IBM Developer
SKILLS NETWORK

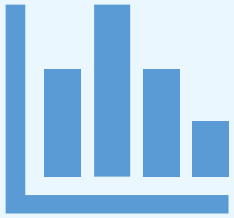
Winning Space Race with Data Science

Victoria Sintsova
28.02.2024

Outline

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

Executive Summary



Summary of methodologies

Data collection

Data wrangling

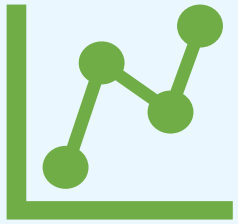
EDA with data visualization

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

- SpaceX advertises Falcon 9 rocket launches on its website at a cost of $62 million, while other providers have the price as high as $165 million. Much of the savings is due to the fact that SpaceX can reuse the first stage.
- Therefore, if predicting whether or not the Falcon 9 first stage will successfully land, it will make possible to determine the cost of the launch. This information can be used if another company wants to bid for a rocket launch 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

- Executive Summary
- Data collection methodology
- Perform data wrangling
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models

Data Collection

- To create the dataset, SpaceX launch data was collected using:
- SpaceX REST API (this API provides launch data including information on the rocket used, payload delivered, launch specifications, landing specifications, and landing outcome), The SpaceX REST API endpoints, or URLs, starts with api.spacexdata.com/v4/;
- Wikipedia page on the Falcon 9 launch, data was obtained using web scrapping (Python library BeautifulSoup) .

Data Collection - SpaceX API

Getting Response from API

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

Converting Response to a .json file

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

Apply custom functions to clean data

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

Assign list to dictionary then dataframe

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

Filter dataframe and export to flat file (.csv)

```
data_falcon9 = df[df['BoosterVersion']!='Falcon 1']
data_falcon9.to_csv('dataset_part_1.csv', index=False)
```

GitHub URL to Notebook

Data Collection – Web Scrapping

Getting Response from HTML

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

Creating BeautifulSoup Object

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

Finding tables

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

Getting column names

```
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
```

Creation of dictionary

```
launch_dict= dict.fromkeys(column_names)
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']=[]
```

Appending data to keys

```
extracted_row = 0
for table_number,table in enumerate(soup.find_all('table',"wikitable plainrowheaders collapsible")):
    for rows in table.find_all("tr"):
        if rows.th:
            ...
            ...
```

Converting dictionary to dataframe

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

Dataframe to .CSV

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

[GitHub URL to Notebook]

Data Wrangling

Exploratory data analysis (EDA) was performed to find some patterns in the data and determine what would be the label for training supervised models. In the dataset, there are several different cases of successful (TRUE) and unsuccessful (False) booster landed. The location of the landing is also indicated: ocean (Ocean), ground pad (RTLS), drone ship (ASDS).

These results are converted into training marks, where 1 means that the booster landed successfully, 0 - that the landing was unsuccessful.

The dataset also shows the specific orbits that each launch targeted (HEO, LEO, MEO, GEO).

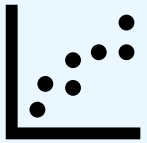
Process of Data Wrangling

- Perform Exploratory Data Analysis (EDA) on dataset
- Calculate the number of launches on each site
- Calculate the number and occurrence of each orbit
- Calculate the number and occurence of mission outcome of the orbits
- Create a landing outcome label from Outcome column
- Export dataset as .CSV

GitHub URL to Notebook

EDA with Data Visualization

Scatter Graph



Show how much one variable is affected by another

Flight Number VS Payload Mass

Flight Number VS Launch Site

Payload VS Launch Site

Orbit VS Flight Number

Payload VS Orbit Type

Bar Graph

Makes it easy to compare sets of data between different groups at a glance

Show big changes in data over time

Success Rate VS Orbit

Line Graph

Show data variables and trends very clearly and can help to make predictions about the results of data not yet recorded

Success Rate VS Year

GitHub URL to Notebook

EDA with SQL

SQl queries were used to gather information about the dataset.
Example of some questions about the dataset and SQL queries to get answers:

- Displaying the names of the unique launch sites in the space mission
  - %sql SELECT DISTINCT Launch_Site from SPACEXTABLE;
- Displaying the total payload mass carried by boosters launched by NASA (CRS)
  - %sql select sum(PAYLOAD_MASS__KG_) as payloadmass from SPACEXTABLE;
- Displaying average payload mass carried by booster version F9 v1.1
  - %sql select avg(PAYLOAD_MASS__KG_) as payloadmass from SPACEXTABLE;
- Listing the total number of successful and failure mission outcomes
  - %sql select count(MISSION_OUTCOME) as missionoutcomes from SPACEXTABLE GROUP BY MISSION_OUTCOME;
- Ranking the count of successful landing_outcomes between the date 2010-06-04 and 2017-03-20 in descending order
  - %sql SELECT Landing_Outcome FROM SPACEXTABLE WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' ORDER BY DATE DESC;

GitHub URL to Notebook

Building an Interactive Map with Folium

Visualization the Launch Data into an interactive map.
For each pair of Latitude and Longitude Coordinates at each launch site added a Circle Marker around each launch site with a label of the name of the launch site.
Assigned the dataframe launch_outcomes(failures, successes) to classes 0 and 1 with Green and Red markers on the map in a MarkerCluster().
Using Haversine's formula calculated the distance from the Launch Site to various landmarks to find various trends about what is around the Launch Site to measure patterns. Lines are drawn on the map to measure distance to landmarks.
Example of some trends in which the Launch Site is situated in:
•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

GitHub URL to Notebook

Build a Dashboard with Plotly Dash

Used Python Anywhere to host the website live 24/7.

The dashboard is built with Flask and Dash web framework.

Graphs:

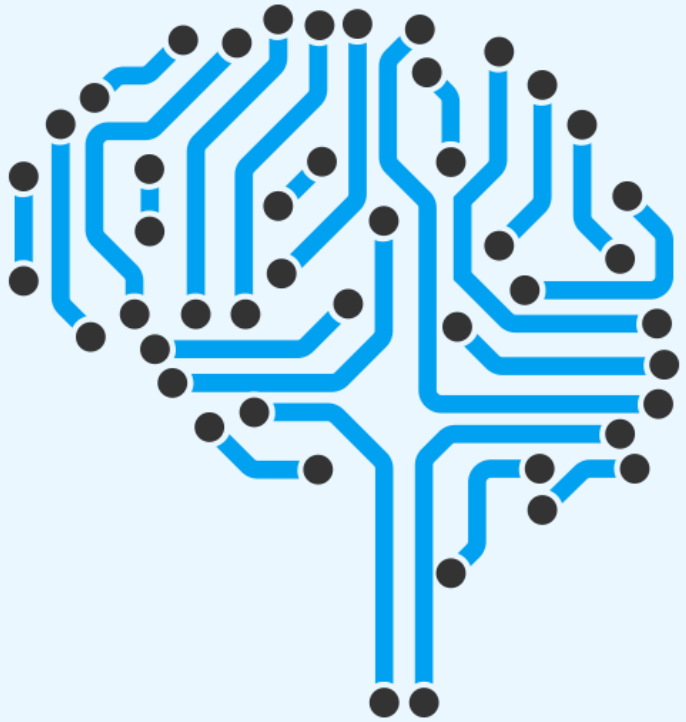
- Pie Chart showing the total launches by a certain site/all sites:
  - display relative proportions of multiple classes of data
  - size of the circle can be made proportional to the total quantity it represents
- Scatter Graph showing the relationship with Outcome and Payload Mass (Kg) for the different Booster Versions:
  - it shows the relationship between two variables
  - it is the best method to show you a non-linear pattern
  - the range of data flow, i.e. maximum and minimum value, can be determined
  - observation and reading are straightforward

GitHub URL to Notebook

Predictive Analysis (Classification)

BUILDING MODEL

- Load dataset into NumPy and Pandas
- Transform Data
- Split data into training and test data sets
- Decide which type of machine learning algorithms will be used
- Set parameters and algorithms to GridSearchCV
- Fit datasets into the GridSearchCV objects and train dataset.



EVALUATING MODEL

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

IMPROVING MODEL

- Feature Engineering
- Algorithm Tuning

FINDING 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

Results

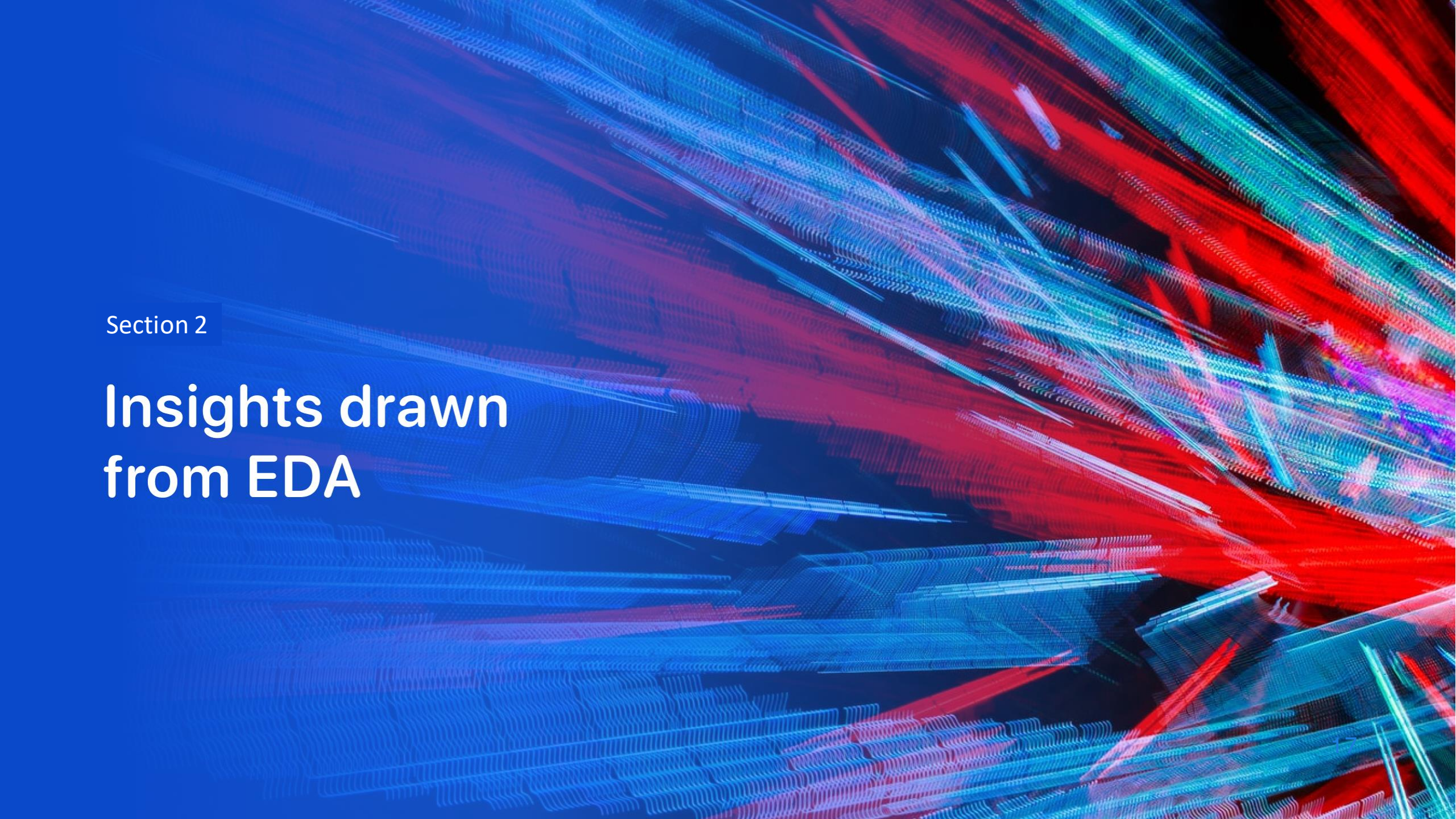
EXPLORATORY DATA ANALYSIS RESULTS

INTERACTIVE ANALYTICS DEMO IN SCREENSHOTS

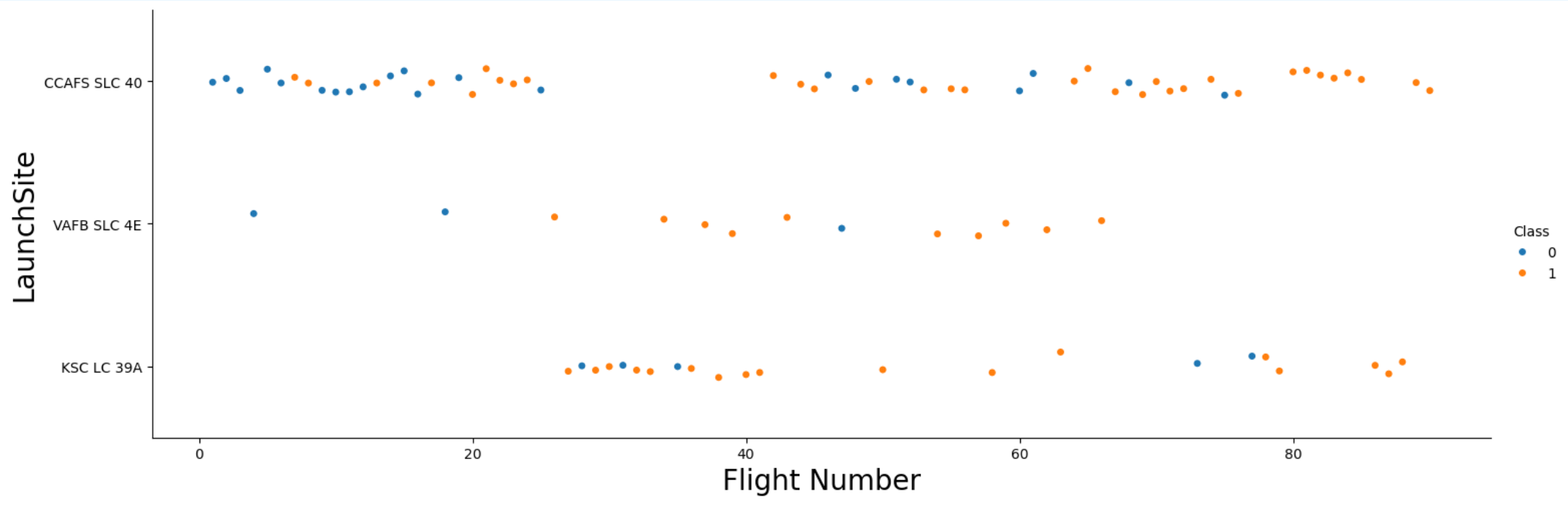
PREDICTIVE ANALYSIS RESULTS

Section 2

Insights drawn from EDA

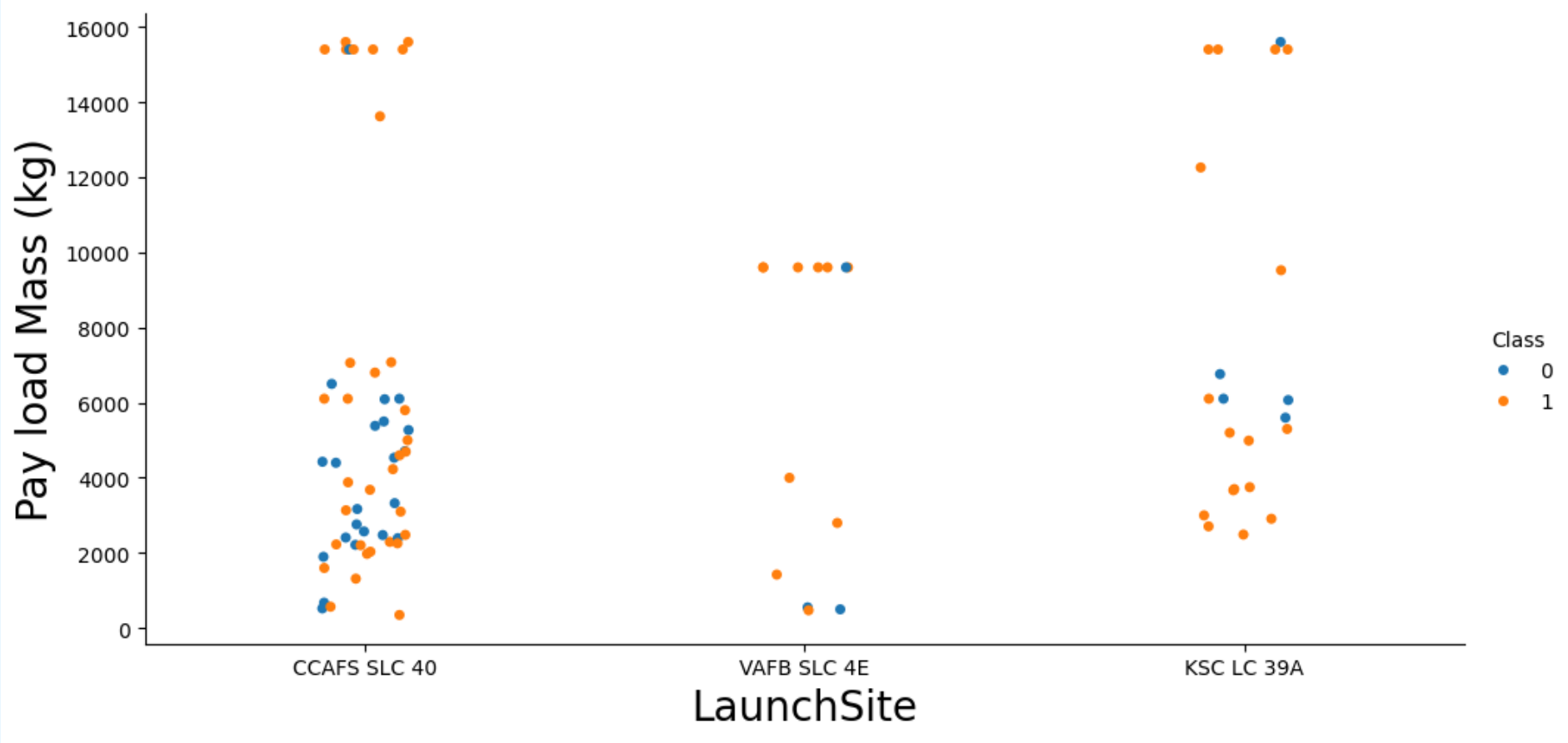


Flight Number vs Launch Site



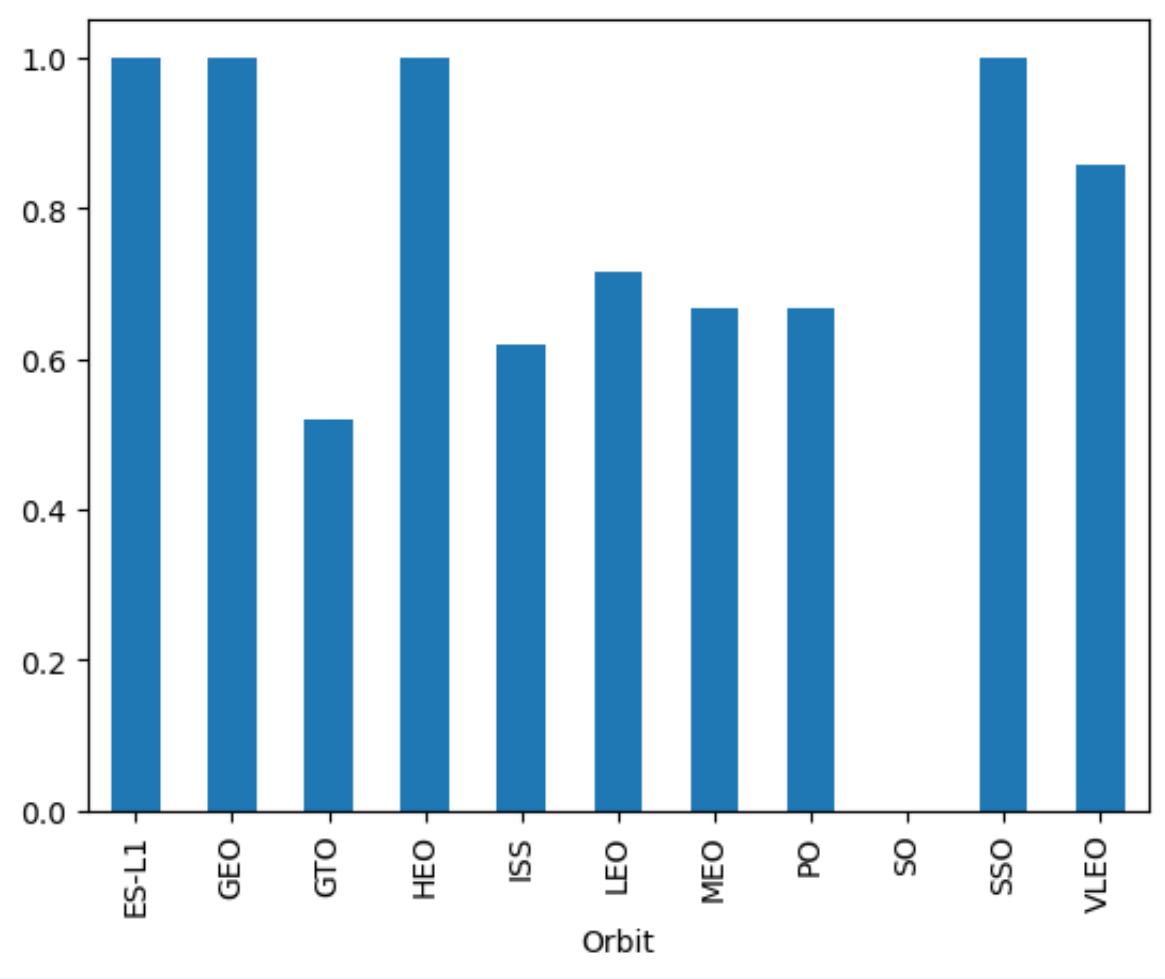
The more amount of flights at a launch site the greater the success rate at a launch site.

Payload vs. Launch Site



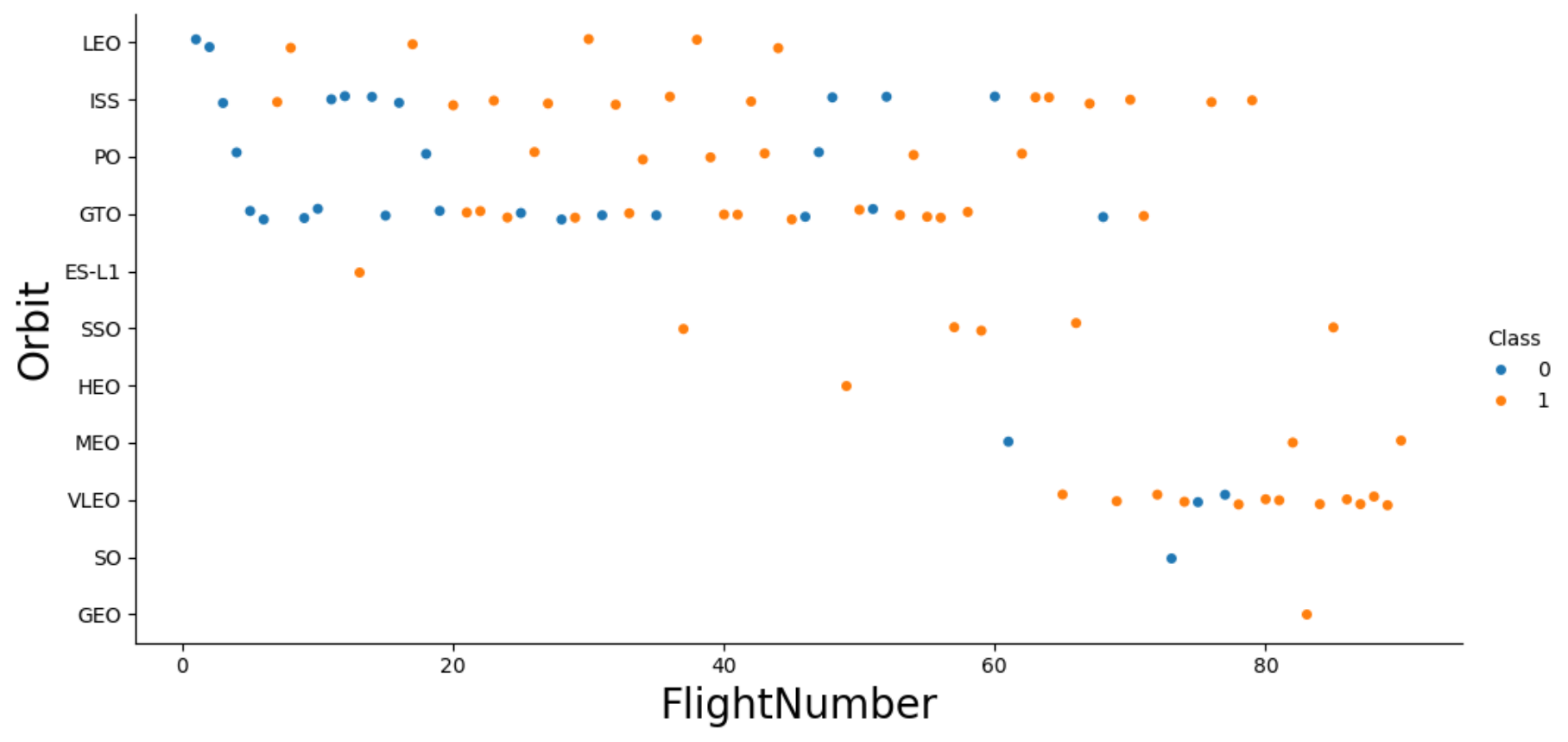
The greater the payload mass for Launch Site CCAFS SLC 40 the higher the success rate for the Rocket. There is not quite a clear pattern to be found using this visualization to make a decision if the Launch Site is dependent on Pay Load Mass for a success launch.

Success Rate vs. Orbit Type



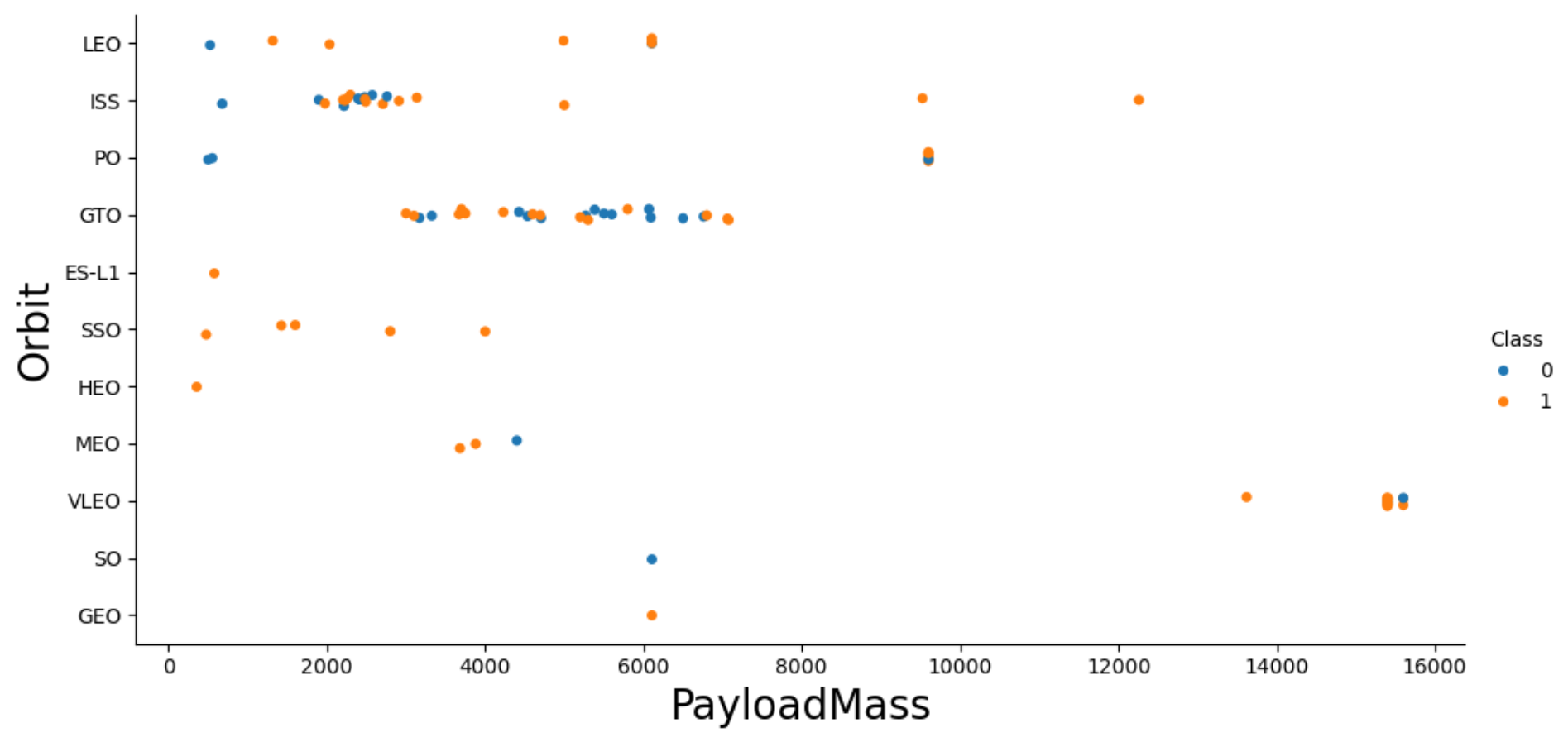
Orbit GEO,HEO,SSO,ES-L1 has the best Success Rate.

Flight Number vs. Orbit Type



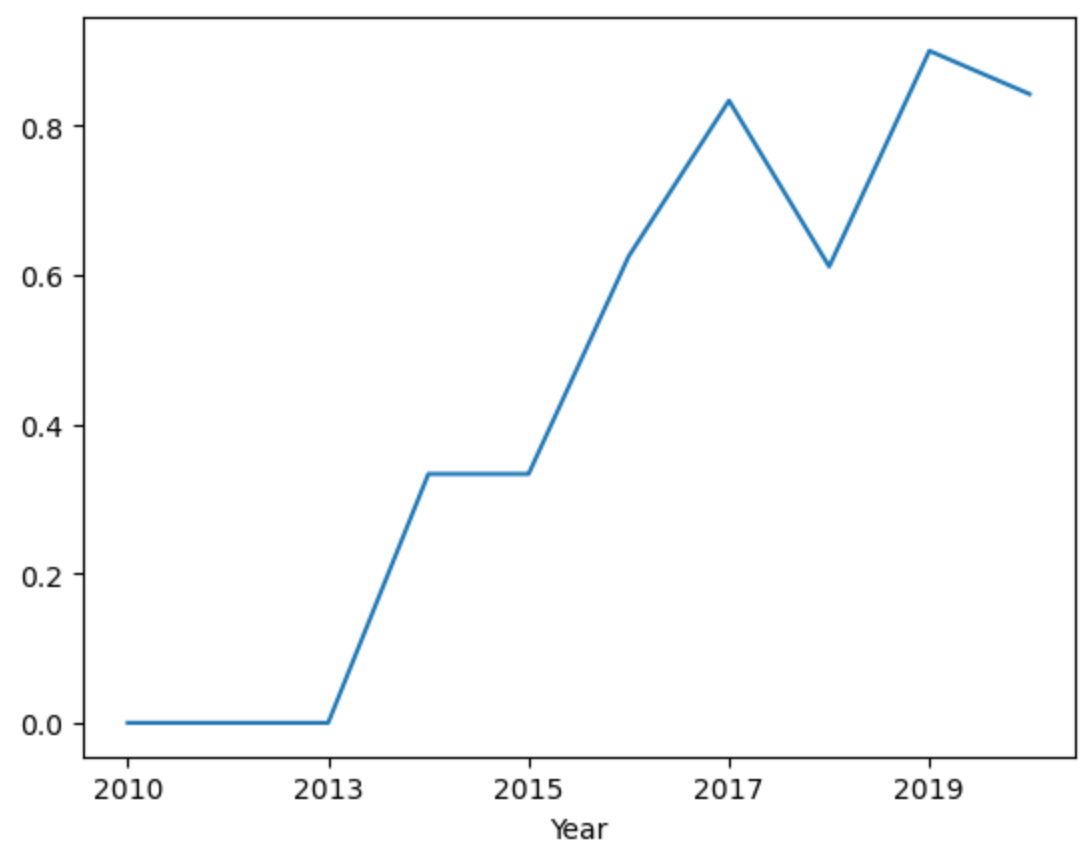
In the LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.

Payload vs. Orbit Type



With heavy payloads the successful landing or positive landing rate are more for PO,LEO and ISS. However for GTO we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission) are both there here.

Launch Success Yearly Trend



From the graph may observe that the success rate from 2013 kept to increase until 2020.

All Launch Site Names

- Assignment task: find the names of the unique launch sites
- SQL query: %sql SELECT DISTINCT Launch_Site from SPACEXTABLE;
- Query explanation: using the word **DISTINCT** in the query means that it will only show Unique values in the **Launch_Site** column from **SPACEXTABLE**
- Result of SQL query:

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

Launch Site Names Begin with 'CCA'

•Assignment task: find 5 records where launch sites begin with 'CCA'

•SQL query: %sql SELECT * from SPACEXTABLE where (LAUNCH_SITE) LIKE 'CCA%' LIMIT 5;

•Query explanation: using the word **LIMIT 5** in the query means that it will only show 5 records from **SPACEXTABLE** and **LIKE** keyword has a wild card with the words **'CCA%'** the percentage in the end suggests that the **Launch_Site** name must start with **CCA**

•Result of SQL query:

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

Total Payload Mass

•Assignment task: calculate the total payload carried by boosters from NASA

•**SQL query:** %sql select sum(PAYLOAD_MASS__KG_) as payloadmass from SPACEXTABLE where Customer = 'NASA (CRS)';

•Query explanation: using the function **SUM** summates the total in the column **PAYLOAD_MASS_KG_**. The **WHERE** clause filters the dataset to only perform calculations on **Customer NASA (CRS)**

•Result of SQL query:

payloadmass
45596

Average Payload Mass by F9 v1.1

- Assignment task: calculate the average payload mass carried by booster version F9 v1.1
- SQL query: %sql select avg(PAYLOAD_MASS__KG_) as payloadmass from SPACEXTABLE where Booster_Version = 'F9 v1.1';
- Query explanation: using the function **AVG** works out the average in the column **PAYLOAD_MASS_KG_** . The **WHERE** clause filters the dataset to only perform calculations on **Booster_version F9 v1.1**
- Result of SQL query:

payloadmass
2928.4

First Successful Ground Landing Date

•Assignment task: find the dates of the first successful landing outcomes on ground pad

•SQL query: %sql select min(DATE) from SPACEXTABLE where Landing_Outcome = 'Success (drone ship)';

•Query explanation: using the function **MIN** works out the minimum date in the column **Date**. The **WHERE** clause filters the dataset to only perform calculations on **Landing_Outcome Success (drone ship)**

•Result of SQL query:

min(DATE)
2016-04-08

Successful Drone Ship Landing with Payload between 4000 and 6000

•Assignment task: list the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000

•SQL query: %sql select BOOSTER_VERSION from SPACEXTABLE where Landing_Outcome='Success (drone ship)' and PAYLOAD_MASS__KG_ BETWEEN 4000 and 6000;

•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**

•Result of SQL query:

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

Total Number of Successful and Failure Mission Outcomes

•Assignment task: calculate the total number of successful and failure mission outcomes

•SQL query: %sql select(select count(MISSION_OUTCOME) from SPACEXTABLE where MISSION_OUTCOME LIKE '%Success%') as Successful,(SELECT Count(MISSION_OUTCOME) from SPACEXTABLE where MISSION_OUTCOME LIKE '%Failure%') as Failure;

•Query explanation: used subqueries to produce the results. The **LIKE ‘%foo%’** wildcard shows that in the record the foo phrase is in any part of the string in the records for example. **PHRASE '(Drone Ship was a Success)' LIKE ‘%Success%’** Word ‘Success’ is in the phrase the filter will include it in the dataset

•Result of SQL query:

Successful	Failure
100	1

Boosters Carried Maximum Payload

•Assignment task: list the names of the booster which have carried the maximum payload mass

•SQL query: %sql select DISTINCT BOOSTER_VERSION as boosterversion from SPACEXTABLE where PAYLOAD_MASS__KG_=(select max(PAYLOAD_MASS__KG_) from SPACEXTABLE);

•Query explanation: using the word **DISTINCT** in the query means that it will only show Unique values in the **Booster_Version** column from **SPACEXTABLE**.

•Result of SQL query:

boosterversion
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

•Assignment task: list the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

•**SQL query:** %sql SELECT substr(Date, 6,2) as month, DATE,BOOSTER_VERSION, LAUNCH_SITE, Landing_Outcome FROM SPACEXTABLE where Landing_Outcome = 'Failure (drone ship)' and substr(Date,0,5)='2015';

•Query explanation: the **substr(Date, 6,2)** function returns month, the **substr(Date,0,5)** function returns year. **WHERE** clause filters year to be **2015** and **Landing_Outcome = 'Failure (drone ship)'**

•Result of SQL query:

month	Date	Booster_Version	Launch_Site	Landing_Outcome
01	2015-01-10	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
04	2015-04-14	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

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

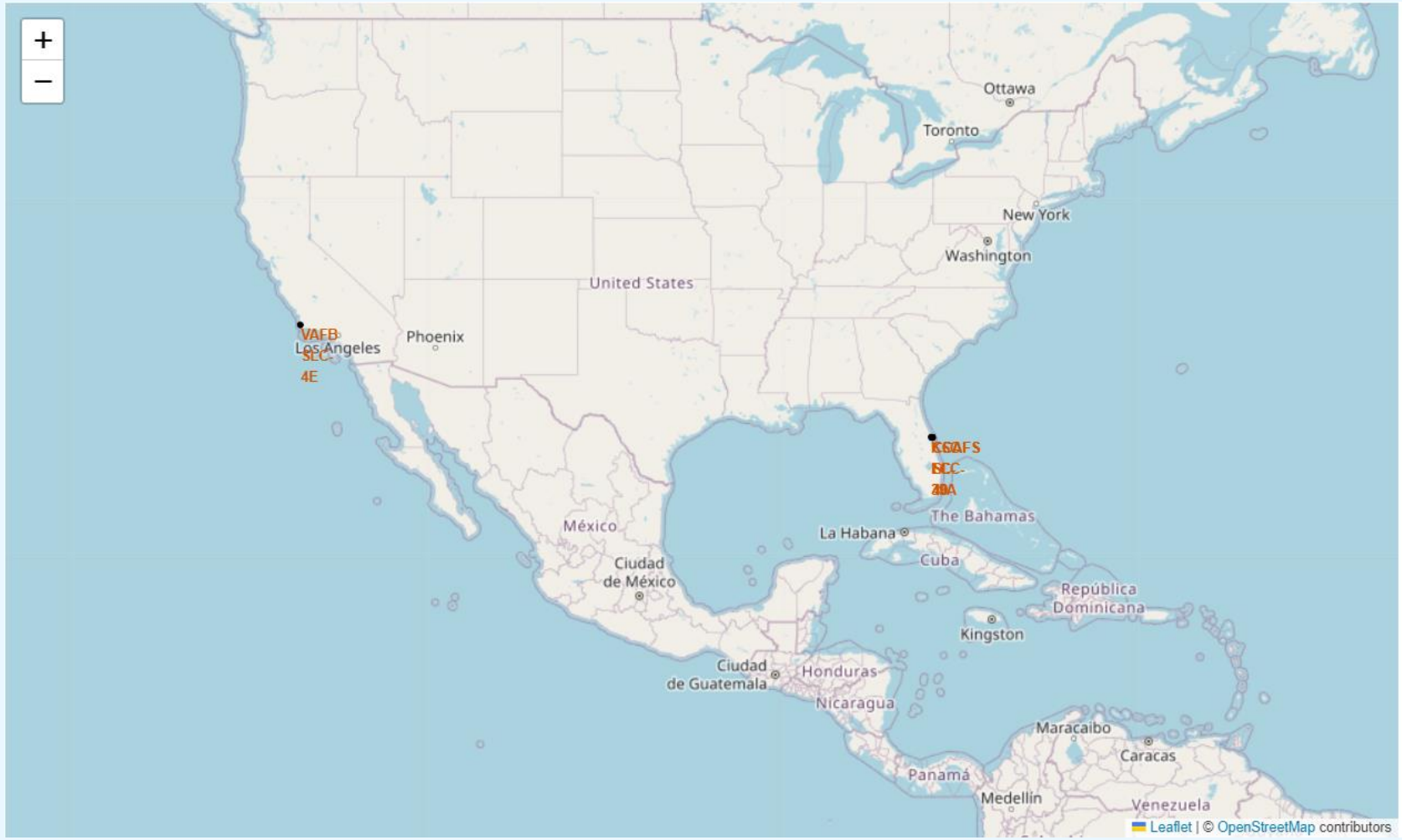
- Assignment task: rank the count of landing outcomes between the date 2010-06-04 and 2017-03-20 in descending order
- SQL query: %sql SELECT Landing_Outcome, count(Landing_Outcome) as amount FROM SPACEXTABLE WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' Group by Landing_Outcome ORDER BY amount DESC;
- Query explanation: function **COUNT** counts records in column. **WHERE** filters data
- Result of SQL query:

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

Section 3

Launch Sites Proximities Analysis

All launch sites global map markers

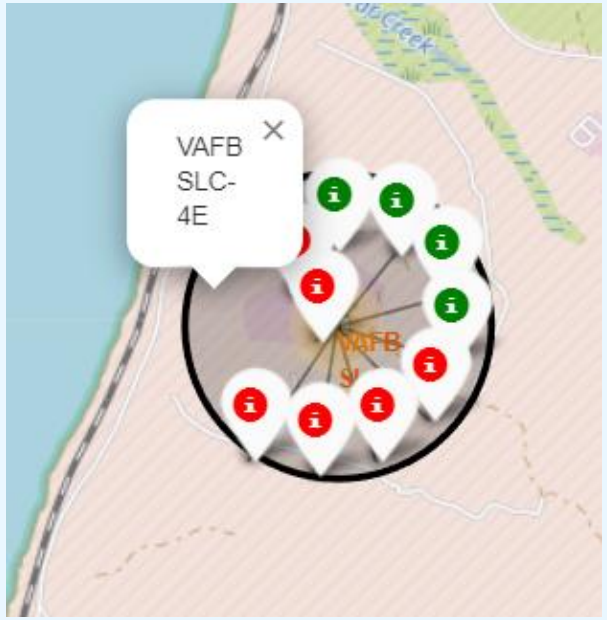


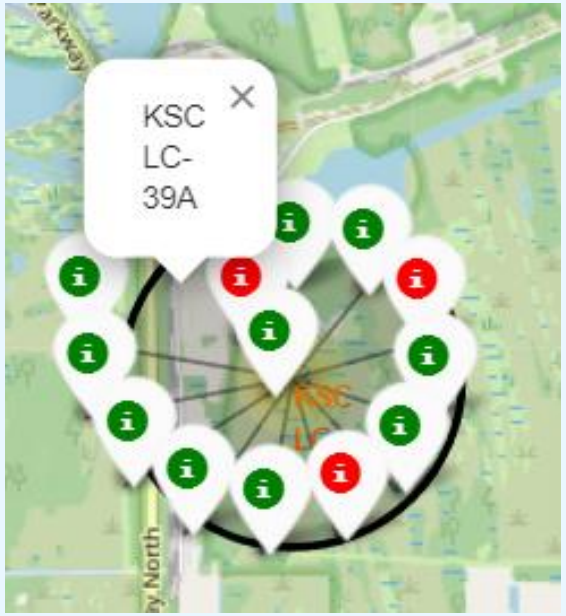
The launch sites are in proximity to the equator and the coast.

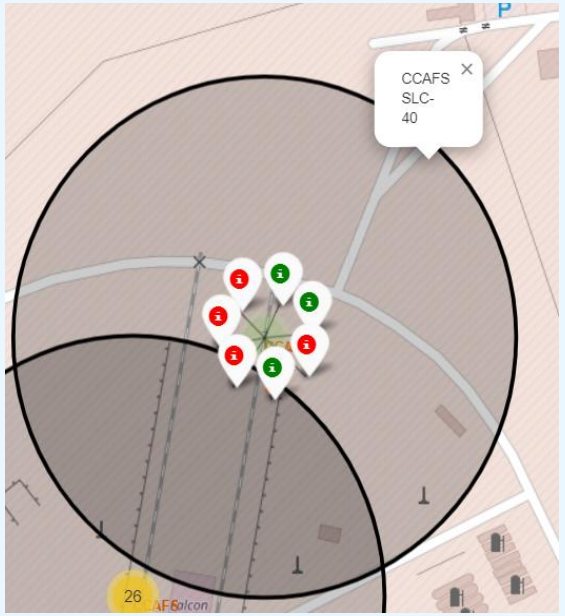
This makes sense as it takes less fuel to get into space from the equator due to the physics of Earth's rotation.

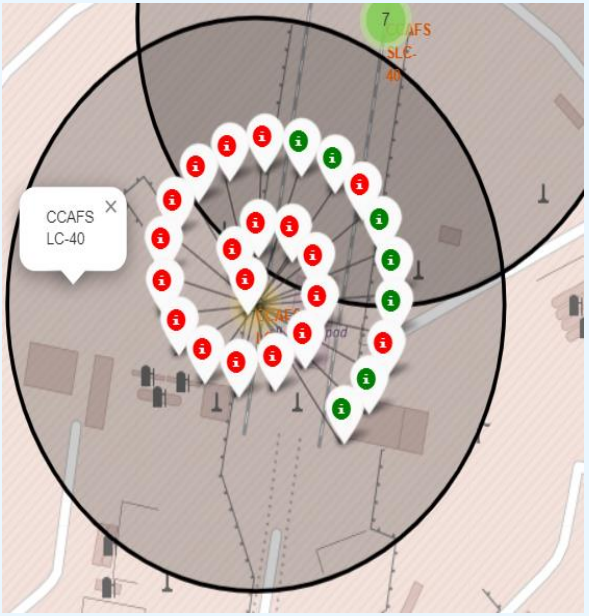
The launch sites in close proximity to the coast are also logical for safety reasons.

Colour Labelled Markers









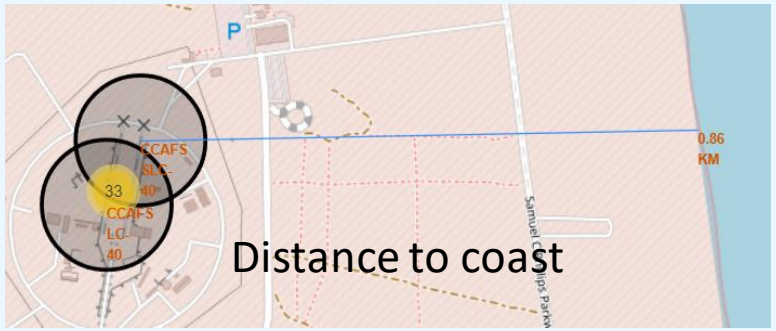
California Launch Site

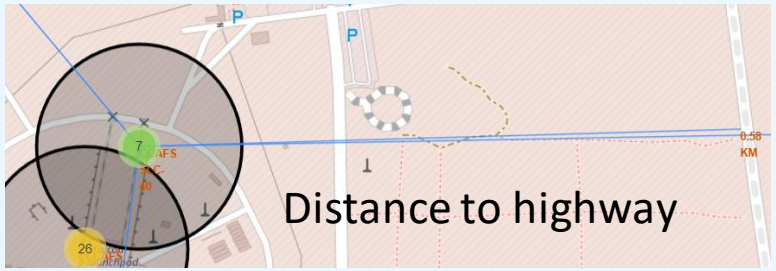
Florida Launch Sites

Green Marker shows successful Launches and Red Marker shows Failure.

From the color-labeled markers in marker clusters, easily identify which launch sites have relatively high success rates.

Calculation of the distances between a launch site to its proximities





The launch sites are in close proximity to equator to minimize fuel consumption by using Earth's ~ 30km/sec eastward spin to help spaceships get into orbit.

Launch sites are in close proximity to coastline (~0.86 km) so they can fly over the ocean during launch, for at least two safety reasons: crew has option to abort launch and attempt water landing minimize people and property at risk from falling debris.

Launch sites are in close proximity to highways (~0.58 km), which allows for easily transport required people and property.

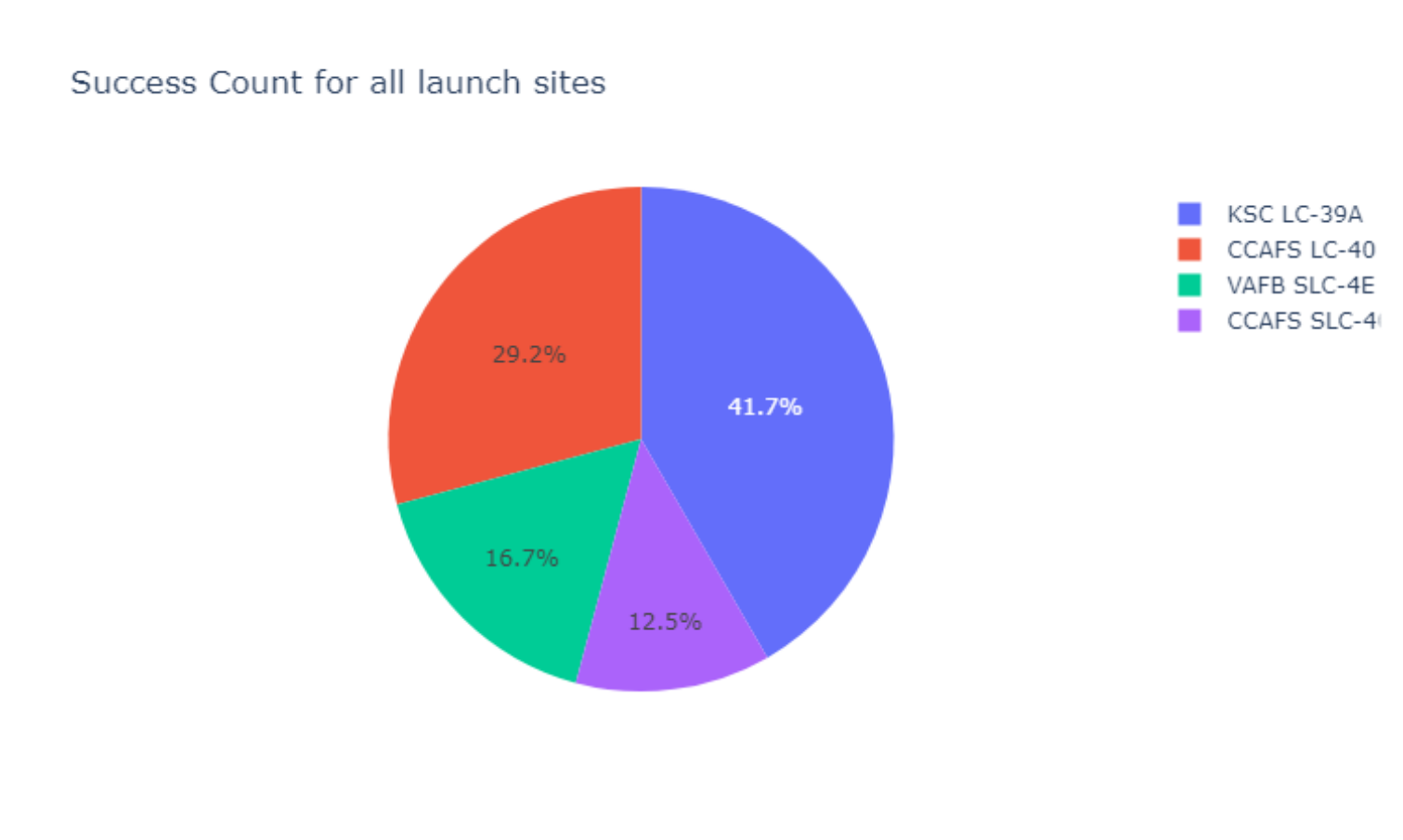
Launch sites are in close proximity to railways (~1.28 km), which allows transport for heavy cargo.

Launch sites are not in close proximity to cities (Cape Canayeral ~18.2 km), which minimizes danger to population dense areas.

Section 4

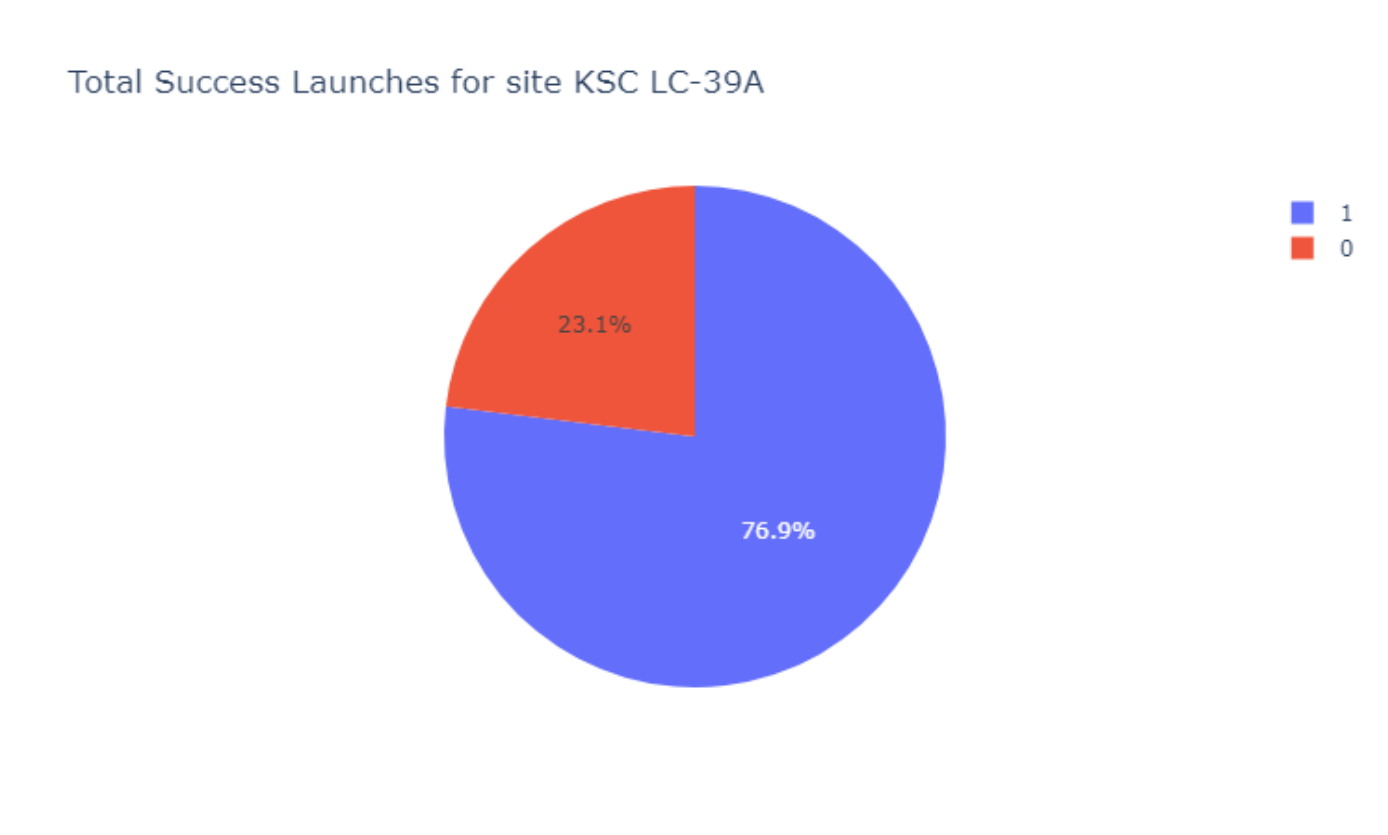
Build a Dashboard with Plotly Dash

The success percentage achieved by each launch site (pie chart)



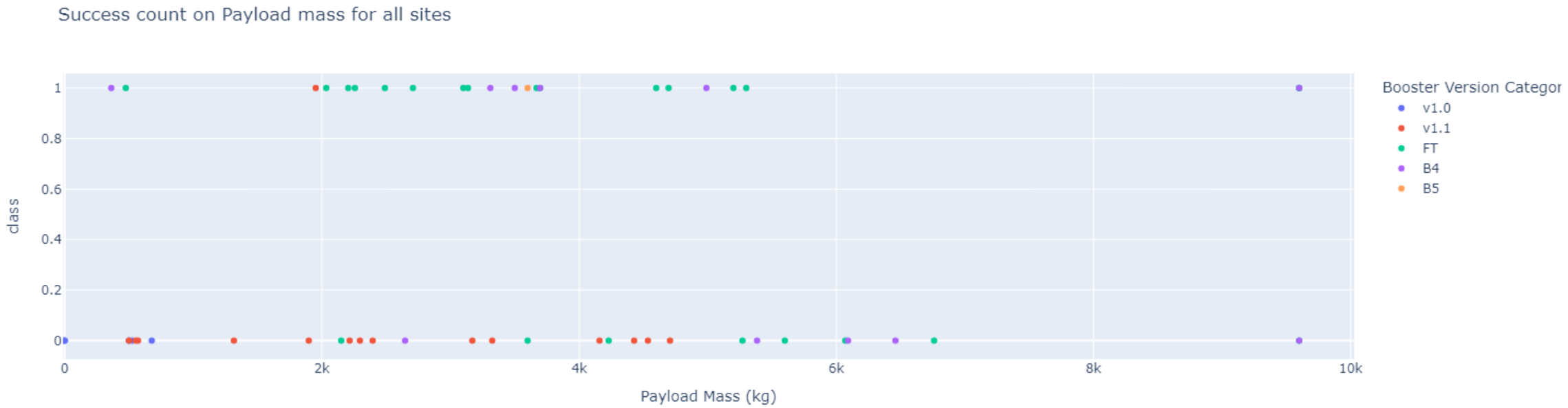
The graph shows that the KSC LC-39A had the most successful launches from all the sites

Pie chart for the launch site with highest launch success ratio



KSC LC-39A achieved a 76.9% success rate, while getting a 26.1% failure rate

Payload vs Launch Outcome scatter plot for all sites with different payload

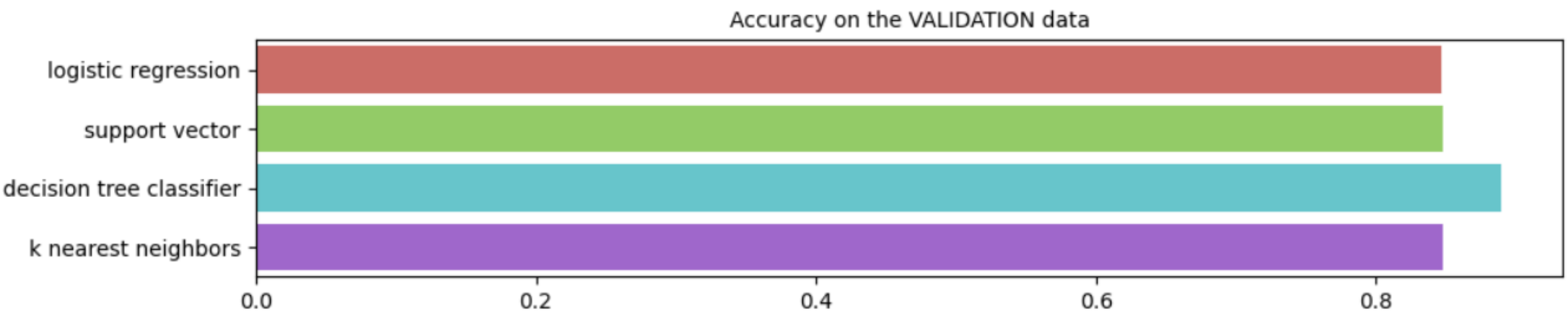


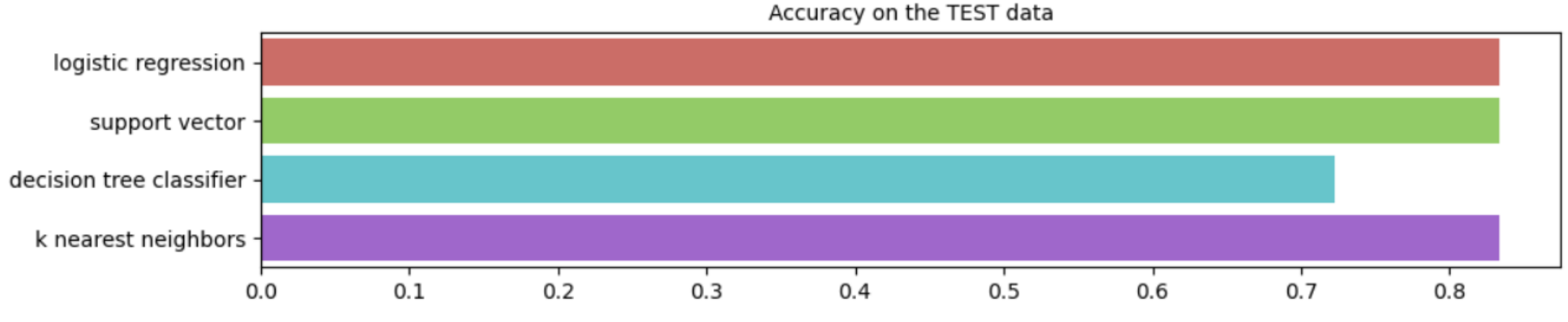
The success rates for low weighted payloads (<5600kg) is higher than the heavy weighted payloads.
F9 Booster version FT has the highest launch success rate.
F9 Booster version v1.1 has the lowest launch success rate.

Section 5

Predictive Analysis (Classification)

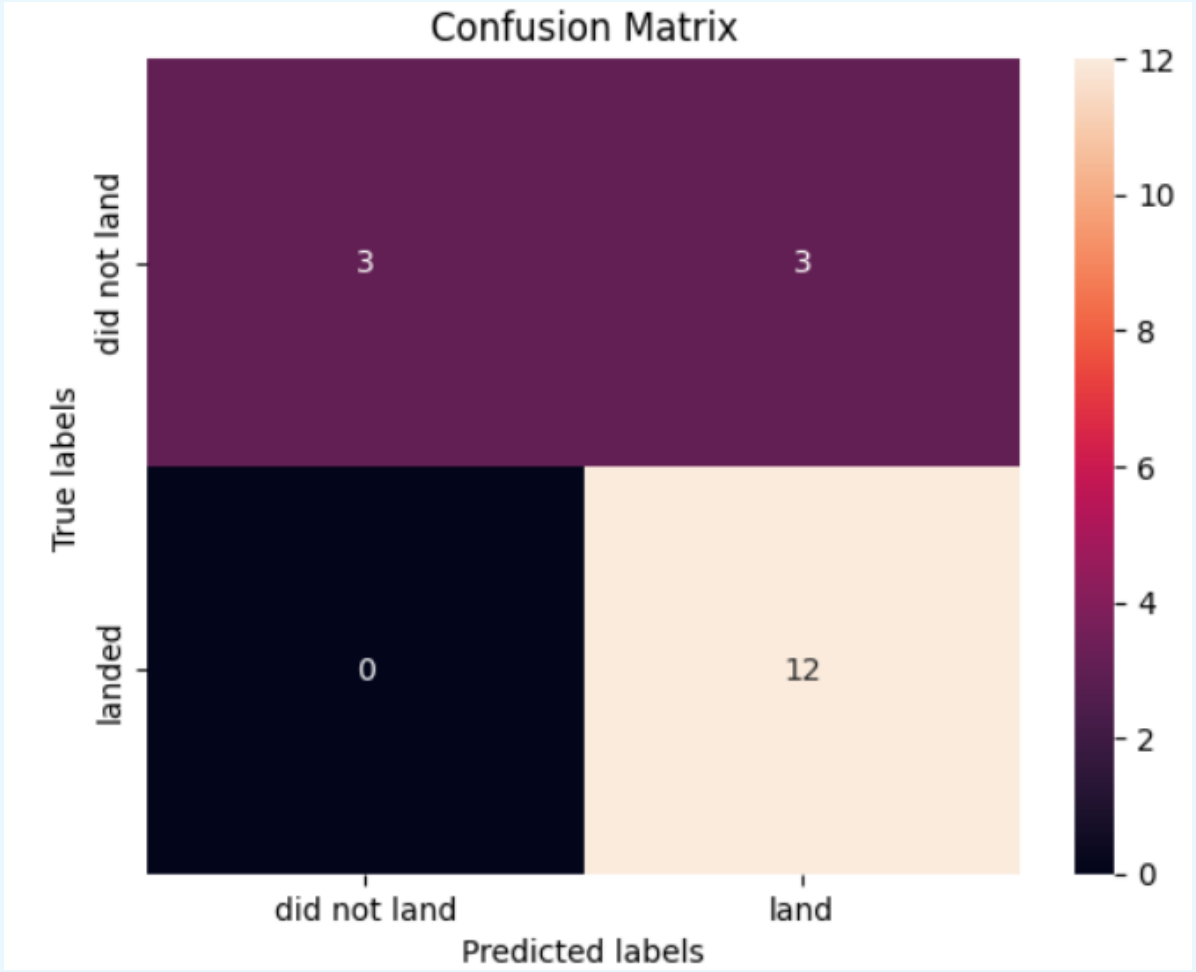
Classification Accuracy





After comparing accuracy of above methods, they all preformed practically the same, except for tree which fit train data slightly better but test data worse

Confusion Matrix



The confusion matrices of the best performing models (4-way-tie) are the same

The major problem is false positives as evidenced by the models incorrectly predicting the 1st stage booster to land in 3 out of 18 samples in the test set

Conclusions

Low weighted payloads perform better than the heavier payloads

The success rates for SpaceX launches is directly proportional time in years they will eventually perfect the launches

We can see that KSC LC-39A had the most successful launches from all the sites

Orbit GEO,HEO,SSO,ES-L1 has the best Success Rate

Machine learning models need to be refined to improve prediction accuracy

Appendix

References

- https://aviationweek.com/defense-space/space/podcast-interview-spacexs-elon-musk - with Elon Musk where he discloses the 1st stage booster to cost upwards of $15 million
- https://datascience.stackexchange.com/a/33050 - Explanation of why you would rebuild your model using the full dataset
- https://www.spacex.com/vehicles/falcon-9/ - Source of SpaceX's advertised $62 million launch price

Thank you!