IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Camila Loiola Brito
Maia
22/10/2023

Outline

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

Executive Summary

- Used the following methodologies
  - Data Collection: API and Scraping
  - Data Wrangling
  - Exploratory SQL queries
  - Exploratory graphics
  - Interactive Maps
  - Interactive Dashboard
  - Predictive Analysis
- Identified launch sites with highest and lowest success rate
- Possible correlation between payload and launch success rate
- K Nearest Neighbours was the best model for prediction, with best accuracy

Introduction

- Falcon 9 rocket launches cost less because SpaceX is able to reuse its first stage

- Goal: determine the price of each launch
  - If we can determine if the first stage will land, we can determine the cost of a launch

Section 1

Methodology

Methodology

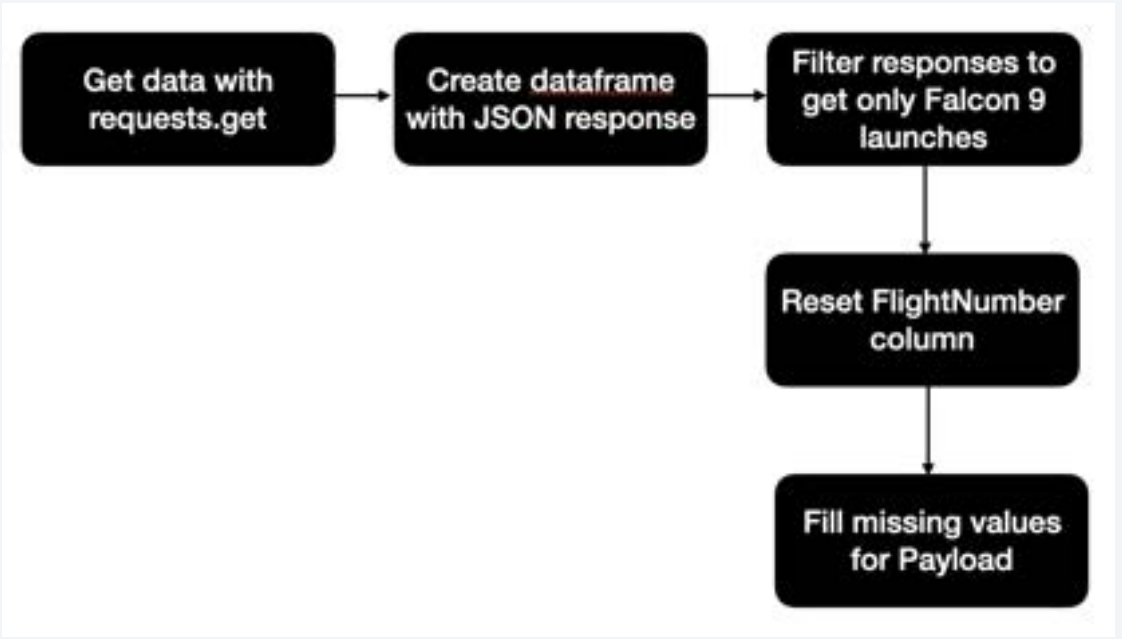
Executive Summary

- Data collection methodology:
  - Describe how data was collected
- Perform data wrangling
  - Describe how data was processed
- 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 – SpaceX API

- Used requests.get to get data from SpaceX API
- Decode the response in JSON and turned it in a dataframe using json.normalize()
- Filtered the dataframe to get only Falcon 9 launches
- Reset FlightNumber column

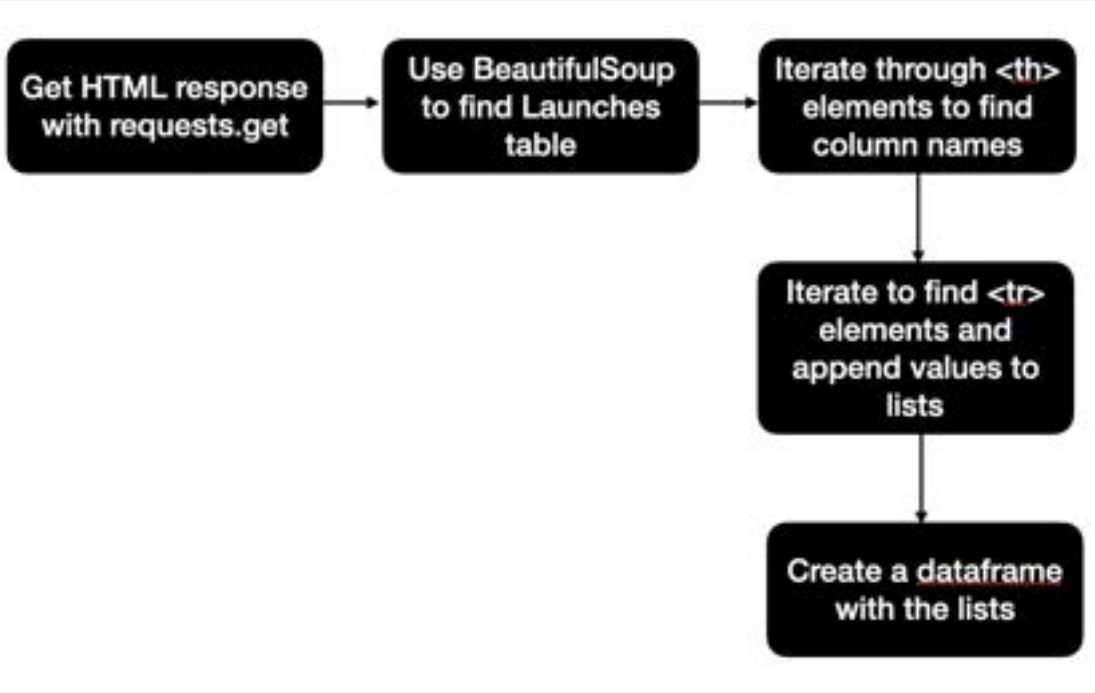
- GitHub URL:
  - https://github.com/camilalbmaia/data-science-notebooks/blob/main/Week%201%20-%201%20-%20Lab%201-%20Collecting%20the%20data.ipynb



Data Collection - Scraping

- Use requests.get to get HTML page as HTTP response
- Find Launches table (third table) using BeautifulSoup
- Iterate through <th> elements to find column names
- Iterated through <tr> elements (table lines) and appending values to lists (one list for each column)
- Created a dataframe with the lists

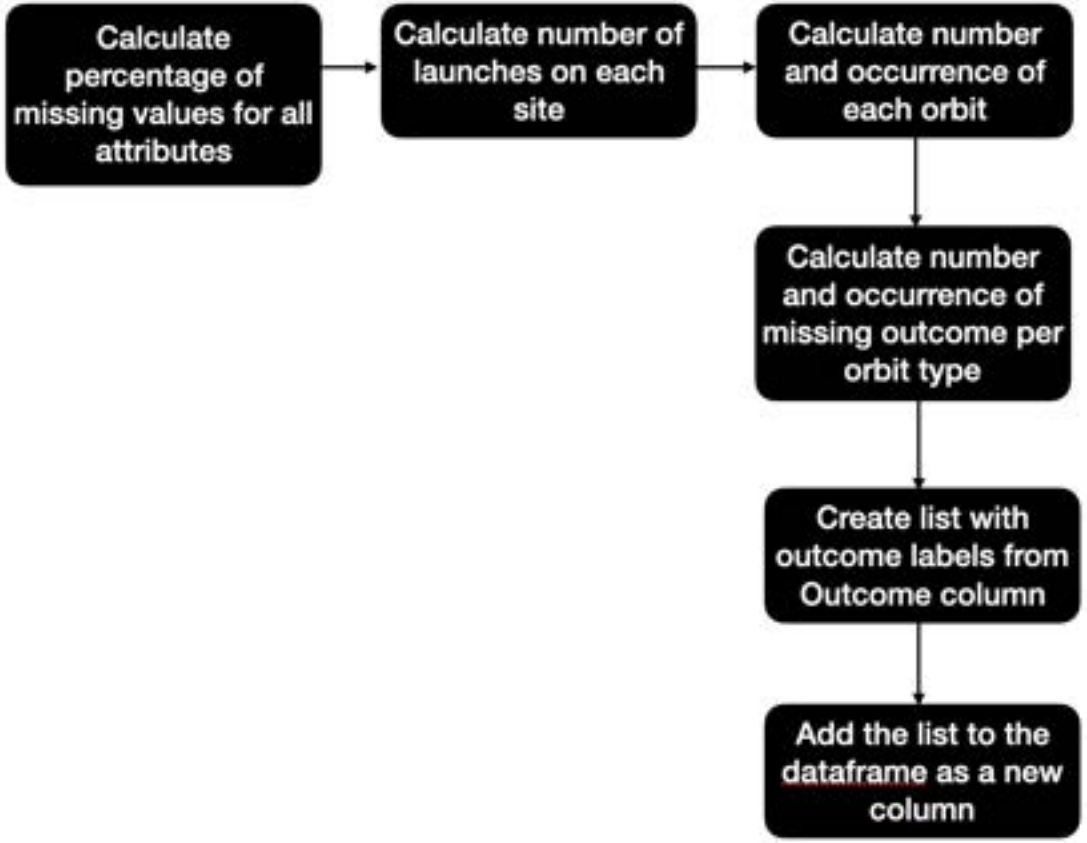
- Github URL:
  - https://github.com/camilalbmaia/data-science-notebooks/blob/main/Week%201%20-%202%20-%20Web%20Scraping.ipynb



Data Wrangling

- Calculate the percentage of missing values for all attributes using isnull()
- Use value_counts() to calculate
  - Number of launches on each site
  - Number and occurrence of each orbit
  - Number and occurrence of missing outcome per orbit type
- Create list (landing_class) with outcome labels (0 or 1) from Outcome column
- Add landing_class list to the dataframe as a new column (“Class")

- Github URL:
  - https://github.com/camilalbmaia/data-science-notebooks/blob/main/Week%201%20-%203%20-%20Lab%202-%20Data%20wrangling.ipynb



EDA with Data Visualization

- Scatter Plot (CatPlot)
  - Investigate how the FlightNumber and Payload variables affect the launch outcome
  - Investigate detailed launch records for each launch site
  - Investigate relationship between launch sites and their payload mass
  - Investigate relationship between FlightNumber and OrbitType
  - Investigate relationship between Payload and OrbitType
- Bar Chart
  - Investigate relationship between success rate and orbit type
- Line Chart
  - Get the average success trend
- GitHub URL:
  - https://github.com/camilalbmaia/data-science-notebooks/blob/main/Week%202%20-%202%20-%20Assignment-%20Exploring%20and%20Preparing%20Data.ipynb

EDA with SQL

- SQL queries
  - SELECT DISTINCT("Launch_Site") from SPACEXTABLE
  - SELECT * from SPACEXTABLE WHERE "Launch_Site" LIKE "CCA%" LIMIT 510
  - SELECT SUM("PAYLOAD_MASS__KG_") from SPACEXTABLE WHERE Customer = "NASA (CRS)"
  - SELECT AVG("PAYLOAD_MASS__KG_") from SPACEXTABLE WHERE "Booster_Version" LIKE "F9 v1.0%"
  - SELECT MIN(Date) from SPACEXTABLE WHERE "Landing_Outcome" = "Success (ground pad)"
  - SELECT "Booster_Version" from SPACEXTABLE WHERE "Mission_Outcome" = "Success" AND "PAYLOAD_MASS__KG_" > 4000 AND "PAYLOAD_MASS__KG_" < 6000
  - SELECT "Mission_Outcome", COUNT(*) from SPACEXTABLE GROUP BY "Mission_Outcome"
  - SELECT DISTINCT("Booster_Version") from SPACEXTABLE WHERE "PAYLOAD_MASS__KG_" = (SELECT MAX("PAYLOAD_MASS__KG_") FROM SPACEXTABLE)
  - SELECT substr(Date, 6,2) as "Month", "Landing_Outcome", "Booster_Version", "Launch_Site" FROM SPACEXTABLE WHERE "Landing_Outcome" = "Failure (drone ship)" AND substr(Date,0,5)="2015"
  - SELECT "Landing_Outcome", COUNT(*) FROM SPACEXTABLE GROUP BY "Landing_Outcome" HAVING Date BETWEEN '2010-06-04' and '2017-03-20' ORDER BY COUNT(*) DESC
- GitHub URL:
  - https://github.com/camilalbmaia/data-science-notebooks/blob/main/Week%202%20-%201%20-%20Assignment-%20SQL%20Notebook%20for%20Peer%20Assignment.ipynb

Build an Interactive Map with Folium

- A map was created having NASA Johnson Space Center as the start location
- The following map objects were added:
    - Circle
        - To add a highlighted circle area with a popup label for NASA Johnson Space Center
        - To add a highlighted circle area with a popup label for each launch site
    - Marker
        - To show icon for NASA Johnson Space Center
        - To show icon for each launch site
    - MarkerCluster
        - To group markers for the laugh outcomes for each site
    - MousePosition
        - To get coordinates for a mouse over a point on the map
    - PolyLine
        - To draw a line between each site to the selected coastline point

- GitHub URL:
    - https://github.com/camilalbmaia/data-science-notebooks/blob/main/Week%203%20-%20Launch%20Sites%20Locations%20Analysis%20with%20Foliu.ipynb

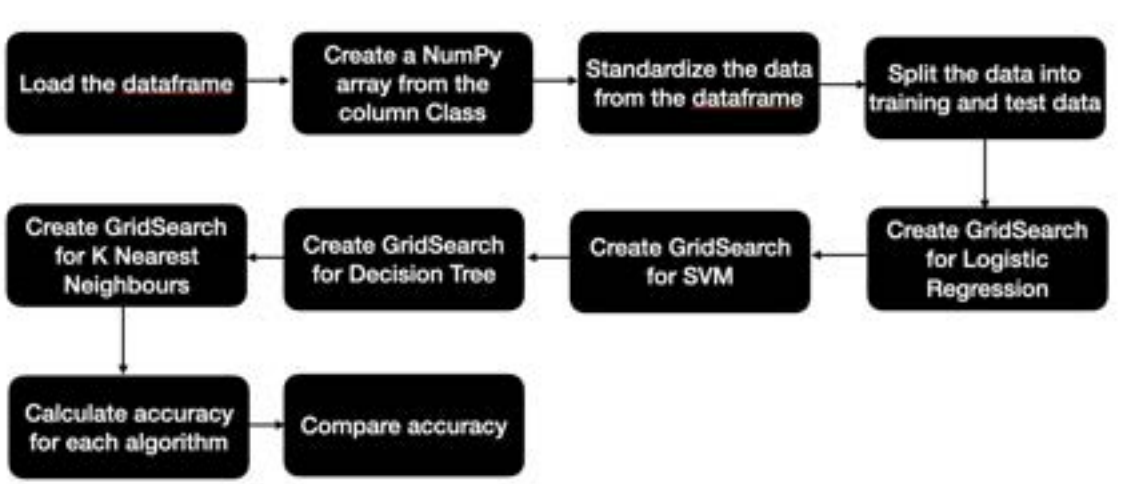
Build a Dashboard with Plotly Dash

- The following graphs were added to the dashboard:
  - Pie Chart
    - To show success rate
  - Scatter Chart
    - To visualize relationship between Payload and success rate
- Two input fields were added:
  - A Dropdown with Launch Sites. The default option is “ALL"
    - If the user selects “ALL”, the success rate is displayed for all launch sites
    - If the user selects a specific launch site, the success rate for the selected site only is displayed
    - The dropdown value is used as input for the two graphics above
  - A RangeSlider with values for Payload
    - The RangeSlider value is used for the second graph (Scatter Chart)
- GitHub URL:
  - https://github.com/camilalbmaia/data-science-notebooks/blob/main/spacex_dash_app.py

Predictive Analysis (Classification)

- Load the dataframe
- Create a NumPy array from the column Class in data
- Standardize the data from the dataframe
- Split the data into training and test data
- Create GridSearch for the following algorithms, and fit the object to find the best parameters
    - Logistic Regression
    - SVM
    - Decision Tree
    - K Nearest Neighbours
- Calculate and compare accuracy of the algorithms to find the one that performs best

- Github URL:
    - https://github.com/camilalbmaia/data-science-notebooks/blob/main/Week%204%20-%20Assignment-%20Machine%20Learning%20Prediction.ipynb



Predictive Analysis

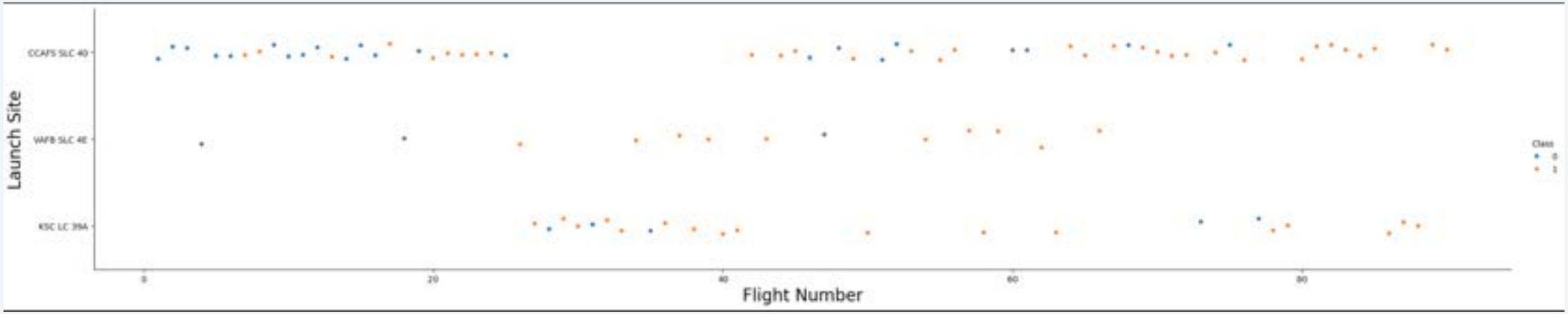
Results

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

Section 2

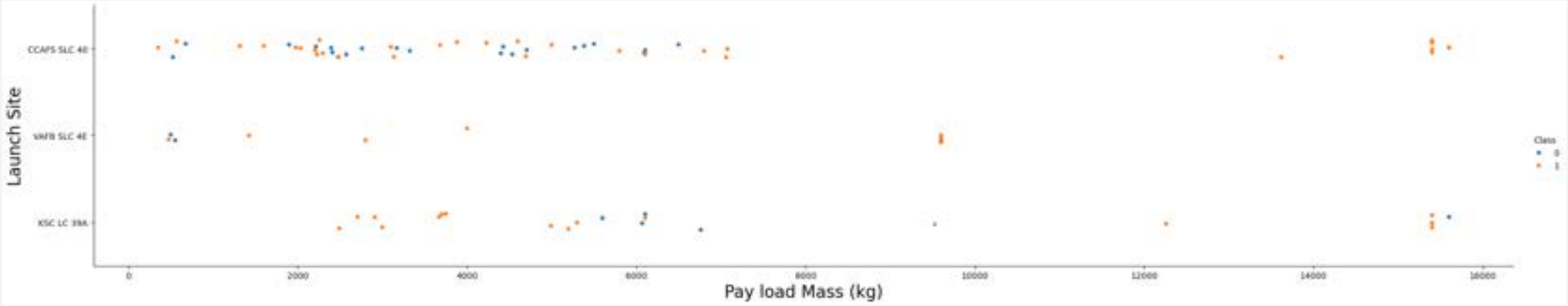
Insights drawn from EDA

Flight Number vs. Launch Site



- There are more failures than success results
- Launch site CCAFS SLC 40 has success rate greater than the other sites
- The latest launches failed in all launch sites
- There seems to be no relationship between Launch Site and Flight Number

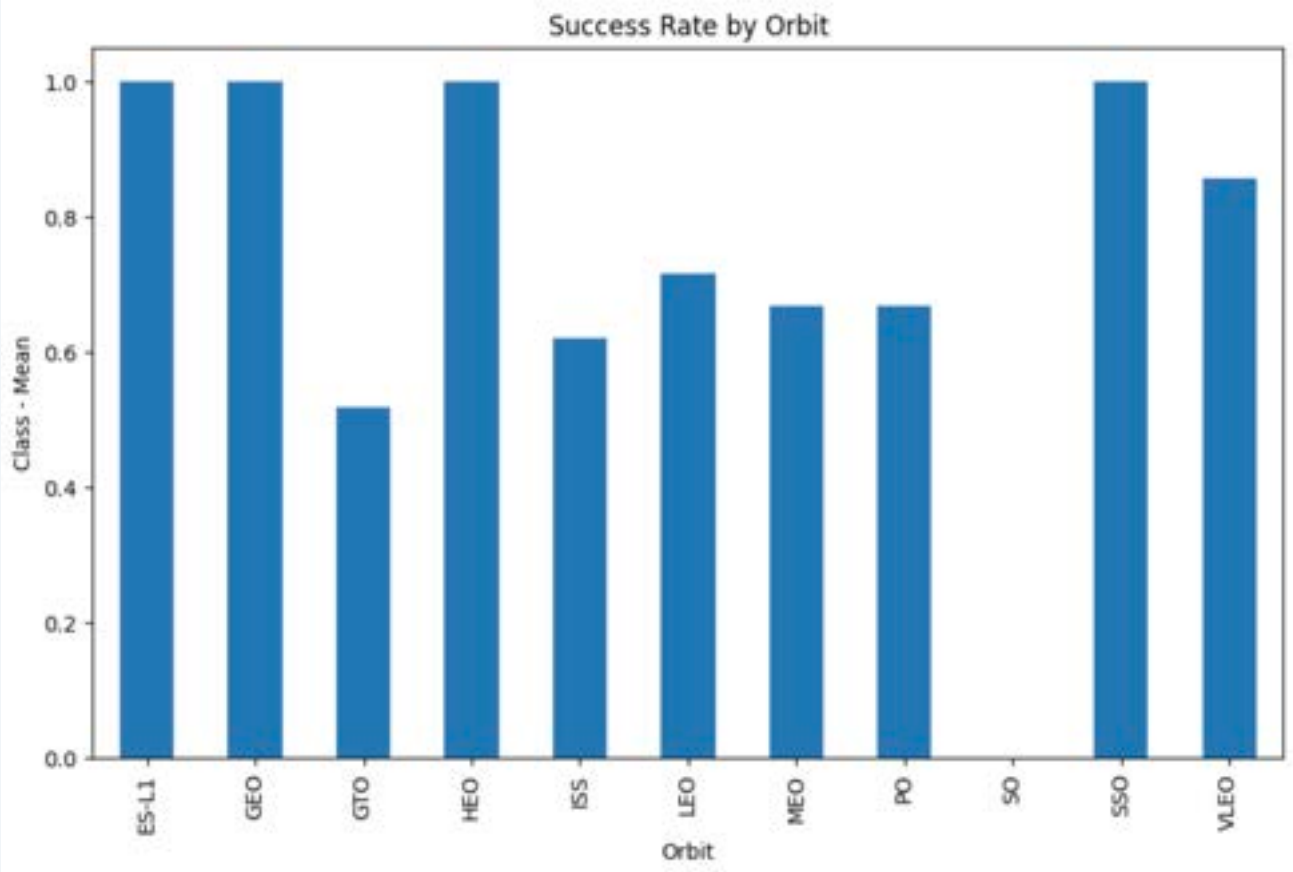
Payload vs. Launch Site



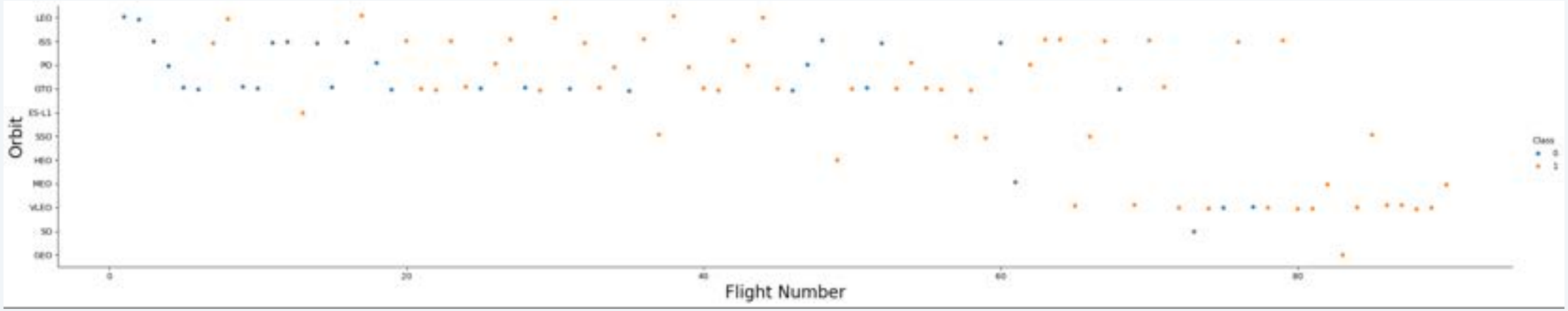
- Except by one launch, all launches with payload greater than 7000 kg failed
- Launch site CCAFS SLC 40 has success rate greater than the other sites
- Launch site VAFB SLC 4E had only two successful launches
- Launch site KSC LC 39A had success launches with payload close to 6000 Kg
- There seems to be no relationship between Launch Site and Payload

Success Rate vs. Orbit Type

- The orbits with higher success rate are: ES-L1, GEO, HEO, and SSO
- There are no successful launches for orbit SO

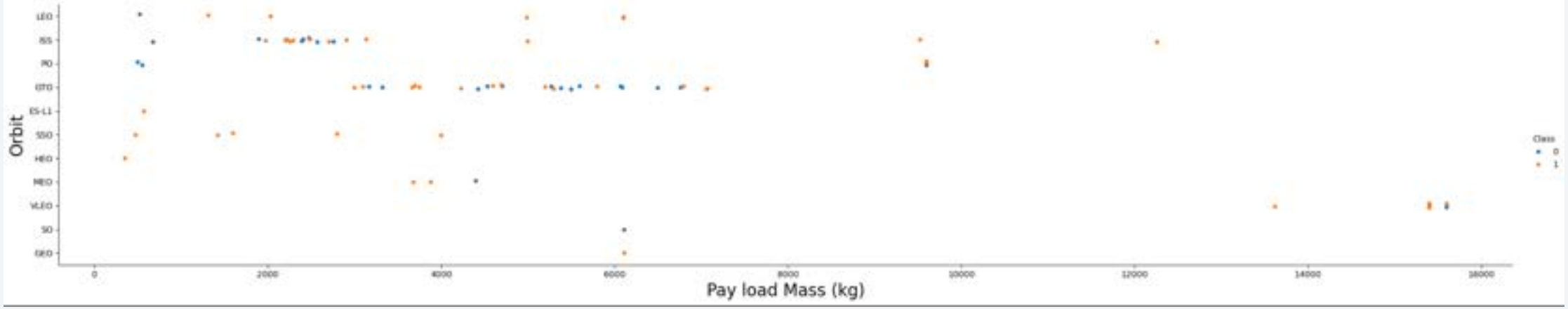


Flight Number vs. Orbit Type



- In the LEO orbit the success appears related to the number of flights
- There seems to be no relationship between Flight Number and most of orbits

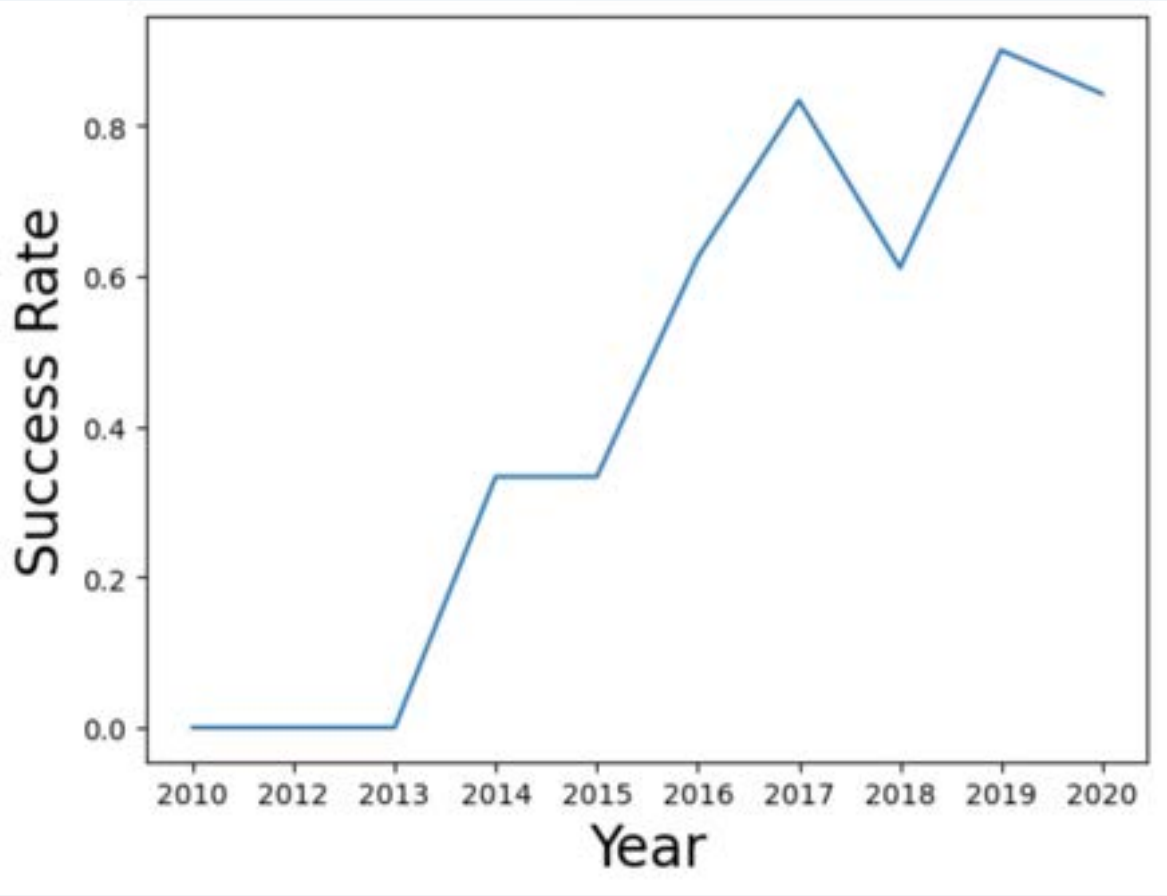
Payload vs. Orbit Type



- With heavy payloads the successful landing or positive landing rate are more for Polar, LEO, and ISS.
- For GTO we cannot distinguish this well as both positive landing rate and negative landing are both there.

Launch Success Yearly Trend

- The success rate since 2013 kept increasing till 2020



All Launch Site Names

- Used DISTINCT to find all Launch Site names

```
%sql SELECT DISTINCT("Launch_Site") from SPACEXTABLE
```

```
 * 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'

- Used LIMIT to find 5 records where launch site names begin with `CCA`

```
%sql SELECT * from SPACEXTABLE WHERE "Launch_Site" LIKE "CCA%" LIMIT 5
```

Python

```
 * 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

- Used SUM to calculate the total payload carried by boosters from NASA

```
%sql SELECT SUM("PAYLOAD_MASS__KG_") from SPACEXTABLE WHERE Customer = "NASA (CRS)"
```

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

SUM("PAYLOAD_MASS__KG_")
45596

Average Payload Mass by F9 v1.1

- Used AVG and LIKE to calculate the average payload mass carried by booster version F9 v1.1

```
%sql SELECT AVG("PAYLOAD_MASS__KG_") from SPACEXTABLE WHERE "Booster_Version" LIKE "F9 v1.1%"
```

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

AVG("PAYLOAD_MASS__KG_")
2534.6666666666665

First Successful Ground Landing Date

- Used MIN to find the first successful ground landing date

```
%sql SELECT MIN(Date) from SPACEXTABLE WHERE "Landing_Outcome" = "Success (ground pad)"
```

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

MIN(Date)
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- Simple query to find the names of the boosters that satisfies the request

```
%sql SELECT "Booster_Version" from SPACEXTABLE WHERE "Mission_Outcome" = "Success" AND "PAYLOAD_MASS__KG_" > 4000 AND "PAYLOAD_MASS__KG_" < 6000

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

Booster_Version
F9 v1.1
F9 v1.1 B1011
F9 v1.1 B1014
F9 v1.1 B1016
F9 FT B1020
F9 FT B1022
F9 FT B1026
F9 FT B1030
F9 FT B1021.2
F9 FT B1032.1
F9 B4 B1040.1
F9 FT B1031.2
F9 FT B1032.2
F9 B4 B1040.2
F9 B5 B1046.2
F9 B5 B1047.2
F9 B5 B1048.3
F9 B5 B1051.2
F9 B5B1060.1
F9 B5 B1058.2
F9 B5B1062.1

Total Number of Successful and Failure Mission Outcomes

- Used COUNT and GROUP BY to find the total number of successful and failure mission outcomes

```
%sql SELECT "Mission_Outcome", COUNT(*) from SPACEXTABLE GROUP BY "Mission_Outcome"
```

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

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

Boosters Carried Maximum Payload

- Used subquery to get the maximum payload mass and then find the boosters that carried that payload

```
%sql SELECT DISTINCT("Booster_Version") from SPACEXTABLE WHERE "PAYLOAD_MASS__KG_" = (SELECT MAX("PAYLOAD_MASS__KG_") FROM SPACEXTABLE)
```

```
 * 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

- Used subset to find the list of landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

```
%sql SELECT substr(Date, 6,2) as "Month", "Landing_Outcome", "Booster_Version", "Launch_Site" FROM SPACEXTABLE WHERE "Landing_Outcome" = "Failure (drone ship)" AND substr(Date,0,5)="2015"
```

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

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

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

- Used GROUP BY, COUNT, and ORDER BY to 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

```
%sql SELECT "Landing_Outcome", COUNT(*) FROM SPACEXTABLE GROUP BY "Landing_Outcome" HAVING Date BETWEEN '2010-06-04' and '2017-03-20' ORDER BY COUNT(*) DESC
```

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

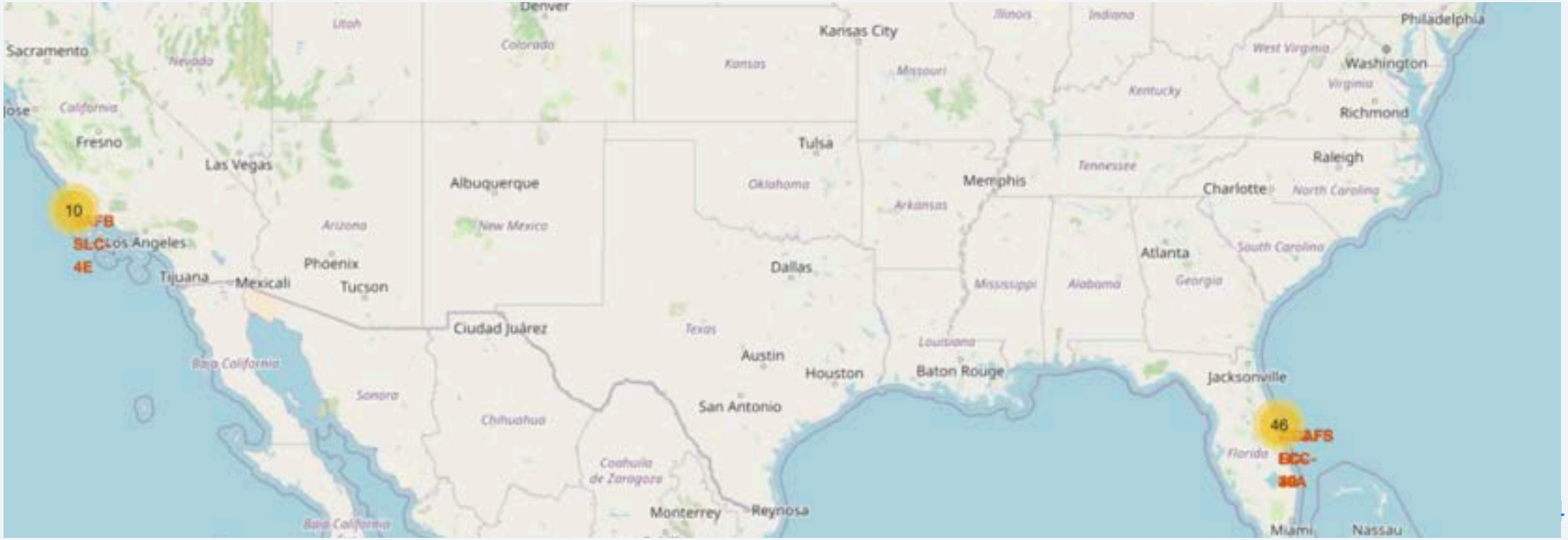
Landing_Outcome	COUNT(*)
No attempt	21
Success (drone ship)	14
Success (ground pad)	9
Failure (drone ship)	5
Controlled (ocean)	5
Uncontrolled (ocean)	2
Precluded (drone ship)	1

Section 3

Launch Sites Proximities Analysis

Launch sites location

- There are 10 launch sites at the West coast and 46 at the East Coast
- Added MarkerClusters and used launch site coordinates to add markers

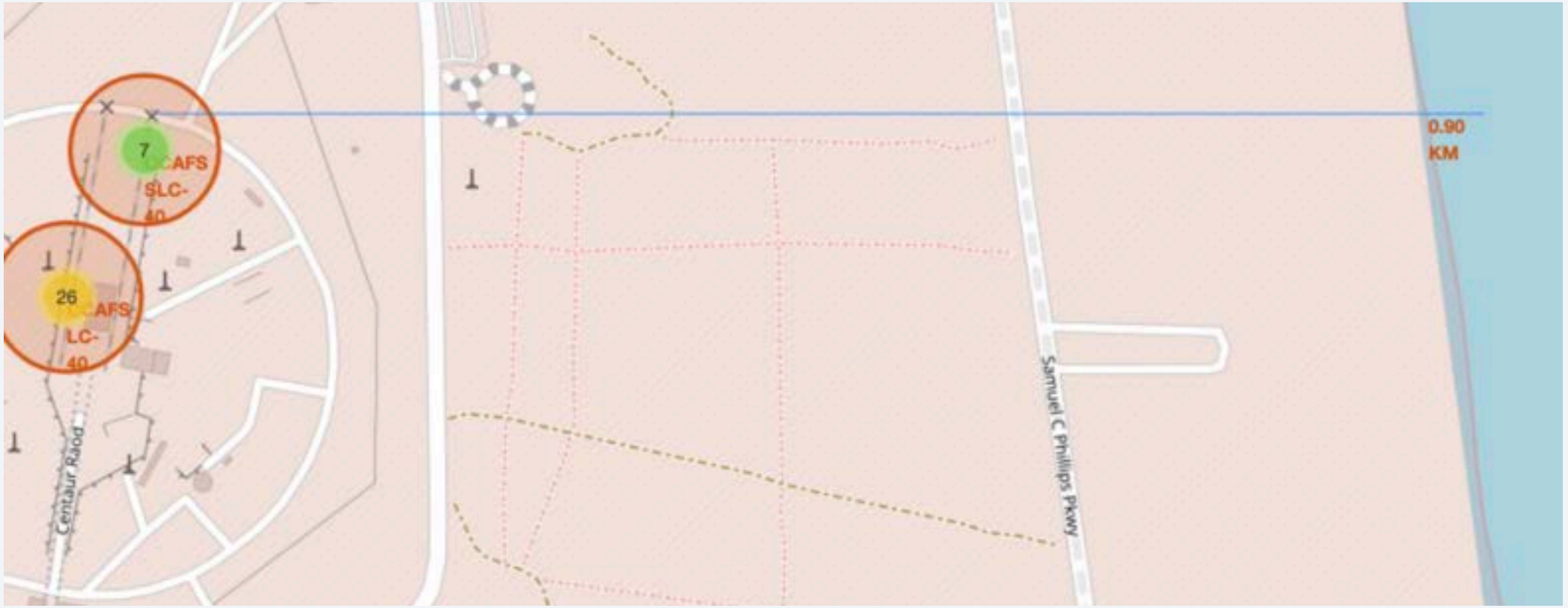


Success and failed launches

- The following map shows an example of the color-labeled launch outcomes on the map

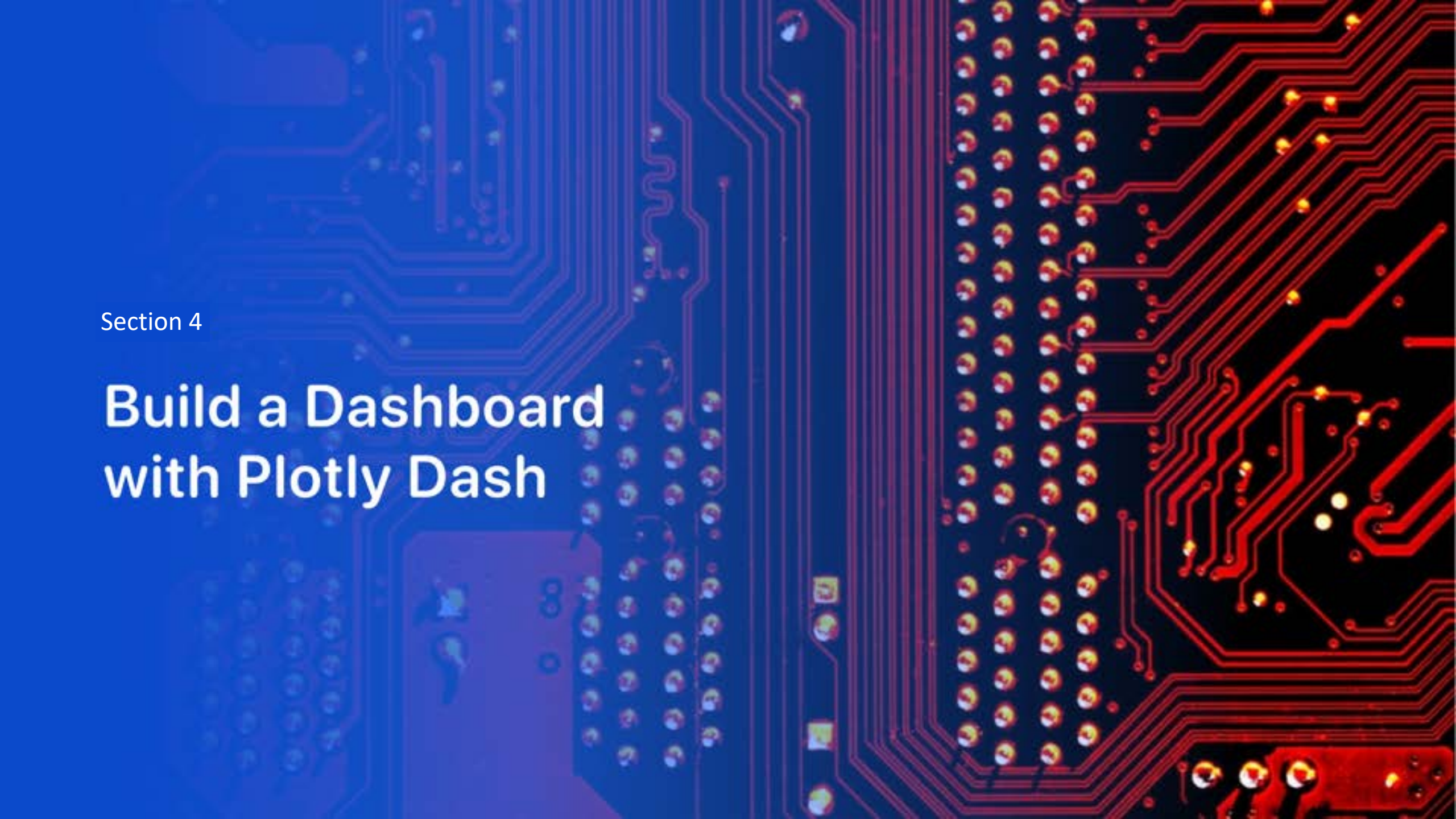
Distance between launch site and coastline

- Used PolyLine to draw distances between launch sites and coastline



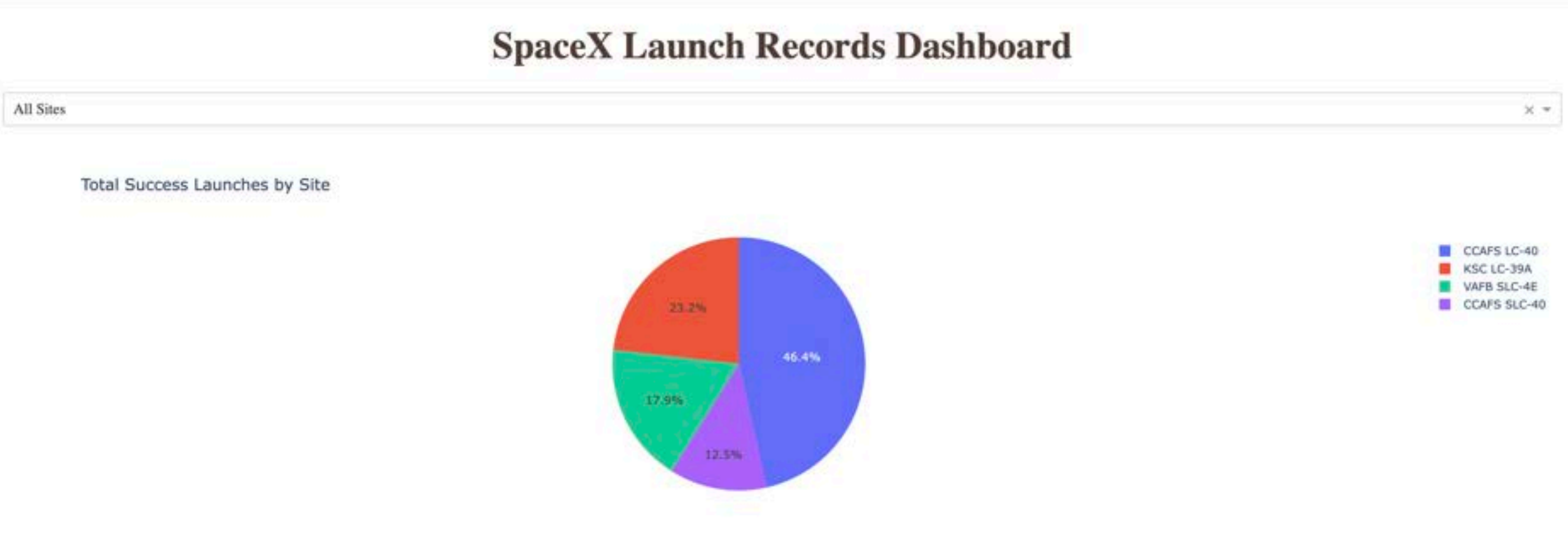
Section 4

Build a Dashboard with Plotly Dash



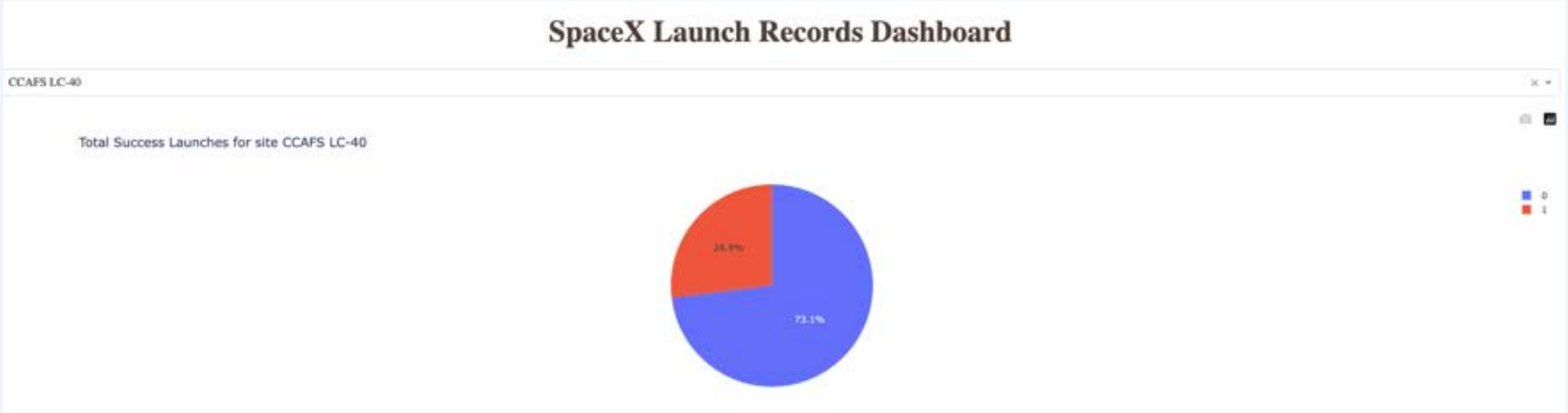
Dashboard - Total Success Launches by Site

- The launch site with higher success rate is CCAFS LC-40
- The launch site with lower success rate is CCAFS SLC-40



Dashboard - Launch results for launch site CCAFS LC-40

- CCAFS LC-40 is the launch site with highest launch success ratio
- 73.1 % of the launches were successful, while 26.9% of the launches failed



Dashboard - Payload vs. Launch Outcome

- Below is the Payload vs. Launch Outcome scatter plot for all sites, with all payload range selected in the range slider
- The booster version that have the largest success rate is FT, while the booster version that have the smallest success rate is V1.1.
- For payload > 6000 there is only one successful launch. All others failed.

Dashboard - Payload vs. Launch Outcome

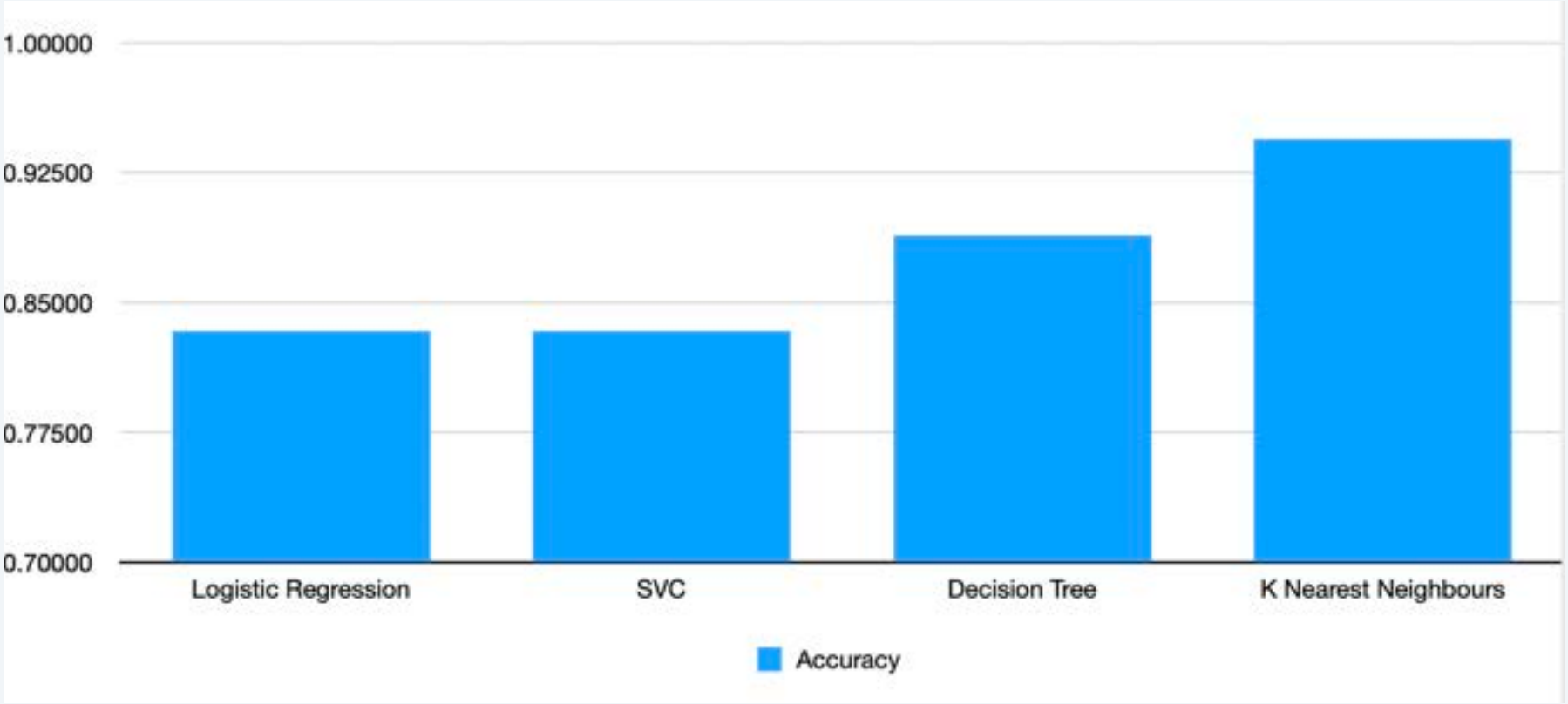
- Details with payload range selected between 2000 and 6000

Section 5

Predictive Analysis (Classification)

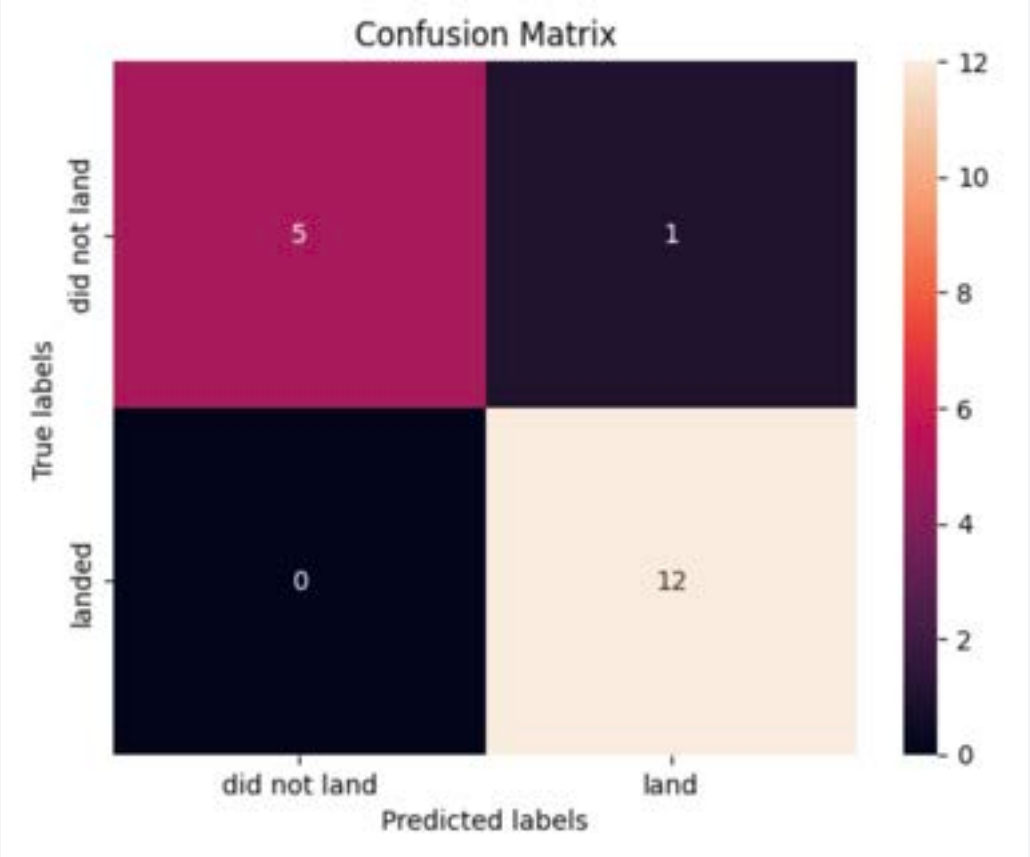
Classification Accuracy

- The following bar chart has the accuracy for all built classification models analyzed
- The model with highest classification accuracy was K Nearest Neighbours



Confusion Matrix for K Nearest Neighbours

- The value in line 1and column 1 represents the launches that were predicted to not land and didn't land: 5
- The value in line 1and column 2 represents the launches that were predicted to land and didn't land: 1
- The value in line 2 and column 1 represents the launches that were predicted to not land and landed: 0
- The value in line 2 and column 2 represents the launches that were predicted to land and landed: 12



Conclusions

- Launch site CCAFS SLC 40 has success rate greater than the other sites
- The orbits with higher success rate are: ES-L1, GEO, HEO, and SSO
- The success rate since 2013 kept increasing till 2020, when decreased again
- The launch site with higher success rate is CCAFS LC-40, while the one with lower success rate is CCAFS SLC-40
- The booster version that have the largest success rate is FT, while the booster version that have the smallest success rate is V1.1.
- For payload > 6000 there is only one successful launch. All others failed.
- As per the confusion matrix of the model with highest accuracy (K Nearest Neighbours), most predictions of this model were correct

Appendix

- Code snippet (Python file) for the dashboard

```
# Read the airline data into pandas dataframe
spacex_df = pd.read_csv("spacex_launch_dash.csv")
max_payload = spacex_df['Payload Mass (kg)'].max()
min_payload = spacex_df['Payload Mass (kg)'].min()

# Dropdown options
list_sites = spacex_df['Launch Site'].unique()
site_options = []
site_options.append({'label': 'All Sites', 'value': 'ALL'})
for site in list_sites:
    site_options.append({'label': site, 'value': site})

# Create a dash application
app = dash.Dash(__name__)

# Create an app layout
app.layout = html.Div(children=[html.H1('SpaceX Launch Records Dashboard',
                                        style={'textAlign': 'center', 'color': '#503036',
                                               'font-size': 40}),
                                # TASK 1: Add a dropdown list to enable Launch Site selection
                                # The default select value is for ALL sites
                                  dcc.Dropdown(id='site-dropdown',
                                    options=site_options,
                                    value='ALL',
                                    placeholder="Select a Launch Site here",
                                    searchable=True
                                    ),
                                html.Br(),

                                # TASK 2: Add a pie chart to show the total successful launches count for all sites
                                # If a specific launch site was selected, show the Success vs. Failed counts for the site
                                html.Div(dcc.Graph(id='success-pie-chart')),
                                html.Br(),

                                html.P("Payload range (Kg):"),
                                # TASK 3: Add a slider to select payload range
                                dcc.RangeSlider(id='payload-slider',
                                    min=0,
                                    max=10000,
                                    step=1000,
                                    value=[min_payload, max_payload]),

                                # TASK 4: Add a scatter chart to show the correlation between payload and launch success
                                html.Div(dcc.Graph(id='success-payload-scatter-chart')),
                                ])
```

Thank you!