IBM DATA SCIENCE PROJECT - SPACEX

By - DIPANKAR ROY

Outline

Executive Summary

Summary of methodologies

- Data Collection
- Data wrangling
- Exploratory Data Analysis (EDA)
- Data Visualization
- EDA with SQL
- Building Interactive Map using Folium
- Building a Dashboard with Plotly Dash
- Prediction Analysis, Classification

Summary of all results

- Exploratory Data Analysis Results
- Interactive Analytics screenshots
- Predictive analysis results

Introduction

Project background and context

SpaceX’s advanced Falcon-9 rockets launches with a cost of 62 million USD whereas other provides a cost of 160 million USD each, Cause SpaceX reuse the first stage. We predict if the Falcon-9 first stage will land Successfully. If we can determine the first stage will land , we can determine the cost of launch. This model can help the other compotators as well.

Problems you want to find answers

- What influences the success of the rochet to land successfully?
- The cause and effect relationship and their impact in determining the success rate of successful landing.
- What are the conditions to met to achieve the best result and ensure best rocket landing rate.

Methodology

Introduction

Data Collection - From SpaceX API

- SpaceX Launce data gathered in SpaceX REST API.

- API Contains Launces, Rocket used, Payload delivered, Launch specifications, Landing specifications and Landing outcome information.
- Data Collection process in right section

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

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

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

```
data_falcon9 = df.loc[df['BoosterVersion']!="Falcon 1"]
```

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

Data Collection - From Wikipedia

- Falcon-9 Launce data from Wikipedia. The popular Source of Data.
- We will do web scraping to collect data from Wikipedia using BeautifulSoup.
- Process are in right side

```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"
```

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

```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(page.text, 'html.parser')
```

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

```
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
```

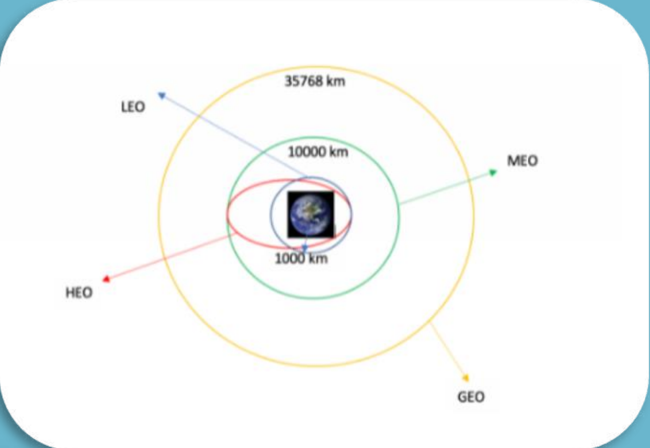
```
launch_dict= dict.fromkeys(column_names)

# Remove an irrelvant column
del launch_dict['Date and time ( )']


launch_dict['Flight No.'] = []
launch_dict['Launch site'] = []
launch_dict['Payload'] = []
launch_dict['Payload mass'] = []
launch_dict['Orbit'] = []
launch_dict['Customer'] = []
launch_dict['Launch outcome'] = []
launch_dict['Version Booster']=[]
launch_dict['Booster landing']=[]
launch_dict['Date']=[]
launch_dict['Time']=[]
```

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

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

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

Data Wrangling



Process

Perform Exploratory Data Analysis EDA on dataset

Calculate the number of launches at each site

Calculate the number and occurrence of each orbit

Calculate the number and occurrence of mission outcome per orbit type

Export dataset as .CSV

Create a landing outcome label from Outcome column

Work out success rate for every landing in dataset

GitHub Notebook Link

Exploratory data analysis

(EDA) using visualization and SQL

Data Viz: Scatter Plot

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

Data Viz: Bar and Line Chart

- Mean Vs Orbit
- Success Rate vs Year

SQL 1

- 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 achieved.

SQL 2

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. Use a subquery.
- 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 (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order

EDA with Data Visualization GitHub Notebook

EDA with SQL GitHub Notebook

Interactive visual analytics

Using Folium and Plotly Dash

Mark all launch sites on a map.

Mark the success/failed launches for each site on the map.

Calculate the distances between a launch site to its proximities.

Folium

Do launch sites keep certain distance away from cities?
- Yes

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

Dashboard

- The Dashboard is built with Flask and Dash web frame.
- Pie Chart Showing Total Launches by a certain site or all sites. Display relative Proportions of multiple classes of data.
- Scatter Plots showing the relationship with Outcome and Payload Mass(kg) for the different Booster Versions

GitHub Source Code

Predictive Analysis

Perform predictive analysis using classification models

Building Model

- Load Dataset into Numpy and Pandas
- Transform Data
- Split Data into Train Test sets.
- Decide which type of Machine Learning algorithm we want to use
- Set our algorithm to GridSearchCV.
- Fit our Dataset into GridSearchCV object and Train our Dataset.

Evaluating Model

- Check Accuracy for each model.
- Get Tuned Hyperparameters for each type of algorithm.
- Plot Confusion matrix

Improving Model

- Feature Engineering
- Algorithm Tuning

Finding Best Model

- Finding the best performing classification model.
- The model with best accuracy score selected as best performing model.

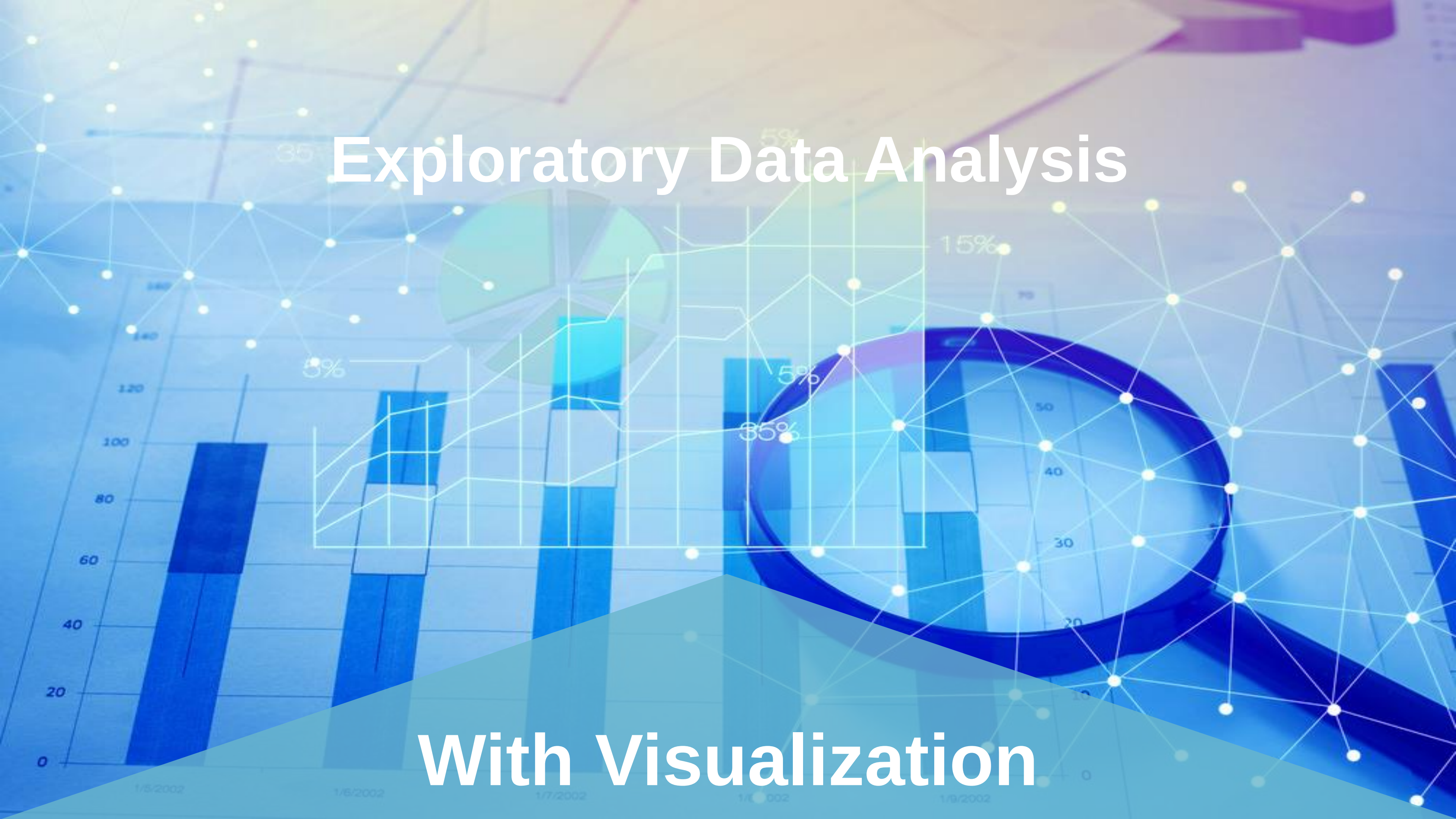
GitHub Source Code

Results

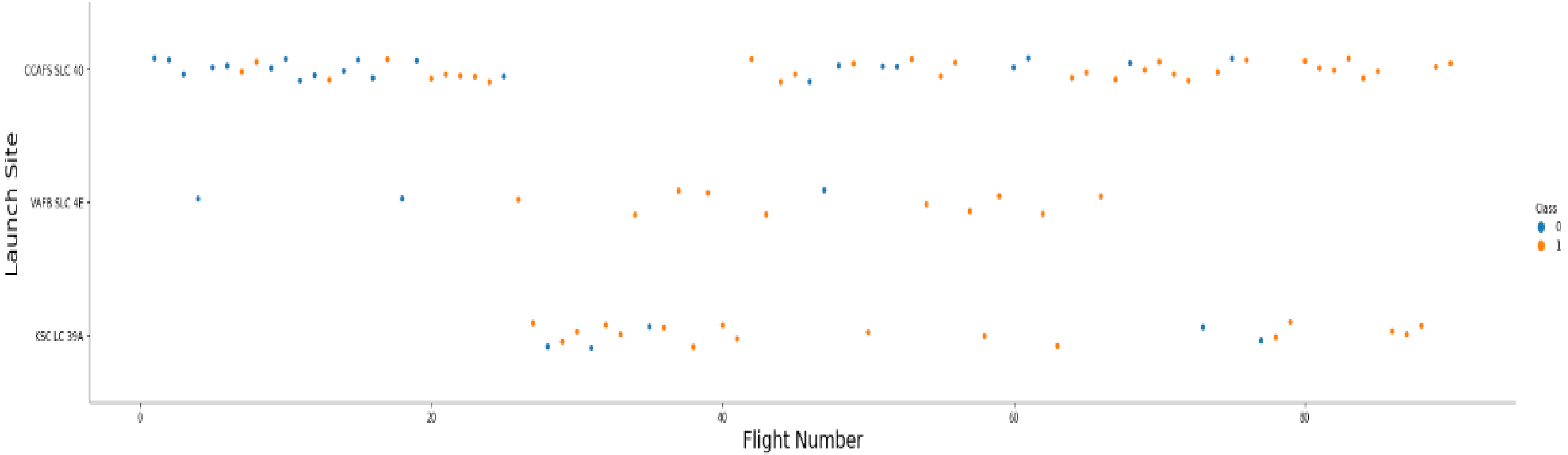
1. Exploratory data analysis results.

2. Interactive analytics demo in screenshots.

3. Predictive analysis results.

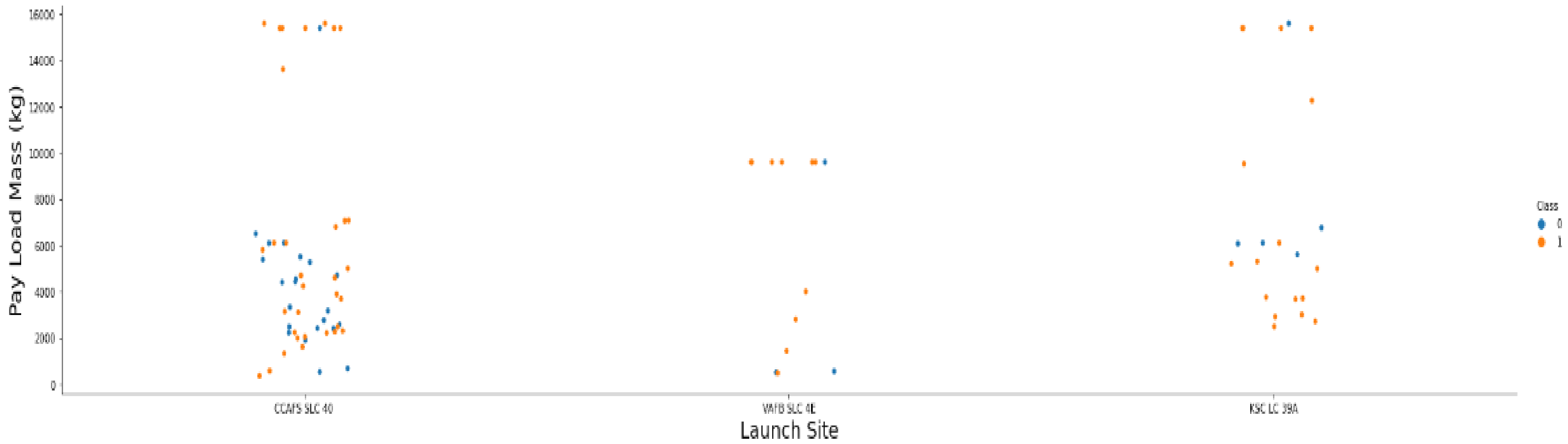

Exploratory Data Analysis
With Visualization

Flight Number vs. Launch Site



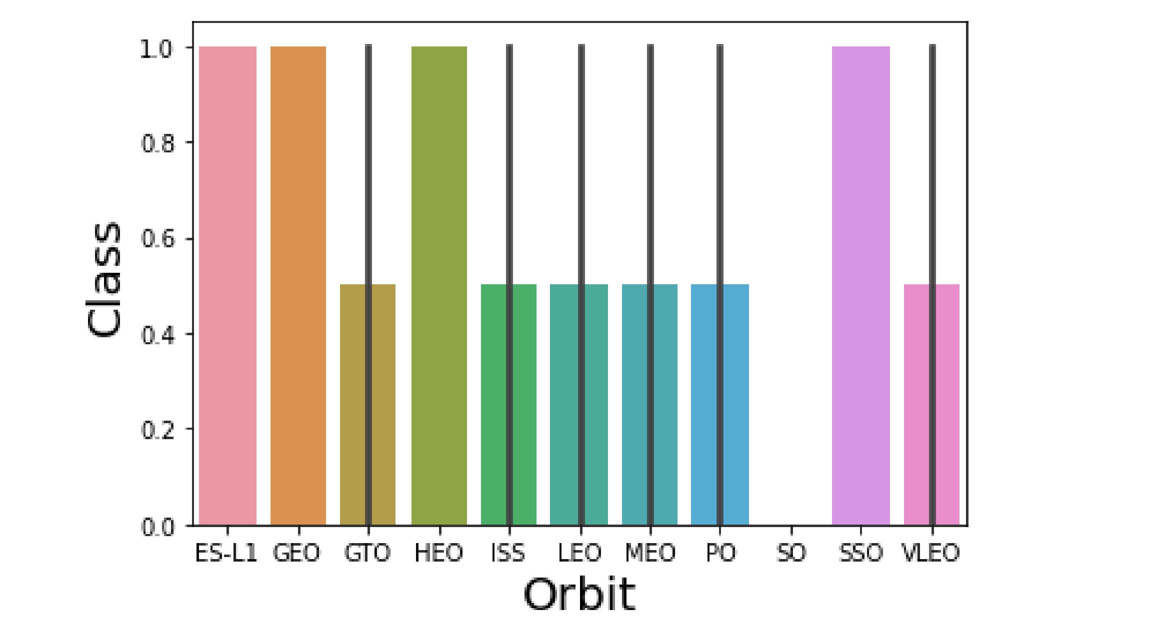
The more number of flights at a launch site the more the success rate at launch site.

Payload vs. Launch Site



The more the payload mass for Launch site CCAFS SLC 40 the more the success rate for the rocket. There is no clear relation that, If the launch site is dependent on pay load mass for success rate.

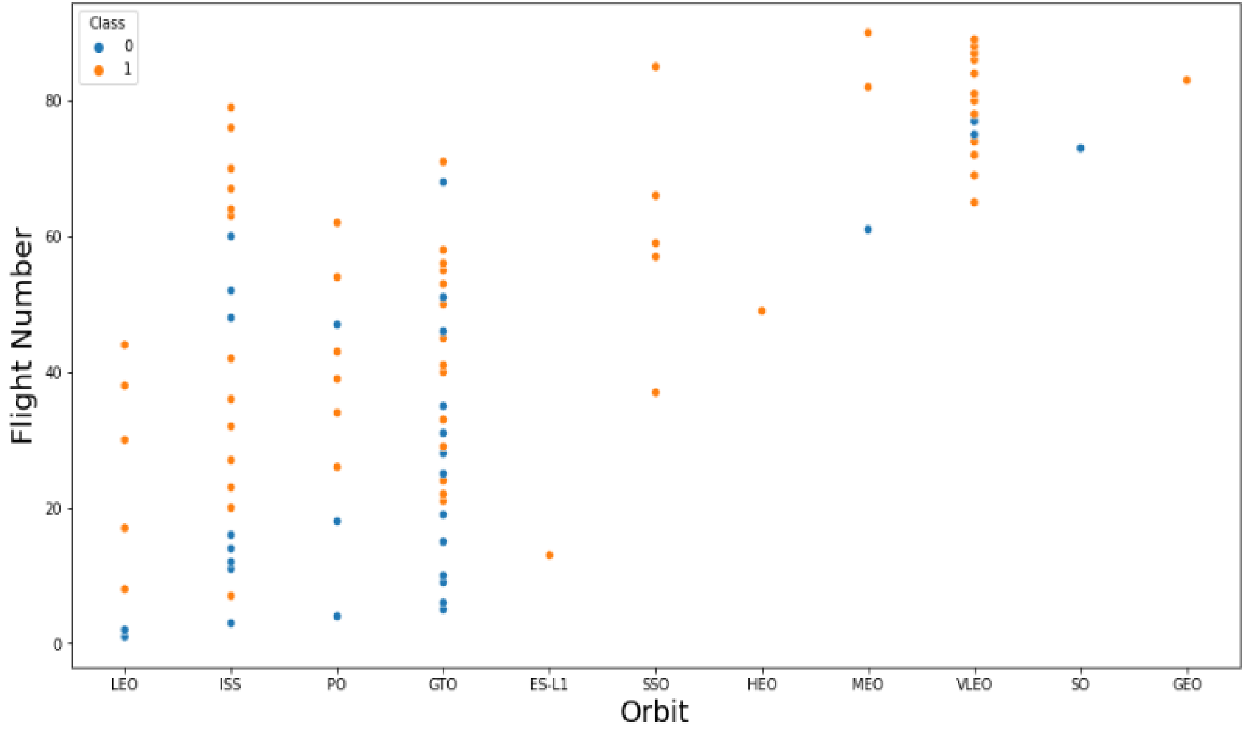
Success rate vs Orbit Type



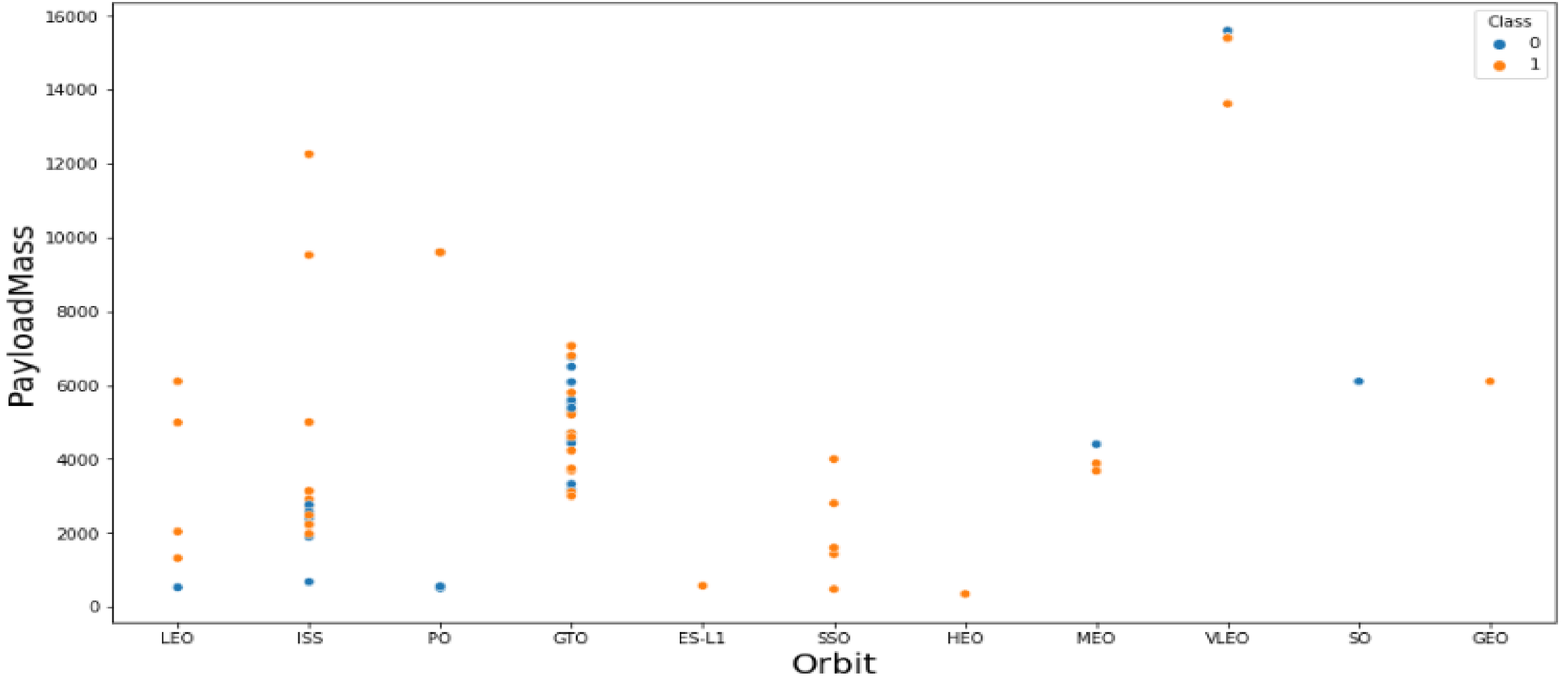
Orbit GEO, HEO, SSO, ES-L1 has the best success rate.

Flight Number vs Orbit Type

- In LEO orbit the su ccess appears relat ed to the number of flights, on the other hand it seems ther e is no relationship between flight numb er in GTO orbit.

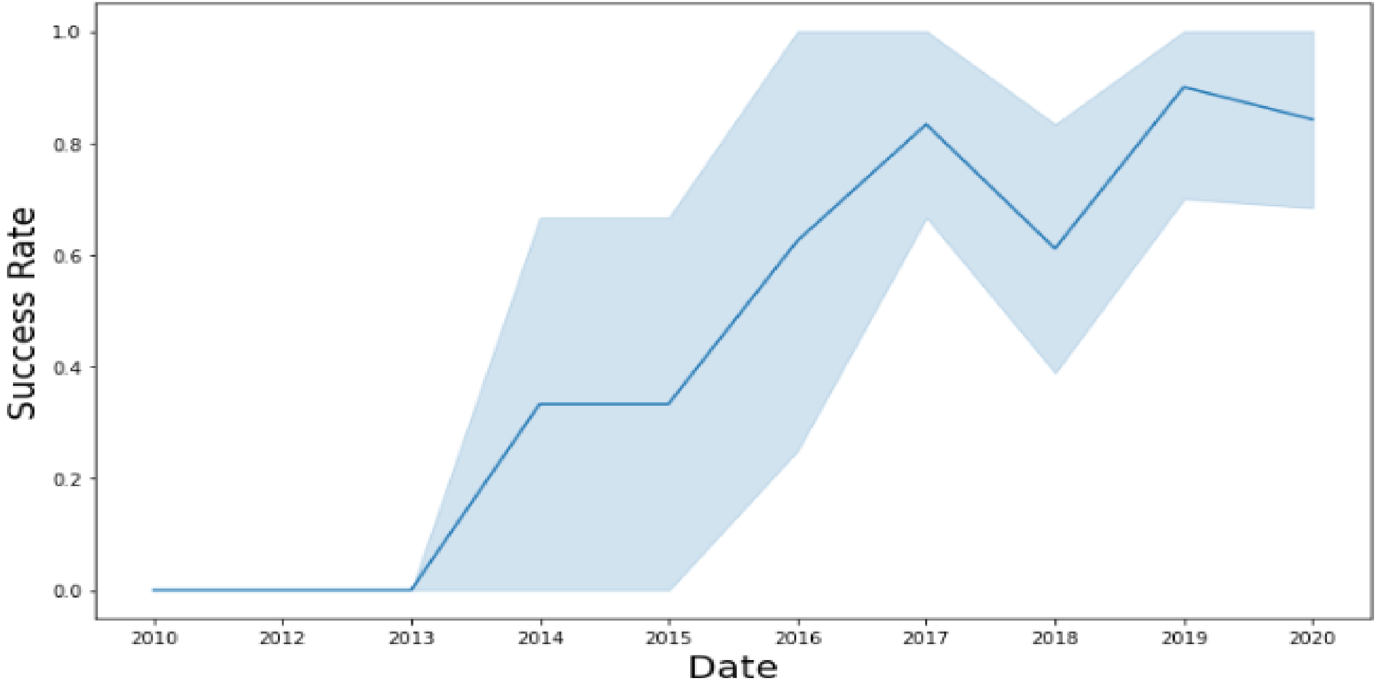


Payload vs Orbit Type



Heavy Payloads have a negative influence on GTO orbits and positive on GTO an d polar LEO orbit.

Launch Success Yearly Trend



you can observe that the success rate since 2013 kept increasing till 2020

Exploratory Data Analysis

With SQL

All Launch Site Names

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

Launch Sites
CCAFS LC-40
CCAFS SLC-40
KSC LC-39A
VAFB SLC-4E

QUERY

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

QUERY explained

- Using DISTINCT in the query above will show the Unique launch_site from SPACEXTBL.

Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with `CCA`

DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__outcome
2010-04-06	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-08-12	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)
2012-08-10	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-01-03	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-12	22:41:00	F9 v1.1	CCAFS LC-40	SES-8	3170	GTO	SES	Success	No attempt

QUERY explained

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

- The query will filter the launch site begins with CCA from SPACEXTBL, and limit 5 will show top 5 results.

Total Payload Mass

Calculate the total payload carried by boosters from NASA

Total Payload Mass
22007

QUERY explained

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

- The query will return the total Sum of payload mass in kg from SPACEXTBL only for NASA.

Average Payload Mass by F9 v1.1

Calculate the average payload mass carried by booster version F9 v1.1

Total Payload Mass
3676

QUERY explained

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

- The query will return the total Sum of payload mass in kg that carried by Falcon-9 with booster version 1.1 from SPACEXTBL.

First Successful Ground Landing Date

Find the dates of the first successful landing outcome on ground pad

Success Ground pad
2017-01-05

QUERY explained

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

- The query will return the first successful landing outcome on ground pad from SPACEXTBL. Min(DATE) shows the earliest date, That means the first.

Successful Drone Ship Landing with Payload between 4000 and 6000

List the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000.

booster_version
F9 FT B1022
F9 FT B1031.2

QUERY explained

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

- The query will return names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000 from SPACEXTBL.

Total Number of Successful and Failure Mission Outcomes

total number of successful and failure mission outcomes

Mission Outcomes
44

QUERY explained

```
%sql select count(MISSION_OUTCOME) from SPACEXTBL where MISSION_OUTCOME = 'Success' or MISSION_OUTCOME = 'Failure (in flight)'
```

- The query will return names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000 from SPACEXTBL.

Boosters Carried Maximum Payload

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

booster_version
F9 B5 B1048.4
F9 B5 B1049.4
F9 B5 B1049.5
F9 B5 B1060.2
F9 B5 B1058.3

QUERY explained

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

- The query will return List of names of the booster which have carried the maximum payload mass. from SPACEXTBL.

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

DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__outcome
2017-03-06	21:07:00	F9 FT B1035.1	KSC LC-39A	SpaceX CRS-11	2708	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
2017-01-05	11:15:00	F9 FT B1032.1	KSC LC-39A	NROL-76	5300	LEO	NRO	Success	Success (ground pad)
2016-08-04	20:43:00	F9 FT B1021.1	CCAFS LC-40	SpaceX CRS-8	3136	LEO (ISS)	NASA (CRS)	Success	Success (drone ship)
2016-06-05	05:21:00	F9 FT B1022	CCAFS LC-40	JCSAT-14	4696	GTO	SKY Perfect JSAT Group	Success	Success (drone ship)

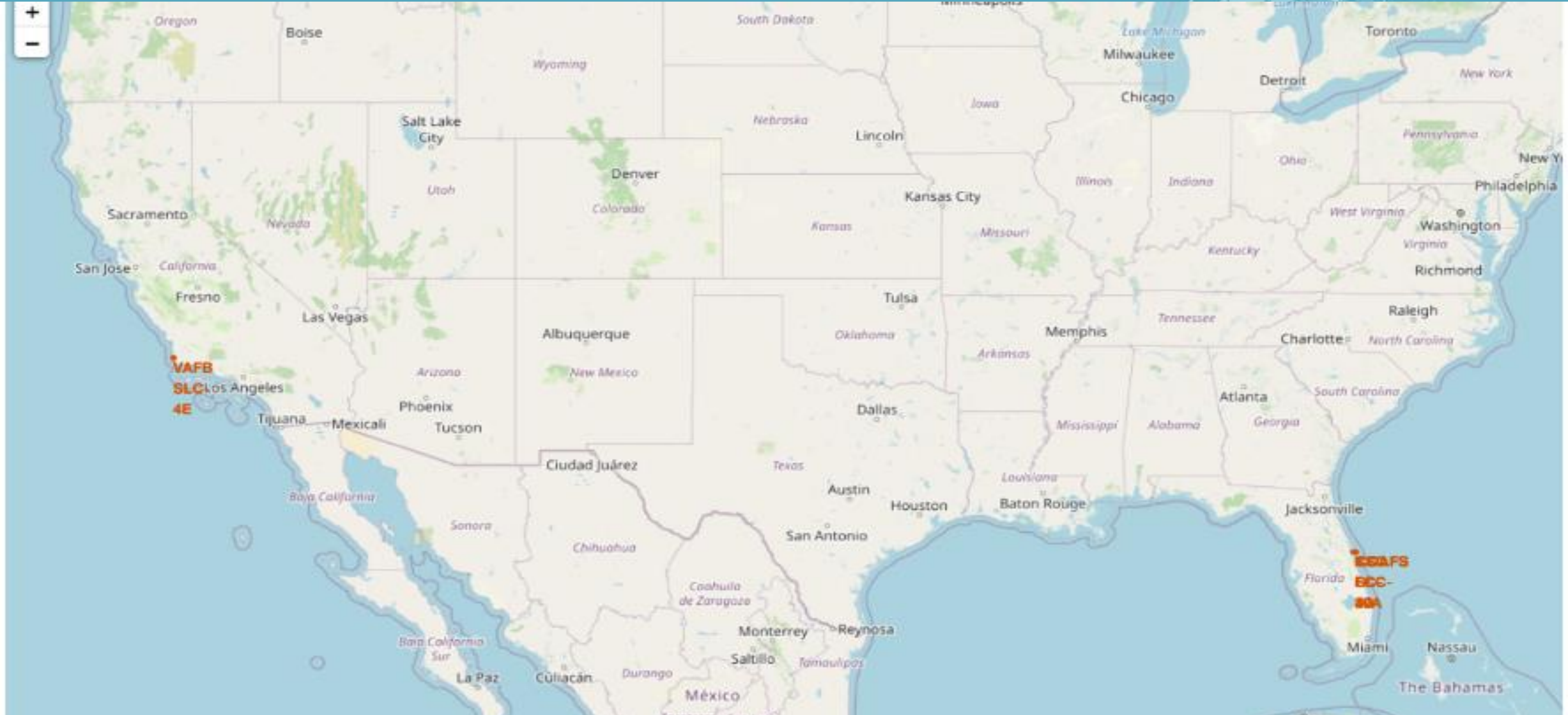
QUERY explained

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

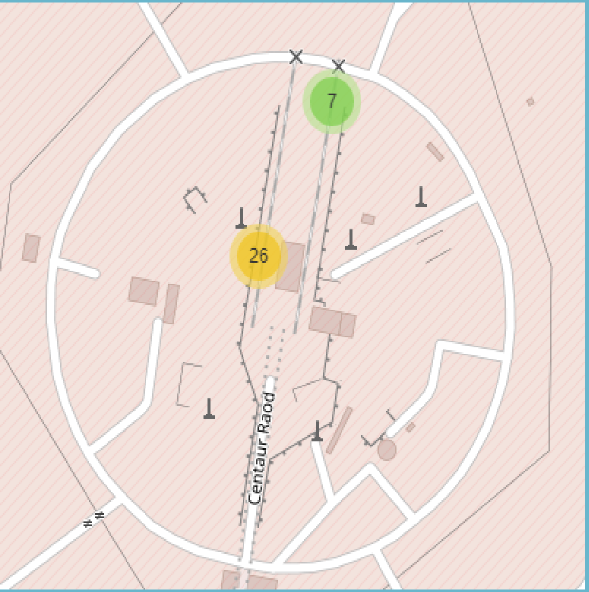
- **Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order**

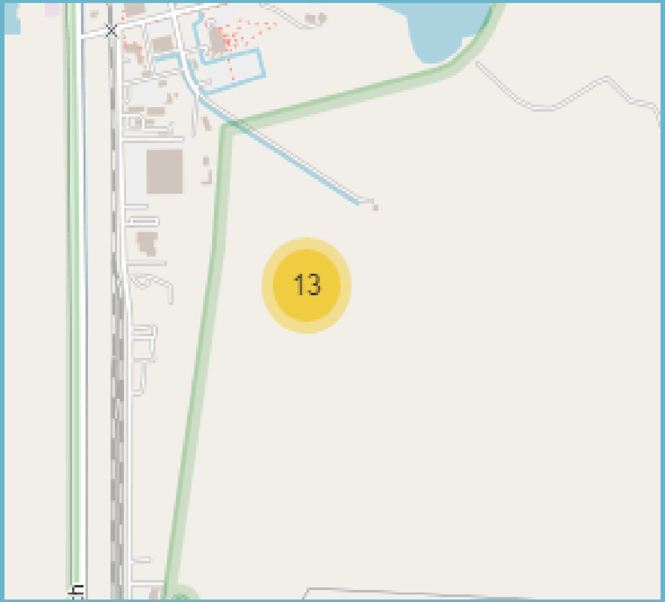
Launch Sites Proximities Analysis

All launch sites Marker

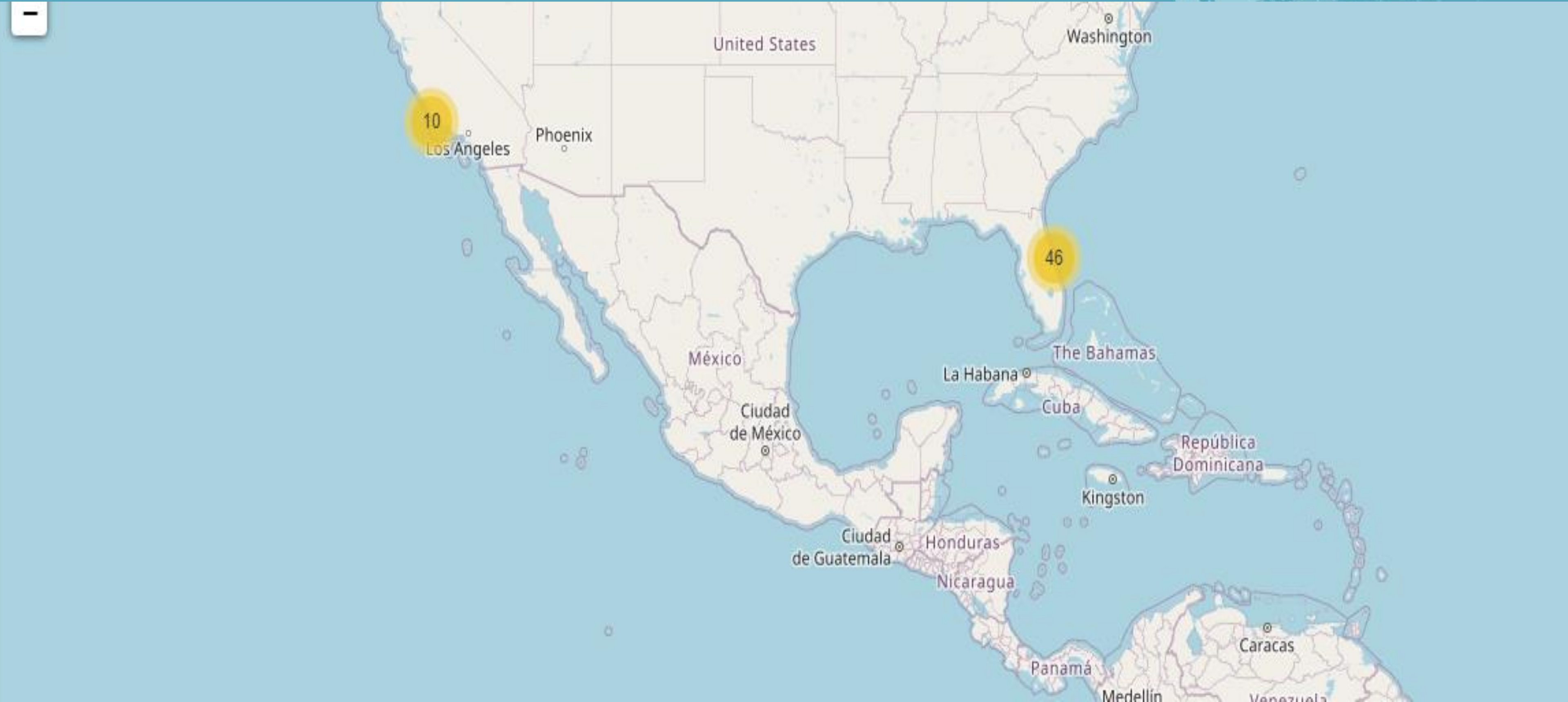


Success/failed launches for each site on the map

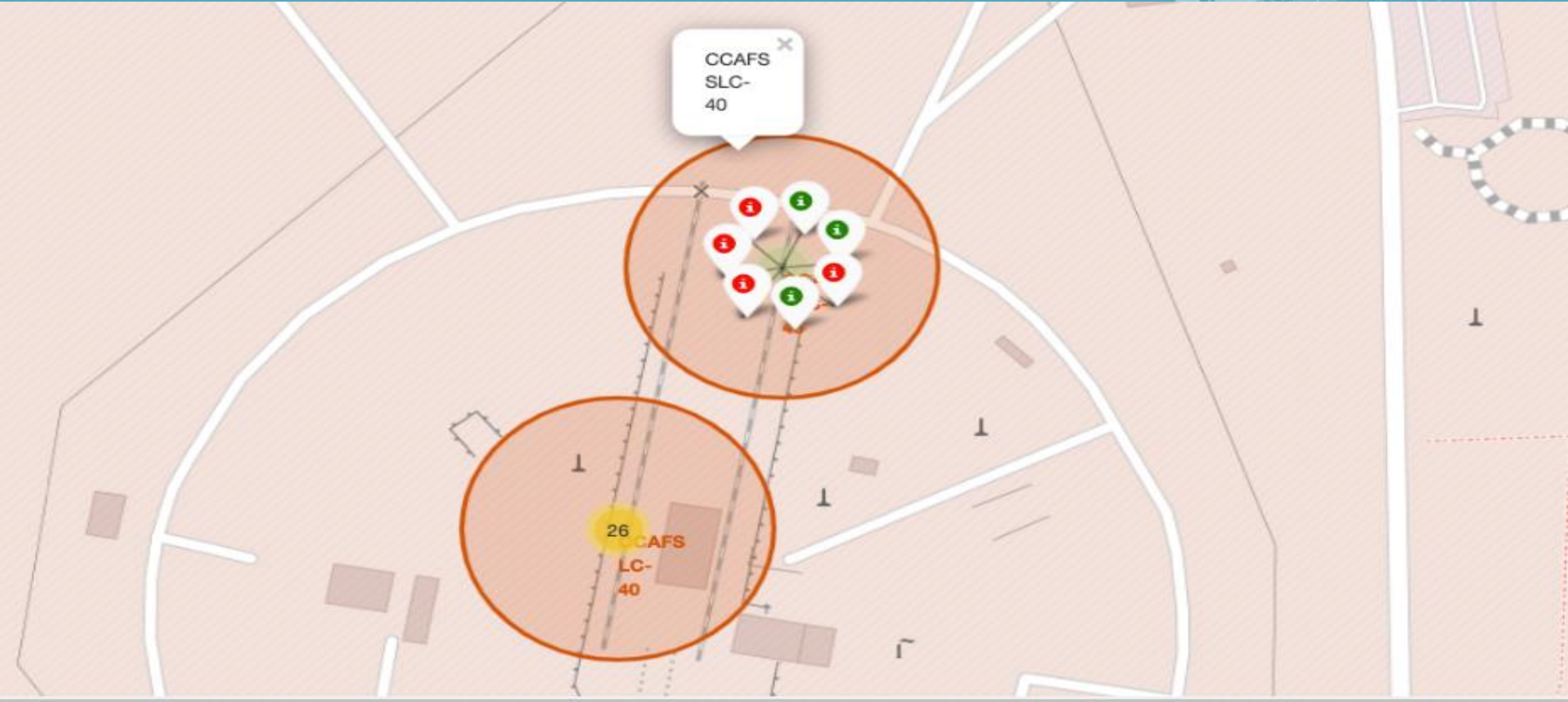




Continue from previous slide

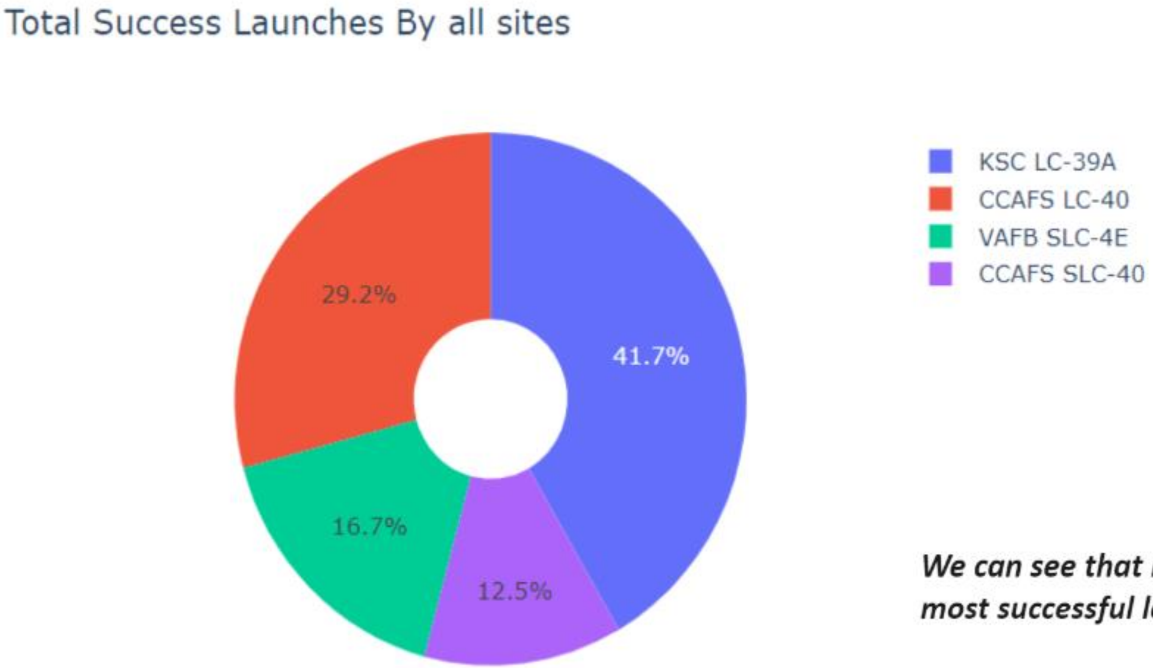


Continue from previous slide



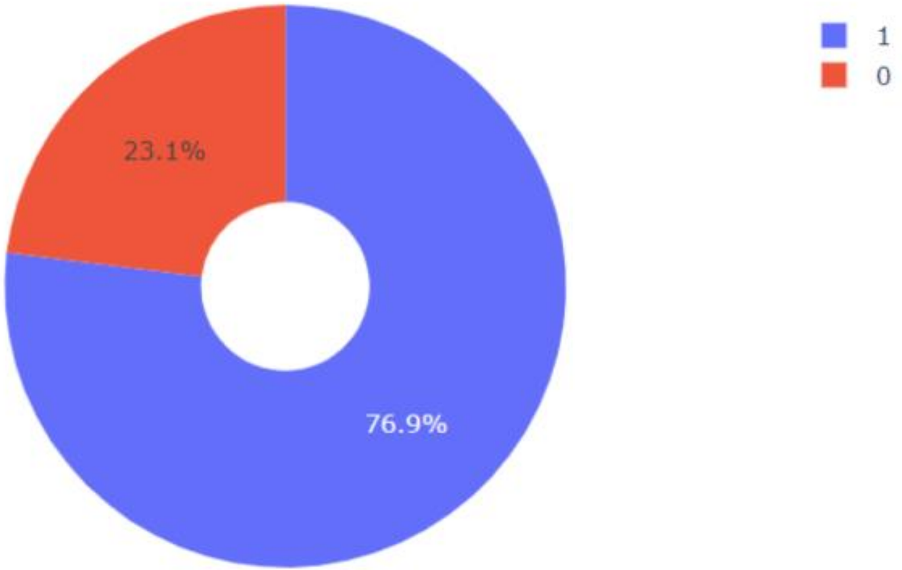
Dashboard with Plotly Dash

Pie chart for the launch



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

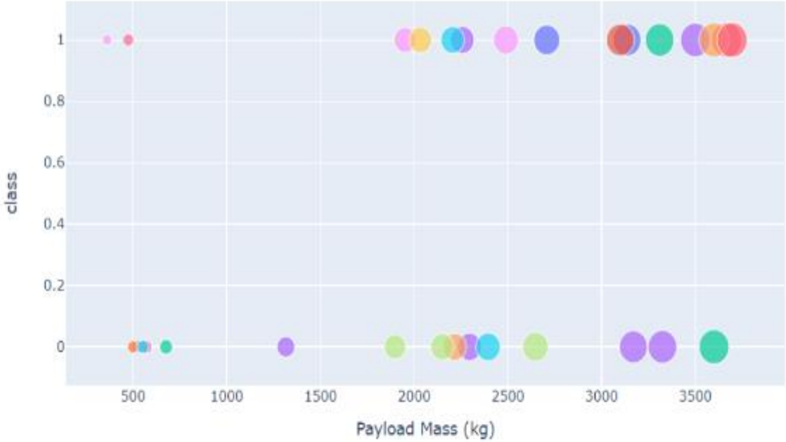
Pie chart for highest launch success ratio



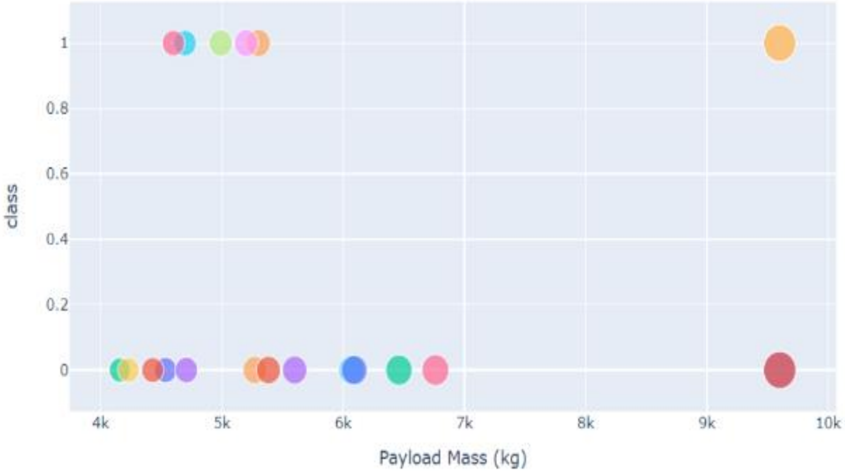
KSC LC-39A achieved a 76.9% success rate while getting a 23.1% failure rate

Payload vs. Launch Outcome scatter plot

Low Weighted Payload 0kg – 4000kg



Heavy Weighted Payload 4000kg – 10000kg



We can see the success rates for low weighted payloads is higher than the heavy weighted payloads

Predictive Analysis

Classification Accuracy

- So Tree Algorithm provides the best Score with 0.87

```
algorithms = {'KNN':knn_cv.best_score_,'Tree':tree_cv.best_score_,'LogisticRegression':logreg_cv.best_score_}
algorithms
```

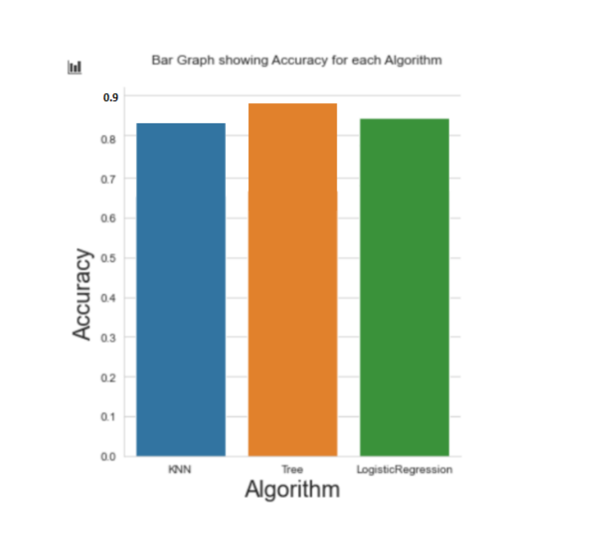
```
1]: {'KNN': 0.8482142857142858,
 'Tree': 0.8767857142857143,
 'LogisticRegression': 0.8464285714285713}
```

```
bestalgorithm = max(algorithms, key=algorithms.get)
```

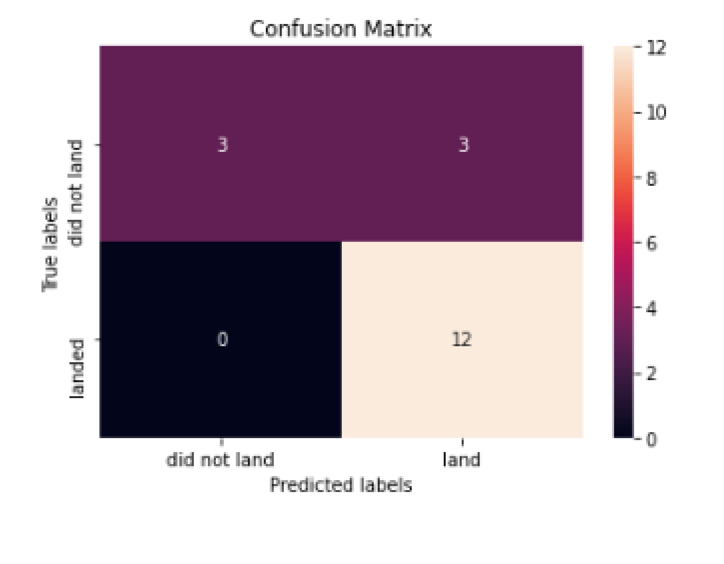
```
Best Algorithm is Tree with a score of 0.8767857142857143
Best Params is : {'criterion': 'entropy', 'max_depth': 10, 'max_features': 'sqrt', 'min_samples_leaf': 4, 'min_samples_split': 10, 'splitter': 'random'}
```

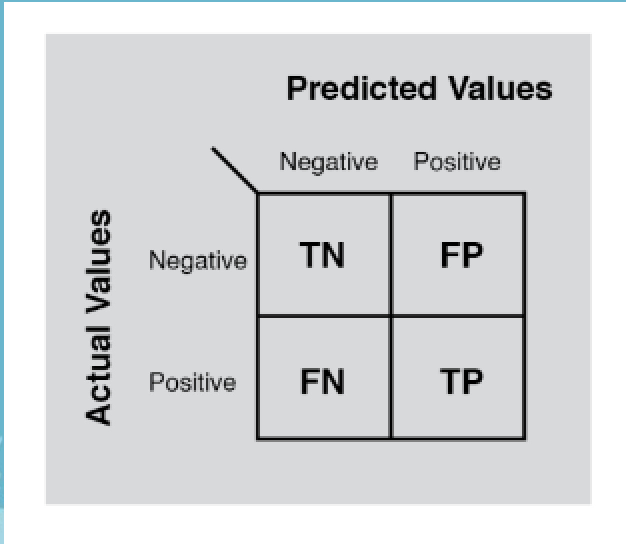
```
print("tuned hpyerparameters :(best parameters) ",tree_cv.best_params_)
print("accuracy :",tree_cv.best_score_)
```

```
tuned hpyerparameters :(best parameters)  {'criterion': 'entropy', 'max_depth': 10, 'max_features': 'sqrt', 'min_samples_leaf': 4, 'min_samples_split': 10, 'splitter':
'random'}
accuracy : 0.8767857142857143
```



Confusion Matrix for Tree





Conclusion

- Tree Classifier Machine learning Algorithm is the best for this Dataset.

- Success rate of SpaceX launches is directly proportional to Time in years. Obviously with time they perfect their launches

- KSC LC -39 A has the most successful launches from all sites.

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

Source Code on GitHub

Thank You