Winning Space Race with Data Science

Marco Bruno

June 2022

IBM Developer
SKILLS NETWORK

Outline

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

Executive Summary

Summary of Methodologies

- Data Collection through 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

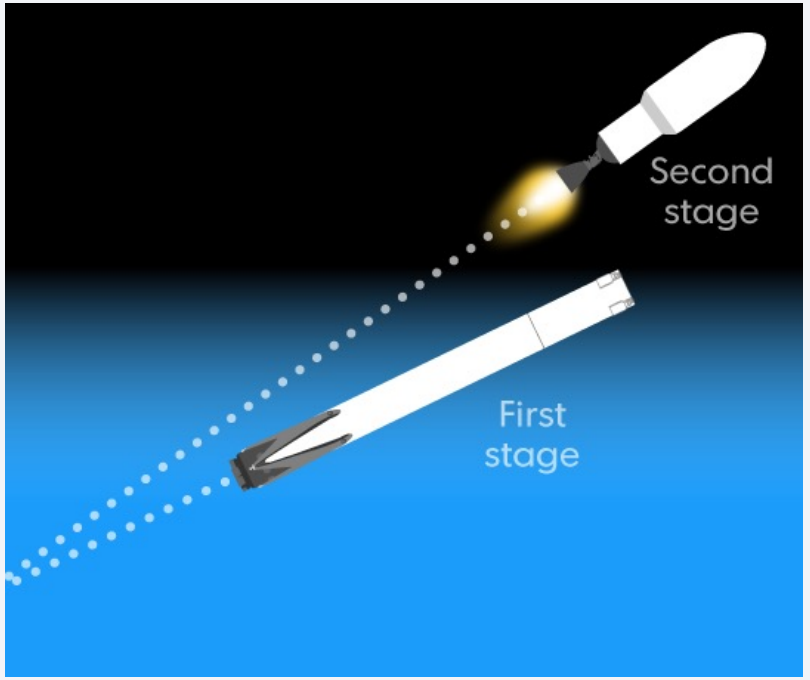
Summary of all Results

- Exploratory Data Analysis result
- Interactive analytics (screenshots)
- Predictive Analytics result from Machine Learning Lab

Introduction

Background

- Rocket launches from SpaceX are relatively inexpensive because it reuses the first stage
- Falcon 9 rocket launches cost $62 million (as advertised in SpaceX website)
- Other providers cost upwards of $165 million each
- Space Y, a new space travel startup, wishes to compete with SpaceX



Introduction

Goals and Targets

- Space Y has mandated Marco Bruno to come up with answers for a preliminary overview on whether future launches could become cheaper and thus, competing head-to-head with SpaceX
- Main objective is to determine the price of each launch by:
  - Identifying all factors that influence the landing outcome
  - The relationship between each variable and how they affect the outcome
  - The best conditions needed to increase the probability of a successful landing by the stage 1 of a rocket

Section 1

Methodology

Methodology

Executive Summary

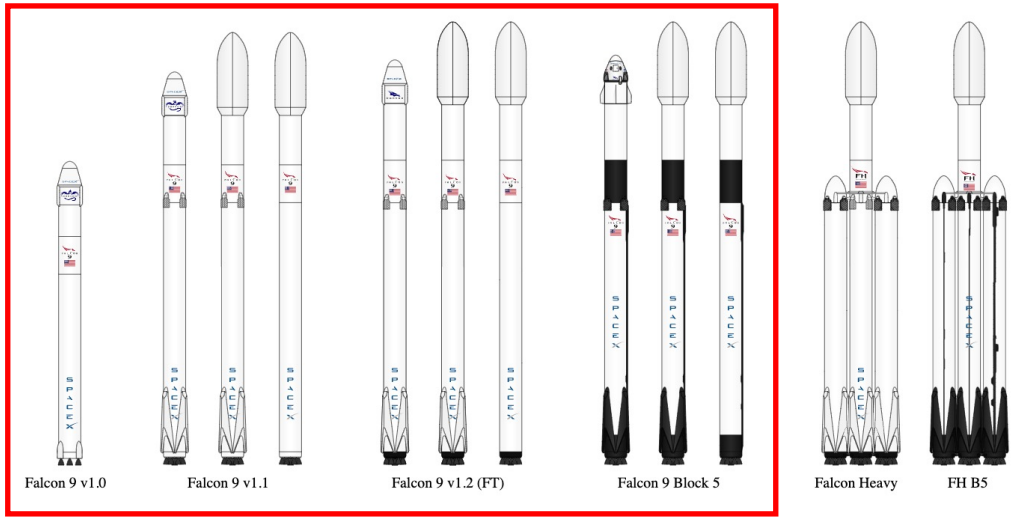
1. Data collection methodology
    - Data about launches and rockets was collected using SpaceX REST API and web scrapping from Wikipedia
2. Data Wrangling
    - Data was processed using one-hot encoding for categorical features
3. Exploratory data analysis (EDA) with was performed visualization and SQL
4. Interactive visual analytics were done with Folium and Plotly Dash
5. Predictive analysis using classification machine learning models
    - Mainly to understand how to build, tune, evaluate classification models

Data Collection

Data collection is the process of gathering and measuring information on targeted variables in an established system, which then enables one to answer relevant questions and evaluate outcomes.

Process Description

- The dataset was collected by REST API and Web Scrapping from Wikipedia
- “get request” was used for the REST API
- The response content was decoded in a JSON file and transformed into a Pandas DataFrame
- The data was then cleaned, filtered to display Falcon 9 rocket only
- BeautifulSoup was used in the web scrapping process, from identifying data tables to generating a DataFrame
- The objective was to generate a ready to use CSV file for further stages of the analysis



Data Collection – SpaceX API

Get Response from REST API → **Decode Response Into JSON file** → **Generate Dataframe** → **Data Transformation** → **Filter Data for Falcon 9 Rockets** → **Export CSV file**

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

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

	static_fire_date_utc	static_fire_date_unix	net	window	rocket	success	failures	details	crew	ships	capsules	
0	2006-03-17T00:00:00.000Z	1.142554e+09	False	0.0	5e9d0d95eda69955f709d1eb	False	[{'time': 33, 'altitude': None, 'reason': 'merlin engine failure'}]	Engine failure at 33 seconds and loss of vehicle	[]	[]	[]	[5eb0e4b5b6c3bb000
1	None	NaN	False	0.0	5e9d0d95eda69955f709d1eb	False	[{'time': 301, 'altitude': 289, 'reason': 'harmonic oscillation leading to premature engine shutdown'}]	Successful first stage burn and transition to second stage, maximum altitude 289 km, Premature engine shutdown at T+7 min 30 s, Failed to reach orbit, Failed to recover first stage	[]	[]	[]	[5eb0e4b6b6c3bb000

```
# Hint data['BoosterVersion']!='Falcon 1'
data_falcon9 = data[data['BoosterVersion']!='Falcon 1']
data_falcon9.head()
```

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	Longit
4	6	2010-06-04	Falcon 9	NaN	LEO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0003	-80.577
5	8	2012-05-22	Falcon 9	525.0	LEO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0005	-80.577
6	10	2013-03-01	Falcon 9	677.0	ISS	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0007	-80.577
7	11	2013-09-29	Falcon 9	500.0	PO	VAFB SLC 4E	False Ocean	1	False	False	False	None	1.0	0	B1003	-120.610
8	12	2013-12-03	Falcon 9	3170.0	GTO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B1004	-80.577

[Data Collection Space X API in Github]

Data Collection – Scraping from Wikipedia

Get Response from HTML → Generate BeautifulSoup Object → Identify Data Tables → Generate Dictionaries → Conversion to DataFrames → Export CSV file

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

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


# Use soup.title attribute


# 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('tr')
html_tables
```

```
headings = []
for key,values in dict(launch_dict).items():
    if key not in headings:
        headings.append(key)
    if values is None:
        del launch_dict[key]


def pad_dict_list(dict_list, padel):
    lmax = 0
    for lname in dict_list.keys():
        lmax = max(lmax, len(dict_list[lname]))
    for lname in dict_list.keys():
        ll = len(dict_list[lname])
        if  ll <lmax:
            dict_list[lname] += [padel] * (lmax - ll)
    return dict_list

pad_dict_list(launch_dict,0)

df=pd.DataFrame(launch_dict)
df.tail()
```

	Flight No.	Launch site	Payload	Payload mass	Orbit	Customer	Launch outcome	Version Booster	Booster landing	Date	Time
328	0	CCSFS	Starlink	15,600 kg	LEO	[SpaceX]	Success\n	F9 B5B1051.10	0	9 May 2021	06:42
329	0	KSC	Starlink	~14,000 kg	LEO	[NASA]	Success\n	F9 B5B1058.8	0	15 May 2021	22:56
330	0	CCSFS	Starlink	15,600 kg	LEO	[Sirius XM]	Success\n	F9 B5B1063.2	0	26 May 2021	18:59
331	0	KSC	SpaceX CRS-22	3,328 kg	LEO	0	0	F9 B5B1067.1	0	3 June 2021	17:29
332	0	CCSFS	SXM-8	7,000 kg	GTO	0	0	F9 B5	0	6 June 2021	04:26

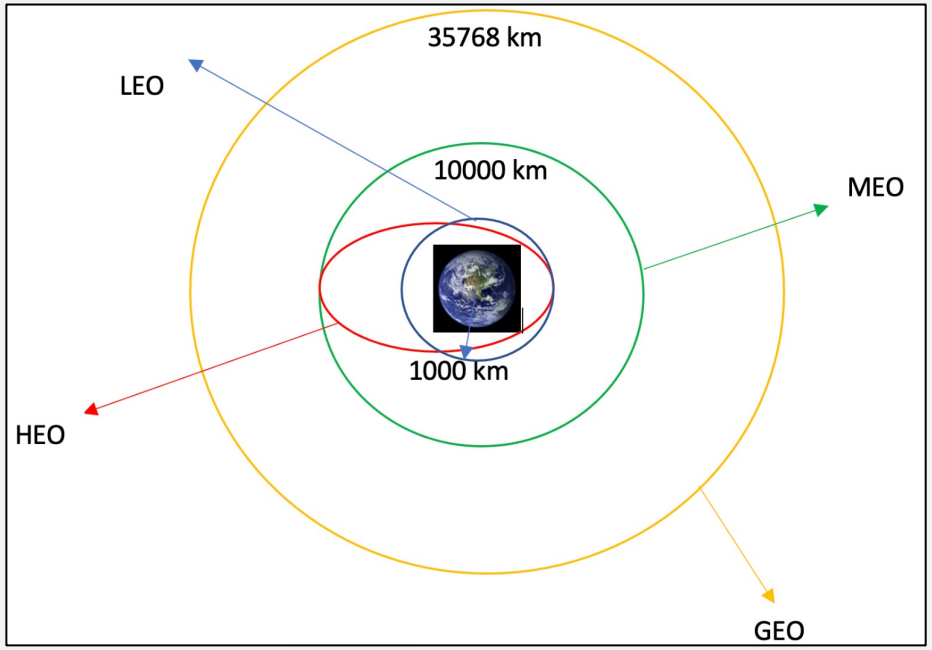
[Data Scrapping from Wikipedia in Github]

Data Wrangling

Data Wrangling is the process of cleaning and unifying messy and complex data sets for easy access and Exploratory Data Analysis (EDA).

Objective

- To perform some Exploratory Data Analysis (EDA) to find some patterns in the data and determine what would be the label for training supervised models
- Each launch aims to an dedicated orbit. The adjacent figure illustrates precisely that
- It's our duty to also understand the economical outcomes from each launch site and successful rates of landing the rocket's phase 1



Data Wrangling

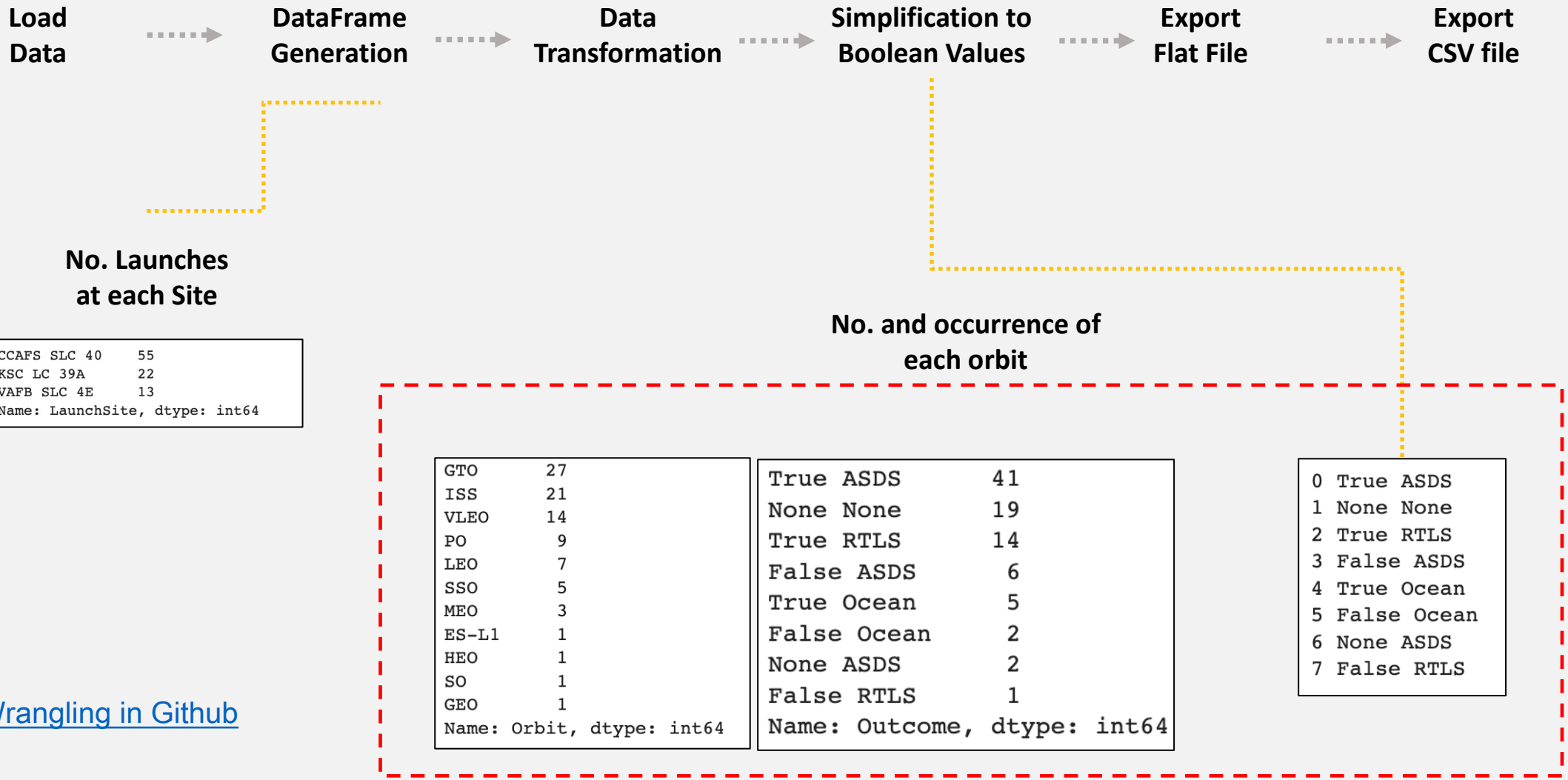
Process Description

- We calculated the number of launches on each site
- Then, the number and occurrence of mission outcome per orbit type was found
- We generated a landing outcome label from the outcome column with Boolean values where ”1” is successful landings and “0” is unsuccessful ones. This makes it easier for further analysis, visualization, and Machine Learning
- The product was exported to a CSV file

NOTE:

There are several different cases where the booster did not land successfully. Sometimes a landing was attempted but failed due to an accident; for example, True Ocean means the mission outcome was successfully landed to a specific region of the ocean while False Ocean means the mission outcome was unsuccessfully landed to a specific region of the ocean. True RTLS means the mission outcome was successfully landed to a ground pad False RTLS means the mission outcome was unsuccessfully landed to a ground pad. True ASDS means the mission outcome was successfully landed on a drone ship False ASDS means the mission outcome was unsuccessfully landed on a drone ship.

Data Wrangling



[Data Wrangling in Github]

EDA

Exploratory Data Analysis is an approach of analysing data sets to summarize their main characteristics, using statistical graphics and other data visualization methods

Objective

- Predict if the Falcon 9 first stage lands successfully assessing the relationship between different variables upon a rocket launch
- Perform queries via SQL in tables to understand what data we have and prepare for other transformations
- Create different visualizations for further deeper analysis by another team

EDA with SQL

SQL is the standard querying language for all the relational databases. It is also the standard for the current big data platforms that use SQL as their key API for their relational databases. A Data Scientist needs SQL in order to handle structured data. This structured data is stored in relational databases. Therefore, in order to query these databases, a data scientist must have a sound knowledge of SQL.

Types of Queries performed

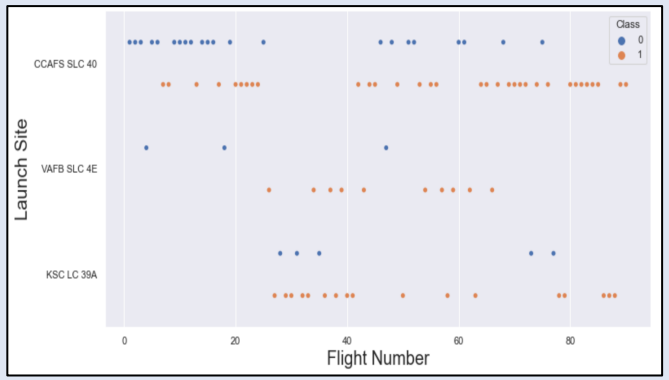
- Clean-up and data type transformations after uploading data in the database
- Querying unique names and values
- Calculating totals
- Calculating totals with unique values
- Several listings to understand number of successful landings, failures and others
- For more info, click below on the link to see the entire set of queries performed

EDA using SQL in Github

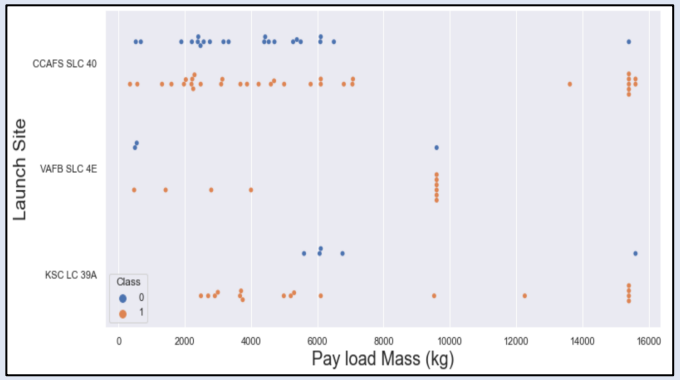
EDA with Data Visualization

Scatter Graphs

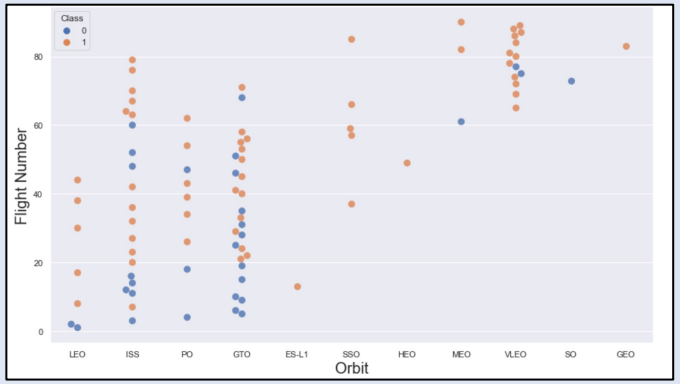
Relationship between Flight Number and Launch Site



Relationship between Pay Load and Launch Site

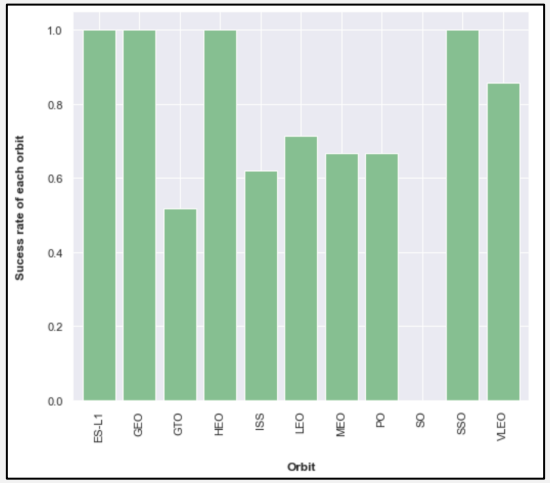


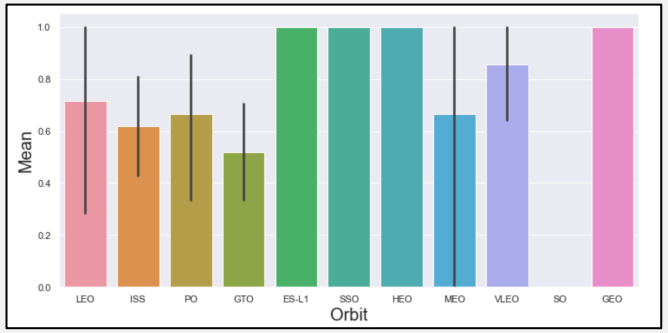
Relationship between Flight Number and Orbit Type



Histograms

Relationship between Success Rate and Orbit Type





[EDA Data Visuals in Github]

Build an Interactive Map with Folium

Folium is a plugin used in Python to create interactive maps. It's required real coordinates (latitude and longitude) to visualise launch sites, create markers and apply conditions and relationships between data points

Objective

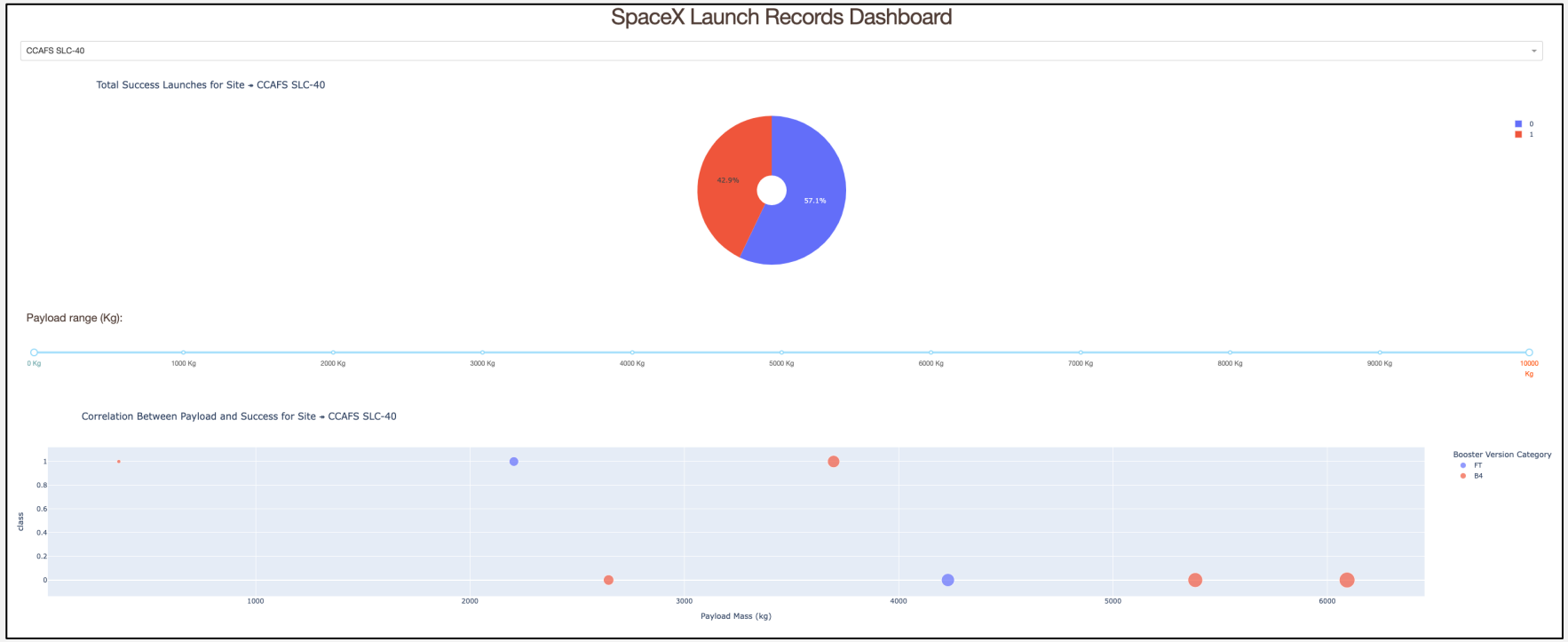
- Mark all launch sites on a map
- Mark the success/failed launches for each site on the map
- Calculate the distances between a launch site to its proximities
- Identify geographical patterns about launch sites

Map produced with Folium in Github

Build a Dashboard with Plotly Dash

With Python, it’s possible to create fast, on-demand and ready to use dashboards using Plotly Dash and publish it in your web browser instantly

Objective (same as Folium section)



[Dashboard in Github]

Predictive Analysis (Classification)

Predictive analytics is a branch of advanced analytics in Machine Learning that makes predictions about future outcomes using historical data combined with statistical modelling, data mining techniques and machine learning. Classification models work by categorising information based on historical data.

Objective

- Perform exploratory Data Analysis and determine Training Labels
- Create a column for the class
- Standardize the data
- Split into training data and test data sets
- Find best Hyperparameter for SVM, Classification Trees and Logistic Regression
- Find the method performs best using test data

Predictive Analysis (Classification)

Building the Model

- Load data and transform it into dataframes
- Transform dataframes into NumPy arrays and standardize it
- Split arrays in training and test data sets
- Select Machine Learning algorithms to be used
- Set paramaters and algos to GridSearchCV
- Fit datasets into objects and train model

Model Evaluation

- Check accuracy of each model
- Get best hyperparameters for each type of alogrithm
- Plot Confusion Matrix

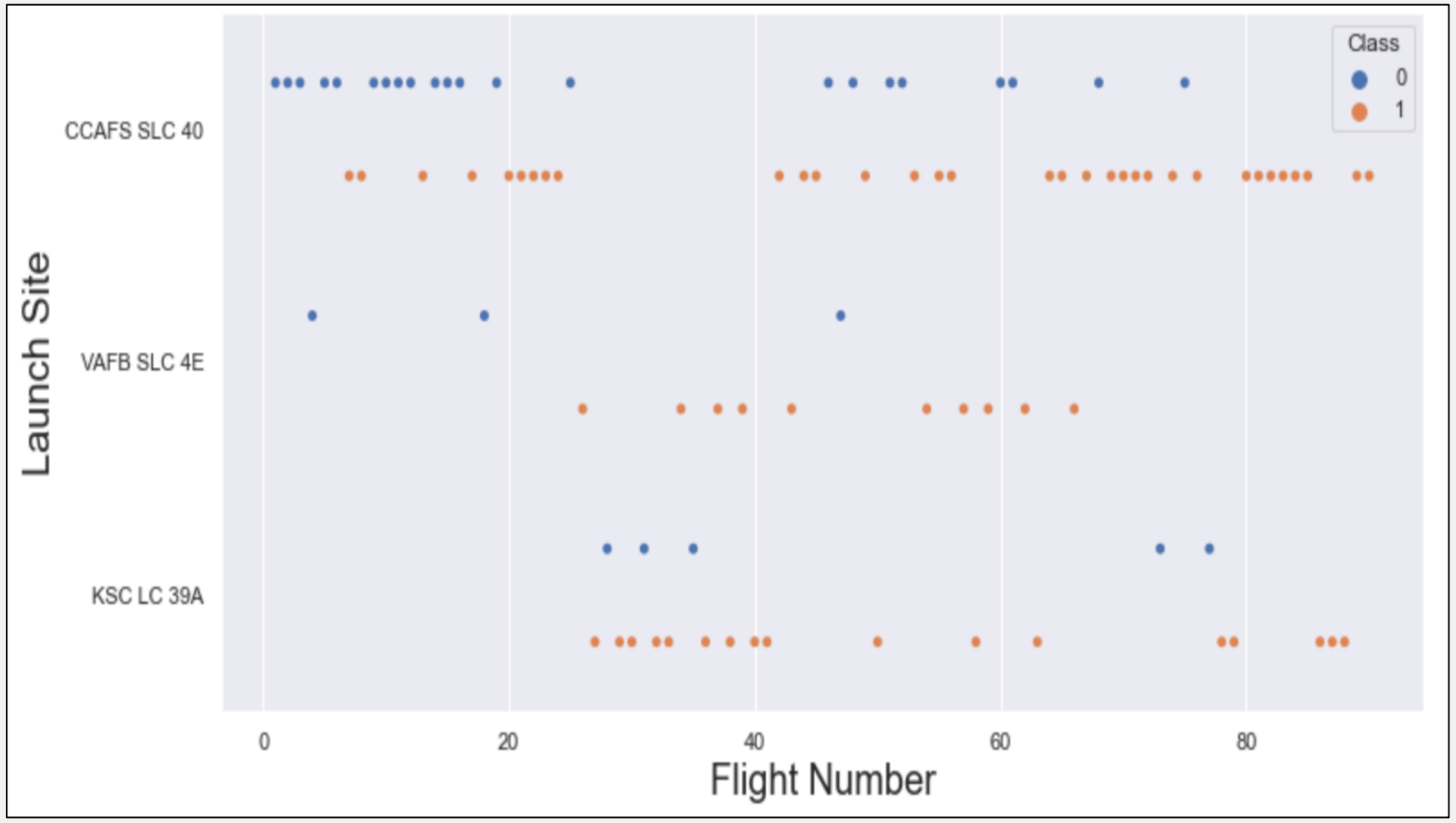
Best Performing Classification Model Identification

- Selection of the best performing model after adequate iterations

Section 2

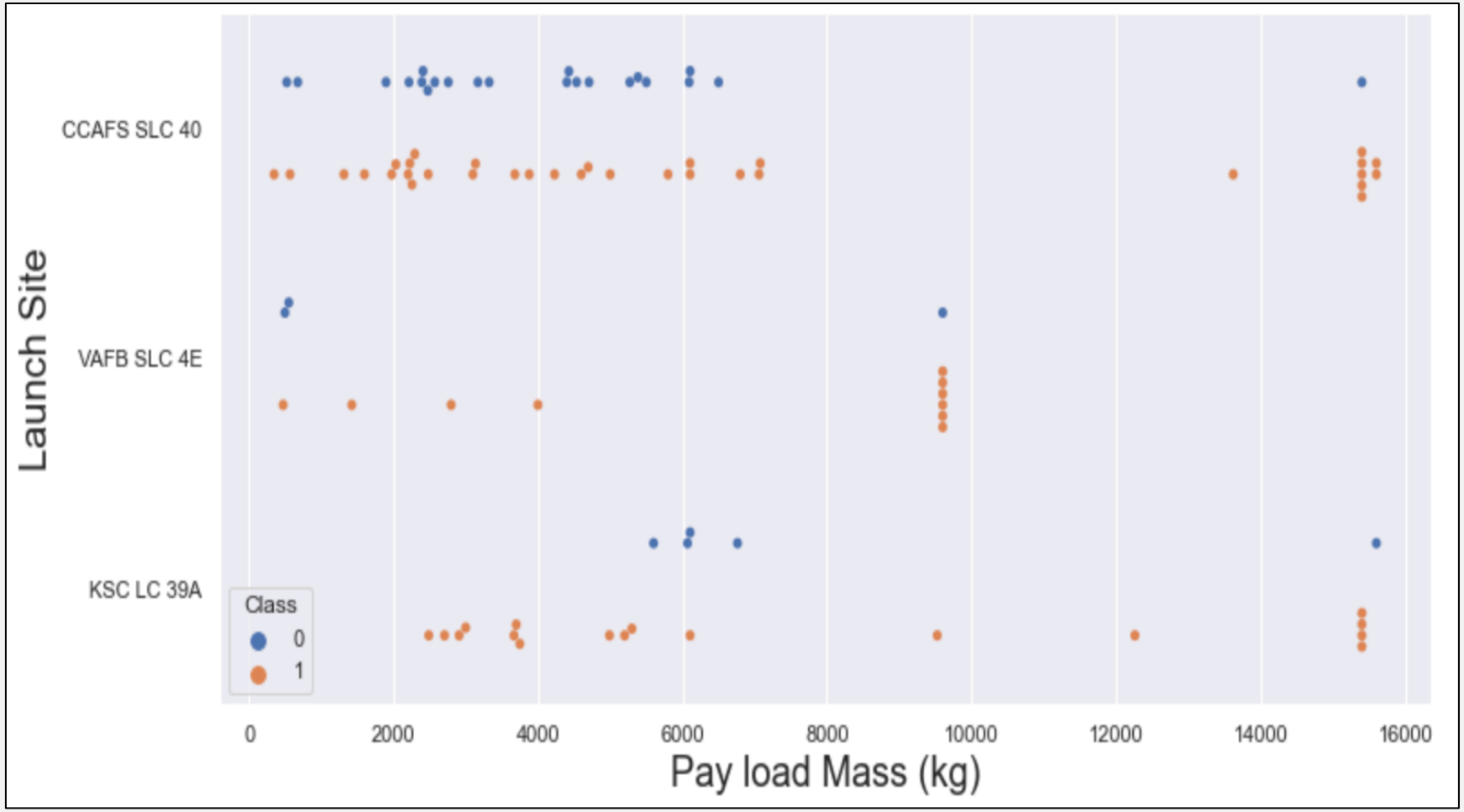
Insights drawn from EDA

Flight Number vs. Launch Site



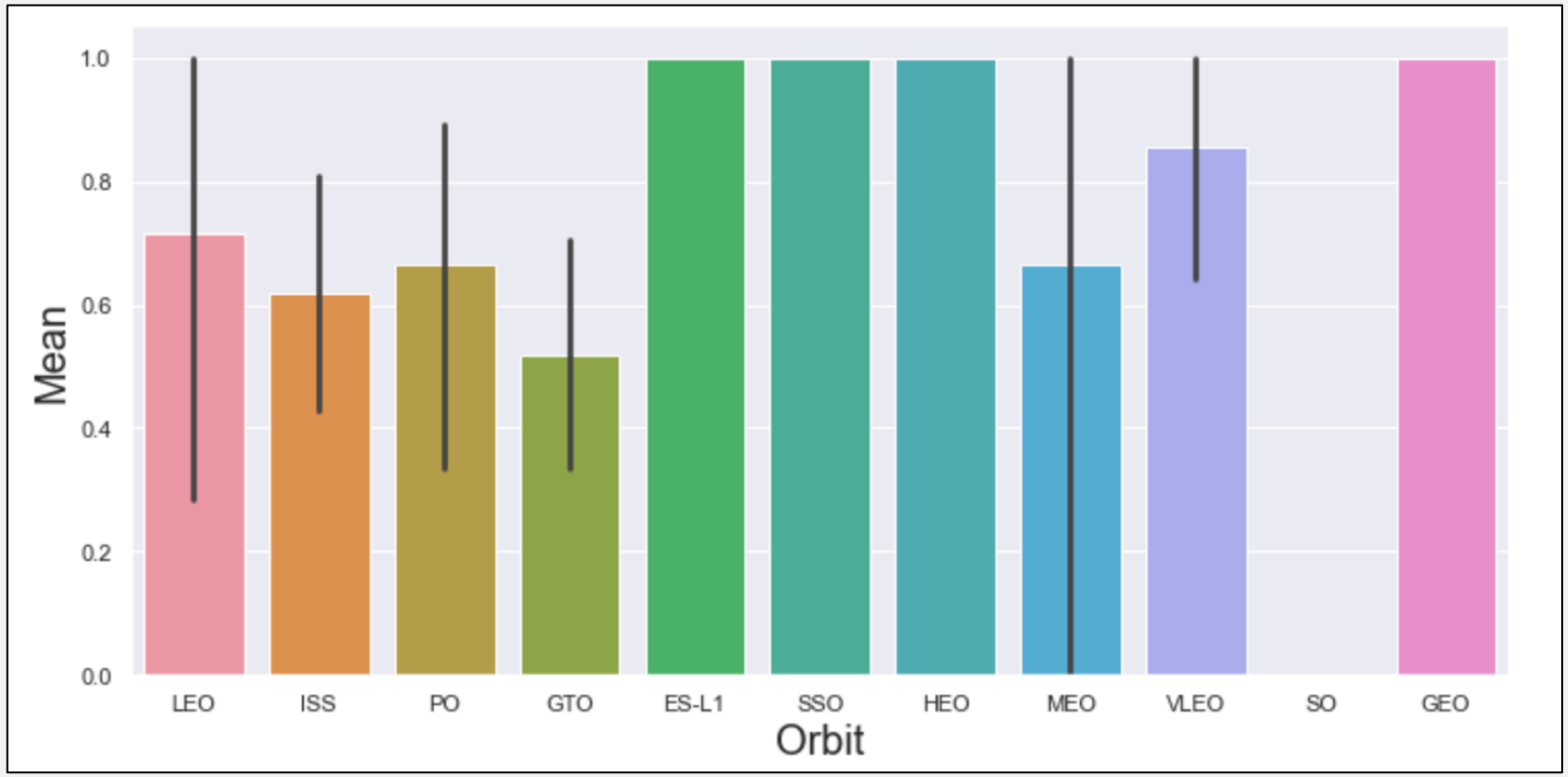
- The success rate increases with the number of flights but no clear pattern is visible

Payload vs. Launch Site



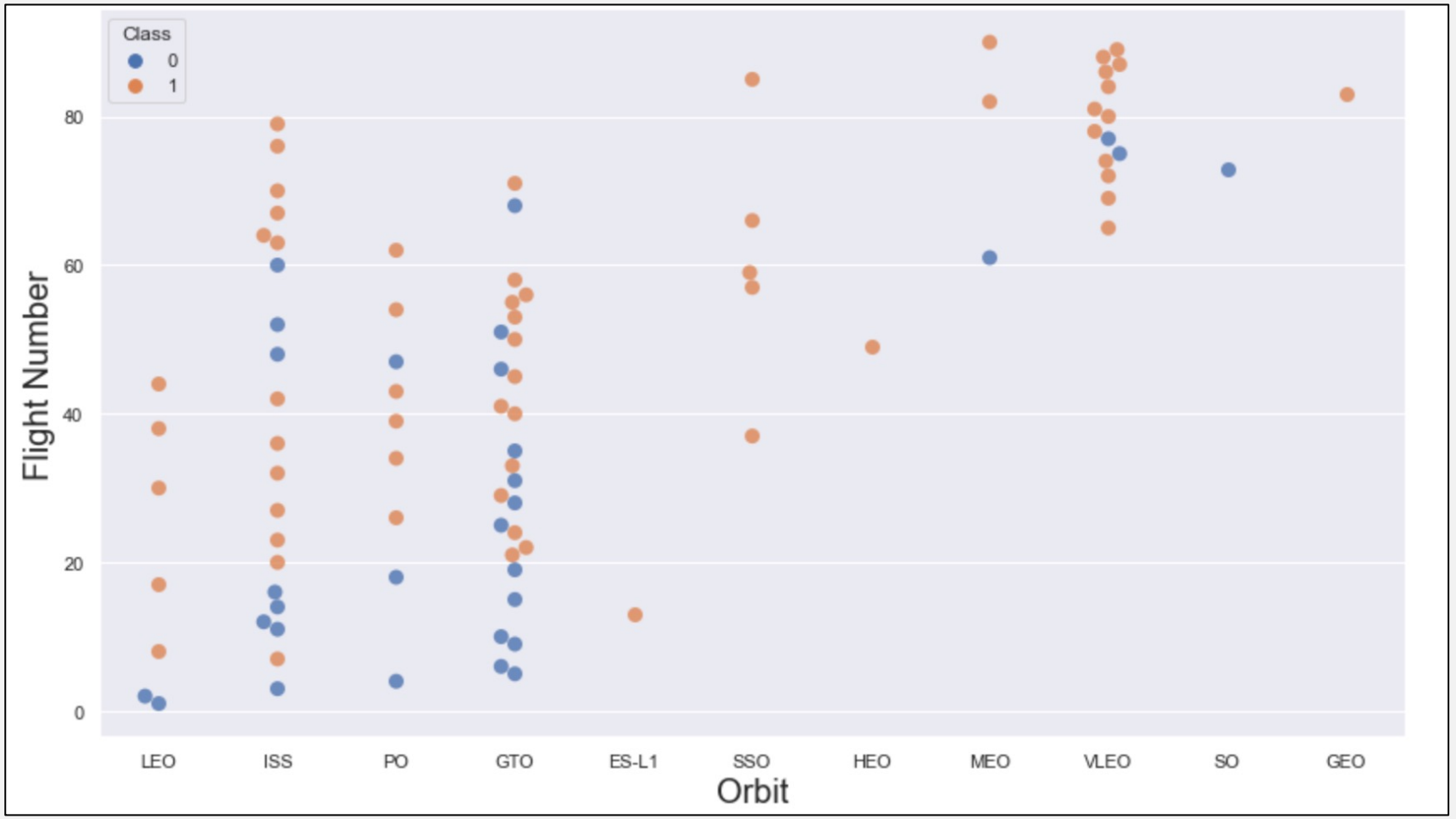
- The success rate of a launch increases is greater than of around 7000 kg. No clear patter for decision making

Success Rate vs. Orbit Type



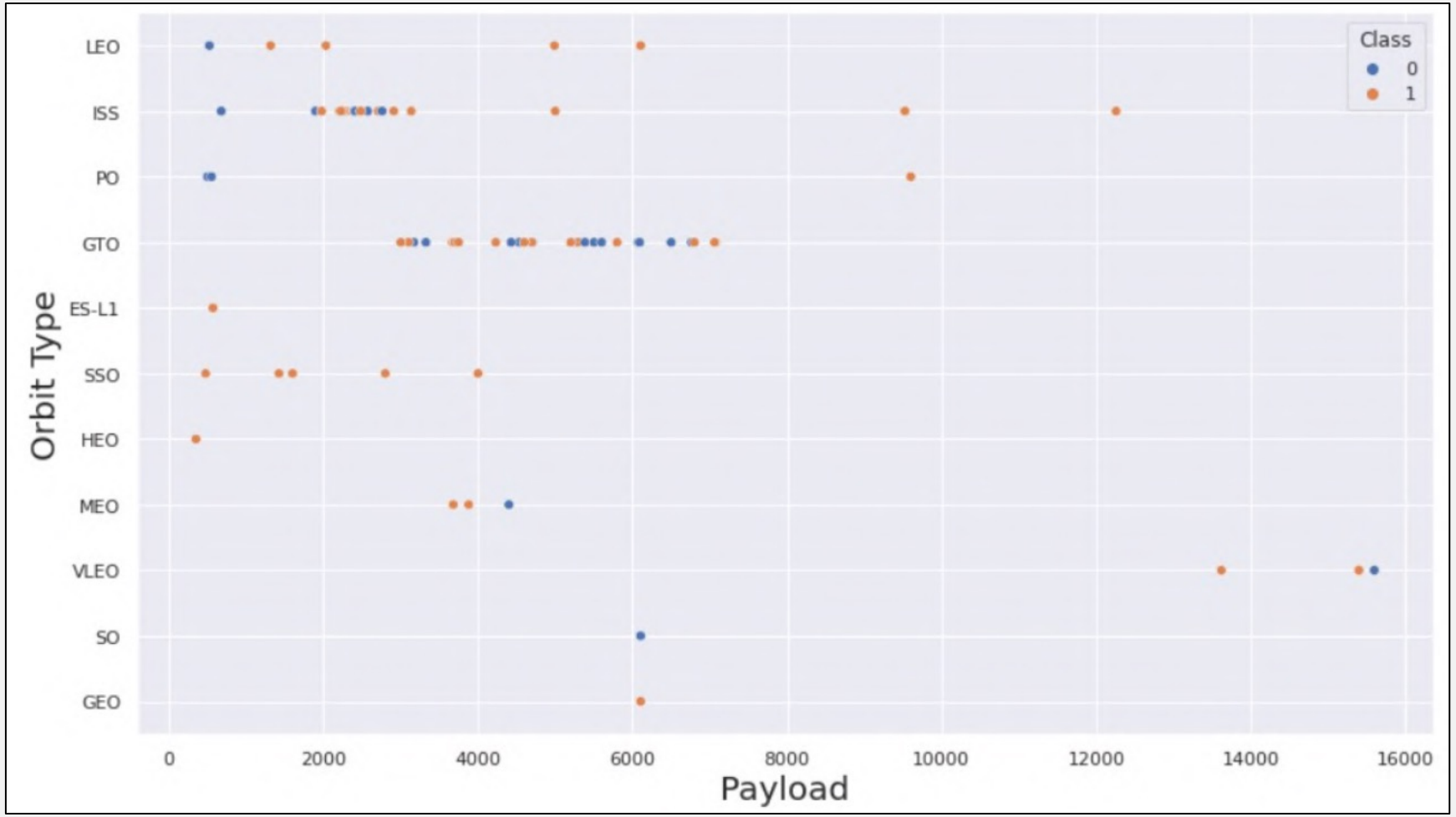
- The success rate of a launch increases is clearly visible for the orbit tapes ES-L1, SSO, HEO and GEO
- We don’t recommend the other orbits

Flight Number vs. Orbit Type



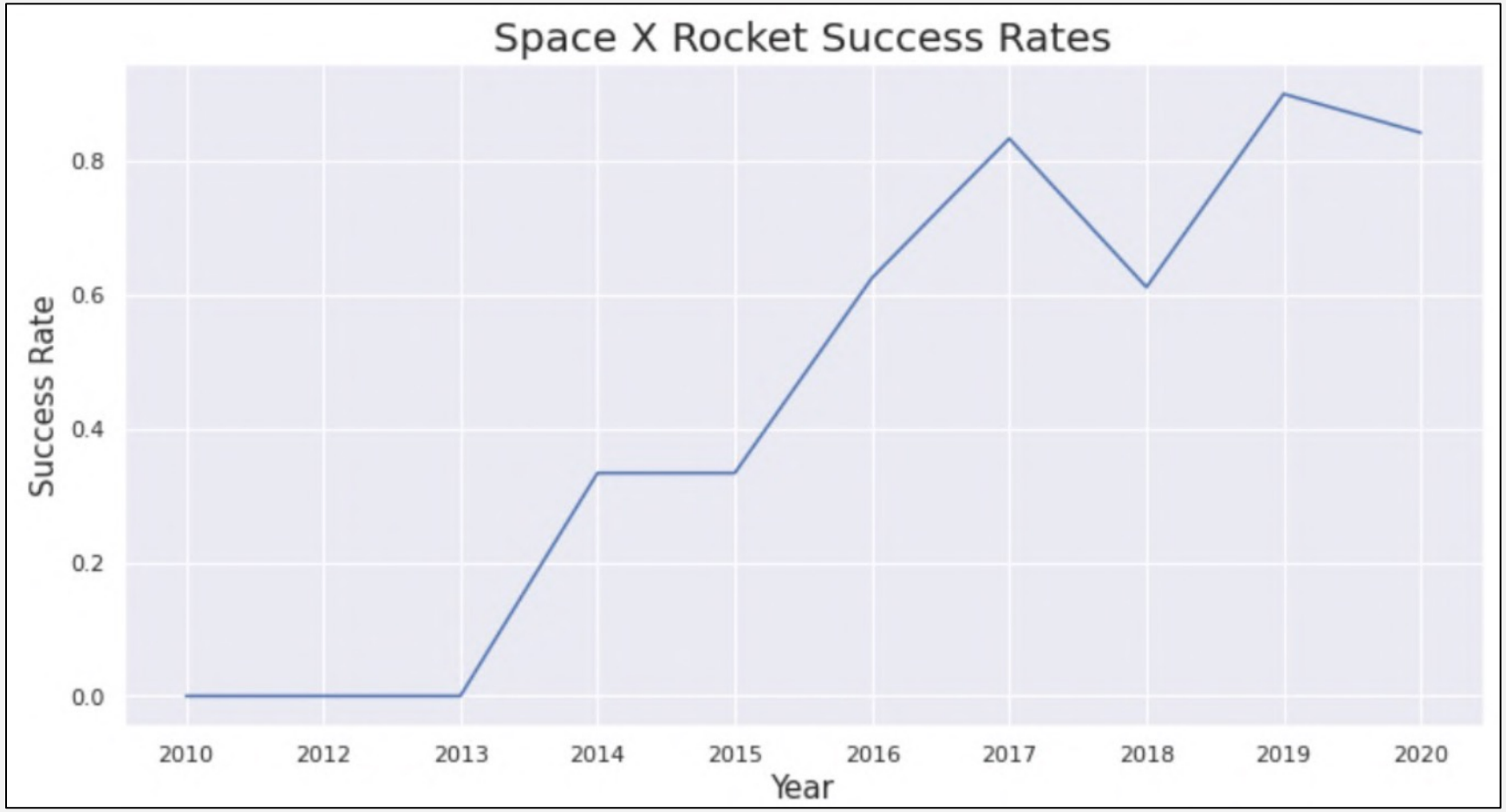
- The scatter plot shows that in general, the bigger the number of flights on each orbit, the bigger is the success rate – mainly for LEO orbit

Payload vs. Orbit Type



- Payload has an negative impact mainly on GTO
- We identify that LEO and ISS orbits have a decent rate of success related to the rocket's payload

Launch Success Yearly Trend



- The success rate of launches performed by SpaceX have increased over time
- Despite a negative impact around 2018, we foresee a positive trend beyond 2020

All Launch Site Names

- Find the names of the unique launch sites

```
%sql SELECT DISTINCT LAUNCH_SITE as "Launch_Sites" FROM SPACEX;
```

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

Launch Site Names Begin with 'CCA'

	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

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) AS "Total Payload Mass by NASA (CRS)
```

Total Payload Mass by NASA (CRS)
45596

Average Payload Mass by F9 v1.1

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) AS "Average Payload Mass by Booster
WHERE BOOSTER_VERSION = 'F9 v1.1';
```

Average Payload Mass by Booster Version F9 v1.1
2928

First Successful Ground Landing Date

```
%sql SELECT MIN(DATE) AS "First Succesful Landing Outcome in Ground Pad
WHERE LANDING__OUTCOME = 'Success (ground pad)';
```

First Succesful Landing Outcome in Ground Pad
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%sql SELECT BOOSTER_VERSION FROM SPACEX 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

```
%sql SELECT COUNT(MISSION_OUTCOME) AS "Successful Mission" FROM SPACEX WHERE MISSION_OUTCOME LIKE 'Success%';
```

Successful Mission
100

```
%sql SELECT COUNT(MISSION_OUTCOME) AS "Failure Mission" FROM SPACEX WHERE MISSION_OUTCOME LIKE 'Failure%';
```

Failure Mission
1

Boosters Carried Maximum Payload

```
%sql SELECT DISTINCT BOOSTER_VERSION AS "Booster Versions which carried the Maximum Payload Mass" FROM SPACEX
WHERE PAYLOAD_MASS__KG_ =(SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEX);
```

Booster Versions which carried the Maximum Payload Mass
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

```
%sql SELECT BOOSTER_VERSION, LAUNCH_SITE FROM SPACEX WHERE DATE LIKE '2015-%' AND \
LANDING__OUTCOME = 'Failure (drone ship)';
```

booster_version	launch_site
F9 v1.1 B1012	CCAFS LC-40
F9 v1.1 B1015	CCAFS LC-40

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

```
%sql SELECT LANDING__OUTCOME as "Landing Outcome", COUNT(LANDING__OUTCOME) AS "Total Count" FROM SPACEX \
WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' \
GROUP BY  LANDING__OUTCOME \
ORDER BY COUNT(LANDING__OUTCOME) DESC ;
```

Landing Outcome	Total Count
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

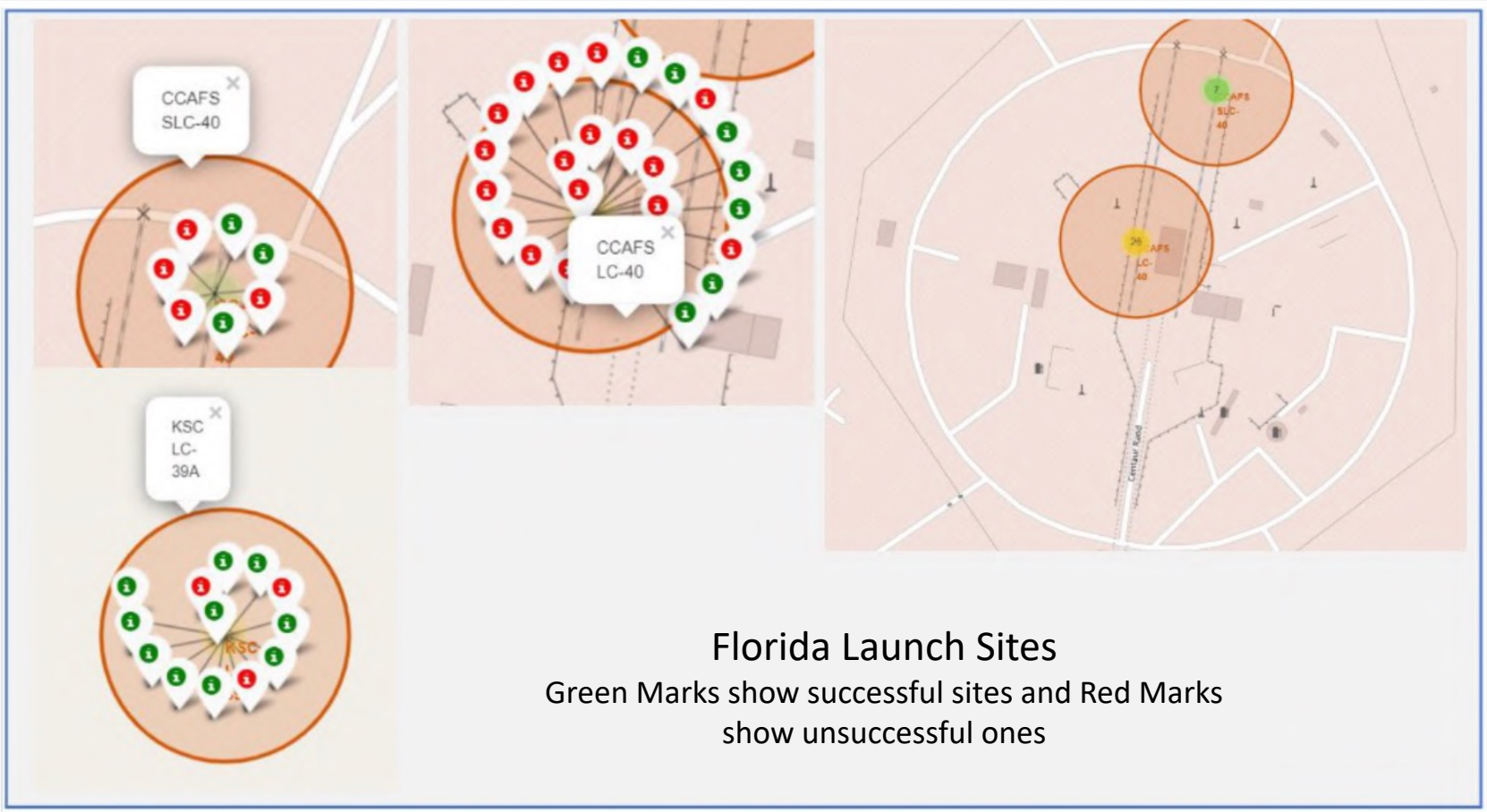
Section 3

Launch Sites Proximities Analysis

Location of the Launch Sites

- Launch sites are located in the USA, either in California or Florida

Markers showing Launch Sites



Florida Launch Sites

Green Marks show successful sites and Red Marks show unsuccessful ones

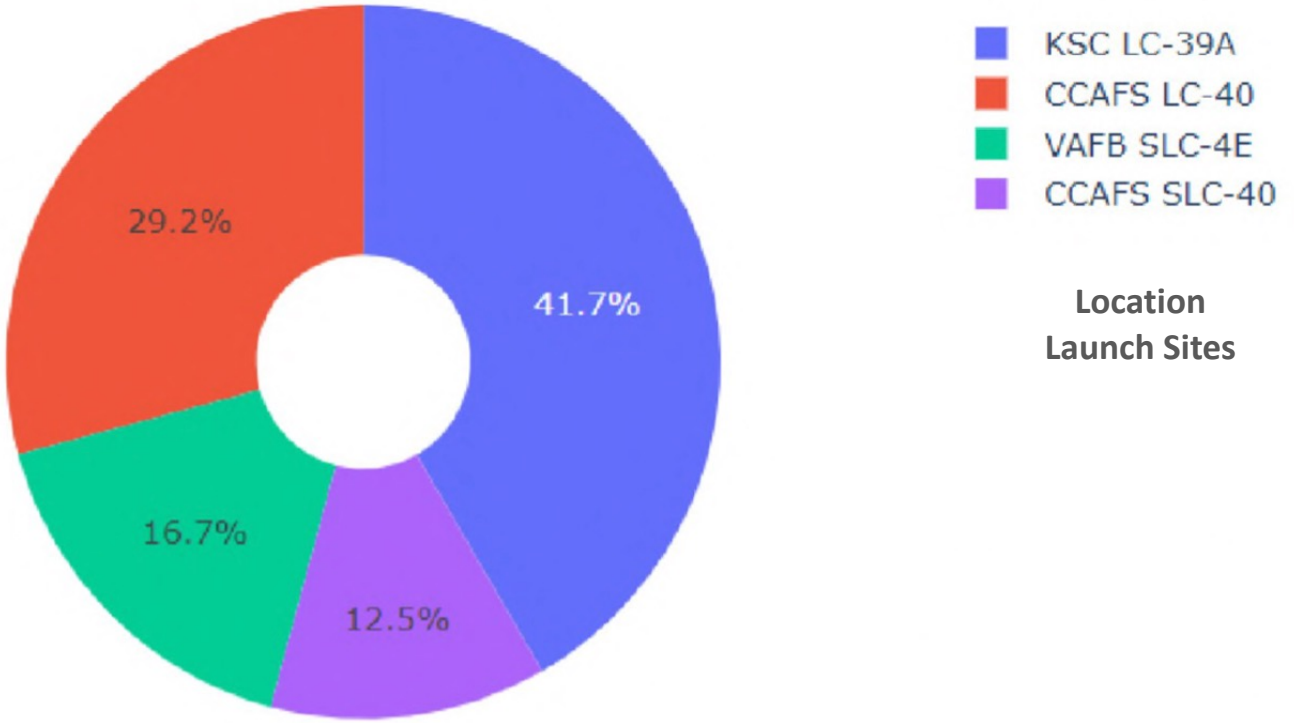
Cali. Launch Sites

Green Marks show successful sites and Red Marks show unsuccessful ones

Section 4

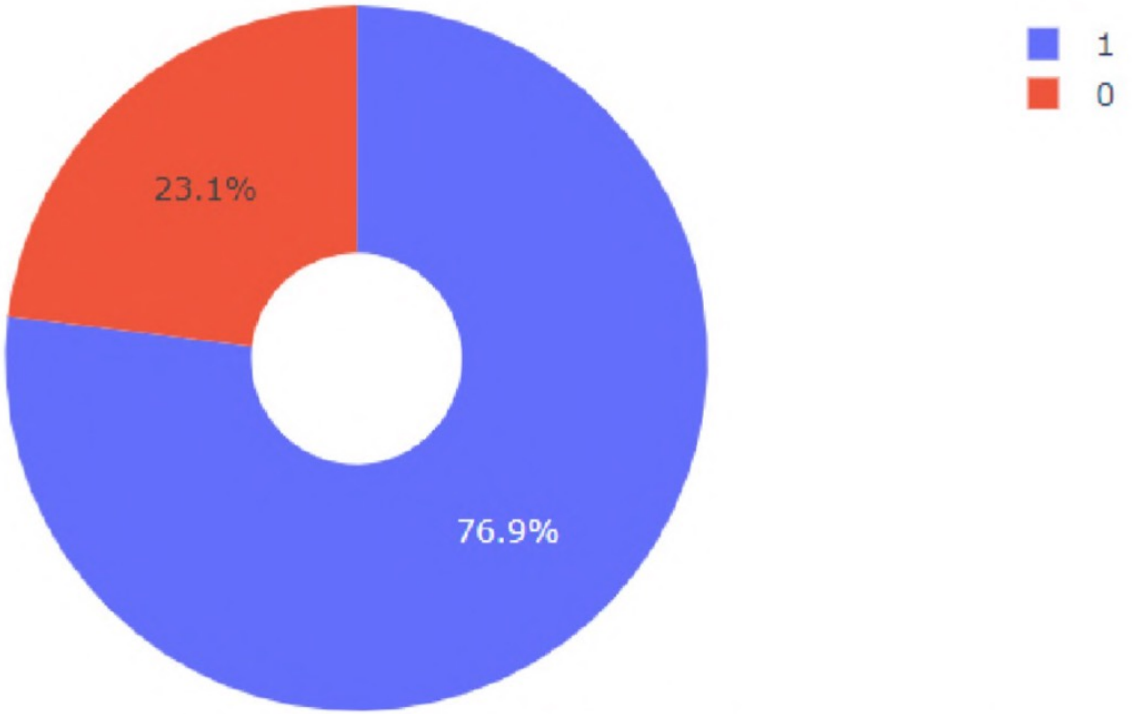
Dashboards With Plotly Dash

Success Rate by Launch Site



- KSC LC-39A has been the most successful launch site of SpaceX

Highest Launch-Success ratio



- KSC LC-39A has achieved a 76.9% success rate to date

Section 5

Predictive Analysis (Classification)

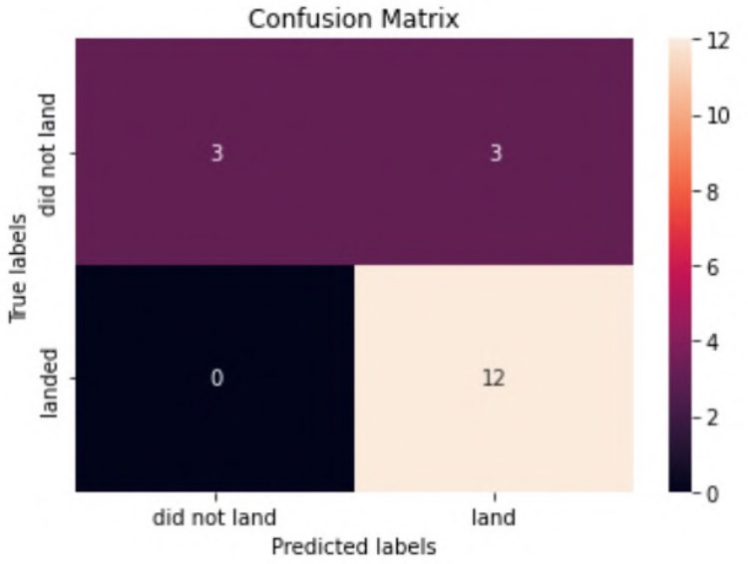
Classification Accuracy

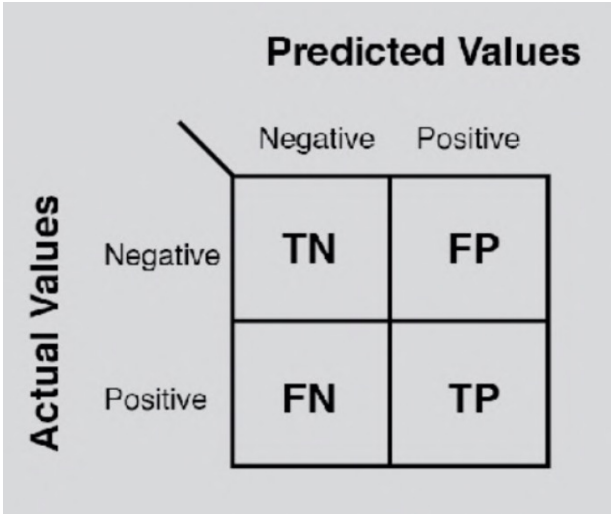
```
algorithms = {'KNN':knn_cv.best_score_,'Tree':tree_cv.best_score_,'LogisticRegression':logreg_cv.best_score_}
bestalgorithm = max(algorithms, key=algorithms.get)
print('Best Algorithm is',bestalgorithm,'with a score of',algorithms[bestalgorithm])
if bestalgorithm == 'Tree':
    print('Best Params is :',tree_cv.best_params_)
if bestalgorithm == 'KNN':
    print('Best Params is :',knn_cv.best_params_)
if bestalgorithm == 'LogisticRegression':
    print('Best Params is :',logreg_cv.best_params_)
```

```
Best Algorithm is Tree with a score of 0.9017857142857142
Best Params is : {'criterion': 'entropy', 'max_depth': 10, 'max_features': 'auto', 'min_samples_leaf': 2, 'min_sampl
es_split': 10, 'splitter': 'random'}
```

- Based on our classification model, we have identified the Tree Algorithm as the most accurate model with an accuracy score of 0.90

Confusion Matrix





- The confusion matrix for the decision tree classifier shows that the classifier can distinguish between the different classes. The major problem is the false positives .i.e., unsuccessful landing marked as successful landing by the classifier.

Conclusions

- The Tree Classifier Algorithm is the best Machine Learning approach for this dataset as it provides a better accuracy
- The success rate for SpaceX launches has increased since 2013
- KSC LC-39A have the most successful launches of any sites; 76.9%
- Orbits ES-L1, GEO, HEO, SSO launch sites have higher success rates
- KSC LC-39A had the most successful launches but increasing payload generates a negative impact on the success launch rate


Thank you