Winning Space Race with Data Science

- Capstone Project

Erik
June 9, 2022

Outline

- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion

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

Introduction

- **Project background**

  Commercial space age is here

  Space X has overall best pricing - $62 million vs $165 million

  Largely due to the ability of recovering their own parts from the rockets - Stage 1

  Space Y wants to compete with Space X

- **Problem**

  Space Y has tasked us to train a machine learning model to predict the successful recovery of parts through - Stage 1 - Recovery

Methodology

Executive Summary

- Data collection methodology:
  - SpaceX API & Web Scraping from Wikipedia page
- Perform data wrangling
  - Classifying true landing as successful and unsuccessful
- 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 with GridSearchCV

Data Collection – API

1.Requesting rocket launch data from API

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

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

2.Converting Response to a JSON file

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

3.Using custom functions to clean data

```
# Call getBoosterVersion
getBoosterVersion(data)
```

```
# Call getPayloadData
getPayloadData(data)
```

```
# Call getLaunchSite
getLaunchSite(data)
```

```
# Call getCoreData
getCoreData(data)
```

4.Combining the columns into a dictionary to create data frame

```
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}
```

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

5.Filtering dataframe and exporting to a CSV

```
data_falcon9 = launch_df[launch_df['BoosterVersion'] == 'Falcon 9']
```

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

Data Collection – Scraping

1.Getting response from HTML

```
html_data = requests.get(static_url).text
```

2.Creating a BeautifulSoup object

```
soup = BeautifulSoup(html_data, 'html5lib')
```

3.Finding all tables and assigning the result to a list

```
html_tables = soup.find_all('table')
```

4.Extracting column name one by one

```
column_names = []
for row in first_launch_table.find_all('th'):
    name = extract_column_from_header(row)
    if(name != None and len(name) > 0):
        column_names.append(name)
```

5.Creating an empty dictionary with keys

```
launch_dict= dict.fromkeys(column_names)
# Remove an irrelvant column
del launch_dict['Date and time ( )']
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']=[]
```

6.Filling up the `launch_dict` with launch records
(Too long to put in here, so please refer to the notebook)

7.Creating a Dataframe and exporting it to a CSV

```
df=pd.DataFrame(launch_dict)
df.to_csv('spacex_web_scraped.csv', index=False)
```

Data Wrangling

- Several instances the booster didn't correctly land on the dataset, and every now and then it tried to land however failed due to accidents.
- Creating training labels with landing outcomes:

  1 = successful / 0 = failure

  -True ASDS, True RTLS, True Ocean - set to ->1

  -False ASDS, None ASDS, False Ocean, False RTLS - set to ->0

Data Wrangling

1.Calculating the number of launches at each site

```
df['LaunchSite'].value_counts()
```

2.Calculating the number and occurrence of each orbit

```
df.Orbit.value_counts()
```

3.Calculating the number and occurrence of mission outcome per orbit type

```
landing_outcomes = df.Outcome.value_counts()
```

4.Creating a landing outcome label from Outcome column

```
landing_class = []
for outcome in df.Outcome:
    if outcome in bad_outcomes:
        landing_class.append(0)
    else:
        landing_class.append(1)
```

```
df['Class']=landing_class
```

5.Calculating the success rate for every landing in dataset

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

```
0.6666666666666666
```

6.Exporting dataset to a CSV

```
df.to_csv("dataset_part_2.csv", index=False)
```

EDA with Data Visualization

- Scatter chart:
  - *Flight Number vs. Launch Site*
  - *Payload vs. Launch Site*
  - *Flight Number vs. Orbit Type*
  - *Payload vs. Orbit Type*
- Bar chart:
  - *Orbit Type vs. Success Rate*
- Line chart:
  - *Year vs. Success Rate*

EDA with SQL

- Loaded data into IBM Db2 database and queried using SQL Python. Following questions:

  - Names of the unique launch sites in the space mission
  - 5 records where the launch sites begin with the string 'CCA'
  - Total payload mass carried by boosters launched by NASA (CRS)
  - Average payload mass carried by booster version F9 v1.1
  - Date when the first successful landing outcome in-ground pad was achieved
  - Names of the boosters with success in drone ship and have payload mass - >4000 & <6000
  - Total number of successful / failure mission outcomes
  - Names of the booster_versions that have carried the maximum payload mass
  - Failed landing_outcomes in drone ship, their booster versions, and launch site names in the year 2015
  - Ranking the count of landing outcomes or Success between the date of 2010-06-04 and 2017-03-20 in descending order.

Build an Interactive Map with Folium

- Folium Map:

  - Markers showing all launch sites and the the success/failed launches for each site on the map
  - Distances between a launch site and its surroundings.
- This allows us to follow the geographical patterns about the launch sites.

Build a Dashboard with Plotly Dash

- Dashboard contains a pie chart and scatter point chart.
  - Pie chart
    - Will show the total success launches by site.
    - Will indicate successful landing distribution across all launch sites.
  - Scatter chart
    - Will show success variation across booster version category, payload mass, and launch sites.
    - Will show all sites/individual site & Payload mass on a slider between 0 and 10000 kg.

Predictive Analysis (Classification)

- Data Analysis and Training Labels.

  Class for column

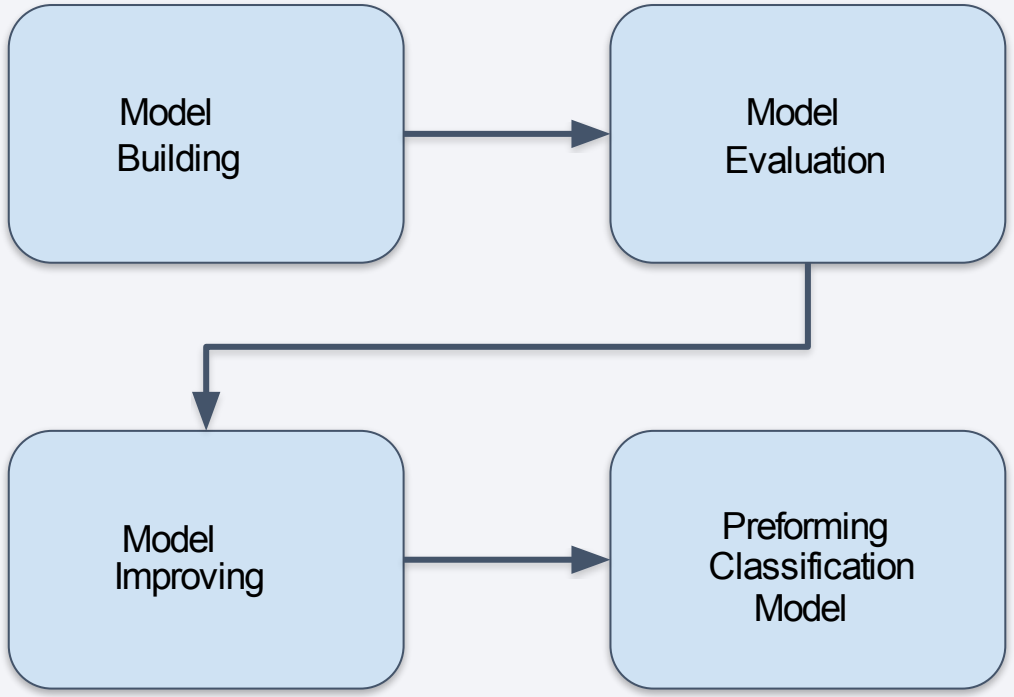
  Standardize the data

  Split into training data and test

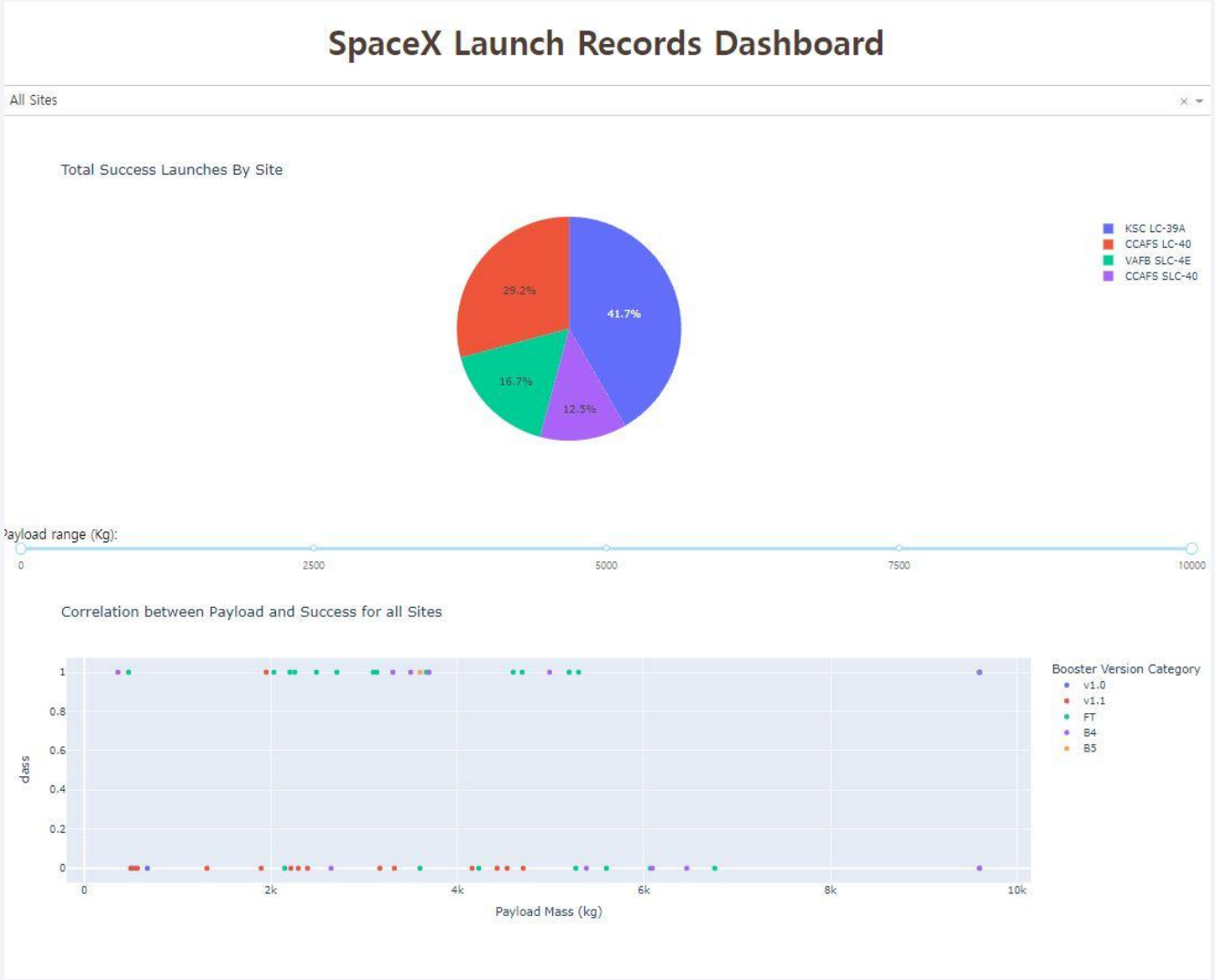
- Classifications :

  Trees

  Logistic Regression

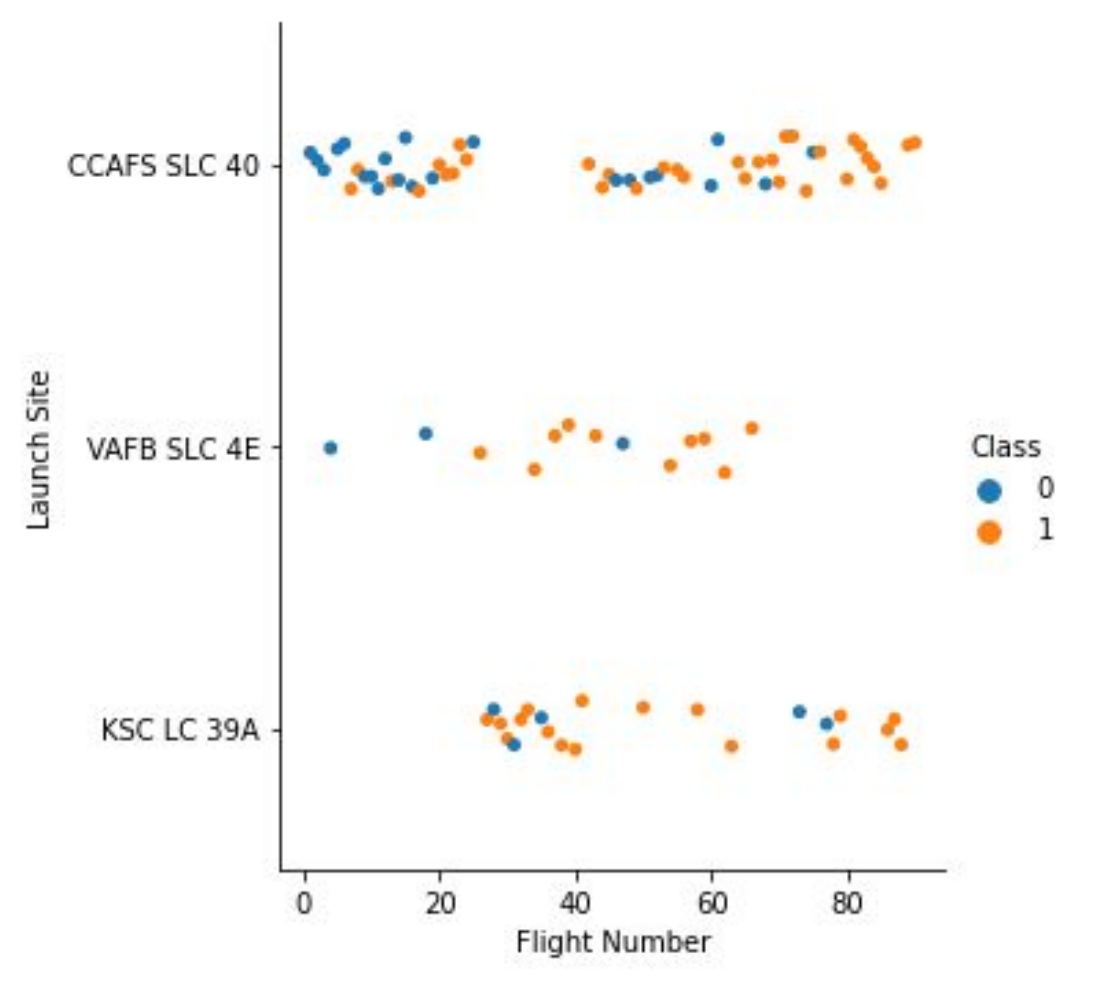


Results



- Plotly Dashboard can be seen to the left.
- EDA results demonstrated with visualization, SQL, and an Interactive Map with Folium.
- All four methods return with 83% of the test data being acurate.

Flight Number vs. Launch Site

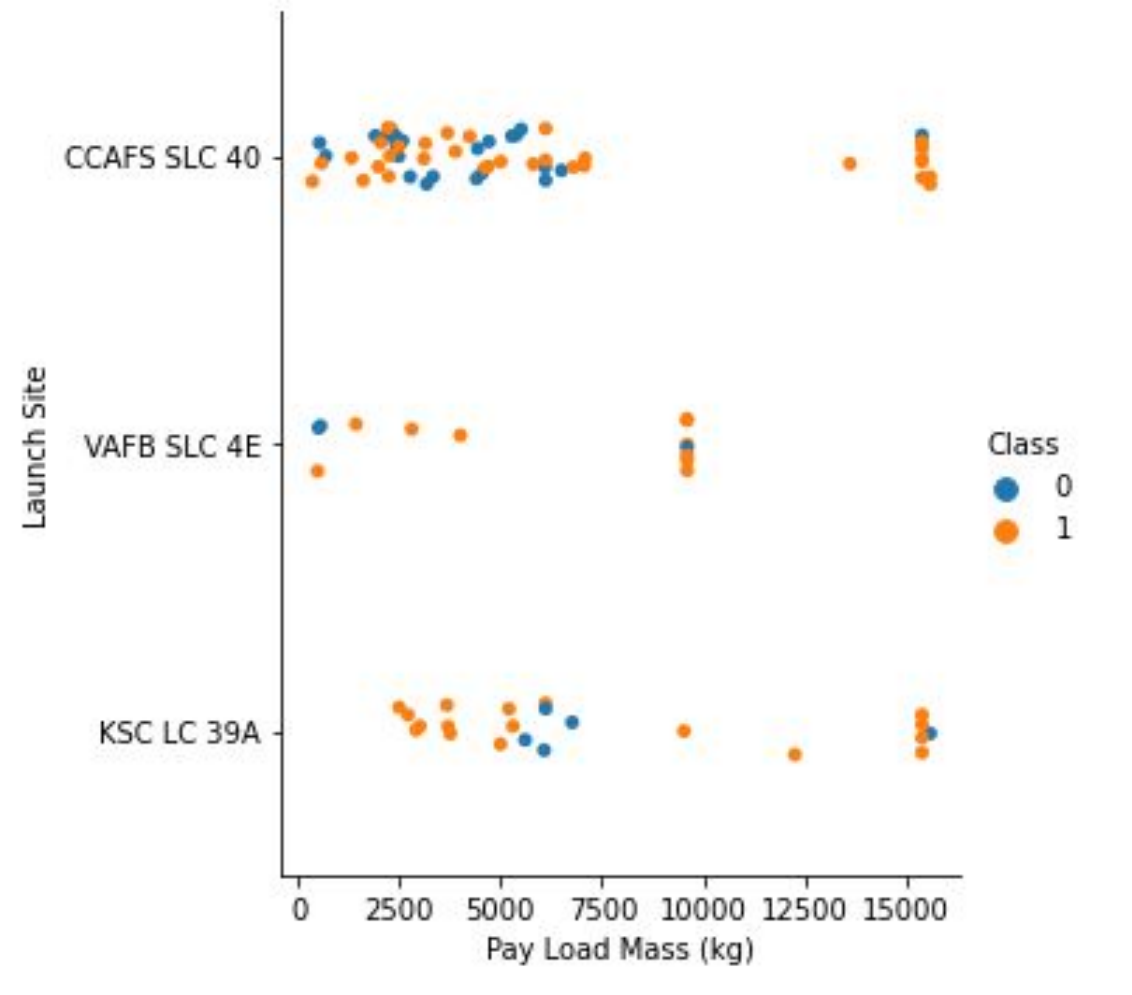


- Class representations:
  0 (blue) 1 (orange)

- Success rate followed in correlation **with the flights increased.**

- CCAFS appears to be the primary launch site due to the amount of volume.

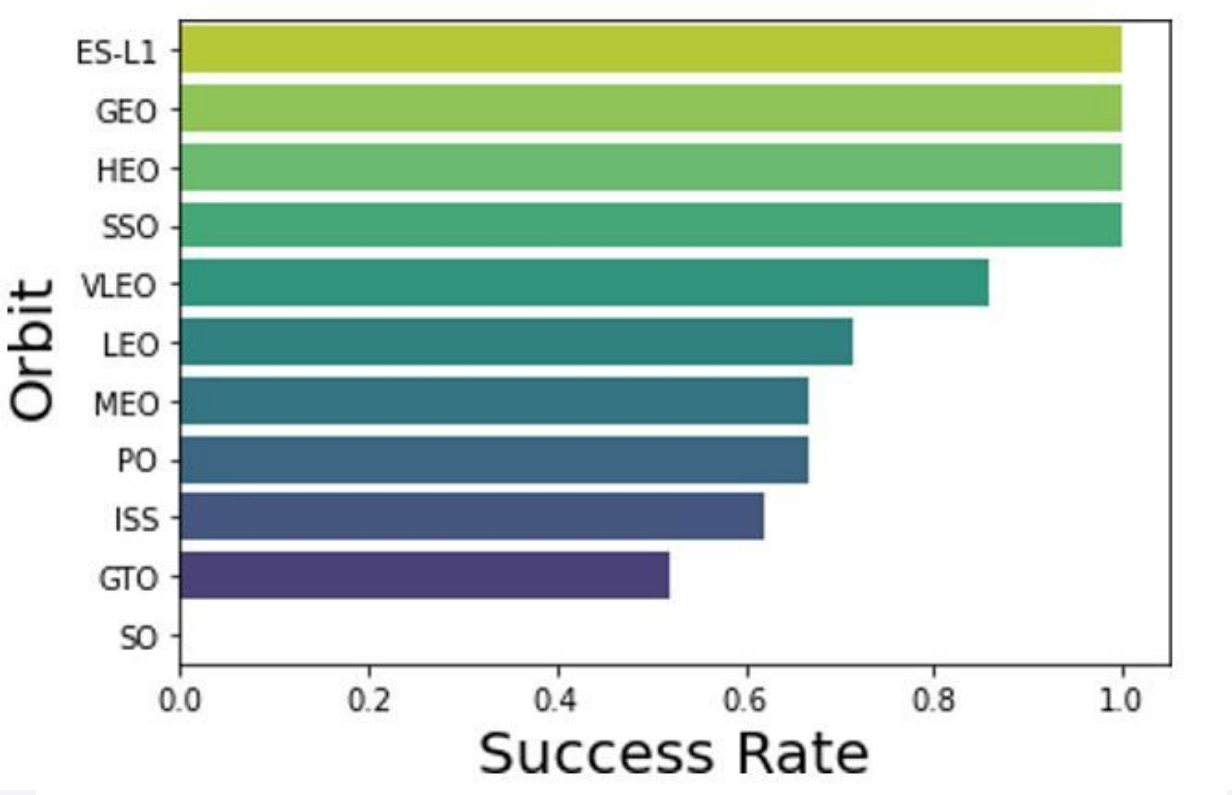
Payload vs. Launch Site



- Class representations:
  0 (blue) 1 (orange)

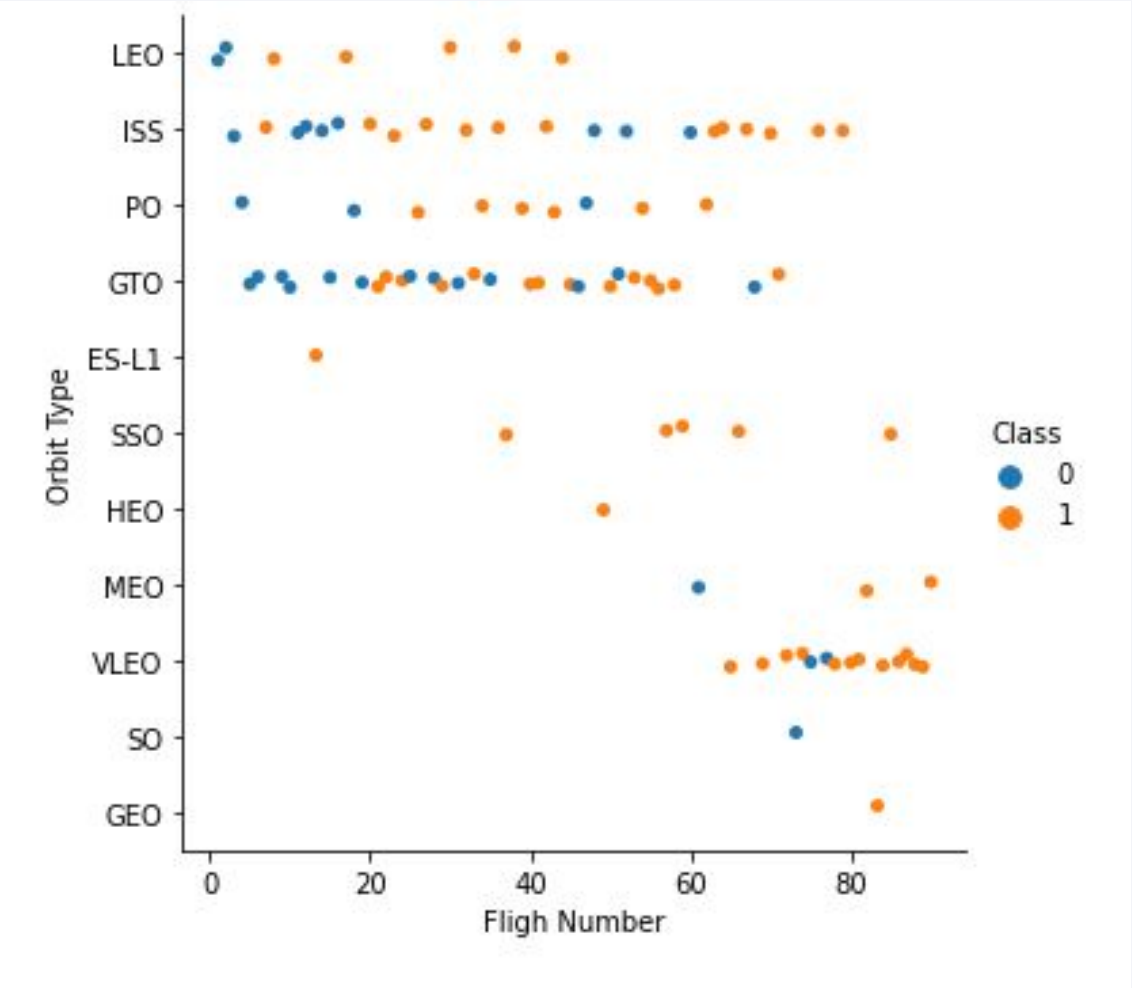
- No clear patterns could be found between successful and pay load mass.

Success Rate vs. Orbit Type



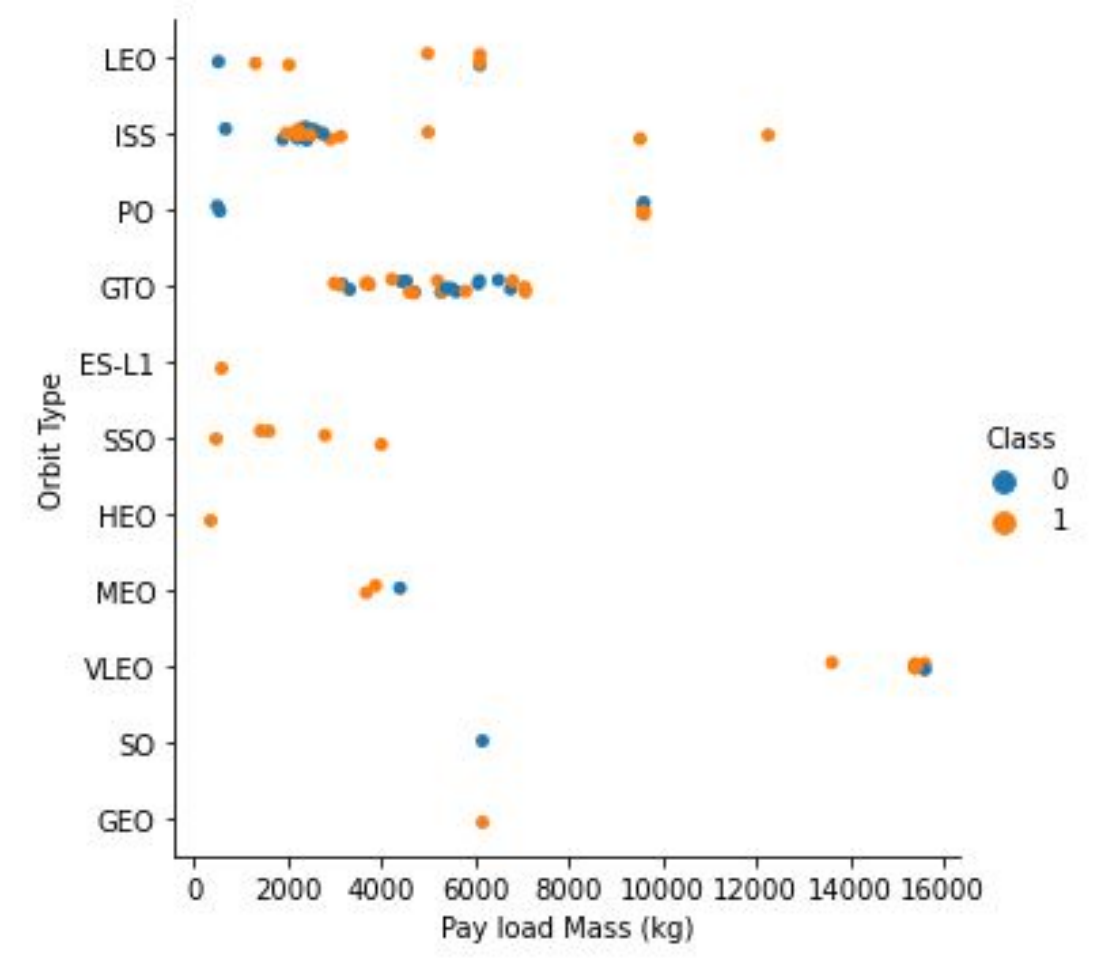
- SSO, HEO, GEO, and ES-L1 **have the highest success rates.**
- GTO is the lowest except for SO. (GTO is 50%-SO failed on single attempt)

Flight Number vs. Orbit Type



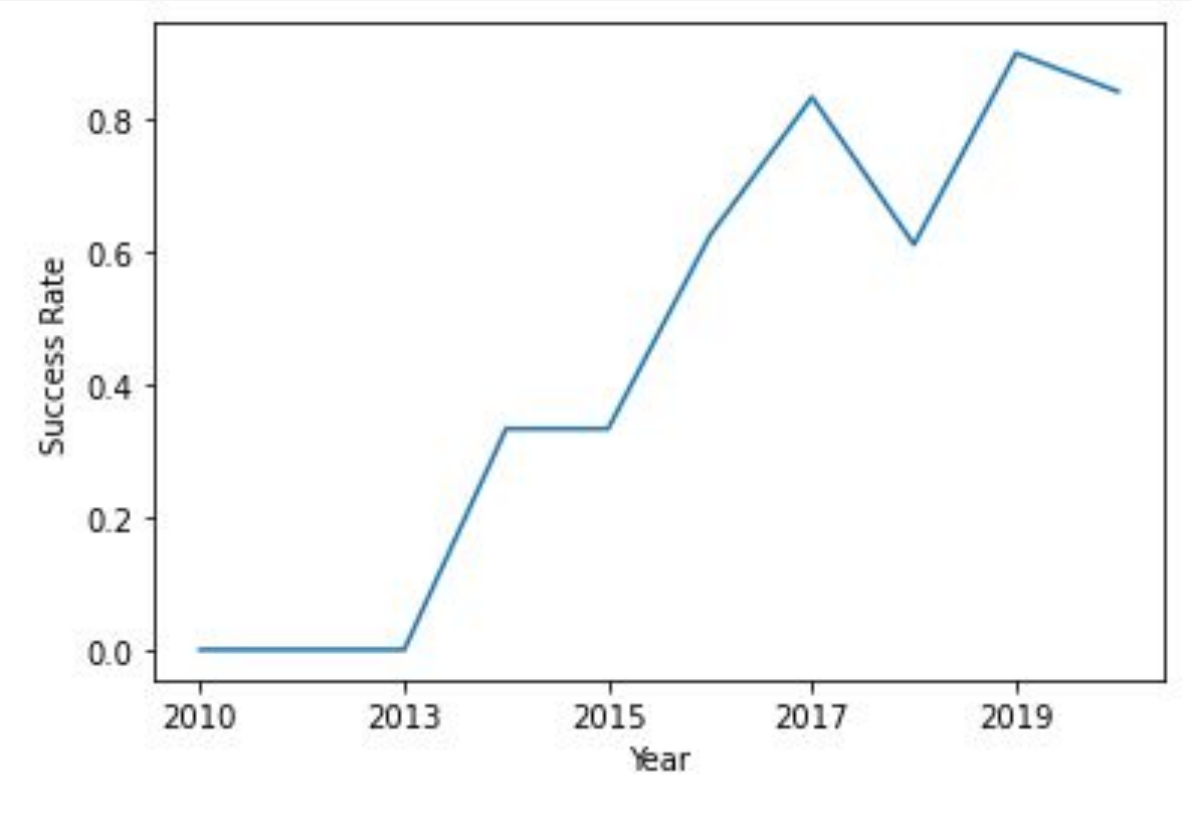
- Class representations:
  0 (blue) 1 (orange)
- Launch outcomes seem to correlated with flight numbers.
- No relationship has been found between GTO Orbit's flight and success variables.
- VLEO has been used the most in recent launches.

Payload vs. Orbit Type



- Class representations:
  0 (blue) 1 (orange)
- Heavy payloads show successful landing rates for LEO and ISS.
- GTO proves difficult to distinguish. Due to the positive and negative landing landing rates being muddled together.

Launch Success Yearly Trend



- Success rate has continued to increase up until 2017.
- Slight dip around 2018.
- Success rate is around 80%

All Launch Site Names

- Query

```
%%sql
SELECT UNIQUE LAUNCH_SITE
FROM SPACEXDATASET;

 * ibm_db_sa://ftb12020:***@0c77d6f2
Done.
```

- Result

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

- Query is used for unique launch site names from database.

- Four unique launch sites: CCAFS LC-40, CCAFS SLC-40, KSC LC-39A, VAFB SLC-4E

Launch Site Names Beginning with 'CCA'

- Query

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

- Result

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

- Only five records displayed from table.

- Together Launch_site name CAA could be called.

Total Payload Mass - NASA

- Query

```
%%sql
SELECT SUM(PAYLOAD_MASS__KG_) AS SUM_PAYLOAD_MASS_KG
FROM SPACEXDATASET
WHERE CUSTOMER = 'NASA (CRS)';
```

```
 * ibm_db_sa://ftb12020:***@0c77d6f2-5da9-48a9-81f8-86
Done.
```

- Result

total_payload_mass_kg
45596

- Query sums the total payload mass in kg.
- Filtered to calculate only if the customer is NASA.

Average Payload Mass by F9 v1.1

- Query

```
%%sql
SELECT AVG(PAYLOAD_MASS__KG_) AS AVG_PAYLOAD_MASS_KG
FROM SPACEXDATASET
WHERE booster_version = 'F9 v1.1'
```

```
 * ibm_db_sa://ftb12020:***@0c77d6f2-5da9-48a9-81f8-8
Done.
```

- Result

avg_payload_mass_kg
2928

- Query calculates the average payload mass or launch (booster version F9v1.1)

First Successful Ground Landing Date

- Query

```
%%sql
SELECT MIN(DATE) AS FIRST_SUCCESS
FROM SPACEXDATASET
WHERE landing__outcome = 'Success (ground pad)';
```

```
 * ibm_db_sa://ftb12020:***@0c77d6f2-5da9-48a9-81
Done.
```

- Result

first_successful_landing_date
2015-12-22

- Query returns first successful ground pad landing date.
- Dataset is filtered to search only if Landing_outcome is success.

Successful Drone Ship Landing with Payload Between 4000 and 6000

- Query

```
%%sql
SELECT booster_version
FROM SPACEXDATASET
WHERE landing__outcome = 'Success (drone ship)' AND payload_mass__kg_ BETWEEN 4001 AND 5999;
```

```
 * ibm_db_sa://ftb12020:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databas
Done.
```

- Result

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

- Query returns the booster version and payload mass between 4000 and 6000.

Total Successful and Failure Mission Outcomes

- Query

```
%%sql
SELECT mission_outcome, COUNT(*) AS no_outcome
FROM SPACEXDATASET
GROUP BY mission_outcome;
```

```
 * ibm_db_sa://ftb12020:***@0c77d6f2-5da9-48a9-
Done.
```

- Result

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

- COUNT() function is used to calculate the total number of columns.
- GROUP BY statement, is used to find the total number.
- SpaceX successfully completed 99% of missions.

Maximum Payload Boosters

- Query

```
%%sql
SELECT booster_version, PAYLOAD_MASS__KG_
FROM SPACEXDATASET
WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXDATASET);
```

```
 * ibm_db_sa://ftb12020:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1
Done.
```

- Subquery, first, then the MAX() function, and second, filter dataset to perform a search if PAYLOAD_MASS_KG_ is the maximum value.

- Version F9 B5 B10xx.x could have carried a maximum payload.

- Result

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

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

- Result

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

- WHERE clause, filters the dataset to perform a search if Landing__outcome is `Failure (drone ship)`.

  The AND operator is used to display a record if additional condition YEAR is `2015`.

- Two failures on drone ships during 2015 were found.

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

- Query

```
%%sql
SELECT landing__outcome, COUNT(*) AS no_outcome
FROM SPACEXDATASET
WHERE landing__outcome LIKE 'Succes%' AND DATE BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY landing__outcome
ORDER BY no_outcome DESC;
```

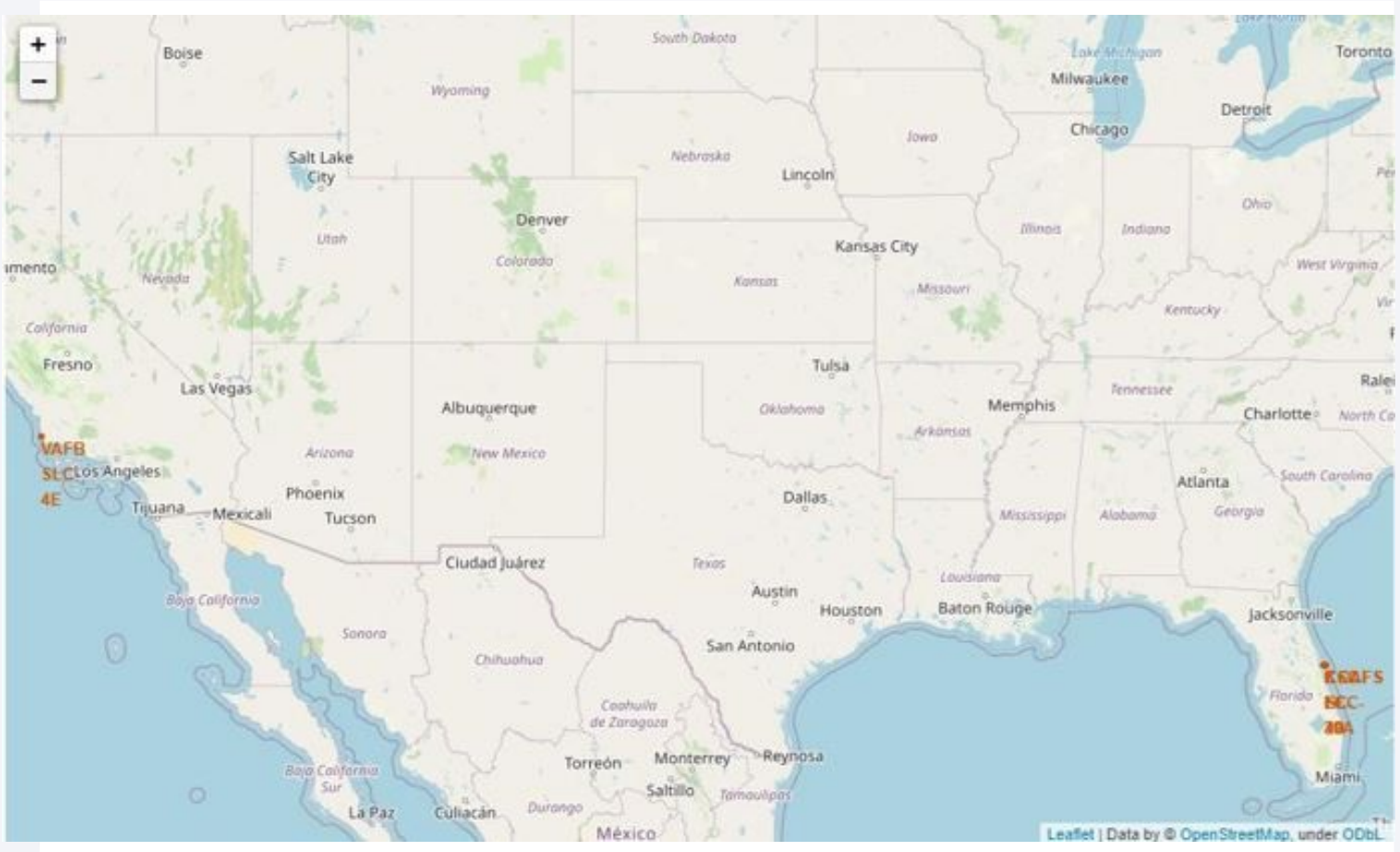
```
 * ibm_db_sa://ftb12020:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg
```

- WHERE clause, filters the dataset to perform a search if the date is between `2010-06-04` and `2017-03-20`.
- ORDER BY is used to sort the records by total number of landing, and DESC to sort the records in descending order.
- Successes and failures between 2010-06-04 and 2017-03-20 as similar.

- Result

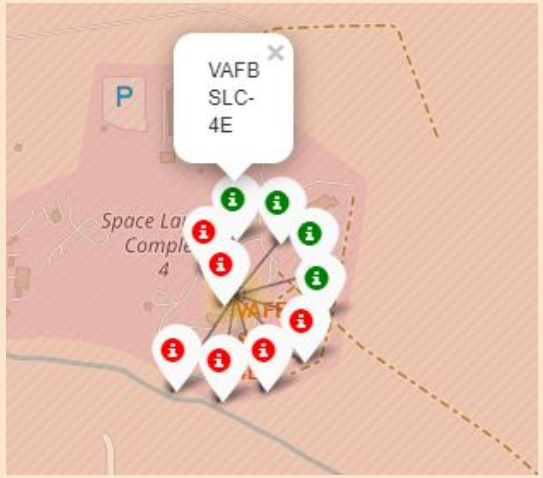
landing__outcome	total_number
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

Launch Site Locations



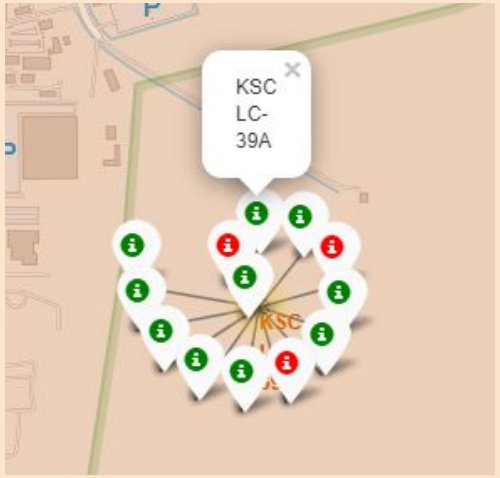
- Left map shows all SpaceX *launch sites.*
- Right map shows all launch sites being in the US.
- All launch sites are near the coast.

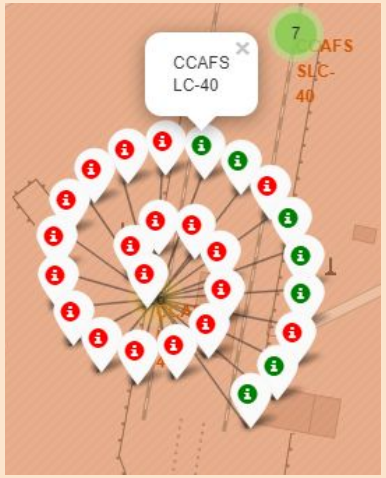
Color - Label Launch Outcomes

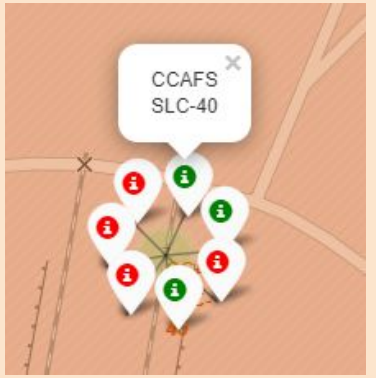


Launch site in California

- Displayed are the marker clusters:
  successful landing (green)
  failure landing (red)







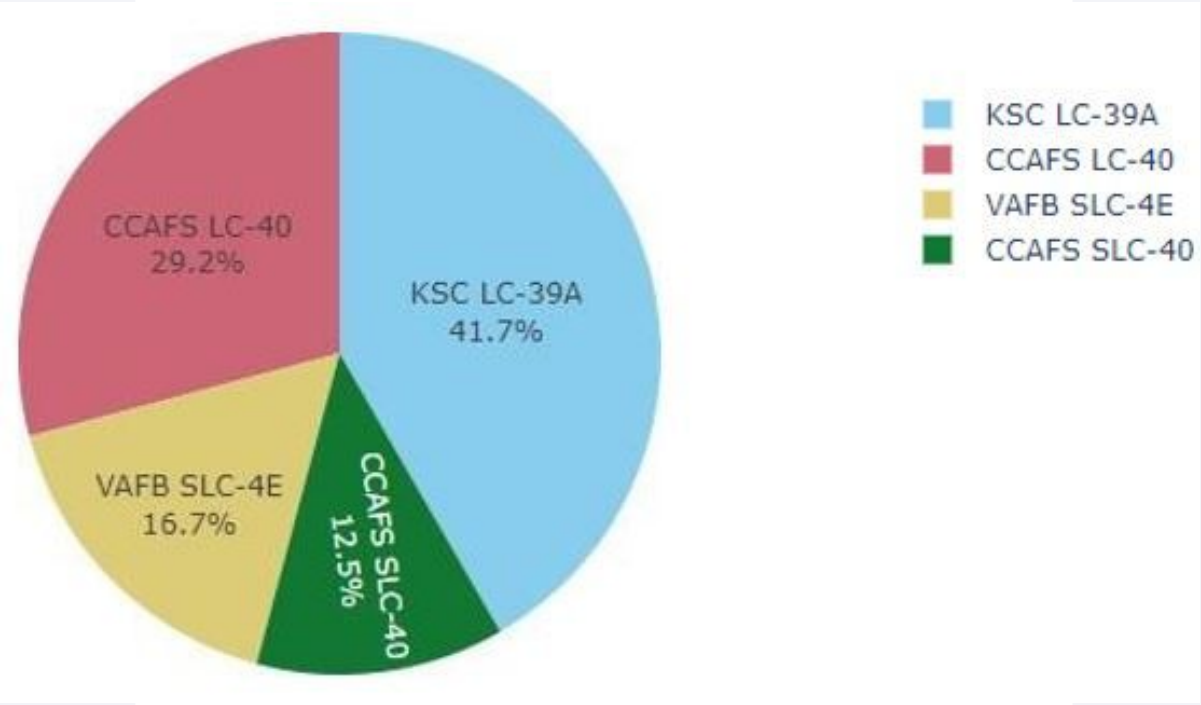
Launch sites in Florida

Proximity of Launch Sites

Are launch sites in close proximity to railways? ***Yes***
Are launch sites in close proximity to highways? ***Yes***
Are launch sites in close proximity to coastline? ***Yes***

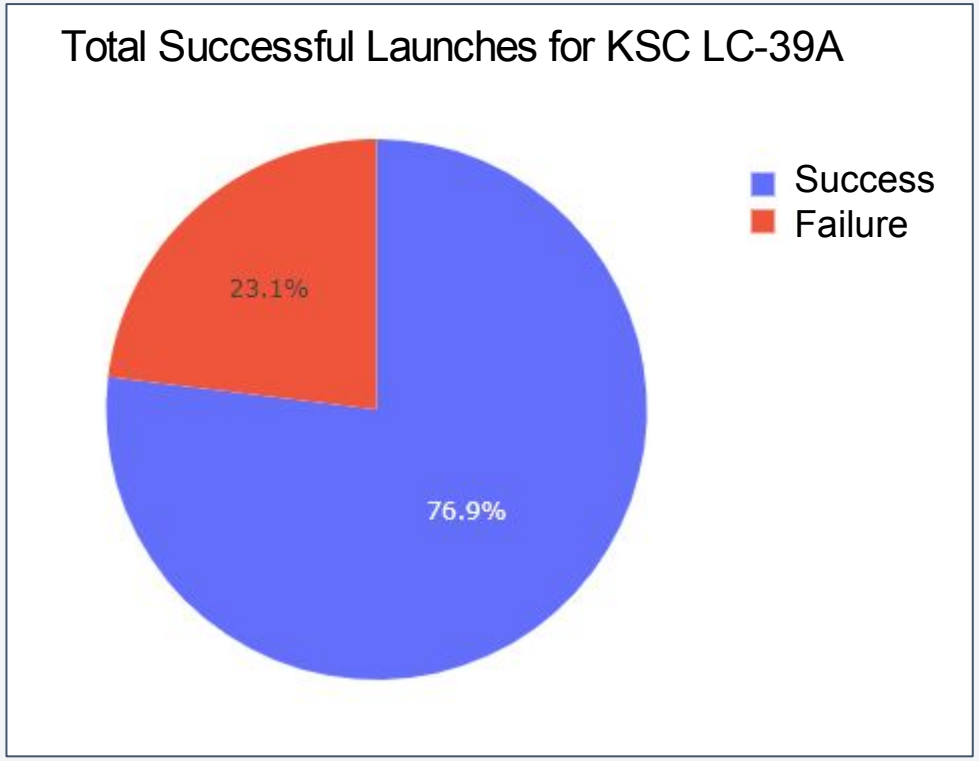
- The launch site are close to railways and highways and also close to the coastline. Making them relatively far from the cities and not posing a threat due to launch failures.

Total Success Launches - All Sites



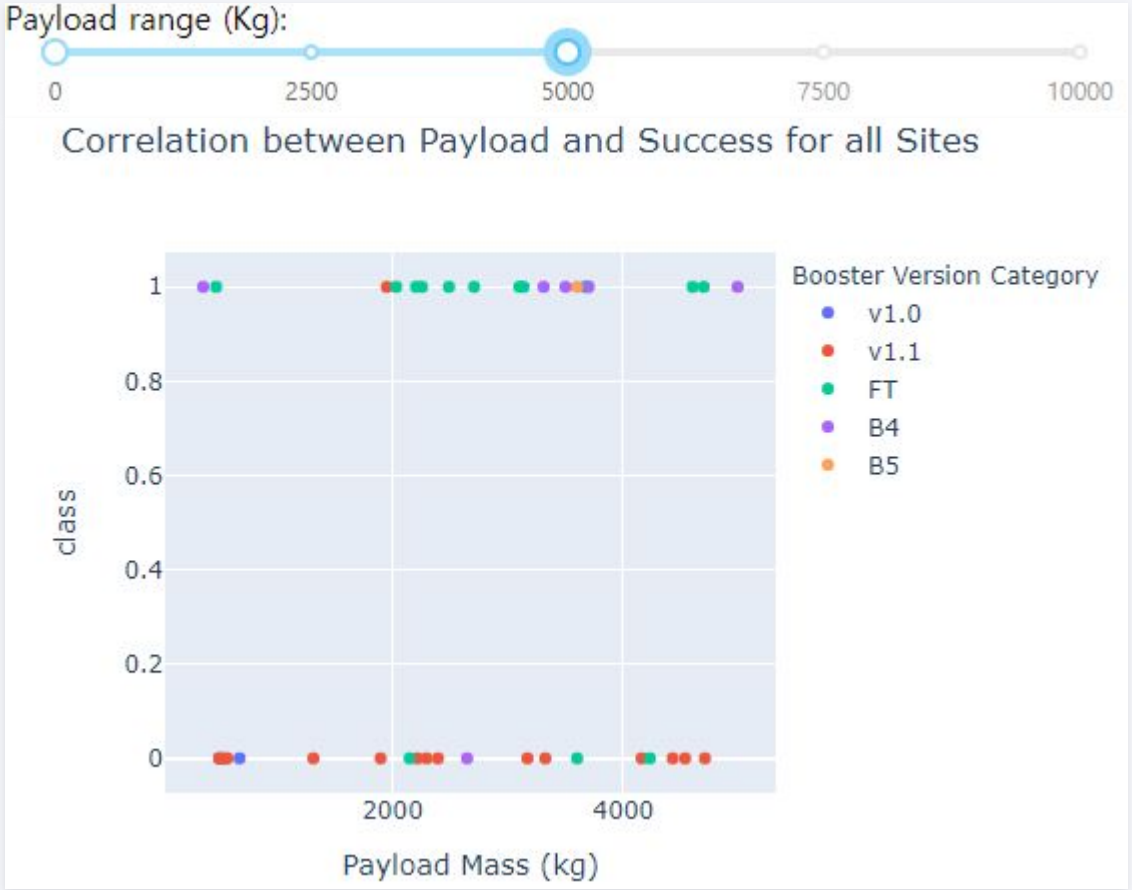
- KSLC-39A Showed the highest launch success among all sites.
- The VAFB SLC-4E had the fewest launch success.

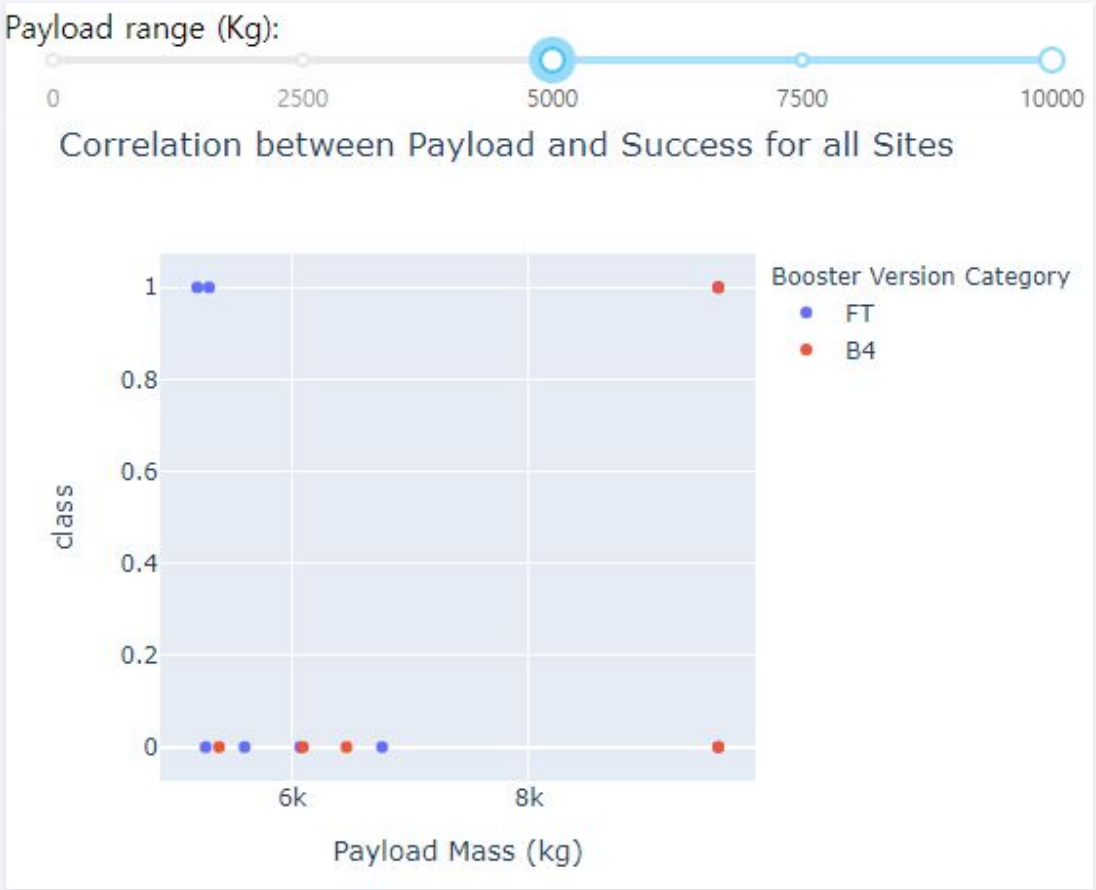
Launch Site with Highest Success Ratio



- KSLC-39A The highest success rate with 10 successful landings (76.9%) and 3 landing failures (23.1%).

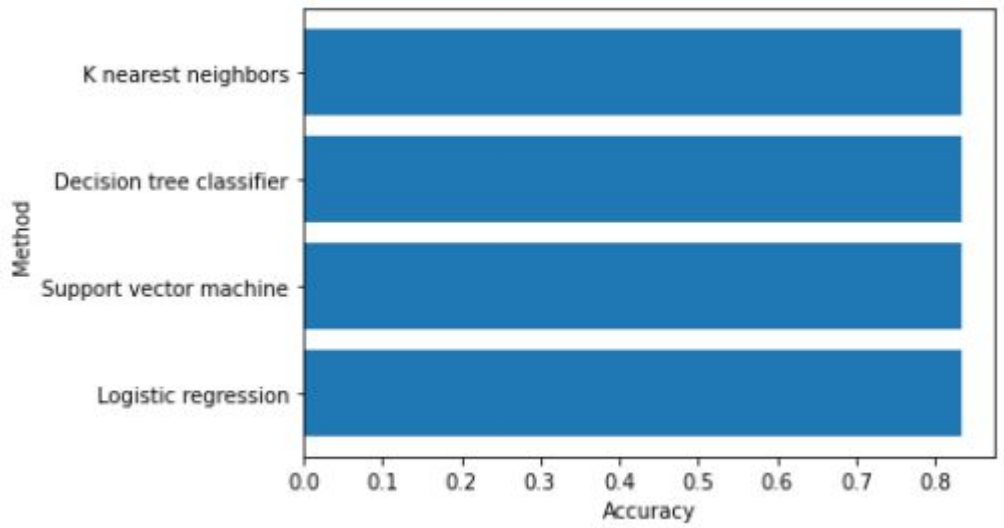
Payload vs. Launch Outcome Scatter Plot - All Sites





- These figures depict the success rate (class 1) for low weighted **payloads*(0-5000 kg)* being higher than that of heavy weighted payloads*(5000-10000 kg)***

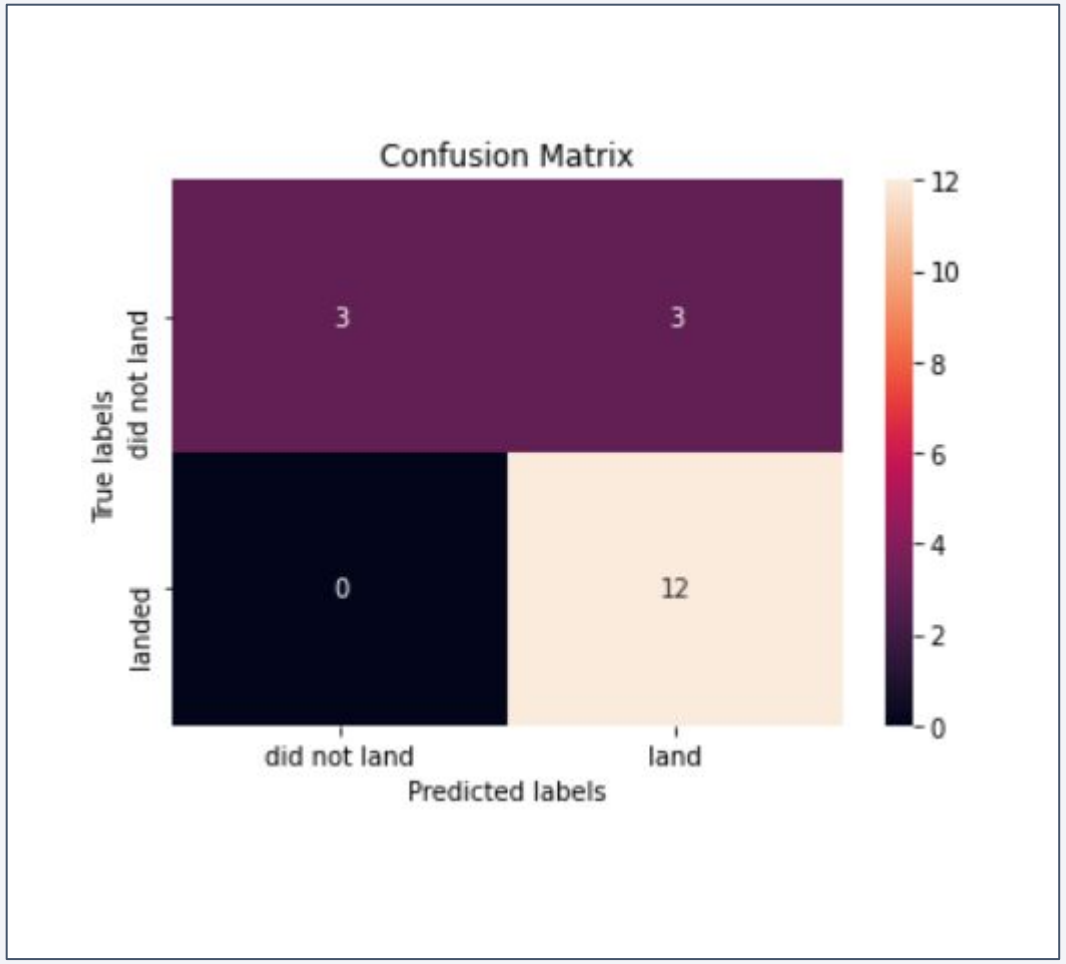
Classification Accuracy



	Method	Accuracy
0	Logistic regression	0.833333
1	Support vector machine	0.833333
2	Decision tree classifier	0.833333
3	K nearest neighbors	0.833333

- The accuracy of all models was v**irtually the same at 83.33%**
- It should be noted that the test size was small at 18.
- More data would be needed to determine the optimal model.

Confusion Matrix



- The confusion matrix is the same for all models. All models performed the same for test.
- The models predicted 12 successful landings and 3 failed landings.
- 3 predictions marked successful when the True label was failure (false positive)

Conclusion

- As the number of flights increased, the success rate increased.
- KSLC-39A has the highest number of launch success among all sites.
- The launch site is close to railways, highways, and coastline, but far from cities.
- All models marked same accuracy (83.33%)
- The success rate of low weight payloads is higher than that of the heavy weighted payloads.
- More data should be collected to better determine the learning model.