Space X

Thiago Amanajás

09.12.2021

OUTLINE

- Executive Summary
- Introduction
- Methodology
- Results
  - Charts
  - Database searches
  - Interactive dashboard
- Conclusion

EXECUTIVE SUMMARY

Methodology

The analysis of the data starts with its extraction from the SpaceX API and the SpaceX Wiki page by Web scrapping, thus continuing with Data Wrangling and an interactive visual EDA and consulting the data using SQL. Furthermore, it is including the utilization of Folium maps to have a better overview of the locations of the launch sites and the implementation of dashboards with Plotly Dash to interact in real-time with the data. Finally, a Predictive Analysis is applied using different algorithms to classify and helping us to predict if the launch will be successful or not.

Results

The exposition of the results is done by data analysis charts, SQL search results, and the screenshots taken from the interactive dashboard.

INTRODUCTION

- What it is about
  - Falcon9 rocket launches advertised on SpaceX site cost 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings are because SpaceX can reuse the first stage. Therefore, determining if the first stage will land will also determine the costs of a launch. This information can be used if an alternate company wants to bid against Space X for a rocket launch.
- Questions to be answered
  - Is there any variable that has a big influence in determining if the launch will be successful?
  - Is there a correlation between success rate and orbit type?
  - What are the right conditions for the launch to achieve successful results?
  - Which algorithm can we use to best predict the success of the launches with this dataset?

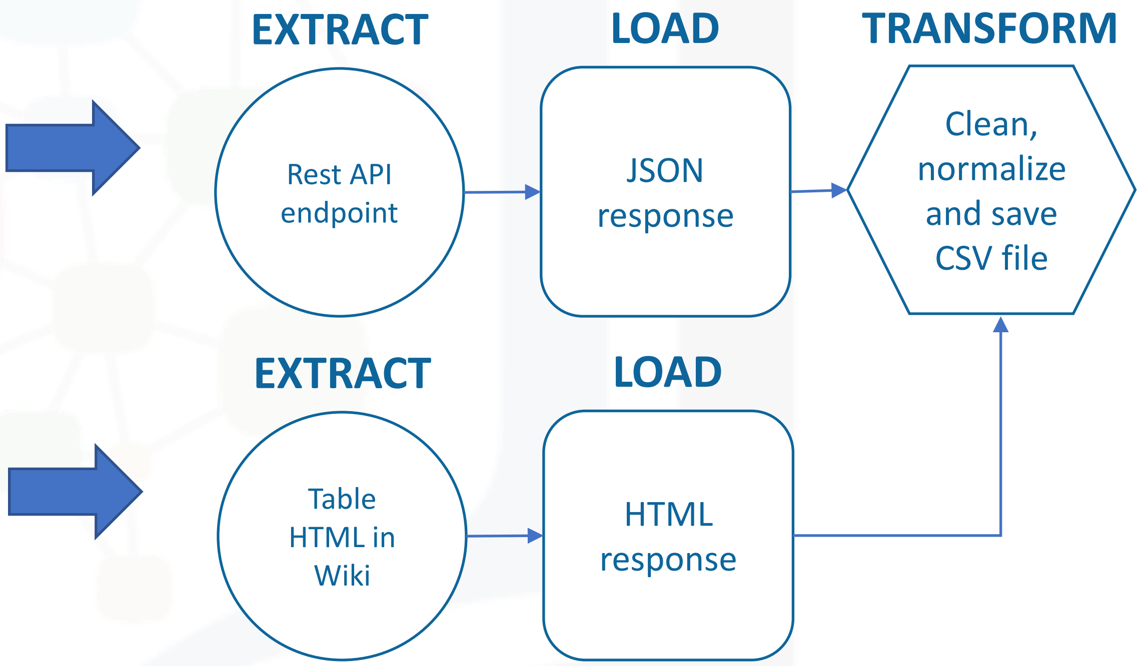
METHODOLOGY SUMMARY

- Collection of data
  - Rest API from SpaceX
  - Web scraping of SpaceX page on Wikipedia
- Data Wrangling
  - Transforming the data to be used in the machine learning step
    - Dropping irrelevant columns
    - One Hot Encoding
- EDA (Exploratory Data Analysis)
  - Data visualization with scatter plots, and bar plots
  - Using SQL
- Folium maps and a dashboard with Plotly Dash
- Predictive analysis (Classification)

DATA COLLECTION

Where & what was collected → How

- We utilized directly the Rest API endpoints from SpaceX and extracted the gathered data after launch.
    - This data contains information about launch sites, landing pads, rocket type, results from the landings and payload in kg.
- Secondly, we extracted the data from the SpaceX page in Wikipedia by web scrapping.
    - This data contains extra columns like, payload name, time of launch, and customer.



SPACEX API

[Access GitHub Lab]

1 Requesting the Rest API endpoint for launches

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

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

2 Cleaning and normalizing the response by filtering, dropping duplicates and applying custom methods

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

```
data = data[['rocket', 'payloads', 'launchpad', 'cores', 'flight_number', 'date_utc']]
data = data[data['cores'].map(len)==1]
data = data[data['payloads'].map(len)==1]
data['cores'] = data['cores'].map(lambda x : x[0])
data['payloads'] = data['payloads'].map(lambda x : x[0])
data['date'] = pd.to_datetime(data['date_utc']).dt.date
data = data[data['date'] <= datetime.date(2020, 11, 13)]
```

```
df_launch.drop_duplicates(subset=None, keep='first', inplace=True)
df_launch.reset_index(drop=True, inplace=True)
```

```
getCoreData(data)
getPayloadData(data)
getLaunchSite(data)
getBoosterVersion(data)
```

3 Saving the CSV file

```
data_falcon9.to_csv('dataset_part_1.csv', index=False)
```

WEB SCRAPPING

Access GitHub Lab

1 Request and load the page code by using BeautifulSoup

```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"
wiki_falcon9 = requests.get(static_url)
soup = BeautifulSoup(wiki_falcon9.content)
```

2 Search for all table elements and select the third table

```
html_tables = soup.find_all("table")
first_launch_table = html_tables[2]
```

3 Get all the column names

```
column_names = []
for column in first_launch_table.find_all("th"):
    name = extract_column_from_header(column)
    if name is not None and len(name) > 0:
        column_names.append(name)
```

4 Create the dictionary with the right columns

```
launch_dict= dict.fromkeys(column_names)

# Remove an irrelvant column
del launch_dict['Date and time ( )']

launch_dict['Flight No.'] = []
launch_dict['Launch site'] = []
launch_dict['Payload'] = []
```

5 Extract the values from the table and append it to the keys

```
extracted_row = 0
#Extract each table
for table_number,table in enumerate(soup.find_all('table',"wikitak
   # get table row
    for rows in table.find_all("tr"):
        #check to see if first table heading is as number correspo
        if rows.th:
            if rows.th.string:
                flight_number=rows.th.string.strip()
```

6 Create the DataFrame and save

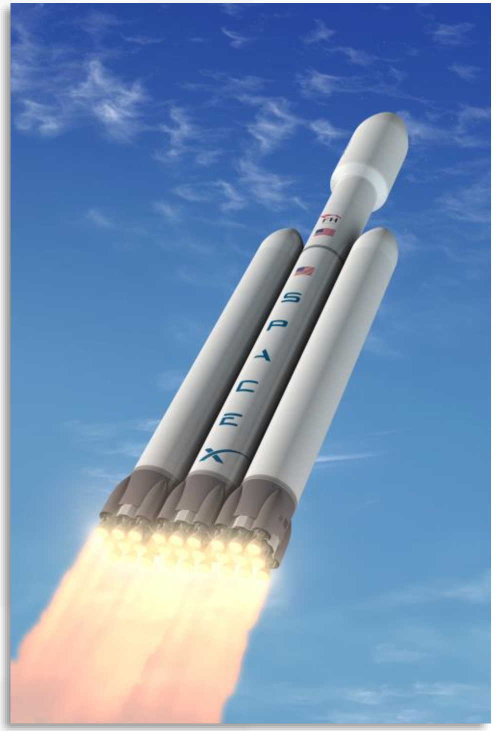
```
df=pd.DataFrame.from_dict(launch_dict, orient='index')
df = df.transpose()
df.to_csv('spacex_web_scraped.csv', index=False)
```

DATA WRANGLING

Access GitHub Lab

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.
- **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.



In order to understand more about the landing attempts, the following operations were executed:

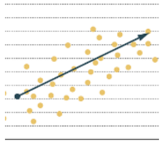
- Calculate the number of launches at each site
- Calculate the number and occurrence of each orbit
- Create a landing outcome label from Outcome column → Convert the outcomes into training labels (Class)
- Calculate the number and occurrence of mission outcome per orbit type
- Save the dataset as CSV to be utilized by the next phases

EDA (Exploratory Data Analysis)

[Access GitHub Lab]

Scatter Graphs

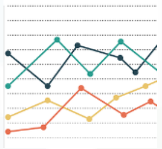
- Flight Number Vs Payload Mass (Kg)
- Flight Number Vs Launch Site
- Flight Number Vs Orbit type
- Launch Site Vs Payload Mass (Kg)
- Payload Mass (Kg) Vs Orbit type



This chart is used here because it facilitates the visualization of correlated variables and how much they can affect each other.

Line Graphs

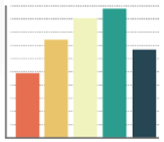
- Launch success yearly trend



This chart is used to visualize the value of something over time because it is easy to see the tendencies of the variables.

Bar Graphs

- Success rate of each orbit type



This chart is used to compare two or more states based on the length of the bar.

EDA using SQL

Access GitHub Lab

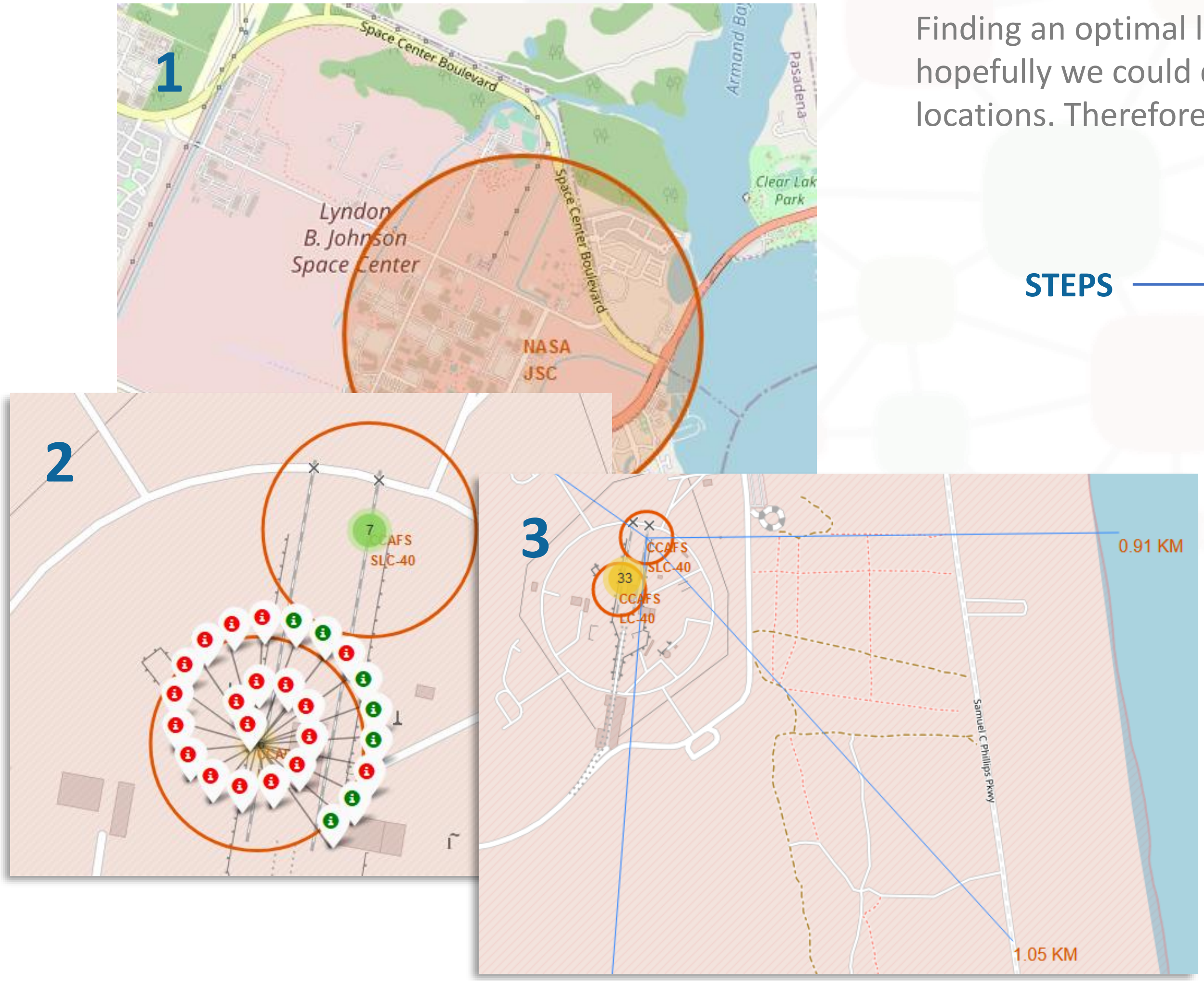
To understand better our dataset, we performed the following questions using SQL to extract more information regarding the launch sites, payload, booster, and landing outcomes.

1. What are the names of the launch sites in the space mission?
2. How many launch sites begin with the string “CCA”?
3. What is the total payload mass carried by boosters launched by NASA (CRS)?
4. What is the average of payload mass carried by booster version F9 v1.1?
5. What date the first successful landing outcome in ground pad was achieved?
6. What are the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000?
7. What is the total number of successful and failure mission outcomes?
8. What are the names of the booster versions which have carried the maximum payload mass?
9. What are the failed landing outcomes in drone ship including their booster versions and launch site names in the year 2015?
10. What is the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) ranked by the highest value between the date 2010-06-04 and 2017-03-20?

Note1: The answers are at the result slides
Note2: The questions were formulated based on the topics of the notebook.
For more insight, check the GitHub Lab.

Creating Folium maps

Access GitHub Lab



Finding an optimal location for building a launch site certainly involves many factors and hopefully we could discover some of the factors by analysing the existing launch site locations. Therefore, the following steps were used to analyse the locations with Folium.

STEPS

1. Marking all launch sites by adding circle format markers and a label containing the name of the site using the latitude and longitude from each launch site location.
2. Marking the success/failed launches for each site on the map. Using a cluster of markers and the different location of the launch sites and adding a red or green marker depending if the outcome of the launch was successful or not.
3. Calculate the distances between a launch site to its proximities by adding a line between railways, highways, and cities and measuring their distances to the launch site.

The above steps were necessary to answer the following:

- Are launch sites near railways? No
- Are launch sites near highways? No
- Are launch sites near coastlines? Yes
- Do launch sites keep certain distance away from cities? Yes

Creating dashboard using Plotly Dash

Access GitHub project

Plotly Dash is a dashboard engine to deliver advanced analytics faster. It is ideal for building and deploying data apps with customized user interfaces. It's particularly suited for anyone who works with data.

1 Define variables of max and min payload

```
spacex_df = pd.read_csv("spacex_launch_dash.csv")
max_payload = spacex_df['Payload Mass (kg)'].max()
min_payload = spacex_df['Payload Mass (kg)'].min()
```

2 Add dropdown with all the launch sites

```
dcc.Dropdown(id='site-dropdown',
    options=[
        {'label': 'All Sites', 'value': 'ALL'},
        {'label': 'CCAFS LC-40', 'value': 'CCAFS LC-40'},
        {'label': 'CCAFS SLC-40', 'value': 'CCAFS SLC-40'},
        {'label': 'KSC LC-39A', 'value': 'KSC LC-39A'},
        {'label': 'VAFB SLC-4E', 'value': 'VAFB SLC-4E'},
    ],
    value='ALL',
    placeholder="Launch Site here",
    searchable=True
    ),
```

3 Add a pie chart to show the successful Vs failed launches and implement the callback method

```
html.Div(dcc.Graph(id='success-pie-chart')),
```

```
@app.callback(Output(component_id='success-pie-chart', component_property='figure'),
              Input(component_id='site-dropdown', component_property='value'))
def get_pie_chart(entered_site):
```

4 Add a slider to select the payload range

```
dcc.RangeSlider(id='payload-slider',
    min=0, max=10000, step=1000,
    marks={0: '0',
        100: '100'},
    value=[min_payload, max_payload]),
```

Next slide to continue

Creating dashboard using **Plotly Dash**

[Access GitHub project](#)

5 Implement the callback method for the payload slider

```
@app.callback(Output(component_id='success-payload-scatter-chart', component_property='figure'),
              Input(component_id='site-dropdown', component_property='value'),
              Input(component_id='payload-slider', component_property='value'))
def get_scatter_chart(entered_site, payload):
```

6 Define the scatter chart element

```
html.Div(dcc.Graph(id='success-payload-scatter-chart')),
])
```

7 Run the dashboard

```
$ python app.py
```

8 See the complete implementation

Dashboard screenshot

Predictive analysis (Classification)

Access GitHub Lab

Constructing the model

- We loaded the data from CSV files using Pandas and NumPy.
- The second step was standardizing the data.
- Furthermore, we split the data into Train set and Test set.
    - Then checked the number of samples.
    - There should be 18 samples.
- Using different machine learning algorithms to decide which one to select for our model. We fit and train our datasets with **GridSearchCV**.

Finding the method that performs best

- We chose the model that has the best accuracy based on the algorithm below that prints the max accuracy found.

```
algorithm = {"KNN":knn_cv.best_score_,
             "Tree":tree_cv.best_score_,
             "LogisticRegression":logreg_cv.best_score_,
             "SVM": svm_cv.best_score_}
params = {"KNN":knn_cv.best_params_,
             "Tree":tree_cv.best_params_,
             "LogisticRegression":logreg_cv.best_params_,
             "SVM": svm_cv.best_params_}
best = max(algorithm, key=algorithm.get)
print('Best Algorithm is', best, 'with a score of', algorithm[best])
print('Best Params is :', params[best])
```

Evaluation

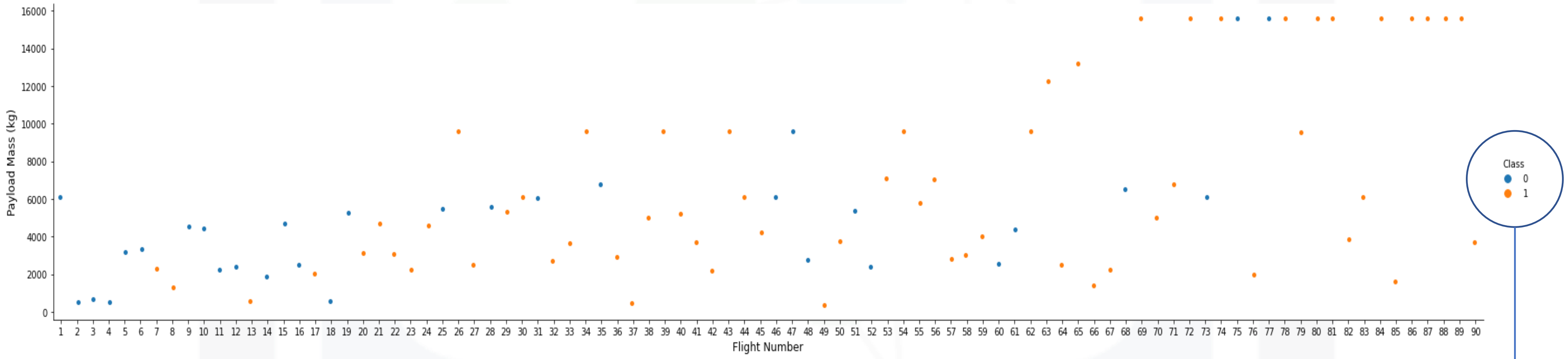
- We tuned the parameters on each algorithm to fit best the algorithm and checked the accuracy against the others.
- We used Confusion Matrix to see how it distinguishes the classes.

RESULTS

- Exploratory data analysis
- Performed predictive analysis
- Interactive dashboard screenshots

EDA visualization results

Payload Mass (Kg) Vs Flight Number

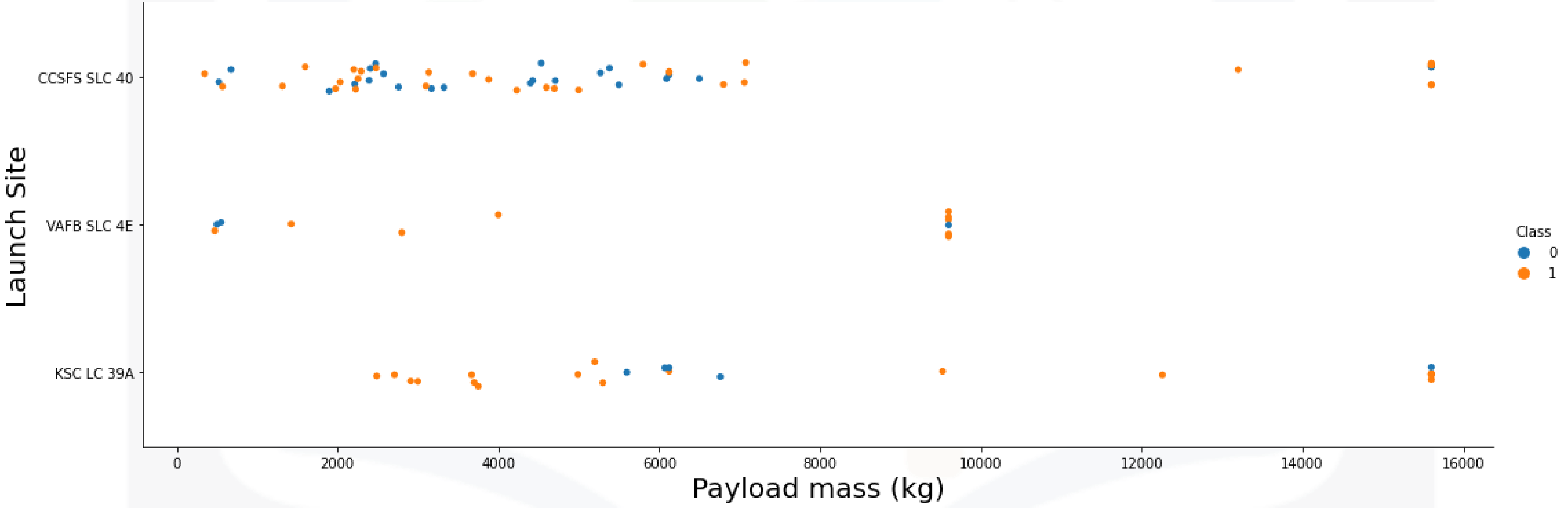


It is not clear if there is a correlation between the payload and the flight number, but it seems that the **success rate** (Class) increases when the payload is lower.

Class = Success rate

EDA visualization results

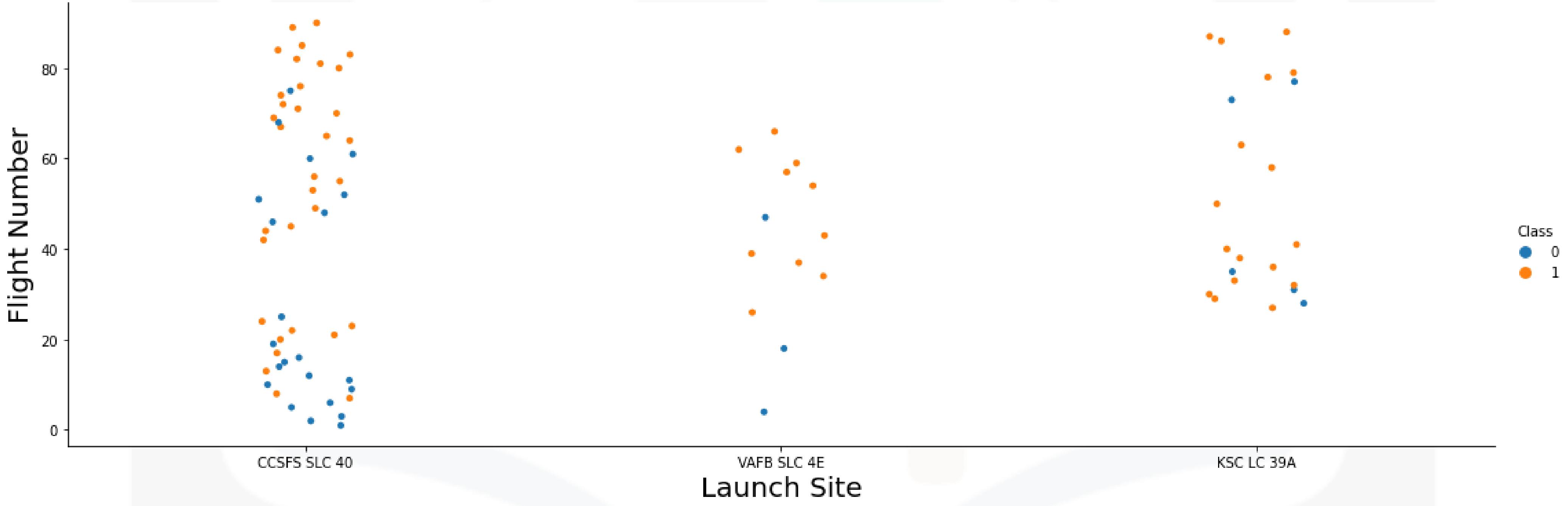
Launch Site Vs Payload Mass (Kg)



The greater the payload mass, the lowest is the success rate. Although, for the VAFB-SLC launch site there are no rockets launched for heavy payload mass (greater than 10000).

EDA visualization results

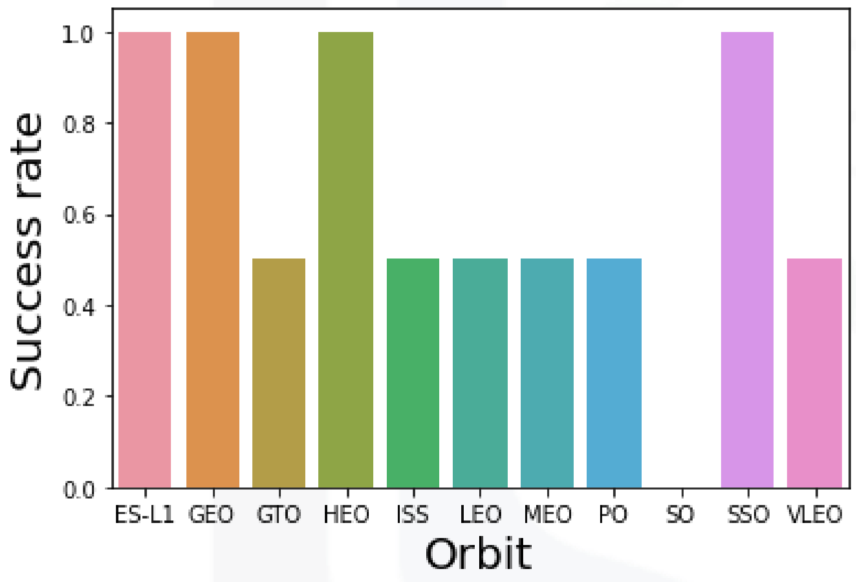
Flight Number Vs Launch Site



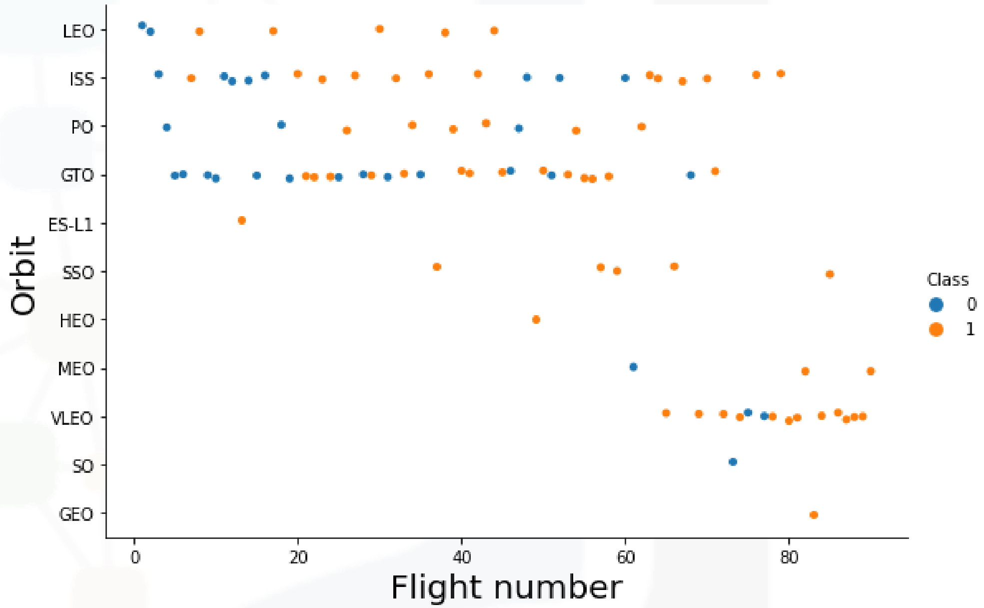
As you can see there seems not to be a correlation between flight number and launch sites to success rate.
CCSFS SLC 40 has more flights than KSC LC 39A, but the latest shows a higher success rate.

EDA visualization results

Class Vs Orbit



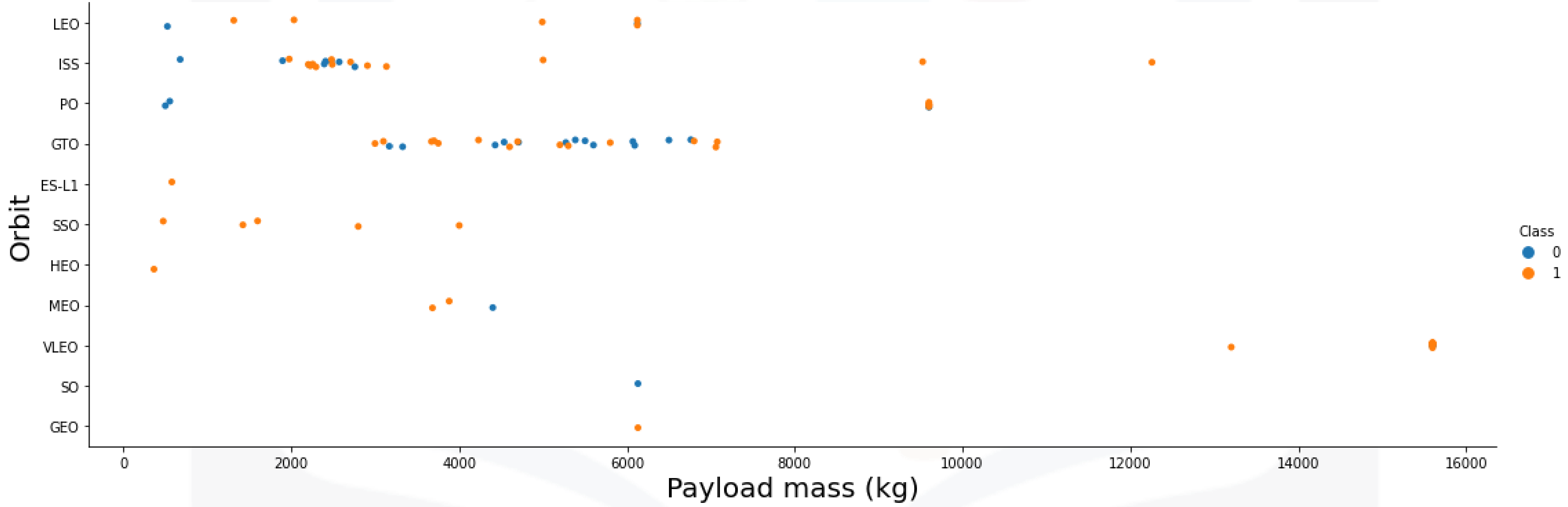
Orbit Vs Flight Number



The left chart shows the success rate regarding the orbit and the right chart the relation between orbit and flight number. ES-L1, SSO, GEO and HEO orbits had few launches compared to the others, but all successful. Therefore, when it comes to GTO orbit there seems to be no relation between flight number and success rate per orbit.

EDA visualization results

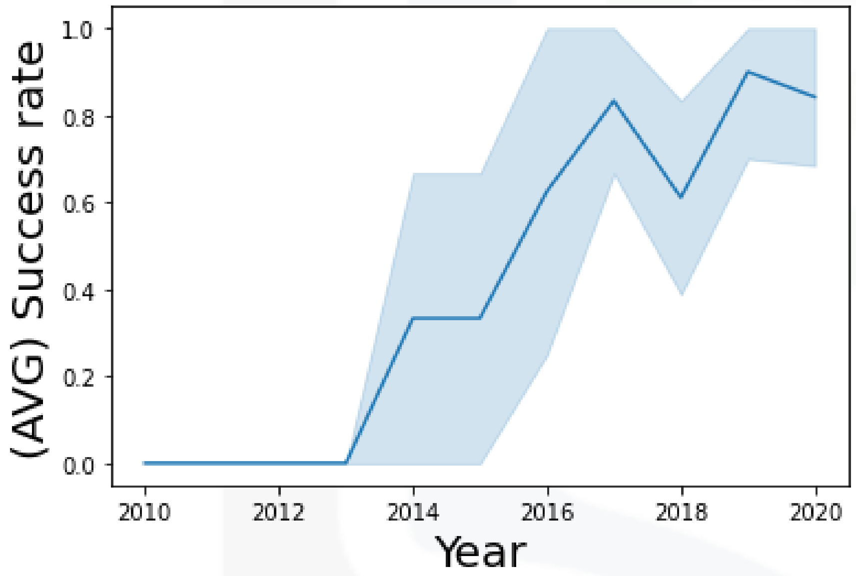
Orbit Vs Payload Mass (Kg)



With heavy payloads, the successful landing or positive landing rate are more for Polar, LEO and ISS. However, for GTO we cannot distinguish it well as both positive landing rate and negative landing (unsuccessful mission) can be seen here.

EDA visualization results

Success Vs Year



This line plot takes into consideration the outcome of the launches and compares it against the years.
As you can observe the success rate kept increasing since 2013.

EDA SQL results

1. What are the names of the launch sites in the space mission?

```
SELECT UNIQUE launch_site FROM SPACEXTBL;
```

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

2. How many launch sites begin with the string "CCA"?

```
SELECT * FROM SPACEXTBL WHERE launch_site LIKE 'CCA%' ORDER BY DATE LIMIT 5;
```

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	None	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	None	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-05-22	07:44:00	F9 v1.0 B0005	CCAFS LC-40	None	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	00:35:00	F9 v1.0 B0006	CCAFS LC-40	None	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	None	677	LEO (ISS)	NASA (CRS)	Success	No attempt

EDA SQL results

3. What is the total payload mass carried by boosters launched by NASA (CRS)?

```
SELECT payload_mass__kg_ FROM SPACEXTBL WHERE customer LIKE 'NASA (CRS)';
```

payload_mass__kg_
500
677
2296
2216
2395
1898
1952
3136
2257
2490
2708
3310
2205
2647
2697
2500

4. What is the average of payload mass carried by booster version F9 v1.1?

```
SELECT AVG(payload_mass__kg_) FROM SPACEXTBL WHERE booster_version LIKE 'F9 v1.1';
```

1
2928

5. What date the first successful landing outcome in ground pad was achieved?

```
SELECT MIN(DATE) FROM SPACEXTBL WHERE landing__outcome LIKE '%ground pad%' and mission_outcome = 'Success';
```

1
2015-12-22

6. What are the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000?

```
SELECT booster_version FROM SPACEXTBL WHERE landing__outcome LIKE '%drone ship%'
and mission_outcome = 'Success' and payload_mass__kg_ > 4000 and payload_mass__kg_ < 6000;
```

booster_version
F9 FT B1020
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
F9 FT B1031.2

Note: For some queries it was used **LIKE** '%string%' because of the different values containing the specific string, e.g.: `LIKE '%drone ship%'` since the results would contain *Failure (drone ship)* or *Success (drone ship)*.

EDA SQL results

7. What is the total number of successful and failure mission outcomes?

```
SELECT COUNT(*) FROM SPACEXTBL WHERE mission_outcome LIKE '%Success%' and landing__outcome LIKE '%Success%';
SELECT COUNT(*) FROM SPACEXTBL WHERE mission_outcome NOT LIKE '%Success%' or landing__outcome LIKE '%Failure%';
```

```
Success:
| 1  |
+----+
| 61 |
+----+
Failure:
| 1  |
+----+
| 11 |
+----+
```

8. What are the names of the booster versions which have carried the maximum payload mass?

```
SELECT booster_version FROM SPACEXTBL WHERE payload_mass__kg_ in (SELECT MAX(payload_mass__kg_) FROM SPACEXTBL);
```

booster_version
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

9. What are the failed landing outcomes in drone ship including their booster versions and launch site names in the year 2015?

```
SELECT landing__outcome, booster_version, launch_site FROM SPACEXTBL
WHERE landing__outcome LIKE '%drone ship%' and YEAR(DATE) = YEAR('2015-01-01');
```

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
Precluded (drone ship)	F9 v1.1 B1018	CCAFS LC-40

10. What is the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) ranked by the highest value between the date 2010-06-04 and 2017-03-20?

```
SELECT landing__outcome, COUNT(landing__outcome) as amount FROM SPACEXTBL
WHERE DATE > '2010-06-04' and DATE < '2017-03-20'
AND landing__outcome LIKE '%Failure (drone ship)%'
OR landing__outcome LIKE '%Success (ground pad)%'
GROUP BY landing__outcome ORDER BY amount DESC
```

landing__outcome	amount
Success (ground pad)	9
Failure (drone ship)	5

Performed predictive analysis

Algorithms

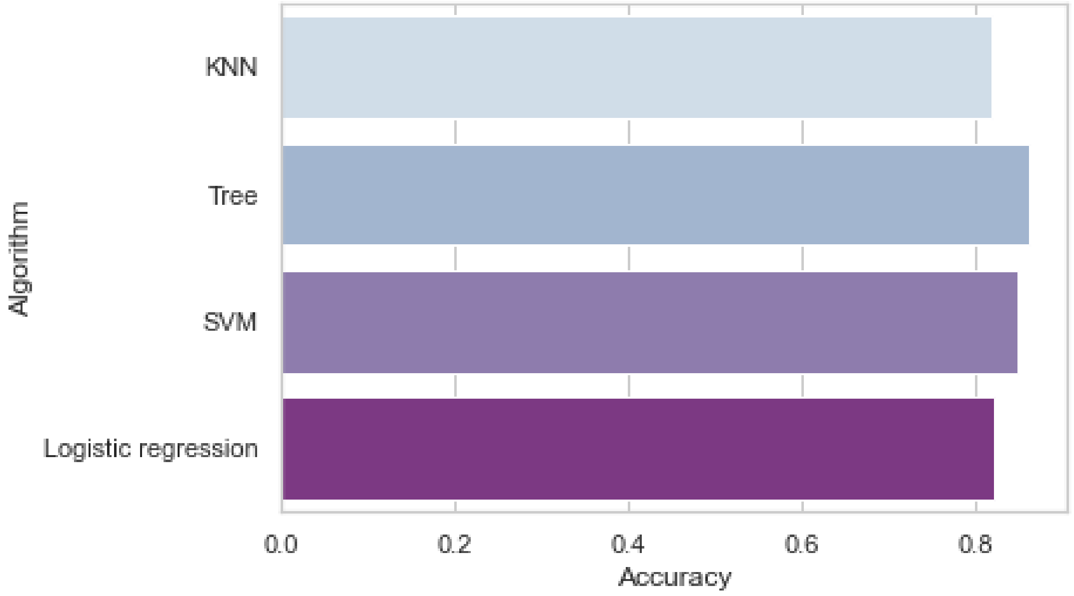
The list of accuracy presents values very close to each other. Although, we can clearly see that the winner is the algorithm Tree with decimals ahead of the others.

```
Best Algorithm is Tree with a score of 0.8625
```

The following are the best parameters for this algorithm.

```
Best Params is : {'criterion': 'gini', 'max_depth': 8,
'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_sam
ples_split': 10, 'splitter': 'best'}
```

	Algorithm	Accuracy
0	KNN	0.819643
1	Tree	0.862500
2	SVM	0.848214
3	Logistic regression	0.821429



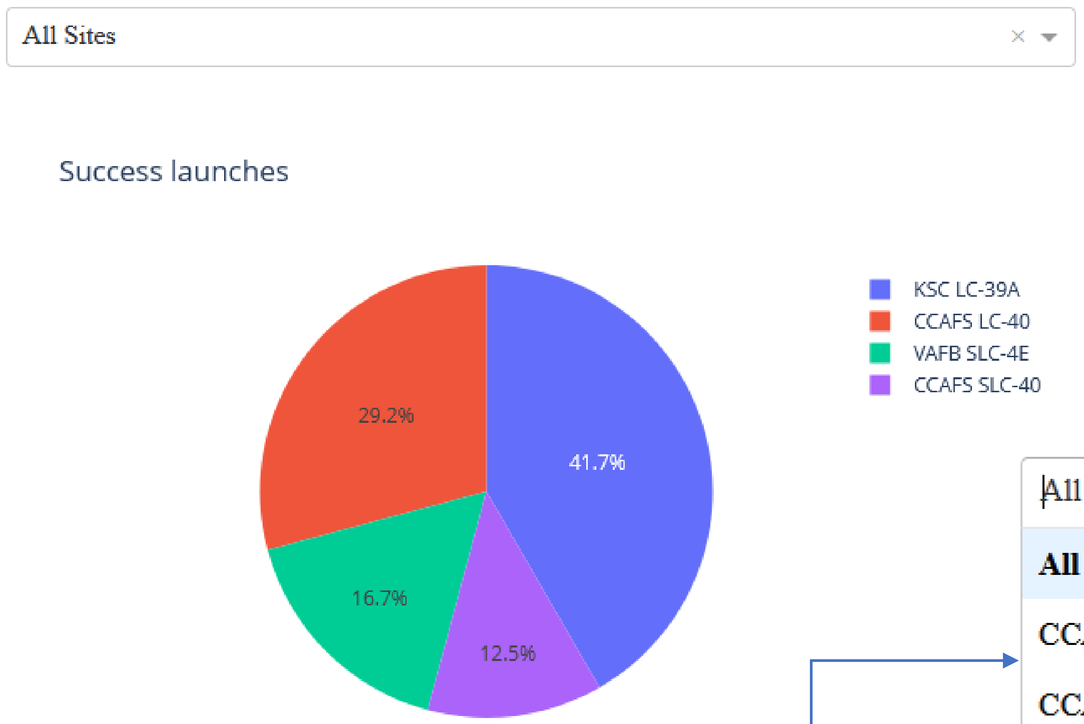
Performed predictive analysis

Confusion Matrix of the Tree algorithm

The matrix shows that the algorithm can identify the different classes. There are no major problems but only the two false negatives, meaning that there is a minimum chance that the algorithm might not identify positive attempts for few launches.

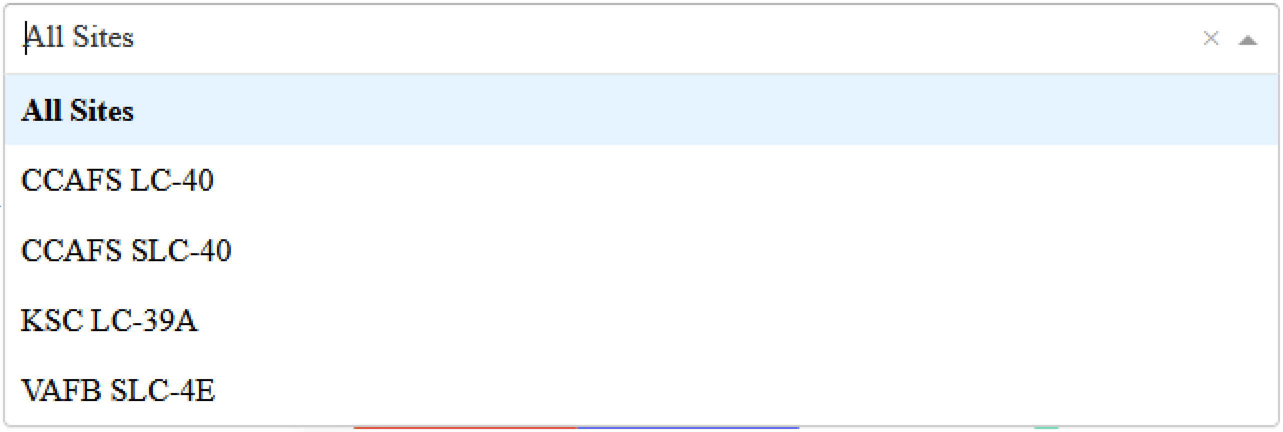
tn	fn
fp	tp

Interactive dashboard screenshots

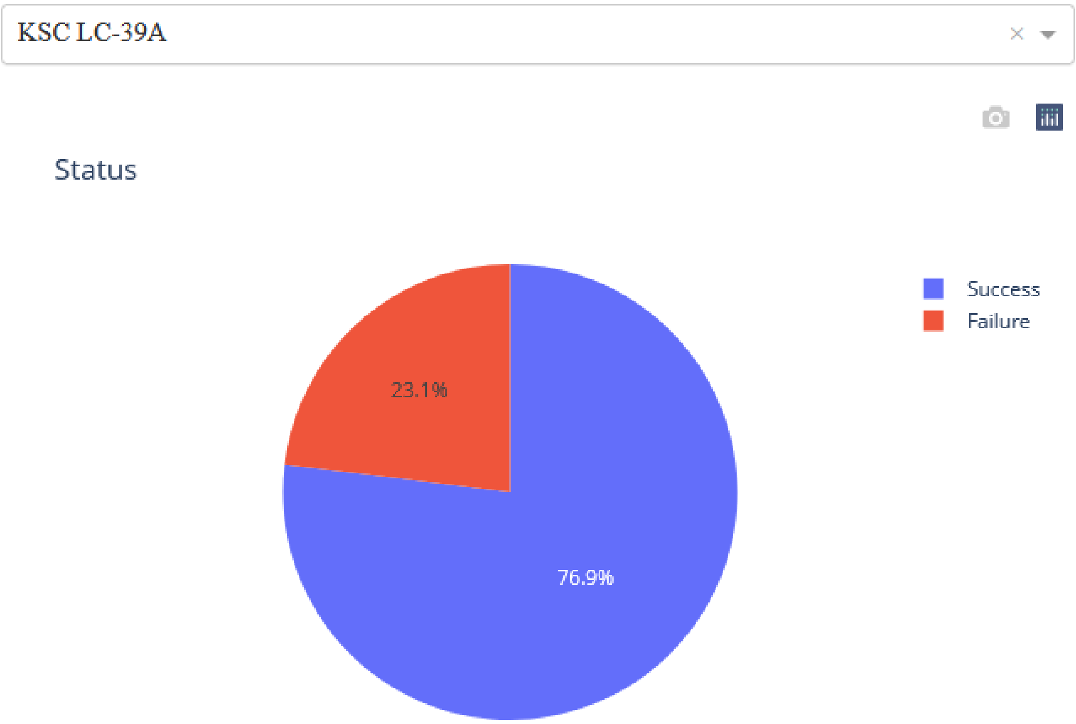


Start

This chart shows the success rate related to launches and all the launch sites.



Interactive dashboard screenshots

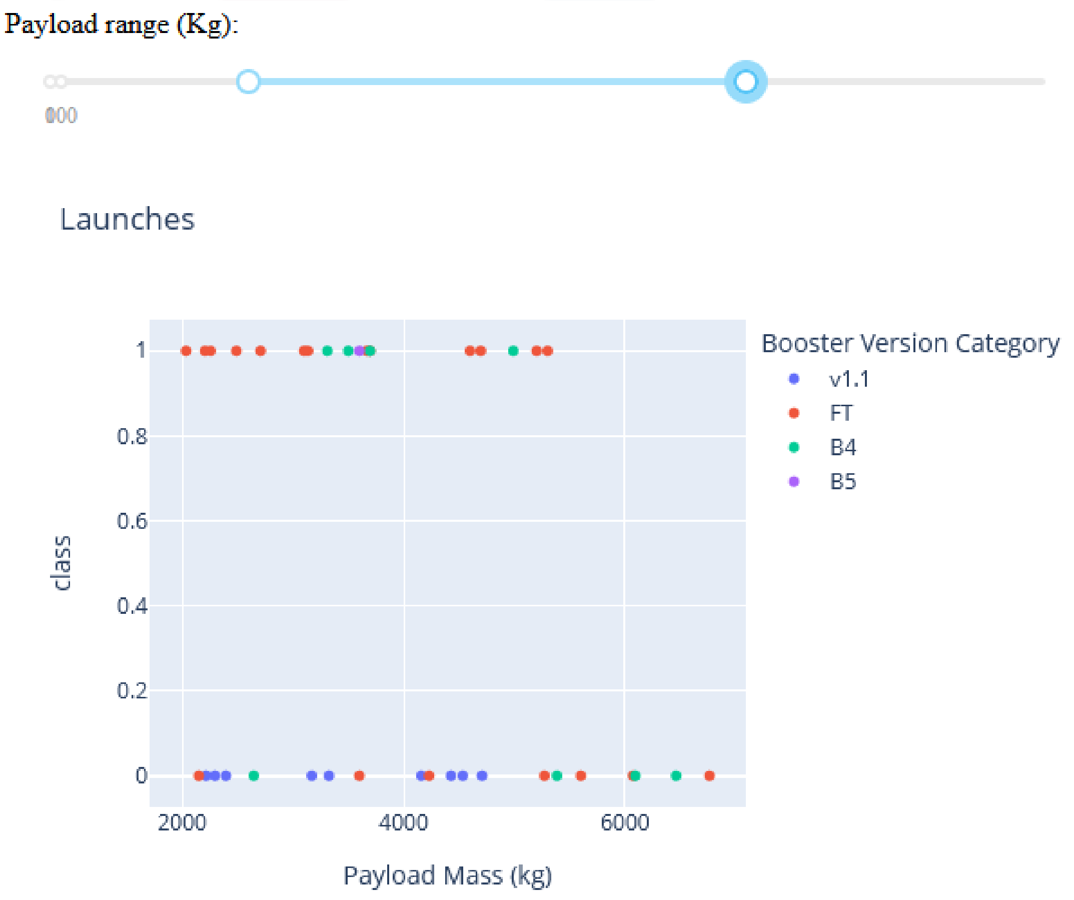


Specific launch site

Status of KSC LC-39A with the highest success rate of all launch sites.

Interactive dashboard screenshots

In this chart the payload between 2k and 6k kilograms had more successful launches with the booster version FT.



Note: Class = Success rate

CONCLUSION

Now we can answer the questions from the start of this presentation:

- Is there any variable that has a big influence in determining if the launch will be successful?
  - The success of the launches of SpaceX relates to the years of experience related on the slide 22. It also is supported by the three variables: Booster version, Orbit, and Payload Mass (Kg).
  - The launch site KSC LC 39A has the highest success rate, thus launch sites also may have an influence on the success rate.
- Is there a correlation between success rate and orbit type?
  - Yes, the orbits ES-L1, SSO, GEO and HEO have the highest success rate having less flights and more successful launches than the others.
- What are the right conditions for the launch to achieve successful results?
  - The success rate of the launch increases when the payload is in between 2k and 6k kilograms. Although other conditions have great impact in the success of the launch like Orbit and Booster version.
  - Most successful launches with payload between 2k and 6k kilograms were using the booster version FT.
- Which algorithm can we use to best predict the success of the launches with this dataset?
  - The Tree Algorithm is the best fit regarding this dataset.