IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Manoel Revez

12th June 2023

Outline

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

Executive Summary

- Summary of methodologies
  - Data collection methodology
  - Data wrangling
  - Exploratory data analysis (EDA) using visualization and SQL
  - Visual analytics using Folium and Plotly Dash
  - Predictive analysis using classification models
- Summary of all results
  - Exploratory data analysis
  - Interactive analytics
  - Predictive analysis

Introduction

- Project background and context
  - SpaceX claims to save millions by reusing the first stage in other launches. Space Y is a company that aims to compete with SpaceX in making launches more economic
  - The purpose of this work is to use public information, train a machine learning model and creating dashboards with Space X launch information in order to predict if SpaceX will reuse the first stage, therefore calculating the price of each launch.

- Problems you want to find answers
  - Which attributes play a meaningful role in predicting the final outcome
  - Can we create a model to accurately predict the outcome of each launch

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - SpaceX REST API (data about launches)
  - Scraping through Wikipedia pages for Falcon 9 Launch data
- Perform data wrangling
  - Identification and processing missing values in each attribute
  - Format the data collected
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Use of Machine Learning Models to predict the outcome of the missions

Data Collection

- SpaceX REST API
  - URL to target a specific endpoint of the API to get past launch data.
  - perform a get request
  - Our response will be in the form of a list of JSON objects.
  - Convert this JSON to a dataframe,

- Scraping through Wiki pages
  - GET method to request the F9 Launch
  - Python BeautifulSoup package to web scrape some HTML
  - Parse the data from those tables
  - Convert them into a Pandas data

Data Collection – SpaceX API

URL used to target past SpaceX

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

Get request to obtain launch data

```
[7]: response = requests.get(spacex_url)
```

Decode the response content as a Json file

Turn it into a Pandas dataframe using .json_normalize()

```
[11]: # Use json_normalize meethod to convert
      data=pd.json_normalize(response.json())
```

Summary of the data obtained

- For further information check the following link: https://github.com/MRevez/Applied-Data-Science-Capstone/blob/main/jupyter-labs-spacex-data-collection-api_MRevez.ipynb

```
[23]: # Show the head of the dataframe
      dados.describe()
```

[23]:

	FlightNumber	PayloadMass	Flights	Block	ReusedCount	Longitude	Latitude
count	94.000000	88.000000	94.000000	90.000000	94.000000	94.000000	94.000000
mean	54.202128	5919.165341	1.755319	3.500000	3.053191	-75.553302	28.581782
std	30.589048	4909.689575	1.197544	1.595288	4.153938	53.391880	4.639981
min	1.000000	20.000000	1.000000	1.000000	0.000000	-120.610829	9.047721
25%	28.250000	2406.250000	1.000000	2.000000	0.000000	-80.603956	28.561857
50%	52.500000	4414.000000	1.000000	4.000000	1.000000	-80.577366	28.561857
75%	81.500000	9543.750000	2.000000	5.000000	4.000000	-80.577366	28.608058
max	106.000000	15600.000000	6.000000	5.000000	13.000000	167.743129	34.632093

Data Collection - Scraping

```
[4]: static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heav
```

```
[15]: # use requests.get() method with the provided static_url

F9=requests.get(static_url)
```

```
[360]: 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.`
            print(flight_number)
            launch_dict['Flight No.'].append(flight_number)
            datatimelist=date_time(row[0])
```

List of Falcon 9 launches → GET method to request the F9 Launch → Create a BeautifulSoup object → Extract tables → Parse all launch tables → Convert to dataframe

```
[14]: # Use BeautifulSoup() to create a BeautifulSoup object from a response text content

soup = BeautifulSoup(F9.text, "html.parser")
```

```
[20]: # Use the find_all function in the BeautifulSoup object, with element type `table`

print('Classes of each table:')
for table in soup.find_all('table'):
    print(table.get('class'))
html_tables = soup.find_all('table')

# Assign the result to a list called `html_tables`
```

```
[ ]: df=pd.DataFrame(launch_dict)
df.head()
```

[363]:

	Flight No.	Launch site	Payload	Payload mass	Orbit	Customer	Launch outcome	Version Booster	Booster landing	Date	Time
0	1	CCAFS	Dragon Spacecraft Qualification Unit	Dragon Spacecraft Qualification Unit	LEO	SpaceX	Success\n	F9 v1.0B0003.1	Failure	4 June 2010	18:45
1	2	CCAFS	Dragon	Dragon	LEO	NASA	Success	F9 v1.0B0004.1	Failure	8 December 2010	15:43
2	3	CCAFS	Dragon	Dragon	LEO	NASA	Success	F9 v1.0B0005.1	No attempt\n	22 May 2012	07:44
3	4	CCAFS	SpaceX CRS-1	SpaceX CRS-1	LEO	NASA	Success\n	F9 v1.0B0006.1	No attempt	8 October 2012	00:35
4	5	CCAFS	SpaceX CRS-2	SpaceX CRS-2	LEO	NASA	Success\n	F9 v1.0B0007.1	No attempt\n	1 March 2013	15:10

- For further information check the following link: https://github.com/MRevez/Applied-Data-Science-Capstone/blob/main/jupyter-labs-webscraping_MRevez.ipynb

Data Wrangling

Launches p/ site

```
[7]: # Apply value_counts() on column LaunchSite

     df['LaunchSite'].value_counts()

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

Occurence p/ orbit

```
[8]: # Apply value_counts on Orbit column
     df['Orbit'].value_counts()

[8]: 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
```

Mission outcome p/ orbit

```
[9]: # landing_outcomes = values on Outcome column
     df['Outcome'].value_counts()

[9]: True ASDS      41
     None None      19
     True RTLS      14
     False ASDS      6
     True Ocean      5
     False Ocean     2
     None ASDS       2
     False RTLS      1
     Name: Outcome, dtype: int64
```

Success/unsuccess falcon 9 landings

```
[14]: # landing_class = 0 if bad_outcome
      landing_class=[]
      for i in df['Outcome']:
          #print(i)
          if i in bad_outcomes:
              landing_class.append(0)
          else:
              landing_class.append(1)
      landing_class
      # landing_class = 1 otherwise
```

Append classification variable

```
[15]: df['Class']=landing_class
      df[['Class']].head(8)
```

[15]:

	Class
0	0
1	0
2	0
3	0
4	0
5	0
6	1
7	1

Calculated success rate

```
[18]: df["Class"].mean()

[18]: 0.6666666666666666
```

- For further information check the following link: https://github.com/MRevez/Applied-Data-Science-Capstone/blob/main/labs-jupyter-spacex-Data%20wrangling_MRevez.ipynb

EDA with Data Visualization

- **The following visualizations were produced:**

- FlightNumber vs. Payload and launch outcome (are heavier loads more prone to fail?)
- Flight Number vs Launch Site and launch outcome (are there any location issues related to success/failure?)
- Launch sites vs. payload mass and launch outcome (is a particular location being used for a particularly difficult Payload?).
- Success rate vs. orbit type (is success related to the type of orbit of the launch?)
- FlightNumber vs. Orbit type (experience?)
- Payload vs. Orbit type (is Payload important for the prediction of success/failure?)
- Launch success yearly trend (how important is experience in the success of the launches)

- For further information check the following link: https://github.com/MRevez/Applied-Data-Science-Capstone/blob/main/jupyter-labs-eda-dataviz_MRevez.ipynb

EDA with SQL

- **The following information was extracted:**

- The names of the unique launch sites in the space mission
- 5 records where launch sites begin with the string 'CCA'
- Total payload mass carried by boosters launched by NASA (CRS)
- Average payload mass carried by booster version F9 v1.1
- Date of the first successful landing outcome in ground pad.
- Names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
- Total number of successful and failure mission outcomes
- Names of the booster versions which have carried the maximum payload mass.
- Records with the month names, failure landing outcomes in drone ship ,booster versions, launch site for the months in year 2015.
- Rank the count of successful landing outcomes between the date 04-06-2010 and 20-03-2017 in descending order.

- For further information check the following link: https://github.com/MRevez/Applied-Data-Science-Capstone/blob/main/jupyter-labs-eda-sql-coursera_sqllite_MRevez.ipynb

Build an Interactive Map with Folium

- All launch sites were marked on a map (folium.Circle and folium.Marker were added for each launch site on the site map with the exact location)
- Success/failed launches were marked for each site on the map (green and red color-labeled markers were added with a folium.Marker for easy visualization for each site)
- The distances between a launch site to its proximities were calculated (MousePosition was used to calculate the distance between the launch site and several relevant positions such as railways an cities. A line was drawn between a launch site and the selected locations)

- For further information check the following link: https://github.com/MRevez/Applied-Data-Science-Capstone/blob/main/lab_jupyter_launch_site_location_MRevez.ipynb

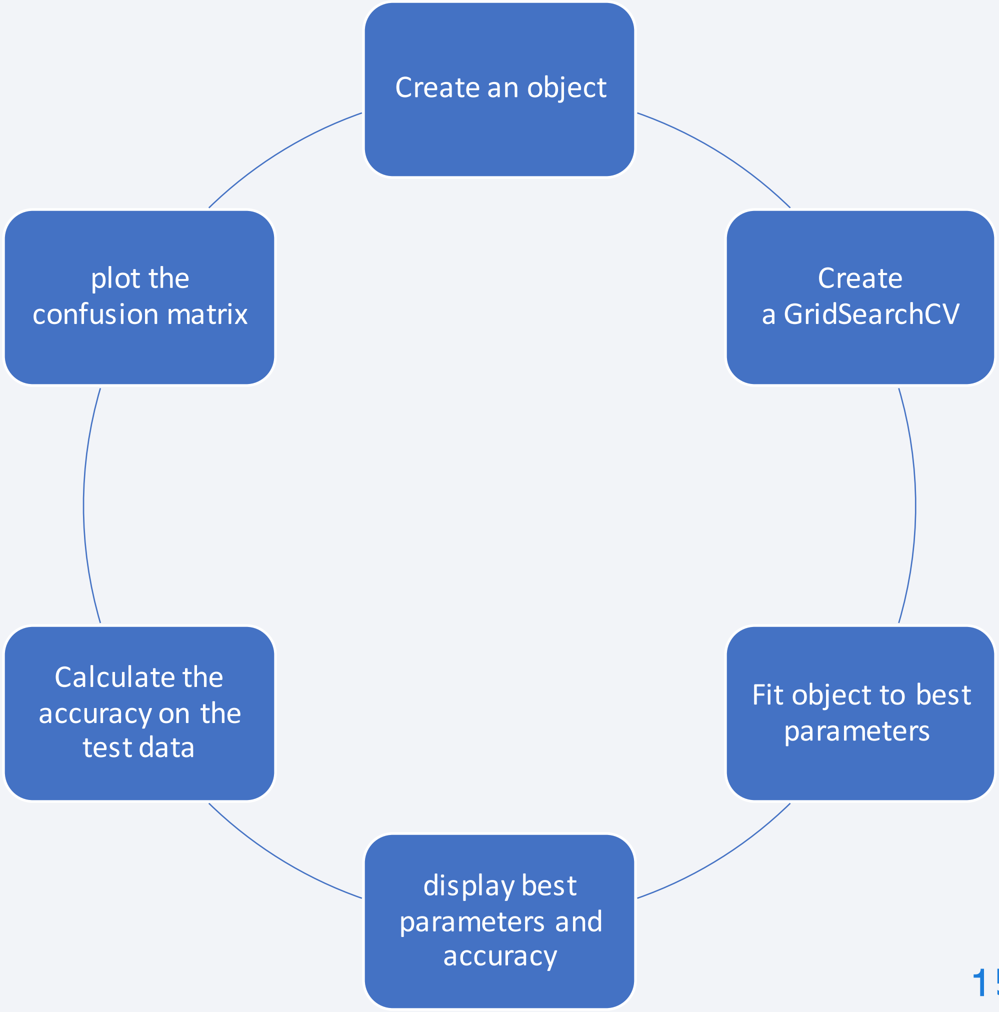
Build a Dashboard with Plotly Dash

- A Plotly Dash application was built, for users to perform interactive visual analytics on SpaceX launch data in real-time.
- This dashboard application contains a pie chart, with a dropdown menu option to allow visualization of the success rate for all sites, as well as the success rate for each site individually.
- A Range Slider was created, allowing the selection of Payload ranges. A function was created to render the success payload scatter plot, with classification of launches, booster version and launch site, for the payload range selected

- For further information check the following link: https://github.com/MRevez/Applied-Data-Science-Capstone/blob/main/dash_interactivity.py

Predictive Analysis (Classification)

- Standardize (Preprocess and Scale) the data in X using the transform preprocessing.StandardScaler()
- With function train_test_split, split the data X and Y into training and test data
- Find best model for logistic regression (LR), support vector machine object (SVM), decision tree classifier (Tree), k nearest neighbors (KNN), using the best fit model loop
- For further information check the following link: https://github.com/MRevez/Applied-Data-Science-Capstone/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5_MRevez.ipynb



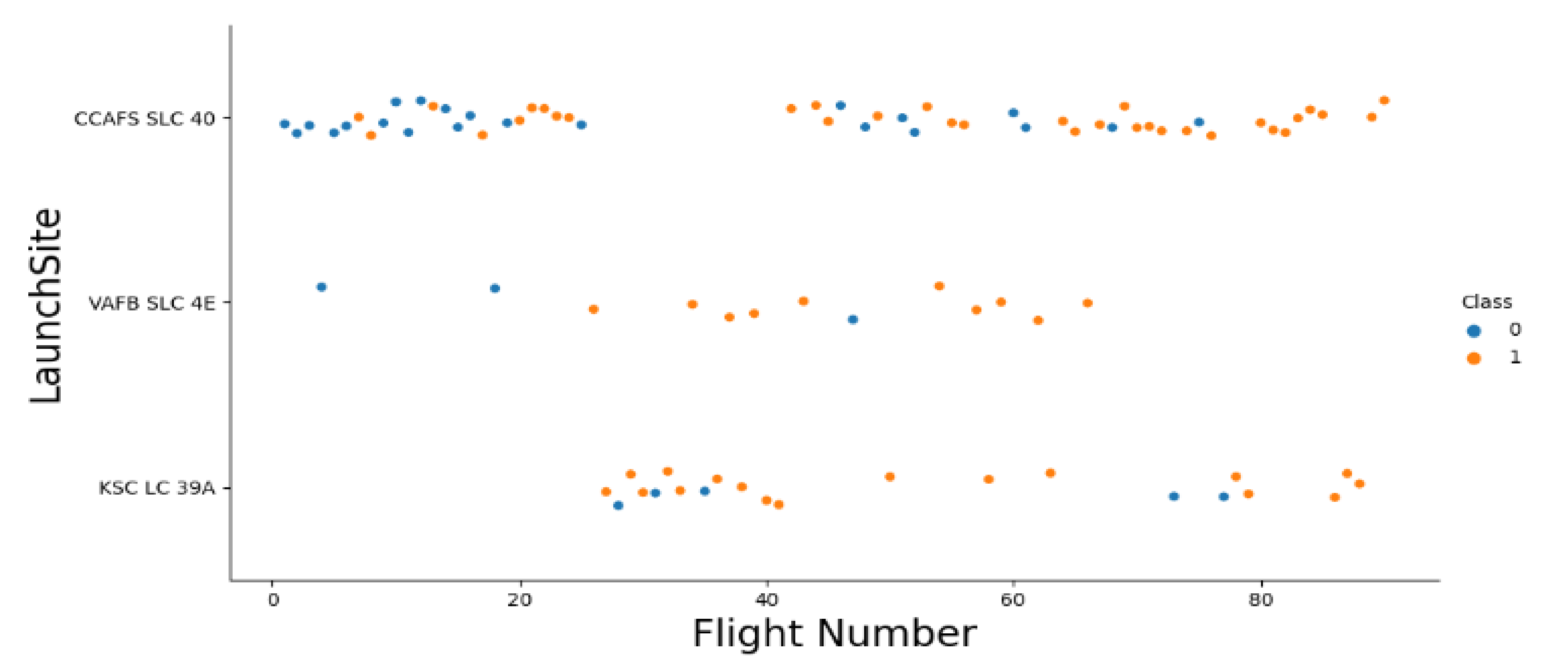
Results

- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

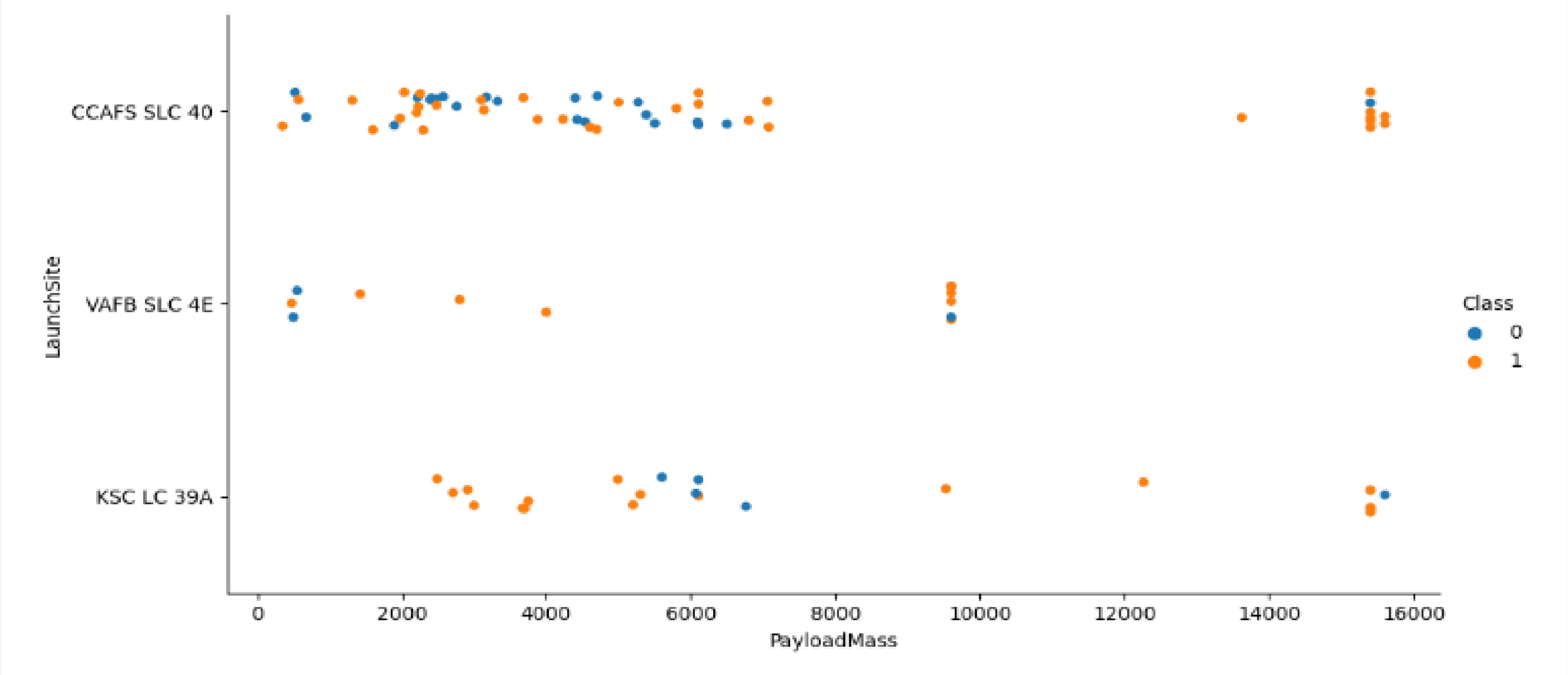
Insights drawn from EDA

Flight Number vs. Launch Site



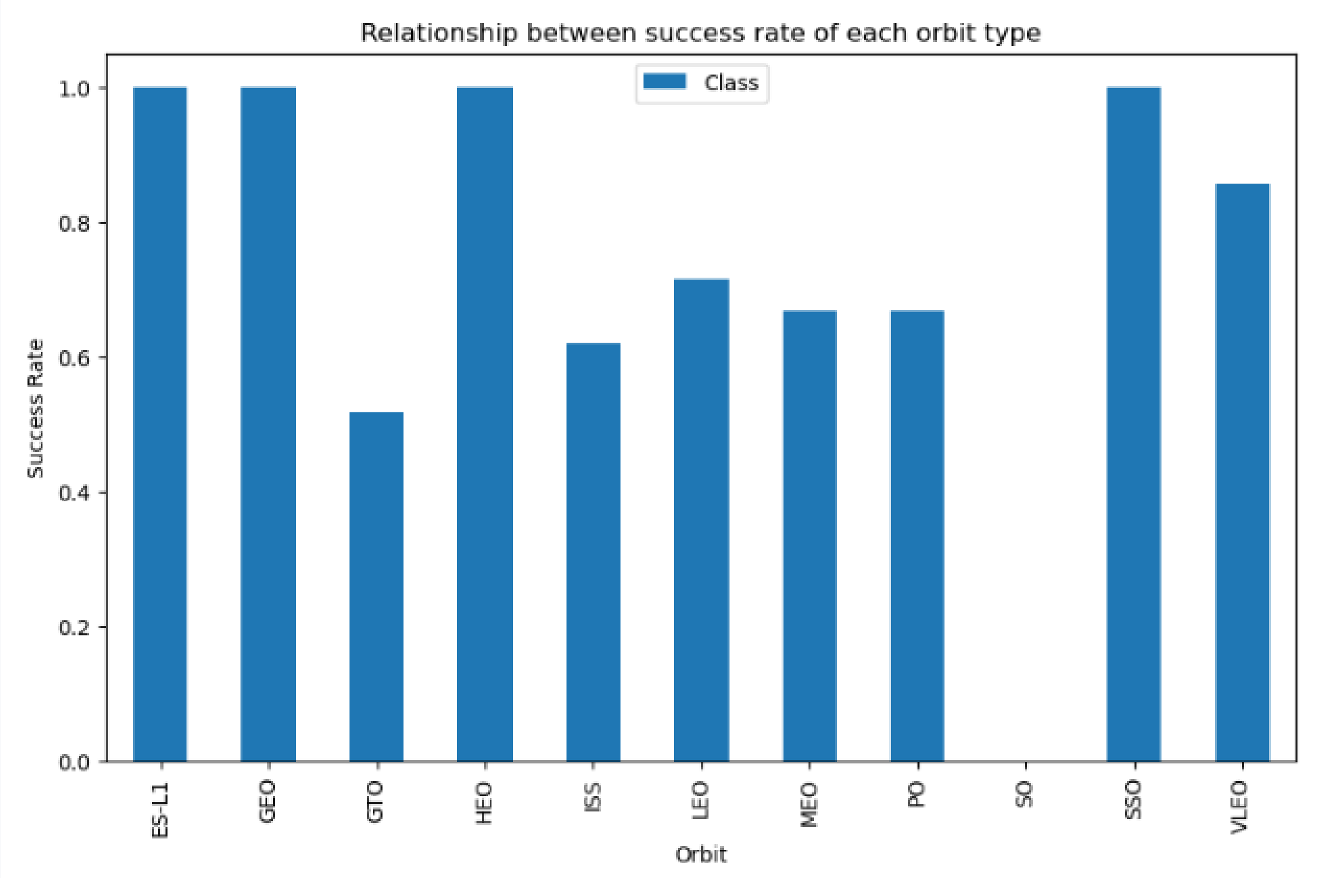
- In the first 25 flights, launches came mostly from CCAFS SLC 40 (East coast, close to the coastline), with low success rates (Experience?)
- Later on, launches were conducted from other sites, with higher success rates. After flight 42 they were conducted from all sites

Payload vs. Launch Site



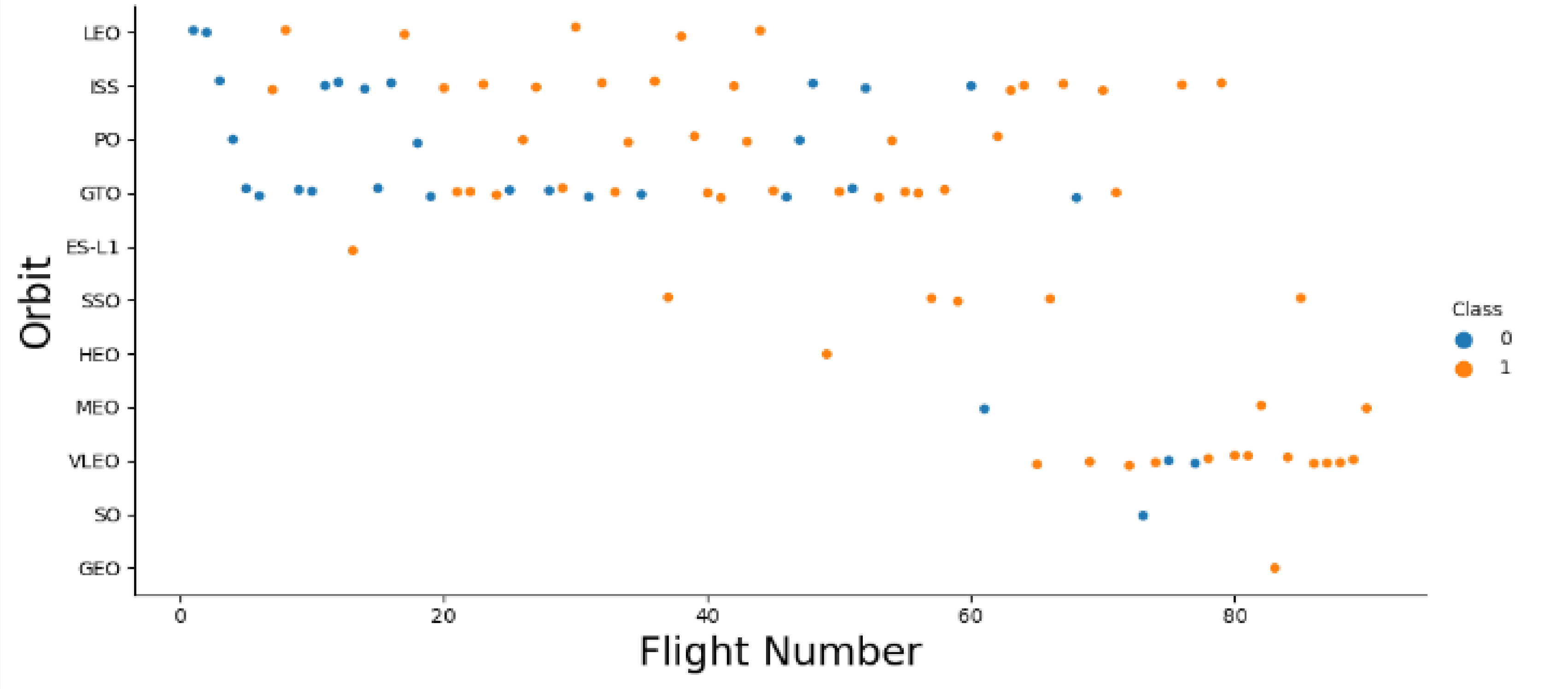
- With Payload Mass greater than 7500kg the mission success rate appears to increases, but there are not many missions to evaluate
- Most launches carry a Payload Mass lower than 7500kg in any location
- For VAFB SLC 4E there are no launches with Payload greater than 10000kg

Success Rate vs. Orbit Type



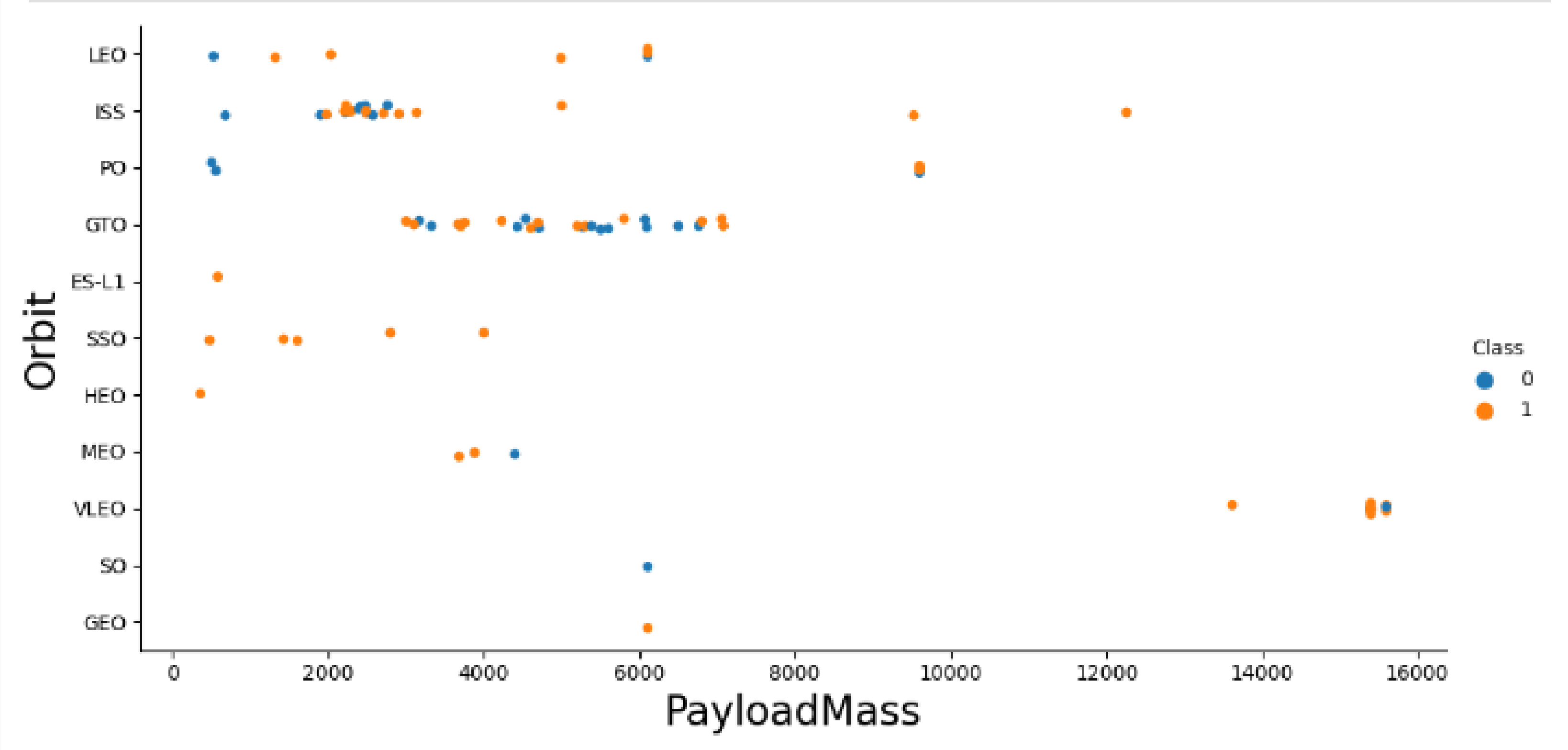
- ES-L1 (lagrange point), GEO, HEO and SSO are the highest success rate launches. They are also the types of orbit with less missions (1,1,1 and 4 respectively)

Flight Number vs. Orbit Type



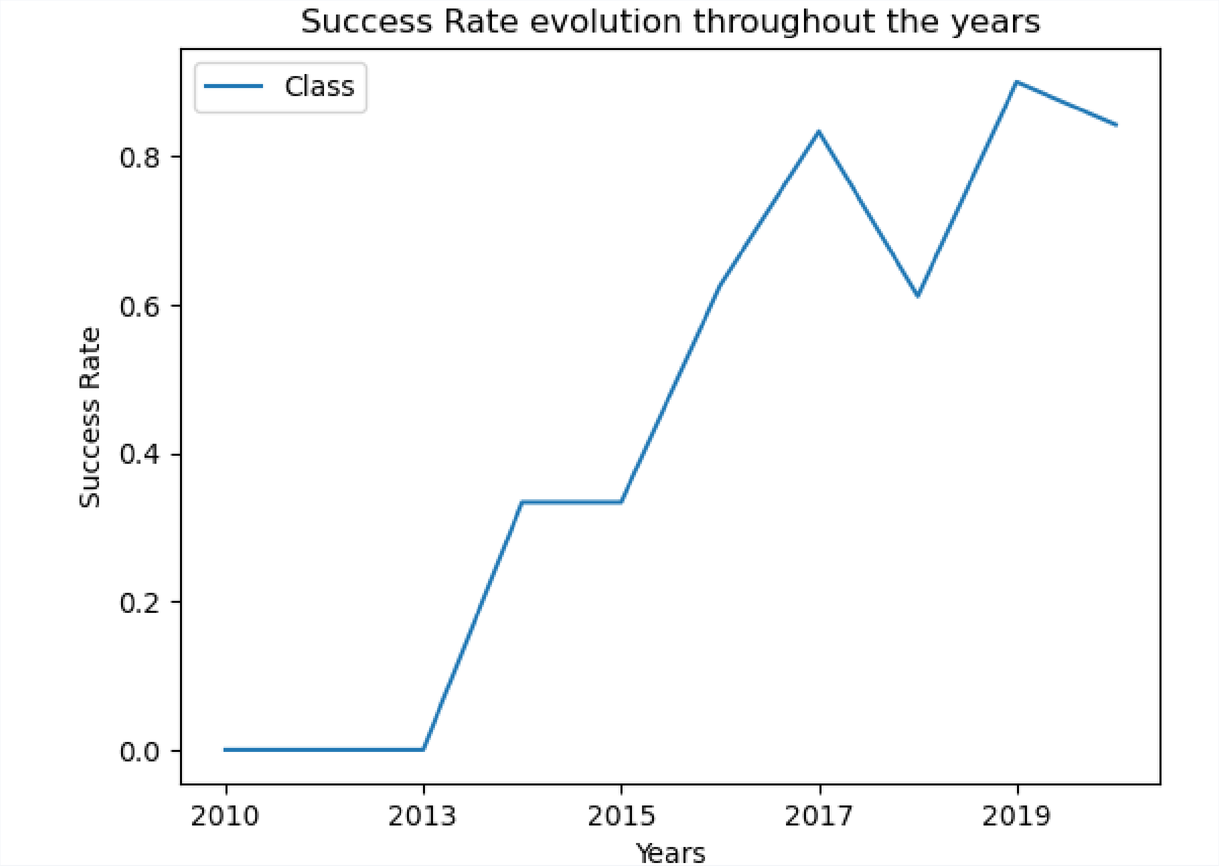
- in the LEO orbit Success appears related to flight number.
- In SSO all flights were successful
- The highest number of flights are done to ISS, GTO, PO, VLEO and LEO orbits
- GTO is the least successful one (also one of the furthest orbit from Earth)

Payload vs. Orbit Type



- There seems to be no close relationship between Payload Mass and mission success in any given orbit.
- ISS and GTO orbits appear to be related to a lower probability of success

Launch Success Yearly Trend



- Between 2010 and 2013 there were no success flights.
- Then the success rate gradually came up to over 80% in the last years

All Launch Site Names

- The DISTINCT statement was used to find all different site names

```
[7]: %sql Select DISTINCT Launch_Site from SPACEXTBL

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

[7]:

Launch_Site
CCAFS LC-40
VAFB SLC-4E
KSC LC-39A
CCAFS SLC-40
None

Launch Site Names Begin with 'CCA'

- The WHERE clause was used to select all launches from location ´CCA´
- The LIMIT clause was used to limit results to 5

```
[8]: %sql Select * from SPACEXTBL where Launch_Site LIKE '%CCA%'Limit 5;

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

[8]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
06/04/2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0.0	LEO	SpaceX	Success	Failure (parachute)
12/08/2010	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0.0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
22/05/2012	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525.0	LEO (ISS)	NASA (COTS)	Success	No attempt
10/08/2012	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500.0	LEO (ISS)	NASA (CRS)	Success	No attempt
03/01/2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677.0	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

- The WHERE clause was used to select all launches from Customer 'NASA'
- The sum function was used to find the total payload carried by boosters from NASA launches (45596 kg)

```
In [22]: %sql Select sum(PAYLOAD_MASS__KG_) from SPACEXTBL where Customer='NASA (CRS)';

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

Out[22]:

sum(PAYLOAD_MASS__KG_)
45596.0

Average Payload Mass by F9 v1.1

- The WHERE clause was used to select all launches from Booster 'F9 v1.1'
- The avg function was used to find the average payload carried by booster version F9 v1.1 (2928.4 kg)

```
[12]: %sql Select avg(PAYLOAD_MASS__KG_) from SPACEXTBL where Booster_Version = 'F9 v1.1';

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

[12]:

avg(PAYLOAD_MASS__KG_)
2928.4

First Successful Ground Landing Date

- The WHERE clause was used to find the launch with ground pad successful output and minimum date
- The min function was used to find minimum date (2015)

```
[54]: %sql Select Date, Landing_Outcome from SPACEXTBL where
(substr(Date, 1, 2)+substr(Date, 4, 2)*100+substr(Date, 7, 4)*10000)=
(select min(substr(Date, 1, 2)+substr(Date, 4, 2)*100+substr(Date, 7, 4)*10000) from SPACEXTBL
 where "Landing_Outcome"='Success (ground pad)')

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

[54]:

Date	Landing_Outcome
22/12/2015	Success (ground pad)

Successful Drone Ship Landing with Payload between 4000 and 6000

- The DISTINCT statement was used to find all different booster version involved in the query
- The WHERE clause was used to find both successful landing outcomes AND Payload mass BETWEEN 4000 and 6000

```
[16]: %sql Select distinct(Booster_Version), PAYLOAD_MASS__KG_ from SPACEXTBL where
("Landing_Outcome" LIKE '%Success (drone ship)%' AND (PAYLOAD_MASS__KG_ between 4000 AND 6000));

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

[16]:

Booster_Version	PAYLOAD_MASS__KG_
F9 FT B1022	4696.0
F9 FT B1026	4600.0
F9 FT B1021.2	5300.0
F9 FT B1031.2	5200.0

Total Number of Successful and Failure Mission Outcomes

- The DISTINCT statement was used to find all different mission outcomes involved in the query
- The Group by statement was used to group all mission outcomes
- Finally the Count function was used to count all missions in the selected Groups (only one unsuccessful in 101 launches)

```
[17]: %sql Select DISTINCt(Mission_Outcome), count(Mission_Outcome) from SPACEXTBL Group by Mission_Outcome

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

[17]:

Mission_Outcome	count(Mission_Outcome)
None	0
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

Boosters Carried Maximum Payload

- A subquery was used to select the value of the maximum payload mass
- The WHERE clause was used to find all records with maximum payload
- The DISTINCT statement was used to find all different booster with max value

```
[18]: %sql Select distinct(Booster_Version), PAYLOAD_MASS__KG_ from SPACEXTBL where
PAYLOAD_MASS__KG_=(Select max(PAYLOAD_MASS__KG_) from SPACEXTBL)

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

[18]:

Booster_Version	PAYLOAD_MASS__KG_
F9 B5 B1048.4	15600.0
F9 B5 B1049.4	15600.0
F9 B5 B1051.3	15600.0
F9 B5 B1056.4	15600.0
F9 B5 B1048.5	15600.0
F9 B5 B1051.4	15600.0
F9 B5 B1049.5	15600.0
F9 B5 B1060.2	15600.0
F9 B5 B1058.3	15600.0
F9 B5 B1051.6	15600.0
F9 B5 B1060.3	15600.0
F9 B5 B1049.7	15600.0

2015 Launch Records

```
•[22]: %sql Select date, substr(Date, 4, 2) as  Month, "Landing_Outcome", Booster_Version, Launch_Site from SPACEXTBL where
("Landing_Outcome" LIKE '%Failure (drone ship)%' AND substr(Date,7,4)='2015');

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

[22]:

Date	Month	Landing_Outcome	Booster_Version	Launch_Site
01/10/2015	10	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
14/04/2015	04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

- The substr function was used to select the year (last 4 numbers of the date field)
- The WHERE clause was used to find both failed landing outcomes in drone ship AND launches occurred in 2015

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

- The DISTINCT statement was used to find all different Landing Outcomes
- The Group by statement was used to group all record in Landing outcomes
- The WHERE clause was used to find the records between the date 2010-06-04 and 2017-03-20
- The count function was used to count all elements in Landing Outcome
- The desc command was used to Rank the results in descending order.

```
[23]: %sql Select distinct("Landing_Outcome"), count("Landing_Outcome") as counts from SPACEXTBL where
((substr(Date, 1, 2)+substr(Date, 4, 2)*100+substr(Date, 7, 4)*10000< 20170319) and
 (substr(Date, 1, 2)+substr(Date, 4, 2)*100+substr(Date, 7, 4)*10000 > 20100603))
Group by "Landing_Outcome" ORDER BY counts desc;

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

[23]:

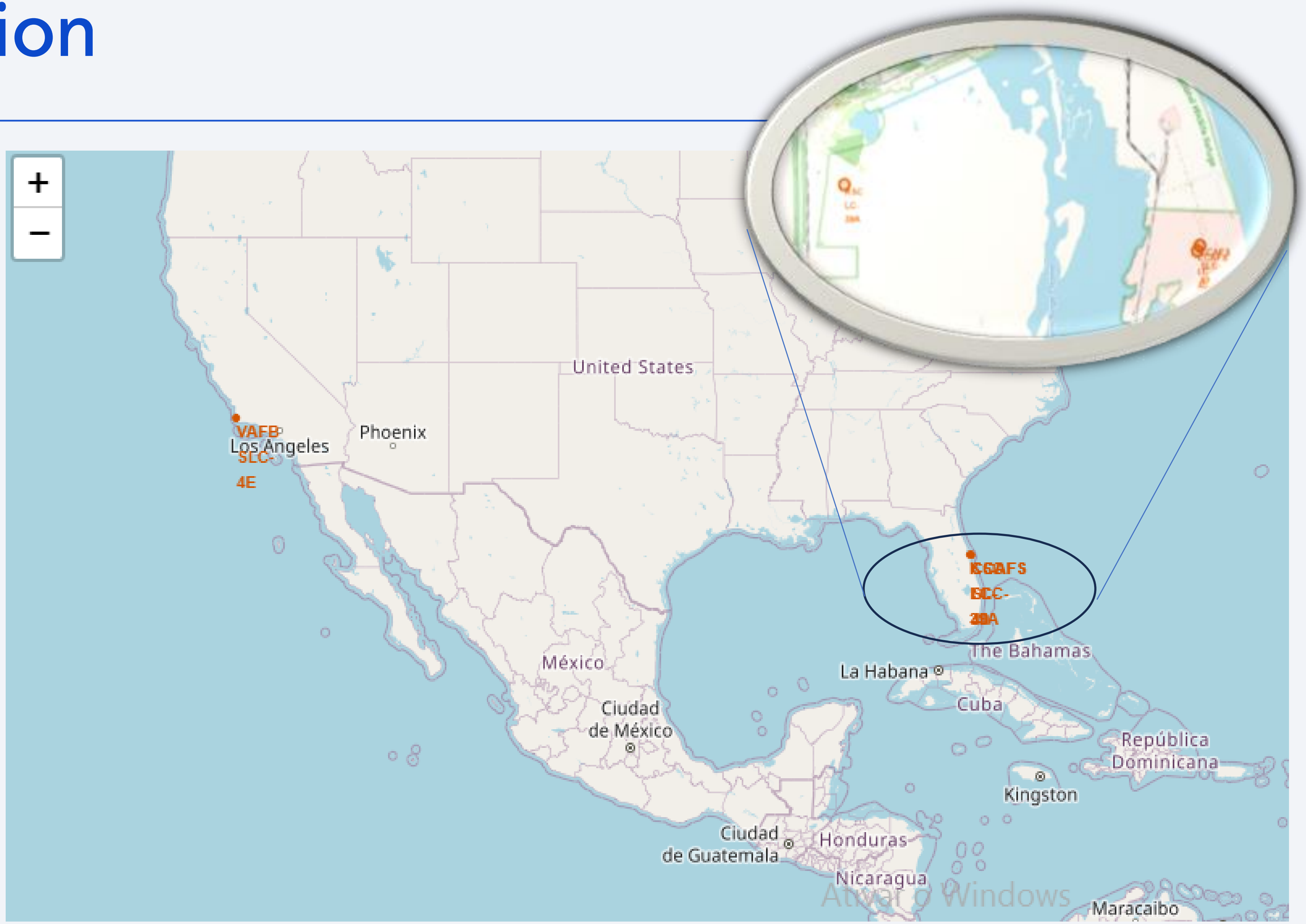
Landing_Outcome	counts
No attempt	10
Success (ground pad)	5
Success (drone ship)	5
Failure (drone ship)	5
Controlled (ocean)	3
Uncontrolled (ocean)	2
Precluded (drone ship)	1
Failure (parachute)	1

Section 3

Launch Sites Proximities Analysis

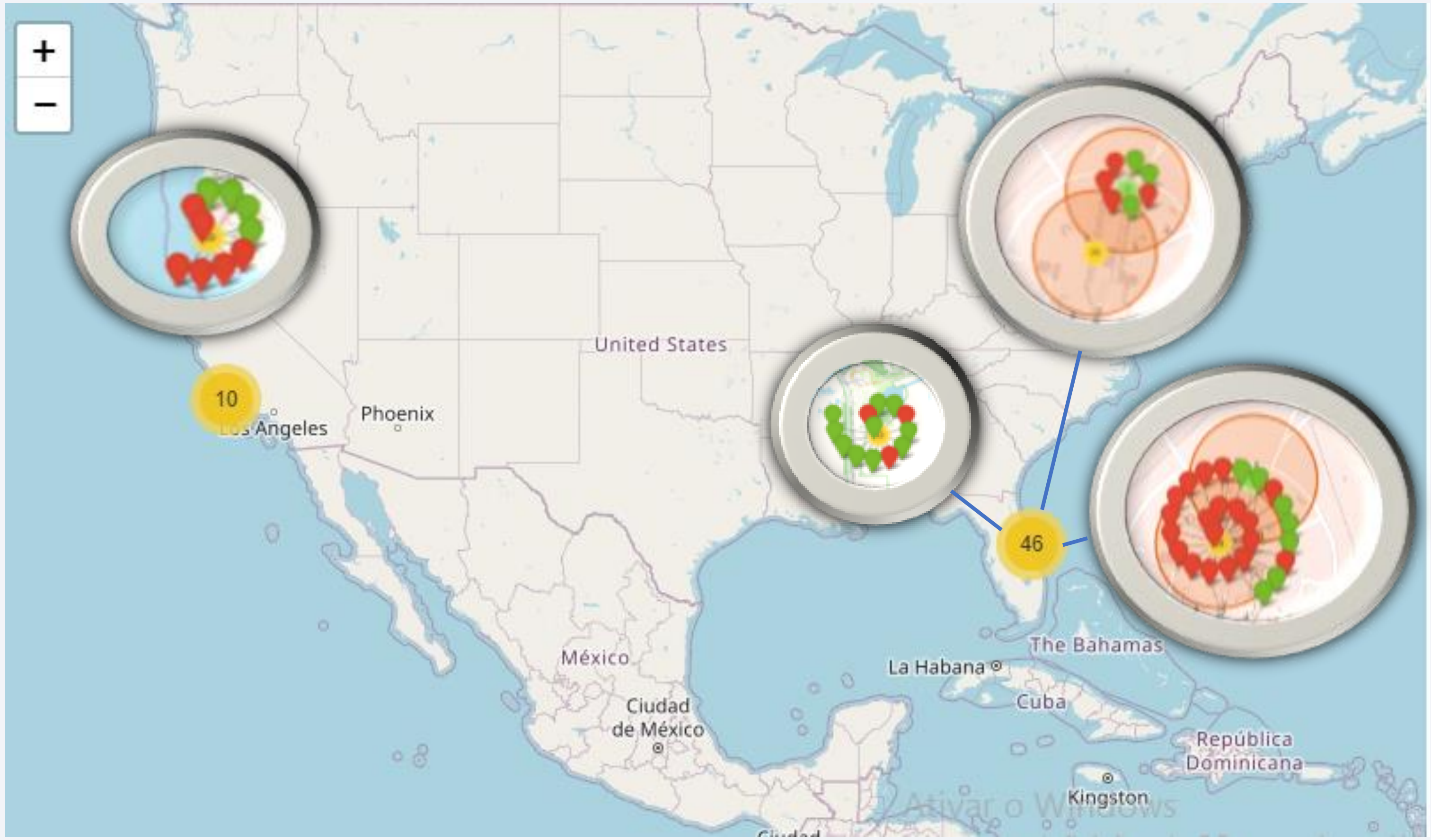
Launch site location

- The East Coast is where most of the launches take place



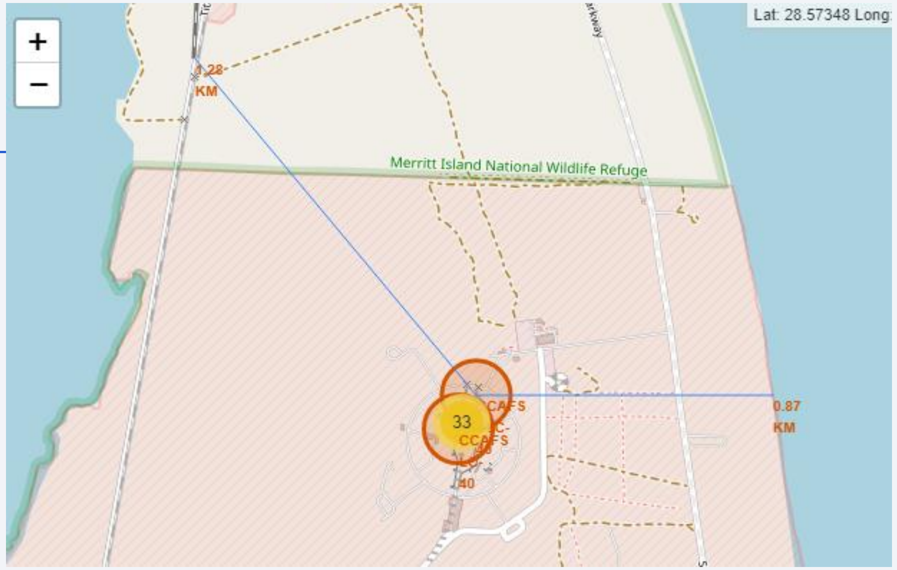
Success/failed launches for each site

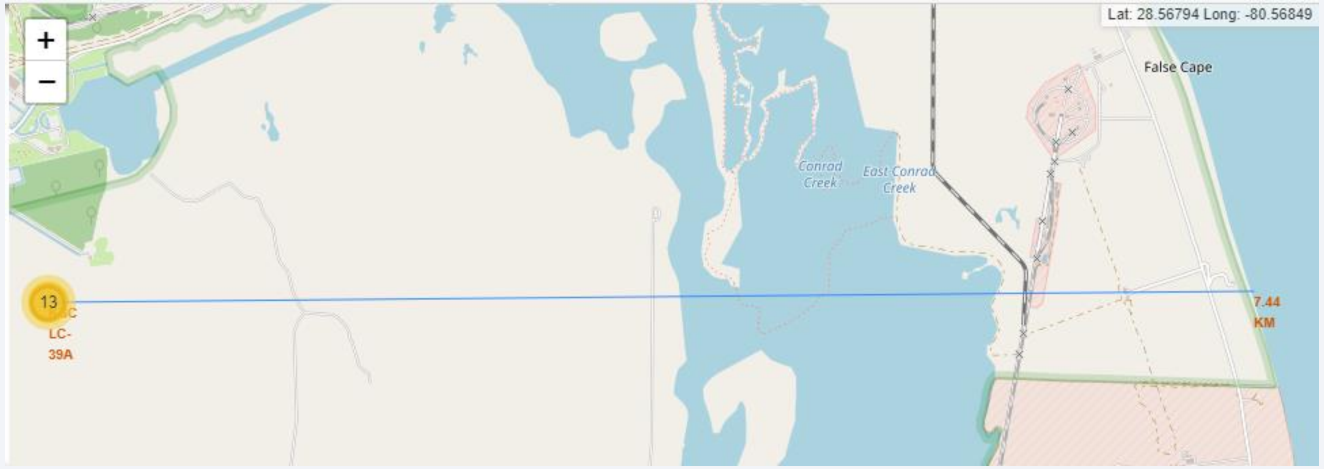
- The launch site that has more successful launches is is the furthest location from the coastline (KSC LC-39A
- The site that has most launches is CCAF SLC-40, but not with a very good success rate

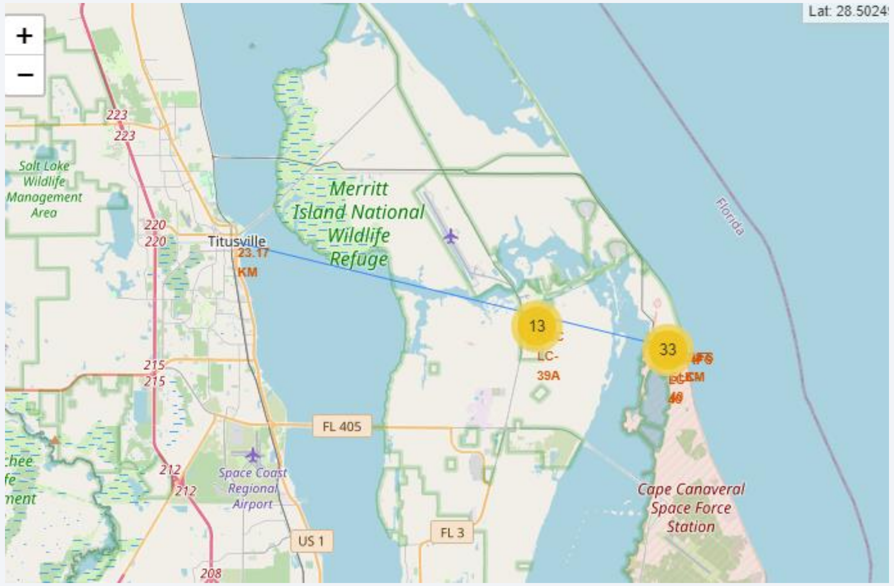


Distance to relevant locations

- KSC LC-39A is 7,44 km away from the coastline. All other locations are closer
- All locations are somewhat far away from cities (safety issues)
- All locations are close to railways, important for transporting materials



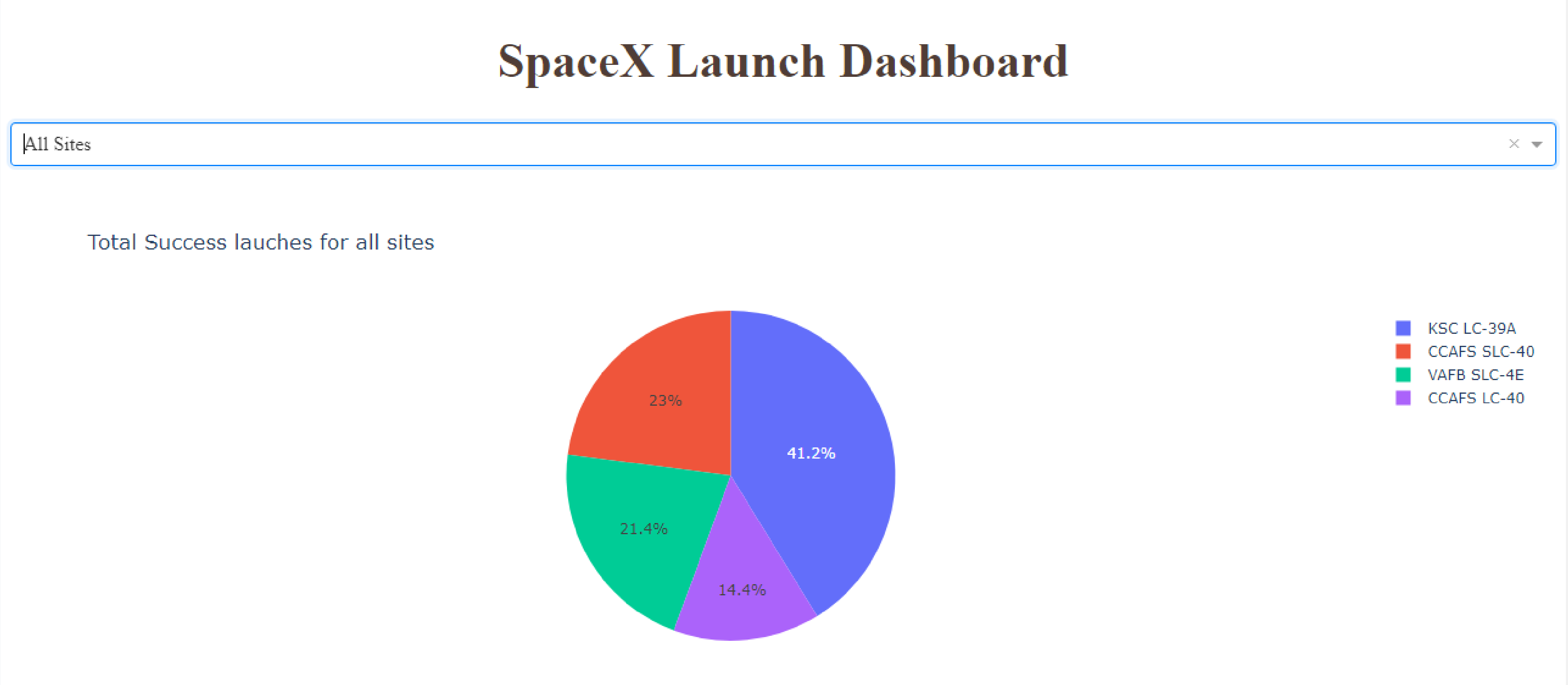




Section 4

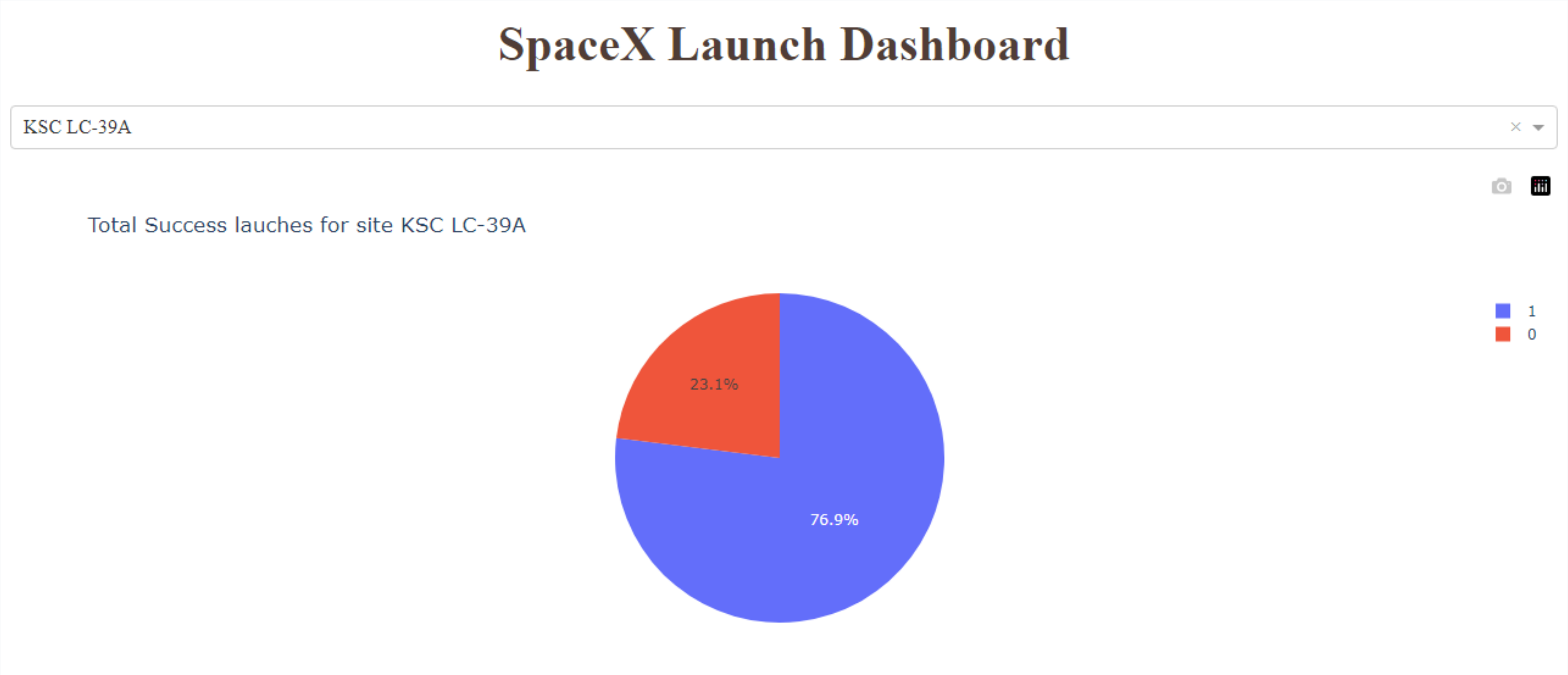
Build a Dashboard with Plotly Dash

SpaceX Success Launch Dashboard



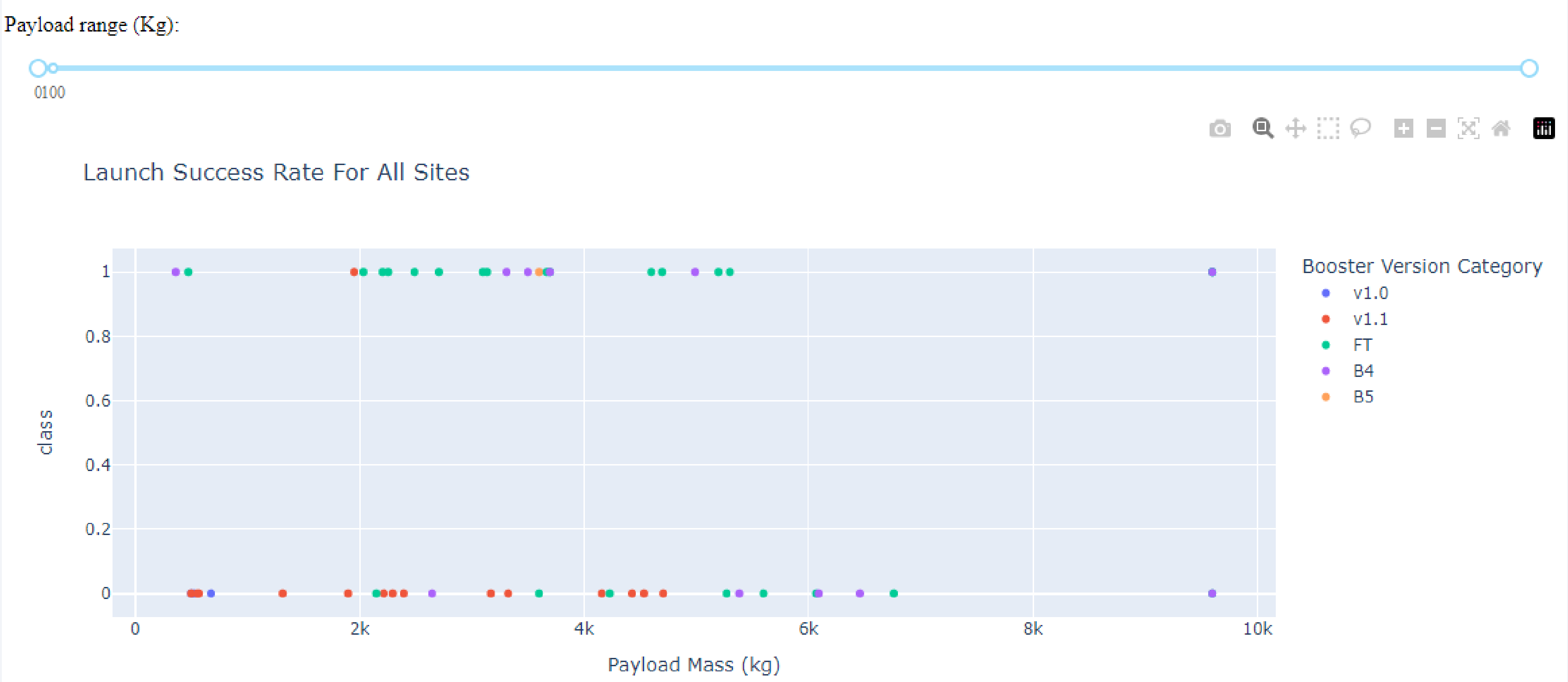
- Piechart of launch success count shows KSC LC-39A as the most successful site, with 41,2% of successful launches coming from this site
- CCAFS LC40 presents itself as the least successful one with 14,4%

Launch site with highest launch success ratio



- KSC LC-39A has the highest success rate with 76.9%

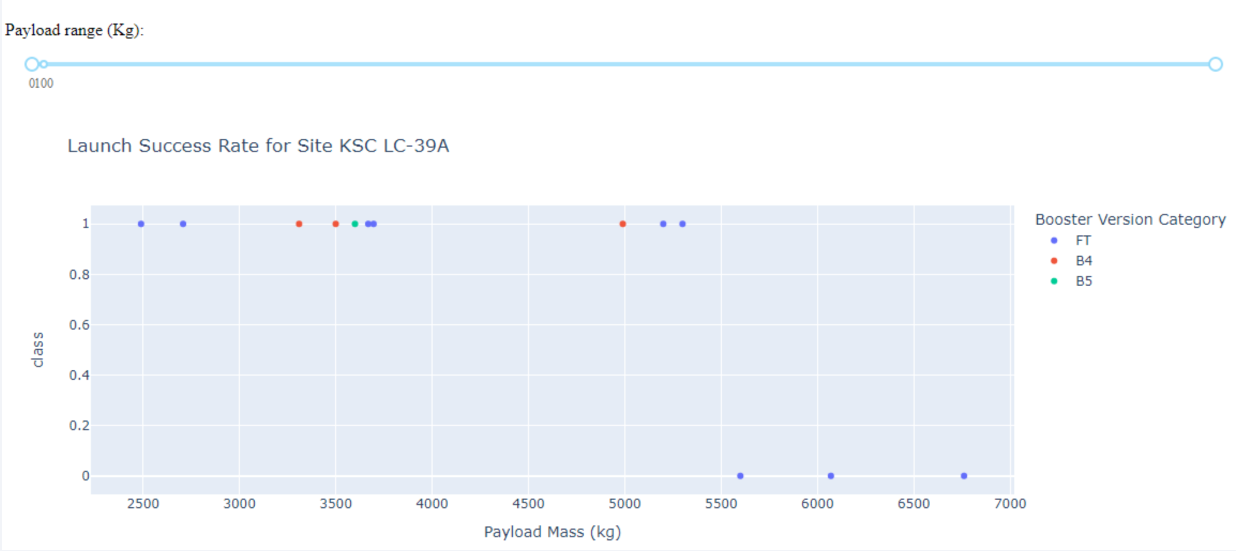
Payload vs. Launch Outcome

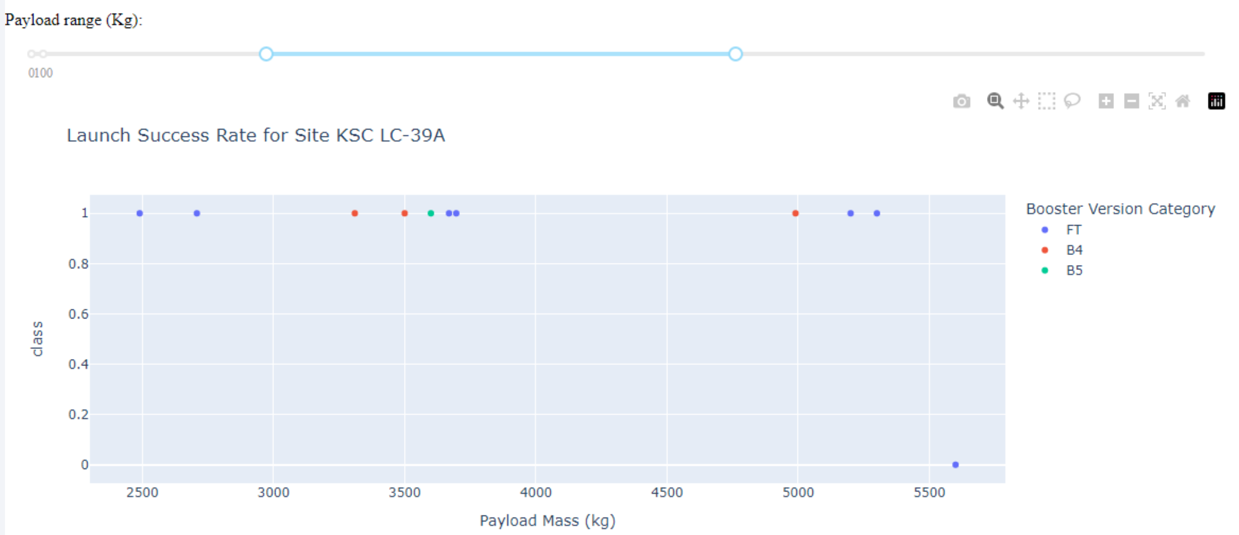


- In all sites, there is only one successful launch over 6761 kg
- Between 5300-9600 kg there are no successful launches
- FT and B4 are the most successful Booster versions

Payload vs. Launch Outcome

- For the most successful site, with payloads higher than 5500kg there are no successful launches in this site
- Between 2500-5300 kg all launches in this site were successful for all Booster versions





Section 5

Predictive Analysis (Classification)

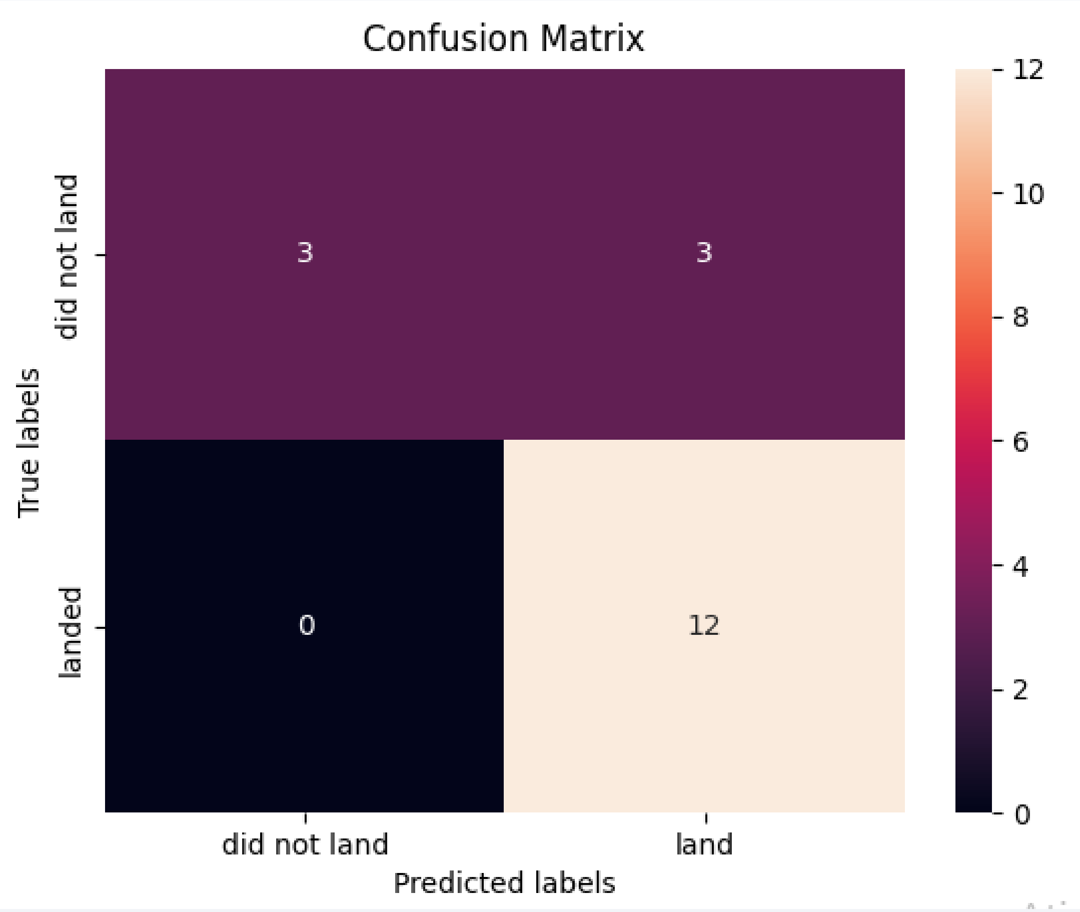
Classification Accuracy

- All values are very similar but Tree and SVM are the ones with higher scores

	Evaluation	KNN	Tree	LR	SVM
0	train	0.861111	0.888889	0.875000	0.888889
1	test	0.833333	0.833333	0.833333	0.833333

Confusion Matrix

- Once again, the performance of the models is very similar, so the confusion matrix looks the same for all of them
- False positives are the biggest problem in the model



Conclusions

- Experience is an important issue concerning the success of the missions. The success rate gradually came up from 0 in 2010 to over 80% in the last years (2019 and 2020).
- KSC LC-39A is the most successful site. It is the furthest location from the coastline.
- The site that has most launches is CCAF SLC-40, but not with a very good success rate, also because it is where the first launces were made.
- The highest number of flights are done to ISS, GTO, PO, VLEO and LEO orbits.
- ISS and GTO orbits appear to be related to a lower probability of success. GTO is the least successful one (also one of the furthest orbit from Earth).
- FT and B4 are the most successful Booster versions.
- All classification models have very similar accuracy results in predicting the landing outcome. Tree and SVM are the ones with higher scores.

Thank you!