IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Jorge Arrocha
September 19, 2021.

GitHub repository: https://github.com/Jorge-Arrocha/Applied-Data-Science-Capstone

Outline

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

Executive Summary

- SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars. Other providers cost upwards of 165 million dollars each.
- Unlike other rocket providers, SpaceX's Falcon 9 can recover the first stage, while the second stage is usually left to decay in orbit or directed to burn up in the planet's atmosphere.
- On a single $62 million rocket launch, the first stage makes up 60% of the total cost ($37.2 million), the second stage comprises 20% ($12.4 million), and the fairing cost is 10% ($6.2 million).
- Therefore, if we can determine if the first stage will land, we can determine the cost of a launch.
- To be able to predict if SpaceX will recover the first stage, we trained machine learning models with public information and selected the one with the highest accuracy.

Introduction

- The commercial space age is here, companies are making space travel affordable for everyone.
- Virgin Galactic is providing suborbital spaceflights.

- ROCKET LAB is a small satellite provider.

- Blue Origin manufactures sub-orbital and orbital reusable rockets.
- Perhaps the most successful is SPACEX

- Their accomplishments include:
  - ✓ Sending spacecraft to the International Space Station.
  - ✓ Starlink, a satellite internet constellation providing satellite Internet access.
  - ✓ Sending manned missions to space.
- One reason SPACEX can do this is the rocket launches are relatively inexpensive, because they can reuse the first stage.

- The purpose of this project is to find a model that will predict, with accuracy, if Falcon 9's first stage will return successfully.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - From the SpaceX REST API we obtained launch data, such as, rocket used, payload delivered, launch specifications, landing specifications, and landing outcome.
  - We targeted a specific endpoint of the API to get past launch data.
  - The response was a list of JSON objects each representing a launch. This structured json data was then converted to a flat-table Pandas DataFrame.
  - Another data source for obtaining Falcon 9 Launch data was web scraping related Wiki pages.
  - We used the Python BeautifulSoup package to web scrape some HTML tables that contained valuable Falcon 9 launch records.
  - The data parsed from those tables was then converted to a Pandas DataFrame.
  - In Watson Studio we loaded the dataframe into a database on the IBM Cloud for further analysis with SQL queries.

Data Collection

- We first targeted a specific endpoint of the SpaceX REST API to get past launch data:
  - The response from the URL 'https://api.spacexdata.com/v4/launches/past' was a list of json objects, each representing a launch.
  - We used the json_normalize method to convert the json response into a Pandas DataFrame.
  - We noticed that some columns didn't have information in them, but some type of identification number.
  - We were provided with a series of helper functions which extracted from alternate endpoints of the same API, the meanings of those ID numbers.
  - From the rocket column we learned the booster's name. Since this project will train models with information about the **Falcon 9 rocket**, we eliminated all rows pertaining to the Falcon 1 rocket booster.
  - From the launchpad column we learned the name of the launch site being used, its latitude and longitude.
  - From the payload column we learned the mass of the payload and the orbit that it went to.
  - From the cores column we learned the outcome of the landing, the type of landing, number of flights with that core, whether grid fins were used, whether the core is reused, whether legs were used, the landing pad used, the block of the core which is a number used to separate version of cores, the number of times this specific core has been reused, and the serial of the core.

Data Collection

- The dataframe constructed with the information from the API included the following attributes:

  - ✓ FlightNumber
  - ✓ Date
  - ✓ BoosterVersion
  - ✓ PayloadMass
  - ✓ Orbit
  - ✓ LaunchSite
  - ✓ Outcome
  - ✓ Flights
  - ✓ GridFins
  - ✓ Reused
  - ✓ Legs
  - ✓ LandingPad
  - ✓ Block
  - ✓ ReusedCount
  - ✓ Serial
  - ✓ Longitude
  - ✓ Latitude

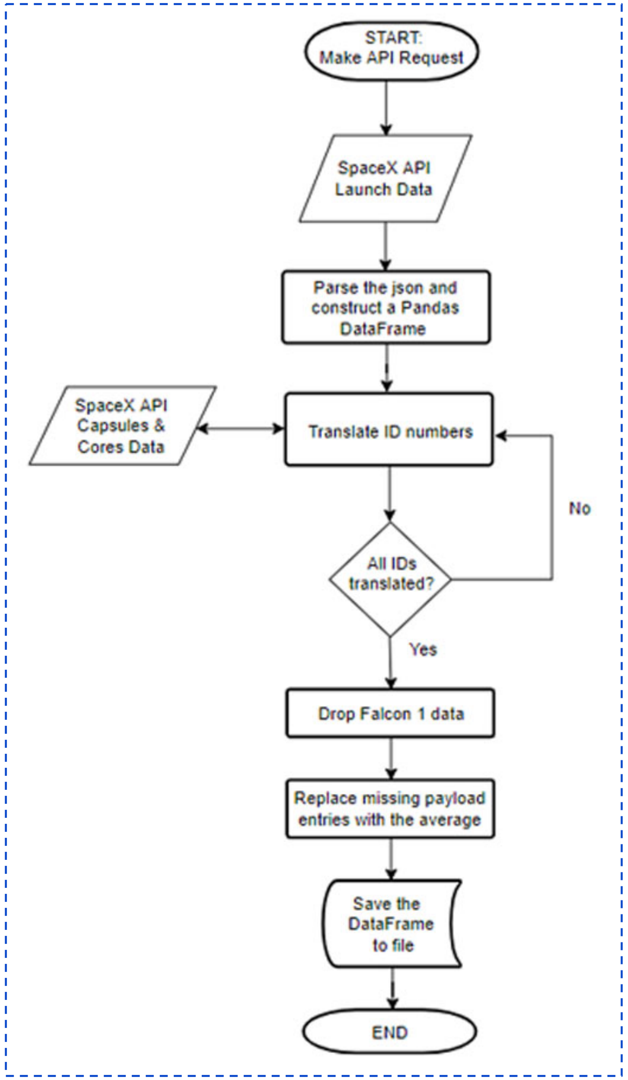
- **After constructing a Pandas DataFrame with the SpaceX API launch data, we proceeded to web scrape launch information from wiki pages:**
  - To keep the project tasks consistent, we were asked to scrape the data from a snapshot of the List of Falcon 9 and Falcon Heavy launches in Wikipedia, updated June 9, 2021.
  - We used the BeautifuSoup library and some helper functions that were provided to aid with the parsing of the html tables obtained from the wiki page.
  - After parsing the html code, we constructed a new Pandas DataFrame with the following attributes:

    - ✓ Flight №
    - ✓ Launch site
    - ✓ Payload
    - ✓ Payload mass
    - ✓ Orbit
    - ✓ Customer
    - ✓ Launch outcome
    - ✓ Booster Version
    - ✓ Booster landing
    - ✓ Date
    - ✓ Time

  - In Watson Studio we loaded the dataframe into a database on the IBM Cloud, so that we could obtain valuable new information with SQL queries.

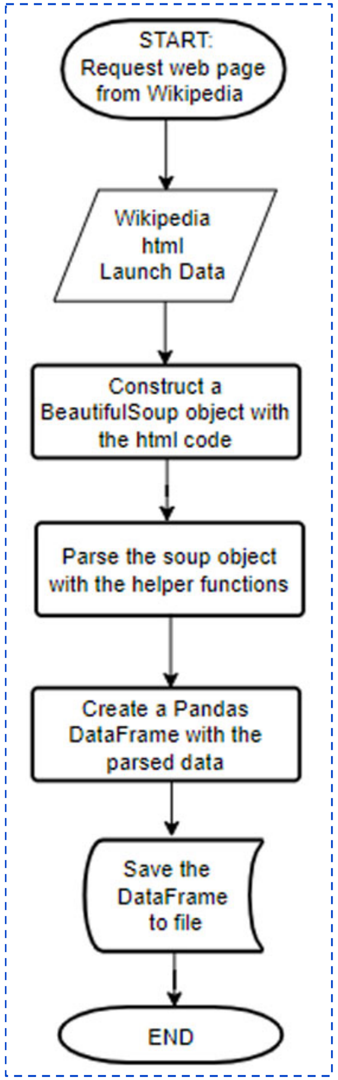
Data Collection – SpaceX API

1) Request launch data from SpaceX API.
2) Parse the json and construct a Pandas DataFrame.
3) Translate the ID numbers with multiple calls to alternate endpoints of the API.
4) Drop all Falcon 1 data.
5) Replace missing payload entries with the average.
6) GitHub URL of the completed SpaceX Data Collection API lab: https://github.com/Jorge-Arrocha/Applied-Data-Science-Capstone/blob/master/SpaceX_Data_Collection_API_Lab.ipynb



Data Collection – Scraping

1) Request launch data from the Wikipedia URL.
2) Construct a BeautifulSoup object with the html code obtained from the request.
3) Parse the soup object with the helper functions.
4) Construct a Pandas DataFrame with parsed data.
5) GitHub URL of the completed Web Scraping Falcon 9 Launches lab: https://github.com/Jorge-Arrocha/Applied-Data-Science-Capstone/blob/master/Web_Scraping_Falcon_9_Launches.ipynb



Data Wrangling

- We transformed the collected raw information into a clean dataset which provided meaningful data on the situations we were trying to address:

1. Wrangling Data using an API.
   - Because the information contained in some columns was not actual data but some identification numbers, we had to use the SpaceX API again, targeting other endpoints to gather specific data for each ID number.
   - The following helper functions were provided to aid in the collection of each ID number's translated data:

     1) getBoosterVersion (API targeted: https://api.spacexdata.com/v4/rockets)
     2) getLaunchSite (API targeted: https://api.spacexdata.com/v4/launchpads)
     3) getPayloadData (API targeted: https://api.spacexdata.com/v4/payloads)
     4) getCoreData (API targeted: https://api.spacexdata.com/v4/cores)

2. Sampling Data.
   - Another issue we found is that the launch information we obtained included data for the Falcon 1 booster and, for our models, we were only interested in Falcon 9's.
   - All rows pertaining to the Falcon 1 rocket booster were dropped from the dataset.

3. Dealing with Nulls.
   - Because not all gathered data is perfect, we ended up with some entries that contained NULL values.
   - In order to make the dataset viable for analysis, we replaced the NULL entries in the PayloadMass column with the average weight of all the payloads contained in the column.

Data Wrangling

4. Outcome.
   - We converted the landing outcomes to Classes y (either 0 or 1).
   - This is the classification variable that represents the outcome of each launch. If the value is zero, the first stage did not land successfully; one means the first stage landed successfully.
5. Turning categorical variables into quantitative values: One-hot Encoding.
   - Many features in the dataset represent categorical variables, with contents expressed in the form of string objects.
   - Most statistical models cannot take in objects or strings as input and for model training only take numbers.
   - One-hot Encoding adds new features corresponding to each unique element in the original feature we would like to encode.
   - The presence of an element in the original feature is set to 1 in its corresponding new feature column, while the rest of the new feature columns created are set to 0.
   - The process is repeated for every categorical variable.
   - Once every string entry from every categorical variable has been set to 1 in a corresponding new feature column (also placing the zeroes where they belong), the columns with original features and their string entries are then dropped from the dataset, leaving it with all number entries.
   - We used the pandas.get_dummies() method to convert the categorical variables to dummy variables of zeroes or ones.

- GitHub URL of the completed SpaceX Data Wrangling lab: https://github.com/Jorge-Arrocha/Applied-Data-Science-Capstone/blob/master/SpaceX_Data_Wrangling.ipynb

EDA with Data Visualization

- To perform Exploratory Data Analysis, we plotted the following charts:
    1. To visualize the relationship between Flight Number and Launch Site, we plotted a scatter point chart with both features, and found that different launch sites had different success rates. CCAFS LC-40 had a success rate of 60%, while KSC LC-39A and VAFB SLC-4E had success rates of 77%.
    2. To visualize the relationship between Payload and Launch Site, we plotted a scatter point chart with both features, and found that missions of site CCAFS SLC-40 with payloads under 8,000 Kg had a lower success rate than those with greater mass. Also, payload mass had no apparent effect on sites VAFB SLC-4E and KSC LC-39A.
    3. To visualize the relationship between success rate of each orbit type, we plotted a bar chart with the success rate of every orbit type, and found a varied assortment of values for the different orbits.
    4. To visualize the relationship between FlightNumber and Orbit type, we plotted a scatter point chart with both features, and we saw that in the LEO orbit the success appeared related to the number of flights; on the other hand, there seemed to be no relationship between flight number when in the GTO orbit.
    5. To visualize the relationship between Payload and Orbit type, we plotted a scatter point chart with both features, and found that missions with payloads under 4,000 Kg in the ISS orbit had a lower success rate than those with greater mass. Payload mass didn’t have any effect when in the GTO orbit.
    6. To visualize the launch success yearly trend, we plotted a line chart of the average success rate for the various years in the dataset, and found that the success rate started increasing in 2014 and continued growing until 2020.
- GitHub URL of the completed EDA with Data Visualization lab: https://github.com/Jorge-Arrocha/Applied-Data-Science-Capstone/blob/master/EDA_with_Data_Visualization.ipynb

EDA with SQL

- The following tasks were completed making SQL queries to an IBM Cloud database containing the data scraped from a wiki page:
  1) Display the names of the unique launch sites in the space mission.
  2) Display 5 records where launch sites begin with the string 'CCA'.
  3) Display the total payload mass carried by boosters launched by NASA (CRS).
  4) Display average payload mass carried by booster version F9 v1.1
  5) List the date when the first successful landing outcome in ground pad was achieved.
  6) List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000.
  7) List the total number of successful and failure mission outcomes.
  8) Using a subquery, list the names of the booster_versions which have carried the maximum payload mass.
  9) List the failed landing_outcomes in drone ship, their booster versions, and launch site names for the in year 2015
  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.
- GitHub URL of the completed 'EDA with SQL' lab: https://github.com/Jorge-Arrocha/Applied-Data-Science-Capstone/blob/master/EDA_with_SQL.ipynb

Build an Interactive Map with Folium

- The launch success rate may depend on many factors such as payload mass, orbit type, and so on.
- It may also depend on the location and proximities of a launch site, i.e., the initial position of rocket trajectories.
- Finding an optimal location for building a launch site certainly involves many factors and hopefully we could discover some of the factors by analyzing the existing launch site locations.
- To analyze launch site geo and proximities, circle markers with a 50-meter radius were placed on the 4 launching sites, as determined by their latitude and longitude coordinates.
- Color-coded flags indicating the successful and failed launch outcomes were created for each site.
- With the help of Folium's MapPosition and PolyLine plugins, lines were drawn from the Vandenberg Space Launch Complex 4E to four near landmarks, specifically, the nearest city Lumpoc, California, nearest road, LaSalle Canyon Road, nearest railway, the Santa Barbara Subdivision MT1, and the nearest coastline, which happens to be the Vandenberg State Marine Reserve. Each line was labeled with the distance it covered.
- GitHub URL of the completed 'Interactive Visual Analytics with Folium' lab: https://github.com/Jorge-Arrocha/Applied-Data-Science-Capstone/blob/master/Interactive_Visual_Analytics_with_Folium.ipynb

Build a Dashboard with Plotly Dash

- Interactive visual analytics enables users to explore and manipulate data interactively with real-time responses. Common interactions include zoom-in and zoom-out, pan, filter, search, and link.
- With interactive visual analytics, users can find visual patterns faster and more effectively.
- To perform interactive visual analytics with the SpaceX data, we built a dashboard application with the Python Plotly Dash package.
- The dashboard application contained input components such as a dropdown list, and a range slider that interacted with a pie chart and scatter point chart.
- The pie chart could display the percentage of successful launches of all sites relative to each other, or, with the help of a drop-down list, an individual site could be chosen to display its successful rate alone.
- The other chart on the dashboard was a scatter plot, which displayed the success rate of the booster version categories that fell within the range of payload masses selected by way of a slider.
- With the aid of the dashboard we could determine that the Kennedy Space Center had the highest rate of successful launches.
- GitHub URL of the completed Plotly Dash lab: https://github.com/Jorge-Arrocha/Applied-Data-Science-Capstone/blob/master/Build_an_Interactive_Dashboard_with_Ploty_Dash.ipynb

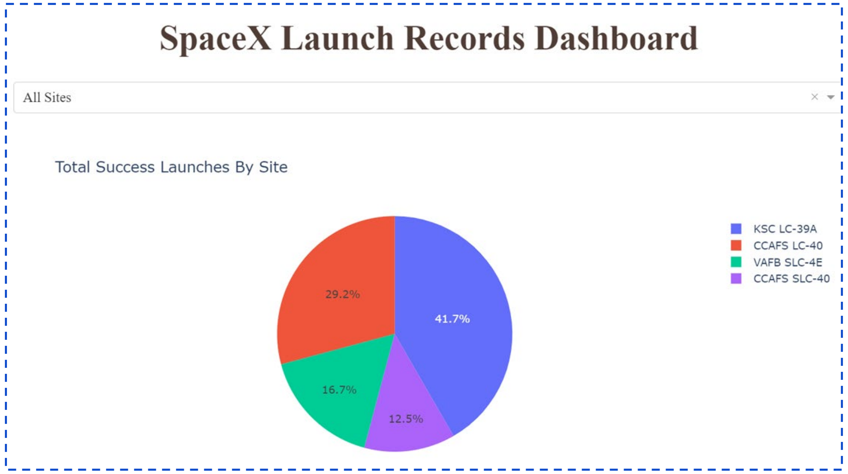
Predictive Analysis (Classification)

- To predict if the first stage of the Falcon 9 lands successfully, we built a machine learning pipeline.
- The process included Preprocessing, allowing us to standardize our data, and Train_test_split, allowing us to split our data into training and testing sets.
- We trained four different models and performed Grid Search with each one, enabling the finding of the hyperparameters that allow a given algorithm to perform best.
- Using the best hyperparameter values we determined the model with the best accuracy using the training data.
- The methods used for training the four different models were: LogisticRegression, Support Vector Machine, Decision Tree, and K Nearest Neighbor.
- Finally, we produced a confusion matrix to detect true and false, positive and negative predictions.
- GitHub URL of the completed ‘Predictive Analysis’ lab: https://github.com/Jorge-Arrocha/Applied-Data-Science-Capstone/blob/master/SpaceX_Machine_Learning_Prediction.ipynb

Results

- Exploratory data analysis results:
  1. As the flight number increases, the first stage is more likely to land successfully.
  2. The more massive the payload, the less likely the first stage will return.
  3. CCAFS SLC-40 has a success rate of 60%, while KSC LC-39A and VAFB SLC-4E have a success rates of 77%.
  4. Missions of site CCAFS SLC-40 with payloads under 8,000 Kg had a lower success rate than those with greater mass.
  5. Payload mass had no apparent effect on sites VAFB SLC-4E and KSC LC-39A.
  6. In the LEO orbit the success appeared related to the number of flights; on the other hand, there seemed to be no relationship between flight number when in GTO orbit.
  7. Missions with payloads under 4,000 Kg in the ISS orbit had a lower success rate than those with greater mass. Payload mass didn’t affect the success rate when in the GTO orbit.
  8. The overall success rate started increasing in 2014 and has continued growing until 2020.

- With interactive analytics we were able to determine that the Kennedy Space Center had the highest share of successful launches.

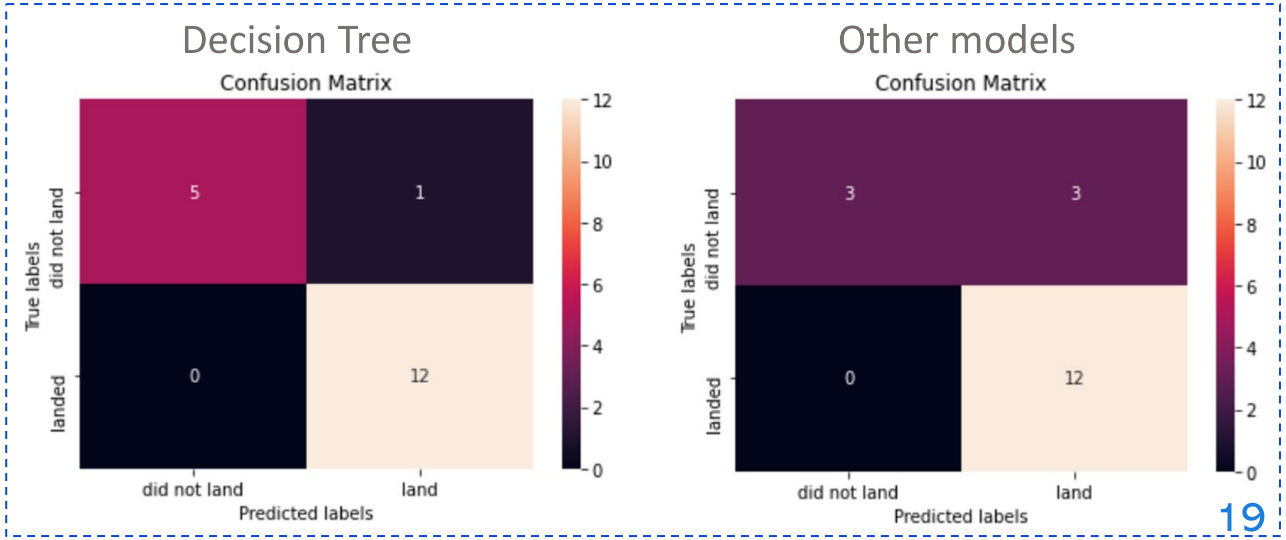


Results

- Predictive analysis results:
  - We trained four different models and performed Grid Search with each one, to find the best hyperparameters.
  - We used the score method on every model to measure the accuracy produced by the hyperparameters found, and obtained the following results:

Model	Accuracy
1) LogisticRegression	83%
2) Support Vector Machine	83%
3) Decision Tree	94%
4) K Nearest Neighbor	83%

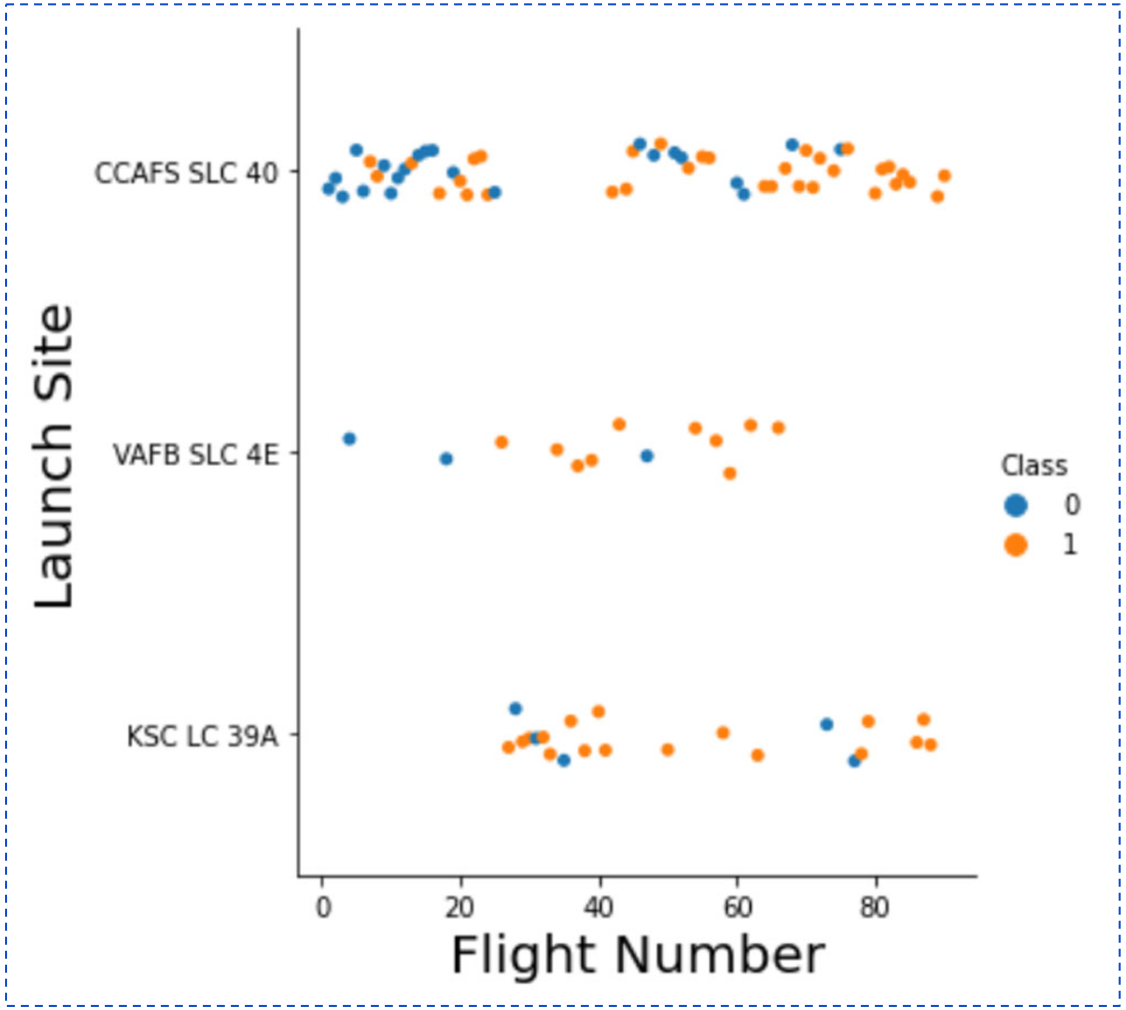
  - Finally, we plotted a confusion matrix for every model, and found that the Decision Tree only had one false positive and zero false negatives, while the other models had three false positives.



Section 2

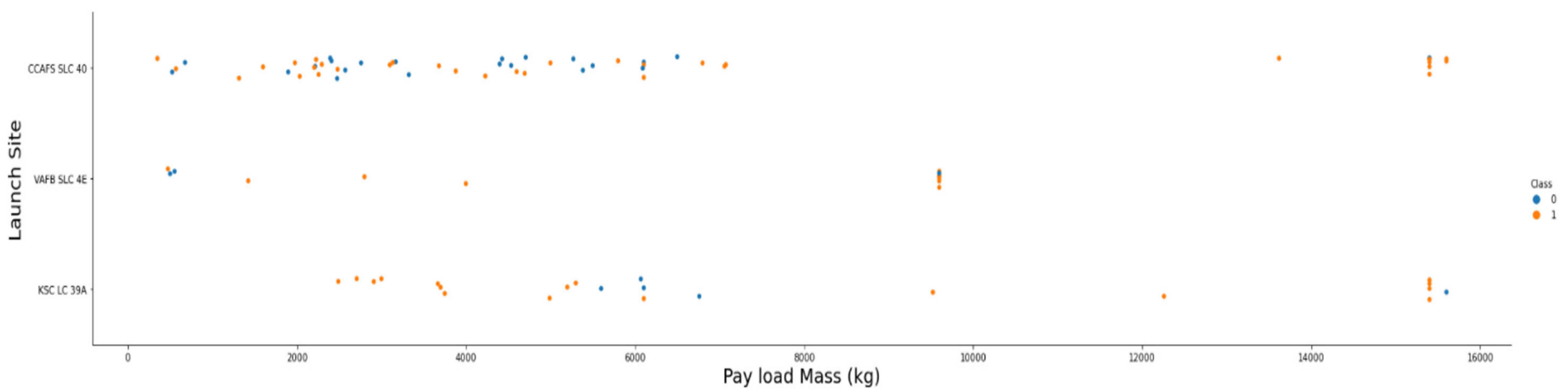
Insights drawn from EDA

Flight Number vs. Launch Site



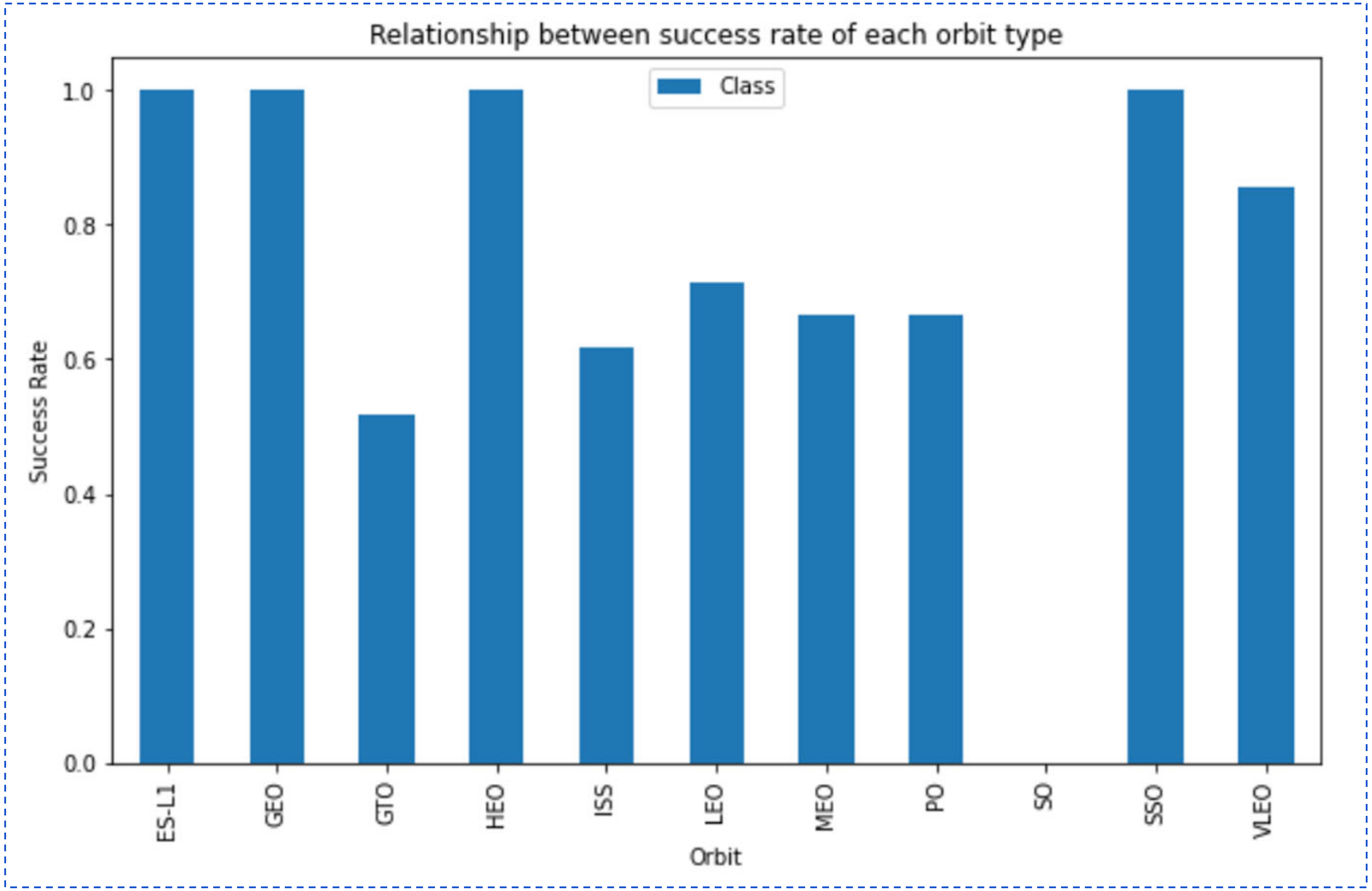
This plot shows how the flight number affects the success rate of each launch site. We saw that different launch sites had different success rates. CCAFS LC-40 had a success rate of 60%, while KSC LC-39A and VAFB SLC-4E had success rates of 77%.

Payload vs. Launch Site



This plot shows how the payload mass affects the success rate of each launch site. Missions of site CCAFS SLC-40 with payloads under 8,000 Kg had a lower success rate than those with greater mass. Payload mass had no apparent effect on sites VAFB SLC-4E and KSC LC-39A.

Success Rate vs. Orbit Type

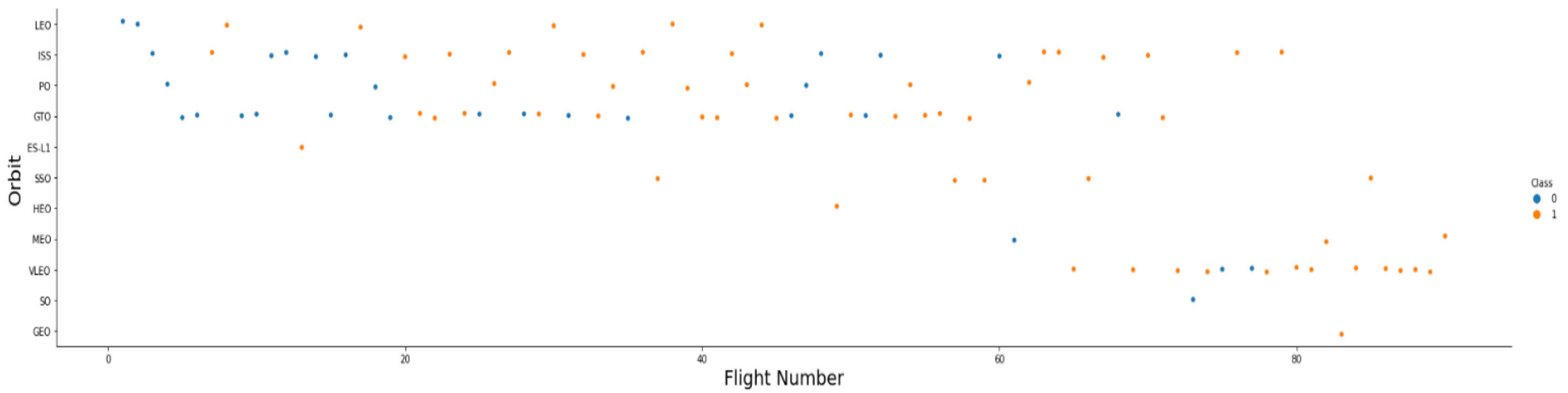


This bar graph displays the success rate of each orbit type, showing a varied assortment of values.

☞ To see an interactive image of different orbits with a few known satellites, please visit:

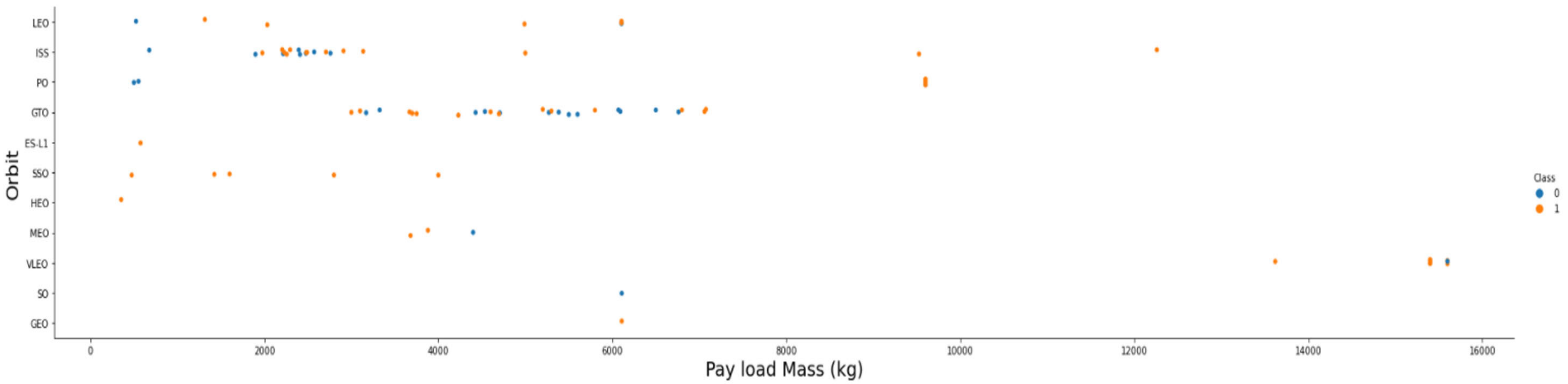
https://upload.wikimedia.org/wikipedia/commons/b/b4/Comparison_satellite_navigation_orbits.svg

Flight Number vs. Orbit Type



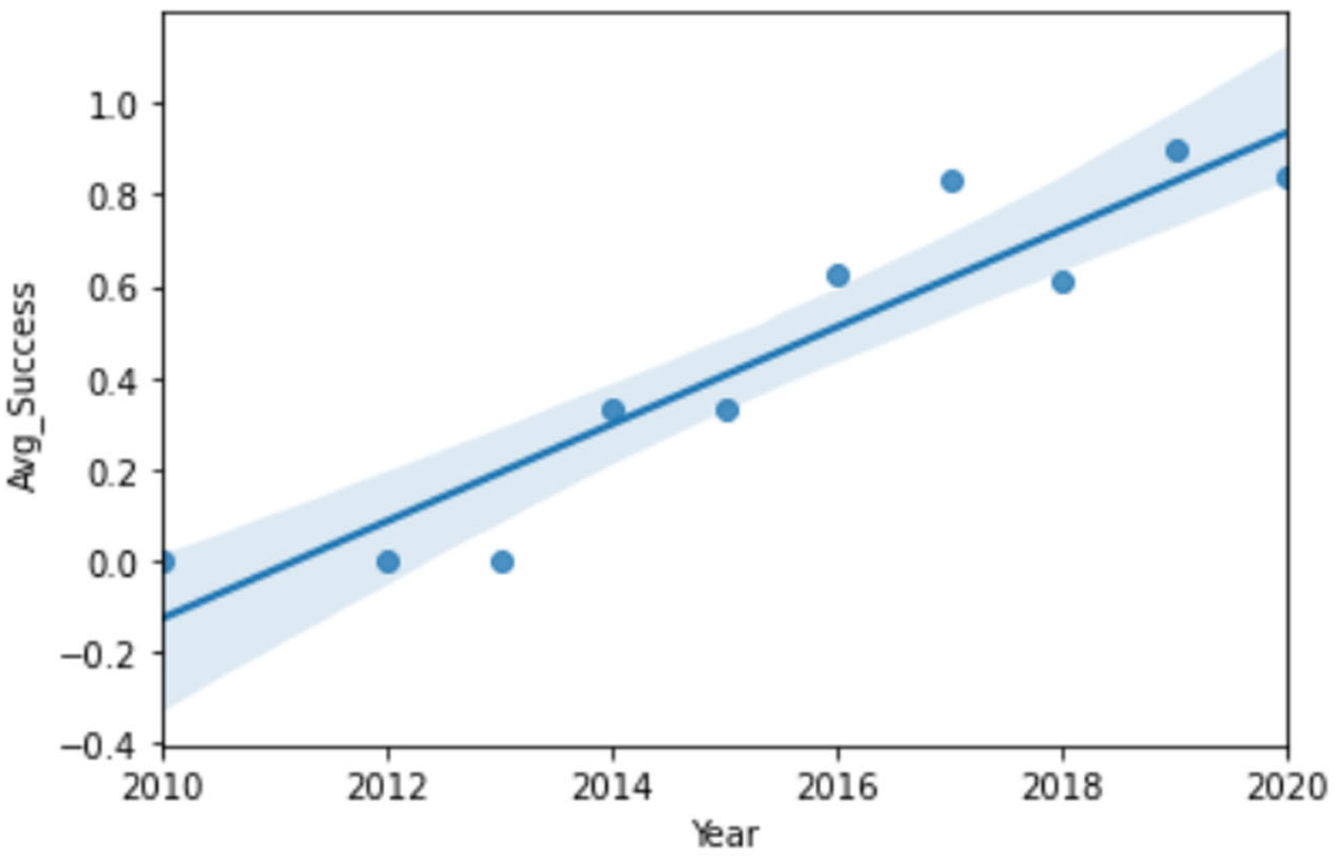
This plot shows how the flight number affects the success rate of each orbit type. The success rate of the LEO orbit appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.

Payload vs. Orbit Type



This plot shows how the payload mass affects the success rate of each orbit type. Missions with payloads under 4,000 Kg in the ISS orbit had a lower success rate than those with greater mass. Payload mass didn't have any effect when in the GTO orbit.

Launch Success Yearly Trend



This line chart plots the trend of the yearly average success rate. We can observe that the success rate started increasing in 2014 and continued growing until 2020.

All Launch Site Names

Task 1

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

```
In [2]: %sql Select Distinct(launch_site) from SPACEXDATASET;

 * ibm_db_sa://znv78891:***@dashdb-txn-sbox-yp-dal09-11.services.dal.bluemix.net:50000/BLUDB
Done.
```

Out[2]:

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

SQL statement to display the unique set of launch site names in the dataset.

Launch Site Names Beginning with ‘CCA’

Task 2

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

In [3]:
```
%%sql Select * from SPACEXDATASET
        Where launch_site like 'CCA%'
        Limit 5
        ;
```

```
 * ibm_db_sa://znv78891:***@dashdb-txn-sbox-yp-dal09-11.services.dal.bluemix.net:50000/BLUDB
Done.
```

Out[3]:

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

SQL statement to display the first five rows that have launch site names beginning with ‘CCA’.

Total Payload Mass

Task 3

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

In [4]:
```
%%sql Select sum(payload_mass__kg_) as nasa_crs_payload
        from SPACEXDATASET
        Where customer = 'NASA (CRS)'
        ;
```

```
 * ibm_db_sa://znv78891:***@dashdb-txn-sbox-yp-dal09-11.services.dal.bluemix.net:50000/BLUDB
Done.
```

Out[4]:

nasa_crs_payload
45596

SQL statement to display the total payload mass carried by boosters launched by NASA (CRS). Answer: 45,596 Kg.

Average Payload Mass by F9 v1.1

Task 4

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

In [5]:
```
%%sql Select avg(payload_mass__kg_) as f9_v11_avg_payload
        from SPACEXDATASET
        Where booster_version = 'F9 v1.1'
        ;
```

```
 * ibm_db_sa://znv78891:***@dashdb-txn-sbox-yp-dal09-11.services.dal.bluemix.net:50000/BLUDB
Done.
```

Out[5]:

f9_v11_avg_payload
2928.400000

SQL statement to display the average payload mass carried by booster version F9 v1.1. Answer: 2,929.40 Kg.

First Successful Ground Landing Date

Task 5

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

Hint:Use min function

In [6]:
```
%%sql Select min(date) as first_ground_pad_landing
        from SPACEXDATASET
        Where landing__outcome = 'Success (ground pad)'
        ;
```

```
 * ibm_db_sa://znv78891:***@dashdb-txn-sbox-yp-dal09-11.services.dal.bluemix.net:50000/BLUDB
Done.
```

Out[6]:

first_ground_pad_landing
2015-12-22

SQL statement to display the date of the first successful ground pad landing.
Answer: Dec 22, 2015.

Successful Drone Ship Landing with Payload Between 4000 and 6000

Task 6

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

In [7]:
```
%%sql Select booster_version as good_boosters_on_drone_ships
        from SPACEXDATASET
        Where landing__outcome = 'Success (drone ship)'
            and payload_mass__kg_ between 4000 and 6000
        ;
```

```
 * ibm_db_sa://znv78891:***@dashdb-txn-sbox-yp-dal09-11.services.dal.bluemix.net:50000/BLUDB
Done.
```

Out[7]:

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

SQL statement to display the names of booster versions with successful landings in drone ships, and payload masses between 4,000 and 6,000 Kg.

Total Number of Successful and Failed Mission Outcomes

Task 7

List the total number of successful and failure mission outcomes

In [8]:
```
%%sql Select mission_outcome, count(*) as count
        from SPACEXDATASET
        Group by mission_outcome
        Order by count desc
        ;
```

```
 * ibm_db_sa://znv78891:***@dashdb-txn-sbox-yp-dal09-11.services.dal.bluemix.net:50000/BLUDB
Done.
```

Out[8]:

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

SQL statement to display the total number of successful and failed mission outcomes.

Boosters That Carried Maximum Payload

Task 8

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

In [9]:
```
%%sql Select booster_version as max_payload_boosters
       from SPACEXDATASET
       Where payload_mass__kg_ = (Select max(payload_mass__kg_) from SPACEXDATASET)
       ;
```

```
 * ibm_db_sa://znv78891:***@dashdb-txn-sbox-yp-dal09-11.services.dal.bluemix.net:50000/BLUDB
Done.
```

Out[9]:

max_payload_boosters
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

SQL statement to display the names of the booster versions that carried the maximum payload mass.

2015 Launch Records

Task 9

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

In [10]:
```
%%sql Select landing__outcome, booster_version, launch_site
        from SPACEXDATASET
        Where landing__outcome = 'Failure (drone ship)'
            and year(date) = '2015'
        ;
```

```
 * ibm_db_sa://znv78891:***@dashdb-txn-sbox-yp-dal09-11.services.dal.bluemix.net:50000/BLUDB
Done.
```

Out[10]:

landing__outcome	booster_version	launch_site
Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

SQL statement to display the failed landings on drone ships of the year 2015. Booster version and launch site names are also displayed.

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

In [11]:
```
%%sql Select landing__outcome, count(*) as count
        from SPACEXDATASET
        Where date between '2010-06-04' and '2017-03-20'
        Group by landing__outcome
        Order by count desc
        ;
```

```
 * ibm_db_sa://znv78891:***@dashdb-txn-sbox-yp-dal09-11.services.dal.bluemix.net:50000/BLUDB
Done.
```

Out[11]:

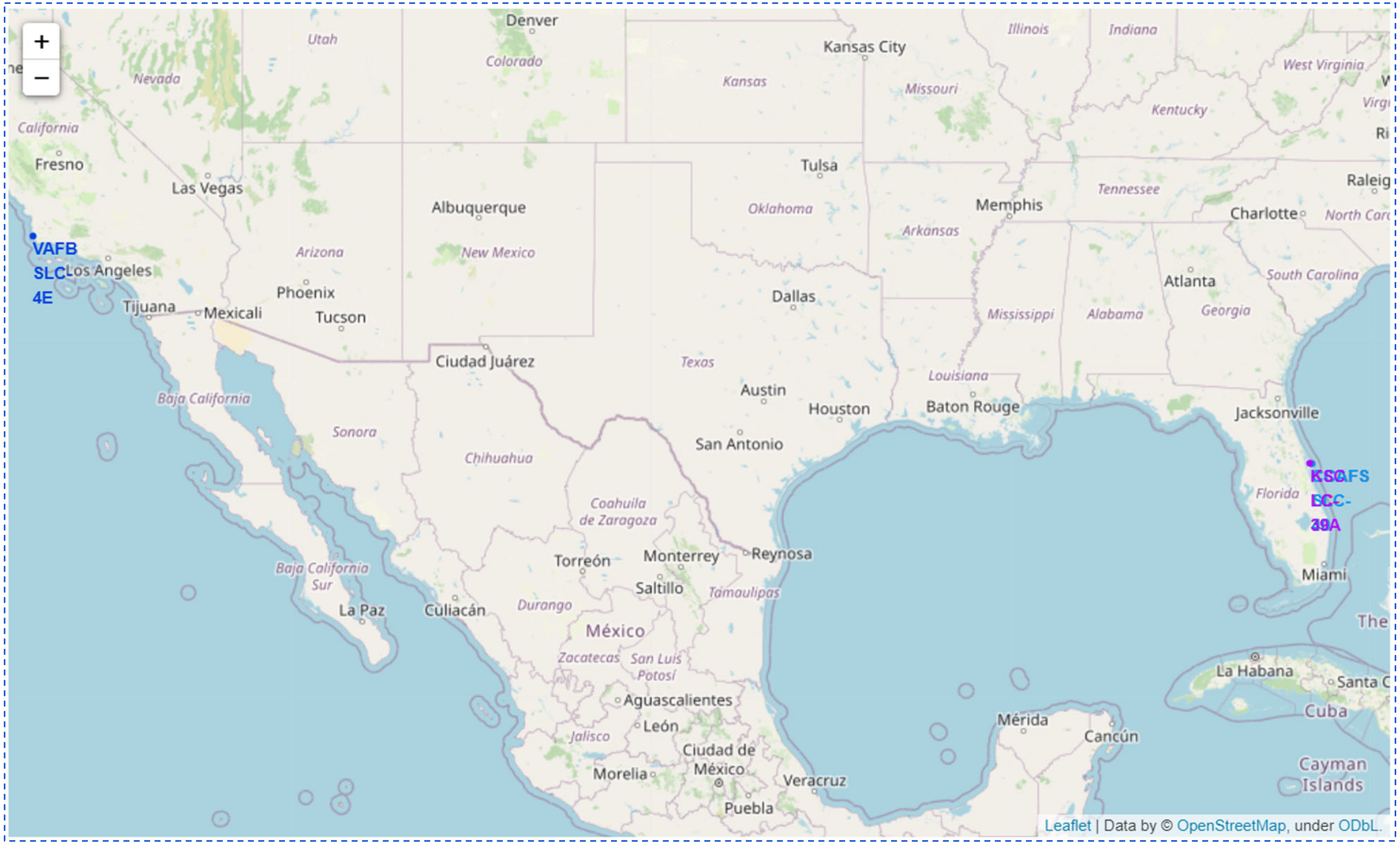
landing__outcome	COUNT
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

SQL statement to display the ranking of landing outcomes from launches carried out between June 4, 2010 and March 20, 2017.

Section 4

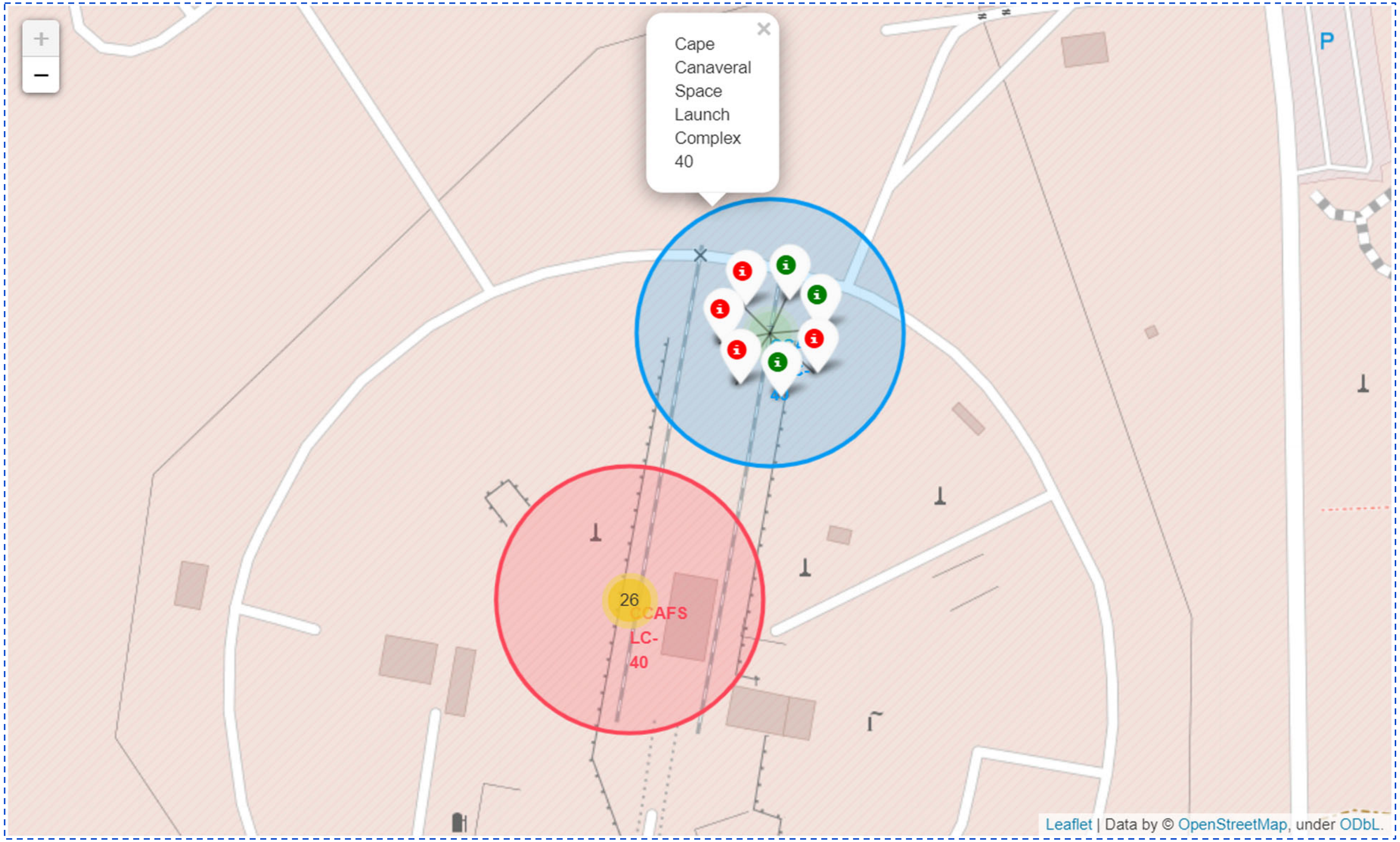
Launch Sites Proximities Analysis

Global map showing location markers from all launch sites



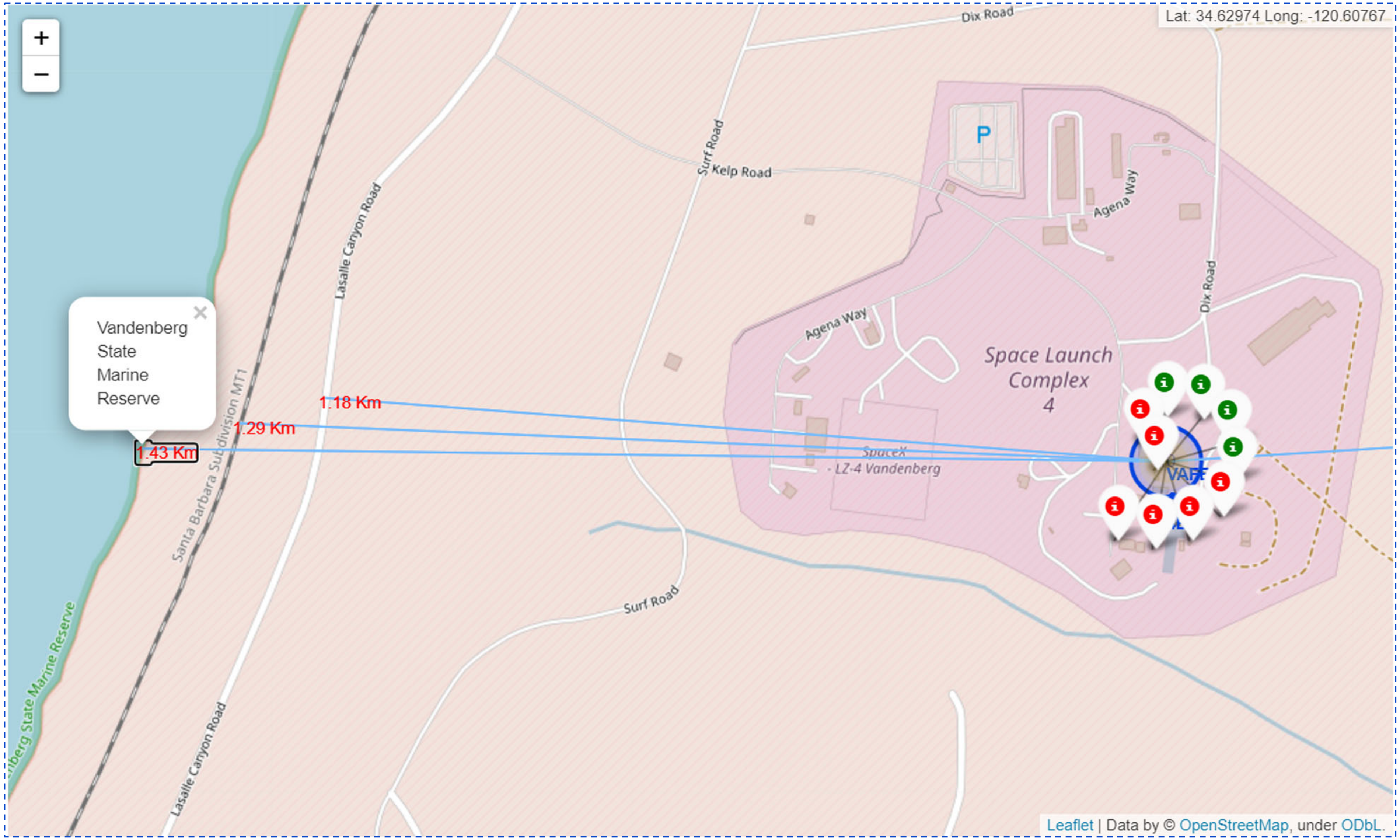
Global map showing all launch sites' location markers, each with a different color.

Local area map showing color-labeled launch outcomes



This map shows the color-labeled launch outcomes of site CCAFS SLC-40, in Cape Canaveral, Florida.

Local map showing nearest road, railway and coastline

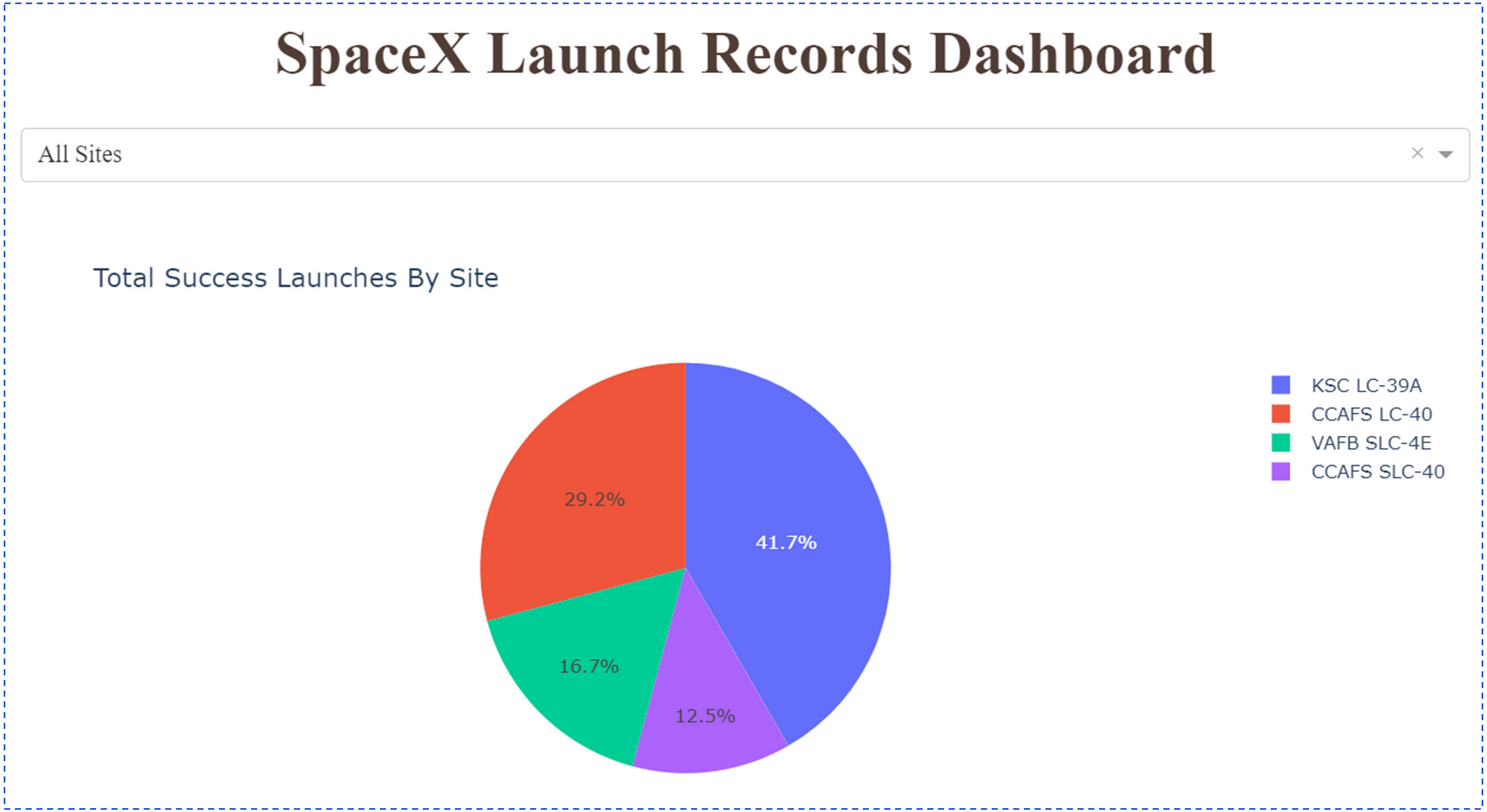


This map shows a major railway system, the Santa Barbara Subdivision MT1, at 1.29 Km from the Vandenberg Space Launch Complex 4E, in California. The distance to the coastline is about 1½ Km, and the LaSalle Canyon Road is just over a kilometer away.

Section 5

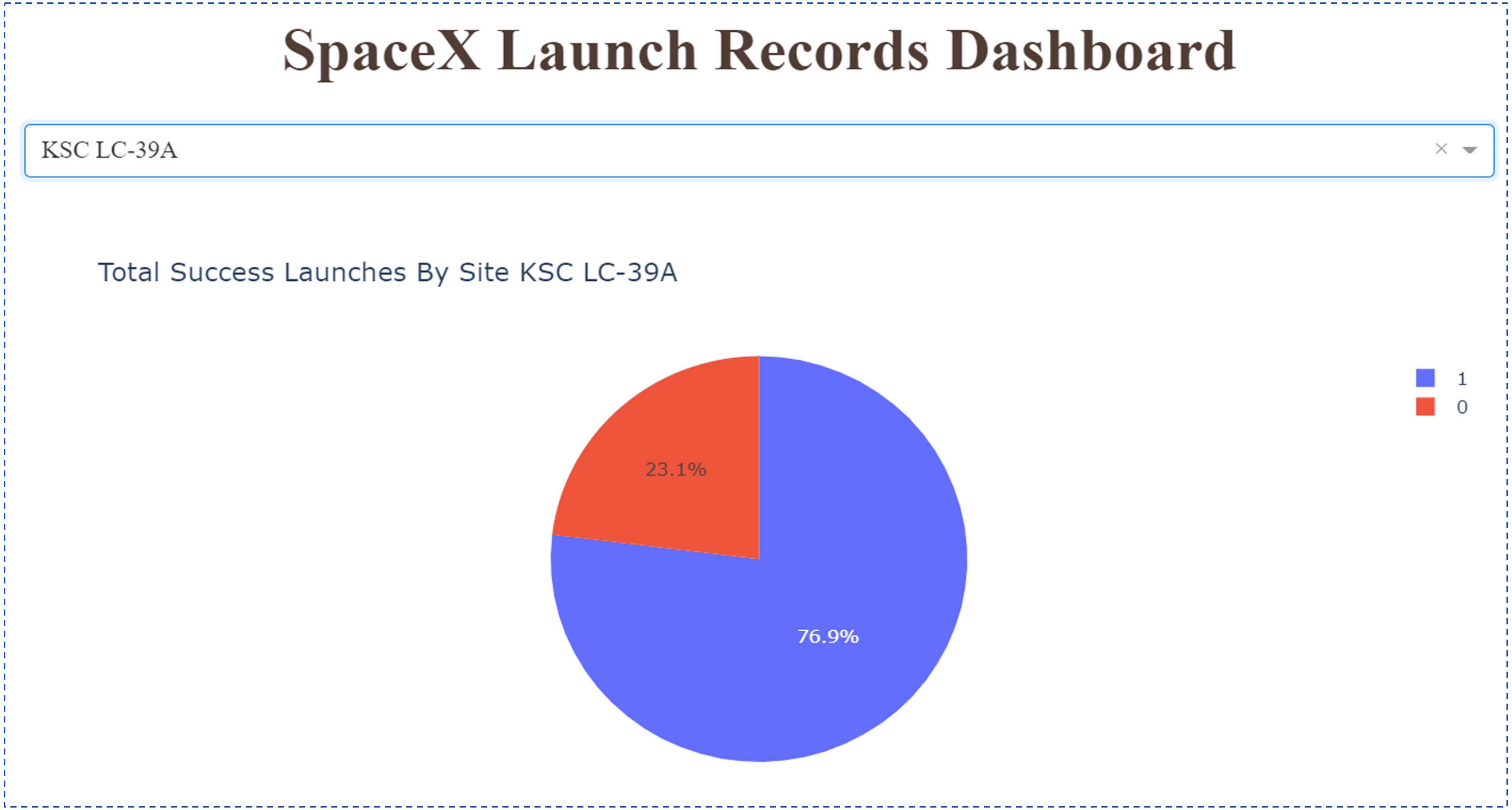
Build a Dashboard with Plotly Dash

Pie chart of successful launches for each site



This pie chart displays the percentage of successful launches each site has relative to each other, showing site KSC LC-39A (Kennedy Space Center) with the higher share of 41.7%.

Pie chart with success rate of most successful launch site



This pie chart displays a success rate of 76.9% from the most successful launch site, KSC LC-39A, at Kennedy Space Center, Florida.

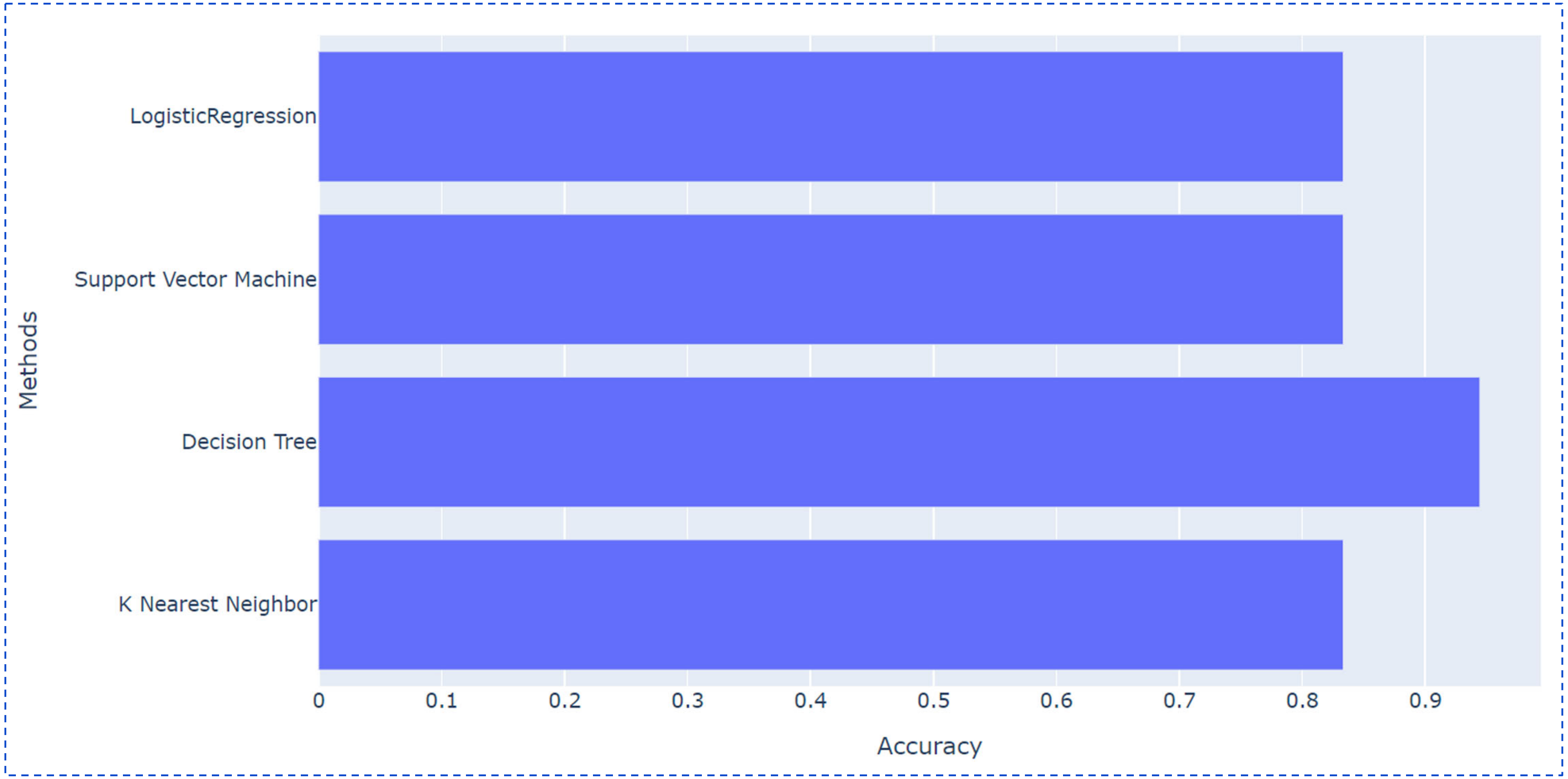
Launch outcomes of different payloads selected by range slider

This scatter plot displays the success rate of the booster version categories that fall within the range of payload masses selected in the slider (2,500 to 7,500 Kg).

Section 6

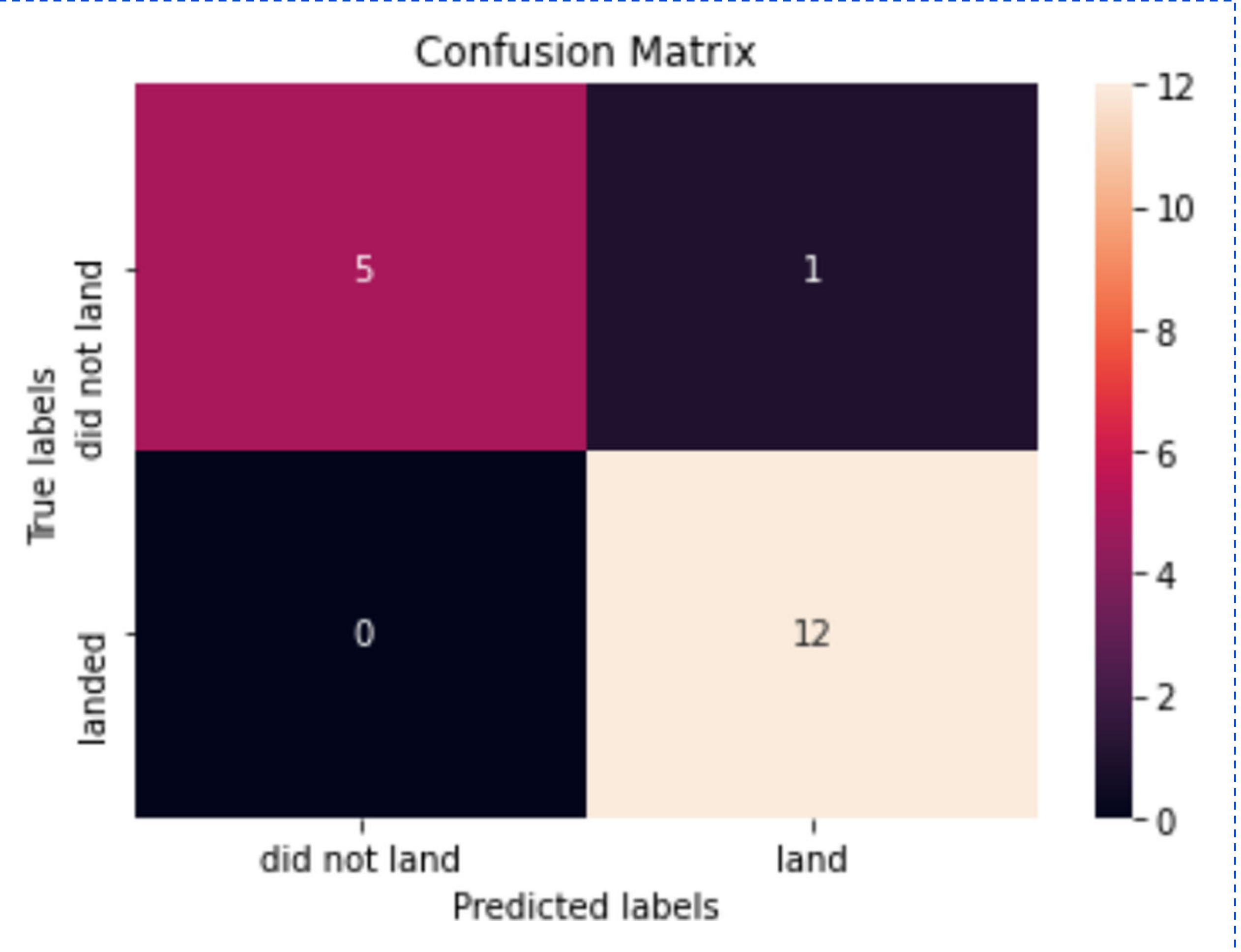
Predictive Analysis (Classification)

Classification Accuracy



The Decision Tree model was found to have the highest accuracy of 94.44%.

Confusion Matrix



The confusion matrix of the best performing model, the Decision Tree method, shows only one false positive out of the 13 predicted landings.

Conclusions

- The Kennedy Space Center was found to have the highest rate of successful launches, and should be the overall preferred launch site.
- We can use the Decision Tree model to predict, with 94% accuracy, if SpaceX will land successfully the first stage.

Appendix

All TODOs in one cell

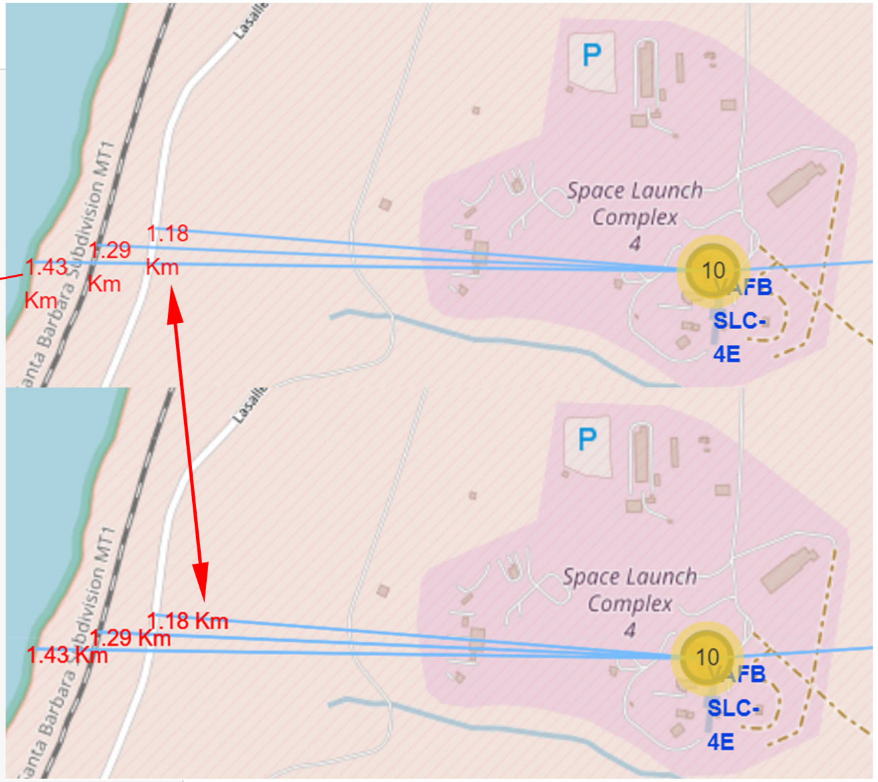
```python
staBarbRail_GPS = [34.63331,-120.62487]
coastLine_GPS   = [34.63300,-120.62634]
laSalleRoad_GPS = [34.63363,-120.62355]
lompoc_GPS      = [34.63929,-120.48441]

labels = ['Santa Barbara Subdivision MT1',
          'Vandenberg State Marine Reserve',
          'LaSalle Canyon Road',
          'Lompoc, California'
         ]

index = 0
for marker in [staBarbRail_GPS, coastLine_GPS, laSalleRoad_GPS, lompoc_GPS]:
    lat1,lon1 = vandenbergSpace_GPS[:]
    lat2,lon2 = marker[:]
    distanceLabel = "{} Km".format(round(calculate_distance(lat1, lon1, lat2, lon2),2))
    folium.Marker(
        location=[lat2, lon2],
        icon=folium.DivIcon(html=f"""<div style="font-family: arial; color: red">{distanceLabel}</div>"""),
        popup=labels[index],
    ).add_to(site_map)

    folium.PolyLine([vandenbergSpace_GPS,marker], color="#75bbfd", weight=1.5, opacity=1).add_to(site_map)
    index += 1

site_map
```



This Python code snippet was used to produce and label four lines to landmarks near the Vandenberg Space Launch Complex 4E, in California. The coordinates of the site (variable vandenbergSpace_GPS) had been loaded in a previous cell.

To avoid the line feed produced when placing a <u>blank</u> space between the number and the measuring unit (Km), I used a different blank character, ASCII 160 (Alt-255 will also work), instead of the regular ASCII 32.

Thank you!