
IBM Developer
SKILLS NETWORK
WINNING SPACE RACE
WITH DATA SCIENCE
THI QUYNH HUONG NGUYEN
07/07/2020
NASA

OUTLINE

EXECUTIVE SUMMARY

SUMMARY OF METHODOLOGIES

Data Collection through API
Data Collection with Web Scraping
Data Wrangling
Exploratory Data Analysis with SQL
Exploratory Data Analysis with Data Visualization
Interactive Visual Analytics with Folium
Machine Learning Prediction

SUMMARY OF ALL RESULTS

Exploratory Data Analysis result
Interactive analytics in screenshots
Predictive Analytics result

INTRODUCTION

❑ Project background and context

- As able to reuse the first stage, SpaceX bids for launches with lower price compared to other providers.
  - *Space X cost for Falcon 9 rocket launces is 62 million dollars, while other competitors cost 165 millions dollars upward*
- In order to bid against SpaceX, we need to determine their possible launch cost by predicting whether the first stage will land. Therefor, this project creates a machine learning pipeline to predict if the first stage will land successfully.
- The project uses a dataset that includes records of all payload carried during a SpaceX mission into outer space.

❑ Problems you want to find answers

- Factors that related to successful landing
- The interaction amongst various features that related to the success rate of a successful landing.
- Operating conditions needs to be in place to ensure a successful landing program.

METHODOLOGY

1. Data collection methodology:

- SpaceX API
- Web-scraping from Wikipedia

2. Perform data wrangling

- One-hot encoding data fields for Machine Learning and data cleaning of null values and irrelevant columns

3. Perform exploratory data analysis (EDA) using visualization and SQL

- Plotting: Scatter Graphs, Bar Graphs to show relationships between variables to show patterns of data

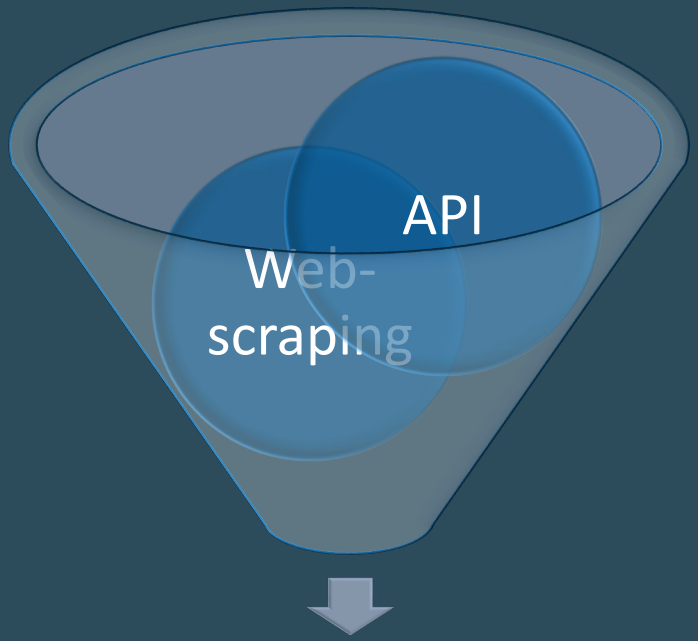
4. Perform interactive visual analytics using Folium and Plotly Dash

- Launch-site locations
- Dashboard application for performing SpaceX launch data in real-time

5. Perform predictive analysis using classification models

- Logistic Regression, K-Nearest Neighbor , Support Vector Machine, Decision Tree

1. Data Collection



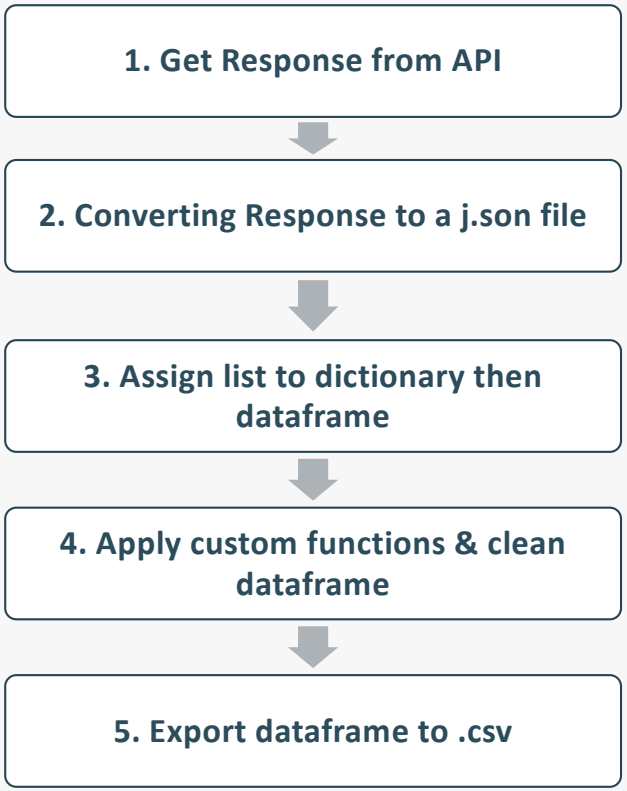
Ready for *data consolidation* and *wrangling*

- ✓ **API**
  - gives data about launces (rocket used, payload delivered, launch specifications, landing specifications, landing outcomes)
  - the SpaceX REST API endpoints, or URL, starts with api.spacexdata.com/v4/
- ✓ **Web Scraping**
  - extract information from Wikipedia with Beautiful Soup

1. Data Collection – SpaceX API

Objective:

Request to the SpaceX API
Clean the requested data



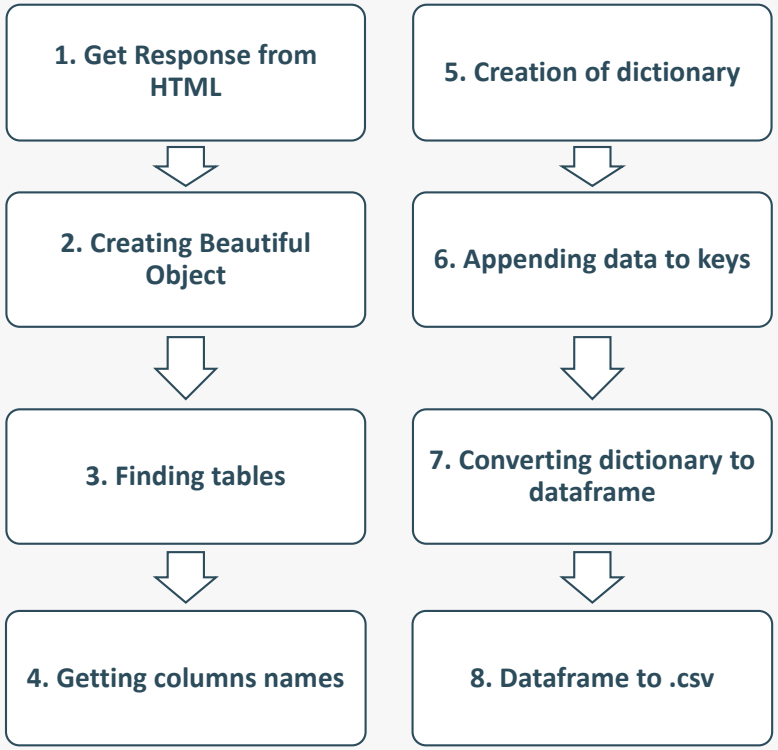

```
# 1.Get response from API
spacex_url="https://api.spacexdata.com/v4/launches/past"

response = requests.get(spacex_url)

# 2.Converting Response to a j.son file
static_json_df=response.json()

data=pd.json_normalize(static_json_df)

# 3. 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}

data=pd.DataFrame.from_dict(launch_dict)

#4. Apply custom functions & clean dataframe
    ## Filter dataframe to only include Falcon 9 launches
data_falcon9=data[(data['BoosterVersion']!='Falcon 1')]
data_falcon9.head()

    ## Dealing with missing values
mean=data_falcon9['PayloadMass'].mean()
data_falcon9['PayloadMass']=data_falcon9['PayloadMass'].replace(np.nan,mean)
data_falcon9['PayloadMass'].isnull().sum()

#5. Export dataframe
data_falcon9.to_csv('dataset_part_1.csv', index=False)
```

https://github.com/kathlyn-huongnguyen/IBM_Data_Science_Capstone/blob/main/Data%20Collection%20API.ipynb

1. Data Collection – Web Scraping

Objective:

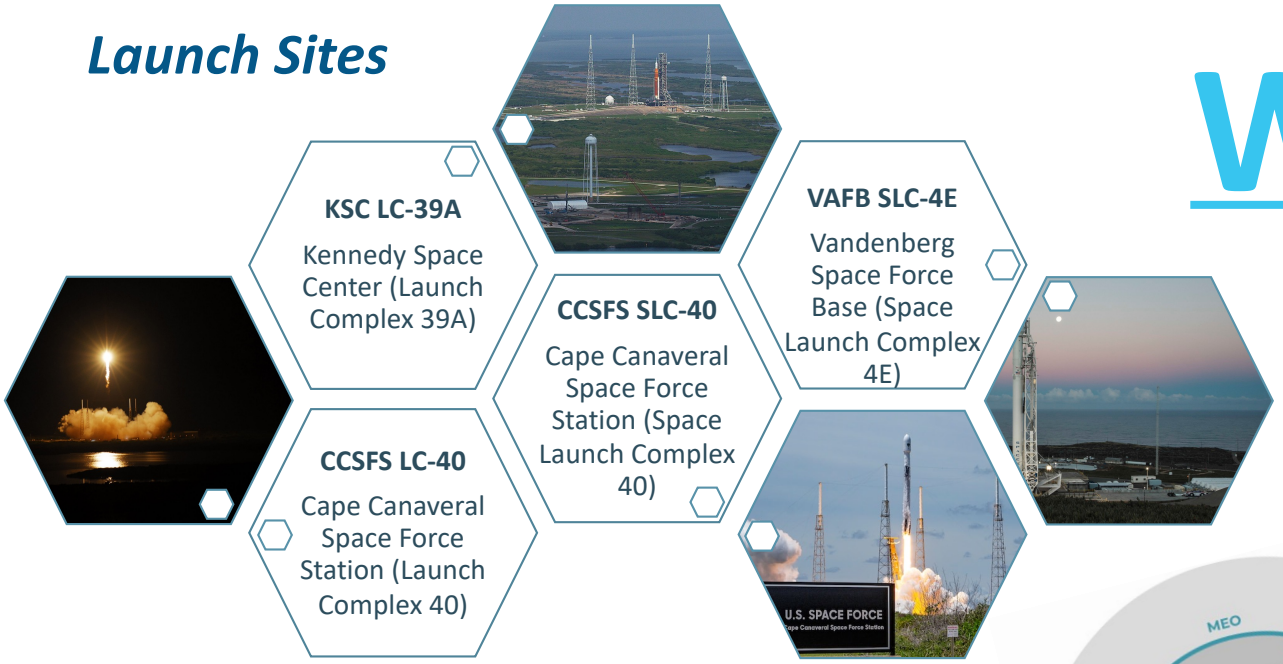
Extract a Falcon 9 launch records HTML table from Wikipedia
Parse the table and convert it into a Pandas data frame




```
#1. Getting Response from HTML
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falc
html_data=requests.get(static_url).text
html_data

#2. Creating BeautifulSoup Object
soup=BeautifulSoup(html_data,'html5lib')

#3. Finding table
soup.find_all('table')

#4. Getting columns names
html_tables=soup.find_all('table')
first_launch_table = html_tables[2]

column_names = []

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. Appending data to keys
##REFER TO OUTPUT 74, 75 NOTEBOOK

#7. Converting dictionary to dataframe
df = pd.DataFrame(launch_dict)

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

https://github.com/kathlyn-huongnguyen/IBM_Data_Science_Capstone/blob/main/Data%20Collection%20with%20Webscraping.ipynb

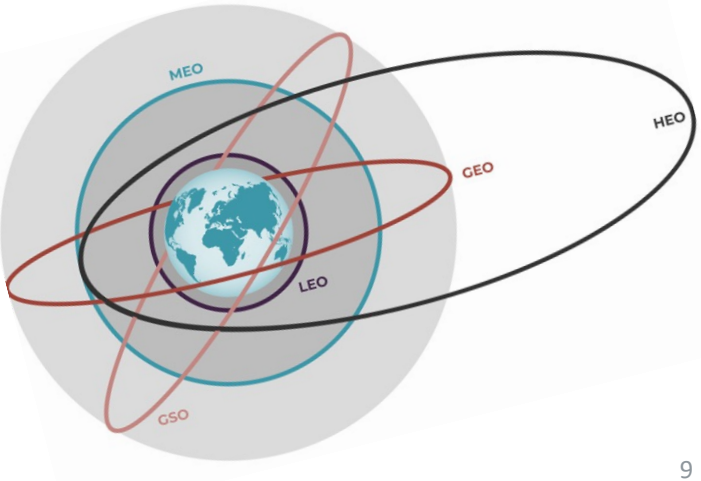
Launch Sites



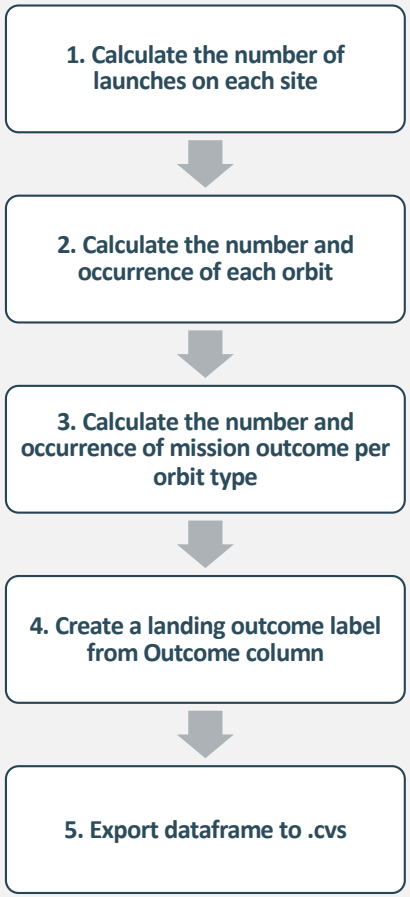
Wrang-ling

2. Data

Orbits



https://github.com/kathlyn-huongnguyen/IBM_Data_Science_Capstone/blob/main/Data%20Wrangling.ipynb



Objective: Exploratory Data Analysis
Determine Training Labels

```
#1. Calculate the number of launches on each site
df['LaunchSite'].value_counts()

#2. Calculate the number and occurrence of each orbit
df['Orbit'].value_counts()

#3. Calculate the number and occurrence of mission outcome per orbit type
landing_outcomes=df['Outcome'].value_counts()

#4. Create a landing outcome label from Outcome column
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 dataframe to .cvs
df.to_csv("dataset_part\_2.csv", index=False)
```

3. EDA - Data Visualization

SCATTER GRAPH

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

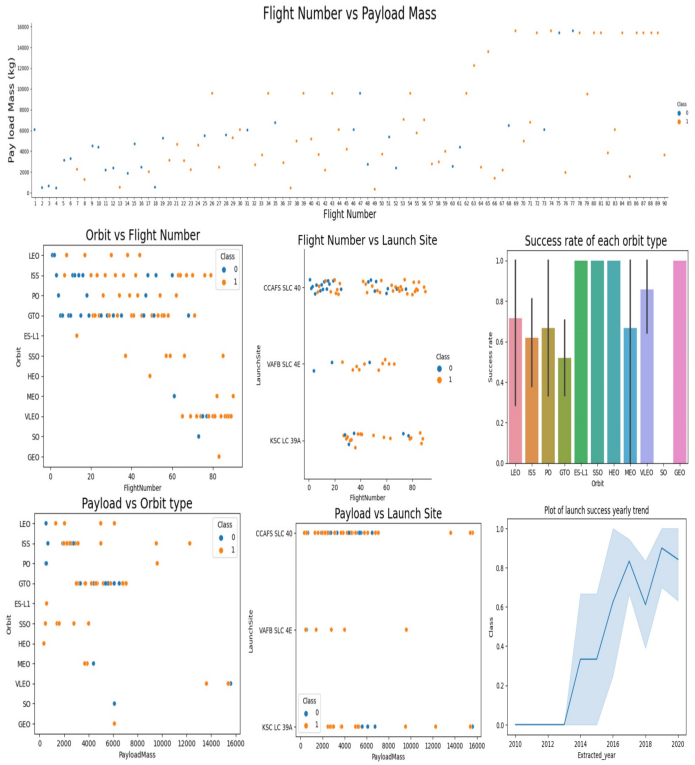
BAR GRAPH

Means vs. Orbit

LINE GRAPH

Success Rate vs Year

https://github.com/kathlyn-huongnguyen/IBM_Data_Science_Capstone/blob/main/EDA%20with%20Visualization.ipynb



SCATTER GRAPH

- Show how much one variable is affected by another
- The relationship between two variables is called their correlation
- Scatter plots usually consists of a large body of data

BAR GRAPH

- A bar diagram makes it easy to compare sets of data between different groups at a glance
- The graph represents categories on one axis and a discrete value in the other
- The goal is to show the relationship between the two axes
- Bar charts can also show big changes in data over time

LINE GRAPH

- Line graphs are useful in that they show data variables and trend very clearly and can help to make predictions about the results of data not yet recorded

3. EDA - SQL

- **Load the SpaceX dataset into IBM-db2, connect to database from Jupyter notebook**

- **Execute SQL queries to get insights from the data**
  - The names of unique launch sites in the space mission
  - 5 records where launch sites begin with the string 'CCA'
  - The total payload mass carried by boosters launched by NASA (CRS)
  - The average payload mass carried by booster version f9 v 1.1
  - The date where the successful landing outcome in drone ship was achieved
  - Names of the boosters which have success in ground pad and have payload mass greater than 4000 but less than 6000
  - The total number of successful and failure mission outcomes
  - Names of booster versions which have carried the maximum payload mass
  - Records which will display the month names, failure landing outcomes in drone ship ,booster versions, launch site for the months in year 2015
  - Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order

https://github.com/kathlyn-huongnguyen/IBM_Data_Science_Capstone/blob/main/EDA%20with%20SQL.ipynb

4. Build an Interactive Map with Folium

- ❑ Mark all launch sites, and add map objects such as markers circles, lines to mark the success or failure of launches for each site on the folium map

- ❑ Assign the feature launch outcomes, (failure or success) to class 0 and 1
  - 0 for failure
  - 1 for success

- ❑ Use the color-labeled marker clusters to identify which launch sites have relatively high success rate
- ❑ Calculate the distances between a launch site to its proximities
- ❑ Answer the questions:
  - Are launch sites near railways, highways and coastlines
  - Do launch sites keep certain distance away from cities

https://github.com/kathlyn-huongnguyen/IBM_Data_Science_Capstone/blob/main/Launch%20Site%20Location.ipynb

5. Build a Dashboard with Plotly Dash

- Build a website live 24/7 so you can play around with the data and view the data
  - 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 straight forward

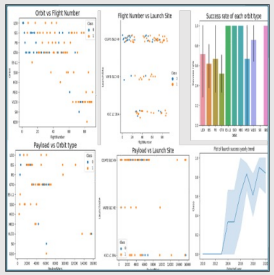
https://github.com/kathlyn-huongnguyen/IBM_Data_Science_Capstone/blob/main/Dash.ipynb

6. Predictive Analysis (Classification)

- BUILDING 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 dataset into the GridSearchCV objects and train our 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 l

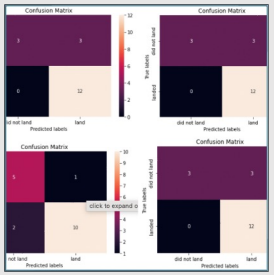
https://github.com/kathlyn-huongnguyen/IBM_Data_Science_Capstone/blob/main/Machine%20Learning%20Prediction.ipynb

RESULT



EDA – INTERACTIVE ANALYTICS

Predictive analysis results



The SVM, KNN, and logistic Regression models are the best in terms of prediction accuracy for this dataset

Low weighted payloads perform better than heavier payloads

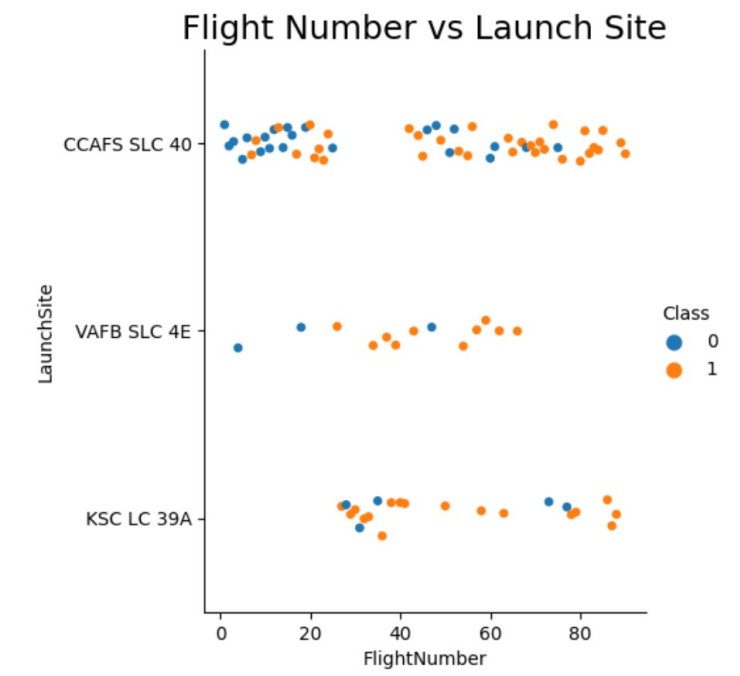
The success rates for SpaceX launches is directly proportional time in years thet will eventually perfect the launches

KSC LC 39A had the most successful launches from all sites

Orbit GEO, HEO, SSO, ES L1 has te best Success Rate


INSIGHTS
DRAWN
FROM
EDA
SPACEX

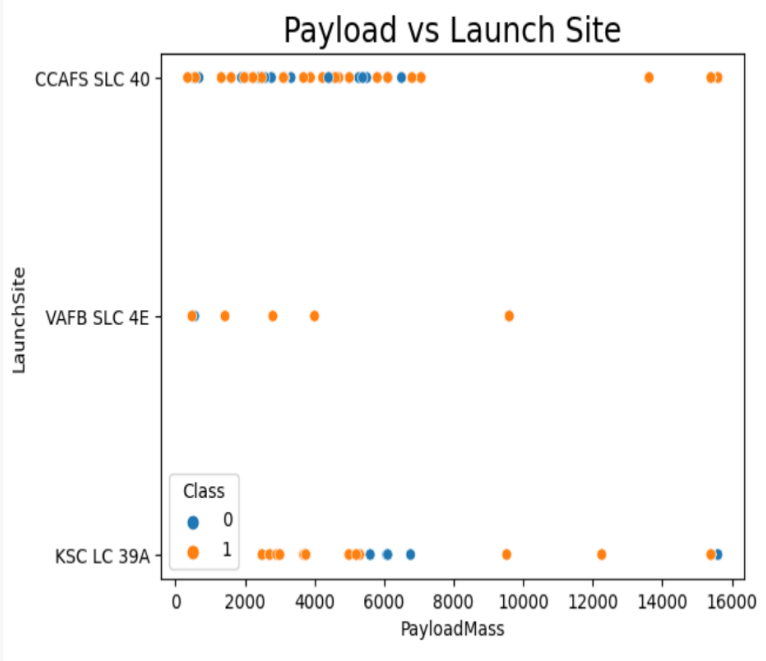
Flight Number vs. Launch Site




```
sns.catplot(x='FlightNumber', y='LaunchSite', data=df, hue='Class')
plt.title('Flight Number vs Launch Site', fontsize=18)
plt.show()
```

As the number of flights increase, the success rate at a launch site increase

Payload vs. Launch Site

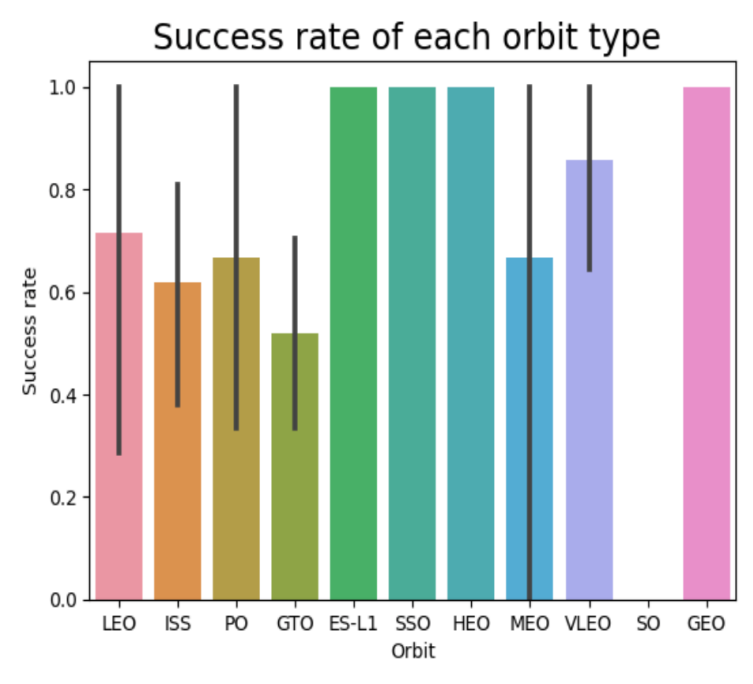



```
sns.scatterplot('PayloadMass','LaunchSite', data=df, hue='Class')
plt.title('Payload vs Launch Site', fontsize=18)
plt.show()
```

The greater the payload mass for Launch Site CCSFS SLC 40 the higher the success rate for the Rocket

The pattern is not clear to make any decision whether the Launch Site success rate is depended on Pay Load Mass

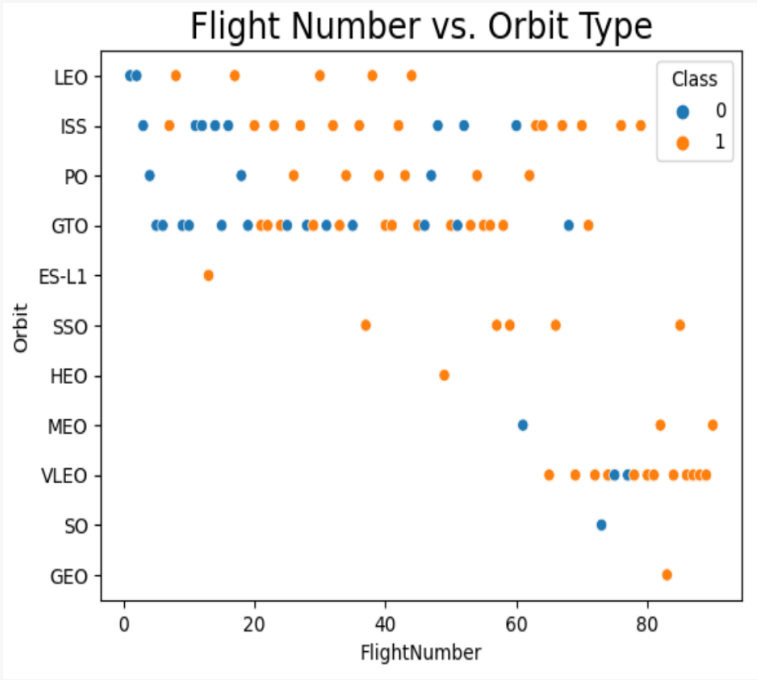
Success rate vs. Orbit Type




```
df_orbit_class=df[['Orbit','Class']]
grouped_df=df[['Orbit','Class']].groupby(df['Orbit'],as_index=True).mean()
grouped_df.rename(columns={'Class':'Success rate'}, inplace=True)

sns.barplot('Orbit','Class', data=df_orbit_class)
plt.title('Success rate of each orbit type ', fontsize=18)
plt.ylabel('Success rate', fontsize=10)
```

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

Flight Number vs. Orbit Type

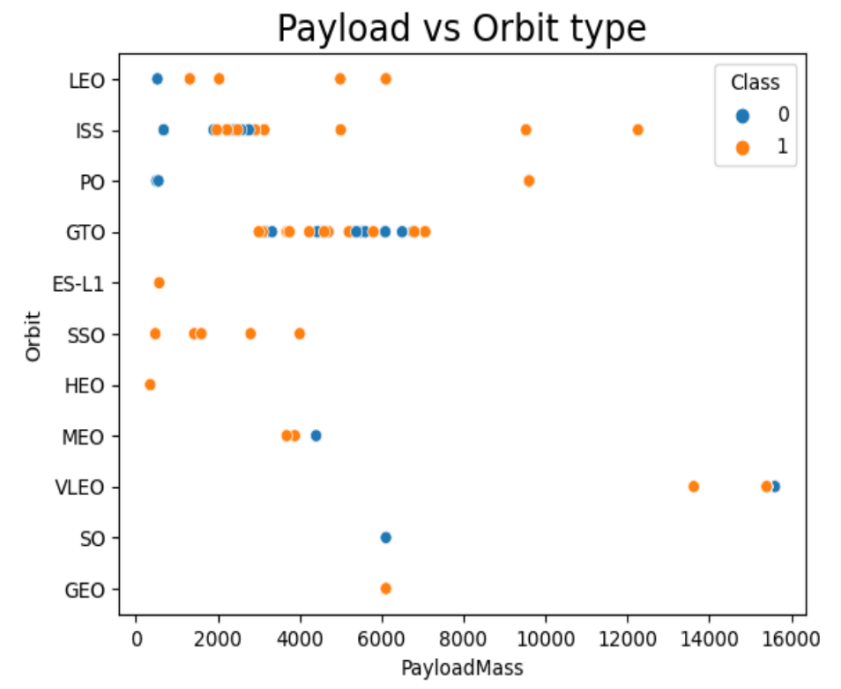



```
plt.title('Flight Number vs. Orbit Type', fontsize=18)
plt.show()
```

In LEO orbit, Success rate appears related to the number of flights

In GTO orbit, there seems to be no relationship between success rate and flight number

Payload vs. Orbit Type

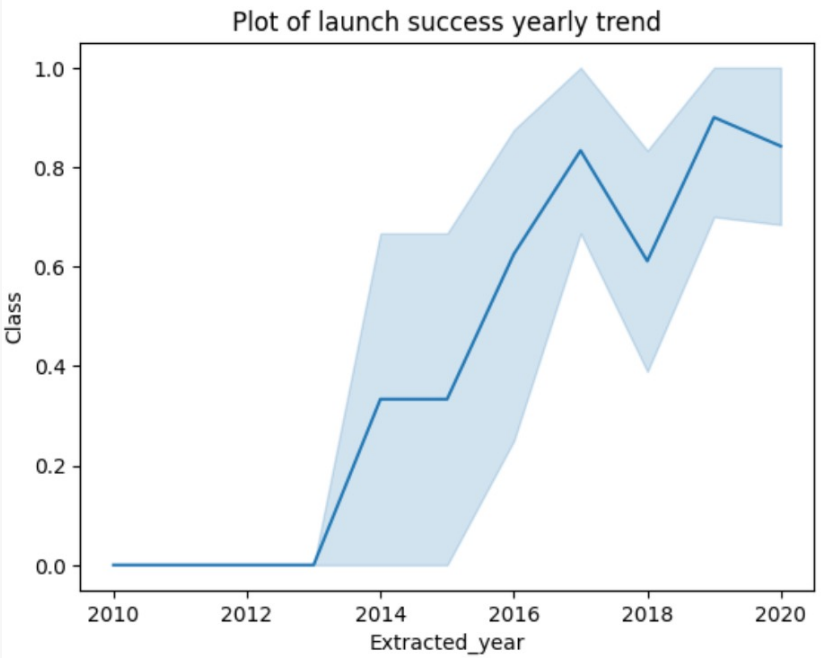



```
sns.scatterplot('PayloadMass','Orbit', data=df, hue='Class')
plt.title('Payload vs Orbit type', fontsize=18)
plt.show()
```

With heavy payloads, the successful landing are more for PO, LEO and ISS orbits

EDA with Visualization

Launch Success Yearly Trend




```
sns.lineplot(data=df_copy, x='Extracted_year', y='Class')
plt.title('Plot of launch success yearly trend');
plt.show()
```

The success rate since 2014 kept on increasing till 2020

EDA with SQL

All launch Sites Names

```
%sql select distinct (Launch_Site) from SPACEXTBL;
```

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

Use keyword DISTINCT to show only unique launch sites from the SpaceX data

Launch site names begin with 'CCA'

```
%%sql select * from SPACEXTBL
where (Launch_site) like '%CCA%' limit 5;
```

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
04/06/2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
08/12/2010	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)
22/05/2012	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
08/10/2012	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
01/03/2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

```
%%sql select count(Launch_site) from SPACEXTBL
where (Launch_site) like '%CCA%';
```

count(Launch_site)
60

Display 5 records where launch sites begin with 'CCA'

EDA with SQL

Total Payload Mass carried by boosters launched by NASA (CRS)

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

sum(PAYLOAD_MASS__KG_)
45596

Total payload carried by boosters launched by NASA (CRS) is 45,596 (kg)

Average Payload Mass by F9 v 1.1

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

avg(PAYLOAD_MASS__KG_)
2534.6666666666665

The average payload mass carried by booster version F9 v1.1 is 2,928.4 kg

EDA with SQL

First successful ground landing rate

```
%%sql select *
from SPACEXTBL
where Landing_Outcome like '%Success (ground pad)%'
order by Date desc
limit 1;
```

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
22/12/2015	1:29:00	F9 FT B1019	CCAFS LC-40	OG2 Mission 2 11 Orbcomm-OG2 satellites	2034	LEO	Orbcomm	Success	Success (ground pad)

The dates of the first successful landing outcome on the ground pad was 22nd December 2015

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%%sql
select Booster_Version, PAYLOAD_MASS__KG_
from SPACEXTBL
where Landing_Outcome like '%Success (drone ship)%'
AND PAYLOAD_MASS__KG_ between 4000 AND 6000;
```

Booster_Version	PAYLOAD_MASS__KG_
F9 FT B1022	4696
F9 FT B1026	4600
F9 FT B1021.2	5300
F9 FT B1031.2	5200

Use *where* clause to filter for boosters which have successfully landed on drone ship and applied the *and* condition to determine successful landing with payload mass greater than 4000 but less than 6000

EDA with SQL

Total number of *successful* and *failure* mission outcomes

```
%%sql select Mission_Outcome, count(Mission_Outcome)
from SPACEXTBL
group by Mission_Outcome;
```

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

Use wildcard like '%' to filter for *where* Mission Outcome was a success or a failure

Boosters carried maximum payload

```
%%sql
select Booster_Version, PAYLOAD_MASS__KG_
from SPACEXTBL
where PAYLOAD_MASS__KG_ =
(select max(PAYLOAD_MASS__KG_)from SPACEXTBL);
```

Booster_Version	PAYLOAD_MASS__KG_
F9 B5 B1048.4	15600
F9 B5 B1049.4	15600
F9 B5 B1051.3	15600
F9 B5 B1056.4	15600
F9 B5 B1048.5	15600
F9 B5 B1051.4	15600
F9 B5 B1049.5	15600
F9 B5 B1060.2	15600
F9 B5 B1058.3	15600
F9 B5 B1051.6	15600
F9 B5 B1060.3	15600
F9 B5 B1049.7	15600

Determine the booster that have carried the maximum payload using a subquery in the *where* clause and the *max()* function

EDA with SQL

Rank landing outcomes between 06-04-2010 and 20-04-2017

```
%%sql
select substr(Date, 4, 2) as month,substr(Date, 7, 4) as year,
Landing_Outcome, Booster_Version, Launch_Site
from SPACEXTBL
where Landing_Outcome like '%Failure (drone ship)%'
and substr(Date, 7, 4) = '2015';
```

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

Select Landing outcomes and the *count* out landing outcomes from the data and used the *where* clause to filter for landing outcomes *between* 06-04-2010 and 20-04-2017

Appy *group by* clause to group the landing outcomes and the *order by* clause to order the grouped landing outcome in descending order

2015 Launch Records

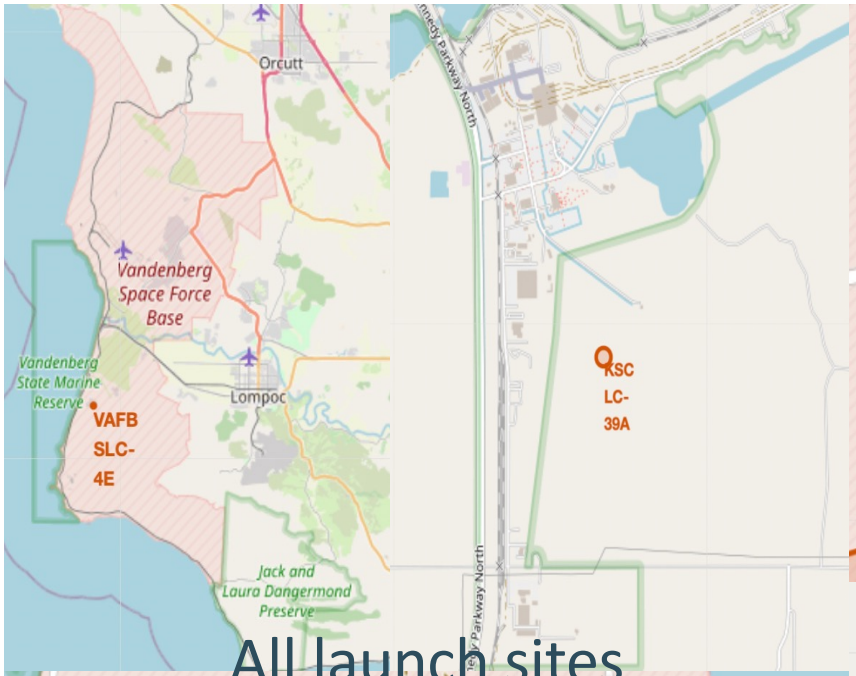
```
%%sql
SELECT Landing_Outcome, COUNT(Landing_Outcome)
FROM SPACEXTBL
WHERE DATE BETWEEN '04-06-2010' AND '20-03-2017'
GROUP BY Landing_Outcome
Having Landing_Outcome like '%Success%'
ORDER BY COUNT(Landing_Outcome) DESC;
```

Landing_Outcome	COUNT(Landing_Outcome)
Success	21
Success (drone ship)	8
Success (ground pad)	6

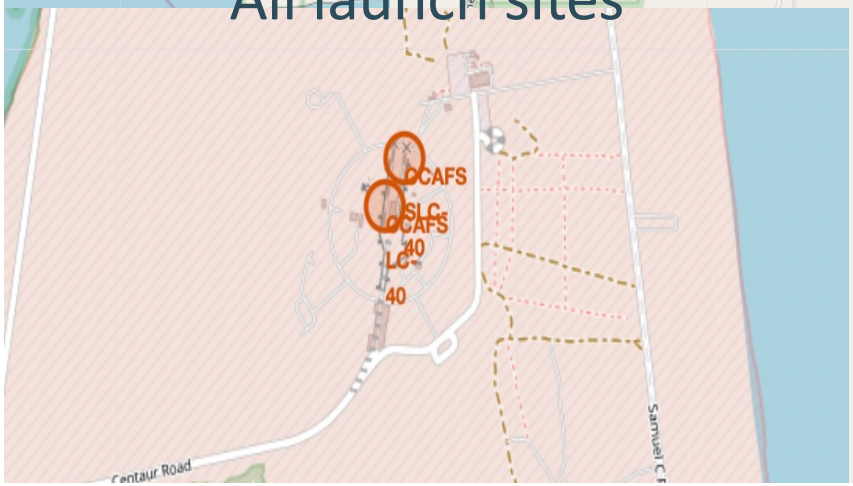
Use combinations of the *where* clause, *like, and,* and *between* conditions to filter for failed landing outcomes in drone ship, their booster versions, and launch site names for year 2015

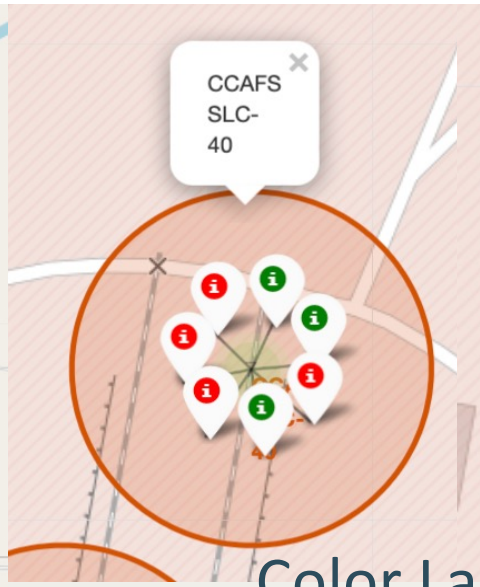

Launch Sites Proximities Analysis

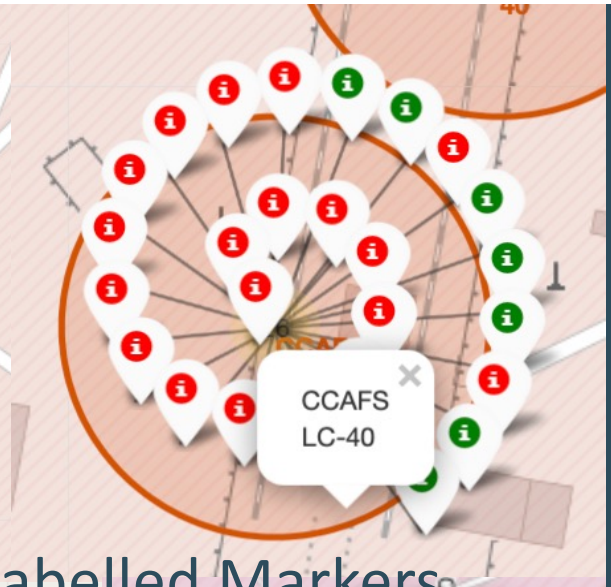
Interactive map with Folium



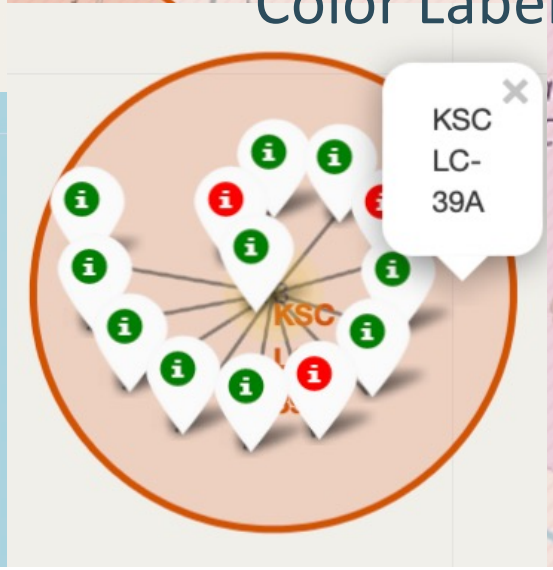
All launch sites

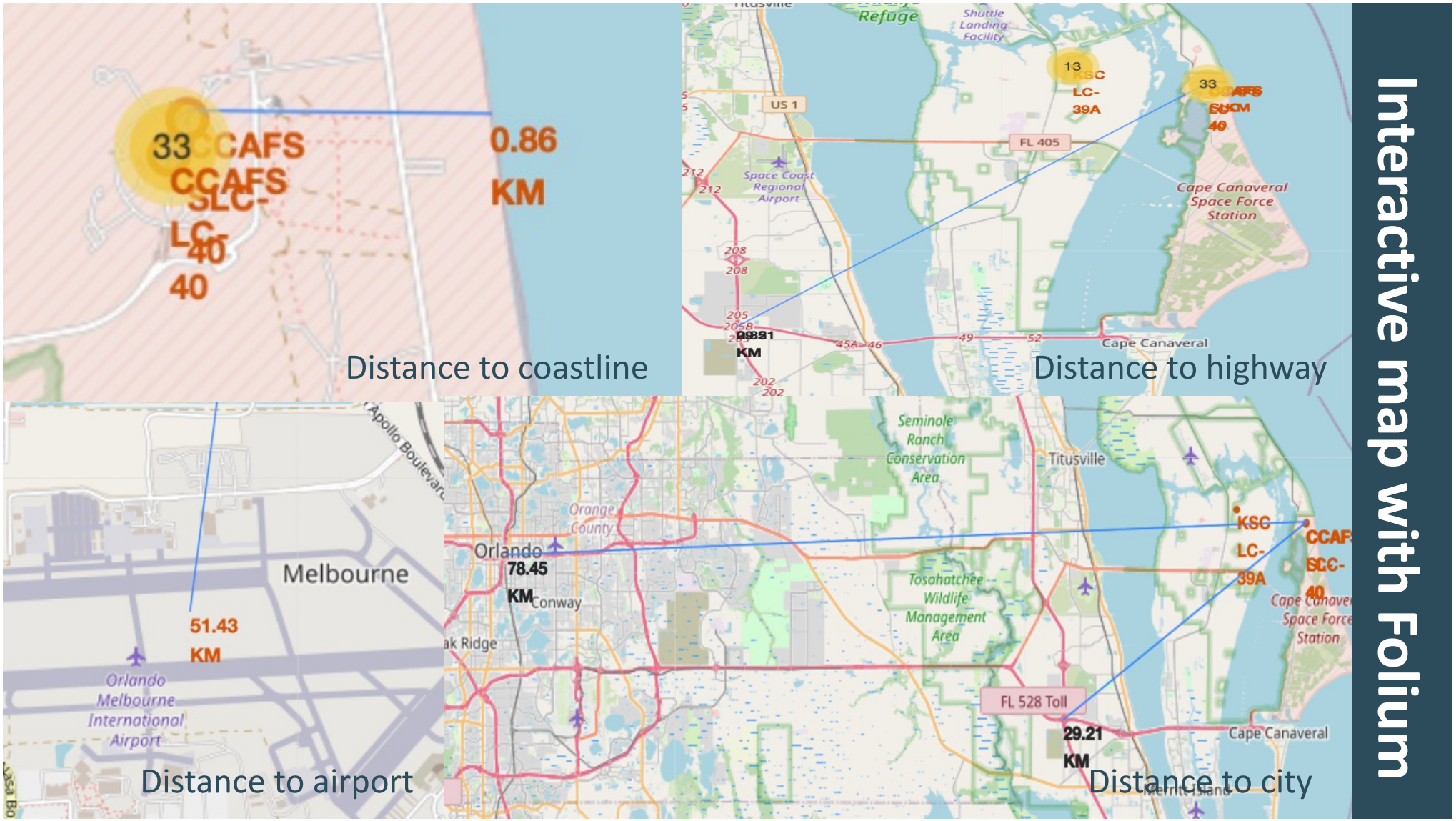






Color Labelled Markers




33
CCAFS
SLC-
40
0.86
KM
Distance to coastline
Refuge
Shuttle Landing Facility
13
KSC
LC-
39A
33
CCAFS
SLC-
40
US 1
FL 405
Space Coast Regional Airport
Cape Canaveral Space Force Station
Cape Canaveral
KM
Distance to highway
Interactive map with Folium
Melbourne
51.43
KM
Orlando Melbourne International Airport
Distance to airport
Seminole Ranch Conservation Area
Titusville
Orange County
Orlando
78.45
KM
Conway
Tosohatchee Wildlife Management Area
KSC
LC-
39A
FL 528 Toll
29.21
KM
Cape Canaveral
Distance to city

Interactive map with Folium

```
closest_highway = 28.56335, -80.57085
closest_railroad = 28.57206, -80.58525
closest_city = 28.10473, -80.64531

distance_highway = calculate_distance(launch_site_lat, launch_site_lon, closest_highway[0], closest_highway[1])
print('distance_highway =',distance_highway, ' km')
distance_railroad = calculate_distance(launch_site_lat, launch_site_lon, closest_railroad[0], closest_railroad[1])
print('distance_railroad =',distance_railroad, ' km')
distance_city = calculate_distance(launch_site_lat, launch_site_lon, closest_city[0], closest_city[1])
print('distance_city =',distance_city, ' km')
```

SpaceX launch sites are in the United States of America coasts

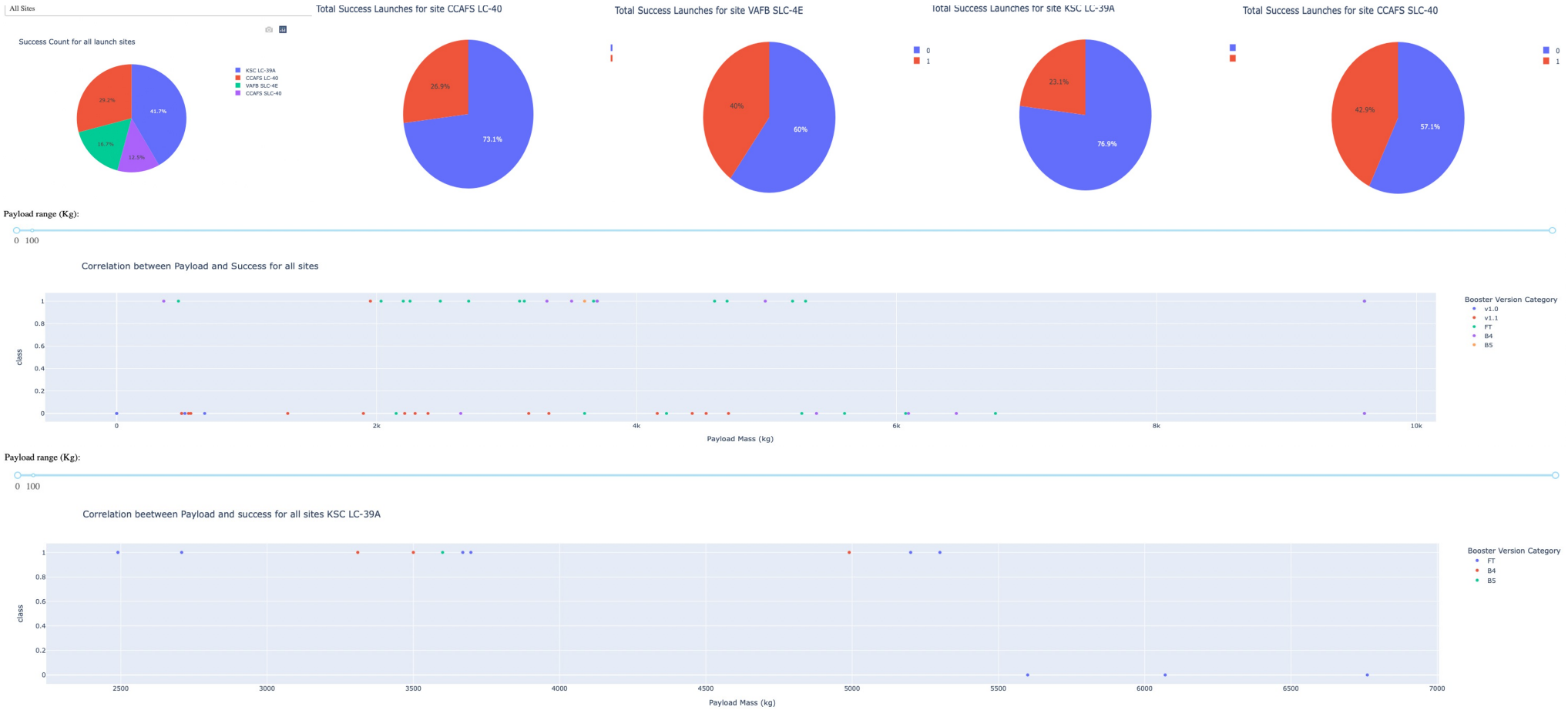
Are launch sites in close proximity to railways? NO

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


Build a Dashboard with Plotly Dash



KSC LC – 39A had the most successful launches from all the sites

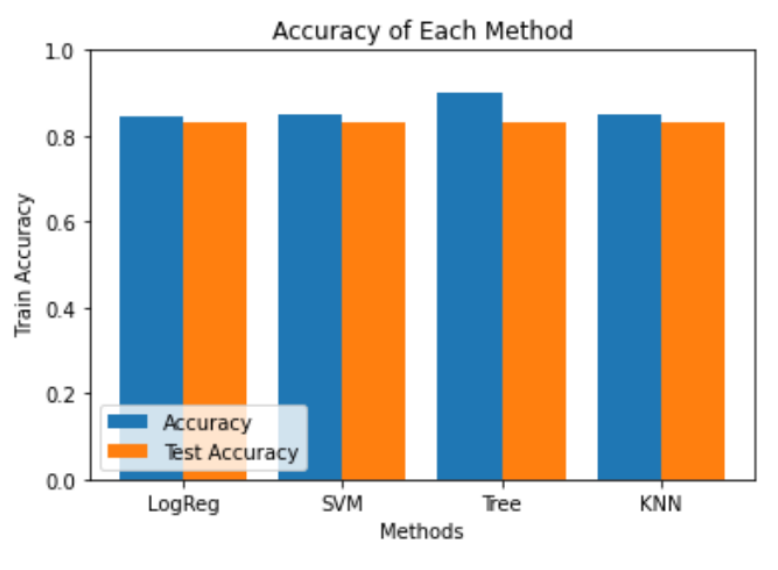
KS LC-39A achieves a 76.9% success rate while getting a 23.1% failure rate

The success rates for low weighted payloads is higher than the heavy weighted payloads


very strong income performance well ahead of prior year
Projected sales of main products in 2013
Distribution marketing participation in the securities market.
Share of market activity
Distribution of the securities market key players
Revenue growth divisions.
TYU division
FRT division
PREDICTIVE ANALYSIS
(CLASSIFICTION)

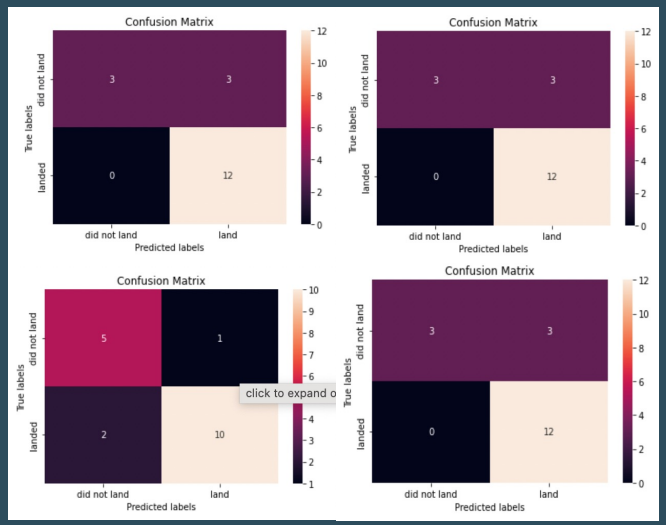
CLASSIFICATION ACCURACY USING TRAINING DATA

Model	Accuracy	TestAccuracy
LogReg	0.84643	0.83333
SVM	0.84821	0.83333
Tree	0.9	0.83333
KNN	0.84821	0.83333

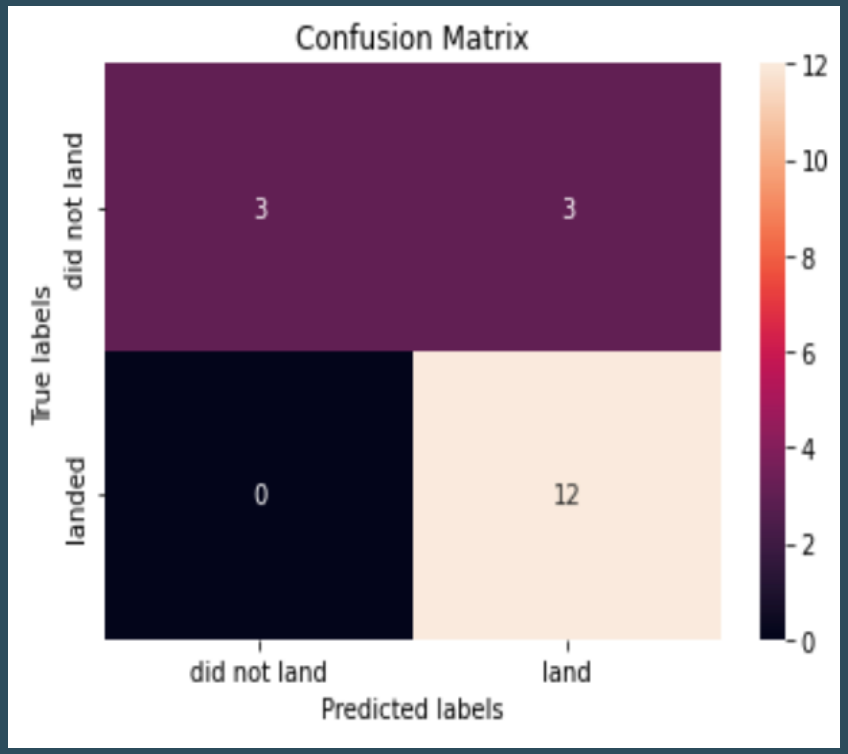


The accuracy for each model is extremely close but the Decision Tree algorithm has the high training accuracy among all

Tree distinguish between the different classes. The major problem is false positive



CLASSIFICATION ACCURACY USING TRAINING DATA



The Decision Tree has 83.33% accuracy on the test set

The Tree Classifier Algorithm is the best for Machines Learning for this dataset

Low weighted payloads performs 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

THANK YOU !