IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Nidhin P

17/01/2022

Outline

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

Executive Summary

- **Summary of methodologies**
- Data Collection
- Data Wrangling
- EDA with SQL
- EDA with Data Visualization
- Building an inttractive map with Folium
- Building a Dashboard with Plotly Dash
- Predictive analysis (Classification)

- **Summary of all results**
- Exploratory data analysis results
- Interactive analytics Screen shots
- Predictive analysis results

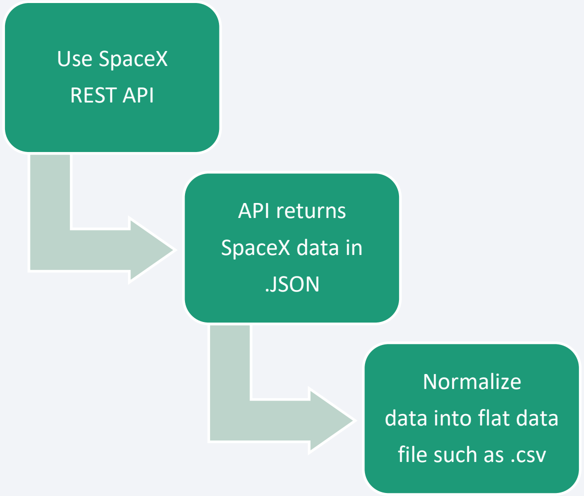
Section 1

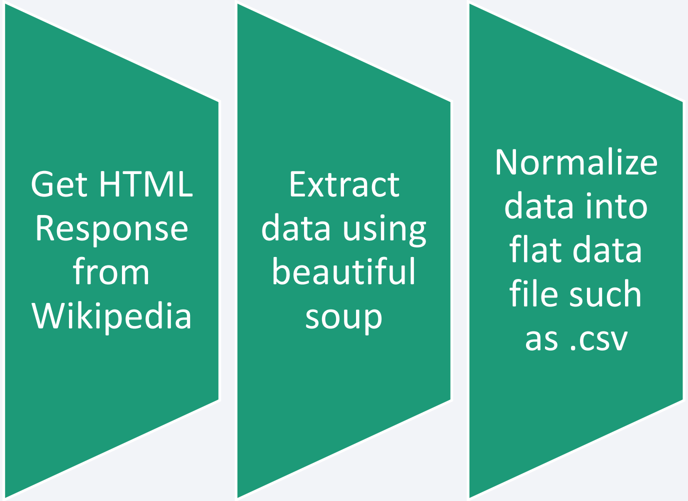
Methodology

Data Collection

The following datasets was collected by,

- Worked with SpaceX launch data that is gathered from the SpaceX REST API.
- This API will give us data about launches, including information about the rocket used, payload delivered, launch specifications, landing specifications, and landing outcome.
- Our goal is to use this data to predict whether SpaceX will attempt to land a rocket or not.
- Another popular data source for obtaining Falcon 9 Launch data is web scraping Wikipedia using BeautifulSoup.





Data Collection – SpaceX API

GitHub url

1 .Getting Response from API

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

2. Converting Response to a .json file

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

3. Apply custom functions to clean data

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

```
getBoosterVersion(data)
```

4. 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}
```

5. Filter dataframe and export to flat file (.csv)

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

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

Data Collection - Scraping

1 .Getting Response from HTML

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

2. Creating BeautifulSoup Object

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

3. Finding tables

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

4. 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
```

5. Creation of dictionary

```
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']=[]
```

[Github url]

6. Appending data to keys (refer) to notebook block 12

```
extracted_row = 0
#Extract each table
for table_number,table in enumerate(
   # get table row
    for rows in table.find_all("tr")
        #check to see if first table
```

7. Converting dictionary to dataframe

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

8. Dataframe to .CSV

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

Data Wrangling

Introduction

In the data set, there are several different cases where the booster did not land successfully. Sometimes a landing was attempted but failed due to an accident; for example, True Ocean means the mission outcome was successfully landed to a specific region of the ocean while False Oceanmeans the mission outcome was unsuccessfully landed to a specific region of the ocean. True RTLS means the mission outcome was successfully landed to a ground pad False RTLS means the mission outcome was unsuccessfully landed to a ground pad. True ASDS means the mission outcome was successfully landed on a drone ship False ASDS means the mission outcome was unsuccessfully landed on a drone ship.

We convert those outcomes into Training Labels with1means the booster successfully landed0means it was unsuccessful.

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 url]()

EDA with Data Visualization

Scatter Graphs being drawn:

- 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

Scatter plots showhow much one variable is affected by another. The relationship between two variables is called their correlation .Scatter plotsusually consist of a large body ofdata.

Bar Graph being drawn:

- Mean VS. Orbit

Line Graph being drawn:

- Success RateVS. Year

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

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.Barcharts can also show big changes in data over time.

Github url

EDA with SQL

Performed SQL queries to gather information about the dataset.

For example, some questions asked about the data we needed information about. Which we are using SQL queries to get the answers in the dataset :

- **Displaying the names of the unique launch sites in the space mission**
- **Displaying 5 records where launch sites begin with the string 'KSC’**
- **Displaying the total payload mass carried by boosters launched by NASA (CRS)**
- **Displaying average payload mass carried by booster version F9 v1.1**
- **Listing the date where the successful landing outcome in drone ship was achieved.**
- **Listing the names of the boosters which have success in ground pad and have payload mass greater than 4000 but less than 6000**
- **Listing the total number of successful and failure mission outcomes**
- **Listing the names of the booster_versions which have carried the maximum payload mass.**
- **Listing the records which will display the month names, successful landing_outcomesin ground pad booster versions, launch_sitefor the months in year 2017**
- **Ranking the count of successful landing_outcomesbetween the date 2010-06-04 and 2017-03-20 in descending order.**

[Github url]

Build an Interactive Map with Folium

To visualize the Launch Data into an interactive map. We took the Latitude and Longitude Coordinates at each launch site and added a *Circle Marker around each launch site with a label of the name of the launch site.*

We assigned the dataframe launch_outcomes(failures, successes) to *classes 0 and 1*with Greenand Redmarkers on the map in a MarkerCluster()

Using Haversine's formula we 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]

Build a Dashboard with Plotly Dash

Used Python Any where to host the 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.

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.

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]

[Link to live web]

Predictive Analysis (Classification)

BUILDING MODEL

- **L**oad 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 datasets into the GridSearchCVobjects 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.

Github url

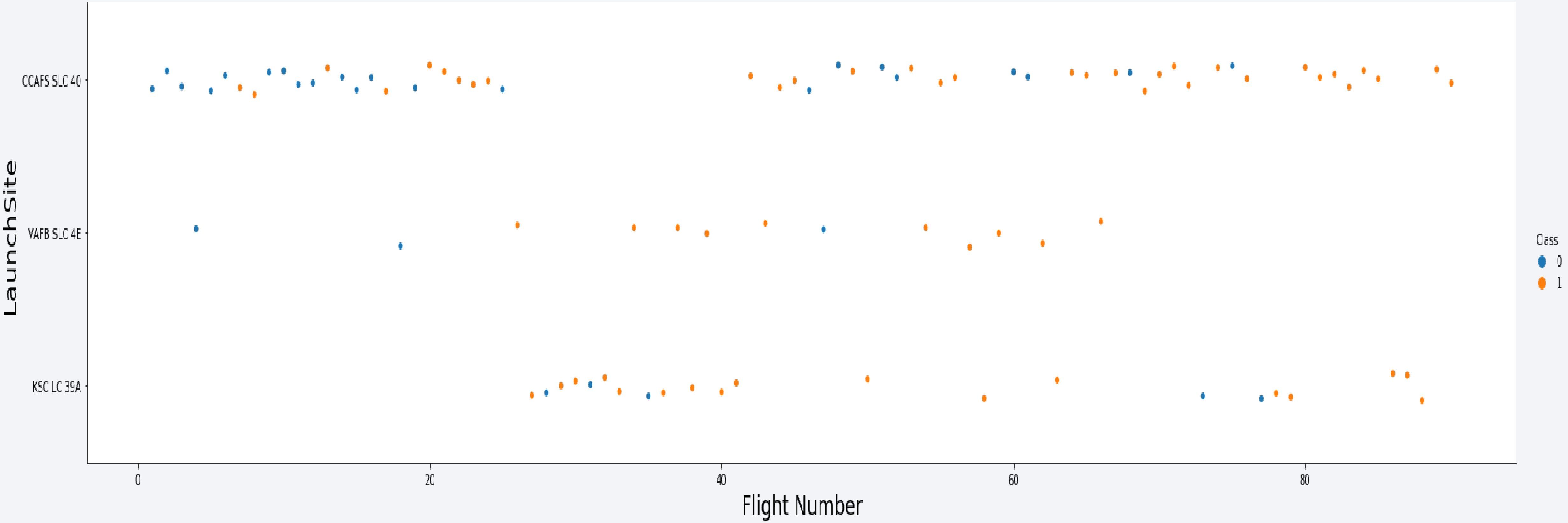
Results

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

Section 2

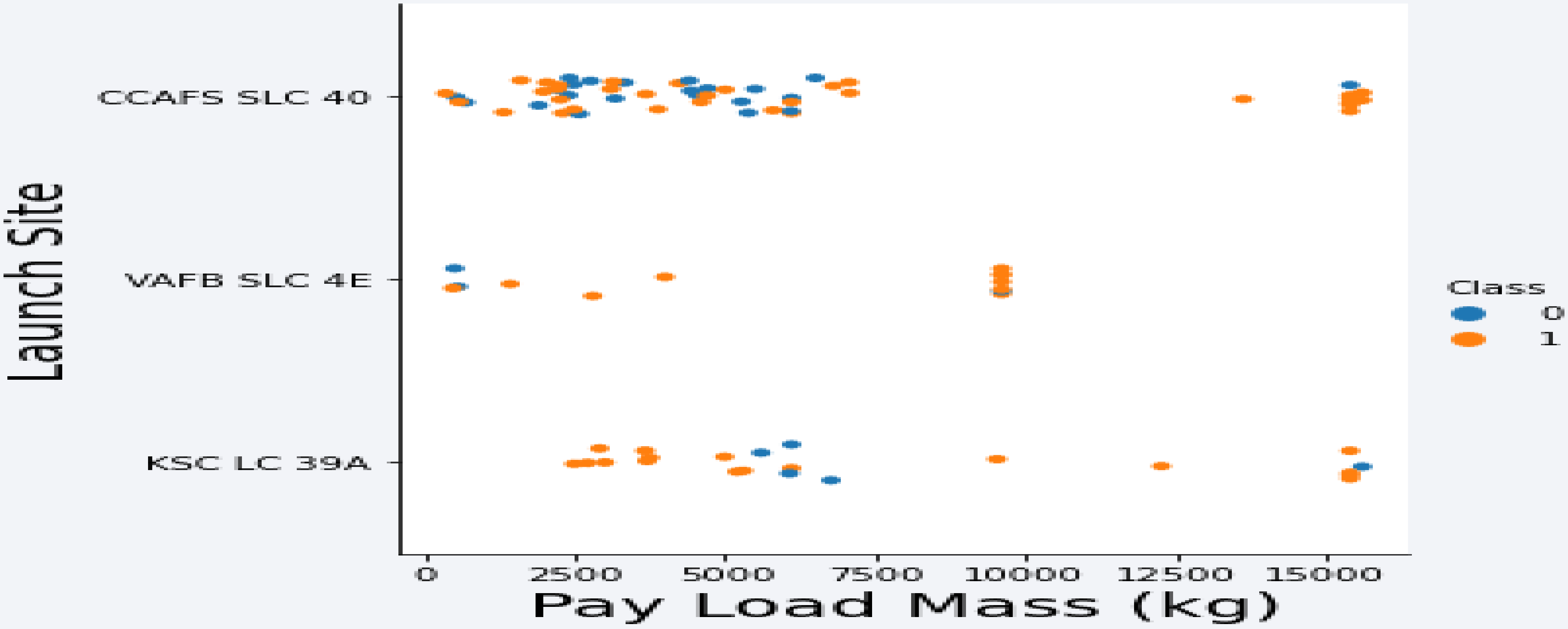
Insights drawn from EDA

Flight Number vs. Launch Site



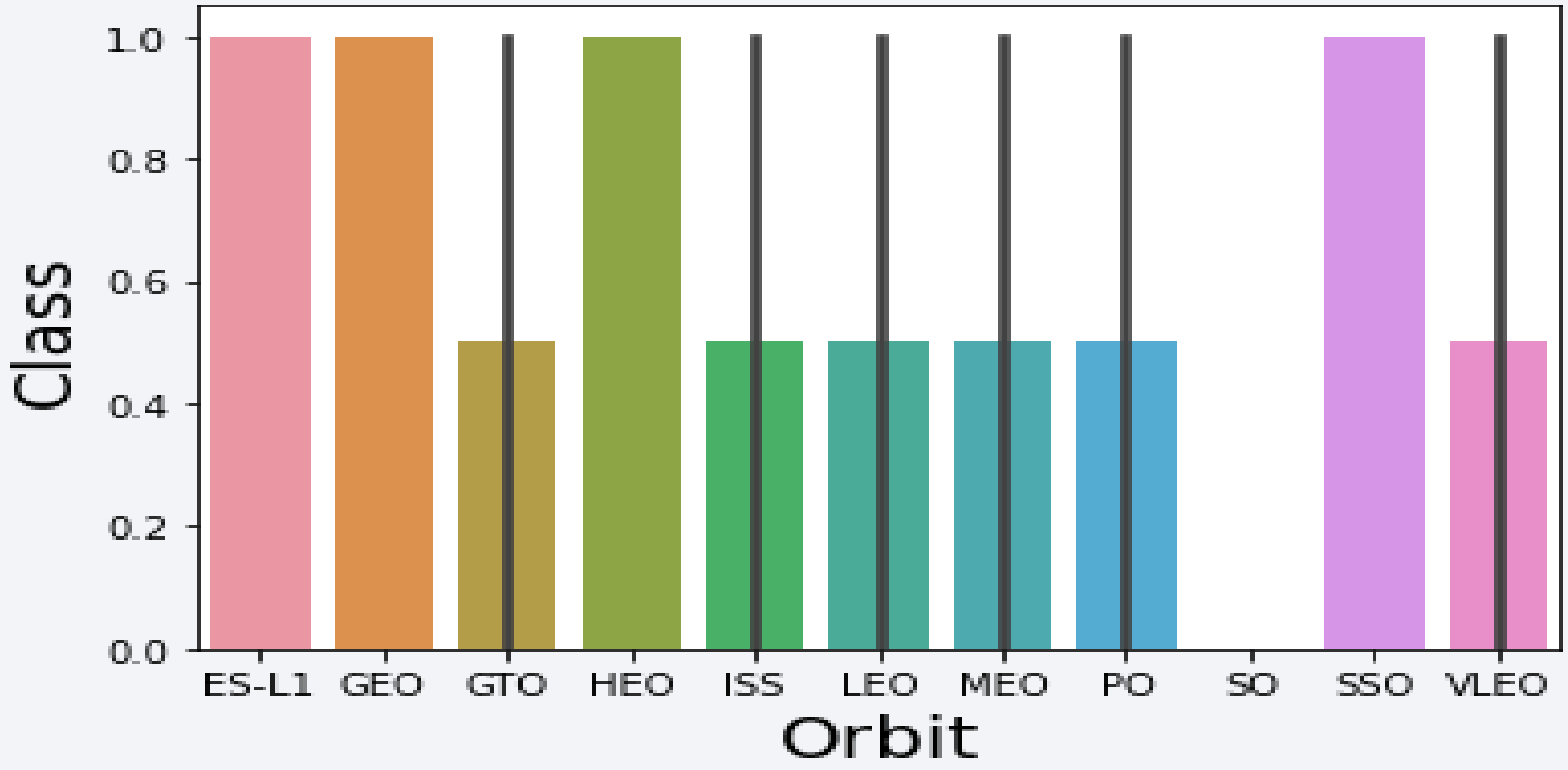
✓ *The more amount of flights at a launch site the greater the success rate at a launch site.*

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 dependant on Pay Load Mass for a success launch.*

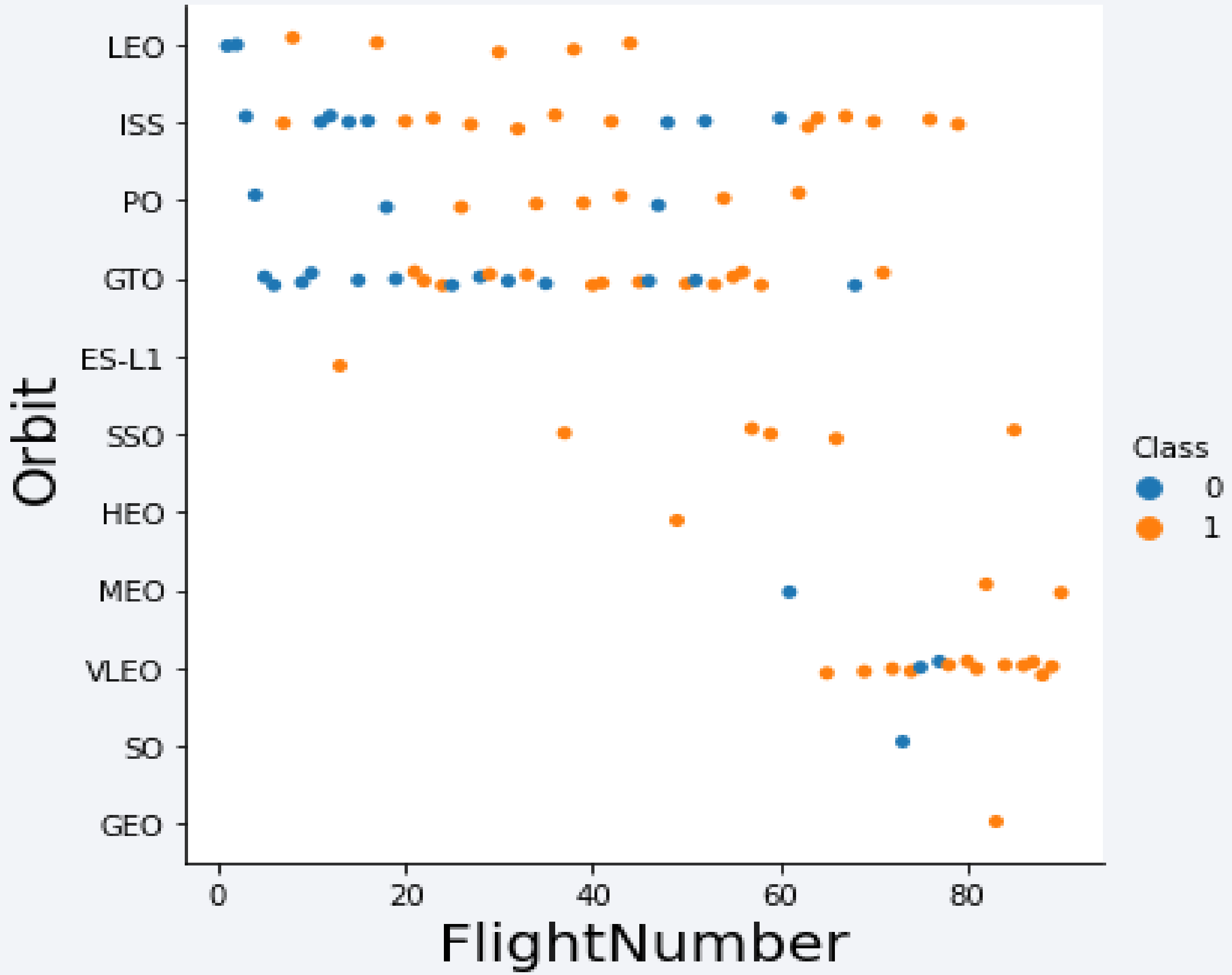
Success Rate vs. Orbit Type



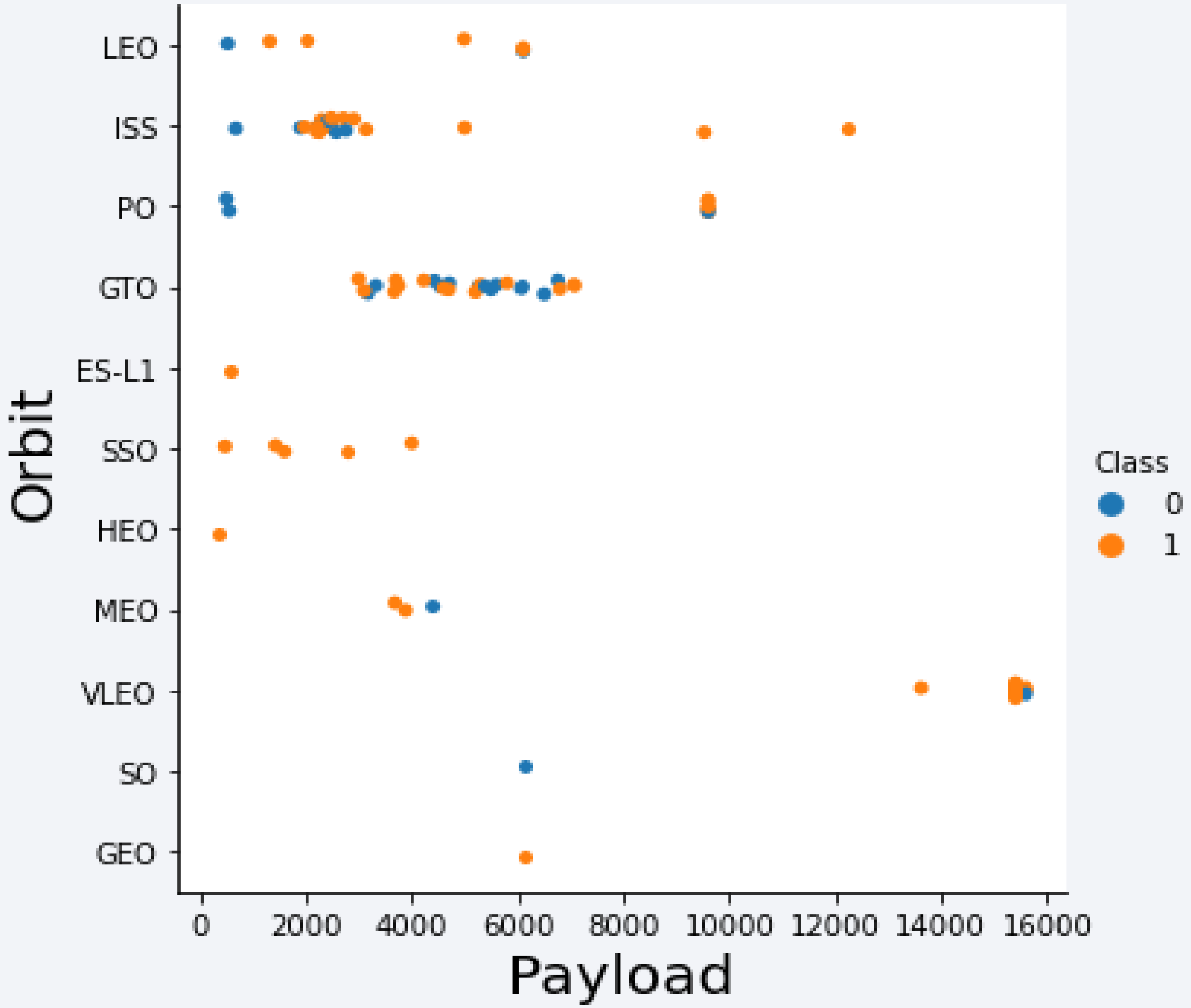
✓ *Orbits GEO, HEO, SSO and ES-L1 have the best Success Rate*

Flight Number vs. Orbit Type

✓ *You should see that 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.*

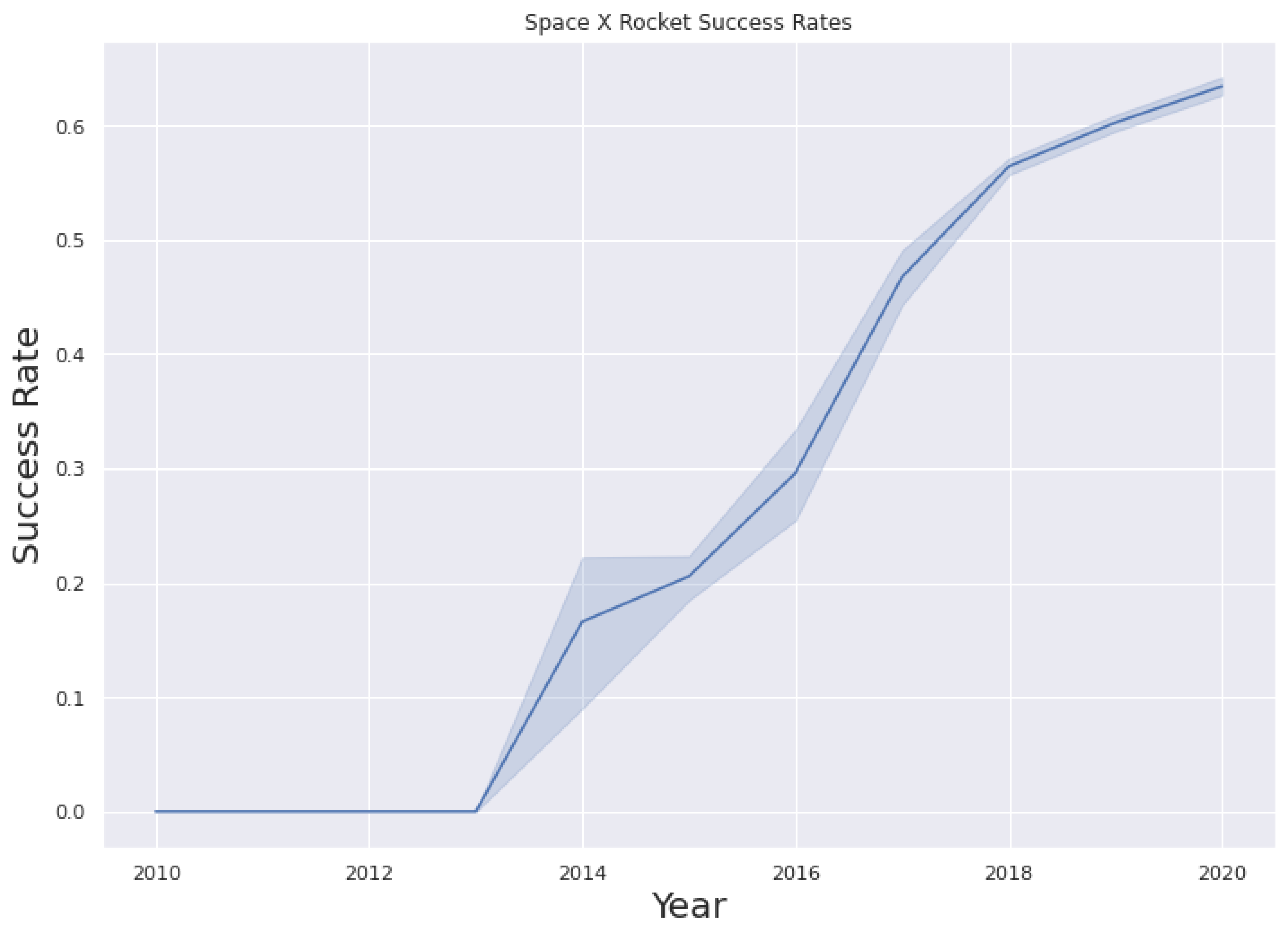


Payload vs. Orbit Type



- ✓ *Heavy payloads have a negative influence on GTO orbits and positive on GEO, LEO and ISS orbits.*

Launch Success Yearly Trend



✓ *The success rate since 2013 kept increasing till 2020*

All Launch Site Names

SQL QUERY:

select distinct(LAUNCH_SITE) from SPACEX;

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

QUERY EXPLAINATION

Using the word ***DISTINCT*** in the query means that it will only show Unique values in the ***Launch_Site*** column from Table ***SpaceX.***

Launch Site Names Begin with 'CCA'

SQL QUERY

select * from SPACEX where LAUNCH_SITE like 'CCA%' limit 5;

QUERY EXPLAINATION

*Using the word **LIMIT 5** in the query means that it will only show 5 records from table **SpaceX** and **LIKE** keyword has a wild card with the word **'CCA%'** the percentage in the end suggests that the Launch_Site name must start with **CCA**.*

DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-05-22	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

SQL QUERY

select sum(PAYLOAD_MASS__KG_) from SPACEX where CUSTOMER = 'NASA (CRS)'

1
45596

QUERY EXPLAINATION

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)***

Average Payload Mass by F9 v1.1

SQL QUERY

```
select avg(PAYLOAD_MASS__KG_) from SPACEX where BOOSTER_VERSION = 'F9 v1.1'
```

1
2928

QUERY EXPLAINATION

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***

First Successful Ground Landing Date

SQL QUERY

```
select min(DATE) from SPACEX where Landing_Outcome = 'Success (ground pad)'
```

1
2015-12-22

QUERY EXPLAINATION

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)***

Successful Drone Ship Landing with Payload between 4000 and 6000

SQL QUERY

select BOOSTER_VERSION from SPACEX where Landing__Outcome = 'Success (drone ship)' and PAYLOAD_MASS__KG_ > 4000 and PAYLOAD_MASS__KG_ < 6000

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

QUERY EXPLAINATION

- 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***

Total Number of Successful and Failure Mission Outcomes

SQL QUERY

select count(MISSION_OUTCOME) from SPACEX where MISSION_OUTCOME = 'Success' or MISSION_OUTCOME = 'Failure (in flight)'

1
100

QUERY EXPLAINATION

The **COUNT** clause counts the number of **MISSION_OUTCOME**

The ***WHERE*** clause filters the dataset to **MISSION_OUTCOME** ***= Success or Failure***

Boosters Carried Maximum Payload

SQL QUERY

select BOOSTER_VERSION from SPACEX where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEX)

booster_version
F9 B5 B1048.4
F9 B5 B1049.4
F9 B5 B1051.3
F9 B5 B1056.4
F9 B5 B1048.5
F9 B5 B1051.4
F9 B5 B1049.5
F9 B5 B1060.2
F9 B5 B1058.3
F9 B5 B1051.6
F9 B5 B1060.3
F9 B5 B1049.7

QUERY EXPLANATION

- Used subqueries to obtain the results
- Here the value for WHERE clause is used as a subquery
- MAX function returns the maximum value in PAYLOAD_MASS_KG

2015 Launch Records

SQL QUERY

```
SELECT MONTH(DATE),MISSION_OUTCOME,BOOSTER_VERSION,LAUNCH_SITE FROM SPACEX
where EXTRACT(YEAR FROM DATE)='2015';
```

1	mission_outcome	booster_version	launch_site
1	Success	F9 v1.1 B1012	CCAFS LC-40
2	Success	F9 v1.1 B1013	CCAFS LC-40
3	Success	F9 v1.1 B1014	CCAFS LC-40
4	Success	F9 v1.1 B1015	CCAFS LC-40
4	Success	F9 v1.1 B1016	CCAFS LC-40
6	Failure (in flight)	F9 v1.1 B1018	CCAFS LC-40
12	Success	F9 FT B1019	CCAFS LC-40

QUERY EXPLANATION

- The function MONTH() select the month from the year
- WHERE clause filters the result
- EXTRACT() function extract the year from the date

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

SQL QUERY

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

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	05:26:00	F9 FT B1026	CCAFS LC-40	JCSAT-16	4600	GTO	SKY Perfect JSAT Group	Success	Success (drone ship)
2016-07-18	04:45:00	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	05:21:00	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	01:29:00	F9 FT B1019	CCAFS LC-40	OG2 Mission 2 11 Orbcomm-OG2 satellites	2034	LEO	Orbcomm	Success	Success (ground pad)

QUERY EXPLANATION

- The like clause has the value Success and the percentage restrict that the value shoul start with Success
- We arrange the table ordered by date in descending order

Section 4

Launch Sites Proximities Analysis

All launch sites global map markers

We can see that the SpaceX launch sites are in the United States of America coasts. Florida and California

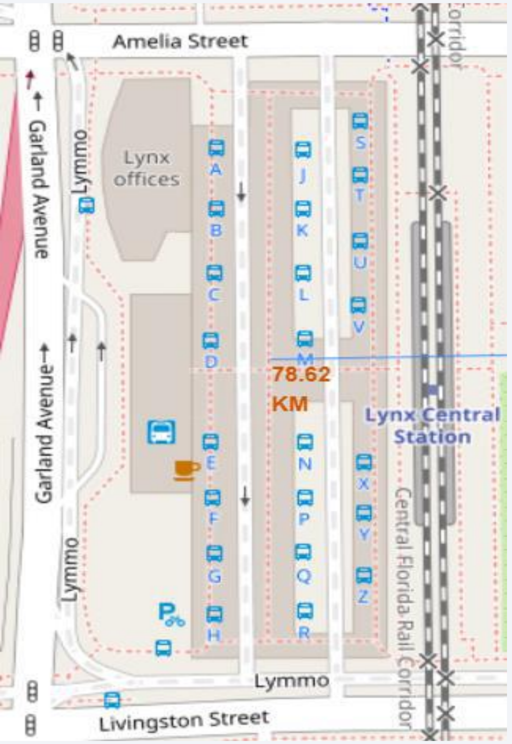
COLORED MARKERS

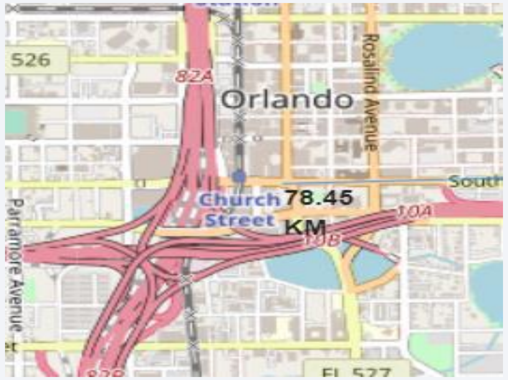
Florida Launch Sites

California Launch Launch Site

Green Marker shows successful Launches and Red marker shows Failures

Launch Sites distance to different landmarks

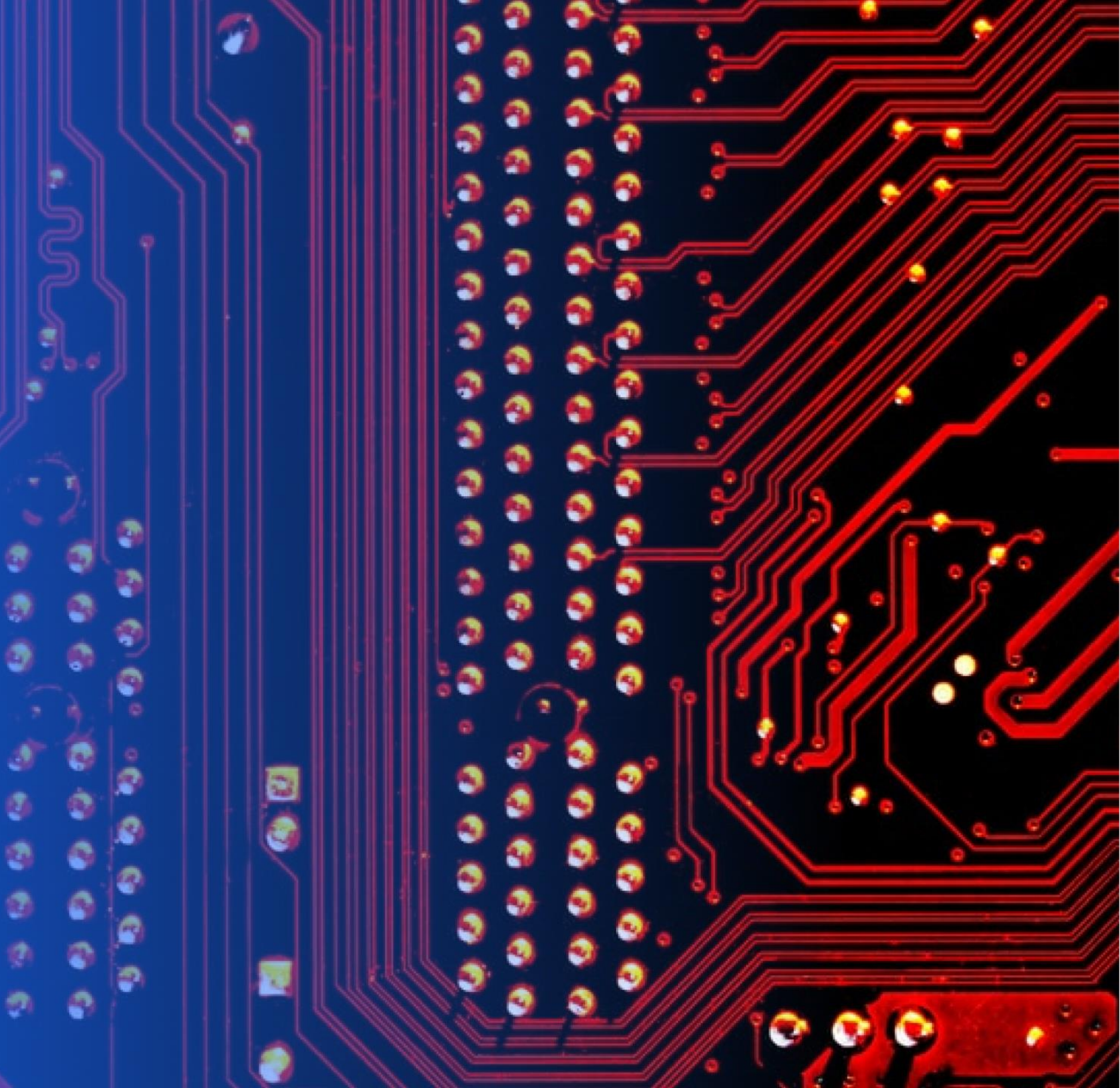




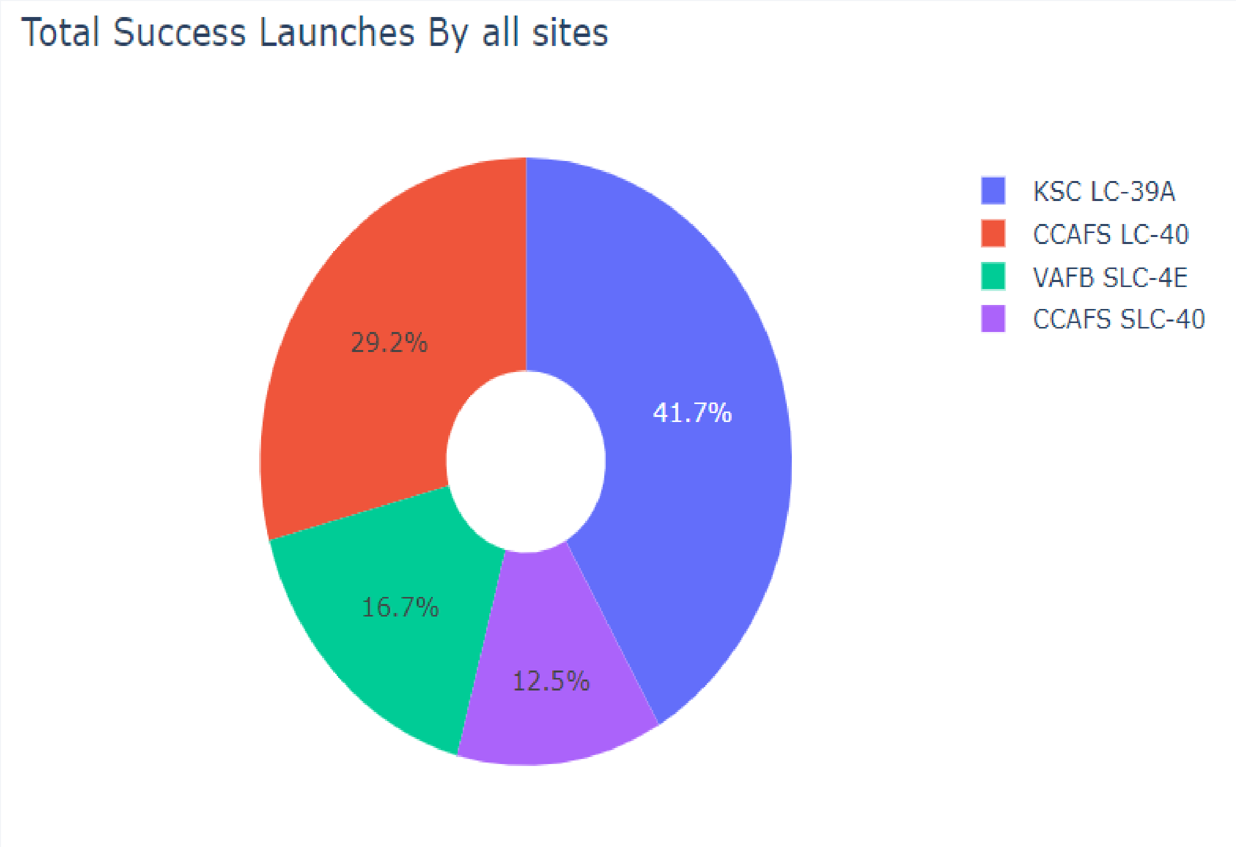
- ❑ Are launch sites in close proximity to railways? No
- ❑ Are launch sites in close proximity to highways? No
- ❑ Are launch sites in close proximity to coastline? Yes
- ❑ Do launch sites keep certain distance away from cities? Yes

Section 5

Build a Dashboard with Plotly Dash



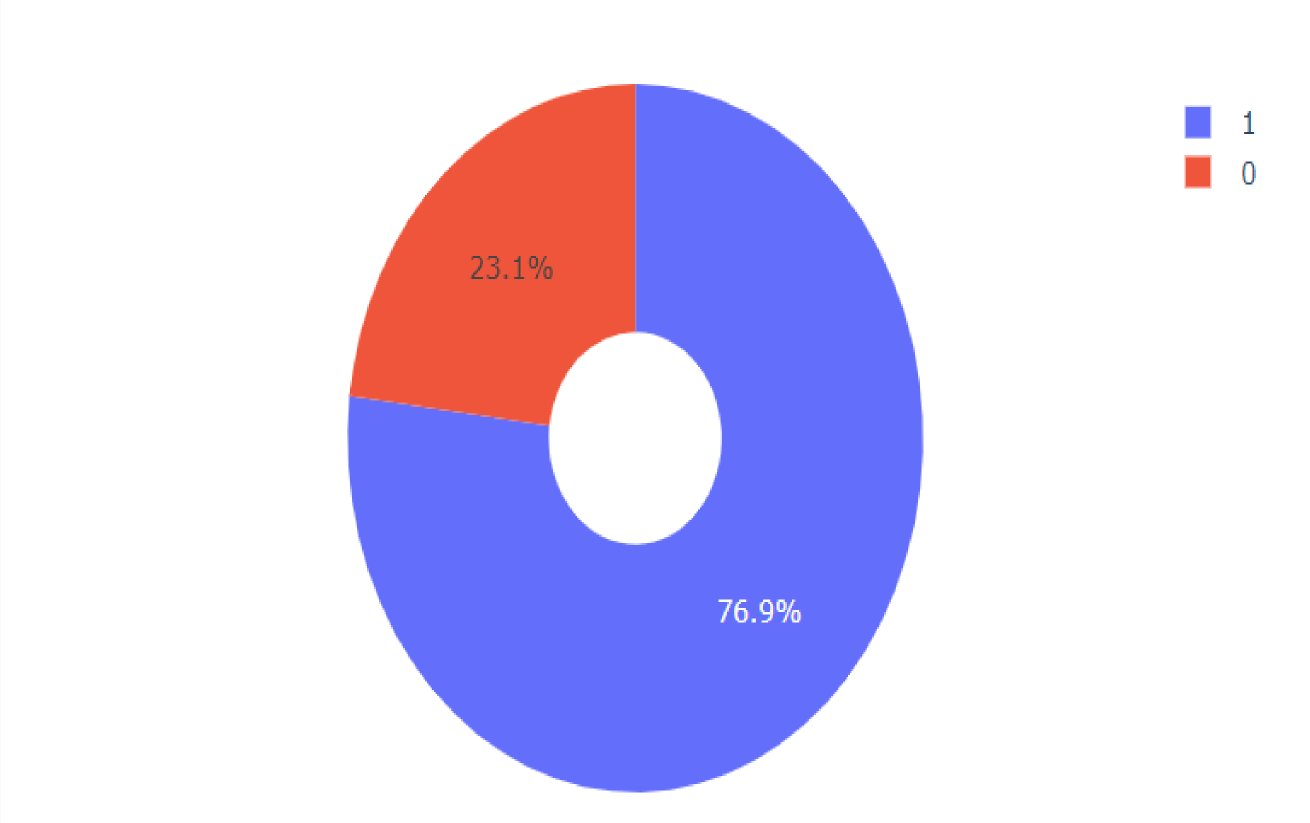
DASHBOARD - Pie chart showing the success percentage achieved by each launch site



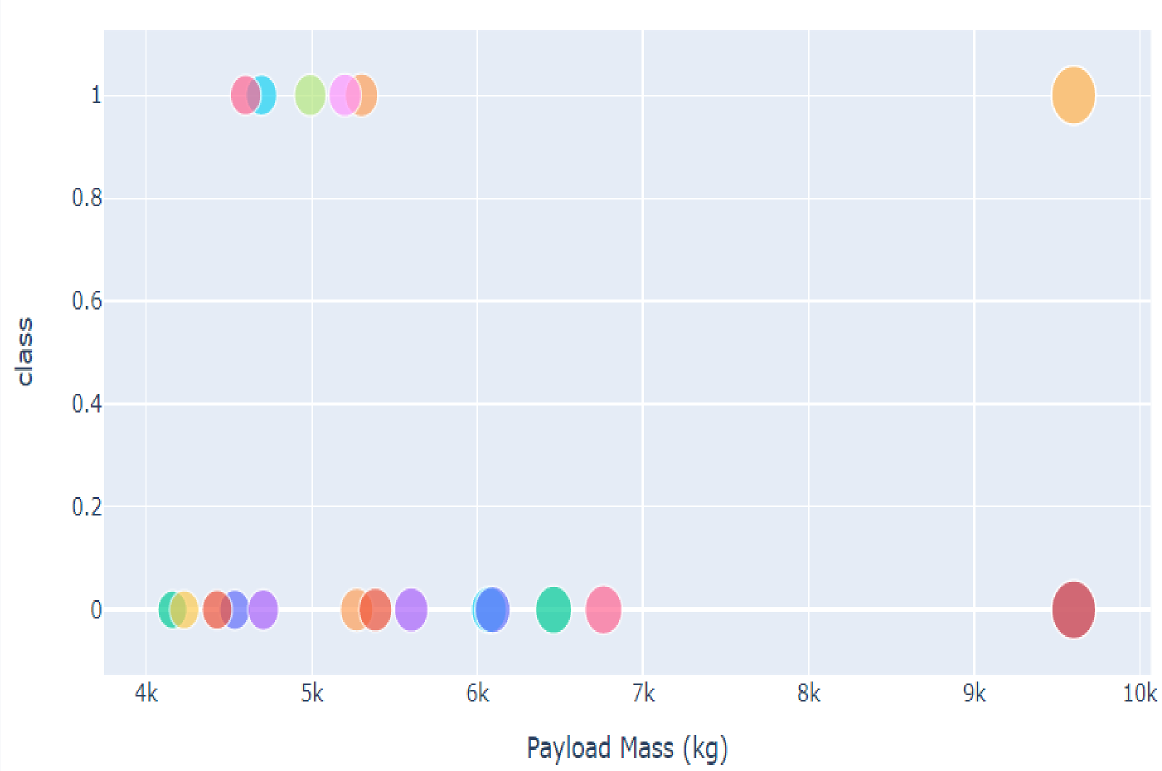
KSC LC-39A had the most successful launches from all the sites followed by CCAFS LC-40

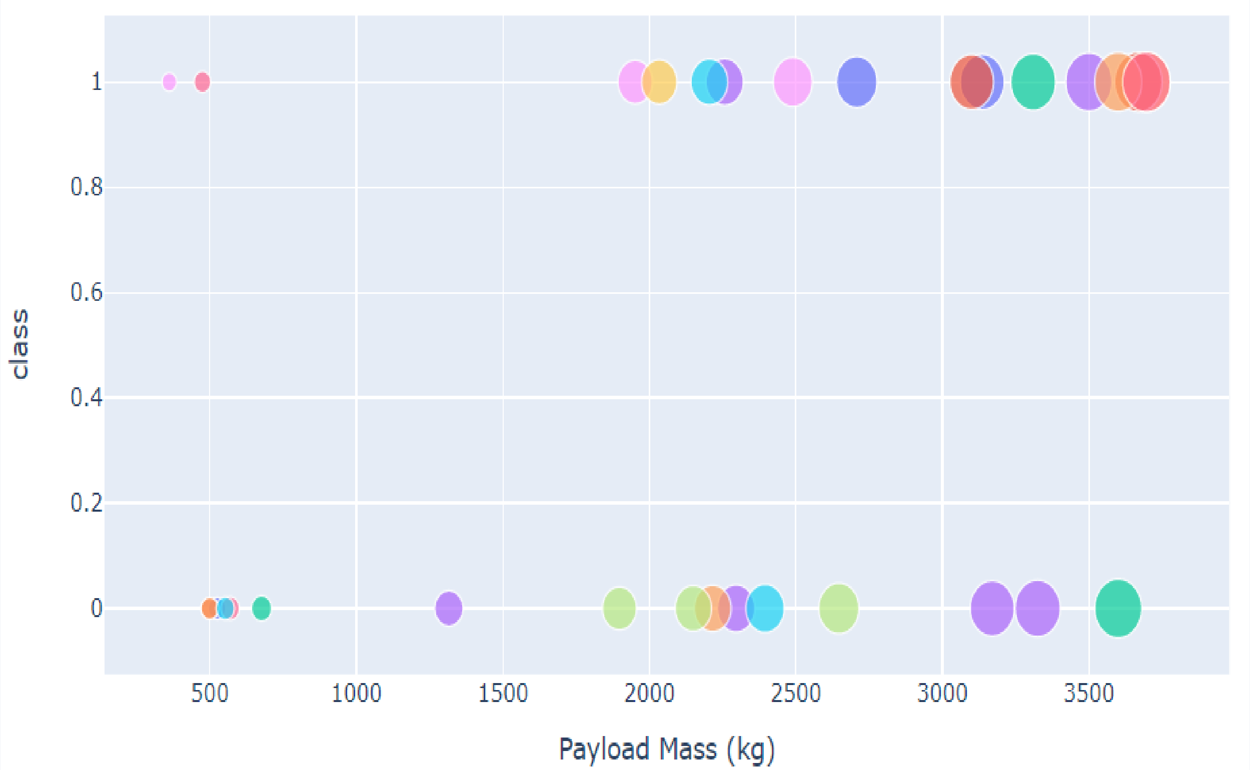
DASHBOARD – Pie chart for the launch site with highest launch success ratio

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



DASHBOARD–Payload vs. Launch Outcome scatter plot for all sites, with different payload selected in the range slider



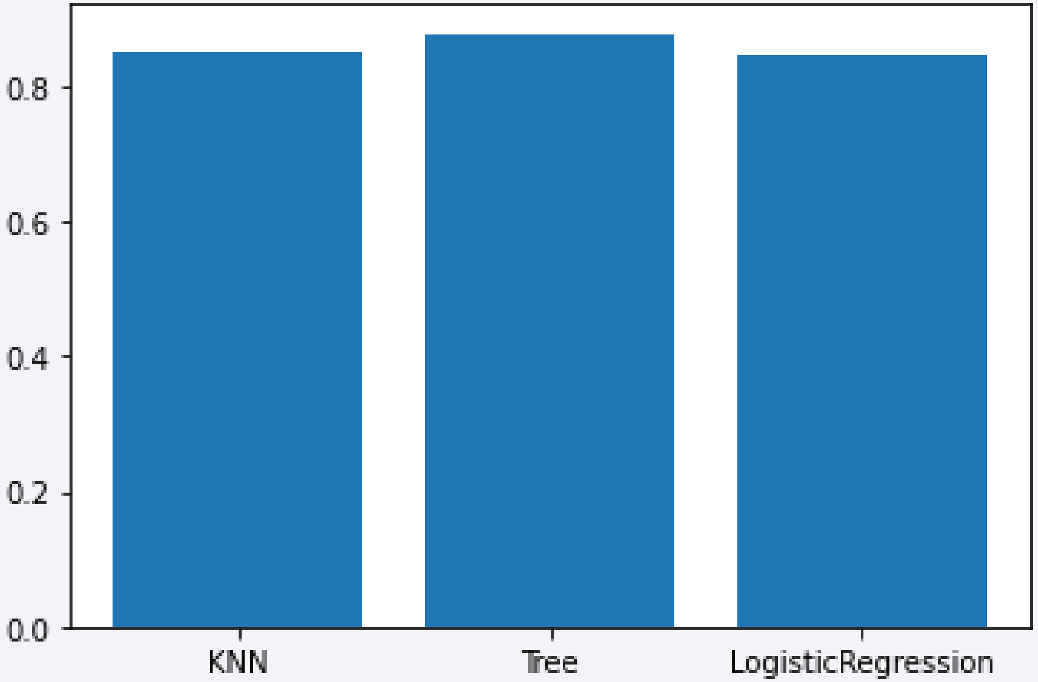


We can see the success rates for low weighted payloads is higher than the heavy weighted payloads

Section 6

Predictive Analysis (Classification)

Classification Accuracy



Best Algorithm is Tree with a score of 0.8767857142857143

Best Params is : {'criterion': 'gini', 'max_depth': 6, 'max_features': 'auto', 'min_samples_leaf': 4, 'min_samples_split': 2, 'splitter': 'best'}

Confusion Matrix

Examining the confusion matrix, we see that Tree can distinguish between the different classes.

Conclusions

- The Tree Classifier Algorithm is the best for Machine Learning for this dataset
- 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

Appendix

- Haversine formula
- PythonAnywhere 24/7 dashboard

Thank you!