IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Ahmed Gharib
1/1/2023

Outline

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

Executive Summary

Summary of Methodologies:

To predict if the first stage would be a success, the following methodologies were used.

- Data Collection
- Data Wrangling
- Exploratory Data Analytics
- Interactive Visual Analytics
- Predictive Analysis

Summary of Results:

The results were in various forms detailed below.

- EDA Results
- Interactive dashboard
- Predictive analysis of classification models

Introduction

- The commercial space age is growing, SpaceX Falcon 9 rockets are leading in the industry currently as they are considerably cheaper than any alternatives provided by competitors, this is because they are reusable. We can use the first stage of the rocket again for another flight, this brings down the costs.
- If we can predict the success rate of the first stage, we can predict the costs. Using this we can determine if an alternative operative should bid for the launch.
- The main purpose of the project is to determine if the first stage will be successful.

Section 1

Methodology

Methodology

I've have performed all the methods stated below

1. Data Collection
   - Collecting Data from SpaceX API
2. Data Wrangling
   - Classifying the status of the landings (success or failure)
3. EDA
   - Using python libraries (Pandas and Matplotlib) to display and determine patterns between variables
4. Predictive Analysis using Classification Models

Data Collection – SpaceX API

GitHub URL

https://github.com/ahmed-gharib89/IBM-Applied-Data-Science-Capstone/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

1
```
spacex_url="https://api.spacexdata.com/v4/launches/past"

response = requests.get(spacex_url)
```

3
```
# Use json_normalize meethod to convert the json result into a dataframe
data = pd.json_normalize(response.json())
```

5
```
# Hint data['BoosterVersion']!='Falcon 1'
data_falcon9 = df.loc[df["BoosterVersion"] == 'Falcon 9']

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

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

2
```
#Global variables
BoosterVersion = []
PayloadMass = []
Orbit = []
LaunchSite = []
Outcome = []
Flights = []
GridFins = []
Reused = []
Legs = []
LandingPad = []
Block = []
ReusedCount = []
Serial = []
Longitude = []
Latitude = []
```

4
```
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}
```

6
```
# Calculate the mean value of PayloadMass column
data_falcon9 = data_falcon9.fillna(value={'PayloadMass': data_falcon9['PayloadMass'].mean()})
```

1. Request (Space X APIs)
2. .JSON file + Lists(Launch Site, Booster Version, Payload Data)
3. Json_normalize to DataFrame data from JSON
4. Cast dictionary to a DataFrame
5. Filter data to only include Falcon 9 launches
6. Imputate missing PayloadMass values with mean

Data Collection - Scraping

GitHub URL

https://github.com/ahmed-gharib89/IBM-Applied-Data-Science-Capstone/blob/main/jupyter-labs-webscraping.ipynb

1
```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"
page = requests.get(static_url).text
```

2
```
# Use BeautifulSoup() to create a BeautifulSoup object
soup = BeautifulSoup(page, "html.parser")
html_tables = soup.find_all("table")
```

3
```
column_names = []

# Apply find_all() function with `th` element on first_launch_table
# Iterate each th element and apply the provided extract_column_from_header() to get a co
# Append the Non-empty column name (`if name is not None and len(name) > 0`) into a list
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)
```

4
```
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']=[]
```

5
```
df=pd.DataFrame(launch_dict)
```

1. Request Wikipedia html
2. Create BeautifulSoup Object
3. Find launch info html table
4. Iterate through table cells to extract data to dictionary
5. Create dictionary into a Pandas DataFrame

Data Wrangling

GitHub URL

https://github.com/ahmed-gharib89/IBM-Applied-Data-Science-Capstone/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

Create a training label with landing outcomes where successful = 1 & failure = 0.

Outcome column has two components: 'Mission Outcome' 'Landing Location'

New training label column 'class' with a value of 1 if 'Mission Outcome' is True and 0 otherwise. Value Mapping:

True ASDS, True RTLS, & True Ocean – set to -> 1

None None, False ASDS, None ASDS, False Ocean, False RTLS – set to -> 0

EDA with Data Visualization

GitHub URL
https://github.com/ahmed-gharib89/IBM-Applied-Data-Science-Capstone/blob/main/jupyter-labs-eda-dataviz.ipynb

- Exploratory Data Analysis performed on variables Flight Number, Payload Mass, Launch Site, Orbit, Class and Year.
- Plots Used:
- Flight Number vs. Payload Mass, Flight Number vs. Launch Site, Payload Mass vs. Launch Site, Orbit vs. Success Rate, Flight Number vs. Orbit, Payload vs Orbit, and Success Yearly Trend
- Scatter plots, line charts, and bar plots were used to compare relationships between variables to
- decide if a relationship exists so that they could be used in training the machine learning model

EDA with SQL

GitHub URL
https://github.com/ahmed-gharib89/IBM-Applied-Data-Science-Capstone/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

The following were the sql queries performed, this is how we brought about our results

- - 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 succesful landing outcome in ground pad was acheived.
- - 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 a 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 04-06-2010 and 20-03-2017 in descending order

Build an Interactive Map with Folium

GitHub URL
https://github.com/ahmed-gharib89/IBM-Applied-Data-Science-Capstone/blob/main/lab_jupyter_launch_site_location.ipynb

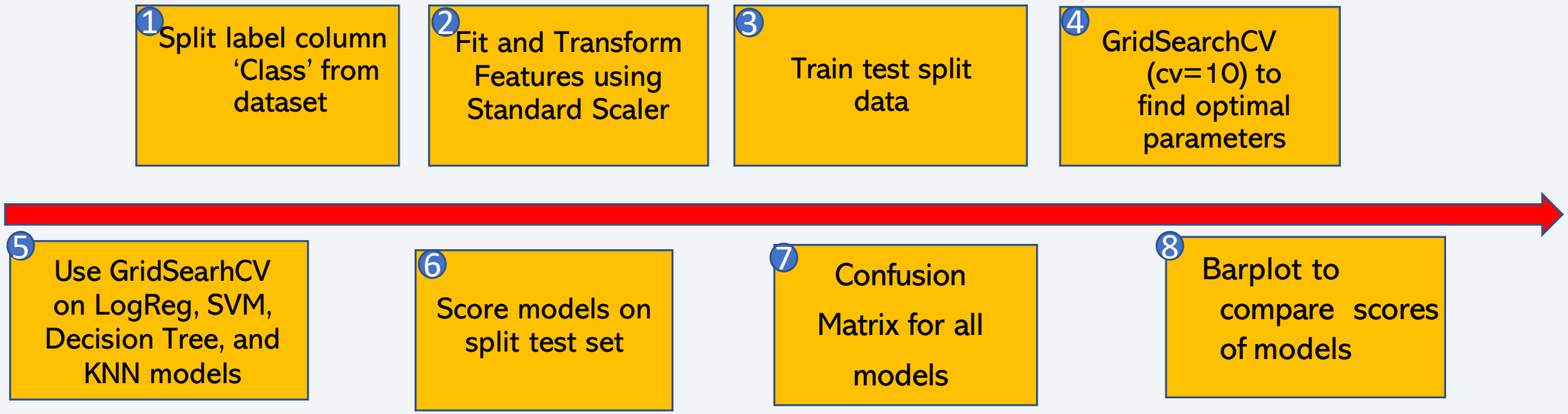
The following tasks and subtasks were dictated to us with in Data Visualization With Folium Notebook.

1. Mark all launch sites on a map
   - Initialise the map using a Folium Map object
   - Add a folium.Circle and folium.Marker for each launch site on the launch map
2. Mark the success/failed launches for each site on a map
   - As many launches have the same coordinates, it makes sense to cluster them together.
   - Before clustering them, assign a marker colour of successful (class = 1) as green, and failed (class = 0) as red.
   - To put the launches into clusters, for each launch, add a folium.Marker to the MarkerCluster() object.
   - Create an icon as a text label, assigning the icon_color as the marker_colour determined previously.
3. Calculate the distances between a launch site to its proximities
   - To explore the proximities of launch sites, calculations of distances between points can be made using the Lat and Long values.
   - After marking a point using the Lat and Long values, create a folium.Marker object to show the distance.
   - To display the distance line between two points, draw a folium.PolyLine and add this to the map.

We completed the task to get our results.

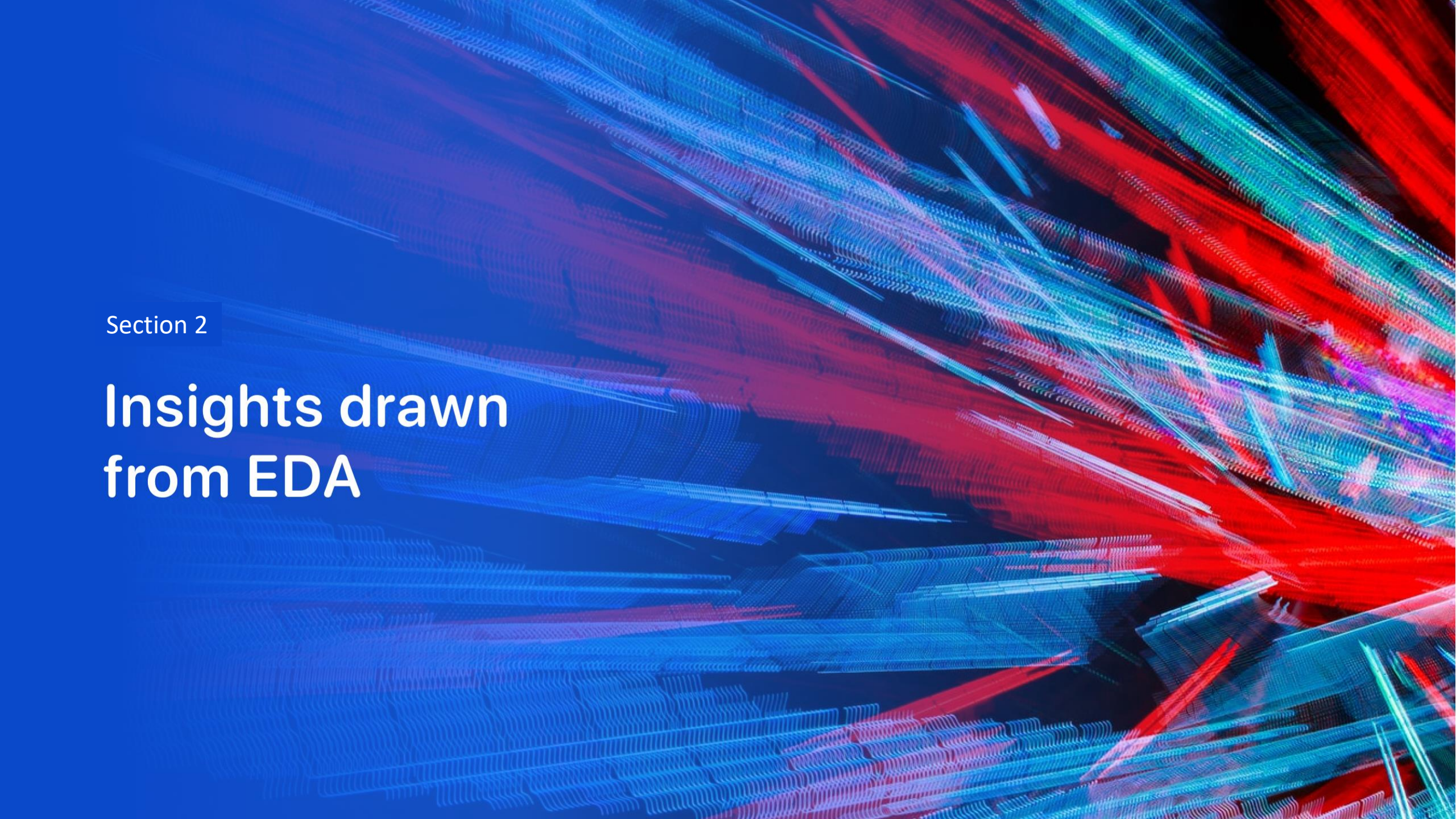
Predictive Analysis (Classification)

GitHub URL
https://github.com/ahmed-gharib89/IBM-Applied-Data-Science-Capstone/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

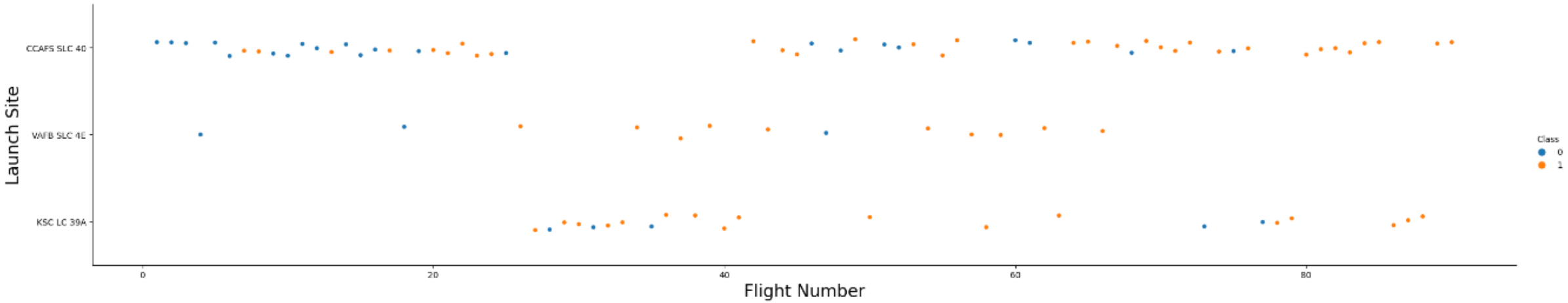


Results

- The Logistic Regression, Support Vector Machine and K nearest neighbors models are the best in terms of accuracy.
- Less weighted payloads have more success rate than heaver payloads.
- KSC LC 39A Launch Site is the best of all sites
- Orbits GEO, HEO, SSO, ES L1 have the best success rate.


Section 2
Insights drawn from EDA

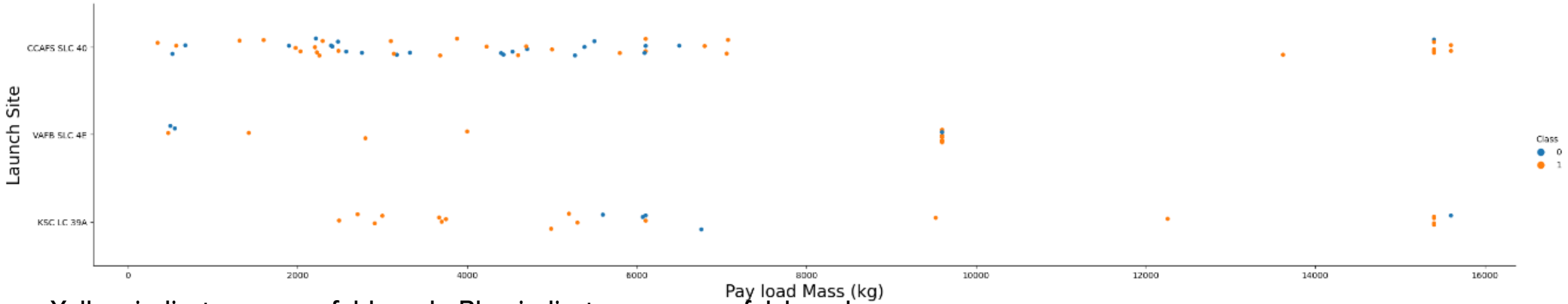
Flight Number vs. Launch Site



Yellow indicates successful launch; Blue indicates unsuccessful launch.

Graphic suggests an increase in success rate as more Flight were launched. Likely a big breakthrough around flight 20 which significantly increased success rate. Above a flight number of around 30, there are significantly more successful landings. CCAFS appears to be the main launch site as it has the most volume.

Payload vs. Launch Site



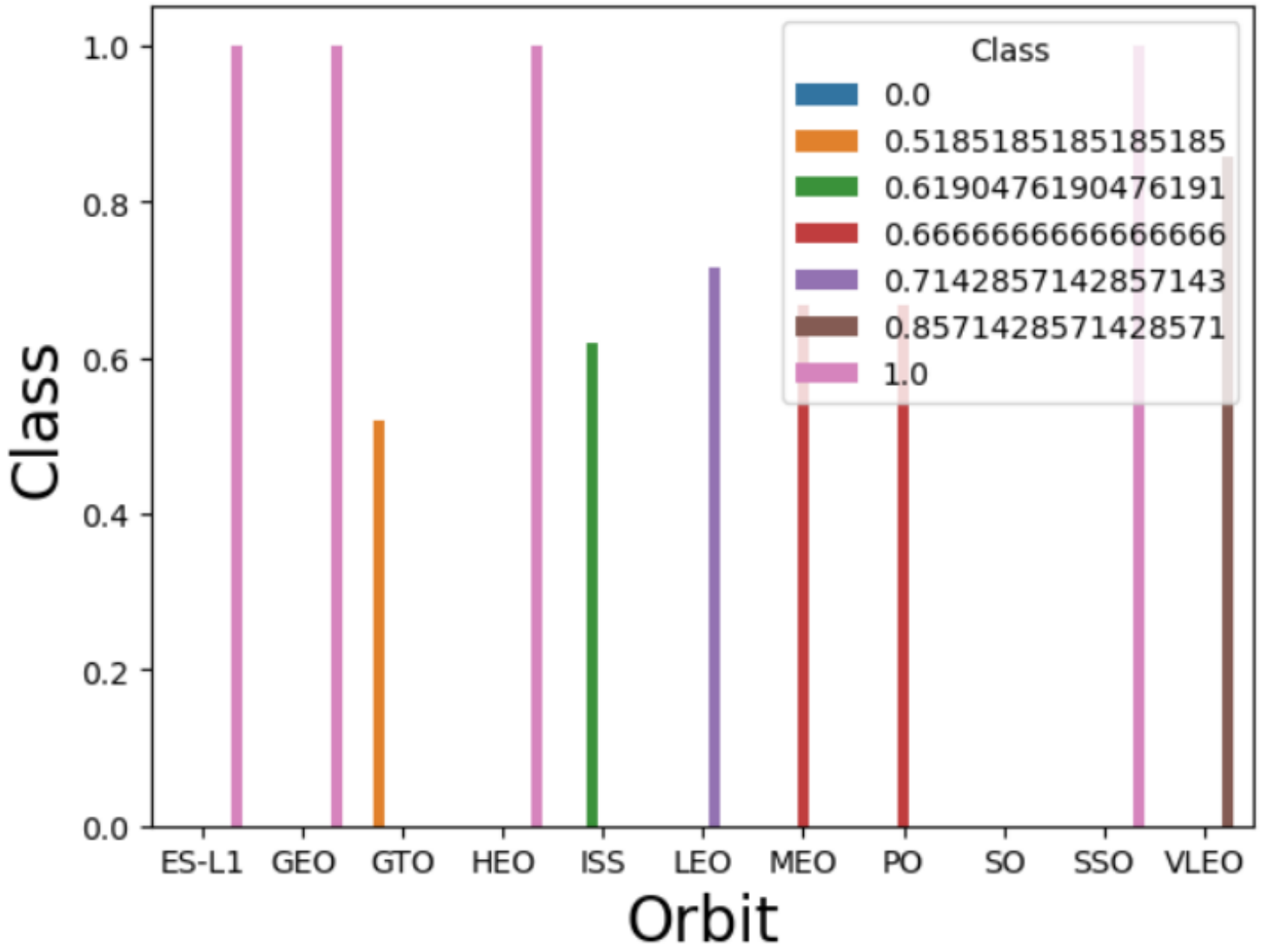
Yellow indicates successful launch; Blue indicates unsuccessful launch.

Payload mass appears to fall mostly between 0-7000 kg. Different launch sites also seem to use different payload mass. Most of the lighter payloads have been launched from CCAFS SLC 40. There is no clear correlation between payload mass and success rate for a given launch site.

Success Rate vs. Orbit Type

The ES-L1, GEO, HEO and SSO orbits have the highest successful rates.

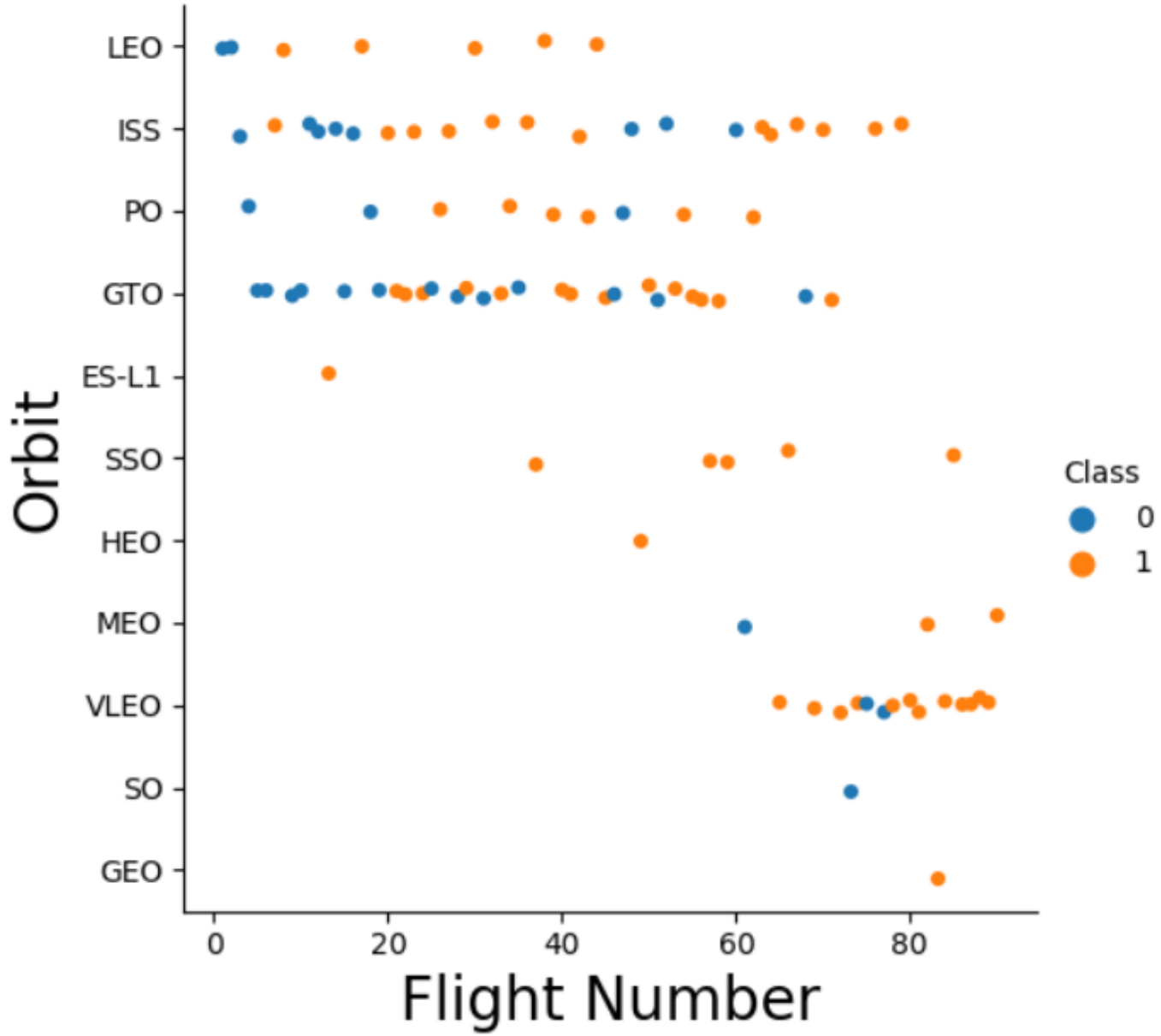
The SO has the lowest success rate.



Flight Number vs. Orbit Type

SpaceX started with LEO orbits which saw moderate success LEO and returned to VLEO in recent launches

As seen previously as the Flight Number increases, the success rate increases.



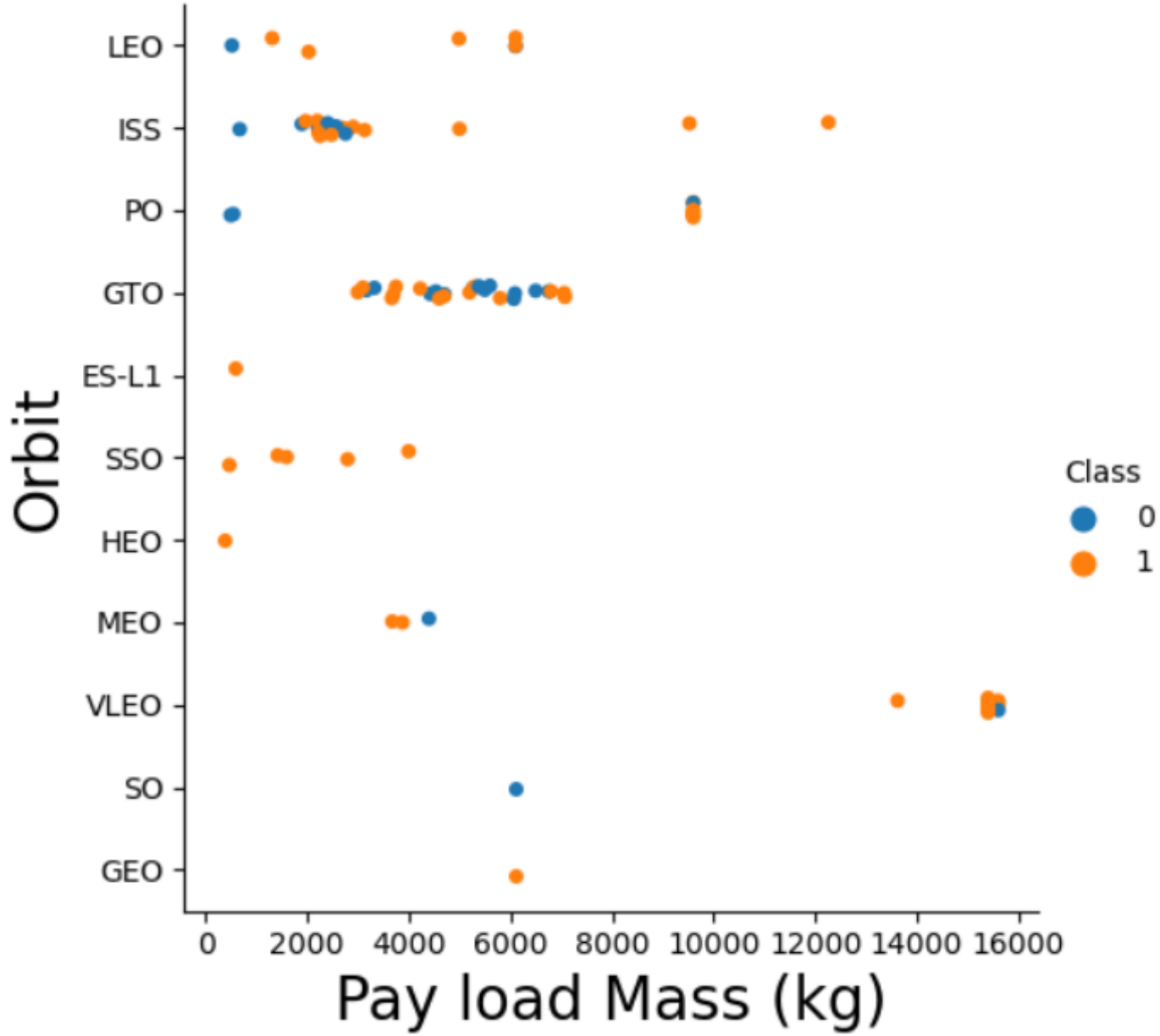
Yellow indicates successful launch; Blue indicates unsuccessful launch.

Payload vs. Orbit Type

The relationship between GTO Orbit and Pay Load Mass seems unclear.

This plot is not particularly useful as not enough launches have occurred on different orbits.

Certain Orbits only operate within certain Pay Load Mass intervals, such as VLEO (Very Low Earth Orbit) only operates under the highest mass.

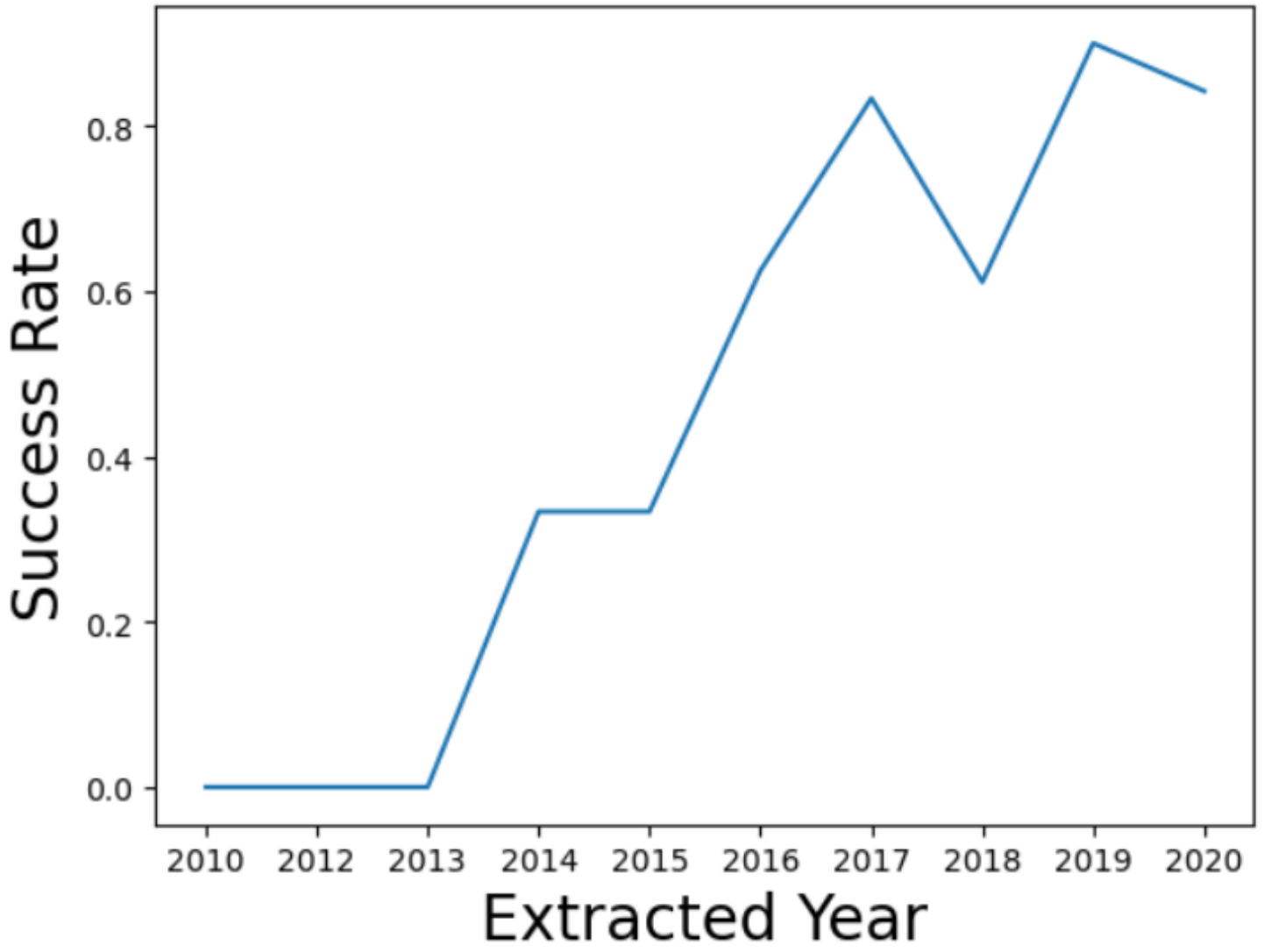


Yellow indicates successful launch; Blue indicates unsuccessful launch.

Launch Success Yearly Trend

Success generally increases over time since 2013 with a slight decrease in 2018

Between 2010 and 2013, all landings were unsuccessful (as the success rate is 0).



All Launch Site Names

```
%sql select DISTINCT(launch_site) from SPACEX
```

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

- The task is for all Launch site names
- There is likely only 3 unique Launch sites, as CCAFS LC-40 and CCAFS SLC-40 are probably one and the same, this aberration may have occurred due to human error.

Launch Site Names Begin with 'CCA'

```
%sql select * from SPACEX\
     where launch_site like 'CCA%' 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	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	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	00:35:00	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

First five entries in database with Launch Site name beginning with CCA.

Total Payload Mass

```
%sql select SUM(payload_mass__kg_) from SPACEX\
    where customer = 'NASA (CRS)'
```

1
45596

This query sums the total payload mass in kg where NASA was the customer.

Average Payload Mass by F9 v1.1

```
%sql select AVG(payload_mass__kg_) from SPACEX\
    where booster_version = 'F9 v1.1'
```

1
2928

This query calculates the average payload mass or launches which used booster version F9 v1.1

First Successful Ground Landing Date

```
%sql select MIN(DATE) from SPACEX\
    where landing__outcome = 'Success (ground pad)'
```

1
2015-12-22

This query returns the first successful ground pad landing date.

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

This query returns the four booster versions that had successful drone ship landings and a payload mass between 4000 and 6000 noninclusively.

Total Number of Successful and Failure Mission Outcomes

```
%sql select mission_outcome,COUNT(mission_outcome) AS TOTAL_NUMBER from SPACEX\
     group by mission_outcome
```

mission_outcome	total_number
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

This query returns a count of each mission outcome.

Boosters Carried Maximum Payload

```
%sql select booster_version from SPACEX\
     where payload_mass__kg_ in (select MAX(payload_mass__kg_) from SPACEX)
```

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

This query returns the booster versions that carried the highest payload mass of 15600 kg.

2015 Launch Records

```
%sql select booster_version, launch_site from SPACEX\
     where landing__outcome = 'Failure (drone ship)' and YEAR(DATE) = '2015'
```

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

The WHERE keyword is used to filter the results for only failed landing outcomes, and only for the year of 2015.

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

```
%sql select landing__outcome,COUNT(landing__outcome) AS TOTAL_NUMBER from SPACEX\
     where date between '2010-06-04' and '2017-03-20'\
     group by landing__outcome\
     order by COUNT(landing__outcome) DESC
```

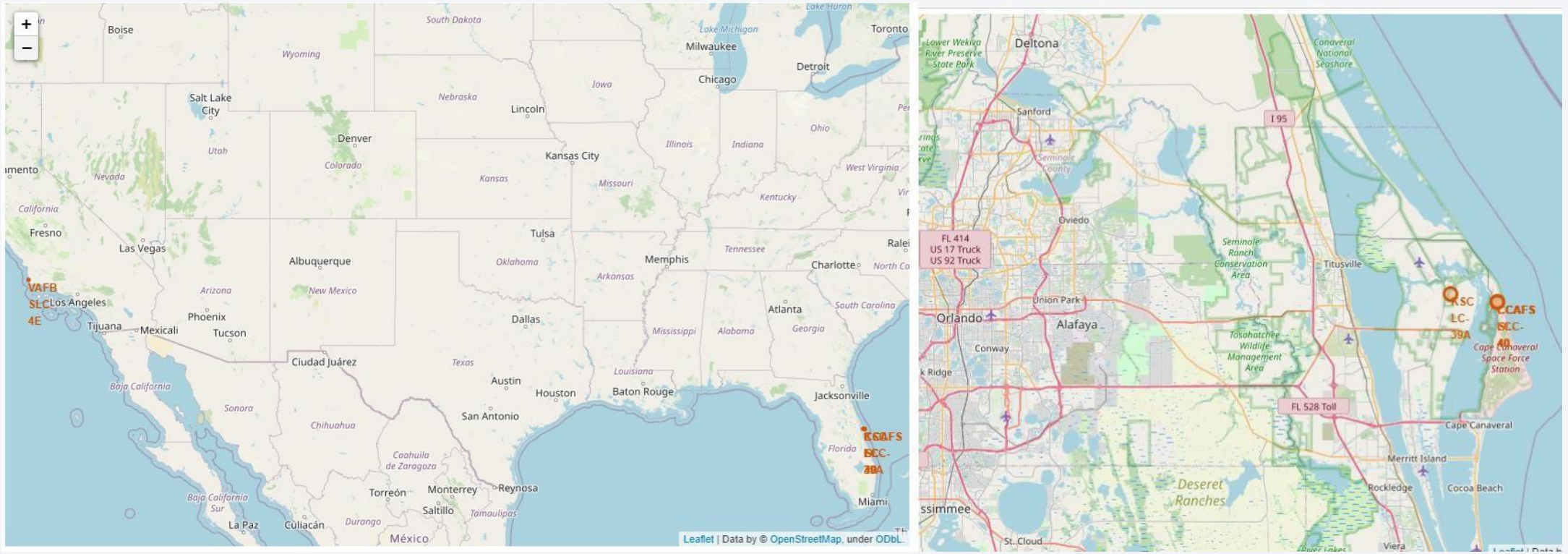
landing__outcome	total_number
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

This query returns every possible landing outcome within the specified dates.

Section 3

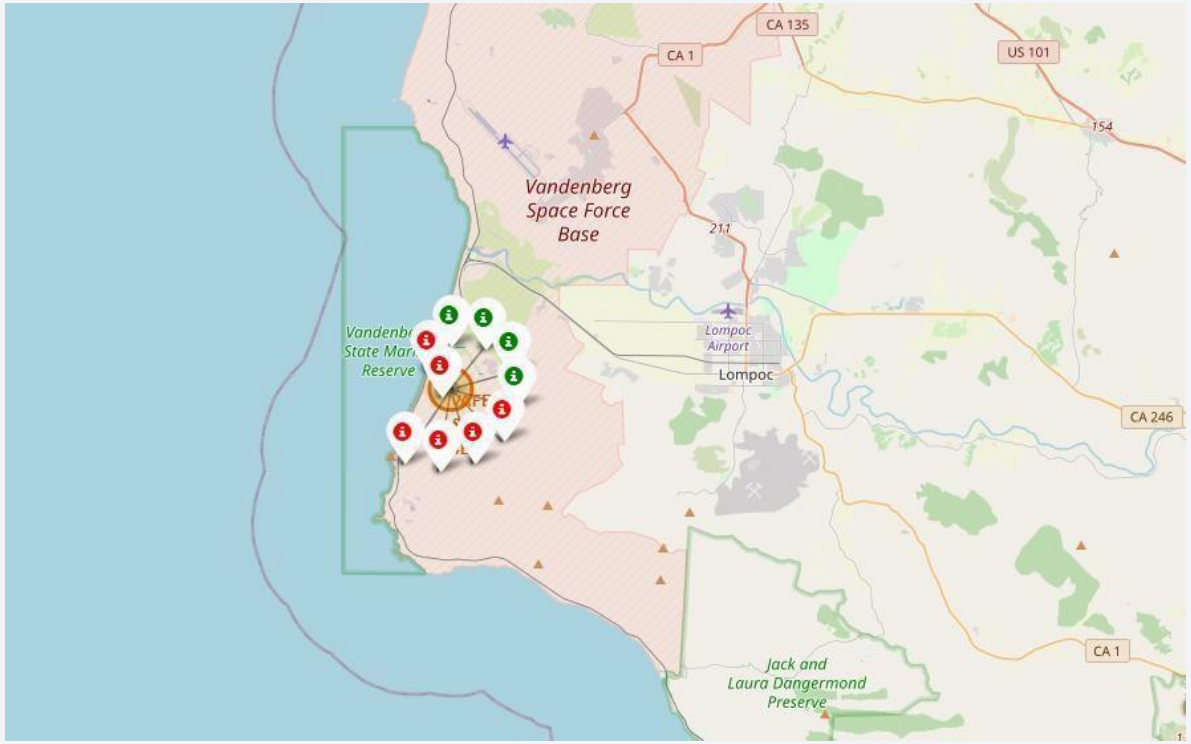
Launch Sites Proximities Analysis

Launch Site locations



As you can see the launch sites are very close and are on the coast.

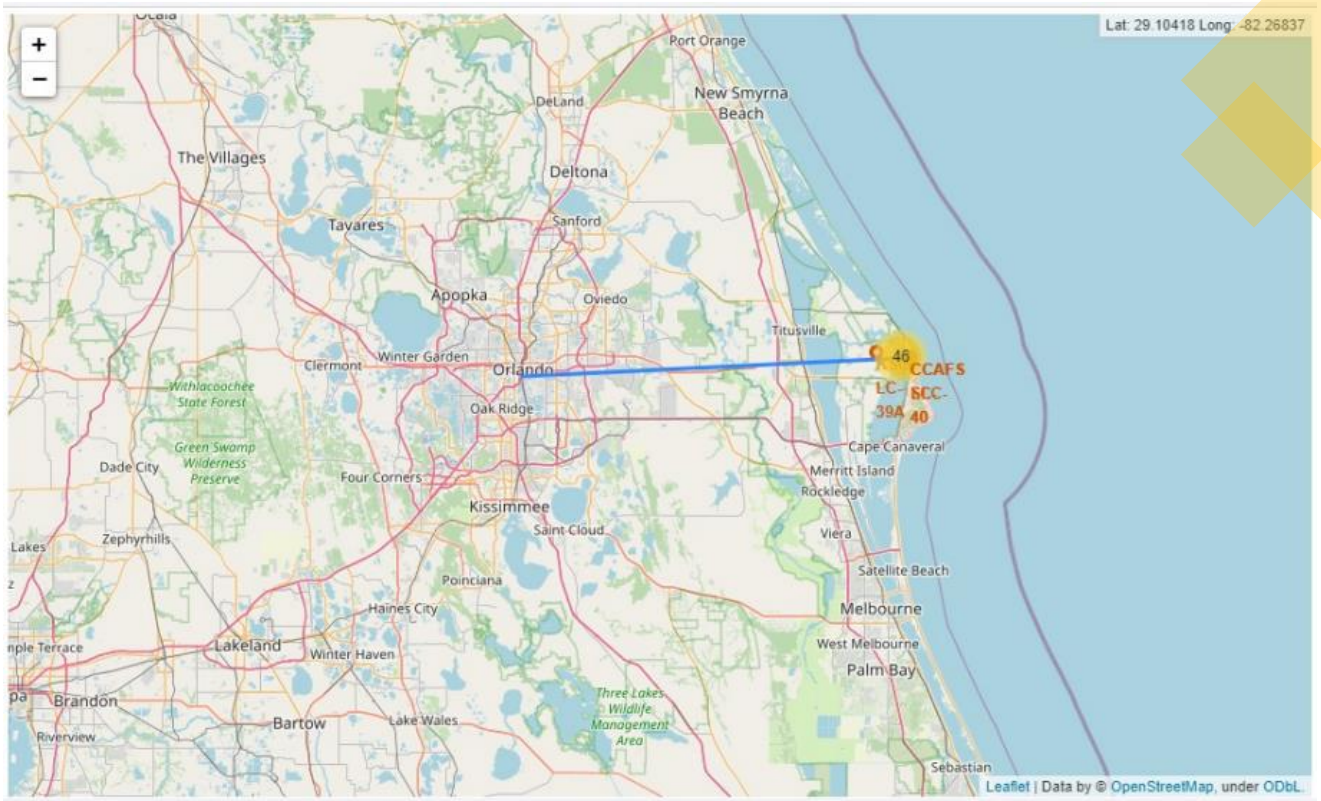
Launch markers



Launches have been grouped into clusters, the green icons stand for successful launches and red for failure.

Key Location Distance

- We can see that the launch site is very close the water, incase something goes wrong with the launch. We can also see that the launch site is considerable distance from any densely populated area.
- The blue line represents the distance from launch area to any location of choice

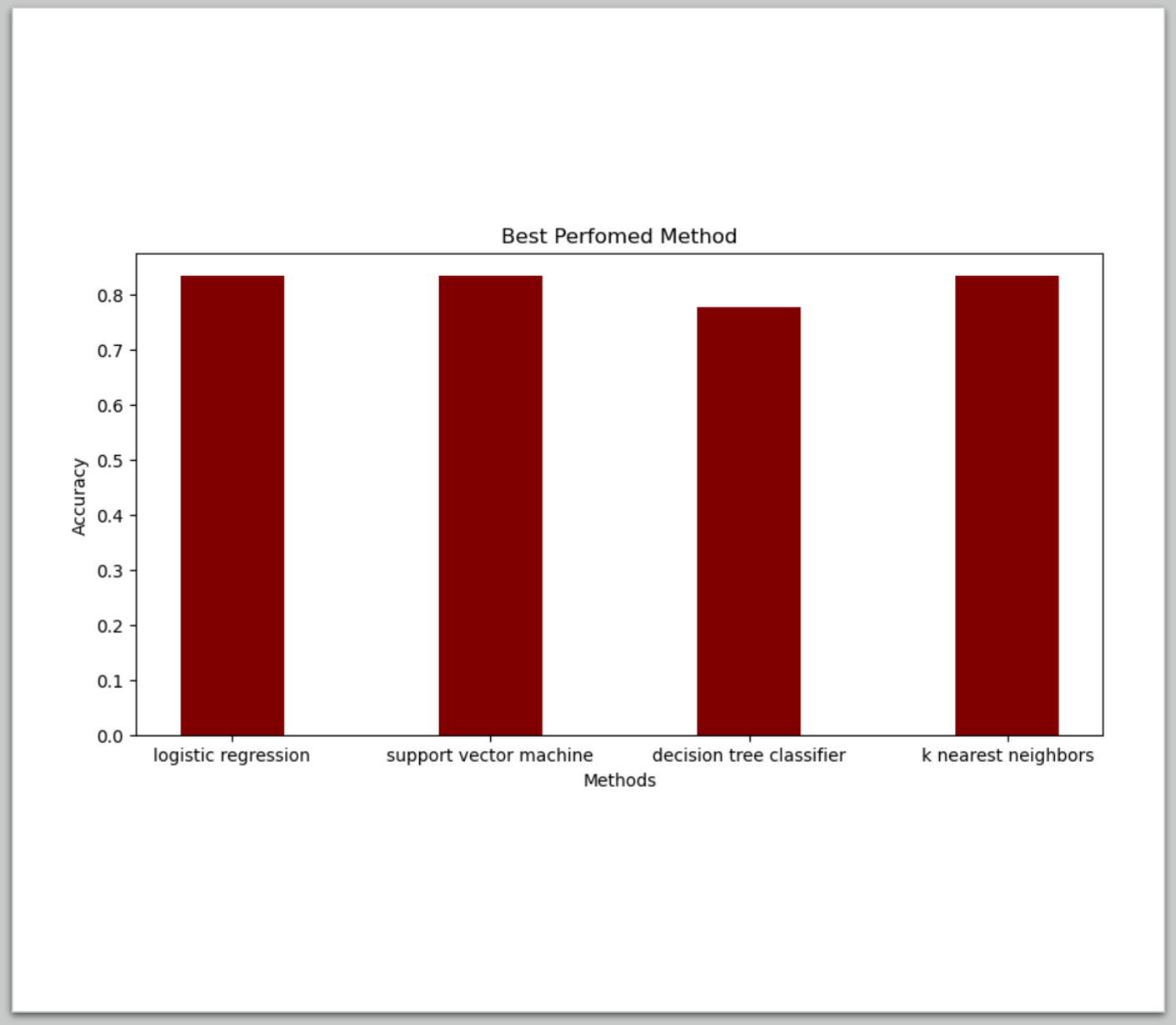


Section 5

Predictive Analysis (Classification)

Classification Accuracy

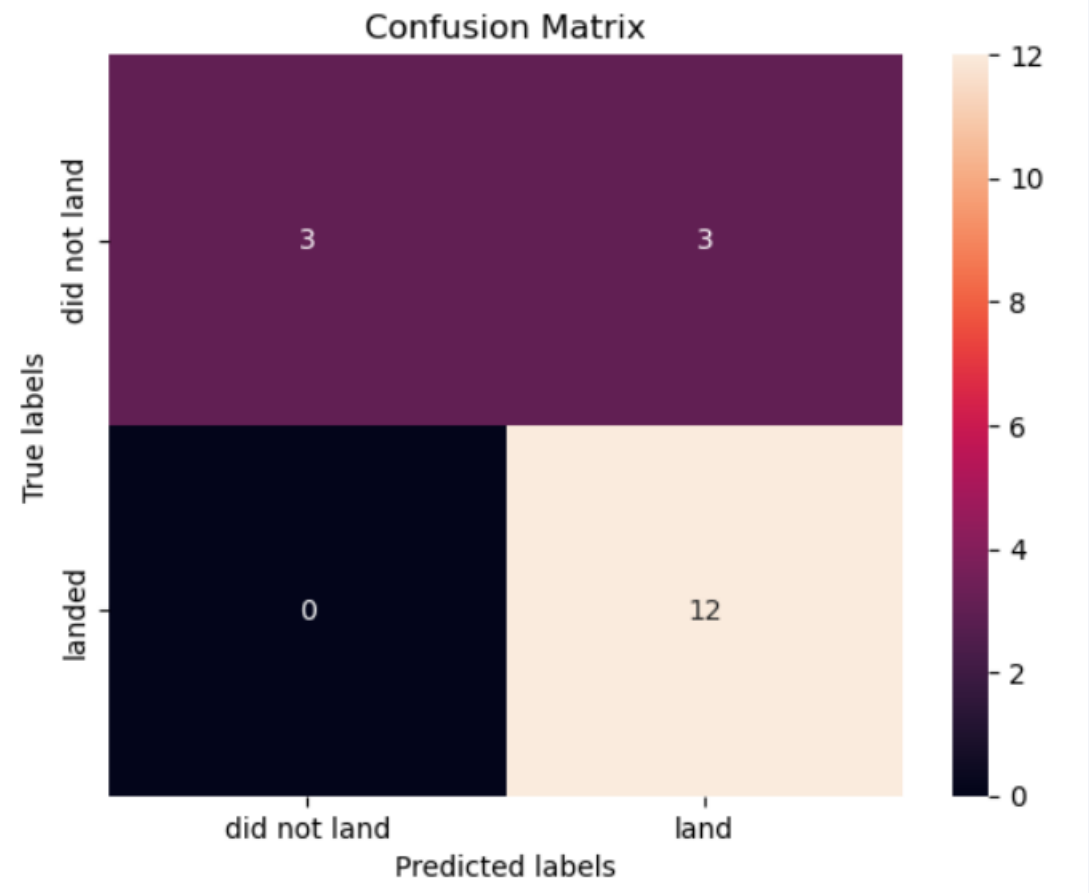
- Logistic regression, SVM and KNN are the models that perform the best and, the decision tree is the model with the less accuracy.



Confusion Matrix

- Correct predictions are on a diagonal from top left to bottom right.

- Since 3 of our models performed the same for the test set, the confusion matrix is the same across models. The models predicted 12 successful landings correctly
- The models predicted 3 unsuccessful landings correctly.
- The models predicted 3 successful landings when this didn't really occur(false positives). Our models over predict successful landings.



Conclusions

- Our task was to develop a machine learning model for Space Y who wants to bid against SpaceX
- Used data from a public SpaceX API and web scraping SpaceX Wikipedia page
- Created data labels and stored data into a DB2 SQL database
- Created a dashboard for visualization
- We created a machine learning model with an accuracy of 83%
- SpaceY can use this model to predict with relatively high accuracy whether a launch will have a successful Stage 1 landing before launch to determine whether the launch should be made or not
- If possible more data should be collected to better determine the best machine learning model and improve accuracy

Thank you!