Winning Space Race with Data Science

<Mohamed MZAOUALI>
<June 11 2022>

Outline

Executive Summary

Analyzing the ‘Space X’ success rate and launch behavior by:

- Exploratory Data analysis and Data visualization and presentation.
- Predictive analysis via multiple machine learning algorithms.

These paradigms resulted in:

- Drawing a general picture of launch operation.
- Succeeding the task of predicting the “First Stage Landing of ‘Space X’ Falcon 9” .

Introduction

Project background and context:

The enterprise is an attempt to understand the logic behind the operational and choices of launching behavior of ‘Space X’.

Problems you want to find answers:

The possibility of predicting “The “First Stage Landing of ‘Space X’ Falcon 9”.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Web-scrapping Data from various remote Data Sources.
- Perform data wrangling
  - Transforming data by feature engineering some attributes.
- 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 and choosing the best classification model.

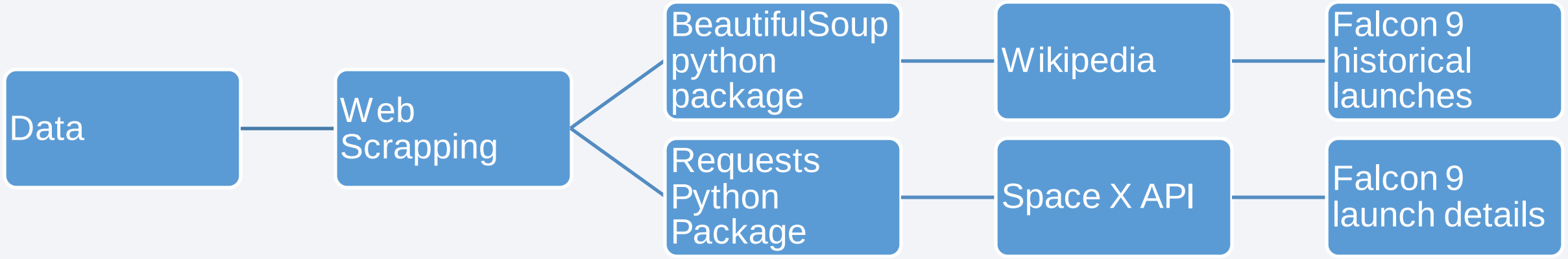
Data Collection

The Data was collected using the methods in the flowchart below.

The detailed methods are explained in the following notebooks:

- https://github.com/mzaoualim/IBM-Data-Science-Professional-Certificate/blob/main/jupyter_labs_webscraping.ipynb

- https://github.com/mzaoualim/IBM-Data-Science-Professional-Certificate/blob/main/jupyter_labs_spacex_data_collection_api.ipynb

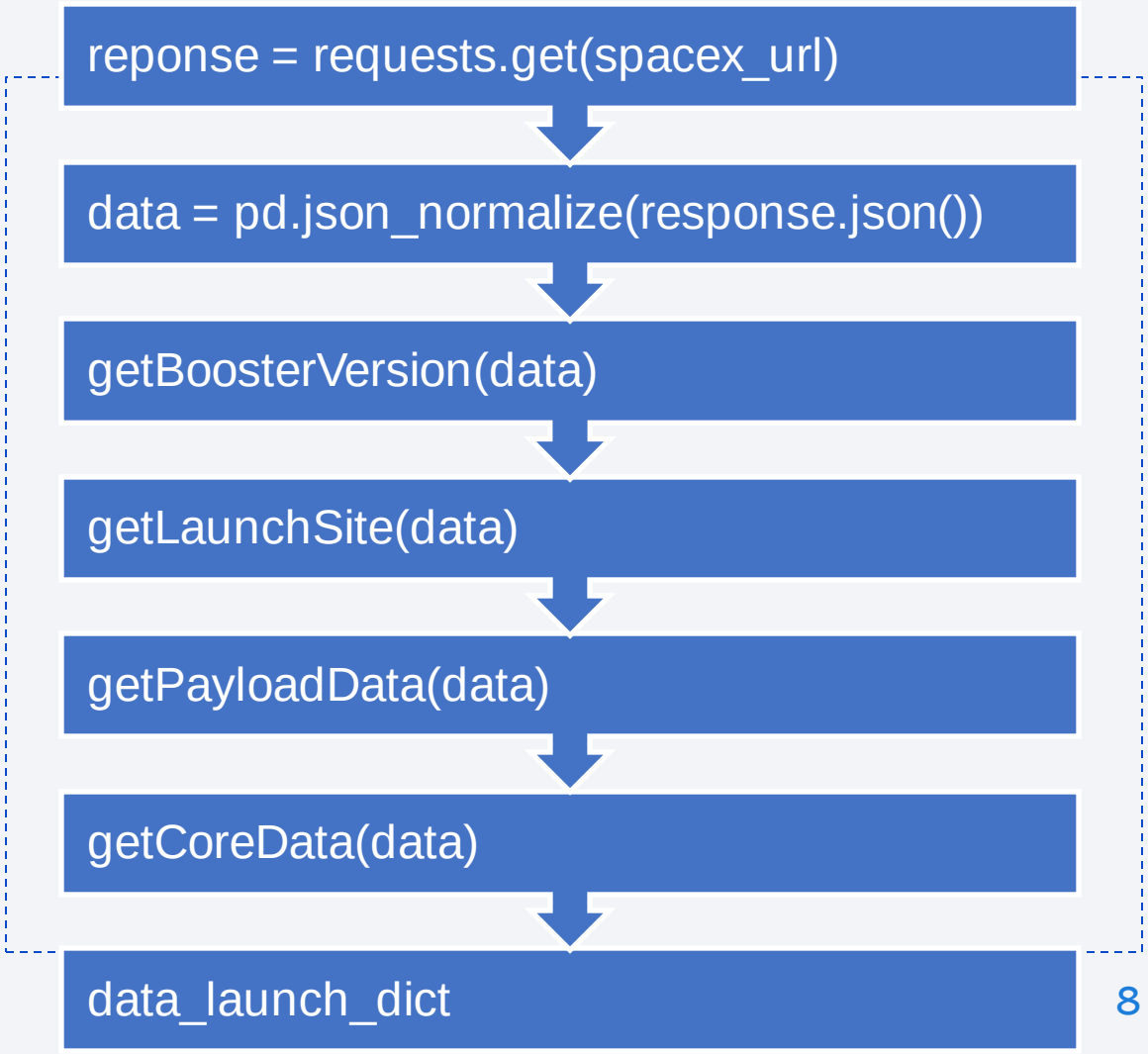


Data Collection – SpaceX API

- Requesting Data from SpaceX API calls using custom functions as summarized in the flowchart.

- The detailed notebook:

https://github.com/mzaoualim/IBM-Data-Science-Professional-Certificate/blob/main/jupyter_labs_spacex_data_collection_api.ipynb

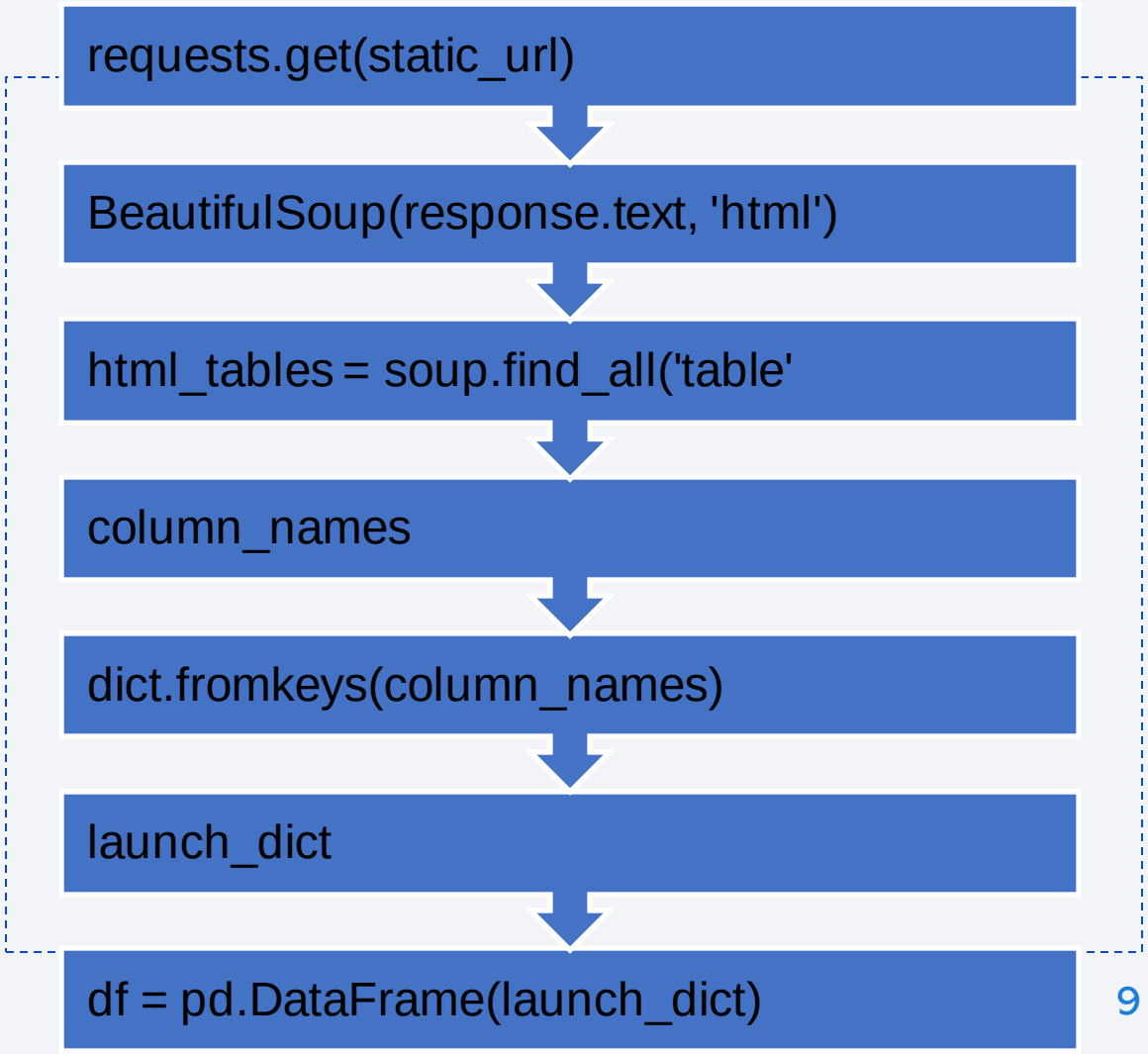


Data Collection - Scraping

- Web Scraping Data from Wikipedia using BeautifulSoup package and some custom functions as summarized in the flowchart.

- The detailed notebook:

https://github.com/mzaoualim/IBM-Data-Science-Professional-Certificate/blob/main/jupyter_labs_webscraping.ipynb



Data Wrangling

The Data Wrangling consisted mainly in feature engineering a new binary variable 'Class' of succeed (1) or failed (0) booster landing as summarized in the flowchart below.

The detailed notebook:
https://github.com/mzaoualim/IBM-Data-Science-Professional-Certificate/blob/main/labs_jupyter_spacex_Data_wrangling.ipynb

EDA with Data Visualization

EDA phase consisted in:

- Exploring visually the relationship between different features.
- Creating dummy variables to categorical columns.

For details and more:

https://github.com/mzaoualim/IBM-Data-Science-Professional-Certificate/blob/main/jupyter_labs_eda_dataviz.ipynb

EDA with SQL

- **Performed SQL queries:**
  - SELECT DISTINCT LAUNCH_SITE FROM SPACEXTBL;
  - FROM SPACEXTBL WHERE LAUNCH_SITE LIKE "%CCA%" LIMIT 5;
  - SELECT SUM(PAYLOAD_MASS__KG_) AS TOTAL_PAYLOAD_NASA FROM SPACEXTBL WHERE Payload LIKE '%CRS%';
  - SELECT AVG(PAYLOAD_MASS__KG_) AS AVERAGE_F9_MASS FROM SPACEXTBL WHERE BOOSTER_VERSION = 'F9 v1.1';
  - SELECT MIN(DATE) AS FIRST_SUCCESS FROM SPACEXTBL WHERE [LANDING _OUTCOME] = 'Success (ground pad)';
  - SELECT DISTINCT Booster_Version FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000 AND [LANDING _OUTCOME] = 'Success (drone ship)';
  - SELECT MISSION_OUTCOME, COUNT(*) AS COUNTS FROM SPACEXTBL GROUP BY MISSION_OUTCOME;
  - SELECT BOOSTER_VERSION AS MAX_BOOSTER FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ == (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL);
  - SELECT BOOSTER_VERSION, LAUNCH_SITE FROM SPACEXTBL WHERE [LANDING _OUTCOME] = 'Failure (drone ship)' AND SUBSTR(DATE, 7, 4) = '2015';
  - SELECT [LANDING _OUTCOME], COUNT(*) AS COUNTS FROM SPACEXTBL WHERE DATE BETWEEN '04-06-2010' AND '20-03-2017' AND [LANDING _OUTCOME] LIKE '%Success%' GROUP BY [LANDING _OUTCOME] ORDER BY COUNTS DESC;
  -
- **For details and more:**
  - https://github.com/mzaoualim/IBM-Data-Science-Professional-Certificate/blob/main/jupyter_labs_eda_sql_coursera_sqllite.ipynb

Build an Interactive Map with Folium

The location (map) analysis was performed via Folium python package by adding:

- Markers and Circles
  - To Mark all launch sites on a map.
  - To Mark the success/failed launches for each site on the map.
- Clusters:
  - To identify the number of launch attempts per each site.
  - To identify success launch attempts per each site.
- Lines
  - Tp calculate the distances between a launch site to its proximities.

- **The detailed notebook:**
  - https://github.com/mzaoualim/IBM-Data-Science-Professional-Certificate/blob/main/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

Plotly Dash

Interactive Dash Boards:

Pie Chart

Scatter Plots

Total Launch

Booster Versions

Launch Sites

Outcomes

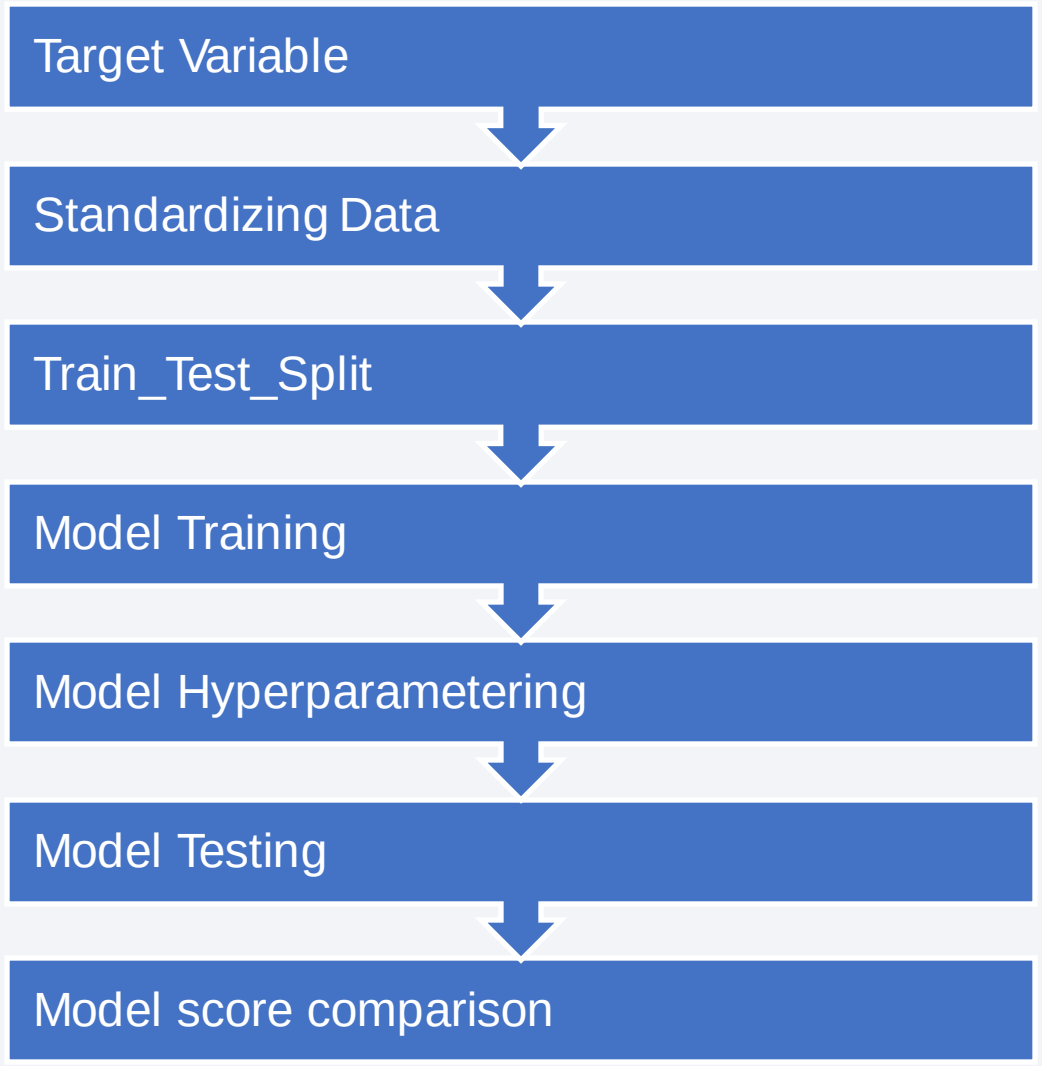
Pay Loads

Predictive Analysis (Classification)

The classification was performed following the phases summarized in the flowchart.

For more details:

https://github.com/mzaoualim/IBM-Data-Science-Professional-Certificate/blob/main/SpaceX_Machine_Learning_Prediction_Part_5.ipynb



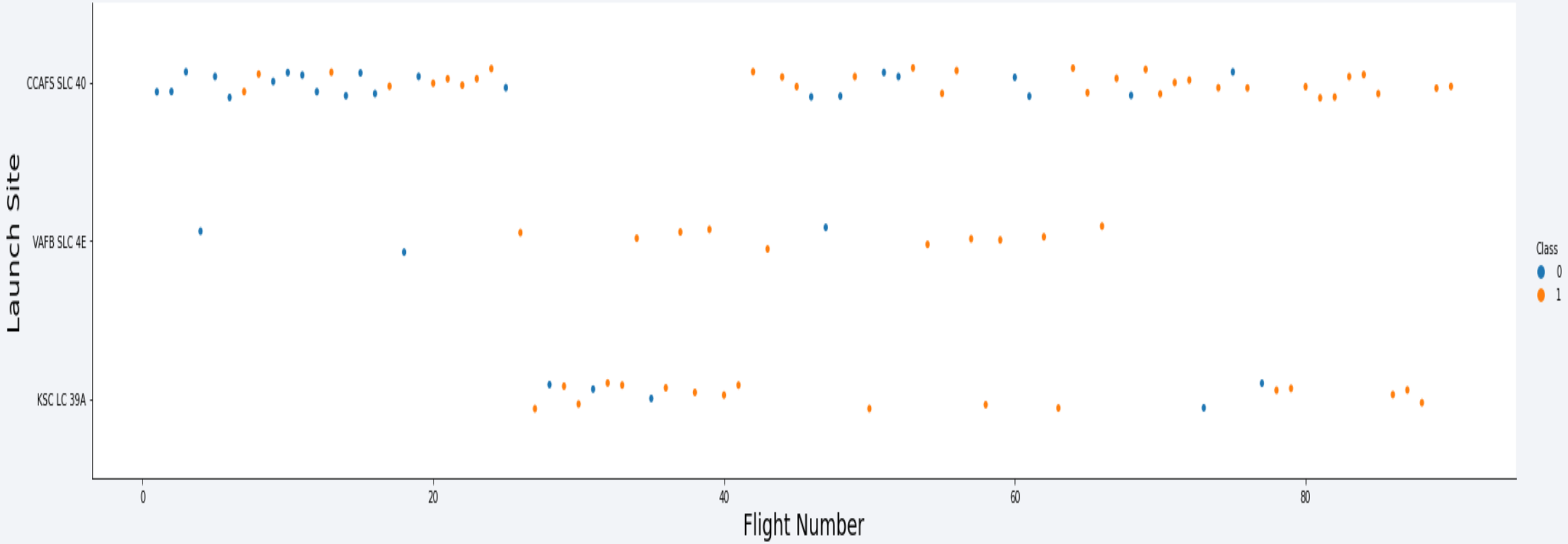
Results

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

Section 2

Insights drawn from EDA

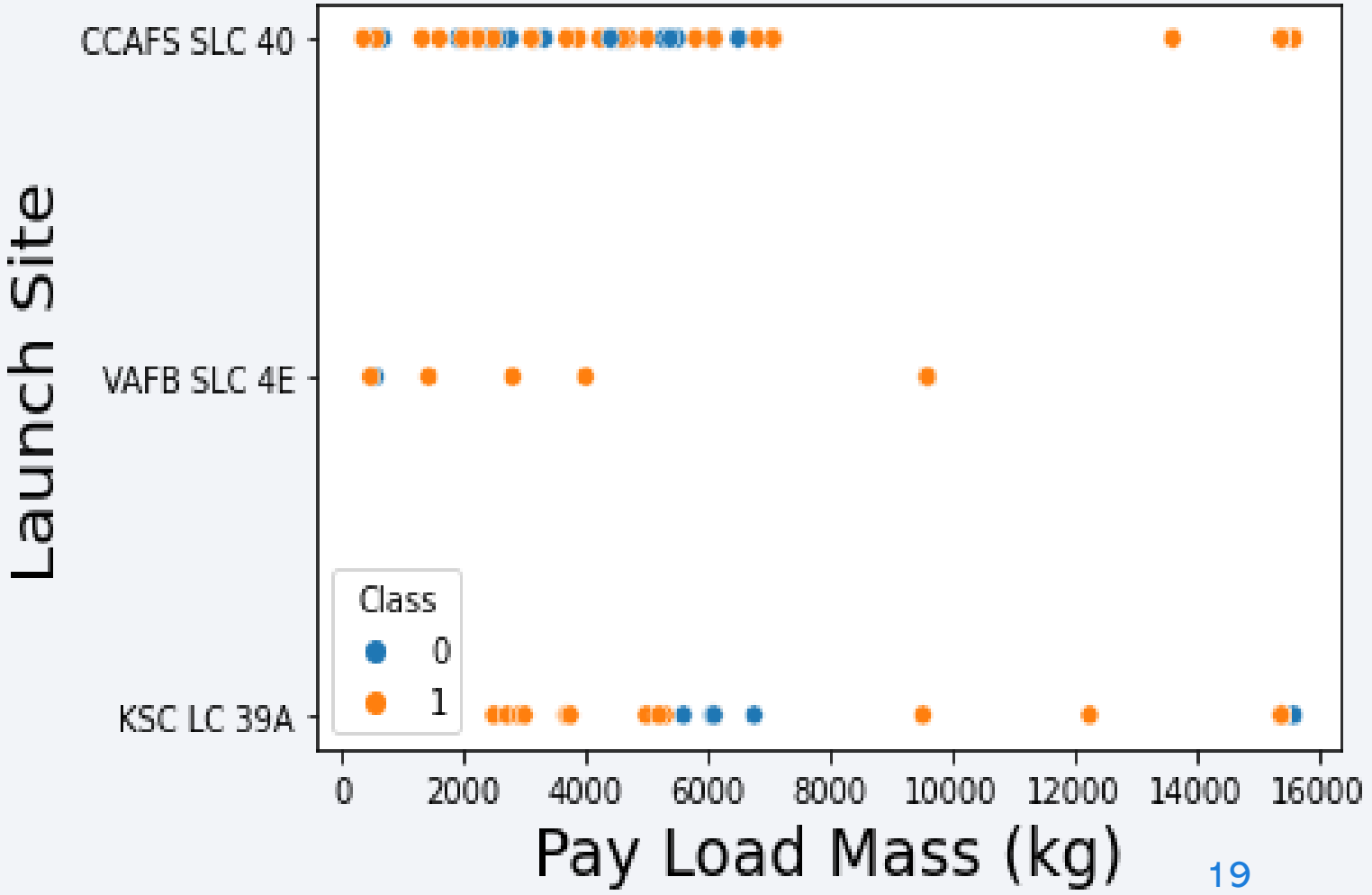
Flight Number vs. Launch Site



- **The more flight attempts, the greater the likelihood of success.**

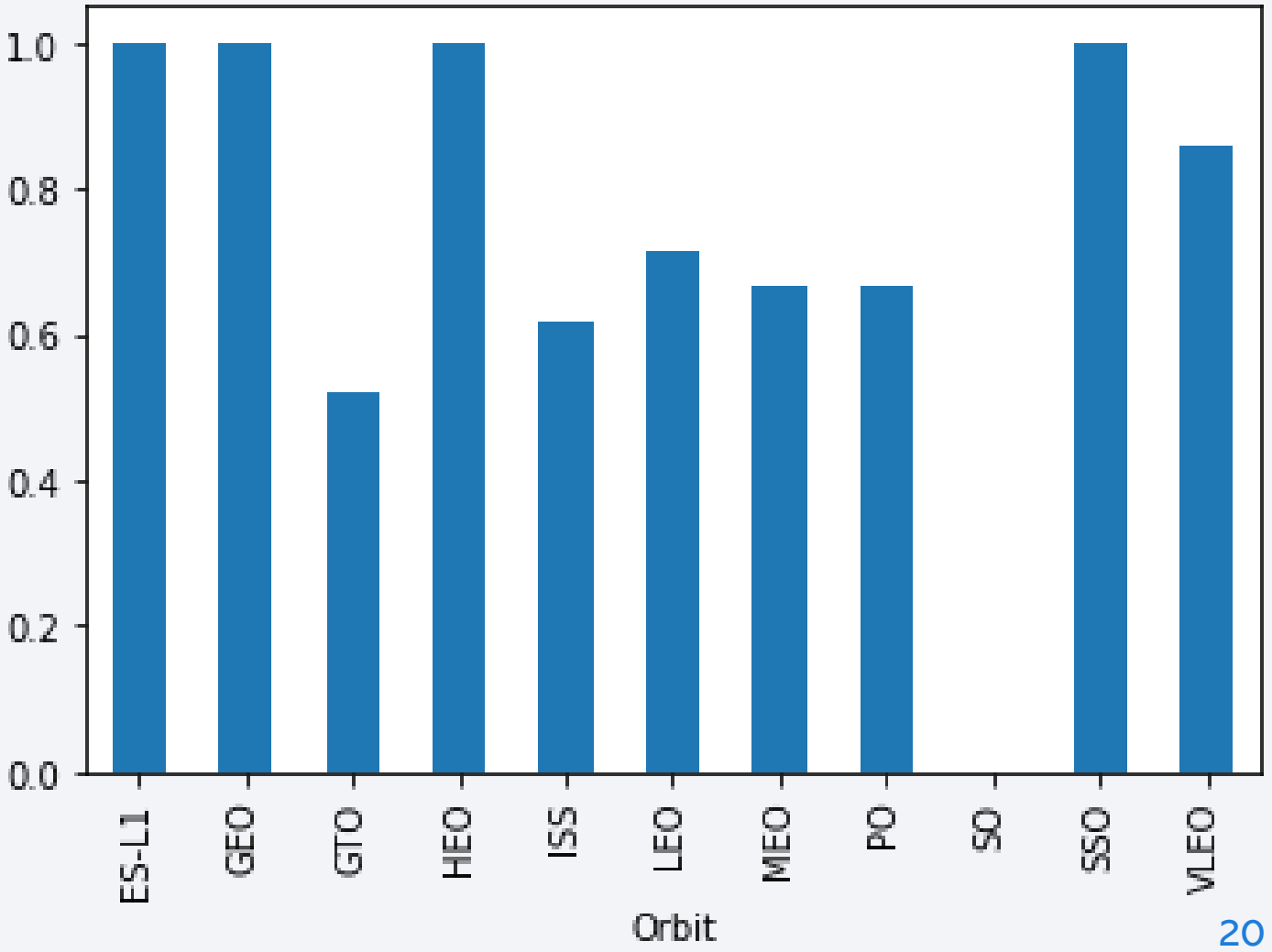
Payload vs. Launch Site

- The Launch Site 'VAFB SLC 4E' has no Pay Load greater than 10000 kg despite having a good record of success.
- The greater the pay loads comes with a good chance of success.



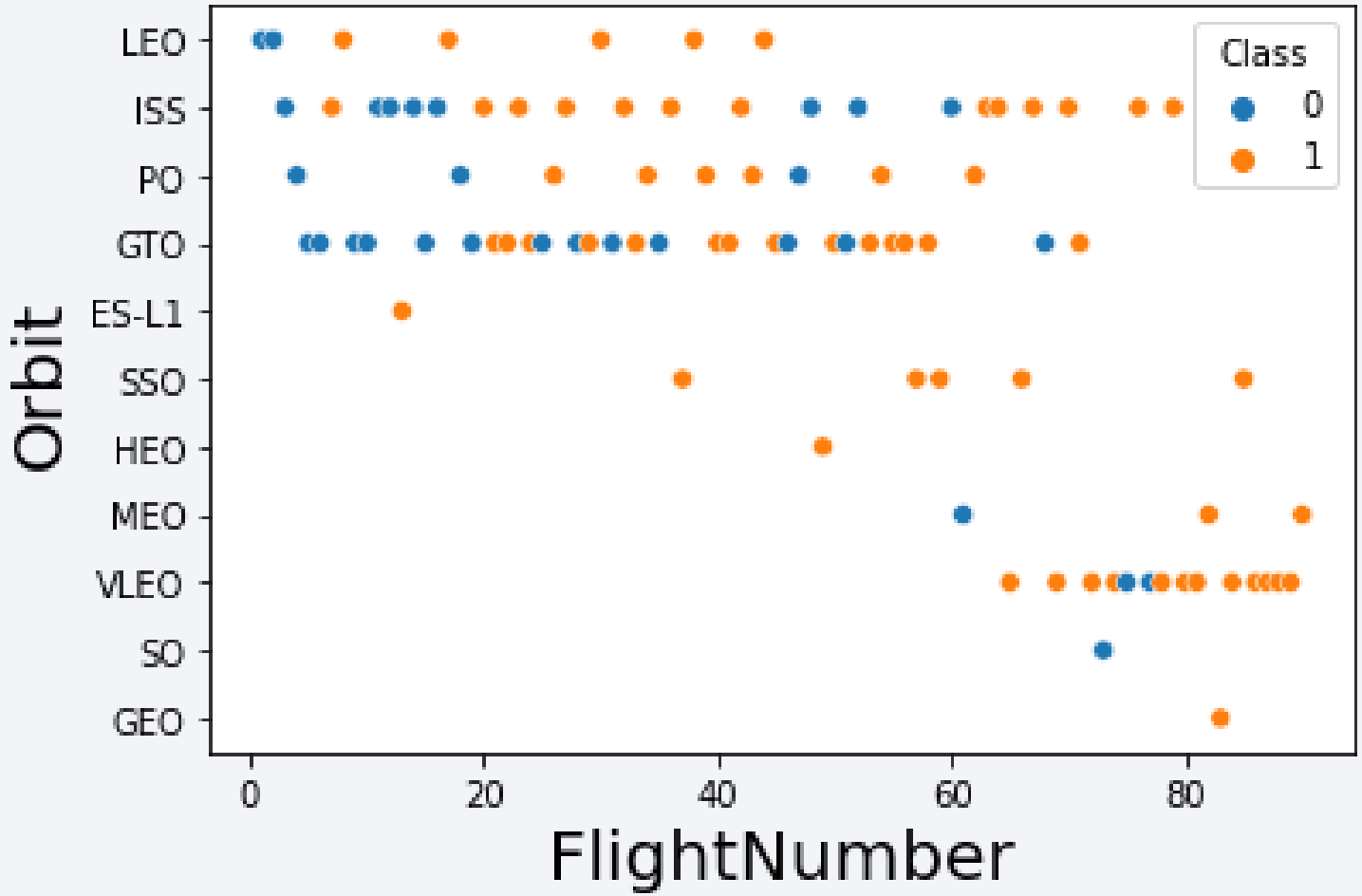
Success Rate vs. Orbit Type

- The 'ES-L1', 'GEO', 'HEO' and 'SSO' orbits are having a guaranteed success.
- The lower the orbit type, the least the success is expected.



Flight Number vs. Orbit Type

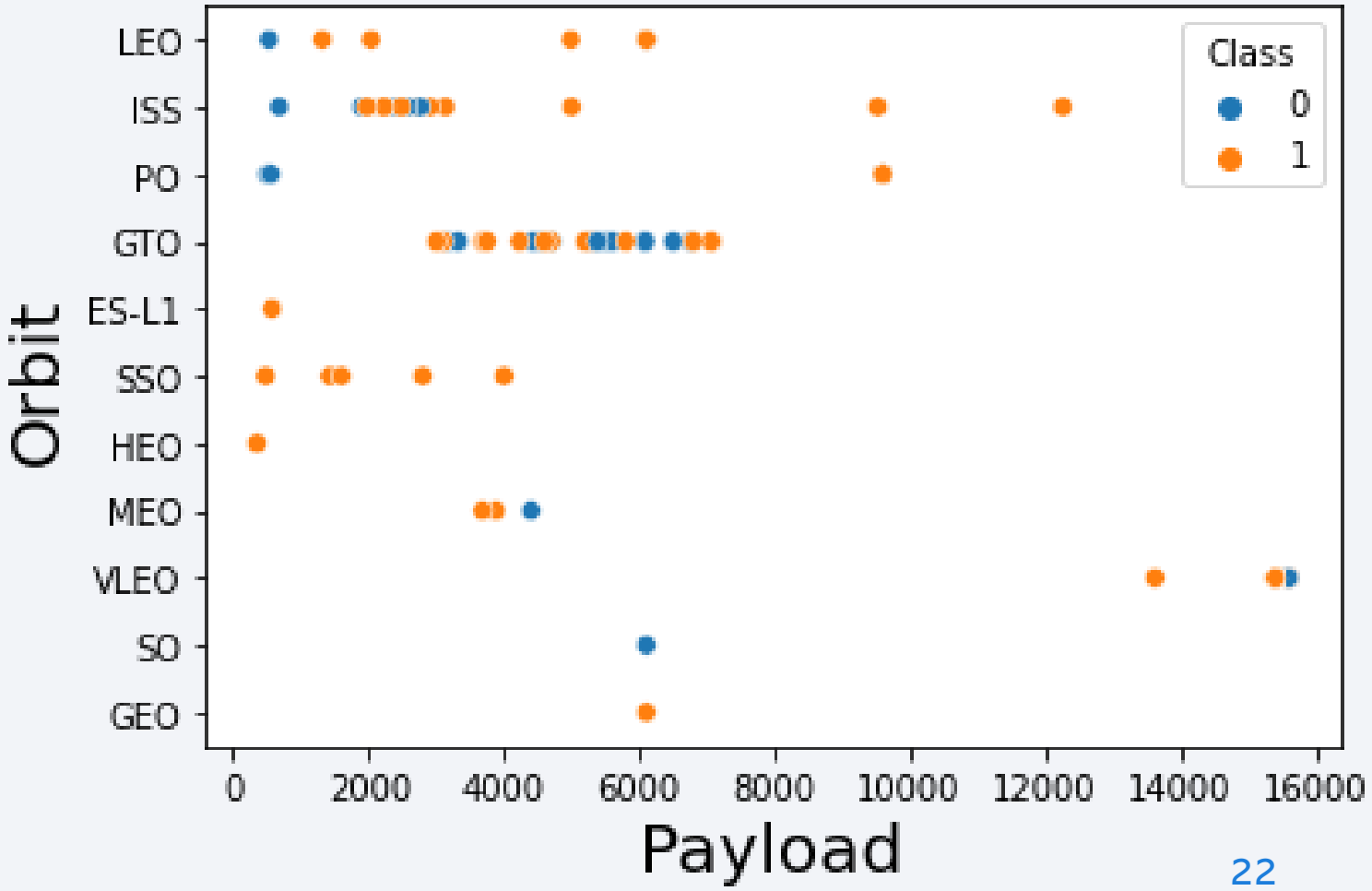
- In the LEO orbit, the Success appears related to the number of flights;
- No relationship between flight number when in GTO orbit.



Payload vs. Orbit Type

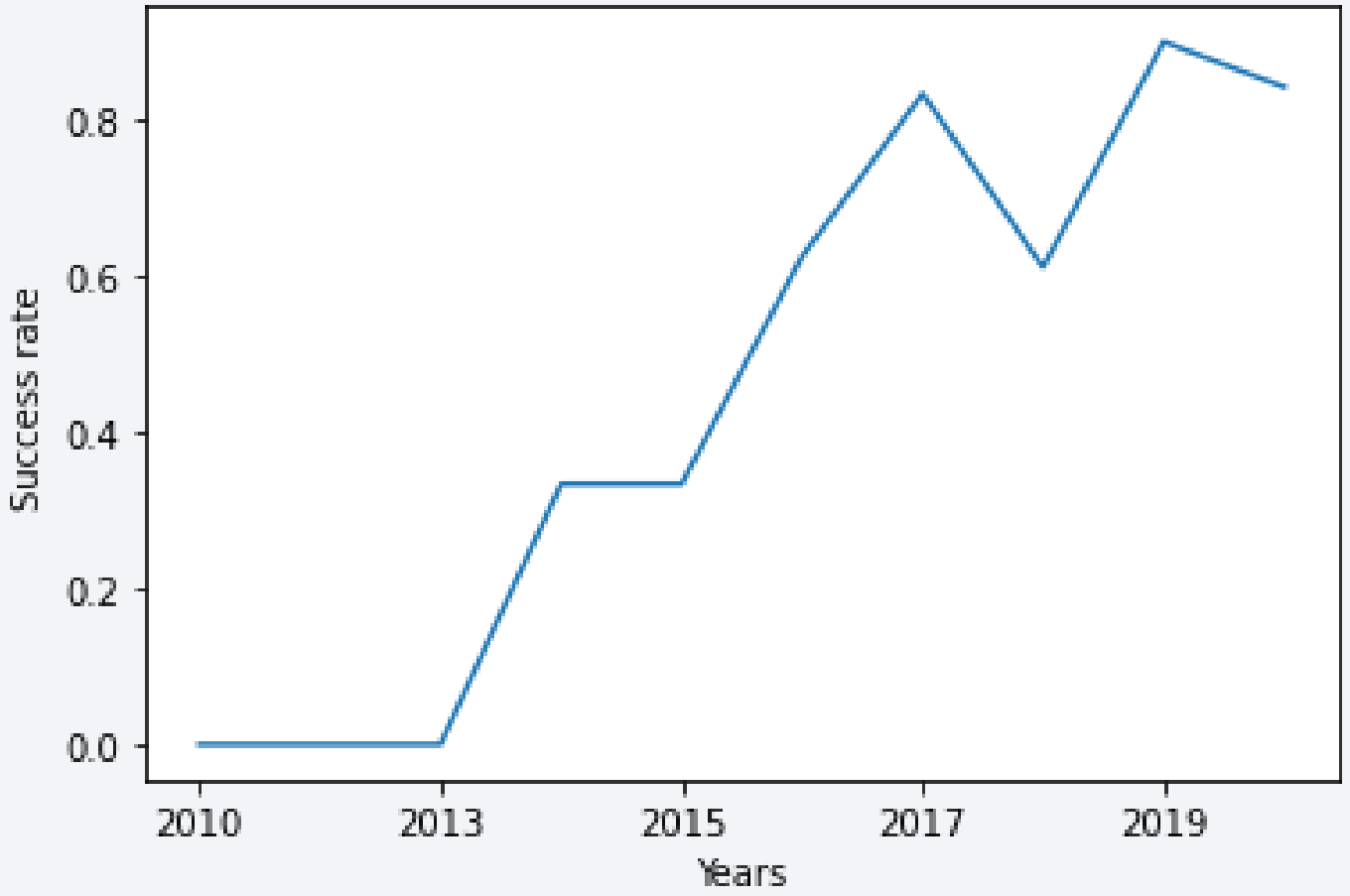
•With heavier payloads, Polar,LEO and ISS mark more success.

•Unlike GTO orbit, the relationship is inconclusive.



Launch Success Yearly Trend

- A good steady improving of success rate.



All Launch Site Names

Task 1

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

In [17]:

```
%%sql
SELECT DISTINCT LAUNCH_SITE
FROM SPACEXTBL;
```

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

Out[17]:

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

Launch Site Names Begin with 'CCA'

Task 2

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

In [114…

```
%%sql
SELECT *
FROM SPACEXTBL
WHERE LAUNCH_SITE LIKE "%CCA%"
LIMIT 5;
```

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

Out[114…

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing _Outcome
04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
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	Success	Failure (parachute)
22-05-2012	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
01-03-2013	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 [25]:

```
%%sql
SELECT SUM(PAYLOAD_MASS__KG_)
AS TOTAL_PAYLOAD_NASA
FROM SPACEXTBL
WHERE Payload LIKE '%CRS%';
```

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

Out[25]:

TOTAL_PAYLOAD_NASA
111268

Average Payload Mass by F9 v1.1

Task 4

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

In [35]:

```
%%sql
SELECT AVG(PAYLOAD_MASS__KG_)
AS AVERAGE_F9_MASS
FROM SPACEXTBL
WHERE BOOSTER_VERSION = 'F9 v1.1';
```

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

Out[35]:

AVERAGE_F9_MASS
2928.4

First Successful Ground Landing Date

Task 5

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

Hint:Use min function

In [115…

```
%%sql
SELECT MIN(DATE) AS FIRST_SUCCESS
FROM SPACEXTBL
WHERE [LANDING _OUTCOME] = 'Success (ground pad)';
```

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

Out[115…

FIRST_SUCCESS
01-05-2017

Successful Drone Ship Landing with Payload between 4000 and 6000

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 [66]:

```
%%sql
SELECT DISTINCT Booster_Version
FROM SPACEXTBL
WHERE PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000 AND [LANDING _OUTCOME] = 'Success (drone ship)'
```

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

Out[66]:

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

Total Number of Successful and Failure Mission Outcomes

Task 7

List the total number of successful and failure mission outcomes

In [77]:

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

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

Out[77]:

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

Boosters Carried Maximum Payload

Task 8

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

In [87]:
```
%%sql
SELECT BOOSTER_VERSION AS MAX_BOOSTER
FROM SPACEXTBL
WHERE PAYLOAD_MASS__KG_ == (SELECT MAX(PAYLOAD_MASS__KG_)
                            FROM SPACEXTBL);
```

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

Out[87]:

MAX_BOOSTER
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

2015 Launch Records

Task 9

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 [89]:

```
%%sql
SELECT BOOSTER_VERSION, LAUNCH_SITE
FROM SPACEXTBL
WHERE [LANDING _OUTCOME] = 'Failure (drone ship)' AND SUBSTR(DATE, 7, 4) = '2015';
```

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

Out[89]:

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

Task 10

Rank the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.

In [109…

```
%%sql
SELECT [LANDING _OUTCOME], COUNT(*) AS COUNTS
FROM SPACEXTBL
WHERE DATE BETWEEN '04-06-2010 ' AND '20-03-2017' AND [LANDING _OUTCOME] LIKE '%Success%'
GROUP BY [LANDING _OUTCOME]
ORDER BY COUNTS DESC;
```

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

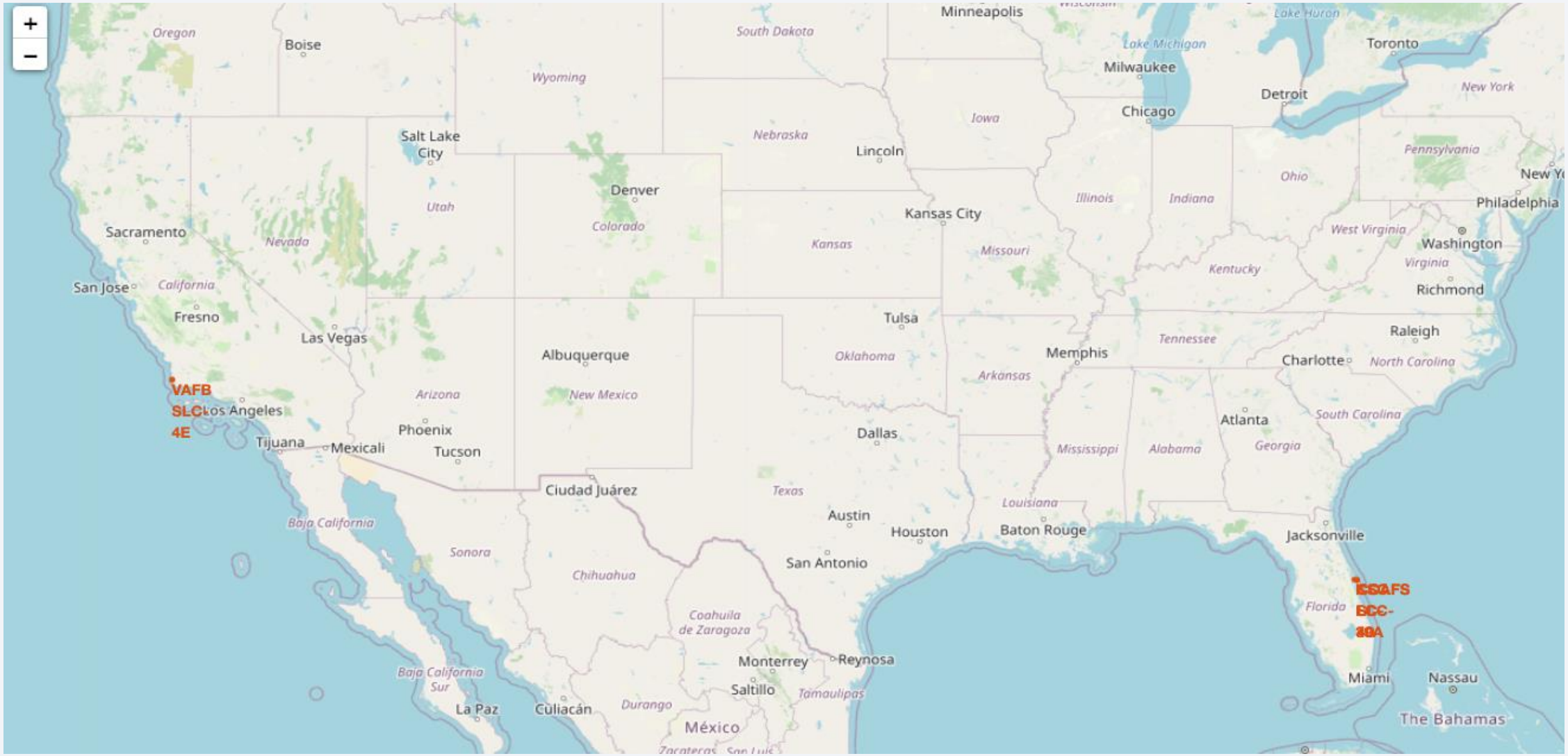
Out[109…

Landing _Outcome	COUNTS
Success	20
Success (drone ship)	8
Success (ground pad)	6

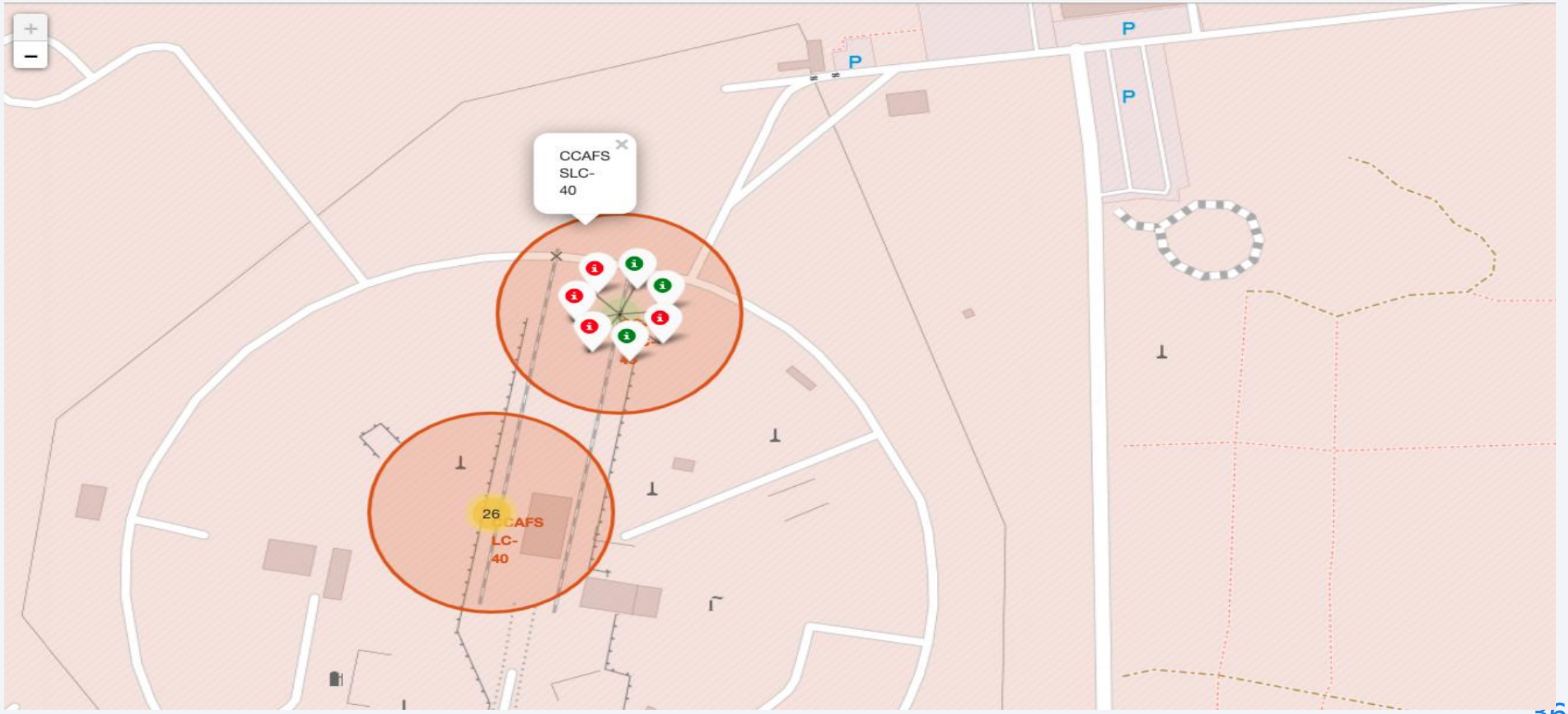
Section 3

Launch Sites Proximities Analysis

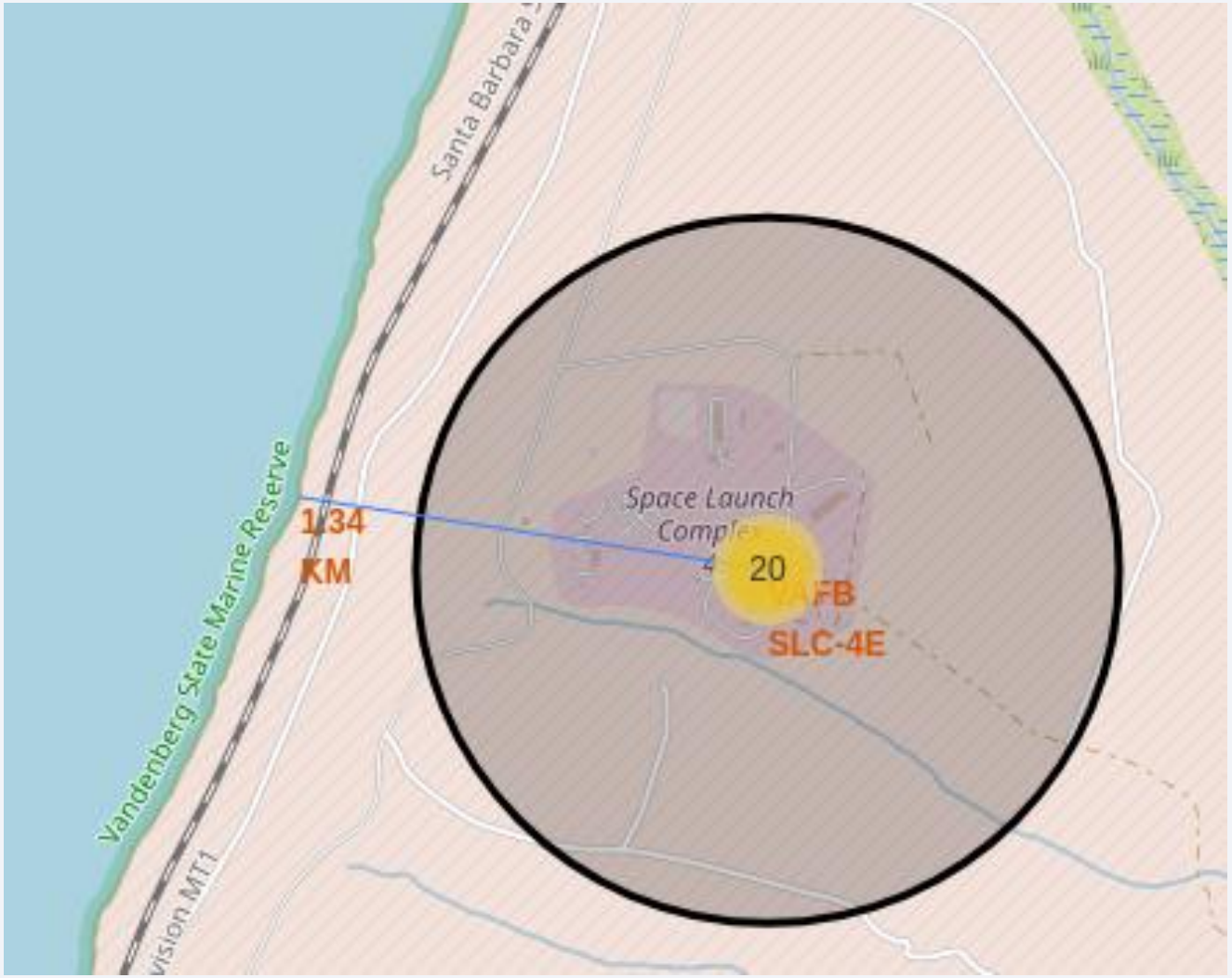
Space X global launch sites



Launching Attempts' Outcomes



Launch site proximity to the ocean

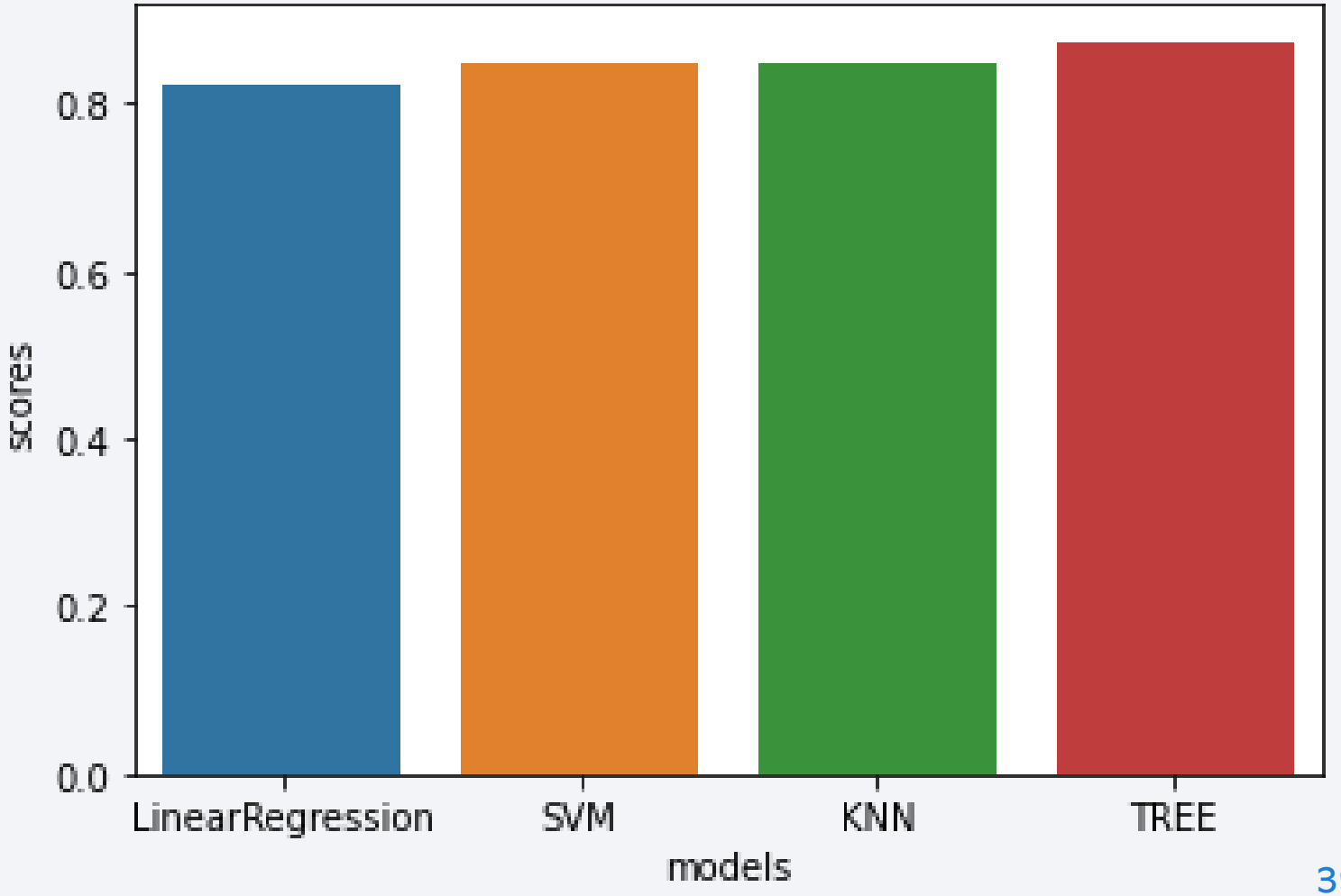


Section 5

Predictive Analysis (Classification)

Classification Accuracy

Decision Tree Classification model realizes the highest score of 87.5%.

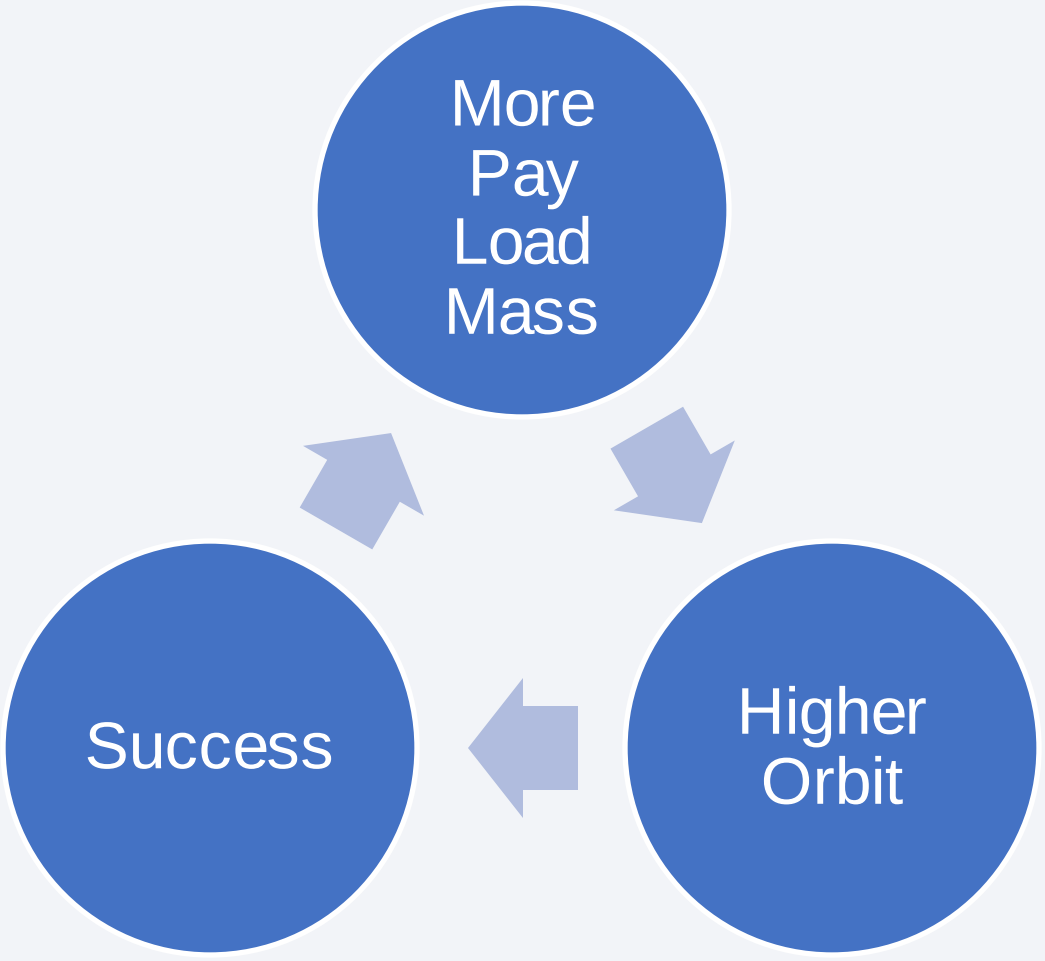


Confusion Matrix

The model successfully predicted 10 success and 3 failures.

A more attention is needed toward false positives and false negatives.

Conclusions



Appendix

GitHub repo for all the notebooks cited in the presentation:

https://github.com/mzaoualim/IBM-Data-Science-Professional-Certificate

Thank you!