IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Chun Tien Shia
March 16, 2024

Outline

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

Executive Summary

- Summary of methodologies
  - Data were downloaded via spaceX.com API and also via web-scraping. The raw data were converted to tables and processed by patching meaningful values to empty spaces,. Certain features were one-hot encoded to facilitate analysis at a later stage. Exploratory data analysis was carried out using scatter, bar, pie and line charts, as well as SQL queries after converting it a SQLLite database. Interactive visual analytics were done by creating a folium map object, which was later marked with all launch sites, and labeled with success/failures at each site. Distances for selected points of interest near a site were calculated using coordinates. A online dashboard was created using Plotly Dash for analyzing success/failure ratios across all and selected sites models; a scatter plot was used to study the correlations between payload, booster version and success rates. Finally, predictive models using Logistic Regression, Support Vector Machine, Decision Tree and k-nearest neighbor methods with training data. The models were later test-fitted for accuracies using testing data sets.
- Summary of results
  - It is found that launch sites need to be located near seaside for maximum safety concern, and railway and highway for efficient transportation, KSC LC-40 39A site has best launch success rate among all sites, but different booster categories have differing success rates with different payloads,. Orbit types ES-L1, GEO, HEO and SSO have great success rates. The predictive model using decision tree model yielded the best prediction outcomes than the other three, with an accuracy of 0.8607

Introduction

- Objective of Project

  The objective of this project is to establish an analytical framework that can be used to help determine how much SpaceY should charge for launches to avoid potential losses.

- Context of Analysis

  - The analysis was confined to counting successful landings of rockets' first stage, which is the primary cost-deciding factor. The remaining journeys of the flights were ignored.
  - The data used for the analysis came from real historical launches.
  - The analysis was founded entirely on data science methodology.

- Problems and Challenges

  To achieve convincing results, the following questions need to be answered::

  - what methodology should be employed?
  - where and how reliable SpaceX launch data may be acquired?
  - how should be the results be visualized and interpreted?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was downloaded from spacex.com, and parsed to proper format with an API.
  - Another set of data also came from Wikipedia.com by web-scraping.
- Perform data wrangling
  - Empty values were filled appropriately, data changed to proper types, and some features on-hot encoded
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Data were divided into training and testing sets: Training set used to build the best-fit models of Logistic Regression, Support Vector Machine, Decision Tree Classification, and k-Nearest Neighbor. Testing set then used to decide the best model for prediction.

Data Collection

Avenues of Data Collections

Request JSON from SpaceX via API
(Convert JSON to Lists to Dictionary to Data Frame)

1. Import modules requests, pandas, numpy & datetime
2. Define list-building functions to extract desired values from composite values
3. Request to spacex.com for the launch file in json format
4. Normalize downloaded json to data frame containing composite values
5. Filter data frame to retain desired categories; convert to proper data types
6. Build final dictionary of lists from list-building functions
7. Convert dictionary to data frame
8. Filter data frame again for specific booster version
9. Renumber Flight# into running sequence and save data in csv format

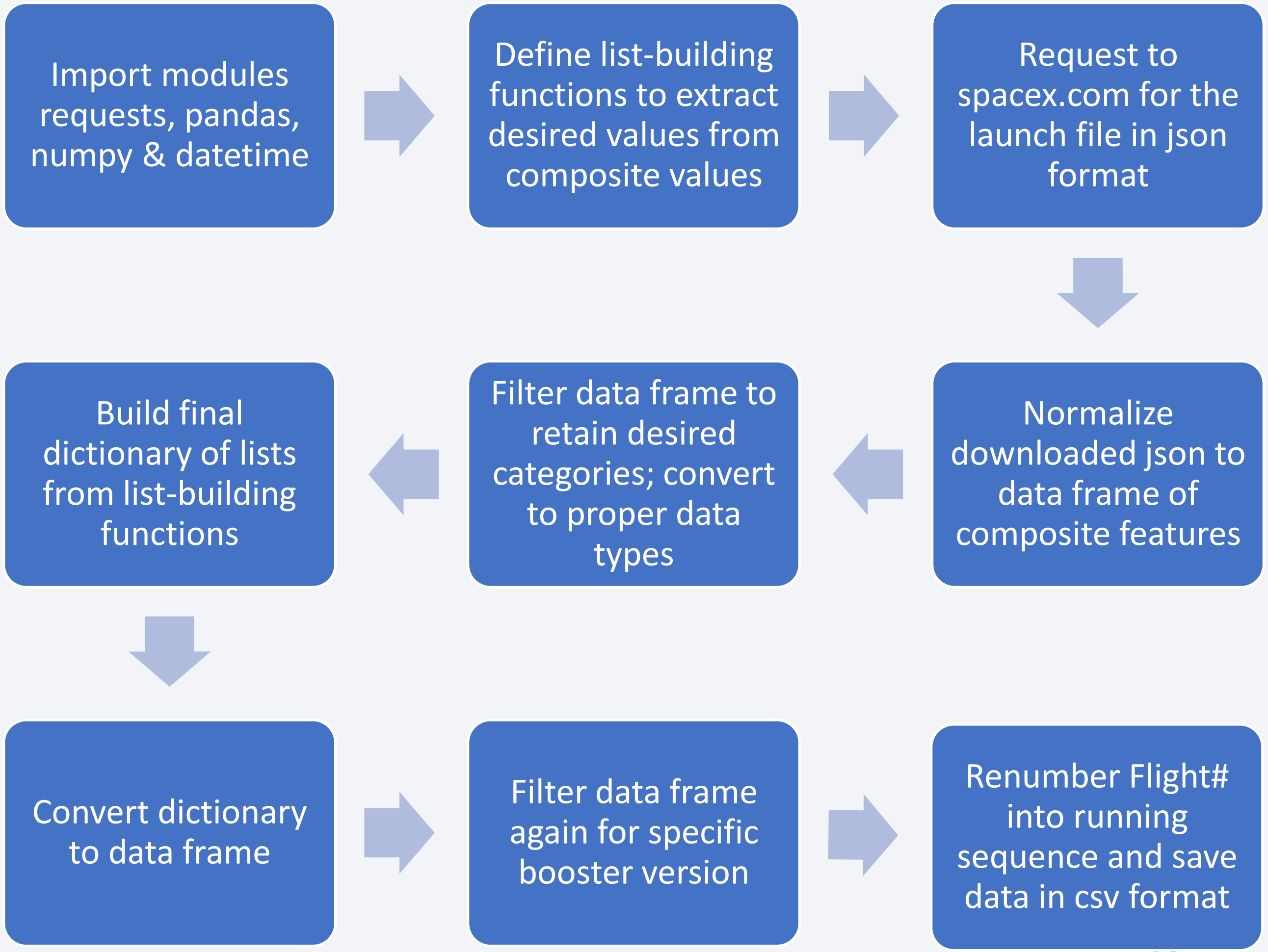
Web-scraping from Wikipedia
(Convert HTML to Data Frame)

1. Import modules sys, requests, pandas, re, BeautifulSoup & unicodeddata
2. Define helper functions to extract relevant values from HTML table cells
3. 'GET' all HTML tables from Wikipedia's webpage
4. Select the required table from downloaded tables
5. Extract column names from table by scanning <th>
6. Initialize an empty dictionary using column names as keys
7. Fill in values in dictionary by iterating through <tr> and <td> by parsing HTML table
8. Convert dictionary to data frame and saved in csv format

Data Collection – SpaceX API

- The process flow of downloading data from SpaceX using an API in JSON format, and converting it into lists, followed by dictionary, and finally into a data frame for further analysis, is shown on the right.
- The Github URL to the jupyter notebook for collecting data with API is given below:

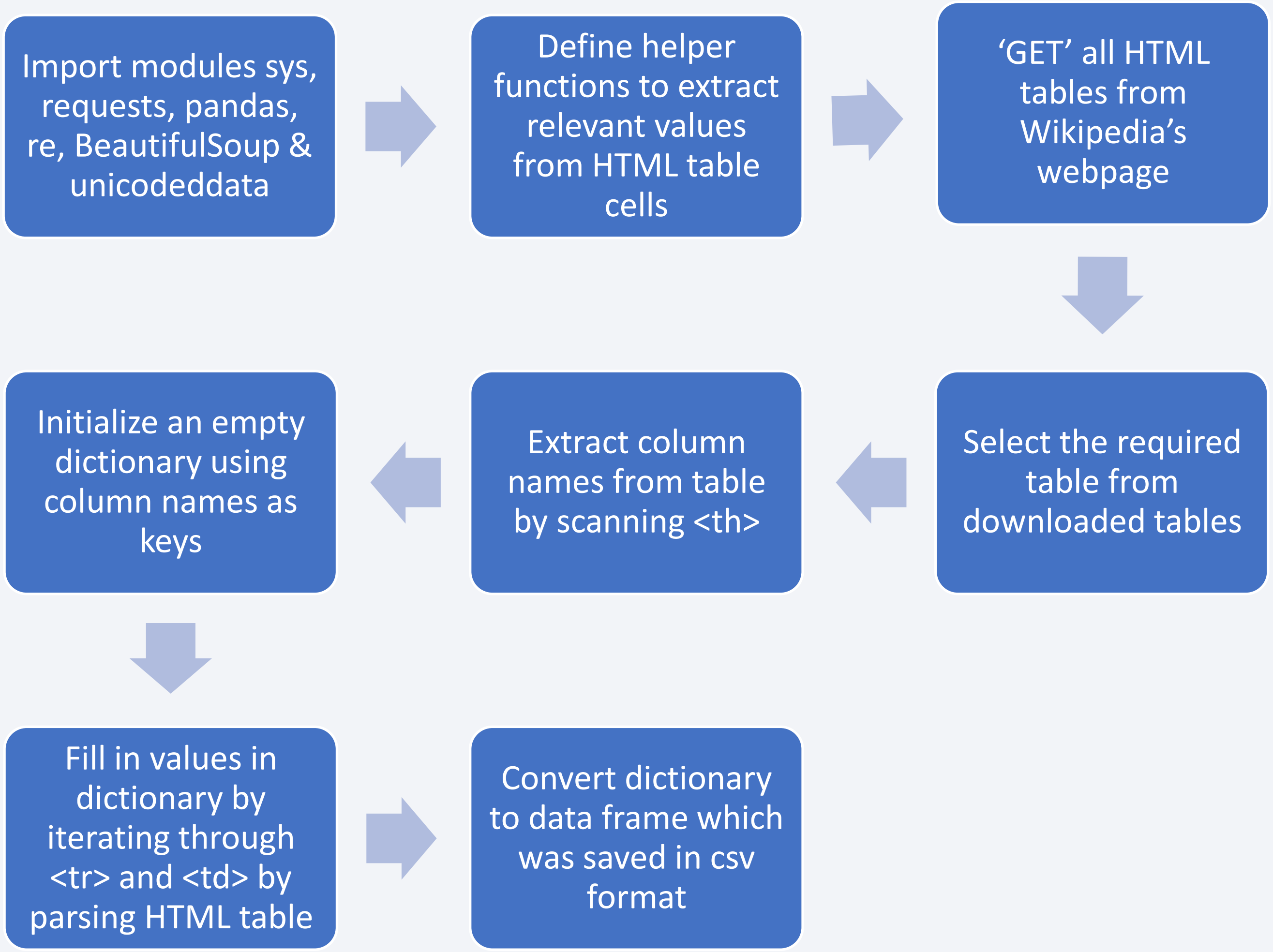
  https://github.com/C-Shia/IBM-Python-Capstone-Project/blob/aa4f145b74e3fb034994f49408354ed1c56b51b1/Week1/jupyter-labs-spacex-data-collection-api.ipynb



Data Collection - Scraping

- The process flow of web-scraping HTML tables from Wikipedia website, choosing the correct table, building dictionary, which was in turn converted into a data frame, is shown on the right
- The Github URL to the jupyter notebook for collecting data through web-scraping is given below:

https://github.com/C-Shia/IBM-Python-Capstone-Project/blob/aa4f145b74e3fb034994f49408354ed1c56b51b1/Week1/jupyter-labs-webscraping.ipynb

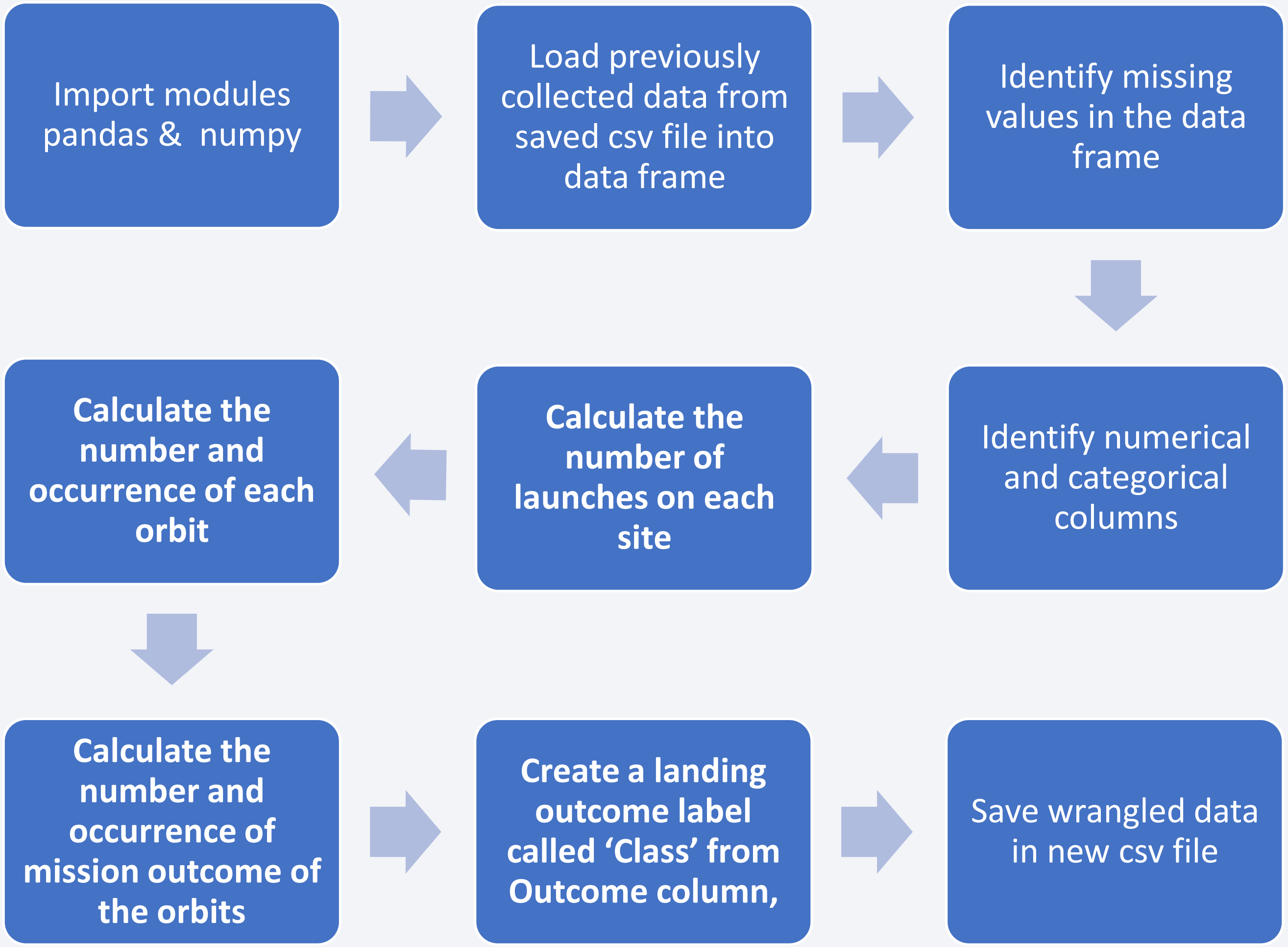


Data Wrangling

- The process flow of analyzing various data columns and eventually converting launch outcomes into a classification variable called ‘Class’, is shown on the. right

- The Github URL to the jupyter notebook for data wrangling is given below:

  https://github.com/C-Shia/IBM-Python-Capstone-Project/blob/aa4f145b74e3fb034994f49408354ed1c56b51b1/Week1/labs-jupyter-spacex-Data%20wrangling.ipynb



EDA with Data Visualization

- Table below summarizes the charts that were plotted and their purposes

Chart Type	X (Horizontal)	Y (Vertical)	Overlay (Color)	Purpose
Scatter	Flight Number	Payload	Launch Outcome	To view how payload and past experience (flight number) would influence the success of landing
Scatter	Flight Number	Launch Site	Launch Outcome	To view the different sites that were used to used for launching for different flights, and which sites/flights had successful landing
Scatter	Payload (kg)	Launch Site	Launch Outcome	To see the variation of different payloads launched at different sites, and which sites had more successes at which payloads
Bar	Orbit	Launch Outcome		To understand which target orbit types had better success landing results
Scatter	Fight Number	Orbit Type	Launch Outcome	To view the variation of orbit targeted at different flights, and how that correlated with the successful landing
Scatter	Payload (lg)	Orbit Type	Launch Outcome	To view how payload and orbit types would influence the landing success or failure
Line	Year	Launch Outcome		To see how as launch experience accumulated over the years may affect landing outcome

- The Github URL to the jupyter notebook for EDA with data visualization is given below:

  https://github.com/C-Shia/IBM-Python-Capstone-Project/blob/aa4f145b74e3fb034994f49408354ed1c56b51b1/Week2/jupyter-labs-eda-dataviz.ipynb

EDA with SQL

- A SQLlite database was created from the launch history data file. The queries made were:
  - Displayed the names of the unique launch sites in the space mission
  - Displayed 5 records where launch sites begin with the string 'CCA' ion historical order
  - Displayed the total payload mass carried by boosters launched by NASA (CRS)
  - Displayed average payload mass carried by booster version F9 v1.1
  - Listed the date when the first successful landing outcome in ground pad was achieved
  - Listed the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
  - Listed the total number of successful and failure mission outcomes
  - Listed the names of the booster versions which have carried the maximum payload mass
  - Listed the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015
  - Ranked 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
- The Github URL to the jupyter notebook for EDA with SQL is given below:

  https://github.com/C-Shia/IBM-Python-Capstone-Project/blob/aa4f145b74e3fb034994f49408354ed1c56b51b1/Week2/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- Table below summarizes the folium map created on launch sites and the objects added:

Folium Object	Purpose
Map	A map object was created to display the US mainland
Circle	Circles centered at the coordinates of each launch site to depict the locality
Marker	Markers were created to show the launch outcome at each site. Marker were also used to denote the distance of 2 points on the map
MarkerCluster	Marker clusters were created to group individual launch outcome markers at each launch site into one object, so these individual markers were displayed together
MousePosition	These were created to display the coordinates in terms of latitude and longitude of a point on the map. The coordinates were then used to calculate straight distances between 2 points
PolyLine	Line objects were used to display the linear distance between a launch site and its proximities in a visual manner

- The Github URL to the jupyter notebook containing the interactive folium map is given below:

  https://github.com/C-Shia/IBM-Python-Capstone-Project/blob/aa4f145b74e3fb034994f49408354ed1c56b51b1/Week3/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

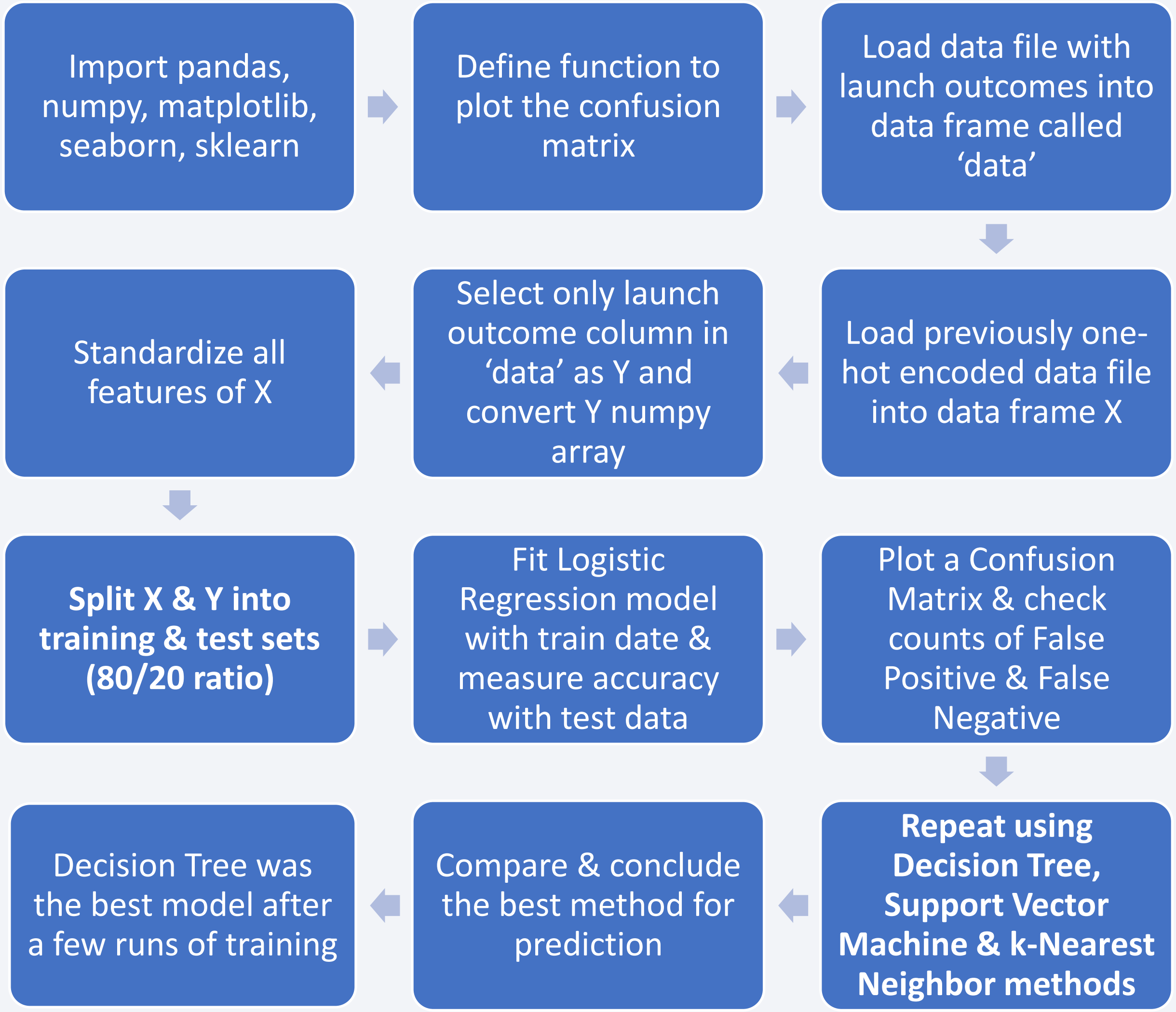
- Table below summarizes the objects added to a Plotly Dashboard and their purposes:

Object	Purpose	Github URL
Dash App Layout with Title	To setup the dashboard's layout and describe its purpose	**Image of Completed Dashboard:** https://github.com/C-Shia/IBM-Python-Capstone-Project/blob/aa4f145b74e3fb034994f49408354ed1c56b51b1/Week3/Completed%20Dash%20Board%20Application.png
Dropdown Box	To allow the user to pick all launch sites or a single site	**Images of All Sites selected:** https://github.com/C-Shia/IBM-Python-Capstone-Project/blob/aa4f145b74e3fb034994f49408354ed1c56b51b1/Week3/Task%202%20Call%20Back%20To%20Update%20Pie%20Chart%20On%20All%20Sites.png **Image of Selected Site (CCAFS LC-40):** https://github.com/C-Shia/IBM-Python-Capstone-Project/blob/aa4f145b74e3fb034994f49408354ed1c56b51b1/Week3/Task%202%20Call%20Back%20To%20Update%20Pie%20Chart%20On%20Selected%20Site.png
Pie Chart	To show the total successful launches count for all sites or ratio of success and failures if a single site was picked	**Image of Pie chart displayed for All Sites**: https://github.com/C-Shia/IBM-Python-Capstone-Project/blob/aa4f145b74e3fb034994f49408354ed1c56b51b1/Week3/Task%202%20Call%20Back%20To%20Update%20Pie%20Chart%20On%20All%20Sites.png **Image of Pie chart displayed for Selected Site (CCFAS LC-40)**: https://github.com/C-Shia/IBM-Python-Capstone-Project/blob/aa4f145b74e3fb034994f49408354ed1c56b51b1/Week3/Task%202%20Call%20Back%20To%20Update%20Pie%20Chart%20On%20Selected%20Site.png
Range Slider	To allow the user to set minimum and maximum payloads of launches	**Image of Range Slider**: https://github.com/C-Shia/IBM-Python-Capstone-Project/blob/aa4f145b74e3fb034994f49408354ed1c56b51b1/Week3/Task%203%20Payload%20RangeSlider.png
Scatter Chart	to show the correlation between payload and launch success	**Image of Scatter displayed for All Sites**: https://github.com/C-Shia/IBM-Python-Capstone-Project/blob/aa4f145b74e3fb034994f49408354ed1c56b51b1/Week3/Task%204%20Scattered%20Plot%20For%20All%20Sites.png **Image of Scatter displayed for Selected Site (CCAFS LC-40)**: https://github.com/C-Shia/IBM-Python-Capstone-Project/blob/aa4f145b74e3fb034994f49408354ed1c56b51b1/Week3/Task%204%20Scattered%20Plot%20For%20Selected%20Site.png

- Github URL of complete python file: https://github.com/C-Shia/IBM-Python-Capstone-Project/blob/aa4f145b74e3fb034994f49408354ed1c56b51b1/Week3/spacex_dash_app.py

Predictive Analysis (Classification)

- The process flow to train and test best-fit models using Logistic Regression, Decision Tree, Support Vector Machine, k-Nearest Neighbor methods is shown on the right. The Decision Tree method gave the best predictive model after repeated trainings.
- The Github URL to the jupyter notebook for predictive anaysis is given below: https://github.com/C-Shia/IBM-Python-Capstone-Project/blob/aa4f145b74e3fb034994f49408354ed1c56b51b1/Week4/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb



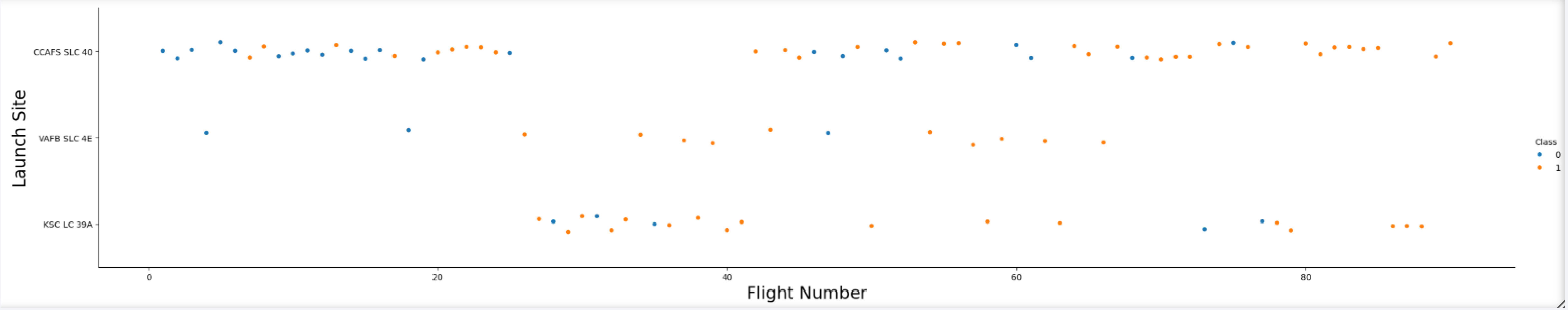
Results

- Exploratory data analysis results – to be discussed in
  - Section 2 – Insights Drawn From EDA
- Interactive analytics demo in screenshots – to be discussed in
  - Section 3 – Launch Site Proximity Analysis
  - Section 4 – Build A Dah With Plotly Dash
- Predictive analysis results – to be discussed in
  - Section 5 – Predictive Analysis (Classification)

Section 2

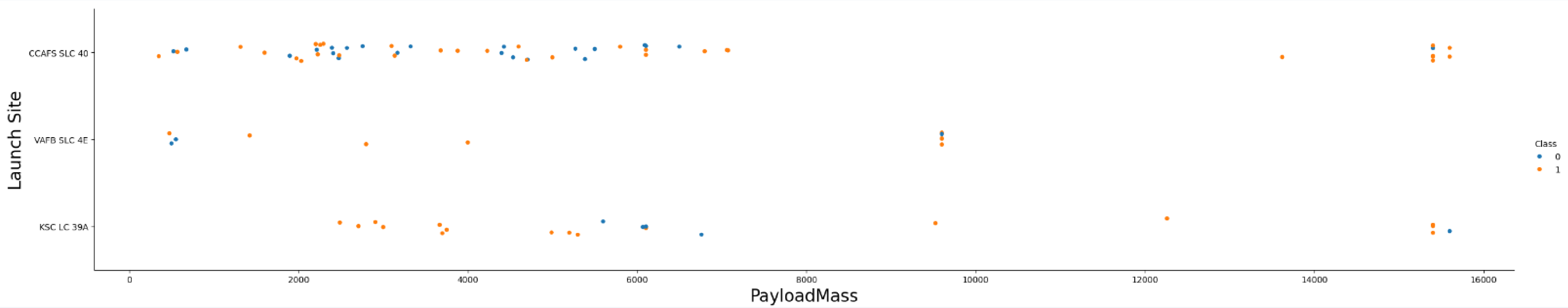
Insights drawn from EDA

Flight Number vs. Launch Site



- It can be seen from the above plot that prior to the 80th flight, the successes and failures were mixed for the three launch sites, CCFAS SLC-40, WFB SLC-4E and KSC LC-39A. After that, the launches at CCAFS SLC-40 and KSC LC-39 A were all successful.

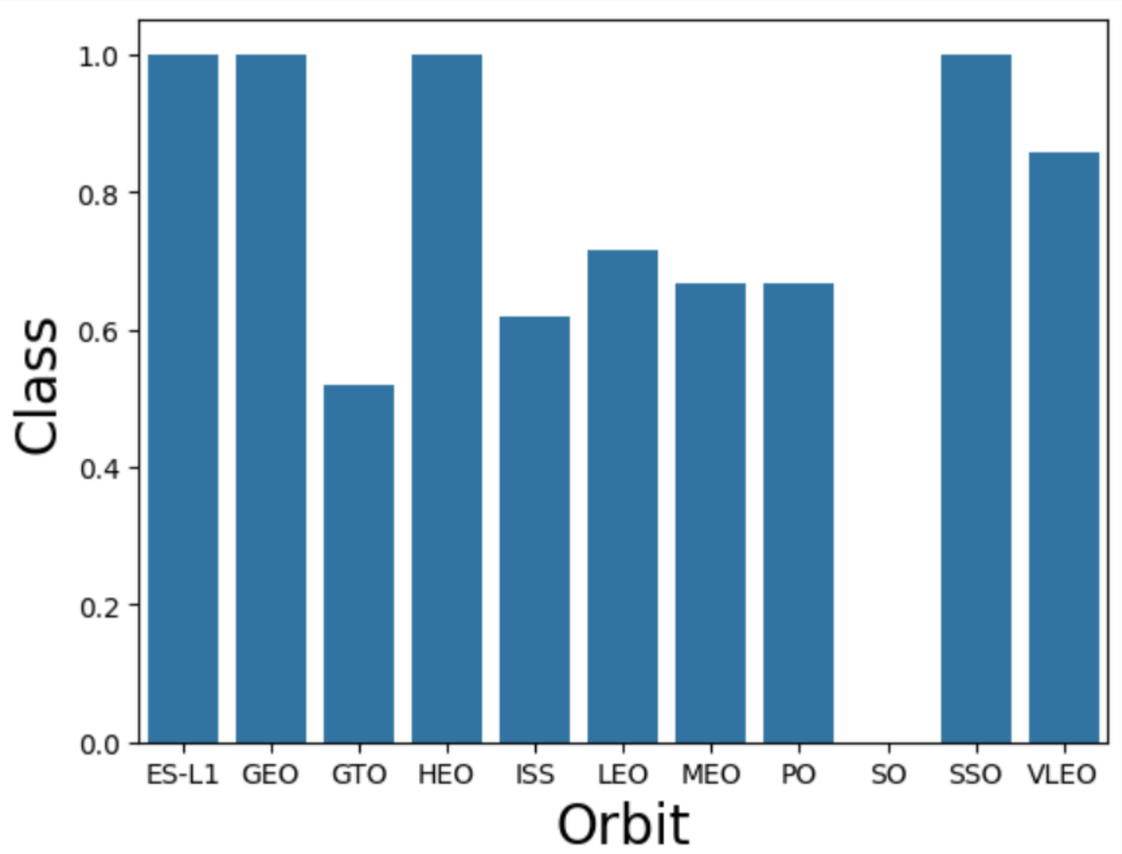
Payload vs. Launch Site



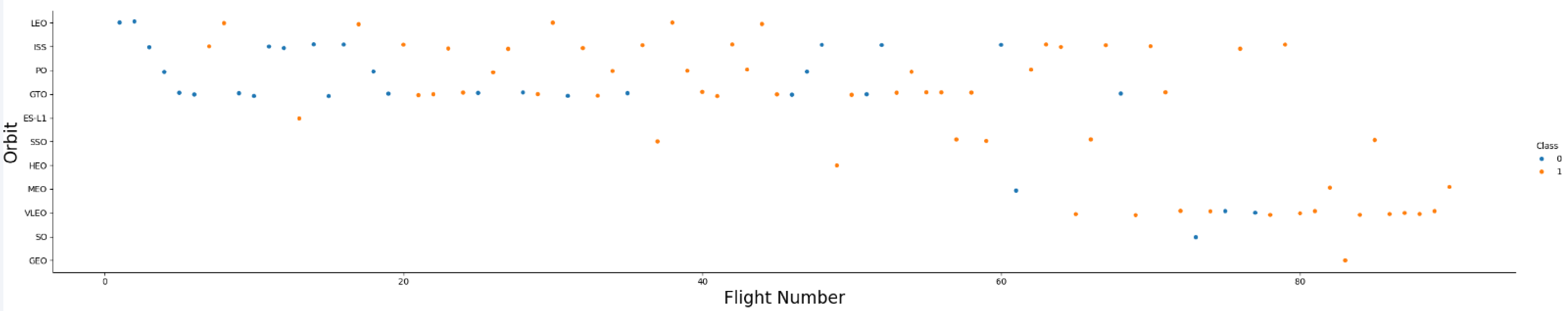
- It can be seen from the above scatter plot that at the VAFB-SLC launch site there were no rockets launched for heavy payload mass greater than 10,000kg. Payloads greater than 10,000kg did not guarantee success whereas those less than 5,000kg from the KSC LC-39A site were all successful.

Success Rate vs. Orbit Type

- From the bar chart on the right, no successes were recorded for orbit type SO, but launches for the ES-L1, GEO, HEO and SSO orbits were all successful.

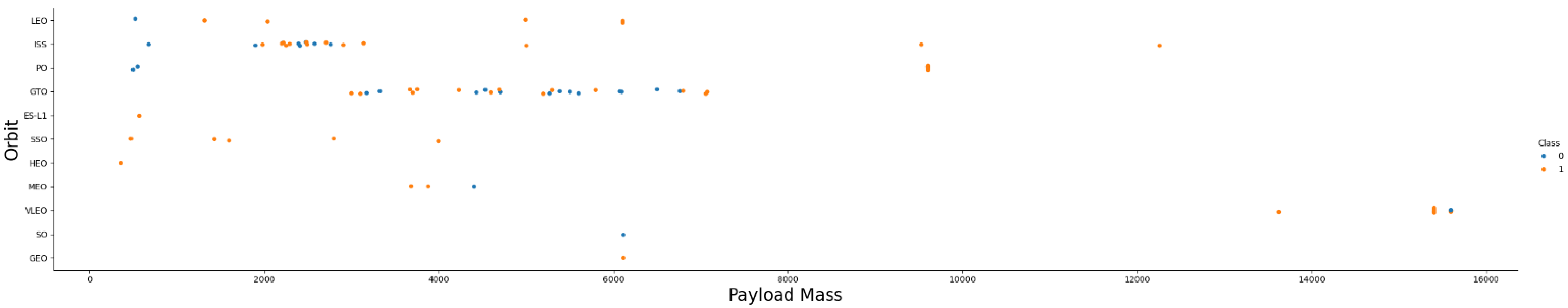


Flight Number vs. Orbit Type



- The above scatter plot shows that for the LEO orbit the success appears to be related to the number of flights; on the other hand, there seems to be no relationship between flight number for the GTO and VLEO orbits.

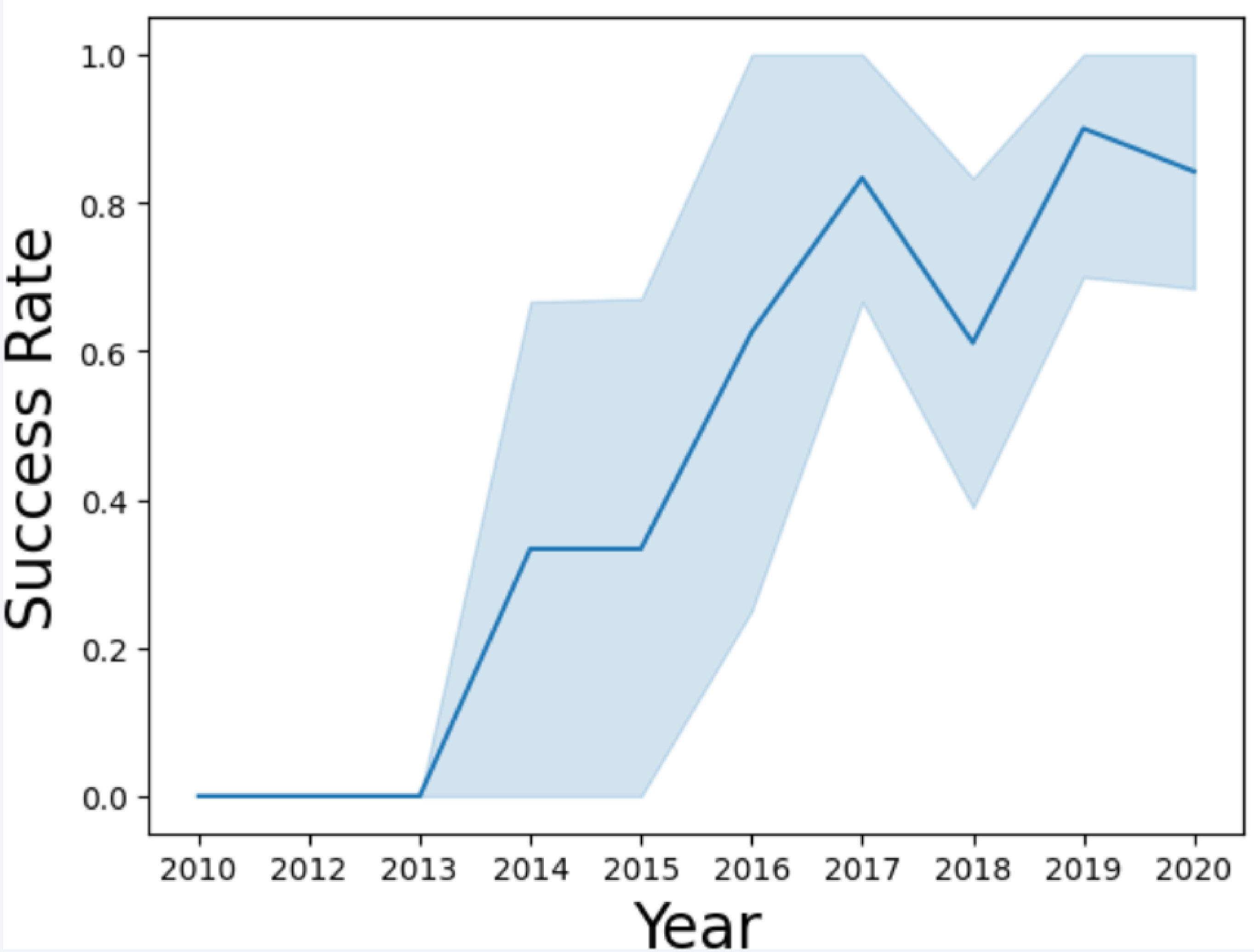
Payload vs. Orbit Type



- The above scatter seems to suggest that With heavy payloads the successful landing or positive landing rate are more for Polar, LEO and ISS, whereas for GTO, both positive landing rate and negative landing (unsuccessful mission) are both there here.

Launch Success Yearly Trend

- The line chart on the right shows that the success rate since 2013 kept increasing till 2017 (stable in 2014) and after 2015 it started increasing.



All Launch Site Names

- The names of the launch sites found in the data file were found to be:

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

- "Select Distinct" was used to display only unique site names from the table;

%sql select distinct "Launch_Site" from SPACEXTBL

Launch Site Names Begin with 'CCA'

- Below table shows 5 records where launch sites begin with `CCA` in chronological order of the launches:

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	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	0:35:00	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

- Records were ordered by date, and the output was limited to 5 records only:

/ *%sql select * from SPACEXTBL where "Launch_Site" like '%CCA%' order by "Date" limit 5*

Total Payload Mass

- The total payload in kg carried by boosters from NASA is given below:

TotalPayloadMass
45596

- The result was derived by filtering records from NASA and summing all payloads:

%sql select sum("PAYLOAD_MASS__KG_") as TotalPayloadMass from SPACEXTBL where "Customer"='NASA (CRS)'

Average Payload Mass by F9 v1.1

- The average payload mass in kilograms carried by booster version F9 v1.1 is shown below:

AveragePayloadMass
2534.666666666665

- Records in the table were filtered to F9 v1.1, and then the average of the payload was calculated and output:

%sql select avg("PAYLOAD_MASS__KG_") as AveragePayloadMass from SPACEXTBL where "Booster_Version" like '%F9 v1.1%'

First Successful Ground Landing Date

- The dates of the first successful landing outcome on ground pad was on the launch on June 4, 2010, as show:

FirstSuccessfulLaunchDate
2010-06-04

- Result was acquired by filtering the records to only successful mission outcome, and then find the smallest (earliest) date:

%sql select min(Date) as FirstSuccessfulLaunchDate from SPACEXTBL where "Mission_Outcome" like '%Success%'

Successful Drone Ship Landing with Payload between 4000 and 6000

- Below lists the names of boosters which had successfully landed on drone ship and had payload mass greater than 4000 but less than 6000

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

- Records were filtered to launches that carried payloads within the given range and also successfully landed on drone ship, and then displayed the booster version column:

%sql select "Booster_Version" as BoosterVersion|from SPACEXTBL where " Landing_Outcome" like '%Success%drone ship%' and ("PAYLOAD_MASS__KG_">4000 and "PAYLOAD_MASS__KG_"<6000)

Total Number of Successful and Failure Mission Outcomes

- The total number of successful and failure mission outcomes is given below:

TotalSuccessandFailure
101

- Records were filtered to contain either successful or failed outcomes, and then display the count of the filtered records:

%sql select count("Mission_Outcome") as TotalSuccessandFailure from SPACEXTBL where "Mission_Outcome" like '%Success%' or "Mission_Outcome" like '%Failure%'

Boosters Carried Maximum Payload

- The names of the booster which have carried the maximum payload mass are shown to the right.
- A subquery was first executed to find the maximum payload, and then a search to find the boosters that carried this maximum payload:

%sql select "Booster_Version“ from SPACEXTBL where "PAYLOAD_MASS__KG_" in (select max("PAYLOAD_MASS__KG_") from SPACEXTBL)

Booster_Version
F9 B5 B1048.4
F9 B5 B1049.4
F9 B5 B1051.3
F9 B5 B1056.4
F9 B5 B1048.5
F9 B5 B1051.4
F9 B5 B1049.5
F9 B5 B1060.2
F9 B5 B1058.3
F9 B5 B1051.6
F9 B5 B1060.3
F9 B5 B1049.7

2015 Launch Records

- The failed landing outcomes in drone ship, their booster versions, and launch site names for in year 2015 are given below, once in January and another in April:

Year	Month	LandingOutcomeInDroneShip	BoosterVersion	LaunchSite
2015	Jan	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
2015	Apr	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

- Records were filtered to find all failed landing on drone ship which happened in 20145 and then displayed the month names, outcomes, boosters and site names:

```
%sql select substr("Date",0,5) as Year,
substr('JanFebMarAprMayJunJulAugSepOctNovDec',substr("Date",6,2)*3-2,3) as Month,
"Landing_Outcome" as LandingOutcomeInDroneShip, "Booster_Version" as BoosterVersion,
"Launch_Site" as LaunchSite|from SPACEXTBL where|substr("Date",0,5)='2015' and
"Landing_Outcome" like '%Fail%Drone%'
```

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

- The ranked counts 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, are show to the right.
- Records were filtered for the required date range, and grouped by landing outcomes. The landing outcomes were then counted and listed in descending order. Displayed both the outcomes and their counts:

```
%sql select "Landing_Outcome", count("Landing_Outcome") as Count from SPACEXTBL where ("Date">='2010-06-04' and "Date"<='2017-03-20') group by "Landing_Outcome" order by count("Landing_Outcome") desc
```

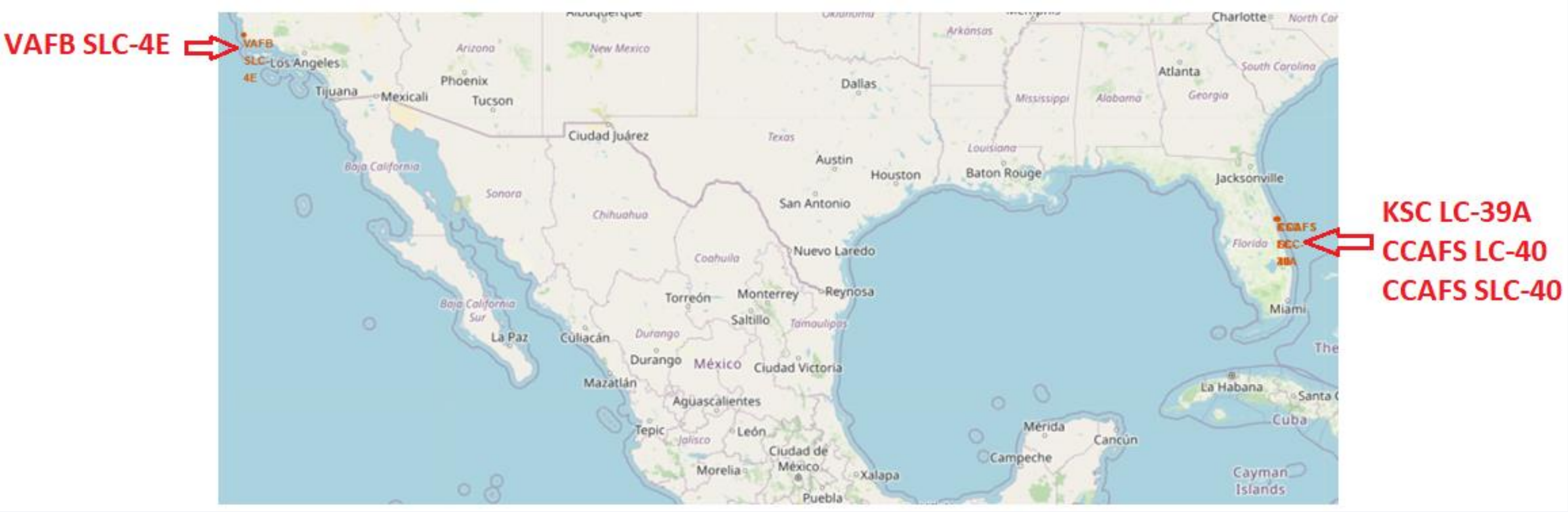
Landing_Outcome	Count
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

Section 3

Launch Sites Proximities Analysis

Locations of All Launch Sites

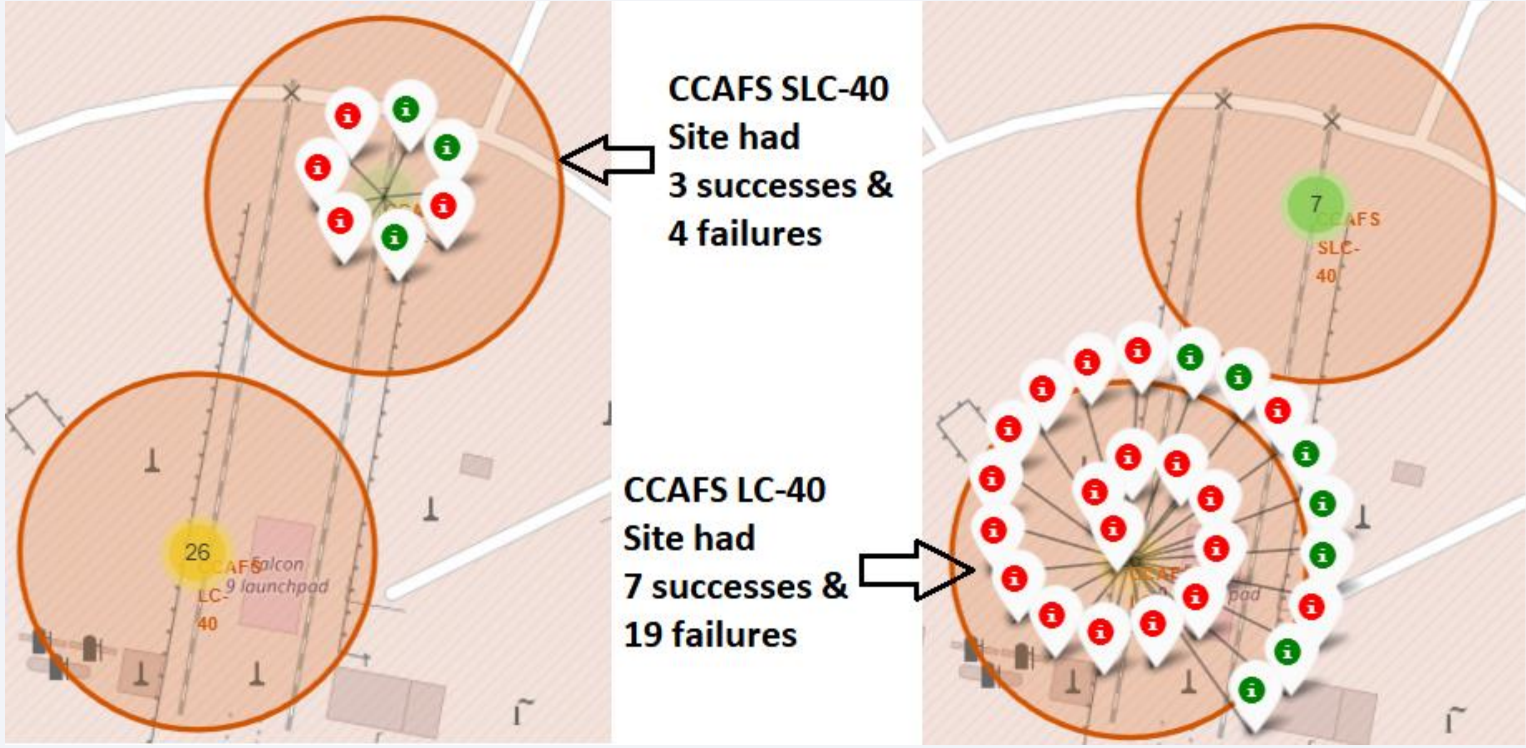
- A folium map of the US continent was created and overlayed with all launch sites:



- The map show the VAFB SLC-4E is located on the West Coast, and the KSC LC-39A, CCAFS LC-40 and CCFAS SLC-40 are located on the East Coast.

Launch Outcomes At Sites Shown In Color Labels

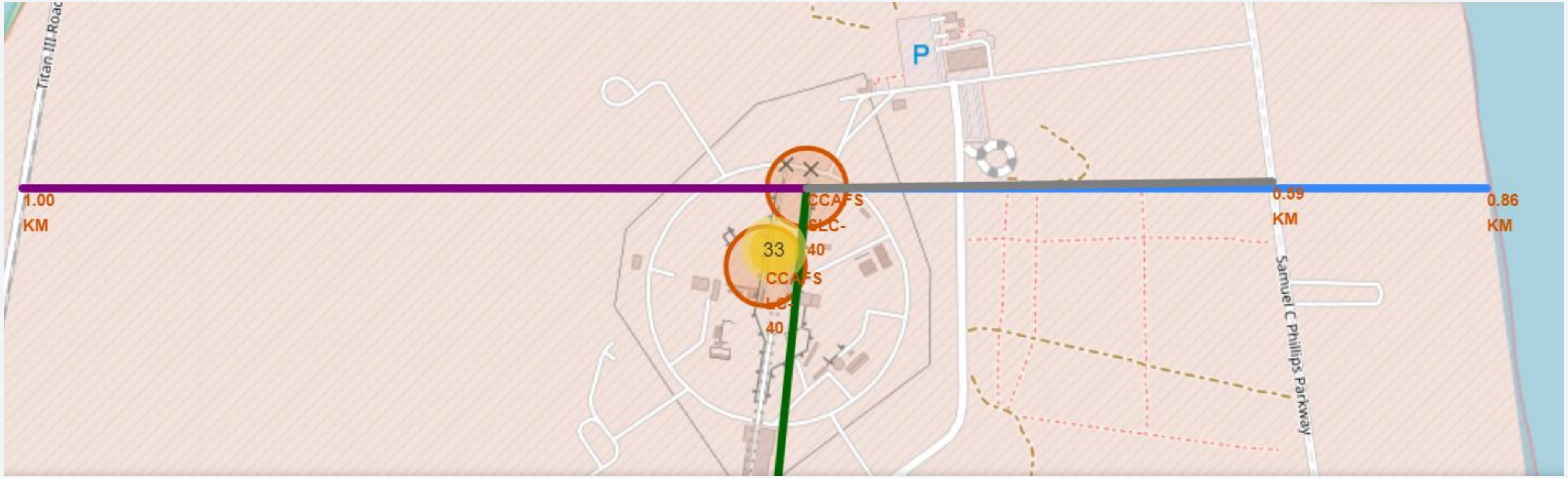
- The launch outcomes were added to each location using folium's MarkerCluster objects:



- The above examples shows 7 launches at CCAFS SLC-40 and 16 launches at CCAFS-40. The green labels are for successes and the red ones are for failures.

Proximity Of A Selected Site

- The screen shot below shows the proximity of the CCAFS SLC-40 site:

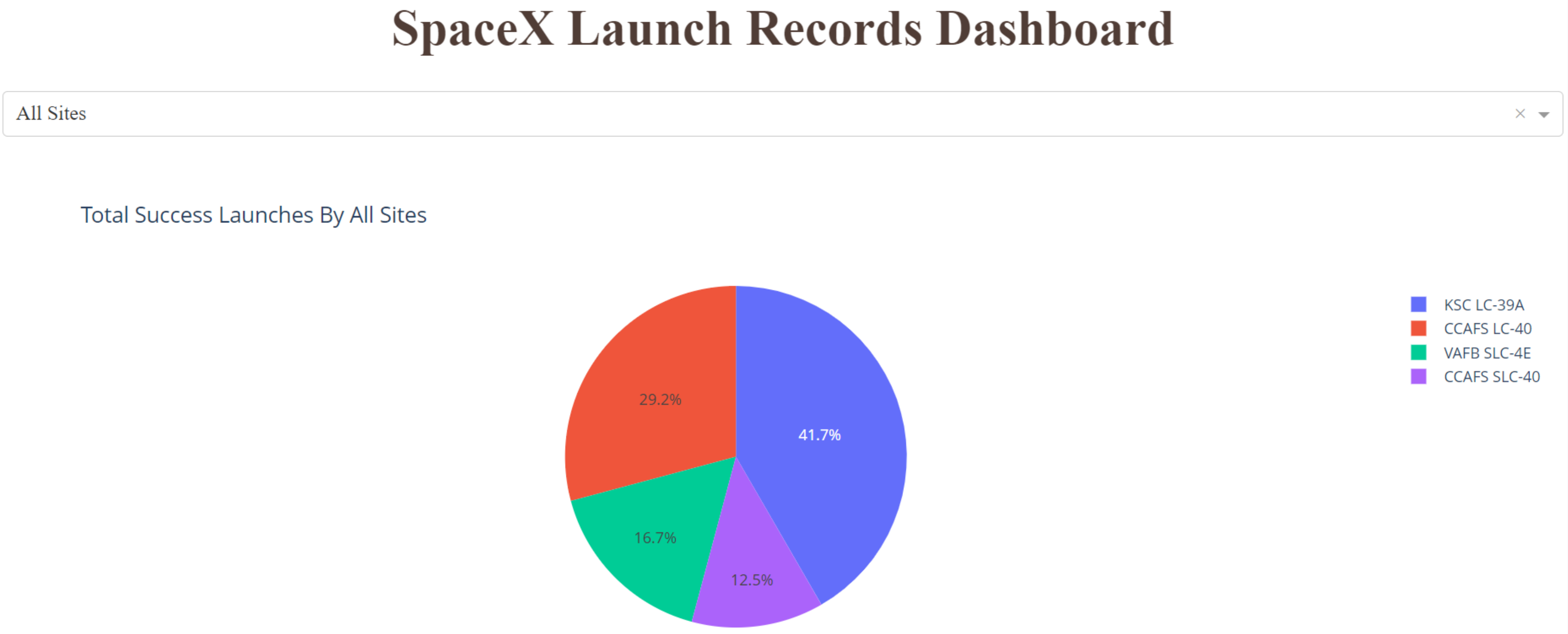


- Lines were drawn from the center of CCAFS SLC-50 to the nearest coastline, railway and highway. The distances can be seen in the picture at the end of the lines, all within 1 km. These short distances illustrate that a launch site should be located close to sea for safety, and closed to railway/highway for easy access.

Section 4

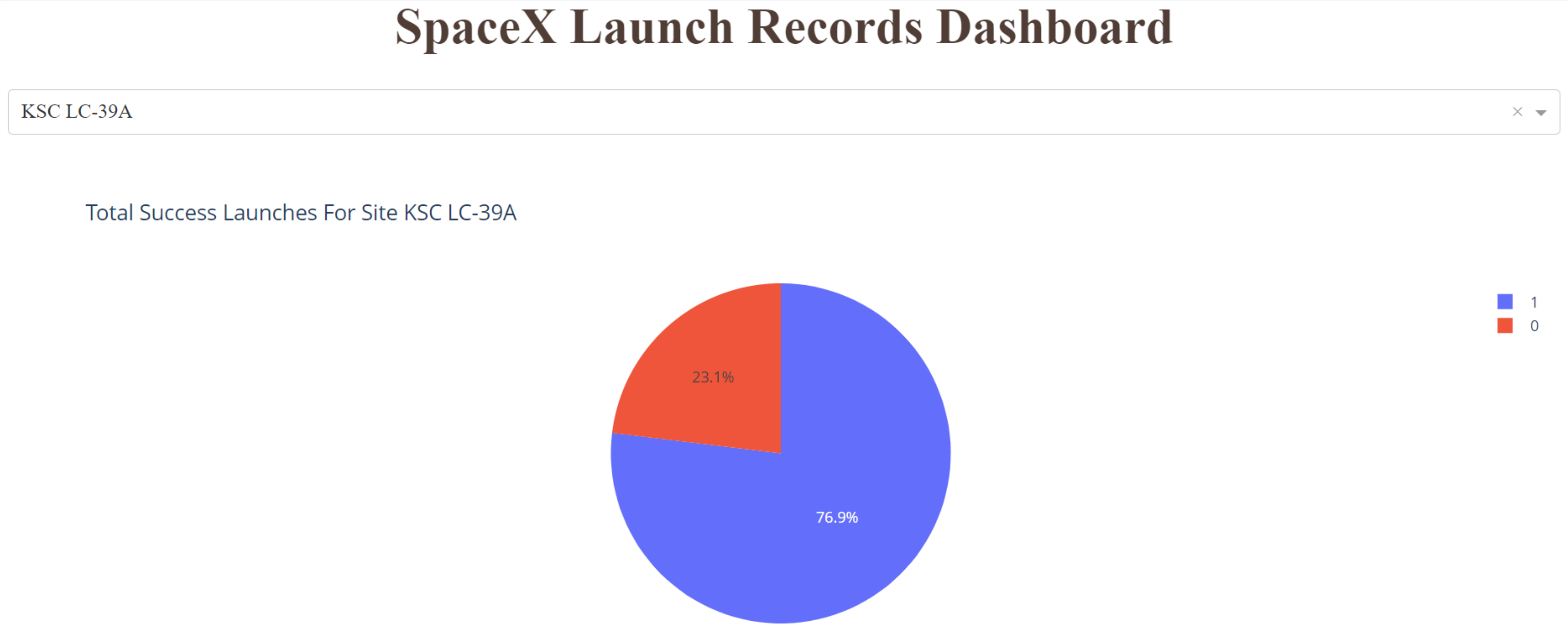
Build a Dashboard with Plotly Dash

Showing Launch Success Using Pie Chart



- The above pie chart shows the ratio of successes across all launch sites. Roughly 41.7% of successful launches came from the KSC LC-39A site.

Launch Site That Had The Highest Success Rate



- The above pie chart shows the KSC LC-39A site had the highest success rate, of almost 77%

Payload & Booster Category vs Launch Success

Booster type B4 had the best success between 3000 and 5000kg range. Type V1.1 had no records of success

Booster type FT and B4 had mixed successes and failures between 4000 and 6000kg.Type V1.1 had all failures

In the payload range of 5000 to 10000k, all booster types did not do well.

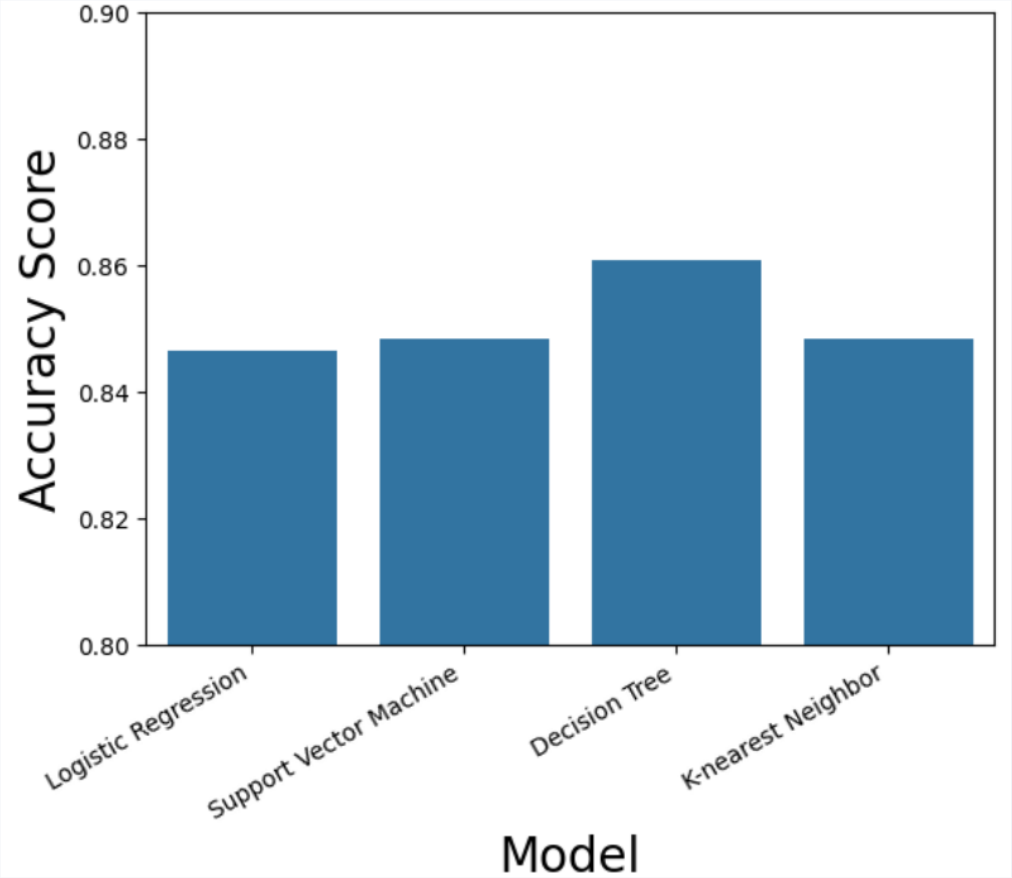
- Booster type FT fared the best for payloads below 3000kg. Type B5 had no record of failure

Section 5

Predictive Analysis (Classification)

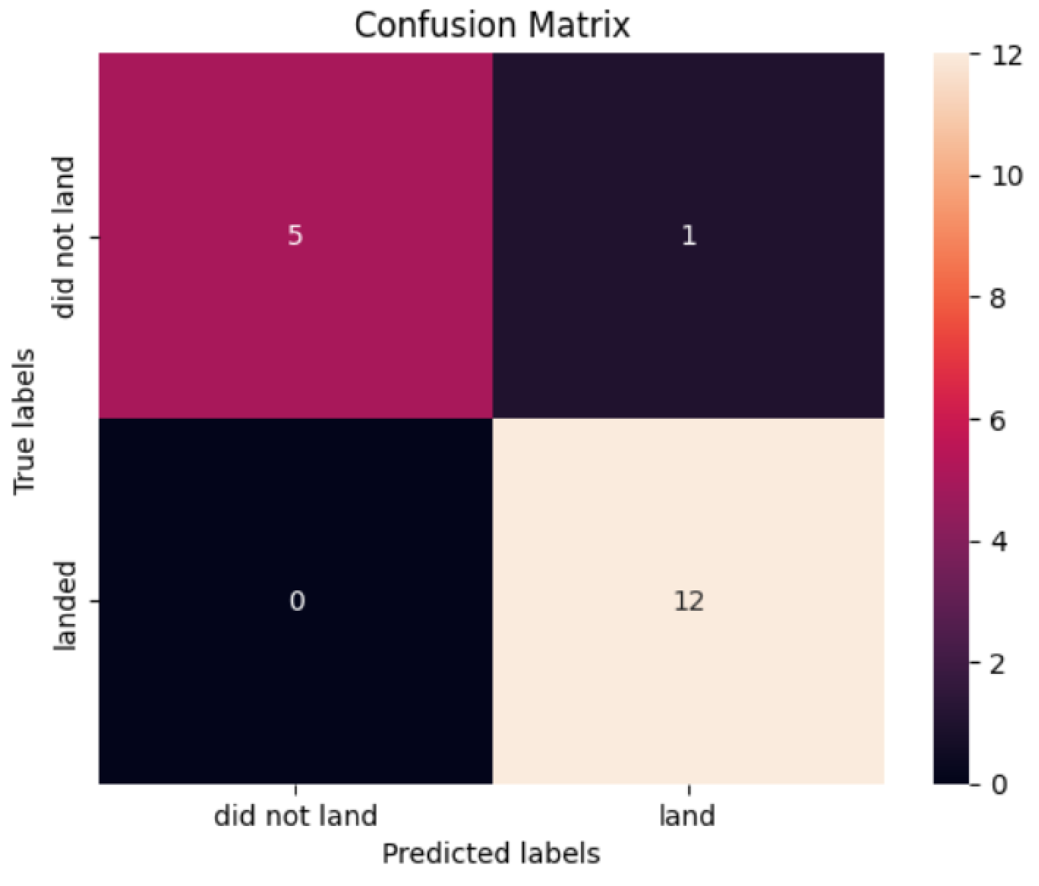
Classification Accuracy

- After performing training and test-fitting using logistic regression, support vector machine, decision tree and k-nearest neighbor, the models' fit accuracies are displayed on right in a bar chart.
- The best model was the decision tree model, after some trial-and-error adjustments of the parameters, and it was found have an accuracy of 0.8607.



Confusion Matrix

- The confusion matrix of the best model, ***decision tree***, is shown on the right.
- The criteria of assessment was based on the total count of false positive plus false negative. All other models (logistic regression, support vector machine and k-nearest neighbor) produced zero false negative and three false positives from the test data. The decision tree model yielded not only zero false negative, but down to one false positive.



Conclusions

- Two avenues of data collection were discussed, namely the spacex.com API, as well as the web-scraping methods.
- Data downloaded need to be processed and prepared for analysis.
- Exploratory data analysis (EDA) was performed through the use various visual tools, such as scatter, bar, pie and line charts. Launch successes were compared with different contributing features.
- A map was created and marked with circles to locate launch sites, color labels to display successes/failures at each site, and measurements of a site’s proximity were taken.
- Finally, predictive analysis was done using four different models and the best was found to be the decision tree model, with an accuracy of 0.8607.

Appendix

- Various python notebooks, scripts and screenshots for this project can be found in the public repository at the Github link below:

  https://github.com/C-Shia/IBM-Python-Capstone-Project/

Thank you!