IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

MUHAMMAD DANIAL BIN ZULKIFLI
7 October 2022

Outline

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

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
  - Machine Learning Prediction
- Summary of all results
  - Exploratory Data Analysis result
  - Interactive analytics in screenshots
  - Predictive Analytics result from Machine Learning Lab

Introduction

- SpaceX is an American spacecraft manufacturer, space launch provider, and a satellite communications corporation headquartered in Hawthorne, California. It was founded in 2002 by Elon Musk, with the goal of reducing space transportation costs to enable the colonization of Mars. It manufactures the Falcon 9 and Falcon Heavy launch vehicles, several rocket engines, Cargo Dragon, crew spacecraft, and Starlink communications satellites.
- In this project, We did the prediction of SpaceX Falcon 9 First Stage Landing

Section 1

Methodology

to collect Falcon 9 historical launch records from a Wikipedia page titled `List of Falcon 9 and Falcon Heavy launches`

Methodology

Executive Summary

- Data collection methodology:
    - Collected by Web scrapping from Wikipedia and SpaceX API
- Perform data wrangling
    - Processed by using one-hot encoding
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
    - How to build, tune, evaluate classification models

Data Collection

1. Performed web scraping to collect Falcon 9 historical launch records from a Wikipedia page titled List of Falcon 9 and Falcon Heavy launches from https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches. Used BeautifulSoup function to extract HTML to Jupyter Notebook.

2. Collected data by requesting from SpaceX API. Decoded the response content as a Json using .json() and turn it into a Pandas dataframe using .json_normalize().

Data Collection – SpaceX API

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

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

```
static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.ap
```

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

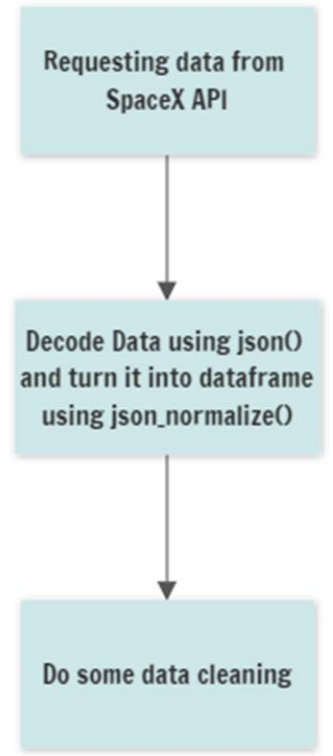
```
# Lets take a subset of our dataframe keeping only the features we want and the flight nu
data = data[['rocket', 'payloads', 'launchpad', 'cores', 'flight_number', 'date_utc']]

# We will remove rows with multiple cores because those are falcon rockets with 2 extra r
data = data[data['cores'].map(len)==1]
data = data[data['payloads'].map(len)==1]

# Since payloads and cores are lists of size 1 we will also extract the single value in t
data['cores'] = data['cores'].map(lambda x : x[0])
data['payloads'] = data['payloads'].map(lambda x : x[0])

# We also want to convert the date_utc to a datetime datatype and then extracting the dat
data['date'] = pd.to_datetime(data['date_utc']).dt.date

# Using the date we will restrict the dates of the launches
data = data[data['date'] <= datetime.date(2020, 11, 13)]
```



[Data Collection using API GITHUB]

Data Collection - Scraping

```
launchdata = requests.get(static_url).text
soup = BeautifulSoup(launchdata,'html5lib')
```

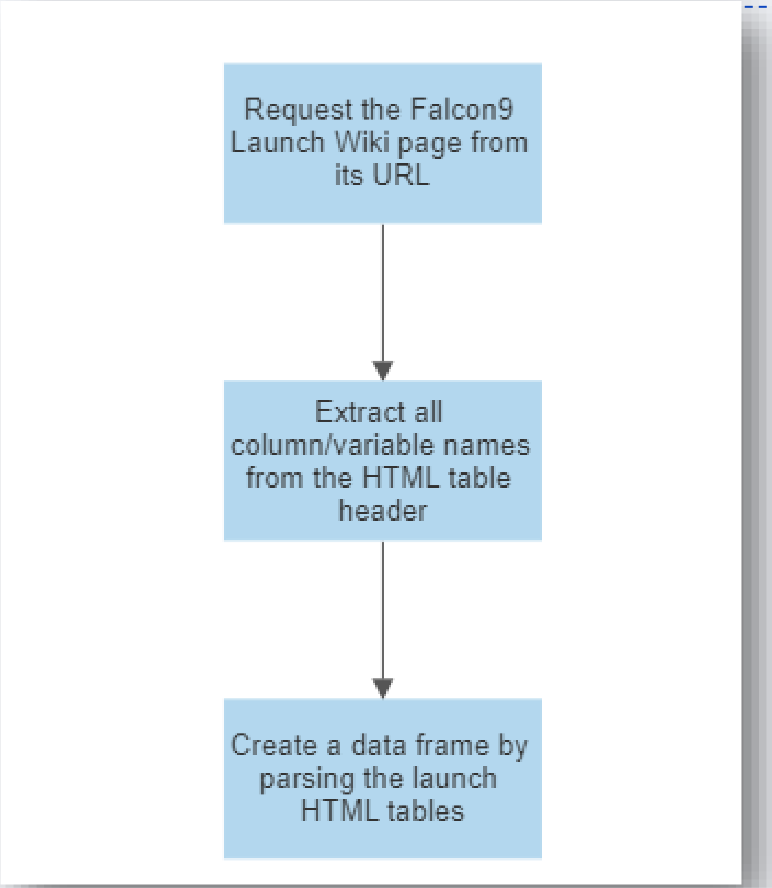
```
html_tables = soup.find_all('table')
```

```
# Let's print the third table and check its content
first_launch_table = html_tables[2]
print(first_launch_table)
```

```
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']=[]
```



Data Collection Web Scraping Github

Data Wrangling

- Data wrangling is the process of removing errors and combining complex data sets to make them more accessible and easier to analyze by using Exploratory Data Analysis (EDA)
- In this project, we calculated number of launches on each site, number and occurrence of each orbit, mission outcome per orbit type, landing outcome and its mean
- GitHub URL - [Data Wrangling]()

```
# Apply value_counts() on column LaunchSite
df['LaunchSite'].value_counts()
```

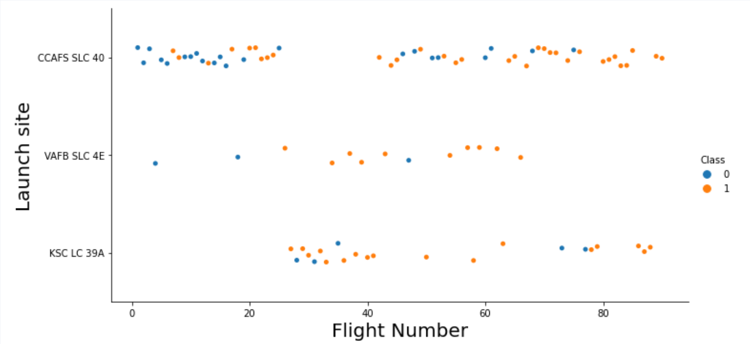
```
# landing_outcomes = values on Outcome column
landing_outcomes = df['Outcome'].value_counts()
landing_outcomes
```

```
# Apply value_counts on Orbit column
df['Orbit'].value_counts()
```

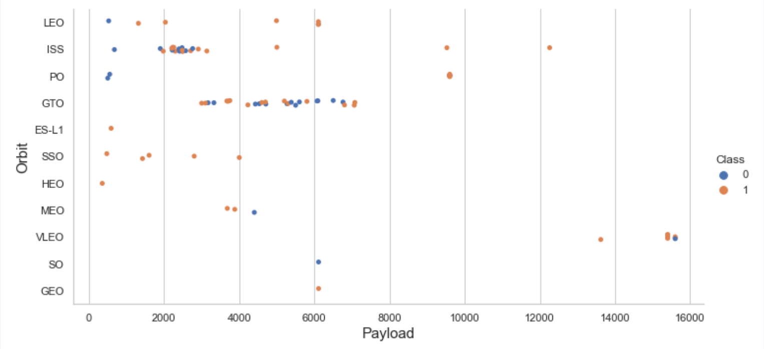
```
df["Class"].mean()
```

EDA with Data Visualization

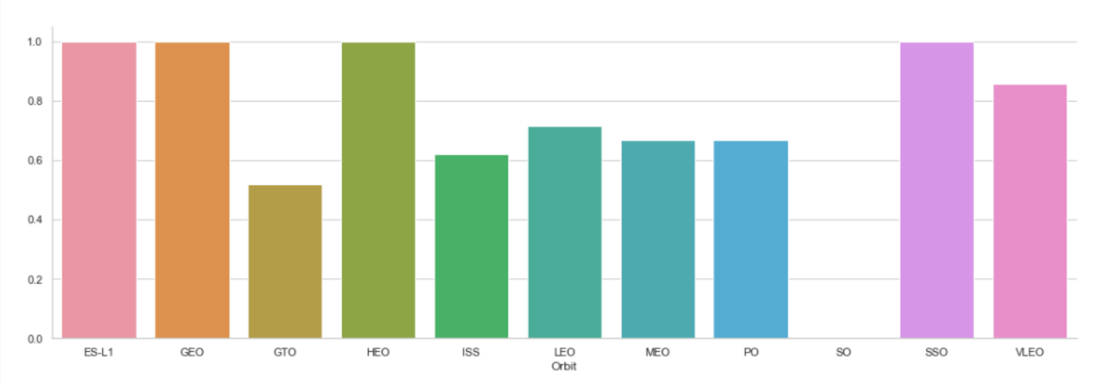
- Scatterplot is used to Visualize the relationship between Flight Number and Launch Site



- Scatterplot also is used to Visualize the relationship between Payload and Orbit type



- Bar chart is used to Visualize the relationship between success rate of each orbit type



- GitHub URL - EDA with Data Visualization

EDA with SQL

- GitHub URL - [EDA with SQL]

- EDA with SQL in this project are:
  - 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 acheived.
  - 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

Predictive Analysis (Classification)

Building the Model

- Load dataset into Numpy and Pandas
- Transform data and split into train and test datasets
- Decide which Machine learning method to use

Evaluating Model

- Check the accuracy for each mode
- plot the confusion matrix

Improving Model

- Feature Engineering and Algorithm Tuning are used

Best Model

- Use model with best accuracy score as best model

- GitHub URL - [Predictive Analysis]

Results

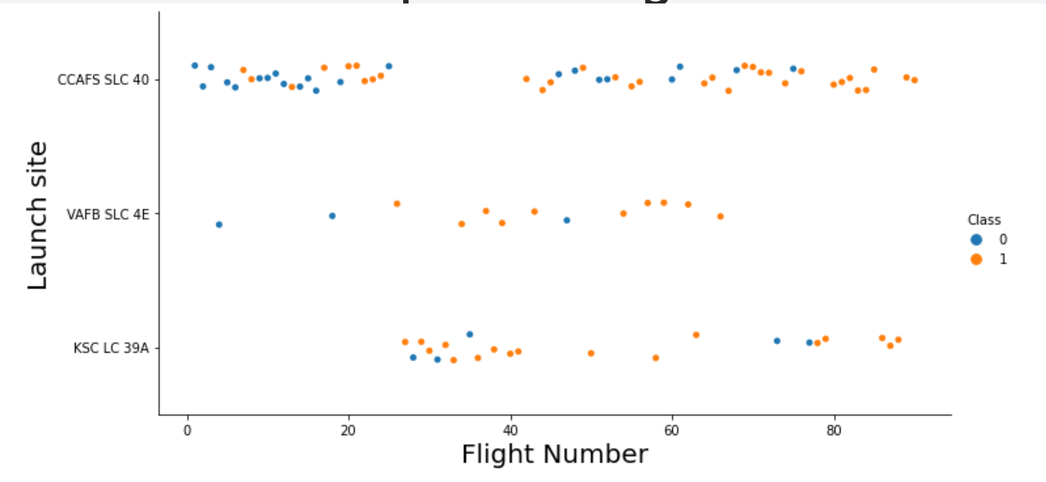
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

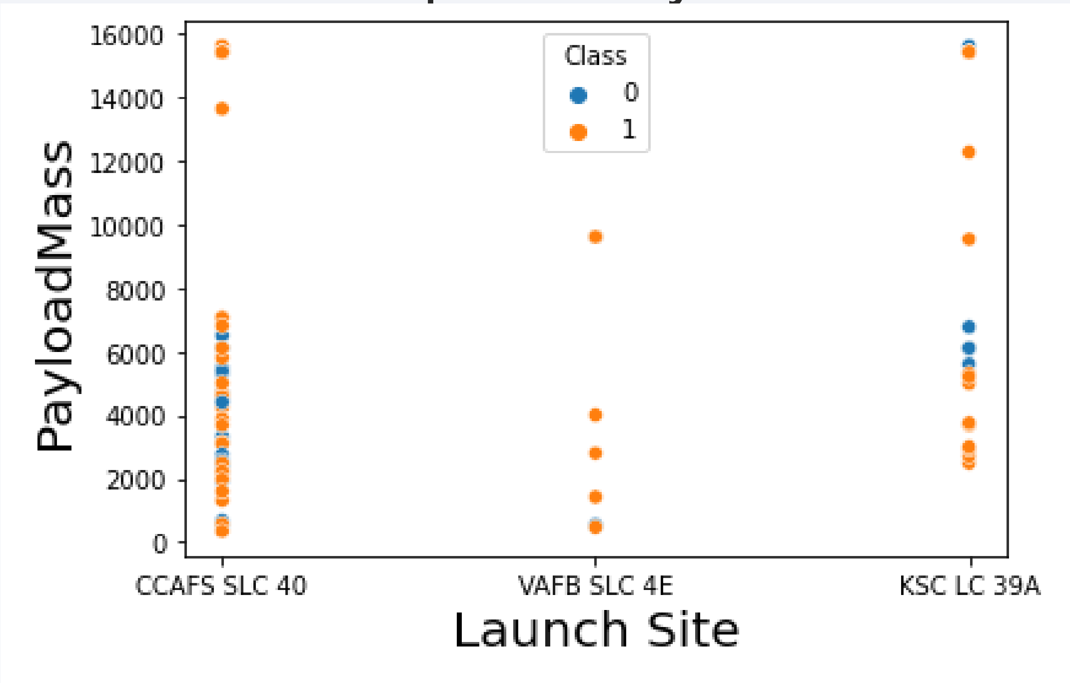
- Show a scatter plot of Flight Number vs. Launch Site



- This scatter plot shows that the larger the flights amount of the launch site, the greater the the success rate will be

Payload vs. Launch Site

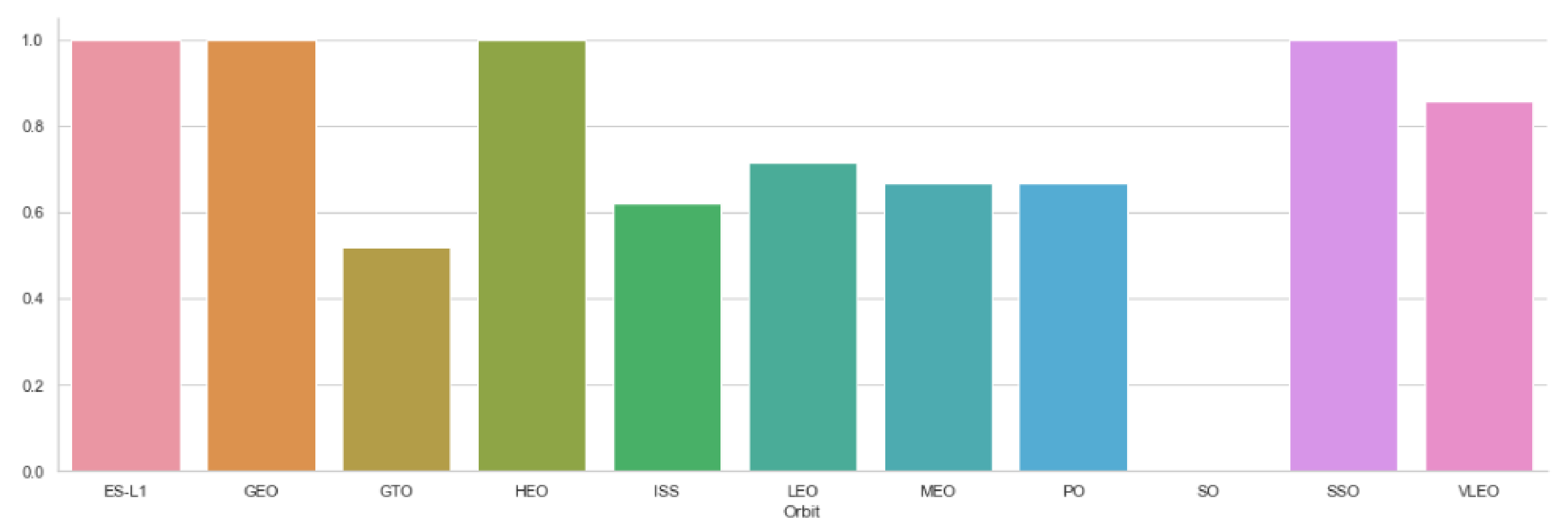
- Show a scatter plot of Payload vs. Launch Site



- This graph shows that after 7000kg of payload mass, the success rate would be higher

Success Rate vs. Orbit Type

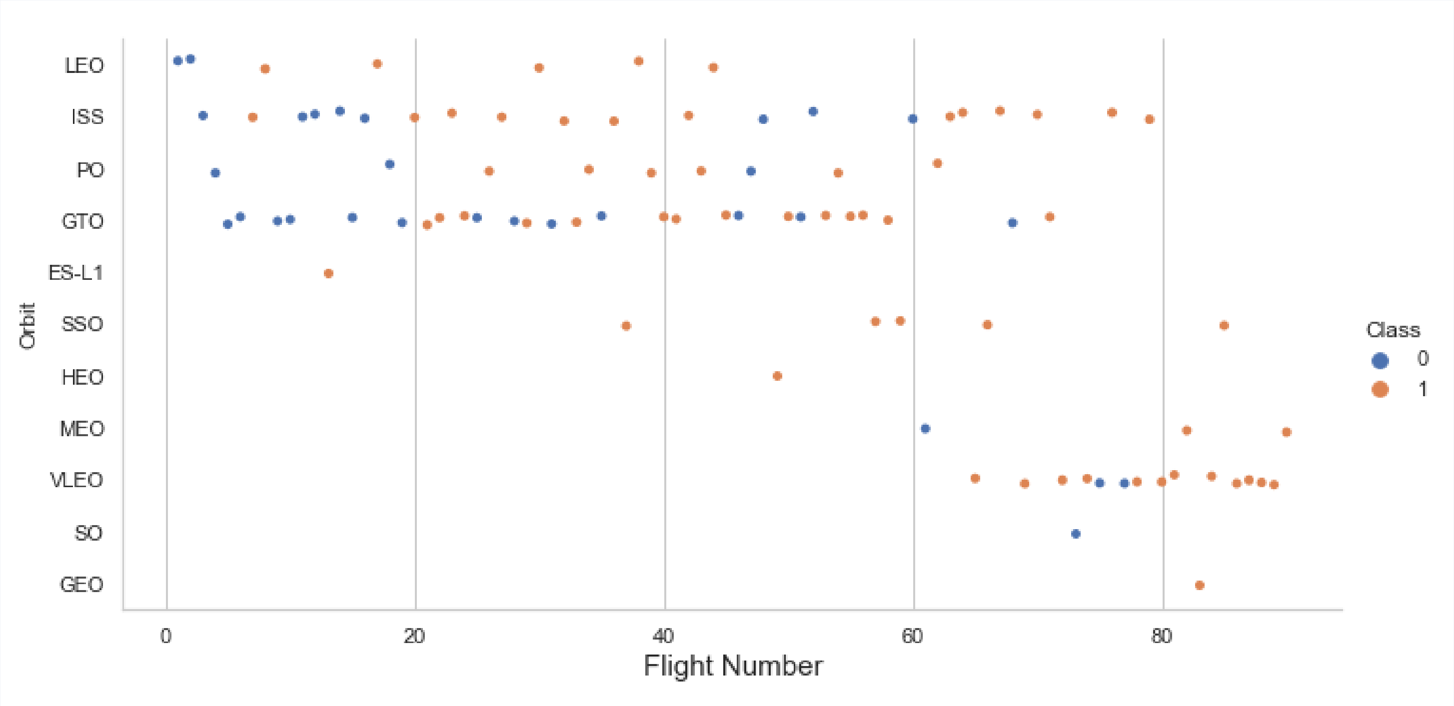
- Show a bar chart for the success rate of each orbit type



- This Bar chart shows that ES-L1, GEO, HEO and SSO have the highest success rate

Flight Number vs. Orbit Type

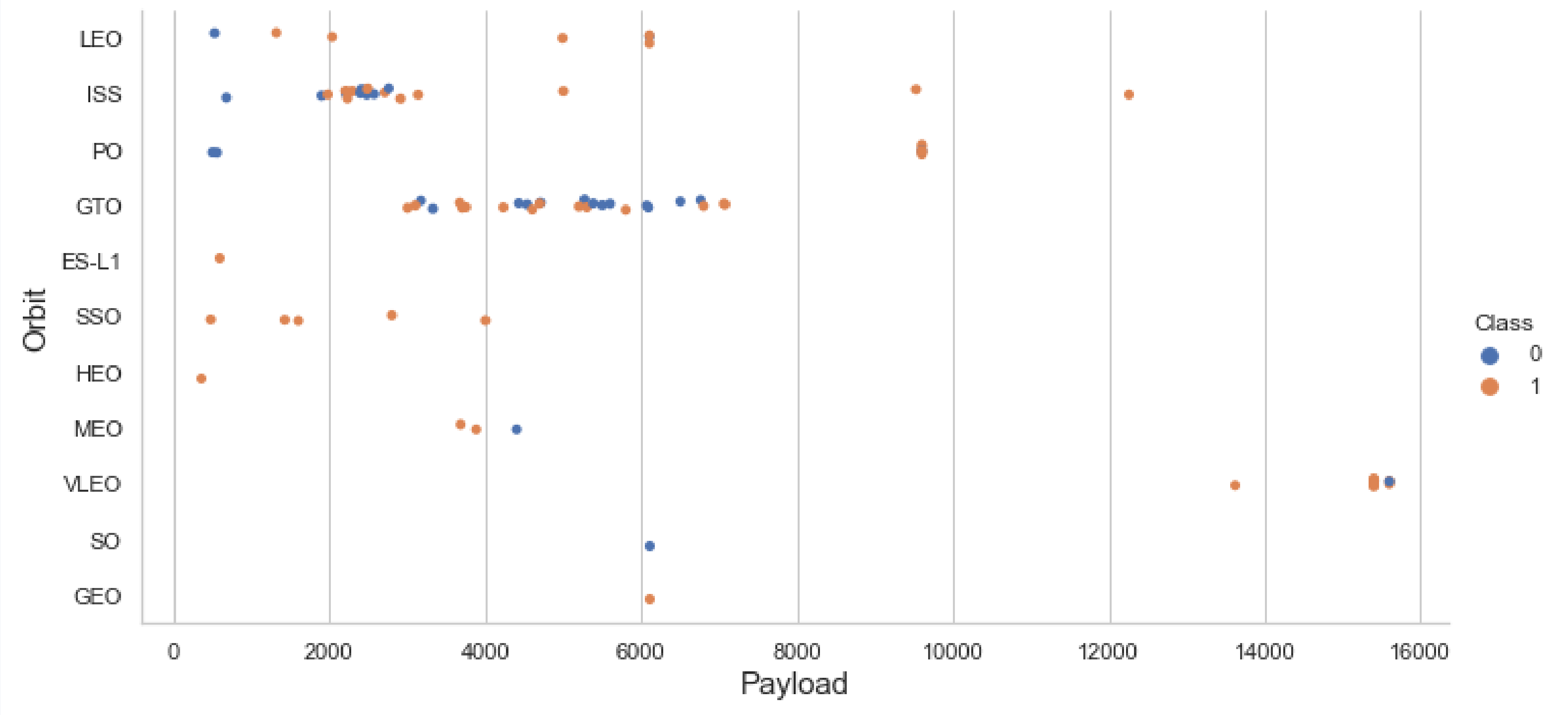
- Show a scatter point of Flight number vs. Orbit type



- this scatter plot demonstrates that the bigger the flight number on each orbit, thehigher the rate of success(particularly LEO orbit), although GTO orbit, which shows nothing connection between the two attributes

Payload vs. Orbit Type

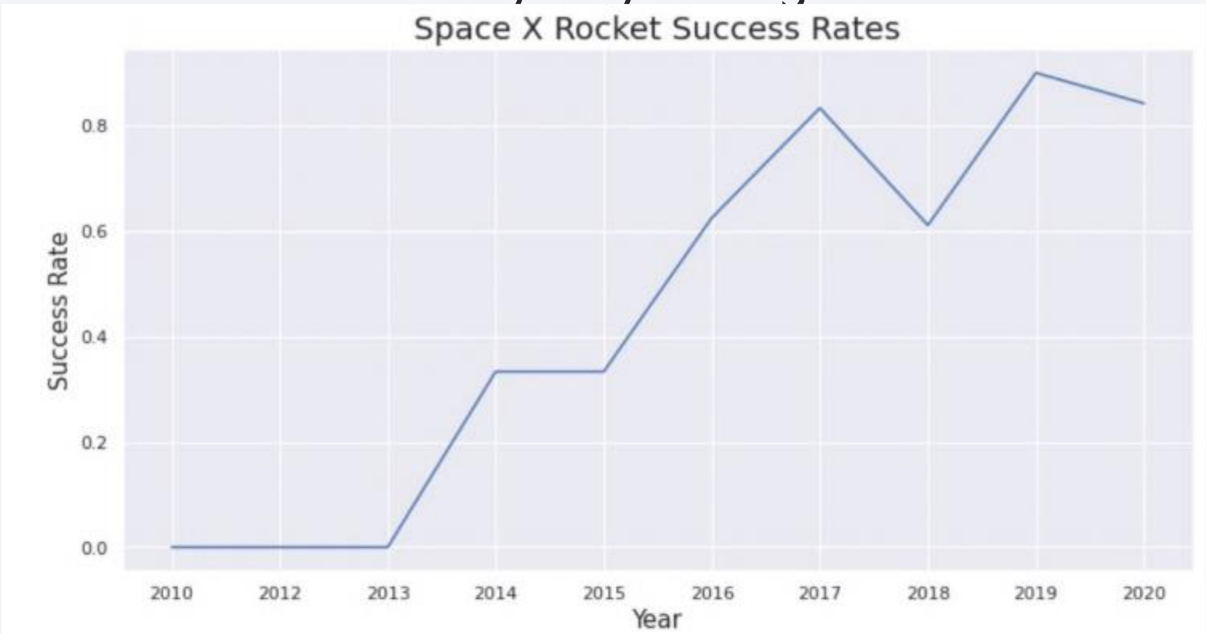
- Show a scatter point of payload vs. orbit type



- There are positive effects of heavier payload on LEO, ISS, and PO orbits. Meanwhile, MEO and VLEO orbits are negatively impacted. There's not so much data for ESL-L1, HEO SO and GEO

Launch Success Yearly Trend

- Show a line chart of yearly average success rate



- This Line chart showed a definite upward pattern of success rate from 2013 to 2020.

All Launch Site Names

- Find the names of the unique launch sites

In [8]:
```
%sql SELECT distinct(launch_site) FROM SPACEX
```

```
 * ibm_db_sa://brf42866:***@125f9f61-9715-46f9-9399-
Done.
```

Out[8]:

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

- We used distinct(launch_site) to get unique name of launch sites

Launch Site Names Begin with 'CCA'

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

```
%sql SELECT * FROM SPACEX WHERE launch_site LIKE 'CCA%' LIMIT 5
```

```
 * ibm_db_sa://brf42866:***@125f9f61-9715-46f9-9399-c8177b21803b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30426/BLUDB
Done.
```

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

- We used command **LIKE ‘CCA%’** to get launch sites begin with ‘CCA’ only

Total Payload Mass

- Calculate the total payload carried by boosters from NASA

```
%%sql
SELECT SUM(payload_mass__kg_) as total_payload_mass FROM SPACEX
    WHERE customer = 'NASA (CRS)'
```

```
 * ibm_db_sa://brf42866:***@125f9f61-9715-46f9-9399-c8177b21803b.c
Done.
```

total_payload_mass
22007

- We used command SUM(payload_mass__kg_) to get total sum of payload mass. Command WHERE is used to get data of customer with id NASA (CRS) only.

- The total payload carried by boosters from NASA is 22007kg

Average Payload Mass by F9 v1.1

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

```
%%sql
SELECT AVG(payload_mass__kg_) as AVG_payload_mass FROM SPACEX
    WHERE booster_version = 'F9 v1.1'
```

```
 * ibm_db_sa://brf42866:***@125f9f61-9715-46f9-9399-c8177b21803b
Done.
```

avg_payload_mass
3676

- We used AVG(payload_mass__kg_) to get average of payload mass and WHERE command to get data of booster version F9 v1.1 only.
- The average payload mass carried by booster version F9 v1.1 is 3676kg

First Successful Ground Landing Date

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

```
%%sql

SELECT min(Date) FROM SPACEX
    WHERE landing__outcome LIKE 'Success%'
```

```
 * ibm_db_sa://brf42866:***@125f9f61-9715-4
Done.
```

1
2016-06-05

- min(Date) command is used to get minimum date. WHERE and LIKE command are used to get date of landing outcome start with keyword ‘Success’
- Date of the first successful landing outcome on ground pad is 5 Jun 2016

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

```
%%sql

SELECT distinct(booster_version) FROM SPACEX
    WHERE landing__outcome = 'Success (drone ship)' AND (payload_mass__kg_ > 4000 AND payload_mass__kg_ < 6000)
```

```
 * ibm_db_sa://brf42866:***@125f9f61-9715-46f9-9399-c8177b21803b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30
Done.
```

booster_version
F9 FT B1031.2
F9 FT B1022

- distinct(booster_version) command is used to get unique name of boost. AND (payload_mass__kg_ > 4000 AND payload_mass__kg_ < 6000) is used to get payload mass greater than 4000 but less than 6000
- List the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000 is F9 FT B1031.2 and F9 FT B1022

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes

```
%%sql

select count(MISSION_OUTCOME) as missionoutcomes from SPACEX
    GROUP BY MISSION_OUTCOME;
```

```
 * ibm_db_sa://brf42866:***@125f9f61-9715-46f9-9399-c8177b21803
Done.
```

missionoutcomes
44
1

- count(MISSION_OUTCOME) command is used to calculate total mission_outcome. GROUP BY function is to separate mission outcome as successful and failure
- The total number of successful mission outcomes is 44 while failure is 1

Boosters Carried Maximum Payload

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

```
%%sql

SELECT booster_version FROM SPACEX
    WHERE payload_mass__kg_ = (SELECT max(payload_mass__kg_) FROM SPACEX)
```

```
 * ibm_db_sa://brf42866:***@125f9f61-9715-46f9-9399-c8177b21803b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30426/BLUDB
Done.
```

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

- WHERE payload_mass__kg_ = (SELECT max(payload_mass__kg_) FROM SPACEX) is a command to set payload__mass__kg_ to maximum value of payload_mass__kg_

2015 Launch Records

- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

```
%%sql

SELECT landing__outcome,booster_version, launch_site FROM SPACEX
    WHERE landing__outcome = 'Failure (drone ship)' and EXTRACT(YEAR FROM DATE)= '2015'
```

```
 * ibm_db_sa://brf42866:***@125f9f61-9715-46f9-9399-c8177b21803b.c1ogj3sd0tgtu0lqde00.dat
Done.
```

landing__outcome	booster_version	launch_site
Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40

- WHERE landing__outcome = 'Failure (drone ship)' and EXTRACT(YEAR FROM DATE)= '2015' to find failure drone ship in year 2015

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

```
SELECT * FROM SPACEXTBL where DAY(DATE)='Friday' LIMIT 5
```

```
 * ibm_db_sa://brf42866:***@125f9f61-9715-46f9-9399-c8177b21803b.
Done.
```

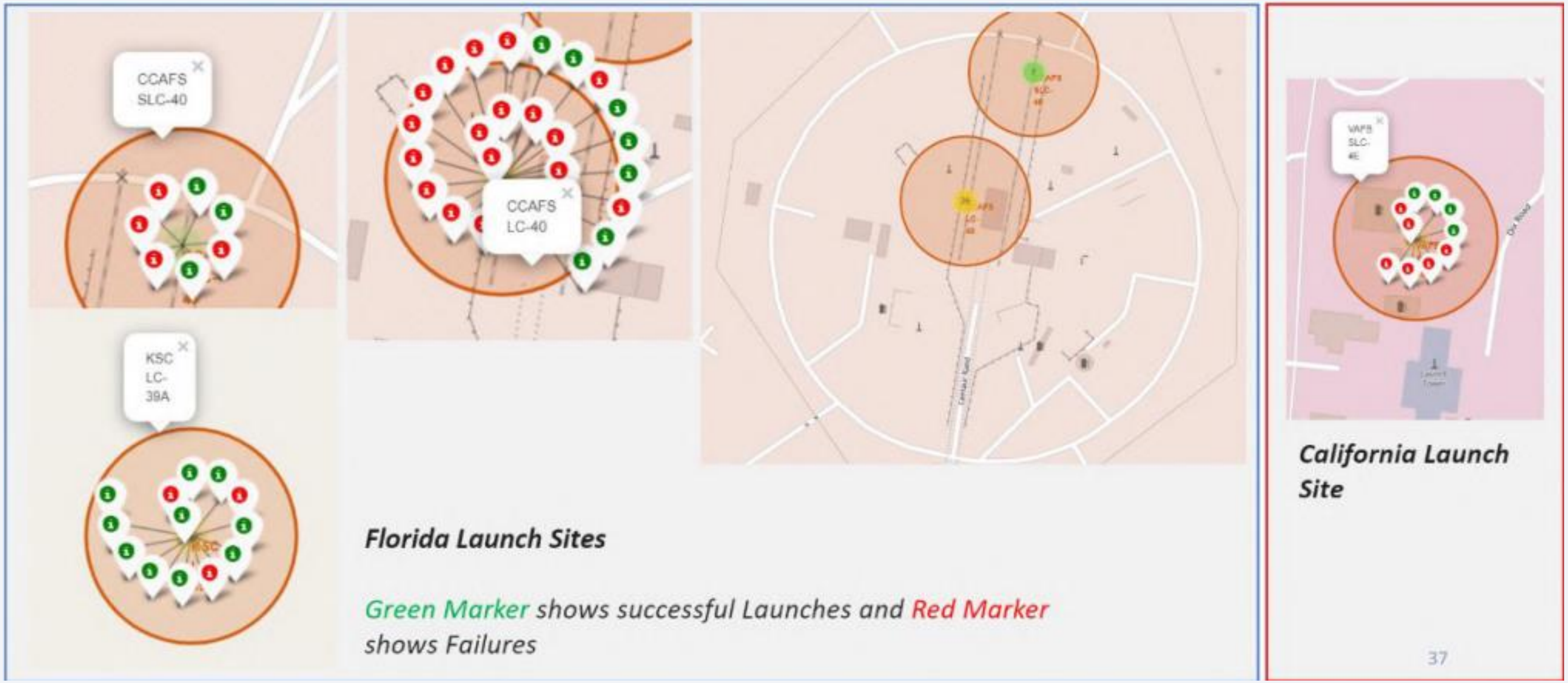
landing__outcome
Success (ground pad)
Success (ground pad)
Success (drone ship)
Success (drone ship)
Failure (drone ship)
Controlled (ocean)
Failure (drone ship)
No attempt
No attempt
No attempt
No attempt
No attempt
No attempt
No attempt
Failure (parachute)

- DAY(DATE) command to find which date is Friday for landing outcomes

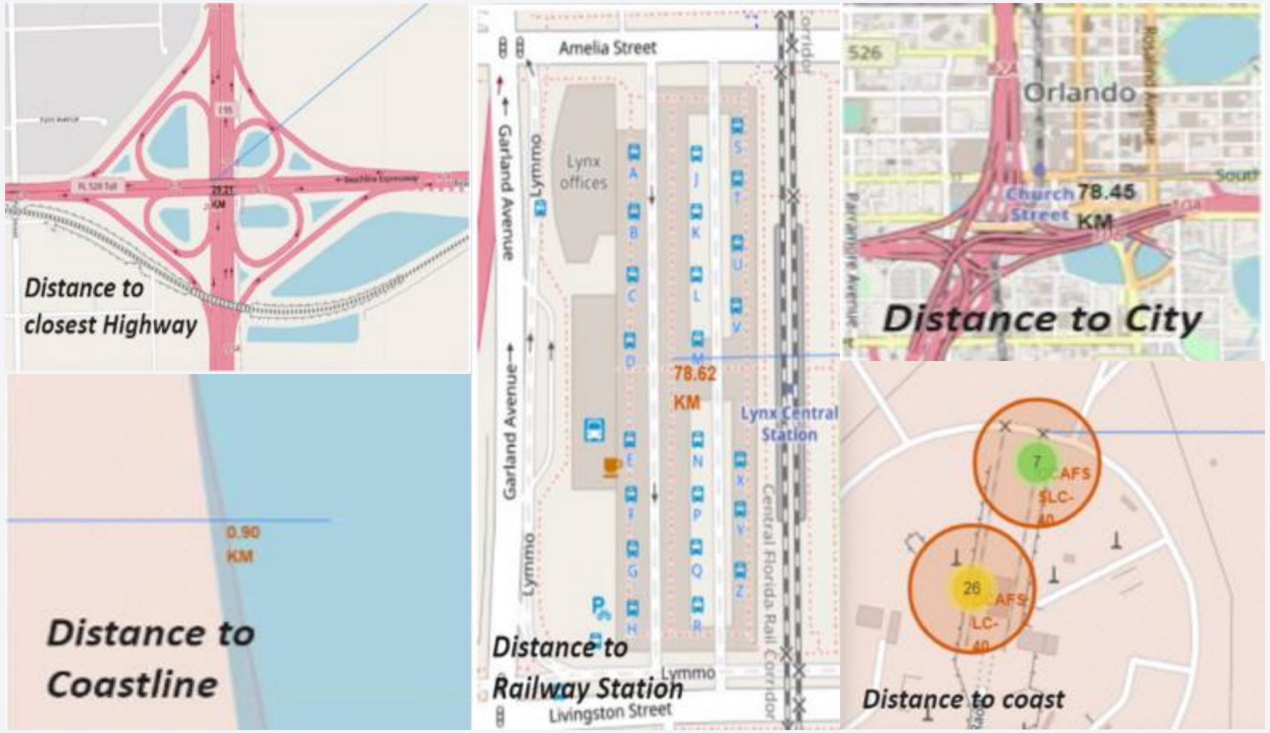
Section 3

Launch Sites Proximities Analysis

Launch sites markers



Launch Sites Distance to Landmarks



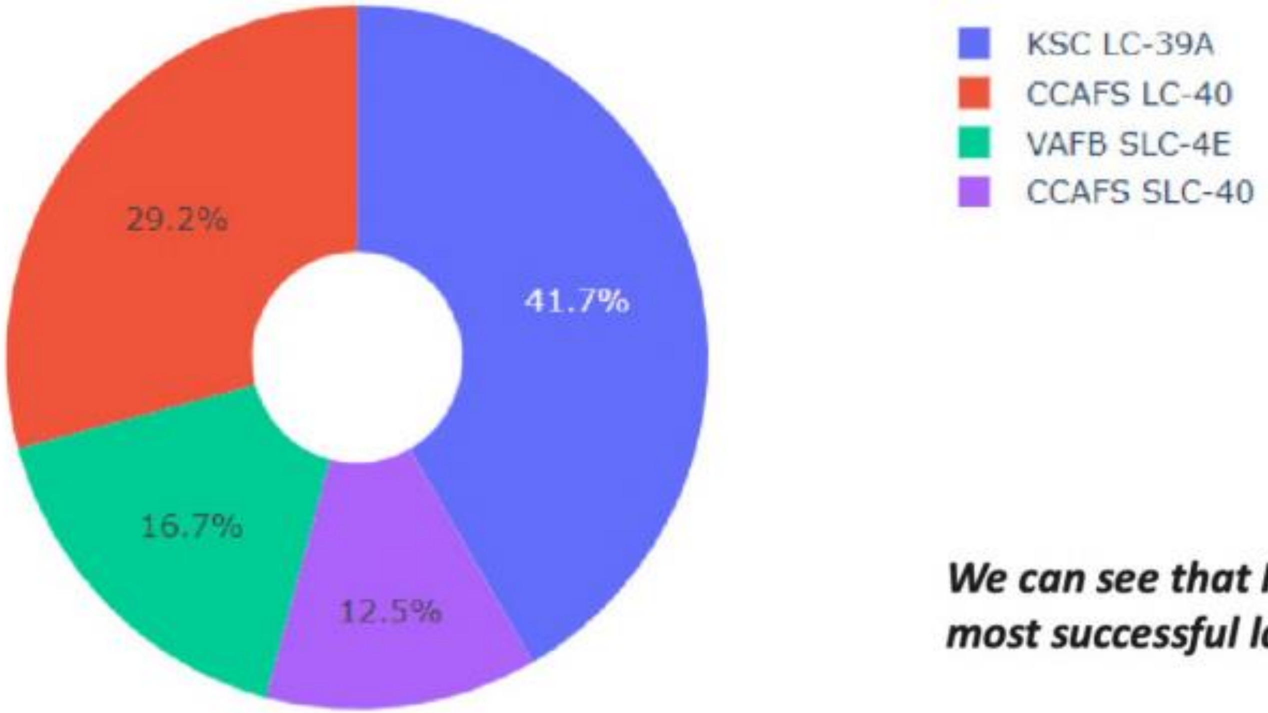
Launch Sites Location

- Launch sites of SpaceX

Section 4

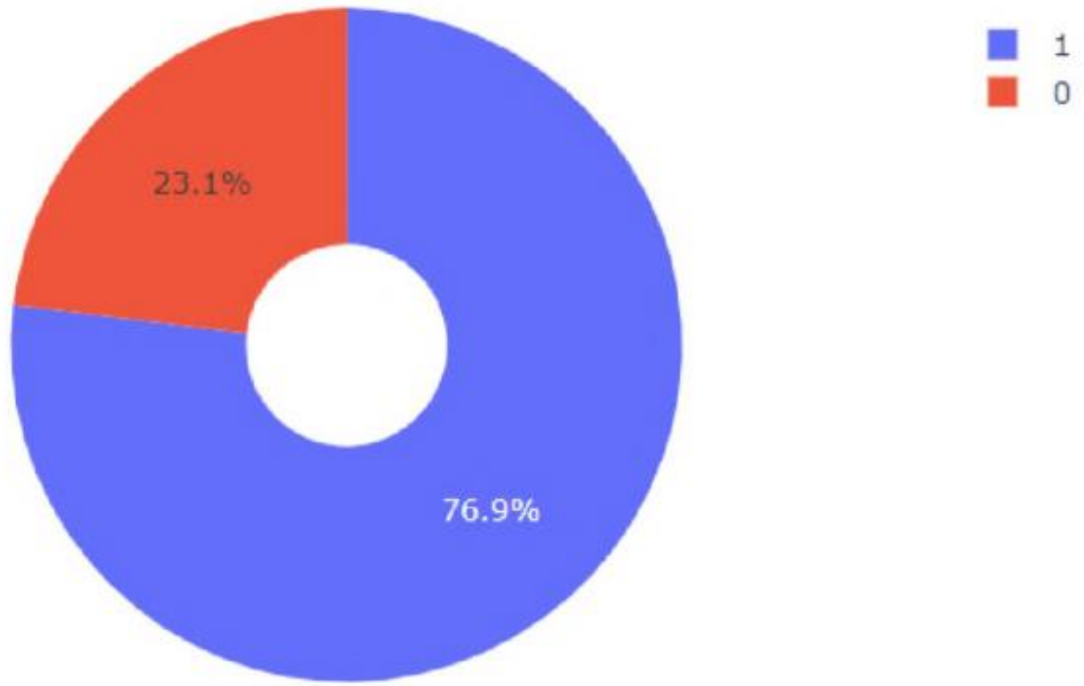
Build a Dashboard with Plotly Dash

Pie chart of percentage of success launch sites



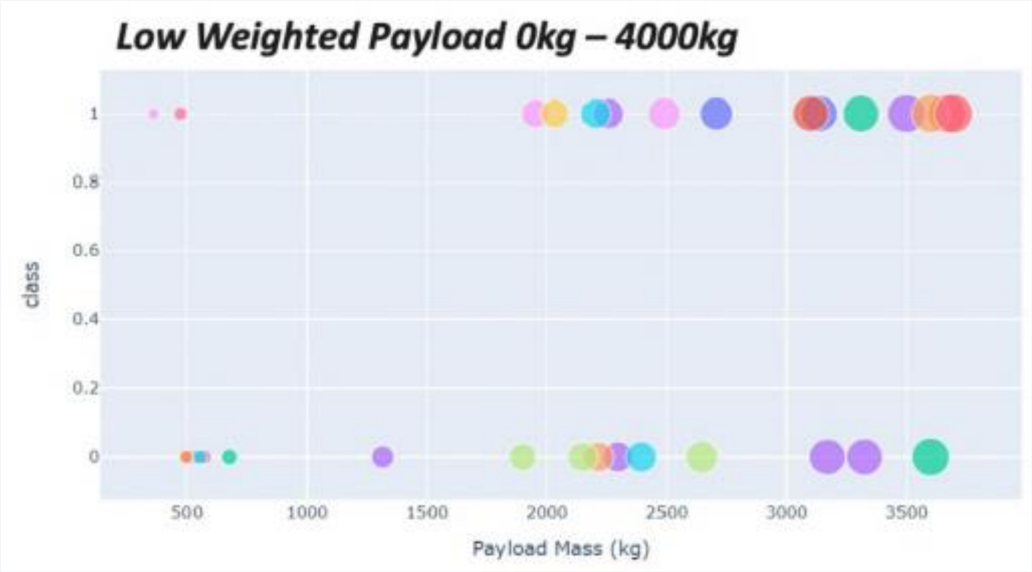
We can see that KSC LC-39A had the most successful launches from all the sites

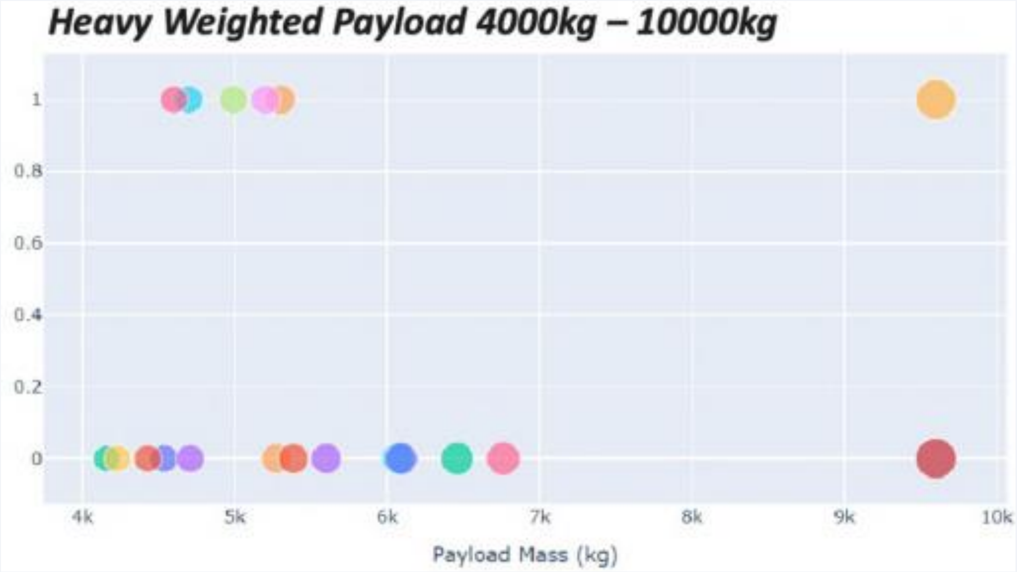
The highest launch-success ratio: KSC LC-39A



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

Payload vs Launch Outocome Scatter Plot





Section 5

Predictive Analysis (Classification)

Classification Accuracy

- By using these codings, we can see that all method has the same accuracy. Hence, I randomly chose Decision tree method for best method.

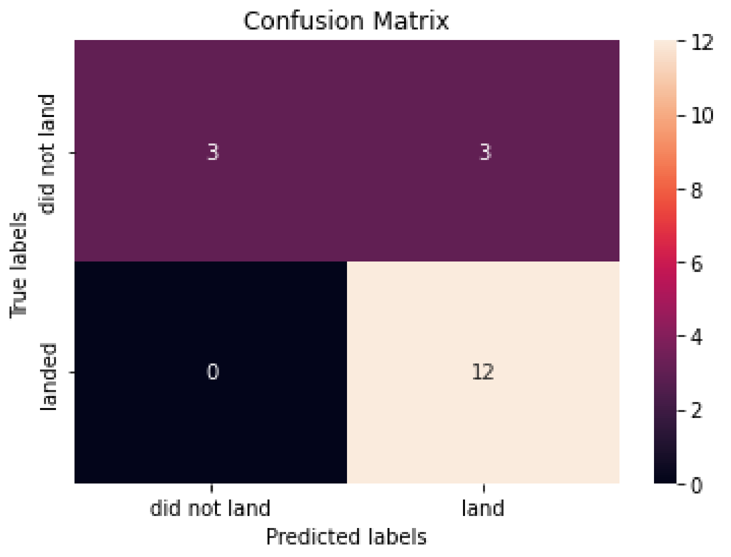
```
print('Accuracy for Logistics Regression method:', logreg_cv.score(X_test, Y_test))
print( 'Accuracy for Support Vector Machine method:', svm_cv.score(X_test, Y_test))
print('Accuracy for Decision tree method:', tree_cv.score(X_test, Y_test))
print('Accuracy for K nearsdt neighbors method:', knn_cv.score(X_test, Y_test))
```

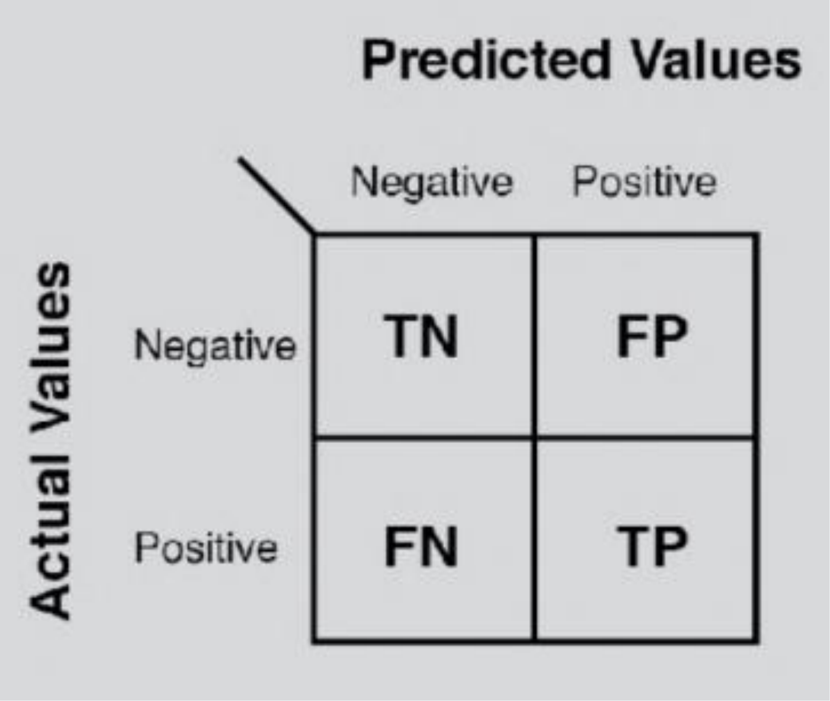
```
Accuracy for Logistics Regression method: 0.8333333333333334
Accuracy for Support Vector Machine method: 0.8333333333333334
Accuracy for Decision tree method: 0.8333333333333334
Accuracy for K nearsdt neighbors method: 0.8333333333333334
```

Confusion Matrix

- Confusion Matrix using Decision Tree Method. The decision tree classifier's confusion matrix demonstrates that it is capable of differentiating between the various classes. False positives are the main issue. i.e., the classifier classifying an unsuccessful landing as a successful landing.

```
yhat = svm_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
```





Conclusions

From these data, we can say that:

- All method has same classification accuracy
- There is a definite upward pattern of success rate of SpaceX launches from 2013 to 2020.
- ES-L1, GEO, HEO and SSO orbits have the highest success rate for SpaceX launches
- The total payload carried by boosters from NASA is 22007kg
- When compared to high weighted payloads, light weighted which below 4,000 kg performed better.

Thank you!