IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

<Jessica Avila>
<Date>

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection
  - Data Wrangling
  - EDA with data visualization
  - EDA with SQL
  - Visual Analysis with Folium Maps
  - Visual Analysis with Dashboards using Plotly Dash
  - Predictive Analysis using Machine Learning's Classification

- Summary of all results
  - Results using EDA
  - Results from Predictive Analysis

Introduction

The commercial space age is here, companies are making space travel affordable for everyone. 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.

One reason SpaceX can accomplish more tasks is that the rocket launches are relatively inexpensive.

Here we'll use classification models to predict if the first stage will succeed and use exploratory data analysis to better visualize the data and how it influences the outcome.
At the end of the analysis we'll be able to predict if the first stage will land successfully and the weight of the parameters in the success.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Request and clean SpaceX API
  - Web Scrapping from Wikipedia's page: List of Falcon 9 and Falcon Heavy launches
- Perform data wrangling
  - Process data using Pandas.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Build, tune and evaluate different machine learning classification models using Sklearn.

Data Collection

- Data was collected using two methodologies:

  - Request of SpaceX API

  - Extract a Falcon 9 launch records HTML from Wikipedia

Data Collection – SpaceX API

1. Start requesting rocket launch data from SpaceX API with the following URL:

In [6]:
```
spacex_url="https://api.spacexdata.com/v4/launches/past"
```

2. Request and parse the SpaceX launch data using the GET request:

In [7]:
```
response = requests.get(spacex_url)
```

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

In [15]:
```
# Use json_normalize meethod to convert the json result into a dataframe
data=pd.json_normalize(response.json())
```

4. Clean and Filter the data using functions like the following:

In [20]:
```
# Call getBoosterVersion
getBoosterVersion(data)
```

Data Collection – SpaceX API

5. Finally construct the dataset using the data combining the columns into a dictionary:

In [25]:
```
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}
```

6. Finish it filtering the Falcon 9 launches:

In [35]:
```
# Hint data['BoosterVersion']!='Falcon 1'
data_falcon9 = launch_data[launch_data['BoosterVersion']!='Falcon 1']
```

- Notebook: [SpaceX-Capstone/jupyter-labs-spacex-data-collection-api.ipynb at main · jparisavila/SpaceX-Capstone (github.com)]

Data Collection - Scraping

- Extracting data from a Wikipedia table using Beautiful Soup. The steps are shown here:

- Notebook: [SpaceX-Capstone/jupyter-labs-webscraping.ipynb at main · jparisavila/SpaceX-Capstone (github.com)]

1. Request the Falcon 9 Launch from Wikipedia

```
obj = requests.get(static_url).text[:]
```

2. Create a BeautifulSoup object from the HTML response

```
soup = BeautifulSoup(obj, 'html.parser')
```

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

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

4. Create a data frame by parsing the launch HTML tables

```
launch_dict= dict.fromkeys(column_names)

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

# 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']=[]
```

5. Create a Dataframe

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

Data Wrangling

- Data wrangling using Pandas to find and replace missing values, calculate the number and occurrence of mission outcome per orbit type and create a landing outcome label from Outcome column.
- Notebook: [SpaceX-Capstone/IBM-DS0321EN-SkillsNetwork_labs_module_1_L3_labs-jupyter-spacex-data_wrangling_jupyterlite.jupyterlite.ipynb at main · jparisavila/SpaceX-Capstone (github.com)]()

1. Calculate the number of launches on each site:

```
# Apply value_counts() on column LaunchSite
df["LaunchSite"].value_counts()
```

```
CCAFS SLC 40    55
KSC LC 39A      22
VAFB SLC 4E     13
Name: LaunchSite, dtype: int64
```

2. Use the method .value_counts() to count ocurrence on each orbit in the column 'Orbit':

```
# Apply value_counts on Orbit column
df["Orbit"].value_counts()
```

```
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
```

3. Calculate occurrence of mission per orbit type:

```
# landing_outcomes = values on Outcome column
landing_outcomes = df["Outcome"].value_counts()
landing_outcomes
```

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

4. Create a landing outcome label from Outcome column:

```
# landing_class = 0 if bad_outcome
# landing_class = 1 otherwise

landing_class=[]
for i in df["Outcome"]:
    if i in bad_outcomes:
        landing_class.append(0)
    else:
        landing_class.append(1)
landing_class[:5]
```

```
[0, 0, 0, 0, 0]
```

EDA with Data Visualization

- Using Matplotlib and Pandas we performed exploratory data analysis and prepared feature engineering, using the variables:

  - Flight Number
  - Payload Mass
  - Launch Site
  - Orbit

- The analysis ends with a line chart success rate versus year for visualizing evolution of success rate launches by year.
- Notebook: [SpaceX-Capstone/IBM-DS0321EN-SkillsNetwork_labs_module_2_jupyter-labs-eda-dataviz.ipynb.jupyterlite (1).ipynb at main · jparisavila/SpaceX-Capstone (github.com)]

EDA with SQL

- To perform exploratory data analysis on the data the following queries were built:

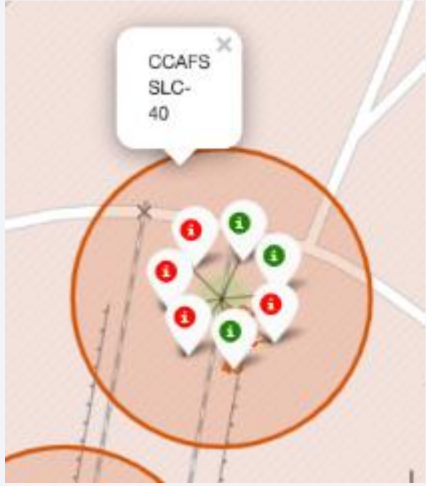
    1. Display the names of the unique launch sites in the space mission
    2. Display 5 records where launch sites begin with the string 'CCA'
    3. Display the total payload mass carried by boosters launched by NASA (CRS)
    4. Display average payload mass carried by booster version F9 v1.1
    5. List the date when the first successful landing outcome in ground pad was achieved.
    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. Use a subquery
    9. List the records which will display the month names, failure landing outcomes in drone ship ,booster versions, launch site for the months 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.

EDA with SQL

- Notebook: SpaceX-Capstone/jupyter-labs-eda-sql-coursera_sqllite-bak-2023-08-01-19-33-31Z.ipynb at main · jparisavila/SpaceX-Capstone (github.com)

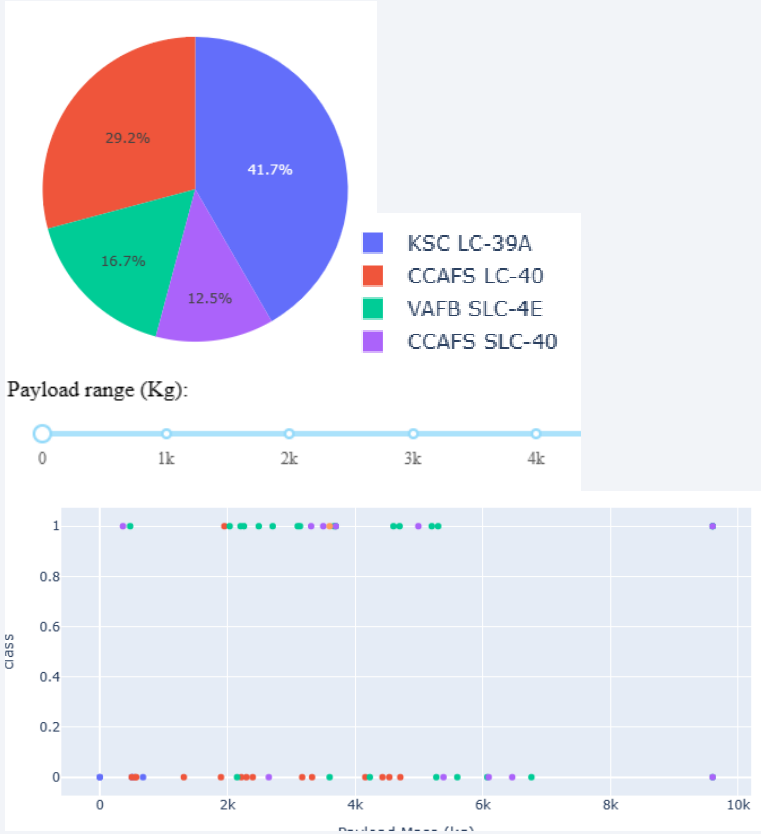
Build an Interactive Map with Folium

- Using an augmented dataset with latitude and longitude we create a map using the Folium library
- Using the circles, markers and clusters we add highlighted areas with a text label on a specific coordinate
- We can observe the map to extract important visual information such as distances from strategic points
- Notebook: [SpaceX-Capstone/IBM-DS0321EN-SkillsNetwork_labs_module_3_lab_jupyter_launch_site_location.jupyterlite.ipynb at main · jparisavila/SpaceX-Capstone (github.com)](https://github.com)



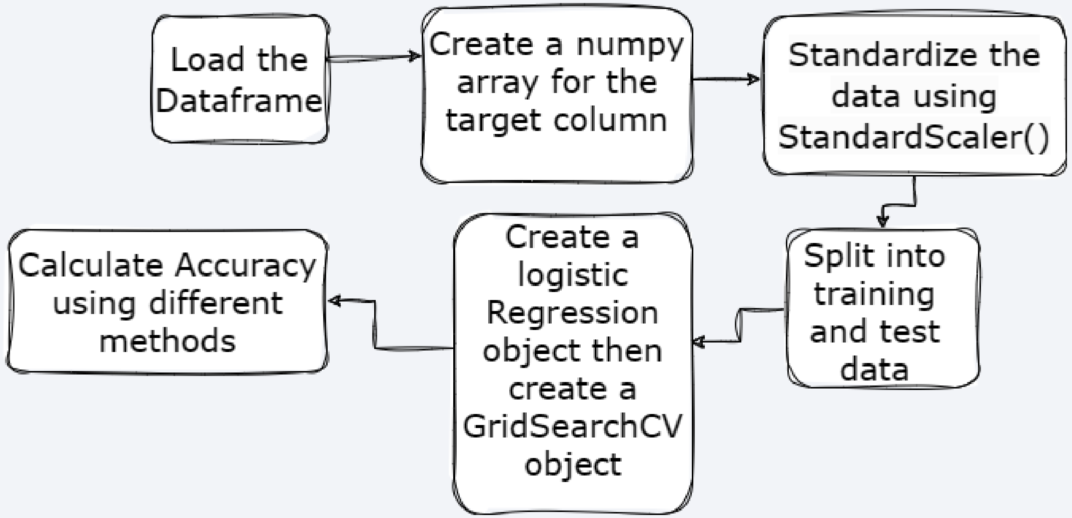
Build a Dashboard with Plotly Dash

- Using Plotly Dash we built a dashboard with a pie chart showing the percentages of success for each site and a scatter plot showing w Booster Version Categories were more successful in determined range of Payload mass.
- Using the interactive dashboard we can move the payload range and extract information about each category for each range of mass.
- Notebook: [SpaceX-Capstone/dash.py at main · jparisavila/SpaceX-Capstone (github.com)]()



Predictive Analysis (Classification)

- For the prediction analysis we build a classification model with the target column being the Class column (1 for a successful launch and 0 otherwise), using the flowchart on the right.
- Notebook: [SpaceX-Capstone/IBM-DS0321EN-SkillsNetwork_labs_module_4_SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb at main · jparisavila/SpaceX-Capstone (github.com)]



Results

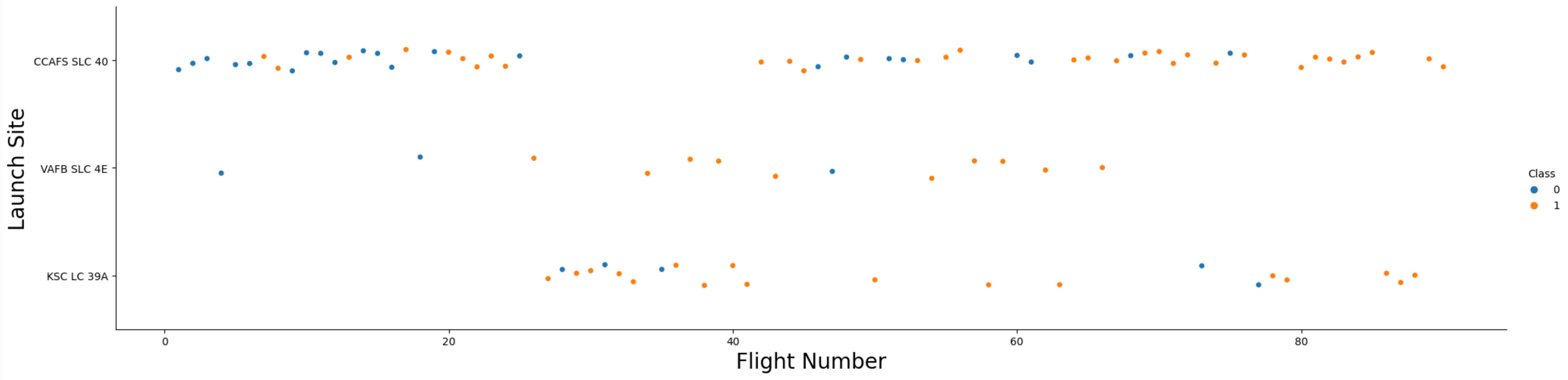
The results are shown as:

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

Section 2

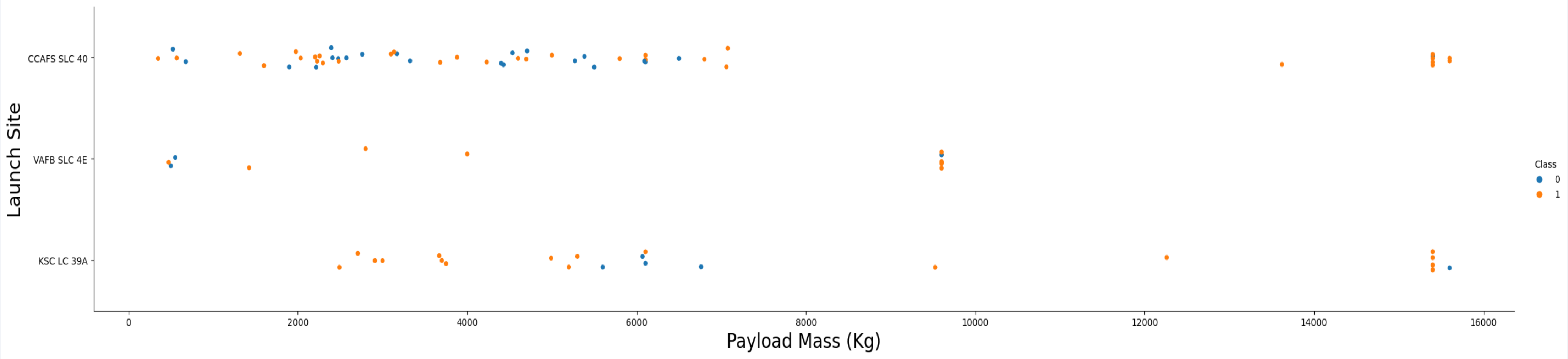
Insights drawn from EDA

Flight Number vs. Launch Site



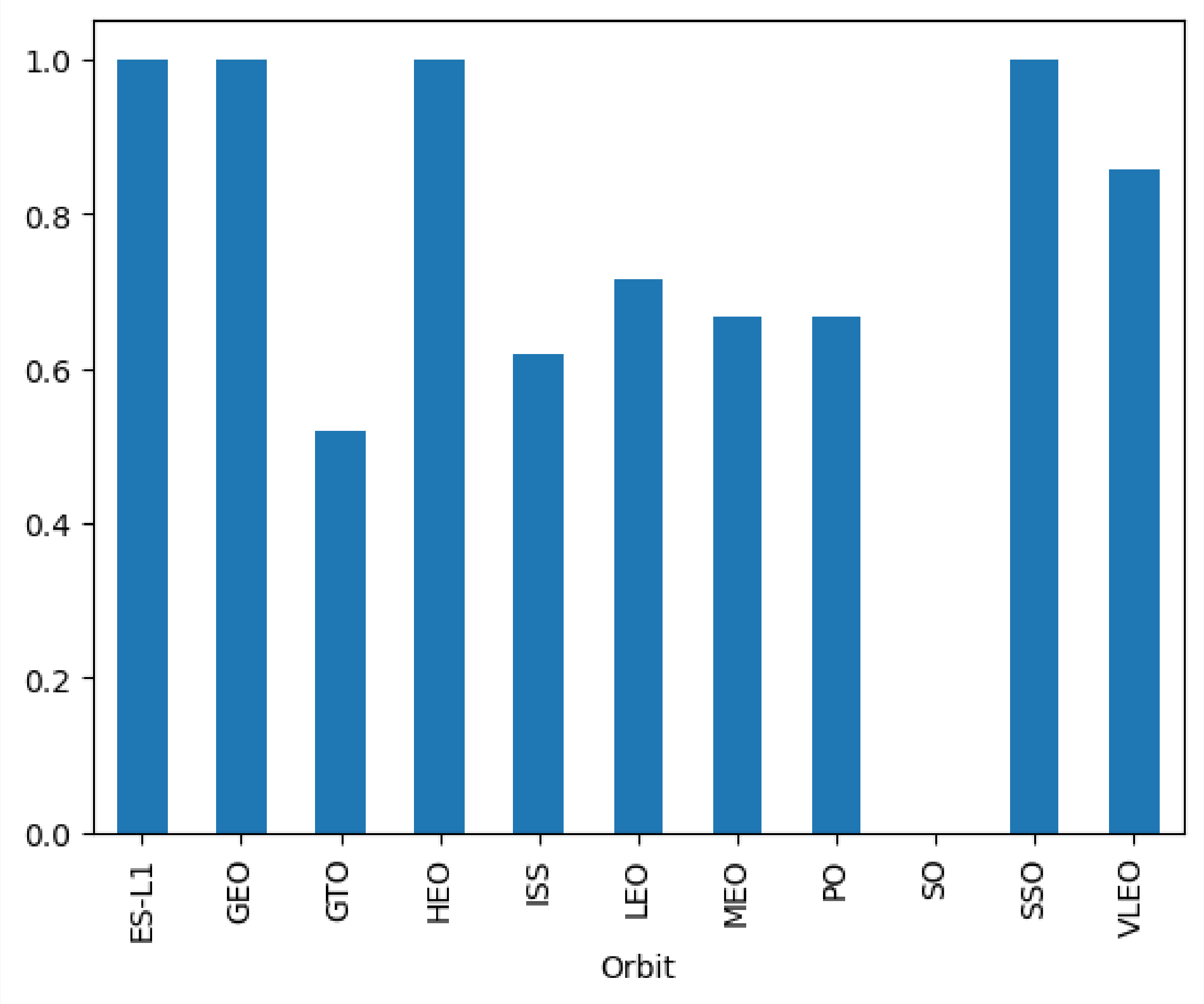
- CCAFS SLC 40 is the most used launch site, and concentrates more failures in na early stage of Falcon 9 project;
- KSL LC 39A shows more successes between flights number 40 and 70;
- VAFB SLC 4E shows a better success rate between flights number 20 and 70.

Payload vs. Launch Site



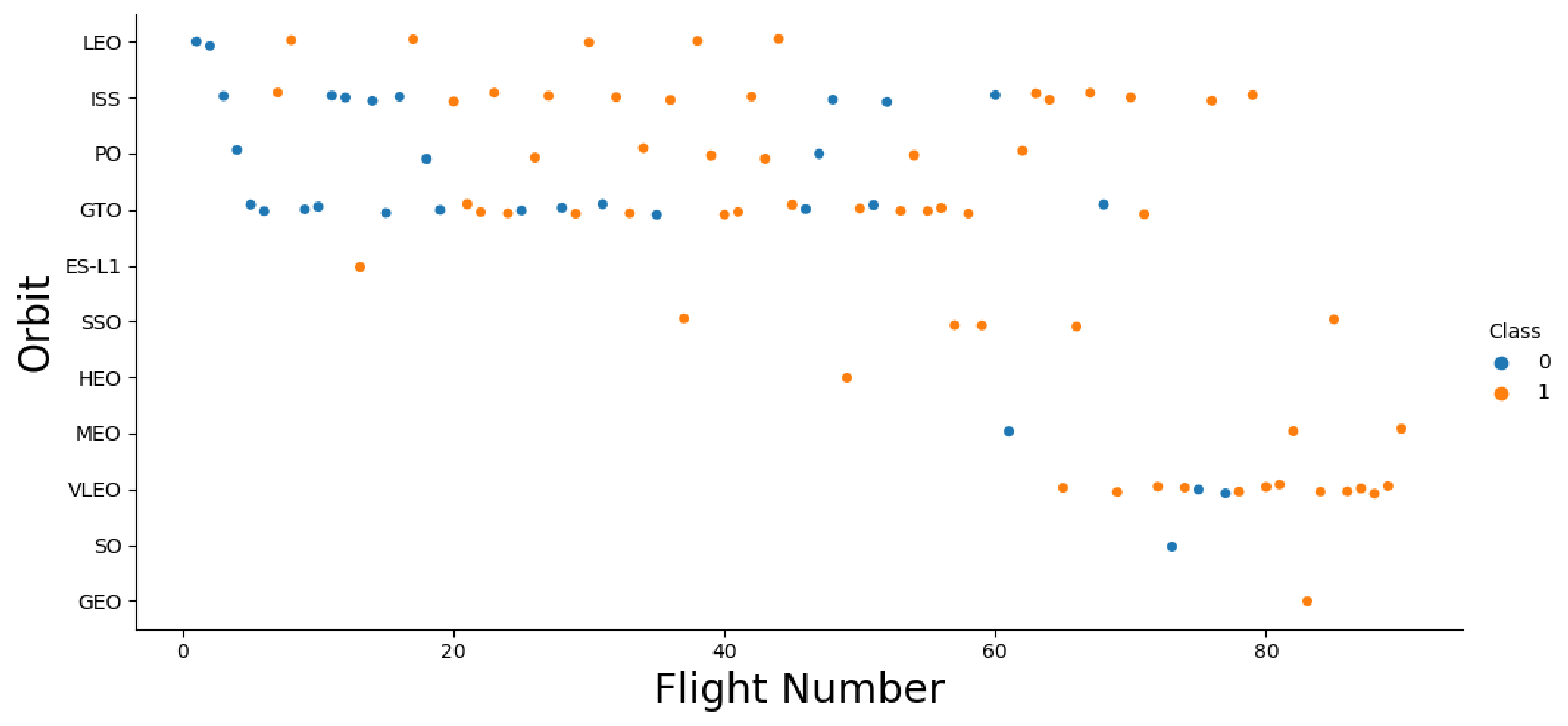
- The majority of lauches are from CCAFS SLC 40 with some successes with massive payload mass ;
- KSC LC 39A shows some success with payload mass between 2000 and 5000 kilograms.

Success Rate vs. Orbit Type



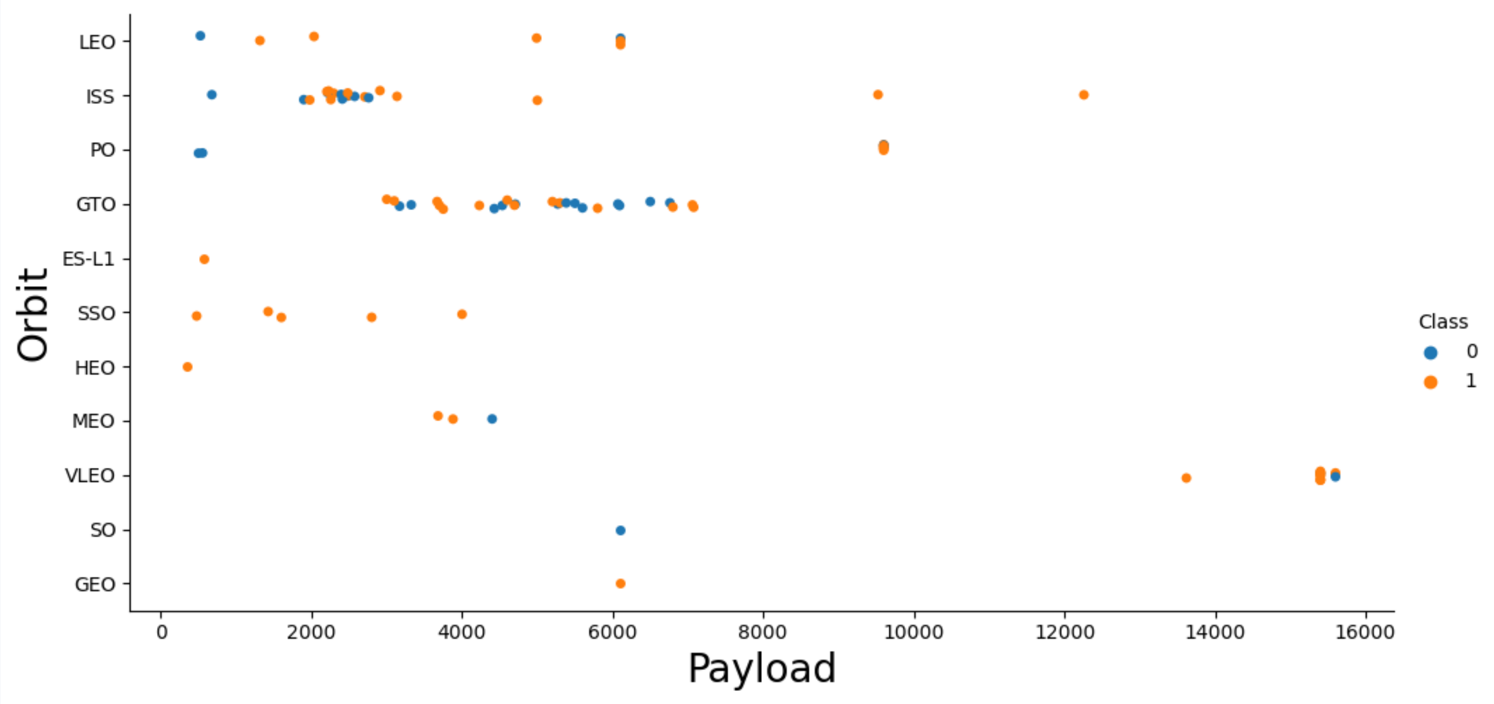
- The orbit types ES-L1, GEO, HEO and SSO
  are the highest success rates.

Flight Number vs. Orbit Type



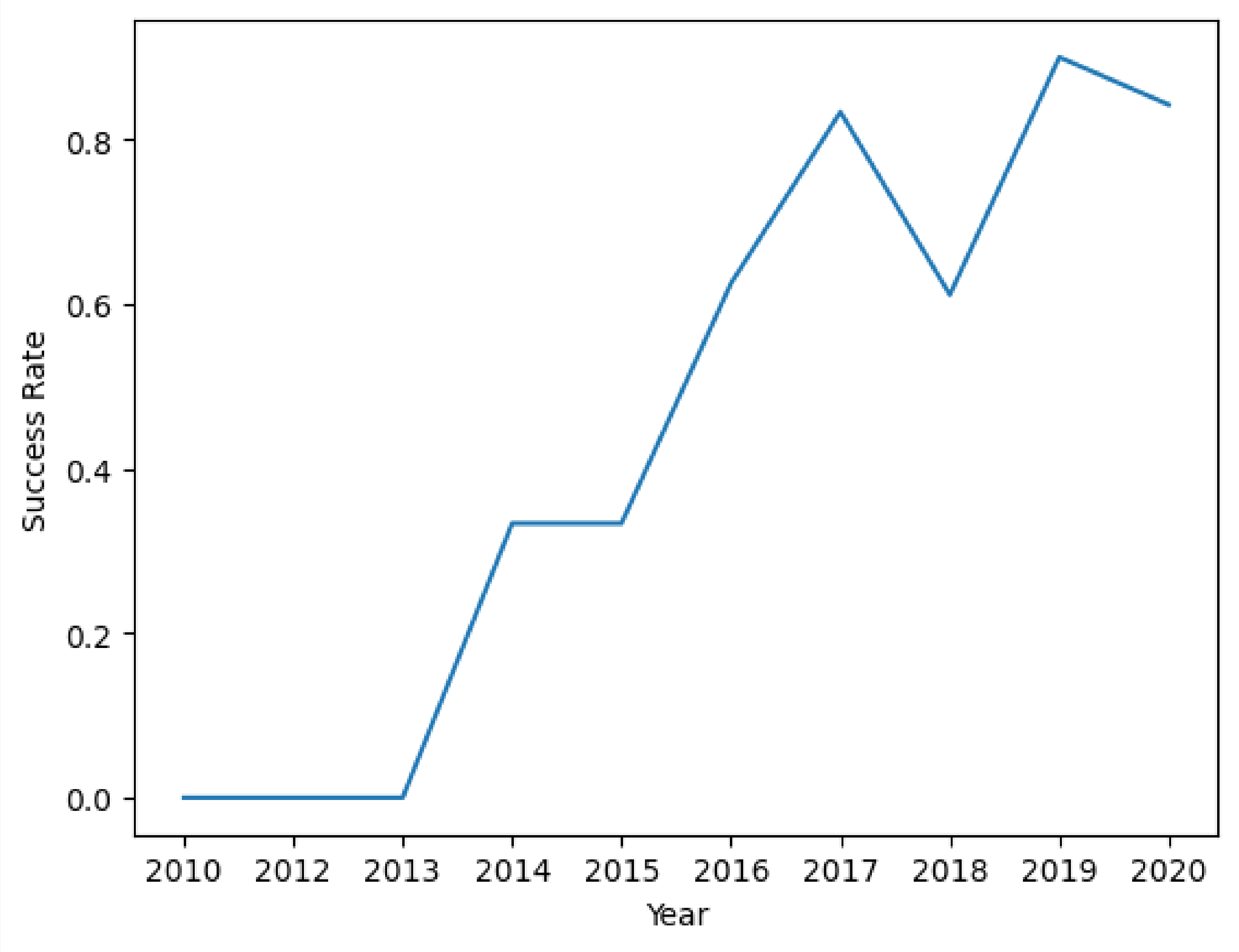
- We have a trend of launches for GTO orbit type until flight number 60;
- The trend changes for the VLEO orbit from flight number around 65 with a better success rate.

Payload vs. Orbit Type



- The launches concentrates at Orbits GTO and ISS;
- For the ISS orbit the payload mass is concentrated between 2000 and 4000 kilograms;
- For the gto the payload mass is concentrated between 3000 and 7000 kilograms.

Launch Success Yearly Trend



- The success rate has increased overs the years;
- The success rate has stablished around 2017 between 60% and 90%.

All Launch Site Names

- The following shows the query that displays the unique launch sites in the space mission:

```
%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'

- The following query displays 5 records where sites begin with the string 'CCA'

```
%sql SELECT * FROM SPACEXTBL WHERE Launch_Site like 'CCA%' LIMIT 5
```

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-04-06	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-08-12	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-05-22	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-08-10	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-01-03	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

- The following query displays the total payload carried by boosters from NASA

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

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

SUM(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

- The followignquery displays the average payload mass carried by booster version F9 v1.1:

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

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

AVG(PAYLOAD_MASS__KG_)
2928.4

First Successful Ground Landing Date

- The following query list the date when the first successful landing outcome in ground pad was achieved:

```
%sql SELECT MIN(Date) FROM SPACEXTBL WHERE Mission_Outcome = 'Success' and Landing_Outcome like '%ground%' ;
```

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

MIN(Date)
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- The following query list the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000:

```
%%sql SELECT Booster_Version FROM SPACEXTBL
where Mission_Outcome = 'Success' and Landing_Outcome like '%drone%'
AND PAYLOAD_MASS__KG_>4000 AND PAYLOAD_MASS__KG_<6000;
```

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

Booster_Version
F9 FT B1020
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
F9 FT B1031.2

Total Number of Successful and Failure Mission Outcomes

- The following query calculates the total number of successful and failure mission outcomes :

```
%%sql SELECT Mission_Outcome, COUNT(Mission_Outcome) from SPACEXTBL

GROUP BY(Mission_Outcome);
```

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

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

Boosters Carried Maximum Payload

- The following query lists the names of the booster which have carried the maximum payload mass

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

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

Booster_Version
F9 B5 B1048.4
F9 B5 B1049.4
F9 B5 B1051.3
F9 B5 B1056.4
F9 B5 B1048.5
F9 B5 B1051.4
F9 B5 B1049.5
F9 B5 B1060.2
F9 B5 B1058.3
F9 B5 B1051.6
F9 B5 B1060.3
F9 B5 B1049.7

2015 Launch Records

- The following query lists the failed landing outcomes in drone ship, their booster versions, and launch site names for in year 2015:

```
%%sql SELECT SUBSTR(Date, 6,2) AS MONTH, SUBSTR(Date, 1, 4) AS YEAR, Launch_Site, Booster_Version, Landing_Outcome from SPACEXTBL
WHERE YEAR = '2015' and Landing_Outcome like '%Failure (drone ship)%'
```

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

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

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

- The following query ranks 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

```
%%sql SELECT Landing_Outcome, COUNT(*) AS Count_Launches FROM SPACEXTBL
WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY Landing_Outcome ORDER BY COUNT_LAUNCHES DESC;
```

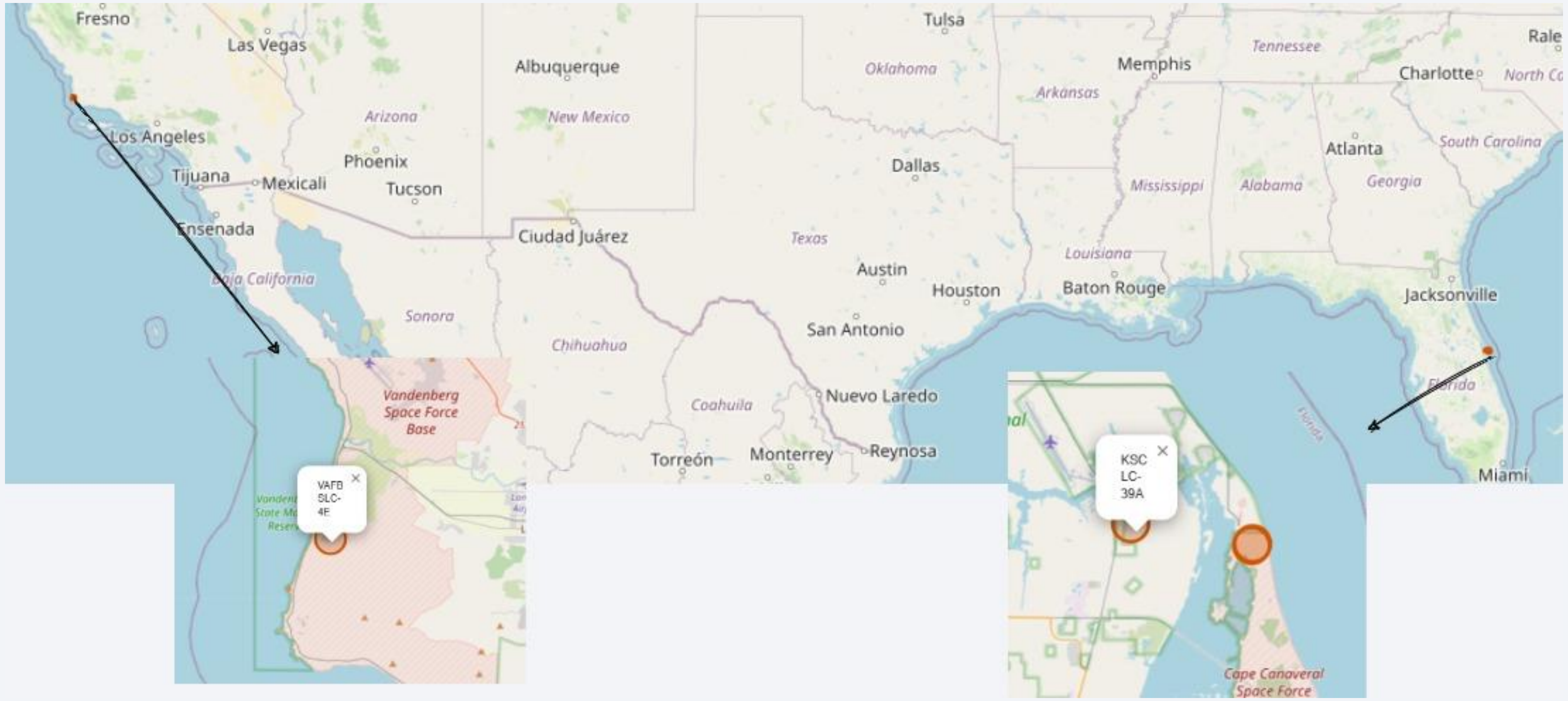
```
 * sqlite:///my_data1.db
Done.
```

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

Section 3

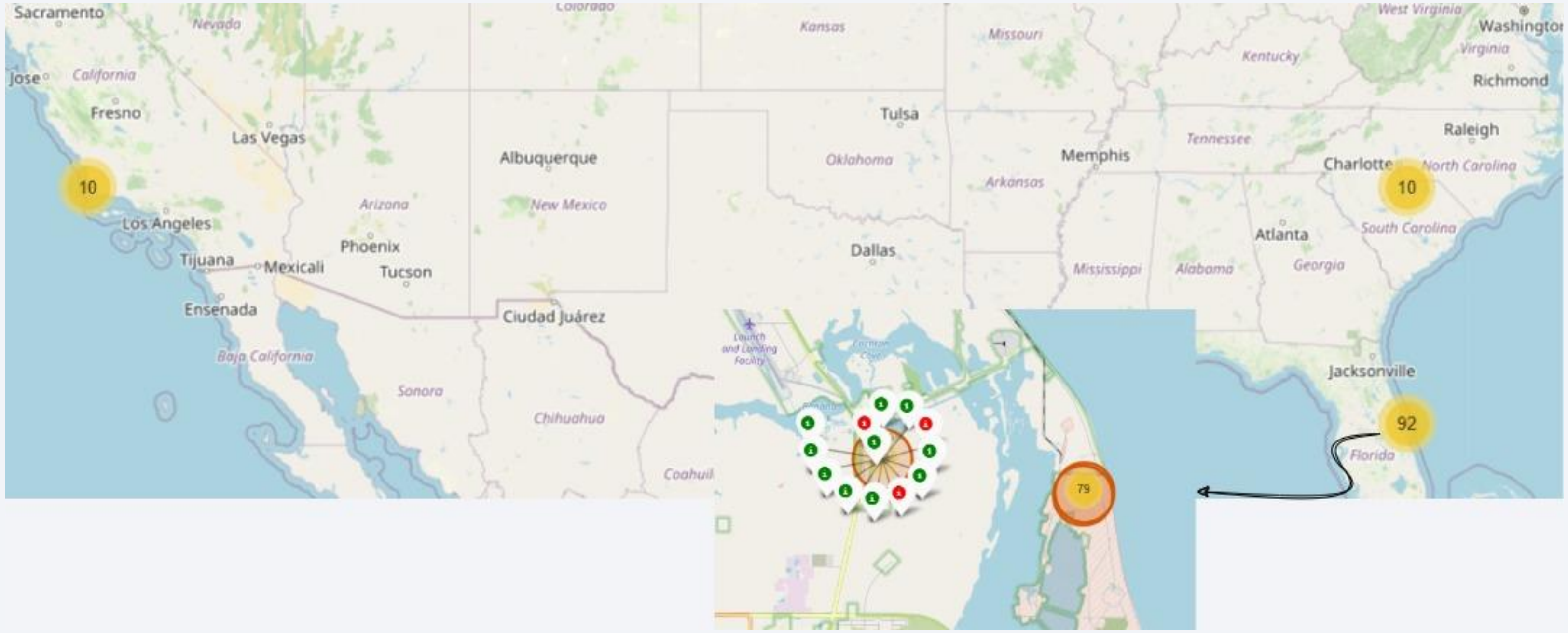
Launch Sites Proximities Analysis

Location Sites Map



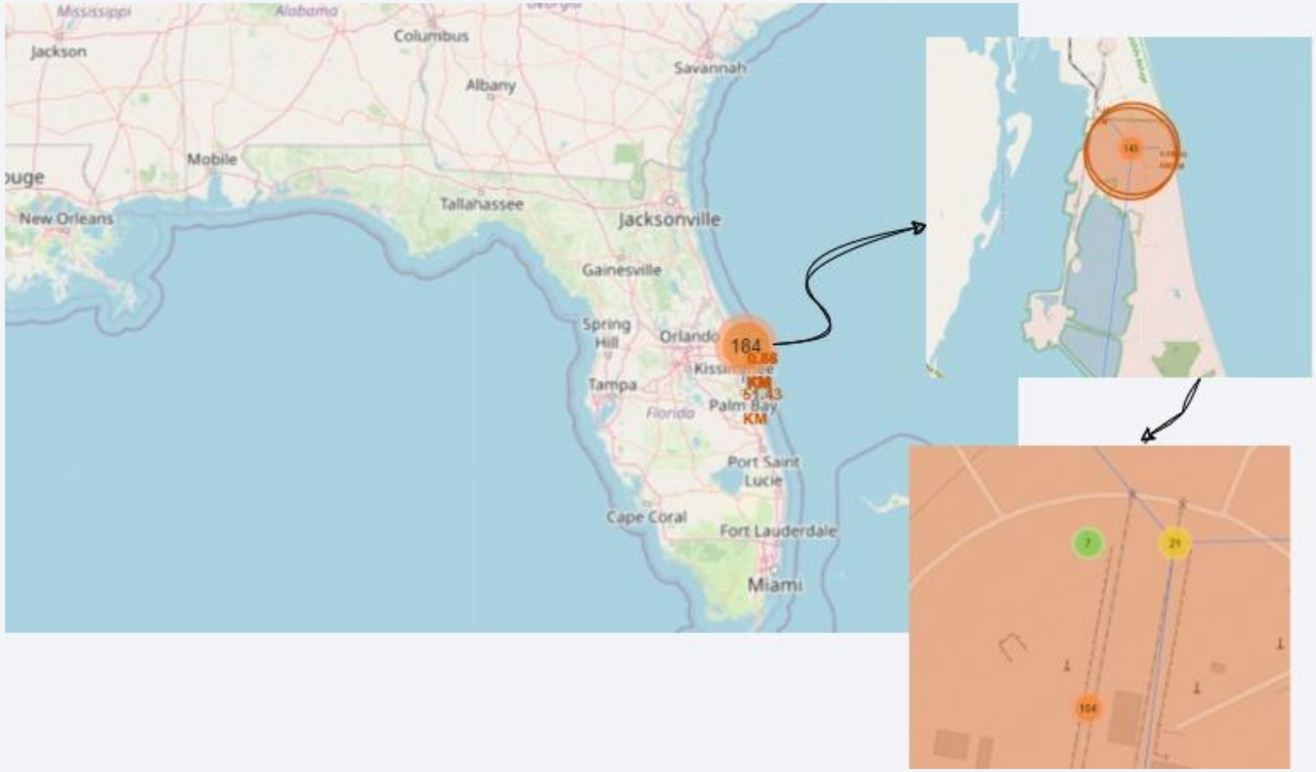
This map shows the locations sites with a marker and also shows their names with a click.

Success and Failed Launches



The map shows clusters of the sites. Clicking the sites with the mouse, it shows the sites with the color green for success and red for failure.

Distances between launch sites to its proximities

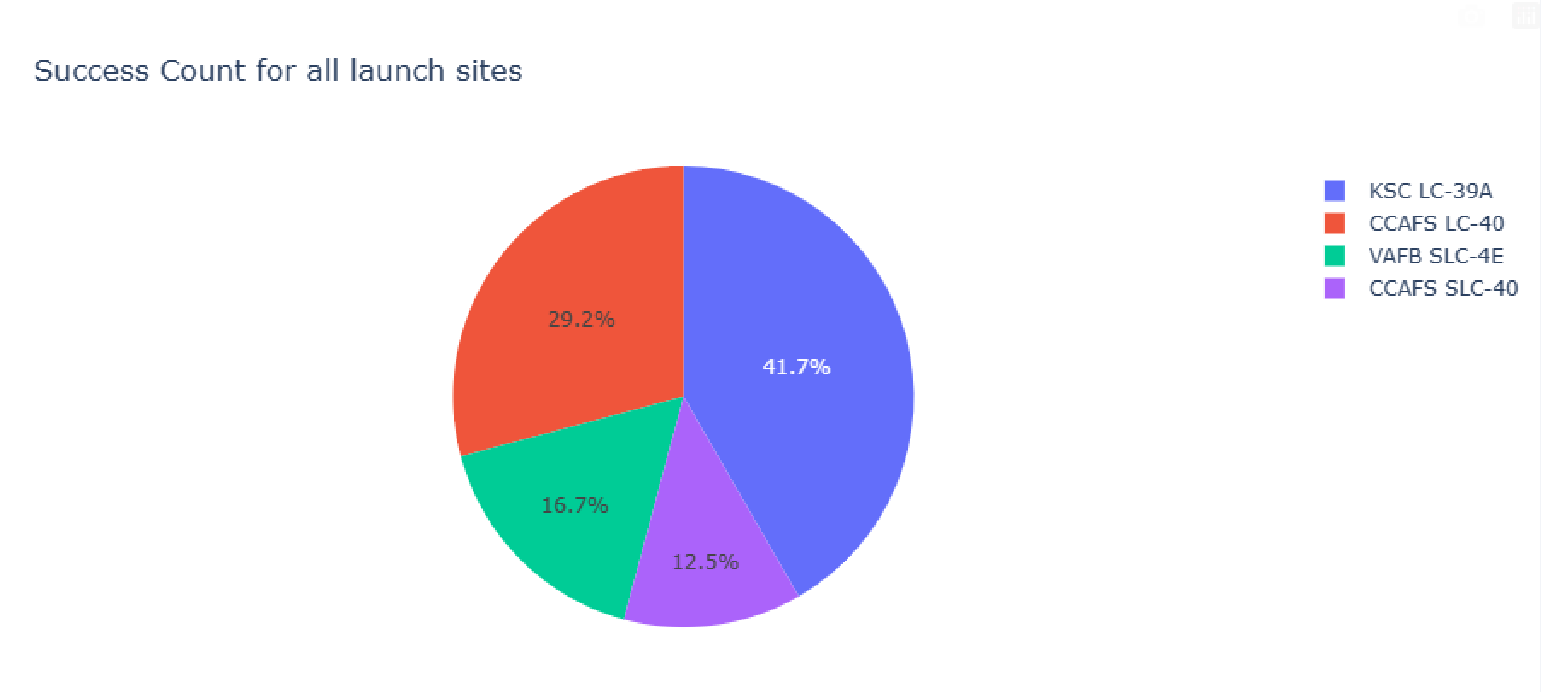


- We can also use Folium Map to calculate and show distances from railways and shores. Clicking the map we can see clearly the sites and the railways.

Section 4

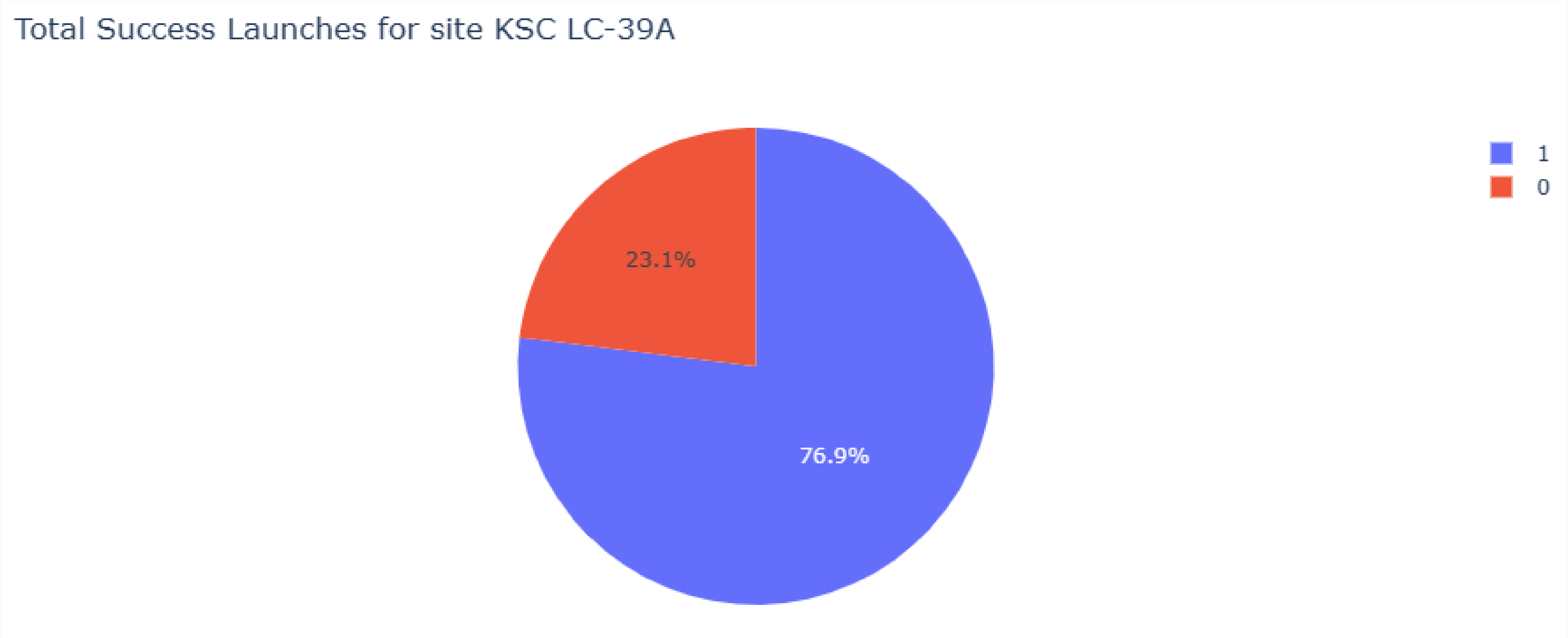
Build a Dashboard with Plotly Dash

Success Count for all sites



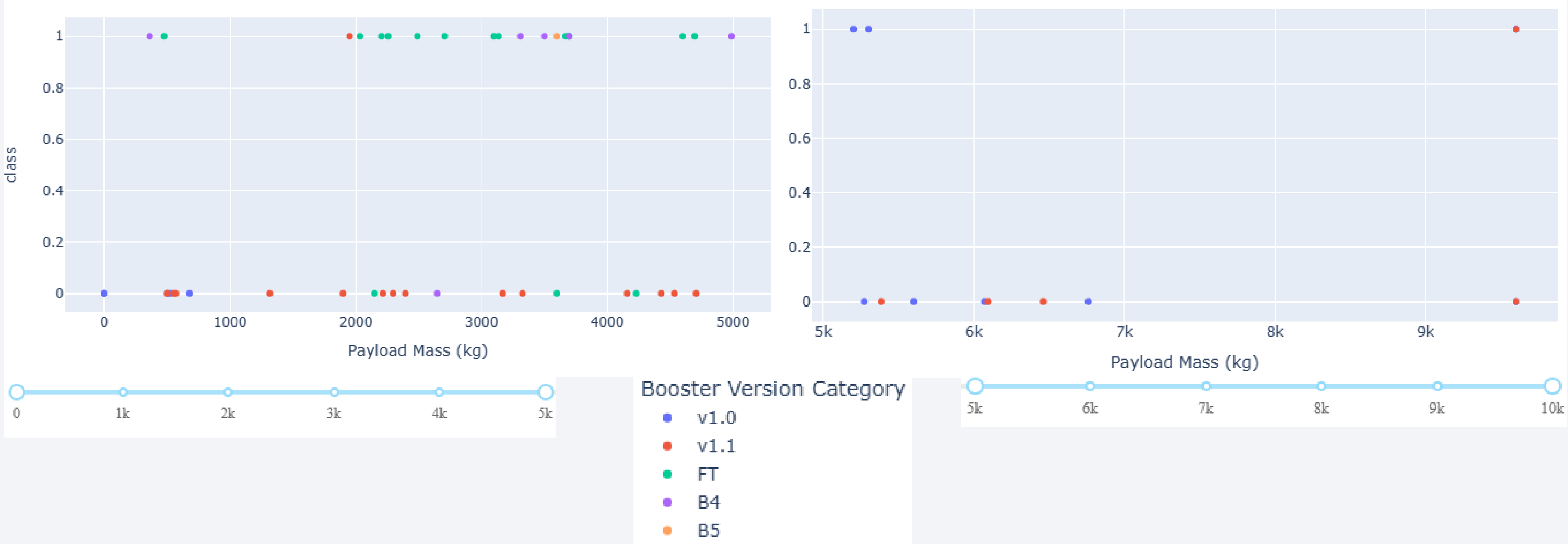
- In this pie chart we have the percentage of succeeded launches for each site. The KSC LC-39A is the more succeeded with 41.7%.

Success Launches for KSC LC 39-A



- The piechart for the launch site with highest launch success ratio (KSC LC 39-A) shows the percentage of success (1 or blue area) versus failure (2 or red area).

Payload vs. Launch Outcome

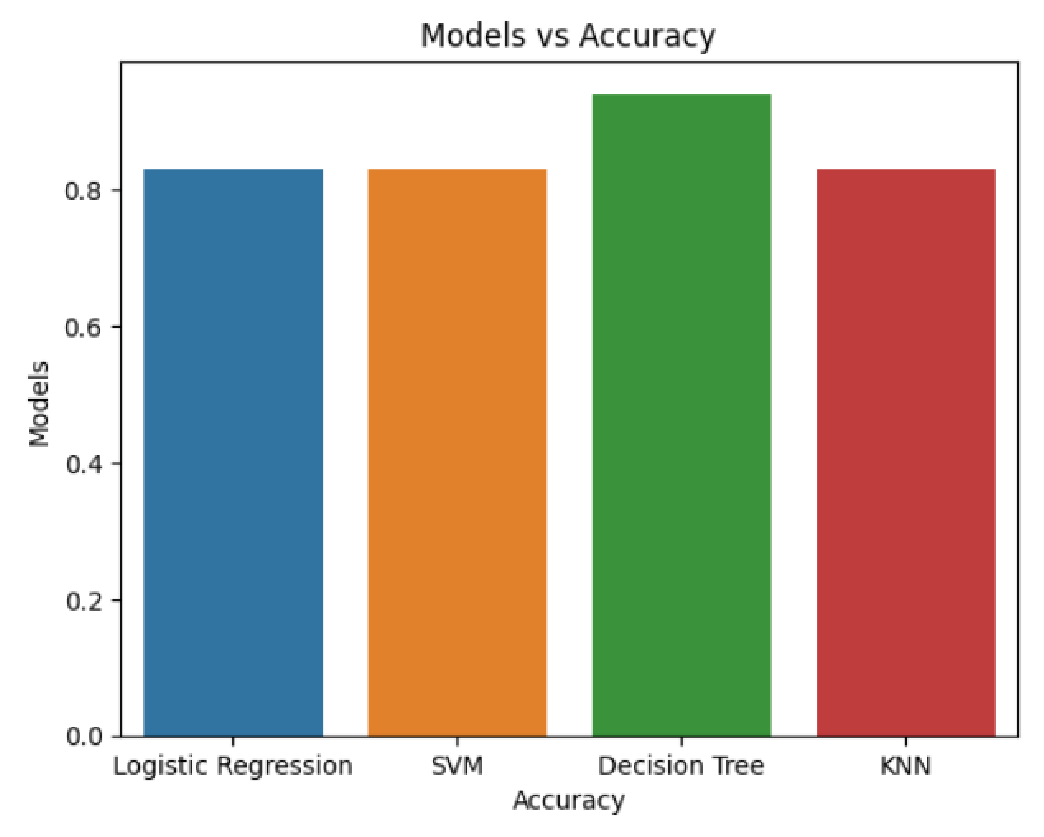


- We have more launches for payload mass from 0 to 5k than for 5k to 10k. There are only two categories for the heavier masses, v1.o and v1.1, which have more failures (0) than successes (1).

Section 5

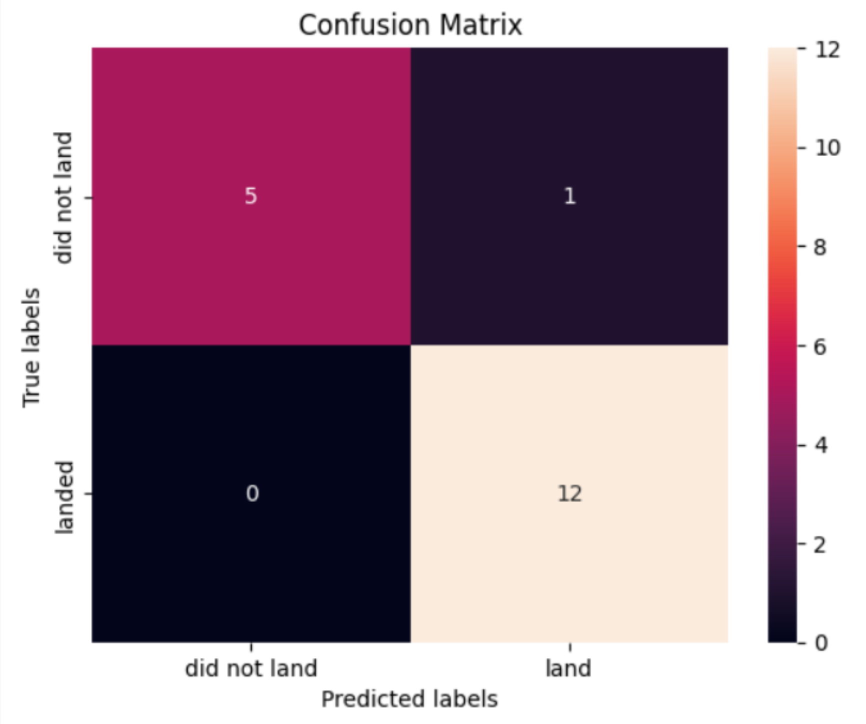
Predictive Analysis (Classification)

Classification Accuracy



In our case, the model with best accuracy was the Decision Tree Model with about 94.44%.

Confusion Matrix



The confusion matrix for the best model (Decision Tree), shows 12 True Positives and 5 True Negative, with a low number of False Positives (1) and a null number for False Negative.

Conclusions

- CCAFS SLC 40 is the most used launch site, and concentrates more failures in na early stage of Falcon 9 project;
- KSL LC 39A shows more successes between flights number 40 and 70;
- VAFB SLC 4E shows a better success rate between flights number 20 and 70.
- The orbit types ES-L1, GEO, HEO and SSO are the highest success rates.
- The success rate has increased overs the years;
- The model with best accuracy was the Decision Tree Model with about 94.44%.

Thank you!