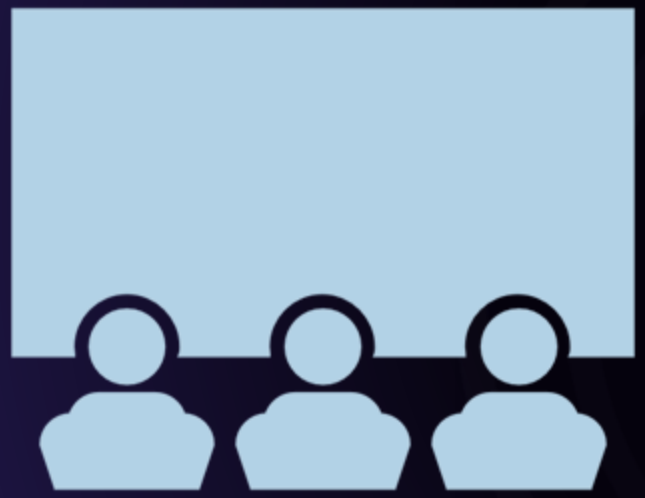
Data Science in Space Exploration

Anjali Sammeta

OUTLINE



- Executive Summary
- Introduction
- Methodology
- Results
  - Visualization – Charts
  - Dashboard
- Discussion
  - Findings & Implications
- Conclusion
- Appendix

EXECUTIVE SUMMARY

- Summary of Methodologies
  - Data Collection via API and Web Scraping
  - Data Wrangling
  - Exploratory Data Analysis with Data Visualization
  - Exploratory Data Analysis with SQL
  - Interactive Map with Folium
  - Dashboards with Plotly Dash
  - Predictive Analysis
- Summary of Results
  - Exploratory Data Analysis results
  - Interactive Maps and Dashboard
  - Predictive Results

INTRODUCTION

Project Background and Context

SpaceX is the most successful company of the commercial space age, making space travel more affordable. The company advertises Falcon 9 rocket on its website to cost $62 million; other providers cost upward of $196 million each. The aim of this project is to predict if the Falcon 9 first stage will successfully land and determine the cost of a launch.

Questions to be Answered

- What are the main characteristics of a successful or failed landing?
- How do variables such as payload mass, launch site, number of flights, and orbits affect the success of the first stage landing?
- What are the conditions which will allow SpaceX to achieve the best landing success rate?

METHODOLOGIES

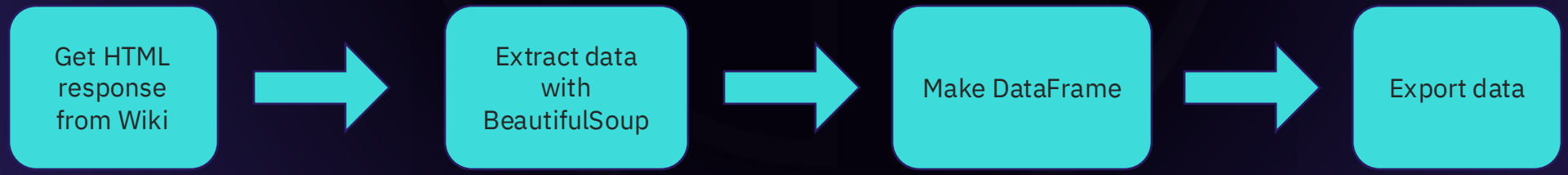
METHODOLOGY

- Data Collection:
  - SpaceX Rest API
  - Web Scrapping from Wikipedia
- Data Wrangling:
  - Dropping unnecessary columns
  - One hot Encoding for classification models
- Exploratory Data Analysis (EDA) using Visualization and SQL
- Interactive Visual Analystics using Folium and Plotly Dash
- Predictive Analysis using Classification Models
  - Building, tuning, and evaluating classification models to ensure the best results

DATA COLLECTION

- Datasets are collected from REST SpaceX API and Web Scrapping Wikipedia
  - The information obtained by the API are rocket, launches, and payload information.
    - The SpaceX REST API URL is api.spacexdata.com/v4/

- The information optained by the webscrapping of Wikipedia are launches, landing, and payload information
  - The URL is https://en.Wikipedia.org/w/index/php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922



DATA COLLECTION – SPACEX API

1. Getting Response from API

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

response = requests.get(spacex_url)
```

2. Convert Response to JSON file

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

3. Transform data

```
# Call getBoosterVersion
getBoosterVersion(data)

# Call getLaunchSite
getLaunchSite(data)

# Call getPayloadData
getPayloadData(data)

# Call getCoreData
getCoreData(data)
```

4. Create a dictionary with data

```
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. Create DataFrame

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

6. Filter DataFrame

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

7. Export File

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

DATA COLLECTION – WEB SCRAPING

1. Getting Response from HTML

```
# use requests.get() method with the provided static_url
# assign the response to a object
data = requests.get(static_url)
```

2. Create BeautifulSoup Object

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

3. Find all tables

```
# Use the find_all function in the BeautifulSoup object,
# Assign the result to a list called `html_tables`
html_tables = soup.find_all('table')
```

4. Get column names

```
column_names = []
# Apply find_all() function with `th` element on fi
# Iterate each th element and apply the provided ext
# Append the Non-empty column name (`if name is not
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)
```

5. 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']=[]
```

6. Create DataFrame from dictionary

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

7. Export file

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

DATA WRANGLING

- In the dataset, there are several cases where the booster did not land successfully
  - True Ocean, True RTLS, True ASDS means the mission was successful
  - False Ocean, False RTLS, False ASDS means the mission was a failure
- We need to transform string variables into categorical variables where 1 means the mission has been successful and 0 means the mission was a failure

1. Calculate launches number for each site

```
# Apply value_counts() on column LaunchSite
df.LaunchSite.value_counts()
```

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

2. Calculate the number and occurrence of each orbit

```
# Apply value_counts on Orbit column
df.Orbit.value_counts()
```

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

3. Calculate the number and occurrence of mission outcome per orbit type

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

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

4. Create landing outcome label from Outcome column

```
# landing_class = 0 if bad_outcome
# landing_class = 1 otherwise
landing_class = []
for key,value in df["Outcome"].items():
    if value in bad_outcomes:
        landing_class.append(0)
    else:
        landing_class.append(1)
df['Class']=landing_class
```

5. Export file

```
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 vs. Flight Number
- Payload Mass vs. Orbit Type
- Orbit vs. Payload Mass

Scatter plots show the relationship between two variables.

Bar Graph

- Success Rate vs. Orbit

Bar graphs show the relationship between numeric and categorical variables.

Line Graph

- Success Rate vs. Year

Line graphs show data variables and their trends.

EDA WITH SQL

Performed SQL queries to gather and understand the data:

- Displaying the names of the unique launch sites in the space mission
- Display 5 records where the launch sites begin with the string ‘CCA’
- Display the total payload mass carried by boosters launched by NASA (CRS)
- Display average payload mass carried by booster version F9 v1.1
- List the date when the first successful landing outcome in ground pad was achieved
- List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
- List the total number of successful and failure mission outcomes
- List the names of the booster versions which have carried the maximum payload mass
- List the record, displaying the month names, failure landing outcome sin drone ship, booster versions, launch sites for the months in 2015
- Rank the count of successful landing outcomes between 06/04/2010 and 03/20/2017 in descending order

INTERACTIVE MAP WITH FOLIUM

- Folium map object is a map centered on NASA Johnson Space Center at Houston, Texas
  - Red circle at NASA Johnson Space Center’s coordinate with label showing its name
  - Red circles at each launch site coordinates with label showing launch site name
  - The grouping of points in a cluster to display multiple and different information for the same coordinates
  - Markers to show successful and unsuccessful landings. Green for successful landing and Red for unsuccessful landing
  - Markers to show distance between launch site locations and plot a line between them
- These objects are created in order to understand the problem and the data better. We can show easily all launch sites, their surroundings and the number of successful and unsuccessful landings.

DASHBOARD WITH PLOTLY DASH

Dashboard has dropdown, pie chart, rangeslider, and scatter plot components

- Dropdown allows a user to choose the launch site or all launch sites
- Pie chart shows the total success and the total failure for the launch site chosen with the dropdown component
- Rangeslider allows a user to select a payload mass in a fixed range
- Scatter chart shows the relationship between two variables, in particular Success vs Payload Mass

PREDICTIVE ANALYSIS

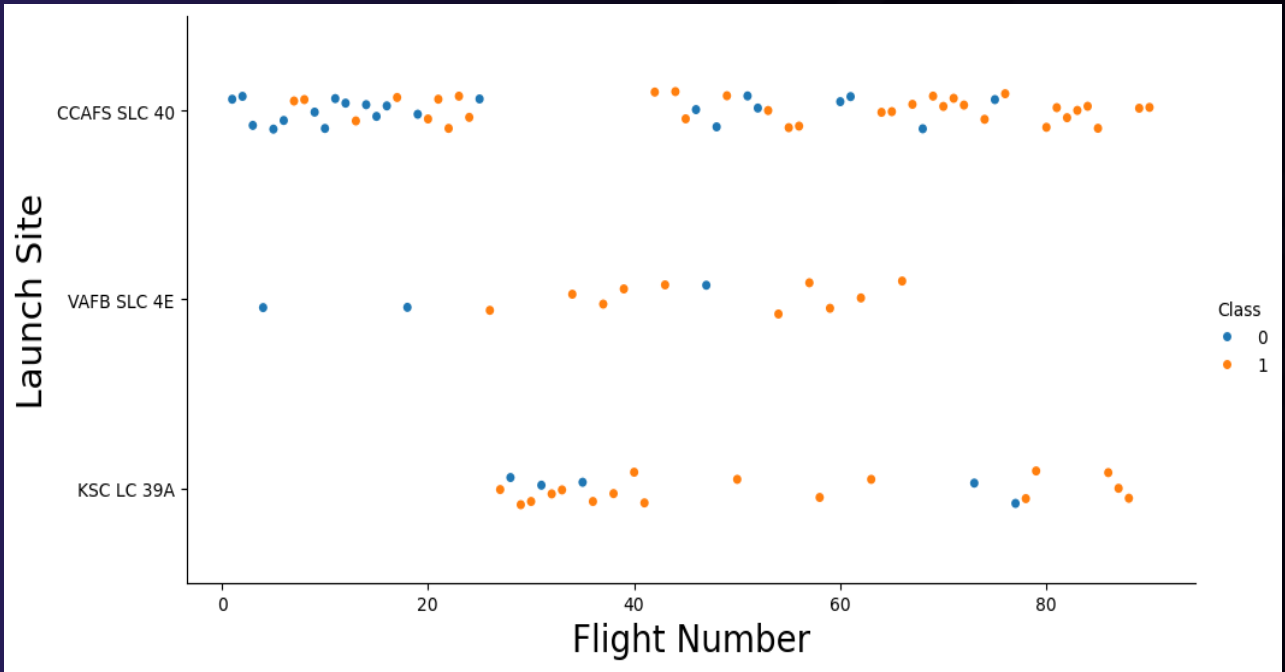
- Data Preparation
  - Load the dataset
  - Normalize data
  - Split data into training and test sets
- Model Preparation
  - Selection of machine learning algorithms
  - Set paramters for each algorithms to GridSearchCV
  - Training GridSearchModel models with training dataset
- Model Evaluation
  - Get best hyperparamters for each type of model
  - Compute accuracy for each model with test dataset
  - Plot Confusion Matrix
- Model Comparison
  - Comparison of models according to their accuracy
  - Choose the model with the best accuracy

RESULTS

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

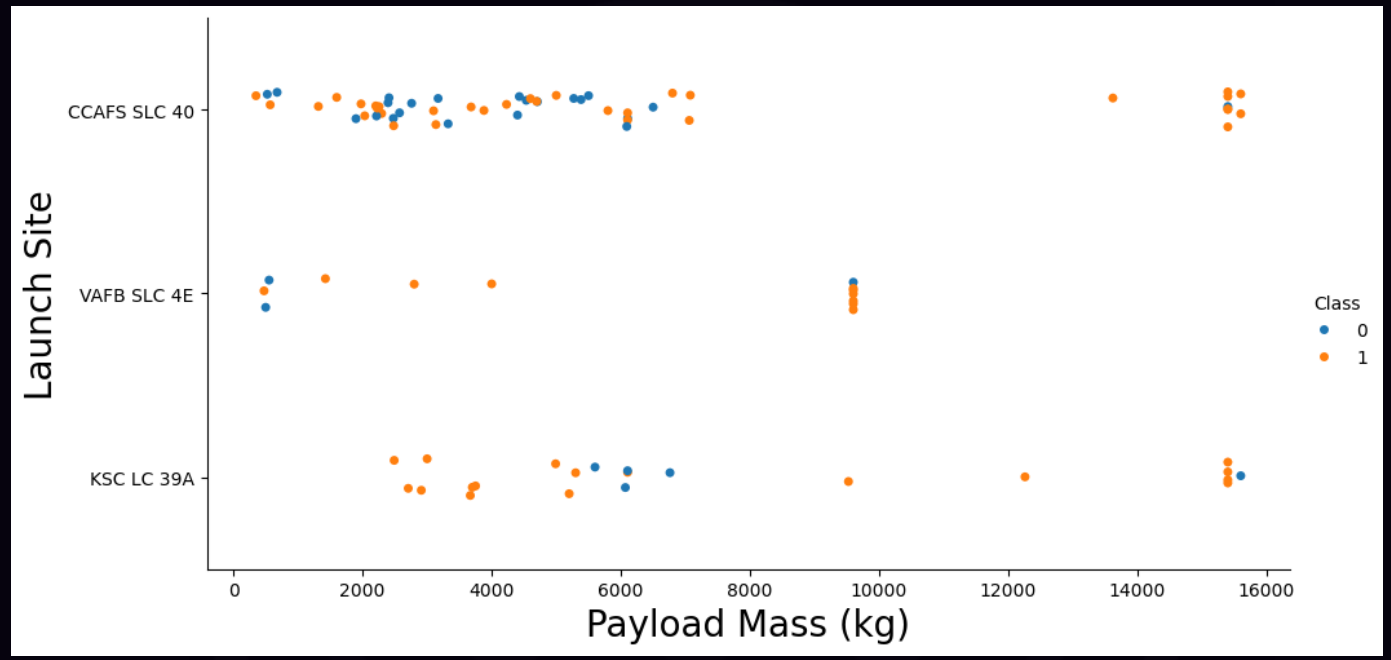
INSIGHTS DRAWN FROM EDA

Flight Number vs. Launch Site



- The earliest flights all failed while the latest flights all succeeded.
- The CCAFS SLC 40 launch site has about a half of all launches
- VAFB SLC 4E and KSC LC 39A have higher success rates
- It can be assumed that each new launch has a higher rate of success

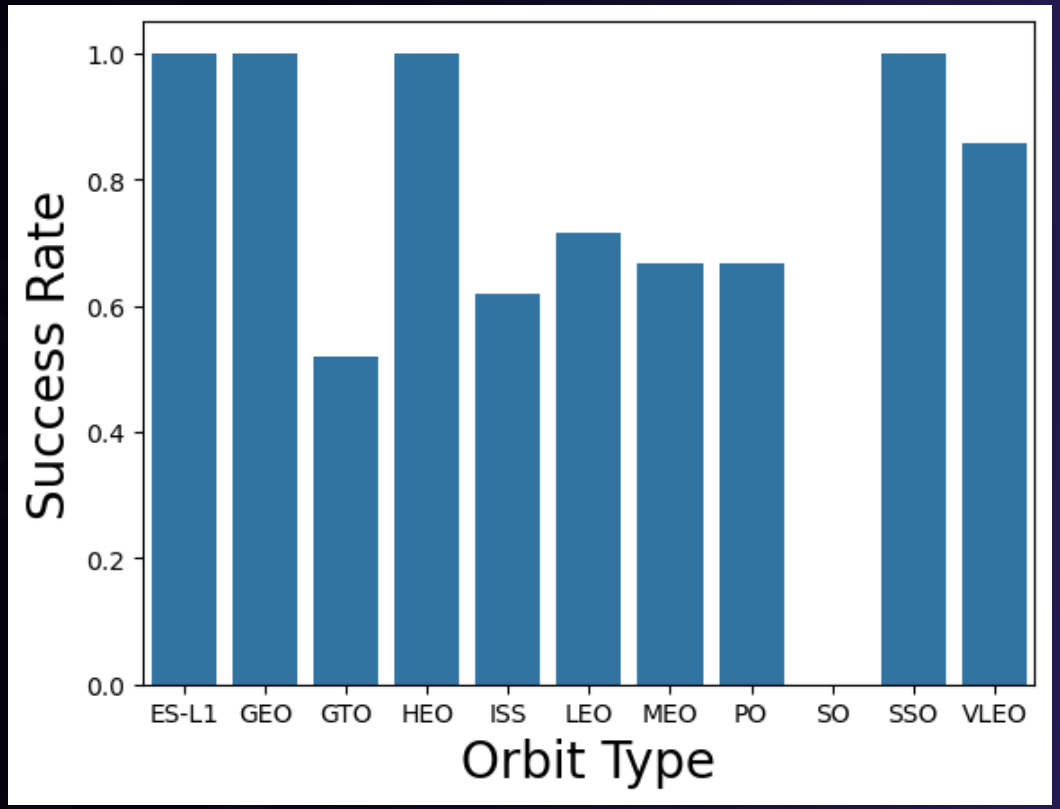
Payload Mass vs. Launch Site



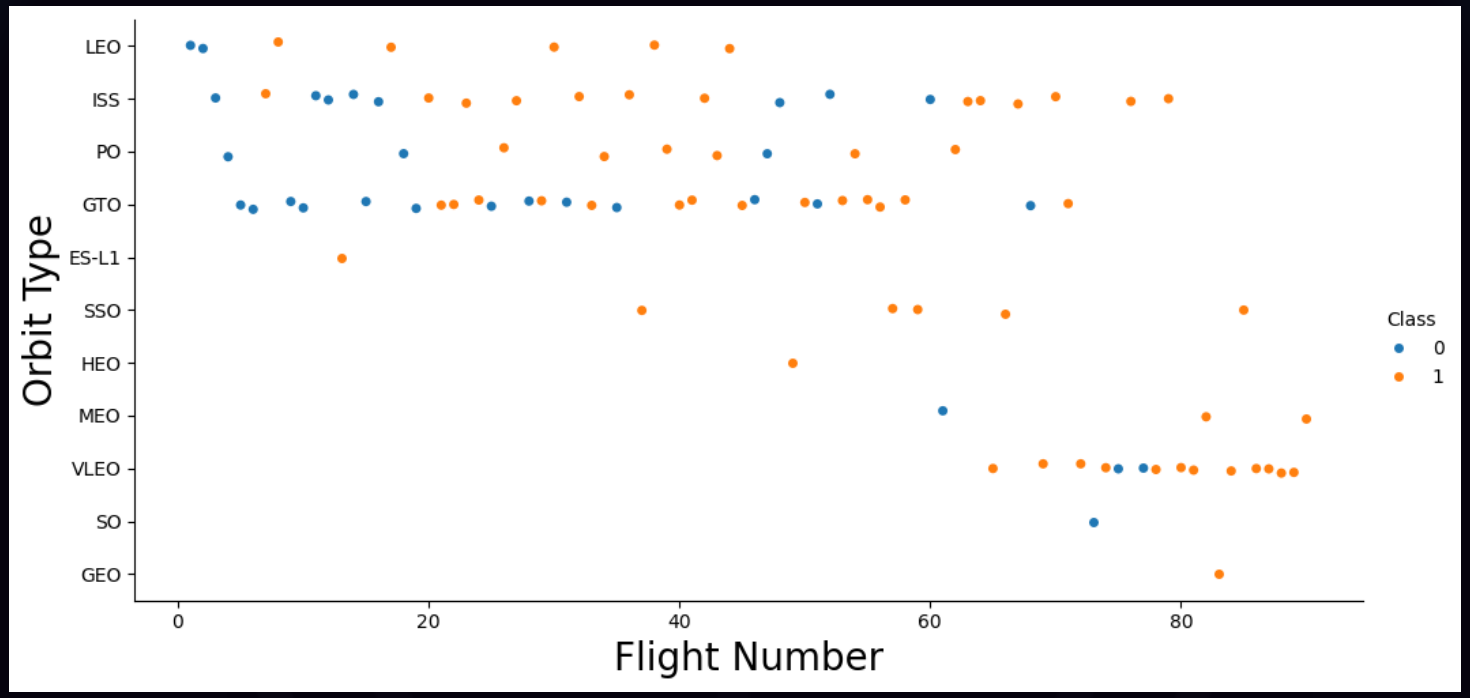
- For every launch site the higher the payload mass, the higher the success rate
- Most of the launches with payload mass over 7000kg were successful
- KSC LC 39A has a 100% success rate for payload mass under 5500kg too.

Success Rate vs. Orbit Type

- Orbits with 100% success rate:
  - ES-L1, GEO, HEO, SSO
- Orbits with 0% success rate:
  - SO
- Orbits with a success rate between 50% and 85%:
  - GTO, ISS, LEO, MEO, PO

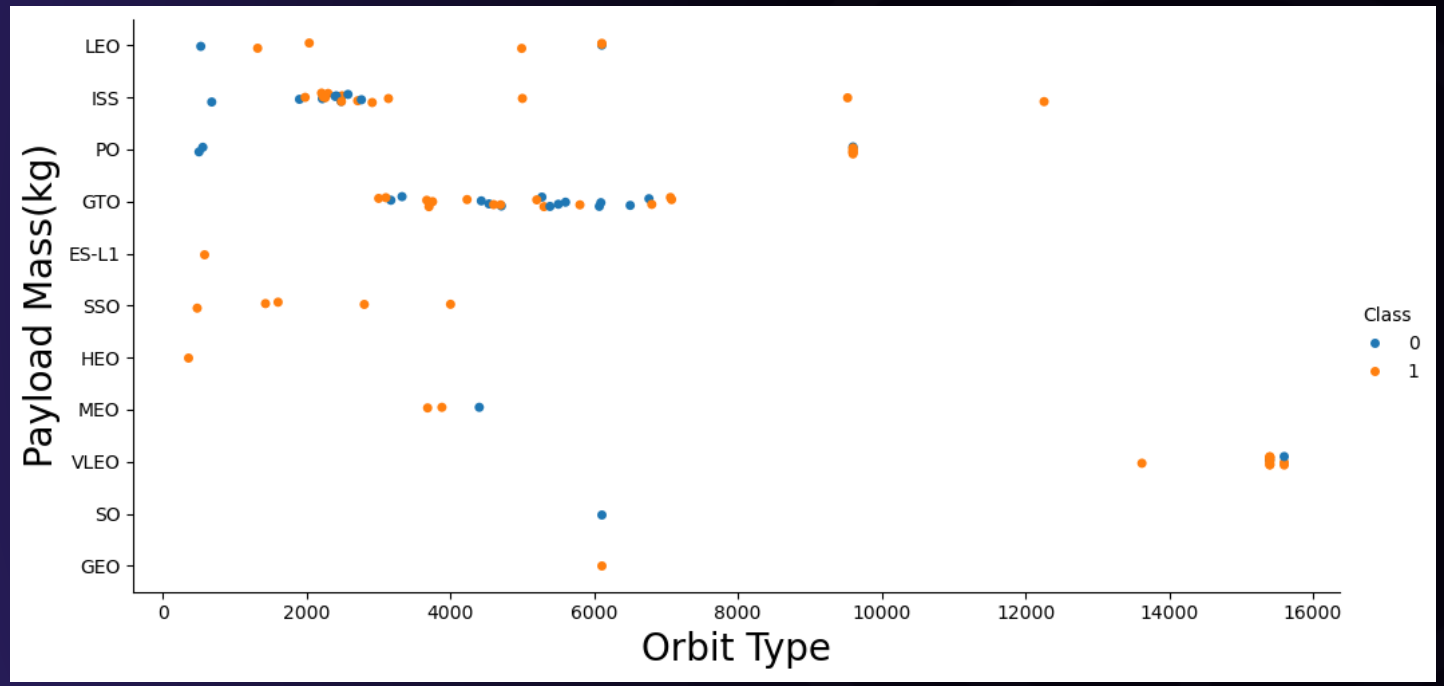


Flight Number vs. Orbit Type



The success rate seems to increase with the number of flights for the LEO orbit. For some orbits like GTO, there is no relation between the success rate and the number of flights.

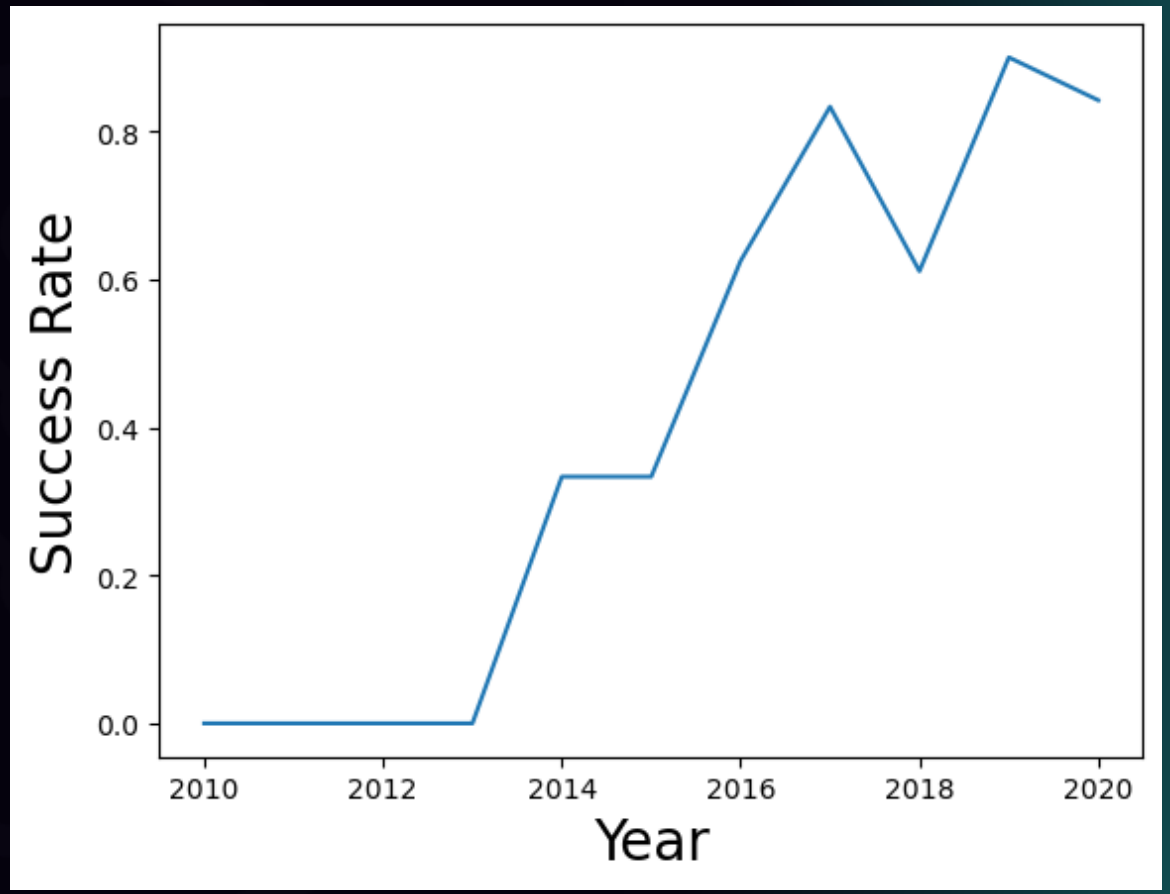
Payload Mass vs. Orbit Type



Heavy payload mass have a negative influence on GTO orbits and positive on GTO and Polar LEO (ISS) orbits

Launch Success Yearly Trend

Since 2013, there has been an increase in the success rate of SpaceX rockets.



All Launch Site Names

```
%sql SELECT distinct(LAUNCH_SITE) from SPACEXTBL
```

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

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

The use of DISTINCT in the query allows to remove duplicate LAUNCH_SITE, so we can find all the names of launch sites without repetition.

‘CCA’ Launch Sites

```
%sql select * from SPACEXTBL where LAUNCH_SITE like 'CCA%' limit 5
```

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

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

The WHERE clause followed by LIKE clause filters the launch sites that contain the substring CCA, and the LIMIT 5 shows 5 records from filtering instead of all of them.

Total Payload Mass

```
%sql select sum(PAYLOAD_MASS__KG_) from SPACEXTBL where CUSTOMER = 'NASA (CRS)'
```

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

sum(PAYLOAD_MASS__KG_)
45596

The query returns the sum of all payload masses where the customers is NASA (CRS)

Average Payload Mass by F9 v1.1

```
%sql select avg(PAYLOAD_MASS__KG_) from SPACEXTBL where BOOSTER_VERSION = 'F9 v1.1'
```

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

avg(PAYLOAD_MASS__KG_)
2928.4

The query returns the average of all payload masses where the booster version contains the substring F9 v1.1

First Successful Ground Landing Date

```
%sql select min(DATE) from SPACEXTBL where LANDING_OUTCOME = 'Success (ground pad)'
```

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

min(DATE)
2015-12-22

The WHERE clause filters dataset to keep only records where landing was successful. With the MIN function, we select the record with the oldest date.

Successful Drone Ship with Payload Mass between 4000 and 6000

```
%sql select BOOSTER_VERSION from SPACEXTBL where LANDING_OUTCOME = 'Success (drone ship)' and PAYLOAD_MASS__KG_ > 4000 and PAYLOAD_MASS__KG_ < 6000
```

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

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

This query returns the booster version where landing was successful and payload mass is between 4000 and 6000 kg.

Total Number of Successful and Failure Mission Outcomes

```
%sql select MISSION_OUTCOME, count(*) AS Total_Count from SPACEXTBL group by MISSION_OUTCOME;
```

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

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

The first SELECT shows the subqueries that return results. The first subquery counts the successful mission, and the second subquery counts the unsuccessful mission. The WHERE clause followed by LIKE clause filters the mission outcomes. The COUNT function counts records filtered.

Boosters Carried Maximum Payload

```
%sql select BOOSTER_VERSION from SPACEXTBL where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEXTBL)
```

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

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

The subquery filters the data by reutnring the heaviest payload mass with MAX function. The main query uses subquery results and returns unique booster version (SELECT and DISTINCT) with the heaviest payload mass.

2015 Launch Records

```
%sql select case substr(DATE, 6, 2) when '01' then 'January' when '02' then 'February' when '03' then 'March' when '04' then 'April'
when '05' then 'May' when '06' then 'June' when '07' then 'July' when '08' then 'August' when '09' then 'September'
when '10' then 'October' when '11' then 'November' when '12' then 'December' end as MONTH_NAME, BOOSTER_VERSION, LAUNCH_SITE, LANDING_OUTCOME
from SPACEXTBL where substr(DATE, 1, 4) = '2015' and LANDING_OUTCOME = 'Failure (drone ship)' order by substr(DATE, 6, 2)
```

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

MONTH_NAME	Booster_Version	Launch_Site	Landing_Outcome
January	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
April	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

The query returns month, booster version, launch site where landing was unsuccessful and landing date took place in 2015. Substr function process date to take month or year.

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

```
%sql select * from SPACEXTBL where LANDING_OUTCOME like 'Success%' and (DATE between '2010-06-04' and '2017-03-20') order by date desc
```

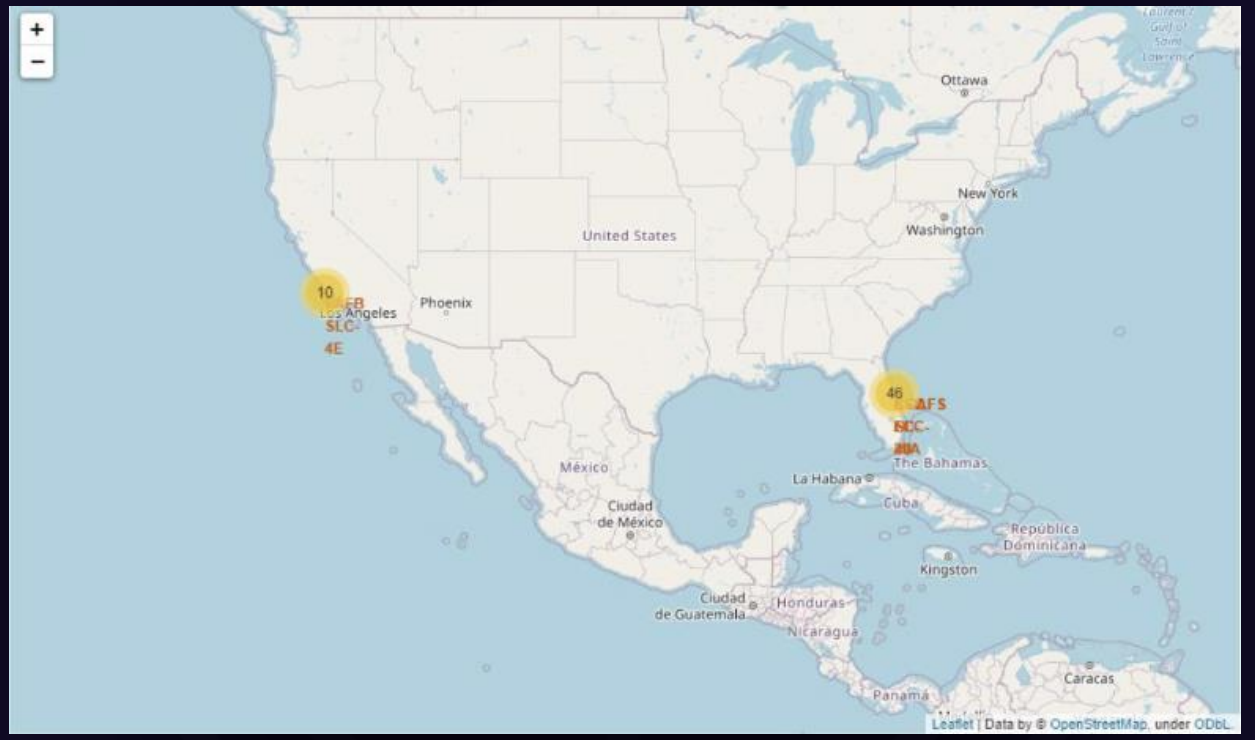
```
 * sqlite:///my_data1.db
Done.
```

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2017-02-19	14:39:00	F9 FT B1031.1	KSC LC-39A	SpaceX CRS-10	2490	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
2017-01-14	17:54:00	F9 FT B1029.1	VAFB SLC-4E	Iridium NEXT 1	9600	Polar LEO	Iridium Communications	Success	Success (drone ship)
2016-08-14	5:26:00	F9 FT B1026	CCAFS LC-40	JCSAT-16	4600	GTO	SKY Perfect JSAT Group	Success	Success (drone ship)
2016-07-18	4:45:00	F9 FT B1025.1	CCAFS LC-40	SpaceX CRS-9	2257	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
2016-05-27	21:39:00	F9 FT B1023.1	CCAFS LC-40	Thaicom 8	3100	GTO	Thaicom	Success	Success (drone ship)
2016-05-06	5:21:00	F9 FT B1022	CCAFS LC-40	JCSAT-14	4696	GTO	SKY Perfect JSAT Group	Success	Success (drone ship)
2016-04-08	20:43:00	F9 FT B1021.1	CCAFS LC-40	SpaceX CRS-8	3136	LEO (ISS)	NASA (CRS)	Success	Success (drone ship)
2015-12-22	1:29:00	F9 FT B1019	CCAFS LC-40	OG2 Mission 2 11 Orbcomm-OG2 satellites	2034	LEO	Orbcomm	Success	Success (ground pad)

Ranking the count of landing outcomes between the date 2010-06-04 and 2017-03-20 in descending order.

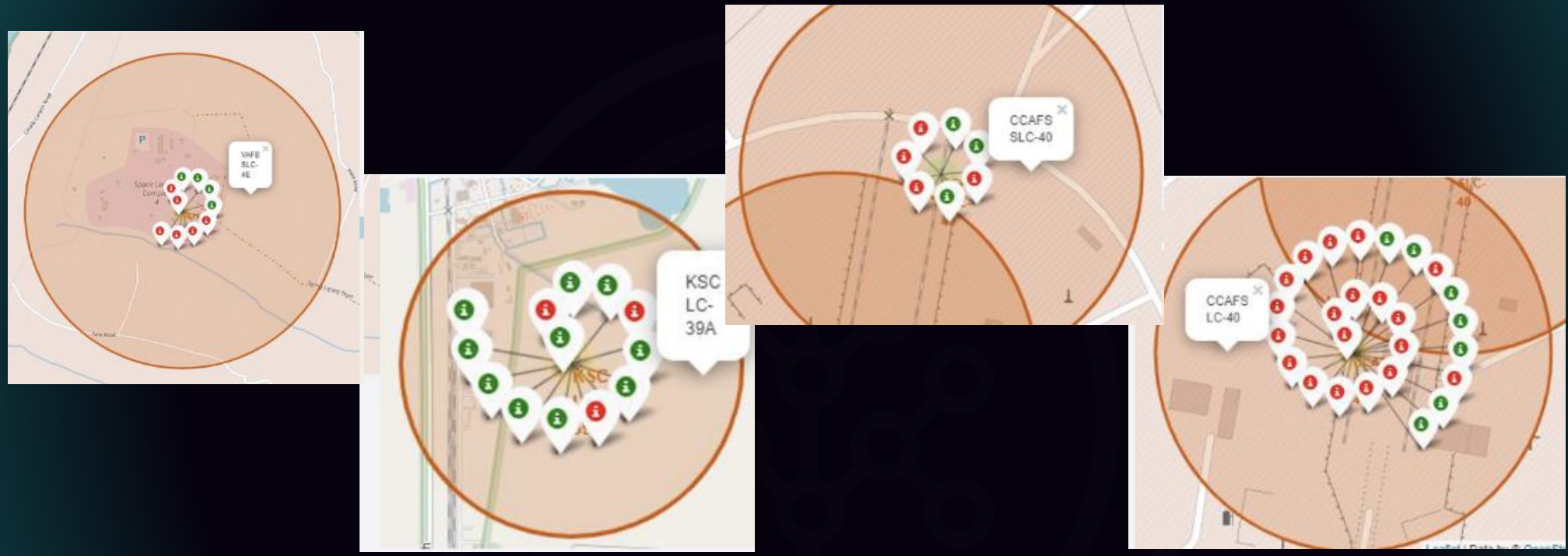
Interactive Map with Folium

Ground Stations



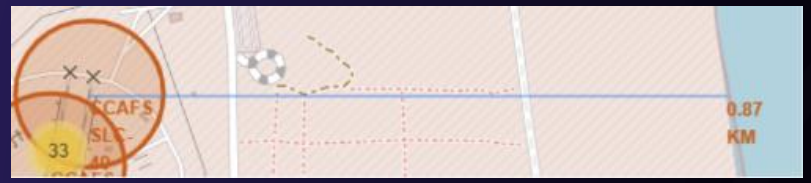
All the SpaceX launch sites are located on the coast of the United States

Color Labeled Markers



Green marker represents successful launches and red marker represents unsuccessful launches. We note that KSC LC-39A has a higher launch success rate.

Distances between CCFAS SLC-40 and its Proximities



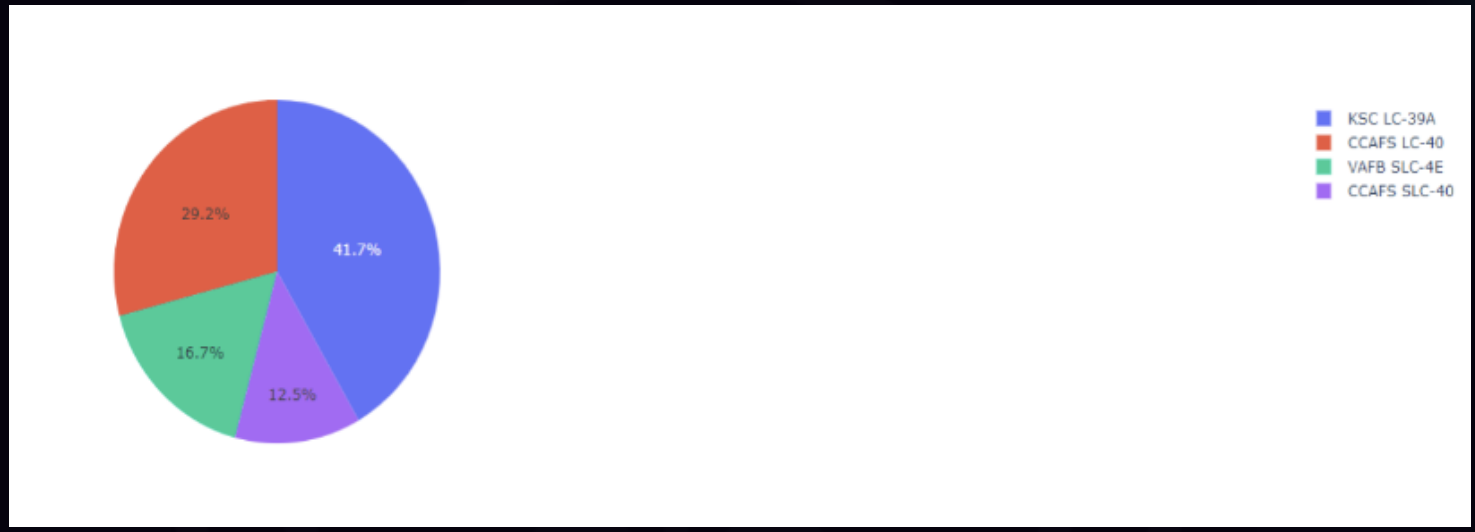
CCAFS SLC-40 is in close proximity to:

- Railways
- Highways
- Coastline

CCAFS SLC-40 does not keep distance away from cities.

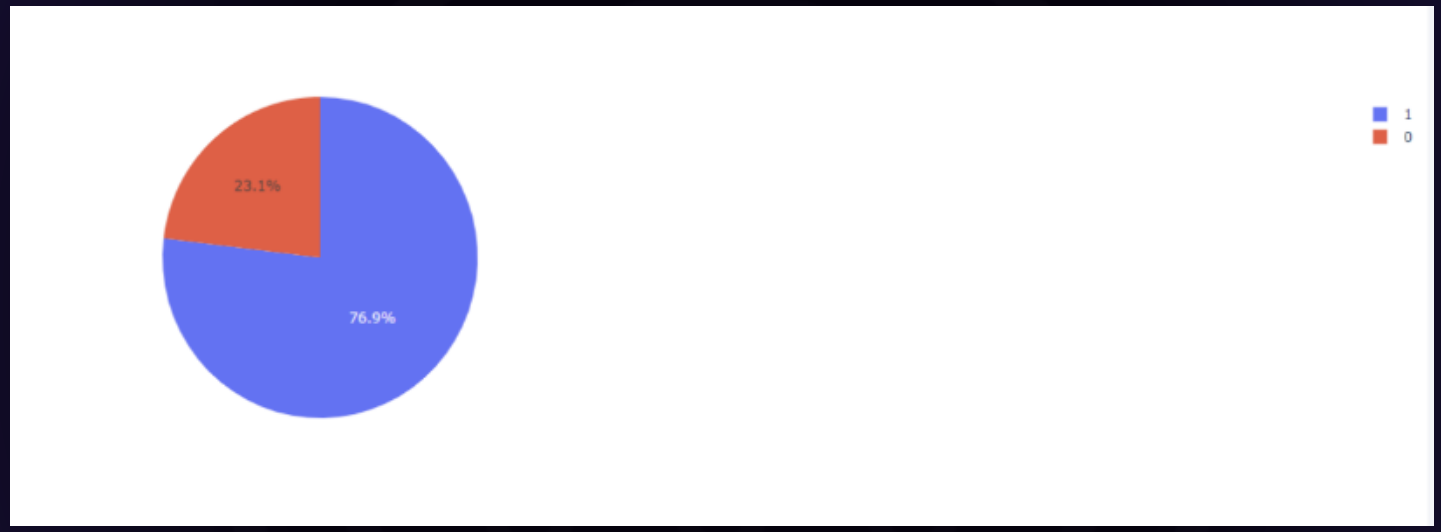
Dashboard with Plotly Dash

Total Success by Site



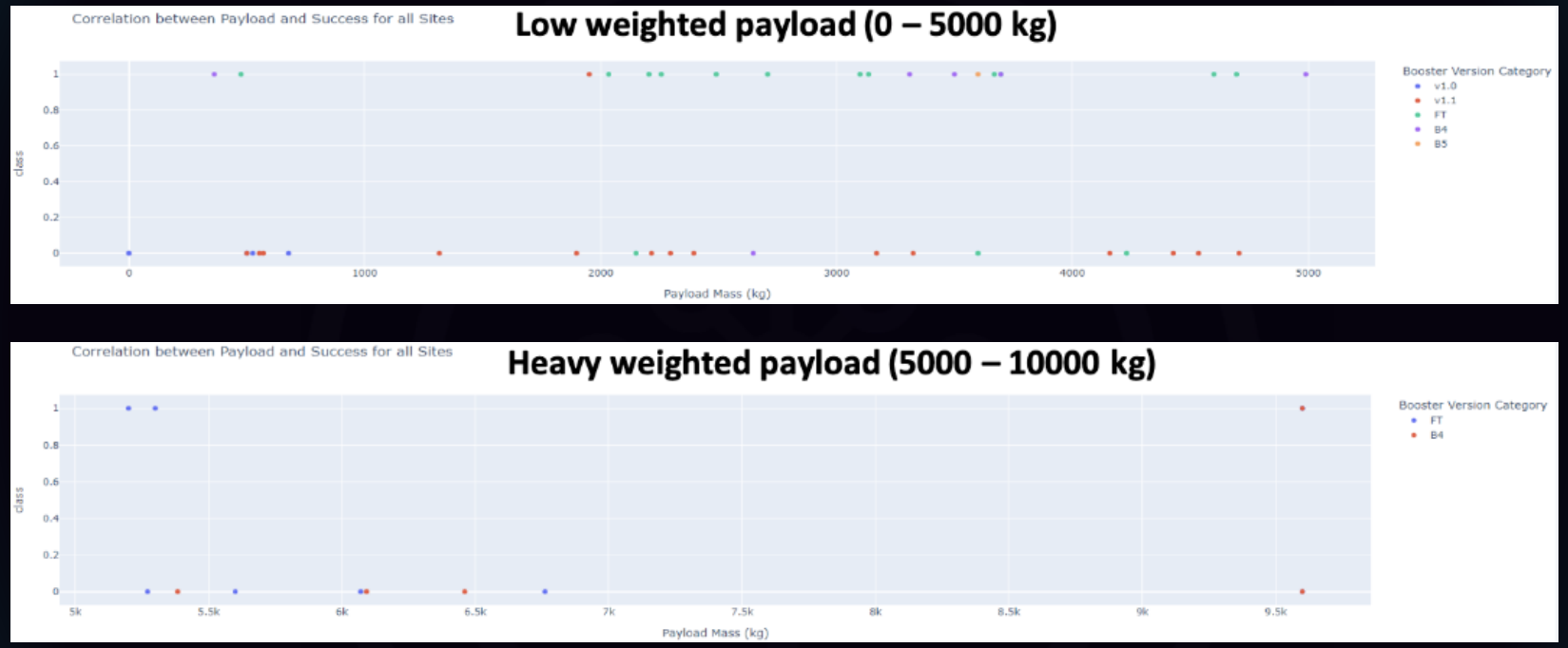
KSC LC-39A has the best success rate of launches

Total Success Launches for Site KSC LC-39A



It has a 76.9% success rate while getting a 23.1% failure rate

Payload Mass vs. Launch Outcome for all Sites

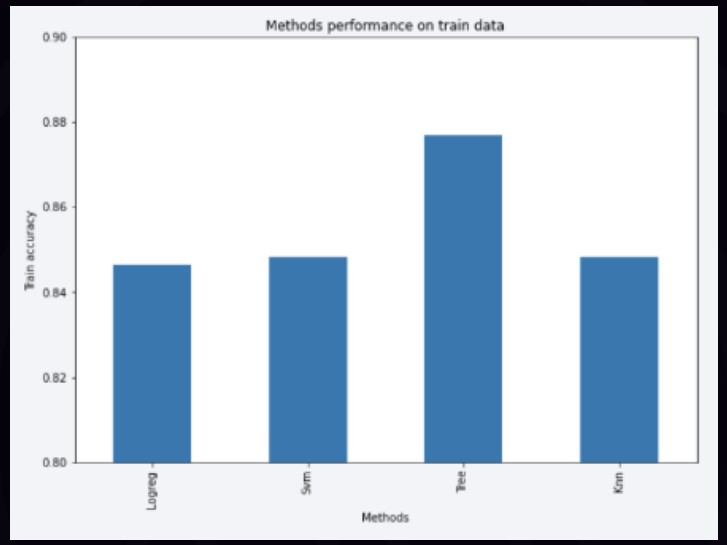


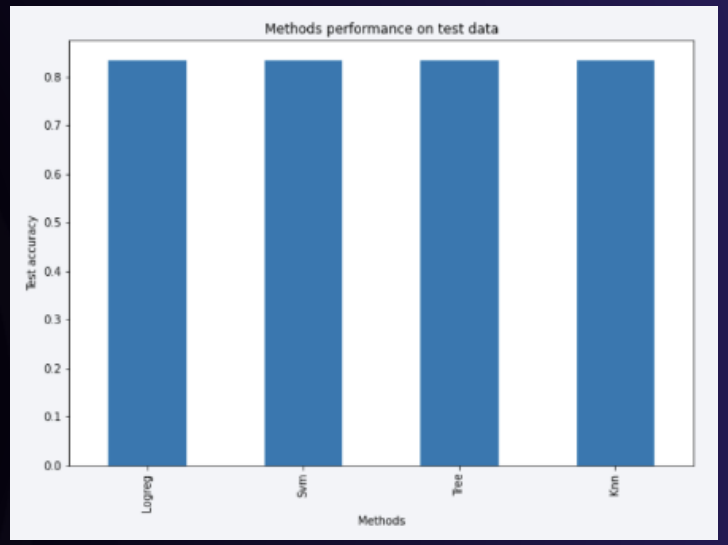
Low weighted payloads have a better success rate than the heavy weighted payloads

Predictive Analysis (Classification)

Classification Accuracy

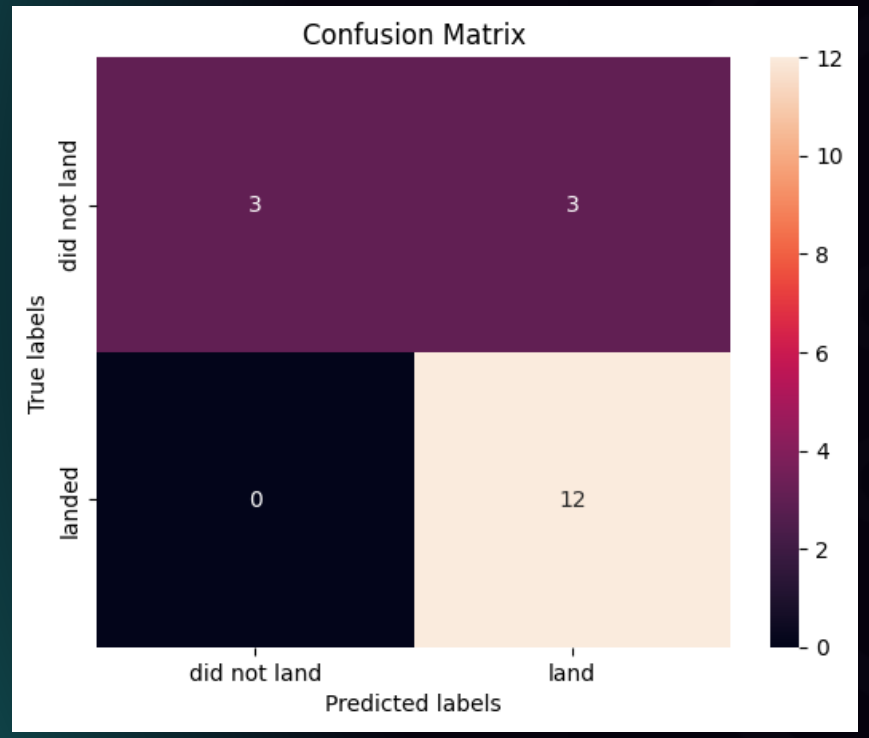
	Accuracy Train	Accuracy Test
Tree	0.876786	0.833333
Knn	0.848214	0.833333
Svm	0.848214	0.833333
Logreg	0.846429	0.833333





For accuracy test, all methods performed similar. We could get more test data to decide between them, but if we really need to choose one right now, we would take the decision tree.

Confusion Matrix



As the test accuracy are all equal, the confusion matrices are also identical. The main problem of these models are false positives.

		Actual values 1	Actual values 0
Predicted values	1	TP	FP
	0	FN	TN

CONCLUSION

- Decision Tree Model is the best algorithm for this dataset
- Launches with a low payload mass show better results than launches with a larger payload mass
- Most of launch sites are in proximity to the Equator line and all the sites are in very close proximity to the coast
- The success rate of launches increases over the years
- KSC LC-39A has the highest success rate of the launches from all the sites
- Orbits ES-L1, GEO, HEO, and SSO have 100% success rate

Thank You!