IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

<Luciana Moraes>
<02/Aug/2025>

Outline

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

Executive Summary

Summary of methodologies

- Data collection with API and Web Scrapping
- Data wrangling
- Exploratory Data Analysis with Data Visualization
- Exploratory Data Analysis with SQL
- Building an interactive map with Folium
- Building a Dashboard with Plotly Dash
- Predictive analysis (Classification)

- Summary of all results
  - Exploratory Data Analysis result
  - Interactive analytics in screenshots
  - Predictive Analytics result

Introduction

Context

- 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. In this lab, you will collect and make sure the data is in the correct format from an API. The following is an example of a successful and launch.

Questions that this study try to answer

- How do variables such as payload mass, launch site, number of flights, and orbits affect interact and if they affect the success of the first stage landing?
- Does the rate of successful landings increase over the years? –
- What is the best algorithm that can be used for binary classification in this case?
- What conditions should be verified to have a successful landing

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was collected using SpaceX API and web scraping
- Perform data wrangling
  - - Filtering the data - Dealing with missing values - Using One Hot Encoding to prepare the data to a binary classification
- Perform exploratory data analysis (EDA) using visualization and SQL
  - Scatter Graphs, Bar Graphs to show relationships between variables to show patterns of data. -Using SQL to find answers for the questions
- Perform interactive visual analytics using Folium and Plotly Dash
  - Circle Marker around each launch. Colored markers with success or failure and Interactive Pie charts with % of success or failure ad the total launches by site
- Perform predictive analysis using classification models
  - Different machine learning models and hyperparameters using GridSearchCV. Best model for the case

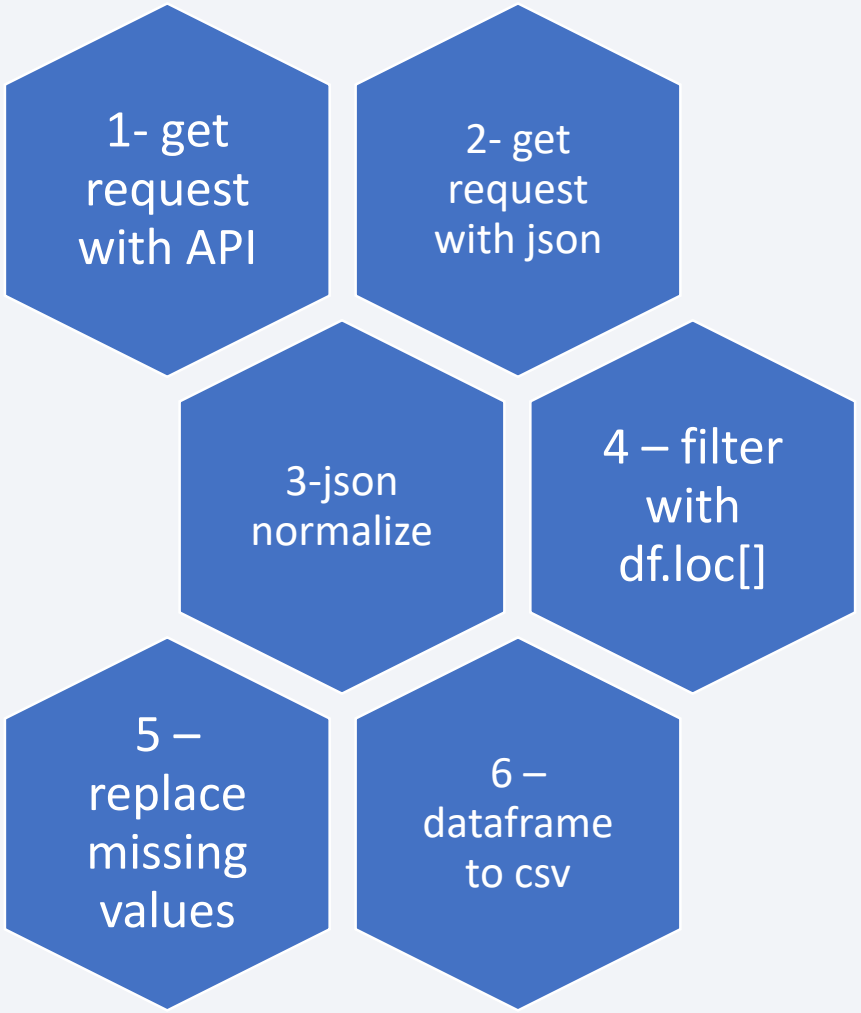
Data Collection

- Data has been collected using API and get request , get request with json using .json_normalize() to turn into a pandas dataframe
- Filter the dataframe and dealed with missing values
- Web Scraping With BeautifulSoup from Wikipedia creating in the end a pandas dataframe and a csv file.

Data Collection – SpaceX API

- data collection with SpaceX REST calls is presented using flowcharts and key phrases
- Details of code in the next slide

- GitHub URL of the completed SpaceX API calls notebook https://github.com/LAMori/Capstone-Data-Science-IBM/blob/main/jupyter-labs-spacex-data-collection-api%20.ipynb



1

Now let's start requesting rocket launch data from SpaceX API with the following URL:

```
spacex_url="https://api.spacexdata.com/v4/launches/past"
```

```
response = requests.get(spacex_url)
```

2

To make the requested JSON results more consistent, we will use the following static response object for this project:

```
static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_call_spacex_api.json'
```

We should see that the request was successfull with the 200 status response code

```
response=requests.get(static_json_url)
```

3

Now we decode the response content as a Json using `.json()` and turn it into a Pandas dataframe using `.json_normalize()`

```
# Use json_normalize meethod to convert the json result into a dataframe
#df_response = pd.json_normalize(response)

response = requests.get(static_json_url).json()
data = pd.json_normalize(response)
```

4

Finally we will remove the Falcon 1 launches keeping only the Falcon 9 launches. Filter the data dataframe using the `BoosterVersion` column to only keep the Falcon 9 launches. Save the filtered data to a new dataframe called `data_falcon9`.

```
# Hint data['BoosterVersion']!='Falcon 1'
data_falcon9 = df.loc[df['BoosterVersion']!="Falcon 1"]
```

Now that we have removed some values we should reset the FlgihtNumber column

```
data_falcon9.loc[:,'FlightNumber'] = list(range(1, data_falcon9.shape[0]+1))
data_falcon9
```

LM

5

Calculate below the mean for the `PayloadMass` using the `.mean()`. Then use the mean and the `.replace()` mean you calculated.

```
# Calculate the mean value of PayloadMass column
mean = data_falcon9['PayloadMass'].mean()
```

```
# Replace the np.nan values with its mean value
data_falcon9['PayloadMass'] = data_falcon9['PayloadMass'].fillna(mean)
```

6

We can now export it to a **CSV** for the next section,but to make the answers consistent,

```
data_falcon9.to_csv('dataset_part_1.csv', index=False)
```

Data Collection - Scraping

- The flow shows web scraping process for this project using key phrases

- The link to this notebook is: https://github.com/LAMori/Capstone-Data-Science-IBM/blob/main/jupyter-labs-webscraping.ipynb

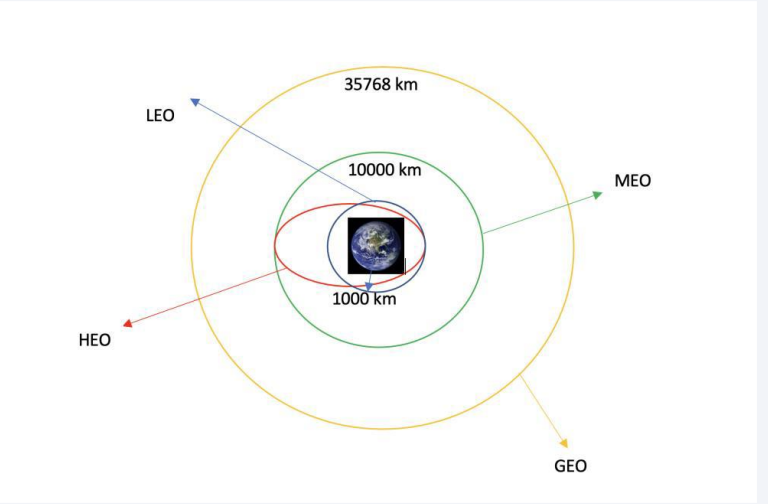
LM

Data Wrangling

- In the data set, there are several different cases where the booster did not land successfully. Sometimes a landing was attempted but failed due to an accident; for example, True Ocean means the mission outcome was successfully landed to a specific region of the ocean while False Ocean means the mission outcome was unsuccessfully landed to a specific region of the ocean. True RTLS means the mission outcome was successfully landed to a ground pad False RTLS means the mission outcome was unsuccessfully landed to a ground pad. True ASDS means the mission outcome was successfully landed on a drone ship False ASDS means the mission outcome was unsuccessfully landed on a drone ship. In this lab we will mainly convert those outcomes into Training Labels with `1` means the booster successfully landed `0` means it was unsuccessful

- Data wrangling flow for this project at next page

Diagram showing common orbit types SpaceX uses

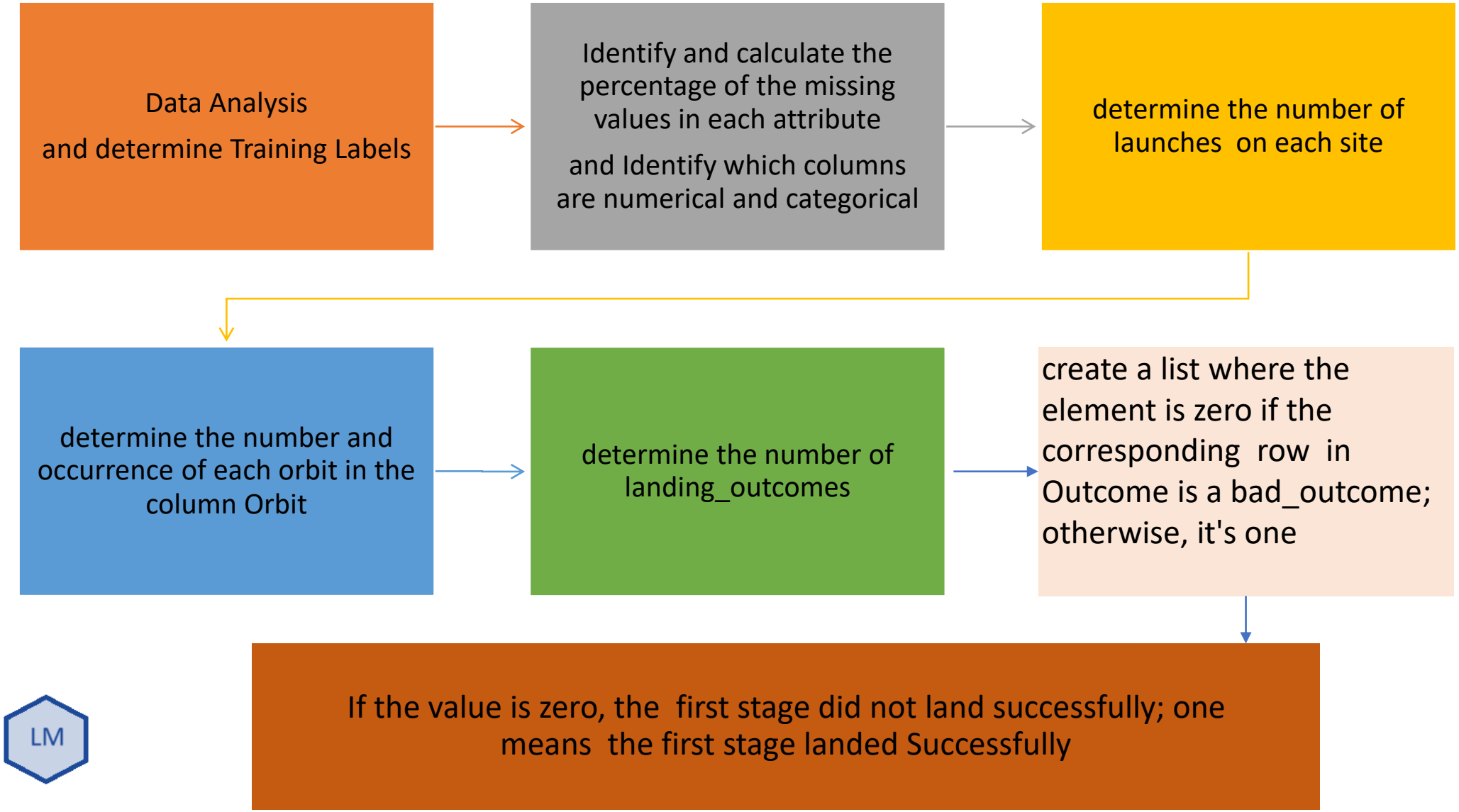


The link to the notebook for this subject is:
https://github.com/LAMori/Capstone-Data-Science-IBM/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

LM

Data Wrangling

Present your data wrangling process using key phrases and flowcharts



LM

EDA with Data Visualization

- Charts were plotted: Flight Number vs. Payload Mass, Flight Number vs. Launch Site, Payload Mass vs. Launch Site, Orbit Type vs. Success Rate, Flight Number vs. Orbit Type, Payload Mass vs Orbit Type and Success Rate Yearly Trend
- In general all those graphics are to understand the relation between the features and the success or failure of the launches
- GitHub URL of completed EDA with data visualization notebook: https://github.com/LAMori/Capstone-Data-Science-IBM/blob/main/edadataviz.ipynb

EDA with SQL

- SQL queries you performed

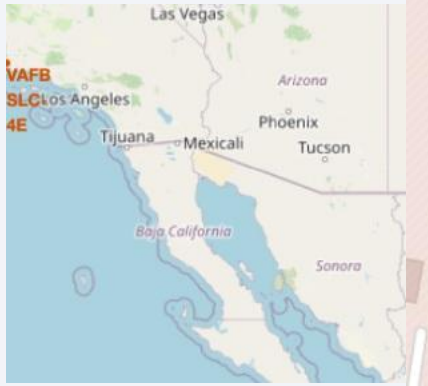
- 1 Display the names of the unique launch sites in the space mission
- 2 Display 5 records where launch sites begin with the string 'CCA'
- 3 Display the total payload mass carried by boosters launched by NASA (CRS)
- 4 Display average payload mass carried by booster version F9 v1.1
- 5 List the date when the first succesful landing outcome in ground pad was acheived.
- 6 List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
- 7 List the total number of successful and failure mission outcomes
- 8 List all the booster versions that have carried the maximum payload mass, using a subquery with a suitable aggregate function.
- 9 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
- 10 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.

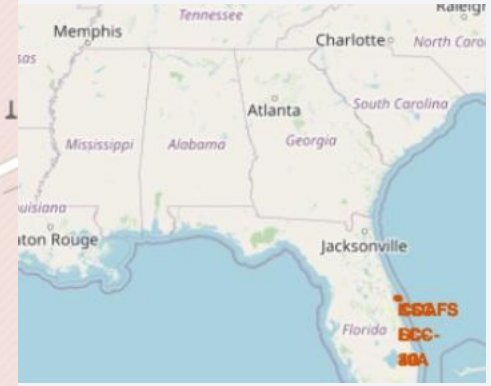
- GitHub URL of completed EDA with SQL notebook: https://github.com/LAMori/Capstone-Data-Science-IBM/blob/main/jupyter-labs-eda-sql-coursera_sqllite1.ipynb

Interactive Map with Folium

- Circle Marker around each launch site with a label of the name of the launch site on the map
- color-labeled (green and red) markers in marker clusters, you should be able to easily identify which launch sites have relatively high success rate

- GitHub URL of completed interactive map with Folium map, https://github.com/LAMori/Capstone-Data-Science-IBM/blob/main/lab_jupyter_launch_site_location.ipynb





Build a Dashboard with Plotly Dash

- It has been built an interactive dashboard with Plotly dash
- We plotted pie charts showing % of success or failure by sites
- We plotted scatter graph showing the relationship with Outcome and Payload Mass (Kg) for the different booster version.
- GitHub URL of completed Plotly Dash lab, https://github.com/LAMori/Capstone-Data-Science-IBM/blob/main/spacex_dash_app.py

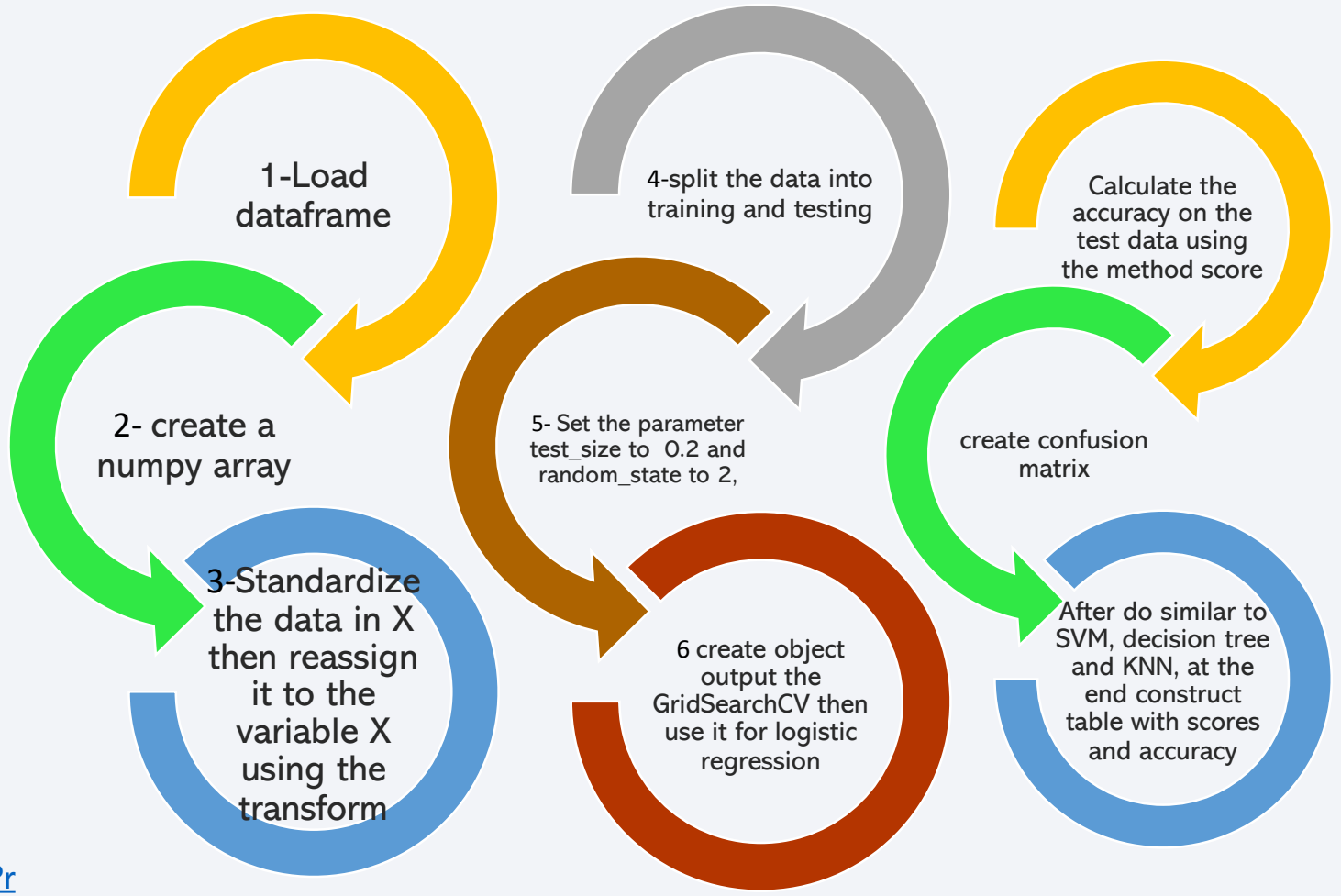
Predictive Analysis (Classification)

- The model development process using key phrases and flowchart, is show at right.
- Test set and entire dataset

	LogReg	SVM	Tree	KNN
Jaccard_Score	0.800000	0.800000	0.800000	0.800000
F1_Score	0.888889	0.888889	0.888889	0.888889
Accuracy	0.833333	0.833333	0.833333	0.833333

	LogReg	SVM	Tree	KNN
Jaccard_Score	0.833333	0.845070	0.830769	0.819444
F1_Score	0.909091	0.916031	0.907563	0.900763
Accuracy	0.866667	0.877778	0.877778	0.855556

- Note book for this subject is in this link: https://github.com/LAMori/Capstone-Data-Science-IBM/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5%20.ipynb



After tuning the best model is decision tree with score of 0.88928

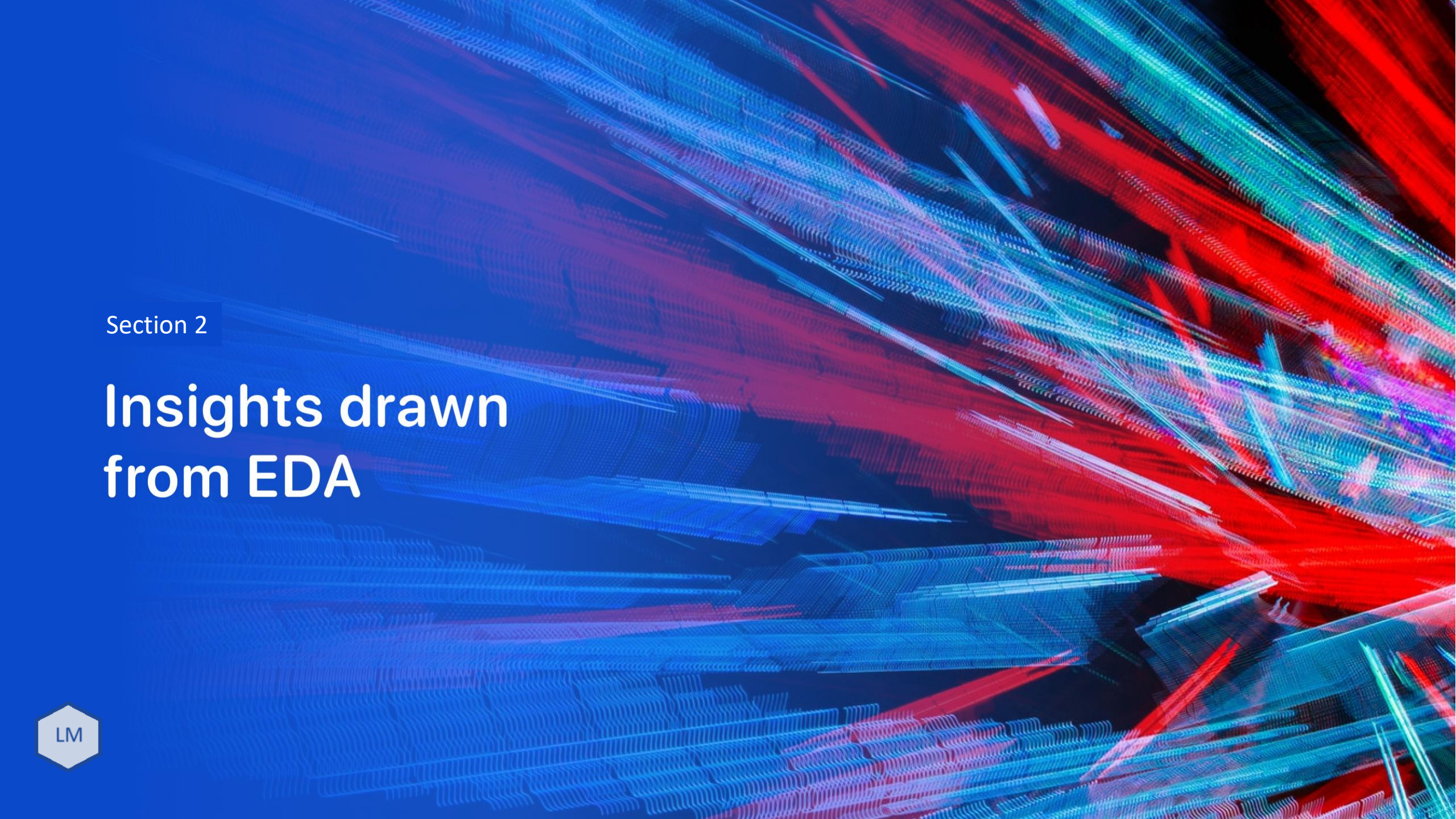
LM

Results

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

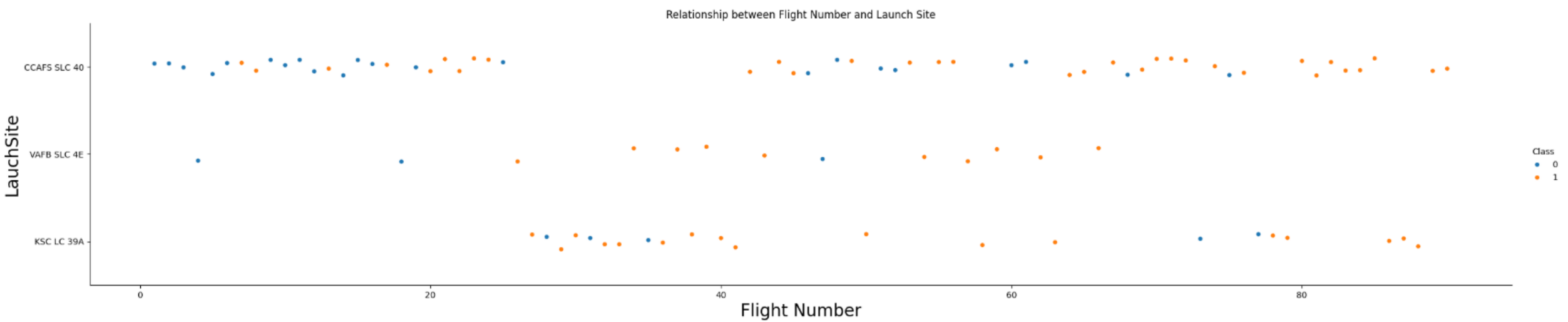
Section 2

Insights drawn from EDA



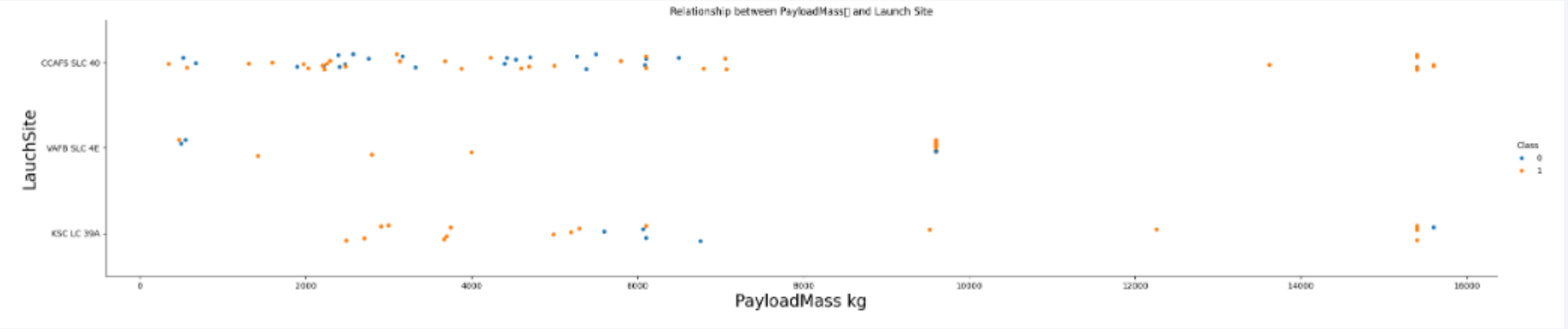
Flight Number vs. Launch Site

- Flight Number vs. Launch Site



- The CCAFS SLC 40 has launched more than the others.
- In the beggining the were more launches that failed and after there were more sucessful launches

Payload vs. Launch Site

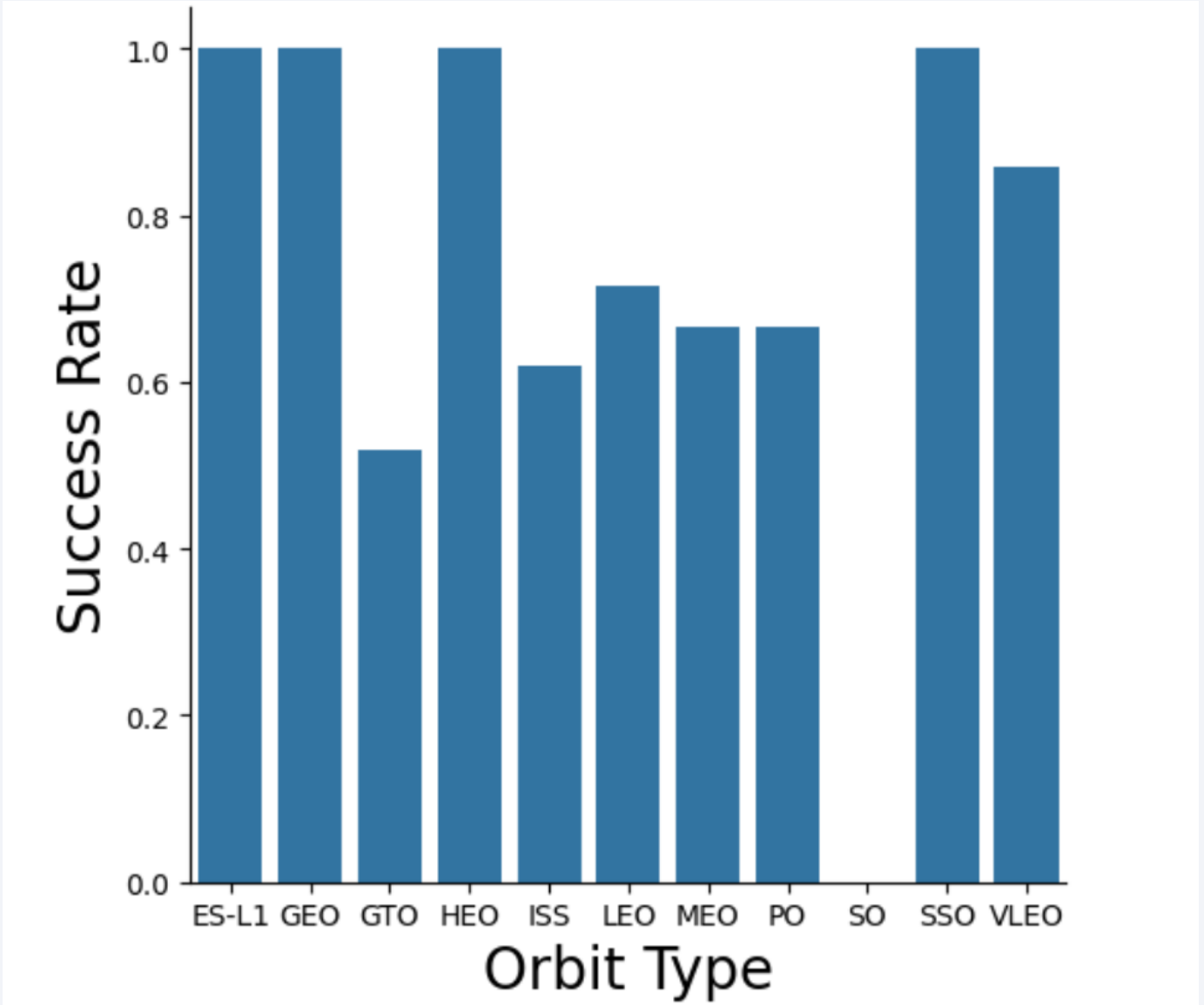


- For every launch site the higher the payload mass, the higher the success rate.
- VAFB-SLC launch site there are no rockets launched for heavypayload mass(greater than 10000)
- **Heavier payloads** (above ~10,000 kg):
- Appear **almost exclusively orange** (Class 1 → Success).
- Very few (if any) blue dots in this range — **suggesting high success rates** for heavy payloads
- **Lighter payloads** (0–6000 kg):
- Also show **mostly successful launches**, but there are **more blue dots** (Class 0 → Failure) scattered across this range.
- This indicates that **failures were more frequent** among lighter payload launches.

LM

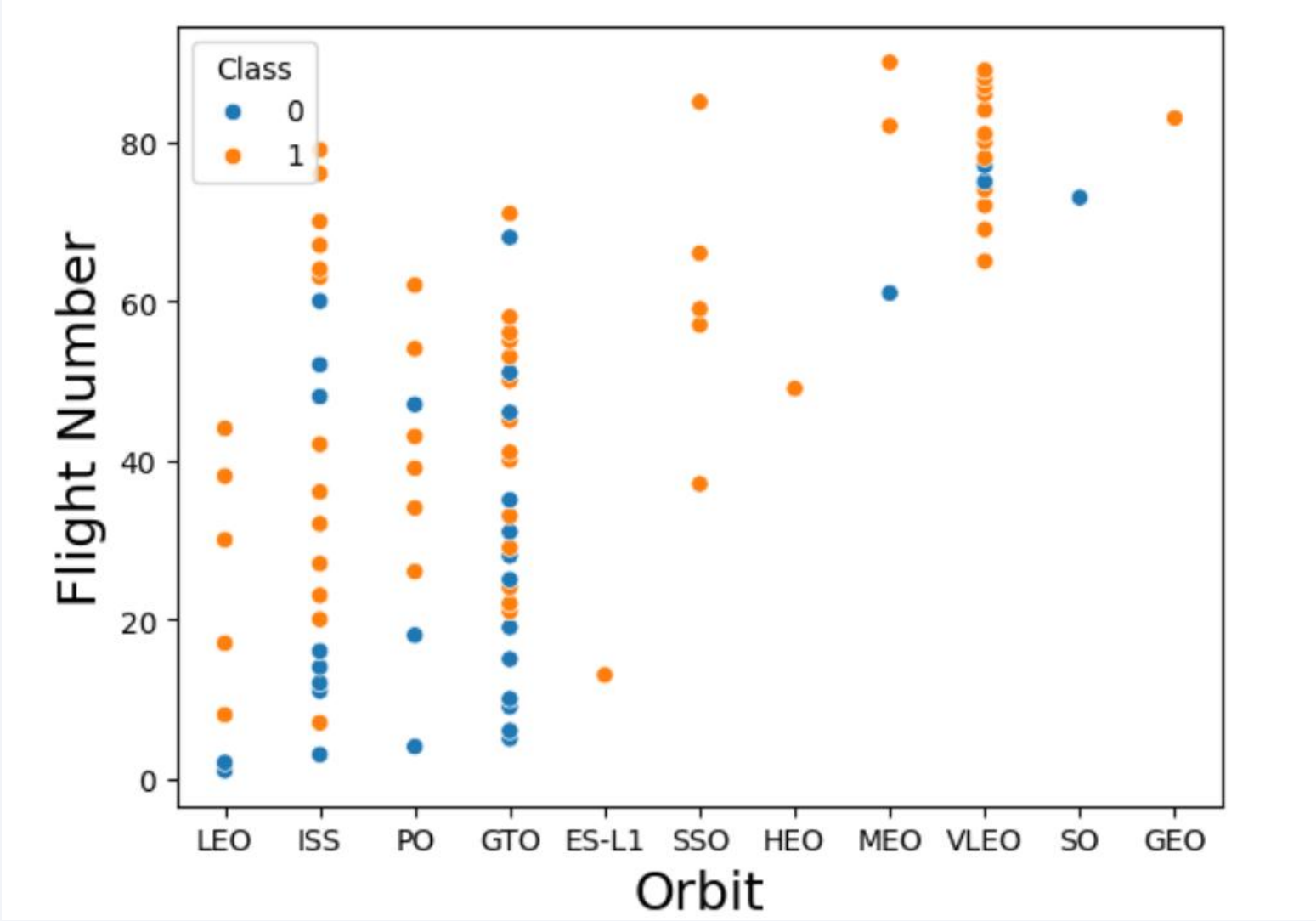
Success Rate vs. Orbit Type

- Orbits with 100% success rate: - ES-L1, GEO, HEO, SSO
- Orbits with 0% success rate: - SO
- Orbits with success rate between 50% and 85%: - GTO, ISS, LEO, MEO, PO



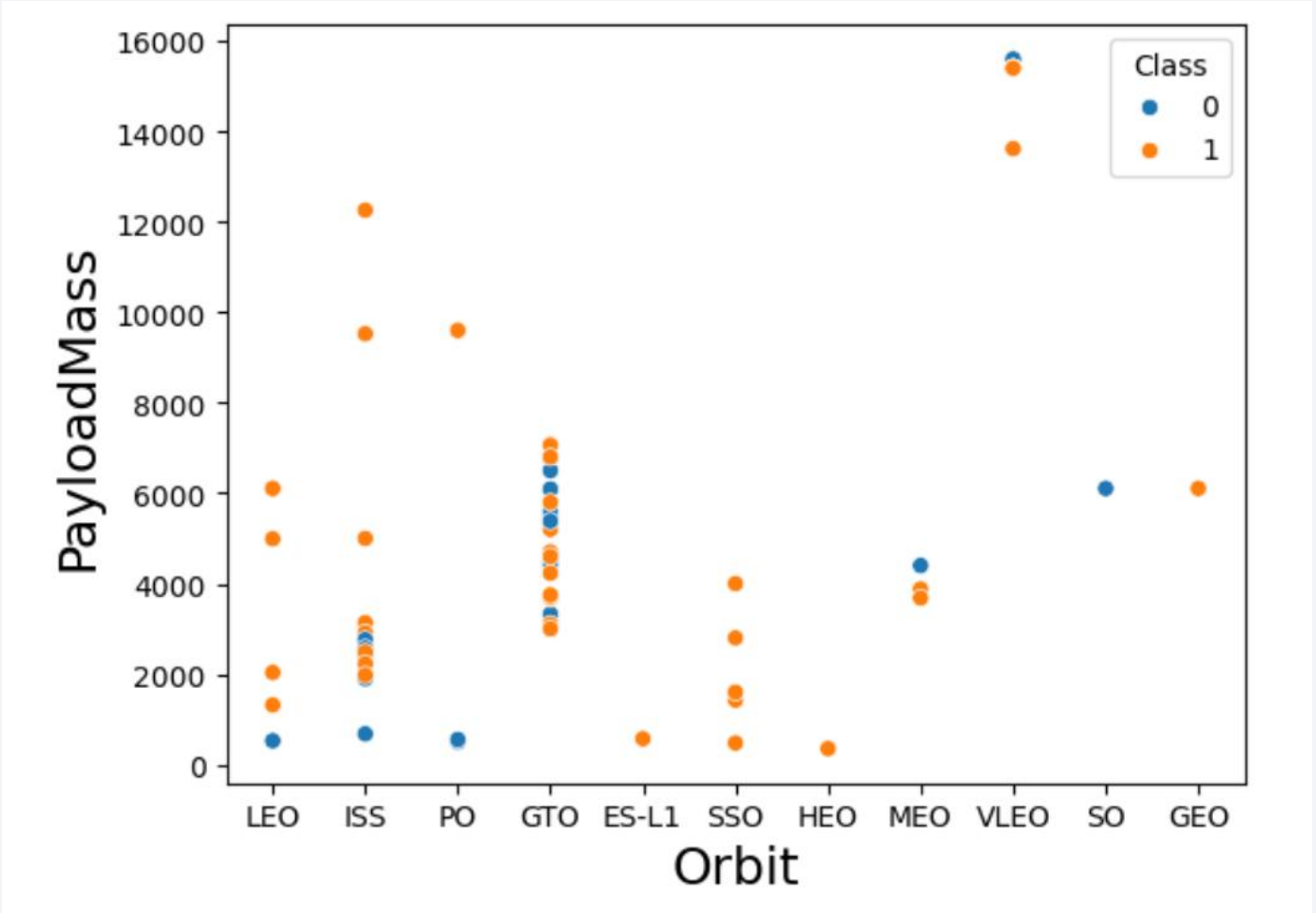
Flight Number vs. Orbit Type

- observe that in the LEO orbit, success seems to be related to the number of flights. Conversely, in the GTO orbit, there appears to be no relationship between flight number and success



Payload vs. Orbit Type

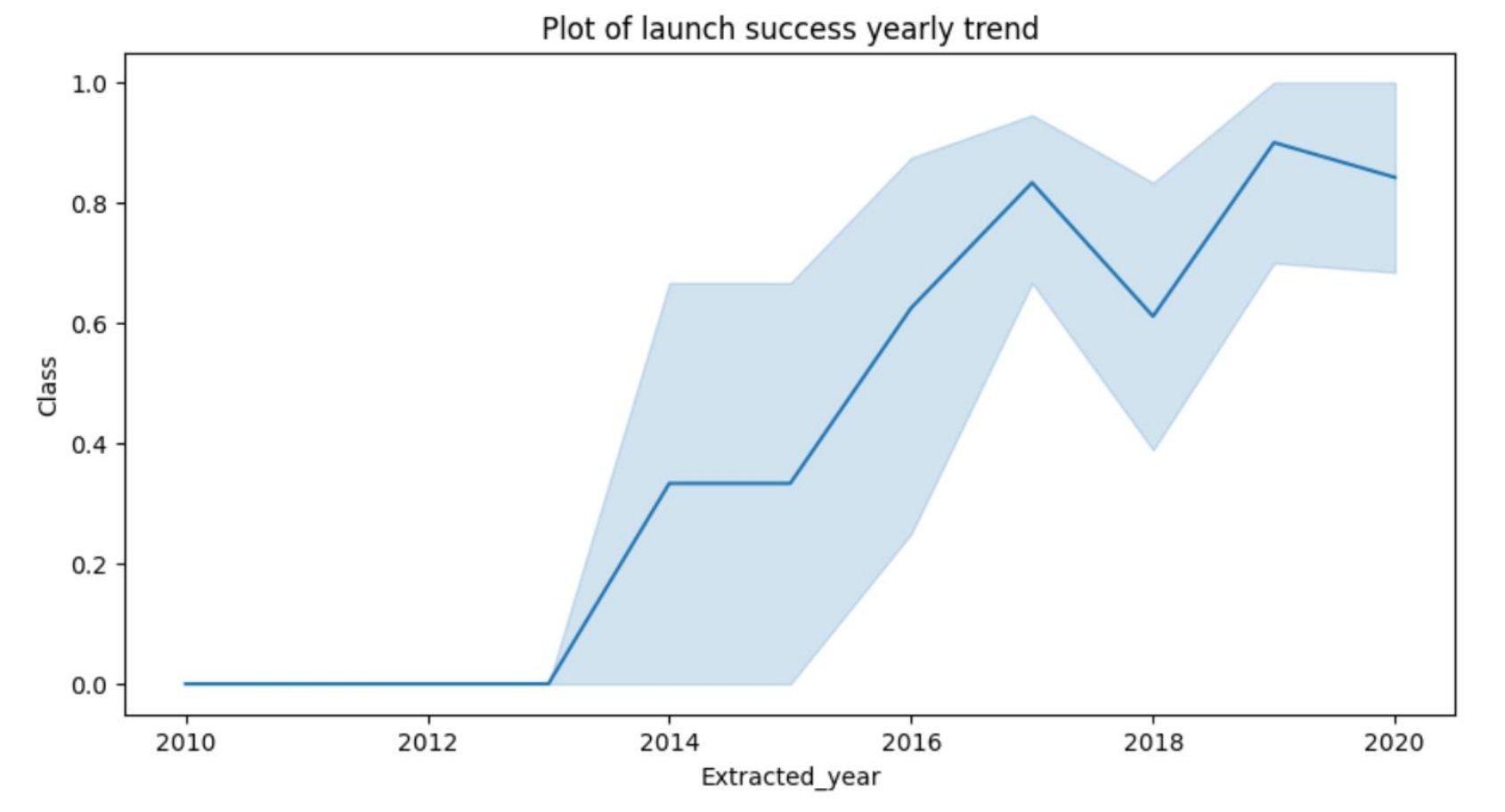
- With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS
- SSO had success with all payloads (all they launched was with payload under 6000kg)
- GTO had more success than fail but it doesn't seem to be related with payload mass



LM

Launch Success Yearly Trend

- With the exception of a slight dip in 2018, the success rate has shown a consistent upward trend from 2013 to 2020



All Launch Site Names

- Find the names of the unique launch sites

Task 1

Display the names of the unique launch sites in the space mission

```
[25]: %sql select distinct(Launch_Site) from SPACEXTABLE
 * sqlite:///my_data1.db
Done.
```

[25]:

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

- Explanation: When display of unique (in this case launche sites) is intended, the sql key word distinct is used in the SQL query

LM

Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with `CCA`

Display 5 records where launch sites begin with the string 'CCA'

```
[27]: %sql select *  from SPACEXTABLE where  Launch_Site like 'CCA%' LIMIT 5
```

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

[27]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	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	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

LM

- Explanation If a number of results is intended, it's used the query with LIMIT <number> at the end for showing only 5 results

Total Payload Mass

- Calculate the total payload carried by boosters from NASA

Task 3

Display the total payload mass carried by boosters launched by NASA (CRS)

```
28]: %sql select sum(PAYLOAD_MASS__KG_) as total_payload_Mass  from SPACEXTABLE where  Customer like 'NASA (CRS)%'
 * sqlite:///my_data1.db
Done.
```

28]:

total_payload_Mass
48213

- Explanation: the key word sum() is used in a select to sumarise a column number. In this case payload mass where customer like ‘NASA (CRS)’

LM

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1

Task 4

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

```
]: %sql select avg(PAYLOAD_MASS__KG_)  as average_payload_mass  from SPACEXTABLE where   Booster_Version like 'F9 v1.1%'
```

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

]:

average_payload_mass
2534.6666666666665

- Explanation the key word avg() on a sql query select the average of column chosen (pay_load_mass_kg_) and like limits the result for only the Booster_Version where it is F9 v1.1

LM

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad

Task 5

List the date when the first succesful landing outcome in ground pad was acheived.

Hint:Use min function

```
]: %sql select min(date) as first_date from SPACEXTABLE  where Landing_Outcome like 'Success (ground pad)'
 * sqlite:///my_data1.db
Done.
]:
```

first_date
2015-12-22

Explanation: The key word min() has been used on the select to find the minimum of the column date where the landing outcome is ‘Success (ground pad)

LM

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

Task 6

List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

```
%sql select Booster_Version  from SPACEXTABLE  where Landing_Outcome like '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

- Explanation Use a select boosters_versions where landing outcome is ‘success (drone ship)’ and payload_mass_kg is between 4000 and 6000

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes

Task 7

List the total number of successful and failure mission outcomes

```
[19]: %sql select count(*) as how_many_success  from SPACEXTABLE  where Mission_Outcome like 'Success%'
```

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

[19]:

how_many_success
100

```
[20]: %sql select count(*) as how_many_failure from SPACEXTABLE  where Landing_Outcome like 'Fail%'
```

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

[20]:

how_many_failure
10

- Explanation: used 2 selects , one to get the total of successful outcomes and one to get the number of failure outcomes

LM

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass

Task 8

List all the booster_versions that have carried the maximum payload mass, using a subquery with a suitable aggregate function.

```
%sql select Booster_Version from SPACEXTABLE  where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) as max_pay_load from SPACEXTABLE)
```

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

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

- Explanation: A sub query is used to get the maximum payload and in the main select the where clause is pay_load_mass_kg_ is equal to the sub query

2015 Launch Records

- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

Task 9 ¶

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.

Note: SQLLite does not support monthnames. So you need to use substr(Date, 6,2) as month to get the months and substr(Date,0,5)='2015' for year.

```
%sql select  (case substr(Date, 6,2) when '01' then 'January'  when'02'   then 'Febuary' when '03' then 'March' when '04' then 'April'
```

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

month	Booster_Version	Launch_Site	Landing_Outcome
January	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
April	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

- Explanation: For the month names it was used a case command comparing with the 2 char from substring of Date beginning on position 6 and using a where that get results only from the year 2015 and outcomes failure (drone ship)

LM

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

Task 10

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.

```
37]: %sql select  Landing_Outcome,  count(*)  as how_many from SPACEXTABLE where Date between '2010-06-04' and '2017-03-20'  group by Landing_Outcome order
```

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

37]:

Landing_Outcome	how_many
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

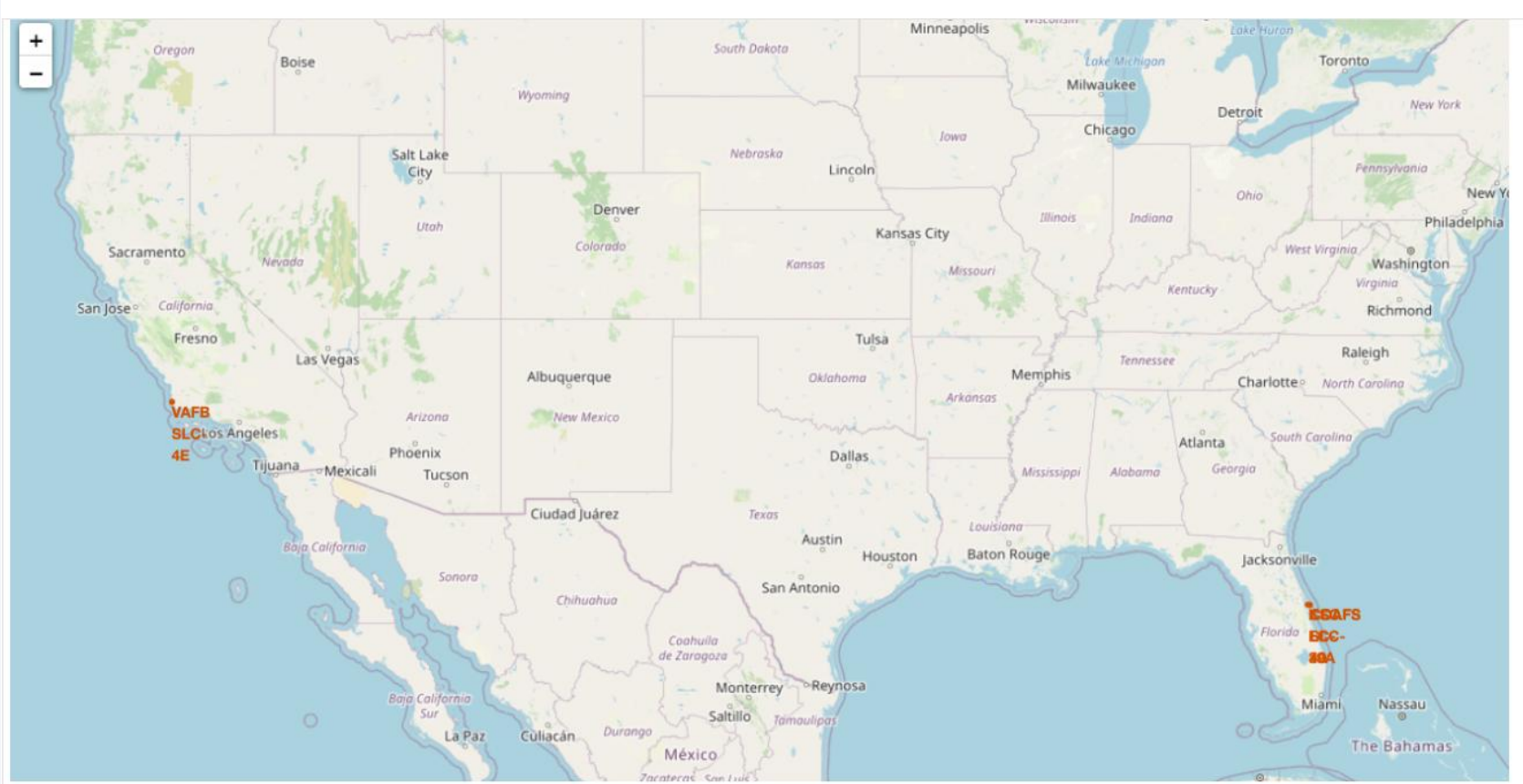
- Explanation: In this case a select count for Landing_outcome between 2010-06-04 and 2017-03-20 group by landing outcome in descending (DESC) order

LM

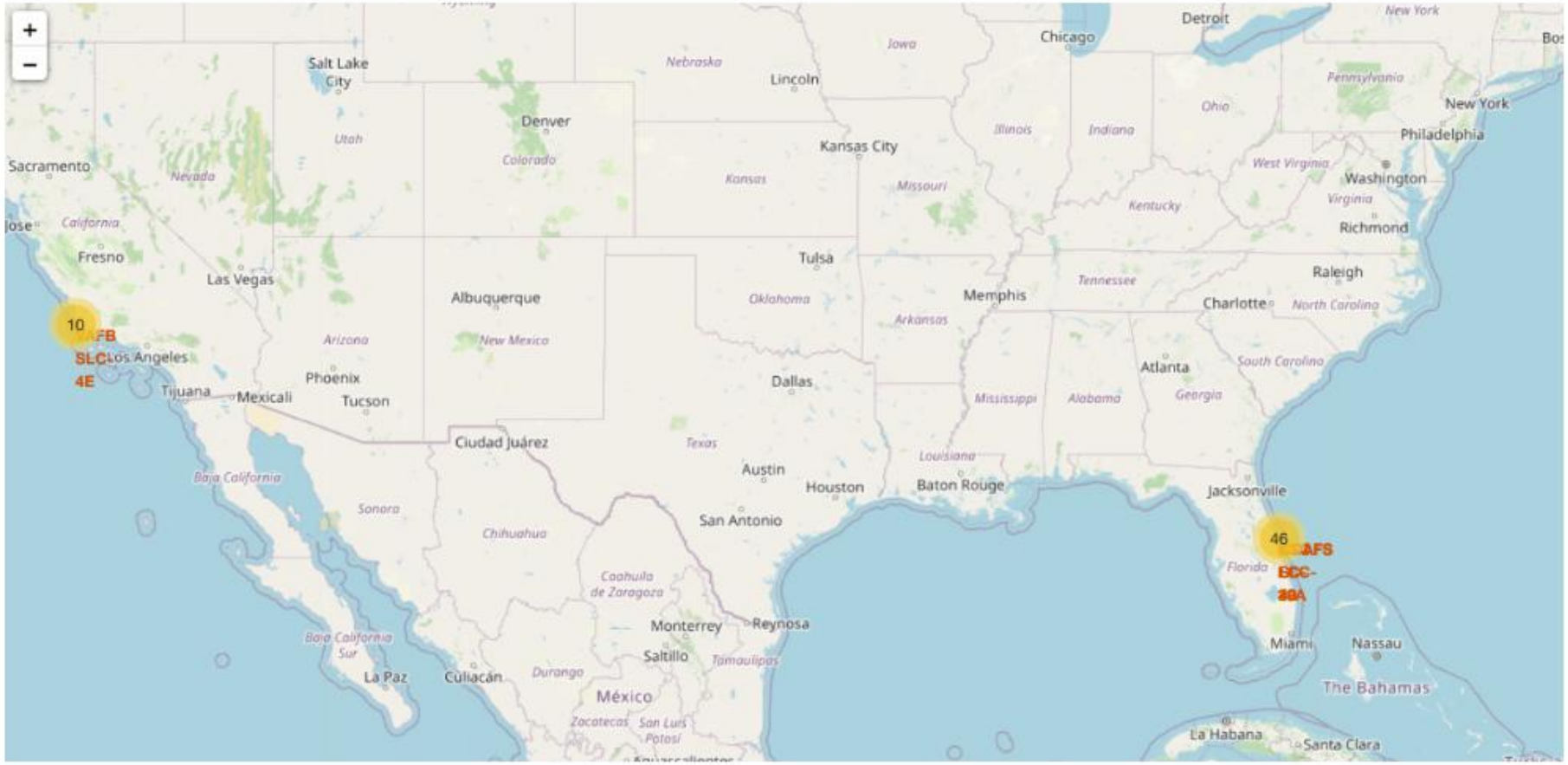

Section 3
Launch Sites
Proximities Analysis
LM

Interactive Visual Analytics with Folium

- All launch sites' location from this project markers on a global map
- * Are all launch sites in proximity to the Equator line?
- Many rockets are launched near the equator because it provides a significant advantage due to Earth's rotation. Launching from the equator, or near it, allows rockets to utilize the Earth's eastward spin, giving them an initial boost in speed. This can reduce the amount of fuel and thrust needed to achieve orbit, making launches more efficient and potentially less expensive, according to Unacademy and IFLScience.
- In this projet the launches happen all at United States of America, that is a bit far from the equator line,. But if we consider the USA, the locations of the launches are in places that from the country are one of the most near to the equator line , because they are in the very south of USA
- Are all launch sites in very close proximity to the coast?
- In our project all launches are near the cost line?, . Not only would the object being launched get a nice boost from the Earth's spin, the object would have the advantage of flying over the ocean, minimizing the risk of having any debris dropping or exploding near people

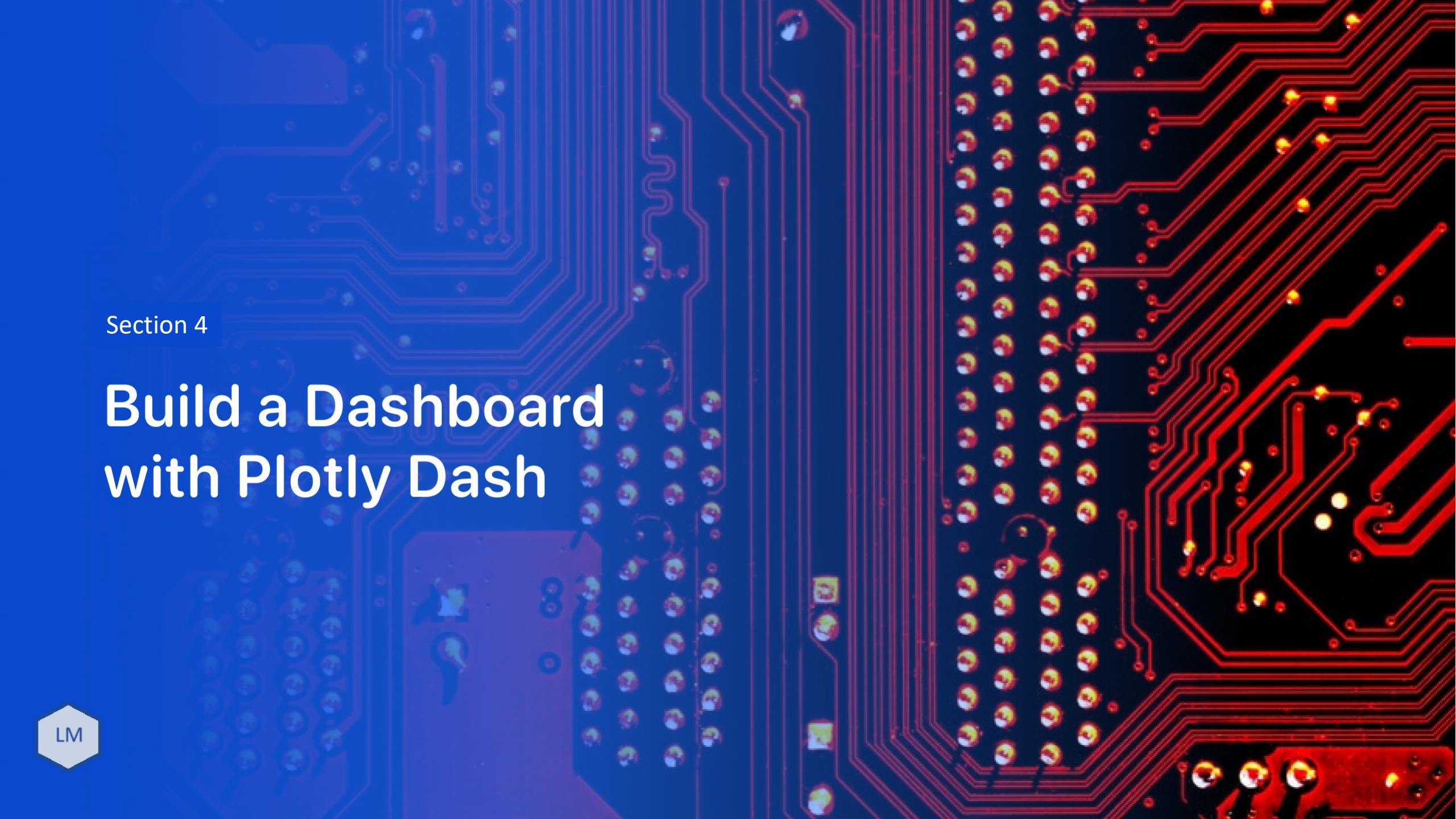


launches by location and numbers



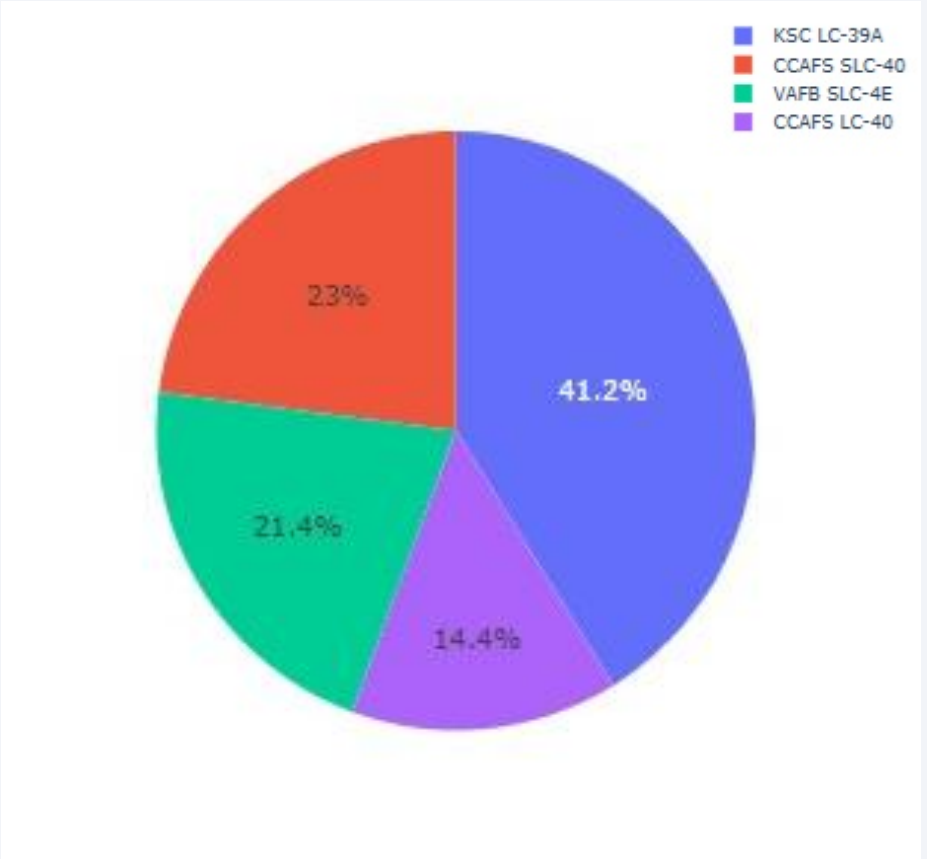
Colour-labeled launch records on the map

- From CCAFS SLC 40 in show the green launches as successful and red one failed.


Section 4
Build a Dashboard with Plotly Dash
LM

Pie chart of all launches (from this project) by site

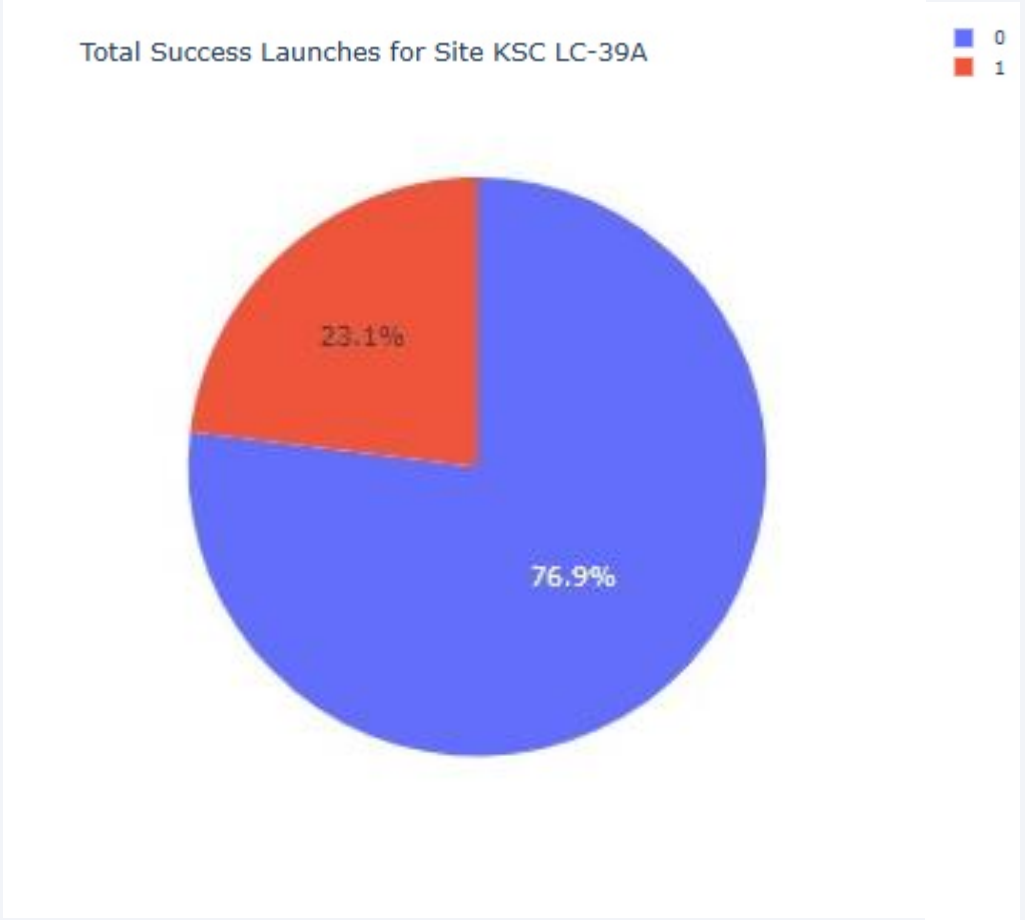
- Launches divided by site
- Most launches come from KSC LC 39A



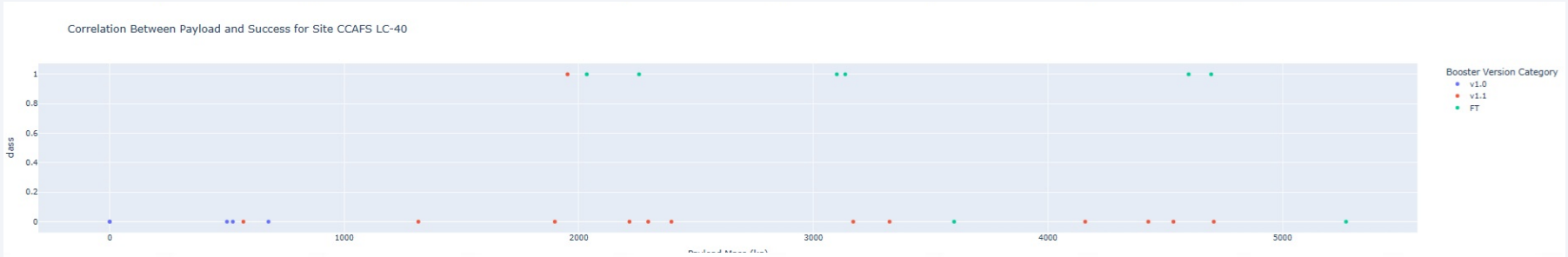
Highest Success ratio of launches

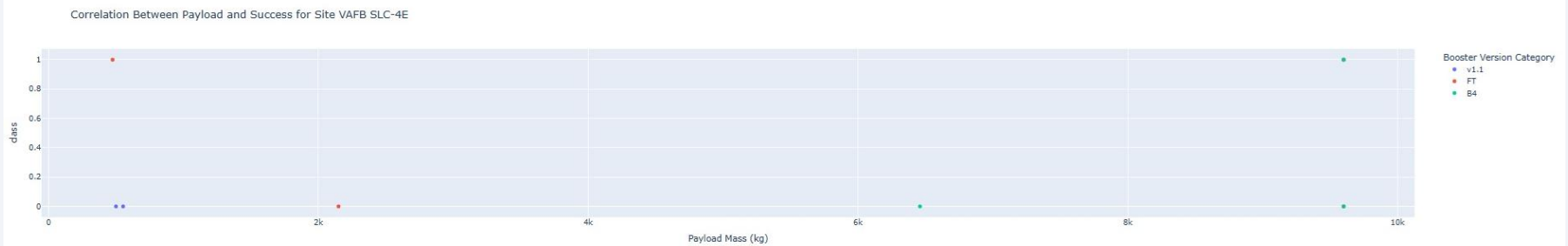
- Piechart showing the highest launch success ratio

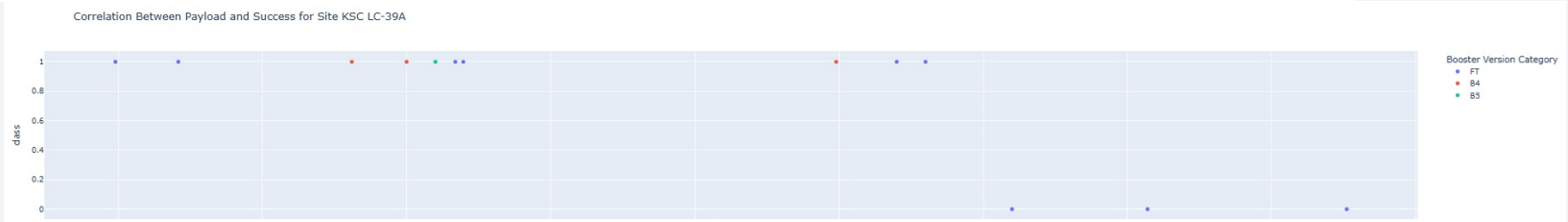
- KSC LC-39A has the highest launch success rate (76.9%)

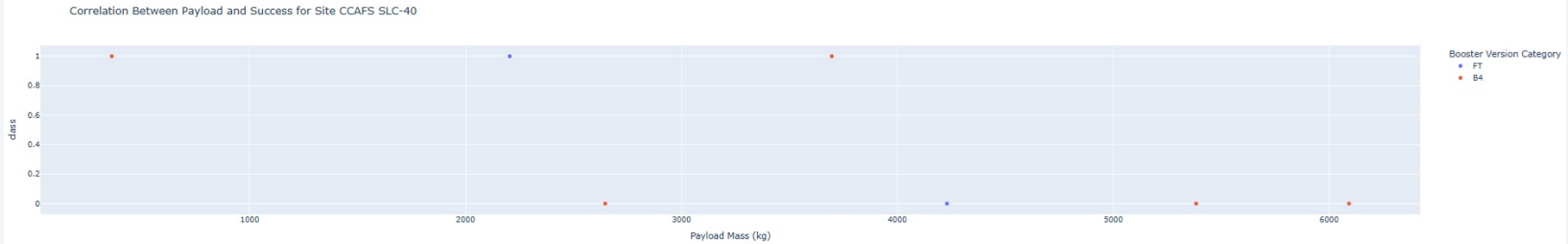


Payload Mass vs. Launch Outcome for all sites









The charts show that payloads between 2000 and 5500 kg have the highest success rate.

Section 5

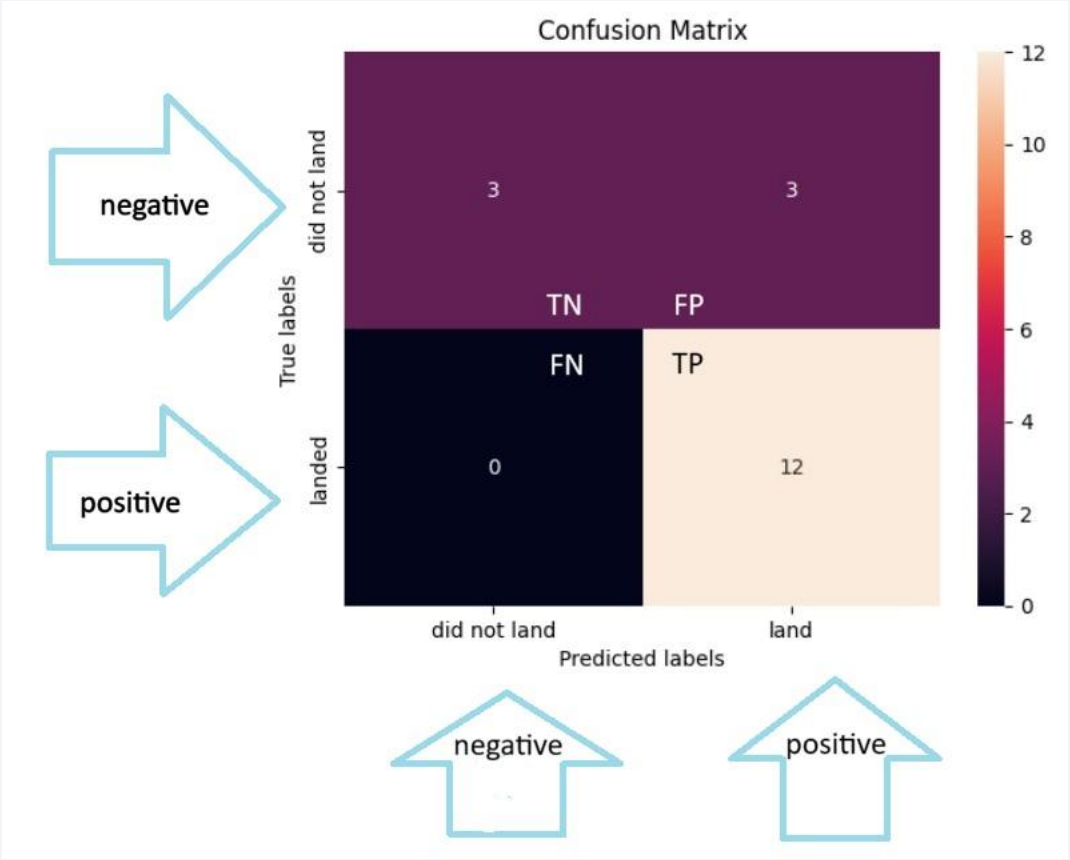
Predictive Analysis (Classification)

Classification Accuracy

Best classification model for this project is decision tree

Confusion Matrix

- Examining the confusion matrix, we see that logistic regression can distinguish between the different classes. We see that the major problem is false positives.



Conclusions

- KSC LC-39A had the most successful launches of any sites.
- The Decision tree classifier is the best machine learning algorithm for this project.
- All the sites are in very close proximity to the coast.
- Launches with a low payload mass show better results than launches with a larger payload mass.
- The success rate of launches increases over the years.
- Lauches are normaly near the cost and a not so close to trains stations and highways, so they might prevent accidents with people in the ground.

Thank you!