Winning Space Race with Data Science

Juan M. Escribano

20 November 2021

Outline

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)

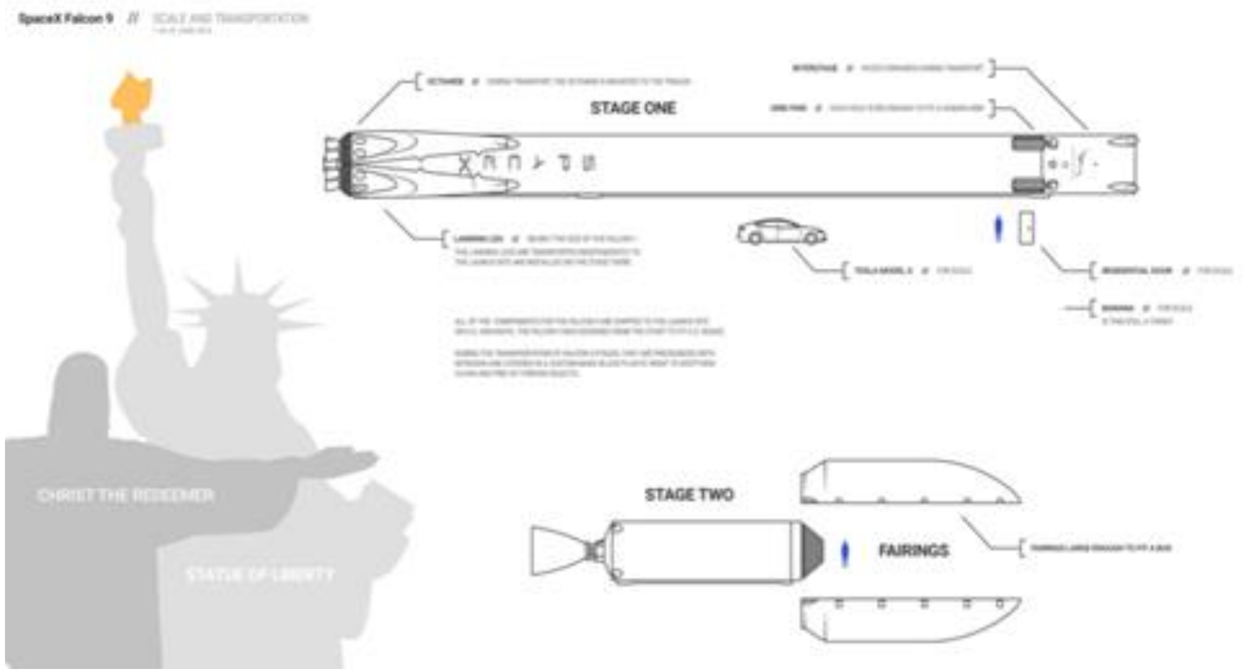
- Summary of all results
  - ✓Exploratory data analysis results
  - ✓Interactive analytics demo in screenshots
  - ✓Predictive analysis results

GitHub URL Code Master Repository

Introduction

Project background and context

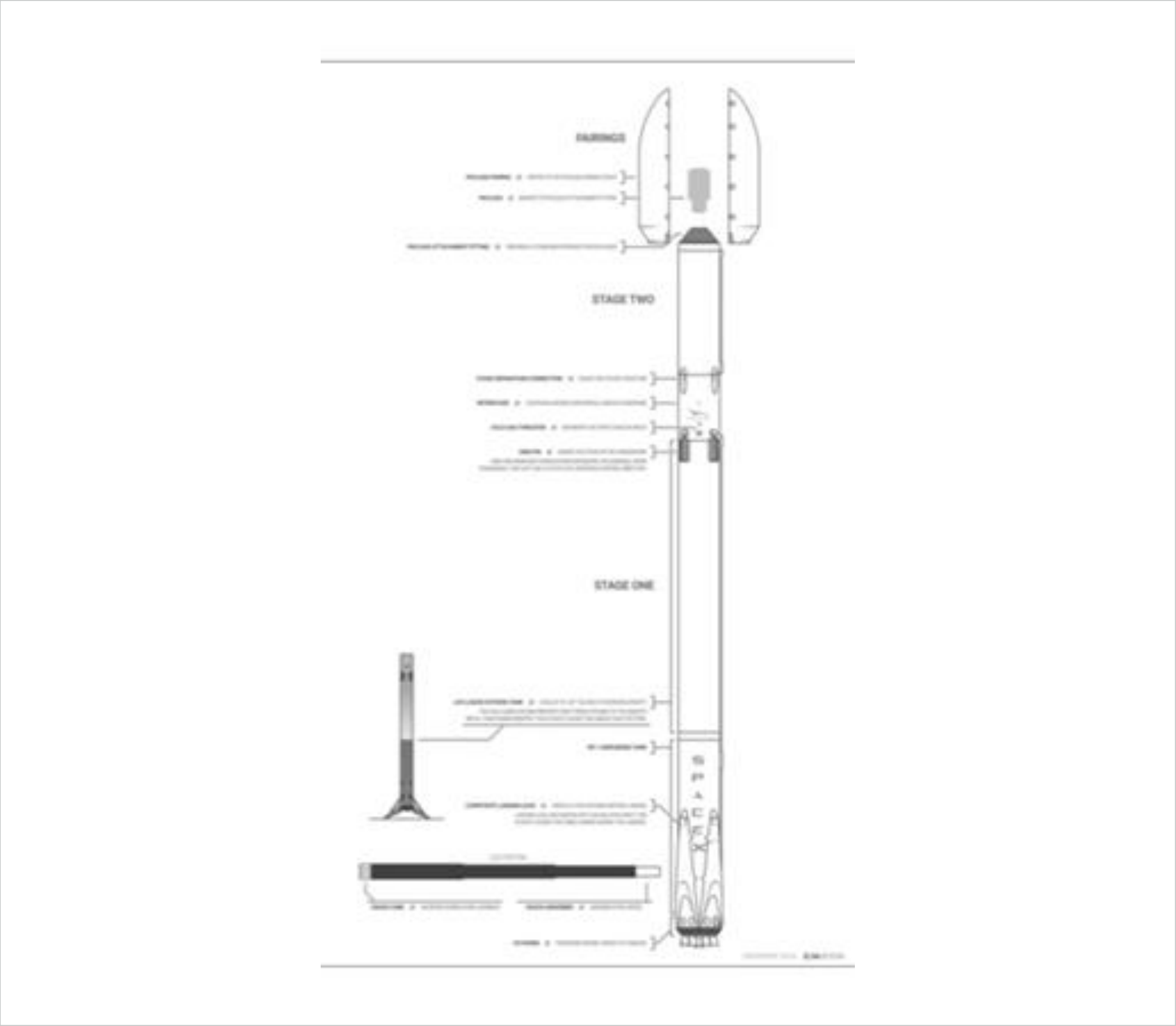
SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upwards of 165 million dollars each, much of the savings is because SpaceX can reuse the stage one. Therefore, if we can forecast if the first stage will land, we can determine the cost of a launch with more certainty. The project will do this by gathering information about Space X past launches weather successful landing and reuse of stage one.



Introduction (cont.)

Problems you want to find answers

- Determine the price of each launch by predict the successful landing and reuse of stage one?
- What variables (*e.g., payload, landing site, etc.)* influence the success of landing rate?



Section 1

Methodology

Executive Summary

Data collection

- Perform *Get* Request to the SpaceX API
- Clean and normalization of the requested data
- Web Scraping

  Wikipedia: List of Falcon 9 and Falcon Heavy launches

Exploratory Data Analysis (EDA) and determine Training Labels

- Find patterns in the data
- Determine Training Labels to determine what would be the label for training supervised models
- Standardize the data
- Construct visual analytics using interactive dashboard
- Perform predictive analysis & Machine Learning Prediction
- Find the method performs best using test data

Methodology

Summary of methodologies

- ✓Data Collection & Wrangling & Data Analysis
- ✓Data Collection: Scraping SpaceX API
- ✓EDA with Data visualization
- ✓EDA with SQL
- ✓Building an interactive map with Folium
- ✓Building a Dashboard with Plotly Dash
- ✓Predictive analysis (Classification)

Data Collection & Wrangling & Data Analysis

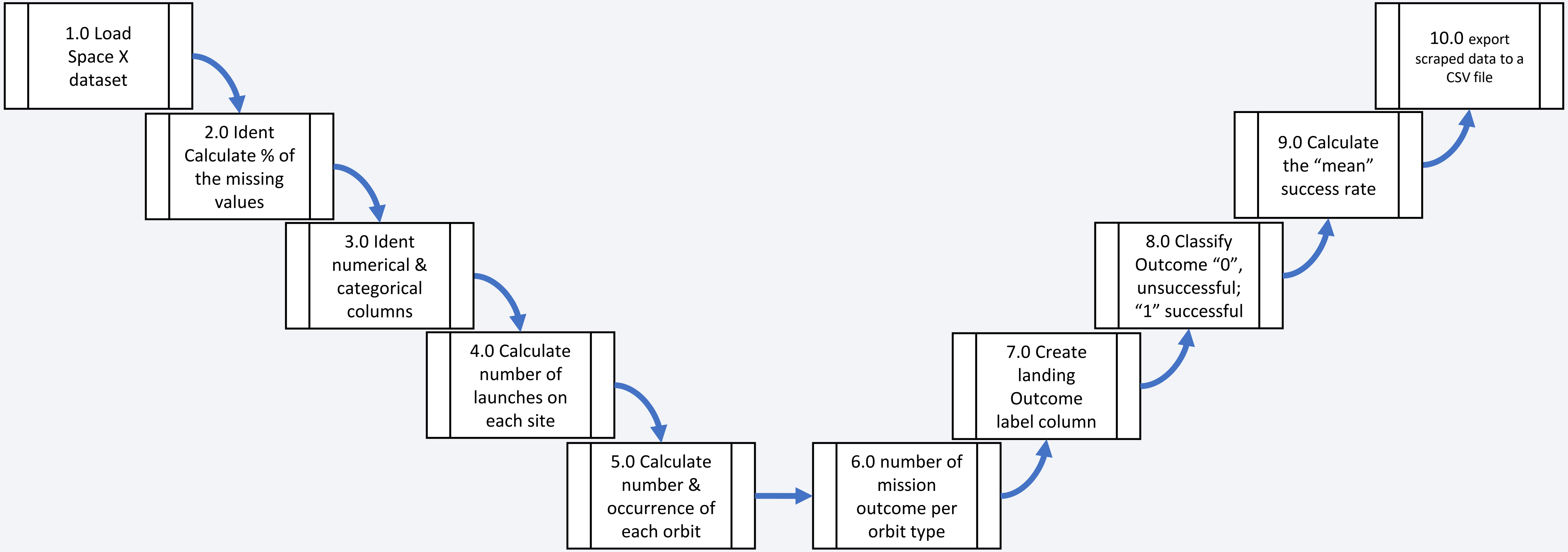
OBJECTIVES

1. Perform Exploratory Data Analysis (EDA)
2. Determine Training Labels
3. Find data patterns
4. Determine what would be the label for training a supervised models
5. convert those outcomes into Training Labels with `1` means the stage one booster successfully landed & `0` means it was unsuccessful
6. export scraped training data to a CSV file

Space X dataset Source: https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/dataset_part_1.csv)

Data Collection & Wrangling & Data Analysis

Data Collection & Wrangling & Data Analysis flowchart



GitHub URL Code: [SpaceX Data wrangling wk1.ipynb]

Data Collection: Scraping SpaceX API

Data sets were collected

- *Perform Get Request to the SpaceX API*
- *Clean and normalization of the requested data*
- *Web Scraping*

  [Wikipedia: List of Falcon 9 and Falcon Heavy launches]

Web Collection process

- *Web scrap Falcon 9 launch records with `BeautifulSoup`*
- *Extract a Falcon 9 launch records HTML table from Wikipedia*
- *Parse the table and convert it into a Pandas data frame*

More specifically, the launch records are stored in a HTML table shown below:

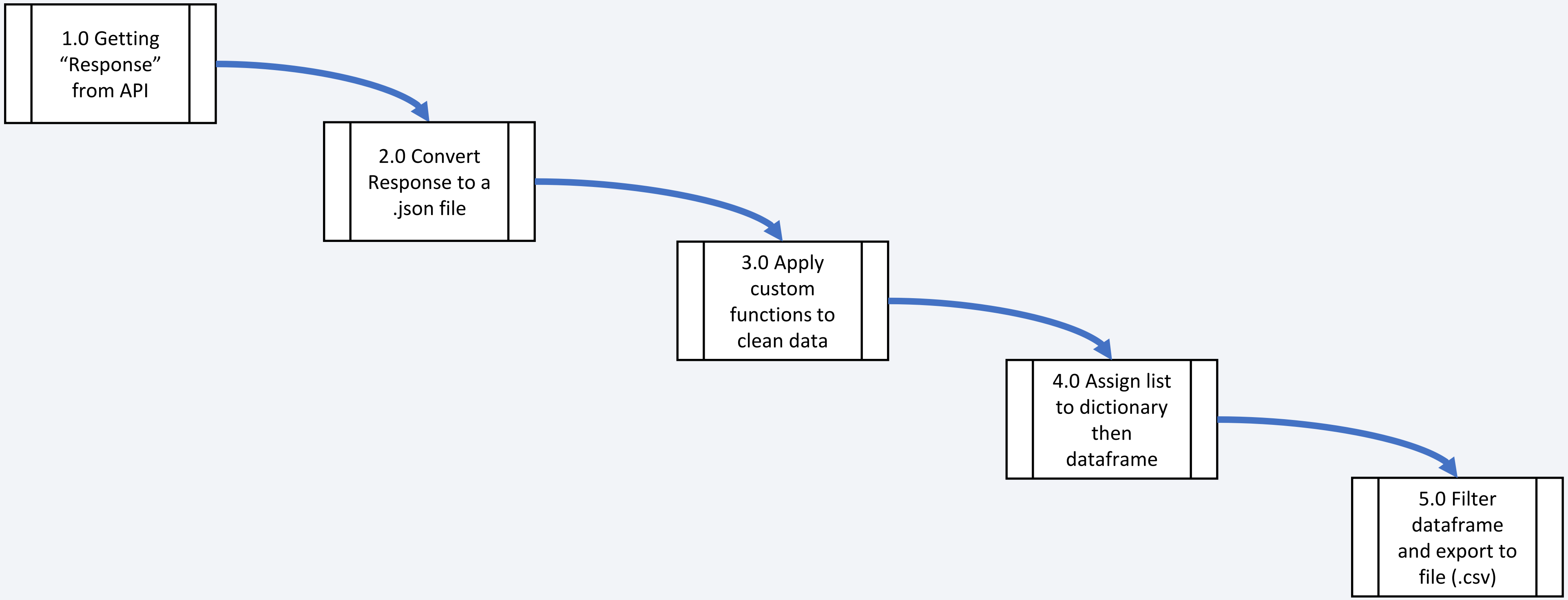
2020 [edit]

In late 2019, Gwynne Shotwell stated that SpaceX hoped for as many as 24 launches for Starlink satellites in 2020,[490] in addition to 14 or 15 non-Starlink launches. At 26 launches, 13 of which for Starlink satellites, Falcon 9 had its most prolific year, and Falcon rockets were second most prolific rocket family of 2020, only behind China's Long March rocket family.[491]

[hide] Flight No.	Date and time (UTC)	Version, Booster[b]	Launch site	Payload[c]	Payload mass	Orbit	Customer	Launch outcome	Booster landing
78	7 January 2020, 02:19:21[492]	F9 B5 ♺ B1049.4	CCAFS, SLC-40	Starlink 2 v1.0 (60 satellites)	15,600 kg (34,400 lb)[5]	LEO	SpaceX	Success	Success (drone ship)
	Third large batch and second operational flight of Starlink constellation. One of the 60 satellites included a test coating to make the satellite less reflective, and thus less likely to interfere with ground-based astronomical observations.[493]								
79	19 January 2020, 15:30[494]	F9 B5 ♺ B1046.4	KSC, LC-39A	Crew Dragon in-flight abort test[495] (Dragon C205.1)	12,050 kg (26,570 lb)	Sub-orbital[496]	NASA (CTS)[497]	Success	No attempt
	An atmospheric test of the Dragon 2 abort system after Max Q. The capsule fired its SuperDraco engines, reached an apogee of 40 km (25 mi), deployed parachutes after reentry, and splashed down in the ocean 31 km (19 mi) downrange from the launch site. The test was previously slated to be accomplished with the Crew Dragon Demo-1 capsule;[498] but that test article exploded during a ground test of SuperDraco engines on 20 April 2019.[419] The abort test used the capsule originally intended for the first crewed flight.[499] As expected, the booster was destroyed by aerodynamic forces after the capsule aborted.[500] First flight of a Falcon 9 with only one functional stage — the second stage had a mass simulator in place of its engine.								
80	29 January 2020, 14:07[501]	F9 B5 ♺ B1051.3	CCAFS, SLC-40	Starlink 3 v1.0 (60 satellites)	15,600 kg (34,400 lb)[5]	LEO	SpaceX	Success	Success (drone ship)
	Third operational and fourth large batch of Starlink satellites, deployed in a circular 290 km (180 mi) orbit. One of the fairing halves was caught, while the other was fished out of the ocean.[502]								
81	17 February 2020, 15:05[503]	F9 B5 ♺ B1056.4	CCAFS, SLC-40	Starlink 4 v1.0 (60 satellites)	15,600 kg (34,400 lb)[5]	LEO	SpaceX	Success	Failure (drone ship)
	Fourth operational and fifth large batch of Starlink satellites. Used a new flight profile which deployed into a 212 km × 386 km (132 mi × 240 mi) elliptical orbit instead of launching into a circular orbit and firing the second stage engine twice. The first stage booster failed to land on the drone ship[504] due to incorrect wind data.[505] This was the first time a flight proven booster failed to land.								
82	7 March 2020, 04:50[506]	F9 B5 ♺ B1059.2	CCAFS, SLC-40	SpaceX CRS-20 (Dragon C112.3 ♺)	1,977 kg (4,359 lb)[507]	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
	Last launch of phase 1 of the CRS contract. Carries *Bartolomeo*, an ESA platform for hosting external payloads onto ISS.[508] Originally scheduled to launch on 2 March 2020, the launch date was pushed back due to a second stage engine failure. SpaceX decided to swap out the second stage instead of replacing the faulty part.[509] It was SpaceX's 50th successful landing of a first stage booster, the third flight of the Dragon C112 and the last launch of the cargo Dragon spacecraft.								
83	18 March 2020, 12:16[510]	F9 B5 ♺ B1048.5	KSC, LC-39A	Starlink 5 v1.0 (60 satellites)	15,600 kg (34,400 lb)[5]	LEO	SpaceX	Success	Failure (drone ship)
	Fifth operational launch of Starlink satellites. It was the first time a first stage booster flew for a fifth time and the second time the fairings were reused (Starlink flight in May 2019).[511] Towards the end of the first stage burn, the booster suffered premature shut down of an engine, the first of a Merlin 1D variant and first since the CRS-1 mission in October 2012. However, the payload still reached the targeted orbit.[512] This was the second Starlink launch booster landing failure in a row, later revealed to be caused by residual cleaning fluid trapped inside a sensor.[513]								
84	22 April 2020, 19:30[514]	F9 B5 ♺ B1051.4	KSC, LC-39A	Starlink 6 v1.0 (60 satellites)	15,600 kg (34,400 lb)[5]	LEO	SpaceX	Success	Success (drone ship)

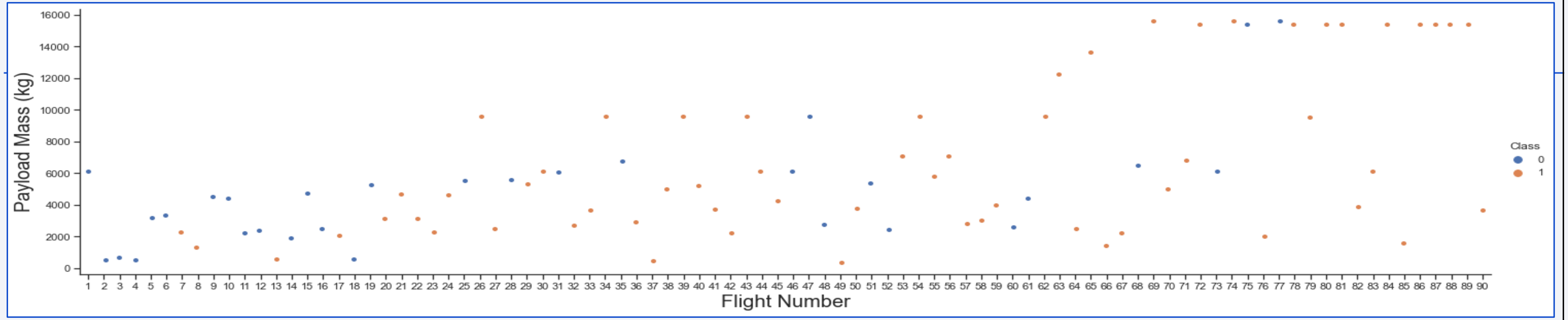
Data Collection: Scraping SpaceX API

Data Collection: Scraping flowchart

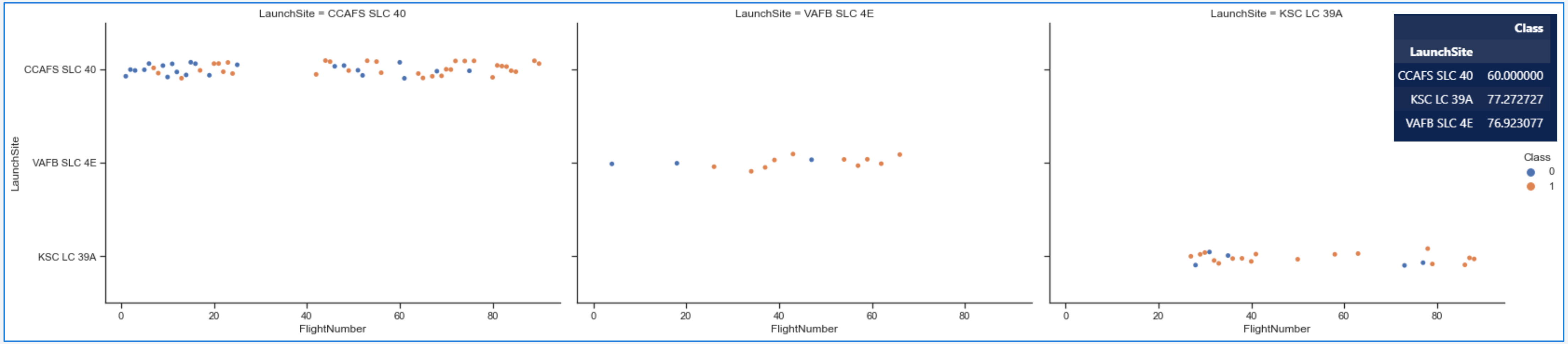


GitHub URL Code: [Data Collection.ipynb]

EDA with Data Visualization



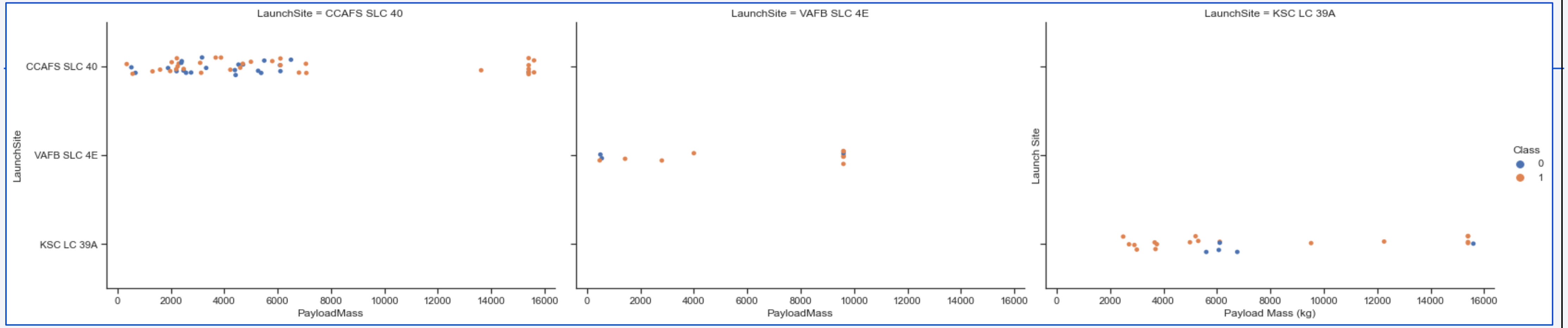
First, let's try to see how the `Flight` (indicating the continuous launch attempts.) and `Payload` variables would affect the launch outcome. Overlaying launch outcomes *'0'* failed vs. *'1'* successful. We see that as the flight increases, the first stage is more likely to land successfully. The payload is also important; it seems the more massive the payload, the less likely the first stage will return.



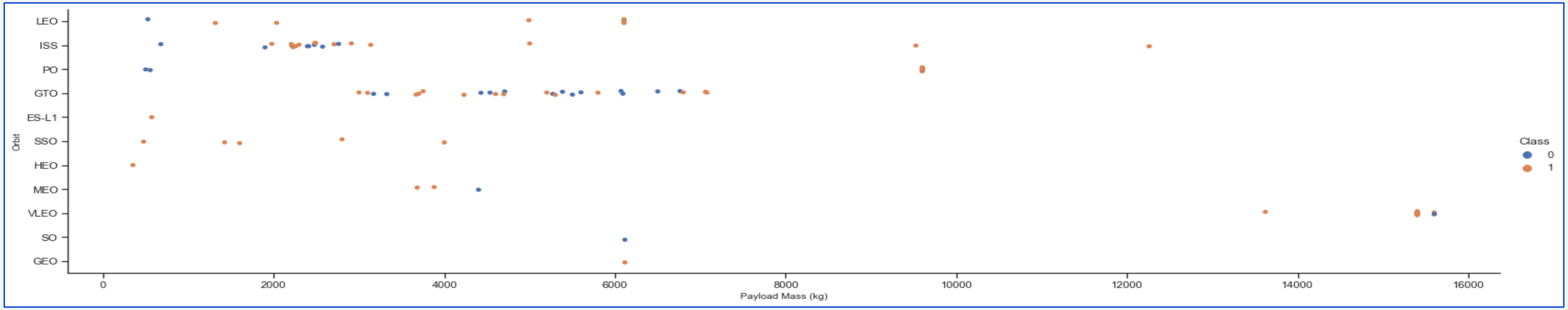
Visualize the relationship between Flights and Launch Site. We see that different launch sites have different success rates. CCAFS LC-40, has a success rate of 60 %, while KSC LC-39A and VAFB SLC 4E has a success rate of 77%.

GitHub URL Code: [jupyter-labs-eda-dataviz-IBM-WIP-002.ipynb]

EDA with Data Visualization (cont.)



We also want to observe if there is any relationship between launch sites and their payload mass. Now if you observe Payload Vs. Launch Site scatter point chart you will find for the VAFB-SLC launch site there are no rockets launched for heavy payload mass(greater than 10000).



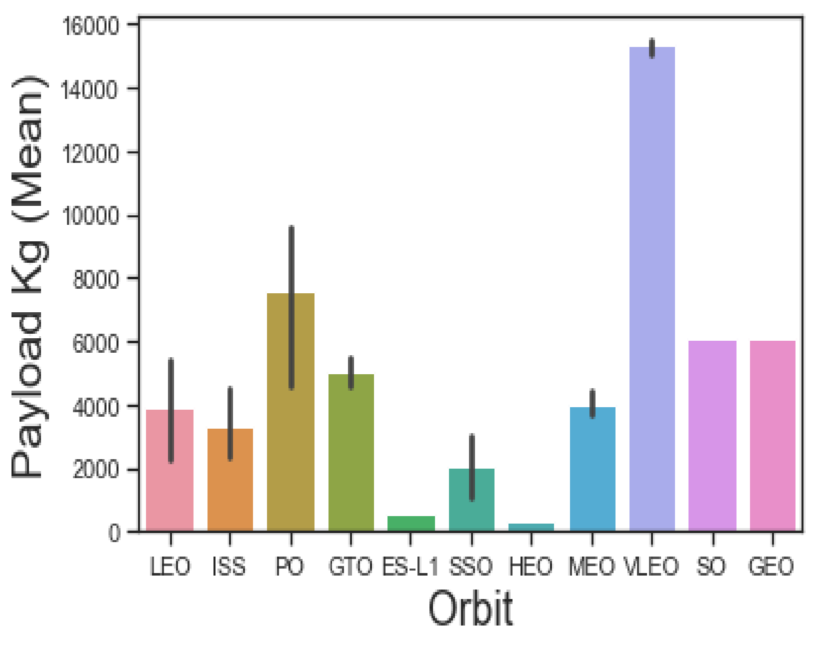
Similarly, we can plot the Orbit vs. Payload scatter point charts to reveal the relationship between Payload and Orbit type. With heavy payloads the successful landing rate are more for Polar, LEO and ISS orbits. However, for GTO we cannot distinguish this well as both positive and negative landing are both there here.

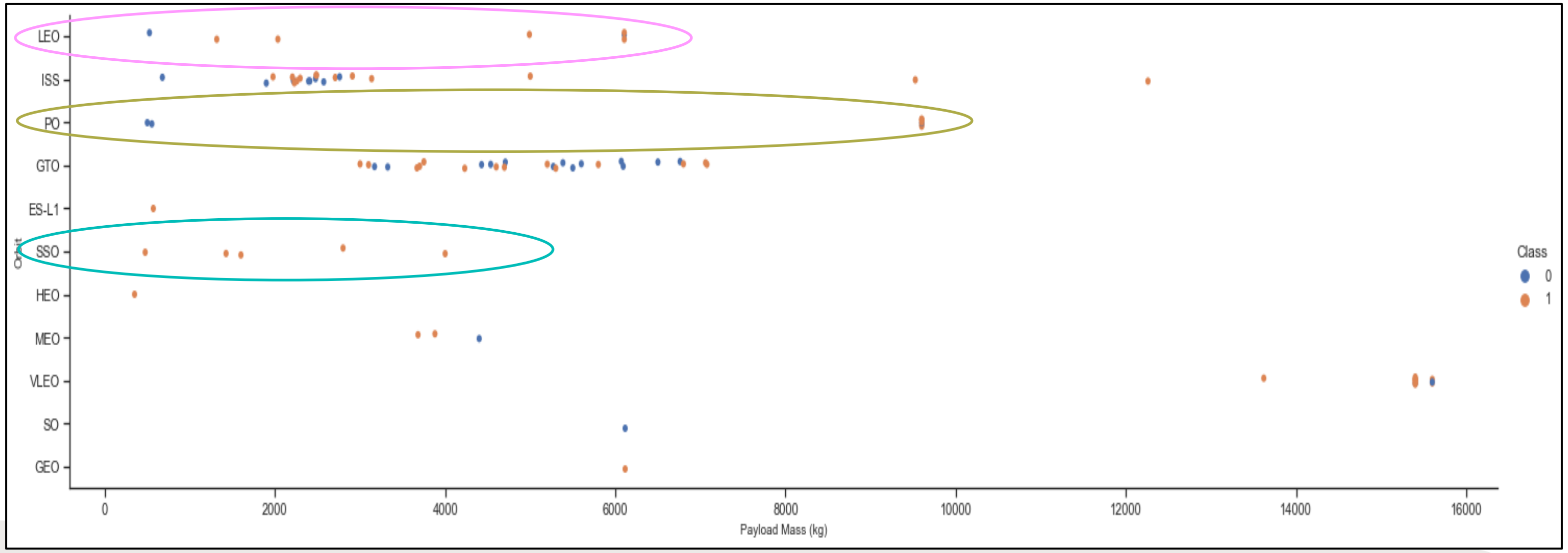
GitHub URL Code: [jupyter-labs-eda-dataviz-IBM-WIP-002.ipynb]

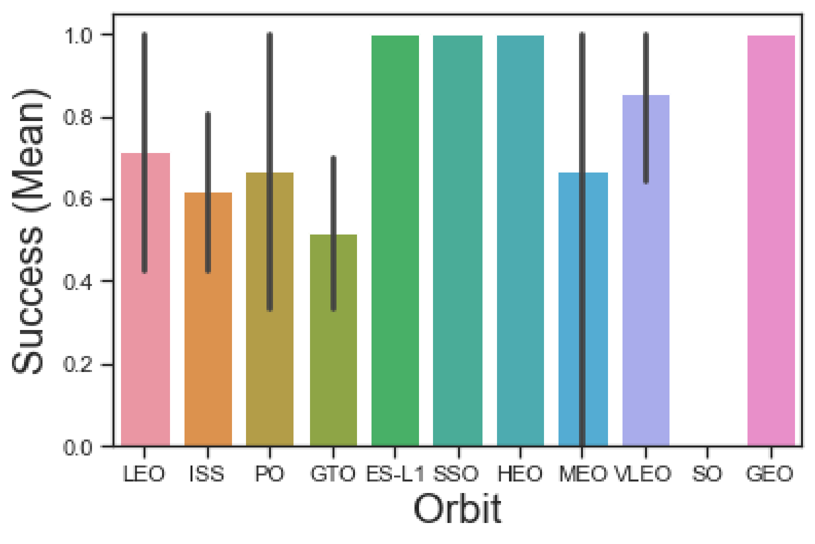
EDA with Data Visualization (cont.)

We also want to observe if there is any relationship between orbit and their payload mass. Now if you observe orbit bar chart launched payload mass(less than 10000) has a higher payload to launches rate of success.

Orbit	Class(Mean)	Payload Kg(Mean)
ES-L1	1.000000	570.000000
GEO	1.000000	6104.959412
GTO	0.518519	5011.994444
HEO	1.000000	350.000000
ISS	0.619048	3279.938095
LEO	0.714286	3882.839748
MEO	0.666667	3987.000000
PO	0.666667	7583.666667
SO	0.000000	6104.959412
SSO	1.000000	2060.000000
VLEO	0.857143	15315.714286







GitHub URL Code: [jupyter-labs-eda-dataviz-IBM-WIP-002.ipynb]

EDA with SQL

- 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 achieved
- 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

GitHub URL Code: [jupyter-labs-eda-sql-coursera-IBM-WIP-002.ipynb]

Build an Interactive Map with Folium

The launch success rate may depend on several factors such as payload mass, orbit type, and so on. It may also depend on the location and proximities of a launch site, i.e., the initial position of rocket trajectories. Finding an optimal location for building a launch site certainly involves many factors and hopefully we could discover some of the factors by analyzing the existing launch site locations. Using an interactive Map like Python Folium library we were able to perform the following;

- ✓ Mark all launch sites on a map
- ✓ Mark the success/failed launches for each site on the map
- ✓ Calculate the distances between a launch site to its proximities to different landmarks

The `spacex_launch_geo.csv` is an augmented dataset with latitude and longitude added for each site. We then visualized those locations by pinning them on a map using `folium.Circle` to add a highlighted circle area with a text label on a specific coordinate. Then using "*folium.map.Marker*" to mark for all launch records. If a launch was successful `*(class=1)*`, it is marked a ***green*** and if a launch was failed, it is marked ***red*** `*(class=0)*`. By using color-labeled markers, you should be able to easily identify which launch sites have relatively high success rates. Similarly, using the `*MousePosition*` to find different landmarks coordinates on the map we are able calculate and draw a distance line between a launch site to its closest landmark e.g., city, railway, highway, etc. After you plot distance lines to the proximities, you can easily visualize and draw conclusions on the launch site and their locations;

- ✓ 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
- ✓ Do launch sites keep certain distance away from cities? Yes

GitHub URL Code: lab_jupyter_launch_site_location-IBM-WIP-002.ipynb

Build a Dashboard with Plotly Dash

- *Dash* apps give a point-&-click interface to models written in Python. *Plotly Dash* apps put complex Python analytics into an interactive easy to use UI. This is powerful too for developers to put data into the hands of business decision makers and operators.
- For our data analysis project, we built a *Plotly Dash* application for users to perform interactive visual analytics on SpaceX launch data in real-time. This dashboard contains interactive modules such as a dropdown list and a range slider. These interact with a pie and a scatter chart. The dashboard application provides the following:
  - Launch Site Drop-down
  - Outcome pie-chart based on selected Launch Site Drop-down
  - Payload Mass Kg (0 to 10k) range slider
  - Launch outcome/payload scatter plot for the different Booster Version Category
- Pie Chart showing the outcome launches site or all sites combined
- Scatter Graph showing the relationship with launch outcome and payload mass (Kg)

GitHub URL Code: [spacex_dash_app_002.py]

Predictive Analysis (Classification)

Model built

GitHub URL Code: [SpaceX_Machine Learning Prediction_Part_5_IBM-WIP.ipynb]

- ✓ Load the dataframe
- ✓ Create a *NumPy* array from the column ***Class*** in data, by applying the method "*to_numpy()*" then assign it to the variable *Y*, make sure the output is a *Pandas* series
- ✓ Standardize the data in *X* then reassign it to the variable *X*
- ✓ Split training and test data using the function "*train_test_split*". The training data is divided into validation and training data. Setting the parameter "*test_size*" to ***20%*** and "*random_state*" to ***2***.
- ✓ Assigned training and test data to the following labels to; *X_train, X_test, Y_train, Y_test* for model evaluation

Evaluated, Improved models

- ✓ Fit the Train and Test objects into each of the classification model to find the best parameters
- ✓ Calculate the accuracy on the test data using the method "*score()*"

Find best performing classification model

- ✓ Find best accuracy score of classification model; SVM, KNN, Classification Trees and Logistic Regression
  - ❖ {'SVM': 0.832142857142857, 'KNN': 0.8446428571428569, 'Tree': 0.8875, 'LogReg': 0.8035714285714285}

Results Content & Visualization

Exploratory data analysis results

Interactive analytics demo in screenshots

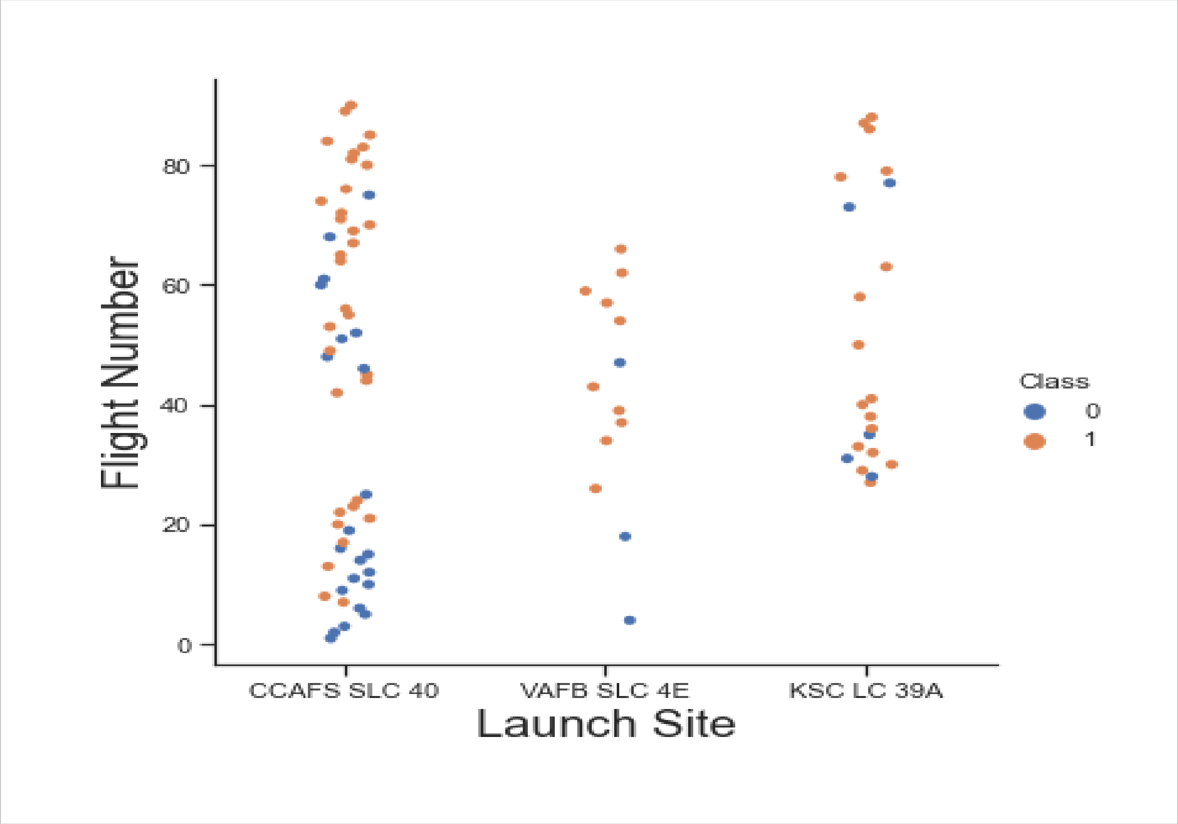
Predictive analysis results

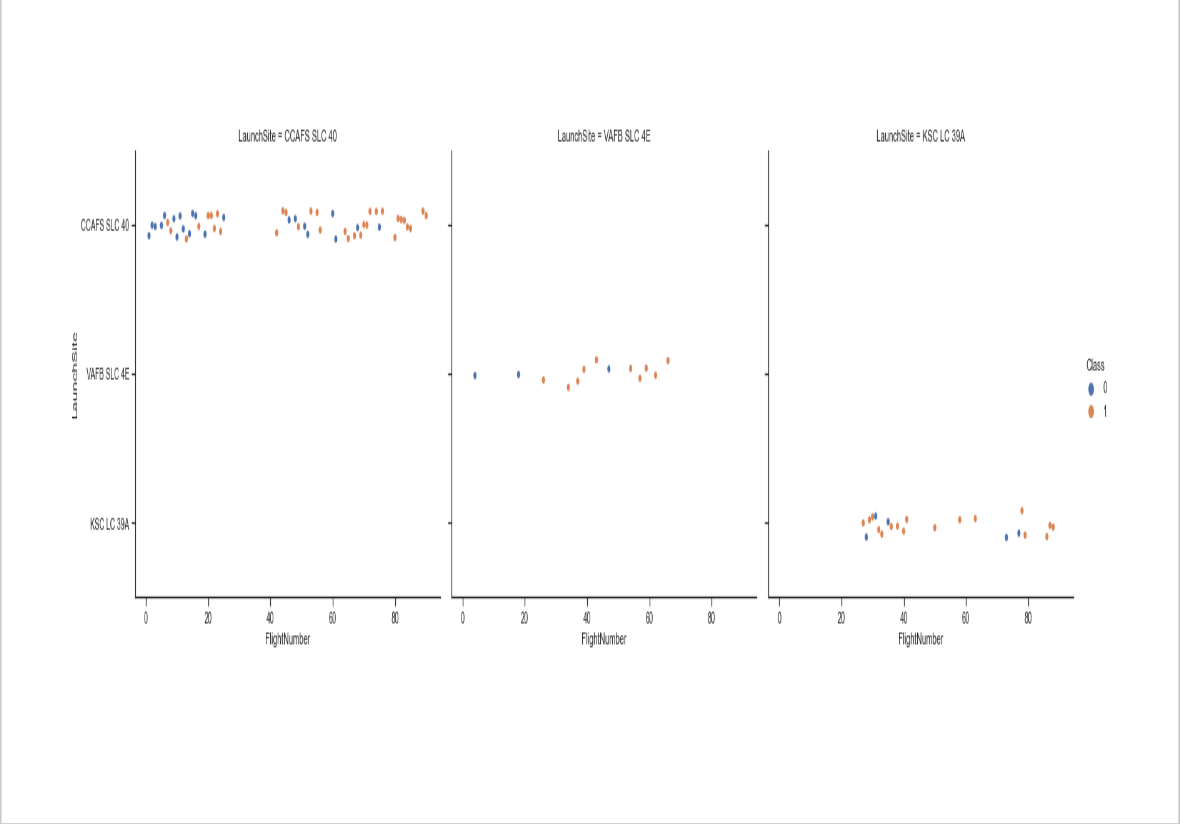
Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

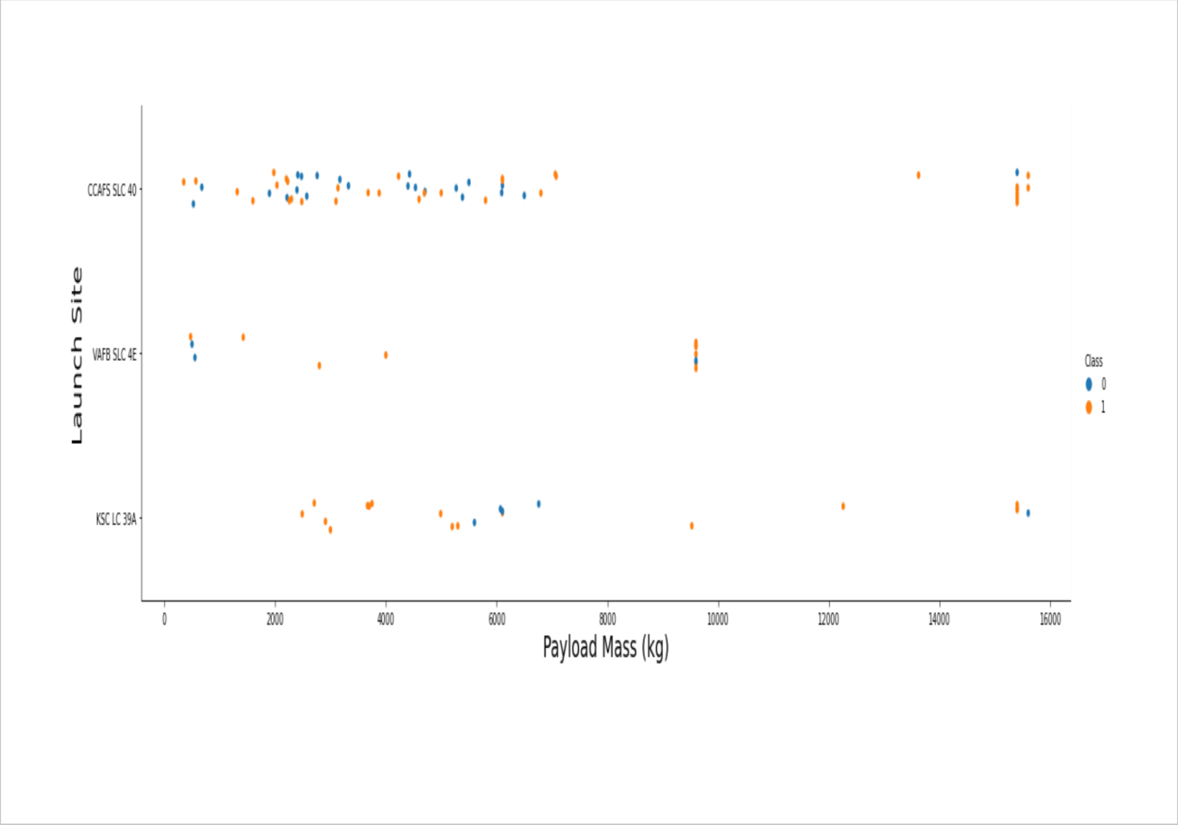
Observed as the number of Launches increases the success rate also increase. There is a direct relationship between these two variables. This can be attributed to lessons learned from earliest lunches.

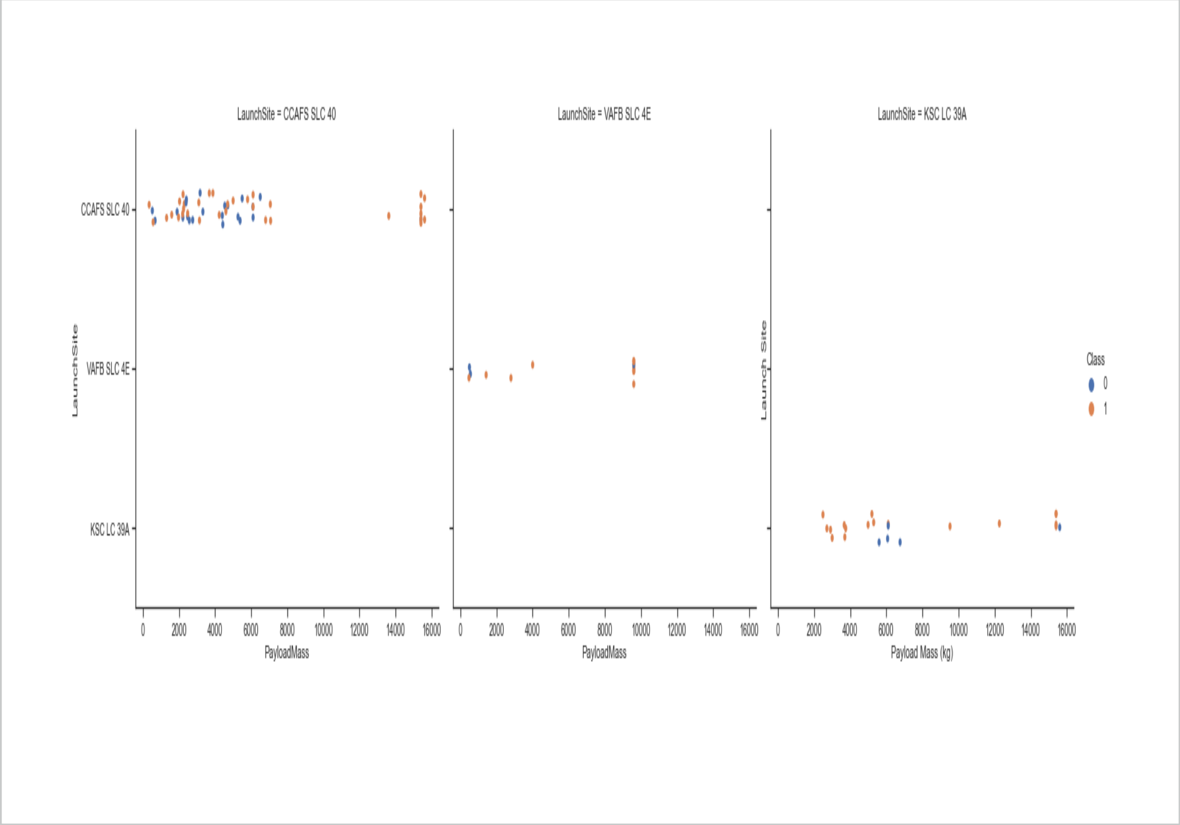




Payload vs. Launch Site

Observed the success rate also is higher for payload under 10,000 Kg. There is a relationship between these two variables. This can also be attributed to lessons learned from earliest lunches.

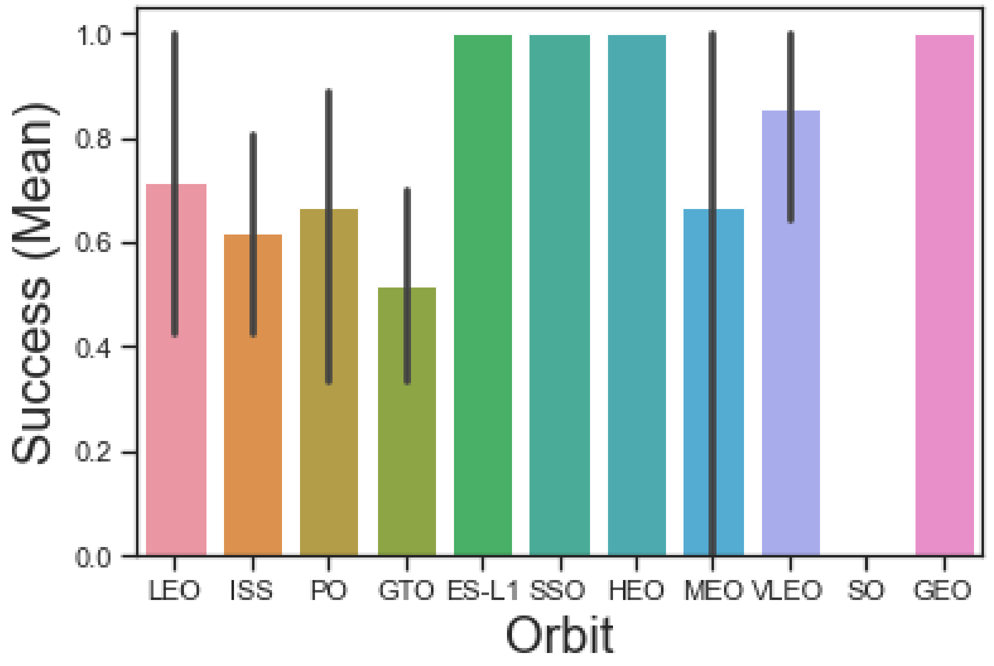


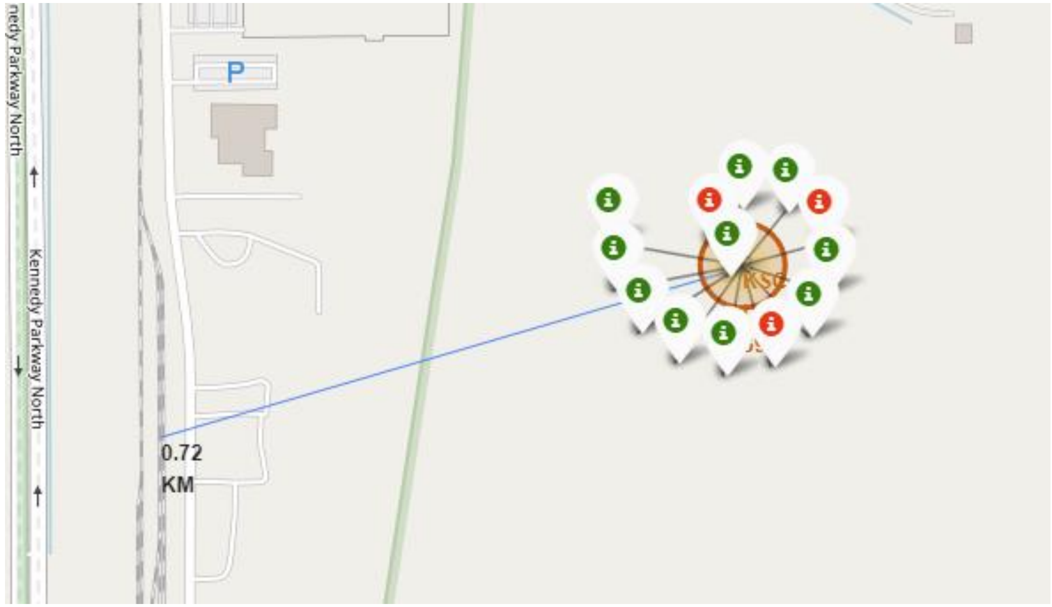


Success Rate vs. Orbit Type

Observed the success (Mean) rate is higher for LEO, ISS and PO. There is a relationship between these two variables. This can also be attributed to lessons learned from earliest lunches. When observing the per site success rate perhaps launch/landing site location can be consider as a success factor.

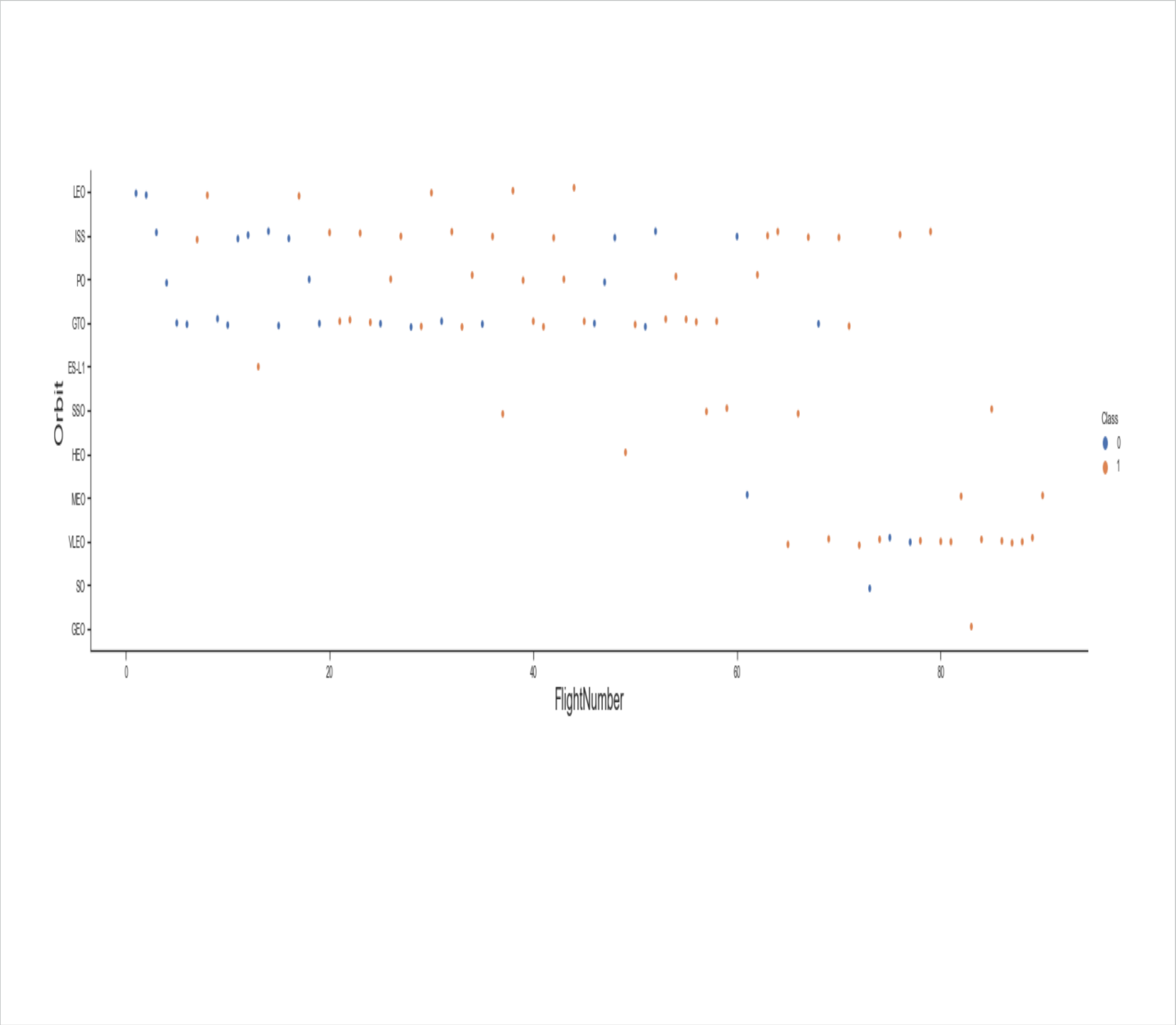
	Class(Mean)
LaunchSite	
CCAFS SLC 40	0.600000
KSC LC 39A	0.772727
VAFB SLC 4E	0.769231





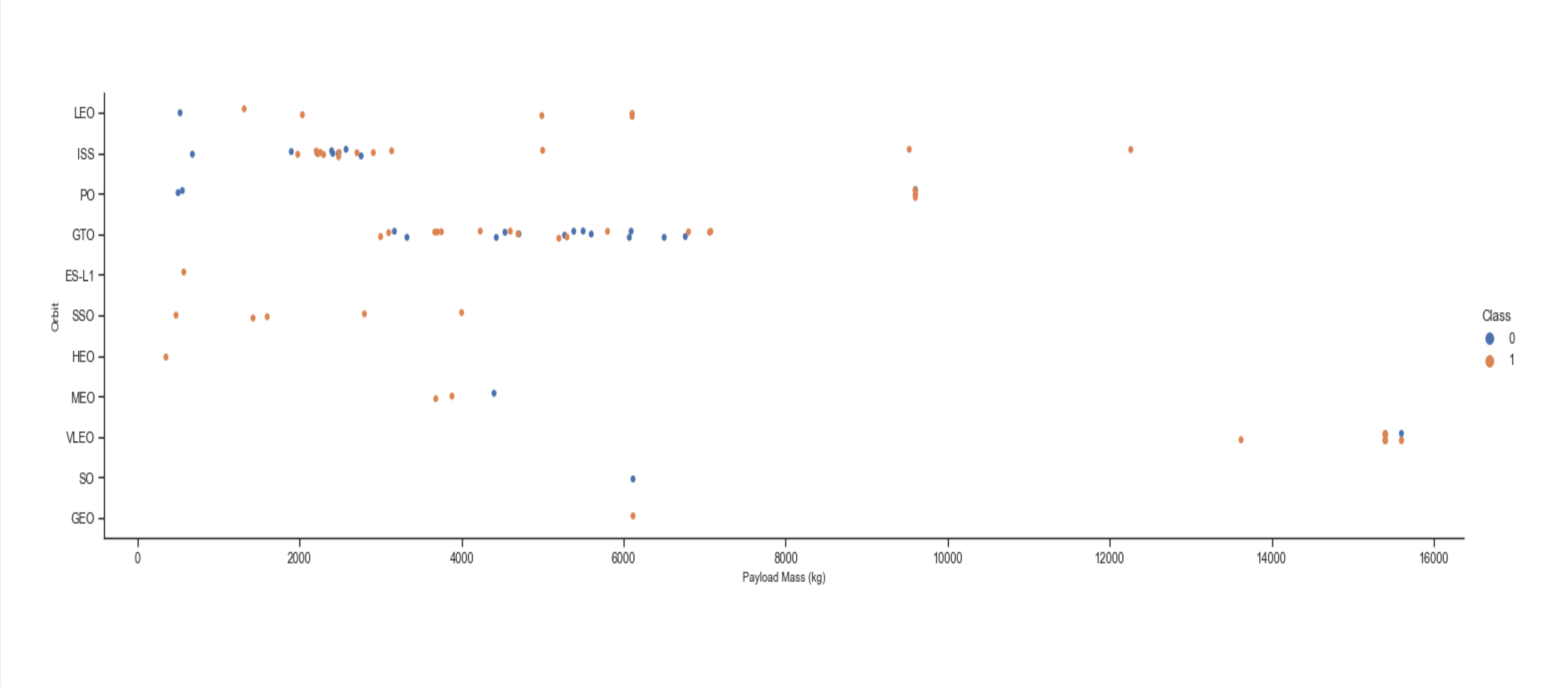
Flight Number vs. Orbit Type

Observed the success (Mean) rate is higher for LEO, ISS and PO. There is a relationship between these two variables. This can also be attributed to lessons learned from earliest lunches. When observing the per site success rate perhaps launch/landing site location can be consider as a success factor.



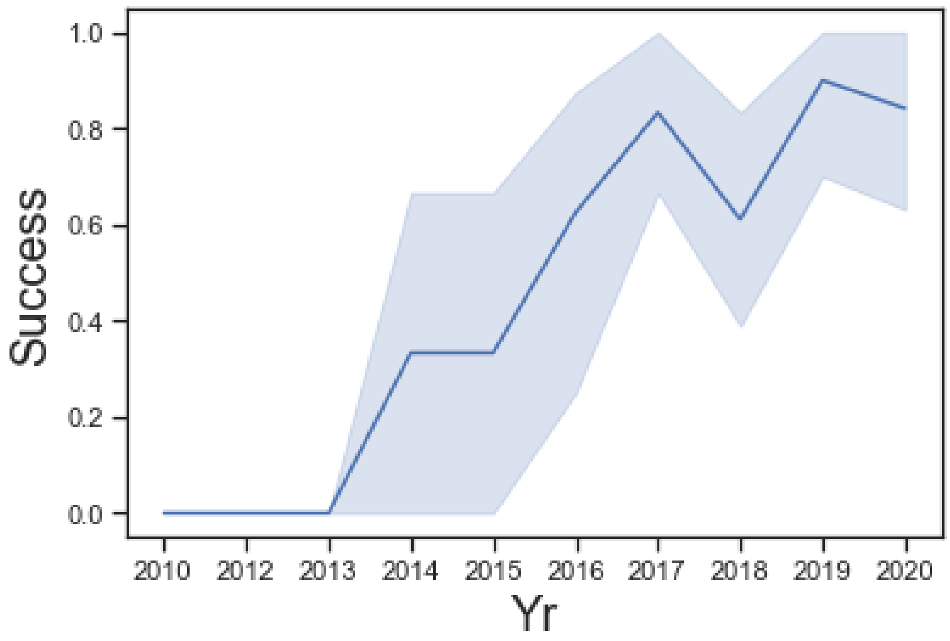
Payload vs. Orbit Type

Observed the success rate also is higher for payload under 10,000 Kg. There is a relationship between these two variables. This can also be attributed to lessons learned from earliest lunches.



Launch Success Yearly Trend

Observed the success rate is trending upward. This can also be attributed to lessons learned from earliest lunches.



All Launch Site Names

Result

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

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

%sql select DISTINCT launch_site from SPACEXDATASET

Explanation

Using "*DISTINCT*" will show one instance or unique result in the launch_site column

Launch Site Names Begin with 'CCA'

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

*%sql select * from SPACEXDATASET where launch_site like 'CCA%' limit 05*

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

Explanation

Using the command “LIKE” will show instance values that contain “CCA” string in the *“launch_site”* column

Total Payload Mass

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

%sql select sum(payload_mass__kg_) as "TotalPayloadMass"

from spacexdataset where customer = 'NASA (CRS)'

Result

totalpayloadmass
45596

Explanation

Using "*SUM*" calculate the total payload mass result launched

from the NASA(CRS) launch site

Average Payload Mass by F9 v1.1

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

```
%sql select avg(payload_mass__kg_) as "AVGPayloadMass"
from spacexdataset where booster_version = 'F9 v1.1'
```

Result

avgpayloadmass
2928

Explanation

"AVG" calculate the average payload mass result launched

using the booster version F9 v1.1

First Successful Ground Landing Date

List the date when the first successful landing outcome in ground pad was achieved.

%sql select min(date) as "date" from spacexdataset where mission_outcome = 'Success' and landing__outcome = 'Success (ground pad)'

Result

DATE
2015-12-22

Explanation

Using the function "*MIN(date)*" results in the minimum date(earliest date) from the column Date or first successful ground landing

Successful Drone Ship Landing with Payload between 4000 and 6000

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 payload_mass__kg_, (booster_version) as "Booster_Name" from spacexdataset where payload_mass__kg_ > 4000 and payload_mass__kg_ < 6000 and landing__outcome = 'Success (drone ship)'

Explanation

Displaying Booster_Version “*WHERE*” filtered by *Success (drone ship) landing_outcome* “*AND*” Payload_MASS_KG_ between 4000 “*AND*” 6000

Result

payload_mass__kg_	booster_name
4696	F9 FT B1022
4600	F9 FT B1026
5300	F9 FT B1021.2
5200	F9 FT B1031.2

Total Number of Successful and Failure Mission Outcomes

List the total number of successful and failure mission outcomes

%sql select (mission_outcome), count(*) from spacexdataset where mission_outcome like '%Success%' or mission_outcome like '%Failure%' group by mission_outcome

Result

mission_outcome	2
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

Explanation

Count number of success and failures and grouping then into mission_outcomes category

Boosters Carried Maximum Payload

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

%sql select booster_version as "BOOSTER", payload_mass__kg_ as "PAYLOAD" from spacexdataset where payload_mass__kg_ = (select max(payload_mass__kg_) from spacexdataset)

Explanation

Using a subquery to pull a list of booster where maximum payload using “*Max(Payload)*” command.

Result

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

2015 Launch Records

List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

%sql SELECT date, landing__outcome, booster_version, launch_site

from spacexdataset where date like '%2015%'

Result

DATE	landing__outcome	booster_version	launch_site
2015-01-10	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
2015-02-11	Controlled (ocean)	F9 v1.1 B1013	CCAFS LC-40
2015-03-02	No attempt	F9 v1.1 B1014	CCAFS LC-40
2015-04-14	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40
2015-04-27	No attempt	F9 v1.1 B1016	CCAFS LC-40
2015-06-28	Precluded (drone ship)	F9 v1.1 B1018	CCAFS LC-40
2015-12-22	Success (ground pad)	F9 FT B1019	CCAFS LC-40

Explanation

Using a query to pull a list of launches during 2015

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

%sql select mission_outcome as "Mission", count(mission_outcome) as "landing__outcome" from spacexdataset where (DATE between '2010-06-04' and '2017-03-20') group by mission_outcome ORDER BY landing__outcome DESC

Result

mission	landing__outcome
Success	30
Failure (in flight)	1

Explanation

Using a query to pull a list of missions *between '2010-06-04' and '2017-03-20'*

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

%sql select landing__outcome, (select count((landing__outcome)) from spacexdataset) from spacexdataset where landing__outcome like '%Success%' or mission_outcome like '%Failure%' and (date between '2010-06-04' and '2017-03-20') group by landing__outcome;

Result

landing__outcome	2
Precluded (drone ship)	101
Success	101
Success (drone ship)	101
Success (ground pad)	101

Explanation

Using a query to pull a list of *mission_outcomes between '2010-06-04' and '2017-03-20'*

Section 4

Launch Sites Proximities Analysis

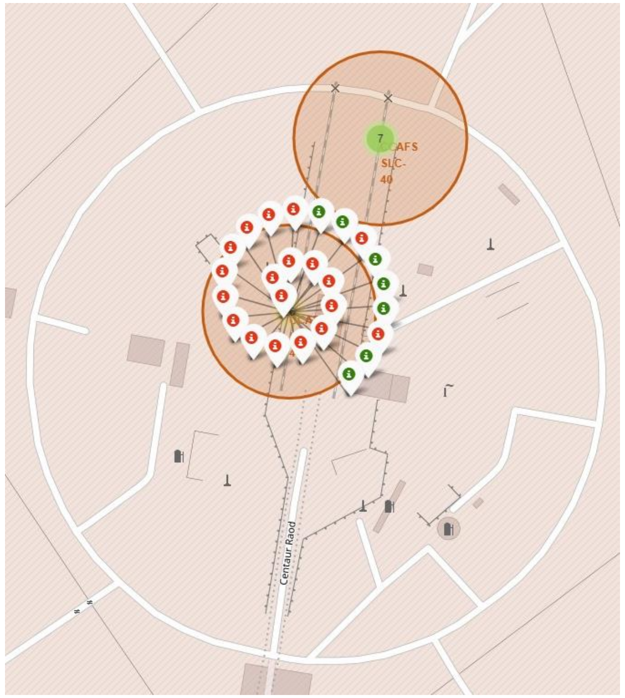
Folium Map East & West Launch Sites

Folium interactive map has features like zoom in/out, MousePosition longitude/latitude. These help users add markers very easily. Plotting distance to important landmarks. Also, it gives an Eagle-eyed view of what is around a launch site. In addition, helps with land survey allows you to understand your land boundaries, etc.

Folium Map East & West

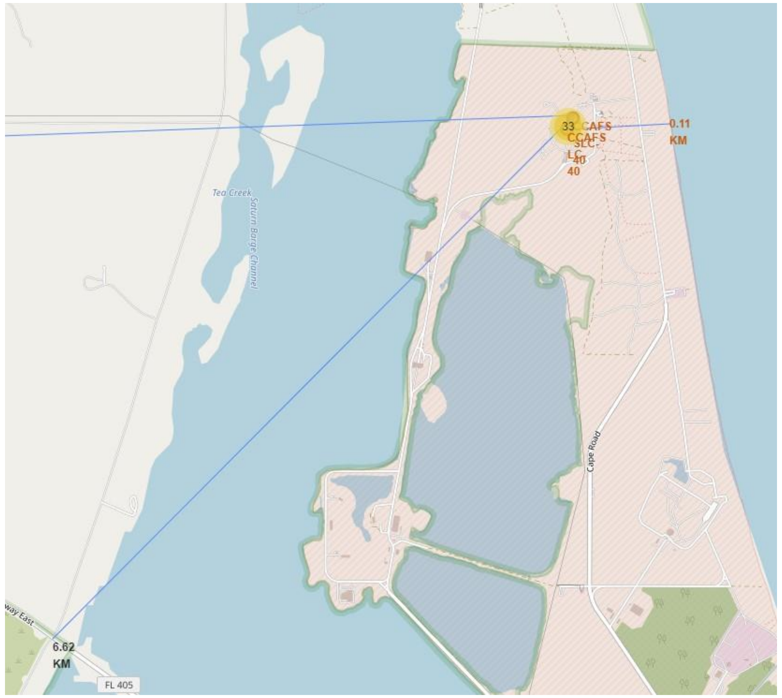
color-labeled launch outcomes on the map **(Failed / Successful)**

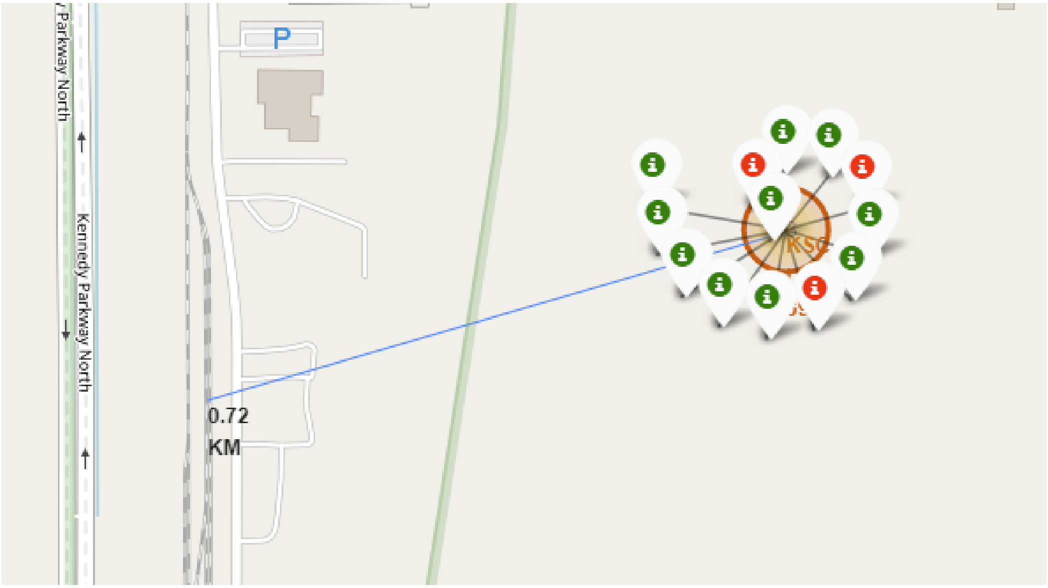
Adding color-coded markers is an easy way to quickly determine the number of launches and success rate. We can easily deduct that CCAFS LC-40 site has a higher rate of success to the number of launches.



Folium Map selected launch site proximities landmark distance

Folium interactive map has features like zoom in/out, MousePosition longitude/latitude. These help users add markers very easily. Plotting distance to important landmarks. Also, it gives a bird eye view of what is around a launch site. In addition, helps with land survey allows you to understand your land boundaries, etc.



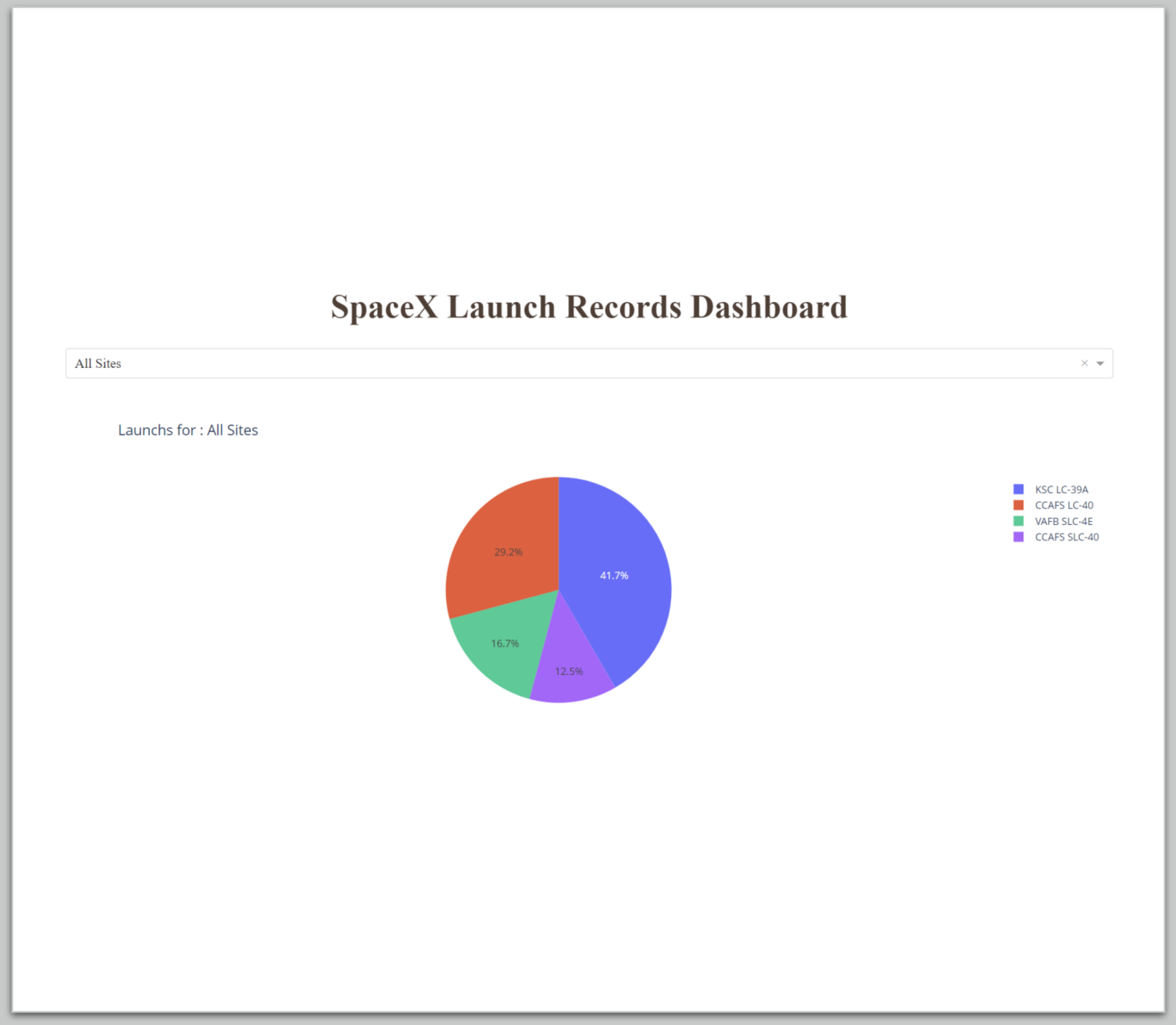


Section 5

Build a Dashboard with Plotly Dash

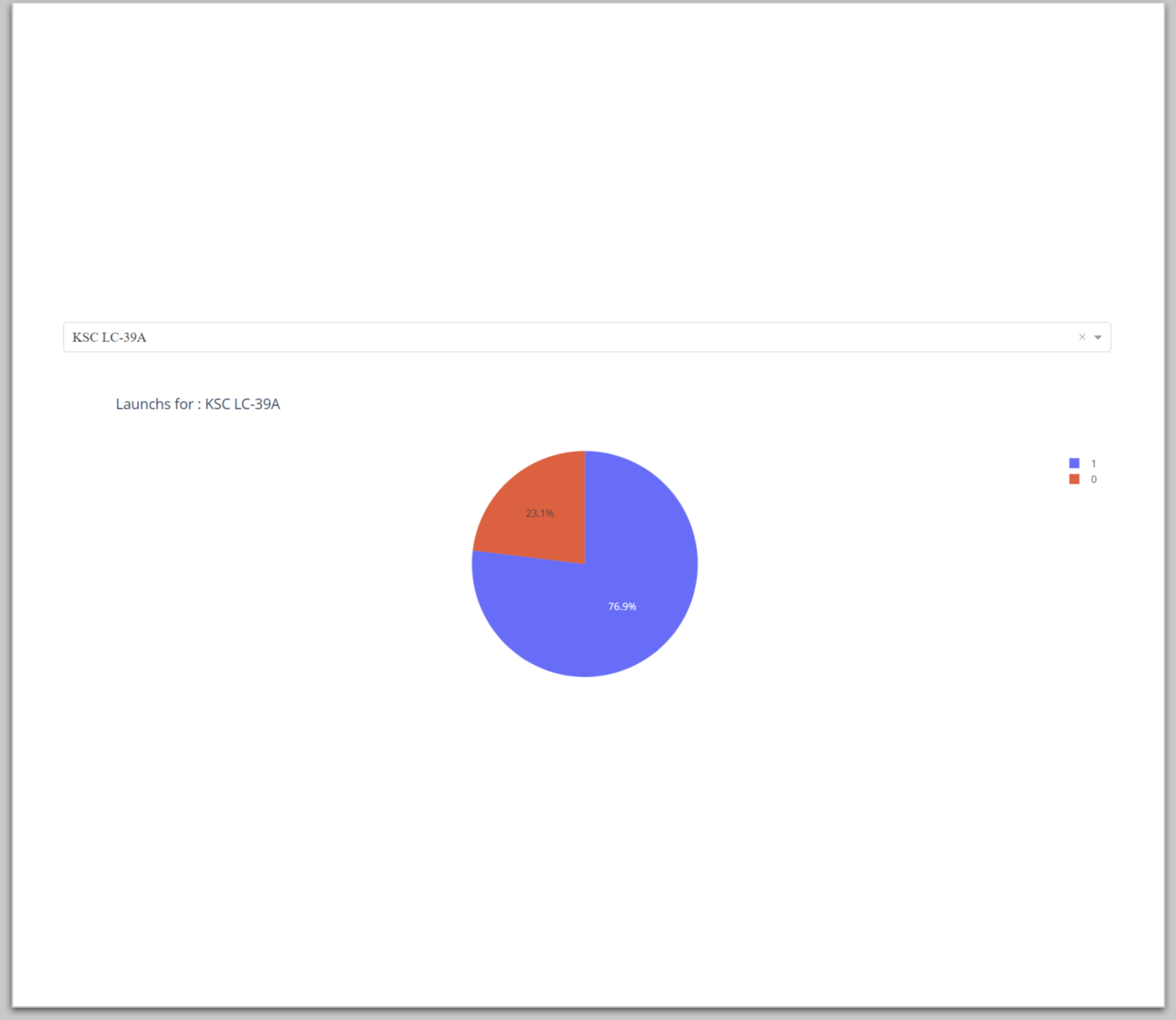
SpaceX Launch Records Dashboard All Sites

- **Explain the important elements and findings on the screenshot**
- We can easily observer how each launch sites rack and quickly see that KSC LC-39A has highest percentage of landing outcomes

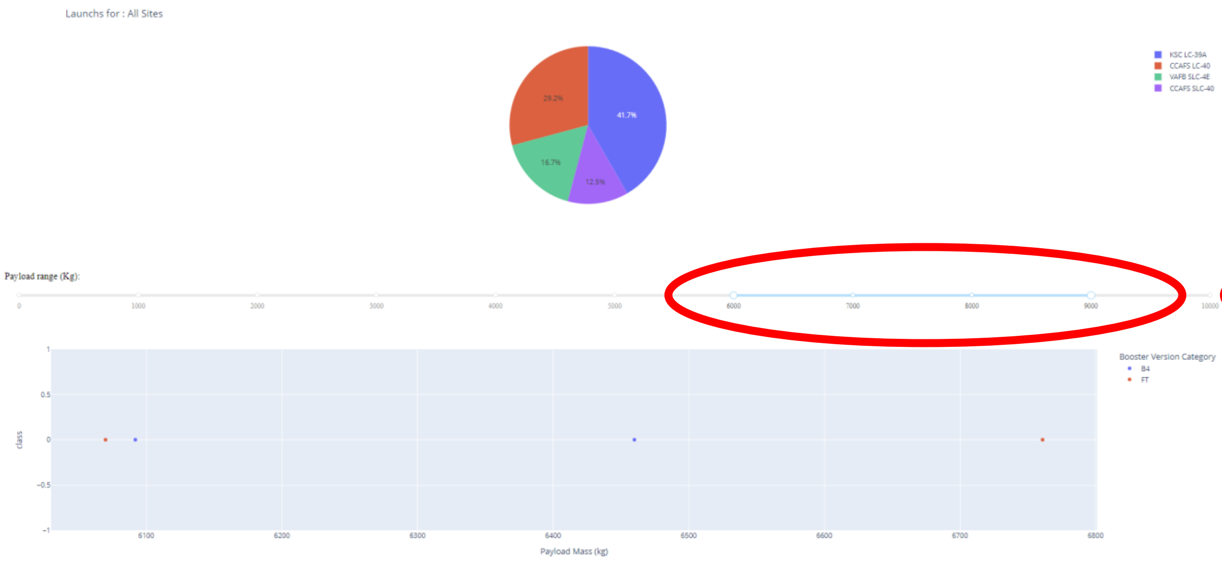


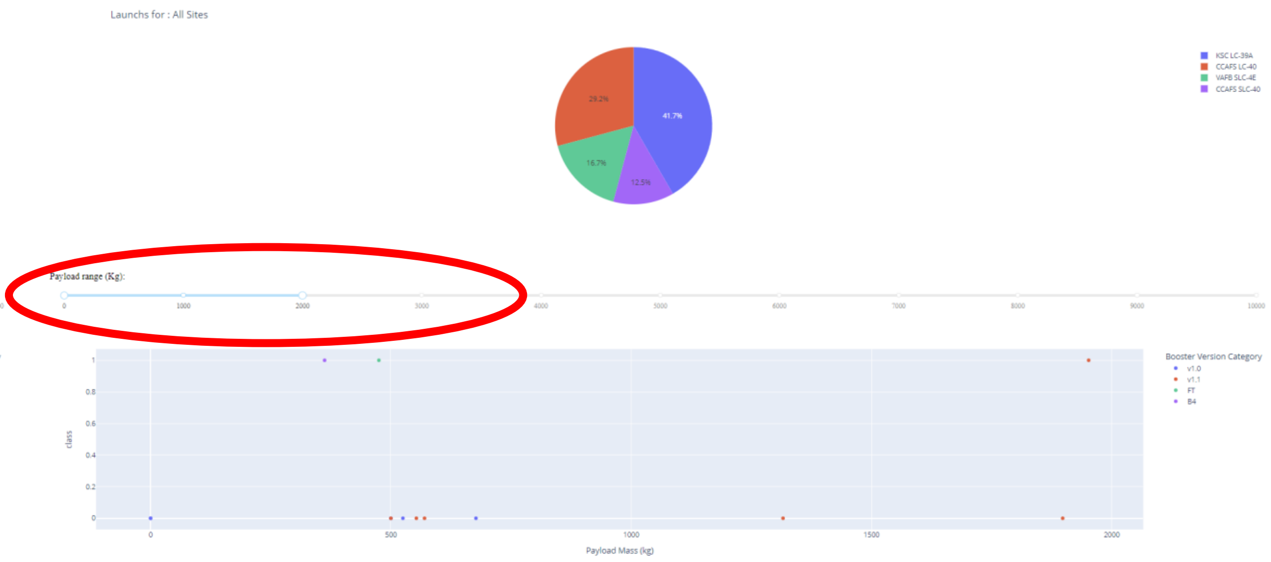
SpaceX Launch Records Dashboard with highest launch success ratio

- **Explain the important elements and findings on the screenshot**
- Pie chart showing KSC LC-39A has highest percentage of successful landing outcomes 76.9%



SpaceX Launch Records Dashboard Payload vs. Launch Outcome scatter plot at different payload range





Explain the important elements and findings on the screenshot

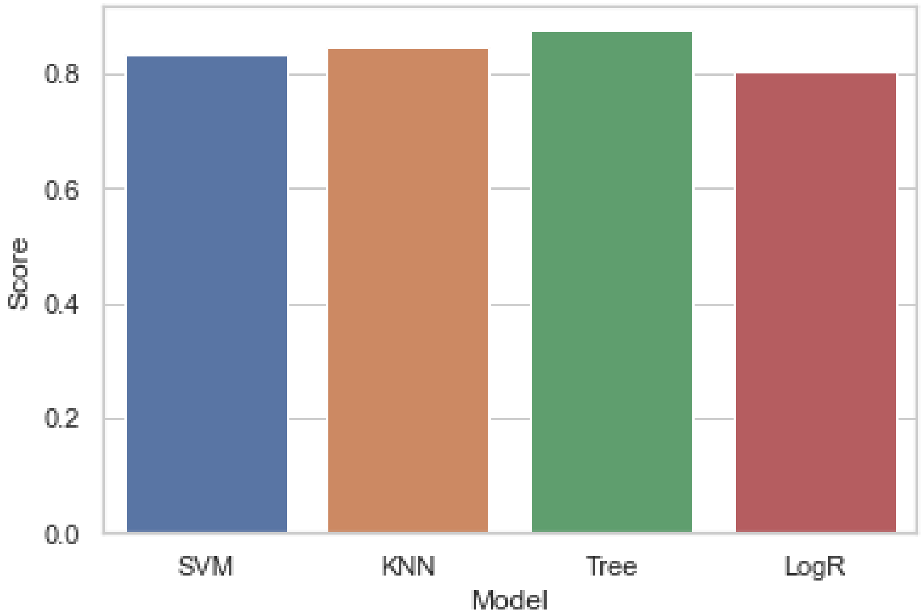
The interactive payload slider range gives the user the option to quickly answer which booster has the largest payload mass capacity. The above 2 screenshots are examples ranges of 6,000 to 9,000 kg and 0 kg and 2,000 kg.

Section 6

Predictive Analysis (Classification)

Classification Accuracy

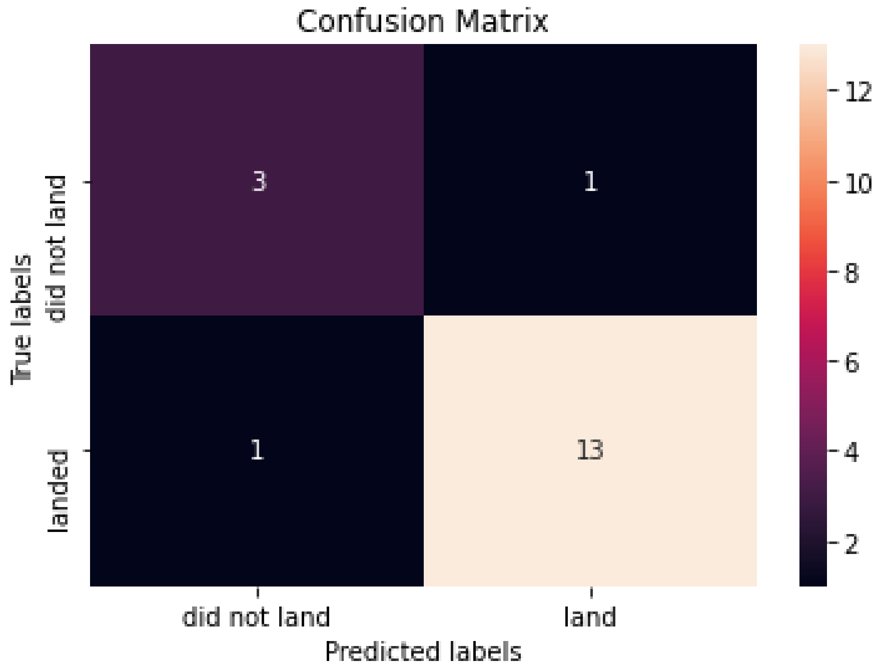
	Model	Scores
0	SVM	0.832143
1	KNN	0.844643
2	Tree	0.875000
3	LogR	0.803571

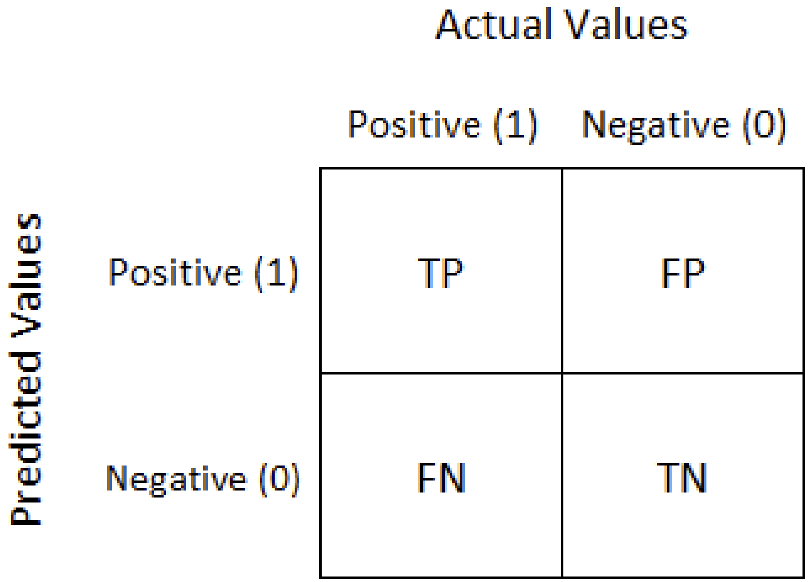


Best Algorithm / Method is Tree with a score of 0.875

Confusion Matrix

- By calculate the accuracy of tree_cv on the test data using the method *"score" and plotting the results into Confusion matrix it provides a quick visual of predicted values as Positive and Negative and actual values as True and False.*
- *Here Tree Decision Model predicted 13 saucerful landing vs. 3 unsuccessful landing*
- Confusion Matrix is a performance measurement for machine learning classification





Conclusions

- The Tree Classifier Algorithm has the highest accuracy rate with a score of 87.5%
- Payload launches with under 10,000 kg has a high rate of mission success
- Yearly launch success is treading upward
- KSC LC-39A had the most successful launches from all launch sites
- Orbit GEO, HEO, SSO, ES-L1 have the higher Success Rate

Thank you!

Appendix 1.0

[GitHub URL Code Master Repository]

Now that our `features_one_hot` dataframe only contains numbers cast the entire dataframe to variable type `float64`

```
# HINT: use astype function
one_hot.astype('float64')
one_hot
```

✓ 0.3s

	Orbit	LaunchSite	LandingPad	Serial
0	0	1	0	1
1	0	1	0	0
2	0	1	0	0
3	0	0	0	0
4	0	1	0	0
...	...	...	...	...
85	0	0	0	0
86	0	0	0	0
87	0	0	0	0
88	0	1	0	0
89	0	1	0	0

90 rows × 4 columns

GitHub URL Code: jupyter-labs-eda-dataviz-IBM-WIP-002.ipynb

Appendix 1.2

[GitHub URL Code Master Repository]

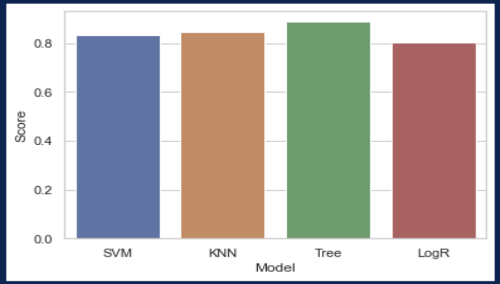
```python
# Creating df with score results e.g. df = pd.DataFrame(data=[[pi, e, phi]], columns=['pi', 'e, 'phi'])
# Examples
# Constructing DataFrame from a dictionary.
# d = {'col1': [1, 2], 'col2': [3, 4]}
# df = pd.DataFrame(data=d)
# df.insert(2, 'new-col', data)
#
score_df = pd.DataFrame([svm_cv.best_score_, knn_cv.best_score_, tree_cv.best_score_, logreg_cv.best_score_])
score_df.insert(0,'Model', ['SVM','KNN','Tree','LogR'])
score_df.rename(columns={0 : 'Scores'},inplace=True)
score_df
```

22] ✓ 0.3s

	Model	Scores
0	SVM	0.832143
1	KNN	0.844643
2	Tree	0.887500
3	LogR	0.803571

```python
# Visualize the built model accuracy for all built classification models, in a bar chart
sns.set_theme(style="whitegrid")
sns.barplot (x="Model", y="Scores", data=score_df)
plt.xlabel("Model")
plt.ylabel("Score")
plt.show()
```

[224] ✓ 0.9s



GitHub URL Code: jupyter-labs-eda-dataviz-IBM-WIP-002.ipynb