IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Carlos Edgar Neves
24th September 2024

Outline

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

Executive Summary

- Summary of methodologies -
  - Data Collection Via API and Web Scrapping
  - Data Wrangling and Analysis with SQL and Visualization
  - Interactive Maps with Folium
  - interactive dashboard with the Plotly Dash
  - Machine Learning Predictive Analysis
- Summary of all results
  - Data Analysis with Interactive Visualization
  - Best Model for Predictive Analysis

Introduction

- Project background and context

  We will predict 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. Therefore if we can determine if the first stage will land, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch.

- Problems you want to find answers

  1. With what factors the rocket will land successfully?
  2. The effect of each relationship of rocket variables on outcome?
  3. Conditions which will aid SpaceX have to achieve the best results?

Section 1

Methodology

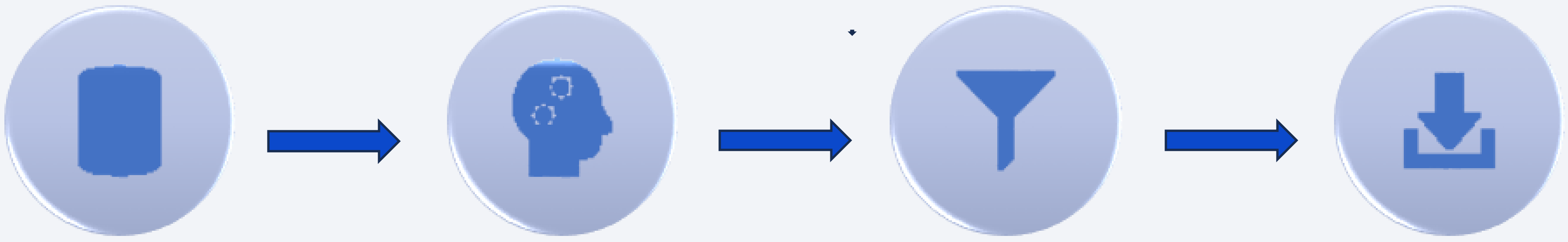
Methodology

Executive Summary

- Data collection methodology:
  - Via SpaceX Rest API
  - Web Scrapping from Wikipedia
- Perform data wrangling
  - One hot enconding data fields for machine Learning and dropping irrelevant columns (Transforming Data for Machine Learning)
  - Perform some Exploratory Data Analysis (EDA) to find some patterns in the data and determine what would be the label for training supervised models
- Perform exploratory data analysis (EDA) using visualization and SQL
  - Scatter and Bar Graphs to show relationship between data
- Perform interactive visual analytics
  - Using Folium and Plotly Dash Visualizations
- Perform predictive analysis using classification models
  - Build, tune, evaluate classification models

Data Collection

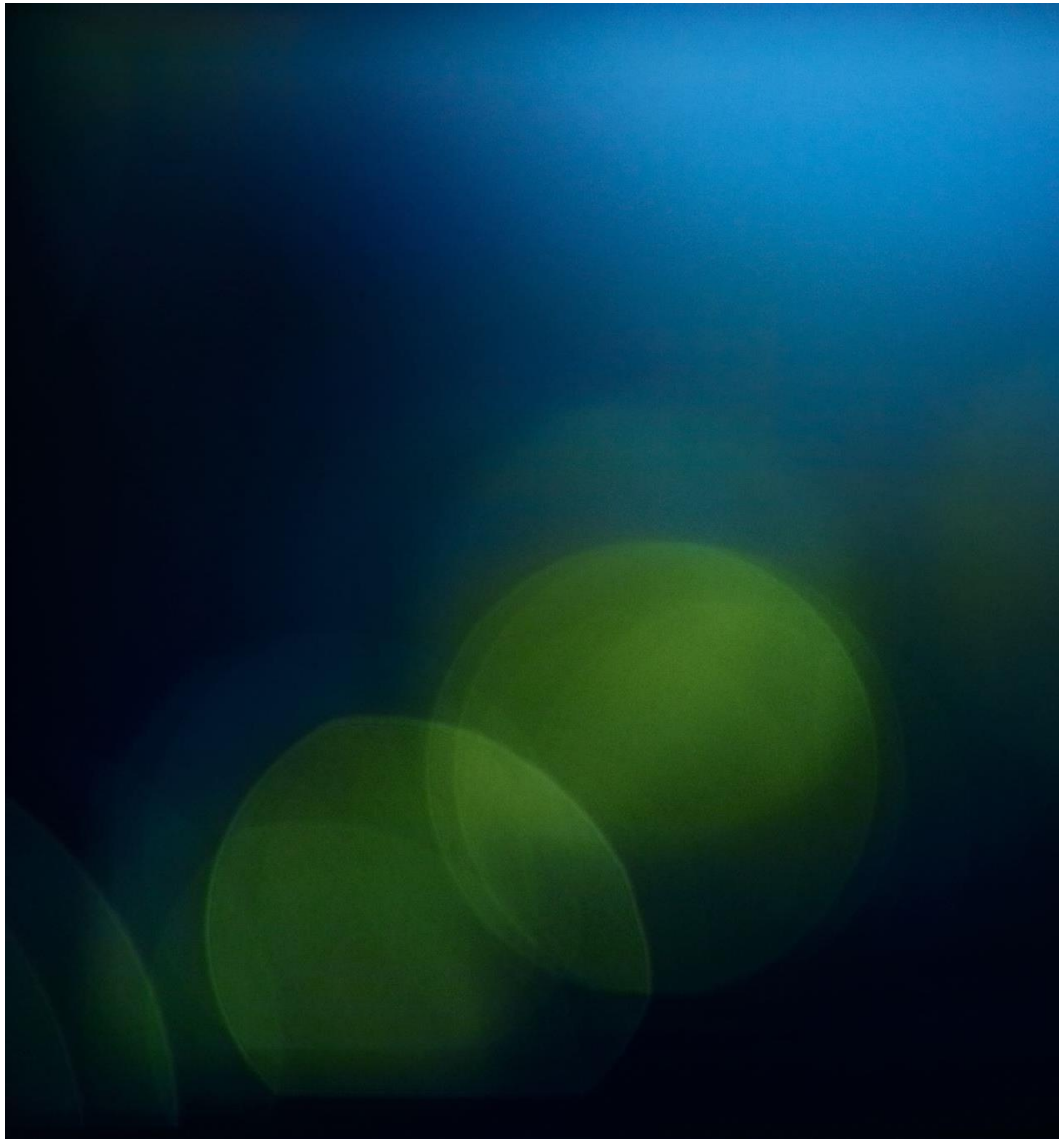
Data collection is the process of collecting and analyzing information on relevant variables in a predetermined, organized way so that one can respond to specific research questions, test hypotheses, and assess results.



1. GETTING DATA FROM THE API OR WEB PAGE
2. TRANSFORM INTO DATAFRAME
3. FILTER, CLEAN, ORGANIZE THE DATAFRAME AS REQUESTED
4. EXPORT THE DATA INTO A CSV FILE

Data Collection – SpaceX API

1. Getting Response from API
2. Converting response to a .json file
3. Apply get functions to get some features
4. Assign list to dictionary then create dataframe
5. Filter dataframe and export to CSV file



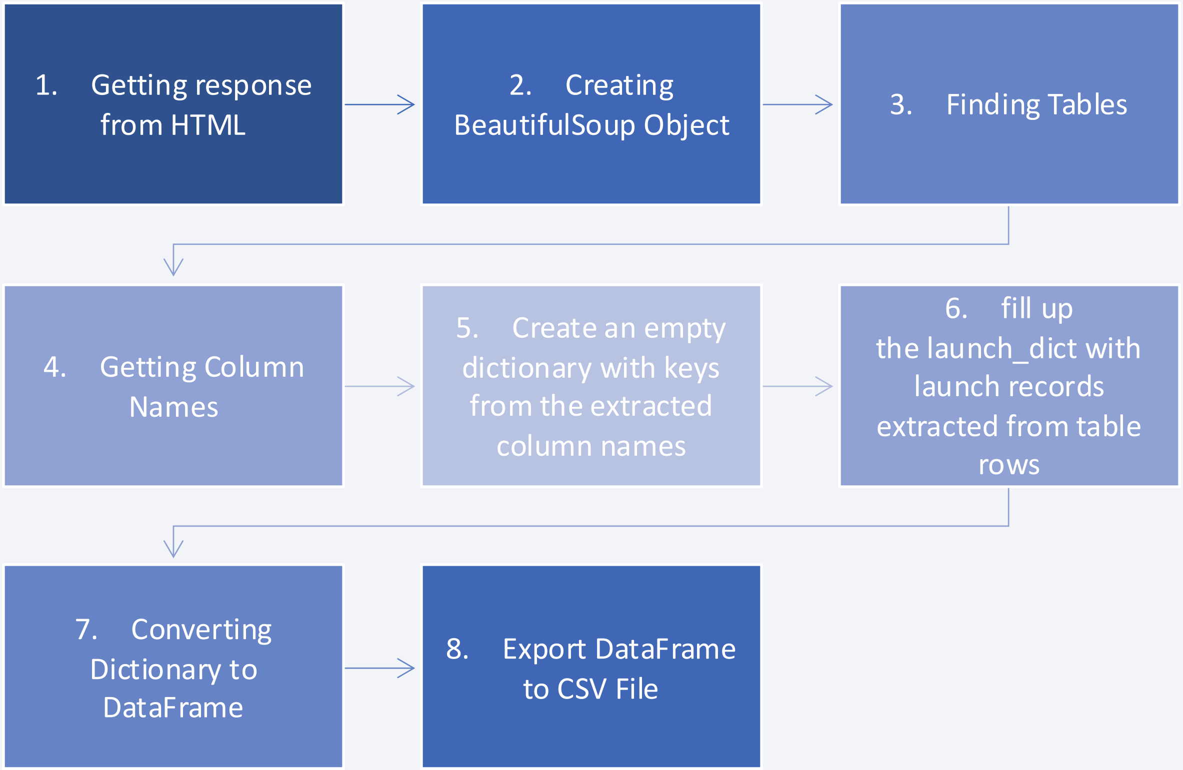
	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused
0	1	2006-03-24	Falcon 1	20.0	LEO	Kwajalein Atoll	None None	1	False	False
1	2	2007-03-21	Falcon 1	NaN	LEO	Kwajalein Atoll	None None	1	False	False
2	4	2008-09-28	Falcon 1	165.0	LEO	Kwajalein Atoll	None None	1	False	False
3	5	2009-07-13	Falcon 1	200.0	LEO	Kwajalein Atoll	None None	1	False	False

Data Collection – SpaceX API

- GitHub URL

https://github.com/CarlosEdgarNeves/Data-Science-and-Machine-Learning-Capstone-Project/blob/main/Week%201%3A%20(A)-jupyter-labs-spacex-data-collection-api%20(4).ipynb

Data Collection - Scraping



	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	24 January 2021	15:00
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	4 June 2010	18:45
3	3	CCAFS	SpaceX CRS-1	4,700 kg	LEO	NASA	Success\n	F9 v1.0B0005.1	No attempt	8 December 2010	15:43
4	4	CCAFS	SpaceX CRS-2	4,877 kg	LEO	NASA	Success\n	F9 v1.0B0006.1	No attempt\n	22 May 2012	07:44
...	...	...	...	...	...	...	...	...	...	...	...
117	117	KSC	Starlink	~14,000 kg	LEO	SpaceX	Success\n	F9 B5B1051.10	Success	4 May 2021	19:01
118	118	CCSFS	Starlink	15,600 kg	LEO	SpaceX	Success\n	F9 B5B1058.8	Success	9 May 2021	06:42
[illegible]	119	KSC	SpaceX CRS-22	3,328 kg	LEO	NASA	Success\n	F9 B5B1063.2	Success	15 May 2021	[illegible]
[illegible]	[illegible]	CCSFS	SXM-8	7,000 kg	GTO	Sirius XM	Success\n	F9 B5B1067.1	Success	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	NaN	NaN	NaN	NaN	NaN	[illegible]	[illegible]	[illegible]	[illegible]

Data Collection - Scraping

- GitHub URL

https://github.com/CarlosEdgarNeves/Data-Science-and-Machine-Learning-Capstone-Project/blob/main/Week%201%3A%20(B)%20jupyter-labs-webscraping.ipynb

Data Wrangling

- Data wrangling is the process of converting raw data into a usable form.
- Converting Those Outcomes into Training Labels where 1 means the Booster successfully landed, 0 means it was unsuccessfully.
  - [GitHub URL](https://github.com/CarlosEdgarNeves/Data-Science-and-Machine-Learning-Capstone-Project/blob/main/Week%201%3A%20(C)%20labs-jupyter-spacex-Data%20wrangling.ipynb)

https://github.com/CarlosEdgarNeves/Data-Science-and-Machine-Learning-Capstone-Project/blob/main/Week%201%3A%20(C)%20labs-jupyter-spacex-Data%20wrangling.ipynb

1. Load CSV
2. Make DataFrame
3. Cleaning Data
4. Simplflying to Boolean Values
5. Export to CSV File

Data Wrangling

1. Calculate number of Launches at each site
2. Calculate number and occurrence of each Orbit
3. Calculate number and occurrence of mission outcome per orbit type
4. Create landing outcome label from outcome column
5. Export Data as CSV

EDA with Data Visualization

- Data exploration refers to the initial step in data analysis in which data analysts use data visualization and statistical techniques to describe dataset characterizations, such as size, quantity, and accuracy, in order to better understand the nature of the data.

1. Load Data
2. Make Dataframe from it
3. Create Visualizations
4. Collect Insights

EDA with Data Visualization

Scatter Graphs Drawn:

- Payload and Flight Number
- Flight Number and Launch Site
- Payload and Launch Site
- Flight Number and Orbit Type
- Payload and Orbit Type

A scatterplot is a graph of data points that might follow a pattern or might not; this pattern is called a correlation or relationship between the variables. The higher the correlation between the two variables, or the stronger the relationship.

Bar Graph Drawn:

- Success Rate VS Orbit Type

A bar chart or bar graph is a chart or graph that presents categorical data with rectangular bars with heights or lengths proportional to the values that they represent.

GitHub URL

https://github.com/CarlosEdgarNeves/Data-Science-and-Machine-Learning-Capstone-Project/blob/main/Week%202%3A%20(B)%20jupyter-labs-eda-dataviz.ipynb

EDA with SQL

SQL is an indispesnsable tool for Data Scientists and analysts as most of the real-world data is stored in databases. It's not only the Standard Language for Relational Database operations, but also an incredible powerful tool for analyzing data and drawing useful insights from it.

GitHub URL
https://github.com/CarlosEdgarNeves/Data-Science-and-Machine-Learning-Capstone-Project/blob/main/Week%202%3A%20(A)%20jupyter-labs-eda-sql-edx_sqllite.ipynb

We performed SQL queries to gather information from given dataset:

- Display the names of the unique launch sites in the space mission
- Display 5 records where launch sites begin with the string 'KSC'
- Display the total payload mass carried by boosters launched by NASA (CRS)
- Display average payload mass carried by booster version F9 v1.1
- List the date where the succesful landing outcome in drone ship was acheived.
- List the names of the boosters which have success in ground pad and have payload mass greater than 4000 but less than 6000
- List the names of the booster_versions which have carried the maximum payload mass. Use a subquery
- List the records which will display the month names, succesful landing_outcomes in ground pad ,booster versions, launch_site for the months in year 2017
- 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

Build an Interactive Map with Folium

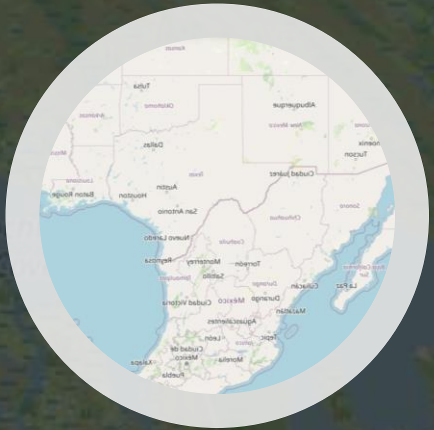
Folium makes it easy to visualize data that's been manipulated in Python on an interactive leaflet map. It enables both the binding of data to a map for choropleth visualizations as well as passing rich vector/raster/HTML visualizations as markers on the map.

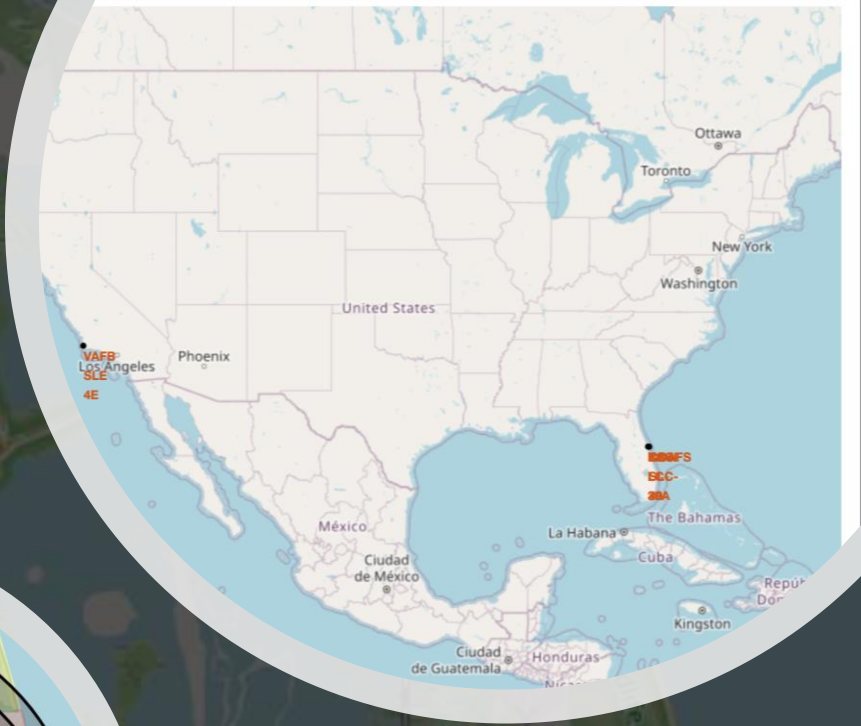
Map Objects	Code	Results
Map Marker	folium.Marker(	Map object to make a mark on map
Icon Marker	folium.icon(	Creat an Icon on Map
Circle Marker	folium.circle(	Create a circle where map is being placed
MousePosition	MousePosition(	Get coordinate for a mouse over a point on the map
Polyline	folium.Polyline(	Create a line between points
Marker Cluster Object	MarkerCluster(	This is a good way to simplify a map containing many markers having the same coordinates

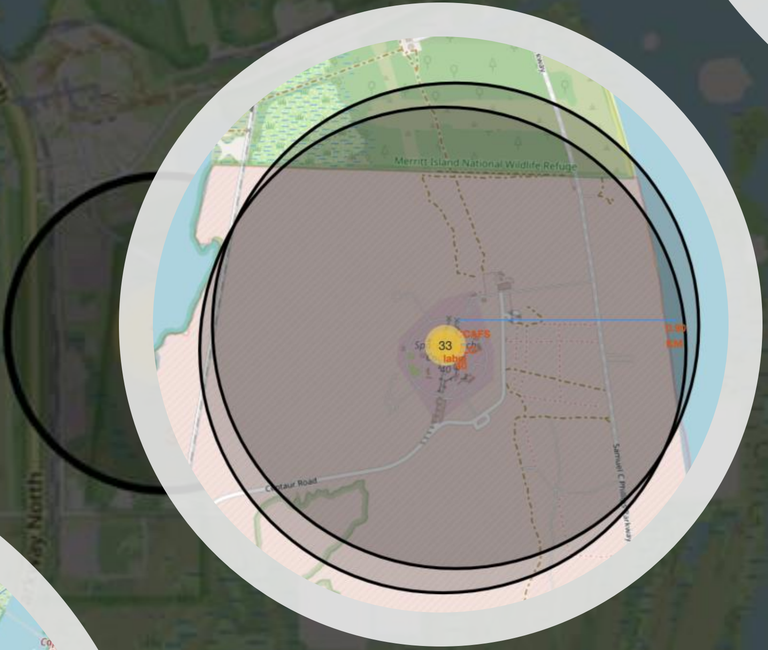
GitHub URL

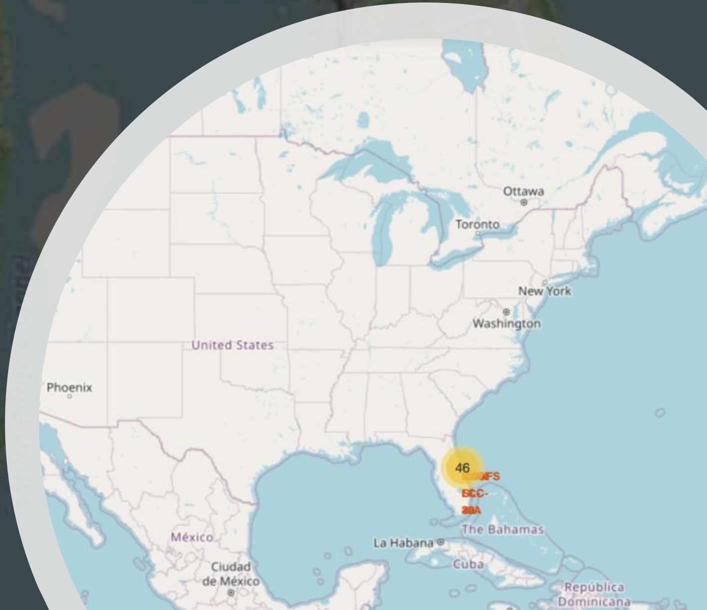
https//github.com/CarlosEdgarNeves/Data-Science-and-Machine-Learning-Capstone-Project/blob/main/Week%203%3A%20(A)%20Folium%20lab_jupyter_launch_site_location.jupyterlite.ipynb

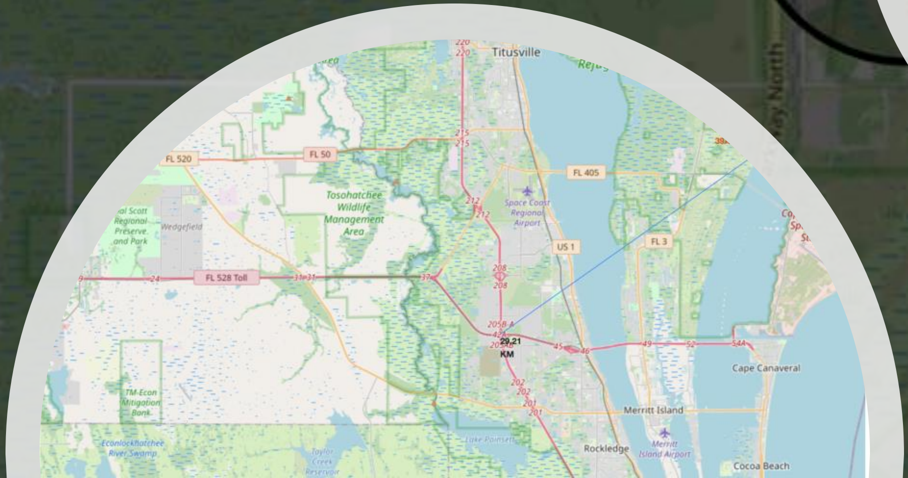
Build an Interactive Map with Folium











Build a Dashboard with Plotly Dash

Pie Chart showing the total success of all launch sites or each launch site

- Percentage of success for all launch sites or percentage of success for certain launch site

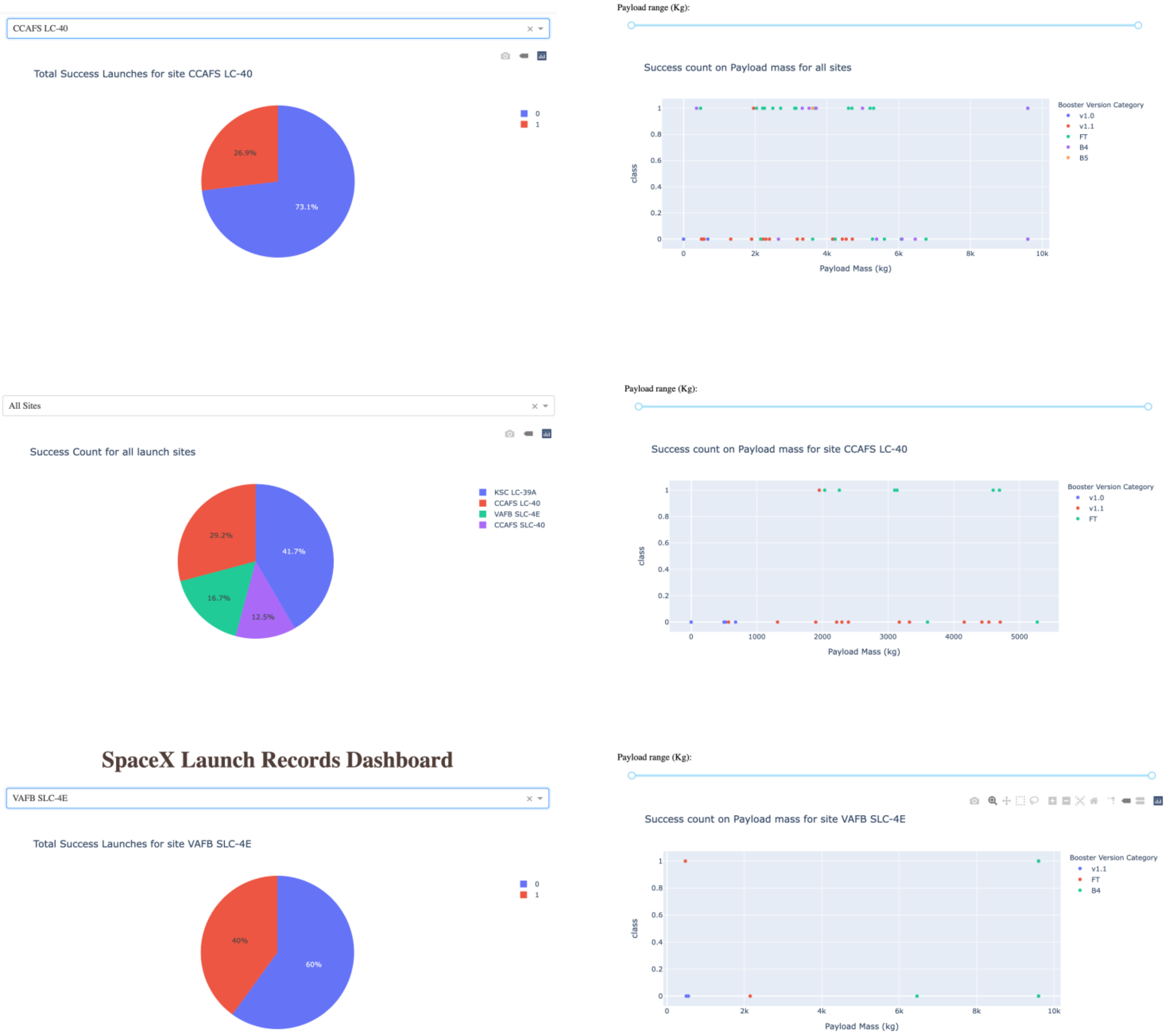
Scatter graph showing the correlation between the payload mass and Success for all sites or certain launch site

- Relationship between success rate on Payload Mass for all launch sites or for certain launch site

Map Objects	**Code**	**Results**
Dash and Components	Import Dash Import Dash_html components as html Import Dash_core components as dcc From dash.dependencies import Input, Output	With Dash Spen source, Dash apps run on your local laptop or server. The Dash Core component library contains a set of higher-level components like sliders, graphs, dropdwns, tables and more.
IPandas	Import pandas as pd	Fetching values from CSV and creating a Dataframe
Plotly	Import plotly.express as px	Plot the Graphs with interactive Plotly library
Dropdown	Dcc.dropdown(	Create a dropdown for launch sites
Rangeslider	Dcc.RangeSlider(	Create a RangeSlider for Payload Mass range selection
Pie Chart	px.pie(	Create the Pie graph for Success percentage display
Scatter Chart	px.Scatter(	Create the Scatter graph for correlation display

Build a Dashboard with Plotly Dash

[GitHub URL](https://github.com/CarlosEdgarNeves/Data-Science-and-Machine-Learning-Capstone-Project/blob/main/Week%203%3A%20(B)%20Build%20an%20Interactive%20Dashboard%20with%20Ploty%20Dash.py)
https://github.com/CarlosEdgarNeves/Data-Science-and-Machine-Learning-Capstone-Project/blob/main/Week%203%3A%20(B)%20Build%20an%20Interactive%20Dashboard%20with%20Ploty%20Dash.py



Predictive Analysis (Classification)

- Classification is a supervised machine learning method where the model tries to predict the correct label of a given input data. In classification, the model is fully trained using the training data, and then it is evaluated on test data before being used to perform prediction on new unseen data.

GitHub URL

- https://github.com/CarlosEdgarNeves/Data-Science-and-Machine-Learning-Capstone-Project/blob/main/Week%204%3A%20(A)%20SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

Predictive Analysis (Classification)

Building Model

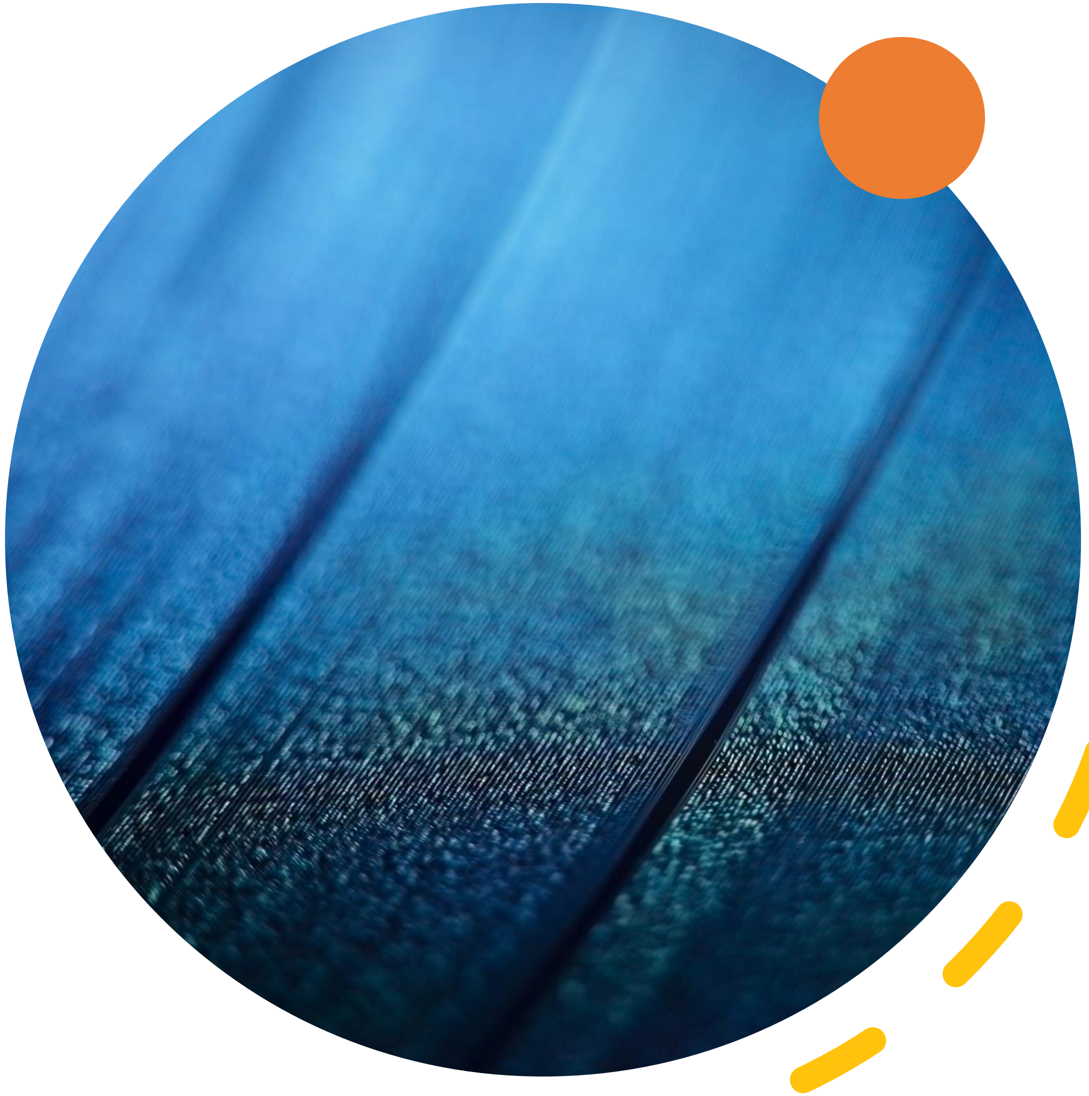
1. Load our feature engineered data into dataframe
2. Transform it into Numpy arrays
3. Standardize and transform data
4. Split data into training and test data sets
5. Check how many test samples have been created
6. List down Machine Learning Algorithms we want to use
7. Set our parameters and algorithms to GridSearchCV
8. Fit our datasets into the GridSearchCV objects and Train the model

Evaluating Model

1. Check accuracy of each Model
2. Get best Hyperparameters for each type of algorithm
3. Plot Confusion Matrix

Finding best performing Classification Model

1. The model with best accuracy score wins the Best Performing Model



Results

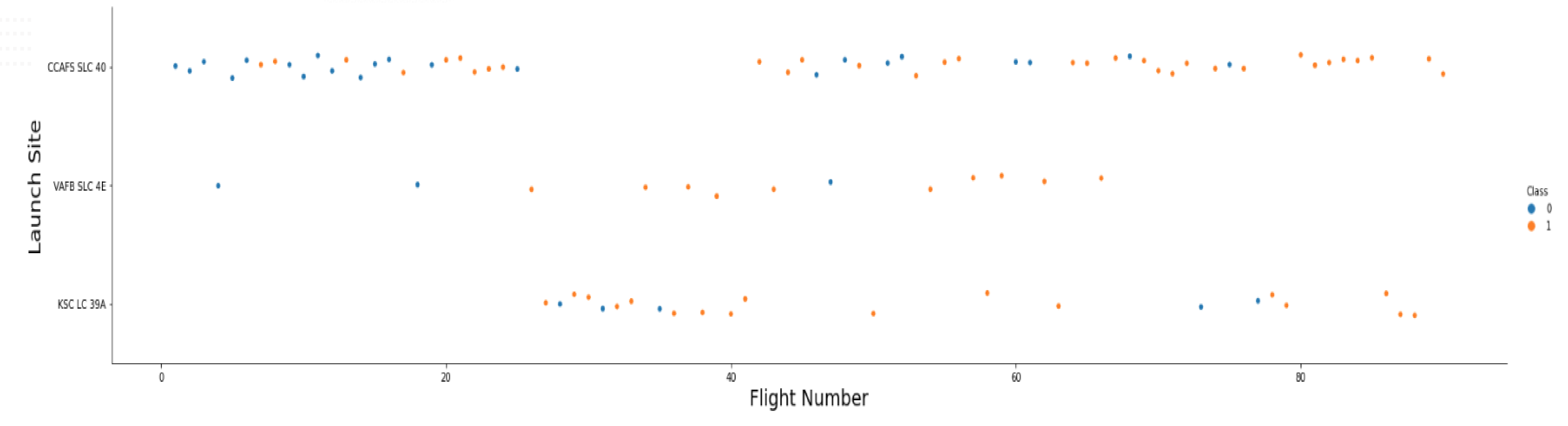
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

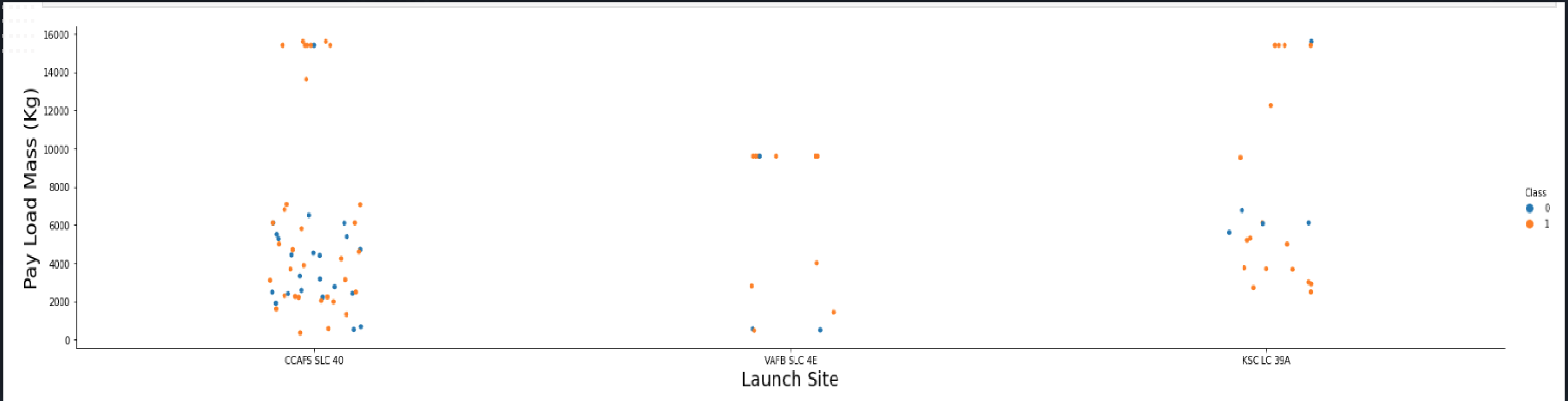
Flight Number vs. Launch Site

- With the increase of the Flight Number the success rate increases for all Launch Sites



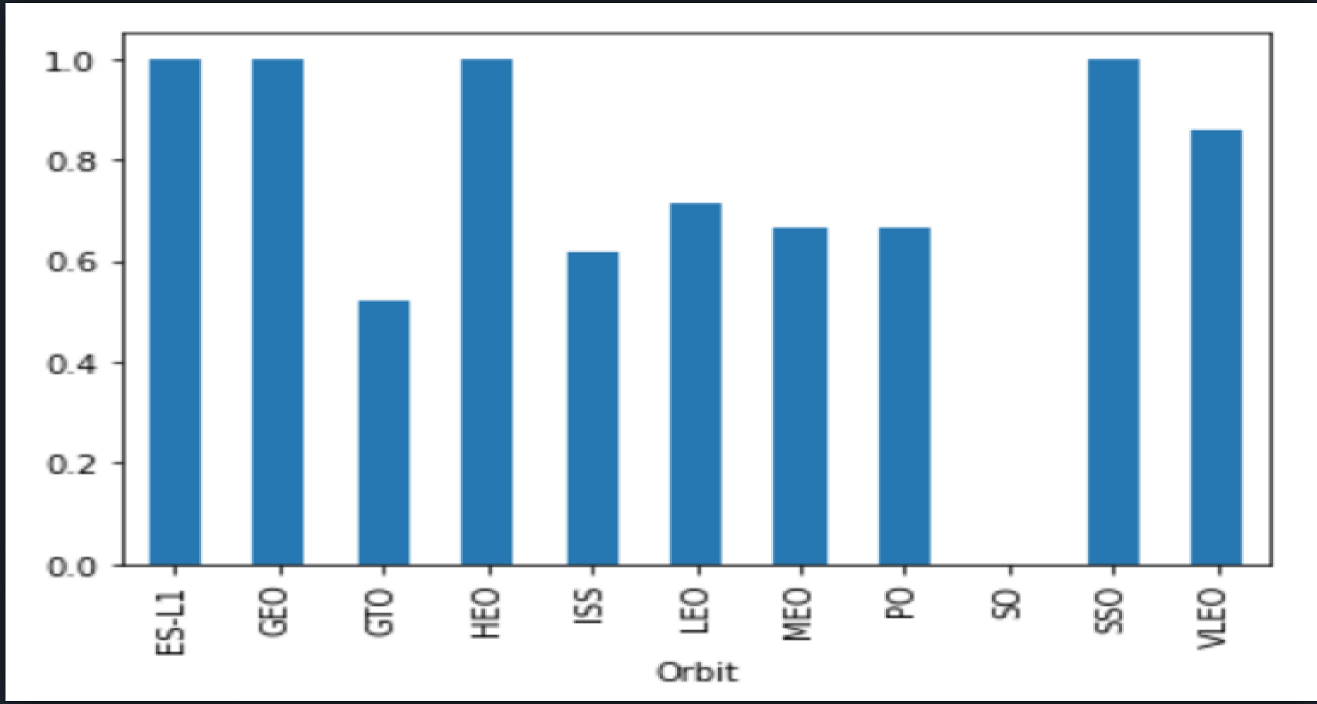
Payload vs. Launch Site

- The greater the Payload Mass (Greater than 8K) the success rate increases.



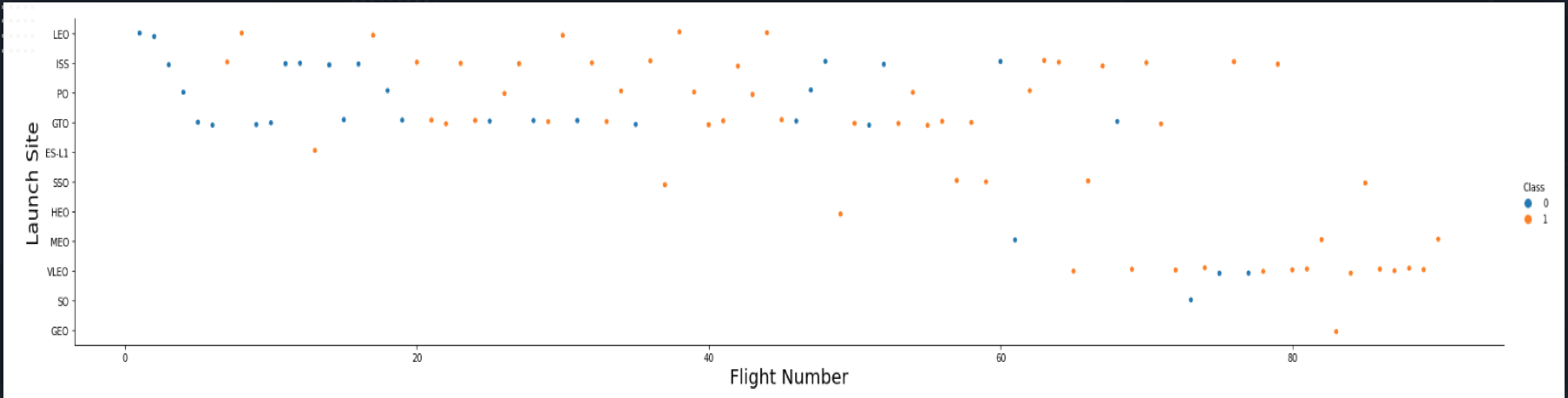
Success Rate vs. Orbit Type

- ES-L1, GEO, HEO and SSO are the Orbits with higher success rate for the rocket.



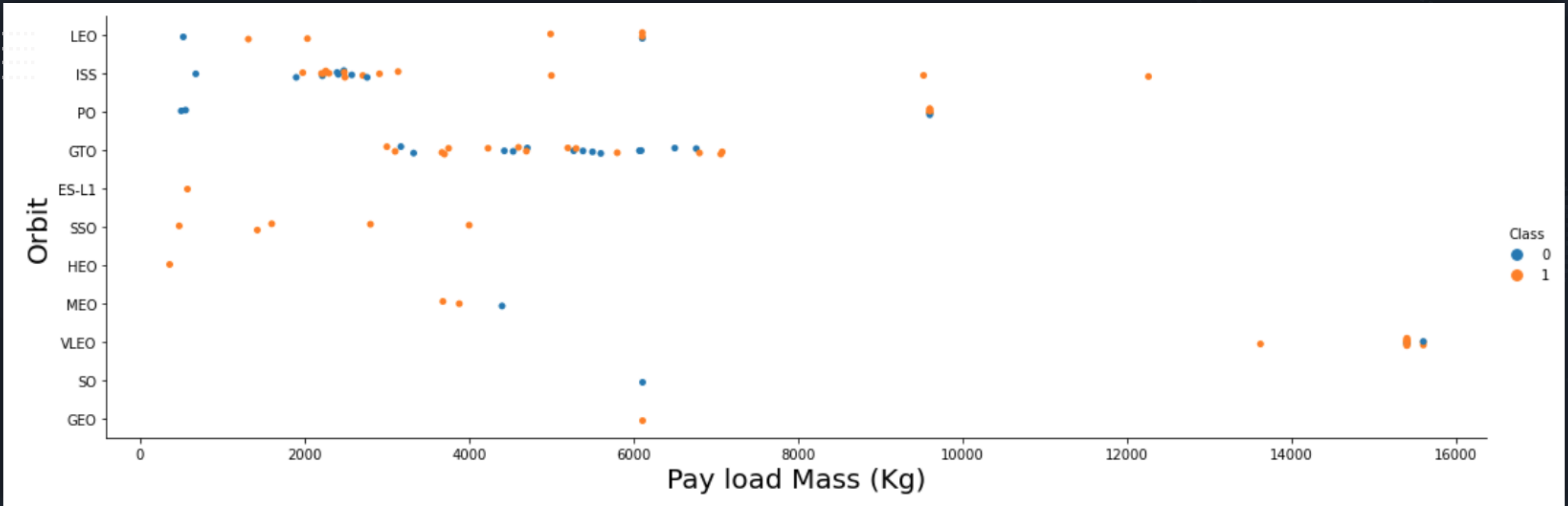
Flight Number vs. Orbit Type

- The LEO orbit the Success appears related to the number of flights.
- on the other hand, there seems to be no relationship between flight number when in GTO orbit.



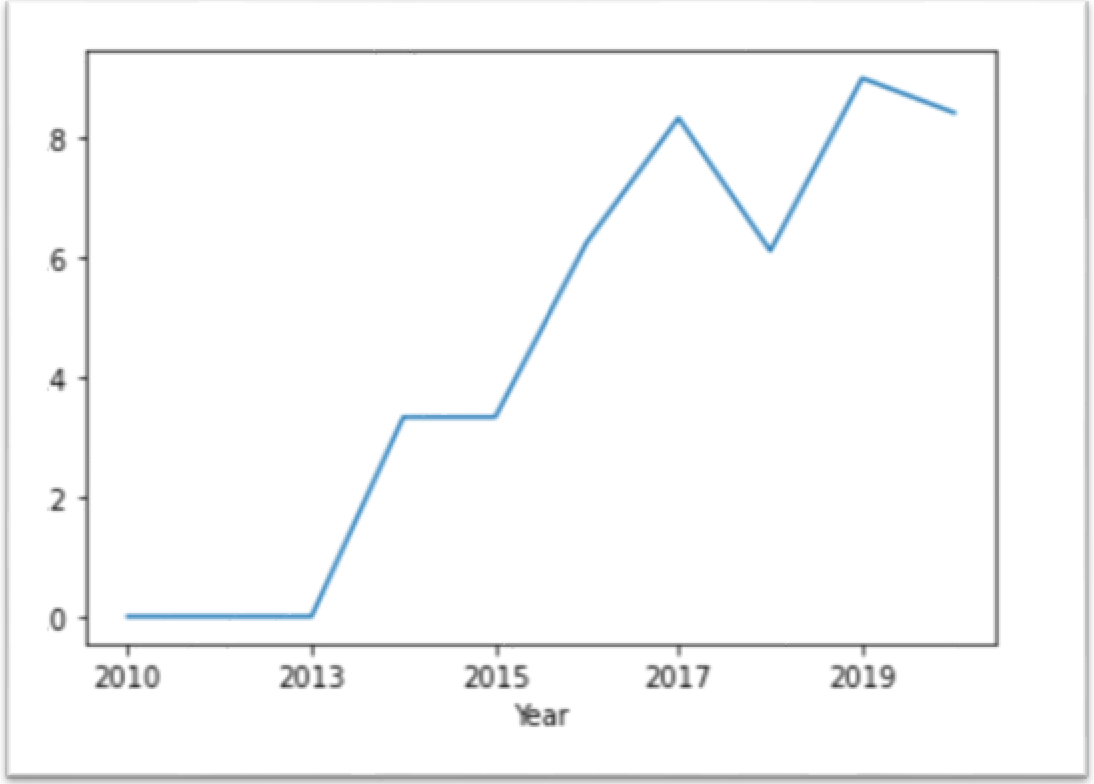
Payload vs. Orbit Type

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



Launch Success Yearly Trend

- We can observe that the success rate since 2013 kept increasing till 2019 (stable in 2014) and after 2015 it started increasing (dip from 2017 to 2018), and a slight dip after 2019.



All Launch Site Names

- The SQL DISTINCT keyword is used to retrieve unique values from a specified column or set of columns in a database table. It eliminates duplicate records, ensuring that only distinct, non-repeated values are returned from the column Launch_Sites from SPACEXTBL table.

```
%sql select distinct LAUNCH_SITE as "Launch_Sites" from SPACEXTBL;
```

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

Launch_Sites
CCAFS LC-40
VAFB SLC-4E
KSC LC-39A
CCAFS SLC-40

Launch Site Names Begin with 'KSC'

- The LIKE operator is used in a WHERE clause to search for a specified pattern in a column in this cause from the column Launch_Site all words starting with "KSC" will be fetched, and Using keyword "LIMIT 5" we fetch 5 records from table SPACEXTBL.

```
%sql SELECT * from "SPACEXTBL" WHERE LAUNCH_SITE LIKE "KSC%" LIMIT 5;
```

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_
2017-02-19	14:39:00	F9 FT B1031.1	KSC LC-39A	SpaceX CRS-10	2490
2017-03-16	6:00:00	F9 FT B1030	KSC LC-39A	EchoStar 23	5600
2017-03-30	22:27:00	F9 FT B1021.2	KSC LC-39A	SES-10	5300
2017-05-01	11:15:00	F9 FT B1032.1	KSC LC-39A	NROL-76	5300
2017-05-15	23:21:00	F9 FT B1034	KSC LC-39A	Inmarsat-5 F4	6070

Total Payload Mass

- The function Sum calculates the Total Sum of the column Payload_MASS_KG_, and WHERE clause it filters the data to fetch Customers with the name "NASA (CRS)".

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) as "Total Payload Mass (Kg)", Customer from "SPACEXTBL" \
WHERE Customer = "NASA (CRS)";
```

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

Total Payload Mass (Kg)	Customer
45596	NASA (CRS)

Average Payload Mass by F9 v1.1

- The LIKE operator is **used in a WHERE clause to search for a specified pattern in a column in this case from the column** Booster_Version **all words starting with "F9 v1.1" will be fetched, and** the function AVG calculates the Average of the column Payload_MASS_KG_ on the Booster_Version starting with **"F9 v1.1"**.

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) as "Payload Mass Kgs", Booster_Version FROM 'SPACEXTBL' \
WHERE Booster_Version LIKE 'F9 v1.1%';

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

Payload Mass Kgs	Booster_Version
2534.6666666666665	F9 v1.1 B1003

First Successful Ground Landing Date

- Using the MIN function, returns the minimum date of the column Date and the WHERE clause filters the data to only perform calculations on the column Landing_Outcome where the succesful landing outcome in drone ship was acheived.

```
%sql SELECT MIN(DATE) FROM "SPACEXTBL" WHERE Landing_Outcome = "Success (drone ship)";

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

MIN(DATE)
2016-04-08

Successful Ground Pad Landing with Payload between 4000 and 6000

- Where clause filters the boosters which have success in ground pad and AND clause filters payload of mass greater than 4000 but less than 6000.

```
%sql SELECT BOOSTER_VERSION, PAYLOAD_MASS__KG_, LANDING_OUTCOME from "SPACEXTBL" \
WHERE LANDING_OUTCOME = "Success (ground pad)" and PAYLOAD_MASS__KG_ > 4000 and PAYLOAD_MASS__KG_ < 6000;
```

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

Booster_Version	PAYLOAD_MASS__KG_	Landing_Outcome
F9 FT B1032.1	5300	Success (ground pad)
F9 B4 B1040.1	4990	Success (ground pad)
F9 B4 B1043.1	5000	Success (ground pad)

Total Number of Successful and Failure Mission Outcomes

- The Function Count calculates the total number of Mission Outcomes from the column Mission_Outcome.
- The function GROUP BY groups all the rows (Success or Failure) with the same column (Mission_Outcome) value.

```
%sql SELECT MISSION_OUTCOME, COUNT("MISSION_OUTCOME") as TOTAL from "SPACEXTBL" GROUP BY "MISSION_OUTCOME";
```

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

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

Boosters Carried Maximum Payload

- Using the Function MAX gives out the maximum PAYLOAD in the column PAYLOAD_MASS_KG_ in subquery.
- WHERE clause filters the Booster Version and Payload which have the maximum Payload Mass.

Booster_Version	Payload	PAYLOAD_MASS__KG_
F9 B5 B1048.4	Starlink 1 v1.0, SpaceX CRS-19	15600
F9 B5 B1049.4	Starlink 2 v1.0, Crew Dragon in-flight abort test	15600
F9 B5 B1051.3	Starlink 3 v1.0, Starlink 4 v1.0	15600
F9 B5 B1056.4	Starlink 4 v1.0, SpaceX CRS-20	15600
F9 B5 B1048.5	Starlink 5 v1.0, Starlink 6 v1.0	15600
F9 B5 B1051.4	Starlink 6 v1.0, Crew Dragon Demo-2	15600
F9 B5 B1049.5	Starlink 7 v1.0, Starlink 8 v1.0	15600

2015 Launch Records

- The SUBSTR() function extracts a substring from a string (starting at any position).
- We use substr(Date,6,2) for month, substr(Date,9,2) for date, substr(Date,0,5),='2017' for year, because SQLLite does not support monthnames. Then list the records which will display the month names, succesful landing_outcomes in ground pad ,booster versions, launch_site for the months in year 2017

```
%sql SELECT SUBSTR(Date, 6, 2) AS month, substr(Date, 9, 2) as Date, Booster_Version, Launch_Site, \
Landing_Outcome FROM SPACEXTBL WHERE SUBSTR(Date, 0, 5) = '2017' AND Landing_Outcome = 'Success (ground pad)';
```

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

month	Date	Booster_Version	Launch_Site	Landing_Outcome
02	19	F9 FT B1031.1	KSC LC-39A	Success (ground pad)
05	01	F9 FT B1032.1	KSC LC-39A	Success (ground pad)
06	03	F9 FT B1035.1	KSC LC-39A	Success (ground pad)
08	14	F9 B4 B1039.1	KSC LC-39A	Success (ground pad)
09	07	F9 B4 B1040.1	KSC LC-39A	Success (ground pad)
12	15	F9 FT B1035.2	CCAFS SLC-40	Success (ground pad)

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

- Count Function counts the total number of Landing Outcomes.
- WHERE clause filters data with DATE BETWEEN "2010-06-04" AND "2017-03-20".
- The function GROUP BY groups all the rows with the same column (Landing_Outcome) value.
- ORDER BY (Landing_Outcome) in descending order.

```
%sql SELECT Landing_Outcome, COUNT(*) as Outcome_Count FROM SPACEXTBL WHERE Date BETWEEN '2010-06-04' \
AND '2017-03-20' GROUP BY Landing_Outcome ORDER BY Outcome_Count DESC;
```

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

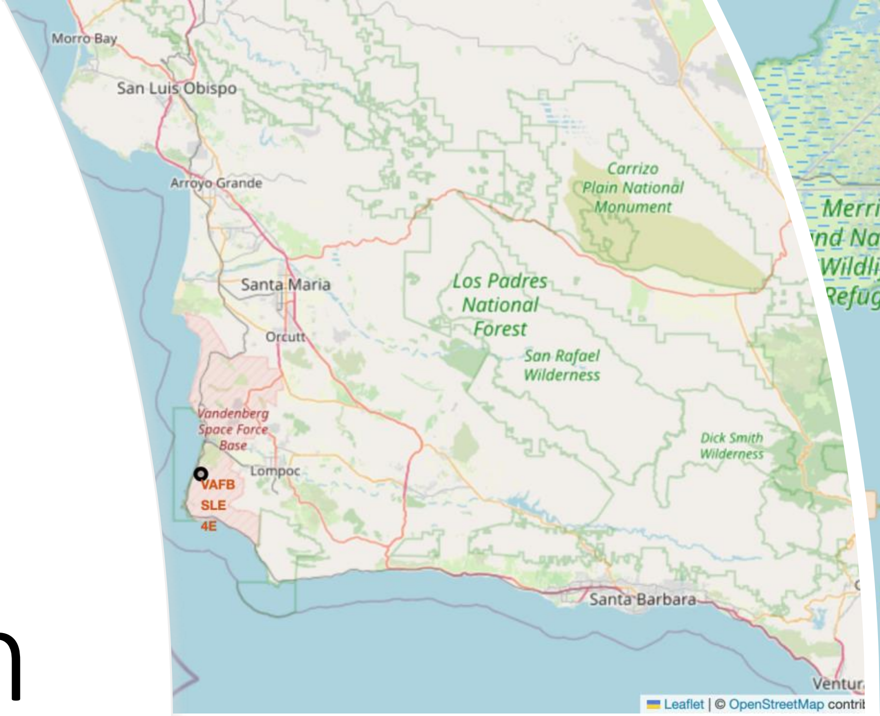
Landing_Outcome	Outcome_Count
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

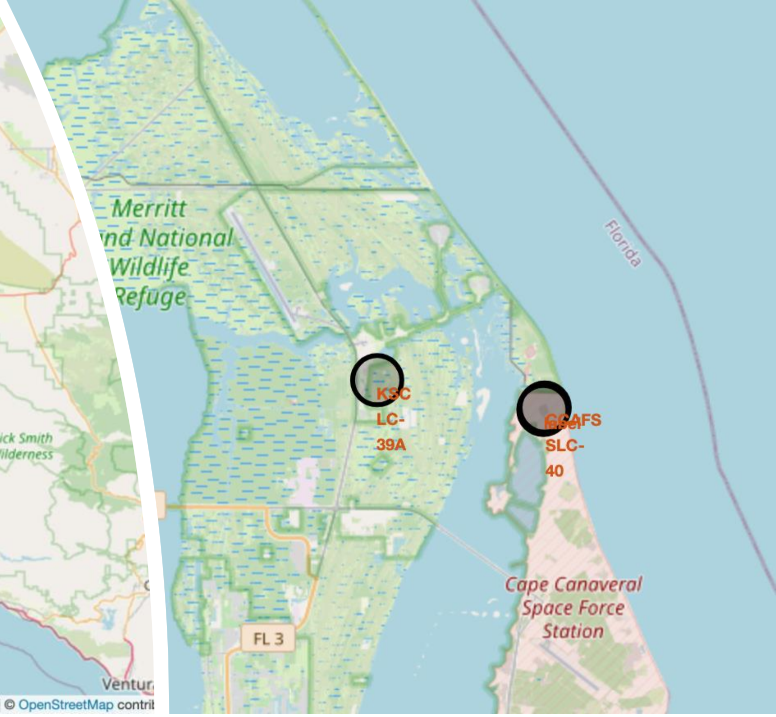
Section 3

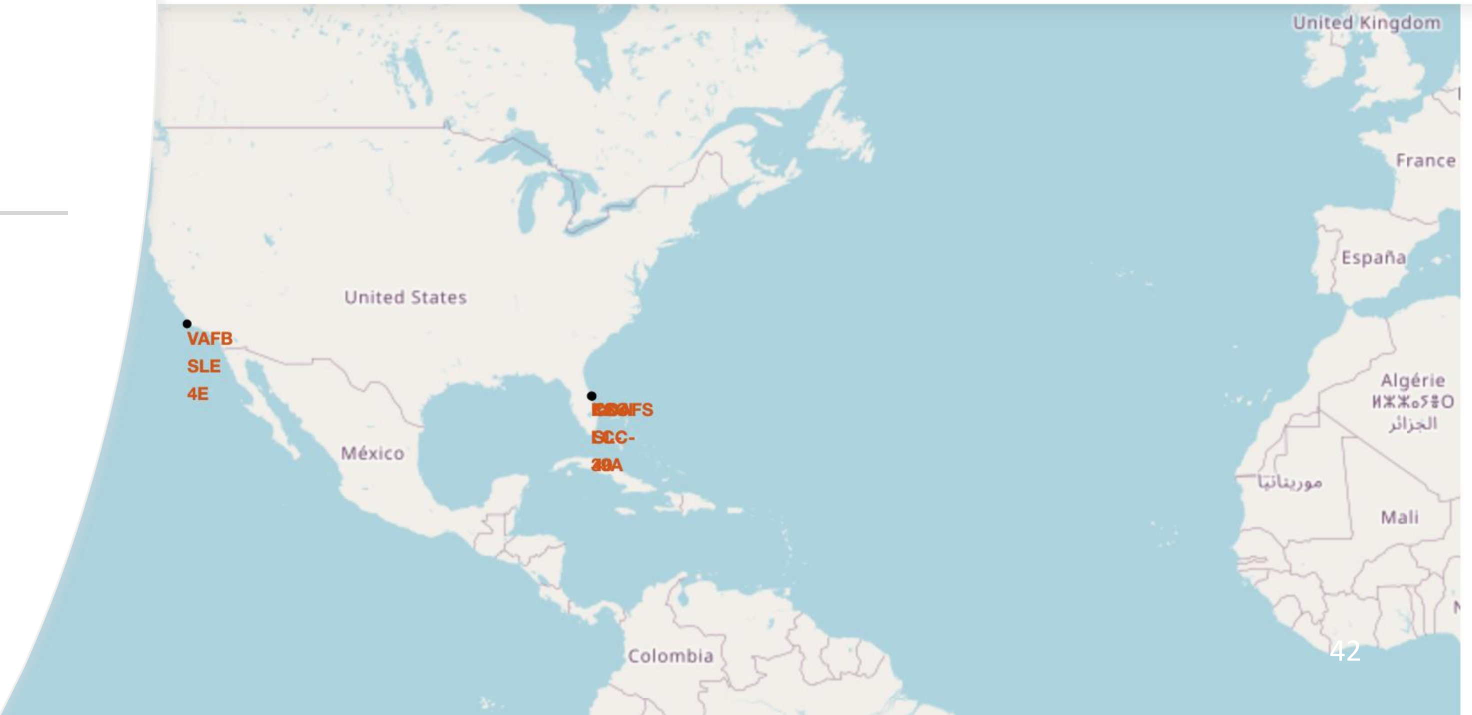
Launch Sites Proximities Analysis

All launch sites in Folium Map

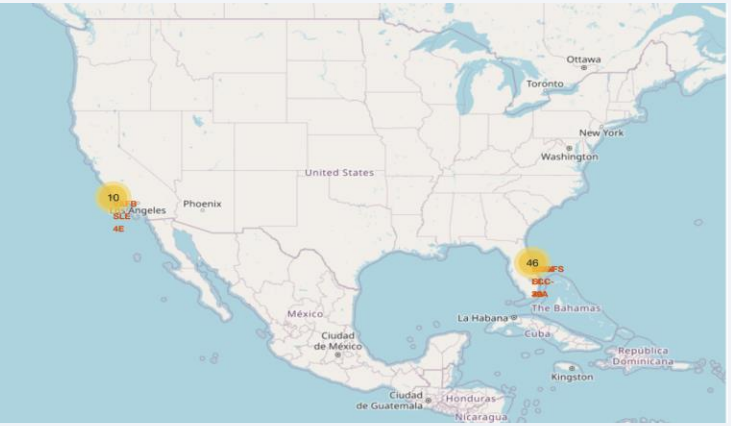
All Launch Sites are in southern coastal areas, some in the east side and others in the west side.





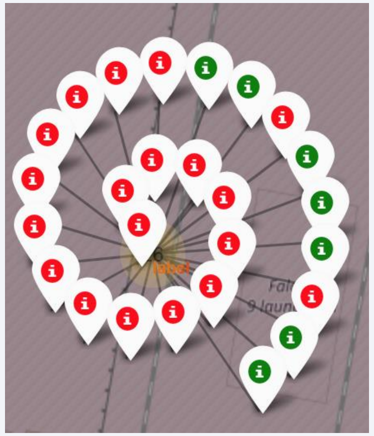


Color Labeled Launch Records

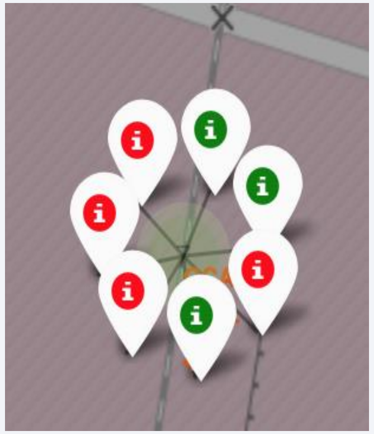


- Green Marker shows successful Launches and Red Marker shows Failures.
- KSC LC-39A have a higher success rate.

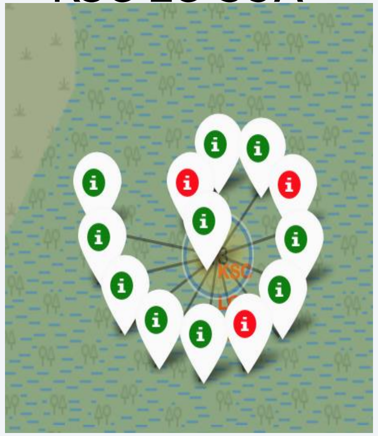
CCAFS LC-40



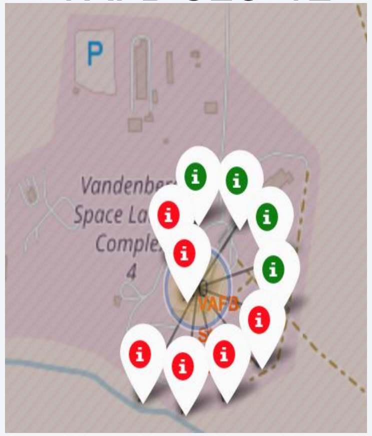
CCAFS SLC-40

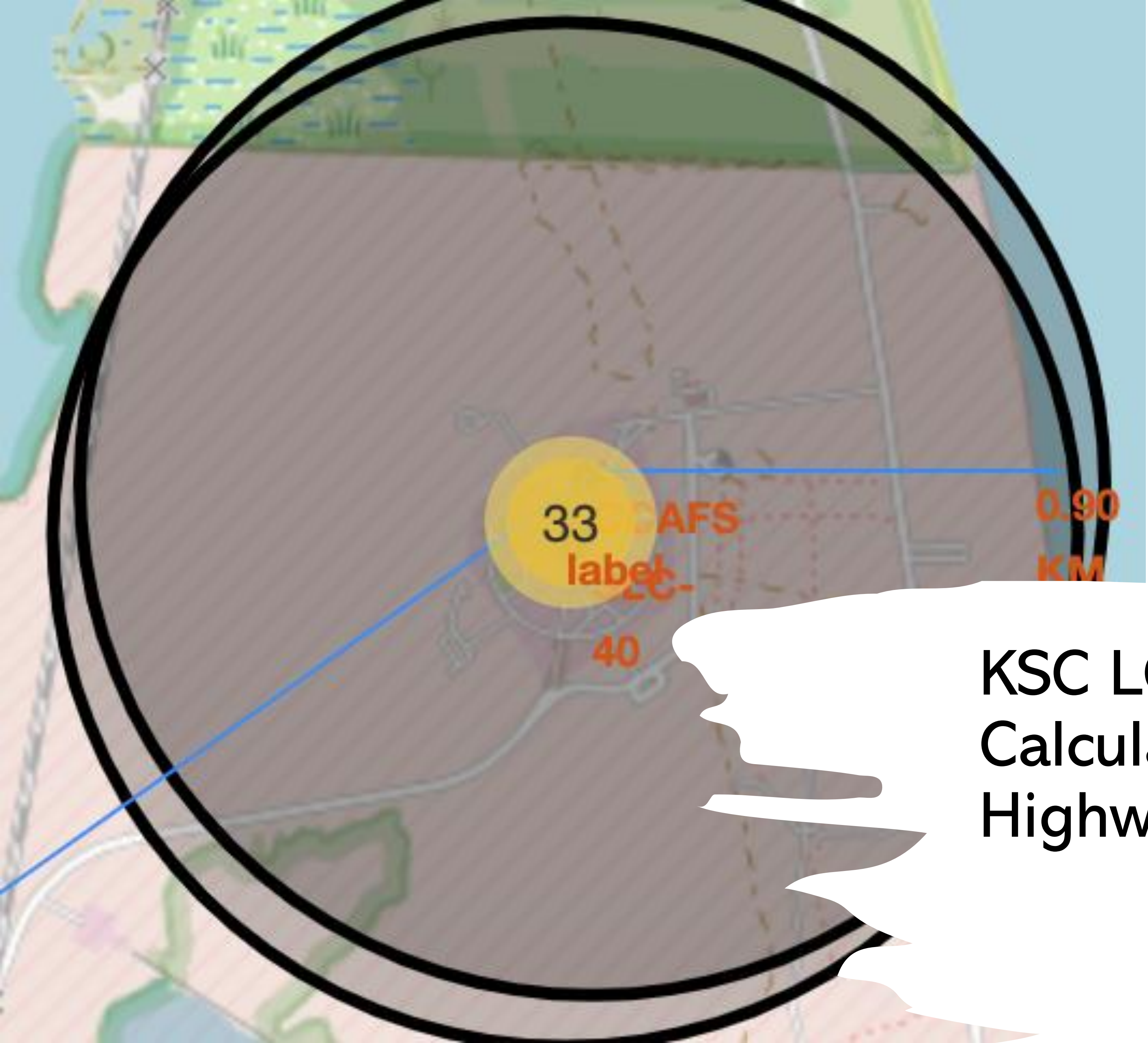


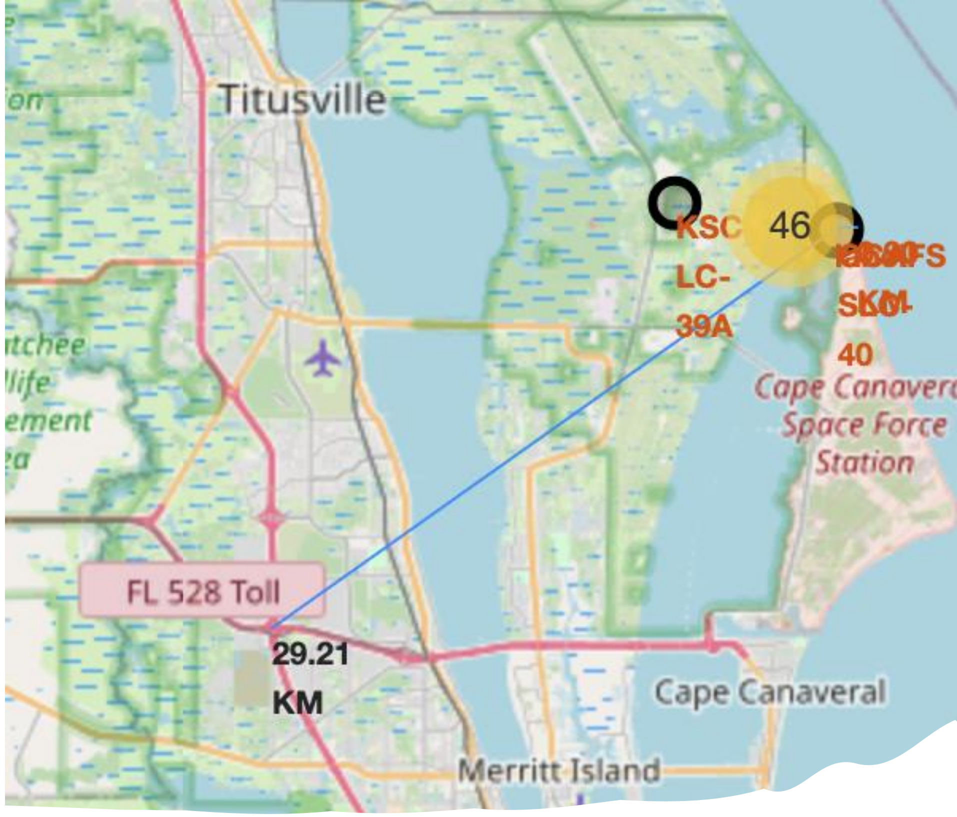
KSC LC-39A



VAFB SLC-4E







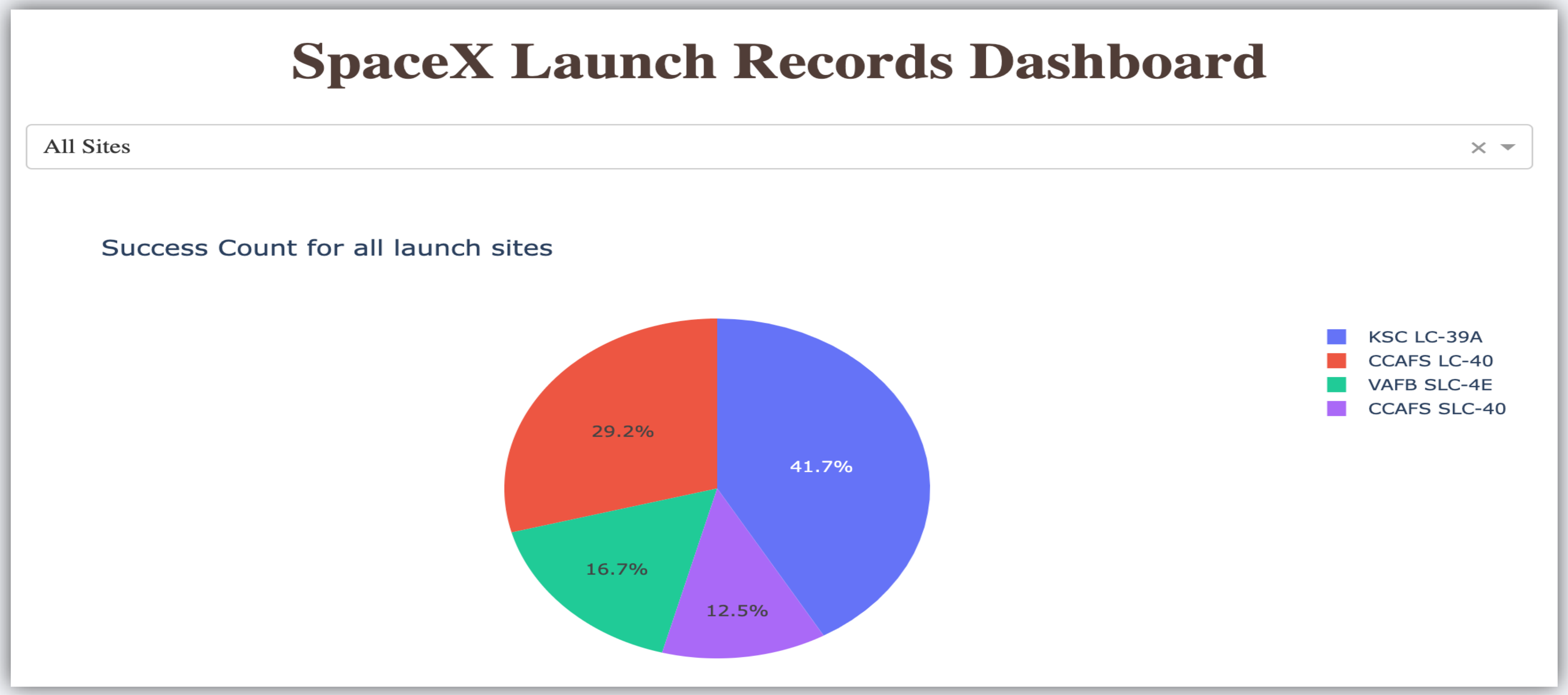
KSC LC-39A Launch Site Calculated distance to Highway and Coastal Zone

Section 4

Build a Dashboard with Plotly Dash

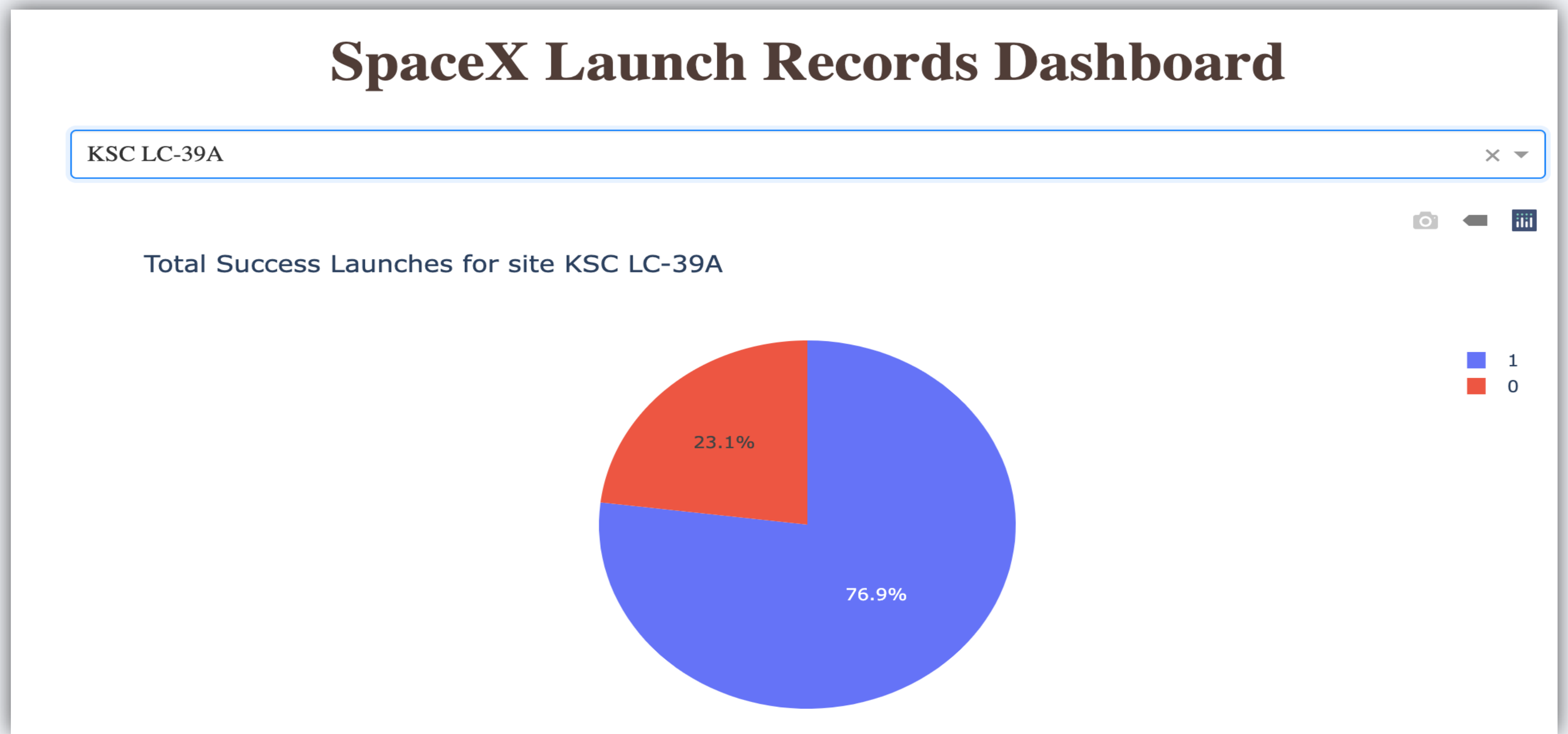
Launch Success Count for All Sites

- KSC LC-39A have a higher success rate, comparing to all other Launch Sites.



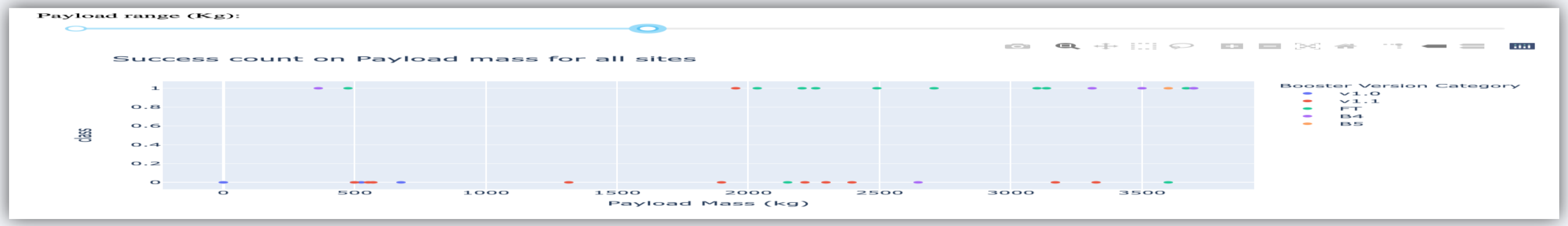
Launch Site with Highest Launch Success Rate

- KSC LC-39A achieved a 76.9% success rate while getting a 23.1% failure rate.



Payload vs. Launch Outcome for All Sites

- Payload Mass 0kg – 4000kg



- Payload Mass 4000kg – 10000kg

- We can see the success rate for the Payload Mass of 0kg – 4000kg is higher than the 4000kg – 10000kg

Payload vs. Launch Outcome for All Sites

- After visual analysis we obtained some insights:
  - KSC LC – 39A is the Launch Site with the highest launch success rate
  - Above the Payload Mass of 4000kg the launch success rate decreases
  - The Booster with the highest launch success rate is FT

Section 5

Predictive Analysis (Classification)

Classification Accuracy

- Logistic Regression: Accuracy - 0.848

```
print("tuned hpyerparameters :(best parameters) ",logreg_cv.best_params_)
print("accuracy :",logreg_cv.best_score_)
```

```
tuned hpyerparameters :(best parameters)  {'C': 0.01, 'penalty': 'l2', 'solver': 'lbfgs'}
accuracy : 0.8464285714285713
```

- Support Vector Machine: Accuracy - 0.848

```
print("tuned hpyerparameters :(best parameters) ",svm_cv.best_params_)
print("accuracy :",svm_cv.best_score_)
```

```
tuned hpyerparameters :(best parameters)  {'C': 1.0, 'gamma': 0.03162277660168379, 'kernel': 'sigmoi
d'}
accuracy : 0.8482142857142856
```

- Decision Tree: Accuracy - 0.873

```
print("tuned hpyerparameters :(best parameters) ",tree_cv.best_params_)
print("accuracy :",tree_cv.best_score_)
```

```
tuned hpyerparameters :(best parameters)  {'criterion': 'entropy', 'max_depth': 6, 'max_features': 'sq
rt', 'min_samples_leaf': 2, 'min_samples_split': 2, 'splitter': 'best'}
accuracy : 0.8732142857142856
```

- K Nearest Neighbors: Accuracy - 0.848

```
print("tuned hpyerparameters :(best parameters) ",knn_cv.best_params_)
print("accuracy :",knn_cv.best_score_)
```

```
tuned hpyerparameters :(best parameters)  {'algorithm': 'auto', 'n_neighbors': 10, 'p': 1}
accuracy : 0.8482142857142858
```

Classification Accuracy

Best Model - "Decision Tree"

```
predictors = [knn_cv, svm_cv, logreg_cv, tree_cv]
best_predictor = ""
best_result = 0
for predictor in predictors:

    predictor.score(X_test, Y_test)
```

```
best_result
```

```
0
```

```
Best_RR = predictor.best_estimator_
```

```
Best_RR
```

```
DecisionTreeClassifier

DecisionTreeClassifier(criterion='entropy', max_depth=6, max_features='sqrt',
                       min_samples_split=5)
```

```
Best_RR = predictor.best_params_
```

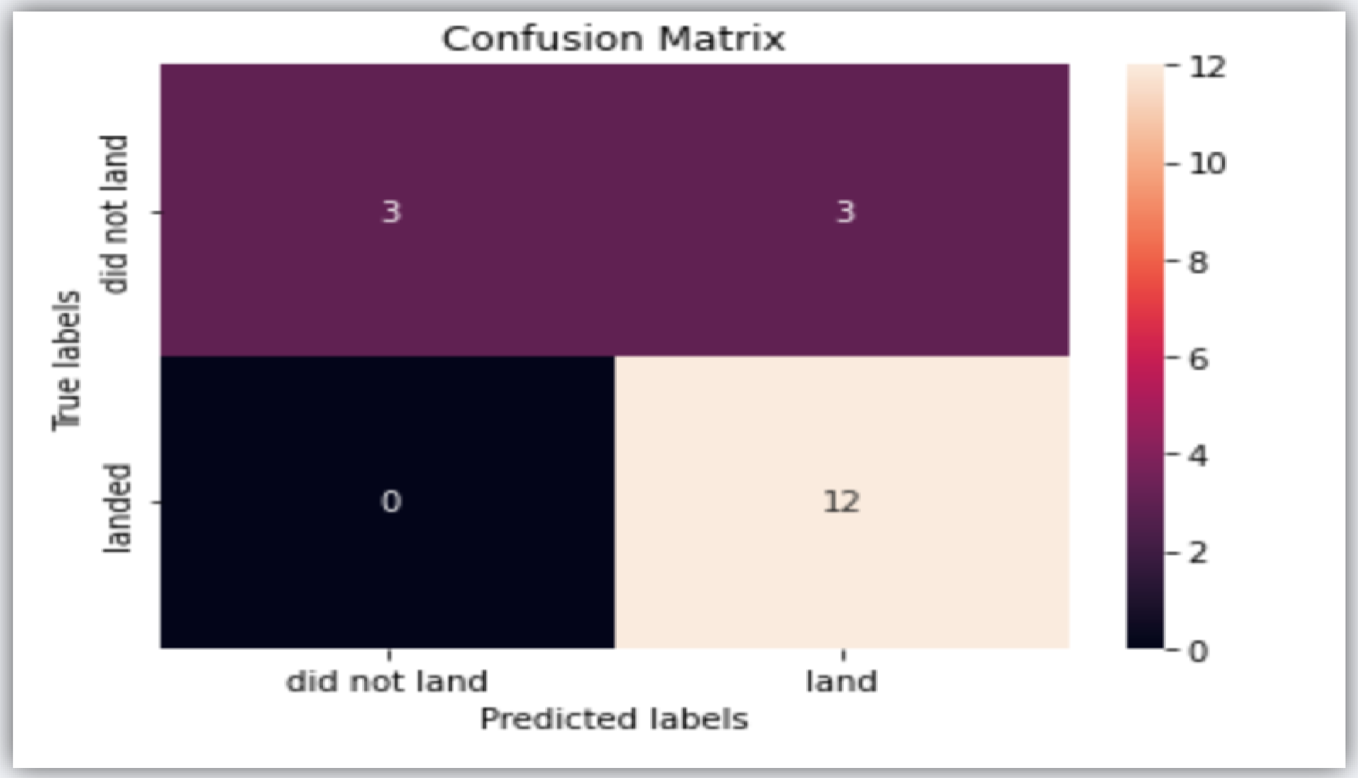
```
Best_RR
```

```
{'criterion': 'entropy',
 'max_depth': 6,
 'max_features': 'sqrt',
 'min_samples_leaf': 1,
 'min_samples_split': 5,
 'splitter': 'best'}
```

Confusion Matrix

- The model that Performs best is the Decision Tree

Decision Tree



Conclusions

- 1. Orbit ES-L1, GEO, HEO, SSO has highest success rate
- 2. Success rate for SpaceX have been increasing with time until it reaches only successuful launches
- 3. KSC LC – 39A have the most success launches, but higher Payload Mass seems to have negative impact on the success of launches
- 4. Decision Tree Classifier is the one that performs Best for Machine Learning Model

ACKNOWLEDGEMENT

- I would like to express my heartfelt gratitude to my instructors, Joseph Santarcangelo and Yan Luo, for their invaluable guidance throughout the completion of this Data Science and Machine Learning Capstone Project.

- I would also like to extend my appreciation to my peers for sharing their questions in the discussion forum. Their inquiries, along with the thoughtful explanations from other instructors, greatly enhanced my understanding and made this journey possible.

Thank you!