IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Julian Stricker
6th May 2022

Outline

- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion

Executive Summary

- Summary of methodologies
  - Data collection
  - Data wrangling
  - EDA with data visualization
  - EDA with SQL
  - Building an interactive map (Folium)
  - Building an interactive 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 aim of the project is to predict if the Falcon 9 first stage will land successfully. SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars while other providers cost upwards of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage. Therefore, if we can determine if the first stage will land, we can determine the cost of a launch. This provides valuable information for other companies who also have intentions of launching rockets into space.

- Problems you want to find answers
  - What factors influence the outcome of the landing?
    - Identify significant variables and their influence on the success rate
  - Under what conditions is a successful landing most likely?

Section 1

Methodology

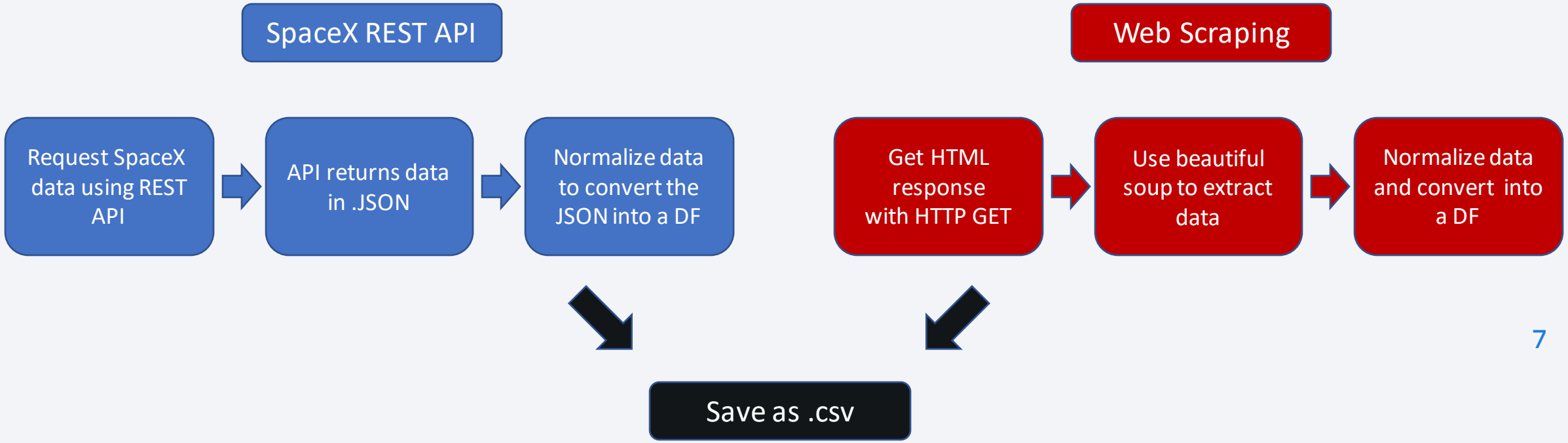
Methodology

Executive Summary

- Data collection methodology:
  - SpaceX Rest API
  - Web Scaping Wikipedia
- Perform data wrangling
  - Transforming data to enable EDA and Machine Learning using One Hot Encoding
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification model
  - How to build, tune and evaluate classification models

Data Collection

- The datasets used were gathered from 2 sources: SpaceX REST API and Wikipedia (Web Scaping)
- The sources provide useful information such as rocket version, payload, launch & landing parameters and landing outcome



Data Collection – SpaceX API

1. Response from API

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

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

2. Covert .JSON into Pandas DF

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

3. Apply custom functions to obtain clean data

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

4. Assign list to dictionary and create DF

```
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}
```

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

5. Filter and export as .csv

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

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

[Link to GitHub Notebook]

Data Collection - Scraping

1. Get HTTP Response

```
# use requests.get() method with the provided static_url
data = requests.get(static_url).text
```

2. Create BeautifulSoup object

```
# Use BeautifulSoup() to create a BeautifulSoup object
soup = BeautifulSoup(data, 'html.parser')
```

3. Select relevant table and extract column names

```
html_tables = soup.find_all('table')
first_launch_table = html_tables[2]
column_names = []
for x in range(len(colnames)):
    name = extract_column_from_header(colnames[x])
    if (name is not None and len(name) > 0):
        column_names.append(name)
```

4. 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 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. Append data into dictionary (full view on GitHub)

```
extracted_row = 0
#Extract each table
for table_number,table in enumerate(soup.find_all('table'
   # get table row
    for rows in table.find_all("tr"):
```

6. Convert to dataframe and extract as .csv

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

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

[Link to GitHub Notebook]

Data Wrangling

- Objectives: Perform EDA to find patterns and determine labels for training supervised models

1. Number of launches on each site

```
df['LaunchSite'].value_counts()
```

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

2. Number and occurrence of each orbit

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

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

[Link to GitHub Notebook]

3. Number and occurence of mission outcome

```
landing_outcomes = df['Outcome'].value_counts()
landing_outcomes
```

```
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: Outcome, dtype: int64
```

ASDS = drone ship
RTLS = ground pad
True = successfull landing
False = unsuccessfull landing

4. Create a landing outcome label

```
# landing_class = 0 if bad_outcome
# landing_class = 1 otherwise
liste = []

for outcome in df['Outcome']:
    if outcome in bad_outcomes:
        liste.append(0)
    else:
        liste.append(1)

landing_class = liste
```

```
df['Class']=landing_class
df[['Class']].head(8)
```

5. Calculate succes rate & export CSV

```
df["Class"].mean()
```

```
0.6666666666666666
```

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

EDA with Data Visualization

Scatter Plots

- Flight Number vs. Payload Mass
- Flight Number vs. Launch Site
- Payload Mass vs. Launch Site
- Orbit Type vs. Flight Number
- Payload Mass vs. Orbit Type

Scatter plots can visualize the relationship (correlation) between variables and is able to represent many data points.

Bar Graph

- Success Rate vs. Orbit

Bar graphs are used to compare different groups of data at a glance. One axis represents the categories and the other axis displays the corresponding discrete value.

Line Graph

- Success Rate vs. Year

Line graphs are often used to visualize the change of a variable over time to identify trends and predict future development.

Link to GitHub Notebook

EDA with SQL

- Performed SQL queries to gather additional insights:

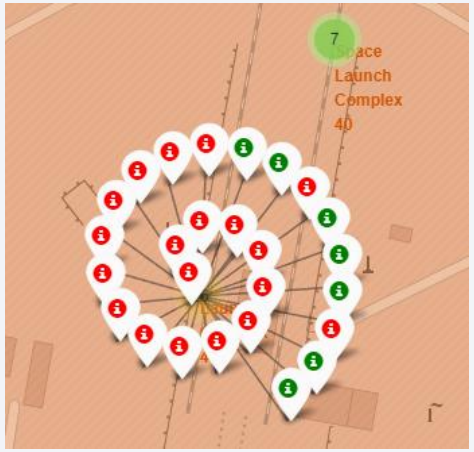
- *Display the names of the unique launch sites in the space mission*
- *Display 5 records where launch sites begin with the string 'CCA'*
- *Display the total payload mass carried by boosters launched by NASA (CRS)*
- *Display average payload mass carried by booster version F9 v1.1*
- *List the date when the first successful landing outcome in ground pad was 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*
- *List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015*
- *Rank the count of landing outcomes (Failure or Success) between the date 2010-06-04 and 2017-03-20, in descending order*

Link to GitHub Notebook

Build an Interactive Map with Folium

- Creation of an interactive map to visualize Lauch Sites by adding objects:

1. Launch site marker



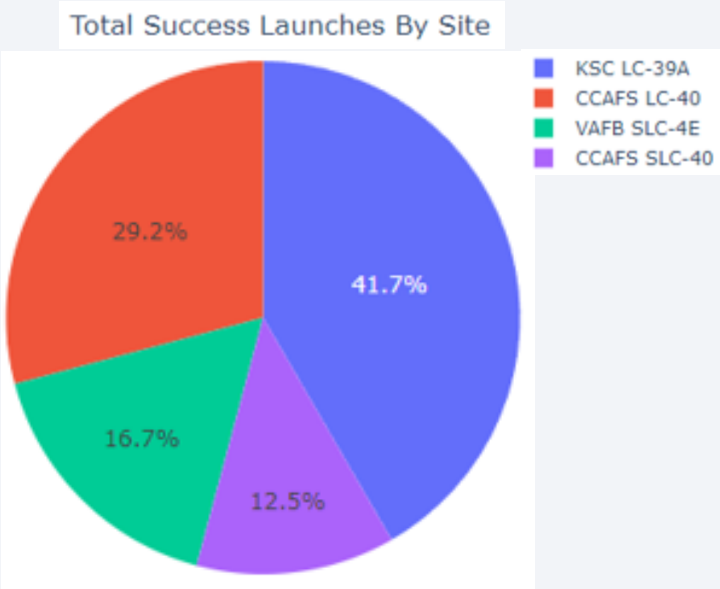
2. Launch outcomes as red/green markers on each site summarized in clusters

3. Distance lines between launch site and its proximities

[Link to GitHub Notebook]

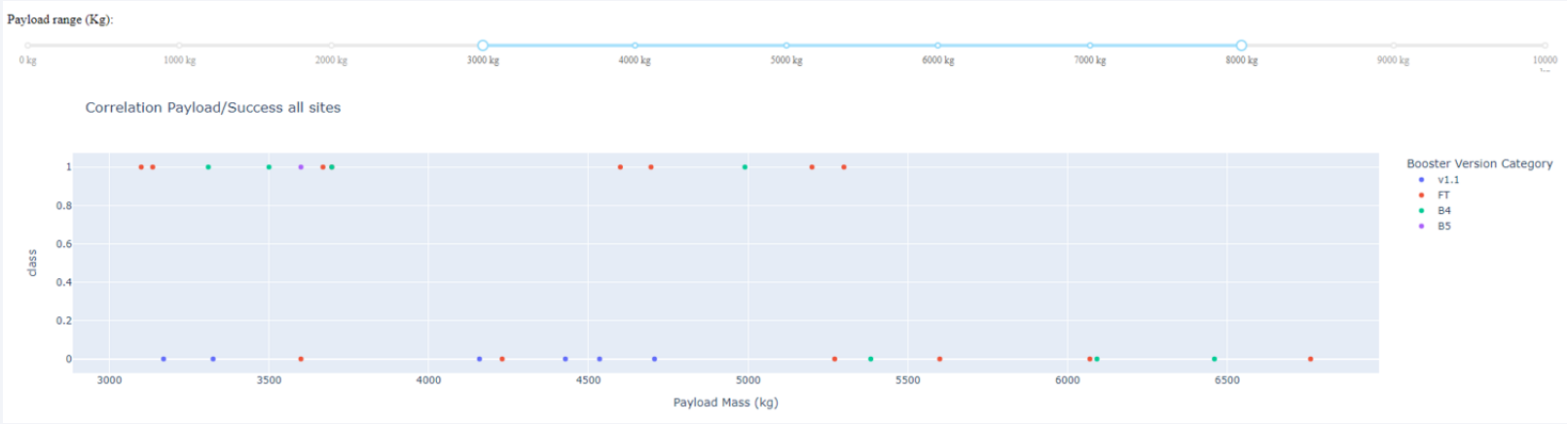
Build an interactive Dashboard with Plotly Dash

Pie Chart(s)



- Displays total success launches by site to enable a comparison
- Interactivity: Switch between 'All Sites' or display success rate of specific launch site

Scatter Plot(s)



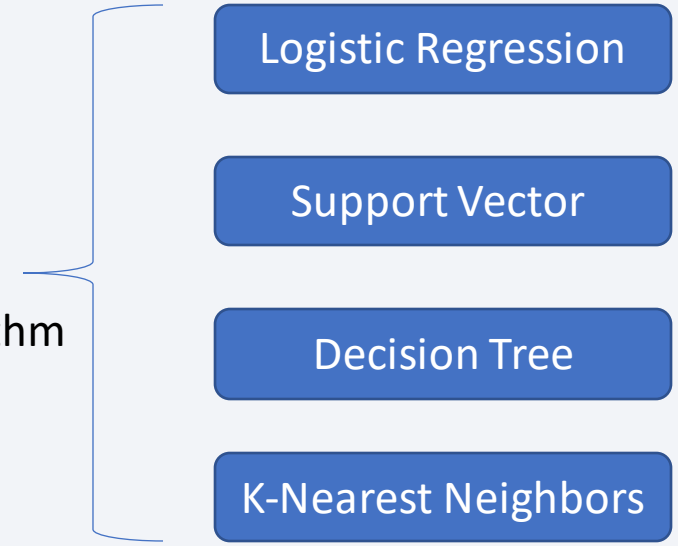
- Displays correlation between Payload Mass and Successful landing to identify significant patterns
- Interactivity: Switch between 'All Sites' or specific launch site and select Payload range

[Link to GitHub Notebook]

Predictive Analysis (Classification)

BUILD MODEL

- Load prepared datasets into Pandas and NumPy
- Transform data
- Split data into train and test subsets
- Select machine learning algorithms to be evaluated
- Create GridSearchCV object & parameters for each algorithm
- Fit & train the dataset – obtain best hyperparameters



EVALUATE MODEL

- Calculate accuracy of each algorithm using best hyperparameters
- Visualization: plot confusion matrix

IMPROVE MODEL

- Feature Engineering
- Algorithm Tuning

FIND THE BEST PERFORMING CLASSIFICATION MODEL

- Select model with best accuracy score

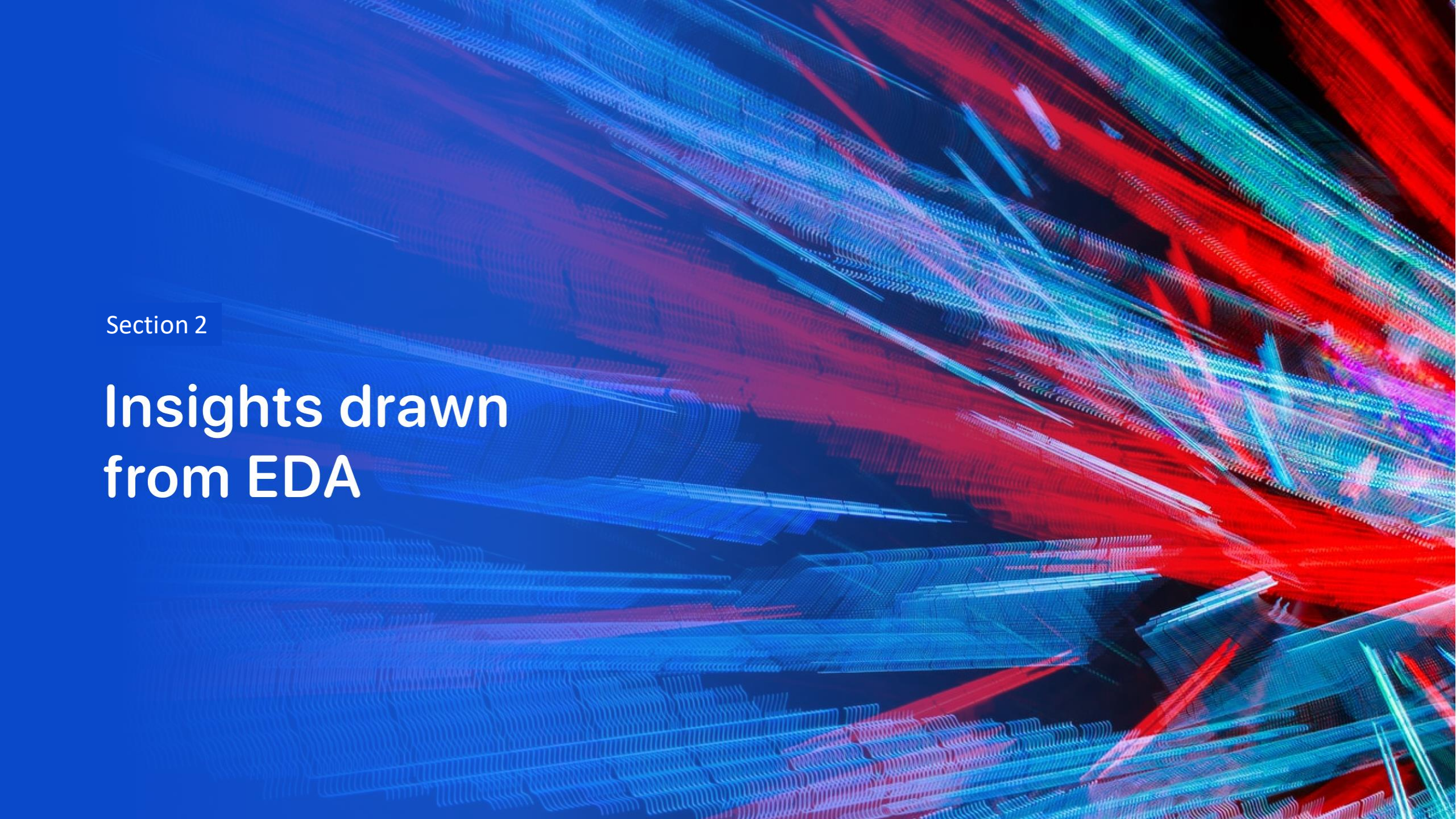
Link to GitHub Notebook

Results

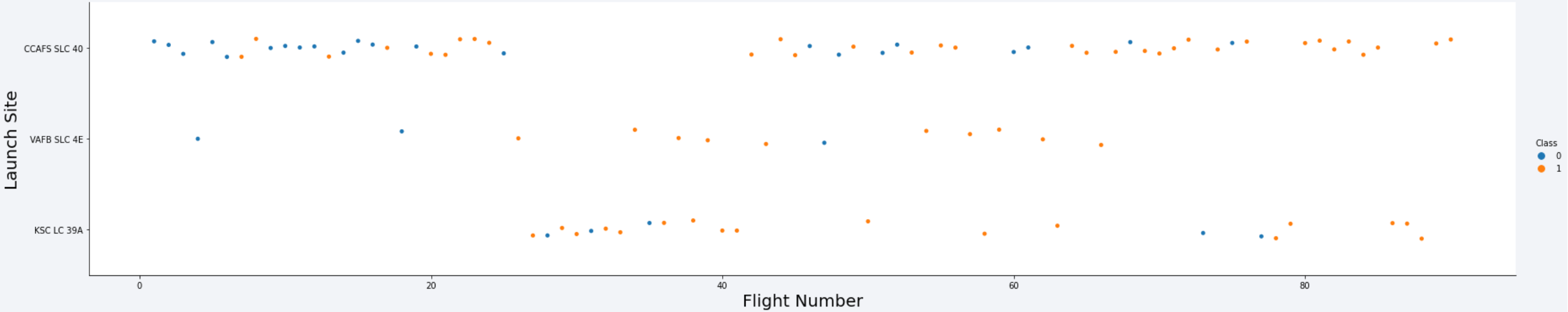
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

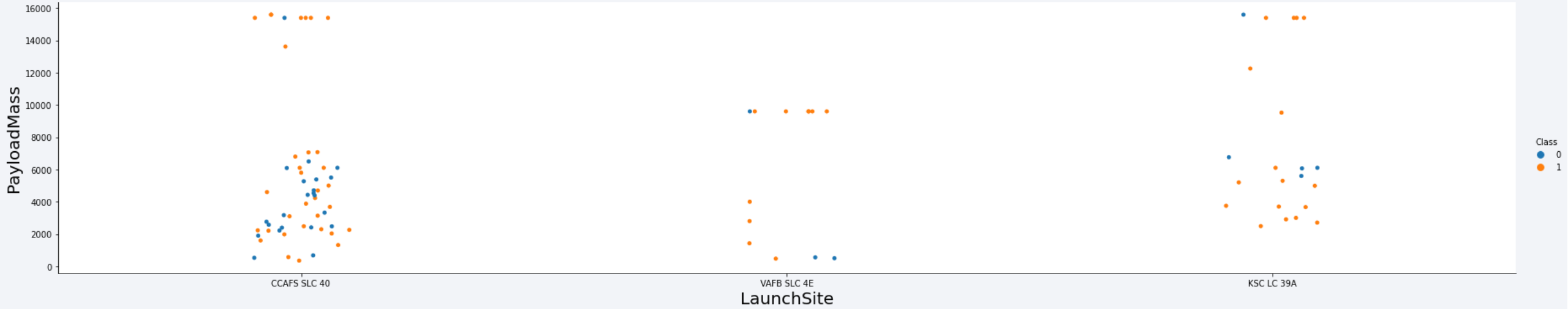


Flight Number vs. Launch Site



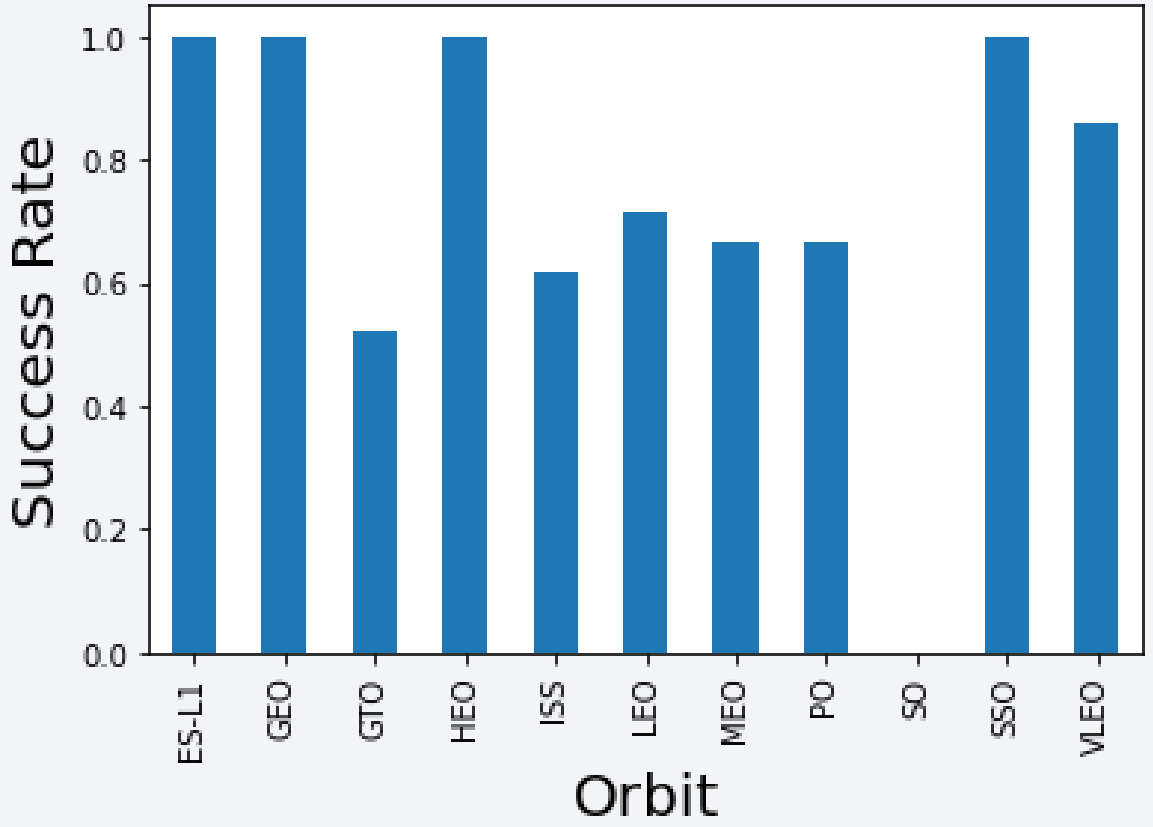
- Success rate increases with Flight Number
- Slightly lower success rate on CCAFS LC-40
- Most launches on CCAFS LC-40

Payload vs. Launch Site



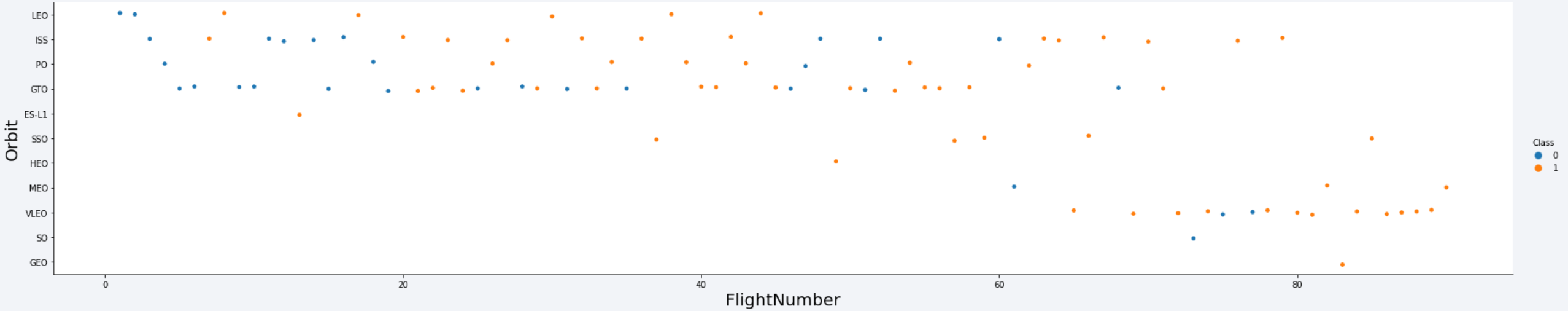
- Success rate on CCAFS LC-40 increases significantly with Payload Mass
- No launches with more than 10000kg Payload Mass on VAFB SLC 4E

Success Rate vs. Orbit Type



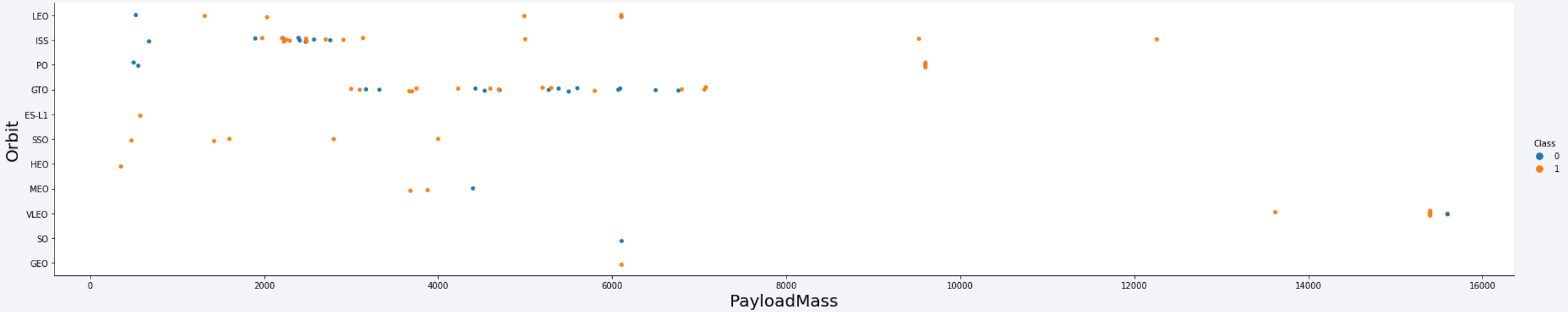
- **ES-L1, GEO, HEO and SSO have 100% Success Rate**
- **GTO: lowest Success Rate**

Flight Number vs. Orbit Type



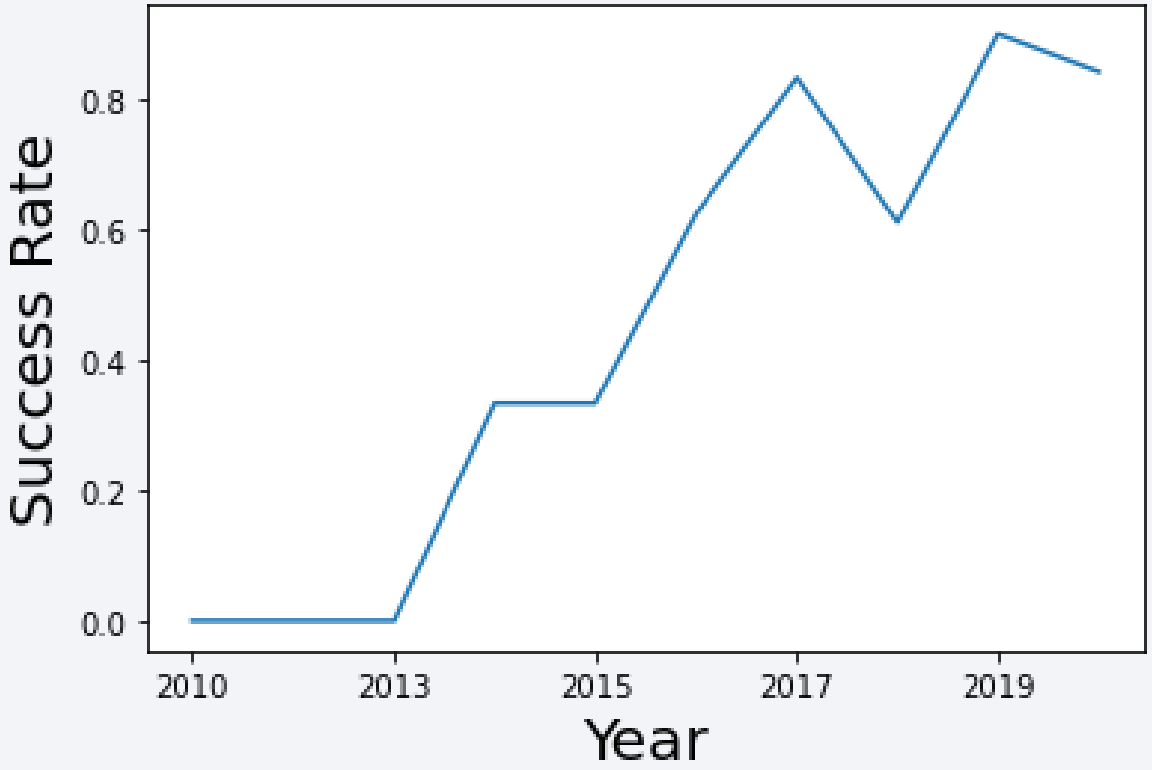
- Increasing Success Rate with Flight Number in LEO Orbit
- No relationship beween Flight Number in GTO Orbit
- Flight Numbers > 60 use VLEO instead LEO Orbit

Payload vs. Orbit Type



- Heavy Payloads indicate a positive Success Rate for SO, LEO and ISS Orbit
- GTO shows no realtionship regarding Payload

Launch Success Yearly Trend



- Success rate kept increasing since 2013 until 2020

All Launch Site Names

Display the names of the unique launch sites in the space mission

```
%sql select distinct(LAUNCH_SITE) from SPACEXDATASET
```

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

QUERY EXPLENATION

- Distinct returns unique values only

Launch Site Names Begin with 'CCA'

Display 5 records where launch sites begin with the string 'CCA'

```
%sql select * from SPACEXDATASET 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

QUERY EXPLENATION

- LIKE 'ABC%' searches for records that begin with 'ABC'
- LIMIT 5: Only first five records will be displayed

Total Payload Mass

Display the total payload mass carried by boosters launched by NASA (CRS)

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

1
45596

QUERY EXPLENATION

- SUM returns cumulative value of the column
- WHERE filters the dataset so it only includes a specific customer

Average Payload Mass by F9 v1.1

Display average payload mass carried by booster version F9 v1.1

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

1
2928

QUERY EXPLENATION

- AVG returns average value of the column
- WHERE filters the dataset so it only includes a specific booster version

First Successful Ground Landing Date

List the date when the first successful landing outcome in ground pad was acheived.

```
%sql select min(DATE) from SPACEXDATASET where landing__outcome = 'Success (ground pad)'
```

1
2015-12-22

QUERY EXPLENATION

- MIN returns lowest value (here: date) of the column
- WHERE filters the dataset so it only includes a specific landing outcome

Successful Drone Ship Landing with Payload between 4000 and 6000

List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

```
%sql select booster_version from SPACEXDATASET where landing__outcome = 'Success (drone ship)' and payload_mass__kg_ BETWEEN 4000 AND 6000
```

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

QUERY EXPLENATION

- WHERE filters the dataset so it only includes a specific landing outcome/payload
- And operator combines multiple statements in the WHERE clause
- BETWEEN X AND Y selects range for payload column

Total Number of Successful and Failure Mission Outcomes

List the total number of successful and failure mission outcomes

```
%sql select mission_outcome, count(mission_outcome) as total_number from SPACEXDATASET group by mission_outcome
```

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

QUERY EXPLENATION

- COUNT returns the total number of records for each mission outcome
- AS operator renames column in output
- GROUP BY returns grouped results, e.g the mission_outcome column contains three possible records

Boosters Carried Maximum Payload

List the names of the booster_versions which have carried the maximum payload mass.

```
%sql select booster_version from SPACEXDATASET where payload_mass__kg_ = (select max(payload_mass__kg_) from SPACEXDATASET)
```

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 EXPLENATION

- Subquery in WHERE clause to filter only the maximum payload mass

2015 Launch Records

List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

```
%sql select booster_version, launch_site from SPACEXDATASET 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

QUERY EXPLENATION

- WHERE filters the dataset so it only includes a specific landing outcome/date
- And operator combines multiple statements in the WHERE clause

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

Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order

```
%sql SELECT count(landing__outcome) as total_number, landing__outcome from SPACEXDATASET where date BETWEEN '2010-06-04' and '2017-03-20' group by lan
ding__outcome order by total_number DESC
```

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

QUERY EXPLENATION

- COUNT returns the total number of records for each landing outcome
- And operator combines multiple statements in the WHERE clause
- GROUP BY returns grouped results
- Order By DESC orders in descending order

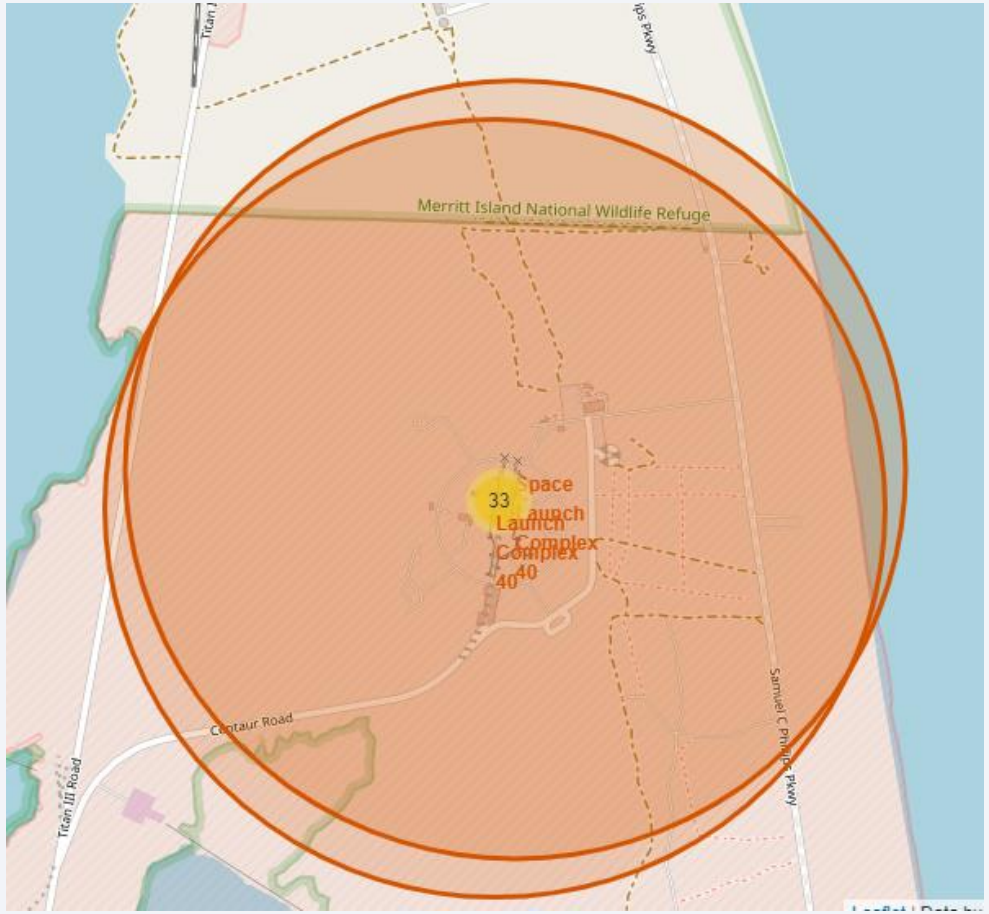

Section 3
Launch Sites
Proximities Analysis

All lauch site markers

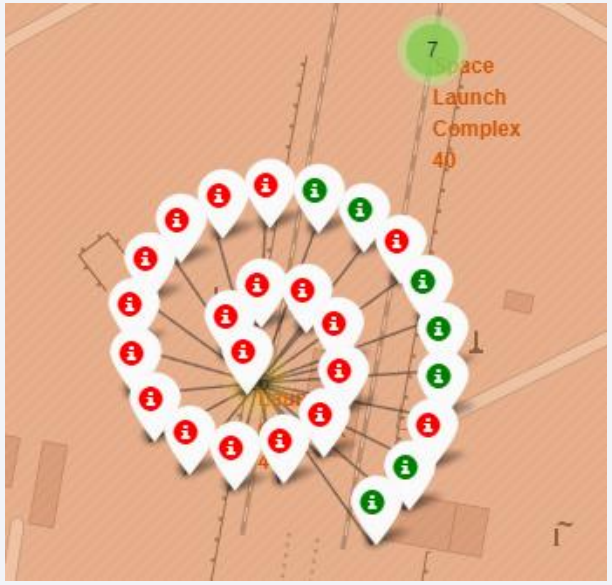
SpaceX Launch Sites

- US east coast: Florida
- US west coast: California

Marker Cluster with color label



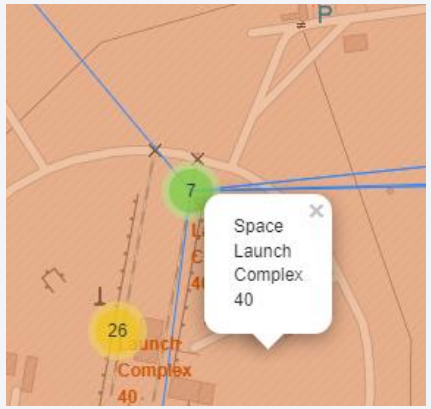
Clustered markers

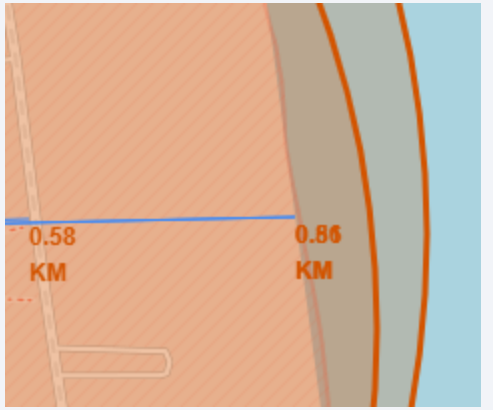


Green: Successfull Launch
Red: Failure

Lauch Site: Distance to Proximities

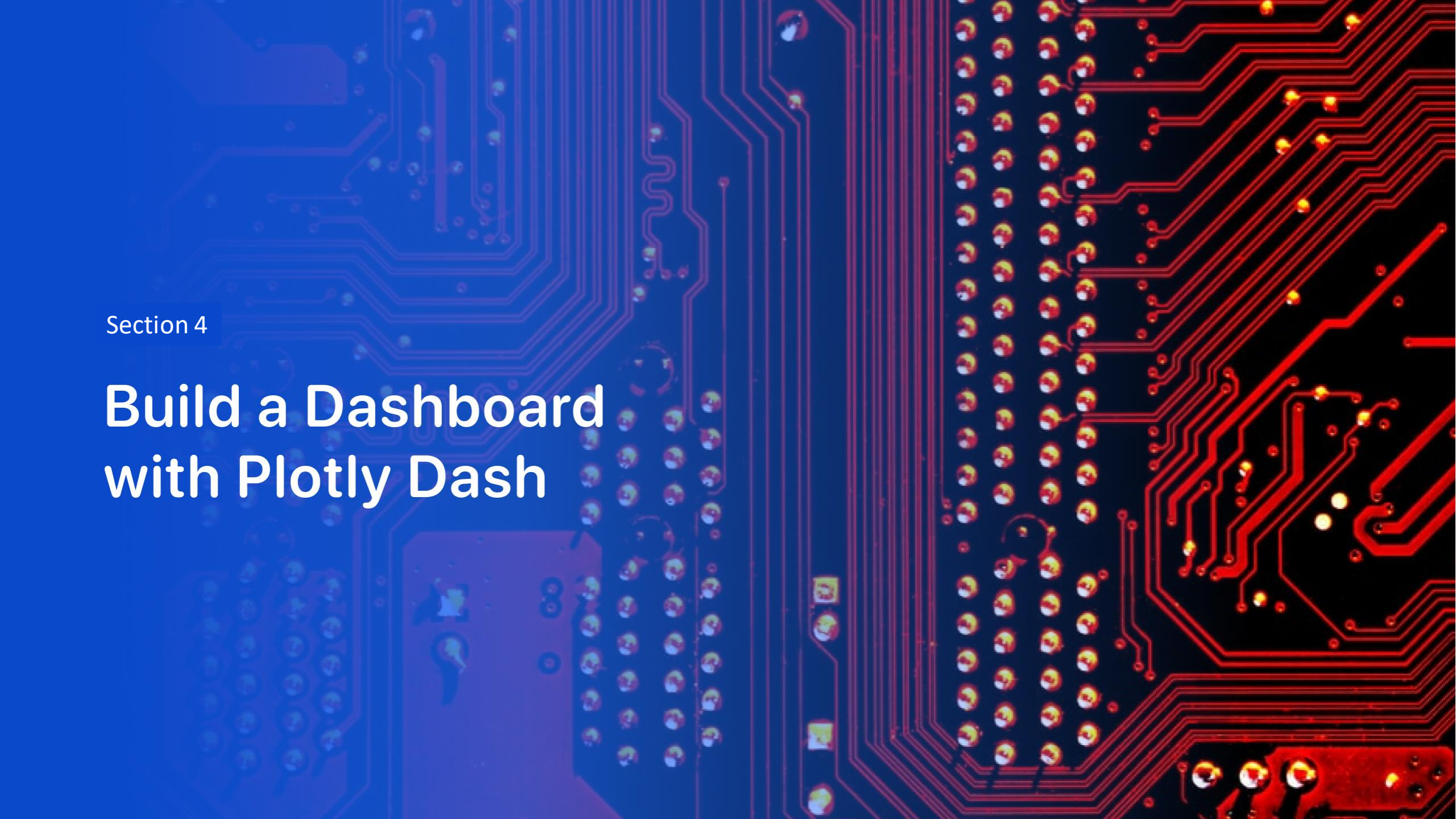
Distance to closest Railway



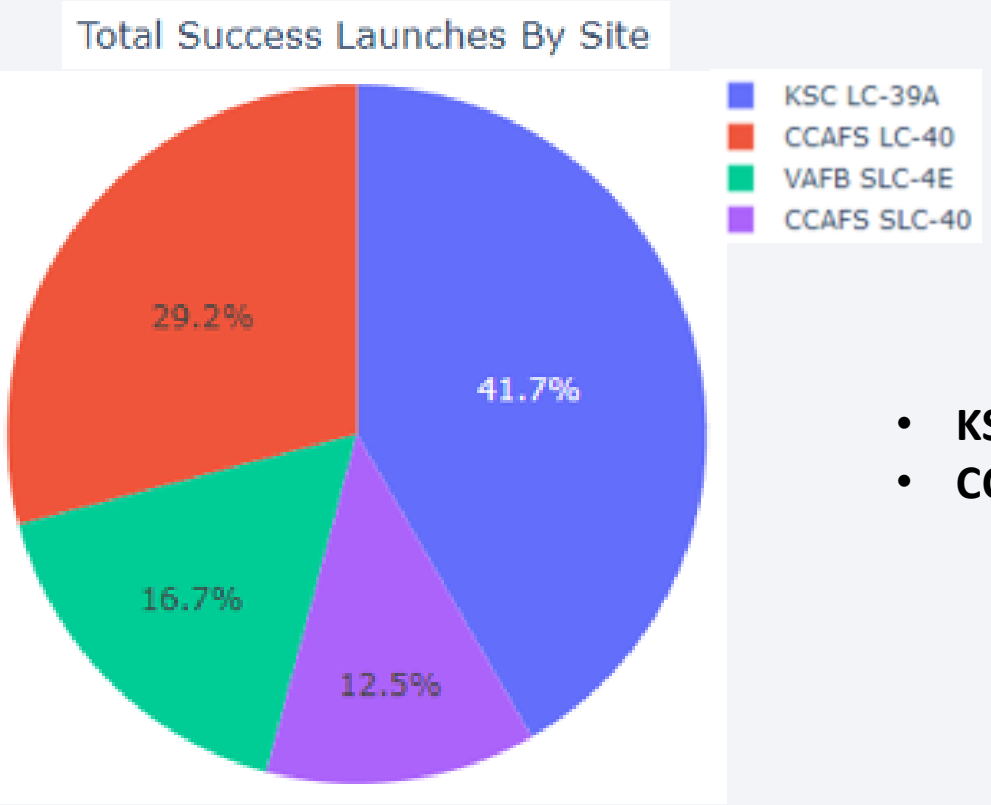


Distance to closest Highway and Coastline

Distance to closest City


Section 4
Build a Dashboard with Plotly Dash

Dashboard: Distribution of Successful Launches

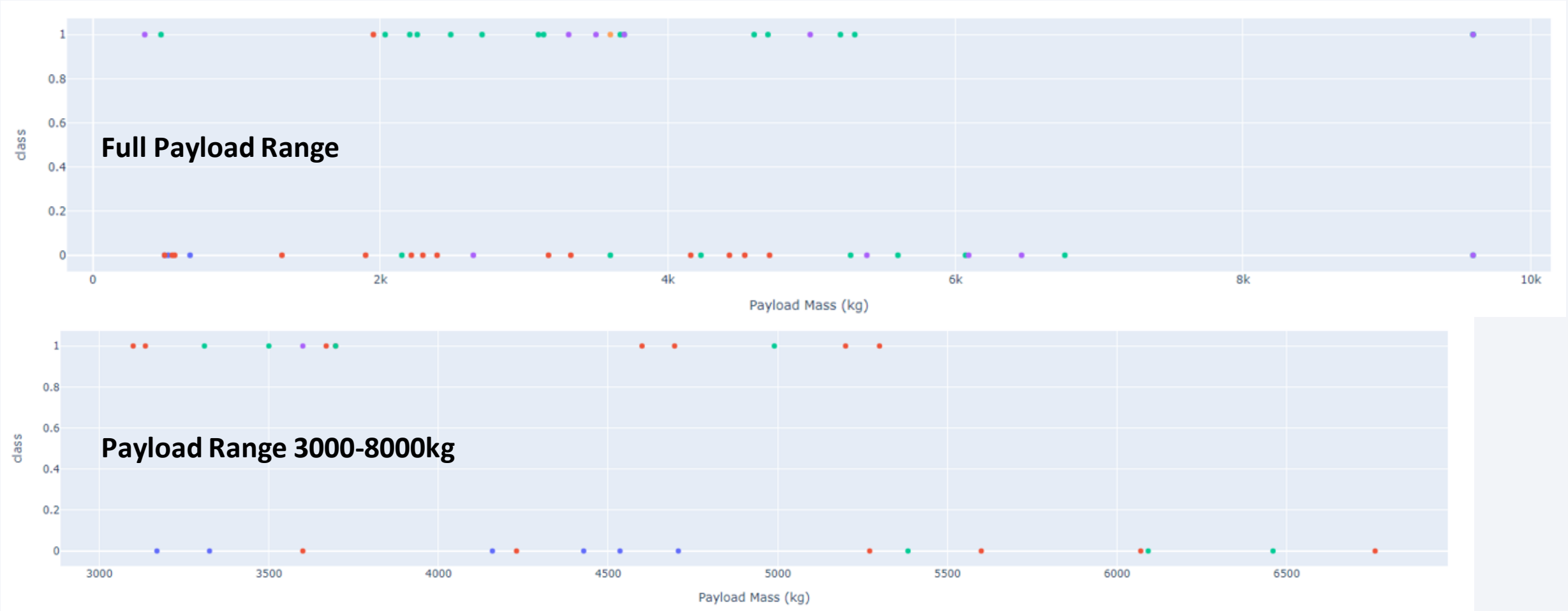


- **KSC LC-39A has the most successful launches**
- **CCAFS SLC-40 has the least successfull launches**

Dashboard: Most Successful Launch Site

- **KSC LC-39A has the highest success rate: 76.9%**

Dashboard: Payload vs. Launch Outcome



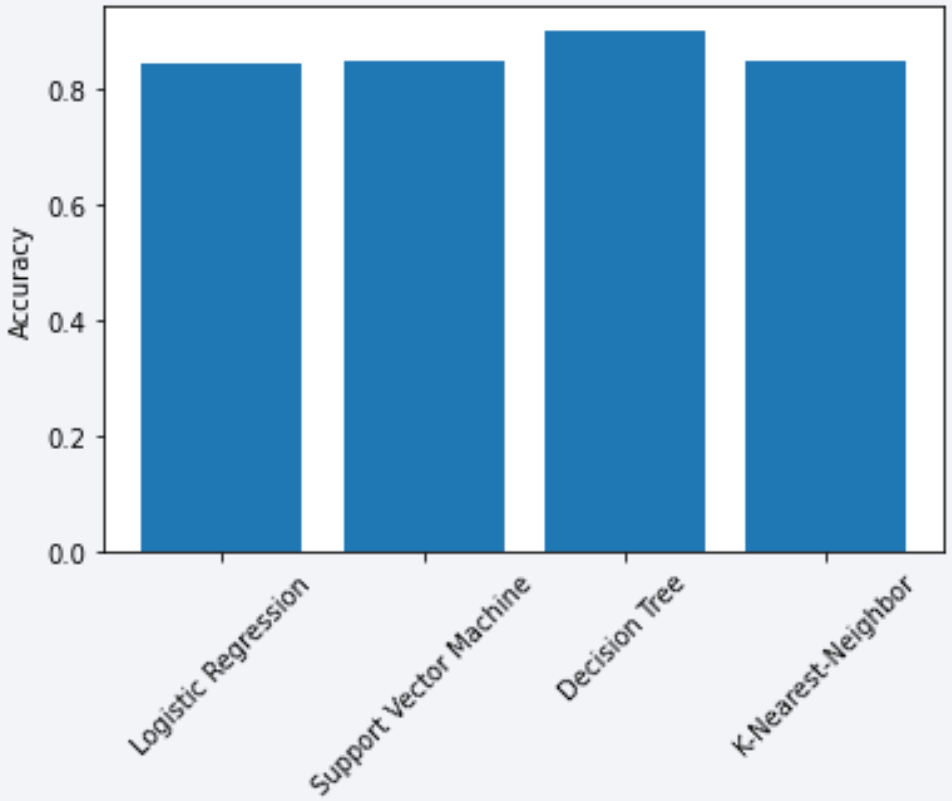
- **Success rate for lower payloads higher than heavy payloads**

Section 5

Predictive Analysis (Classification)

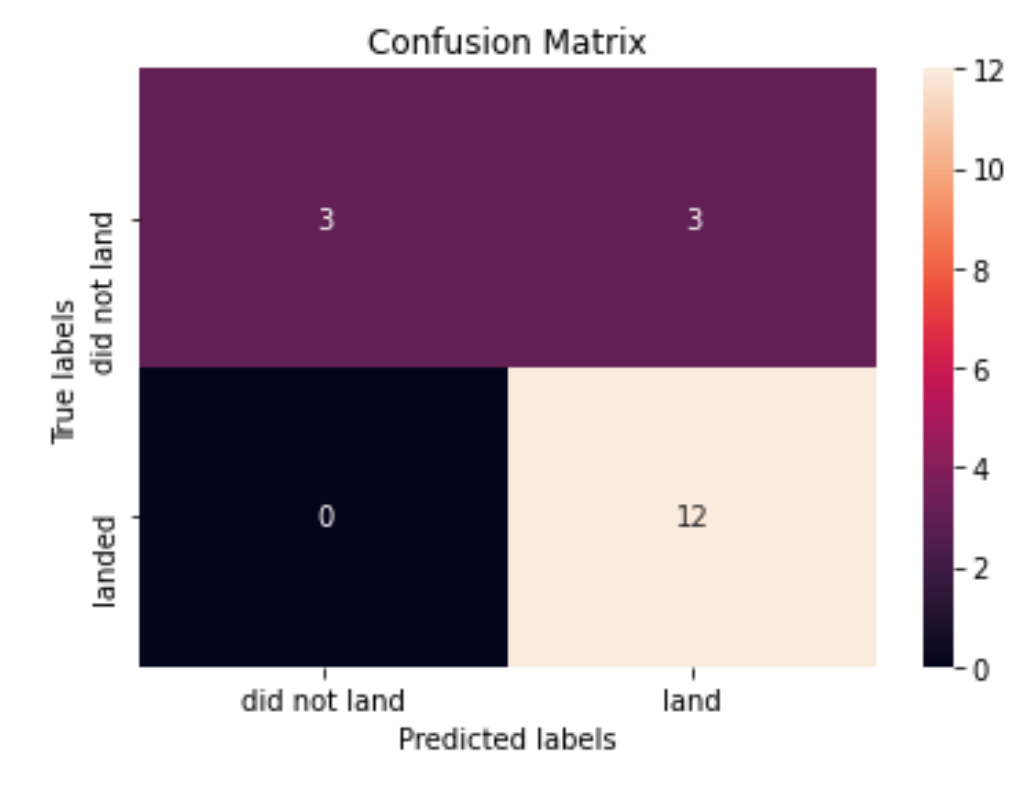
Classification Accuracy

```
Best Algorithm is Tree with a score of 0.9
Best Params is : {'criterion': 'gini', 'max_depth': 16, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 2, 'splitter': 'best'}
```



- **Decision Tree Algorithm has the best accuracy score: 0.9**

Confusion Matrix



	Actually Positive (1)	Actually Negative (0)
Predicted Positive (1)	True Positives (TPs)	False Positives (FPs)
Predicted Negative (0)	False Negatives (FNs)	True Negatives (TNs)

- **Decision Tree predicted correct except for 3 'False Positives'**

Conclusions

- The Decision Tree has proven to be the best algorithm to predict the landing outcome
- Lower weighted launches have a higher success rate
- KSC LC-39A is the Launch Site with the best success rate
- Success rate increases with number of flights (improve by experience)
- Orbits ES-L1, GEO, HEO and SSO have the best success rates

Thank you!