Winning Space Race with Data Science

Mugiebelrahman Awad

June 2022

Executive Summary

- Summary of methodologies:
  - ❑ Data Collection through API and Web scraping.
  - ❑ Data Wrangling to clean and replace the Null fields on the dataset.
  - ❑ EDA with SQL, DATA Visualization, Interactive visualization, Folium.
  - ❑ Machine learning prediction to determine the success or the failure of the landed process.
- Summary of all results:
  - ❑ Discovered a pattern in dataset that can lead to determine the outcome.
  - ❑ A usage of different Machine learning model to ensure the accuracy of the prediction.
  - ❑ A set of data visualization and interactive visuals to enhance the understanding of the patterns.

Table of Content:

Introduction

- Project background and context:

SpaceX has the lowest cost in making skyrockets at 65 million USD and that being contributed to their reusage of the rocket first stage. While other space companies cost exceeding 165 million USD. This project is to replace the scientific equations in predicting whether the rocket will land successfully or destroyed while landing.

- Problems you want to find answers:

To predict whether the launched first stage rocket will land successfully to be reused or it will crash.

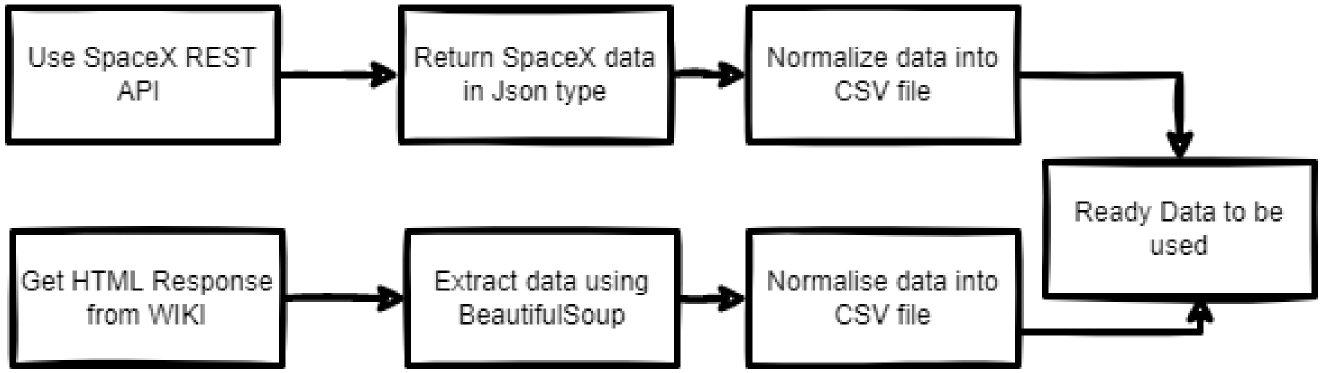
Methodology

Methodology

- Executive Summary
- Data collection methodology:
  - Data was collected using SpaceX REST API and Web scraping a falcon table from Wiki.
- Perform data wrangling
  - Data was cleaned and organized in a data frame, null values were detected and corrected using a mean procedure.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - LR, KNN, SVM, DT models have been built and evaluated for the best classifier.

Data Collection

- SpaceX launched data was gathered from the SpaceX REST API.
- This API will give information about the rocket used, payload delivered, launch specification, landing specification and outcome.
- Another popular data source is Wikipedia by performing a web scraping to obtain falcon 9 dataset and stored in data frame.



Data Collection – SpaceX API

- Data collection with SpaceX REST calls:

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

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

Check the content of the response

```python
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

```python
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()`

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

Here's The GitHub Link.

Data Collection - Scraping

- Web Scraping process:

9th June 2021

```
[4]: 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.

```
[5]: # use requests.get() method with the provided static_url

# assign the response to a object
data  = requests.get(static_url).text
```

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

```
[6]: # Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(data, 'html5lib')
```

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

```
[7]: # Use soup.title attribute
print(soup.title)
```

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

```
[8]: # Use the find_all function in the BeautifulSoup object, with element type `table`
# Assign the result to a list called `html_tables`
html_tables = soup.find_all('table')
```

Starting from the third table is our target table contains the actual launch records.

```
[9]: # Let's print the third table and check its content
first_launch_table = html_tables[2]
```

Here's GitHub [Link.](#)

Data Wrangling

- How data were processed:
  - Identified the null values percentage in each column and correct the missing values.
  - Calculated the launches on each site.
  - Calculated the number of occurrence of mission outcome per Orbit to identify the failures and created a new column indicates the outcome of each launch.

Here's the GitHub Link.

```
df.isnull().sum()/df.count()*100
```

```
for i,outcome in enumerate(landing_outcomes.keys()):
    print(i,outcome)
```

```
bad_outcomes=set(landing_outcomes.keys()[[1,3,5,6,7]])
bad_outcomes
```

```
{'False ASDS', 'False Ocean', 'False RTLS', 'None ASDS', 'None None'}
```

```
# 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)
```

This variable will represent the classification variable t
first stage landed Successfully

```
df['Class']=landing_class
df[['Class']].head(8)
```

EDA with Data Visualization

- Summarizing what charts plotted and why were used:
    - Plotted Pay Load Mass vs Flight number to see if they have an affect on the launch outcome.
        - Observed that with the increase of the payload mass and the flight number the success rates increases.
    - Launch site vs flight Number to identify their affect on the outcome.
        - In CCAFS SLC 40 site after flight number 60 the success rate is sky rocking.
    - Plotted the Orbit vs their outcome success rate.
        - I’ve found that there are few orbit a success rate almost a 100% like ES-l1, GEO, HEO and SSO.
    - Also plotted the launch success rate yearly trend.
        - It’s clear that after 2013 launch outcome had spike and a steady success rate.

  - Here’s GitHub [Link.]

EDA with SQL

- Performed the following SQL queries:

  - Names of unique launch sites in the space mission.
  - Total payload mass carried by boosters launched by NASA (CRS)
  - Average payload mass carried by booster version F9 v1.1
  - Total number of successful and failure mission outcomes
  - Failed landing outcomes in drone ship, their booster version and launch site names.

- Here's GitHub Link.

Build an Interactive Map with Folium

- Summarizing maps, objects added to a folium map:
    - Added each site's location on a map using site's latitude and longitude coordinates.
    - Pinned the Longitudes and latitudes on the map to visualize their location.
    - I’ve used the Object Circle to add a highlighted circle area with a text label on their specific coordinate.
    - A marker was added to each coordinate in the map to popup the location in the map.
    - A marker cluster is added to simplify a map containing many markers having the same coordinate.
    - added a MousePosition on the map to get coordinate for a mouse over a point on the map. As such, while exploring the map, can easily find the coordinates of any points of interests (such as railway).
    - Added PolyLine between a launch site to the selected coastline point with a distance mentioned at the end of the line.

- Here’s GitHub Link.

Build a Dashboard with Plotly Dash

- **Interactive Dashboard:**
    - **A dropdown menu on different launch site selection.**
    - **Plotted pie charts showing the total launches by a certain sites.**
    - **Plotted scatter graph showing the relationship with Outcome and Payload Mass (Kg) for the different booster version.**

- **Here’s GitHub Link.**

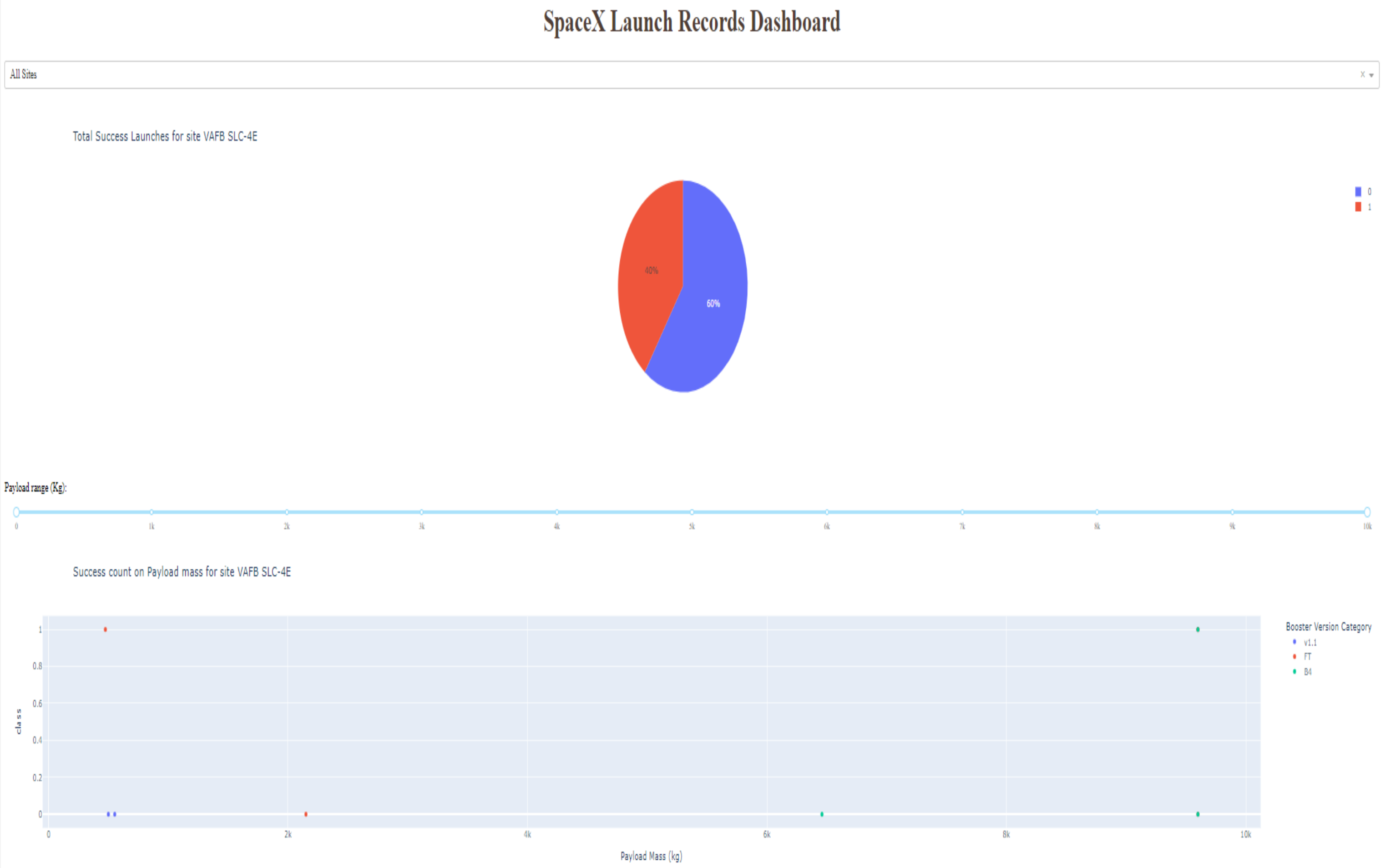
Predictive Analysis (Classification)

- Classification Models:
  - Loaded the data using numpy and pandas, transformed the data, split our data into training and testing.
  - Built different machine learning models and tune different hyperparameters using GridSearchCV.
  - Used accuracy as the metric for our model, improved the model using feature engineering and algorithm tuning.

- Here’s GitHub Link.

Results

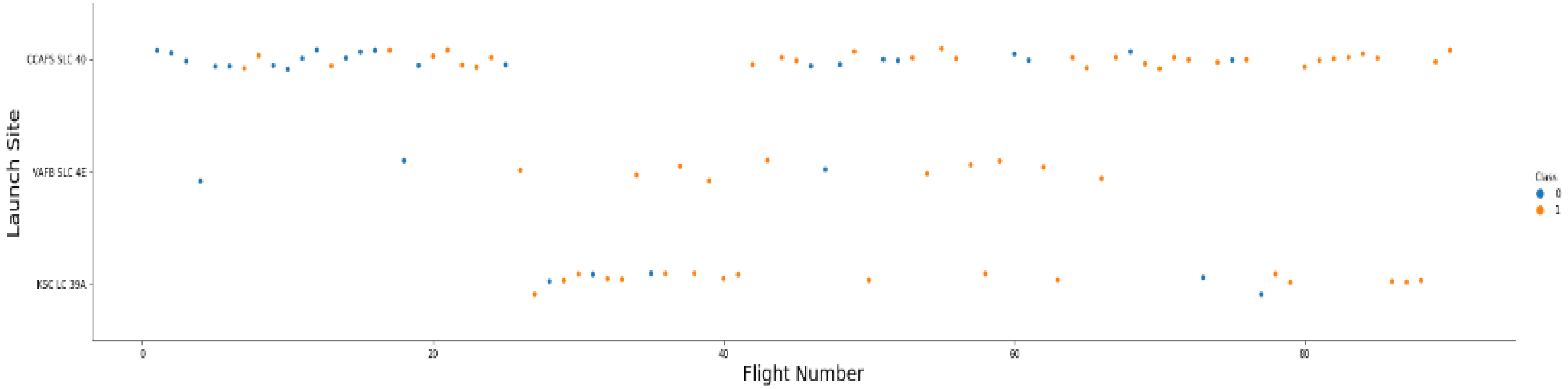
- The SVM, KNN and LR are the best on their prediction's accuracy.
- KSC LC 39A had the most successful launches from all the sites.
- Orbit GEO, HEO, SSO, ES l1 has the best success rate.
- The success rates for SpaceX launches is directly proportional to the years, in their future they will have a more successful rate.



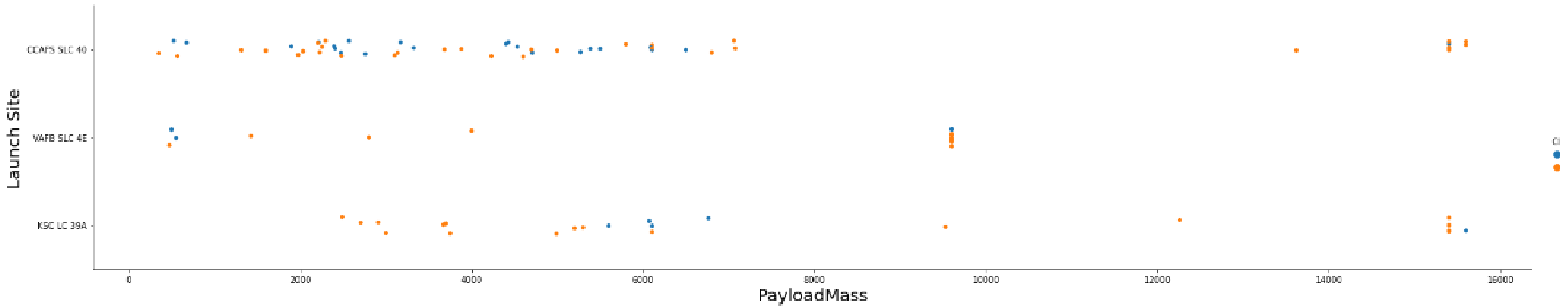
Insights Drawn from EDA

Flight Number vs. Launch Site

- In all the sites where the flight number is greater than 60 the success rate improves.
- In CCAFS SLC49 after flight number 80 the success rate is 100%.



Payload vs. Launch Site

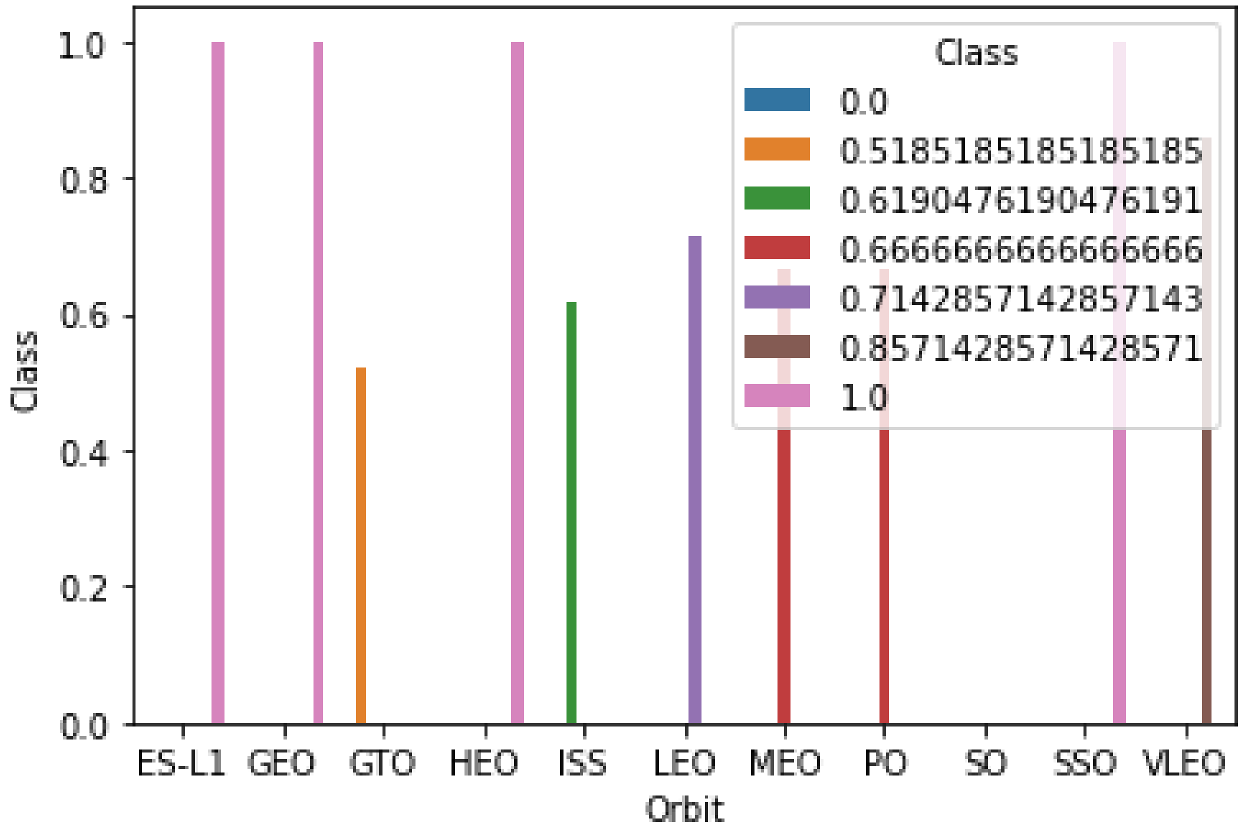


Now if you observe Payload Vs. Launch Site scatter point chart you will find for the VAFB-SLC launchsite there are no rockets launched for heavypayload mass(greater than 10000).

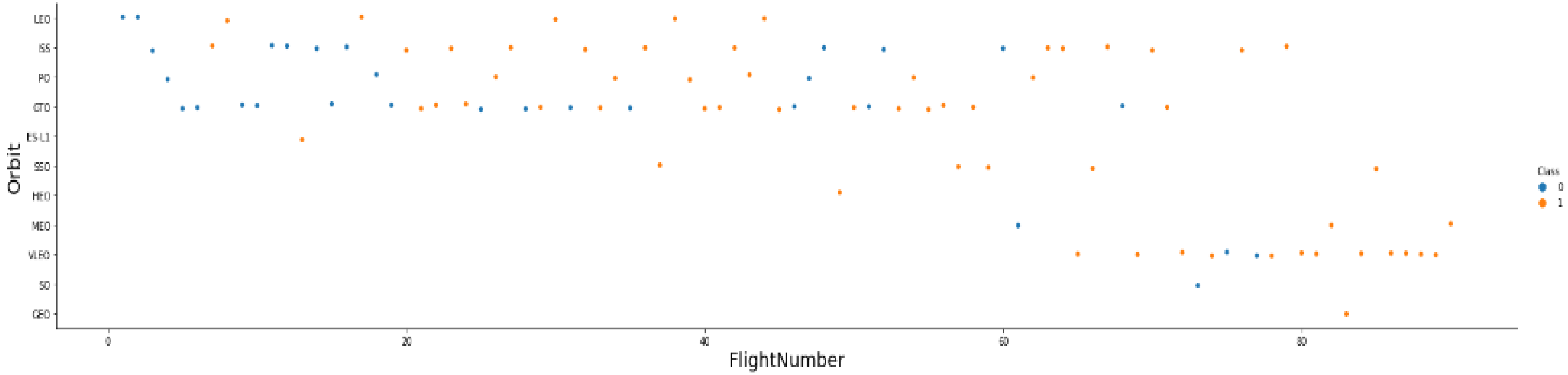
Success Rate vs. Orbit Type

- The following Orbit have a 100% success rate:
  - ES-l1.
  - GEO.
  - HEO.
  - SSO.

  And the Orbit GTO has the lowest success rate.

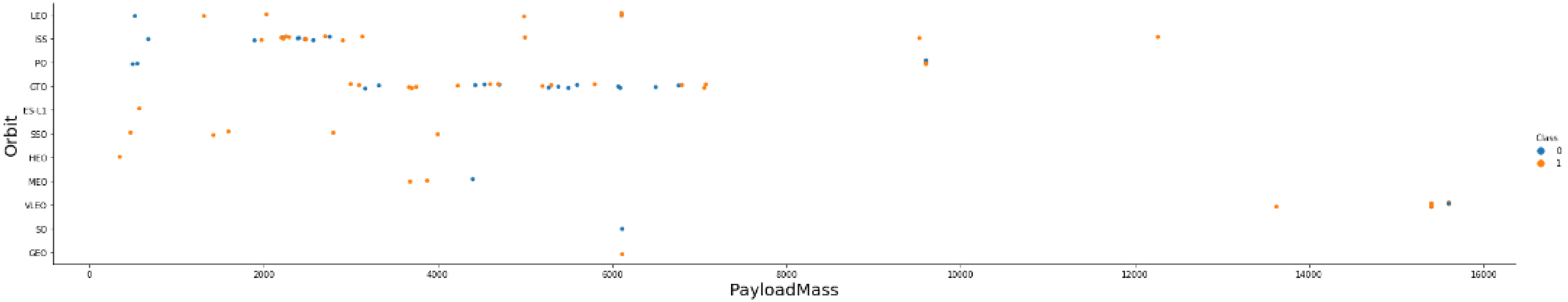


Flight Number vs. Orbit Type



You should see that in the LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.

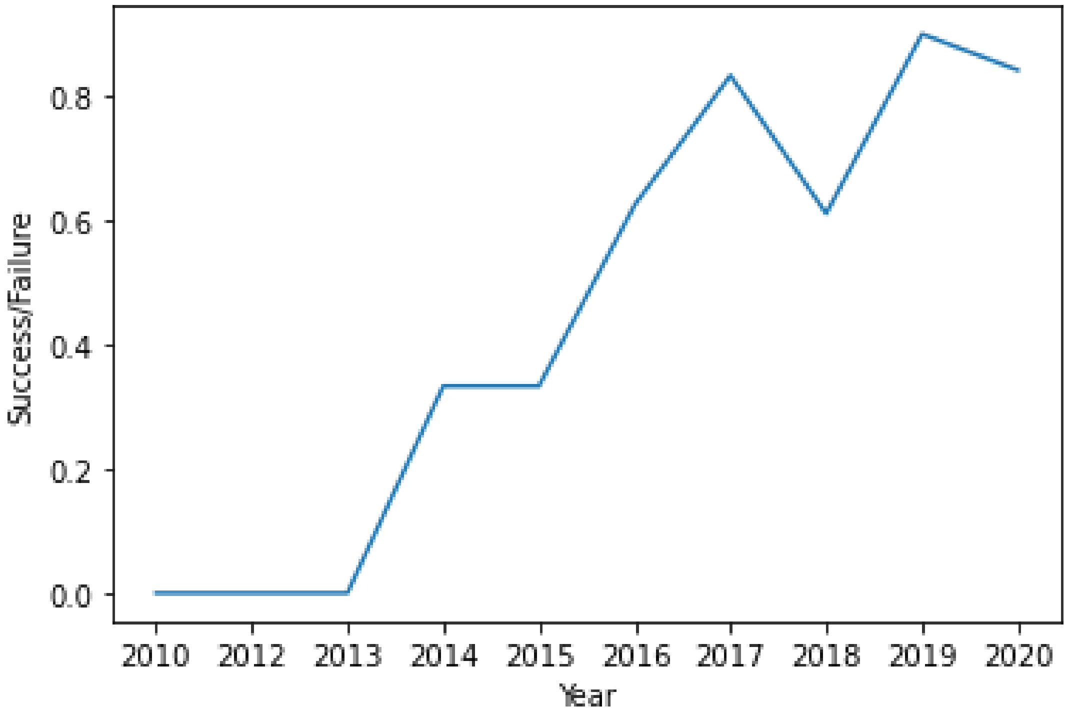
Payload vs. Orbit Type



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

However for GTO we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission) are both there here.

Launch Success Yearly Trend



you can observe that the sucess rate since 2013 kept increasing till 2020

All Launch Site Names

Used the key word DISTINCT to show only unique launch sites from the SpaceX data.

Task 1

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

In [6]:
```
%sql SELECT DISTINCT LAUNCH_SITE FROM SPACEXTBL;
```

```
 * ibm_db_sa://xqx72778:***@b0aebb68-94fa-46ec-a1fc-1c999edb6187.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:31249/bludb
Done.
```

Out[6]:

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

Launch Site Names Begin with 'CCA'

Used the query to display 5 records where launch sites begin with `CCA`

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

```
]: %sql SELECT * FROM SPACEXTBL WHERE LAUNCH_SITE Like 'CCA%' limit 5;
```

```
 * ibm_db_sa://xqx72778:***@b0aebb68-94fa-46ec-a1fc-1c999edb6187.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:31249/bludb
Done.
```

]:

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

Task 3

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

```
In [20]: %sql SELECT sum(payload_mass__kg_) as Total_Payloadmass FROM SPACEXTBL WHERE customer like 'NASA%';
```

```
 * ibm_db_sa://xqx72778:***@b0aebb68-94fa-46ec-a1fc-1c999edb6187.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:31249/bludb
Done.
```

Out[20]:

total_payloadmass
99980

Average Payload Mass by F9 v1.1

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

:4]:
```
%sql SELECT AVG(payload_mass__kg_) as AVG_Payloadmass FROM SPACEXTBL WHERE booster_version = 'F9 v1.1';
```

```
 * ibm_db_sa://xqx72778:***@b0aebb68-94fa-46ec-a1fc-1c999edb6187.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:31249/bludb
Done.
```

:4]:

avg_payloadmass
2928

First Successful Ground Landing Date

```
[26]: %sql SELECT min(DATE) as List_date FROM SPACEXTBL WHERE landing__outcome = 'Success (ground pad)';

 * ibm_db_sa://xqx72778:***@b0aebb68-94fa-46ec-a1fc-1c999edb6187.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:31249/bludb
Done.
```

[26]:

list_date
2015-12-22

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

7]:

```
%%sql
SELECT BOOSTER_VERSION
FROM SPACEXTBL
WHERE LANDING__OUTCOME = 'Success (drone ship)'
    AND 4000 < PAYLOAD_MASS__KG_ < 6000;
```

```
 * ibm_db_sa://xqx72778:***@b0aebb68-94fa-46ec-a1fc-1c999edb6187.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:31249/bludb
Done.
```

7]:

booster_version
F9 FT B1021.1
F9 FT B1023.1
F9 FT B1029.2
F9 FT B1038.1
F9 B4 B1042.1
F9 B4 B1045.1
F9 B5 B1046.1

Total Number of Successful and Failure Mission Outcomes

List the total number of successful and failure mission outcomes

```
%%sql
SELECT MISSION_OUTCOME, COUNT(MISSION_OUTCOME) AS TOTAL_NUMBER
FROM SPACEXTBL
GROUP BY MISSION_OUTCOME;
```

```
 * ibm_db_sa://xqx72778:***@b0aebb68-94fa-46ec-a1fc-1c999edb6187.c3n41cmd0nqnrk39u98g.
Done.
```

mission_outcome	total_number
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

Boosters Carried Maximum Payload

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

29]:

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

```
 * ibm_db_sa://xqx72778:***@b0aebb68-94fa-46ec-a1fc-1c999edb6187.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:31249/bludb
Done.
```

29]:

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

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

```
%%sql
SELECT LANDING__OUTCOME, BOOSTER_VERSION, LAUNCH_SITE
FROM SPACEXTBL
WHERE Landing__Outcome = 'Failure (drone ship)'
    AND YEAR(DATE) = 2015;
```

```
 * ibm_db_sa://xqx72778:***@b0aebb68-94fa-46ec-a1fc-1c999edb6187.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:31249/bludb
Done.
```

landing__outcome	booster_version	launch_site
Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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

```
[ ]: %%sql
SELECT LANDING__OUTCOME, COUNT(LANDING__OUTCOME) AS TOTAL_NUMBER
FROM SPACEXTBL
WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY LANDING__OUTCOME
ORDER BY TOTAL_NUMBER DESC
```

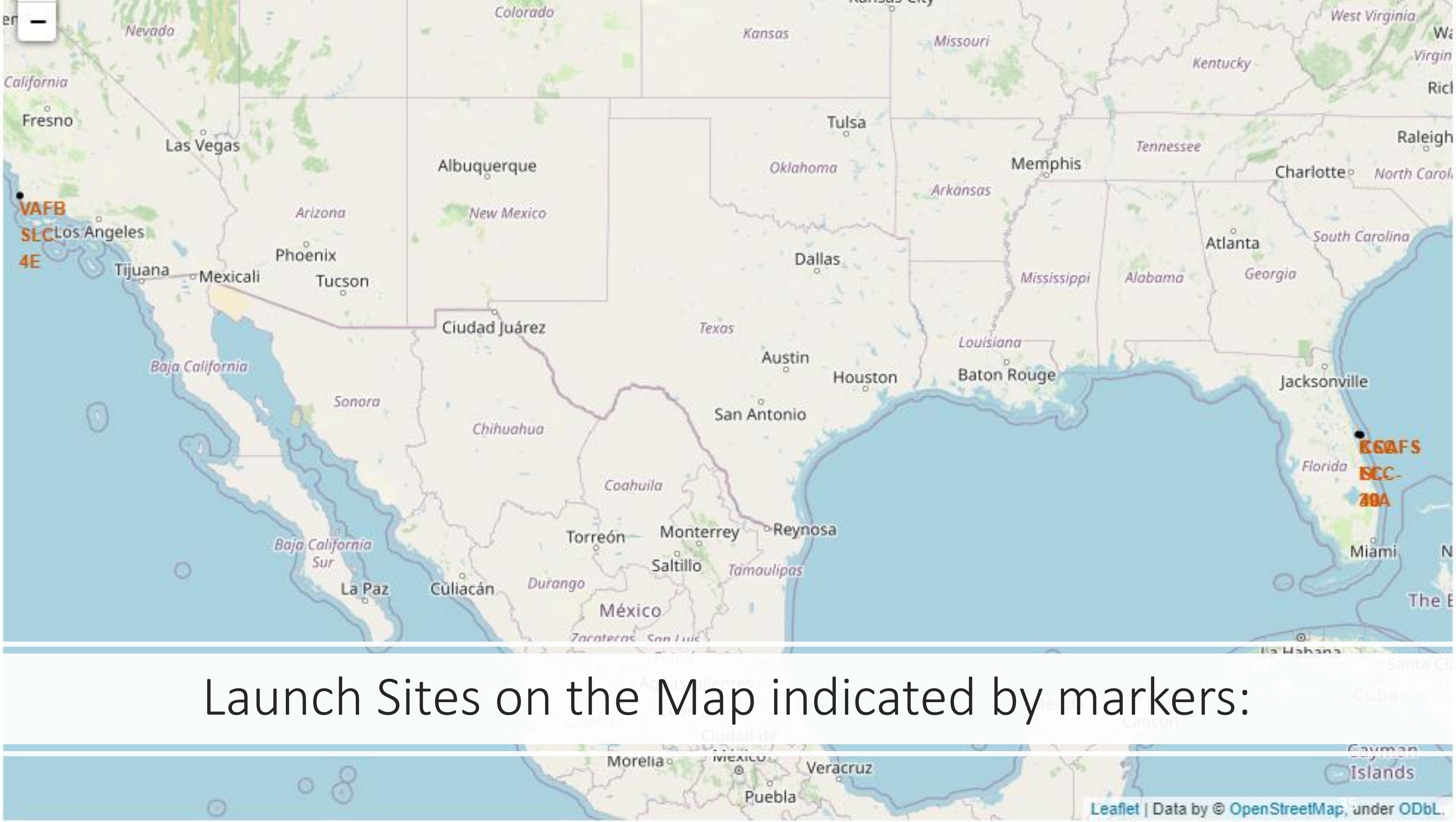
```
 * ibm_db_sa://xqx72778:***@b0aebb68-94fa-46ec-a1fc-1c999edb6187.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:31249/bludb
Done.
```

[ ]:

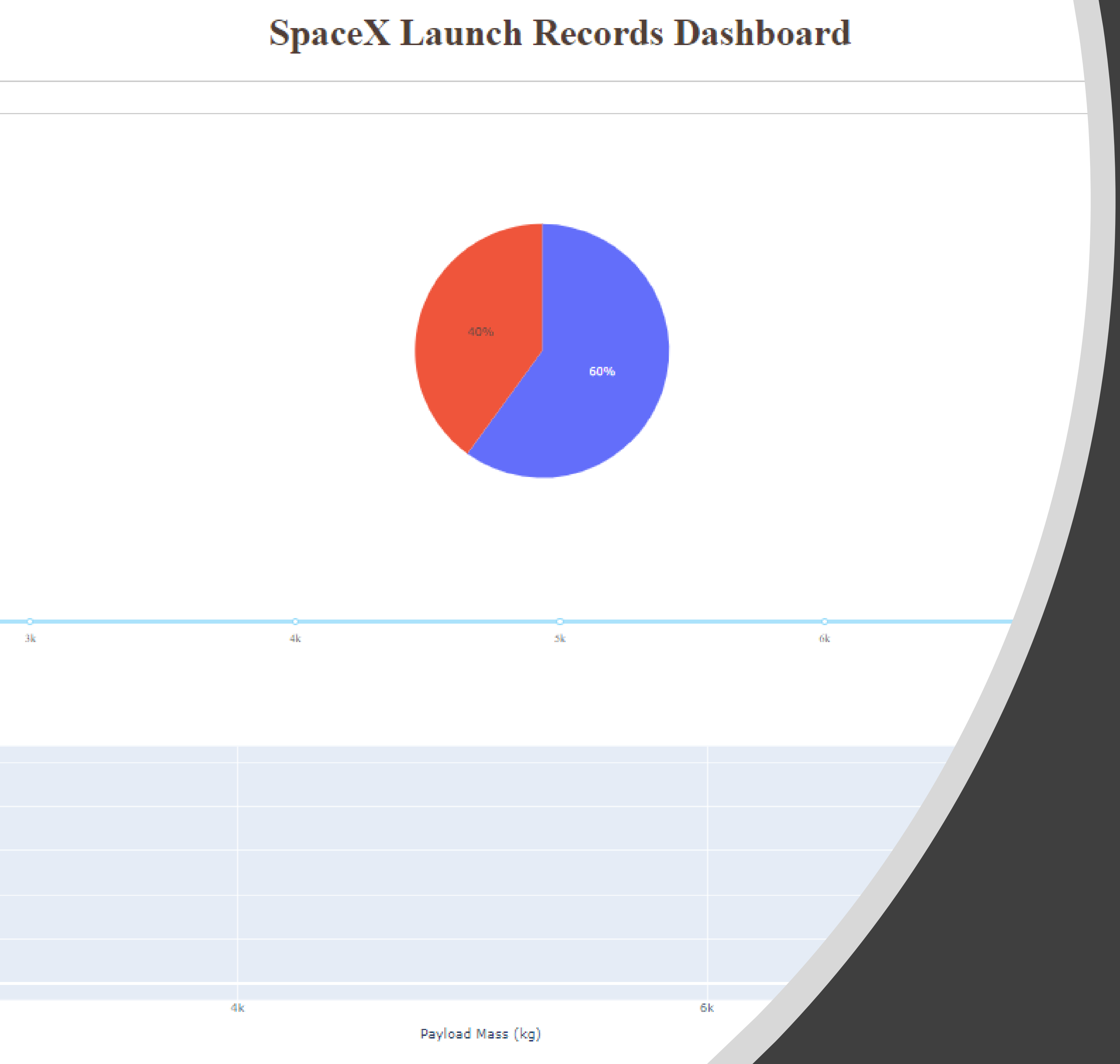
landing__outcome	total_number
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

Launch Site Proximities Analysis

Launch Sites on the Map indicated by markers:



Dashboard with Plotly Dash



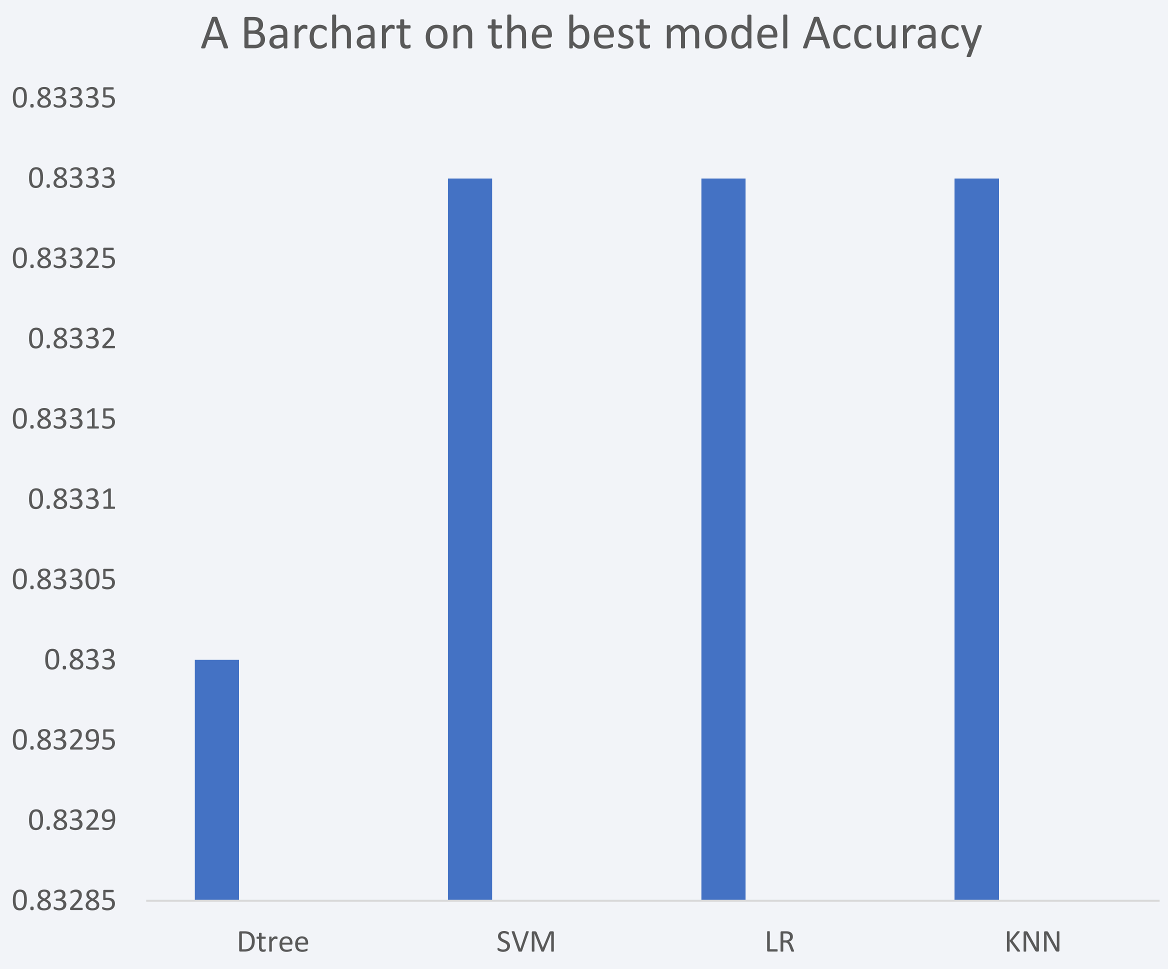
Dashboard:

- A Pie chart show the success rate based on the select Launch site.
- A scatter plot chart to show the success count on the payload mass for the selected site.

Predictive analysis
(Classifications)

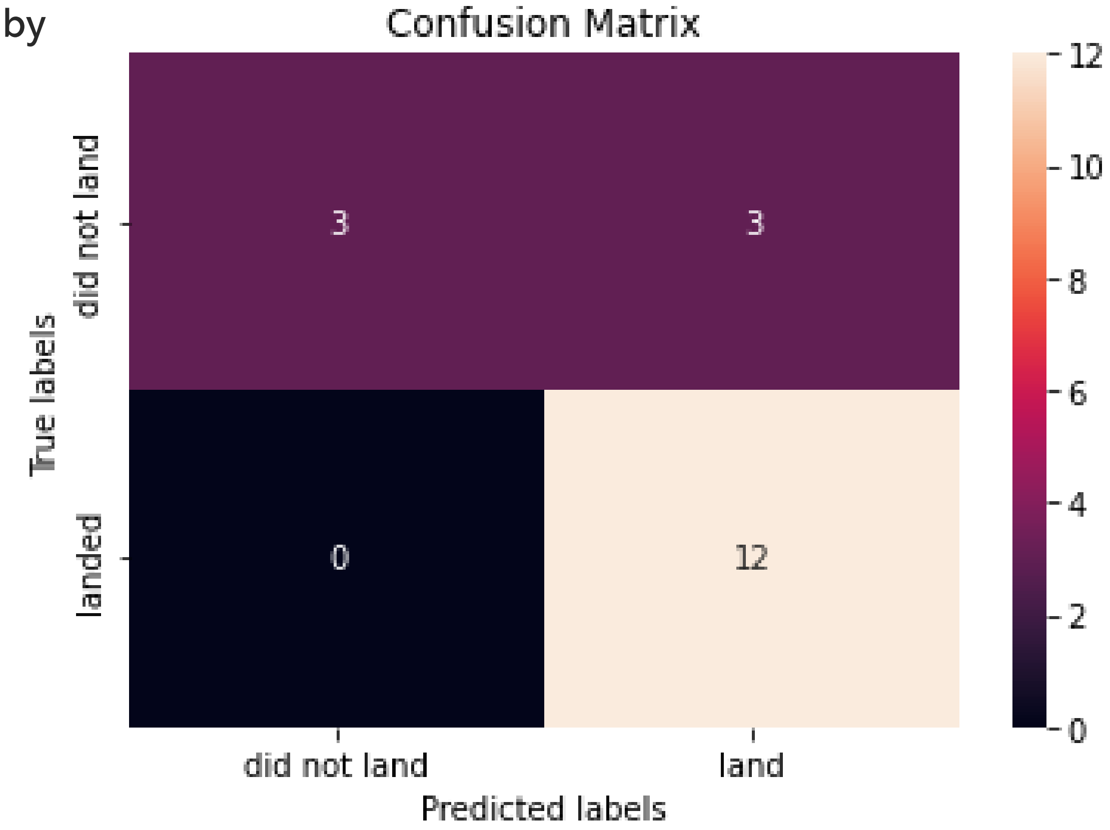
Classification Accuracy

- SVM, LR and KNN has the best model accuracy.



Confusion Matrix

- All the models resulted in the same confusion Matrix.
- The major problem is the false positives .i.e., Unsuccessful landing marked as successful landing by the classifier.



Conclusions

- The larger the flight amount at a launch site, the greater the success rate at a launch site.
- Launch success rate started to increase in 2013 till 2020.
- Orbits ES-L1, GEO, HEO, SSO, VLEO had the most success rate.
- KSC LC-39A had the most successful launches of any sites.
- All the models have the almost the same accuracy.

Thank You !