IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Yosua Daniel Chandra
November, 6 2022

Outline

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

Executive Summary

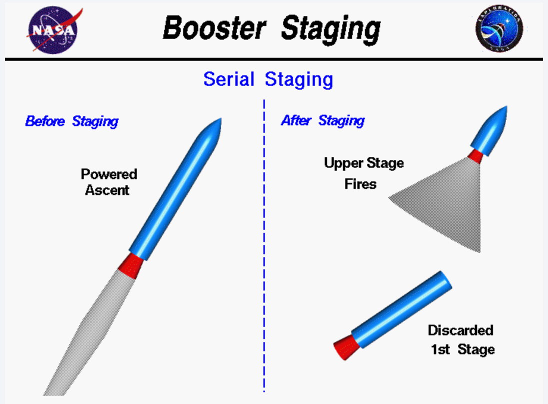
- Summary of methodologies
- Summary of all results

Introduction

- SpaceY is an an spacecraft manufacturer, Founded by Alan Musk with the goal of reducing space transportation costs to enable the colonization of Mars
- SpaceY Falcon 9 using reusable 1st stage to reduce space transportation cost, sometimes the first stage does not land or crash

Based on that we conclude two question:

- What factors determine if the rocket will land successfully?
- The interaction amongst various features that determine the success rate of a successful landing.
- What operating conditions needs to be in place to ensure a successful landing program



Section 1

Methodology

Methodology

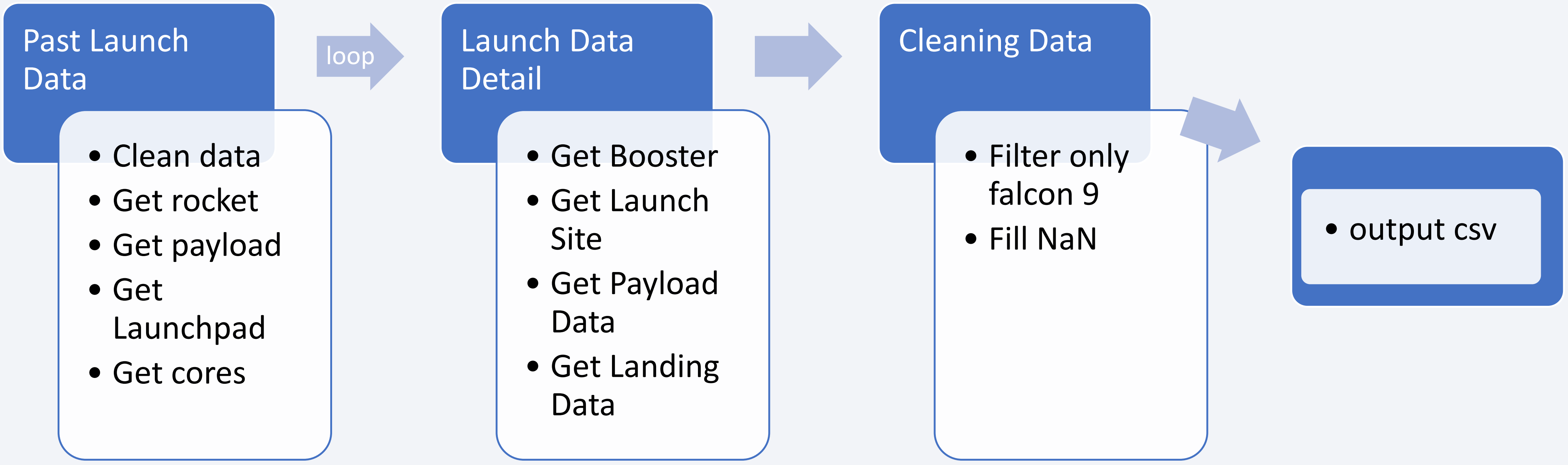
Executive Summary

- Data collection methodology:
  - Data source we used in this research is [SpaceX API] and [Wikipedia Falcon 9] launches
- Perform data wrangling
  - 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
  - How to build, tune, evaluate classification models

Data Collection

- Describe how data sets were collected.
- Data collection was done using get request to the SpaceX API.
- Next, we decoded the response content as a Json using .json() function call and turn it into a pandas dataframe using .json_normalize().
- We then cleaned the data, checked for missing values and fill in missing values where necessary.
- In addition, we performed web scraping from Wikipedia for Falcon 9 launch records with BeautifulSoup.
- The objective was to extract the launch records as HTML table, parse the table and convert it to a pandas dataframe for future analysis.

Data Collection - SpaceX API



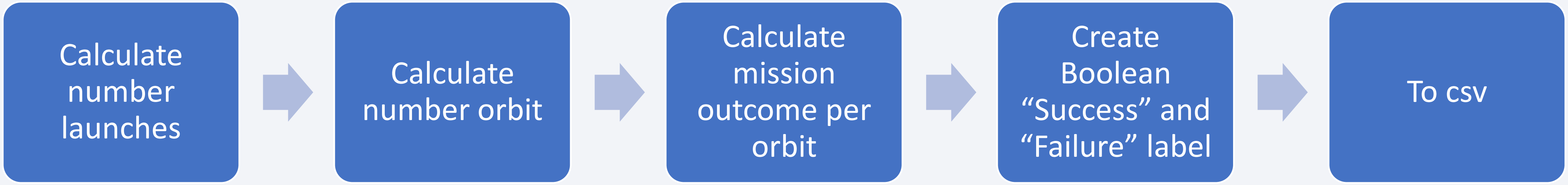
- We gather data from SpaceX API, clean and did formatting and data wrangling https://github.com/yosuadc3/ibmcourse/blob/master/Final%20Project%20SpaceX/jupyter-labs-spacex-data-collection-api.ipynb

Data Collection - Scraping

- We parsed Falcon 9 launch record from wikipedia

https://github.com/yosuadc3/ibmcourse/blob/master/Final%20Project%20SpaceX/jupyter-labs-webscraping.ipynb

Data Wrangling



- Perform EDA and determine training label
- Calculating number of launches and labeling row as Success and Failure

https://github.com/yosuadc3/ibmcourse/blob/master/Final%20Project%20SpaceX/labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

- We use few kind of chart using variable combination to draw insight
  - Catplot on flight number, launch site, and payload mass
  - Barplot on success rate and orbit type
  - Catplot on flight number, orbit type, and payload mass
  - Line plot yearly success trend

https://github.com/yosuadc3/ibmcourse/blob/master/Final%20Project%20SpaceX/jupyter-labs-eda-dataviz.ipynb

EDA with SQL

- We perform eda with sql on local mysql to find out about
  - The names of unique launch sites in the space mission.
  - The total payload mass carried by boosters launched by NASA (CRS)
  - The average payload mass carried by booster version F9 v1.1
  - The total number of successful and failure mission outcomes
  - The failed landing outcomes in drone ship, their booster version and launch site names.

https://github.com/yosuadc3/ibmcourse/blob/master/Final%20Project%20SpaceX/jupyter-labs-eda-sql-coursera.ipynb

Build an Interactive Map with Folium

- We marked all launch sites, and added map objects such as markers, circles, lines to mark the success or failure of launches for each site on the folium map.
- We assigned the feature launch outcomes (failure or success) to class 0 and 1.i.e., 0 for failure, and 1 for success.
- Using the color-labeled marker clusters, we identified which launch sites have relatively high success rate.
- We calculated the distances between a launch site to its proximities. We answered some question for instance:
  - Are launch sites near railways, highways and coastlines.
  - Do launch sites keep certain distance away from cities.

https://github.com/yosuadc3/ibmcourse/blob/master/Final%20Project%20SpaceX/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

- We built an interactive dashboard with Plotly dash
- We plotted pie charts showing the total launches by a certain sites
- We plotted scatter graph showing the relationship with Outcome and Payload Mass (Kg) for the different booster version.

https://github.com/yosuadc3/ibmcourse/blob/master/Final%20Project%20SpaceX/dashboard.py

Predictive Analysis (Classification)

- loaded the data using numpy and pandas, transformed the data, split our data into training and testing.
- We built different machine learning models and tune different hyperparameters using GridSearchCV.
- We used accuracy as the metric for our model, improved the model using feature engineering and algorithm tuning.
- Then we list all model with accuracy and select the most accurate.

https://github.com/yosuadc3/ibmcourse/blob/master/Final%20Project%20SpaceX/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

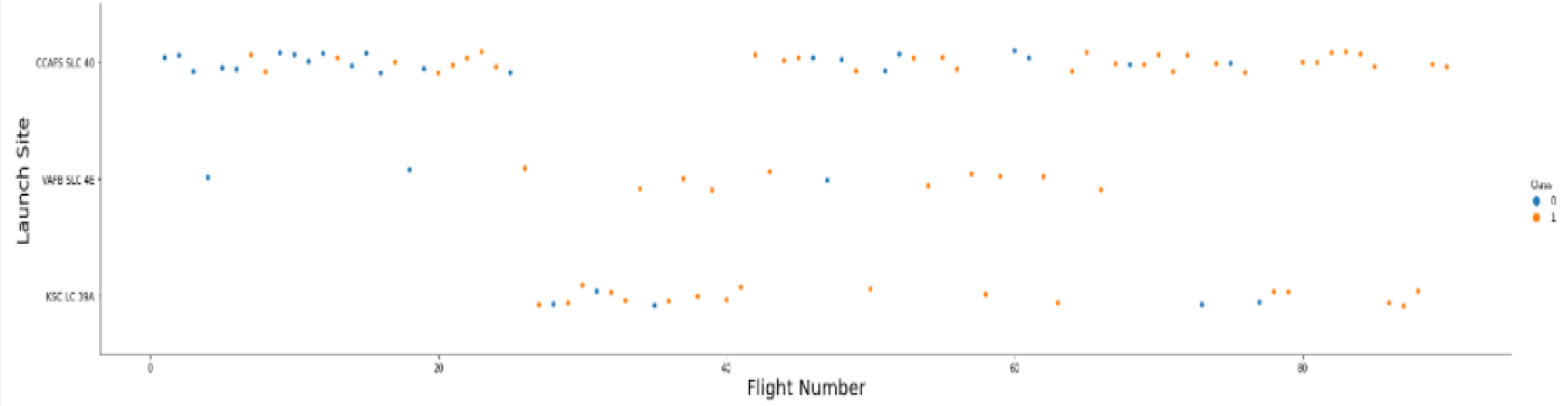
Results

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

Section 2

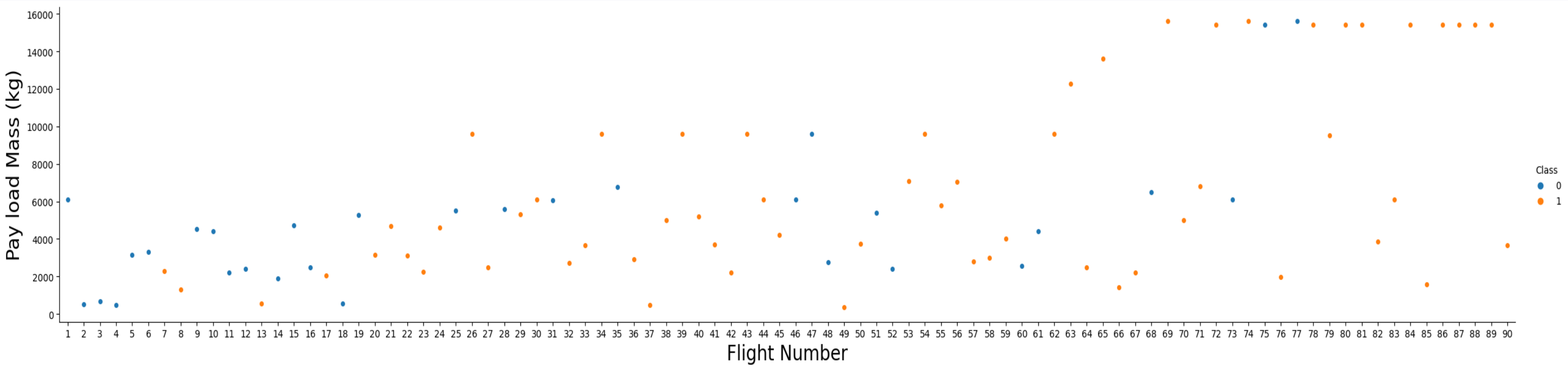
Insights drawn from EDA

Flight Number vs. Launch Site



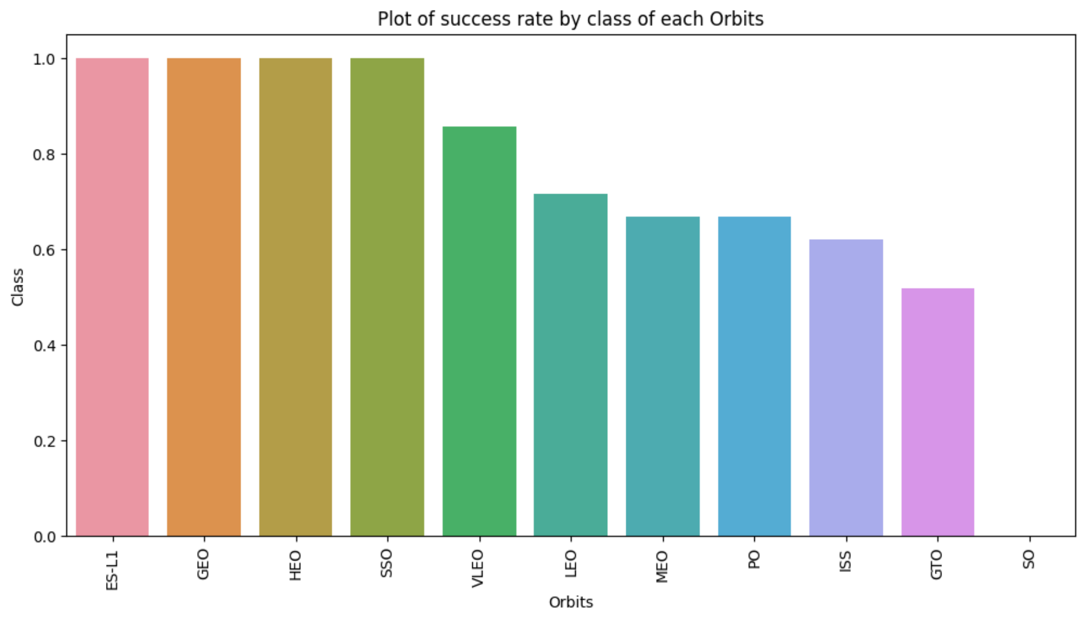
- From the plot, we found that the larger the flight amount at a launch site, the greater the success rate at a launch site.

Payload vs. Launch Site

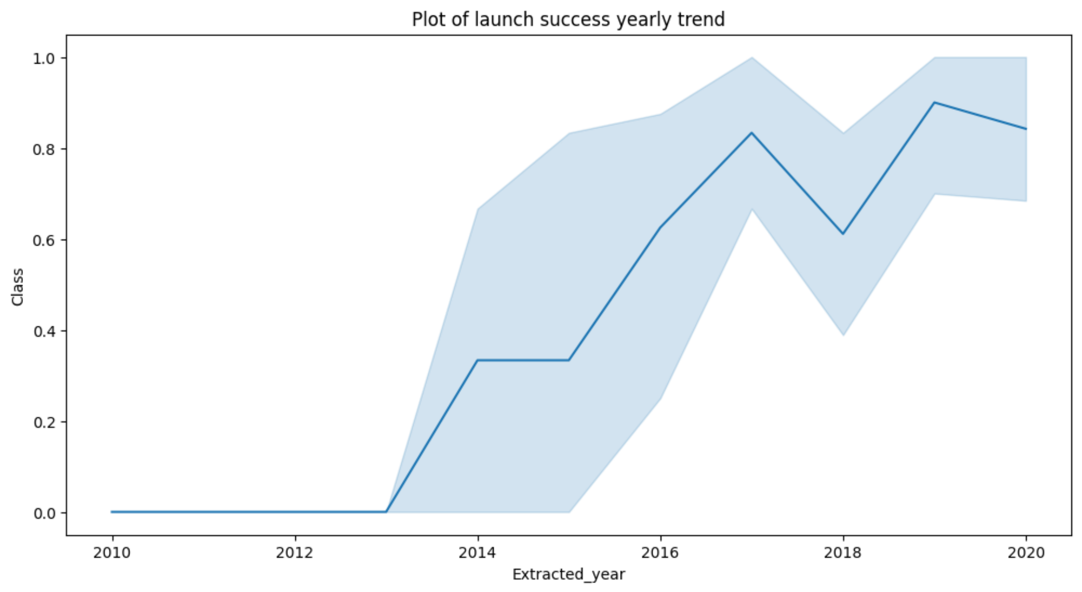


- Based on Scatter Plot we can conclude that 20 latest flight have payload mass more than 140000kg

Success Rate vs. Orbit Type



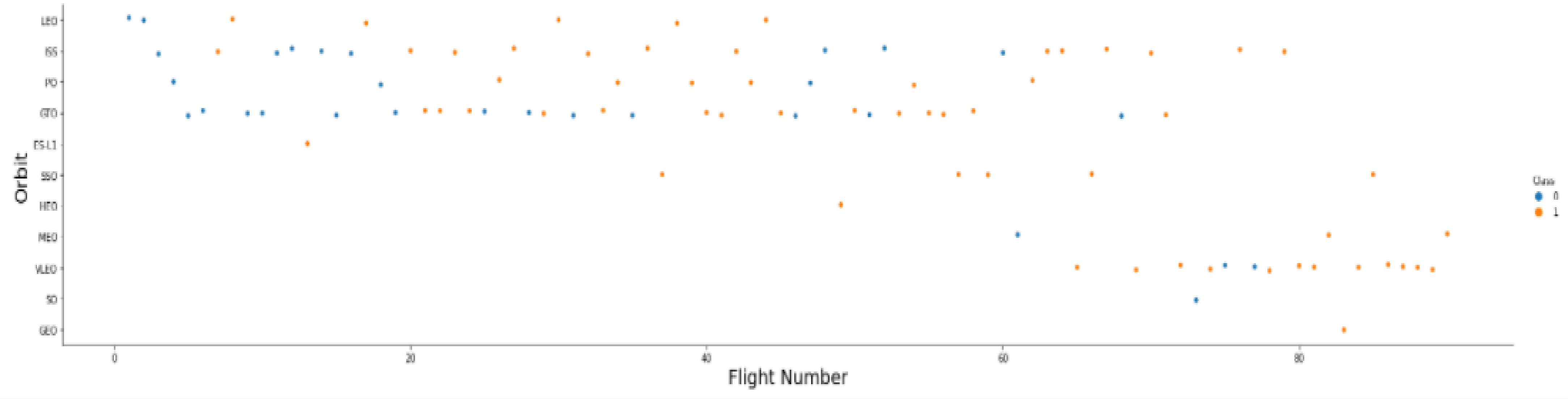
4 orbits have higher success rate than the rest



Success rate is increasing since 2013

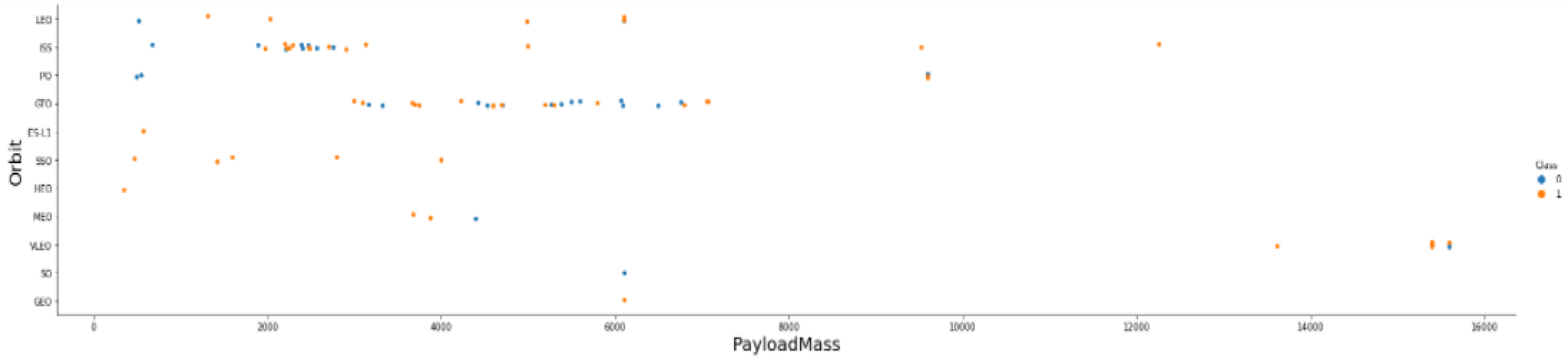
Flight Number vs. Orbit Type

- The plot below shows the Flight Number vs. Orbit type. We observe that in the LEO orbit, success is related to the number of flights whereas in the GTO orbit, there is no relationship between flight number and the orbit.

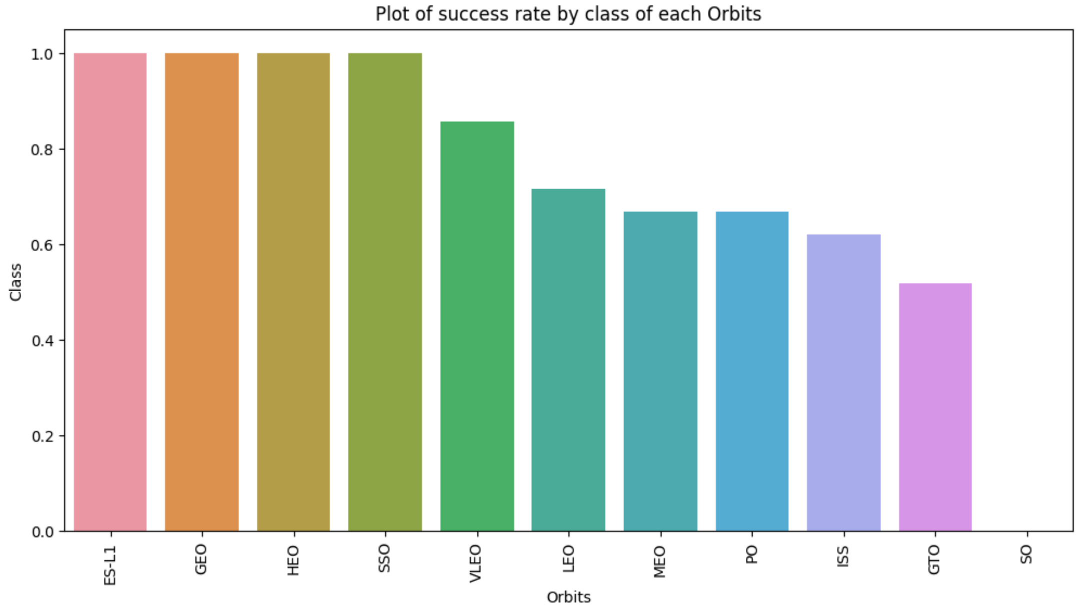


Payload vs. Orbit Type

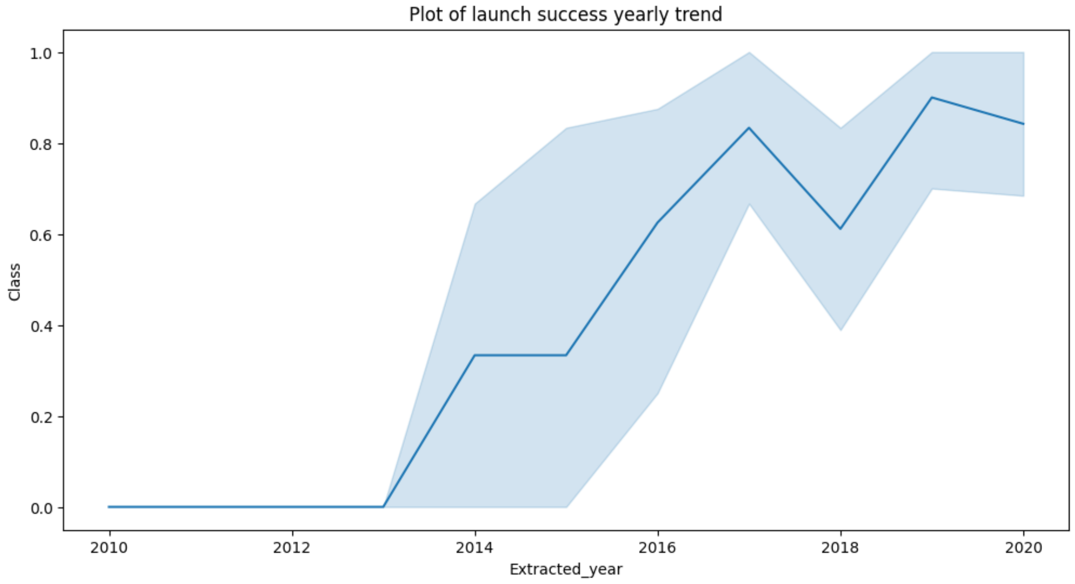
- We can observe that with heavy payloads, the successful landing are more for PO, LEO and ISS orbits.



Launch Success Yearly Trend



4 orbits have higher success rate than the rest



Success rate is increasing since 2013

All Launch Site Names

- We used the key word **DISTINCT** to show only unique launch sites from the SpaceX data.

```
%sql select distinct Launch_Site from spacex
```

* mysql://dc:***@localhost/databasecourse
4 rows affected.

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

Launch Site Names Begin with 'CCA'

```
%sql select * from spacex where Launch_Site like 'CCA%' limit 5
```

Python

* mysql://dc:***@localhost/databasecourse
5 rows affected.

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing _Outcome
2010-04-06 00:00:00	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-08-12 00:00:00	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 00:00:00	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-08-10 00:00:00	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-01-03 00:00:00	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

+ Code + Markdown

- We used the query above to display 5 records where launch sites begin with `CCA`

Total Payload Mass

- We calculated the total payload carried by boosters from NASA as 45596 using the query below

```
%sql select sum(PAYLOAD_MASS__KG_   ) from spacex where Customer = 'NASA (CRS)'
```

```
* mysql://dc:***@localhost/databasecourse
1 rows affected.
```

sum(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

- We calculated the average payload mass carried by booster version F9 v1.1 as 2928.4

```
%sql select AVG(PAYLOAD_MASS__KG_) from spacex where Booster_Version LIKE 'F9 v1.1%'
```

```
 * mysql://dc:***@localhost/databasecourse
1 rows affected.
```

AVG(PAYLOAD_MASS__KG_)
2534.6667

First Successful Ground Landing Date

- We observed that the dates of the first successful landing outcome on ground pad was 22nd December 2015

```
%sql select min(Date) from spacex where `Landing _Outcome` = 'Success (ground pad)'
```

* mysql://dc:***@localhost/databasecourse
1 rows affected.

min(Date)
2015-12-22 00:00:00

Successful Drone Ship Landing with Payload between 4000 and 6000

- We used the **WHERE** clause to filter for boosters which have successfully landed on drone ship and applied the **AND** condition to determine successful landing with payload mass greater than 4000 but less than 6000

```
%sql SELECT Booster_Version FROM spacex WHERE `Landing _Outcome` = 'Success (drone ship)' and `PAYLOAD_MASS__KG_` between 4000 and 6000
```

* mysql://dc:***@localhost/databasecourse
4 rows affected.

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

Total Number of Successful and Failure Mission Outcomes

- We used wildcard like '%' to filter for **WHERE** MissionOutcome was a success or a failure.

```
%%sql
SELECT
    COUNT(IF(Mission_Outcome LIKE 'Success%',1,null)) AS `SUCCESS MISSION`,
    COUNT(IF(Mission_Outcome LIKE 'Failure%',1,null)) AS `FAILURE MISSION`
FROM spacex
```

* mysql://dc:***@localhost/databasecourse
1 rows affected.

SUCCESS MISSION	FAILURE MISSION
100	1

Boosters Carried Maximum Payload

- We determined the booster that have carried the maximum payload using a subquery in the **WHERE** clause and the **MAX()** function.

```
SELECT Booster_Version, PAYLOAD_MASS__KG_ FROM spacex WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM spacex) ORDER BY Booster_Version
```

* mysql://dc:***@localhost/databasecourse
12 rows affected.

Booster_Version	PAYLOAD_MASS__KG_
F9 B5 B1048.4	15600
F9 B5 B1048.5	15600
F9 B5 B1049.4	15600
F9 B5 B1049.5	15600
F9 B5 B1049.7	15600
F9 B5 B1051.3	15600
F9 B5 B1051.4	15600
F9 B5 B1051.6	15600
F9 B5 B1056.4	15600
F9 B5 B1058.3	15600
F9 B5 B1060.2	15600
F9 B5 B1060.3	15600

2015 Launch Records

- We used a combinations of the **WHERE** clause, **LIKE**, **AND**, and **BETWEEN** conditions to filter for failed landing outcomes in drone ship, their booster versions, and launch site names for year 2015

```
%%sql

SELECT `Booster_Version`, `Launch_Site`, `Landing _Outcome`
        FROM spacex
        WHERE `Landing _Outcome` LIKE 'Failure (drone ship)'
            AND Date BETWEEN '2015-01-01' AND '2015-12-31'
```

* mysql://dc:***@localhost/databasecourse
2 rows affected.

Booster_Version	Launch_Site	Landing _Outcome
F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

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

- We selected Landing outcomes and the **COUNT** of landing outcomes from the data and used the **WHERE** clause to filter for landing outcomes **BETWEEN** 2010-06-04 to 2010-03-20.
- We applied the **GROUP BY** clause to group the landing outcomes and the **ORDER BY** clause to order the grouped landing outcome in descending order.

```
%%sql
SELECT `Landing _Outcome`, COUNT(`Landing _Outcome`)
    FROM spacex
    WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20'
    GROUP BY `Landing _Outcome`
    ORDER BY COUNT(`Landing _Outcome`) DESC
```

* mysql://dc:***@localhost/databasecourse
8 rows affected.

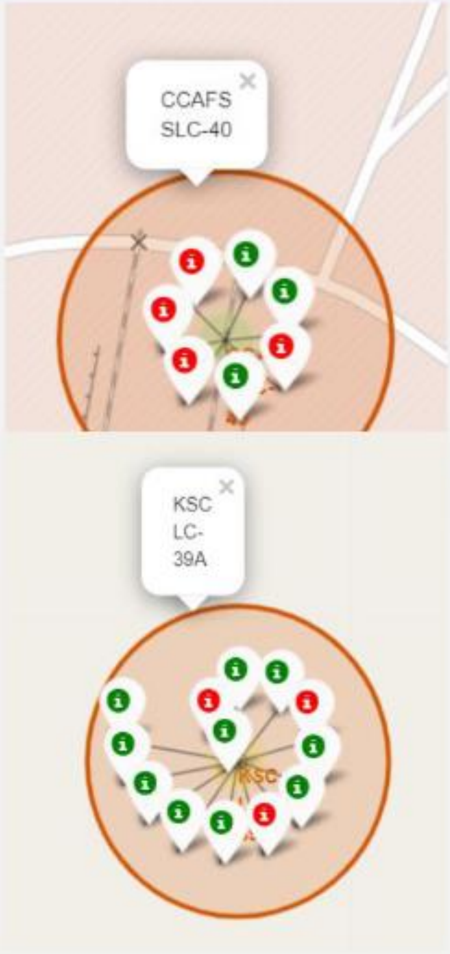
Landing _Outcome	COUNT(`Landing _Outcome`)
No attempt	10
Failure (drone ship)	5
Success (ground pad)	5
Success (drone ship)	5
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	1
Precluded (drone ship)	1

Section 3

Launch Sites Proximities Analysis

All launch sites global map markers

Markers showing launch sites with color labels



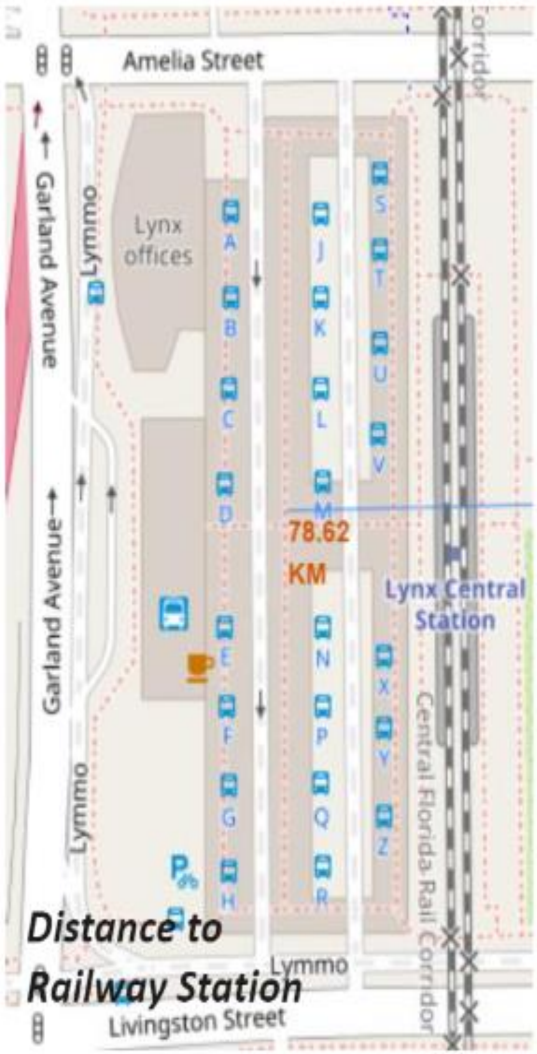
California Launch Site

37

Florida Launch Sites

Green Marker shows successful Launches and Red Marker shows Failures

Launch Site distance to landmarks

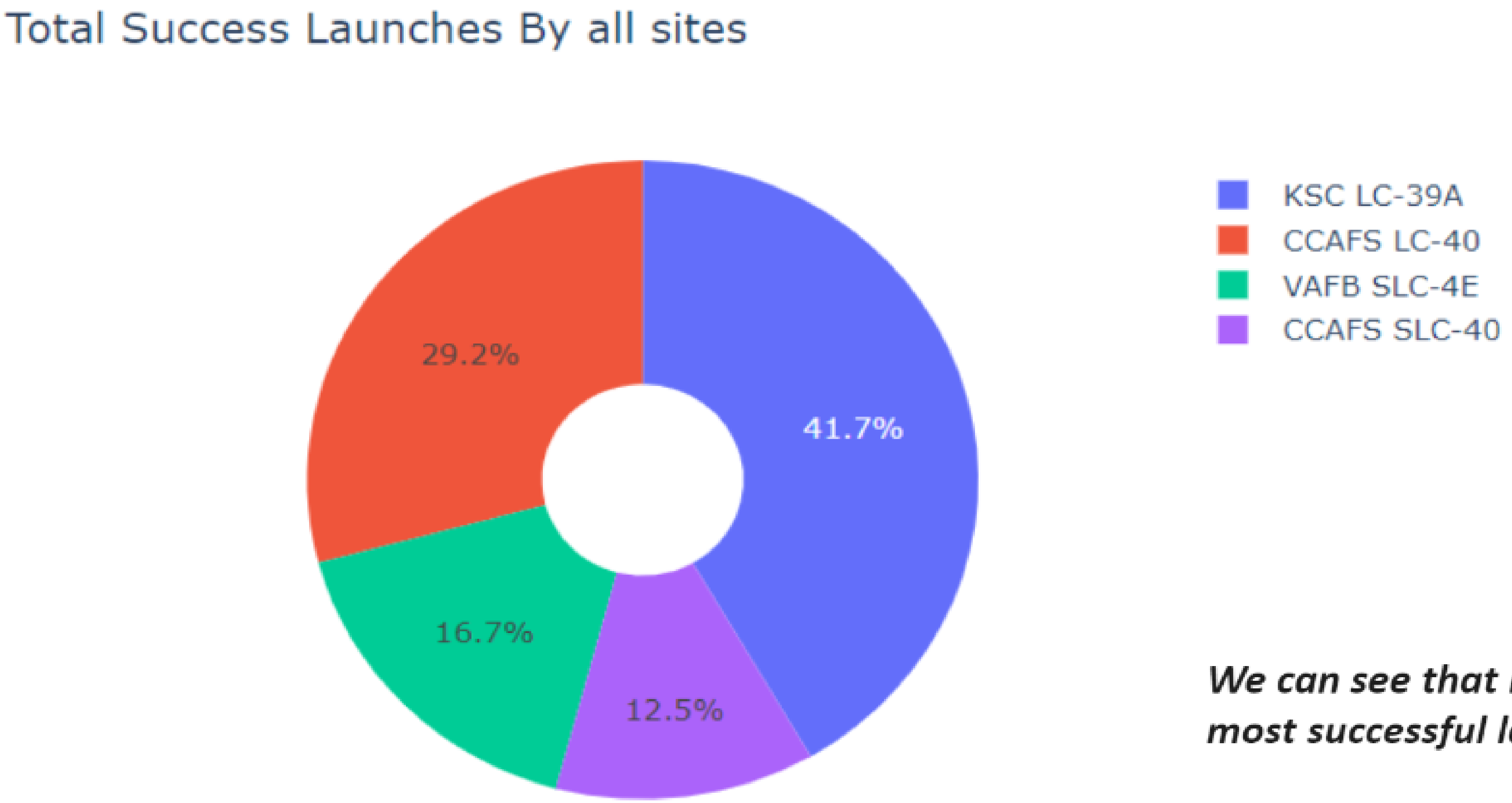


•Are launch sites in close proximity to railways? No
•Are launch sites in close proximity to highways? No
•Are launch sites in close proximity to coastline? Yes
•Do launch sites keep certain distance away from cities? Yes

Section 4

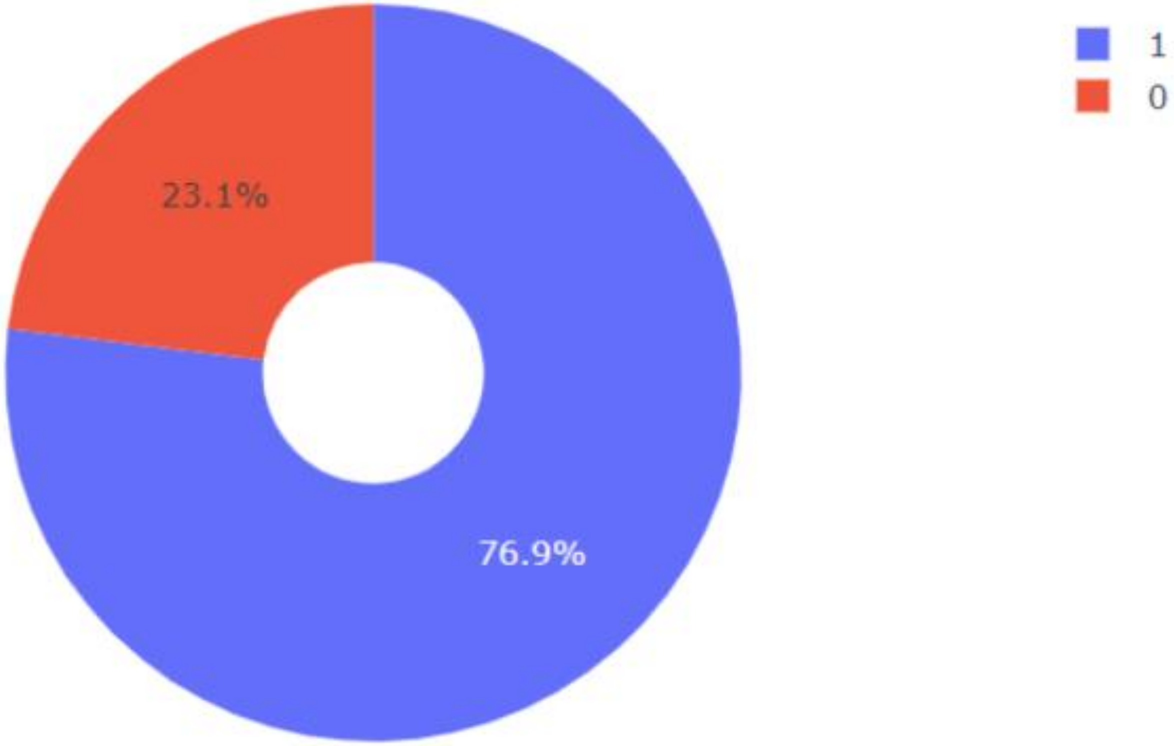
Build a Dashboard with Plotly Dash

Pie chart showing the success percentage achieved by each launch site



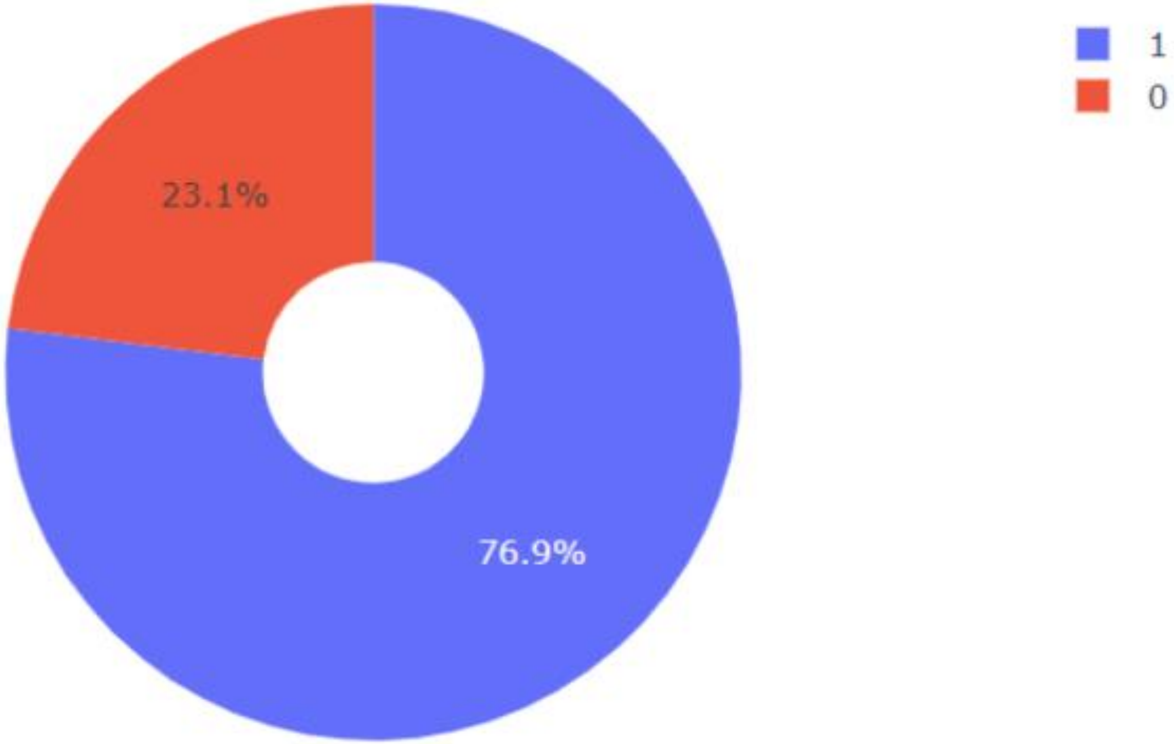
We can see that KSC LC-39A had the most successful launches from all the sites

Pie chart showing the Launch site with the highest launch success ratio



KSC LC-39A achieved a 76.9% success rate while getting a 23.1% failure rate

Pie chart showing the Launch site with the highest launch success ratio



KSC LC-39A achieved a 76.9% success rate while getting a 23.1% failure rate

Section 5

Predictive Analysis (Classification)

Classification Accuracy

- The decision tree classifier is the model with the highest classification accuracy

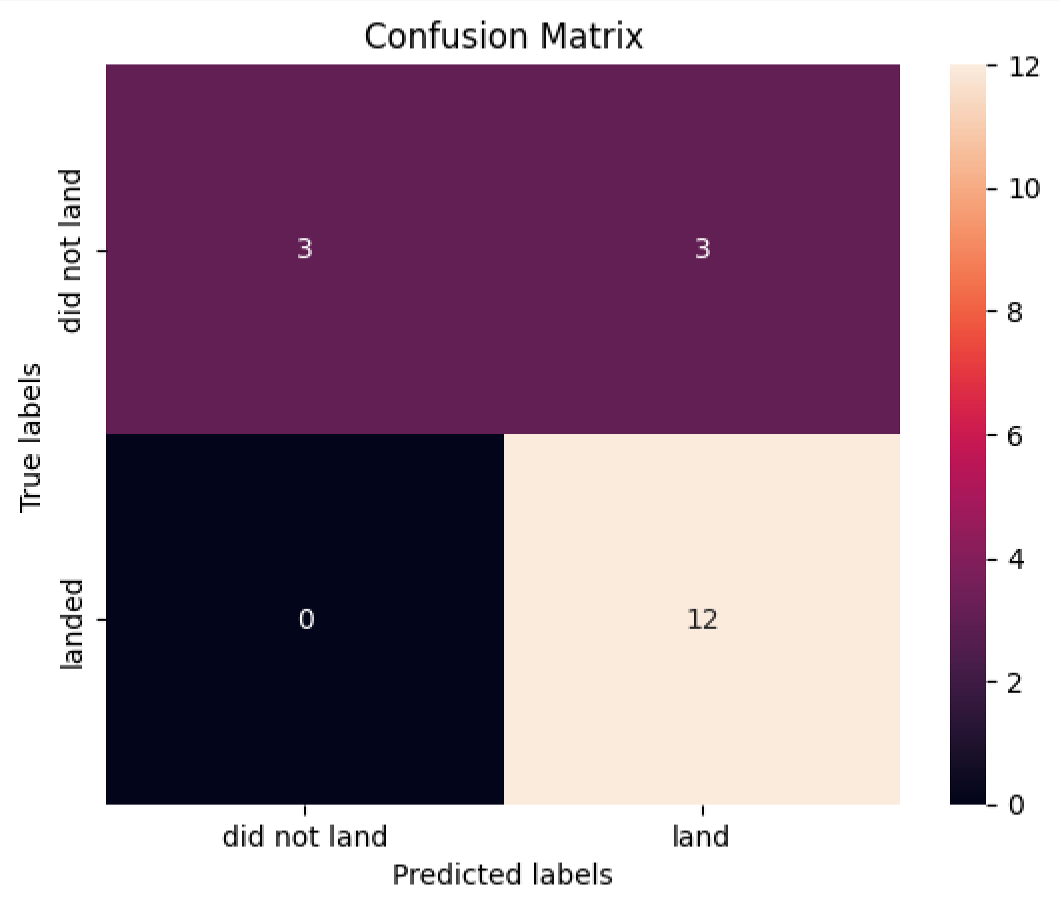
```python
models = {'KNeighbors':knn_cv.best_score_,
          'DecisionTree':tree_cv.best_score_,
          'LogisticRegression':logreg_cv.best_score_,
          'SupportVector': svm_cv.best_score_}

bestalgorithm = max(models, key=models.get)
print('Best model is', bestalgorithm,'with a score of', models[bestalgorithm])
if bestalgorithm == 'DecisionTree':
    print('Best params is :', tree_cv.best_params_)
if bestalgorithm == 'KNeighbors':
    print('Best params is :', knn_cv.best_params_)
if bestalgorithm == 'LogisticRegression':
    print('Best params is :', logreg_cv.best_params_)
if bestalgorithm == 'SupportVector':
    print('Best params is :', svm_cv.best_params_)
```

✓ 0.1s

```
Best model is DecisionTree with a score of 0.875
Best params is : {'criterion': 'entropy', 'max_depth': 8, 'max_features': 'auto', 'min_samples_leaf': 2, 'min_samples_split': 2, 'splitter': 'random'}
```

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.
- The decision tree successfully predict 12 landed and 3 did not land



Conclusions

We can conclude that:

- 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.
- The Decision tree classifier is the best machine learning algorithm for this task.

Thank you!