IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Edgar Tijerina Tamez
07/Oct/2025

Outline

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

Executive Summary

- Exploratory data analysis was conducted using a predictive approach to find patterns, distributions, correlations and anomalies, using the SpaceX REST API as the main data source. Complementary data was sourced from Wikipedia using web scrapping techniques. Data was cleaned and prepared for machine learning using One Hot Encoding. Interactive visual analytics were developed with Folium as well interactive dashboards using Plotly Dash. Machine learning pipelines were set up to test multiple classifiers, including Logistic Regression, Support Vector Machine (SVM), Decision Tree Classifier and K-Nearest Neighbors (KNN). Model hyperparameters where optimized and the results visualized on Confusion matrices.

- From the exploratory data analysis we can observe that landing success rate has been increasing since 2010 reaching 80% success rate as of 2020. We can also observe that Geostationary Orbits (GTO) had the lowest success rate with 50%. All machine learning prediction models reached an 83% accuracy. The confusion matrices demonstrated the ability of the model to predict with perfect accuracy True Positives (Successful landings) but struggled with predict True Negatives (Failed landing).

Introduction

- Launching rockets is a multi million-dollar endeavor. Right now, the space launch market is entirely dominated by SpaceX, composing over 75% of all commercial launches as of 2023. Their main advantage is their ability to reuse their rockets, cutting their costs per launch from $150 million to $97 million.
- Being able to predict whether the first stage will land would give an advantage to competitors by estimating launch costs.

Section 1

Methodology

Methodology

Executive Summary

Data was collected from two main sources: SpaceX API using Python's Requests module and Wikipedia Falcon 9 launch records using BeautifulSoup.

Data was cleaned and transformed using Pandas and Numpy. One hot encoding was used to transform the data in preparation for machine learning.

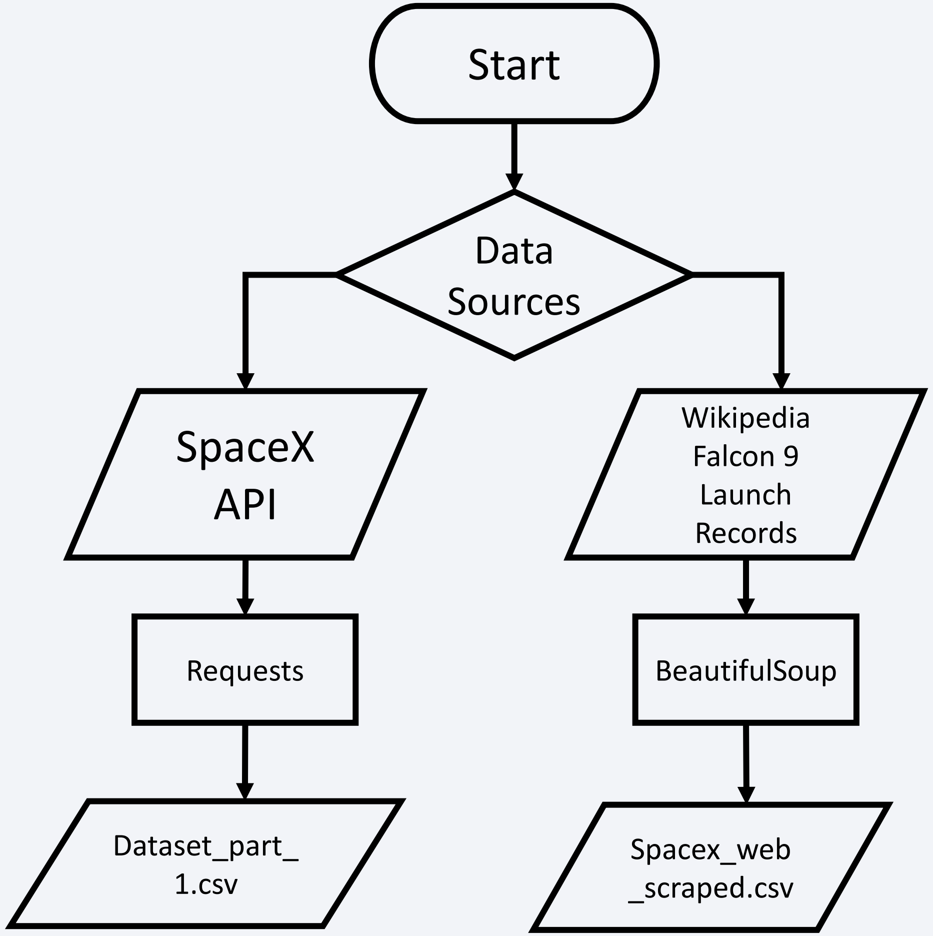
EDA was performed using matplotlib and seaborn for plotting and key data points were extracted using SQL.

Interactive visual analytics were developed with Folium as well interactive dashboards using Plotly Dash.

Machine learning pipelines were set up to test multiple classification models, including Logistic Regression, Support Vector Machine (SVM), Decision Tree Classifier and K-Nearest Neighbors (KNN). Model hyperparameters where optimized and the results visualized on Confusion matrices.

Data Collection

Data was collected from two main sources: SpaceX API using Python's Requests module and Wikipedia Falcon 9 launch records using BeautifulSoup.

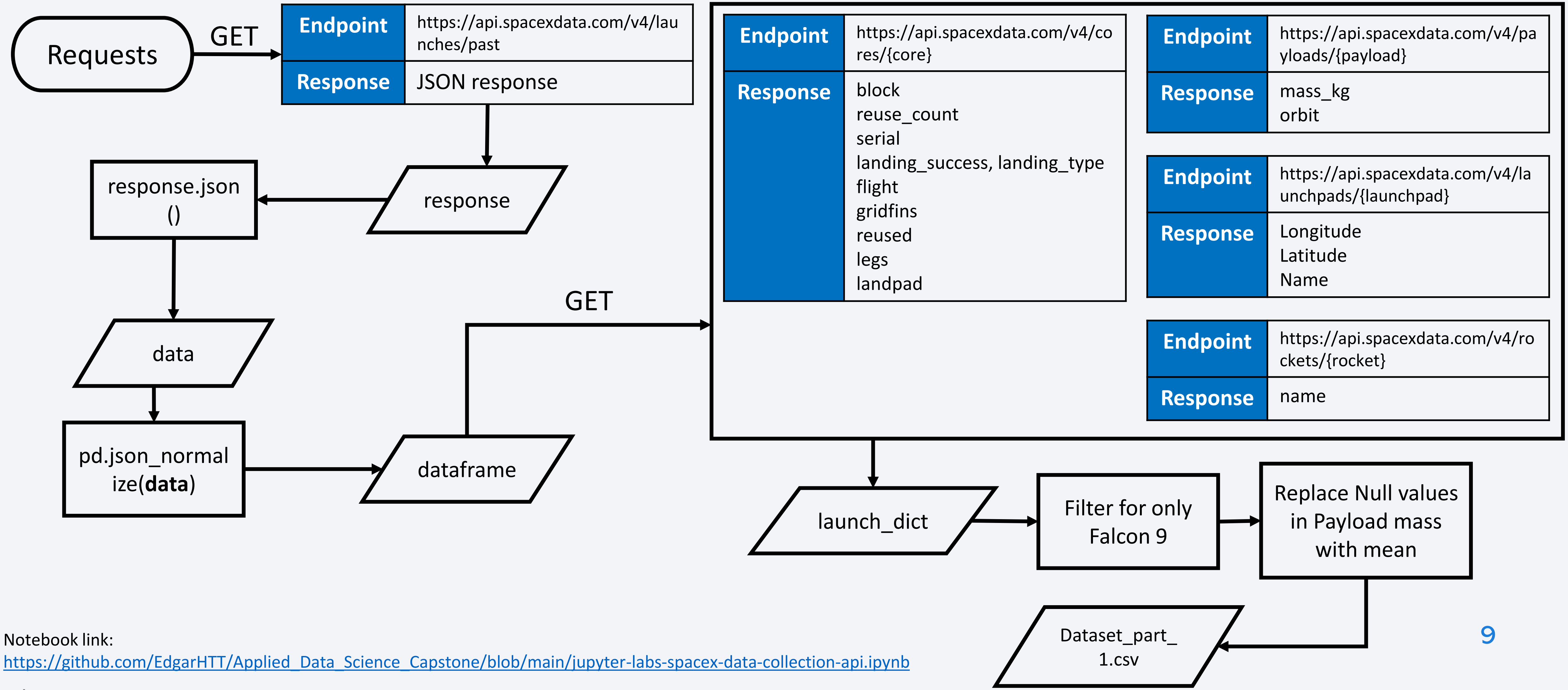


Data Collection – SpaceX API

- The **primary source** of data is the SpaceX REST API -> **api.spacexdata.com/v4/**
- Its key endpoints are:
  - **/launches/past** -> past launch data (main dataset).
  - **/capsules**, **/cores**, **/payloads**, **/launchpads** -> detailed info by ID.
- Using Python requests we obtained **JSON** objects
- By using **json_normalise** from **Pandas DataFrame** we transform the data into a **tabular structure**

Notebook link:
https://github.com/EdgarHTT/Applied_Data_Science_Capstone/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

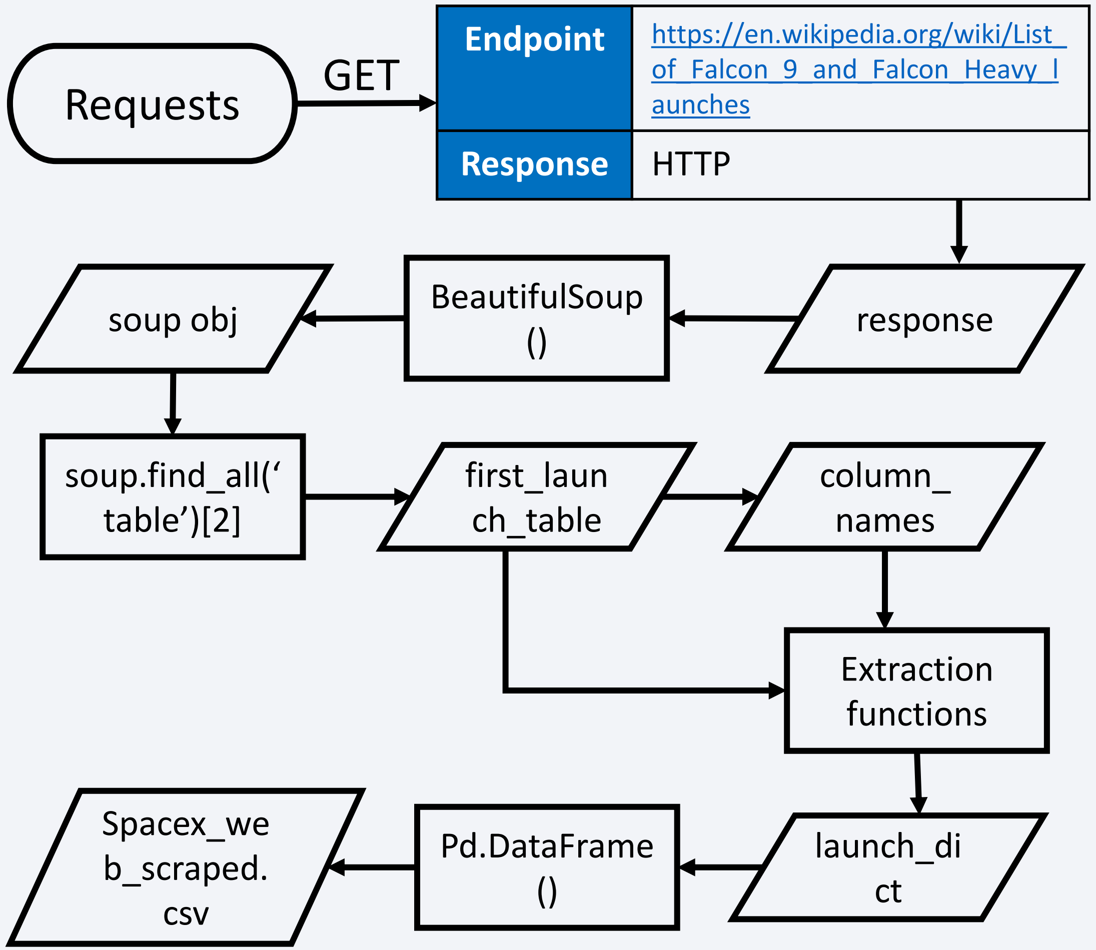
Data Collection – SpaceX API



Notebook link:
https://github.com/EdgarHTT/Applied_Data_Science_Capstone/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

Data Collection - Scraping

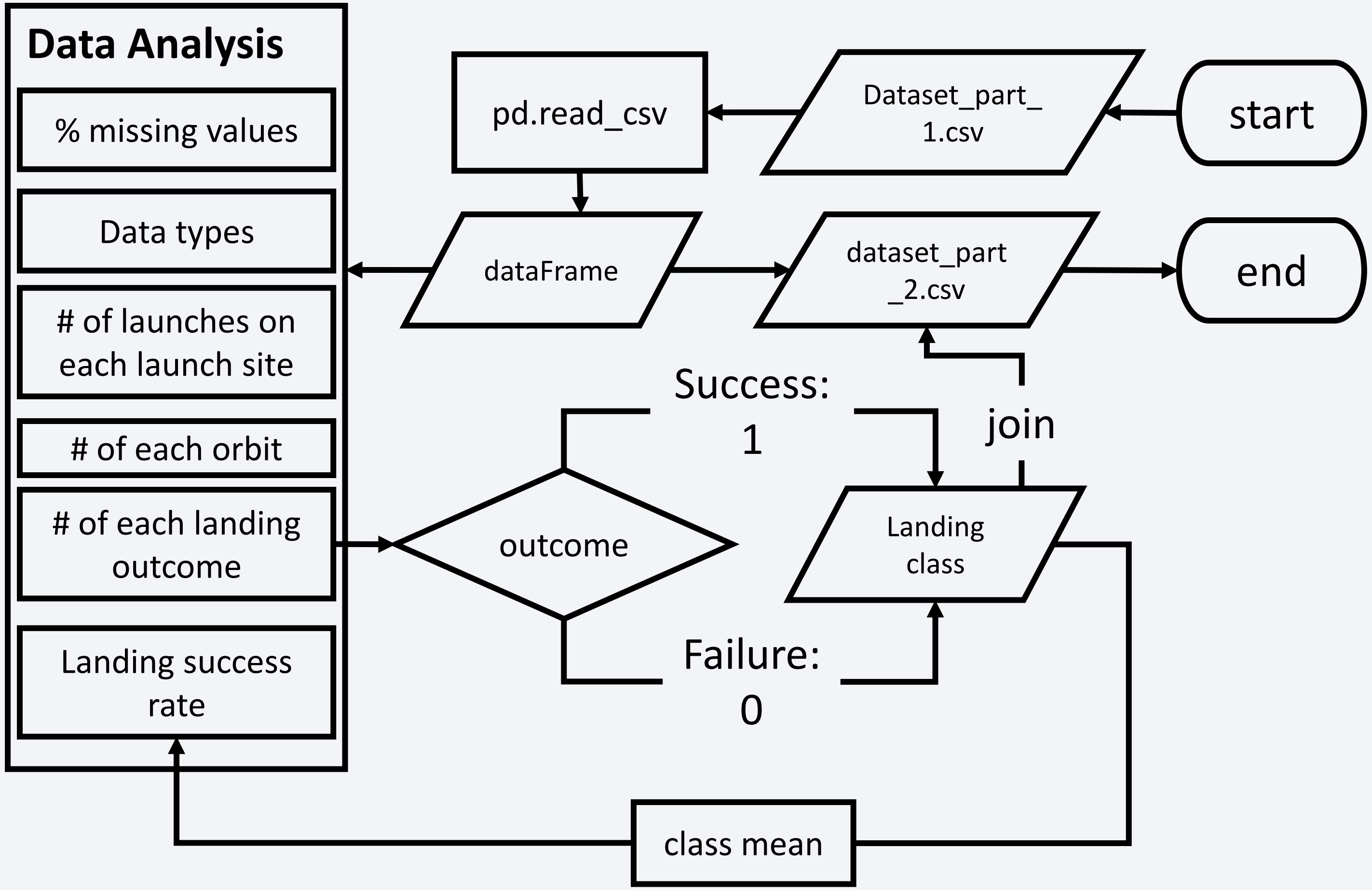
Requests was used in tandem with BeautifulSoup to parse and scrape the List of Falcon 9 and Falcon Heavy launches from Wikipedia. The parsed data was converted into a Pandas Dataframe and stored for further use.



Notebook link:
https://github.com/EdgarHTT/Applied_Data_Science_Capstone/blob/main/jupyter-labs-webscraping.ipynb

Data Wrangling

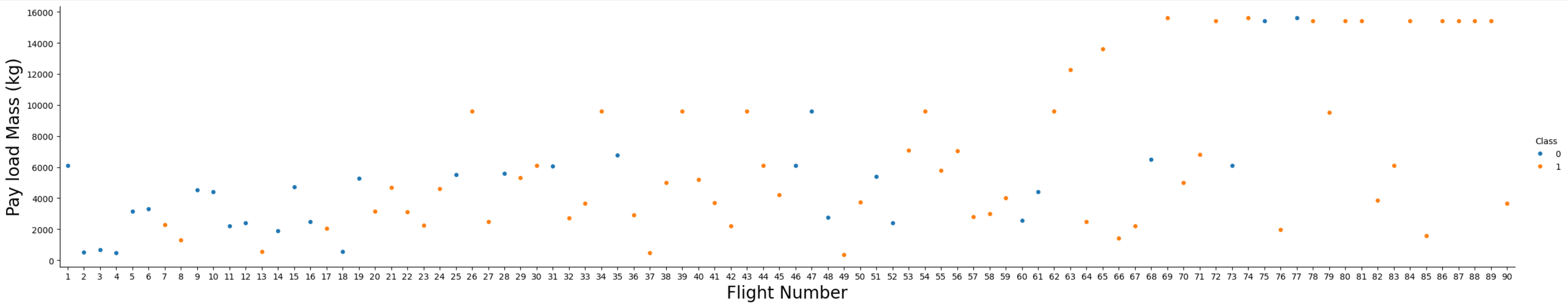
Data wrangling conforms exploratory data analysis and the determination of training labels. By analyzing the dataset, we classified the outcome of each landing in a binary format and incorporated it into the dataset for future feature engineering.



Notebook:
https://github.com/EdgarHTT/Applied_Data_Science_Capstone/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

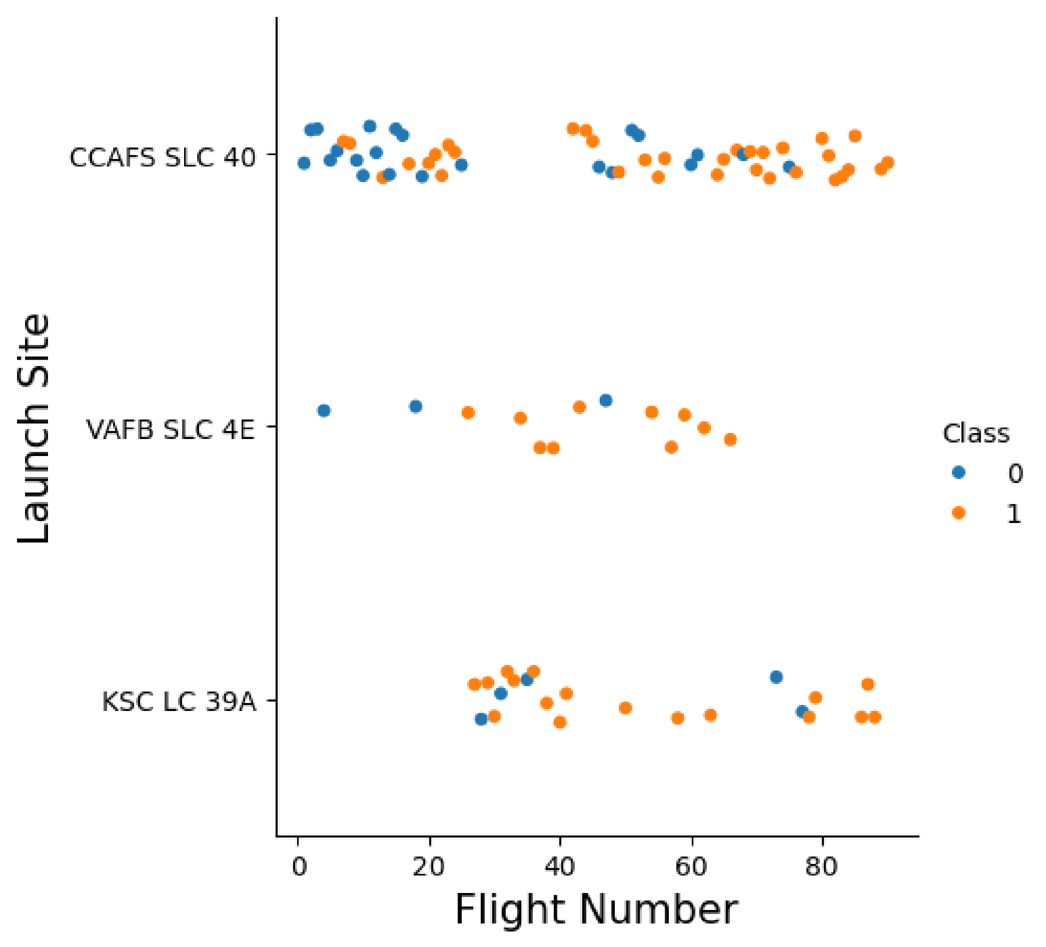
As part of the exploratory data analysis, visualizations are developed to obtain a deeper understanding and find insights.



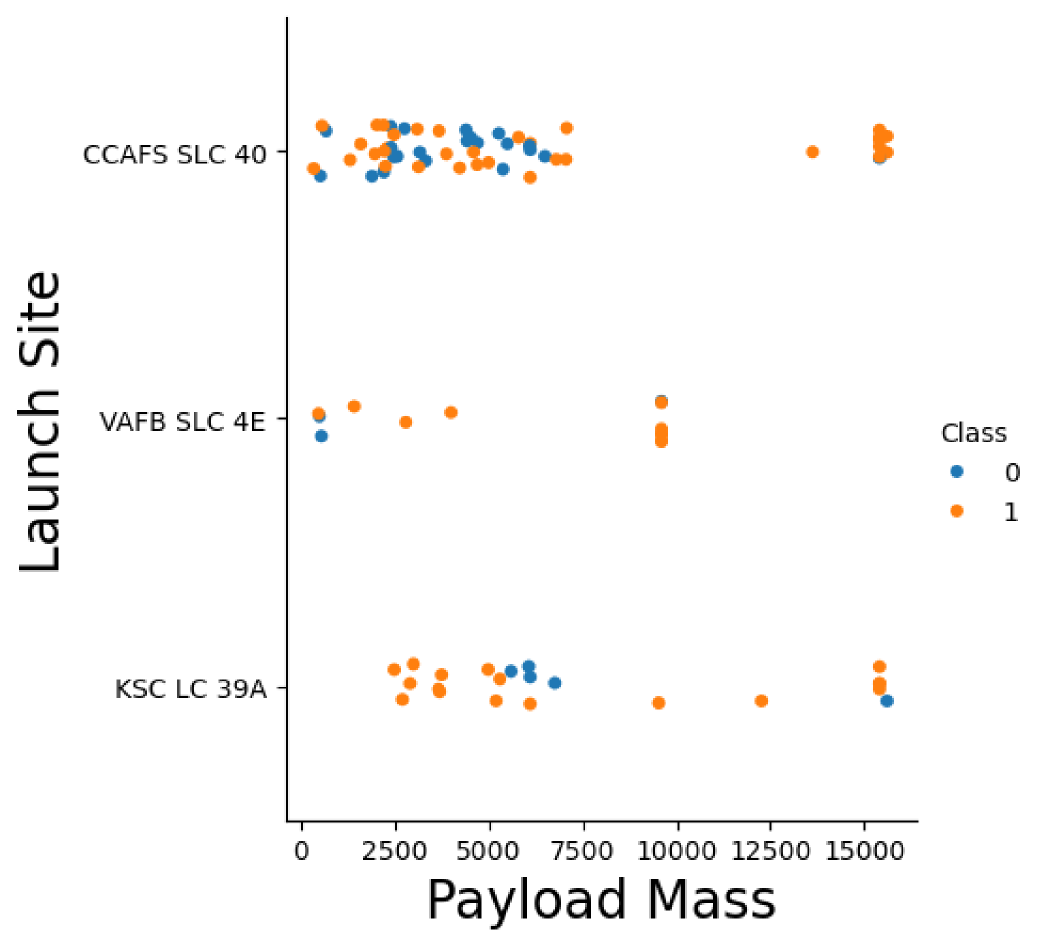
By plotting Flight number against Payload mass, we can observe that the success rate of each launch increases. Payload mass and landing success doesn’t seem to be correlated

EDA with Data Visualization

By visualizing the relationship between flight number and launch site, we can observe that the first launches done by SpaceX was on Cape Canaveral and that most of their failures were done there.



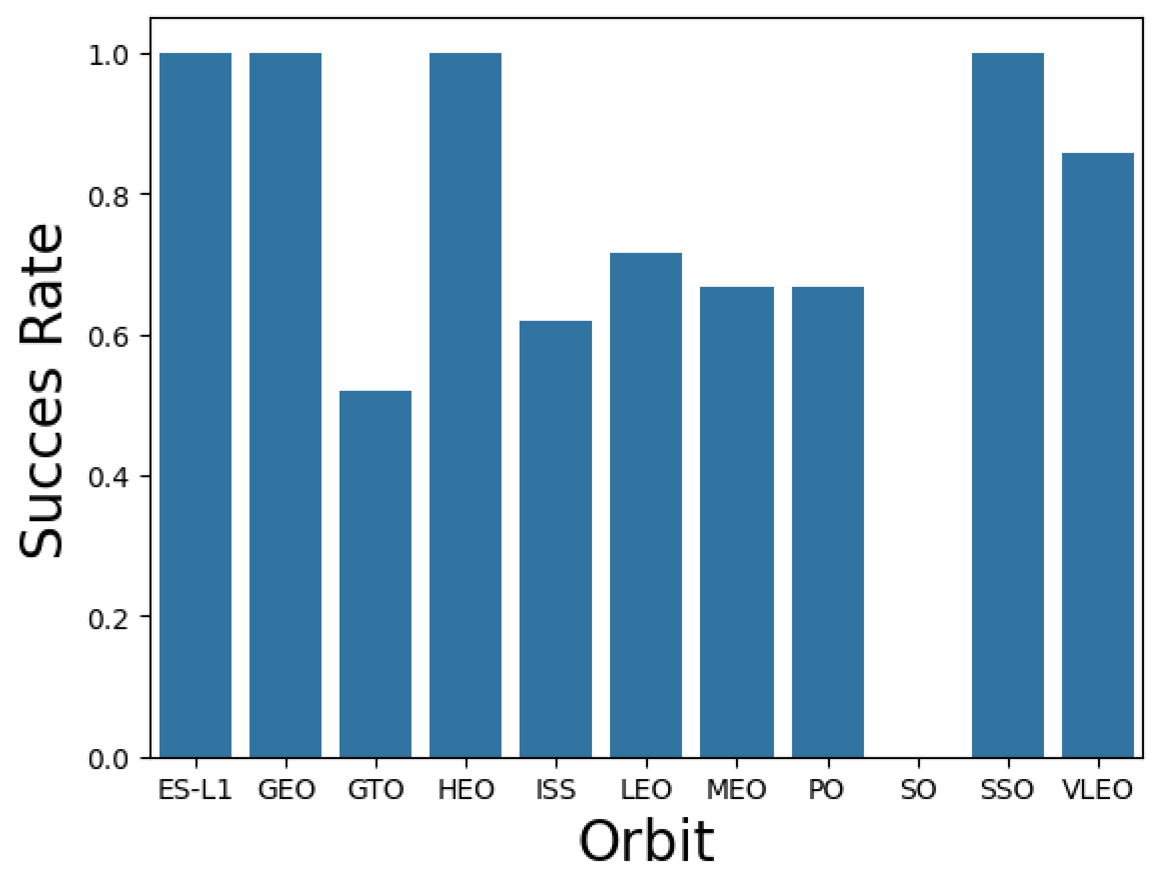
EDA with Data Visualization



And by plotting against payload mass, we can see that for the VAFB-SLC launch site no heavy payload rockets were launched.

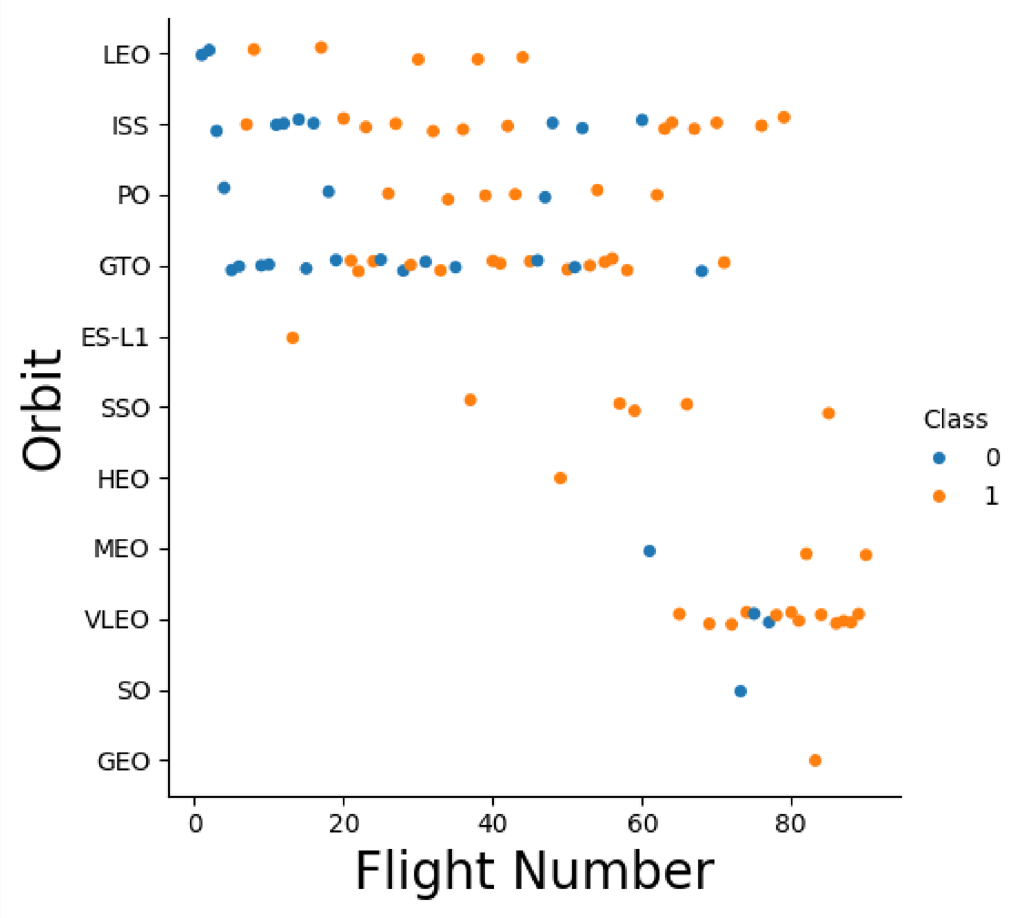
EDA with Data Visualization

By plotting the success rate of each orbit, we can see that Geo Stationary Orbits (GTO) are the most difficult launches to land.

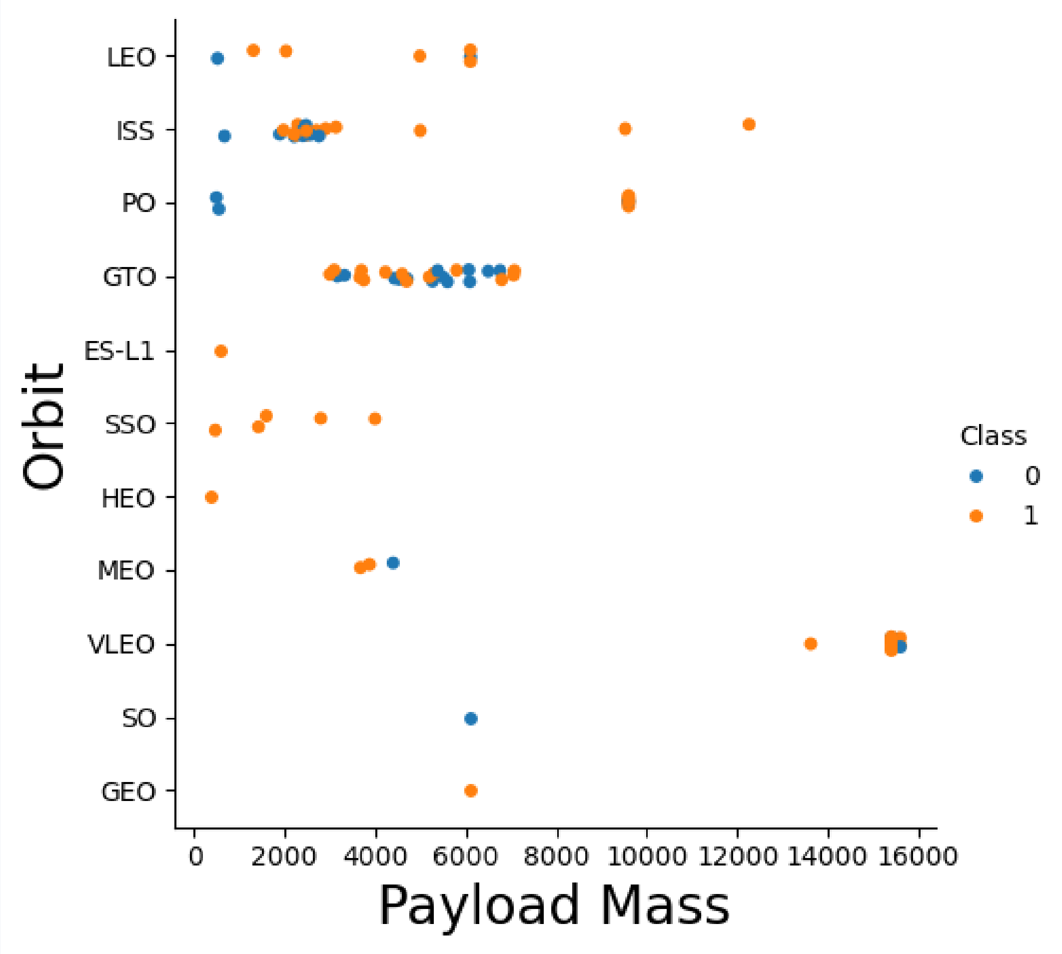


EDA with Data Visualization

As of now the most common mission by SpaceX is Very Low Earth Orbit (VLEO). This orbit is most common for satellite missions like their Starlink service



EDA with Data Visualization

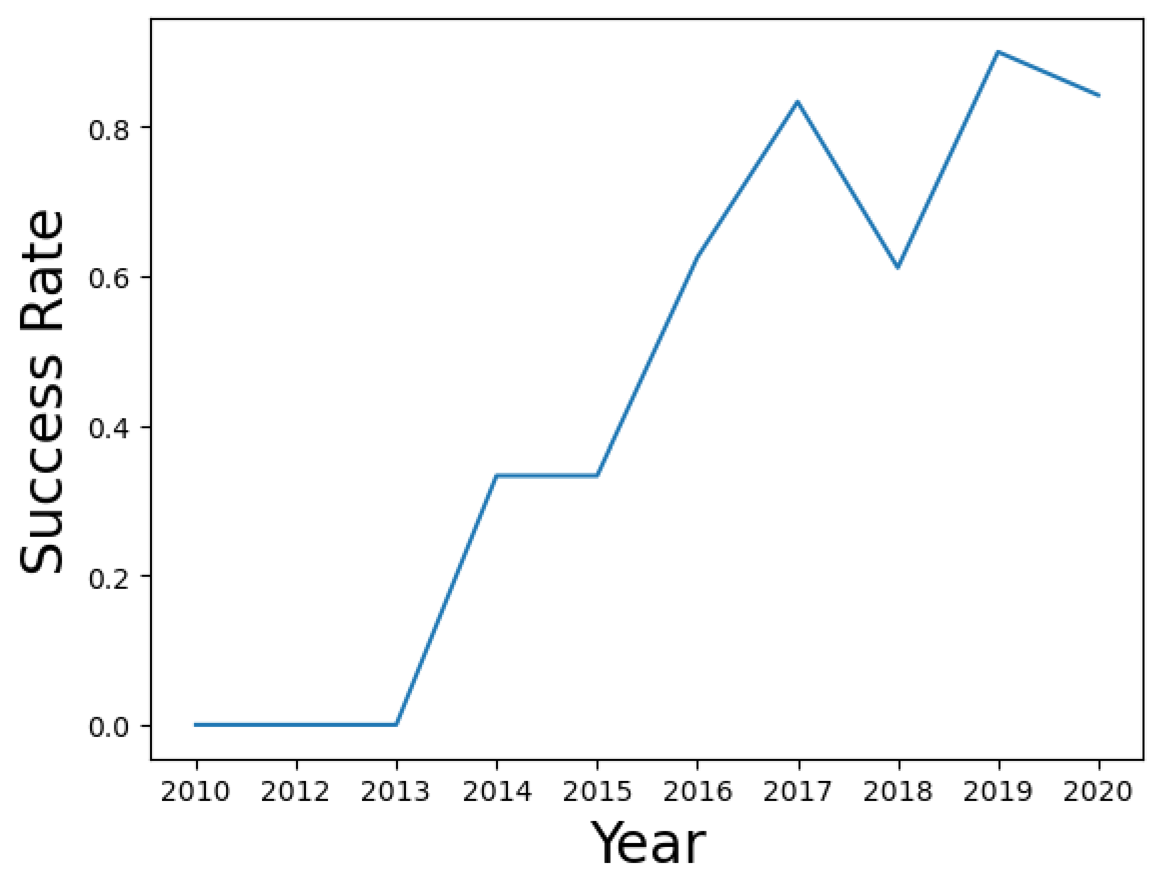


With heavy payloads the successful landing rate is increased for Polar, LEO and ISS.

However, for GTO, it’s difficult to distinguish between successful and unsuccessful landings

EDA with Data Visualization

Finally, we can observe that the success rate kept increasing from 2013 till 2020



EDA with SQL

SQL was also used to perform exploratory data analysis. Here are the queries used:

- %sql **PRAGMA table_info([SPACEXTABLE])**
  - This query gives us details about the table's columns.

cid	name	type	notnull	dflt_value	pk
0	Date	TEXT	0	None	0
1	Time (UTC)	TEXT	0	None	0
2	Booster_Version	TEXT	0	None	0
3	Launch_Site	TEXT	0	None	0
4	Payload	TEXT	0	None	0
5	PAYLOAD_MASS__KG_	INT	0	None	0
6	Orbit	TEXT	0	None	0
7	Customer	TEXT	0	None	0
8	Mission_Outcome	TEXT	0	None	0
9	Landing_Outcome	TEXT	0	None	0

Notebook:
https://github.com/EdgarHTT/Applied_Data_Science_Capstone/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

EDA with SQL

%sql SELECT DISTINCT("Launch_Site") FROM SPACEXTABLE

- Displays the name of the unique launch sites

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

%%sql

SELECT * FROM SPACEXTABLE

WHERE "Launch_Site" LIKE 'CCA%'

LIMIT 5

- Displays 5 records where launch sites begin with "CCA"

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

EDA with SQL

```
%%sql
SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTABLE
   WHERE Customer = 'NASA (CRS)’
```

SUM(PAYLOAD_MASS_KG_)
45596

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

```
%%sql
SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTABLE
   WHERE "Booster_Version" LIKE "F9 v1.1%“
```

AVG(PAYLOAD_MASS_KG_)
2534.666666666665

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

EDA with SQL

```
%%sql
SELECT MIN(DATE) FROM SPACEXTABLE
   WHERE Landing_Outcome = "Success (ground pad)"
```

- Date of the first successful landing outcome in ground pad

MIN(DATE)
2015-12-22

```
%%sql
SELECT Booster_Version FROM SPACEXTABLE
   WHERE Landing_Outcome = "Success (drone ship)" AND
PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000
```

- Lists the names of the boosters which have successfully landed on a drone ship and had a mass between 4000 and 6000 kgA

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

EDA with SQL

```
%%sql
SELECT Landing_Outcome, COUNT(Landing_Outcome) FROM SPACEXTABLE
  GROUP BY Landing_Outcome LIKE "Success%" OR "Failure%"
```

- List the total number of successful and failure mission outcomes (note that the names between parenthesis do not indicate the class of outcome)

Landing_Outcome	COUNT(Landing_Outcome)
Failure (parachute)	40
Success (ground pad)	61

```
%%sql
SELECT Booster_Version FROM SPACEXTABLE
  WHERE PAYLOAD_MASS__kg_ =
    (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE)
  GROUP BY Booster_Version
```

- Lists all the booster versions that have carried the maximum payload

Booster_Version
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

EDA with SQL

```
%%sql
SELECT SUBSTR(DATE,6,2), Landing_Outcome, Booster_Version, Launch_Site FROM SPACEXTABLE
   WHERE SUBSTR(DATE,0,5) = '2015' AND Landing_Outcome = 'Failure (drone ship)’
```

- List and displays month number, landing outcome, booster version and launch site of rockets launched on the year 2015 which ended on a failure to land on a drone ship

SUBSTR(DATE,6,2)	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

EDA with SQL

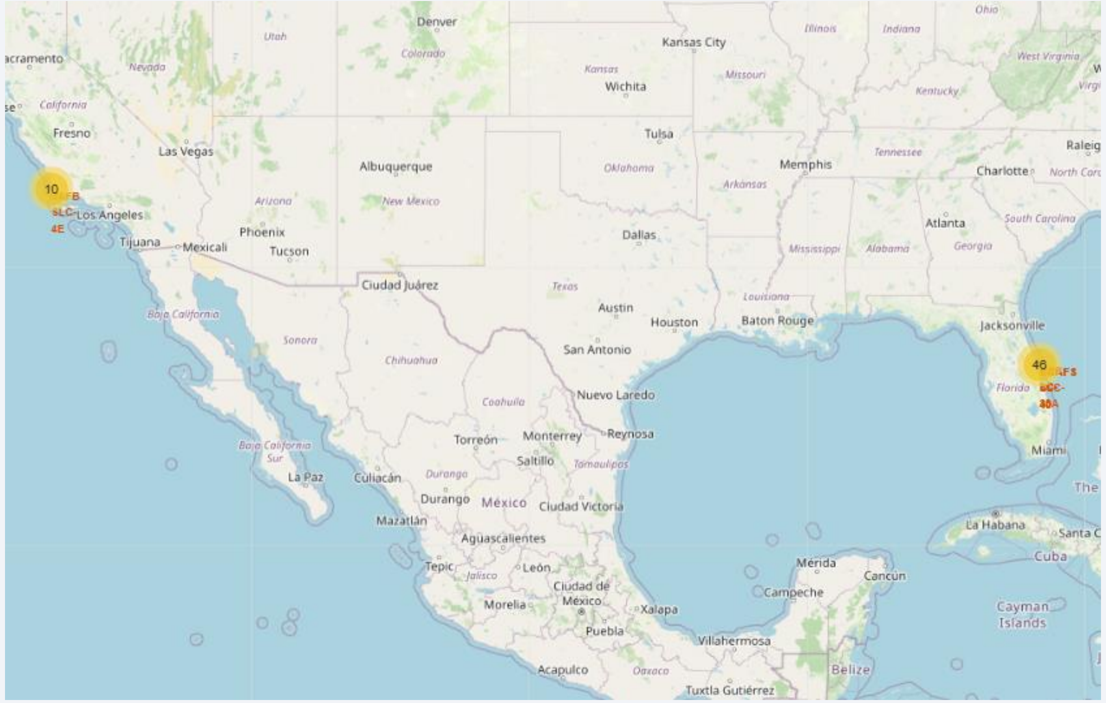
```
%%sql
SELECT Landing_Outcome, COUNT(Landing_Outcome) FROM SPACEXTABLE
  WHERE DATE BETWEEN "2010-06-04" AND "2017-03-20"
  GROUP BY Landing_Outcome
  ORDER BY COUNT(Landing_Outcome) DESC
```

- Ranks by count of landing outcomes between the date of 2010-06-04 and 2017-03-20 , in descending order.

Landing_Outcome	COUNT(Landing_Outcome)
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

Build an Interactive Map with Folium

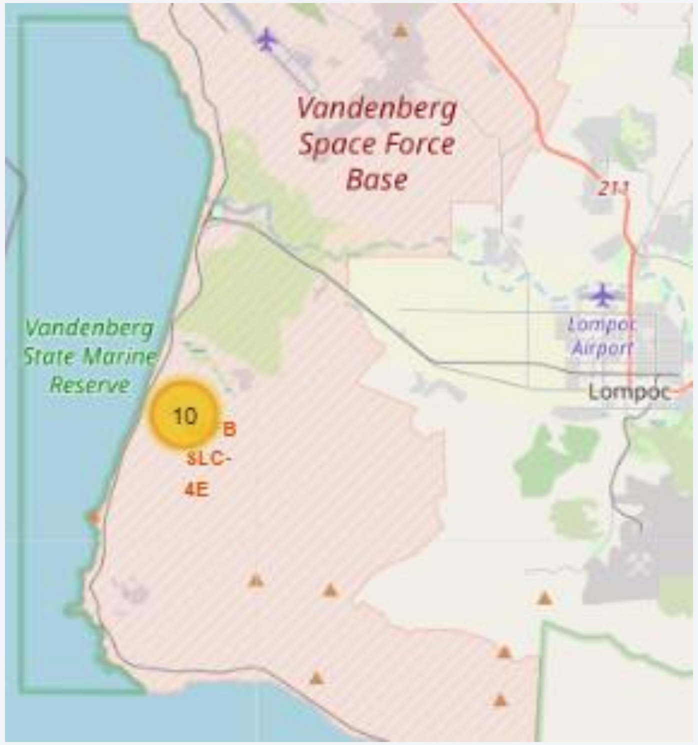
To explore how locations and its proximities affect the landing success rate an interactive map was constructed. Details were marked, these included calculated proximities to railways, cities, coastlines and highways. Successful and failed landings were clustered into their respective launch site.

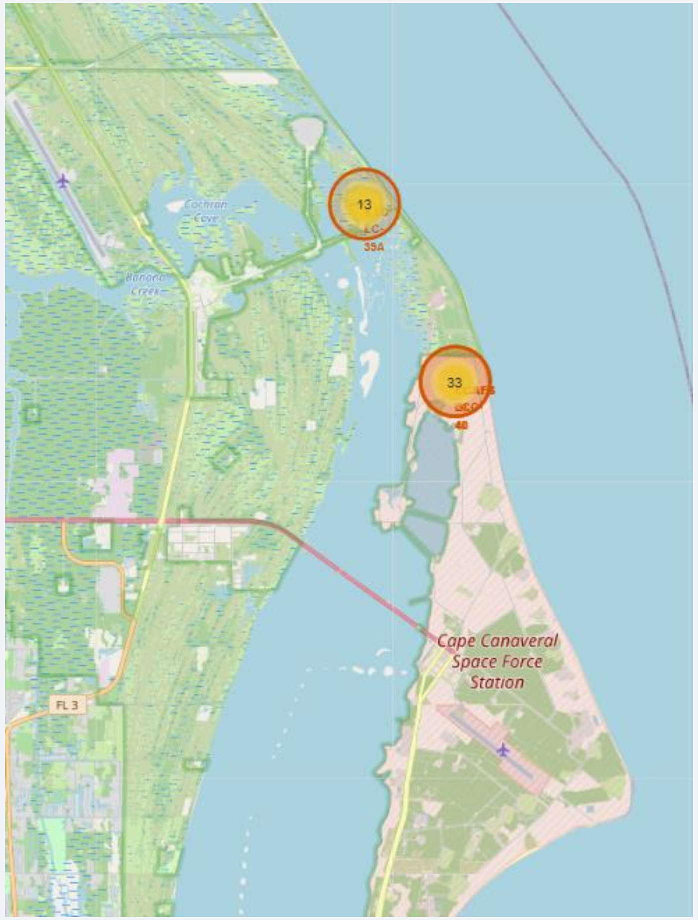


Notebook:
https://github.com/EdgarHTT/Applied_Data_Science_Capstone/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

Clustering let us visualize the number of launches per launch site.



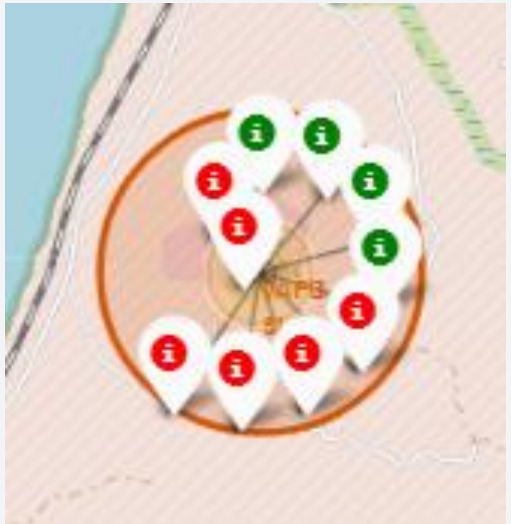


Edgar HTT

Build an Interactive Map with Folium

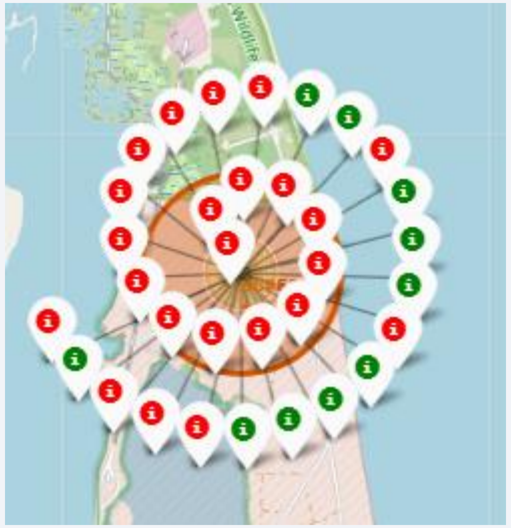
Clustering also let us see failed and successful launches easily

VAFB SLC-4E



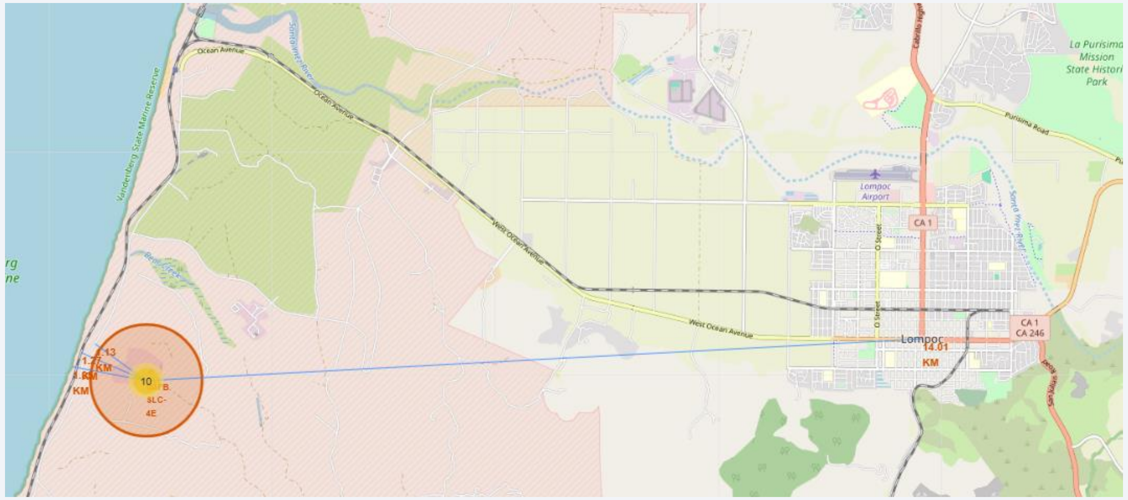
KSC LC-39A

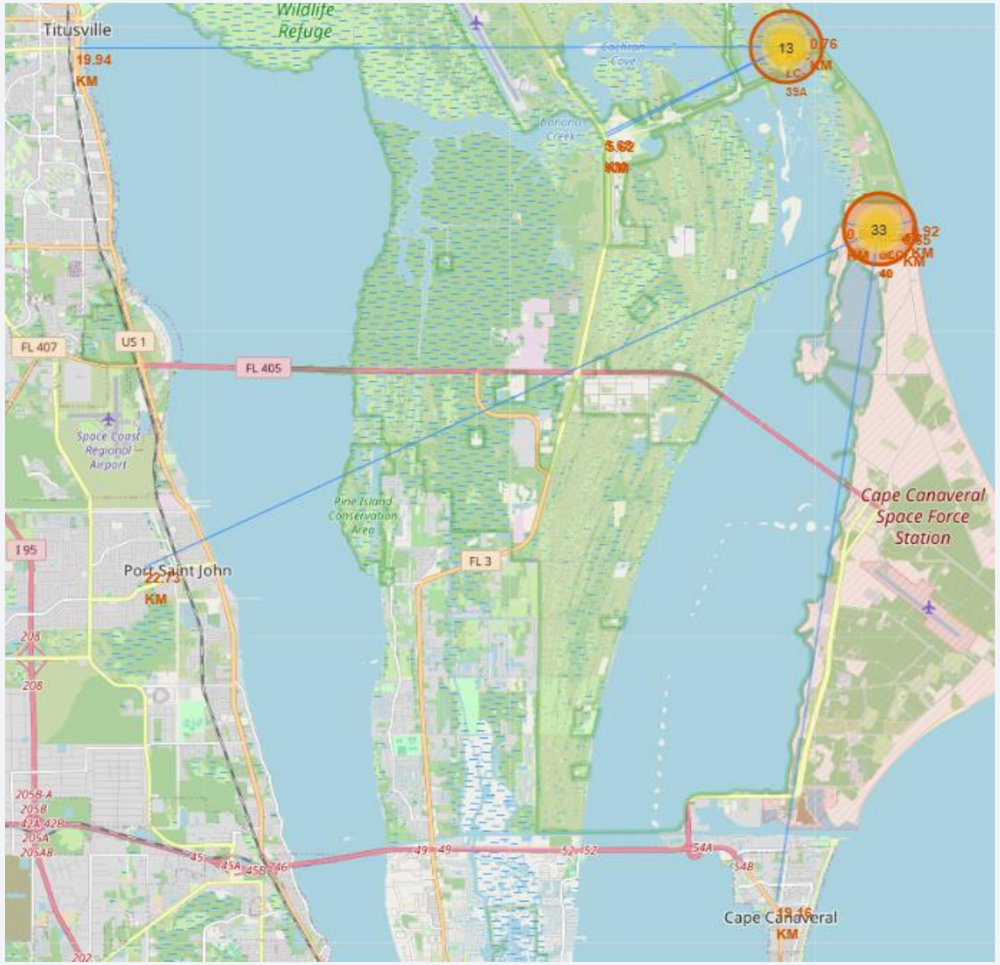
CCAFS LC-40
&
CCAFS SLC-40



Build an Interactive Map with Folium

The closest city was Lompoc with 14 km. Launch sites are usually close to a coastline and some public infrastructure, like highways and railways. But their location is instead the determining factor of which type of mission can the site launch.

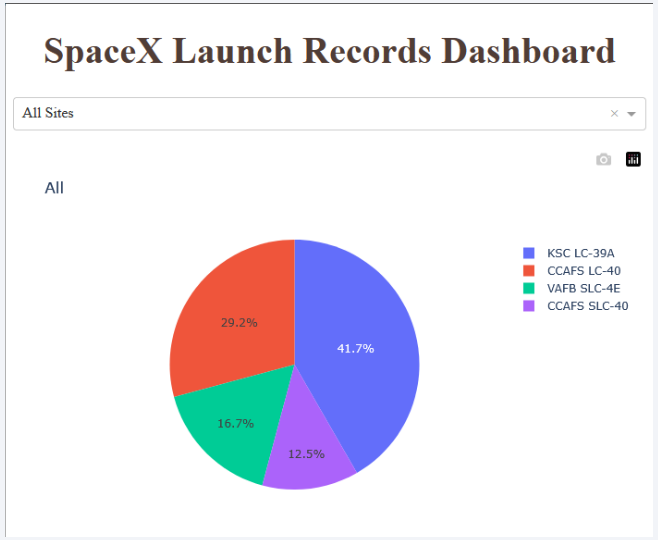


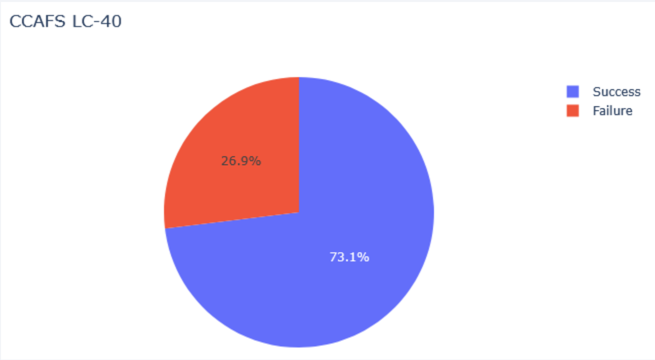


Build a Dashboard with Plotly Dash

A dashboard using Plotly Dash with some plots and interactions were developed to better explore the data extracted

A pie plot showcases the share of launches per site. You can select the site on a dropdown list which then shows the share of successful and failed launches



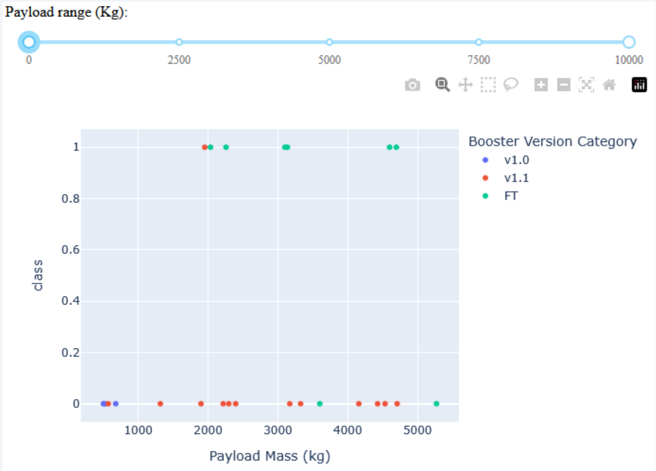


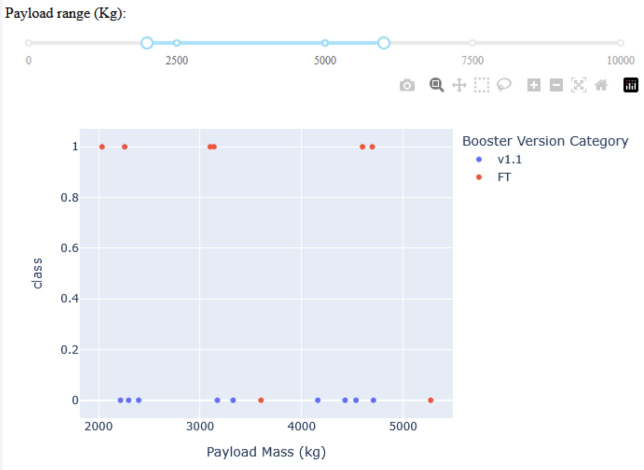
Notebook:
https://github.com/EdgarHTT/Applied_Data_Science_Capstone/blob/main/spacex-dash-app.py

Build a Dashboard with Plotly Dash

A scatter plot is used to represent the different outcomes of each launch, separated by booster version.

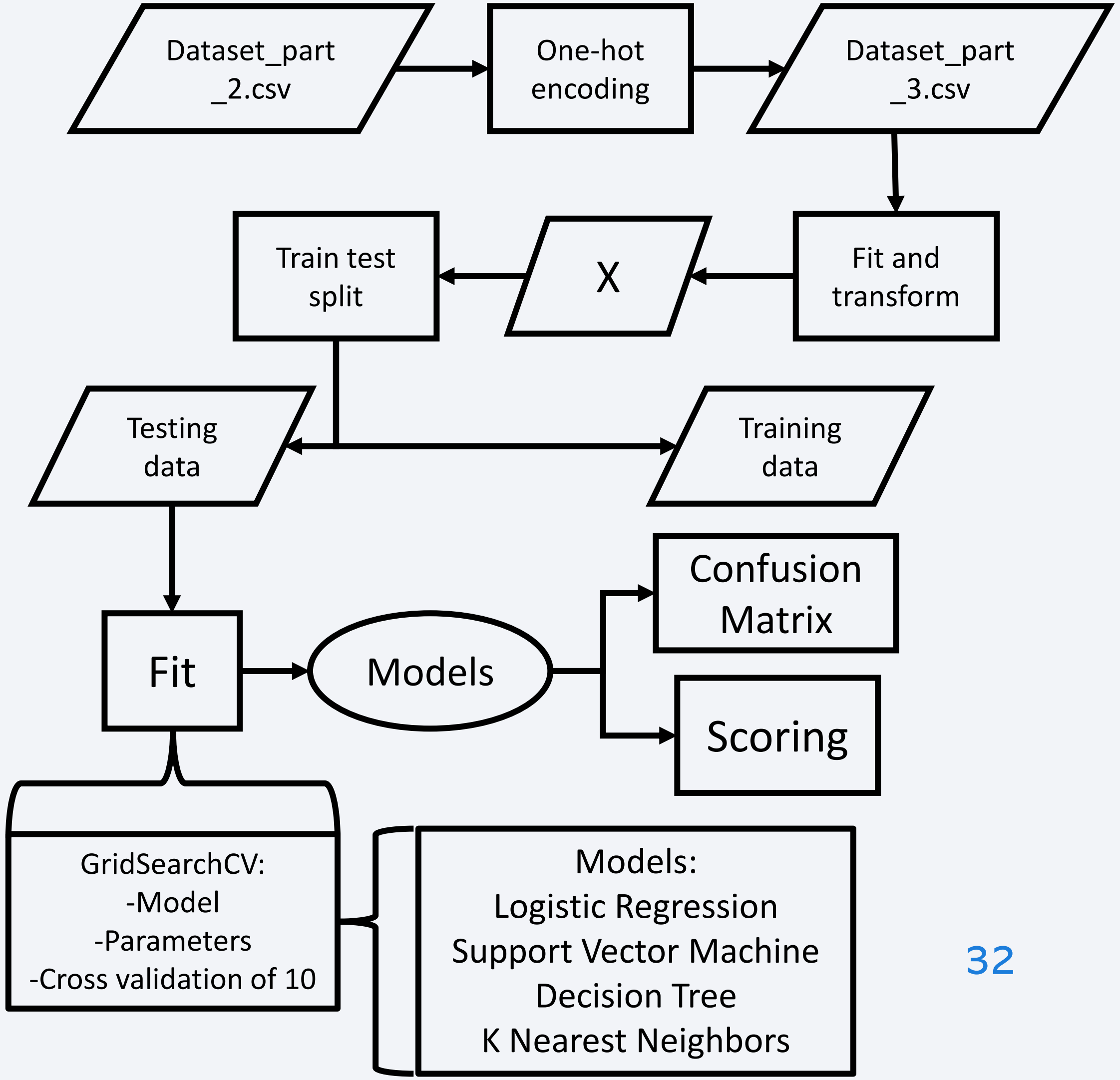
A slider can be used to filter by payload range





Predictive Analysis (Classification)

To build and train the classification model, feature engineering was performed on the dataset. Categorical data was transformed into numerical values by using One-hot encoding techniques. Once the dataset was ready, we proceeded to fit, transform and split into training and testing data. To determine the optimal hyper parameters, grid search was performed on each model. Score was calculated and a confusion matrix plotted to find the best model.

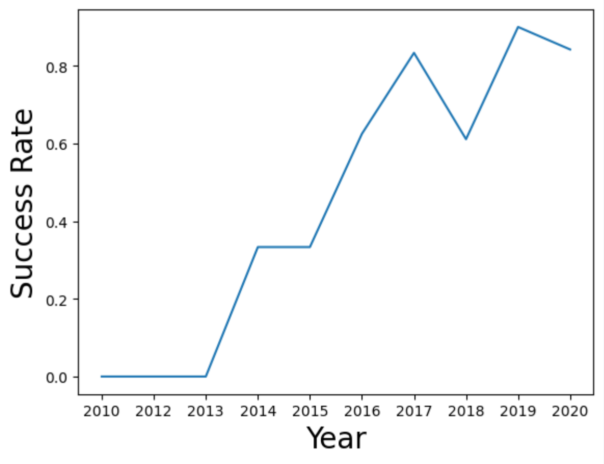


Notebook:
https://github.com/EdgarHTT/Applied_Data_Science_Capstone/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

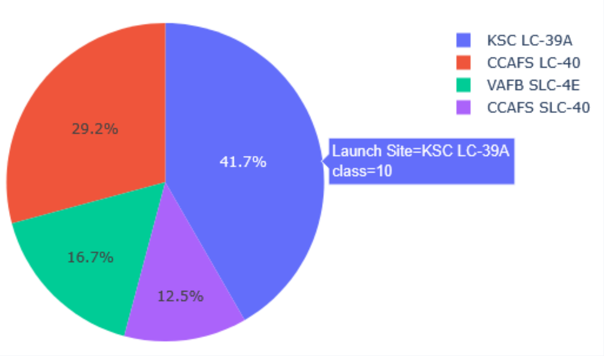
Results

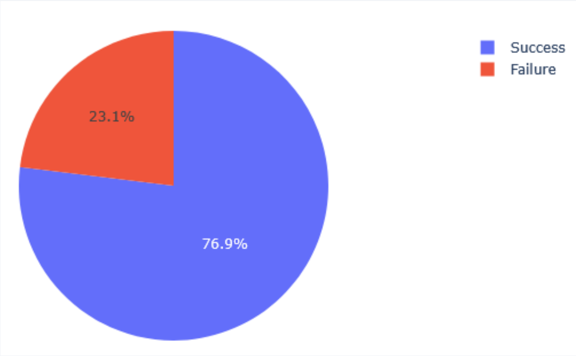
These are some of the most important results obtained of each section:

Exploratory data analysis:
The success rate of each launch has been increasing since 2013



Interactive analytics demo:
Kennedy space center LC-39A is the most used launch pad. With a 78% success rate



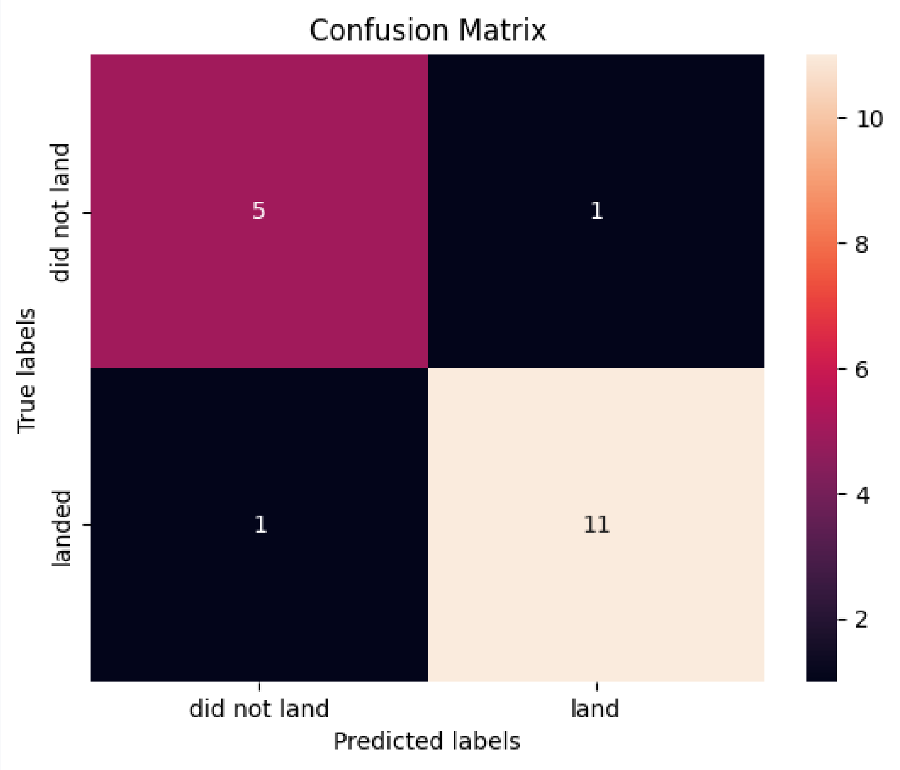


Results

These are some of the most important results obtained of each section:

Predictive analysis results:

Of the 4 models tested, decision tree classifier obtained the best results with a score of 0.89 and a good distribution of Truth table predictions on test data.



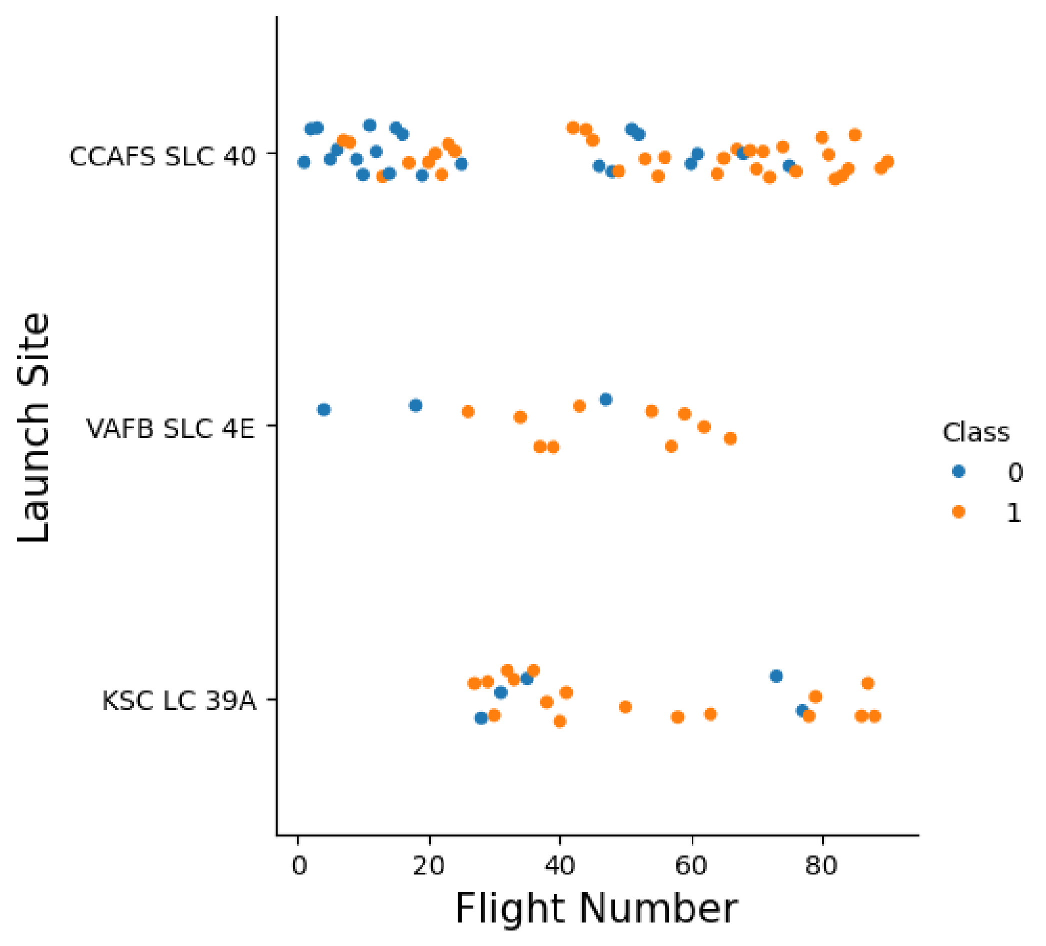

```
In [87]: print("tuned hpyerparameters :(best parameters) ",tree_cv.best_params_)
         print("accuracy :",tree_cv.best_score_)

tuned hpyerparameters :(best parameters)  {'criterion': 'gini', 'max_depth': 2, 'max_features': 'sqrt', 'min_samples_leaf': 4
, 'min_samples_split': 5, 'splitter': 'best'}
accuracy : 0.8892857142857145
```

Section 2

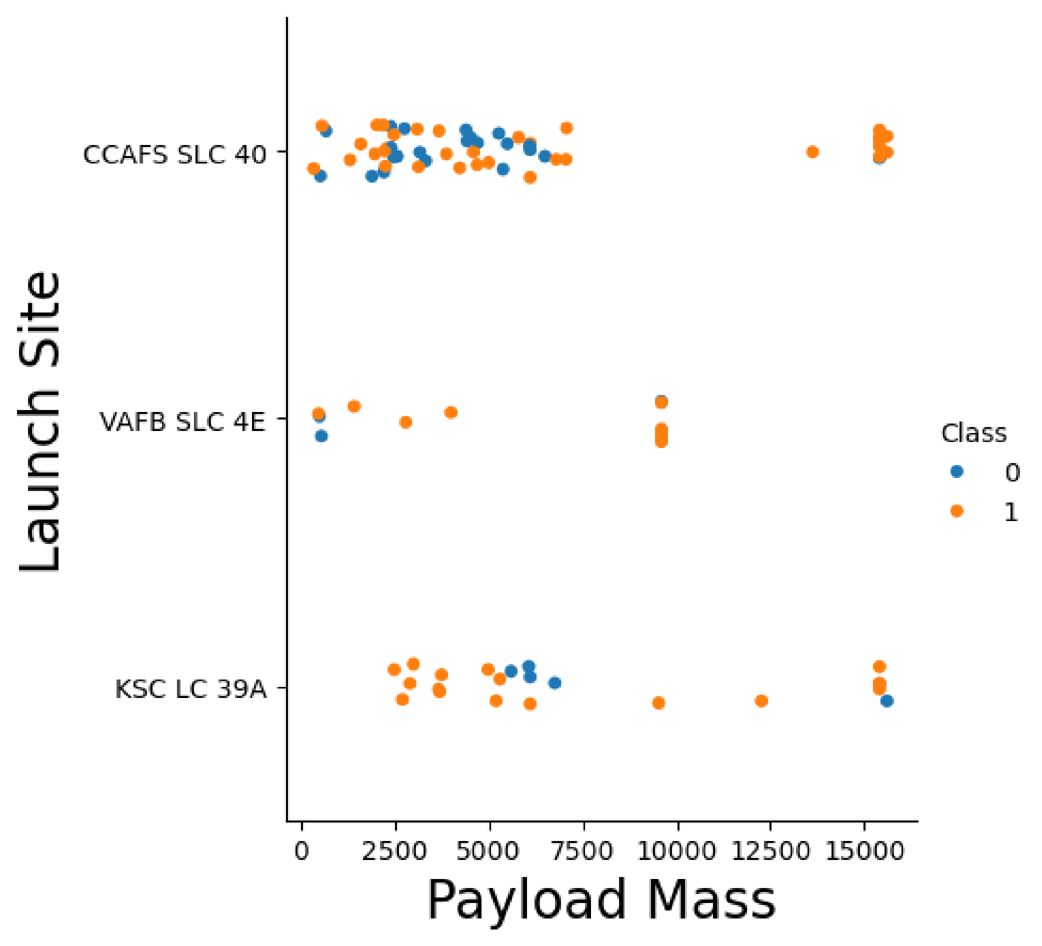
Insights drawn from EDA

Flight Number vs. Launch Site

By visualizing the relationship between flight number and launch site, we can observe that the first launches done by SpaceX was on Cape Canaveral and that most of their failures were done there.



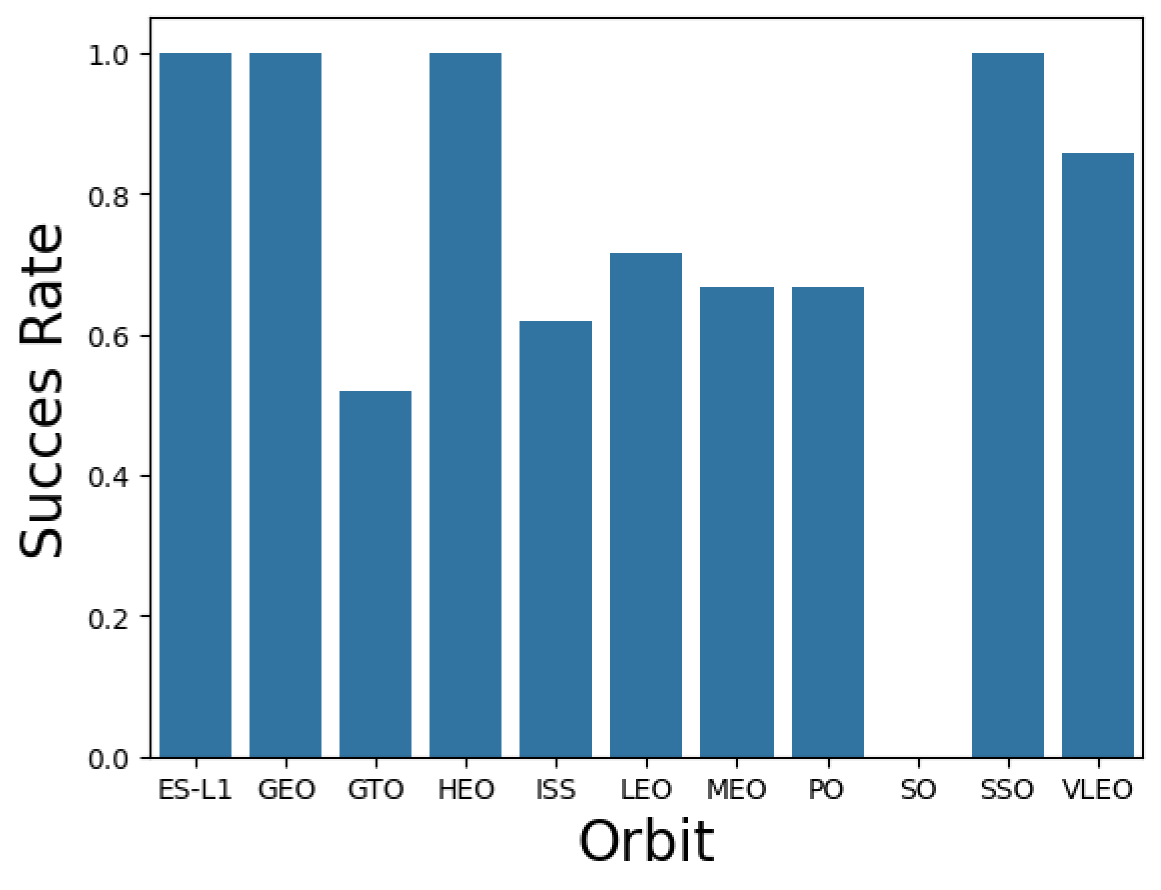
Payload vs. Launch Site



And by plotting against payload mass, we can see that for the VAFB-SLC launch site no heavy payload rockets were launched.

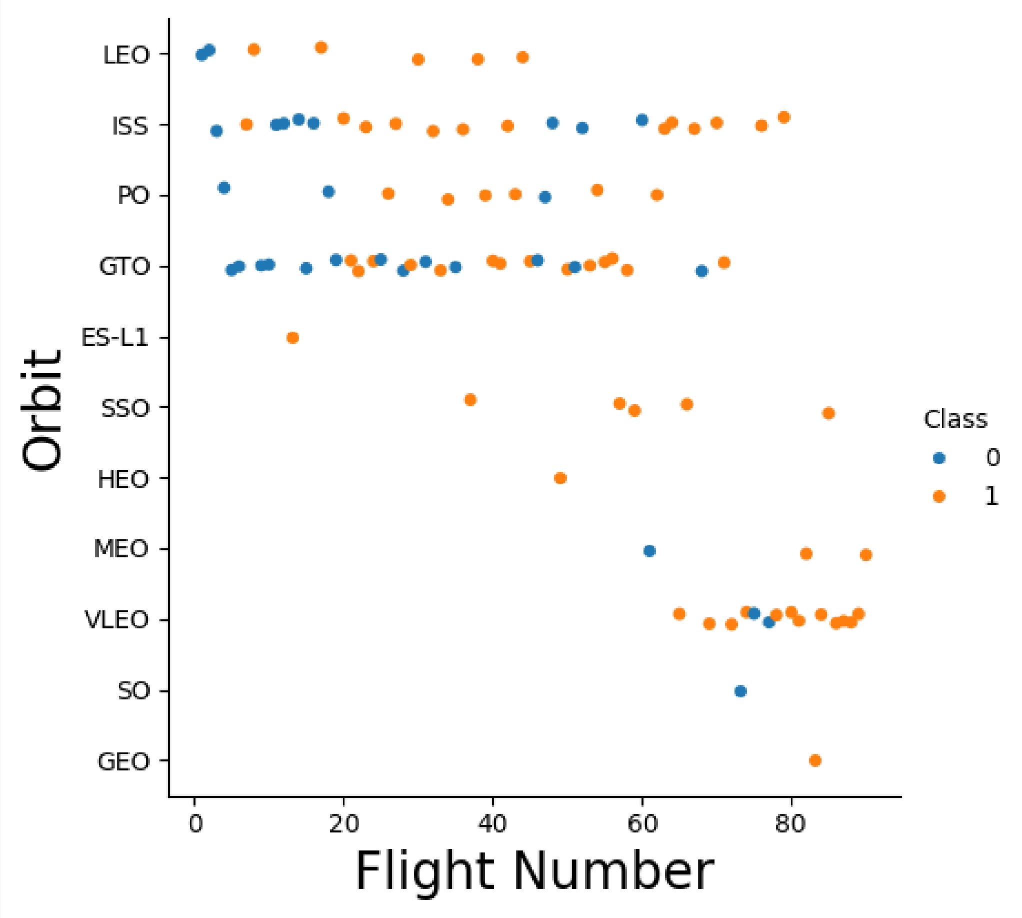
Success Rate vs. Orbit Type

By plotting the success rate of each orbit, we can see that Geo Stationary Orbits (GTO) are the most difficult launches to land.



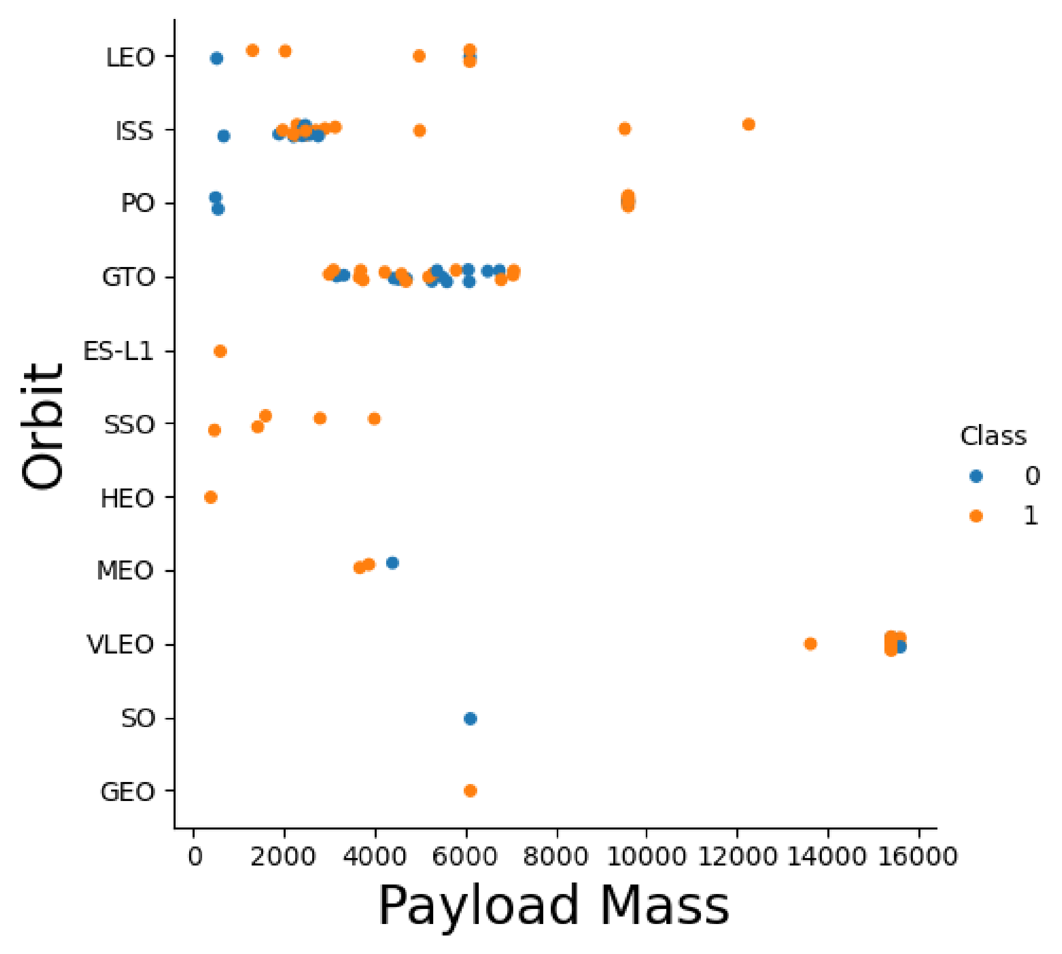
Flight Number vs. Orbit Type

As of now the most common mission by SpaceX is Very Low Earth Orbit (VLEO). This orbit is most common for satellite missions like their Starlink service



Edgar HTT

Payload vs. Orbit Type

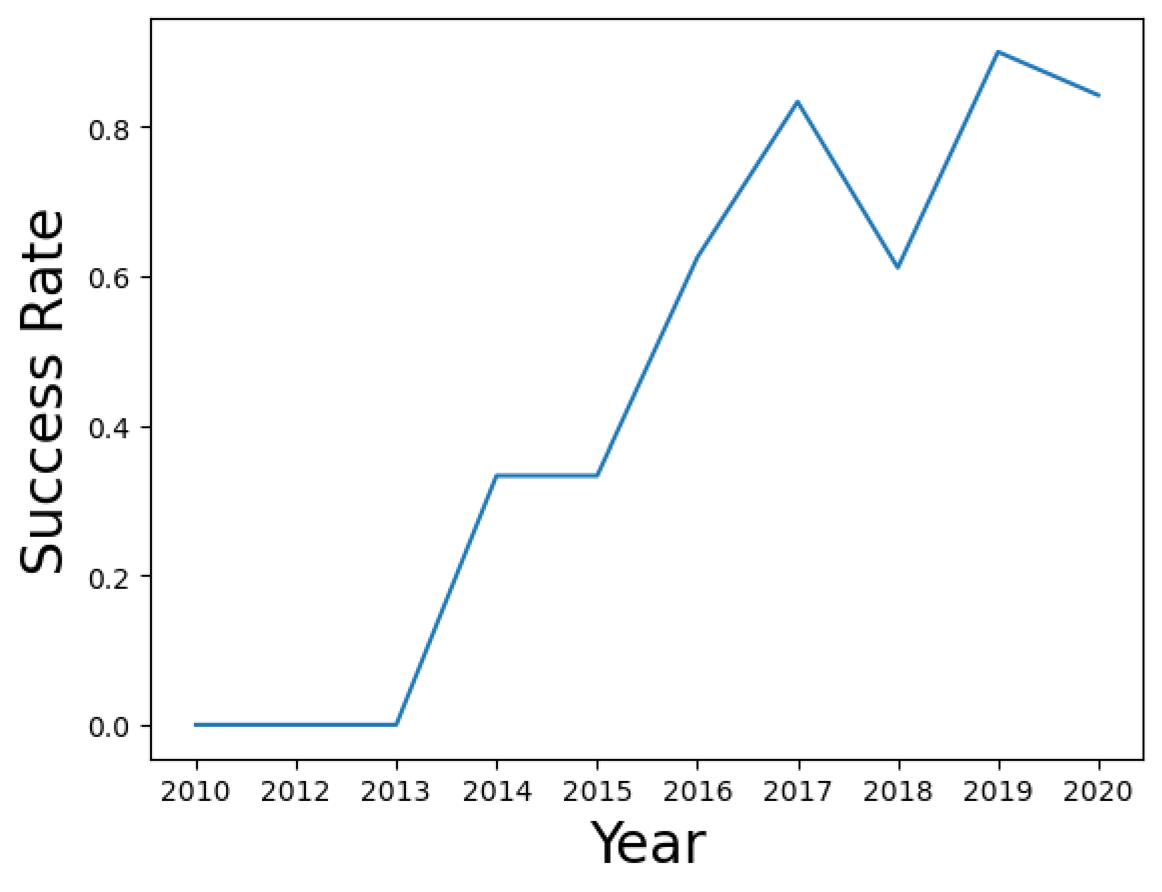


With heavy payloads the successful landing rate is increased for Polar, LEO and ISS.

However, for GTO, it’s difficult to distinguish between successful and unsuccessful landings

Launch Success Yearly Trend

Finally, we can observe that the success rate kept increasing from 2013 till 2020



All Launch Site Names

%sql **SELECT DISTINCT("Launch_Site") FROM SPACEXTABLE**

- Displays the name of the unique launch sites

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

Launch Site Names Begin with 'CCA'

```
%%sql
SELECT * FROM SPACEXTABLE
   WHERE "Launch_Site" LIKE 'CCA%'
   LIMIT 5
```

- Displays 5 records where launch sites begin with “CCA”

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

```
%%sql
SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTABLE
   WHERE Customer = 'NASA (CRS)’
```

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

SUM(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

```
%%sql
SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTABLE
   WHERE "Booster_Version" LIKE "F9 v1.1%“
```

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

AVG(PAYLOAD_MASS__KG_)
2534.6666666666665

First Successful Ground Landing Date

```
%%sql
SELECT MIN(DATE) FROM SPACEXTABLE
   WHERE Landing_Outcome = "Success (ground pad)"
```

- Date of the first successful landing outcome in ground pad

MIN(DATE)
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%%sql
SELECT Booster_Version FROM SPACEXTABLE
   WHERE Landing_Outcome = "Success (drone ship)" AND
PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000
```

- Lists the names of the boosters which have successfully landed on a drone ship and had a mass between 4000 and 6000 kgA

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

Total Number of Successful and Failure Mission Outcomes

```
%%sql
SELECT Landing_Outcome, COUNT(Landing_Outcome) FROM SPACEXTABLE
  GROUP BY Landing_Outcome LIKE "Success%" OR "Failure%“
```

- List the total number of successful and failure mission outcomes (note that the names between parenthesis do not indicate the class of outcome)

Landing_Outcome	COUNT(Landing_Outcome)
Failure (parachute)	40
Success (ground pad)	61

Boosters Carried Maximum Payload

```
%%sql
SELECT Booster_Version FROM SPACEXTABLE
   WHERE PAYLOAD_MASS__kg_ =
      (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE)
   GROUP BY Booster_Version
```

- Lists all the booster versions that have carried the maximum payload

Booster_Version
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

```
%%sql
SELECT SUBSTR(DATE,6,2), Landing_Outcome, Booster_Version, Launch_Site FROM SPACEXTABLE
   WHERE SUBSTR(DATE,0,5) = '2015' AND Landing_Outcome = 'Failure (drone ship)’
```

- List and displays month number, landing outcome, booster version and launch site of rockets launched on the year 2015 which ended on a failure to land on a drone ship

SUBSTR(DATE,6,2)	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

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

```
%%sql
SELECT Landing_Outcome, COUNT(Landing_Outcome) FROM SPACEXTABLE
   WHERE DATE BETWEEN "2010-06-04" AND "2017-03-20"
   GROUP BY Landing_Outcome
   ORDER BY COUNT(Landing_Outcome) DESC
```

- Ranks by count of landing outcomes between the date of 2010-06-04 and 2017-03-20 , in descending order.

Landing_Outcome	COUNT(Landing_Outcome)
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

Section 3

Launch Sites Proximities Analysis

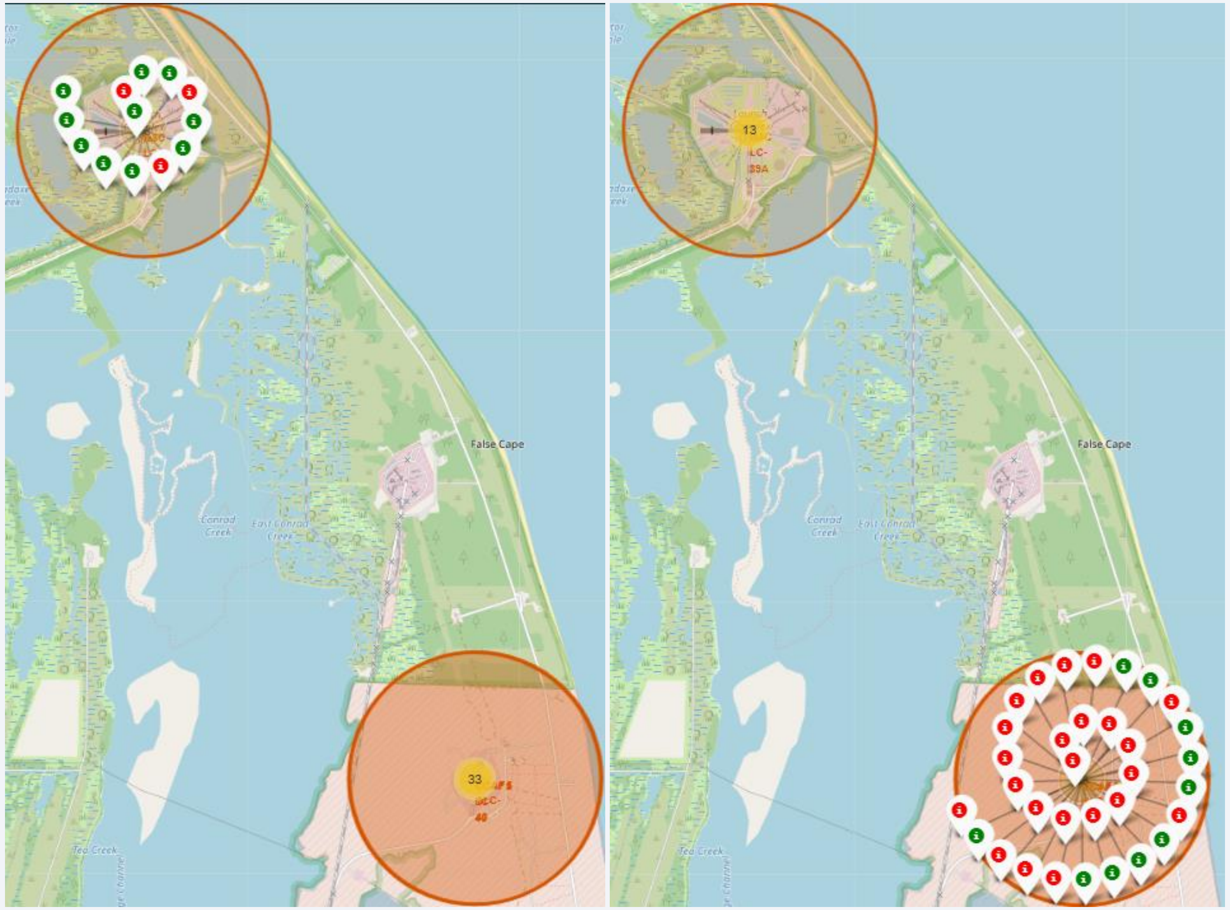
SpaceX launch sites

The three launch sites, are shown on the us map. These include the Cape Canaveral locations and Vandenberg space launch complex

Edgar HTT

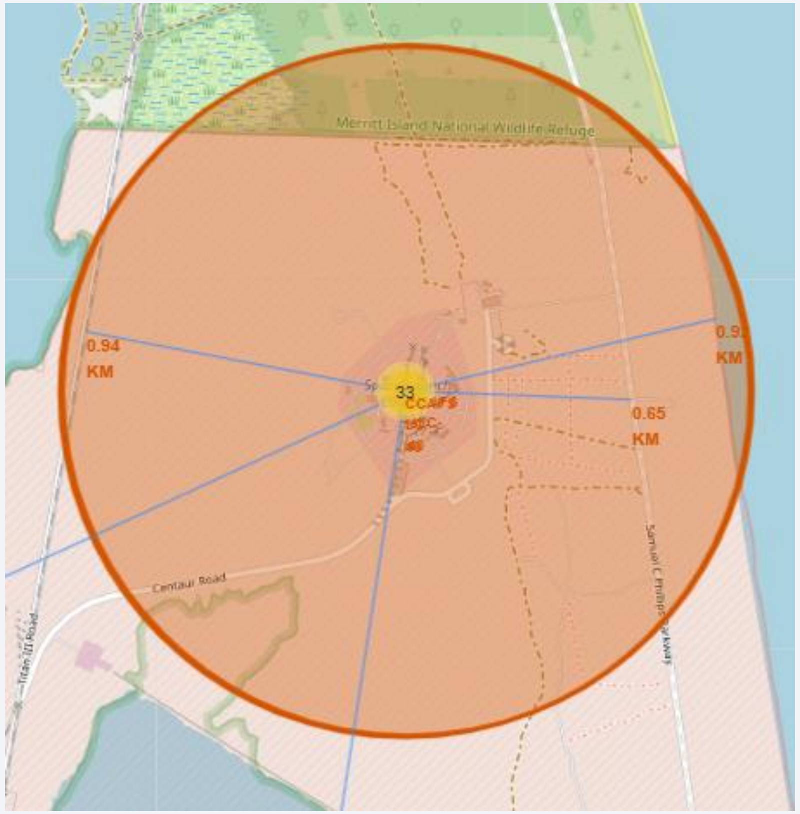
Landing Sites Data Clusters

Each location contains the information of failures and successes in a cluster form. Clicking on them reveals all landings.



Launch Sites Proximities

Points of interest were selected for each launch site and their distance calculated and annotated on the global map. The points of interest selected were highways, railways, coastlines and major cities. The main pattern found was the position of the launch site and the direction of the main major city was the determining factor of the type of orbit the launch site can launch.



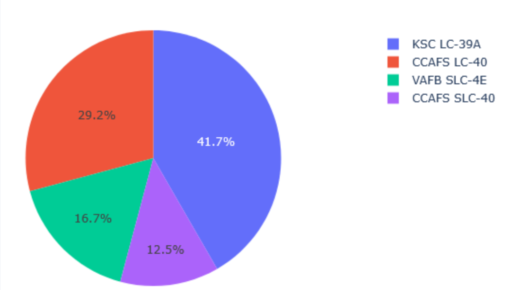
Edgar HTT

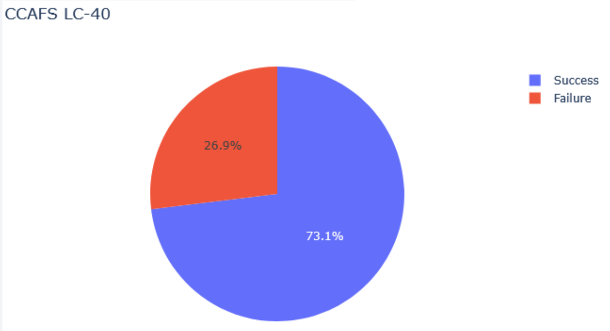
Section 4

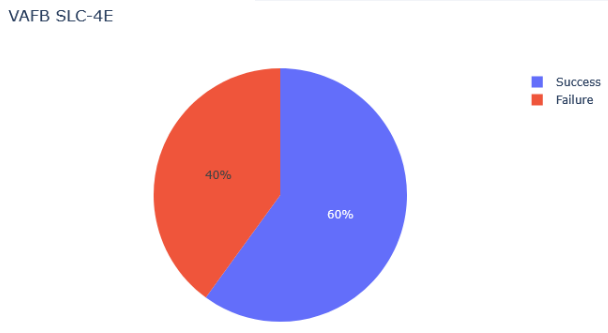
Build a Dashboard with Plotly Dash

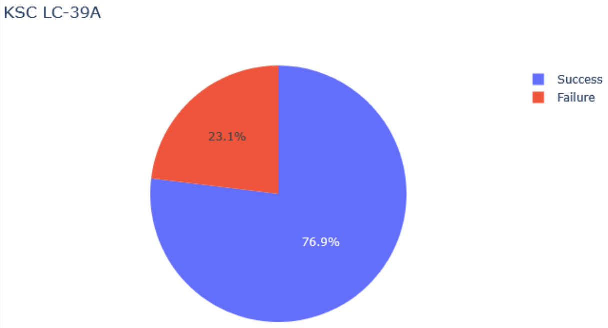
Share of Launches and Rate of Successful Landings

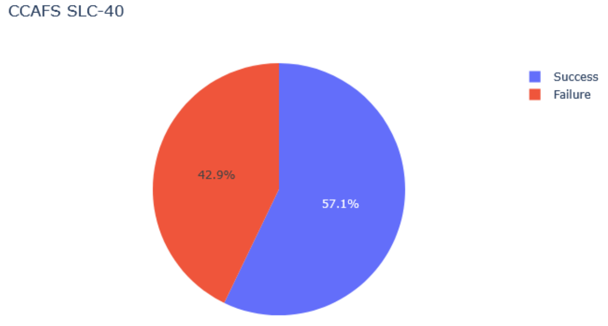
The interactive pie chart showcases the share of launches per site and when selected we can visualize their success rate







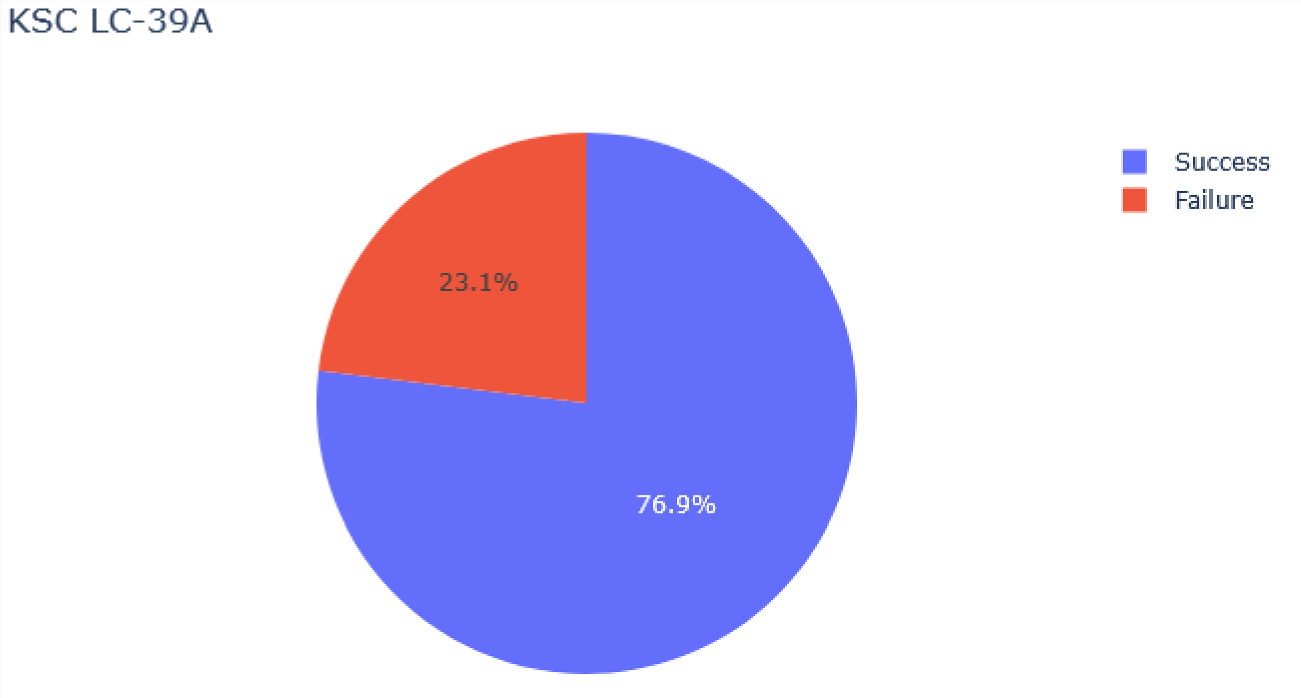




Edgar HTT

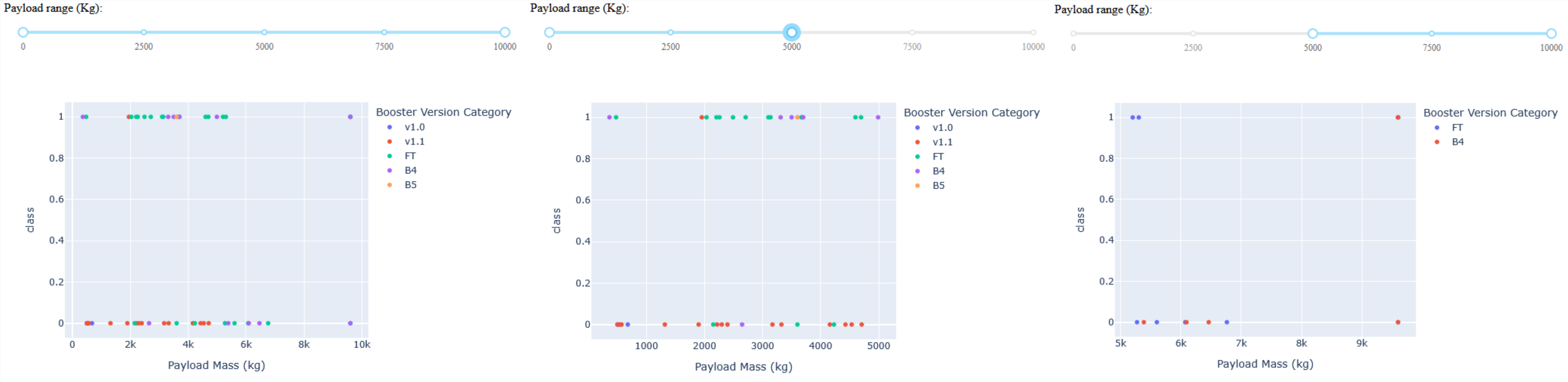
Most Successful Launch Site

Thanks to the interactive quality of the developed pie chart, we can identify with ease the most successful site.



Payload vs. Launch Outcome

The dashboard is also capable to showcase payload vs. launch outcome. We can filter by payload mass using the slider. The scatter plot also considers the launch site selected. We can observe that for launches above 5000 kg the success rate is drastically lower.

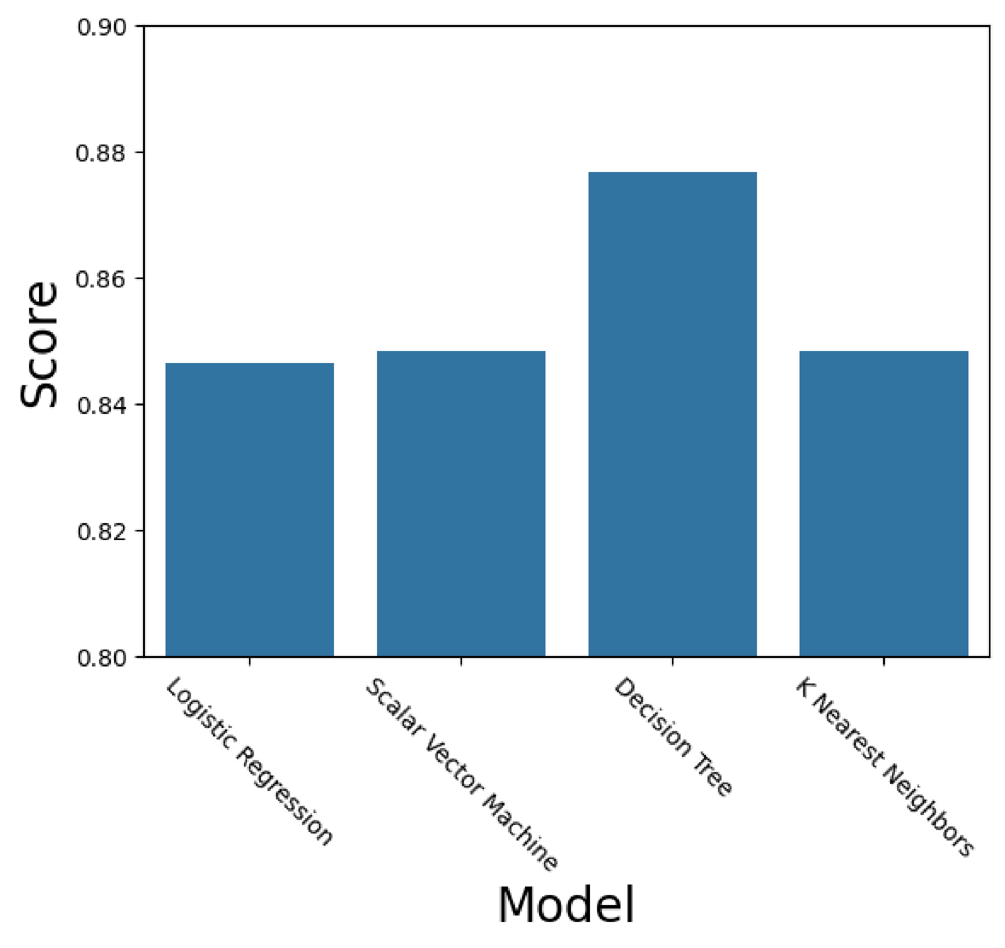


Section 5

Predictive Analysis (Classification)

Classification Accuracy

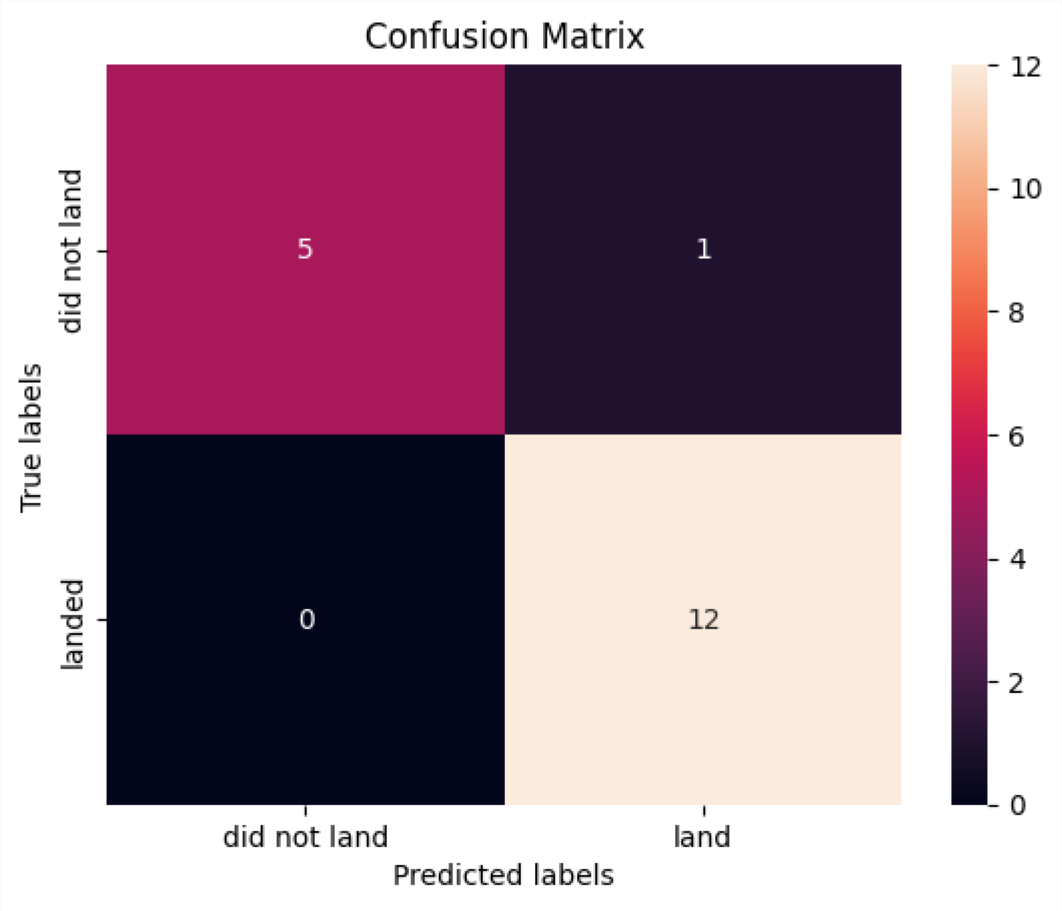
- The model with the highest classification accuracy was shown to be the decision tree classifier.



Edgar HTT

Confusion Matrix

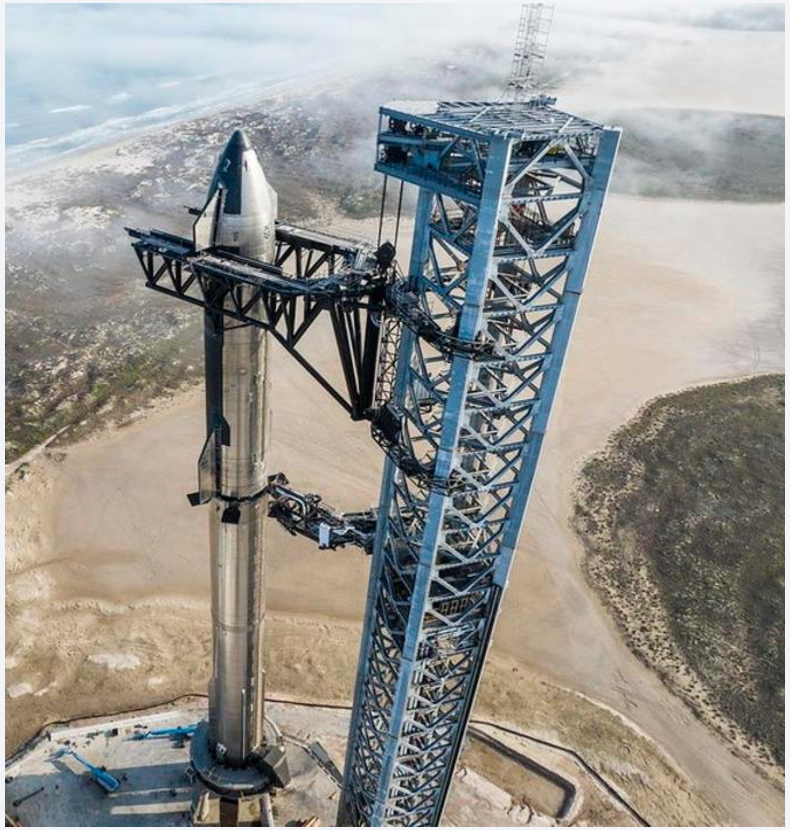
- While the most accurate model obtained was the decision tree model, these are known to be unstable models, so their results may vary substantially depending on the initial test data used.



Edgar HTT

Conclusions

- SpaceX landing success rate has been increasing since 2013 to above 80% by 2023.
- The success rate is dependent on the difficulty of the mission. Geostationary Transfer Orbit, being the most difficult to land.
- Launch site location and proximities depends on the mission to be performed.
- Landing outcomes can be predicted up to an 88% of accuracy with a decision tree model.



Appendix

- A small piece of code was developed to speed up the calculation and annotation of each point of interest to the nearest launch site on the map.

```python
def closest_launch_site(point):
    """
    Finds closest launch site and returns it's name
    """

    min_distance_site = str()
    min_distance = 10000

    for name in launch_site_cors:
        distance = calculate_distance(
            launch_site_cors[name]['lat'], launch_site_cors[name]['long'],
            point['lat'], point['long'])

        if distance < min_distance:
            min_distance = distance
            min_distance_site = name

    return min_distance_site


for point in points_interest_cors.values():

    # Find closest launch site
    launch_site_name = closest_launch_site(point)

    # Calculate distance
    distance = calculate_distance(
        launch_site_cors[launch_site_name]['lat'],
        launch_site_cors[launch_site_name]['long'],
        point['lat'],
        point['long']
    )

    # Add distance marker
    folium.Marker(
        (point['lat'], point['long']),
        icon=DivIcon(
            icon_size=(20,20),
            icon_anchor=(0,0),
            html=f'<div style="font-size: 12px; color:#d35400;"><b>{distance:10.2f} KM</b></div>'
        )
    ).add_to(site_map)

    # Add connecting line
    coordinates = [
        (launch_site_cors[launch_site_name]['lat'], launch_site_cors[launch_site_name]['long']),
        (point['lat'], point['long'])
    ]
    folium.PolyLine(locations=coordinates, weight=1).add_to(site_map)

site_map
```

Thank you!