IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Renan Massena de Oliveira
14 April 2022

Executive Summary

Based on the need to reduce costs and increase the performance of aerospace expeditions, this project aims to predict the successful landing of the first stage of a rocket.

Data from SpaceX Falcon 9 launches was analyzed. The data was collected through an API and web scraping, the steps of cleaning, transforming and then visualizing and manipulating the data through Exploratory Analysis (EDA) were performed, for the prediction of future launches, classification models were used and the proper evaluations of the models.

The developed models presented a good accuracy evaluation and returned satisfactory results, with good confidence.

Introduction

Space exploration is an important topic for the development of human knowledge and new technologies. Investment in rocket launches into space is a business that involves great cost and risk and needs constant improvement.

The aerospace manufacturer SpaceX, was able to reduce the cost of its missions by developing a technology capable of reusing the first stage of the rocket.

SpaceX announces that for its Falcon 9 rocket, it is possible to reduce the cost from $165 million to $62 million if the first stage lands safely.

- **Research Question**: Is it possible to predict a successful first stage rocket landing?
- **Goal**: To predict the successful first stage landing of a rocket such as the Falcon 9.

To find the answers and the best results for this project, data on the Falcon 9 launches from the manufacturer SpaceX was analyzed.

Section 1

Methodology

Methodology

Executive Summary

Methodological steps:

- Data collection:

Data was collected via the SpaceX API and scraping the Wikipedia page "Falcon 9 and Falcon Heavy launch list".

- Data Dispute:

The data was cleaned, transformed and observed in order to be able to create a new column that would serve as the target for the models.

- Exploratory Data Analysis (EDA) using visualization and SQL.
- Interactive visual analysis using Folium and Plotly Dash.
- Predictive analysis using classification models.
- After feature engineering, the data was standardized and partitioned. Classification models were built and evaluated.

Data Collection

To help in finding the answers to the problem of this project, it was necessary to collect data on SpaceX and Falcon 9 and observe which part could help in the prediction.

- The data was collected in two stages:
  - Through the SpaceX API
  - Scraping the Wikipedia page "Falcon 9 and Falcon Heavy launch list".

Objectives:	
API	WEB SCRAPING
Request to the SpaceX API	Extract a Falcon 9 launch records HTML table from Wikipedia
Clean the requested data	Parse the table and convert it into a Pandas data frame

Data Collection – SpaceX API

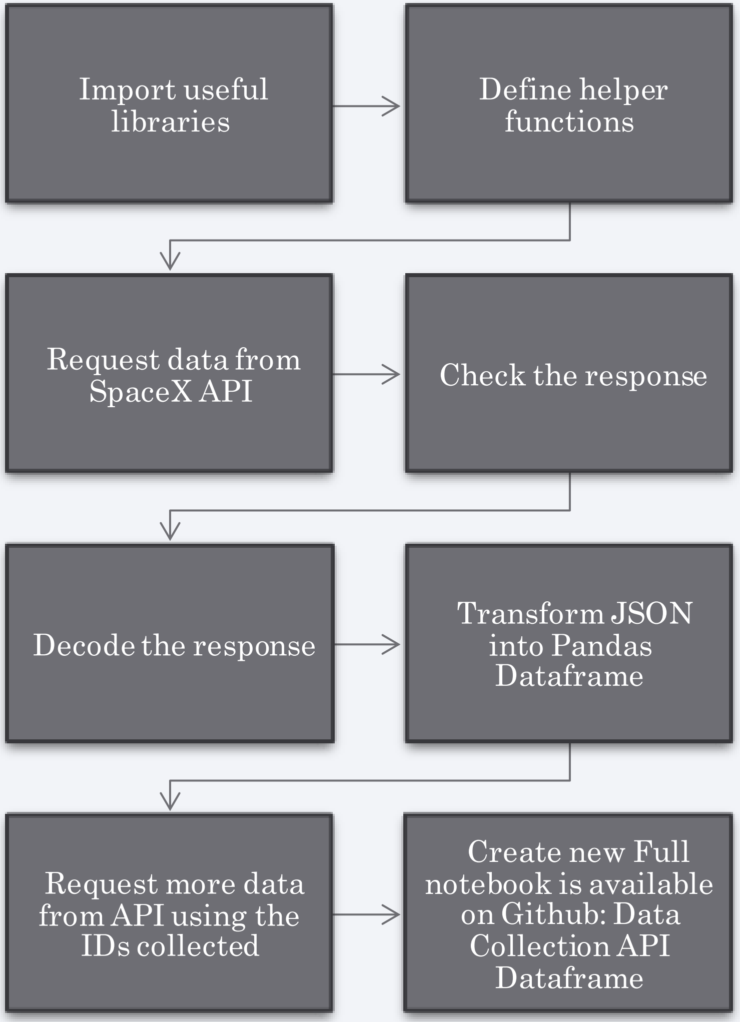
The SpaceX API was accessed using the request library, the rocket launch data was requested.

The response was through the JSON language, which was transformed into a Pandas Dataframe.

```
response = requests.get(static_json_url)
response.json()
data = pd.json_normalize(response.json())
```

Much of the data present in the Dataframe was ID numbers and contained no other information. A new access was required and will be detailed on the next slide.

The notebook is available on Github:
[Data Collection API]



Data Collection – SpaceX API

Using the IDs, the information considered most relevant was requested.

- The new data collected was first stored in lists and later in a dictionary, to be transformed into a new Pandas Dataframe.

```
launch_dict = {'FlightNumber': list(data['flight_number']),
'Date': list(data['date']),
'BoosterVersion':BoosterVersion,
'PayloadMass':PayloadMass,
'Orbit':Orbit,
'LaunchSite':LaunchSite,
'Outcome':Outcome,
'Flights':Flights,
'GridFins':GridFins,
'Reused':Reused,
'Legs':Legs,
'LandingPad':LandingPad,
'Block':Block,
'ReusedCount':ReusedCount,
'Serial':Serial,
'Longitude': Longitude,
'Latitude': Latitude}
```

Used IDs	Information collected
Rocket	Booster version
Payload	Payload Mass, Orbit
Launchpad	Launch site's name, Latitude, Longitude
Cores	Outcome, landing type, number of flights, gridfins use, core reusability, legs use, landing pad, core block, number of time the core was used, serial of the core.

Data Collection – SpaceX API – Data Wrangling

- After collecting the data from the SpaceX API, the Dataframe was filtered to include only the Falcon 9 launches and checked for missing values.
- Missing values were identified in two columns: PayloadMass and LandingPad.
- The null values in LandingPad represent when the landing pads were not used, so no action was performed.
- For the missing values in the PayloadMass column, the substitution function was used to replace the NAN Value with the average PayloadMass.

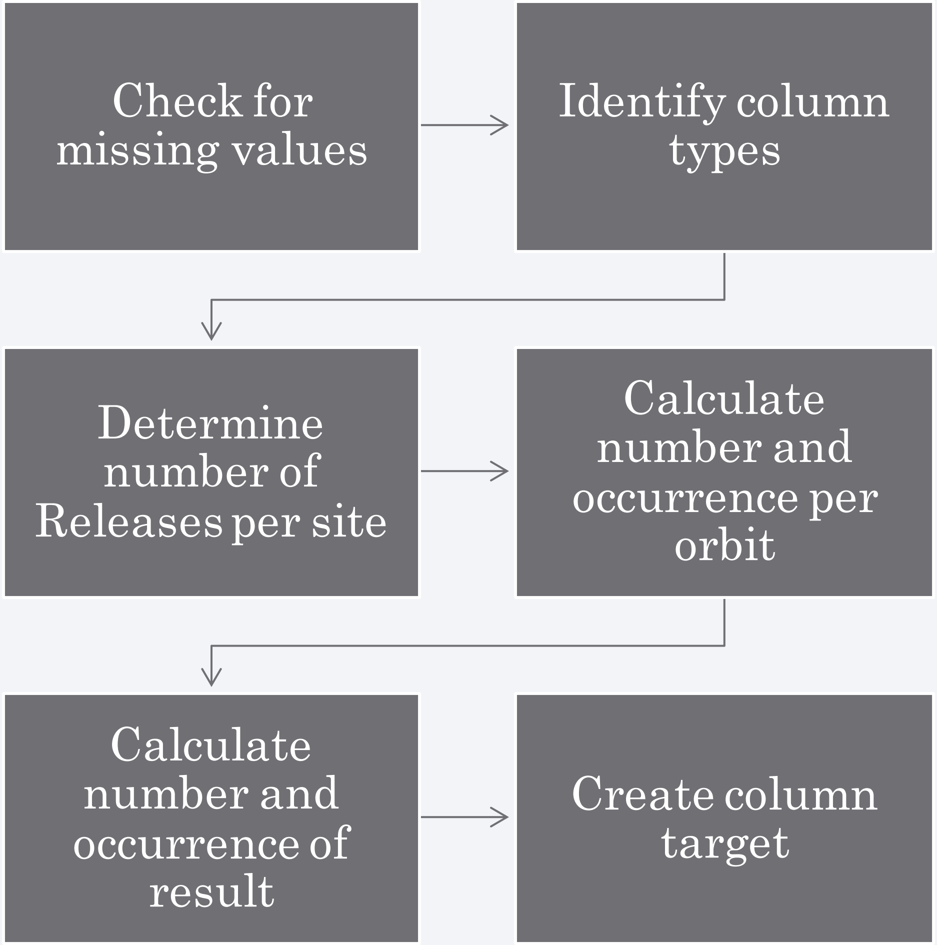
```
# Calculate the mean value of PayloadMass column
data_falcon9['PayloadMass'].mean()
# Replace the np.nan values with its mean value
data_falcon9.replace(np.nan, data_falcon9['PayloadMass'].mean(), inplace = True)
```

```
data_falcon9.isnull().sum()

FlightNumber       0
Date               0
BoosterVersion     0
PayloadMass        5
Orbit              0
LaunchSite         0
Outcome            0
Flights            0
GridFins           0
Reused             0
Legs               0
LandingPad        26
Block              0
ReusedCount        0
Serial             0
Longitude          0
Latitude           0
dtype: int64
```

Data Wrangling

- Data wrangling consists of the processes of cleaning, transforming, and mapping data.
- An exploratory data analysis (EDA) was performed to find data patterns and define which label to use for supervised model training.
- The data discussion process was divided into:
  - Checking for missing values.
  - Identifying the type of columns, whether numeric or categorical.
  - Analyzing the posting data.
  - Looking at the result of the landing.
  - Defining a new column with the result that will serve as the target, consisting of 1 for successful landing and 0 for landing failure.



The notebook is available on Github:

[Data match - EDA]

Data Wrangling – Defining Target

- To define the target, the data about each launch were analyzed.
- As part of the analysis, it was calculated:
  - The number of launches for each site;
  - The number and occurrences for each orbit;
  - The number and occurrences of mission outcomes.
- It was observed:
  - The landing site with more launches was CCAFS SLC 40 with 55 launches;
  - More launches were made towards the orbit GTO, a geosynchronous orbit that is a high Earth orbit that allows satellites to match Earth's rotation;
  - More than 66% of the launches were able to successfully land the first stage.
- A column named “Class” was created with values that show if a launch had their first stage landed successfully or not:

Class	Info
0	First stage did not land successfully
1	First stage landed successfully

Outcome	Frequency	Meaning	Class
True ASDS	41	successfully lan ded to a drone ship	1
None None	19	failure to land	0
True RTLS	14	successfully lan ded to a ground pad	1
False ASDS	6	unsuccessfully la nded to a drone ship	0
True Ocean	5	successfully lan ded to a specific region of the o cean	1
False Ocean	2	unsuccessfully la nded to a specif ic region of the ocean	0
None ASDS	2	failure to land	0
False RTLS	1	unsuccessfully la nded to a groun d pad	0

EDA with Data Visualization

- To analyze possible relations between several variables in the data, a myriad of charts were plotted.
- Summary of charts:
  - Scatter plot between Flight Number and Launch Site
  - Scatter plot between Payload Mass and Launch Site
  - Bar chart observing the success rate for Orbit type
  - Scatter plot between Flight Number and Orbit type
  - Scatter plot between Payload Mass and Orbit type
  - Line chart showing the Launch success yearly trend

Full notebook is available on Github:
Data Visualization - EDA

EDA with SQL

SQL queries performed:

- Unique launch sites in the space mission
- First 5 records where launch sites begin with the string 'CCA'
- The total payload mass carried by boosters launched by NASA (CRS)
- The average payload mass carried by booster version F9 v1.1
- The date when the first successful landing outcome in ground pad was achieved
- The names of the boosters which have success landing in drone ship and have payload mass greater than 4000 but less than 6000
- The total number of successful and failure mission outcomes
- The names of the booster versions which have carried the maximum payload mass
- The failed landing outcomes in drone ship, their booster versions, and launch site names for in year 2015
- The ranking of the count of landing outcomes between 2010-06-04 and 2017-03-20, in descending order

The notebook is available on Github:
[EDA with SQL]

Build an Interactive Map with Folium

Maps created with Folium:

- All Launch Sites map: it contains circles and markers to highlight the location of each launch site area.
- Success and failed outcomes per site: colored-marker were inserted for each launch to show which one had a successful or unsuccessful outcome.
- Proximities of launch site: With markers, a few sites like railways, highways, cities and coastline, that are in the proximities of a launch site, were highlighted in the map. The distance between them was calculated and displayed on the map. Finally, a line was plot to better show that distance.

The notebook is available on Github:
[Interactive Visualization with Folium]

Build a Dashboard with Plotly Dash

A Dashboard was made with Plotly Dash, containing:

- Pie charts with information about the outcome by launch sites.
- A dropdown input was created for the dashboard to allow the selection of a specific launch site or all sites to analyze the data.
- Scatter plots with the relation between payload mass and outcome per booster version, for a selected launch site or for all sites.
- It also contains a slider that allows filtering the scatter plots between a range of payload mass.

The notebook is available on Github:
[Dashboard with Plotly Dash]

Predictive Analysis - Classification

- To be able to predict if a first stage rocket would successfully land, a machine learning pipeline was created, considering the data collected previously.
- Before building the classification models, a features engineering stage was necessary to select and prepare which data would be used to help the prediction.
- Classification models algorithms imported for this project:
  - Logistic Regression
  - Support Vector Machine
  - Decision Tree
  - K Nearest Neighbors

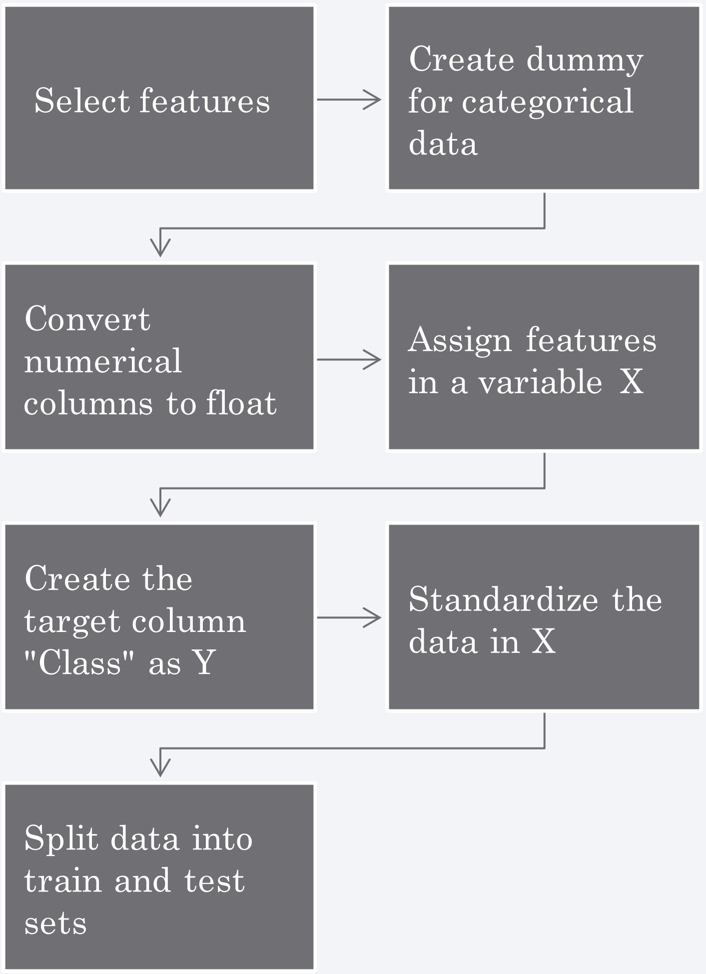
Objectives:

- ✓ Perform EDA and determine training labels.
- ✓ Create a column for the class
- ✓ Standardize the data
- ✓ Split into training data and test data
- ✓ Find best hyperparameter
- ✓ Find the best method using test data

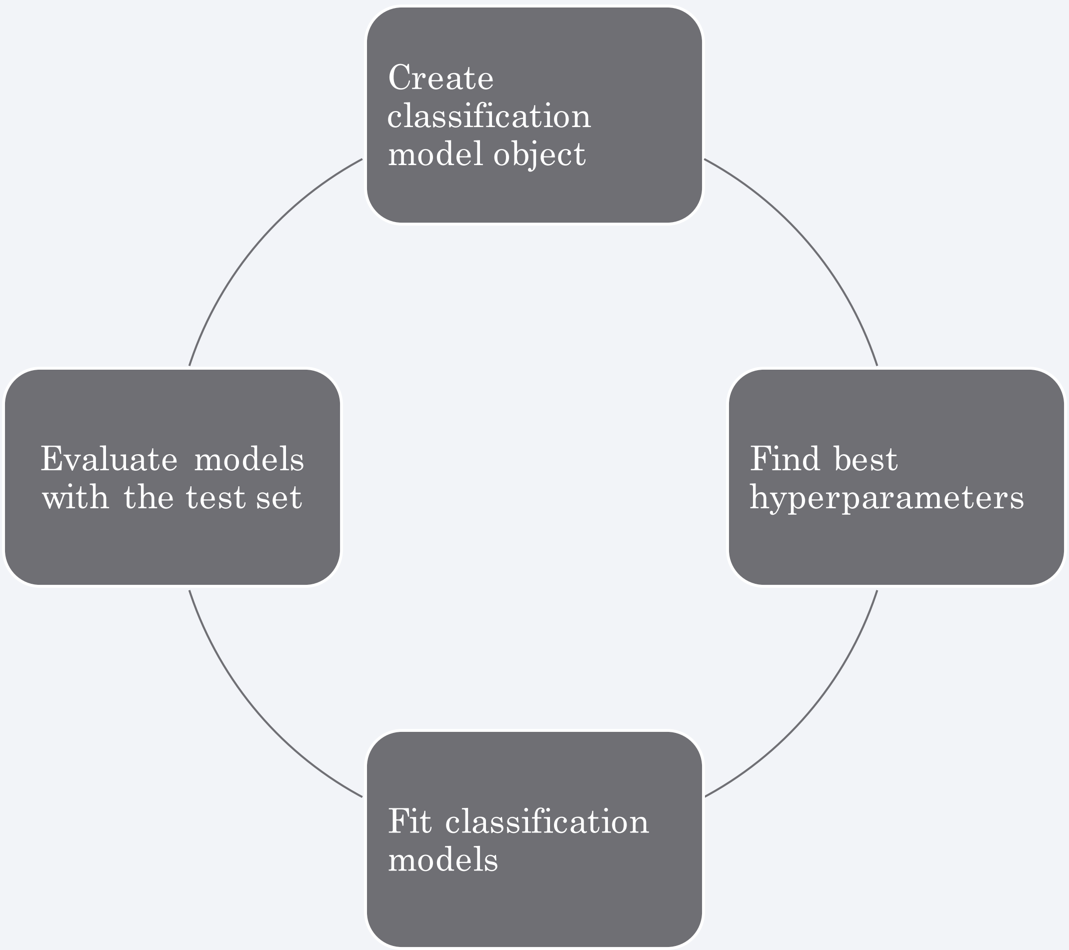
The notebook is available on Github:
Machine Learning Prediction

Predictive Analysis - Flowcharts

Data Preparation Process



Model Training and Evaluation Process



Predictive Analysis - Features engineering

- It was selected all features that would be used in the models to predict the success of landing the first stage rocket.
- Dummy variables were created for the categorical columns: Orbits, LaunchSite, LandingPad and Serial.
- All numeric columns were converted to float64.

Selected features

	FlightNumber	PayloadMass	Orbit	LaunchSite	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial
0	1	6104.959412	LEO	CCAFS SLC 40	1	False	False	False	NaN	1.0	0	B0003
1	2	525.000000	LEO	CCAFS SLC 40	1	False	False	False	NaN	1.0	0	B0005
2	3	677.000000	ISS	CCAFS SLC 40	1	False	False	False	NaN	1.0	0	B0007
3	4	500.000000	PO	VAFB SLC 4E	1	False	False	False	NaN	1.0	0	B1003
4	5	3170.000000	GTO	CCAFS SLC 40	1	False	False	False	NaN	1.0	0	B1004

Predictive Analysis – Classification Models

- After features engineering, the data selected were assigned to a variable called X. The data in this variable were standardized with the StandardScaler( ) method.
- A target variable (Y) was assigned with data from the “Class” column and transformed into a Numpy array.
- The variables X and Y were split into two sets: training and test data.
- Using GridSearchCV( ), the best parameters were found. The best parameters for each model can be seen in the table.
- To evaluate the model, the accuracy for each model was calculated using the score method and a confusion matrix was also plotted.

Model	Best Parameters
Logistic Regression	{'C': 0.01, 'penalty': ‘12’, 'solver': 'lbfgs'}
SVM	{'C': 1.0, 'gamma': 0.03162277660168379, 'kernel': 'sigmoid'}
Decision Tree	'criterion': 'gini’, 'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 10, 'splitter': 'random'}
KNN	{'algorithm': 'auto’, 'n_neighbors': 10, 'p': 1}

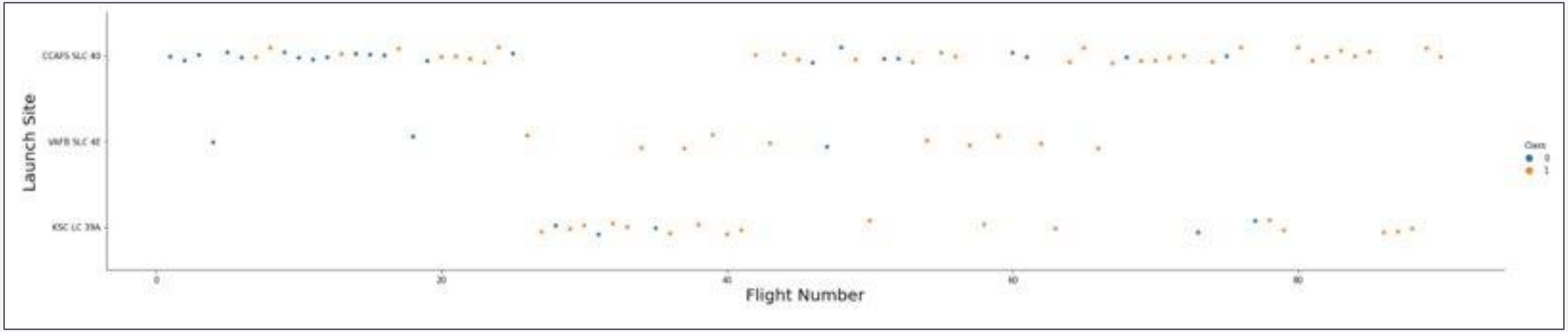
Results

- The exploratory data analysis (EDA) allowed us to analyze the relation between variables and to discover the proper ones to use for predicting if a first stage would land successfully.
- The selected columns that were used as features for the models, after EDA stage, were: FlightNumber, PayloadMass, Orbit, LaunchSite, Flights, GridFins, Reused, Legs, LandingPad, Block, ReusedCount, Serial.
- The “Class” column was the target for the prediction, that indicates the outcome of the first stage landing.
- Of the four classification models, the Decision Tree model had a better result with the train set, obtaining an accuracy of 0.87.
- With the test set, all models had the same accuracy of 0.83 and the same confusion matrix

Section 2

Insights drawn from EDA

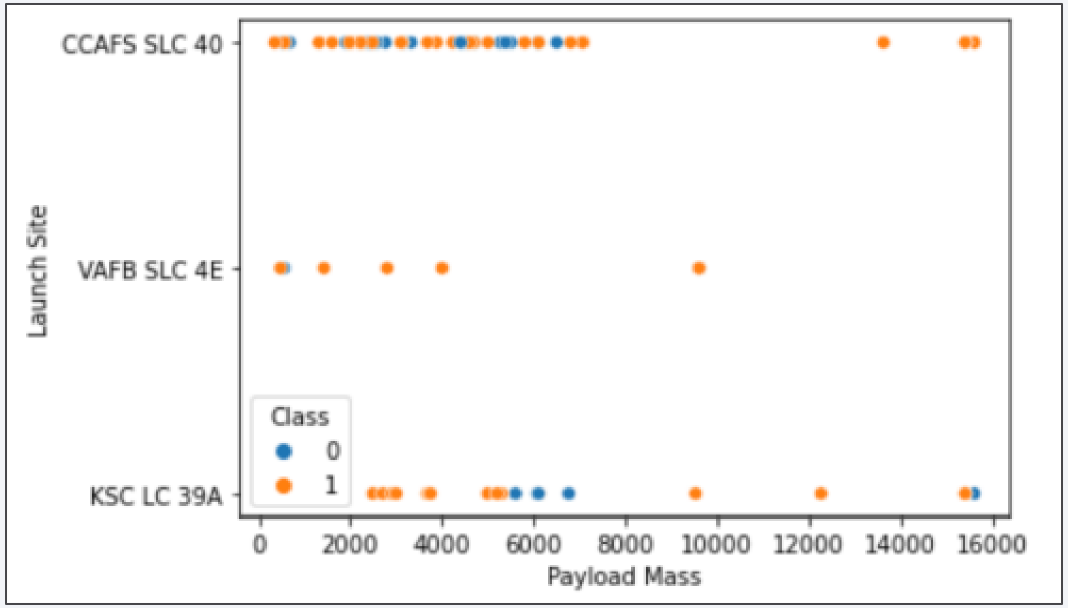
Flight Number vs. Launch Site



The chart shows where each flight was launched and their outcome.

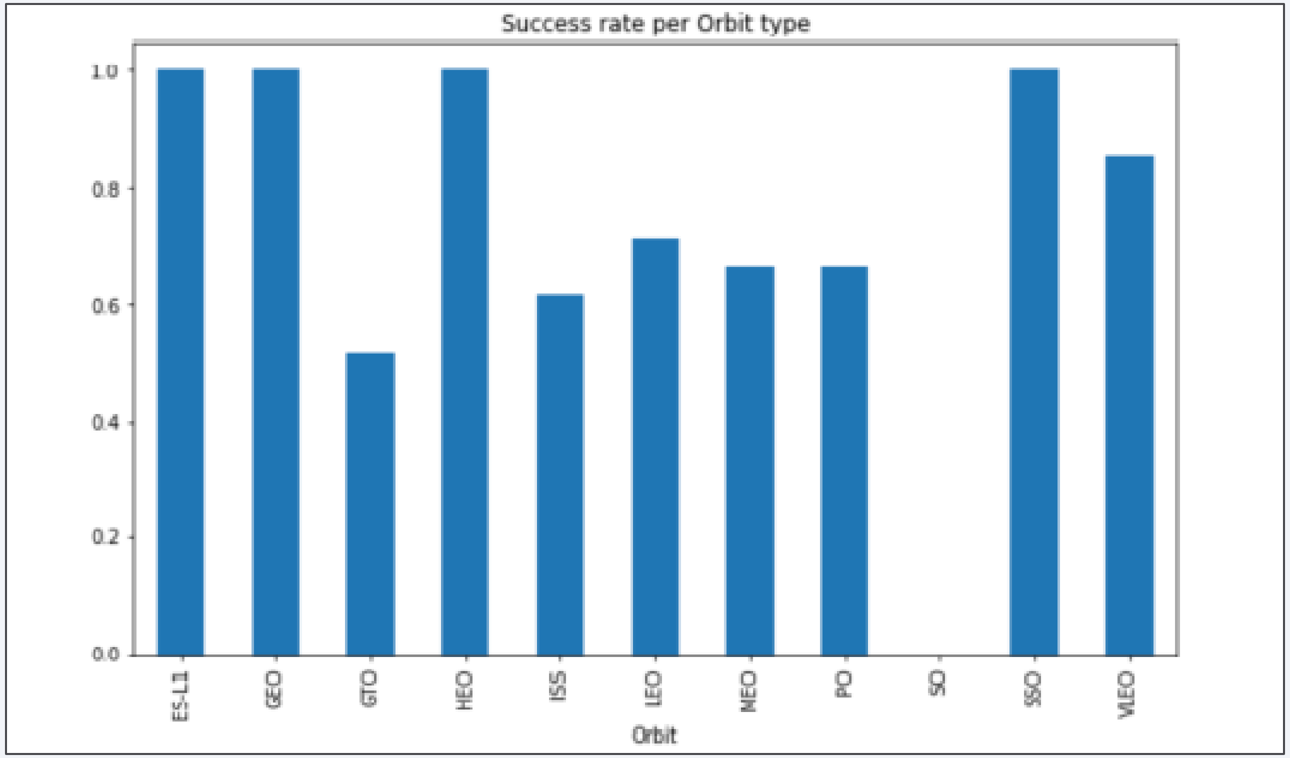
- It is possible to observe that most launches occurred at CCAFS SLC 40.
- It is possible to observe a gap with no flights in CCAFS SLC 40 between flights 25 and 40 and almost all flights in this gap were launched at KSC LC 39A.
- With Flight Numbers more than 30, the sucess rate for the Rocket is increasing

Payload vs. Launch Site



- Now if you observe Payload Vs. Launch Site scatter point chart you will find for the VAFB-SLC launchside there are no rockets launched for heavypayload mass (greater than 10000). And payload mass greater than 7000Kg have the highter success rate for the rocket.
- The chart show the payload mass of each flight per launch site.
- Almost all payload mass are under 10000 kg.
- No heavy payload was launched at VAFB SLC.

Success Rate vs. Orbit Type

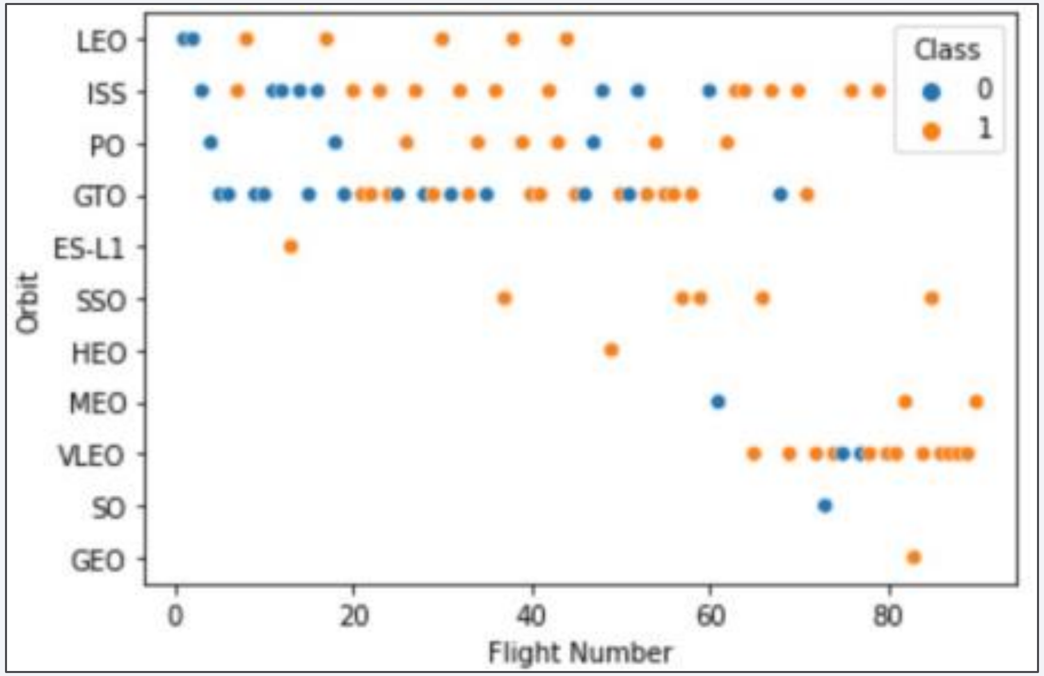


All missions were successful in four orbits:

- ES-L1
- GEO
- HEO
- SSO

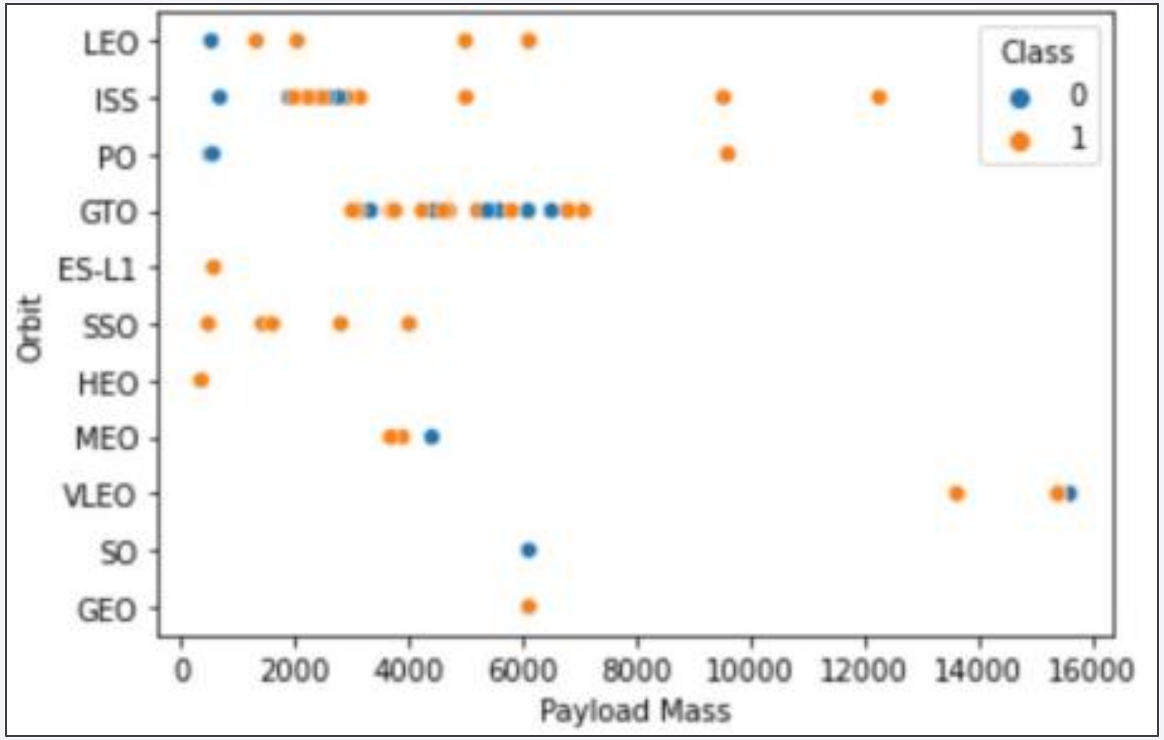
- The GTO orbit has the least success rate. Half of the missions to GTO were successful in landing the first stage.

Flight Number vs. Orbit Type



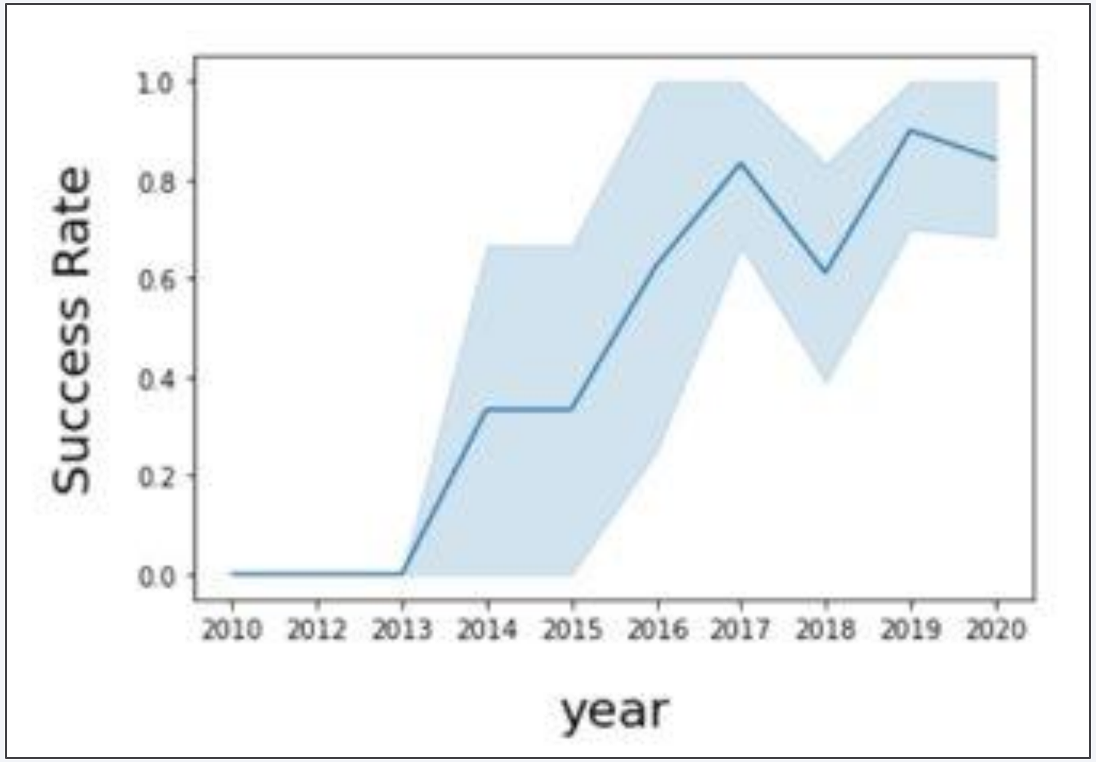
- It is possible to observe that the number of flights to LEO orbit seems to be related with the mission success. After the initial failures, all flights were successful in landing the first stage.
- The same does not apply to GTO orbit, which concentrates the higher number of missions that failed in landing the first stage, but regardless of number of flights.

Payload vs. Orbit Type



- It seems that with payload mass greater than 5000 kg, the rate of successful landings are higher for Polar, Leo and ISS orbits.
- For GTO the same could not be observed, since both landing results (successful and failure) are present, through all payload mass range.

Launch Success Yearly Trend



- With this line chart, it is possible the notice an increase of successful rate from 2013.
- Although there was no year that did not have at least one unsuccessful first stage landing

All Launch Site Names

```
%sql SELECT DISTINCT LAUNCH_SITE as "Launch_Sites" FROM SPACEX;
```

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

The DISTINT statement is used to return only distinct (different) values. Inside the SpaceX table, the column Launch_Sites contained many duplicate values; and in our case we just want to list the different (distinct) values.

Launch Site Names Begin with 'CCA'

```
%sql SELECT LAUNCH_SITE from SPACEXTBL where (LAUNCH_SITE) LIKE 'CCA%' LIMIT 5;
```

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	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	00:35:00	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

The SQL SELECT LIMIT statement is used to retrieve records from one or more tables in a database and limit the number of records returned based on a limit value.
The LIKE command is used in a WHERE clause to search for a specified pattern in a column.

Total Payload Mass

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) AS "Total Payload Mass by NASA (CRS)"
FROM SPACEX WHERE CUSTOMER = 'NASA (CRS)';
```

Total Payload Mass by NASA (CRS)
45596

The SUM() function returns the total sum of a numeric column.

Average Payload Mass by F9 v1.1

```
%sql select avg(PAYLOAD_MASS__KG_) as payloadmass from SPACEXTBL;
```

payloadmass
2928

The AVG statement calculates Average in the PAYLOAD_MASS_KG_ column.

First Successful Ground Landing Date

```
%sql select min(DATE) from SPACEXTBL;
```

2015-12-22

The MIN() function returns the smallest value of the selected column. In our case, the DATE column.

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%sql select BOOSTER_VERSION from SPACEXTBL where
LANDING__OUTCOME='Success (drone ship)'
and PAYLOAD_MASS__KG_ BETWEEN 4000 and 6000;
```

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

Using WHERE with AND statement we can expand the power of the filter associating with a condition. Using the BETWEEN statement we can indicate a specific range to the filter.

Total Number of Successful and Failure Mission Outcomes

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

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

Successful Mission
100

Failure Mission
1

- To find the booster that carried the maximum payload mass, the use of a subquery was required. The query returned booster versions where payload mass was equal to the maximum value for payload mass. To be able to use the MAX aggregator function in the WHERE clause, a subquery was made

- The COUNT() function returns the number of rows that matches a specified criterion. Using the LIKE statement, we can filter every letters in quotes " " .

Boosters Carried Maximum Payload

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

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

- The columns with the booster version, landing outcome and the launch site values were selected according to their landing outcome and date, through a WHERE clause and the LIKE operator for finding all 2015 records.
- The DISTINT statement is used to return only distinct (different) values (BOOSTER VERSION column).
- We use the WHERE statement associating with MAX statement to figure out the maximum payload value in the PAYLOAD_MASS_KG_ column. In this case we had used a subquery to improve the filter.

2015 Launch Records

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

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

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

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

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

Landing Outcome	Total Count
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

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

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

Section 3

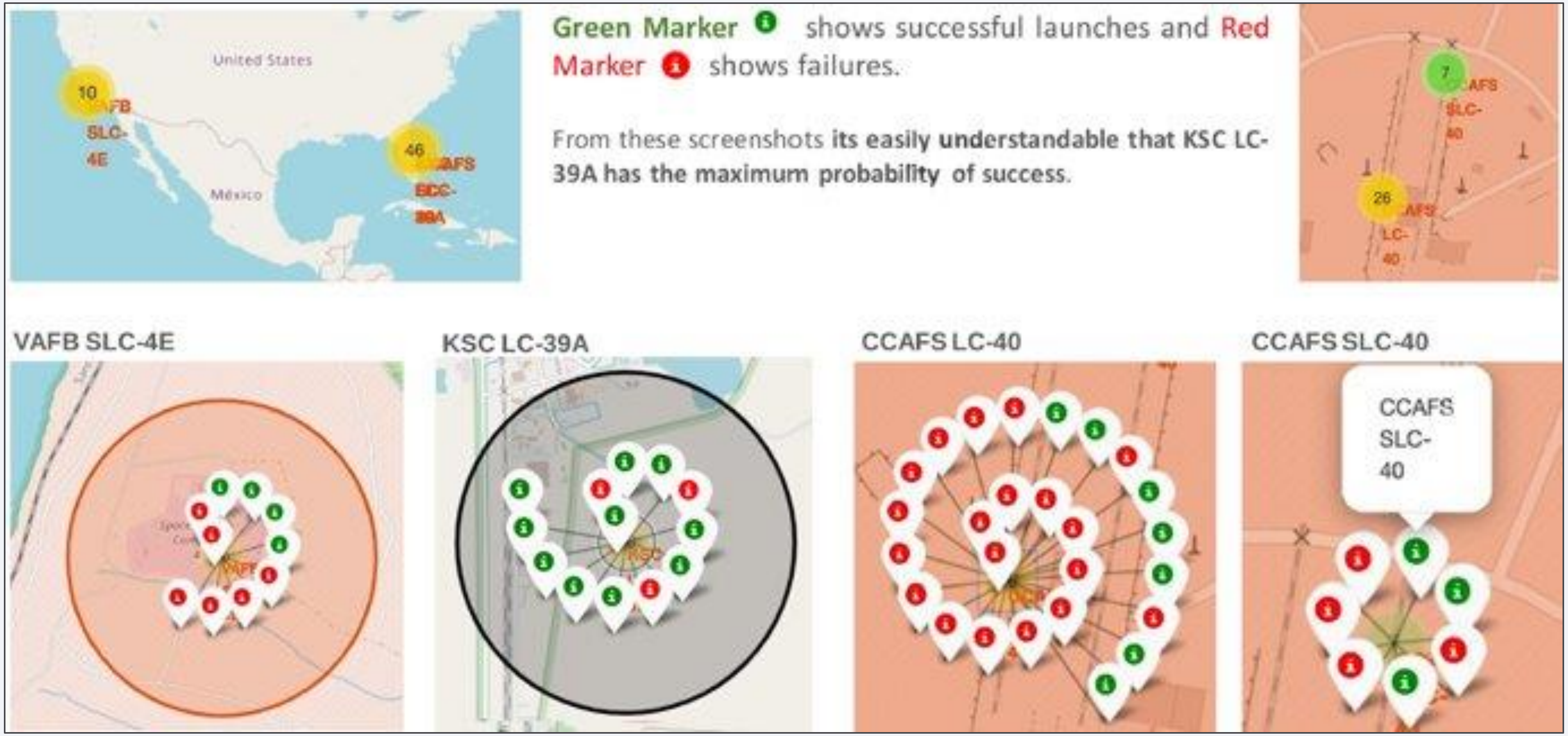
Launch Sites Proximities Analysis

All Launch Sites on Folium Map

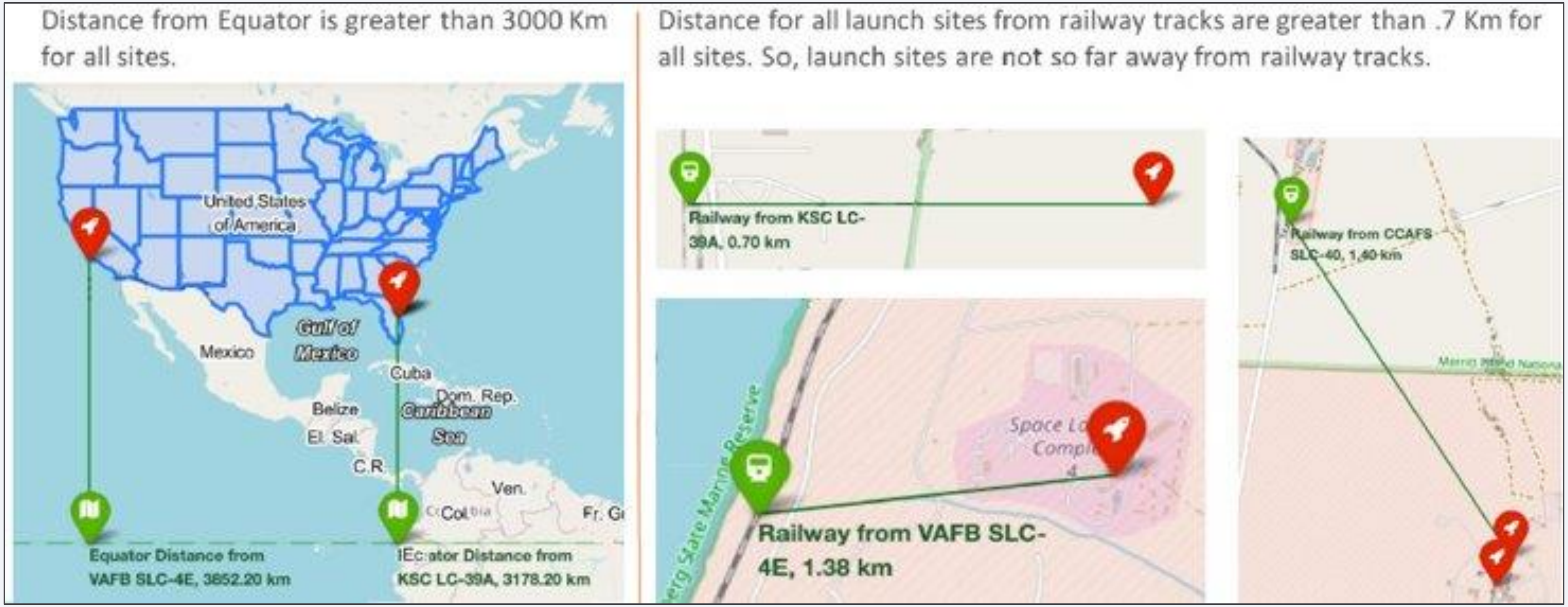
We can see that the SpaceX launch sites are near to the United States of America coasts i.e., Florida and California Regions.

- All launch sites were inserted in an interactive world map (a folium Map Object), that allows zooming into each launch location. The latitude and longitude were used for this purpose.
- A circle and a marker was also inserted on the map, to highlight each launch area.
- Most launch sites are in the east coast of the United States.
- All lunch sites are in the south part of the US and very near the ocean.

Color Labeled Launch Records



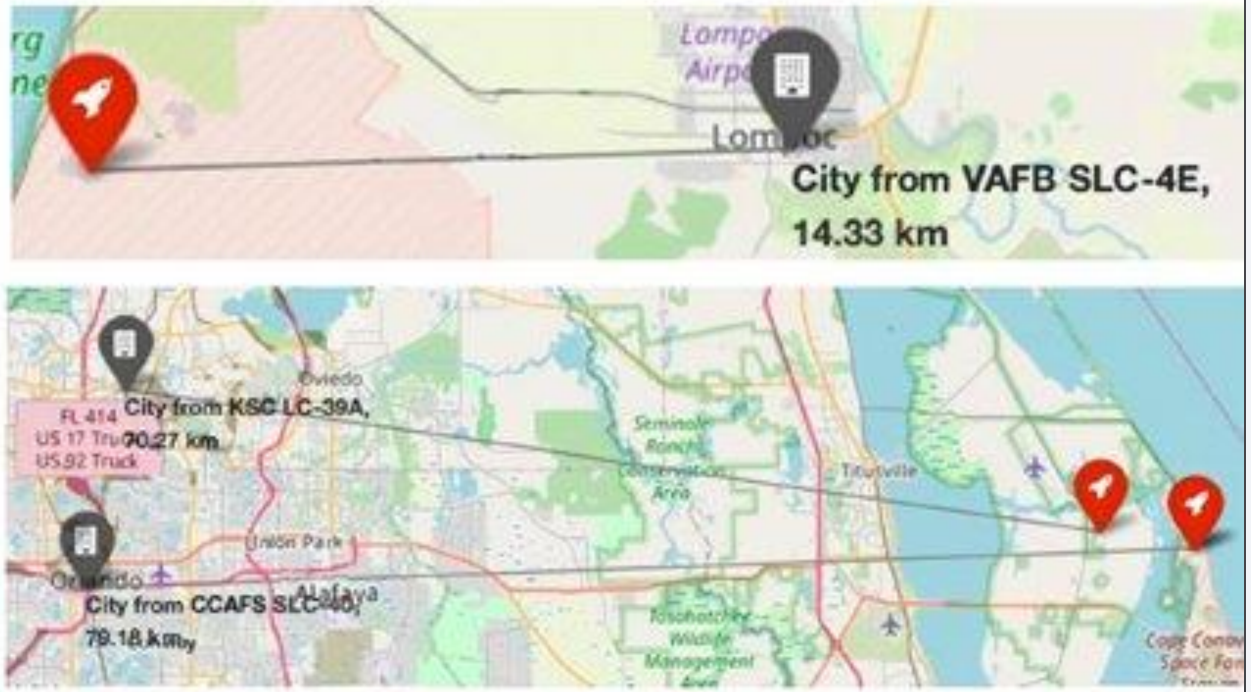
Launch Site Distances from Equator & Railways



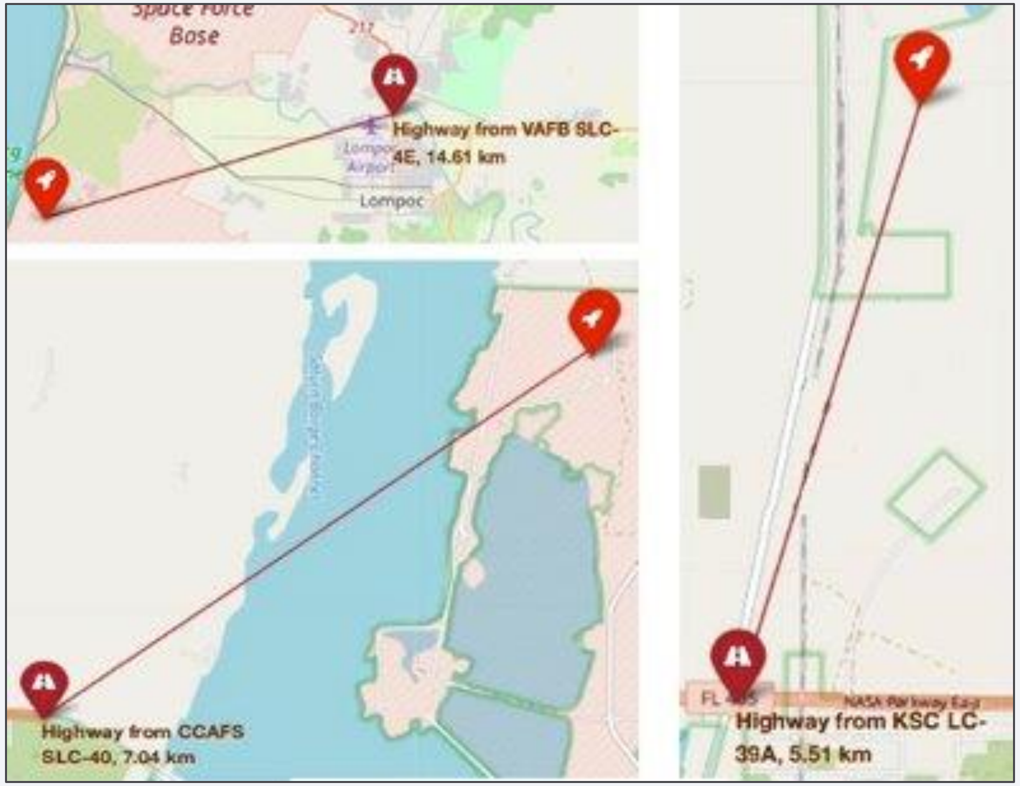
Launch Site Distances from Coastlines & Cities

Distance for all launch sites from coastline is less than 4 Km.

Distance for all launch sites from cities is greater than 14 Km for all sites. So, launch sites are far away from cities.

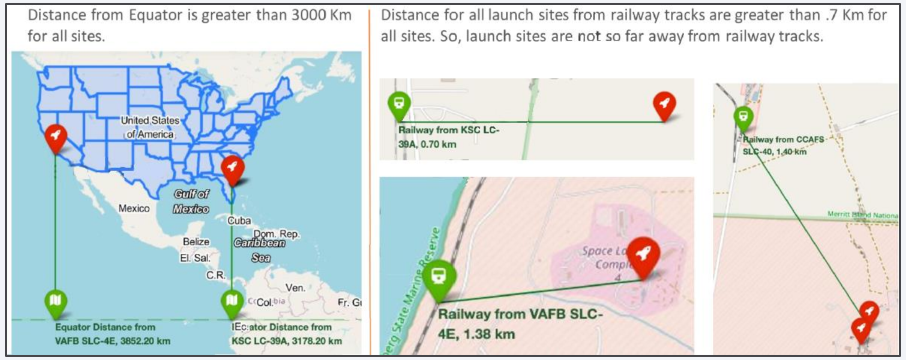


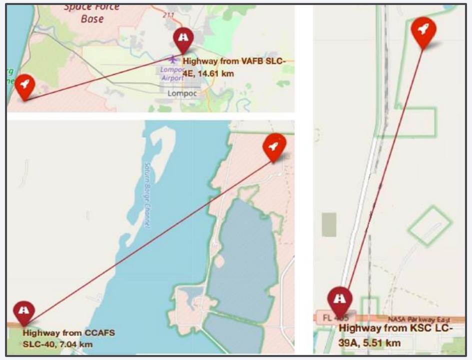
Launch Site Distances from Highways

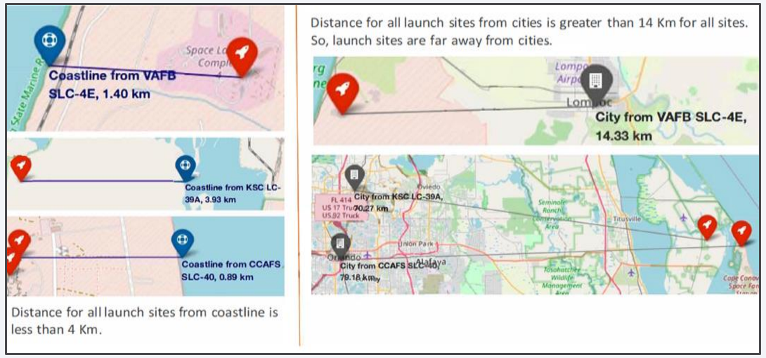


Distance for all launch sites from highways is greater than 5 Km for all sites. So, launch sites are relatively far away from highways.

Launch Site Distances Resume







Distance for all launch sites from highways is greater than 5 Km for all sites. So, launch sites are relatively far away from highways.

The launch sites keep certain distance away from Cities (15 Km> distance > 80 Km).

There is no launch sites in proximity to the Equator Line. (4000 Km> distance > 3000 Km).

There is no launch sites in close proximity to Highways (15 Km> distance > 5 Km).

There is launch sites in close proximity to Coastline (5 Km> distance > 5 Km).

There is launch sites in close proximity to Railways. (2 Km> distance >5 Km).

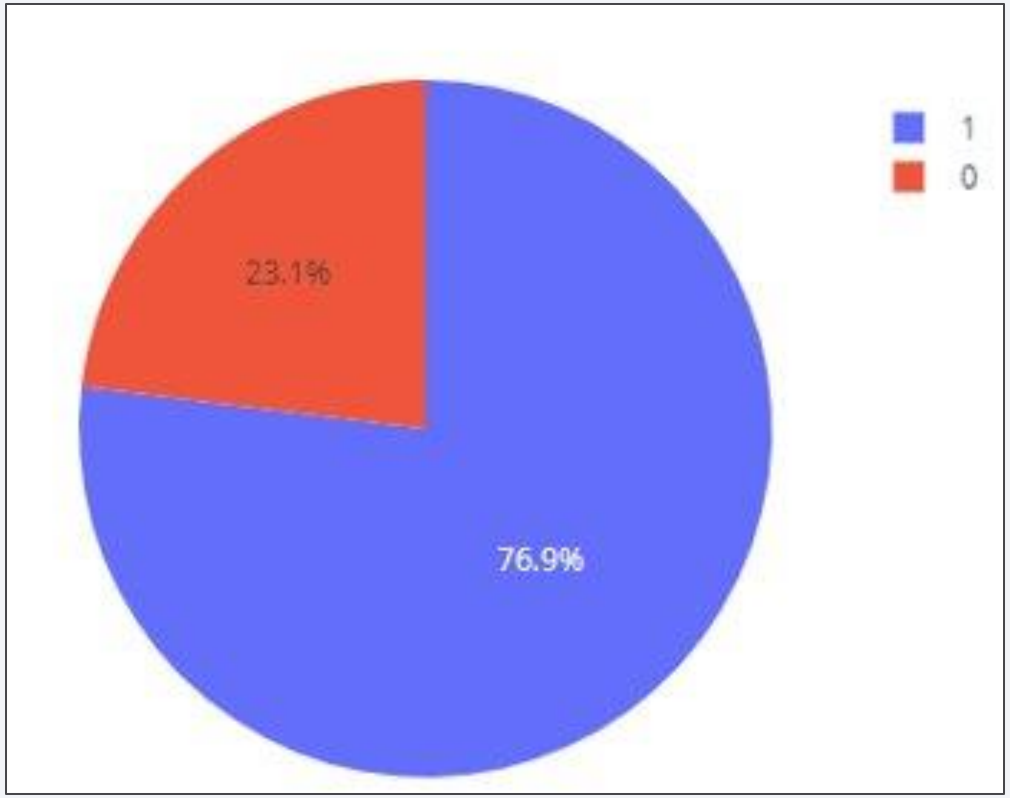
Section 4

Build a Dashboard with Plotly Dash

Launch Success for All Sites

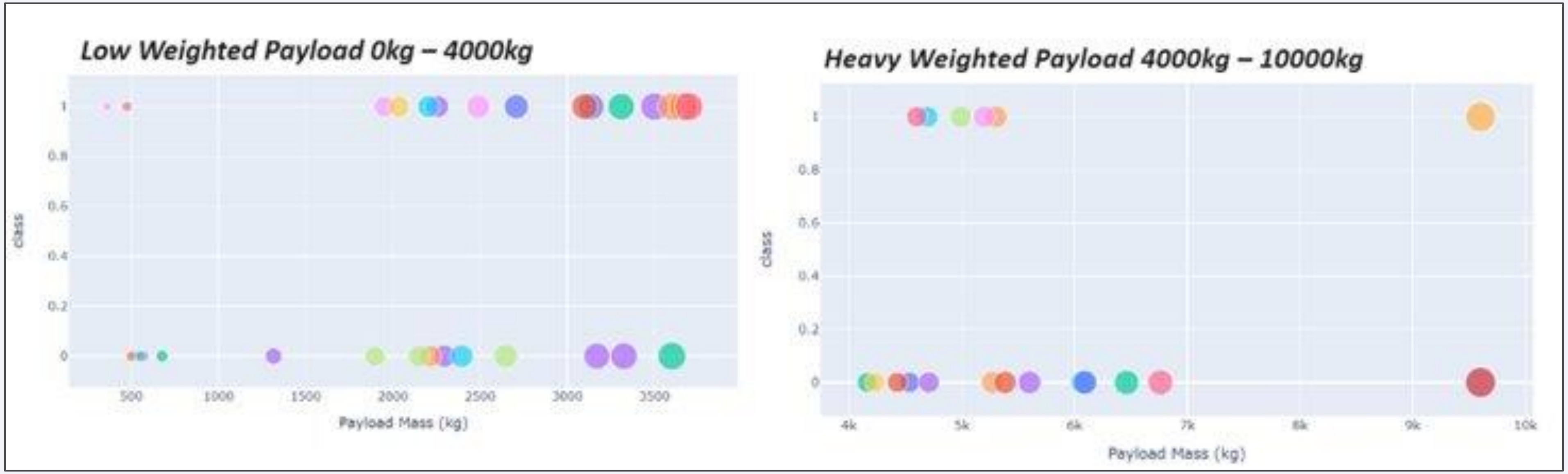
We can see that KSC LC-39A had the most successful launches from all the sites.

Launch Site with Highest Launch Success Ratio



KSC LC-39A achieved a 76.9% success rate (1) while getting a 23.1% failure rate (0).

Payload vs. Launch Outcome Scatter Plot for All Sites

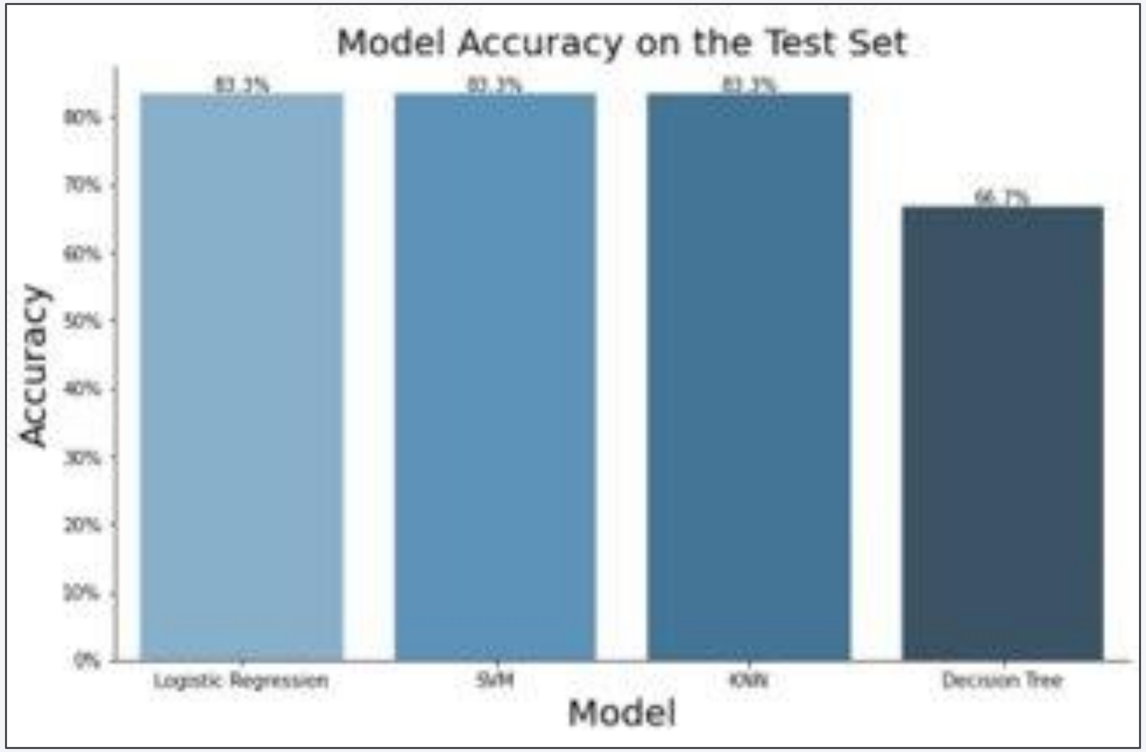


The success rates for low weighted payloads is higher than the heavy weighted payloads.

Section 5

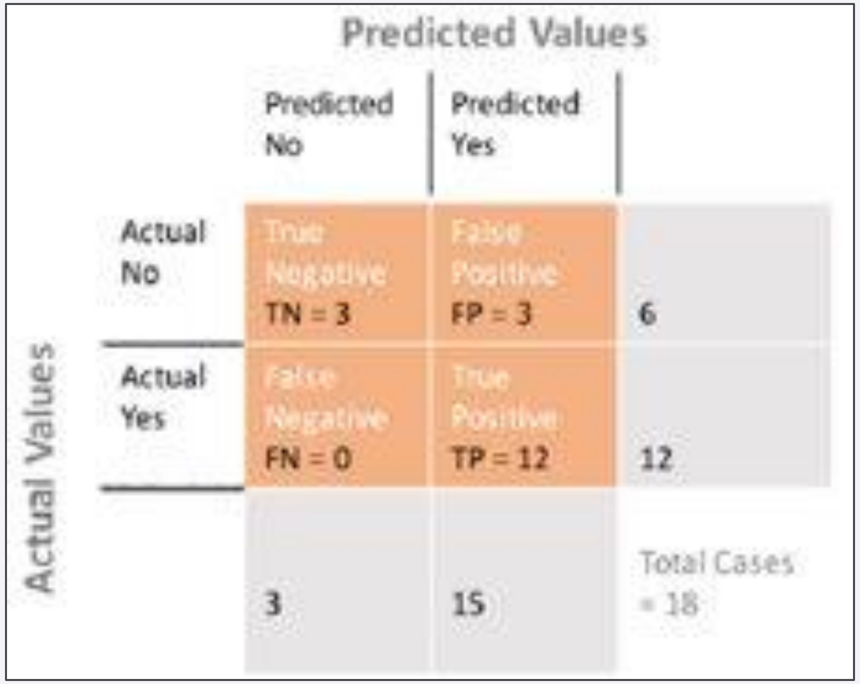
Predictive Analysis (Classification)

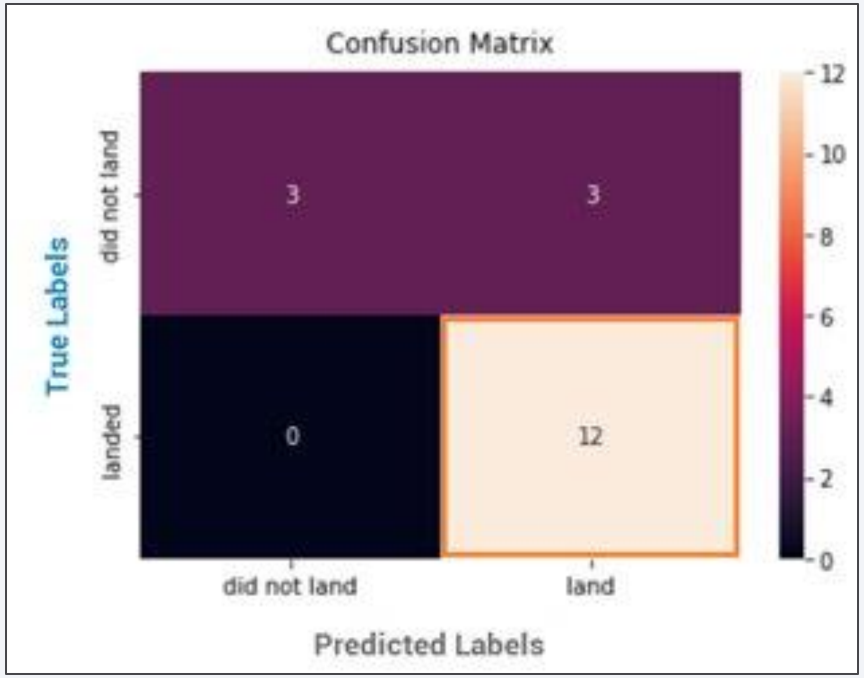
Classification Accuracy



Four machine learning models were used. The Logistic Regression, SVM and KNN models has the highest classification accuracy, 83.3%. The Decision Tree model has the lower, 66,7%.

Confusion Matrix





The Logistic Regression, SVM and KNN models has the same Confusion Matrix. The "landed" x "land" block (True Positive =12) shows the landings success.

Conclusions

We had pursuit what factors determine if the rocket will land successfully, what features impact the rate of successful landing and the conditions which will aid SpaceX have to achieve the best results. So, the data show us that:

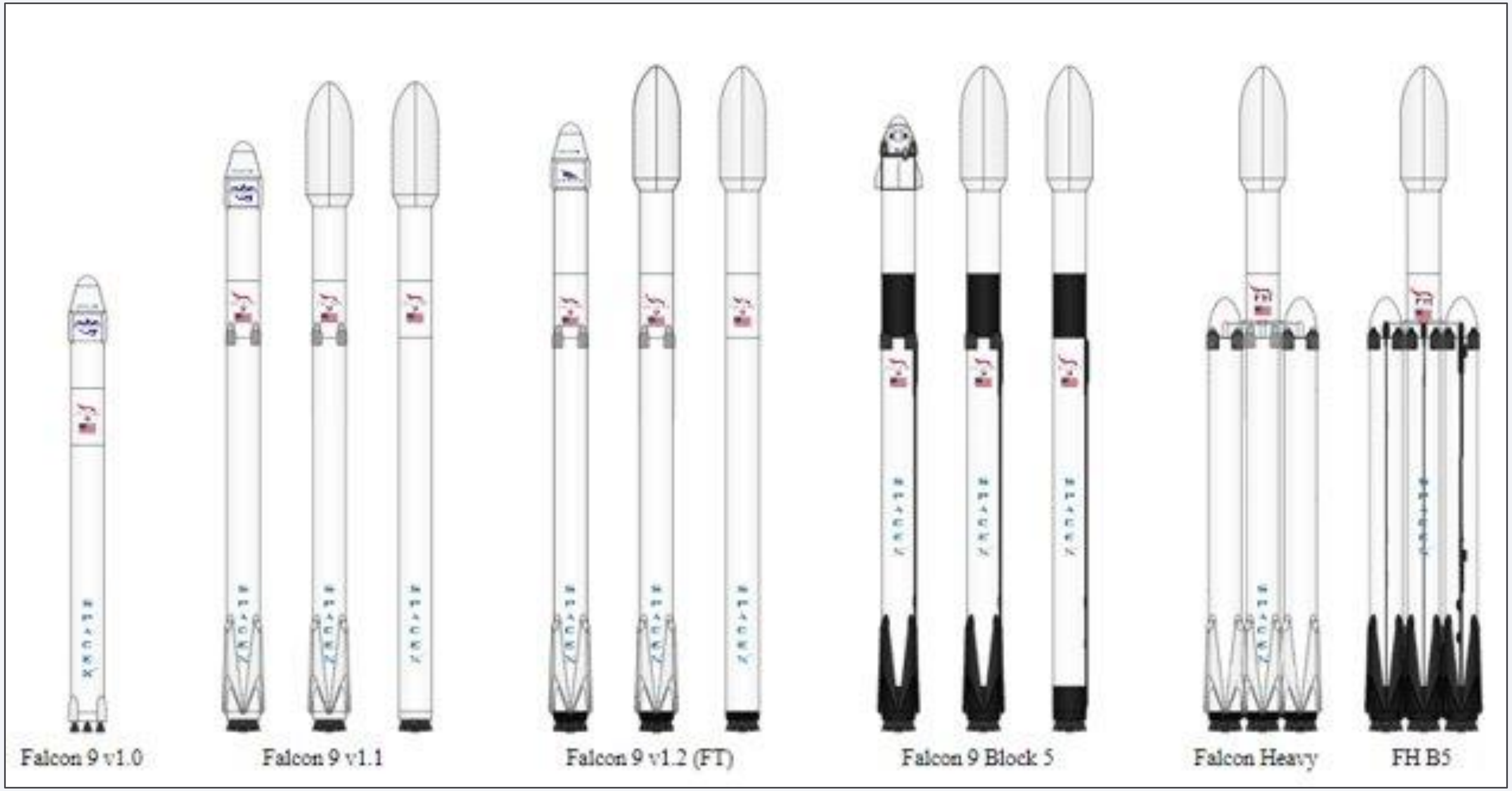
- The best Orbits are: ES-LI, GEO, HEO, SSO.
- KSC LC-39A is best place to make launches, with 76.9% success rate .
- Low weighted payloads perform better than the heavier payloads. However, in the CCAFS SLC 40 Launch Site the payload mass greater than 7000Kg have the higher success rate for the Rocket.
- LR, SVM and KNN Algorithms are the best Machine Learning Models for provided Dataset.

Success rates for SpaceX launches has been increasing with Time. Considering that there are variables that significantly alter the landing success rate, it is possible to say that it is a matter of time for Space X to reach an incredible level of performance and precision.

Appendix

Appendix

Falcon Models



Appendix

<table>
<tr><th>[hide] Flight No.</th><th>Date and time (UTC)</th><th>Version, Booster [b]</th><th>Launch site</th><th>Payload[c]</th><th>Payload mass</th><th>Orbit</th><th>Customer</th><th>Launch outcome</th><th>Booster landing</th></tr>
<tr><td rowspan="2">1</td><td>4 June 2010, 18:45</td><td>F9 v1.0[7] B0003[8]</td><td>CCAFS, SLC-40</td><td>Dragon Spacecraft Qualification Unit</td><td>No payload (excl. Dragon Mass)</td><td>LEO</td><td>SpaceX</td><td>Success</td><td>Failure[9][10] (parachute)</td></tr>
<tr><td colspan="9">First flight of Falcon 9 v1.0.[11] Used a boilerplate version of Dragon capsule which was not designed to separate from the second stage.(more details below) Attempted to recover the first stage by parachuting it into the ocean, but it burned up on reentry, before the parachutes even got to deploy.[12]</td></tr>
<tr><td rowspan="2">2</td><td>8 December 2010, 15:43[13]</td><td>F9 v1.0[7] B0004[8]</td><td>CCAFS, SLC-40</td><td>Dragon demo flight C1 (Dragon C101)</td><td>Classified (excl. Dragon Mass)</td><td>LEO (ISS)</td><td>NASA (COTS) NRO</td><td>Success[9]</td><td>Failure[9][14] (parachute)</td></tr>
<tr><td colspan="9">Maiden flight of SpaceX's Dragon capsule, consisting of over 3 hours of testing thruster maneuvering and then reentry.[15] Attempted to recover the first stage by parachuting it into the ocean, but it disintegrated upon reentry, again before the parachutes were deployed.[12] (more details below) It also included two CubeSats,[16] and a wheel of Brouère cheese. Before the launch, SpaceX discovered that there was a crack in the nozzle of the 2nd stage's Merlin vacuum engine. So Elon just had them cut off the end of the nozzle with a pair of shears and launched the rocket a few days later. After SpaceX had trimmed the nozzle, NASA was notified of the change and they agreed to it.[17]</td></tr>
<tr><td rowspan="2">3</td><td>22 May 2012, 07:44[18]</td><td>F9 v1.0[7] B0005[8]</td><td>CCAFS, SLC-40</td><td>Dragon demo flight C2+[19] (Dragon C102)</td><td>525 kg (1,157 lb)[20] (excl. Dragon mass)</td><td>LEO (ISS)</td><td>NASA (COTS)</td><td>Success[21]</td><td>No attempt</td></tr>
<tr><td colspan="9">The Dragon spacecraft demonstrated a series of tests before it was allowed to approach the International Space Station. Two days later, it became the first commercial spacecraft to board the ISS.[19] (more details below)</td></tr>
<tr><td rowspan="3">4</td><td rowspan="2">8 October 2012, 00:35[22]</td><td rowspan="2">F9 v1.0[7] B0006[8]</td><td rowspan="2">CCAFS, SLC-40</td><td>SpaceX CRS-1[23] (Dragon C103)</td><td>4,700 kg (10,400 lb) (excl. Dragon mass)</td><td>LEO (ISS)</td><td>NASA (CRS)</td><td>Success</td><td rowspan="2">No attempt</td></tr>
<tr><td>Orbcomm-OG2[24]</td><td>172 kg (379 lb)[25]</td><td>LEO</td><td>Orbcomm</td><td>Partial failure[26]</td></tr>
<tr><td colspan="9">CRS-1 was successful, but the secondary payload was inserted into an abnormally low orbit and subsequently lost. This was due to one of the nine Merlin engines shutting down during the launch, and NASA declining a second reignition, as per ISS visiting vehicle safety rules, the primary payload owner is contractually allowed to decline a second reignition. NASA stated that this was because SpaceX could not guarantee a high enough likelihood of the second stage completing the second burn successfully which was required to avoid any risk of secondary payload's collision with the ISS.[27][28][29]</td></tr>
</table>

[Wikipedia Web Scraping]()

Appendix

Data Wrangling Session

```
import pandas as pd
import numpy as np
```

```
df.isnull().sum()/df.count()*100
```

```
FlightNumber      0.000
Date              0.000
BoosterVersion    0.000
PayloadMass       0.000
Orbit             0.000
LaunchSite        0.000
Outcome           0.000
```

```
df.dtypes
```

```
FlightNumber        int64
Date               object
BoosterVersion     object
PayloadMass       float64
```

```
# Apply value_counts() on column LaunchSite
df["LaunchSite"].value_counts()
```

```
CCAFS SLC 40    55
KSC LC 39A      22
VAFB SLC 4E     13
Name: LaunchSite, dtype: int64
```

```
landing_outcomes = df["Outcome"].value_counts()
landing_outcomes
```

```
True ASDS      41
None None      19
True RTLS      14
```

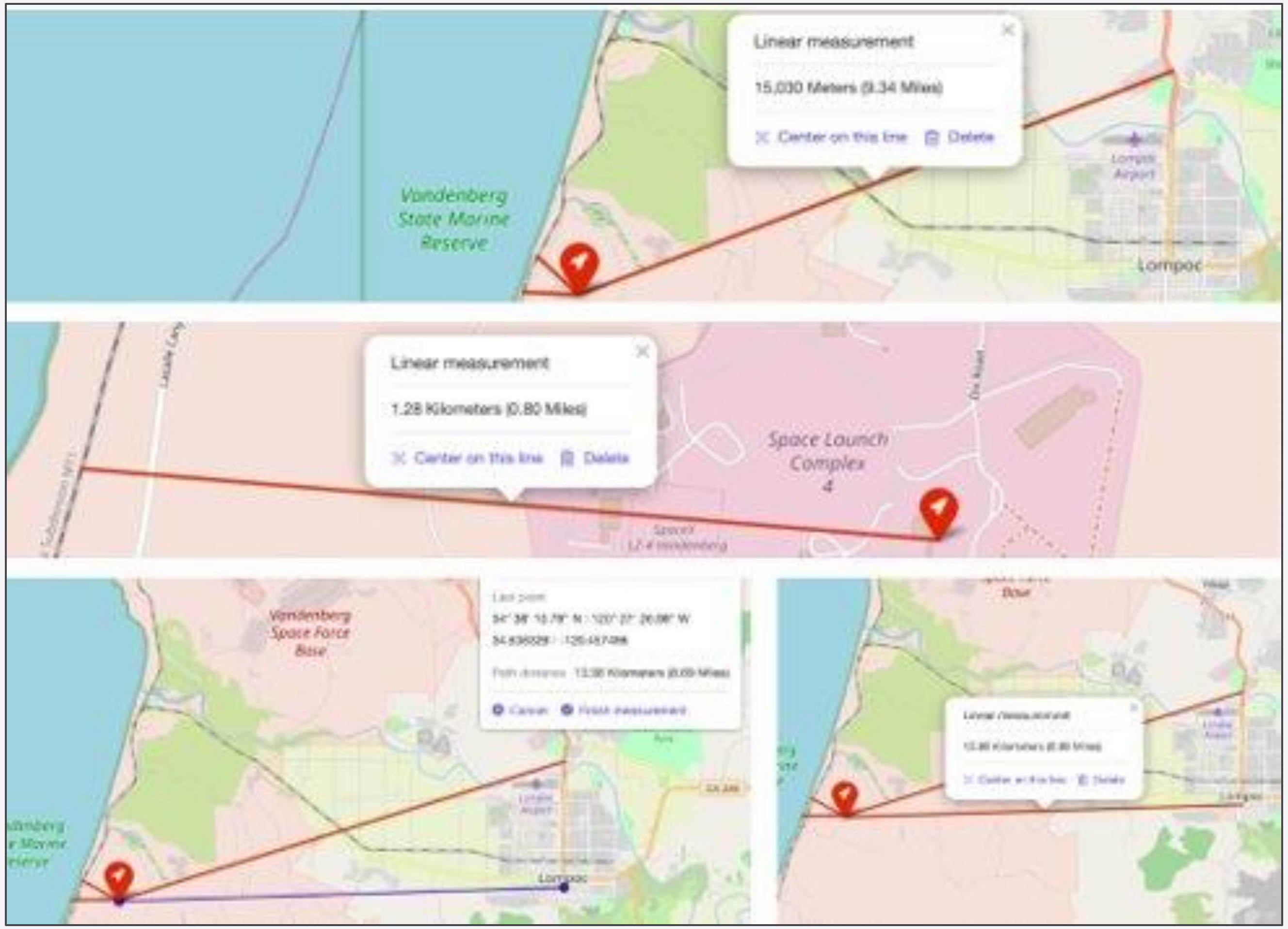
```
df["Class"].mean()*100
```

```
66.66666666666666
```

```
df.to_csv("dataset_part_2.csv", index=False)
```

Appendix

Proximities



Appendix

Proximities

Appendix

SQL query successful and failure mission outcomes.

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

Successful Mission
100

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

Failure Mission
1

Thank you!