IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Juan Pablo Retamal
2024-11-15

Outline

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

Executive Summary

In this capstone, we will predict if the Falcon 9 first stage will land successfully.

To answer the central question, the following is used:

- Data collection and data wrangling.
- Exploratory Analysis Using SQL, Exploratory Analysis Using Pandas and Matplotlib.
- Interactive Visual Analytics with Folium, Build an Interactive Dashboard with Ploty Dash.
- Predictive Analysis (Classification) using Machine learning Prediction.

Summary of results:

- **Data collection and data wrangling:** Request to the SpaceX API, Clean the requested data. Web scrap Falcon 9 launch records with `BeautifulSoup`
- **Exploratory Data Analysis:** Orbits with the highest success rates: ES-L1, SSO, HEO, GEO. Launch success yearly trend: the sucess rate since 2013 kept increasing until 2020. In the LEO orbit, success seems to be related to the number of flights.
- **Interactive Visual Analytics and Interactive Dashboard:** Launch sites are near to the Equator line and coastline.
- **Predictive Analysis (Classification) using Machine learning Prediction:** The method performs best: Decision tree.

	Best scores
Logistic regresssion	0.846429
SVM	0.848214
Decision tree	0.885714
KNN	0.848214

Introduction

SpaceX advertises Falcon 9 rocket launches on its website, with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage. Therefore if we can determine if the first stage will land, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch.

Falcon 9 first stage will land successfully?: According to the study conducted to answer this question, It is necessary to divide the data into training data and test data, to find the best hyperparameter for SVM, classification trees and logistic regression. The method that works best with the test data is then found.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
    - Collect and Clean the requested data using SpaceX API and Web scraping.
- Perform data wrangling
    - Web scrap Falcon 9 launch records with `BeautifulSoup`
        - Extract a Falcon 9 launch records HTML table from Wikipedia
        - Parse the table and convert it into a Pandas data frame
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
    - Create a machine learning pipeline to predict if the first stage will land, tune, evaluate classification models. Find best Hyperparameter for SVM, Classification Trees and Logistic Regression.

Data Collection

- Collect and Clean the requested data using SpaceX API and Web scraping.
- Stages of the process:
  - Collect and Clean the requested data using SpaceX API:
    - ➢ Request and parse the SpaceX launch data using the GET request.
    - ➢ Filter the dataframe to only include `Falcon 9` launches.
    - ➢ Data Wrangling: Dealing with Missing Values.
  - Web scrap Falcon 9 launch records with `BeautifulSoup`:
    - ➢ Request the Falcon9 Launch Wiki page from its URL.
    - ➢ Extract all column/variable names from the HTML table header.
    - ➢ Create a data frame by parsing the launch HTML tables.

Data Collection – SpaceX API

- Request to the SpaceX API: URLs used:

  https://api.spacexdata.com/v4/rockets/

  https://api.spacexdata.com/v4/launchpads/

  https://api.spacexdata.com/v4/payloads/

  https://api.spacexdata.com/v4/cores/

  https://api.spacexdata.com/v4/launches/past
- Filter the data to only include `Falcon 9` launches.
- Calculate the mean for the PayloadMass, use the mean to replace missing values then the number of missing values of the PayLoadMass change to zero.
- There is not missing values in the dataset, except for in LandingPad:

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	Longitude	Latitude
4	1	2010-06-04	Falcon 9	6123.547647	LEO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0003	-80.577366	28.561857
5	2	2012-05-22	Falcon 9	525.000000	LEO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0005	-80.577366	28.561857
6	3	2013-03-01	Falcon 9	677.000000	ISS	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0007	-80.577366	28.561857
7	4	2013-09-29	Falcon 9	500.000000	PO	VAFB SLC 4E	False Ocean	1	False	False	False	None	1.0	0	B1003	-120.610829	34.632093
8	5	2013-12-03	Falcon 9	3170.000000	GTO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B1004	-80.577366	28.561857
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
89	86	2020-09-03	Falcon 9	15600.000000	VLEO	KSC LC 39A	True ASDS	2	True	True	True	5e9e3032383ecb6bb234e7ca	5.0	12	B1060	-80.603956	28.608058
90	87	2020-10-06	Falcon 9	15600.000000	VLEO	KSC LC 39A	True ASDS	3	True	True	True	5e9e3032383ecb6bb234e7ca	5.0	13	B1058	-80.603956	28.608058
91	88	2020-10-18	Falcon 9	15600.000000	VLEO	KSC LC 39A	True ASDS	6	True	True	True	5e9e3032383ecb6bb234e7ca	5.0	12	B1051	-80.603956	28.608058
92	89	2020-10-24	Falcon 9	15600.000000	VLEO	CCSFS SLC 40	True ASDS	3	True	True	True	5e9e3033383ecbb9e534e7cc	5.0	12	B1060	-80.577366	28.561857
93	90	2020-11-05	Falcon 9	3681.000000	MEO	CCSFS SLC 40	True ASDS	1	True	False	True	5e9e3032383ecb6bb234e7ca	5.0	8	B1062	-80.577366	28.561857

90 rows × 17 columns

- Export data to csv file.
- GitHub URL of the completed SpaceX API calls notebook: https://github.com/JPCLX/Capstone/blob/main/01.-%20Data%20collection%20API_JPR.ipynb

Data Collection - Scraping

- **Web scraping Falcon 9 and Falcon Heavy Launches Records from Wikipedia.**
- Web scraping to collect Falcon 9 historical launch records from a Wikipedia page titled `List of Falcon 9 and Falcon Heavy launches:

  https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches
- Request the Falcon9 Launch Wiki page from its URL
- Extract all column/variable names from the HTML table header.
- Create a data frame by parsing the launch HTML tables.
- After fill in the parsed launch record values into `launch_dict` and create a dataframe from it. The result is:

	Flight No.	Launch site	Payload	Payload mass	Orbit	Customer	Launch outcome	Version Booster	Booster landing	Date	Time
0	1	CCAFS	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success\n	F9 v1.07B0003.18	Failure	4 June 2010	18:45
1	2	CCAFS	Dragon	0	LEO	NASA	Success	F9 v1.07B0004.18	Failure	8 December 2010	15:43
2	3	CCAFS	Dragon	525 kg	LEO	NASA	Success	F9 v1.07B0005.18	No attempt\n	22 May 2012	07:44
3	4	CCAFS	SpaceX CRS-1	4,700 kg	LEO	NASA	Success\n	F9 v1.07B0006.18	No attempt	8 October 2012	00:35
4	5	CCAFS	SpaceX CRS-2	4,877 kg	LEO	NASA	Success\n	F9 v1.07B0007.18	No attempt\n	1 March 2013	15:10
...	...	...	...	...	...	...	...	...	...	...	...
237	117	CCSFS	Starlink	15,600 kg	LEO	SpaceX	Success\n	F9 B5B1051.10657	Success	9 May 2021	06:42
238	118	KSC	Starlink	~14,000 kg	LEO	SpaceX	Success\n	F9 B5B1058.8660	Success	15 May 2021	22:56
239	119	CCSFS	Starlink	15,600 kg	LEO	SpaceX	Success\n	F9 B5B1063.2665	Success	26 May 2021	18:59
240	120	KSC	SpaceX CRS-22	3,328 kg	LEO	NASA	Success\n	F9 B5B1067.1668	Success	3 June 2021	17:29
241	121	CCSFS	SXM-8	7,000 kg	GTO	Sirius XM	Success\n	F9 B5	Success	6 June 2021	04:26

242 rows × 11 columns

- Export data to csv file.
- GitHub URL of the completed web scraping notebook. https://github.com/JPCLX/Capstone/blob/main/02.-%20Data%20collection%20Webscraping_JPR.ipynb

Data Wrangling

- Perform some Exploratory Data Analysis (EDA) to find some patterns in the data and determine what would be the label for training supervised models.
- In the data set, there are several different cases where the booster did not land successfully. Sometimes a landing was attempted but failed due to an accident; for example, True Ocean means the mission outcome was successfully landed to a specific region of the ocean while False Ocean means the mission outcome was unsuccessfully landed to a specific region of the ocean. True RTLS means the mission outcome was successfully landed to a ground pad False RTLS means the mission outcome was unsuccessfully landed to a ground pad True ASDS means the mission outcome was successfully landed on a drone ship False ASDS means the mission outcome was unsuccessfully landed on a drone ship.
- The objective is mainly converting those outcomes into Training Labels with `1` means the booster successfully landed `0` means it was unsuccessful.
- Steps:
  - ➢ Calculate the number of launches on each site.
  - ➢ Calculate the number and occurrence of each orbit.
  - ➢ Calculate the number and occurence of mission outcome of the orbits.
  - ➢ Create a landing outcome label from Outcome column.
  - ➢ Export data to csv file.
- GitHub URL of the completed data wrangling related notebooks.
  https://github.com/JPCLX/Capstone/blob/main/03.-%20Data%20Wrangling_JPR.ipynb

EDA with SQL

SQL Queries

- **Display:**
  - ➢ Names of unique launch sites in the space mission.
  - ➢ 5 records where launch site begins with string ‘CCA’.
  - ➢ Total payload mass carried (Kg) by boosters launched by NASA (CRS).
  - ➢ Average payload mass (Kg) carried by booster version F9 v1.1.
- **List:**
  - ➢ Date when the first successful landing outcome in ground pad was achieved.
  - ➢ Names of boosters which had success landing on drone ship and have payload mass (kg) greater than 4,000 but less than 6,000
  - ➢ Total number of successful and failure mission outcomes.
  - ➢ Names of the booster versions which have carried the maximum payload mass.
  - ➢ Records which will display the month names, failure landing outcomes in drone ship ,booster versions, launch site for the months in year 2015.
  - ➢ Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order.
- GitHub URL of the completed EDA with SQL notebook. https://github.com/JPCLX/Capstone/blob/main/04.-%20EDA%20SQL_JPR.ipynb

EDA with Data Visualization

- **Charts.**
  - Flight Number vs. Payload Mass (Kg).
  - Flight Number vs. Launch Site.
  - Payload Mass (kg) vs. Launch Site.
  - Orbit vs Success rate.
  - Flight Number vs Orbit.
  - Payload Mass (kg) vs. Orbit type
  - Launch success yearly trend.

- **Analysis.**
  - Scatter plots show the relationship between variables. If a relationship exists,
  - they could be used in machine learning model.
  - Show comparisons among discrete categories with and relationships among
  - the categories and a measured value with Bar charts.
  - Line charts show trends in data over time (time series).

- GitHub URL of the completed EDA with datavisualization notebook. https://github.com/JPCLX/Capstone/blob/main/05.-%20EDA%20Data%20Visualization_JPR.ipynb

Build an Interactive Map with Folium

- **Markers of Launch Sites, using Folium:**
  - ➢ Circle at NASA Johnson Space Center's coordinate, using latitude and longitude, with a popup label showing its name.
  - ➢ Circle object based on its coordinate (Lat, Long) values. In addition, add Launch site name as a popup label. For each launch site.
- **Colored Markers of Launch Outcomes.**
  - ➢ Colored markers of launches at each site to show success (green) and fail (red).
- **Distances Between a Launch Site to Proximities.**
  - ➢ Distance from the launch site KSC LC-39A to railway.
  - ➢ Distance from the launch site KSC LC-39A to highway.
  - ➢ Distance from the launch site KSC LC-39A to coastline.
  - ➢ Distance from the launch site KSC LC-39A to Titusville City.

GitHub URL of the completed interactive map with Folium map. https://github.com/JPCLX/Capstone/blob/main/06.-%20Interactive%20Visual%20Analytics%20with%20Folium_JPR.ipynb

Build a Dashboard with Plotly Dash

- **Launch Site Dropdown list.**
  - There are four different launch sites, and you must first see which one has the highest number of successes. Next, you select a specific site and check its detailed success rate (class=0 vs. class=1).To do this, a drop-down menu is used to select the different launch sites.
- **Pie-chart based on selected site dropdown.**
  - Selected launch site from site dropdown list and show a pie chart, visualizing launch success counts for al sites.
  - If a specific launch site is selected, include the only data for the selected site. Then, render and return a pie chart graph to show the success (class=1) count and failed (class=0) count for the selected site.
- **Range Slider to Select Payload.**
  - Add a Range Slider to Select Payload.
- **Success-payload-scatter-chart scatter plot.**
  - If all sites are selected, scatter plot display all values for variable Payload Mass (kg) and variable class.
  - If a specific launch site is selected, scatter chart show values of Payload Mass (kg) and class for the selected site, and color-label the point using Booster Version Category likewise.

- GitHub URL of the completed Plotly Dash lab. https://github.com/JPCLX/Capstone/blob/main/07.-%20Spacex_dash_app_JPR.py

Predictive Analysis (Classification)

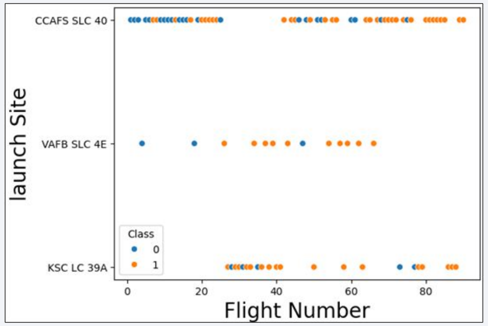
- Create a NumPy array from the column Class in data.
- Standardize the data in X then reassign it to the variable X using the transform provided below.
- Split the data into training and testing data using the function train_test_split. The training data is divided into validation data, a second set used for training data; then the models are trained and hyperparameters are selected using the function GridSearchCV.
- Use a GridSearchCV object, con cv = 10, in algorithms: Logistic regression, Support vector machine (SVM) , Decision tree, K nearest neighbors.
- Calculate the accuracy on the test data using the method score and plot the confusion matrix, for all the models.
- Finally, find the method performs best.

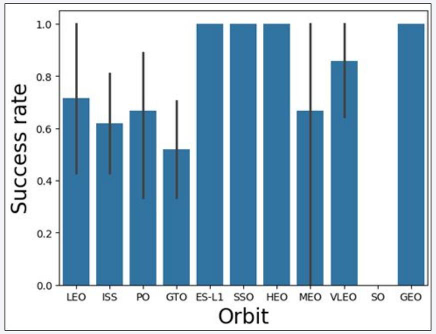
- GitHub URL of your completed predictive analysis lab. https://github.com/JPCLX/Capstone/blob/main/08.-%20Machine%20Learning%20Prediction_JPR.ipynb

Results

- Exploratory data analysis results.
  - ➢ The CCAFS SLC 40 launch site has the largest number of launches.
  - ➢ KSC LC 39A launch site and VAFB-SLC launch site have the highest proportion of successful launches.
  - ➢ Orbits with the highest success rates (100%): ES-L1, SSO, HEO, GEO.
- Predictive analysis results.
  - ➢ The method performs best: Decision tree:

	Best scores
Decision tree	0.885714



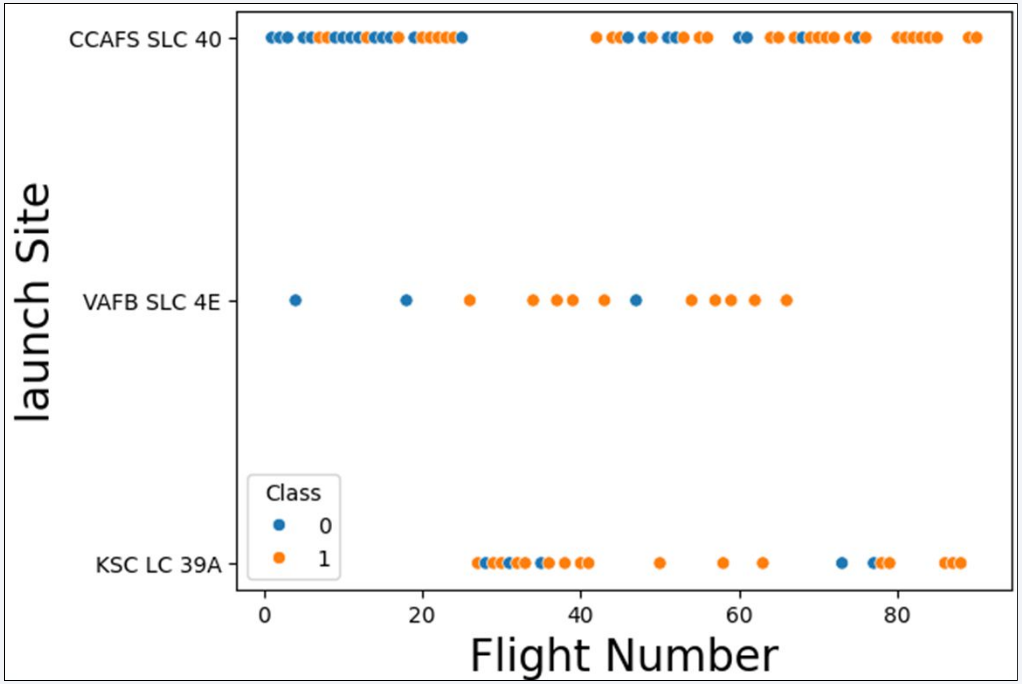


Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

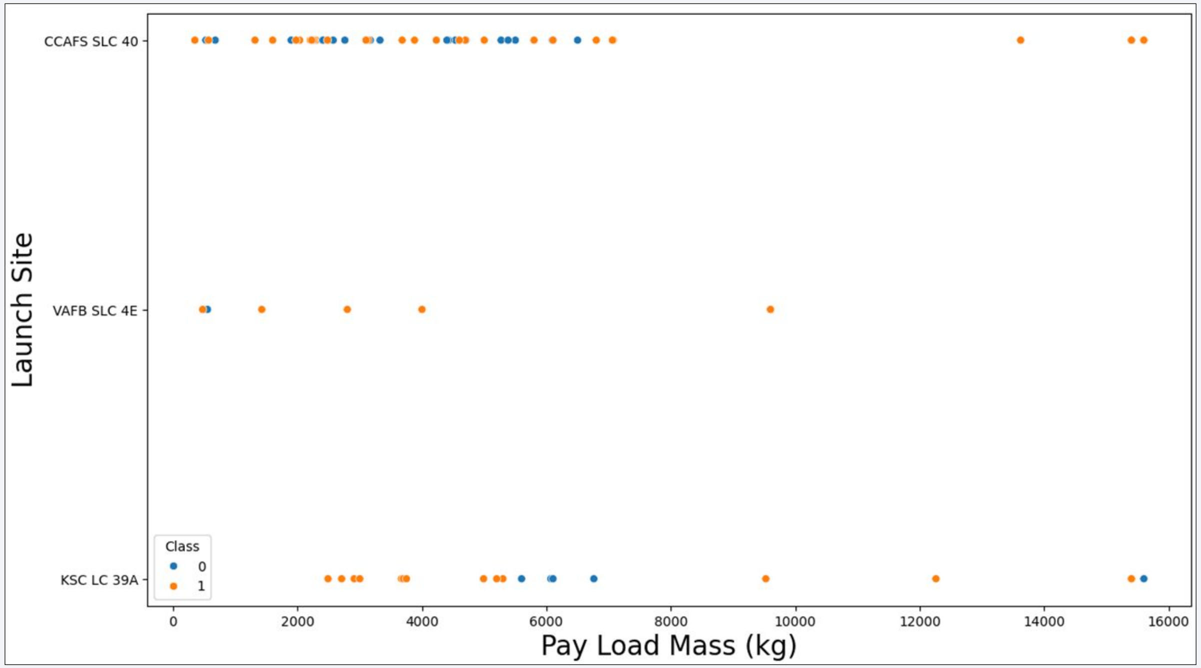
- Scatter plot of Flight Number vs. Launch Site.



- The CCAFS SLC 40 launch site has the largest number of launches.
- KSC LC 39A launch site and VAFB-SLC launch site have the highest proportion of successful launches.

Payload vs. Launch Site

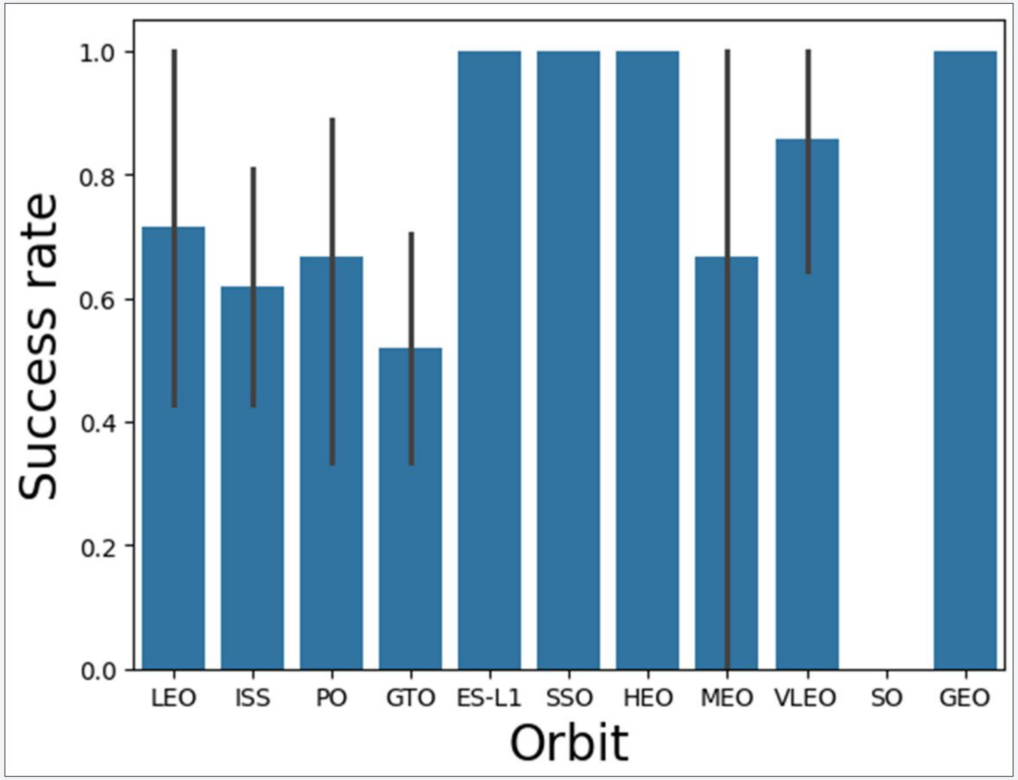
- Scatter plot of Payload vs. Launch Site.



- CCAFS SLC 40 launch site there are rockets launched for heavy payload mass (greater than 14000).
- VAFB-SLC launch site there are no rockets launched for heavy payload mass (greater than 10000 kg).
- KSC LC 39A launch site there are rockets launched for heavy payload mass (greater than 14000).

Success Rate vs. Orbit Type

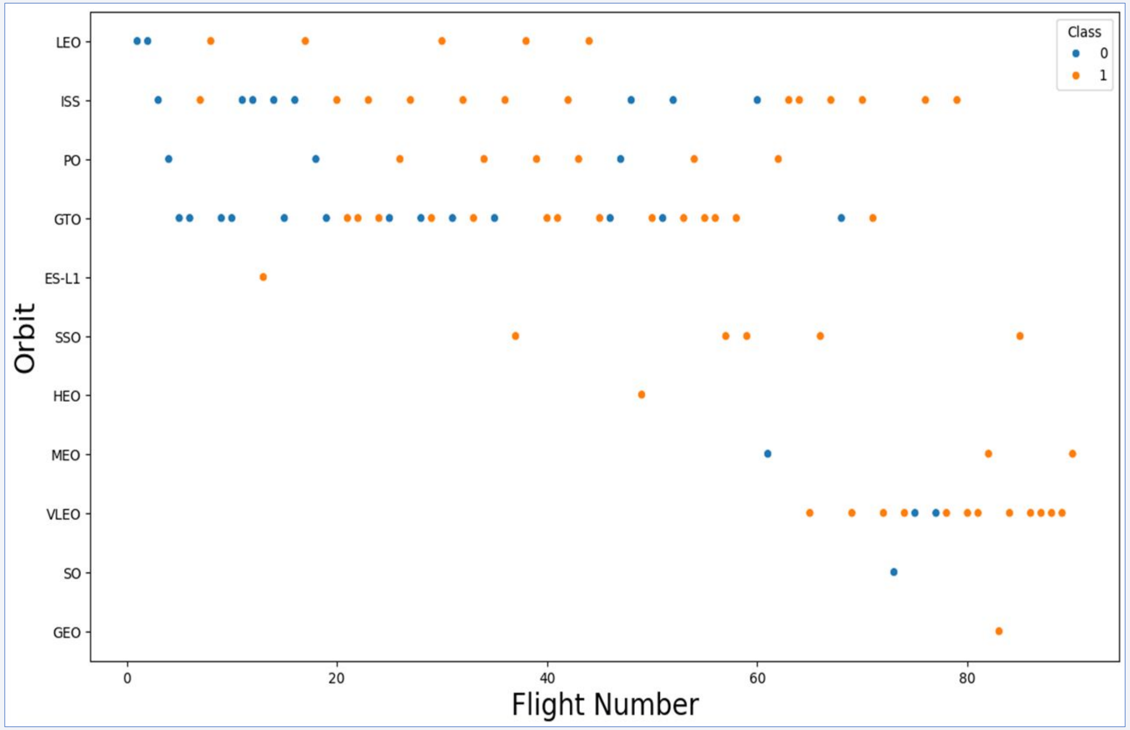
- Bar chart for the success rate of each orbit type.



- Orbits with the highest success rates (100%): ES-L1, SSO, HEO, GEO.
- Orbits with success rates between 50% and 85%: LEO, ISS, PO, GTO, MEO, VLEO.
- Orbit with 0% of success rates: SO.

Flight Number vs. Orbit Type

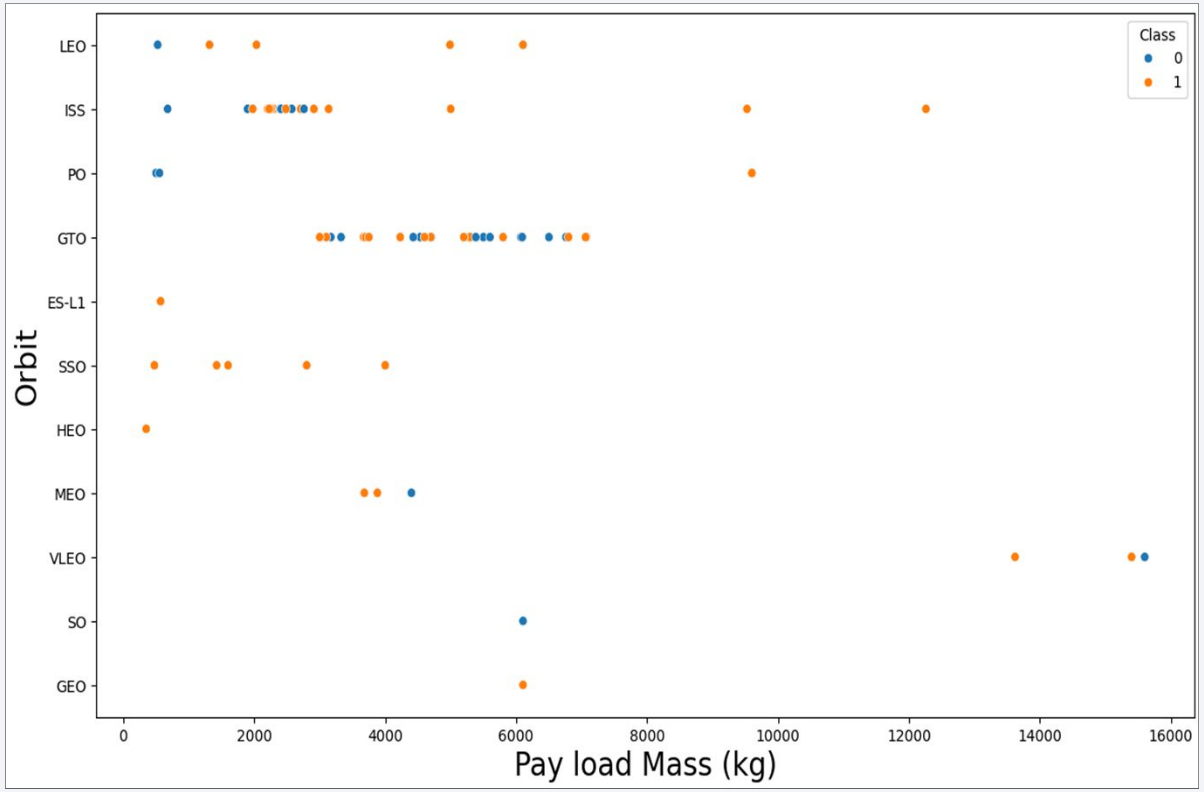
- Scatter point of Flight number vs. Orbit type.



- In the LEO orbit, success seems to be related to the number of flights. Conversely, in the GTO orbit, there appears to be no relationship between flight number and success.

Payload vs. Orbit Type

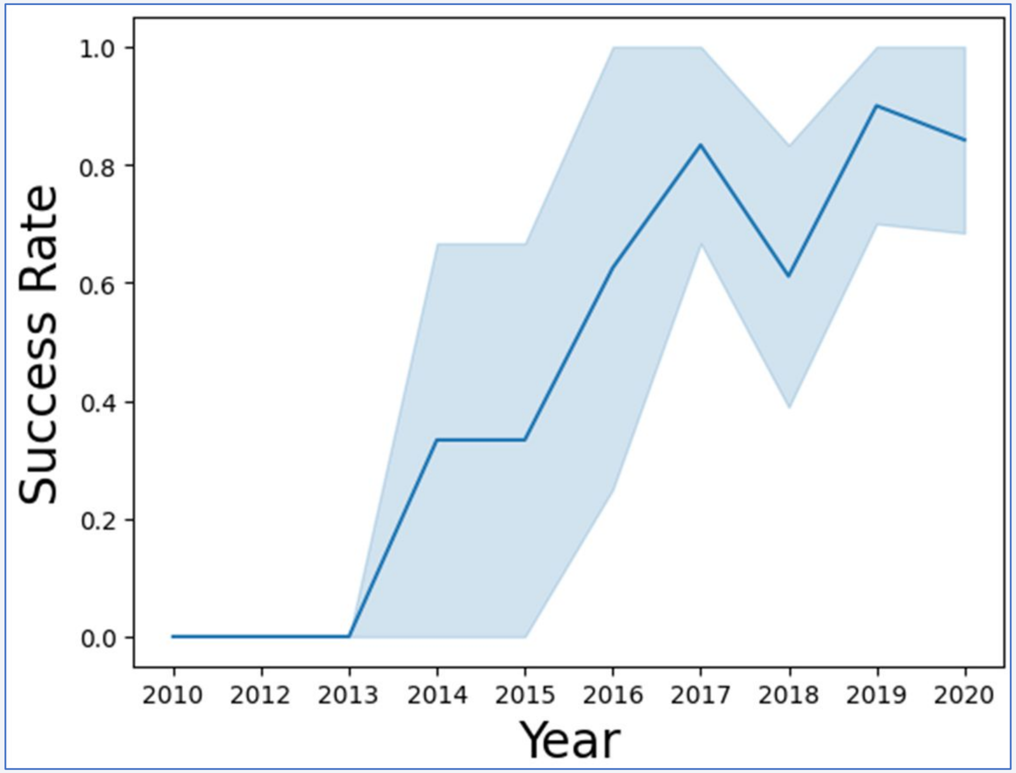
- Scatter point of payload vs. orbit type.



- With heavy payloads the successful landing or positive landing rate are more for Polar, LEO and ISS.
- However, for GTO, it's difficult to distinguish between successful and unsuccessful landings as both outcomes are present.

Launch Success Yearly Trend

- Line chart of yearly average success rate.



- The success rate since 2013 kept increasing until 2020.

All Launch Site Names

```
%sql SELECT DISTINCT LAUNCH_SITE as 'Launch_Site' FROM SPACEXTBL;
```

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

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

The names of the unique launch sites

Launch Site Names Begin with 'CCA'

- Show 5 records where launch sites begin with `CCA`

```
%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 is:

```
%sql select sum(PAYLOAD_MASS__KG_) as 'Total payload mass carried by boosters launched by NASA (CRS) (Kg)' from SPACEXTBL where customer = 'NASA (CRS)';
```

✓ 0.0s

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

Total payload mass carried by boosters launched by NASA (CRS) (Kg)
45596

Average Payload Mass by F9 v1.1

- The average payload mass carried by booster version F9 v1.1, is

```
%sql select avg(PAYLOAD_MASS__KG_) as 'Average payload mass (Kg) carried by booster version F9 v1.1' from SPACEXTBL where Booster_Version = 'F9 v1.1';
```

✓ 0.0s

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

Average payload mass (Kg) carried by booster version F9 v1.1
2928.4

First Successful Ground Landing Date

- The date of the first successful landing outcome on ground pad, is:

```
%sql select min(date) as 'Date when the first succesful landing outcome in ground pad was acheived' from SPACEXTBL WHERE Landing_Outcome = 'Success (ground pad)';
```

✓ 0.0s

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

Date when the first succesful landing outcome in ground pad was acheived
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- List of the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000, is:

```
%sql select Booster_Version as 'Names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000' from SPACEXTBL where Landing_Outcome = 'Success (drone ship)'
and PAYLOAD_MASS__KG_ between 4000 and 6000;
```

✓ 0.0s

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

Names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
F9 FT B1031.2

Total Number of Successful and Failure Mission Outcomes

- Total number of successful and failure mission outcomes, is:

```
%sql SELECT Total_number_of_successful_mission_outcomes, Total_number_of_failure_mission_outcomes FROM (SELECT COUNT(*) as Total_number_of_successful_mission_outcomes
from SPACEXTBL where Mission_Outcome like 'Success%') success_table, (SELECT COUNT(*) as Total_number_of_failure_mission_outcomes from SPACEXTBL where Mission_Outcome like 'Failure%') failure_table
```

✓ 0.0s

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

Total_number_of_successful_mission_outcomes	Total_number_of_failure_mission_outcomes
100	1

Boosters Carried Maximum Payload

- List of the names of the booster which have carried the maximum payload mass, is:

```
%sql select distinct Booster_Version as 'names of the booster_versions which have carried the maximum payload mass' from SPACEXTBL where PAYLOAD_MASS__KG_ =(select max(PAYLOAD_MASS__KG_) from SPACEXTBL);
```

✓ 0.0s

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

names of the booster_versions which have carried the maximum payload mass
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

2015 Launch Records

- List of the failed landing outcomes in drone ship, their booster versions, and launch site names for in year 2015, is:

```
%sql select substr(Date,6,2) as month, Booster_Version, Launch_Site from SPACEXTBL where Landing_Outcome = 'Failure (drone ship)' and substr(Date,0,5)='2015';
```

✓ 0.0s

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

month	Booster_Version	Launch_Site
01	F9 v1.1 B1012	CCAFS LC-40
04	F9 v1.1 B1015	CCAFS LC-40

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

- 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, in descending order, is:

```
%sql select Landing_Outcome, count(Landing_Outcome) as 'landing outcomes' from SPACEXTBL where Date Between '2010-06-04' and '2017-03-20' Group by Landing_Outcome Order by Count(Landing_Outcome) Desc;
```

✓ 0.0s

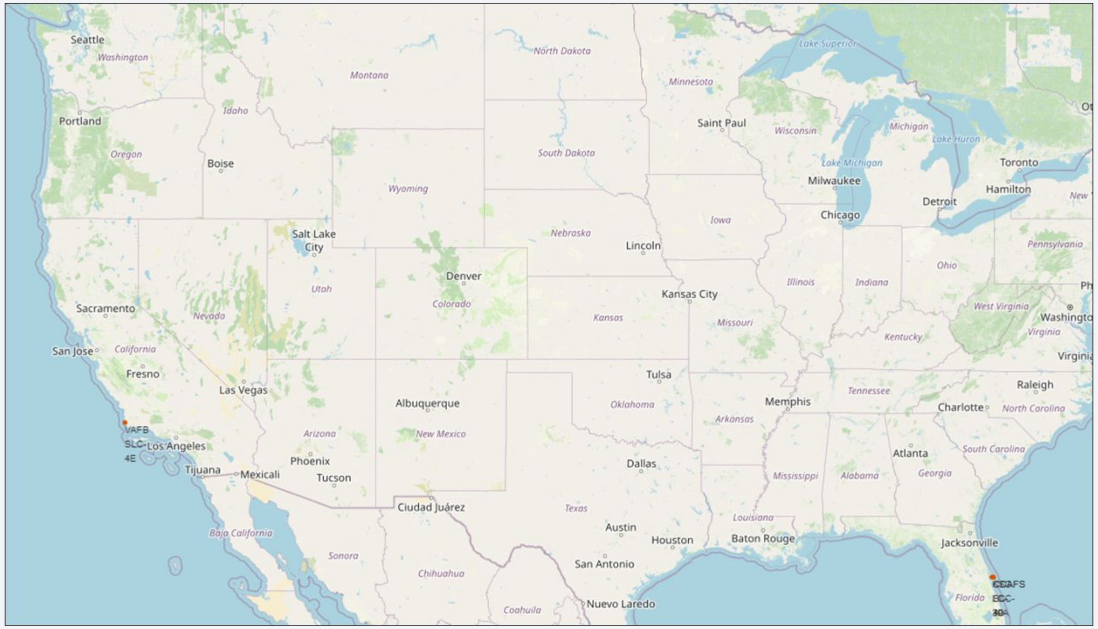
* sqlite:///my_data1.db
Done.

Landing_Outcome	landing outcomes
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

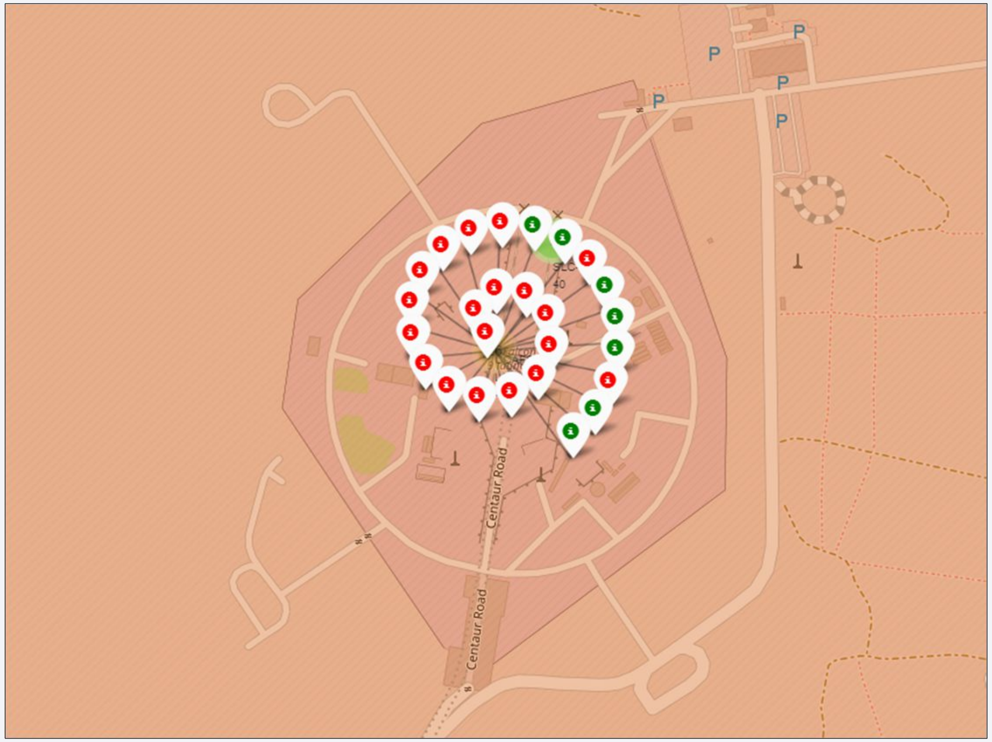
Launch Sites Proximities Analysis

Folium Map: All launch sites location markers on a global map.



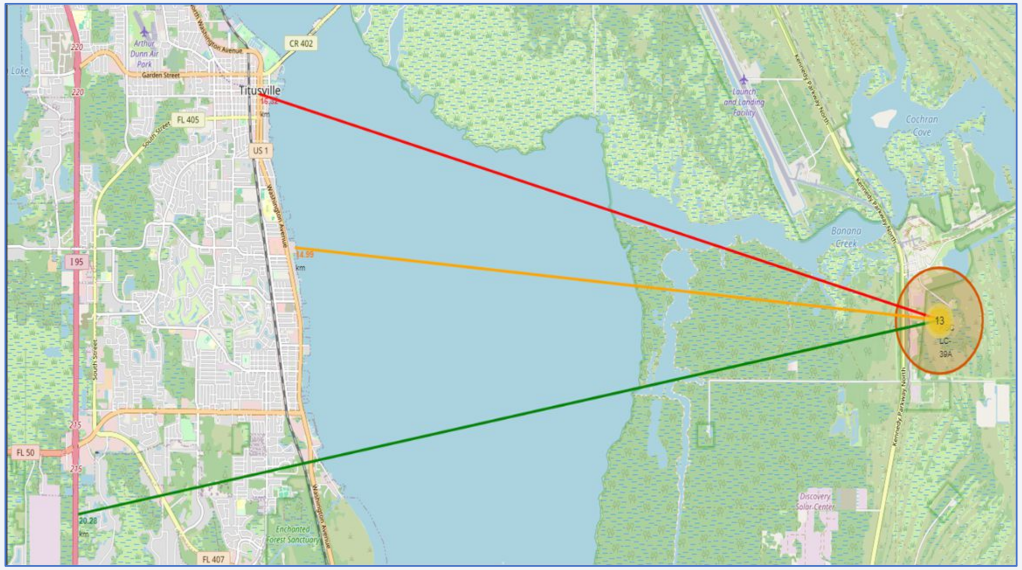
- Rocket launches are performed close to the Equator line, to take advantage of the Earth's rotation. The linear velocity of the Earth's surface is maximum towards the Equator.
- They are also launched near the coast, because it is very desirable to have a sparsely populated safety range, in case of accidents; an ocean is ideal for this.

Folium Map: Success/failed launches for each site on the map.



- From the color-labeled markers in marker clusters, you should be able to easily identify which launch sites have relatively high success rates.
- **Red** markers for **unsuccessful** launches.
- **Green** markers for **successful** launches.
- CCAFS SLC-40 has a greater number of failed launches.

Folium Map: Selected launch site to its proximities such as railway, highway, coastline, with distance calculated and displayed.

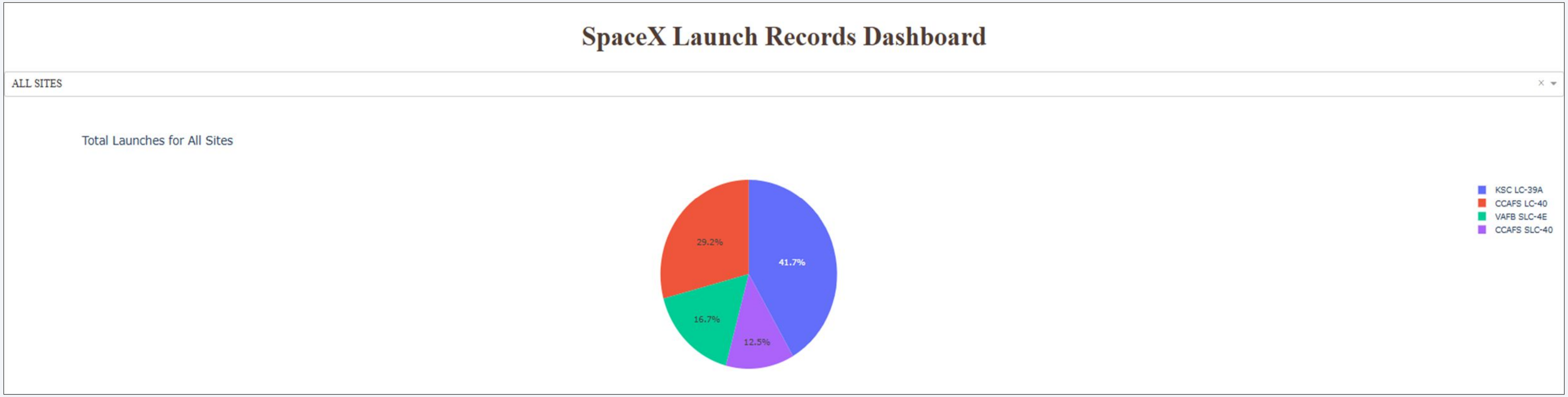


- Distance from the launch site KSC LC-39A to railway: 15.23 km (Close proximity).
- Distance from the launch site KSC LC-39A to highway: 20.28 km (Close proximity).
- Distance from the launch site KSC LC-39A to coastline: 14.99 km (Close proximity).
- Distance from the launch site KSC LC-39A to Titusville City: 16.32 km (Close proximity).

Section 4

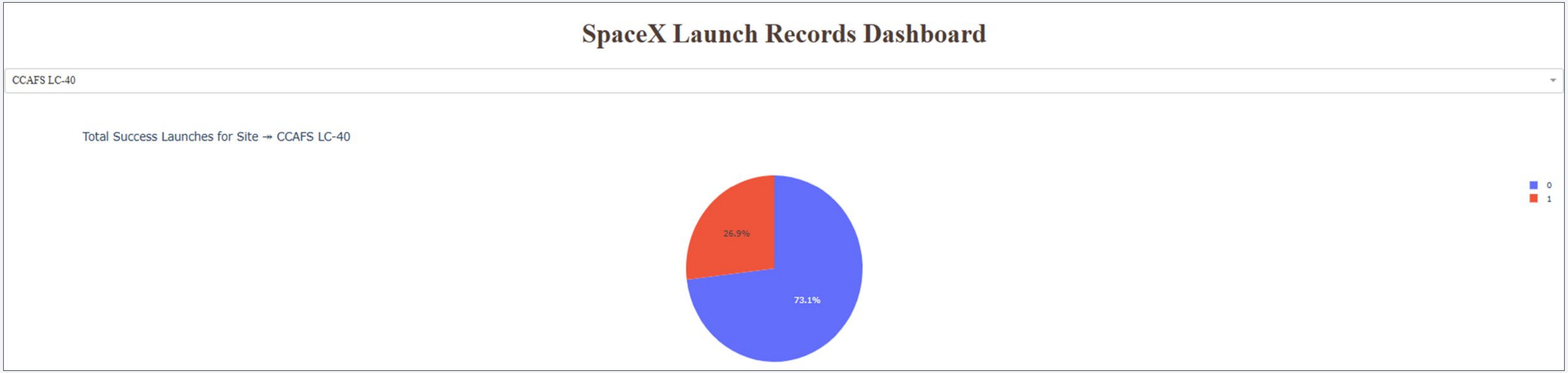
Build a Dashboard with Plotly Dash

Spacex Launch success count for all sites



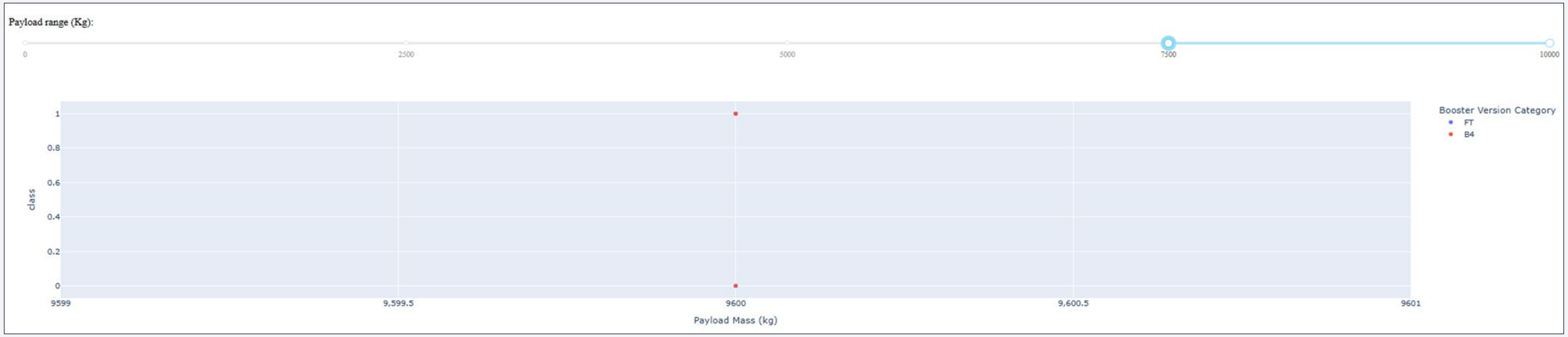
- The launch site LC-39A has the largest successful launches (41.7%).

Spacex Launch site with highest launch success ratio



- Launch site: KSC LC-39A has the highest launch success rate (73.1%).

Spacex Payload Mass vs. Launch Outcome for all sites.
(with different payload selected in the range slider).



Payload Mass between 2.000 kg and 7.500 kg, have the highest success rate.

Note: 0 -> unsuccessful outcome / 1-> successful outcome.

Section 5

Predictive Analysis (Classification)

Classification Accuracy

- Using GridSearchCV object, con cv = 10, in algorithms:
  - Logistic regression.
  - Support vector machine (SVM).
  - Decision tree.
  - K nearest neighbors.

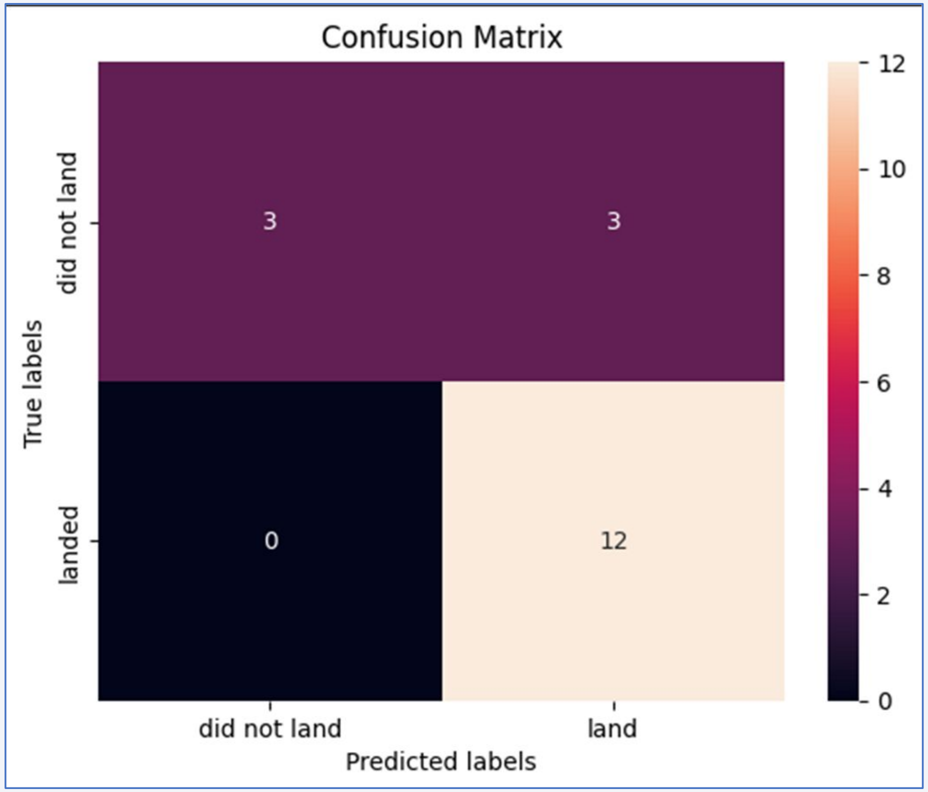
- Calculate the accuracy on the test data using the method score:

	Best scores
Logistic regresssion	0.846429
SVM	0.848214
Decision tree	0.860714
KNN	0.848214

- The method performs best: Decision tree.

Confusion Matrix of the best performing model

- Confusion Matrix of the best performing model : Decision tree.



- Examining the confusion matrix:
- Decision tree can distinguish between the different classes.
- **The problem** is false positives.
- **True Positive:** 12 (True label is landed, Predicted label is also landed).
- **False Positive:** 3 (True label is not landed, Predicted label is landed).

Conclusions

- For all models, the accuracy is similar. The method performs best: Decision tree.
- Decision tree can distinguish between the different classes. The problem is false positives.
- The launch site LC-39A has the largest successful launches (41.7%) and the highest launch success rate (73.1%).
- Payload Mass between 2.000 kg and 7.500 kg, have the highest success rate.
- Rocket launches are performed close to the Equator line, to take advantage of the Earth's rotation. The linear velocity of the Earth's surface is maximum towards the Equator.
- They are also launched near the coast, because it is very desirable to have a sparsely populated safety range, in case of accidents; an ocean is ideal for this.
- The CCAFS SLC 40 launch site has the largest number of launches.
- KSC LC 39A launch site and VAFB-SLC launch site have the highest proportion of successful launches.
- Orbits with the highest success rates (100%): ES-L1, SSO, HEO, GEO.
- Orbit with 0% of success rates: SO.
- With heavy payloads the successful landing or positive landing rate are more for Polar, LEO and ISS.
- However, for GTO, it's difficult to distinguish between successful and unsuccessful landings as both outcomes are present.
- The success rate since 2013 kept increasing until 2020.

Appendix

Import Libraries and Define Auxiliary Functions

```
import piplite
await piplite.install(['numpy'])
await piplite.install(['pandas'])
await piplite.install(['seaborn'])
```

```
!pip install numpy
!pip install pandas
!pip install seaborn
```

Load the dataframe

Load the data

```
from js import fetch
import io

URL1 = "https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/dataset_part_2.csv"
resp1 = await fetch(URL1)
text1 = io.BytesIO((await resp1.arrayBuffer()).to_py())
data = pd.read_csv(text1)
```

```
data = pd.read_csv("https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/dataset_part_2.csv")
```

Create a NumPy array from the column `Class` in `data`, by applying the method `to_numpy()` then assign it to the variable `Y`, make sure the output is a Pandas series (only one bracket df['name of column']).

```
Y = data['Class'].to_numpy()
```

Standardize the data in `X` then reassign it to the variable `X` using the transform provided below.

```
# students get this
transform = preprocessing.StandardScaler()
```

```
X = transform.fit(X).transform(X)
```

Create a decision tree classifier object then create a `GridSearchCV` object `tree_cv` with cv = 10. Fit the object to find the best parameters from the dictionary `parameters`.

```
parameters = {'criterion': ['gini', 'entropy'],
     'splitter': ['best', 'random'],
     'max_depth': [2*n for n in range(1,10)],
     'max_features': ['auto', 'sqrt'],
     'min_samples_leaf': [1, 2, 4],
     'min_samples_split': [2, 5, 10]}

tree = DecisionTreeClassifier()
```

```
tree_cv = GridSearchCV(tree, parameters, scoring='accuracy', cv=10)
tree_cv = tree_cv.fit(X_train, Y_train)
```

Calculate the accuracy of tree_cv on the test data using the method `score`:

```
tree_accuracy = tree_cv.score(X_test, Y_test)
tree_accuracy
```

```
0.8333333333333334
```

We can plot the confusion matrix

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


Thank you!