IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Naif Ganadily
May 19, 2022

Outline

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

Executive Summary

Summary of methodologies

- Data collection
- Data wrangling
- EDA with data visualization
- EDA with SQL
- Building an interactive map with Folium
- Building a dashboard with Plotly Dash
- Predictive analysis (Classification)

Summary of all results

- EDA results
- Interactive analytics
- Predictive analysis

Introduction

Project background and context

- Commercial Space Age is here! There are a few companies we can look at, but the most successful one is as of now doughtily SpaceX. One example of their many successes is the price of their Falcon9 rocket approx. 62 mil $ Due to the re-use of the first stage in landing the rocket while other providers cost upward of 165 million dollars each for this project, I will be the head data scientist of a new company SpaceY headed by the billionaire Allon Mask.

Problems you want to find answers

- What influences whether a rocket will land successfully or not?
- The impact of certain variables on the success rate of a successful landing

Section 1

Methodology

Methodology

- Executive Summary
- Data collection methodology:
  - SpaceX Rest API
  - Web Scraping from Wikipedia
- Perform data wrangling
  - One Hot Encoding data field for Machine Learning and data cleaning of null values and irrelevant columns
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - LR, KNN, SVM, and DT models have been built and evaluated for the best classifier

Data Collection

- Data was collected using various methods
- Data collection was done using get request to the SpaceX API
- The response came in for a JSON file and was normalized into a pandas data frame using .json_normalize()
- Collected data was cleaned, checked for missing values, and if necessary missing values were filled
- Another data collection was done performing web scraping from Wikipedia for Falcon 9 launch records with BeautifulSoup
- Launch records were extracted from an HTML table and converted into a pandas data frame for future analysis

Data Collection – SpaceX API

Data collection with SpaceX API

Naif Ganadily's Github Link to Data Collection SpaceX API

In [9]:
```
static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_call_spacex_api.json'
```

We should see that the request was successfull with the 200 status response code

In [10]:
```
response.status_code
```

Out[10]:
```
200
```

Data Collection - Scraping

Web scraping from Wikipedia

[Naif Ganadily's Github Link to Data Collection Web Scraping]

```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"
```

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

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

```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(response.text, 'html')
```

Data Wrangling

- Check for null values
- Calculate the number of launches on each site
- Calculate the number and occurrence of each orbit
- Calculate the number and occurrence of mission outcomes per orbit type
- Create a landing outcome label
- Handle null values
- You need to present your data wrangling process using key phrases and flowcharts
- [Naif Ganadily's Github Link to Data Wrangling]

EDA with Data Visualization

- We explored the data by visualizing the relationship between flight number and launch Site, payload and launch site, success rate of each orbit type, flight number and orbit type, the launch success yearly trend.
- Naif Ganadily's Github Link to EDA with Data Visualization

EDA with SQL

- SQL queries performed in Jupiter notebook:
- Displaying the names of the unique launch sites in the space mission
- Displaying 5 records where launch sites begin with the string ‘CCA’
- Display the total payload mass carried by boosters launched by NASA (CRS)
- Display average payload mass carried by booster version F9 v1.1
- List the date when the first successful landing outcome in the ground pad was achieved.
- List the names of the boosters which have success in drone ships and have a payload
- mass greater than 4000 but less than 6000
- List the total number of successful and failed mission outcomes
- List the names of the booster_versions which have carried the maximum payload
- mass. Use a subquery
- List the failed landing_outcomes in drone ships, their booster versions, and launch site
- Names for in the year 2015
- Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground

Build an Interactive Map with Folium lab

We marked all launch sites and added map objects such as markers, circles, and lines to mark the success or failure of launches for each site on the folium map.

We calculated the distances between a launch site to its proximities. We answered some questions for instance:

Are the launch sites near railways, highways, and coastlines?

Do launch sites keep a certain distance away from cities?

Naif Ganadily's Github Link to Interactive Map with Floium lab

Build a Dashboard with Plotly Dash

We built an interactive dashboard with Plotly dash

We plotted pie charts showing the total launches by a certain sites

We plotted a scatter graph showing the relationship between Outcome and Payload Mass (Kg) for the different booster versions.

Naif Ganadily's Github Link to Dashboard with Plotly Dash

[Naif Ganadily's Github Link to Predictive Analysis]

Predictive Analysis (Classification)

- Loaded the data using NumPy and pandas, transformed the data, and split our data into training and testing.
- Built different machine learning models and tune different hyperparameters using GridSearchCV.
- Used accuracy as the metric for our model, improved the model using feature engineering and algorithm tuning.
- Fond the best performing classification model.

Results

Low weighted payloads perform better than the heavier payloads

Success Rate for Space X increased over the years

The orbits GEO, HEO, SSO, ES L 1have the best success rates

As you can see in the screenshot below all models have a similar accuracy and are suited for this dataset

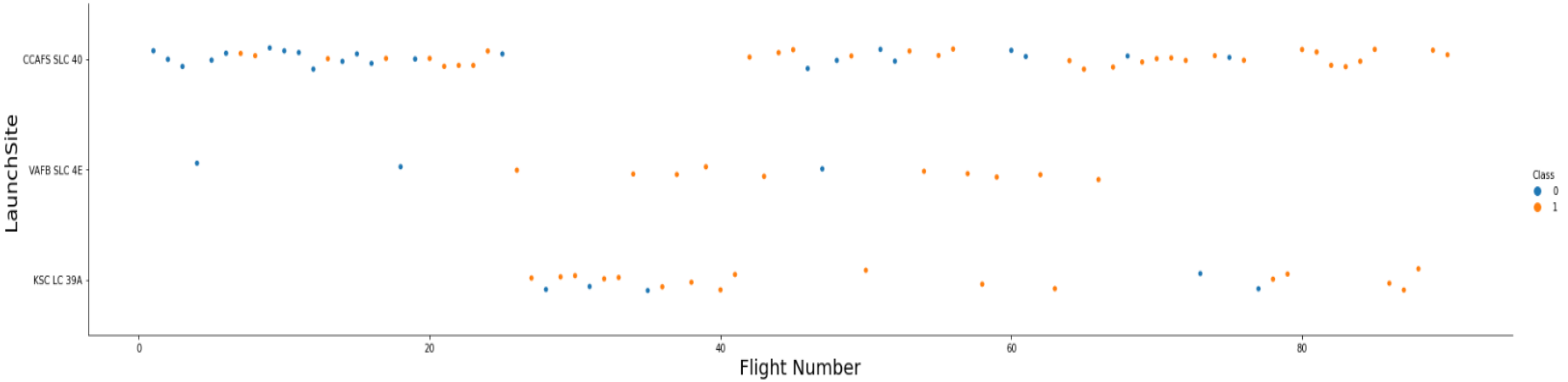
Model	Accuracy	TestAccuracy
LogReg	0.84643	0.83333
SVM	0.84821	0.83333
Tree	0.88929	0.61111
KNN	0.84821	0.83333

Section 2

Insights drawn from EDA

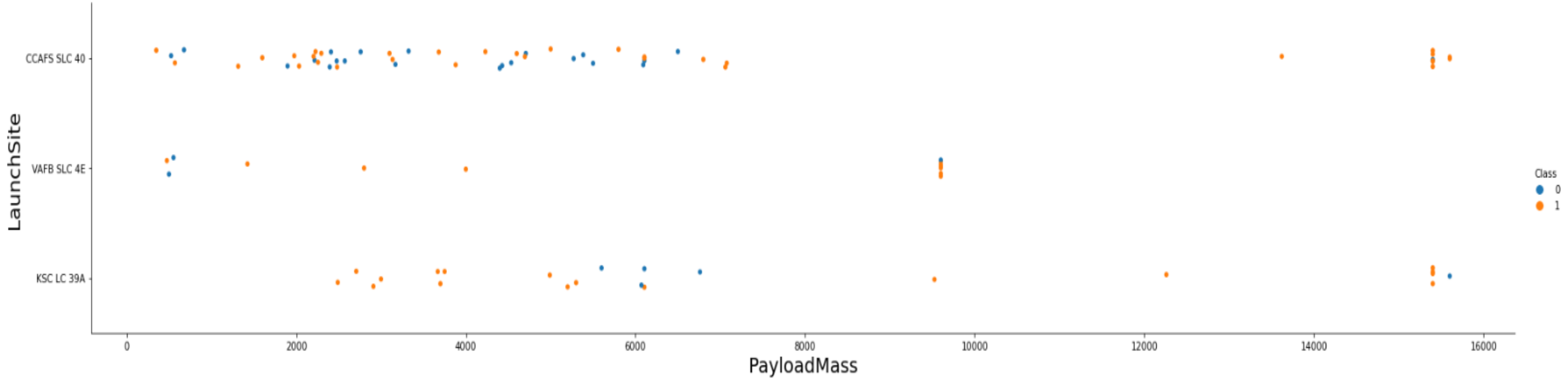
Flight Number vs. Launch Site

- We can hear that the launches from site CCAFS SLC 40 are significantly denser with flight numbers than the other sites



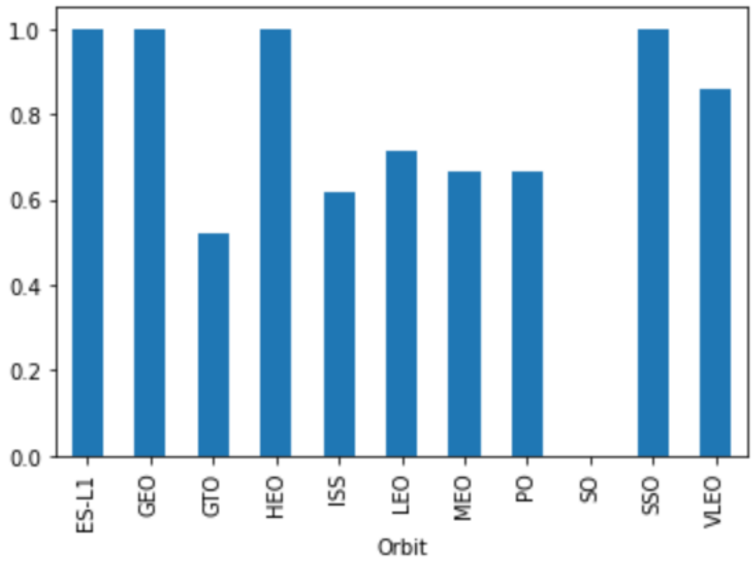
Payload vs. Launch Site

- As we can see most launches are with a PayloadMass below 8000 and LaunchSite CCAFS SLC 40 is where most of the launches.



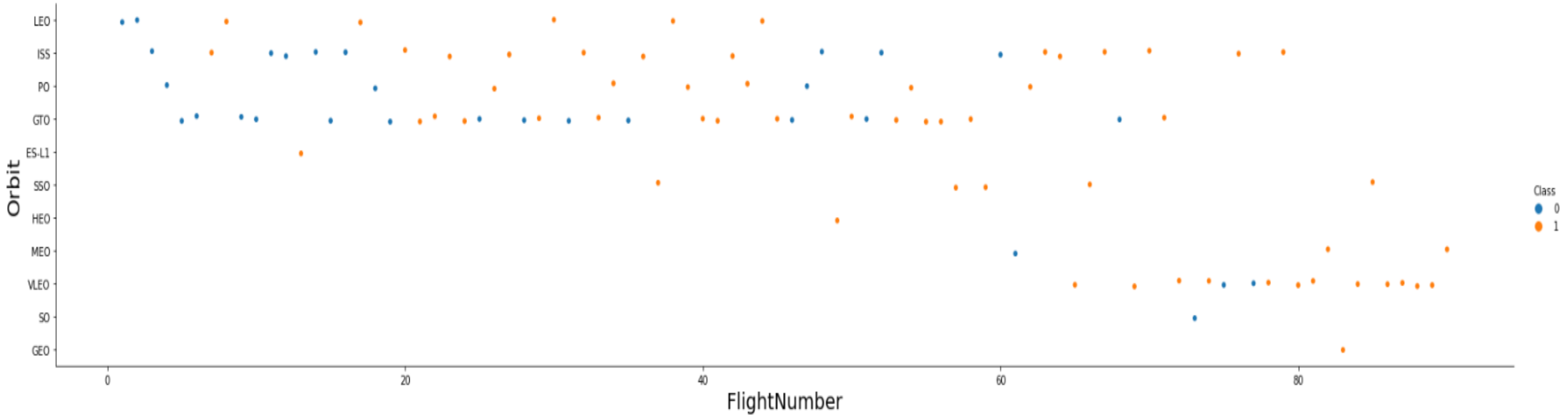
Success Rate vs. Orbit Type

- The Highest success rates of the orbit types are ES-L1, GEO, HEO, and SSO.



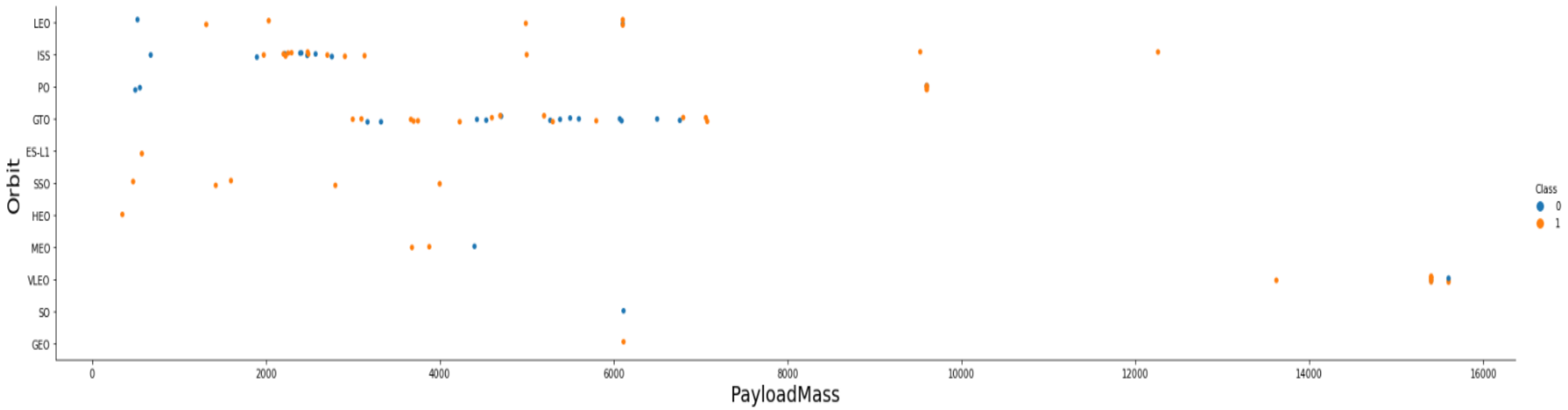
Flight Number vs. Orbit Type

- As we can see there is a major shift to targeting VLEO orbit



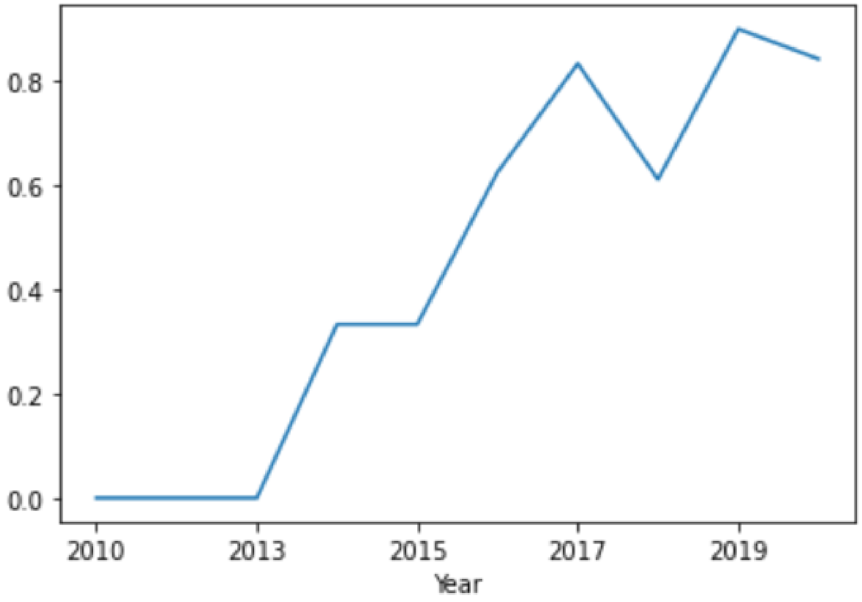
Payload vs. Orbit Type

- Orbit VLEO has a heavy PayloadMass while orbit type GTO which is between 2000 and 8000 PayloadMass, thus we see different PayloadMasses to different orbit types.



Launch Success Yearly Trend

- This graph shows an increasing success rate over the years



All Launch Site Names

```
sql SELECT DISTINCT LAUNCH_SITE FROM SPACEXTBL ORDER BY 1;
```

```
 * ibm_db_sa://fvp19040:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.datab
ases.appdomain.cloud:32733/bludb
Done.
```

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

Launch Site Names Begin with 'CCA'

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

```
sql SELECT * FROM SPACEXTBL WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5;
```

```
 * ibm_db_sa://fvp19040:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/bludb
Done.
```

DATE	time_utc_	booster_version	launch_site	payload	payload_mass_kg_	orbit	customer	mission_outcome	landing_outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-05-22	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

Total Payload Mass

```
sql SELECT SUM(PAYLOAD_MASS__KG_) AS TOTAL_PAYLOAD FROM SPACEXTBL WHERE PAYLOAD  LIKE '%CRS%';
```

```
 * ibm_db_sa://fvp19040:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/bludb
Done.
```

total_payload
111268

Average Payload Mass by F9 v1.1

```
sql SELECT AVG(PAYLOAD_MASS__KG_) AS AVG_PAYLOAD FROM SPACEXTBL WHERE BOOSTER_VERSION = 'F9 v1.1';
```

```
 * ibm_db_sa://fvp19040:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/bludb
Done.
```

avg_payload
2928

First Successful Ground Landing Date

```
sql SELECT MIN(DATE) AS FIRST_SUCCESS_GP FROM SPACEXTBL WHERE LANDING__OUTCOME = 'Success (ground pad)';
```

```
 * ibm_db_sa://fvp19040:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/bludb
Done.
```

first_success_gp
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

```
sql SELECT DISTINCT BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000 AND LANDING__OUTCOME = 'Success (drone ship)';
```

```
 * ibm_db_sa://fvp19040:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/bludb
Done.
```

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

Total Number of Successful and Failure Mission Outcomes

```
sql SELECT MISSION_OUTCOME, COUNT(*) AS QTY FROM SPACEXTBL GROUP BY MISSION_OUTCOME ORDER BY MISSION_OUTCOME;
```

```
 * ibm_db_sa://fvp19040:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/bludb
Done.
```

mission_outcome	qty
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

Boosters Carried Maximum Payload

```
sql SELECT DISTINCT BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL) ORDER BY BOOSTER_VERSION;
```

```
 * ibm_db_sa://fvp19040:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/bludb
Done.
```

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

2015 Launch Records

List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

```
sql SELECT BOOSTER_VERSION, LAUNCH_SITE FROM SPACEXTBL WHERE LANDING__OUTCOME = 'Failure (drone ship)' AND DATE_PART('YEAR', DATE) = 2015;
```

```
 * ibm_db_sa://fvp19040:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/bludb
Done.
```

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

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

```
sql SELECT LANDING__OUTCOME, COUNT(*) AS QTY FROM SPACEXTBL WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY LANDING__OUTCOME ORDER BY QTY DE
```

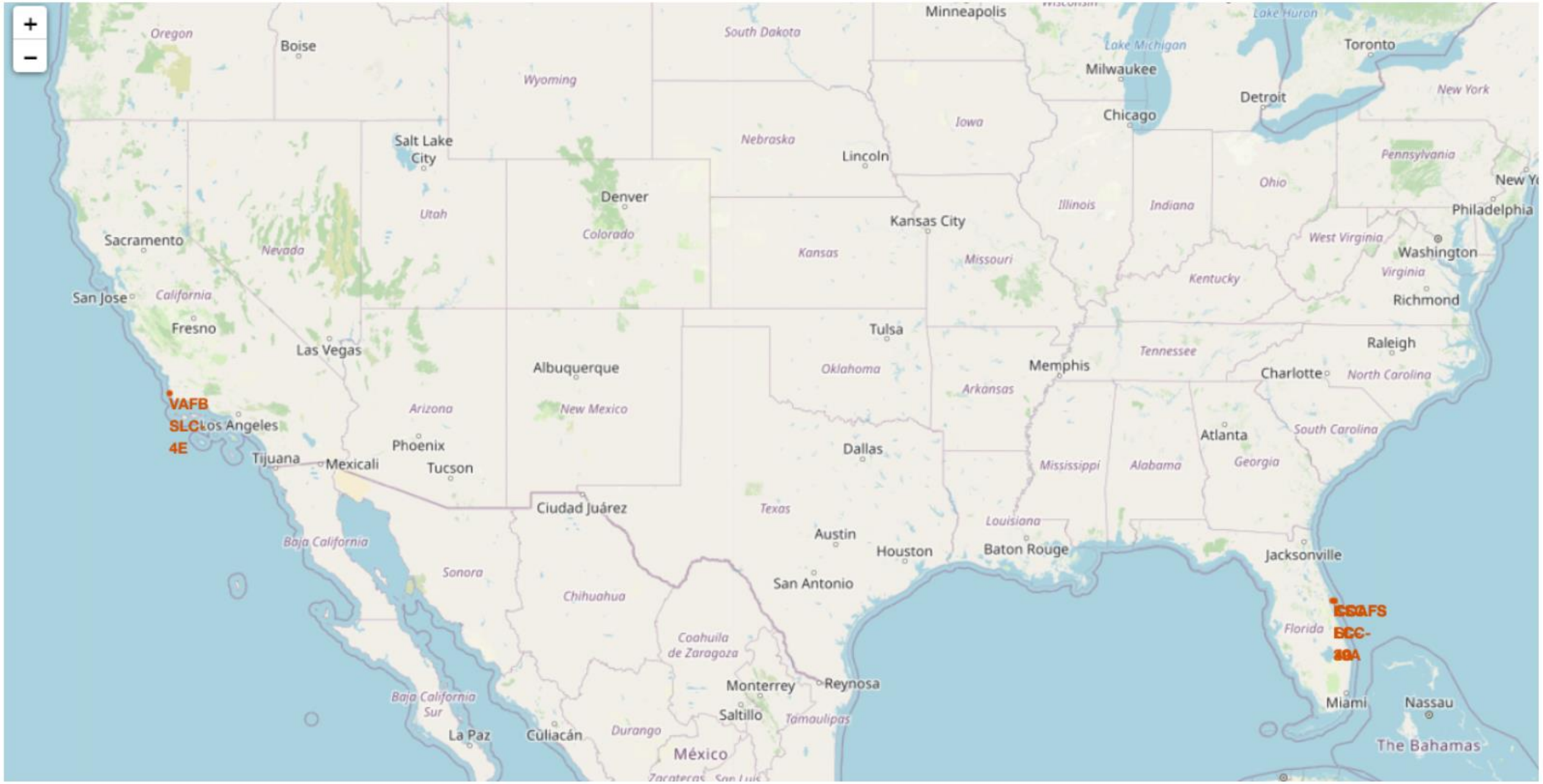
```
 * ibm_db_sa://fvp19040:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/bludb
Done.
```

landing__outcome	qty
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

Section 3

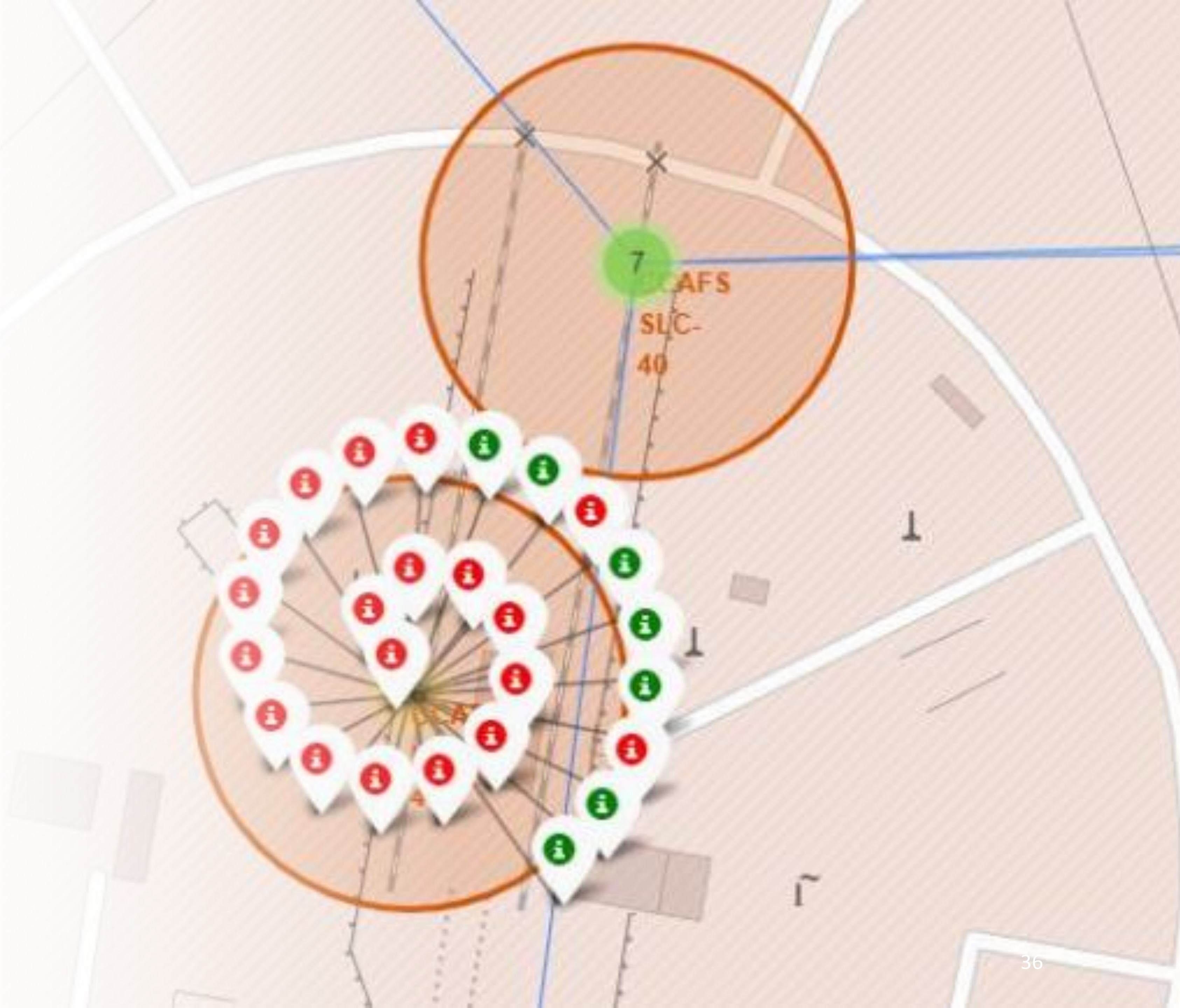
Launch Sites Proximities Analysis

Map of the Launch Sites



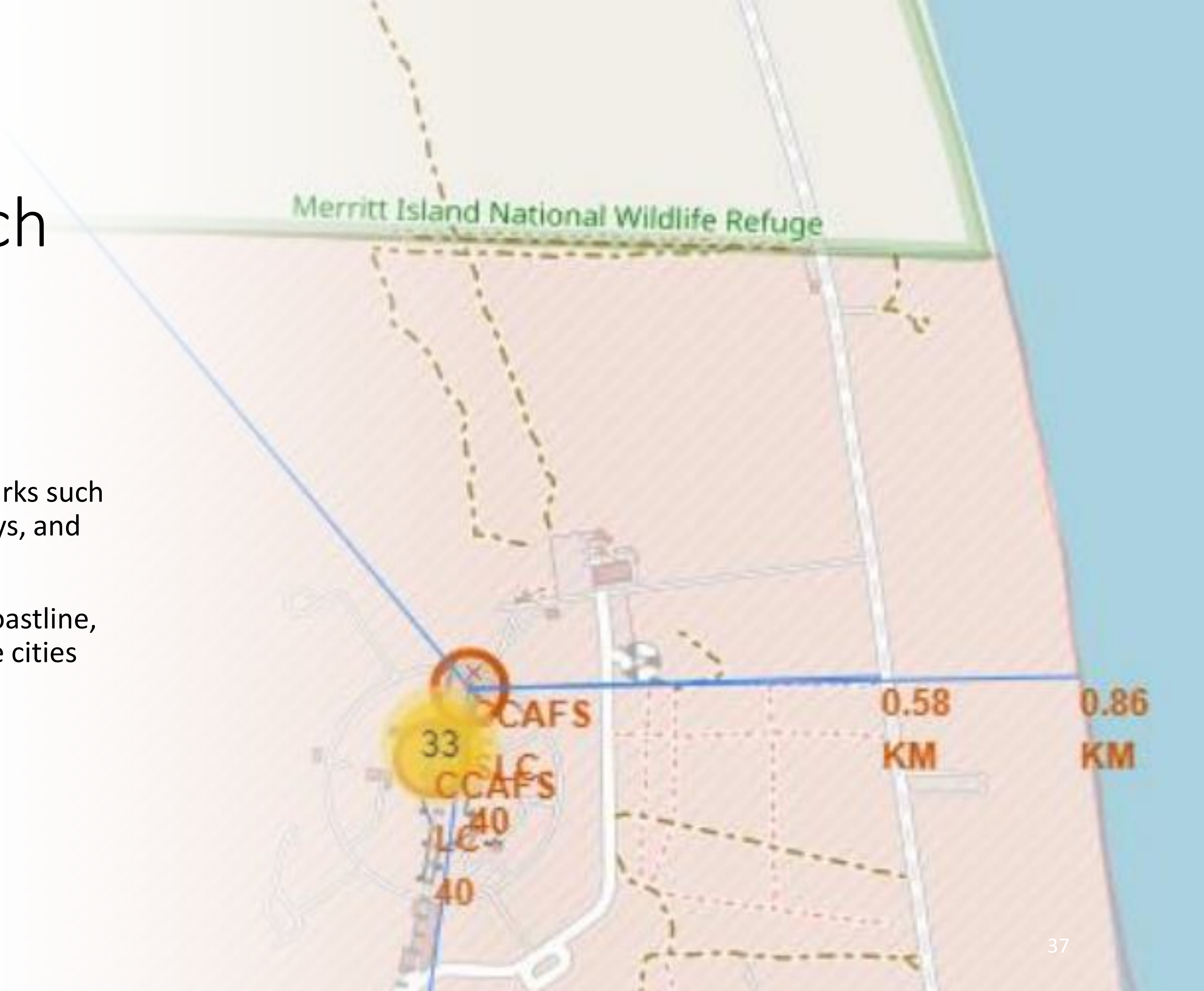
Launch Sites Color Marked

- Florida Site
- Green = Successful
- Red = Unsuccessful
- Markers of unsuccessful and successful launches



The distance between launch sites and landmarks

- Launch site distance to landmarks such as coastlines, railways, highways, and cities
- Launch sites are close to the coastline, railway, and highways but large cities are far away

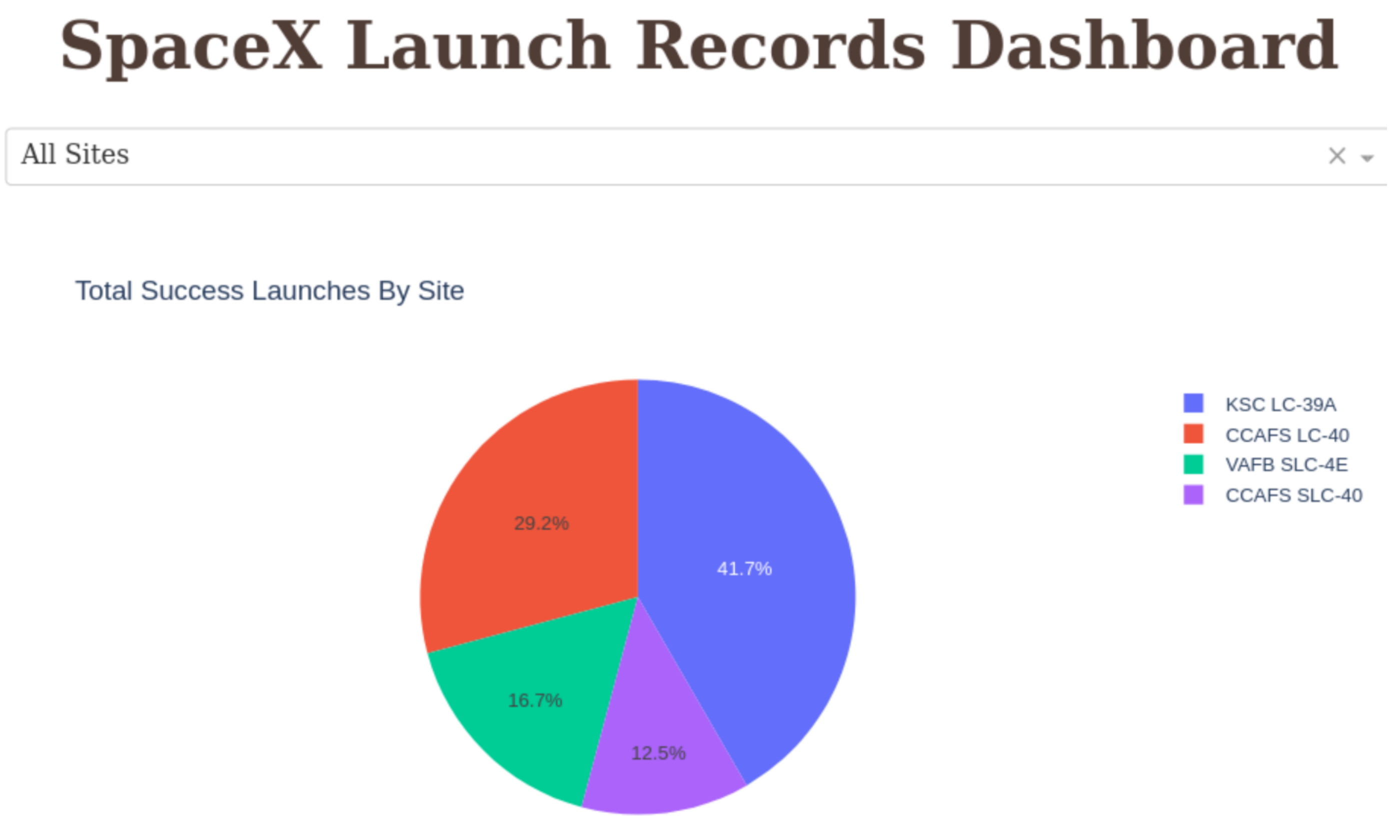


Section 4

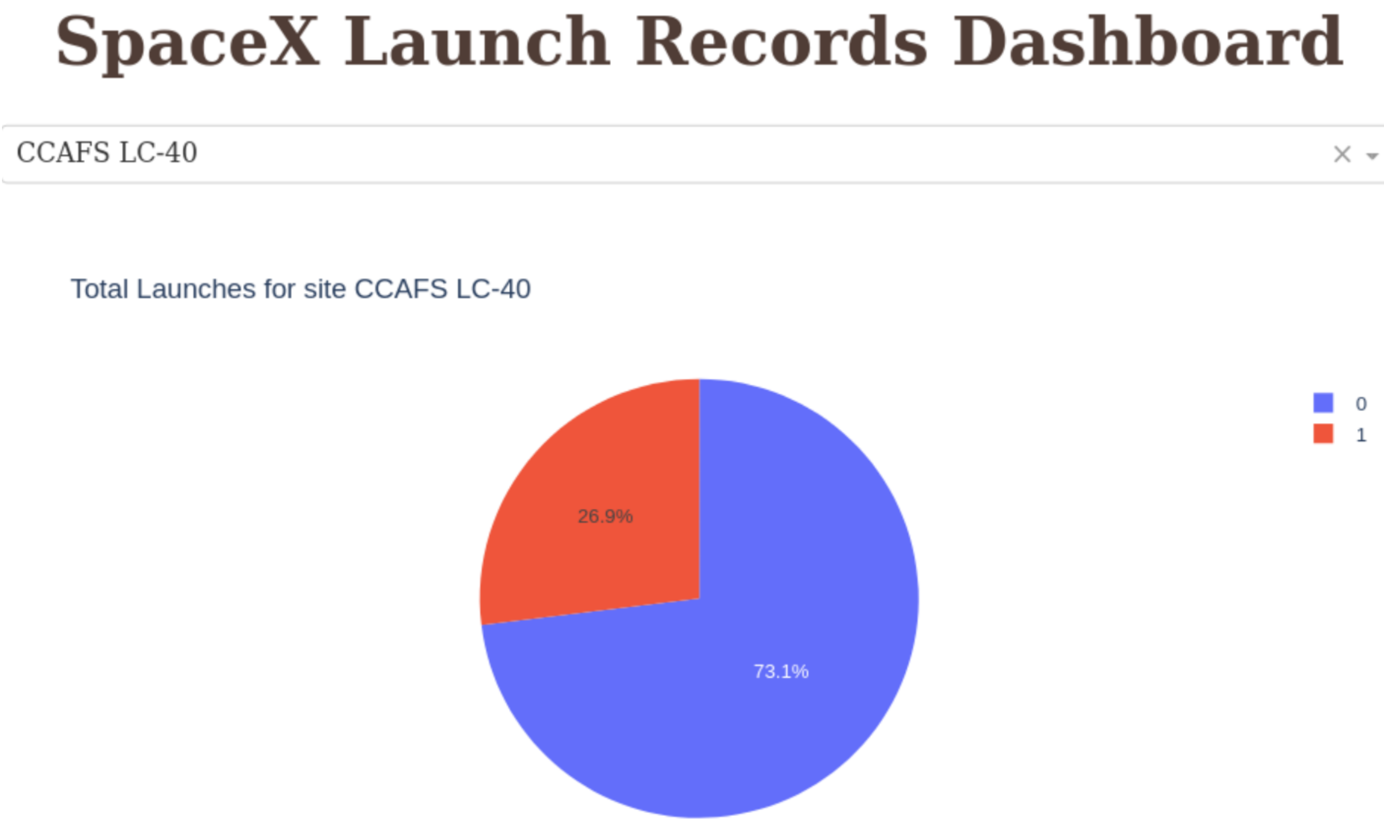
Build a Dashboard with Plotly Dash

Total Success Launches By Site

- KSC LC-39A is the highest percentage of success
- CCAFS SLC-40 is the lowest percentage of success



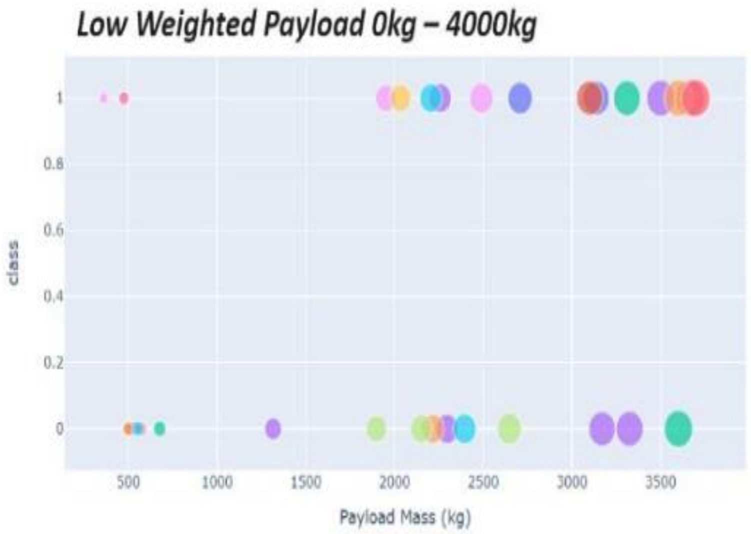
Total Launches for site CCAFS LC-40

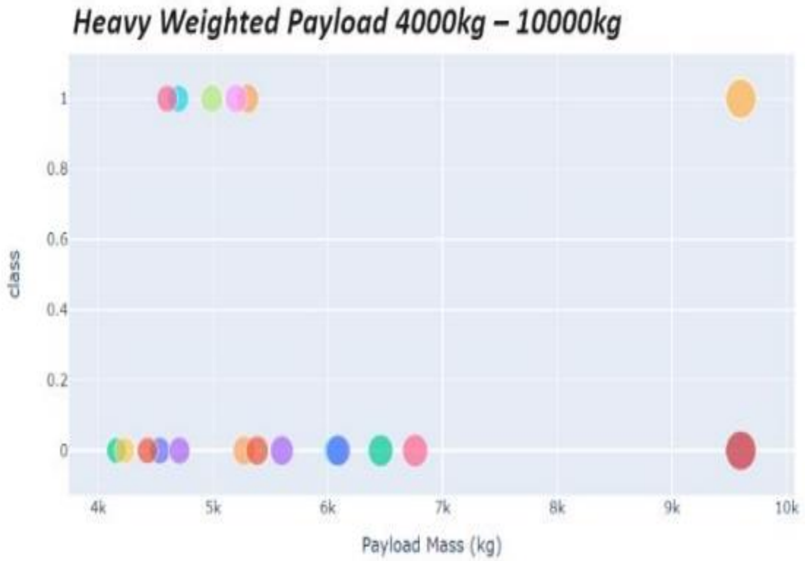


- We made a Pie chart showing the launch site with the highest launch success Ratio

Total Success Launches By Site

- Scatter plots of PayloadMass vs Launch Outcome for all sites, with different payloads selected in the range slider



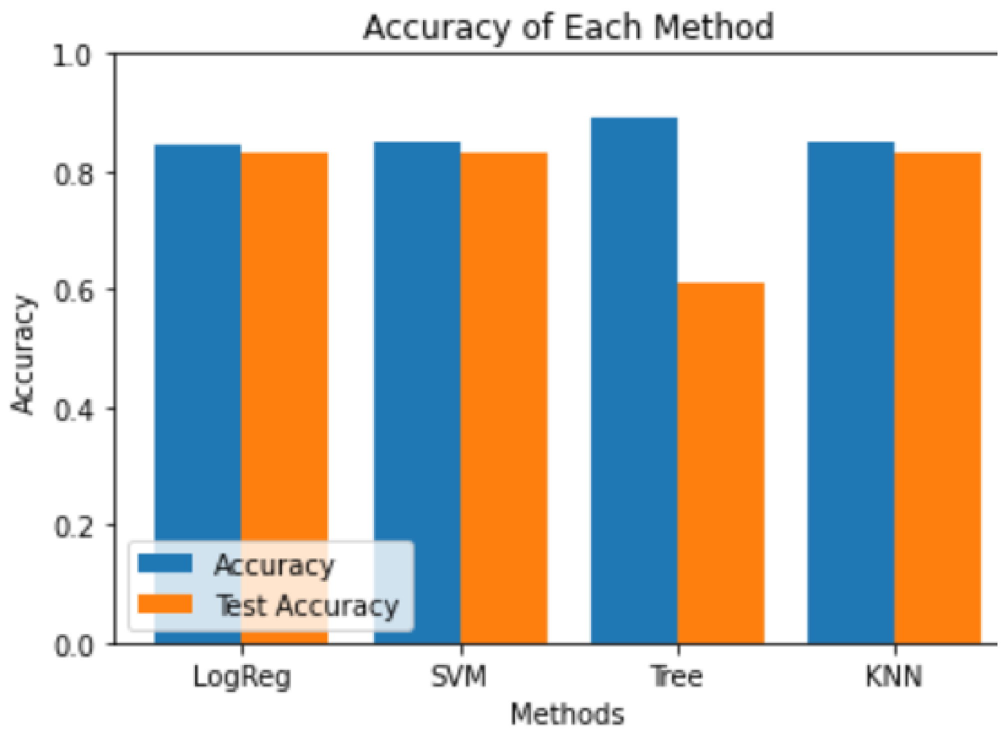


Section 5

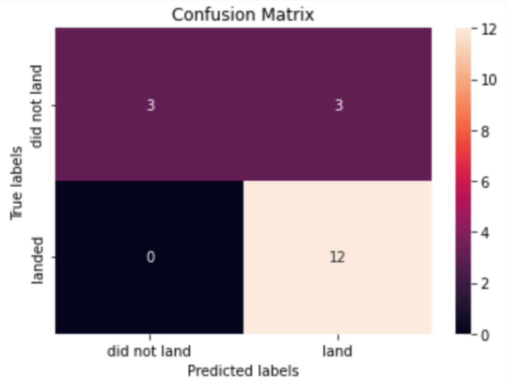
Predictive Analysis (Classification)

Classification Accuracy

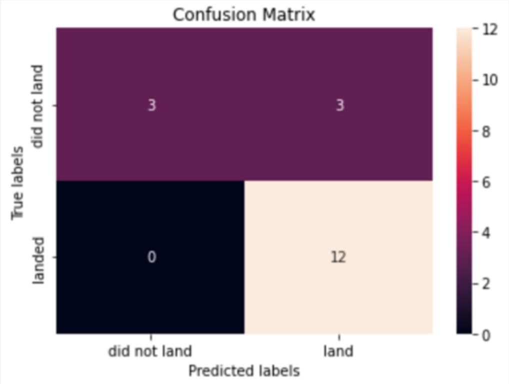
- We see this bar chart the accuracy is almost the same in all our models.
- The test accuracy is important for transparency



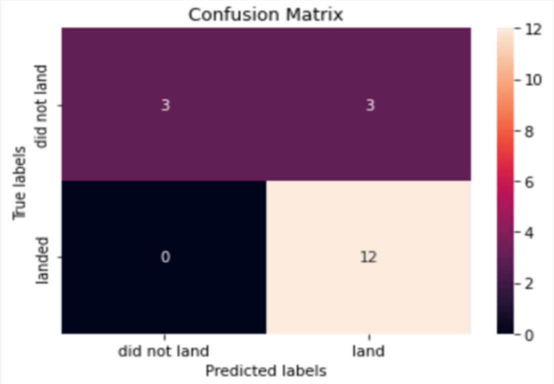
Confusion Matrix



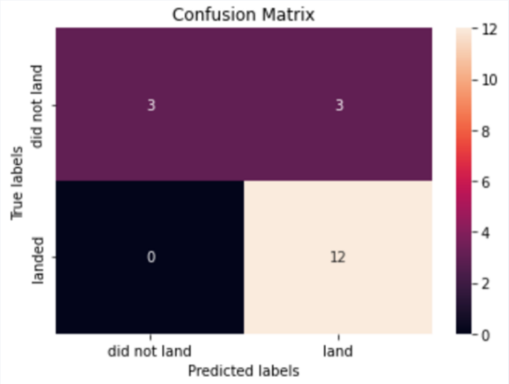
LogRe
g



SVM



KNN



Tre
e

Here are the four confusion matrices of our four models. We can see they are all the same which supports the accuracy score.

Conclusions

We can conclude that:

- The payloads with lower mass have more success than the ones with more masses.
- The larger the flight amount at a launch site, the greater the success rate at a launch site.
- The LC-39A site had the most successful landing
- Launch success rates started to increase in 2013 and till 2020.
- Orbits types: - ES-L1, GEO, HEO, SSO, and VLEO had the most success rate.
- KSC LC-39A had the most successful launches of any site.
- After training four different machine learning models we can choose every model for our prediction due to them showing promising and accurate scores.

Thank you!