IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Fabio Disconzi
25/08/2022

Outline

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

Executive Summary

- Summary of methodologies
- Summary of all results

Introduction

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

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data collected via SPACEX API
  - Web scraping of wikipedia dedicated pages (Bautiful Soup library)
- Perform data wrangling
  - Nan data was processed and categorical data have been encoded into numerical data (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
  - How to build, tune, evaluate classification models

Data Collection

- Data comes from SPACE X free APIs (gets requests were used)
- Data were collected in .json format and imported in a Python envirnment
- Using Jupyter notebook a process o data import was put in place in order to generate a dataframe with all needed data in it
- Some data come from wikipedia; they were imported by some "web scraping techniques"

Data Collection – SpaceX API

- From APIs were imported by a get requests

- https://github.com/fabiodisconzi/dataScienceCourse/blob/master/Data%20COLLECTION%20API.ipynb

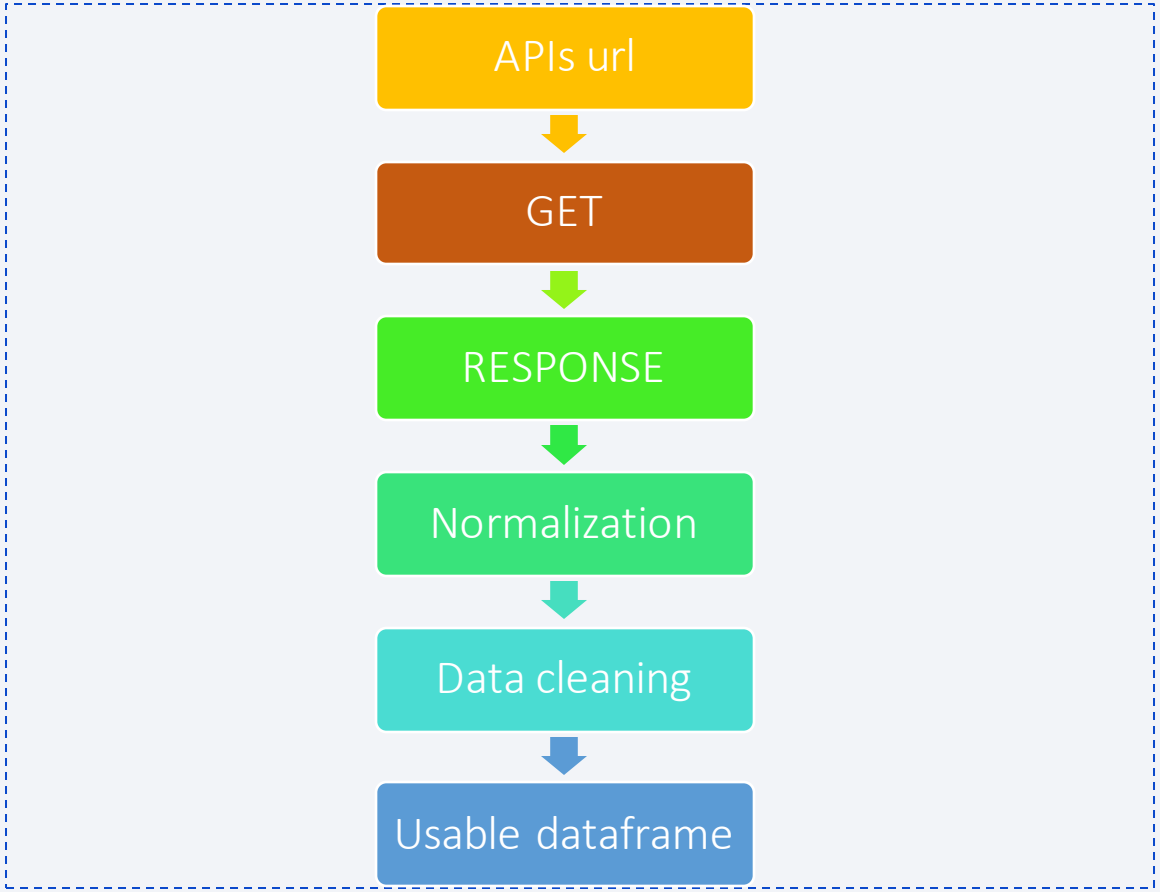
In [6]:
```
spacex_url="https://api.spacexdata.com/v4/launches/past"
```

In [7]:
```
response = requests.get(spacex_url)
```

Check the content of the response

In [8]:
```
print(response.content)
```

```
b'[{"fairings":{"reused":false,"recovery_attempt":false,"reco
```



Data Collection - Scraping

- Webscraping affected data of Falcon 9 Launch Record
- Data scraped from wikipedia page
- Beautiful SOUP library

- https://github.com/fabiodisconzi/dataScienceCourse/blob/master/jupyter-labs-webscraping.ipynb

First, let's perform an HTTP GET method to request the Falcon9 Launch HTML page, as an HTTP response.

In [7]:
```
# use requests.get() method with the provided static_url
# assign the response to a object
dataWeb = requests.get(static_url).text
#dataWeb
```

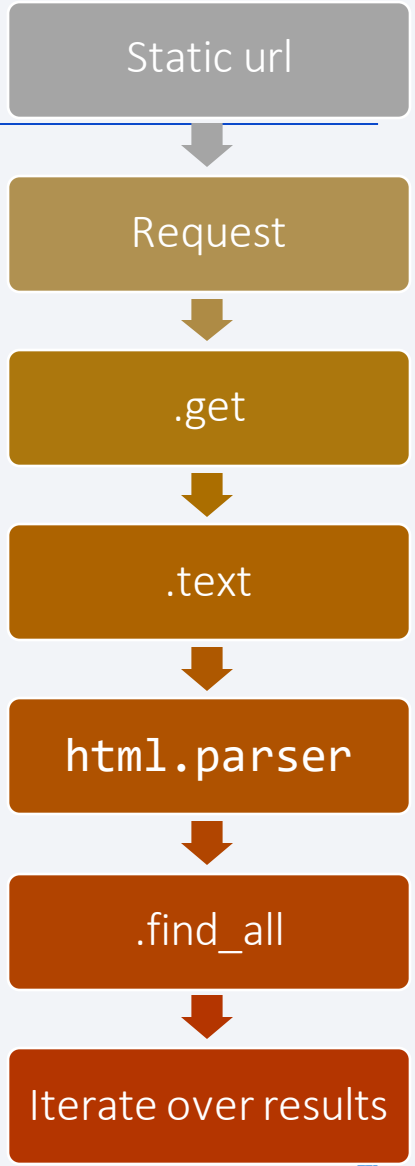
Create a `BeautifulSoup` object from the HTML `response`

In [9]:
```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
data_soup= BeautifulSoup(dataWeb, "html.parser")
#data_soup
```

Print the page title to verify if the `BeautifulSoup` object was created properly

In [10]:
```
# Use soup.title attribute
title_page=data_soup.title
print("tag object:",title_page)
```

```
tag object: <title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>
```



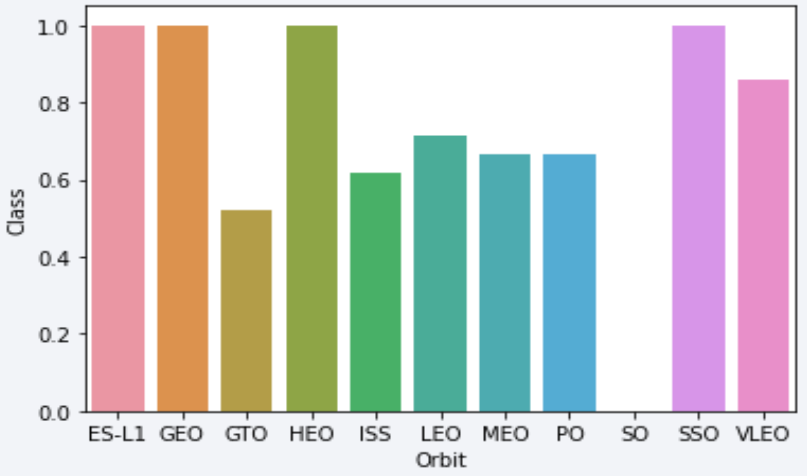
Data Wrangling

- Exploratory Data Analysis (EDA) to find some patterns in the data were performed;
- Data analysis was performed to determine Training Labels (for machine learing processes)
- *values_counts()* was applied to find best Launch site, orbits and to evaluate the success rate.

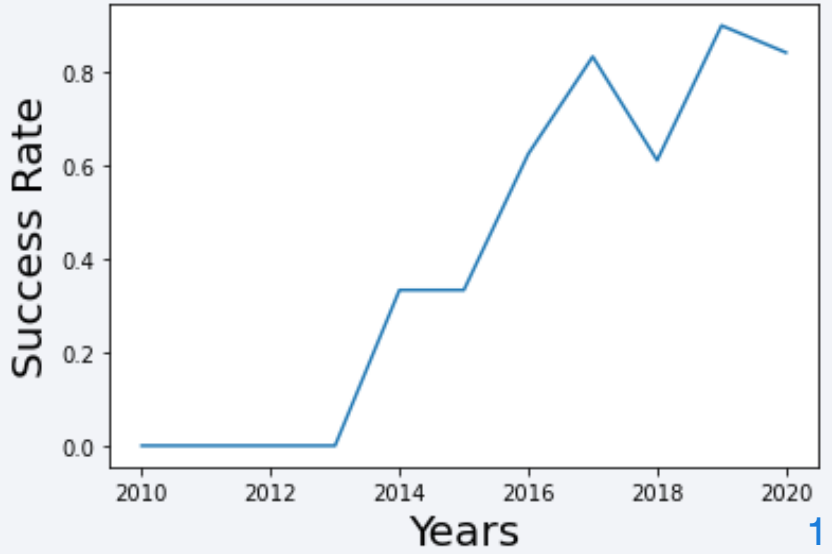
- https://github.com/fabiodisconzi/dataScienceCourse/blob/master/labs-jupyter-spacex-Data%20wrangling%20(1).ipynb

EDA with Data Visualization

- which **orbits have higher success rate**? A bar plot displays clearly the best orbits



- how is the **success rate trend**? The line plot shows an encouraging trend



- https://github.com/fabiodisconzi/dataScienceCourse/blob/master/jupyter-labs-eda-dataviz.ipynb

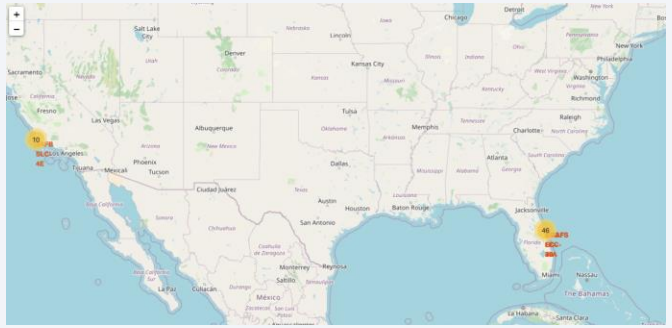
EDA with SQL

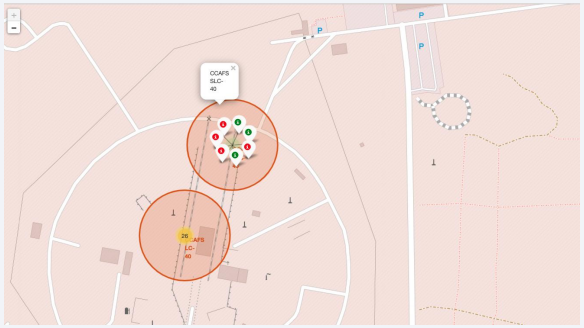
- **%sql** select DISTINCT(LAUNCH_SITE) from SPACEXTBL;
- **%sql** SELECT LAUNCH_SITE from SPACEXTBL WHERE LAUNCH_SITE LIKE "CCA%" LIMIT 0,5
- **%sql** select SUM(PAYLOAD_MASS__KG_) as payloadmass from SPACEXTBL;
- **%sql** select AVG(P
- AYLOAD_MASS__KG_) as AVGpayloadmass from SPACEXTBL WHERE Booster_Version='F9 v1.1';
- **%sql** SELECT MIN(Date) FROM SPACEXTBL WHERE UPPER(Mission_Outcome) = 'SUCCESS'
- **%sql** SELECT Booster_Version, PAYLOAD_MASS__KG_ FROM SPACEXTBL WHERE UPPER(Mission_Outcome) = 'SUCCESS' and (PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 6000)
-
- **%sql** SELECT COUNT(*) as conteggio, Mission_Outcome FROM SPACEXTBL group by Mission_Outcome
- **%sql** SELECT Booster_Version, PAYLOAD_MASS__KG_ FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL)
- **%sql** SELECT substr(Date, 4, 2) as month, "Landing _Outcome",BOOSTER_VERSION,LAUNCH_SITE FROM SPACEXTBL where substr(Date,7,4)='2015';
- **%sql** SELECT Launch_Site, count("Landing _Outcome")as OUTCOME FROM SPACEXTBL WHERE Date BETWEEN '04-06-2010' AND '20-03-2017' group by Launch_Site ORDER BY OUTCOME DESC;

https://github.com/fabiodisconzi/dataScienceCourse/blob/master/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- markers for all launch records
- Marker clusters since many markers have the same coordinate
- Color associated to class outcomes (0: unsuccess, 1 success)
- Interesting infrastructure distances were calculated (highway, railways, etc.)

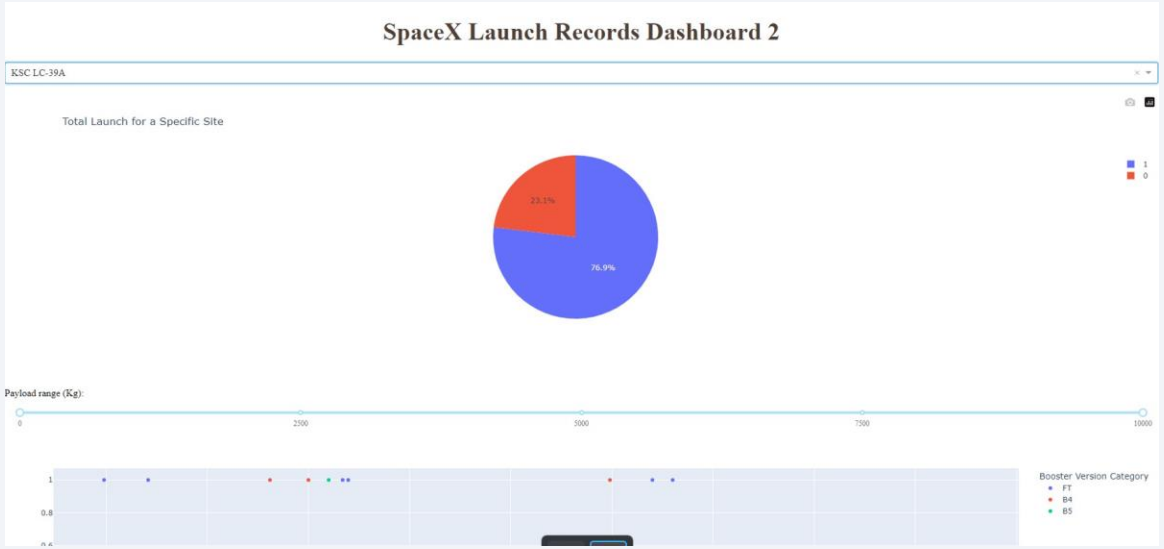




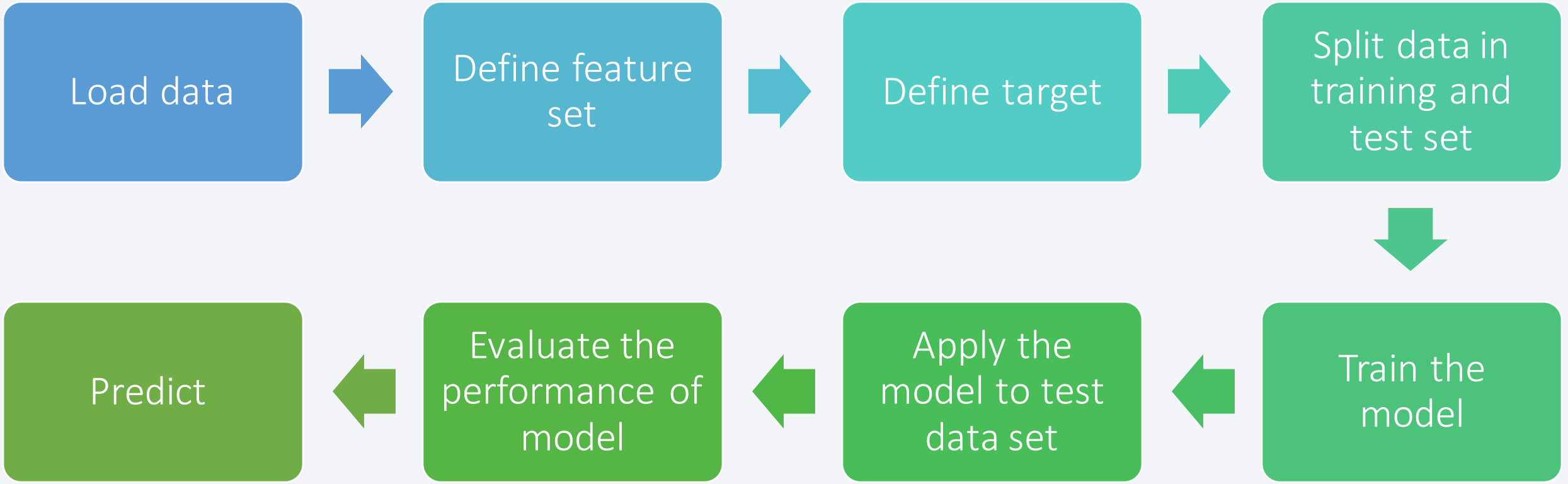
https://github.com/fabiodisconzi/dataScienceCourse/blob/master/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

- Hboard with plotly and Dash libraries
- Pie charts let to stakeholder to immediately understand the performances and outcomes of each launch site and the effect of payload.



Predictive Analysis (Classification)



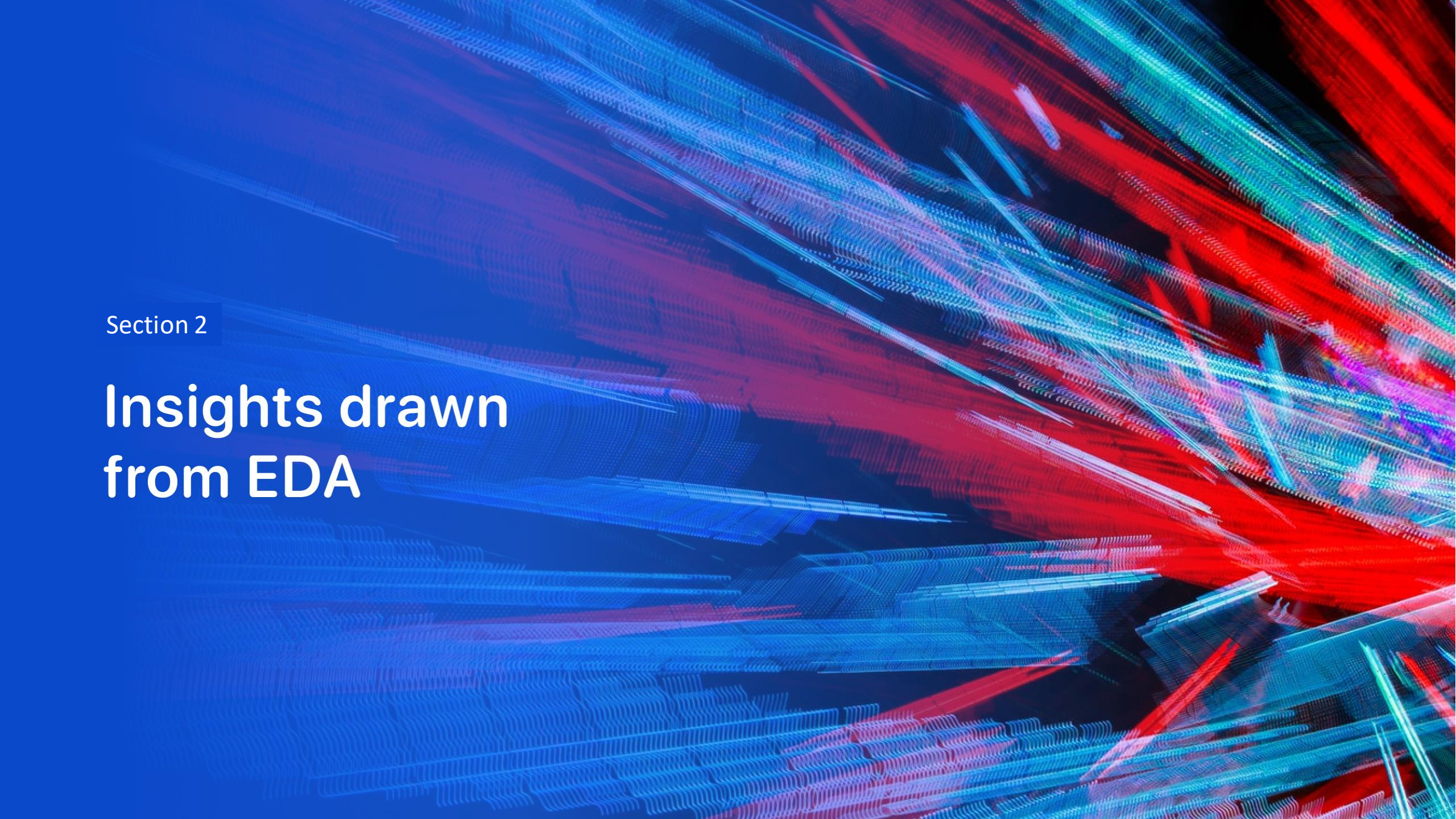
https://github.com/fabiodisconzi/dataScienceCourse/blob/master/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results

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

Section 2

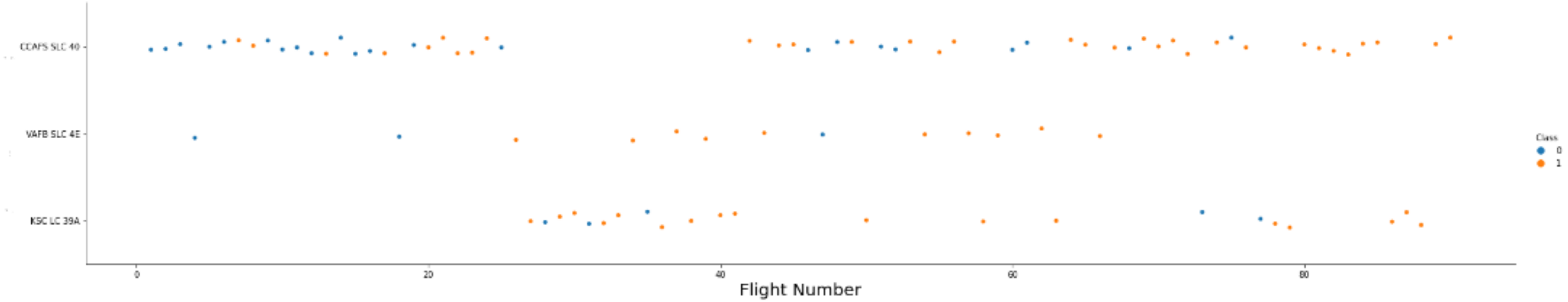
Insights drawn from EDA



Flight Number vs. Launch Site

In [8]:

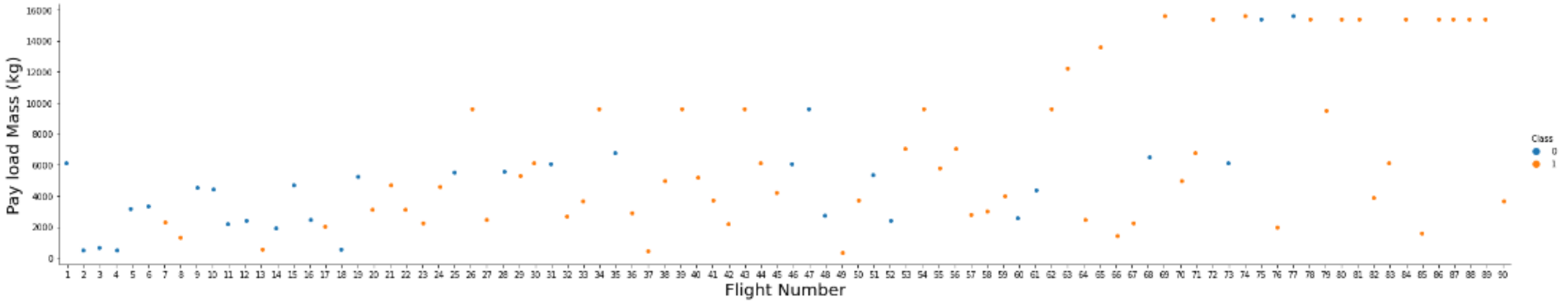
```
# Plot a scatter point chart with x axis to be Flight Number and y axis to be the launch site, and hue to be the class value
sns.catplot(y="LaunchSite", x="FlightNumber", hue="Class", data=df, aspect = 5)
plt.xlabel("Flight Number",fontsize=20)
plt.ylabel("Pay load Mass (kg)",fontsize=20)
plt.show()
```



- The greater the launch site, higher is the success rate (more orange points)

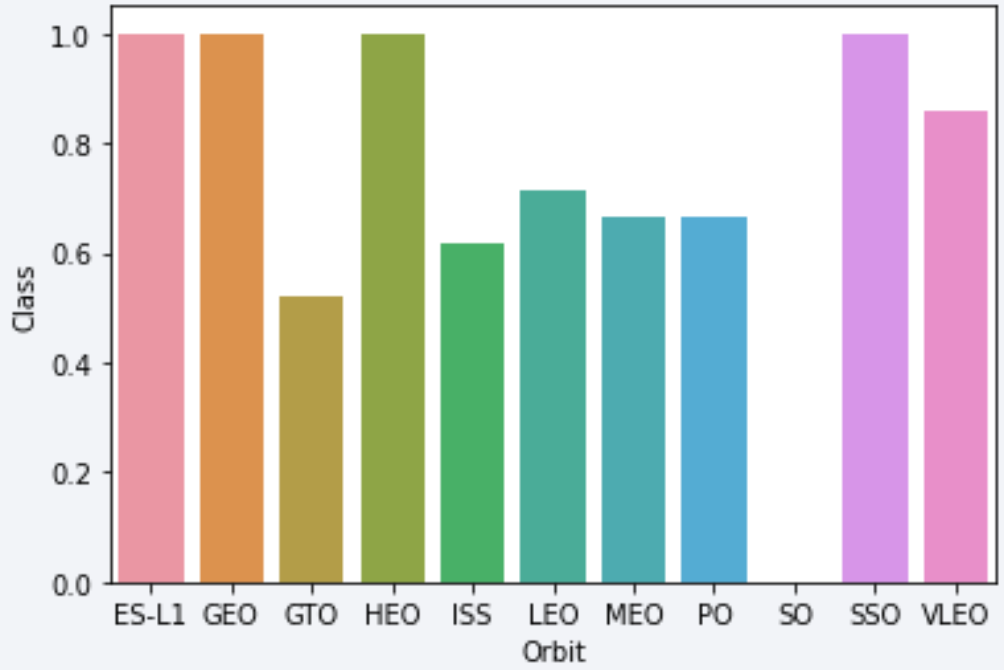
Payload vs. Launch Site

```
sns.catplot(y="PayloadMass", x="FlightNumber", hue="Class", data=df, aspect = 5)
plt.xlabel("Flight Number",fontsize=20)
plt.ylabel("Pay load Mass (kg)",fontsize=20)
plt.show()
```



Last flight with higher paylod mass are also more successful.

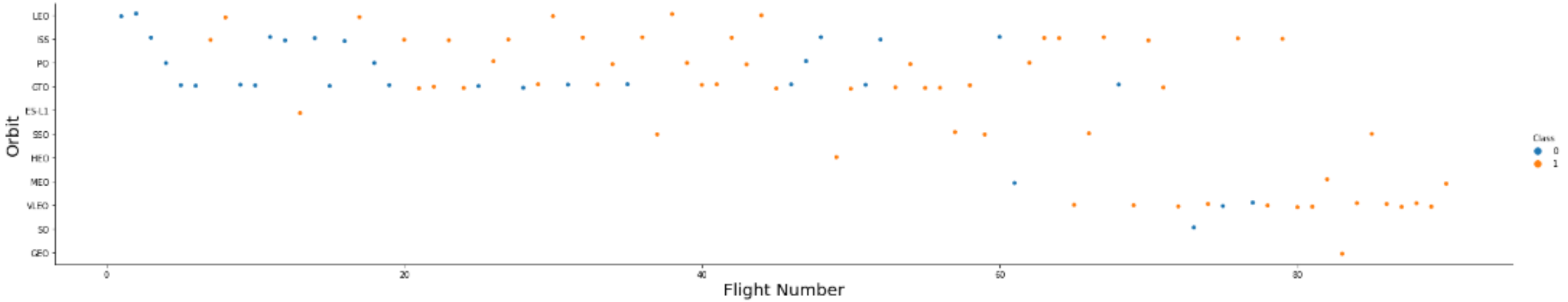
Success Rate vs. Orbit Type



There are some orbits with a 100% successful rate.

Flight Number vs. Orbit Type

```
# Plot a scatter point chart with x axis to be FlightNumber and y axis to be the Orbit, and hue to be the class value
sns.catplot(y="Orbit", x="FlightNumber", hue="Class", data=df, aspect = 5)
plt.xlabel("Flight Number",fontsize=20)
plt.ylabel("Orbit",fontsize=20)
plt.show()
```

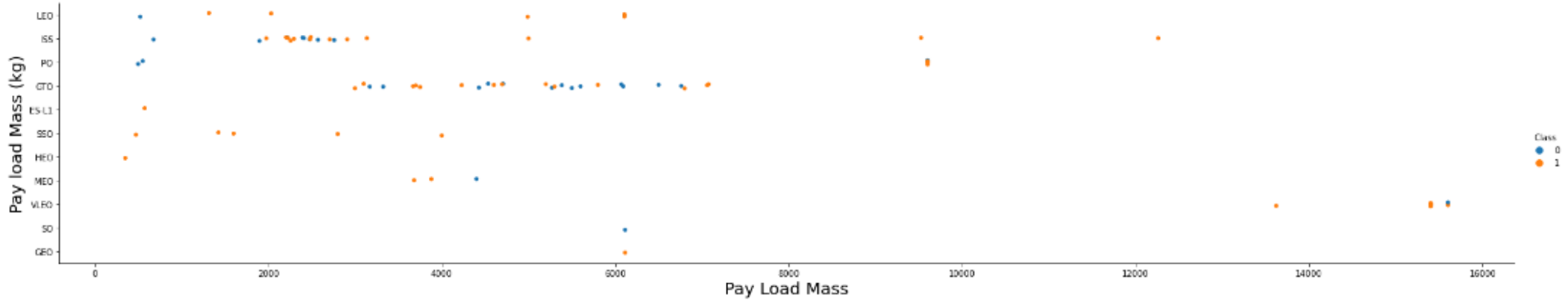


- in the LEO orbit the Success appears related to the number of flight

Payload vs. Orbit Type

- With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS

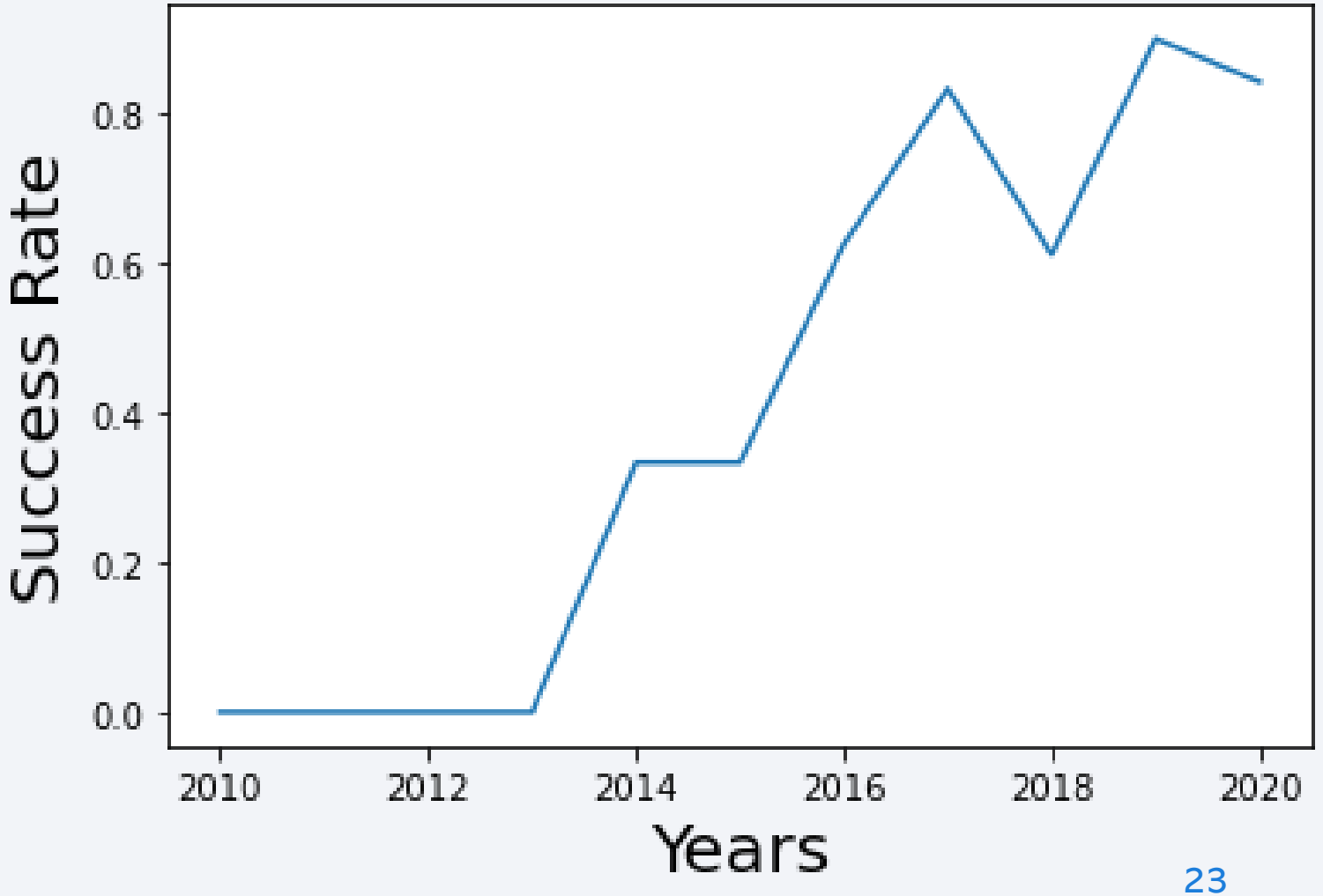
```
# Plot a scatter point chart with x axis to be Payload and y axis to be the Orbit, and hue to be the class value
sns.catplot(y="Orbit", x="PayloadMass", hue="Class", data=df, aspect = 5)
plt.xlabel("Pay Load Mass",fontsize=20)
plt.ylabel("Pay load Mass (kg)",fontsize=20)
plt.show()
```



With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS.

Launch Success Yearly Trend

- the sucess rate since 2013 kept increasing till 2020



All Launch Site Names

- Launch_Site

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

```
%sql select DISTINCT(LAUNCH_SITE) from SPACEXTBL;
```

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

Launch Site Names Begin with 'CCA'

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

In [12]:
```
%sql SELECT LAUNCH_SITE from SPACEXTBL WHERE LAUNCH_SITE LIKE "CCA%" LIMIT 0,5
```

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

Out[12]:

Launch_Site
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40

Total Payload Mass

Task 3

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

In [13]:
```
%sql select SUM(PAYLOAD_MASS__KG_) as payloadmass from SPACEXTBL;
```

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

Out[13]:

payloadmass
619967

Average Payload Mass by F9 v1.1

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

In [14]:
```
%sql select AVG(PAYLOAD_MASS__KG_) as AVGpayloadmass from SPACEXTBL WHERE Booster_Version='F9 v1.1';
```

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

Out[14]:

AVGpayloadmass
2928.4

First Successful Ground Landing Date

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

Hint:Use min function

In [32]:
```
%sql SELECT MIN(Date) FROM SPACEXTBL WHERE UPPER(Mission_Outcome) = 'SUCCESS'
```

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

Out[32]:

MIN(Date)
01-03-2013

Successful Drone Ship Landing with Payload between 4000 and 6000

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

In [31]:
```
%sql SELECT Booster_Version, PAYLOAD_MASS__KG_ FROM SPACEXTBL WHERE UPPER(Mission_Outcome) = 'SUCCESS' and (PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS_
```

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

Out[31]:

Booster_Version	PAYLOAD_MASS__KG_
F9 v1.1	4535
F9 v1.1 B1011	4428
F9 v1.1 B1014	4159
F9 v1.1 B1016	4707
F9 FT B1020	5271
F9 FT B1022	4696
F9 FT B1026	4600
F9 FT B1030	5600
F9 FT B1021.2	5300
F9 FT B1032.1	5300
F9 B4 B1040.1	4990
F9 FT B1031.2	5200
F9 FT B1032.2	4230
F9 B4 B1040.2	5384
F9 B5 B1046.2	5800
F9 B5 B1047.2	5300
F9 B5 B1048.3	4850
F9 B5 B1051.2	4200
F9 B5B1060.1	4311
F9 B5 B1058.2	5500
F9 B5B1062.1	4311

Total Number of Successful and Failure Mission Outcomes

List the total number of successful and failure mission outcomes

In [33]:
```
%sql SELECT COUNT(*) as conteggio, Mission_Outcome FROM SPACEXTBL group by Mission_Outcome
```

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

Out[33]:

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

Boosters Carried Maximum Payload

List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

In [35]:
```
%sql SELECT Booster_Version, PAYLOAD_MASS__KG_ FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL)
```

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

Out[35]:

Booster_Version	PAYLOAD_MASS__KG_
F9 B5 B1048.4	15600
F9 B5 B1049.4	15600
F9 B5 B1051.3	15600
F9 B5 B1056.4	15600
F9 B5 B1048.5	15600
F9 B5 B1051.4	15600
F9 B5 B1049.5	15600
F9 B5 B1060.2	15600
F9 B5 B1058.3	15600
F9 B5 B1051.6	15600
F9 B5 B1060.3	15600
F9 B5 B1049.7	15600

2015 Launch Records

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, 4, 2) as month to get the months and substr(Date,7,4)='2015' for year.

In [39]:
```
%sql SELECT substr(Date, 4, 2) as month, "Landing _Outcome",BOOSTER_VERSION,LAUNCH_SITE FROM SPACEXTBL where substr(Date,7,4)='2015';
```

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

Out[39]:

month	Landing _Outcome	Booster_Version	Launch_Site
01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
02	Controlled (ocean)	F9 v1.1 B1013	CCAFS LC-40
03	No attempt	F9 v1.1 B1014	CCAFS LC-40
04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40
04	No attempt	F9 v1.1 B1016	CCAFS LC-40
06	Precluded (drone ship)	F9 v1.1 B1018	CCAFS LC-40
12	Success (ground pad)	F9 FT B1019	CCAFS LC-40

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.

In [44]:
```
%sql SELECT Launch_Site, count("Landing _Outcome")as OUTCOME FROM SPACEXTBL WHERE Date BETWEEN '04-06-2010' AND '20-03-2017' group by Launch_Site ORDE
```

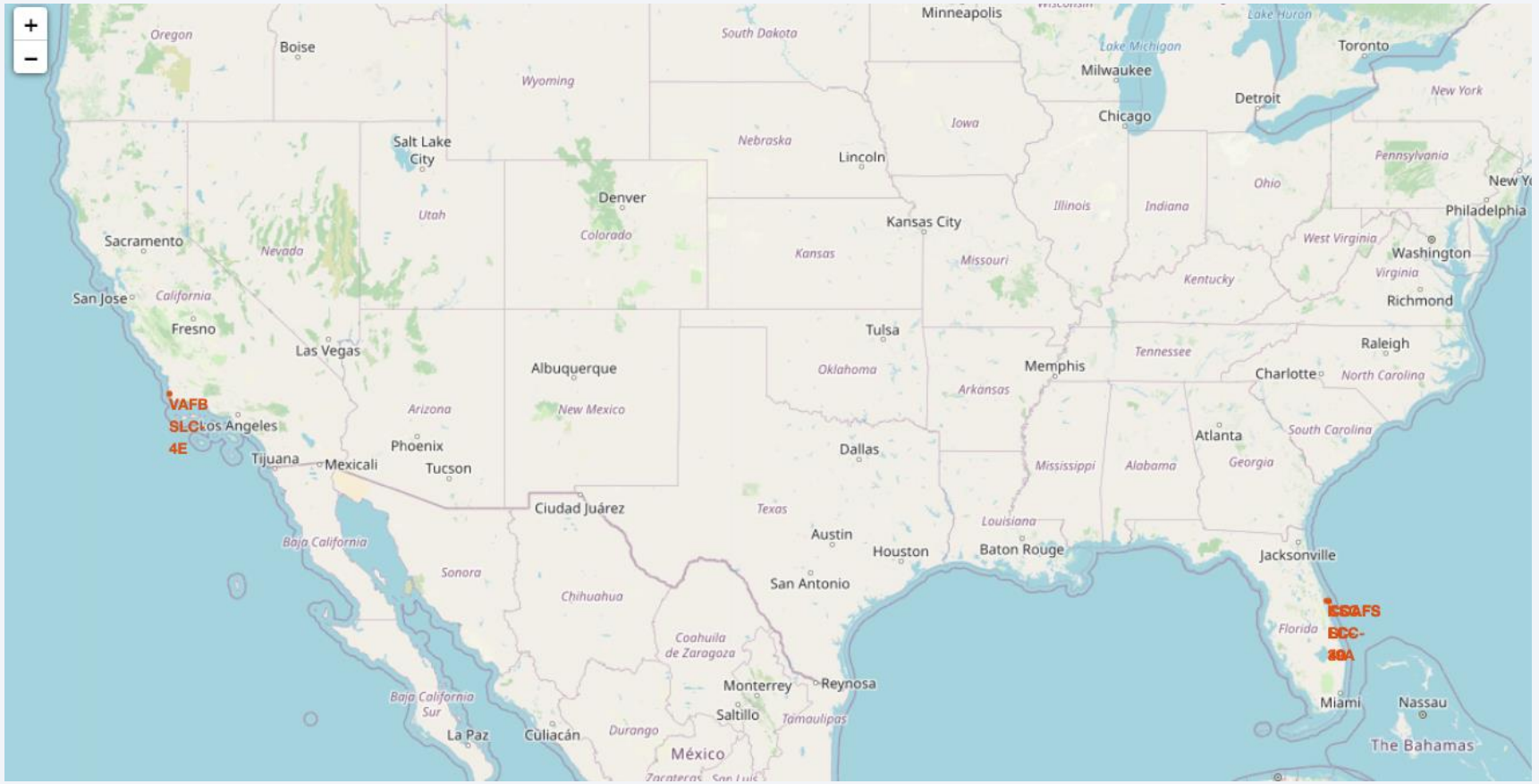
```
 * sqlite:///my_data1.db
Done.
```

Out[44]:

Launch_Site	OUTCOME
CCAFS SLC-40	19
KSC LC-39A	16
CCAFS LC-40	16
VAFB SLC-4E	6

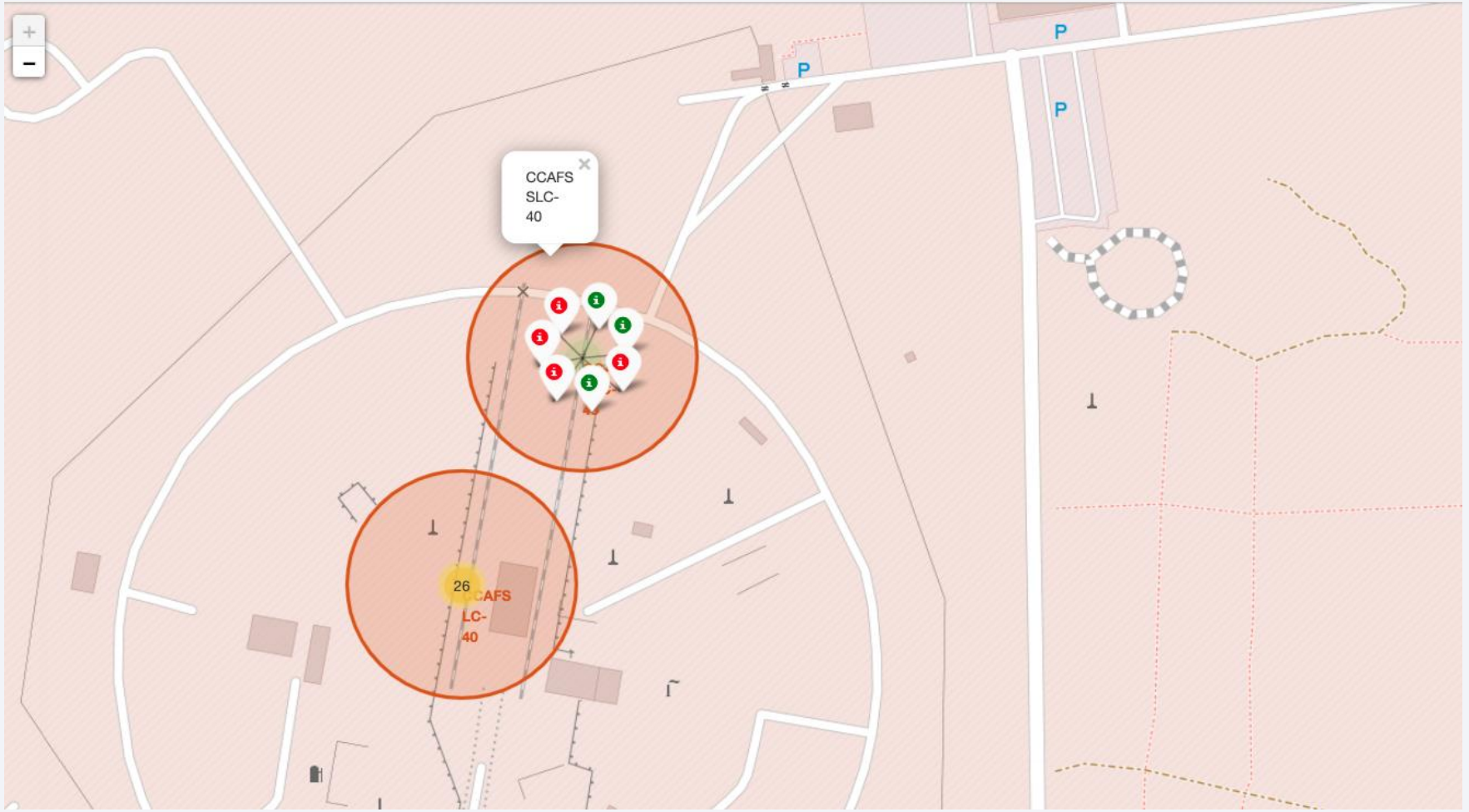

Section 3
Launch Sites
Proximities Analysis

Launching sites



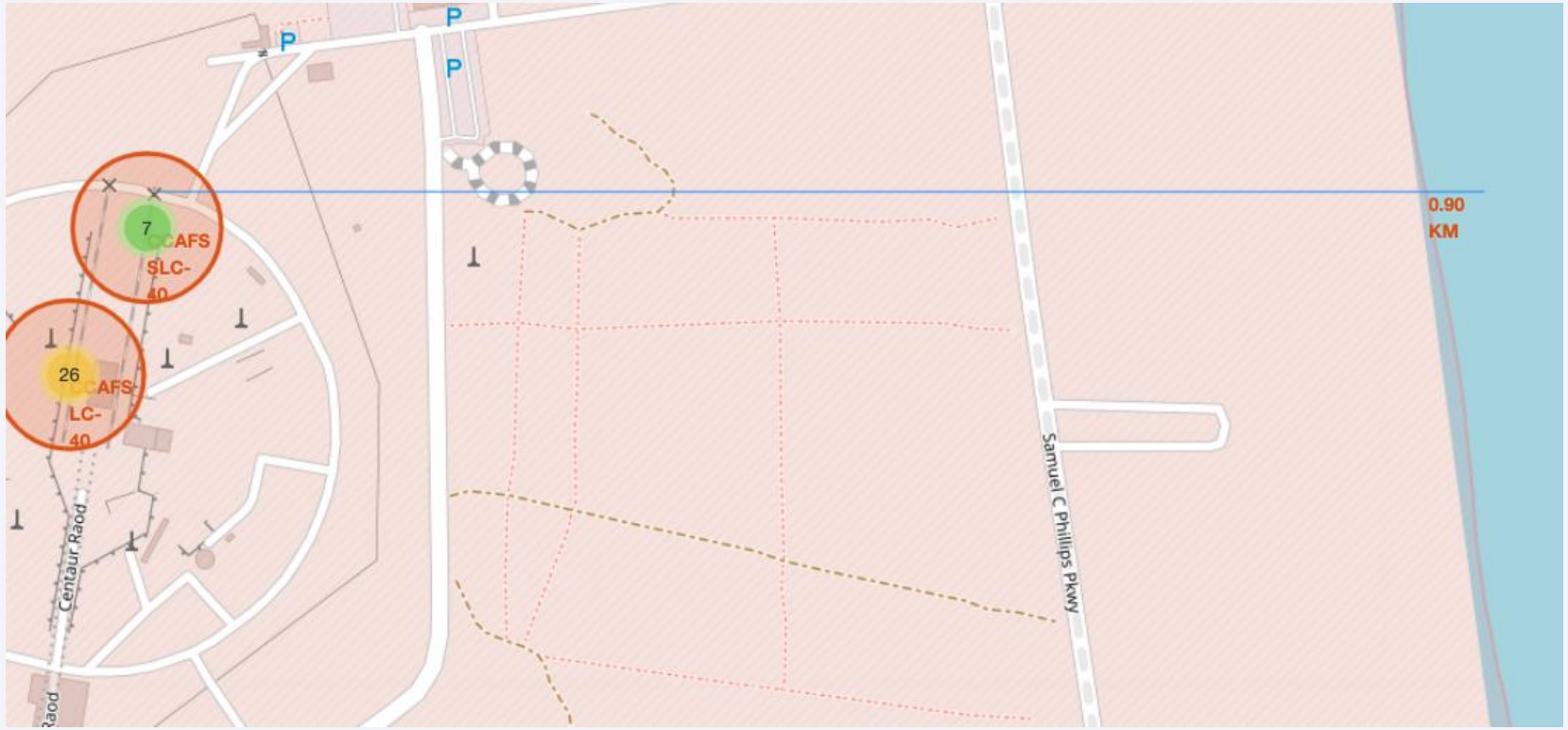
- Markers in the map represent the coordinate of the launch sites.
- They are all located in USA

Successful launch sites

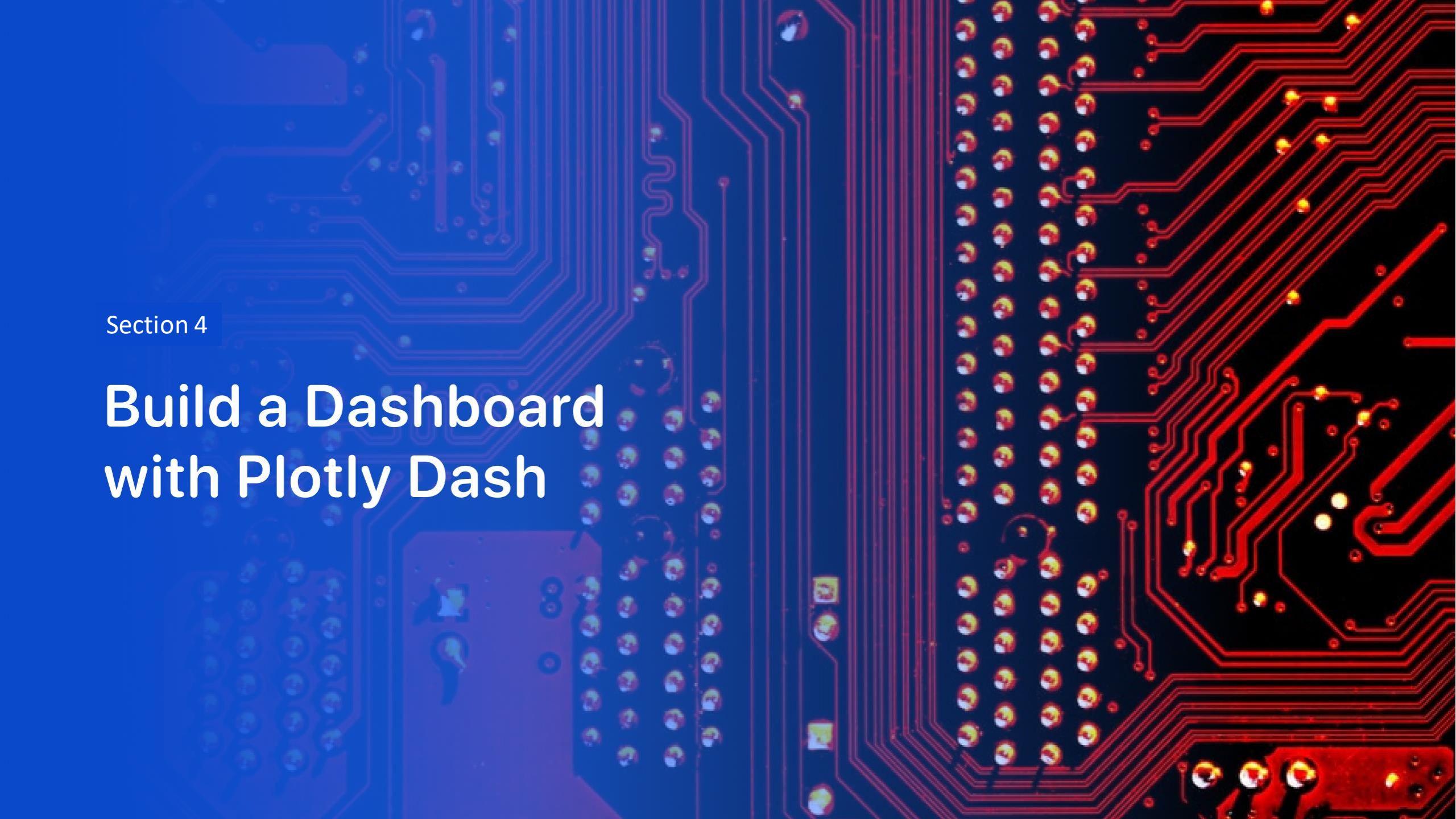


- Where are the most successful launch sites.
- Zoom and pan the maps included into interactive dashboard to find where space Y will have bigger probability to succed.

Infrastructures ready



- All selected launch sites for space Y are at low distances from transport and energy infrastructures.


Section 4
Build a Dashboard with Plotly Dash

Successful launches

- Where does Lauch Y success?

Almost 80% of success rate!

<Dashboard Screenshot 3>

- Replace <Dashboard screenshot 3> title with an appropriate title
- Show screenshots of Payload vs. Launch Outcome scatter plot for all sites, with different payload selected in the range slider
- Explain the important elements and findings on the screenshot, such as which payload range or booster version have the largest success rate, etc.

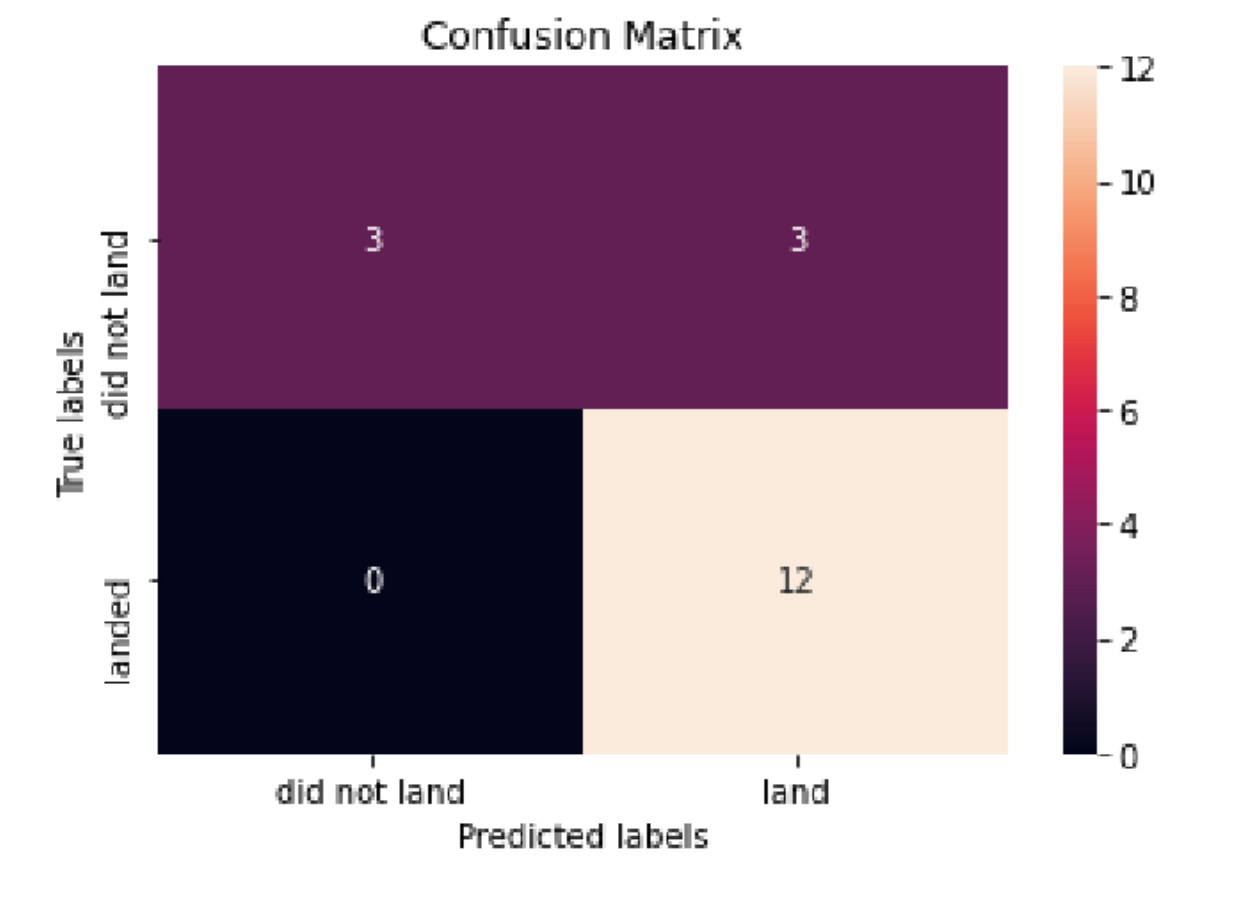
Section 5

Predictive Analysis (Classification)

Classification Accuracy

- Decision tree is the model with the highest accuracy.

Confusion Matrix



- Biggest problem is the false positive

Conclusions

- Technology still improving
- Payload mass is not a problem
- Some orbits are better than others
- At this stage of development classification tree is the most suitable ML model to adopt
- The trend for successful rate is steadly increasing

Appendix

- All notebook:
- https://github.com/fabiodisconzi/dataScienceCourse/tree/master

Thank you!