IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

<Name>
<Date>

Outline

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

Executive Summary

Summary of methodologies

Data collection via API, web scraping

Exploratory Data Analysis (EDA) with visualization

Interactive Map with Folium

Dashboards with Plotly Dash

Summary of all results

Exploratory Data Analysis results

Interactive maps and Dashboard

Machine learing prediction results

Introduction

- Project background and context
- This project is about to predict if the Falcon 9 first stage is successful or not. SpaceX uncover on their website about the Falcon 9 rocket launch cost and it is about 62 million dollars. Other providers cost it about 165 million dollars each. The price difference is explained by the fact that SpaceX can reus the first stage. For providers is very important to has successful launch with lease cost price. Of course this information very interesting for other companies if they want to complet it with SpaceX for a rocket launch.
- Problems you want to find answers
- What are the cause of a successful or failed landing?
- What are the effects of each relationship of the rocket variables on the success or failure of a landing?
- What are the conditions which will allow SpaceX to achieve the landing success rate?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:

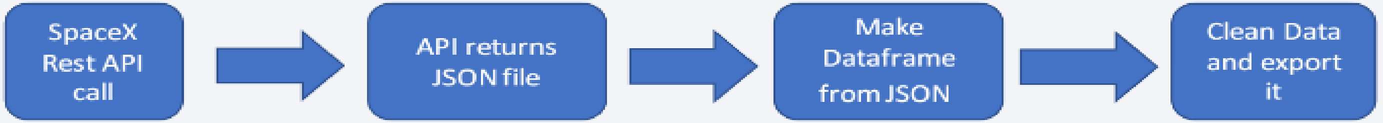
  .REST API SpaceX

  .Web Scraping from Wikipedia

- Perform data wrangling
  - Drop columns they are not used
  - Replace null values with mean value
  - Exploratory Data Analysis(EDA) to find some patterns in the data
- 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

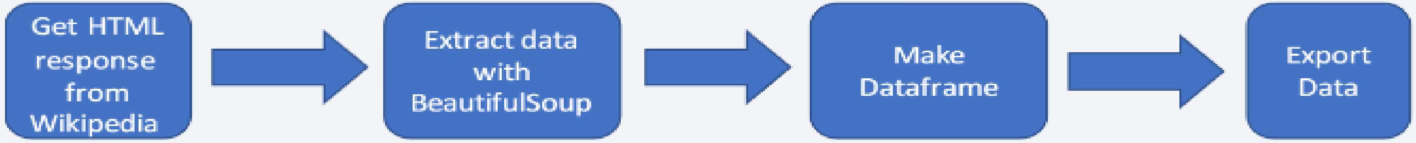
- Describe how data sets were collected.
- Datasets are collected from Rest API SpaceX and webscraping Wikipedia
- The information received from API are rocket,launches and payload columns
- The URL of the SpaceX REST API is api.spacexdata.com/v4/
- You need to present your data collection process use key phrases and flowcharts



SpaceX Rest API call → API returns JSON file → Make Dataframe from JSON → Clean Data and export it

The information received by webscraping from wikipedia are launches,landing and payload information

URL is = https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922



Get HTML response from Wikipedia → Extract data with BeautifulSoup → Make Dataframe → Export Data

Data Collection – SpaceX API

1.Getting response from API

SpaceX_url = url
Response = requests.get(SpaceX_url)

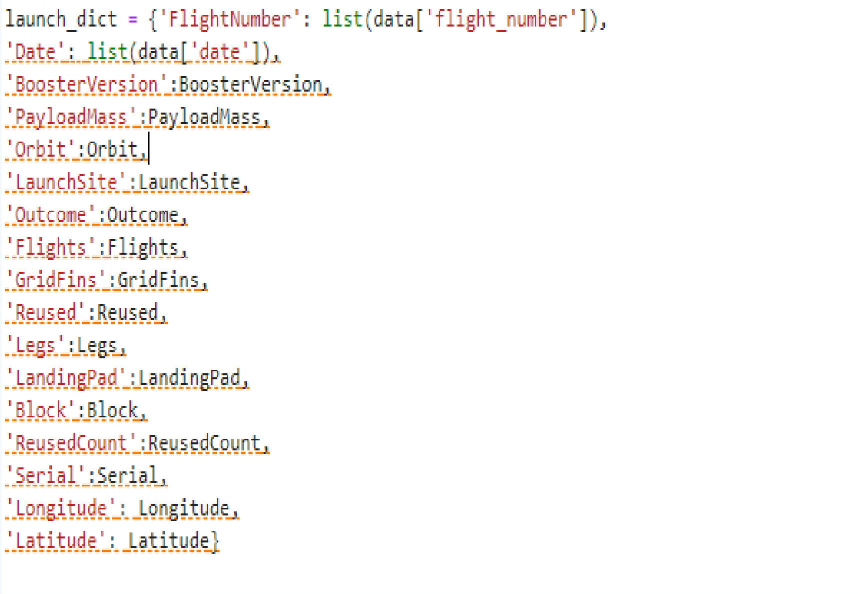
2.Convert Response to JSON file

Data = response.json()
Data pd.json_normalize(data)

3.Transform data with python functions

getLaunchSite(data)
getPayloadData(data)
getCoreData(data)
getBoosterVersion

4.Create dictionary with data



```
launch_dict = {'FlightNumber': list(data['flight_number']),
'Date': list(data['date']),
'BoosterVersion':BoosterVersion,
'PayloadMass':PayloadMass,
'Orbit':Orbit,
'LaunchSite':LaunchSite,
'Outcome':Outcome,
'Flights':Flights,
'GridFins':GridFins,
'Reused':Reused,
'Legs':Legs,
'LandingPad':LandingPad,
'Block':Block,
'ReusedCount':ReusedCount,
'Serial':Serial,
'Longitude': Longitude,
'Latitude': Latitude}
```

5.Create dataframe

Data = pd.DataFrame(launch_dict)

6.Filter dataframe and choose only falcone 9

Data_falcone9 =
data[data['Boosterversion']=='Falcone 9']

7.Export to file

Data_falcone9.to_csv('Data')

Data Collection - Scraping

1.Getting Response from HTML

Response = requests.get(static_url)

2.Create BeautifulSoup Object

Soup = BeautifulSoup(respons.text,'parser.html')

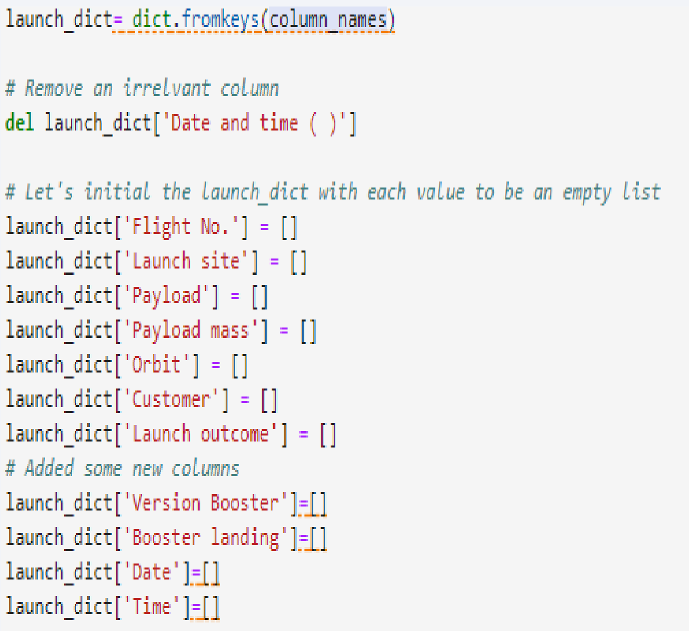
3.Find all tables

Tables = Soup.findAll('table')

4.Get column names

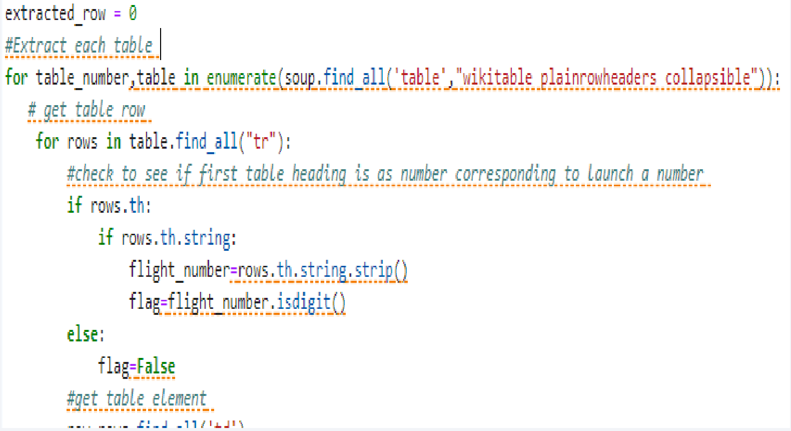
```
column_names = []
first_launch_table = first_launch_table.find_all('th')
for th in first_launch_table:
   column_name = extract_column_from_header(th)
   if (column_name is not None and len(column_name) > 0):
      column_names.append(column_name)
```

5.Creating Dictionary



```
launch_dict= dict.fromkeys(column_names)

# Remove an irrelvant column
del launch_dict['Date and time ( )']

# Let's initial the launch_dict with each value to be an empty list
launch_dict['Flight No.'] = []
launch_dict['Launch site'] = []
launch_dict['Payload'] = []
launch_dict['Payload mass'] = []
launch_dict['Orbit'] = []
launch_dict['Customer'] = []
launch_dict['Launch outcome'] = []
# Added some new columns
launch_dict['Version Booster']=[]
launch_dict['Booster landing']=[]
launch_dict['Date']=[]
launch_dict['Time']=[]
```

6.Add data to keys



```
extracted_row = 0
#Extract each table
for table_number,table in enumerate(soup.find_all('table',"wikitable plainrowheaders collapsible")):
   # get table row
    for rows in table.find_all("tr"):
        #check to see if first table heading is as number corresponding to launch a number
        if rows.th:
            if rows.th.string:
                flight_number=rows.th.string.strip()
                flag=flight_number.isdigit()
        else:
            flag=False
        #get table element
```

7.Create dataframe from dictionary

Df = pd.DataFrame(launch_data)

8.Export to file

Df.to_css('SpaceX_file.csv',index = False)

Data Wrangling

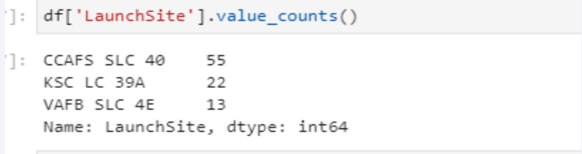
- **In this data collection there are several time where the booster did not land successfully .**

  . True RTLS,True Ocean,True ASDS means the mission are successfully occurred.

  . False Ocean,False RTLS,False ASDS means the missions were not successful.

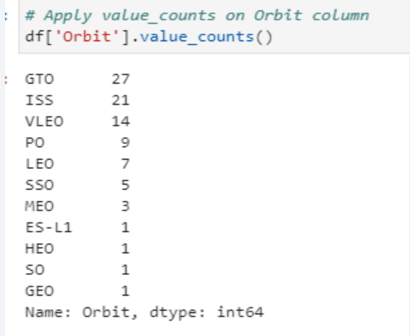
- **We should transform our data to catagorical variables which means 1 the mission has been successful and 0 means the mission is not successful.**

1 .Calculate the number of launches on each site



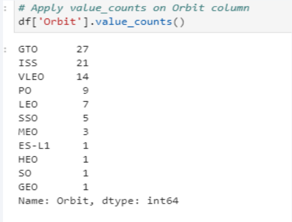
```
]: df['LaunchSite'].value_counts()

]: CCAFS SLC 40    55
   KSC LC 39A      22
   VAFB SLC 4E     13
   Name: LaunchSite, dtype: int64
```

2.Calculate the number and occurrence of each orbit



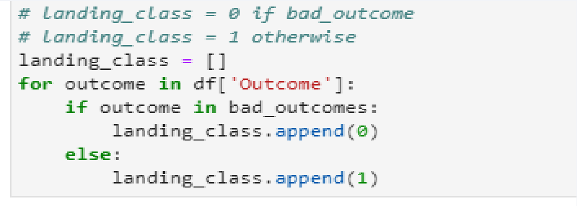
```
: # Apply value_counts on Orbit column
  df['Orbit'].value_counts()

: GTO      27
  ISS      21
  VLEO     14
  PO        9
  LEO       7
  SSO       5
  MEO       3
  ES-L1     1
  HEO       1
  SO        1
  GEO       1
  Name: Orbit, dtype: int64
```

3.Calculate the number and occurrence of each orbit



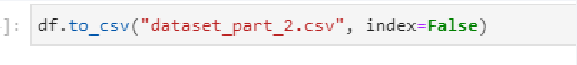
```
: # Apply value_counts on Orbit column
  df['Orbit'].value_counts()

: GTO      27
  ISS      21
  VLEO     14
  PO        9
  LEO       7
  SSO       5
  MEO       3
  ES-L1     1
  HEO       1
  SO        1
  GEO       1
  Name: Orbit, dtype: int64
```

4.Create a Landing outcome label from outcome column



```
# landing_class = 0 if bad_outcome
# landing_class = 1 otherwise
landing_class = []
for outcome in df['Outcome']:
    if outcome in bad_outcomes:
        landing_class.append(0)
    else:
        landing_class.append(1)
```

5. Export the data



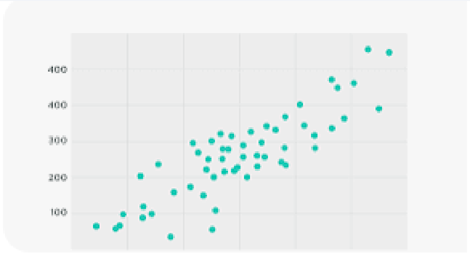
```
]: df.to_csv("dataset_part_2.csv", index=False)
```

- Add the GitHub URL of your completed data wrangling related notebooks, as an external reference and peer-review purpose

GitHub URL = Jamsnoori/IBM-SpaceX-Data-Science-Capston: Hands-on Lab: Complete the Data Collection API Lab (github.com)

EDA with Data Visualization

- <u>Scatter plot</u>
- Flight number vs Payload Mas
- Flight number vs Launch Site
- Payload vs Launch site
- Orbit vs Flight number

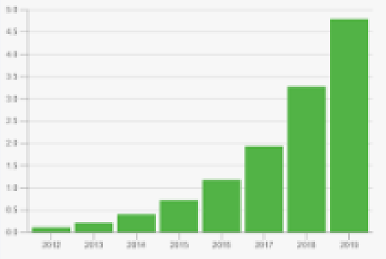


Scatter plot discover relationship between variables

Which 's called correlation relationship

. **Bar Graph**

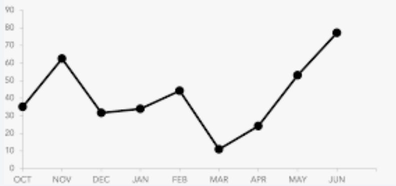
. Successfully rate vs.Orbit



Bar graph shows the relationship between numerical and categorical features

. Line Graph

. Successfully rate vs Year



Line graphs show data variables and their trends. Line graphs can help global behavior and make prediction for unseen data

.Add the GitHub URL of your completed EDA with data visualization notebook, as an external reference and peer-review purpose

Github url = IBM SpaceX Data Science Capston/Week_2_EDA_with_Visualization_lab.ipynb at main · Jamsnoori/IBM SpaceX Data Science Capston

EDA with SQL

- **We performed SQL queries to gather and understand the dataset**
  - Displaying the names of the unique launch sites in the space mission.
  - Displaying 5 records where lauch sites begin with string 'CCA'
  - Displaying the total payload mass carried by boosters launched by NASA (CRS)
  - Displaying average payload mass carried by booster version f9 v1.1
  - List the date when the first successful landing outcome in ground pad was archieved.
  - List the total number of successful and failure mission outcomes.
  - List the names of the booster_version which have carried the maximum payload mass
  - List the records which will display the month names, failure landing,booster versions,launch_site for the months in year 2015
  - Rank the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order
- Github url = IBM-SpaceX-Data-Science-Capston/Week_2_Exploratory_ Data_ Analysis_ Using_SQL.ipynb at main · Jamsnoori/IBM-SpaceX-Data-Science-Capston · GitHub

Build an Interactive Map with Folium

- Markers of all launch sites:
  - Added marker with Circle,Popup Label and Text Label of NASA Johnson Spaces Center using its latitude and longitude coordinate as a start location.
  - Added Markers with Circle, Popup Label and Text Label of all Launch Sites using their latitude and longitude coordinates to show their geographical locations and proximity to Equator and coasts.
- Coloured Markers of the launch outcomes for each Launch Sites:
  - Added coloured Markers of success (Green) and fialed (Red) lauches using Marker Cluster to indentify which launch sites have relatively high success rates.
- Distances between a launch Site to its proximities:
  - Added coloured Lines to show distances between the launch Site KSC LC-39A

  (as an example) and its proximities like Railway, Highway,Coastline and Closest City

Build a Dashboard with Plotly Dash

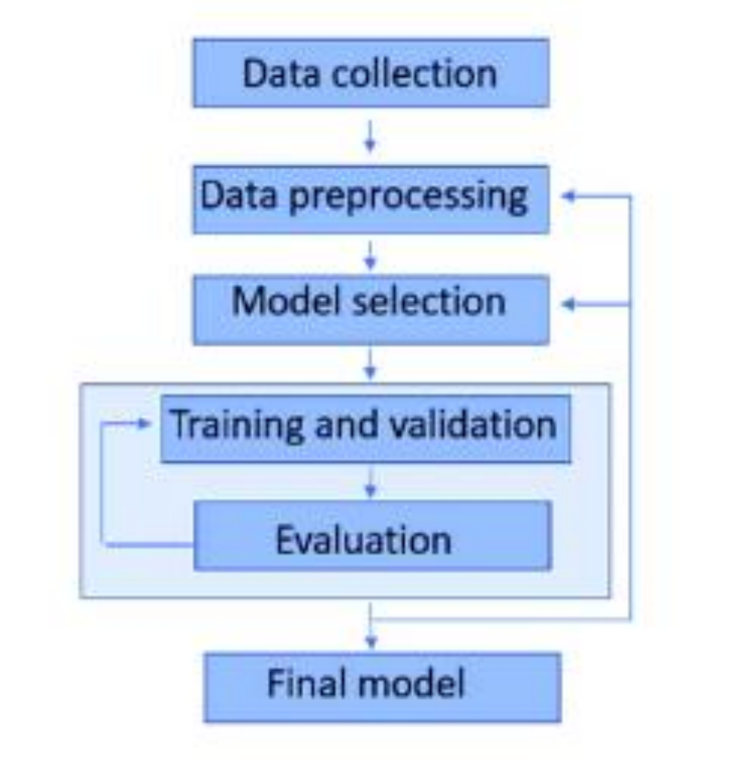
- **Dashboard has dropdown,pie chart,rangeslider and scatter plot components**
  - Dropdown allows us to choose the launch site or all launch sites

    (dash_core_components.Dropdown)
  - Pie chat shows the total success and the total failure for the launch site chosen with the dropdown component (plotly.express.pie)
  - Rangeslider allows a user to select a payload mass in fixed range

    (dash_core_components.RangeSlider)
  - Scatter chart shows the relationship between two variables, in particular success vs Payload Mass (plotly.express.scatter)

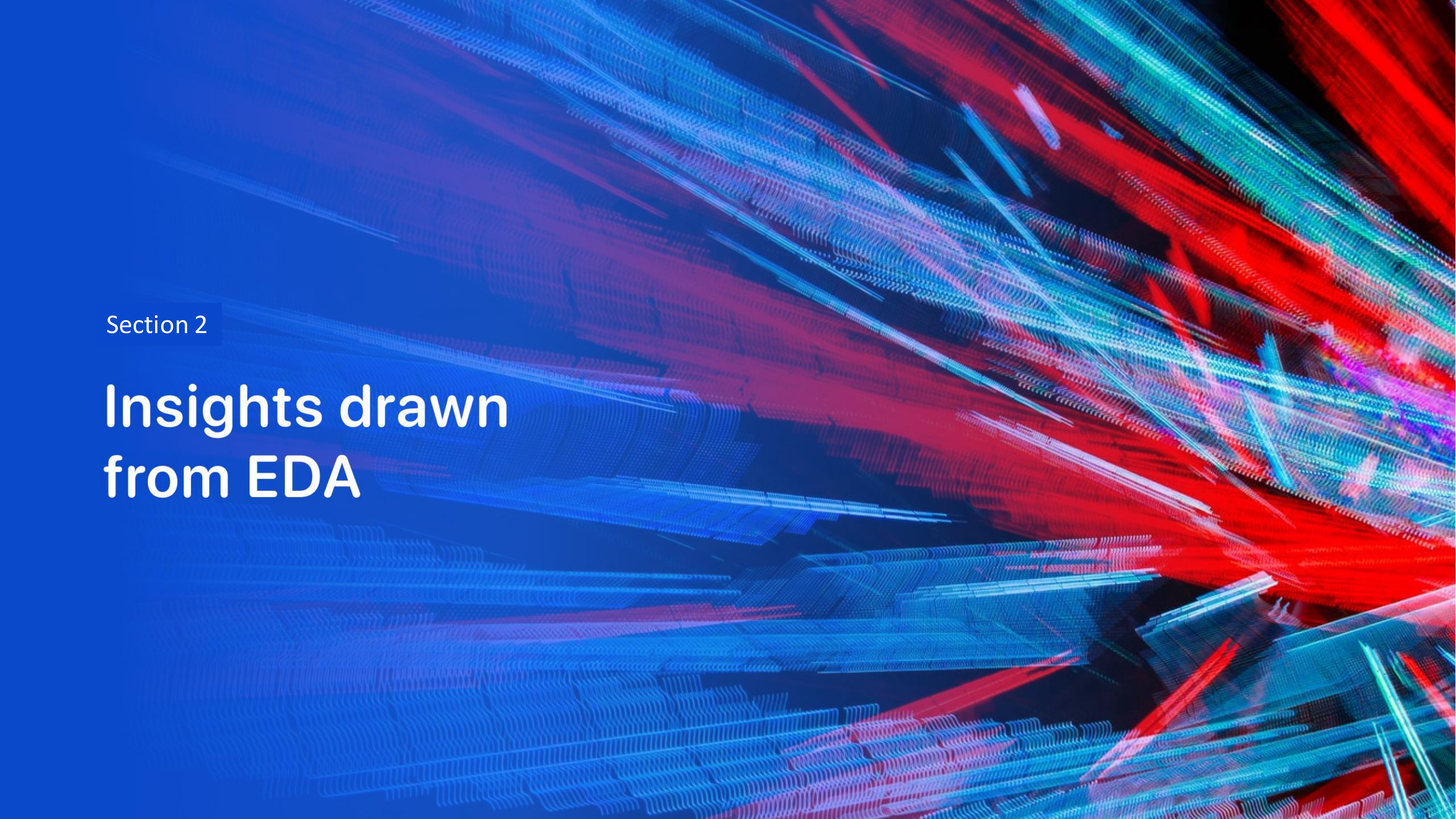
Predictive Analysis (Classification)

- Data preparation
  - Load dataset
  - Normalize data
  - Split data into training and test sets
- Model preparation
  - Selection of machine learning algorithms
  - Set parameters for each algorithm to GridSearchCV
  - Training GridSearchModel models with training dataset
- Model evaluation
  - Get best hyperparameters for each type of model
  - Compute accuracy for each model with training dataset
  - Plot Confusion Matrix
- GitHub url = [IBM-SpaceX-Data-Science-Capston/Week_4_Machine_Learning_Prediction_Lab.ipynb at main · Jamsnoori/IBM-SpaceX-Data-Science-Capston · GitHub]



Results

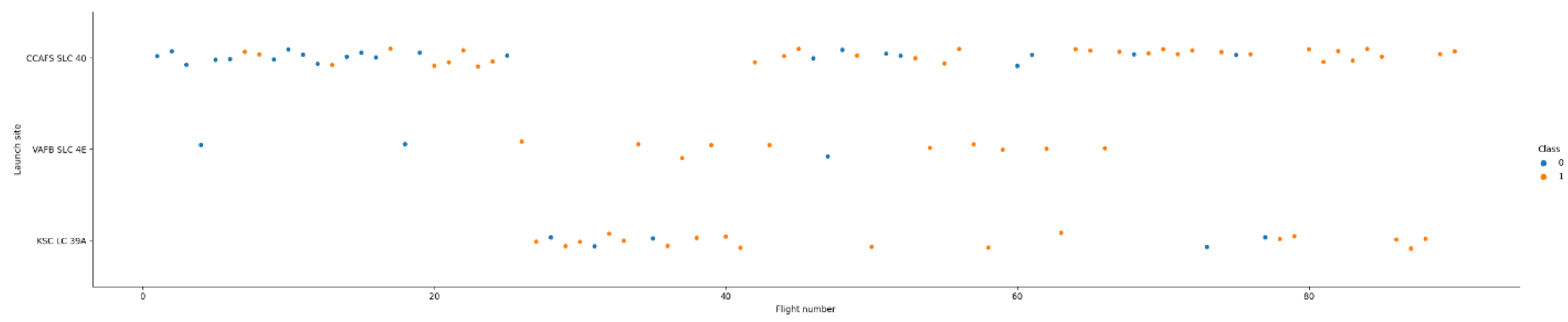
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results



Section 2

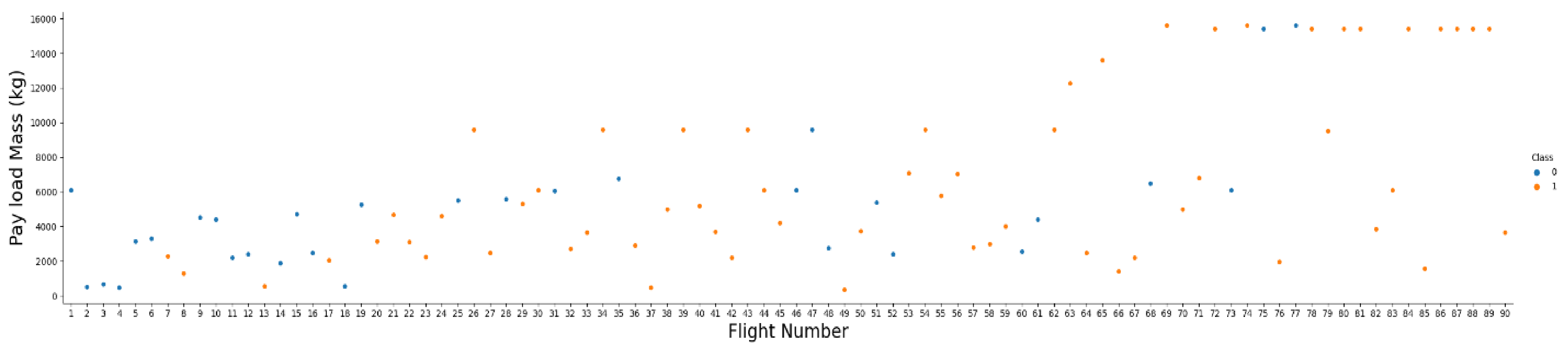
Insights drawn from EDA

Flight Number vs. Launch Site



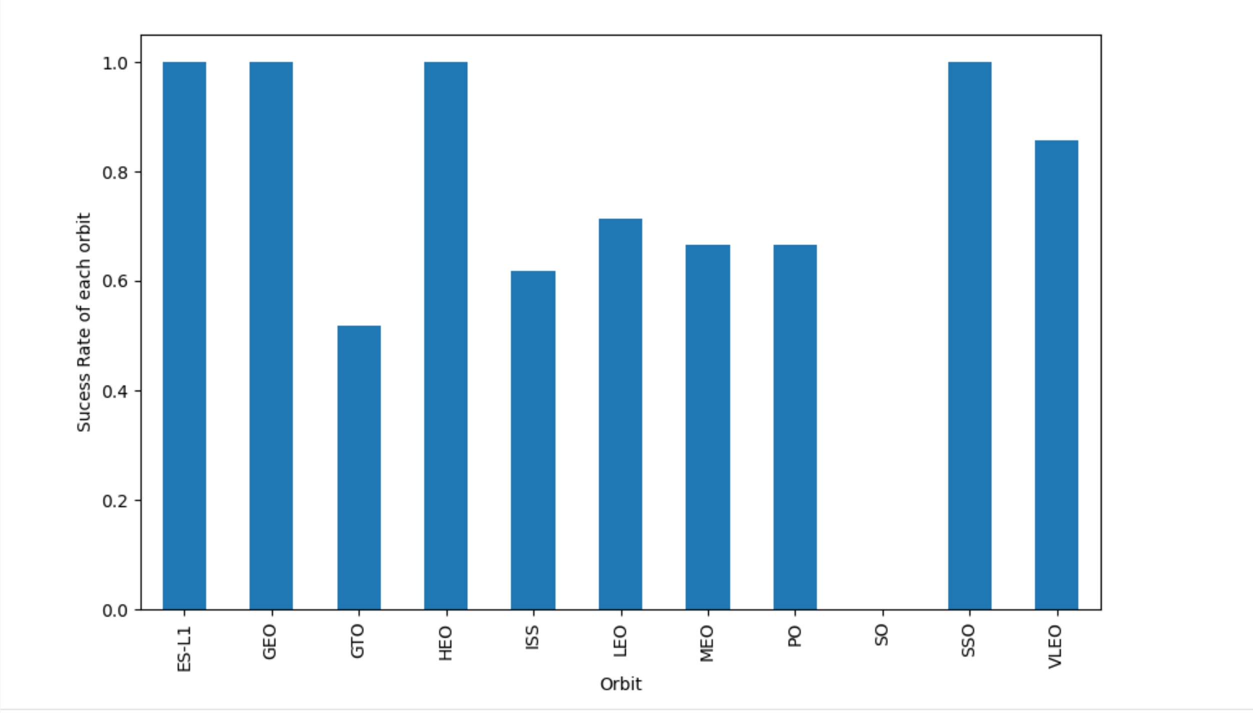
We discovered that the success rate is increasing in each site

Payload vs. Launch Site



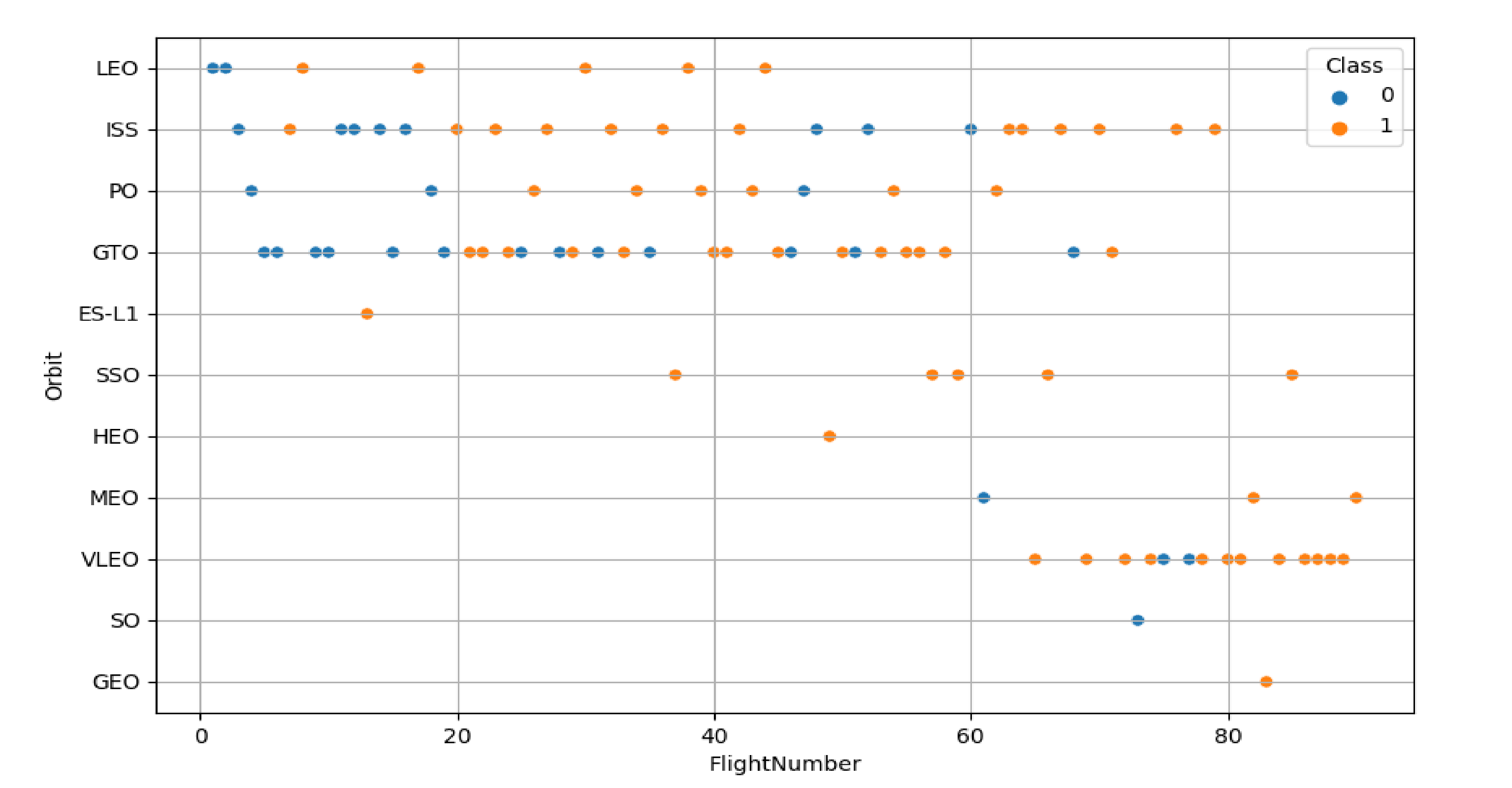
Denpending on the launch site, a heavier payload may be a consideration for successful landing . On the other hand, a to heavy payload can make a landing fail.

Success Rate vs. Orbit Type



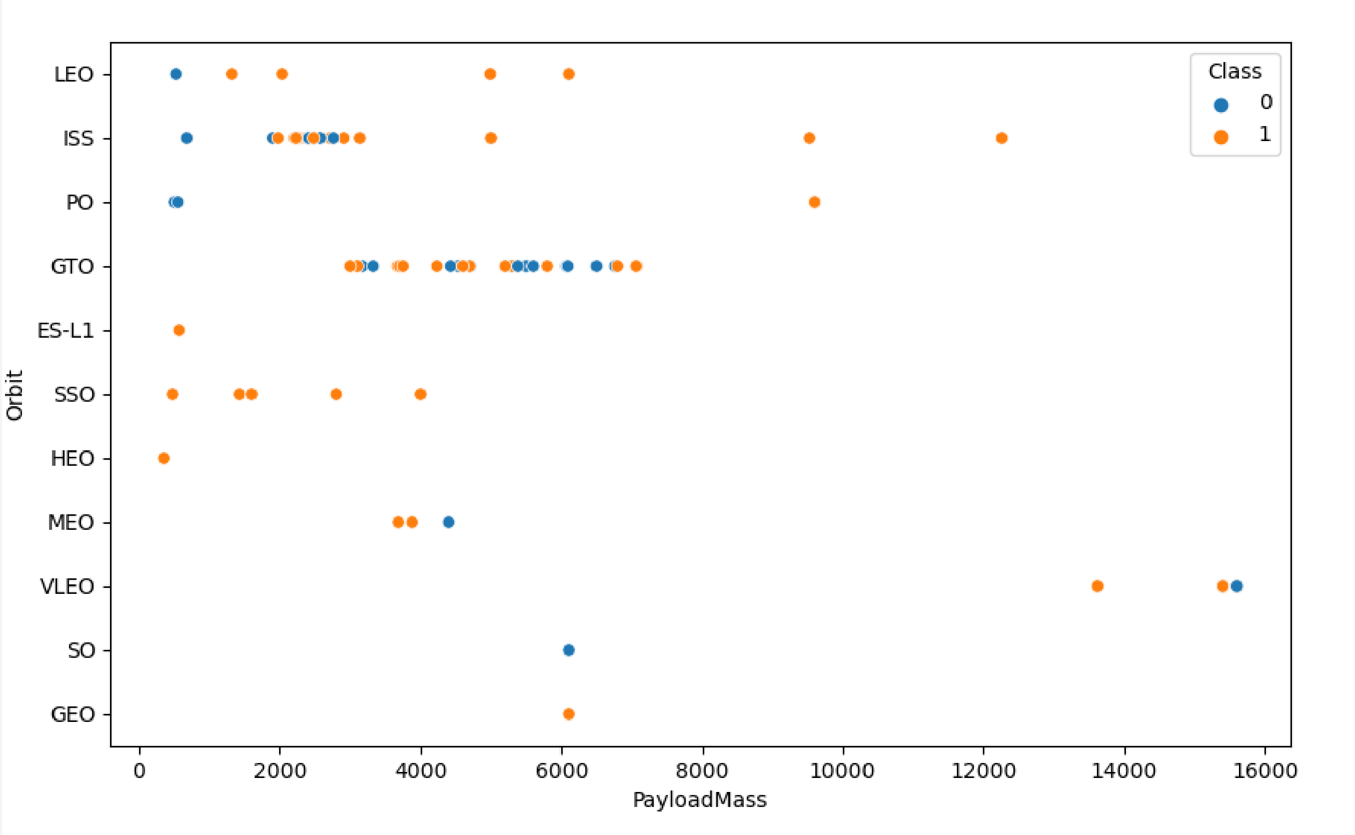
- We evaluate the success rate for different orbit types with this type of graphic. We can see the most successful are ES-L1,GEO,HEO and SSO.

Flight Number vs. Orbit Type



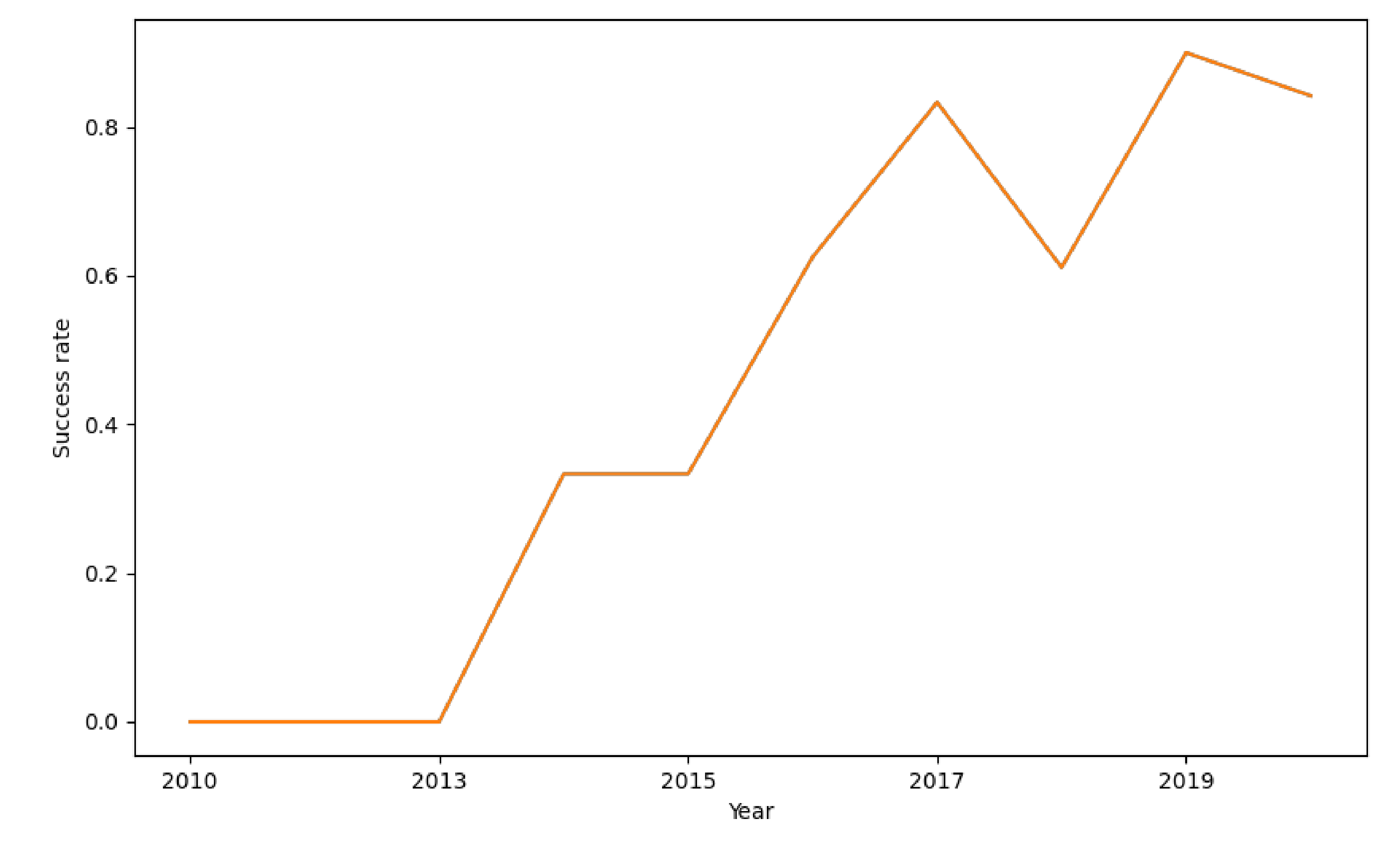
- We see that the success rate are increasing with the number of flights for the LEO orbit. For some orbits like GTO, there is no relation between the success rate and the number of flights. But we can suppose that the high success rate of some orbits like SSO or HEO is due to the knowledge learned during former launches for other orbits

Payload vs. Orbit Type



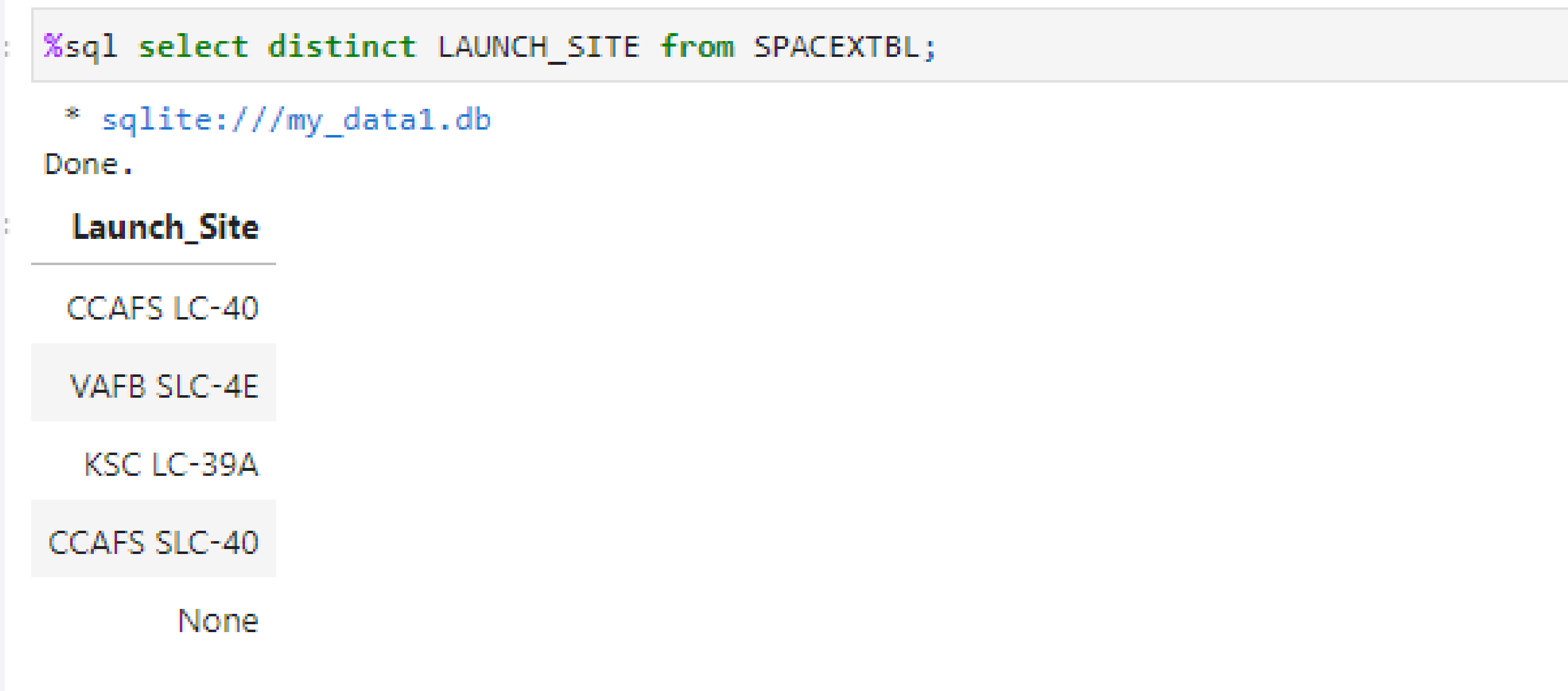
- The weight of the payloads can have a great impact on the success rate of the launches in certain orbits. For example, heavier payloads improve the success rate for the LEO orbit. Another finding is that decreasing the payload weight for GTO orbit improve the success of a launch.

Launch Success Yearly Trend



- You can see it on graphic chart since 2013 increase the success rate in the SpaceX Rocket

All Launch Site Names



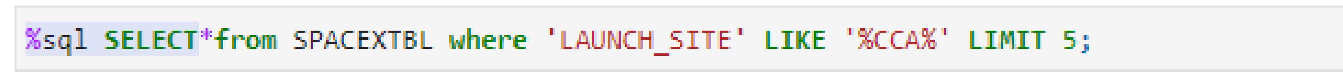
```
%sql select distinct LAUNCH_SITE from SPACEXTBL;
```

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

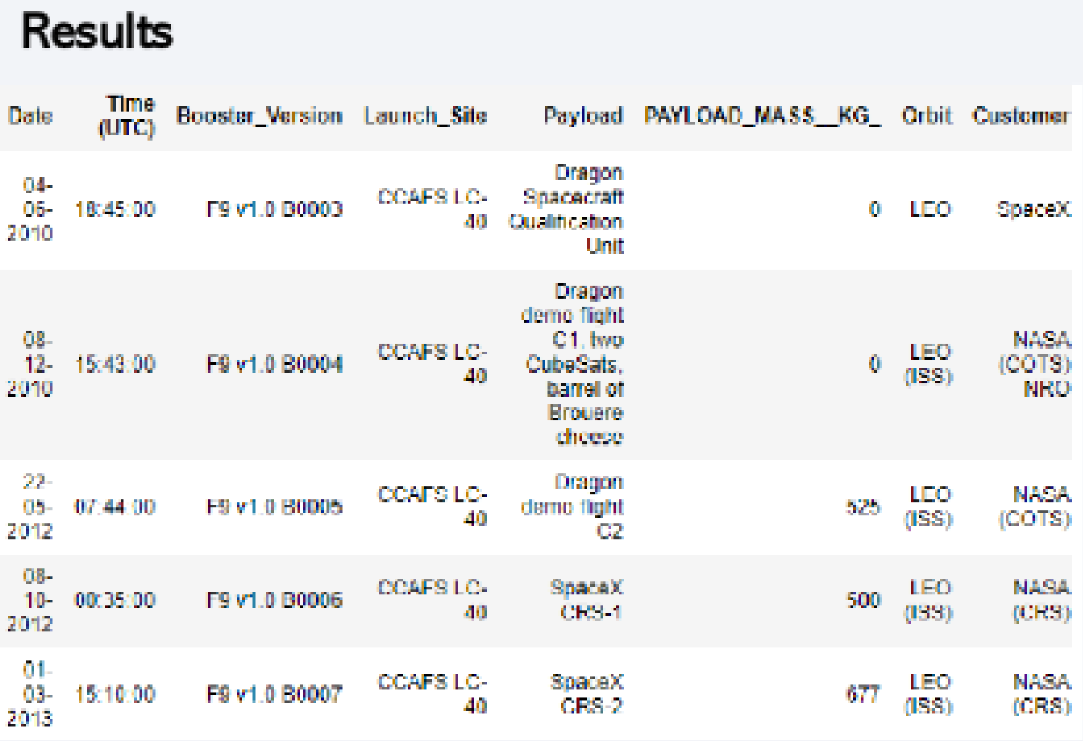
Launch_Site
CCAFS LC-40
VAFB SLC-4E
KSC LC-39A
CCAFS SLC-40
None

- Distinct in the query language looking for only unique values and remove the duplication

Launch Site Names Begin with 'CCA'



```
%sql SELECT*from SPACEXTBL where 'LAUNCH_SITE' LIKE '%CCA%' LIMIT 5;
```

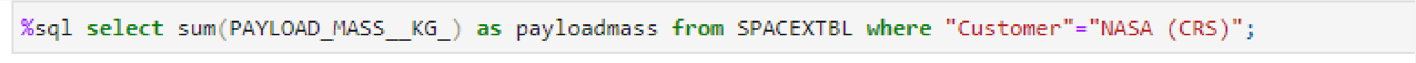


Results

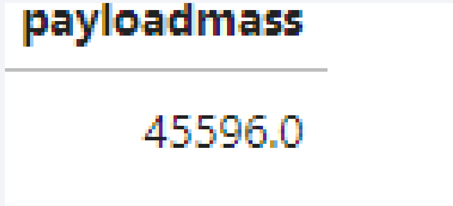
Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer
04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX
08-12-2010	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO
22-05-2012	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)
08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)
01-03-2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)

- The where note followed by like note filters launch sites that contain the substring CCA. LIMIT 5 shows only 5 rows from the results.

Total Payload Mass



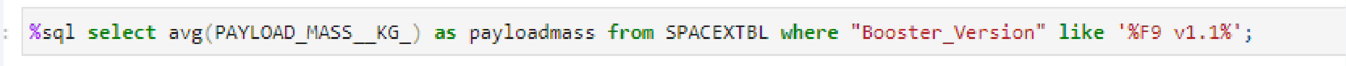
```
%sql select sum(PAYLOAD_MASS__KG_) as payloadmass from SPACEXTBL where "Customer"="NASA (CRS)";
```



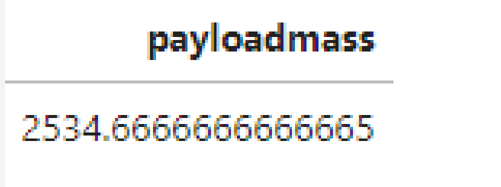
```
payloadmass
    45596.0
```

- This query returns the sum of all payload masses wheren the customer is NASA (CRS)

Average Payload Mass by F9 v1.1



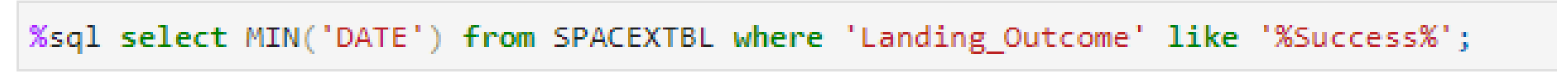
```
%sql select avg(PAYLOAD_MASS__KG_) as payloadmass from SPACEXTBL where "Booster_Version" like '%F9 v1.1%';
```



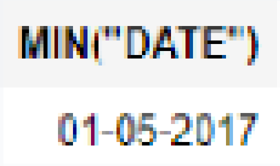
payloadmass
2534.6666666666665

- This query returns the average of all payload masses where the booster version contains the substring F9 v1.1

First Successful Ground Landing Date



```
%sql select MIN('DATE') from SPACEXTBL where 'Landing_Outcome' like '%Success%';
```

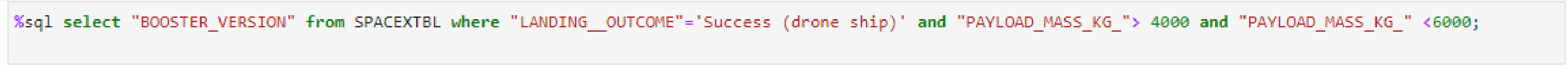


MIN("DATE")
01-05-2017

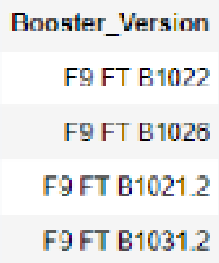
- We select the older successful landing with code.

The where code bring only the successful landing . With the MIN function you choose only the oldest launch date

Successful Drone Ship Landing with Payload between 4000 and 6000



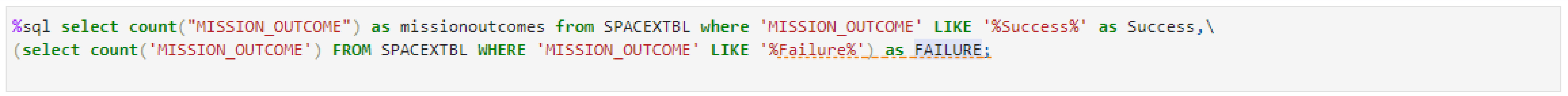
```
%sql select "BOOSTER_VERSION" from SPACEXTBL where "LANDING__OUTCOME"='Success (drone ship)' and "PAYLOAD_MASS_KG_"> 4000 and "PAYLOAD_MASS_KG_" <6000;
```



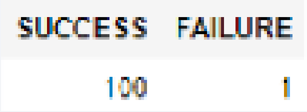
Booster_Version
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
F9 FT B1031.2

- This query returns the booster version where landing was successful and payload mass is between 4000 and 6000kg. The WHERE and AND clauses filter the dataset.

Total Number of Successful and Failure Mission Outcomes



```
%sql select count("MISSION_OUTCOME") as missionoutcomes from SPACEXTBL where 'MISSION_OUTCOME' LIKE '%Success%' as Success,\
(select count('MISSION_OUTCOME') FROM SPACEXTBL WHERE 'MISSION_OUTCOME' LIKE '%Failure%') as FAILURE;
```

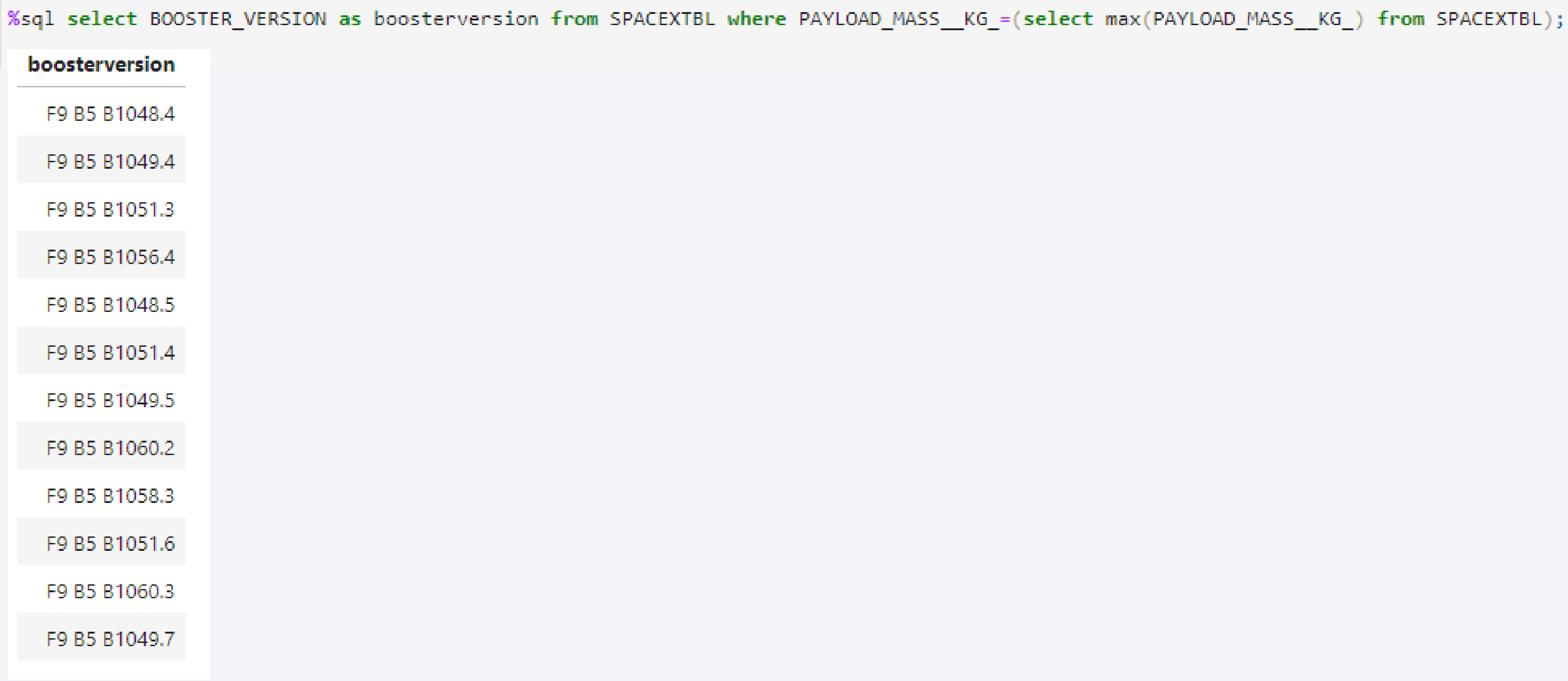


SUCCESS	FAILURE
100	1

- With the first Select , we show the subqueries that return results. The first subquery counts the successful mission. The second subquery counts the unsuccessful mission.

The WHERE clause followed by LIKE clause filters mission outcome. The COUNT function counts records filtered

Boosters Carried Maximum Payload

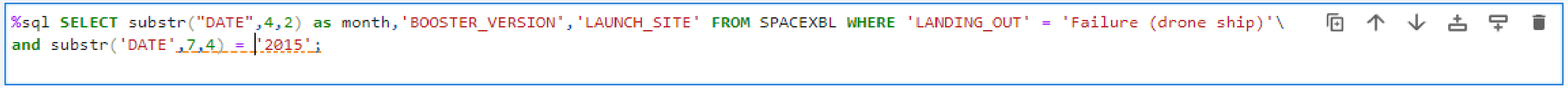


```
%sql select BOOSTER_VERSION as boosterversion from SPACEXTBL where PAYLOAD_MASS__KG_=(select max(PAYLOAD_MASS__KG_) from SPACEXTBL);
```

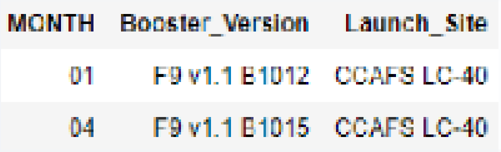
boosterversion
F9 B5 B1048.4
F9 B5 B1049.4
F9 B5 B1051.3
F9 B5 B1056.4
F9 B5 B1048.5
F9 B5 B1051.4
F9 B5 B1049.5
F9 B5 B1060.2
F9 B5 B1058.3
F9 B5 B1051.6
F9 B5 B1060.3
F9 B5 B1049.7

- We used a subquery to filter data by returning only the heaviest payload mass with MAX function. The main query uses subquery restults and returns unique booster version SELECT DISTINCT with the heaviest payload mass.

2015 Launch Records



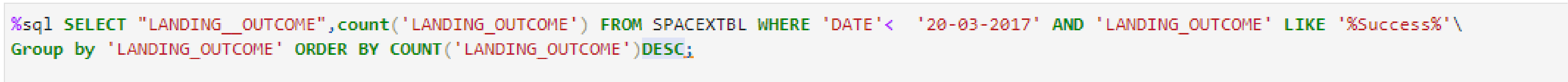
```
%sql SELECT substr("DATE",4,2) as month,'BOOSTER_VERSION','LAUNCH_SITE' FROM SPACEXBL WHERE 'LANDING_OUT' = 'Failure (drone ship)'\
and substr('DATE',7,4) = '2015';
```



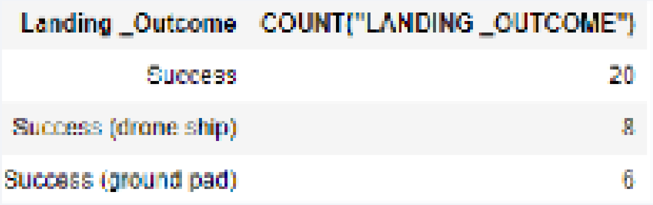
MONTH	Booster_Version	Launch_Site
01	F9 v1.1 B1012	CCAFS LC-40
04	F9 v1.1 B1015	CCAFS LC-40

- This query returns month ,booster version,launch site where landing was unsuccessful and landing date took place in 2015. Substr(DATE,4,2) shows month.substr(Date,7,4) shows the year

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



```
%sql SELECT "LANDING__OUTCOME",count('LANDING_OUTCOME') FROM SPACEXTBL WHERE 'DATE'<  '20-03-2017' AND 'LANDING_OUTCOME' LIKE '%Success%'\
Group by 'LANDING_OUTCOME' ORDER BY COUNT('LANDING_OUTCOME')DESC;
```



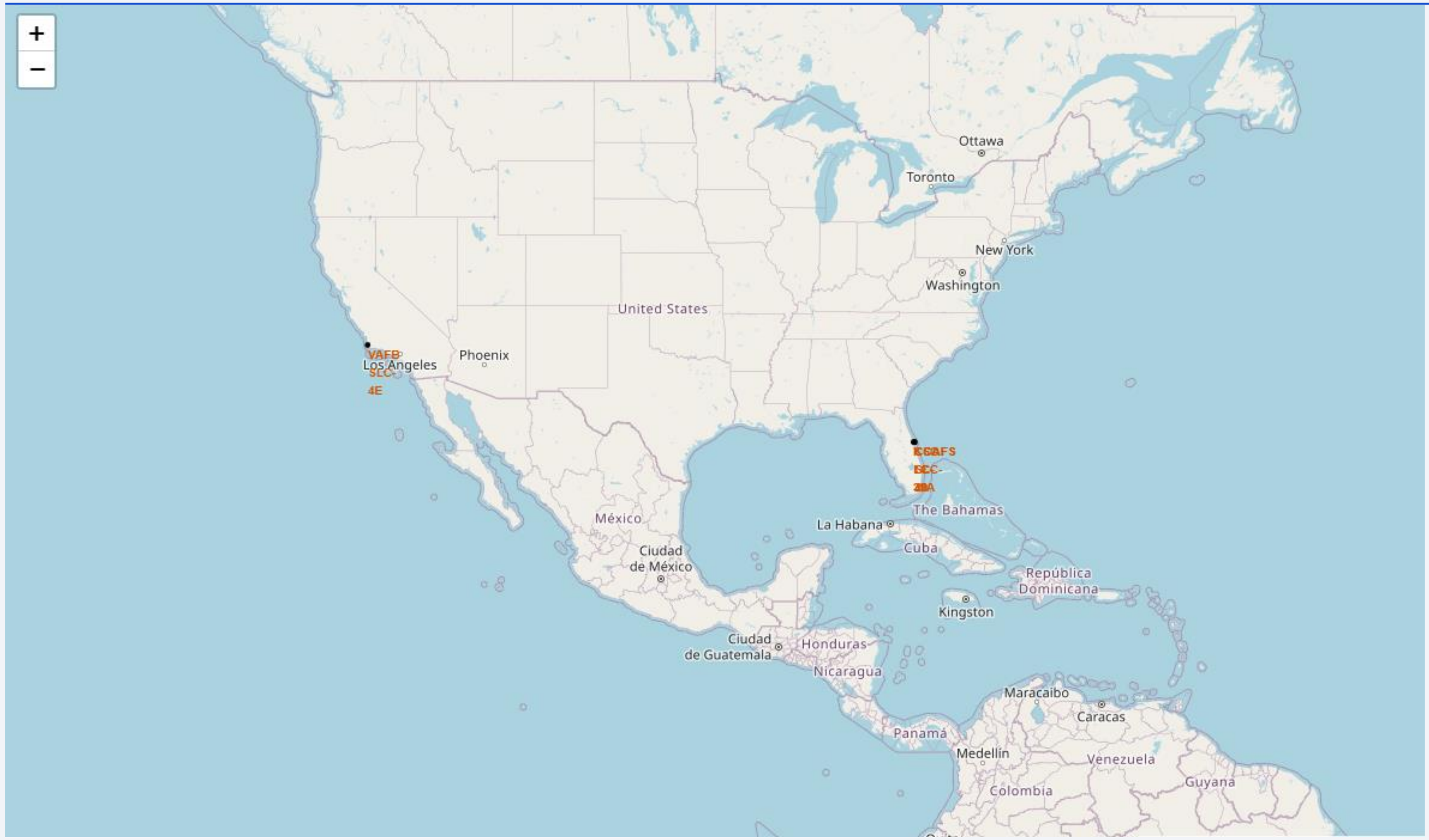
Landing _Outcome	COUNT("LANDING _OUTCOME")
Success	20
Success (drone ship)	8
Success (ground pad)	6

This query returns landing outcomes and their count where mission was successful and date is between 04-06-2010 and 20-30-2017 . The GROUP BY clause groups results by landing outcome and ORDER BY COUNT DESC shows results in decreasing order.

Section 3

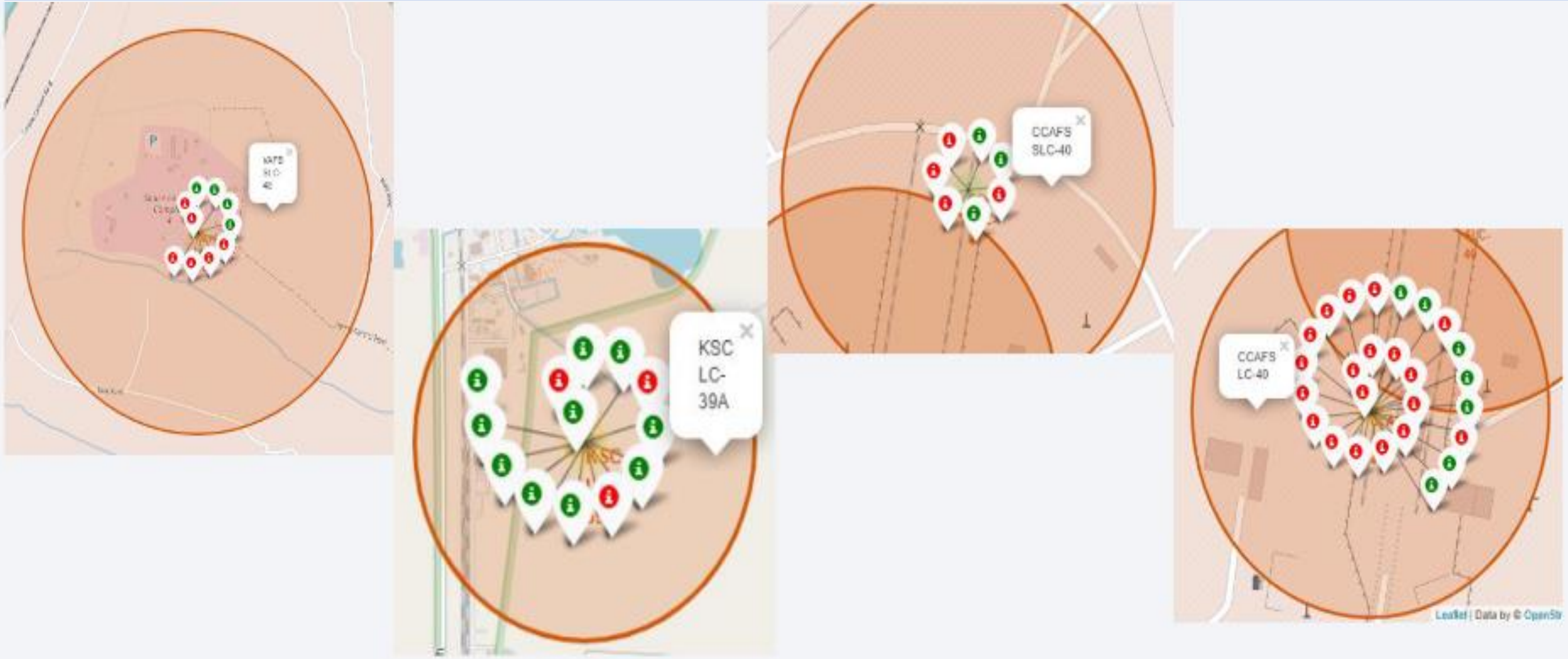
Launch Sites Proximities Analysis

Folium map-Launch stations



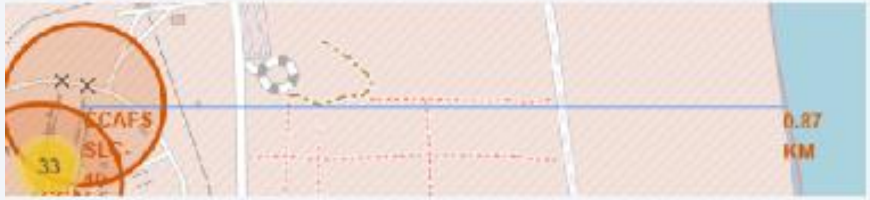
We can see launch station of SpaceX near to the sea and they are located to the United States

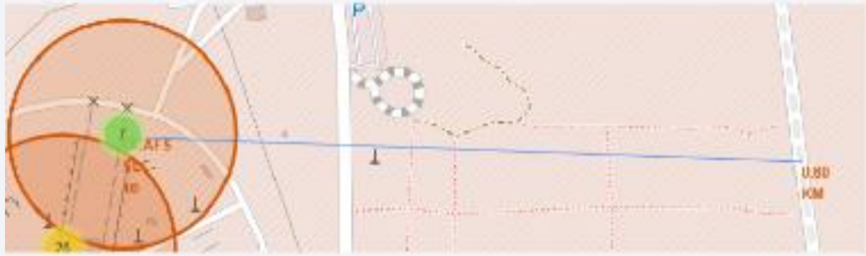
Folium map – Color Labeled Markers

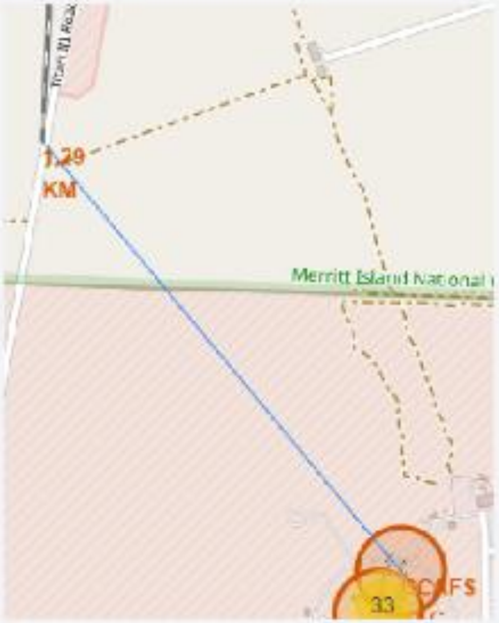


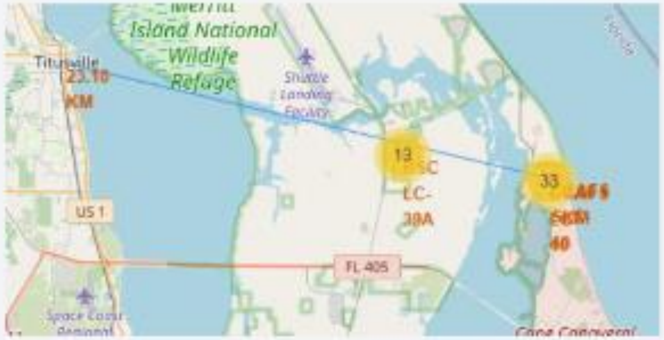
Green markers represents successful launches.Red markers represents unsuccessful launches. We note that KSC LC-39A has a higher launch success rate.

Folium Map – Distances between CCAFS SLC-40 and its proximities









CCAFS SLC-40 is close to railways

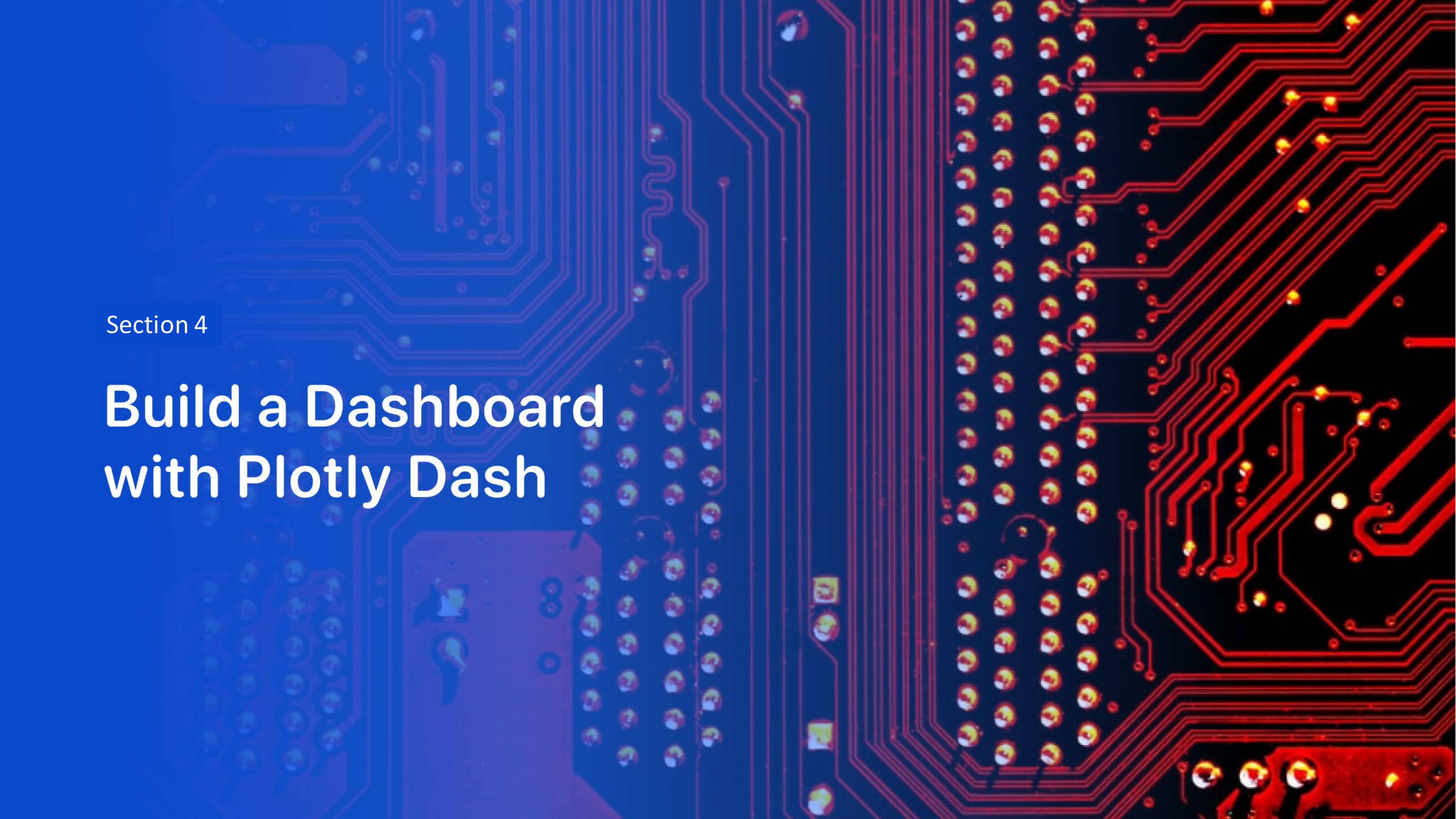
CCAFS SLC-40 is close to highways

CCAFS SLC-40 is close to coastline

Do CCAFS SLC-40 keeps certain distance away from cities? No

Section 4

Build a Dashboard with Plotly Dash

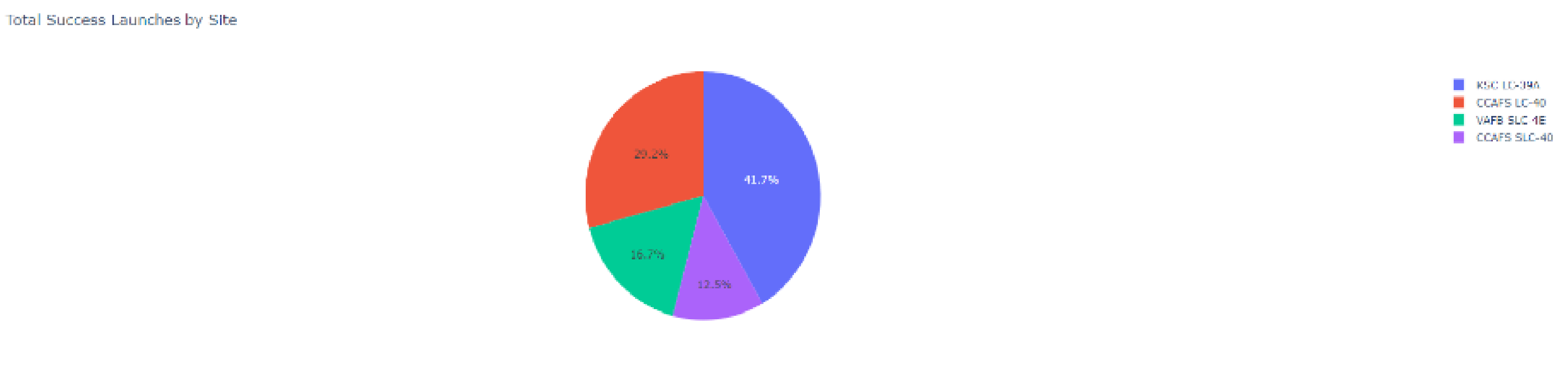


Dashboard – Total success by

- Replace <Dashboard screenshot 1> title with an appropriate title

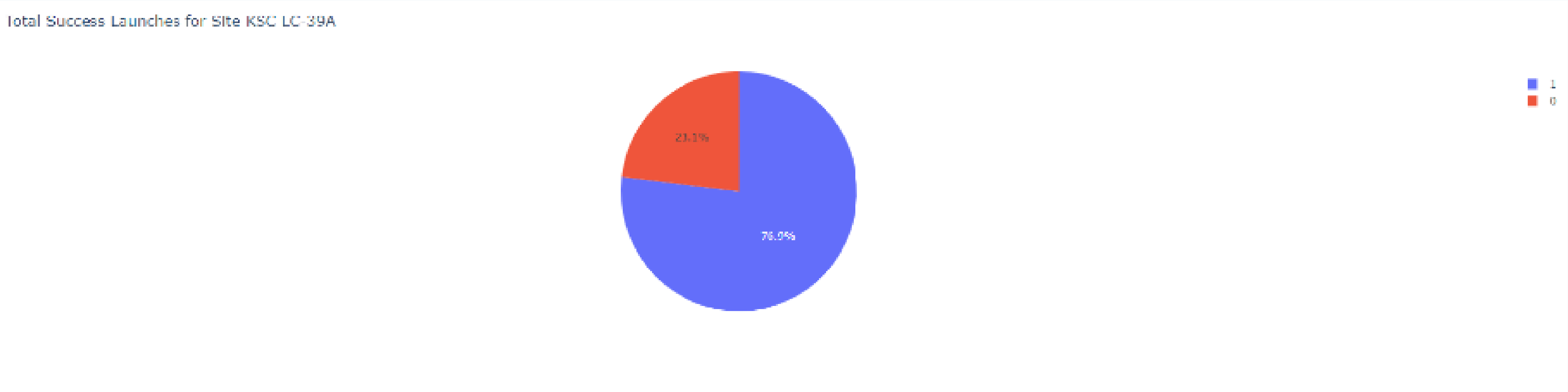
- Show the screenshot of launch success count for all sites, in a piechart

Dashboard – Total success by



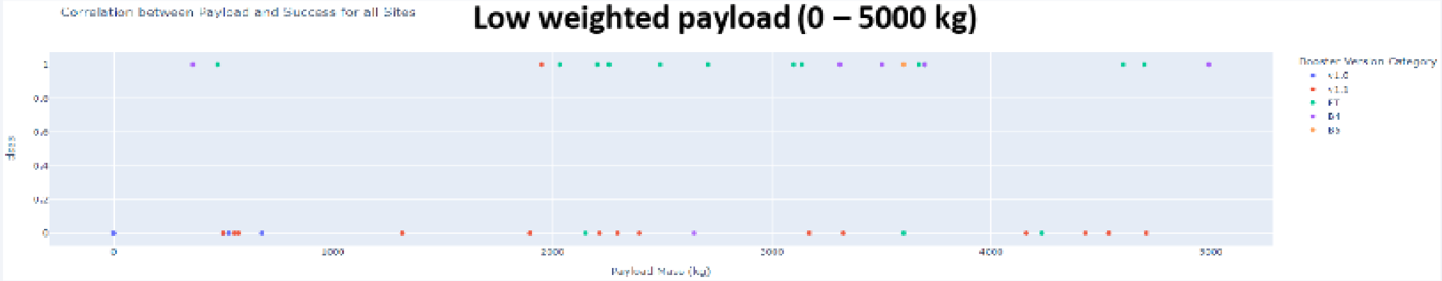
We can see that KSC LC-39A has the best success rate of launches

Dashboard-Total success launches for Site KCS LC-39A

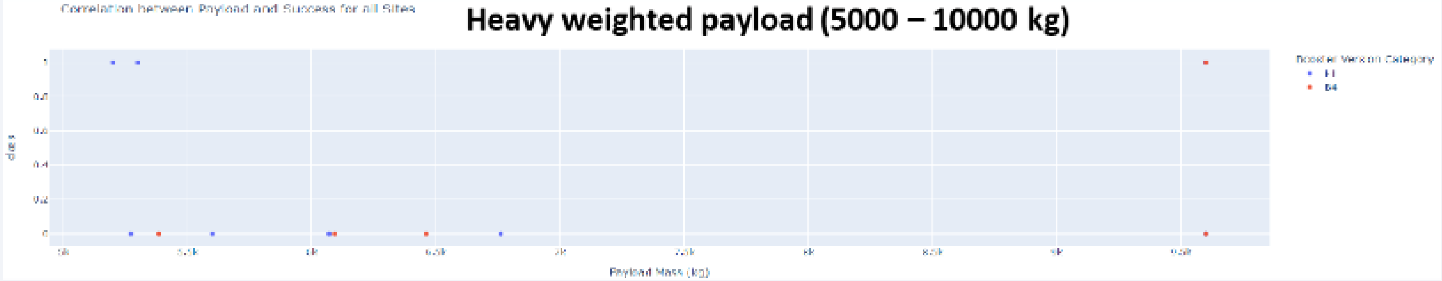


We see that KSC LC-39A has achieved a 76.9% success rate while getting a 23.1 failure rate.

Dashboard –Payload mass vs outcome for all sites with different payload mass selected



Low weighted payload (0 – 5000 kg)



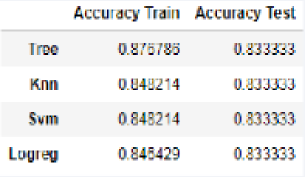
Heavy weighted payload (5000 – 10000 kg)

Low weighted payloads have a better success rate than the heavy weighted payload.

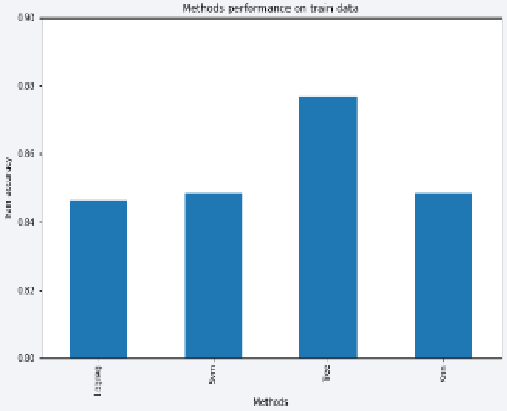
Section 5

Predictive Analysis (Classification)

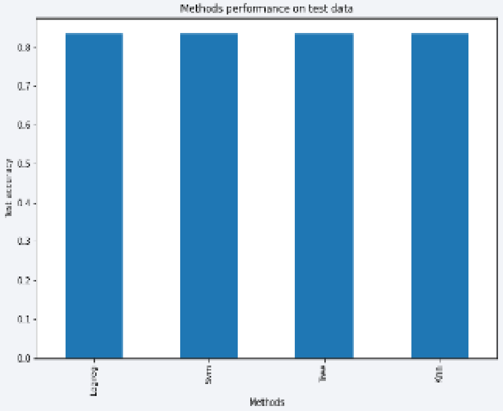
Classification Accuracy



	Accuracy Train	Accuracy Test
Tree	0.876786	0.833333
Knn	0.848214	0.833333
Svm	0.848214	0.833333
Logreg	0.846429	0.833333



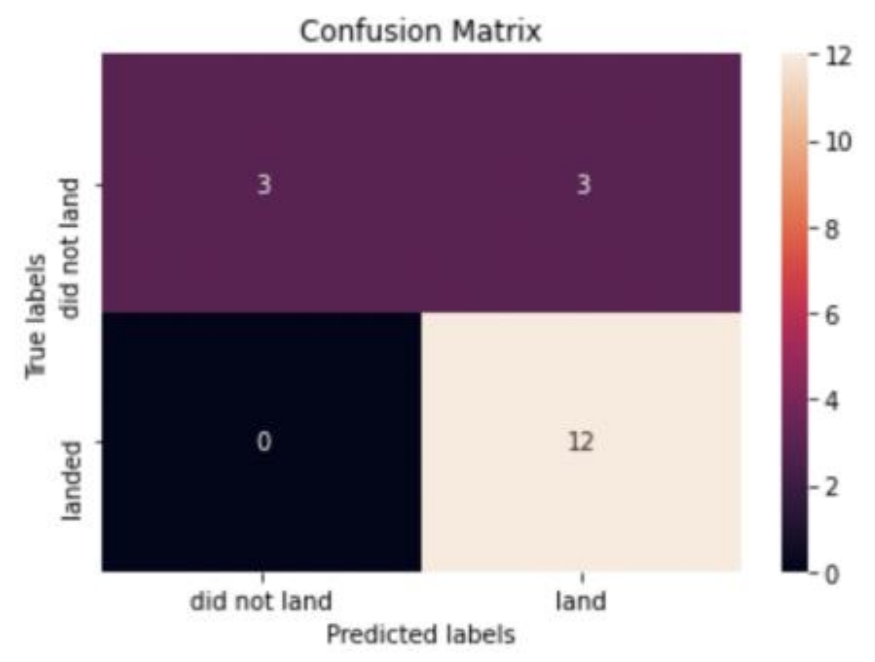
Methods performance on train data

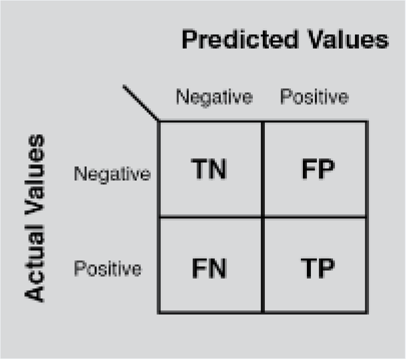


Methods performance on test data

For accuracy test, all methods performed similar. We could get more test data to decide between them. But right choice would be decision tree

Confusion Matrix





Actual Values \ Predicted Values	Negative	Positive
Negative	TN	FP
Positive	FN	TP

As the test accuracy are all equal, the confusion matrices are also indentical. The main problem of these models are false positives

Conclusions

- The success of a mission can be expained by several factors such as the launch site, the orbit and especailly the number of previous launches. Indeed, we can assume the there has been a gain in knowledge between launches tha allowed to go from a launch failure to a success.
- The orbits with the best success rates are GEO, HEO,SSO and ES-L1
- Depending on the orbits, the payload mass can be a criterion to take into account for the success of a mission. Some orbits require a light or heavy payload mass. But generally low weighted payloads perform better than the heavy weighted payloads.
- With the current data, we cannot explain why some launch sites are better than other (KCS LC-39A is the best launch site). To get an answare to this problem, we could obtain atmospheric or other relevant data.
- For this dataset, we choose the Decision Tree Algorithm as the best model even if the test accuracy between all the models used is identical. We choose Decision Tree Algorithm because it has a better train accuracy

Thank you!