



IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Murphy Gildner
November 27, 2023



Outline



EXECUTIVE
SUMMARY



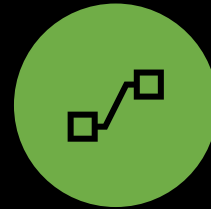
INTRODUCTION



METHODOLOGY



RESULTS

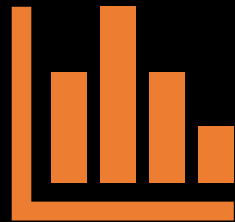


CONCLUSION



APPENDIX

Executive Summary



Summary of methodologies:

- Data Collection
- Data Wrangling
- EDA with Visualization and SQL
- Folium Maps
- Dashboard Generation with Plotly
- Predictive Analysis



Summary of all results:

Classification and visualization of various launch site specifications (success/failure, size/mass, orbit type, etc.)

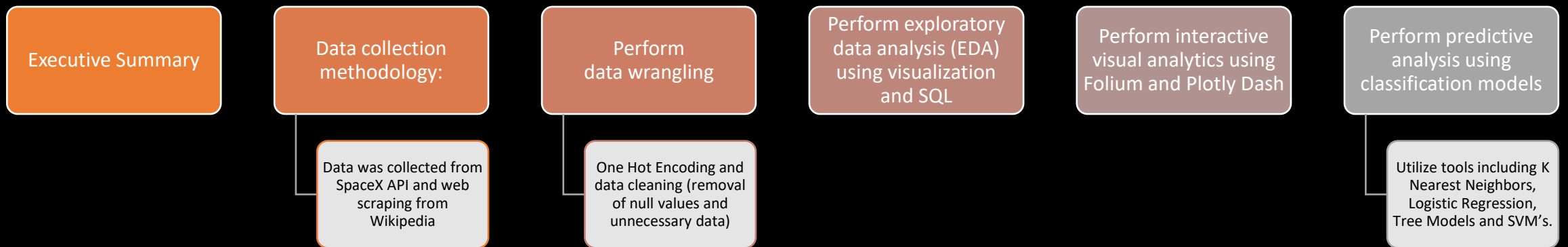
Introduction

- Project background and context:
 - This project outlines the historical data captured from SpaceX launch site datasheets. SpaceX is a space exploration and discovery endeavor with a goal of affordable space expeditions. Analysis of various key variables (mass, orbit type, rocket type, launch location, success rates etc) are need to be performed to determine the optimal parameters for safe and affordable space travel.
- Problems you want to find answers:
 - How do the different variables stated above affect the success rate of expeditions?
 - Have optimization techniques been implemented to increase rate of success?
 - What is the best algorithm for classification of available data?

Section 1

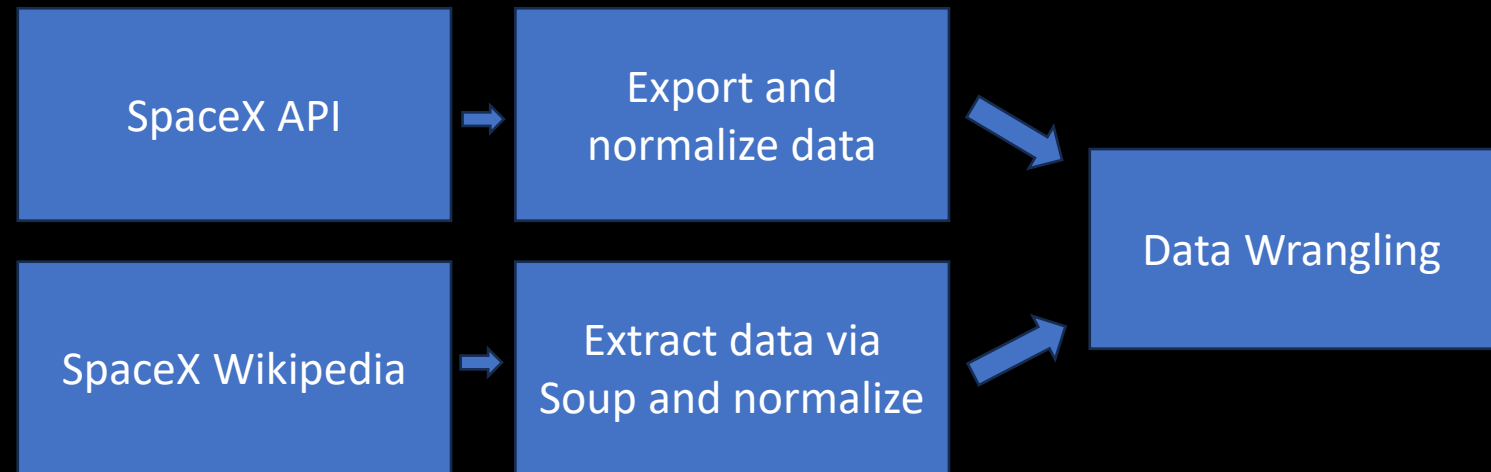
Methodology

Methodology



Data Collection

- Datasets were collecting utilizing SpaceX's API and web scraping Wikipedia via BeautifulSoup.
- The data exported included information including launch sites, date, payload delivery, landing outcome, rocket type, and more.
- Data had to be normalized be removal of null values, irrelevant columns, and entity manipulation (integer/float vs string, etc).



<https://github.com/mgildner27/Capstone-Project/blob/main/jupyter-labs-spacex-data-collection-api.ipynb>

Data Collection – SpaceX API

Connect to SpaceX API

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

Request and Normalize SpaceX Launch Data

```
[9]: static_json_url="https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/dataset
[12]: # Use json_normalize meethod to convert the json result into a dataframe
data = pd.json_normalize(response.json())
```

Call Significant Data Points

```
[19]: # Call getLaunchSite
getLaunchSite(data)
[20]: # Call getPayloadData
getPayloadData(data)
[21]: # Call getCoreData
getCoreData(data)
```

Create Dictionary and Data Frame

Finally lets construct our dataset using the data we have obtained. we we combine

```
[22]: 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,
                    'ReusedCount': ReusedCount,
                    'Serial': Serial,
                    'Longitude': Longitude,
                    'Latitude': Latitude}
```

Then, we need to create a Pandas data frame from the dictionary launch_dict.

```
[23]: # Create a data from launch_dict
df = pd.DataFrame.from_dict(launch_dict)
```

Filter and Export of Data Frame

```
: # Hint data['BoosterVersion']!='Falcon 1'
data_falcon9=df[df['BoosterVersion']!='Falcon 1']
data_falcon9.to_csv('dataset_part_1.csv', index=False)
```

<https://github.com/mgildner27/Capstone-Project/blob/main/jupyter-labs-spacex-data-collection-api.ipynb>

Data Collection - Scraping

Calling HTML and creating BeautifulSoup Object

TASK 1: Request the Falcon9 Launch Wiki page from its URL.
First, let's perform an HTTP GET method to request the Falcon9 Launch HTML page, as an HTTP response.

```
[1]: # Use requests.get() method with the provided static_url
# assign the response to a object
data = requests.get(static_url).text

Create a BeautifulSoup object from the HTML response:

[2]: # Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(data, "html.parser")

Print the page title to verify if the BeautifulSoup object was created properly
```

Finding Tables

Check the external reference link towards the end of this lab

```
[11]: # 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')
```

Checking/Extracting Column Names

```
[13]: column_names = []

# Apply find_all() function with 'th' element on first_launch_table
# Iterate each th element and apply the provided extract_column_from_header() to get a column name
# Append the non-empty column name ("If name is not None and len(name) > 0") into a list called column_names
for row in first_launch_table.find_all('th'):
    name = extract_column_from_header(row)
    if (name != None and len(name) > 0):
        column_names.append(name)
```

Check the extracted column names

```
[14]: print(column_names)

['Flight No.', 'Date and time ( )', 'Launch site', 'Payload', 'Payload mass', 'Orbit', 'Customer', 'Launch outcome']
```

Creating Dictionary

```
[15]: launch_dict= dict.fromkeys(column_names)

# Remove an irrelevant 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']=[]
```

Extract/Append Data (for each column)

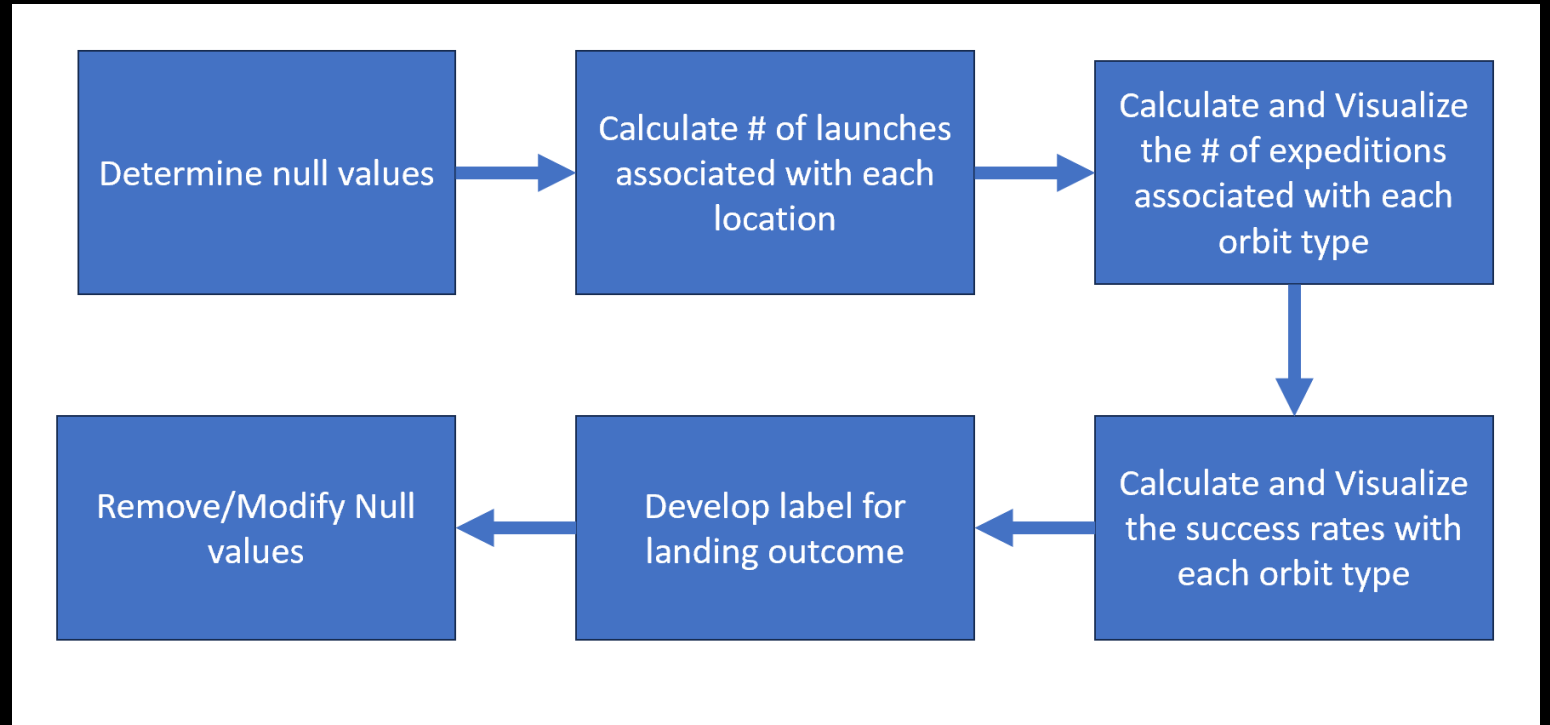
```
[17]: extracted_row = 0
# Extract each table
for table_number, table in enumerate(soup.find_all('table', "wikitable plainrowheaders collapsible")):
    # Get table row
    for row in table.find_all('tr'):
        # Check to see if first table heading is as number corresponding to launch a number.
        if row.th:
            if row.th.string:
                flight_number= row.th.string.strip()
                flag=flight_number.isdigit()
            else:
                flag=False
            # Get table content
            rows=table.find_all('tr')
            # If it is a number, save it in a dictionary
            if flag:
                extracted_row += 1
                # Flight Number value
                # TODO: Append the flight_number into launch_dict with key 'Flight No.'
                launch_dict['Flight No.'].append(flight_number)
                print(flight_number)
                datetime.strptime(row[1])
```

Data Frame Creation

```
[18]: df= pd.DataFrame({ key:pd.Series(value) for key, value in launch_dict.items() })
```

<https://github.com/mgildner27/Capstone-Project/blob/main/jupyter-labs-webscraping.ipynb>

Data Wrangling



https://github.com/mgildner27/Capstone-Project/blob/main/labs-jupyter-spacex-data_wrangling_jupyterlite.jupyterlite.ipynb

EDA with Data Visualization

- Several different visualization tools/charts were used to depict SpaceX data:
 - Scatter plots were used to visualize several key relationships including: flight number vs launch site, flight number vs orbit, payload mass vs launch site, and payload vs orbit.
 - Bar charts were used to visualize the Orbit type and expedition outcome (Success or Failure)
 - A Line plot was used to visualize the increase in success rate over time.
- <https://github.com/mgildner27/Capstone-Project>

EDA with SQL

- SQL queries were used for the following outputs:
 - Names of distinct launch sites
 - Displaying 5 records where Launch Site included 'KSC'
 - Calculating total payload mass of all expeditions combined
 - Calculating average payload mass of all expeditions
 - Listing dates of successful landing outcome
 - Listing the names of boosters with successful landings on a Ground Pad with a weight of 4000-6000kg.
 - Total number of Failed vs Successful missions
 - Booster types that carried maximum payload
 - Ranking the counts of successful landings between 2010-06-04 and 2017-03-20
- <https://github.com/mgildner27/Capstone-Project>

Build an Interactive Map with Folium

- The Maps generated with Folium were utilized to visualize the location of the unique launch sites, the rates of successful and failed expeditions, as well as proximity to the nearest traintracks.
- <https://github.com/mgildner27/Capstone-Project>

Build a Dashboard with Plotly Dash

- The Plotly Dashboard that was generated was used to create an interactive visualization that allows for data to be deliberately selected for analysis. Drop downs were included that allowed filtering of data and, thus, affecting the charts/graphs/tools that were selected for data visualization. Scatter plots and Donut Charts were used for analysis and comparison.
- <https://github.com/mgildner27/Capstone-Project>

Predictive Analysis (Classification)

- Several predictive analysis tools (KNN, LR, SVM, and Tree Models) were used to evaluate the classification model that was generated. It was found that the highest accuracy found within the model was 83.3%.
- <https://github.com/mgildner27/Capstone-Project>

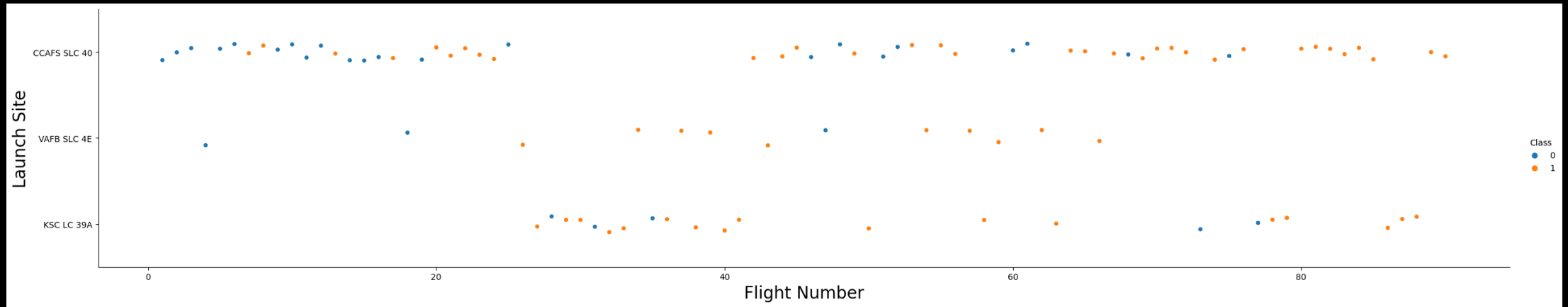
Results

- Several different conclusions can be made from the exploratory data analysis performed including:
 - Success rates increased over time
 - Expeditions with lower payload mass had a higher success rate
 - GEO, HEO, SSO, and ES L1 orbits had the highest rates of success
 - The most successful launch site was KSC LC 39A.
 - The majority of SpaceX expeditions were conducted at CCAFS SLC 40.

The background of the slide is an abstract composition. It features a solid blue area on the left side, which transitions into a dynamic pattern of diagonal streaks in shades of blue and red on the right. These streaks are layered over a faint, light blue grid pattern, creating a sense of depth and movement.

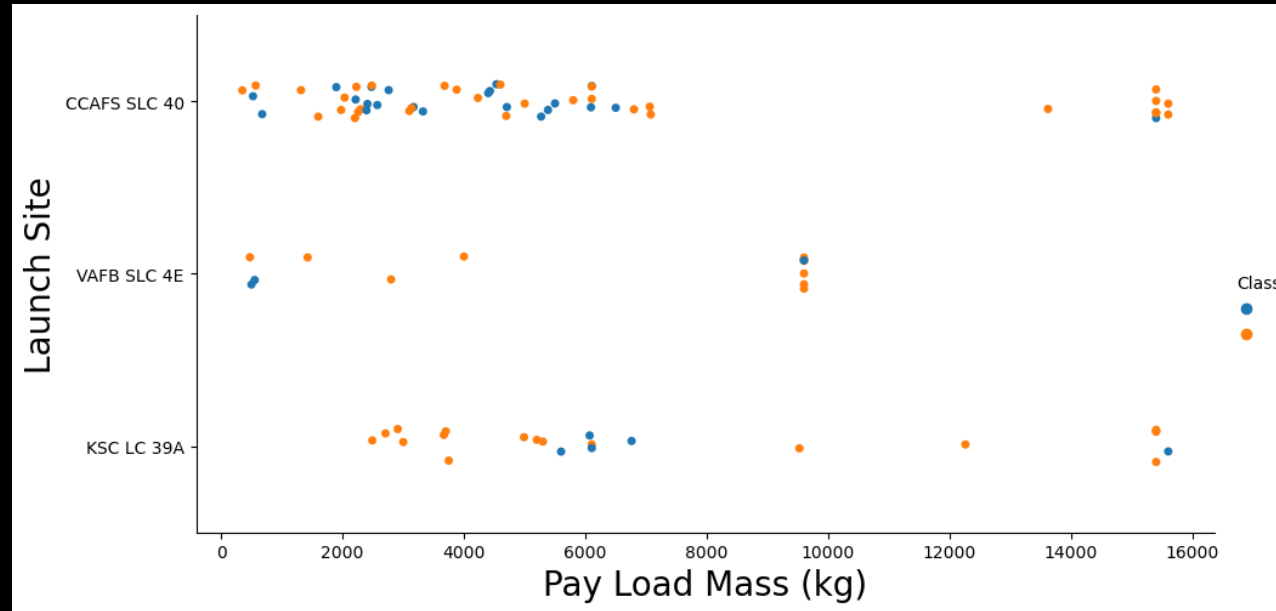
Section 2

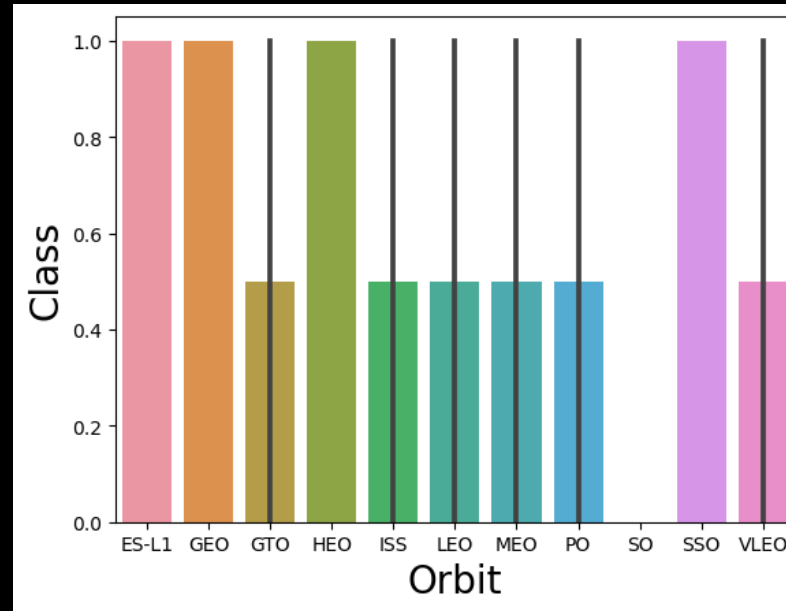
Insights drawn from EDA



Flight Number vs. Launch Site

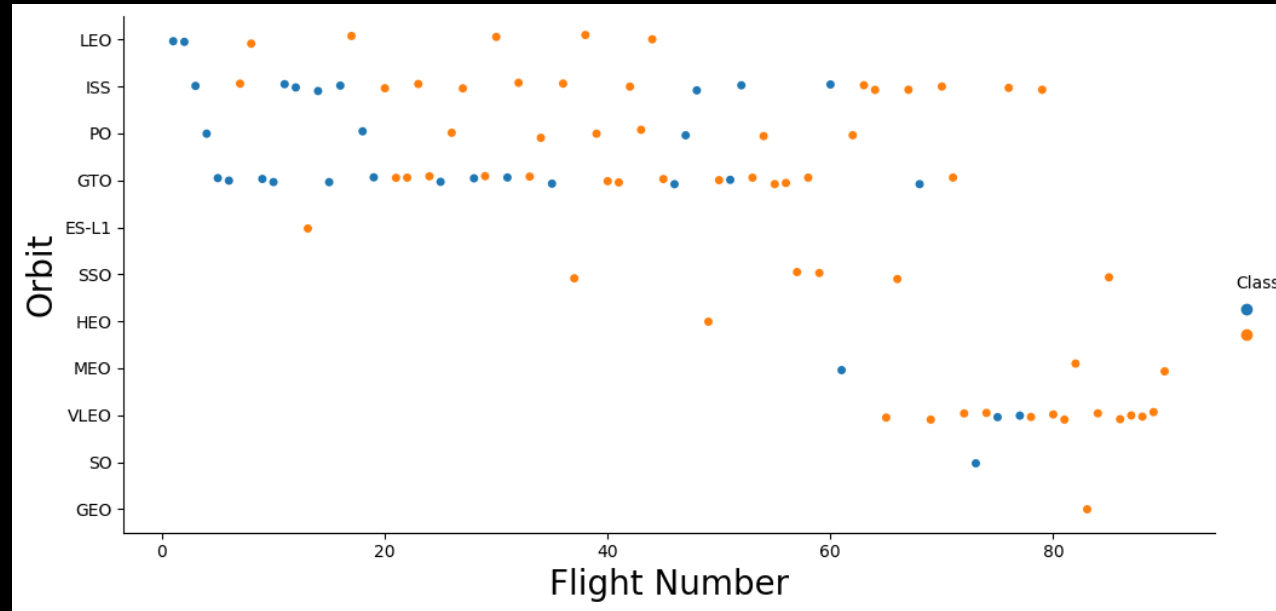
- This scatter plot depicts all SpaceX launches, organized by Launch site and Class.
- Plot is visualized using the flight number along the X-axis with the Launch Site (CCAFS SLC 40, VAFB SLC 4E, or KSC LC 39A).
- The hue variable was assigned based off the column “Class”, being assigned a 0 or 1 with the associated color scheme or blue or orange.





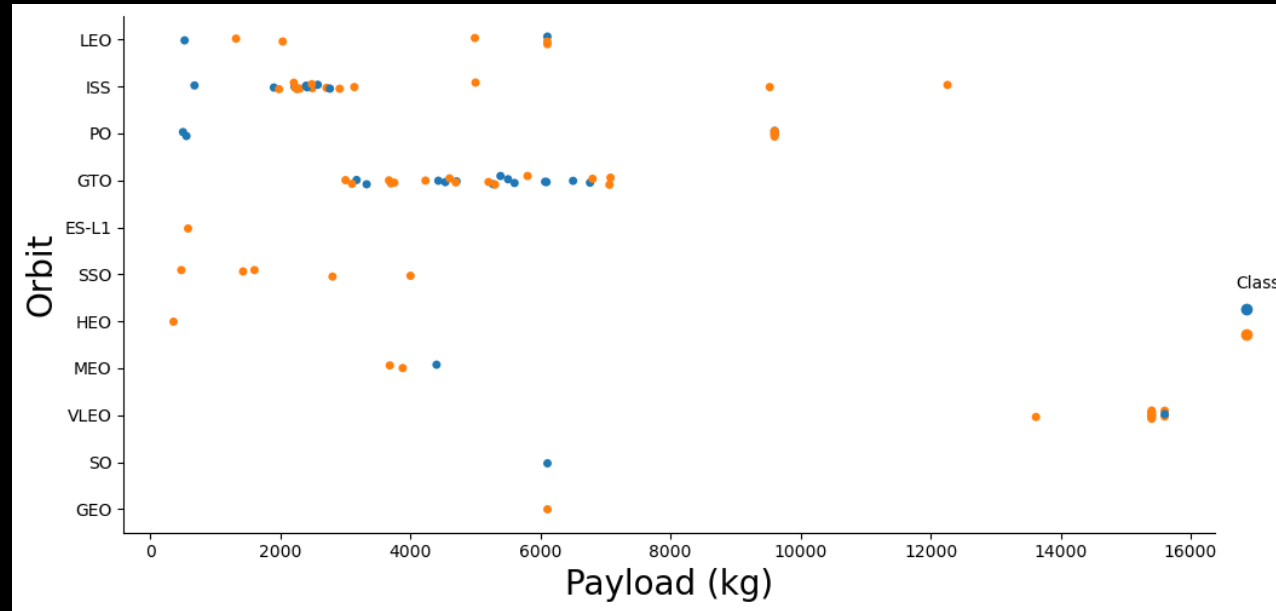
Success Rate vs. Orbit Type

- The bar chart above depicts the success rates based off the desired orbit destination and class.
- The success rates are depicted by the class type, indicating that ES-L1, GEO, and SSO orbits exhibit the highest rates of success.



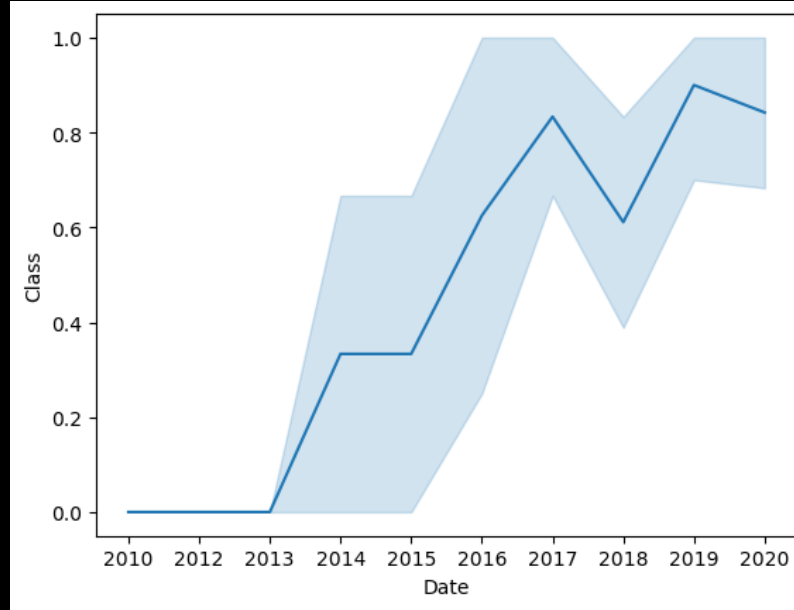
Flight Number vs. Orbit Type

- The scatter plot above visualizes the flight numbers, associated by Orbit type and Class/success rates.
- The earlier expeditions appear to have targeted LEO, ISS, PO, and GTO orbits primarily, whereas later launches were aiming for VLEO orbit.



Payload vs. Orbit Type

- The scatter plot above visualizes the SpaceX launches, arranged by payload mass along the X-axis and Orbit type along the Y-axis. The plot is further analyzed by investigating the class type.
- Most payloads are under 8000kg and target mostly ISS and GTO orbits.



Launch Success Yearly Trend

- The line chart above depicts the success rates of SpaceX expeditions over time.
- Between 2010 and 2020, there is stark evidence that success rates have increased.

All Launch Site Names

```
%sql SELECT distinct(Launch_Site) FROM SPACEXTBL:
```

Launch Site
CCAFS LC-40
CCAFS SLC-40
KSC LC-39A
VAFB SLC-4E

Display 5 records where launch sites begin with the string 'KSC'

```
[9]: %sql SELECT LAUNCH_SITE FROM SPACEXTBL WHERE (LAUNCH_SITE) LIKE "KSC%" LIMIT 5
```

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

Done.

```
[9]: Launch_Site
```

KSC LC-39A

KSC LC-39A

KSC LC-39A

KSC LC-39A

KSC LC-39A

Launch Site Names
Begin with 'KSC'

- This line of code selects 5 distinct instances in which expeditions were conducted at Launch Site KSC LC-39A.

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

```
[10]: %sql SELECT SUM(PAYLOAD_MASS__KG_) AS PAYLOADMASS FROM SPACEXTBL
```

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

```
Done.
```

```
[10]: PAYLOADMASS
```

```
619967
```

Total Payload Mass

- This Line of code takes the sum of the payload mass from all launch numbers in the SpaceX data frame.

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

```
[11]: %sql SELECT avg(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE(Booster_Version) ='F9 v1.1'
```

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

Done.

```
[11]: avg(PAYLOAD_MASS__KG_)
```

2928.4

Average Payload Mass
by F9 v1.1

- This line of code takes the average payload mass from the SpaceX table where the booster version is F9 v1.1

```
[12]: %sql SELECT MIN(DATE) FROM SPACEXTBL WHERE(Mission_Outcome)='Success'
```

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

```
Done.
```

```
[12]: MIN(DATE)
```

```
2010-04-06
```

First Successful Ground
Landing Date

- This line of code finds the earliest date in which a successful expedition was performed by using the “MIN” function on the date column where “Mission Outcome”= Success.

List the names of the boosters which have success in ground pad and have payload mass greater than 4000 but less than 6000

```
[13]: %sql SELECT BOOSTER_VERSION FROM SPACEXTBL WHERE(Landing_Outcome)='Success (ground pad)' and PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000
```

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

Done.

```
[13]: Booster_Version
```

```
F9 FT B1032.1
```

```
F9 B4 B1040.1
```

```
F9 B4 B1043.1
```

Successful Drone Ship
Landing with Payload
between 4000 and 6000

- This line of code selects the booster version types from the SpaceX table where the landing outcome was successful in ground pad, with a payload mass between 4000-6000kg.
- The booster versions that succeeded during their expeditions were F9 FT B1032.1, F9 B4 B1040.1, and F9 B4 B1043.1

[14]: %sql SELECT COUNT(Mission_Outcome) AS MISSIONOUTCOMES FROM SPACEXTBL GROUP BY Mission_Outcome

* sqlite:///my_data1.db

Done.

[14]: MISSIONOUTCOMES

1
98
1
1

Total Number of
Successful and
Failure Mission
Outcomes

The line of code above groups the total number of expeditions by mission outcome.

List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

```
[15]: %sql SELECT Booster_Version AS BOOSTERVERSIONS FROM SPACEXTBL WHERE PAYLOAD_MASS_KG_=(SELECT MAX(PAYLOAD_MASS_KG_) FROM SPACEXTBL)
* sqlite:///my_data1.db
Done.
```

[15]: BOOSTERVERSIONS

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

Boosters Carrying Maximum Load

The line of code above selects the booster version from the SpaceX table where payload mass is at the maximum value, via subquery.

```
[24]: %sql SELECT DATE AS MONTH, LANDING_OUTCOME, BOOSTER_VERSION, LAUNCH_SITE FROM SPACEXTBL WHERE LANDING_OUTCOME = 'Success (ground pad)' AND DATE LIKE '%2017%'
* sqlite:///my_data1.db
Done.
```

```
[24]:
```

MONTH	Landing_Outcome	Booster_Version	Launch_Site
2017-02-19	Success (ground pad)	F9 FT B1031.1	KSC LC-39A
2017-01-05	Success (ground pad)	F9 FT B1032.1	KSC LC-39A
2017-03-06	Success (ground pad)	F9 FT B1035.1	KSC LC-39A
2017-08-14	Success (ground pad)	F9 B4 B1039.1	KSC LC-39A
2017-07-09	Success (ground pad)	F9 B4 B1040.1	KSC LC-39A
2017-12-15	Success (ground pad)	F9 FT B1035.2	CCAFS SLC-40

2017 Launch Records

- The line of code above selects the date, landing outcome, booster version, and launch site of successful expeditions that occurred in 2017.

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

```
[19]: %sql SELECT * from SPACEXTBL where Landing_Outcome like 'Success%' and (Date Between '2010-06-04' and '2017-03-20') order by date desc
* sqlite:///my_data1.db
Done.
```

[19]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS_KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2017-02-19	14:39:00	F9 FT B1031.1	KSC LC-39A	SpaceX CRS-10	2490	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
2017-01-14	17:54:00	F9 FT B1029.1	VAF8 SLC-4E	Iridium NEXT 1	9600	Polar LEO	Iridium Communications	Success	Success (drone ship)
2016-08-14	5:26:00	F9 FT B1026	CCAFS LC-40	JCSAT-16	4600	GTO	SKY Perfect JSAT Group	Success	Success (drone ship)
2016-07-18	4:45:00	F9 FT B1025.1	CCAFS LC-40	SpaceX CRS-9	2257	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
2016-05-27	21:39:00	F9 FT B1023.1	CCAFS LC-40	Thaicom 8	3100	GTO	Thaicom	Success	Success (drone ship)
2016-05-06	5:21:00	F9 FT B1022	CCAFS LC-40	JCSAT-14	4696	GTO	SKY Perfect JSAT Group	Success	Success (drone ship)
2016-04-08	20:43:00	F9 FT B1021.1	CCAFS LC-40	SpaceX CRS-8	3136	LEO (ISS)	NASA (CRS)	Success	Success (drone ship)
2015-12-22	1:29:00	F9 FT B1019	CCAFS LC-40	OG2 Mission 2 11 Orbcomm-OG2 satellites	2034	LEO	Orbcomm	Success	Success (ground pad)

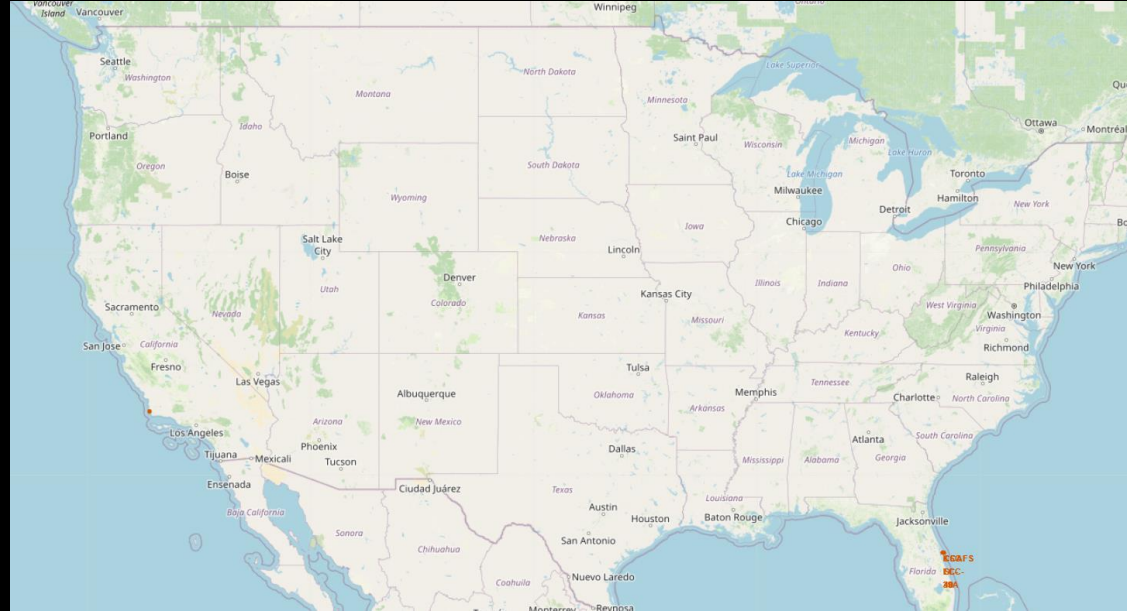
Rank Landing Outcomes Between 2010-06-04 and 2017-03-20

- The output from the code above selects the expeditions based off of successful landing outcomes between the dates of 2010-06-04 and 2017-03-20, ordered by descending date.

A satellite view of Earth from space, showing the curvature of the planet and the glowing city lights of the Eastern United States and parts of Canada at night. The background is a deep blue gradient.

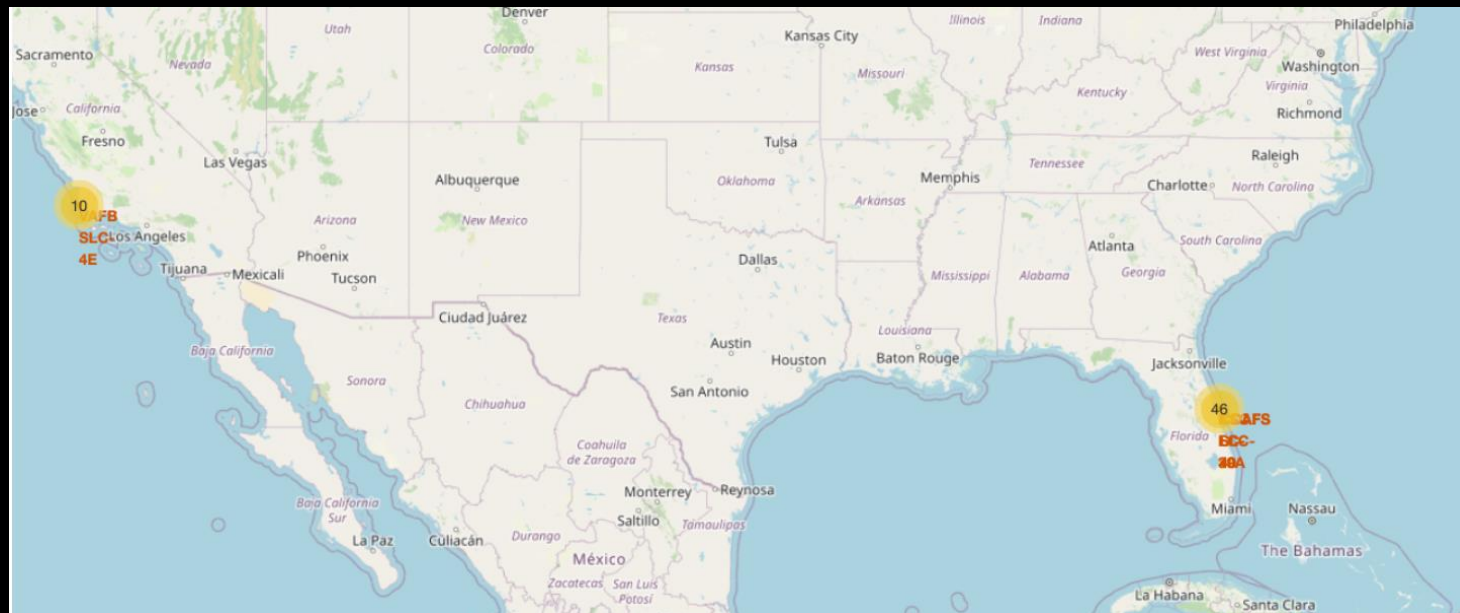
Section 3

Launch Sites Proximities Analysis



Map of Launch Site Locations

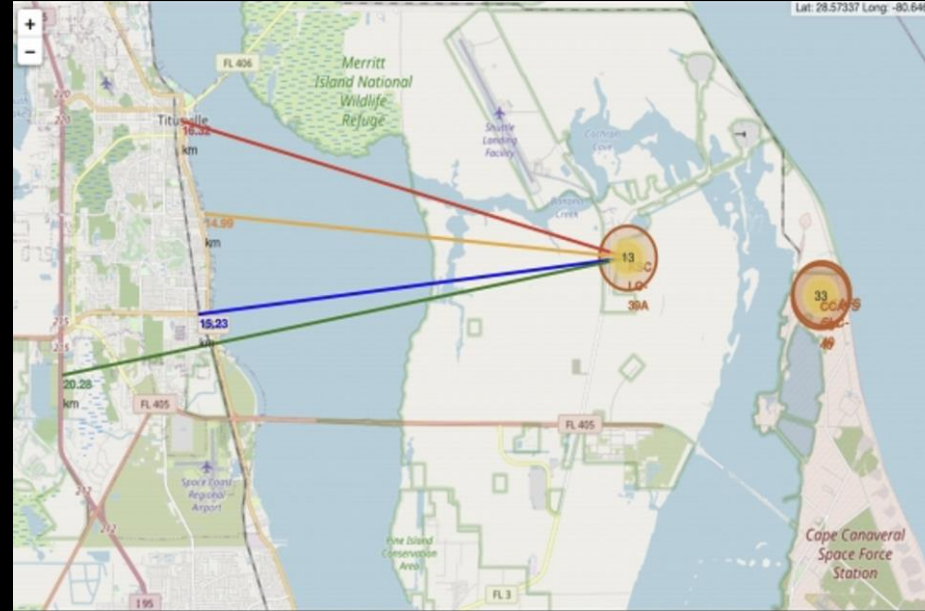
- The folium map above depicts the location of all 3 SpaceX locations with VAFB SLC-4E in southern California and KSC LC-39A, CCAFS SLC-40, and CCAFS LC-40 on the eastern coast of Florida.



Map of Color-Labeled Launch Outcomes

- The map above visualizes the launch locations and their respective success/failure outcomes.

Map Visualizing Distance/Proximity



- Based off the Folium map, it can be determined that Launch Site KSC LC-39A is approximately 20.28km from the nearest highway, 15.23 nearest railroad, and 14.99 to the nearest coastline.



Section 4

Build a Dashboard with Plotly Dash

Total Success Launches by Site



Successful Launches (All Sites)

- The chart above displays the percentage of each launch site relative to the total number of successful launches.
 - KSC LC39A constitutes for 41.2% of the total number of SpaceX successes, followed by CCAFS SLC-40 (23%), VAFB SLC-4E (21.4%), CCAFS LC-40 (14.4%)

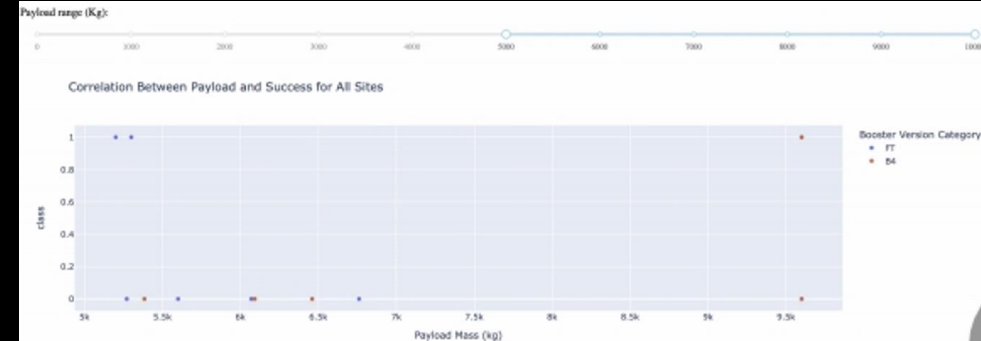
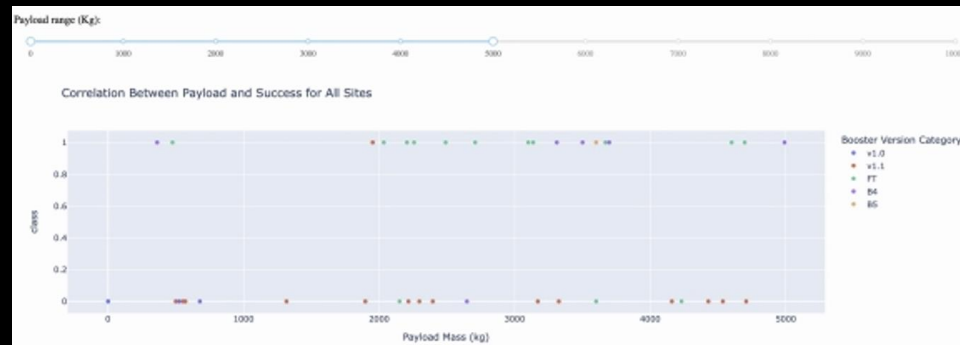
Total Success Launches for Site KSC LC-39A



High Launch Site Success Rates (KSC LC- 39A)

- Launch Site KSC LC-39A had the highest success rate out of all sites, with a rate of 76.9%.

Payload Mass vs Success Rate



- Based off the vizualizaiton tools, these charts depict that payload masses between 2000-5000kg have the highest success rates.
- Booster type v1.1 also seems to positively affect success rates.

Section 5

Predictive Analysis (Classification)

Classification Accuracy

Test:

[109]:	LogReg	SVM	Tree	KNN
Jaccard_Score	0.800000	0.800000	0.800000	0.800000
F1_Score	0.888889	0.888889	0.888889	0.888889
Accuracy	0.833333	0.833333	0.833333	0.833333

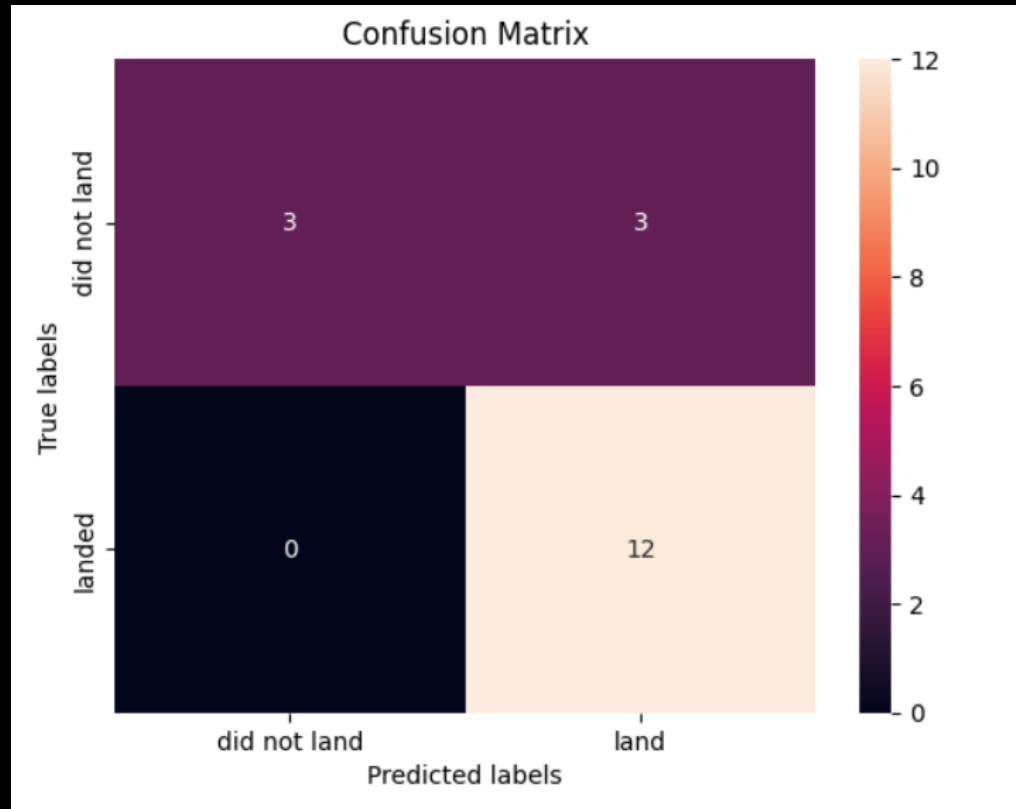
Entire Dataset:

	LogReg	SVM	Tree	KNN
Jaccard_Score	0.833333	0.845070	0.882353	0.819444
F1_Score	0.909091	0.916031	0.937500	0.900763
Accuracy	0.866667	0.877778	0.911111	0.855556

Based off the test model, the optimal classification algorithm cannot be determined due to shared values across the board.

When incorporating the entire dataset, it can be concluded that the Decision tree classifier is the best of this data set as it has the highest jaccard, f1, and accuracy scores.

Confusion Matrix



Best model is DecisionTree with a score of 0.875

Best params is : {'criterion': 'entropy', 'max_depth': 4, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 10, 'splitter': 'best'}

Conclusions

- Many conclusions can be drawn from this data analysis including:
 - KSC LC-39A had the highest success rate out of all launch sites
 - Success rates for all launch sites have increased over years
 - Launches with a lower payload mass tend to be more successful
 - The decision tree model is the ideal algorithm for classification.
 - ES-L1, GEO, and SSO orbits exhibit the highest rates of success

Thank you!

