IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Adrian Brasser
March 2022

Outline

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

Executive Summary

- Summary of methodologies
  - Data collection
  - Data wrangling
  - EDA with data visualization
  - EDA with SQL
  - Building an interactive map with Folium
  - Building a Dashboard with Plotly Dash
  - Predictive analysis (Classification)
- Summary of all results
  - Exploratory data analysis results
  - Interactive analytics demo in screenshots
  - Predictive analysis results

Introduction

- Project background and context
  - The goal is to predict if the SpaceX Falcon 9 first stage will land successfully. The outcome of the landing of the first stage significantly influences the estimated cost of the delivery of goods into space. This prediction is valuable for our company SpaceY to set a competing price of our services in an auction against SpaceX
- Problems you want to find answers
  - What influences if the rocket will land successfully?
  - The effect each relationship with certain rocket variables will impact in determining the success rate of a successful landing.
  - What conditions does SpaceX have to achieve to get the best results and ensure the best rocket success landing rate.

Section 1

Methodology

Methodology

- Data collection methodology:
  - Use SpaceX Rest API
  - Web Scrapping from Wikipedia
- Perform data wrangling (Transforming data for Machine Learning)
  - One Hot Encoding data fields for Machine Learning and dropping irrelevant columns
- Perform exploratory data analysis (EDA) using visualization and SQL
  - Plotting: Scatter plots, bar charts to show relationships between variables to show patterns of data
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection

- We worked with SpaceX launch data that is gathered from the SpaceX REST API.
- This API will give us data about launches, including information about the rocket used,
- payload delivered, launch specifications, landing specifications, and landing outcome.
- Our goal is to use this data to predict whether SpaceX will attempt to land a rocket or not.
- The SpaceX REST API endpoints, or URL, starts with api.spacexdata.com/v4/.
- Another popular data source for obtaining Falcon 9 Launch data is web scraping Wikipedia using BeautifulSoup.

SpaceX API

1. Use SpaceX REST API
2. API returns SpaceX data in .JSON
3. Normalize data into flat data structure like a pandas dataframe

Web Scraping

1. Get HTML Response from Wikipedia
2. Extract data using beatiful soup or pandas
3. Normalize data into flat data structure like a pandas dataframe

Data Collection – SpaceX API

- 1 .Get Response from API
- 2. Convert Response to a .json file
- 3. Apply custom functions to clean data
- 4. Assign list to dictionary then dataframe
- 5. Filter dataframe and export to flat file (.csv)
- GitHub URL: Link

1

```
spacex_url="https://api.spacexdata.com/v4/launches/past"
response = requests.get(spacex_url).json()
```

2

```
response = requests.get(static_json_url).json()
data = pd.json_normalize(response)
```

3

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

```
getBoosterVersion(data)
```

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}
```

5

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

Data Collection - Scraping

- 1.Get Response from HTML
- 2. Create BeautifulSoup Object
- 3. Find tables
- 4. Get column names
- 5. Create of dictionary
- 6. Append data to keys
- 7. Convert dictionary to dataframe
- 8. Dataframe to .csv
- GitHub URL: Link

1
```
page = requests.get(static_url)
```

2
```
soup = BeautifulSoup(page.text, 'html.parser')
```

3
```
html_tables = soup.find_all('table')
```

4
```
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
```

5
```
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'] = []
launch_dict['Payload mass'] = []
launch_dict['Orbit'] = []
launch_dict['Customer'] = []
launch_dict['Launch outcome'] = []
launch_dict['Version Booster']=[]
launch_dict['Booster landing']=[]
launch_dict['Date']=[]
launch_dict['Time']=[]
```

6
```
In [12]: extracted_row = 0
         #Extract each table
         for table_number,table in enumerate(
            # get table row
             for rows in table.find_all("tr")
                 #check to see if first table
```

7
```
df = pd.DataFrame.from_dict(launch_dict)
```

8
```
df.to_csv('spacex_web_scraped.csv', index=False)
```

Data Wrangling

- Load data into dataframe
- Check for missing values and clean it
- Check data types and see if they make sense
- Overview the data using value_counts
- Classify outcomes success, failure into 0 and 1 for later use in model

- GitHub URL to notebook: Link

EDA with Data Visualization

- Visualize data using:
  - Catplots
    - Using for example (scatterplots using y="LaunchSite", x="FlightNumber", hue="Class“)
  - Barplots
    - Sucess Rate vs. Orbit
  - Lineplots
    - Success Rate vs. Year

  - GitHub URL to notebook: Link

EDA with SQL

Many different questions where asked about the data we needed information. We used SQL to query the data to solve the questions below.

- Display the names of the unique launch sites in the space mission
- Display 5 records where launch sites begin with the string 'KSC’
- 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 where the successful landing outcome in drone ship was achieved.
- List the names of the boosters which have success in ground pad 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.
- List the records which will display the month names, successful landing_outcomes in ground pad ,booster versions, launch_site for the months in year 2017
- Rank the count of successful landing_outcomes between the date 2010 06 04 and 2017 03 20 in descending order.

- GitHub URL to notebook: Link

Build an Interactive Map with Folium

- To visualize the Launch Data into an interactive map we took the coordinates (Latitude and Longitude) of each launch site and added a Circle Marker with a label displaying the name of the launch site.
- As we want to analyze the launch outcomes we added a red info sign for failures and a green one for successes.
- Further interesting information was the distance of the launch site to water, railroads and nearby cities.

- GitHub URL to notebook: Link

Build a Dashboard with Plotly Dash

- The dashboard is built with plotly dash and can be published on a website.
- The dashboard includes can be switched between showing all launch sites (LS) or a specific one
  - Donut:
    - All: Donut chart to show the distribution of successful launches by launch sites
    - Specific LS: Success Rate
  - Scatterplot
    - All: Show Correlation between Payload and success for all launch sites
    - Specific LS: Show Correlation between Payload and success for a specific LS
- The individual launches can be filtered by payload mass. This is done with a slider.

- GitHub URL to notebook: Link

Predictive Analysis (Classification)

BUILD MODEL

- Load our dataset into NumPy and Pandas
- Transform Data
- Split our data into training and test data sets
- Check how many test samples we have
- Decide which type of machine learning algorithms we want to use
- Set our parameters and algorithms to GridSearchCV
- Fit our datasets into the GridSearchCVobjects and train our dataset.

EVALUATE MODEL

- Check accuracy for each model
- Tune hyperparameters for each type of algorithms
- Plot Confusion Matrix

IMPROVE MODEL

- Feature Engineering
- Algorithm Tuning

FIND THE BEST PERFORMING CLASSIFICATION MODEL

- The model with the best accuracy score wins the best performing model
- In the notebook there is a dictionary of algorithms with scores at the bottom of the notebook.

GitHub URL to notebook: Link

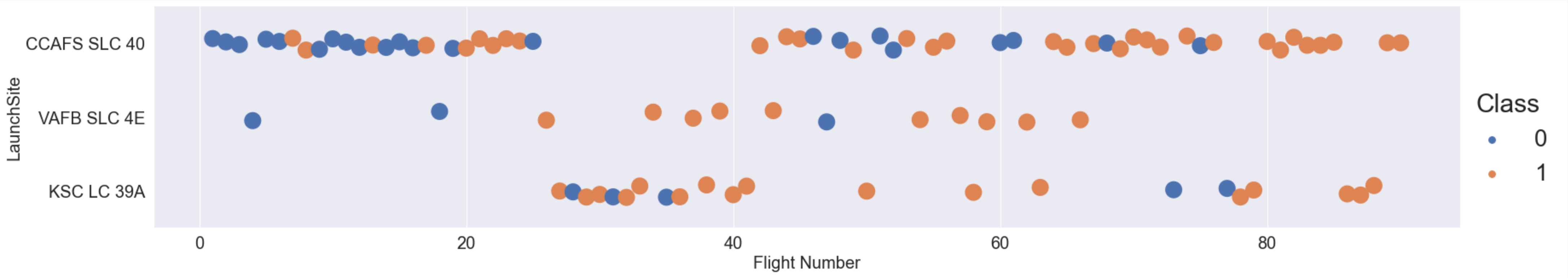
Results

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

Section 2

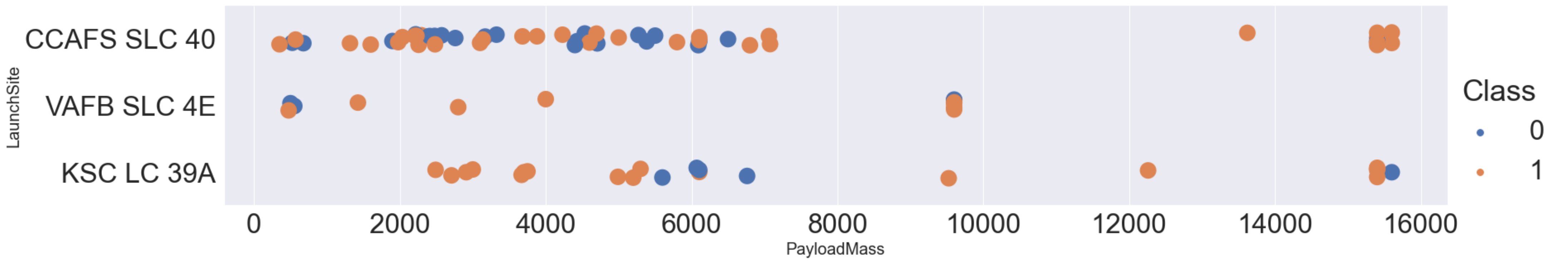
Insights drawn from EDA

Flight Number vs. Launch Site



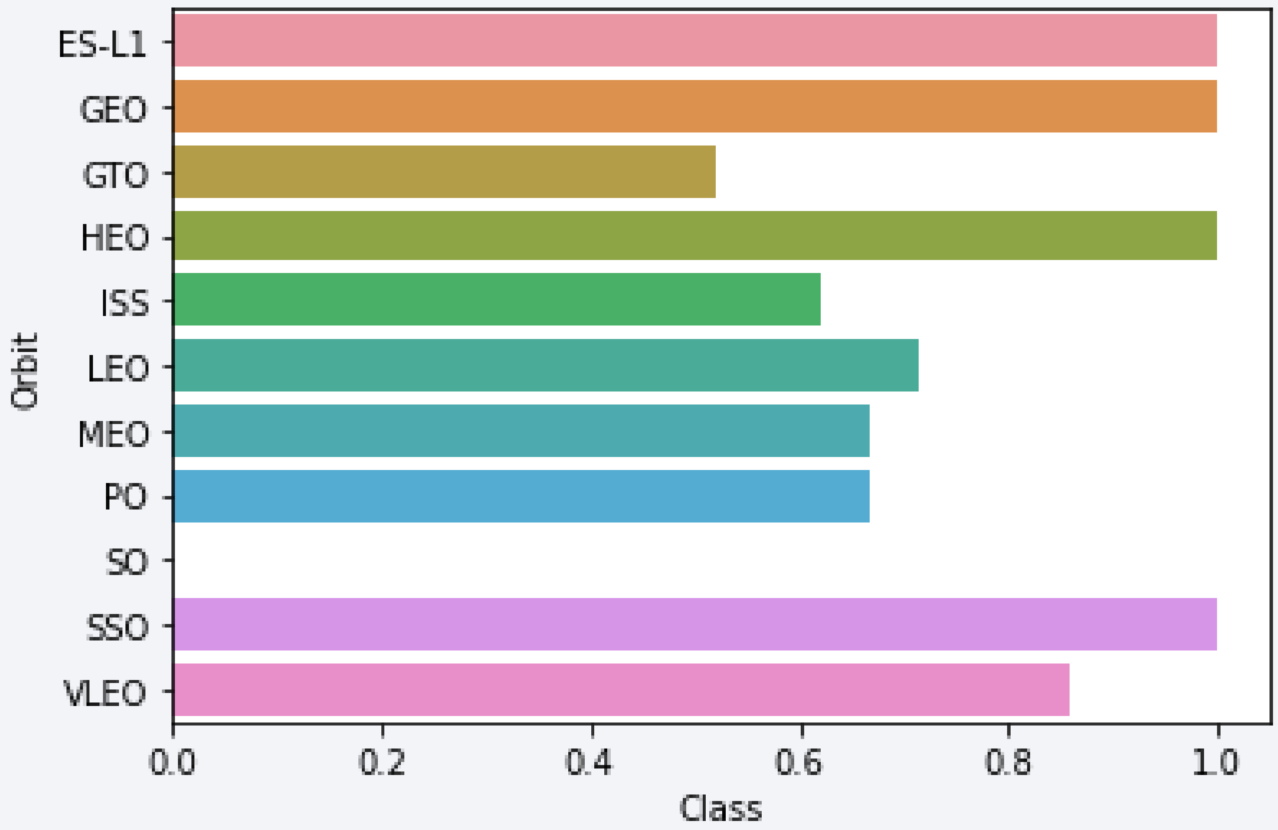
First launches were mostly failures while later launches were mostly successes.

Payload vs. Launch Site



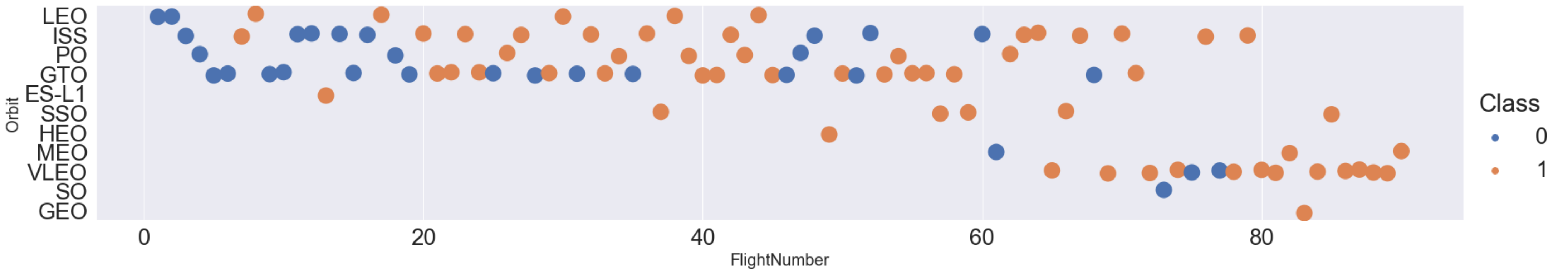
The higher the payload the lower the success probability (Class)

Success Rate vs. Orbit Type



ES-L1, GEO, HEO and SSO have perfect success rates.

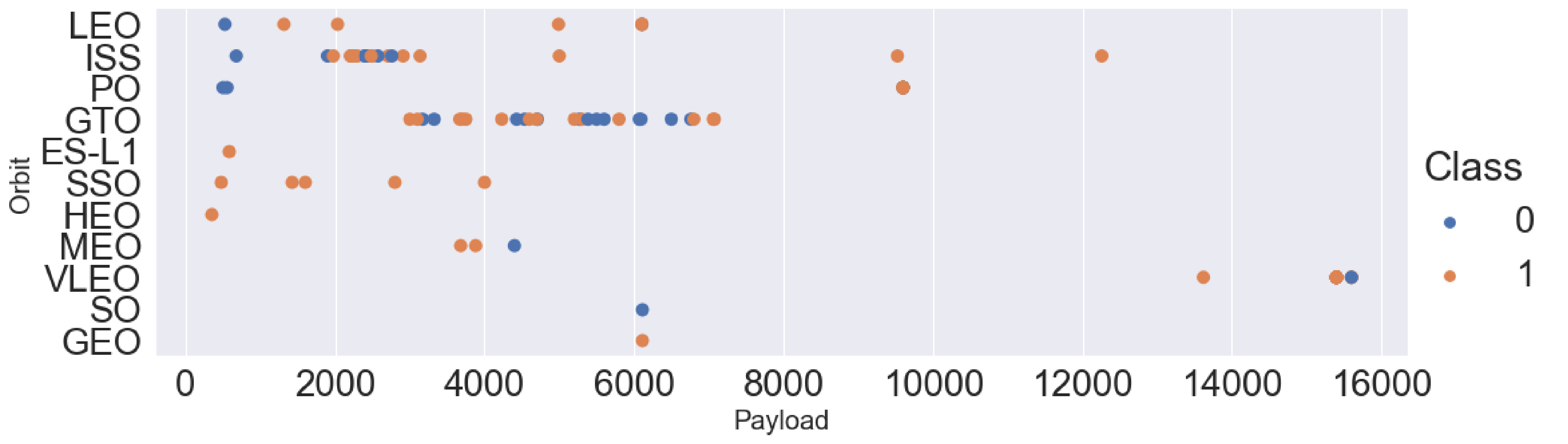
Flight Number vs. Orbit Type



First launches were mostly failures while later launches were mostly successes.

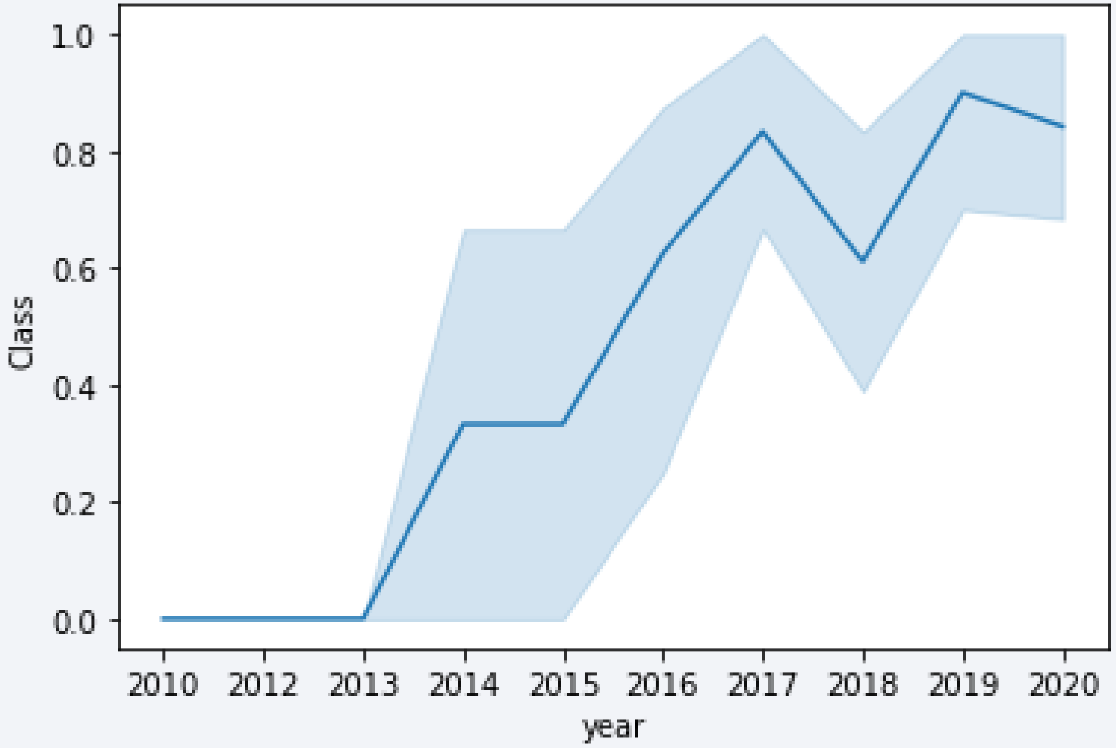
First Flights per Orbit where mostly failures.

Payload vs. Orbit Type



The Payload is correlated to the orbit type.

Launch Success Yearly Trend



The later the year the higher is the success rate (Class).

All Launch Site Names

SQL Query

Select unique(launch_site) from spacexdataset

Result

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

Query Explanation

Using the word ***Unique*** in the query means that it will only show Unique values in the ***launch_site*** column from spacexdataset

Launch Site Names Begin with 'CCA'

SQL Query

Select *
from spacexdataset
where launch_site Like 'CCA%'
limit 5

Query Explanation

Use **Like** and **%** to find names with wildcards
Limit the result set to 5 entries

Result

DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success
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
2012-05-22	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success
2012-10-08	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success

Total Payload Mass

SQL Query

select sum(payload_mass__kg_)
from spacexdataset
where customer like 'NASA%'

Query Explanation

Use **Like** and **%** to find names with wildcards
Sum the column

Result

1
99980

Average Payload Mass by F9 v1.1

SQL Query

```
select avg(payload_mass__kg_)
from spacexdataset
where booster_version like 'F9 v1.1%'
```

Query Explanation

Use **Like** and **%** to find names with wildcards
Average the column

Result

1
2534

First Successful Ground Landing Date

SQL Query

select min(DATE)
from spacexdataset
where landing__outcome = 'Success (ground pad)'

Result

1
2015-12-22

Query Explanation

Take minimum of the column
Contrain with **where**

Successful Drone Ship Landing with Payload between 4000 and 6000

SQL Query

```
select booster_version
from spacexdataset
where landing__outcome = 'Success (drone ship)'
and (payload_mass__kg_ > 4000 and payload_mass__kg_ < 6000)
```

Result

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

Query Explanation

Selecting only
Booster_Version
The
WHERE clause filters the dataset to
Landing_Outcome =
Success (drone ship)
The
AND clause specifies additional
filter conditions
Payload_MASS_KG
4000 AND Payload_MASS_KG 6000

Total Number of Successful and Failure Mission Outcomes

SQL Query

```
Select
SUM(CASE when mission_outcome like 'Success%' THEN 1 ELSE 0
END) as successful,
SUM(CASE when mission_outcome like 'Fail%' THEN 1 ELSE 0 END)
as failure
from spacexdataset
```

Result

successful	failure
100	1

Query Explanation

Use **CASE** to assign 1 when success and 0 when not successful.
Sum it up

Boosters Carried Maximum Payload

SQL Query

Select booster_version
from spacexdataset
where payload_mass__kg_ = (Select max(payload_mass__kg_) from spacexdataset)

Result

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

Query Explanation

Use **Subquery** to find the maximum payload first.
Then compare the payload mass from the table and return when it matches.

2015 Launch Records

SQL Query

```
select booster_version, launch_site, DATE
from spacexdataset
where landing__outcome = 'Failure (drone ship)'
and YEAR(DATE) = '2015'
```

Result

booster_version	launch_site	DATE
F9 v1.1 B1012	CCAFS LC-40	2015-01-10
F9 v1.1 B1015	CCAFS LC-40	2015-04-14

Query Explanation

Get **Year** of the date column and compare if it is in the year 2015.

And if it is a failure

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

SQL Query

```
Select landing__outcome, Count(*) as Count
from spacexdataset
where (landing__outcome = 'Failure (drone ship)' or
landing__outcome = 'Success (ground pad)')
and (Date > '2010-06-04' and Date < '2017-03-20')
group by landing__outcome
order by Count desc
```

Result

landing__outcome	COUNT
Failure (drone ship)	5
Success (ground pad)	3

Query Explanation

Group by landing outcome groups the occurences in that column.
Order by count orders by the count column.

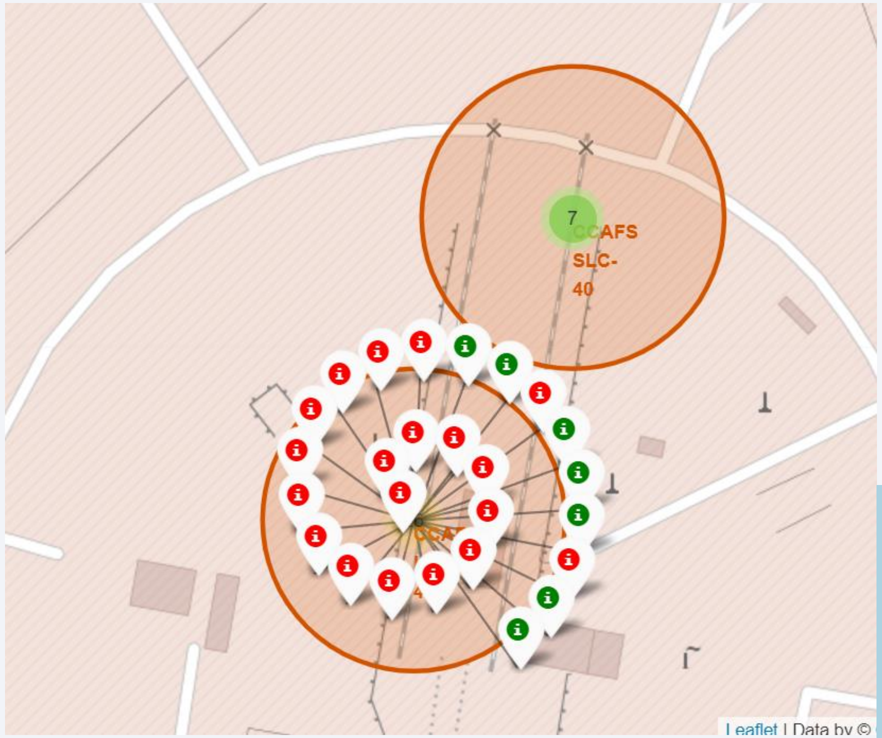
Section 3

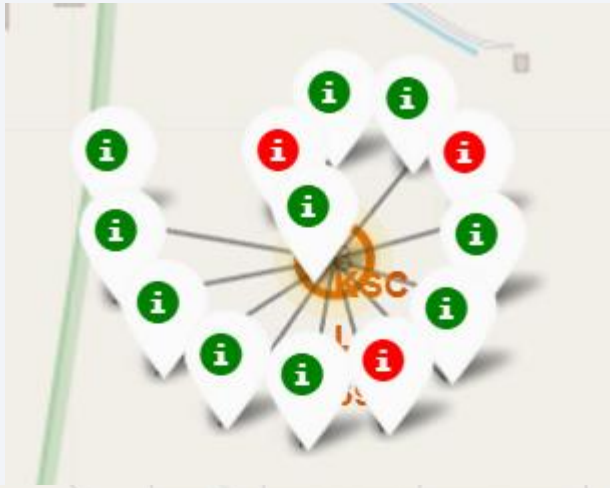
Launch Sites Proximities Analysis

SpaceX launch Sites

We can see that the SpaceX launch sites are in the west and east coasts of the United States of America.

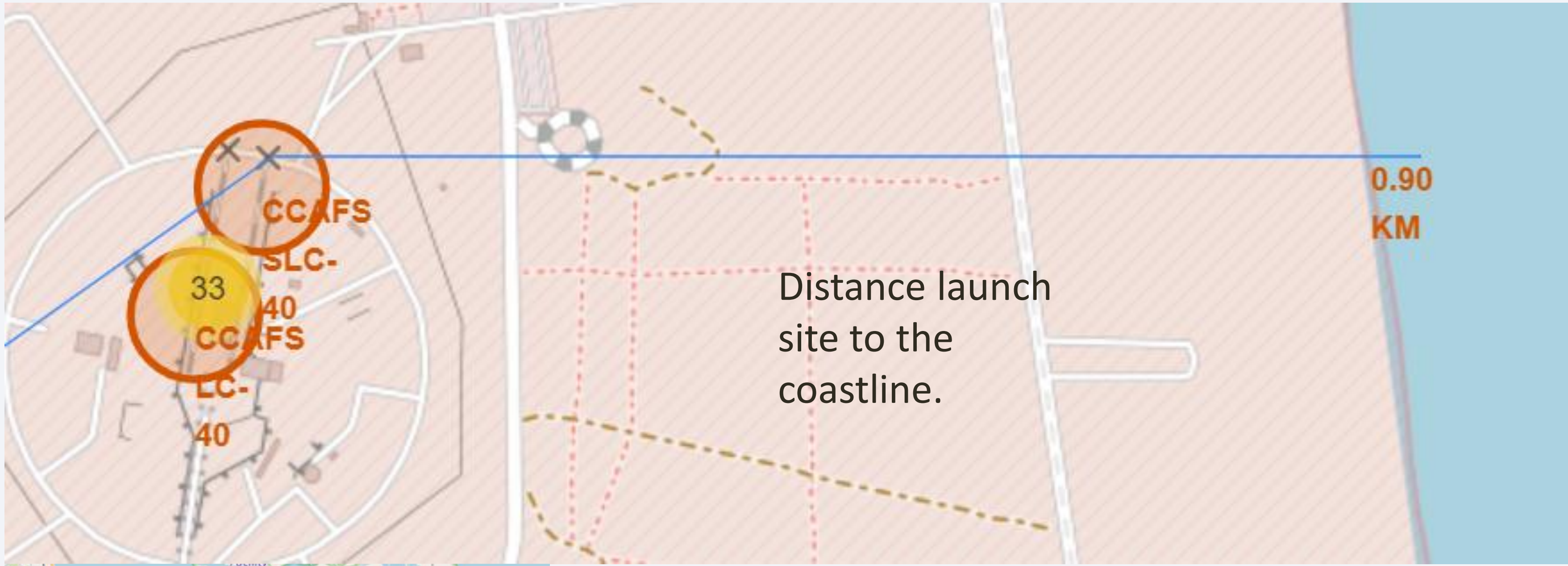
Colour Labelled Markers



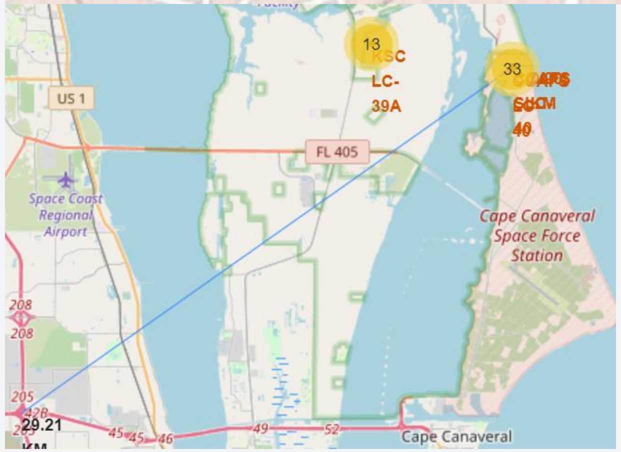


Examples of the Florida launch sites with succesful launches (green) and failures (red).

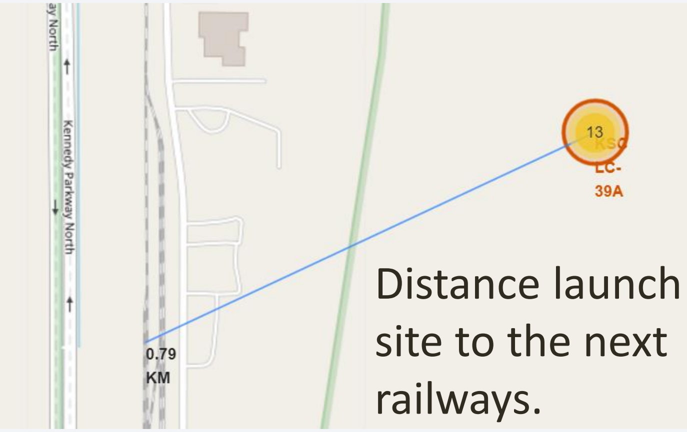
Distance to Specific Objects



Distance launch site to the coastline.



Distance launch site to the next big city (Florida, 29.21km).



Distance launch site to the next railways.

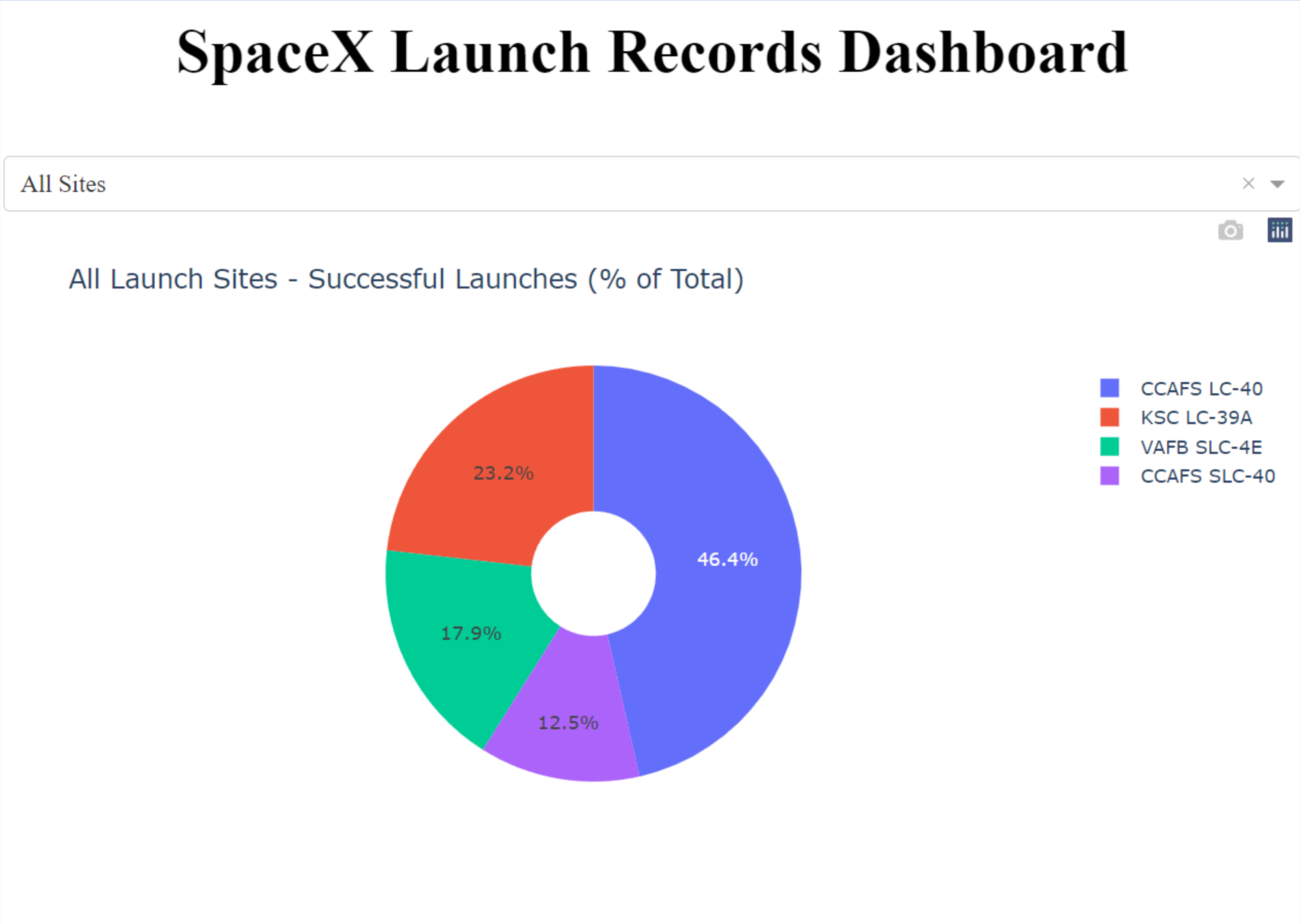
Questions:

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

Section 4

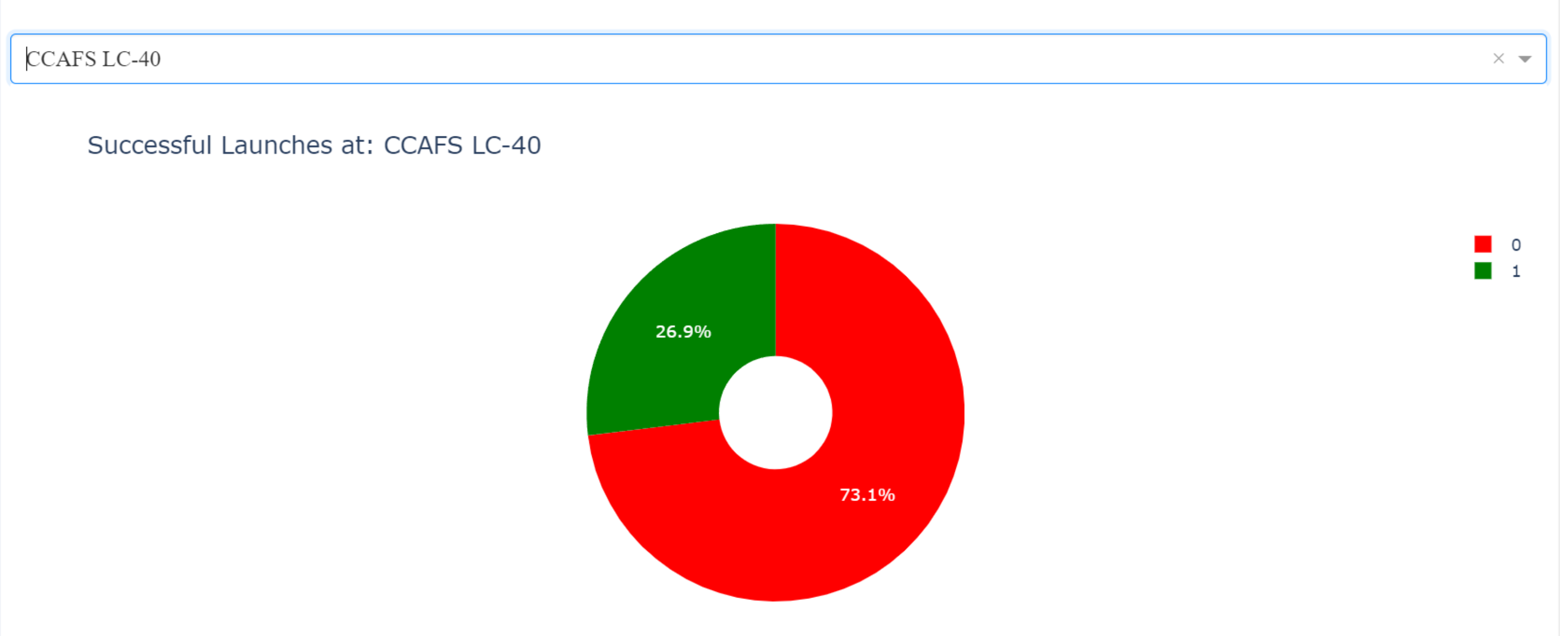
Build a Dashboard with Plotly Dash

Successful Launches (% of Total)



The biggest share of the successful launches came from launch site: CCAFS LC-40

Fraction of Successful Launches



Despite that 73.1% of the launches were failures this launch site contributed to most of the successful launches.

Correlation between Payload and Success

Higher Payload mass leads to failures more often.

Section 5

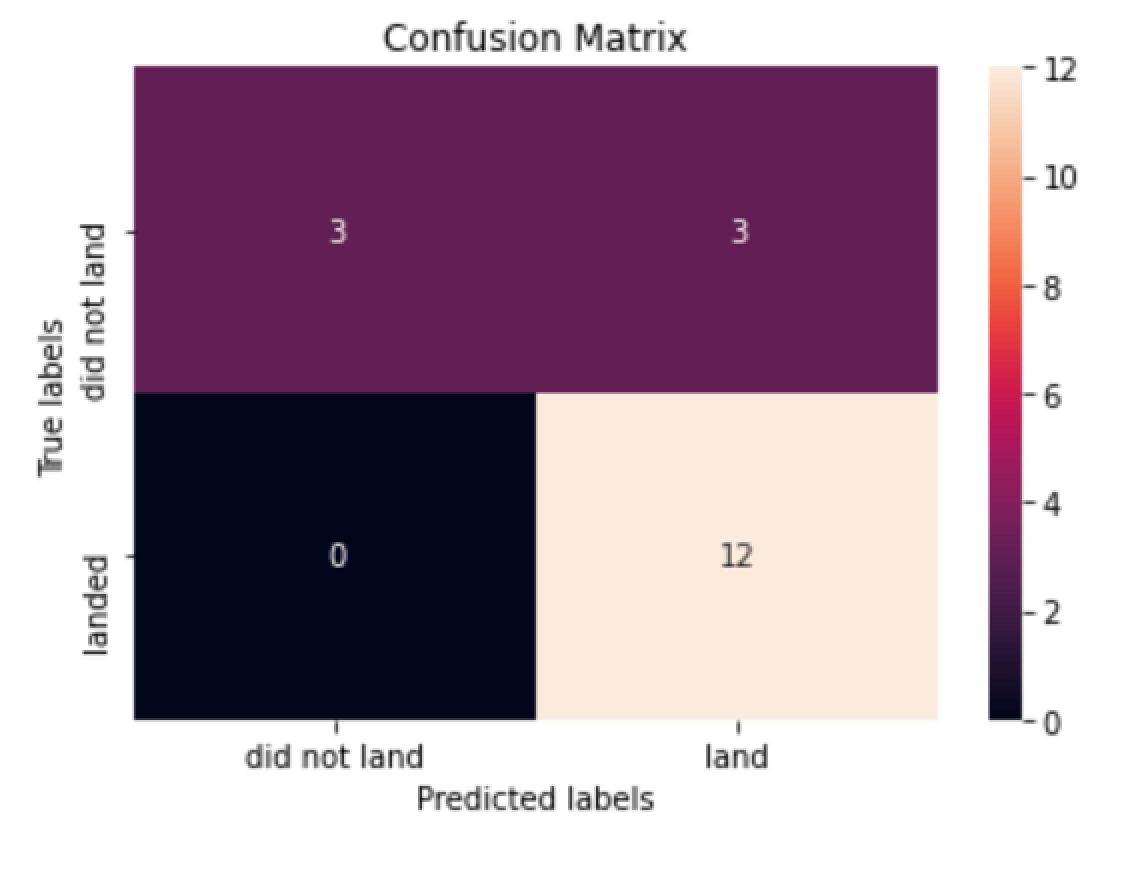
Predictive Analysis (Classification)

Classification Accuracy

KNN, LogisticRegression and SVM all have a Score of 83.33%

All of these 3 Models seem to be ok to use.

Confusion Matrix



The Confusion Matrix shows the **true labels** in the rows and what the model predicted (**predicted labels**) in the columns.

All 12 successful landings were predicted correctly.
The 6 not failed landings were 3 times predicted as successful and 3 times as failures.

Conclusions

- What we found:
- Higher Payload → lower probability of success.
- With more launch experience the success rate is higher.
- The biggest share of the successful launches came from launch site: CCAFS LC-40
- The orbits ES-L1, GEO, HEO and SSO have perfect success rates.

Appendix

- The calculation and visualization of the results was done with python using Jupyter Notebooks and PyCharm.

Thank you!