IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Amulya Chilukuri
15-December-2022

Outline

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

Executive Summary

- Summary of methodologies
  - ➢ Data Collection using API
  - ➢ Data Collection by Web Scrapping
  - ➢ Data Wrangling
  - ➢ Exploratory Data Analysis using SQL
  - ➢ Exploratory Data Analysis with Data Visualization
  - ➢ Interactive Data Visualization and Analytics using Folium
  - ➢ Interactive Data Visualization and Analytics with Plotly Dash
  - ➢ Machine Learning and Predictive Analysis
- Summary of all results
  - ➢ Exploratory Data Analysis results
  - ➢ Interactive Analytics – with Screenshots
  - ➢ Predictive Analytics 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 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. This can be achieved by using machine learning to predict the successful landing of the rocket in the first stage, which is the basis of this project.

- Problems you want to find answers

  - ➢ What are the conditions of operation that need to be in place to ensure the successful landing of the rocket in the first stage.
  - ➢ Find out how different features involved in the launch, influence and interact with each other, in the launching of the rocket
  - ➢ Predict the successful landing of the rocket in the first stage using all factors influencing the landing.

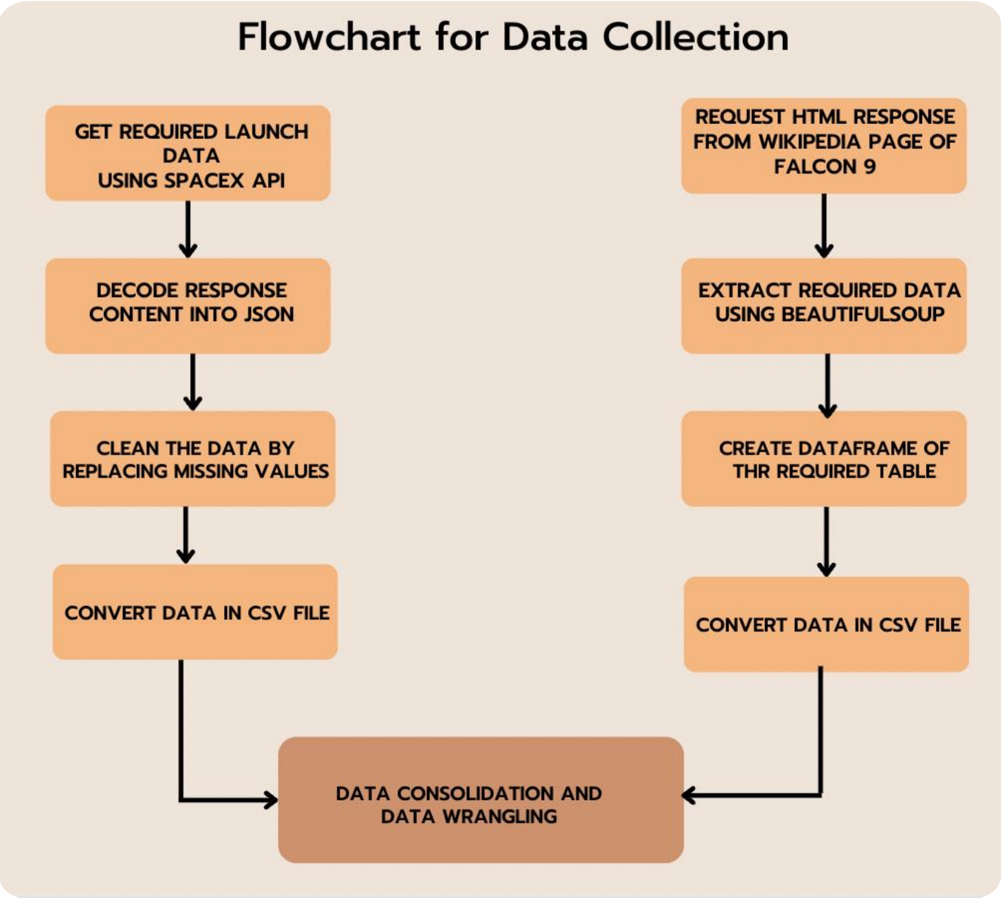
Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was collected using SpaceX API and Web Scrapping the Wikipedia page for Falcon 9 Rocket and Falcon heavy launches records.
- Perform data wrangling
  - Data was analysed to find patterns that helped determine the training labels for successful and unsuccessful booster landings.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Used Linear Regression, Support Vector Machine, Decision Tree Classifier, K-Nearest Neighbors classification models to perform the predictive analysis. The best classification model was evaluated.



Data Collection

- SPACEX API was used to collect data such as rockets used, information on launching and landing, payload mass and launch outcomes.
- Data was also collected by means of web scrapping, from Wikipedia page consisting of Falcon 9's Launch information, using BeautifulSoup .

Data Collection – SpaceX API

1.GET REQUEST for Rocket Launch Data using API.

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

2. Converting response to a .JSON file and then to a pandas data frame

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

```
]: # Use json_normalize meethod to convert the json result into a data
data = pd.json_normalize(response.json())
print(data.head())
```

3. Get other data in a similar way and assign to a dictionary and convert to 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}
```

```
]: # Create a data from launch_dict
data_falcon9 = pd.DataFrame(launch_dict)
```

4. Clean the data by replacing missing values with mean.

```
: # Calculate the mean value of PayloadMass column
# Replace the np.nan values with its mean value
data_falcon9['PayloadMass'] = data_falcon9['PayloadMass'].fillna(data_falcon9['PayloadMass'].mean())
data_falcon9.isnull().sum()
```

5. convert the cleaned data to csv.

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

- SPACEX API was used to request and extract the required data, clean the requested data. Some basic Data Wrangling and formatting was done on this data.

The link to the notebook is:

https://github.com/amulyachilukuri/Capstone-project/blob/main/Week1_datacollection.ipynb

Data Collection - Scraping

- Webscrapping of Wikipedia page done using Beautifulsoup.

- The github url for this is:

https://github.com/amulyachilukuri/Capstone-project/blob/main/Week1_WebScrapping.ipynb

1. Get response request from the HTML of Wikipedia page:

```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Fal
```

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

2. Create the BeautifulSoup Object:

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

3. Find the required tables:

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

4. Extract the columns names:

```
thelement = soup.find_all('th')
for x in range(len(thelement)):
    try:
     name = extract_column_from_header(thelement[x])
     if (name is not None and len(name) > 0):
        column_names.append(name)
    except:
     pass
```

5. Create dictionary and assign it to data frame, then convert to csv.

```
launch_dict= dict.fromkeys(column_names)

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

# Let's initial the launch_dict with eac
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']=[]
```

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

Data Wrangling

- Exploratory Data analysis was performed to find patterns and assign training labels.
- GitHub URL for data wrangling related notebook:

https://github.com/amulyachilukuri/Capstone-project/blob/main/Week1_DataWrangling.ipynb

1. Data collected is loaded into a dataframe:

```
df=pd.read_csv(dataset_part_1_csv)
df.head(10)
```

2. No. Of launches for each site was calculated:

```
df['LaunchSite'].value_counts()
```

```
CCAFS SLC 40    55
KSC LC 39A      22
VAFB SLC 4E     13
Name: LaunchSite, dtype: int64
```

3. No. Of occurences of each orbit was calculated: `df['Orbit'].value_counts()`
4. Landing outcomes is calculated: `landing_outcomes = df['Outcome'].value_count`
5. We create training labels for the outcomes:

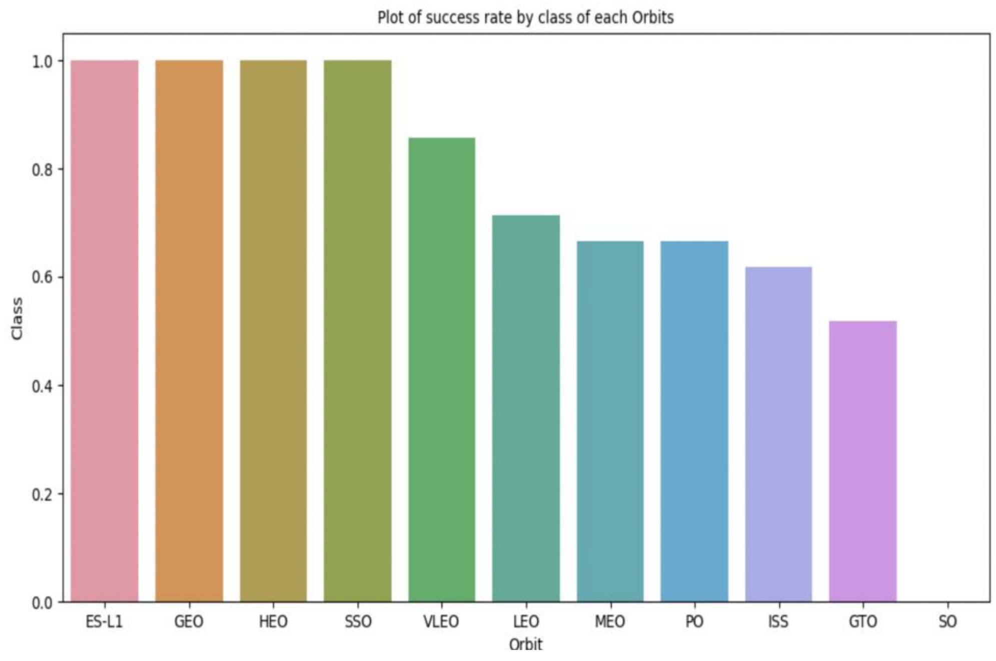
```
# landing_class = 0 if bad_outcome
# landing_class = 1 otherwise
landing_class = df['Outcome'].replace({'False Ocean': 0, 'False ASDS': 0, 'None None': 0, 'None
df['Outcome'] = df['Outcome'].astype(int)
df.info()
```

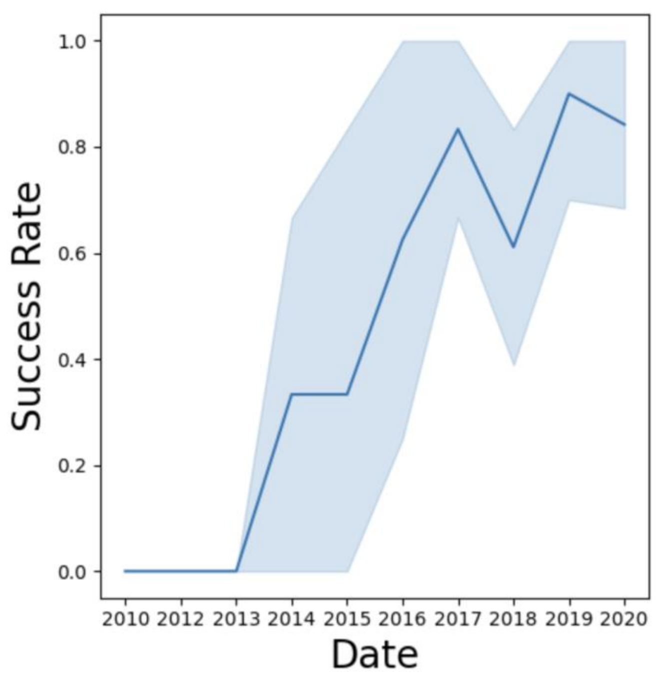
```
df['Class']=df['Outcome']
df[['Class']].head(8)
```

	Class
0	0
1	0
2	0
3	0
4	0
5	0
6	1
7	1

EDA with Data Visualization

- Data was explored by visualizing the relationship between Flight Number and Launch Site, Payload and Launch Site, success rate of each orbit type, Flight Number and orbit type, Payload and Orbit type and launch success yearly trend.
- GitHub URL of EDA with data visualization notebook: https://github.com/amulyachilukuri/Capstone-project/blob/main/Week2_EDAVisualization.ipynb





EDA with SQL

- The SPACEX dataset was loaded into IBM's database console and accessed from Jupyter notebook.
- The below is the summary of SQL queries that were performed to gain insights into the data:
  - The names of the unique launch sites in the space mission were queried and got.
  - The total payload mass carried by boosters launched by NASA (CRS) were displayed.
  - The average payload mass carried by booster version F9 v1.1 was displayed.
  - The date when the first successful landing outcome in ground pad was achieved is listed.
  - Names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000 were displayed.
  - The total number of successful and failure mission outcomes.
  - The month names, failure landing outcomes in drone ship ,booster versions, launch site for the months in year 2015.
- GitHub URL of EDA with SQL notebook: https://github.com/amulyachilukuri/Capstone-project/blob/main/Week2_EDAWith_SQL.ipynb

Build an Interactive Map with Folium

- The launch sites of the SPACEX rockets were marked on an interactive map leaflet, which is a special feature of Folium to easily visualize data on maps. Map objects such as markers, circles and lines were added to mark the success or failure of each launch at these sites.

- Launch outcomes (Failure or success) were marked with class 0 for failure and class 1 for success. Colored labelled marker clusters were used to identify which launch sites had relatively high success rates.

- Distance between the launch sites and proximities to the coastline, nearest railway line, nearest highway and nearest city were calculated and marked on the map. Line were used to connect to these proximities for better visualize the distances.

- Based on this, it was found that:
  1. Launch sites were nearer to the coastlines, to ensure safety to people and property. The trajectory of the rocket remains over the ocean, rather than on land.
  2. Launchsites are in close proximity to the railways, to enable faster transport of raw material like fuel, rocket/spaceship launch material.
  3. Launchsites are in close proximity to the highways, to enable quicker transport of people.
  4. Launchsites indeed keep a long distance away from cities - so that civilians and civilian property are not harmed due to failed missions causing failling debris.

- GitHub URL of interactive map with Folium map: https://github.com/amulyachilukuri/Capstone-project/blob/main/Week3_jupyter_launch_site_location.jupyterlite.ipynb

Build a Dashboard with Plotly Dash

- An interactive dashboard was created using Plotly Dash.
- Pie charts were plotted showing total launches for each site and also for all sites together.
- A scatter plot was plotted between Pay Load Mass and the Outcome for different booster versions.
- The dashboard was made interactive with the following:

  > A drop down selection for the launch sites.

  > A slider to select the Pay Load range.
- The GitHub URL of the completed Plotly Dash lab is: https://github.com/amulyachilukuri/Capstone-project/blob/main/spacex_dash_app.py

Predictive Analysis (Classification)

- Data was loaded using numpy and pandas and transformed the data.
- The data was then split into training and testing set.
- Using GridSearchCV, machine learning models were built and tuned with different hyperparameters.
- The accuracy of each model was calculated for Support Vector Machine, Logical Regression, Decision Trees and K-Nearest Neighbor models, using the test data.
- The GitHub URL for predictive analysis: https://github.com/amulyachilukuri/Capstone-project/blob/main/Week5_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

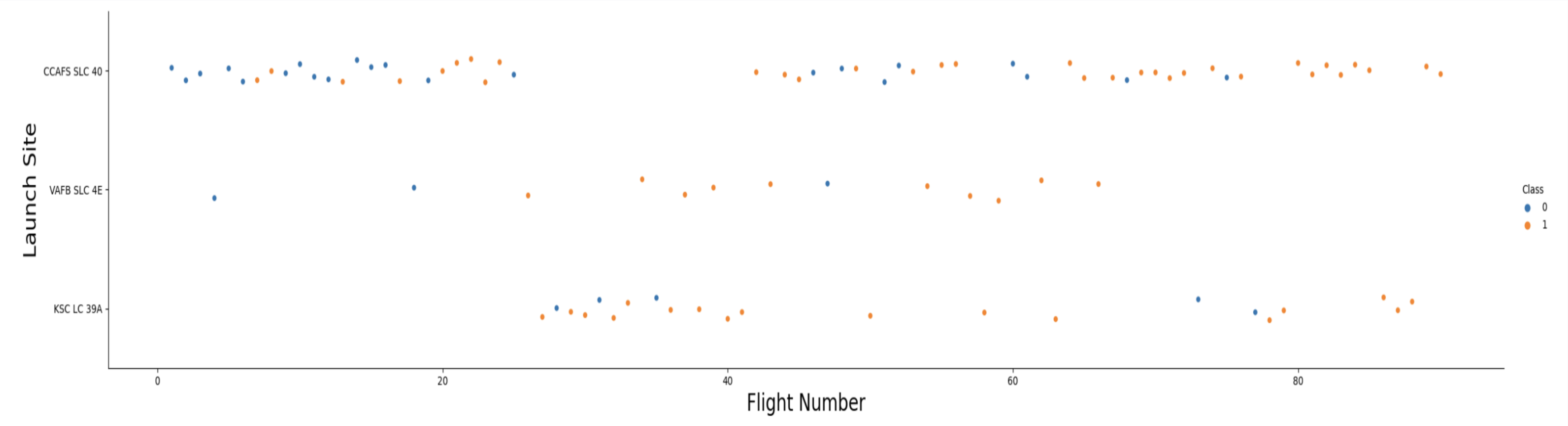
Results

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

Section 2

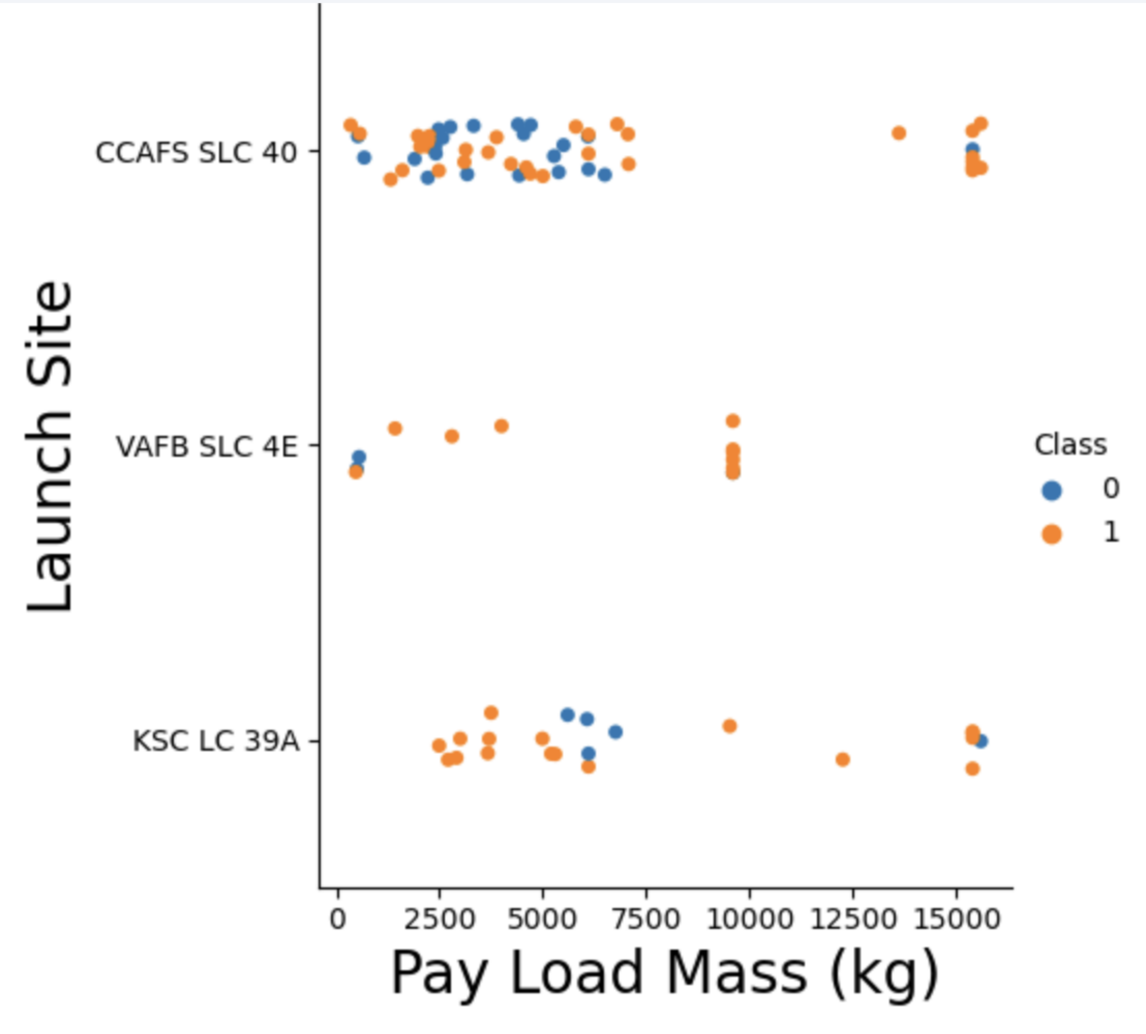
Insights drawn from EDA

Flight Number vs. Launch Site



From the above scatter plot, it can be inferred that as the flight number increases, the greater the success rate of launch at the various launch sites.

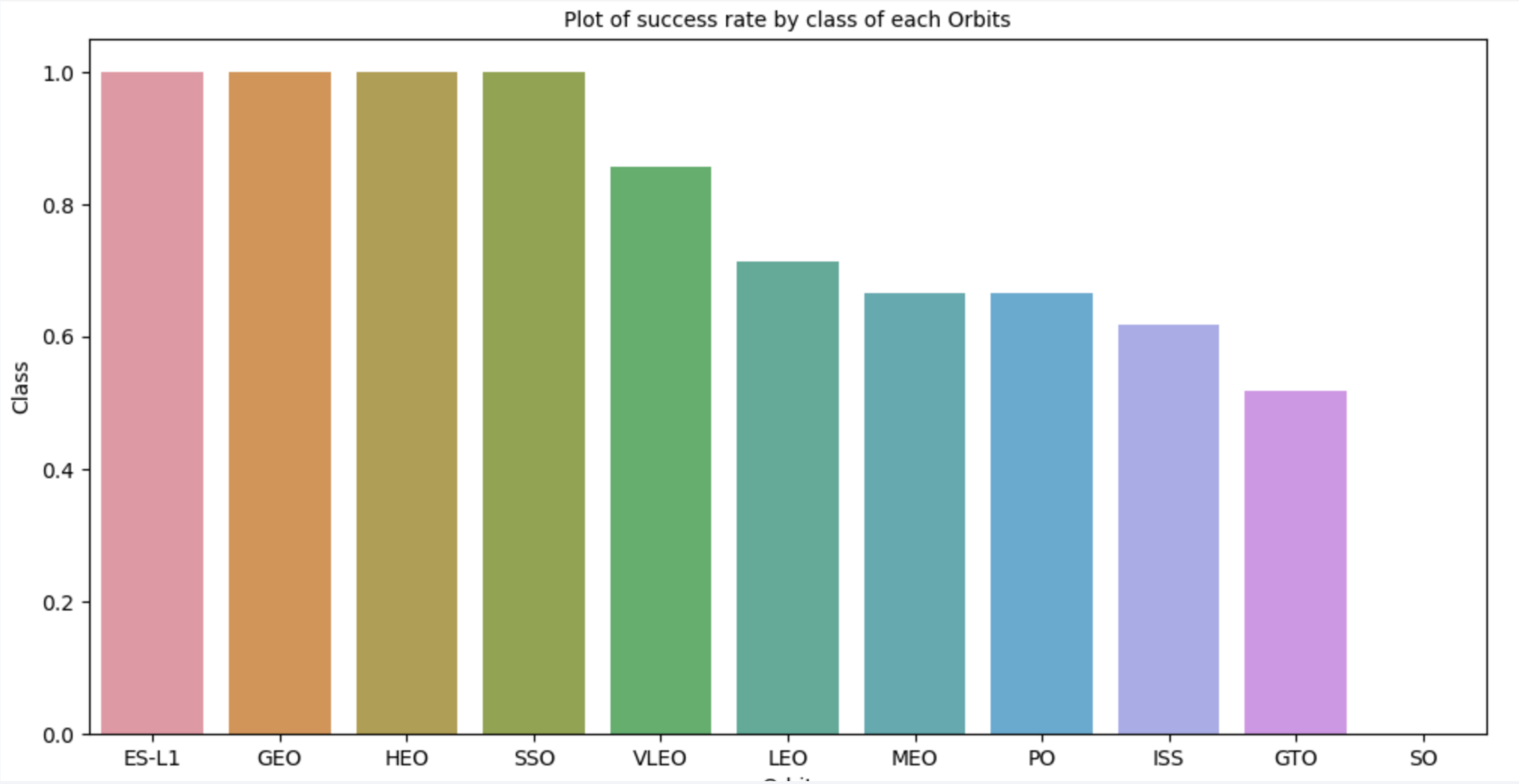
Payload vs. Launch Site



From this scatter plot, it can be inferred that as the Pay Load Mass increases, the success rate of launch decreases, at the various launch sites.

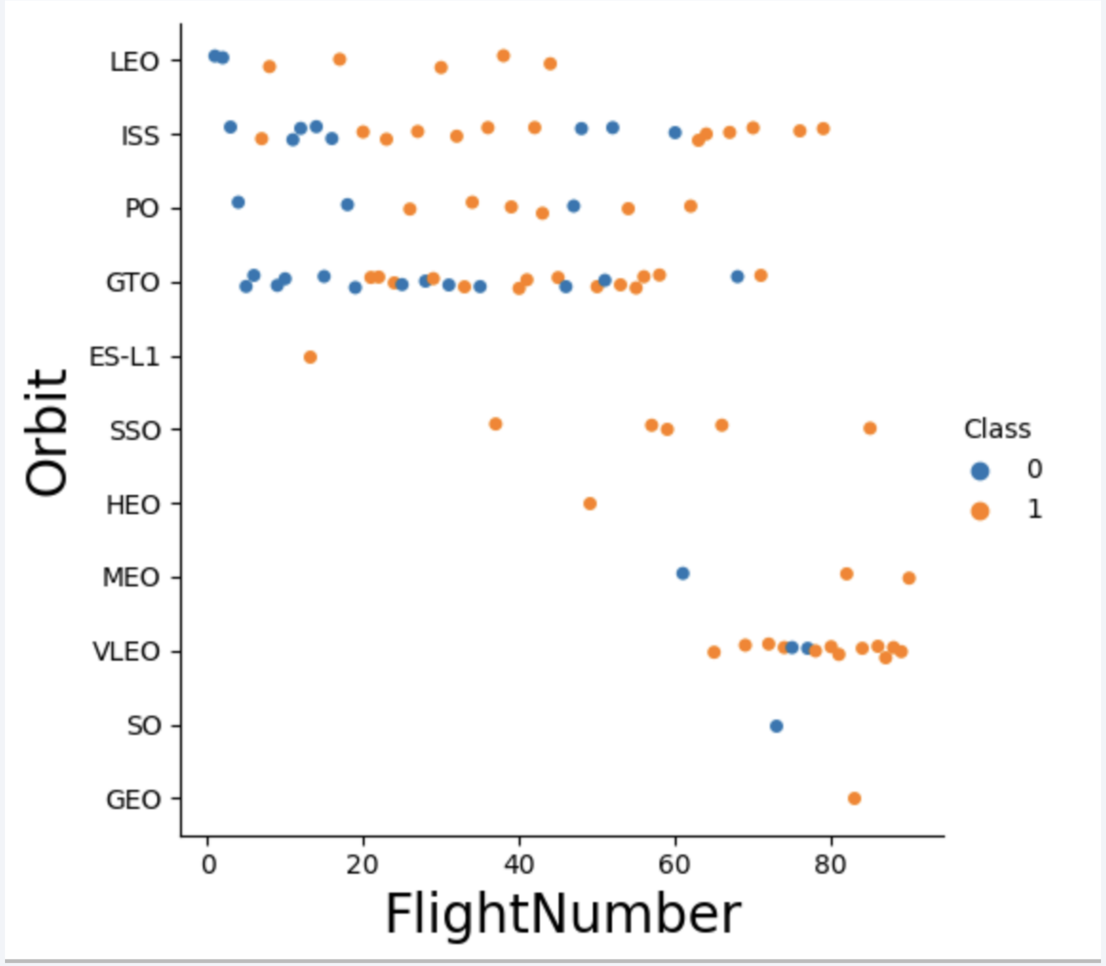
It is interesting to note that the there are no rockets Pay load mass > 10000, at the launch site VAFB-SLC.

Success Rate vs. Orbit Type



This plot tells us that ES-L1, GEO, HEO and SSO had the highest success rates of launch.

Flight Number vs. Orbit Type

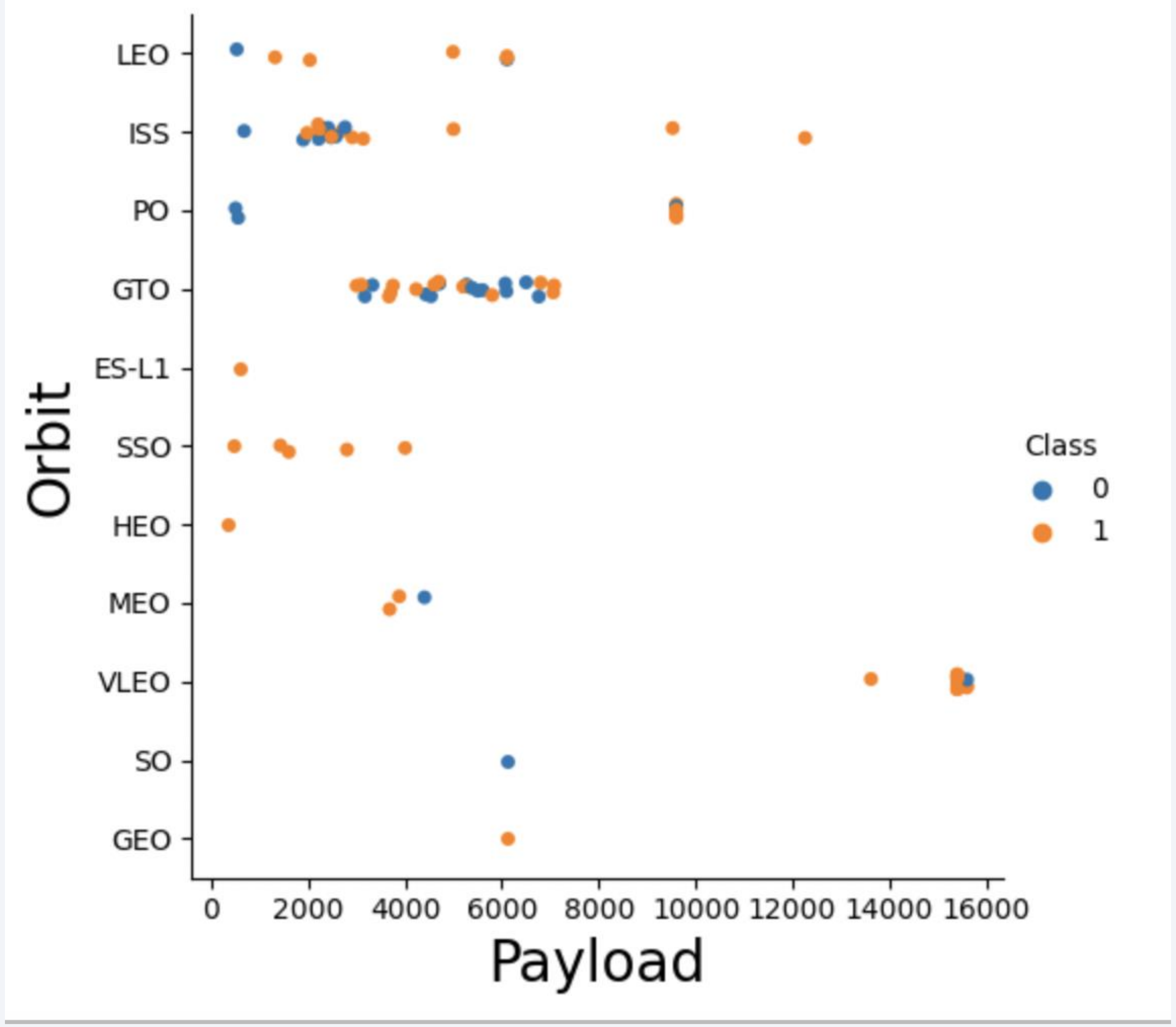


This is a scatter plot of Flight Number vs Orbit.

We can notice that in the LEO orbit, success rate is increasing with increase in flight number.

Whereas, in the case of GTO orbit, there is no relation between the Orbit and the Flight Number.

Payload vs. Orbit Type



This figure shows the scatterplot between Pay Load and Orbit type.

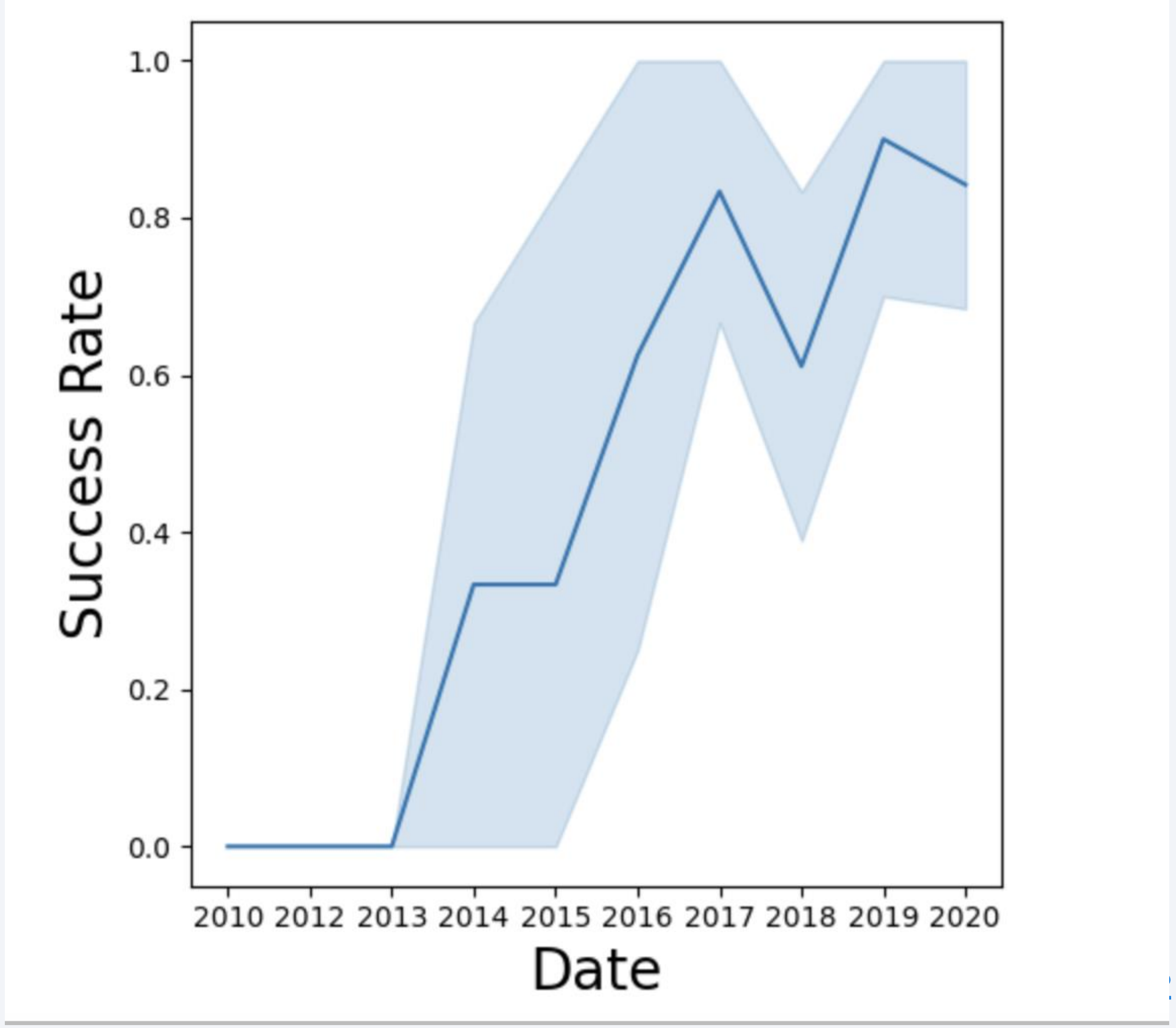
It can be inferred that for Polar, LEO and ISS orbits have higher success in landing as the Pay Load increases.

However, in the case of GTO there is no relationship between Orbit type and Payload.

Launch Success Yearly Trend

The plot shows the yearly trend of success rate of launch.

Success rate is observed to increase from Year 2013 and kept increasing until 2020.



All Launch Site Names

Using DISTINCT keyword, the list of unique launch sites is derived from the SPACEX dataset.

Totally there are 4 unique launch sites as seen in the screen shot.

```
%sql SELECT DISTINCT(LAUNCH_SITE) FROM SPACEXTBL
```

```
 * ibm_db_sa://bjl90930:***@b70af05b-76e4-4bca-a1f5-
Done.
```

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

Launch Site Names Begin with 'CCA'

```
%sql SELECT * from SPACEXTBL WHERE LAUNCH_SITE like 'CCA%' limit 5
```

```
 * ibm_db_sa://bjl90930:***@b70af05b-76e4-4bca-a1f5-23dbb4c6a74e.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32716/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	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 Launch_Site column can be filtered for names starting with 'CCA' as seen in the query in the above snapshot.

Total Payload Mass

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) as TOTAL_PAYLOAD_MASS FROM SPACEXTBL WHERE Customer = 'NASA (CRS)'
```

```
 * ibm_db_sa://bjl90930:***@b70af05b-76e4-4bca-a1f5-23dbb4c6a74e.c1ogj3sd0tgtu0lqde00.databases.appdoma
Done.
```

total_payload_mass
45596

The total payload carried by boosters from NASA, can be calculated using the SUM function and is found to be 45596 as seen in the result in the above snapshot.

Average Payload Mass by F9 v1.1

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) as AVERAGE_PAYLOAD_MASS FROM SPACEXTBL WHERE BOOSTER_VERSION = 'F9 v1.1'
```

```
 * ibm_db_sa://bjl90930:***@b70af05b-76e4-4bca-a1f5-23dbb4c6a74e.c1ogj3sd0tgtu0lqde00.databases.appdomain.c
Done.
```

average_payload_mass
2928

The average payload mass carried by booster version F9 v1.1 is calculated using AVG function and the result is 2928.

First Successful Ground Landing Date

```
%sql select MIN(Date) from SPACEXTBL WHERE Landing__Outcome = 'Success (ground pad)'

 * ibm_db_sa://bjl90930:***@b70af05b-76e4-4bca-a1f5-23dbb4c6a74e.c1ogj3sd0tgtu0lqde00
Done.
```

1
2015-12-22

The dates of the first successful landing outcome on ground pad, is 22nd December 2015.

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%sql SELECT BOOSTER_VERSION FROM SPACEXTBL WHERE landing__outcome = 'Success (drone ship)' and  payload_mass__kg_ > 4000 and payload_mass__kg_<6000
```

```
 * ibm_db_sa://bjl90930:***@b70af05b-76e4-4bca-a1f5-23dbb4c6a74e.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32716/BLUDB
Done.
```

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

- The WHERE clause was used to filter the names of boosters which have successfully landed on drone ship and the AND condition was applied to determine the successful landing with payload mass greater than 4000 but less than 6000.

Total Number of Successful and Failure Mission Outcomes

List the total number of successful and failure mission outcomes

```
]: %sql select count(MISSION_OUTCOME) as Succesful_missions from SPACEXTBL where MISSION_OUTCOME = 'Success'
```

```
 * ibm_db_sa://bjl90930:***@b70af05b-76e4-4bca-a1f5-23dbb4c6a74e.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32716/BLUDB
Done.
```

]:

succesful_missions
99

```
]: %sql select count(MISSION_OUTCOME) as Failed_missions from SPACEXTBL where MISSION_OUTCOME = 'Failure (in flight)'
```

```
 * ibm_db_sa://bjl90930:***@b70af05b-76e4-4bca-a1f5-23dbb4c6a74e.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32716/BLUDB
Done.
```

]:

failed_missions
1

The WHERE clause is used to determine the number of Successful and Failed mission outcomes.

Boosters Carried Maximum Payload

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

 * ibm_db_sa://bjl90930:***@b70af05b-76e4-4bca-a1f5-23dbb4c6a74e.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:3271
Done.
```

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

List of the names of the boosters which have carried the maximum payload mass is determined using the WHERE Clause and MAX() function in the sub-Query.

2015 Launch Records

```
]: %sql SELECT substr(Date, 6, 2) AS month,Landing__Outcome, booster_version,launch_site from SPACEXTBL where Landing__Outcome = 'Failure (drone ship)' \
     and substr(Date,1,4)='2015'
```

```
 * ibm_db_sa://bjl90930:***@b70af05b-76e4-4bca-a1f5-23dbb4c6a74e.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32716/BLUDB
Done.
```

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

The substr function is used to determine the month, landing outcome, booster version and launch site for failed landing outcomes in drone ship, for the year 2015.

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

```
%sql select Landing__Outcome, count(Landing__Outcome) from SPACEXTBL where (DATE between '2010-06-04' and '2017-03-20') \
    GROUP BY Landing__Outcome order by COUNT(Landing__Outcome) desc
```

```
 * ibm_db_sa://bjl90930:***@b70af05b-76e4-4bca-a1f5-23dbb4c6a74e.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32716/BLUDB
Done.
```

landing__outcome	2
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

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 is displayed using the WHERE clause that includes the dates and GROUP BY clause to group the landing outcome and ORDER BY clause to order the landing outcomes in descending order.

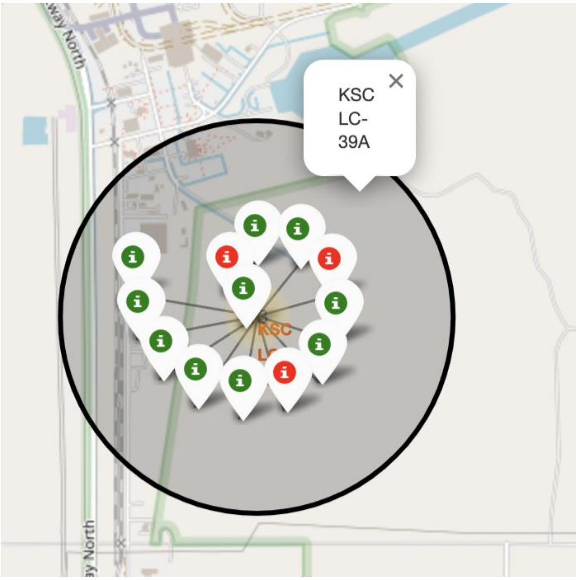
Section 3

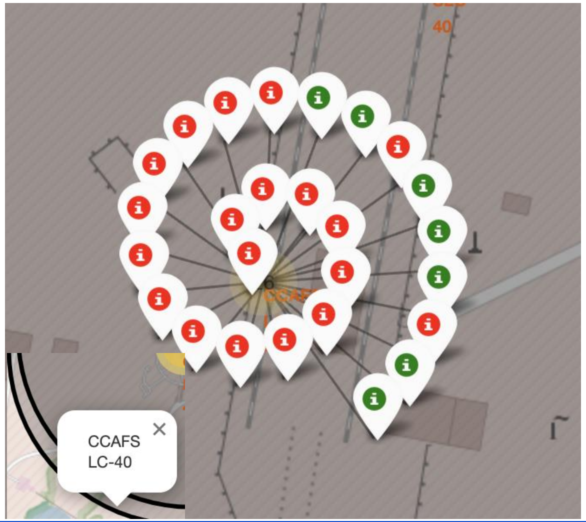
Launch Sites Proximities Analysis

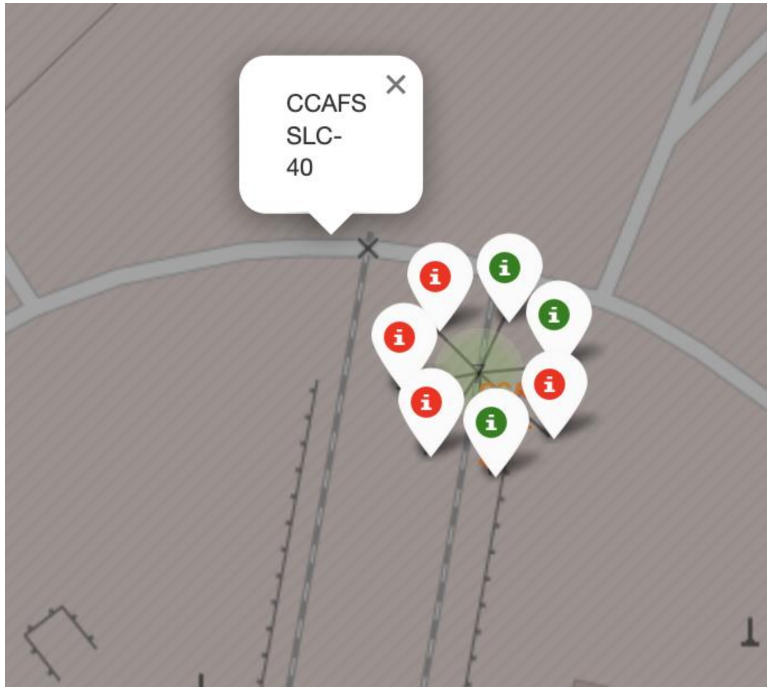
All Launch Sites on the Global map with markers

It can be observed that all the SPACEX launch sites are located in the United States of America at California and Florida.

Markers showing launch sites with colored labels

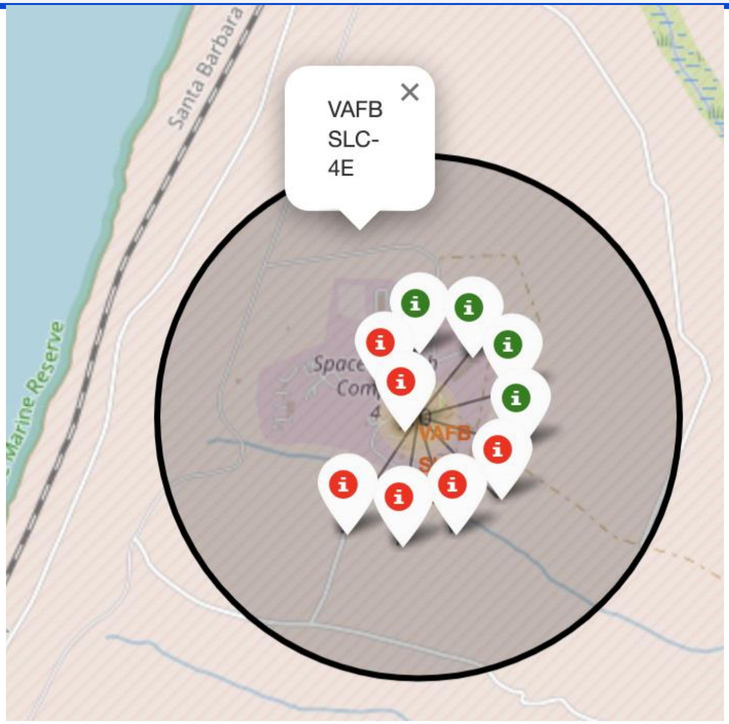






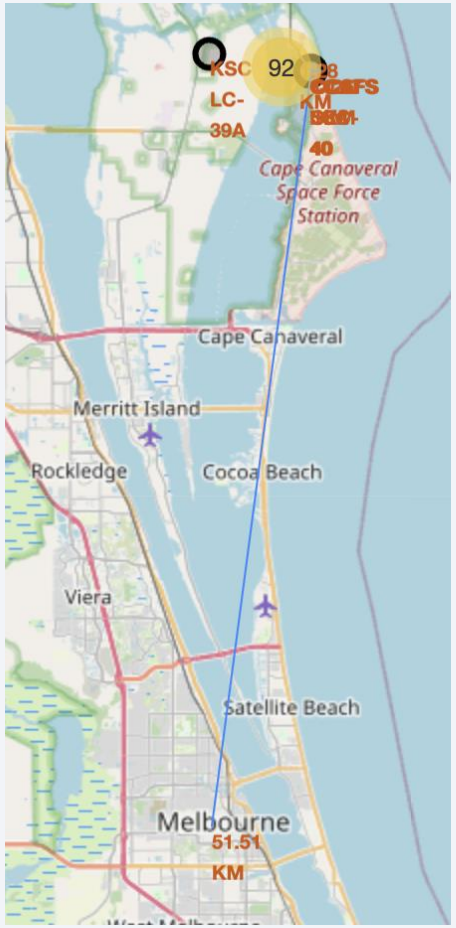
Florida Launch Sites

Green markers show successful launches. Red markers show failed launches



California Launch Sites

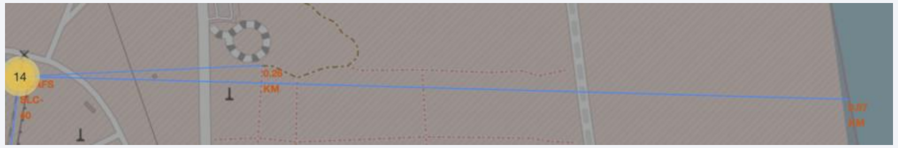
Distance from launch sites to various landmarks



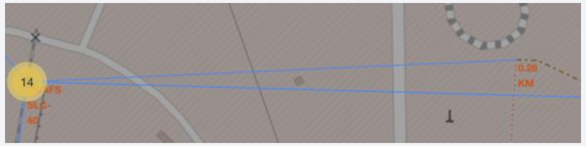
Distance to nearest city

Distance to nearest railway line.

Distance to nearest coastline.



Distance to nearest highway.

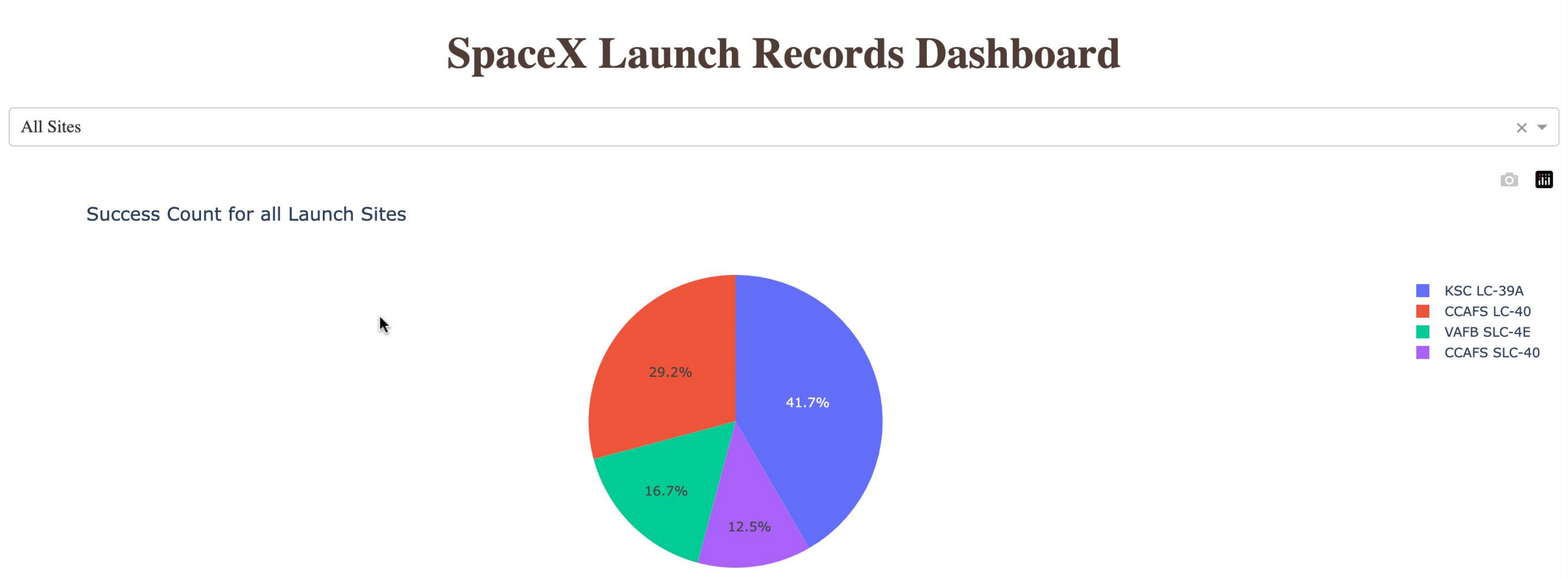


- Are launch sites in close proximity to railways? - No
- Are launch sites in close proximity to highways?-No
- Are launch sites in close proximity to coastline?-Yes
- Do launch sites keep certain distance away from cities?-Yes

Section 4

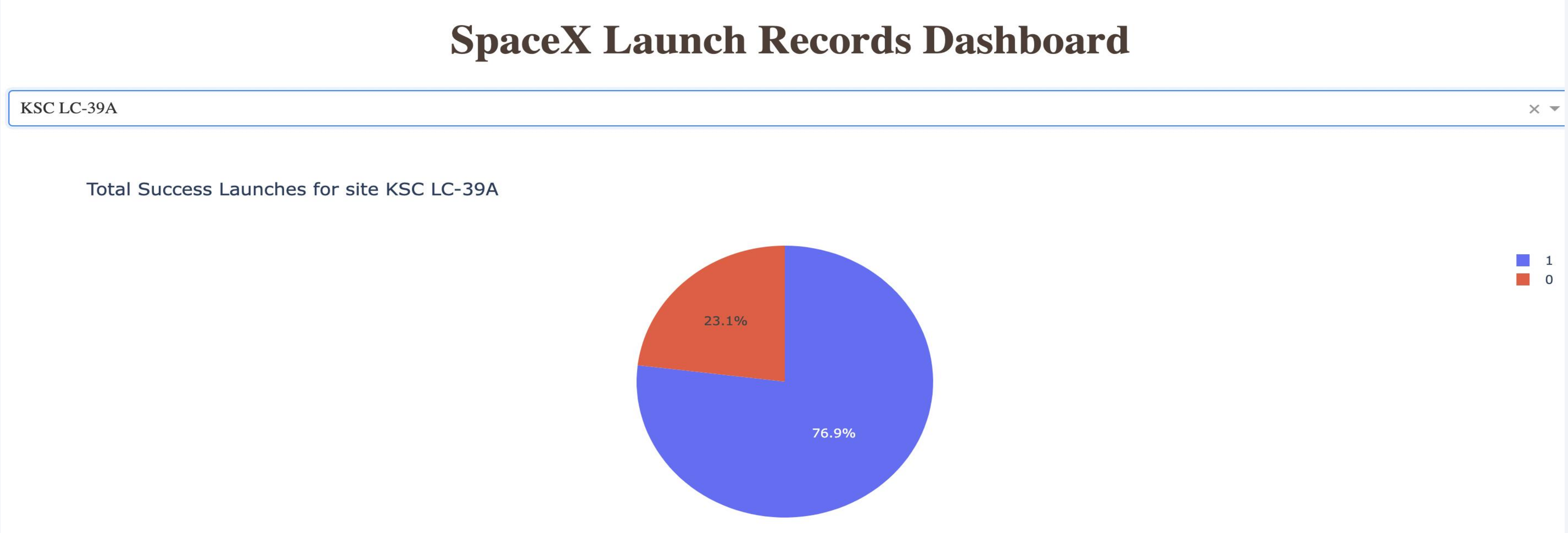
Build a Dashboard with Plotly Dash

Pie chart showing success rate in % for all Launch Sites.



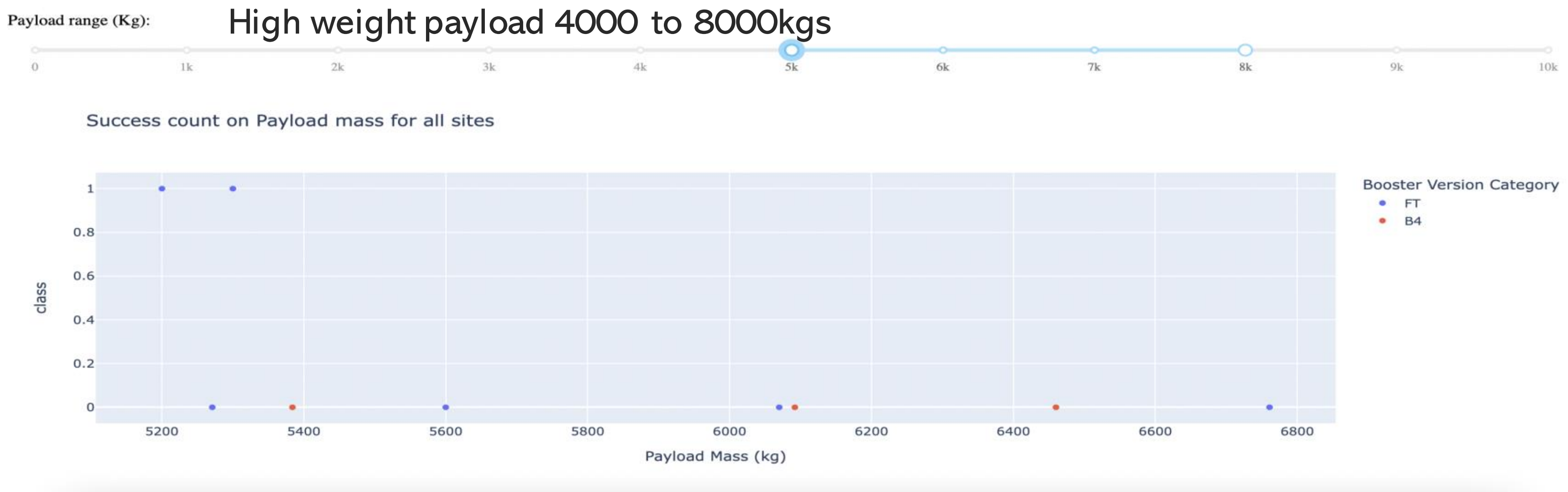
KSC LC-39A had the most successful rocket launch rate amongst all the sites.

Pie chart showing the Launch Site with the highest launch success ratio



KSL LC-39A is seen to have a 76.9% success rate and a failure rate of 23.1%.

Pay load vs Launch outcome scatterplot, for different Payload selected in slider range.



Success rate of launches is higher for payloads that are lower in weight, than when compared to payloads that are higher in weight.

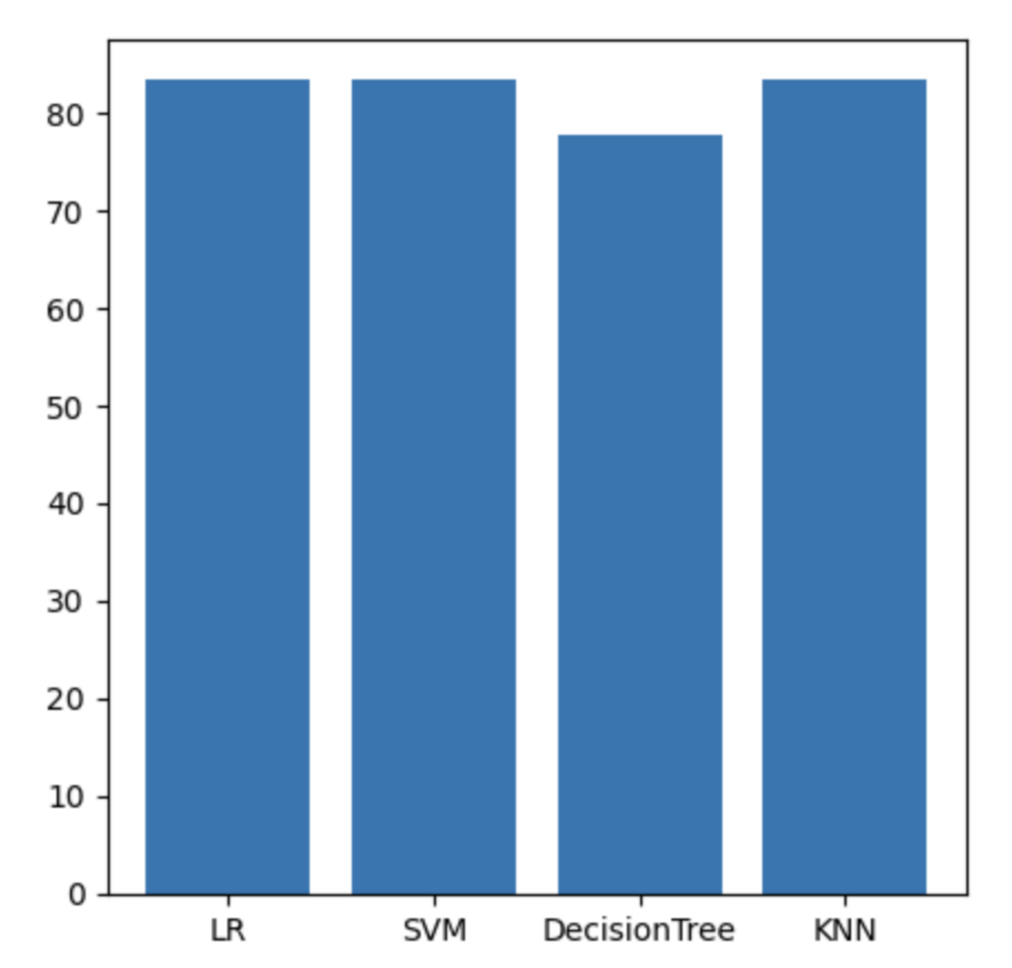
Section 5

Predictive Analysis (Classification)

Classification Accuracy

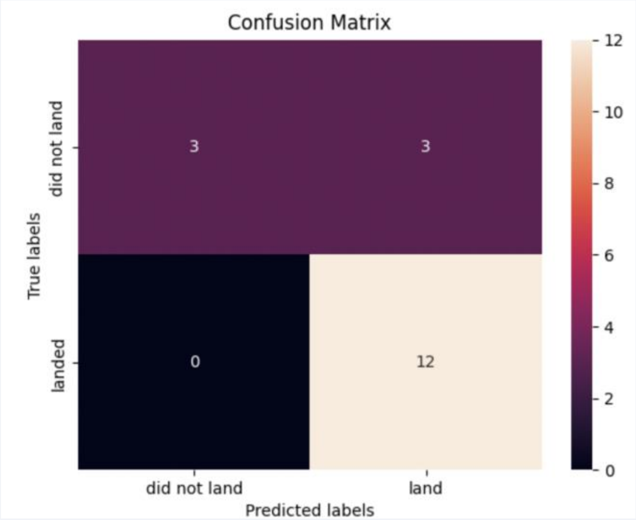
The bar chart shows the classification accuracy of Logical Regression, Support Vector Machine, Decision Tree and K-nearest neighbors.

The Logical Regression, Support Vector Machine and K-nearest neighbors have the highest prediction accuracy.

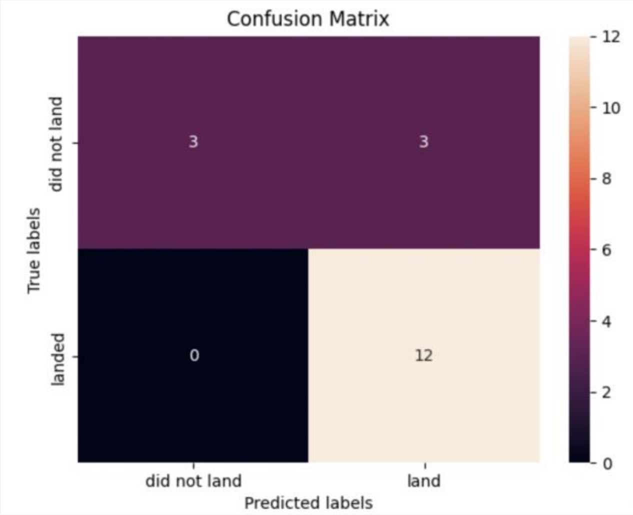


Confusion Matrix

Logical Regression



Support Vector Machine



The confusion matrix for Logical regression model and Support Vector Machine models is shown. Both models are able to distinguish between the different classes, but a major problem is false positives.

Conclusions

- Logical Regression, Support Vector Machine and K-nearest neighbors are the best models for prediction accuracy for this data set.
- The success rate of launches has increased steadily from 2013 onwards, until 2020.
- Pay loads that were lower in weight performed better, than pay loads that were higher in weight.
- KSC LC-39A is the most successful launch site amongst all other sites.
- Orbits SSO, GEO, HEO and ES-L1 had the most success rate of launching.

Thank you!