IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Terence Chow Tak Chuen
November 30, 2021

Outline

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

Executive Summary

- **Summary of methodologies**
  1. Data Collection with API
  2. Data Collection with Web Scraping
  3. Data Wrangling
  4. EDA with SQL
  5. EDA with Data Visualization
  6. Building an Interactive Map with Folium
  7. Building an Interactive Dashboard with Plotly Dash
  8. Predictive Analysis (Classification)

- **Summary of all results**
  - Exploratory Data Analysis (EDA) – Findings
  - Interactive Map & Dashboard – Insights
  - Predictive Analysis – Results

Introduction

Project background and context

The era of commercial space travel is arriving soon with several companies striving to make space travel affordable for everyone. A frontrunner is SpaceX and one of the key reasons is the relatively inexpensive cost of their rocket launches.

SpaceX promotes Falcon 9 rocket launches with a price tag of 62 million dollars, as compared to other costlier alternatives provided by its competitors upwards of around 165 million dollars. Much of the savings can likely be attributed to the reusability of the Falcon 9 rocket's first stage via re-landing.

This project aims to study various SpaceX Falcon 9 rocket launch factors to predict if the first stage will successfully land, hence determining the overall cost of a rocket launch. This information will be useful for competing companies when bidding against SpaceX for a rocket launch.

Problems we want to find answers

- Correlations between each rocket launch variables and a successful landing outcome
- Conditions to obtain the best results and predict the best successful landing outcome

Section 1

Methodology

Methodology

Executive Summary

Data collection methodology:

- SpaceX API and Web Scraping (Falcon 9 and Falcon Heavy Launch Records from Wikipedia)

Perform data wrangling

- Convert values into training labels with the landing outcome as success or failure.

Perform exploratory data analysis (EDA) using visualization and SQL

Perform interactive visual analytics using Folium and Plotly Dash

Perform predictive analysis using classification models

- Determine best hyperparameters for Support Vector Machines (SVM), Classification Trees and Logistic Regression.

Data Collection

The data collection process includes a combination of API requests from the SpaceX API and web scraping data from a table in the Wikipedia page of SpaceX, *Falcon 9 and Falcon Heavy Launches Records*.

SpaceX API

1. Python request launch data via SpaceX API
2. SpaceX API responds with launch data in JSON file
3. Normalize data into CSV file

Web Scraping

1. Get HTML response from Wikipedia web page
2. Extract data using Python (Beautiful Soup)
3. Normalize data into CSV file

GitHub

Data Collection – SpaceX API

1. Request launch data from Space X API

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

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

2. Convert response to a JSON file

```
response_json = response.json()
df = pd.json_normalize(response_json)
```

3. Format data using custom functions

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

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

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

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

4. Create data frame by combining columns into a dictionary

```
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}
```

5. Filter data frame and export to CSV file

```
data_falcon9.to_csv('dataset_part\_1.csv', index=False)
```

GitHub

Data Collection – Web Scraping

1. Get response from HTML

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

2. Create a Beautiful Soup object

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

3. Find table and extract column names

```
# Assign the result to a list called `html_tables`
html_tables = soup.find_all('table')
```

```
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)
```

4. Create empty dictionary and append with launch records data

```
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']=[]
```

5. Create data frame and export to CSV file

```
df=pd.DataFrame(launch_dict)
```

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

GitHub

Data Wrangling

Dataset covers a range of landing scenarios

- True Ocean: Successful landing in a specific area of the ocean
- False Ocean: Failure landing in a specific area of the ocean
- True RTLS: Successful landing on the ground pad
- False RTLS: Unsuccessful landing on the ground pad
- True ASDS: Successful landing on the drone ship
- False ASDS: Unsuccessful landing on the drone ship

Conversion of results into training labels

- 1 = Success ; 0 = Failure

GitHub

Data Wrangling

1. Calculate number of launches at each site

```
# Apply value_counts() on column LaunchSite
df['LaunchSite'].value_counts()
```

2. Calculate number and occurrence of each orbit

```
# Apply value_counts on Orbit column
df.Orbit.value_counts()
```

3. Calculate number and occurrence of mission outcome per orbit type

```
# landing_outcomes = values on Outcome column
landing_outcomes = df.Outcome.value_counts()
print(type(landing_outcomes))
landing_outcomes
```

4. Create landing outcome label

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

5. Calculate success rate

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

```
0.6666666666666666
```

6. Create data frame and export to CSV file

```
df.to_csv("dataset_part\_2.csv", index=False)
```

GitHub

EDA with SQL

Dataset was loaded into a table (**SPACEXTBL**) hosted by **IBM Db2 cloud database**. The following **SQL queries** were executed to explore and analyze the dataset:

- Display the names of the unique launch sites in the space mission
- Display 5 records where launch sites begin with the string 'CCA'
- Display the total payload mass carried by boosters launched by NASA (CRS)
- Display average payload mass carried by booster version F9 v1.1
- List the date when the first successful landing outcome in ground pad was acheived
- List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
- List the total number of successful and failure mission outcomes
- List the names of the booster_versions which have carried the maximum payload mass. Use a subquery.
- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
- 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.

EDA with Data Visualization

Scatter Plot

- Flight Number vs Launch Site
- Payload vs Launch Site
- Flight Number vs Orbit Type
- Payload vs Orbit Type

Scatter plots show how much one variable is affected by another. The relationship between two variables is called a correlation. This plot is generally composed of large data bodies.

Bar Chart

- Orbit Type vs Success Rate

Bar charts make it easy to compare datasets between multiple groups. One axis represents a category, and the other axis represents a discrete value. The purpose of this chart is to indicate the relationship between the two axes.

Line Chart

- Year vs Success Rate

Line charts show data variables and trends very clearly and helps predict the results of data that has not yet been recorded.

GitHub

Build an Interactive Map with Folium

Objects created and added onto the Folium map

- Markers of all launch sites
- Markers of both success/failed launches for each site
- Lines of distances between a launch site to its proximities

Geographical features of launch sites analyzed

- Estimated proximity to railways
- Estimated proximity to highways
- Estimated proximity to coastline
- Estimated proximity to city centers

Build a Dashboard with Plotly Dash

Dashboard application consists of a pie chart and a scatter plot

Pie Chart

- Displays total success launches by sites.
- Options can be selected to indicate a successful landing distribution across all launch sites or success rates of each individual launch sites.
- Chart can be used for comparison of successful outcomes across all launch sites.

Scatter Plot

- Displays relationship between Outcome and Payload mass (kg) by different boosters.
- Options can be selected for all launch sites or individual sites, with an additional slider between 0 to 10,000 kg.
- Plot can be used to gauge the attributes of successful outcomes based on launch point, payload mass and booster version categories.

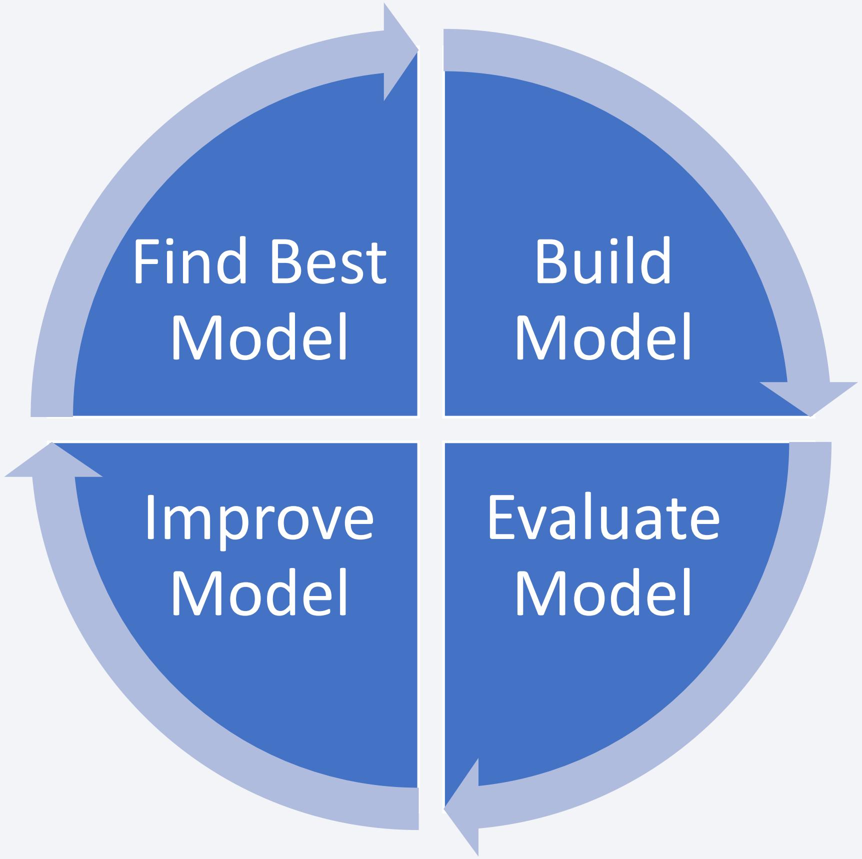
Predictive Analysis (Classification)

Perform exploratory data analysis (EDA) to determine training labels

- Create column for class
- Standardize data
- Split into training and test data

Determine best hyperparameter for SVM, Classification Trees and Logistic Regression

- Find best performing method using test data



Results

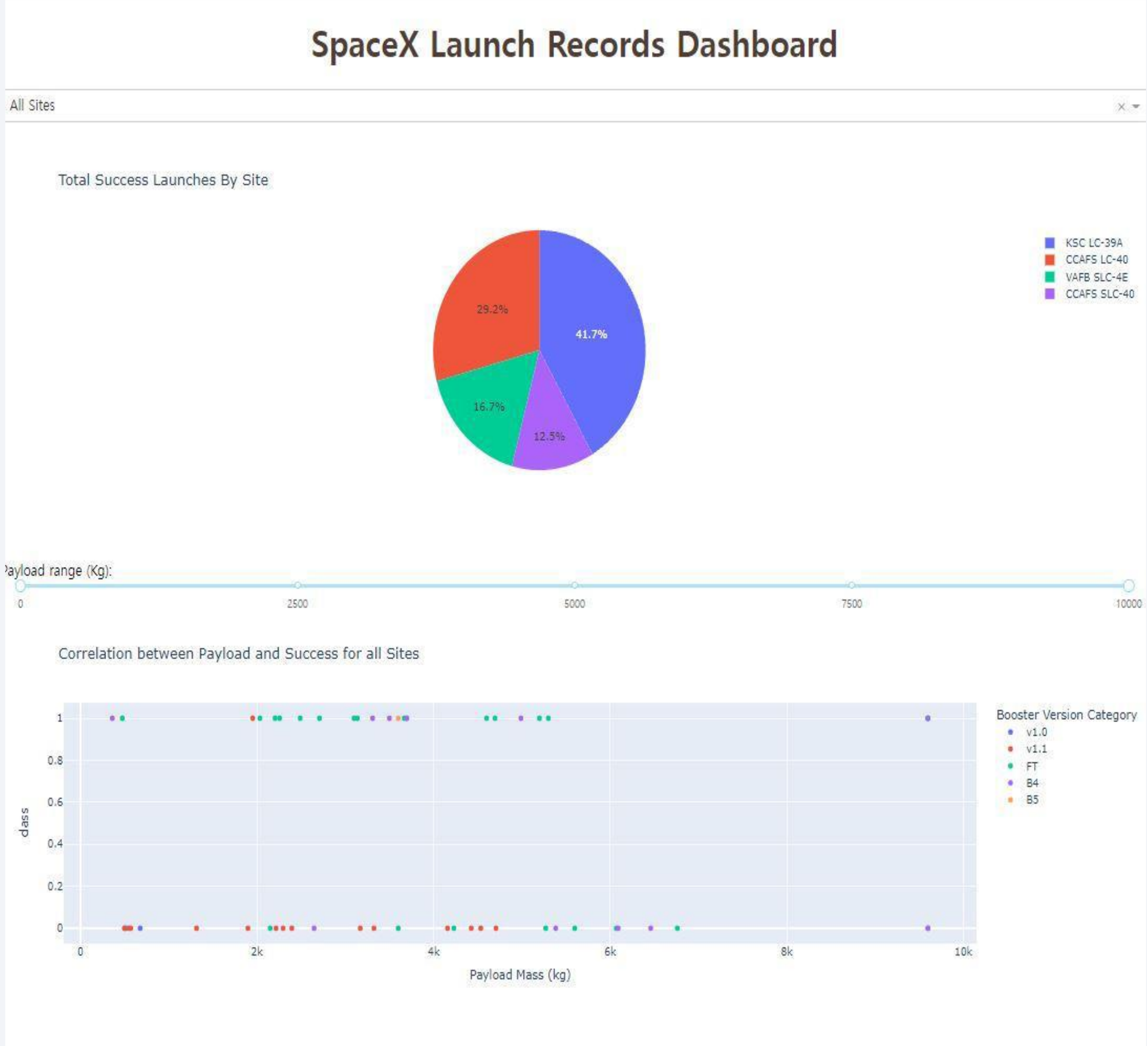
Exploratory data analysis results

Success rate has been steadily climbing up over time as the number of flights increases across all launch sites.

Several orbit types have seen complete successes and appear to cater for specific mass ranges.

Predictive analysis results

Comparing the accuracy of the four methods, all return the same accuracy of approximately 83% for test data.

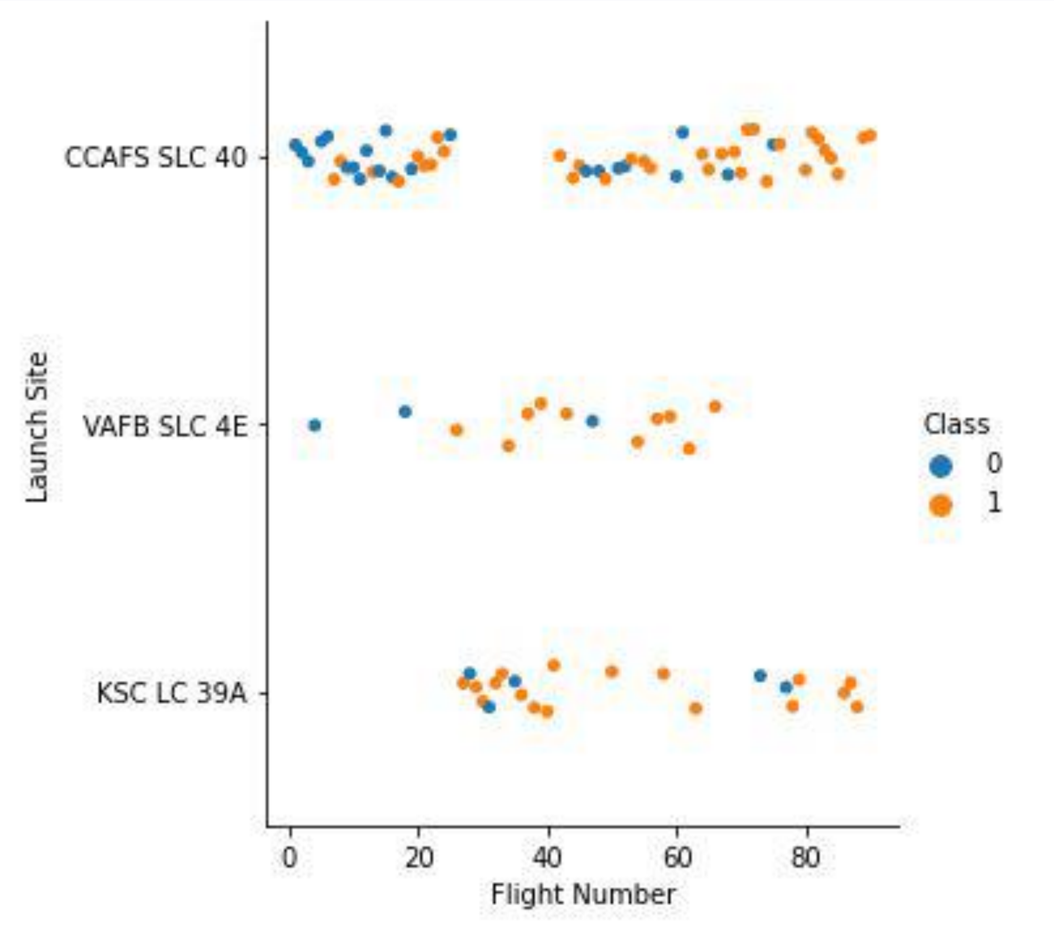


Section 2

Insights drawn from EDA

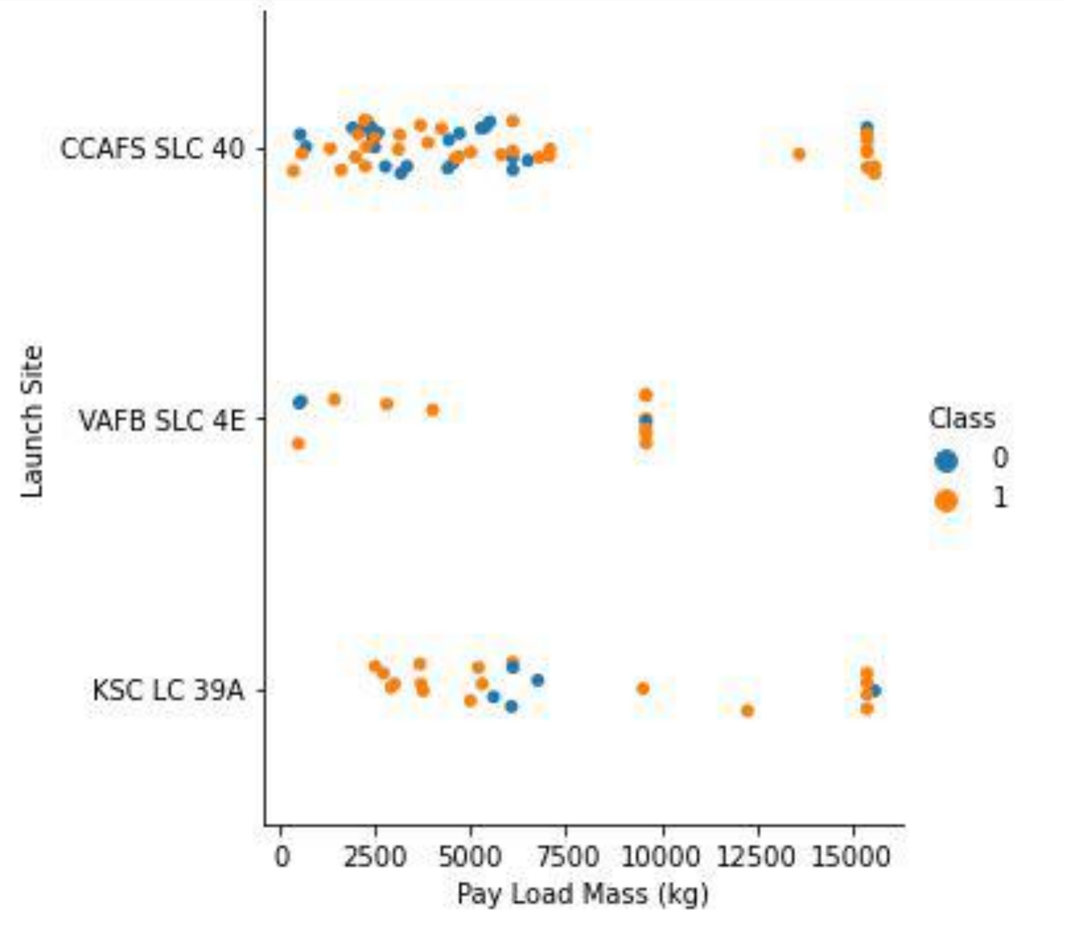
Flight Number vs. Launch Site

- Class 0 (blue) represents failed launch and Class 1 (orange) represents successful launch.
- Figure shows that success rate increased as the number of flights increased for all launch sites.



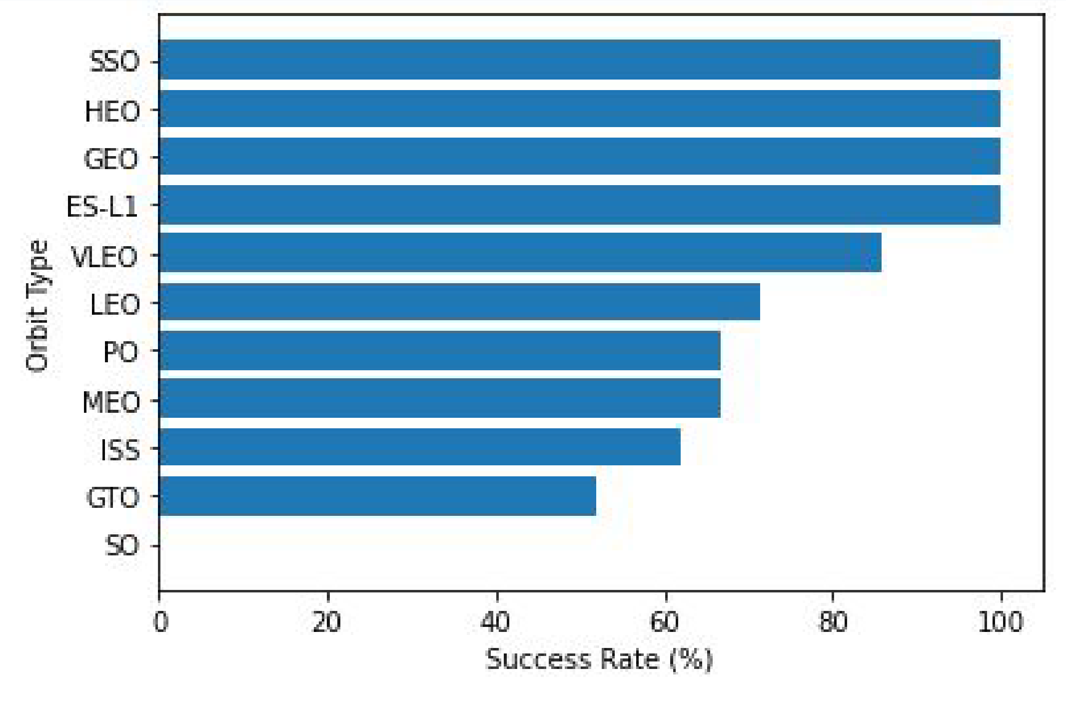
Payload vs. Launch Site

- Class 0 (blue) represents failed launch and Class 1 (orange) represents successful launch.
- Figure shows no clear pattern between success rate and pay load mass.



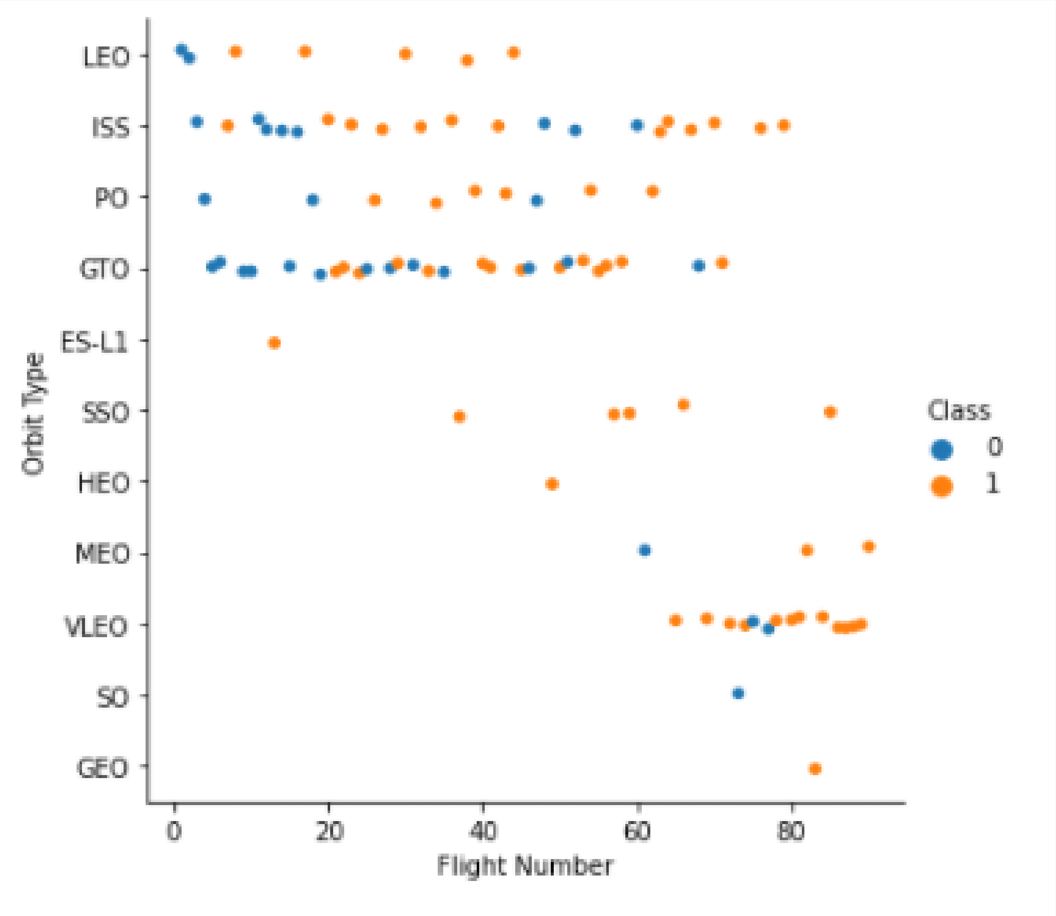
Success Rate vs. Orbit Type

- Orbit types **SSO**, **HEO**, **GEO** and **ES-L1** have the highest success rates (*100%).*
- In contrast, the success rate of orbit type **GTO** is only *50%.*
- Lowest success rate (*0%*) is observed with orbit type **SO**, recording failure in only a single attempt made.



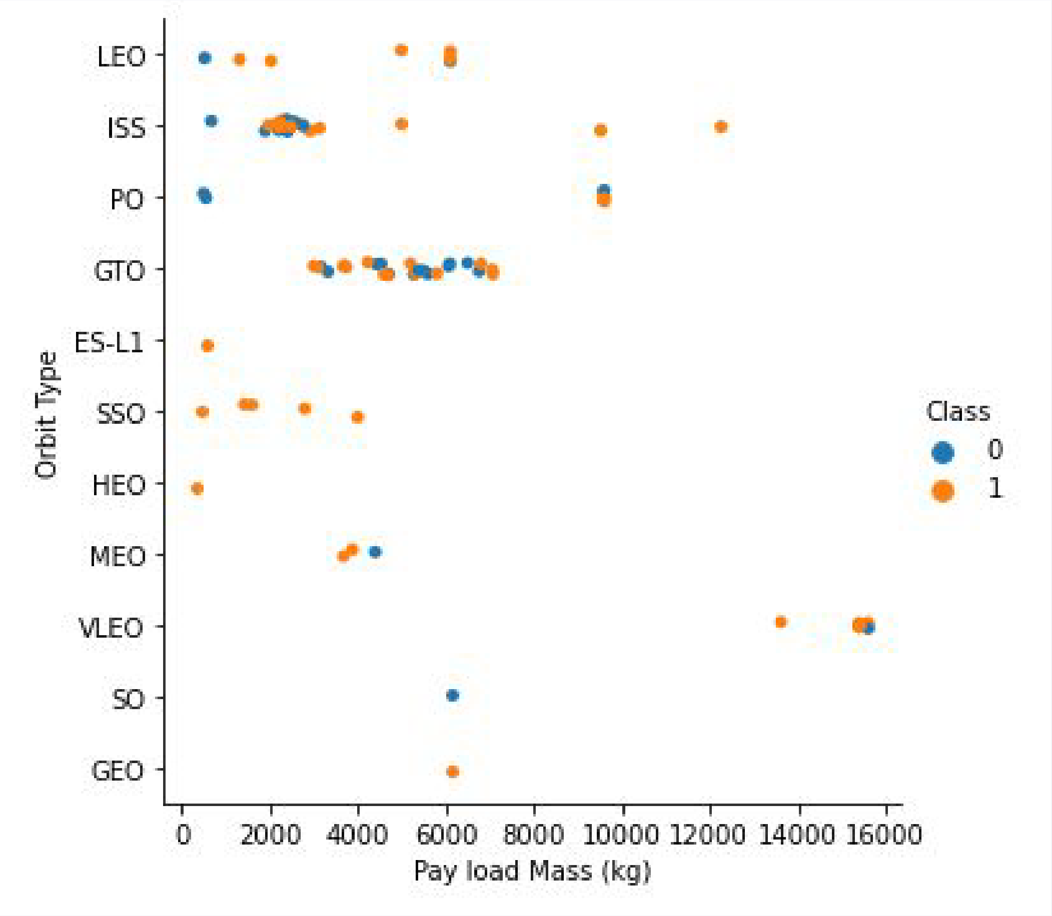
Flight Number vs. Orbit Type

- Class 0 (blue) represents failed launch and Class 1 (orange) represents successful launch.
- Figure shows that success rate generally increased with more flights.
- Gradual shift towards **VLEO** in most recent flights with high success rate.



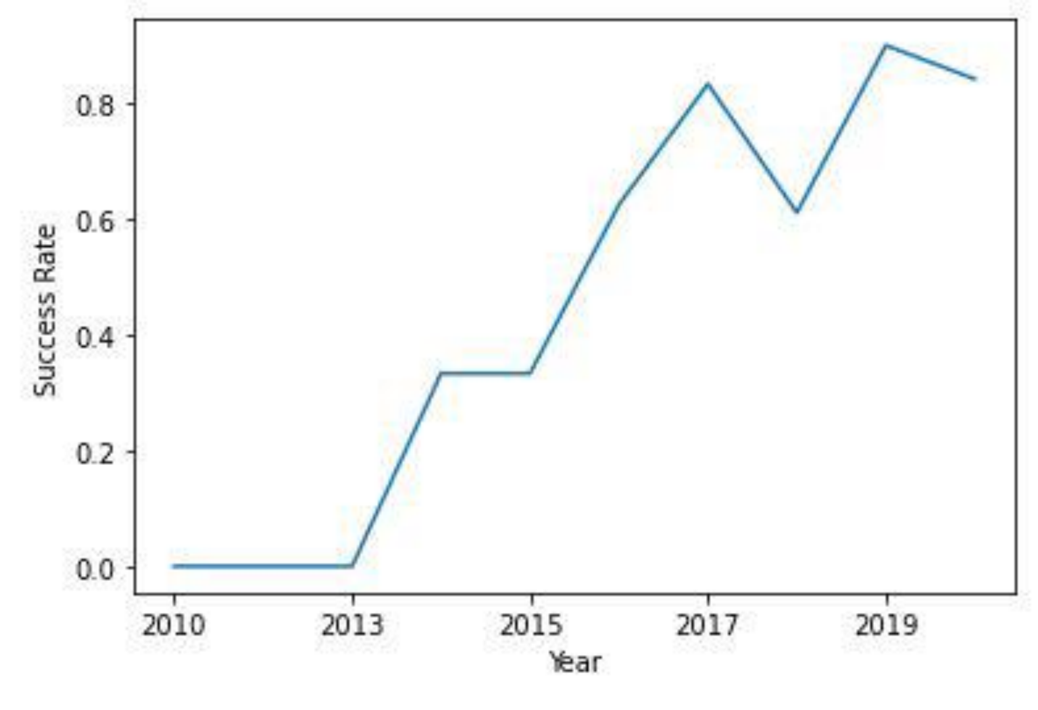
Payload vs. Orbit Type

- Class 0 (blue) represents failed launch and Class 1 (orange) represents successful launch.
- Figure shows that heaviest payloads have been successful with **VLEO** and ISS.
- Mixed results for **GTO** in the low to mid range masses.



Launch Success Yearly Trend

- Since 2013, the success rate has continued to increase steadily until **2017**.
- The success rate dipped slightly in **2018**.
- Recent data has indicated a success rate of about *80%*.



Section 3

EDA with SQL

All Launch Site Names

Query

```
%%sql
SELECT DISTINCT LAUNCH_SITE
FROM SPACEXTBL
```

Result

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

- DISTINCT clause used to query only unique values stored in the launch_site column from the SPACEXTBL table
- There are four unique launch sites.

Launch Site Names Begin with 'CCA'

Query

```
%%sql
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

- LIMIT clause used to query only the first five rows from the SPACEXTBL table
- LIKE operator and percent sign (%) filtered only for Launch_site name starting with ‘CCA’

Total Payload Mass

Query

```
%%sql
SELECT SUM(PAYLOAD_MASS__KG_) AS total_payload_mass_kg
FROM SPACEXTBL
WHERE CUSTOMER = 'NASA (CRS)'
```

Result

total_payload_mass_kg
45596

- SUM function used to calculate the total sum of column PAYLOAD_MASS__KG_
- WHERE clause filters the dataset to specify performing the calculations only if Customer is NASA (CRS)

Average Payload Mass by F9 v1.1

Query

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

Result

avg_payload_mass_kg
2928

- AVG function used to calculate the average value of column PAYLOAD_MASS__KG_
- WHERE clause filters the dataset to specify performing the calculations only if Booster_version is F9 v1.1

First Successful Ground Landing Date

Query

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

Result

first_successful_landing_date
2015-12-22

- MIN function used to find out the earliest date value in column DATE
- WHERE clause filters the dataset to specify performing the calculations only if Landing_outcome is Success (ground pad)

Successful Drone Ship Landing with Payload between 4000 and 6000

Query

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

Result

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

- WHERE clause filters the dataset to display rows only if Landing_outcome is Success (ground pad)
- AND operator specifies an additional condition of PAYLOAD_MASS__KG_ being between 4000 and 6000

Total Number of Successful and Failure Mission Outcomes

Query

```
%%sql
SELECT MISSION_OUTCOME, COUNT(*) AS total_number
FROM SPACEXTBL
GROUP BY MISSION_OUTCOME
```

Result

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

- COUNT function used to calculate the total number of columns
- GROUP BY statement groups rows that have the same values into summary rows to find out the total number in each Mission_outcome

Boosters Carried Maximum Payload

Query

```
%%sql
SELECT DISTINCT BOOSTER_VERSION, PAYLOAD_MASS__KG_
FROM SPACEXTBL
WHERE PAYLOAD_MASS__KG_ = (
    SELECT MAX(PAYLOAD_MASS__KG_)
    FROM SPACEXTBL);
```

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

booster_version	payload_mass__kg_
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

- DISTINCT clause used to query only unique values stored in the launch_site column from the SPACEXTBL table
- WHERE clause contains a nested subquery
  i. Find maximum value of the payload using MAX function
  ii. Filter dataset to perform search if PAYLOAD_MASS__KG_ is the maximum value of the payload

2015 Launch Records

Query

```
%%sql
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 display rows only if Landing_outcome is Failure (drone ship)
- AND operator specifies an additional condition of Year (Date) being 2015

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

Query

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

Result

landing__outcome	total_number
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3

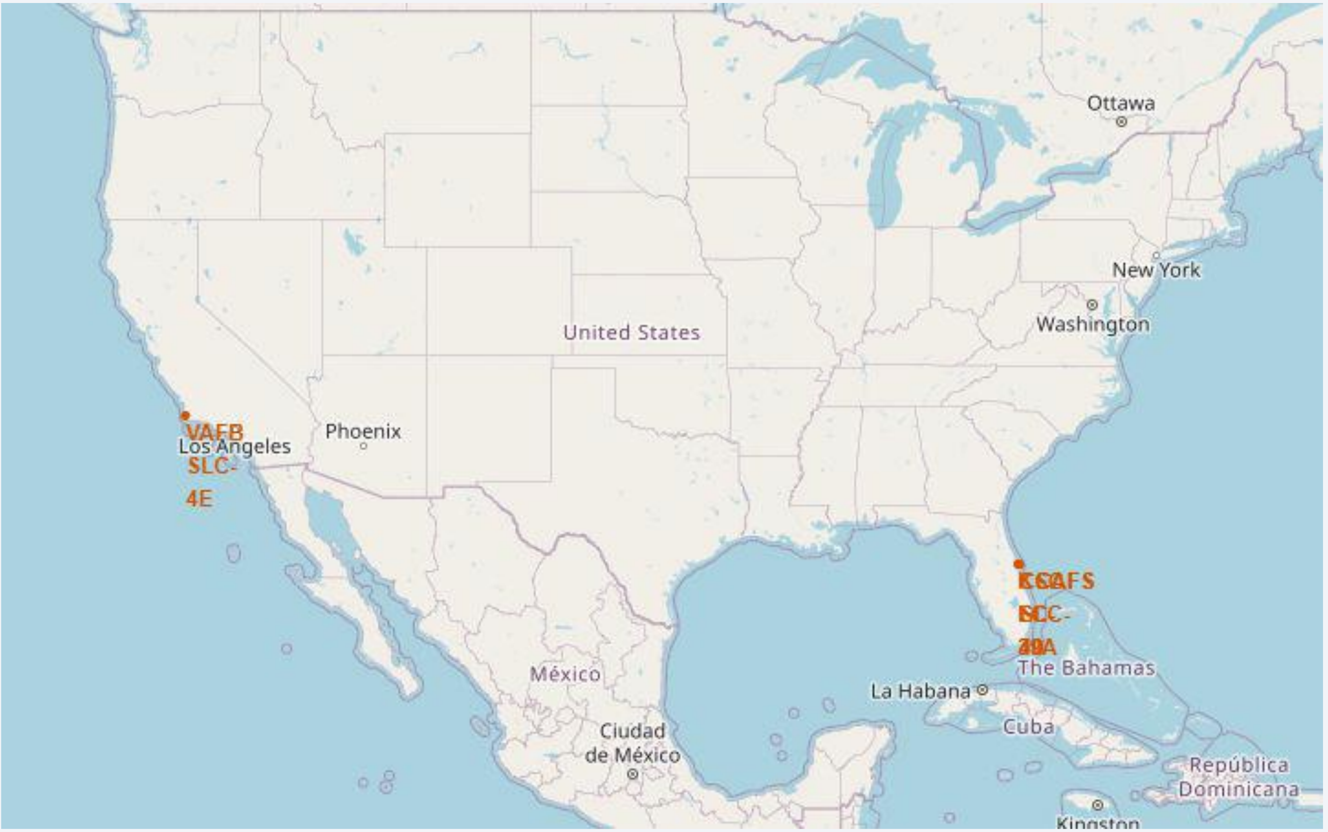
landing__outcome	total_number
Success (ground pad)	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

- WHERE clause filters the dataset to display rows only if Date is between 2010-06-04 and 2017-03-20
- GROUP BY statement groups rows that have the same values in Landing_outcome column into summary rows
- ORDER BY statement sorts values in the total_number column by descending order using DESC keyword

Section 4

Launch Sites Proximities Analysis

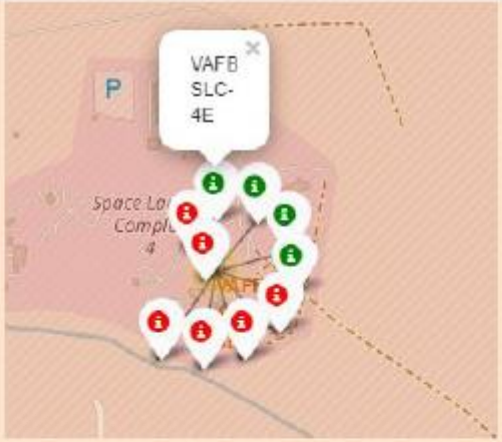
Launch Site – Locations



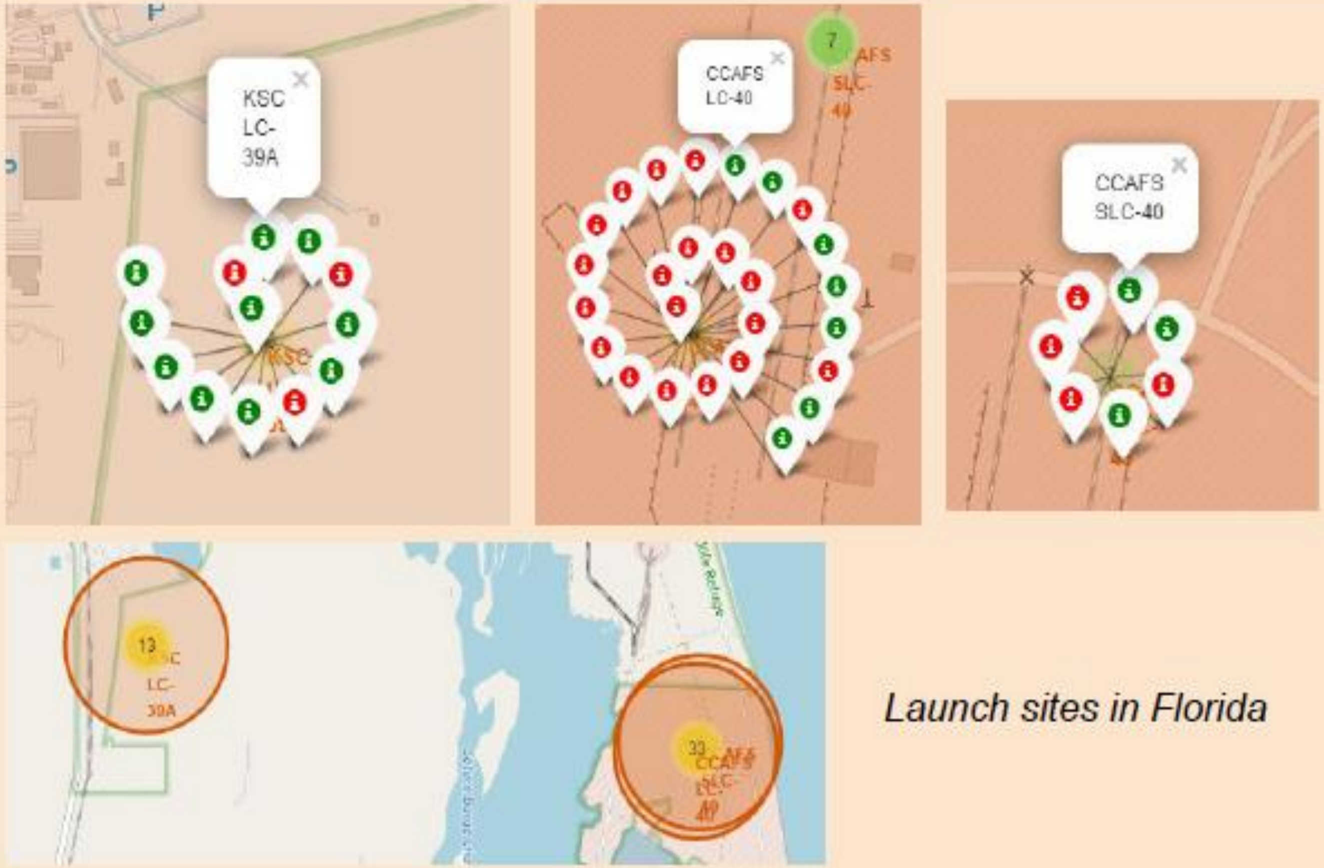
Maps show all SpaceX launch sites are:

- Located in the United States
- Located near the coast

Launch Outcomes – Color Labeled



Launch site in California

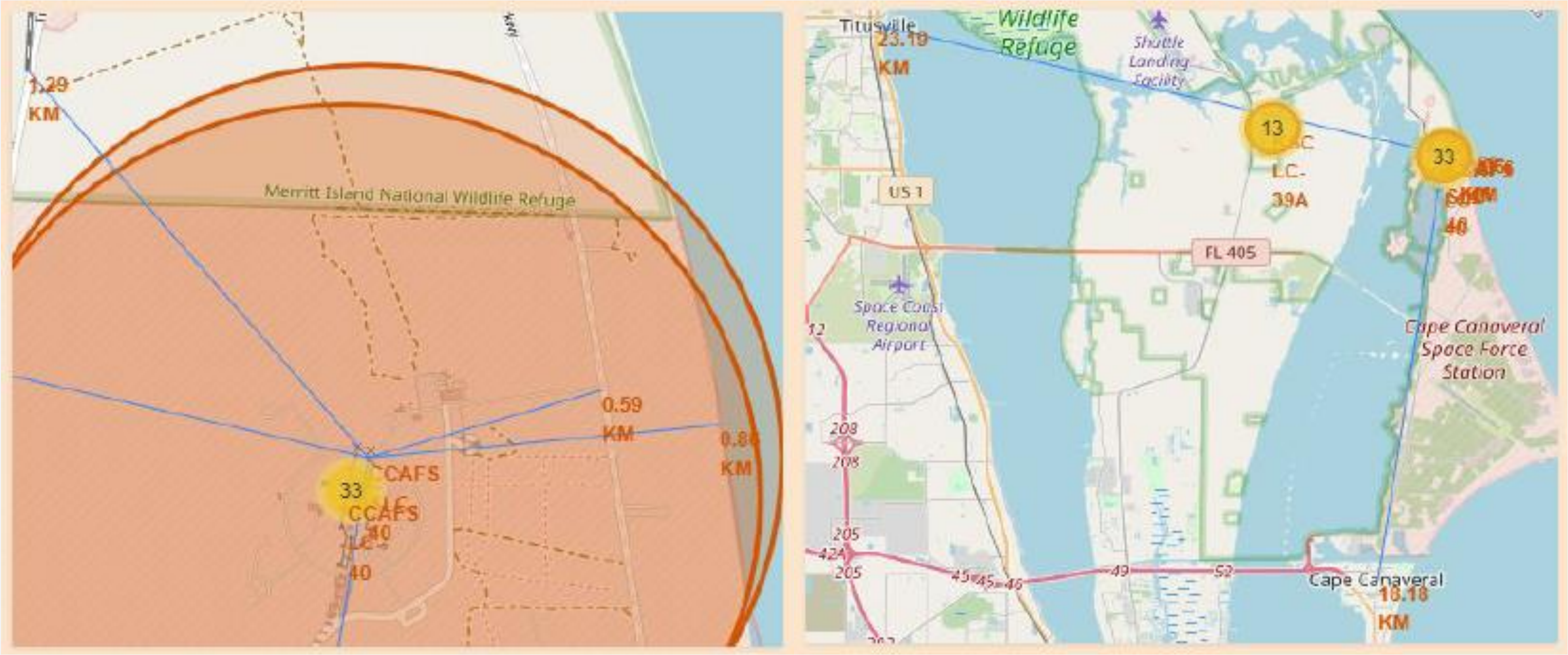


Launch sites in Florida

Maps show all SpaceX launch outcomes as:

- Successful landings (green)
- Failed landings (red)

Launch Sites - Proximities



Maps show that SpaceX launch sites are:

- Close to railways and highways
- Close to the coastline
- Distanced away from the city centers

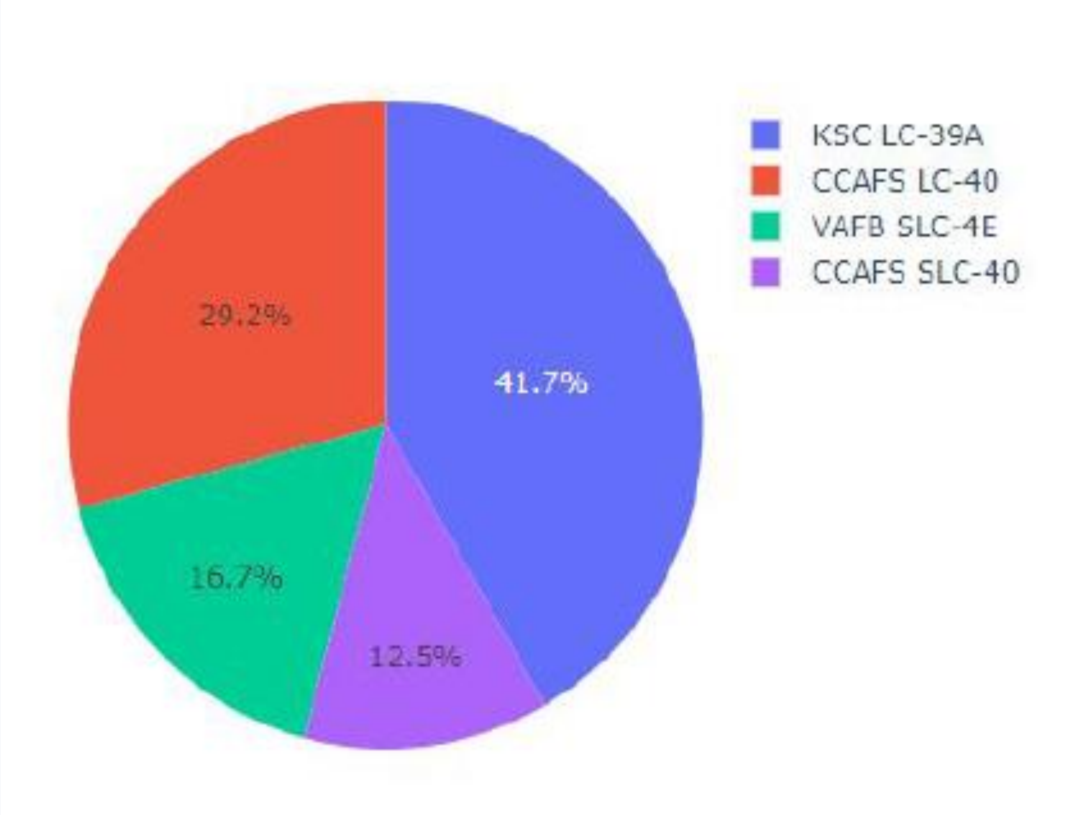
Section 5

Build a Dashboard with Plotly Dash

Launch Successes for all sites

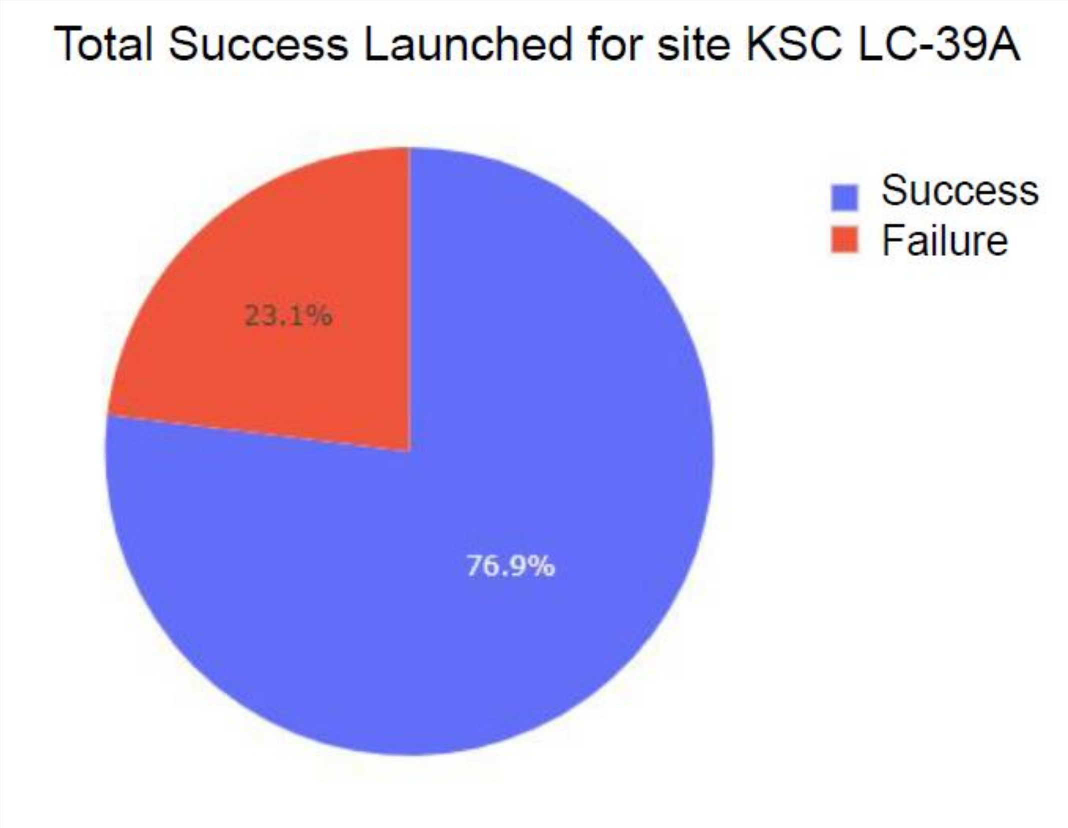
- **KSLC-39A** recorded the most launch successes amongst all sites.

- **VAFB SLC-4E** has the fewest launch successes. Possible reasons could be:
  - Data sample size is small
  - Location driven since it is situated in the west coast



Launch Site with Highest Launch Success Ratio

- **KSLC-39A** has the highest success rate with 10 successful landings (*76.9%*) and 3 failed landings (*23.1%*)



Payload vs. Launch Outcome

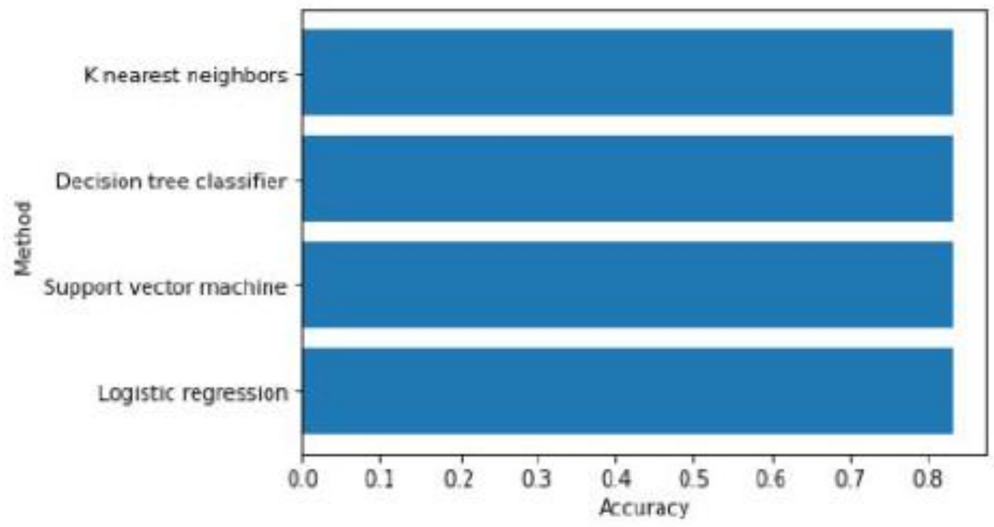
Launch success rate (Class 1) for light-weighted payloads (<5,000 kg) is **higher** than heavy-weighted payloads (>5,000 kg).

Section 6

Predictive Analysis (Classification)

Classification Accuracy

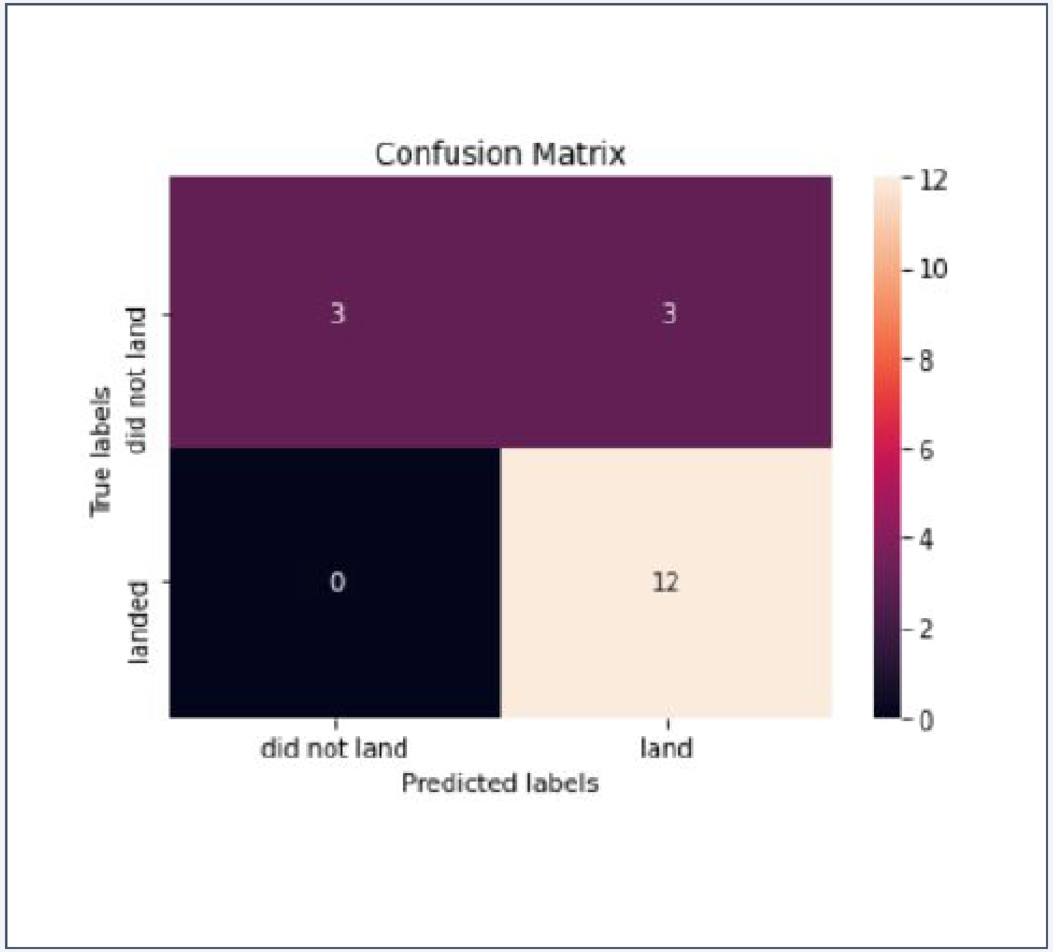
- With the test set, all models scored the same accuracy at *83.33%.*
- However, considering the small test size of 18, more data is required to test for and determine the best optimal model.



	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

Confusion Matrix

- With the test set, all models have the same confusion matrix since accuracy score was the same as well.
- 12 true successful and 3 true failed landings were correctly predicted.
- However, there were also 3 wrongly predicted successful landings when true label was failure (false positives).
- Overall, there were more successful landings predicted by the models.



Conclusions

- ✓ As the number of flights increased over time, the success rate has also increased steadily in step, with most recent success rate exceeding 80%.
- ✓ Orbital types SSO, HEO, GEO and ES-L1 have the highest success rate (*100%).*
- ✓ Launch sites are near to railways, highways and coastline, but further away from city centers.
- ✓ KSLC-39A recorded the highest number of successful launches and the highest success rate amongst all launch sites.
- ✓ Launch success rate of light-weighted payloads is higher than that of heavy-weighted payloads.
- ✓ All models achieved the same accuracy (83.33%), however more data is required to determine the best optimal model due to small data size used.

Appendix

GitHub Repository – IBM Data Science

https://github.com/chowtak/IBM-Data-Science

Coursera – Applied Data Science Capstone Course

https://www.coursera.org/learn/applied-data-science-capstone/home/welcome

Thank you!