IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Juan Manuel Mesa

July 1st 2022

Outline

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

Executive Summary

- Summary of methodologies

- Data Collection with API

- Data Wrangling

- Exploratory Data analysis with sql and Data visualization

- Visual analytic with Folium

- Machine learning modeling

- Summary of all results

- EDA result

- Visualization (Plots, Maps, Charts)

- Predictive modeling

Introduction

- Project background and context

Space X is an Aerospatiale company, under the leadership of Elon Musk. SpaceX offers rocket launches services much cheaper than other companies, mostly because of the reusability of the First stage of their rockets, in particular the Falcon 9, which is capable of land after a lunch. We aim to create a predictive model that could tell if the rocket will be able to land successfully.

- Problems you want to find answers

What features determine a successful landing?

What is the most accurate model to describe the outcome of the landing?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - The data was collected using a SpaceX API
- Perform data wrangling
  - One Hot enconding and further manipulation was performed to the data
- 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

- Data was collected using SpaceX API
- We defined functions to get : Booster version, Launch, site, PayLoad Data and Core Data.
- We defined the request to get the data with the requests.get() function
- We normalized the JSON and transformed it into a pandas DataFrame with pd.json_normalize() function
- We manipulated the data, to select only data from the Falcon 9
- Finally, we did some data engineering and handled missing values.

Data Collection – SpaceX API

- Present your data collection with SpaceX REST calls using key phrases and flowcharts

- GitHub Url

```python
static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_call_spacex_api.json'
```

Python

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

```python
response.status_code
```

Python

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

```python
# Use json_normalize meethod to convert the json result into a dataframe
static_json_df = response.json()
data = pd.json_normalize(static_json_df)
```

Python

Using the dataframe `data` print the first 5 rows

Data Wrangling

- Exploratory Data analysis was performed.
- Calculated number of lunches from the different launch site, and the target orbit of the rocker
- At the end, we exported the data set as a .csv file.

```
df.head(5)
```

Python

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	Longitude	Latitude	Class
0	1	2010-06-04	Falcon 9	6104.959412	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0003	-80.577366	28.561857	0
1	2	2012-05-22	Falcon 9	525.000000	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0005	-80.577366	28.561857	0
2	3	2013-03-01	Falcon 9	677.000000	ISS	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0007	-80.577366	28.561857	0
3	4	2013-09-29	Falcon 9	500.000000	PO	VAFB SLC 4E	False Ocean	1	False	False	False	NaN	1.0	0	B1003	-120.610829	34.632093	0
4	5	2013-12-03	Falcon 9	3170.000000	GTO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B1004	-80.577366	28.561857	0

We can use the following line of code to determine the success rate:

- https://github.com/Juanma814/IBM-Data-Science-Capstone-project/blob/1c3f0772dbce1eb8744cec6b4e013cf38c869832/Data%20Wrangling.ipynb

EDA with Data Visualization

- We displayed different charts, a line chart to evaluate how the landind success varies with the year.
- Also, we displayed bar charts to compare landing rates between the rockets with different target orbit.
- Finally, we also displayed some scatter plot, to analyse possible relationship between variables.
- https://github.com/Juanma814/IBM-Data-Science-Capstone-project/blob/1c3f0772dbce1eb8744cec6b4e013cf38c869832/Data%20visualization.ipynb

EDA with SQL

- We used the **WHERE** clause to filter for boosters which have successfully landed.
- We use **AND** condition.
- We used the wildcard ‘%’ while filtering with **WHILE**
- Finally, we also filtered with the function **Max()**

- https://github.com/Juanma814/IBM-Data-Science-Capstone-project/blob/1c3f0772dbce1eb8744cec6b4e013cf38c869832/EDA%20with%20SQL.ipynb

Build an Interactive Map with Folium

- We added Markers and Marker Cluster in the coordinates of Rocket's Launches. It was possible to identify location of those Launches (Florida and California, USA).
- Also, lines were drawn to the nearest coast, city and road.
- Add the GitHub URL of your completed interactive map with Folium map, as an external reference and peer-review purpose

- https://github.com/Juanma814/IBM-Data-Science-Capstone-project/blob/1c3f0772dbce1eb8744cec6b4e013cf38c869832/Site%20location.ipynb

Build a Dashboard with Plotly Dash

- We built interactive Dashboards with Plotly
- We plotted scatter graph with the relationship between Outcome and Payload Mass (Kg) for the different booster version
- Finally, we plotted pi charts with the Launchings per Launch Site

Predictive Analysis (Classification)

- We manipulated the data using standar tools, as pandas and numpy.
- We divided our data into a train set, and a test set.
- We train different models using GridSearch to find the best parameters.
- We used the score function, and the test set to evaluate the models, and find the best one.
- We found the best model for this case.

- https://github.com/Juanma814/IBM-Data-Science-Capstone-project/blob/1c3f0772dbce1eb8744cec6b4e013cf38c869832/SpaceX_Machine%20Learning%20Prediction.ipynb

Results

- Exploratory Data analysis was performed. And relations between the descriptors were found.
- Plots, Interactive Dashs, Charts, and Graphs are presented with the founding in the next chapter.
- The best predictive model for this case was the DecisionTree algorithm

Find the method performs best:

```python
models = {'KNeighbors':knn_cv.best_score_, 'DecisionTree':tree_cv.best_score_,'LogisticRegression':logreg_cv.best_score_,'SupportVector': svm_cv.best_score_}
Max_score = 0
for score in models:
    if models[score] >= Max_score:
        Max_score = models[score]
        key_max_score = score

print('Best method is :'+ str(key_max_score))
print('Max_Score is :'+ str(Max_score))
```

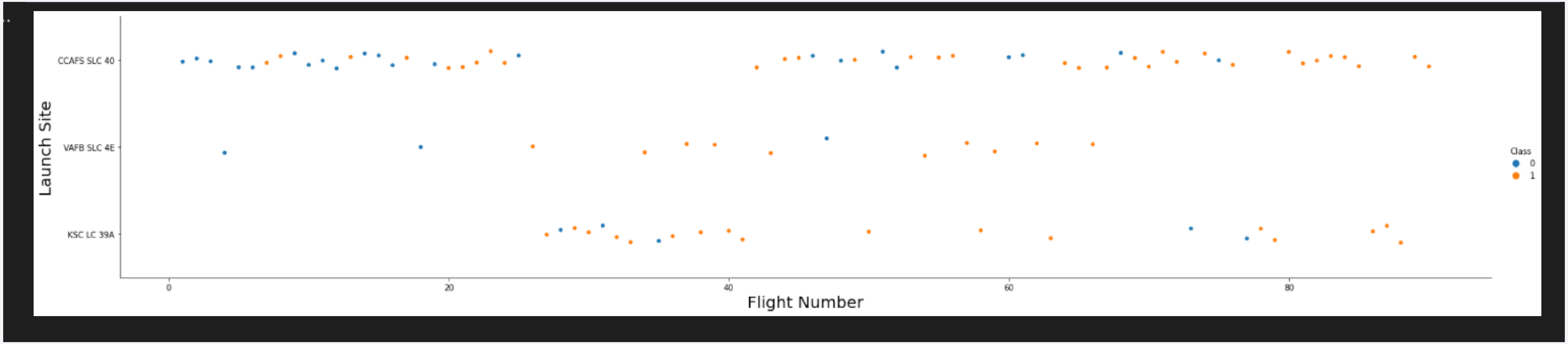
Python

```
Best method is :DecisionTree
Max_Score is :0.8714285714285713
```

Section 2

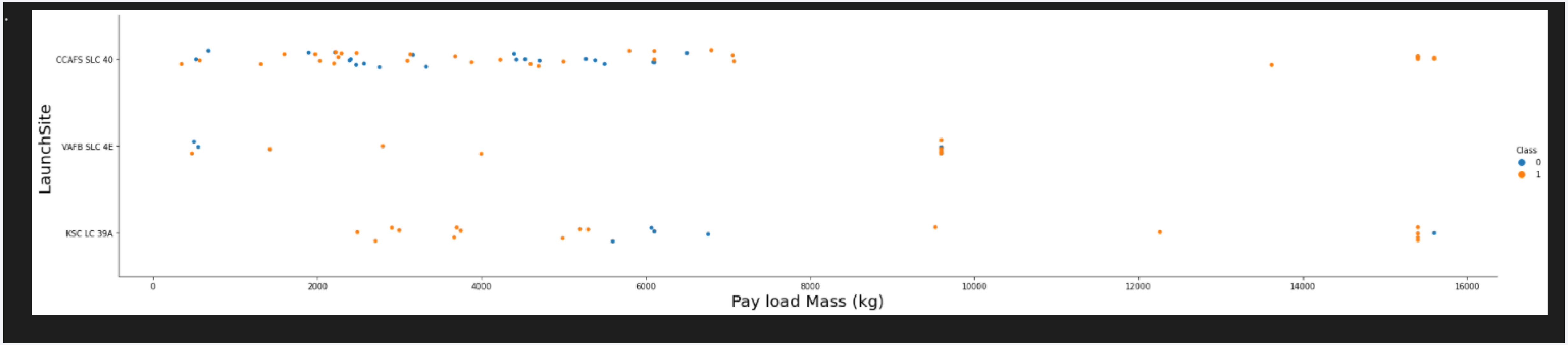
Insights drawn from EDA

Flight Number vs. Launch Site



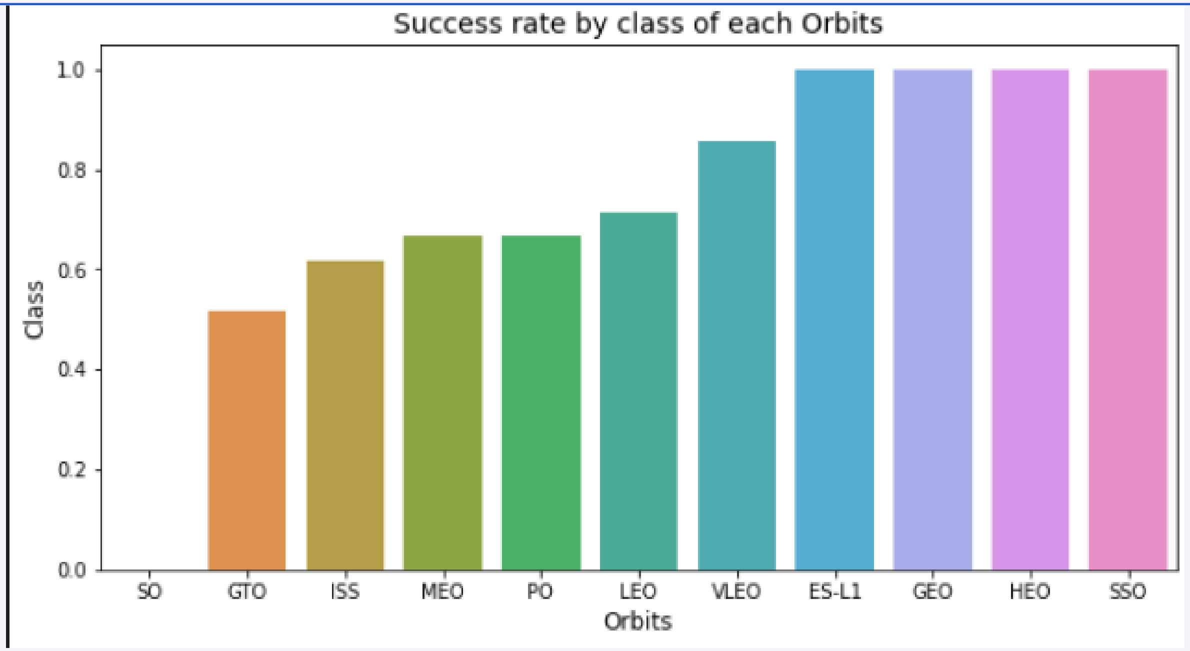
- The greater the flight number, the greater the success rate

Payload vs. Launch Site



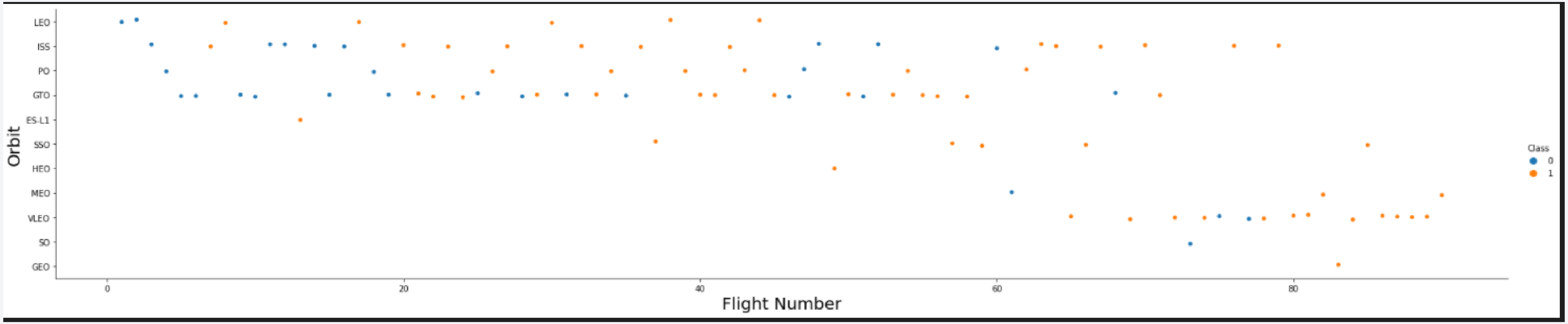
- No heavy Rockets were luched from VAFB SLC 4E
- The Higher the Pay Load, the Higher the success rate

Success Rate vs. Orbit Type



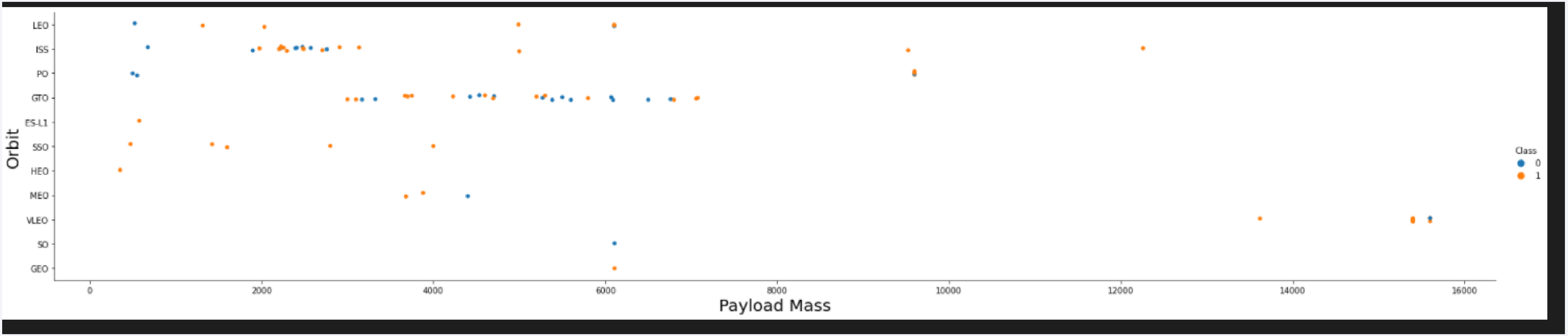
- ES-L1, GEO, HEO, and SSO has the greater success rate, almost complete.

Flight Number vs. Orbit Type



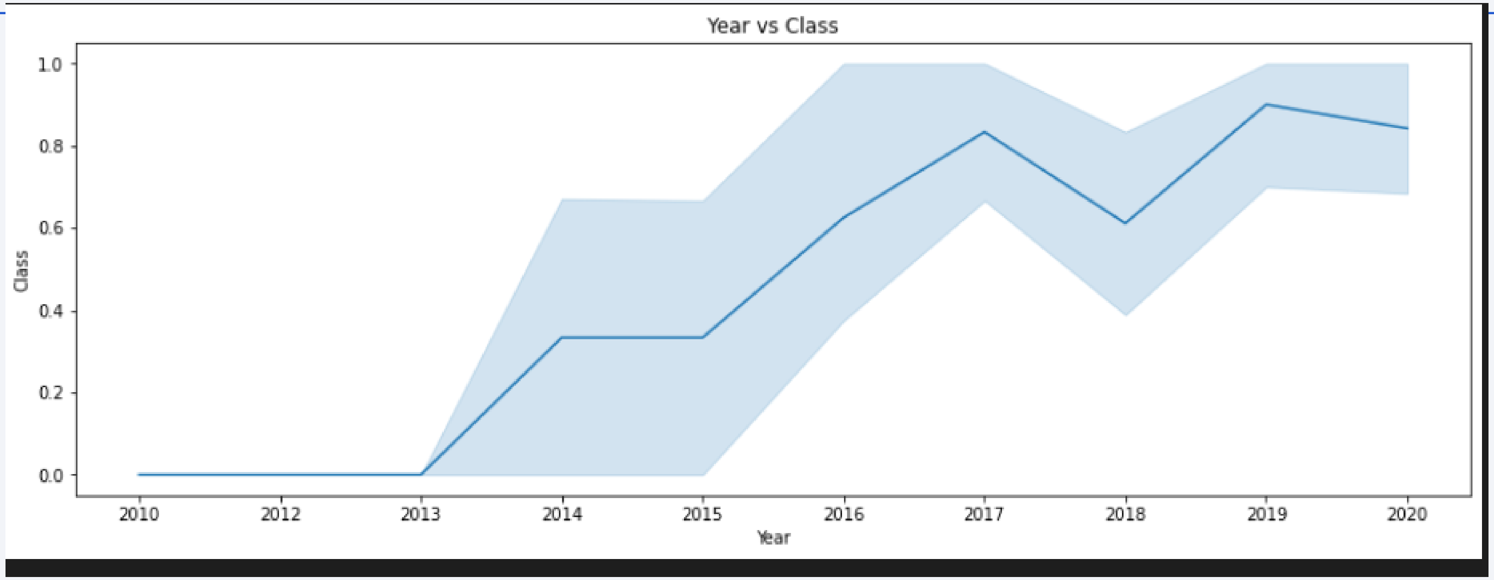
Highest Fight number has more chance to be VLEO

Payload vs. Orbit Type



- Hight Payload (Higher than 14000) are exclusibly VLEO orbiT type

Launch Success Yearly Trend



- Success rate is being increasing since 2013

All Launch Site Names

```
task_1 = '''
        SELECT DISTINCT LaunchSite
        FROM SpaceX
'''
create_pandas_df(task_1, database=conn)
```

	launchsite
0	KSC LC-39A
1	CCAFS LC-40
2	CCAFS SLC-40
3	VAFB SLC-4E

- There are 4 different launch sites

Launch Site Names Begin with 'CCA'

Display 5 records where launch sites begin with the string 'CCA'

In [11]:

```
task_2 = '''
        SELECT *
        FROM SpaceX
        WHERE LaunchSite LIKE 'CCA%'
        LIMIT 5
        '''
create_pandas_df(task_2, database=conn)
```

Out[11]:

	date	time	boosterversion	launchsite	payload	payloadmasskg	orbit	customer	missionoutcome	landingoutcome
0	2010-04-06	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
1	2010-08-12	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of...	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2	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
3	2012-08-10	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
4	2013-01-03	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

- We display five rows with CCA as first three letters in the launchsite

Total Payload Mass

Display the total payload mass carried by boosters launched by NASA (CRS)

```
task_3 = '''
        SELECT SUM(PayloadMassKG) AS Total_PayloadMass
        FROM SpaceX
        WHERE Customer LIKE 'NASA (CRS)'
        '''
create_pandas_df(task_3, database=conn)
```

	total_payloadmass
0	45596

- Total payload carried by NASA is 45596

Average Payload Mass by F9 v1.1

Display average payload mass carried by booster version F9 v1.1

```
task_4 = '''
        SELECT AVG(PayloadMassKG) AS Avg_PayloadMass
        FROM SpaceX
        WHERE BoosterVersion = 'F9 v1.1'
        '''
create_pandas_df(task_4, database=conn)
```

	avg_payloadmass
0	2928.4

- Average payload mass carried by booster version F9 v1.1 is 2928.4

First Successful Ground Landing Date

```
task_5 = '''
        SELECT MIN(Date) AS FirstSuccessfull_landing_date
        FROM SpaceX
        WHERE LandingOutcome LIKE 'Success (ground pad)'
        '''
create_pandas_df(task_5, database=conn)
```

	firstsuccessfull_landing_date
0	2015-12-22

- First successful landing outcome on ground pad was 22 December 2015

Successful Drone Ship Landing with Payload between 4000 and 6000

```
task_6 = '''
        SELECT BoosterVersion
        FROM SpaceX
        WHERE LandingOutcome = 'Success (drone ship)'
            AND PayloadMassKG > 4000
            AND PayloadMassKG < 6000
        '''
create_pandas_df(task_6, database=conn)
```

	boosterversion
0	F9 FT B1022
1	F9 FT B1026
2	F9 FT B1021.2
3	F9 FT B1031.2

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

Total Number of Successful and Failure Mission Outcomes

List the total number of successful and failure mission outcomes

```
task_7a = '''
        SELECT COUNT(MissionOutcome) AS SuccessOutcome
        FROM SpaceX
        WHERE MissionOutcome LIKE 'Success%'
        '''

task_7b = '''
        SELECT COUNT(MissionOutcome) AS FailureOutcome
        FROM SpaceX
        WHERE MissionOutcome LIKE 'Failure%'
        '''
print('The total number of successful mission outcome is:')
display(create_pandas_df(task_7a, database=conn))
print()
print('The total number of failed mission outcome is:')
create_pandas_df(task_7b, database=conn)
```

```
The total number of successful mission outcome is:
```

	successoutcome
0	100

```
The total number of failed mission outcome is:
```

	failureoutcome
0	1

- Total number of successful and failure mission outcomes

Boosters Carried Maximum Payload

List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

In [17]:

```
task_8 = '''
         SELECT BoosterVersion, PayloadMassKG
         FROM SpaceX
         WHERE PayloadMassKG = (
                                 SELECT MAX(PayloadMassKG)
                                 FROM SpaceX
                                 )
         ORDER BY BoosterVersion
         '''
create_pandas_df(task_8, database=conn)
```

Out[17]:

	boosterversion	payloadmasskg
0	F9 B5 B1048.4	15600
1	F9 B5 B1048.5	15600
2	F9 B5 B1049.4	15600
3	F9 B5 B1049.5	15600
4	F9 B5 B1049.7	15600
5	F9 B5 B1051.3	15600
6	F9 B5 B1051.4	15600
7	F9 B5 B1051.6	15600
8	F9 B5 B1056.4	15600
9	F9 B5 B1058.3	15600
10	F9 B5 B1060.2	15600
11	F9 B5 B1060.3	15600

- Name of the booster which have carried the maximum payload mass

2015 Launch Records

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

In [18]:

```
task_9 = '''
        SELECT BoosterVersion, LaunchSite, LandingOutcome
        FROM SpaceX
        WHERE LandingOutcome LIKE 'Failure (drone ship)'
            AND Date BETWEEN '2015-01-01' AND '2015-12-31'
        '''
create_pandas_df(task_9, database=conn)
```

Out[18]:

	boosterversion	launchsite	landingoutcome
0	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
1	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

- Failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 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))

In [19]:
```
task_10 = '''
        SELECT LandingOutcome, COUNT(LandingOutcome)
        FROM SpaceX
        WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20'
        GROUP BY LandingOutcome
        ORDER BY COUNT(LandingOutcome) DESC
        '''
create_pandas_df(task_10, database=conn)
```

Out[19]:

	landingoutcome	count
0	No attempt	10
1	Success (drone ship)	6
2	Failure (drone ship)	5
3	Success (ground pad)	5
4	Controlled (ocean)	3
5	Uncontrolled (ocean)	2
6	Precluded (drone ship)	1
7	Failure (parachute)	1

- Landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order

Section 3

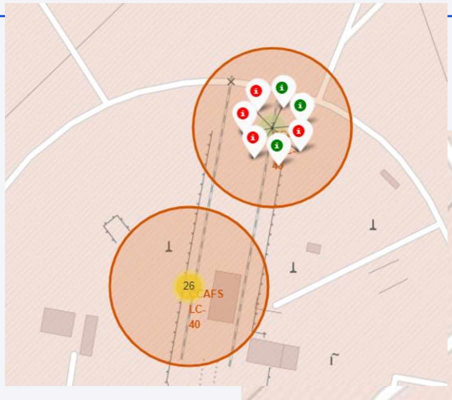
Launch Sites Proximities Analysis

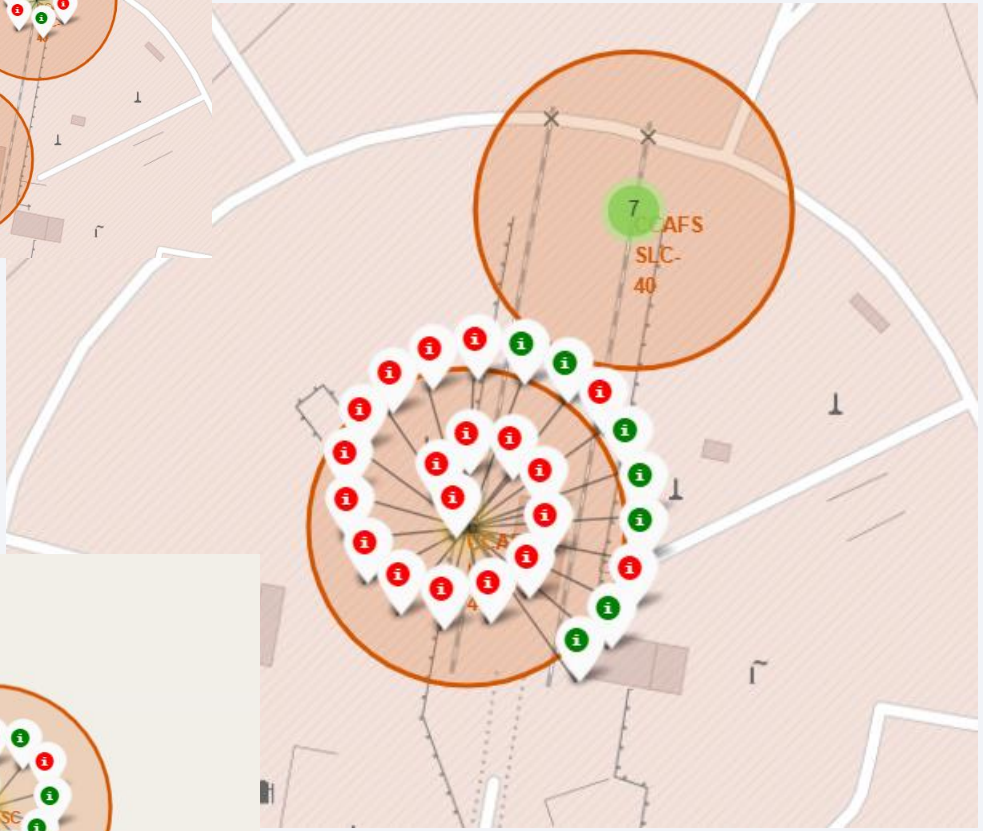
Global Map

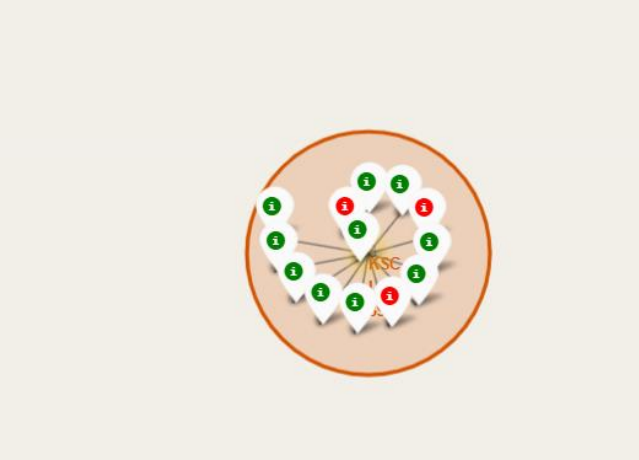
- Launching sites are in the USA. In the west(California) and east (Florida) coast

Launch sites

- We added Marker clusters for success lunches (green) and failed lunches(red).







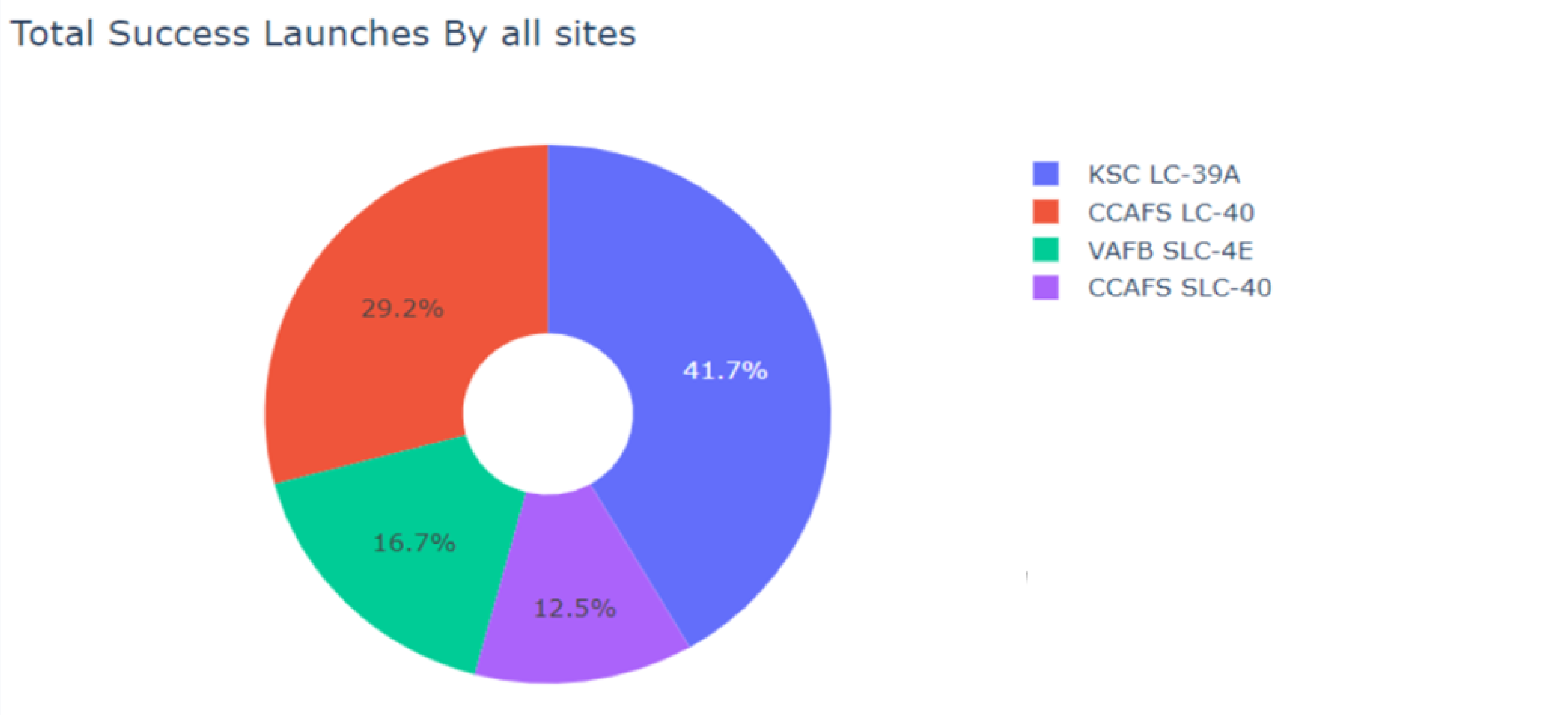
<Folium Map Screenshot 3>

Lunch sites are close to the coast (Lest than 2 km), and far away from railways, roads and cities.

Section 4

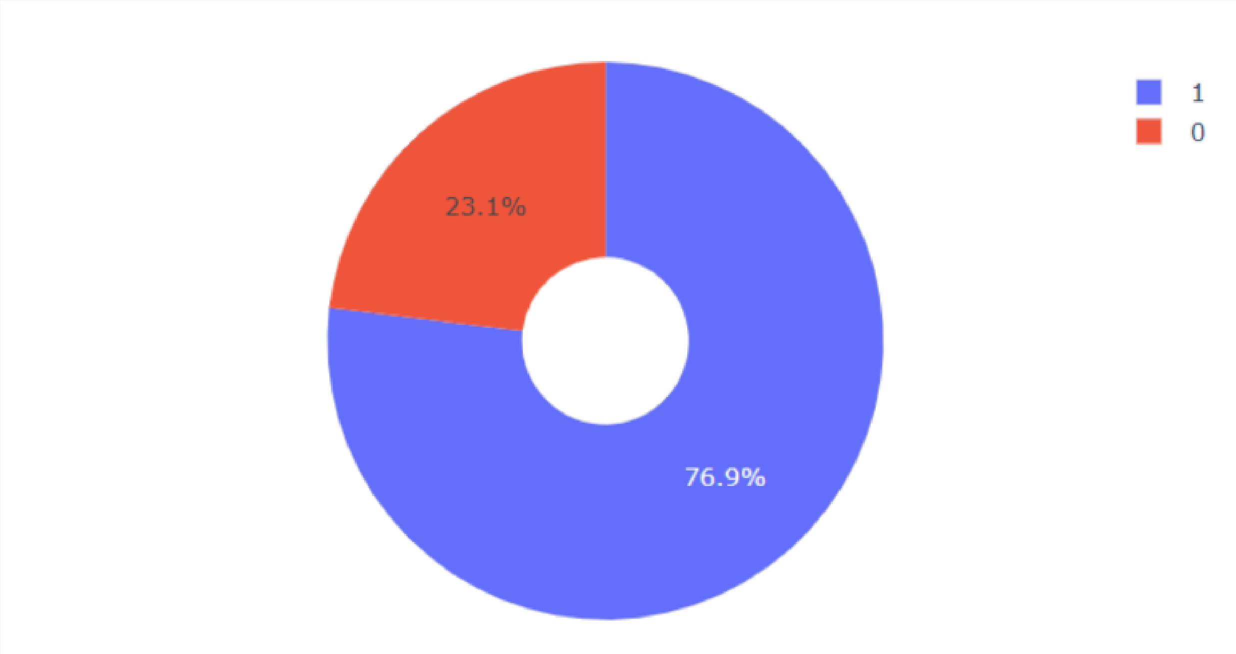
Build a Dashboard with Plotly Dash

Pie Chart 1



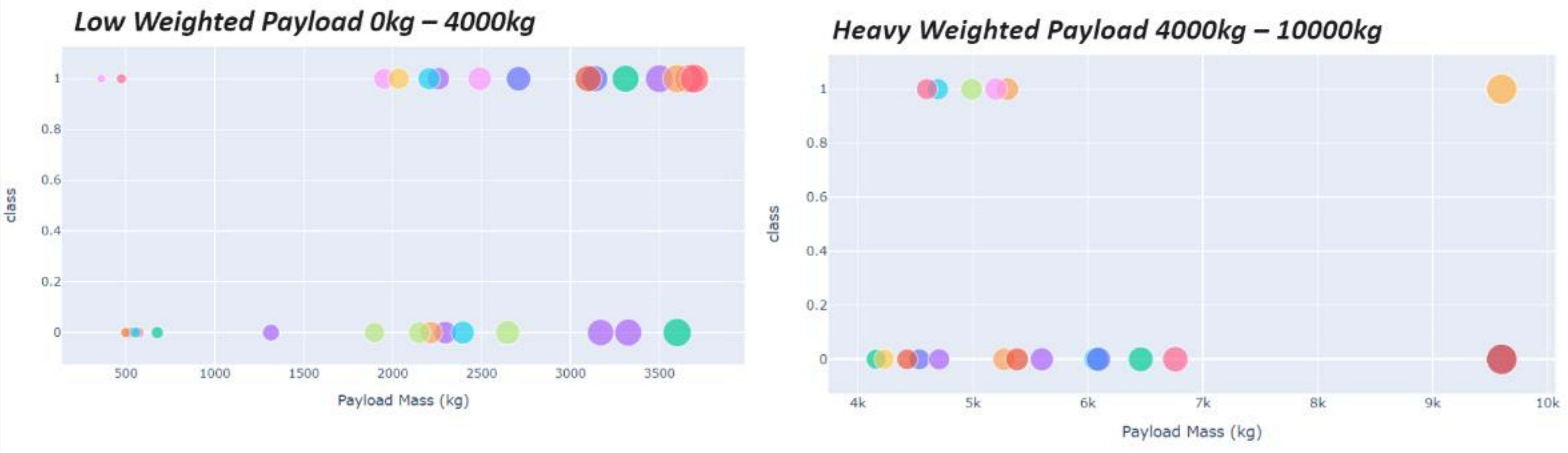
- KSC LC-39A is the lunching site with higher success rate.

Pie Chart 2



- KSC LC-39A has 76.9% success in launchings.

Pie Chart 3

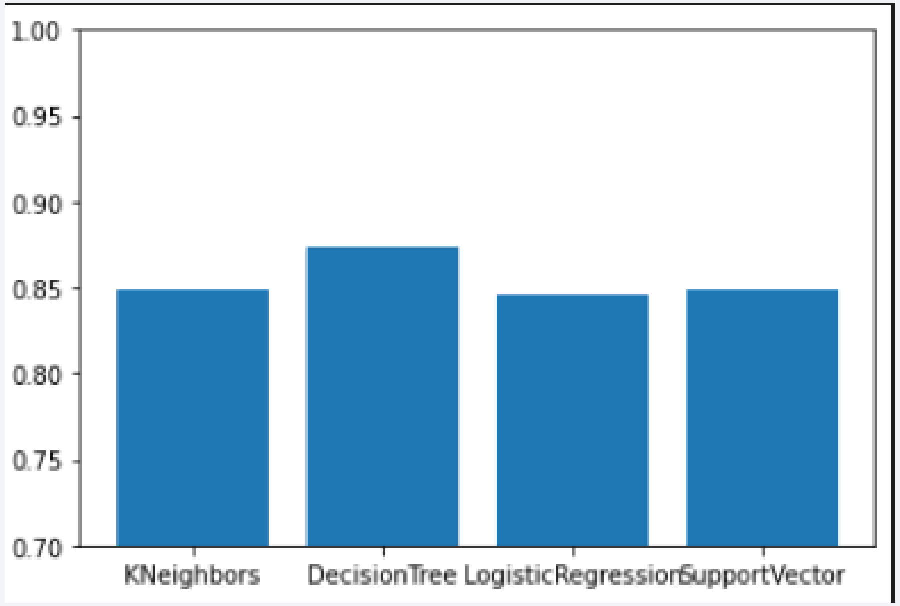


- The lower the payload mass, the higher the success rate.

Section 5

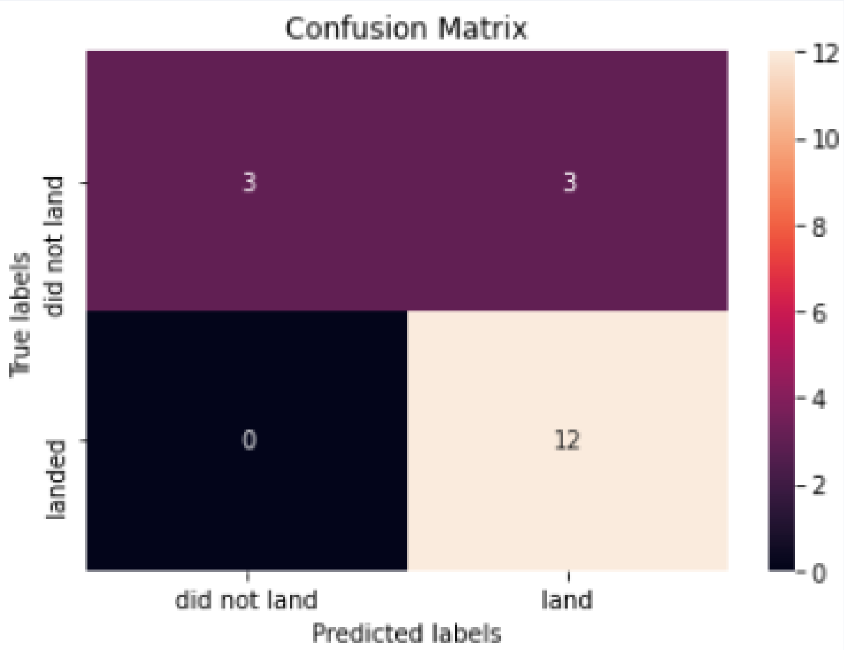
Predictive Analysis (Classification)

Classification Accuracy



- Decision tree is the most accurate model.

Confusion Matrix



- 12 true positive, 0 false negative, 3 true negatives and 3 false negatives.

Conclusions

- The larger the flight amount at a launch site, the greater the success rate at a launch site(related to next point).
- Launch success have been increasing since 2013
- KSC LC-39A had the most successful launches of any sites.
- The Decision tree classifier is the best model to predict the outcome of a lunch.

Appendix

- Include any relevant assets like Python code snippets, SQL queries, charts, Notebook outputs, or data sets that you may have created during this project

Thank you!