

Winning Space Race with Data Science

Hoang Ngoc Huy
07-Apr-2024



Outline

01 Executive Summary

02 Introduction

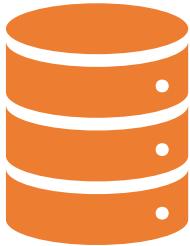
03 Methodology

04 Results

05 Conclusion

06 Appendix

Executive Summary



Summary of methodologies

- Data collection using Web scraping and SpaceX REST API
- Data Wrangling
- Exploratory Data Analysis (EDA) with SQL and Data Visualization
- Interactive visual analytics with Folium
- Machine Learning Prediction

Summary of all results

- It was possible to collect valuable data from public sources;
- EDA allowed us to identify which features are the best to predict the success of launching;
- Machine learning prediction showed the best model to predict which characteristics are important to drive this opportunity in the best way, using all collected data

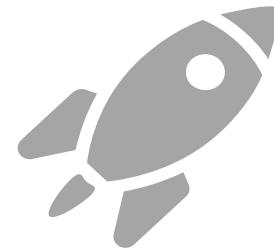
Introduction



Project background and context

SpaceY has objective is to evaluate the viability of competing with SpaceX. The project's goal is to create a machine learning system to predict whether the Falcon 9 first stage will land successfully.

SpaceX advertises Falcon 9 rocket launches on its website for \$62 million; other suppliers cost up to \$165 million each, with most of the savings coming from SpaceX being able to reuse the first stage.



Problems you want to find answers

What factors determine if the rocket will land successfully?

The interaction among various features that determine the success rate of a successful landing

Where is the best place (operating conditions) to ensure a successful launch?

Section 1

Methodology

Methodology

- **Executive Summary**
 - Data collection methodology:
 - Describe how data was collected data from Space X was obtained from 2 sources:
 - SpaceX API (<https://api.spacexdata.com/v4/rockets/>)
 - Web scraping: Wikipedia -
https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches
 - Perform data wrangling
 - Collected data was enriched by creating a landing outcome label based on outcome data after summarizing and analyzing 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
 - Data that was collected until this step were normalized, divided into training and test data sets, and evaluated by four different classification models, being the accuracy of each model evaluated using different combinations of parameters.

Data Collection

- **The data was collected using various methods**
 - Data collection was done using get request to the SpaceX API.
 - Next, we decoded the response content as a J SON using a .json() function call and turned it into a pandas data frame using .json_normalize().
 - We then cleaned the data, checked for missing values, and filled in missing values where necessary.
 - In addition, we performed web scraping from Wikipedia for Falcon 9 launch records with BeautifulSoup.
 - The objective was to extract the launch records as an HTML table, parse the table, and convert it to a pandas data frame for future analysis.

Data Collection – SpaceX API

- We used the get request to the SpaceX API to collect data, clean the requested data, and do some basic data wrangling and formatting
- The link to the GitHub repository is
<https://github.com/hoangngochuy/Applied-Data-Science-Capstone/blob/main/jupyter-labs-spacex-data-collection-api.ipynb>

01

- ❑ Get request rocket launch data using API
 - spacex_url="https://api.spacexdata.com/v4/launches/past"
 - response = requests.get(spacex_url)

02

- ❑ Use json_normalize method to convert the json result into a dataframe
 - static_json_url = '../API_call_spacex_api.json'
 - response.status_code
 - data = pd.json_normalize(response.json())

03

- ❑ Filter the data frame to only include Falcon 9 launches
 - data_falcon9 = df[df['BoosterVersion']!='Falcon 1']
 - data_falcon9.loc[:, 'FlightNumber'] = list(range(1,data_falcon9.shape[0]+1))

04

- ❑ Dealing with Missing Values
 - # Calculate the mean value of PayloadMass column
 - payloadmassavg = data_falcon9['PayloadMass'].mean()
 - # Replace the np.nan values with its mean value
 - data_falcon9['PayloadMass'].replace(np.nan,payloadmassavg,inplace=True)

Data Collection - Scraping

- We applied web scrapping to web scrap Falcon 9 launch records with BeautifulSoup
- We parsed the table and converted it into a pandas data frame
- The link to the GitHub repository is <https://github.com/hoangngochuy/Applied-Data-Science-Capstone/blob/main/jupyter-labs-webscraping.ipynb>

01

❑ Request the Falcon9 Launch Wiki page from its URL

- data = requests.get(static_url)
- html = data.content
- soup = BeautifulSoup(html, "html.parser")
- print(soup.title)

02

❑ Extract all column/variable names from the HTML table header

- html_tables = soup.find_all('table')
- first_launch_table = html_table[2]
- 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)

03

❑ Create a data frame by parsing the launch HTML tables

We will create an empty dictionary with keys from the extracted column names in the previous task. Later, this dictionary will be converted into a Pandas dataframe

- launch_dict = dict.fromkeys(column_names)
- Fill up the launch_dict with launch records extracted from tables rows
- df = pd.DataFrame(launch_dict)

04

❑ Export data to csv

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

Data Wrangling

We performed Exploratory Data Analysis (EDA)

We calculated the number of launches per site; calculated the number and occurrence of each orbit

We calculated the number and occurrence of mission outcomes of the orbits

We created a landing outcome label from the outcome column and exported the results to csv file.

The link to the GitHub repository is

<https://github.com/hoangngochuy/Applied-Data-Science-Capstone/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb>

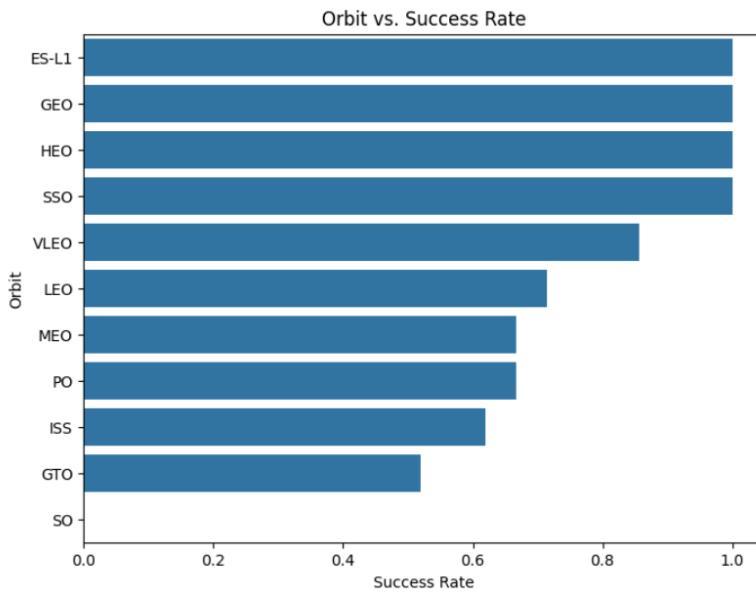
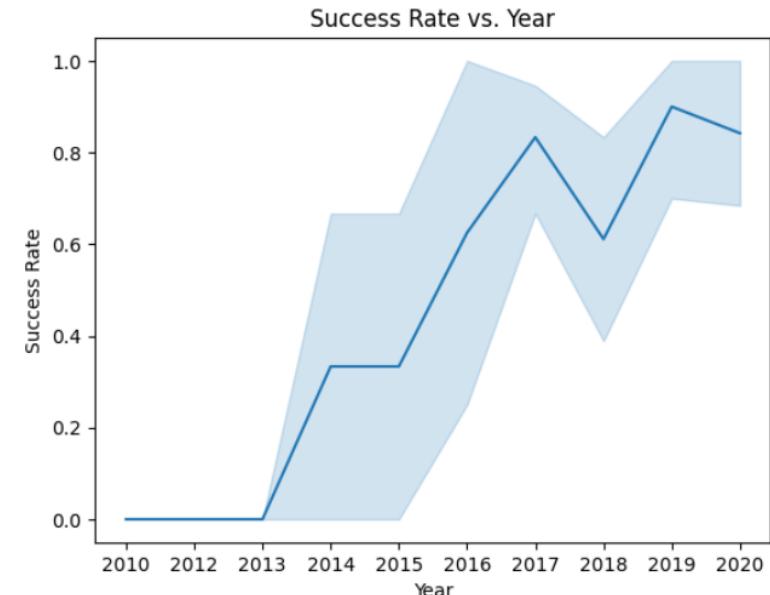
1. EDA

2. Summarizations

3. Creation of landing outcome label

EDA with Data Visualization

- To explore data, scatterplots, and bar plots were used to visualize the relationship between the pair of features and the success rate of each orbit type, flight number and orbit type, and the launch success yearly trend.
- The link to the GitHub repository is
<https://github.com/hoangngochuy/Applied-Data-Science-Capstone/blob/main/EDA%20with%20Visualization%20Lab.ipynb>



EDA with SQL

- Load the SpaceX dataset into a PostgreSQL database
- Apply EDA with SQL to get insight from the data. The following SQL queries to find out:
 - The names of unique launch sites in the space mission.
 - Top 5 launch sites whose name begins with the string 'CCA';
 - Total payload mass carried by boosters launched by NASA (CRS);
 - Average payload mass carried by booster version F9 v1.1;
 - Date when the first successful landing outcome in the ground pad was achieved;
 - Names of the boosters that have success in drone ships and have payload mass between 4000 and 6000 kg;
 - Total number of successful and failed mission outcomes;
 - Names of the booster versions which have carried the maximum payload mass;
 - Failed landing outcomes in drone ships, their booster versions, and launch site names for the year 2015;
 - Rank of the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20.
- The link to the GitHub repository is https://github.com/hoangngochuy/Applied-Data-Science-Capstone/blob/main/jupyter-labs-eda-sql-coursera_sqlite.ipynb

Build an Interactive Map with Folium



Marked all launch sites on a map, and added map objects such as markers, circles, and lines to mark the success or failure of launches for each site on the folium map.



Assigned the feature launch outcomes (failure or success) to classes 0 and 1 for each site on the map



Using the color-labeled marker clusters, we identified which launch sites have relatively high success rates.



It calculated the distances between a launch site and its proximities.



The link to the GitHub repository is https://github.com/hoangngochuy/Applied-Data-Science-Capstone/blob/main/lab_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

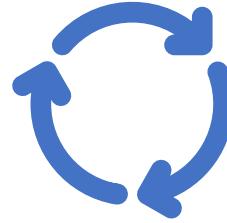
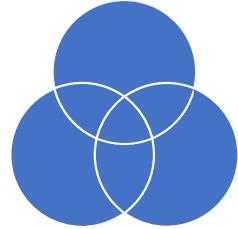
An interactive dashboard with Plotly Dash was used to visualize data

- A pie chart shows Total Success Launches by certain sites
- A scatter graph shows the Correlation between Payload and Success for the different booster version

The link to the GitHub repository is https://github.com/hoangngochuy/Applied-Data-Science-Capstone/blob/main/spacex_dash_app.py



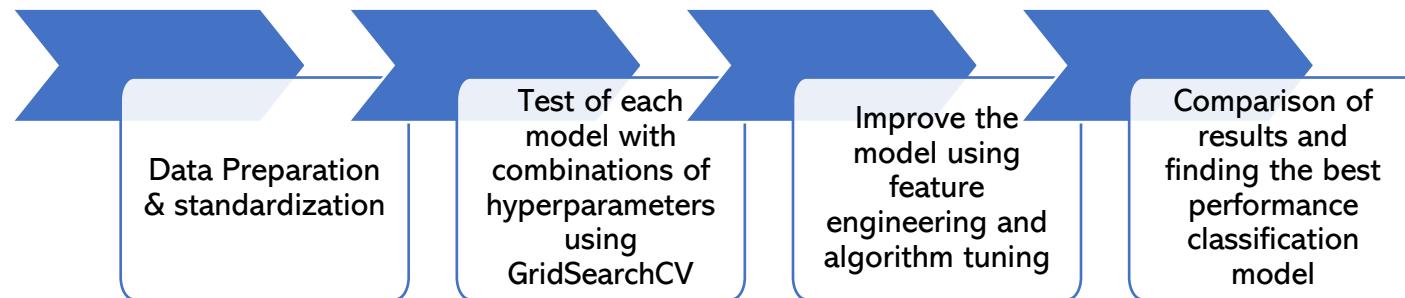
Predictive Analysis (Classification)



Comparison of four classification models:

- Logistic Regression
- Support Vector Machine (SVM)
- Decision Tree
- K Nearest Neighbors (KNN)

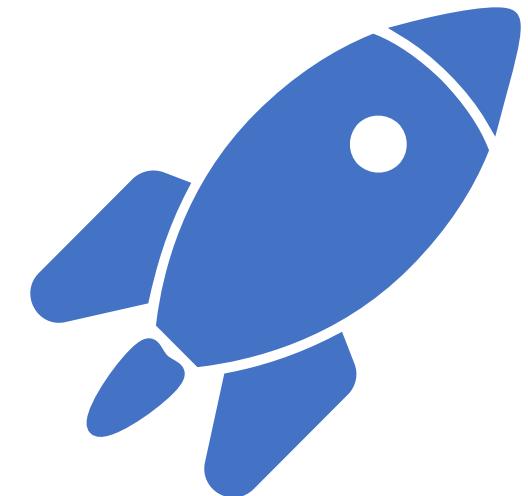
Model development process:

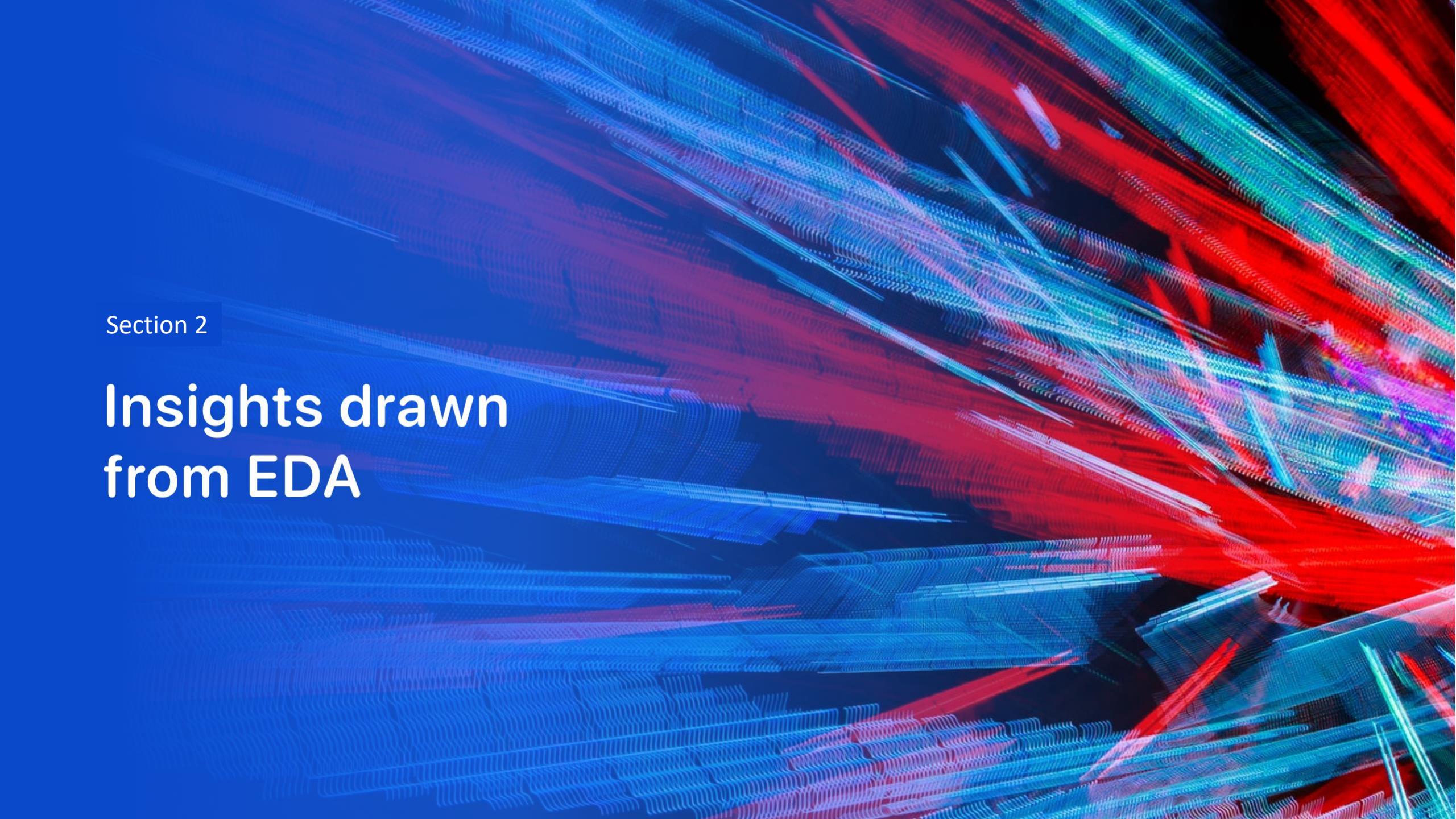


The link to the GitHub repository is https://github.com/hoangngochuy/Applied-Data-Science-Capstone/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results

- **Exploratory data analysis results:**
 - Space X uses 4 different launch sites;
 - The average payload of F9 v1.1 booster is 2,928 kg;
 - The first successful landing outcome happened in 2015 five years after the first launch;
 - Many Falcon 9 booster versions were successful at landing in drone ships having payloads above the average;
 - Almost 100% of mission outcomes were successful;
 - Two booster versions failed at landing in drone ships in 2015: F9 v1.1 B1012 and F9 v1.1 B1015;
 - The number of landing outcomes became as better as the years passed.



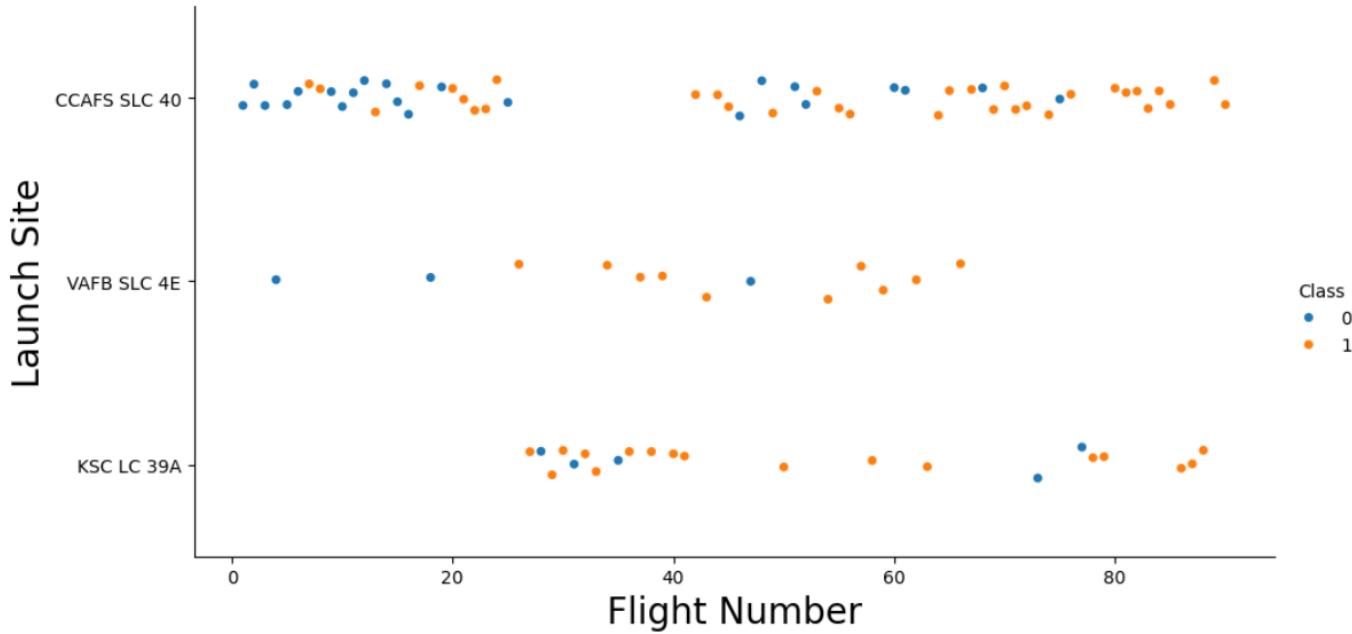
The background of the slide features a complex, abstract digital visualization. It consists of numerous thin, glowing lines that create a sense of depth and motion. The lines are primarily blue and red, with some green and purple highlights. They form a grid-like structure that curves and twists across the frame, resembling a three-dimensional space or a network of data points. The overall effect is futuristic and dynamic.

Section 2

Insights drawn from EDA



Flight Number vs. Launch Site

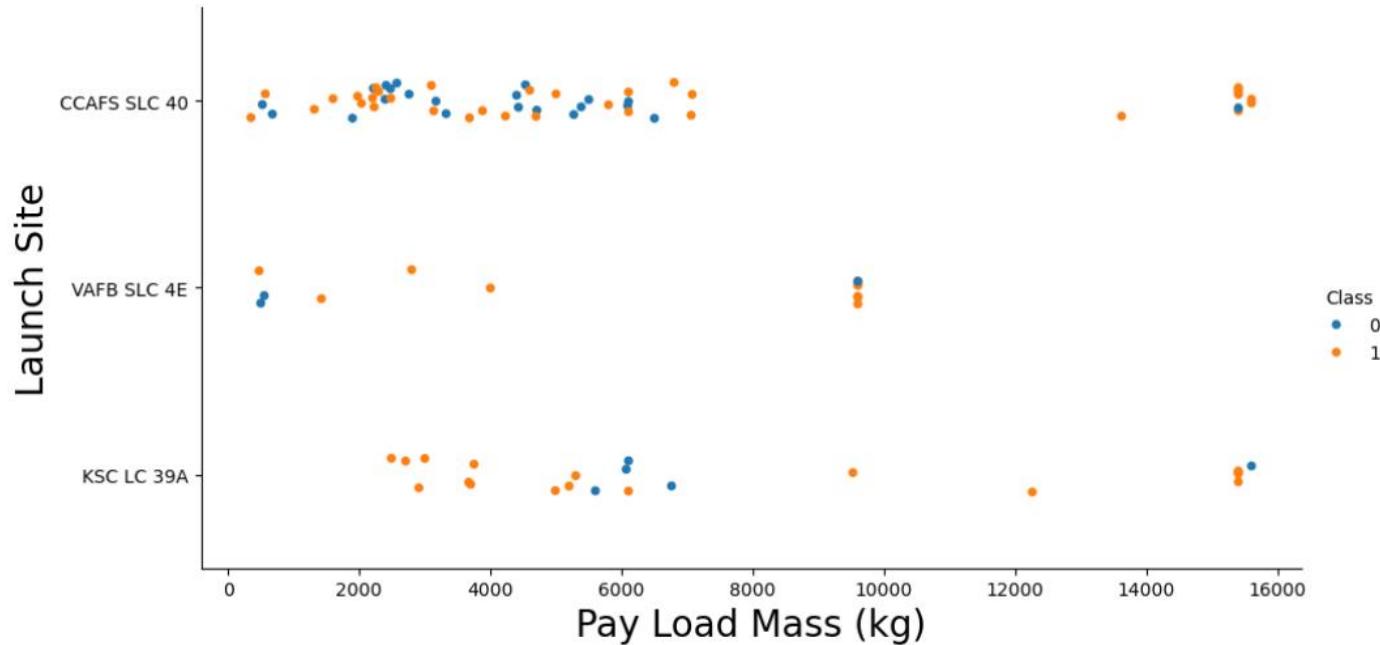


Now try to explain the patterns you found in the Flight Number vs. Launch Site scatter point plots.

- We found the larger the flight amount the greater the success rate for each launch site
- CCAFS SLC 40 is the best launch site
- The success rate improved over time



Payload vs. Launch Site

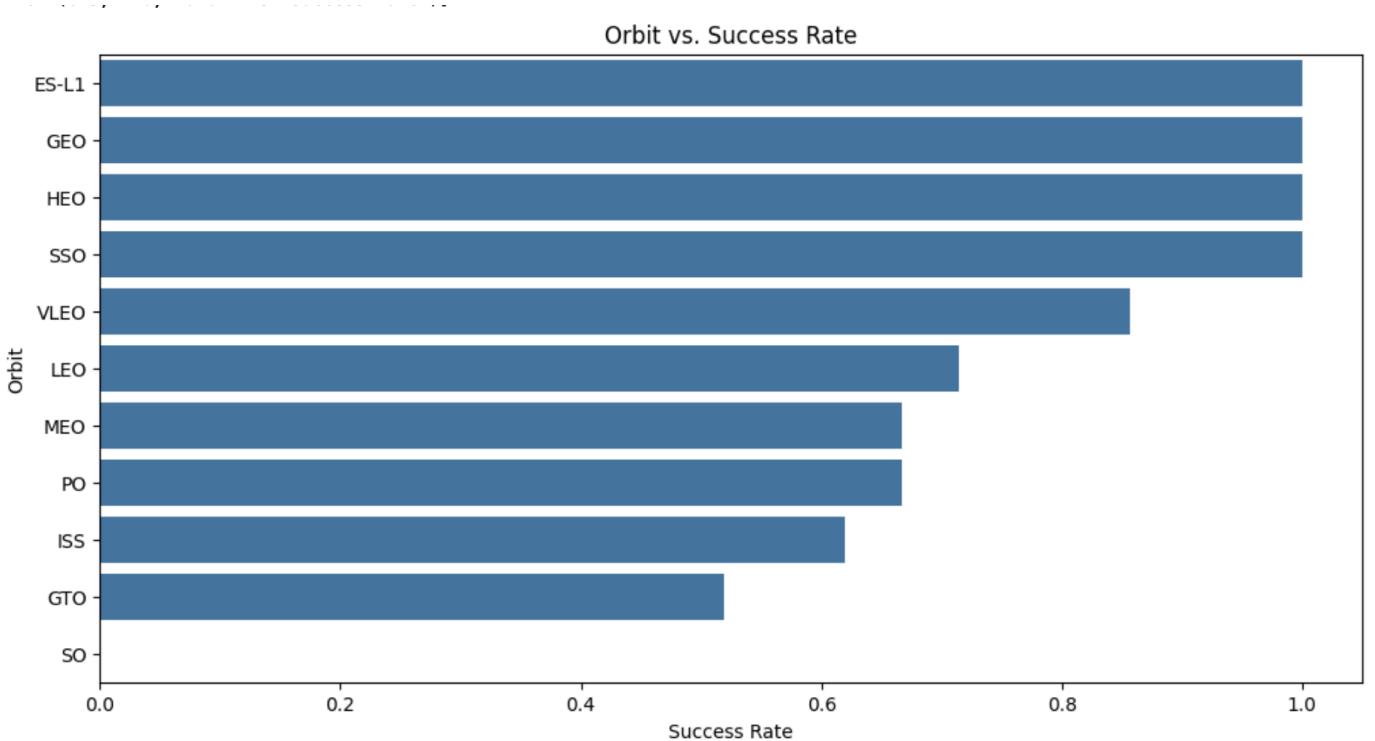


Now if you observe Payload Vs. Launch Site scatter point chart you will find for the VAFB-SLC launchsite there are no rockets launched for heavy payload mass(greater than 10000).

* The greater the payload mass for launch site CCAFS SLC 40 the higher the success rate for the rockets



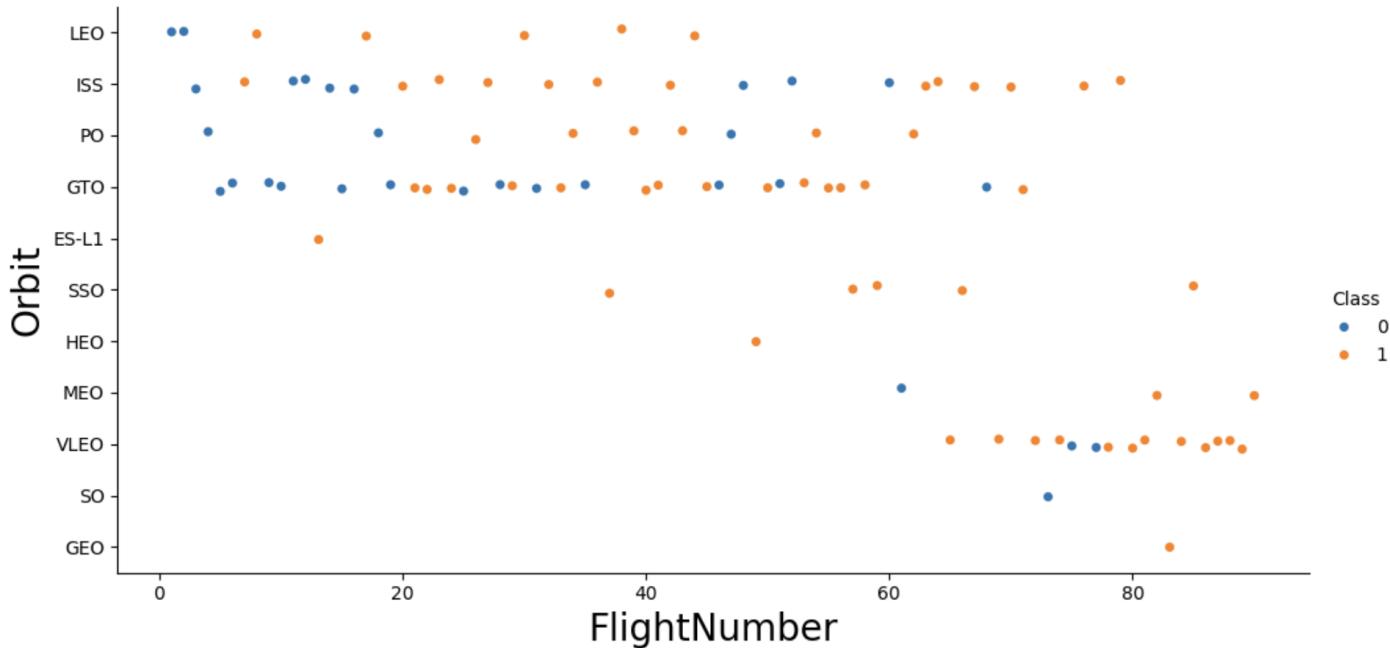
Success Rate vs. Orbit Type



- Analyze the plotted bar chart try to find which orbits have high sucess rate.
- From the plot, we can see that ES-L1, GEO, HEO, SSO had the most success rate (100%) and next VLEO is above 80%



Flight Number vs. Orbit Type

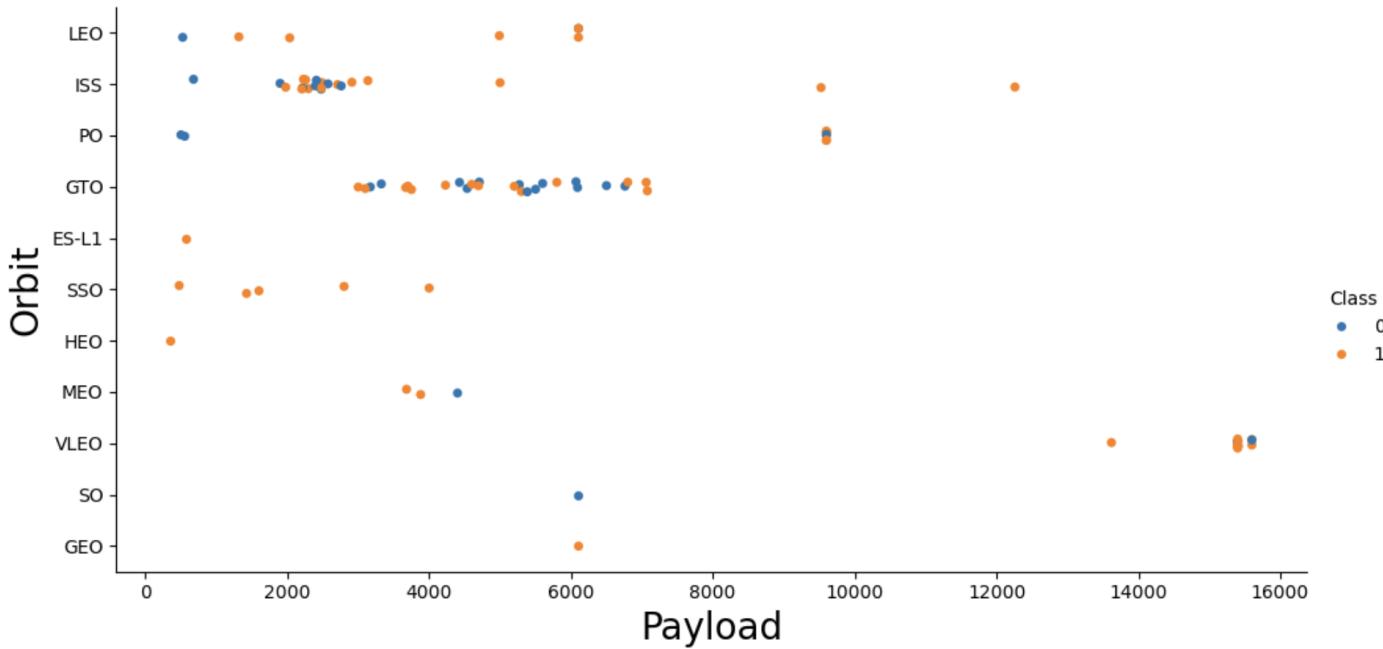


You should see that in the LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.

- The success rate improved over time to all orbits;
- VLEO orbit seems a new business opportunity, due to the recent increase in its frequency.



Payload vs. Orbit Type



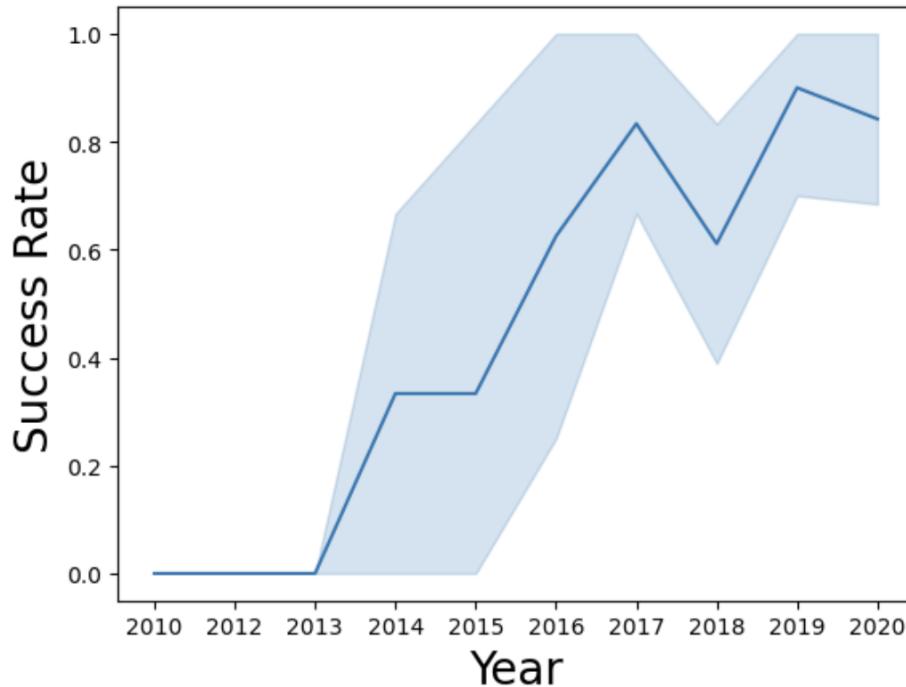
With heavy payloads the successful landing or positive landing rate are more for Polar, LEO and ISS.

However for GTO we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission) are both there here.

- There is no relation between payload and success rate to orbit GTO;
- ISS orbit has the widest range of payload and a good rate of success;
- There are few launches to the orbits SO and GEO.



Launch Success Yearly Trend



you can observe that the sucess rate since 2013 kept increasing till 2020

- The success rate started increasing in 2013 and continued until 2020;
- It seems that the first three years were a period of adjustments and improvement of technology

All Launch Site Names

There are four unique launch sites by selecting unique occurrences of “LAUNCH_SITE” values from the SpaceX data:

Launch_Site

	Launch_Site
0	CCAFS LC-40
1	VAFB SLC-4E
2	KSC LC-39A
3	CCAFS SLC-40

Display the names of the unique launch sites in the space mission

```
[8]: %sql SELECT Distinct LAUNCH_SITE FROM SPACEXTBL  
* sqlite:///my_data1.db  
Done.  
[8]: Launch_Site  
CCAFS LC-40  
VAFB SLC-4E  
KSC LC-39A  
CCAFS SLC-40
```

Launch Site Names Begin with 'CCA'

We used the query to display 5 records where launch sites begin with 'CCA' and limit 5. The first five samples of Cape Canaveral launches

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

```
[9]: %sql SELECT * FROM SPACEXTBL WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5  
* sqlite:///my_data1.db  
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	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	0:35:00	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

Total Payload Mass

The total payload carried by boosters from NASA: **45,596 KG .**

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

```
[10]: %sql SELECT SUM(PAYLOAD__MASS__KG_) FROM SPACEXTBL WHERE CUSTOMER='NASA (CRS)'  
* sqlite:///my_data1.db  
Done.  
[10]: SUM(PAYLOAD__MASS__KG_)  
45596
```

(*) Total payload calculated above, by summing all payloads whose codes contain 'CRS', which corresponds to NASA.

Average Payload Mass by F9 v1.1

Average payload mass carried by booster
version F9 v1.1: **2,982.4 KG .**

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
```

(*) By filtering data by the booster version
above and calculating the average payload
mass we obtained the value of 2,928.4 kg.

First Successful Ground Landing Date

The first successful landing outcome on the ground pad: **2015-12-22**

List the date when the first succesful landing outcome in ground pad was acheived.

Hint: Use min function

```
[14]: %sql SELECT min(DATE) FROM SPACEXTBL WHERE LANDING_OUTCOME='Success (ground pad)'  
* sqlite:///my_data1.db  
Done.  
[14]: min(DATE)  
2015-12-22
```

(*) By filtering data by successful landing outcome on the ground pad and getting the minimum value for the date it's possible to identify the first occurrence, which happened on December 22, 2015

Successful Drone Ship Landing with Payload between 4000 and 6000

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

```
[15]: %sql SELECT BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ between 4000 and 6000 AND LANDING_OUTCOME='Success (drone ship)'
```

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

```
Done.
```

```
[15]: Booster_Version
```

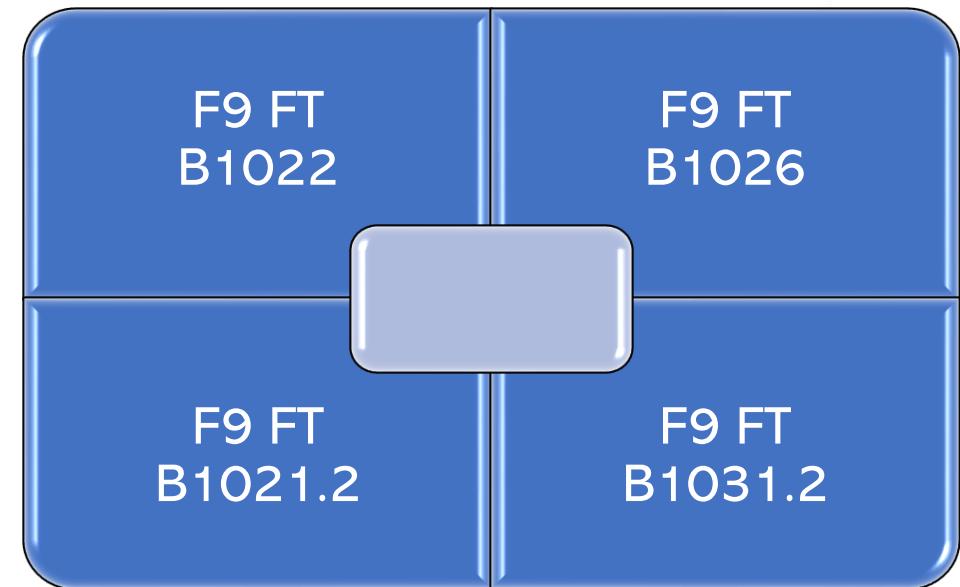
```
F9 FT B1022
```

```
F9 FT B1026
```

```
F9 FT B1021.2
```

```
F9 FT B1031.2
```

Selecting distinct booster versions according to the filters above, these 4 are the result:



Total Number of Successful and Failure Mission Outcomes

- We used wildcards like '%' to filter for WHERE Mission Outcome was a success or a failure.

List the total number of successful and failure mission outcomes

```
[20]: %sql SELECT MISSION_OUTCOME, COUNT(*) \
FROM SPACEXTBL \
WHERE MISSION_OUTCOME LIKE '%Success%' OR MISSION_OUTCOME LIKE '%Failure%' \
GROUP BY MISSION_OUTCOME
```

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

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

- Grouping mission outcomes and counting records for each group led us to the summary:

Misson Outcome	Total Number
Success	100
Failure	1
Total	101

Boosters Carried Maximum Payload

- Boosters which have carried the maximum payload mass

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

```
[17]: %sql SELECT BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS_KG_ = (SELECT MAX(PAYLOAD_MASS_KG_) FROM SPACEXTBL)  
* sqlite:///my_data1.db  
Done.
```

```
[17]: 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
```

These are the boosters that have carried the maximum payload mass registered in the dataset

2015 Launch Records

Failed landing outcomes in drone ships, their booster versions, and launch site names for in year 2015

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.

Note: SQLite does not support monthnames. So you need to use substr(Date, 6,2) as month to get the months and substr(Date,0,5)='2015' for year.

```
[18]: %sql SELECT SUBSTR("DATE",6,2) AS MONTH_NAME, \
    LANDING_OUTCOME AS LANDING_OUTCOME, \
    BOOSTER_VERSION AS BOOSTER_VERSION, \
    LAUNCH_SITE AS LAUNCH_SITE \
    FROM SPACEXTBL WHERE LANDING_OUTCOME = 'Failure_(drone_ship)' AND SUBSTR("DATE",0,5) = '2015'
```

* sqlite:///my_data1.db

Done.

```
[18]:
```

MONTH_NAME	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 list has only two occurrences:

Booster Ver.	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

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 LANDING_OUTCOME, COUNT(*) AS COUNT_LAUNCHES FROM SPACEXTBL \
WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' \
GROUP BY LANDING_OUTCOME \
ORDER BY COUNT_LAUNCHES DESC;
```

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

Landing_Outcome	COUNT_LAUNCHES
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1



Ranking of all landing outcomes between the dates 2010-06-04 and 2017-03-20



This view of data alerts us that “No attempt” must be considered.

The background of the slide is a photograph taken from space at night. It shows the curvature of the Earth's horizon against a dark blue sky. Numerous glowing yellow and white points represent city lights, concentrated in coastal and urban areas. In the upper right quadrant, there are bright green and yellow bands of light, likely the Aurora Borealis or Australis. The overall atmosphere is dark and mysterious.

Section 3

Launch Sites Proximities Analysis

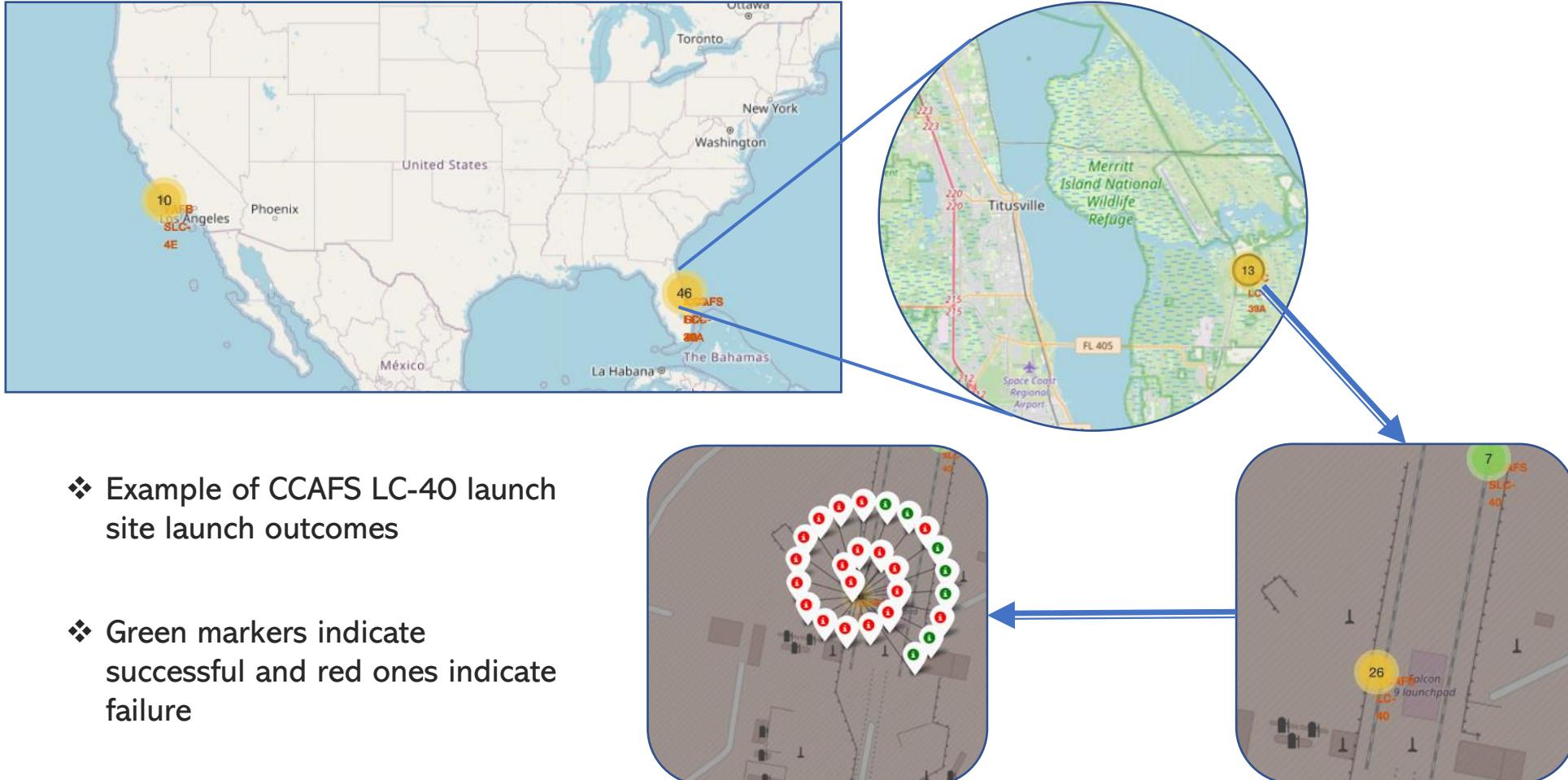


Mark all launch sites on a map

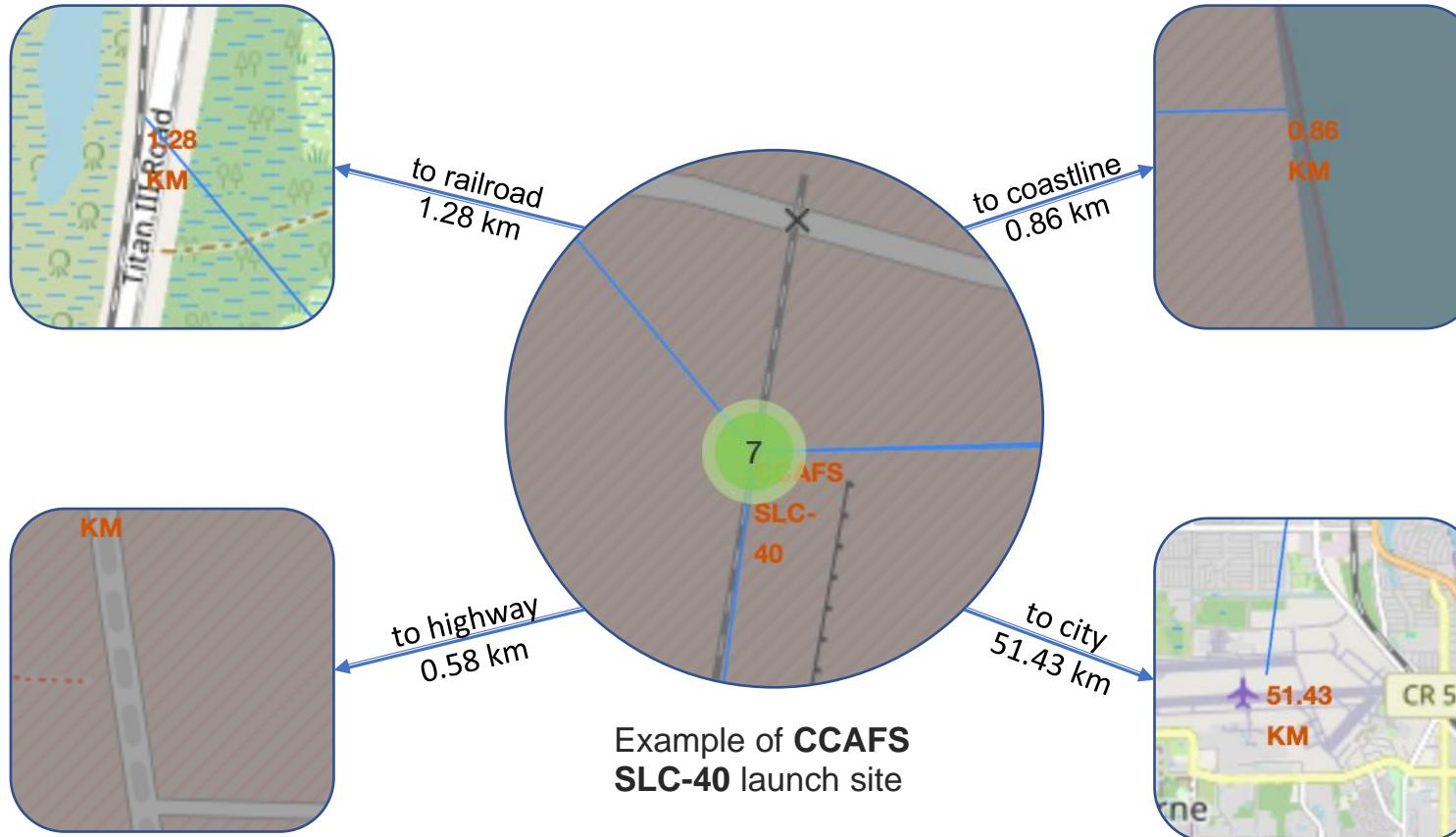
* Launch sites are near the sea, probably for safety, but not too far from roads and railroads.



Mark the success/failed launches for each site on the map

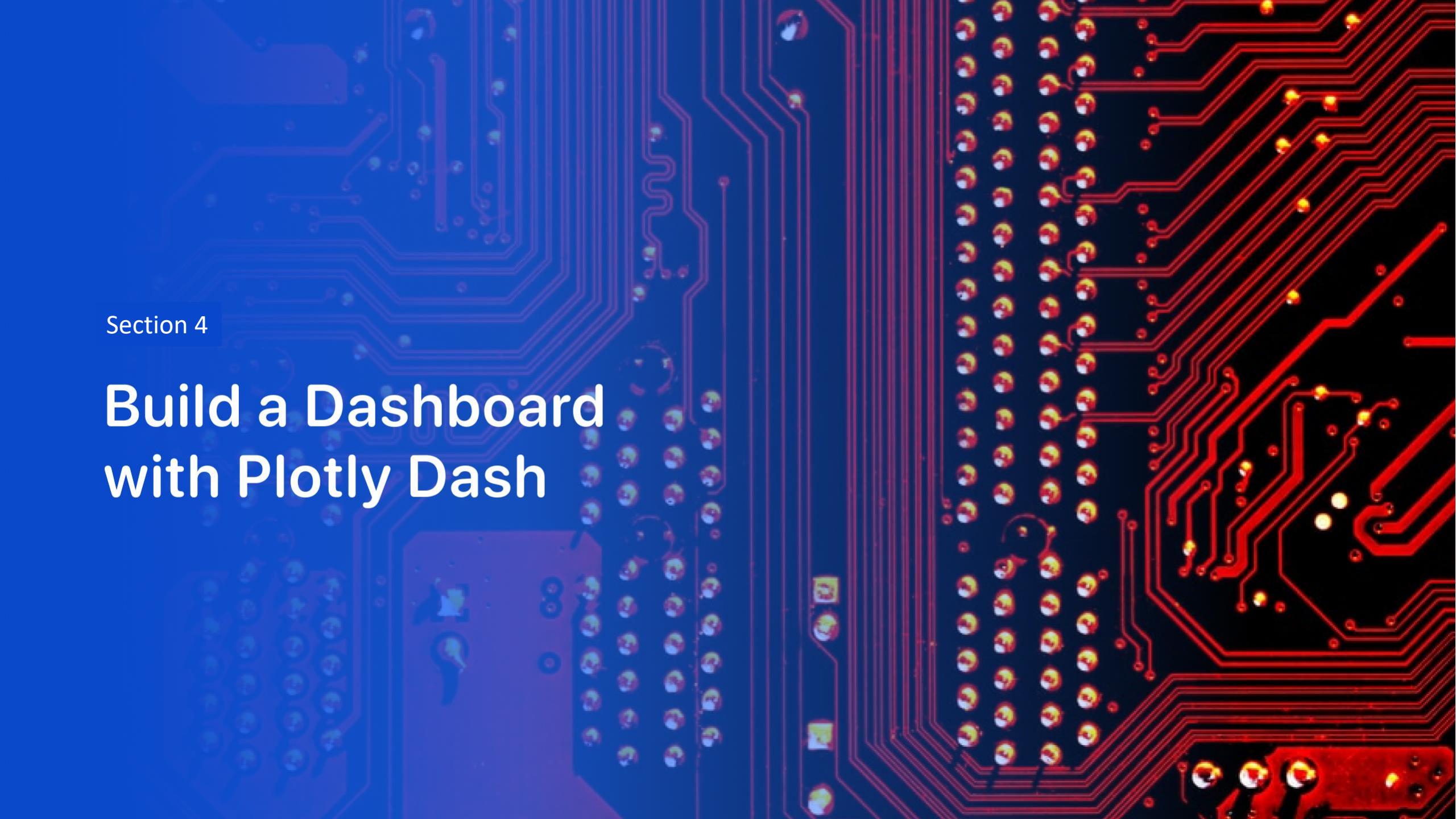


Distances between a launch site to its proximities



After you plot distance lines to the proximities, you can answer the following questions easily:

- Are launch sites near railways? – Yes (~1.28km)
- Are launch sites near highways? – Yes (~0.58km)
- Are launch sites near coastline? – Yes (~0.86km)
- Do launch sites keep certain distance away from cities? – Yes (~51.43km)



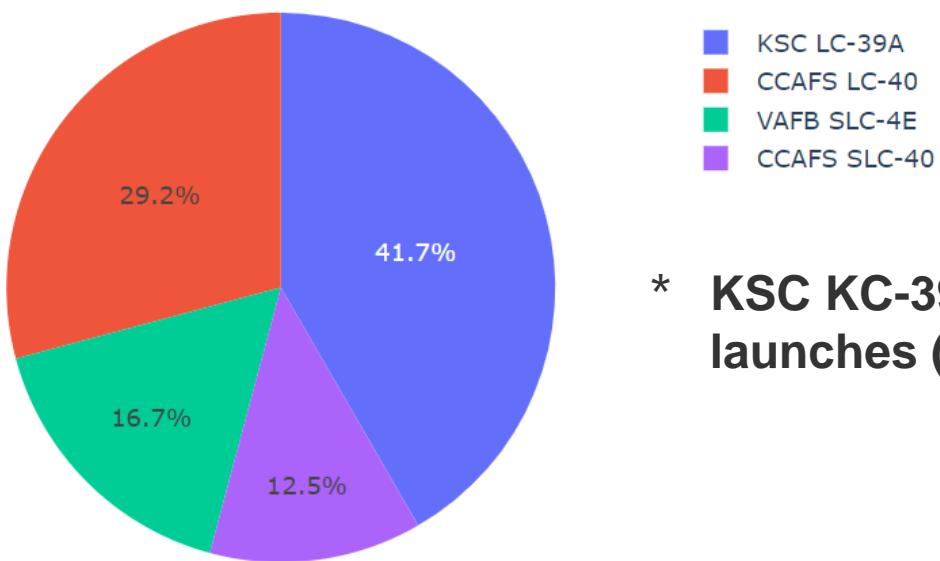
Section 4

Build a Dashboard with Plotly Dash

Total Success Launches by Site

All Sites x ▾

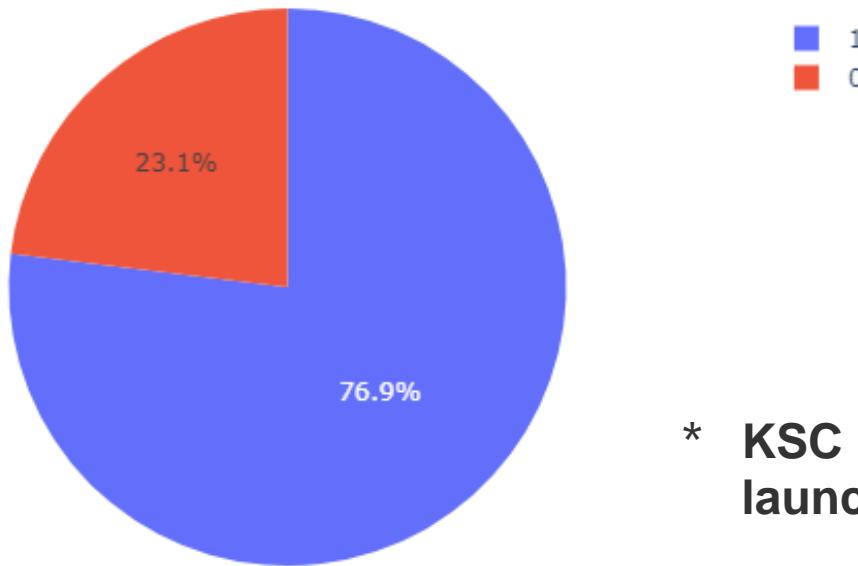
Total Success Launches by Site



- * **KSC KC-39A has the largest successful launches (10 launches ~ 41.7%)**

The highest launch success ratio

Total Success Launches for site KSC LC-39A

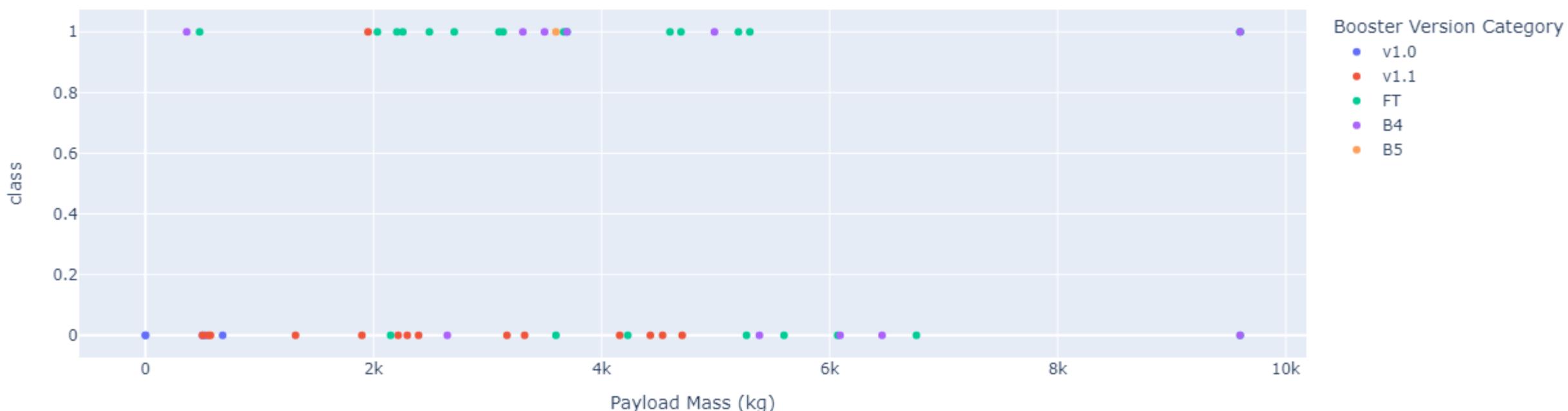


- * KSC KC-39A also has the highest launch success rate (76.9%)

Correction between Payload Mass and Success Launches

- Payload range from **2k** to **4k** has the highest launch success rate
 - Payload range from **6k** to **8k** has the lowest launch success rate
 - F9 Booster with **FT** version has the highest launch success rate

Correction between Payload and Success for all Sites

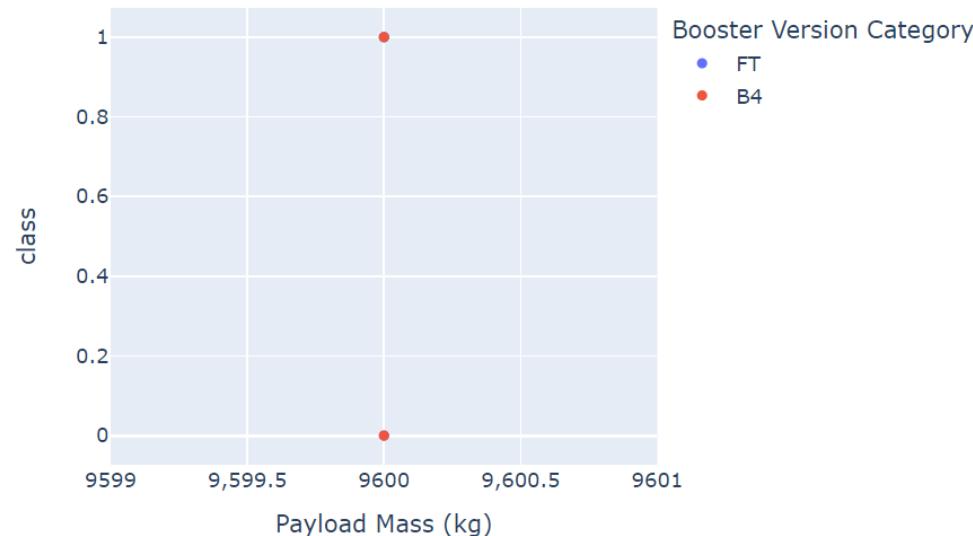


Correction between Payload Mass and Success Launches

Payload range (Kg):



Correction between Payload and Success for all Sites



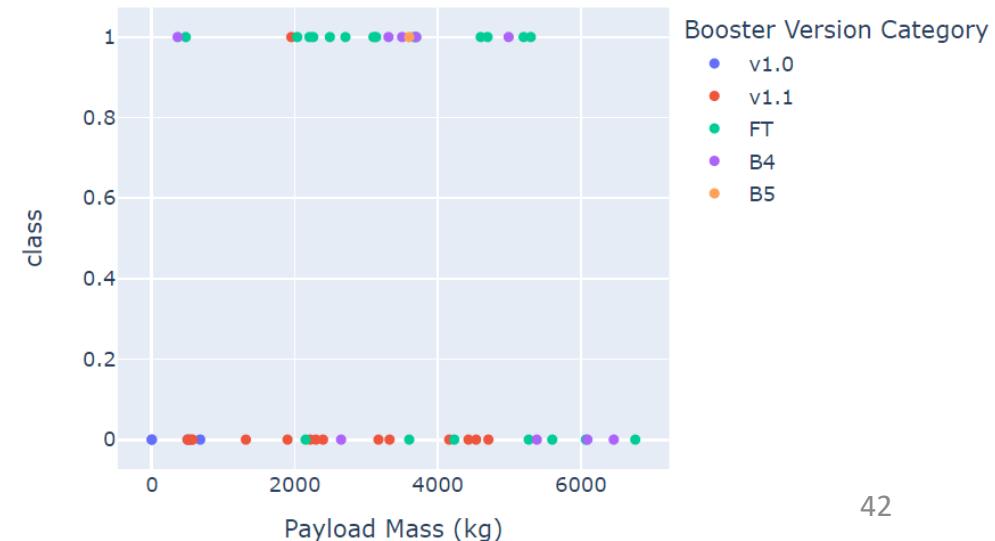
- There's not enough data to estimate risk of launches over 7,000kg

- Payloads under 6,000kg and FT boosters are the most successful combination.

Payload range (Kg):



Correction between Payload and Success for all Sites



The background of the slide features a dynamic, abstract design. It consists of several curved, overlapping bands of color. A prominent band on the left is a bright blue, while another on the right is a warm yellow. These colors transition into lighter, more diffused tones towards the edges of the frame. The overall effect is one of motion and depth.

Section 5

Predictive Analysis (Classification)

Classification Accuracy

- Four classification models were tested, and their accuracies were showed in report
- The model with the highest classification accuracy is the Decision Tree Classifier

Find the method performs best:

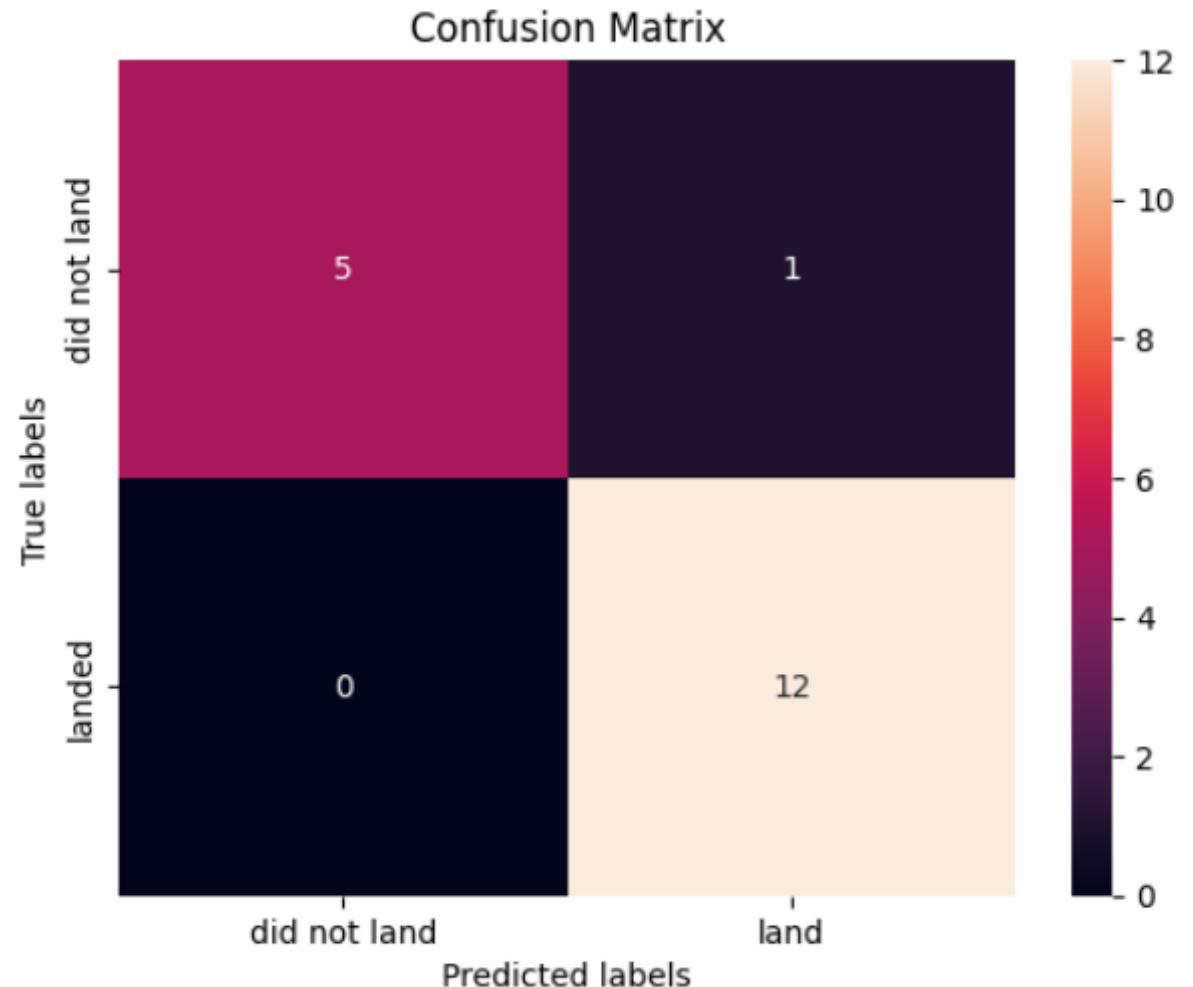
```
: rp = pd.DataFrame({'Method' : ['Test Accuracy']})  
  
knn_core=knn_cv.score(X_test, Y_test)  
tree_core=tree_cv.score(X_test, Y_test)  
svm_core=svm_cv.score(X_test, Y_test)  
logreg_core=logreg_cv.score(X_test, Y_test)  
  
rp['Logistic_Reg'] = [logreg_core]  
rp['SVM'] = [svm_core]  
rp['Decision Tree'] = [tree_core]  
rp['KNN'] = [knn_core]  
  
rp = rp.loc[0]  
rp
```

```
: Method          Test Accuracy  
Logistic_Reg    0.833333  
SVM            0.833333  
Decision Tree   0.944444  
KNN            0.833333  
Name: 0, dtype: object
```

Confusion Matrix of Decision Tree Classifier

- The confusion matrix of the Decision Tree Classifier proves its accuracy by showing the big numbers of true positives and true negatives compared to the false ones.
- The major problem is the false positives. i.e., an unsuccessful landing is marked as a successful landing by the classifier.

```
yhat = tree_cv.predict(X_test)  
plot_confusion_matrix(Y_test,yhat)
```



Conclusions



Space X uses 4 different launch sites and the larger the flight amount the greater the success rate at a launch site.



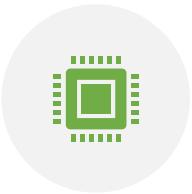
Launch success rate started to increase in 2013 till 2020. The number of landing outcomes became as better as the years passed.



Orbits ES-L1, GEO, HEO, SSO, VLEO had the most success rate (**over 80%**).



KSC LC-39A had the most successful launches of any sites.



The Decision tree classifier is the best machine learning algorithm for this task and can be used to predict successful landings and increase profits.

Appendix

Applied-Data-Science-Capstone / lab_jupyter_launch_site_location.ipynb ↑ Top

Preview | Code | Blame | 627 lines (627 loc) · 135 KB | Code 55% faster with GitHub Copilot | Raw |

TODO: For each launch result in `spacex_df` data frame, add a `folium.Marker` to `marker_cluster`

In [13]:

```
# Add marker_cluster to current site_map
site_map.add_child(marker_cluster)

# for each row in spacex_df data frame
# create a Marker object with its coordinate
# and customize the Marker's icon property to indicate if this launch was successed or failed,
# e.g., icon=folium.Icon(color='white', icon_color=row['marker_color'])
for index, record in spacex_df.iterrows():
    # TODO: Create and add a Marker cluster to the site map
    # marker = folium.Marker(...)
    marker_cluster.add_child(marker)

site_map
```

Out[13]: Make this Notebook Trusted to load map: File -> Trust Notebook

Folium can not show maps on the GitHub repository

Thank you!

