IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Jeffrey Strickland, Ph.D.
24 November 2021

Outline

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

Executive Summary

- Can SpaceX maintain their OPTEMPO under current first-stage recovery operations?
- Using an accurate predictive model, SpaceX can recover 76% of launched stage-one boosters
- Data analytics
    - Data collection – Web scraping, Wikipedia, internal source searches, etc.
    - Data wrangling – data cleaning, preprocessing, normalizing
    - Exploratory data analysis (EDA)
        - Structured query language (SQL) for data mining
        - Visualization using interactive charts and maps
- Analysis Methodology
    - Using insights from ED
    - Dashboard development
    - Predictive modeling and analytics
    - Multiple classifier models
    - Training the models
    - Hyperparameter tuning to find best model parameters
    - Scoring models and best choice

Introduction

- Since the cancellation of NASA’s manned-space flight program in 2010, this role was captured by the commercial industry, led by SpaceX. Due to it’s relative cost, determine if SpaceX will reuse the first stage ($62 MM each).
- Critical Operational Issues (COIs):

1. How many mission outcomes result in first-stage recovery?
2. How many launches resulted in successful mission outcomes?
3. Which launch sites [1] provide the highest number of recoveries and recovery rates?
4. Which payloads result in the most first-stage recoveries?
5. Which booster versions resulted in fails due to drone ship during 2015

[1] Each launch site is restricted in launch angle (retrograde or prograde). CCABF takes advantage of Earth's natural rotation by launching toward the east from Cape Canaveral. The speed at which the Earth rotates at Cape Canaveral is roughly 914 mph, helping to give rockets some extra speed to reach their destination (Moon-shots can only be achieved from CCAFB). Launches from VSFB fly southward, allowing payloads to be placed in high-inclination orbits such as polar or Sun-synchronous orbit, which allow full global coverage on a regular basis and are often used for weather, Earth observation, and reconnaissance satellites. (Space Notes, 2020)

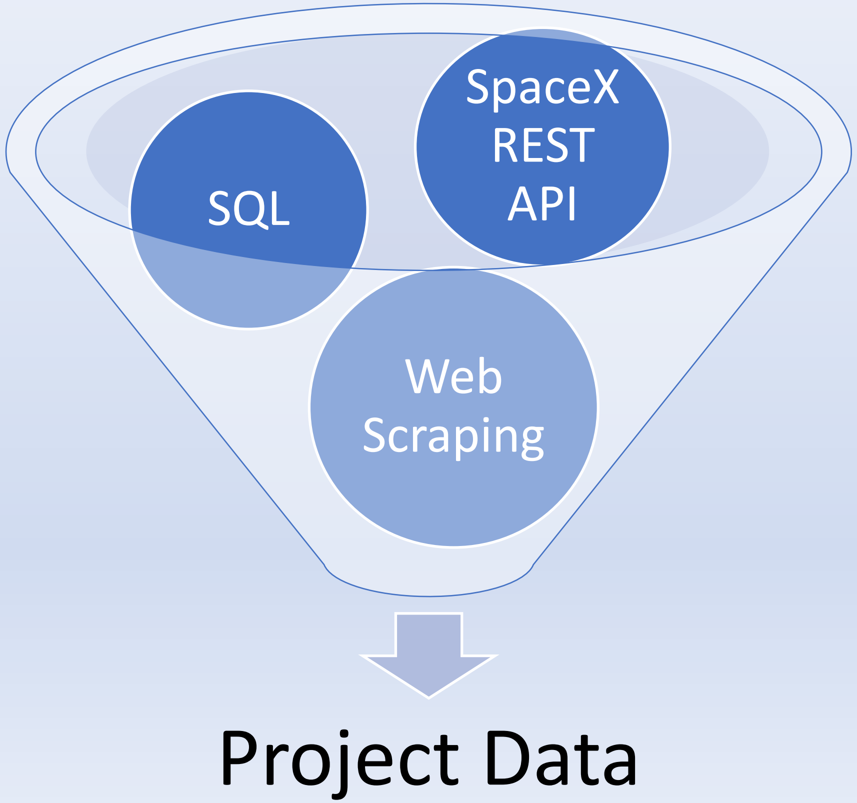
Methodology

Methodology

- Data collection methodology:
  - Previously acquired data
  - Web scraping
- Perform data wrangling
  - Data cleansing
  - Imputing missing data and Normalizing data
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Hyperparameter tuning
  - Model scoring (variety of metrics)

Data Collection

- Data using SpaceX REST API or Representational State Transfer Application Programming Interface in the IBM Cloud environment, including:
  - Launch Sites
  - Orbits
  - First-stage Recovery
  - Payloads
  - Booster Types
- Structured Query Language (SQL) magic queries in the IBM Cloud environment
- Web Scraping or Web Data Extraction from external websites such as SpaceX



Github Link https://github.com/stricje1/Data_Science_Capstone/blob/main/Data%20Collection%20API%20Lab.ipynb

Data Collection – SpaceX API

API Query | Response
- Spacex_url =https://api.spacexdata.com/v4/rockets/
- Get response using the spacex_url (a .json file)

.json Conversion
- Normalize response.json using Pandas

Data Cleaning
- Get Booster Version data
- Get Launch Site data
- Get Payload data
- Get Core data

Dictionary Assignment
- Create Launch Dictionary:
  - Flight Number
  - Date
  - Orbit
  - Etc.

```
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,
```

Construct Dataframe
- Filter Falcon 9 first stage
- Pandas dataframe

	FlightNumber	Date	BoosterVersion	PayloadMass
4	6	2010-06-04	Falcon 9	NaN
71	83	2019-08-06	Falcon 9	6500.0
69	80	2019-06-12	Falcon 9	1425.0
68	79	2019-05-24	Falcon 9	13200.0

Link/Send to Github
- https://github.com/stricje1/Data_Science_Capstone

stricje1 Merge pull request #7 from stricje1/master ...

Data	Add files via upload
Sample_Code	Add files via upload
Data Collection API Lab.ipynb	Published from Watson Studio.
EDA Lab.ipynb	Published from Watson Studio.
EDA with Visualization lab.ipynb	Published from Watson Studio.
Interactive Maps.ipynb	Rename New_maps.ipynb to Interactive Maps.ipynb
LICENSE	Initial commit

Data Collection – Web Scraping

Get Web Response

- static_url="https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"

Create Beautiful Soup Object

- soup=BeautifulSoup(launch_data)

Create a `BeautifulSoup` object from the HTML `response`

```
# Use BeautifulSoup() to create a BeautifulSoup object from a
from bs4 import BeautifulSoup

#soup = BeautifulSoup(launch_data, 'html5lib')
soup = BeautifulSoup(launch_data)
```

Print the page title to verify if the `BeautifulSoup` object was created properly

```
# Use soup.title attribute
print(soup.title)
```

Locate Tales

- html_tables=soup.find_all('table')
- first_launch_table=html_tables[2]
- print(first_launch_table)

Get Column Names

- column_names=[]
- For Loop
- Extract column Name

Append Data

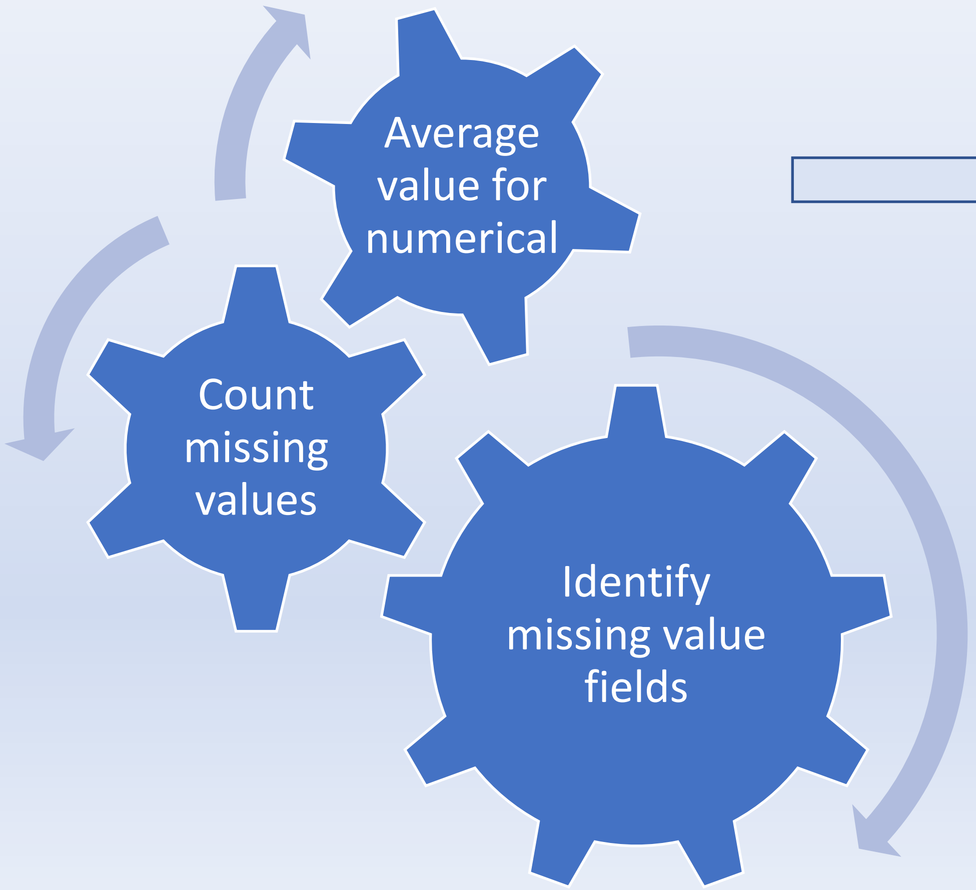
- column_names.append(col_name)

Dictionary to Dataframe

- Launch dictionary
- Enter Initial Values

Github Link:
https://github.com/stricje1/Data_Science_Capstone/blob/main/Web%20Scraping.ipynb

Data Wrangling – Missing Data



Apply average value to missing values

Before

```
In [34]: data_falcon9.isnull().sum()

Out[34]: FlightNumber      0
         Date              0
         BoosterVersion    0
         PayloadMass       5
         Orbit             0
         LaunchSite        0
         Outcome           0
```

After

```
# Calculate the mean value of PayloadMass column
data_falcon9.mean(axis = 1, skipna = True)
pm_mean = data["PayloadMass"].mean()

# Replace the np.nan values with its mean value
data_falcon9['PayloadMass'].fillna(value=pm_mean, inplace=True)
#print('Updated Dataframe:')
#print(data_falcon9)
data_falcon9.isnull().sum()
```

```
FlightNumber      0
Date              0
BoosterVersion    0
PayloadMass       0
Orbit             0
LaunchSite        0
Outcome           0
```

Github Link:
https://github.com/stricje1/Data_Science_Capstone/blob/main/Data%20Collection%20API%20Lab.ipynb

Exploratory Data Analysis (EDA)

Perform EDA

- Collect the number of Missions at each site , etc.
  - Calculate the number and occurrence of mission outcome per orbit type
  - Export data as .csv file
- Calculate the number and occurrence of each orbit, etc.
  - Create a landing outcome label from Outcome column
  - Work out success rate for every landing data type

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

```
Orbit
GTO       27
ISS       21
VLEO      14
PO         9
LEO        7
SSO        5
MEO        3
ES-L1      1
GEO        1
HEO        1
SO         1
dtype: int64
```

```
# Apply value_counts() on column LaunchSite
df.value_counts("LaunchSite")
```

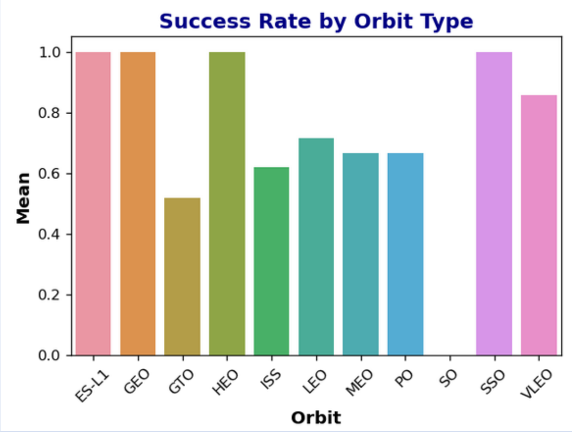
```
LaunchSite
CCAFS SLC 40    55
KSC LC 39A      22
VAFB SLC 4E     13
dtype: int64
```

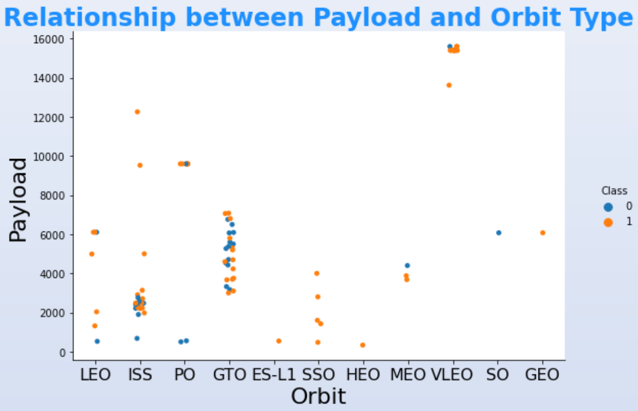
Flight number vs Orbit type
Payload vs Orbit Type
Success based on Orbit Type
Annual Success Trend
Payload vs Launch Site
Payload vs Booster Type

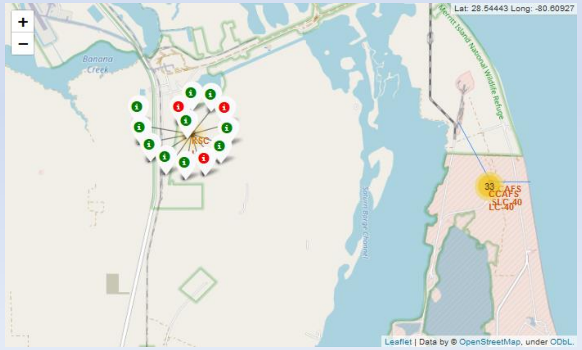
EDA with Data Visualization

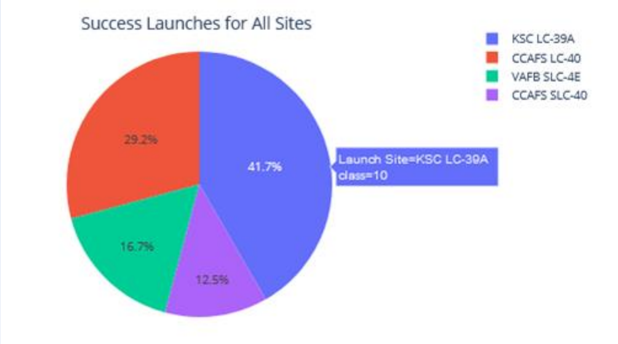
Plots

- Scatterplot Flight Number and Launch Site
- Scatterplot Payload and Launch Site
- Bar-plot success rate of each Orbit type
- Scatterplot of Flight Number and Orbit type
- Scatterplot of Payload and Orbit type with Sliders
- Time series plot of launch success yearly trend









Github Link: https://github.com/stricje1/Data_Science_Capstone/blob/main/EDA%20with%20Visualization%20lab.ipynb

EDA with SQL

- Unique launch site search
- Five records where 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.
- List the total number of successful and failure mission outcomes
- Names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

```
%sql SELECT payload,
    landing_outcome,
    payload_mass_kg
FROM SPACEXDATASET
WHERE landing_outcome LIKE 'Success (drone ship)'
AND payload_mass_kg BETWEEN 4000 AND 6000;
```

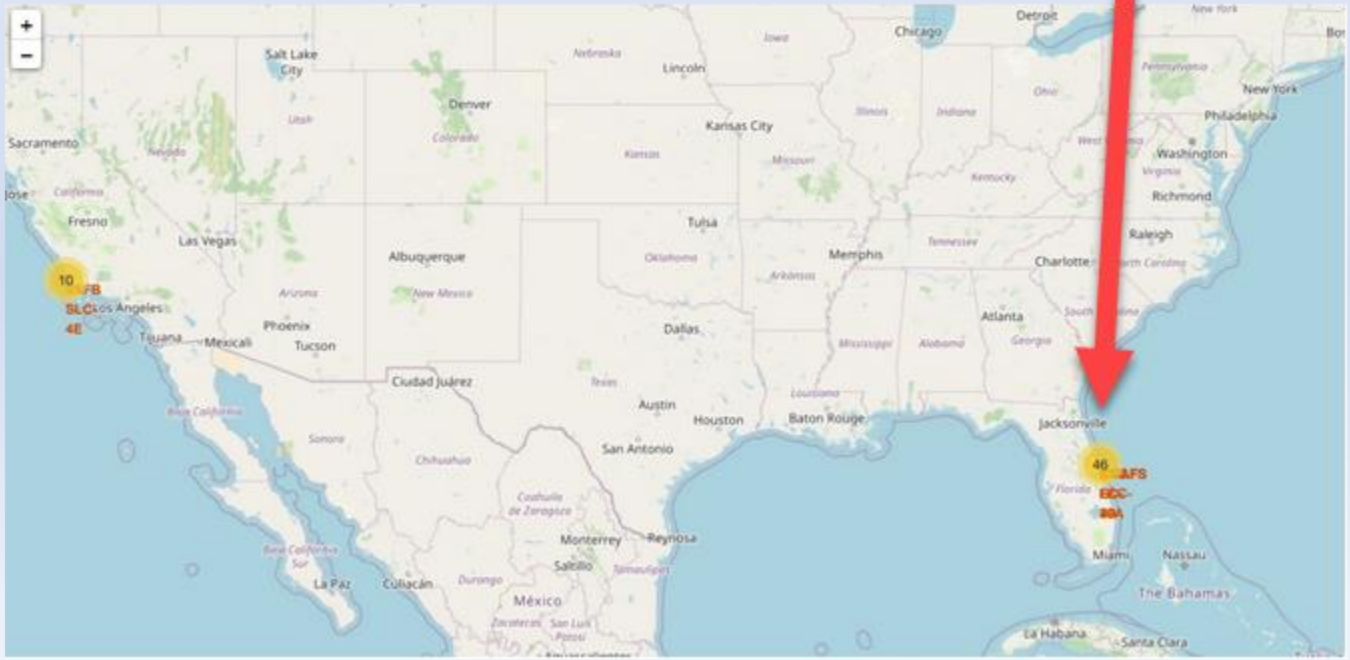
payload	landing_outcome	payload_mass_kg
JCSAT-14	Success (drone ship)	4696
JCSAT-16	Success (drone ship)	4600
SES-10	Success (drone ship)	5300
SES-11 / EchoStar 105	Success (drone ship)	5200

DATE	landing_outcome	booster_version	launch_site
2015-01-10	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
2015-04-14	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

Build an Interactive Map with Folium

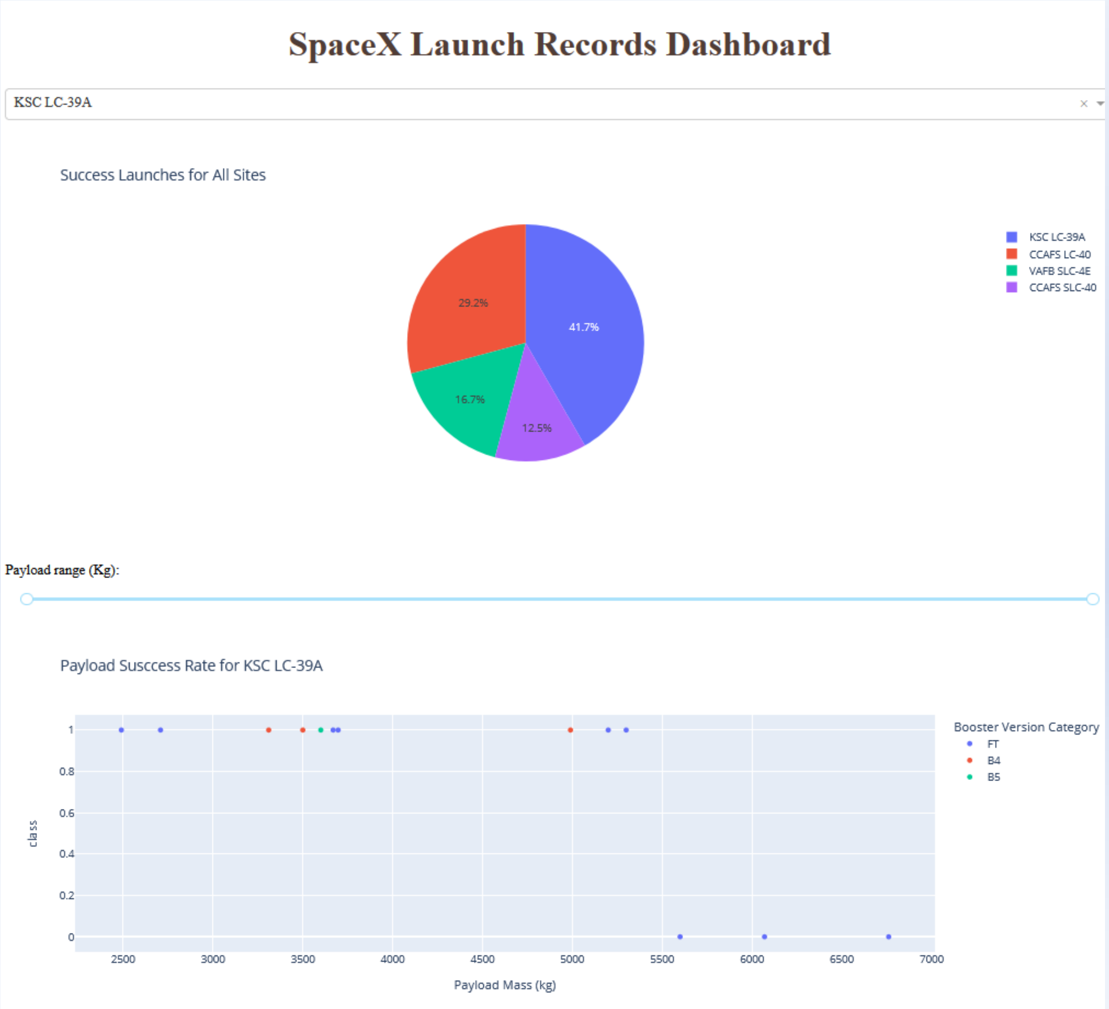
- This map of the United States contains the east and west coast launch sites with city markers, cluster markers, circle markers, and popups.
- We added these markers to the map object to drill-down for addition insights for any given launch site

- Later, we add line-of-site segments to show the distance between launch sites and other objects like coast lines, railroads, and nearby cities.
- Here, we can see that the cluster of 46 becomes two clusters of 26 and 7. (Strickland, J., 2019 B, pp. 58-60)



Github Link: https://github.com/stricje1/Data_Science_Capstone/blob/main/Interactive_maps.ipynb

Build a Dashboard with Plotly Dash



- The dashboard's pie chart shows the percentage of successful launches for all sites
- Interactive pie chart of overall success based on launch site
- Scatter plots at the bottom can be manipulated by the site selector at the top of the dashboard (a pulldown menu)
- Helps count successful and failed launched for different ranges in payload mass

Github Link: https://github.com/stricje1/Data_Science_Capstone/blob/main/spacex_dash_app.ipynb

Predictive Analysis (Classification)

Logistic Regression Classifier
- Best LR parameters
- GridSearch Model

Support Vector Machine Classifier
- Best SVM parameters
- GridSearch Model

Tree Classification Model
- Best Tree parameters
- GridSearch Model

K-Nearest Neighbor Classifier
- Best KNN parameters
- GridSearch Model

Model Type

Data selection and prep (train Test split)

Model Definition

Training (fitting) the model with standardized training data

Tuning **hyper-parameters** via Grid Search

Making model predictions

Scoring the model with the Test data

Github Link: https://github.com/stricje1/Data_Science_Capstone/blob/main/Predictive%20Model.ipynb

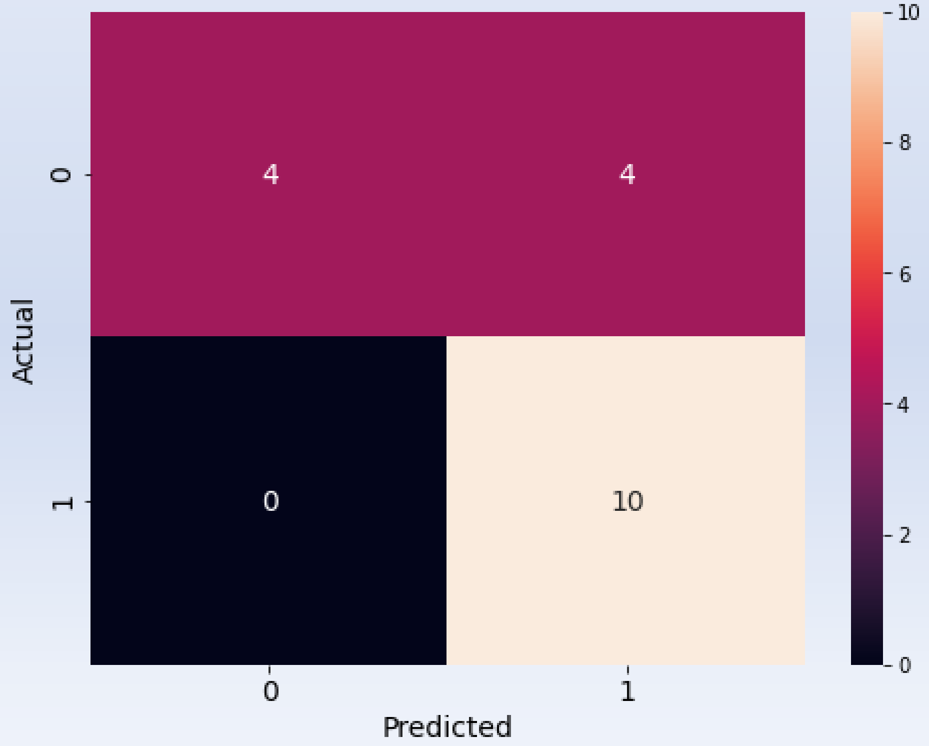
Best Model: K Nearest Neighbor Classifier

K-Nearest Neighbor Classifier

- Parameters: algorithm': 'auto', 'n_neighbors': 1, 'p': 1
- GS Accuracy: 0.7778

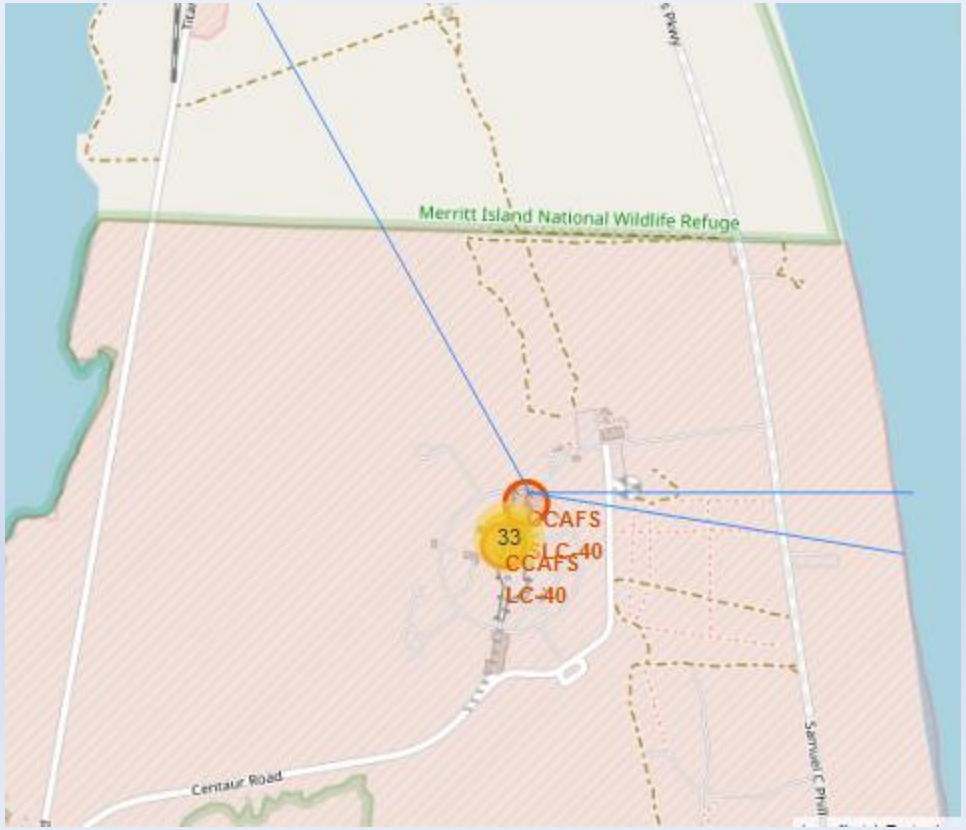
	precision	recall	f1-score	support
0	1.00	0.50	0.67	8
1	0.71	1.00	0.83	10
accuracy			0.78	18
macro avg	0.86	0.75	0.75	18
weighted avg	0.84	0.78	0.76	18

Success Summary
18 cases
10 True Positive
4 False Positive



Results

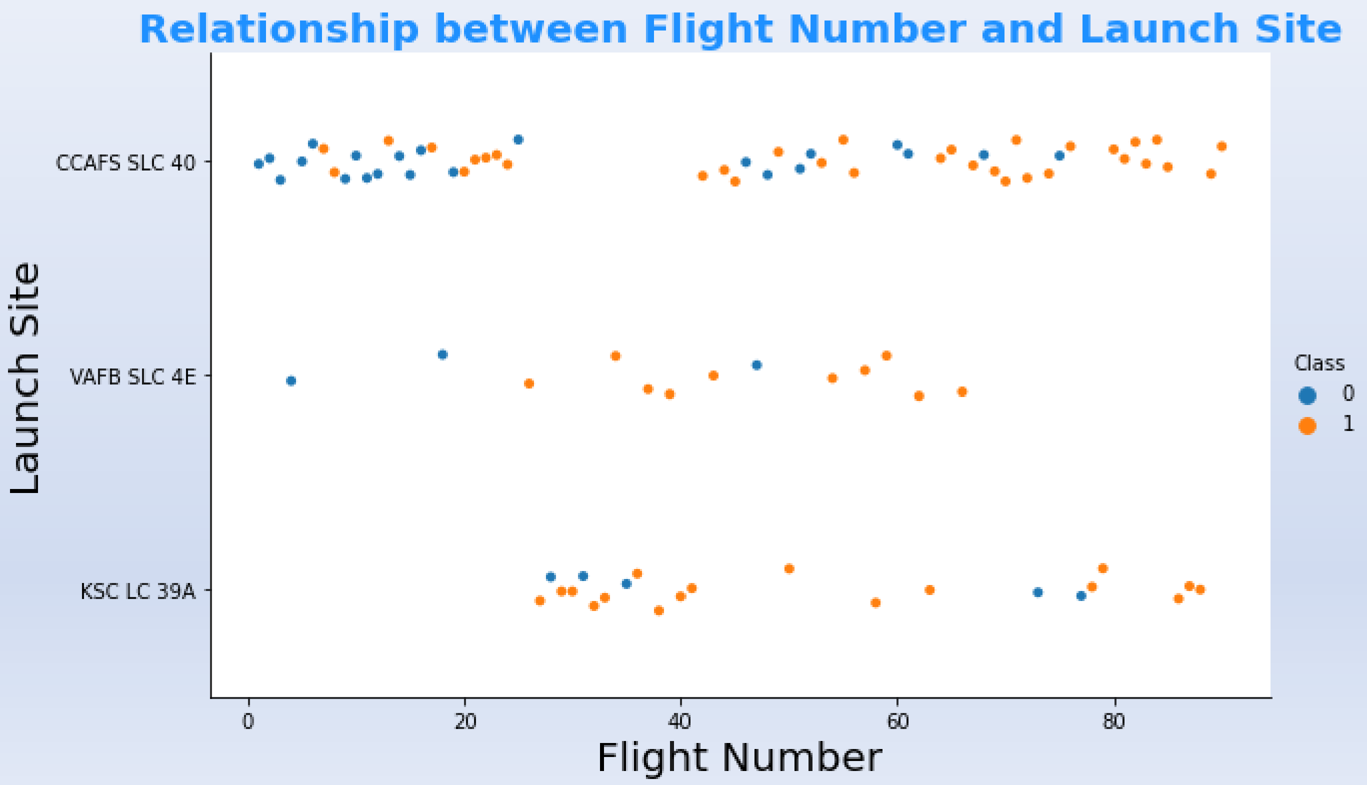
- Exploratory data analysis results: Launches can be characterized by launch site, payload, booster type
- Mission success characterized by launch site, payload, orbit type
- Interactive analytics using pie charts and maps for proximity analysis (roads, railroads, highway, coastlines, and cities)
- Predictive analysis results: Success can be classified & predicted using machine learning models



EDA Insights

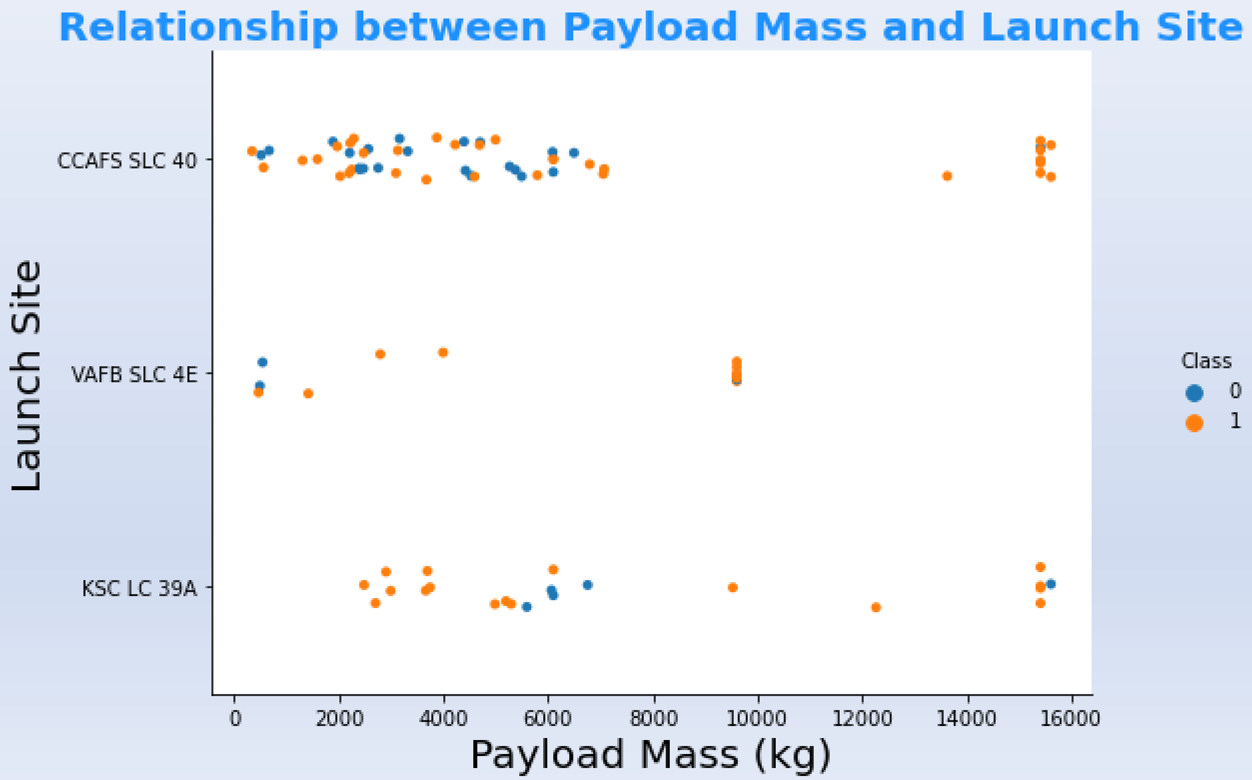
Flight Number vs. Launch Site

- Scatter plot of Flight Number vs. Launch Site
    - Orange = Success
    - Blue = Failure
- Insights:
    - More flights at CCAFS SLC 40
    - Later flights more successful at each site



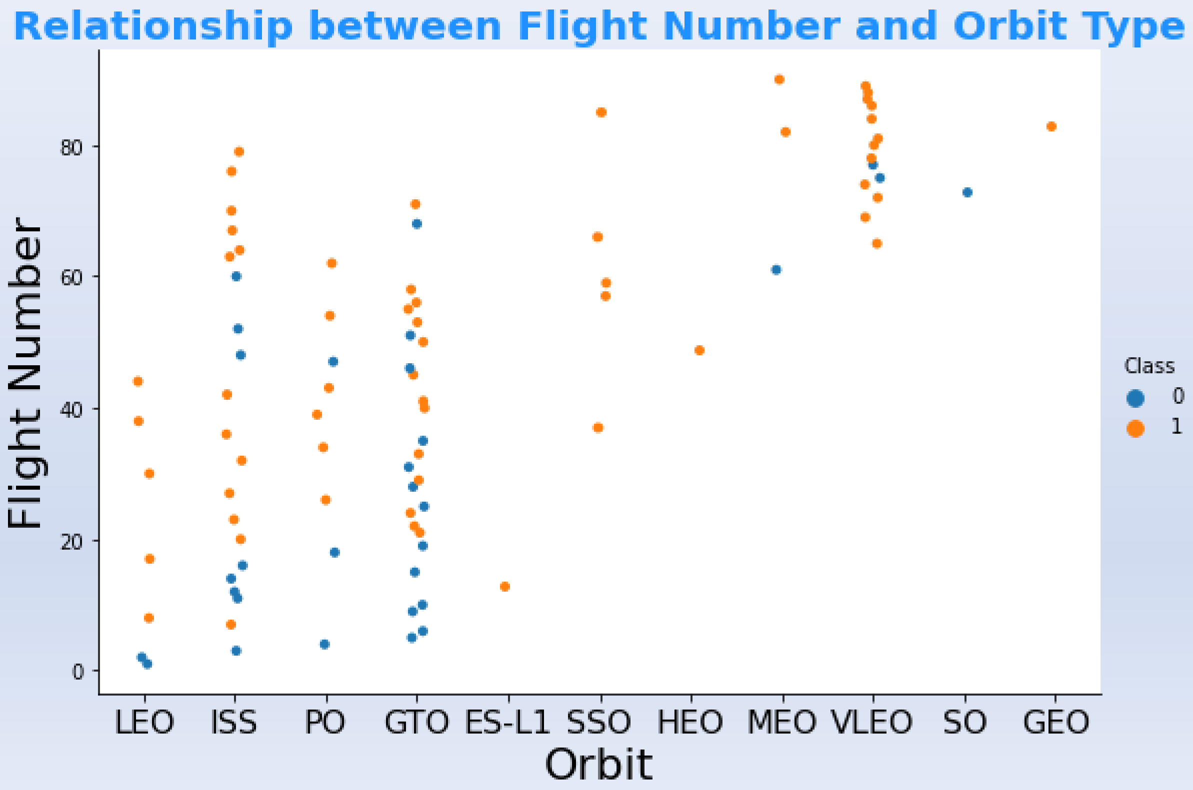
Payload vs. Launch Site

- Scatter plot of Payload vs. Launch Site
  - Orange = Success
  - Blue = Failure
- Insights:
  - More success for higher payloads (>8,000 kg)
  - Most lower payloads are launched ay CCAFS SLC 40
  - Nearly all payloads launched from VAFB (VSFB) SLC 4E are successful



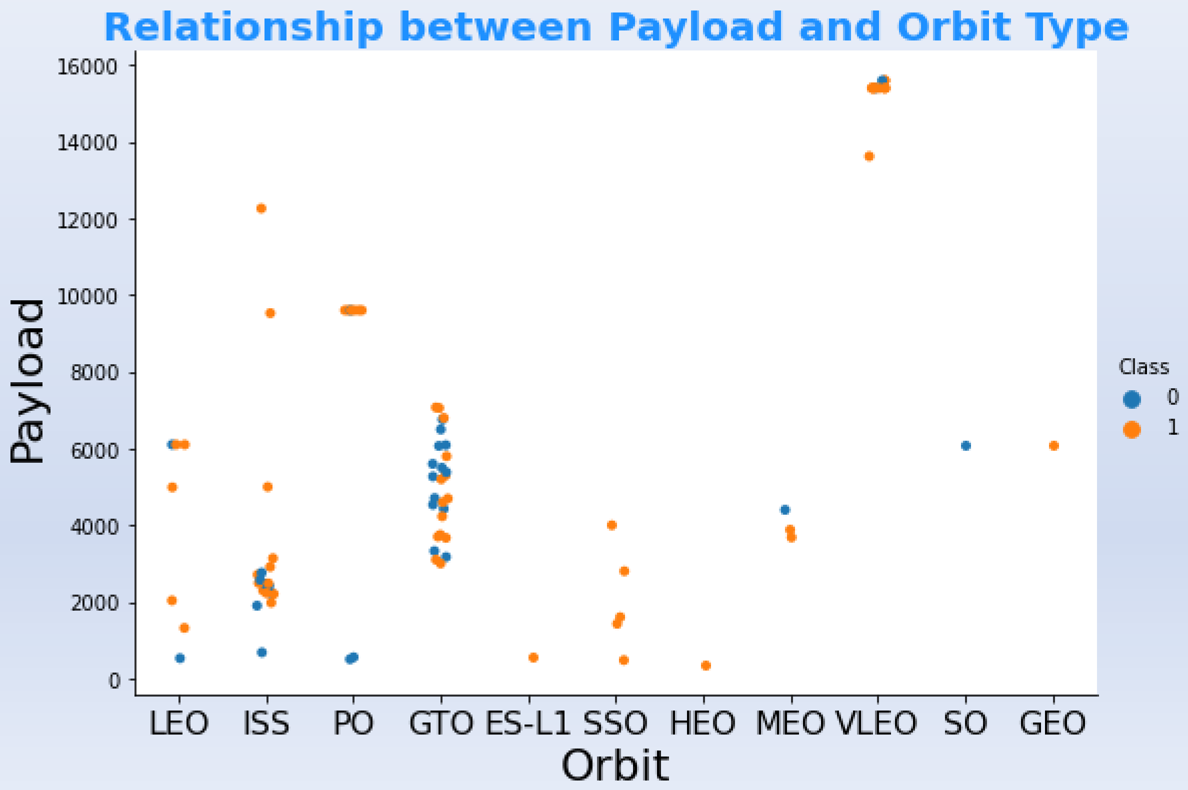
Flight Number vs. Orbit Type

- Scatter plot of Flight number vs. Orbit type:
  - Orange = Success
  - Blue = Failure
- Insights:
  - Most flights occur at lower orbits (LEO, ISS, PO)
  - Flight at low orbits VELO and SSO are very successful
  - No clear pattern for higher orbits, except the earlier flights were seldom successful (i.e., GTO)



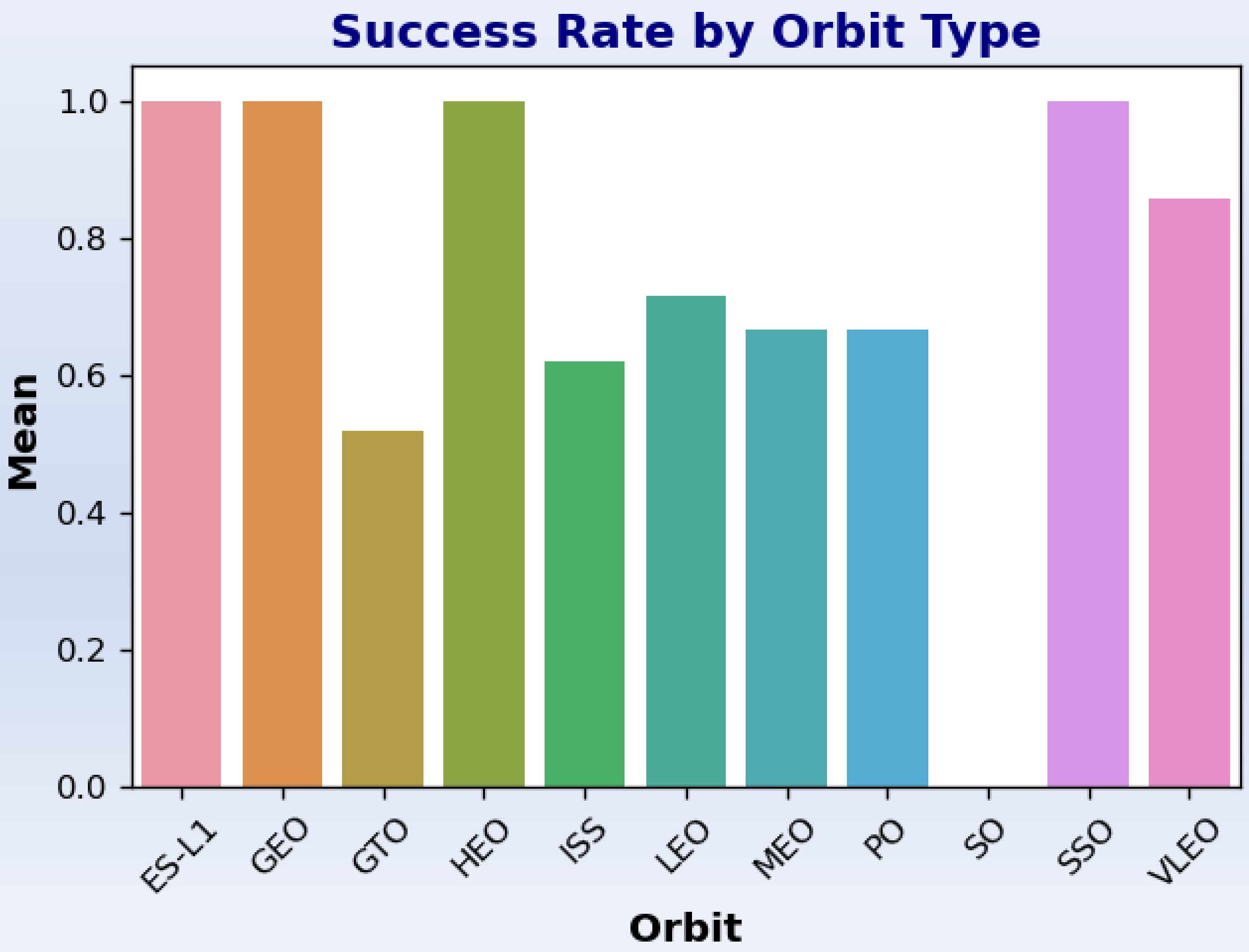
Payload vs. Orbit Type

- Scatter plot of payload vs. orbit type
    - Orange = Success
    - Blue = Failure
- Insights:
    - Most unsuccessful flights occur at GTO (a high orbit)
    - Most unsuccessful launches occur at lower payloads
    - High payloads at lower orbits were more successful (i.e. VLEO & ISS)



Success Rate vs. Orbit Type

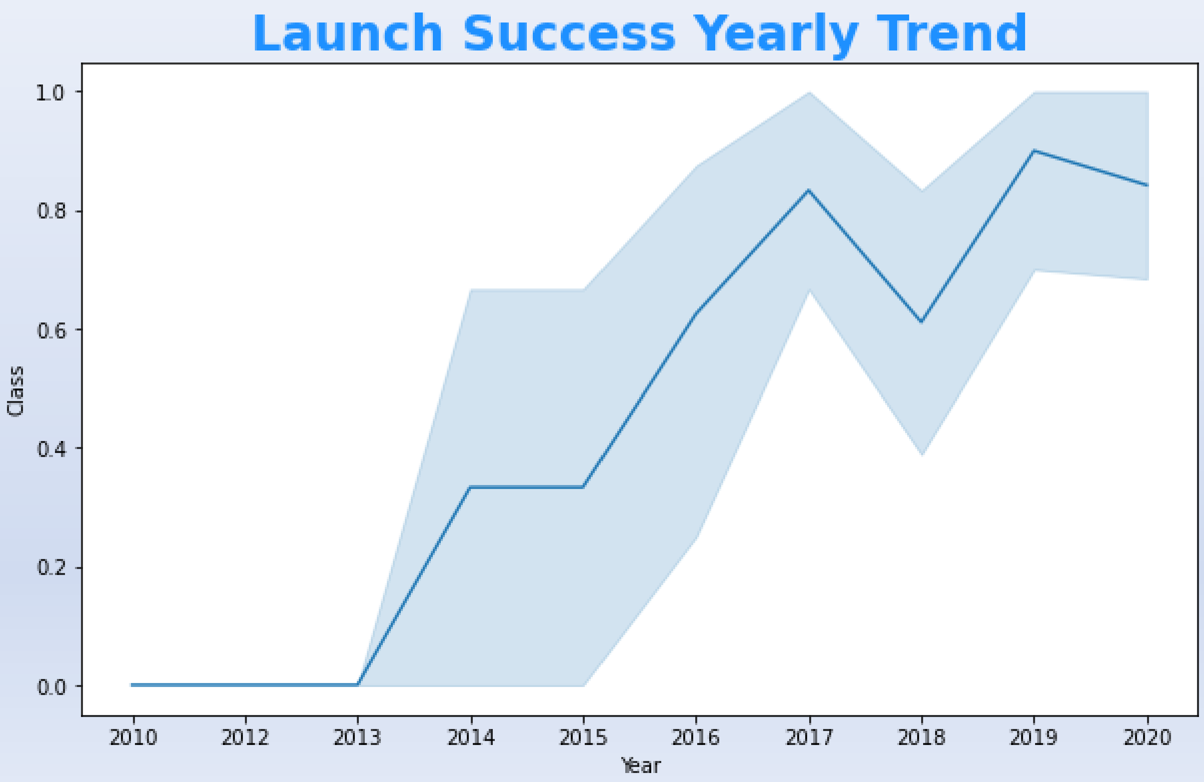
- The orbits with the highest average success rates are: ES-L1, GEO, HEO, and SSO
- Higher orbits (GEO & HEO) and lower orbits (ES-L1, SSO & VLEO have a higher success rate



Launch Success Yearly Trend

- Line chart of yearly average success rate

- Insights:
    - More successes occurred in later years
    - A negative times series trend (slope) occurred in 2018 (Strickland, J. 2020 B)



Visual EDA Success Results Summary

Success	Category
Lower orbits	LEO, SSO, VLOE
Higher orbits	GEO, HEO
Higher payloads	14K km- 16K km
Lower payloads	1K km - 4K km
Later Flights	Flight No. > 60
Lunch Sites	KSC LC 39A, VAFB SLC4E
Later Dates	2017 and 2019

All Launch Site Names

- Names of the unique (4) launch sites
- Using the SQL Keyword “UNIQUE”

Task 1

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

```
%sql SELECT UNIQUE "LAUNCH_SITE" from SPACEXDATASET;
```

```
 * ibm_db_sa://pbw80164:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.clou
d:31321/bludb
Done.
```

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

Launch Site Names Begin with 'CCA'

- Five records where launch sites begin with `CCA` (CCAFS LC-40)
- Using SQL keywords/conditions “WHERE”, “LIKE”, and “LIMIT”

Task 2

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

```
%sql SELECT LAUNCH_SITE FROM SPACEXDATASET WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5;
```

```
 * ibm_db_sa://pbw80164:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.clou
d:31321/bludb
Done.
```

launch_site
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40

Total Payload Mass

- Total payload carried by boosters from NASA (619,967 kg)
- Using SQL aggregate function “SUM” is 619,967 kg

Task 3

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

```
%sql SELECT SUM(payload_mass_kg) FROM SPACEXDATASET;
```

```
 * ibm_db_sa://pbw80164:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.clou
d:31321/bludb
Done.
```

1
619967

Average Payload Mass by F9 v1.1

- Calculated average payload mass carried by booster version F9 v1.1
- USING SQL aggregate function "AVG" (average) and keywords "WHERE" and "LIKE" is 2,534 kg

Task 4

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

```
%sql SELECT avg(payload_mass_kg) FROM SPACEXDATASET WHERE booster_version LIKE 'F9 v1.1%';
```

```
 * ibm_db_sa://pbw80164:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.clou
d:31321/bludb
Done.
```

1
2534

First Successful Ground Landing Date

- Dates of the first successful landing outcome on ground pad
- Using SQL aggregate function “MIN” and keywords “LIKE” and “WHERE” is 22 Dec 2015

Task 5

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

Hint:Use min function

```
%sql SELECT MIN(date) FROM SPACEXDATASET WHERE landing_outcome LIKE 'Success%';
```

```
 * ibm_db_sa://pbw80164:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.clou
d:31321/bludb
Done.
```

1
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- Names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000
- Using SQL keywords " "WHERE", "LIKE", "AND", and "BETWEEN" as seen in the output below

Task 6

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, landing_outcome, payload_mass_kg FROM SPACEXDATASET WHERE landing_outcome LIKE 'Success (dr
one ship)' AND payload_mass_kg BETWEEN 4000 AND 6000;
```

```
 * ibm_db_sa://pbw80164:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.clou
d:31321/bludb
Done.
```

payload	landing_outcome	payload_mass_kg
JCSAT-14	Success (drone ship)	4696
JCSAT-16	Success (drone ship)	4600
SES-10	Success (drone ship)	5300
SES-11 / EchoStar 105	Success (drone ship)	5200

Total Number of Successful and Failure Mission Outcomes

- Total number of successful and failure mission outcomes
- Using SQL keywords “WHERE” and “LIKE” with intermediate clauses “GROUP BY” and “ORDER BY” yielding the output below

Task 7

List the total number of successful and failure mission outcomes

In [20]:
```
%%sql
SELECT count(Mission_Outcome) as "Mission Outcome"
from SPACEXDATASET where Mission_Outcome LIKE '%Success%'
GROUP BY Mission_Outcome
ORDER BY Mission_Outcome DESC;
```

```
 * ibm_db_sa://pbw80164:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:31321/bludb
Done.
```

Out[20]:

Mission Outcome
1
99

Boosters Carried Maximum Payload

- Names of boosters which have carried the maximum payload mass: 12 total
- Using SQL keywords “DISTINCT” and “AS” with aggregate function “MAX” and intermediate clauses “GROUP BY” and “ORDER BY”

Task 8

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

```
%%sql
SELECT DISTINCT Booster_Version, MAX(PAYLOAD_MASS_KG) AS Maximum_Payload_Mass
FROM SPACEXDATASET GROUP BY Booster_Version ORDER BY Maximum_Payload_Mass DESC;
```

booster_version	maximum_payload_mass
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

- Failed `landing_outcomes` in drone ship, their booster versions, and launch site names for in year 2015
- Using SQL keywords “AND” and “BETWEEN” with multiple variables as seen in the table below

Task 9

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 landing_outcome LIKE N'%Failure%' AND
date BETWEEN '2015-01-01' AND '2015-12-31';
```

```
 * ibm_db_sa://pbw80164:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.clou
d:31321/bludb
Done.
```

DATE	landing_outcome	booster_version	launch_site
2015-01-10	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
2015-04-14	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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

- Ranked-count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order

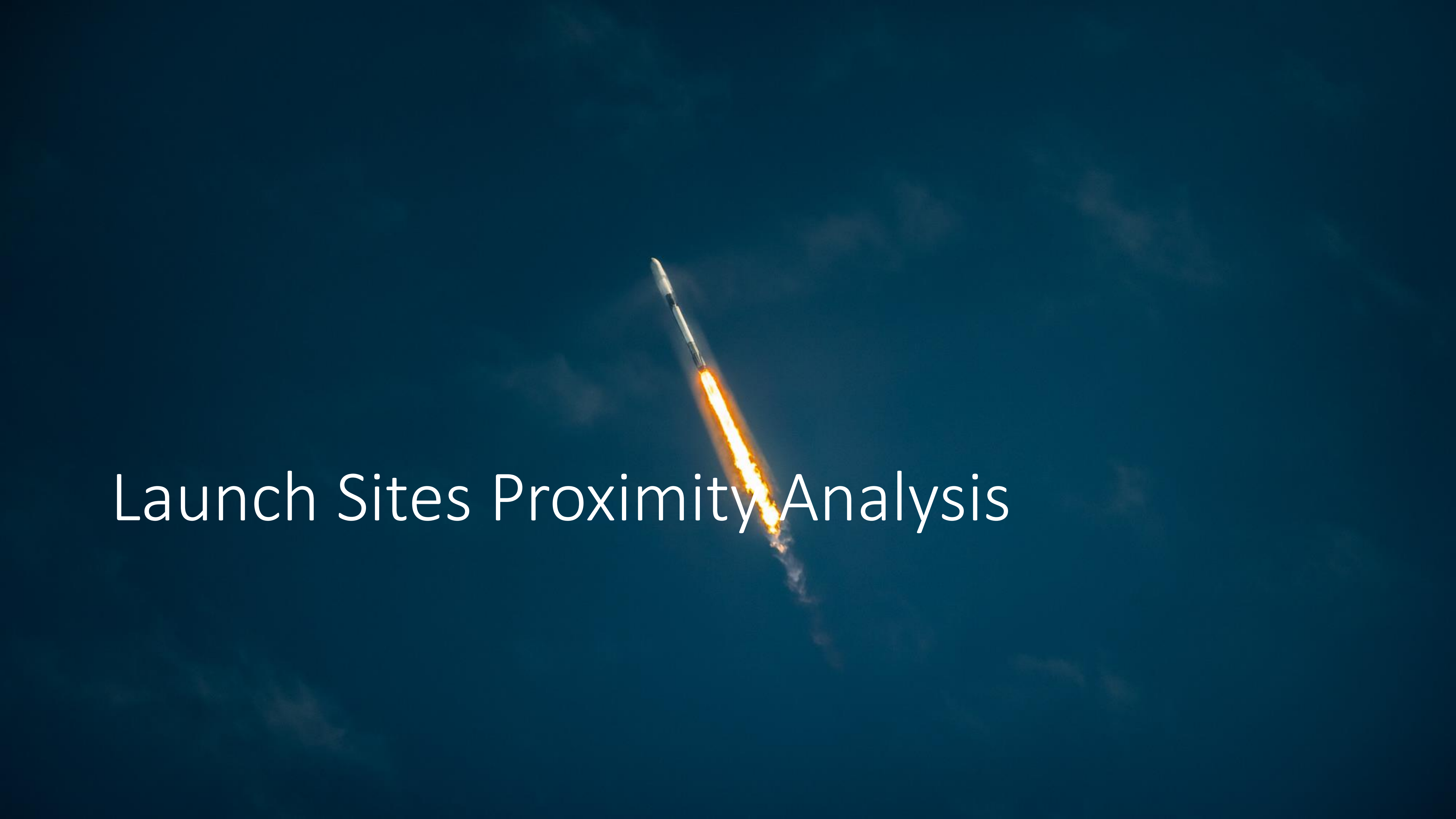
- Using SQL keywords “AS”, “WHERE”, “BETWEEN”, and intermediate clauses “GROUP BY” and “ORDER BY” as seen in the table to the right

landing_outcome	ranked landing outcomes
Uncontrolled (ocean)	2
Success (ground pad)	3
Success (drone ship)	5
Precluded (drone ship)	1
No attempt	10
Failure (parachute)	2
Failure (drone ship)	5
Controlled (ocean)	3

Task 10

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, COUNT(landing_outcome) AS "ranked landing outcomes"
FROM SPACEXDATASET
WHERE date BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY landing_outcome
ORDER BY landing_outcome DESC;
```

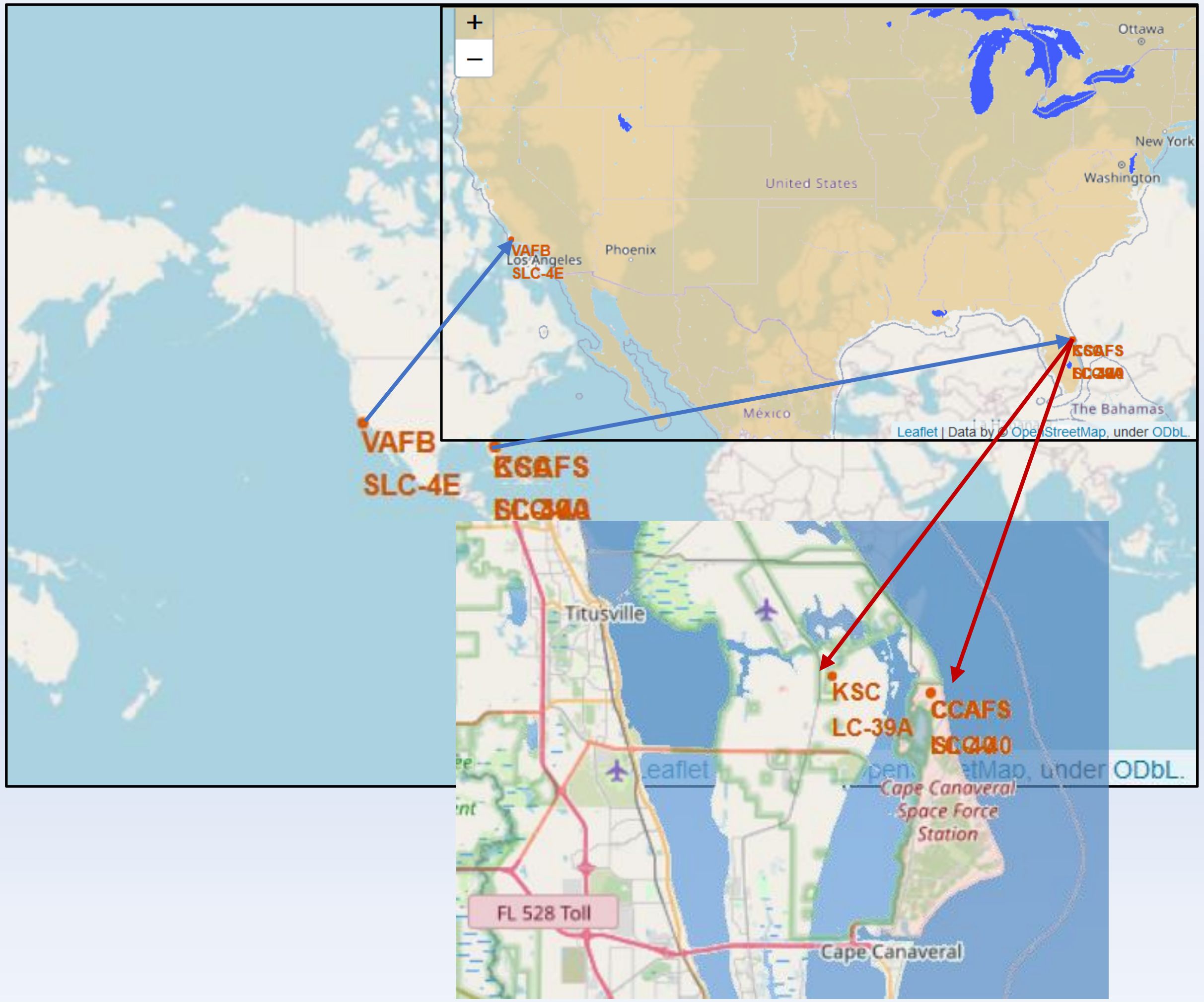

Launch Sites Proximity Analysis

Base US Map the East and West Coast Launch Sites

- The global map shows the US map with one west coast launch site and two east coast launch site (requires zooming in)

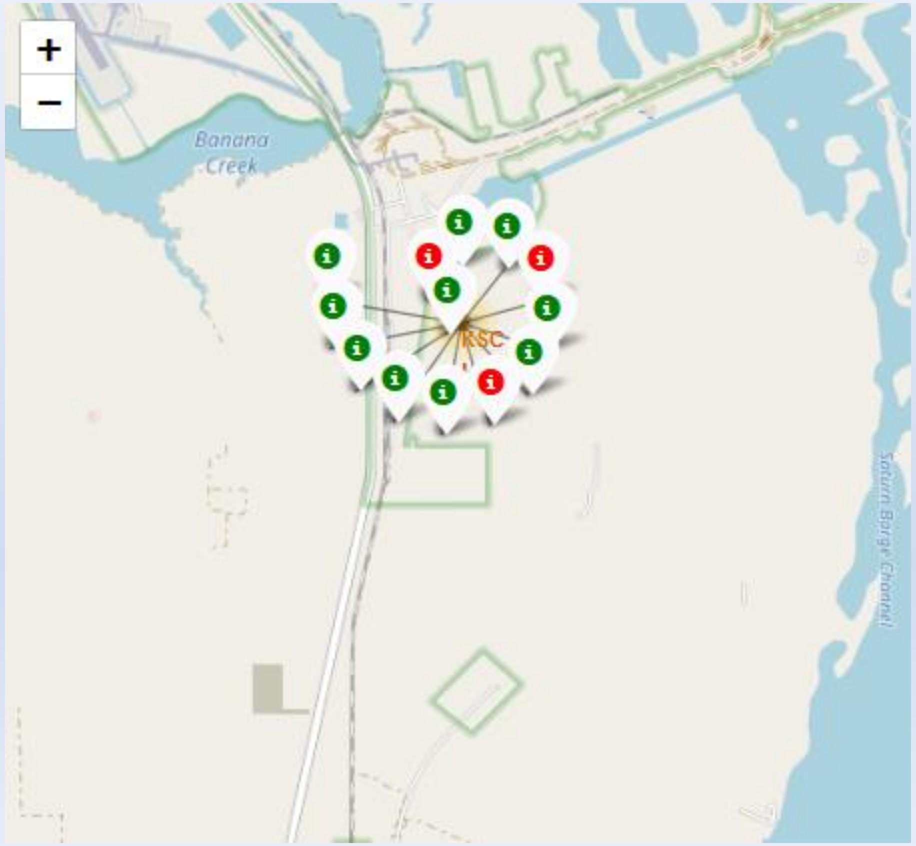
- Zooming in on the US only (or North America) still shows two sites

- Zooming in on Florida shows the two distinct east coast launch sites.



Kennedy Spaceflight Center

- The map showing Kennedy Spaceflight Center only reveals 13 separate launches
- The red markers show 3 Class 0 launches (unsuccessful stage-one recoveries)
- The green markers show 10 Class 1 launches (successful stage-one recoveries)

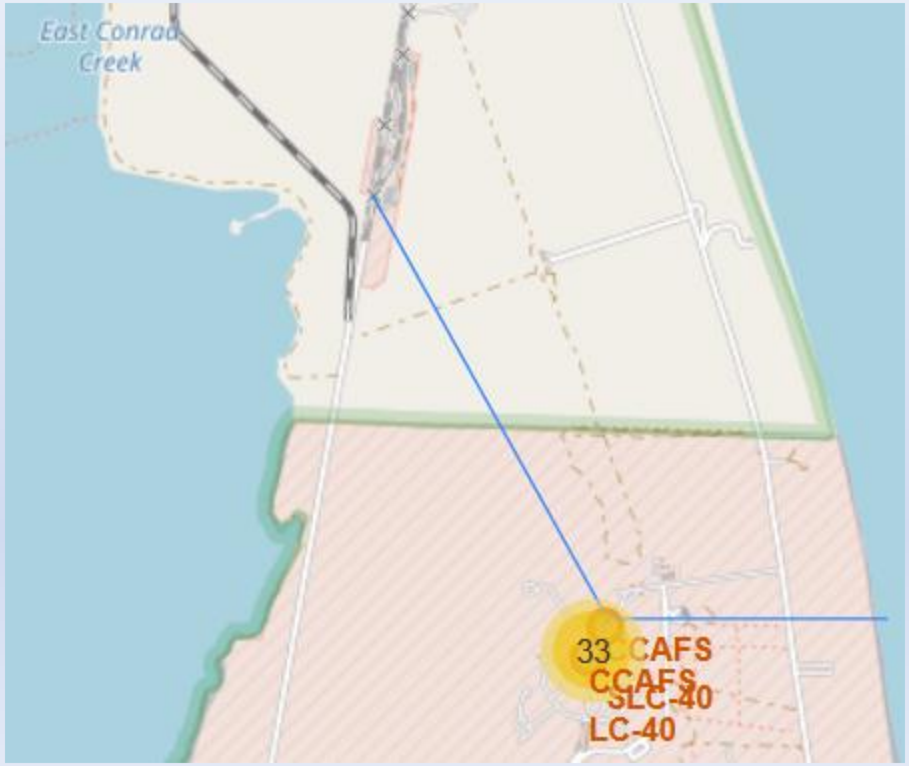


Cape Canaveral Launches

- Map showing Cape Canaveral Space Launch Complex 40 located on Cape Canaveral Air Force Station
- The complex has multiple launch pads, for example, the two "x-marks" at the "green 7" inside the red circle.
- Other pads are located at the "yellow 26"
- Blue lines show the line-of-sight (LOS) vectors from LC 40 and the selected map features: east coastline and launch pad LC 39B
- Rail lines transport launch vehicles to the pads

The Falcon 9 Launch Pad at LC 40

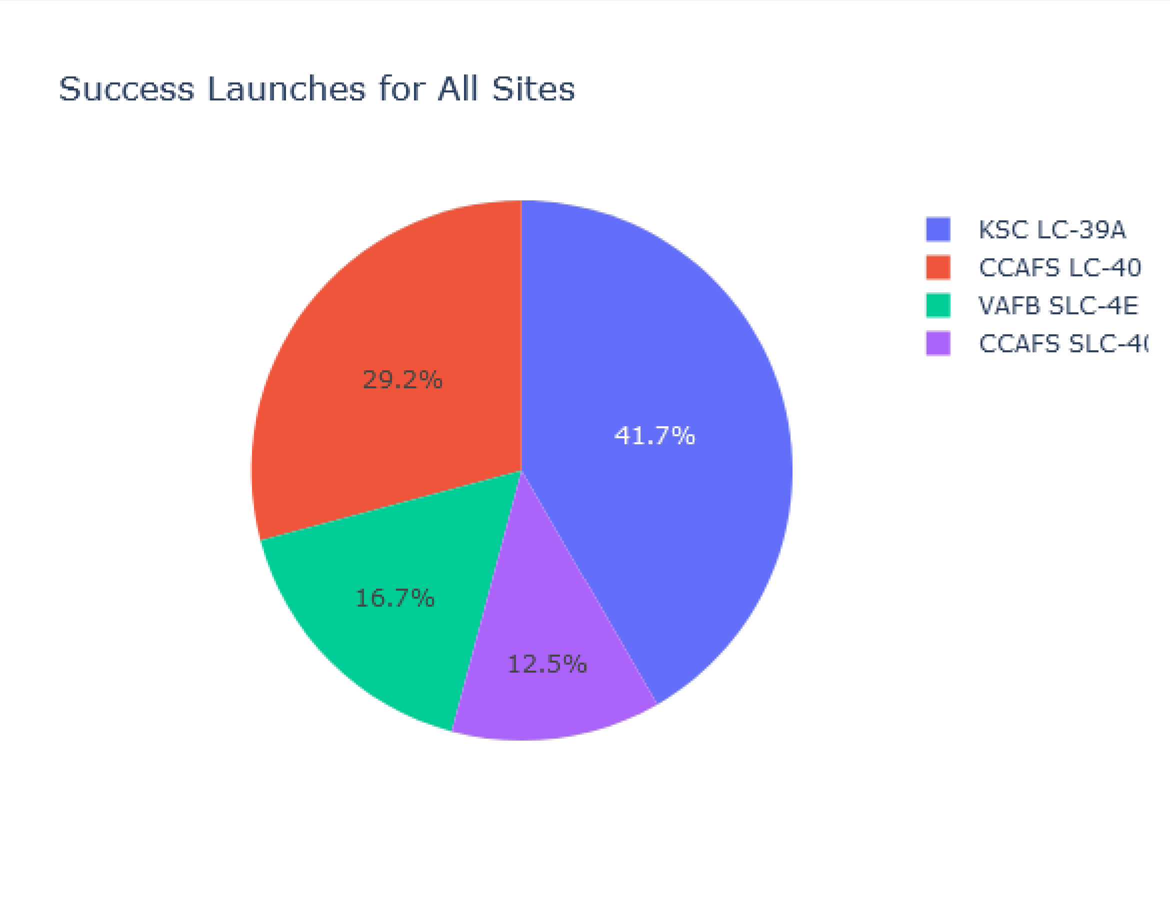
- Launch pads have to be customized for each type of launch vehicle
- CCAFS has 30 launch pads (LCs)
- LC 40 launches Falcon 9
- LC 39B launched the NASA Ares I Crew Launch Vehicle (a program I worked on in 2007)
- The diagonal blue line running from SE to NW is the LOS between LC 40 and LC 39B (0.89 miles)



Dashboard with Plotly Dash

Success Launches for all Sites

- The pie chart shows successful launch rates for four distinct launch sights, as depicted
- For example, 41.7% of successful launches were from KSC LC 39A, the Falcon Heavy launch pad
- Explain the important elements and findings on the screenshot



All Launch Sites: Payload vs.

- Scatter plot of Class for All Launch Sites – Payload (kg) vs. Booster Version category

- Booster v 1.0 & v 1.1 had very few successes.

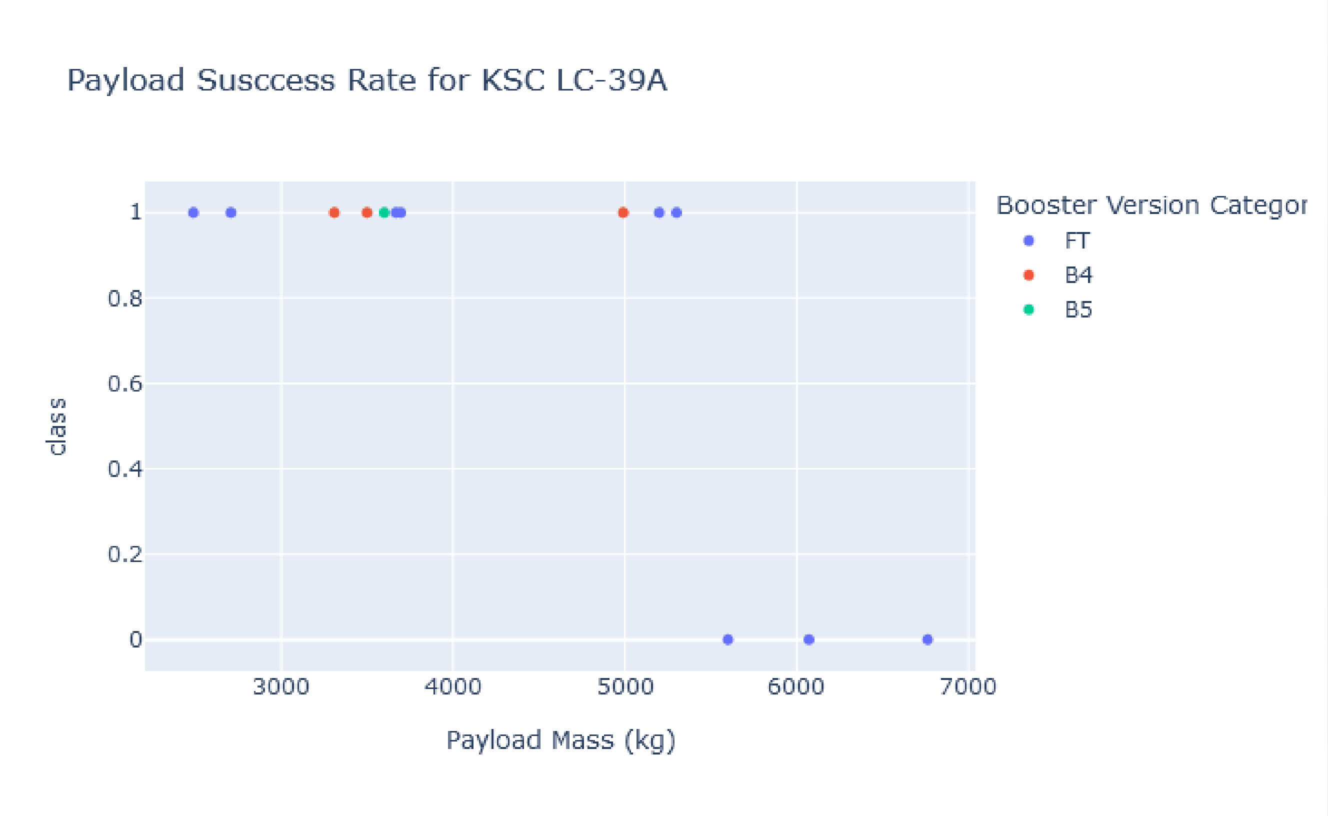
- Booster v 1.0 was launched with low payloads only

- Booster FT & B4 faired much better

- Booster B5 has only one flight

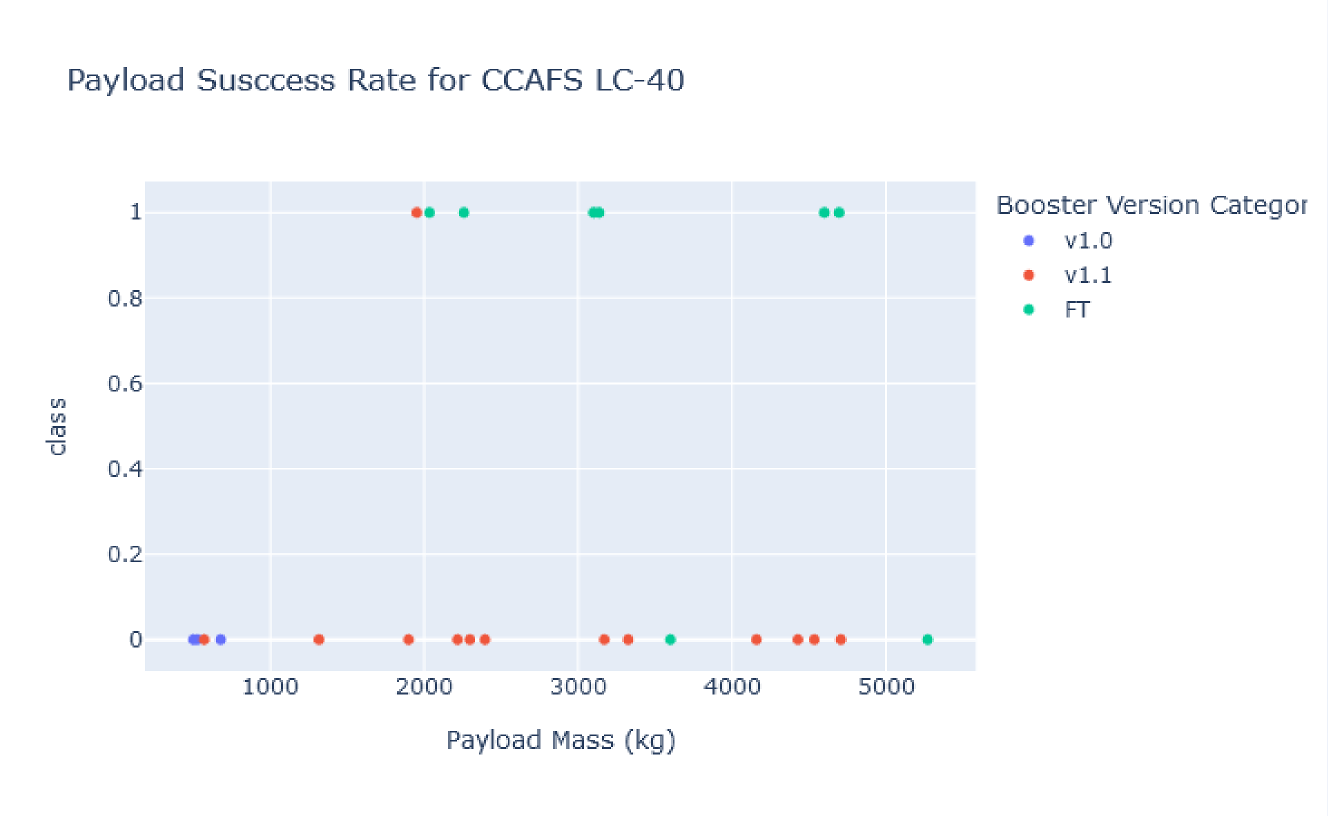
KSC LC-39A Payload vs. Launch Outcome scatter plot

- Scatter plot of Class for KSC LC-39A – Payload (kg) vs. Booster Version category
- Booster FT performed well with lower payloads (< 5,599 kg)
- Boosters B4 & B5 performed well but all flights were with lower payloads



CCAFS SLC-40 [2] Payload vs. Launch Outcome scatter plot

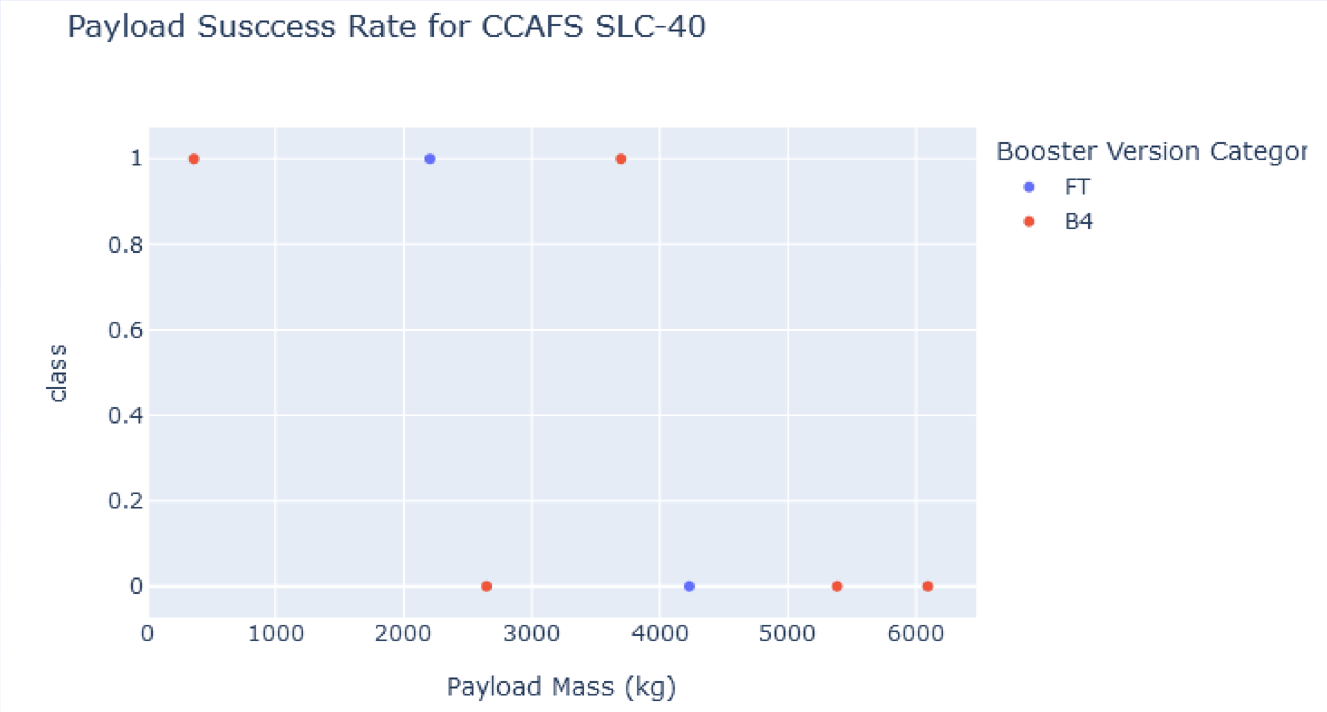
- Scatter plot of Class for CCAFS SLC 40 – Payload (kg) vs. Booster Version category
- As seen for all sites, Boosters v1.0 and v1.1 did not perform well
- Booster FT performed much better with only 2 failures
- All launches were a lower payloads (< 5,500)



[2] In Sep 2016 LC-40 was heavily damaged during a static fire test. When it was reopened in Dec 2017, it was renamed Space Launch Complex 40 (SLC-40)

CCAFS LC-40 [3] Payload vs. Launch Outcome scatter plot

- Scatter plot of Class for CCAFS LC 40 – Payload (kg) vs. Booster Version category
- The data is inconclusive, with very few launches (all launches occurred before Sep 2016)



[3] In 2007, 45th Space Wing leased Launch Complex 40 (LC 40) to SpaceX for launching Falcon 9 vehicles.

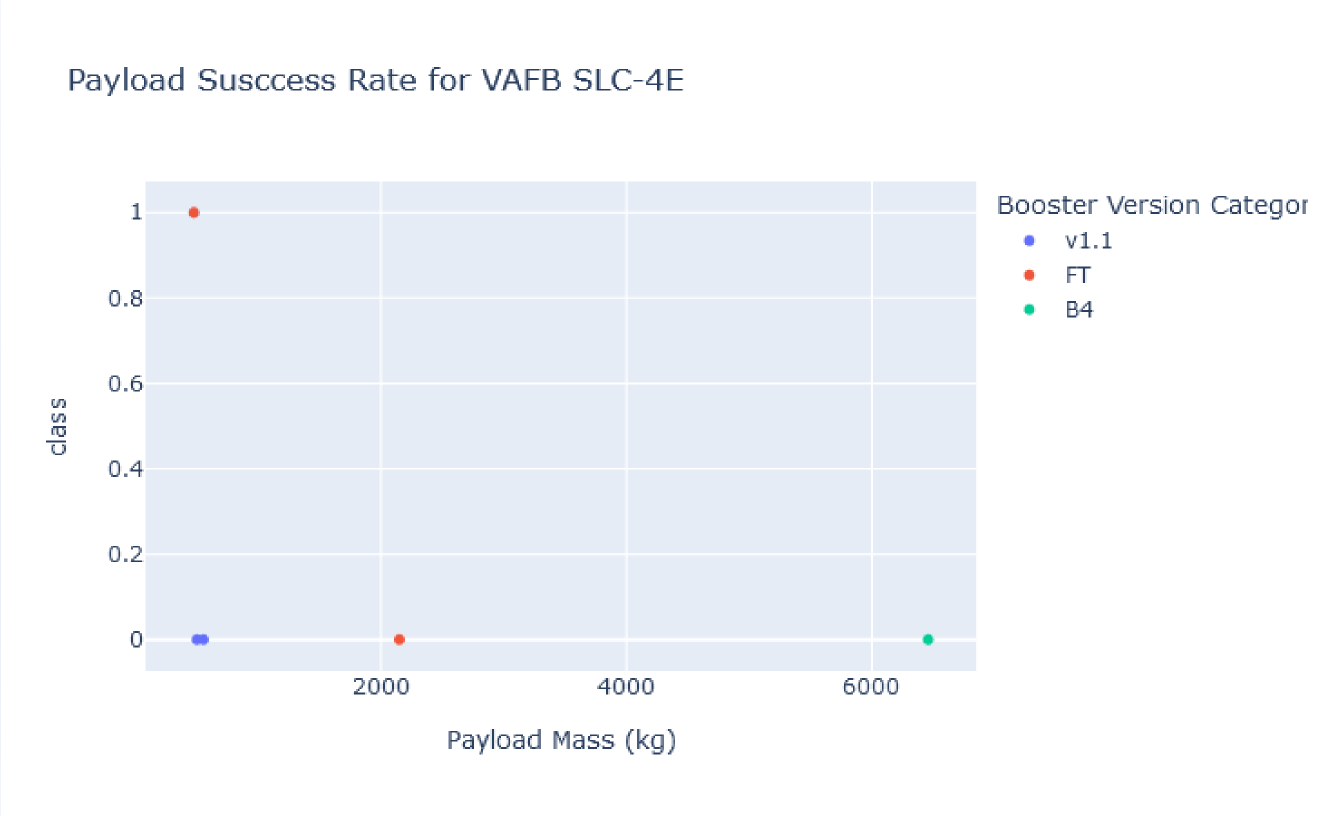
Wade, M., (2016)

VAFB SLC-4E Payload vs. Launch Outcome scatter plot

- Scatter plot of Class for VSFB SLC 4E – Payload (kg) vs. Booster Version category

- The data is inconclusive, with very launches

- However, v1.1 again showed a very high failure rate



Predictive Analysis (Classification)

Model Selection and Hyperparameter Tuning

- Support Vector Machine Classifier
- Logistic Regression Classifier
- K-Nearest Neighbor Classifier
- Decision Tree Classifier

- Grid Search for Best Parameters (Strickland, J. 2019 A, p. 446)
  - Input data: contain features important to the problem
  - Model parameters: the machine learning algorithms use to adjust data
  - Hyperparameters: variables that govern the model training process

Model Definition

```
knn_parameters = {'n_neighbors': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10],
                  'algorithm': ['auto', 'ball_tree', 'kd_tree', 'brute'],
                  'p': [1,2]}
KNN = KNeighborsClassifier()
```

Grid Search

```
knn_cv = GridSearchCV(KNN, knn_parameters)
knn_cv.fit(X_train,Y_train)
```

```
GridSearchCV(estimator=KNeighborsClassifier(),
             param_grid={'algorithm': ['auto', 'ball_tree', 'kd_tree', 'brute'],
                         'n_neighbors': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10],
                         'p': [1, 2]})
```

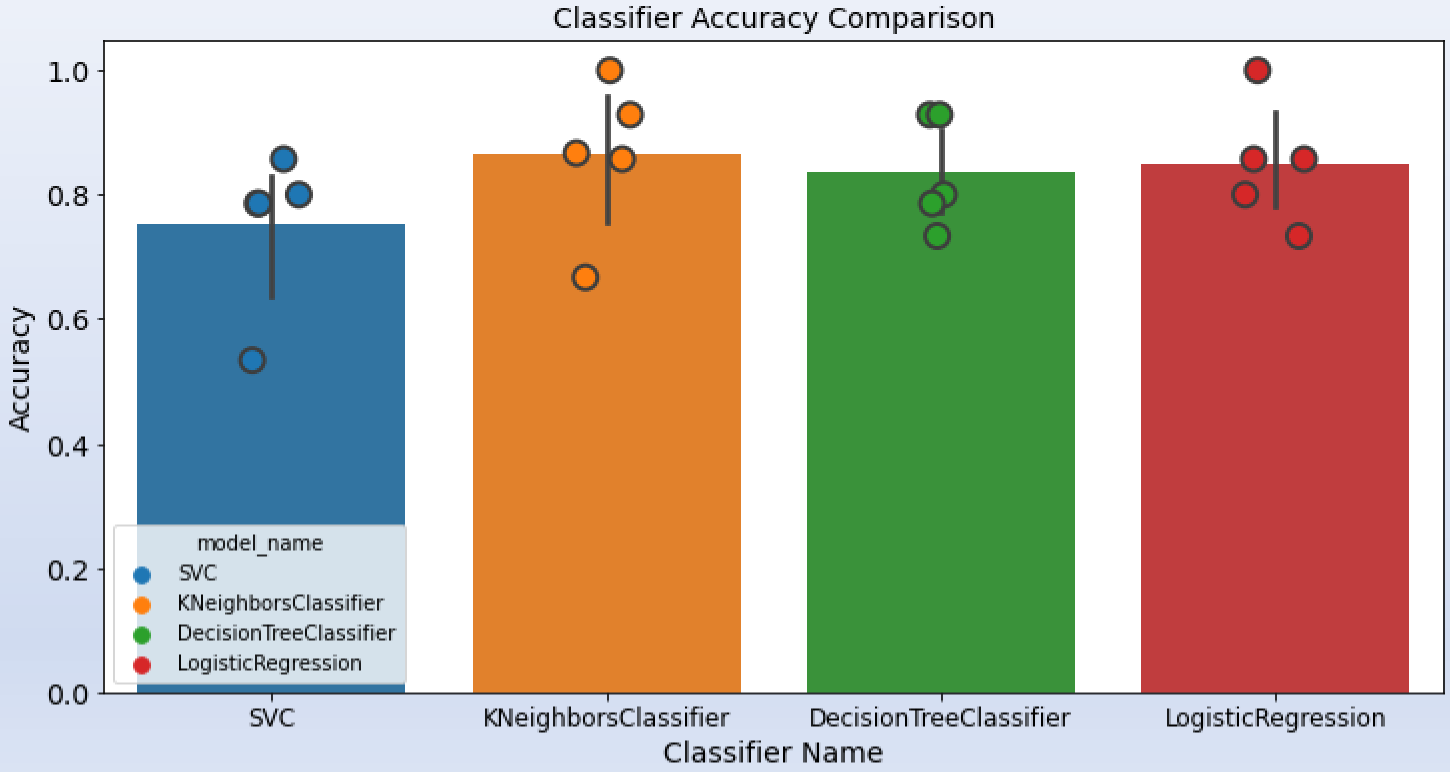
Tuned Hyperparameters

```
print("tuned hpyerparameters :(best parameters) ",knn_cv.best_params_)
print("accuracy :",knn_cv.best_score_)
```

```
tuned hpyerparameters :(best parameters)  {'algorithm': 'auto', 'n_neighbors': 1, 'p': 1}
accuracy : 0.8638095238095238
```

Classification Accuracy

- Model Bar Plots using GridSearch best parameters
- Demonstrates model accuracy for all built classification models
- The K-Nearest Neighbor Classifier has highest classification accuracy

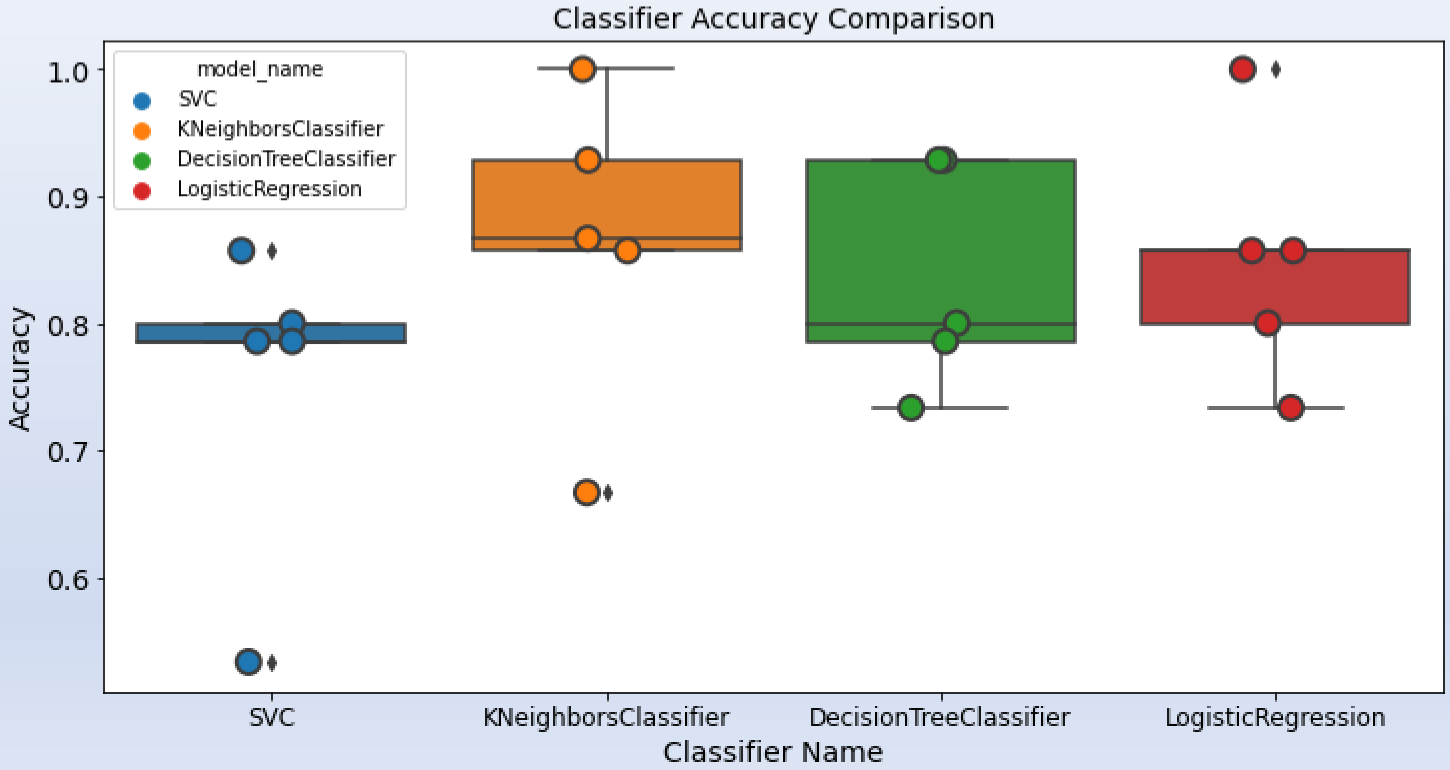



```
In [74]: cv_df.groupby('model_name').accuracy.mean()

Out[74]: model_name
         DecisionTreeClassifier    0.835238
         KNeighborsClassifier      0.863810
         LogisticRegression        0.849524
         SVC                       0.752381
         Name: accuracy, dtype: float64
```

Classification Accuracy

- Model box plots using GridSearch best parameters (Strickland, J., 2020 A, p. 125)
- Box plots provide a richer story
- Demonstrates model accuracy for all built classification models
- The K-Nearest Neighbor Classifier has highest classification accuracy

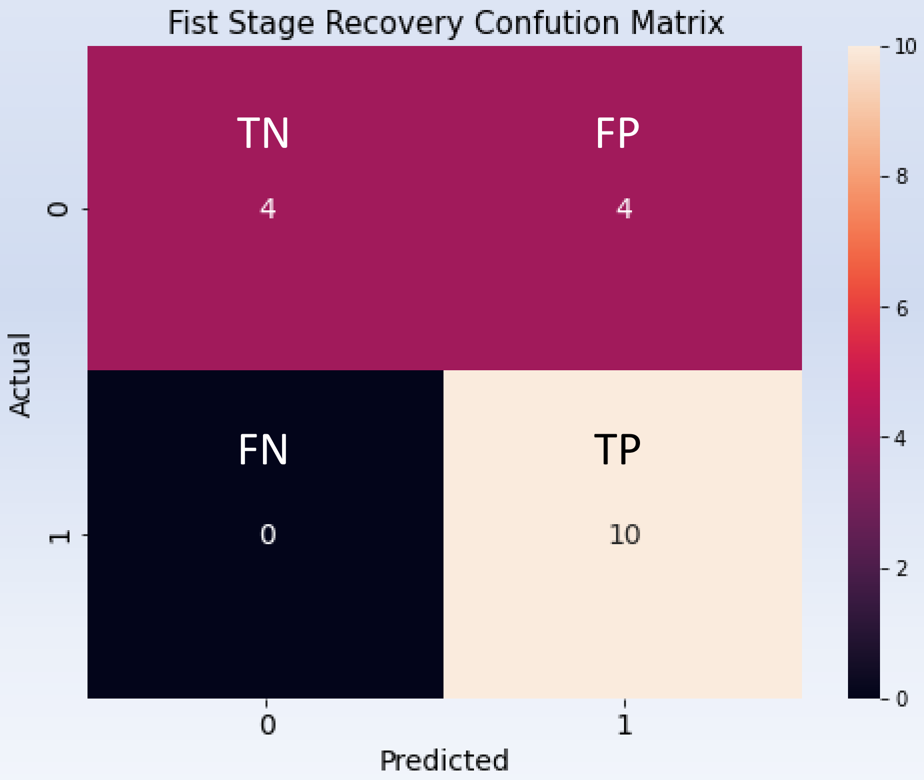


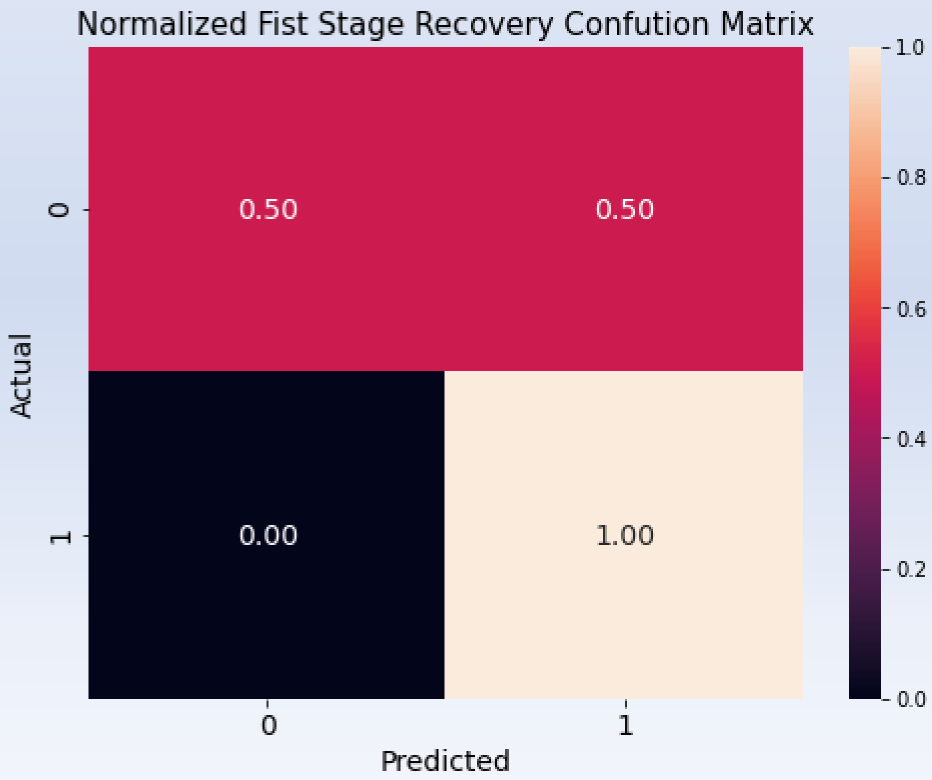

```
In [74]: cv_df.groupby('model_name').accuracy.mean()

Out[74]: model_name
         DecisionTreeClassifier    0.835238
         KNeighborsClassifier      0.863810
         LogisticRegression        0.849524
         SVC                       0.752381
         Name: accuracy, dtype: float64
```

Confusion Matrix

- Here is the confusion matrices for the K-Nearest Neighbor classifier
- It shows 100% True Positives for successful first stage recovery (Strickland, J., 2020 A, p. 131)





Conclusions

- Point 1
- Point 2
- Point 3
- Point 4
- …

Conclusions Part I: Summary of Results

- 1. Summary of your findings from EDA with data visualization
- 2. Summary of your findings from EDA with SQL
- 3. Summary of your findings from interactive visual analytics
- 4. Summary of your classification model results

- SpaceX can complete most of their stage-one recovery operations taking into account the launch sites, payloads, booster types and orbits.
- SpaceX can complete most of their stage-one recovery operations considering past launch records, payload and booster matching with launch sites
- SpaceX is using launch sites appropriately and is therefore achieving a high rate of stage-one recovery.
- SpaceX can achieve a 76% success rate of stage-one recovery according to the KNN predictive model

Conclusions Part II: Cost Benefit Analysis

Given 76% Predictive Model Accuracy, SpaceX will pay 123% less than vehicles with no stage-one recovery

Cost per launch with recovered stage-one					
$62	MM				
Cost per launch with new stage-one					
$165	MM				
Model based probability of stage-one recovery					
0.76	Weighted Average				
Future Missions					
50					
	Missions	Cost Per (Millions)	Recovery (Millions)	Cost No Model (Millions)	No Recovery (Millions)
Successful Recovery	38	$ 62.00	$ 2,356.00	$ 1,550.00	$ -
Unsuccessful Recovery	12	$ 165.00	$ 1,980.00	$ 4,125.00	$ 8,250.00
Total Cost			$ 4,336.00	$ 5,675.00	$ 8,250.00
Percent Difference (More)			-123%	-67%	-0%

REFERENCES

Strickland, Jeffrey. (2020 A). *Data Science Applications using Python and R: Text Analytics*. Lulu.com. ISBN 978-1-716-89644-6

Strickland, Jeffrey. (2020 B). *Time Series Analysis & Forecasting using Python and R*. Lulu.com. ISBN 978-1-716-45113-3

Strickland, Jeffrey. (2019 A). *Data Science Applications using R*. Lulu.com. ISBN 978-0-359-81042-0

Strickland, Jeffrey. (2019 B). *Predictive Crime Analysis using R*. Lulu.com. ISBN 978-0-359-43159-5

Strickland, Jeffrey. (2015). *Missile Flight Simulation*, 2nd Edition. Lulu.com. ISBN 978-1-329-64495-3

Wade, Mark. (2016). Launch complexes at Cape Canaveral Air Force Station. Jpg. Wikipedia Commons

US Space Force. (2020). “Space Notes: Introduction to Space Operations.”

Appendix

- None