Winning Space Race with Data Science

Florian Woelzl
10.01.2022

Outline

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

Executive Summary

Summary of methodologies

- Data Understanding
- Data Preparation (Data collection / visualization)
- Model Building
- Model Evaluation

Summary of all results

- The best performing ML Method is Logistic Regression. Second is the Decision Tree.
- With increasing time and years the mean Payload Mass of the Rockets increased and the landing success rate increased.

- The KSC LC-39A Launch Site had the highest success rate with 76.9%

Introduction

Project background and context

The main aim of this project is to predicted if the Falcon 9 first stage will land successfully. 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.

Questions and Problems to be solved

One of the main focuses was to figure out what the factors are that determine a successful landing of a Falcon 9 Rocket. What parameters influence the rockets successful landing and what circumstances does SpaceX need to achieve this.

Section 1

Methodology

Methodology

Executive Summary

Data collection methodology

- Web scrapping from the Wikipedia Site
- Import Data using SpaceX Rest API

Perform data wrangling

- Select relevant Features / Drop missing Data / Create Dummy variables

Perform exploratory data analysis (EDA) using visualization and SQL

Perform interactive visual analytics using Folium and Plotly Dash

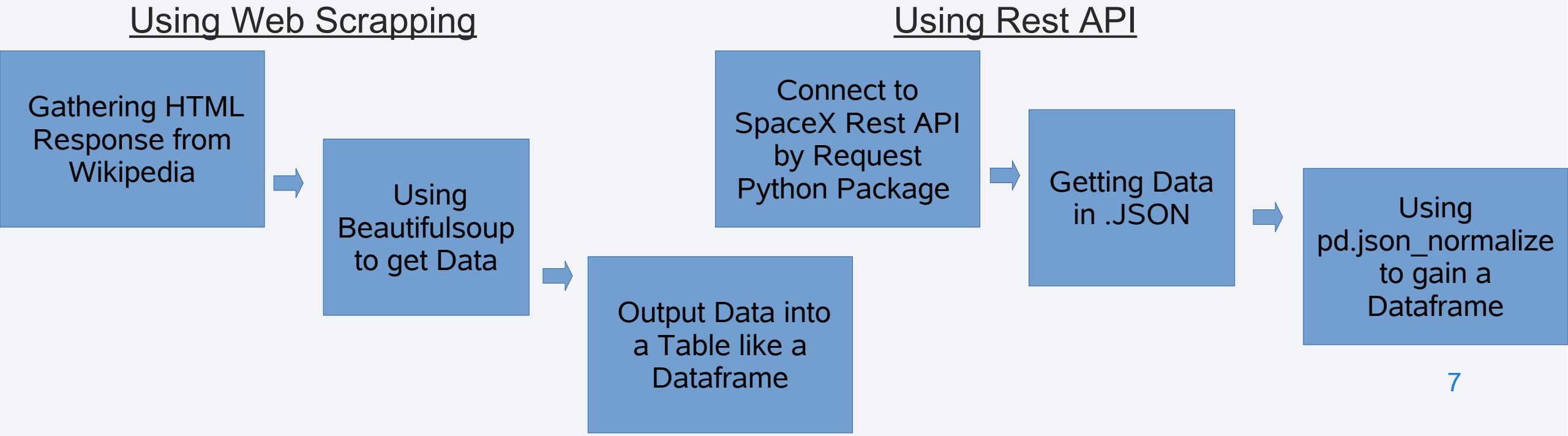
Perform predictive analysis using classification models

- Standard Scale Variables / Train Test Split / Grid search Hyper parameters
- Used ML Models: Logistic Regression, Support Vector Machine, Decision Trees, K Nearest Neighbors

Data Collection

The used Datasets were collected by:

- Web Scrapping of the Falcon9 Wikipedia Website using Beautifulsoup
  Gained Columns: Flight No, Payload, Payload Mass, Orbit, Customer
- By Using the SpaceX Rest API and the Request Python Package
  Gained Variables: Booster Version, Payload Mass, Orbit, Launch Site, Outcome



Data Collection – SpaceX API

In the linked Jupyternotebook a get reques was made to the SpaceX Rest API.

In addition some basic data wrangling and formatting was conducted.

1. Request rocket launch data from SpaceX API

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

2. Decode the response content as a Json turn it into a Pandas Dataframe

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

3. Use Custom Functions and convert list to Dataframe

```
# Create a data from launch_dict
data = pd.DataFrame(launch_dict)
data.head()
```

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

https://github.com/umbalito/IBM-Data-Science-Capstone/blob/ca66b08500f6e99ec51900e9a7bfe9caf88652c3/W1jupyter-labs-spacex-data-collection-api_V1.ipynb

Data Collection - Scraping

We conducted Web scarping to collect Falcon 9 historic launch Data from the Wikipedia page ‘List of Falcon 9 and Falcon Heavy launches’ using BeautifulSoup

- Extraction of a Falcon 9 launch data HTML table
- Parsing of the table and conversion into a Pandas Dataframe

1.Use request.get() Method

```
# assign the response to a object
response = requests.get(static_url)
```

2.Create BeautifulSoup() Object

```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(response.text,'html.parser')
```

3. Find Tables

```
# Assign the result to a list called `html_tables`
html_tables = soup.find_all('table')
```

4. Extract Column Names

```
for flt in flt_list:
    name = extract_column_from_header(flt)
    if name is not None and len(name) > 0:
        column_names.append(name)
```

5.Creating a Dictionary

6.Parse Soup Object

7.Import Dictionary to Dataframe

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

4. Save Dataframe to .CSV

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

https://github.com/umbalito/IBM-Data-Science-Capstone/blob/29a678bb50a733e4bde82b2cbca2cc0a3022958a/W1jupyter-labs-webscraping_V1.ipynb

Data Wrangling

First we find some patterns in the data to determine what would be the label for training supervised models. We identified missing values and defined the set of values which mean a bad langing outcome.
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. In this lab we will mainly convert those outcomes into Training Labels with `1` means the booster successfully landed `0` means it was unsuccessful.

1.Identify missing values

```
df.isnull().sum()/df.count()*100
```

2. Define set of no landing outcomes

```
bad_outcomes=set(landing_outcomes.keys()[[1,3,5,6,7]])
bad_outcomes
```

3. Aggregate to Landing_class

```
landing_class = []
for Outc in df['Outcome']:
    if Outc in bad_outcomes:
        landing_class.append(0)
    else:
        landing_class.append(1)
```

4. Merge and Save to CVS

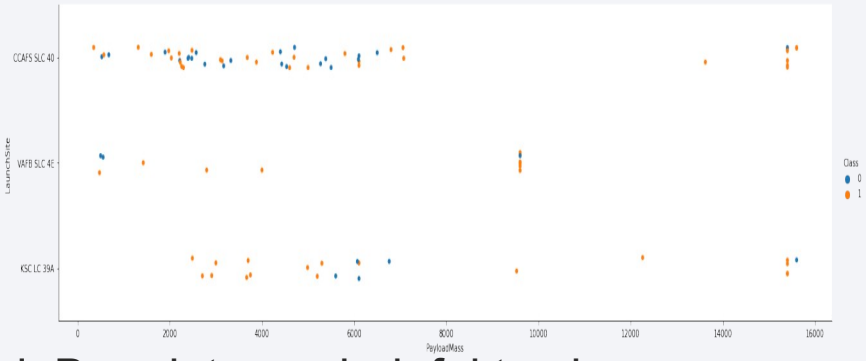
```
df['Class']=landing_class
df.to_csv("dataset_part__2.csv", index=False)
```

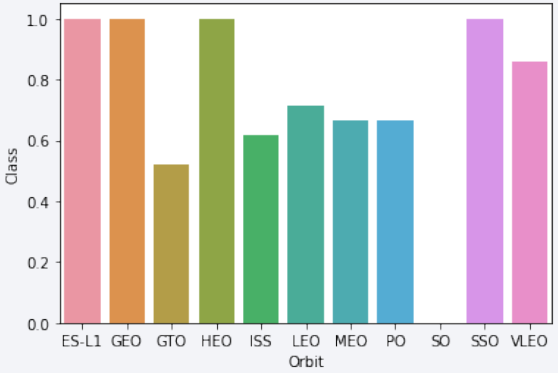
https://github.com/umbalito/IBM-Data-Science-Capstone/blob/29a678bb50a733e4bde82b2cbca2cc0a3022958a/W1labs-jupyter-spacex-Data%20wrangling_V1.ipynb

EDA with Data Visualization

We performed some Exploratory Data Analysis (EDA) and Feature Engineering using `Pandas` and `Matplotlib`.

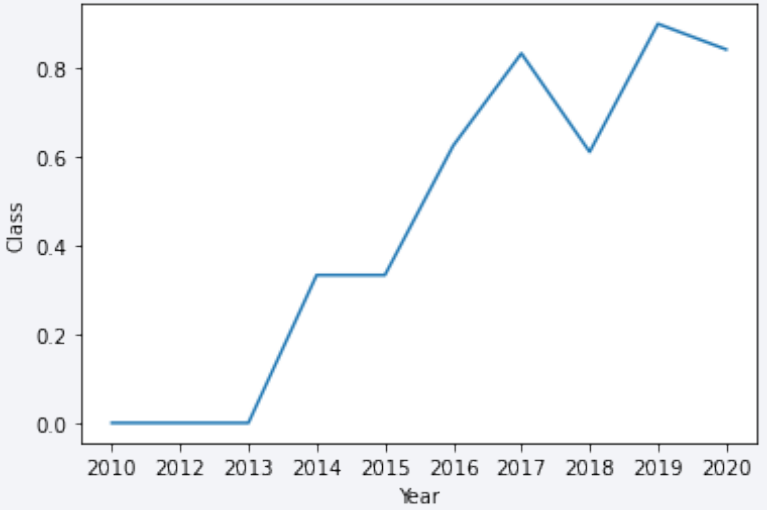
Scatter plots were used for Flight Number VS. Payload Mass, Flight Number VS. Launch Site, Payload VS. Launch Site, Flight Number VS. Orbit, Payload VS. Orbit Type. The reason to use scatter plots is that they show well the dependency between continuous variables.





The bar plot success rate over orbit type was plotted. Bar plots are helpful to show categorical dependencies and to show easily difference between groups.

The success rate over time was plotted as a line chart. Line Charts are handy to plot continuous variables over time. Its possible to see trends in line charts.



https://github.com/umbalito/IBM-Data-Science-Capstone/blob/29a678bb50a733e4bde82b2cbca2cc0a3022958a/W2bjupyter-labs-eda-dataviz_V1.ipynb

EDA with SQL

https://github.com/umbalito/IBM-Data-Science-Capstone/blob/29a678bb50a733e4bde82b2cbca2cc0a3022958a/W2ajupyter-labs-eda-sql-coursera_V1.ipynb

List of SQL queries which were performed:

- Displayed the names of the unique launch sites in the space mission
- Displayed 5 records where launch sites begin with the string 'KSC'
- Displayed the total payload mass carried by boosters launched by NASA (CRS)
- Displayed average payload mass carried by booster version F9 v1.1
- Listed the date where the successful landing outcome in drone ship was achieved.
- Listed the names of the boosters which have success in ground pad and have payload mass greater than 4000 but less than 6000
- Listed the total number of successful and failure mission outcomes
- Listed the names of the booster_versions which have carried the maximum payload mass.
- Listed the records which will display the month names, successful landing_outcomes in ground pad ,booster versions, launch_site for the months in year 2015
- Ranked the count of successful landing_outcomes between the date 2010-06-04 and 2017-03-20 in descending order

Build an Interactive Map with Folium

The Latitude and Longitude Coordinates from the the Launch Dataset were used to visualize the launch sites on an interactive map. A Circle Marker was added around each launch site with a label of the name of the launch site to better identify the Launchsite.

Green and Red markers were added to the map by assigning the dataframe launch_outcomes(failures, successes) to classes 0 and 1 with MarkerCluster(). The reason is to get an idea of the success rate of the different sites.

The distance to different launch sites were calculated using a distancemetric.

To get an idea what is around the launch site, lines are drawn on the map and the distance to landmarks were calculated.

https://github.com/umbalito/IBM-Data-Science-Capstone/blob/29a678bb50a733e4bde82b2cbca2cc0a3022958a/W31lab_jupyter_launch_site_location_V1.ipynb

Build a Dashboard with Plotly Dash

In order to build the Dashboard Flask and Dash was used.

1) a Launch Site Drop-down Input Component was added

It was added to enable the option to select and view the Graphs depending on the launch site.

2) a callback function to render success-pie-chart based on selected site drop down was added

It was added to get a visual impression on the success rate of the launch sites.

3) a Range Slider to Select Payload was added

It was added to enable the option to select and view the Graphs depending on the Payload.

4) a callback function to render the success-payload-scatter-chart scatter plot was added

It was added to get insights on the success rate of Payload mass relation.

https://github.com/umbalito/IBM-Data-Science-Capstone/blob/29a678bb50a733e4bde82b2cbca2cc0a3022958a/W32_labspacex_dash_app.py

Predictive Analysis (Classification)

1. Model Building
- Load our dataset into NumPy and Pandas
- Transform Data
- Split our data into training and test data sets
- Decide which type of machine learning algorithms we want to use
- Set our parameters and use GridSearchCV
- train our Model by using the GridSearchCV.

2. Evaluating Model
- Check accuracy for each model
- Plot Confusion Matrix

3. Improving Model
- Feature Engineering
- Finding better Hyperparameters

4. Finding The Best Performing Classification Model
- The model with the best accuracy score found by the Grid Search is the best performing model

https://github.com/umbalito/IBM-Data-Science-Capstone/blob/29a678bb50a733e4bde82b2cbca2cc0a3022958a/W4SpaceX_Machine%20Learning%20Prediction_Part_5_V1.ipynb

Results

- Exploratory data analysis results

  With time the success rate of the rocket landings increases.
  The Launch site KSC LS-39A has the highest success rate.

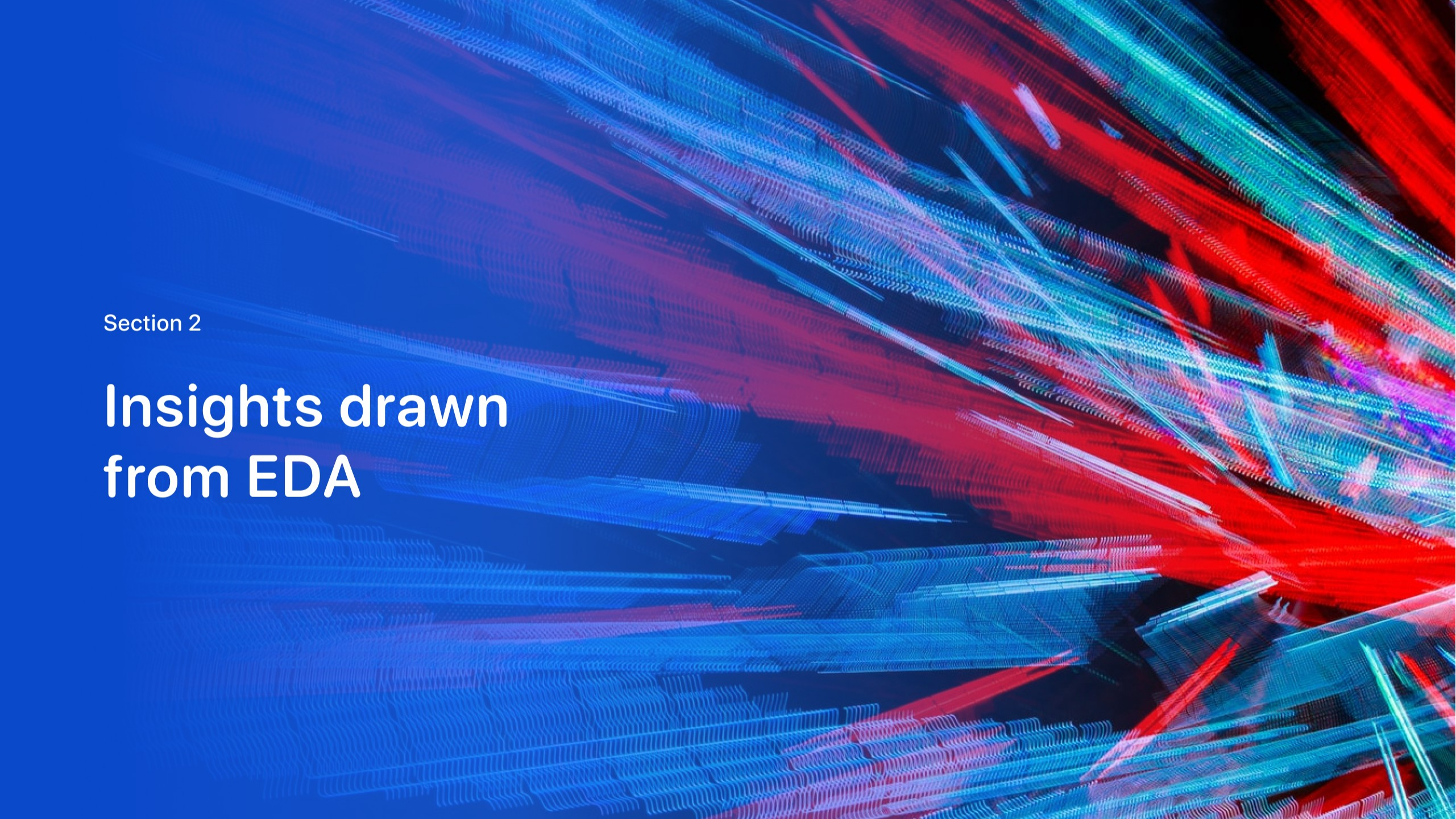
- Interactive analytics demo in screenshots

- Predictive analysis results

  The logistic Regression with an accuracy of 0.83 performs best

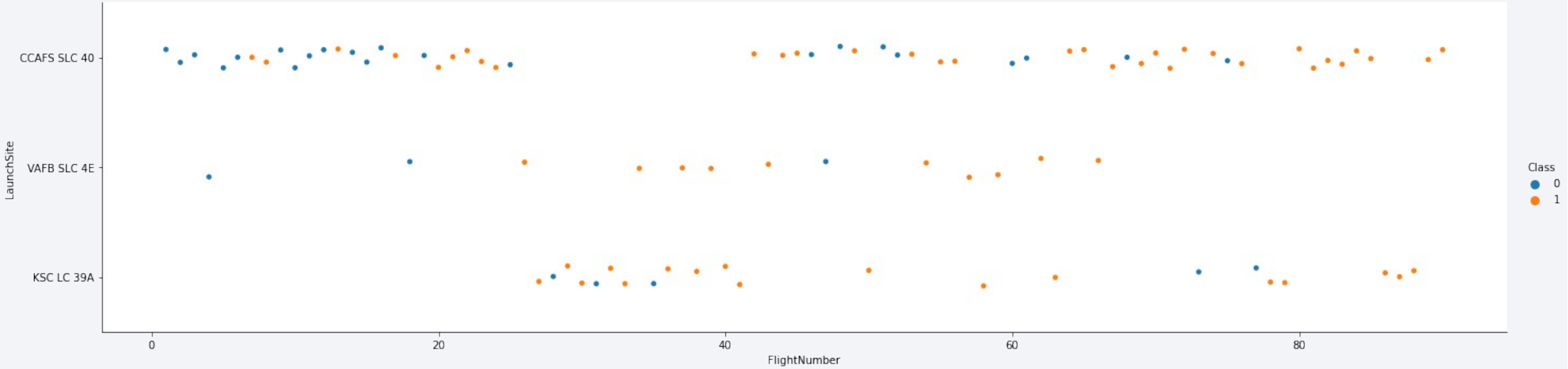
Section 2

Insights drawn from EDA



Flight Number vs. Launch Site

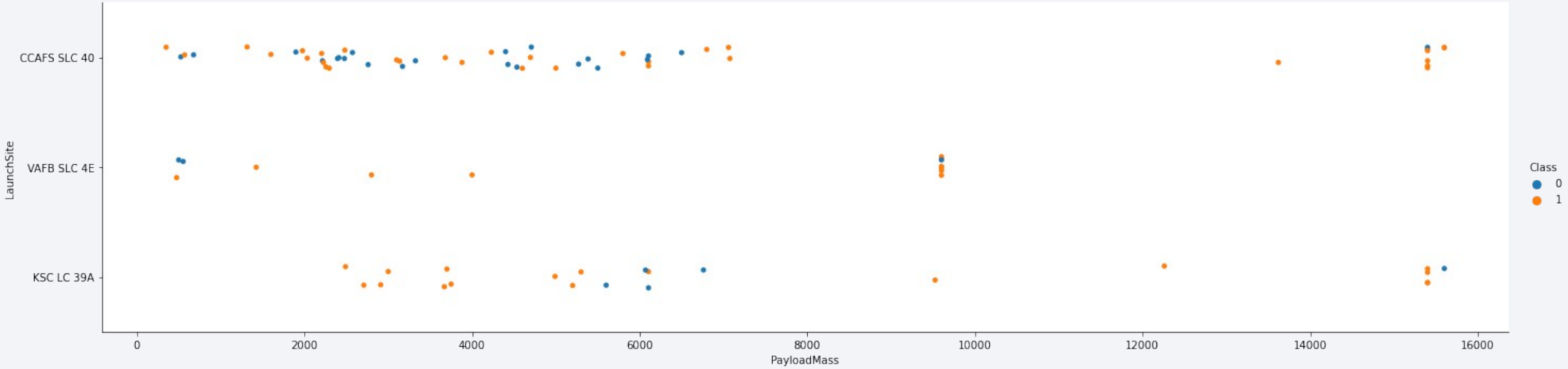
Scatter plot of Flight Number vs. Launch Site



We see that with increasing Flight number the Launch site CCAFD was used.

Payload vs. Launch Site

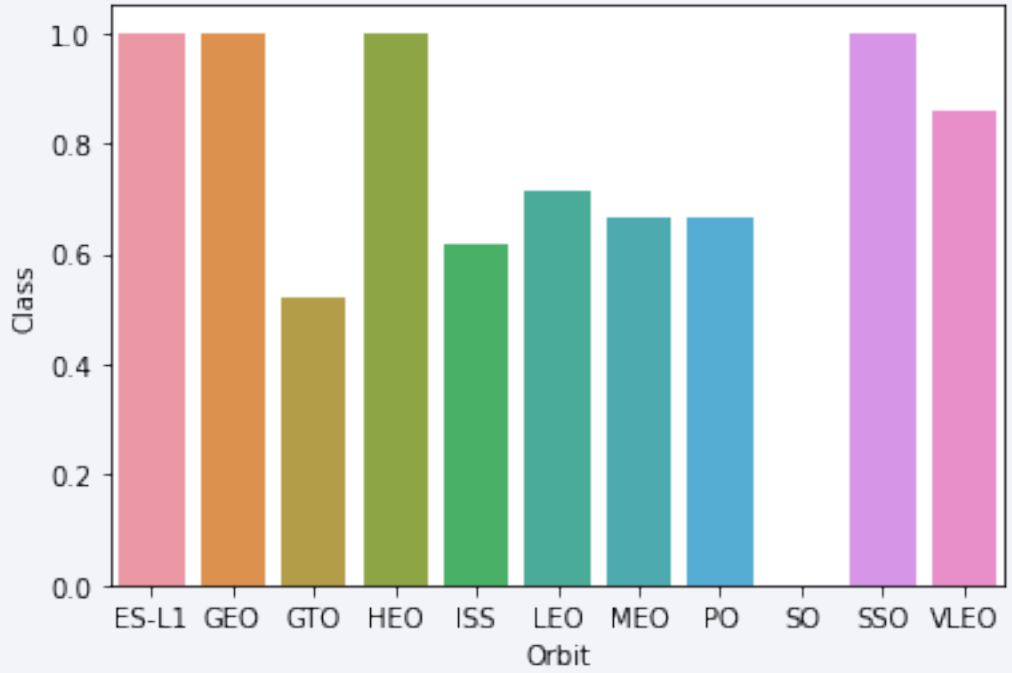
Scatter plot of Payload vs. Launch Site



We see that at high Payload masses the CCAFS and KSC Launch Site were used.

Success Rate vs. Orbit Type

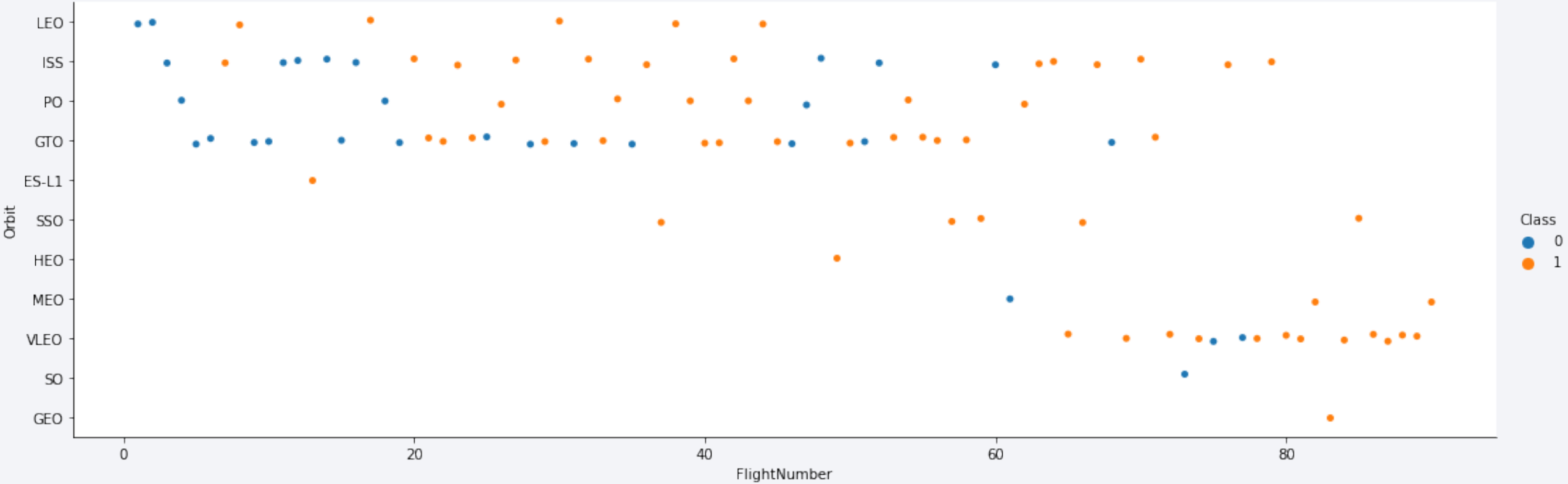
Bar chart of Success Rate vs. Orbit Type



We see that the Orbit ES-L1, GEO,HEO and SSO have a high success rate.

Flight Number vs. Orbit Type

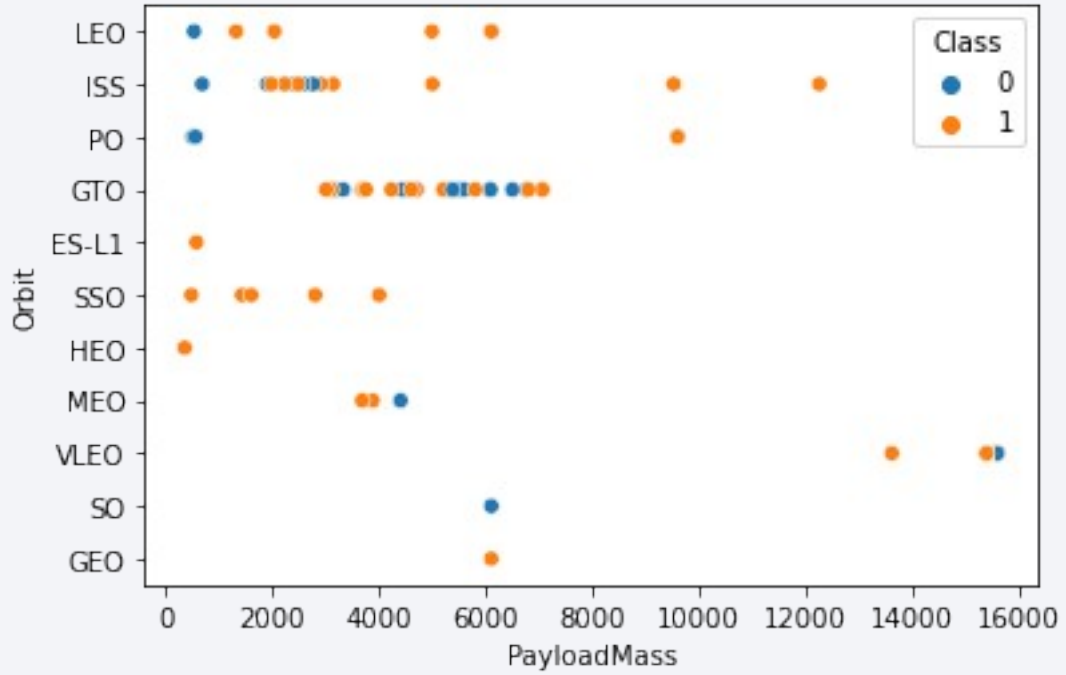
Scatter plot of Payload vs. Launch Site



We see that only at high Flight numbers GEO, SO and VLEO Obits were used.

Payload vs. Orbit Type

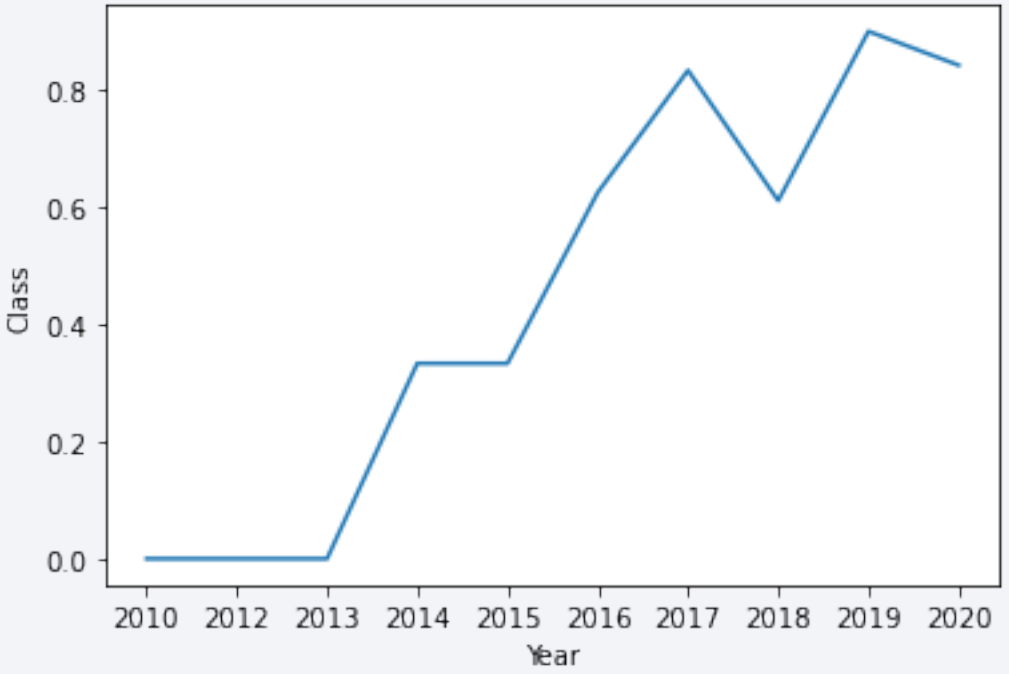
Scatter plot of Payload Mass vs. Orbit Type



We see that at high Payload masses only the VLEO Orbit was used.

Launch Success Yearly Trend

Line plot of Yearly Sucess Rate vs. Time in Years



We see that the Success Rate increases with the Years.

All Launch Site Names

Find the names of the unique launch sites

Query: SELECT DISTINCT Launch_Site FROM SPACEX

Output:

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

Explanation: The Distinct clause leads to an output of only distinct Launch Sites

Launch Site Names Begin with 'CCA'

Find 5 records where launch sites begin with `CCA`

Query: SELECT * FROM SPACEX where Launch_Site LIKE 'CCA%'
Output:

	index	Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
0	0	04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
1	1	08-12-2010	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	2	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
3	3	08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
4	4	01-03-2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Explanation: The LIKE and the ‘%’ leads to an query where the where clause condition matches all entries with CCA and additional letters

Total Payload Mass

Calculate the total payload carried by boosters from NASA

Query: SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEX where
Customer = 'NASA (CRS)'

Output:

	SUM(PAYLOAD_MASS__KG_)
0	45596

Explanation: The SUM() leads to an aggregation of the whole column by summarizing the entries

Average Payload Mass by F9 v1.1

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

Query: SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEX where Booster_Version LIKE 'F9 v1.1%'
Output:

	AVG(PAYLOAD_MASS__KG_)
0	2534.666667

Explanation: The AVG() leads to an aggregation of the whole column by averaging the entries

First Successful Ground Landing Date

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

Query: SELECT MIN(DATE) FROM SPACEX where Landing_Outcome = 'Success (ground pad)'

Output:

	MIN(DATE)
0	01-05-2017

Explanation: The MIN() condition leads to the output where the minimal date value with a valid where condition is shown.

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

Query: SELECT Booster_Version FROM SPACEX where Landing_Outcome = 'Success (drone ship)' and PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000
Output:

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

Explanation: The Payload 4000 AND 6000 condition leads to the output where the Payload is between 4000 and 6000.

Total Number of Successful and Failure Mission Outcomes

Calculate the total number of successful and failure mission outcomes

Query: SELECT Landing_Outcome, Count() FROM SPACEX Group by Landing_Outcome

Output:

	Landing_Outcome	Count()
0	Controlled (ocean)	5
1	Failure	3
2	Failure (drone ship)	5
3	Failure (parachute)	2
4	No attempt	21
5	No attempt	1
6	Precluded (drone ship)	1
7	Success	38
8	Success (drone ship)	14
9	Success (ground pad)	9
10	Uncontrolled (ocean)	2

Explanation: The Group by clause Groups the output by Landing_outcome

Boosters Carried Maximum Payload

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

Query: SELECT Booster_Version FROM SPACEX where PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEX)

Output:

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

Explanation: By nesting a second Sql query the Max Payload Mass was found first

2015 Launch Records

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

Query: SELECT Landing_Outcome, Booster_Version, Launch_Site FROM SPACEX where Landing_Outcome Like'%drone%' and DATE Like '%2015'
Output:

	Landing_Outcome	Booster_Version	Launch_Site
0	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
1	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40
2	Precluded (drone ship)	F9 v1.1 B1018	CCAFS LC-40

Explanation: The LIKE and the ‘%’ leads to an query where the where clause condition matches all entries with drone and 2015 and additional letters.

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

Query: SELECT Landing_Outcome, Count() AS Cnt FROM SPACEX where DATE between 2010-06-04 and 2017-03-20 Group by Landing_Outcome Order by Cnt DESC

Output:

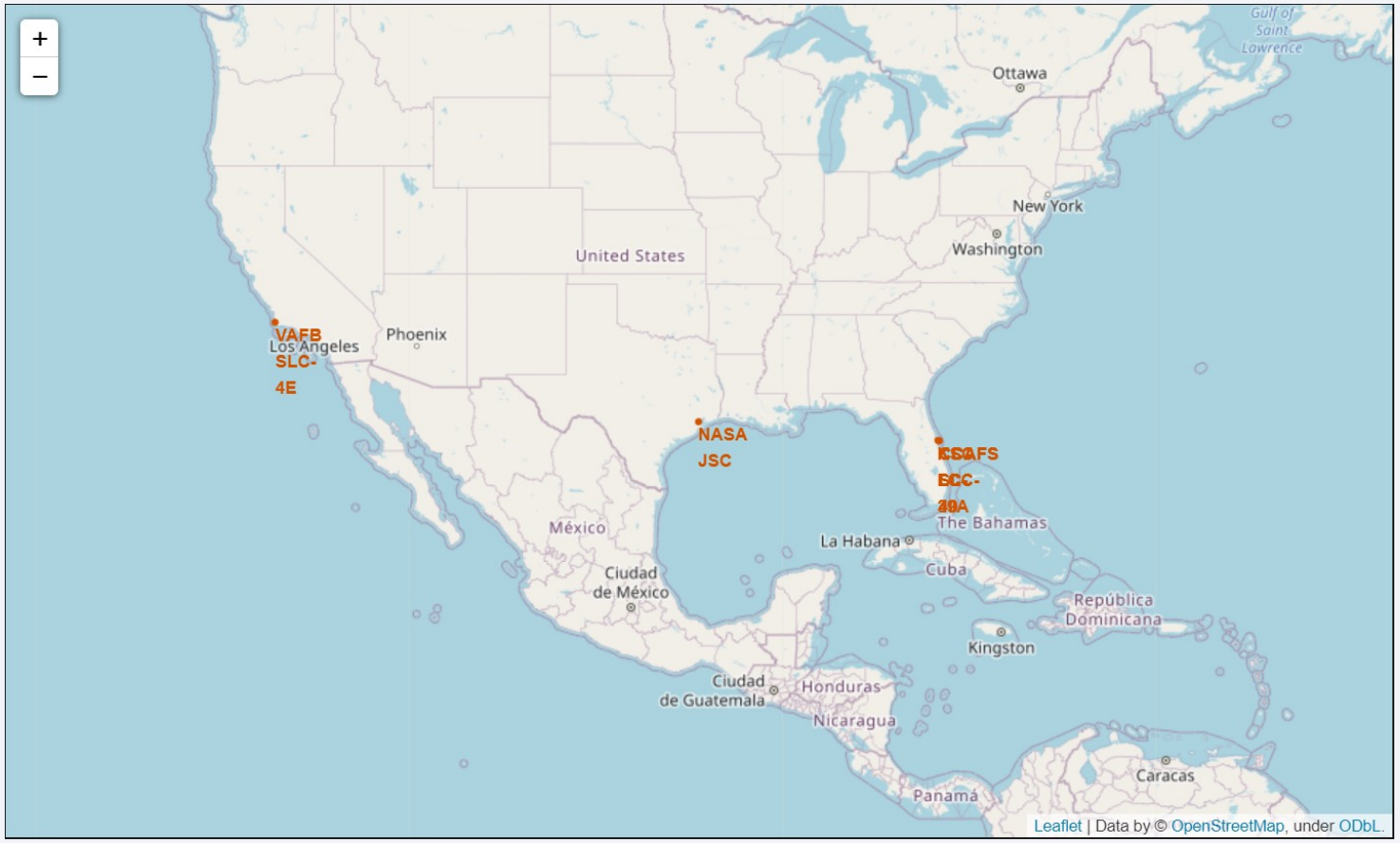
	Landing_Outcome	Cnt
0	Success	38
1	No attempt	21
2	Success (drone ship)	14
3	Success (ground pad)	9
4	Failure (drone ship)	5
5	Controlled (ocean)	5
6	Failure	3
7	Uncontrolled (ocean)	2
8	Failure (parachute)	2
9	Precluded (drone ship)	1
10	No attempt	1

Explanation: The Group by clause and the count() agglomerates the output. The order by clause orders the output.

Section 4

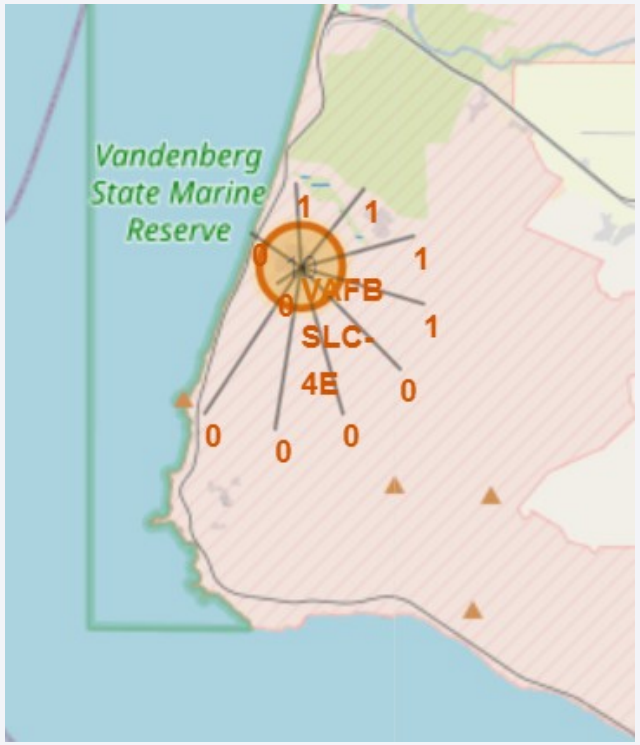
Launch Sites Proximities Analysis

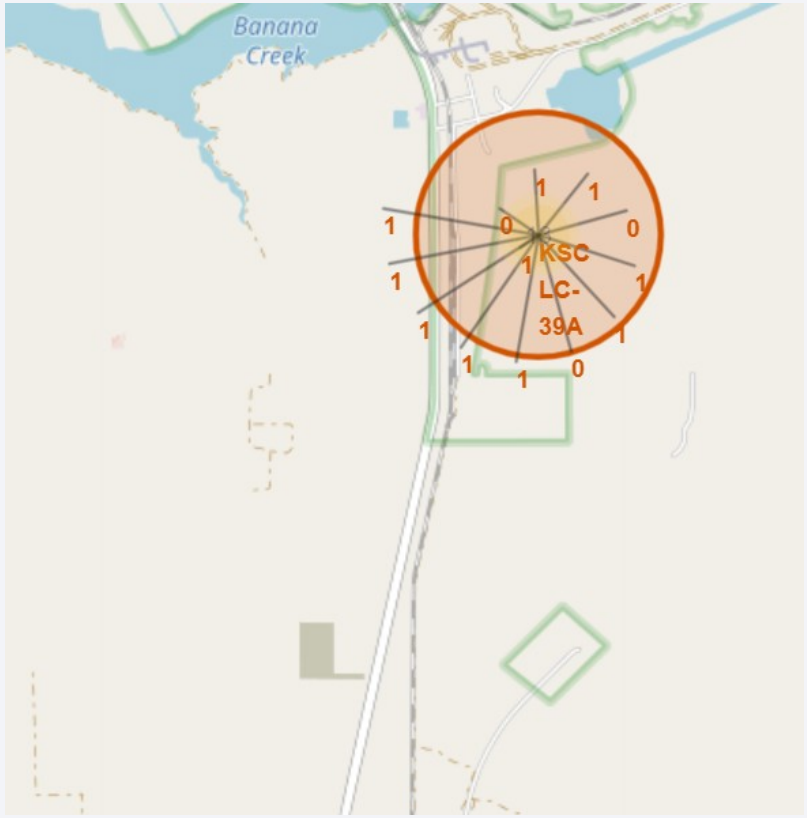
Map of all SpaceX Launch Sites

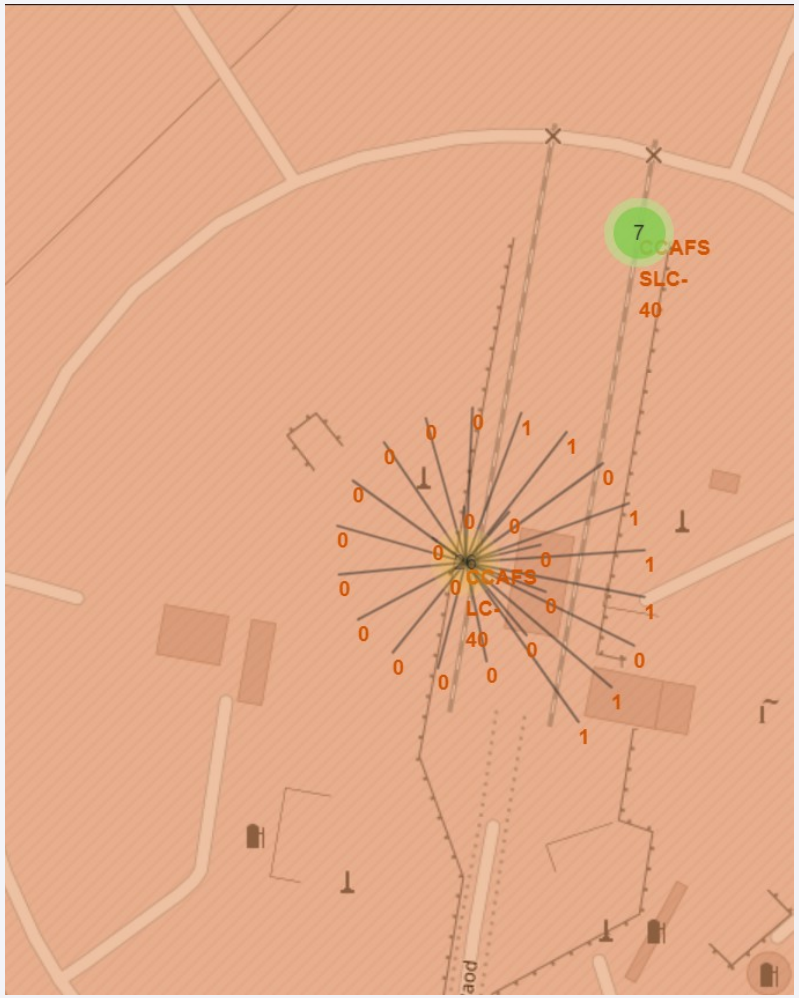


All SpaceX launch sites are in the United States. They are close to the coasts ether Atlantic or Pacific.

Launch outcomes on the map



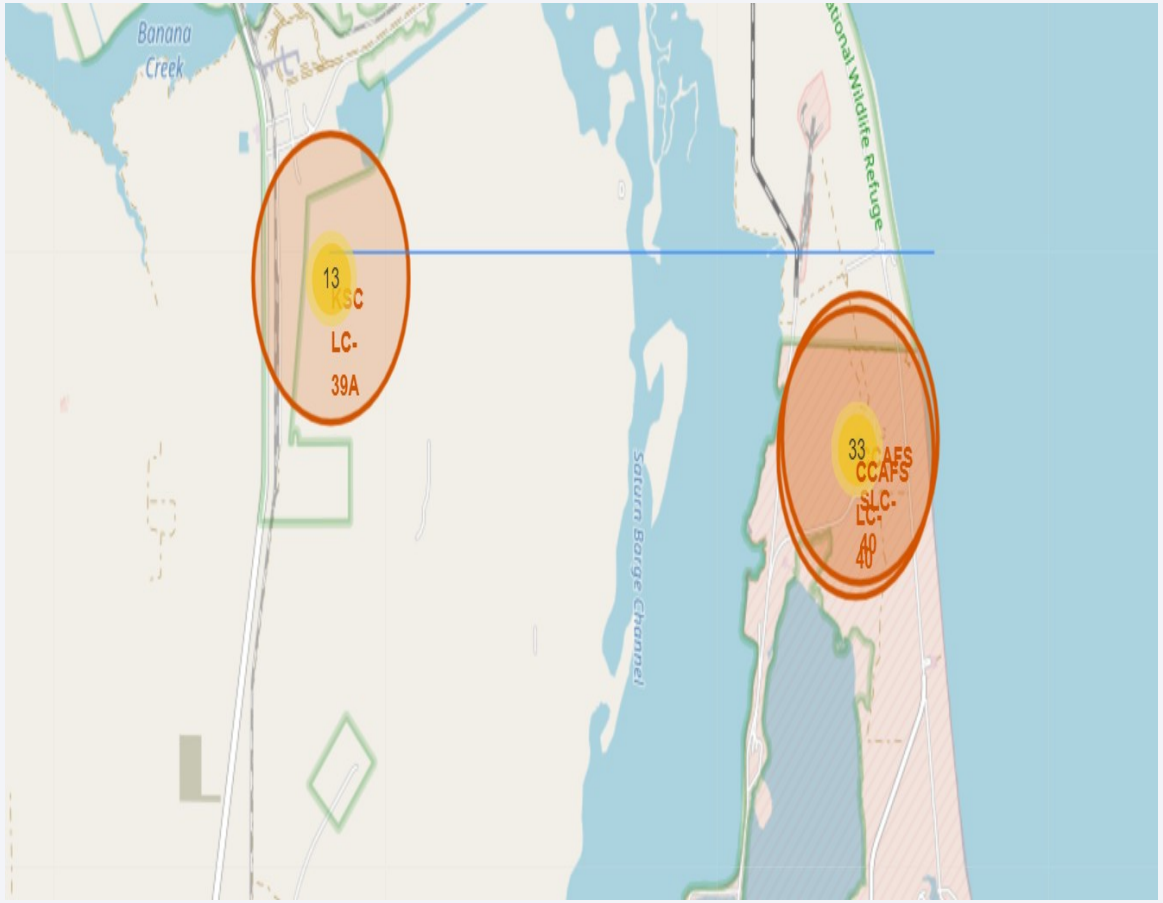




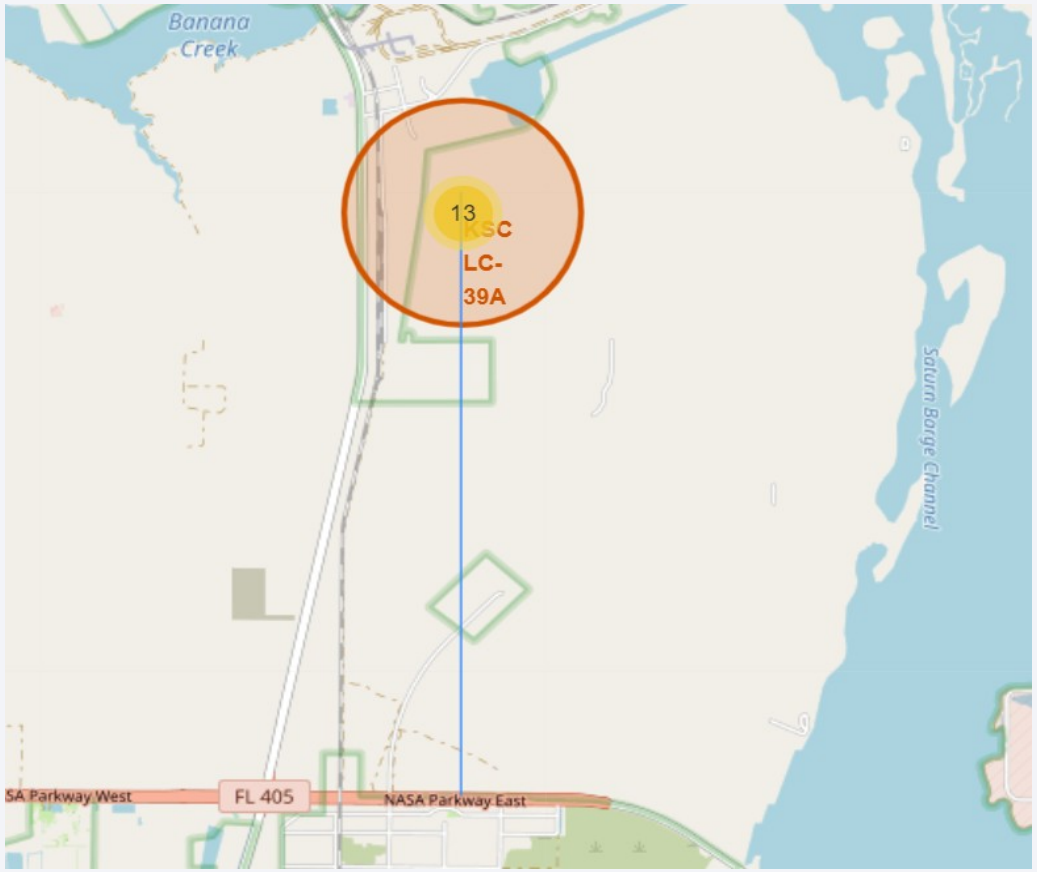
1 Shows successful landing outcomes 0 unsuccessful landing outcomes.

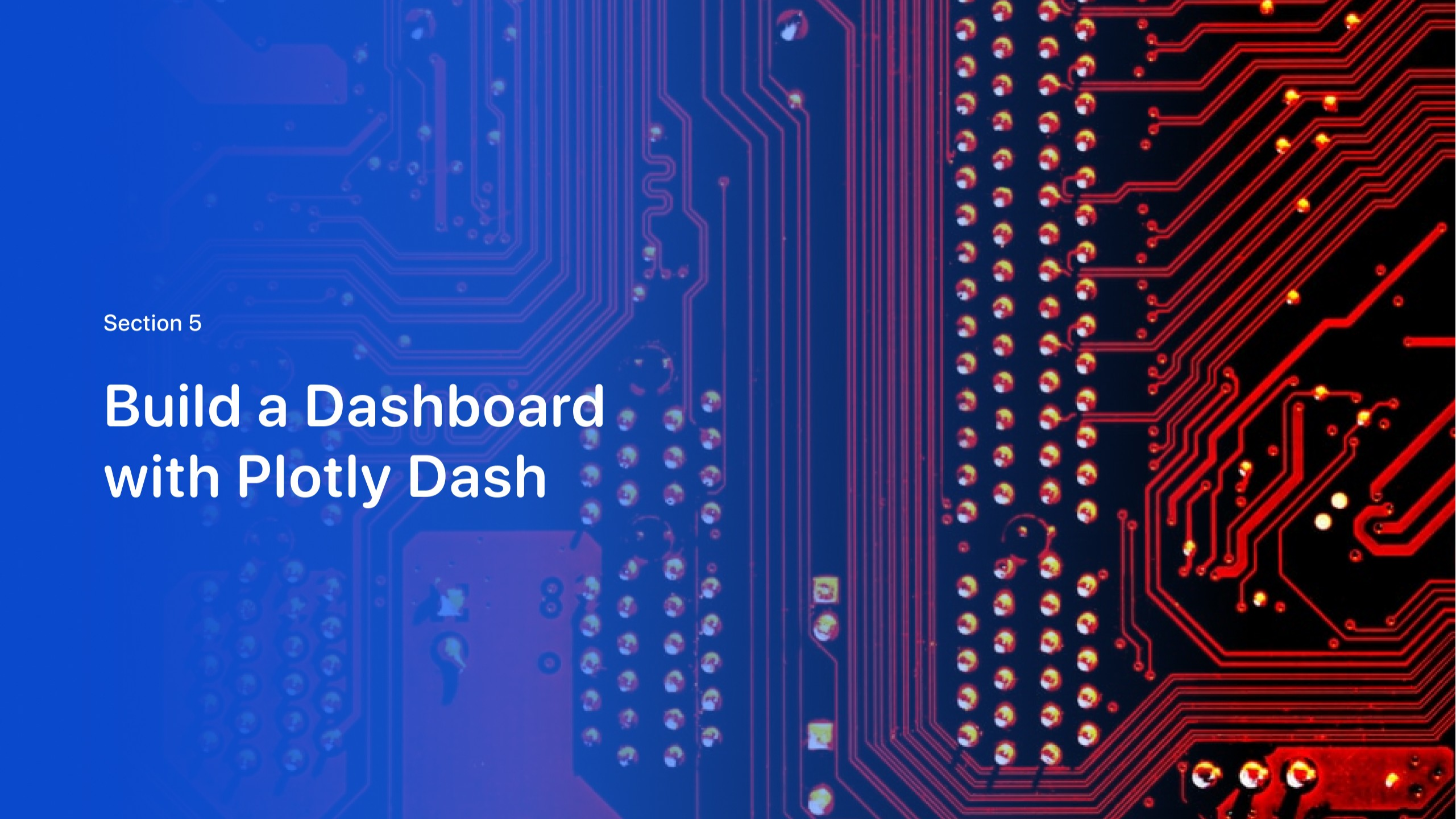
Distance to different landmarks

KSC Launch Site to Coast Line 6KM

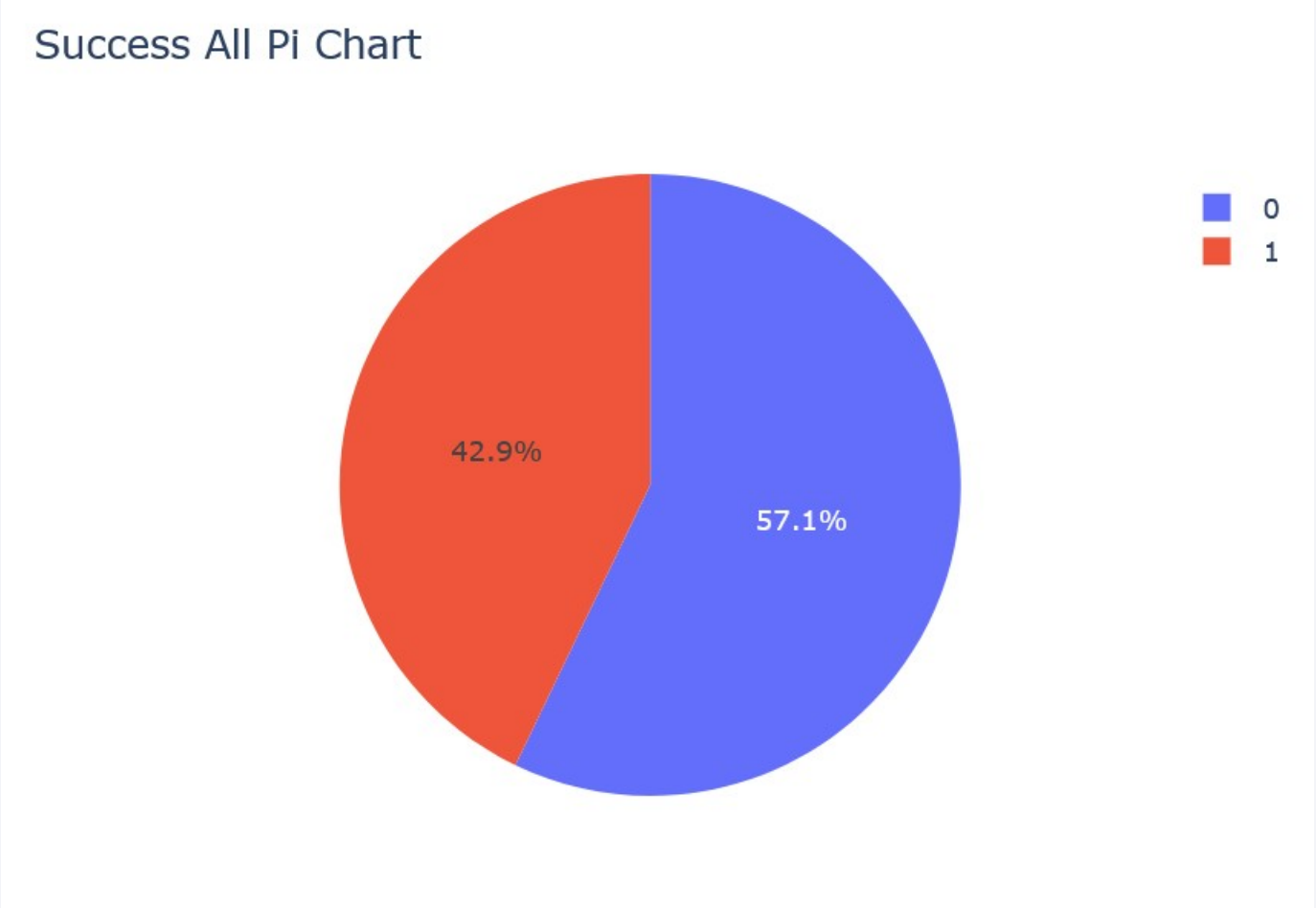


KSC Launch Site to Nasa Parkway East 5km



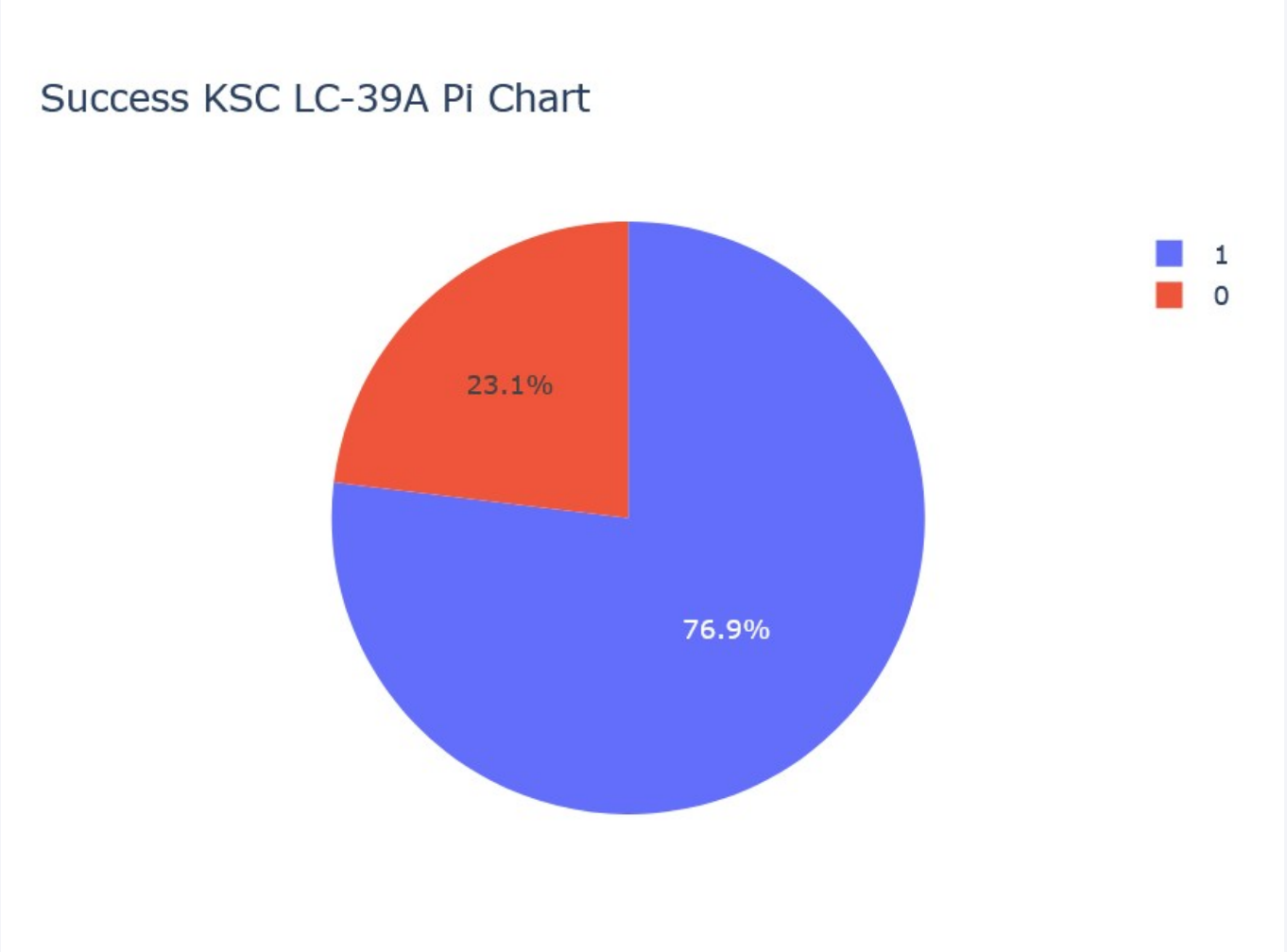

Section 5
Build a Dashboard with Plotly Dash

Launch success count for all sites



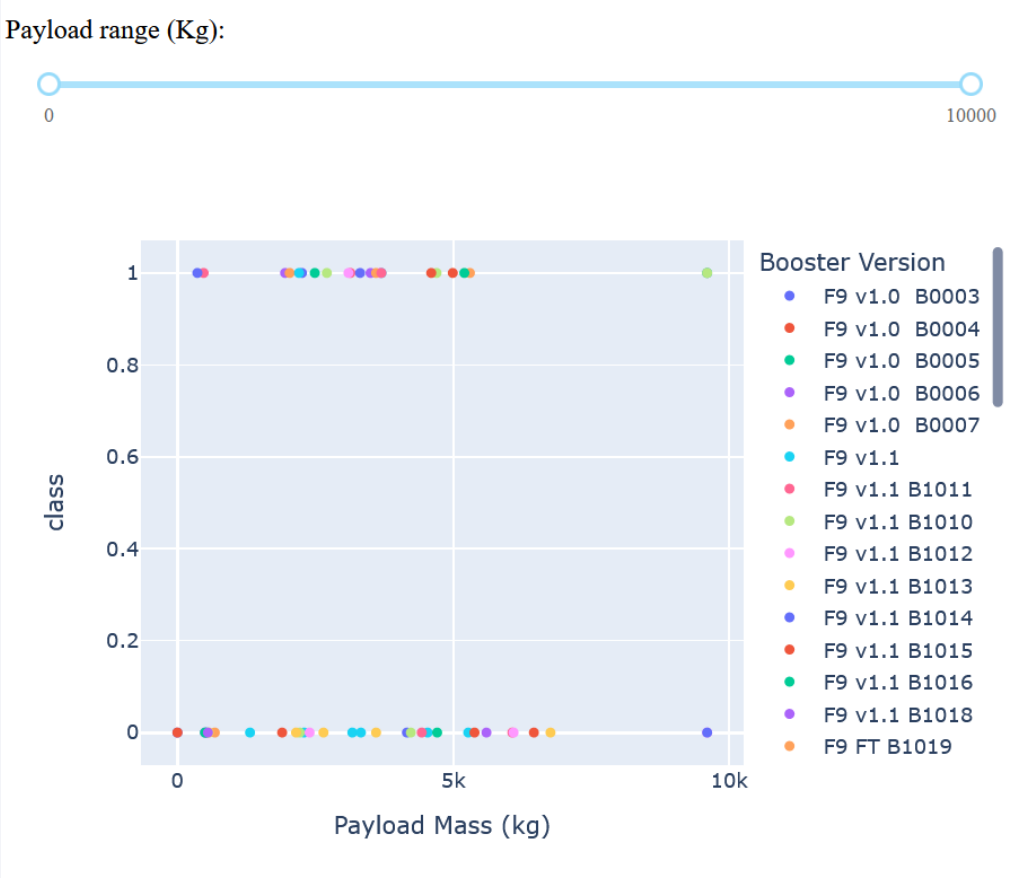
The overall Launch success is 42.9%

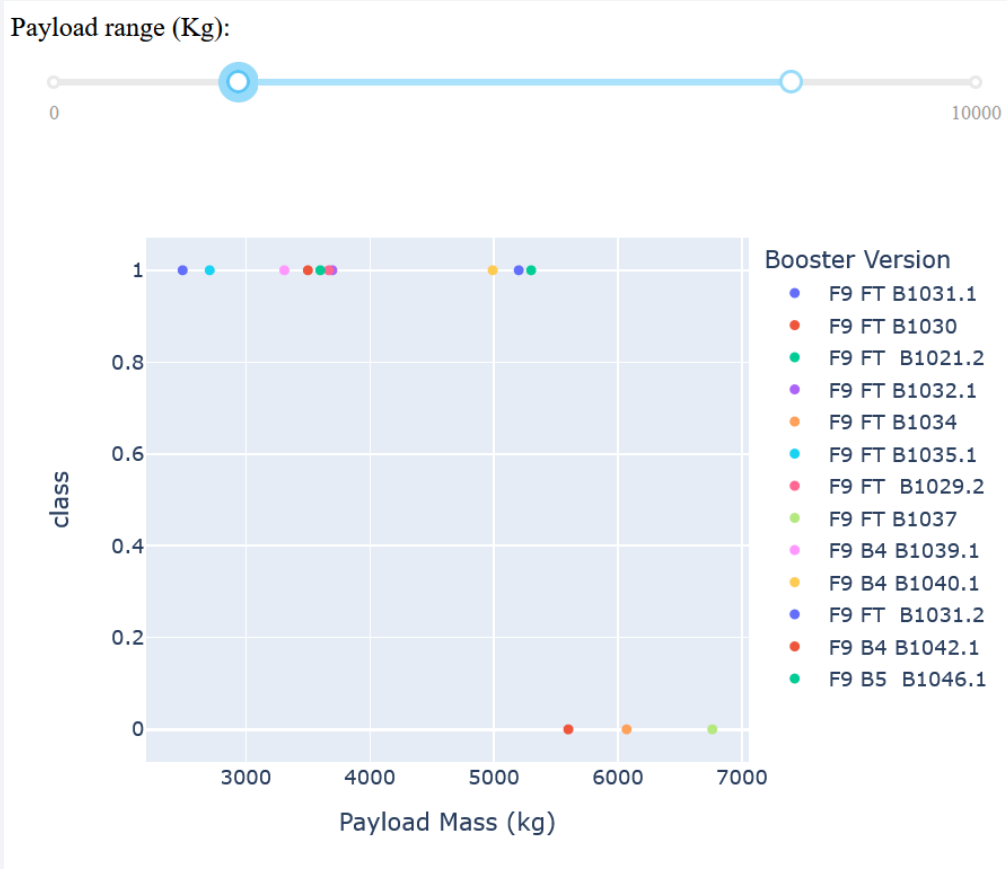
Pie chart of launch success for best launch site



The KSC LC-39A Launch Site had the highest success rate with 76.9%

Payload vs. Launch Outcome scatter plot for all sites



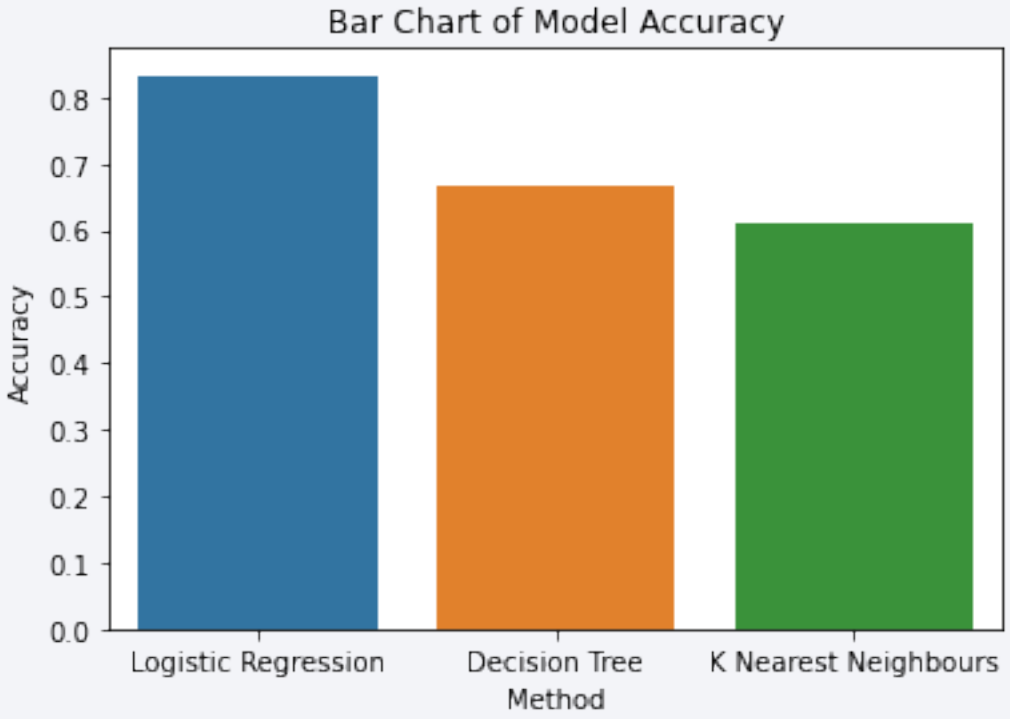


The Launches with low mass Payload have a higher success rate than the ones with heavy Payload

Section 6

Predictive Analysis (Classification)

Classification Accuracy



The logisitc Regression as a model has the highest classification accuracy

Confusion Matrix of Landing Prediction

The confusion matrix of the landing Prediction using Logistic Regression shows that we have three false positives meaning three rockets were predicted to land where in reality they did not land. We don’t have any false negatives meaning we predicted all true landings as landings.

Conclusions

The best performing Machine Learning Method out of the four applied is the Logistic Regression. Second best Method is the Decision Tree.

With increasing time and years the mean Payload Mass of the Rockets increased

With increasing time and years the landing success rate increased. This can be explained by overall improvements in the landing proceess.

The KSC LC-39A Launch Site had the highest success rate with 76.9%.

All SpaceX launch sites are in the United States. They are close to the coasts ether Atlantic or Pacific.

We see that at high Payload masses the CCAFS and KSC Launch Site was used.

Appendix

Investigation of the Spacex_launch_dash.cvs Table

```
[3]
1  spacex_df = pd.read_csv("spacex_launch_dash.csv")
2  np.unique(spacex_df['Launch Site'])
```

Python

```
array(['CCAFS LC-40', 'CCAFS SLC-40', 'KSC LC-39A', 'VAFB SLC-4E'],
      dtype=object)
```

```
[6]
1  spacex_df.head()
```

Python

	Unnamed: 0	Flight Number	Launch Site	Mission Outcome	class	Payload Mass (kg)	Booster Version	Booster Version Category
0	0	1	CCAFS LC-40	Success	0	0.0	F9 v1.0 B0003	v1.0
1	1	2	CCAFS LC-40	Success	0	0.0	F9 v1.0 B0004	v1.0
2	2	3	CCAFS LC-40	Success	0	525.0	F9 v1.0 B0005	v1.0
3	3	4	CCAFS LC-40	Success	0	500.0	F9 v1.0 B0006	v1.0
4	4	5	CCAFS LC-40	Success	0	677.0	F9 v1.0 B0007	v1.0

Thank you!