IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Muhammmed S. Karipardu̇c

06/10/2023

Outline

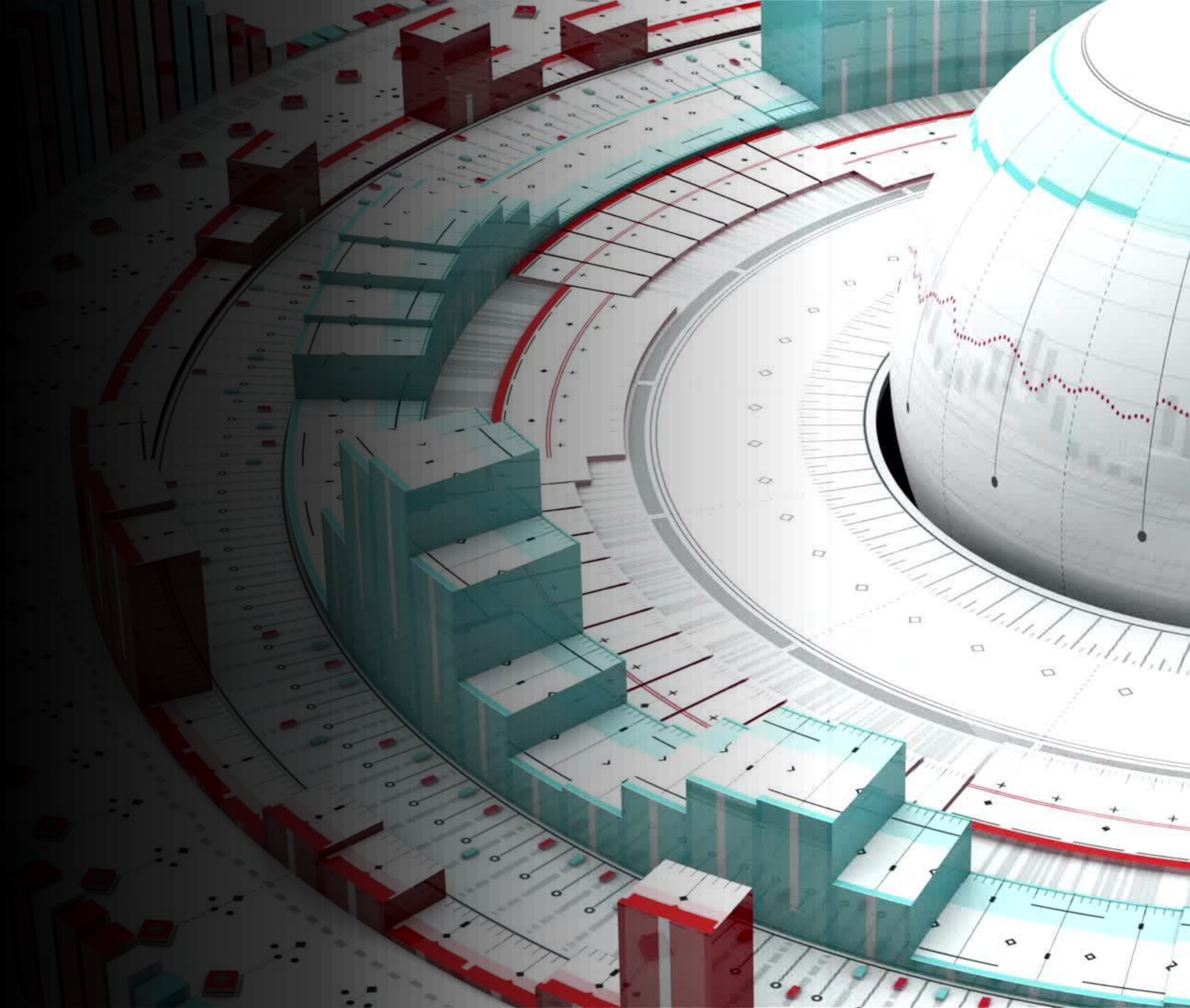
- EXECUTIVE SUMMARY
- INTRODUCTION
- METHODOLOGY
- RESULTS
- CONCLUSION
- APPENDIX

Executive Summary

- The methodologies used in this Data Science Capstone Project were as follows:
  - Data Collection with API with Python
    - The SpaceX Rest API was used to collect the data.
  - Data Collection with Webscraping with Python
  - Data Wrangling with Python
    - Through this method, I was able to filter the data as well as identify null values.
  - Exploratory Data Analysis (EDA) with SQL
  - EDA with Data Visualization with Python
  - Interactive Visual Analytics with Foliumlabs
    - This allowed for a great way for the data to be visualized through different means of display.
  - Machine Learning Prediction
    - Allowed for us to view how different way of predictions vary in accuracy.
  - The Summary of the results are as follows:
    - EDA Results
    - Visual Analytics Results
    - Machine Learning

Introduction

- Project background and context
    - The purpose of this Capstone is to predict whether the first stage of a SpaceX Rocket, Falcon 9, will land successfully. Determining this will allow us to estimate the overall cost of the launch. We will use publicly available information and Data Science methodologies, Data Visualization, and Machine Learning to support our conclusion.
- Problems you want to find answers to:
    - Discover the relationship between Payload Mass, Flight Number, Launch site, and different types of orbits and their overall effect on the success of the launch.
    - Analyze numerous years' worth of landing data to determine whether the rate of successful landing is increasing or decreasing.
    - Using Machine Learning Algorithms to determine which one is the most optimal for binary classification.



Section 1

Methodology

Methodology

- Executive Summary
- Data collection methodology:
    - SpaceX Rest API was used to collect the data.
    - WebScraping was used to collect the pertinent information from Wikipedia.
- Perform data wrangling
    - Data Wrangling via Python was utilized to filter the data as well as identify null values.
- 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

- Data was collected via the SpaceX Rest API from [SpaceX API] as well as from [Wikipedia.]

- GitHub URL of the completed SpaceX API calls notebook: [Github]

01

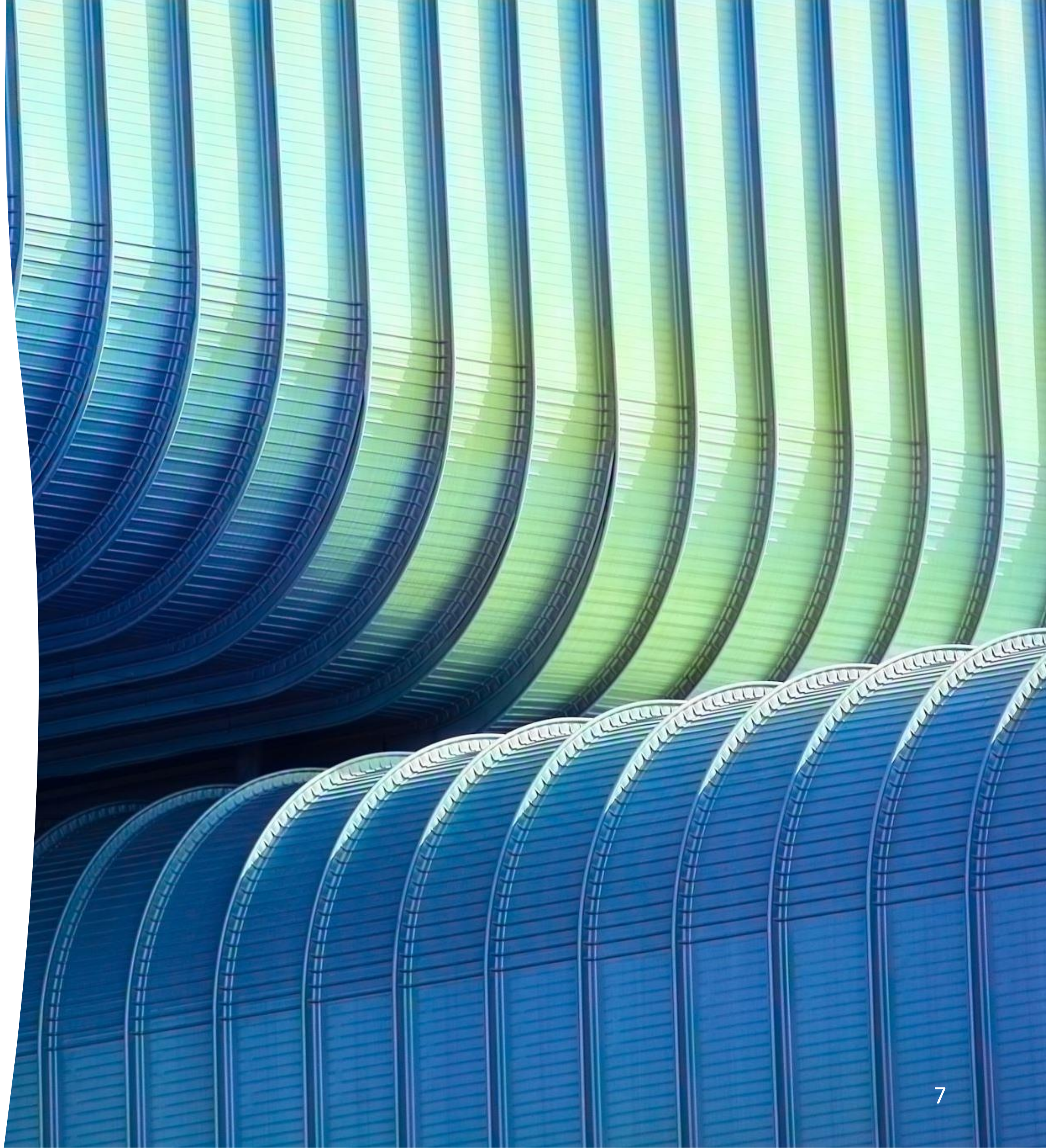
Request and parse the SpaceX launch data using the GET request

02

Filter the dataframe to only include *Falcon 9* launches.

03

Deal with missing/null values.



Data Collection - Scraping

- Use Webscraping to collect the information from here.
- Utilize and manipulate the collected data to extract relevant information AND create dataframes.
- Github Link.

Request	Request the Falcon 9 Wiki Page from its URL.
Extract	Extract all column/variable names from the HTML table header.
Create	Create a data frame by parsing the launch HTML tables.

Data Wrangling

- Collect information from here.
- Github.

1

Collect the number of launches on each site

2

Calculate the number and occurrence of each orbit

3

Calculate the number and occurrence of mission outcome per orbit type.

EDA with Data Visualization

- The following charts were plotted to analyze the relationship between different variables:
  - Payload vs. Flight Number
  - Launch Site vs. Flight Number
  - Launch Site vs. Payload Mass (kg)
  - Orbit Type vs. Payload Mass (kg)
- Tasks Summary:
  - Bar charts allow for visual and numerical comparison between the different categories.
  - Line charts (scatter plots) allow us to see the relationship between the numerous variables.
  - Github.

EDA with SQL

- The following query tasks were completed:
  - Display the names of the distinct launch sites
  - Using SQL, show the 5 records that being with string ‘CCA’
  - Show the total payload mass carried by boosters launched by NASA (CRS)
  - Average payload mass carried by booster version F9 v1.1
  - Display the first successful landing outcome date in the ground pad
  - Display the names of the boosters that were successful in the drone ship and have a payload mass between 4000 and 6000 kg
  - Display the total number of successful and failed missions]
  - Display the booster versions with the maximum payload mass
  - For the months in the year 2015, display the month names, failure landing outcome in drone ship, booster versions and launch sites
  - Rank the count of successful landing outcomes between the dates of 04-06-2010 and 20-03-2017 (desc).
  - Github link.

Build an Interactive Map with Folium

- The red circles represent launch sites coordinates and the blue circle represents the NASA Johnson Space Center’s Coordinates
- Green markers represent successful outcomes whereas red markers represent failed outcomes.
- Colored show the distance between coastline, railway, city, and highway and the launch site.
- [Github.]

Build a Dashboard with Plotly Dash

- **Dropdown** menu allows for you to select what sites to view data from.
- **Pie chart** allows for a visual representation of the successful launches for all sites in the numerical form of a percentage.
- **Slider** allows for the user to select the payload mass.
- **The scatter chart** allows for the user to view the correlation between the payload and launch success.
- Github.

Predictive Analysis (Classification)

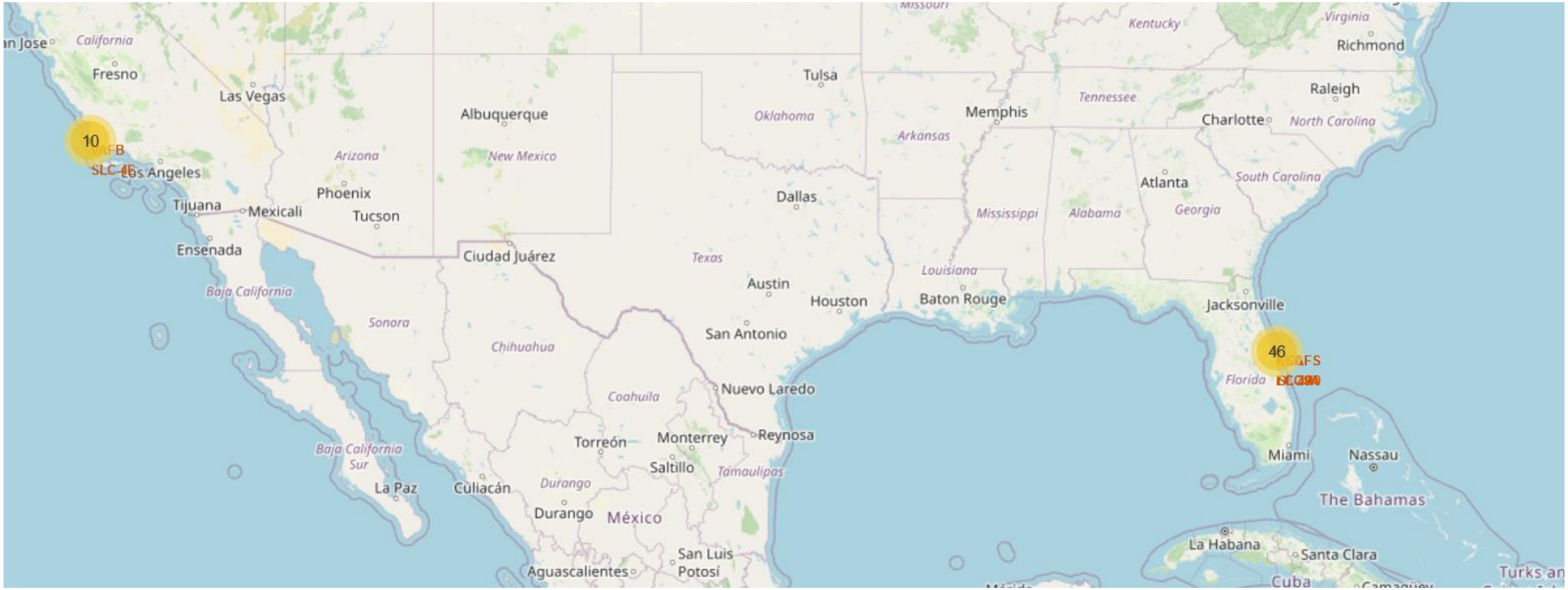
- Trained and tested the following classification methods to determine which one is the best one and its ideal parameters:
  - SVM
  - Classification Trees
  - Logistic Regression
  - K Nearest Neighbors (KNN)
  - Github.

Create	Create a numpy array from the column 'Class' in data
Create	Create a logistic regression object and fit the parameters and calculate the accuracy using method score
Create	Create a knn object and fit the parameters and calculate the accuracy using method score
Find	Find the method that performs the best
Create	Create a svm and fit the parameters and calculate the accuracy using method score
Standardize	Standardize the data in X
Split	Split the data into X_train, X_test, Y_train, Y_test
Create	Create a tree classifier object and fit the parameters and calculate the accuracy using method score

Results

- Exploratory data analysis results
  - As the years went by, so did the success of the launches
  - The following orbits have a 100% success rate
    - SSO, HEO, GEO, ES-L1
  - VAFB SLC 43 and KSC LC-39A have the highest success rates at 77%
- Predictive Analytics results
  - Of all the methods tested, the best method is the Tree method, which has an accuracy rating of 90.36%
  - The parameters for this method are criterion, gini, max depth 2, max features, auto, min samples leaf 2, min samples split 10, and splitter, best.
- Interactive Folium Labs results
  - The launches were nearly all in coastal launch sites.
  - Green circles indicate success whereas red circles indicate failures.

Results (cont.)



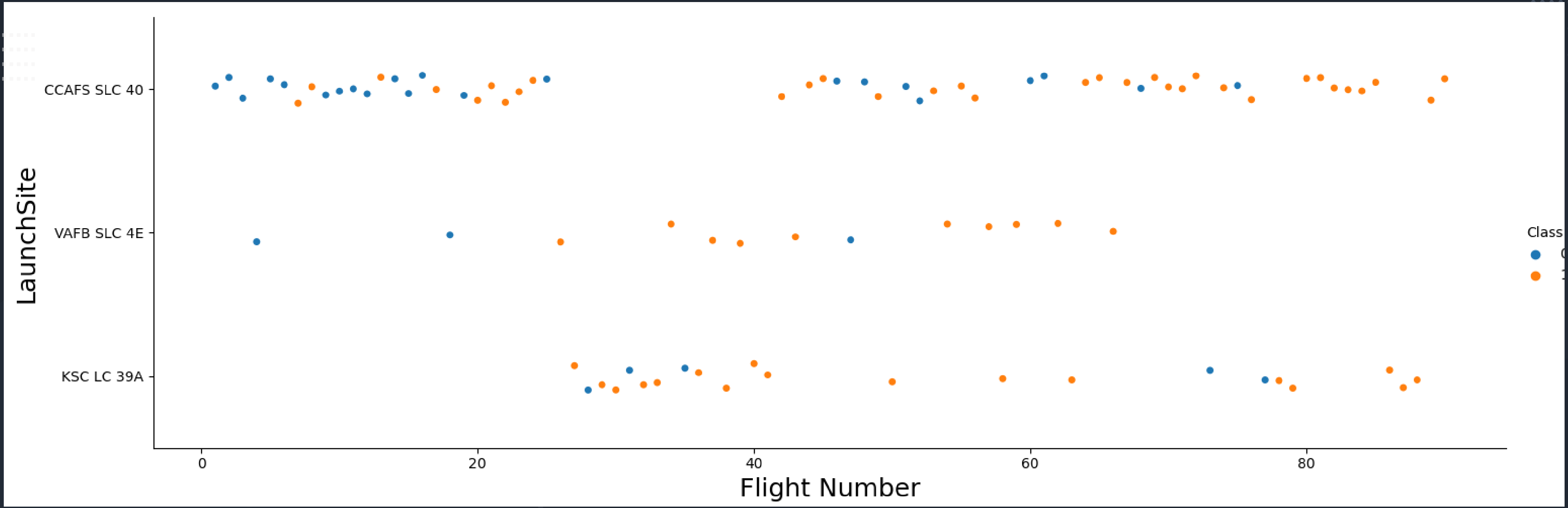
Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

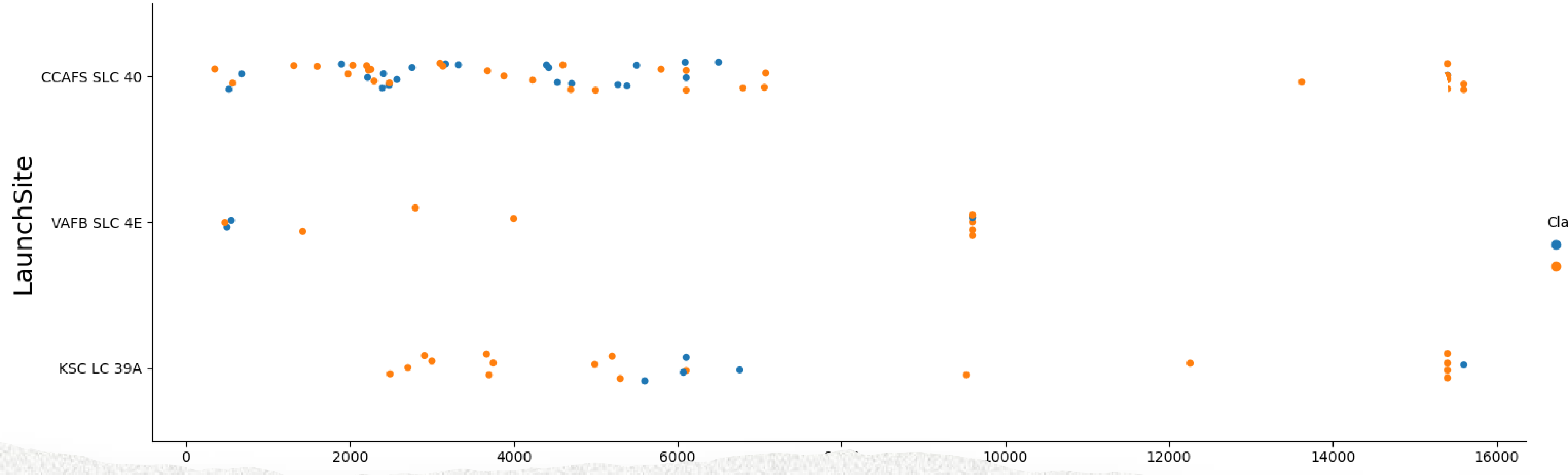
- Show a scatter plot of Flight Number vs. Launch Site

- Show the screenshot of the scatter plot with explanations



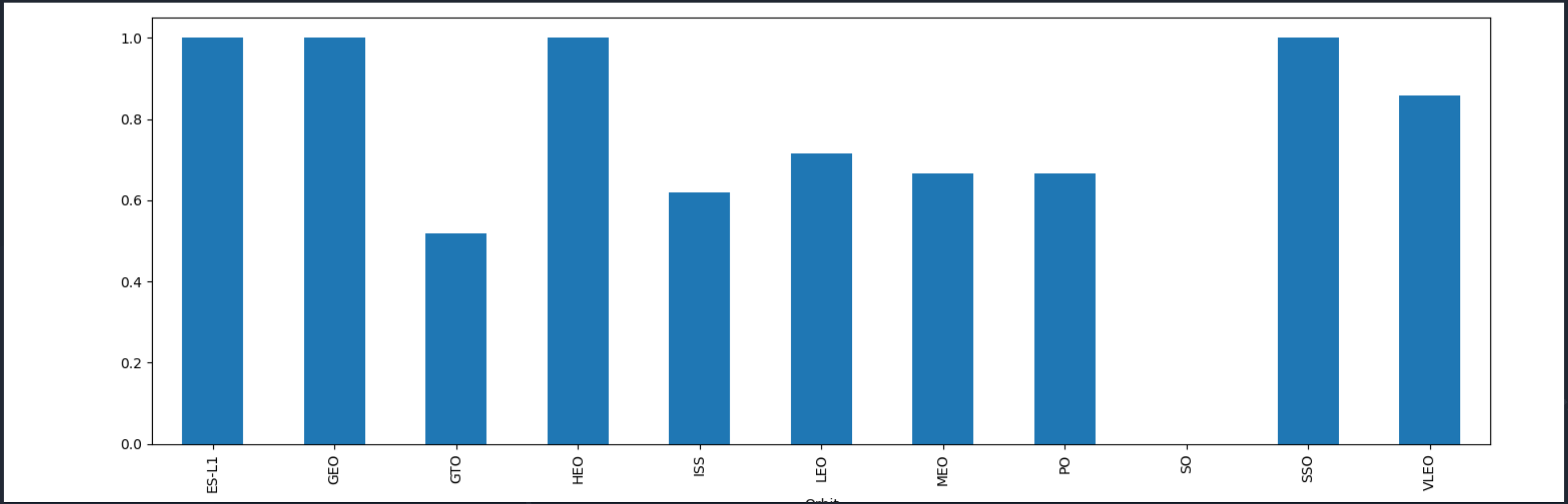
Flight Number vs . Launch Site

- CCAFS SLC 40 has the highest success rate.
- As time went on, the flights had a higher success rate.
- Which indicates that earlier flights were less successful.



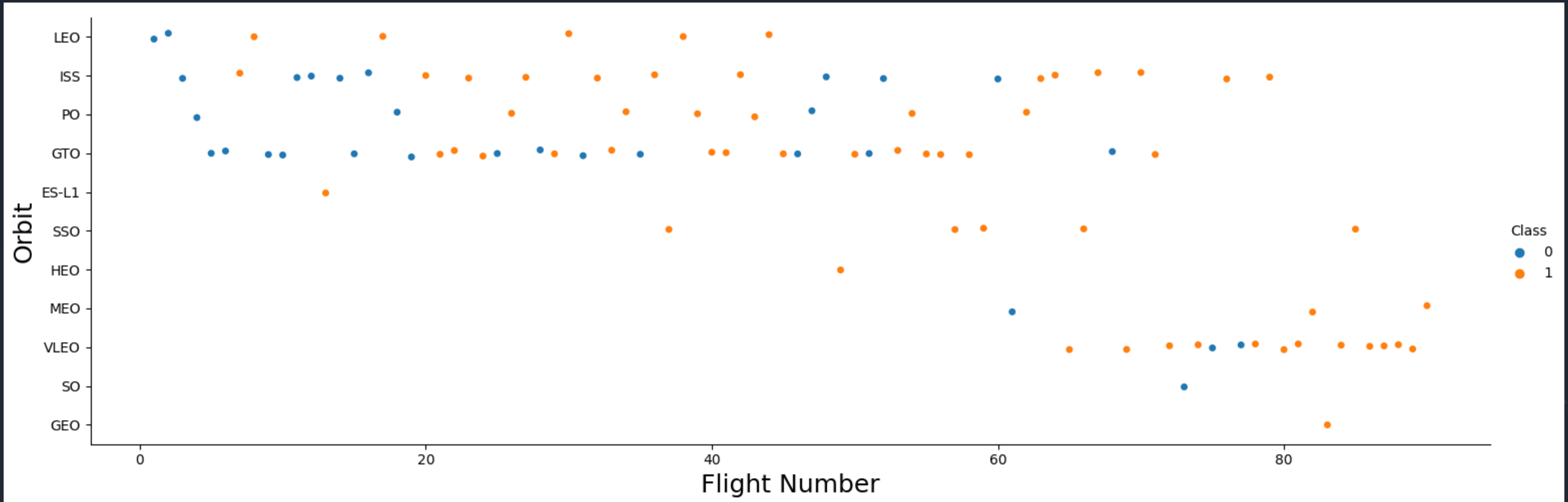
Payload vs. Launch Site

- In general, the higher the payload mass, the higher the success rate.
- There were more launches with lighter payload masses.
- KSC LC 39A was the most successful launch site.



Success Rate vs. Orbit Type

- ES-L1, GEO, HEO, and SSO were all 100% successful.
- SO was 0% successful.
- All other Orbit types were between 50-80% successful.

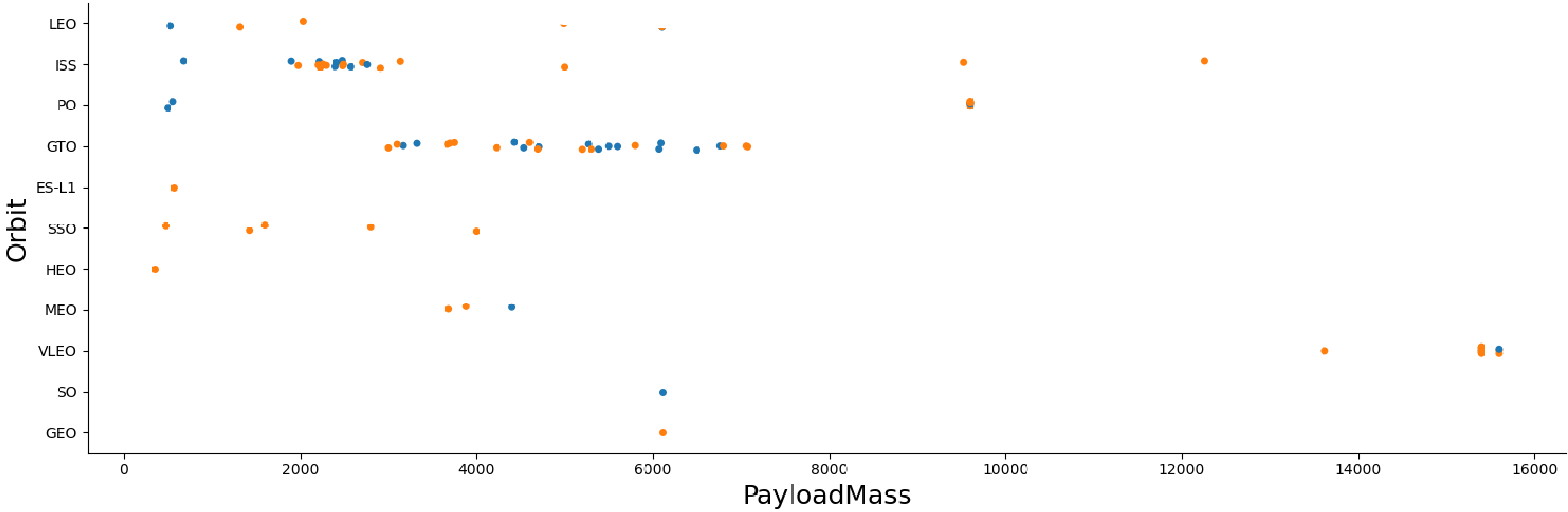


Flight Number vs. Orbit Type

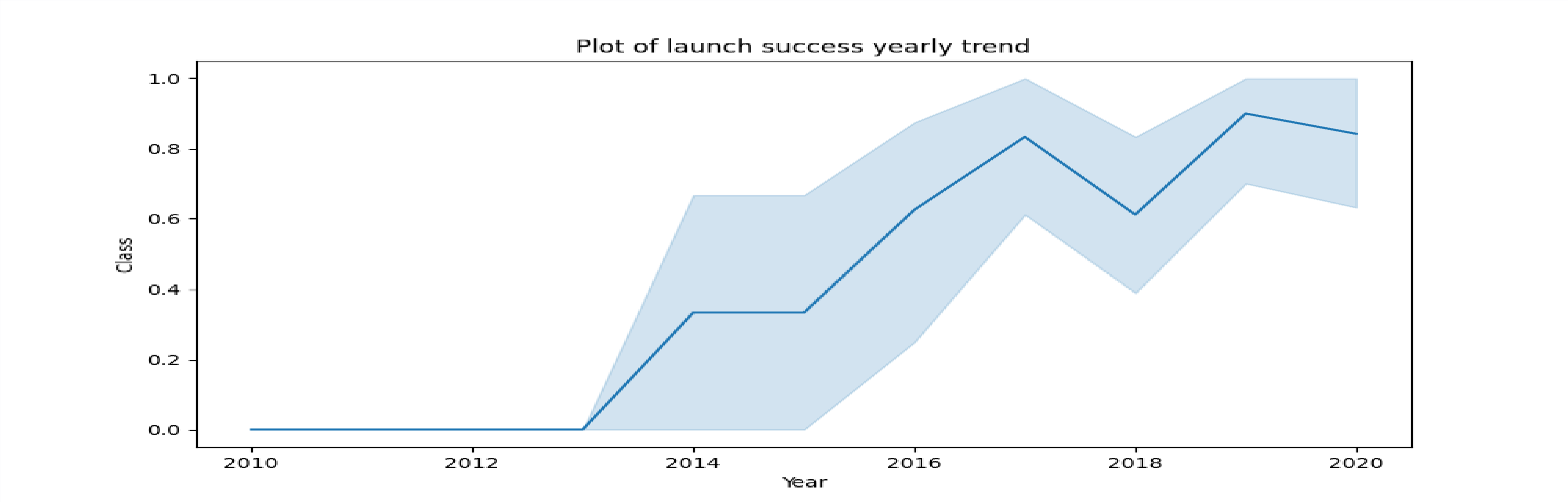
- SSO had only five launches, and it was successful.
- VLEO had the lowest failure rate out of all orbits with more than seven launches.
- GTO had the highest rate of failure from all orbits with more than seven launches.
- In general, the higher the flight number, the higher the success rate.

Payload vs. Orbit Type

- Most launches were done for payload masses under 8000 kg.
- The SSO and HEO orbit works best with lower payload masses.
- The VLEO and PO orbits works the best with higher payload masses.
- There is no correlation between payload mass and success for orbits ISS and GTO.



Launch Success Yearly Trend



- Success improved semi-steadily until 2016.
- From 2017 until 2018 there was a decline in success.
- From 2018 until 2019 success increased.
- From 2019 until 2020 the success declined.

All Launch Site Names

%sql watson://yyy33800:dwNKg8J3L0IBd6CP@1bbf73c5-d84a-4bb0-85b9-ab1a4348f4a4.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:32286/BLUDB?security=SSL\n

%sql SELECT DISTINCT(LAUNCH_SITE) from SPACEXTBL WHERE Launch_Site <> 'None'

SELECT UNIQUE RETURNS AN ERROR HENCE WHY WE ARE USING SELECT DISTINCT

<u>Launch_Sites</u>

CCAFS LC-40

VAFB SLC-4E

KSC LC-39A

CCAFS SLC-40

```
%sql watson://yyy33800:dwNKg8J3L0IBd6CP@1bbf73c5-d84a-4bb0-85b9-ab1a4348f4a4.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:32286/BLUDB?security=SSL\n

%sql SELECT DISTINCT(LAUNCH_SITE) from SPACEXTBL WHERE Launch_Site <> 'None'

## SELECT UNIQUE RETURNS AN ERROR HENCE WHY WE ARE USING SELECT DISTINCT
```

```
Connection info needed in SQLAlchemy format, example:
               postgresql://username:password@hostname/dbname
               or an existing connection: dict_keys(['sqlite:///my_data1.db'])
Can't load plugin: sqlalchemy.dialects:watson
Connection info needed in SQLAlchemy format, example:
               postgresql://username:password@hostname/dbname
               or an existing connection: dict_keys(['sqlite:///my_data1.db'])
 * sqlite:///my_data1.db
```

Launch Site Names Begin with 'CCA'

- %sql SELECT * FROM SPACEXTBL where LAUNCH_SITE like'CCA%' LIMIT 5

```
%sql SELECT * FROM SPACEXTBL where LAUNCH_SITE like'CCA%' LIMIT 5

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
06/04/2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0.0	LEO	SpaceX	Success	Failure (parachute)
12/08/2010	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0.0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
22/05/2012	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525.0	LEO (ISS)	NASA (COTS)	Success	No attempt
10/08/2012	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500.0	LEO (ISS)	NASA (CRS)	Success	No attempt
03/01/2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677.0	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

- %sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE CUSTOMER = 'NASA (CRS)';

```
%sql SELECT SUM(PAYLOAD_MASS__KG_)  FROM SPACEXTBL WHERE CUSTOMER = 'NASA (CRS)';
```

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

SUM(PAYLOAD_MASS__KG_)
45596.0

Average Payload Mass by F9 v1.1

- Present your query result with a short explanation here

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

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE BOOSTER_VERSION = 'F9 v1.1';
```

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

AVG(PAYLOAD_MASS__KG_)
2928.4

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad
- Present your query result with a short explanation here

```
%sql SELECT MIN(Date) FROM SPACEXTBL WHERE Landing_Outcome = 'Success (ground pad)'
```

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

MIN(Date)
01/08/2018

Successful Drone Ship Landing with Payload between 4000 and 6000

- Present your query result with a short explanation here

```
%sql SELECT Booster_Version FROM SPACEXTBL WHERE Landing_Outcome = 'Success (drone ship)' AND Payload_Mass__KG_ BETWEEN 4000 AND 6000
```

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

Booster_Version
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
F9 FT B1031.2

Total Number of Successful and Failure Mission Outcomes

- Present your query result with a short explanation here

```
%sql SELECT Mission_Outcome, COUNT(*) AS SUM FROM SPACEXTBL GROUP BY Mission_Outcome
```

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

Mission_Outcome	SUM
None	898
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass
- Present your query result with a short explanation here

```
%sql SELECT BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL);
```

```
 * 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

- Present your query result with a short explanation here

```
%sql SELECT substr(Date,4,2), MISSION_OUTCOME,BOOSTER_VERSION,LAUNCH_SITE FROM SPACEXTBL WHERE substr(Date,7,4)='2015'
```

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

substr(Date,4,2)	Mission_Outcome	Booster_Version	Launch_Site
10	Success	F9 v1.1 B1012	CCAFS LC-40
11	Success	F9 v1.1 B1013	CCAFS LC-40
02	Success	F9 v1.1 B1014	CCAFS LC-40
04	Success	F9 v1.1 B1015	CCAFS LC-40
04	Success	F9 v1.1 B1016	CCAFS LC-40
06	Failure (in flight)	F9 v1.1 B1018	CCAFS LC-40
12	Success	F9 FT B1019	CCAFS LC-40

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

- Present your query result with a short explanation here

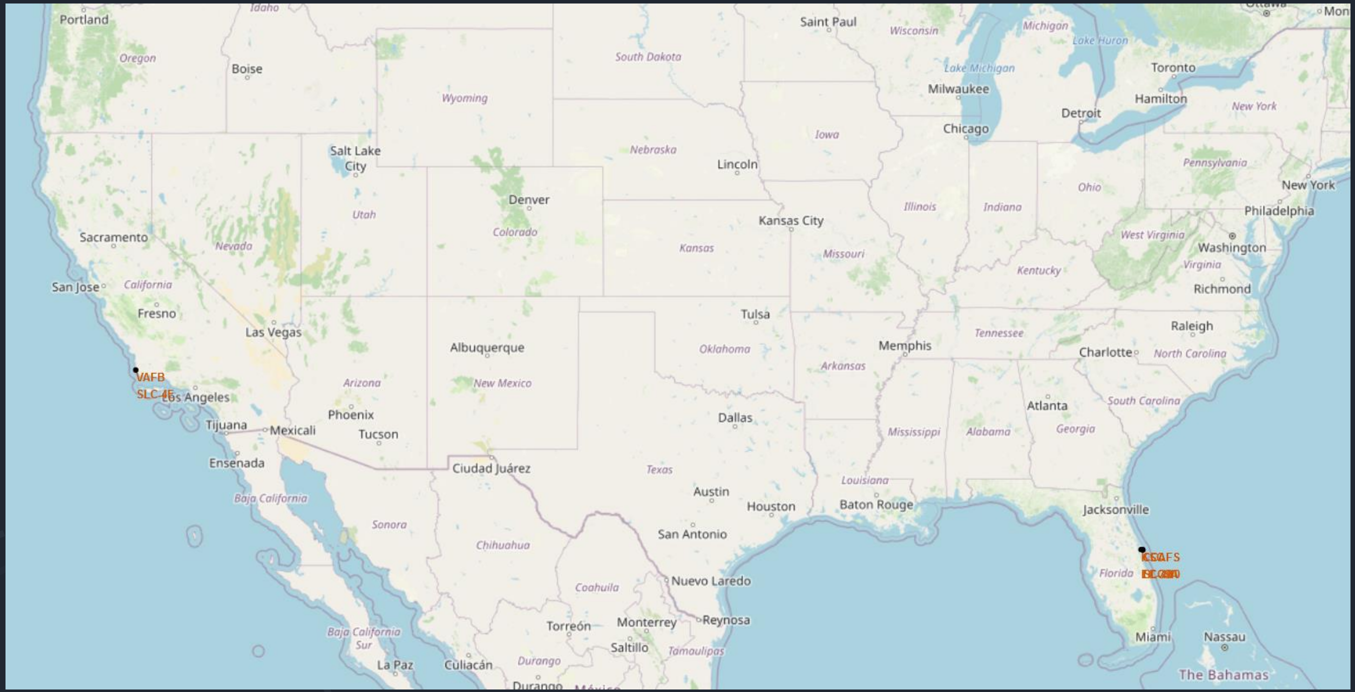
```
%sql SELECT Landing_Outcome, count(*) as Count_Outcomes FROM SPACEXTBL WHERE DATE BETWEEN '04-06-2010' AND '20-03-2017' GROUP BY Landing_Outcome \
ORDER BY Count_Outcomes DESC
```

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

Landing_Outcome	Count_Outcomes
Success	20
No attempt	10
Success (drone ship)	8
Success (ground pad)	7
Failure (drone ship)	3
Failure	3
Failure (parachute)	2
Controlled (ocean)	2
No attempt	1

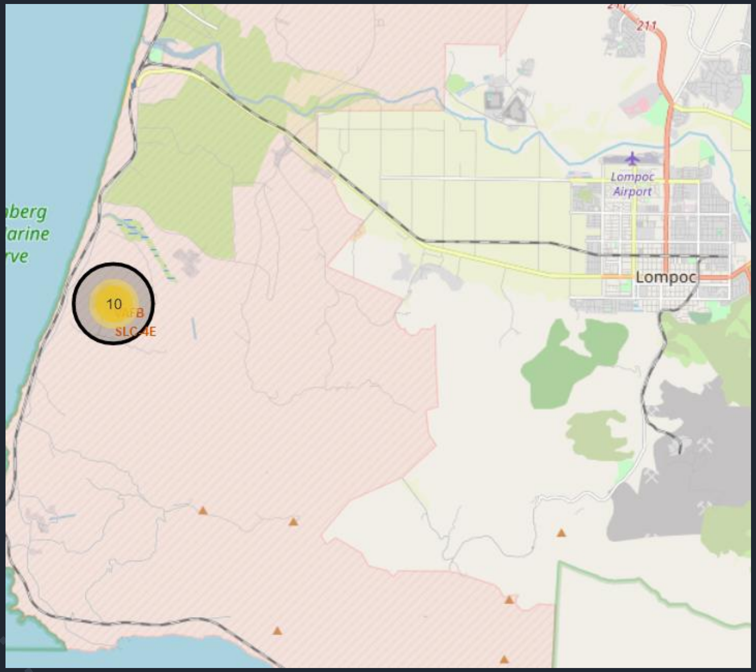
Section 3

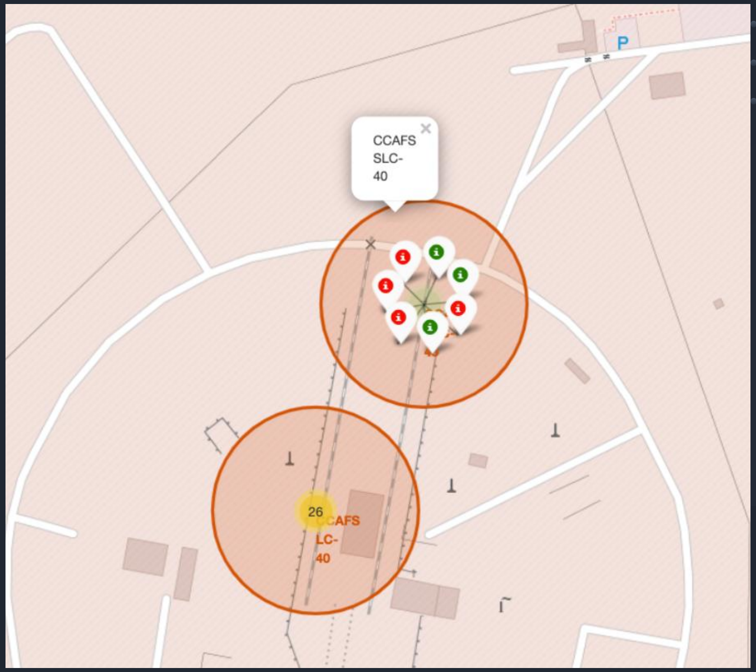
Launch Sites Proximities Analysis



<Folium Map Screenshot 1>

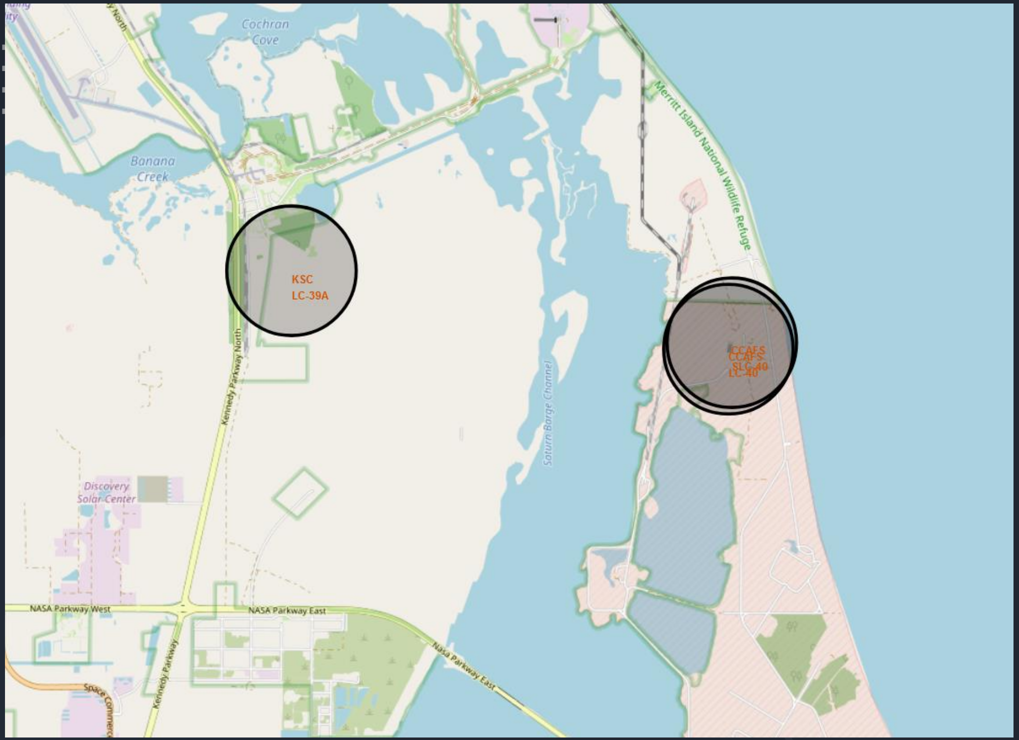
- Replace <Folium map screenshot 1> title with an appropriate title
- Explore the generated folium map and make a proper screenshot to include all launch sites' location markers on a global map
- Explain the important elements and findings on the screenshot

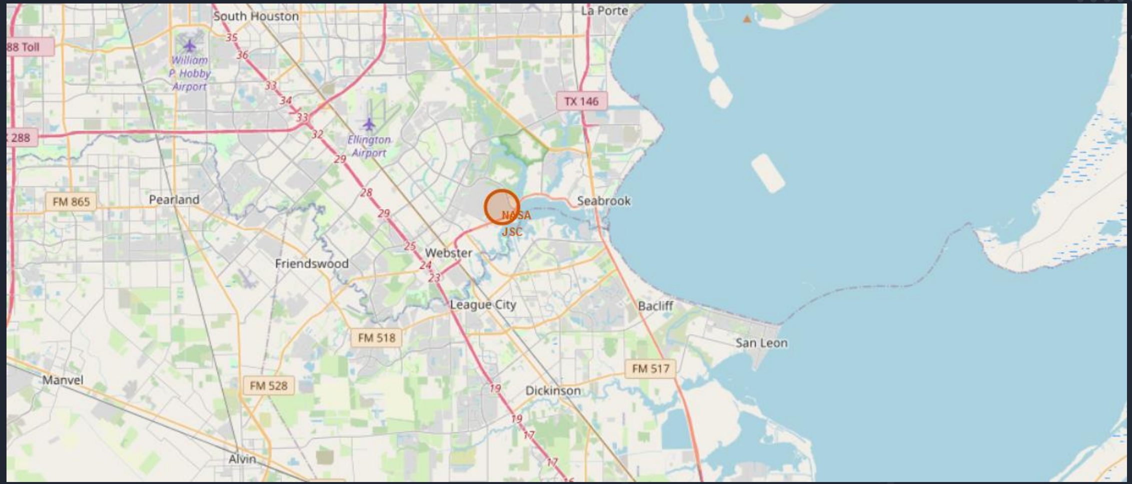




<Folium Map Screenshot 2>

- Replace <Folium map screenshot 2> title with an appropriate title
- Explore the folium map and make a proper screenshot to show the color-labeled launch outcomes on the map
- Explain the important elements and findings on the screenshot



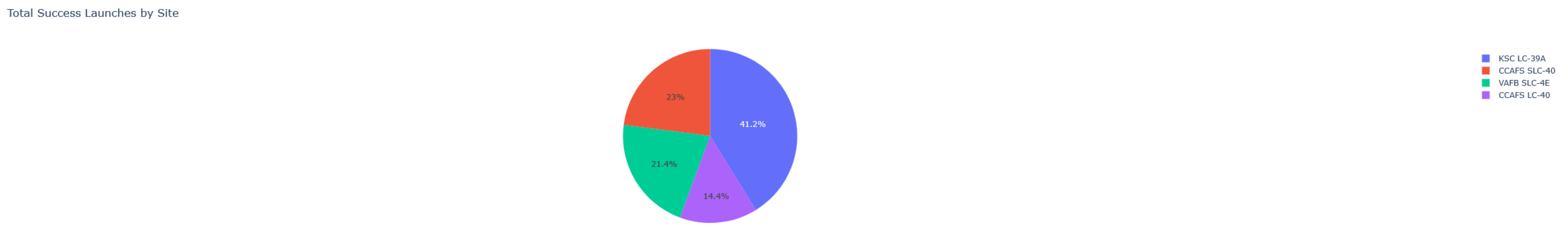


<Folium Map Screenshot 3>

- Replace <Folium map screenshot 3> title with an appropriate title
- Explore the generated folium map and show the screenshot of a selected launch site to its proximities such as railway, highway, coastline, with distance calculated and displayed
- Explain the important elements and findings on the screenshot

Section 4

Build a Dashboard with Plotly Dash

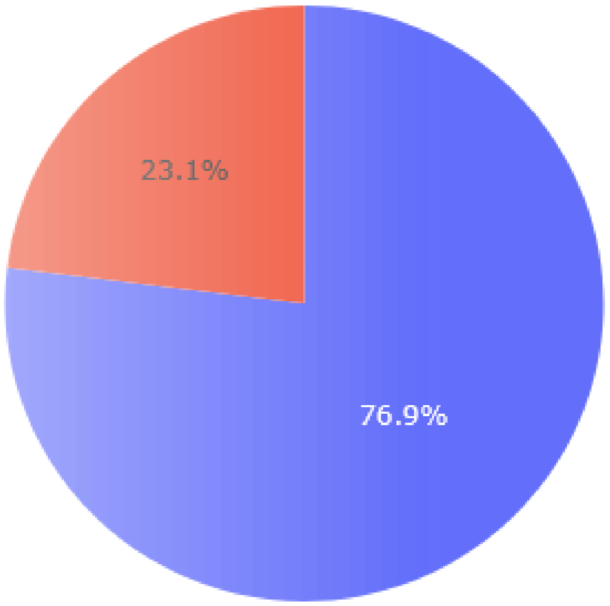


Total Success Launches by site (All sites)

- As can be seen from the pie chart, KSC-LC 39A has the highest share of successful launches whereas CCAFS LC-40 has the lowest.

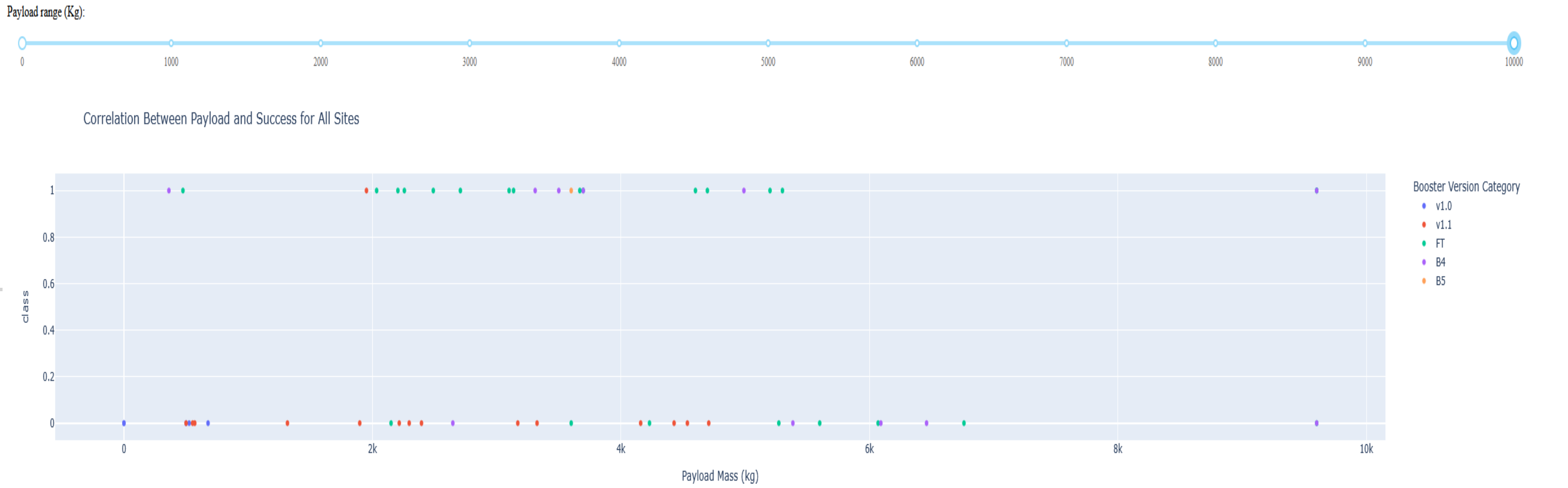
Most Successful Launch site

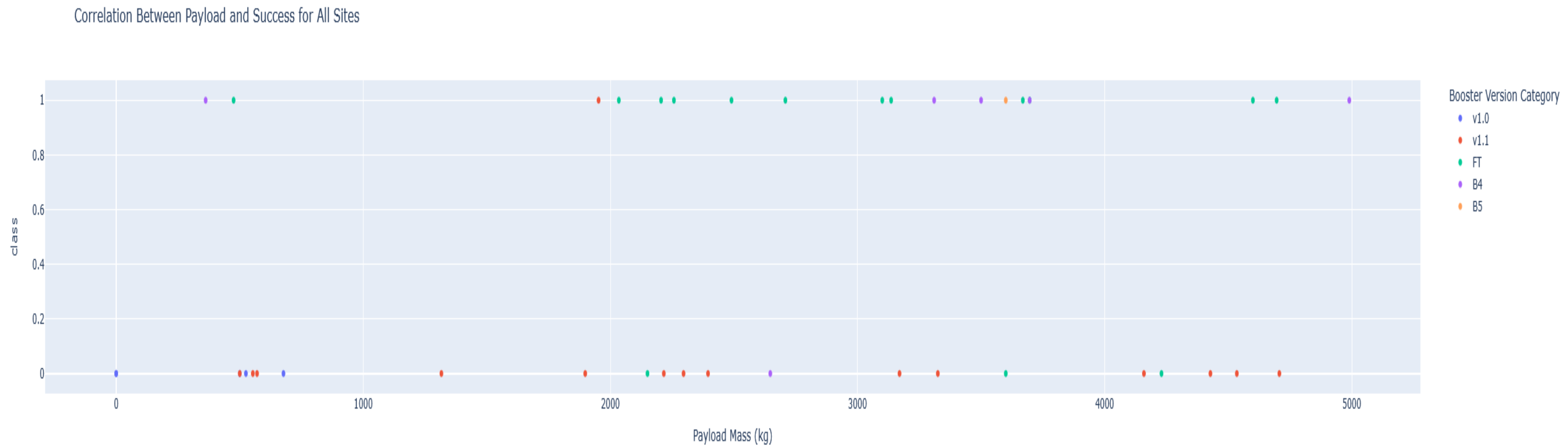
- As mentioned in the previous slide, KSC LC-39A has the highest share of successful launches.
- A major 76.9% of the launches on this site were successful!



Correlation between payload mass and success at all sites

- Most launches are with payload masses under 6000 kg.
- FT has the most successful launches.



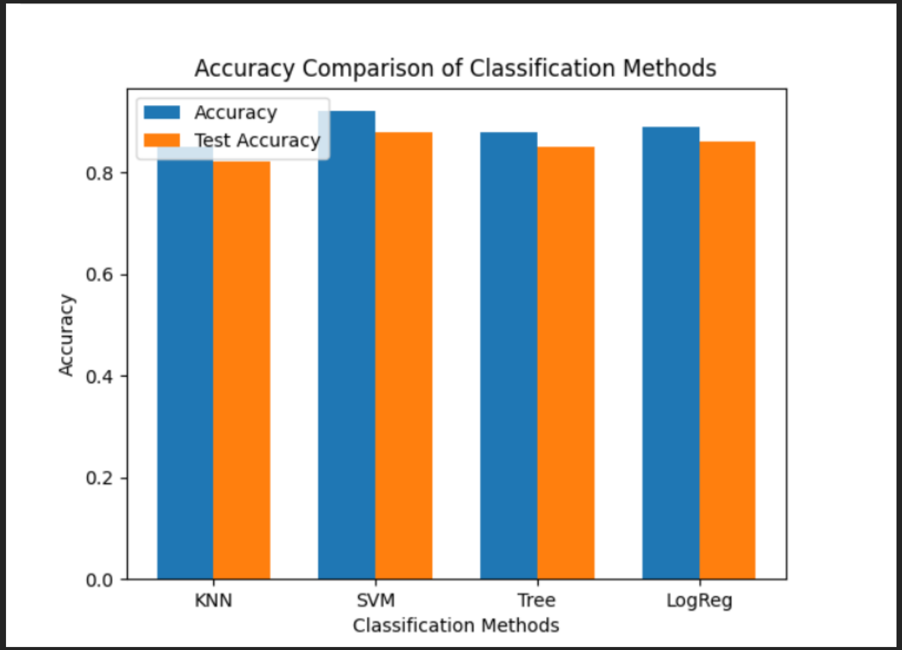


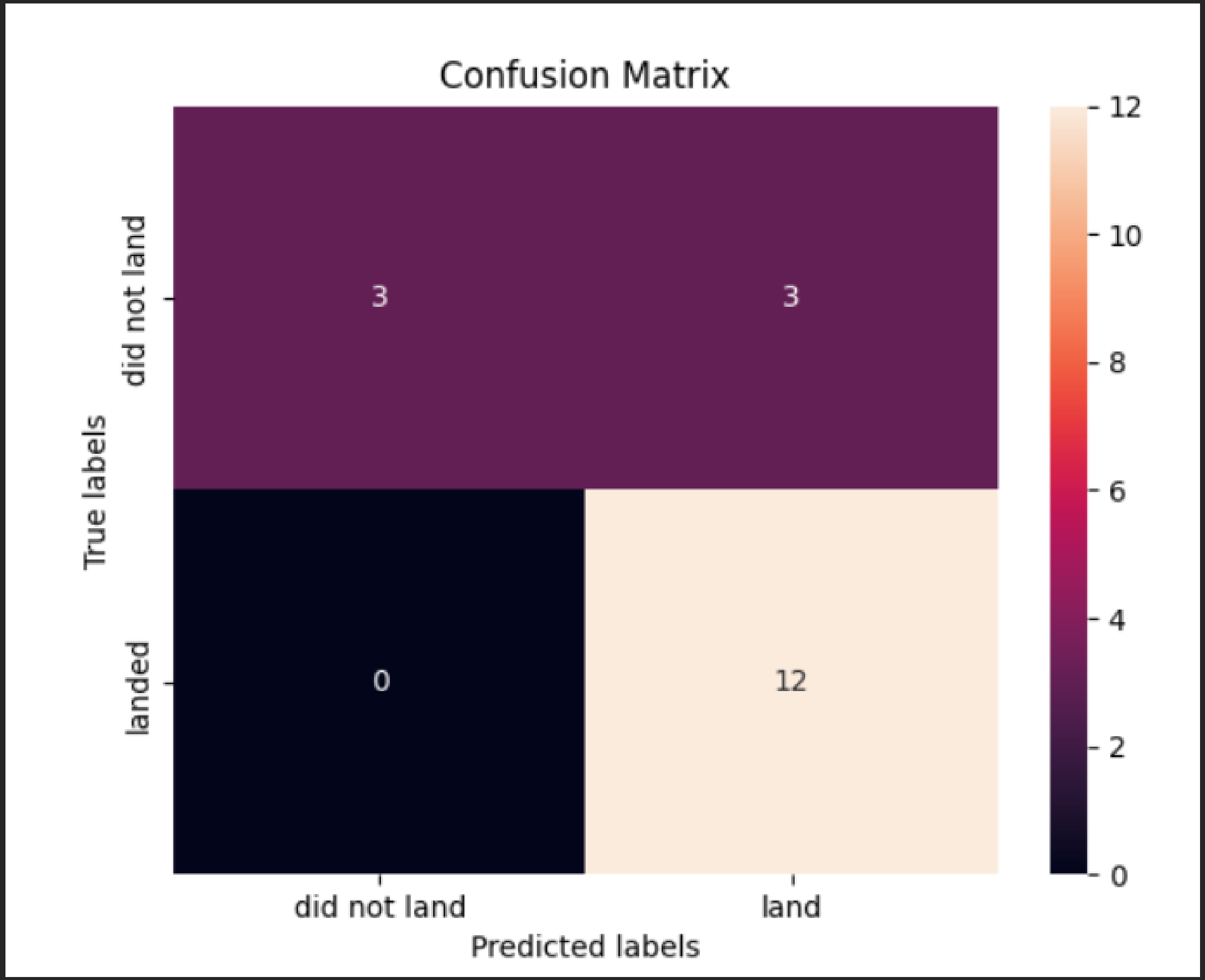
Section 5

Predictive Analysis (Classification)

Classification Accuracy

- As can be seen on this chart, the tree model has the highest accuracy, at slightly over 90%.





Confusion Matrix

- As can be seen by the confusion matrix on the left, there are only a total of three false positives overall, which further affirms that this model is the most accurate of all methods tested.

Conclusions

- Tree method is the most accurate method for Machine Learning classification with an accuracy score of just over 90%.
- Failure risk increases significantly for payloads over 6000 kg.
- KSC LC-39A is the best launch site with a success rate of just around 75% as well as having the highest number of successful launches.
- Despite the peaks and valleys, it can be concluded over the years, the launch success rate is higher.

Thank you!