IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Anton Stanev
10.10.2023

Outline

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

Executive Summary

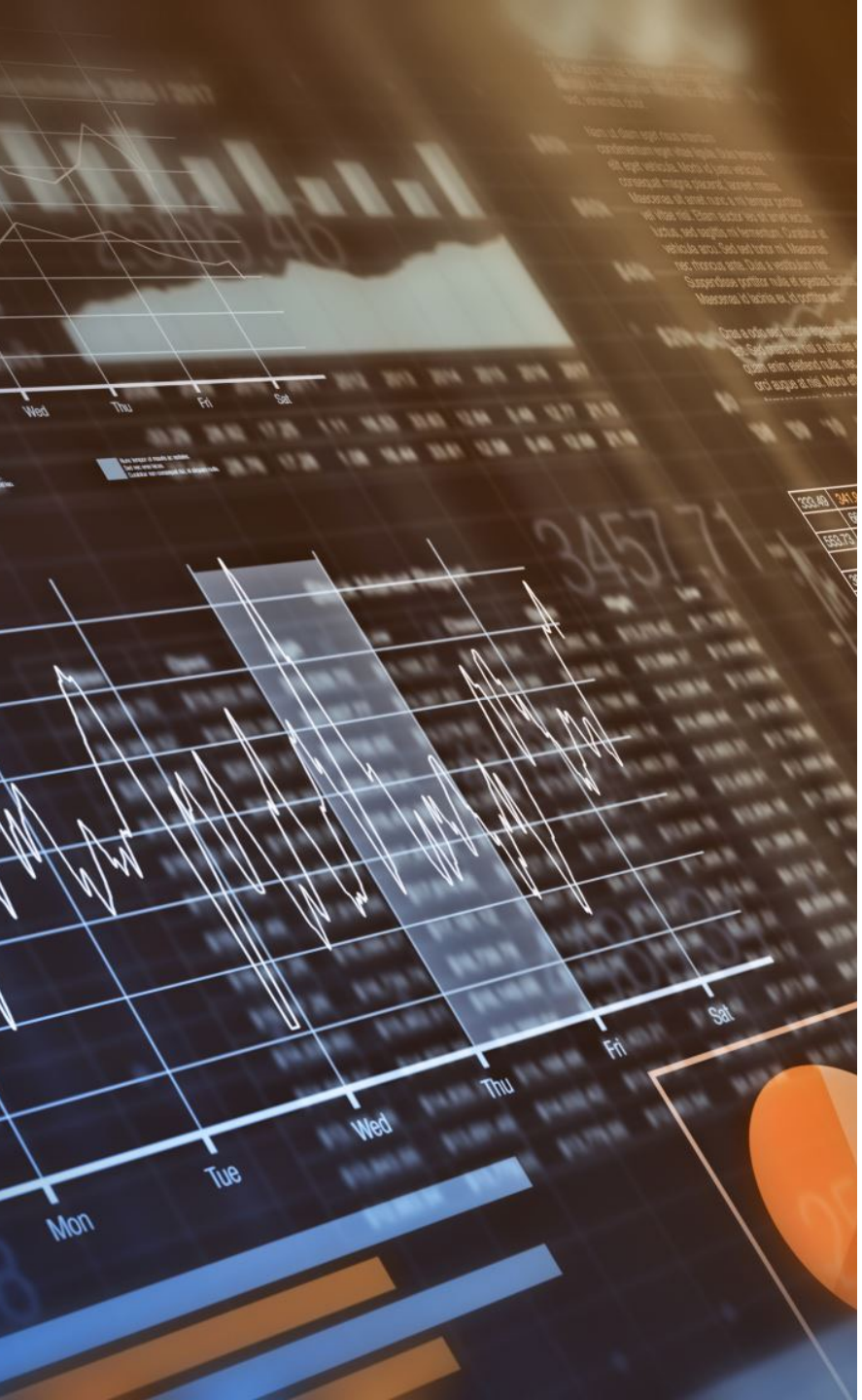
- Summary of methodologies
- Summary of all results

Introduction

- Project background and context
- Problems you want to find answers

Section 1

Methodology



Methodology

- Executive Summary
- Data collection methodology:
  - The data was collected from Wikipedia using the WebScraping method and the BeautifulSoup library
- Perform data wrangling
  - The data was cleaned from missing values, standardized and enchanced
- 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

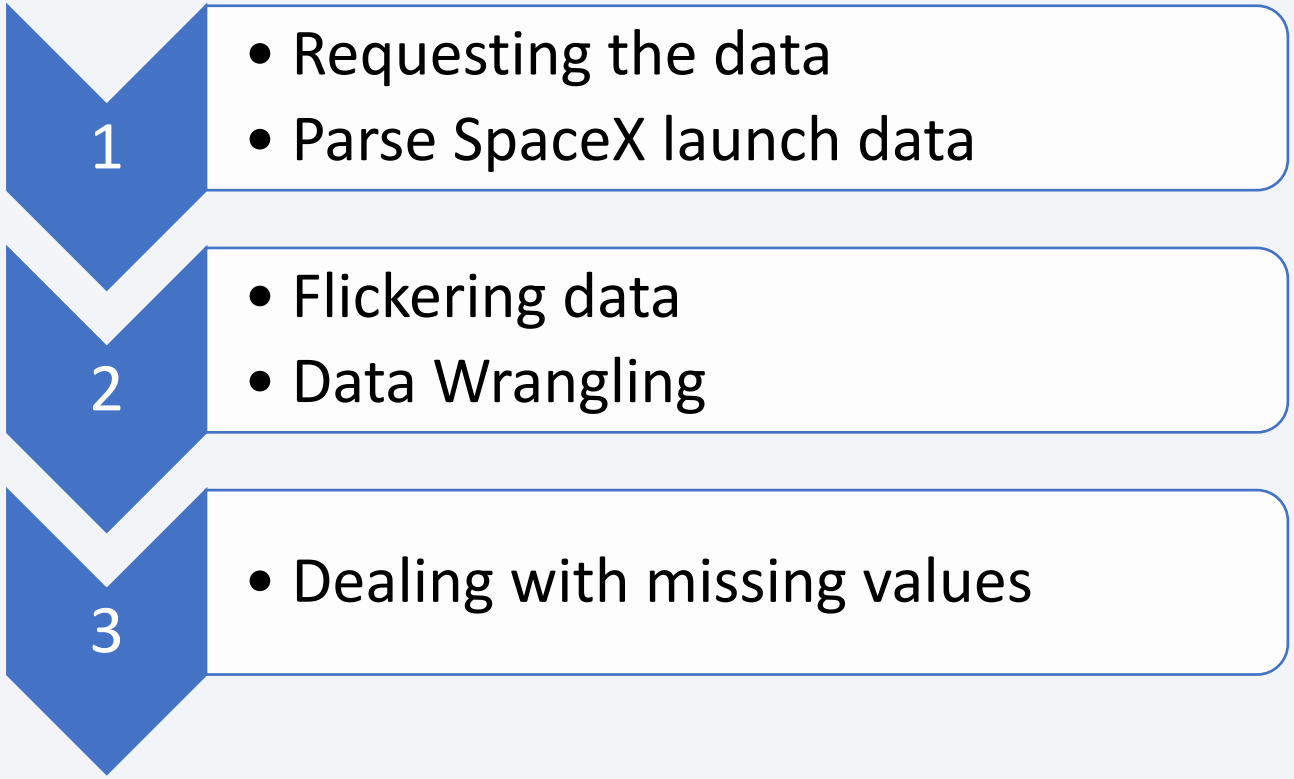
- The data was obtained from the Wikipedia page for the project of SpaceX which is an American spacecraft manufacturer and launch service provider.
- The technology used to collect the data is known as Web Scraping which pulls data from HTML and XML files.

	Flight No.	Launch site	Payload	Payload mass	Orbit	Customer	Launch outcome	Version Booster	Booster landing	Date	Time
0	1	CCAFS	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success\n	F9 v1.0B0003.1	Failure	4 June 2010	18:45
1	1	CCAFS	Dragon	0	LEO	NASA	Success	F9 v1.0B0003.1	Failure	4 June 2010	18:45
2	2	CCAFS	Dragon	525 kg	LEO	NASA	Success	F9 v1.0B0004.1	No attempt\n	8 December 2010	15:43

Data Collection – SpaceX API

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

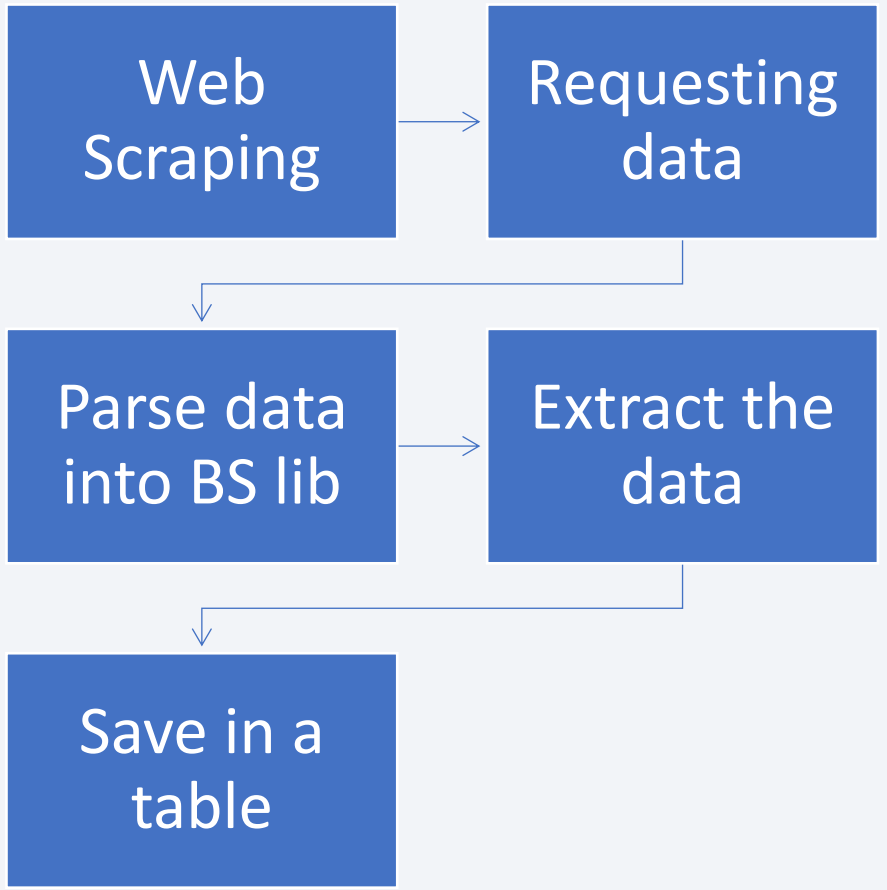
- GitHub URL - https://github.com/Anton-Stanev/Data-Science-Capstone/blob/main/jupyter-labs-spacex-data-collection-api.ipynb



Data Collection - Scraping

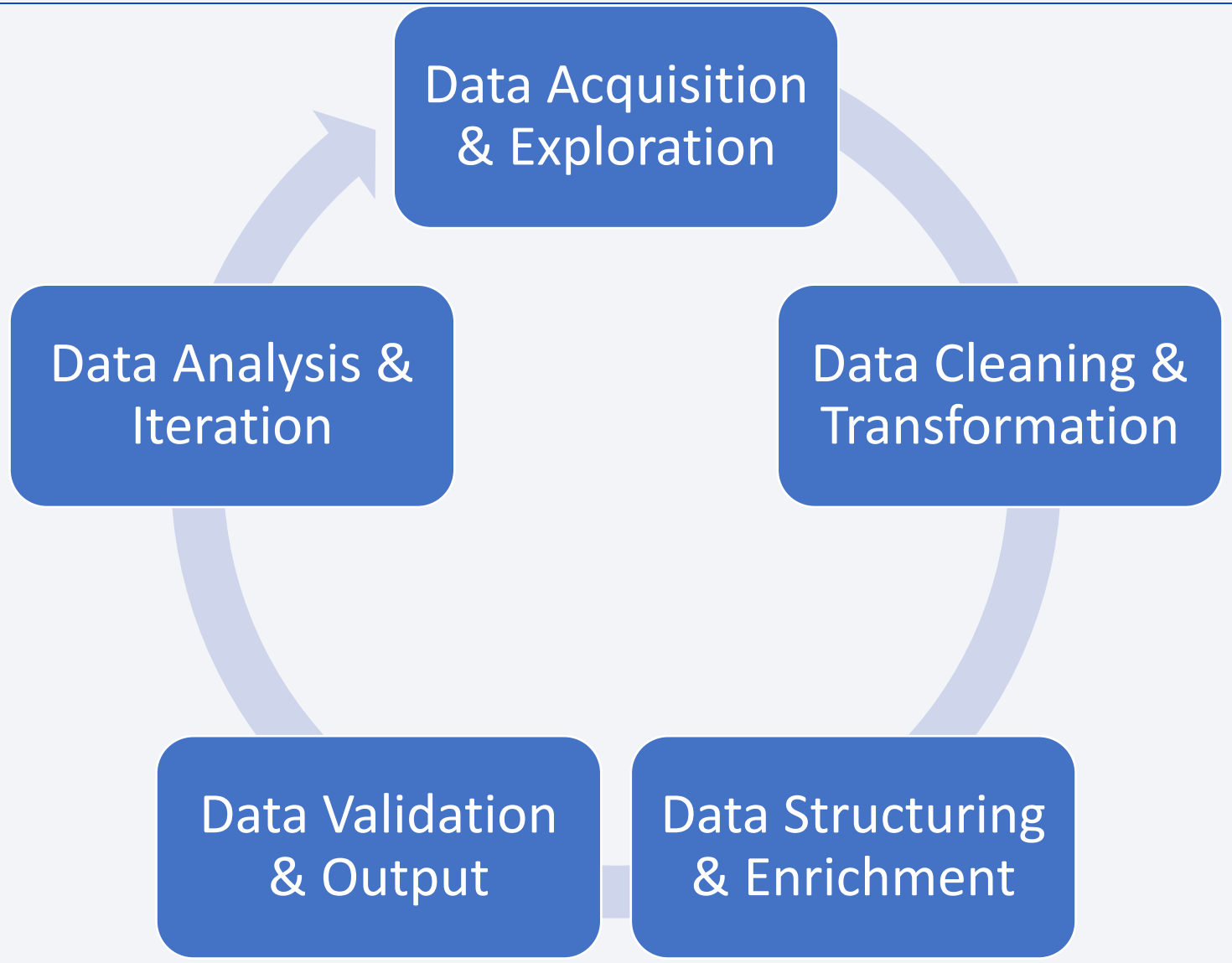
- Web Scraping presentation and flowchart

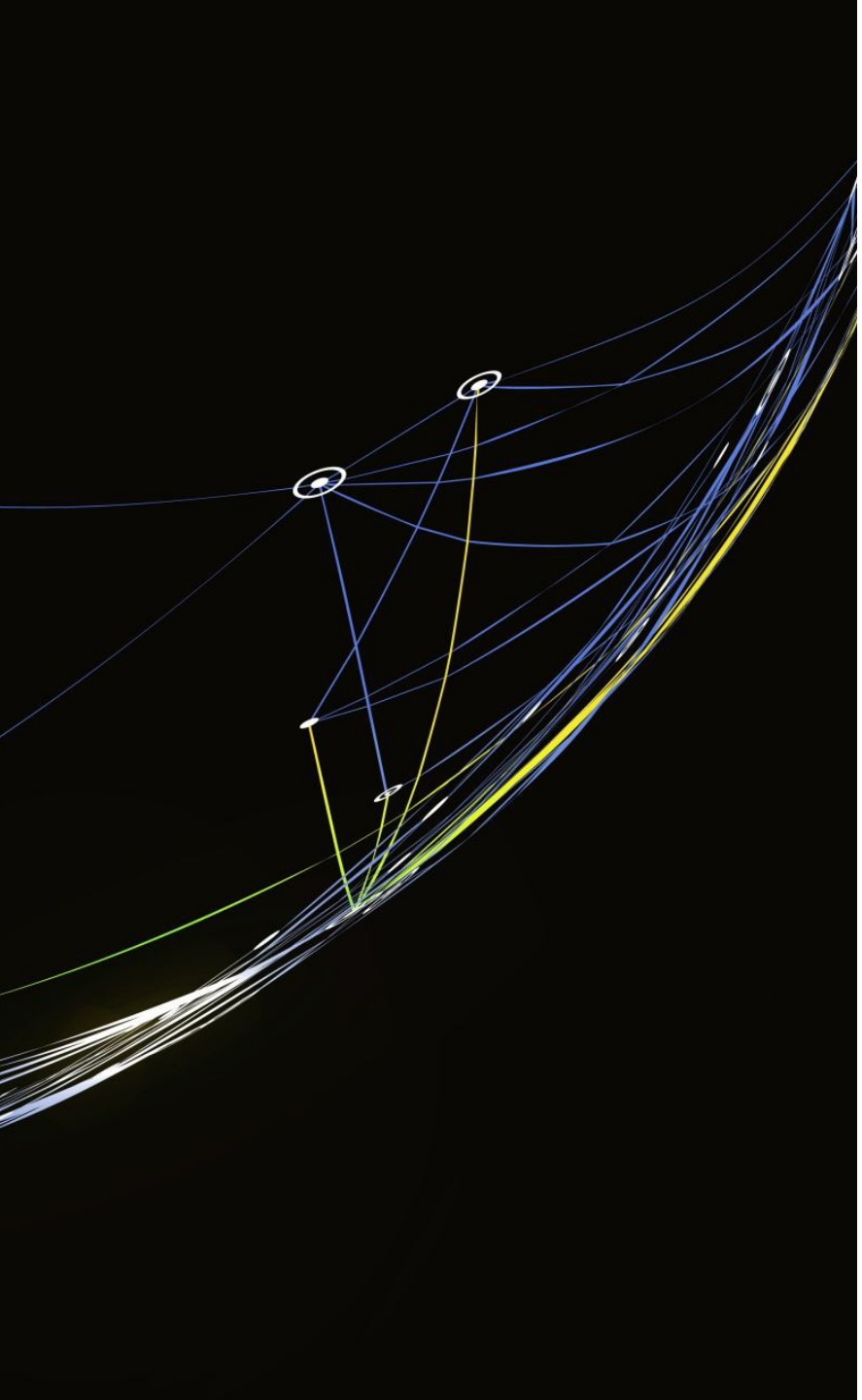
- GitHub URL - https://github.com/Anton-Stanev/Data-Science-Capstone/blob/main/jupyter-labs-webscraping.ipynb



Data Wrangling

- GitHub URL - https://github.com/Anton-Stanev/Data-Science-Capstone/blob/main/labs-jupyter-spacex-Data wrangling.ipynb





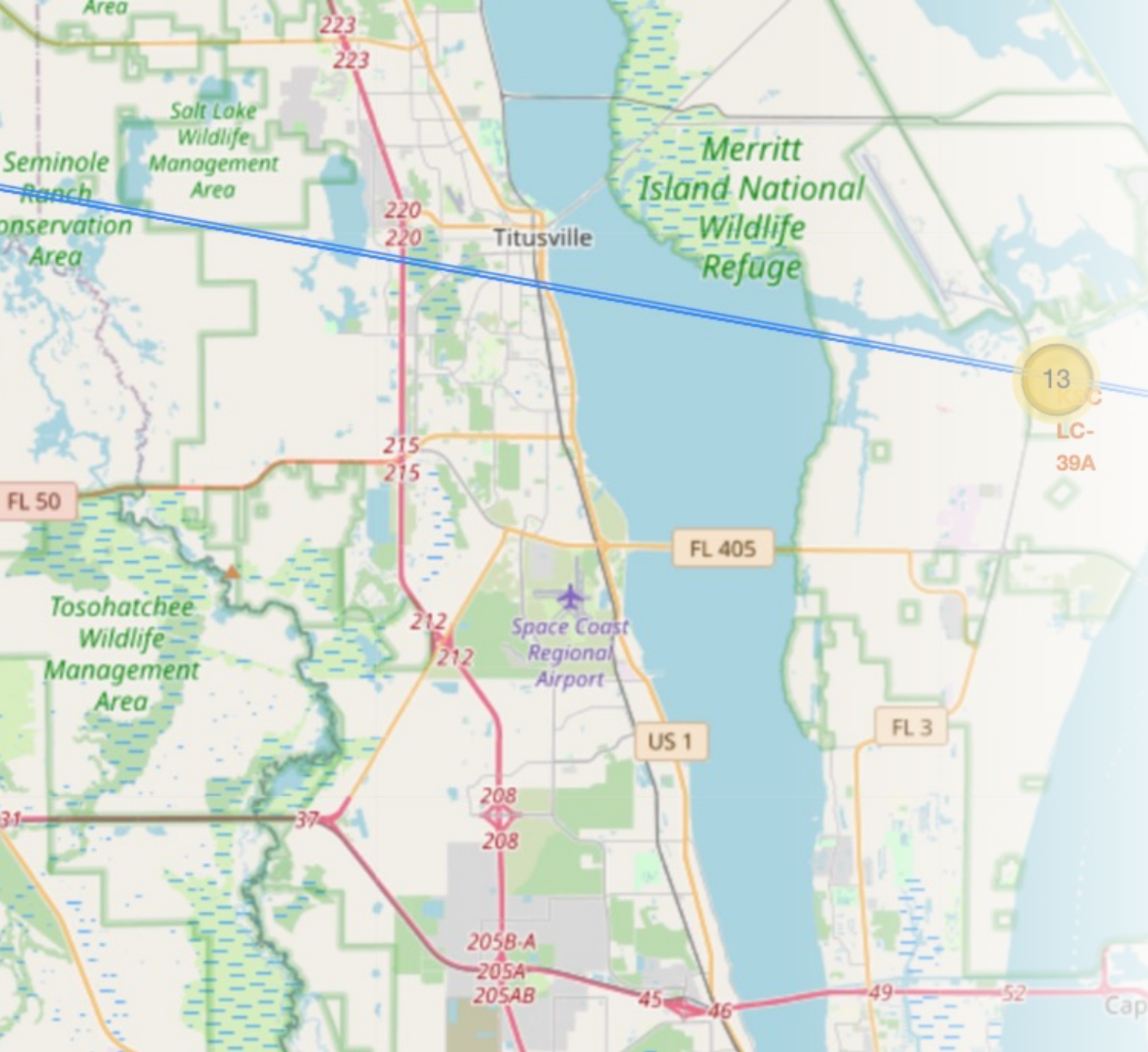
EDA with Data Visualization

- Vertical Chevron List - Used for data collection as there are few steps which are executed one after another
- Repeating Bending Process – Used for Web Scraping as it has a similar vision to the table output of the task
- Continuous Cycle – For Data Wrangling the process might be iterative.

- GitHub URL - https://github.com/Anton-Stanev/Data-Science-Capstone/blob/main/jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb

EDA with SQL

- %sql create table SPACEXTABLE as select * from SPACEXTBL where Date is not null
- %sql select distinct Launch_Site from SPACEXTABLE
- %sql select * from SPACEXTABLE where Launch_Site like 'CCA%' limit 5
- %sql select SUM(PAYLOAD_MASS__KG_) from SPACEXTABLE where Customer like 'NASA (CRS)'
- %sql select AVG(PAYLOAD_MASS__KG_) from SPACEXTABLE where Booster_Version like 'F9 v1.1%'
- %sql select min(Date) from SPACEXTABLE where Landing_Outcome like 'Suc%'
- %sql select * from SPACEXTABLE limit 20
- %sql select Booster_Version from SPACEXTABLE where PAYLOAD_MASS__KG_ > 4000 and PAYLOAD_MASS__KG_ < 6000 and Landing_Outcome = 'Success (ground pad)'
- %sql select count(Mission_Outcome) from SPACEXTABLE
- %sql select Booster_Version from SPACEXTABLE where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEXTABLE)
- %sql select * from SPACEXTABLE where substr(Date,0,5) = '2015'
- %sql select Landing_Outcome from SPACEXTABLE where (Landing_Outcome = 'Failure (drone ship)' or Landing_Outcome = 'Success (ground pad)') and (Date > '2010-06-04' and Date < '2017-03-20')
- GitHub URL - https://github.com/Anton-Stanev/Data-Science-Capstone/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb



Build an Interactive Map with Folium

- GitHub URL - https://github.com/Anton-Stanev/Data-Science-Capstone/blob/main/lab_jupyter_launch_site_location.jupyterlite.ipynb

Build a Dashboard with Plotly Dash

- GitHub URL - https://github.com/Anton-Stanev/Data-Science-Capstone/blob/main/lab_jupyter_launch_site_location.jupyterlite.ipynb
- The locations added on the map are the ones where the tests for the landing of the SpaceX shuttles are made
- I Choose to add a line through the whole states from east to west, to determine what is the distance between the two most distant points of landings
- The locations latitude and longitude were obtained with the help of MousePosition Folium Plugin

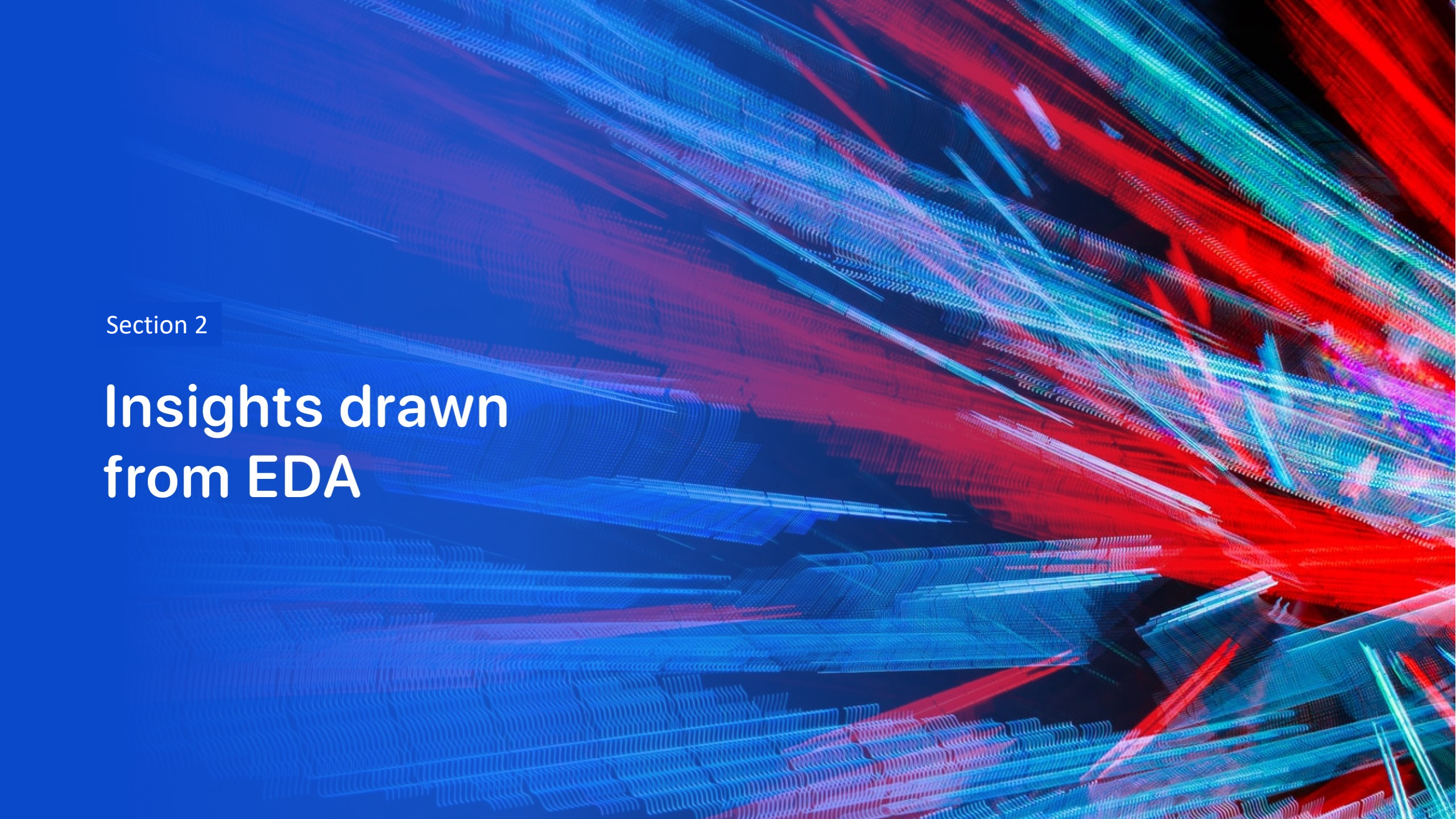
Predictive Analysis (Classification)

GitHub URL - https://github.com/Anton-Stanev/Data-Science-Capstone/blob/main/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

- I made an evaluation between several different models and their score on the actual data. Here are the names of the models used:
- `LogisticRegression`
- `SVC`
- `DecisionTreeClassifier`
- `KNeighborsClassifier – Best Performance Model`

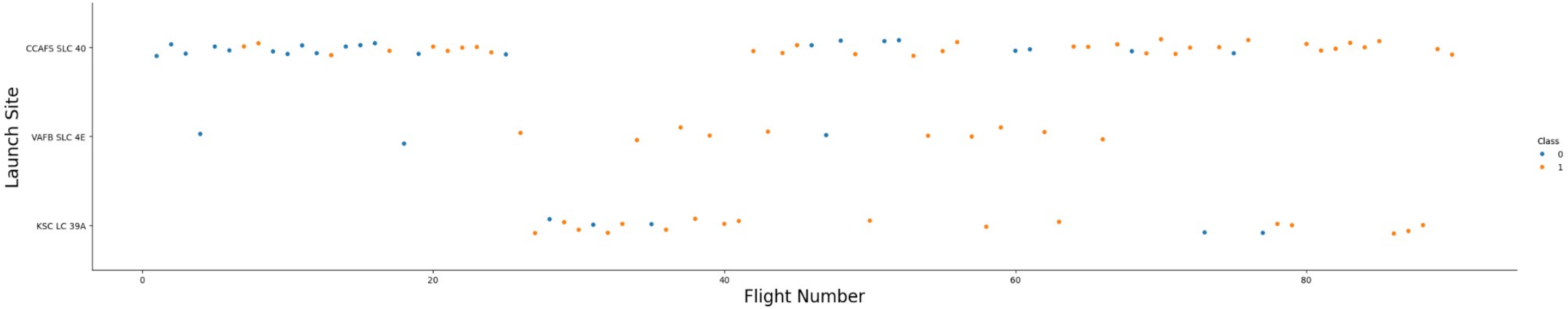
Results

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


Section 2
Insights drawn from EDA

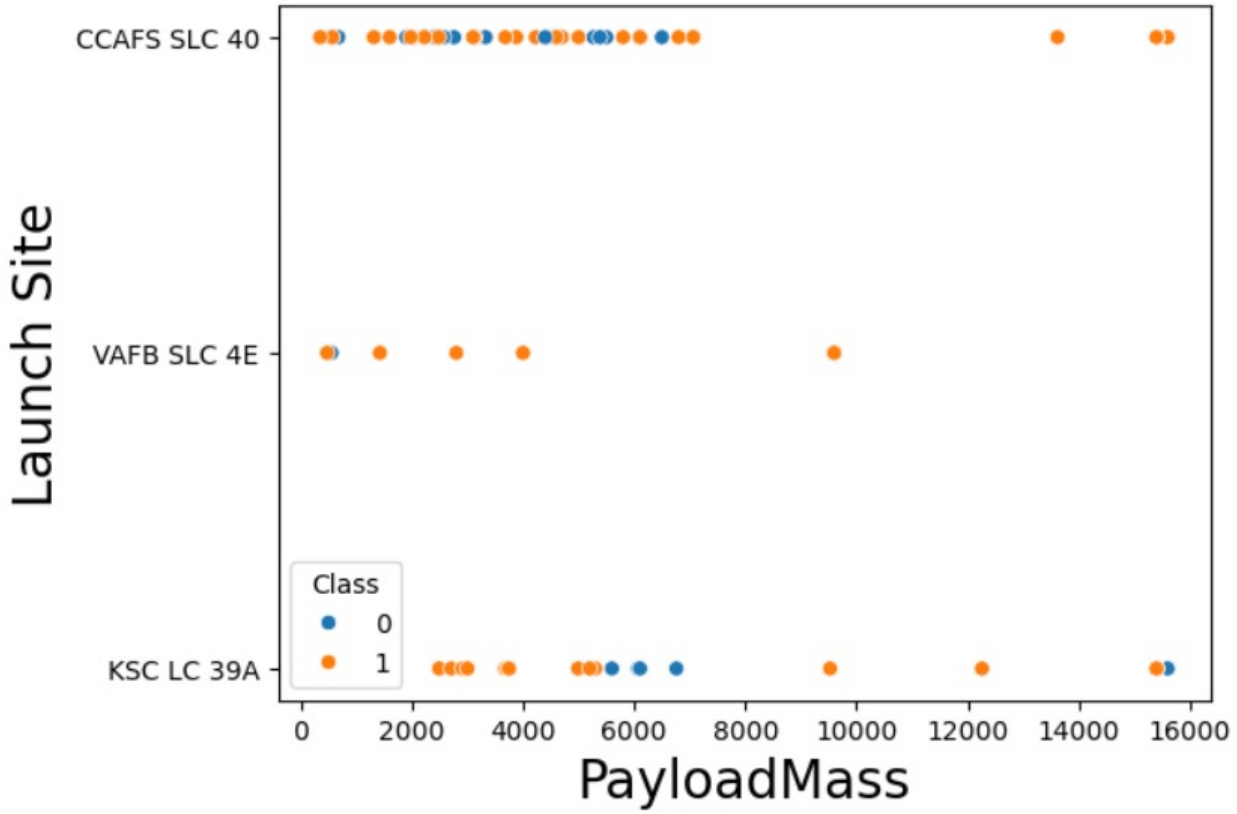
Flight Number vs. Launch Site

- Show a scatter plot of Flight Number vs. Launch Site
- We can see that we started KSC LC 39A after the 30th flight, and the very end of the VAFB SLC 4E is about 65th flight number.

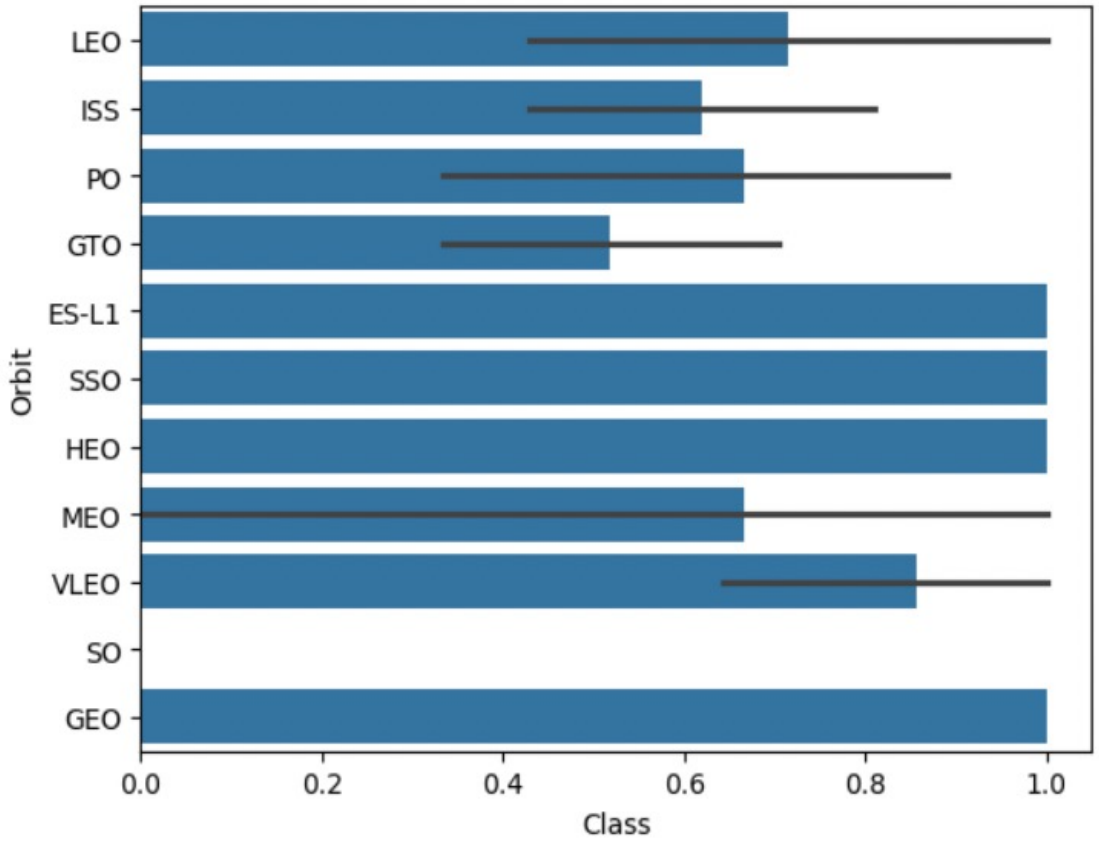


Payload vs. Launch Site

- Show a scatter plot of Payload vs. Launch Site
- From this plot, we obtain information that the heaviest payloads are performed by CCAFS and KSC launch sites.

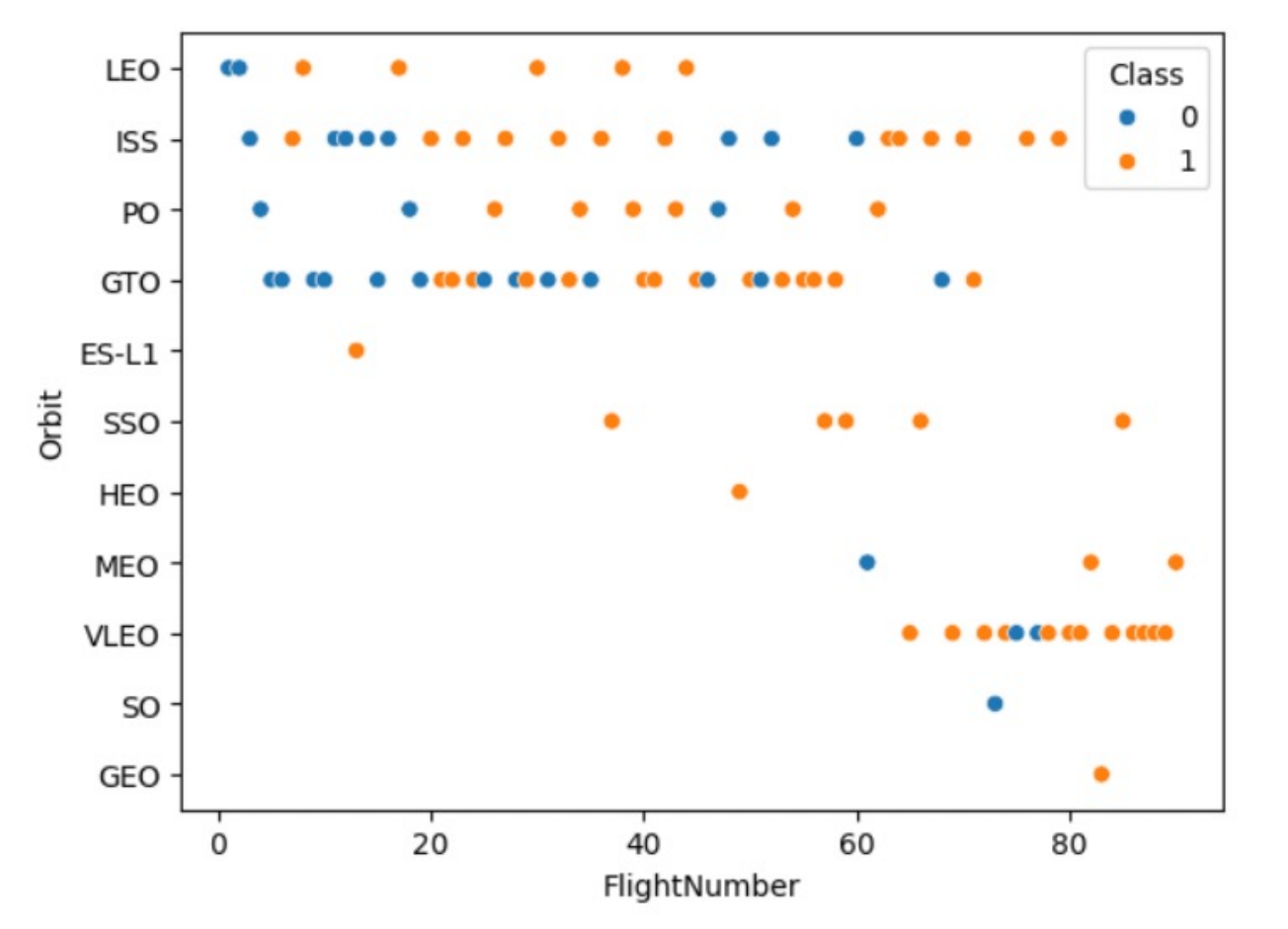


Success Rate vs. Orbit Type



- Show a bar chart for the success rate of each orbit type

- We can see that there are 3 Orbits that have no failures within the launches. ES-L1, SSO, and HEO.



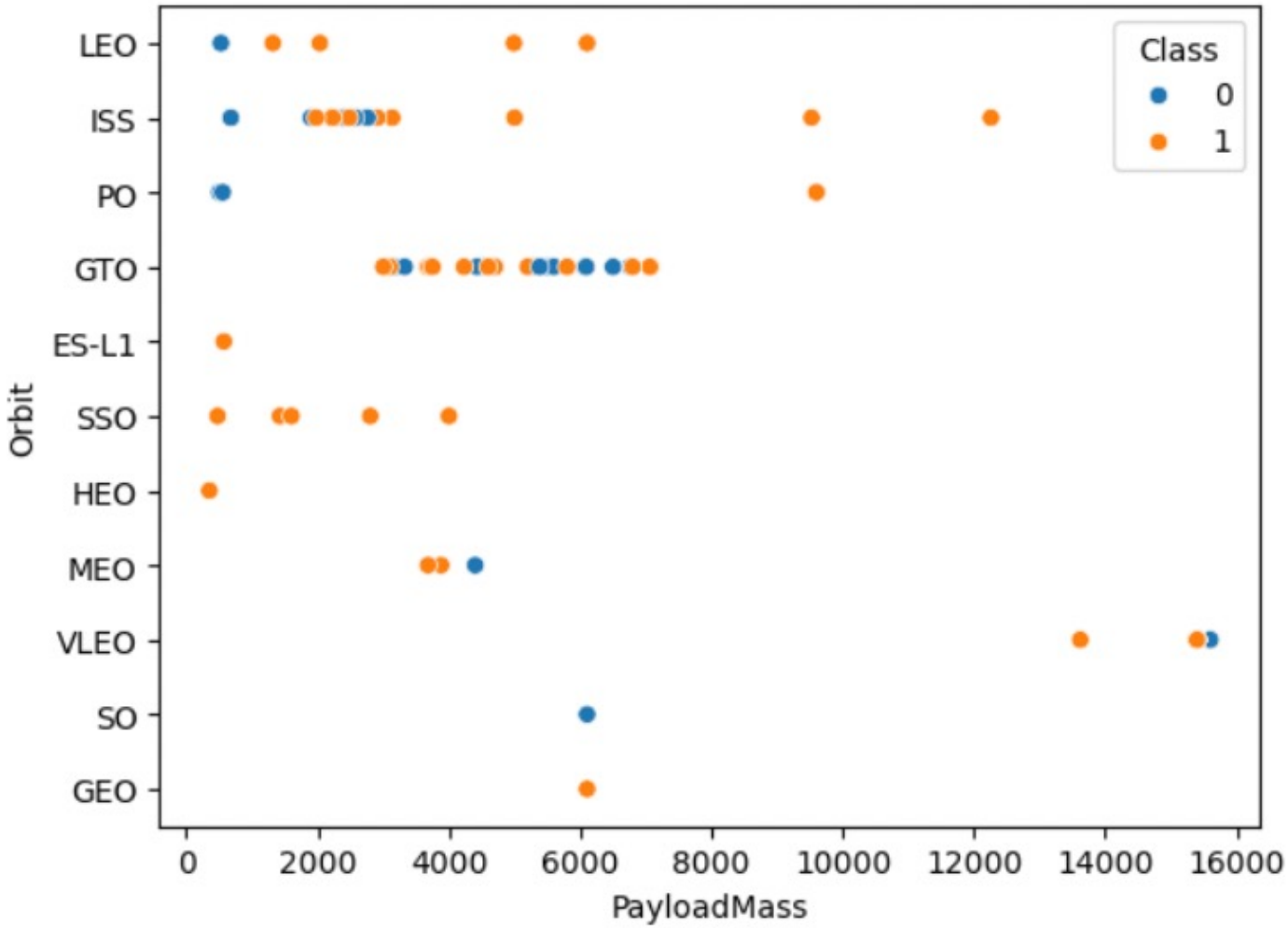
Flight Number vs. Orbit Type

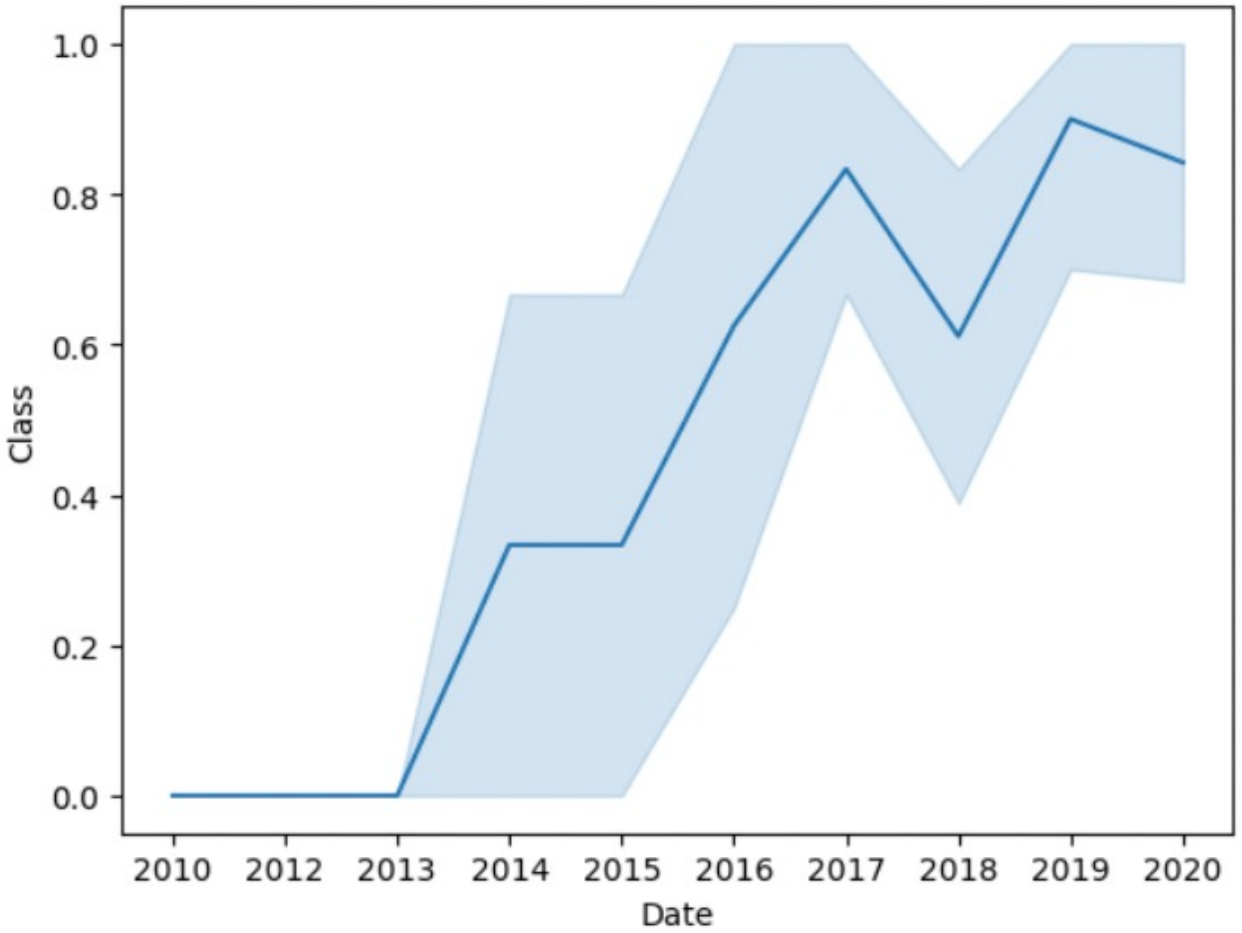
- Show a scatter point of Flight number vs. Orbit type

- We see a big failure rate in the lowest flight numbers, which means that with time SpaceX have gathered experience and have less failures.

Payload vs. Orbit Type

- Show a scatter point of payload vs. orbit type
- We can obtain information from this chart that the heaviest payloads went to the VLEO Orbit, starting from 13K to 16K, where the heaviest one is a failure.





Launch Success Yearly Trend

- Show a line chart of yearly average success rate
- After the year of 2013, we see a scaling success with the launching and landings of the SpaceX shuttles.

All Launch Site Names

Launch Site
CCAFS LC-40
VAFB SLC-4E
KSC LC-39A
CCAFS SLC-40

- Find the names of the unique launch sites
- %sql select distinct Launch_Site from SPACEXTABLE
- Selecting all the unique names from the Launch Site column in the SPACEXTABLE
- These are the four different locations which are used for shuttle landings

Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with `CCA`
- %sql select * from SPACEXTABLE where Launch_Site like 'CCA%' limit 5
- We select every column data from the table where the Launch Site name starts with CCA and limit the output to 5

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

- Calculate the total payload carried by boosters from NASA
- %sql select SUM(PAYLOAD_MASS__KG_) from SPACEXTABLE where Customer like 'NASA (CRS)'
- We select the sum of the payload in kg from the table where the Customer column equals 'NASA (CRS)'

SUM(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

- 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%'
- We select the average Payload mass in kg from the table where the Booster_Version column starts with 'F9 v1.1'

AVG(PAYLOAD_MASS__KG_)
2534.6666666666665

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad
- %sql select min(Date) from SPACEXTABLE where Landing_Outcome like 'Success%'
- We select the minimum Date from the table where the column Landing_Outcome starts with a string of 'Success'

min(Date)
2015-12-22

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
- %sql select Booster_Version from SPACEXTABLE where PAYLOAD_MASS__KG_ > 4000 and PAYLOAD_MASS__KG_ < 6000 and Landing_Outcome = 'Success (ground pad)’
- We select the Booster_Version column where the mass is above 4000 and below 6000, only for Landing_Outcome equal to 'Success (ground pad)’

Booster_Version
F9 FT B1032.1
F9 B4 B1040.1
F9 B4 B1043.1

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes
- %sql select count(Mission_Outcome) from SPACEXTABLE
- We select the total count of Mission Outcome rows in the table

count(Mission_Outcome)
101

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass
- %sql select Booster_Version from SPACEXTABLE where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEXTABLE)
- We select the Booster_Version where the Payload mass is the maximum of the table's Payload mass, using subquery

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

- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
- %sql select * from SPACEXTABLE where substr(Date,0,5) = '2015'
- We select every column where the Year Date is 2015

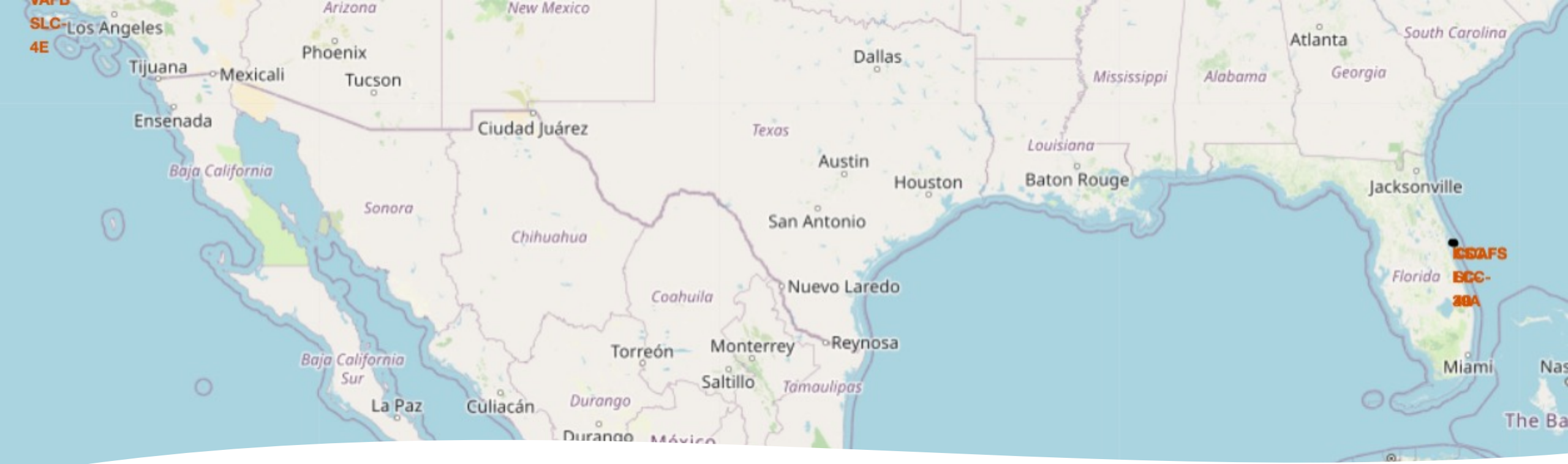
Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2015-10-01	09:47:00	F9 v1.1 B1012	CCAFS LC-40	SpaceX CRS-5	2395	LEO (ISS)	NASA (CRS)	Success	Failure (drone ship)
2015-11-02	23:03:00	F9 v1.1 B1013	CCAFS LC-40	DSCOVR	570	HEO	U.S. Air Force NASA NOAA	Success	Controlled (ocean)
2015-02-03	03:50:00	F9 v1.1 B1014	CCAFS LC-40	ABS-3A Eutelsat 115 West B	4159	GTO	ABS Eutelsat	Success	No attempt
2015-04-14	20:10:00	F9 v1.1 B1015	CCAFS LC-40	SpaceX CRS-6	1898	LEO (ISS)	NASA (CRS)	Success	Failure (drone ship)
2015-04-27	23:03:00	F9 v1.1 B1016	CCAFS LC-40	Turkmen 52 / MonacoSAT	4707	GTO	Turkmenistan National Space Agency	Success	No attempt
2015-06-28	14:21:00	F9 v1.1 B1018	CCAFS LC-40	SpaceX CRS-7	1952	LEO (ISS)	NASA (CRS)	Failure (in flight)	Precluded (drone ship)
2015-12-22	01:29:00	F9 FT B1019	CCAFS LC-40	OG2 Mission 2 11 Orbcomm-OG2 satellites	2034	LEO	Orbcomm	Success	Success (ground pad)

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
- %sql select Landing_Outcome from SPACEXTABLE where (Landing_Outcome = 'Failure (drone ship)' or Landing_Outcome = 'Success (ground pad)') and (Date > '2010-06-04' and Date < '2017-03-20')
- We select Landing_Outcome where it's either Failure or Success and the date is between 2010 and 2017

Landing_Outcome
Failure (drone ship)
Failure (drone ship)
Success (ground pad)
Failure (drone ship)
Failure (drone ship)
Failure (drone ship)
Success (ground pad)
Success (ground pad)
Success (ground pad)
Success (ground pad


Section 3
Launch Sites
Proximities Analysis



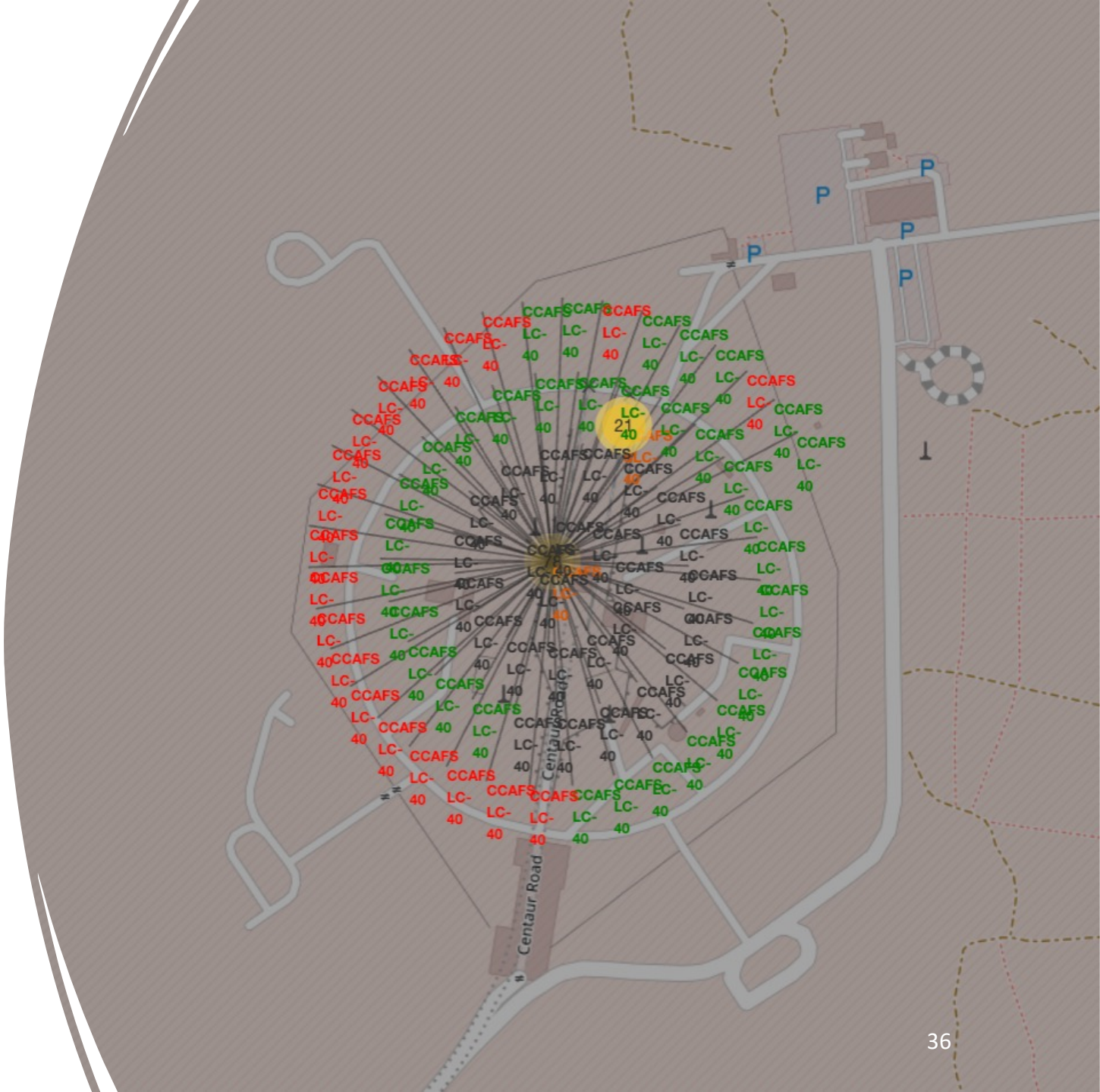
Explore Launch Sites

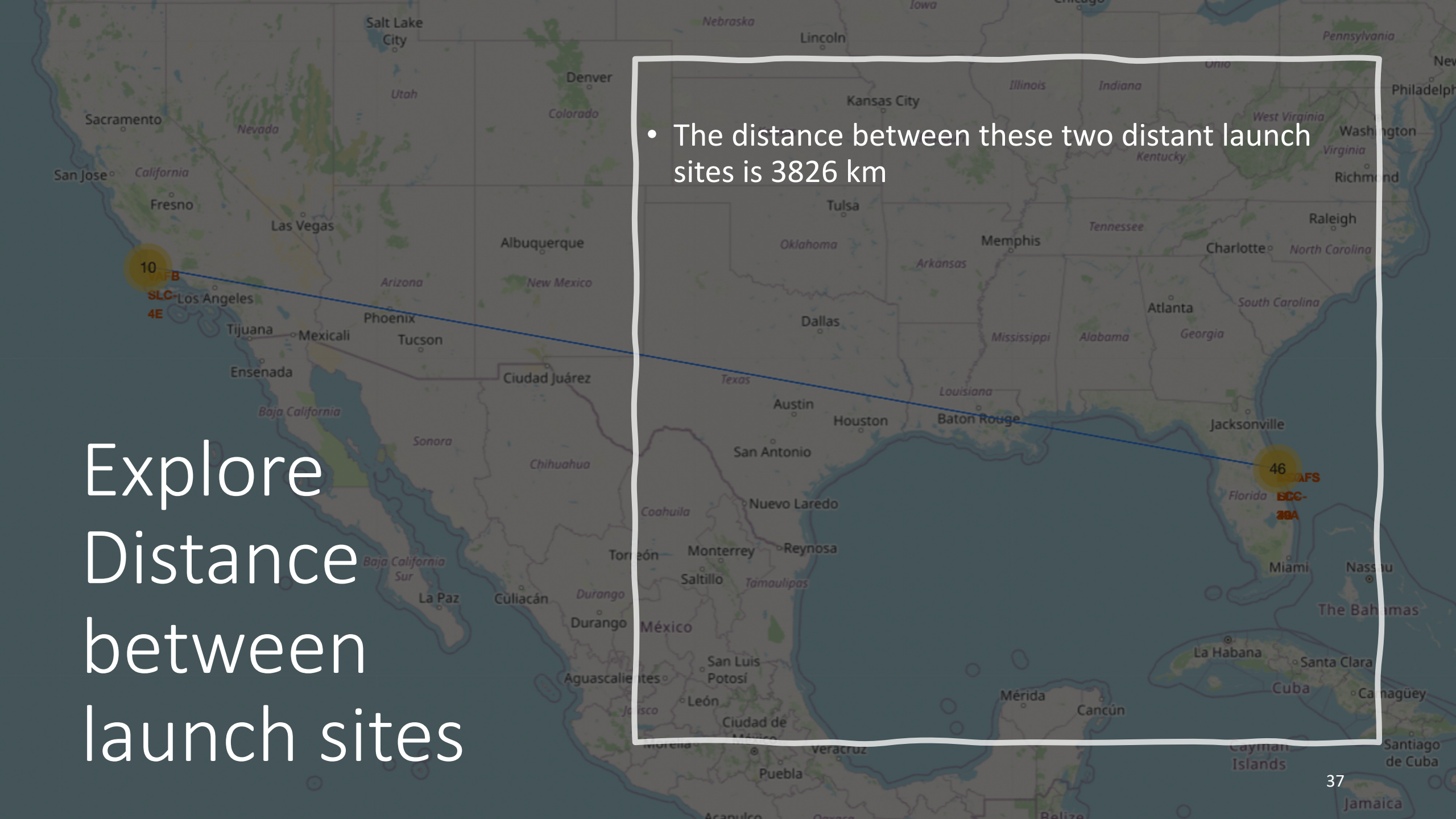
- Replace <Folium map screenshot 1> title with an appropriate title
- Explore the generated folium map and make a proper screenshot to include all launch sites' location markers on a global map
- Here we have the three different locations which are placed at the very East and West sea sides of the USA.

Color Labeled Launch Sites

- Replace <Folium map screenshot 2> title with an appropriate title

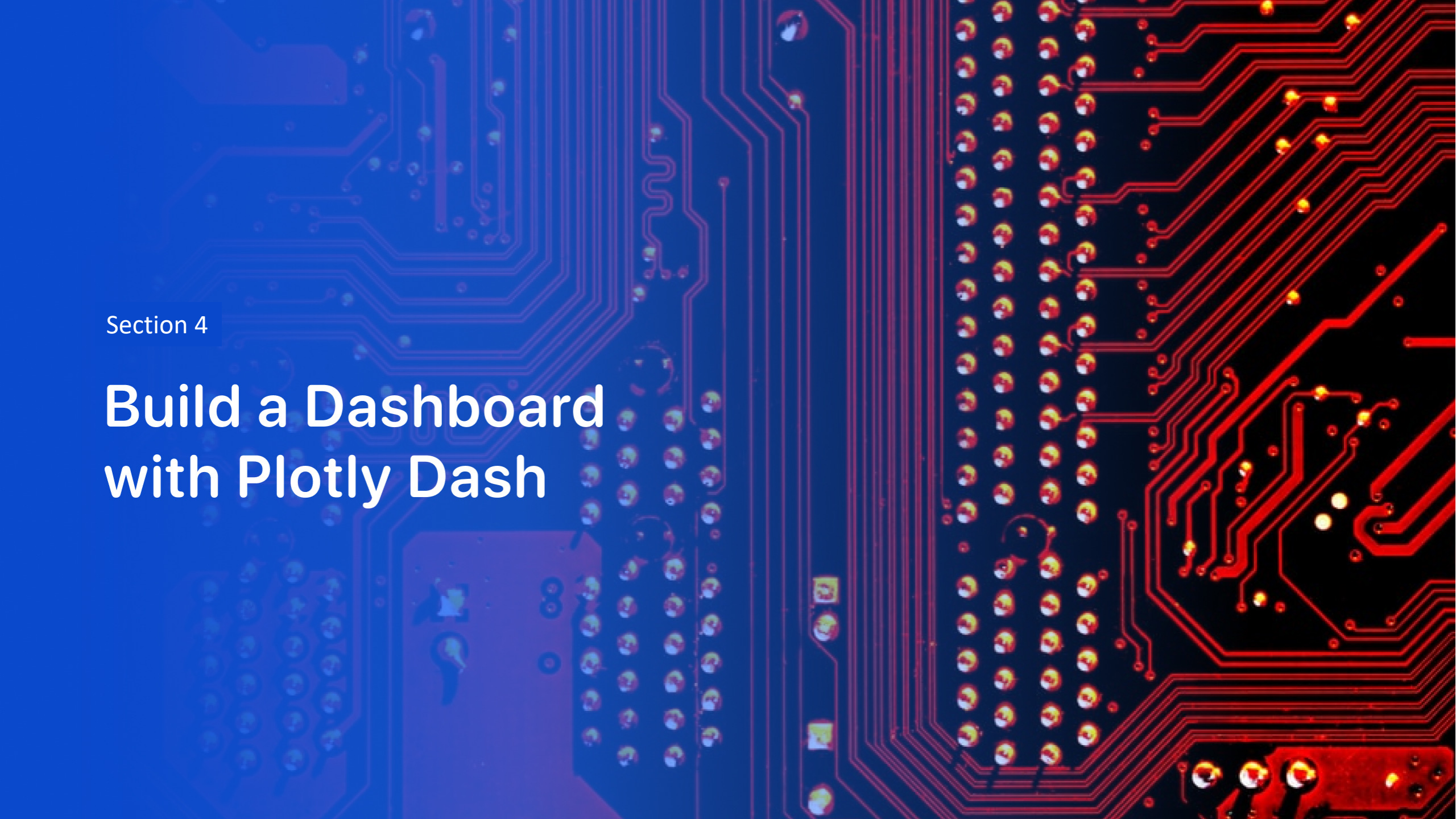
- Here we can obtain information of the successful and the failed launches for CCAFS LC-40 site





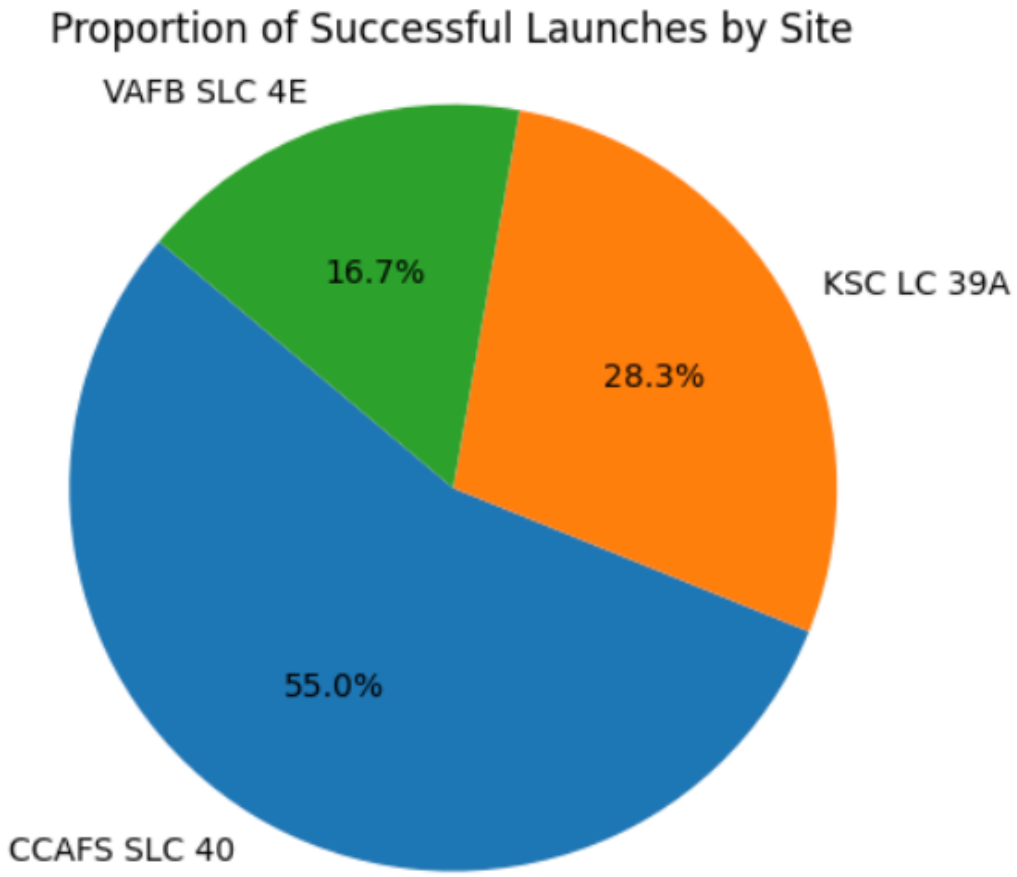
Explore Distance between launch sites

- The distance between these two distant launch sites is 3826 km


Section 4
Build a Dashboard with Plotly Dash

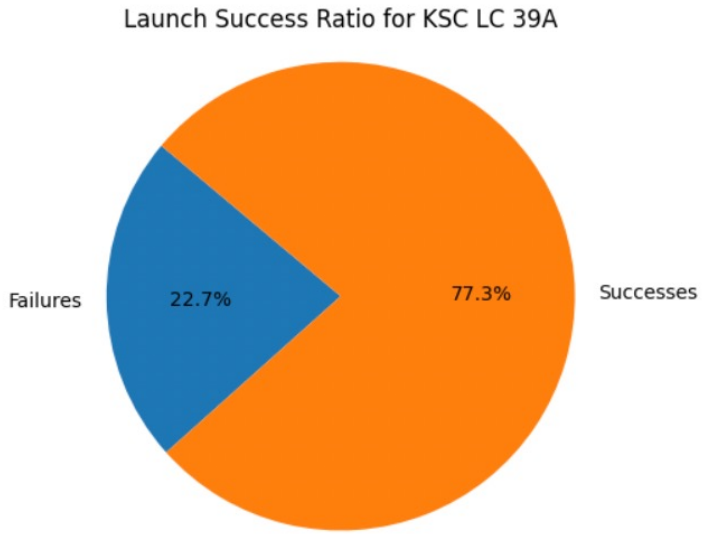
Proportions of successful Launches

- Replace <Dashboard screenshot 1> title with an appropriate title
- We see here that the most successful site is the CCAFS, followed by KSC and finally comes the VAFB



KSC LC 39A Launch Success Ratio

- Replace <Dashboard screenshot 2> title with an appropriate title
- Wee see that nearly ¾ of the launches are successful for the KSC launch site



Different Scatt

- At the first scatter we have a range between 2500 and 5000 kg. We see almost half of the launches

- The second scatter has the full range of payload 0 to 10000 where we can obtain more information

Section 5

Predictive Analysis (Classification)

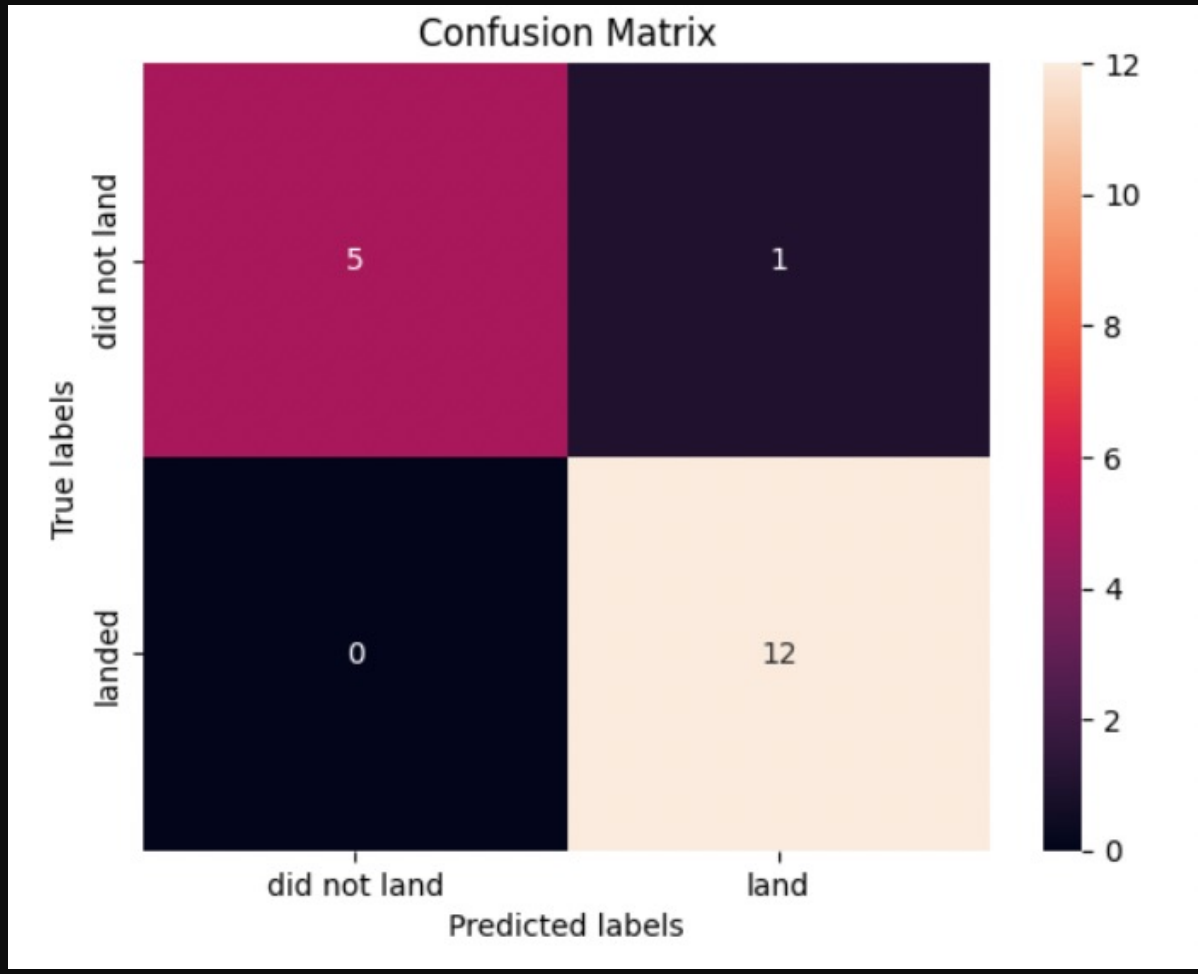
Classification Accuracy

- Visualize the built model accuracy for all built classification models, in a bar chart

- KNN and LogReg models have both equal high score

Confusion Matrix of KNN Model

- On the first row we have 5 True Positives and 1 False Negative
- On the second row we have 0 False Positives and 12 True Negatives.
- In overall we can obtain the score of the model by diving the success indexes by the all predictions, which means to divide 17/18 and we will obtain the 0.9444444 score of the model.



Conclusions

- Point 1: Among the experience gained and the rise of the Flight Number SpaceX has fewer failure attempts
- Point 2: The proximity to the ocean is strategic, allowing for safer launches and the potential to perform sea landings or tests. It mitigates risks to populated areas and facilitates the recovery of rocket stages via drone ships in the ocean, thus enabling reusability—a cornerstone of SpaceX's approach to reducing launch costs.
- Point 3: SpaceX has established a significant focus on reusability, as observed in its frequent use of reusable Falcon 9
- Point 4: According to the different orbits we obtained information that in different orbits, we have different success ratings, which may be useful for calculating the price for launching.

```python
# Create an app layout
app.layout = html.Div(children=[html.H1('SpaceX Launch Records Dashboard',
                                        style={'textAlign': 'center', 'color': '#503D36',
                                               '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=[
                                                 {'label': 'All Sites', 'value': 'ALL'},
                                                 {'label': 'CCAFS LC-40', 'value': 'CCAFS LC-40'},
                                                  {'label': 'VAFB SLC-4E', 'value': 'VAFB SLC-4E'},
                                                  {'label': 'KSC LC-39A', 'value': 'KSC LC-39A'},
                                             ],
                                              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, marks={0: '0', 2500: '2500', 5000: '5000', 7500: '7500', 10000: '10000'}, value=[min_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')),
                                ])

# TASK 2:
# Add a callback function for `site-dropdown` as input, `success-pie-chart` as output
@app.callback(Output(component_id='success-pie-chart', component_property='figure'),
              Input(component_id='site-dropdown', component_property='value'))
def pie_chart(site_dropdown):
    if site_dropdown == 'ALL':
        fig = px.pie(spacex_df, values='class', names='Launch Site', title='Pie Chart')
        return fig
    else:
        filtered_df = spacex_df[spacex_df['Launch Site'] == site_dropdown]
        filtered_df = filtered_df.groupby(['Launch Site', 'class']).size().reset_index(name='class count')
        fig = px.pie(filtered_df, values='class count', names='class', title='Pie Chart')
        return fig

# TASK 4:
# Add a callback function for `site-dropdown` and `payload-slider` as inputs, `success-payload-scatter-chart` as output
@app.callback(Output(component_id='success-payload-scatter-chart', component_property='figure'),
              [Input(component_id='site-dropdown', component_property='value'),
               Input(component_id="payload-slider", component_property="value")])
def scatter_chart(site_dropdown, payload_slider):
    if site_dropdown == 'ALL':
        fig = px.scatter(spacex_df, x='Payload Mass (kg)', y='class', color='Booster Version Category', title='Scatter Chart')
        return fig
    else:
        filtered_df = spacex_df[spacex_df['Launch Site'] == site_dropdown]
        filtered_df = filtered_df[(filtered_df['Payload Mass (kg)'] >= payload_slider[0]) & (filtered_df['Payload Mass (kg)'] <= payload_slider[1])]
        fig = px.scatter(filtered_df, x='Payload Mass (kg)', y='class', color='Booster Version Category', title='Scatter Chart')
        return fig
```

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!