Optimizing Space Exploration with Data Science

David Robertson
March 29, 2023

Outline

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

Executive Summary

- Summary of methodologies
  - Data collection – SpaceX API
  - Data Collection – Web Scraping
  - Data wrangling
  - Exploratory Data Analysis (EDA) with Data Visualization
  - EDA with SQL
  - Interactive Mapping with Folium
  - Dashboarding with Plotly Dash
  - Predictive Analysis (Classification)
- Summary of all results
  - EDA Results
  - Interactive Analytics
  - Predictive Analysis (Classification)

Introduction

- Project background and context

  SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, with much of the savings due to SpaceX's ability to re-use the first stage. If it can be determined if the first stage will land, the cost of a launch can be determined. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch. This goal of the project is to create a machine learning pipeline to predict if the first stage will land successfully.

- Problems to address

  What factors determine if the rocket will land successfully?

  What interaction amongst various features can determine the success rate of a successful landing?

  What operating conditions need to be in place to ensure a successful landing program?

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - SpaceX API
  - Web Scraping
- Perform data wrangling
  - One Hot Encoding data fields for Machine Learning
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models

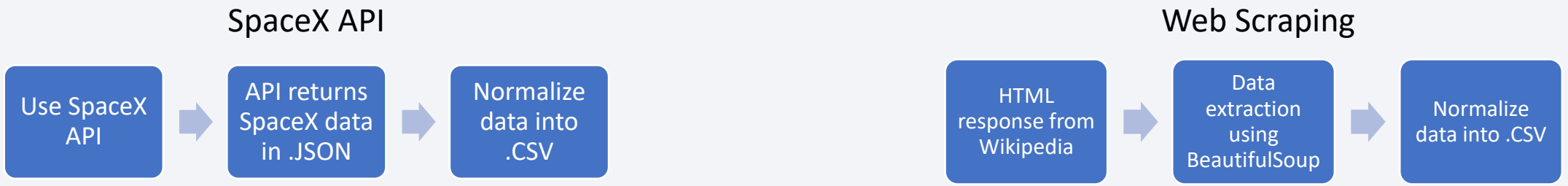
Data Collection

Data was collected using SpaceX API through a GET request.

Response content was decoded as a Json using json() function call and then converted to a pandas dataframe using json_normalize(). Data was cleaned and missing values replaced as appropriate.

Additional data extraction performed using web scraping from Wikipedia with BeautifulSoup.

Launch Data collected included information about the rocket used, payload delivered, launch specifications, landing specifications, and landing outcome. All data was converted to a pandas dataframe for analysis.

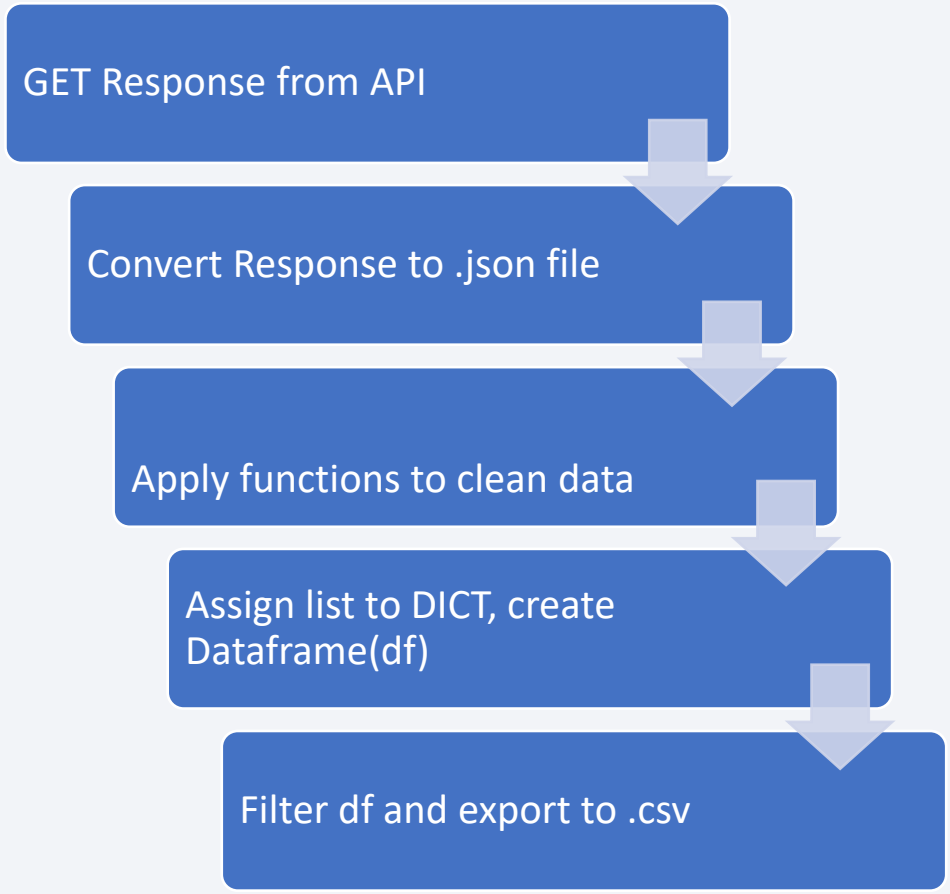


Data Collection – SpaceX API

Data was collected using SpaceX API through a GET request.

Response content was decoded as a Json using json() function call and then converted to a pandas dataframe using json_normalize(). Data was cleaned and missing values replaced as appropriate.

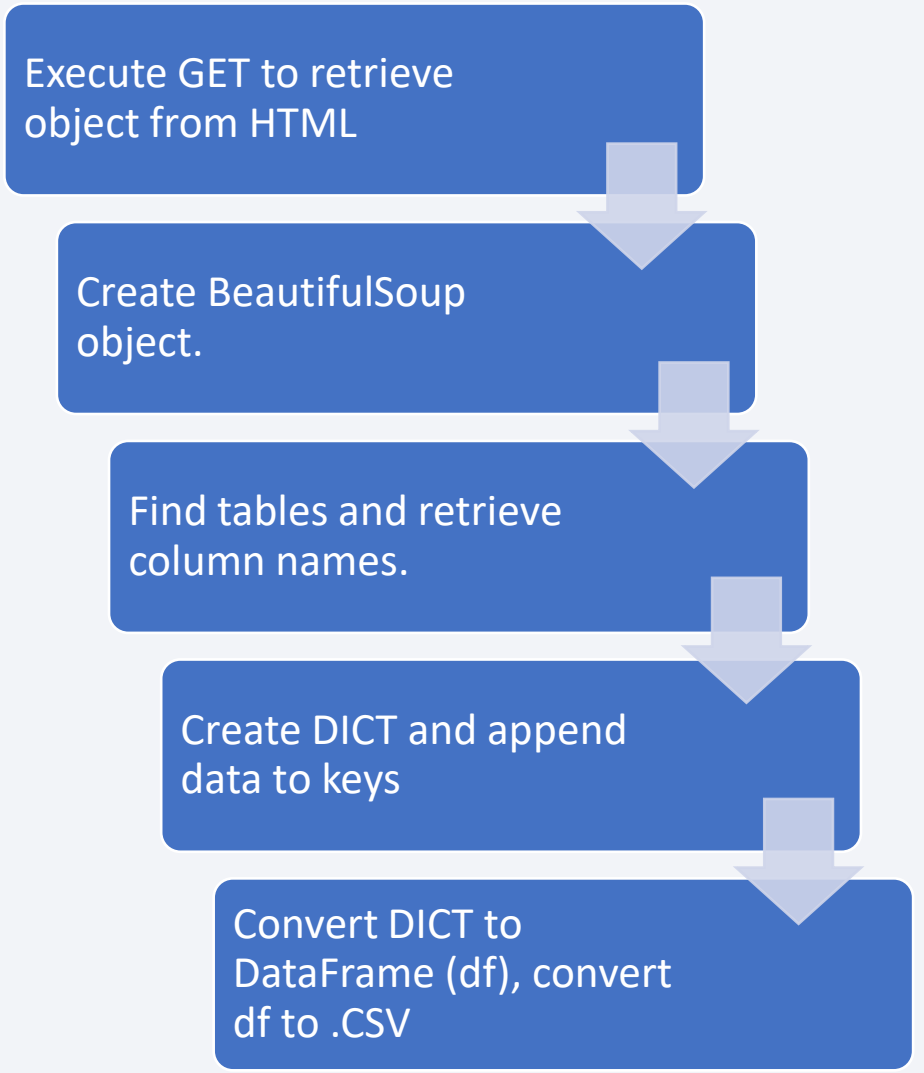
GitHub URL of the completed SpaceX API calls notebook:
https://github.com/drobertso/Capstone/blob/f1f6b787ae7efaceb343c194db9c28e89495148c/SpaceX%20Data%20Collection%20API.ipynb



Data Collection - Scraping

Performed web scraping to collect Falcon 9 historical launch records from a Wikipedia using GET request and BeautifulSoup. Created a Dictionary then DataFrame by parsing the data HTML. Converted df to .CSV

GitHub URL of the completed web scraping notebook:
https://github.com/drobertso/Capstone/blob/f1f6b787ae7efaceb343c194db9c28e89495148c/SpaceX%20Webscraping.ipynb



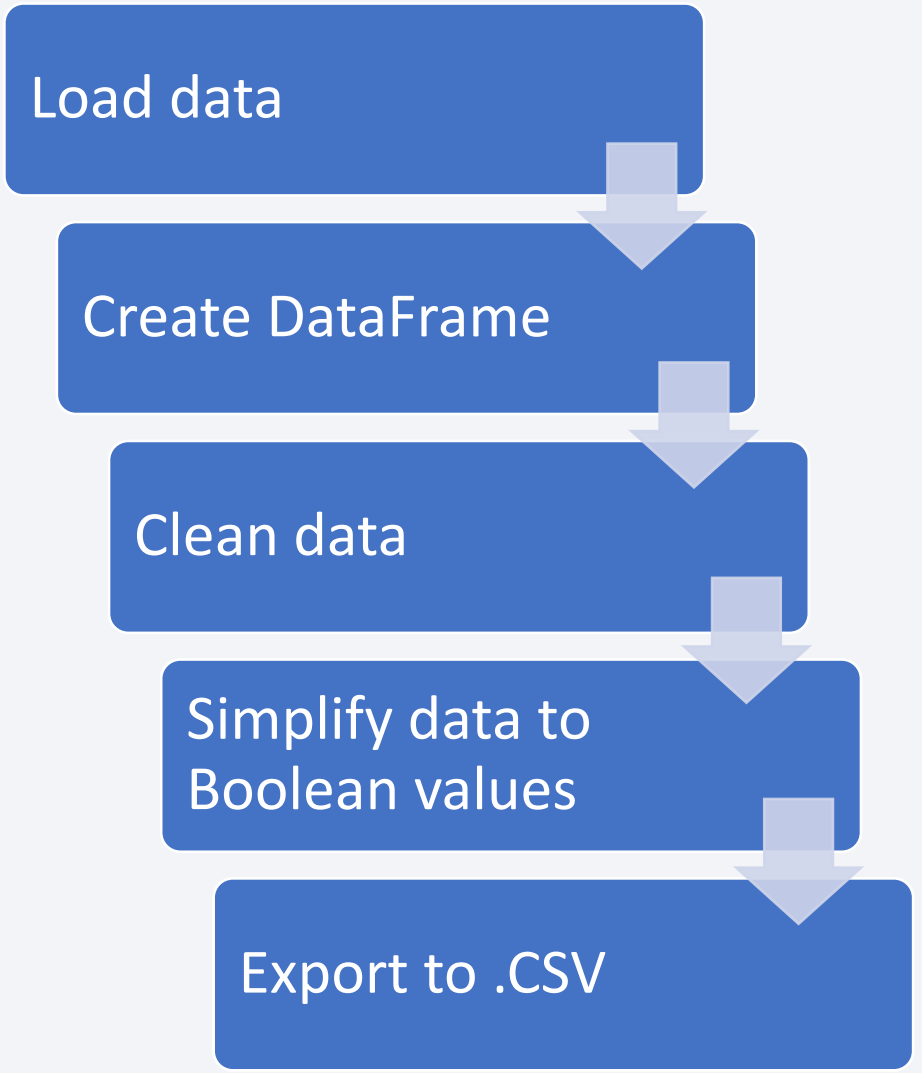
Data Wrangling

Data was collected then used to create a Pandas DF, then filtered using the BoosterVersion column to only keep the Falcon 9 launches.

Data was cleaned (i.e. missing data values were replaced with the mean value of the associated column.)

Exploratory Data Analysis (EDA) was conducted to identify patterns in the data and determine appropriate training supervised models.

GitHub URL of completed data wrangling related notebooks: https://github.com/drobertso/Capstone/blob/f1f6b787ae7efaceb343c194db9c28e89495148c/SpaceX%20Data%20Wrangling.jupyterlite.ipynb



EDA with Data Visualization

Scatter plots were used to visualize the relationship between Flight Number / Launch Site, Payload / Launch Site, Flight Number / Orbit type, Payload / Orbit type.

- Scatter plots show how much one variable is affected by another, the relationship between the two being called correlation. Scatter plots usually consist of a large body of data.

Bar chart was used to visualize the relationship between success rate of each orbit type.

- Bar charts provide easy visual comparison between different groups, with categories on one axis and a discrete value on the other. Bar charts can also show big changes in data over time.

Line plot was used to visualize the launch success yearly trend.

- Line graphs are useful in that they show data variables and trends very clearly and can help to make predictions about the results of data not yet recorded.

GitHub URL of completed EDA with data visualization notebook:
https://github.com/drobertso/Capstone/blob/fc97d52bb50681e9b7bab1503163aed14e9bdf28/SpaceX%20EDA%20DataViz.ipynb.jupyterlite.ipynb

EDA with SQL

EDA was executed with SQL to get insight from the data. The following queries were utilized:

- Display the names of the unique launch sites in the space mission
- Display 5 records where launch sites begin with the string 'KSC'
- Display the total payload mass carried by boosters launched by NASA (CRS)
- Display the average payload mass carried by booster version F9 v1.1
- List the date where the successful landing outcome on a drone ship was achieved.
- List the names of the boosters which have success on the ground pad 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.
- List the records which will display the month names, successful landing outcomes on ground pad, booster versions, and launch site for the months in year 2017
- Rank the count of successful landing outcomes between the date 2010 06 04 and 2017 03 20 in descending order.

GitHub URL of completed EDA with SQL notebook:
https://github.com/drobertso/Capstone/blob/0d21f03ad0750fbd17693f3214118b0b05ace6f7/SpaceX%20EDA%20with%20SQL.ipynb

Build an Interactive Map with Folium

To better visualize the Launch Data on an interactive map, Latitude and Longitude coordinates for each launch site was leveraged to add Circle Markers around each launch site with a label of the name of the launch site.

Launch_outcomes (failures, successes) were assigned to classes 0 and 1 in the df, then with Green and Red markers on the map in a MarkerCluster() to distinguish launch sites with high success rates.

Lines are drawn on the map to measure distance to select landmarks and distances between launch sites and landmarks calculated to determine trends and patterns.

GitHub URL of completed interactive map with Folium map:
https://github.com/drobertso/Capstone/blob/e3785055b331b70fbfd3079867984ce862ce1740/SpaceX%20Launch%20Site%20Locations%20with%20Folium.jupyterlite.ipynb

Build a Dashboard with Plotly Dash

An interactive dashboard was built with Plotly dash that included charts for visualization and analysis support. Chart type used and reason for selection are below.

Pie chart was used to show the total launches by a certain sites/all sites.
- These display relative proportions of multiple classes of data
- Size of circle can be made proportional to the total quantity it represents.

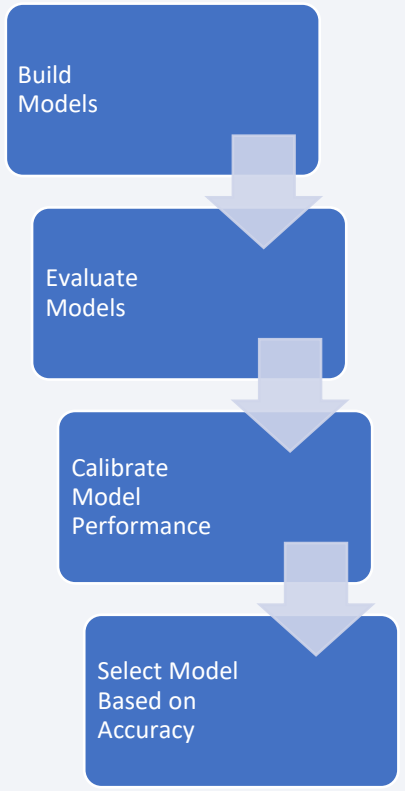
Scatter graph was used to show the relationship with Outcome and Payload Mass (Kg) for the different booster versions.
- Show the relationship between two variables.
- Best method to show you a non-linear pattern.
- Supports determination of the range of data flow, i.e. maximum and minimum value.
- Observation and reading are straightforward.

To support analysis – the following interactions were also added:
- Launch Site Drop-down Input Component
- Callback function to render success-pie-chart based on selected site dropdown
- Range Slider to Select Payload
- Callback function to render the success-payload-scatter-chart scatter plot

GitHub URL of completed Plotly Dash lab:
https://github.com/drobertso/Capstone/blob/f1f6b787ae7efaceb343c194db9c28e8949514 8c/SpaceX%20Interactive%20Dash%20with%20Plotly

Predictive Analysis (Classification)

In order to find the best performing model using the test data between SVM, Classification Trees, k nearest neighbors and Logistic Regression, the progressive steps below were used.



Build Models

- Load dataset into NumPy and Pandas
- Transform Data
- Split data into training and test data sets, and check number of test samples
- Assess machine learning algorithms
- Set parameters and algorithms to GridSearchCV
- Fit datasets into the GridSearchCV objects and train dataset.

Evaluate Models

- Check accuracy for each model
- Get tuned hyperparameters for each type of algorithms
- Plot Confusion Matrix

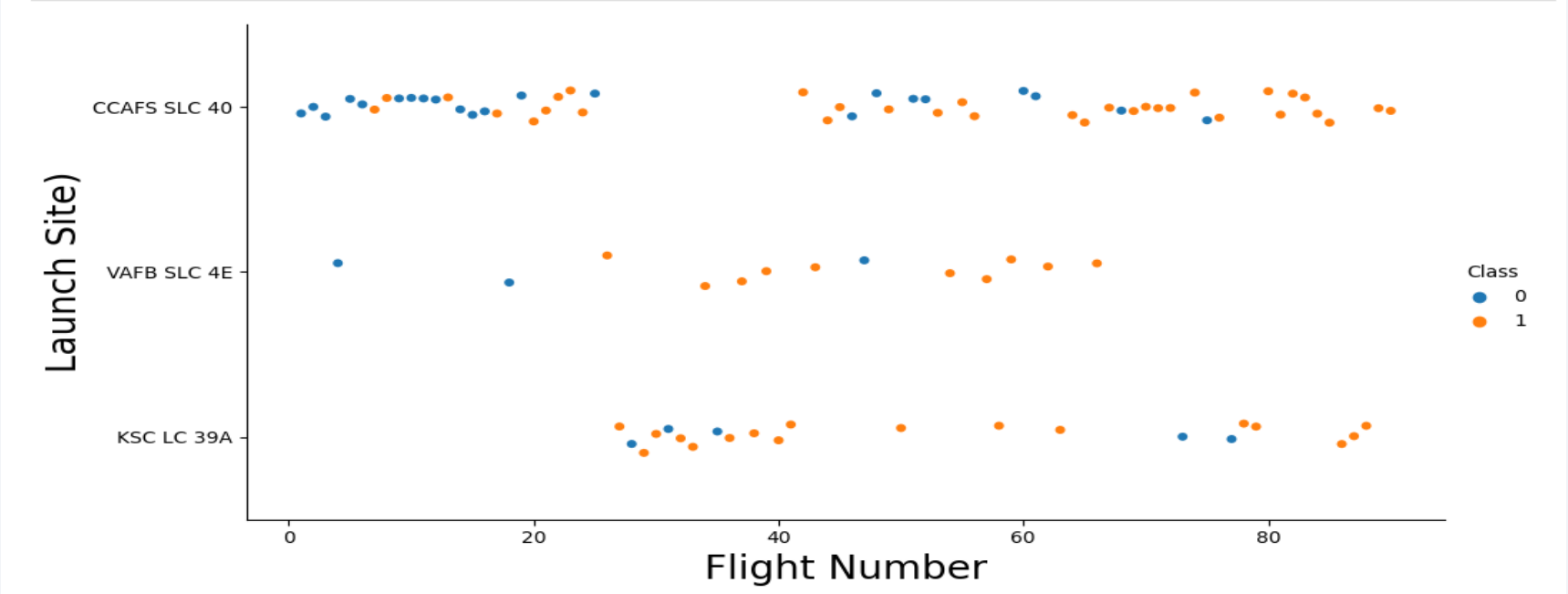
Calibrate Model Performance (Feature Engineering and Algorithm Tuning)

Select model (model with the best accuracy score)

GitHub URL of completed predictive analysis lab:
https://github.com/drobertso/Capstone/blob/f1f6b787ae7efaceb343c194db9c28e89495148c/SpaceX%20Machine%20Learning%20Prediction.jupyterlite.ipynb

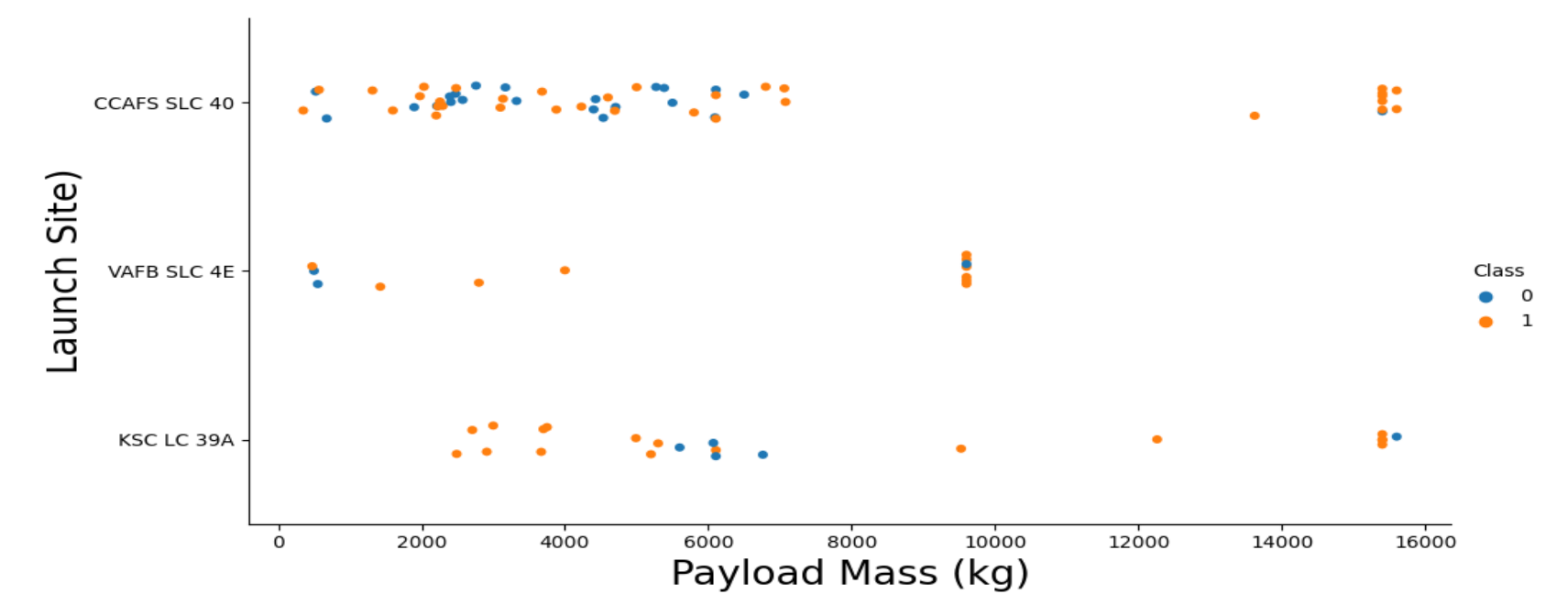
Insights Drawn from EDA

Flight Number vs. Launch Site



The chart indicates a greater success rate as the volume of flights increases at each site.

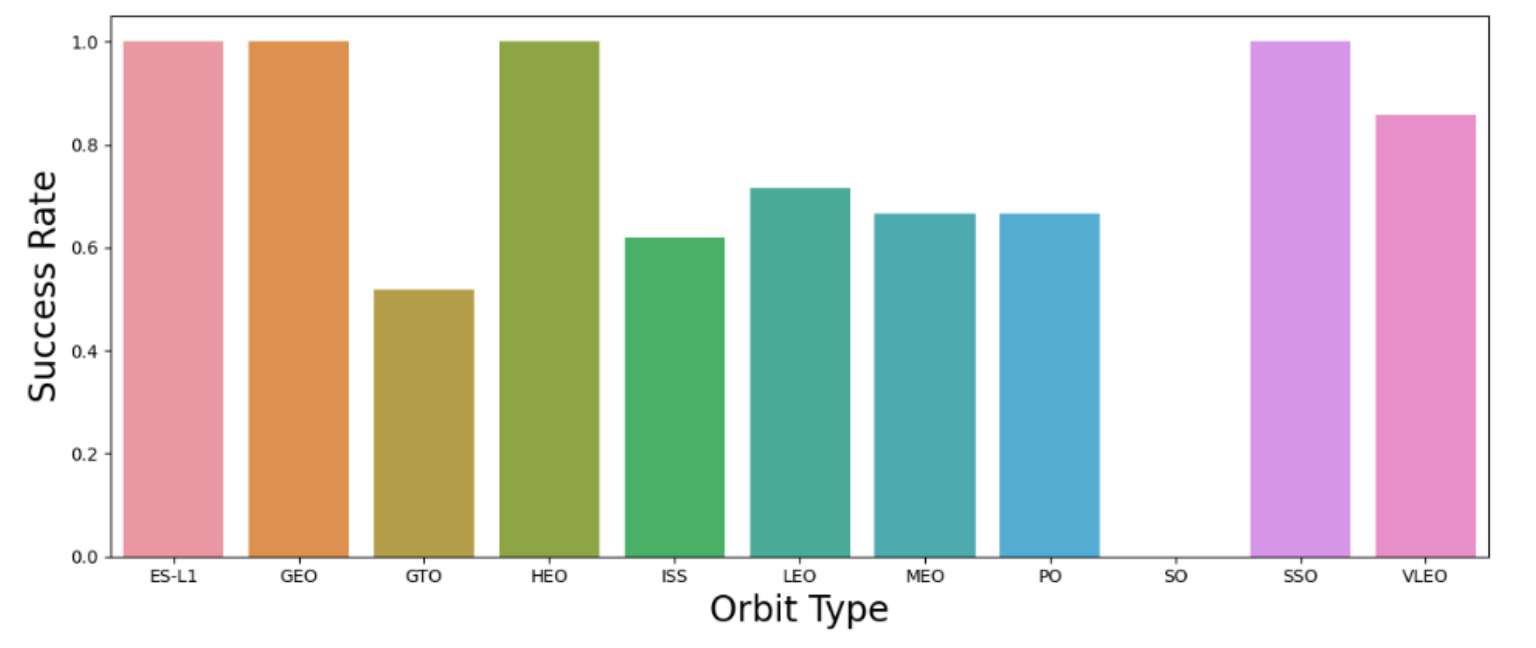
Payload vs. Launch Site



The chart indicates the following:

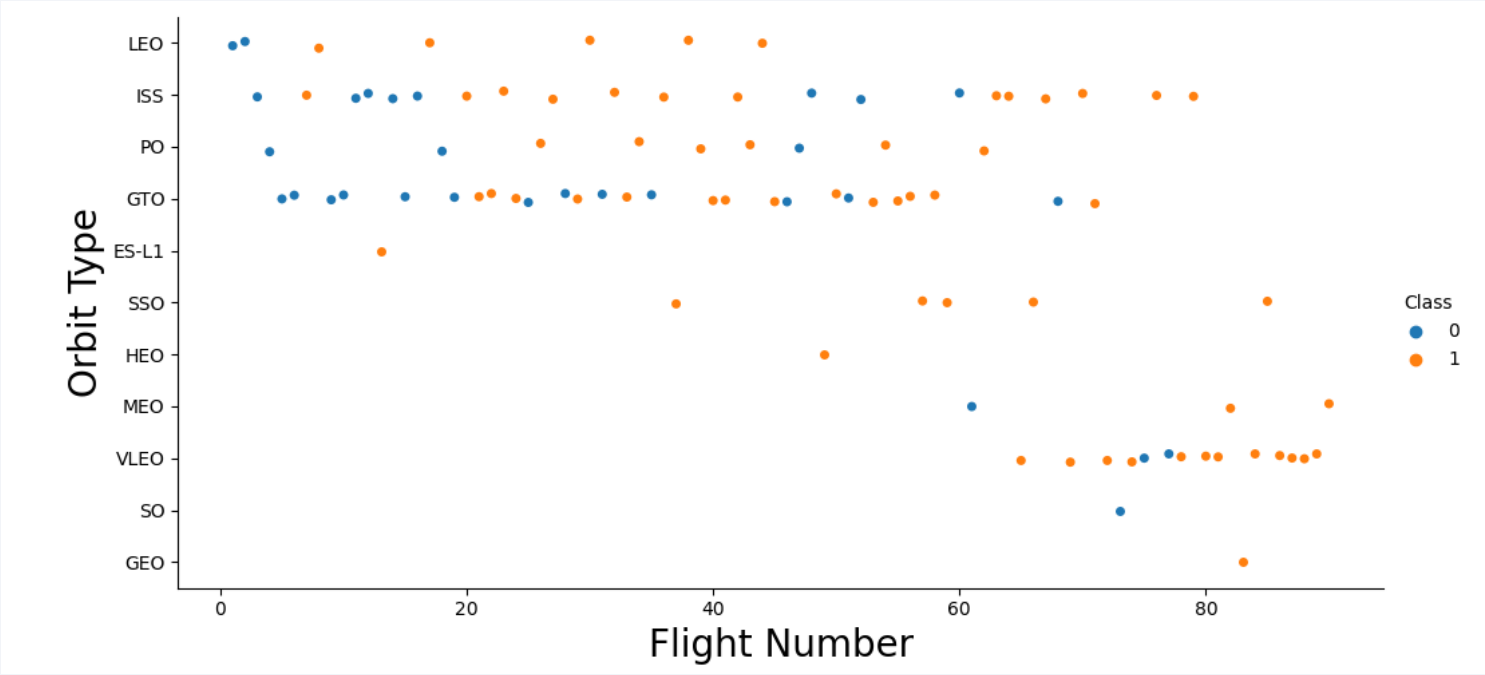
- Payload limit of 10000 at site VAFB SLC-40
- Larger payloads appear to have a higher launch success rate

Success Rate vs. Orbit Type



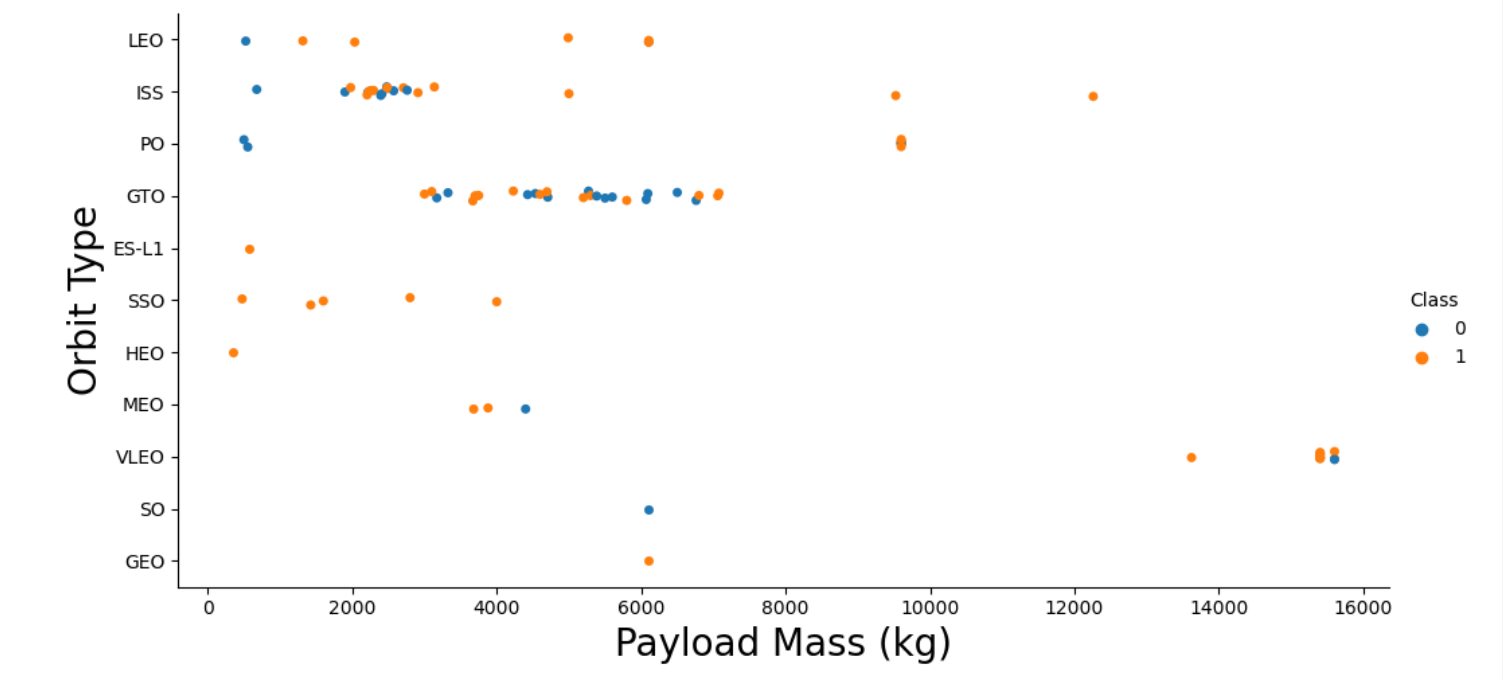
The bar chart indicates that orbits ES-L1, GEO, HEO, and SSO have a the highest success rates (100%.)

Flight Number vs. Orbit Type



- The Leo orbit has a significant increase in success rate with an increase in the number of flights.
- There does not appear to be any such pattern with the ISS, PO, or GTO orbits.

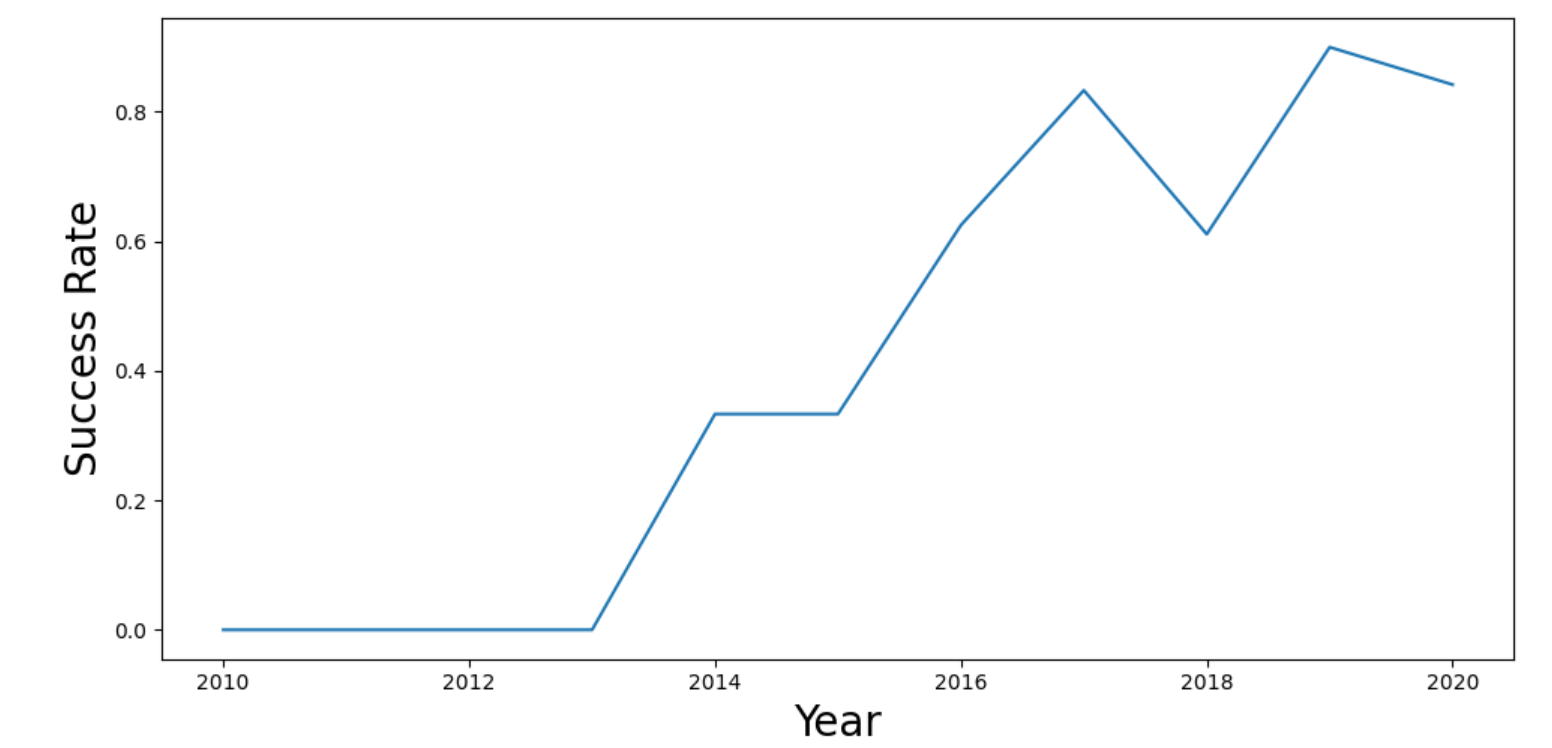
Payload vs. Orbit Type



Increased payload appears to have a higher success rate in orbits LEO, ISS and PO.

No clear pattern exists for GTO orbit relative to payload and success rate.

Launch Success Yearly Trend



With exception of 2018 and 2020, there has been steady increase in success rate since 2013.

All Launch Site Names

DISTINCT allows us to extract only unique launch site identifiers from SPACEXTBL.

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

```
%sql select DISTINCT(LAUNCH_SITE) from SPACEXTBL
```

```
 * 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'

Use of **%** with CCA allows us to extract site names that start with “CCA” (regardless of what comes after in the name.

LIMIT 5 allows us to limit the extraction to 5 sites.

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

```
%sql select * from SPACEXTBL 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
04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
08-12-2010	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)
22-05-2012	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
01-03-2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

SUM function allows us to summate the Payload Mass in the column.

WHERE function allows us to define summation for only Customer NASA (CRS).

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

```
%sql select sum(PAYLOAD_MASS__KG_) from SPACEXTBL where CUSTOMER = 'NASA (CRS)'
```

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

sum(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

AVG function allows us to calculate the average Payload Mass in the column.

WHERE function allows us to limit scope to the F9 v1.1 booster.

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

```
%sql select avg(PAYLOAD_MASS__KG_) from SPACEXTBL where BOOSTER_VERSION = 'F9 v1.1'
```

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

avg(PAYLOAD_MASS__KG_)
2928.4

First Successful Ground Landing Date

MIN function allows us to determine the earliest launch date.

WHERE function allows us to narrow function to a Success (ground pad) landing outcome.

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

Hint:Use min function

```
%sql select min(DATE) from SPACEXTBL where "Landing _Outcome" = "Success (ground pad)";
```

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

min(DATE)
01-05-2017

Successful Drone Ship Landing with Payload between 4000 and 6000

WHERE function allows us to narrow view to a Success (drone ship) landing outcome population.

BETWEEN function allows us to define a Payload Mass range within we need our population to fall.

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

```
%sql select BOOSTER_VERSION from SPACEXTBL where "Landing _Outcome" ='Success (drone ship)' and PAYLOAD_MASS__KG_ BETWEEN 4000 AND 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

COUNT allows us to quantify the number of missions.

GROUP BY allows us to distinguish missions using Mission Outcome (success/failure).

List the total number of successful and failure mission outcomes

```
%sql Select MISSION_OUTCOME,count(MISSION_OUTCOME) as count from SPACEXTBL GROUP BY MISSION_OUTCOME
```

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

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

Boosters Carried Maximum Payload

DISTINCT allows us to extract only unique Booster Version identifiers from SPACEXTBL

MAX allows us to limit the data extraction to only those Boosters that carried the max payload.

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

```
%sql select distinct BOOSTER_VERSION from SPACEXTBL where PAYLOAD_MASS__KG_ = (select MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL)
```

```
 * 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

SUBSTR(Date,7,4) and **"Landing_Outcome"=Failure** allows us to narrow scope to failed landings in year 2015.

SUBSTR(Date,4,2) allows us to distinguish the month, and list it along with the other specified column data.

List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.

Note: SQLLite does not support monthnames. So you need to use substr(Date, 4, 2) as month to get the months and substr(Date,7,4)='2015' for year.

```
%sql select substr(Date,4,2),"Landing _Outcome",BOOSTER_VERSION,LAUNCH_SITE from SPACEXTBL where substr(Date,7,4) = '2015' and "Landing _Outcome" = 'F
```

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

substr(Date,4,2)	Landing _Outcome	Booster_Version	Launch_Site
01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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

GROUP BY with BETWEEN allows us to narrow scope to a specific date range while also distinguishing success/failed landing outcomes.

Rank the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.

```
%sql select "Landing _Outcome", count("Landing _Outcome") as count from SPACEXTBL group by "Landing _Outcome" having "DATE" between '04-06-2010' and '
```

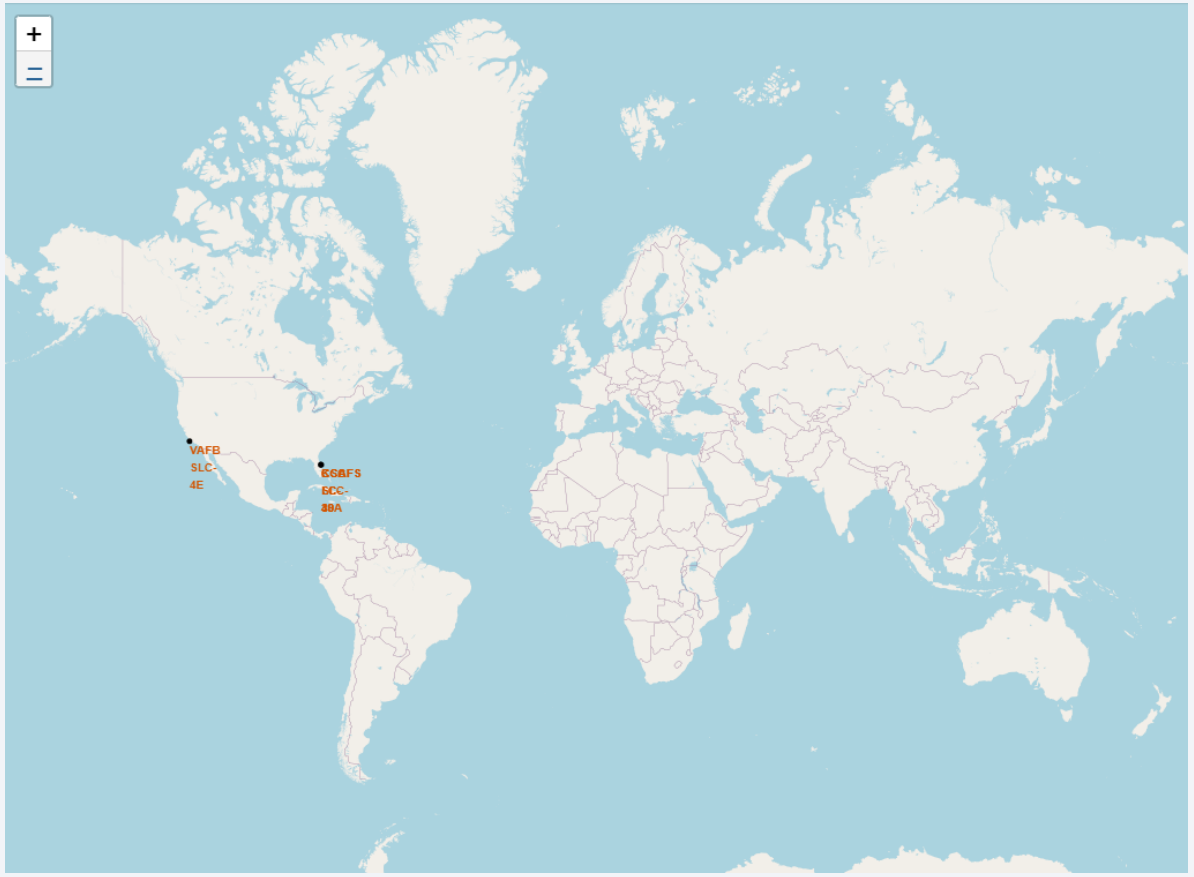
```
 * sqlite:///my_data1.db
Done.
```

Landing _Outcome	count
Controlled (ocean)	5
Failure (drone ship)	5
Success (drone ship)	14
No attempt	1
Failure	3
Failure (parachute)	2

Launch Sites Proximities Analysis

Launch Sites - Global Map with Location Markers

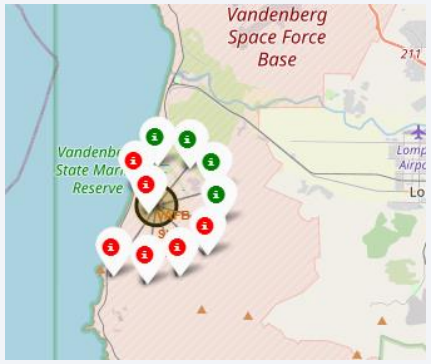
The map indicates the SpaceX launch sites are all in the United States (specifically California and Florida.)



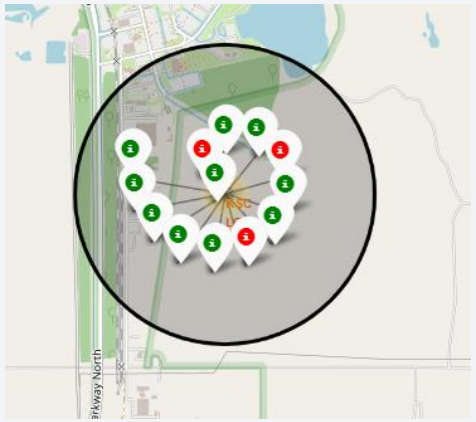
Launch Sites – Localized with Success Color Labels

The clusters below represent the number of missions for each launch site, where Green markers represent successful launches while Red markers indicate failure.

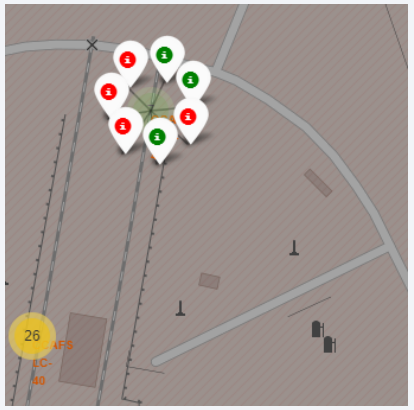
VAFB SLC-4E



KSC LC-39A



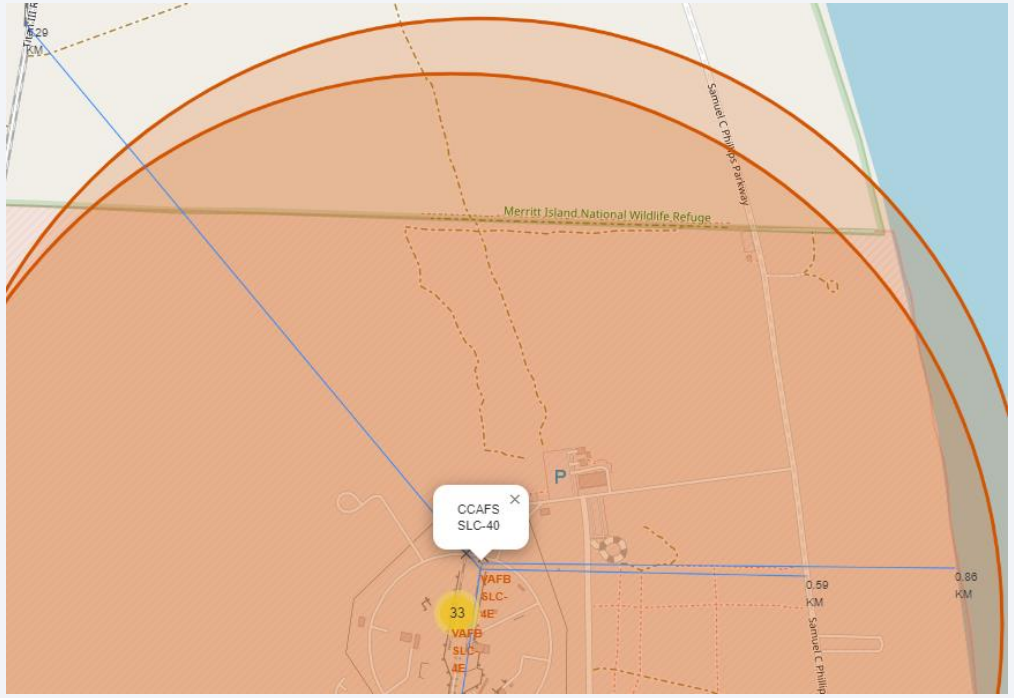
CCAFS SLC-40



CCAFS LC-40

Launch Sites – Distance to Select Landmarks

This map depicts distance of the Railway (5.29 km), Highway (0.59 km) and Coastline (0.86 km) from site CCAF SLC-40 in Florida.

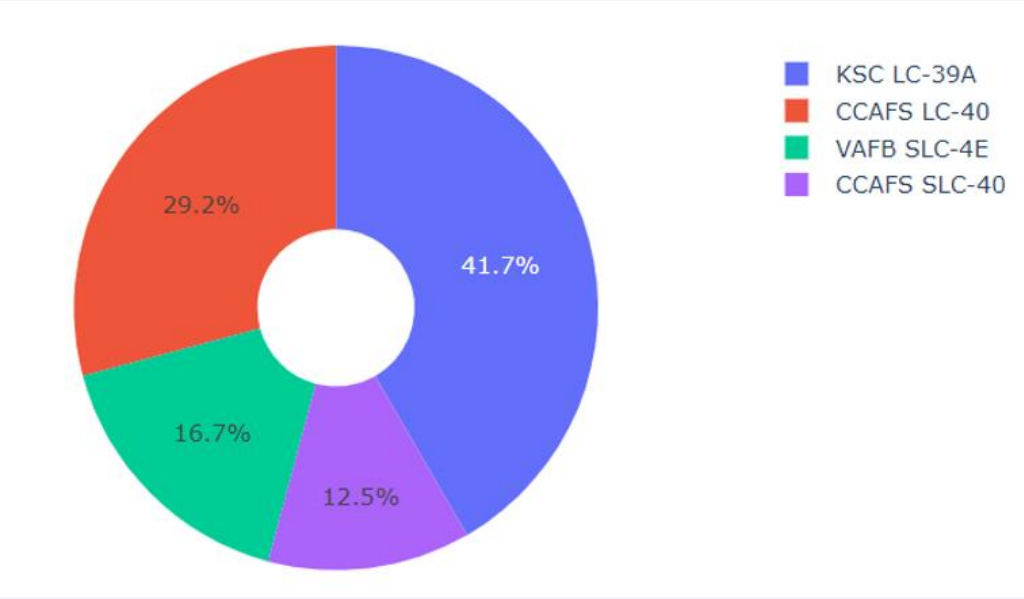


Building a Dashboard with Plotly Dash

plotly | Dash

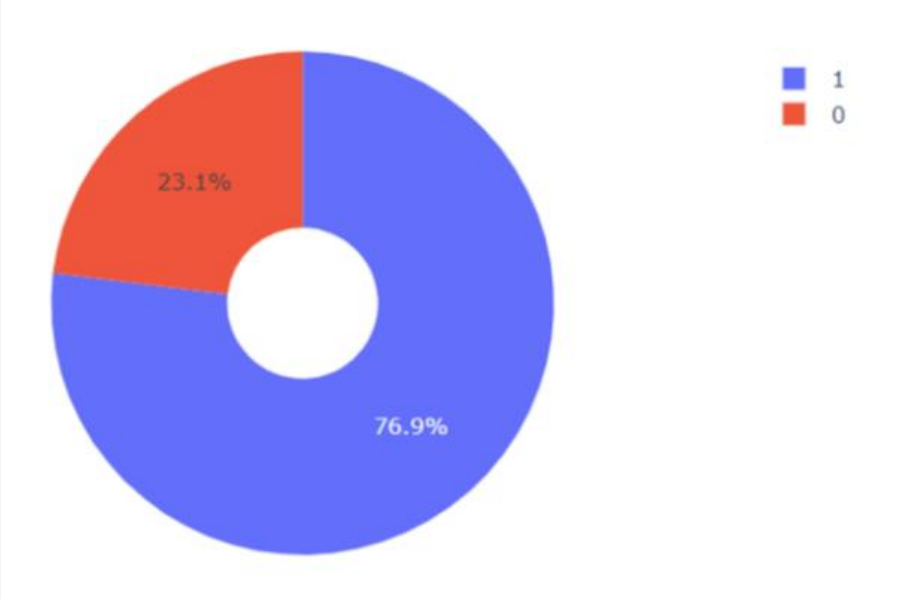
Launch Success by Site – Pie Chart

The pie chart visually represents the breakdown of successful launches by launch site, with 41.7% of total successful launches occurring at site KSC LC-39A.



Launch Site Success – Site with Highest Success

The pie chart visually represents the breakdown of total launches at site KSC LC-39A, with 76.9% of those launches deemed successful.

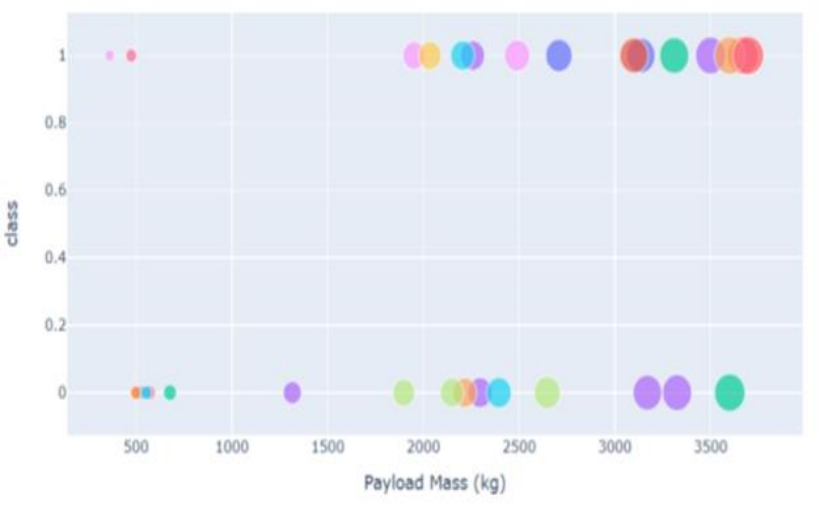


Launch Site Success – Scatterplot Comparison

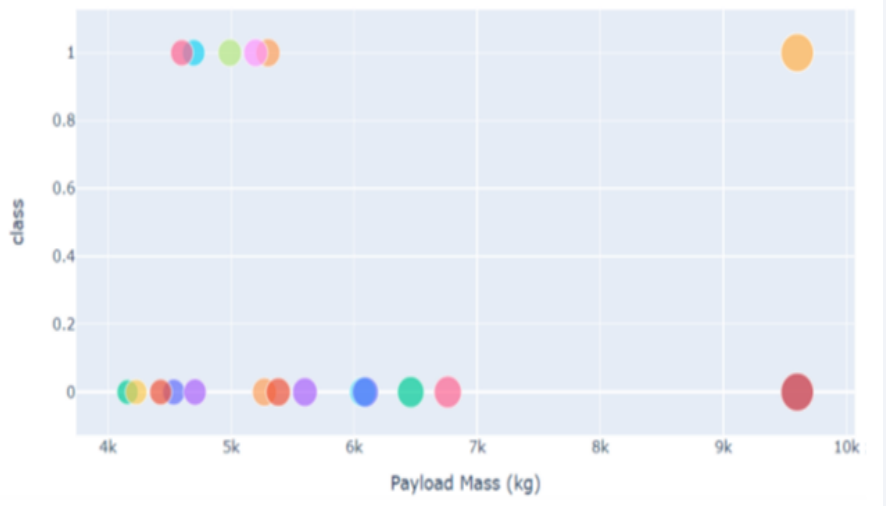
Charts below indicate a higher success rate with Low Range Payloads (larger volume at classification 1 than 0) than launches with High Range Payloads (larger volume at classification 0 than 1.)

- Class 1 indicates success, 0 indicates failure
- Circle size indicates volume
- Circle color indicates booster version

Low Range Payload (0kg – 4000kg)



High Range Payload (4000kg – 10000kg)

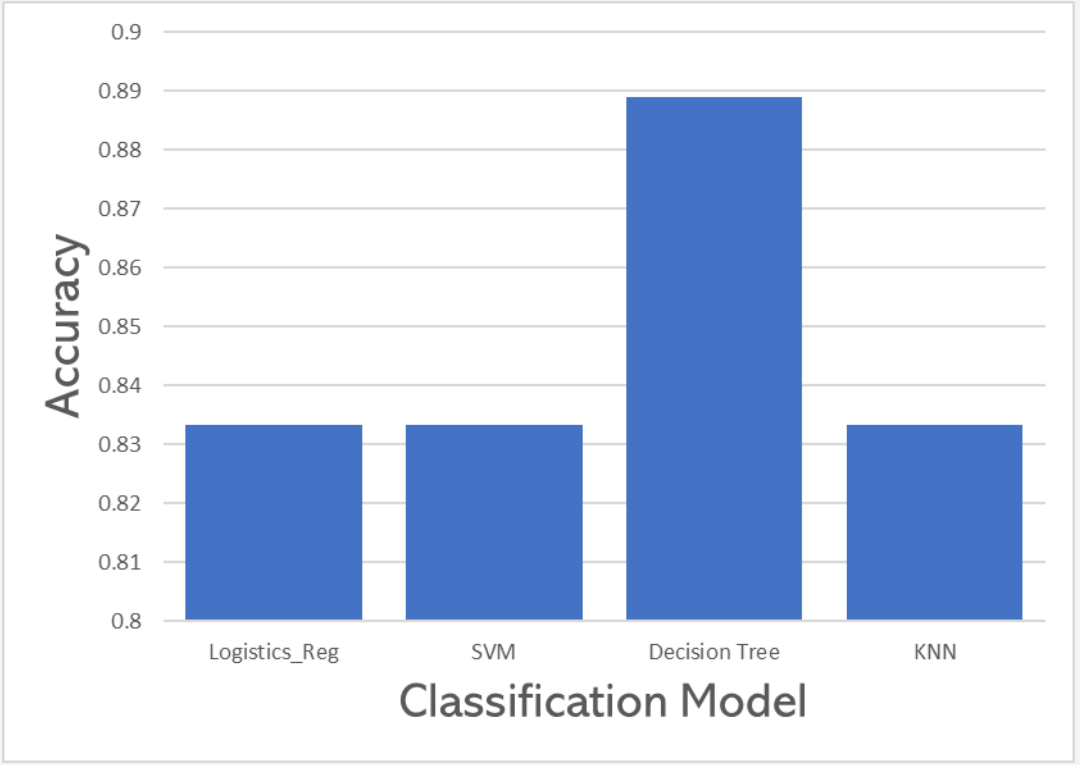


Predictive Analysis (Classification)

Classification Accuracy

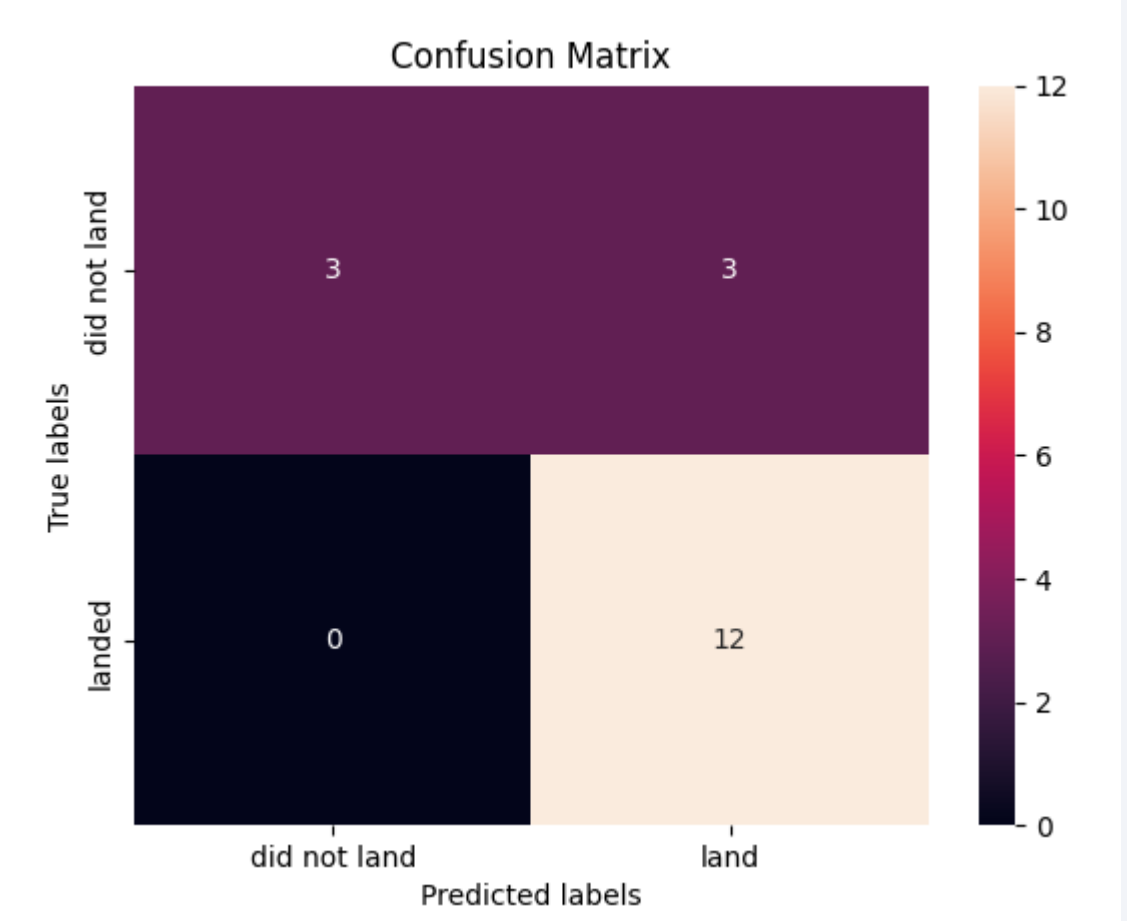
Decision tree has the highest accuracy at 88.89%.

Method	Test Data Accuracy
Logistic_Reg	0.833333
SVM	0.833333
Decision Tree	0.888889
KNN	0.833333



Confusion Matrix

The Confusion Matrix for the Decision Tree was the same as the other classification models as all four are able equally distinguish between the different classes. The major problem is false positives for all the models.



Conclusions

Based on the data analysis, we can conclude the following:

- The larger the volume of flights at a launch site, the greater the success rate at that site.
- With exception of 2018 and 2020, there has been a steady increase in the Launch Success rate.
- Launch site KSC LC-39A had the most successful launches of any sites, but success rate is negatively impacted by increasing payloads.
- Low weight payloads had higher success rates than heavy payloads
- The Success Rate is highest in orbits GEO,HEO,SSO and ES-L1.
- The Decision Tree classifier is the best machine learning algorithm for this dataset.

Appendix

The following code was particularly helpful in filtering to a specific Booster version then resetting the line identifier.

```
# Hint data['BoosterVersion']!='Falcon 1'
filt =  df['BoosterVersion']!=  'Falcon 1'
data_falcon9 = df.loc[filt]
data_falcon9.head()
```

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	Longitude	Latitu
4	6	2010-06-04	Falcon 9	NaN	LEO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0003	-80.577366	28.561
5	8	2012-05-22	Falcon 9	525.0	LEO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0005	-80.577366	28.561
6	10	2013-03-01	Falcon 9	677.0	ISS	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0007	-80.577366	28.561
7	11	2013-09-29	Falcon 9	500.0	PO	VAFB SLC 4E	False Ocean	1	False	False	False	None	1.0	0	B1003	-120.610829	34.632
8	12	2013-12-03	Falcon 9	3170.0	GTO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B1004	-80.577366	28.561

Now that we have removed some values we should reset the FlgihtNumber column

```
data_falcon9.loc[:,'FlightNumber'] = list(range(1, data_falcon9.shape[0]+1))
data_falcon9
```


Thank You!