IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Shaniece Gooden
1/14/25

Outline

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

Executive Summary

- Summary of methodologies
  - Data about Falcon 9 first-stage landings was collected using REST API and web scraping. The data was processed using SQL, and exploratory analysis created visualizations to help determine what factors affected the success of landings. Folium was used to create maps to determine launch site proximity, and a dashboard was created to better visualize launch records using Plotly. Finally machine learning was used to predict if the first stage of Falcon 9 will land successfully, and to determine the best classification method for the data.
- Summary of all results
  - Once the data was successfully collected and formatted into a Pandas dataframe the average of the Payload Mass column was used to fill in missing values. Missing values in the Landing Pad column were retained to signify when a landing pad was not used. Using SQL I determined how many successful and failed missions there were based on flight number, payload mass, and booster version. Exploratory analysis revealed that these factors as well as flight number were strong predictors of successful landings. Using predictive analysis it was found that any classification method (SVM, Decision Tree, k-Nearest Neighbors, and Logistic Regression) could be used to predict successful landing, but a Decision Tree model would likely be best.

Introduction

- Project background and context
  - SpaceX has gained worldwide attention for a series of historic rocket launches. It is the only private company ever to return a spacecraft from low-earth orbit. SpaceX advertises Falcon 9 launches with a cost of $62 million, whereas other companies cost at least $165 million. Much of this savings is because SpaceX can reuse the first stage. If it can be determined if the first stage will land, then the cost of the launch can be calculated. This information can be used by competitors to SpaceX for bids on rocket launches.
- Problems you want to find answers
  - What is the success rate of Falcon 9 first-stage landings?
  - What factors affect the success of Falcon 9 landings?
  - Can we predict a combination of factors that will lead to a successful landing?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Rocket launch data was collected from SpaceX API, and through web scraping the data was converted into a Pandas dataframe.
- Perform data wrangling
  - Data was processed by creating a Class column to classify successful and failed landings. One-hot encoding was applied to all categorical features.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - The data was standardize and then split into testing and training sets. GridSearch was used on these sets to tune the hyperparameters and determine the most accurate classification model.

Data Collection

- The data was collected from the SpaceX API and then decoded using JSON functions. From the JSON response, the necessary data was extracted, filtering for the Falcon 9 rocket.
- The columns Payload Mass and Landing Pad have some missing values. The missing values for Landing Pad will be retained to signify when a landing pad was not used. For the missing values in Payload Mass I replaced them with the average of all Falcon 9 payload masses.
- Data from Wikipedia was collected through web scraping function BeautifulSoup on the JSON response. This allowed the data to be converted to HTML format and parsed in that format.
- The parsed data was converted into a Pandas dataframe.

Data Collection – SpaceX API

- https://github.com/ShanieceG/IBM-Capstone/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

```python
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

```python
response=requests.get(static_json_url)
```

```python
response.status_code
```

```
200
```

Now we decode the response content as a Json using `.json()` and turn it into a Pandas dataframe using `.json_normalize()`

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

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

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

Check the content of the response

```python
print(response.content)
```


```
b'[{"fairings":{"reused":false,"recovery_attempt":false,"recovered":false,"ships":[]},"lin
mages2.imgbox.com/5b/02/QcxHUb5V_o.png"},"reddit":{"campaign":null,"launch":null,"media":n
s://www.youtube.com/watch?v=0a_00nJ_Y88","youtube_id":"0a_00nJ_Y88","article":"https://www
n.wikipedia.org/wiki/DemoSat"},"static_fire_date_utc":"2006-03-17T00:00:00.000Z","static_f
ess":false,"failures":[{"time":33,"altitude":null,"reason":"merlin engine failure"}],"deta
[],"payloads":["5eb0e4b5b6c3bb0006eeb1e1"],"launchpad":"5e9e4502f5090995de566f86","flight_
0,"date_local":"2006-03-25T10:30:00+12:00","date_precision":"hour","upcoming":false,"cores
lse,"landing_attempt":false,"landing_success":null,"landing_type":null,"landpad":null}],"a
airings":{"reused":false,"recovery_attempt":false,"recovered":false,"ships":[]},"links":{"
2.imgbox.com/80/a2/bkWotCIS_o.png"},"reddit":{"campaign":null,"launch":null,"media":null,"
w.youtube.com/watch?v=Lk4zQ2wP-Nc","youtube_id":"Lk4zQ2wP-Nc","article":"https://www.space
a.org/wiki/DemoSat"},"static_fire_date_utc":null,"static_fire_date_unix":null,"net":false,
titude":289,"reason":"harmonic oscillation leading to premature engine shutdown"}],"detail
remature engine shutdown at T+7 min 30 s, Failed to reach orbit, Failed to recover first s
d":"5e9e4502f5090995de566f86","flight_number":2,"name":"DemoSat","date_utc":"2007-03-21T01
on":"hour","upcoming":false,"cores":[{"core":"5e9e289ef35918416a3b2624","flight":1,"gridfi
ing_type":null,"landpad":null}],"auto_update":true,"tbd":false,"launch_library_id":null,"i
ered":false,"ships":[]},"links":{"patch":{"small":"https://images2.imgbox.com/6c/cb/na1tzh
null,"launch":null,"media":null,"recovery":null},"flickr":{"small":[],"original":[]},"pres
60","article":"http://www.spacex.com/news/2013/02/11/falcon-1-flight-3-mission-summary","w
null,"static_fire_date_unix":null,"net":false,"window":0,"rocket":"5e9d0d95eda69955f709d1e
led to collision between stage 1 and stage 2"}],"details":"Residual stage 1 thrust led to 
0e4b6b6c3bb0006eeb1e3","5eb0e4b6b6c3bb0006eeb1e4"],"launchpad":"5e9e4502f5090995de566f86",
```

Data Collection - Scraping

- https://github.com/ShanieceG/IBM-Capstone/blob/main/jupyter-labs-webscraping.ipynb

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

Next, request the HTML page from the above URL and get a `response` object

TASK 1: Request the Falcon9 Launch Wiki page from its URL

First, let's perform an HTTP GET method to request the Falcon9 Launch HTML page, as an HTTP response.

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

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

```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
sour = BeautifulSoup(response.content, "html.parser")
```

```
extracted_row = 0
#Extract each table
for table_number,table in enumerate(sour.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
            launch_dict['Flight No.'].append(flight_number)
            #print(flight_number)
            datatimelist=date_time(row[0])

            # Date value
            date = datatimelist[0].strip(',')
            launch_dict['Date'].append(date)
            #print(date)

            # Time value
            time = datatimelist[1]
            launch_dict['Time'].append(time)
            #print(time)

            # Booster version
            # TODO: Append the bv into launch_dict with key `Version Booster`
            bv=booster_version(row[1])
            if not(bv):
                bv=row[1].a.string
            launch_dict['Version Booster'].append(bv)
```

Data Wrangling

- First I ensured that the only column with missing values was the Landing Pad column. Then I confirmed the data types of each column.
- The number of different outcomes were calculated (True Ocean, False Ocean, etc.). The different outcomes were organized and then split into two different variables, "landing outcomes" and "bad outcomes". The bad outcomes consists of all failed outcomes and landing outcomes consists of all successful outcomes. These two variables were used to create a Class column in the dataframe, where 0 is a bad/failed outcome and 1 is a landing/successful outcome.
- https://github.com/ShanieceG/IBM-Capstone/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

- Explored the following relationships with the corresponding observations:
  - Launch site vs. Flight number: Launch site CCAFS SLC 40 had more successful flights before flight number 20
  - Payload mass vs. Launch site: Site VAFG-SLC had no rockets launched with a payload mass greater than 10,000kg
  - Flight number vs. Orbit type: LEO orbit had more successful flights with a higher number of flights
  - Payload mass vs. Orbit type: Heavy payloads had more successful landings in orbits Polar, LEO, and ISS
  - Success rate for each orbit type: Orbits ES-L1, GEO, HEO, and SSO have the highest success rates.
  - Annual success trend: Successful landings increased after 2013 up until 2020.
- https://github.com/ShanieceG/IBM-Capstone/blob/main/edadataviz.ipynb

EDA with SQL

- Performed SQL queries to:
  - List all launch sites
  - Calculate the average payload mass for booster F9 v1.1
  - Find the date of the first successful landing on ground pad
  - List the total of successful and failed missions
  - Determine which boosters carried the heaviest payloads
- https://github.com/ShanieceG/IBM-Capstone/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- Each launch site was marked on a Folium map with its name and a circle marker to better visualize the locations of the launch sites
- Marker clusters were added to each launch site to show all the successful (green) and failed (red) launches to easily visualize the success rate of each launch site
- Added distance lines to each launch site to determine their proximity to other geographical markers such as railways, highways, coast lines, and major cities. Thus if a company wanted to build a launch site they would have a general idea of how far they must be from these geographical markers.
- https://github.com/ShanieceG/IBM-Capstone/blob/main/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

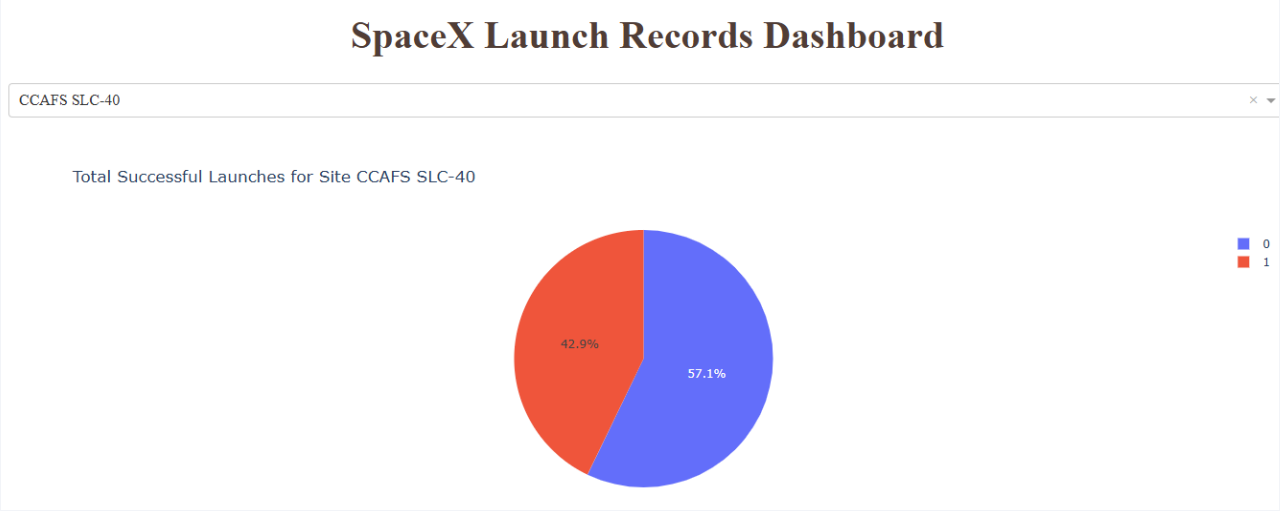
- Added a pie chart to the dashboard to show the total number of successful launches for each site. If a specific site is selected then it will show the ratio of successful and failed launches.
- A scatter chart was added to show the correlation between payload and launch success for each launch site. A slider was added to adjust the payload range and easily visualize how many successful or failed launches occurred for each site.
- https://github.com/ShanieceG/IBM-Capstone-Dashboard/blob/main/spacex_dash_app%20(1).py

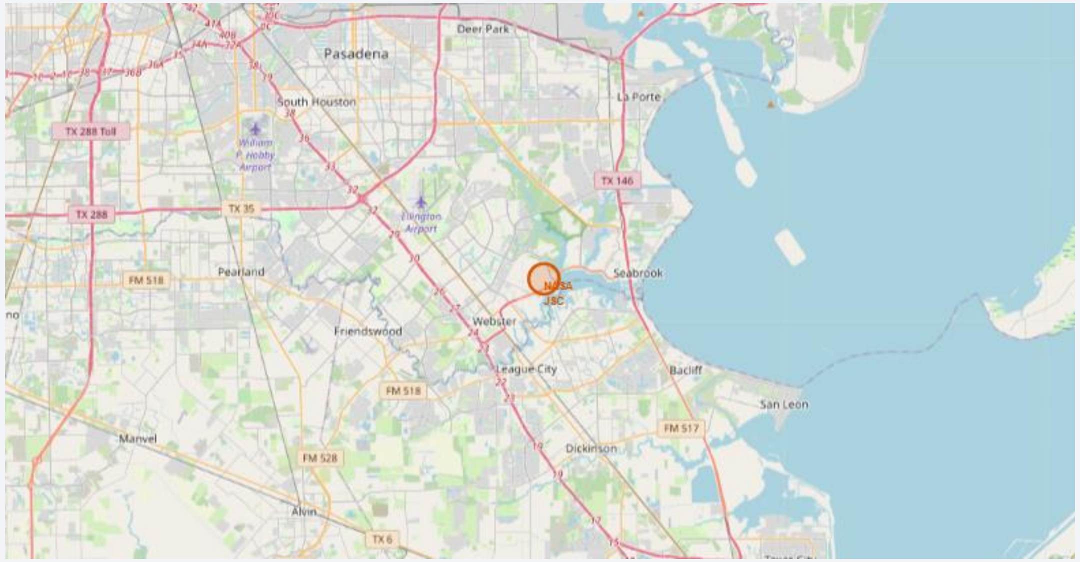
Predictive Analysis (Classification)

- Using the Pandas dataframe of the SpaceX data, set the Class column to be the independent variable, Y, and standardize the rest of the data into variable X. Both Y & X were spilt into their own training and testing sets.
- Using GridSearchCV to determine the best hyperparameters for four classification methods:
    - Logistic Regression
    - Decision Tree
    - K-Nearest Neighbors
    - Support Vector Machine
- For each method we calculated the accuracy of the model on the test data and verified with a confusion matrix. All models had an accuracy of 83.3%. However the Decision Tree has a accuracy of 87.5% for the training data.
- https://github.com/ShanieceG/IBM-Capstone-Predictive-Analysis/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5%20(1).ipynb

Results

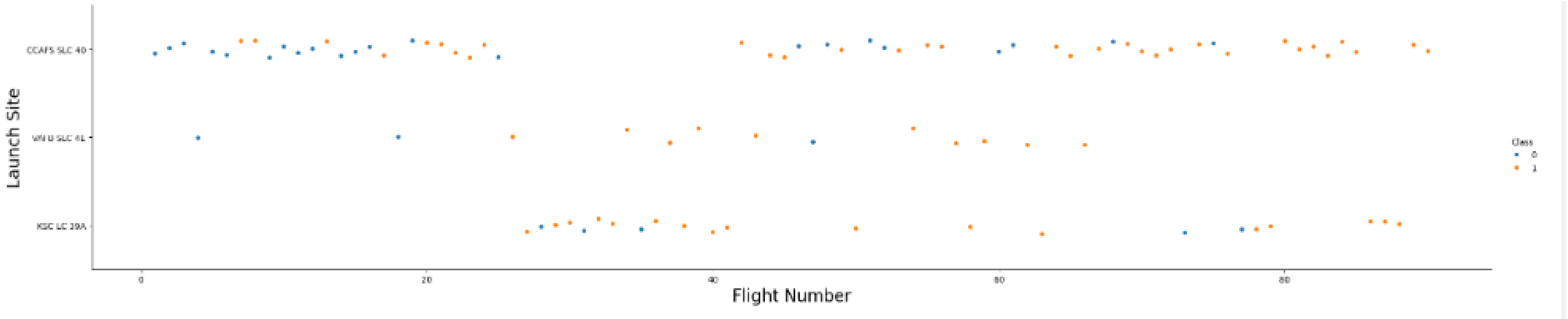
- Exploratory data analysis revealed that factors that determine a successful launch are payload mass and flight number.
- Interactive maps allow better visualization of launch sites and their success rates
- Predictive analysis revealed that any of the 4 classification methods would suffice as a model for launches.





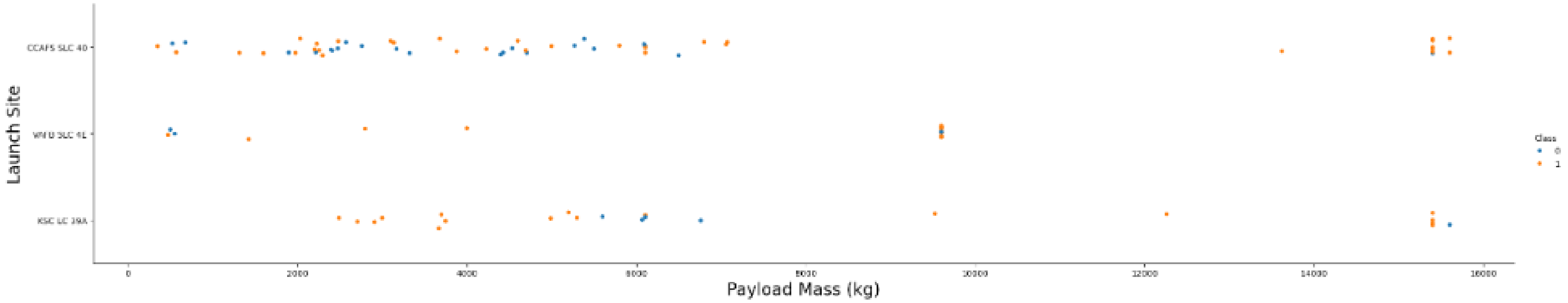
Section 2

Insights drawn from EDA



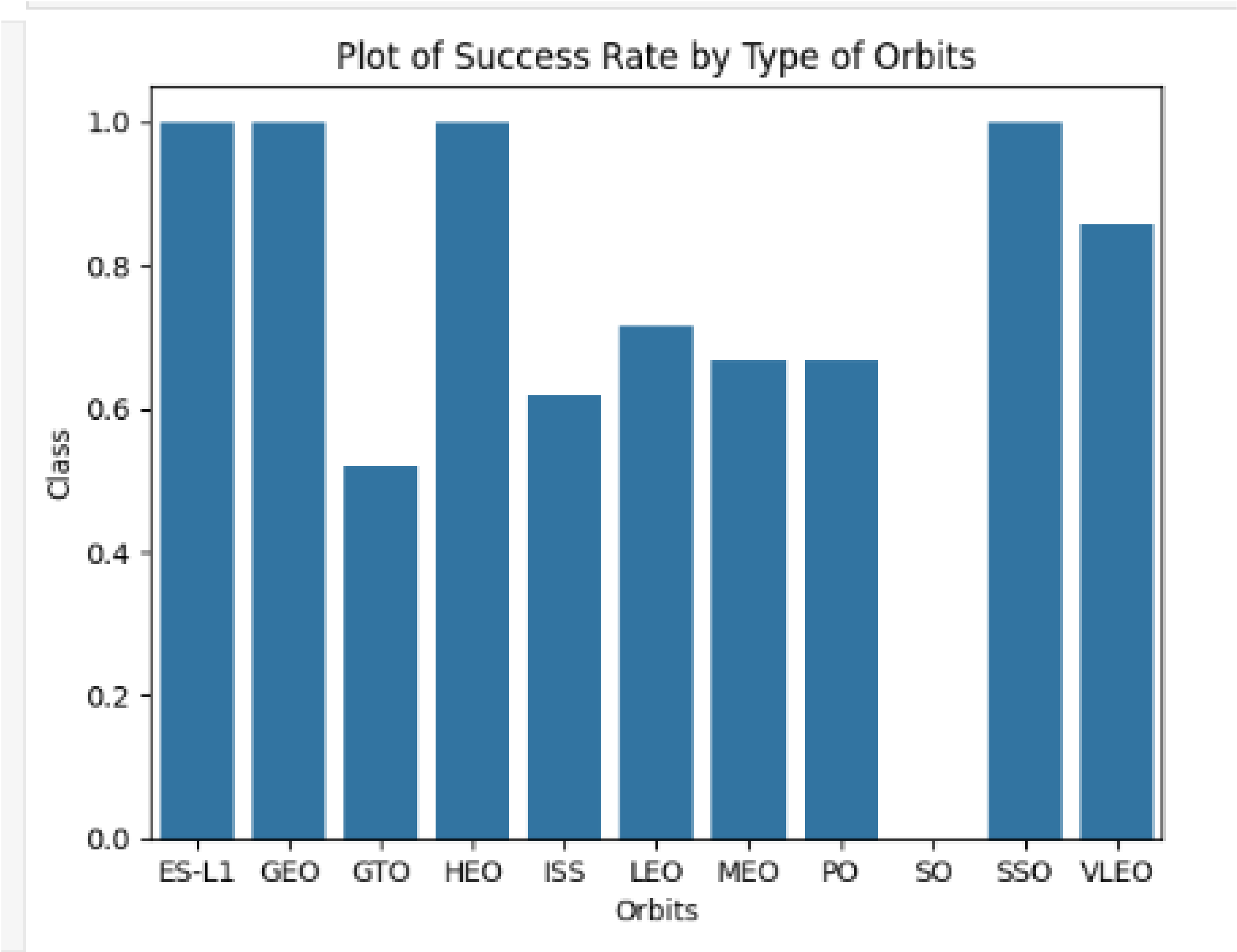
Flight Number vs. Launch Site

- The launch sites tend to have more successful landings in later flights rather than earlier ones.



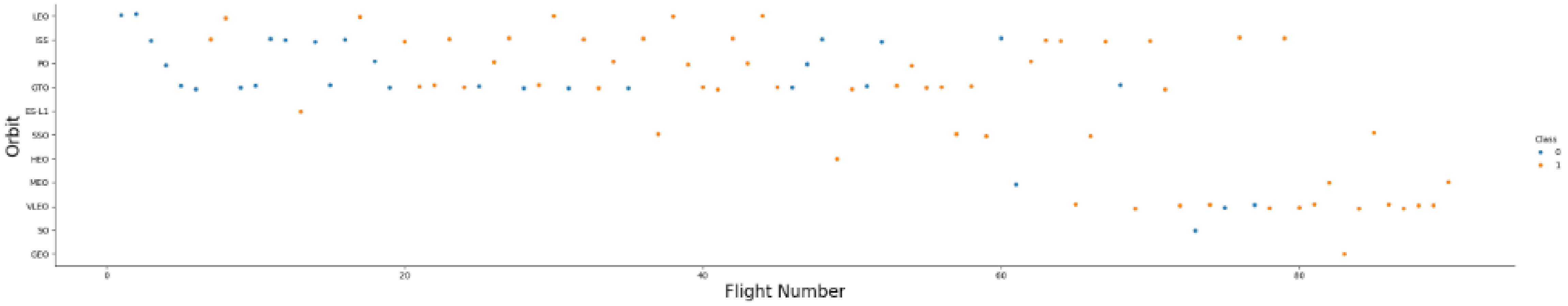
Payload vs. Launch Site

- Site VAFB-SLC did not launch any rockets with a payload over 10,000kg.
- Landings had more success with lighter payloads.



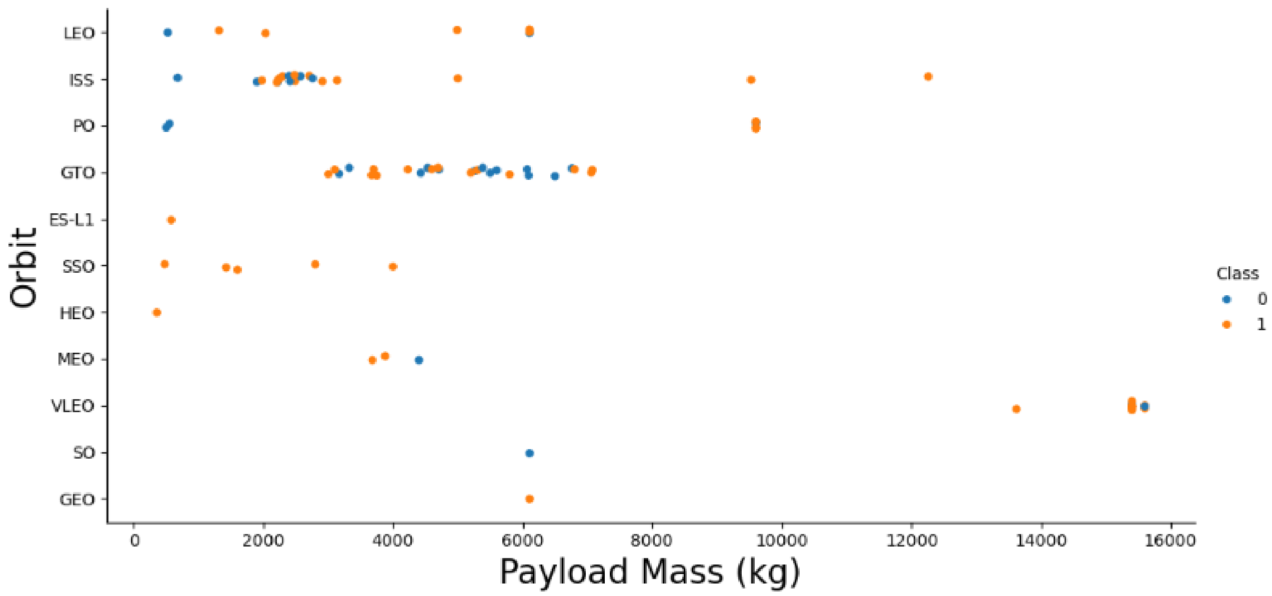
Success Rate vs. Orbit Type

- Orbits ES-L1, GEO, HEO, and SSO have the highest success rates
- Orbits around 35,786km relative to the Earth tend to be more successful



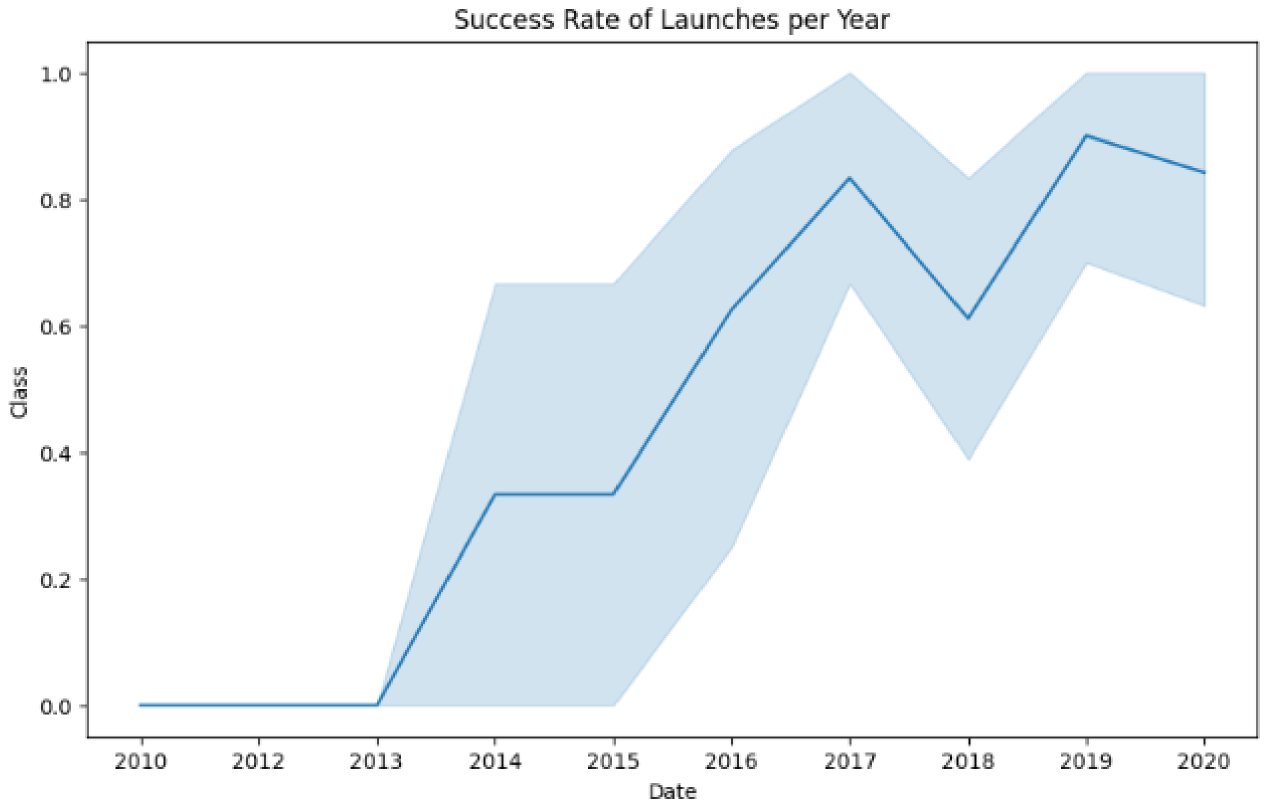
Flight Number vs. Orbit Type

- LEO orbit success seems to be related to flight number, but GTO orbit does not.
- Flight number with orbit are not good indicators of a successful landing.



Payload vs. Orbit Type

- Heavy payloads had more successful landings in PO, LEO, and ISS orbits.
- GTO still has not correlation for successful and failed landings based on payload.
- Payload and orbit may be a better indicator of successful landings compared to flight number, but not enough confidence.



Launch Success Yearly Trend

- Successful landings increased after 2013 and up until 2017. As well as between 2018 and 2019.

All Launch Site Names

- Query to find all distinct launch sites for Falcon 9

```
%%sql SELECT DISTINCT Launch_Site
    FROM SPACEX
```

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

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

Launch Site Names Begin with 'CCA'

- Query for all launch site information for any site beginning with "CCA".

```
%%sql SELECT *
    FROM SPACEX
    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
2010-06-04	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	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	0:35:00	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 SPACEX
    WHERE customer = "NASA (CRS)"
```

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

total_payload
45596

- Calculate the total payload carried by boosters from NASA

Average Payload Mass by F9 v1.1

```
%%sql
SELECT AVG(payload_mass__kg_) AS avg_payload_mass
FROM SPACEX
WHERE booster_version='F9 v1.1'
```

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

avg_payload_mass
2928.4

- Calculate the average payload mass carried by booster version F9 v1.1

First Successful Ground Landing Date

```
%%sql
SELECT MIN(date)
FROM SPACEX
WHERE landing_outcome="Success (ground pad)"
```

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

MIN(date)
2015-12-22

- Query of date(s) of the first successful landing on ground pad

Successful Drone Ship Landing with Payload between 4000 and 6000

- Query for names of boosters which have successfully landed on drone ship and had payload mass greater than 4,000 kg but less than 6,000 kg.
- Only 4 boosters have successfully landed on drone ship with a payload between 4,000 and 6,000 kg

```
%%sql
SELECT booster_version
FROM SPACEX
WHERE landing_outcome = "Success (drone ship)"
    AND payload_mass__kg_>4000
    AND payload_mass__kg_<6000
```

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

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

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes
- 99% of missions are successful

```
%%sql
SELECT COUNT(mission_outcome) AS Successful_missions
FROM SPACEX
WHERE mission_outcome LIKE 'Success%'
```

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

Successful_missions
100

```
%%sql
SELECT COUNT(mission_outcome) AS Failed_missions
FROM SPACEX
WHERE mission_outcome LIKE 'Failure%'
```

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

Failed_missions
1

Boosters Carried Maximum Payload

- Query that lists the names of the booster which have carried the maximum payload mass
- Maximum payload mass is 15,600 kg and 12 boosters have carried this payload mass

```
%%sql
SELECT booster_version, payload_mass__kg_
FROM SPACEX
WHERE payload_mass__kg_ = (
    SELECT MAX(payload_mass__kg_)
    FROM SPACEX)
ORDER BY booster_version
```

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

booster_version	payload_mass__kg_
F9 B5 B1048.4	15600
F9 B5 B1048.5	15600
F9 B5 B1049.4	15600
F9 B5 B1049.5	15600
F9 B5 B1049.7	15600
F9 B5 B1051.3	15600
F9 B5 B1051.4	15600
F9 B5 B1051.6	15600
F9 B5 B1056.4	15600
F9 B5 B1058.3	15600
F9 B5 B1060.2	15600
F9 B5 B1060.3	15600

2015 Launch Records

- Query that lists the failed landing outcomes in drone ship, their booster versions, and launch site names for in year 2015

- Only 2 failed landing outcomes in 2015. Both launched from CCAFS LC-40, one in January and one in April.

```
%%sql
SELECT landing_outcome, booster_version, SUBSTR(date, 6,2) AS month, launch_site
FROM SPACEX
WHERE landing_outcome = "Failure (drone ship)" AND SUBSTR(date,0,5)='2015'
```

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

landing_outcome	booster_version	month	launch_site
Failure (drone ship)	F9 v1.1 B1012	01	CCAFS LC-40
Failure (drone ship)	F9 v1.1 B1015	04	CCAFS LC-40

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

- Query that ranks 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
- The landing outcome with the highest count is "No attempt".

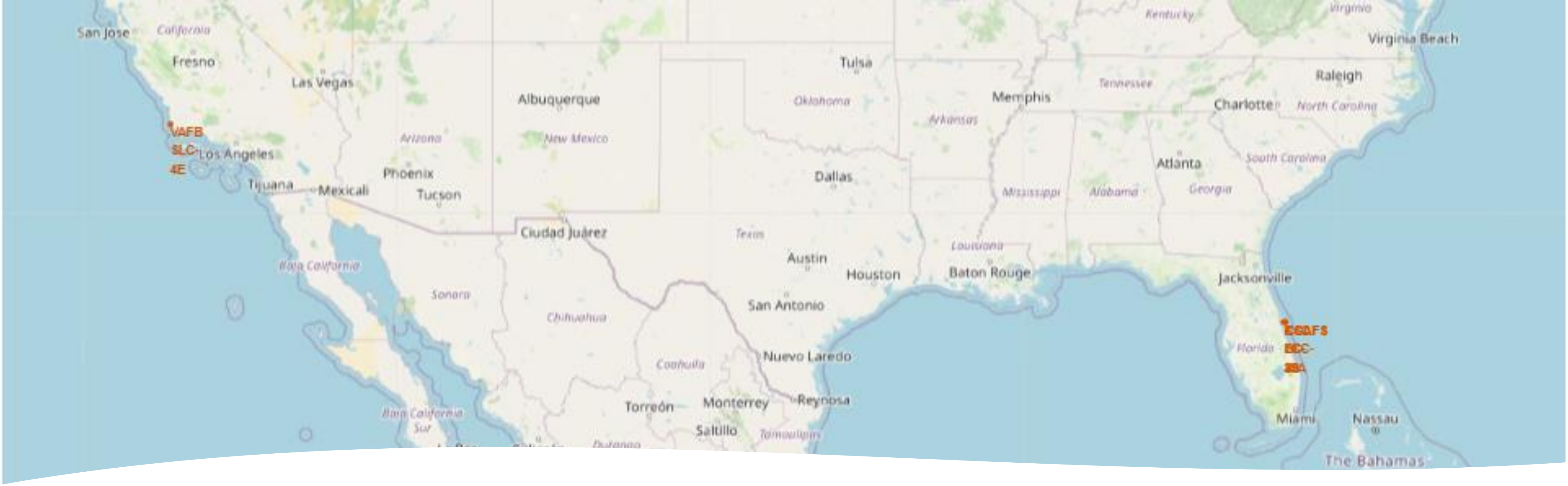
```
%%sql
SELECT landing_outcome, COUNT(landing_outcome) AS count_outcome
FROM SPACEX
WHERE date BETWEEN "2010-06-04" AND "2017-03-20"
GROUP BY landing_outcome
ORDER BY count_outcome DESC
```

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

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

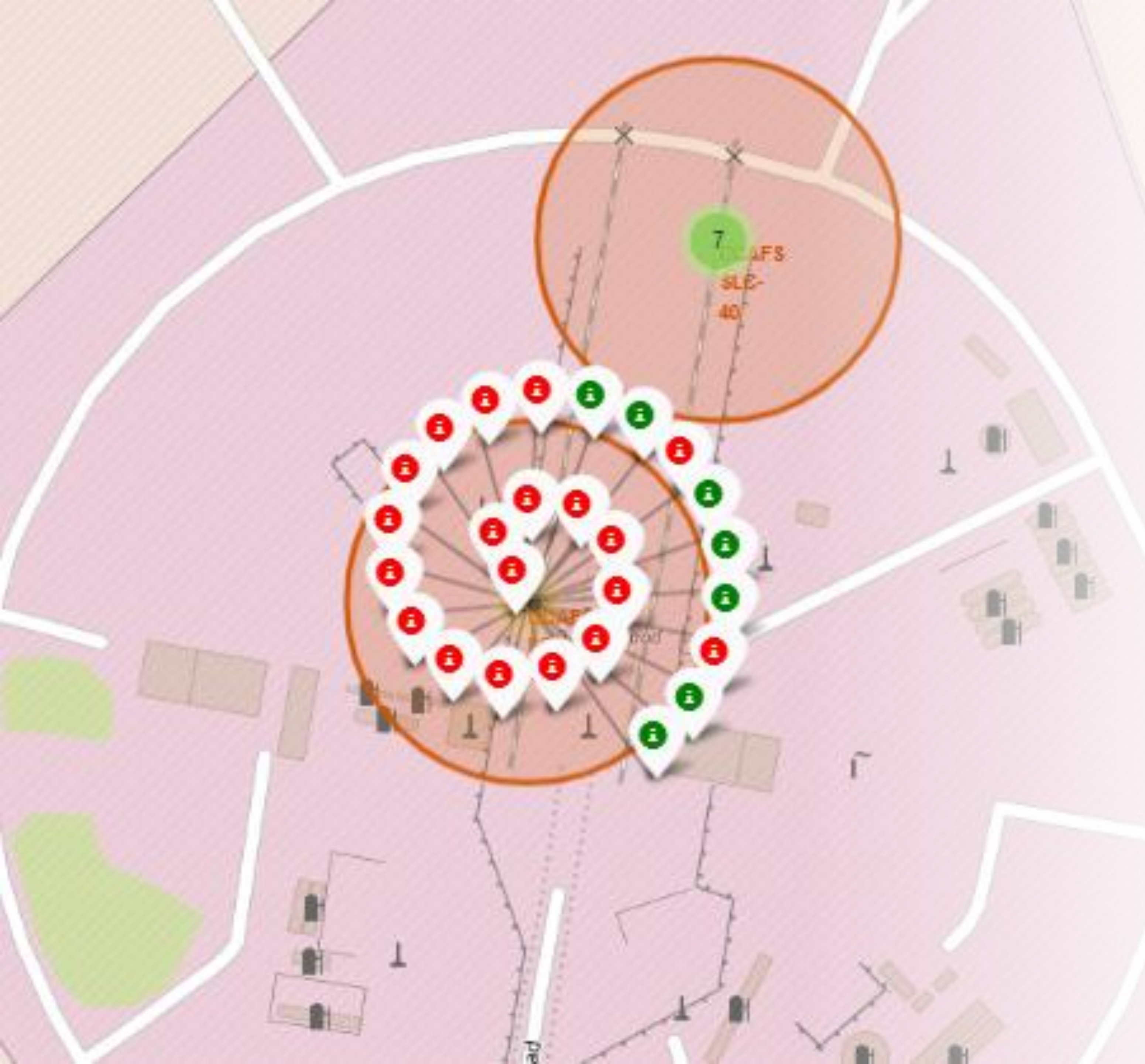
Section 3

Launch Sites Proximities Analysis



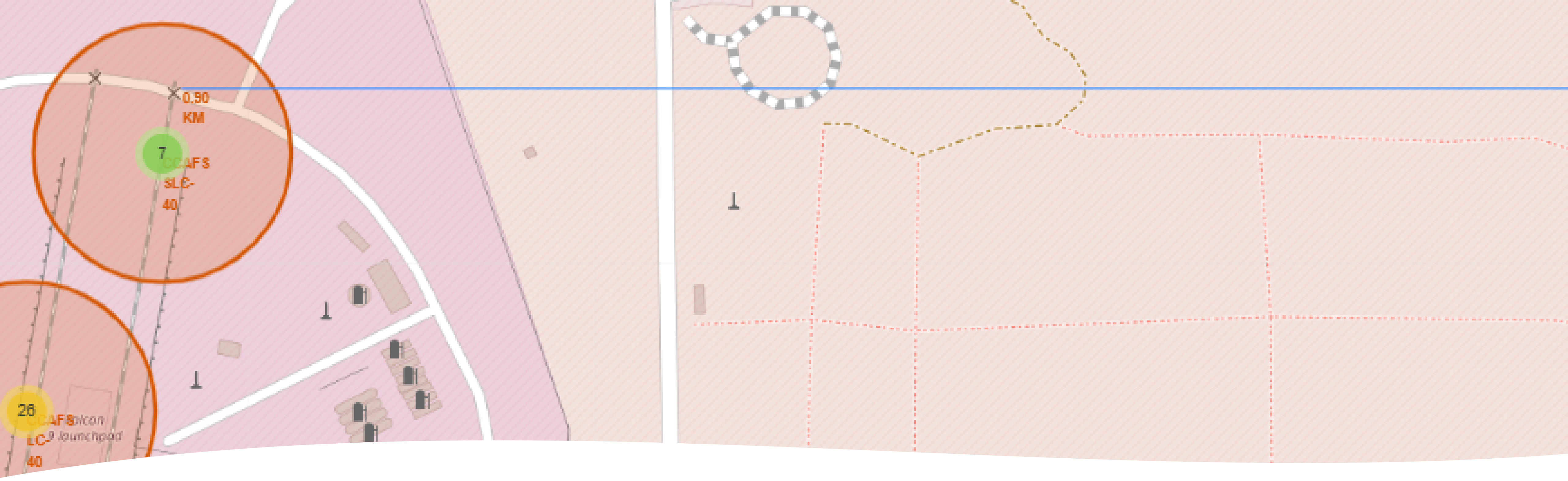
Folium Map Launch Sites

- Map that shows the physical locations of all the launch sites, so there is better visualization of where the launch sites are located geographically rather than just longitude and latitude data.



Folium Map Launch Outcomes

- By zooming in on the map and clicking on the launch site marker, markers indicating the launch outcomes appears
- A successful outcome is represented by a green marker, and a failed outcome is represented by a red marker. The markers are placed at the exact coordinates of the launches.
- This allows a useful visualization of the proportion of successful and failed launches for each site



Folium Map Proximities

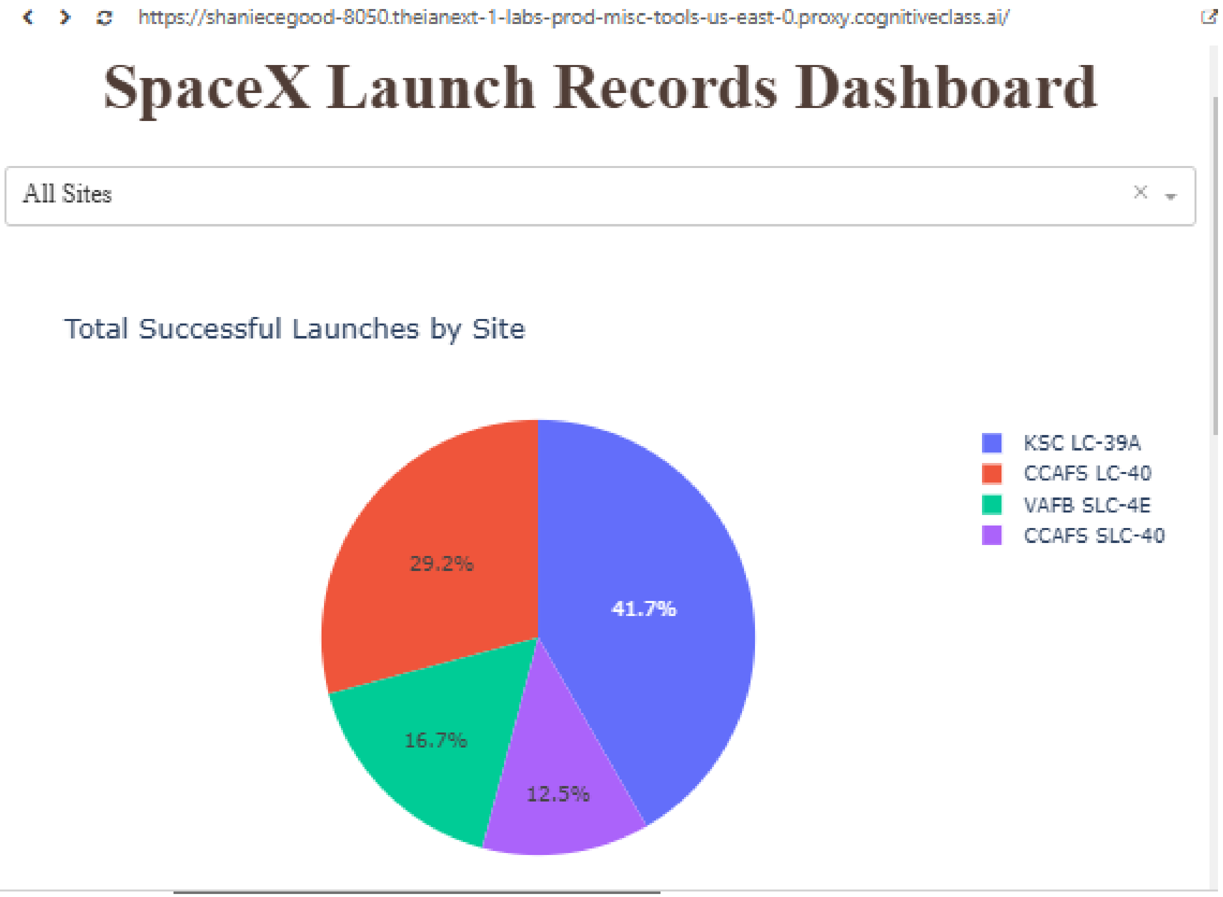
- Adding distance markers that also show the distance between the launch site and other geographical markers
- The blue line represents the distance between the launch site and the ocean. Zooming in shows that launch site CCAFS-SLC 40 is 0.50 km from the ocean.
- This allows quick calculation and reference of launch sites proximities to important geographical markers, such as bodies of water, highways, and major cities. Thus there is already information about where to possibly build another launch site if needed,

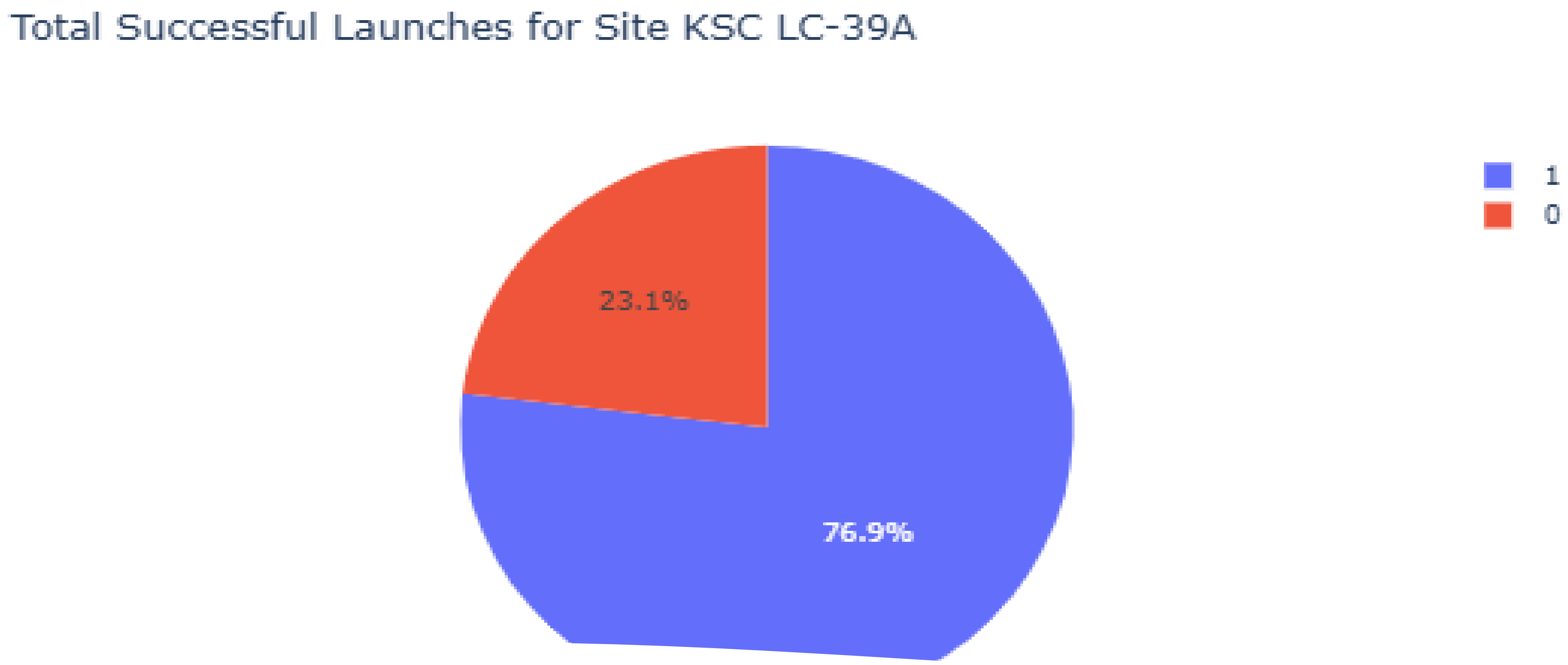
Section 4

Build a Dashboard with Plotly Dash

Dashboard Pie Chart All Sites

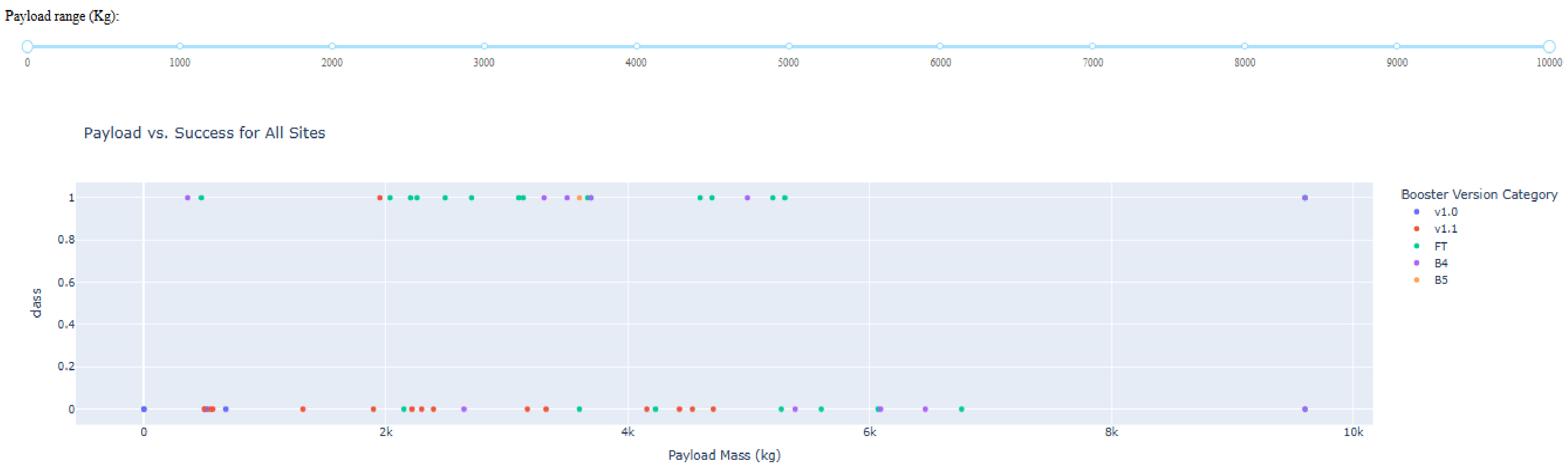
- Success count for all sites, in a pie chart
- Launch site KSC LC-39A has the most successful launches compared to all the other launch sites. With 41.7% of launches being successful.





Dashboard Pie Chart Site with Highest Success Ratio

- Pie chart for the launch site with highest launch success ratio
- Site KSC LC-39A has the highest success ratio with 76.9% successful launches and 23.1% failed launches.



Dashboard Payload Success with Range Slider

- Payload vs. Launch Outcome scatter plot for all sites. Can select specific payload amount of range with range slider. Dot color corresponds to booster version per legend on right
- This scatter plot can be used to easily determine which boosters have more success with certain payload ranges.
- See Appendix for more plots showing some boosters success rates

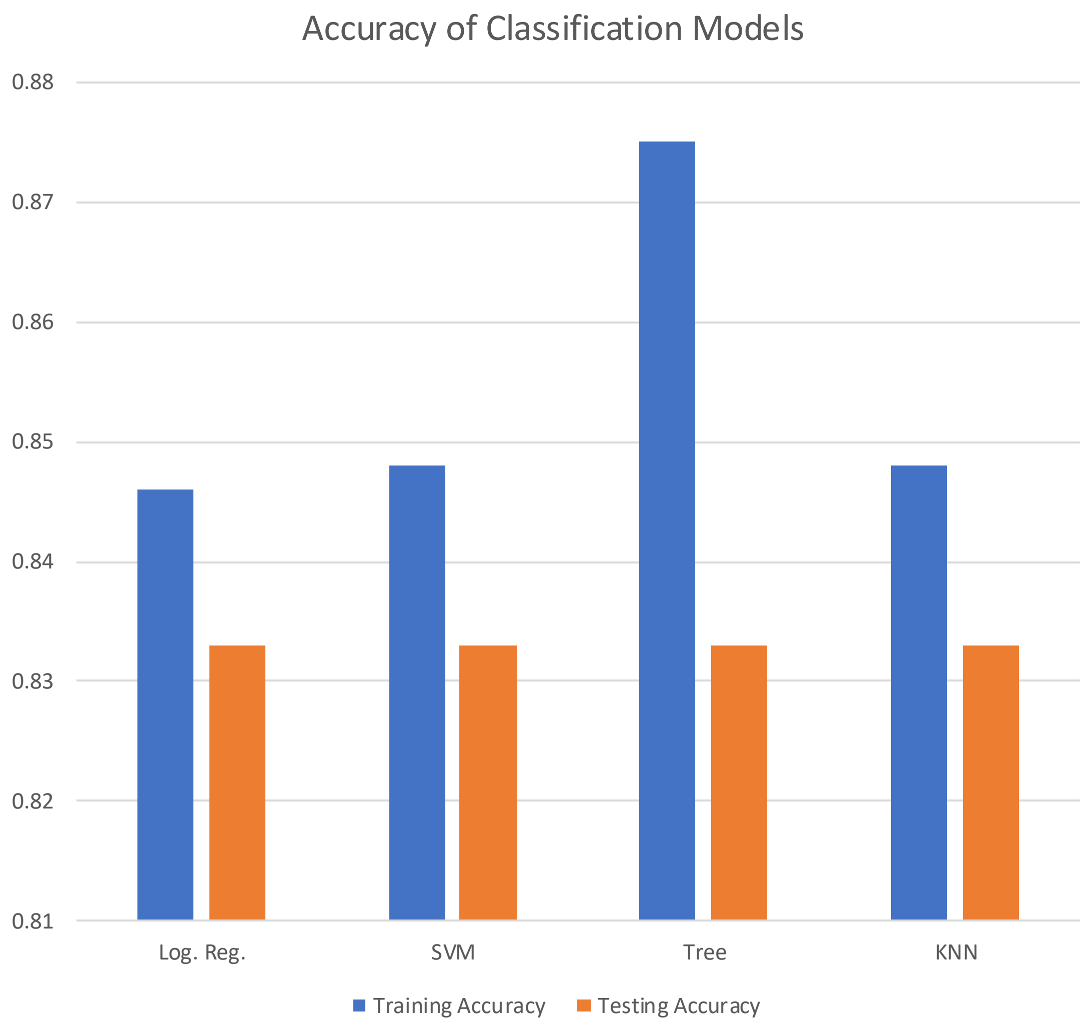
Section 5

Predictive Analysis (Classification)

Classification Accuracy

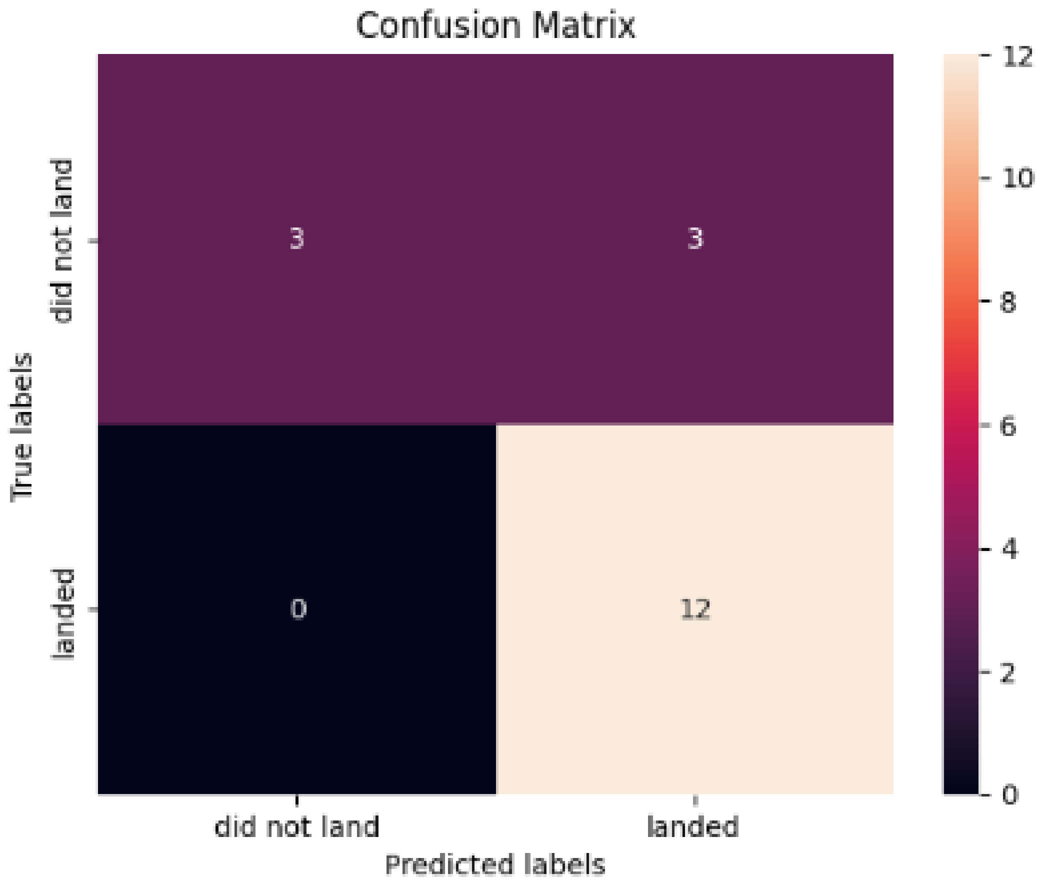
- Visualization of the built model accuracy for all built classification models

- The accuracy of all models on the test data is 83.3%. The model with the highest accuracy on the training data is the Decision tree with an accuracy of 87.5%. Thus the Decision Tree is the best model to use.



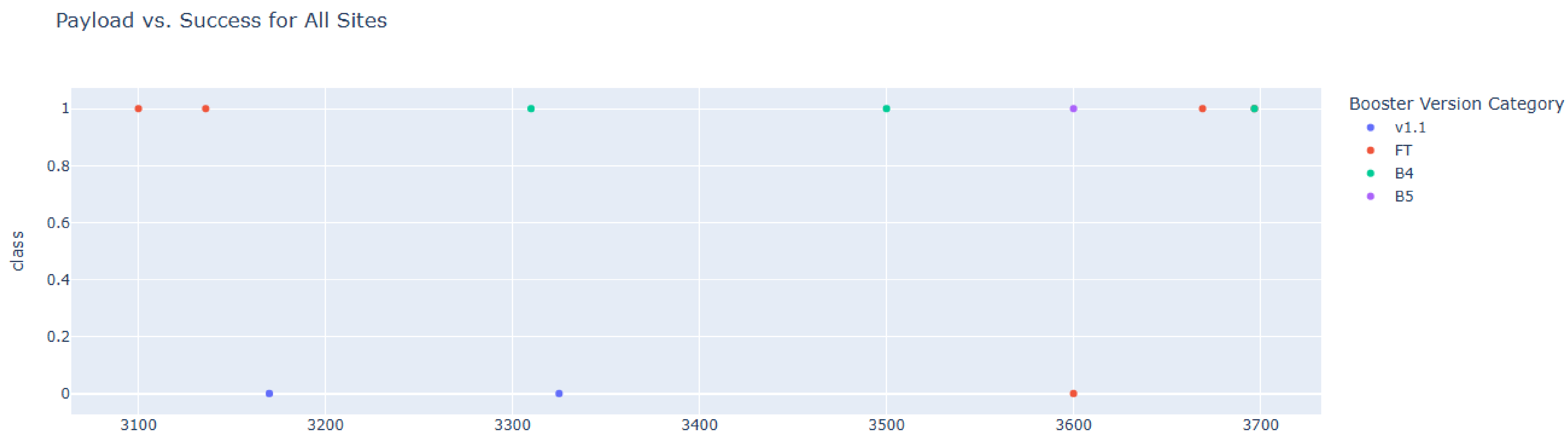
Confusion Matrix

All models had the same confusion matrix that shows the models did not produce any false positives.



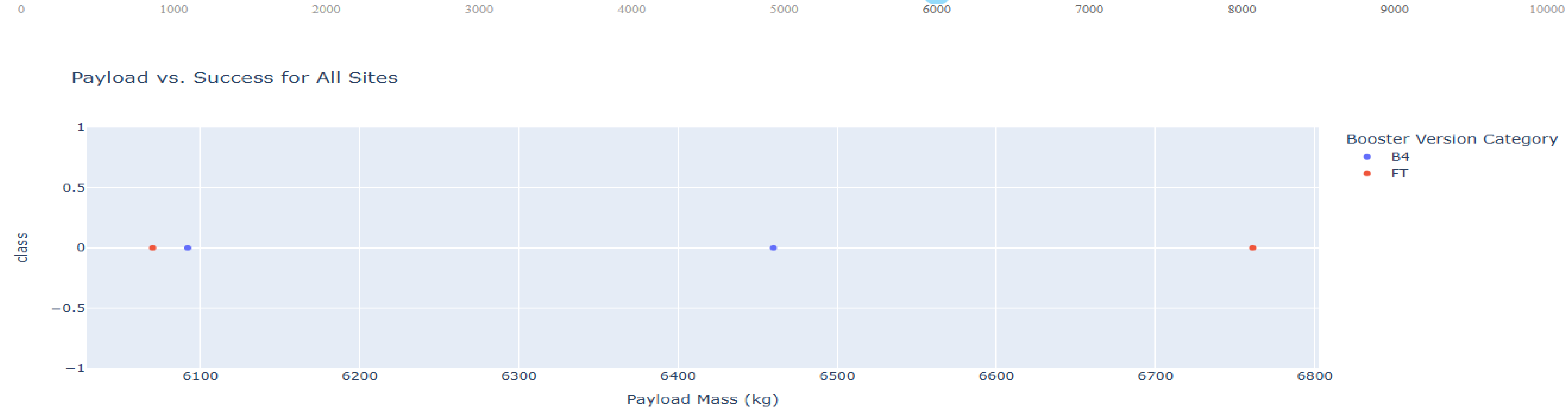
Conclusions

- After web scraping data from SpaceX API added a column to the dataframe that classifies if a landing was successful or failed.
- Using data visualization and SQL, was able to determine that the **best factors for predicting landings were booster version, payload mass, and flight number per launch site. For each site the lighter the payload and with more launches, the launches tend to be more successful. Successful launches increased from 2013 up to 2017.**
- Created Folium map to easily visualize where launch sites are located geographically and show the exact locations of failed and successful launches at each launch site.
- Created a dashboard that showed a pie chart of the ratio of successful launches for each launch site. Also included a scatter plot of successful and failed launces for each booster version given a range of payload mass. The range of payload mass can be changed using a slider. **Site KSC LC-39A had the highest ratio of successful launches**.
- Predictive analysis with 4 different classification models, which all gave an **accuracy on the test data of 83.3%. However it was decided that Decision Tree will be the best model because it has the highest accuracy on the training data at 87.5%.**



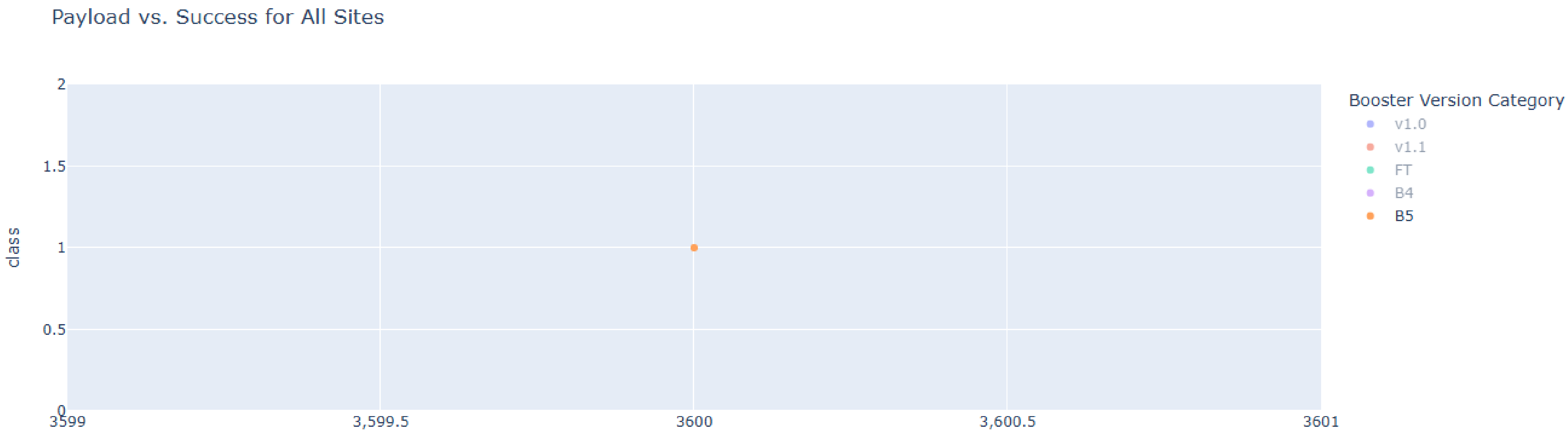
Appendix 1

- Payload range 3,000-4,000 kg had the highest success ratio.



Appendix 2

- Payload range 6,000-9,000 kg had the lowest success ratio, with no successful launches.



Appendix 3

- B5 booster has the highest success ratio. Booster had 1 flight only that had a successful landing, with a payload of 3,600 kg.

Thank you!