IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Thi Thanh Chương
14 June 2023

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection with API and Web Scraping
  - Data Wrangling
  - Exploratory Data Analysis with SQL and Data Visualization
  - Interactive Visual Analytics and Dashboards with Folium and Ploty Dash
  - Machine Learning Prediction
- Summary of all results
  - Exploratory Data Analysis
  - Interactive Visual Analytics and Dashboards
  - Find the method performs best using test data

Introduction

- Project background and context

  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. We will also determine if SpaceX will reuse the first stage. Instead of using rocket science to determine if the first stage will land successfully, we will train a machine learning model and use public information to predict if SpaceX will reuse the first stage.

- Problems you want to find answers

  - What factors affect landing results?
  - How does the relationship between each variable affect the outcome?
  - What are the necessary conditions to increase the probability of a successful landing?

Section 1

Methodology

Methodology

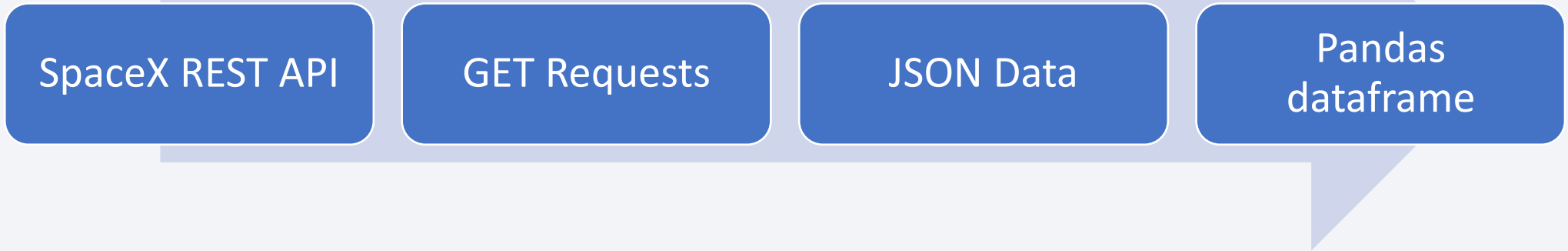
Executive Summary

- Data collection methodology:
  - Data was collected from SpaceX API and web scraping from Wikipedia
- Perform data wrangling
  - The data was processed for categorical features by one-hot encoding.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Build model Machine Learning include SVM, Classification Trees and Logistic Regression.
  - Use GridSearchCV for tune best Hyperparameter for model.
  - Evaluate classification by confusion matrix.

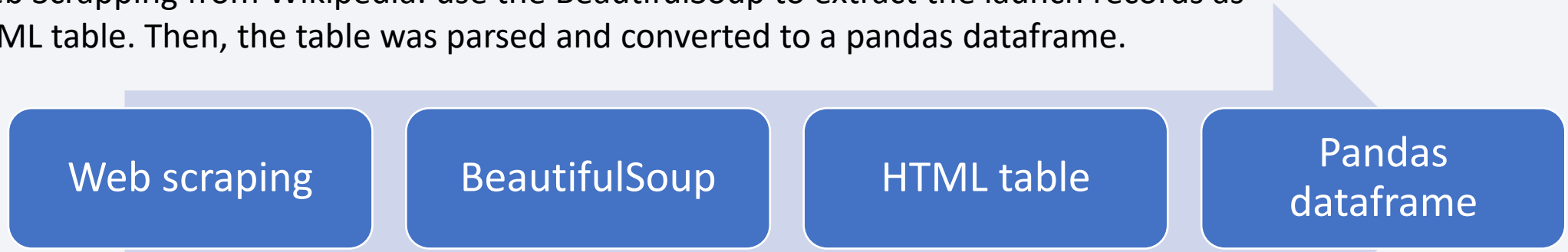
Data Collection

The dataset was collected by REST API and Web Scrapping from Wikipedia.

REST API : get request to the SpaceX API. Response is data in JSON format. Then, the JSON data was converted to a pandas dataframe using json_normalize() function.



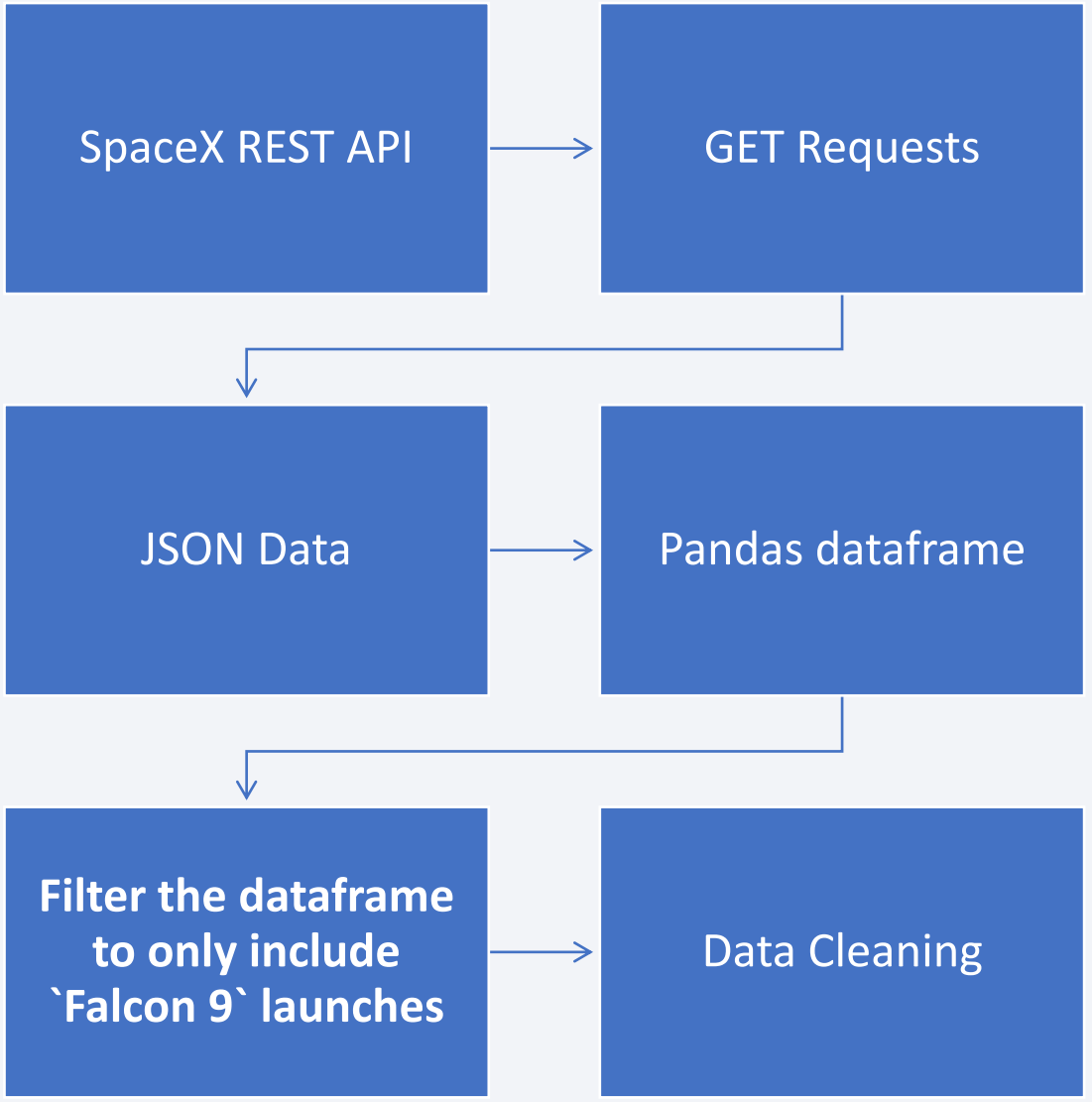
Web Scrapping from Wikipedia: use the BeautifulSoup to extract the launch records as HTML table. Then, the table was parsed and converted to a pandas dataframe.



Data Collection – SpaceX API

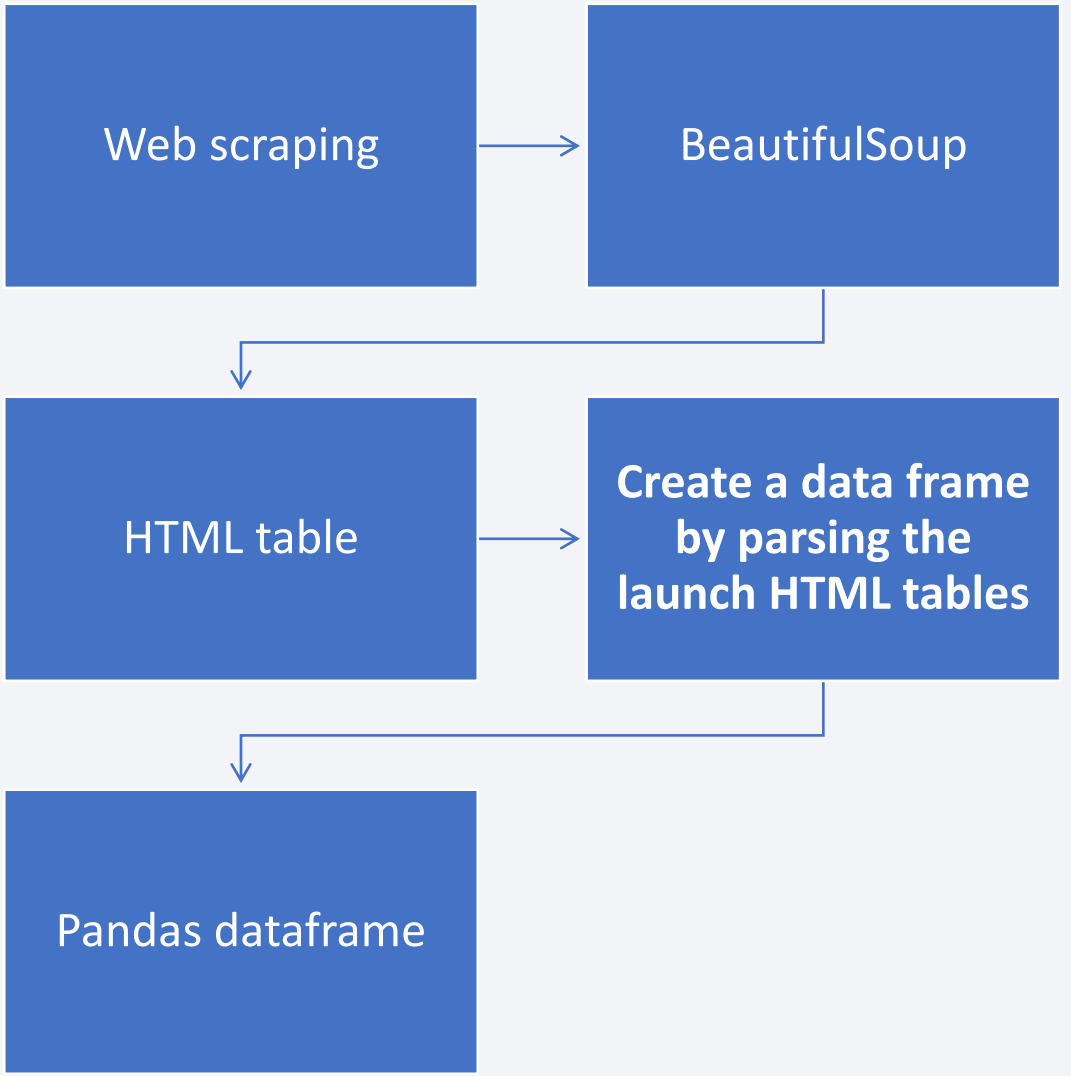
Link Github:

AppliedDataScienceCapstone/jupyter-labs-spacex-data-collection-api.ipynb at main · thichuong/AppliedDataScienceCapstone · GitHub



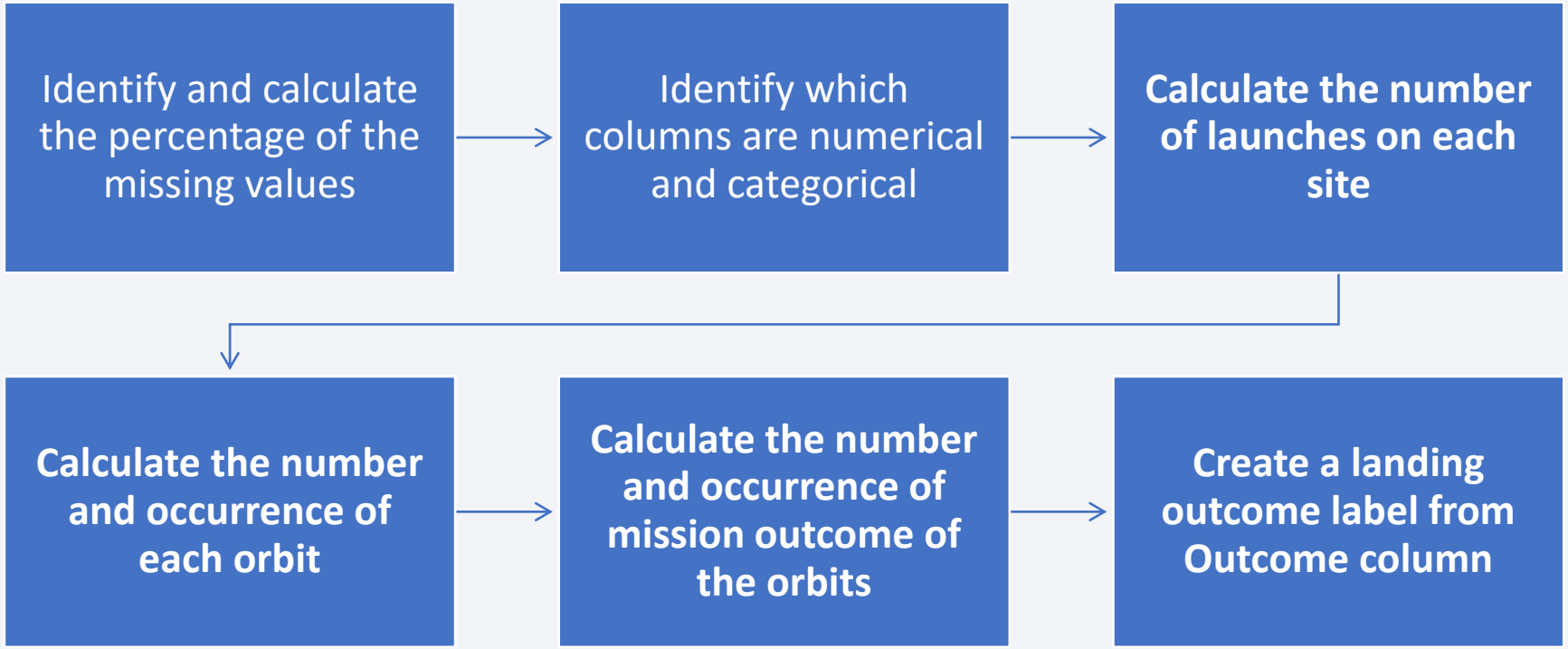
Data Collection - Scraping

Link Github:
[AppliedDataScienceCapstone/jupyter-labs-webscraping.ipynb at main · thichuong/AppliedDataScienceCapstone · GitHub]



Data Wrangling

Perform some Exploratory Data Analysis (EDA) to find patterns in the data and determine what would be the label for training supervised models.
Training Labels with `1` means it was successful. `0` means it was unsuccessful.



Link Github: AppliedDataScienceCapstone/labs-jupyter-spacex-Data wrangling.ipynb at main · thichuong/AppliedDataScienceCapstone · GitHub

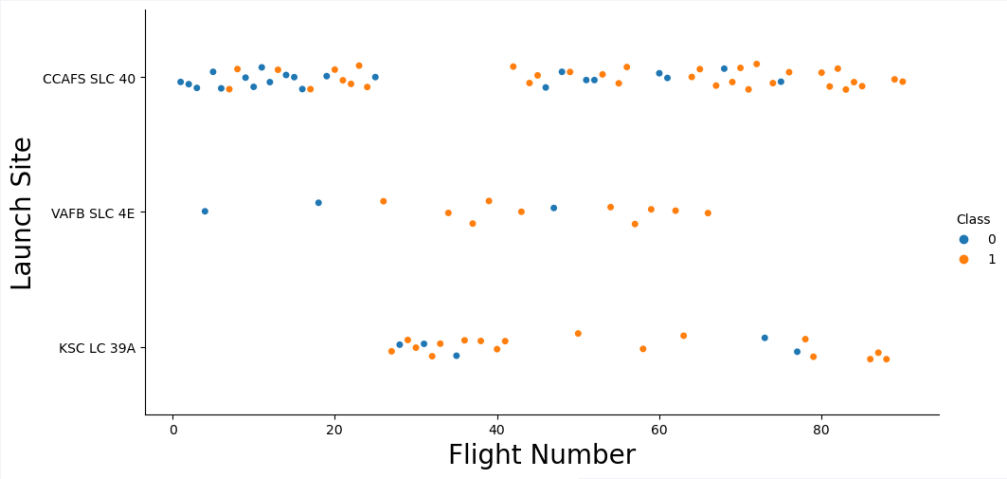
EDA with Data Visualization

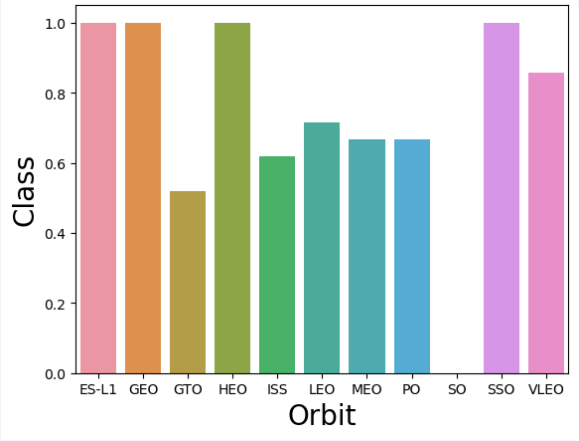
Catplot: `Visualize the relationship`

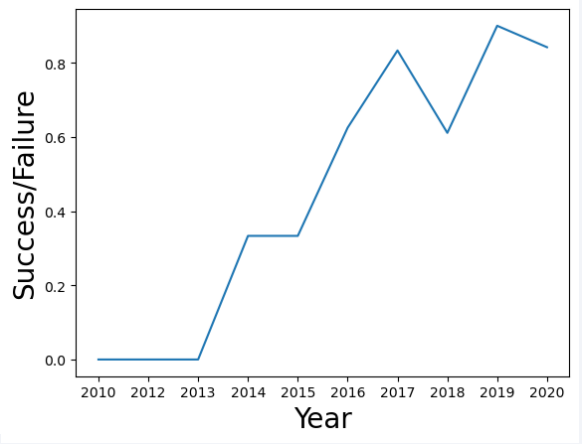
- Between Payload and Flight Number.
- Between Flight Number and Launch Site.
- Between Payload and Launch Site.
- Between Flight Number and Orbit Type.
- Between Payload and Orbit Type.

Barplot: `Visualize the relationship between success rate of each orbit type`

Line chart: `Visualize the launch success yearly trend`







Link Github: [AppliedDataScienceCapstone/jupyter-labs-eda-dataviz.ipynb at main · thichuong/AppliedDataScienceCapstone · GitHub]

EDA with SQL

the SQL queries performed

- Display the names of the unique launch sites in the space mission
  - SELECT DISTINCT "Launch_Site" FROM SPACEXTBL
- Display 5 records where launch sites begin with the string 'CCA’
  - SELECT * FROM SPACEXTBL where "Launch_Site" like "CCA%" LIMIT 5
- Display the total payload mass carried by boosters launched by NASA (CRS)
  - SELECT SUM("PAYLOAD_MASS__KG_") FROM SPACEXTBL where "Customer" like "NASA (CRS)“
- Display average payload mass carried by booster version F9 v1.1
  - SELECT AVG("PAYLOAD_MASS__KG_") FROM SPACEXTBL where "Booster_Version" like "F9 v1.1“
- List the date when the first succesful landing outcome in ground pad was achieved
  - SELECT MIN(Date) FROM SPACEXTBL where Landing_Outcome = "Success (ground pad)"
- List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
  - SELECT "Booster_Version", "PAYLOAD_MASS__KG_" FROM SPACEXTBL WHERE ("PAYLOAD_MASS__KG_" BETWEEN 4000 AND 6000) AND "Landing_Outcome" = "Success (drone ship)"

Link Github: AppliedDataScienceCapstone/jupyter-labs-eda-sql-coursera_sqllite.ipynb at main · thichuong/AppliedDataScienceCapstone · GitHub

EDA with SQL

the SQL queries performed

- List the total number of successful and failure mission outcomes.
  - SELECT Mission_Outcome, Count(*) FROM SPACEXTBL GROUP BY Mission_Outcome
- Listing the names of the booster_versions which have carried the maximum payload mass.
  - SELECT DISTINCT Booster_Version FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL)
- List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015..
  - SELECT substr(Date, 4, 2) AS Month, Landing_Outcome, Booster_Version, Launch_Site FROM SPACEXTBL WHERE substr(Date,7,4)='2015' AND Landing_Outcome = 'Failure (drone ship)'
- Rank the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.
  - SELECT Landing_Outcome, COUNT(Landing_Outcome) as Total FROM SPACEXTBL WHERE (Date BETWEEN '04/06/2010' AND '20/03/2017') AND Landing_Outcome LIKE 'Success%' GROUP BY Landing_Outcome ORDER BY Total DESC

Link Github: AppliedDataScienceCapstone/jupyter-labs-eda-sql-coursera_sqllite.ipynb at main · thichuong/AppliedDataScienceCapstone · GitHub

Build an Interactive Map with Folium

Summarize what map objects:

- folium.Circle to add a highlighted circle area with a text label on a specific coordinate
- folium.Marker to add a highlighted circle area with a text label on a specific coordinate for each launch site on the site map
- MarkerCluster to simplify a map containing many markers having the same coordinate.
- MousePosition to get coordinate for a mouse over a point on the map
- Folium.PolyLine to draw a PolyLine between a launch site to the selected coastline point

Link Github: AppliedDataScienceCapstone/lab_jupyter_launch_site_location.ipynb at main · thichuong/AppliedDataScienceCapstone · GitHub

Build a Dashboard with Plotly Dash

This dashboard application contains input components such as a dropdown list and a range slider to interact with a pie chart and a scatter point chart.

- **Dropdown list** to enable Launch Site selection
- **Slider** to select payload range
- **Plotted pie charts**: showing the total launches by a certain sites
- **Plotted scatter graph:** showing the relationship with Outcome and Payload Mass (Kg) for the different booster version.

Link Github: AppliedDataScienceCapstone/spacex_dash_app.py at main · thichuong/AppliedDataScienceCapstone · GitHub

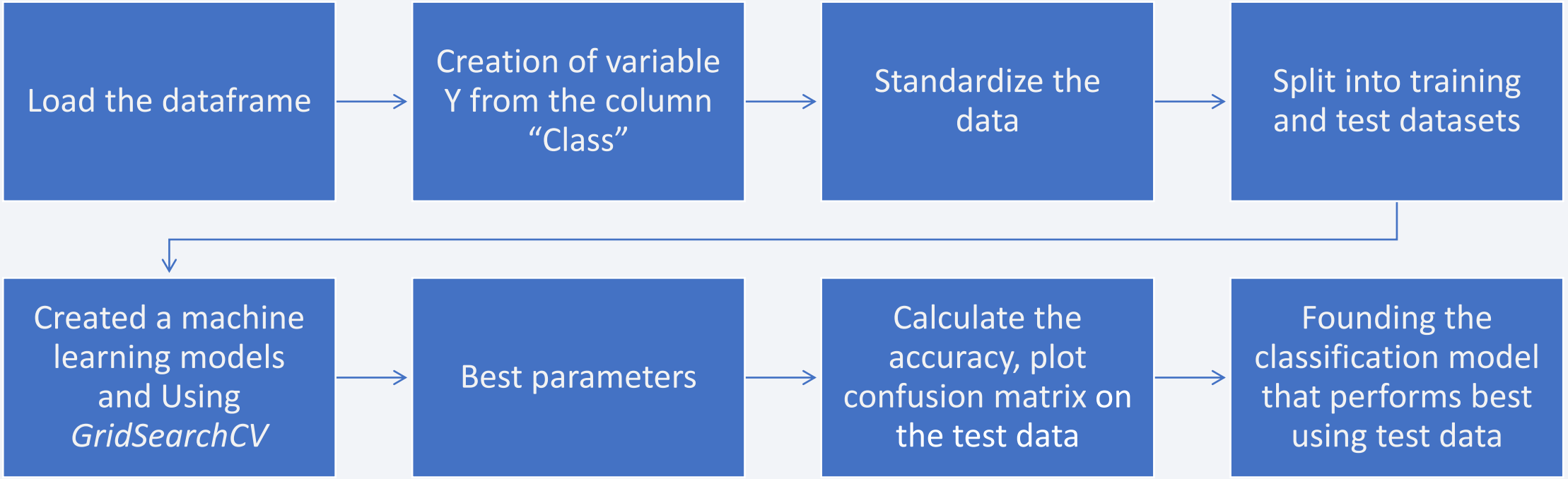
Predictive Analysis (Classification)

Summarize of the model developments:

- Load the dataframe
- Standardize the data
- Split into training and test datasets
- Created a machine learning models
- Using *GridSearchCV*, found the best machine learning method for predictions.
- Founding the classification model that performs best using test data

Link Github: [AppliedDataScienceCapstone/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb at main · thichuong/AppliedDataScienceCapstone · GitHub]

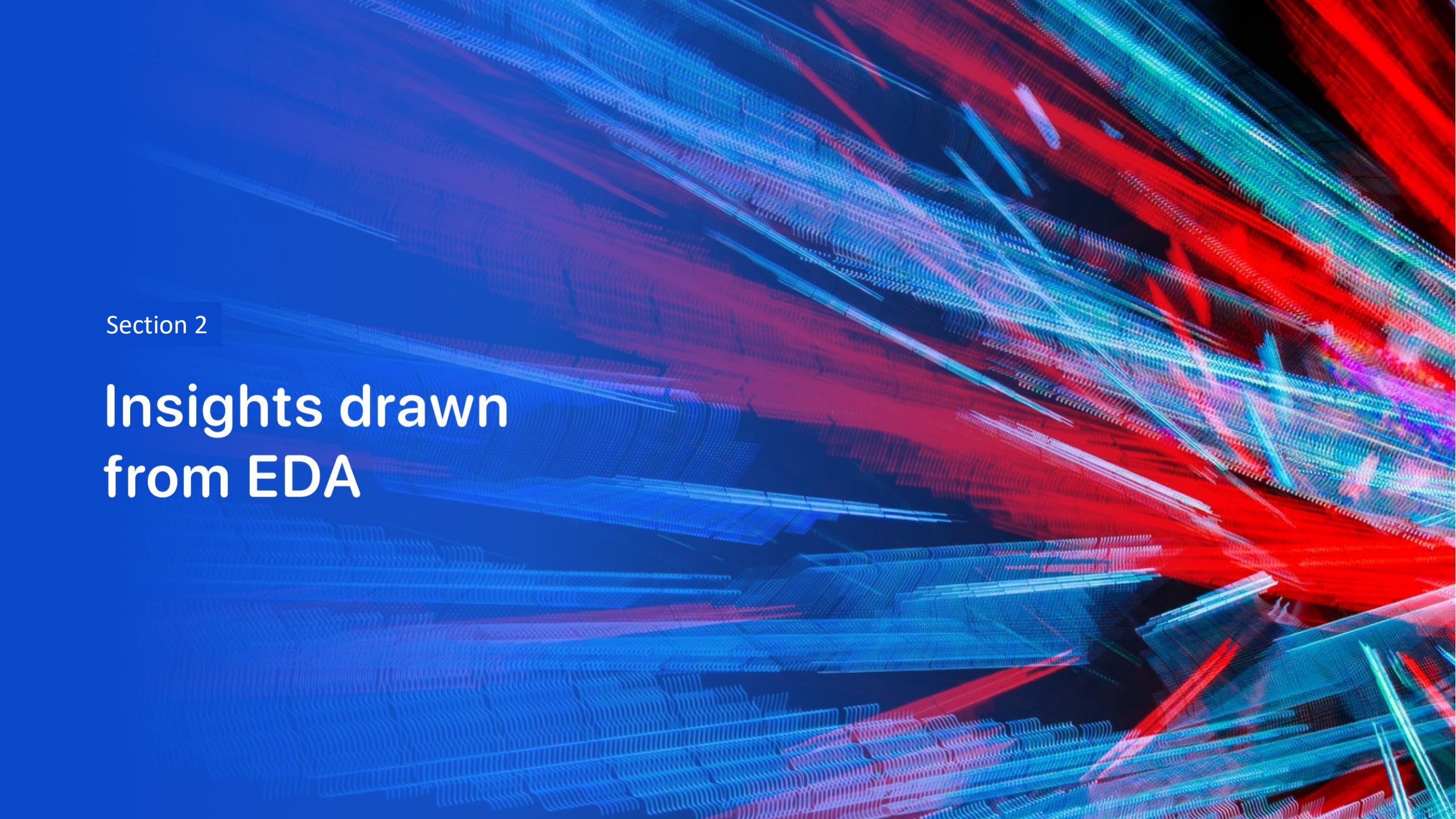
Predictive Analysis (Classification)



Link Github: [AppliedDataScienceCapstone/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb at main · thichuong/AppliedDataScienceCapstone · GitHub]

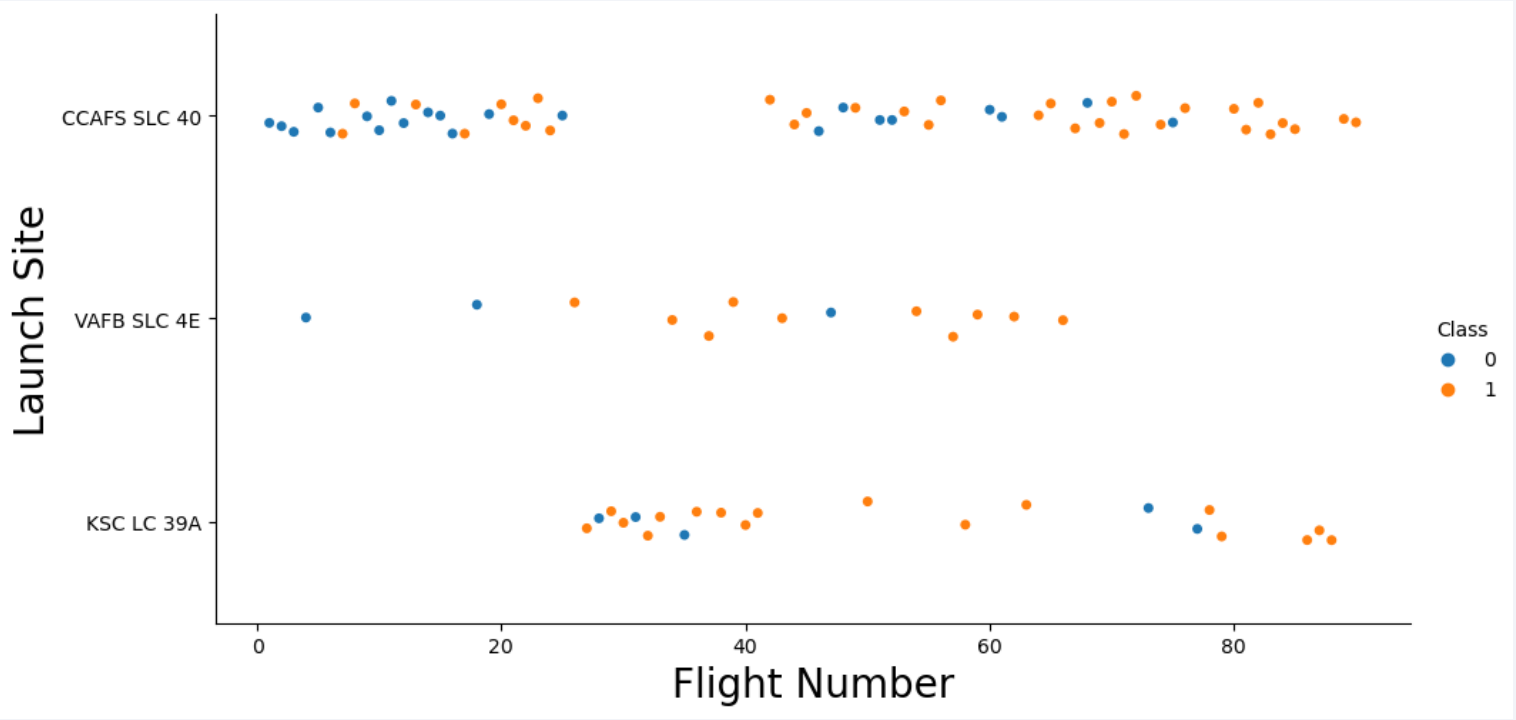
Results

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


Section 2
Insights drawn from EDA

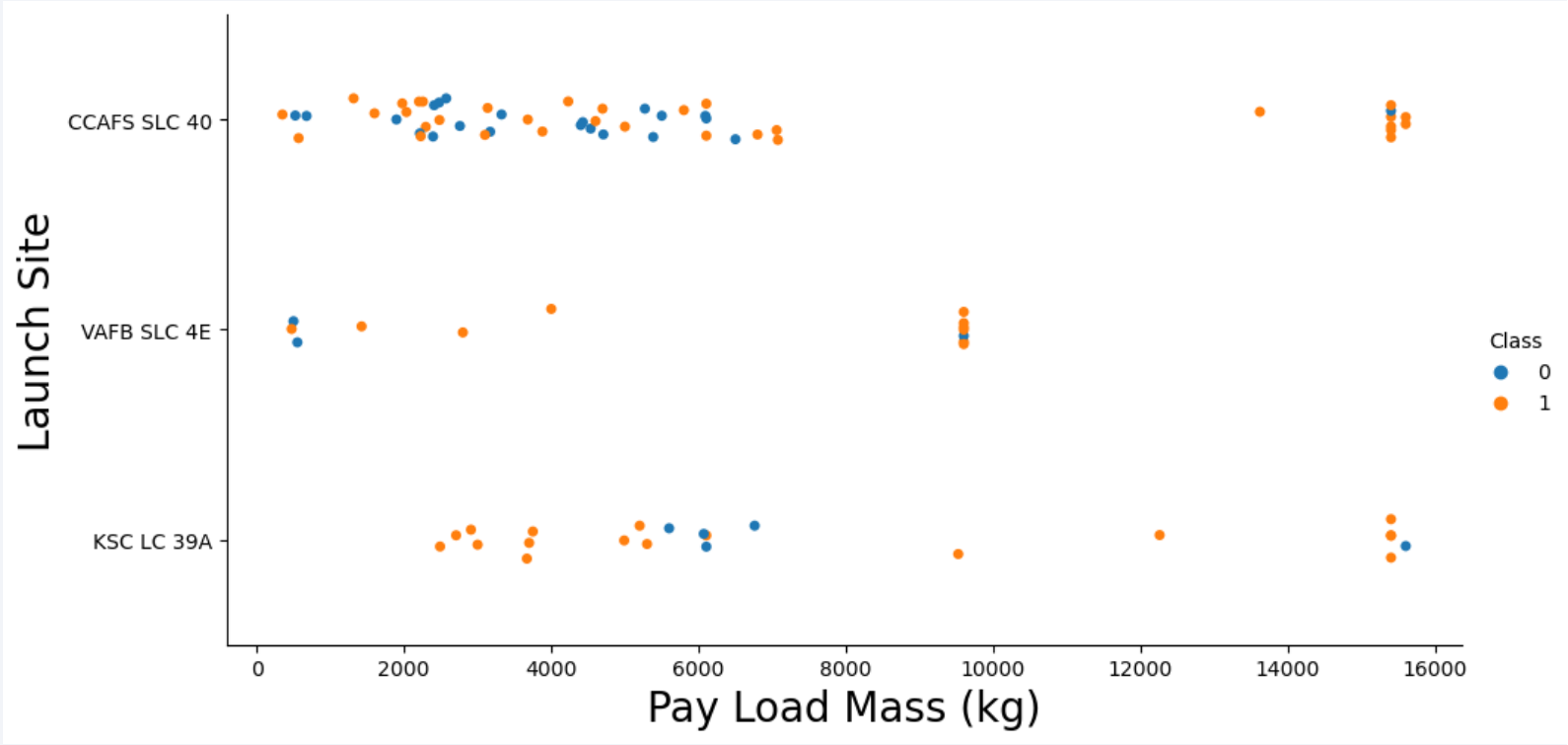
Flight Number vs. Launch Site

- This scatter plot shows that there were more successful landings as the flight numbers increased.



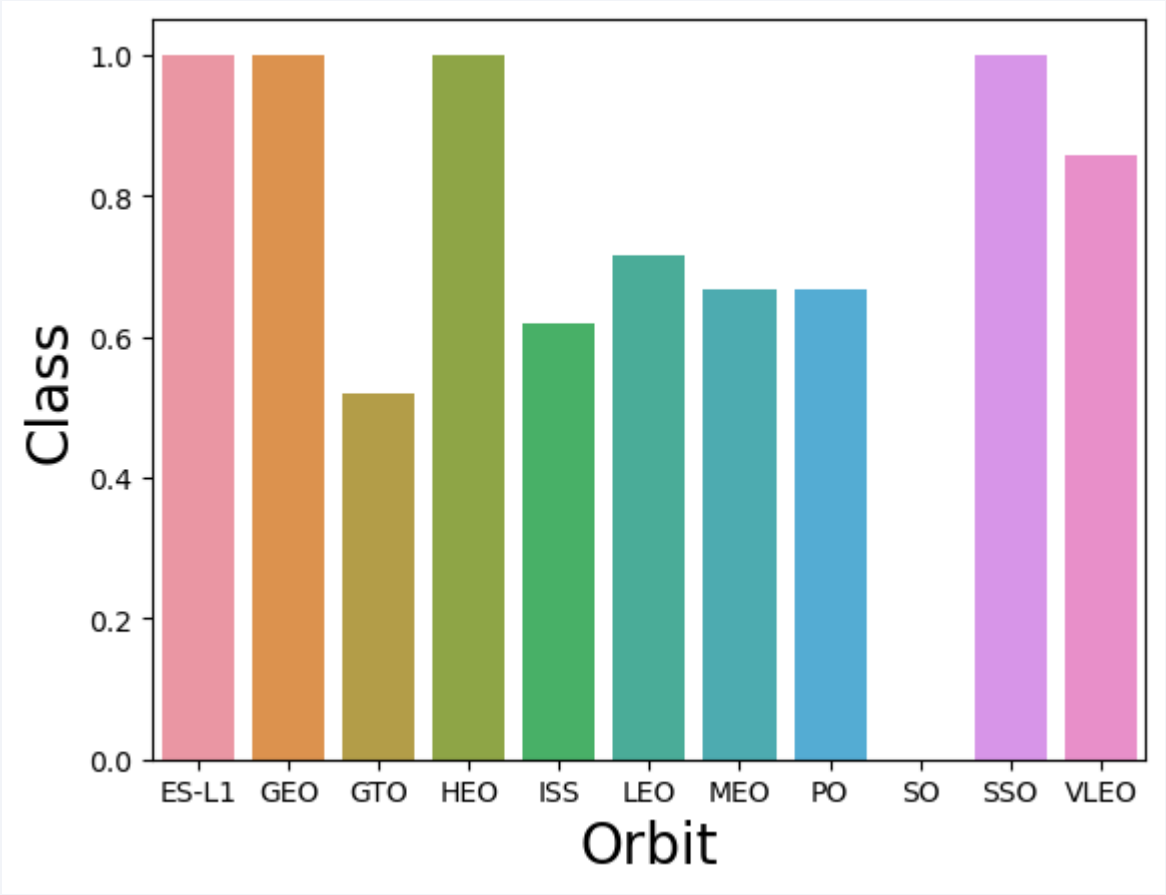
Payload vs. Launch Site

- The probability of the success rate will be highly increased when Payload Mass increased .
- the VAFB-SLC launch site there are no rockets launched for heavy payload mass(greater than 10000).



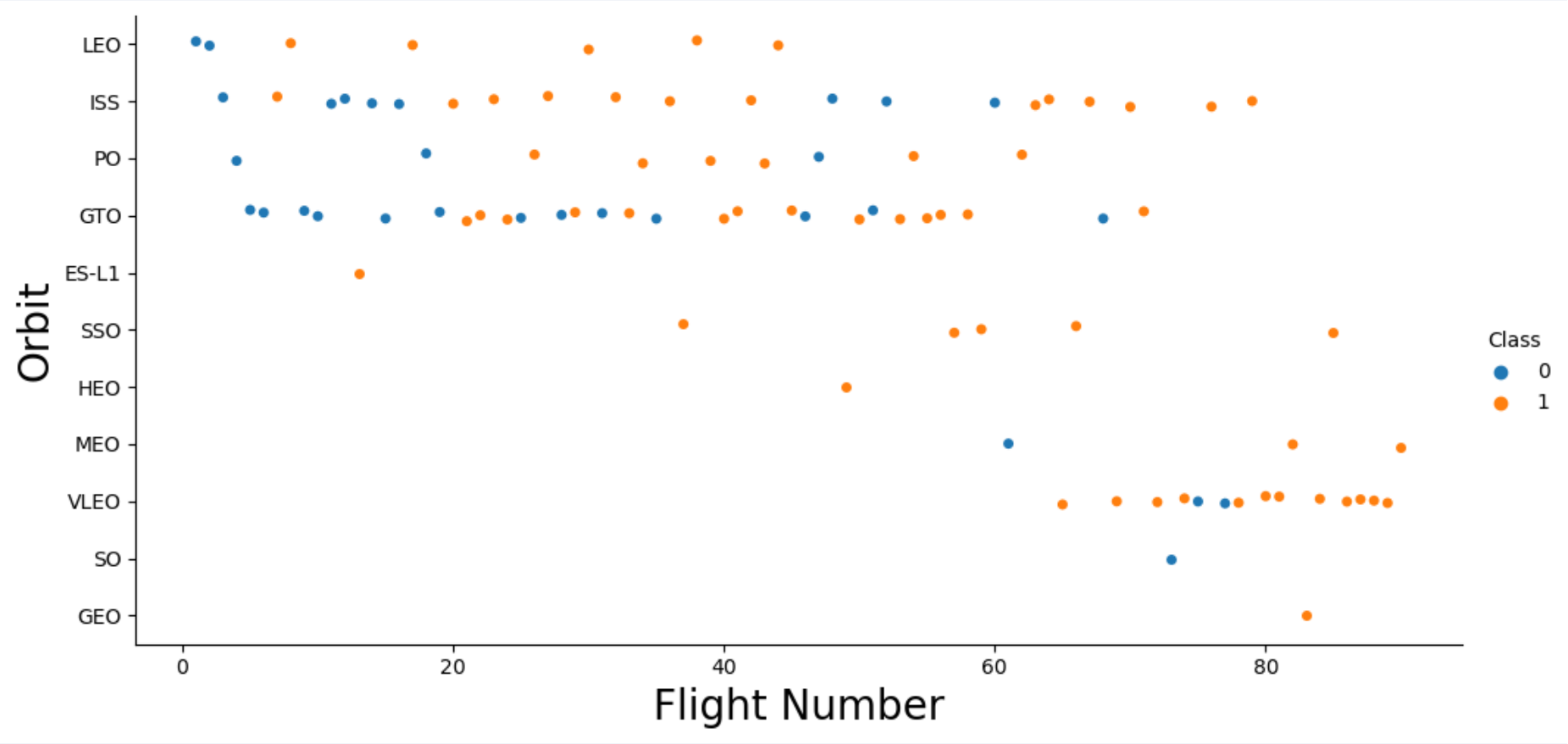
Success Rate vs. Orbit Type

- Some orbits has 100% success rate such as SSO, HEO, GEO AND ES-L1 while SO orbit produced 0% rate of success.



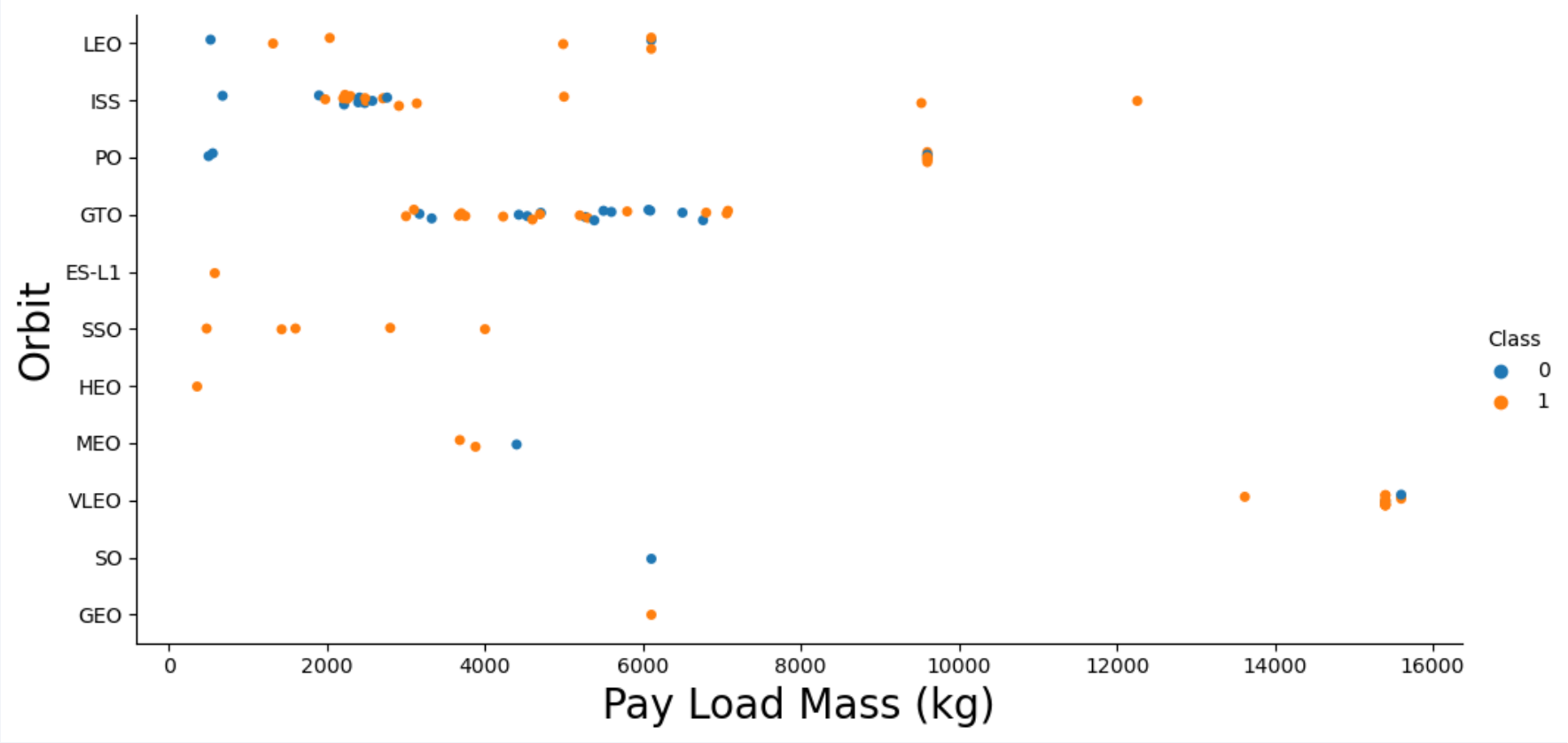
Flight Number vs. Orbit Type

- the Success rate related to the number of flights for each orbit



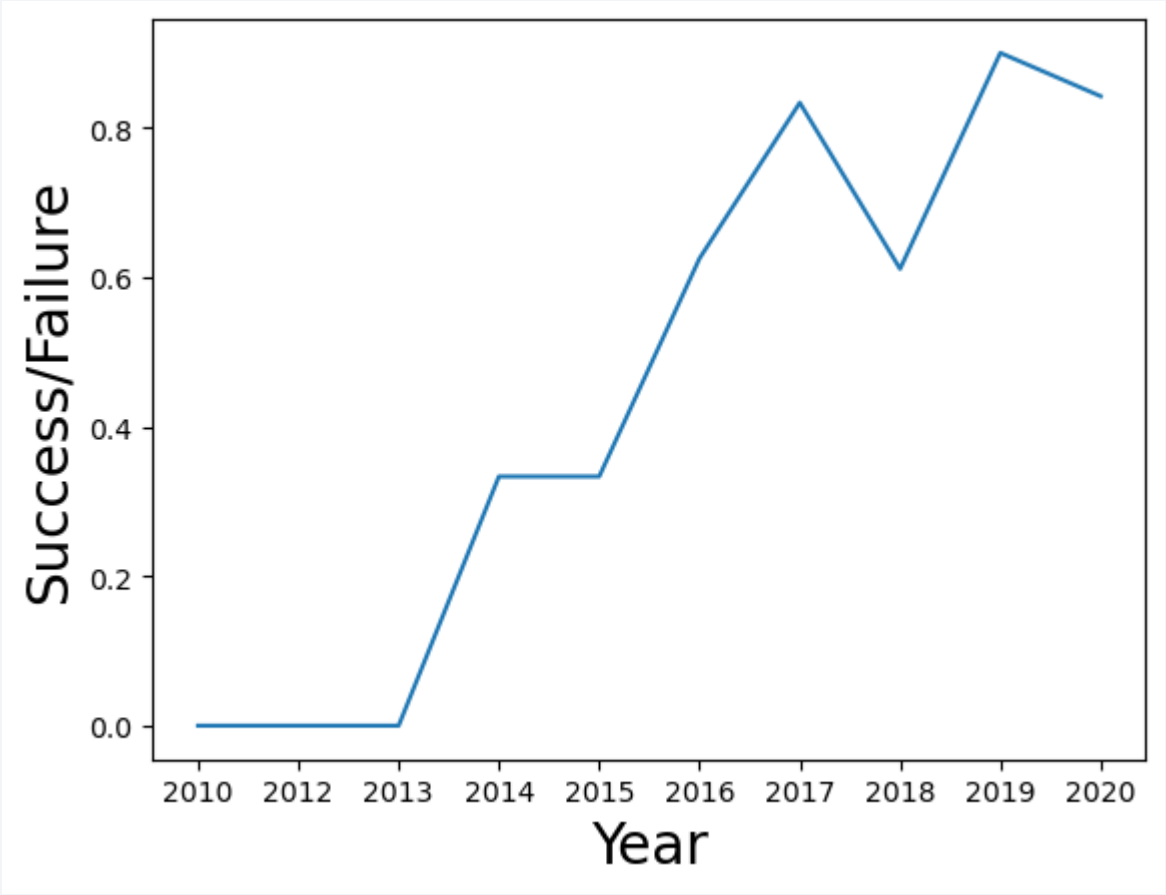
Payload vs. Orbit Type

- With heavy payloads, the successful landing are more for PO, LEO and ISS orbits.



Launch Success Yearly Trend

- the success rate has increased from 2013 to 2020.



All Launch Site Names

- used the **DISTINCT** statement to find only unique launch sites from the SpaceX data.

```
1  %%sql
2  SELECT DISTINCT "Launch_Site" FROM SPACEXTBL
```

✓ 0.0s

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

Launch_Site
CCAFS LC-40
VAFB SLC-4E
KSC LC-39A
CCAFS SLC-40
None

Launch Site Names Begin with 'CCA'

```
# Display 5 records where launch sites begin with the string 'CCA'
data = cur.execute('SELECT * FROM SPACEXTBL where "Launch_Site" like "CCA%" LIMIT 5').fetchall()
pd.DataFrame(data)
```

✓ 0.0s

	0	1	2	3	4	5	6	7	8	9
0	06/04/2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0.0	LEO	SpaceX	Success	Failure (parachute)
1	12/08/2010	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of...	0.0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2	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
3	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
4	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

- We used the query to display 5 records where launch sites begin with `CCA`

Total Payload Mass

```
1  data = cur.execute('SELECT SUM("PAYLOAD_MASS__KG_") FROM SPACEXTBL where "Customer" like "NASA (CRS)"').fetchall()
2  data
```

✓ 0.0s

```
[(45596.0,)]
```

- the total payload carried by boosters from NASA using SUM function as 45596

Average Payload Mass by F9 v1.1

```
1  data = cur.execute('SELECT AVG("PAYLOAD_MASS__KG_") FROM SPACEXTBL where "Booster_Version" like "F9 v1.1"').fetchall()
2  data
```

✓ 0.0s

```
[(2928.4,)]
```

- The average payload mass carried by F9 v1.1 was 2928.4 kg

First Successful Ground Landing Date

```
1  data = cur.execute('SELECT MIN(Date) FROM SPACEXTBL where Landing_Outcome = "Success (ground pad)"').fetchall()
2  data
```

✓ 0.0s

```
[('01/08/2018',)]
```

- The first successful landing outcome on ground pad was 1st August 2018.

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%%sql
SELECT "Booster_Version", "PAYLOAD_MASS__KG_" FROM SPACEXTBL
WHERE ("PAYLOAD_MASS__KG_" BETWEEN 4000 AND 6000) AND "Landing_Outcome" = "Success (drone ship)"
```

✓ 0.0s

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

Booster_Version	PAYLOAD_MASS__KG_
F9 FT B1022	4696.0
F9 FT B1026	4600.0
F9 FT B1021.2	5300.0
F9 FT B1031.2	5200.0

- Using the **WHERE** clause and the AND operater.

Total Number of Successful and Failure Mission Outcomes

```
1  %%sql
2  SELECT Mission_Outcome, Count(*) FROM SPACEXTBL GROUP BY Mission_Outcome
```

✓ 0.0s

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

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

- Using COUNT function and GROUP BY statement.

Boosters Carried Maximum Payload

- using a subquery in the WHERE clause and the MAX() function.

```
%%sql
SELECT DISTINCT Booster_Version, PAYLOAD_MASS__KG_ FROM SPACEXTBL
WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL)
```

✓ 0.0s

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

Booster_Version	PAYLOAD_MASS__KG_
F9 B5 B1048.4	15600.0
F9 B5 B1049.4	15600.0
F9 B5 B1051.3	15600.0
F9 B5 B1056.4	15600.0
F9 B5 B1048.5	15600.0
F9 B5 B1051.4	15600.0
F9 B5 B1049.5	15600.0
F9 B5 B1060.2	15600.0
F9 B5 B1058.3	15600.0
F9 B5 B1051.6	15600.0
F9 B5 B1060.3	15600.0
F9 B5 B1049.7	15600.0

2015 Launch Records

```
%%sql
SELECT substr(Date, 4, 2) AS Month, Booster_Version, Launch_Site FROM SPACEXTBL
WHERE substr(Date,7,4)='2015' AND Landing_Outcome = 'Failure (drone ship)'
```

✓ 0.0s

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

Month	Booster_Version	Launch_Site
10	F9 v1.1 B1012	CCAFS LC-40
04	F9 v1.1 B1015	CCAFS LC-40

- List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.
- Using WHERE clause, AND, and “substr”.

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

- Rank the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.
- Using COUNT, WHERE, BETWEEN, AND, GROUP BY, ORDER BY

```
%%sql
SELECT Landing_Outcome,COUNT(Landing_Outcome) as Total FROM SPACEXTBL
WHERE (Date BETWEEN '04/06/2010' AND '20/03/2017') AND Landing_Outcome like 'Success%'
GROUP BY Landing_Outcome
ORDER BY Total DESC

```

✓ 0.0s

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

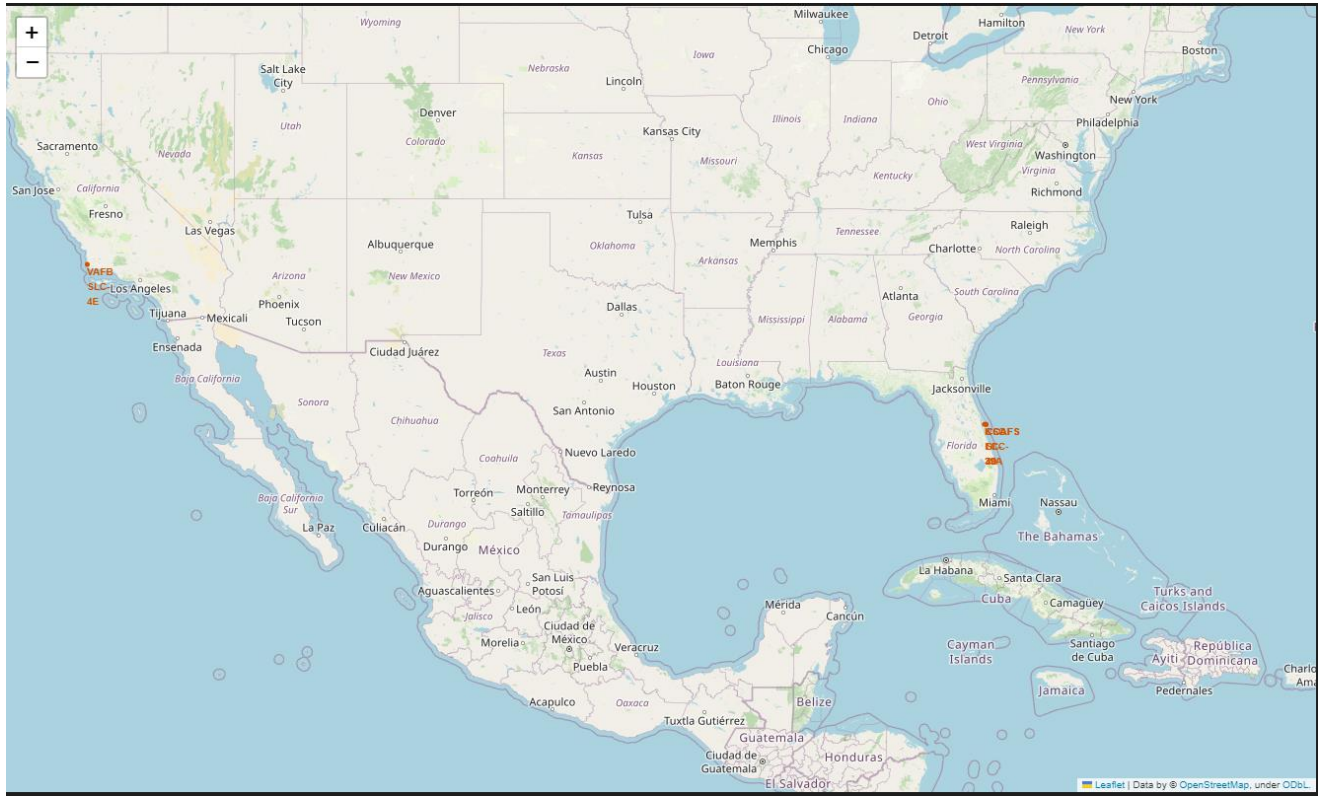
Landing_Outcome	Total
Success	20
Success (drone ship)	8
Success (ground pad)	7

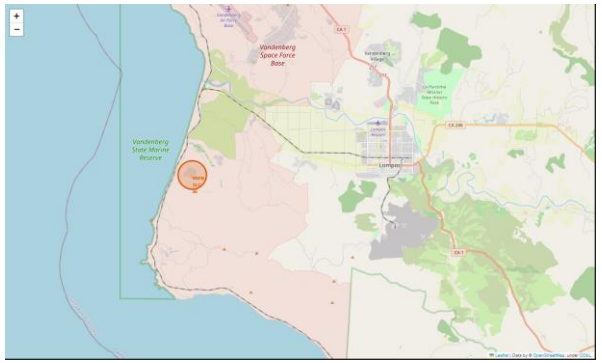
Section 3

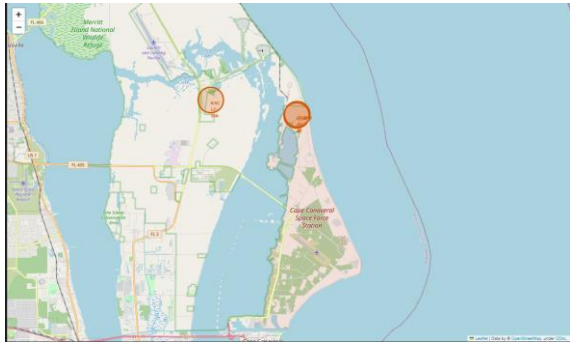
Launch Sites Proximities Analysis

All launch sites locations

We can see that all the SpaceX launch sites

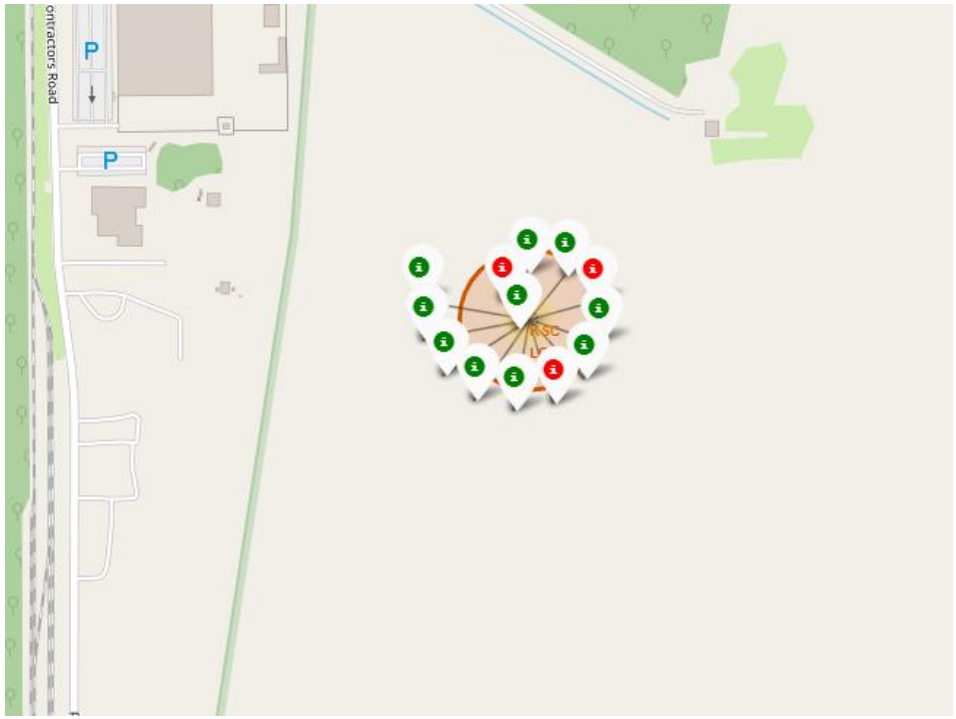


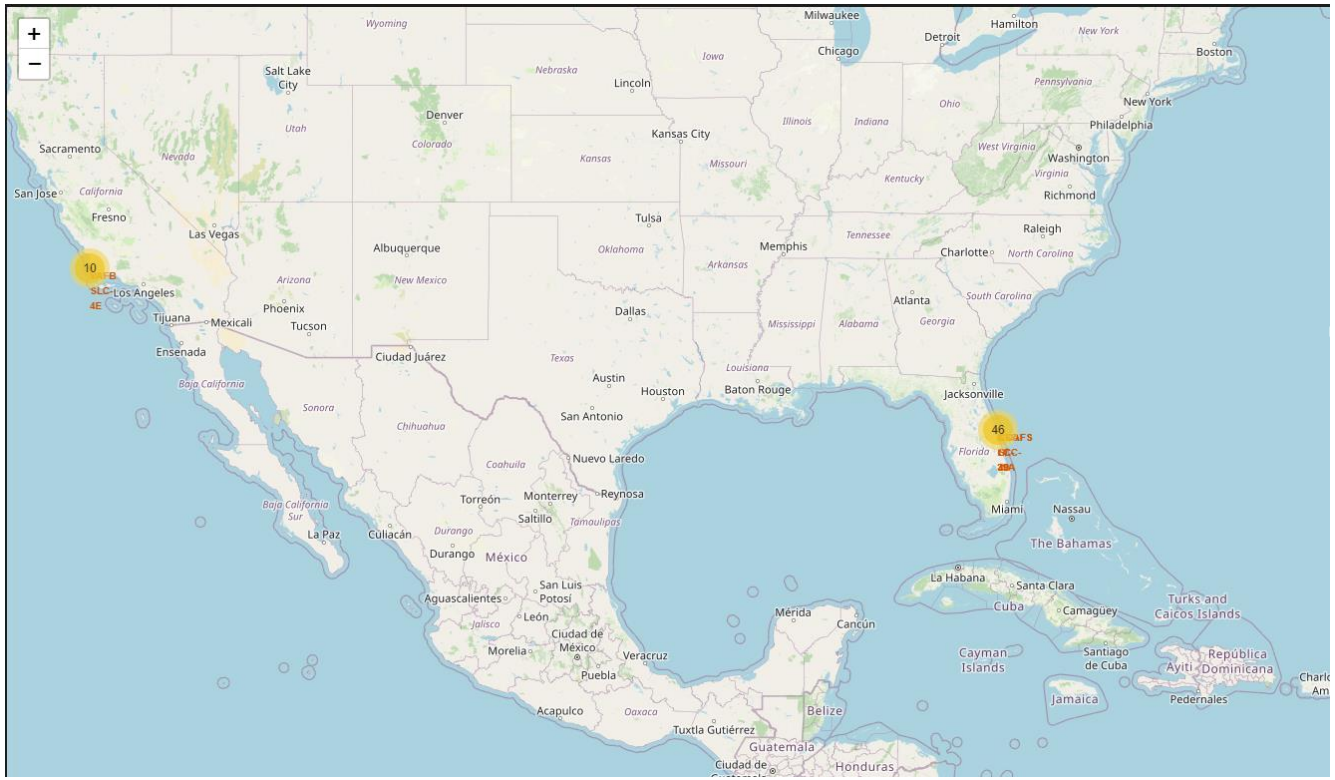




the success/failed launches for each site

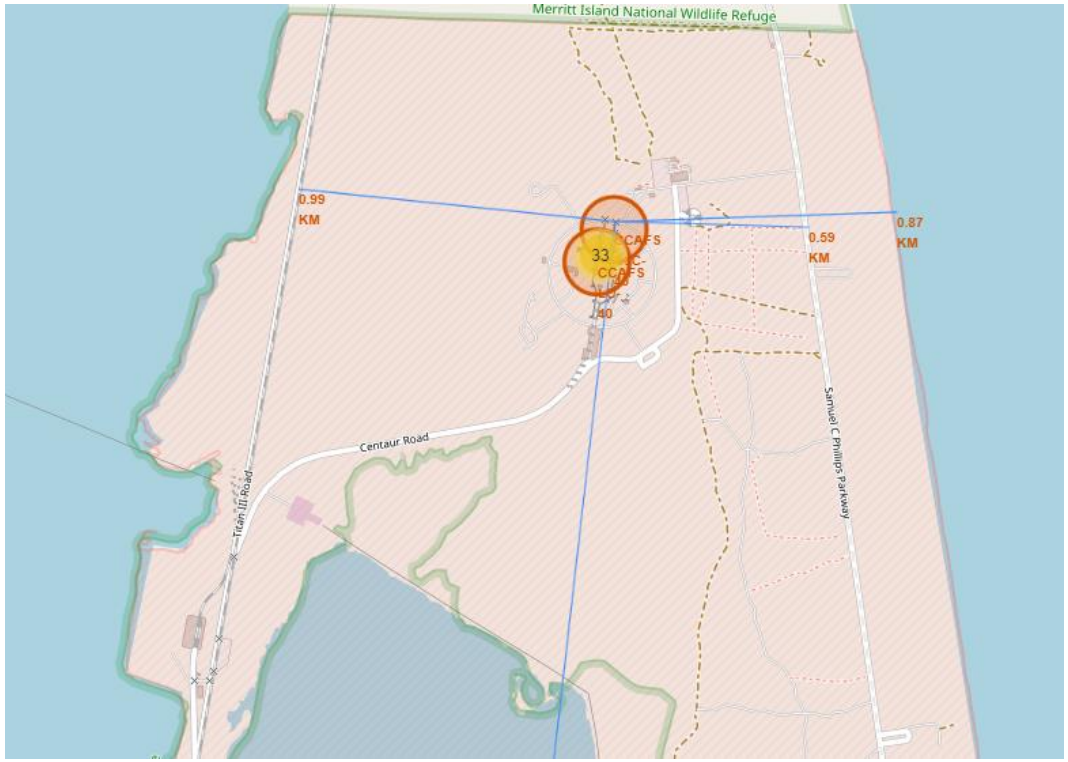
shows a green marker if a launch was successful, and a red marker if a launch was failed.

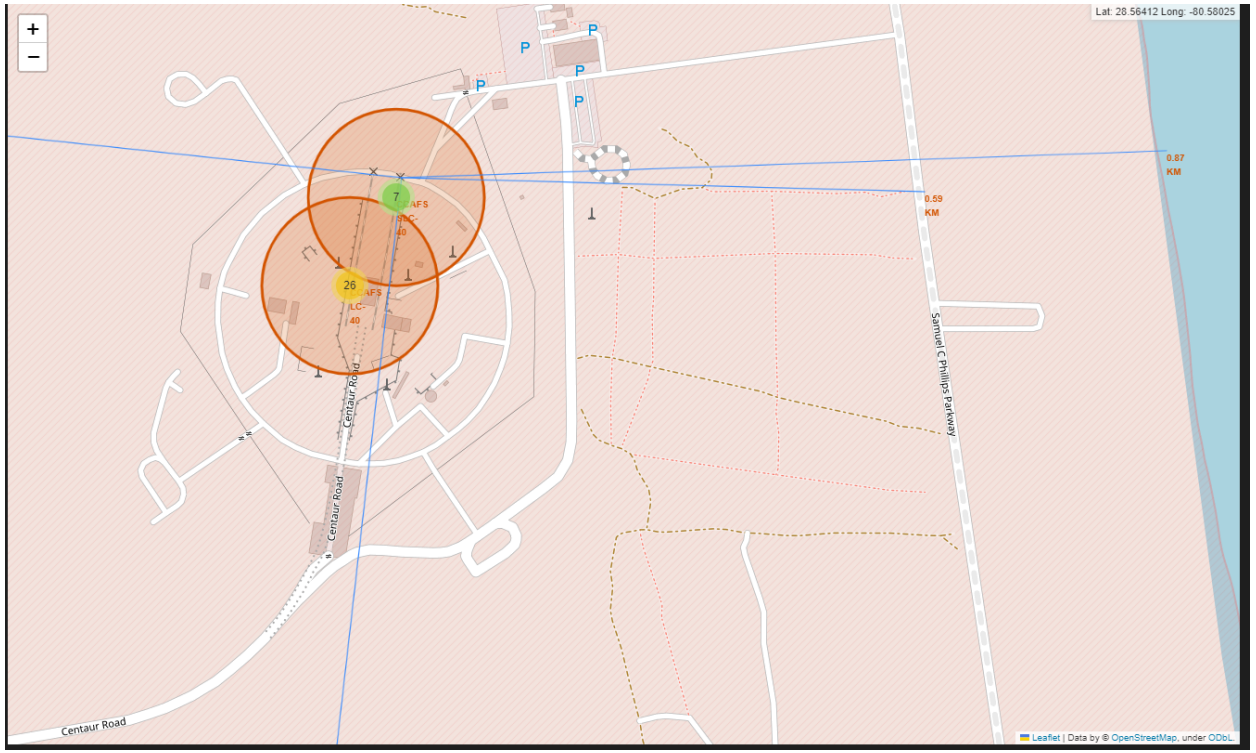


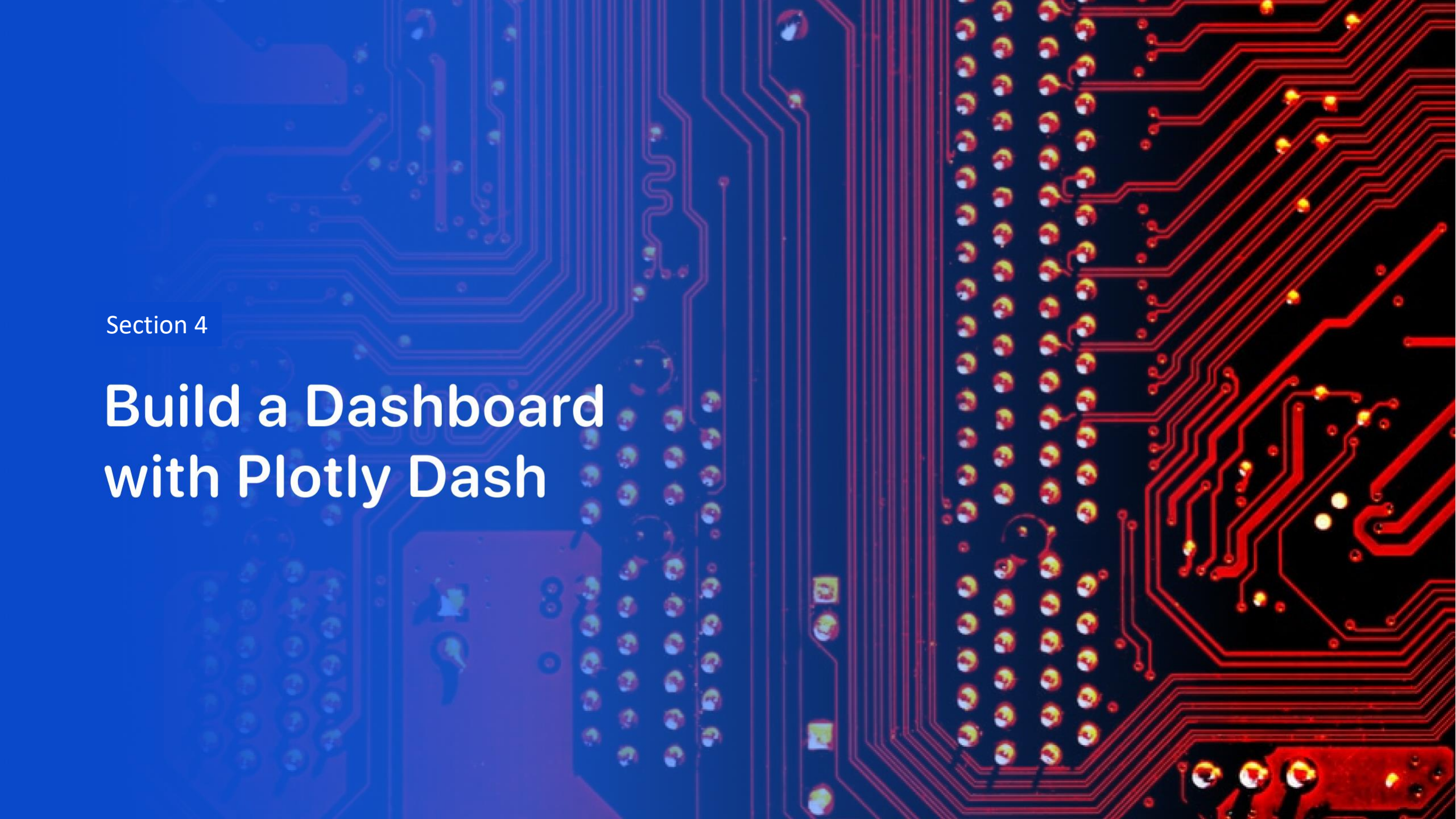


Launch site to its proximities

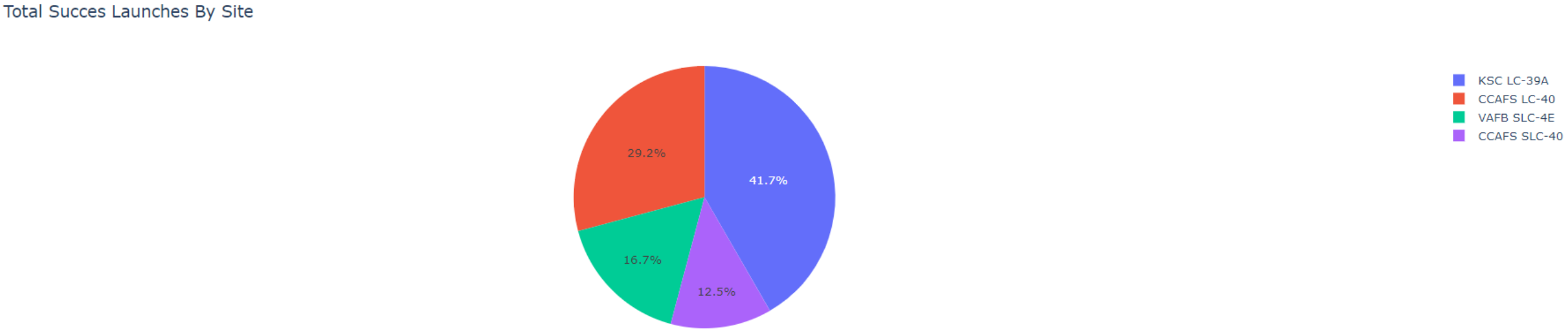
- Draw a line between a launch site to its closest city, railway, highway





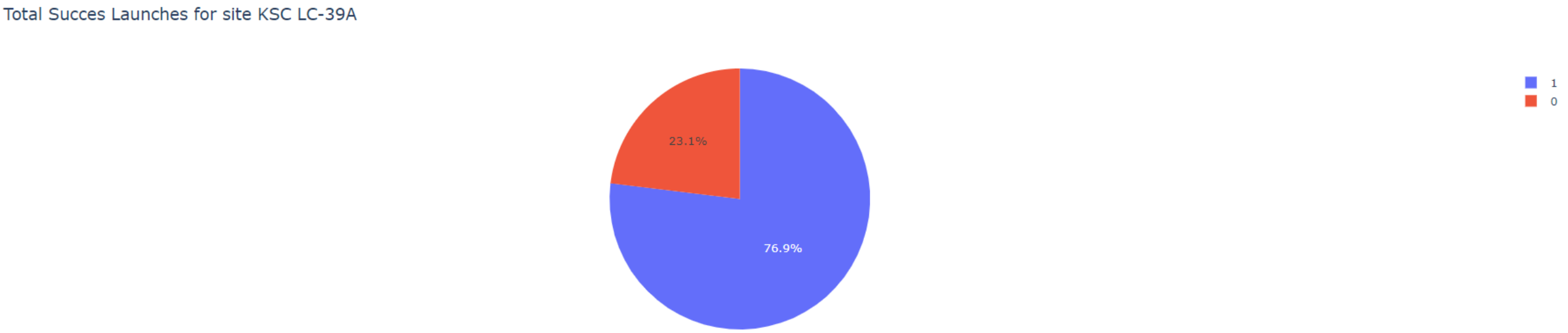

Section 4
Build a Dashboard with Plotly Dash

Total Successful Launches By Site



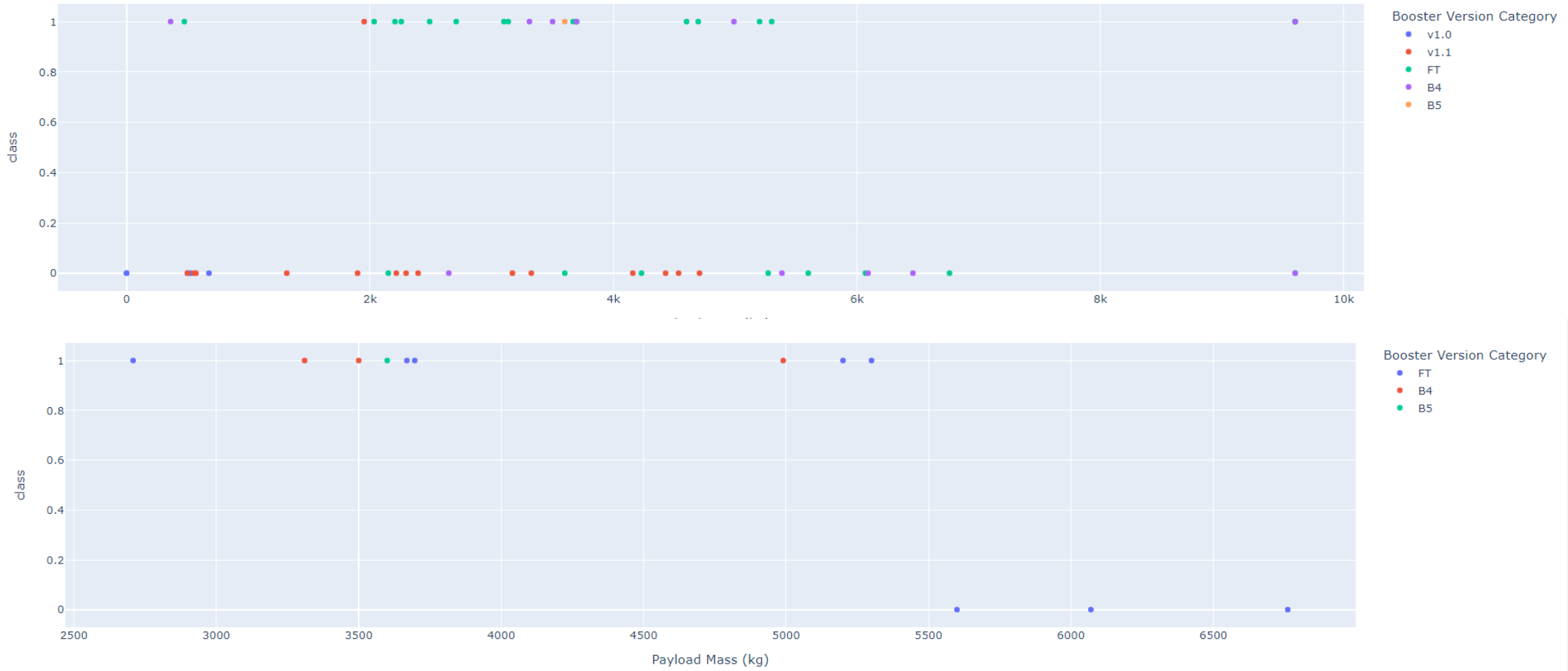
- The KSC LC-39A has the highest total launch success from all the site

Highest launch success ratio



- 76.9% of the total launches at site KSC LC-39A were successful

Payload vs. Launch Outcome scatter plot for all sites



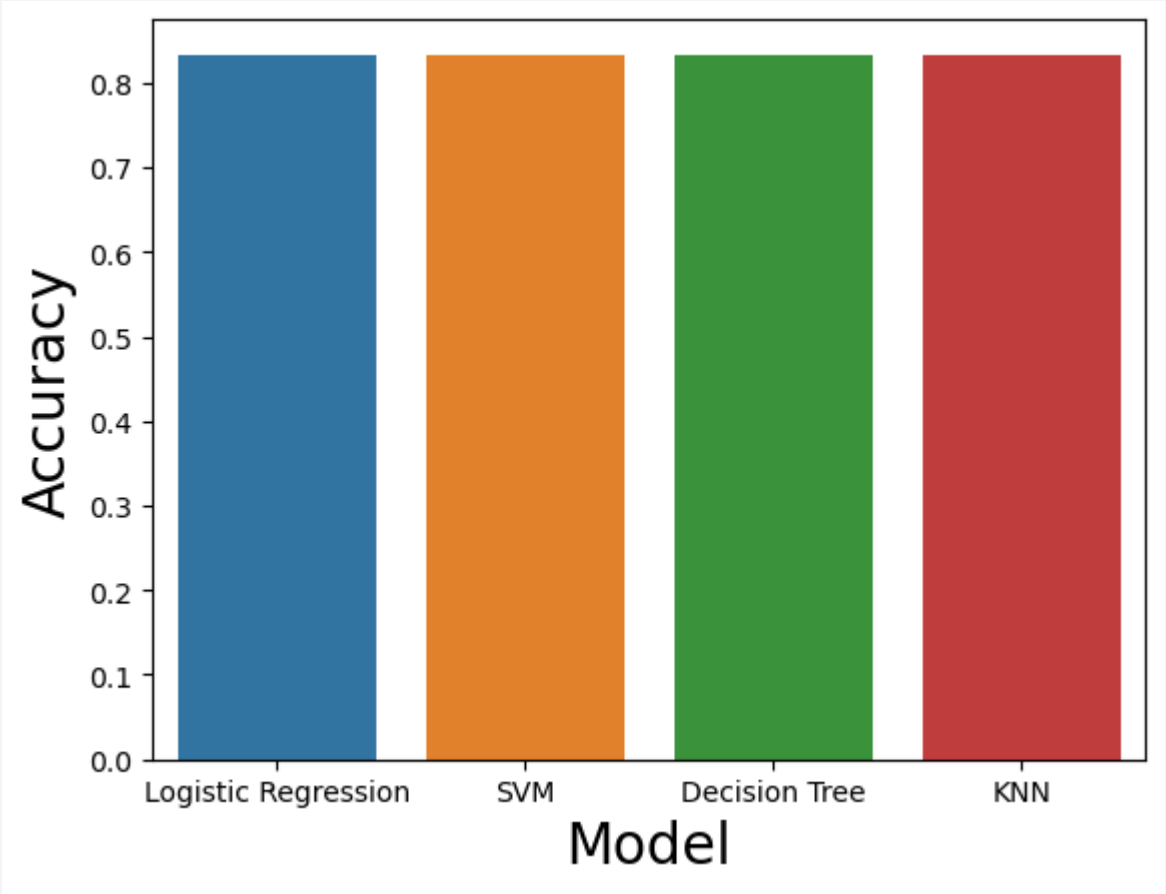
- the success rate for low weighted payload is higher than heavy weighted payload

Section 5

Predictive Analysis (Classification)

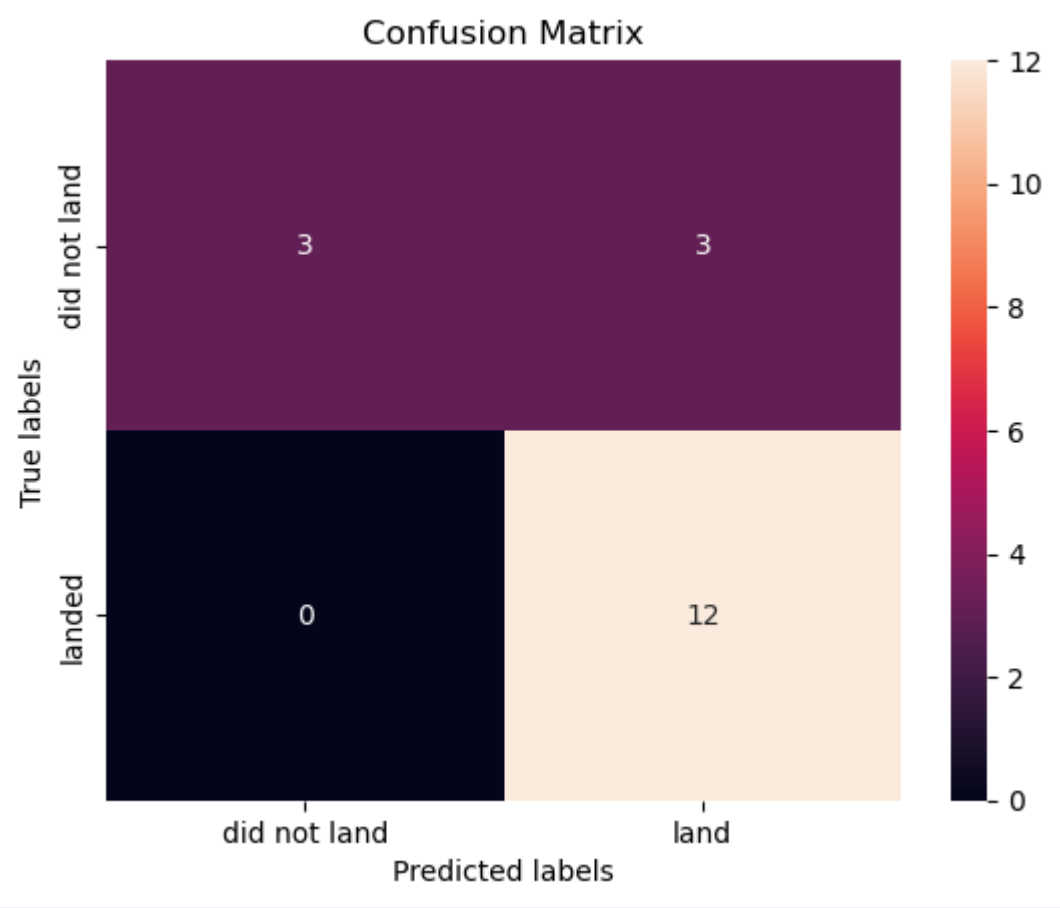
Classification Accuracy

- All the models have an accuracy score of 83.33% using test data.



Confusion Matrix

- All models have identical Confusion Matrix
- Have 3 False Positive.



Conclusions

- All Machine Learning Algorithm have the same accuracy score.
- The KSC LC-39A has the highest total launch success from all the site
- the success rate has increased from 2013 to 2020.
- the Success rate related to the number of flights for each orbit.

Appendix

- For Python code snippets, SQL queries, charts, Notebook outputs, and data sets, follow this GitHub link:

  thichuong/AppliedDataScienceCapstone (github.com)

Thank you!