IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Aaron A.
2024/10/28

Outline

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

Executive Summary

- Data on SpaceX Falcon 9 was collected and analyzed for insights on successful landing outcomes. Predictive models were trained on relevant mission parameters to predict success.
- Summary of all results

Introduction

With the arrival of the commercial space age, many companies are making efforts to provide affordable space travel. The success of SpaceX in particular offers insights on how to better implement and design working and affordable rocket launches.

For SpaceY, the present study seeks to determine successful first stage launch landings and predict prices for launches based on mission parameters such as: payload mass, orbit, booster version, launch sites, and more.

Section 1

Methodology

Methodology

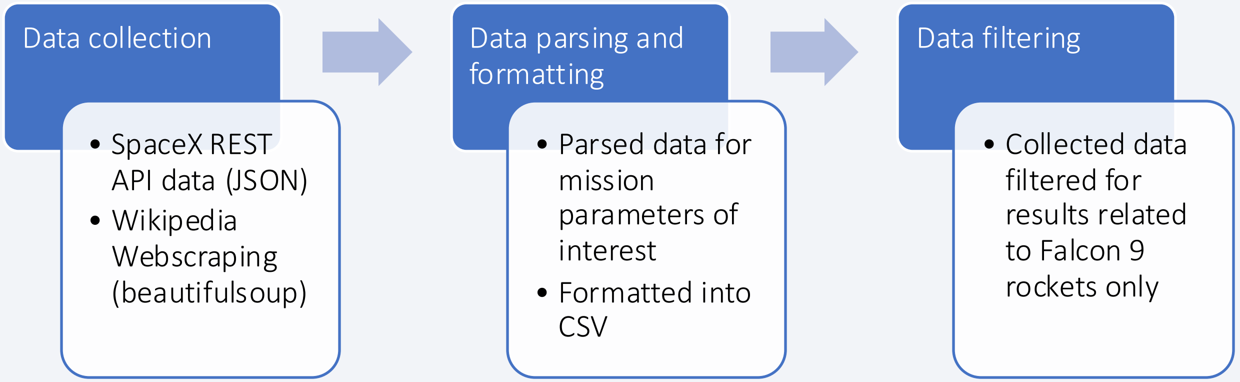
Executive Summary

- Data collection methodology:
    - SpaceX launch data was collected through the SpaceX API and formatted using Python
    - Data was filtered for results related to Falcon 9 rocket launches only as this is the leading rocket of interest.
- Perform data wrangling
    - Records of Falcon 9 and Falcon Heavy launches were scraped from Wikipedia
    - Exploratory data analysis was conducted to calculate the number of launches on each site, the number and occurrence of each orbit, the number and occurrence of mission outcome of the orbits, and to create a landing outcome label.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
    - Models were generated with an 80/20 train-test split and evaluated for accuracy using GridSearchCV approach
    - Models included: Logistic Regression, SVM, KNN, and Decision Tree.

Data Collection

Data collected:

- SpaceX launch data via SpaceX REST API
  - Provides data on booster type, payload, launch specifications, landing outcomes, etc.
  - API endpoints: api.spacexdata.com/v4/
- Falcon 9 launch data scraped from Wikipedia via BeautifulSoup
- Data was filtered for results related to Falcon 9 rocket launches only as this is the leading rocket of interest.



Data Collection – SpaceX API

Data collection with SpaceX REST

Get Response from SpaceX API

↓

Convert Response to JSON

↓

Parse for relevant data

↓

Create and filter dataframe

↓

Export to CSV

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

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

```
data = pd.json_normalize(response.json())
```

```
getBoosterVersion(data)
getLaunchSite(data)
getPayloadData(data)
getCoreData(data)

launch_dict = {'FlightNumber': list(data['flight_number']),
'Date': list(data['date']),
'BoosterVersion':BoosterVersion,
'PayloadMass':PayloadMass,
'Orbit':Orbit,
'LaunchSite':LaunchSite,
'Outcome':Outcome,
'Flights':Flights,
'GridFins':GridFins,
'Reused':Reused,
'Legs':Legs,
'LandingPad':LandingPad,
'Block':Block,
'ReusedCount':ReusedCount,
'Serial':Serial,
'Longitude': Longitude,
'Latitude': Latitude}

data_from_launch = pd.DataFrame.from_dict(launch_dict)
```

```
data_falcon9 = data_from_launch[data_from_launch["BoosterVersion"] == 'Falcon 9'].copy()
```

```
data_falcon9.to_csv('dataset_part_1.csv', index=False)
```

[GitHub - Data Collection]

Data Collection - Scraping

Webscraping from Wikipedia with BeautifulSoup

Request Falcon9 Launch Wiki Page from URL

```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"
response = requests.get(static_url)
soup = BeautifulSoup(response.content,'html.parser')
```

Extract all column/variable names from HTML Table header

```
launch_dict= dict.fromkeys(column_names)

# Remove an irrelvant column
del launch_dict['Date and time ( )']

# Let's initial the launch_dict with each value to be an empty list
launch_dict['Flight No.'] = []
launch_dict['Launch site'] = []
launch_dict['Payload'] = []
launch_dict['Payload mass'] = []
launch_dict['Orbit'] = []
launch_dict['Customer'] = []
launch_dict['Launch outcome'] = []
# Added some new columns
launch_dict['Version Booster']=[]
launch_dict['Booster landing']=[]
launch_dict['Date']=[]
launch_dict['Time']=[]
```

Create dataframe by parsing launch HTML tables

```
df= pd.DataFrame({ key:pd.Series(value) for key, value in launch_dict.items() })
df
```

	Flight No.	Launch site	Payload	Payload mass	Orbit	Customer	Launch outcome	Version Booster	Booster landing	Date	Time
0	1	CCAFS	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success\n	F9 v1.07B0003.18	Failure	4 June 2010	18:45
1	1	CCAFS	Dragon	0	LEO	NASA	Success	F9 v1.07B0003.18	Failure	4 June 2010	18:45
2	2	CCAFS	Dragon	525 kg	LEO	NASA	Success	F9 v1.07B0004.18	No attempt\n	8 December 2010	15:43
3	3	CCAFS	SpaceX CRS-1	4,700 kg	LEO	NASA	Success\n	F9 v1.07B0005.18	No attempt	22 May 2012	07:44
4	4	CCAFS	SpaceX CRS-2	4,877 kg	LEO	NASA	Success\n	F9 v1.07B0006.18	No attempt\n	8 October 2012	00:35
...	...	...	...	...	...	...	...	...	...	...	...
329	117	KSC	Starlink	~14,000 kg	LEO	None	Success\n	F9 B5B1051.10657	Success	9 May 2021	06:42
330	118	CCSFS	Starlink	15,600 kg	LEO	None	Success\n	F9 B5B1058.8660	Success	15 May 2021	22:56
331	119	KSC	SpaceX CRS-22	3,328 kg	LEO	NaN	NaN	F9 B5B1063.2665	NaN	26 May 2021	18:59
332	120	CCSFS	SXM-8	7,000 kg	GTO	NaN	NaN	F9 B5B1067.1668	NaN	3 June 2021	17:29
333	121	NaN	NaN	NaN	NaN	NaN	NaN	F9 B5	NaN	6 June 2021	04:26

GitHub - Webscraping

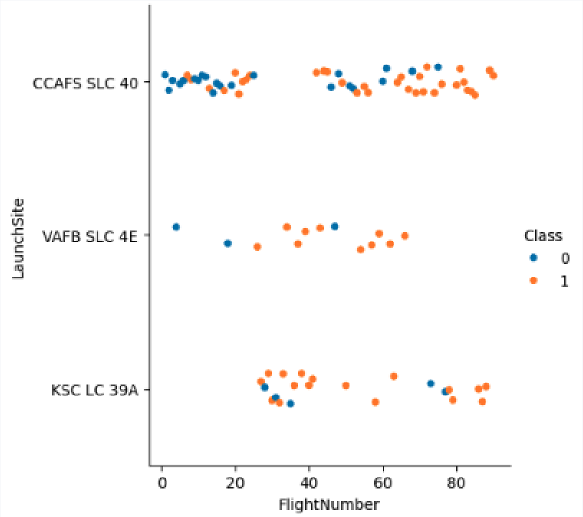
Data Wrangling

1. Identify and calculate missing values
2. Replace missing values in 'PayloadMass' with average payload
3. Calculate number of launches on each site
4. Calculate number and occurrence of each orbit
5. Calculate number and occurrence of mission outcome of the orbits
6. Create a landing outcome label based on Outcome
7. Export cleaned data as CSV

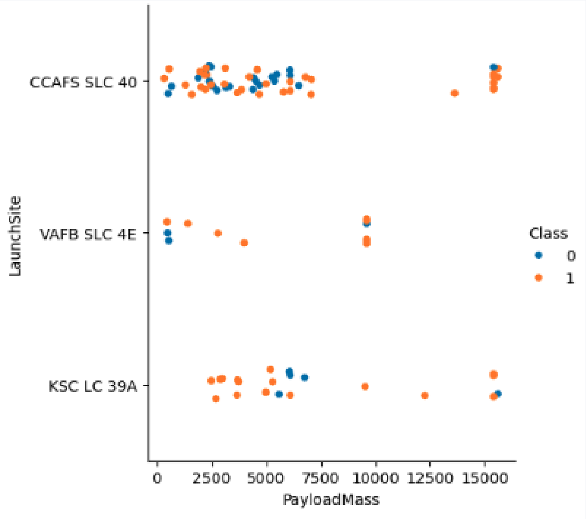
[GitHub - Data Wrangling]

EDA with Data Visualization

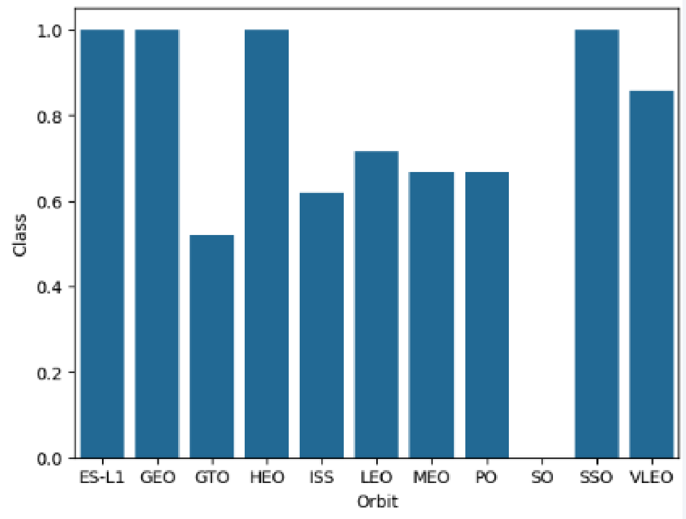
Flight Number vs. Launch Site



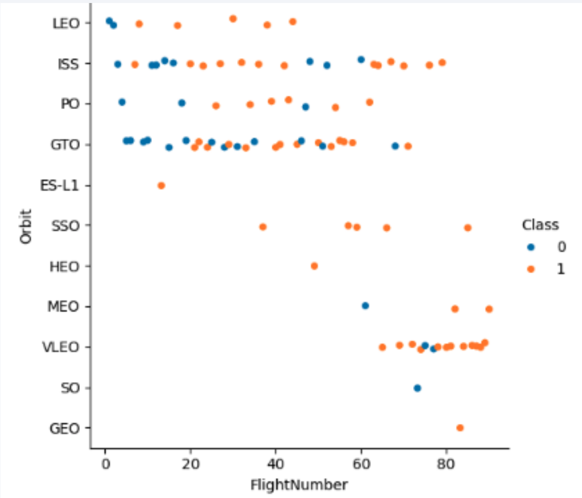
Payload Mass vs. Launch Site



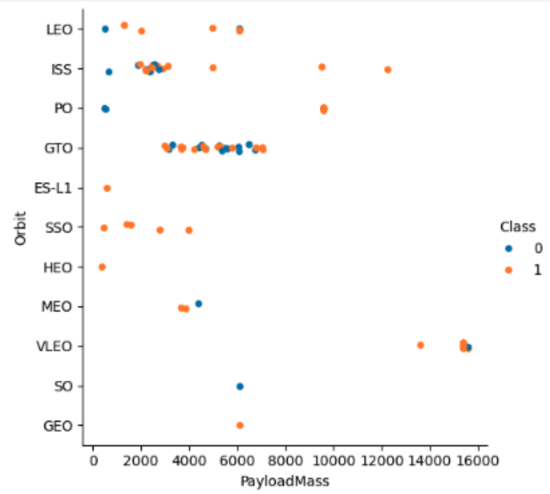
Relationship between Success Class and Orbit



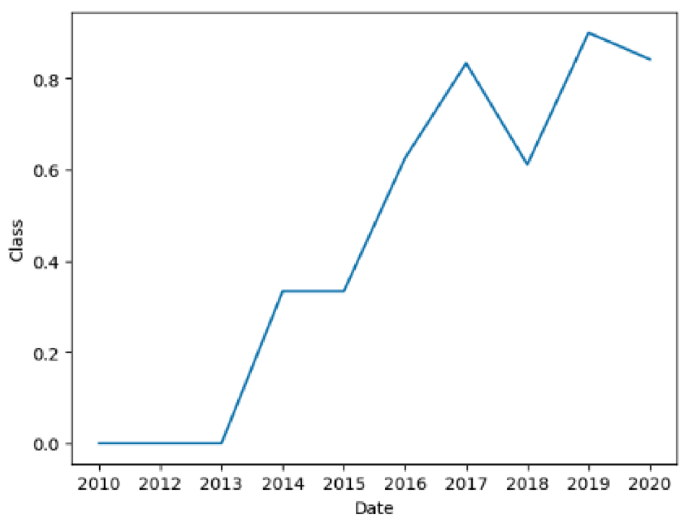
Flight Number vs. Orbit



Payload Mass vs. Orbit



Launch Success Yearly Trend



[GitHub - EDA Data Visualization]

EDA with SQL

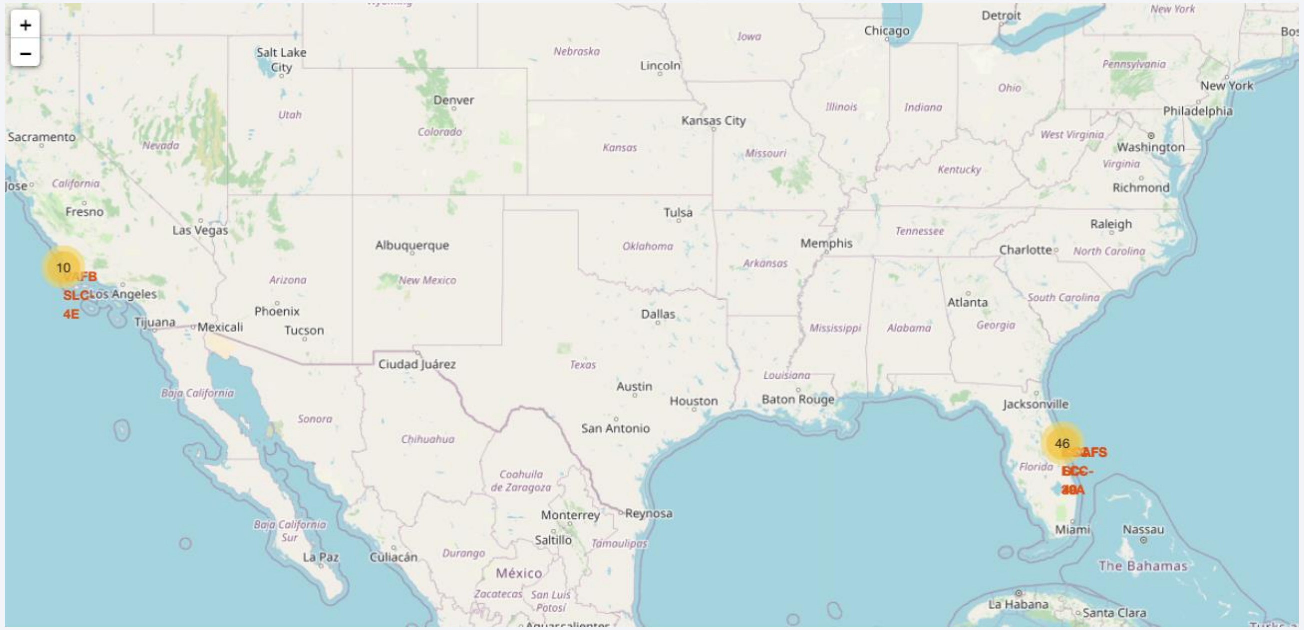
- SQL Queries
  - Create SPACEXTABLE from SpaceX CSV
  - Displayed names of unique launch sites in the space mission
  - Displayed 5 records where launch sites begin with the string 'CCA'
  - Displayed the total payload mass carried by boosters launched by NASA (CRS)
  - Displayed average payload mass carried by booster version F9 v1.1
  - Listed the data when the first successful landing outcome in ground pad was achieved
  - Listed the names of the boosters which had success in drone ship and had payload mass between 4000 and 6000 kg
  - Listed the total number of successful and failed mission outcomes
  - Listed the names of the booster versions which had carried the maximum payload mass with a subquery
  - Listed records that displayed the month, failure landing outcomes, booster version, and launch site for the year 2015
  - Ranked the count of landing outcomes between the dates 2010-06-04 and 2017-03-20 in descending order

[GitHub - EDA with SQL]()

Build an Interactive Map with Folium

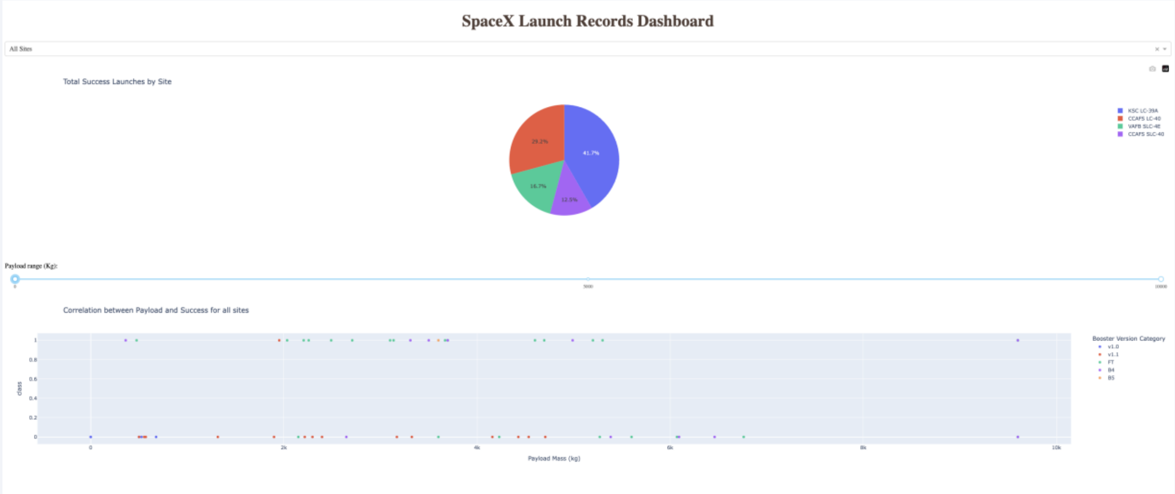
- Created an interactive map with Folium with markers and circles to map launch sites geographically colored by success (green) and failure (red)
    - Adding these to a map would help provide some insights regarding geographical patterns about launch sites. Launch success may be dependent on location so finding an optimal location for building a launch site can be critical to success.

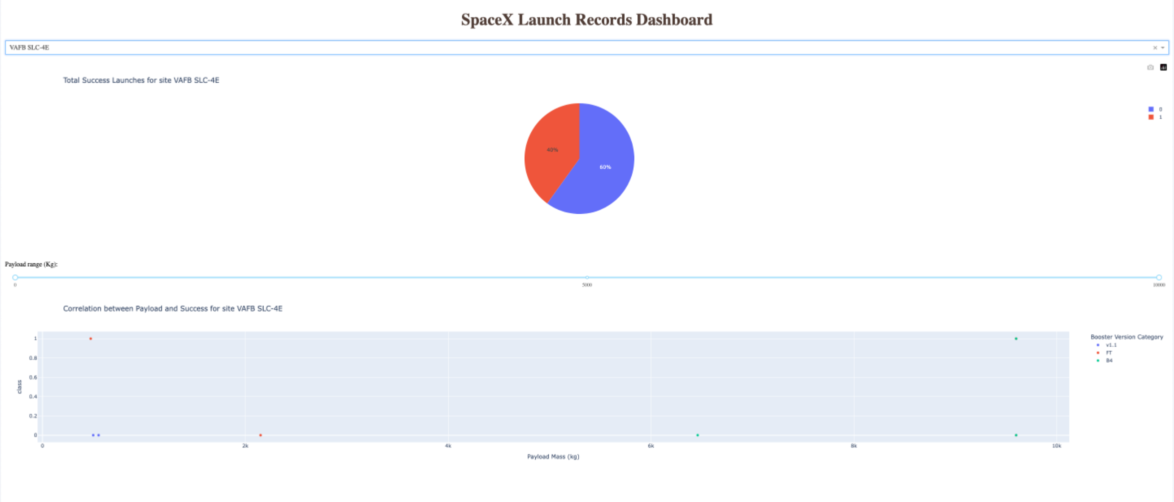
[GitHub - Folium Interactive Map]



Build a Dashboard with Plotly Dash

- The Plotly Dashboard consisted of pie charts and scatterplots to show overall success for all launch sites, a breakdown of success and failure for specific sites and an interactive scaler to visualize the impact of weighted payloads.

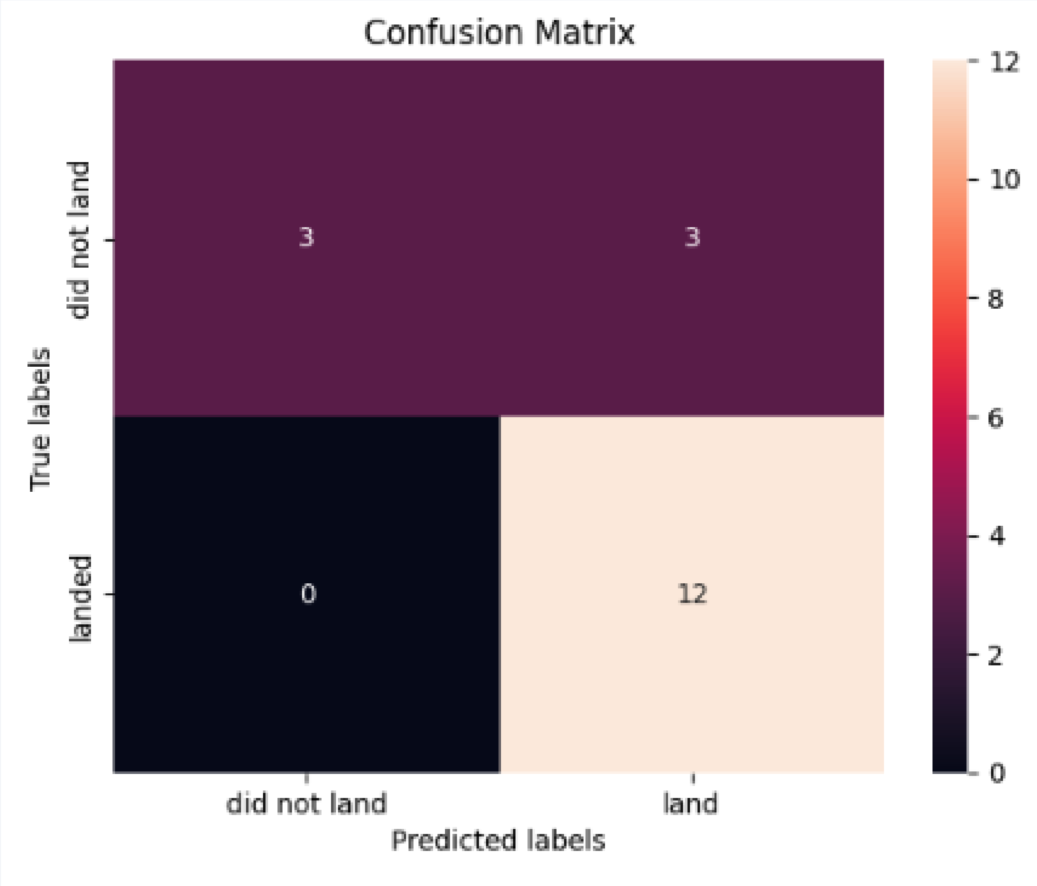




[GitHub - SpaceX Plotly Dashboard]

Predictive Analysis (Classification)

- Objective: Create a model to predict if a launch will land successfully
- Models evaluated: Logistic Regression (LR), Support Vector Machine (SVM), Decision Tree (DT), and K-Nearest Neighbor (KNN)
- Models were evaluated on an 80/20 train-test split of scaled numerical launch data. Categorical values were one-hot encoded.
- All models achieved accuracy ~83%.



GitHub - SpaceX ML Classification

Results

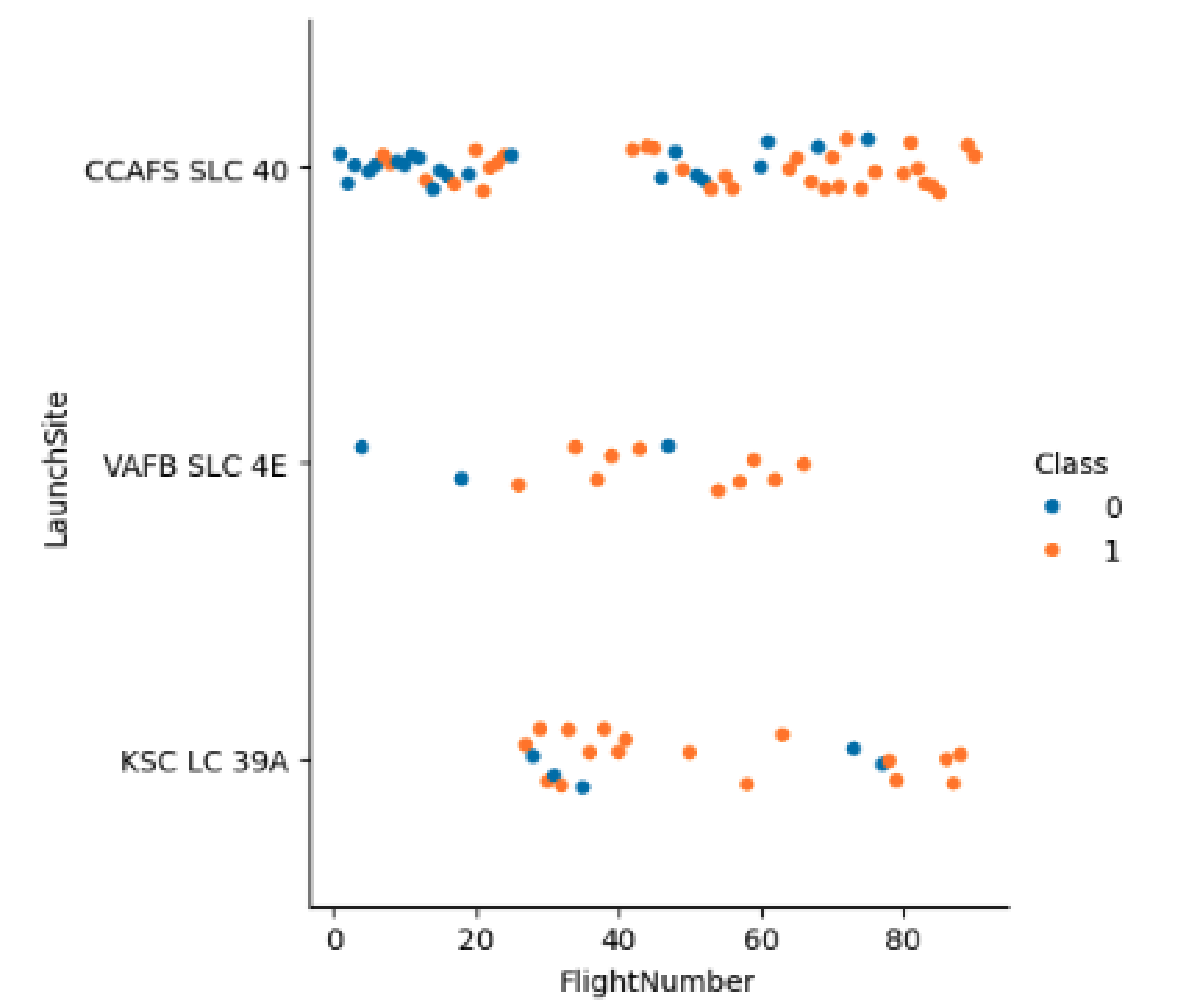
- Exploratory data analysis results
  - API and Webscraping were sufficient in generating a working dataset of SpaceX launch data
  - EDA with SQL was effective in exploring SpaceX launch data
  - EDA with visualization (Python and Plotly) provides informative information on site specific success
- Predictive analysis results
  - All models achieved accuracy ~83%.

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

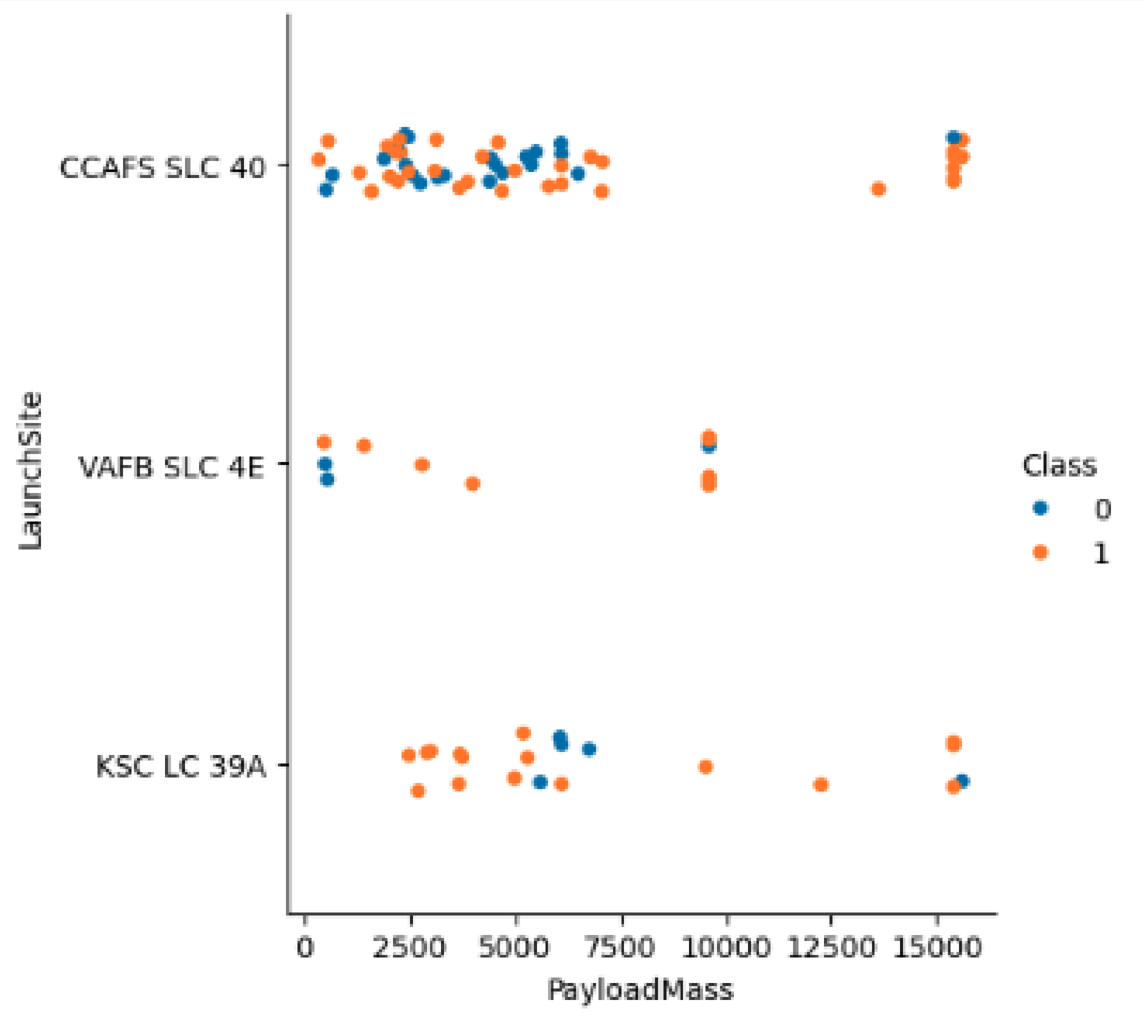
Increased flight numbers suggests increased success across launch sites.



Payload vs. Launch Site

Increasing payload does not indicate positive correlation with launch site success rate.

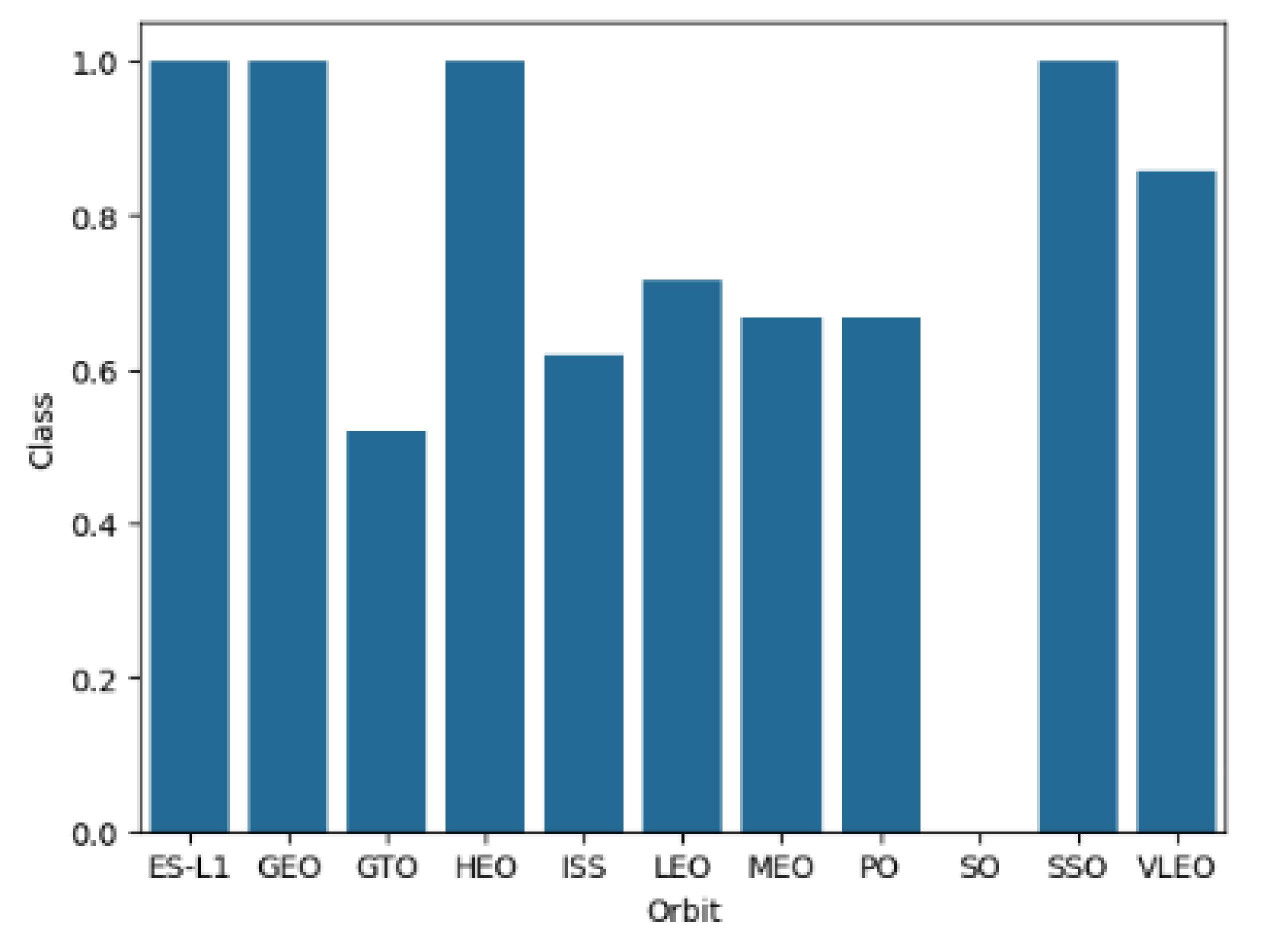
Note, VAFB SLC 4E did not launch rockets with a heavy payload.



Success Rate vs. Orbit Type

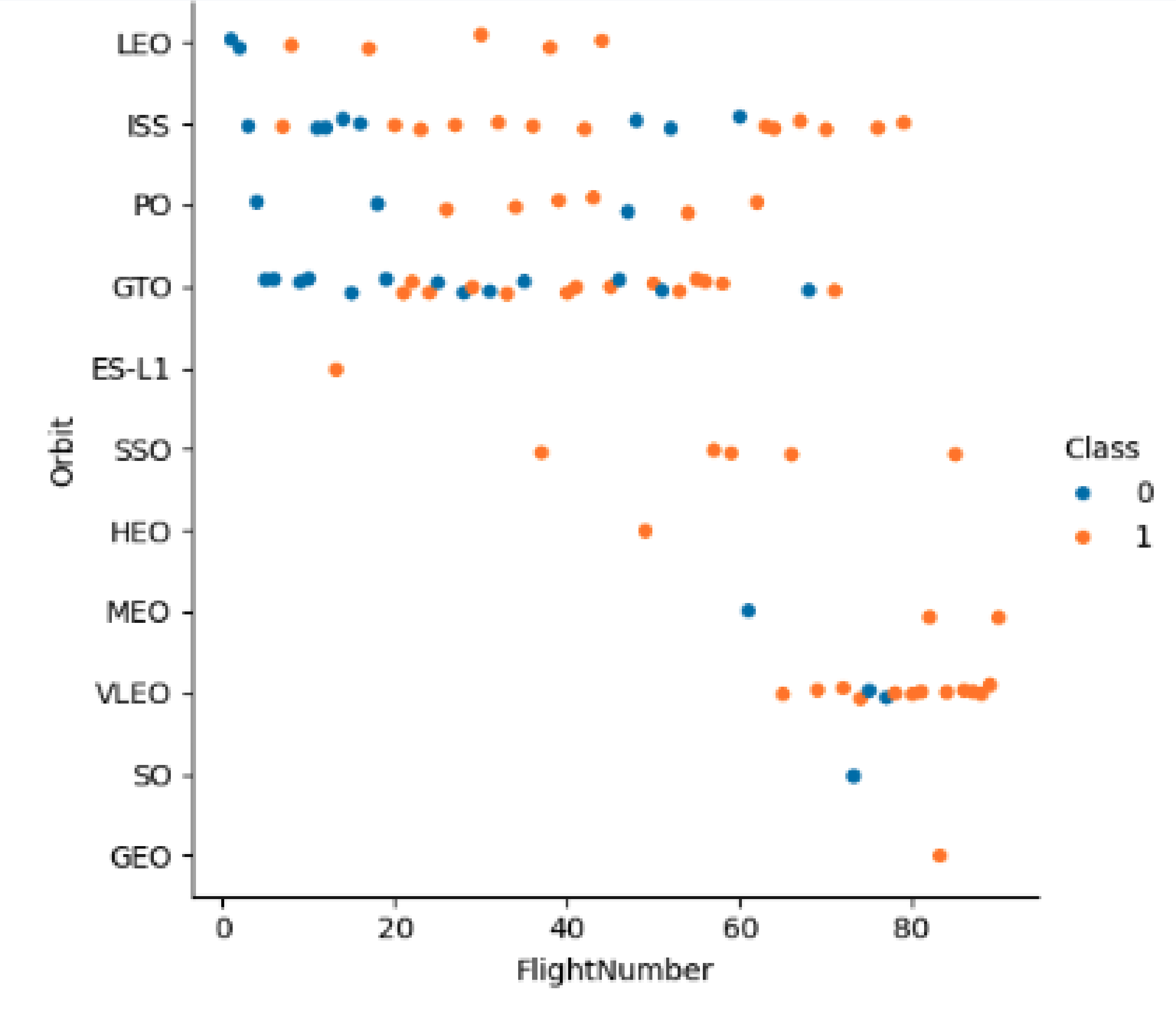
ES-L1, GEO, HEO, and SSO orbits have a 100% success rate.

SO orbit had a success rate of 0%.



Flight Number vs. Orbit Type

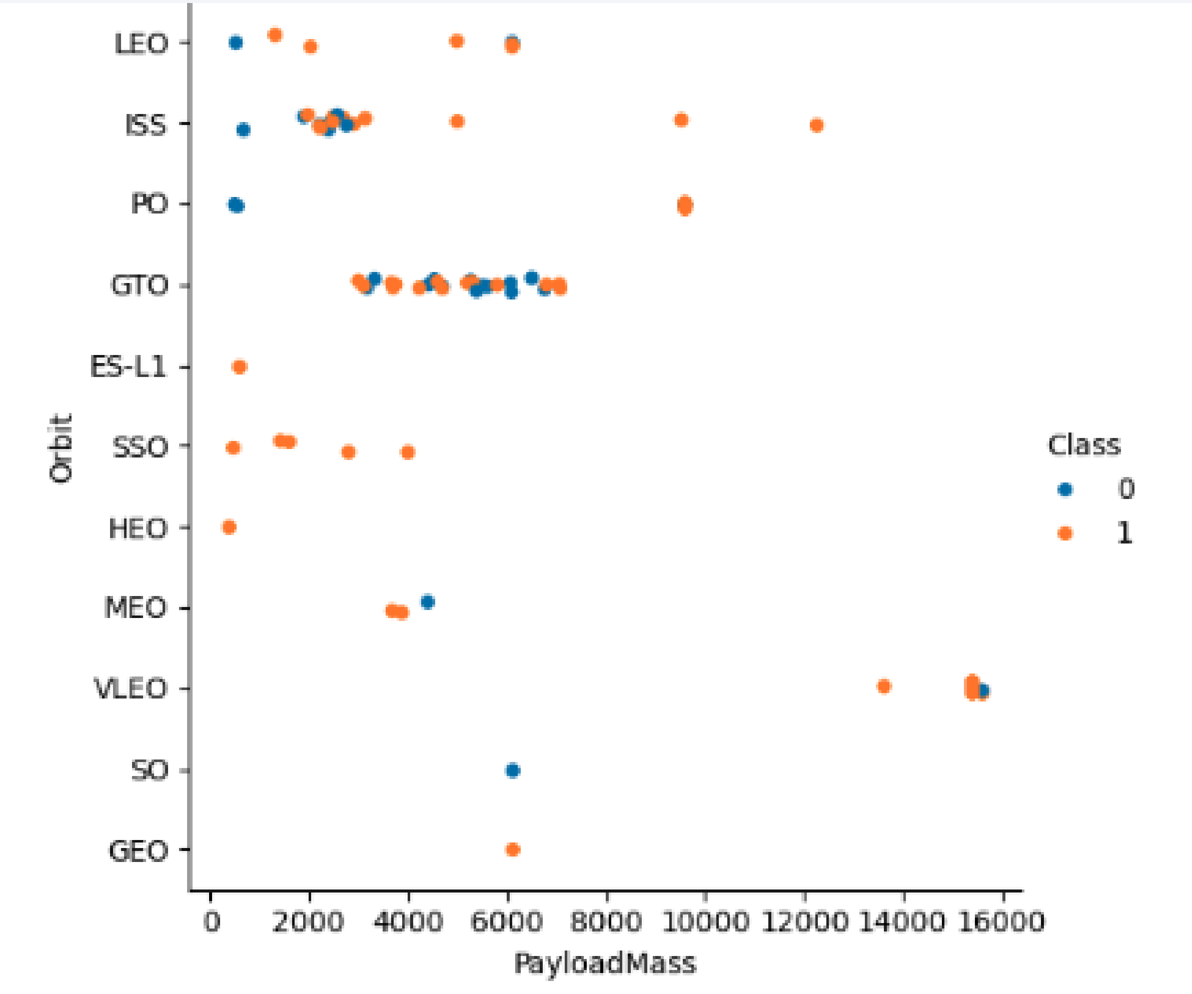
In the LEO orbit, success rate seems to be correlated with the number of flights. The GTO orbit appears to have no relationship between flight number and success.



Payload vs. Orbit Type

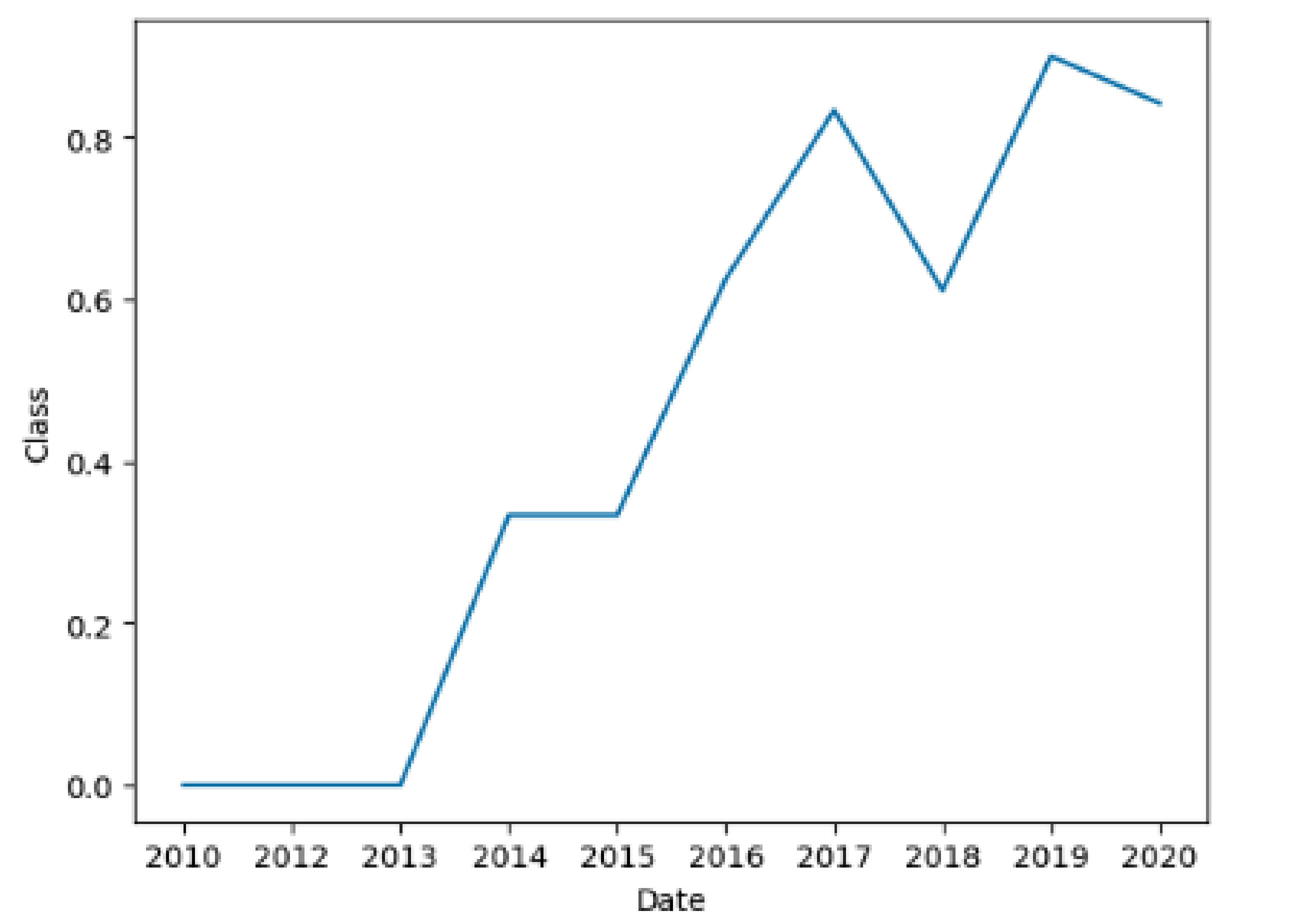
Heavier payloads are postively correlated with PO, LEO and ISS orbits.

However GTO does not have a clear relationship with payload and success rate.



Launch Success Yearly Trend

Success rate has increased since 2013 until 2020.



All Launch Site Names

Unique launch site names can be queried using DISTINCT

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

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

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

Launch Site Names Begin with 'CCA'

Querying for the first 5 launch sites that start with 'CCA' can be done using LIKE and LIMIT.

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

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-06-04	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

Calculate the total payload carried by boosters from NASA using SUM() function and WHERE to specify that Customer is NASA

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE Customer LIKE 'NASA (CRS)'
```

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

SUM(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

Calculate the average payload mass carried by booster version F9 v1.1 using the AVG() function and a WHERE clause to specify Booster_Version

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) as Average_Payload FROM SPACEXTABLE WHERE Booster_Version LIKE 'F9 v1.1'
```

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

Average_Payload
2928.4

First Successful Ground Landing Date

To find the first successful landing outcome on a ground pad, the query must use the MIN() function to find the earliest date from the table where the landing outcome is equal to 'Success (ground pad)'

```
%sql SELECT MIN(DATE) FROM SPACEXTABLE WHERE Landing_Outcome LIKE 'Success (ground pad)'
```

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

MIN(DATE)
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

List all successful booster versions using a WHERE clause specifying Mission_Outcome = 'Success' with a BETWEEN clause to limit payload between 4000 and 6000

```
%sql SELECT Booster_Version FROM SPACEXTABLE WHERE (Mission_Outcome = 'Success')
AND (PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000)
```

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

Booster_Version
F9 v1.1
F9 v1.1 B1011
F9 v1.1 B1014
F9 v1.1 B1016
F9 FT B1020
F9 FT B1022
F9 FT B1026
F9 FT B1030
F9 FT B1021.2
F9 FT B1032.1
F9 B4 B1040.1
F9 FT B1031.2
F9 FT B1032.2
F9 B4 B1040.2
F9 B5 B1046.2
F9 B5 B1047.2
F9 B5 B1046.3
F9 B5 B1048.3
F9 B5 B1051.2
F9 B5B1060.1
F9 B5 B1058.2
F9 B5B1062.1

Total Number of Successful and Failure Mission Outcomes

The total number of successful and failure outcomes can be displayed by selecting the outcome and COUNT from the table after grouping by mission outcome.

```
%sql SELECT Mission_Outcome, COUNT(*) AS total FROM SPACEXTABLE GROUP BY Mission_Outcome
```

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

Mission_Outcome	total
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

Boosters Carried Maximum Payload

List the names of booster versions that carried tha maximum payload mass by using a subquery to get the MAX payload from the whole table.

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

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

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

2015 Launch Records

List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015 by using SUBSTR to extract the Month and Year from 'Date'.

```
%sql SELECT SUBSTR(Date,6,2) AS Month, Landing_Outcome, Booster_Version, Launch_Site
FROM SPACEXTABLE
WHERE SUBSTR(Date,0,5)='2015' AND Landing_Outcome <> 'Success'
```

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

Month	Landing_Outcome	Booster_Version	Launch_Site
01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
02	Controlled (ocean)	F9 v1.1 B1013	CCAFS LC-40
03	No attempt	F9 v1.1 B1014	CCAFS LC-40
04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40
04	No attempt	F9 v1.1 B1016	CCAFS LC-40
06	Precluded (drone ship)	F9 v1.1 B1018	CCAFS LC-40
12	Success (ground pad)	F9 FT B1019	CCAFS LC-40

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

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 using COUNT and BETWEEN in the WHERE clause.

Using ORDER BY count and DESC will return the ranking in descending order.

```
%sql SELECT Landing_Outcome, COUNT(*) AS count FROM SPACEXTABLE
WHERE (Date BETWEEN '2010-06-04' AND '2017-03-20') ORDER BY count DESC
```

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

Landing_Outcome	count
Failure (parachute)	31

Section 3

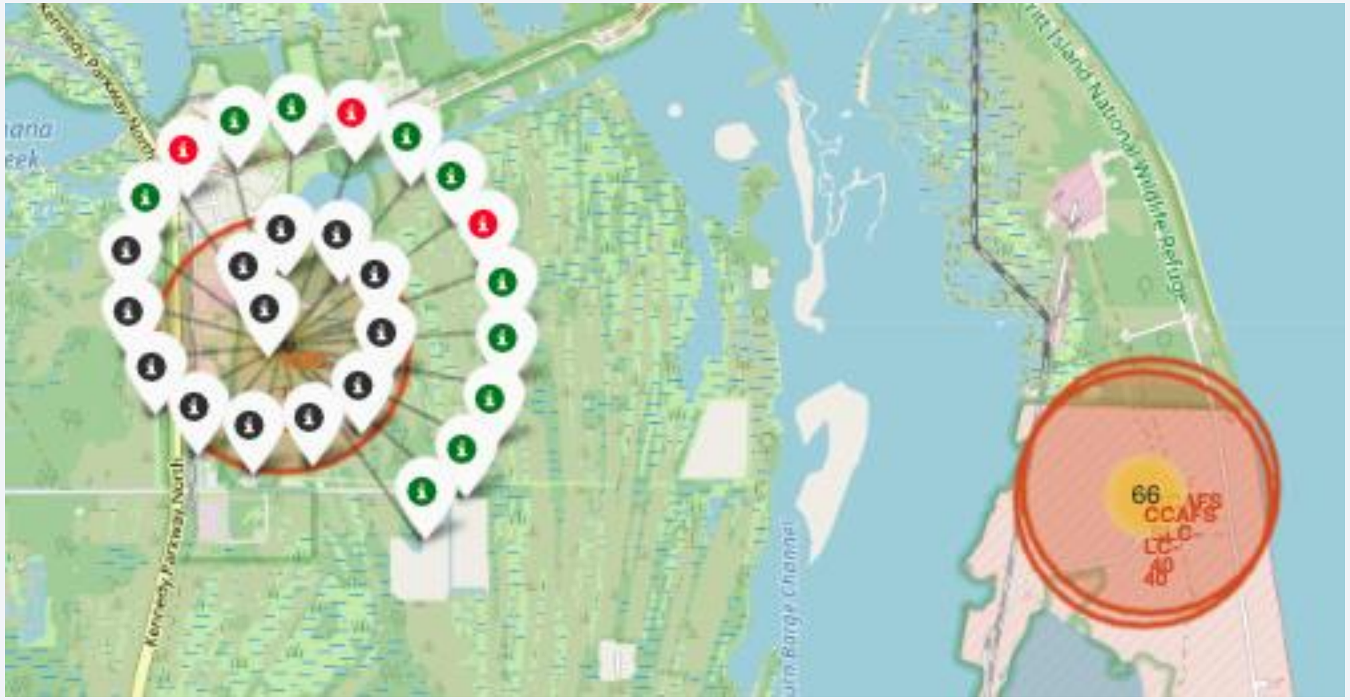
Launch Sites Proximities Analysis

All Launch Site Locations

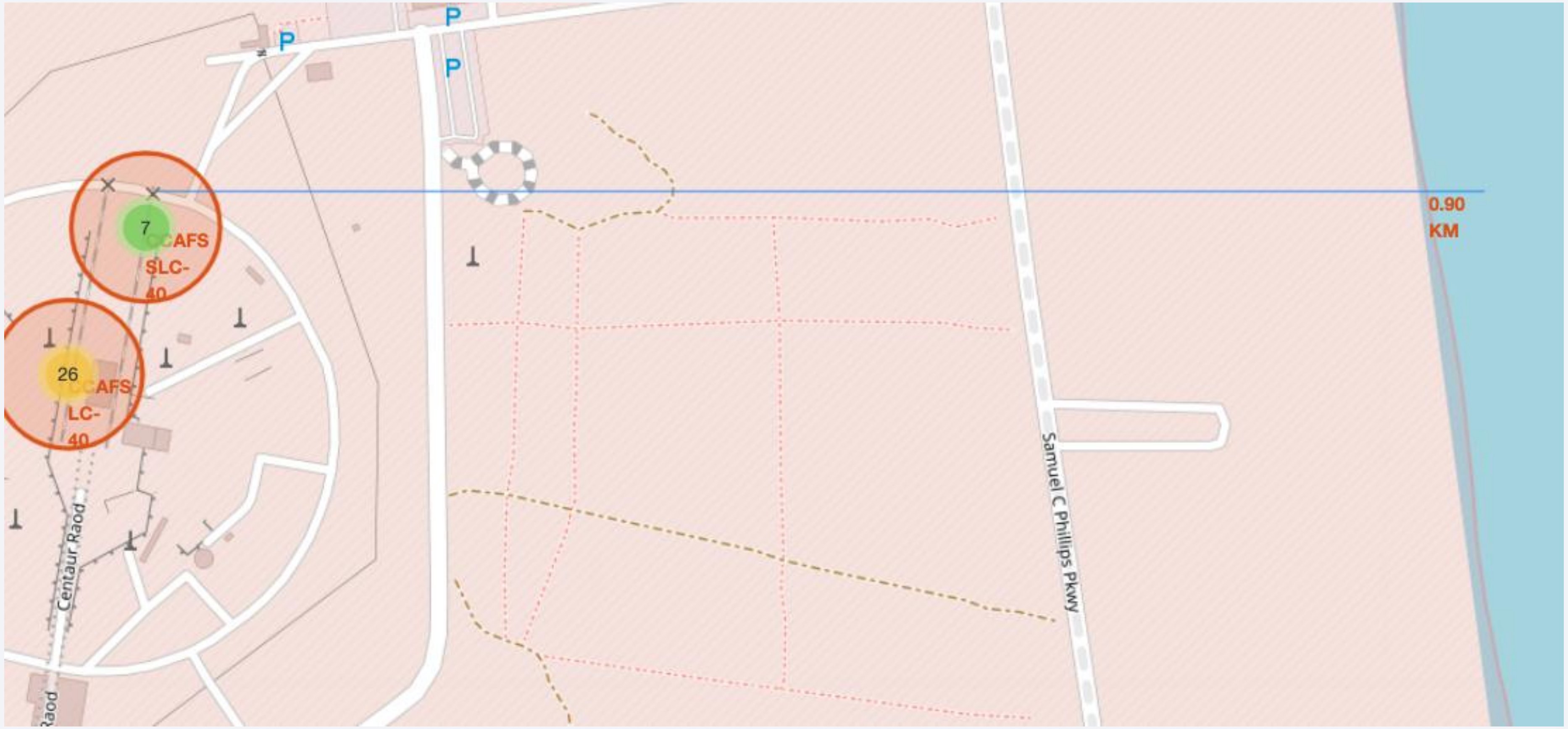
All launch sites are within the United States, specifically Florida and California (yellow)

Success-Failure Labeled Launch Outcomes

Successful launch outcomes are indicated in green, failed launch outcomes are indicated in red



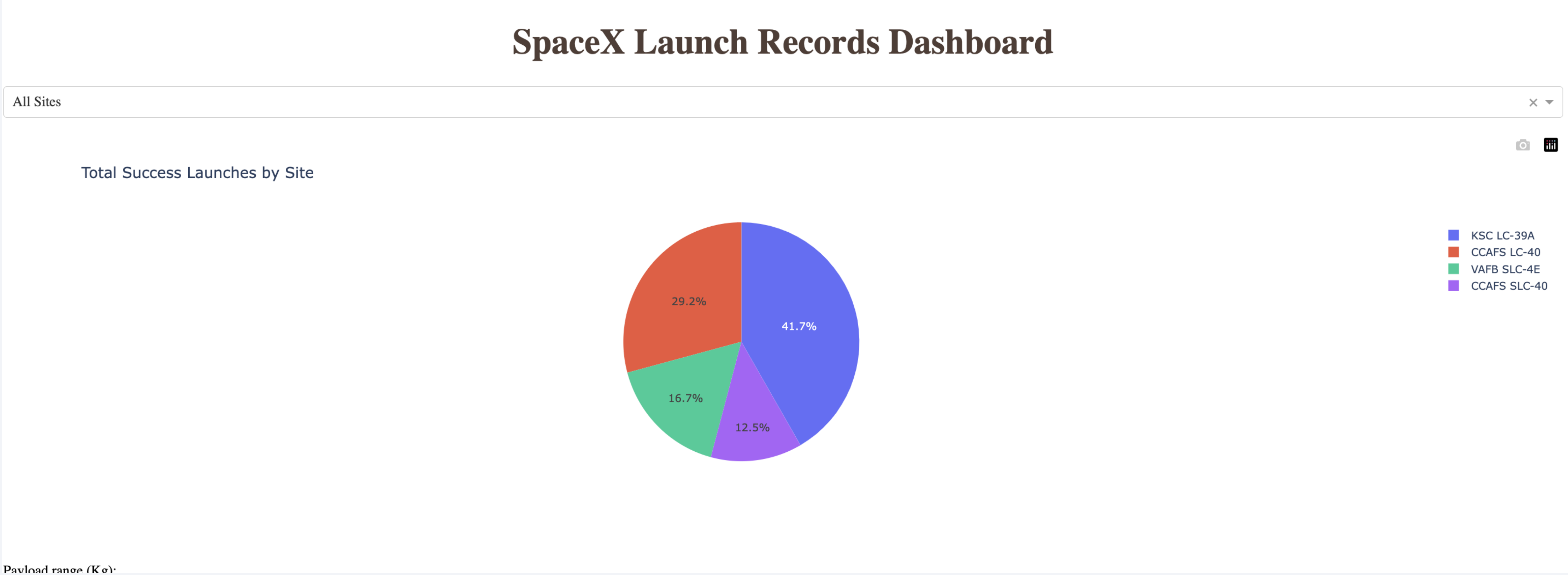
Launch Site to Proximities



Section 4

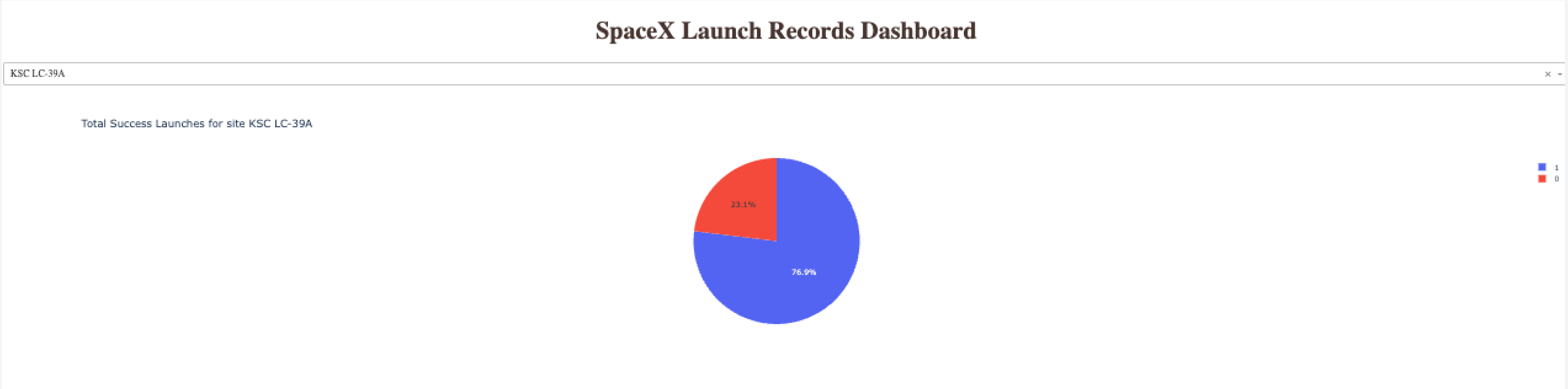
Build a Dashboard with Plotly Dash

Total Success Launches by Site



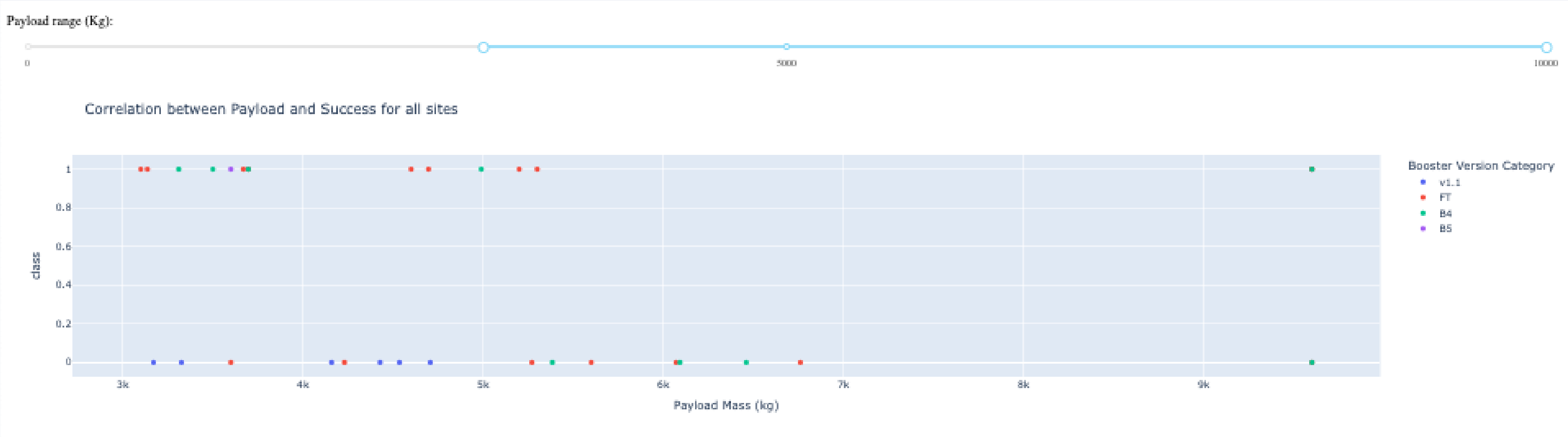
KSC LC-39A had the highest success rate with 41.7%. CCAFS LC-40, VAFB SLC-4E, and CCAFS SLC-40 had 29.2%, 16.7%, and 12.5% success respectively.

Highest Launch Success Ratio



KSC LC-39A has the highest launch success ration with 76.9% .

Payload and Launch Outcome for Different Boosters



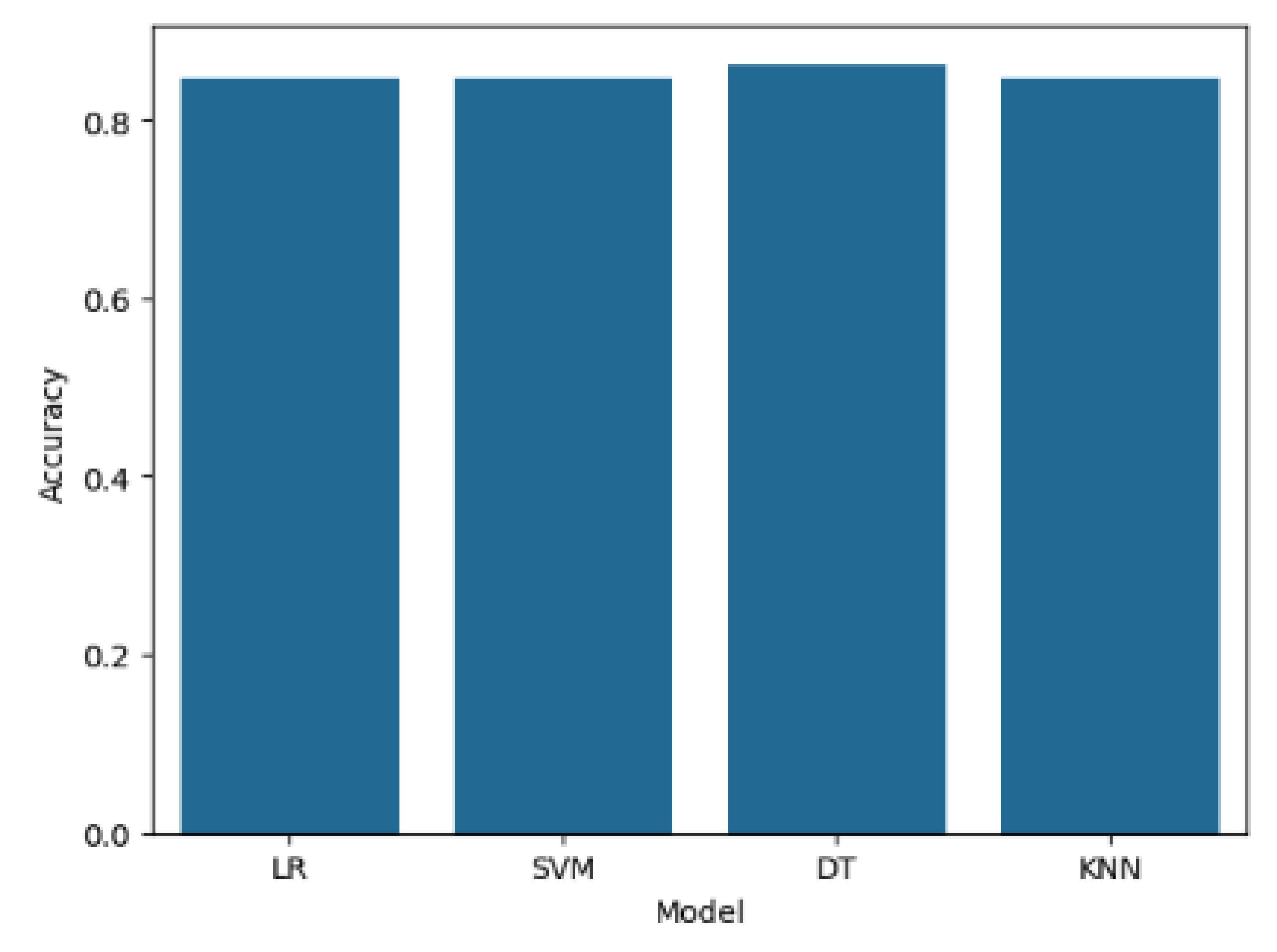
Payload vs Launch Outcome classification for payloads between 3000 and 10000 kg. As payload increases, FT and B4 booster versions have better success.

Section 5

Predictive Analysis (Classification)

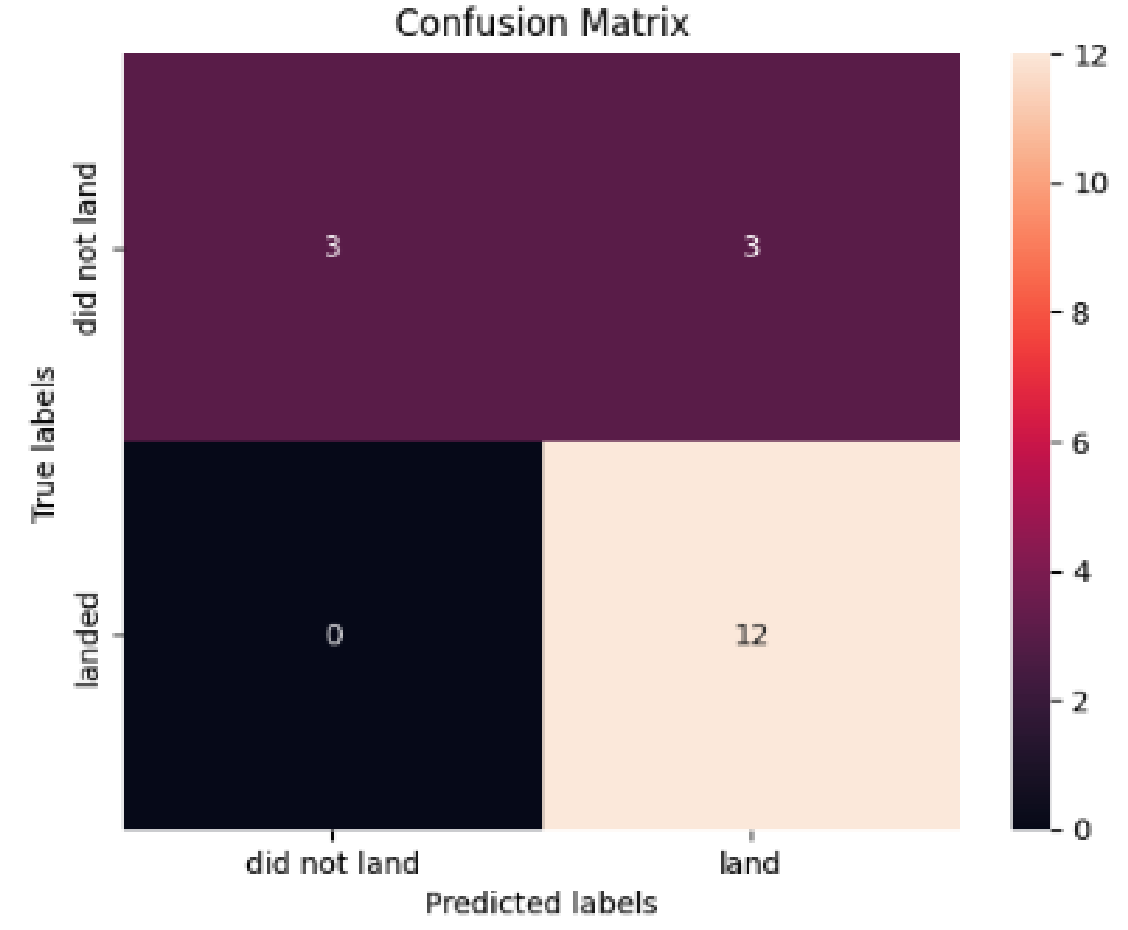
Classification Accuracy

- Bar plot showing best measured classification accuracy for Logistic Regression (LR), Support Vector Machine (SVM), Decision Tree (DT), and K-Nearest Neighbor (KNN) models.
- All models have almost equal accuracy, but DT has a slight advantage.



Confusion Matrix

As we see here, the model is able to distinguish between classes, but does indicate some false positives – predicts that 3 landed, but in reality they did not land.



Conclusions

- Highest success rate belonged to launches from KSC LC-39A
- There is some positive correlation between payload and orbit for signaling successful launch outcomes, however it correlations are poor at payloads < 5000 kg.
- The Decision Tree Model had the best accuracy.

Appendix

All code can be found on GitHub at: https://github.com/aar0n-aguh0b/coursera/tree/main/capstone

Thank you!