IBM Developer
SKILLS NETWORK

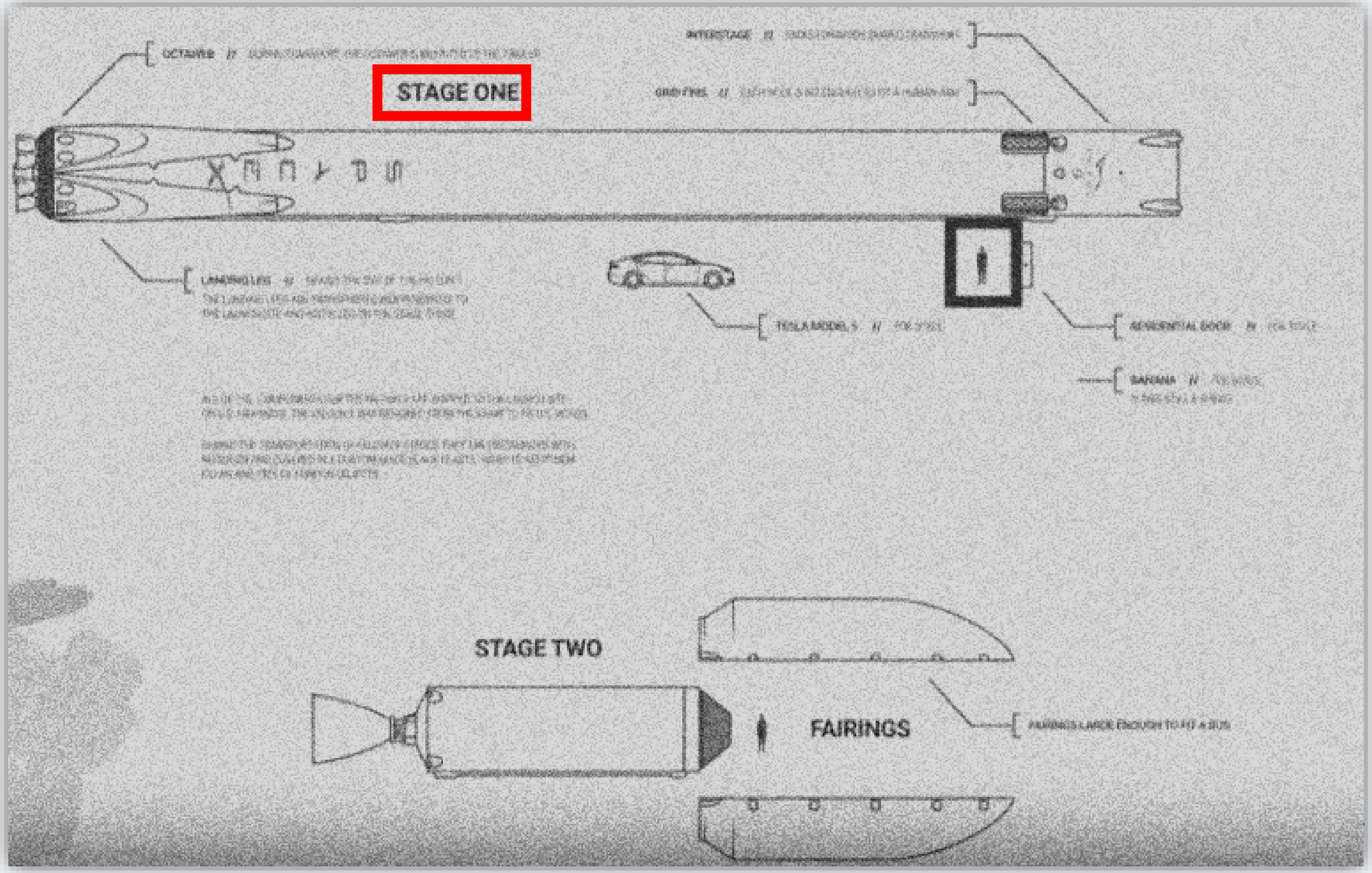
Winning Space Race with Data Science

Daniel E. Christello

November 2023

Outline

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



Presentation check-list (only for evaluation purpose) (*)

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

- ✓ Uploaded the URL of your GitHub repository including all the completed notebooks and Python files (1 pt)
- ✓ Uploaded your completed presentation in PDF format (1 pt)
- ✓ Completed the required Executive Summary slide (1 pt)
- ✓ Completed the required Introduction slide (1 pt)
- ✓ Completed the required data collection and data wrangling methodology related slides (1 pt)
- ✓ Completed the required EDA and interactive visual analytics methodology related slides (3 pts)
- ✓ Completed the required predictive analysis methodology related slides (1 pt)
- ✓ Completed the required EDA with visualization results slides (6 pts)
- ✓ Completed the required EDA with SQL results slides (10 pts)
- ✓ Completed the required interactive map with Folium results slides (3 pts)
- ✓ Completed the required Plotly Dash dashboard results slides (3 pts)
- ✓ Completed the required predictive analysis (classification) results slides (6 pts)
- ✓ Completed the required Conclusion slide (1 pts)
- ✓ Applied your creativity to improve the presentation beyond the template (1 pts)
- ✓ Displayed any innovative insights (1 pts)

(*) This slide is not for a business presentation, is only for a capstone.

Daniel E. Christello

Executive Summary

- Project Goal: **Estimate the cost of each launch of the Falcon 9 model rocket in the future based mainly on the prediction of the landing success and reusability of the first stage of the rocket:**

Summary of methodologies

- Gathering information about Space X, using public information.
- Analyzing the data and creating visualization and discussion panels.
- Training a machine learning model to predict whether SpaceX will reuse the first stage.

- Collecting data from public sources:
  - SpaceX API
  - and by scrapping SpaceX Wikipedia page
    - using Beautiful soup library for it.

- Analyzing meaningfull data will be used for machine learning models
  - Landing Outcome, Booster Versión, Date, FlightNumber, etc.

- Creating labels column 'Class' to denote all the successful landings.

- Performing EDA (Exploratory Data Analysis) using SQL magic.
  - Exploring with queries using group by, aggregate functions and so on.
  - Using Interactive visual analytics. (Plotly Dashboard, Folium)

- Converting categorical variables using one hot encoding, preparing the data for a machine learning model that will predict if the first stage will successfully land.

Summary of results

It's observed that:

- the success rate since 2013 has improved. (via launch number)
- different launch sites have different success rates.(via launch site)
- if we overlay the result of the landing outcomes as a color we see that CCAFS LC-40, has a success rate of 60%, but if the mass is above 10,000 kg the success rate is 100%.

Therefore, it is possible combine multiple features into different machine learning models:

#	method	score	ranking
1	tree	0.901786	Decision Tree
2	KNN	0.848214	K Nearest Neighbors
3	svm	0.848214	Support Vector Machine
4	log	0.846429	Logistic Regression

- Best Accuracy Score is given by Decision Tree Method (0,9 aprox)
- The rest of the algorithms used have almost the same result (0.85 aprox).
- In any case an accuracy score greater than 0.83 is a good indicator of the result of each method.

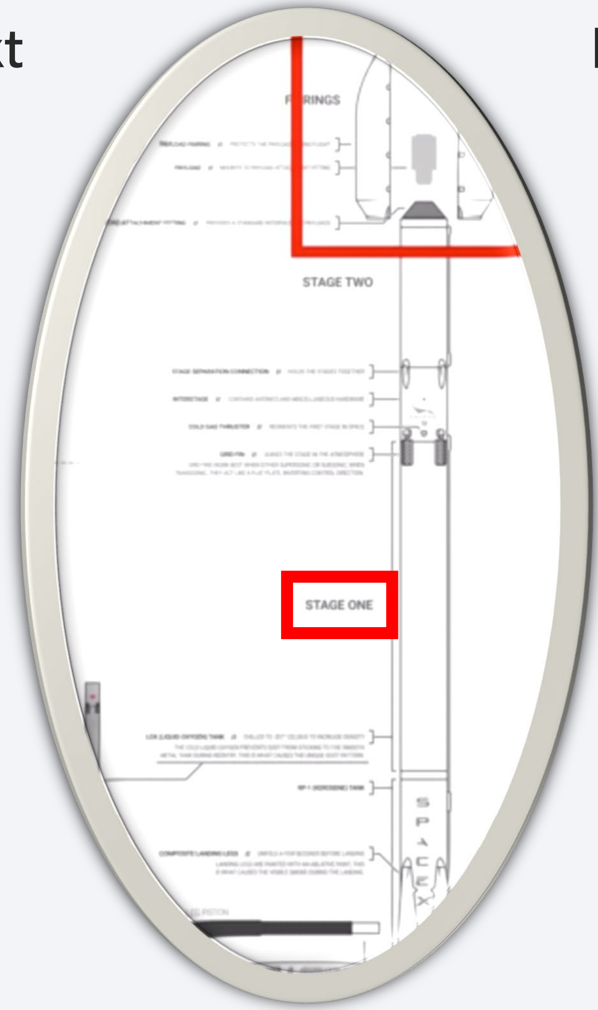
Daniel E. Christello

Introduction

Project background and context

We conducted our study based on the Space X Falcon 9 experience because:

- ❑ Space X is one of the most **experienced** rocket manufacturers and has achieved achievements in:
  - Trips to the International Space Station
  - Starlink Internet Constellation Building
  - Manned missions to space
- ❑ The Falcon 9 series has been successful in terms of total **cost**:
  - In almost all rockets, the second stage helps put the payload into orbit, but most of the work is done by the **first stage**.
  - Falcon 9 rocket launches **cost** about $62 million; Other providers cost more than $165 million each; Much of the savings is because SpaceX can reuse **the first stage**.



Problems to find answers

It is possible to predict the success or failure of the landing of future launches and this helps us determine the costs of each space expedition?

- ❑ What are the **data necessary** to make these predictions and what are the data sources that can be used to build the data set?
  - What can be observed from the available data and which of these data should be used to make future estimates
- ❑ What are the **machine learning** methods that can be used to estimate the success and failure of upcoming space expeditions?
  - There are methods with a sufficient level of precision to perform future calculations.
  - Which of these methods is the most appropriate or reliable to use in the future?

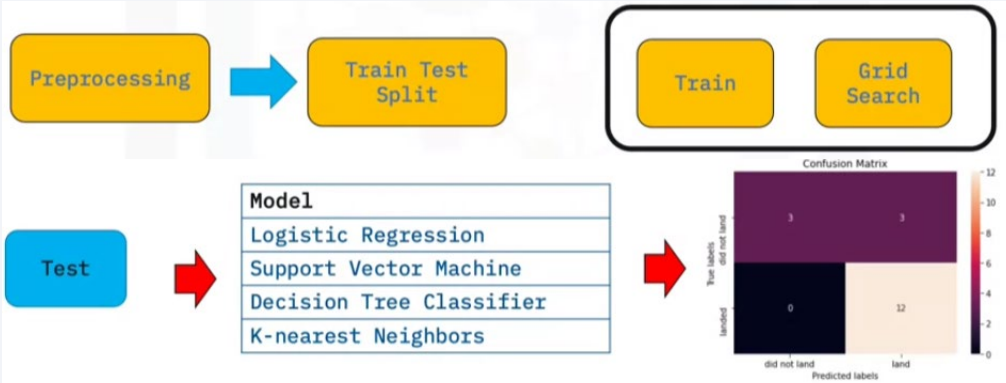
Section 1

Methodology

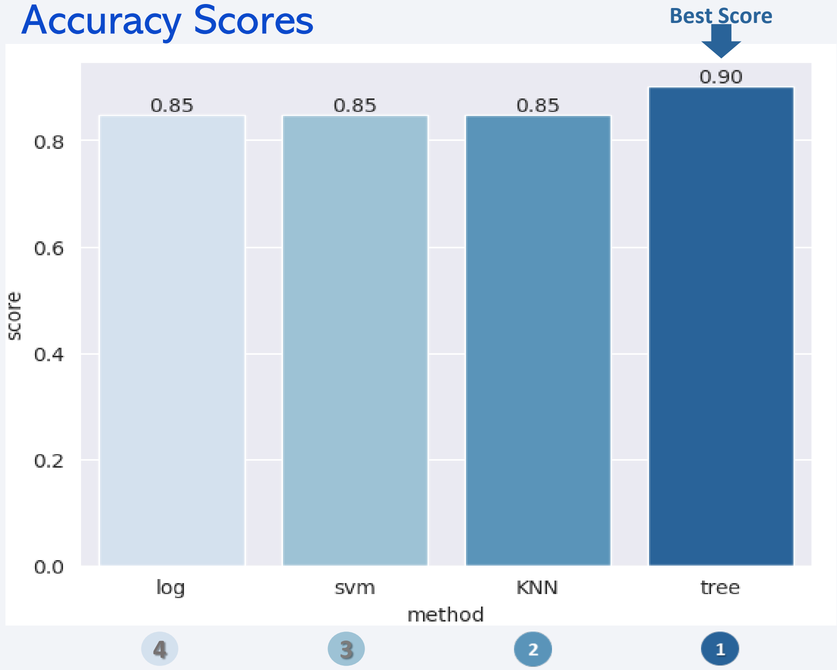
Methodology

Executive Summary

- Data collection methodology:
- Collecting data from public sources:
  - SpaceX site through API
  - SpaceX Wikipedia page using Beautiful soup library for it.
- Perform data wrangling
- Cleaning and Analyzing meaningfull data will be used for ML models
- Creating labels column ‘Class’ which denotes all the successful landings. Filtering.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Creating Machine Learning Models



Accuracy Scores



Data Collection – SpaceX API

Flowchart of SpaceX API calls reference 1 to 7

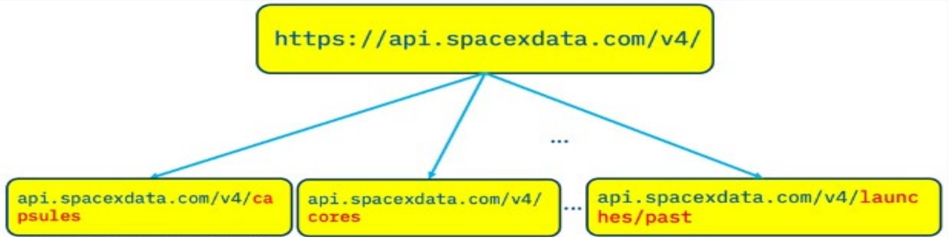
0 • GitHub URL of the completed SpaceX API calls notebook:

- https://github.com/DaniCh-Coder/IBM-Applied-Data-Science-Capstone/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

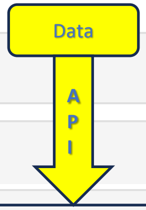
1 • GitHub URL of the Space-X API:

- https://github.com/r-spacex/SpaceX-API

2



Now let's start requesting rocket launch data from SpaceX API with the following URL:




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

3

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

4

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

Using the dataframe `data` print the first 5 rows

```
# Get the head of the dataframe
data.head(5)
```

static_fire_date_utc	static_fire_date_unix	net	window	rocke

5 • Once we 've got the dataframe we can get data like:

- Booster version form Rocket, Payload from PayloadMass, Launch site from Launchpad and all the data we need to the study

6 • Dealing with Missing Values

- We must deal with these missing values in PayloadMass.
- The LandingPad column will retain None values to represent when landing pads were not used.

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	

FlightNumber	0
Date	0
BoosterVersion	0
PayloadMass	0
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	

7 • Export Dataframe into a CSV

- Data_falcon9.to_csv('dataset_part_1.csv', index=False)

Daniel E. Christello

Data Collection - Scraping

Flowchart of Web Scraping reference from 1 to 3

1 Request Wiki page

a) Use request.get() method.

b) Assign the response to a object.

c) Create a BeautifulSoup object.

First, let's perform an HTTP GET method to request the Falcon9 Launch

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

<Response [200]>

Create a `BeautifulSoup` object from the HTML `response`

```
# Use BeautifulSoup() to create a BeautifulSoup object from a
resp_html = response.text
soup = BeautifulSoup(resp_html)
```

Print the page title to verify if the `BeautifulSoup` object was created

```
# Use soup.title attribute
soup.title
```

2 Extract column names

a) Look for the relevant content tables.

b) Iterate each element table to get column name.

```
# Use the find_all function in the BeautifulSoup obje
# Assign the result to a list called `html_tables`
html_tables = soup.find_all(name='tr')
```

Starting from the third table is our target table contains the ac

```
# Let's print the third table and check its content
first_launch_table = html_tables[2]
print(first_launch_table)
```

```
<tr>
<th scope="col">Flight No.
</th>
```

3 Parse HTML tables

a) Create a dictionary with keys.

b) Iterate on rows to parse.

c) fill each column of a dictionary.

d) Create a dataframe.

```
extracted_row = 0
#Extract each table
for table_number,table in enumerate(soup.find_all('table',"wikitable plainrowheaders collapsible")):
   # get table row
    for rows in table.find_all("tr"):
        #check to see if first table heading is as number corresponding to launch a number
        if rows.th:
            if rows.th.string:
                flight_number=rows.th.string.strip()
                flag=flight_number.isdigit()
        else:
            flag=False
        #get table element
        row=rows.find_all('td')
        #if it is number save cells in a dictonary
        if flag:
            extracted_row += 1
            # Flight Number value
            # TODO: Append the flight_number into launch_dict with key `Flight No.`
            launch_dict['Flight No.'].append(flight_number)

            # print(launch_dict['Flight No.'])
            # print(flight_number)

            datatimelist=date_time(row[0])
```

Git Hub URL Web Scraping Notebook: https://github.com/DaniCh-Coder/IBM-Applied-Data-Science-Capstone/blob/main/jupyter-labs-webscraping.ipynb

Falcon 9 Wikipedia Page : https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches

Daniel E. Christello

Data Wrangling

Flowchart of Data Wrangling reference from 1 to 4.

- Exploratory Data Analysis (EDA) to find patterns in the data and determine what would be the label for training supervised models.

Git Hub URL Data Wrangling Notebook: https://github.com/DaniCh-Coder/IBM-Applied-Data-Science-Capstone/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

1

Number of **launches** on **each site**

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

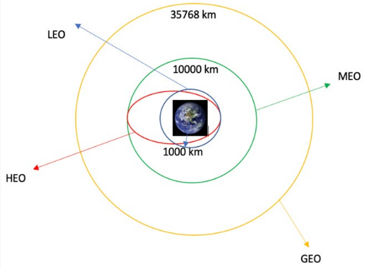
```
CCAFS SLC 40    55
KSC LC 39A      22
VAFB SLC 4E     13
Name: LaunchSite, dtype: int64
```

2

Number and occurrence of **each orbit**

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

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



3

Number and occurence of mission **outcome of the orbits**

```
for i,outcome in enumerate
    print(i,outcome)
```

```
0 True ASDS
1 None None
2 True RTLS
3 False ASDS
4 True Ocean
5 False Ocean
6 None ASDS
7 False RTLS
```

4

Outcome label from Outcome column

```
# landing_class = 0 if bad_outcome
# landing_class = 1 otherwise
landing_class = [ 1 if 'True' in x else 0 f
landing_class
```

```
[0,
 0,
 0,
 0,
 0,
 0,
 1,
 1,
```

EDA with SQL

Git Hub URL EDA with SQL Notebook: https://github.com/DaniCh-Coder/IBM-Applied-Data-Science-Capstone/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

```
%sql create table SPACEXTABLE as select * from SPACEXTBL where Date is not null
```

SQL summary

10 queries

Daniel E. Christello

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

SELECT DISTINCT "Launch_Site" FROM SPACEXTABLE

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

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

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

%sql SELECT SUM("PAYLOAD_MASS__KG_") FROM SPACEXTABLE where Payload LIKE "%CRS%"

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

%sql SELECT SUM("PAYLOAD_MASS__KG_") FROM SPACEXTABLE where Booster_Version = "F9 v1.1"

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

%sql SELECT MIN("Date") FROM SPACEXTABLE WHERE Landing_Outcome LIKE "%Success%“

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

%sql SELECT DISTINCT Booster_Version FROM SPACEXTABLE WHERE Landing_Outcome LIKE "%Success%" AND "PAYLOAD_MASS__KG_" BETWEEN 4000 AND 6000

List the total number of successful and failure mission outcomes

%sql SELECT COUNT("*") FROM SPACEXTABLE WHERE Mission_Outcome LIKE "%Success%" OR Mission_Outcome LIKE "%Failure%"

List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.

%%sql SELECT SUBSTR(Date,0,5) AS "Year", SUBSTR(Date, 6,2) AS "Month" , "Launch_Site", "Booster_Version", "Landing_Outcome"
FROM SPACEXTABLE WHERE SUBSTR(Date, 0, 5)='2015' AND Landing_Outcome LIKE "%Failure (drone ship)%" ORDER BY SUBSTR(Date, 6,2)

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.

%%sql SELECT "Landing_Outcome", COUNT(*) FROM (SELECT * FROM SPACEXTABLE WHERE "Landing_Outcome" <> "No attempt") WHERE SUBSTR(Date, 0, 10) BETWEEN '2010-06-04' AND '2017-03-20’
GROUP BY "Landing_Outcome"

Build an Interactive Map with Folium

Reference: Object (Green). Why (Blue).

Git Hub URL Folium Notebook: https://github.com/DaniCh-Coder/IBM-Applied-Data-Science-Capstone/blob/main/lab_jupyter_launch_site_location.jupyterlite.ipynb

Objects Summary

- Circle
- Marker
- Popup
- Polyline
- Circle.Marker

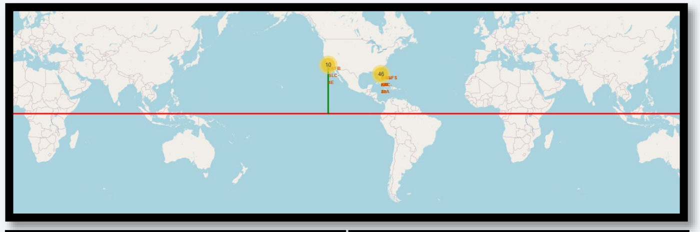
Sites Locations

- *For each launch site, add a Circle object based on its coordinate (Lat, Long) values. In addition, add Launch site name as a popup label.*
- *This is the quickest way to mark sites on map.*

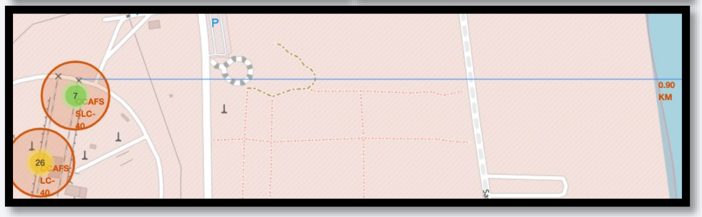
Launch Success/Failure

- *Using Marker Cluster to Group*
- *To make a clean visualizatión of all success and failures*

Site Distances

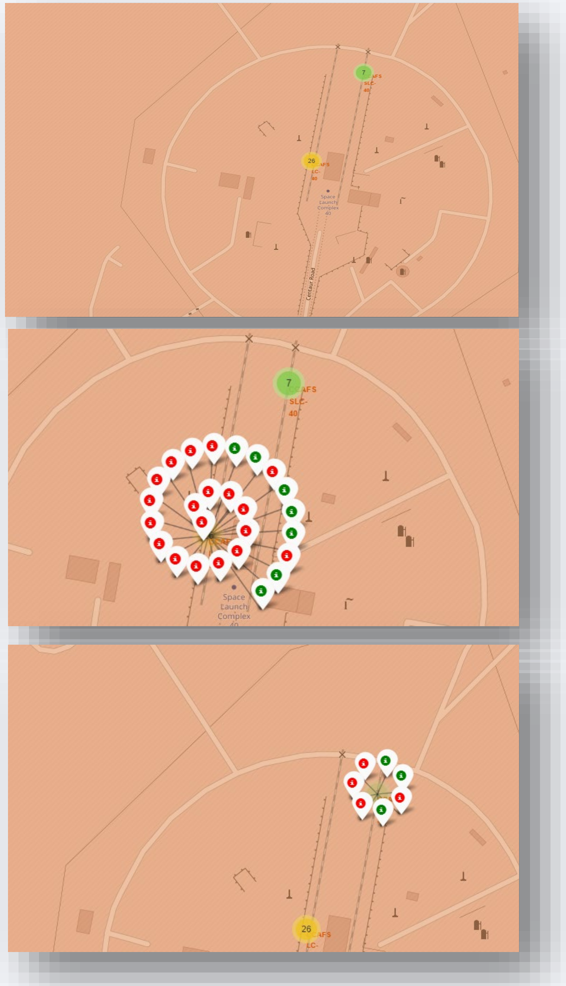



```
def haversine(lat1, lon1, lat2, lon2):
    rad=math.pi/180
    dlat=lat2-lat1
    dlon=lon2-lon1
    R=6372.795477598
    a=(math.sin(rad*dlat/2))**2 + math.cos
    distancia=2*R*math.asin(math.sqrt(a))
    return distancia
```



- *Using Polyline to draw distances to equator and the coast*
- *Building a haversine function to calculate distances*

Success/Failure Zoom



Build a Dashboard with Plotly Dash

Reference **Graph** Why.

Git Hub URL EDA with SQL Notebook: https://github.com/DaniCh-Coder/IBM-Applied-Data-Science-Capstone/blob/main/spacex_dash_app.py

Building Summary

- Graph
  - Pie Chart **Success Count**
    - All launch sites — Overall performance of success landing on sites
    - Selected launch site — Which site has the most success in landings
  - Scatter **Correlation Payload-Success**
    - All Sites — Overall relationship between Payload Mass and Success Launch
    - Selected site — To see what relationship exists between payload mass and success by site

Results of methodology applied

Recalling issues analysed

- Number of **launches** on **each site**
- Number and occurrence of **each orbit**
- Occurence of mission **outcome of the orbits**

```
CCAFS SLC 40    55   GTO      27   0 True ASDS
KSC LC 39A      22   ISS      21   1 None None
VAFB SLC 4E     13   VLEO     14   2 True RTLS
                     PO        9   3 False ASDS
                     LEO       7   4 True Ocean
                     SSO       5   5 False Ocean
                     MEO       3   6 None ASDS
                     ES-L1     1   7 False RTLS
                     HEO       1
                     SO        1
                     GEO       1
```

- 10 Questions analysis answered through **SQL**

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

MASS__KG_")
111268

MIN("Date")
2015-12-22

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

Year	Month	Launch_Site
2015	04	CCAFS LC-40
2015	10	CCAFS LC-40

Landing_Outcome	COUNT(*)
Controlled (ocean)	3
Failure (drone ship)	5
Failure (parachute)	1
Precluded (drone ship)	1

Results

It is possible combine multiple features into different machine learning models:

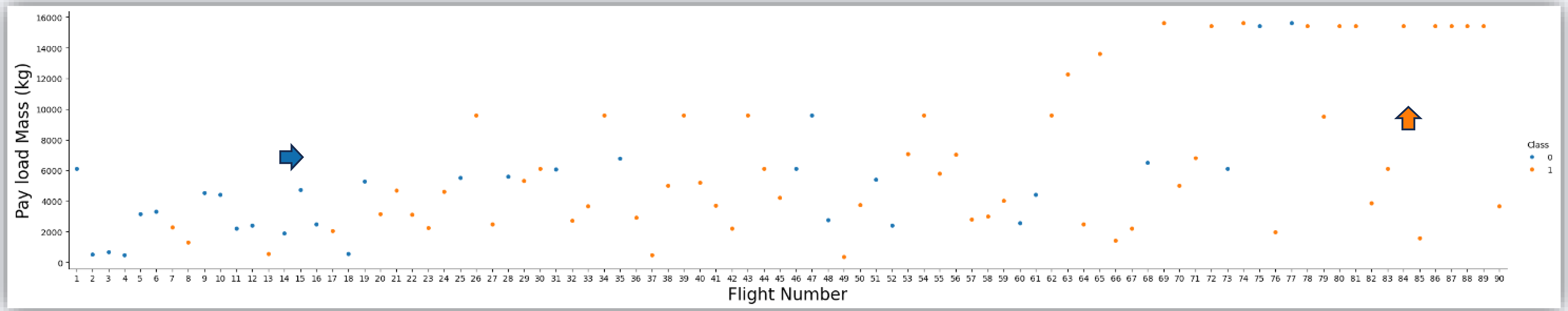
- ❑ There are multiple data that is valuable in order to build Machine Learning Models
- ❑ Several methods of Machine Lerning can be applied to find good estimations.
- ❑ In any case an accuracy score greater than 0.8 is a good indicator of the result of each method.
- ❑ In further steps we go through other analysis to understand the main variables. For example:
  - The quantity of flight has incidence on success improvement?
  - Is there any relation between payload mass and return success?
  - What are the incidence of the orbit in flights?
  - Witch other relationships and incidences can be founded?

Section 2

Insights drawn from EDA

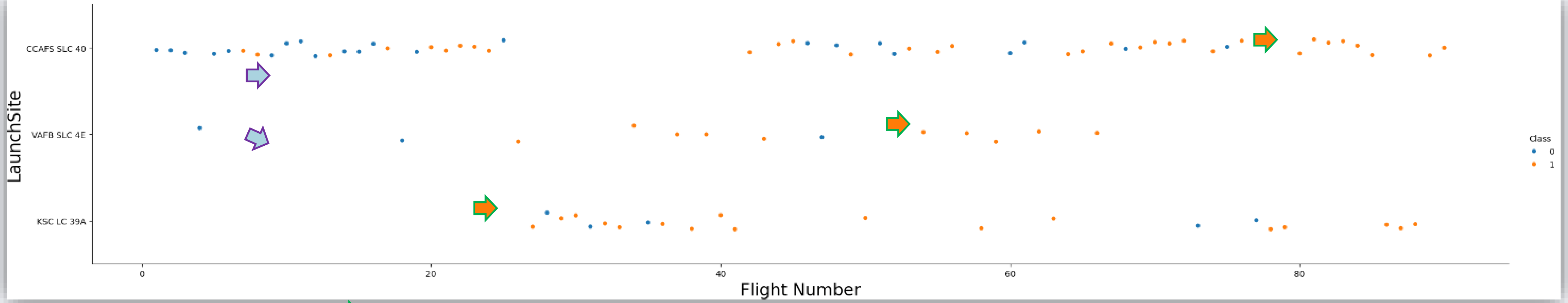
Flight Number, Payload Mass, Lauch Site and Success

Flight Number vs. Payload Mass



As the flight number increases, the first stage is more likely to land successfully.
The more massive the payload, the more likely the first stage will return.

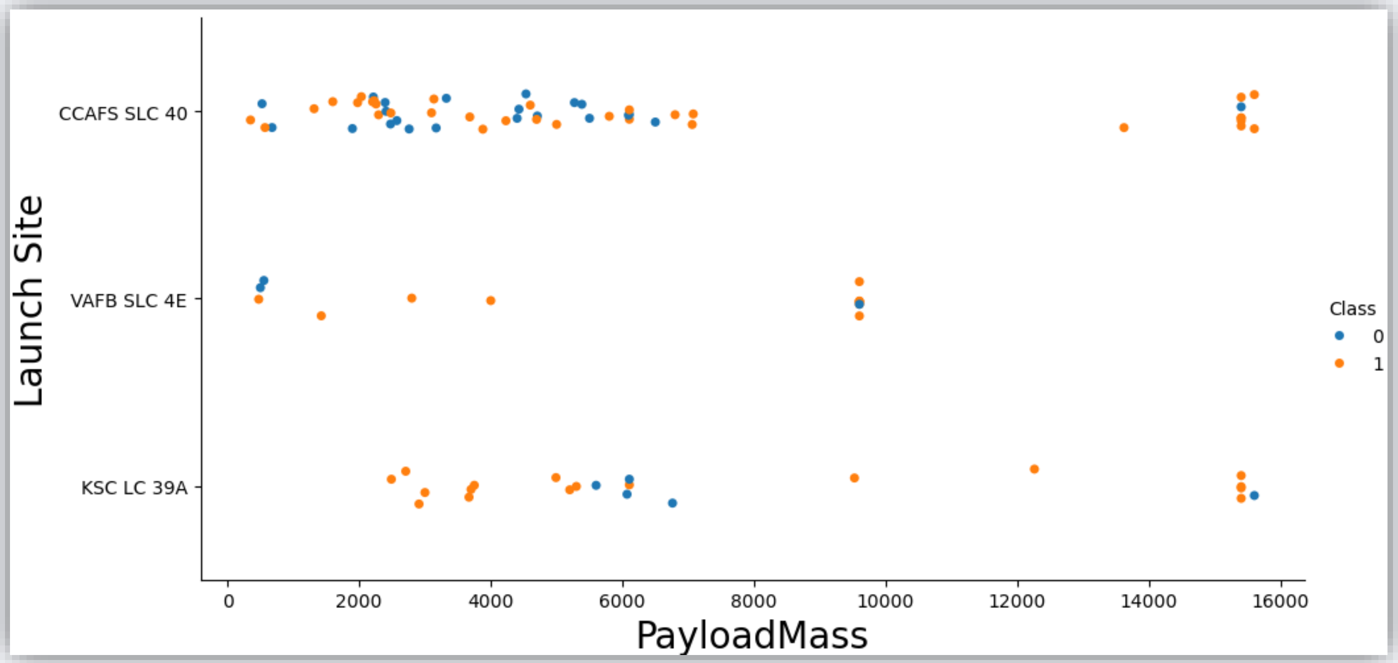
Flight Number vs. Launch Site



As the flight number increases, the probability of success increases on each site.
The experience of successes and failures is transferred between sites, resulting in a greater number of total successes.
- WAFB SLC 4E is the site with the fewest crashes between releases

Payload vs. Launch Site

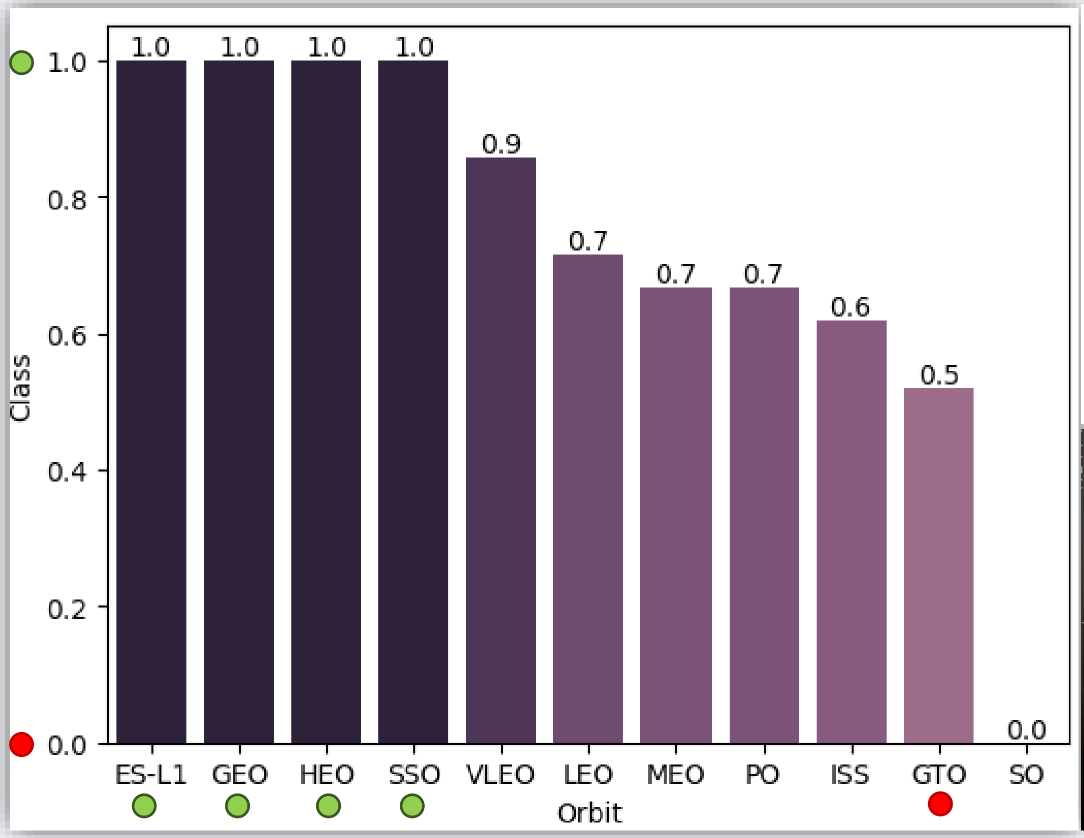
Payload vs. Launch Site

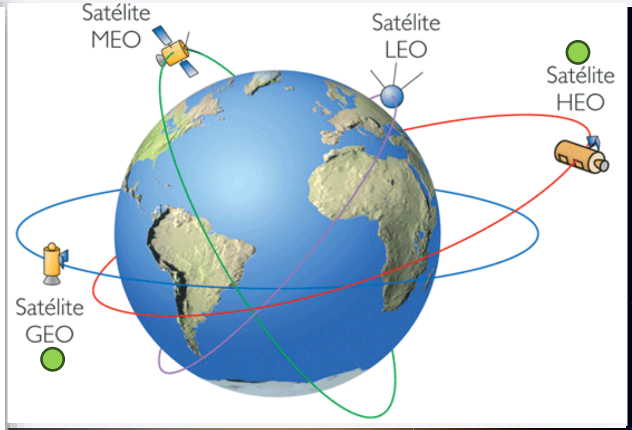


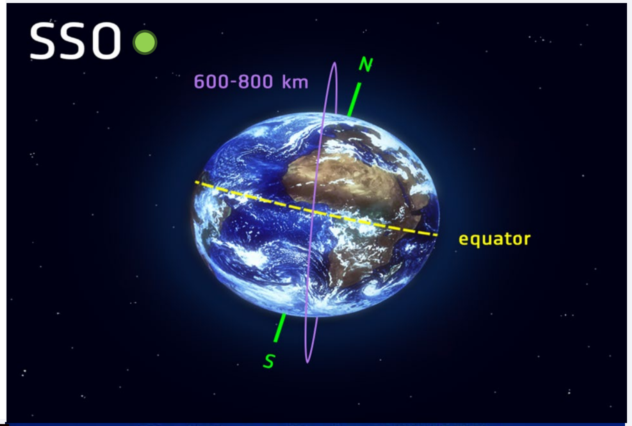
- In VAFB-SLC launchsite there are no rockets launched for heavypayload mass (greater than 10000)

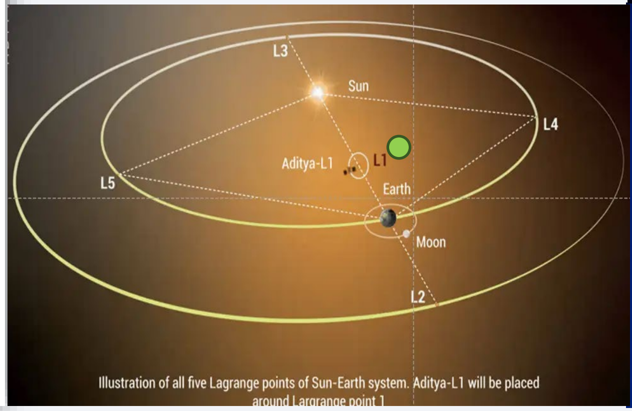
Daniel E. Christello

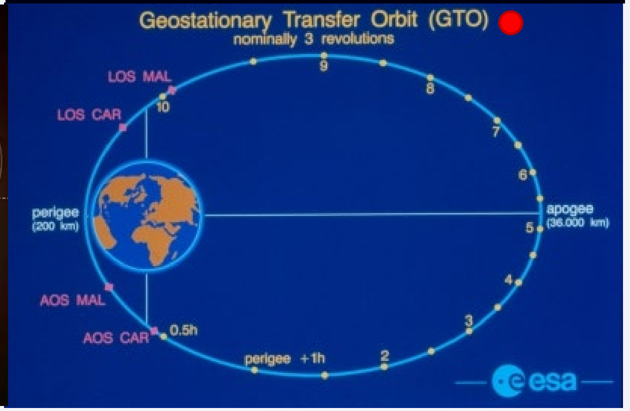
Success Rate vs. Orbit Type







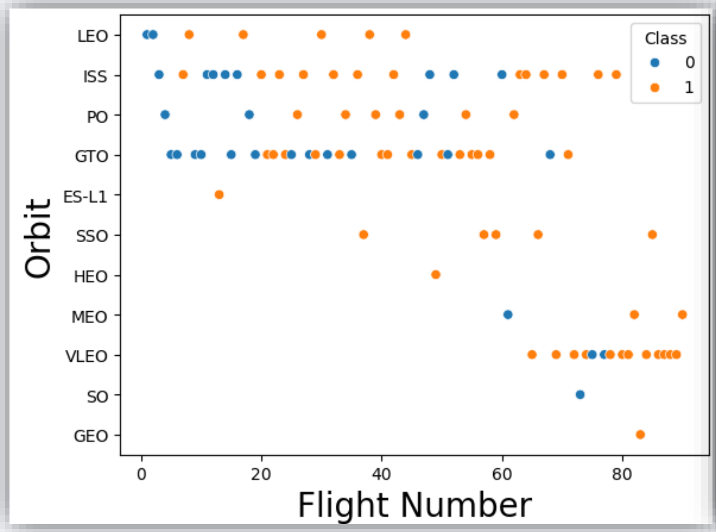




- ES-L1, GEO, HEO, SSO and VLEO have *highest* success rate.
- GTO and ISS have *lowest* success rate.
- LEO, MEO, PO have 0,7 success rate.

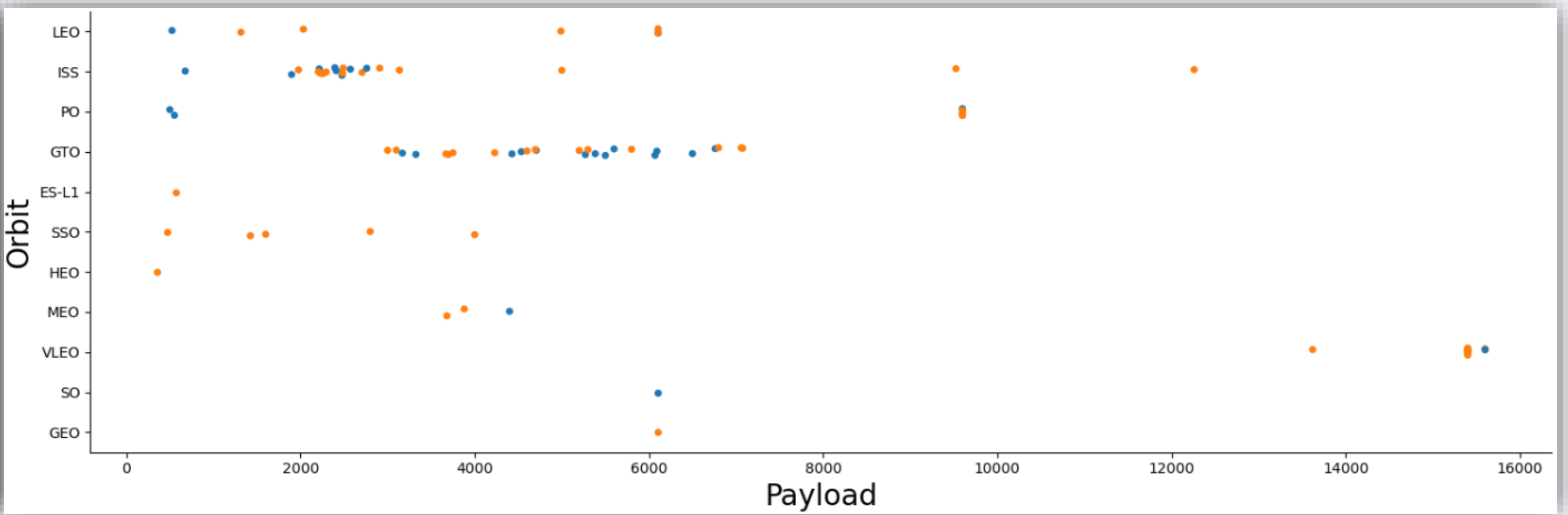
Orbit and its relation with payload and number of flights

Flight Number vs Orbit Type



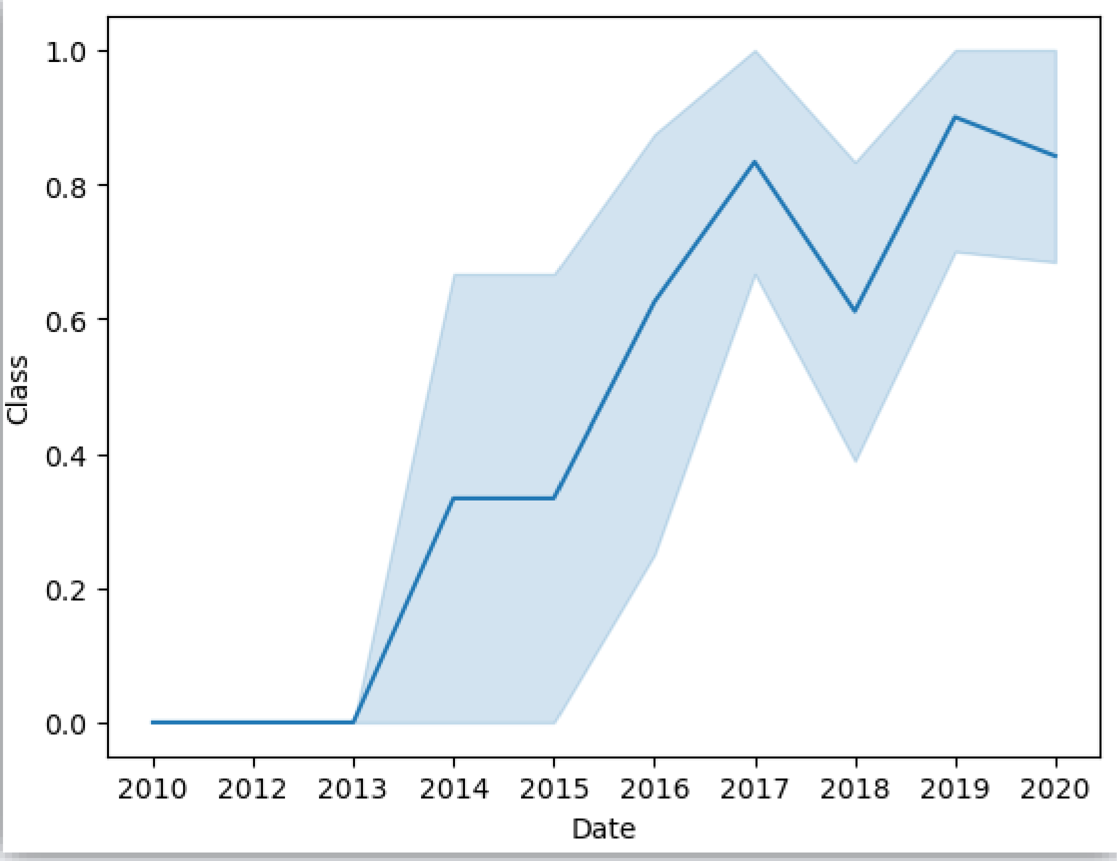
- LEO, VLEO: Strong learning experience. As flight number increases success increases.
- GTO: There is no relationship between flight number and success.

Payload vs Orbit Type



- PO, LEO, ISS: have better landing performance with heavy payloads
- GTO: There is no relationship payload and orbit, related to successful or unsuccessful landing.

Launch Success Yearly Trend



Date	Class
2010	0.000000
2012	0.000000
2013	0.000000
2014	0.333333
2015	0.333333
2016	0.625000
2017	0.833333
2018	0.611111
2019	0.900000
2020	0.842105

- Since 2013 the success rate is increasing.

All Launch Site Names

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

```
%sql SELECT DISTINCT "Launch_Site" FROM SPACEXTABLE
```

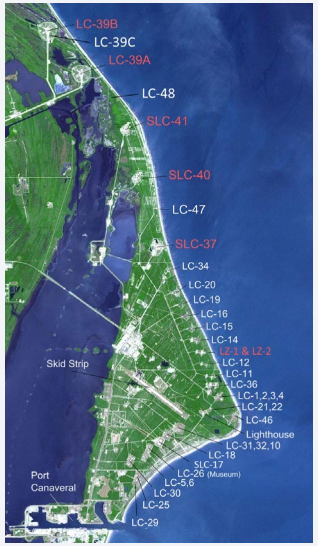
```
 * sqlite:///my_data1.db
Done.
```

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

Explanation

The data contains several Space X launch facilities:

- Cape Canaveral Space Launch Complex 40
- **VAFB SLC 4E** , Vandenberg Air Force Base Space Launch Complex 4E **(SLC-4E)**, Kennedy Space Center Launch Complex 39A
- **KSC LC 39A** .The location of each Launch Is placed in the column



Launch Site Names Begin with 'CCA'

```
%sql SELECT * FROM SPACEXTABLE 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
2010-04-06	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-08-12	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
2012-08-10	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-01-03	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

- Many of the launches are made on Cape Canaveral (CC).
- Looking on Lanuch_Site column is possible to find all the Launching on CC but list only five with LIMIT 5.
- The five launches listed are made to LEO and LEO (ISS).
- Landing Outcome were not successful.

Total Payload Mass

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

```
%sql SELECT SUM("PAYLOAD_MASS__KG_") FROM SPACEXTABLE where Payload LIKE "%CRS%"
```

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

SUM("PAYLOAD_MASS__KG_")
111268

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

```
%sql SELECT SUM("PAYLOAD_MASS__KG_") FROM SPACEXTABLE where Booster_Version = "F9 v1.1"
```

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

SUM("PAYLOAD_MASS__KG_")
14642

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

Hint:Use min function

```
%sql SELECT MIN("Date") FROM SPACEXTABLE WHERE Landing_Outcome LIKE "%Success%"
```

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

MIN("Date")
2015-12-22

Analysis of success

Successful Drone Ship Landing with Payload between 4000 and 6000

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

```
%sql SELECT DISTINCT Booster_Version FROM SPACEXTABLE WHERE Landing_Outcome LIKE "%Success%" AND "PAYLOAD_MASS__KG_" BETWEEI
```

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

Booster_Version
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
F9 FT B1032.1
F9 B4 B1040.1
F9 FT B1031.2
F9 B4 B1043.1
F9 B5 B1046.2
F9 B5 B1047.2
F9 B5 B1046.3
F9 B5 B1048.3
F9 B5 B1051.2
F9 B5B1060.1
F9 B5 B1058.2
F9 B5B1062.1

- There are 15 different booster version where payload mass is between 4000 and 6000 and landing outcome were successful.

Total Number of Successful and Failure Mission Outcomes

List the total number of successful and failure mission outcomes

```
%sql SELECT COUNT("*") FROM SPACEXTABLE WHERE Mission_Outcome LIKE "%Success%" OR Mission_Outcome LIKE "%Failure%"
```

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

COUNT("*")
101

- There are 101 Successful and failure mission outcomes.
- The others are not attempt.

Boosters and 2015 launch records

Boosters Carried Maximum Payload

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

```
%sql SELECT DISTINCT "Booster_Version" FROM SPACEXTABLE WHERE "PAYLOAD_MASS__KG_" = (SELECT MAX("PAYLOAD_MASS__KG_") FROM SP
```

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

Booster_Version
F9 B5 B1048.4
F9 B5 B1049.4
F9 B5 B1051.3
F9 B5 B1056.4
F9 B5 B1048.5
F9 B5 B1051.4
F9 B5 B1049.5
F9 B5 B1060.2
F9 B5 B1058.3
F9 B5 B1051.6
F9 B5 B1060.3
F9 B5 B1049.7

- There are 12 different booster version which have carried the maximum payload mass.

2015 Launch Records

List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.

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

```
%%sql SELECT SUBSTR(Date,0,5) AS "Year", SUBSTR(Date, 6,2) AS "Month" , "Launch_Site", "Booster_Version", "Landing_Outcome"
         WHERE SUBSTR(Date, 0, 5)='2015' AND Landing_Outcome LIKE "%Failure (drone ship)%" ORDER BY SUBSTR(Date, 6,2)
```

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

Year	Month	Launch_Site	Booster_Version	Landing_Outcome
2015	04	CCAFS LC-40	F9 v1.1 B1015	Failure (drone ship)
2015	10	CCAFS LC-40	F9 v1.1 B1012	Failure (drone ship)

- There are 2 launches on CC with Failure Landing Otucomes in 2015:
  - 1 on April
  - 1 on October

Daniel E. Christello

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

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

Task 10

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.

```
%%sql SELECT "Landing_Outcome", COUNT(*) FROM
            (SELECT * FROM SPACEXTABLE WHERE "Landing_Outcome" <> "No attempt")
        WHERE SUBSTR(Date, 0, 10) BETWEEN '2010-06-04' AND '2017-03-20'
        GROUP BY "Landing_Outcome"
```

There are 12 different booster version which have carried the maximum payload mass.

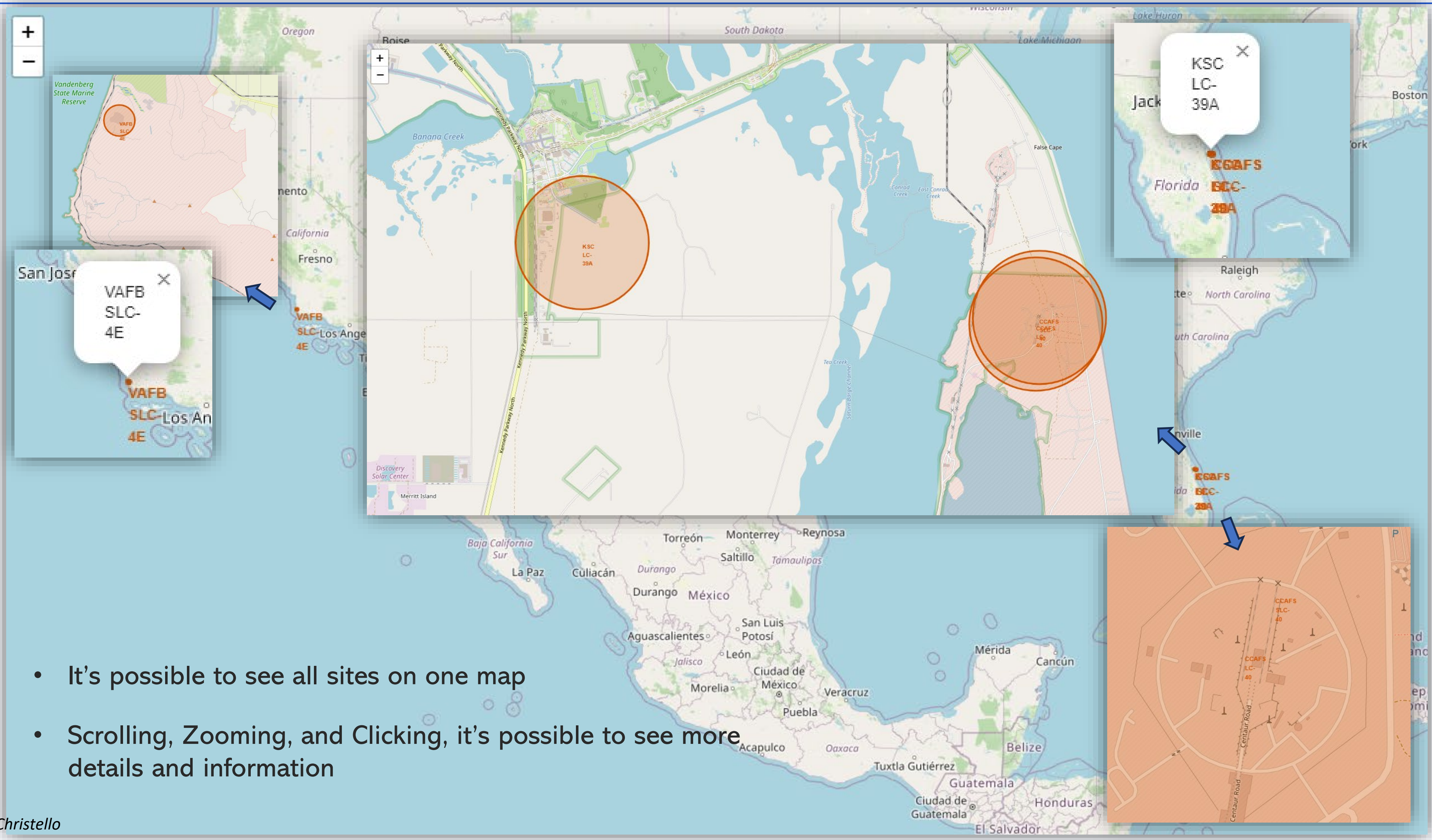
```
 * sqlite:///my_data1.db
Done.
```

Landing_Outcome	COUNT(*)
Controlled (ocean)	3
Failure (drone ship)	5
Failure (parachute)	1
Precluded (drone ship)	1
Success (drone ship)	5
Success (ground pad)	5
Uncontrolled (ocean)	2

Section 3

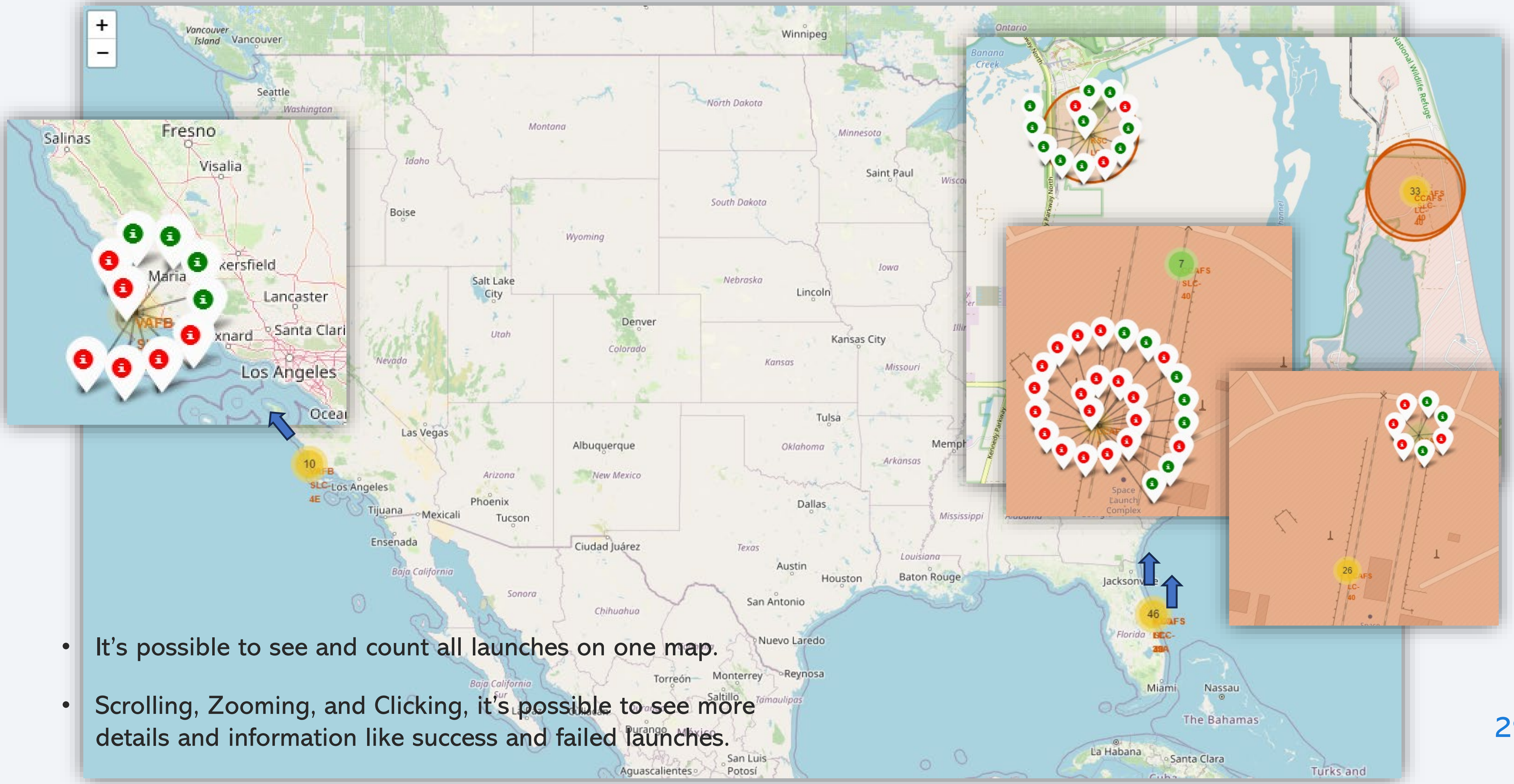
Launch Sites Proximities Analysis

All launch sites on a map¶



- It's possible to see all sites on one map
- Scrolling, Zooming, and Clicking, it's possible to see more details and information

Success/failed launches for each site on the map



- It's possible to see and count all launches on one map.
- Scrolling, Zooming, and Clicking, it's possible to see more details and information like success and failed launches.

Distances between a launch site to its proximities¶

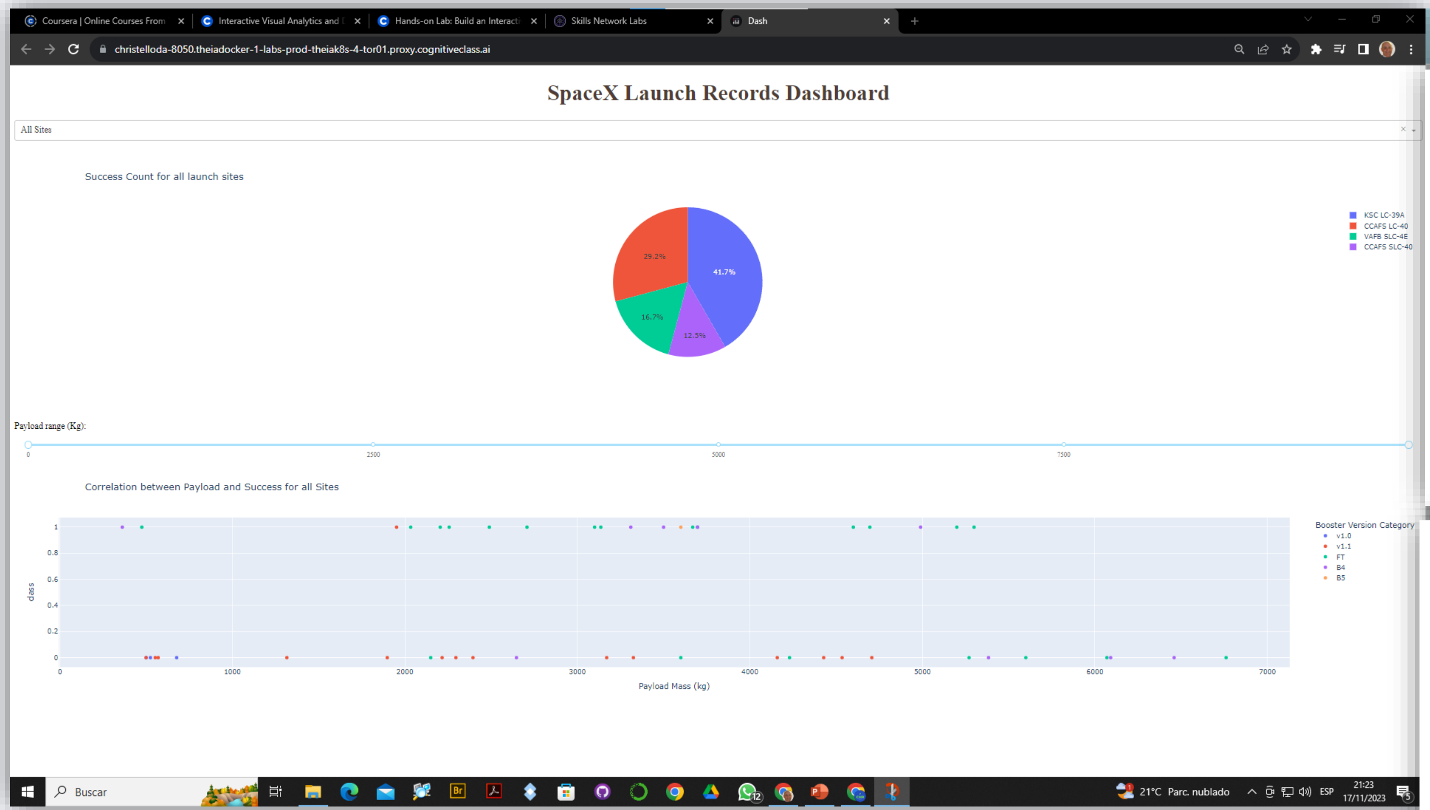
In the map you can see at a glance that:
* All sites are in locations close to the coast, at a distance of approximately 1 to 2 kilometers.
* All sites are far from the equator.
* The southernmost site is the one closest to the equator and is located at a distance of approximately: 3200km

Section 4

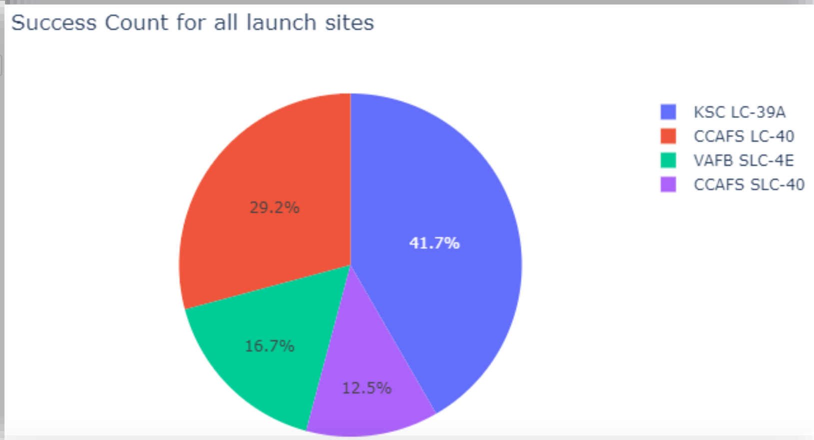
Build a Dashboard with Plotly Dash

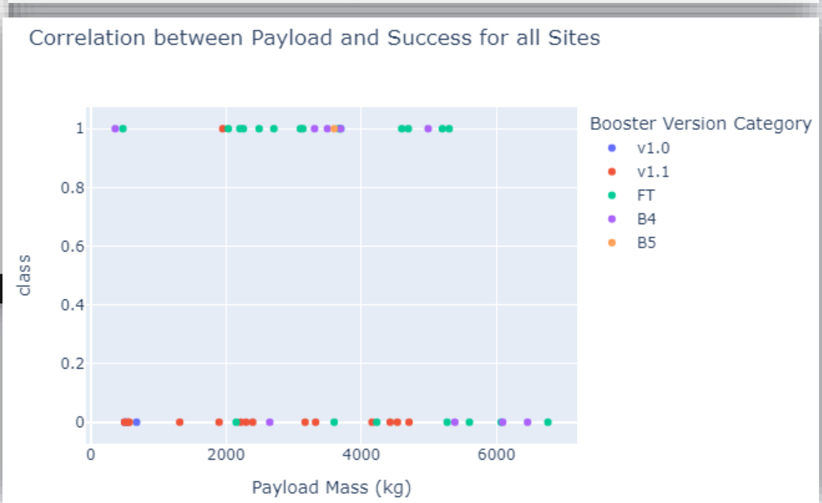
Space X Launch Record Dashboard

<Dashboard Screenshot 1>



Elements





Pie Chart - Count

Success Count Ranking
- 1° KSC LC-39A
- 2° CCAFS LC-40
- 3° WAFB SLC-4E
- 4° CCAFS SLC-40

Findings

Correlation - Scatter

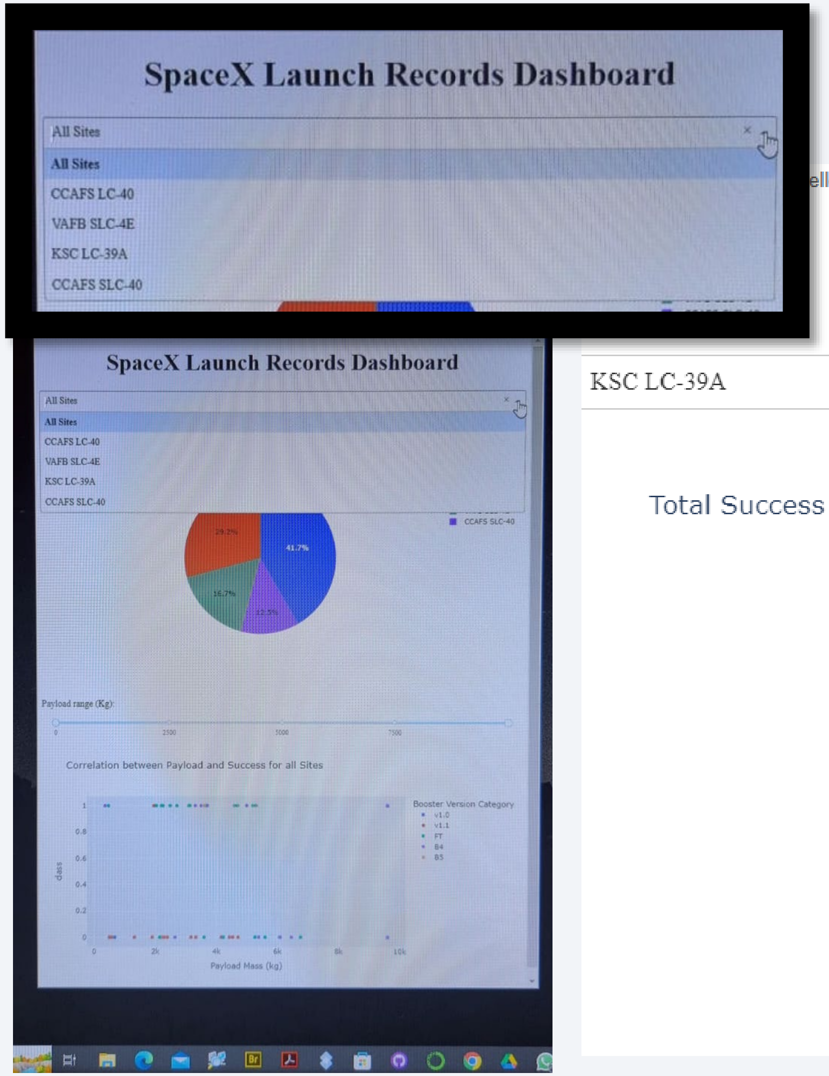
Booster Version Participation Relevance
- 1° FT and v1.1
- 2° v.1.0, B4 and B5

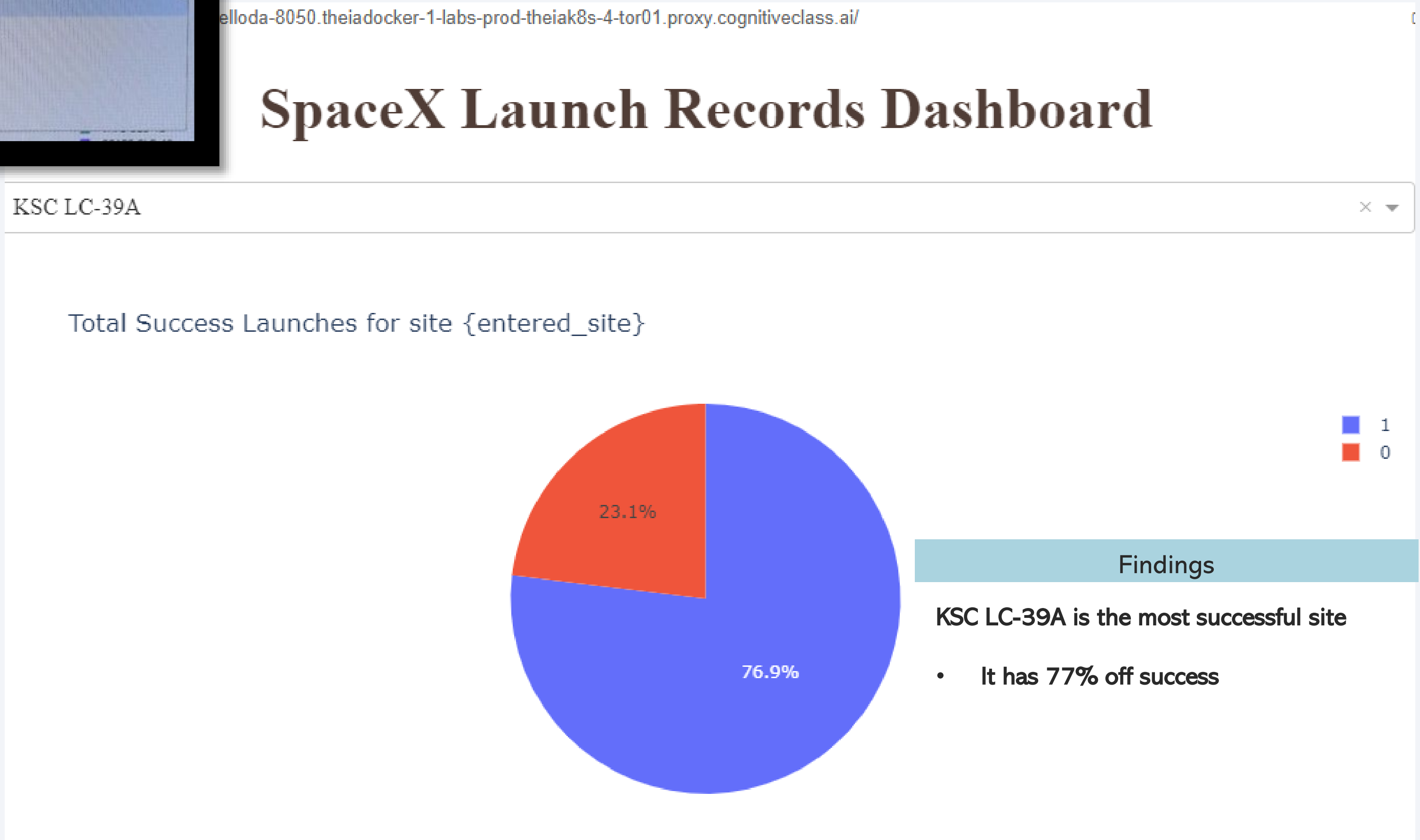
Booster Version and Payload
- B4 and FT is used with more payload
- All but v1.0 used between 2500 and 5000kg

Daniel E. Christello

Launch Site Drop Down Menu

<Dashboard Screenshot 2>





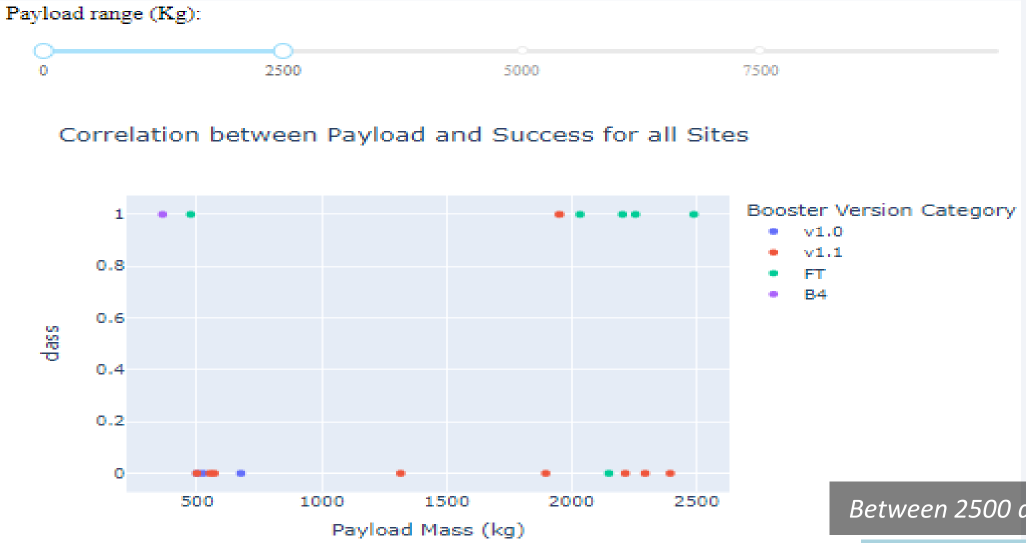
Findings

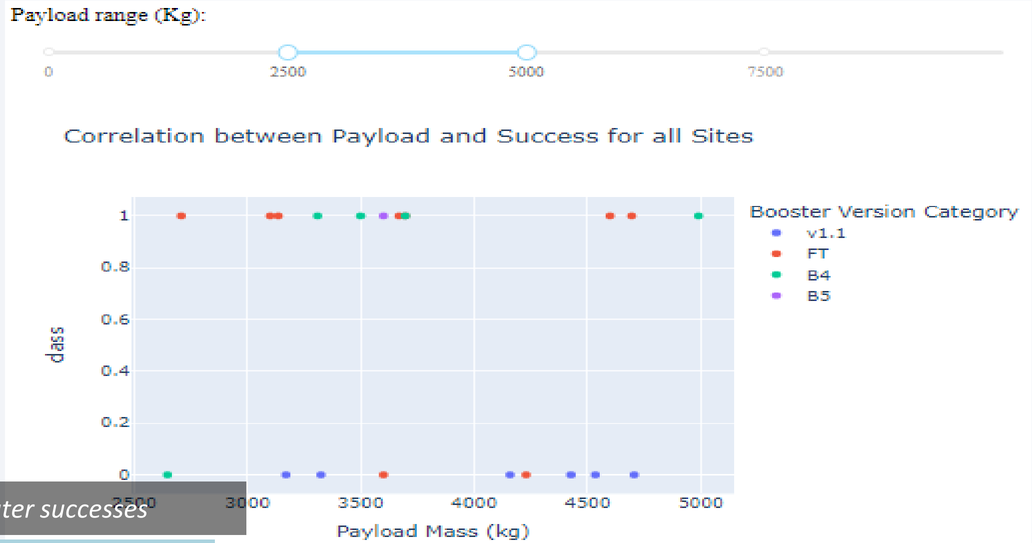
KSC LC-39A is the most successful site

- It has 77% off success

Range Slider to Select Payload

<Dashboard Screenshot 3>





Between 2500 and 5000 greater successes

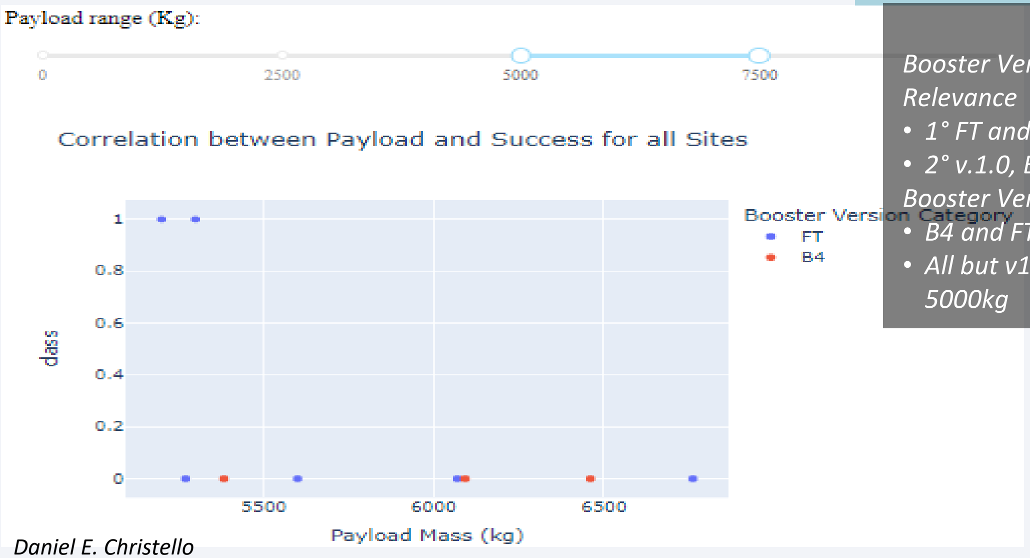
Findings

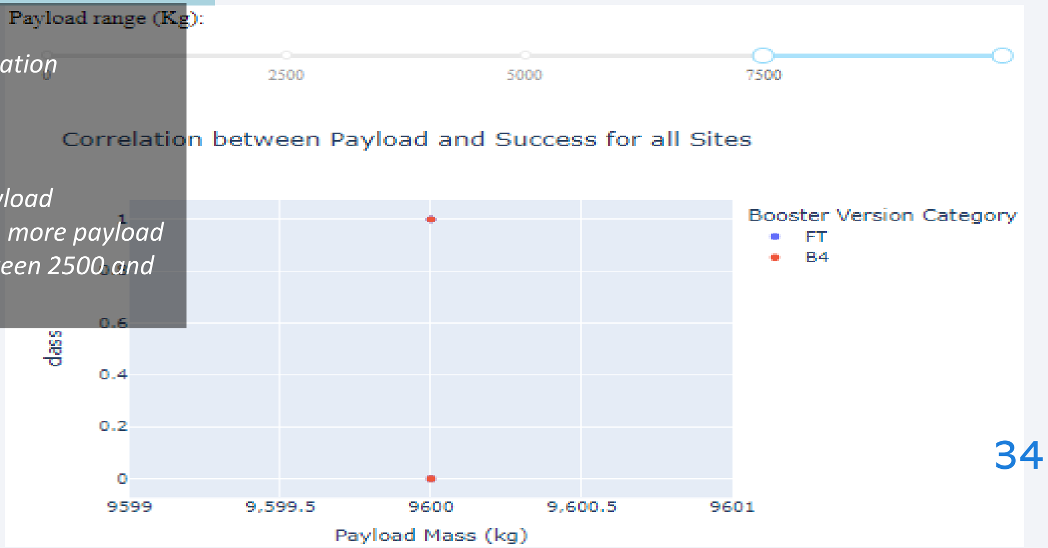
Booster Version Participation Relevance

- *1° FT and v1.1*
- *2° v.1.0, B4 and B5*

Booster Version and Payload

- *B4 and FT is used with more payload*
- *All but v1.0 used between 2500 and 5000kg*





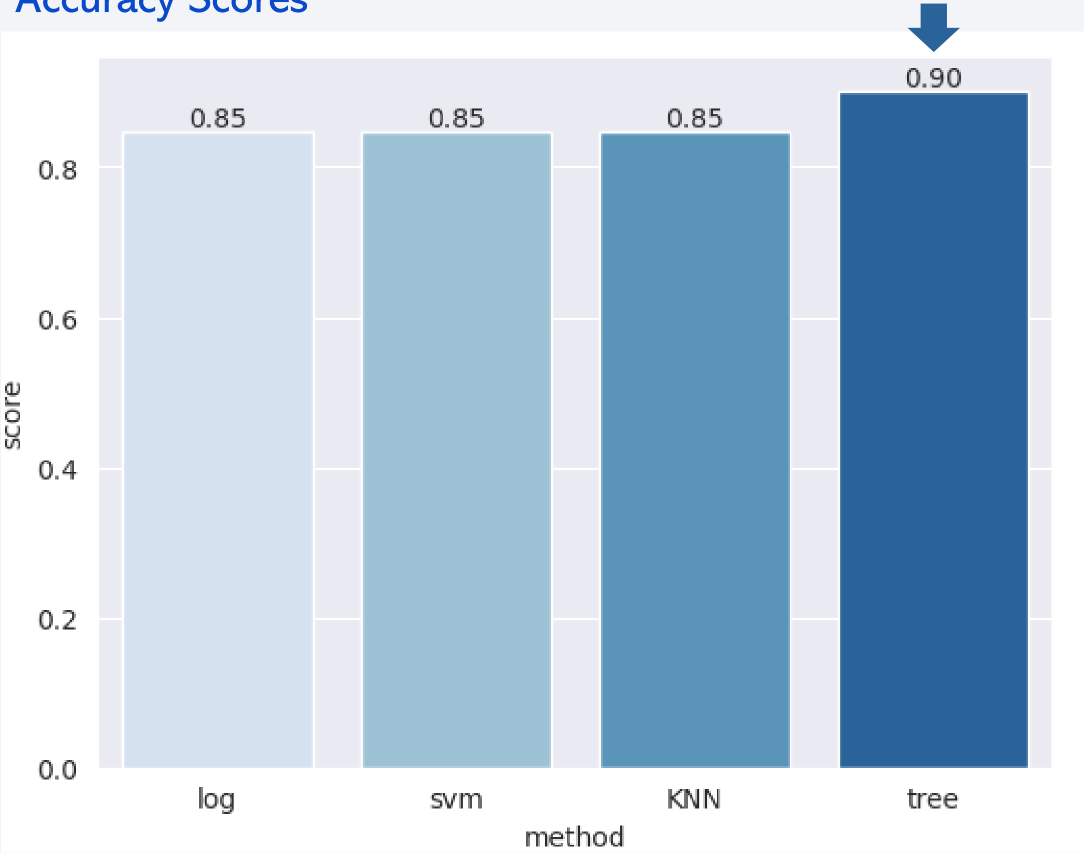
Section 5

Predictive Analysis (Classification)

Classification Accuracy

Accuracy Scores

Best Score



ranking 4 3 2 1

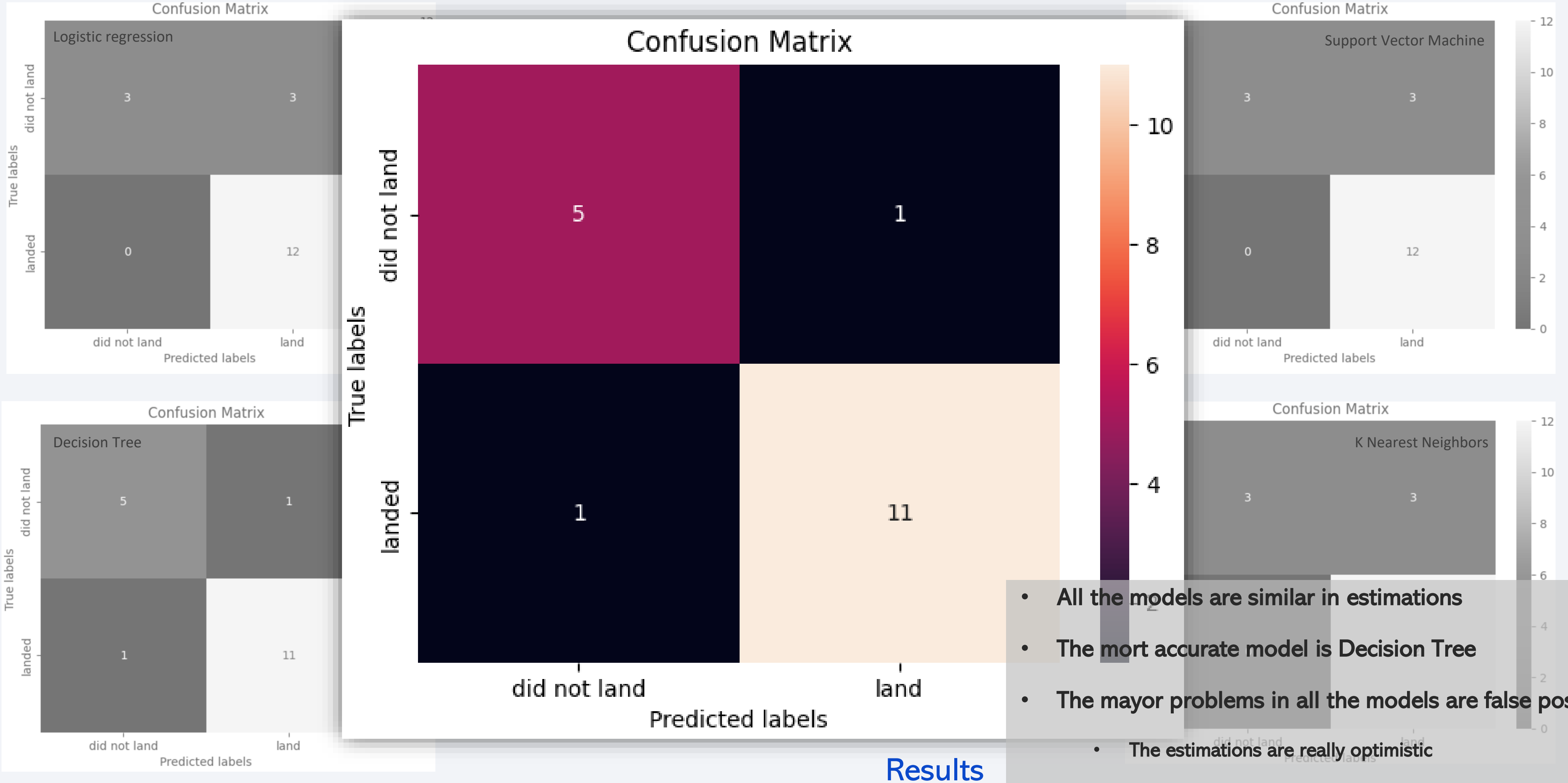
Results

#	method	score	ranking
1	tree	0.901786	Decision Tree
2	KNN	0.848214	K Nearest Neighbors
3	svm	0.848214	Support Vector Machine
4	log	0.846429	Logistic Regression

- Best Accuracy Score is given by Decision Tree Method (0,9 aprox)
- The rest of the algorithms used have almost the same result (0.85 aprox).
- In any case an accuracy score greater than 0.85 is a good indicator of the result of each method.

Daniel E. Christello

Confusion Matrix



- All the models are similar in estimations
- The mort accurate model is Decision Tree
- The mayor problems in all the models are false positives (11/12)
  - The estimations are really optimistic

Results and Conclusions

Results of methodology applied

Recalling issues analysed

- As the flight number increases, the first stage is more likely to land successfully.
- The more massive the payload, the more likely the first stage will return.
- As the flight number increases, the probability of success increases on each site.
- The experience of successes and failures is transferred between sites,
- resulting in a greater number of total successes.
- WAFB SLC 4E is the site with the fewest crashes between releases

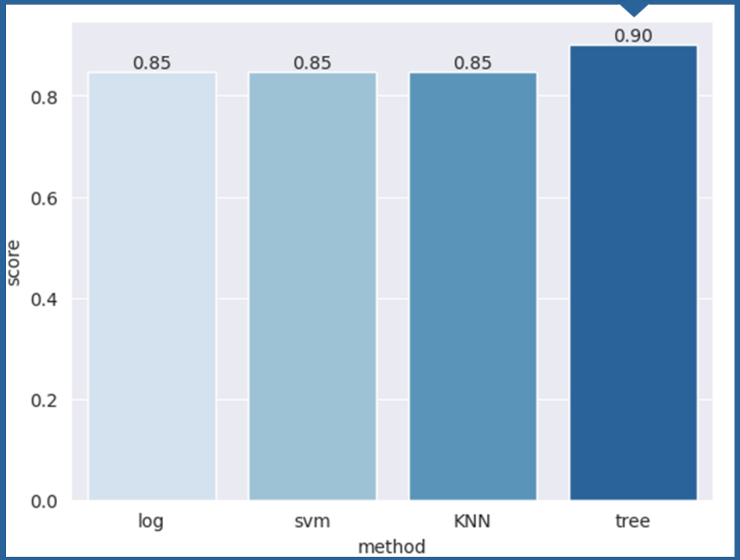
- In VAFB-SLC launchsite there are no rockets launched for heavypayload
    - mass (greater than 10000)
- ES-L1, GEO, HEO, SSO and VLEO have highest success rate.
- GTO and ISS have lowest success rate.
- LEO, MEO, PO have 0,7 success rate.
- Since 2013 the success rate is increasing.

Booster Version Participation Relevance
- 1° FT and v1.1
- 2° v.1.0, B4 and B5

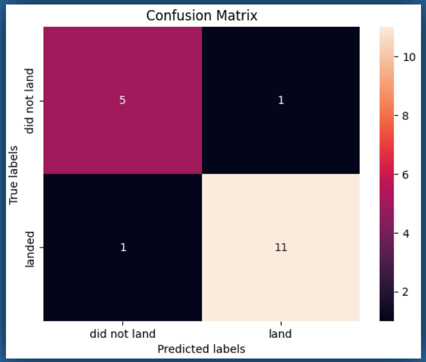
Booster Version and Payload
- B4 and FT is used with more payload
- All but v1.0 used between 2500 and 5000kg
- Between 2500 and 5000kg are more successful landing

Res

❑ Several methods of Machine Lerning can be applied to find good estimations.



4 3 2 1



General Results

Observations

- There are 101 Successful and failure mission outcomes.
- The others are not attempt.

Landing Success Count Ranking
- 1° KSC LC-39A
- 2° CCAFS LC-40
- 3° WAFB SLC-4E
- 4° CCAFS SLC-40

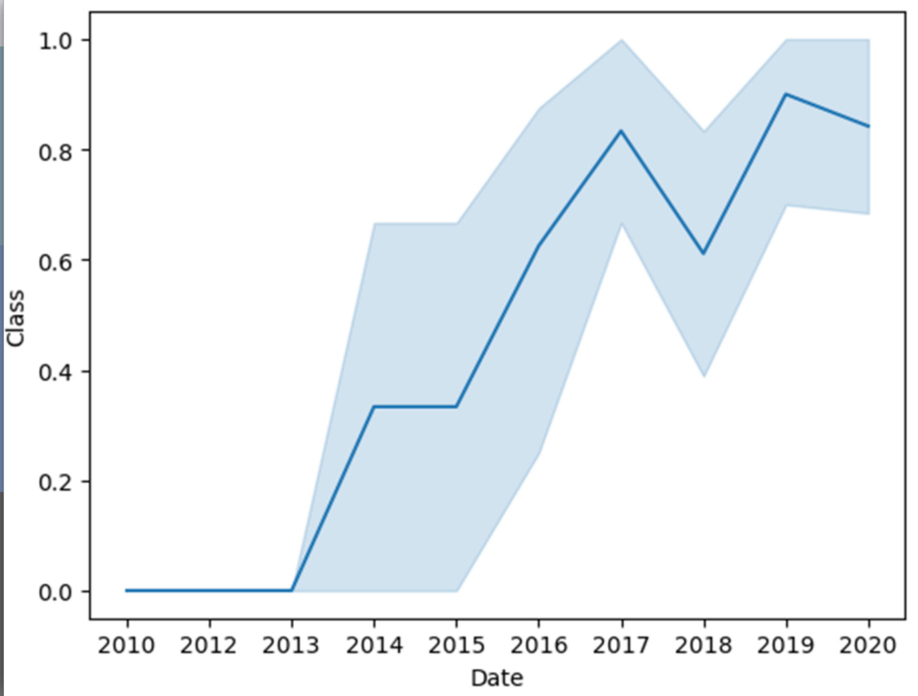
Booster Version Participation Relevance
- *1° FT and v1.1*
- *2° v.1.0, B4 and B5*

Booster Version and Payload
- *B4 and FT is used with more payload*
- *All but v1.0 used between 2500 and 5000kg*

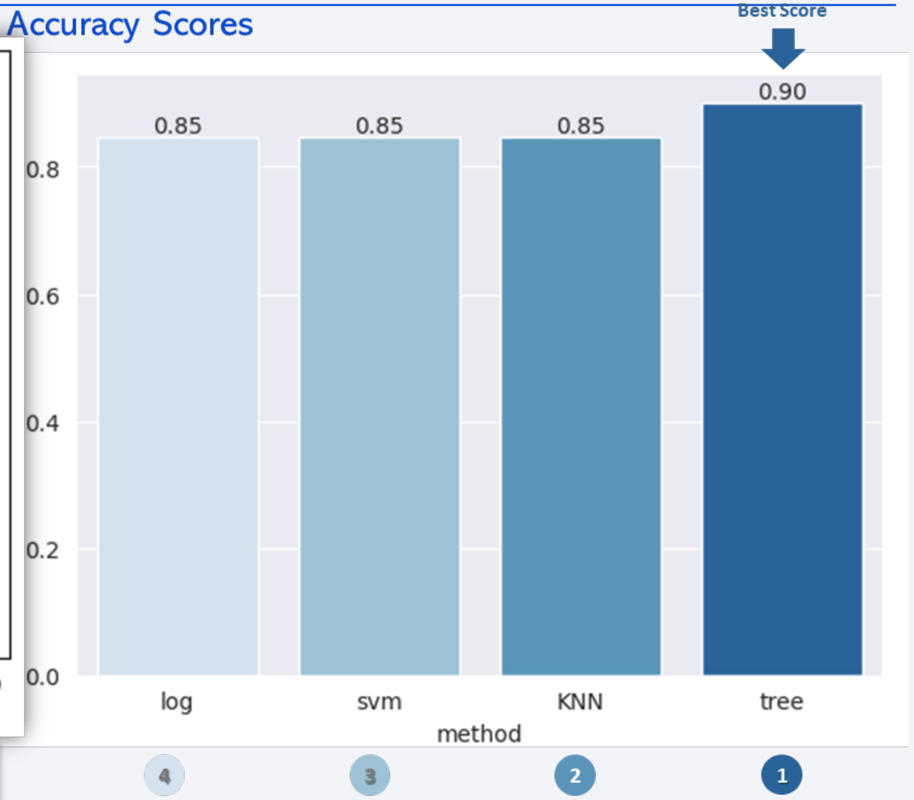
- There are 12 different booster version which have carried the maximum payload mass.

KSC LC-39A is the most successful site

- It has 77% off success



- Since 2013 the success rate is increasing.



- All ML models are similar in estimations
- The mort accurate model is Decision Tree
- The mayor problems in all the models are false positives (11/12)
  - The estimations are really optimistic

- **Cost reduction is possible**
- **ML Models are useful to help cost reduction probabilities based on reusability of stage 1**

Daniel E. Christello

Conclusion

Goal

Estimate the cost of each launch of the Falcon 9 model rocket in the future based mainly on the prediction of the landing success and reusability of the first stage of the rocket:

Conclusions

It's possible to build a Machine Learning model to accomplish the Goal

- Gathering information about Space X, you will use public information.
- Analyzing the data and creating visualization and discussion panels.
- Training a machine learning model to predict whether SpaceX will reuse the first stage.

It's observed that:

- the success rate since 2013 has improved. (via launch number)
- different launch sites have different success rates.(via launch site)
- the landing outcomes we see that CCAFS LC-40, has a success rate of 60%, but if the mass is above 10,000 kg the success rate is 100%.

Therefore, it is possible combine multiple features into different machine learning models:

- Best Accuracy Score is given by Decision Tree Method (0,9 aprox)
- The rest of the algorithms used have almost the same result (0.85 aprox).
- In any case an accuracy score greater than 0.83 is a good indicator of the result of each method.

Appendix

- Uploaded the URL of your GitHub repository including all the completed notebooks and Python files (1 pt)
- Uploaded your completed presentation in PDF format (1 pt)
- Completed the required Executive Summary slide (1 pt)
- Completed the required Introduction slide (1 pt)
- Completed the required data collection and data wrangling methodology related slides (1 pt)
- Completed the required EDA and interactive visual analytics methodology related slides (3 pts)
- Completed the required predictive analysis methodology related slides (1 pt)
- Completed the required EDA with visualization results slides (6 pts)
- Completed the required EDA with SQL results slides (10 pts)
- Completed the required interactive map with Folium results slides (3 pts)
- Completed the required Plotly Dash dashboard results slides (3 pts)
- Completed the required predictive analysis (classification) results slides (6 pts)
- Completed the required Conclusion slide (1 pts)
- Applied your creativity to improve the presentation beyond the template (1 pts)
- Displayed any innovative insights (1 pts)

Appendix

In the map above you can see at a glance that:

- All sites are in locations close to the coast, at a distance of approximately 1 to 2 kilometers.
- All sites are far from the equator.
- The southernmost site is the one closest to the equator and is located at a distance of approximately: 3200km

Haversine distance

To calculate the approximate average distance I created a haversine function.

```
def haversine(lat1, lon1, lat2, lon2):
    rad=math.pi/180
    dlat=lat2-lat1
    dlon=lon2-lon1
    R=6372.795477598
    a=(math.sin(rad*dlat/2))**2 + math.cos(rad*lat1)*math.cos(rad*lat2)*(math.sin(rad*dlon/2))**2
    distancia=2*R*math.asin(math.sqrt(a))
    return distancia
```

Distance to the coast (Estimation with Haversine funtion)

Taking the CCAFS LC-40 site and calculating the distance to the **coast** gives approximately 1Km

```
haversine(28.562302,-80.577356,28.562302,-80.568)
```

```
0.9139855466753368
```

Distance to the Equator (Estimation with Haversine funtion)

Taking the CCAFS LC-40 site and calculating the distance to the **Equator** gives approximately 3200Km

```
haversine(28.562302,-80.577356,0,-80.577356)
```

```
3176.878132425572
```

```
folium.PolyLine(
    locations=[(0, -360), (0, 360)],
    tooltip="Equator",
    color="red",
    line_cap="round",
).add_to(site_map)

folium.PolyLine(
    locations=[(34.632834,-120.610745), (0, -120.610745)],
    tooltip="Distance to Ecquator",
    color="green",
    line_cap="round",
).add_to(site_map)

folium.Map().add_child(
    folium.LatLngPopup().add_to(site_map)
)
site_map
```

Thank you!

Daniel E. Christello