IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Liyan Tian
03/06/2022

Outline

Executive Summary

Summary of methodologies

- Data Collection
- Data Wrangling
- EDA with Data Visualization
- EDA with SQL
- Building Interactive Map with Folium
- Building Interactive Dashboard with Ploty Dash
- Predictive Analysis (Classification)

Summary of all results

- Exploratory Data Analysis results
- Interactive Visual Analytics results
- Prediction results

Introduction

Project background and context

- SpaceX advertises Falcon 9 rocket launches on its website, with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage. Therefore if we can determine if the first stage will land, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch. In this project, we will predict if the Falcon 9 first stage will land successfully.

Problems you want to find answers

- What attributes are correlated with successful landings?
- How to predict if the first stage of Falcon 9 will land successfully?

Section 1

Methodology

Methodology

Data collection methodology:

- Data Collection with SpaceX REST API
- Data Collection with Web Scrapping from Wikipedia

Perform data wrangling

- Dropping irrelevant columns and applying OneHotEncoder to the categorical columns

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

- Data Collection with SpaceX REST API
  - SpaceX REST API gives 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.
  - The SpaceX REST API endpoints, or URL, starts with api.spacexdata.com/v4/. The end points can be different (We will be working with the endpoint api.spacexdata.com/v4/launches/past).

SpaceX REST API → Get Data in JSON → Normalize the structured json data into a flat table

- Data Collection with Web Scrapping from Wikipedia
  - Using the Python BeautifulSoup package to web scrape some HTML tables that contain valuable Falcon 9 launch records

Request Data from URL → Extract Data using BeautifulSoup → Convert HTML Tables into a Data Frame

Data Collection – SpaceX API

- Following are steps to collect data from SpaceX API:
  - 1. Getting response from API
  - 2. Converting the response into a Json object
  - 3. Cleaning data
  - 4. Converting data into data frame
  - 5. Filtering data and dealing with N/A values.
- Here is the GitHub URL of the completed SpaceX API calls notebook

1. Getting response from API

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

2. Decoding the response as a Json and convert it into a dataframe

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

3. Apply custom functions to clean data

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

4. Construct dataset into 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}
df = pd.DataFrame.from_dict(launch_dict)
```

5. Filter the dataframe and dealing with missing values

```
data_falcon9 = df.loc[df['BoosterVersion']!='Falcon 1']
mean = data_falcon9['PayloadMass'].mean()
data_falcon9['PayloadMass'] = data_falcon9['PayloadMass'].replace(np.nan, mean)
```

Data Collection – Scraping

- Following are steps to scrap data from Wikipedia:
  - 1. Getting response from HTML
  - 2. Creating BeautifulSoup object
  - 3. Finding tables
  - 4. Extracting column names
  - 5. Creating data frames
- Here is the GitHub URL of the completed Web Scraping notebook

1. Getting response from HTML

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

2. Creating BeautifulSoup object

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

3. Finding tables

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

4. Extracting 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. Creating Dictionary then dataframe

```
launch_dict= dict.fromkeys(column_names)
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']=[]
...
df = pd.DataFrame.from_dict(launch_dict)
```

Data Wrangling

- 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, ***False Ocean*** (unsuccessful landing to the ocean), ***False RTLS*** (unsuccessful landing to a ground pad), ***False ASDS*** (unsuccessful landing on a drone ship).
- We mainly convert those outcomes into Training Labels with 1 means the booster successfully landed 0 means it was unsuccessful.
- Following are steps of data wrangling process
  - Calculating the number of launches for each site
  - Calculating the number and occurrence of each orbit
  - Calculating the number and occurrence of mission outcome per orbit type
  - Creating a landing outcome label from Outcome column
- Here is the GitHub URL of the completed data wrangling notebook

EDA with Data Visualization

Scatter Plot: showing how much one variable is affected by another.

- FlightNumber vs. PayloadMass
- FlightNumber vs. LaunchSite
- LaunchSite vs. PayloadMass
- FlightNumber vs. Orbit
- PayloadMass vs. Orbit

Bar Chart: comparing data between different groups

- Mean vs. Orbit

Line Chart: showing data trends

- Success Rate vs. Year

Here is the GitHub URL of the completed EDA with data visualization notebook

EDA with SQL

Performed SQL queries:

- *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.*
- *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 between the date 2010-06-04 and 2017-03-20, in descending order*

Here is the GitHub URL of the completed EDA with SQL notebook

Build an Interactive Map with Folium

Mark all launch sites on a map

- Adding a circle and marker for each launch site

Mark the success/failed launches for each site on the map to see which sites have high success rates

- Creating markers for all launch records. If a launch was successful, then we use a green marker and if a launch was failed, we use a red marker
- Creating marker clusters to simplify a map containing many markers having the same coordinate.

Calculate the distances between a launch site to its proximities

- Adding a MousePosition on the map to get coordinate for a mouse over a point on the map
- Calculating the distance between the launch site and the coastline/Railway/Highway/City

Here is the GitHub URL of the completed interactive map with Folium notebook

Build a Dashboard with Plotly Dash

Pie Chart

- Showing the total success launches by all sites
- Showing the total success launches by a certain cite

Scatter Plot

- Showing the relationship between the Outcome and the Payload Mass (kg) for different Booster Versions

Here is the GitHub URL of the completed Plotly Dash lab notebook

Predictive Analysis (Classification)

Building Model

- Load data: data X contains independent variables; variable Y is the response variable.
- Standardize data in X
- Split data into training/testing datasets
- Select the machine learning methods: KNN, Decision Tree, SVM, Logistic Regression
- Build model and find hyperparameters using GridSearchCV function. Training model using training dataset

Evaluating Model

- Calculate the accuracy on the test data using the method score
- Plot Confusion Matrix

Improving Model

- Feature Engineering
- Algorithm Tunning

Finding the best performing classification model

- The model with the best accuracy score is the best performing model

Here is the GitHub URL of the completed predictive analysis notebook

Results

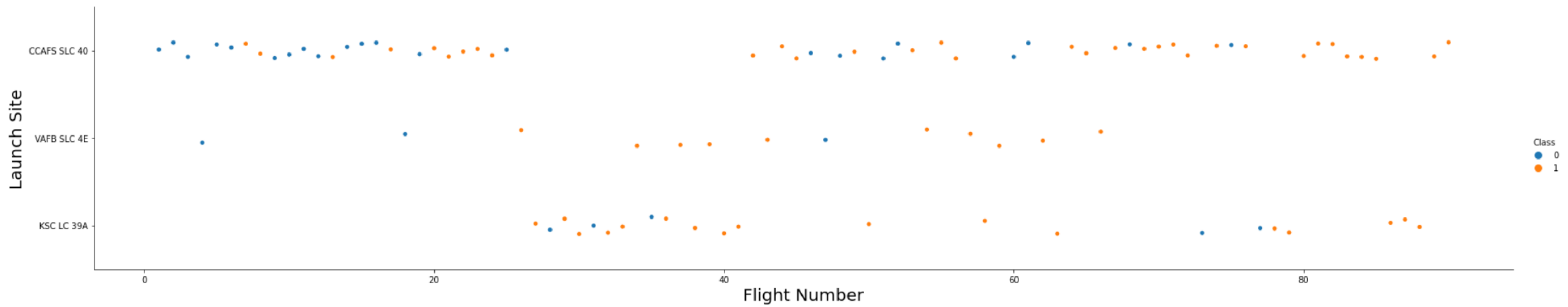
Exploratory data analysis results

Interactive analytics demo in screenshots

Predictive analysis results

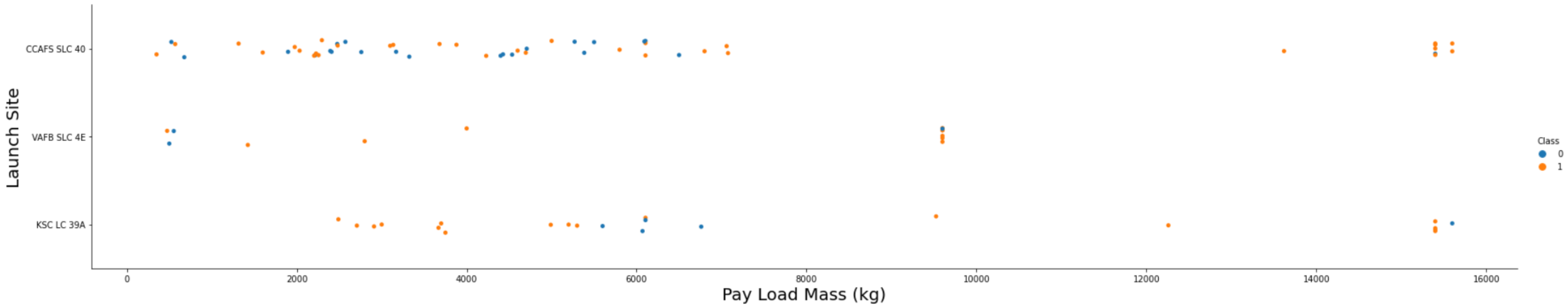
Section 2

Insights drawn from EDA



Flight Number vs. Launch Site

- Scatter plot of Flight Number vs. Launch Site
- We can see the more flight number at a launch site the greater the success rate at a launch site.

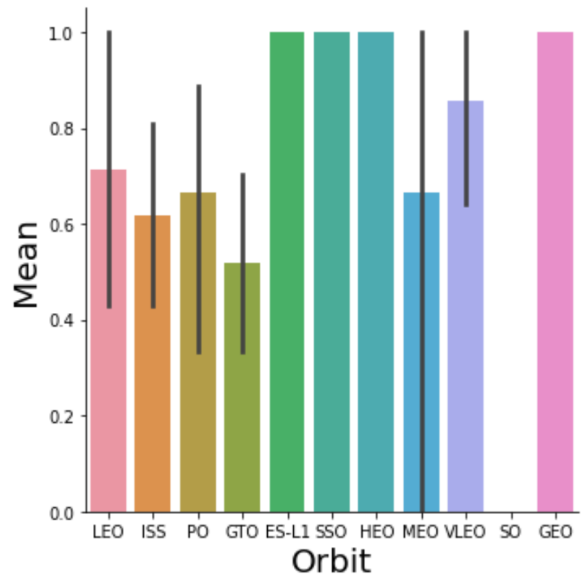


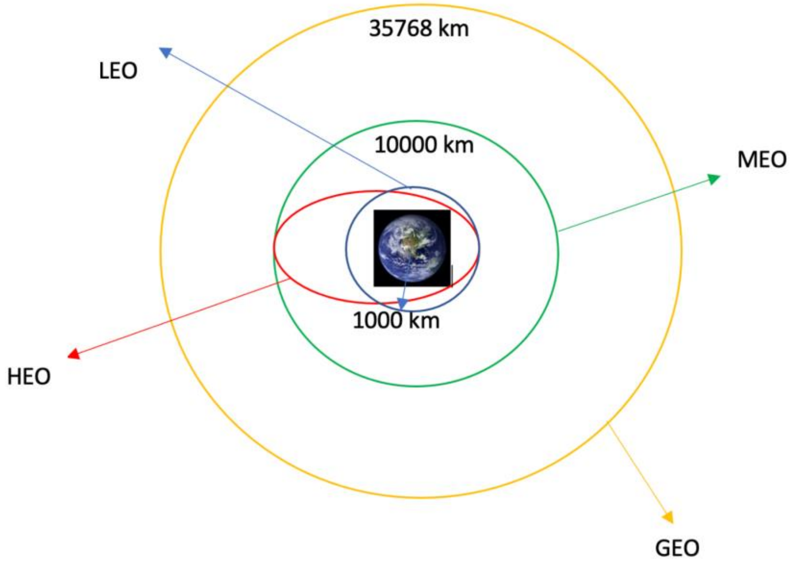
Payload vs. Launch Site

- Scatter plot of Payload vs. Launch Site
- For the VAFB-SLC launch site there are no rockets launched for heavy payload mass (greater than 10000).
- There is no clear pattern if the Payload Mass at a Launch Site has impact on the success rate.

Success Rate vs. Orbit Type

- Bar chart for the success rate of each orbit type
- Orbit ES-L1, SSO, HEO, GEO has the best success rate.

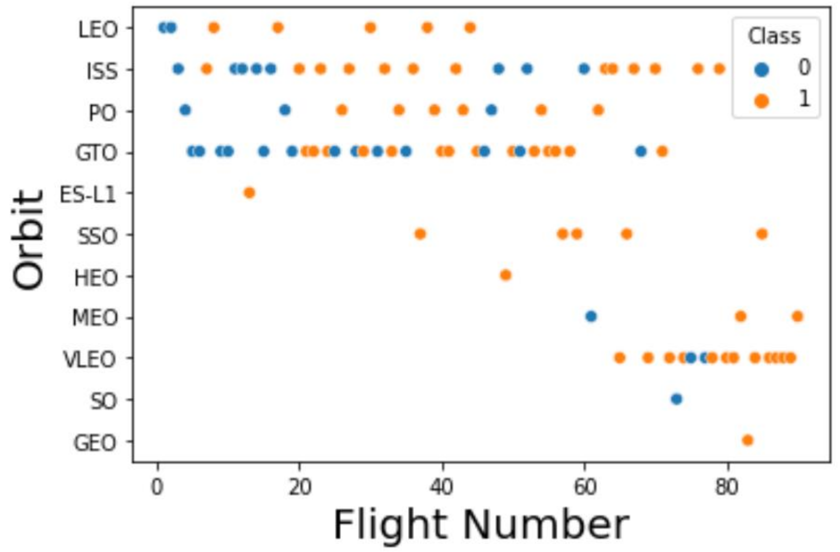




Flight Number vs. Orbit Type

- Scatter point of Flight number vs. Orbit type

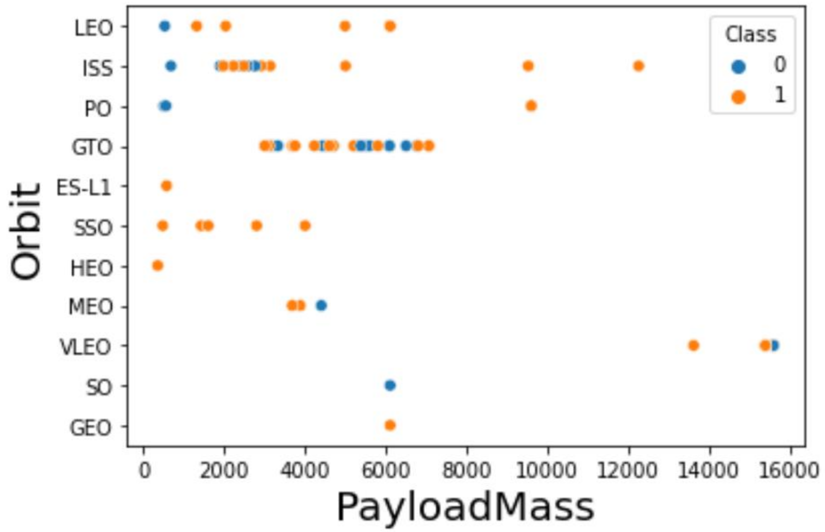
- In the LEO orbit, the success rate appear to be related to the number of flights.
- In the GTO orbit, there is no clear relation of the success rate and number of flight.



Payload vs. Orbit Type

- Scatter point of payload vs. orbit type

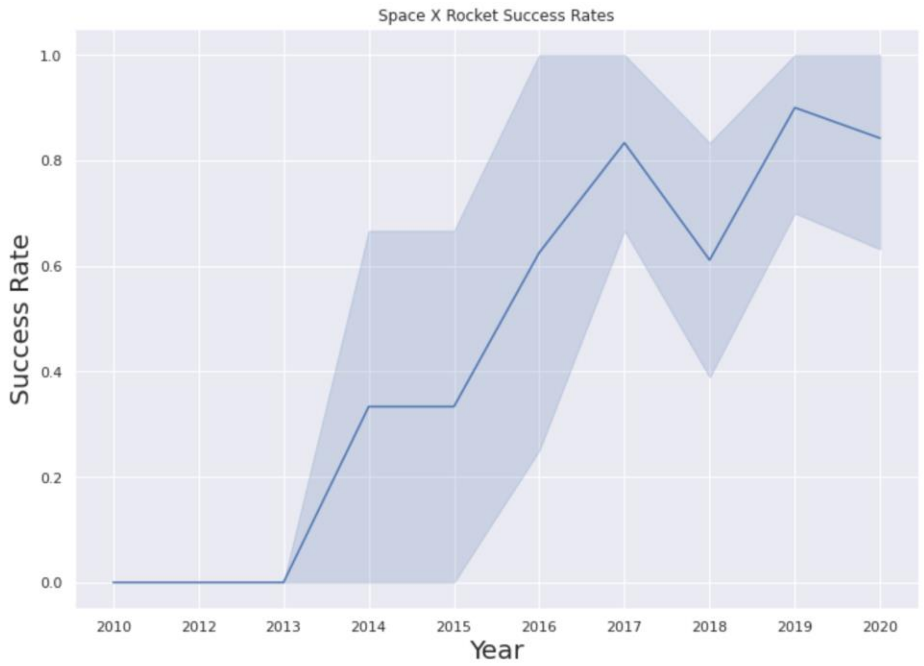
- Heavy payload has positive impact on success on the LEO/ISS/PO orbit.
- In the GTO orbit, there is no clear relation of the success rate and payload.



Launch Success Yearly Trend

- Line chart of yearly average success rate

- In general, the success rate has an increase trend since 2013. For some reason, the success rate decreased in 2018 and 2020.



All Launch Site Names

SQL Query:

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

Result:

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

The distinct keyword in the query returned the unique values of the Launch Site.

Launch Site Names Begin with 'CCA'

SQL Query:

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

Result:

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

The like keyword has a wild card with the word and percentage in the end means the launch site must start with CCA. The limit keyword means only showing 5 records.

Total Payload Mass

SQL Query:

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

Result:

1
45596

The sum function summarize the total payload mass. The where clause filters the dataset to only perform calculation on customer NASA(CRS).

Average Payload Mass by F9 v1.1

SQL Query:

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

Result:

1
2928

The avg function calculates the average payload mass. The where clause filters the dataset to only perform calculation on F9 V1.1 Booster.

First Successful Ground Landing Date

SQL Query:

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

Result:

The min function performed on Date can return the first date. The where clause filters the dataset to only perform function on Success Ground Landing.

1
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

SQL Query:

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

Result:

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

The where clause can contain several conditions to do the filter.

Total Number of Successful and Failure Mission Outcomes

SQL Query:

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

Result:

The count function can return the total number of the outcomes.

1
100

Boosters Carried Maximum Payload

SQL Query:

```
select BOOSTER_VERSION
from SPACEXTBL
where PAYLOAD_MASS__KG_
=
(select max(PAYLOAD_MASS_
_KG_) from SPACEXTBL)
```

Result:

Subquery can be used in where clause to filter the payload mass to the max payload mass.

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

2015 Launch Records

SQL Query:

```
select
BOOSTER_VERSION,
LAUNCH_SITE from
SPACEXTBL
where Landing__Outcome
 = 'Failure (drone
ship)' and year(DATE)
= 2015
```

Result:

booster_version	launch_site
F9 v1.1 B1012	CCAFS LC-40
F9 v1.1 B1015	CCAFS LC-40

Year() function can be performed on date to get the year of the date.

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

SQL Query:

select LANDING__OUTCOME, count(*) as countnum from SPACEXTBL
where DATE between '2010-06-04' and '2017-03-20'
group by LANDING__OUTCOME
order by count(*) desc

Result:

landing__outcome	countnum
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

Group by keyword groups the data into different groups. Order by keyword sorts output in descending/ascending order.

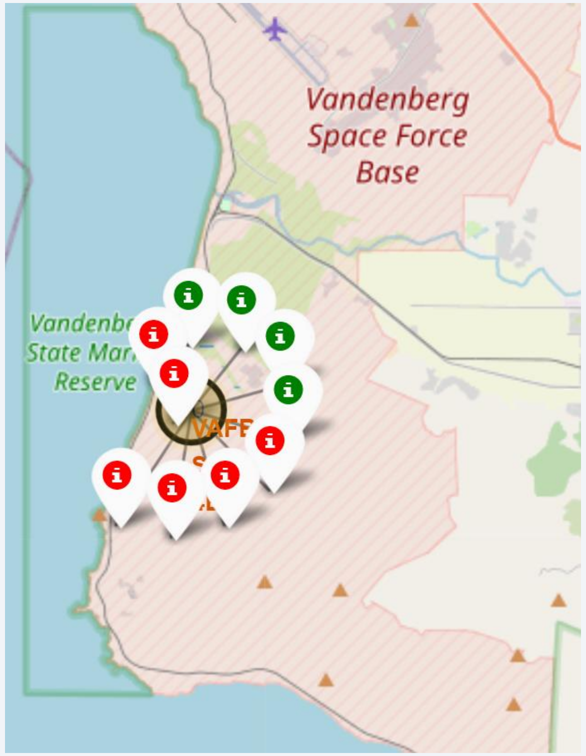
Section 3

Launch Sites Proximities Analysis

All Launch Sites Global Map

SpaceX launch sites all locate in Florida/California US

Color-labeled launch outcomes Map

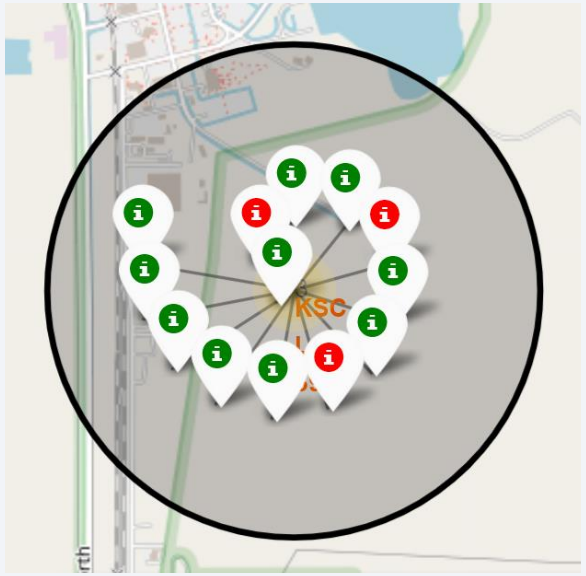


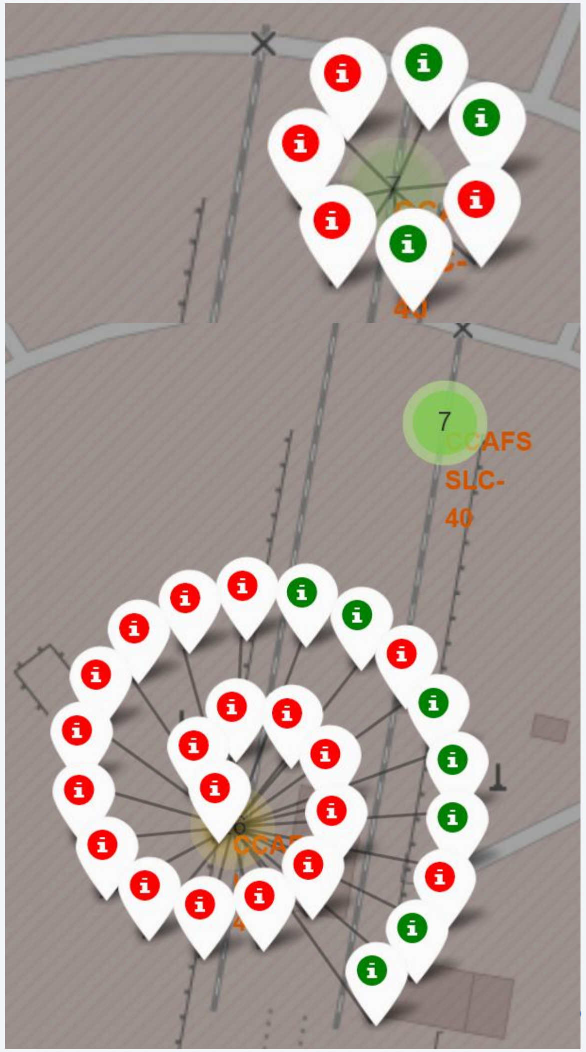
Green marker:

successful launches

Red marker:

failed launches

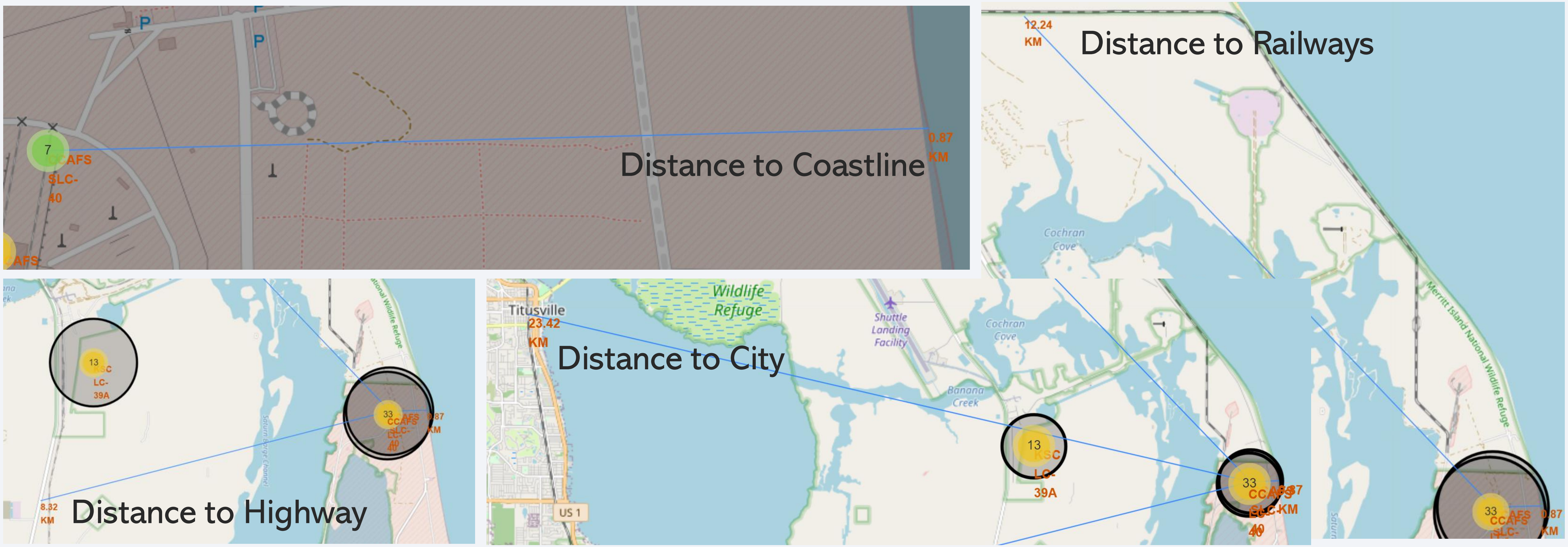




California Launch Site

Florida Launch Sites

Distances between a launch site to its proximities

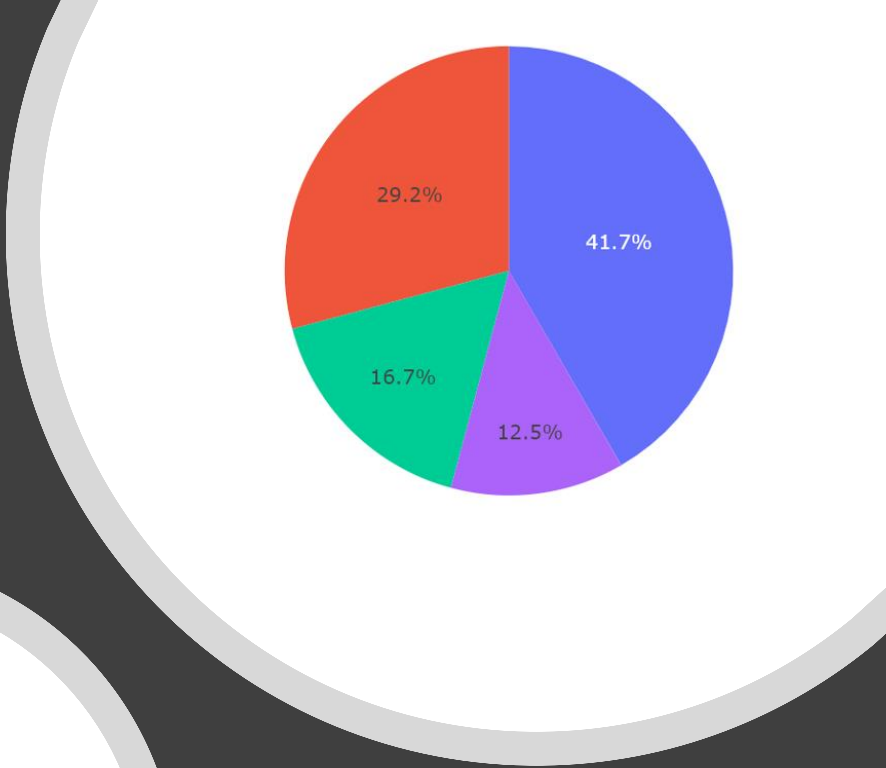


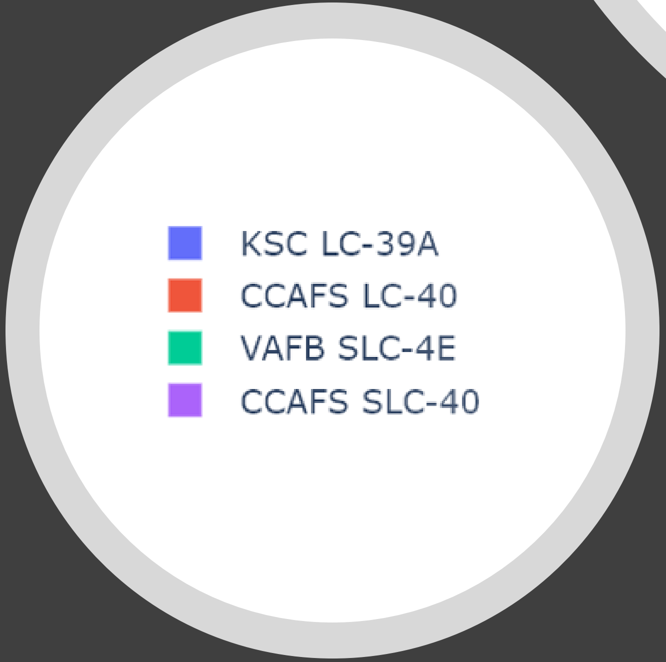
- Are launch sites in close proximity to railways? No
- Are launch sites in close proximity to highways? No
- Are launch sites in close proximity to coastline? Yes
- Do launch sites keep certain distance away from cities? Yes

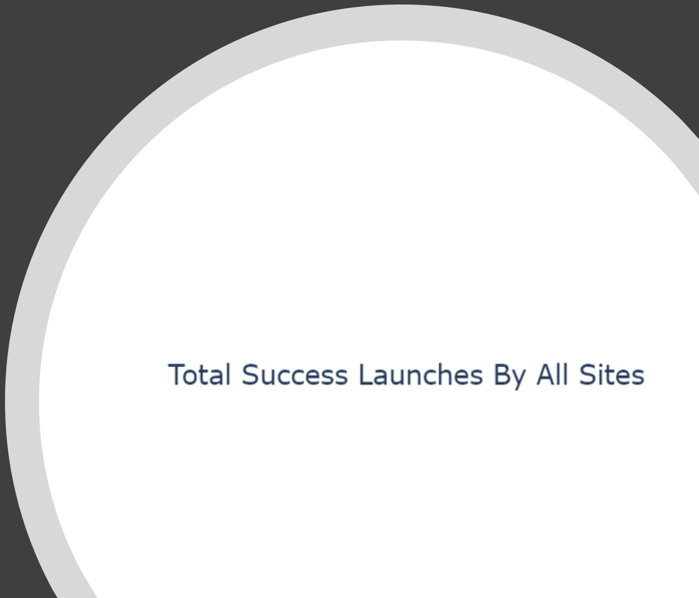
Section 4

Build a Dashboard with Plotly Dash

Pie chart of launch success count for all sites

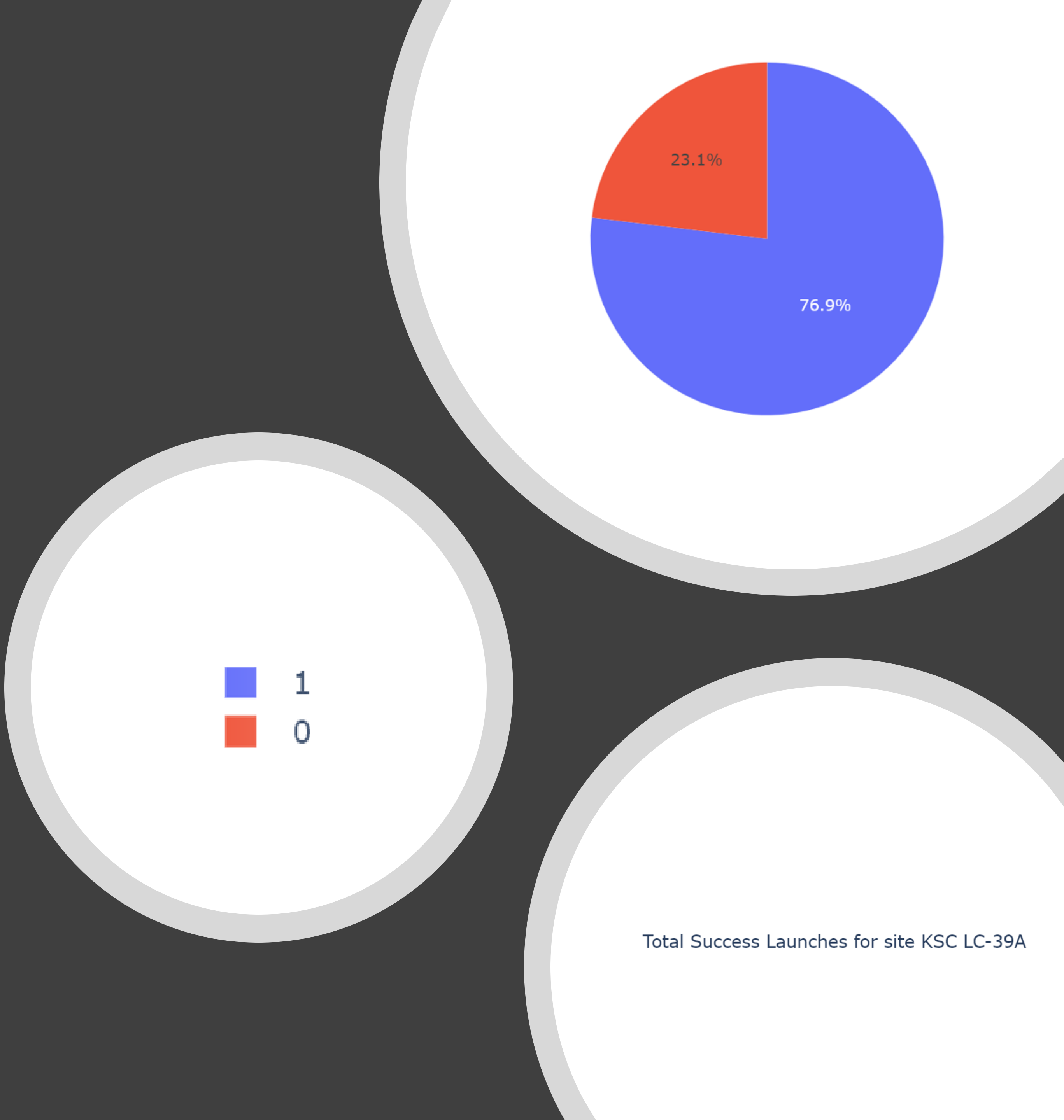






- KSC LC-39A has the most successful launches from all sites.

Pie chart for the launch site with highest launch success ratio

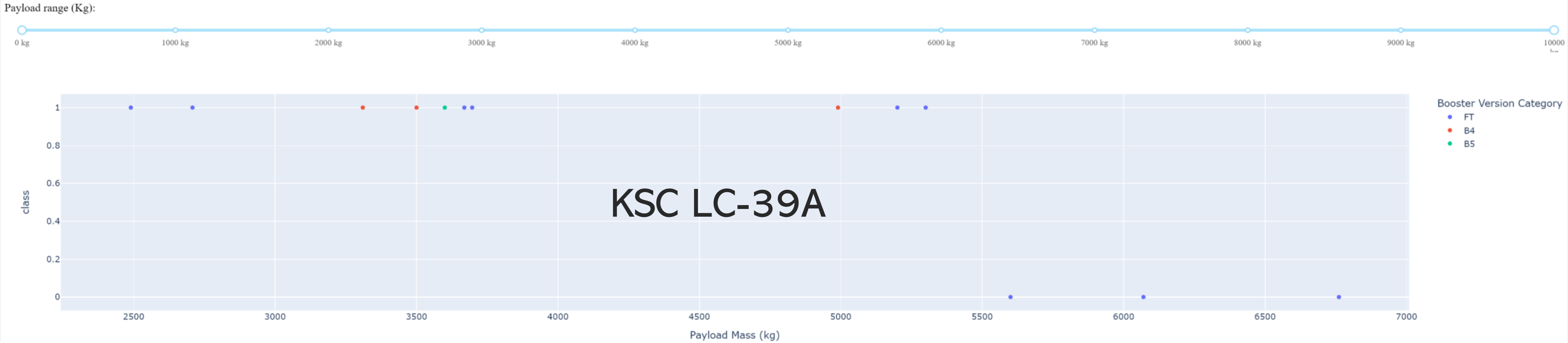


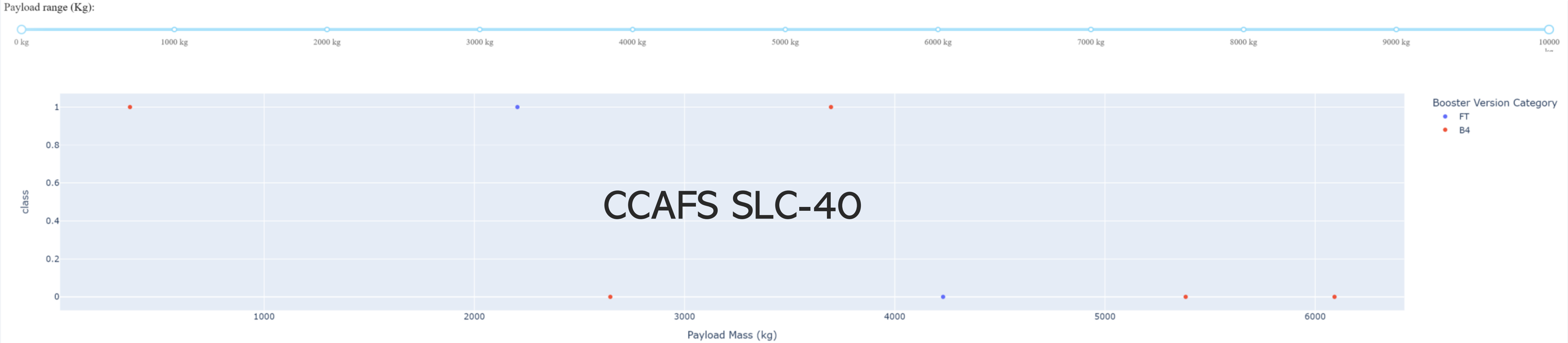
- KSC LC-39A has a 76.9% success rate and 23.1% failure rate.

Scatter plot of Payload vs. Launch Outcome for all sites

- There is no clear pattern of payload vs. Launch outcome.

Scatter plot of Payload vs. Launch Outcome for all sites





- At these two sites, low weighted payloads has a higher success rate than heavy weighted payloads.

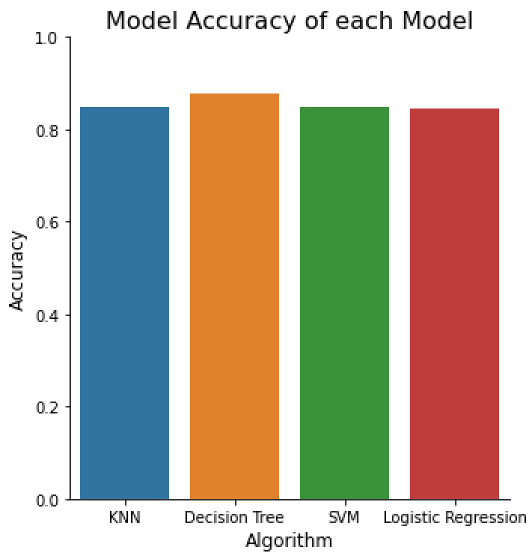
Section 5

Predictive Analysis (Classification)

Method	Score
Decision Tree	0.876786
KNN	0.848214
SVM	0.848214
Logistic Regression	0.846429

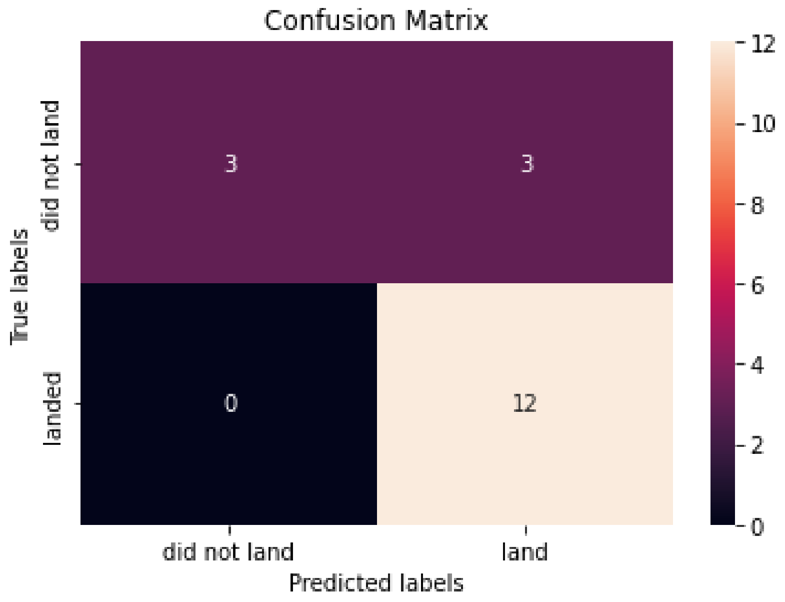
Classification Accuracy

- All built classification models have very close accuracy.
- Decision Tree model has highest accuracy.
- After selecting the best hyperparameters for the decision tree using the validation data, the decision tree achieved 87.68% accuracy on test data.



Confusion Matrix

- The confusion matrix of the decision tree model shows the model can distinguish between the different classes.
- In the test data, there are 12 cases of success landing and 6 cases of failed landing.
- The predicted labels show the model recognized 3 of the failed landing as success landing. So, there are 3 False Positive predicts.



Conclusions

- SpaceX launch sites all locate in Florida/California, US close to coastlines.
- The launch sites keep certain distance away from cities.
- The success rate for SpaceX Launches has an increase trend since 2013.
- KSC LC-39A launch site has the most successful launches from all sites.
- Low weighted payloads have higher success rate than heavy weighted payloads.
- Orbit ES-L1, SSO, HEO, GEO have the best success rate.
- In the LEO orbit, the success rate appears to be related to the number of flights.
- The more flight number at a launch site the greater the success rate at a launch site.
- The Decision Tree classification method performs best.

Appendix

[Data Collection with API.ipynb]

[Data Collection with Web Scrapping.ipynb]

[Data Wrangling.ipynb]

[EDA with Visualization lab.ipynb]

[EDA with SQL.ipynb]

[Data Visualization with Folium.ipynb]

[Data Visualization with Ploty Dash.py]

[Machine Learning Prediction.ipynb]

Thank you!