IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Benedict Yong
12/07/2022

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection
  - Data Wrangling
  - EDA with Data Visualization
  - EDA with SQL
  - Building an interactive map with Folium
  - Building a dashboard with Plotly Dash
  - Predictive Analysis (Classification)
- Summary of all results
  - EDA Results
  - Interactive Analytics
  - Predictive Analysis

Introduction

- Project background and context
  - SpaceX is a revolutionary company who has disrupted the space industry by offering rocket launches specifically Falcon 9 as low as 62 million dollars; while other providers cost upward of 165 million dollar each. Most of this savings is thanks to SpaceX astounding idea to reuse the first stage of the launch by re-landing the rocket to be used on the next mission. Repeating this process will make the price decrease even further. As a data scientist of a start up rivaling SpaceX, the goal of this project is to create a machine learning pipeline to predict the landing outcome of the first stage in the future. This project is crucial in identifying the right price to bid against SpaceX for a rocket launch.
- Problems we want to find answers to;
  - Identifying all factors that influence the landing outcome
  - The relationship between each variables and how it is affecting the outcomes
  - The best conditions needed to increase the probability of a successful landing

Section 1

Methodology

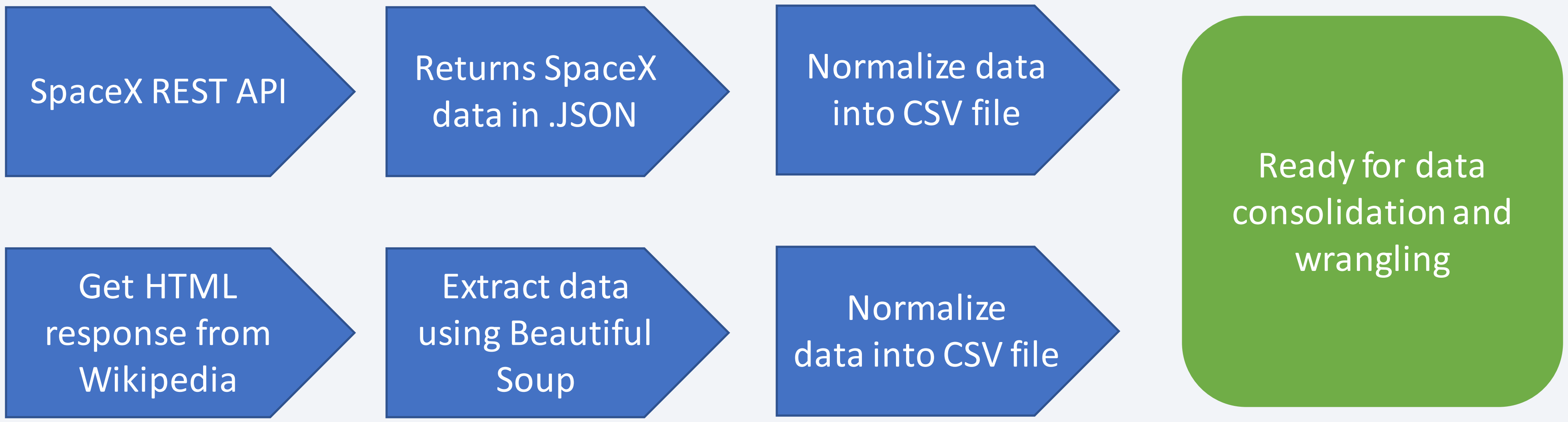
Methodology

Executive Summary

- Data collection methodology:
  - SpaceX rest API
  - Web Scrapping from Wikipedia
- Perform data wrangling
  - One Hot Encoding data fields for Machine Learning and data cleaning of null values and irrelevant 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
  - LR, KNN, SVM, DT models have been built and evaluated for the best classifier

Data Collection

- Describe how data sets were collected.



Data Collection – SpaceX API

- Data collection with SpaceX REST API calls

1. Getting response from API

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

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

2. Converting response into .json file

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

3. Cleaning data

```
# Lets take a subset of our dataframe keeping only the features we want and the flight number, and date_utc.
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 rocket boosters and rows that have multiple payloads
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 the list and replace the feature.
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 date leaving the time
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)]
```

4. Assigning 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 csv file

```
# Create a data from launch_dict
launch_data= pd.DataFrame.from_dict(launch_dict)
```

```
data_falcon9 = launch_data.loc[launch_data.BoosterVersion=='Falcon 9']
```

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

•Add the GitHub URL of the completed SpaceX API calls notebook (must include completed code cell and outcome cell), as an external reference and peer-review purpose

Data Collection - Scraping

- Web Scrapping Process

1. Getting response from HTML

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

page = requests.get(static_url)
```

2. Creating Beautiful Soup Object

```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content

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

3. Finding tables

```
# Use the find_all function in the BeautifulSoup object, with element type `table`
# Assign the result to a list called `html_tables`

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

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. Creating 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 notebook block 19)

```
[19]: 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"):
        #check to see if first table heading is as number corresponding to launch a number
        if rows.th:
            if rows.th.string:
                flight_number=rows.th.string.strip()
                flag=flight_number.isdigit()
        else:
            flag=False
```

7. Converting dictionary to dataframe

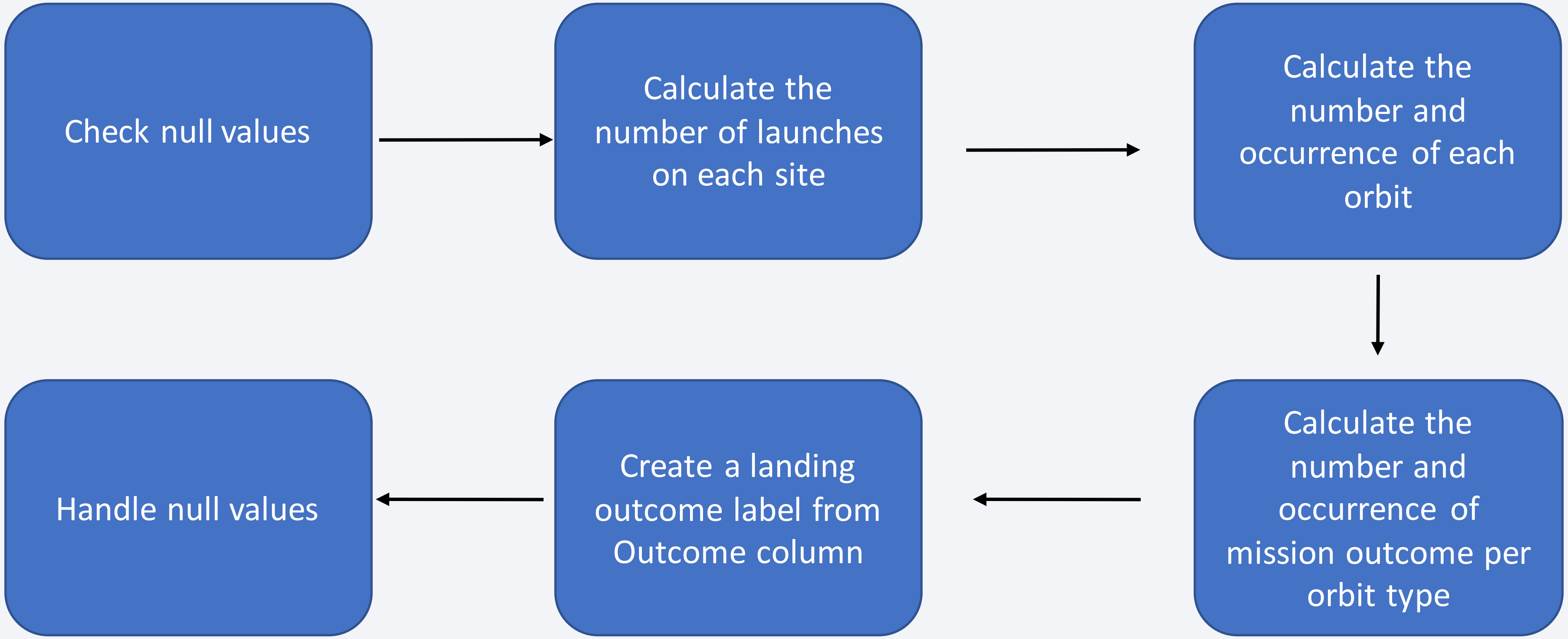
```
df=pd.DataFrame(launch_dict)
```

8. Dataframe to CSV file

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

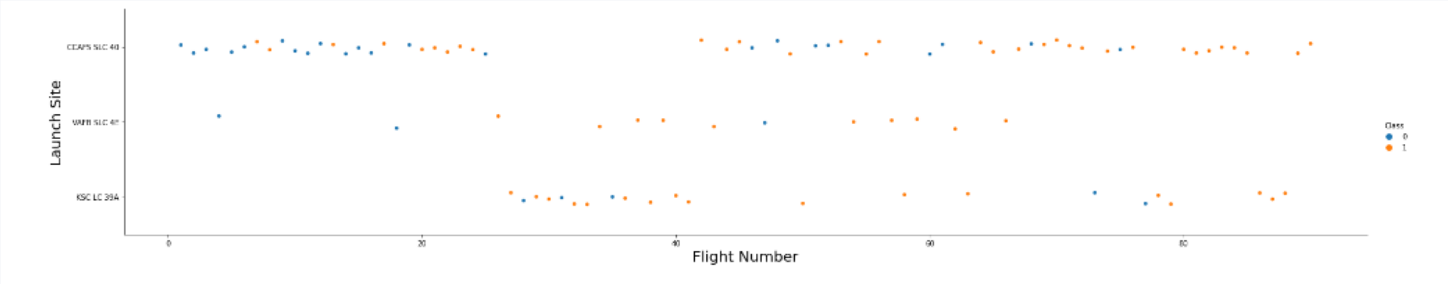
Add the GitHub URL of the completed web scraping notebook, as an external reference and peer-review purpose

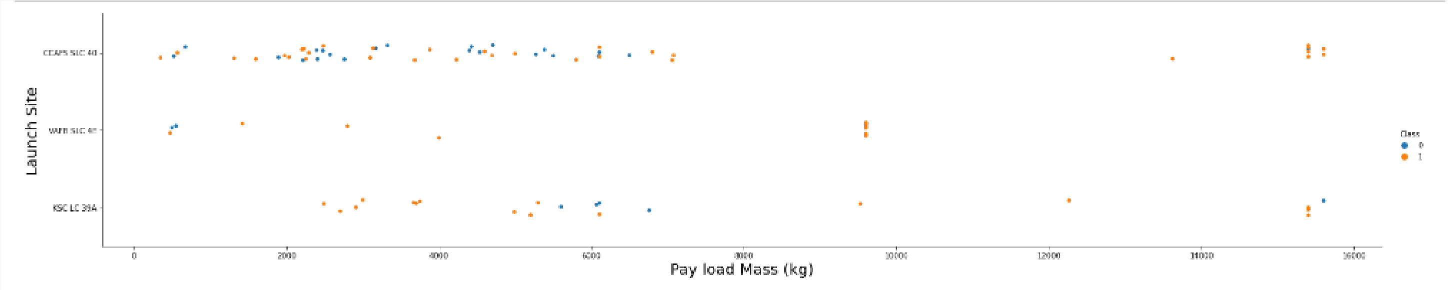
Data Wrangling



•Add the GitHub URL of your completed data wrangling related notebooks, as an external reference and peer-review purpose

EDA with Data Visualization

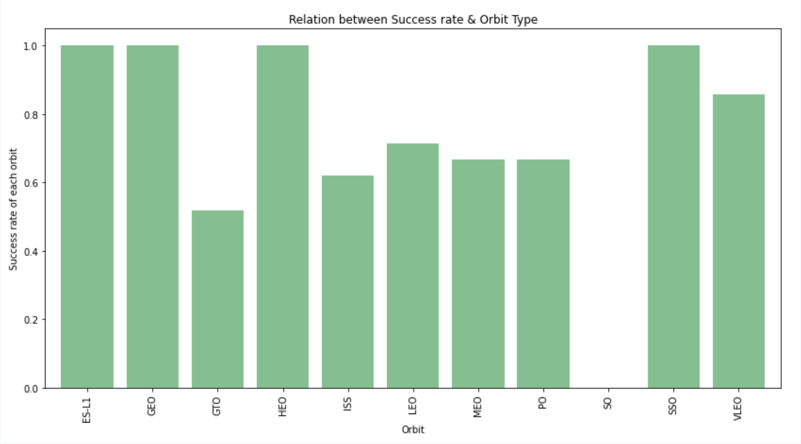


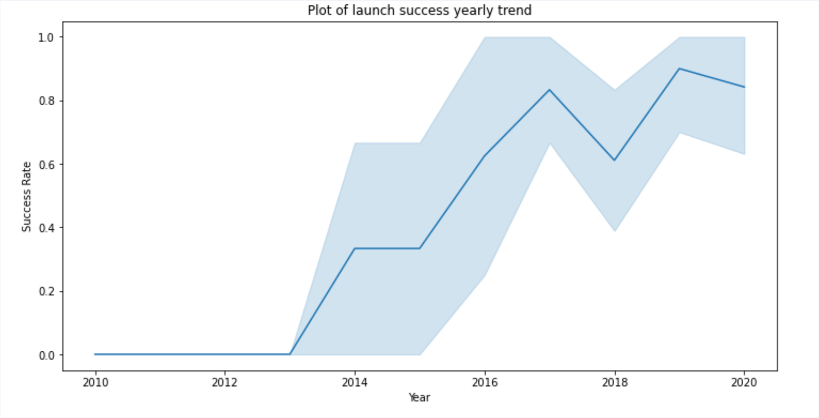


- Using scatterplots to visualize relationship between attributes;
  - Payload & Flight Number
  - Flight Number & Launch Site
  - Payload & Launch Site
  - Flight Number & Orbit Type
  - Payload & Orbit Type

•Add the GitHub URL of your completed EDA with data visualization notebook, as an external reference and peer-review purpose

EDA with Data Visualization





- Use bar graph to determine which orbits have the highest probability of success
- Follow with a line graph to show trends or pattern of attribute over time (launch success yearly trend)

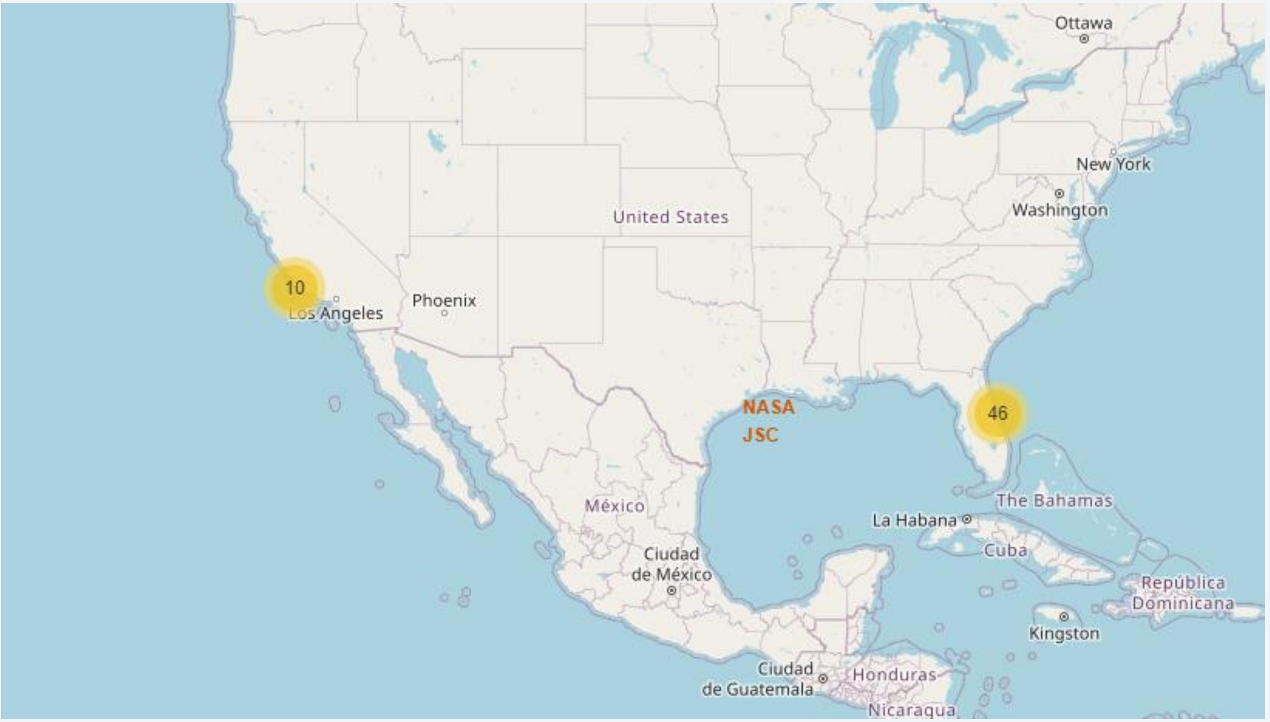
•Add the GitHub URL of your completed EDA with data visualization notebook, as an external reference and peer-review purpose

EDA with SQL

- Display names of unique launch sites in space mission
- Display 5 records where launch sites begin with string 'CCA'
- Display total payload mass carried by boosters launched by NASA (CRS)
- Display average payload mass carried by booster version f9 V1.1
- List date when the first successful landing outcome in ground pad was achieved
- List names of boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
- List total number of successful and failure mission outcomes
- List names of booster versions which have carried maximum payload mass using a subquery
- List failed landing outcomes in drone ship, their booster versions and launch site names in 2015
- Rank count of landing outcomes between 2010/06/04 and 2017/03/20 in descending order

•Add the GitHub URL of your completed EDA with SQL notebook, as an external reference and peer-review purpose

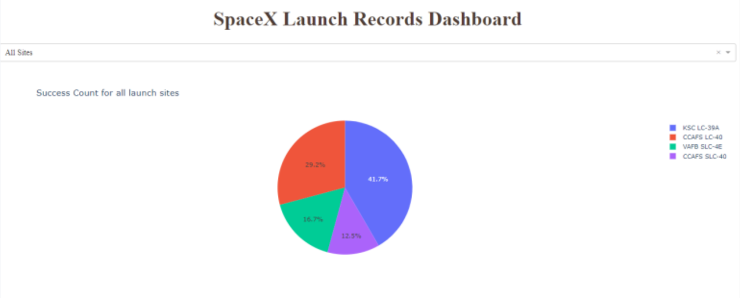
Build an Interactive Map with Folium

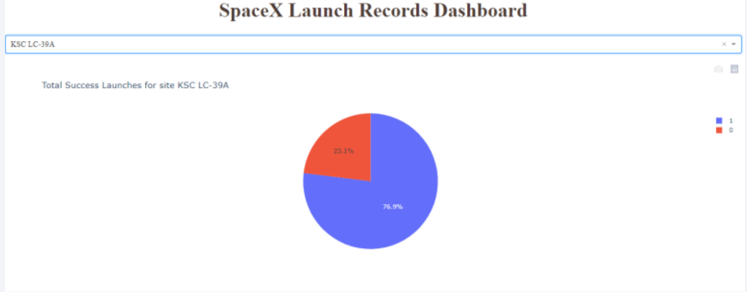


- Map objects added;
  - Markers
  - MarkerClusters
  - Polylines
- Map markers and distance lines have been added to the map to find an optimal location for building a launch site

•Add the GitHub URL of your completed interactive map with Folium map, as an external reference and peer-review purpose

Build a Dashboard with Plotly Dash





- Built an interactive dashboard with Plotly Dash allowing users to explore the data as required
- Plotted pie charts to show total launches by certain sites
- Plotted scatter graph showing relationship between Outcomes and Payload Mass (kg) for different booster version

•Add the GitHub URL of your completed Plotly Dash lab, as an external reference and peer-review purpose

Predictive Analysis (Classification)

Build Model
1) Load datasets into Numpy & Pandas
2) Transform data and split into train and test sets
3) Determine which ML type to use
4) Set parameters and algorithms to GridSearchCV and fit to dataset

Evaluate Model
1) Check accuracy for each model
2) Get tuned hyperparameters for each type of algorithm
3) Plot Confusion Matrix

Find the Best Model
1) Compare best accuracy score to find the best performing model

Add the GitHub URL of your completed predictive analysis lab, as an external reference and peer-review purpose

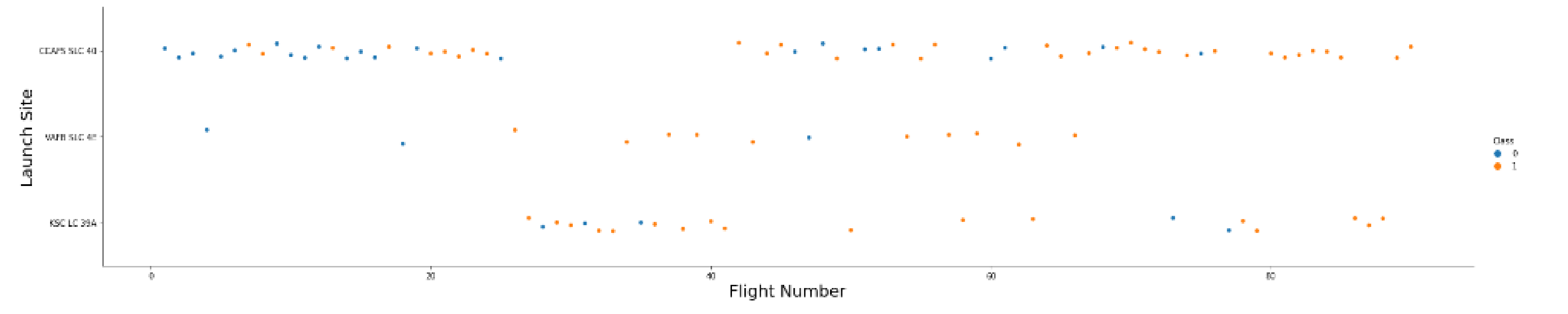
Results

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

Section 2

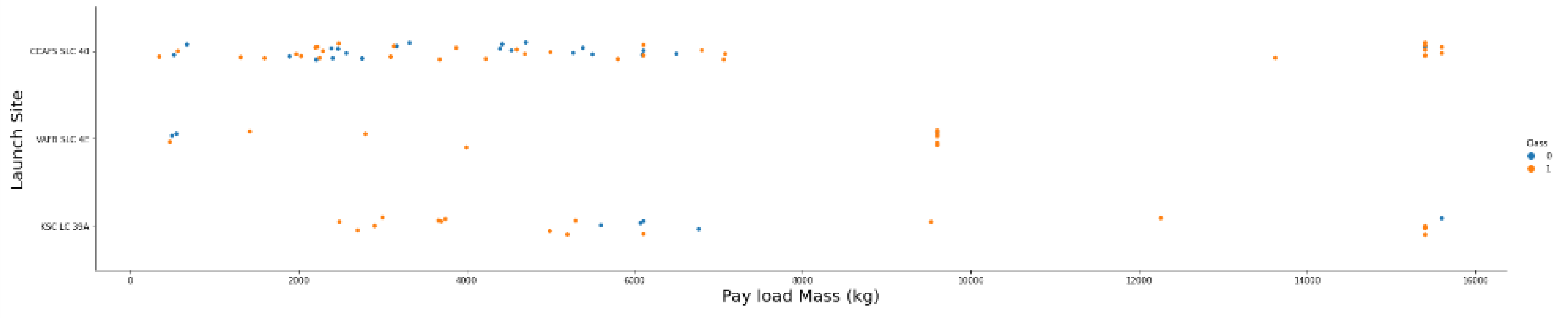
Insights drawn from EDA

Flight Number vs. Launch Site



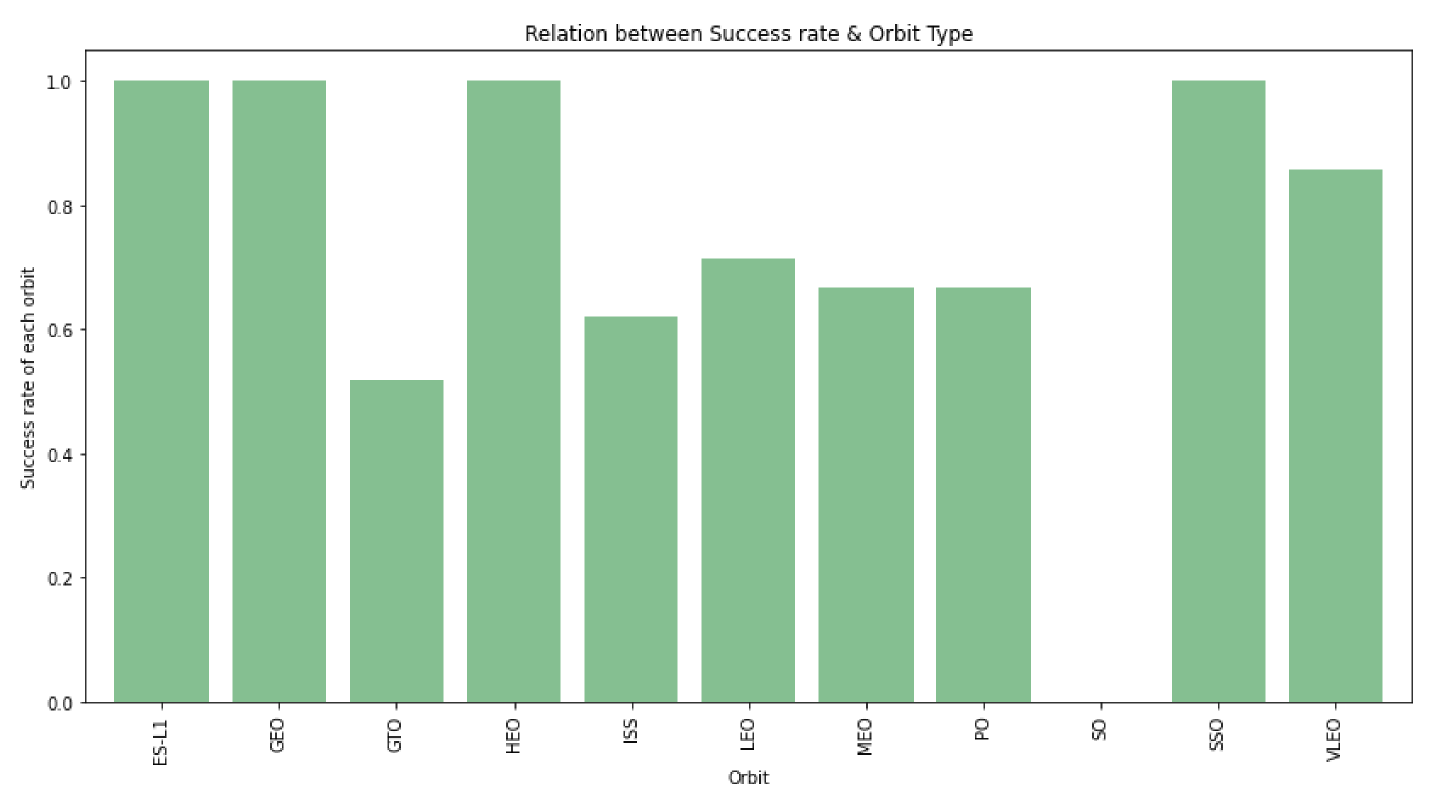
- With a higher number of flights at the launch sites, we observe that the success rate generally increases

Payload vs. Launch Site



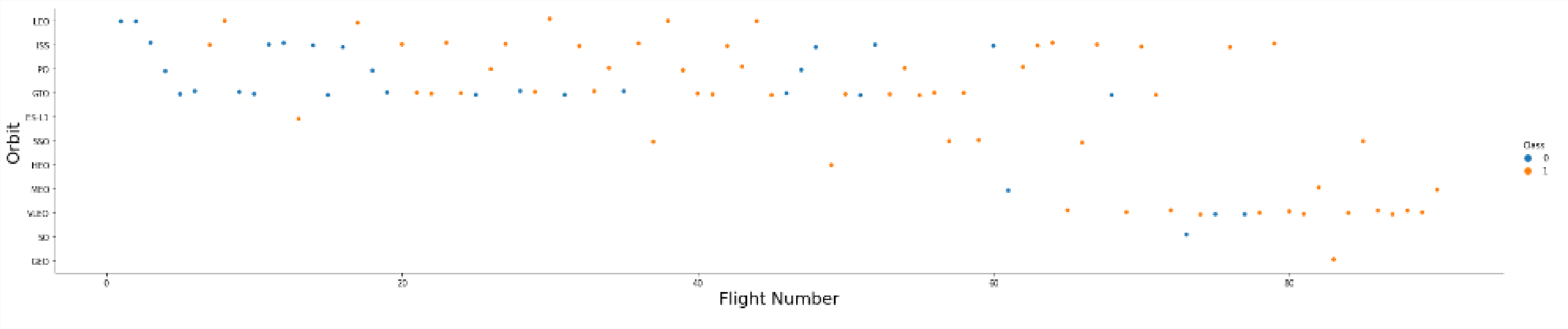
- There are no rockets launched for heavy payload mass greater than 10000 for VAFB_SLC launch site
- Once payload mass is greater than 7000kg, the probability of success rate is highly increased

Success Rate vs. Orbit Type



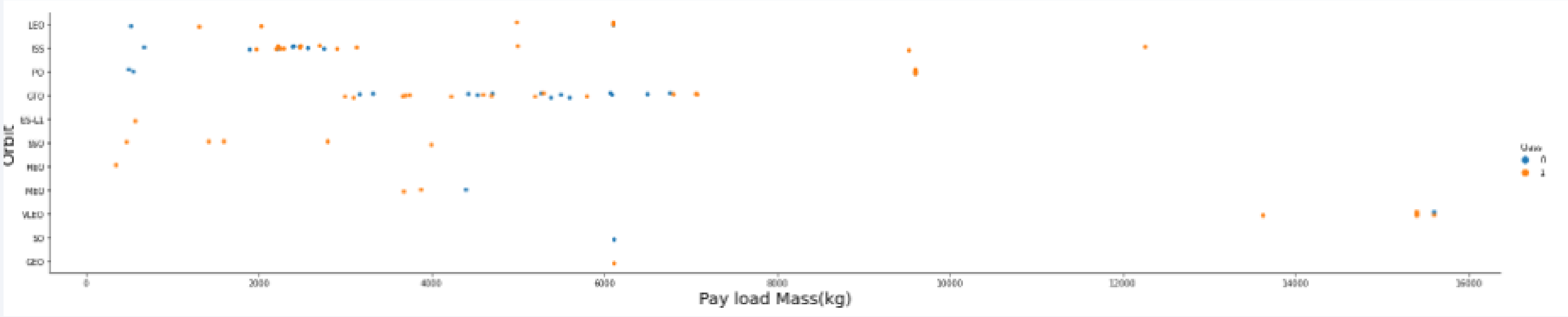
- The orbits ES-L1, GEO, HEO, SSO are among the highest success rate while SO produced 0% success rate

Flight Number vs. Orbit Type



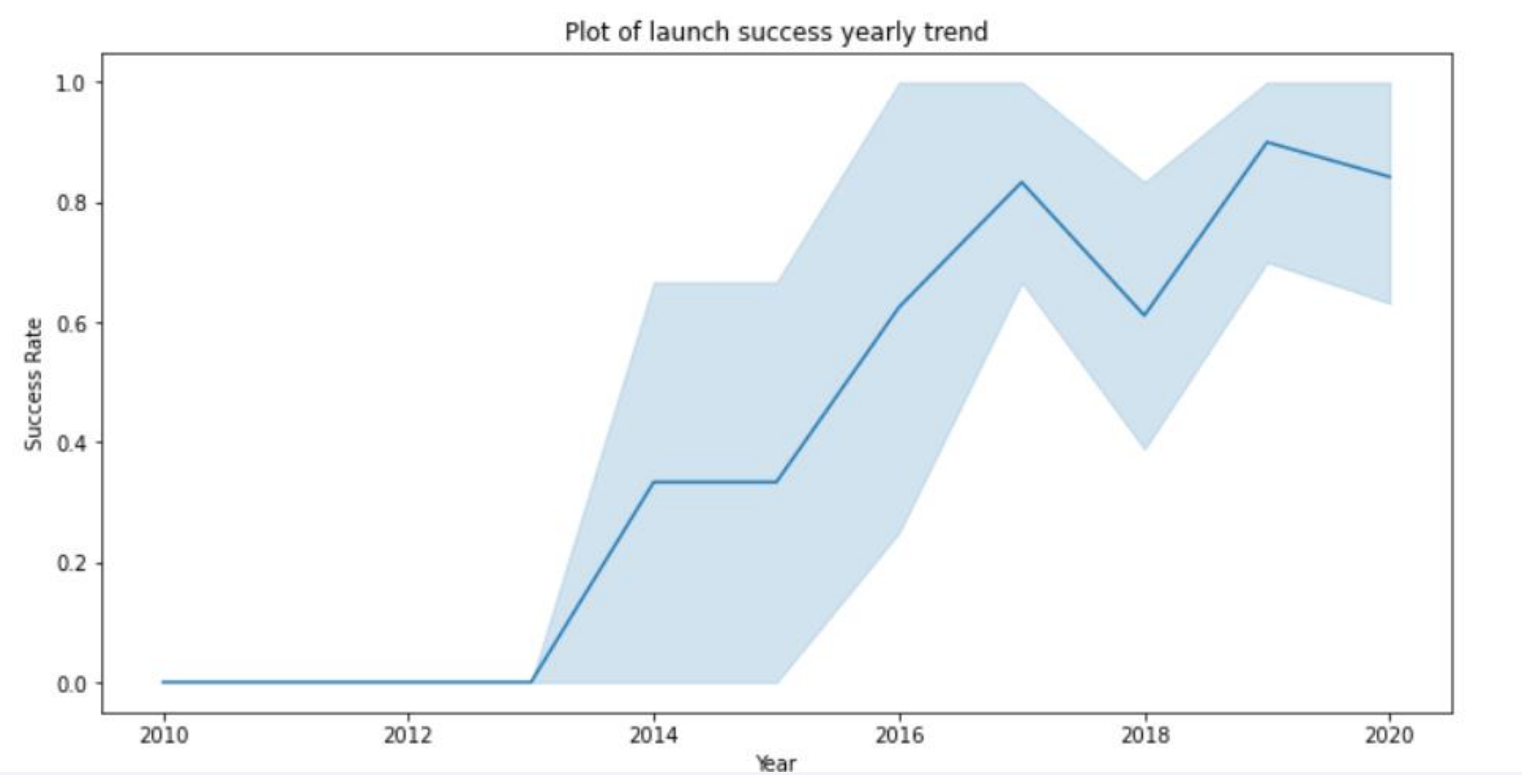
- Generally the larger the flight number on each orbit, the higher the success rate

Payload vs. Orbit Type



- With heavy payloads the successful landing rate are more for Polar,LEO and ISS.
- However for GTO we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission) are both there here.

Launch Success Yearly Trend



- Success rate has been increasing since 2013 till 2020

All Launch Site Names

- We used the function DISTINCT to show only unique launch sites from the data

```
%sql SELECT DISTINCT LAUNCH_SITE FROM SPACEXDATASET ORDER BY 1
```

```
 * ibm_db_sa://qdd61416:***@764264db-9824-4b7c-82df-40d1b13897c2.
Done.
```

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

Launch Site Names Begin with 'CCA'

```
%sql SELECT * FROM SPACEXDATASET WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5
```

```
 * ibm_db_sa://qdd61416:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/bludb
Done.
```

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	None	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	None	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-05-22	07:44:00	F9 v1.0 B0005	CCAFS LC-40	None	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	00:35:00	F9 v1.0 B0006	CCAFS LC-40	None	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	None	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

Display the total payload mass carried by boosters launched by NASA (CRS)

```
%sql SELECT SUM(payload_mass__kg_) from SPACEXDATASET WHERE CUSTOMER = 'NASA (CRS)'
```

```
 * ibm_db_sa://qdd61416:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8
oud:32536/bludb
Done.
```

]:

1
45596

Average Payload Mass by F9 v1.1

Display average payload mass carried by booster version F9 v1.1

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXDATASET WHERE BOOSTER_VERSION LIKE 'F9 v1.1 %'
```

```
 * ibm_db_sa://qdd61416:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdom
Done.
```

4]:

1
2337

First Successful Ground Landing Date

```
%sql SELECT MIN(DATE) FROM SPACEXDATASET WHERE LANDING__OUTCOME = 'Success (ground pad)'
```

```
 * ibm_db_sa://qdd61416:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.c
536/bludb
Done.
```

1
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%sql SELECT DISTINCT BOOSTER_VERSION FROM SPACEXDATASET WHERE PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000 AND LANDING__OUTCOME = 'SUCCESS (DRONE SHIP)';

 * ibm_db_sa://qdd61416:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/bludb
Done.
```

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

Total Number of Successful and Failure Mission Outcomes

List the total number of successful and failure mission outcomes

```
%sql SELECT MISSION_OUTCOME, COUNT(*) FROM SPACEXDATASET GROUP BY MISSION_OUTCOME
```

```
 * ibm_db_sa://qdd61416:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lc
Done.
```

]:

mission_outcome	2
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

Boosters Carried Maximum Payload

```
%sql SELECT  BOOSTER_VERSION, PAYLOAD_MASS__KG_ FROM SPACEXDATASET WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXDATASET)
```

```
 * ibm_db_sa://qdd61416:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/bludb
Done.
```

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

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 SPACEXDATASET WHERE LANDING__OUTCOME = 'Failure (drone ship)' AND DATE LIKE '2015%'
```

```
 * ibm_db_sa://qdd61416:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/bludb
Done.
```

5]:

landing__outcome	booster_version	launch_site
Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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

```
%sql SELECT LANDING__OUTCOME, COUNT(*) AS QTY FROM SPACEXDATASET WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY LANDING__OUTCOME ORDER BY QTY DESC
```

```
 * ibm_db_sa://qdd61416:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/bludb
Done.
```

landing__outcome	qty
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

Section 3

Launch Sites Proximities Analysis

Location of Launch Sites

- We can see that all launch sites are close to the coastlines

Markers showing launch sites with color labels

Florida Launch Site

California Launch Site

- Green Markers – Successful launches | Red Markers - Failures

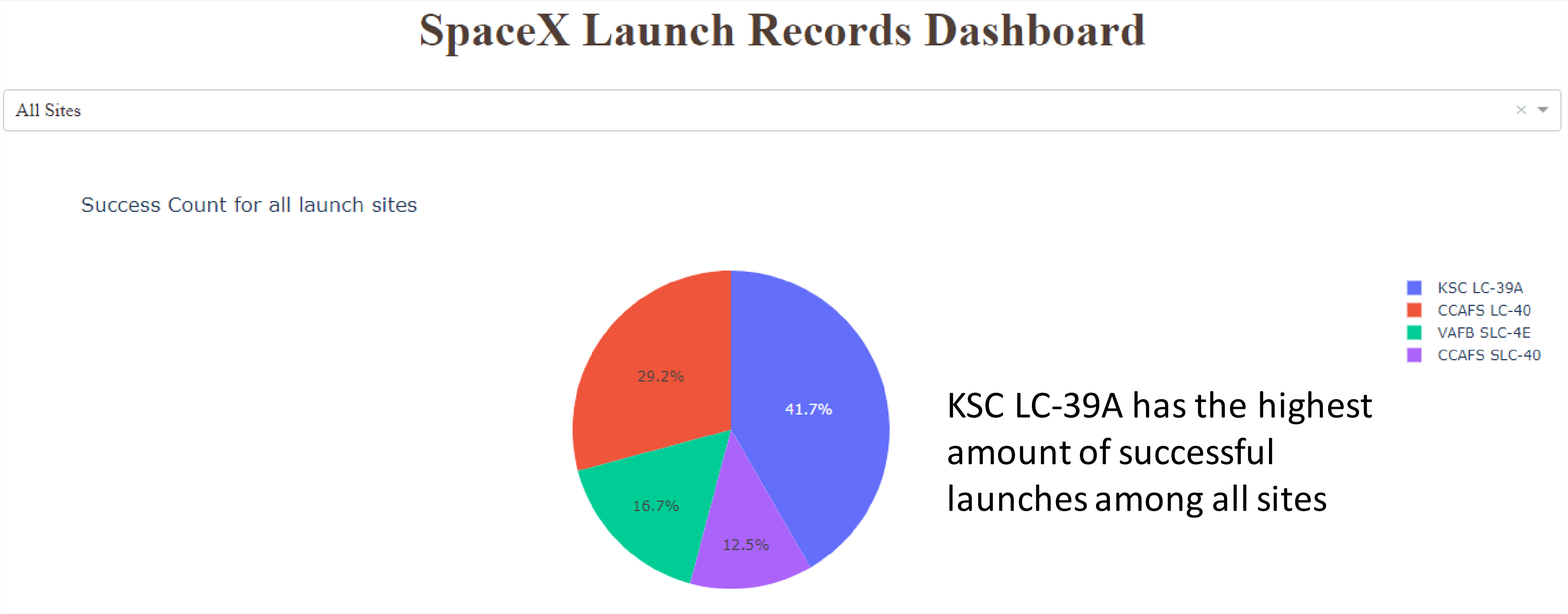
Launch sites proximities to landmarks

- Distance of Florida launch site to coastline

Section 4

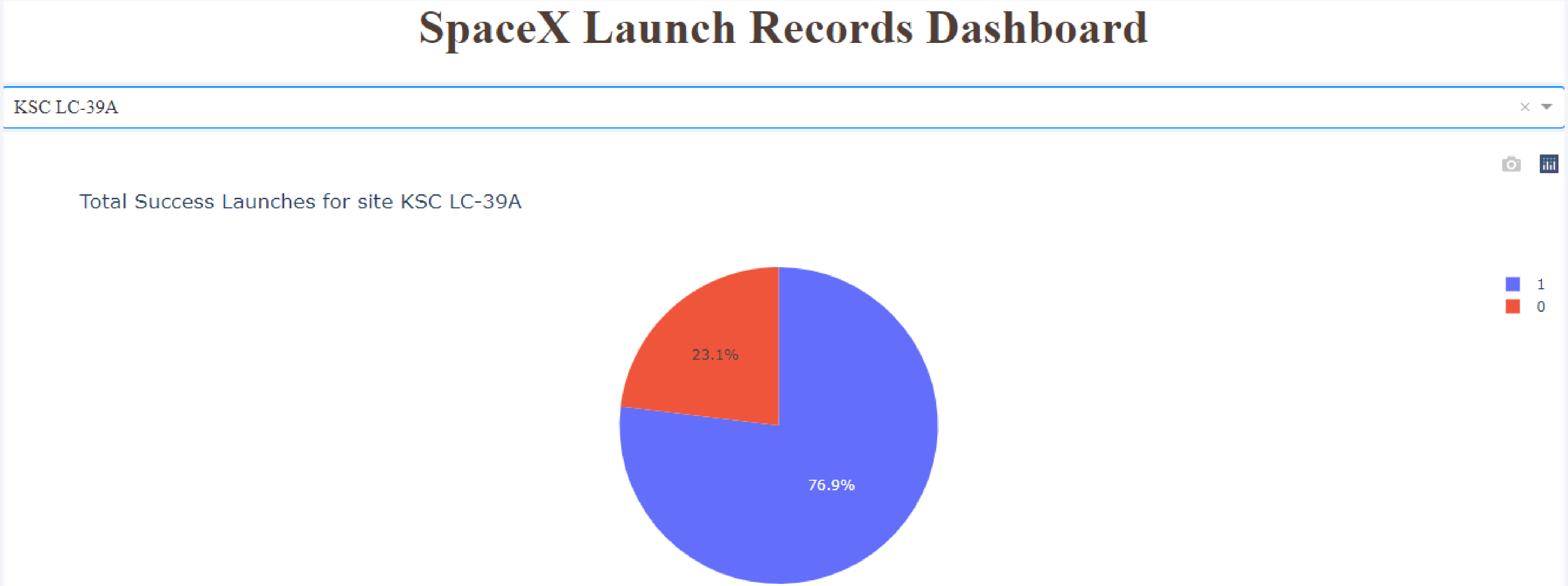
Build a Dashboard with Plotly Dash

Total Success Counts for All Launch Sites



KSC LC-39A has the highest amount of successful launches among all sites

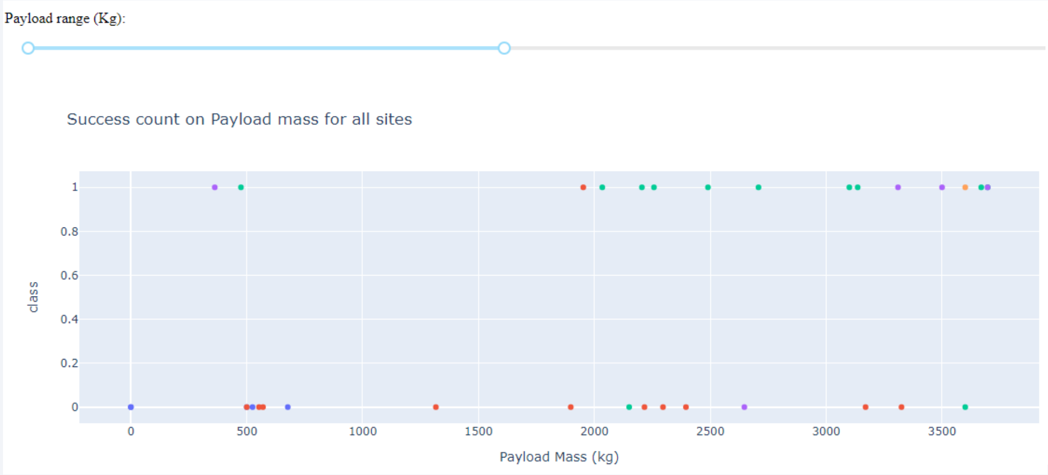
Launch Site with Highest Launch Success Ratio



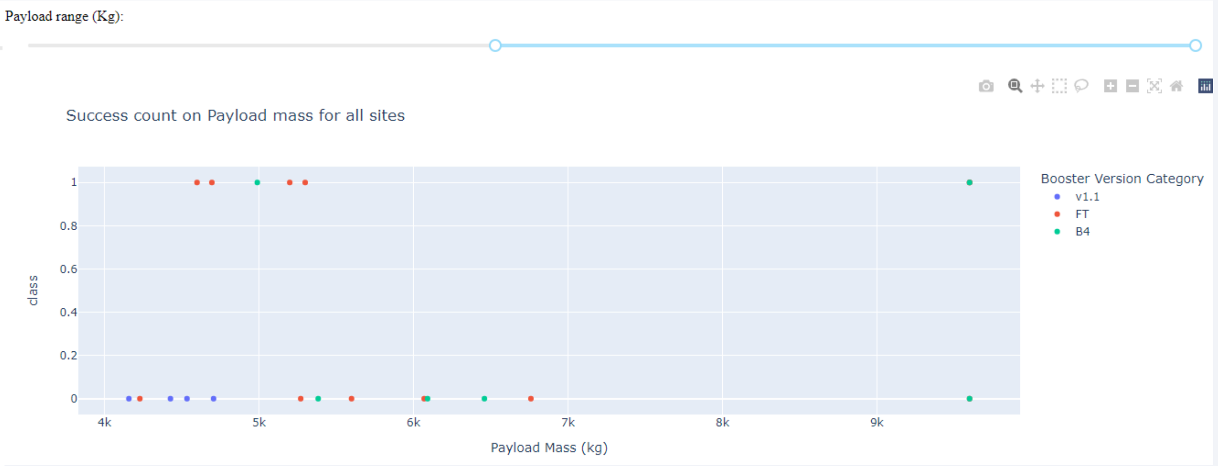
- KSC LC-39A achieved a 76.9% success rate while getting a 23.1% failure rate

Payload vs Launch Outcome (All Launch Sites)

Low weighted Payload range 0 – 4000kg



Heavy weighted Payload range 4000 - 6000kg

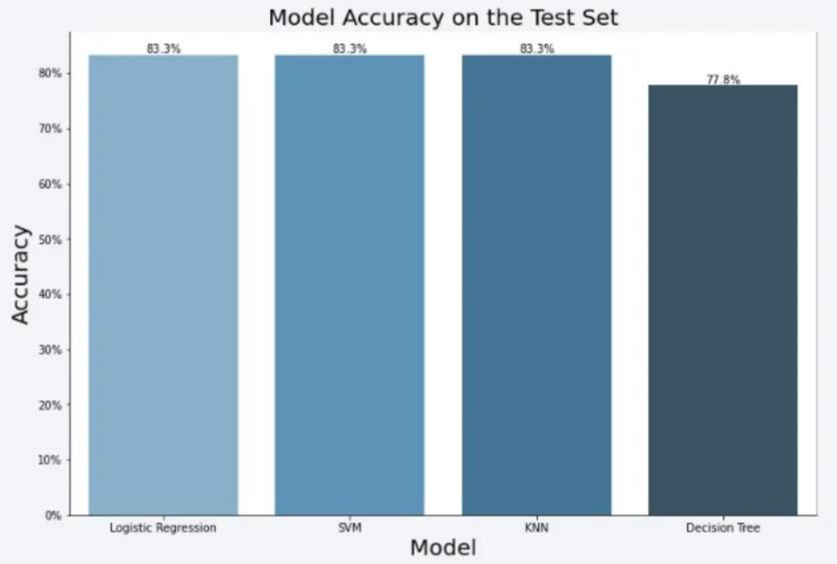


- Success rate for low weighted payload range is higher than their heavy counterparts

Section 5

Predictive Analysis (Classification)

Classification Accuracy

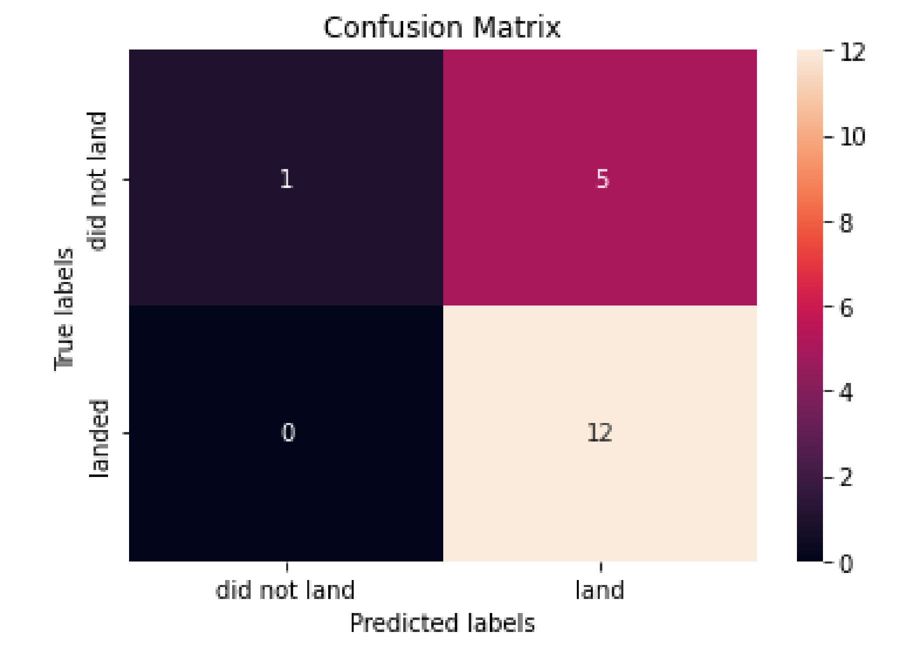



```
print('Accuracy for Logistics Regression method:', logreg_cv.score(X_test, Y_test))
print( 'Accuracy for Support Vector Machine method:', svm_cv.score(X_test, Y_test))
print('Accuracy for Decision tree method:', tree_cv.score(X_test, Y_test))
print('Accuracy for K nearsdt neighbors method:', knn_cv.score(X_test,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

```
Accuracy for Logistics Regression method: 0.8333333333333334
Accuracy for Support Vector Machine method: 0.8333333333333334
Accuracy for Decision tree method: 0.7777777777777778
Accuracy for K nearsdt neighbors method: 0.8333333333333334
```

Confusion Matrix

The confusion matrix for the decision tree classifier shows that it can distinguish between the different classes, however the major problem is the false positives i.e; unsuccessful landing marked as successful landing by the classifier



Conclusions

- The SVM, KNN and Logistic Regression Models are the best in terms of the prediction accuracy for this dataset.
- Low weighted payloads perform better than heavier payloads.
- The success rates for SpaceX launches is directly proportional to time. In years they will eventually perfect their launches.
- KSC LC-39A had the most successful launches among all sites.
- Orbits GEO, HEO, SSO, ES L1 have the best success rates.

Thank you!