Winning Space Race with Data Science

Alexandre S. Maeda
28/Jan/2023

Outline

- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion

Executive Summary

- Summary of methodologies
  - Data collection
  - Data Wrangling
  - EDA with SQL and data visualization
  - Interactive map built with folium
  - Dashboard with Plotly Dash
  - ML for Predictive analysis
- Summary of all results
  - EDA results
  - Interactive analytics
  - Predictive analysis

Introduction

- Project background and context
  - We have identified that for SpaceY be able to compete with Space X our client needs to know what are the best practices, the key parameter what the probability of the first stage to land successfully. Having a successful land of the first stage means saving almost 200 milion dollars.
- Problems you want to find answers
  - The project task is to predict what are the chances of the first stage to land successfully, and how the other components like payload, booster, and launch sites can impact on that chance, using SpaceX Falcon 9 as a model.

Section 1

Methodology

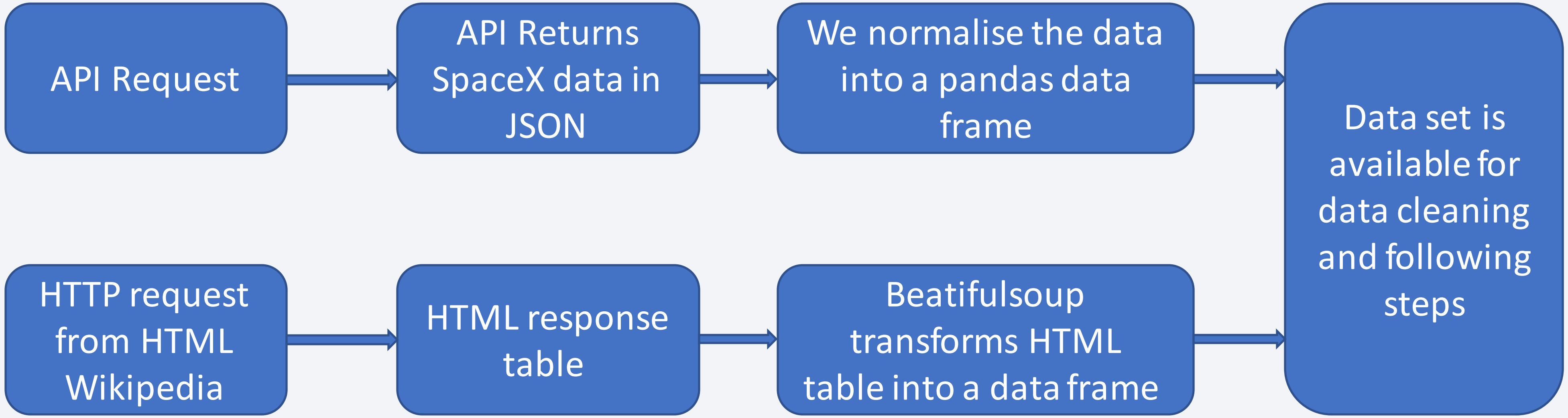
Methodology

Executive Summary

- Data collection methodology:
- The Data was collected using SpaceX API and through Wikipedia using web scraping
- Perform data wrangling
  - First we checked how many launches each site had
  - Second we group and count how many successful and failure launches went to each orbit destination.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - We standardized the data and separated into test (20% of the data) and training (80% the data) samples them we started to fit and train the data on each model to obtain the best approach on each model and the accuracy of each model and them we use a Matrix confusion to analyze the relation of False Negatives and False Positives.

Data Collection

- The Data was collected using SpaceX API (api.spacexdata.com/V4/) and Web Scraping Wikipedia website.
- The data flow is explained below with the flowchart:



Data Collection – SpaceX API

1. Get a response from API

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

[7]  response = requests.get(spacex_url)
```

2. Convert the JSON file into a Pandas Data Frame

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

3. Append some specific data to the Data Frame

```
[18] # Call getLaunchSite
     getLaunchSite(data)

[19] # Call getPayloadData
     getPayloadData(data)

[20] # Call getCoreData
     getCoreData(data)
```

```
[21] 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}
```

Data Collection – SpaceX API

4. Exclude Falcon9 from the Data Frame

```
# Hint data['BoosterVersion']!='Falcon 1'
data_falcon9 = df.loc[df['BoosterVersion'] != 'Falcon 1']
data_falcon9
```
[46]

5. Substitute the missing values for the mean

```
# Calculate the mean value of PayloadMass column
mean = data_falcon9['PayloadMass'].mean()
# Replace the np.nan values with its mean value
data_falcon9['PayloadMass'] = data_falcon9['PayloadMass'].replace(np.nan, mean)
data_falcon9.isnull().sum()
```
Testing

[64]

6. Export the Data to a CSV file

```
data_falcon9.to_csv('C:\Ale\SQL\Portfolio Projects\Final project\dataset_part_1.csv', index=False)
```

GitHub URL:

https://github.com/alexsmaeda/IBM_final_project/blob/main/1_jupyter-labs-spacex-data-collection-api%20(Data%20Collection%20API).ipynb

Data Collection - Scraping

1. Request to get the data from the HTML

```
x = requests.get(static_url)
```

2. Create a Beatifulsoup Object with the HTML response

```
page = x.content
soup = BeautifulSoup(page, "html.parser")
```

3. Find all the tables on the website, using the Beatifulsoup Object

```
html_tables = soup.find_all('table')
```

4. Getting all the column names using the "th" as a reference

```
column_names = []
temp = soup.find_all('th')
for x in range(len(temp)):
    try:
     name = extract_column_from_header(temp[x])
     if (name is not None and len(name) > 0):
        column_names.append(name)
    except:
     pass
```

Data Collection - Scraping

5. Create an empty dictionary with the column names

```
launch_dict= dict.fromkeys(column_names)

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

# Let's initial the launch_dict with each value to be an empty list
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']=[]
```

6. Parsing and appending the values to the dictionary

```
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')
```

```
launch_dict["Flight No."].append(flight_number)
```

Data Collection - Scraping

7. Convert the dictionary into Data Frame

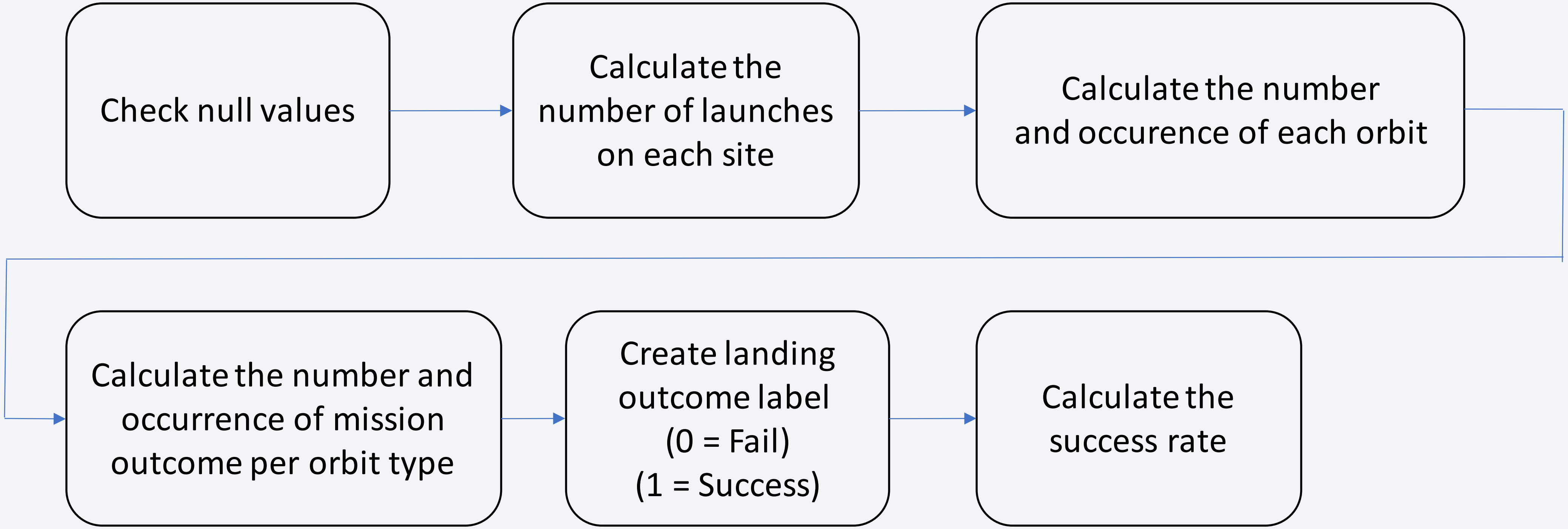
```
df = pd.DataFrame.from_dict(launch_dict)
```

6. Convert the dictionary into CSV

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

GitHub URL: https://github.com/alexsmaeda/IBM_final_project/blob/main/1_jupyter-labs-spacex-data-collection-api%20(Data%20Collection%20API).ipynb

Data Wrangling



GitHub URL: https://github.com/alexsmaeda/IBM_final_project/blob/main/3_labs-jupyter-spacex-data_wrangling_jupyterlite.jupyterlite%20(Data%20Wrangling).ipynb

EDA with Data Visualization

- Charts plotted:
  - Scatter Chart (This plot allow us to easily see how two variables are related and how it impact the outcome.)
    - Lauch Site X Payload Mass,
    - Lauch Site X Flight Number,
    - Lauch Site X Payload Mass,
    - Orbit Type X Flight Number,
    - Orbit Type X Payload Mass
  - Line Chart (This chart makes easy to visualize how the company is improving through the years.)
    - Success Rate X Year
  - Bar Char (This plot allow us to quickly visualize what is the successful rate on each orbit.)
    - Lauch Site X Payload Mass

GitHub URL: https://github.com/alexsmaeda/IBM_final_project/blob/main/5_IBM-DS0321EN-SkillsNetwork_labs_module_2_jupyter-labs-eda-dataviz.ipynb.jupyterlite%20(Exploratory%20Data%20Analysis%20for%20Data%20Visualization).ipynb

EDA with SQL

- SQL queries you performed:
  - 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. Use subquery
  - 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
  - Rank the count of successful landing_outcomes between the date 2010 and 2017 in descending order.

GitHub URL: https://github.com/alexsmaeda/IBM_final_project/blob/main/4_jupyter-labs-eda-sql-coursera_sqllite%20(Exploratory%20Analysis%20Using%20SQL).ipynb

Build an Interactive Map with Folium

- Objects used on map:
  - Circle
  - Marker
  - Lines
- We use those objects to mark and label the launch sites, distance from other public facilities and successful and failure launches on the map

GitHub URL: https://github.com/alexsmaeda/IBM_final_project/blob/main/6_IBM-DS0321EN-SkillsNetwork_labs_module_3_lab_jupyter_launch_site_location.jupyterlite%20(Launch%20Sites%20Locations%20Analysis%20with%20Folium).ipynb

Build a Dashboard with Plotly Dash

- Plots/graphs and interactions you have added to a dashboard:
  - Dropdown list (The dropdown list was added to enable the user to select which launch site analyze).
  - Pie Chart (To show the total successful launches count for all sites and if a specific site was selected, it shows the Success Vs Failed counts for the site).
  - Slider (Allowing the user to select the desired Payload Range).
  - Scatter Chart (To show the correlation between Payload and Launch success).

GitHub URL: https://github.com/alexsmaeda/IBM_final_project/blob/main/7_plotly_dash.py

Predictive Analysis (Classification)

Create a Numpy Array from the column Class so we can analyze the outcome ("1"Successful or "0"Failure).

Standardize the data.

Split the data into train and test data sets.

Fit the object to find the best parameter for the model.

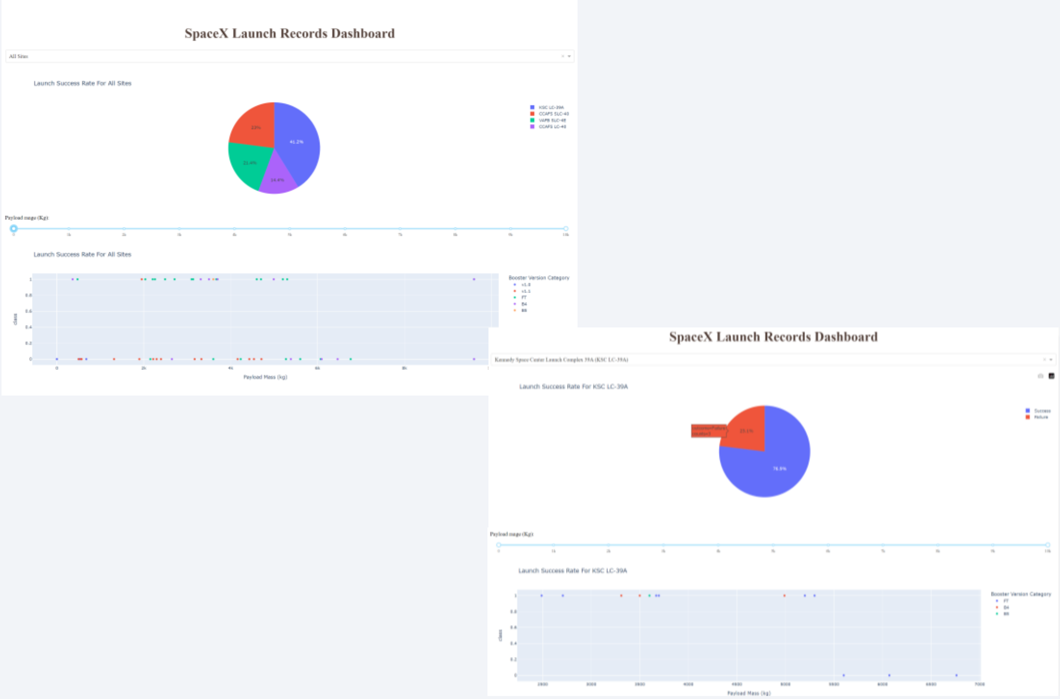
Calculate the accurancy of the model using the test data.

Plot a confusion matrix to analyse false positives and false negatives.

GitHub URL: https://github.com/alexsmaeda/IBM_final_project/blob/main/8_IBM-DS0321EN-SkillsNetwork_labs_module_4_SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite%20(Complete%20the%20Machine%20Learning%20Prediction%20lab).ipynb

Results

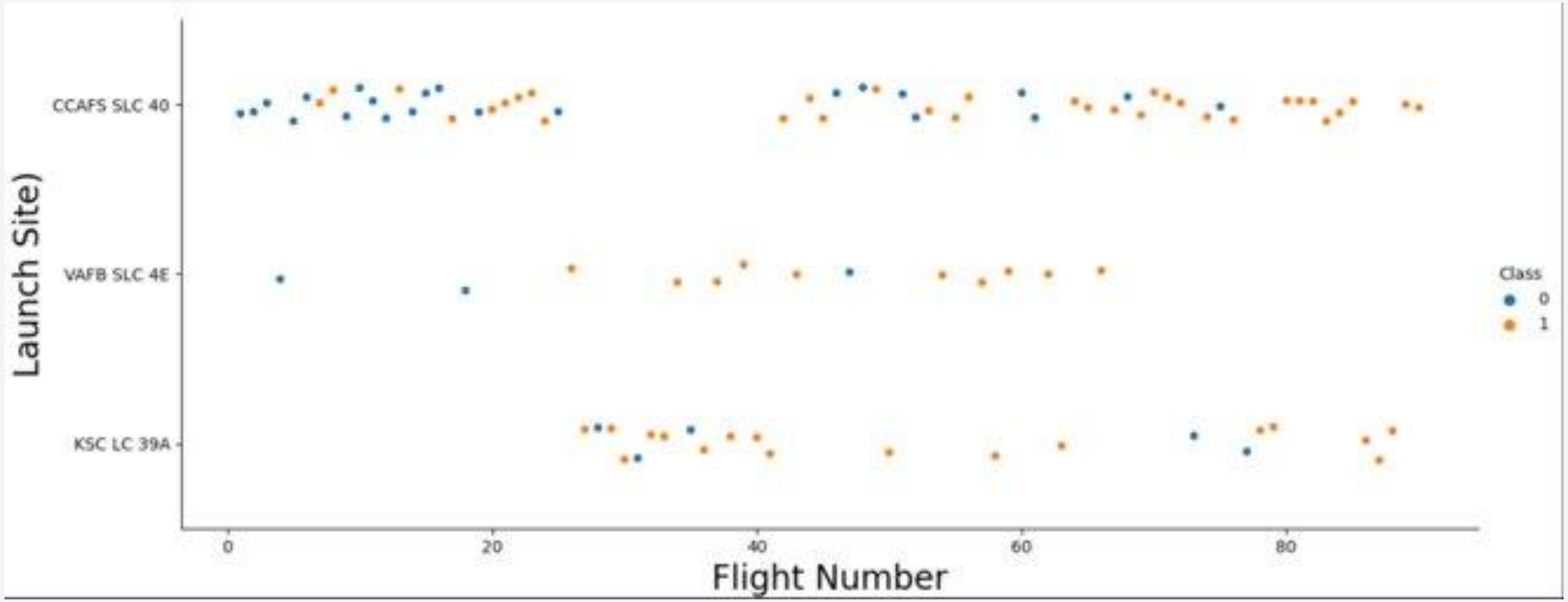
- 2k to 5k payloads perform better.
- The success rates for SpaceX launches is directly proportional to the time in years if they keep the same pace they will eventually perfect the launches
- KSC LC 39A had the most successful launches from all sites
- The SVM, KNN, Logistic Regression and Decision Tree models have the same accuracy of 0.8333333333333334, based on the experiment we found that SVM is the best fit for the project because it has the ability to handle high-dimensional data and the ability to perform well with small datasets.



Section 2

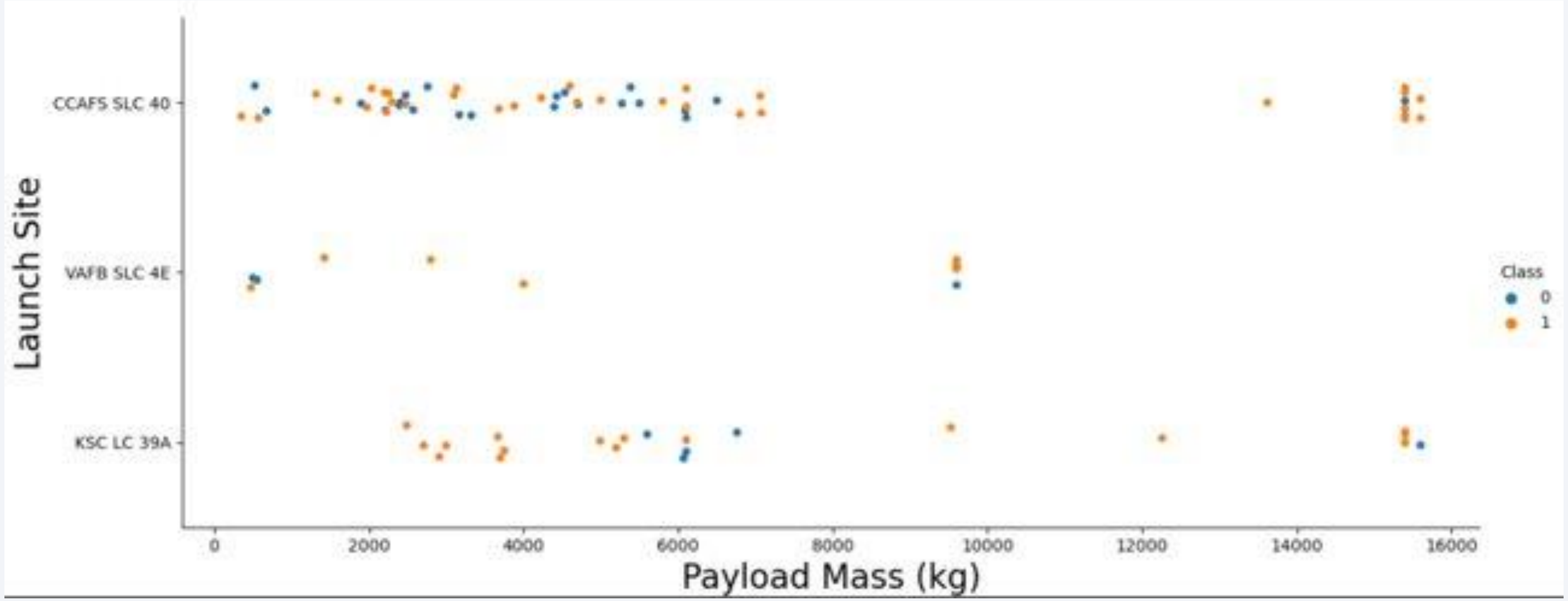
Insights drawn from EDA

Flight Number vs. Launch Site



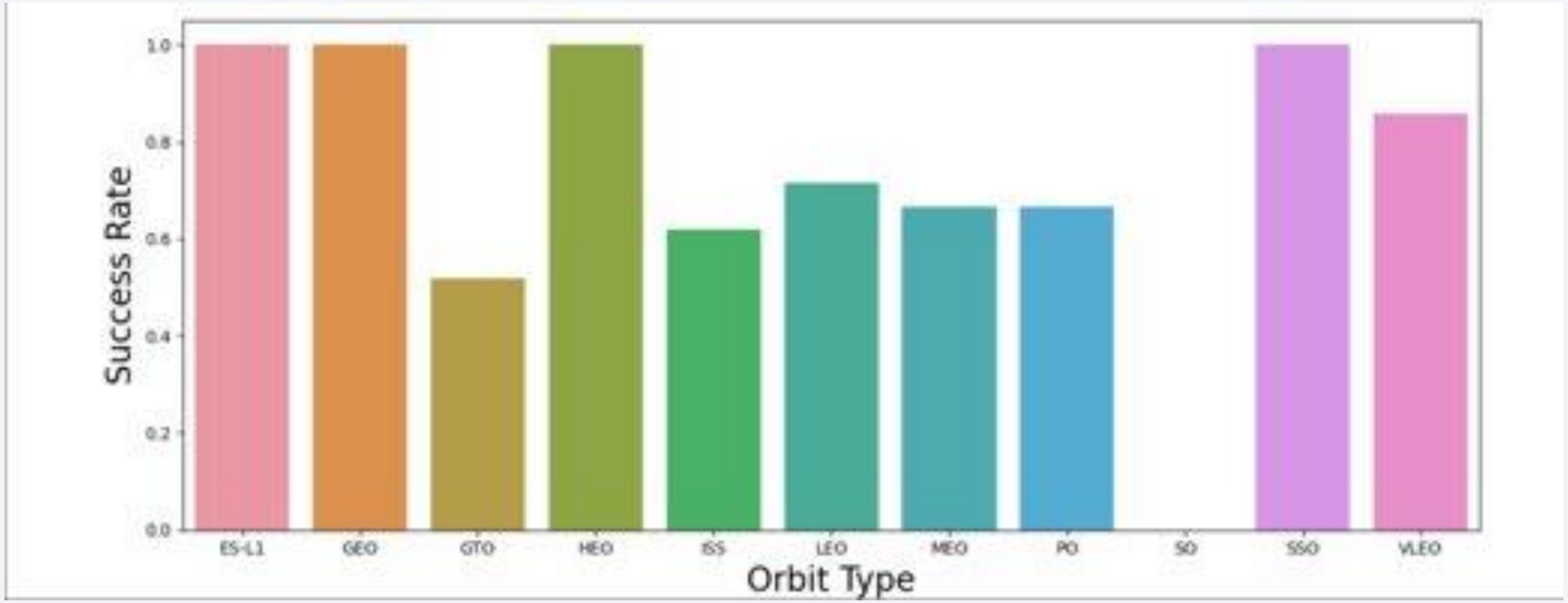
- The number of launches from the site of CCAFS SLC 40 is significantly higher than launches from other sites

Payload vs. Launch Site



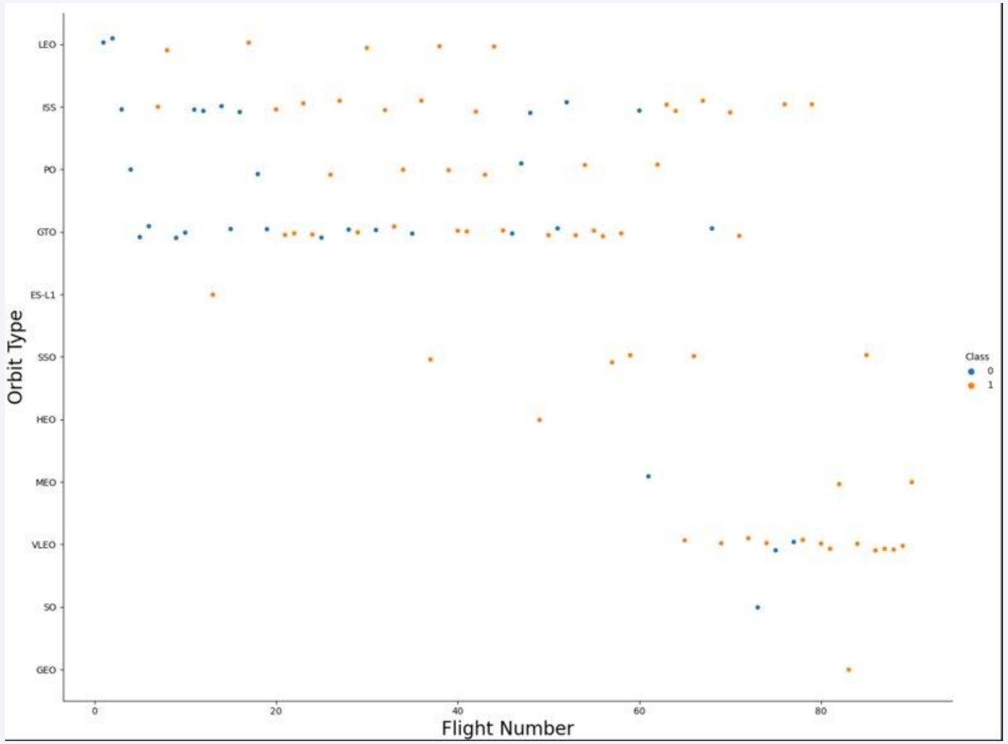
- Most of the launches were made with the payload between 1.5k and 7k.

Success Rate vs. Orbit Type



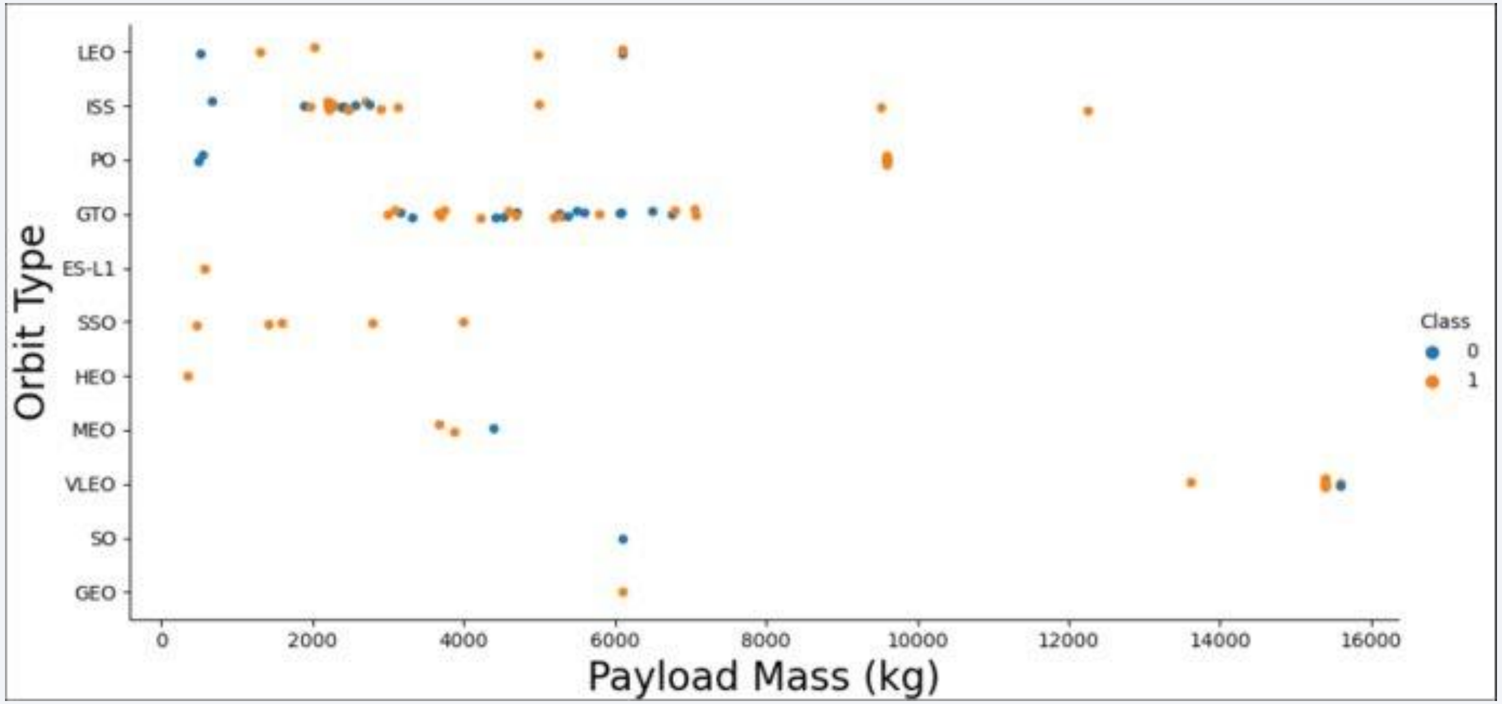
- ES-L1, GEO, HEO, SSO have 100% successful rate.

Flight Number vs. Orbit Type



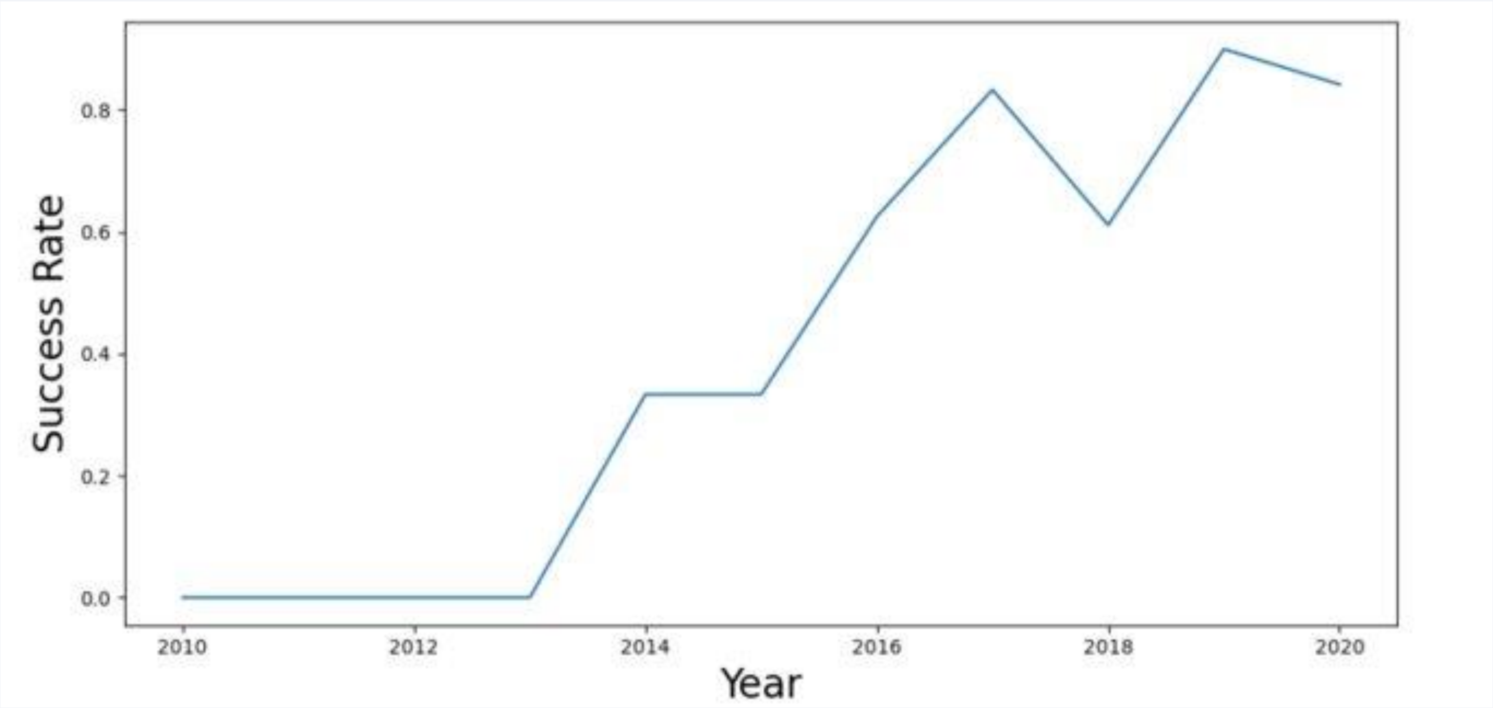
- Flights number greater than 20 have a higher successful rate in all orbits.

Payload vs. Orbit Type



- ISS present a strong correlation between the payload of 2000 and 4000, and GTO as well between the payload of 2500 and 7000.

Launch Success Yearly Trend



- From 2013 to 2017 the Success launch rate has increased, after that it slightly decreased and started to increase again, but with a slower ration than the early years of SpaceX.

All Launch Site Names

```
In [16]: sql SELECT DISTINCT LAUNCH_SITE FROM SPACEXTBL ORDER BY 1;

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

Out[16]:

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

- We have 4 distinct launch sites

Launch Site Names Begin with 'CCA'

```
In [19]: sql SELECT * FROM SPACEXTBL WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5;

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

Out[19]:

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

- Five of the records from the launch site beginning with 'CCA' had a success mission outcome.

Total Payload Mass

```
In [48]: sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE CUSTOMER LIKE 'NASA (CRS)';

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

Out[48]:

SUM(PAYLOAD_MASS__KG_)
45596

- The total payload mass is 45.596 kg

Average Payload Mass by F9 v1.1

In [30]:
```
sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE BOOSTER_VERSION LIKE 'F9 v1.1%';
```

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

Out[30]:

AVG(PAYLOAD_MASS__KG_)
2534.666666666665

- The average payload mass carried by booster version F9 v1.1 is 2534.667 kg

First Successful Ground Landing Date

```
In [90]: sql SELECT * FROM SPACEXTBL WHERE LANDING_OUTCOME = "Success (ground pad)";
```

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

Out[90]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	LANDING_OUTCOME
22-12-2015	01:29:00	F9 FT B1019	CCAFS LC-40	OG2 Mission 2 11 Orbcomm-OG2 satellites	2034	LEO	Orbcomm	Success	Success (ground pad)
18-07-2016	04:45:00	F9 FT B1025.1	CCAFS LC-40	SpaceX CRS-9	2257	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
19-02-2017	14:39:00	F9 FT B1031.1	KSC LC-39A	SpaceX CRS-10	2490	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
01-05-2017	11:15:00	F9 FT B1032.1	KSC LC-39A	NROL-76	5300	LEO	NRO	Success	Success (ground pad)
03-06-2017	21:07:00	F9 FT B1035.1	KSC LC-39A	SpaceX CRS-11	2708	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
14-08-2017	16:31:00	F9 B4 B1039.1	KSC LC-39A	SpaceX CRS-12	3310	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
07-09-2017	14:00:00	F9 B4 B1040.1	KSC LC-39A	Boeing X-37B OTV-5	4990	LEO	U.S. Air Force	Success	Success (ground pad)
15-12-2017	15:36:00	F9 FT B1035.2	CCAFS SLC-40	SpaceX CRS-13	2205	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
08-01-2018	01:00:00	F9 B4 B1043.1	CCAFS SLC-40	Zuma	5000	LEO	Northrop Grumman	Success (payload status unclear)	Success (ground pad)

```
In [108…: sql SELECT * FROM SPACEXTBL WHERE LANDING_OUTCOME = "Success (ground pad)" ORDER BY DATE DESC LIMIT 1;
```

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

Out[108…:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	LANDING_OUTCOME
22-12-2015	01:29:00	F9 FT B1019	CCAFS LC-40	OG2 Mission 2 11 Orbcomm-OG2 satellites	2034	LEO	Orbcomm	Success	Success (ground pad)

- The date of the first successful landing outcome on ground pad was 22/12/2015

Successful Drone Ship Landing with Payload between 4000 and 6000

In [112…

```
sql SELECT BOOSTER_VERSION FROM SPACEXTBL WHERE LANDING_OUTCOME = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000;
```

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

Out[112…

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

- The names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000 are:
  - F9 FTB1022
  - F9 FTB1026
  - F9 FTB1021.2
  - F9 FtB1031.2

Total Number of Successful and Failure Mission Outcomes

In [114…

```
sql SELECT MISSION_OUTCOME, COUNT(*) FROM SPACEXTBL GROUP BY MISSION_OUTCOME
```

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

Out[114…

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

- The total number of successful missions outcomes are 100 and failure mission outcomes is 1

Boosters Carried Maximum Payload

In [116…

```
%%sql
SELECT BOOSTER_VERSION, PAYLOAD_MASS__KG_
FROM SPACEXTBL
WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL);
```

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

Out[116…

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

- The names of the booster which have carried the maximum payload mass are:
  - 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

In [119…

```
%%sql
SELECT substr(Date, 4, 2) AS month, LANDING_OUTCOME, BOOSTER_VERSION, LAUNCH_SITE
FROM SPACEXTBL
WHERE LANDING_OUTCOME = 'Failure (drone ship)' AND substr(Date,7,4)='2015';
```

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

Out[119…

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

- The failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015 are:
  - F9 V1.1 B1012 (booster version) – CCAFS LC-40 (launch site)
  - F9 V1.1 B1015 (booster version) – CCAFS LC-40 (launch site)

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

```
[40]: %%sql
SELECT ROW_NUMBER() OVER (ORDER BY Date(Date,"%d-%m-%Y") DESC) AS No,
    Date, LANDING_OUTCOME AS "LANDING_OUTCOME (2010 TO 2017)"
FROM SPACEXTBL
WHERE LANDING_OUTCOME LIKE'Success%' AND
substr(Date,7,4) BETWEEN '2010' AND '2017'

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

[40]:

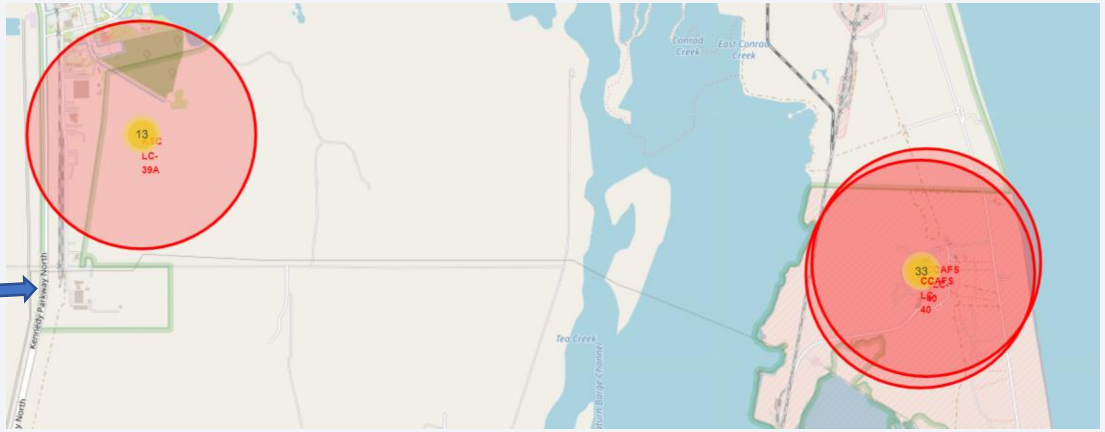
No	Date	LANDING_OUTCOME (2010 TO 2017)
1	22-12-2015	Success (ground pad)
2	08-04-2016	Success (drone ship)
3	06-05-2016	Success (drone ship)
4	27-05-2016	Success (drone ship)
5	18-07-2016	Success (ground pad)
6	14-08-2016	Success (drone ship)
7	14-01-2017	Success (drone ship)
8	19-02-2017	Success (ground pad)
9	30-03-2017	Success (drone ship)
10	01-05-2017	Success (ground pad)
11	03-06-2017	Success (ground pad)
12	23-06-2017	Success (drone ship)
13	25-06-2017	Success (drone ship)
14	14-08-2017	Success (ground pad)
15	24-08-2017	Success (drone ship)
16	07-09-2017	Success (ground pad)
17	09-10-2017	Success (drone ship)
18	11-10-2017	Success (drone ship)
19	30-10-2017	Success (drone ship)
20	15-12-2017	Success (ground pad)

- Here is the ranking count of successful landing_outcomes between the date 2010 and 2017 in descending order.

Section 3

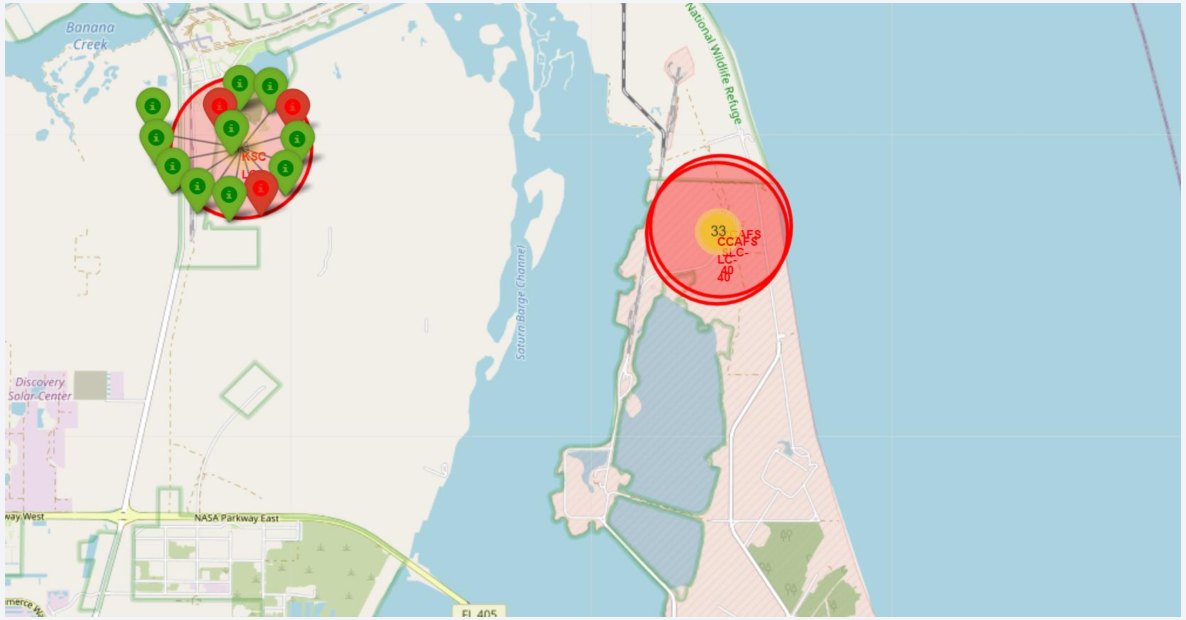
Launch Sites Proximities Analysis

Location markers of the launch sites



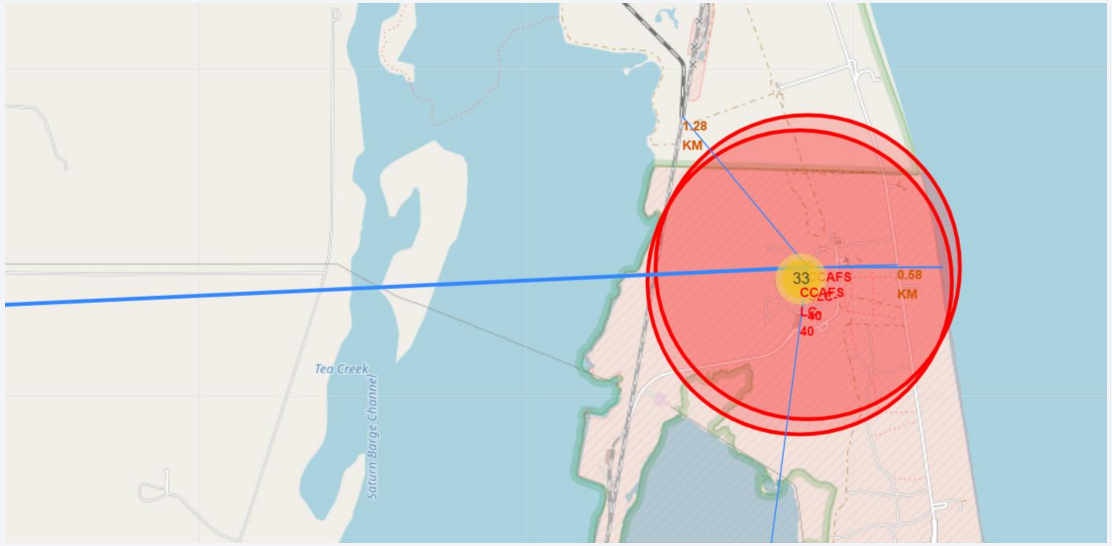
- We can see all the launch site locations and the respective numbers of launches for each site

Launch site outcomes



- From the map we can quickly see how many successful and failures launches each launch site had.

Launch site proximities evaluation

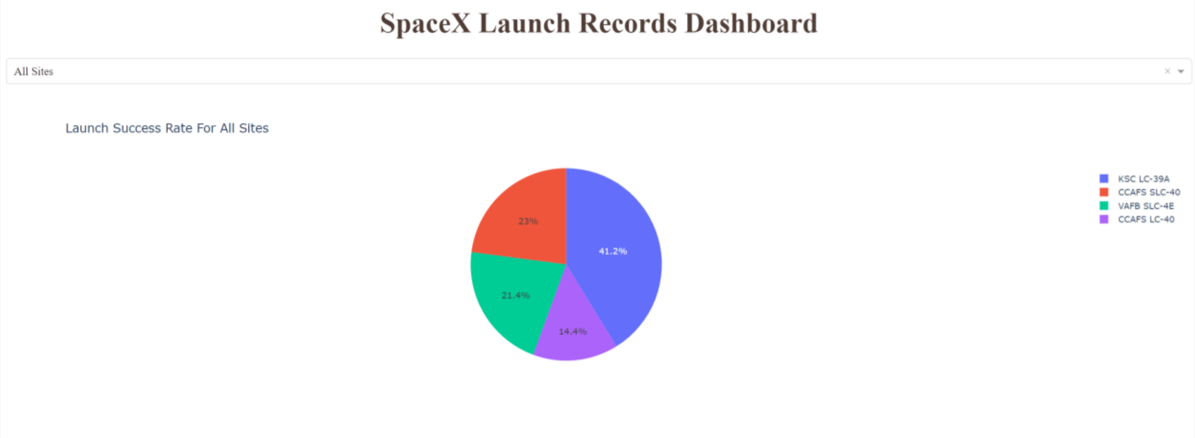


- If we zoom in we can see how far a selected launch site is to its proximities such as railway, highway and coastline.

Section 4

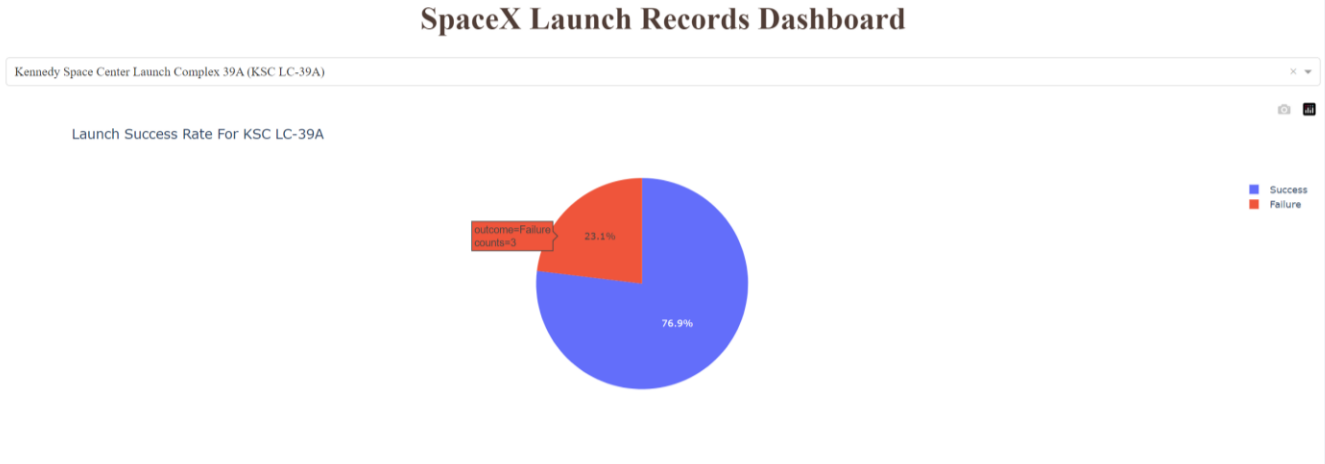
Build a Dashboard with Plotly Dash

Launch success count for all sites



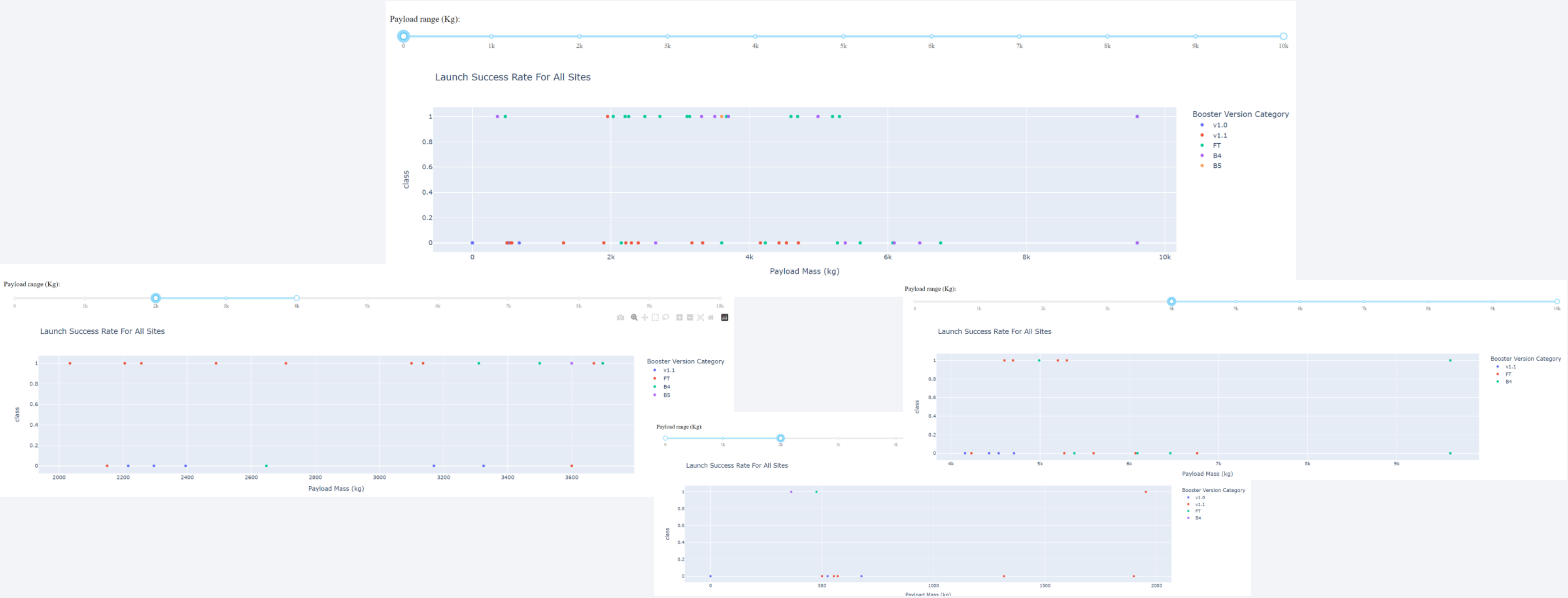
- In this chart we can see that the KSC LC 39-A launch site had the highest success rate of 41.2%

Highest launch success ratio evaluation



- Here we can see that KSC LC 39-A had a 76.9 success rate and 23.1 failure rate.

Payload vs Launch Outcome

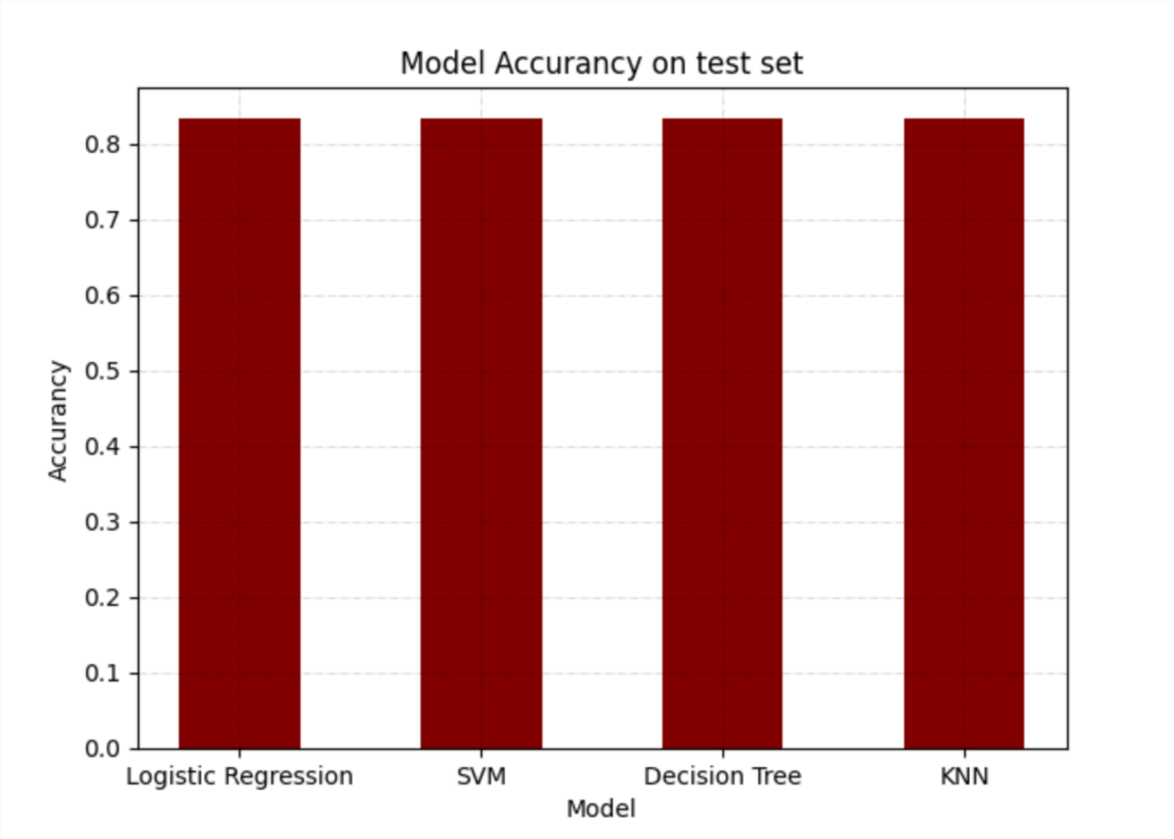


- We can see that payloads from 2000 to 4000kg had a higher success rate

Section 5

Predictive Analysis (Classification)

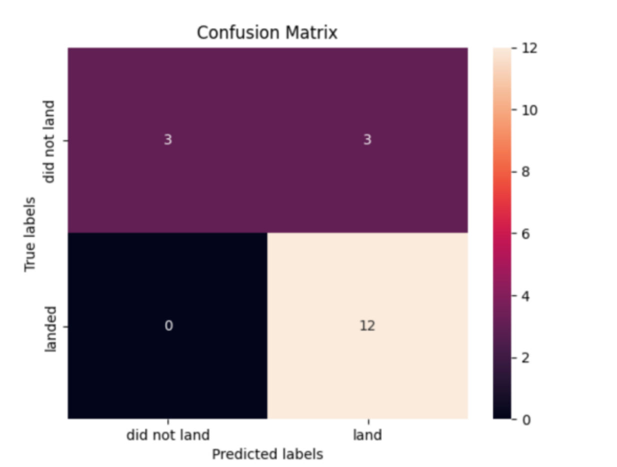
Classification Accuracy



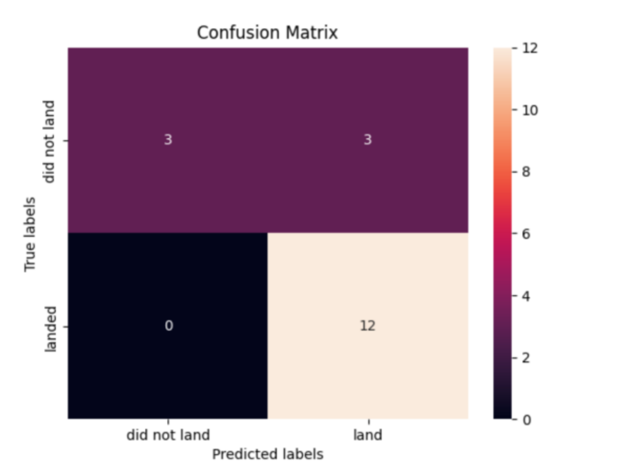
- All models deliver the same performance of 0.833 on test set.

Confusion Matrix

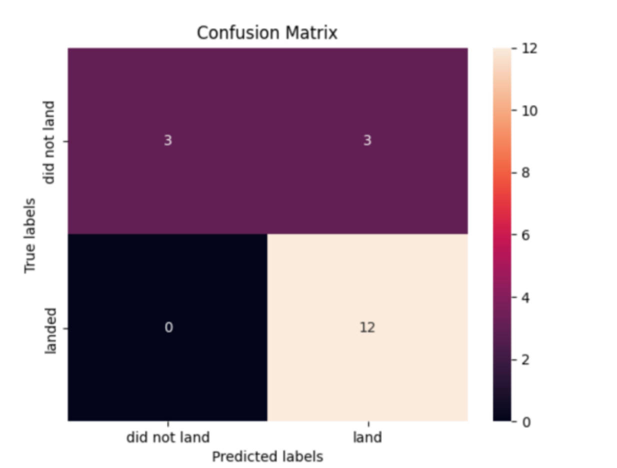
- Logistic regression



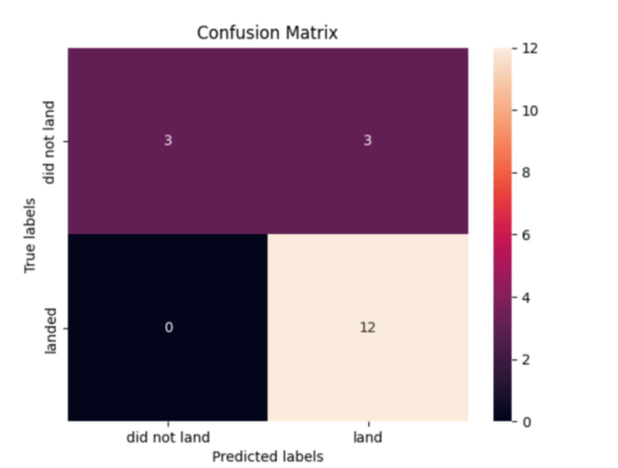
- SVM



- Decision Tree



- KNN



- All models present the same confusion matrix of:
  - 3 True Positives
  - 3 False Positives
  - 0 False Negatives
  - 12 True Negatives

Conclusions

- The SVM is the best model in terms of prediction accuracy for this data set, because of its ability to handle high-dimensional data and the ability to perform well with small datasets.
- Payloads between 2000-4000kg perform better than other payloads.
- KSC LC39A had the most successful launches from all site, and we should analyze it deeper.
- Launches to the orbits ES-L1, GEO, HEO and SSO had the most successful rate and also should be analyzed deeper.
- SpaceX had a very steep success ratio curve from 2013-2017, but it seems to be slowing down after 2017.

Appendix

```
In [233…
%%sql
SELECT LANDING_OUTCOME AS "LANDING_OUTCOME (2010 TO 2017)", COUNT(LANDING_OUTCOME) AS 'SUCCESS COUNT'
FROM SPACEXTBL
WHERE LANDING_OUTCOME LIKE'Success%' AND
substr(Date,7,4) BETWEEN '2010' AND '2017'
GROUP BY LANDING_OUTCOME;

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

Out[233…

LANDING_OUTCOME (2010 TO 2017)	SUCCESS COUNT
Success (drone ship)	12
Success (ground pad)	8

- We can see that from 2010 to 2017, the drone ship landing success rate outcome has been larger than the ground pad landing success rate outcome.

Thank you!