IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Xing Wei, Chan
29th July 2022

Outline

Executive Summary

- Summary of methodologies
  - Gather Data from the following sources:
    1. SpaceX API
    2. Wikipedia
  - Parse data into suitable formats and data structure
  - Perform Exploratory Data Analysis
  - Perform Interactive Analytics
  - Fit data into various classification models
  - Determine the best model to predict if the first stage will land
  - Apply the built model into prediction of future rocket launches

- Summary of all results
  - KSC LC-39A has the highest success rate out of all launch sites
  - FT Booster has highest successful launch rate
  - All 4 classification models performs almost equally well while Decision Tree performs slightly better

Introduction

- Project background and context
  - SpaceX is an American spacecraft manufacturer, space launch provider, and a satellite communications corporation.
  - SpaceX's accomplishments include: Sending spacecraft to the International Space Station. Starlink, a satellite internet constellation providing satellite Internet access. Sending manned missions to Space.
  - Much of the savings behind SpaceX's cheap rocket launch is because SpaceX can reuse the first stage.
  - SpaceY would like to compete with SpaceX.
- Problems you want to find answers
  - Main goal is to build a machine learning model to predict whether the first stage will land successfully, which is one of the biggest determinant on the cost of the rocket launch.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology
  - A portion of data are collected from requesting from SpaceX API and stored as a dataframe
  - The rest of the data are scraped from SpaceX Past Launch Record (Wikipedia) and stored as a dataframe
- Perform data wrangling
  - Filter to only ‘Falcon 9’ booster version
  - Replace all NULL values from the Payload Mass with the average value
  - Create ‘Class’ column from ‘Landing Outcome’ column to change all different outcomes to 0 (Failed) or 1 (Successful)
- Perform exploratory data analysis (EDA) using visualization and SQL

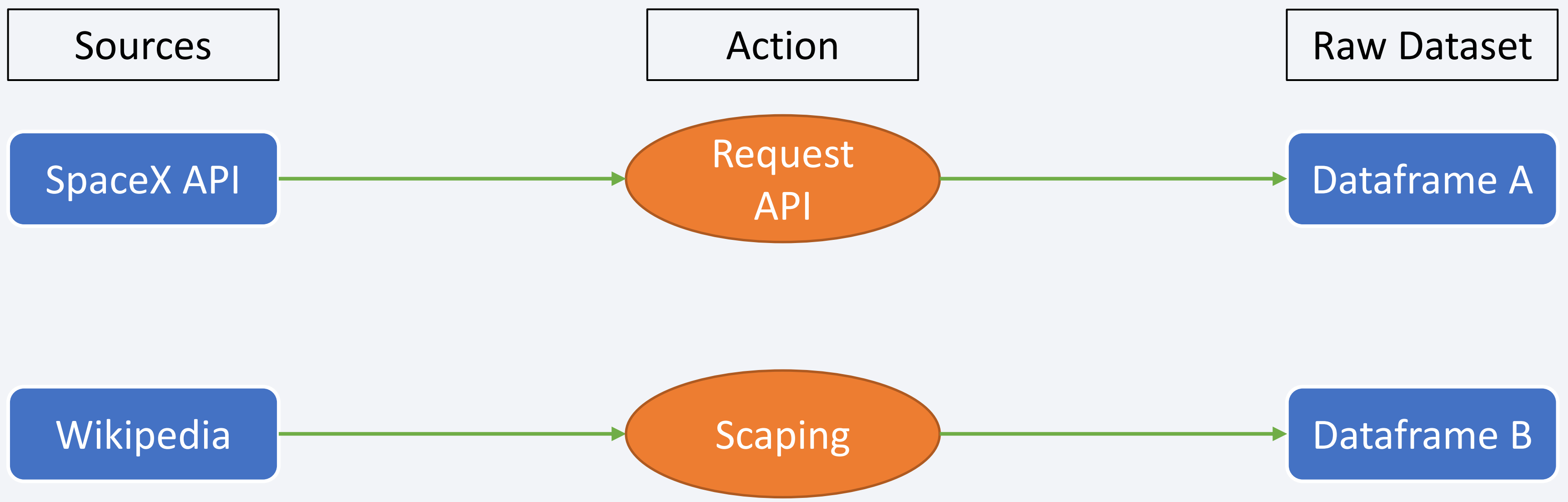
Methodology

Executive Summary

- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Using StandardScaler to normalize the train dataset X
  - The train dataset Y is the 'Class' column of the dataframe
  - X and Y are split into 80% train dataset and 20% test dataset with train_test_split()
  - 4 classification models are tested
    (Linear Regression, Decision Tree, Support Vector Machines, K-Nearest Neighbors4)
  - Use GridSearchCV to test all relevant parameters to find the best parameters for each of the model
  - Using the accuracy scoring and confusion matrix to determine which model performs the best in predicting whether the first stage of a rocket launch will land successfully

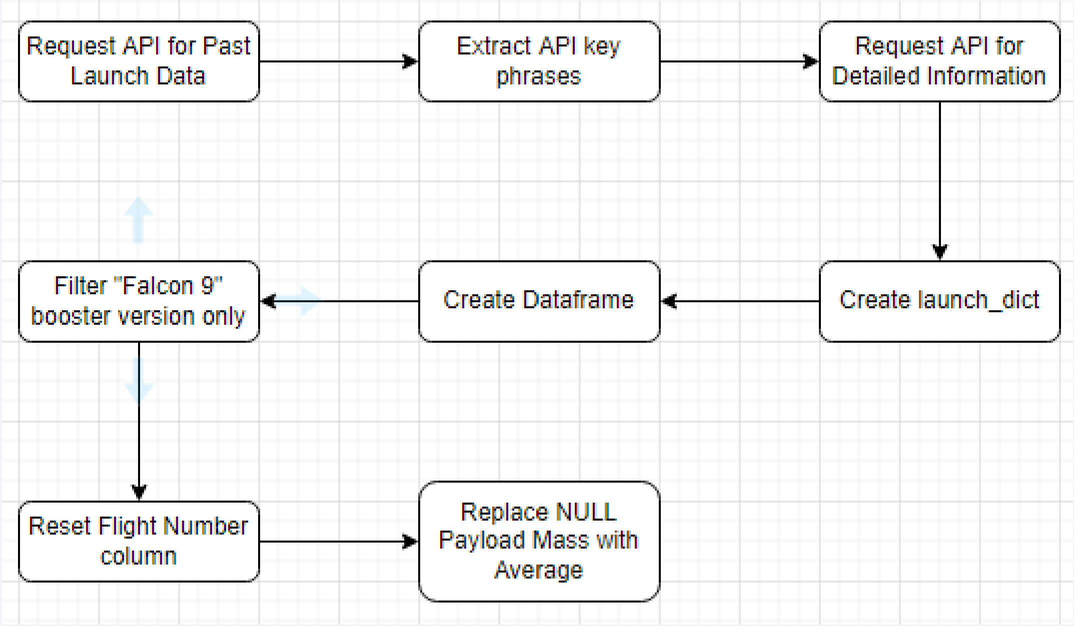
Data Collection

- Collection of datasets
  - 'rocket', 'payloads', 'launchpad', 'cores', 'flight_number', 'date_utc’ are collected from the calling SpaceX API (https://api.spacexdata.com/v4/...) and stored as a dataframe A
  - ‘Flight No.’, ‘Launch site’, ‘Payload’, ‘Payload mass’, ‘Orbit’, ‘Customer’, ‘Launch outcome’, ‘Version Booster’, ‘Booster landing’ are scraped from Wikipedia (Here) and stored as a dataframe B
- You need to present your data collection process use key phrases and flowcharts



Data Collection – SpaceX API

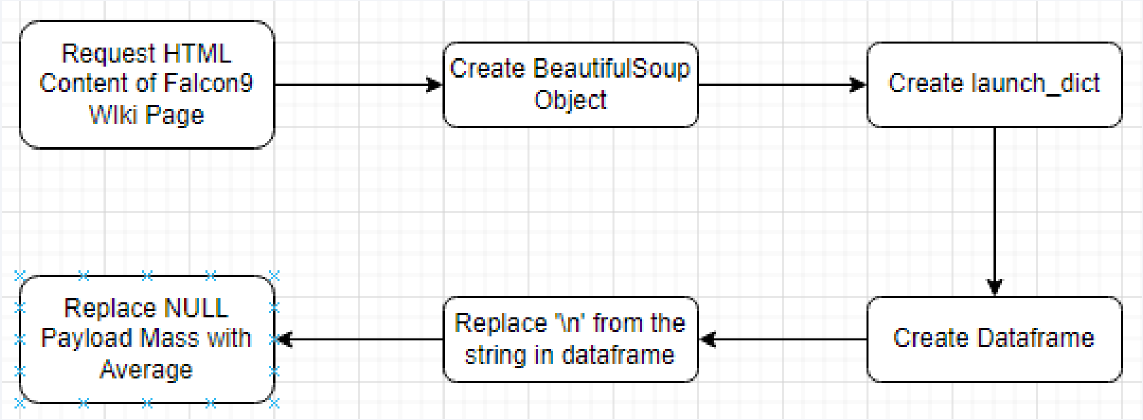
- Request past launch data from SpaceX API
- Extract Key phrases from past launch data
- Request SpaceX API for detailed information of all key phrases collected
- Create launch_dict dictionary to contain all useful data
- Create a dataframe from the launch_dict
- Filter dataframe to contain “Falcon 9” booster version only
- Reset the “Flight Number” column
- Replace the NULL value from “Payload Mass” column with the mean value



Jupyter Notebook (Github):https://github.com/pi31416chan/Coursera-Applied-Data-Science-Capstone/blob/07e61d917346674b1d013b39d6612eda41309e1b/1.%20Data%20Collection%20API.ipynb

Data Collection - Scraping

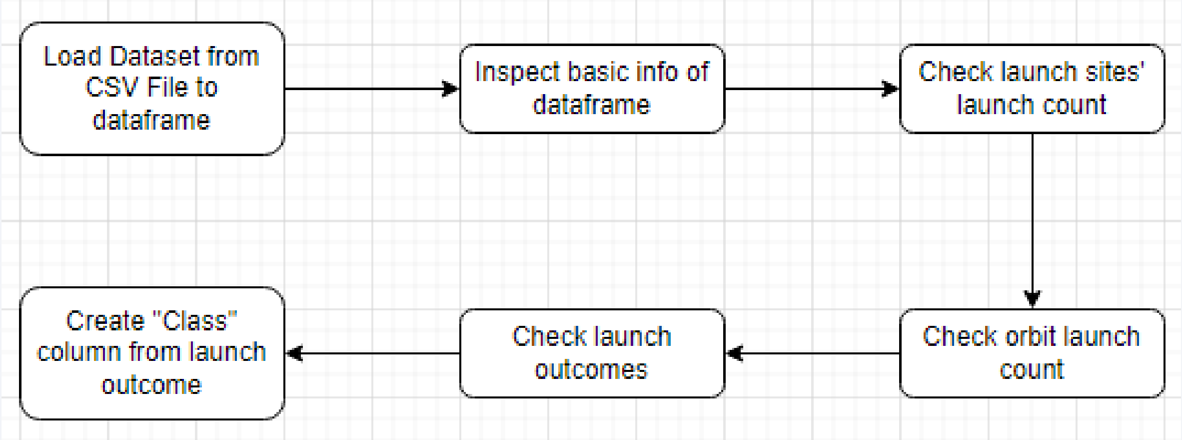
- Request HTML content from Falcon 9 Wikipedia page
- Create BeautifulSoup object from the HTML content
- Extract the table containing required information from the BeautifulSoup object
- Create launch_dict dictionary
- Create a dataframe from the launch_dict
- Replace ‘\n’ from the strings in the dataframe
- Replace the NULL value from “Payload Mass” column with the mean value



Jupyter Notebook (Github): https://github.com/pi31416chan/Coursera-Applied-Data-Science-Capstone/blob/07e61d917346674b1d013b39d6612eda41309e1b/2.%20Data%20Collection%20with%20Web%20Scraping.ipynb

Data Wrangling

- Load previously collected dataset from CSV file into dataframe
- Inspect basic information of the dataframe (dtypes, describe(), shape, …)
- Check launch count of different launch sites
- Check launch count of different orbits
- Check launch outcomes
- Create “Class” column from launch outcome to convert launch outcome from text to 0 (Failure) or 1 (Successful)



Jupyter Notebook (Github): https://github.com/pi31416chan/Coursera-Applied-Data-Science-Capstone/blob/07e61d917346674b1d013b39d6612eda41309e1b/3.%20Data%20wrangling.ipynb

EDA with Data Visualization

- Scatter Plot of Flight Number vs Payload Mass (hue = 'Class')
  - To see how the successful rate (Class) changes with Flight Number or Payload Mass
- Scatter Plot of Flight Number vs Launch Site (hue = 'Class')
  - To visualize the relationship between Flight Number, Launch Site and Class
- Scatter Plot of Payload Mass vs Launch Site (hue = 'Class')
  - To visualize the relationship between Payload Mass, Launch Site and Class
- Bar Chart for the sucess rate of each orbit
  - To visualize the relationship between success rate of each orbit type
- Scatter Plot of Flight Number vs Orbit Type (hue = 'Class')
  - To visualize the relationship between FlightNumber and Orbit type
- Scatter Plot of Payload Mass vs Orbit Type (hue = 'Class')
  - To visualize the relationship between Payload Mass and Orbit type
- Line Chart of Year vs Success Rate
  - To visualize the launch success yearly trend

Jupyter Notebook (Github): https://github.com/pi31416chan/Coursera-Applied-Data-Science-Capstone/blob/07e61d917346674b1d013b39d6612eda41309e1b/5.%20Exploratory%20Data%20Analysis%20with%20Data%20Visualization.ipynb

EDA with SQL

1. SELECT DISTINCT(Launch_Site) FROM spacex_capstone.spacextbl;
2. SELECT Launch_Site FROM spacex_capstone.spacextbl WHERE Launch_Site LIKE "CCA%" LIMIT 5;
3. SELECT SUM(PAYLOAD_MASS__KG_) FROM spacex_capstone.spacextbl WHERE Customer = "NASA (CRS)";
4. SELECT AVG(PAYLOAD_MASS__KG_) FROM spacex_capstone.spacextbl WHERE Booster_Version = "F9 v1.1";
5. SELECT MIN(`Date`),Landing_Outcome FROM spacex_capstone.spacextbl WHERE Landing_Outcome = "Success (ground pad)";
6. SELECT Booster_Version FROM spacex_capstone.spacextbl WHERE Landing_Outcome = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000;
7. SELECT Mission_Outcome,COUNT(Mission_Outcome) FROM spacex_capstone.spacextbl GROUP BY Mission_Outcome;
8. SELECT Booster_Version,PAYLOAD_MASS__KG_ FROM spacex_capstone.spacextbl WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM spacex_capstone.spacextbl);
9. SELECT Landing_Outcome,Booster_Version,Launch_Site FROM spacex_capstone.spacextbl WHERE Landing_Outcome LIKE "%drone%" AND Landing_Outcome LIKE "%Fail%" AND YEAR(`Date`) = 2015;
10. SELECT Landing_Outcome,COUNT(Landing_Outcome) FROM spacex_capstone.spacextbl WHERE `Date` BETWEEN '2010-06-04' AND '2017-03-20’ GROUP BY Landing_Outcome ORDER BY COUNT(Landing_Outcome) DESC;

Jupyter Notebook (Github): https://github.com/pi31416chan/Coursera-Applied-Data-Science-Capstone/blob/07e61d917346674b1d013b39d6612eda41309e1b/4.%20Exploratory%20Data%20Analysis%20with%20SQL.ipynb

Build an Interactive Map with Folium

1. Circle
   1. To highlight the area of NASA headquarter & multiple launch sites
2. Marker
   1. To mark the coordinate of each rocket launch
3. MarkerCluster
   1. To hold nearby markers into a cluster
4. Icon
   1. To illustrate the launch outcome of each rocket launch with color
5. MousePosition
   1. To get the coordinate where the mouse pointer points at in the map
6. Line
   1. To illustrate the distance from the nearest public objects (City, Highway, Railway)

Jupyter Notebook (Github): https://github.com/pi31416chan/Coursera-Applied-Data-Science-Capstone/blob/07e61d917346674b1d013b39d6612eda41309e1b/6.%20Interactive%20Visual%20Analytics%20with%20Folium.ipynb

Build a Dashboard with Plotly Dash

Interactions:

1. Dropdown List
   - To select the Launch Site
2. Range Slider Bar
   - To select the range of Payload Mass

Charts:

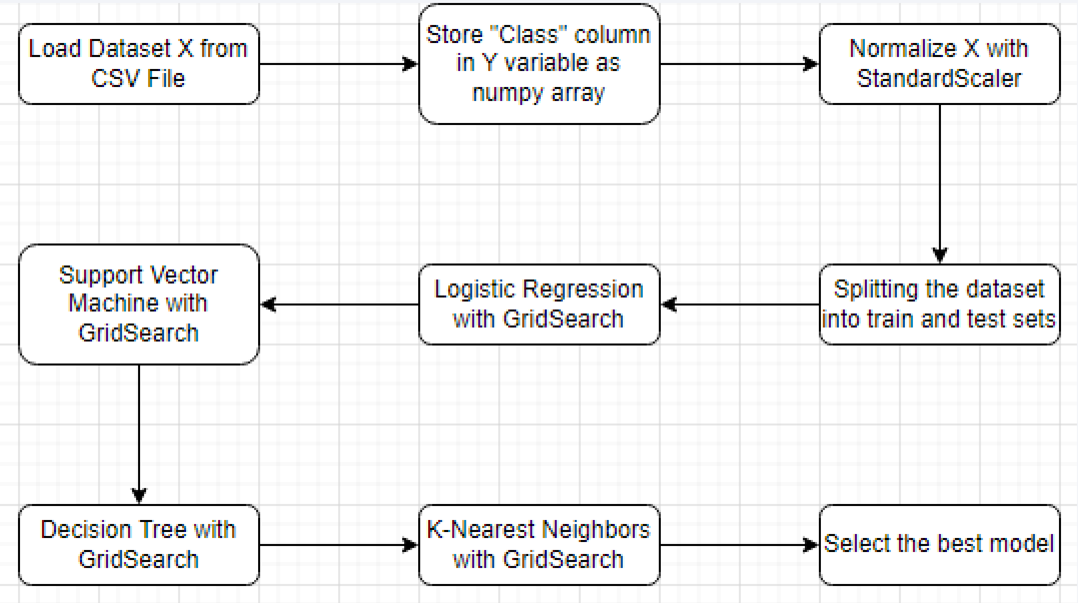
1. Pie Chart
   - To visualize the Success Rate of different launch sites
2. Scatter Plot
   - To visualize the Payload Mass, Success Rate and Booster Version

Jupyter Notebook (Github): https://github.com/pi31416chan/Coursera-Applied-Data-Science-Capstone/blob/07e61d917346674b1d013b39d6612eda41309e1b/7.%20Interactive%20Dashboard.ipynb

Source Code (Python): https://github.com/pi31416chan/Coursera-Applied-Data-Science-Capstone/blob/07e61d917346674b1d013b39d6612eda41309e1b/7.%20Interactive%20Dashboard.py

Predictive Analysis (Classification)

- Load dataset from CSV file as X
- Store the "Class" column into Y as a numpy array
- Normalize data of X with StandardScaler
- Split the dataset into train (80%) and test (20%) datasets
- Use GridSearchCV to get best parameters for Logistic Regression, plot confusion matrix
- Use GridSearchCV to get best parameters for Support Vector Machine , plot confusion matrix
- Use GridSearchCV to get best parameters for Decision Tree , plot confusion matrix
- Use GridSearchCV to get best parameters for K-Nearest Neighbors , plot confusion matrix
- Use accuracy score and confusion matrix to select the best classification model

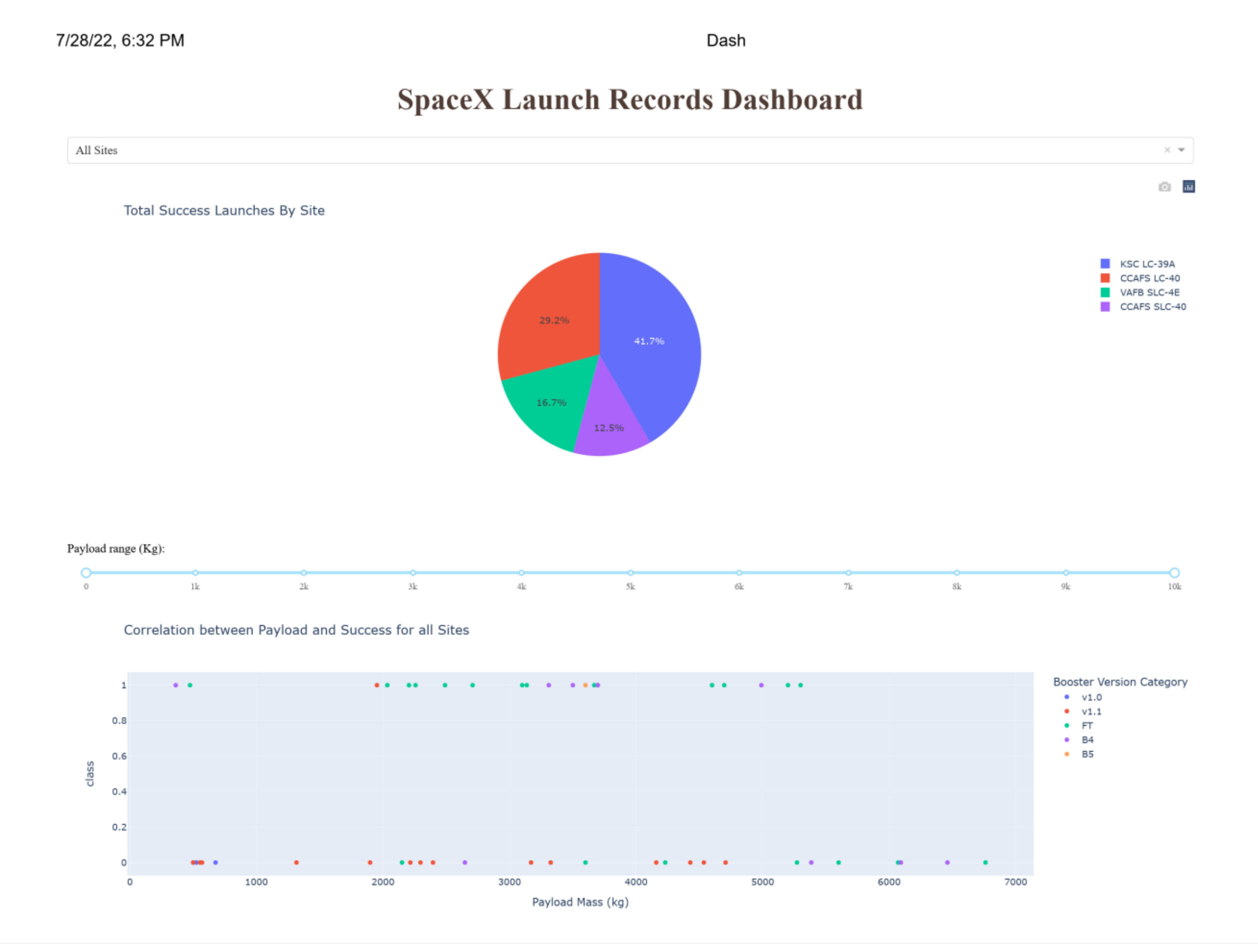


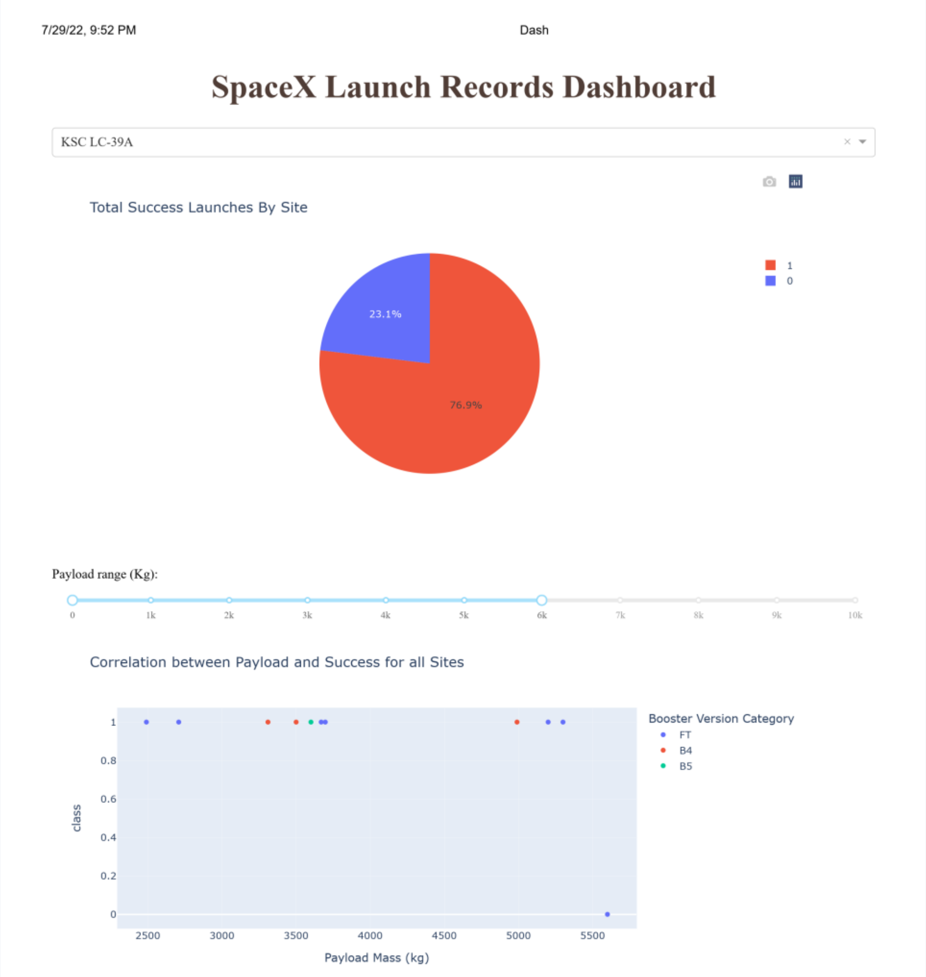
Jupyter Notebook (Github): https://github.com/pi31416chan/Coursera-Applied-Data-Science-Capstone/blob/07e61d917346674b1d013b39d6612eda41309e1b/8.%20Machine%20Learning%20Prediction.ipynb (Ignore the long auto-generated warning due to some features are deprecated and scroll to the bottom, thanks)

Results (EDA)

- Only 4 launch sites are involved
  - CCAFS LC-40, CCAFS SLC-40, KSC LC-39A, VAFB SLC-4E
- KSC LC-39A site has largest successful launch number & highest successful launch rate
  - 100% success rate at payload <= 5000 and very high success rate at payload > 8000
- Success Rate of heavier Payload appears to be higher
- SSO orbit type has 100% success rate
- VLEO orbit type has very high success rate (~87%)
- Success Rate has been increasing since 2013

Results (Interactive Analytics)





Results

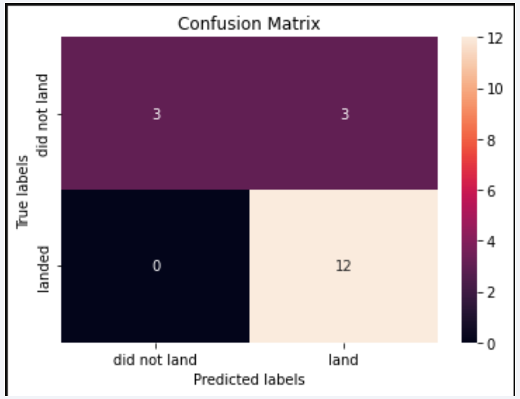
Logistic Regression

Best Parameters:

- C = 0.01

Accuracy Score (Test Dataset):

- 0.8333



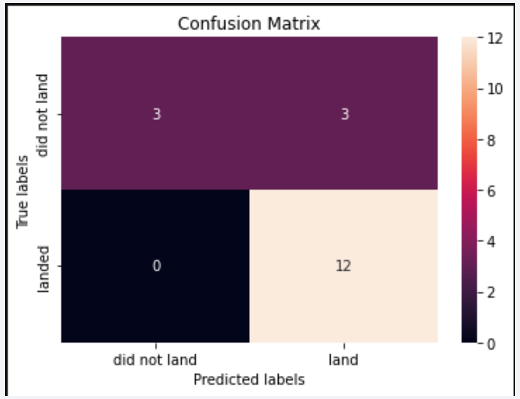
Decision Tree

Best Parameters:

- criterion = entropy
- max_depth = 4
- max_features = sqrt
- splitter = random

Accuracy Score (Test Dataset):

- 0.8333



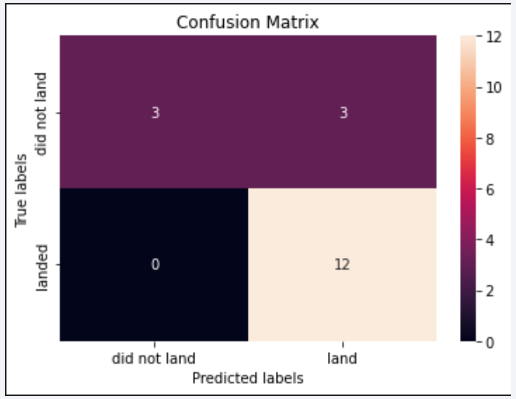
Support Vector Machine

Best Parameters:

- C = 1.0
- gamma = 0.03162277…
- kernel = sigmoid

Accuracy Score (Test Dataset):

- 0.8333



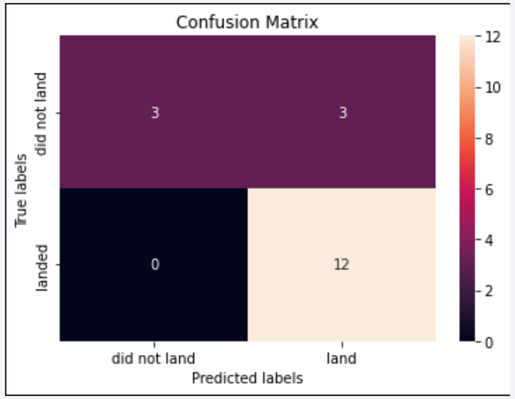
Support Vector Machine

Best Parameters:

- algorithm = auto
- n_neighbors = 10
- p = 1

Accuracy Score (Test Dataset):

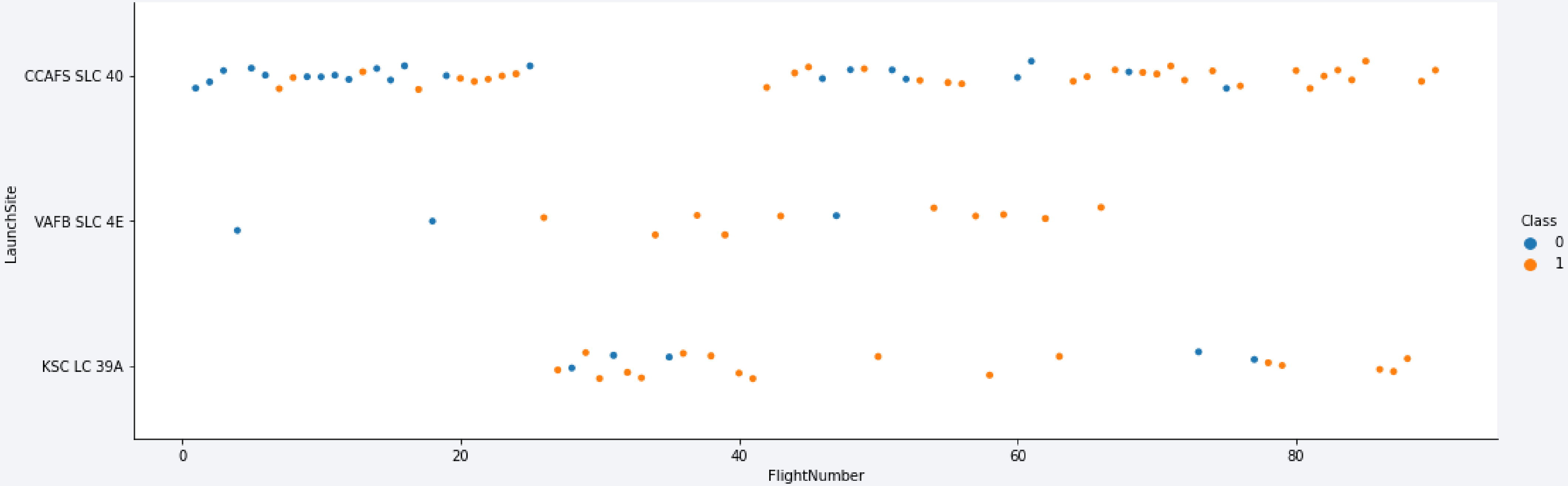
- 0.8333



Section 2

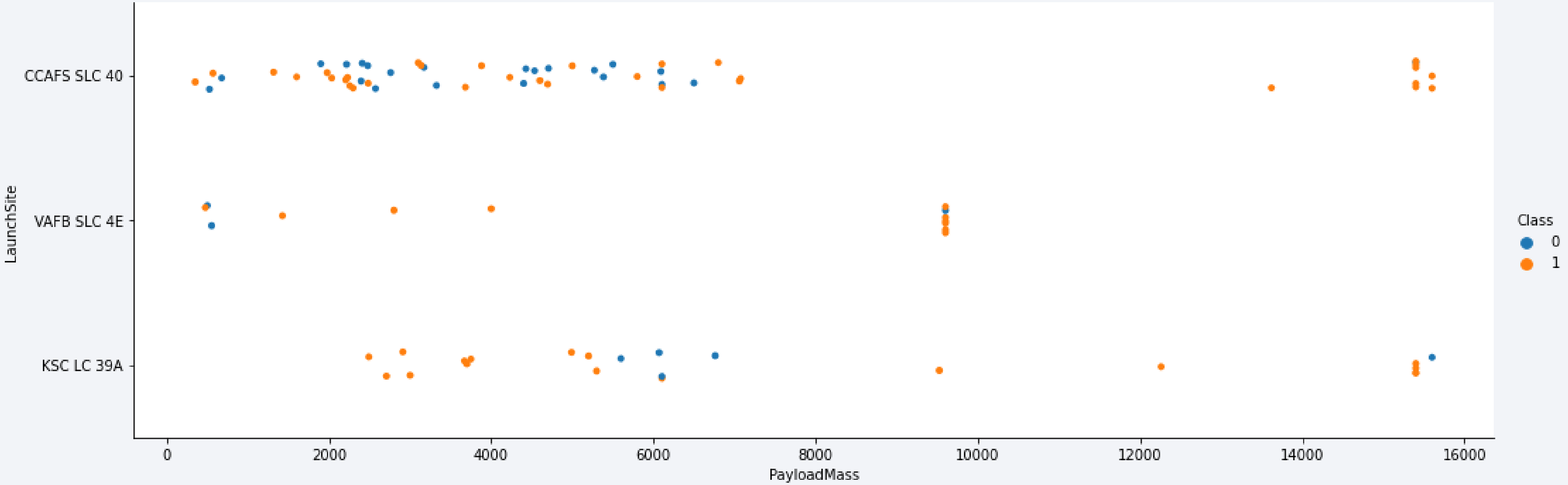
Insights drawn from EDA

Flight Number vs. Launch Site



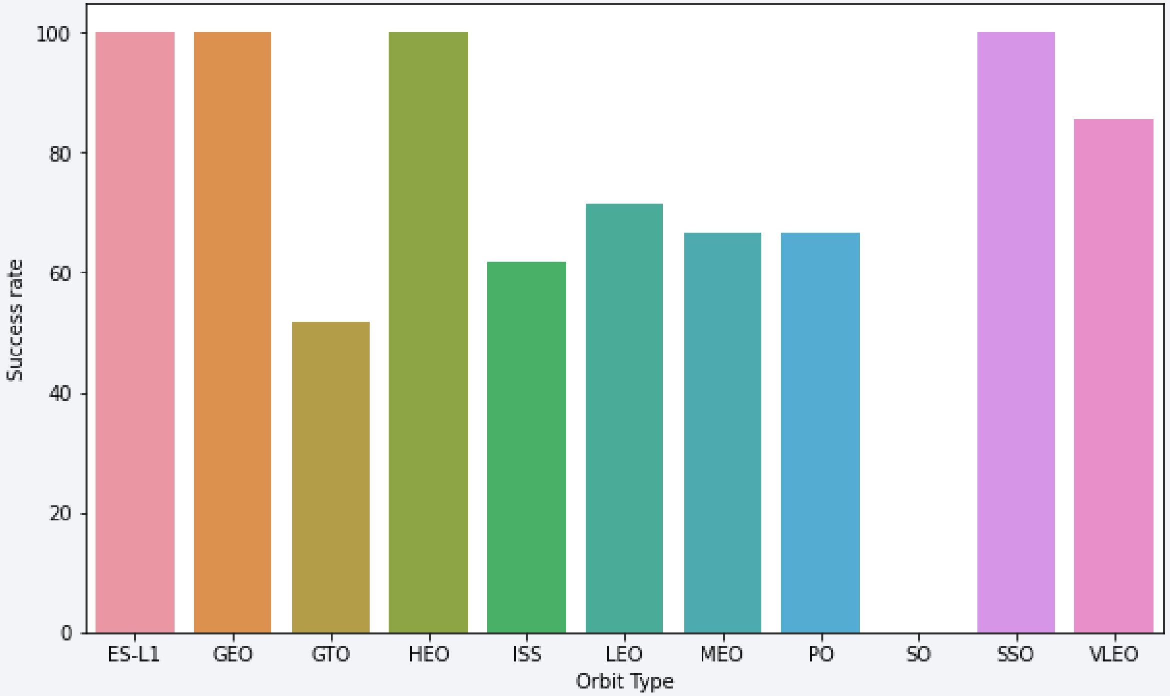
- Flight Number and Launch Site does not have a very strong relationship with each other
- There is higher success rate in recent launches (Flight Number > 80)

Payload vs. Launch Site



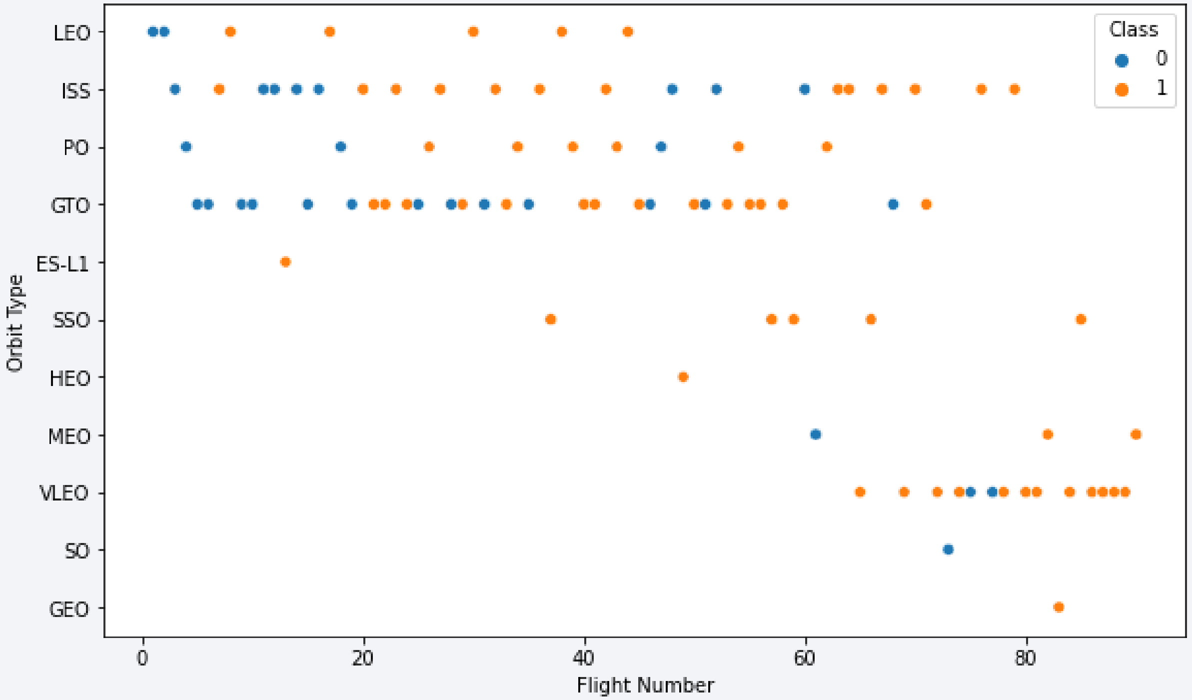
- CCAFS SLC 40 has 100% success rate when payload mass > 12000
- VAFB SLC 4E does not have heavy payload (> 10000)
- KSC LC 39A has 100% success rate when payload mass < 5500 (approximately)

Success Rate vs. Orbit Type



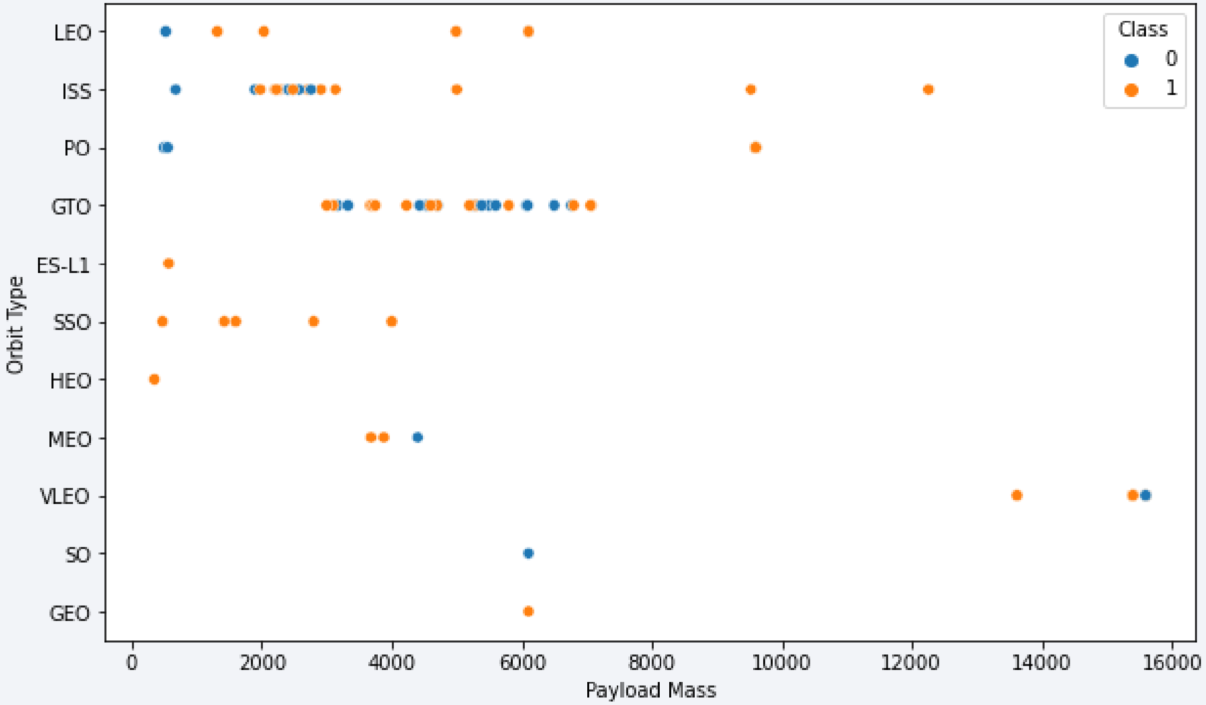
- ES-L1, GEO, HEO, and SSO orbit type has 100% success rate
- VLEO has very high success rate (>80%)

Flight Number vs. Orbit Type



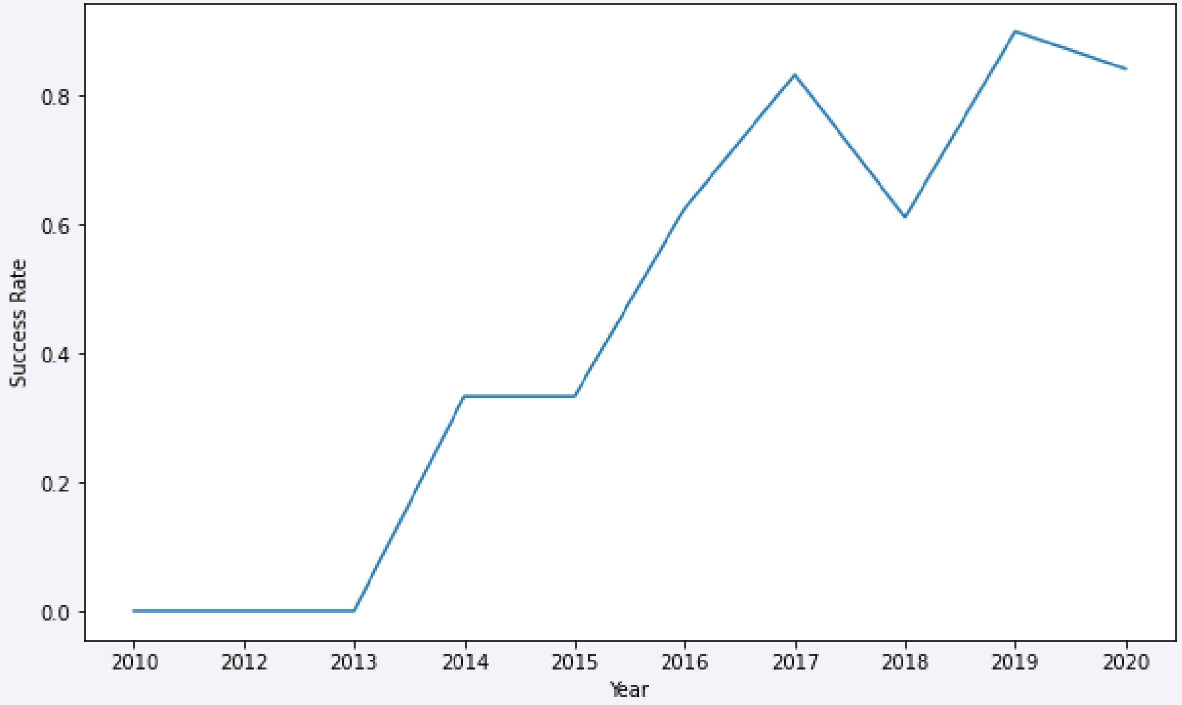
- LEO orbit the Success appears to increase when the number of flights increase
- There seems to be no relationship between flight number when in GTO orbit
- SSO orbit type has 100% success rate while the rest of orbit types with 100% successrate has only 1 launch count

Payload vs. Orbit Type



- With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS
- For GTO we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission) are both there

Launch Success Yearly Trend



- The sucess rate since 2013 kept increasing until 2020 which shows a slight drop

SQL Results Notice

- Please take note I conducted the lab with a local MySQL database, this is because I was unable to connect to IBM_DB2 service when using the magic command (%sql), kindly refer to the discussion section of this capstone project course if you don't face this issue yourself to understand more about this issue
- As a result, the results of the below section might be displayed in a slightly different way because I don't know how differently MySQL behaves compared to IBM_DB2
- Thanks for understanding

All Launch Site Names

- Query:
  - SELECT DISTINCT(Launch_Site) FROM spacex_capstone.spacextbl;
- Result:
  - [('CCAFS LC-40',), ('VAFB SLC-4E',), ('KSC LC-39A',), ('CCAFS SLC-40',)]
- Explanation:
  - There are only 4 launch sites available from the dataset, CCAFS LC-40, VAFB SLC-4E, KSC LC-39A, CCAFS SLC-40

Launch Site Names Begin with 'CCA'

- Query:
  - SELECT Launch_Site FROM spacex_capstone.spacextbl WHERE Launch_Site LIKE "CCA%" LIMIT 5;
- Result:
  - [('CCAFS LC-40’,), ('CCAFS LC-40',), ('CCAFS LC-40',), ('CCAFS LC-40',), ('CCAFS LC-40',)]
- Explanation:
  - There are only first 5 launch sites begin with “CCA” are shown above

Total Payload Mass

- Query:
  - SELECT SUM(PAYLOAD_MASS__KG_) FROM spacex_capstone.spacextbl WHERE Customer = "NASA (CRS)";
- Result:
  - [(Decimal('45596'),)]
- Explanation:
  - The sum of all payload mass sent by “NASA (CRS)” is 45596 KG

Average Payload Mass by F9 v1.1

- Query:
  - SELECT AVG(PAYLOAD_MASS__KG_) FROM spacex_capstone.spacextbl WHERE Booster_Version = "F9 v1.1";
- Result:
  - [(Decimal('2928.4000'),)]
- Explanation:
  - The average payload mass by “F9 v1.1” booster is 2928.40 KG

First Successful Ground Landing Date

- Query:
  - SELECT MIN(`Date`),Landing_Outcome FROM spacex_capstone.spacextbl WHERE Landing_Outcome = "Success (ground pad)";
- Result:
  - [(datetime.date(2015, 12, 22), 'Success (ground pad)')]
- Explanation:
  - The first successful ground landing date is 2015-15-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- Query:
  - SELECT Booster_Version FROM spacex_capstone.spacextbl
    WHERE Landing_Outcome = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000;
- Result:
  - [('F9 FT B1022',), ('F9 FT B1026',), ('F9 FT B1021.2',), ('F9 FT B1031.2',)]
- Explanation:
  - The 4 boosters with successful drone ship landing with payload between 4000 and 6000 kg are shown above

Total Number of Successful and Failure Mission Outcomes

- Query:
  - SELECT Mission_Outcome,COUNT(Mission_Outcome) FROM spacex_capstone.spacextbl GROUP BY Mission_Outcome;
- Result:
  - [('Success', 98),
    ('Failure (in flight)', 1),
    ('Success (payload status unclear)', 1),
    ('Success ', 1)]
- Explanation:
  - There are 100 successful mission and 1 failed mission outcomes

Boosters Carried Maximum Payload

- Query:
  - SELECT Booster_Version,PAYLOAD_MASS__KG_ FROM spacex_capstone.spacextbl
    WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM spacex_capstone.spacextbl);
- Result:
  - [('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)]
- Explanation:
  - The names of the boosters carried maximum payload are shown above

2015 Launch Records

- Query:
  - SELECT Landing_Outcome,Booster_Version,Launch_Site FROM spacex_capstone.spacextbl WHERE
    Landing_Outcome LIKE "%drone%" AND
    Landing_Outcome LIKE "%Fail%" AND
    YEAR(`Date`) = 2015;
- Result:
  - [('Failure (drone ship)', 'F9 v1.1 B1012', 'CCAFS LC-40'),
    ('Failure (drone ship)', 'F9 v1.1 B1015', 'CCAFS LC-40')]
- Explanation:
  - The failed landing outcome details are shown above

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

- Query:
  - SELECT Landing_Outcome,COUNT(Landing_Outcome) FROM spacex_capstone.spacextbl
    WHERE `Date` BETWEEN '2010-06-04' AND '2017-03-20’
    GROUP BY Landing_Outcome
    ORDER BY COUNT(Landing_Outcome) DESC;
- Result:
  - [('No attempt', 10), ('Failure (drone ship)', 5), ('Success (drone ship)', 5), ('Controlled (ocean)', 3), ('Success (ground pad)', 3), ('Failure (parachute)', 2), ('Uncontrolled (ocean)', 2), ('Precluded (drone ship)', 1)]
- Explanation:
  - The ranking of landing outcomes between 2010-06-04 and 2017-03-20 are shown above

Section 3

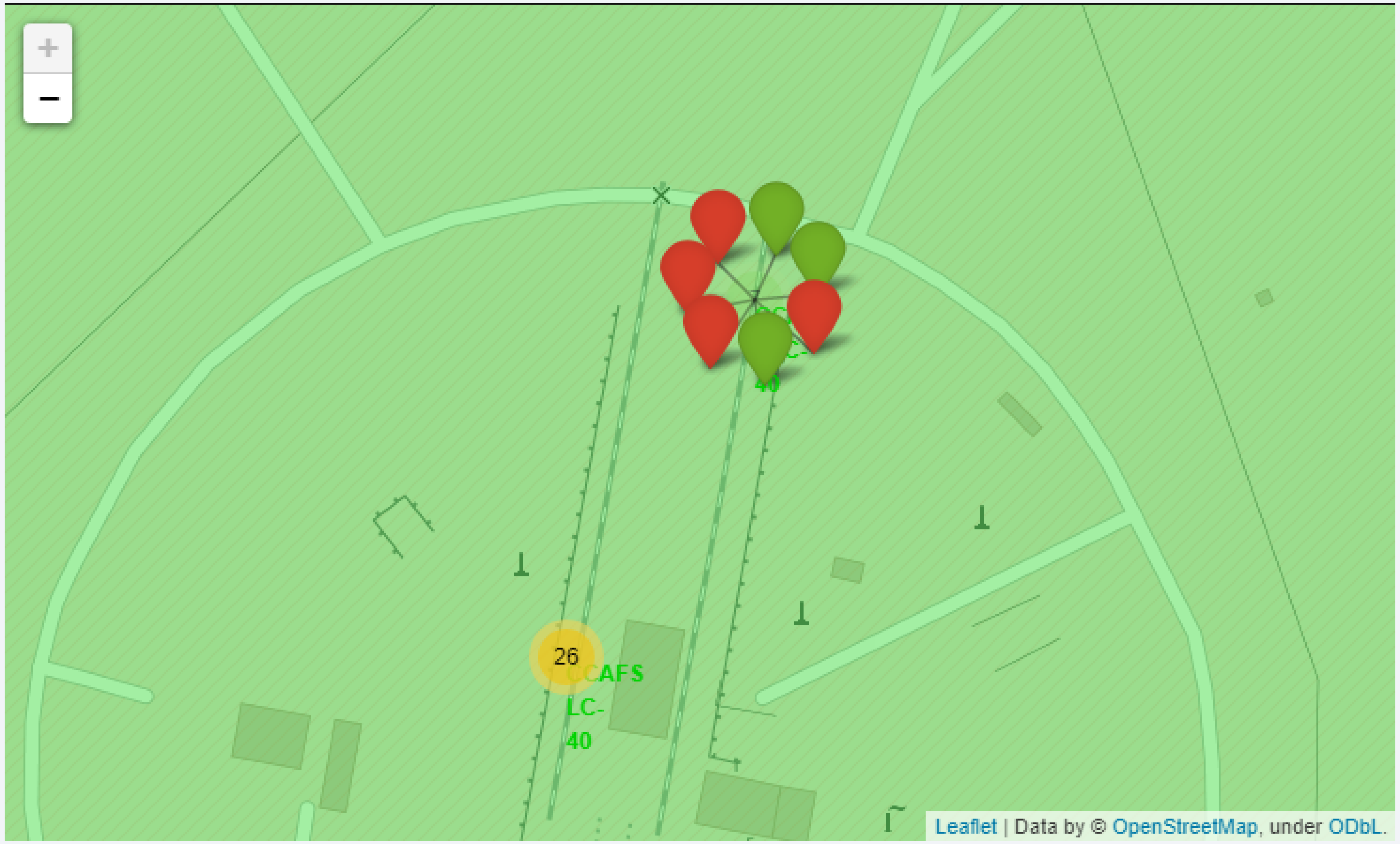
Launch Sites Proximities Analysis

Launch Sites Location

- All launch sites are located nearby coastline
- All launch sites are located at around 30 degree north from equator

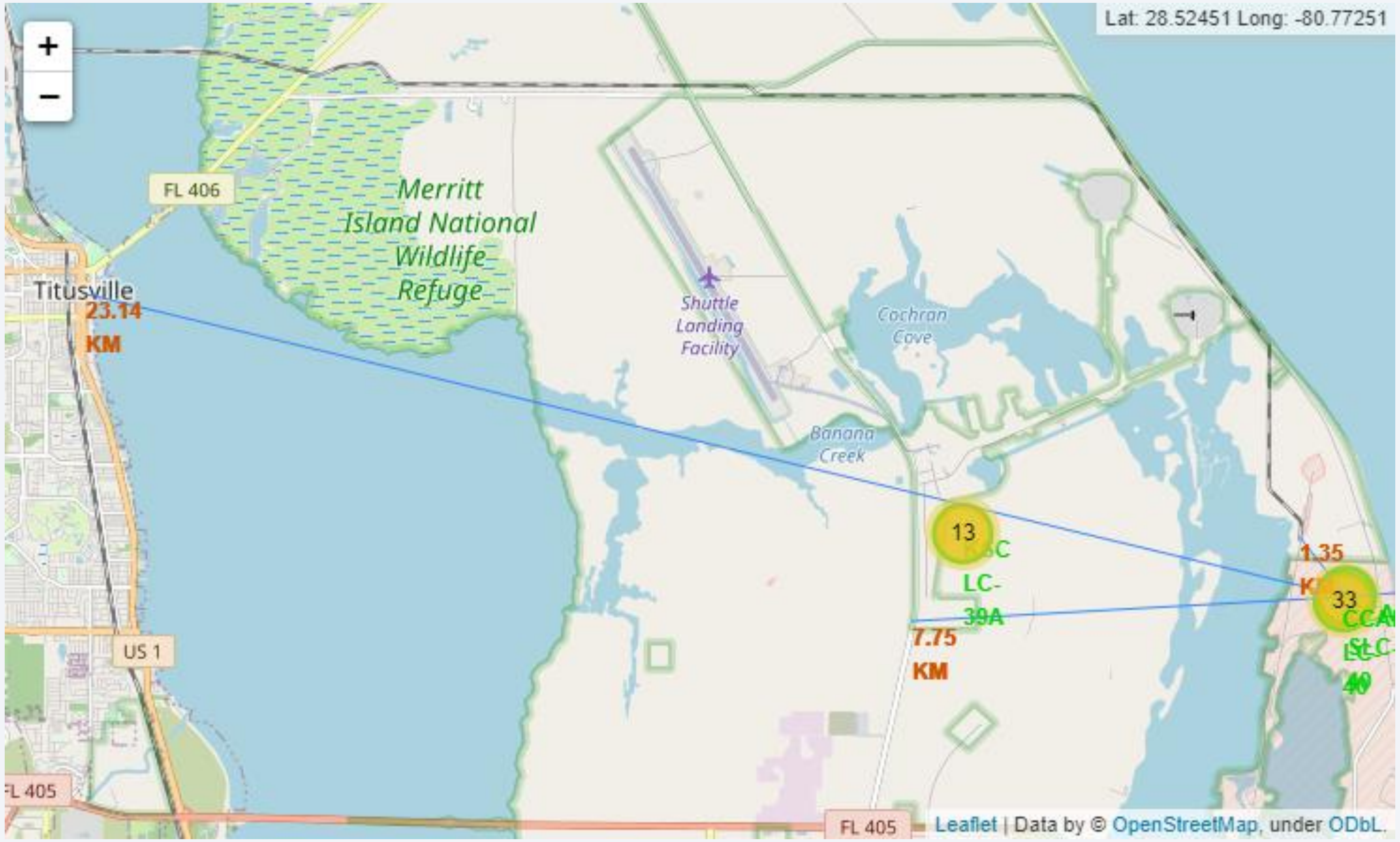
Landing Outcome of Launches in Each Site

- The number 26 and 7 are the clusters of launch markers at each site
- Green marker represents successful launch outcome
- Red marker represents unsuccessful launch outcome
- NOTE: There is an issue with folium unable to load the icon on each marker, please do understand this is the reason why the icons are not displayed properly



Distances to nearest attractions

- The line display the direction and distance from the launch site to the nearest attractions
- The text in red display the distance from the launch site to the nearest attractions
- City (Titusville): 23.14 KM
- Highway: 7.75 KM
- Railway: 1.35 KM

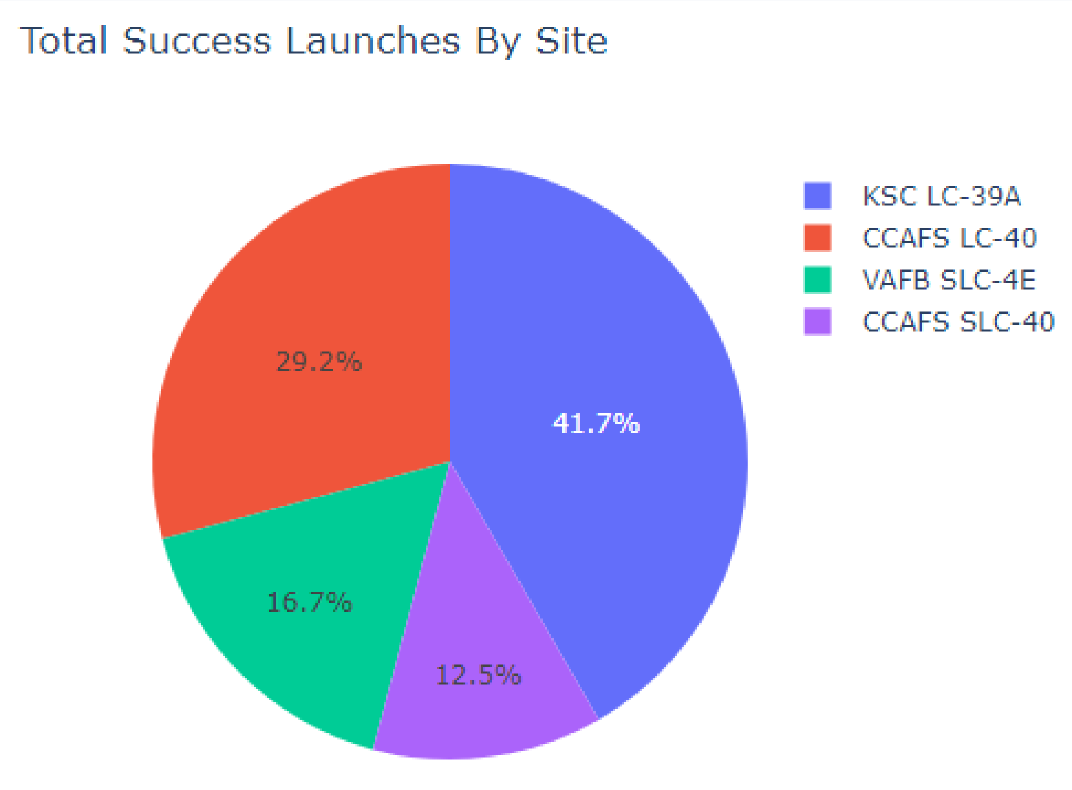


Section 4

Build a Dashboard with Plotly Dash

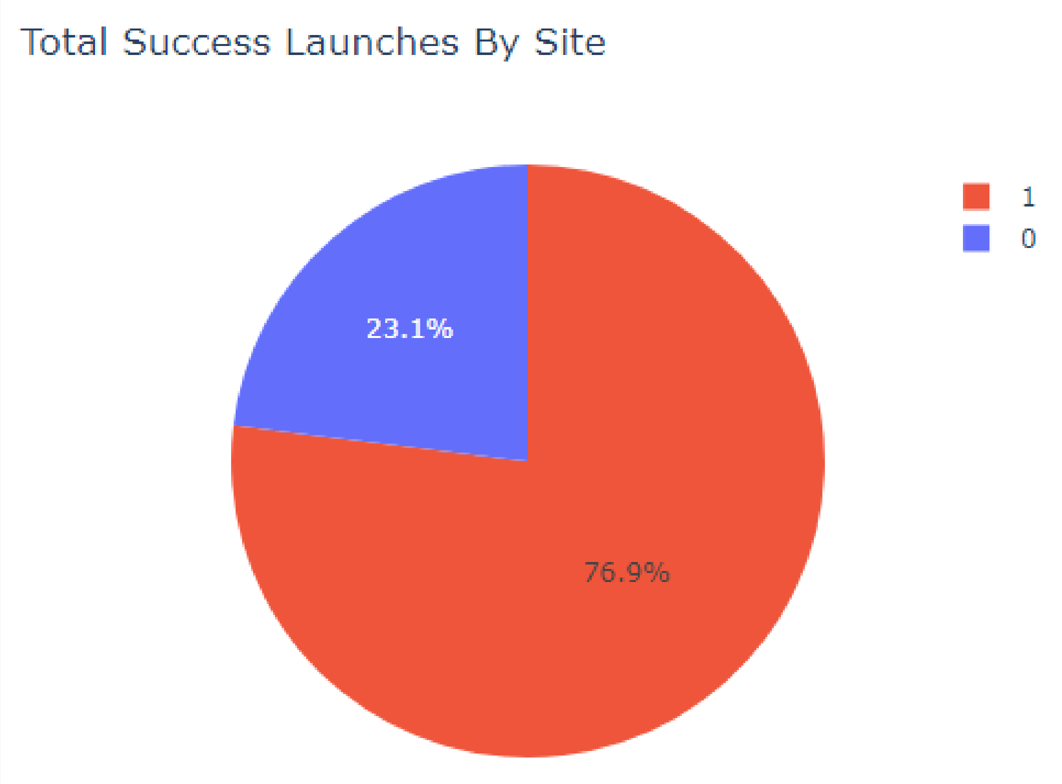
Total Success Launches By Site

- KSC LC-39A has the highest number of successful launches
- CCAFS SLC-40 has the lowest number of successful launches

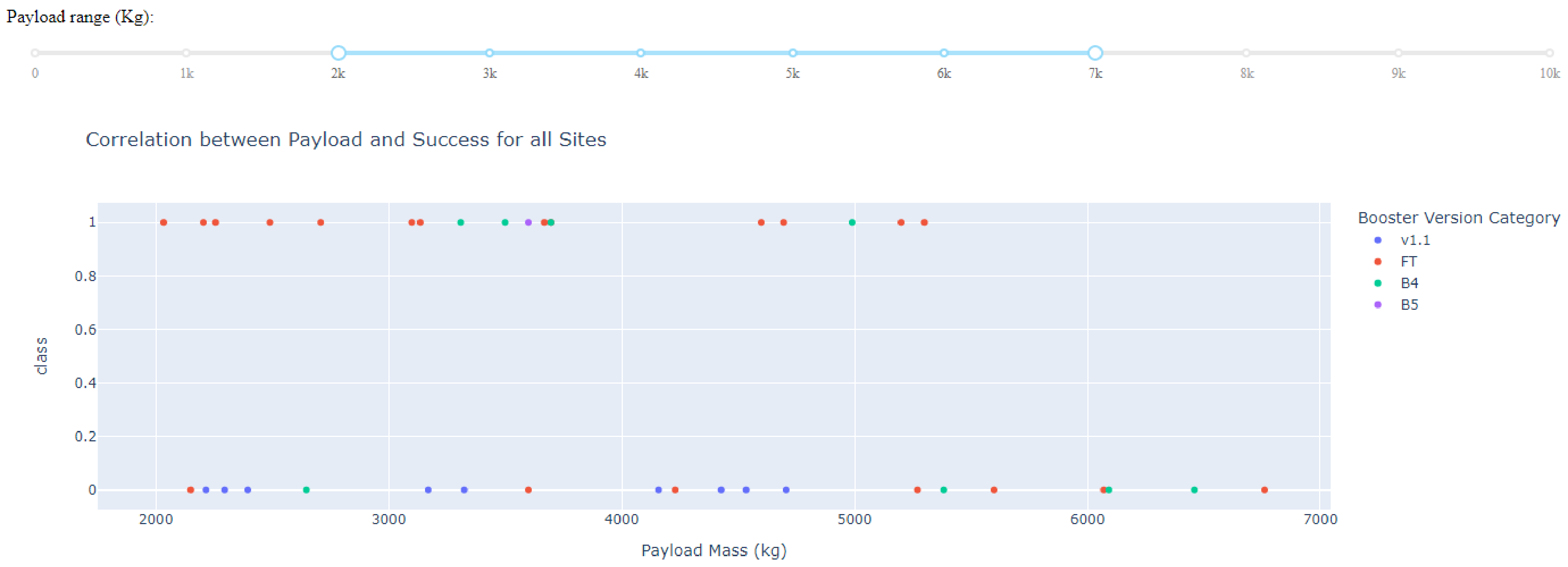


KSC LC-39A Launches

- KSC LC-39A has 76.9% successful launch (10 out of 13 launches)
- KSC LC-39A has only 23.1% successful launch (3 out of 13 launches)



Correlation between Payload and Success for all Sites



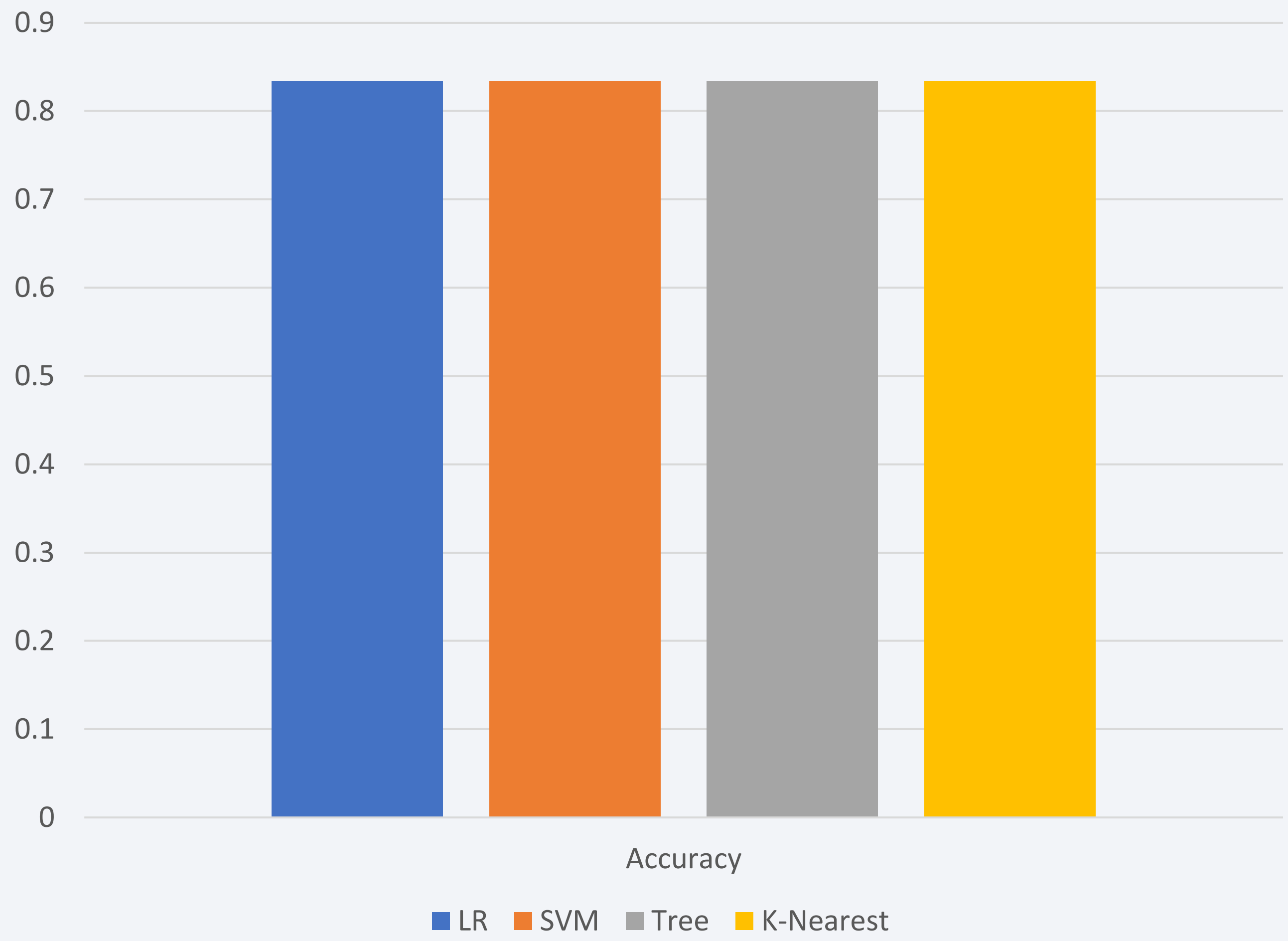
- Payload between 2000 to 5500 has highest success rate
- FT booster has highest success rate out of all booster versions

Section 5

Predictive Analysis (Classification)

Classification Accuracy

- All 4 classification models show the same accuracy score after using GridSearchCV to determine the best parameters
- While decision tree has the highest accuracy score 0.875 with the training dataset



Confusion Matrix

- This model performs the best at predicting True Positive
- While it did predict all 3 True Negative correctly, but given the small number of prediction (only 3), we can't make much conclusion from it
- This model also performs worst in False Positive, there are 3 cases where it predicted to be successful launch but in fact they were not

Conclusions

- The best suited classification model in addressing this problem is decision tree model with accuracy as high as 83.3333%

EXTRA (Innovation Section)

- I was trying to combine all 4 models together since all 4 models has 83.3333% of accuracy score with the test dataset, so I can't actually judge easily which is the best model to use, so why not use all and see if it actually gives any improvement?
- The concept is if 3 out of the 4 models predicted the same outcome, then we will take that outcome as the predicted outcome
- If only 2 out of the 4 models predicted the same outcome, then we will take it as successful launch prediction
- So I did a test on getting all yhat from the test_x datasets and compare all 4 yhat together and see if they are different from each other, and only if they differ from each other, this approach is doable
- It turns out that all 4 yhats equals each other, that means all 4 models predicted exactly the same using the test_x dataset
- This renders me no point continue trying to combine all 4 models into one because the combined model would have given the exact same predictions as any one of the model

Appendix

- Github Repository: https://github.com/pi31416chan/Coursera-Applied-Data-Science-Capstone

Thank you!