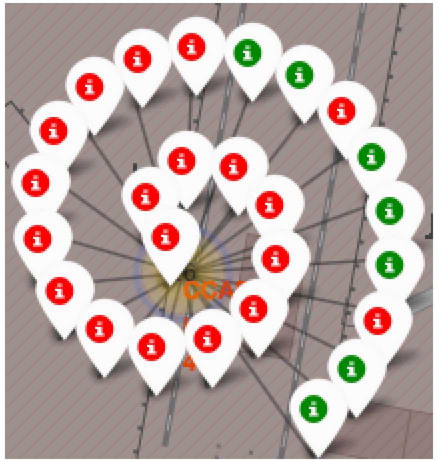
In this slides we can see that site KSL LC 40 has high number of successful launches

VAF SLC 4E

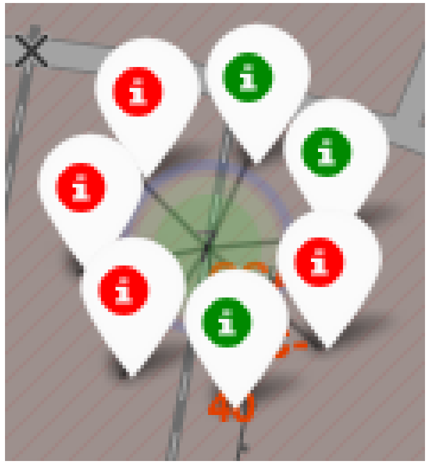
KSC LC 39A

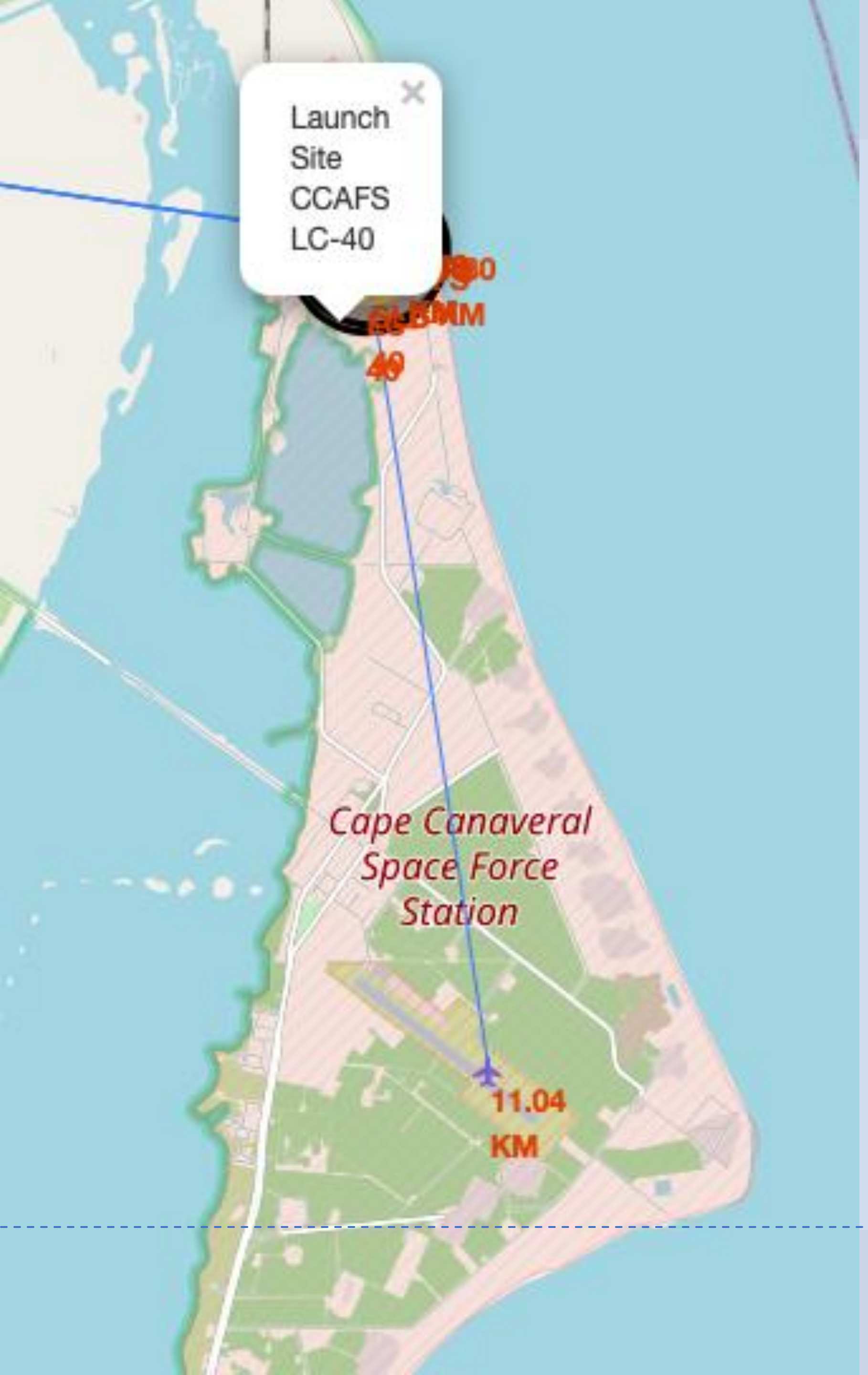


CCAFS SLC-40



CCAFS SLC-40

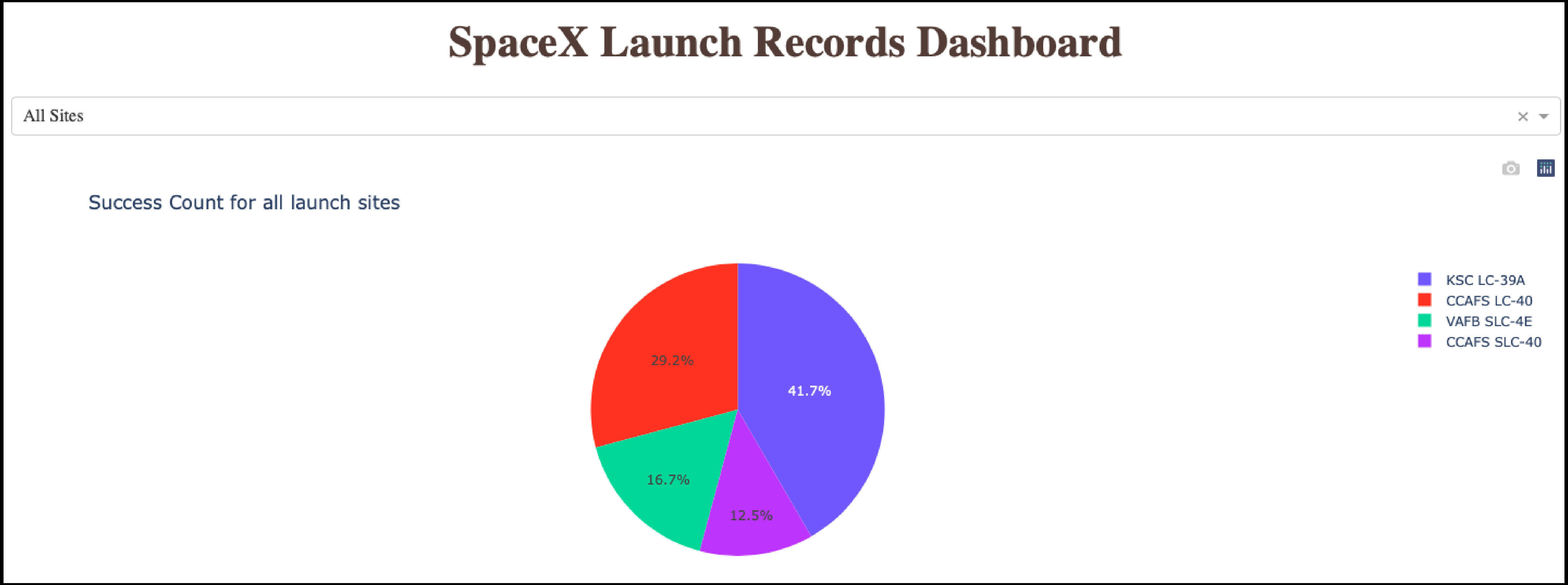




Distance between a launch site to Cape Canaveral SFS Skid Strip

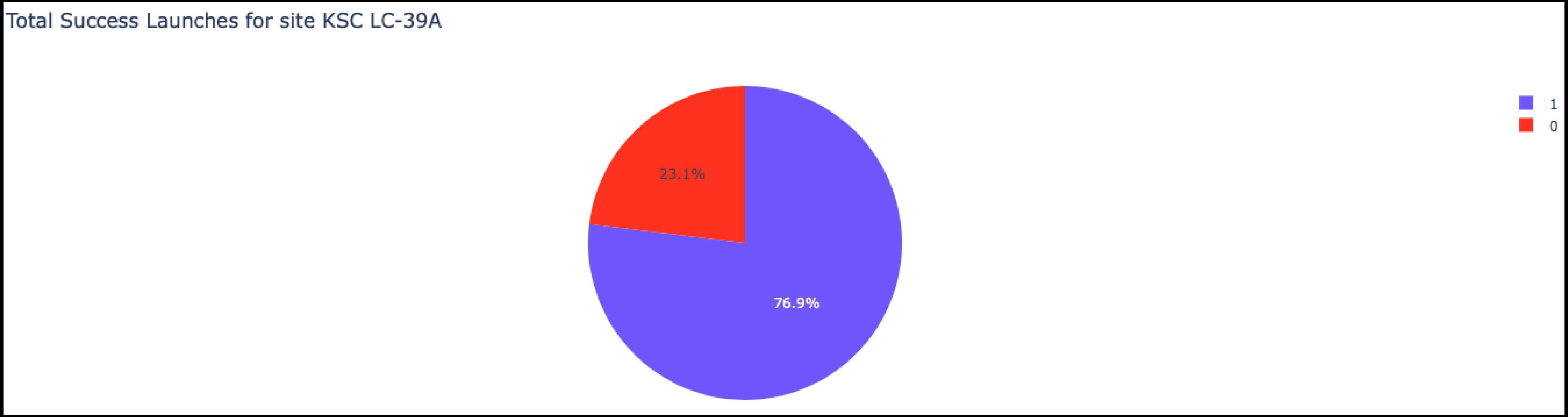
Distance from Launch Site CCAFS LC-40 more than 11 km

Build Dashboard with Plotly Dash



Launch success count for all sites

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

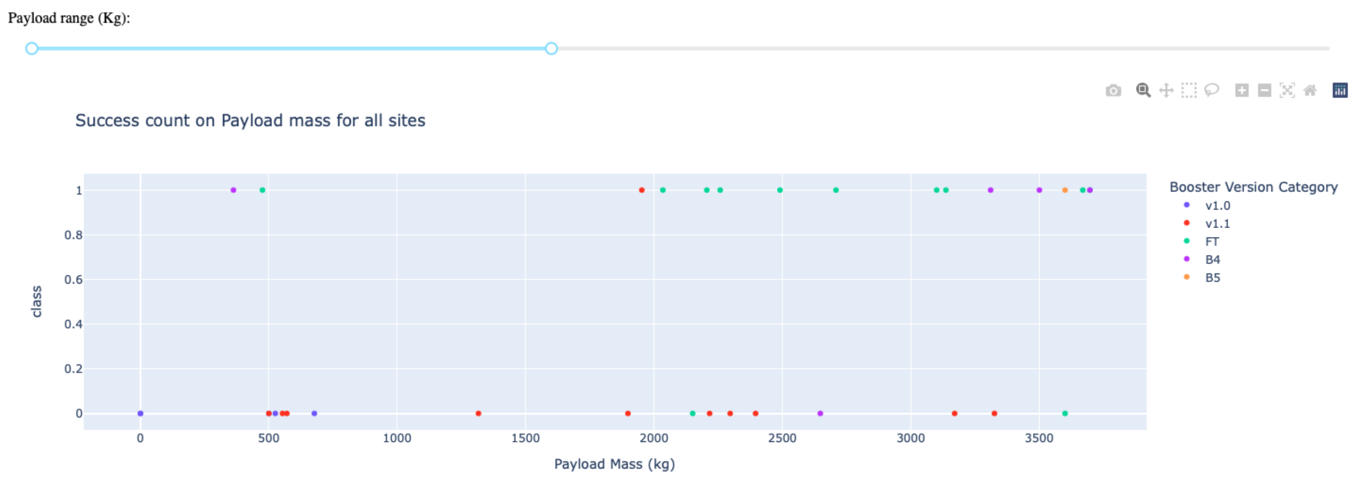


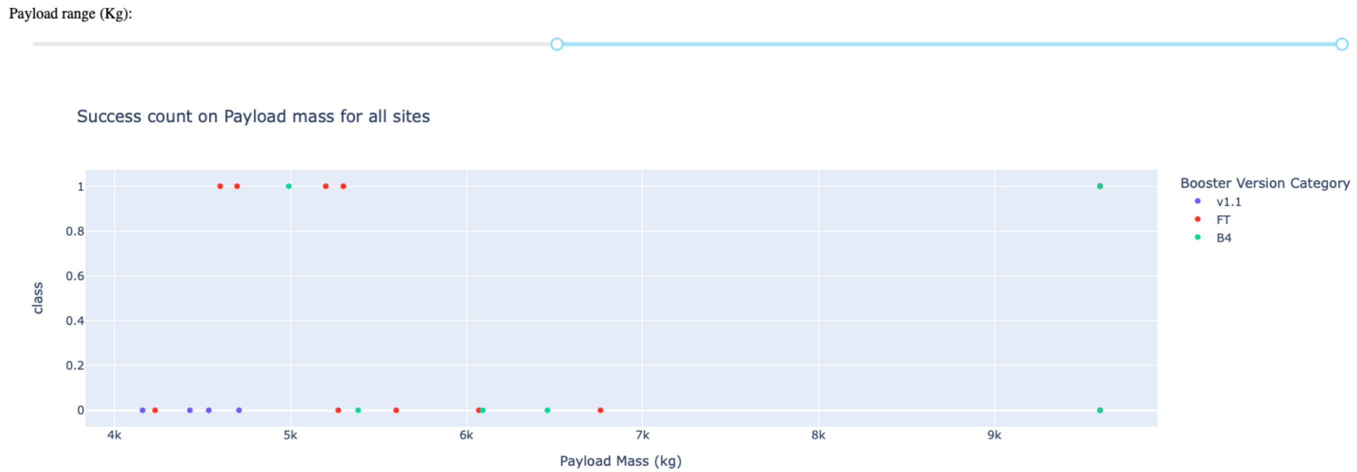
Launch site with highest launch success ratio

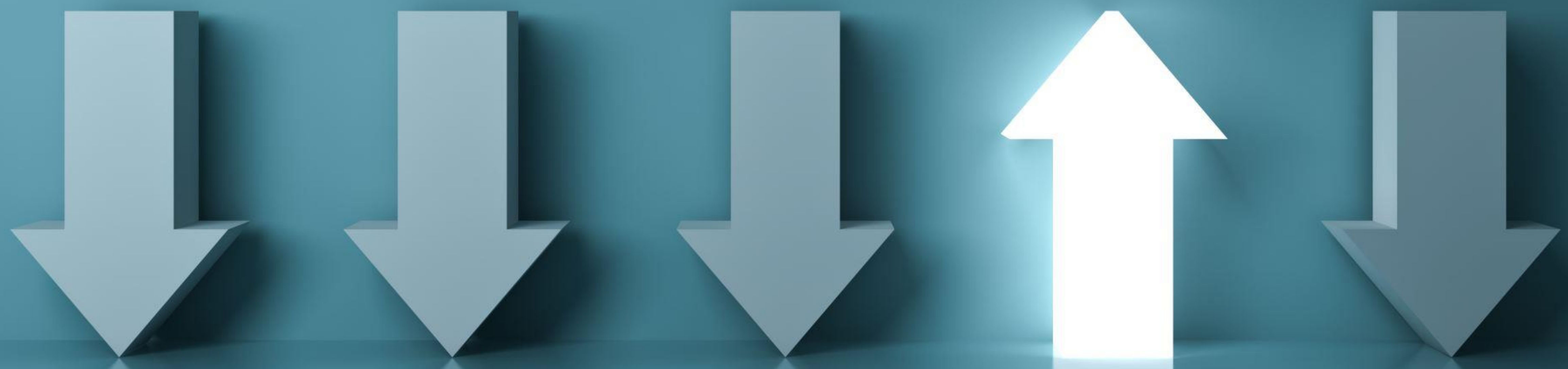
- KSC LC 39 has 76,9% success rate and 23,1% failure rate

Payload vs. Launch Outcome Scatter Plot for all sites

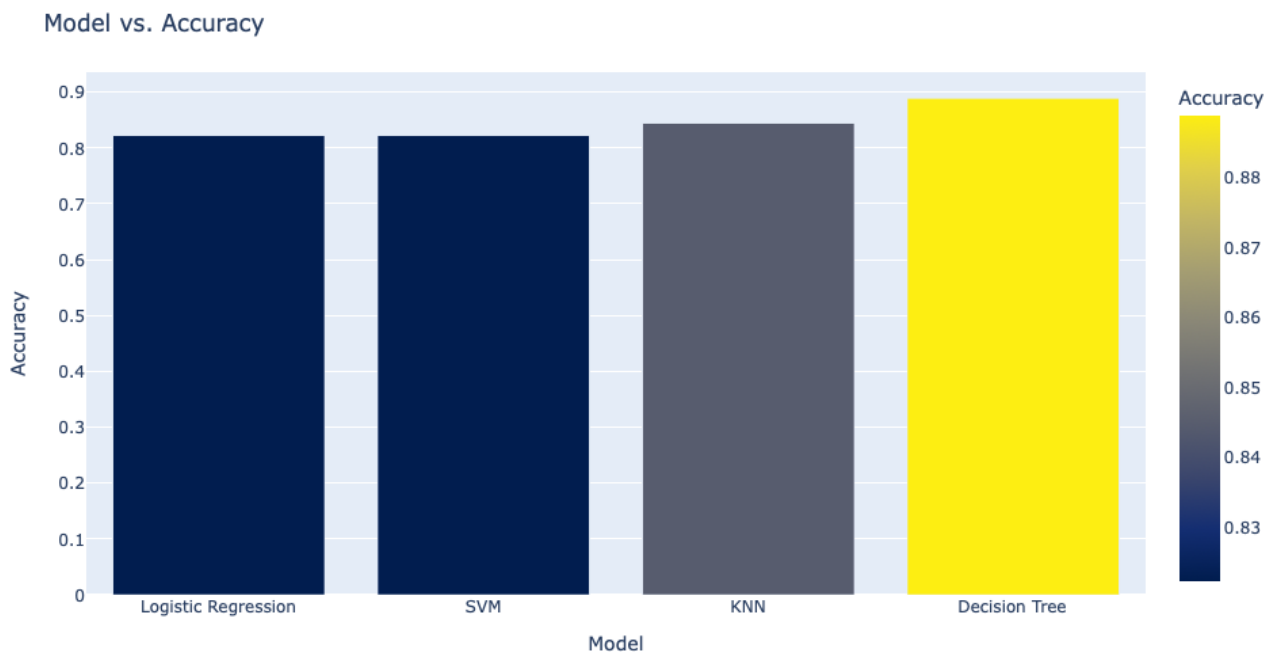
We can see for low weighted Payload the success rate higher than for heavy weighted Pa







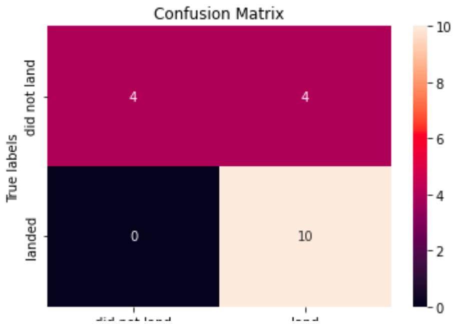
Predictive Analysis (Classification)

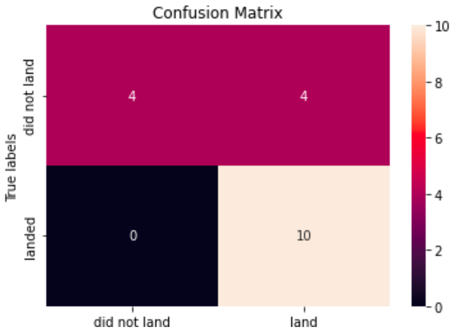


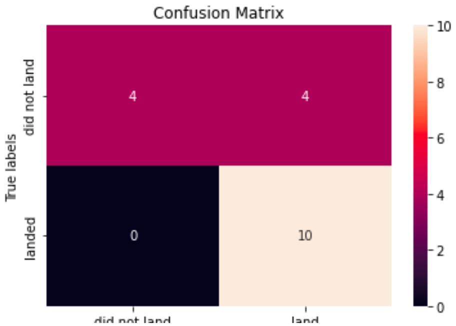
- We can see the Decision Tree
- model has highest accuracy

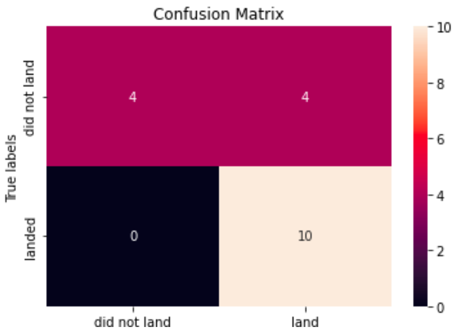
Classification Accuracy

Confusion Matrix





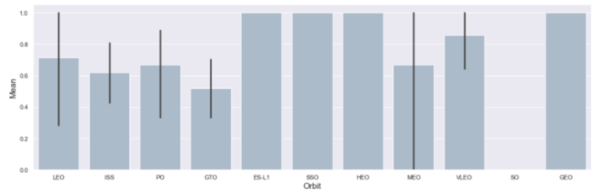




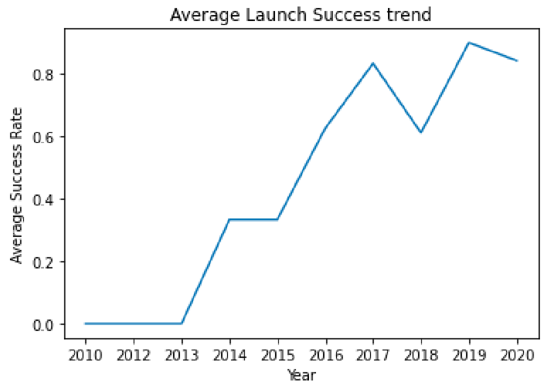
- Practically all these algorithms give the same Confusion Matrix

Conclusions

- The Orbits ES L1, SSO, HEO, GEO has highest success rates



- The sucess rate since 2013 kept increasing till 2020



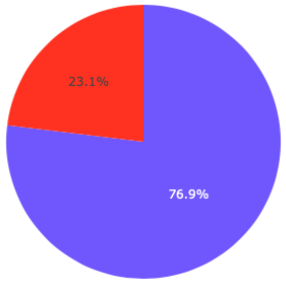
Conclusions

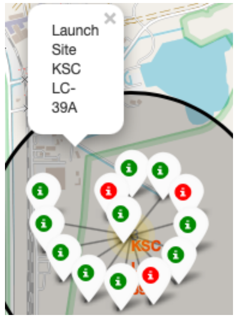
3. KSC LC 39 has 76,9% success rate

and 23,1% failure rate.

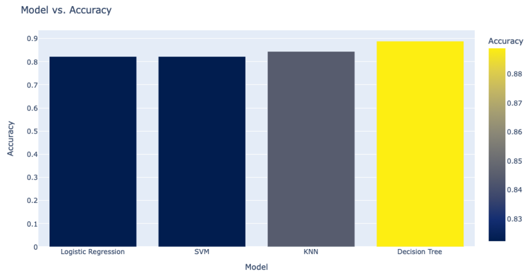
This the best result from all

Sites





4. The Decision Tree model has highest accuracy



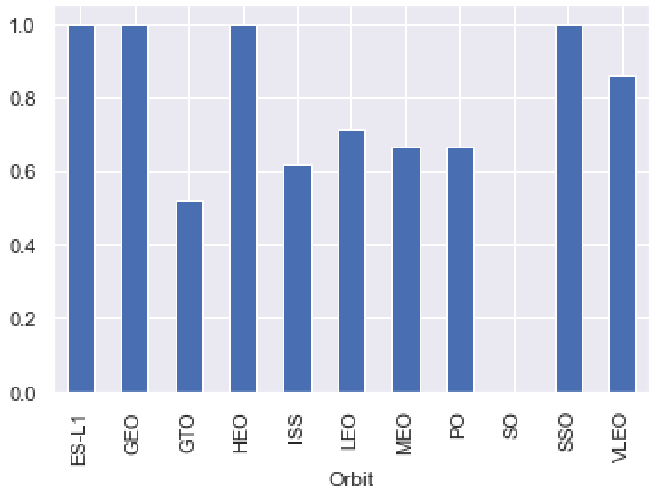
Appendix

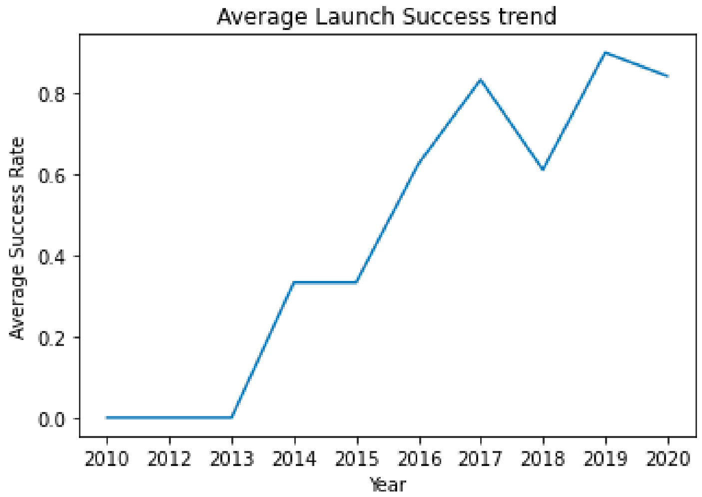
Interactive Plotly

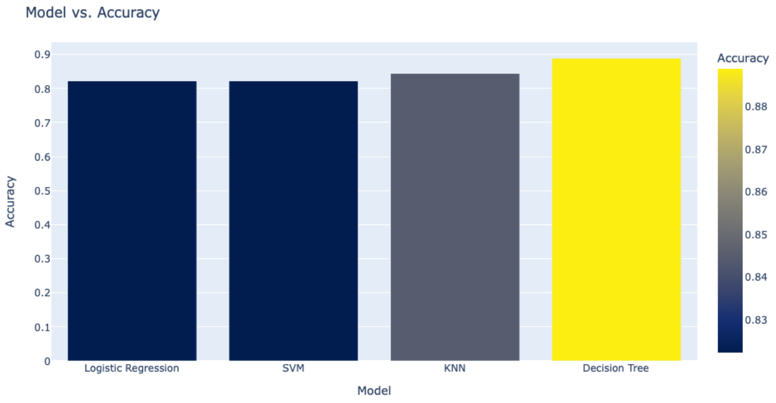
Folium Mouse Position and MarkerCluster plugins Tool

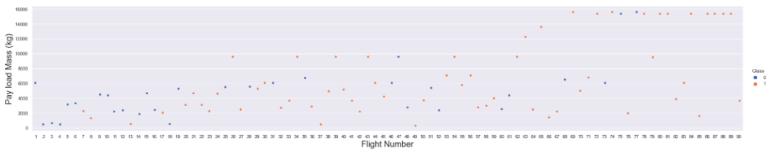
Basic Decision Tree Constructor

Interactive Plotly









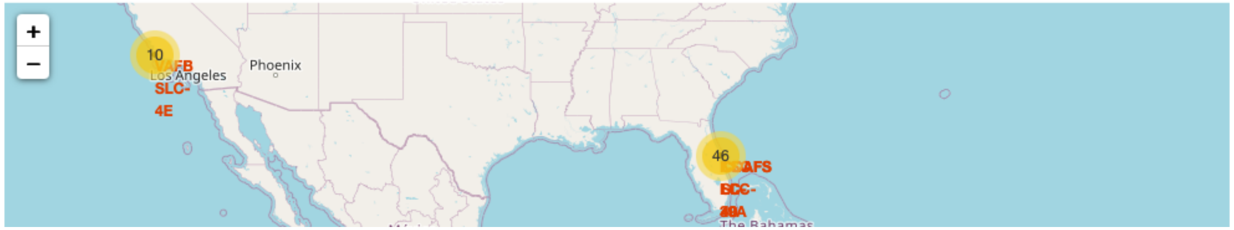
Helps to explore data

Folium Mouse Position and Marker Cluster plugins Tool

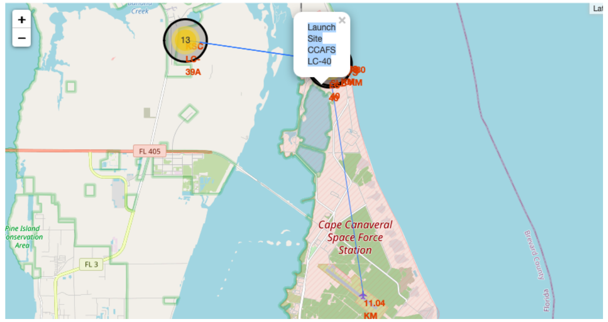
Help mark point and distance on map

```
    marker = folium.Marker([row['Lat'], row['Long']],icon=folium.Icon(color='white', icon_color=row['marker_color'])).
add_to(marker_cluster)
    marker_cluster.add_child(marker)

site_map
```

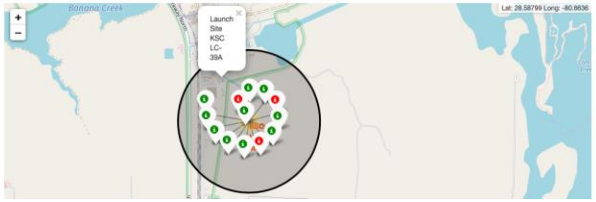



```
lines_airport=folium.PolyLine(locations=([28.562302, -80.577356], [28.46416, -80.56068]), weight=1)
site_map.add_child(lines_airport)
site_map
```

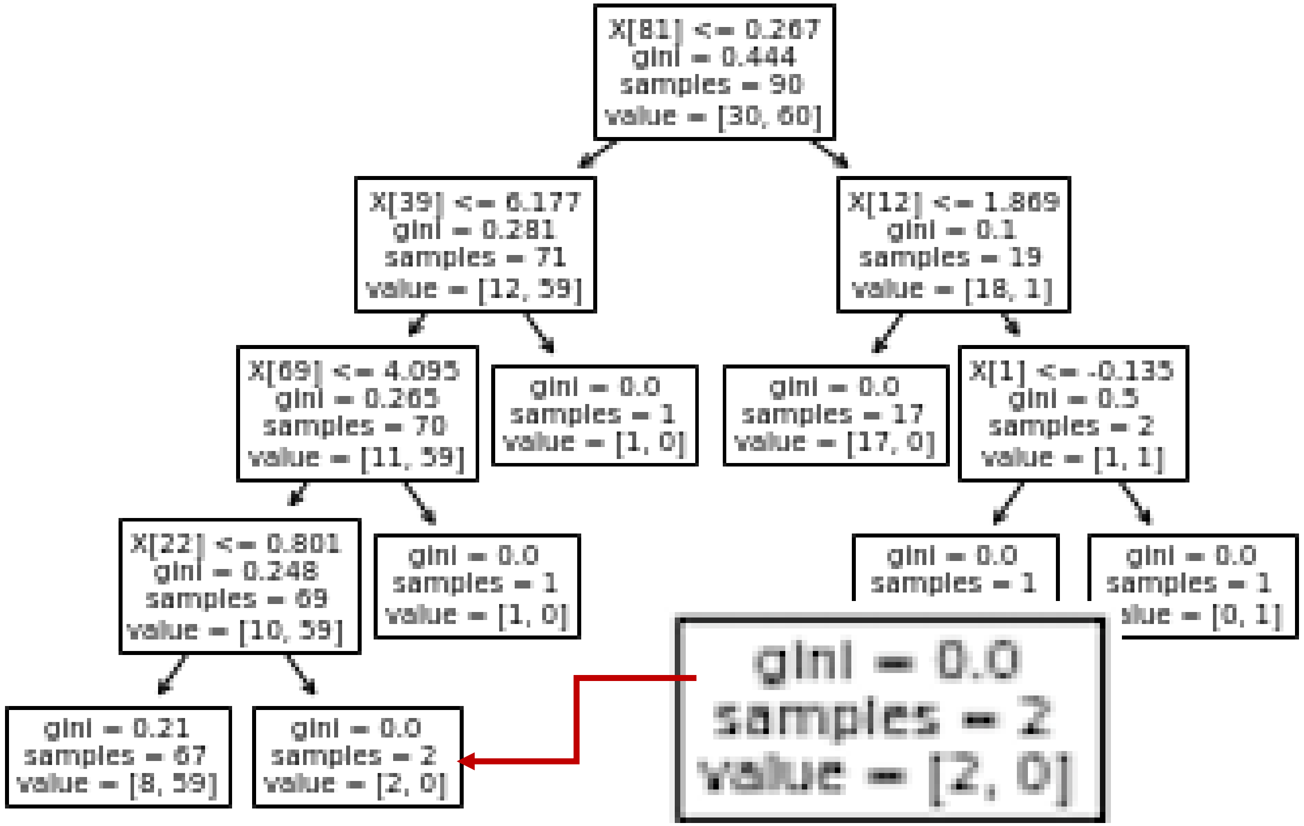



```
mouse_position = MousePosition(
    position='topright',
    separator=' Long: ',
    empty_string='NaN',
    lng_first=False,
    num_digits=20,
    prefix='Lat:',
    lat_formatter=formatter,
    lng_formatter=formatter,
)

site_map.add_child(mouse_position)
site_map
```



Basic Decision Tree Constructor



As you can see 'gini' near '0' - the probability of the success will be highest