IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Ramón Colombás Calafat
11/07/2024

Outline

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

Executive Summary

- ❑ Data W Summary of methodologies
  - ❖ Data collection using SpaceX API
  - ❖ Data Collection with Web Scraping
  - ❖ 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
  - ❖ Machine Learning Predictive analysis (Classification)
- ❑ Summary of all results
  - ❖ Exploratory Data Analysis result
  - ❖ Interactive analytics in screenshots
  - ❖ Predictive Analytics result

Introduction

- ❑ Project background and context
  - ❖ SpaceX is a company that designs, manufactures, and launches advanced rockets and spacecraft. SpaceX's accomplishments include:
    - ➢ Sending spacecraft to the International SpaceStation.
    - ➢ Starlink, a satellite internet constellation
  - ➢ SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars SpaceX's. Other providers cost upwards of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage
  - ➢ If we can determine if the first stage will land with Machine Learning Models, we can determine the cost of a launch.
- ❑ Problems you want to find answers
  - ➢ What factors influence the landing success on first stage?
  - ➢ The effect of each relationship of rocket variables on outcome

Section 1

Methodology

Methodology

Executive Summary

- ❑ Perform Data collection methodology:
  - ❖ Using SpaceX Rest API
  - ❖ Using Web Scrapping from
- ❑ Data Wrangling
  - ❖ Filtering the data
  - ❖ Dealing with missing values
  - ❖ One-hot encoding was applied to categorical features
- ❑ 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, tune, evaluate classification models and creating best machine learning model

Data Collection

- ❑ Enter the url of the web page to analyze for SpaceX
- ❑ Data collection using get request to the SpaceX API
- ❑ Decoded the response code using .json() function call and turn into a pandas dataframe using .json_normalize()
- ❑ Use the API to get information for each launch
- ❑ Filter the Data Frame to include Falcon 9 launches with Web Scrapping
- ❑ Clean the data, checked for missing values and fill in missing values where necessary

Data Collection – SpaceX API

- ❑ SpaceX API by making a get request to the SpaceX API
- ❑ GET request by parse the SpaceX launch data
- ❑ Decode the response content as a Json result and then converted into a Pandas data frame
- ❑ GitHub URL: [Data Collection API]()

Task 1: Request and parse the SpaceX launch data using the GET request

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

```
[9]: static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_call_spacex_ap
```

```
[10]: response = requests.get(static_json_url)
```

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

```
[11]: response.status_code
```

```
[11]: 200
```

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

```
[12]: # Use json_normalize meethod to convert the json result into a dataframe

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

Data Collection - Scraping

- ❑ Webscrap Falcon 9 launch records with BeautifulSoup
- ❑ Parse the table and convert it into a pandas dataframe
- ❑ GitHub URL: [Data Collection Scrapping]

TASK 1: Request the Falcon9 Launch Wiki page from its URL

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

```
# use requests.get() method with the provided static_url
# assign the response to a object
response = requests.get(static_url)
```

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

```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
data  = response.text
soup = BeautifulSoup(data,"html.parser")
```

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

```
# Use soup.title attribute
tag_object=soup.title
```

Data Wrangling

- Calculate the number of launches on each site
- Calculate the number and occurrence of each orbit
- Calculate the number and occurence of mission outcome per orbit type
- Create a landing outcome label from Outcome column using one-hot encoding
- GitHub URL: Data Wrangling

EDA with Data Visualization

- ❑ Scatter plots show the relationship between variables.
- ❑ Bar charts show comparisons among discrete categories.
- ❑ Line charts show trends in data over time (time series).
- ❑ GitHub URL: [EDA with Data Visualization]

EDA with SQL

- Load the SpaceX dataset into a SQL database
- Apply EDA with SQL to get insight from the data.
- Get queries to perform on the dataset
- GitHub URL: EDA with SQL

Build an Interactive Map with Folium

- ❑ Folium Markers were used to show the Space X launch sites and their nearest important landmarks like railways, highways and cities
- ❑ Polylines were used to connect the launch sites to their nearest land marks.
- ❑ We build an interactive map with folium:
  - ❖ Markers of all Launch Sites
  - ❖ Coloured Markers of the launch outcomes for each Launch Site
  - ❖ Distances between a Launch Site to its proximities:
- ❑GitHub URL: [Build an Interactive Map with Folium]

Build a Dashboard with Plotly Dash

- ❑ We Build a Dashboard with Plotly Dash:
  - ❖Launch Sites Dropdown List
  - ❖Pie Chart showing Success Launches (All Sites/Specific Site)
  - ❖Slider of Payload Mass Range
  - ❖Scatter Chart of Payload Mass vs. Success Rate by Booster Versions
- ❑ GitHub URL: [Build a Dashboard with Plotly Dash]()

Predictive Analysis (Classification)

- Load the data using numpy and pandas, transform and split into training and testing data
- Build machine learning models and tune hyperparameters using GridSearchCV.
- Use accuracy as the metric for our model, improve the model using feature engineering and algorithm tuning.
- Find the best performing classification model
- GitHub URL: [Predictive Analysis]

Results

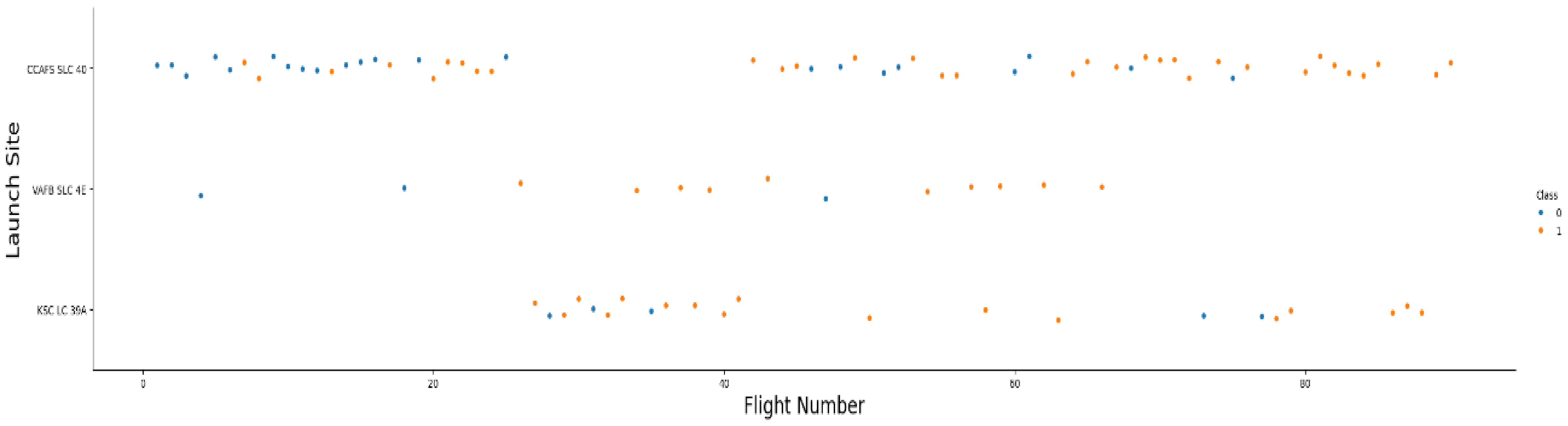
- The exploratory data analysis revealed that the success rate of the Falcon 9 landings was 66.67%
- The predictive analysis showed Logistic Regression, SVM, Decision tree and KNN algorithms with similar accuracy 83,33 %

Section 2

Insights drawn from EDA

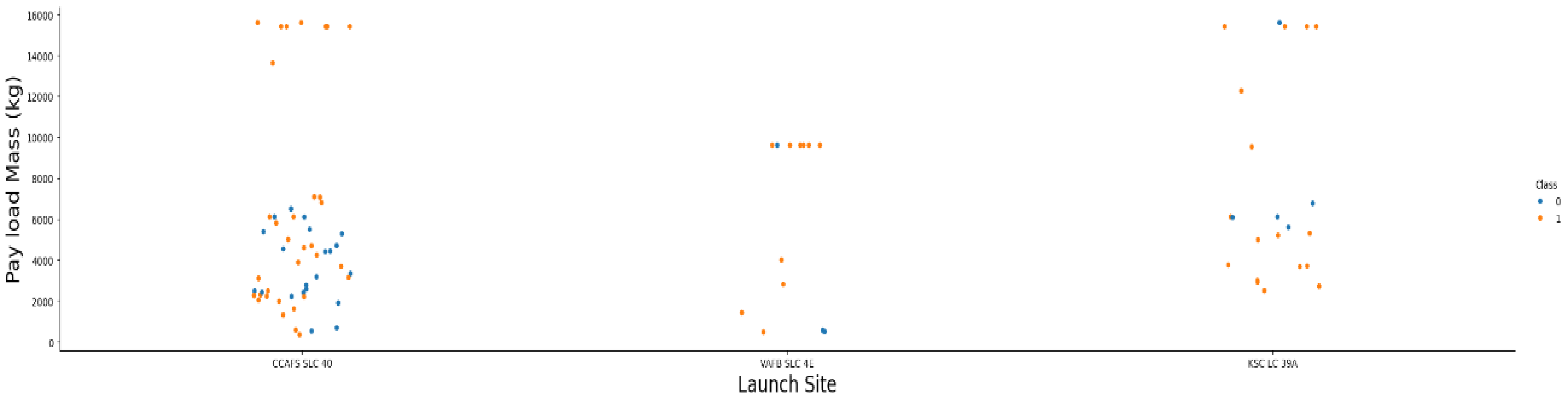
Flight Number vs. Launch Site

- ❑ We found that the larger the flight amount at a launch site, the greater the success rate at a launch site explanations
- ❑ There seems to be an increase in successful flights after the 40th launch



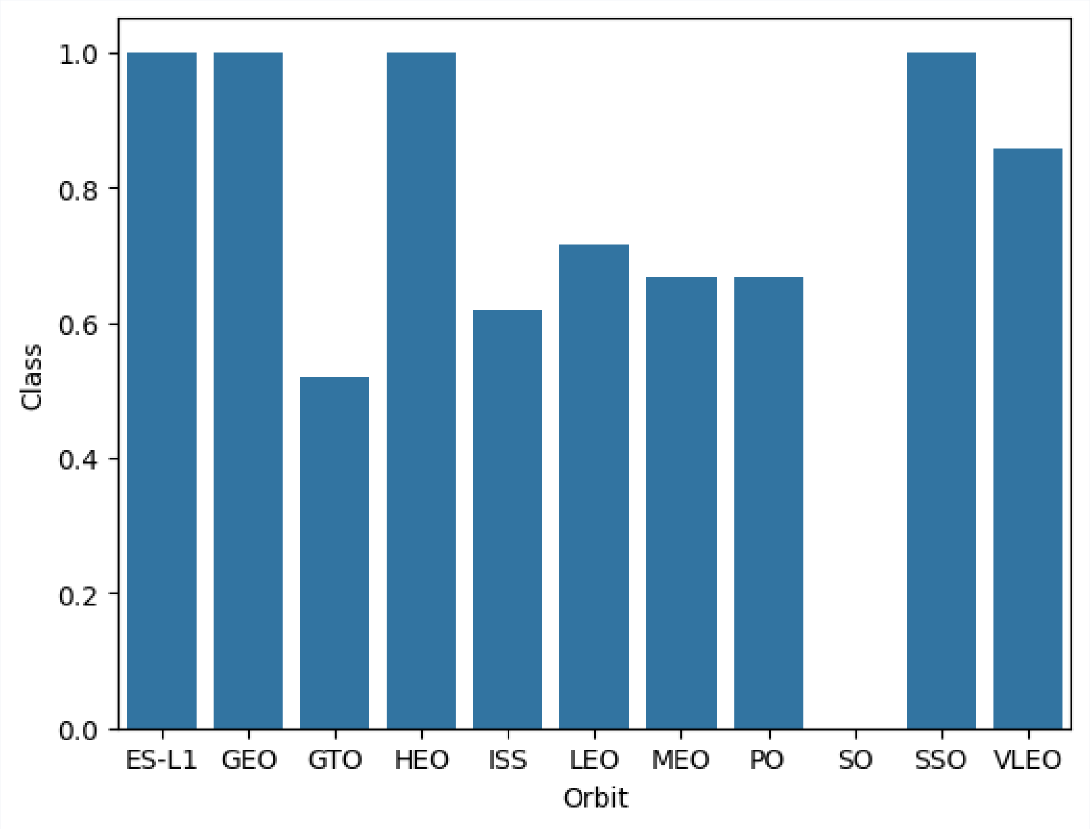
Payload vs. Launch Site

- ❑ Most of the launches with payload mass over 7000 kg were successful.
- ❑ KSC LC 39A has a 100% success rate for payload mass under 5500 kg too



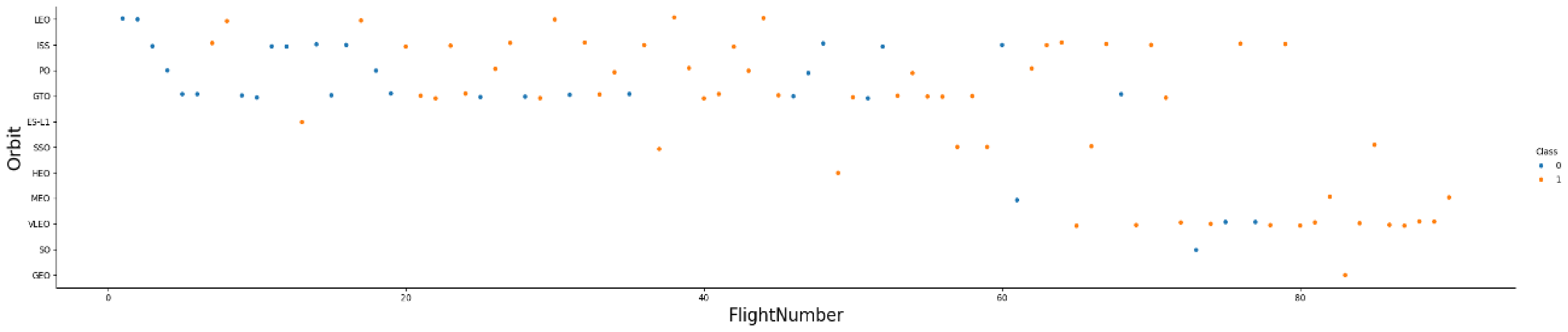
Success Rate vs. Orbit Type

- ES-L1, GEO, HEO, SSO, VLEO had the most success rate.
- SO with 0% success rate



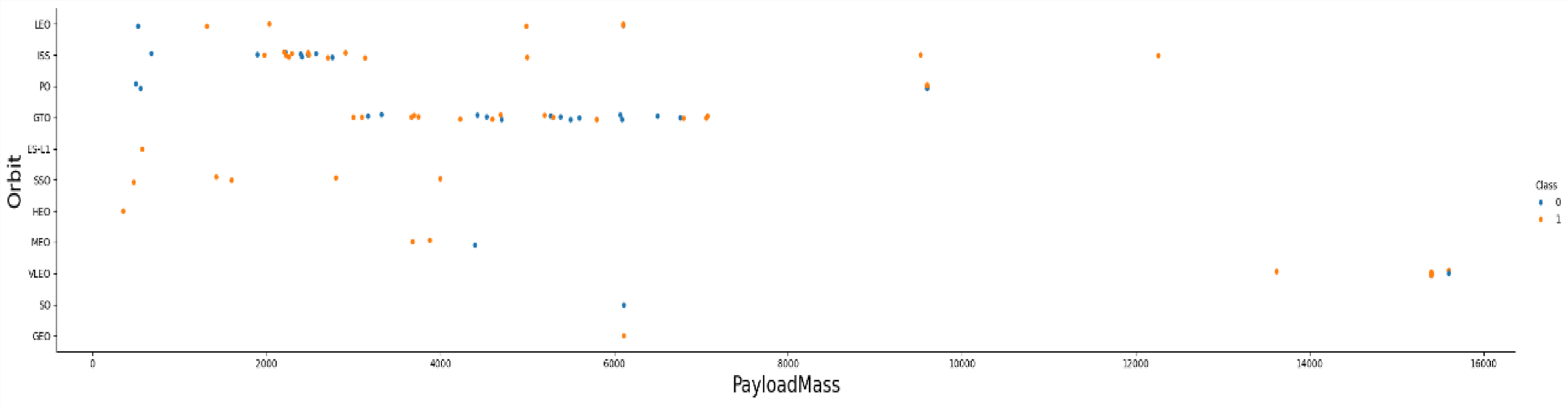
Flight Number vs. Orbit Type

- ❑ In GTO orbit, there is no relationship between flight number and the orbit.
- ❑ In LEO orbit, success is related to the number of flights



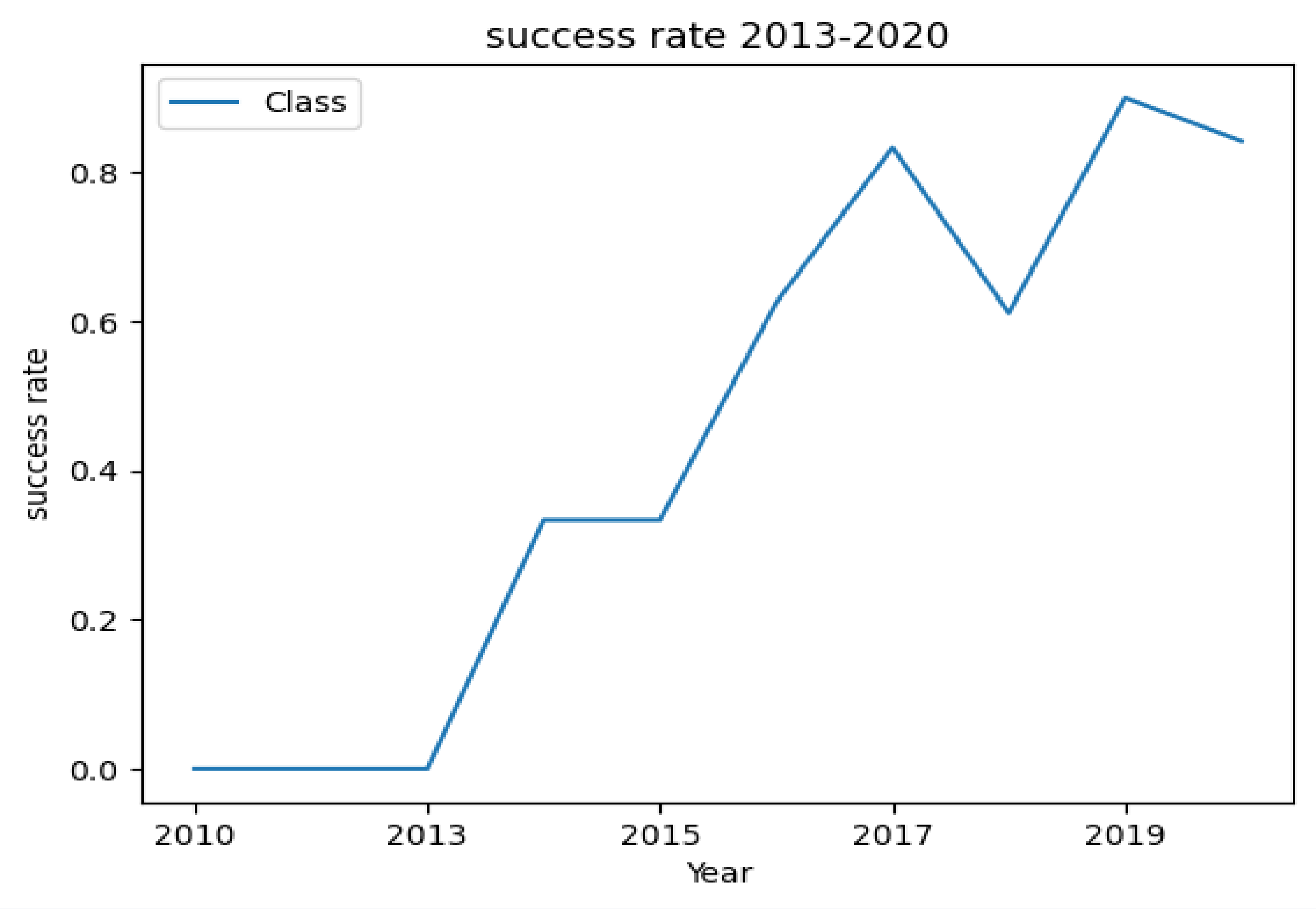
Payload vs. Orbit Type

- ❑ With heavy payloads, the successful landing is better for IISS, PO and LEO orbits.



Launch Success Yearly Trend

- ❑ Success rate since 2013 go increasing till 2019
- ❑ There is a gap in 2018 and in 2020.



All Launch Site Names

- ❑ SELECT DISTINCT return only the unique rows from the *launch_site* column

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

```
%sql select DISTINCT Launch_Site from SPACEXTABLE
```

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

Launch Site Names Begin with 'CCA'

- ❑ SELECT with LIMIT 5 find 5 records where launch sites begin with `CCA`

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

In [11]:

```
task_2 = '''
        SELECT *
        FROM SpaceX
        WHERE LaunchSite LIKE 'CCA%'
        LIMIT 5
        '''
create_pandas_df(task_2, database=conn)
```

Out[11]:

	date	time	boosterversion	launchsite	payload	payloadmasskg	orbit	customer	missionoutcome	landingoutcome
0	2010-04-06	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
1	2010-08-12	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of...	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2	2012-05-22	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
3	2012-08-10	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
4	2013-01-03	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

- With SELECT SUM clause the total payload results 45.596 kg.

```
%sql select sum(payload_mass__kg_) as total_payload_mass from SPACEXDATASET where customer = 'NASA (CRS)';
```

```
 * ibm_db_sa://wzf08322:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90108kqblod8lcg.databases.appdomain.cloud:31198/bludb
Done.
```

total_payload_mass
45596

Average Payload Mass by F9 v1.1

- With AVG function the average payload Mass from Booster Version F9 v1.1 results 2928.4 kg.

```
%sql select Booster_Version, avg(PAYLOAD_MASS__KG_) /
from SPACEXTABLE where Booster_Version ='F9 v1.1'
```

Booster_Version	avg(PAYLOAD_MASS__KG_)
F9 v1.1	2928.4

First Successful Ground Landing Date

- With min function, first successful ground landing date is 2015-12-22

```
%sql select min(Date) from SPACEXTABLE \
where Landing_Outcome ='Success (ground pad)'
```

min(Date)
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- SELECT with WHERE clause y with ‘AND’ condition clause

```
%sql select Booster_Version from SPACEXTABLE
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

Total Number of Successful and Failure Mission Outcomes

- With COUNT() function is used to count the number of occurences of different mission and filter for seven first positions

```
%sql select substr(Mission_Outcome,1,7) as Outcome, \
count(*) from SPACEXTABLE  group by 1
```

Outcome	count(*)
Failure	1
Success	100

Boosters Carried Maximum Payload

```
%sql select distinct Booster_Version from SPACEXTABLE \
where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) \
from SPACEXTABLE)
```

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

- With a subquery to get the Boosters Carried Maximum Payload

2015 Launch Records

```
#%sql select Date as month, \
if (Landing_Outcome='Failure (drone ship)',1,0) as month, \
Booster_Version, Launch_Site from SPACEXTABLE

%sql select case substr(Date, 6,2) \
when '01' then 'Enero' \
when '04' then 'Abril' \
end as month, \
Landing_Outcome, \
Booster_Version, Launch_Site FROM SPACEXTABLE  WHERE substr(Date,0,5)='2015'  \
and Landing_Outcome='Failure (drone ship)'
```

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

- With CASE and SUBSTR clauses

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

```
%sql select Landing_Outcome, count(*) from SPACEXTABLE
where Date between '2010-06-04' and '2017-03-20' group by
Landing_Outcome order by 2 desc
```

Landing_Outcome	count(*)
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2

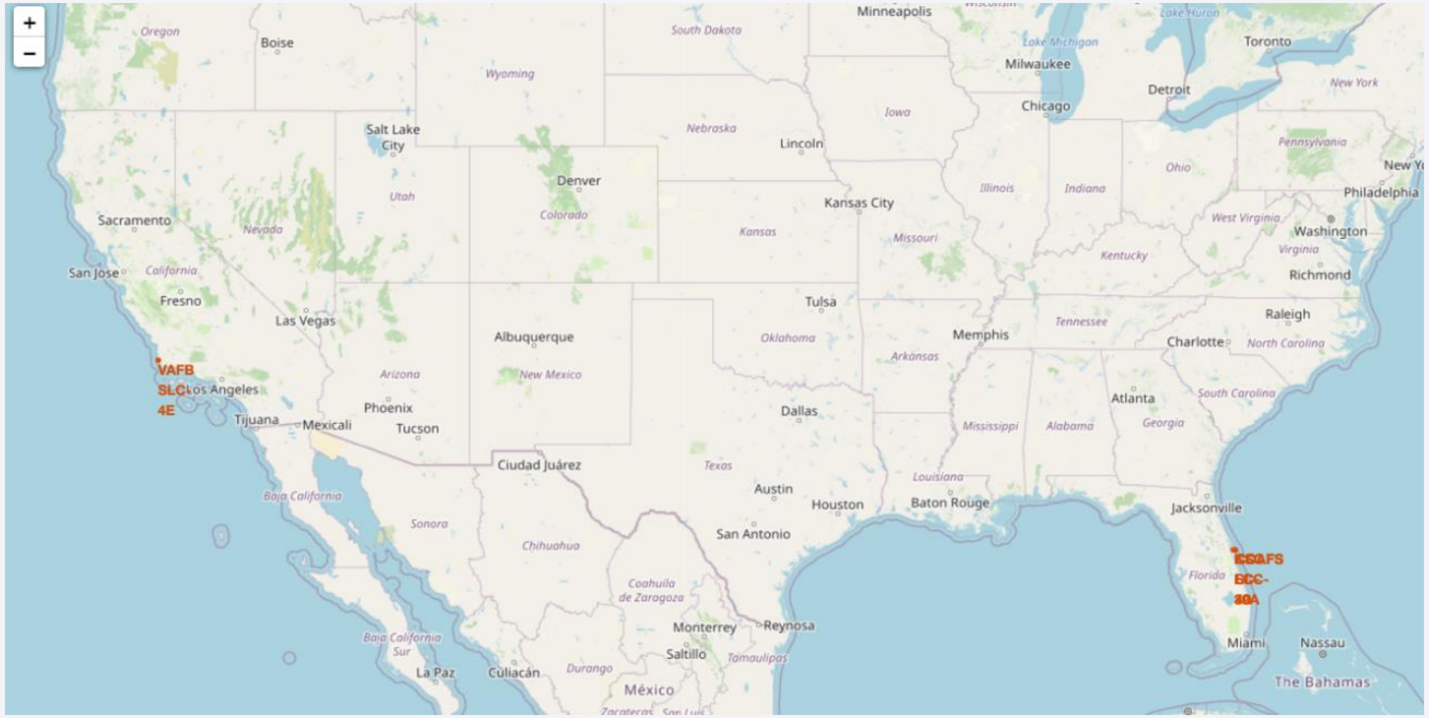
- With COUNT, BETWEEN GROUP BY and ORDER BY clauses

Section 3

Launch Sites Proximities Analysis

<Folium Map Screenshot 1>

The generated map with marked launch sites



<Folium Map Screenshot 2>

- From the color-labeled markers in marker clusters, we be able to easily identify which launch sites have relatively high success rates

<Folium Map Screenshot 3>

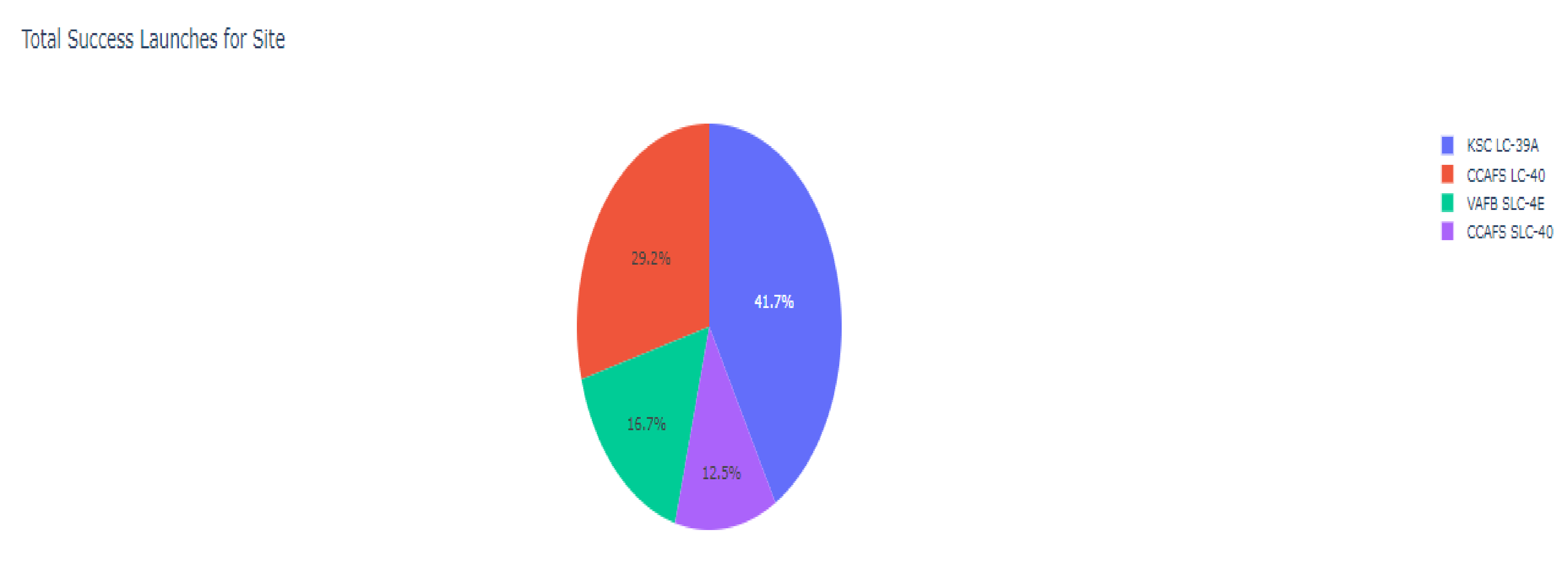
- Draw a line between a launch site to its closest city, railway, highway

Section 4

Build a Dashboard with Plotly Dash

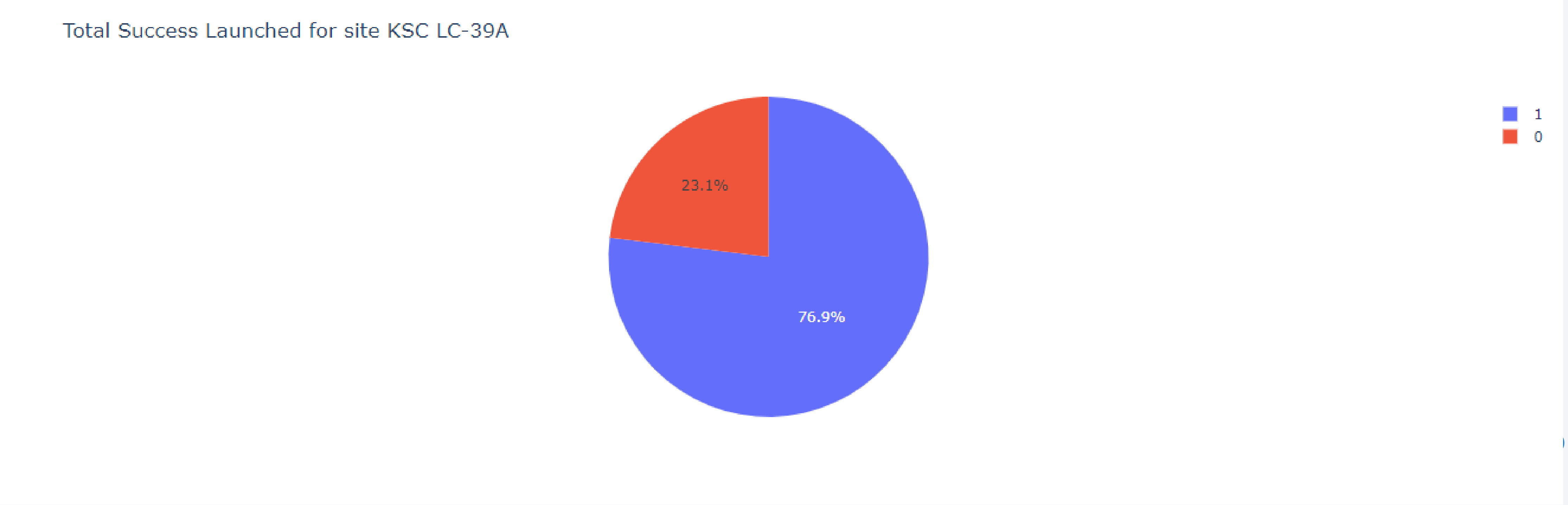
<Dashboard Screenshot 1>

- ❑KSC LC-39A Launch site has the most successful launchest
- ❑CCAFS SLC-40 Launch only 12,5 %



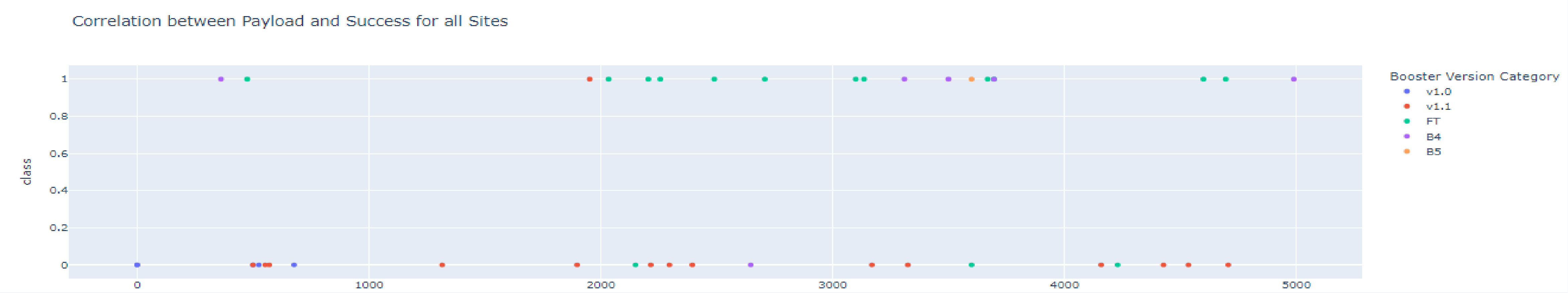
<Dashboard Screenshot 2>

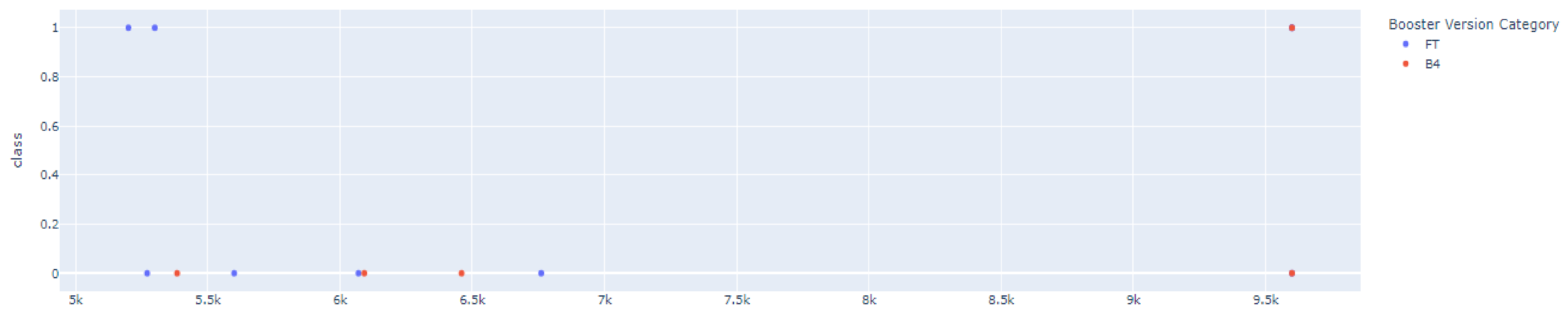
- ❑ KSC LC-39A with 76,9 % success has the highest success rate



<Dashboard Screenshot 3>

- The booster version v1.1 that has the largest success rate in both weight ranges



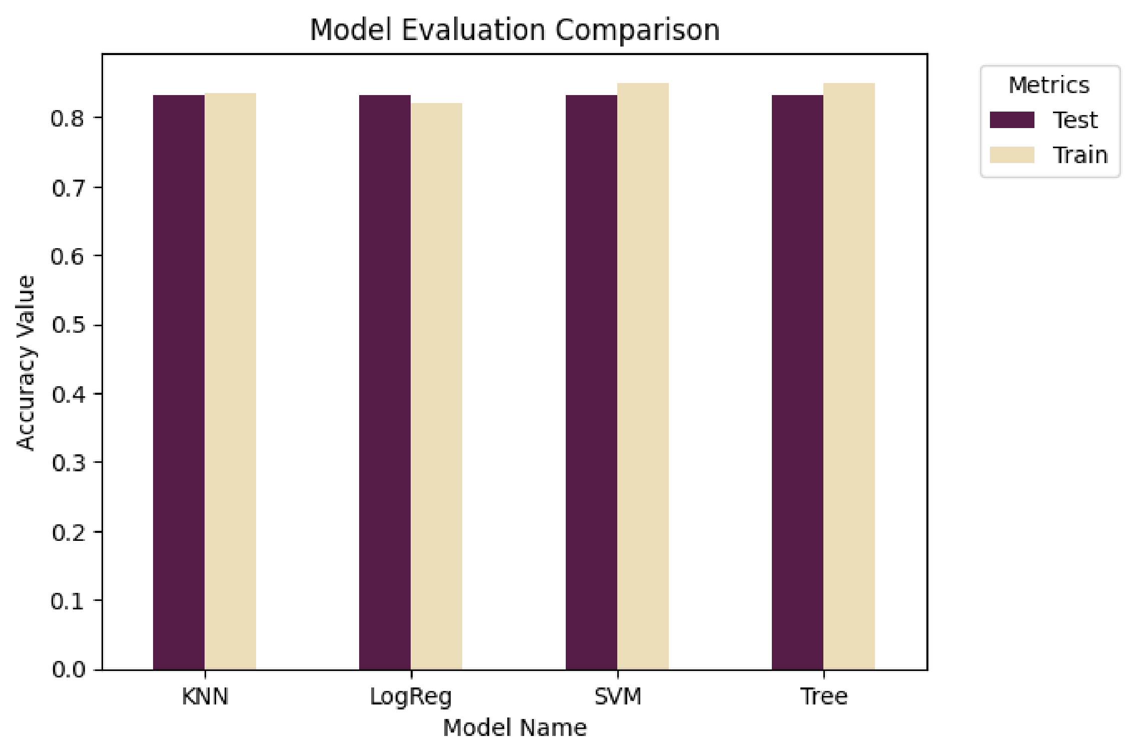


Section 5

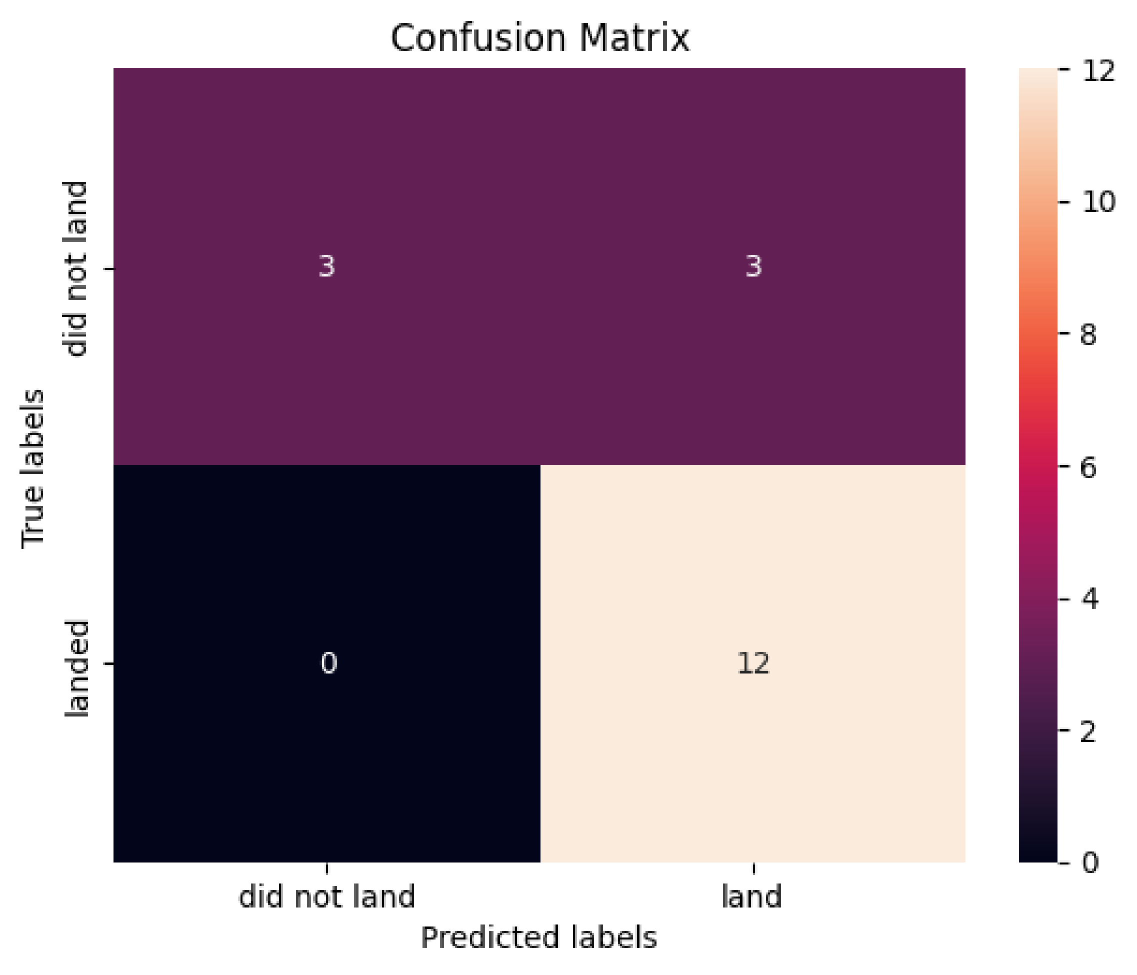
Predictive Analysis (Classification)

Classification Accuracy

The predictive analysis showed Logistic Regression, SVM, Decision tree and KNN algorithms with similar accuracy 83,33 %



Confusion Matrix



- 12 successful landings when the True label was successful (True Positive)
- 3 unsuccessful landings when the True label was failure (True Negative).
- The problem is 3 successful landings when the True label was unsuccessful landing (False Positive).

Conclusions

- Positive correlation between number of flights and success rate as the success rate has improved over the years
- Orbits SSO, GEO, ES-L1 and HEO have 100% success rate
- Launches with a low payload mass show better results than launches with a larger payload mass
- KSC LC-39A has the highest success rate of the launches
- The predictive analysis showed Logistic Regression, SVM, Decision tree and KNN algorithms with similar accuracy

Appendix

- Applied Data Science Capstone: https://www.coursera.org/learn/applied-data-science-capstone
- GIT hub repository: https://github.com/rcolombas/Proyecto-curso-Capstone

Thank you!