IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Andrey Kobelev
January 2025

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection with API
  - Data Collection with Web Scraping
  - Data Wrangling
  - Exploratory Data Analysis with SQL
  - Exploratory Data Analysis with Data Visualization
  - Interactive Visual Analytics with Folium
  - Machine Learning Prediction
- Summary of all results
  - Exploratory Data Analysis result
  - Interactive analytics in screenshots
  - Predictive Analytics result

Introduction

- Project background and context

  Falcon 9 launch costs of about $62 millions, while other providers cost exceeding $165 million. The Faclon 9 cost advantage stemmed from the reuse of the first stage. Singling out the factors of a successful landing of the first stage, one can save millions by making more launches successful in landing the first stage and re-use it.

  This goal of the project is to built a machine learning, modelling the first stage landing successfully.

- Problems you want to find answers
  - Factors impact the rocket successfully landing
  - How to increase successful landing rate?
  - Operating conditions to ensure a successful landing program

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was collected from the SpaceX REST API and by scraping wiki pages
- Perform data wrangling
  - Raw data had JSON object and HTML tables formats. To perform the visualization and analysis the data was transformed into pandas dataframe.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Machine Learning model was build to determine the conditions for the first stage of Falcon 9 to land successfully

Data Collection

Two methods for Data collection were used:

- get request to the SpaceX API

  followed with decoding the response content as a Json using .json() function call and converting it into a pandas dataframe with .json_normalize(). The ata was cleaned, checked for missing values and filled in missing values as necessary

- web scraping from Wikipedia with BeautifulSoup

  followed with extracting the launch records as HTML table, parsing the table and converting the data into a pandas dataframe for future analysis.

Data Collection – SpaceX API

- Get request was used to collect the data. The data was cleaned and formatted.

- GitHub link: https://github.com/AndreyKobelev/IBM-Data-Science-Capstone-SpaceX/blob/main/1.jupyter-labs-spacex-data-collection-api.ipynb

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:

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

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

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

```
response.status_code
```

```
200
```

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

Data Collection - Scraping

- To scrape the Falcon 9 launch records from wiki, we used BeatifulSoup.
- We parsed the response and converted into the dataframe for the analysis and visualization.
- GitHub URL: https://github.com/AndreyKobelev/IBM-Data-Science-Capstone-SpaceX/blob/main/2.%20jupyter-labs-webscraping.ipynb

```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"
```

Next, request the HTML page from the above URL and get a `response` object

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

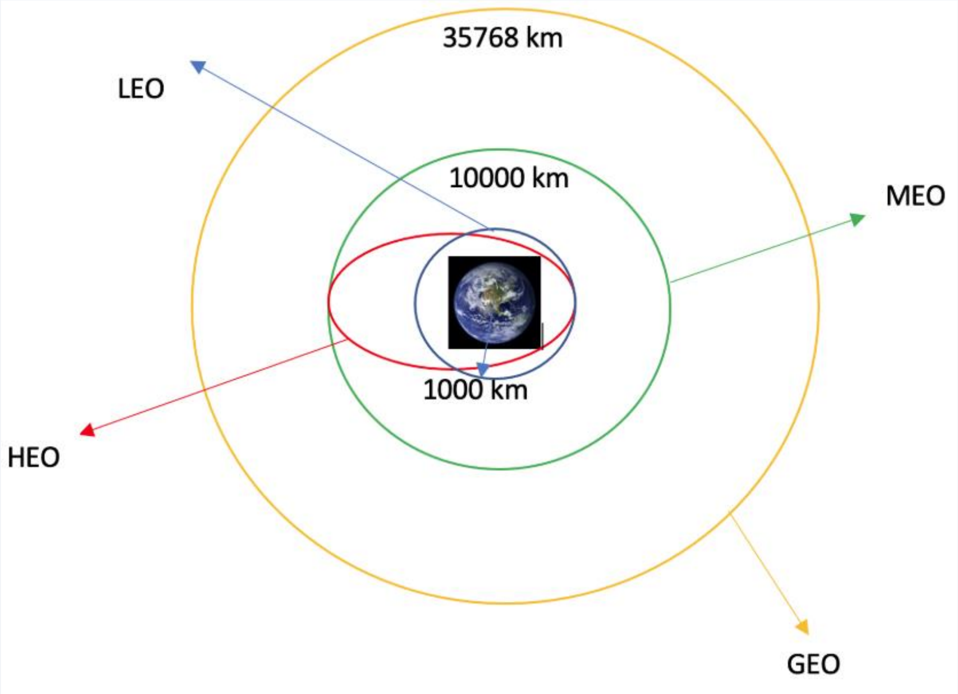
First, let's perform an HTTP GET method to request the Falcon9 Launch HTML page, as an HTTP response.

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

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

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

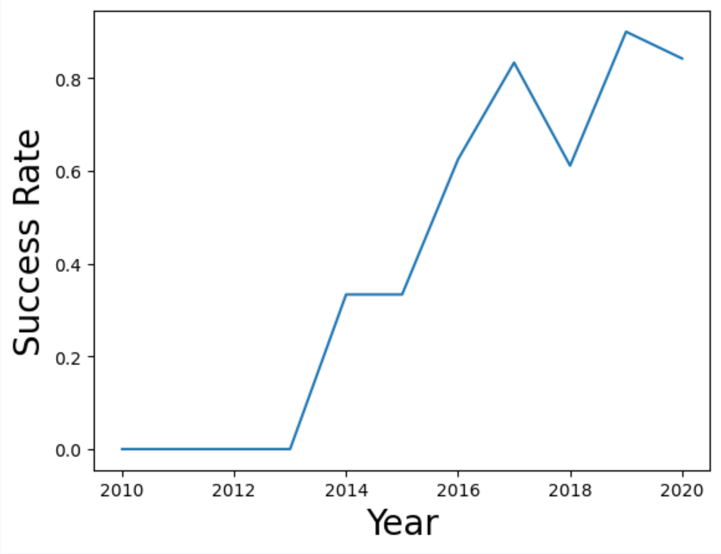
Data Wrangling

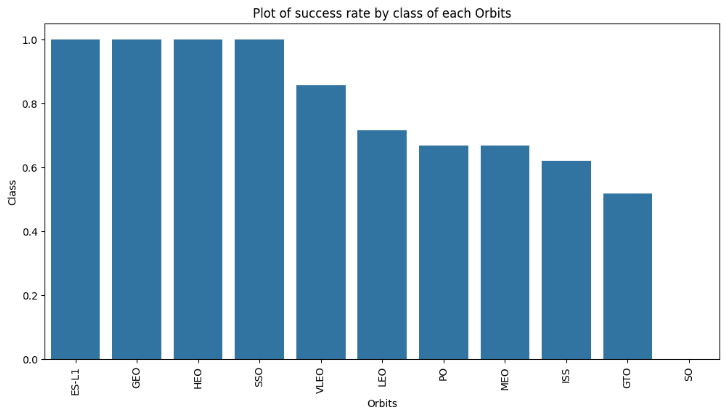


- Exploratory Data Analysis (EDA) was performed to determine the training labels.
- The number of launches at each site and occurrence of each orbits were calculated
- GitHub URL: https://github.com/AndreyKobelev/IBM-Data-Science-Capstone-SpaceX/blob/main/3.%20labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

- To perform EDA we visualized the relations among flight numbers and launch sites, payload and launch sites, success rate for each orbit type, flight number and orbit type, the launch success yearly trend
- GitHub URL: https://github.com/AndreyKobelev/IBM-Data-Science-Capstone-SpaceX/blob/main/5.%20edadataviz.ipynb





EDA with SQL

- SpaceX dataset was loaded into a SQLite database. The sql queries were made to extract:
  - The names of unique launch sites
  - The total payload mass
  - The average payload mas
  - The total number of successful and failure outcomes
  - The failed landing outcomes
- GitHub URL: https://github.com/AndreyKobelev/IBM-Data-Science-Capstone-SpaceX/blob/main/4.%20jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- Marked launch sites and added map objects (markers, circles, lines etc) for each site on the folium map
- the feature launch outcomes were assigned as 0 (failure) and 1 (success)
- with the color-labeled marker clusters, we identified the launch sites with high success rates
- the distances between a launch site to its proximities were calculated. The proximity of the launch sites to cities, railways, highways and coastlines was evaluated.
- GitHub URL: https://github.com/AndreyKobelev/IBM-Data-Science-Capstone-SpaceX/blob/main/6.%20lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

- Interactive dashboard was built using Plotly Dash
  - pie charts with the total launches by each site
  - scatter graph for the relationship between Outcome and Payload Mass for different booster version
- GitHub URL: https://github.com/AndreyKobelev/IBM-Data-Science-Capstone-SpaceX/blob/main/7.%20spacex_dash_app.py

Predictive Analysis (Classification)

- The data was loaded using numpy and pandas, transformed the data, andsplit the data into training and testing subsets.
- Different machine learning models were used with hyperparameters optimized using GridSearchCV.
- Accuracy of the model was assessed. Feature improvement and algorithm fine tuning were used to refine the models.
- We found the best performing classification model.
- GitHub URL: https://github.com/AndreyKobelev/IBM-Data-Science-Capstone-SpaceX/blob/main/8.%20SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results

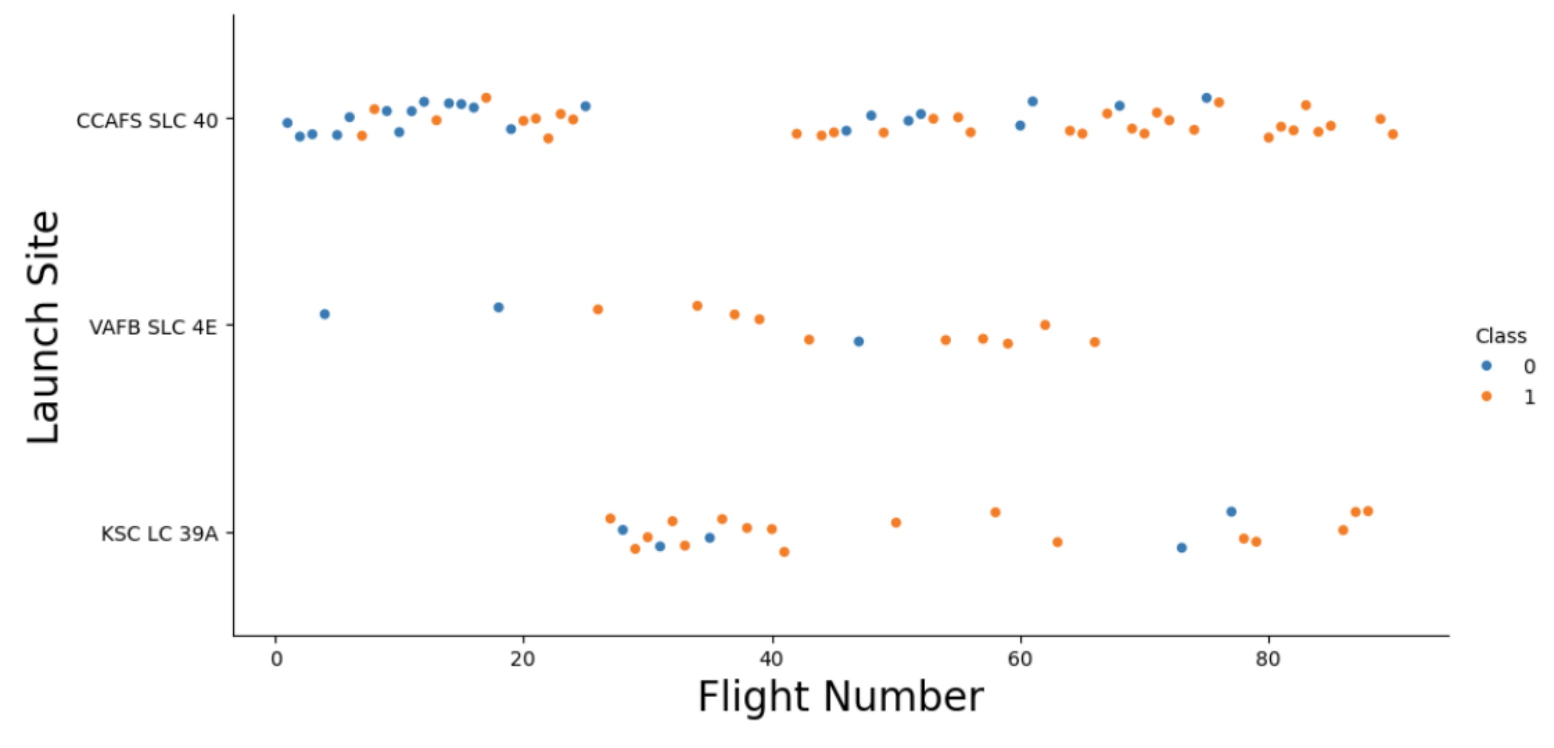
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

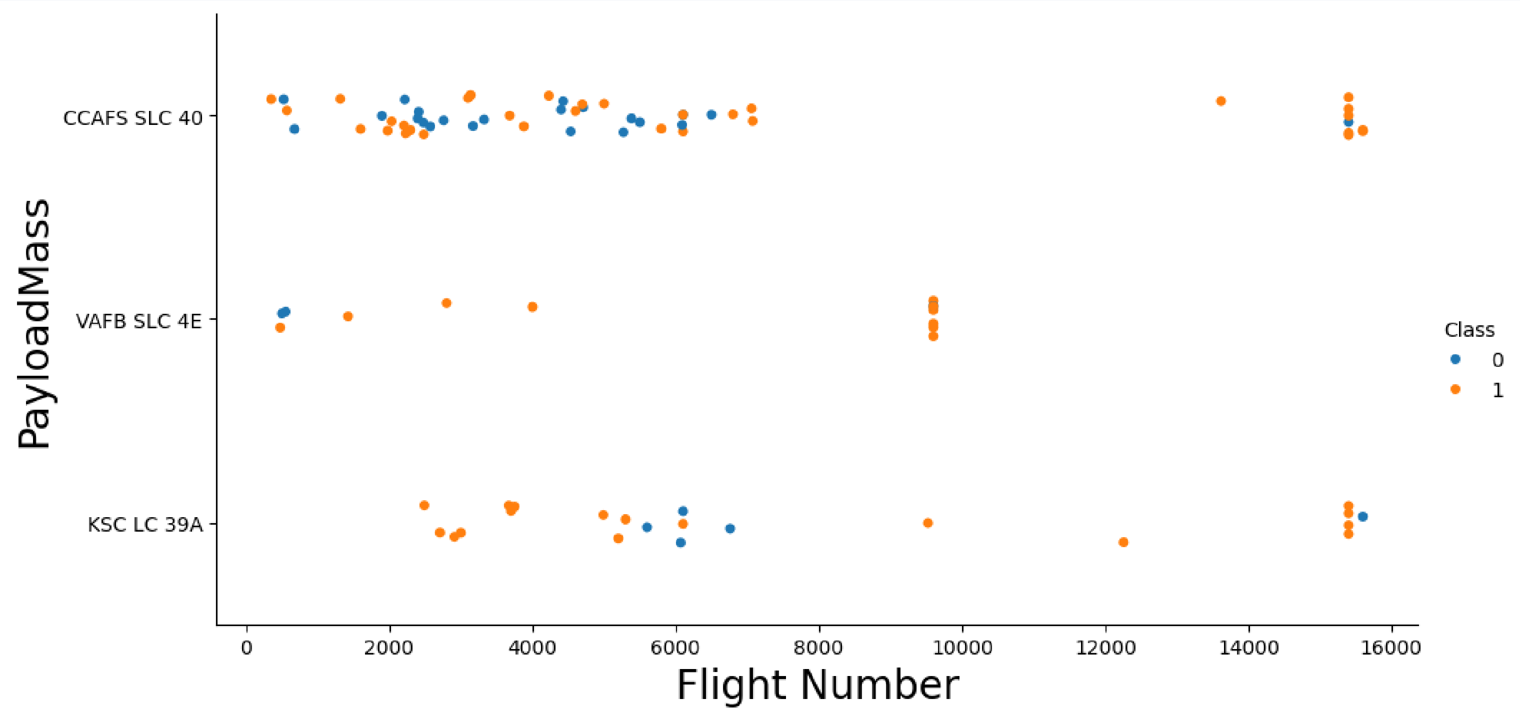
Flight Number vs. Launch Site

- Success rate follows the experience: the more number of landing attempt, the higher the success rate



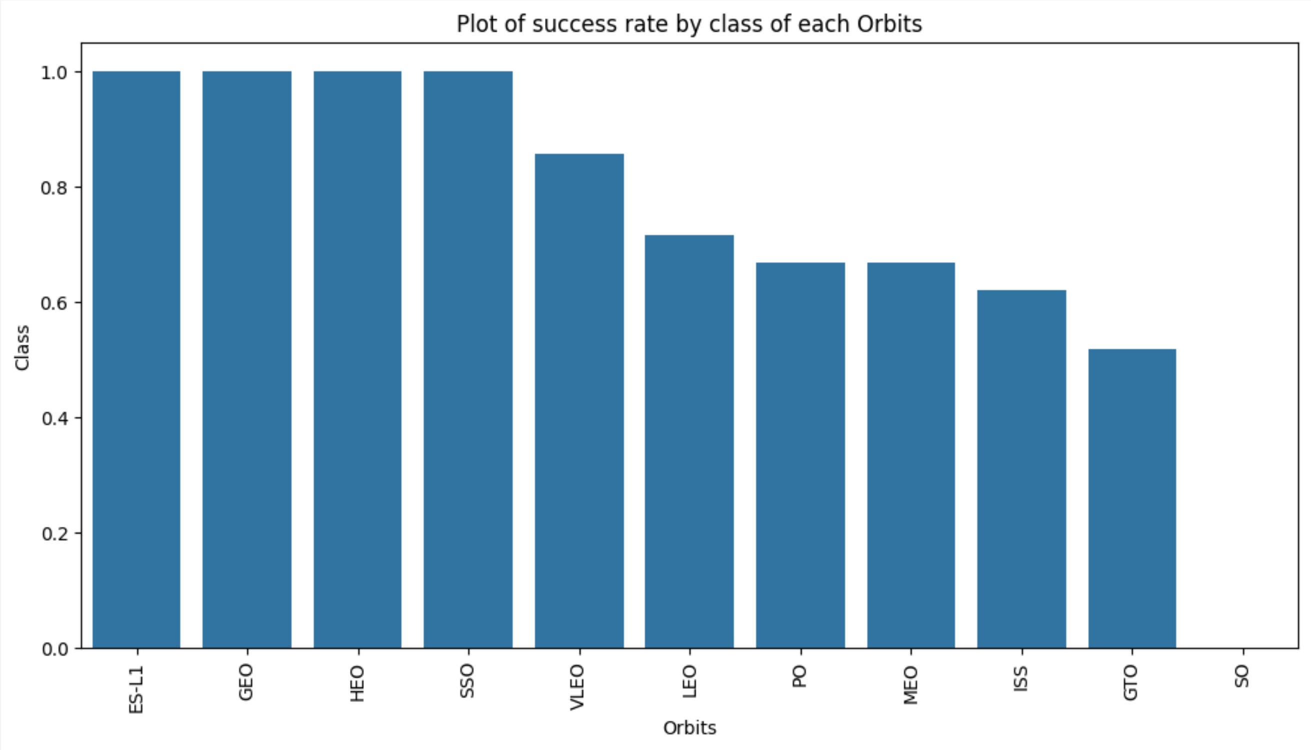
Payload vs. Launch Site

- Heavier payloads rockets are being tested in deeper details: the heavier payload, the higher the success rate. However, heavier payloads might be an indication of a specific types of orbits that are “easier” for the 1st stage landing.



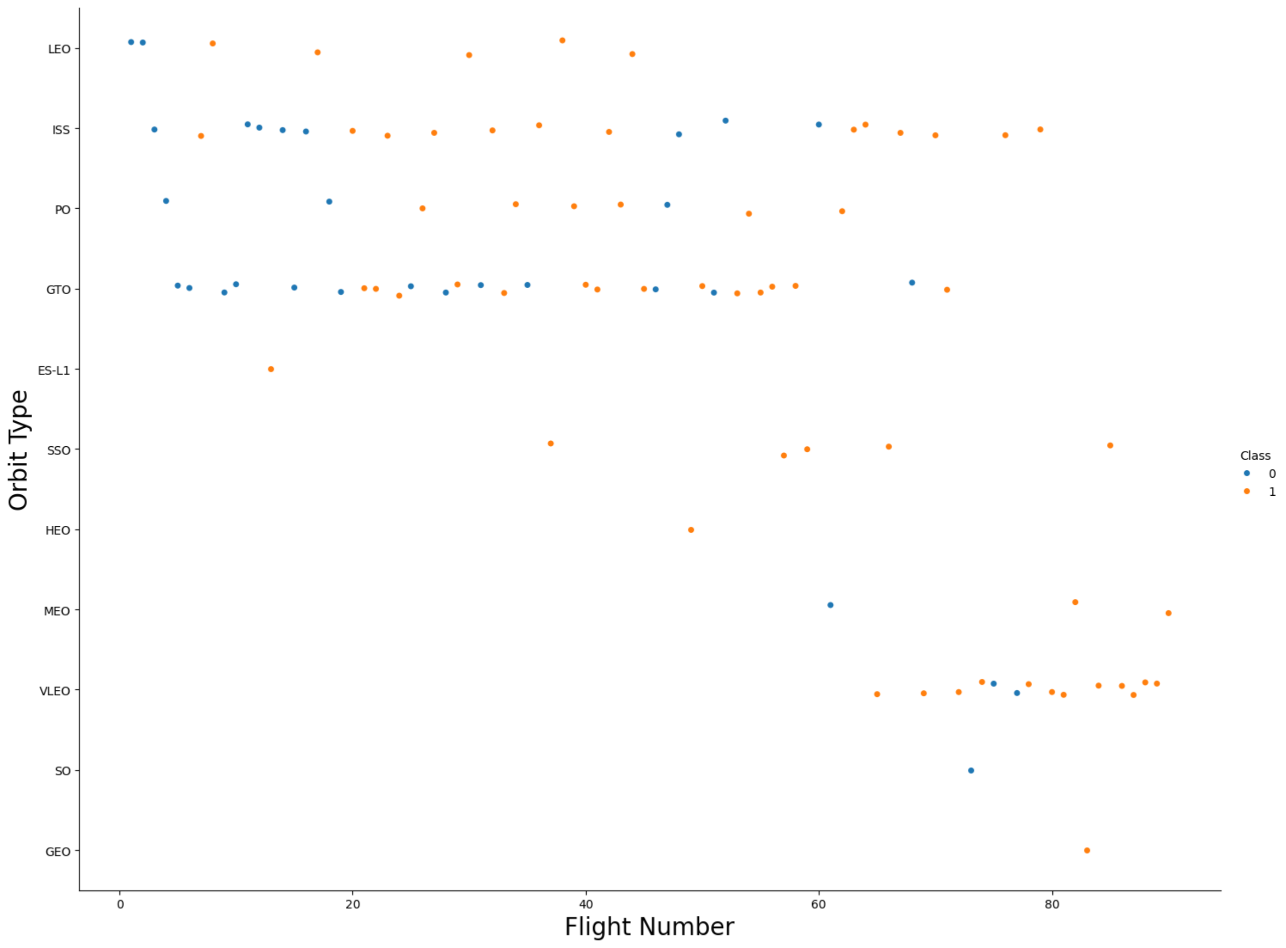
Success Rate vs. Orbit Type

- High altitude orbits and stationary ones have success stage 1 landing rate close to 100%.
- Lower altitude orbits or ones with more complex trajectory have 30-40% less success rate in landing of the first stage.



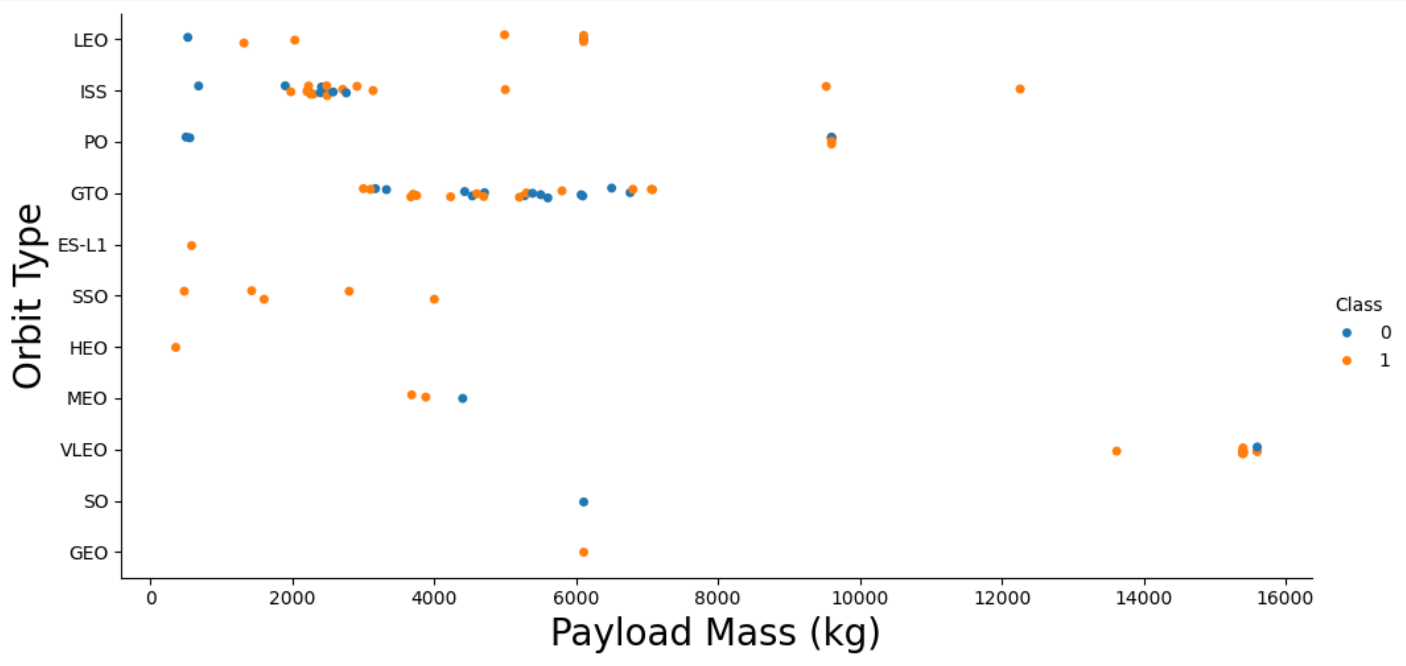
Flight Number vs. Orbit Type

- Learning curve does exist: the more flight number, the higher the success landing rate for any orbit



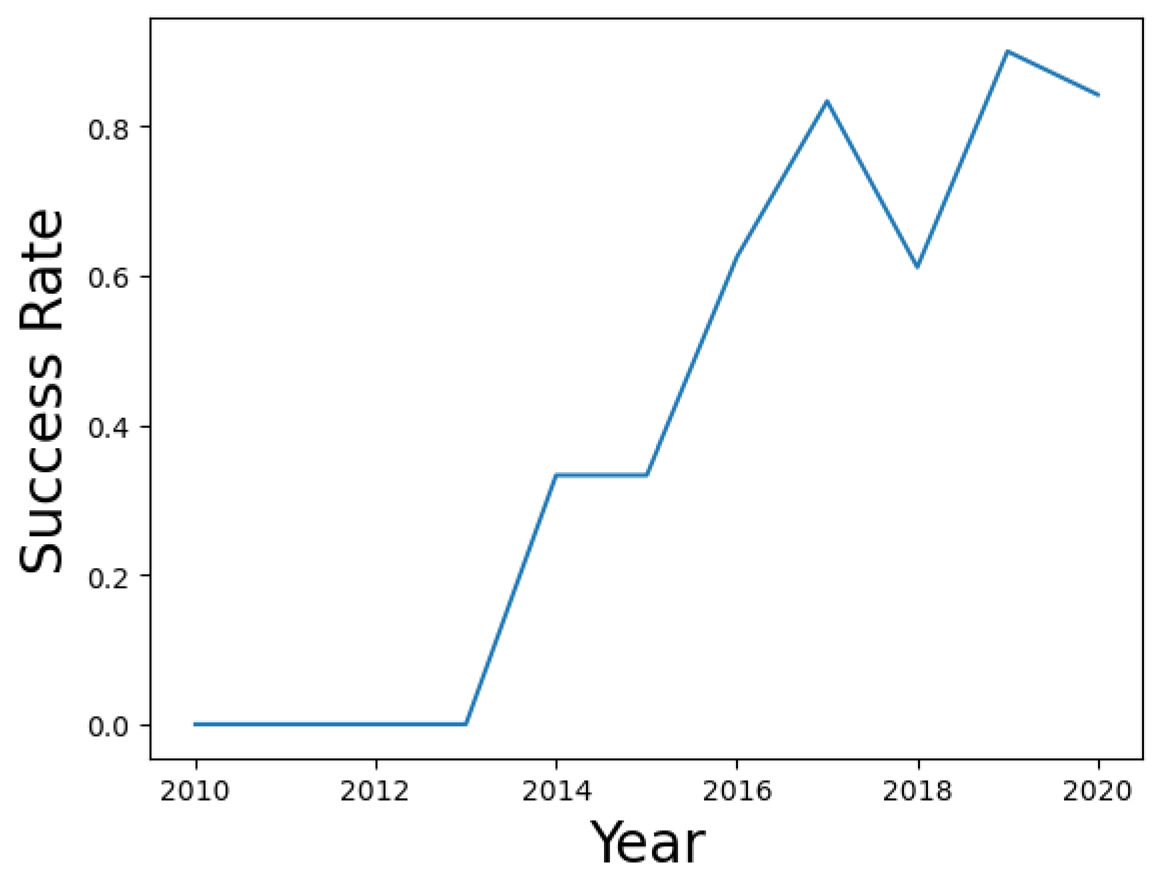
Payload vs. Orbit Type

- GTO is the most difficult for landing stage 1. Regardless of the payload, success rate does not converge to a pattern
- For LEO and ISS payload affects positively on the success rate of the tage 1 landing. Mechanism is yet unknown, perhaps higher scrutiny in pre-launch testing
- For the others, the orbit type seems to be more important than the payload. However, for many orbits there is not yet enough statistics.
-



Launch Success Yearly Trend

- Obviously, over the years the engineers have been steadily improving the technologies leading to higher success rate for the first stage landing.



All Launch Site Names

- SELECT **DISTINCT** query was used to extract unique names of the launch sites

Task 1

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

```
sql SELECT DISTINCT LAUNCH_SITE FROM SPACEXTBL ORDER BY 1
```

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

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

Launch Site Names Begin with 'CCA'

- **LIMIT** query was used to get exactly 5 data points on the launch sites ‘CCA’.

Task 2

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

In [12]:
```
sql SELECT * FROM SPACEXTBL WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5;
```

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

Out[12]:

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

Total Payload Mass

- SUM() function was used to extract the total payload mass directly from the DB without transforming the table to dataframe.
- Total payload mass is: 111 268 kg

Task 3

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

In [13]:
```
sql SELECT SUM(PAYLOAD_MASS__KG_) AS TOTAL_PAYLOAD FROM SPACEXTBL WHERE PAYLOAD  LIKE '%CRS%';
```

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

Out[13]:

TOTAL_PAYLOAD
111268

Average Payload Mass by F9 v1.1

- As previously we used AVG() SQL function to extract average value directly from the DB without need of using data frames and further processing in python.
- Average payload mass carried by booster version F9 v1.1 is 3 metric tons

Task 4

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

In [14]:

```
sql SELECT AVG(PAYLOAD_MASS__KG_) AS AVG_PAYLOAD FROM SPACEXTBL WHERE BOOSTER_VERSION = 'F9 v1.1';
```

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

Out[14]:

AVG_PAYLOAD
2928.4

First Successful Ground Landing Date

- To extract the first successful ground Landing Date we used MIN() function
- The first successful landing on the ground happened on Dec 22, 2015

Task 5

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

Hint:Use min function

In [16]:

```
sql SELECT MIN(DATE) AS FIRST_SUCCESS_GP FROM SPACEXTBL WHERE LANDING_OUTCOME = 'Success (ground pad)';
```

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

Out[16]:

FIRST_SUCCESS_GP
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- To list the names of boosters that have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000 we used:

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

In [18]:
```
sql SELECT DISTINCT BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000 AND LANDING_OUTCOME = 'Succ
```

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

Out[18]:

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

Total Number of Successful and Failure Mission Outcomes

- To calculate the total number of successful and failure mission outcomes:

Task 7

List the total number of successful and failure mission outcomes

In [19]:
```
sql SELECT MISSION_OUTCOME, COUNT(*) AS QTY FROM SPACEXTBL GROUP BY MISSION_OUTCOME ORDER BY MISSION_OUTCOME;
```

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

Out[19]:

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

Boosters Carried Maximum Payload

- To list the names the booster that have carried the maximum payload, a subquery in the WHERE clause and the MAX() function were used

Task 8

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

```
sql SELECT DISTINCT BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL)
```

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

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

2015 Launch Records

- To show the month names we had to create detailed code, since SQLLite does not support monthnames()
- WHERE, LIKE, AND and BETWEEN to filter the information were used

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

Date	Month_Name	Landing_Outcome	Booster_Version	Launch_Site
2015-01-10	January	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
2015-04-14	April	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

```
%%sql SELECT  Date, CASE
       WHEN SUBSTR(Date, 6, 2) = '01' THEN 'January'
       WHEN SUBSTR(Date, 6, 2) = '02' THEN 'February'
       WHEN SUBSTR(Date, 6, 2) = '03' THEN 'March'
       WHEN SUBSTR(Date, 6, 2) = '04' THEN 'April'
       WHEN SUBSTR(Date, 6, 2) = '05' THEN 'May'
       WHEN SUBSTR(Date, 6, 2) = '06' THEN 'June'
       WHEN SUBSTR(Date, 6, 2) = '07' THEN 'July'
       WHEN SUBSTR(Date, 6, 2) = '08' THEN 'August'
       WHEN SUBSTR(Date, 6, 2) = '09' THEN 'September'
       WHEN SUBSTR(Date, 6, 2) = '10' THEN 'October'
       WHEN SUBSTR(Date, 6, 2) = '11' THEN 'November'
       WHEN SUBSTR(Date, 6, 2) = '12' THEN 'December'
   END AS Month_Name,
   Landing_Outcome,   Booster_Version,  Launch_Site  FROM
SpaceXTBL  WHERE Landing_Outcome LIKE 'Failure (drone
ship)'   AND SUBSTR(Date, 1, 4) = '2015';
```

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

- COUNT, WHERE and BETWEEN were used to filter the required information
- Further, GROUP BY and ORDER BY clauses were applied to order grouped landing outcomes.

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.

```
sql SELECT LANDING_OUTCOME, COUNT(*) AS QTY FROM SPACEXTBL WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY LANDINC
```

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

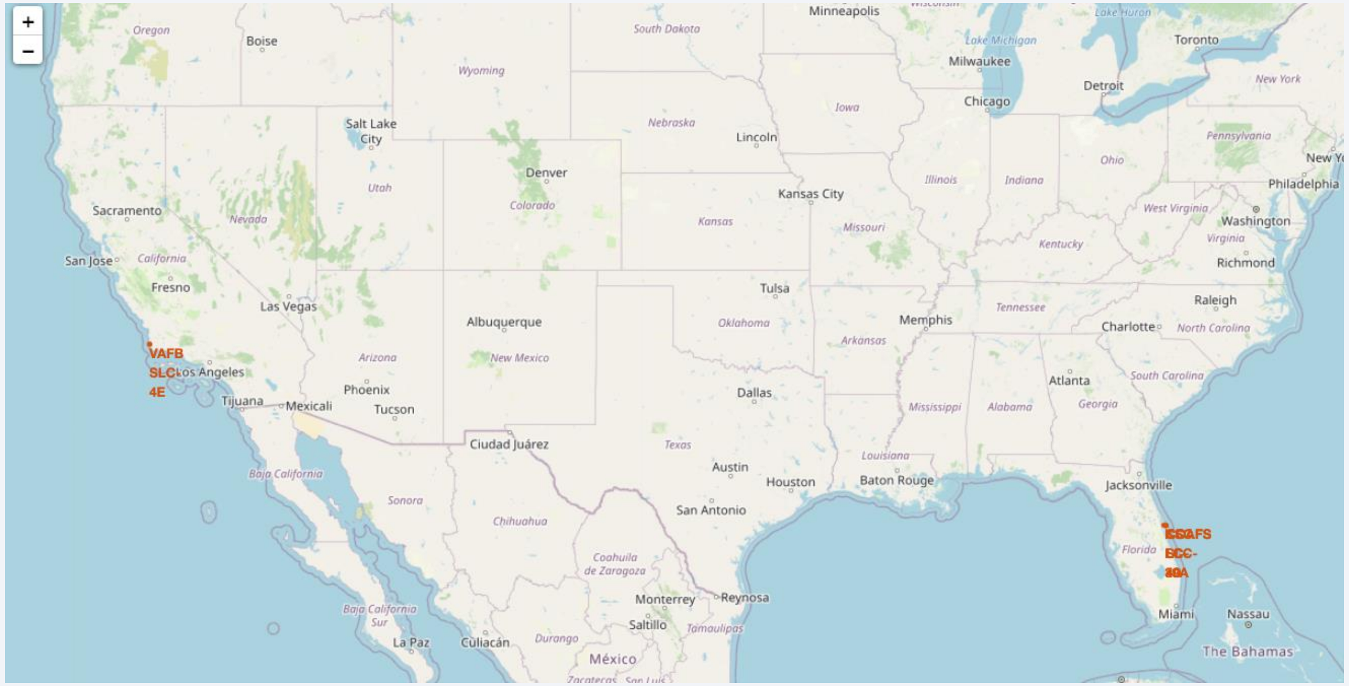
Landing_Outcome	QTY
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

Section 3

Launch Sites Proximities Analysis

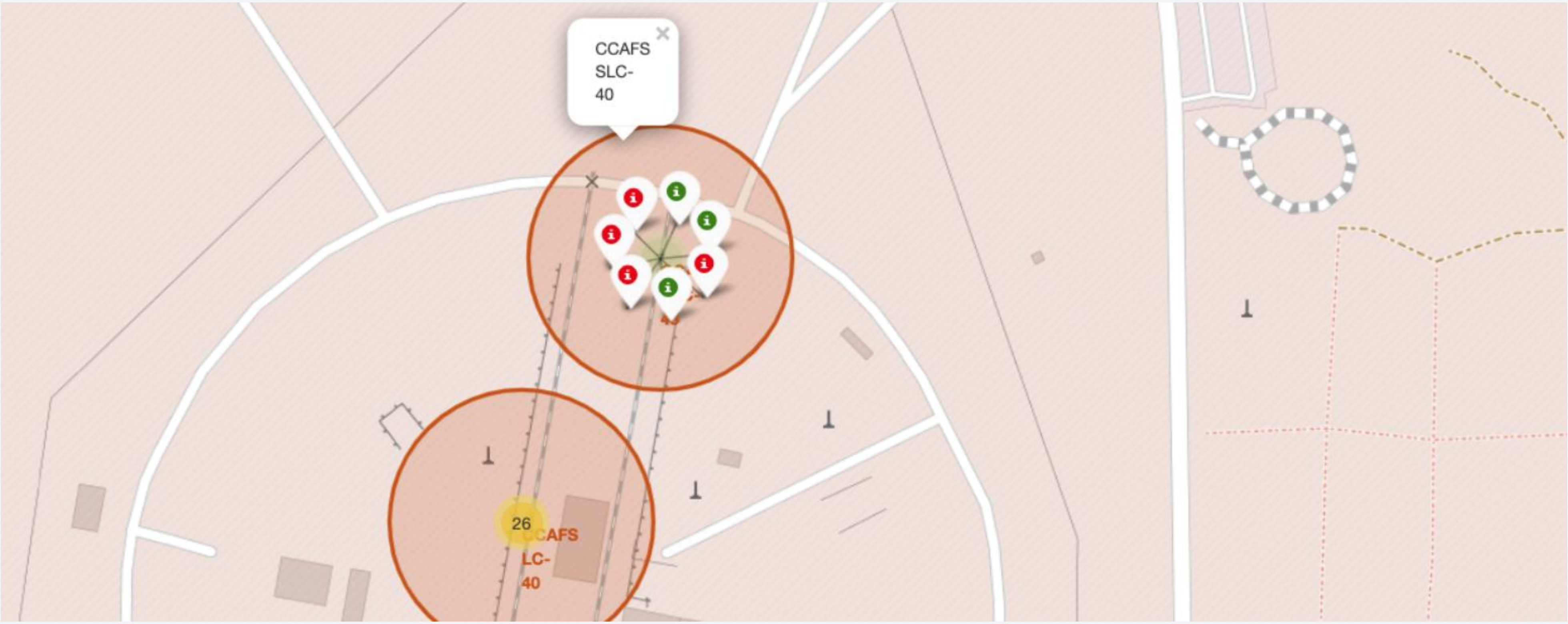
SpaceX Launch Sites

- All SpaceX Launch sites are in a very close proximity to the coast line and close to equator to partially balance the gravity and to cut fuel costs



Launch Sites with color-coded labels

- Green marker shows Success, while red label show Failures.



Launch Sites distance to landmarks

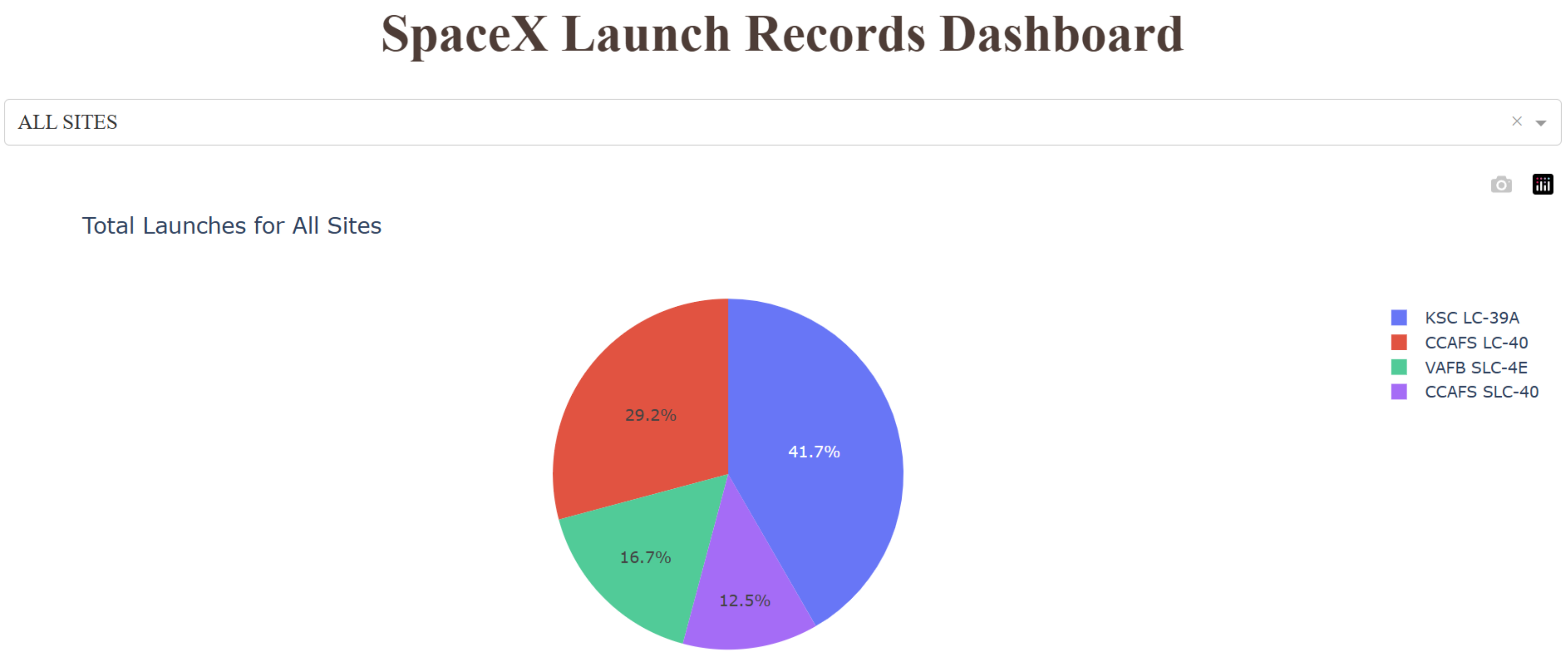
- Launch sites generally keep certain distance from cities but very close to the coastlines, None of the Launch sites are in close proximity to highways or railways

Section 4

Build a Dashboard with Plotly Dash

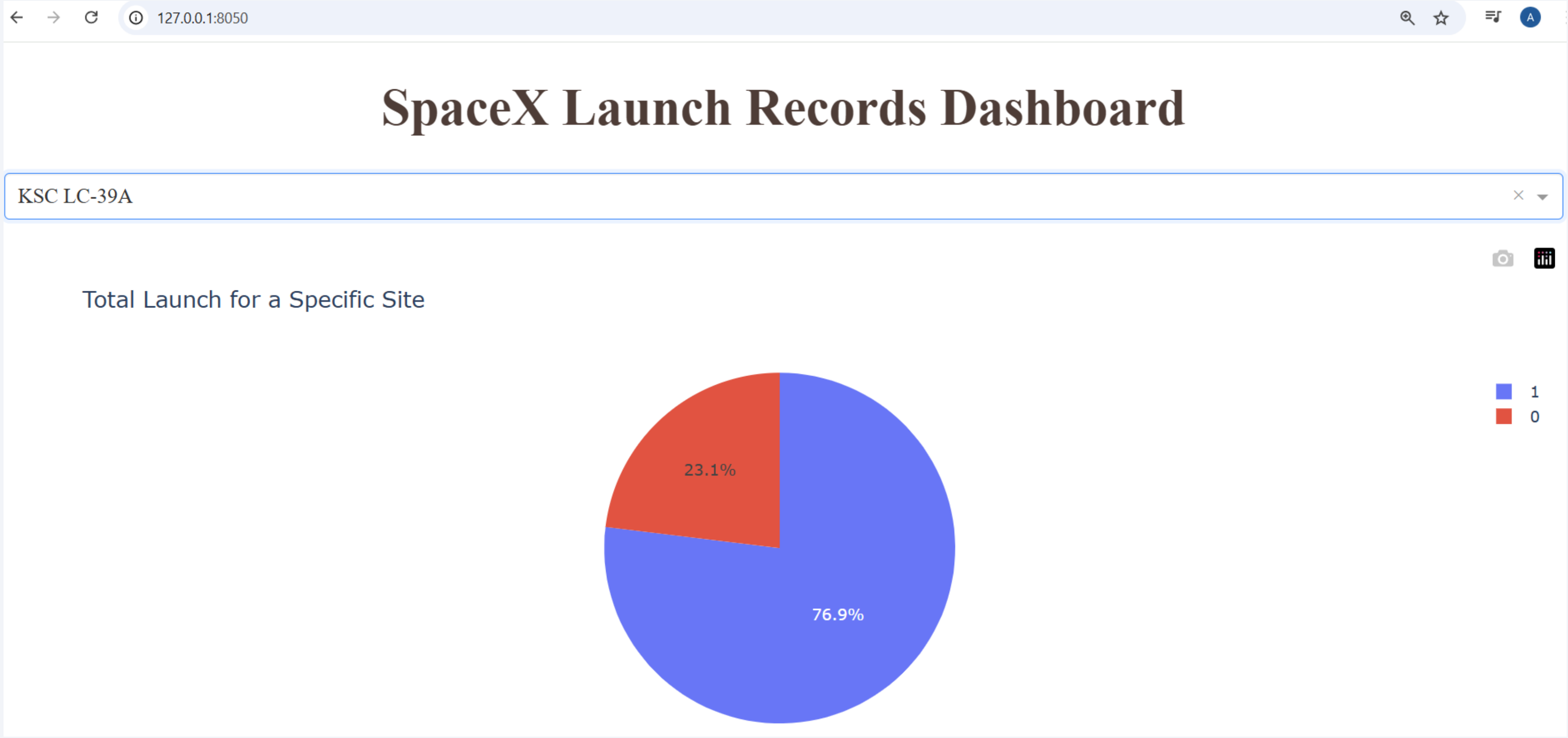
Success rate by the Launch sites

- More of 70% of all successful landings are happened on two sites: KSC LA-39A and CCAFS LC-40



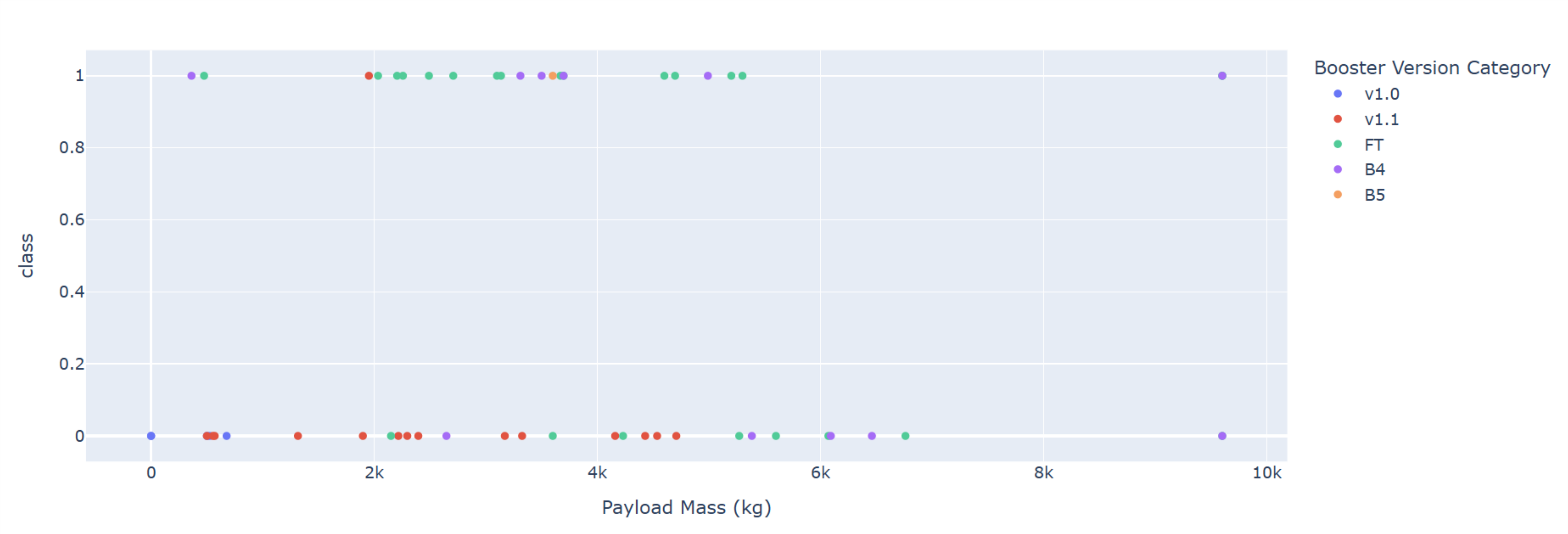
KSC LC-39A 3 landings out of 4 successful!

- KSC LC-39A has the highest success rate of landing the first stage, representing about 77%. Blue color shows success, while red shows failure to land the first stage



All sites cumulative Payload vs. Launch Outcome

- For different Booster Version Category the success rate vary

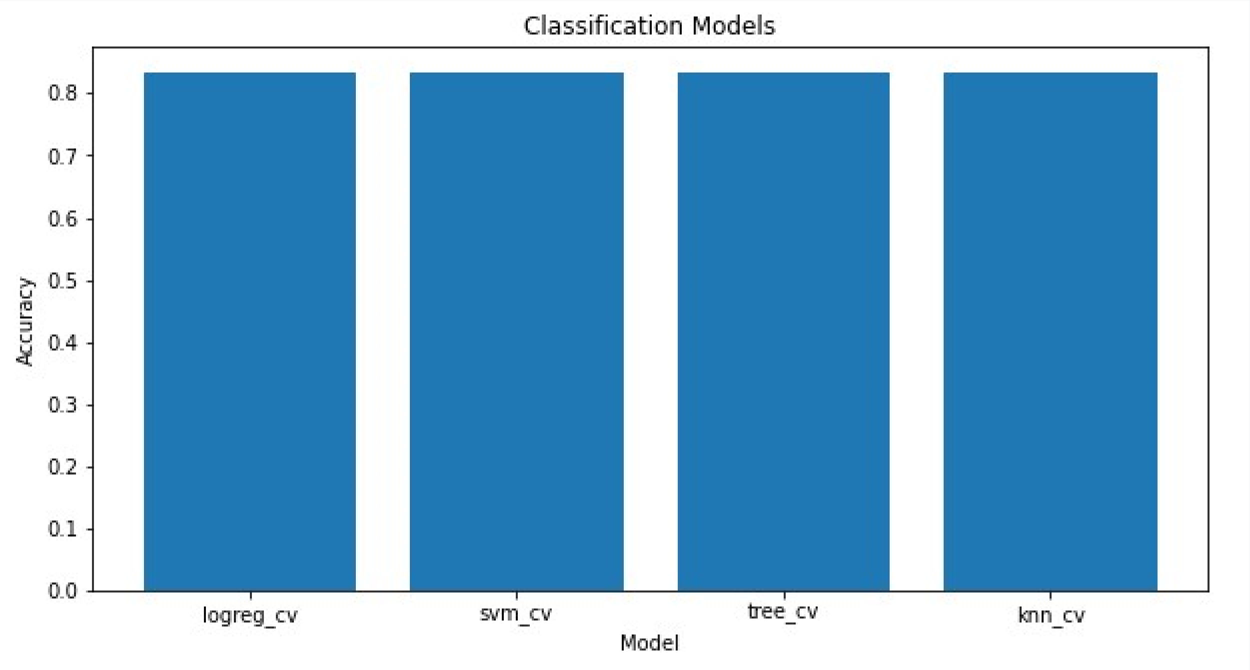


Section 5

Predictive Analysis (Classification)

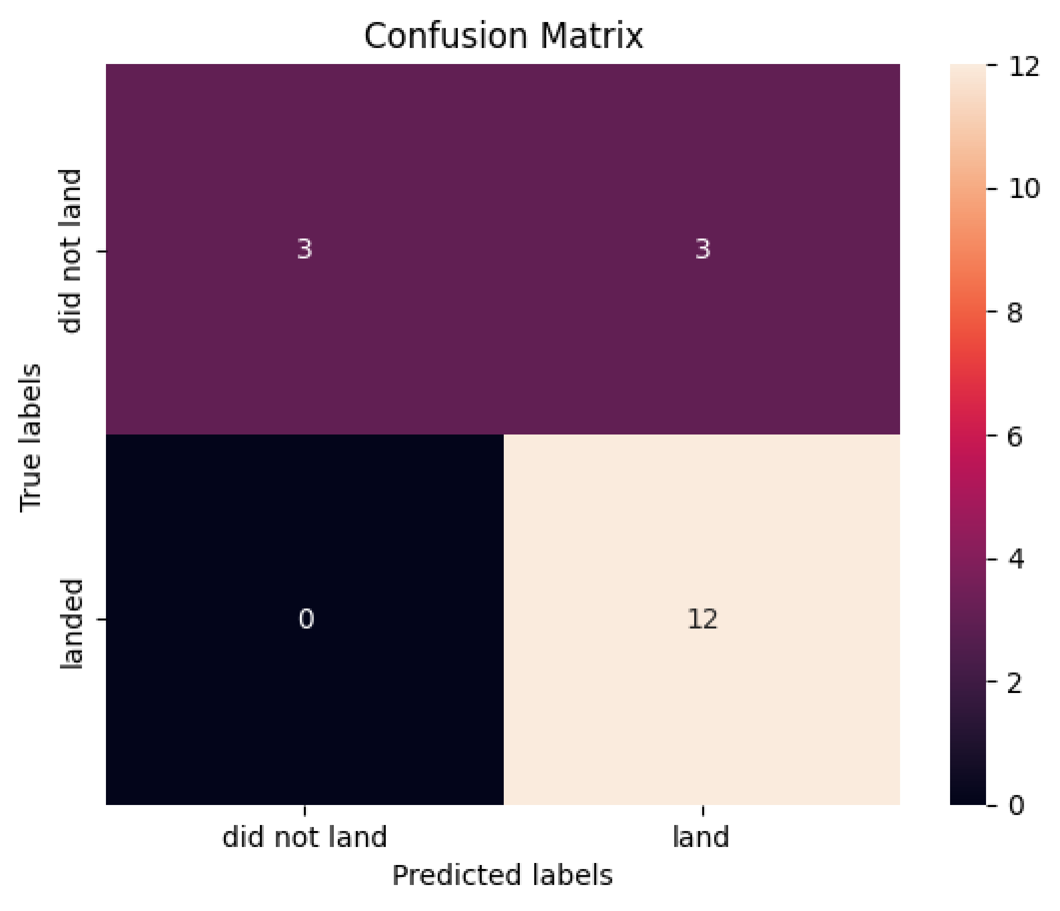
Classification Accuracy

- The accuracy is the same for all models and is equal to 0.8(3)



Confusion Matrix

- The confusion matrix is the same across all the models. The major problem is the false positives, meaning failure to land as success by the model.



Conclusions

- Learning curve does exist:
  - Success rate has been increasing since 2013
  - The more the number of attempt to land at a certain site, the greater the success rate at this site
- High altitude and/or stable orbits have almost 100% success rate in landing the 1st stage
- KSC LC-39A has the most successful landing among all the sites
- ML algorithms are having the same accuracy
- ML was quite instrumental in our objective to predict if the first stage landing of our competitor will land and as such to evaluate the launch cost

Appendix

Extra materials, including datasets, code, and others are available at GitHub Repository

Thank you!