IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Guilherme Mene Ale Primo
10/02/2024

Outline

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

Executive Summary

- Summary of methodologies
- Summary of all results

Introduction

- Project background and context
  - The success and the cost reduction of the Falcon 9 from the SPACEX aims to the capacity of reusage of the first stage during launch. This can represent a reduction of ~100 million dollars for each launch to the aerial space program. To cover some technical aspects of that activity, this report summarize the prediction of the Falcon 9 first stage reusability. The results of that kind of the prediction can represent a significant cost reduction to the rocket company.
- Problems you want to find answers
  - Can launch and landing data provide good information about future successful ground landing attempts?
  - What are the main conditions to predict the land of a launched rocket ?
  - What are the conditions which will provide a successful land and what are the relationship with the rocket effects and variables provided from past lands data?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Web scraping from Falcon 9 table (Wikipedia)
  - REST API (SpaceX)
- Perform data wrangling
  - Handling the missing values
  - Remove unnecessary data from data frames
  - Encoding the categorical data
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash

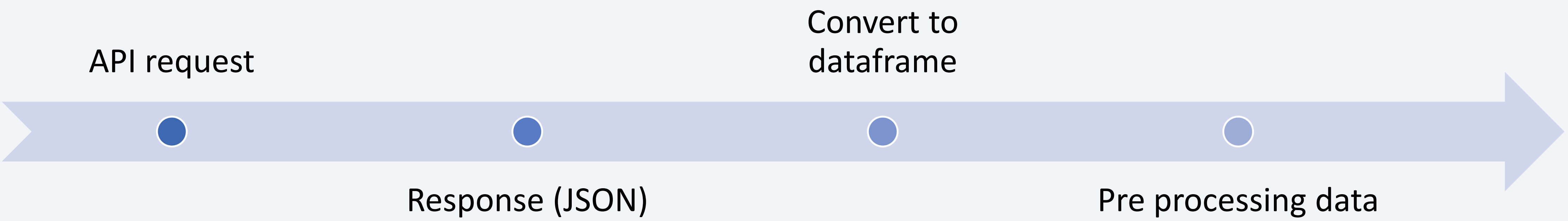
Methodology

Executive Summary

- Perform predictive analysis using classification models
  - The classification models were built using scikit-learn library
  - The tune process consisted of using GridSearchCV to parameterize the hyperparameters
  - The evaluation of the models were carried out using score and accuracy methods using the validation dataset

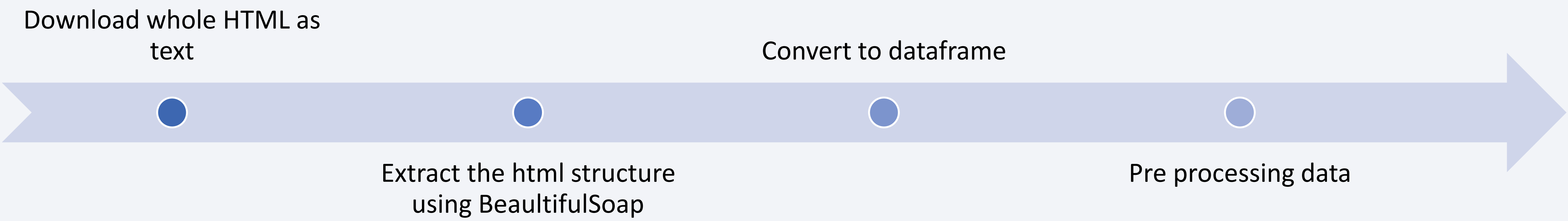
Data Collection

- Describe how data sets were collected.
  - The REST API data were retrieved using the link api.spacexdata.com/v4/ and the process consisted of:



Data Collection

- Describe how data sets were collected.
  - The Webscrapping data were retrieved using the link https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922 and the process consisted of:



Data Collection – SpaceX API

Code

Make request

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

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

Convert response Json to dataframe

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

Transform data

```python
# Call getLaunchSite
getLaunchSite(data)
```

```python
# Call getPayloadData
getPayloadData(data)
```

```python
# Call getCoreData
getCoreData(data)
```

Create dictionary of data

```python
launch_dict = {'FlightNumber': list(data['flight_number']),
'Date': list(data['date']),
'BoosterVersion':BoosterVersion,
'PayloadMass':PayloadMass,
'Orbit':Orbit,
'LaunchSite':LaunchSite,
'Outcome':Outcome,
'Flights':Flights,
'GridFins':GridFins,
'Reused':Reused,
'Legs':Legs,
'LandingPad':LandingPad,
'Block':Block,
'ReusedCount':ReusedCount,
'Serial':Serial,
'Longitude': Longitude,
'Latitude': Latitude}
```

Convert dict to dataframe

```python
# Create a data from launch_dict
df = pd.DataFrame.from_dict(launch_dict)
```

Select specific data from dataframe

```python
# Hint data['BoosterVersion']!='Falcon 1'
data_falcon9 = df[df['BoosterVersion'] != 'Falcon 1']
```

```python
data_falcon9.loc[:,'FlightNumber'] = list(range(1, data_falcon9.shape[0]+1))
data_falcon9
```

Handling missing values

```python
# Calculate the mean value of PayloadMass column
PayloadMassMean = data_falcon9['PayloadMass'].mean()
# Replace the np.nan values with its mean value
data_falcon9['PayloadMass'].replace(np.nan, PayloadMassMean, inplace=True)
```

Save the final file

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

Data Collection - Scraping

Code

Request HTML text

```
response = requests.get(static_url).text
```

Convert to soup

```
soup = BeautifulSoup(response, 'html5lib')
```

Get column names

```
column_names = []

for row in first_launch_table.find_all('th'):
    name = extract_column_from_header(row)
    if (name != None and len(name) > 0):
        column_names.append(name)
```

Create dictionary

```
launch_dict= dict.fromkeys(column_names)

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

# Let's initial the launch_dict with each
launch_dict['Flight No.'] = []
launch_dict['Launch site'] = []
launch_dict['Payload'] = []
launch_dict['Payload mass'] = []
launch_dict['Orbit'] = []
launch_dict['Customer'] = []
launch_dict['Launch outcome'] = []
# Added some new columns
launch_dict['Version Booster']=[]
launch_dict['Booster landing']=[]
launch_dict['Date']=[]
launch_dict['Time']=[]
```

Extract table

```
extracted_row = 0
#Extract each table
for table_number,table in enumerate(soup.find_all('table',"wikitable plainrowheaders collapsible")):
   # get table row
    for rows in table.find_all("tr"):
        #check to see if first table heading is as number corresponding to launch a number
        if rows.th:
            if rows.th.string:
                flight_number=rows.th.string.strip()
                flag=flight_number.isdigit()
        else:
            flag=False
        #get table element
        row=rows.find_all('td')
        #if it is number save cells in a dictonary
        if flag:
            extracted_row += 1
            # Flight Number value

            # TODO: Append the flight_number into launch_dict with key `Flight No.`
            launch_dict['Flight No.'].append(flight_number)
            #print(flight_number)
            datatimelist=date_time(row[0])

            # Date value
            # TODO: Append the date into launch_dict with key `Date`
            date = datatimelist[0].strip(',')
            launch_dict['Date'].append(date)
            #print(date)
```

Convert to dataframe

```
df= pd.DataFrame({ key:pd.Series(value) for key, value in launch_dict.items() })
```

Save data

```
df.to_csv('spacex_web_scraped.csv', index=False)
```

Data Wrangling

Code

- There are several land types on dataframe representing the different land status, outcomes and sites.
- According to the outcome of the land type, we need transform the categorical to numeric data and save the dataframe:

Launch Site count

```
# Apply value_counts() on column LaunchSite
df['LaunchSite'].value_counts()
```

```
LaunchSite
CCAFS SLC 40    55
KSC LC 39A      22
VAFB SLC 4E     13
Name: count, dtype: int64
```

Orbit type count

```
# Apply value_counts on Orbit column
df['Orbit'].value_counts()
```

```
Orbit
GTO      27
ISS      21
VLEO     14
PO        9
LEO       7
SSO       5
MEO       3
ES-L1     1
HEO       1
SO        1
GEO       1
Name: count, dtype: int64
```

Outcome count

```
# landing_outcomes = values on Outcome column
landing_outcomes = df['Outcome'].value_counts()
landing_outcomes
```

```
Outcome
True ASDS      41
None None      19
True RTLS      14
False ASDS      6
True Ocean      5
False Ocean     2
None ASDS       2
False RTLS      1
Name: count, dtype: int64
```

Categorical to numeric transformation

```
# landing_class = 0 if bad_outcome
# landing_class = 1 otherwise
landing_class = []
for i, row in df.iterrows():
    if row['Outcome'] in bad_outcomes:
        landing_class.append(0)
    else:
        landing_class.append(1)
```

Save the file

```
df.to_csv("dataset_part_2.csv", index=False)
```

EDA with Data Visualization

[Code]

- To understand the data structure and the variables therein some plots were created to preview the database features:
  - Scatter Plots
    - Flight Number vs Payload Mass (using sns.catplot)
    - Flight Number vs Launch Site (using sns.catplot)
    - Payload vs Launch Site (using sns.scatterplot)
    - Flight Number vs Orbit (using sns.scatterplot)
    - Payload Mass vs Orbit (using sns.scatterplot)
  - Bar Graph
    - Showing the success rate of each orbit
  - Line Plot
    - Showing the success rate over the years

EDA with SQL

Code

- The SQL queries were used to retrieve more information about the database and help to understand the data:
  - Display the names of the unique launch sites in the space mission
  - Display 5 records where launch sites begin with the string 'CCA'
  - Display the total payload mass carried by boosters launched by NASA (CRS)
  - Display average payload mass carried by booster version F9 v1.1
  - List the date when the first successful landing outcome in ground pad was achieved.
  - List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
  - List the total number of successful and failure mission outcomes
  - List the names of the Booster Versions which have carried the maximum payload mass.

Build an Interactive Map with Folium

Code

- Folium map was created with the NASA Johnson Space Center and the list below summarize the all task performed using the folium along the data:
  - Mark all launch sites on a map
    - Folium Circle and Markers with Launch Sites
  - Mark the success/failed launches for each site on the map
    - The success and failed launches were plotted using Folium Cluster Markers. The green was used for success and the red for failed launches.
  - Calculate the distances between a launch site to its proximities
    - The distance calculation using two coordinates points and plotting using Folium Polyline

Build a Dashboard with Plotly Dash

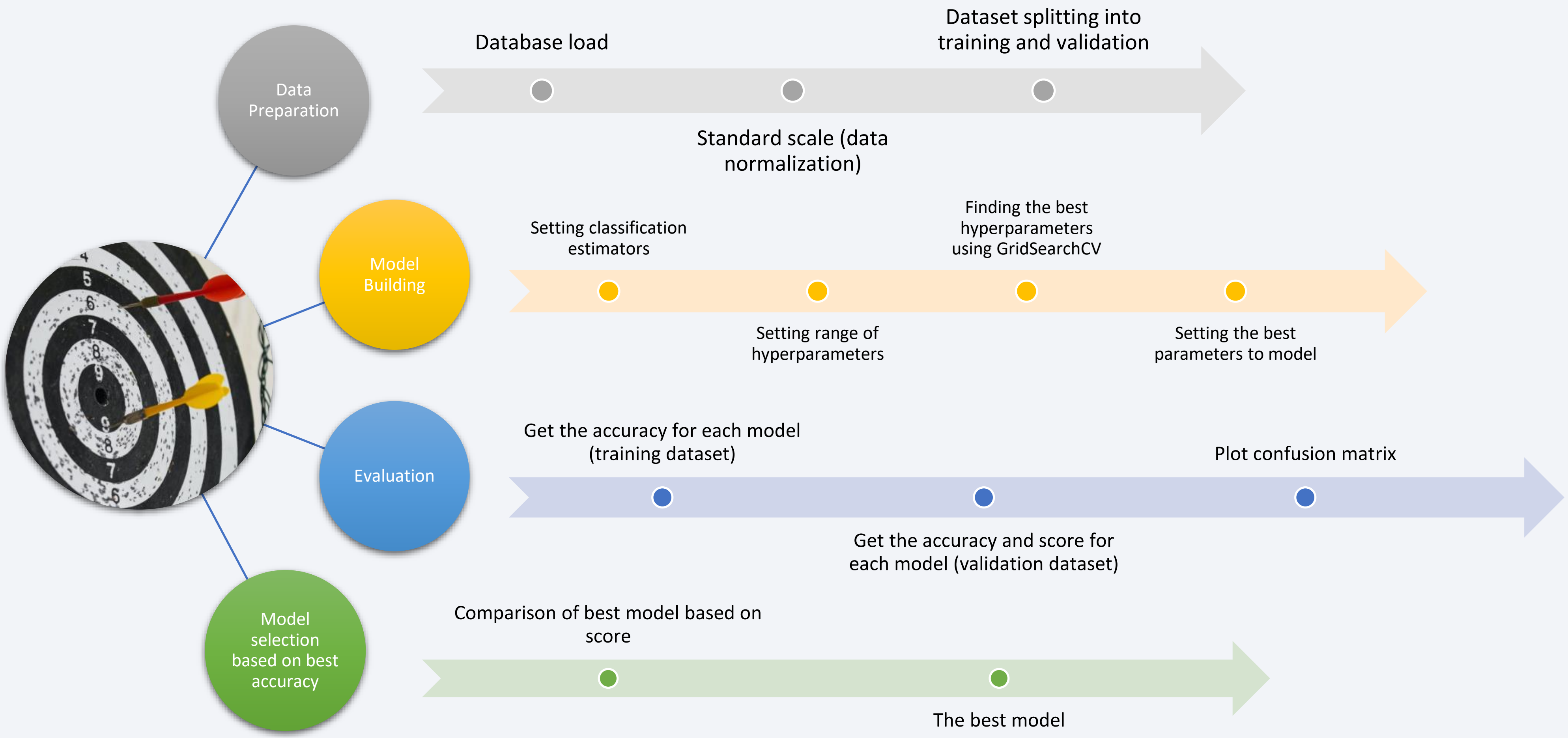
Code

- The Dashboard have four main components:
  - The dropdown show the launch site options
  - The pie chart summarizes the success rate of each launch sites
  - The correlation between payload mass and success rate is demonstrated in the scatter plot with a range slider that's provides the capability to select the payload mass range

Predictive Analysis (Classification)

Code

The prediction of the landing condition were separated on four steps:



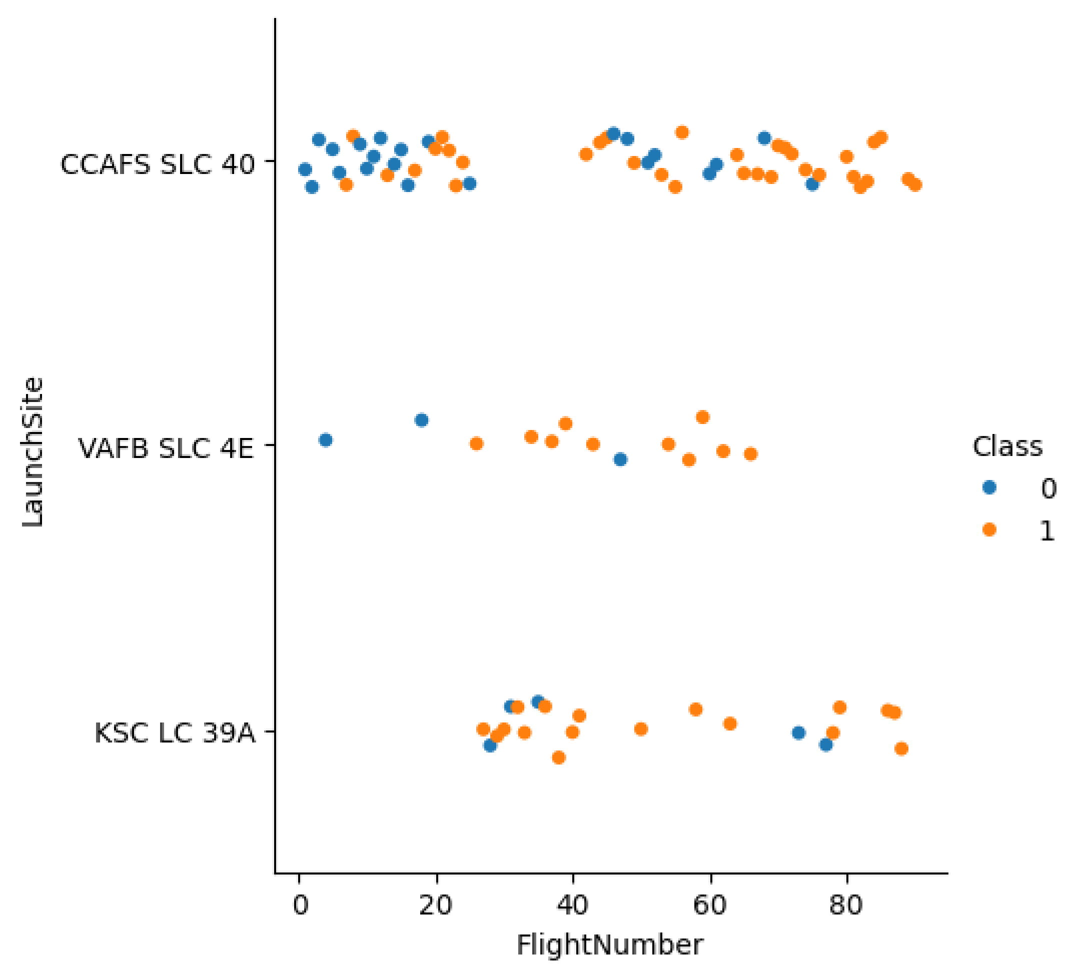
Results

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

Section 2

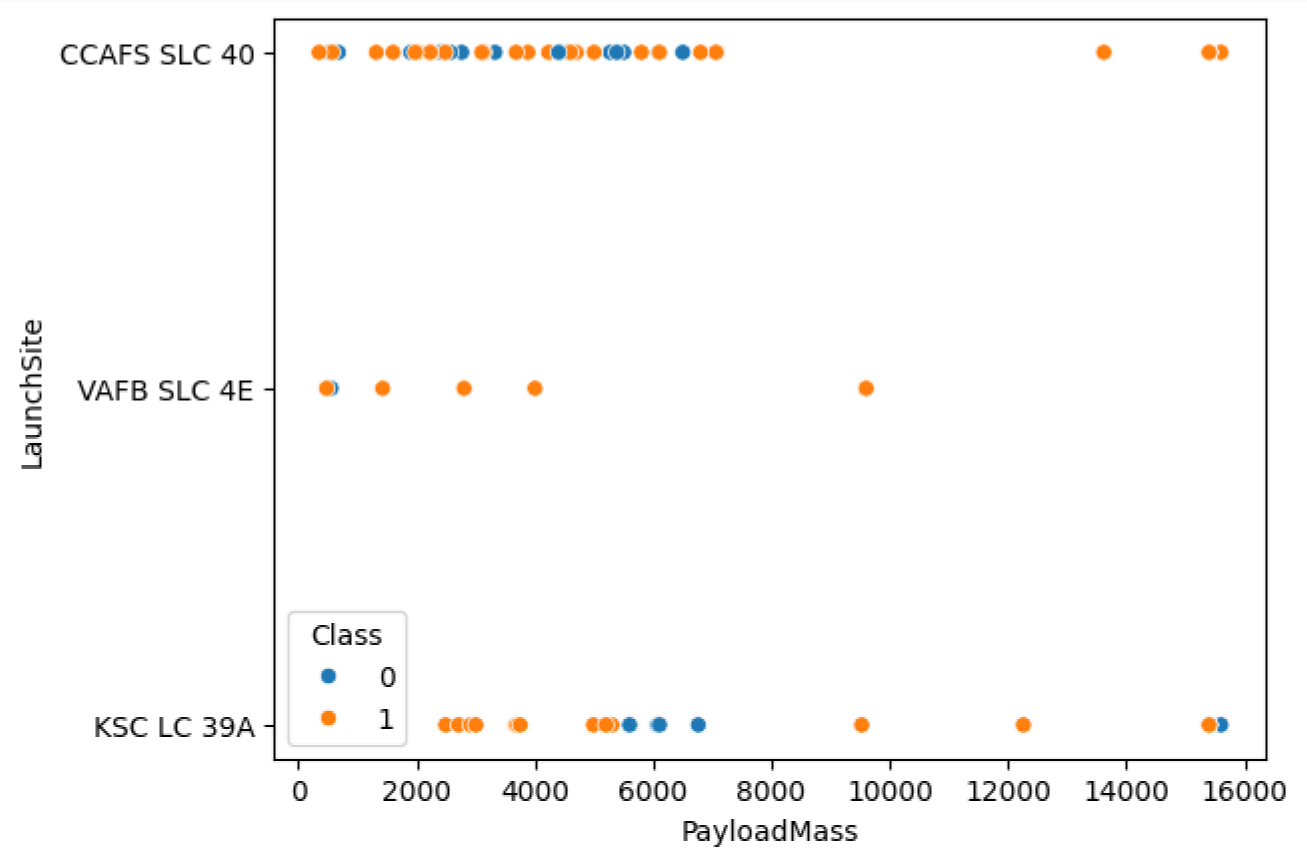
Insights drawn from EDA

Flight Number vs. Launch Site



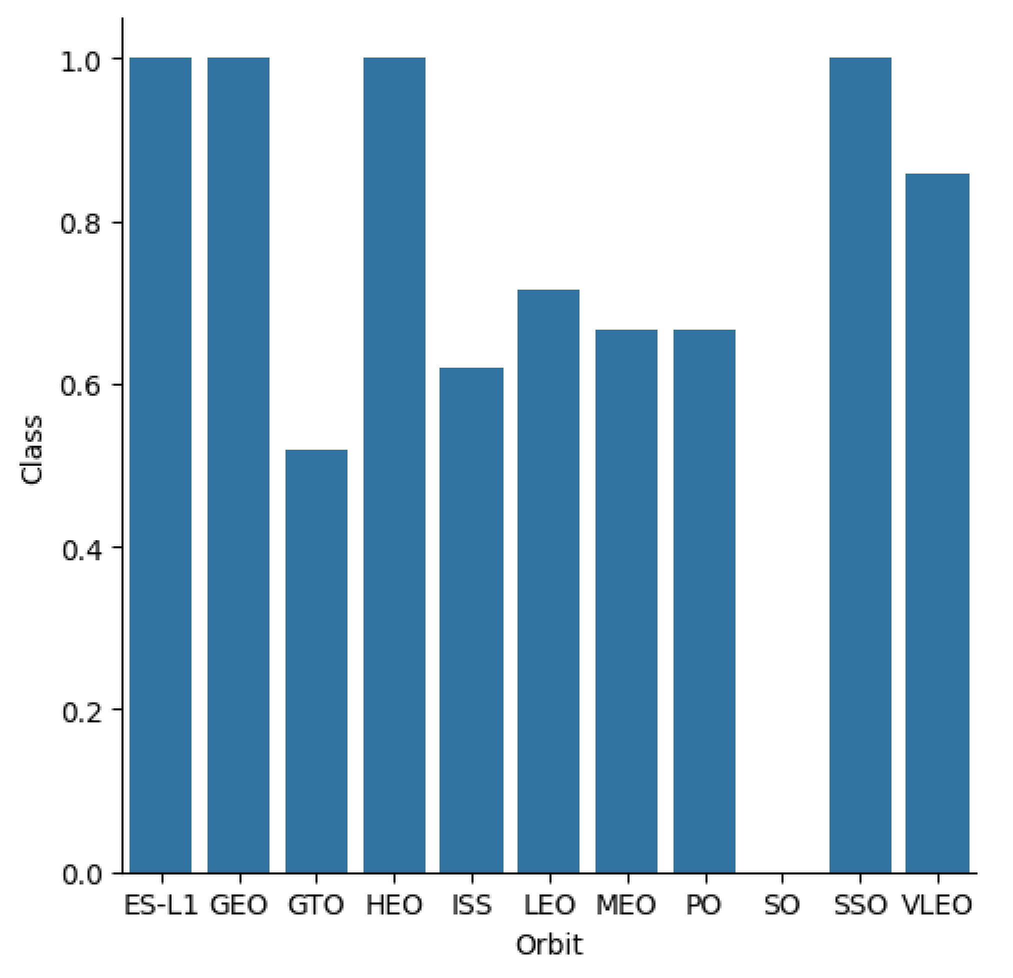
- In recent flights the number of success rates has been increased, especially at the VAFB SLC 4E launch site.

Payload vs. Launch Site



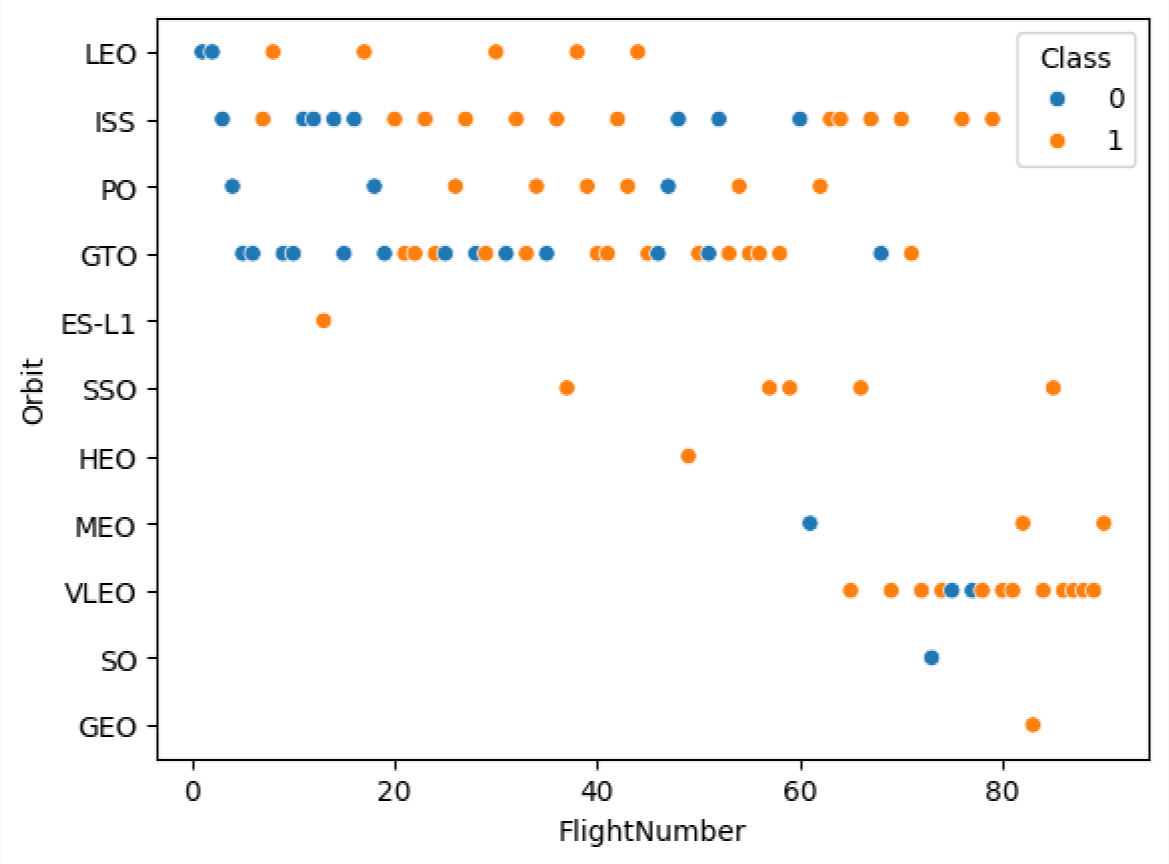
- The CCAFS SLC 40 Launch Site show good successful rate with heavy payloads
- The VAFB SLC 4E show good payload ranging between ~1000 to 10000 Kg with major successful rate
- The KSC LC 39A varies on payload mass vs success rate

Success Rate vs. Orbit Type



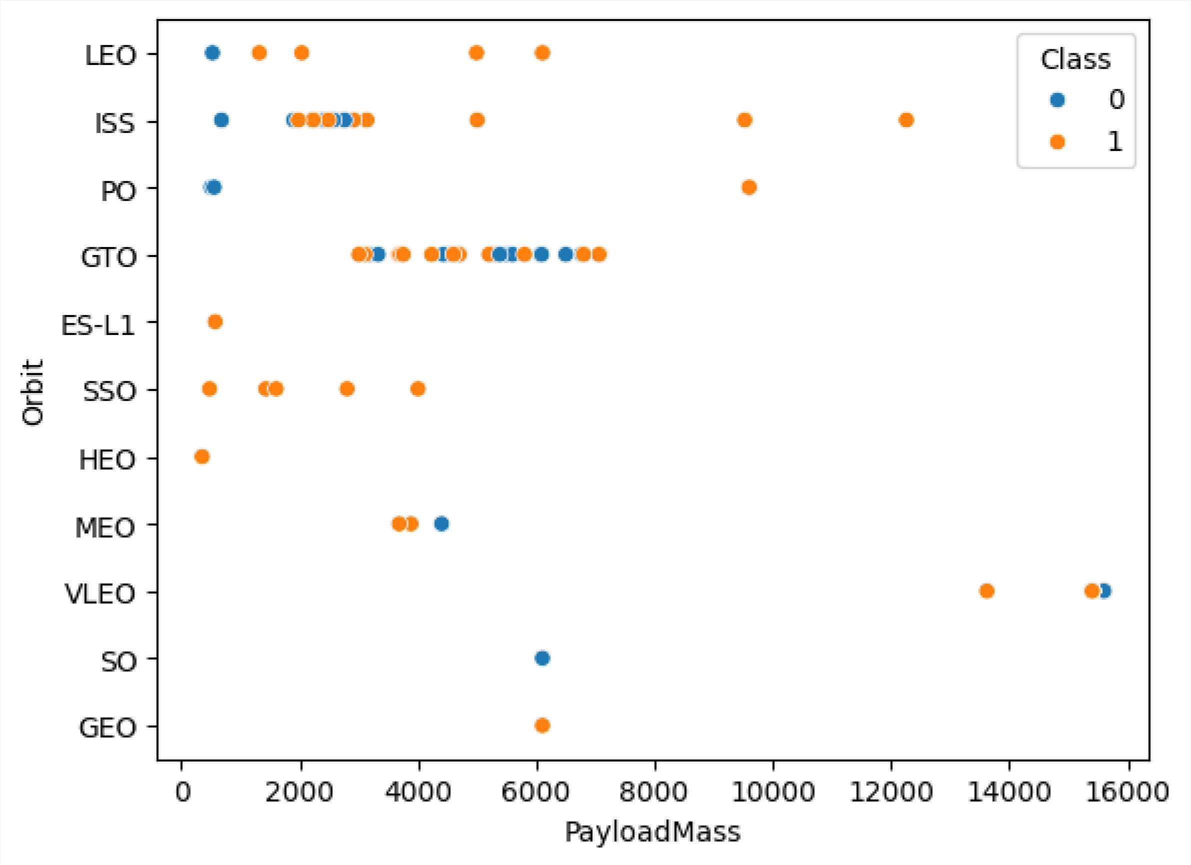
- The ES-L1, GEO, HEO, and SSO shows the most high success rate against the others orbits
- The VLEO represent a success rate ~ 85%
- The GTO, ISS, LEO, MEO, and PO exhibit the lower success rate, ranging from ~ 50 to 70 %

Flight Number vs. Orbit Type



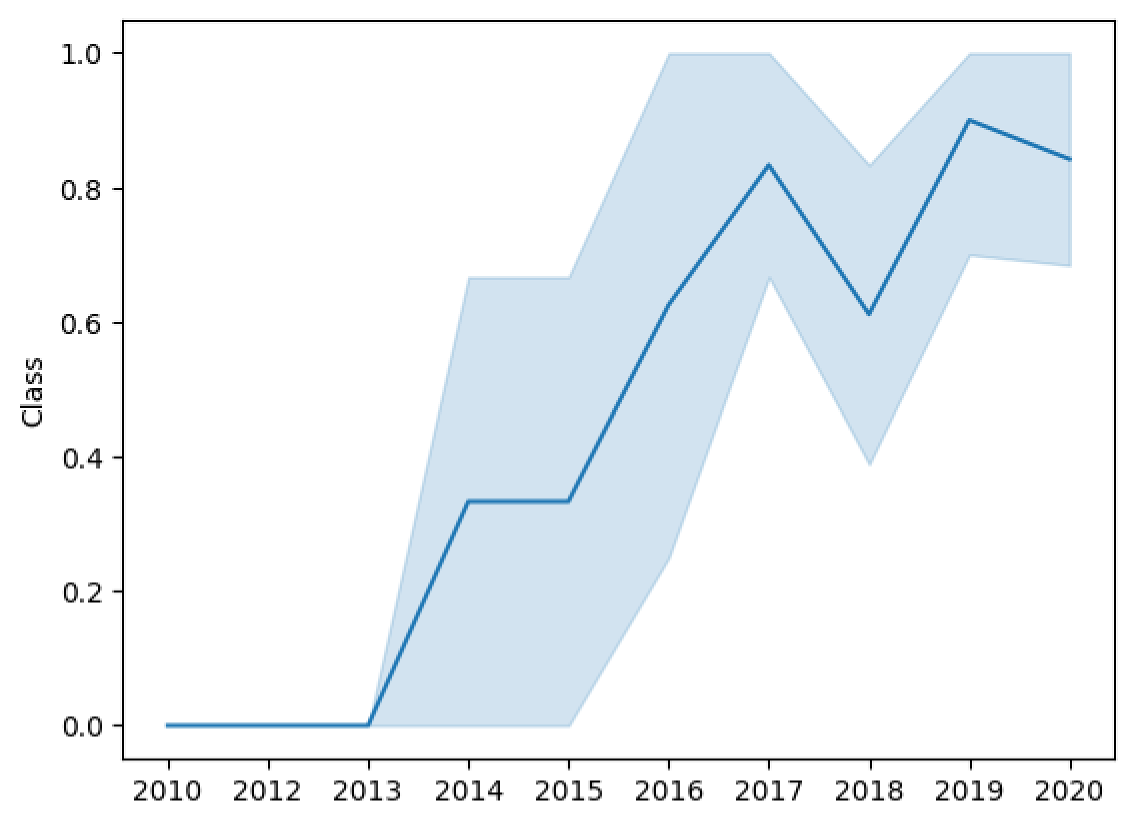
- The success rate were increased to almost orbit types except the GTO, VLEO, and SO.
- A concern about SO and GEO is that there are too few samples for more accurate analysis

Payload vs. Orbit Type



- The payload mass can drive to a success rate over orbit types. For example to the ISS and LEO, there are heavy payload with good rate of successful land

Launch Success Yearly Trend



- The success rate from the SpaceX rockets were increased since 2013. However, a large increase in this success rate occurred by 2015, with two peaks in 2017 and 2019.

All Launch Site Names

SQL Query

```
%sql select distinct "Launch_Site" from SPACEXTABLE
```

Results

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

- Using select method to gather all unique values (distinct) from launch site column

Launch Site Names Begin with 'CCA'

SQL Query

```
1  %sql select "Launch_Site" from SPACEXTABLE where "Launch_Site" like '%CCA%' limit 5
```

Results

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

- Using select method to gather 5 rows (limit 5) where Launch Site ‘like’ CCA

Total Payload Mass

SQL Query

```
1  %sql select sum(PAYLOAD_MASS__KG_) as TotalPayloadMass_kg from SPACEXTABLE where "Payload" like '%CRS%'
✓ 0.0s
```

Results

TotalPayloadMass_kg
111268

- Return the sum of the Payload as TotalPayloadMass_kg
- Using where Payload 'like' CRS

Average Payload Mass by F9 v1.1

SQL Query

```
%sql select avg(PAYLOAD_MASS__KG_) as Avg_Payload_kg from SPACEXTABLE where "Booster_Version" like '%F9 v1.1%'
```

Results

Avg_Payload_kg
2534.6666666666665

- Return the average of Payload using Booster version ‘like’ F9 v1.1

First Successful Ground Landing Date

SQL Query

```
%sql select min(DATE) from SPACEXTABLE where "Landing_Outcome" like '%Success%'
```

Results

min(DATE)
2015-12-22

- Return the minimum date for landing outcome with 'success' in the value of the cells

Successful Drone Ship Landing with Payload between 4000 and 6000

SQL Query

```
%%sql select "Booster_Version"
    from SPACEXTABLE
    where "Landing_Outcome"
    like '%Success (drone ship)%' and
    "PAYLOAD_MASS__KG_" between 4000 and 6000
```

Results

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

- Return the Booster version
- Using where landing outcome have 'Success (drone ship)' on the values and payload ranges from 4000 and 6000

Total Number of Successful and Failure Mission Outcomes

SQL Query

```
%%sql select "Mission_Outcome", count("Mission_Outcome")
    as Count
    from SPACEXTABLE
    group by "Mission_Outcome"
```

Results

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

- Return the Mission outcome with count of the mission outcome status grouping by mission outcome

Boosters Carried Maximum Payload

SQL Query

```
%%sql select "Booster_Version"
    from SPACEXTABLE
    where "PAYLOAD_MASS__KG_" =
    (select max("PAYLOAD_MASS__KG_") from SPACEXTABLE)
```

Results

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
F9 B5 B1051.6
F9 B5 B1060.3
F9 B5 B1049.7

- Return the boosters with maximum payload using sub query

2015 Launch Records

SQL Query

```
%%sql select substr(Date, 6,2) as Month, "Landing_Outcome", "Booster_Version", "Launch_Site"
    from SPACEXTABLE
    where "Landing_Outcome" like '%Failure (drone ship)%' and
    substr(Date, 0,5)='2015'
```

Results

Month	Landing_Outcome	Booster_Version	Launch_Site
01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

- Return the records with their month, failure landing outcome, booster version and launch site for the launches from 2015

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

SQL Query

```
%%sql select "Landing_Outcome", count(*) as Count
    from SPACEXTABLE
    where Date between '2010-06-04' and '2017-03-20' and
    "Landing_Outcome" like '%Failure (drone ship)%' or
    "Landing_Outcome" like '%Success (ground pad)%'
    group by "Landing_Outcome"
    order by Count desc
```

Results

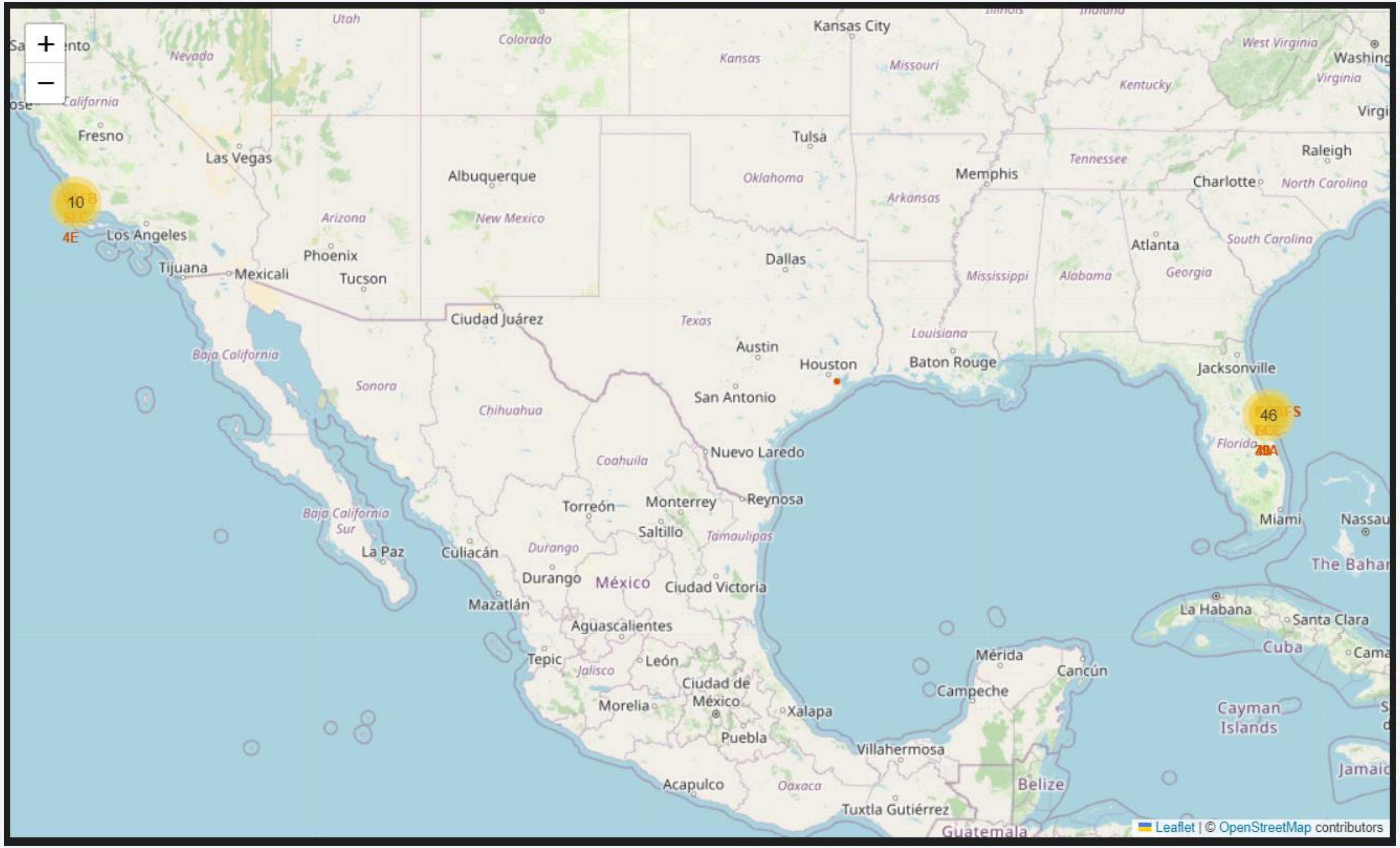
Landing_Outcome	Count
Success (ground pad)	9
Failure (drone ship)	5

- Return the count of the success (ground pad) and Failure (drone ship) records from DB

Section 3

Launch Sites Proximities Analysis

Folium Map Launch Sites

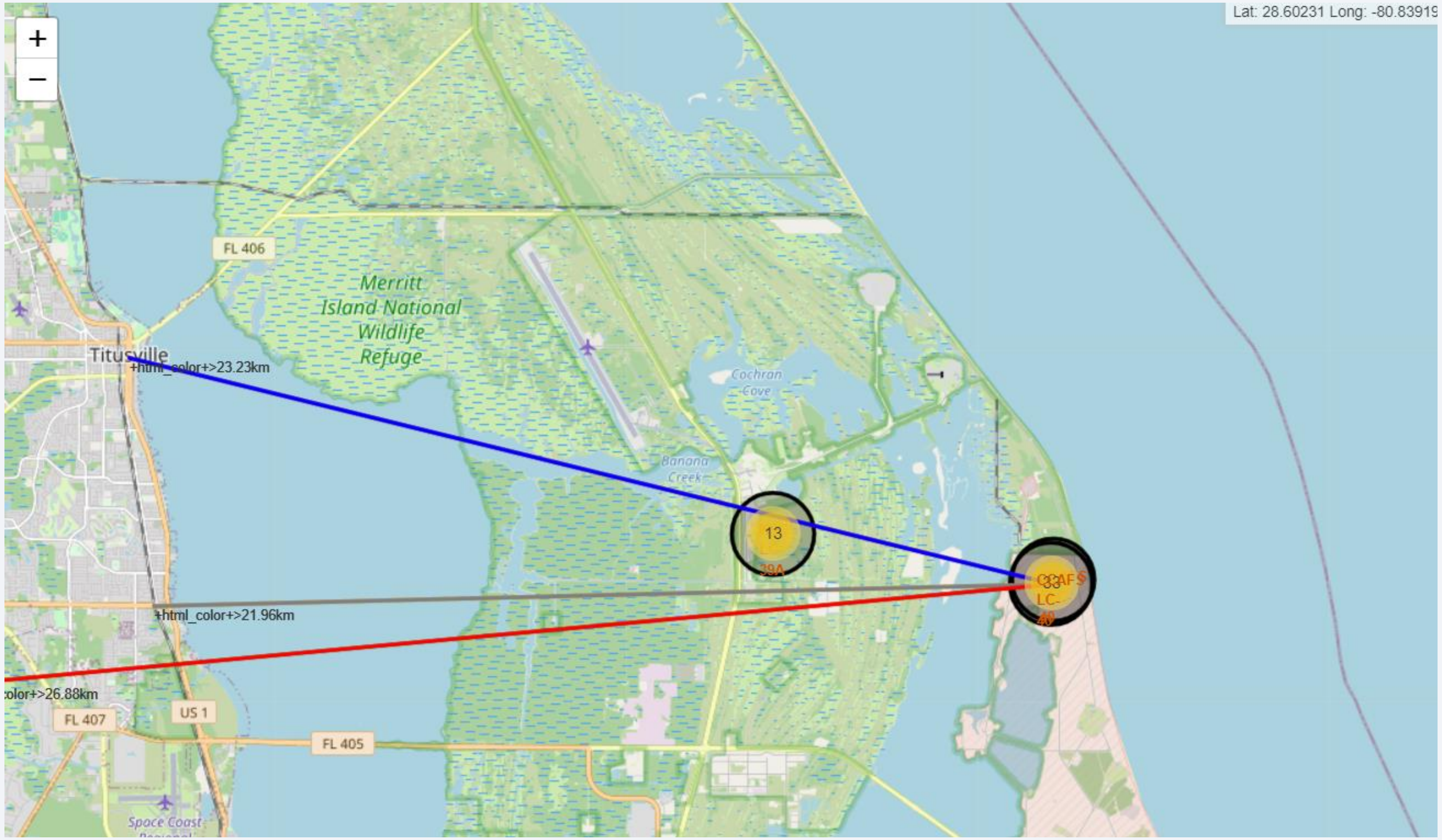


- The main launch sites of the US are from coast lines of Miami and California

Folium Map – Success / Failed Launches

- The Launch Site with higher success rate is KSC LC-39A

The Launch Sites facilities

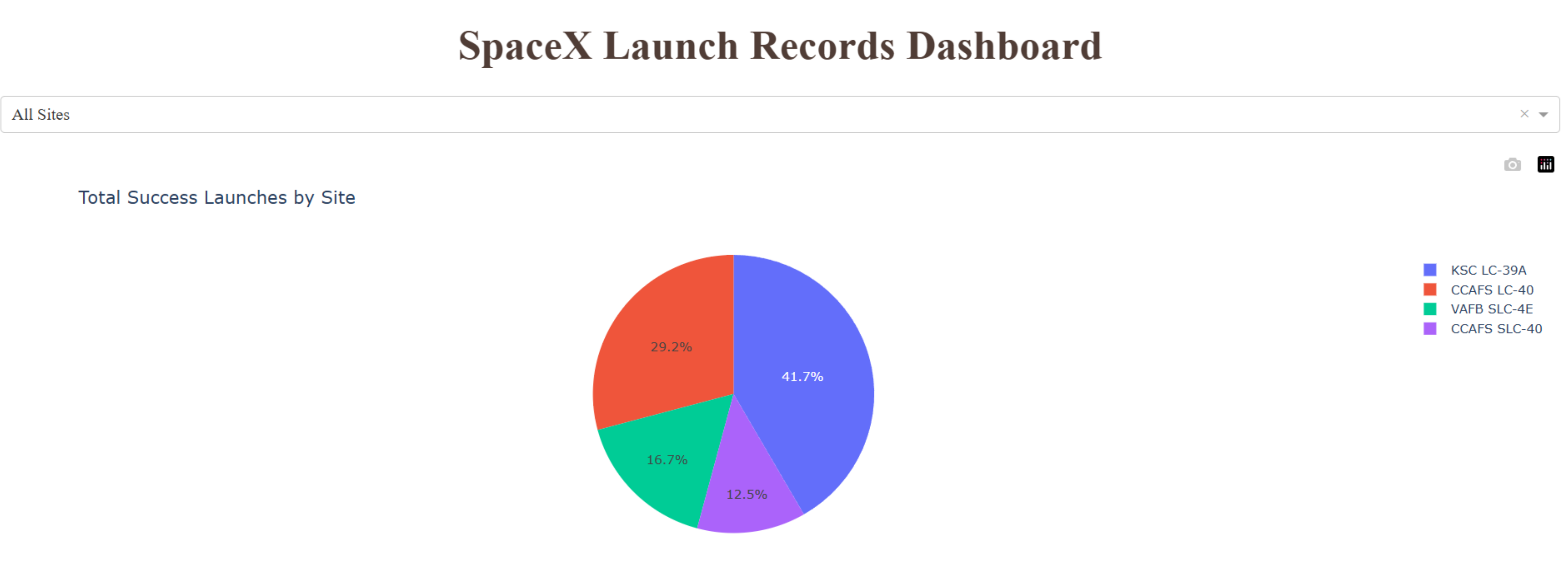


- The chosen launch site was the CCAFS SLC-40 that's:
- Almost 0.86 km from coast line
- ~ 22km from nearest railway
- ~ 23 km from nearest city, and
- ~ 27 km from nearest highway

Section 4

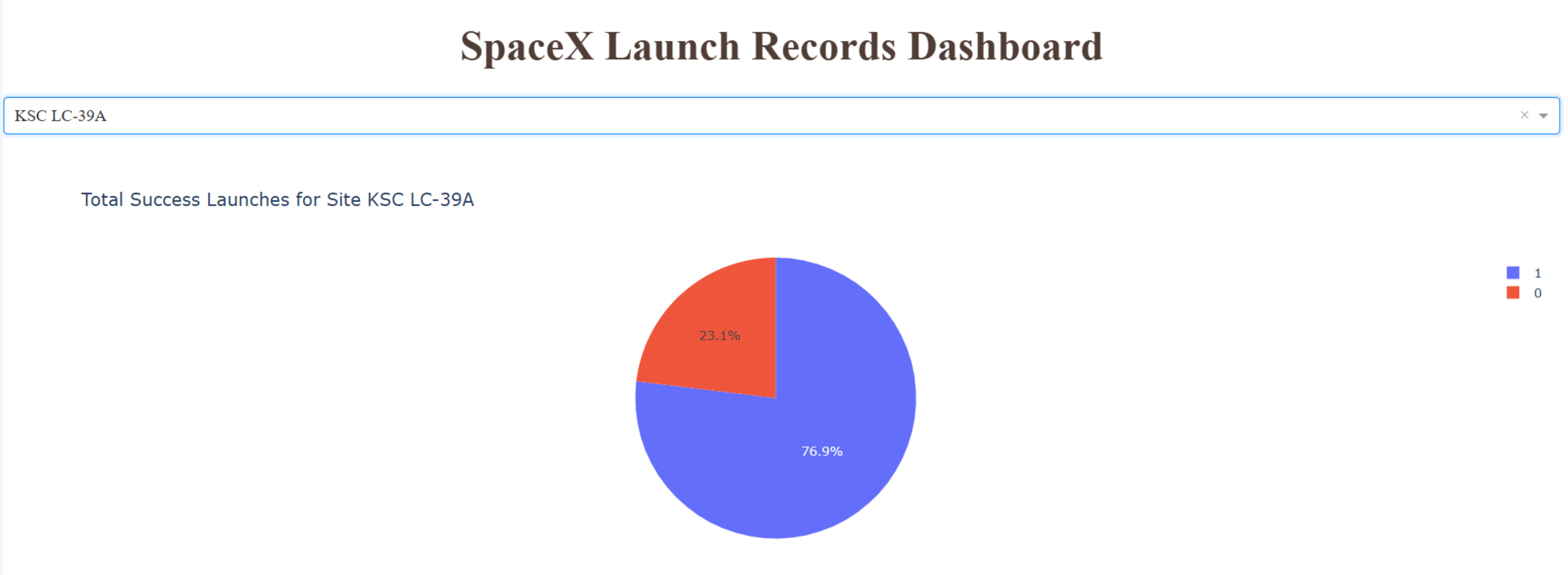
Build a Dashboard with Plotly Dash

Successful rate by Launch Sites



The best Launch Site was the KSC LC-39^{a} with 41.7%

The KSC LC-39A Launch Site



The KSC LC-39[a] shows 76.9% of success rate

Correlation between Payload Mass vs Success Rate

The best range of payload for the success rate are from 2000 kg to ~ 5300 kg

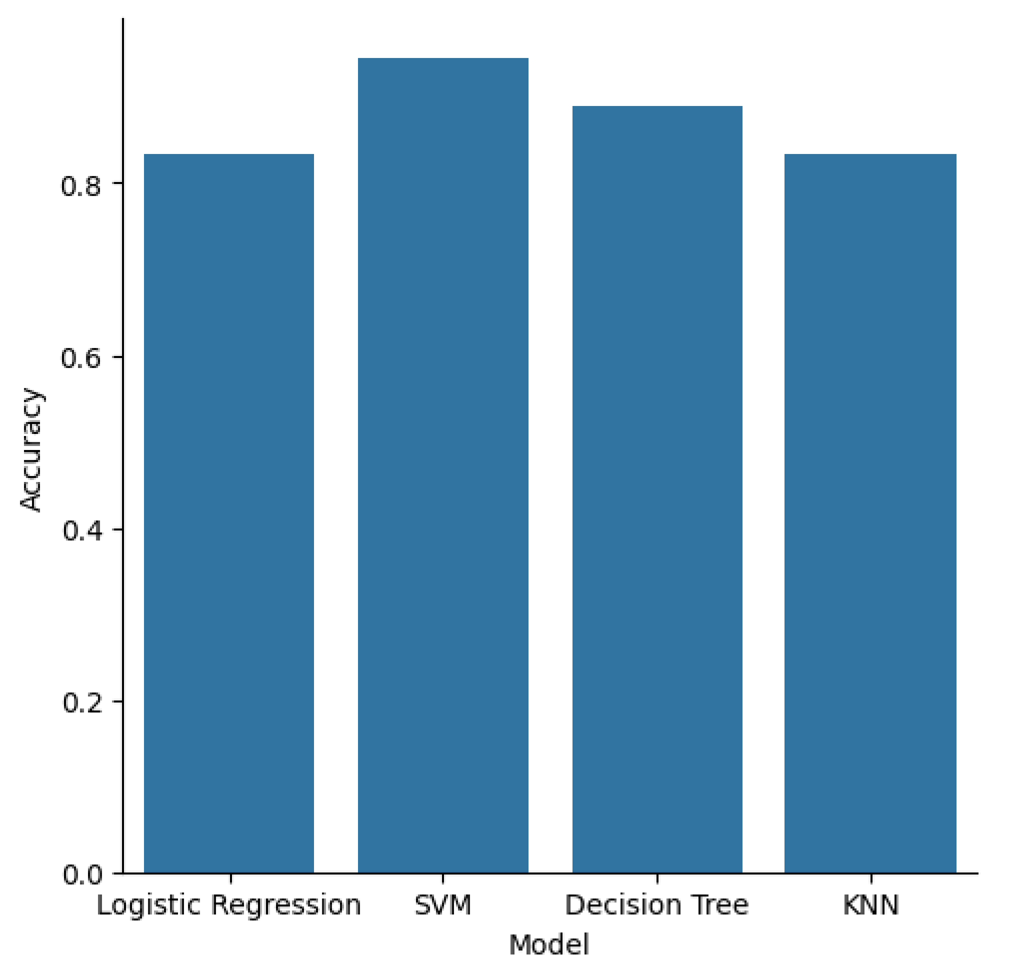
Section 5

Predictive Analysis (Classification)

Classification Accuracy

	Model	Accuracy
0	Logistic Regression	0.833333
1	SVM	0.944444
2	Decision Tree	0.888889
3	KNN	0.833333

The best model accuracy was calculated from the SVM model with 0.944 and the second high accuracy was from Decision Tree with 0.888

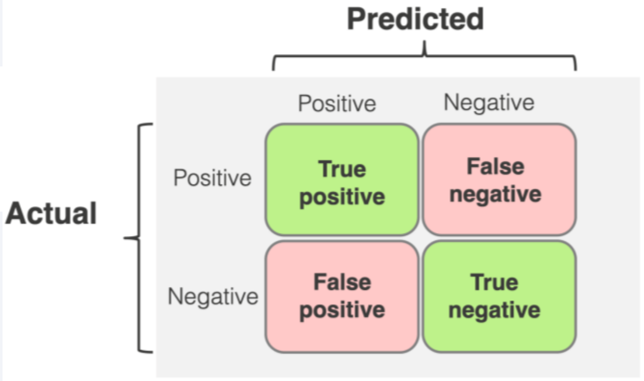


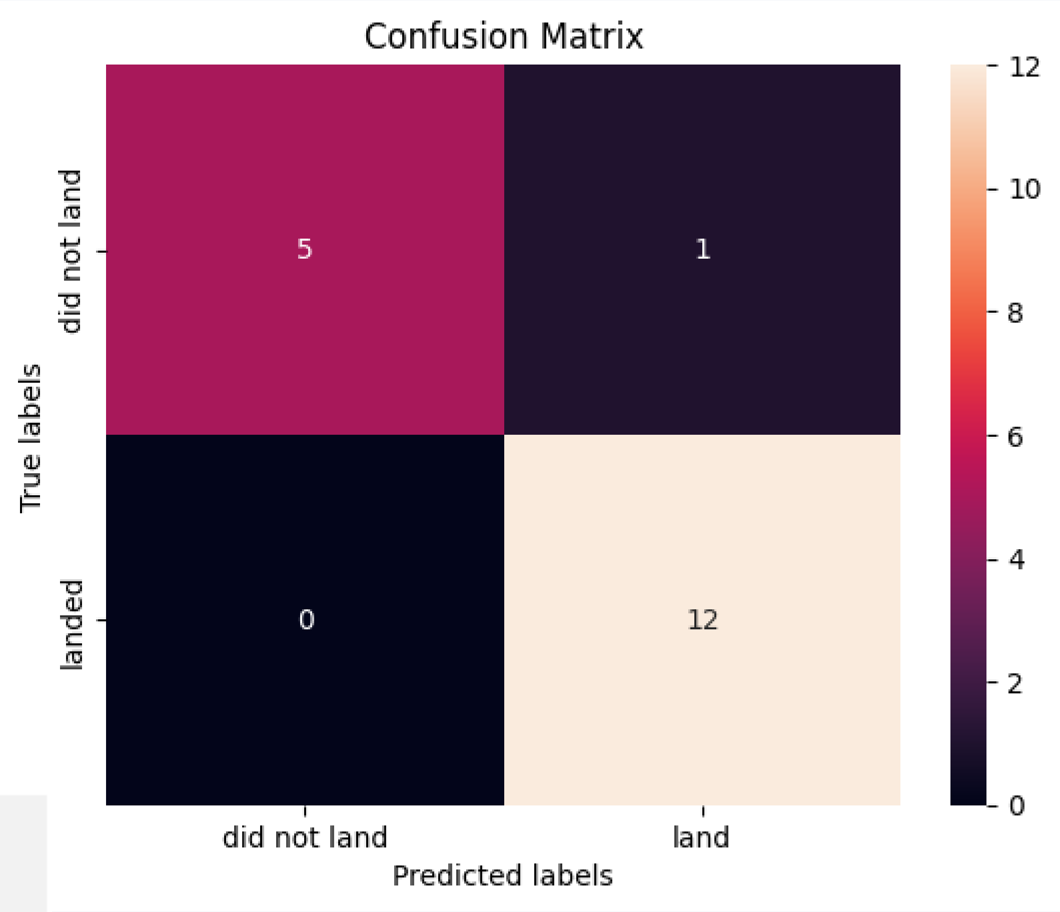
Confusion Matrix

The SVM model shows good accuracy when we visualize the confusion matrix. There is only 1 false negative prediction, while the true positive and negative measurements account for 17.

The accuracy is calculated using the true positive, true negative, false positive, and false negative from the model using the equation below:

$$\text{Accuracy} = \frac{\sum \text{TP} + \text{TN}}{\sum \text{TP} + \text{FP} + \text{FN} + \text{TN}}$$





Conclusions

- The datasets provided a good structure for the Data Analysis of the SpaceX landing and launches
- During the EDA process we can observe some bullet points:
  - The success rate increase over time
  - Payload Mass can infer the success rate of the launch mission
  - The ES-L1, GEO, HEO, and SSO shows the most high success rate against the others orbits
  - The success rate increased considerably in 2013, followed by 2017 and 2019
- The KSC LC-39A showed the best launch site rate
- Launch missions take place in the southern part of the USA, due to its proximity to the equator.
- The classification models performed similarly, with SVM showing the best accuracy amongst others

Thank you!