IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

<Name>
<Date>

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection through API
  - Data Collection with Web Scraping
  - Data Wrangling
  - Exploratory Data Analysis with SQL
  - Exploratory Data Analysis with Data Visualization
  - Interactive Visual Analytics with Folium
  - Machine Learning Prediction
- Summary of all results
  - Explanatory Data Analysis result
  - Interactive analytics
  - Predictive Analytics (Machine learning Lab)

Introduction

SpaceX is a revolutionary company who has disrupt the space industry by offering a rocket launches specifically Falcon 9 as low as 62 million dollars; while other providers cost upward of 165 million dollar each. Most of this saving thanks to SpaceX astounding idea to reuse the first stage of the launch by re-land the rocket to be used on the next mission. Repeating this process will make the price down even further. As a data scientist of a startup rivaling SpaceX, the goal of this project is to create the 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.

Things to notice to achieve that goal are for example, all factor that influence the landing outcome, relation between variables, and which are the best conditions for a successful landing.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was collected using SpaceX REST API and web scrapping a Wikipedia page
- Perform data wrangling
  - Data was processed using one-hot encoding for categorical features
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models. And find the best for this case.

Data Collection

Data collection is the process of gathering and measuring information on targeted variables in an established system, which then enables one to answer relevant questions and evaluate outcomes. As mentioned, the dataset was collected by the REST API and web scrapping a Wikipedia page.

For REST API, you first use a get request. Then, we decode the response content as a Json file and turn it into a pandas dataframe using json_normalize(). We then cleaned the data, checked for missing values and fill with whatever needed.

For web scrapping, we used functions from the BS4 library to extract the launch records as HTML table, parsed the table and convert it to a pandas dataframa for further analysis.

Data Collection – SpaceX API

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

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

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

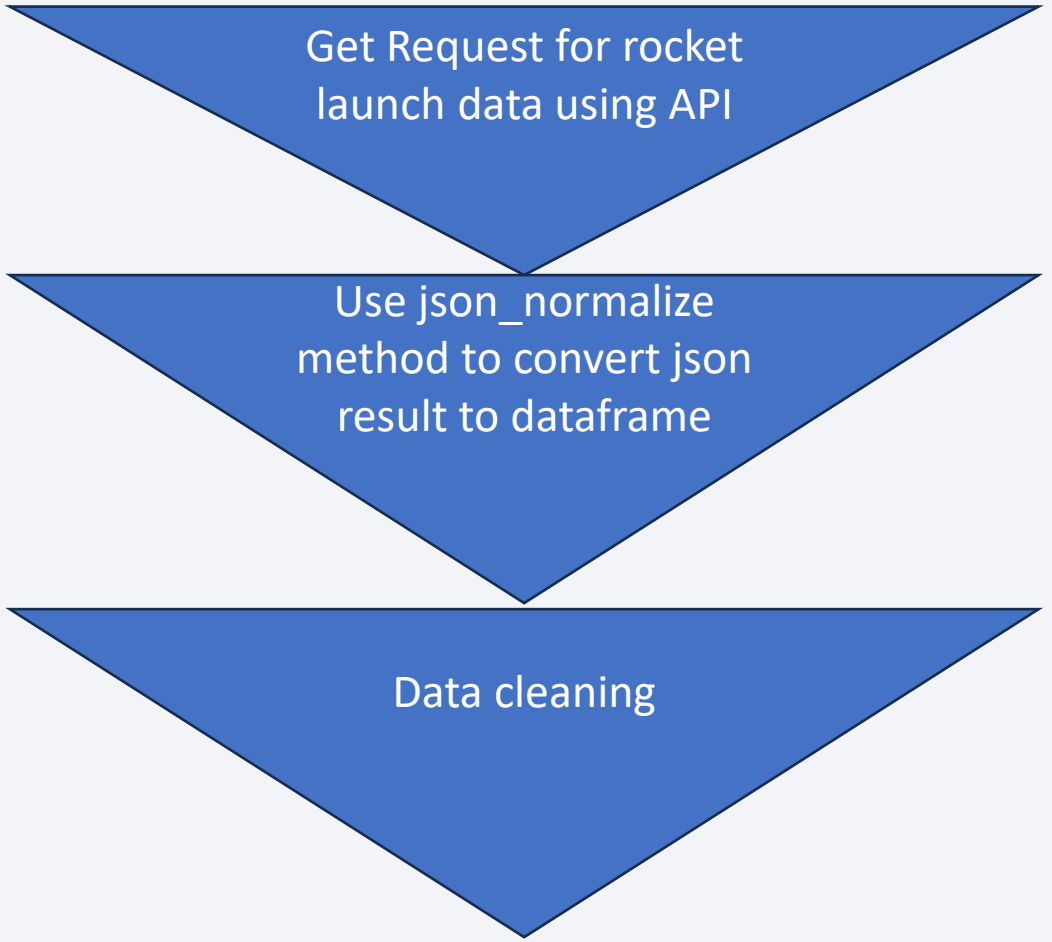
```
# Lets take a subset of our dataframe keeping only the features we want a
nd the flight number, and date_utc.
data = data[['rocket', 'payloads', 'launchpad', 'cores', 'flight_number',
'date_utc']]

# We will remove rows with multiple cores because those are falcon rocket
s with 2 extra rocket boosters and rows that have multiple payloads in a
single rocket.
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 s
ingle 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 ex
tracting 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)]
```



Reference: https://github.com/JhonnyJoestarr/Capstone-DS-Project/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

Data Collection - Scraping

Request the Falcon9 Launch Wiki page from url

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

Create a BeautifulSoup from the HTML response

```
# Use BeautifulSoup() to create a BeautifulSoup object from a response te
xt content
soup = BeautifulSoup(data,'html.parser')
```

Extract all column names from the HTML header

```
extracted_row = 0
#Extract each table
for table_number,table in enumerate(soup.find_all('table',"wikitable plai
nrowheaders collapsible")):
   # get table row
    for rows in table.find_all("tr"):
        #check to see if first table heading is as number corresponding t
o launch a number
        if rows.th:
            if rows.th.string:
                flight_number=rows.th.string.strip()
                flag=flight_number.isdigit()
        else:
            flag=False
```

Reference: https://github.com/JhonnyJoestarr/Capstone-DS-Project/blob/main/jupyter-labs-webscraping.ipynb

Data Wrangling

- Data Wrangling is the process of cleaning and unifying complicated and complex datasets for an easier access and Explanatory Data Analysis (EDA).
- We will first calculate the number of launches on each site, then calculate the number and occurrence of mission outcome per orbit type.
- We then create a landing outcome column. This will make it easier for further analysis, visualization, and ML. Lastly, we will export the result to a CSV.

Use the method `.value_counts()` to determine the number and occurrence of each orbit in the column `Orbit`

```
# Apply value_counts on Orbit column
df.Orbit.value_counts()
```

```
GTO      27
ISS      21
VLEO     14
PO        9
LEO       7
SSO       5
MEO       3
ES-L1     1
HEO       1
SO        1
GEO       1
Name: Orbit, dtype: int64
```

```
# landing_outcomes = values on Outcome column
landing_outcomes = df.Outcome.value_counts()
landing_outcomes
```

```
True ASDS      41
None None      19
True RTLS      14
False ASDS      6
True Ocean      5
False Ocean     2
None ASDS       2
False RTLS      1
Name: Outcome, dtype: int64
```

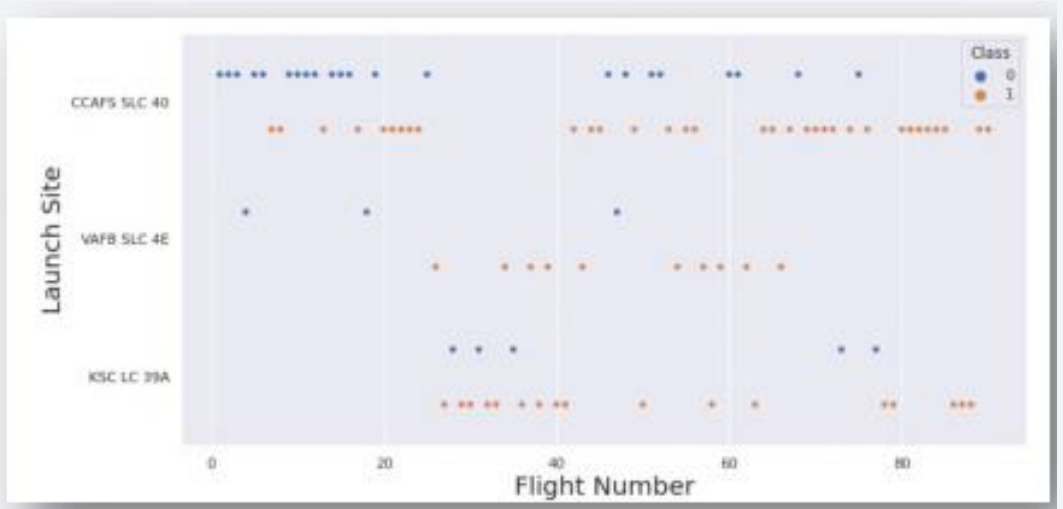

Reference: https://github.com/JhonnyJoestarr/Capstone-DS-Project/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

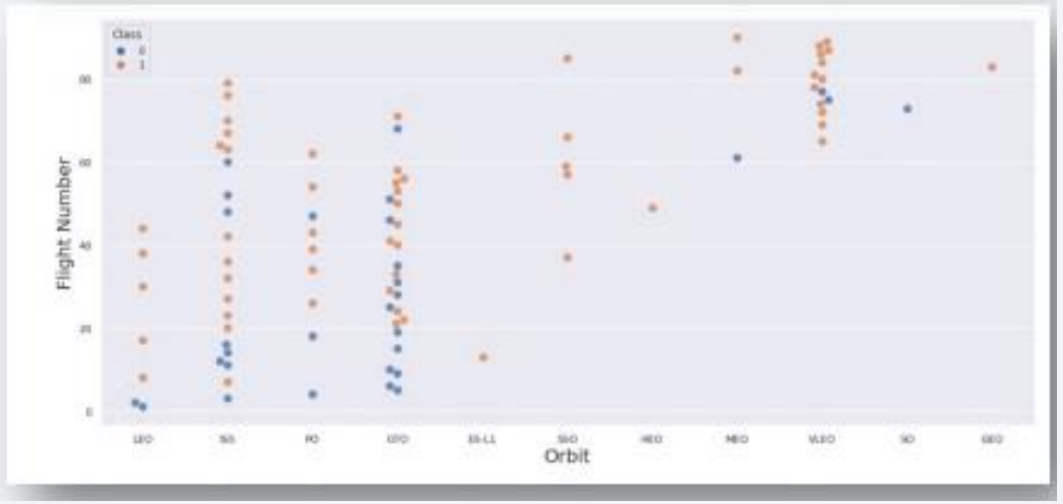
EDA with Data Visualization

We started by using a scatter graph to find the relationship between the attributes such as:

- Payload and flight number.
- Flight number and Launch site
- Flight number and Orbit Type
- Payload and Orbit Type

Scatter plots show dependency of attributes on each other. Once a pattern Is determined from the graphs it is very easy to see which factors are affecting the most to the success of the landing outcomes.





Reference: https://github.com/JhonnyJoestarr/Capstone-DS-Project/blob/main/jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb

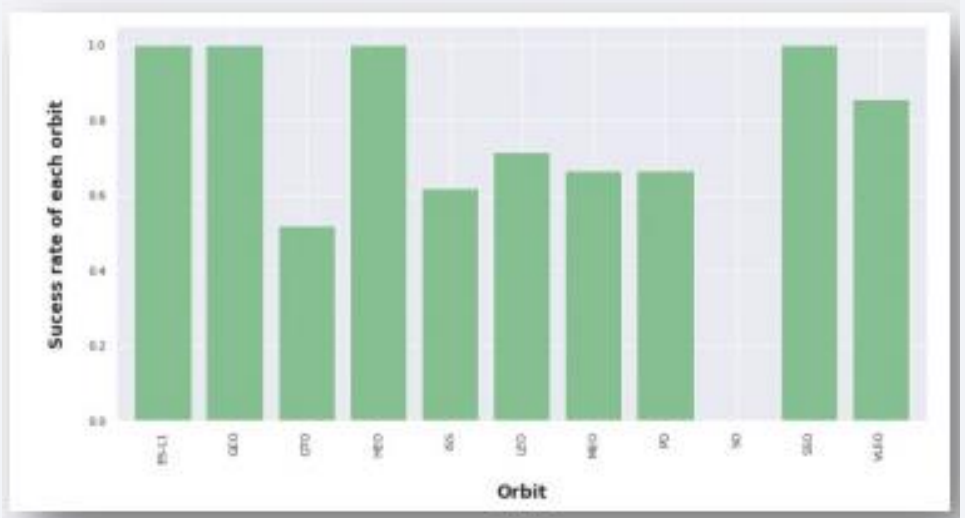
EDA with Data Visualization

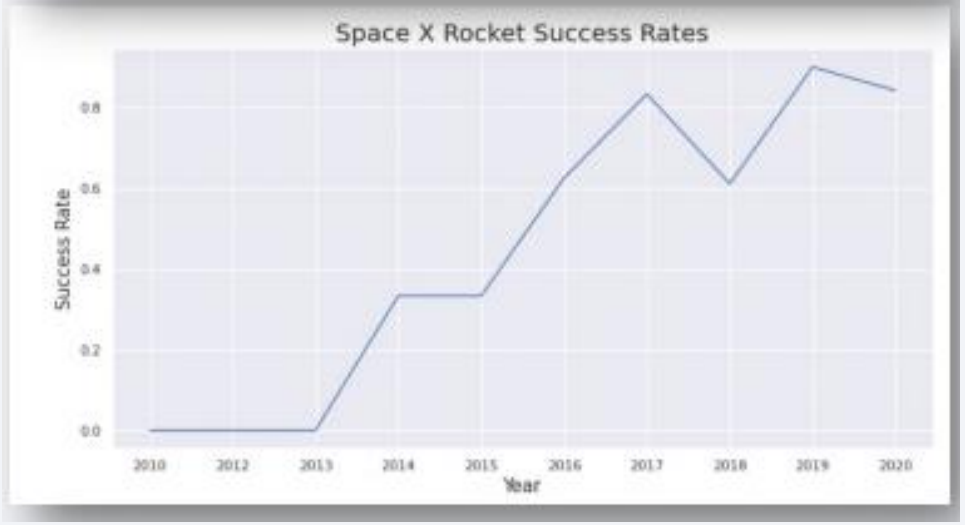
For further analysis we use other visualization tools such as bar graphs and line plots graph, in order to grasp a better comprehension of the data.

Bar graphs are one of the easiest way to interpret the relationship between the attributes. In this case, we are using it to see which orbits have the highest probability of success.

We then use the line graph to show a trend over time, in this case the launch success over the last 10 years.

We then use feature engineering to predict the success in the future module by creating dummy variables out of categorical columns.





Reference: https://github.com/JhonnyJoestarr/Capstone-DS-Project/blob/main/jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb

EDA with SQL

- Using SQL magic in our python environment we performed many queries to get a better understanding of the dataset. Some examples are:
  - Displaying the names of the launch sites
  - Displaying 5 records where the launch site begins with the letters “CCA”.
  - Displaying the total payload mass carried by booster launched by NASA (CRS).
  - Displaying the average payload mass carried by booster version F9 v1.1.
  - Listing the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000.
  - Listing the names of the booster versions which have carried the maximum payload
  - Rank the count of landing outcomes or success between the date 2010-06-04 and 2017-03-20 in descending order.

Reference: https://github.com/JhonnyJoestarr/Capstone-DS-Project/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- To visualize the launch data into an interactive map. We took the latitude and longitude coordinates at each launch site and added a circle marker around each launch label of the name of the launch site.
- We then assigned the dataframe launch_outcomes (with categorical values failure, success) to classes 0 and 1 respectively, and showed them on the map with MarkerCluster().
- This all was done In order to see the zone and locate near places.

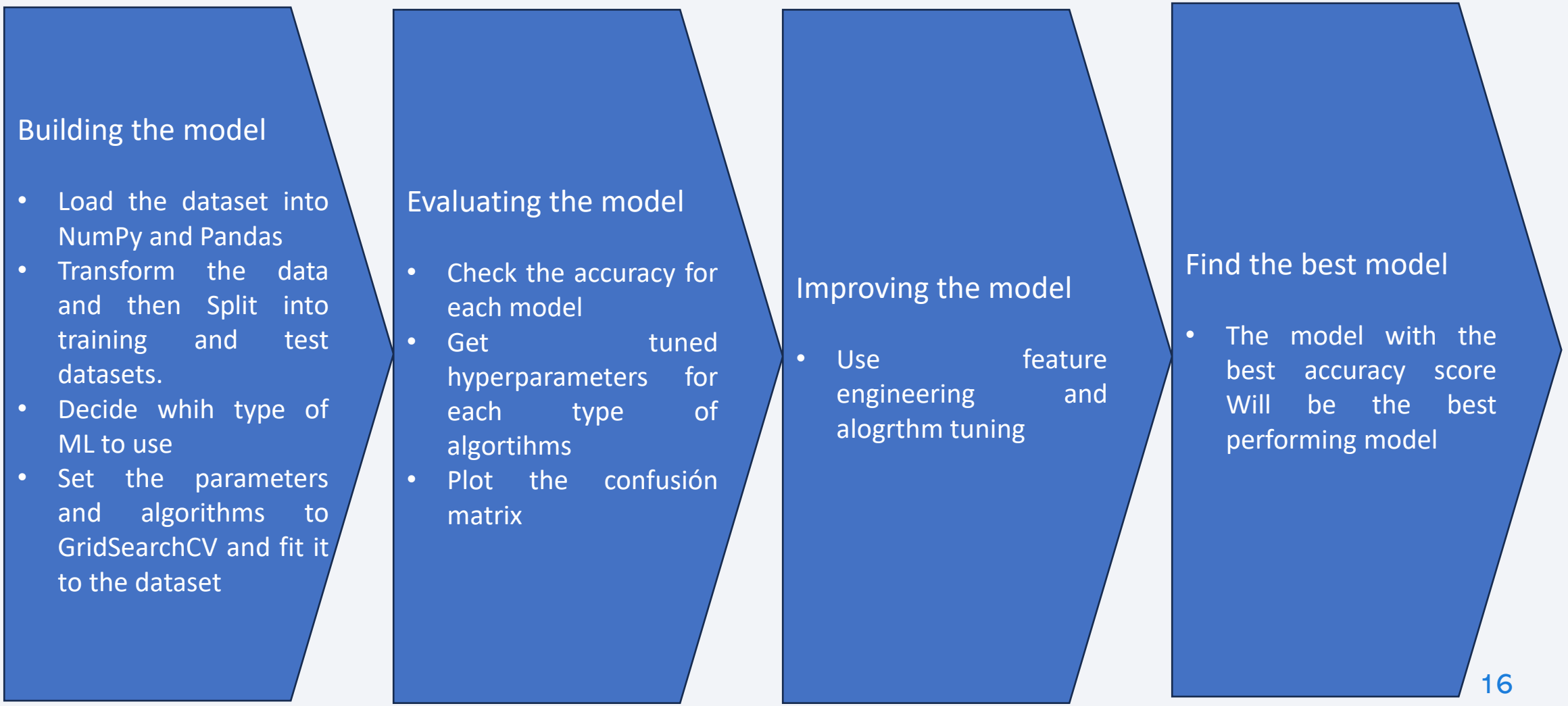
Reference: https://github.com/JhonnyJoestarr/Capstone-DS-Project/blob/main/lab_jupyter_launch_site_location.jupyterlite.ipynb

Build a Dashboard with Plotly Dash

- To show the information in a more interactive way and get a better understanding of it we built an interactive dashboard with Plotly Dash.
- We plotted pie charts showing the total launches by certain selected sites.
- We then plotted a scatter graph showing the relation between Outcome and Payload Mass (Kg) for different booster versions.

Reference: https://github.com/JhonnyJoestarr/Capstone-DS-Project/blob/main/spacex_dash_app.py

Predictive Analysis (Classification)

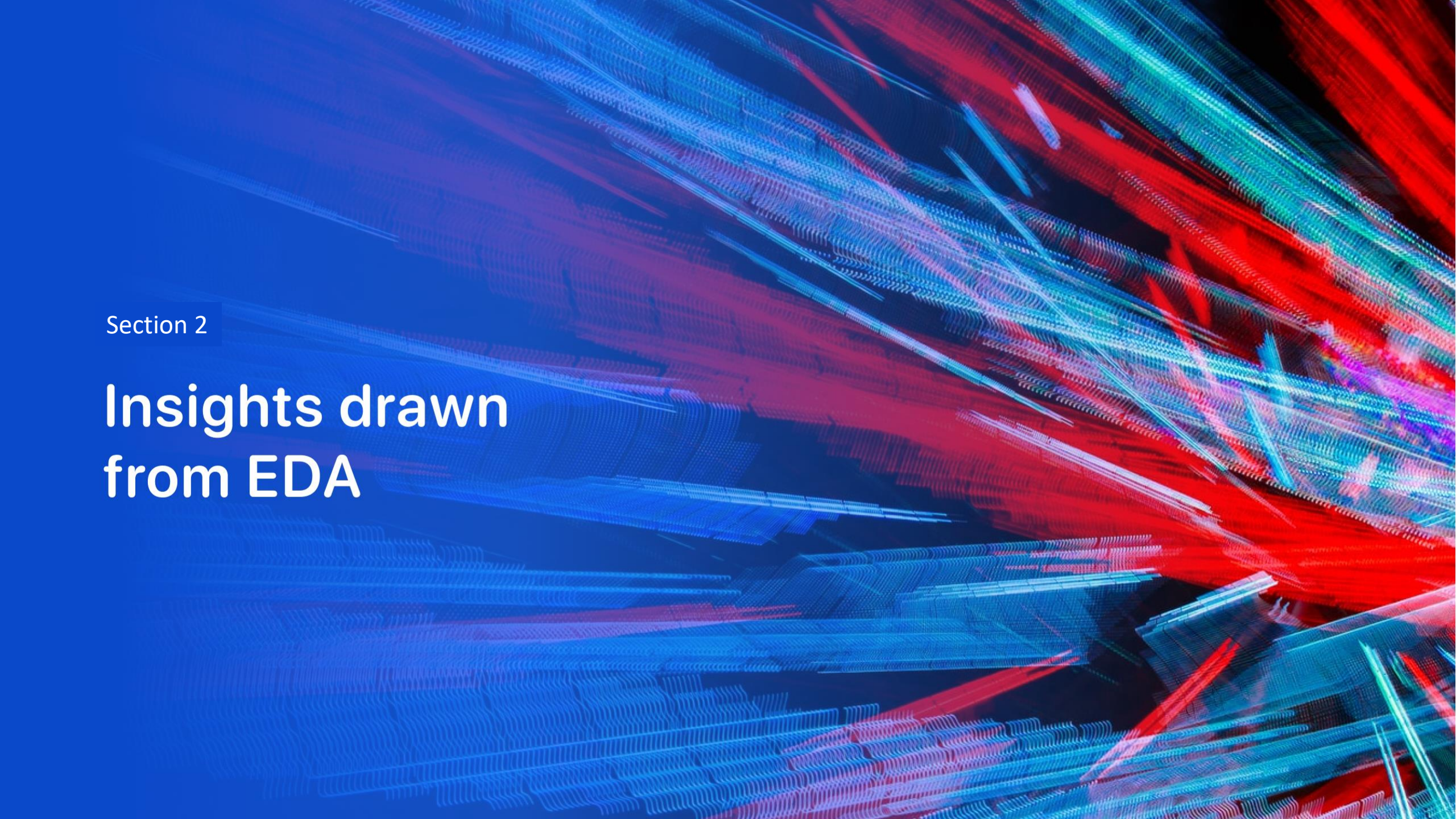


Reference: https://github.com/JhonnyJoestarr/Capstone-DS-Project/blob/main/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

Results

Results will be categorized to three main results:

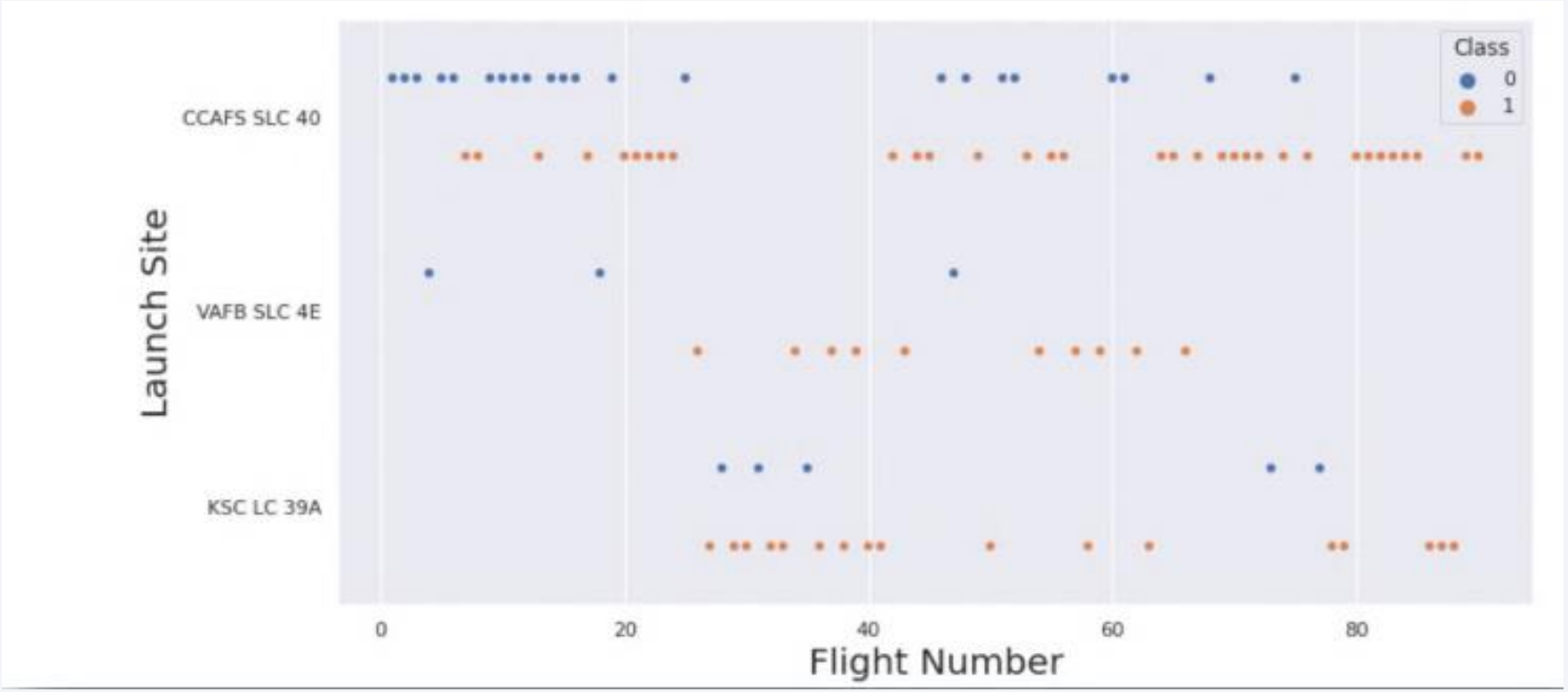
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results


Section 2
Insights drawn from EDA

Flight Number vs. Launch Site

This scatter plot shows that the larger the flights amount of the launch site, the greater the success rate will be.

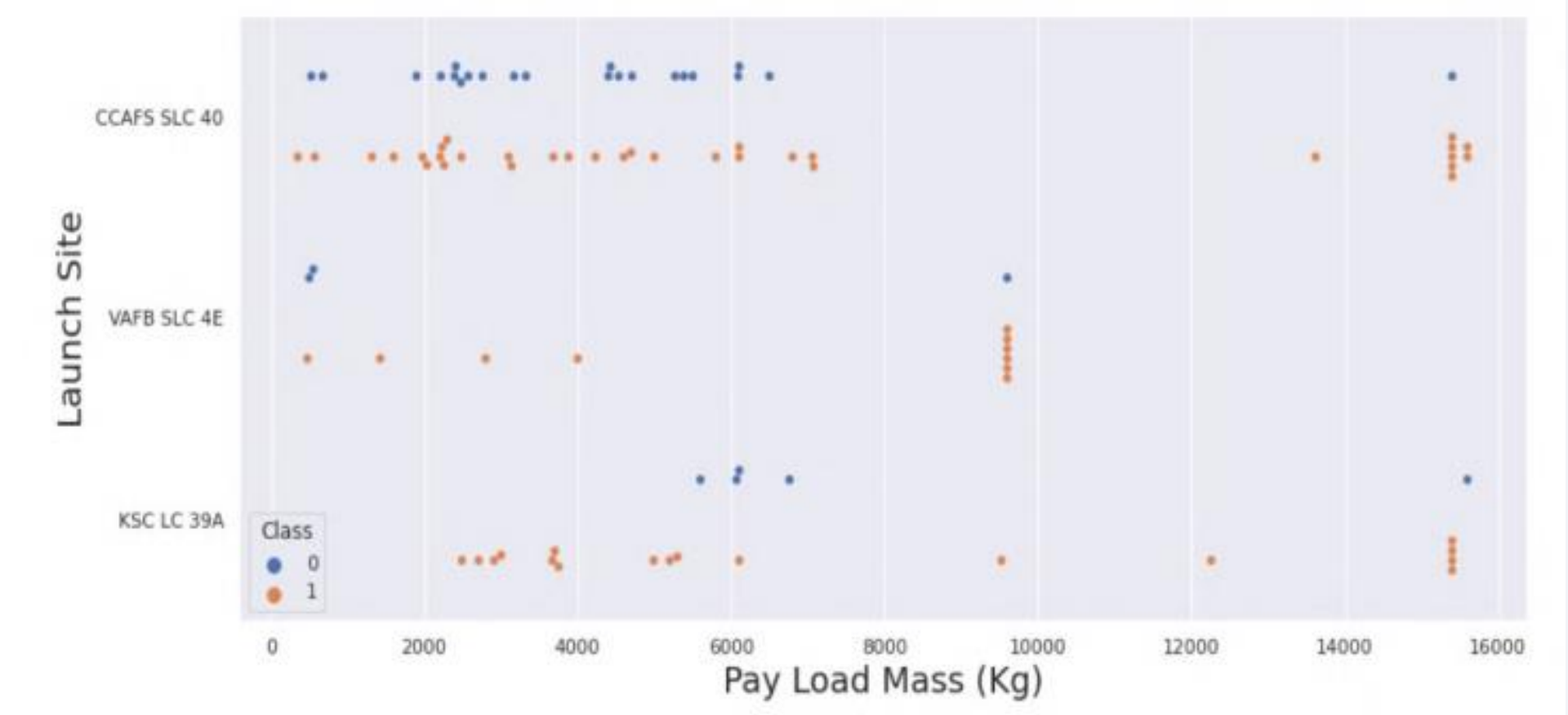
However, site CCAFS SLC40 doesn't really follow this pattern.



Payload vs. Launch Site

This scatter plot shows once the payload mass is greater tan seven tons the success rate highly increases.

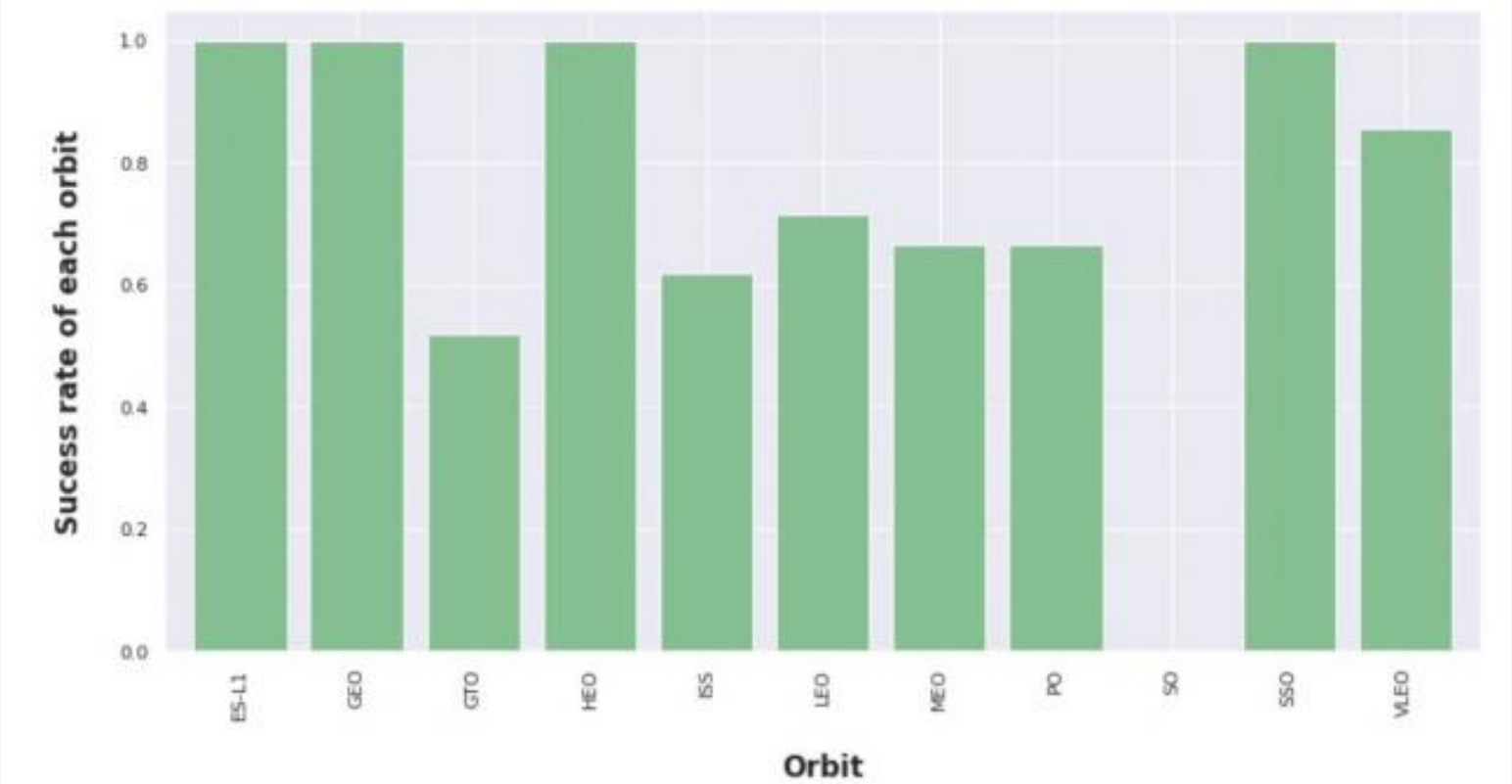
However, there isn't a clear pattern for a relation between payload mass and launch site.



Success Rate vs. Orbit Type

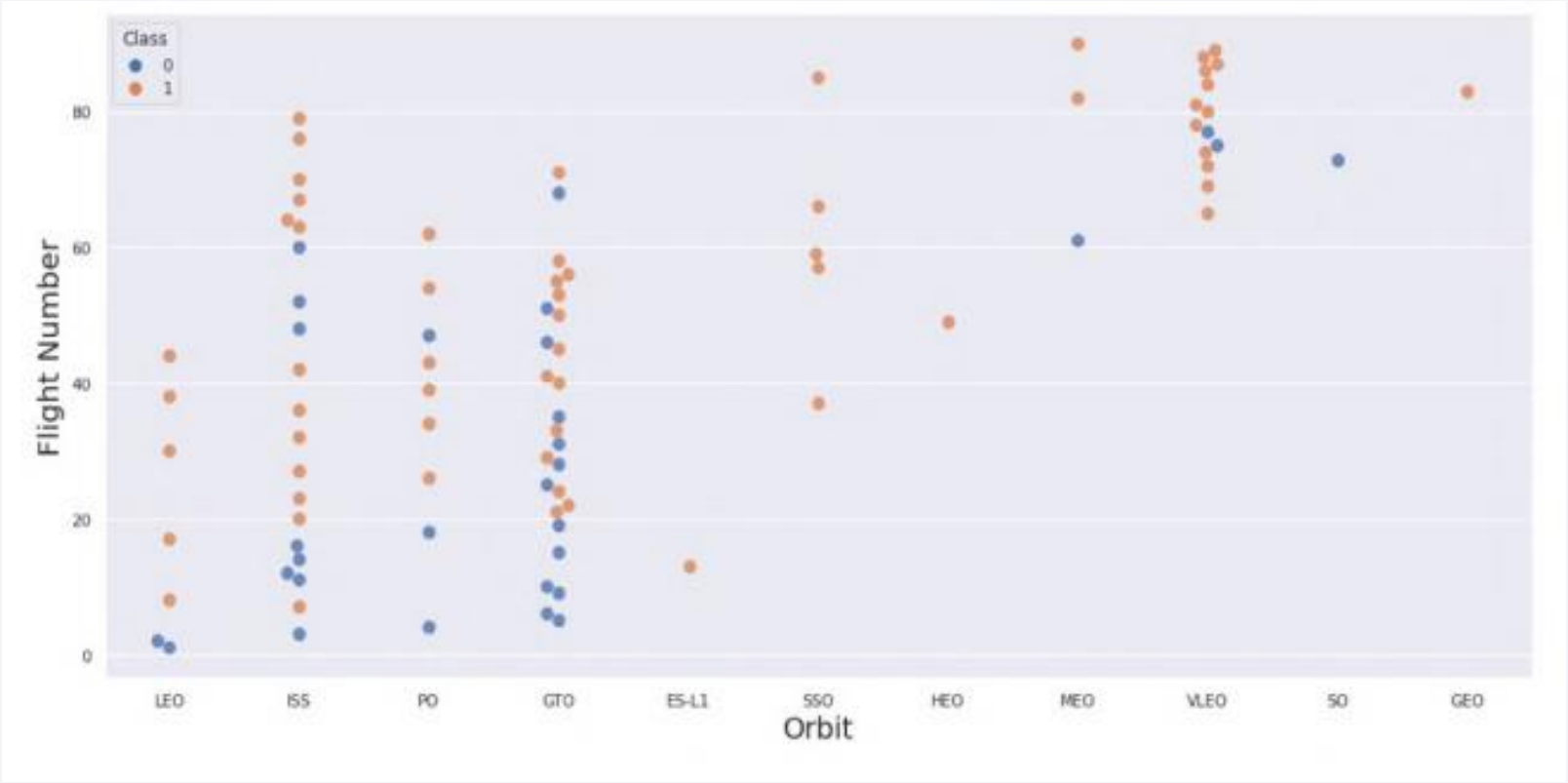
The figure depicted the possibility of the orbits influencing the landing outcomes as some orbits have 100% success rate such as SSO, HEO, GEO and ES-L1 whilo SO orbit has a 0% rate of success.

However, deeper anaylisis show that some of this orbits have only one occurrence such as the mentioned before, which means this dataset needs more data to see a real pattern and we can't draw conclusión for those orbits yet.



Flight Number vs. Orbit Type

- This scatter plot shows that generally, the larger the flight number on each orbit, the greater the success rate except for GTO which doesn't show any relation to other attributes.
- Orbit that only has one occurrence should also be excluded from the above statement as we need more information.

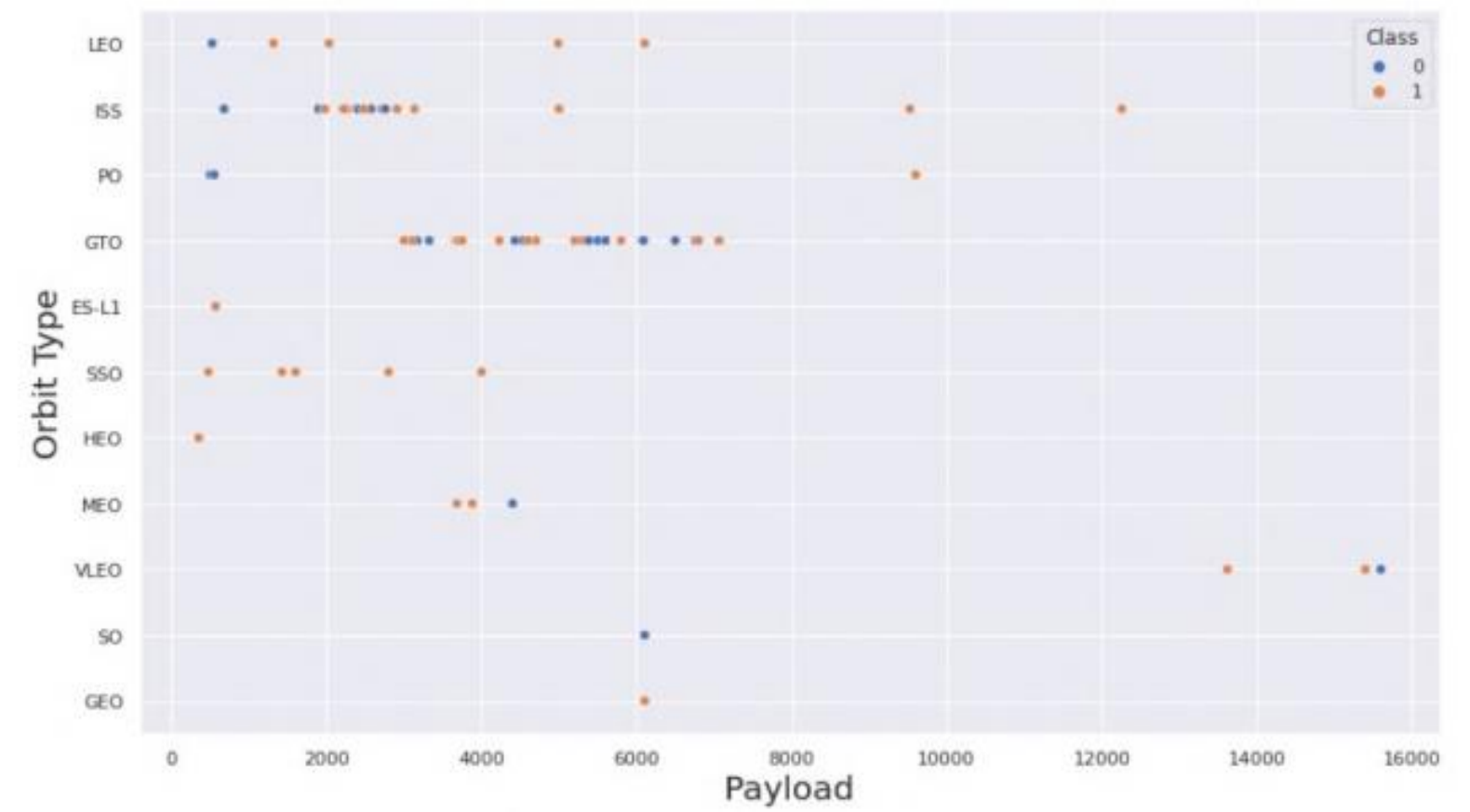


Payload vs. Orbit Type

Heavier payload has positive impact on LEO, ISS and PO orbit. However, it has negative impact on MEO and VLEO orbit.

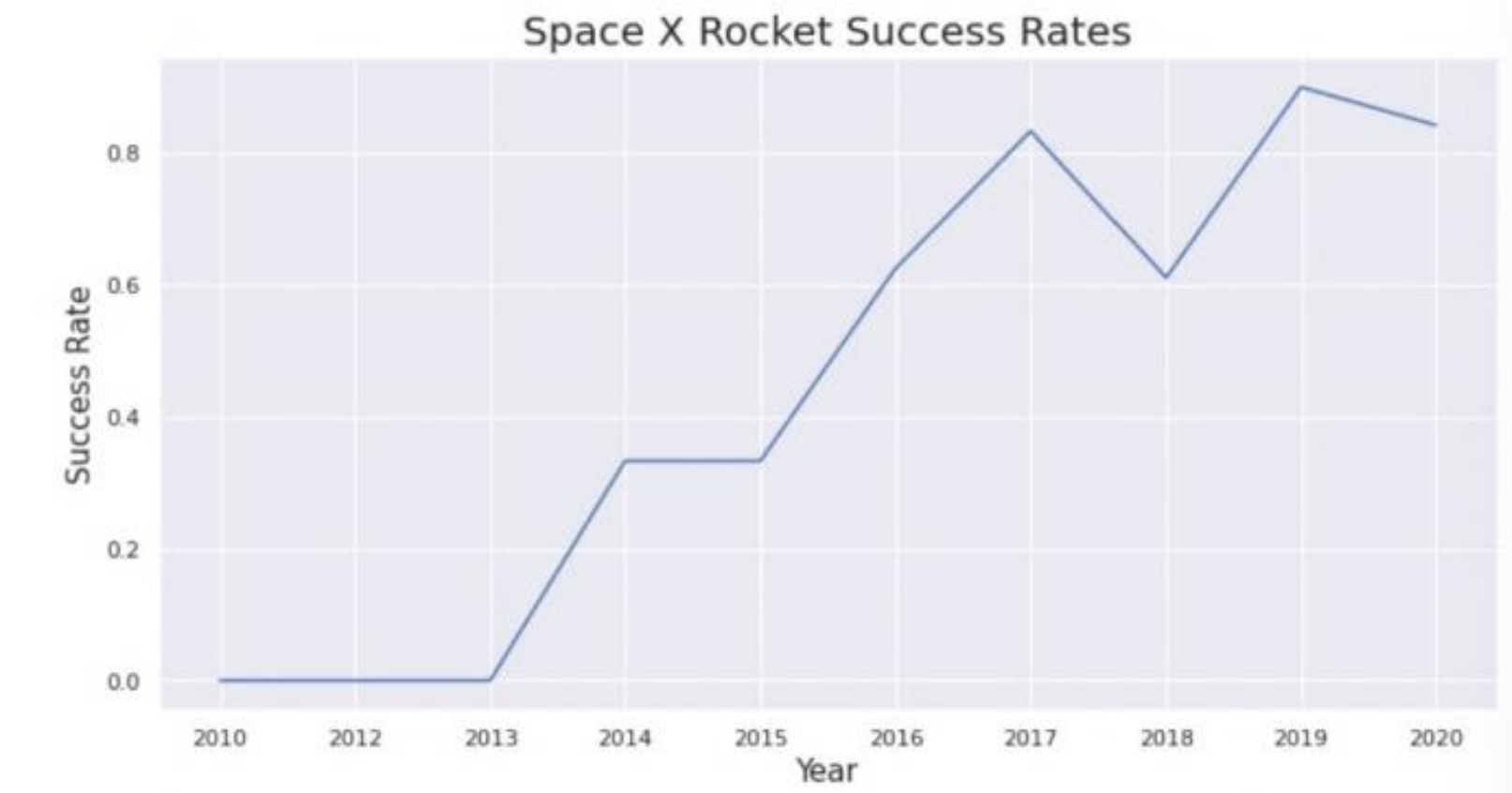
GTO orbit seem to depict no relation between the attributes.

Again, we can't draw any conclusión from orbits with just one occurrence.



Launch Success Yearly Trend

The graph depicts an increasing trend from year 2013 until 2020. If it follows the same pattern, we can expect the success rate to reach near 100% levels in upcoming years.



All Launch Site Names

We used the key word DISTINCT to only show unique values from the Launch Sites column in the table.

In [5]:
```
%sql SELECT DISTINCT LAUNCH_SITE as "Launch_Sites" FROM SPACEX;
```

```
 * ibm_db_sa://zpw86771:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj
sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.
```

Out[5]:

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

Launch Site Names Begin with 'CCA'

We used the query above to display 5 records where launch sites begin with “CCA”.

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

In [11]:
```
task_2 = '''
        SELECT *
        FROM SpaceX
        WHERE LaunchSite LIKE 'CCA%'
        LIMIT 5
        '''
create_pandas_df(task_2, database=conn)
```

Out[11]:

	date	time	boosterversion	launchsite	payload	payloadmasskg	orbit	customer	missionoutcome	landingoutcome
0	2010-04-06	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
1	2010-08-12	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of...	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2	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
3	2012-08-10	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
4	2013-01-03	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

We calculate the total payload carried by boosters from NASA as 45596 using the query below.

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

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) AS "Total Payload Mass by NASA (CRS)
```

```
 * ibm_db_sa://zpw86771:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3
sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.
```

Total Payload Mass by NASA (CRS)
45596

Average Payload Mass by F9 v1.1

We calculated the average payload mass carrier by booster version F9 v1.1 as 2928.4

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

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) AS "Average Payload Mass by Booster
WHERE BOOSTER_VERSION = 'F9 v1.1';
```

```
 * ibm_db_sa://zpw86771:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3
sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.
```

Average Payload Mass by Booster Version F9 v1.1
2928

First Successful Ground Landing Date

We use the MIN() function to find the result. We then observe the first successful landing outcome on ground pad was on December 22nd, 2015.

```
%sql SELECT MIN(DATE) AS "First Succesful Landing Outcome in Ground Pad
WHERE LANDING__OUTCOME = 'Success (ground pad)';
```

```
 * ibm_db_sa://zpw86771:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3
sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.
```

First Succesful Landing Outcome in Ground Pad
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

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

```
%sql SELECT BOOSTER_VERSION FROM SPACEX WHERE LANDING__OUTCOME = 'Success (drone ship)' \
AND PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 6000;
```

```
 * ibm_db_sa://zpw86771:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32731/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

We used wildcard -like- “%” to filter for WHERE Mission Outcome was a success or a failure using the first part of the string.

List the total number of successful and failure mission outcomes

```
%sql SELECT COUNT(MISSION_OUTCOME) AS "Successful Mission" FROM SPACEX WHERE MISSION_OUTCOME LIKE 'Success%';
```

```
 * ibm_db_sa://zpw86771:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.
```

Successful Mission
100

```
%sql SELECT COUNT(MISSION_OUTCOME) AS "Failure Mission" FROM SPACEX WHERE MISSION_OUTCOME LIKE 'Failure%';
```

```
 * ibm_db_sa://zpw86771:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.
```

Failure Mission
1

Boosters Carried Maximum Payload

We determined the booster that have carried the maximum payload using a subquery in the WHERE clause and the MAX() function.

```
%sql SELECT DISTINCT BOOSTER_VERSION AS "Booster Versions which carried the Maximum Payload Mass" FROM SPACEX
WHERE PAYLOAD_MASS__KG_ =(SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEX);
```

```
 * ibm_db_sa://zpw86771:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.clogj3sd0tgtu0lqde00.databases.appdomain.clou
d:32731/bludb
Done.
```

Booster Versions which carried the Maximum Payload Mass
F9 B5 B1048.4
F9 B5 B1048.5
F9 B5 B1049.4
F9 B5 B1049.5
F9 B5 B1049.7
F9 B5 B1051.3
F9 B5 B1051.4
F9 B5 B1051.6
F9 B5 B1056.4
F9 B5 B1058.3
F9 B5 B1060.2
F9 B5 B1060.3

2015 Launch Records

We used a combination of the WHERE clause, LIKE, AND, and BETWEEN conditions to filter for failed landing outcomes in drone ship, their booster versions, and launch site names for year 2015.

```
%sql SELECT BOOSTER_VERSION, LAUNCH_SITE FROM SPACEX WHERE DATE LIKE '2015-%' AND \
LANDING__OUTCOME = 'Failure (drone ship)';
```

```
 * ibm_db_sa://zpw86771:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.clogj3sd0tgtu0lqde00.
databases.appdomain.cloud:32731/bludb
Done.
```

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

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

```
%sql SELECT LANDING__OUTCOME as "Landing Outcome", COUNT(LANDING__OUTCOME) AS "Total Count" FROM SPACEX \
WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' \
GROUP BY  LANDING__OUTCOME \
ORDER BY COUNT(LANDING__OUTCOME) DESC ;
```

```
 * ibm_db_sa://zpw86771:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.
```

Landing Outcome	Total Count
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

We selected Landing Outcomes and the COUNT of landing Outcomes from the data and used the WHERE clause to filter for landing outcomes BETWEEN 2010-06-04 to 2010-03-20.

We applied the GROUP BY clause to group the landing outcomes and the ORDER BY clause to order the grouped landing outcome in descending order.

Section 3

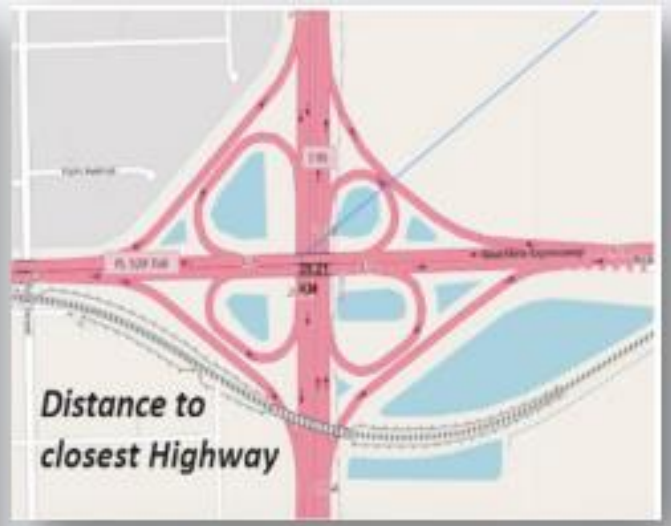
Launch Sites Proximities Analysis

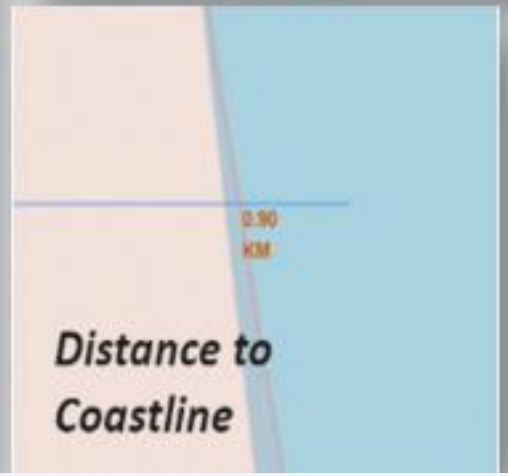
Location of all the Launch Sites

We can see that all the SpaceX launch sites are located inside the United States.

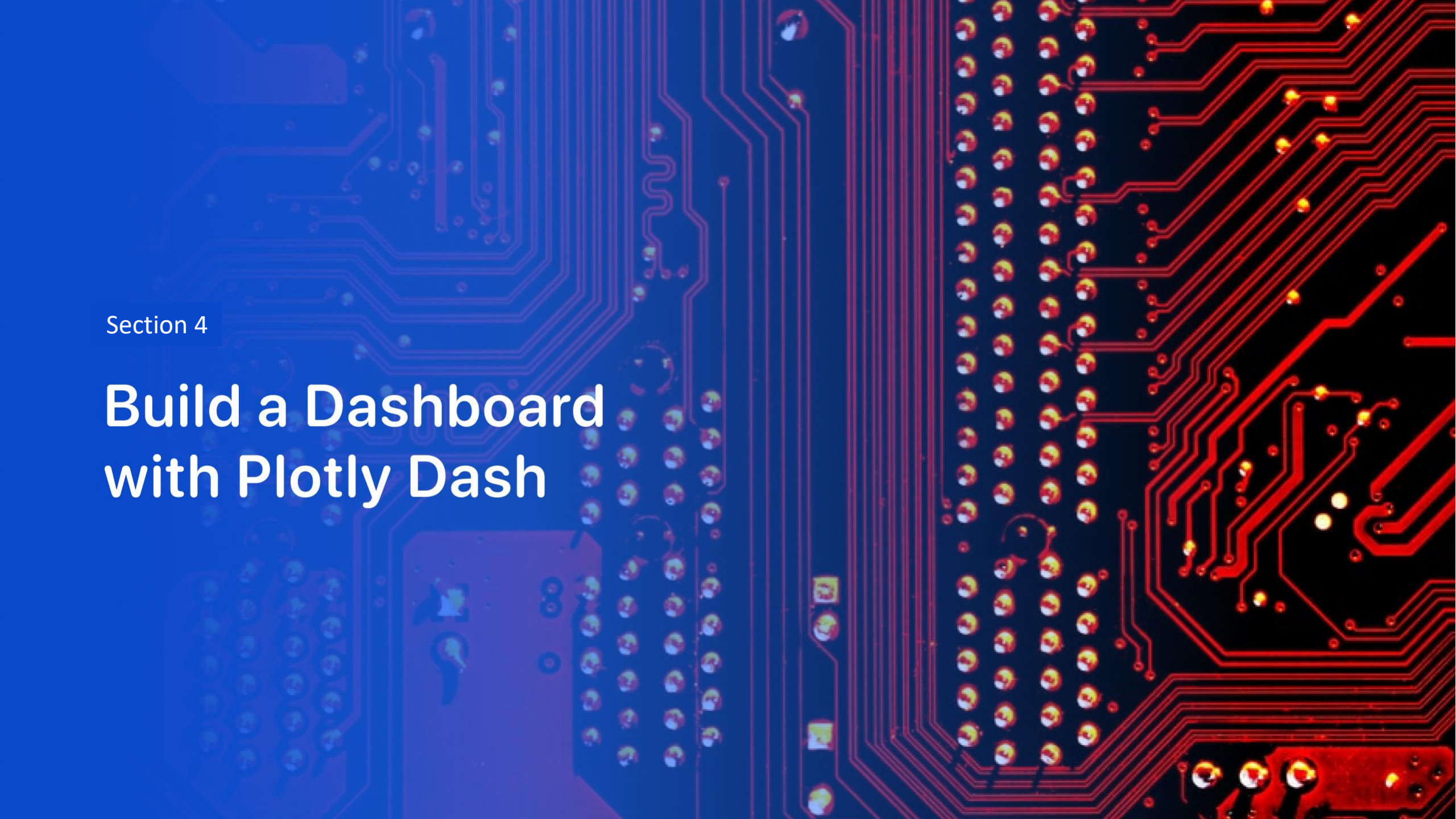
Markers showing launch sites with color labels

Launch Sites Distance to Landmarks

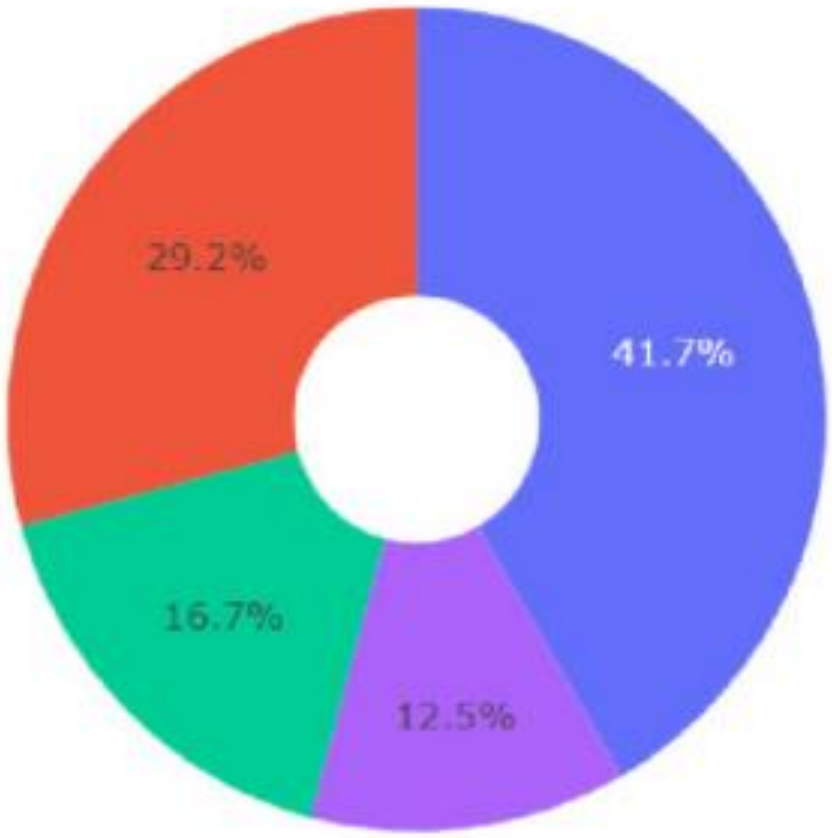




- Are launch sites in close proximity to railways? No
- Are launch sites in close proximity to highways? No
- Are launch sites in close proximity to coastline? Yes
- Do launch sites keep certain distance away from cities? Yes


Section 4
Build a Dashboard with Plotly Dash

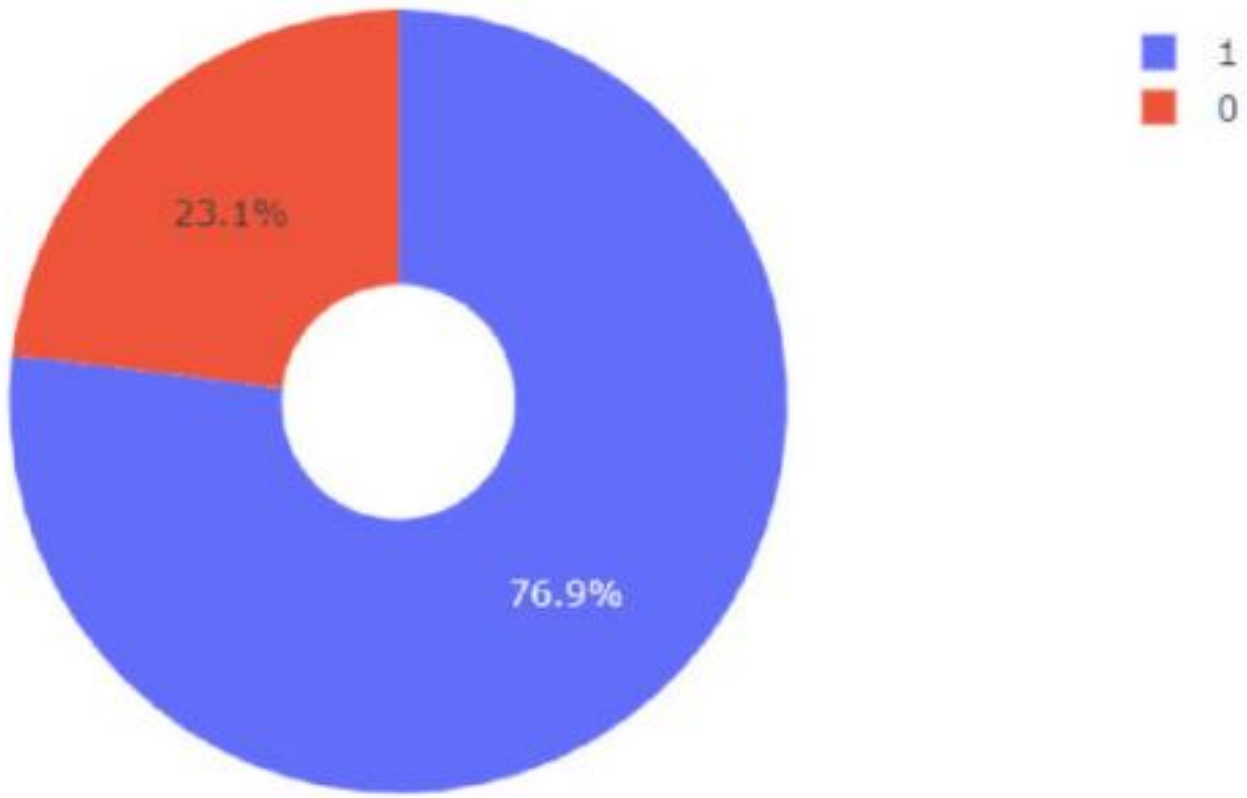
The success percentage by each sites



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

We can see that KSC LC-39A had the most successful launches from all the sites

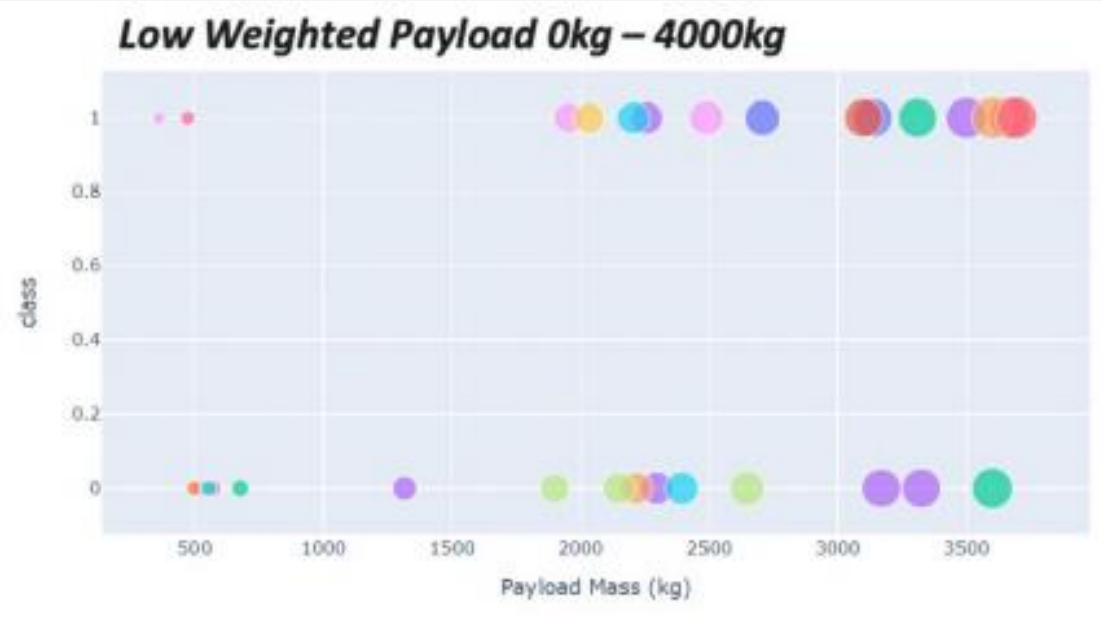
Highest launch-success ratio: KSC LC-39A

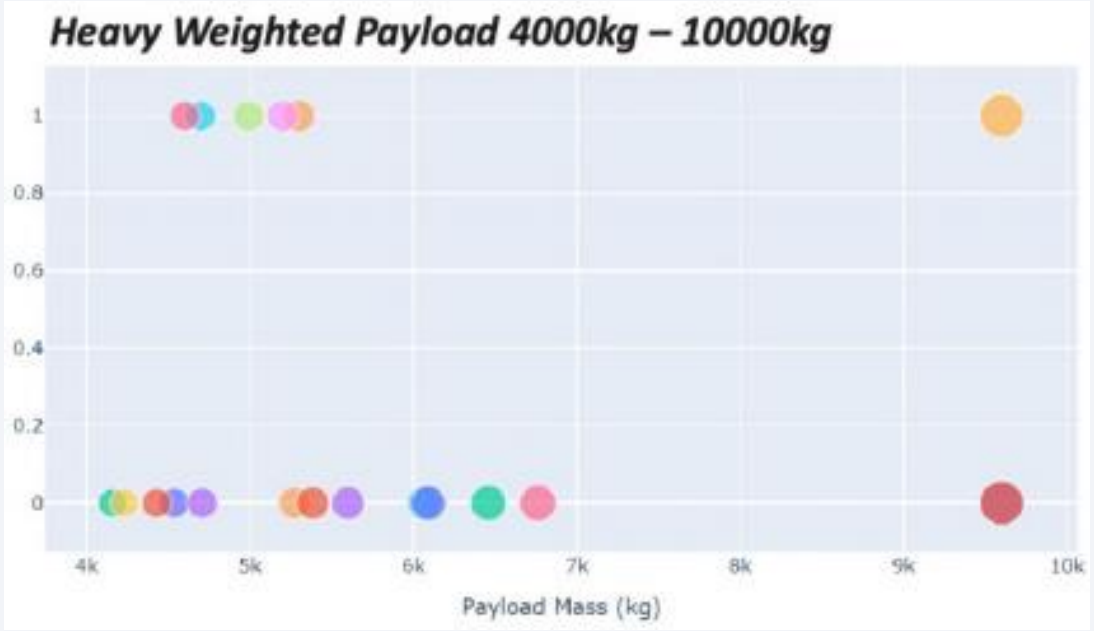


KSC LC-39A achieved a 76.9% success rate while getting a 23.1% failure rate

Payload vs Launch Outcome Scatter Plot

We can see that all the success rate for low weighted payload is higher than heavy weighted payload.





Section 5

Predictive Analysis (Classification)

Classification Accuracy

As we can see, by using the below we couldn't identify any "best" algorithm as all methods had the same accuracy

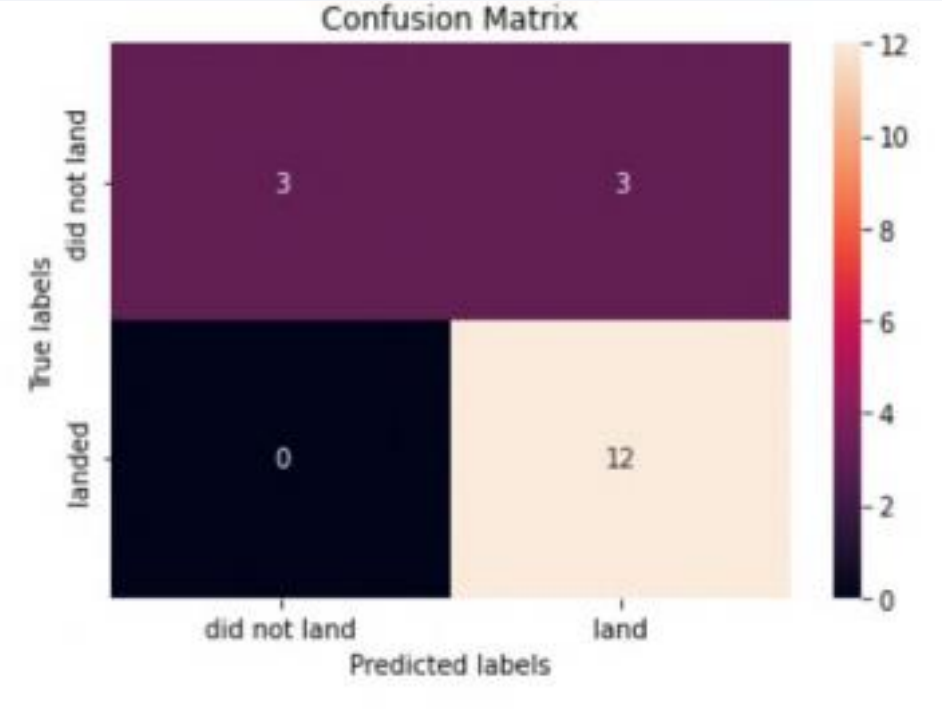
Find the method performs best:

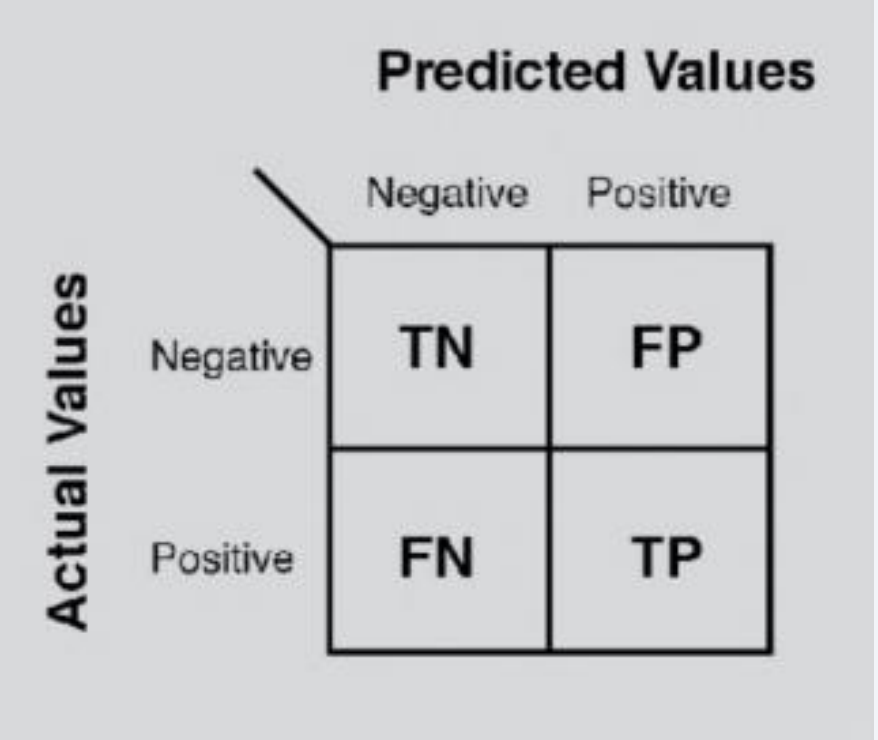
```
]: print(f'LogReg: {logreg_cv.score(X_test,Y_test)}, SVM: {svm_cv.score(X_test,Y_test)}, Tree: {tree_cv.score(X_test,Y_test)},
   print("All methods have the same accuracy, and have the same performance")
```

```
LogReg: 0.8333333333333334, SVM: 0.8333333333333334, Tree: 0.8333333333333334, KNN: 0.8333333333333334
All methods have the same accuracy, and have the same performance
```

Confusion Matrix

The confusion matrix for all models was the same. The major problem is the false positives, unsuccessful landing marked as successful landing by the classifier for example.





Conclusions

We can conclude that:

- All ML models would be a good approach for this dataset
- The low weighted payloads (which define as 4000kg and below) performed better than the heavy weighted´payloads.
- Starting from the year 2013, the success rate for SpaceX launches has increased in a proportional ratio to time in years, this trend may continue until reaching levels near 100%.
- KSC LC-39A has the greatest success rate of all sites with 76.9%
- SSO orbit has the greatest success rate with 100% and more than one occurrence.

Appendix

For first successful landing date query I thought of another method to get it

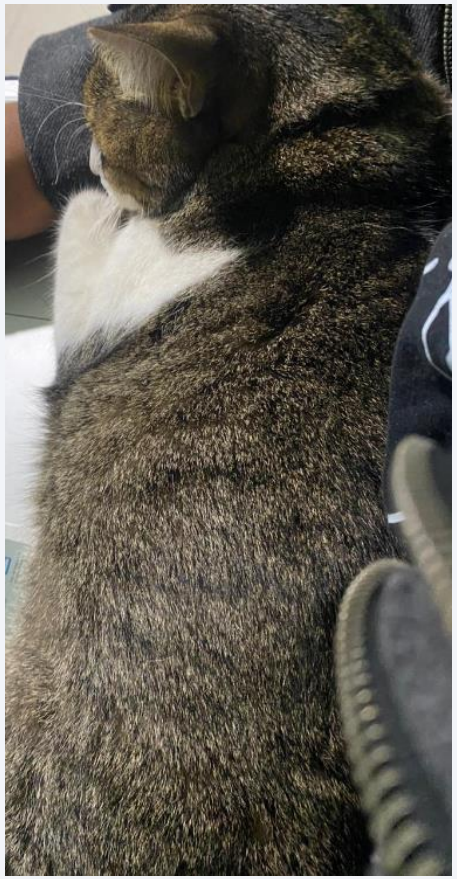
```
%sql SELECT * FROM SPACEXTABLE WHERE Landing_Outcome LIKE "%s (ground pad%" ORDER BY Date ASC LIMIT 1
```

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2015-12-22	1:29:00	F9 FT B1019	CCAFS LC-40	OG2 Mission 2 11 Orbcomm-OG2	2034	LEO	Orbcomm	Success	Success (ground pad)

Thank you!

Final Words



After finishing this journey of months, I feel happy, I'd like to thank UADY for giving me the chance of taking this course, even if I didn't continue studying there, I appreciated this as I got the chance to learn the things that a real Data Scientist does, and this experience only made me sure of the career path I want for myself. I'm still young so I also want to thank my parents for giving the opportunity to study where I want to, even if it is somewhere far from them or from any other family member, I really love them for it and I hope to find a way to give back even more than what they are giving me as of now. I'd like to also thank my friends as they have encouraged me to follow the path I'm taking and helped me in so many ways. Thanks to IBM for preparing such an entertaining and amazing course and thanks to Coursera for having a wonderful platform in which to take it.

Also, thanks to my sister-in-law's cat Todd for being with me during the whole making of this presentation.

I've learnt a lot and I hope to keep learning and one day apply all this knowledge.