IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Akshay M

2 Feb 2023

Outline

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

Executive Summary

- Summary of methodologies
  - Data collection
  - Data wrangling
  - Exploratory Data Analysis with Data Visualization
  - Exploratory Data Analysis with SQL
  - Building an interactive map with Folium
  - Building a Dashboard with Plotly Dash
  - Predictive analysis (Classification)
- Summary of all results
  - Exploratory Data Analysis results
  - Interactive analytics demo
  - Predictive analysis results

Introduction

Project background and context

SpaceX company 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 based on whether SpaceX can reuse the first stage. Hence the importance of the reuse of the first stage is paramount. This is a major factor in determining the cost of the launch.

Questions to be answered

- What are the factors that determine the success of a launch?
- How do factors (payload mass, number of flights, launch site and orbits affect the success?
- Which algorithm will provide the best accuracy of prediction?

Section 1

Methodology

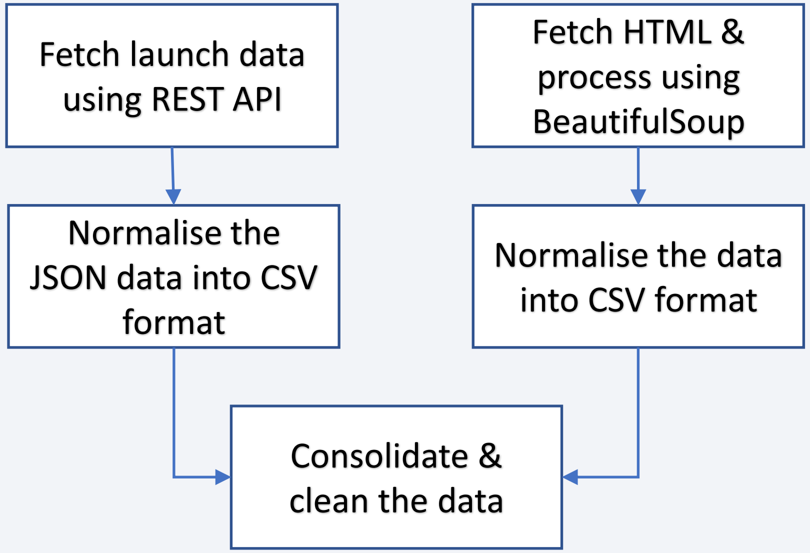
Methodology

Executive Summary

- Data collection methodology:
  - Using Spacex REST API
  - Web scrapping from Wikipedia
- Perform data wrangling
  - 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
  - DT, SVM, KNN and LR models have been built and evaluated to find out the best performing classifier

Data Collection

- Data Sources
  - SpaceX REST API endpoint was used to fetch the launch data
  - Web scraping using Beautiful Soup
- Data Wrangling
  - The data was consolidated and cleaned



Data Collection – SpaceX API

1. Fetch response, convert to JSON
2. Normalize to Data Frame
3. Call functions to extract info
4. Make dictionary out of list
5. Convert dictionary to Data Frame
6. Filter out records
7. Create CSV file

URL : https://github.com/akshayravim/courserarepo/blob/1f8bb603244639d801ce83119aa43ff3f3598274/jupyter-labs-spacex-data-collection-api.ipynb

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

```
# Use json_normalize meethod to conver
data = pd.json_normalize(response)
```

```
# Call getPayloadData
getPayloadData(data)
```

```
# Call getBoosterVersion
getBoosterVersion(data)
```

```
# Call getLaunchSite
getLaunchSite(data)
```

```
# Call getCoreData
getCoreData(data)
```

```
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, orient='index').transpose()
```

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

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

Data Collection - Scraping

1. Fetch the HTML response
2. Parse contents using BeautifulSoup
3. Find all tables
4. a. Iterate & find names

   b. Create dictionary

5. Populate column names
6. Create Data frame from dictionary
7. Write CSV file of the Data Frame data

URL:
https://github.com/akshayravim/courserarepo/blob/d330bf395085e1824f1edbc527c0b6ddfb7c3b5e/jupyter-labs-webscraping.ipynb

```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_
response = requests.get(static_url)
```

```
# Use BeautifulSoup() to create a BeautifulSoup object f
soup = BeautifulSoup(response.content, 'html.parser')
```

```
# Assign the result to a list called `html_tables`
html_tables=soup.find_all('table')
```

```
column_names = []
flt = first_launch_table.find_all('th')
for row in flt:
    name = extract_column_from_header(row)
    if name is not None and len(name) > 0:
        column_names.append(name)
```

```
launch_dict= dict.fromkeys(column_names)

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

# Let's initial the launch_dict with each
launch_dict['Flight No.'] = []
launch_dict['Launch site'] = []
```

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

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

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

Data Wrangling

1. Fetch the CSV file
2. Calculate the percentage of the missing values
3. Determine the number of launches on each site and the number and occurrence of each orbit in the column Orbit
4. Determine the number of landing outcomes
5. Create a set of outcomes where the second stage did not land successfully
6. Create a list where the element is zero or one based on the outcome
7. Determine the success rate
8. Export it to a CSV

```
URL = 'https://cf-courses-data.s3.us.cloud-object-storage.appdom
resp = await fetch(URL)
dataset_part_1_csv = io.BytesIO((await resp.arrayBuffer()).to_py
```

```
df.isnull().sum()/df.shape[0]*100
```

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

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

```
landing_outcomes = df['Outcome'].value_counts()
```

```
bad_outcomes=set(landing_outcomes.keys()[[1,3,5,6,7]])
```

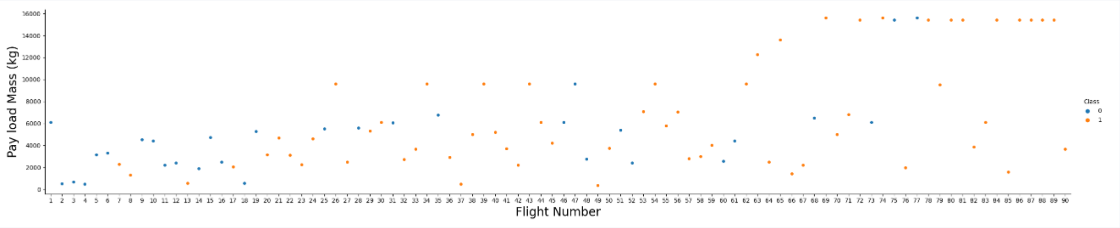
```
landing_class = df['Outcome'].replace({'False Ocean': 0, 'False
df['Outcome'] = df['Outcome'].astype(int)
```

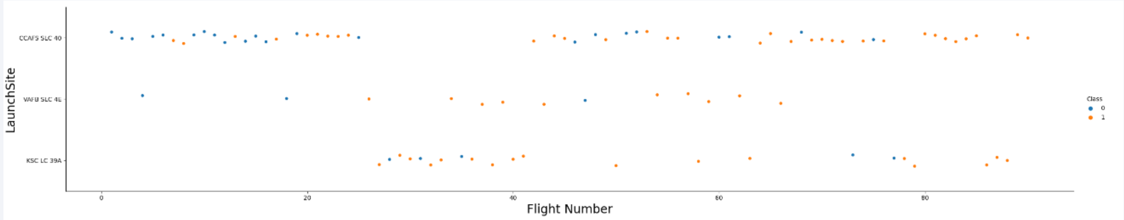
```
df["Class"].mean()
```

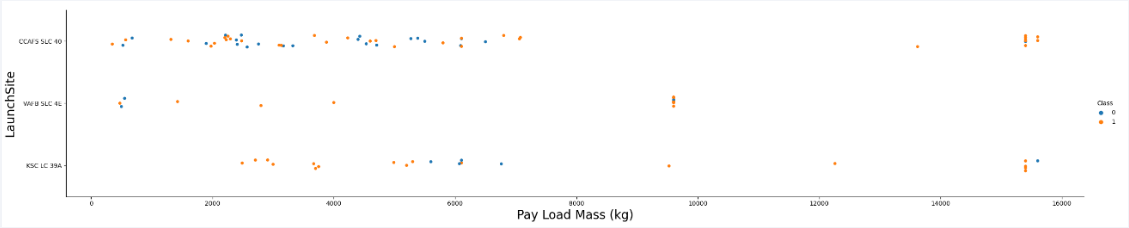
```
df.to_csv("dataset_part_2.csv", index=False)
```

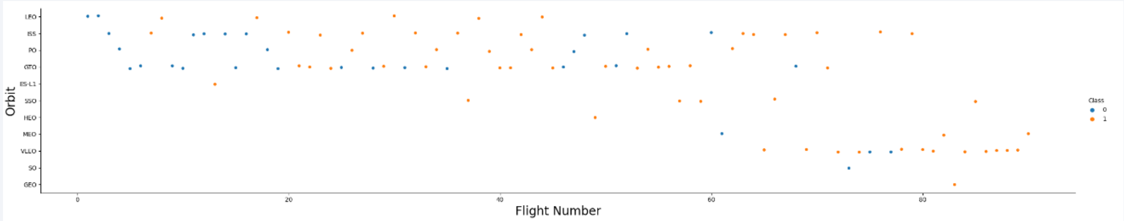
URL: https://github.com/akshayravim/courserarepo/blob/4136eaba27cc53260faf9700e9b32dad496e46ef/IBM-DS0321EN-SkillsNetwork_labs_module_1_L3_labs-jupyter-spacex-data_wrangling_jupyterlite.jupyterlite.ipynb

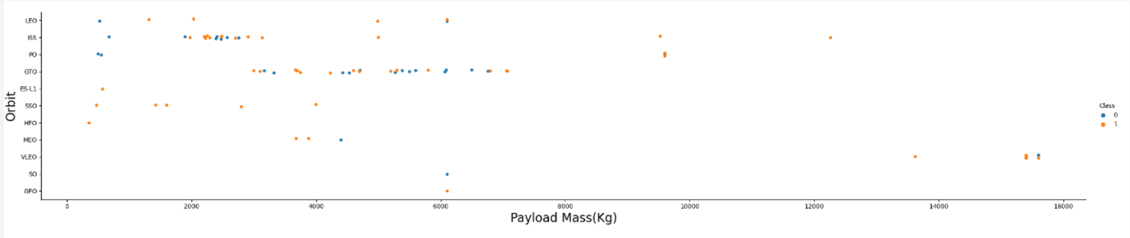
EDA with Data Visualization











Aim is to visualize the relationship between Payload Mass, Flight Number, Flight Number, Launch Site and Orbit type features

EDA with SQL

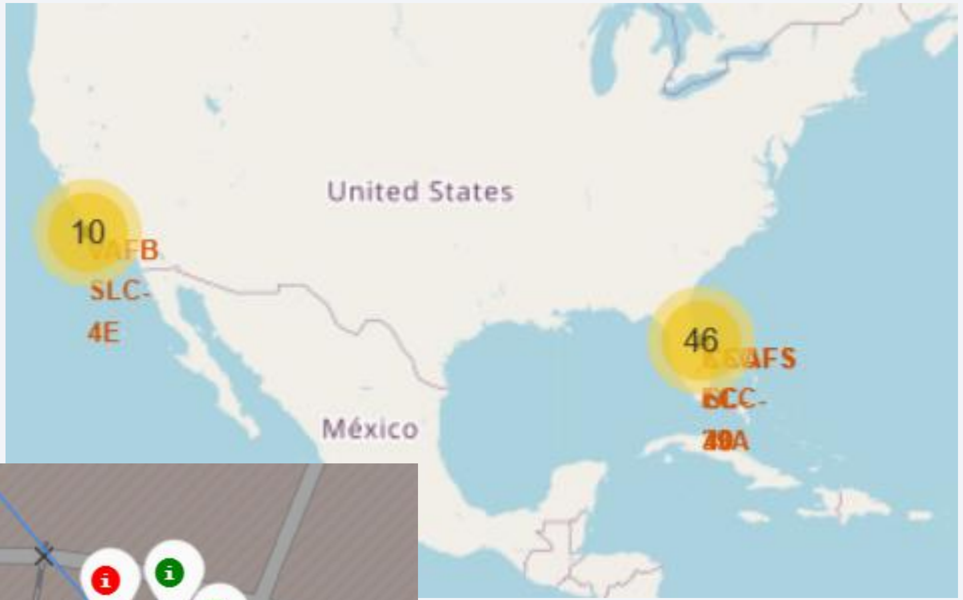
SQL queries performed include:

- **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 achieved.**
- **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 records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.**
- **Rank the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.**

- https://github.com/akshayravim/courserarepo/blob/62190067c0634e2e066541386a304fb737d6434f/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

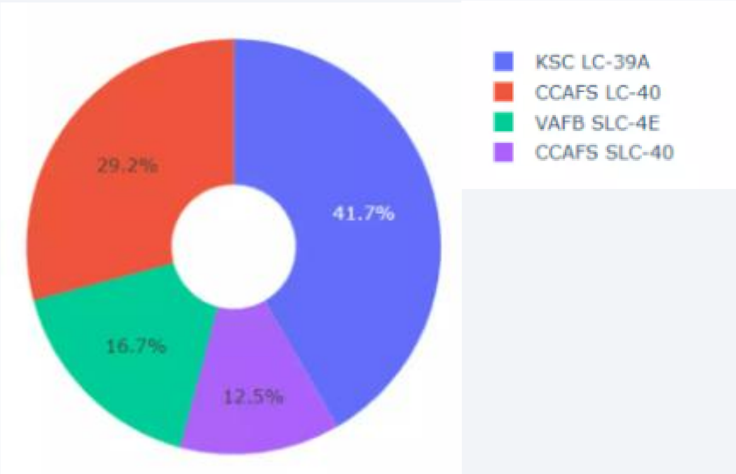
- Circles with numbers were used to depict the number of launches
- Lines were drawn to show and help calculate the distance between the launch sites and the railways, highways etc.
- Red markers were used to indicate failed launches
- Green markers were used to indicate successful launches



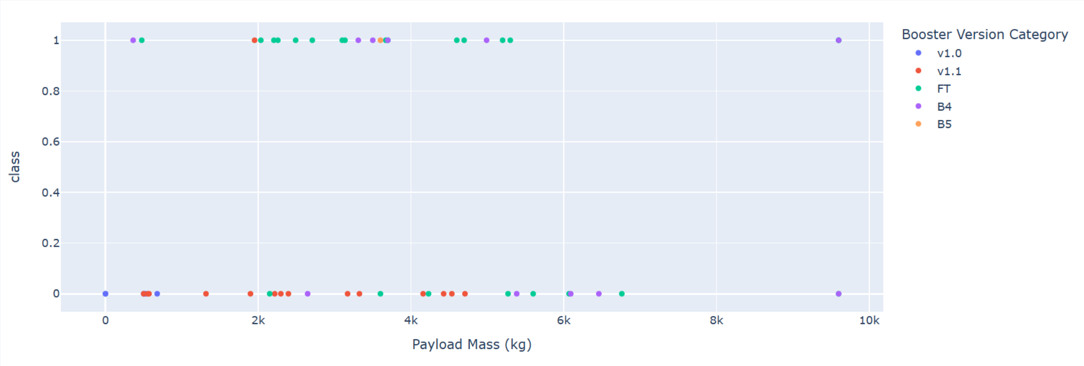
URL:
https://github.com/akshayravim/courserarepo/blob/6e54a15509c62f8c9f9257ebdc3e1ecad938a840/IBM-DS0321EN-SkillsNetwork_labs_module_3_lab_jupyter_launch_site_location.jupyterlite.ipynb

Build a Dashboard with Plotly Dash

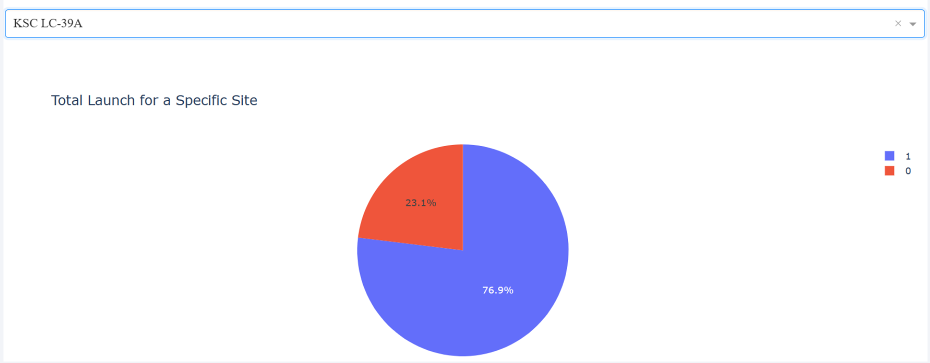
To display the successful launches from the various sites

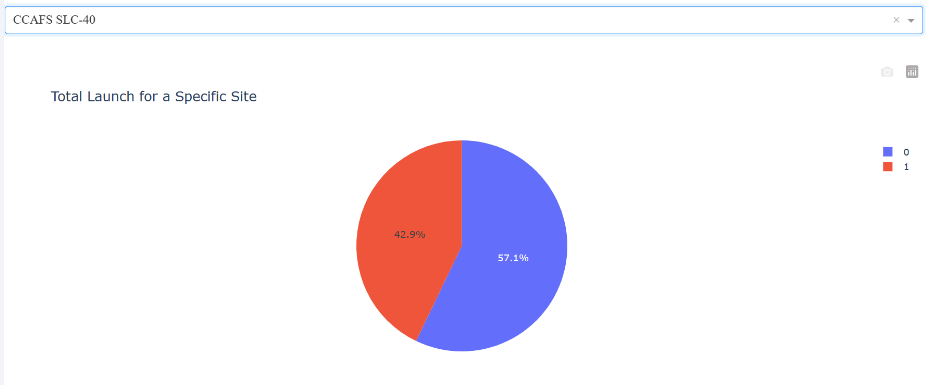


To visualize the correlation between the payload mass and the success or failure



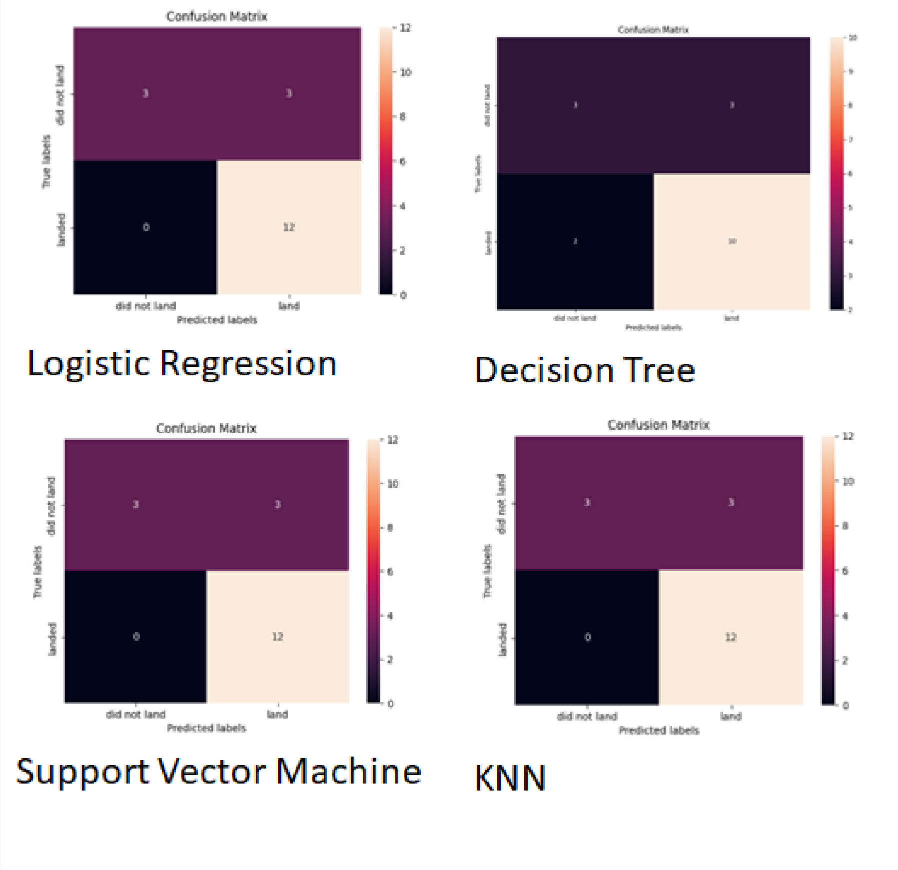
In order to identify the launch sites with the highest and lowest launch success rates





Predictive Analysis (Classification)

- Logistic Regression, SVM and KNN achieved 83.33% accuracy, while Decision Tree achieved 72%



https://github.com/akshayravim/courserarepo/blob/60f6585156dccfecbfcb6f44cea151dd7b3dd457/IBM-DS0321EN-SkillsNetwork_labs_module_4_SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite%20(1).ipynb

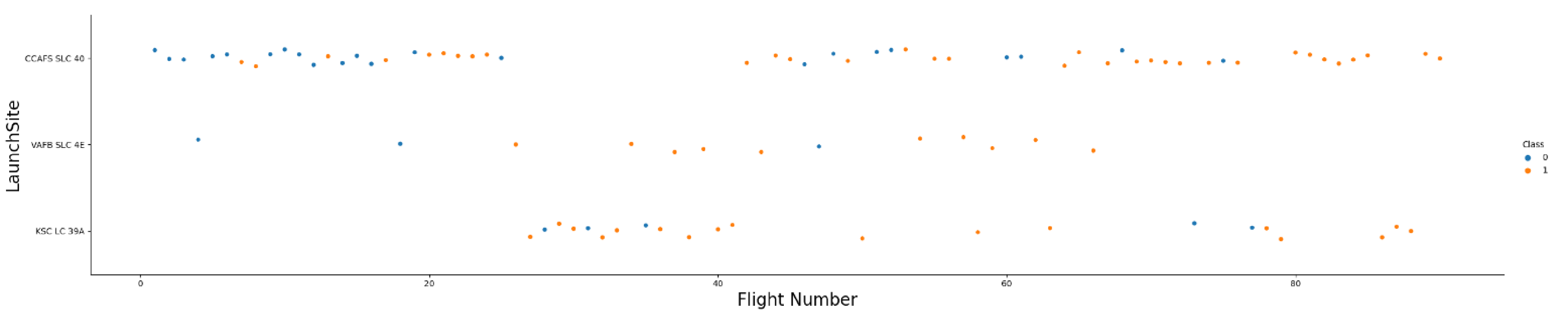
Results

- KSC LC-39A launch site had the most successful launches when compared to other sites
- Among the algorithms, Logistic Regression, KNN and SVM had the highest accuracy
- The Orbits VLEO, SSO, HEO, ESL-1 had the most success rates
- Heavier payloads have failed in comparison to lighter payloads
- Most of the launch sites are in proximity to the equator and the coast lines.
- Lastly, probably due to technological advances and/or learning from mistakes the launch success rates have been increasing over the years.

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

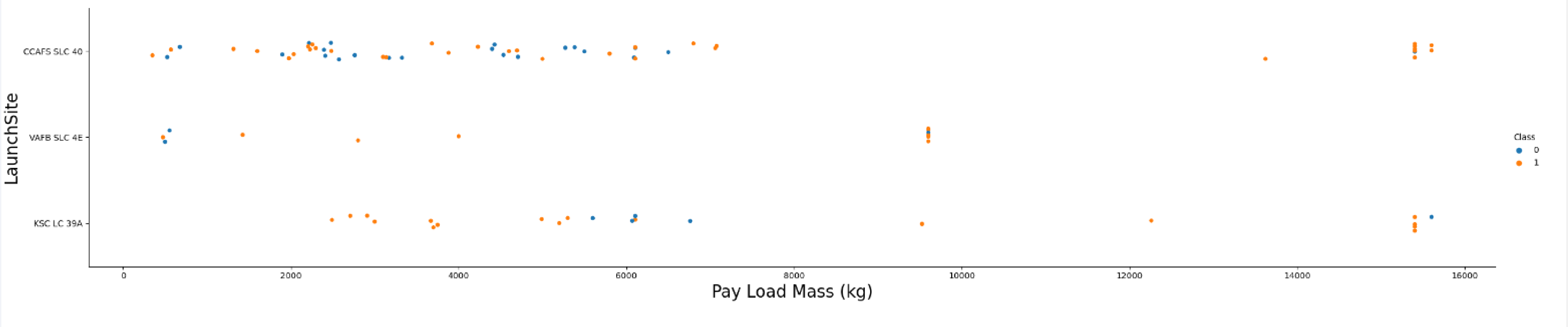


Insight:

Launches from the CCAFS SLC 40 are higher than launches from other sites.

URL:
https://github.com/akshayravim/courserarepo/blob/83e8712a8f45a5e0dff286a5d9b737b0a7bbb611/IBM-DS0321EN-SkillsNetwork_labs_module_2_jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb

Payload vs. Launch Site



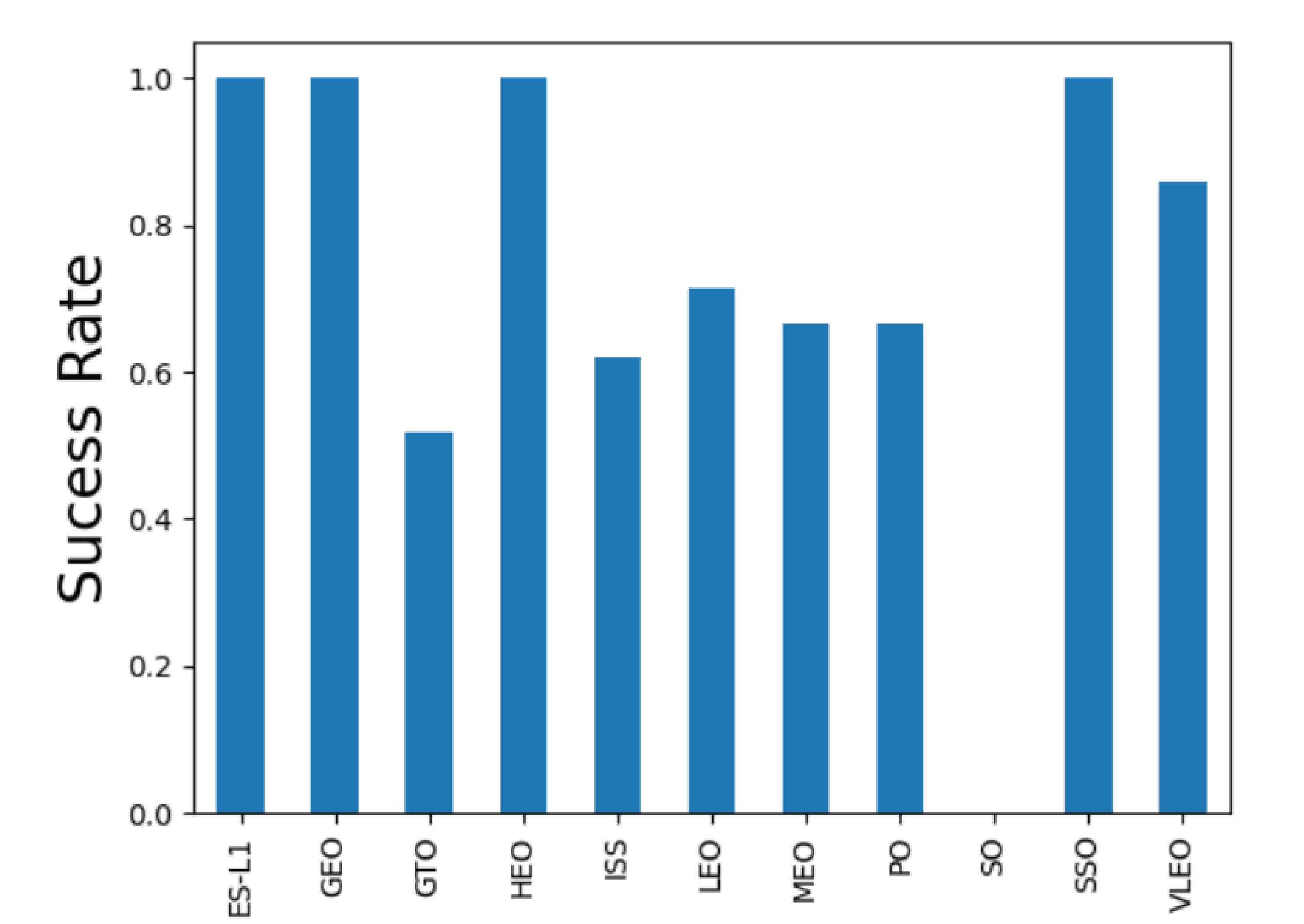
Insight:

Majority of the launches from the CCAFS SLC 40 have had lower pay load mass.

The CCAFS SLC 40 has launched the lowest to the highest pay load masses.

URL:
https://github.com/akshayravim/courserarepo/blob/83e8712a8f45a5e0dff286a5d9b737b0a7bbb611/IBM-DS0321EN-SkillsNetwork_labs_module_2_jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb

Success Rate vs. Orbit Type

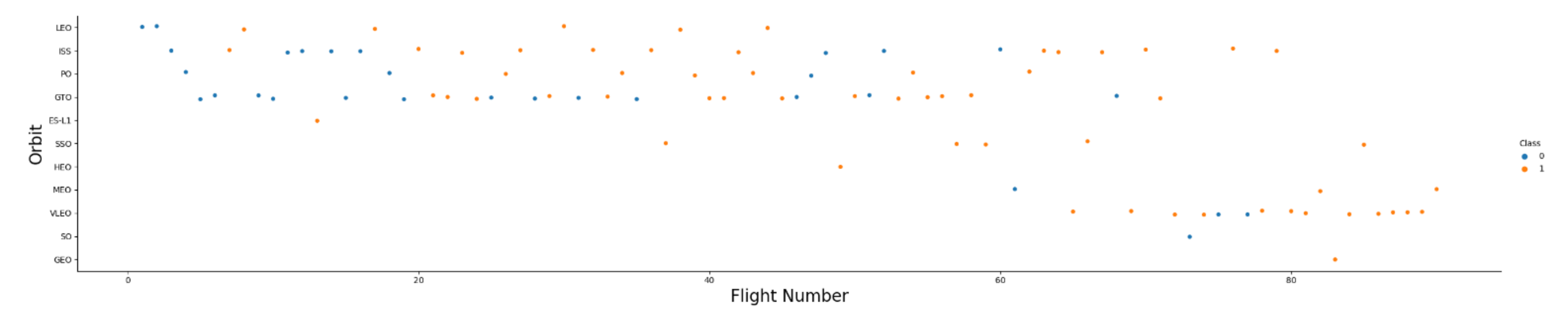


Insight:

Orbit type of ES-L1, GEO, HEO, SSO have the highest success rates

URL: https://github.com/akshayravim/courserarepo/blob/83e8712a8f45a5e0dff286a5d9b737b0a7bbb611/IBM-DS0321EN-SkillsNetwork_labs_module_2_jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb

Flight Number vs. Orbit Type

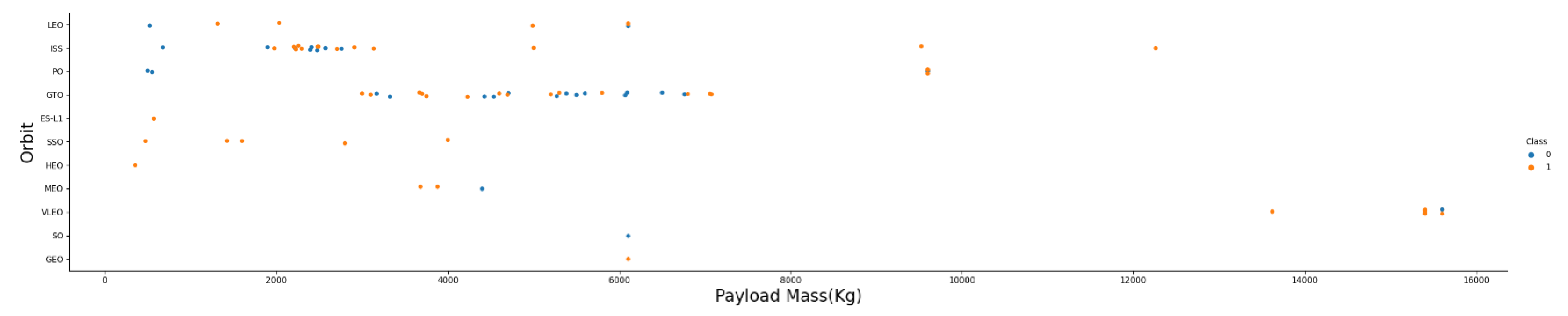


Insight:

For the latest flight number(s), VLEO launches have been used. Appears that the trend is shifting to VLEO for the newer launches.

URL:
https://github.com/akshayravim/courserarepo/blob/83e8712a8f45a5e0dff286a5d9b737b0a7bbb611/IBM-DS0321EN-SkillsNetwork_labs_module_2_jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb

Payload vs. Orbit Type

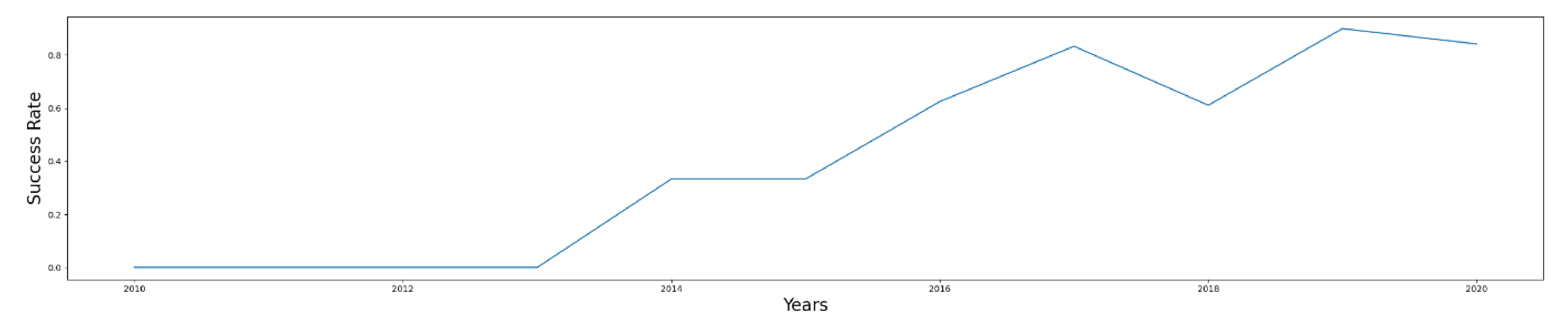


Insight:

Around 2000 payload mass (kg) for ISS and in the range of 4000 to 8000 payload mass (kg) there appears to correlation.

URL:
https://github.com/akshayravim/courserarepo/blob/83e8712a8f45a5e0dff286a5d9b737b0a7bbb611/IBM-DS0321EN-SkillsNetwork_labs_module_2_jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb

Launch Success Yearly Trend



Insight:

Launch success rates has been increasing since 2013 possibly due to technology advances.

URL:
https://github.com/akshayravim/courserarepo/blob/83e8712a8f45a5e0dff286a5d9b737b0a7bbb611/IBM-DS0321EN-SkillsNetwork_labs_module_2_jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb

All Launch Site Names

```
%sql select distinct launch_site from spacextbl
```

```
 * sqlite:///my_data1.db
Done.
```

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

Explanation:

Fetches the unique (distinct) launch sites from the table

URL:
https://github.com/akshayravim/courserarepo/blob/62190067c0634e2e066541386a304fb737d6434f/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Launch Site Names Begin with 'CCA'

```
%sql SELECT LAUNCH_SITE from SPACEXTBL where (LAUNCH_SITE) LIKE 'CCA%' LIMIT 5;
```

```
 * sqlite:///my_data1.db
Done.
```

Launch_Site
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40

Explanation:

Fetches the launch sites from the table where the site names begin with ‘CCA’

URL:
https://github.com/akshayravim/courserarepo/blob/62190067c0634e2e066541386a304fb737d6434f/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Total Payload Mass

```
%sql select sum(PAYLOAD_MASS__KG_) from spacextbl where Customer like 'NASA (CRS)%'
```

```
 * sqlite:///my_data1.db
Done.
```

sum(PAYLOAD_MASS__KG_)
48213

Explanation:

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

URL:
https://github.com/akshayravim/courserarepo/blob/62190067c0634e2e066541386a304fb737d6434f/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Average Payload Mass by F9 v1.1

```
%sql select avg(PAYLOAD_MASS__KG_) from spacextbl where Booster_Version like 'F9 v1.1%'
```

```
 * sqlite:///my_data1.db
Done.
```

avg(PAYLOAD_MASS__KG_)
2534.6666666666665

Explanation:

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

URL:
https://github.com/akshayravim/courserarepo/blob/62190067c0634e2e066541386a304fb737d6434f/jupyter-labs-eda-sql-coursera_sqllite.ipynb

First Successful Ground Landing Date

```
%sql select min(date) from spacextbl where Mission_Outcome = 'Success'

 * sqlite:///my_data1.db
Done.
```

min(date)
01-03-2013

Explanation:

Lists the date when the first successful landing outcome in ground pad was achieved.

URL:
https://github.com/akshayravim/courserarepo/blob/62190067c0634e2e066541386a304fb737d6434f/jupyter-labs-eda-sql-coursera_sqllite.ipynb

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_ BETWEEN 4000 and 6000;
```

```
 * sqlite:///my_data1.db
Done.
```

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

Explanation:

Lists the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000e

URL:
https://github.com/akshayravim/courserarepo/blob/62190067c0634e2e066541386a304fb737d6434f/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Total Number of Successful and Failure Mission Outcomes

```
%sql select count(*),"Mission_Outcome" from spacextbl group by "Mission_Outcome"
```

```
 * sqlite:///my_data1.db
Done.
```

count(*)	Mission_Outcome
1	Failure (in flight)
98	Success
1	Success
1	Success (payload status unclear)

Explanation:

Lists the total number of successful and failure mission outcomes

URL:
https://github.com/akshayravim/courserarepo/blob/62190067c0634e2e066541386a304fb737d6434f/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Boosters Carried Maximum Payload

```
%sql select booster_version from spacextbl where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from spacextbl)
```

```
 * sqlite:///my_data1.db
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

URL: https://github.com/akshayravim/courserarepo/blob/62190067c0634e2e066541386a304fb737d6434f/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Explanation:

Lists the names of the booster versions which have carried the maximum payload mass.

2015 Launch Records

%sql select case substr(Date,4,2) when '01' then 'January' when '02' then 'Febuary' when '03' then 'March' when '04' then 'April' when '05' then 'May' when '06' then 'June' when '07' then 'July' when '08' then 'August' when '09' then 'September' when '10' then 'October' when '11' then 'November' when '12' then 'December' else '' end as Month, "Landing _Outcome", Booster_version from spacextbl where "Landing _Outcome" like 'Failure (drone ship)' and substr(Date,7,4)='2015'

Month	Landing _Outcome	Booster_Version
January	Failure (drone ship)	F9 v1.1 B1012
April	Failure (drone ship)	F9 v1.1 B1015

Explanation:

List the records which will display the month names, failure landing outcomes in drone ship ,booster versions, launch site for the months in year 2015.

URL:
https://github.com/akshayravim/courserarepo/blob/62190067c0634e2e066541386a304fb737d6434f/jupyter-labs-eda-sql-coursera_sqllite.ipynb

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

```
%sql select date, "Landing _Outcome", Booster_version from spacextbl where "Landing _Outcome" like 'Failure (drone ship)' and substr(Date,7,4)='2015'
```

```
 * sqlite:///my_data1.db
Done.
```

Date	Landing _Outcome	Booster_Version
10-01-2015	Failure (drone ship)	F9 v1.1 B1012
14-04-2015	Failure (drone ship)	F9 v1.1 B1015

Explanation:

Ranked count of successful landing outcomes between the date 04-06-2010 and 20-03-2017 in descending order.

URL:
https://github.com/akshayravim/courserarepo/blob/62190067c0634e2e066541386a304fb737d6434f/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Section 3

Launch Sites Proximities Analysis

All site marked on the map

Findings:

1. All the launch sites are in proximity to the Equator line.
2. All the launch sites are in proximity to the coast.

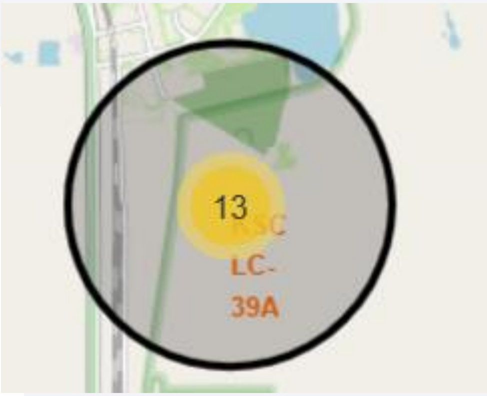
Explanation:

This makes sense as it takes less fuel to escape the earth's gravity to get into space into space from the equator due Earth's rotation, and secondly for safety reasons the launch sites are closer to the coast.

Successful outcomes marked on map

Outcomes marked on the map

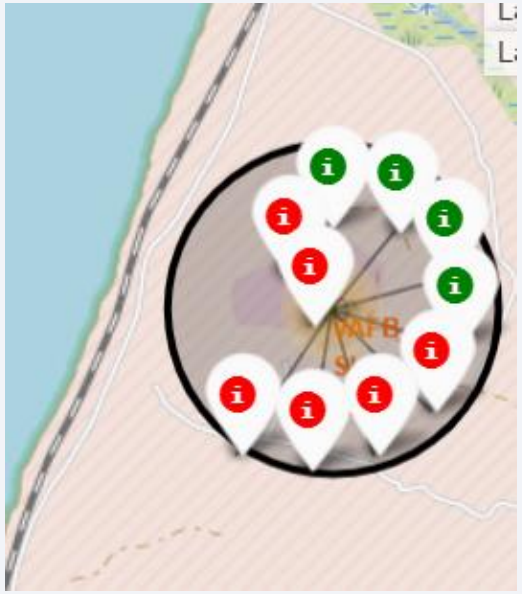
East coast sites



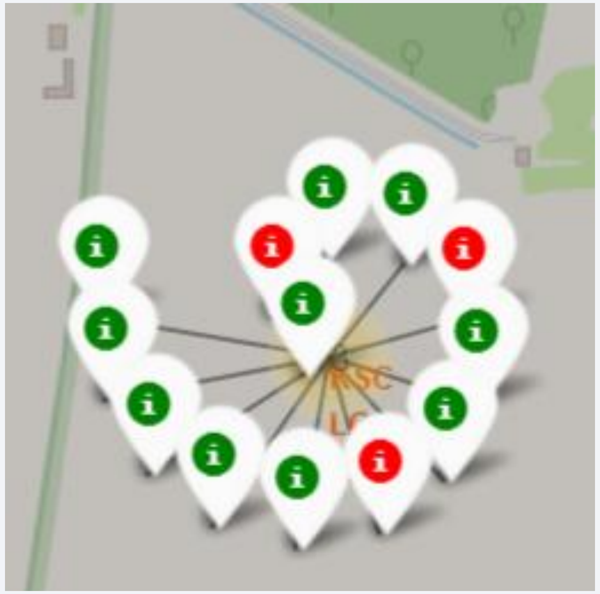
West coast site

Launch sites with color labels

VAFB SLC-4E

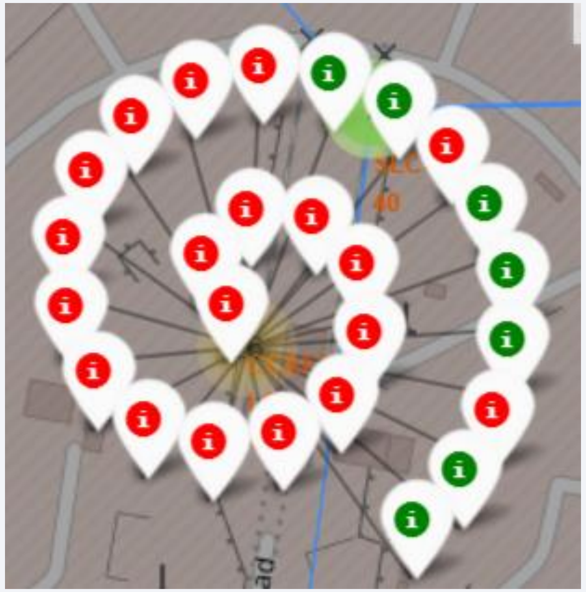


KSC LC-39A



CCAFS SLC-40

CCAFS LC-40

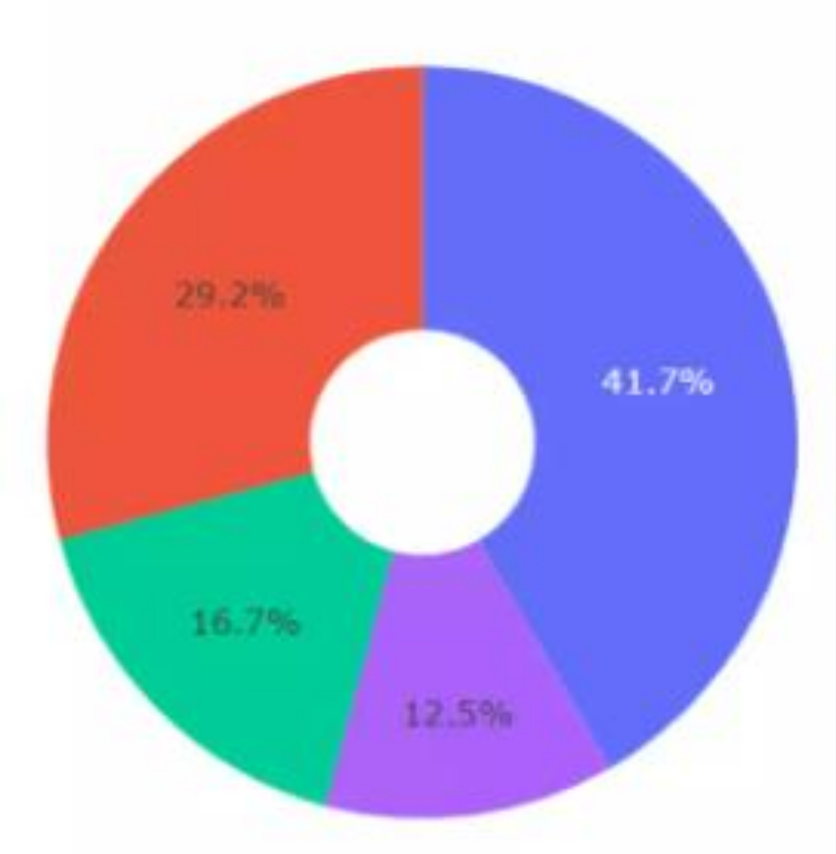


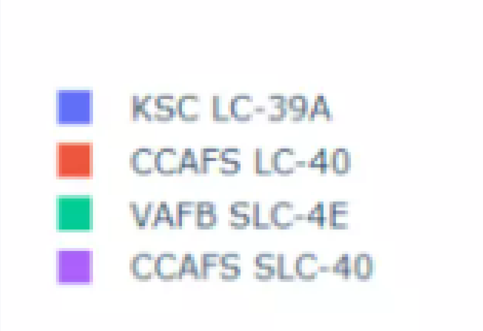
Successful launches marked in Green
Failed launches marked in Red

Section 4

Build a Dashboard with Plotly Dash

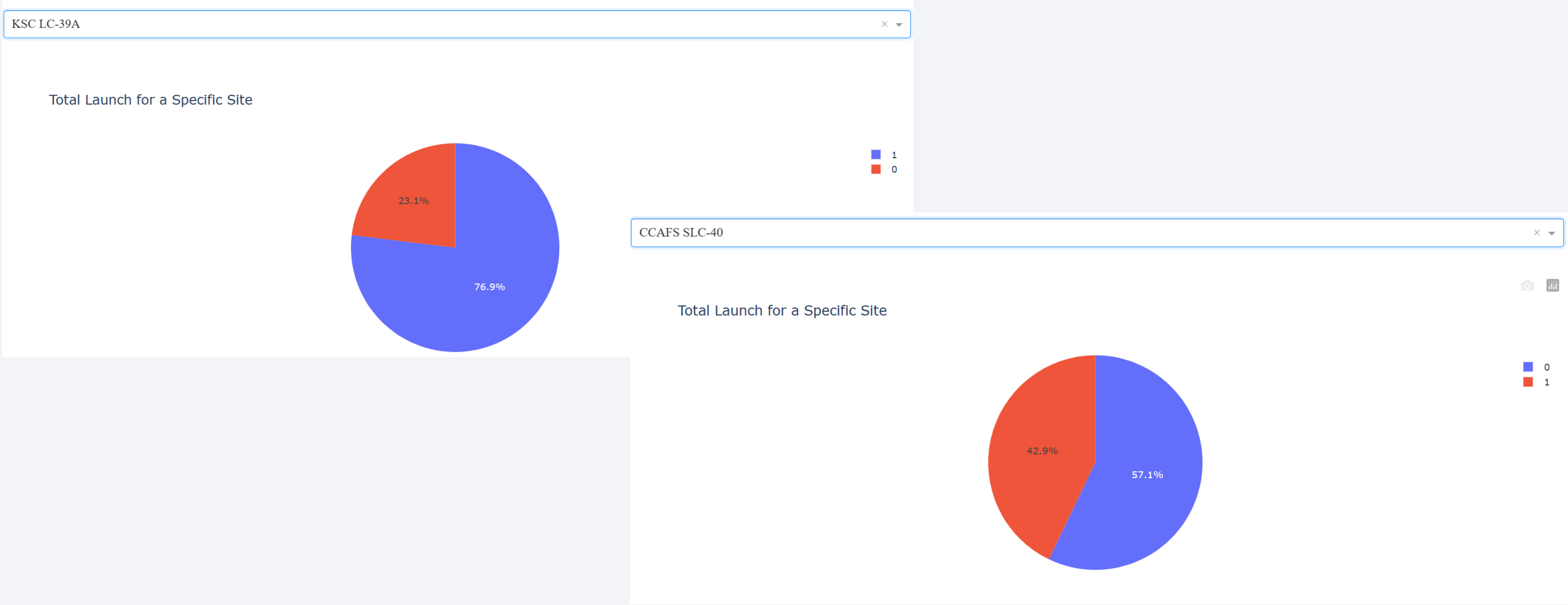
Total successful launches – all sites





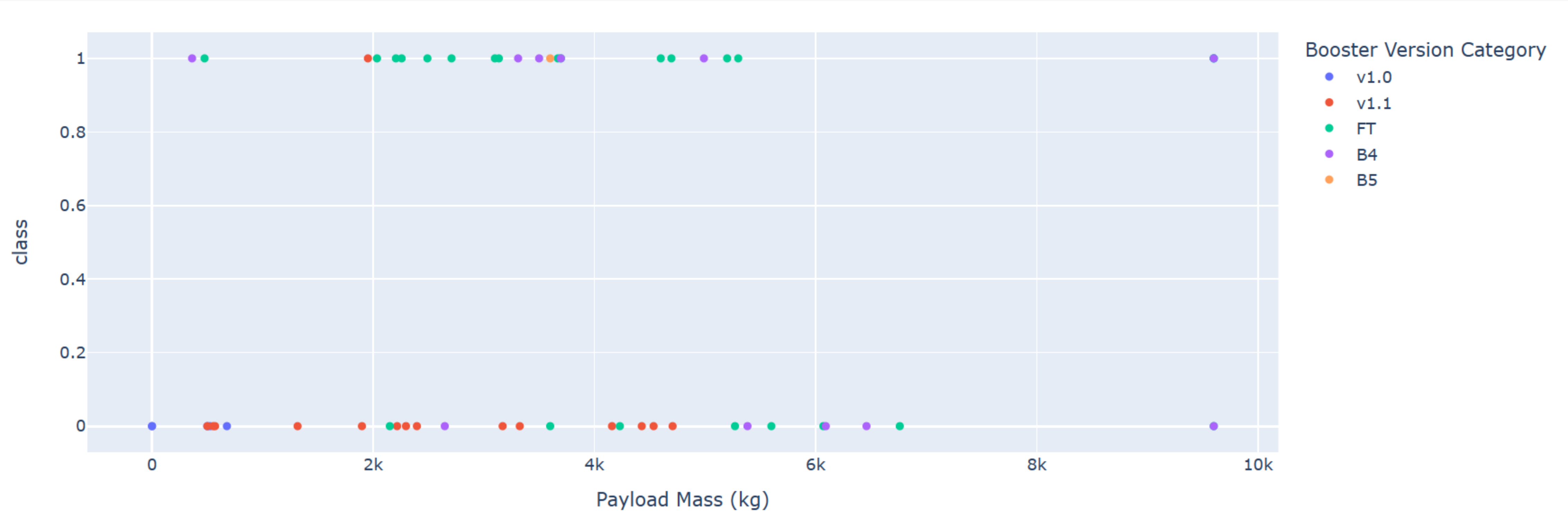
KSC LC-39A has the most successful launches when compared to all other sites

Success rate by launch site



KSC LC-39A achieved the highest of 76.9% success rate while CCAFS SLC-40 has the lowest of 57.1% success rate

Payload mass vs Launch outcome



The low weighted(2K-5K) payloads have a higher success rate than the high weighted(7K-10K) payloads

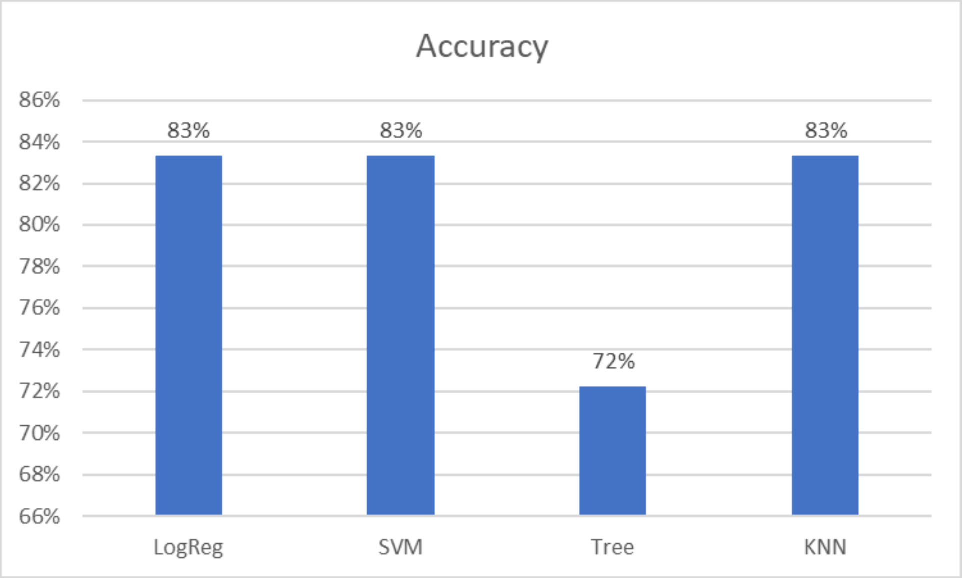
Section 5

Predictive Analysis (Classification)

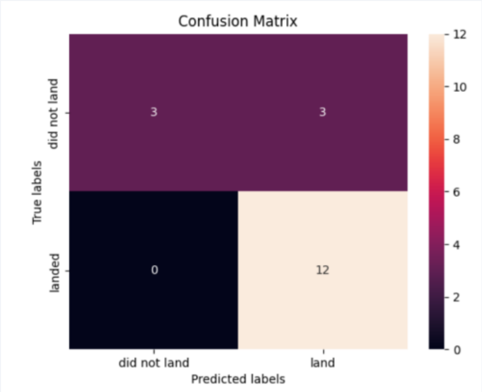
Classification Accuracy

	LogReg	SVM	Tree	KNN
Jaccard_Score	0.800000	0.800000	0.666667	0.800000
F1_Score	0.888889	0.888889	0.800000	0.888889
Accuracy	0.833333	0.833333	0.722222	0.833333

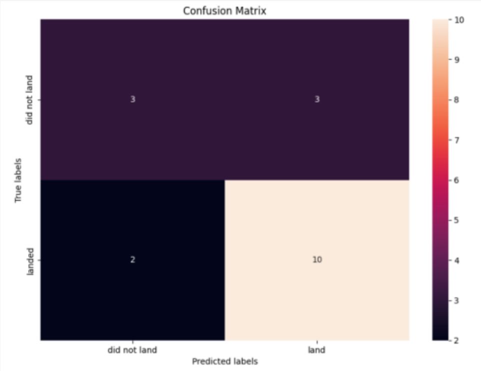
Among the algorithms, Logistic Regression, KNN and SVM had the highest accuracy



Confusion Matrix

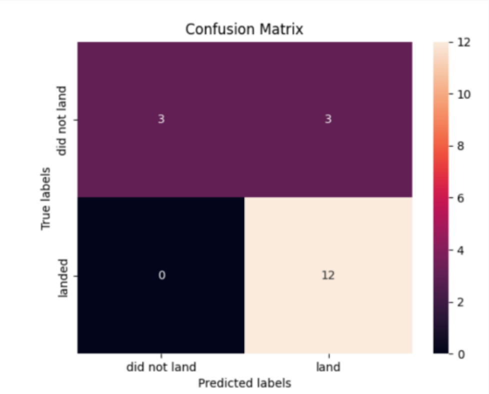


Logistic Regression

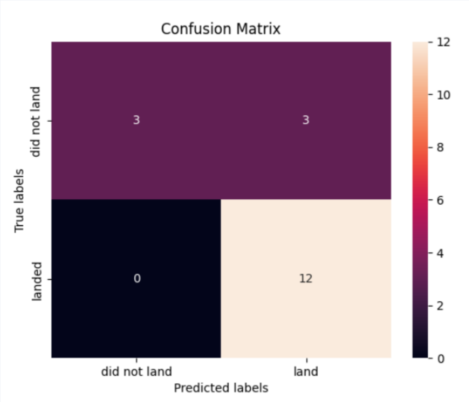


Decision Tree

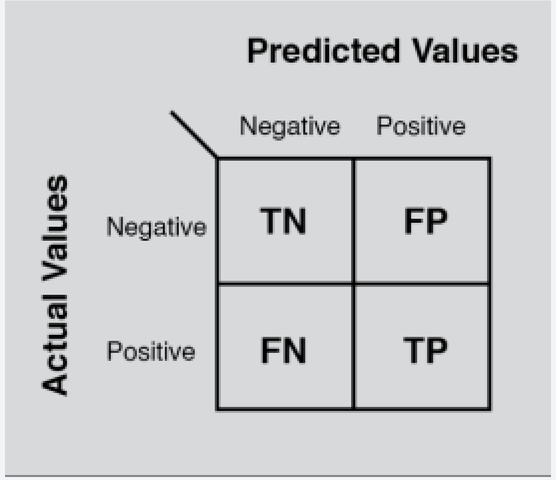
Decision Tree has the lowest accuracy in prediction among the other algorithms deployed.



Support Vector Machine



KNN



Conclusions

- KSC LC-39A launch site had the most successful launches when compared to other sites
- Among the algorithms, Logistic Regression, KNN and SVM had the highest accuracy
- The Orbits VLEO, SSO, HEO, ESL-1 had the most success rates
- Heavier payloads have failed in comparison to lighter payloads
- Most of the launch sites are in proximity to the equator and the coast lines.
- Lastly, probably due to technological advances and/or learning from mistakes the launch success rates have been increasing over the years.

Thank you!