IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Mithun Nagesh Shet
18/04/2022

Outline

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

Executive Summary

- **Summary of methodologies**
  - -Data Collection through API and Web Scraping
  - -Data Wrangling
  - -Exploratory Data Analysis with SQL
  - -Exploratory Data Analysis with Data Visualization
  - -Interactive Visual Analytics with Folium
  - -Building Interactive Dashboard with Plotly
  - -Predictive Analysis
- **Summary of all results**
  - -Exploratory Data Analysis result
  - -Interactive analytics of Dashboard through screenshots
  - -Predictive Analysis result

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. This information can be used if an alternate company wants to bid against space X for a rocket launch. The goal of this project is to study the launch data and find a predictive model to predict if the first stage of Falcon 9 rocket will land successfully or not.

Problems you want to find answers

-What are the different attributes that contribute for the successful landing of Boosters?

-The relationship of the attributes with booster landing.

-What attributes help SpaceX to achieve best landing rates of their boosters.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - SpaceX Rest API
  - Webscrapping from Wikipedia
- Performed data wrangling
  - The data was collected through get_request from Spacex API. The response content was a Json file and was retrieved using .json() function call and was converted to a pandas dataframe using .json_normalize().
  - The categorical data was One Hot Encoded and irrelevant columns were dropped.
- Performed exploratory data analysis (EDA) using visualization and SQL
  - Plots such as scatter plots,bar plots,regplot were used to find the relationship between variables.
- Performed interactive visual analytics using Folium and Plotly Dash
- Performed predictive analysis using classification models to build, tune, evaluate classification models

Data Collection – SpaceX API

- Get request to the SpaceX API to collect data, used custom functions to extract data, cleaned the requested data and assigned to new variable names. The null values in Payload_mass was replaced by mean value.
- Finally the Dataframe was exported as csv file.
- The Github url of the file:

[Spacex_project/Data Collection API Lab.ipynb at master · Mith1201/Spacex_project (github.com)]

1 .Getting Response from API

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

2 . Use json_normalize method to convert the json result into a dataframe

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

3 . Use custom functions to extract data

```
# Call getBoosterVersion
getBoosterVersion(data)
```

```
# Call getLaunchSite
getLaunchSite(data)
```

```
# Call getPayloadData
getPayloadData(data)
```

```
# Call getCoreData
getCoreData(data)
```

4 . Construct the dataset using the data obtained and combine the columns into a dictionary.

```
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}
```

5 . Extract data for Falcon 9 and export to csv

```
data_falcon9=data[data['BoosterVersion']=='Falcon 9']
data_falcon9.to_csv('dataset_part\_1.csv', index=False)
```

Data Collection - Scraping

- The data was scraped from the wikipage using BeautifulSoup .
- The columns were renamed and relevant columns from table for Falcon 9 was retrieved.
- The new Dataframe was later exported as csv.
- The Github url of the file:

Spacex_project/Data Collection with webscrapping.ipynb at master · Mith1201/Spacex_project (github.com)

1 . Request the Falcon9 Launch Wiki page from its URL and create Beautifulsoup object

```
# use requests.get() method with the provided static_url
Data=requests.get(static_url).text
# assign the response to a object
soup=BeautifulSoup(Data,"html5lib")
```

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

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

```
res = soup.find_all('th')
for x in range(len(res)):
    try:
     name = extract_column_from_header(res[x])
     if (name is not None and len(name) > 0):
        column_names.append(name)
    except:
         pass
```

3 . Create a data frame by parsing the launch HTML tables and export to csv

```
launch_dict= dict.fromkeys(column_names)

# Remove an irrelvant column
#del launch_dict['Date and time (UTC)']

# 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']=[]
```

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

Data Wrangling

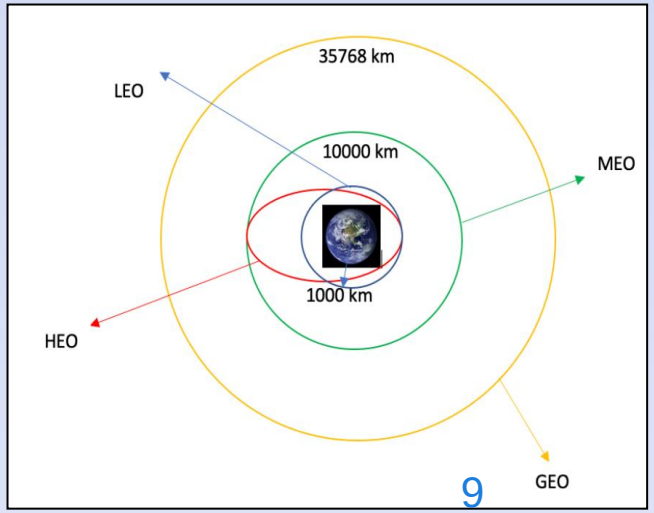
Introduction

- 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.
- Here we will mainly convert those outcomes into Training Labels with 1 means the booster successfully landed 0 means it was unsuccessful.

Process

Perform EDA on dataset → Calculate the number of launches at each site → Calculate the number and occurrence of each orbit → Create a landing outcome label from Outcome column → Create a Class variable with successful landing is assigned to 1 and unsuccessful to 0 → Export dataset as .CSV

Common Orbit types used by Spacex:



[Github Url for notebook]

EDA with Data Visualization

- Scatter Graphs used for:
  - 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
  - Scatter plots are used to determine the correlation between different variables.

- Bar Graph used for:
  - Mean Success Rate VS Orbit
  - Bar graph gives a good comparison between different categories. Highest success rate was found in ESL1,GEO,HEO and SSO.
- Line Graph used for:
  - Success Rate VS Year
  - Line graph gives trends of variables over the period of time and also good vizualization for predictions
  - Its clearly seen the success rate has improved over the years from time of its inception.

Github Url for Notebook

EDA with SQL

- Created service credentials to access database in jupyter notbook.
- Created schema in DB2 and spaceX dataset was uploaded which was used to extract information using sql.
- Task 1 included displaying the names of the unique launch sites in the space mission.
- Task 2 included displaying 5 records where launch sites begin with the string 'CCA‘.
- Task 3 included displaying the total payload mass carried by boosters launched by NASA (CRS).
- Task 4 included displaying average payload mass carried by booster version F9 v1.1.
- Task 5 included listing the date when the first successful landing outcome in ground pad was achieved.
- Task 6 included listing the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000.
- Task 7 included listing the total number of successful and failure mission outcomes.
- Task 8 included listing the names of the booster versions which have carried the maximum payload mass.
- Task 9 included listing the failed landing outcomes in drone ship, their booster versions, and launch site names for in year 2015
- Task 10 included 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 Link for notebook

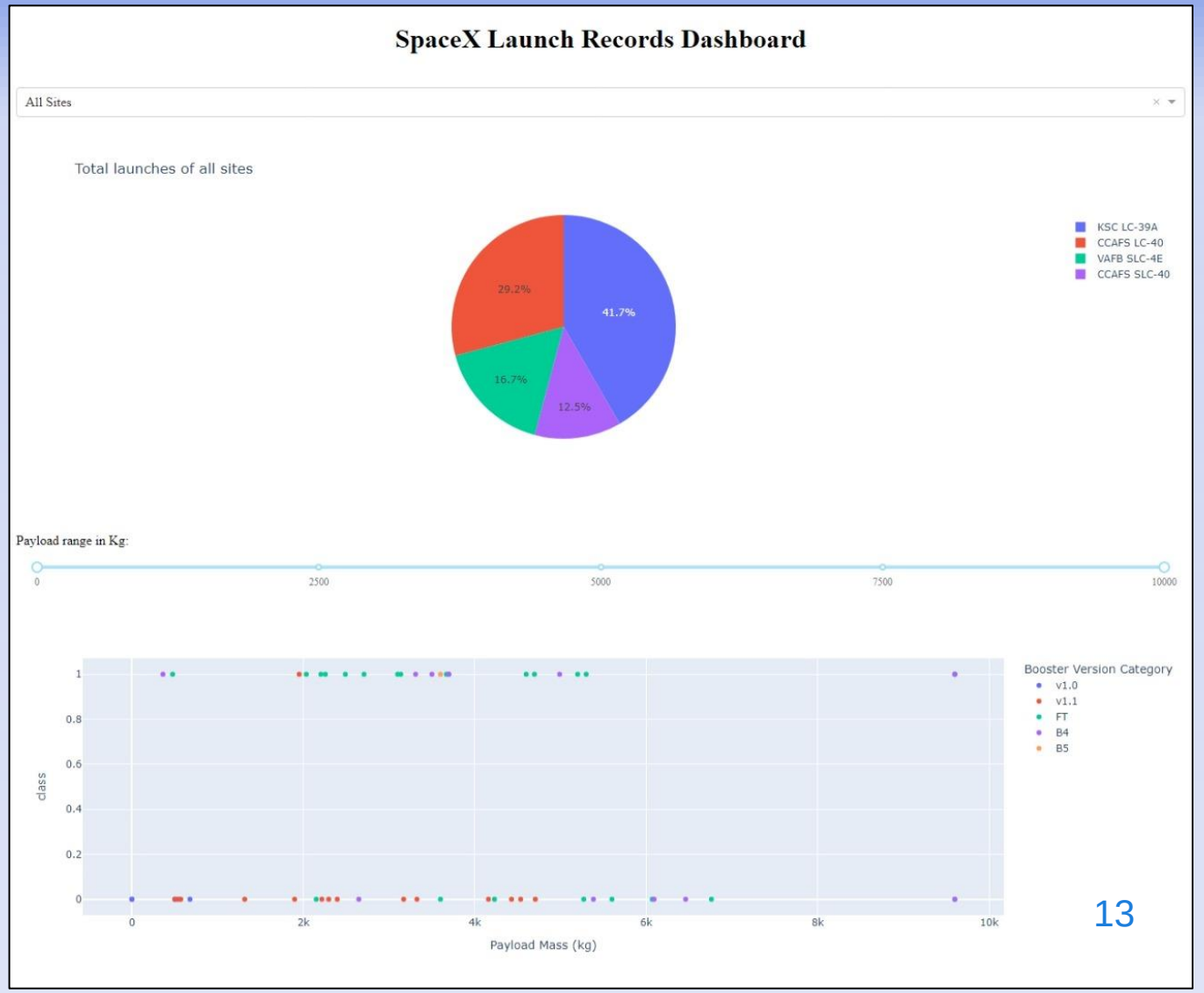
Build an Interactive Map with Folium

- Marked all launch sites, and added map objects such as markers, circles, lines to mark the success or failure of launches for each site on the folium interactive map. The information of Latitudes and Longitudes for respective launch sites were extracted from the table for marking purpose.
- We assigned the launch_outcomes with 0(failure) and 1 (Success) to Green and Red markers on the map in cluster.(MarkerCluster()).
- We calculated the distance from the Launch site to landmarks at its close proximity such as Railways,Airports,coast etc. Lines were drawn on the map for better visualization of the proximity from launch sites.
- This helped us answer some questions like:
  - Are launch sites close to railway?
  - Are launch sites close to coast line?
  - Are launch site close to highways?
  - Are launch sites close to city center?

Github Url for notebook

Build a Dashboard with Plotly Dash

- Interactive Dashboard was built using Plotly Dash.
- The pie chart showing the total launches for All Sites or specific launch sites.
- The pie chart also provides information about successful(Class 1) and unsuccessful(Class 0) launches for specific launch sites.
- The scatter graph showing the relationship with Outcome and Payload Mass(Kg) for the different booster versions.



[Github Url link for notebook]

Predictive Analysis (Classification)

Building Model

- Load our dataset into NumPy and Pandas
- Split our data into training and test data sets
- Normalize the data
- Set our parameters and algorithms to GridSearchCV for our ML Models
- Fit our datasets into the GridSearchCV objects and train our dataset.

Evaluating the Model

- Check the accuracy for each model
- Get optimized hyperparameters for each type of algorithms
- Plot confusion Matrix to check the performance of the model

Find the best performing classification model

- The model with best accuracy score is the best performing model
- For our study, Tree model was found best classification model

Github Url for the notebook

Results

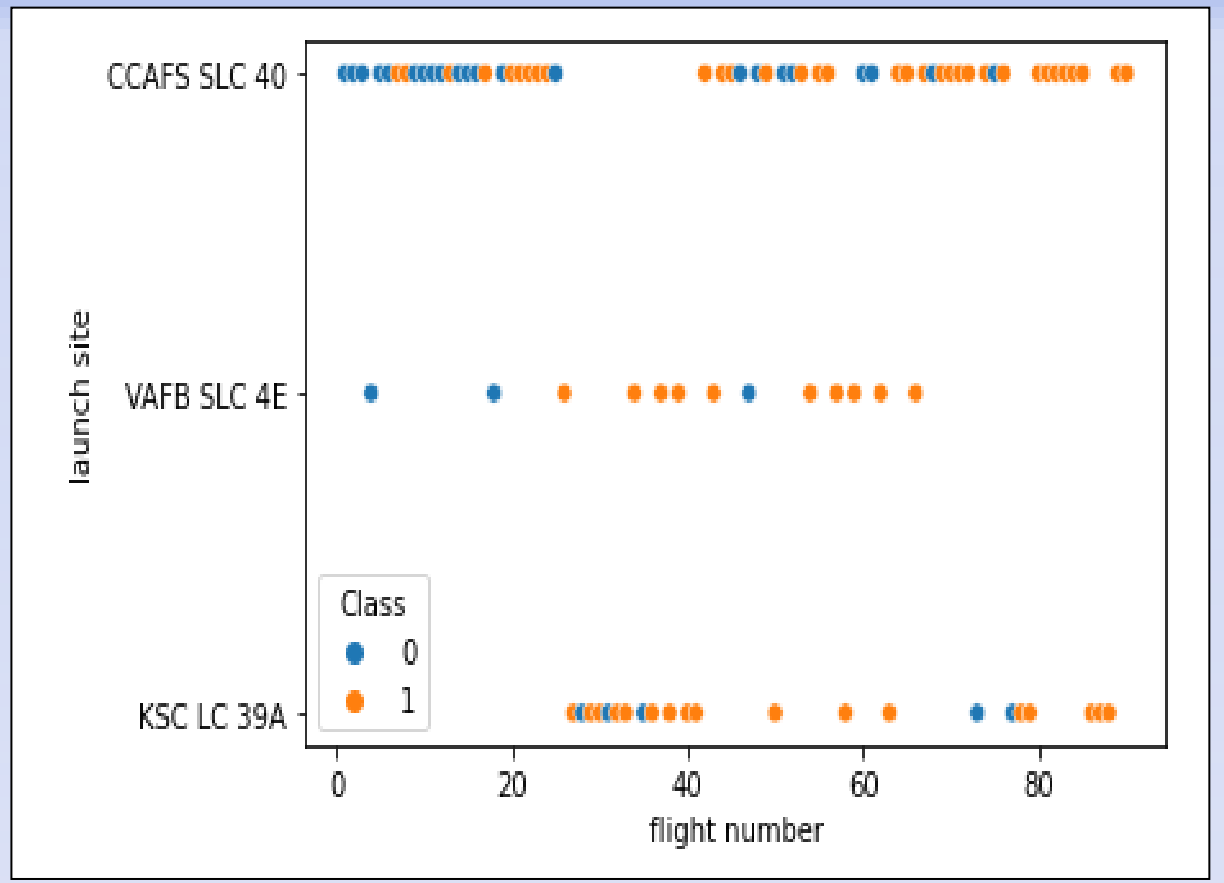
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

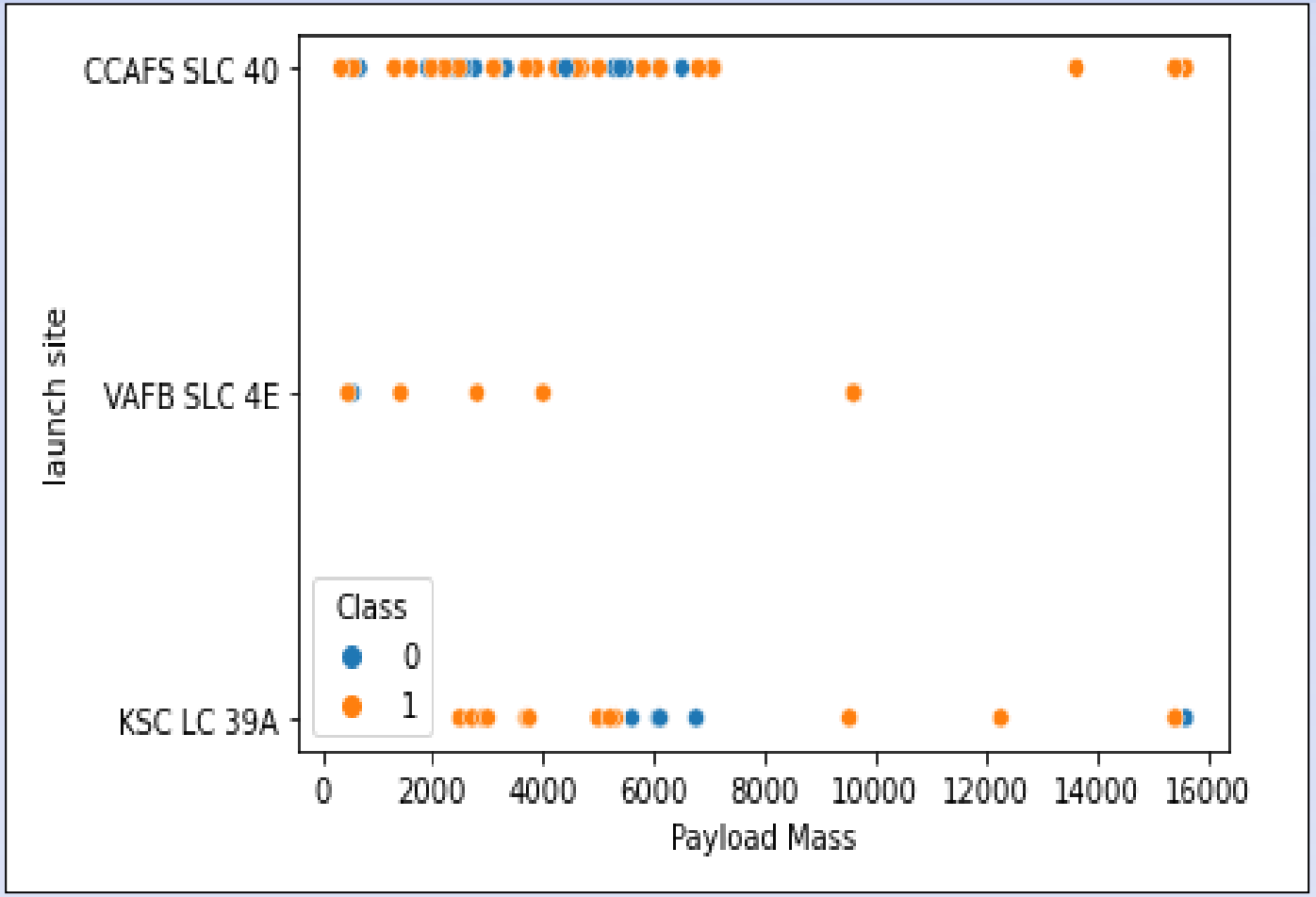
Flight Number vs. Launch Site

From the plot, it can be found that the larger the amount of flights at a launch site, the greater the success rate at a launch site.



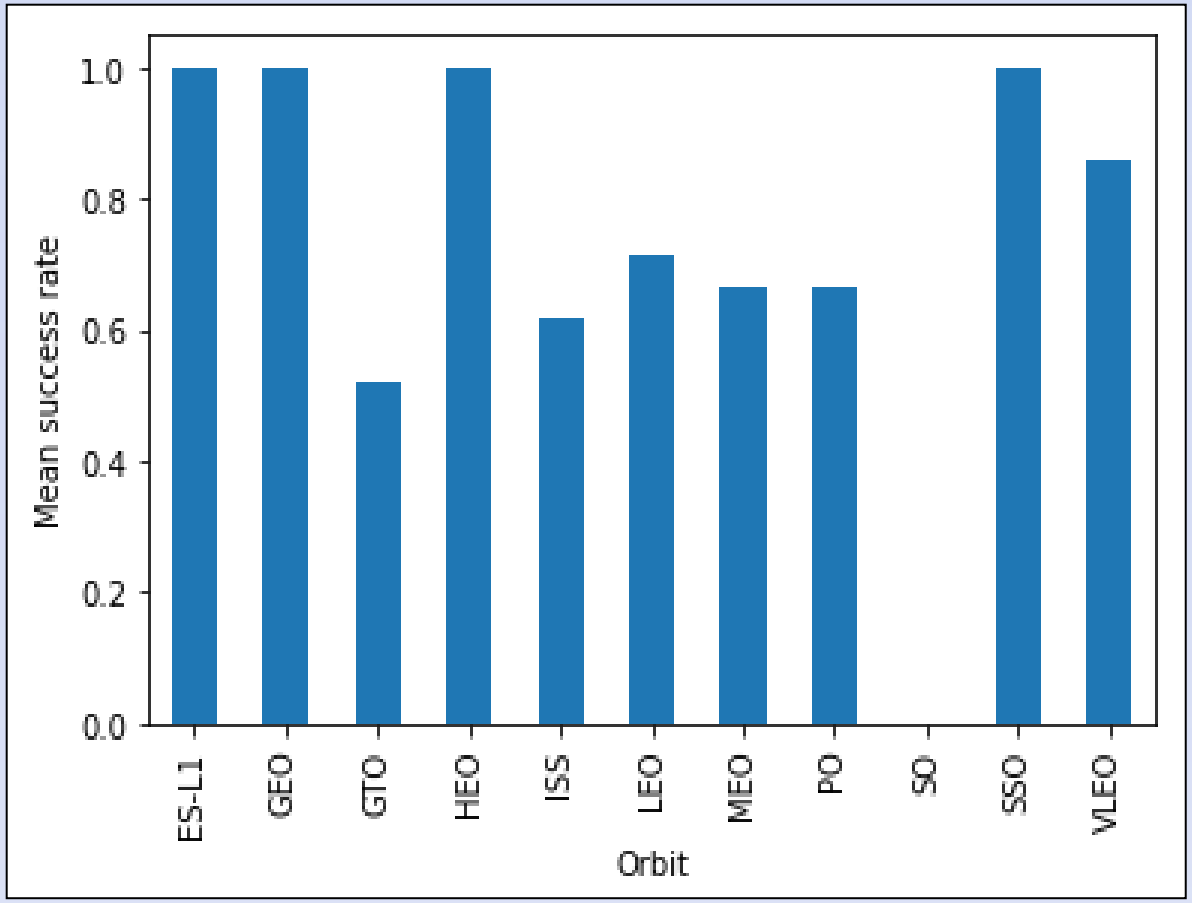
Payload vs. Launch Site

- We can observe that VAFB-SLC launch site there are no rockets launched for heavy payload mass(greater than 10000)
- It can also be seen that success rate is higher for higher payload mass.
- There is no clear pattern between the launch site and payload mass and hence hard to interpret its effect on success rate.



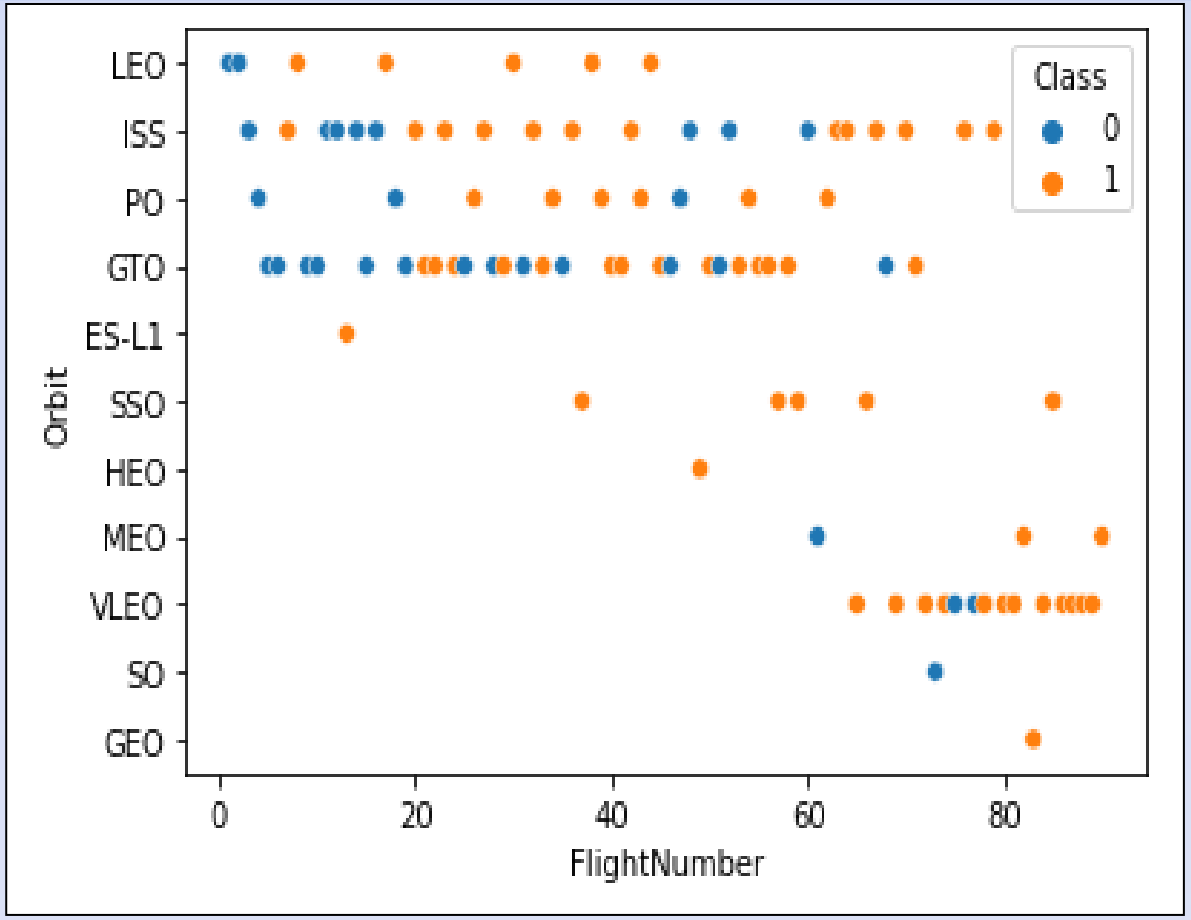
Success Rate vs. Orbit Type

- From the bar graph , we can observe Orbits ES-L1,GEO,HEO,SSO have best success rate.
- The orbit SO has almost zero success rate.



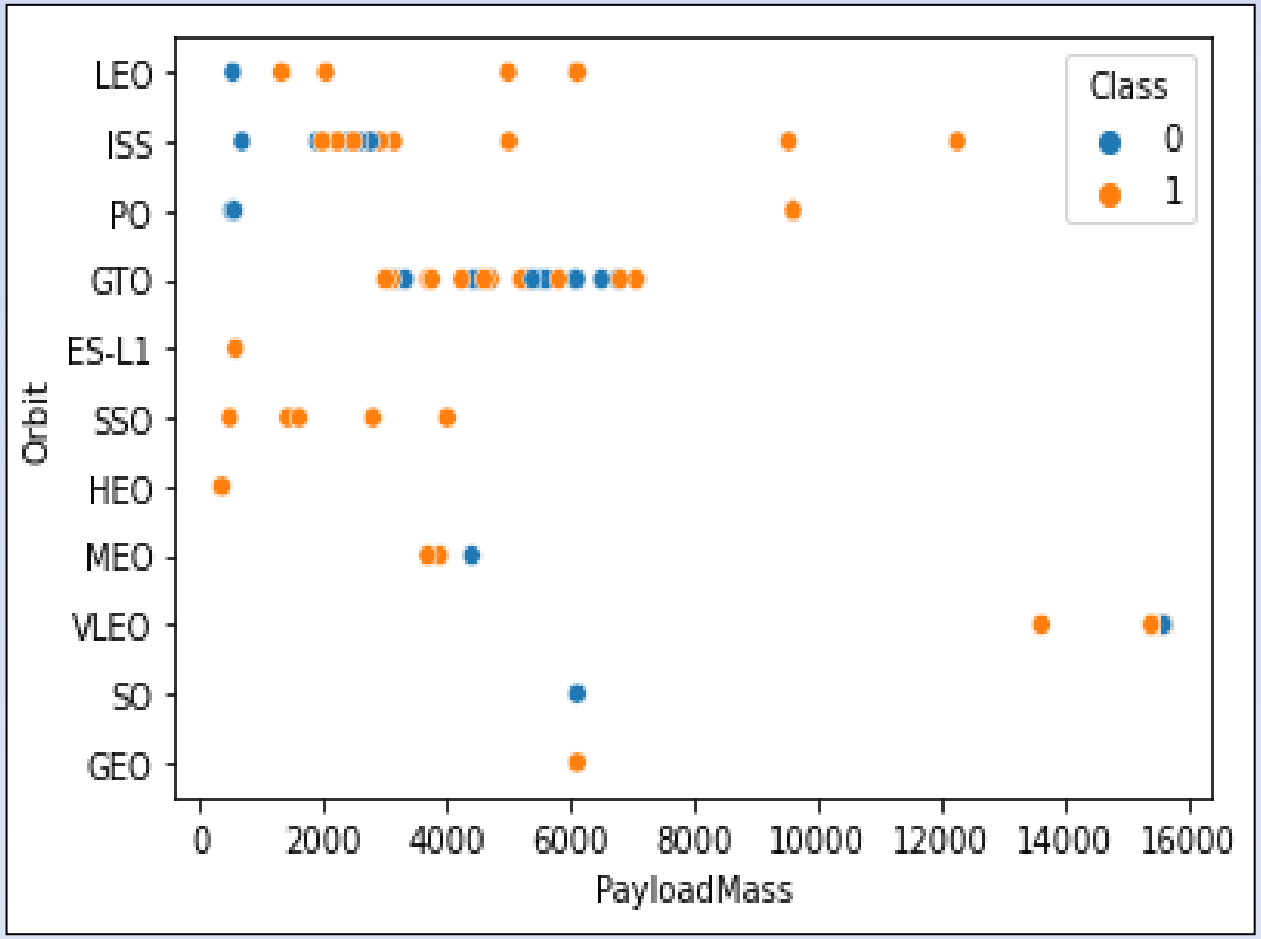
Flight Number vs. Orbit Type

- We observe 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.



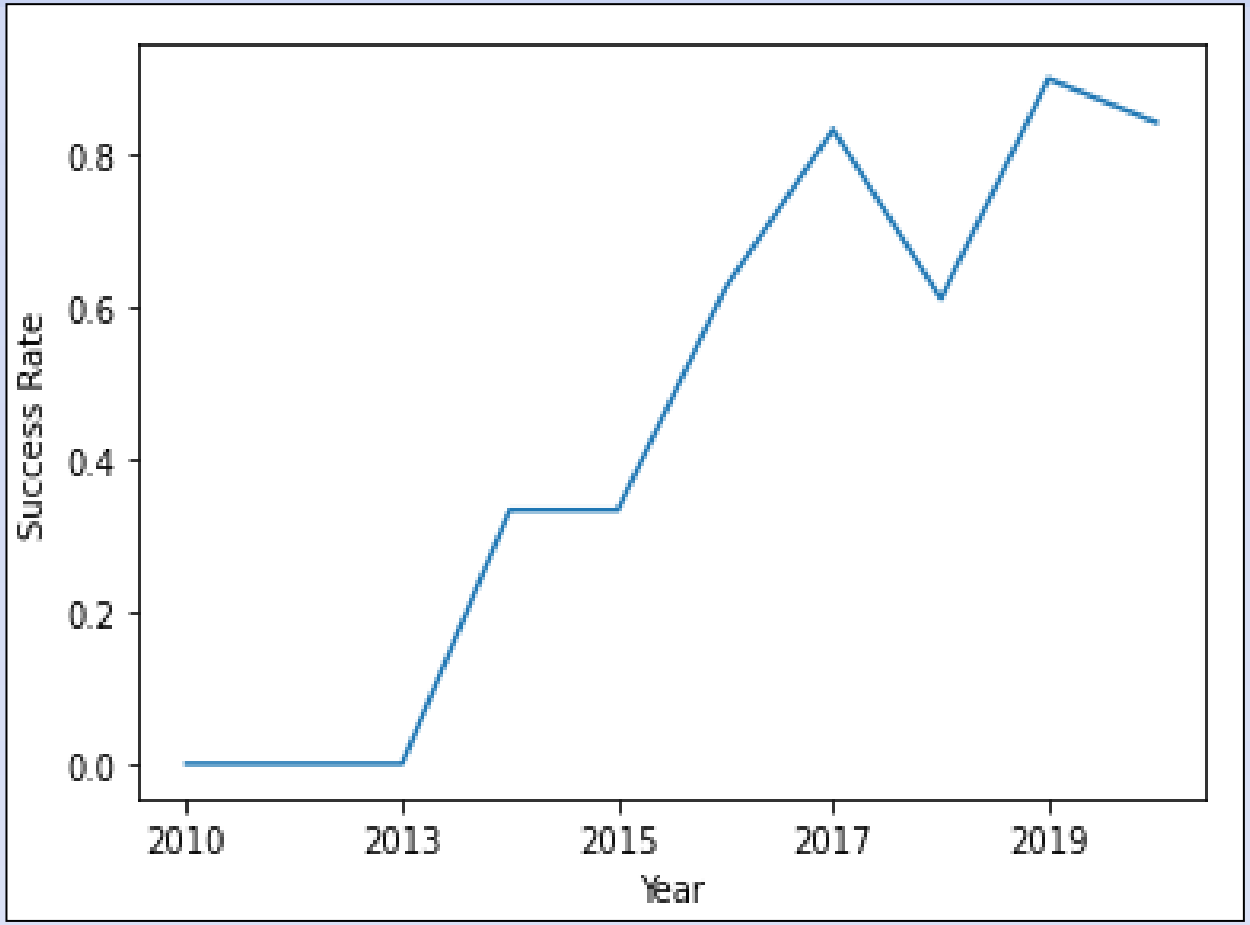
Payload vs. Orbit Type

- We observe that 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

- We observe that success rate has shown a increasing trend from 2010 with highest observed in the year 2019.



All Launch Site Names

SQL Query

%sql Select
DISTINCT(launch_site)

From spacextbl

Explanation:

We use Distinct in the query to show unique launch sites from table spacextbl

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

Launch Site Names Begin with 'CCA'

SQL Query

```
%%sql select * from spacextbl where launch_site like 'CCA%' limit 5;
```

Explanation:

We use LIKE keyword as wildcard with words ‘CCA%’ to filter out launch_site starting with CCA. LIMIT 5 is used to limit the result to 5 rows.

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 Query

```
%%sql select sum(payload_mass__kg_)
as total_payload from spacextbl where
customer='NASA (CRS)';
```

total_payload
45596

Explanation:

SUM summates the total in the column payload_mass_kg. The WHERE clause filters the results to customer NASA(CRS).

Average Payload Mass by F9 v1.1

SQL Query

```
%%sql select avg(payload_mass__kg_) as average_payload_mass from spacextbl
where booster_version like 'F9 v1.1';
```

average_payload_mass
2928

Explanation:

Function AVG is used to calculate the average in the column payload_mass__kg. The WHERE clause filters results to booster version ‘F9 v1.1’

First Successful Ground Landing Date

SQL Query

```
%%sql select min(Date) as
first_successful_landing from spacextbl

where landing__outcome like '%ground
pad%';
```

first_successful_landing
2015-12-22

Explanation:

MIN function is used to find the date of first successful ground landing. LIKE is used as wildcard to filter landing_outcome with groundpad with successful landing.

Successful Drone Ship Landing with Payload between 4000 and 6000

SQL Query

```
%%sql select
booster_version,payload_mass__kg_ from
spacextbl

where landing__outcome = 'Success (drone
ship)' and payload_mass__kg_ between 4000
and 6000;
```

booster_version	payload_mass__kg_
F9 FT B1022	4696
F9 FT B1026	4600
F9 FT B1021.2	5300
F9 FT B1031.2	5200

Explanation:

Queried booster_version and payload_mass_kg from table spacextbl. WHERE clause is used to filter the data for successful landing for drone ship with payload mass between 4000 and 6000.

Total Number of Successful and Failure Mission Outcomes

SQL Query

```
%sql Select MISSION_OUTCOME,count(MISSION_OUTCOME) as count from SPACEXTBL GROUP BY MISSION_OUTCOME;
```

mission_outcome	COUNT
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

Explanation:

Queried Mission_outcome and its COUNT from spacextbl

with results grouped by Mission_outcome.

Boosters Carried Maximum Payload

SQL Query

```
%%sql select
booster_version,payload_mass__kg_ from
spacextbl

where payload_mass__kg_ in (select
max(payload_mass__kg_) from spacextbl);
```

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

Explanation:

Queried Booster_version and payload mass from spacextbl with payload mass was subqueried from spacextbl using MAX function to get maximum payload for respective booster version.

2015 Launch Records for Failed Landing of Droneship

SQL Query

```
%%sql select monthname(Date) as
Month,year(Date) as
Year,landing__outcome,booster_version,launch_site
from spacextbl

where landing__outcome in 'Failure (drone ship)'
and year(DATE)=2015;
```

MONTH	YEAR	landing__outcome	booster_version	launch_site
January	2015	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
April	2015	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

Explanation:

Queried month,Year,landing_outcome,Booster version and Launch site from table spacextbl. WHERE clause used to filter the results with landing outcome as ‘Failure(drone ship)’ fot the year 2015.

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

SQL Query

```
%%sql select
landing__outcome,count(landing__outcome) as
total_outcome from spacextbl
Where Date>'2010-06-04' and Date<'2017-03-20'
group by landing__outcome order by
count(landing__outcome) desc;
```

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

Explanation:

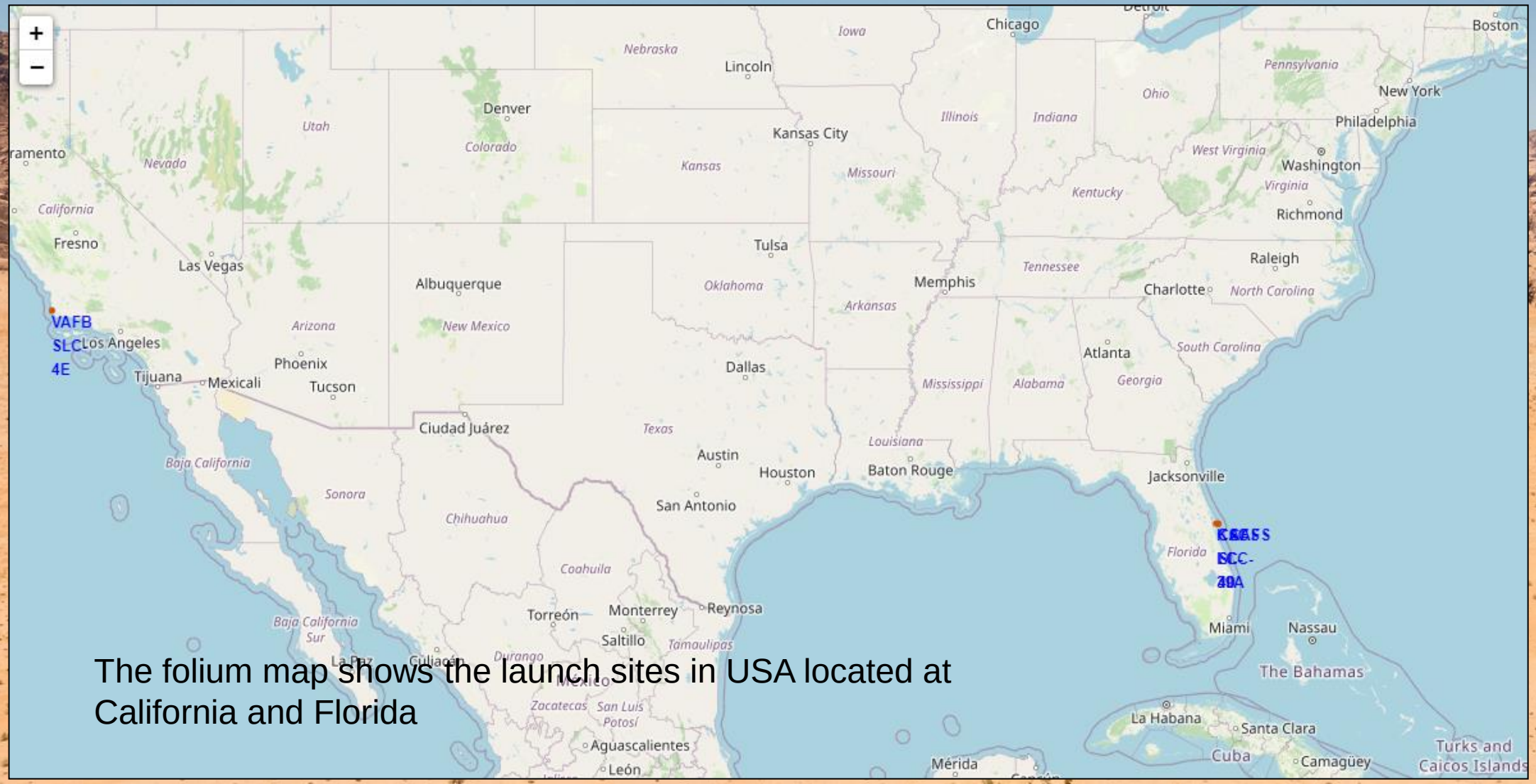
Queried landing outcome and its COUNT from spacextbl.

WHERE clause used to filter the landing outcome between 2010-06-04 and 2017-03-20. The results were ranked based on highest total outcome using Order By DESC clause.

Section 3

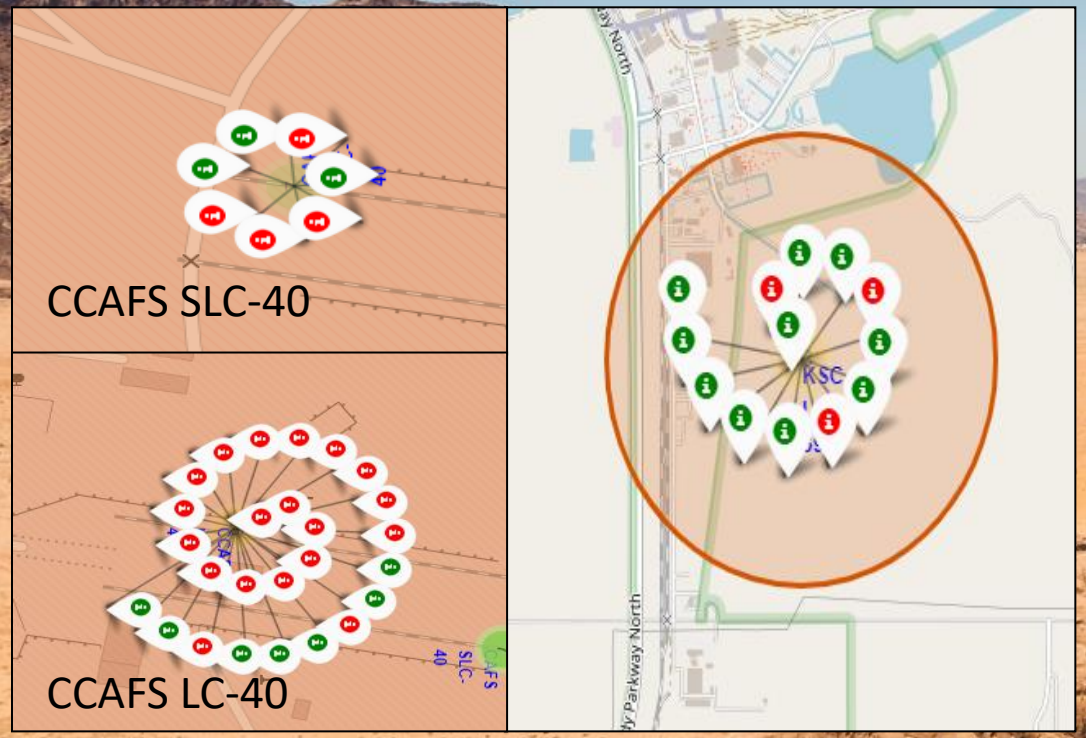
Launch Sites Proximities Analysis

All Launch sites in USA

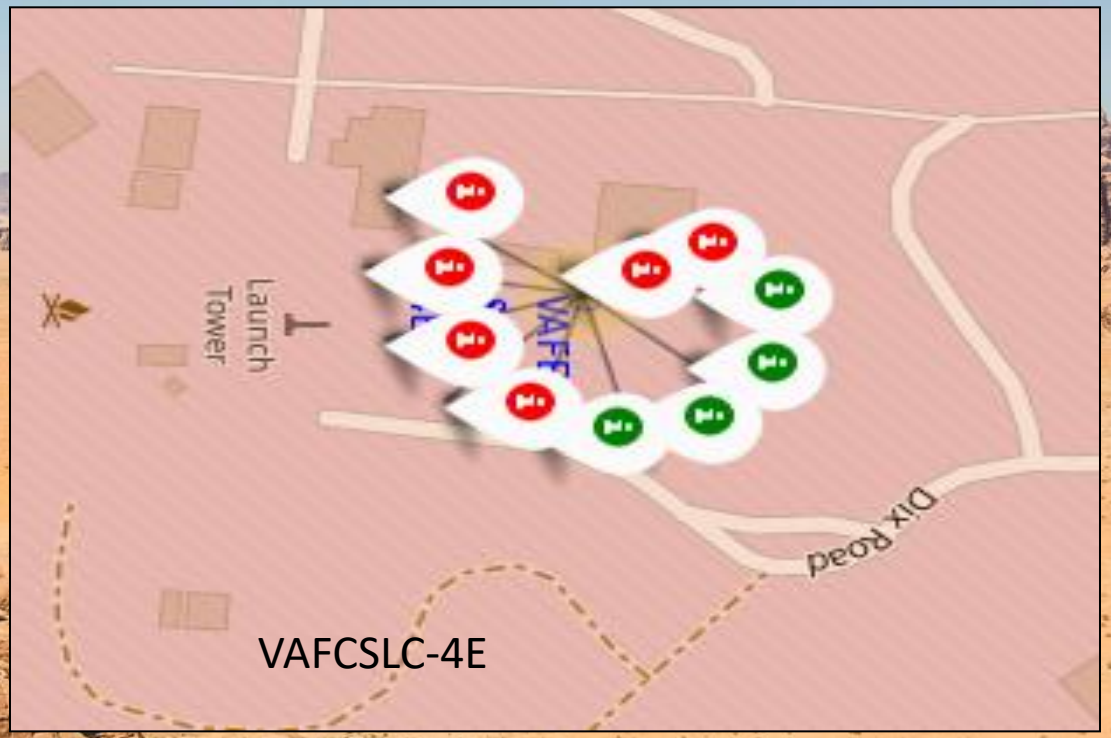


The folium map shows the launch sites in USA located at California and Florida

Labeled Markers for Launch sites

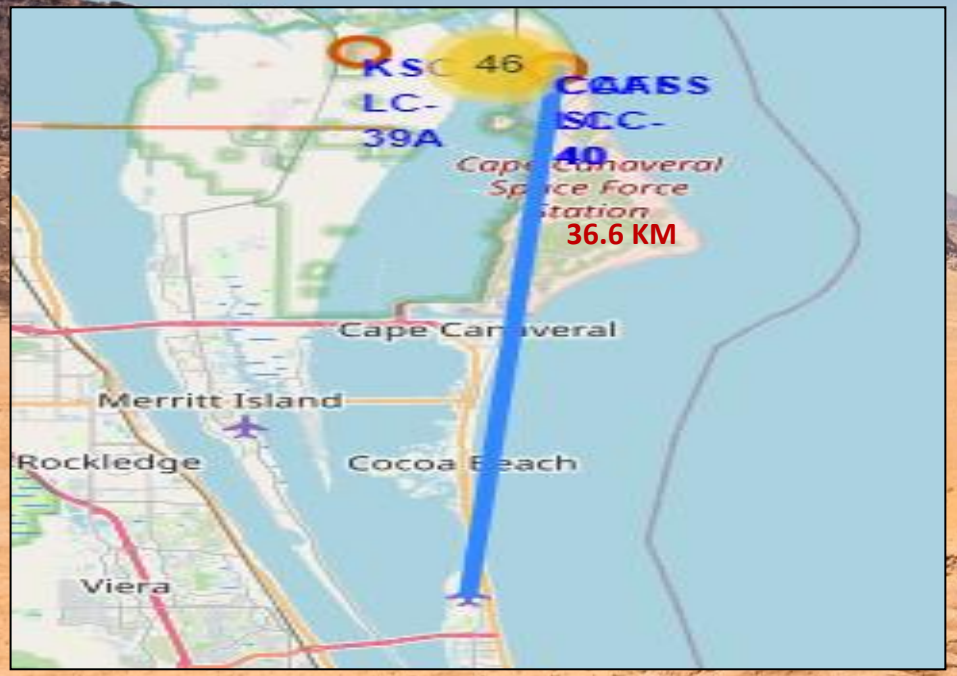


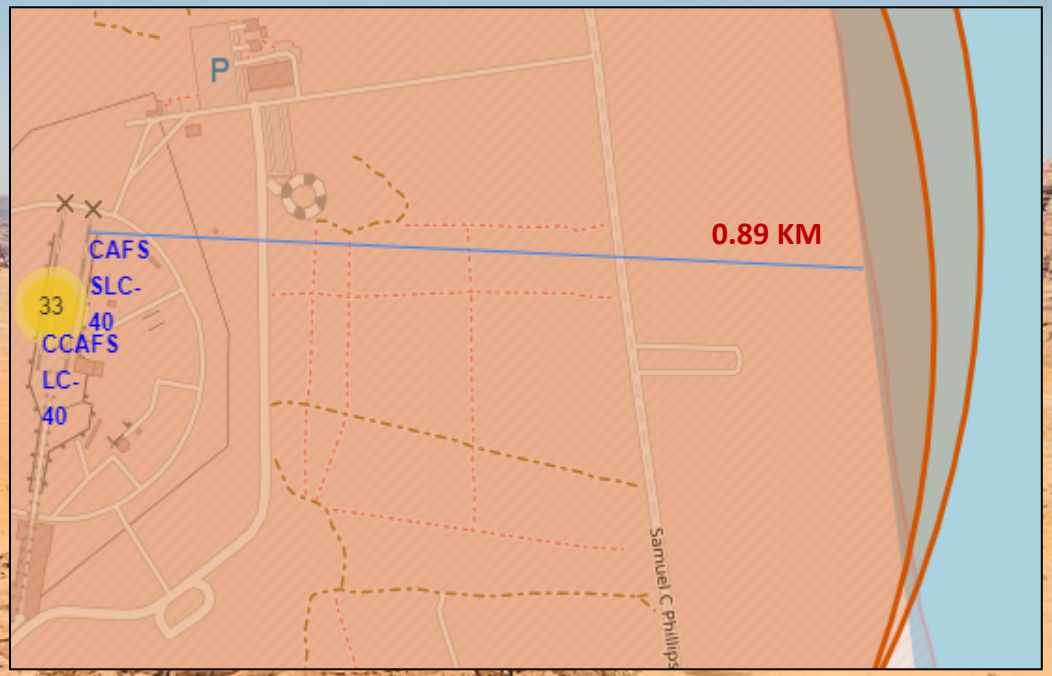
Florida Launch site with green marker showing successful launches and Red marker showing Failure



California Launch site with green marker showing successful launches and Red marker showing Failure

Proximity of Launch site to nearest Landmarks



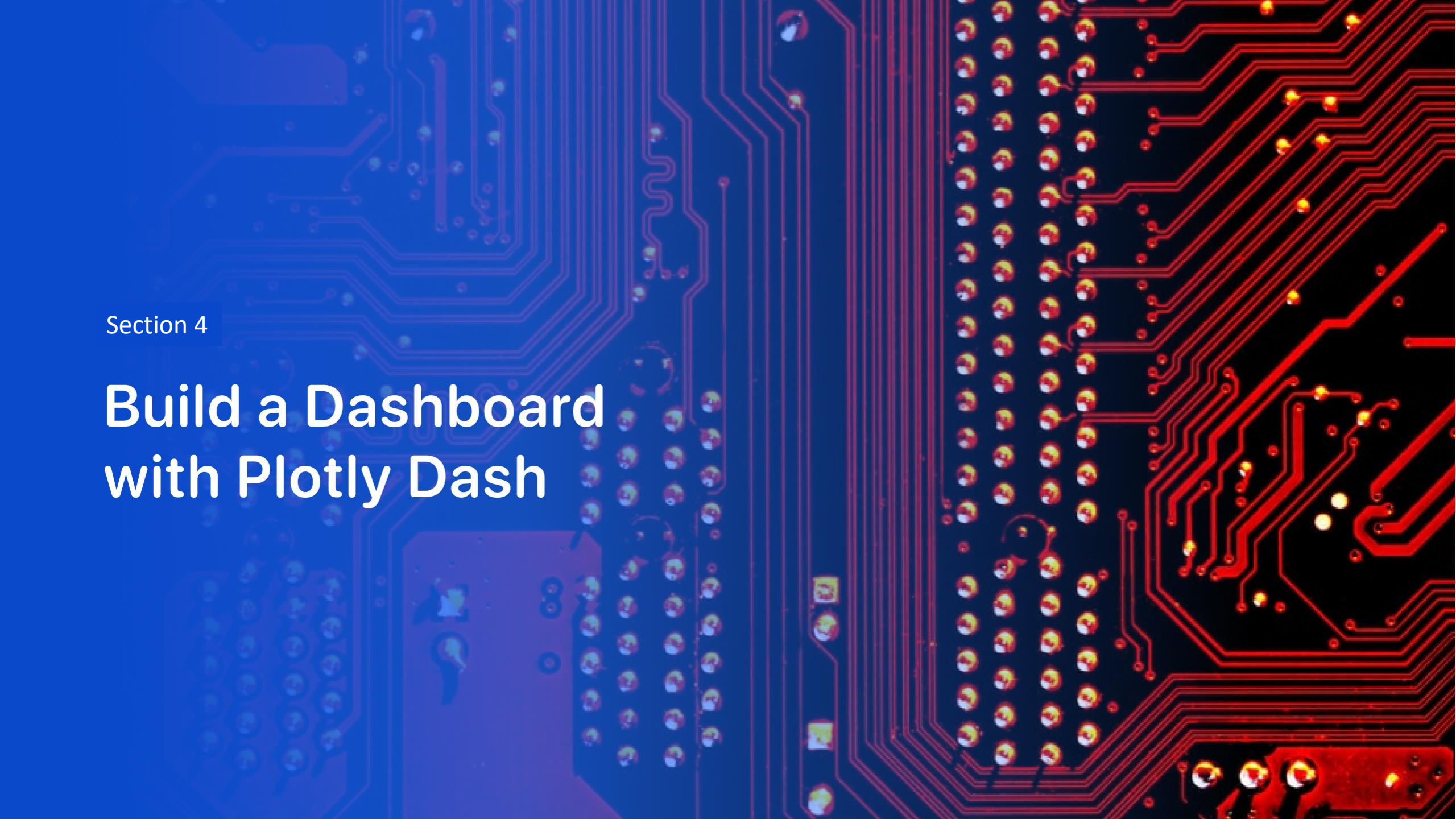


Distance to nearest Patrik Space force base

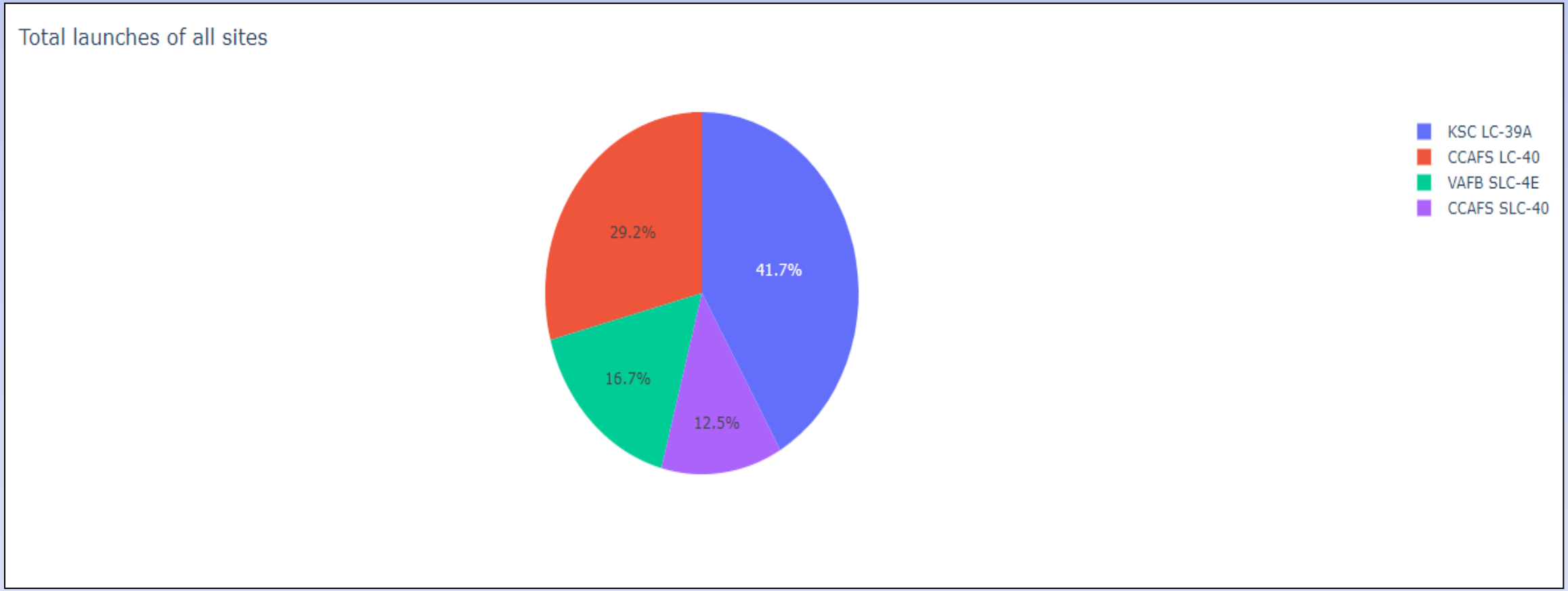
The distance is about 36.6 km and is not in close proximity to nearest flight zone

Distance to nearest Coast Line

The distance is about 0.89 km from CAFSSLC-40 and is in close proximity to coast.

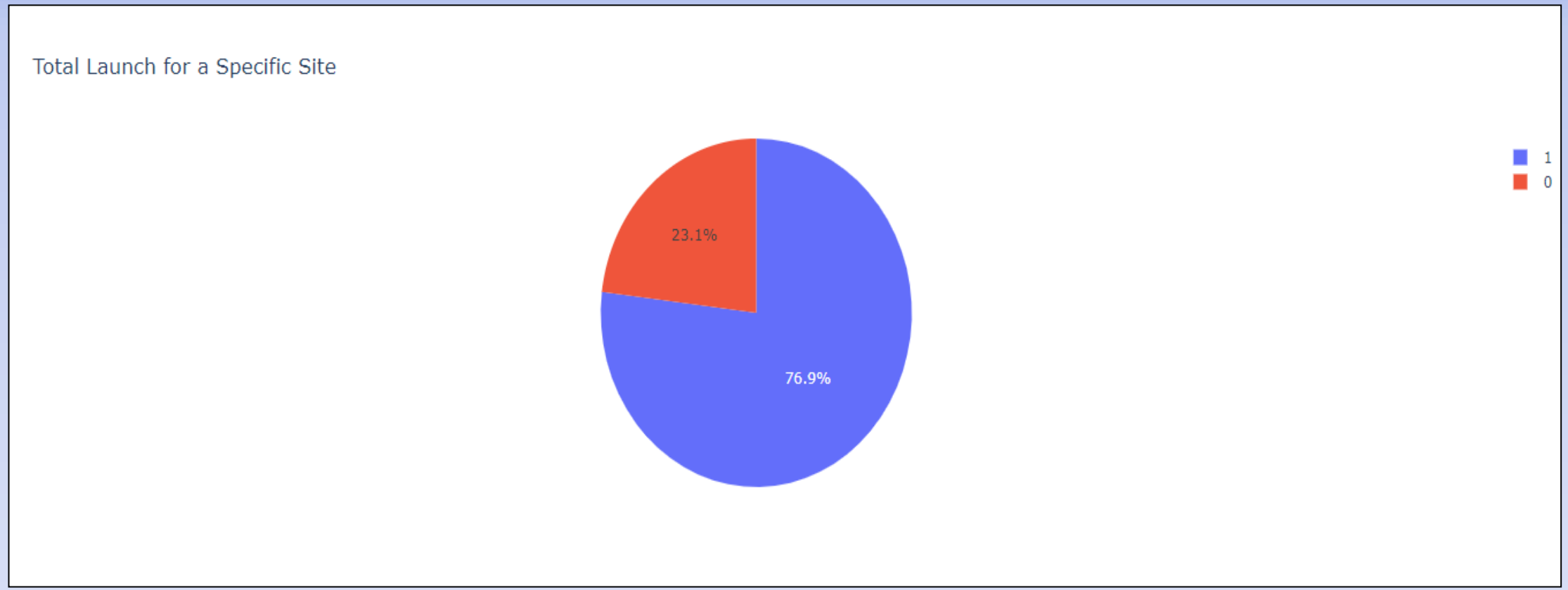

Section 4
Build a Dashboard with Plotly Dash

Dashboard Pie chart showing success percentage of all launch sites



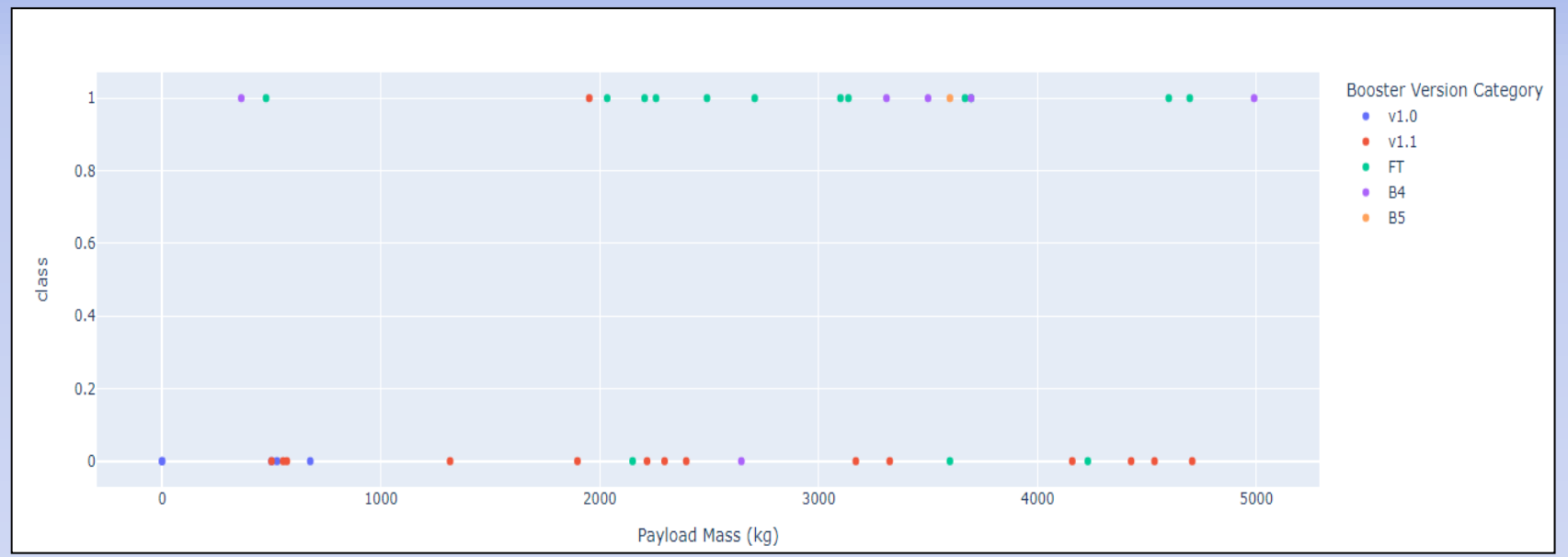
KSC LC-39A had the most successful launch perecntage of all launch sites

Dashboard Piechart for launch site with highest success rate



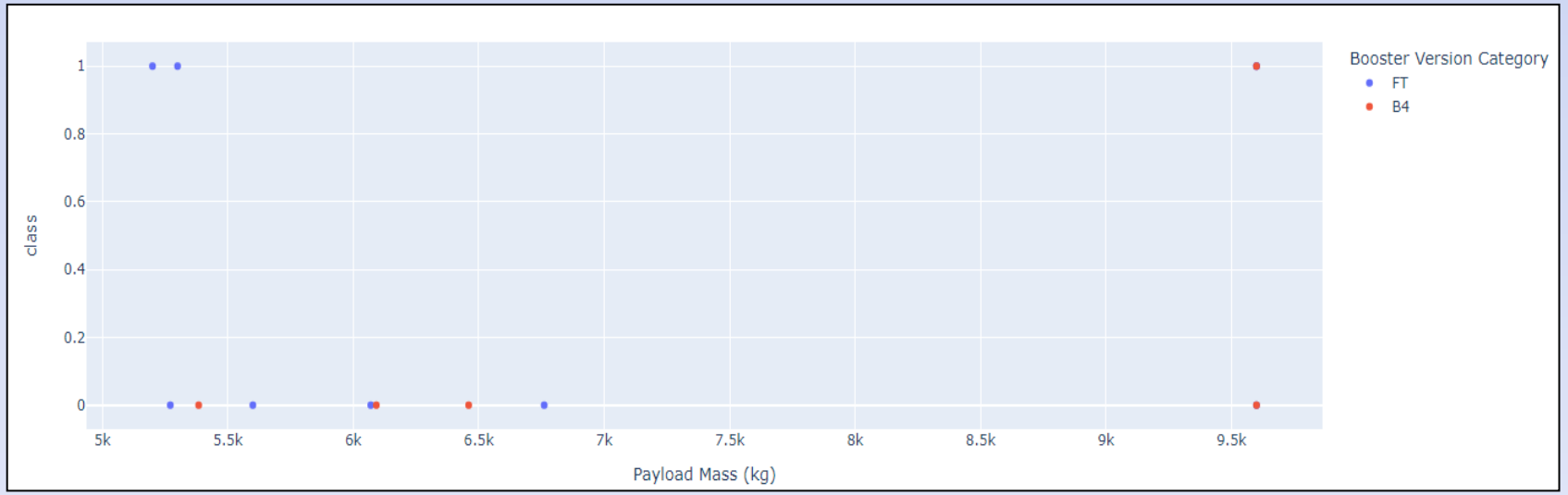
KSC LC-39A had a success rate of 76.9% and failure of 23.1%.

Dashboard- Scatter plot for payload VS Success rate(Class) for all launch sites



Payload Mass between 0 to 5000 kg

Success rates observed for lower payload was higher. Booster version FT showed good success rate for lower payload mass



Payload Mass between 5000 to 10000 kg

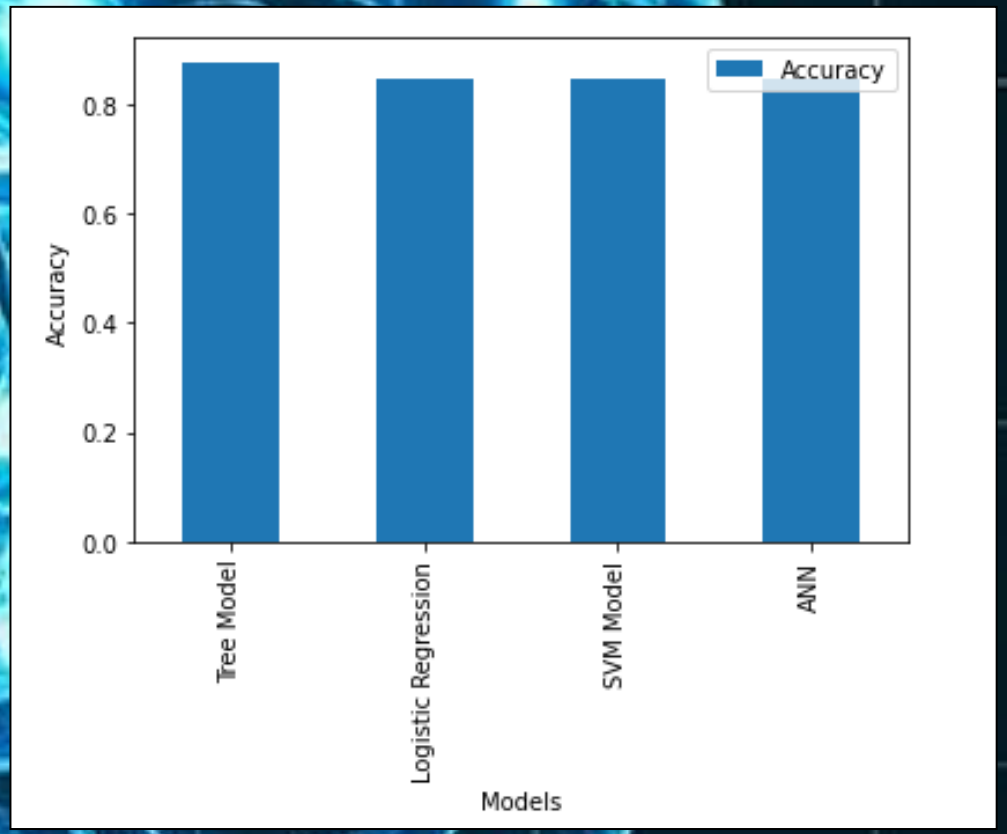
Success rates observed for higher payload was lower. Booster version FT showed good success rate for higher payload mass within 5500 kg.

Section 5

Predictive Analysis (Classification)

Classification Accuracy

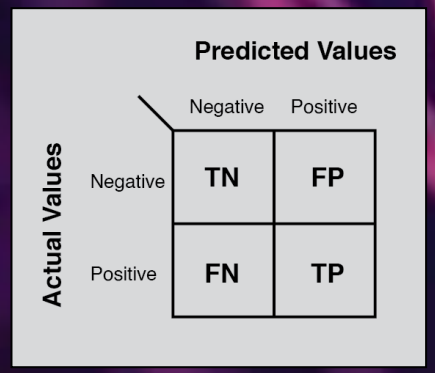
Model	Accuracy
Tree Model	0.876786
Logistic Regression	0.846429
SVM Model	0.848214
ANN	0.848214

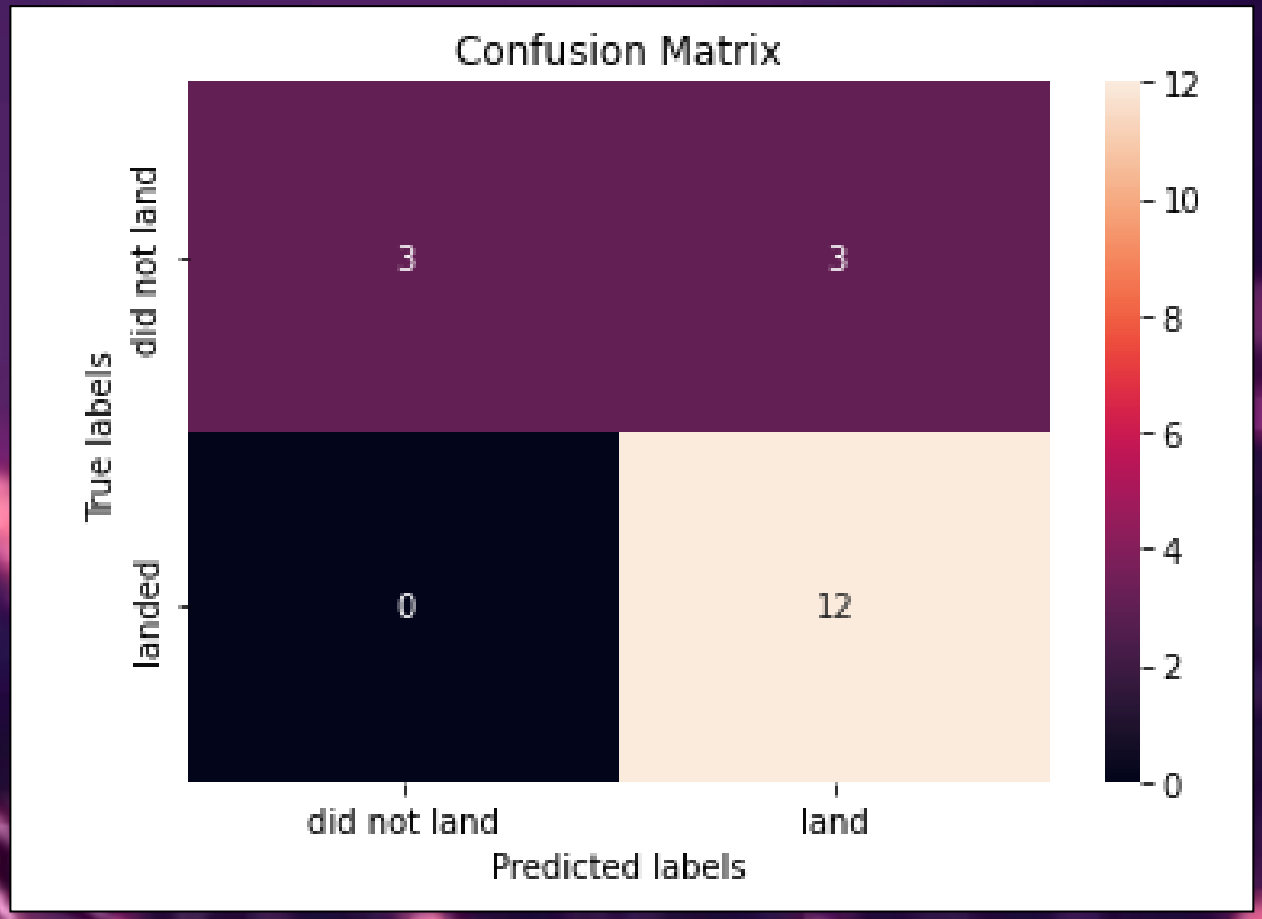


Tree Model showed highest Accuracy among the classification models with an accuracy of 87.67%.
The optimised parameters obtained from Grid search for tree classification model is:
tuned hpyerparameters :(best parameters)
{'criterion': 'entropy', 'max_depth': 12, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 10, 'splitter': 'random'}
The model showed an accuracy of 83.33% on Test data

Confusion Matrix

The confusion matrix for the decision tree classifier shows that the classifier can distinguish between the different classes. The major problem is the false positives .i.e., unsuccessful landing marked as successful landing by the classifier.





Conclusions

- Higher success rates was found for lower payload mass(within 5.5 tons) and booster version FT performed well.
- The success rate for Spacex launches have improved significantly from 2010 to present.
- The launch site KSC LC-39A had highest number of successful launches.
- The orbits used by Spacex like GEO,HEO,SSO,ES-L1 had highest success rate and orbit SO had amost zero success rate.
- Tree Classification model showed better accuracy among classification models and will be good model to predict future launches for higher success rate.

Appendix

- **IBM Watson studio**
- **SQL DB2 relational database**
- **Jupyter notebook with python libraries.**
- **Plotly Dash**
- **Folium interactive map**
- **Scikit learn Library**
- **Haversine formula to measure distance on maps**

Thank you!