DATA SCIENCE CAPSTONE PROJECT

Nattawat Tanalurkmongkol
https://github.com/job-nattawat
22/07/2023

AGENDA

Executive Summary

Introduction

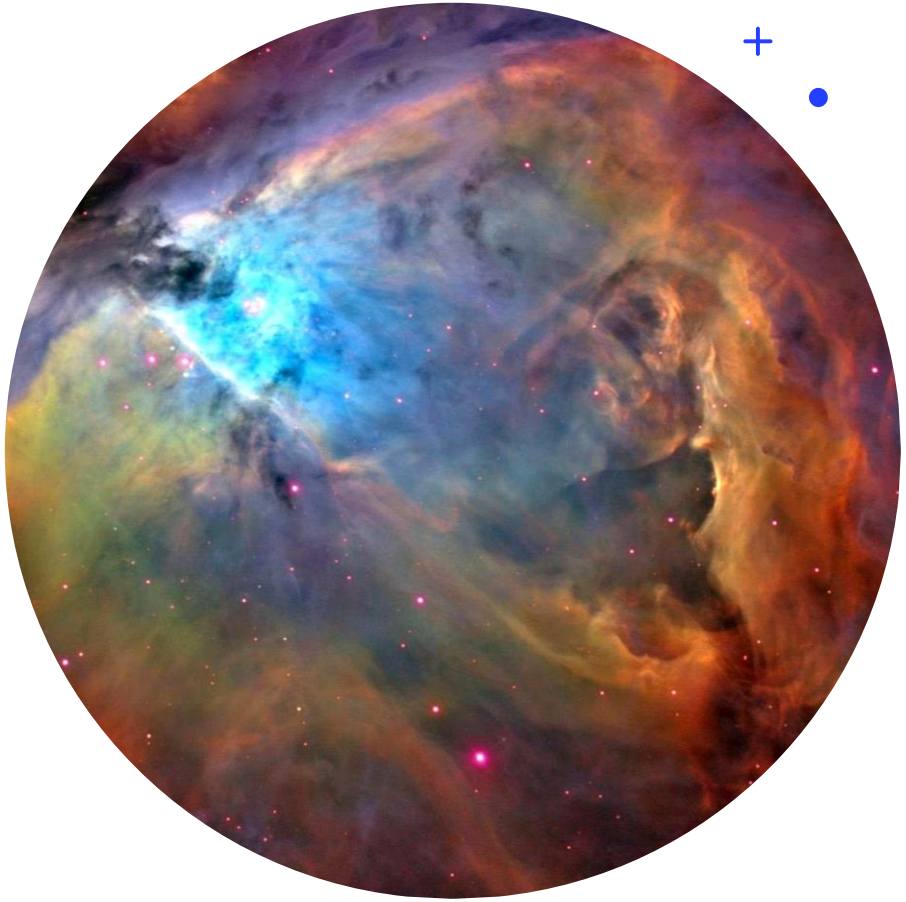
Methodology

Results

Conclusion

Appendix

Executive Summary



Conducted comprehensive data analysis on SpaceX's successful landings using a combination of data from the SpaceX Wikipedia page and open SpaceX API. Employed SQL, data visualization, folium maps, and dashboards to explore and preprocess the data. Utilized single hot encoding to convert categorical variables to binary format and standardized data for machine learning. Implemented Logistic Regression, Support Vector Machine, Decision Tree Classifier, and K Nearest Neighbors models, achieving an average accuracy rate of 83.33% for predicting successful landings. While all models showed promising results, further data augmentation is recommended to enhance model determination and accuracy.

INTRODUCTION

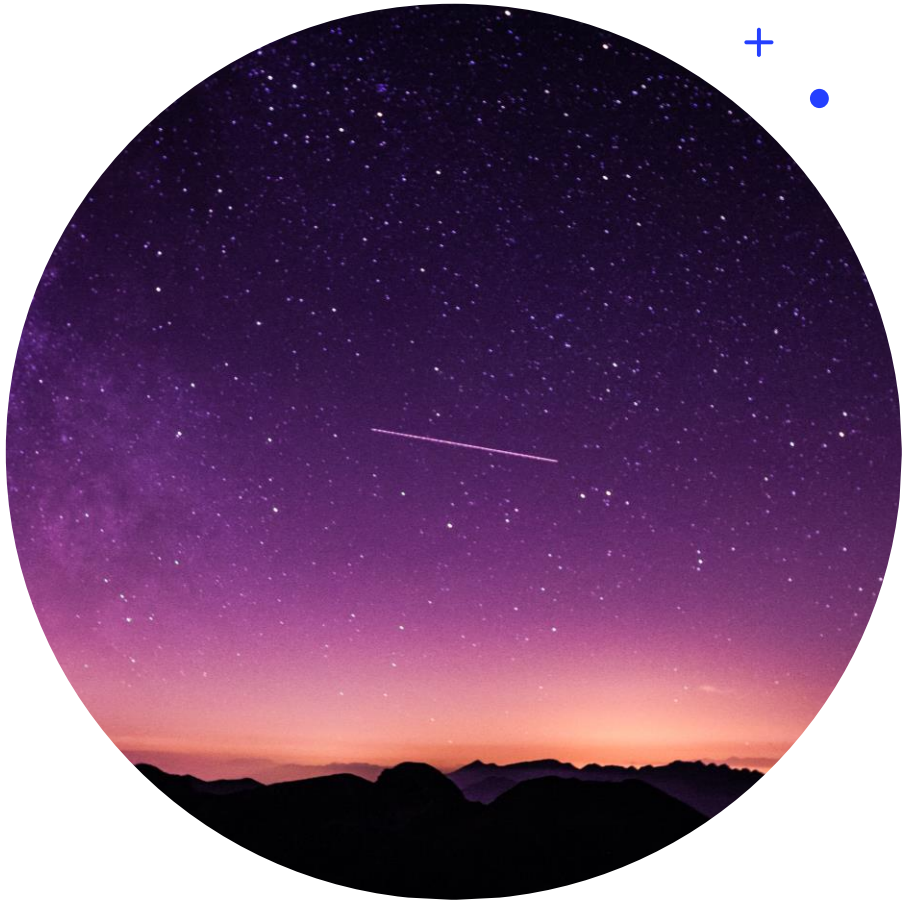
Background:

Commercial Space Age is Here Space X has best pricing ($62 million vs. $165 million USD) Largely due to ability to recover part of rocket (Stage 1) Space Y wants to compete with Space X

Problem:

Space Y tasks us to train a machine learning model to predict successful Stage 1 recovery

METHODOLOGY



Data collection methodology :

Combined data from SpaceX public API and SpaceX Wikipedia page

Perform data wrangling :

Classifying true landings as successful and unsuccessful otherwise

Perform exploratory data analysis (EDA) :

Visualization and SQL

Perform interactive visual analytics :

Folium and Plotly Dash

Perform predictive analysis :

Classification models Tuned models using GridSearchCV

METHODOLOGY

Data Collection Overview

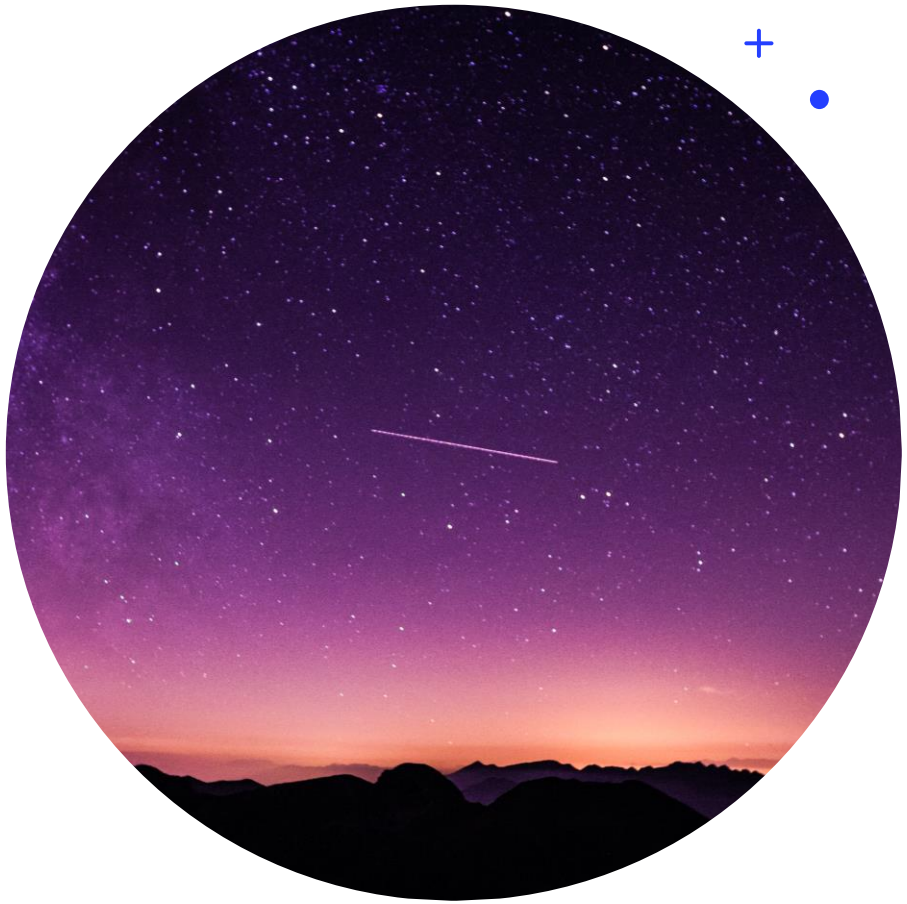
The data collection process involved a two-fold approach:

Space X API:

Utilized API requests from Space X's public API to gather relevant data.

Web Scraping:

Extracted data from a table in Space X's Wikipedia entry using web scraping techniques.



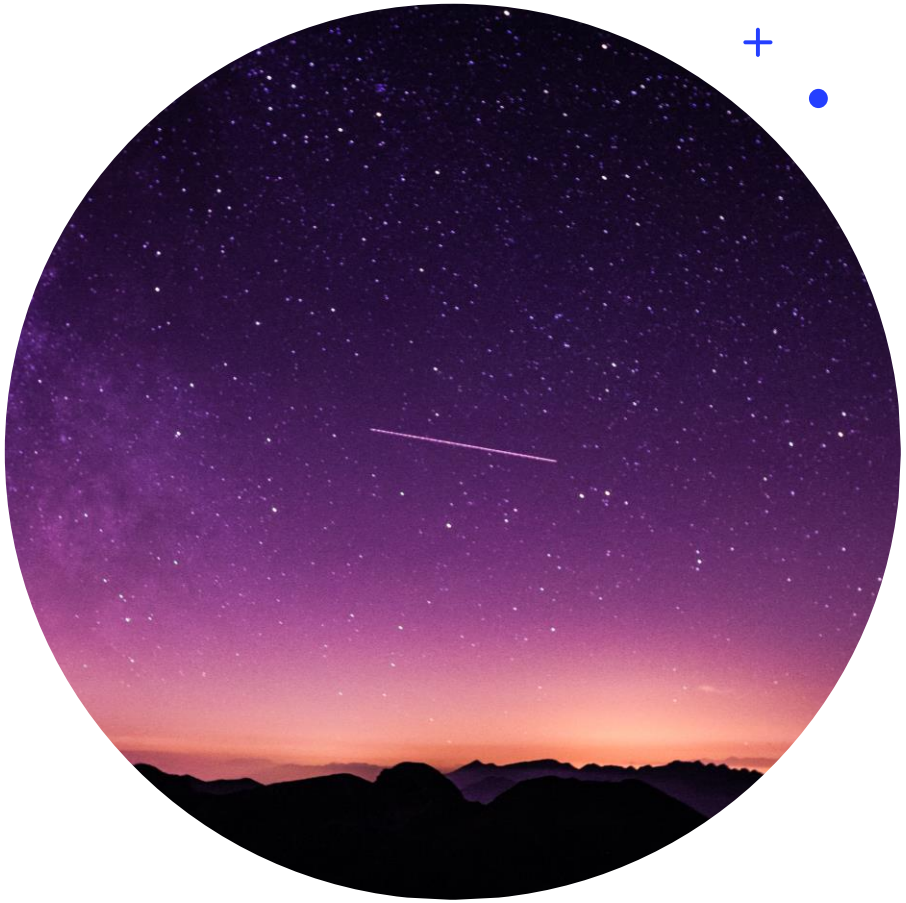
Data Collection Overview

Space X API Data Columns:

The following columns were collected through API requests: FlightNumber, Date, BoosterVersion, PayloadMass, Orbit, LaunchSite, Outcome, Flights, GridFins, Reused, Legs, LandingPad, Block, ReusedCount, Serial, Longitude, Latitude,

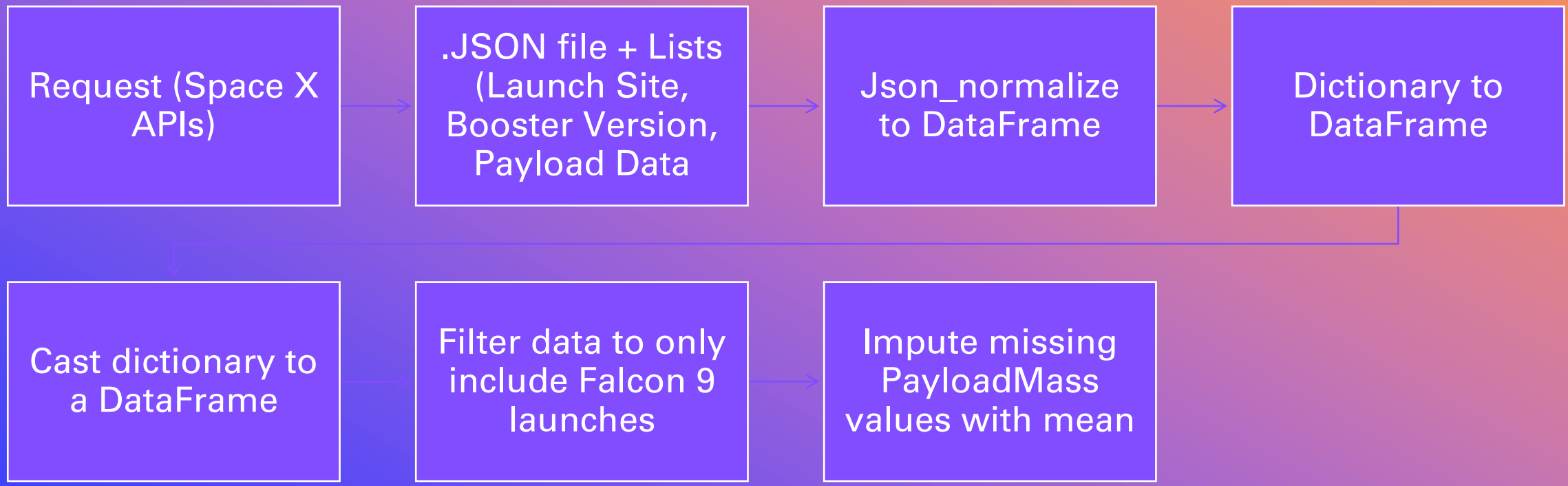
Wikipedia Web Scrape Data Columns:

The web scraping process retrieved the following columns from the Wikipedia table: Flight No., Launch site, Payload, PayloadMass, Orbit, Customer, Launch outcome, Version Booster, Booster landing, Date, Time



DATA COLLECTION – SPACEX API

https://github.com/job-nattawat/IBM_Data_Science/blob/main/Applied%20Data%20Science%20Capstone/jupyter-labs-spacex-data-collection-api.ipynb

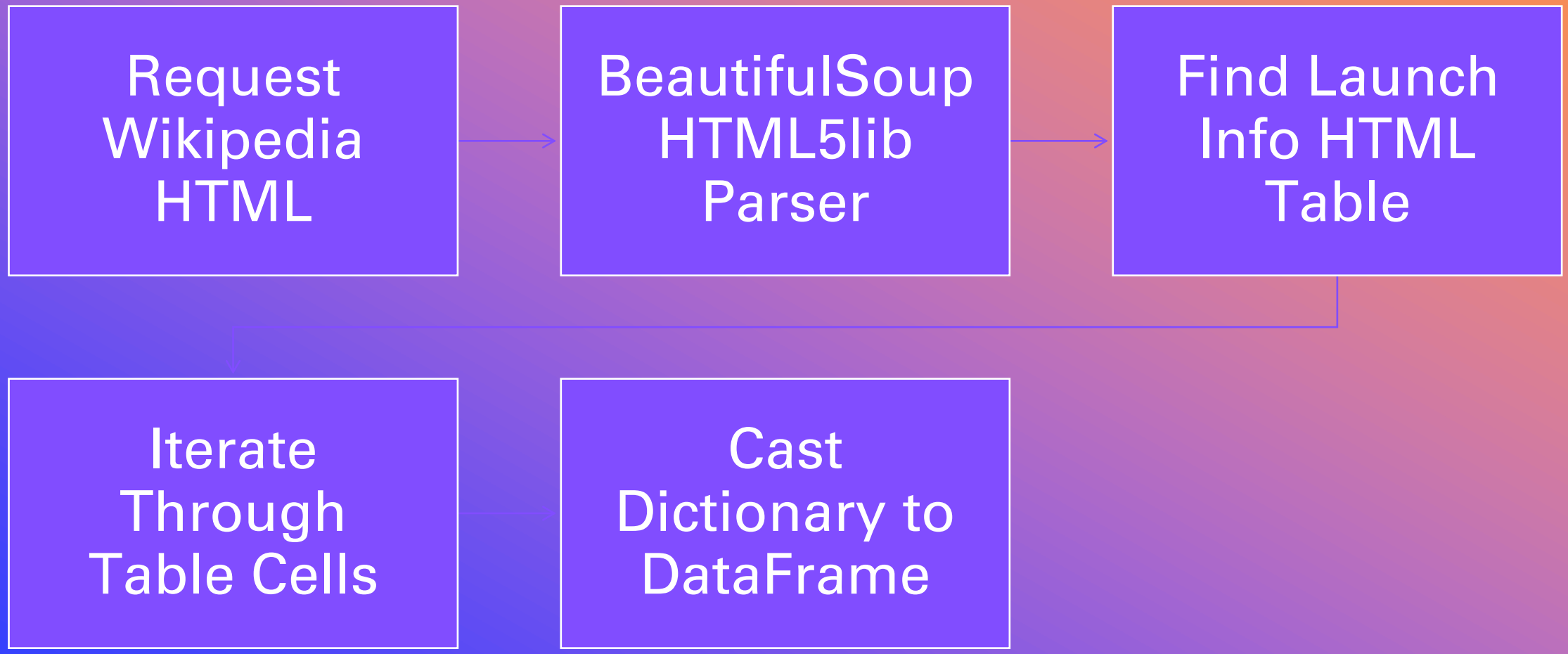


DATA COLLECTION – SPACEX API

- Request (Space X APIs):
  Sent API requests to Space X's APIs to retrieve data.
- .JSON file + Lists (Launch Site, Booster Version, Payload Data):
  Received the data in JSON format and extracted relevant information such as Launch Site, Booster Version, and Payload Data.
- Json_normalize to DataFrame:
  Utilized json_normalize to convert the JSON data into a structured DataFrame.
- Dictionary to DataFrame:
  Organized the relevant data into a dictionary format.
- Cast dictionary to a DataFrame:
  Transformed the dictionary into a DataFrame for further analysis.
- Filter data to only include Falcon 9 launches:
  Filtered the DataFrame to include data related to Falcon 9 launches only.
- Impute missing PayloadMass values with mean:
  Addressed missing values in the PayloadMass column by imputing them with the mean value.

DATA COLLECTION – WEB SCRAPING

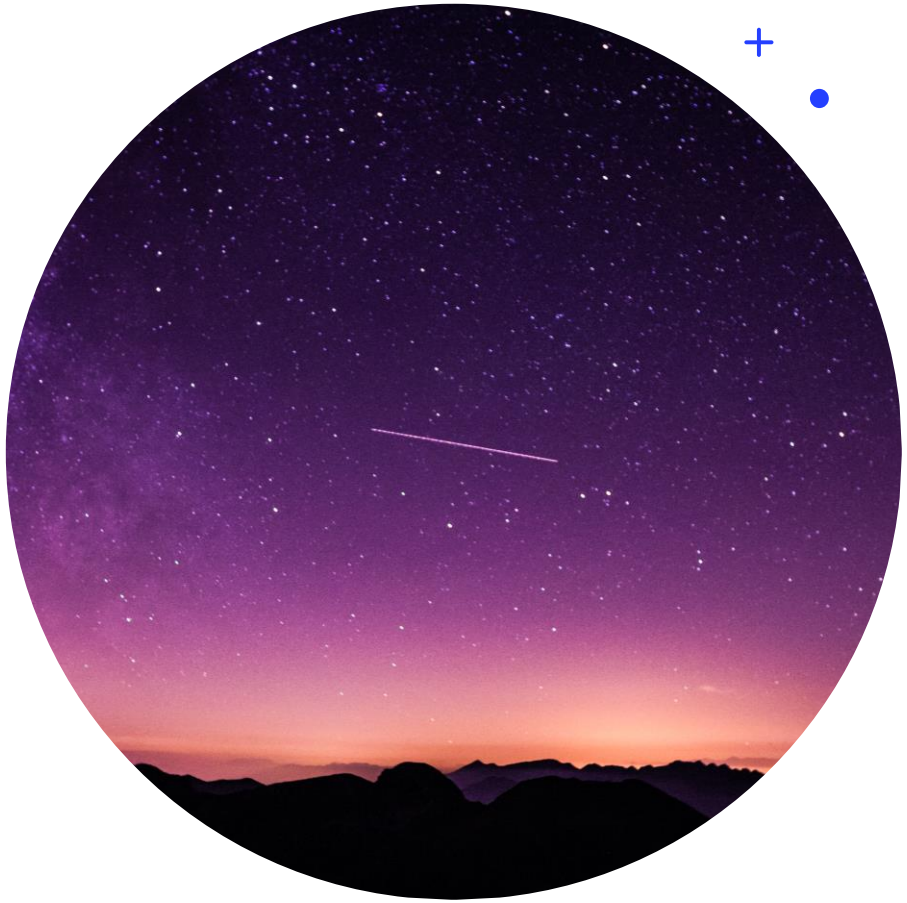
https://github.com/job-nattawat/IBM_Data_Science/blob/main/Applied%20Data%20Science%20Capstone/jupyter-labs-webscraping.ipynb



DATA COLLECTION – WEB SCRAPING

- Request Wikipedia HTML:
  Retrieved the HTML content of the Space X Wikipedia page using appropriate HTTP requests.
- BeautifulSoup HTML5lib Parser:
  Utilized the BeautifulSoup library with the HTML5lib parser to parse and extract data from the HTML content.
- Find Launch Info HTML Table:
  Identified and located the HTML table containing the launch information on the Wikipedia page.
- Iterate Through Table Cells:
  Iterated through the table cells to extract relevant data and organized it into a dictionary.
- Cast Dictionary to DataFrame:
  Converted the dictionary containing the extracted data into a structured DataFrame for further analysis.

Data Wrangling



Outcome Column Components:
The 'Outcome' column consists of two components: 'Mission Outcome' and 'Landing Location.'

Training Label Creation:
To create the training label, a new column named 'class' was introduced to the dataset.

Value Mapping:
The 'class' column is assigned a value of 1 if the 'Mission Outcome' component is True and the 'Landing Location' component is either "True ASDS," "True RTLS," or "True Ocean."

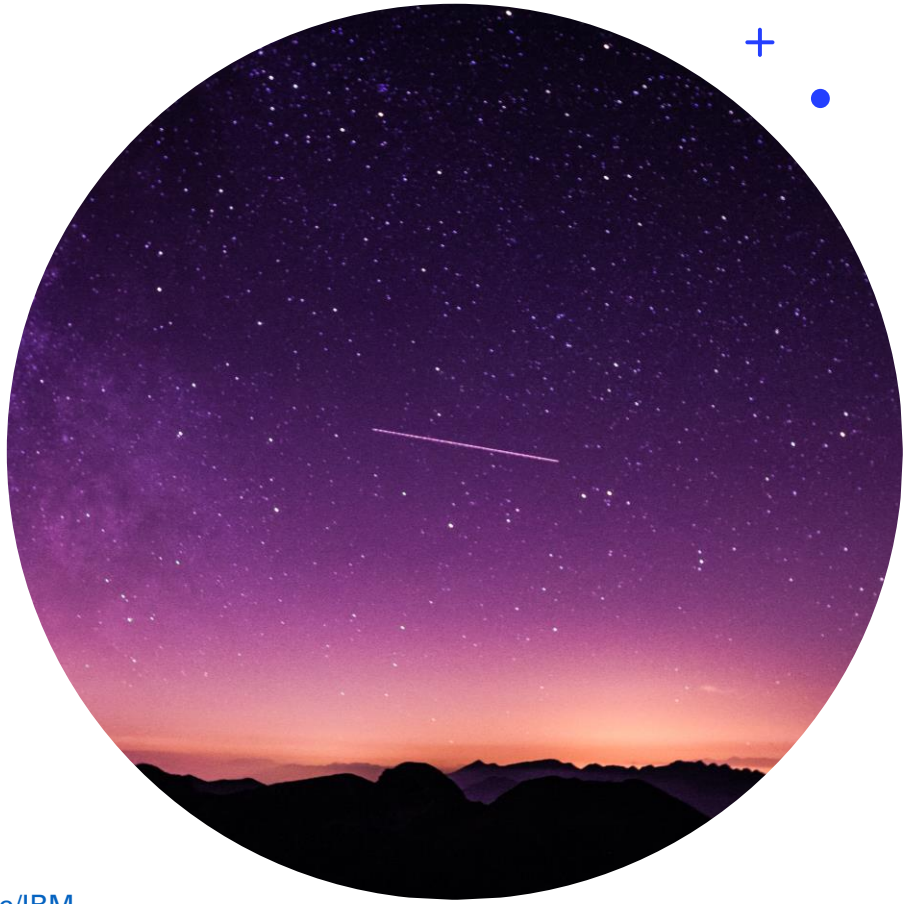
Value Mapping Cont'd:
The 'class' column is set to 0 if the 'Mission Outcome' component is either "None None," "False ASDS," "None ASDS," "False Ocean," or "False RTLS."

https://github.com/job-nattawat/IBM_Data_Science/blob/main/Applied%20Data%20Science%20Capstone/IBM-DS0321EN-SkillsNetwork_labs_module_1_L3_labs-jupyter-spacex-data_wrangling_jupyterlite.jupyterlite.ipynb

EDA with Data Visualization

The following variables were subjected to exploratory data analysis:

Flight Number, Payload Mass, Launch Site, Orbit, Class (Training Label), and Year.



https://github.com/job-nattawat/IBM_Data_Science/blob/main/Applied%20Data%20Science%20Capstone/IBM-DS0321EN-SkillsNetwork_labs_module_2_jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb

EDA with Data Visualization

Flight Number vs. Payload Mass:
Scatter plot used to examine the relationship between Flight Number and Payload Mass.

Flight Number vs. Launch Site:
Scatter plot to explore the distribution of Flight Number across different Launch Sites.

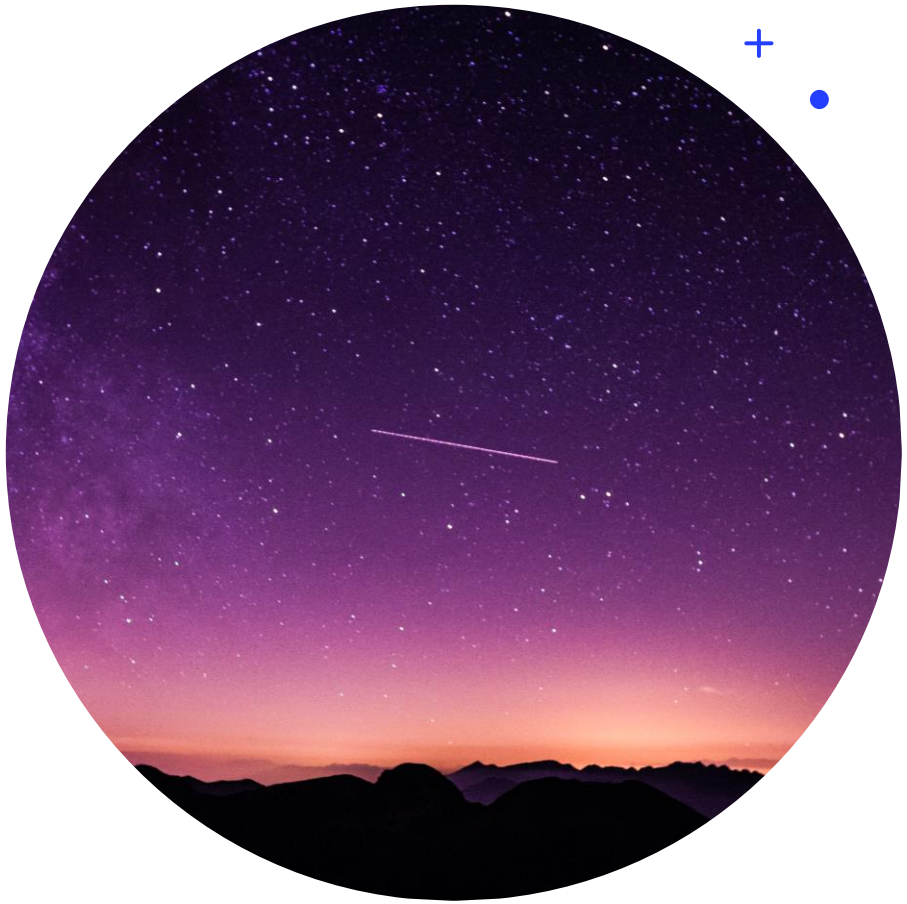
Payload Mass vs. Launch Site:
Scatter plot highlighting the relationship between Payload Mass and Launch Site.

Orbit vs. Success Rate:
Bar plot used to visualize the success rate for each Orbit type.

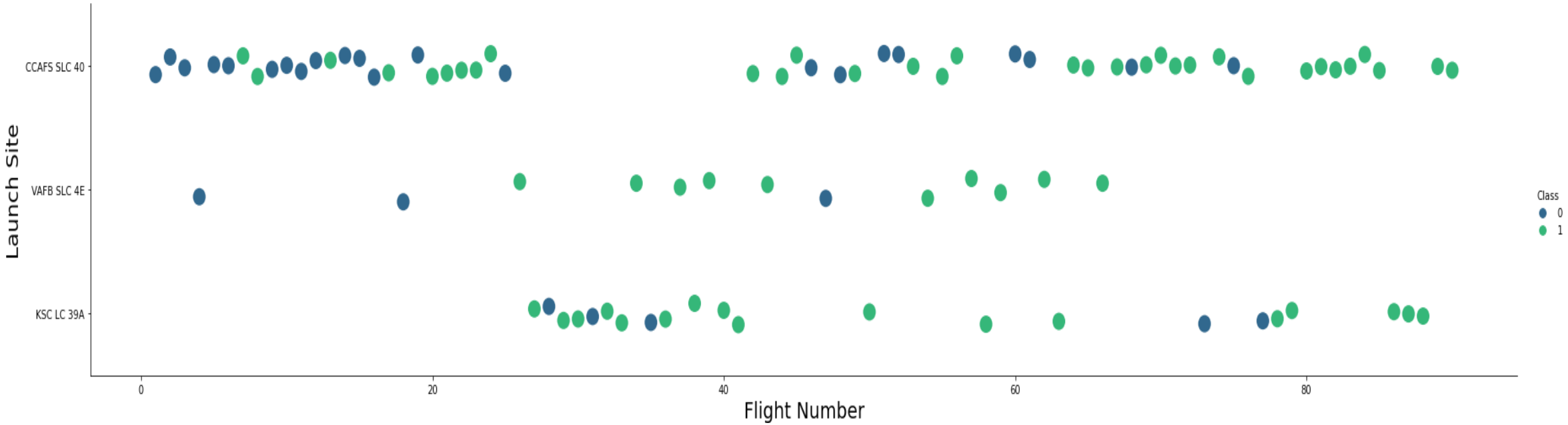
Flight Number vs. Orbit:
Scatter plot to analyze the relationship between Flight Number and the Orbit type..

Payload vs. Orbit:
Scatter plot to observe the relationship between Payload and Orbit.

Success Yearly Trend:
Line chart displaying the yearly trend of successful launches.

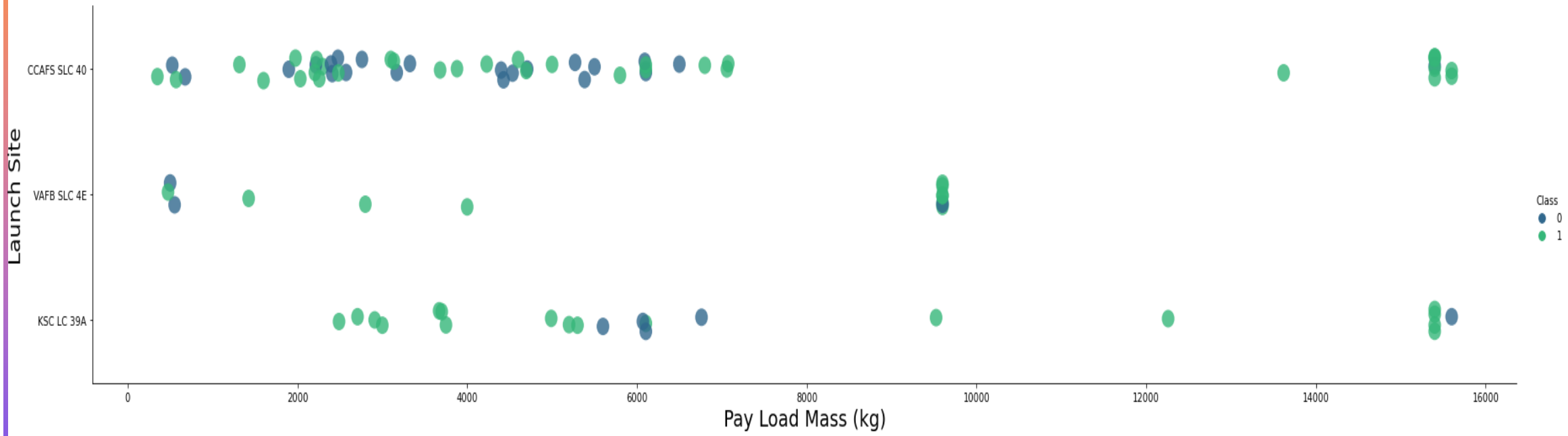


Flight Number vs. Launch Site



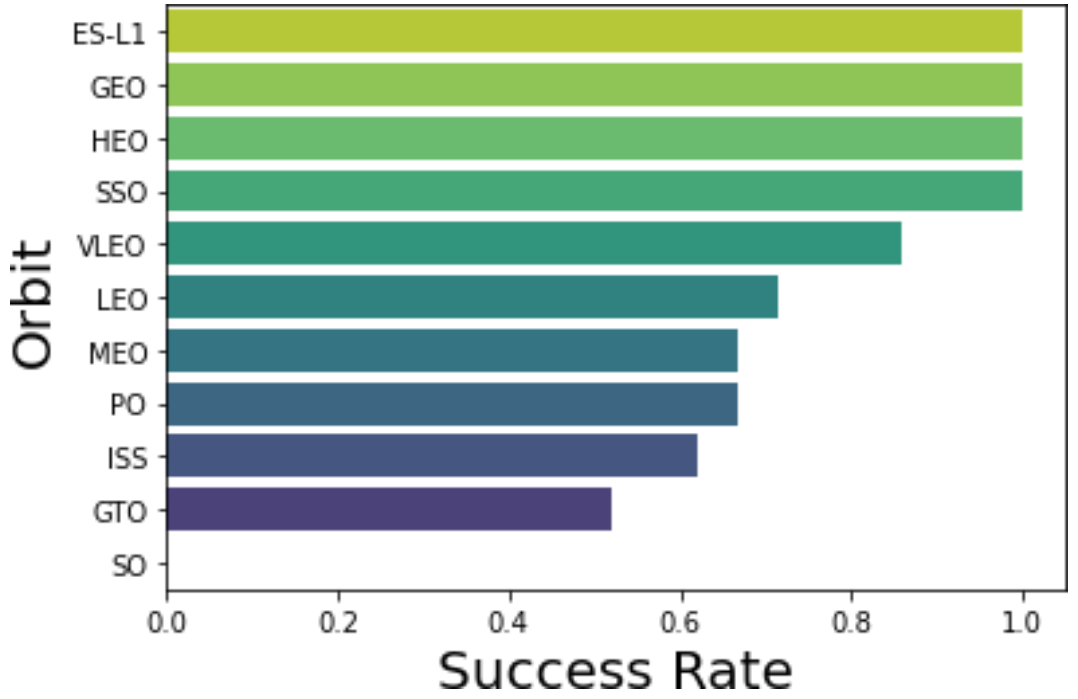
- Green indicates successful launch; Purple indicates unsuccessful launch.
- Graphic suggests an increase in success rate over time (indicated in Flight Number). Likely a big breakthrough around flight 20 which significantly increased success rate. CCAFS appears to be the main launch site as it has the most volume.

Payload vs. Launch Site



- Green indicates successful launch; Purple indicates unsuccessful launch.
- Payload mass appears to fall mostly between 0-6000 kg. Different launch sites also seem to use different payload mass.

Success rate vs. Orbit type



ES-L1 (1), GEO (1), HEO (1) have 100% success rate (sample sizes in parenthesis)

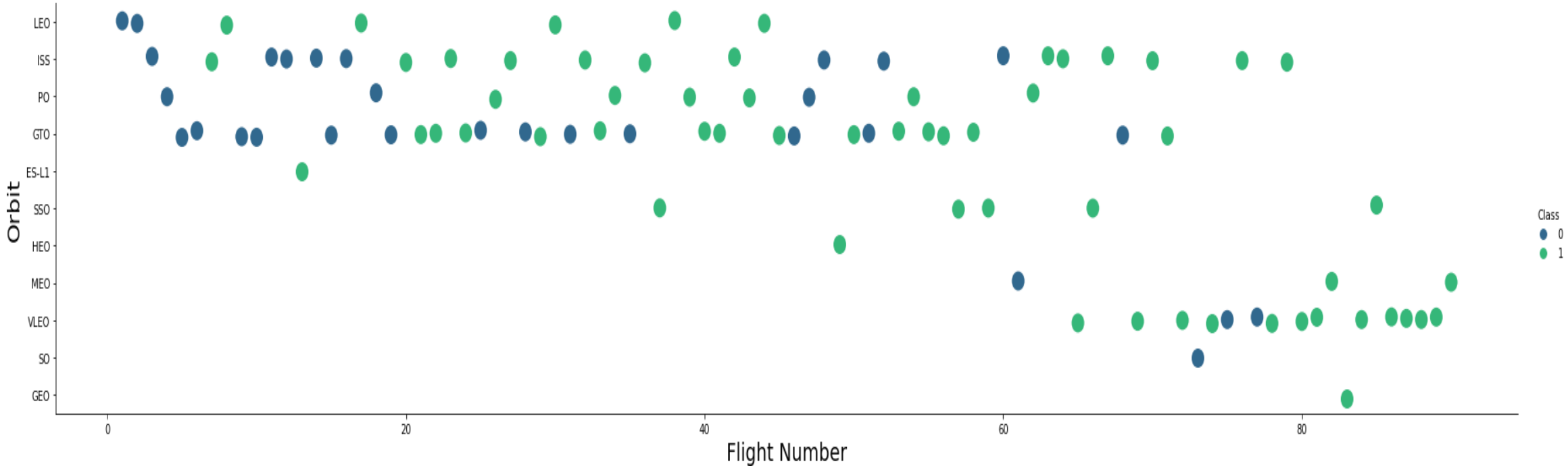
SSO (5) has 100% success rate

VLEO (14) has decent success rate and attempts

SO (1) has 0% success rate

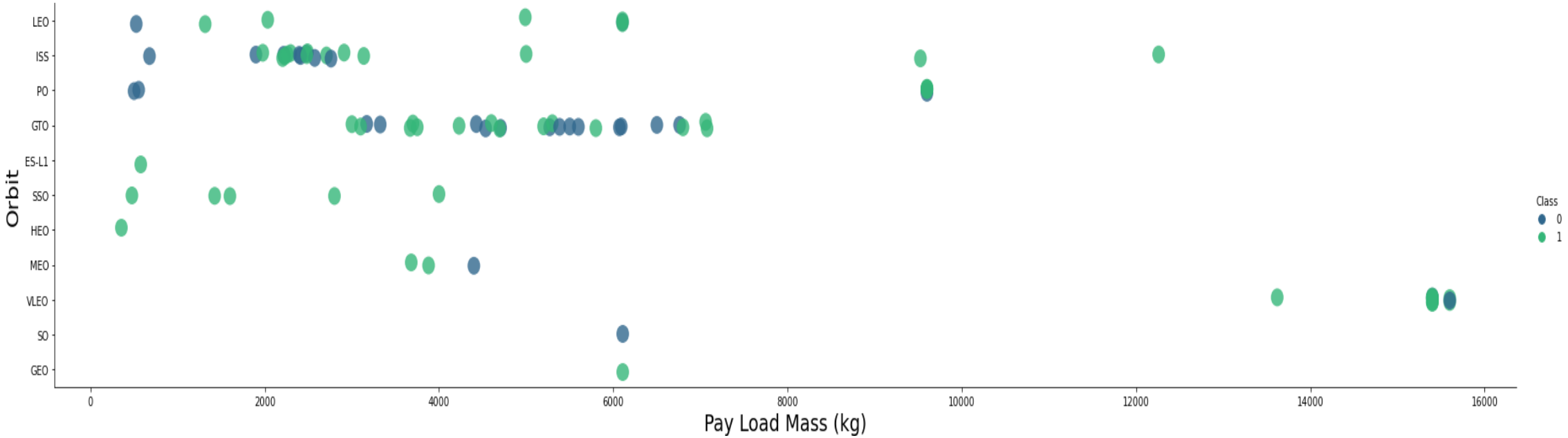
GTO (27) has the around 50% success rate but largest sample

Flight Number vs. Orbit type



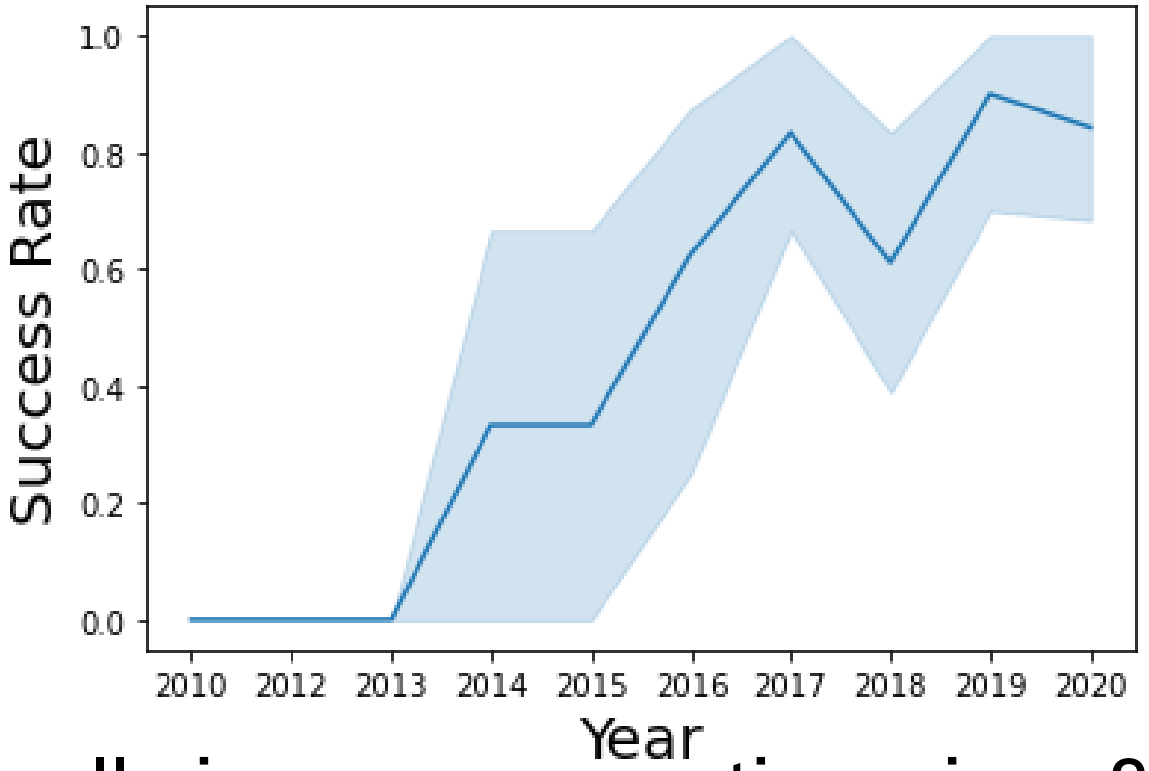
- Green indicates successful launch; Purple indicates unsuccessful launch.
- Launch Orbit preferences changed over Flight Number.
- Launch Outcome seems to correlate with this preference.
- SpaceX started with LEO orbits which saw moderate success LEO and returned to VLEO in recent launches
- SpaceX appears to perform better in lower orbits or Sun-synchronous orbits

Payload vs. Orbit type



- Green indicates successful launch; Purple indicates unsuccessful launch.
- Payload mass seems to correlate with orbit
- LEO and SSO seem to have relatively low payload mass
- The other most successful orbit VLEO only has payload mass values in the higher end of the range

Launch Success Yearly Trend



- Success generally increases over time since 2013 with a slight dip in 2018 Success in recent years at around 80%

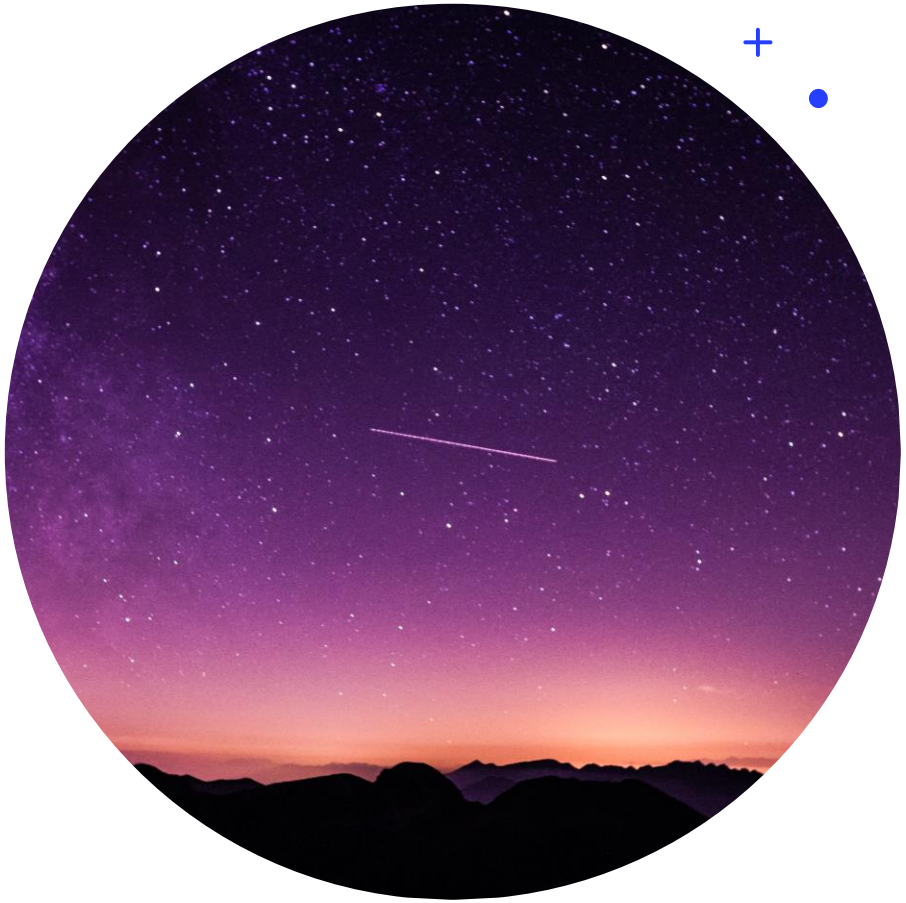
EDA with SQL

Loaded Dataset:
The dataset was loaded into the data analysis environment for further processing and querying.

SQL Python Integration:
Utilized SQL queries with Python integration to interact with the dataset and retrieve relevant information.

Gaining Insights:
Various queries were executed to gain a better understanding of the dataset and extract valuable insights.

Query Examples:
Queries were made to retrieve information such as launch site names, mission outcomes, various payload sizes of customers, booster versions, and landing outcomes.



https://github.com/job-nattawat/IBM_Data_Science/blob/main/Applied%20Data%20Science%20Capstone/jupyter-labs-eda-sql-coursera_sqllite.ipynb

All Launch Site Names

- Query unique launch site names from database.
- CCAFS SLC-40 and CCAFSSLC-40 likely all represent the same launch site with data entry errors.
- CCAFS LC-40 was the previous name.
- Likely only 3 unique launch_site values:
  CCAFS SLC-40, KSC LC-39A, VAFB SLC-4E

Launch_Site
CCAFS LC-40
VAFB SLC-4E
KSC LC-39A
CCAFS SLC-40
None

Launch Site Names Beginning with `CCA`

First five entries in database with Launch Site name beginning with CCA.

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
06/04/2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0.0	LEO	SpaceX	Success	Failure (parachute)
12/08/2010	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0.0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
22/05/2012	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525.0	LEO (ISS)	NASA (COTS)	Success	No attempt
10/08/2012	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500.0	LEO (ISS)	NASA (CRS)	Success	No attempt
03/01/2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677.0	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass from NASA

- This query sums the total payload mass in kg where NASA was the customer.
- CRS stands for Commercial Resupply Services which indicates that these payloads were sent to the International Space Station (ISS).

sum
45596.0

Average Payload Mass by F9 v1.1

- This query calculates the average payload mass or launches which used booster version F9 v1.1
- Average payload mass of F9 1.1 is on the low end of our payload mass range

Average
2534.6666666666665

First Successful Ground Pad Landing Date

- This query returns the first successful ground pad landing date.
- First ground pad landing wasn’t until the end of 2015.
- Successful landings in general appear starting 2014.

Date
01/06/2014

Successful Drone Ship Landing with Payload Between 4000 and 6000

- This query returns the four booster versions that had successful drone ship landings and a payload mass between 4000 and 6000 noninclusively.

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

Total Number of Each Mission Outcome

- This query returns a count of each mission outcome.
- SpaceX appears to achieve its mission outcome nearly 99% of the time.
- This means that most of the landing failures are intended.
- Interestingly, one launch has an unclear payload status and unfortunately one failed in flight.

Mission_Outcome	Count
None	898
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

Boosters that Carried Maximum Payload

- These booster versions are very similar and all are of the F9 B5 B10xx.x variety.
- This likely indicates payload mass correlates with the booster version that is used.

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 Failed Drone Ship Landing Records

- This query returns the Month, Landing Outcome, Booster Version, Payload Mass (kg), and Launch site of 2015 launches where stage 1 failed to land on a drone ship.
- There were two such occurrences.

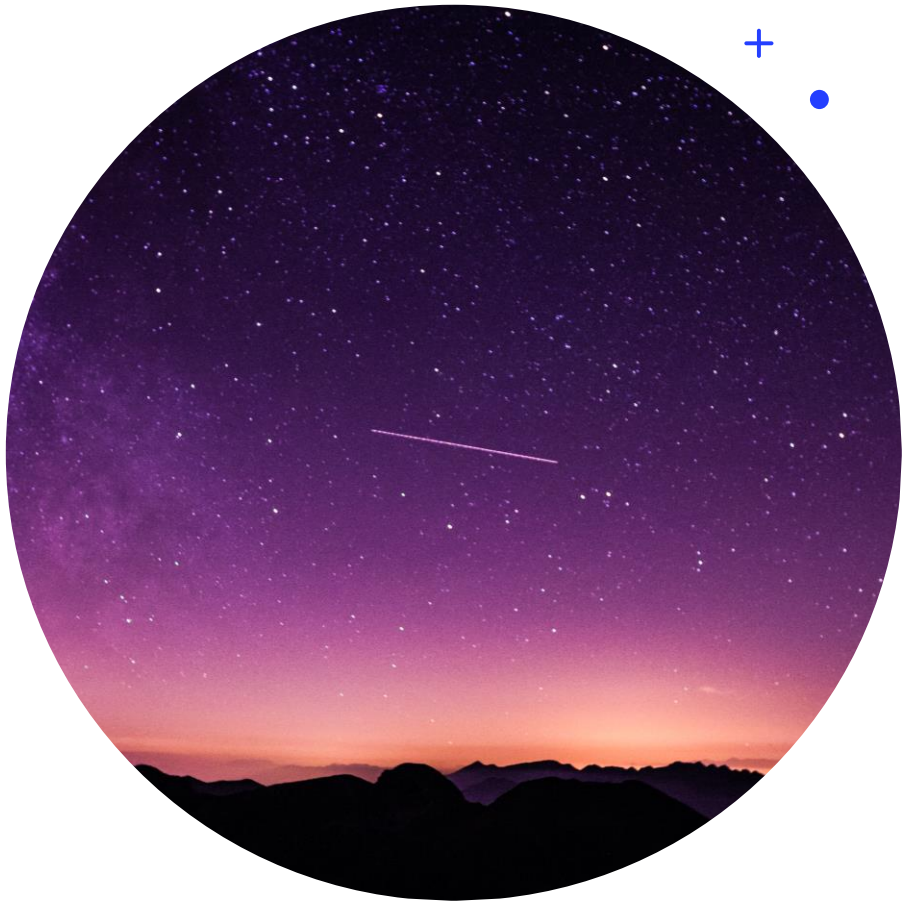
Month	Landing_Outcome	Booster_Version	Launch_Site
10	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

Ranking Counts of Successful Landings Between 2010-06-04 and 2017-03-20

- This query returns a list of successful landings and between 2010-06-04 and 2017-03-20 inclusively.
- There were 0 successful landings in total during this time period

Landing_Outcome	count

Build an interactive map with Folium



Launch Sites Marking:
Utilized Folium maps to mark the locations of launch sites on the map, providing a geographical context for each launch site.

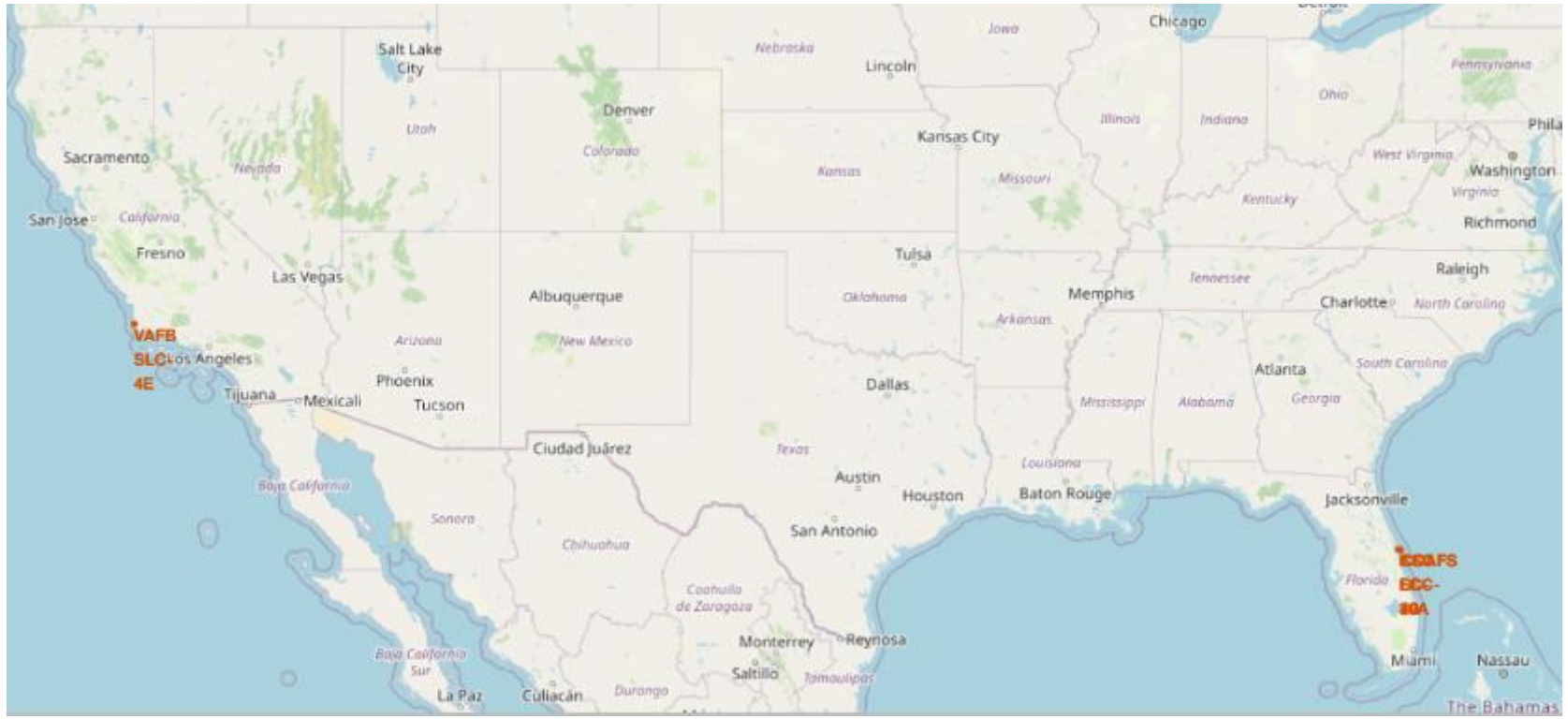
Successful and Unsuccessful Landings:
Incorporated markers or different symbols on the Folium maps to indicate the locations of both successful and unsuccessful landings, offering a visual representation of the outcomes.

Proximity Example:
Implemented a proximity example on the Folium map, showcasing key locations such as Railway, Highway, Coast, and City in relation to the launch sites. This enables us to understand why specific launch sites are chosen based on their proximity to significant infrastructure and populated areas.

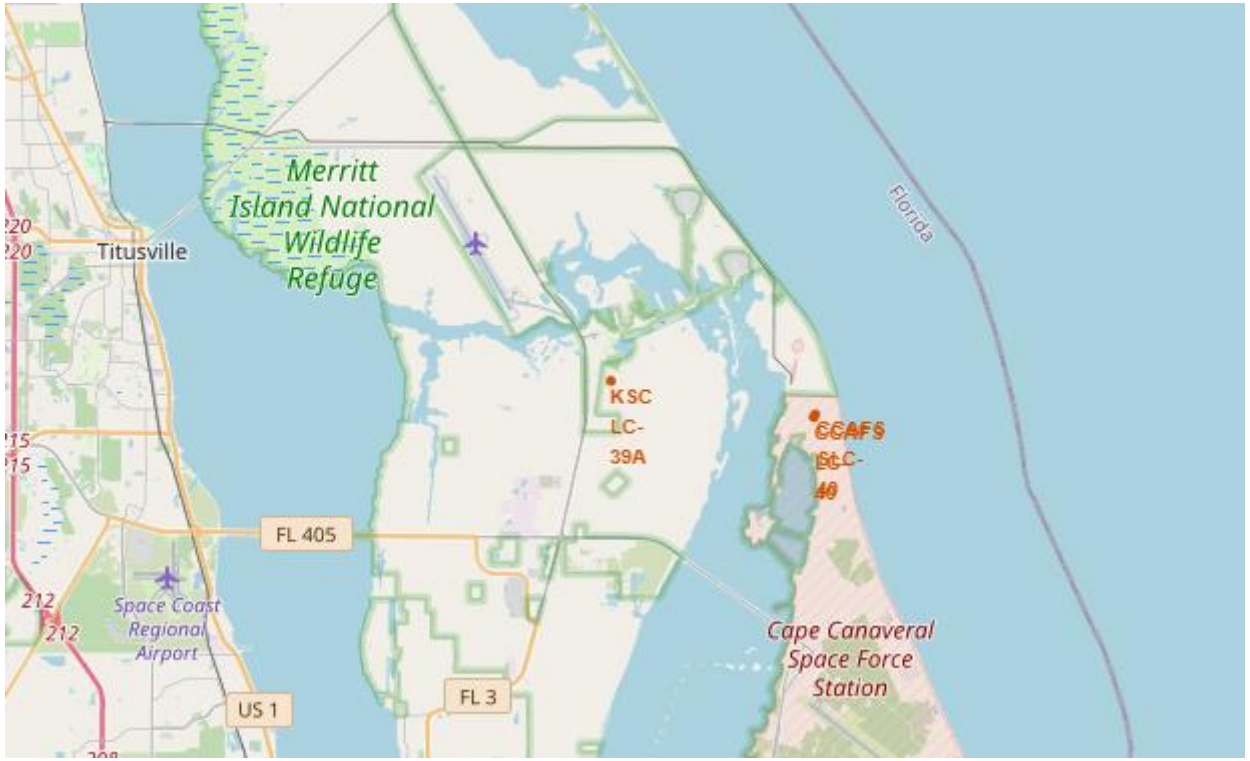
Visualizing Successful Landings:
The Folium map visualization displays the successful landing sites in relation to their geographic locations, providing insights into the distribution of successful landings across different regions.

https://github.com/job-nattawat/IBM_Data_Science/blob/main/Applied%20Data%20Science%20Capstone/IBM-DS0321EN-SkillsNetwork_labs_module_3_lab_jupyter_launch_site_location.jupyterlite.ipynb

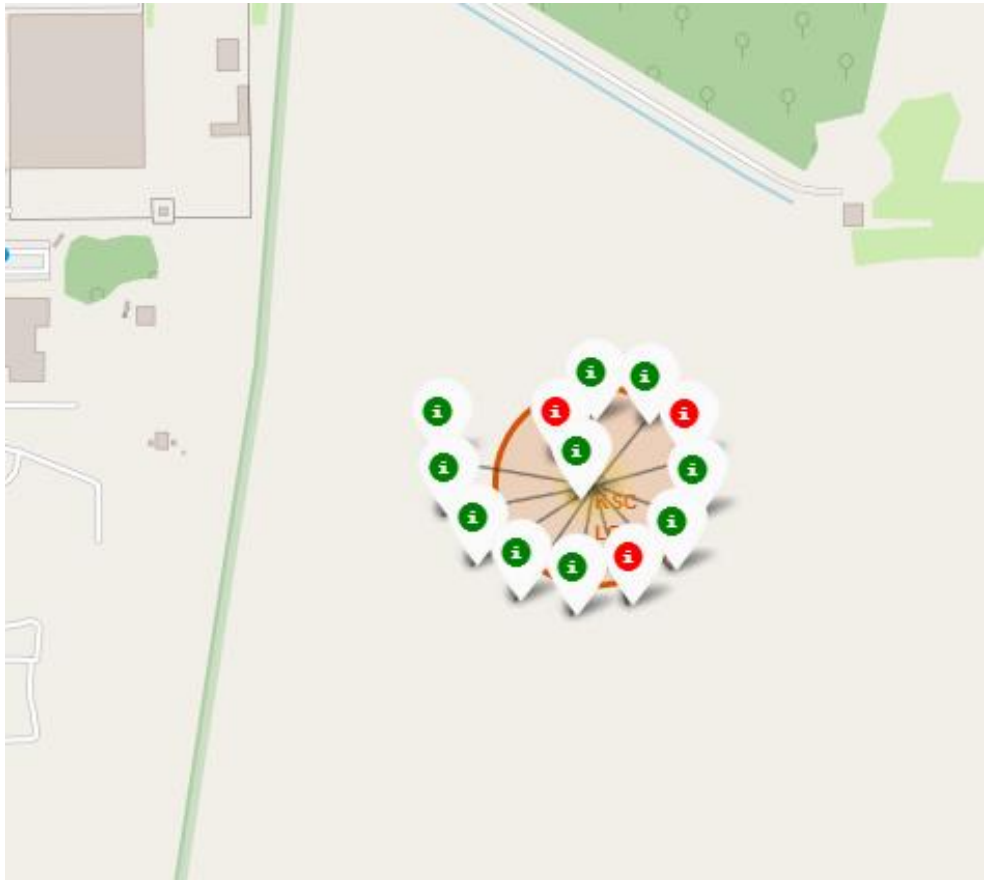
Launch Site Locations

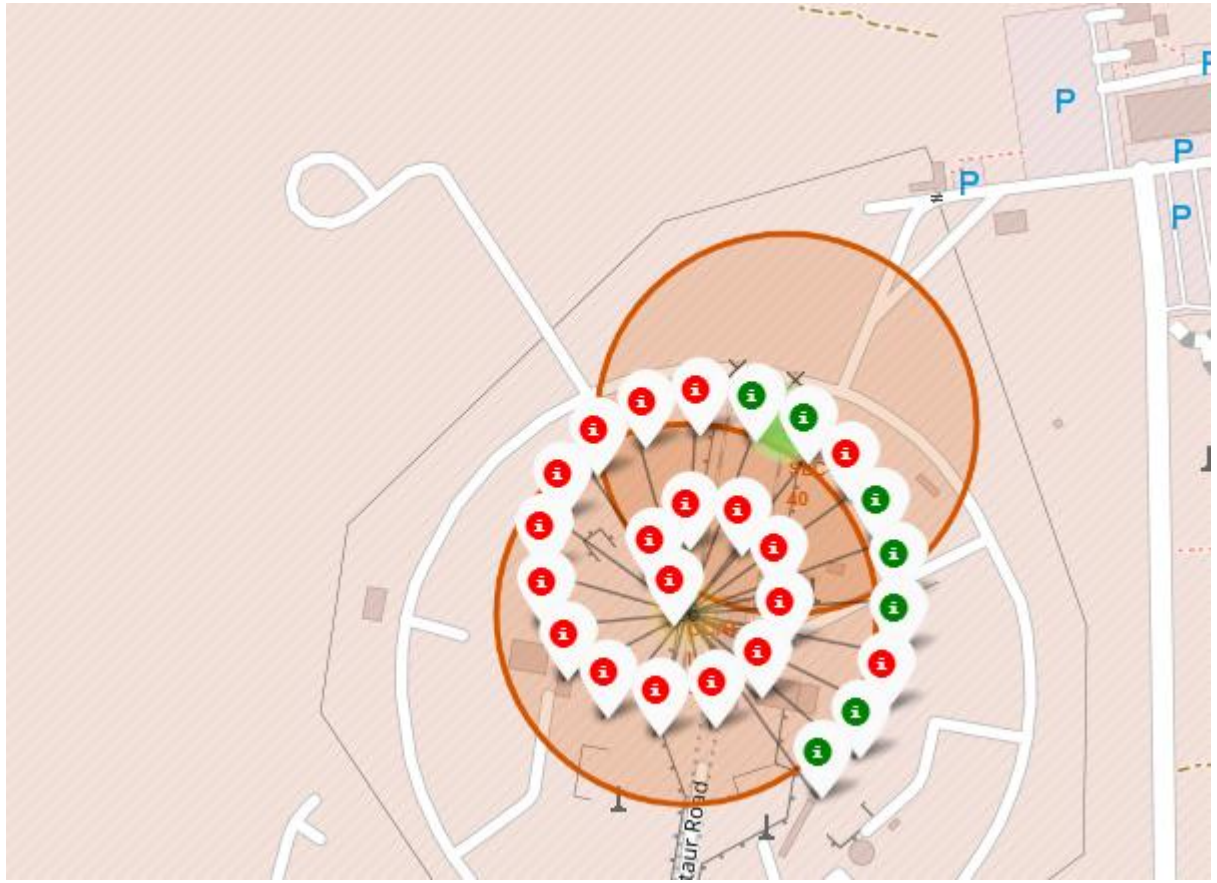


Launch Site Locations

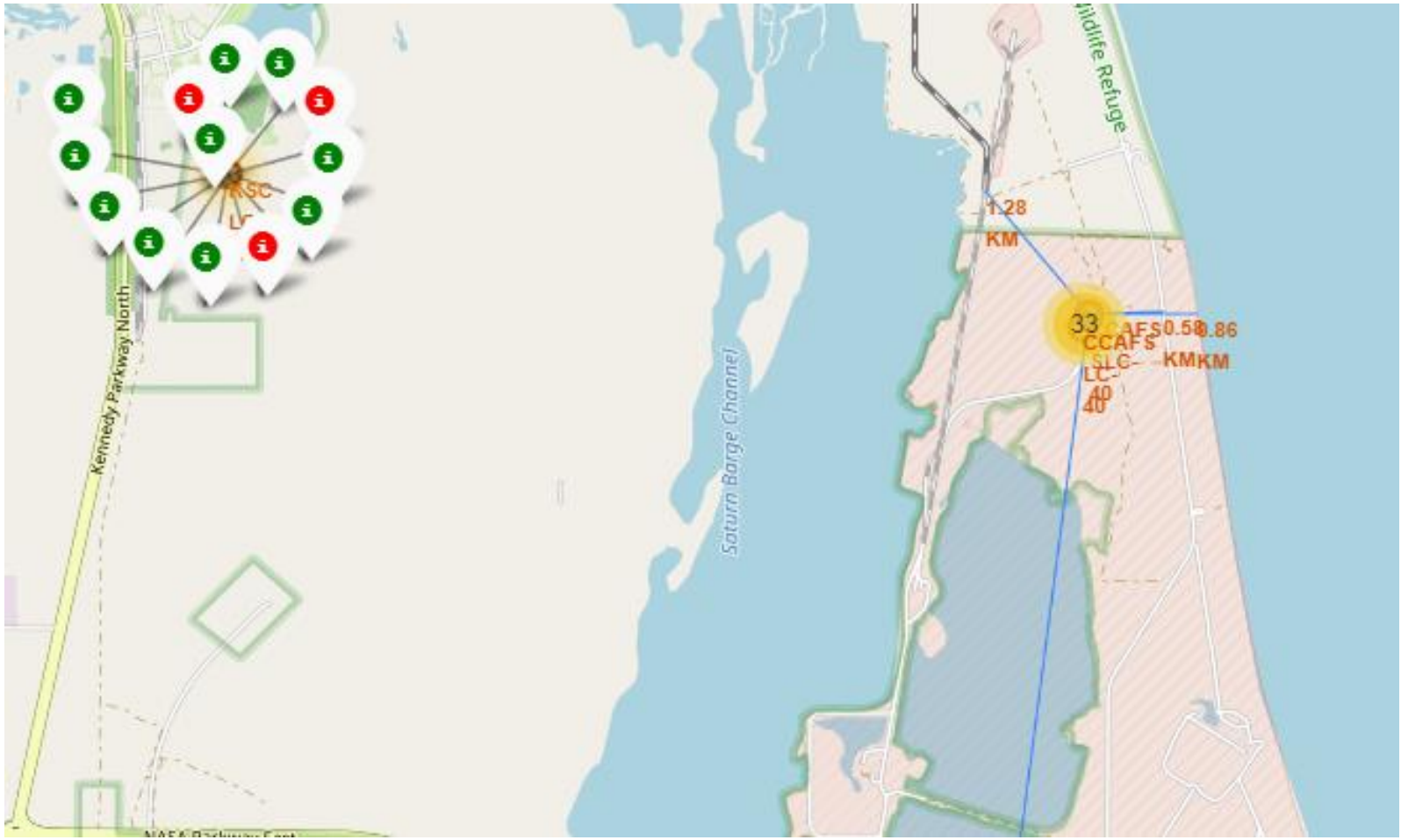


Color-Coded Launch Markers

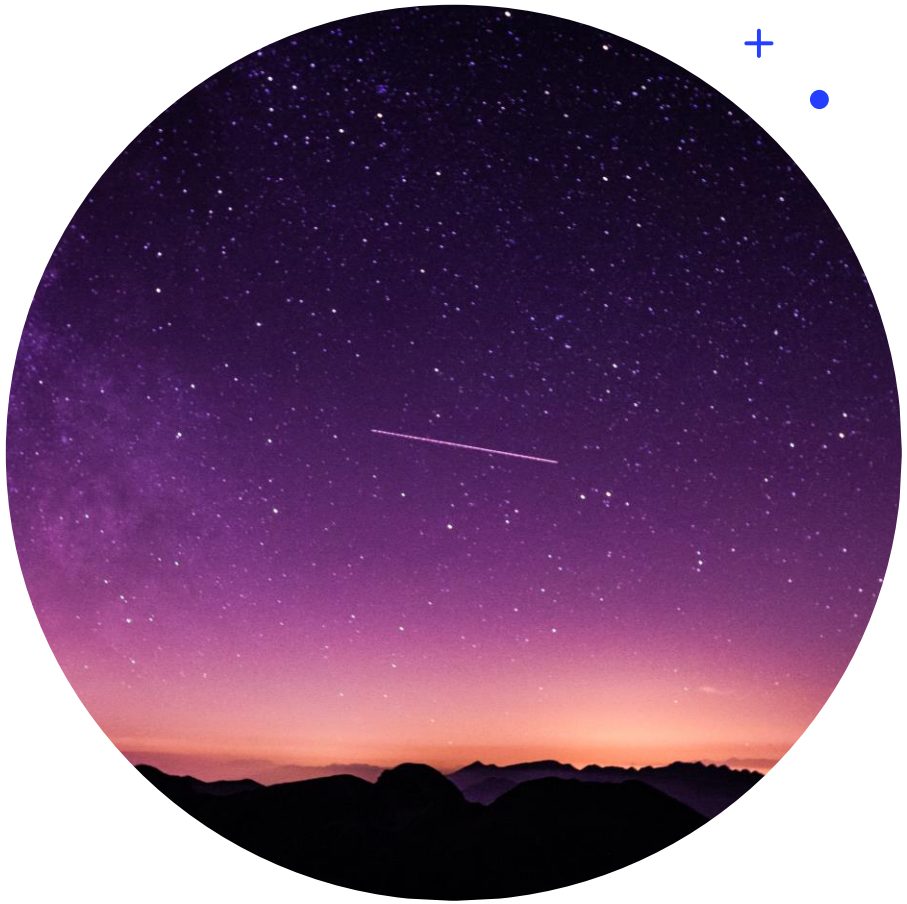




Key Location Proximities



Build a Dashboard with Plotly Dash



The dashboard consists of two main visualizations:
a pie chart and a scatter plot.

The pie chart allows users to toggle between two views. One view displays the distribution of successful landings across all launch sites, providing an overview of overall success rates. The other view enables users to select individual launch sites to visualize their specific success rates.

The scatter plot is interactive and accepts two inputs. Users can choose to view data for all launch sites combined or for individual launch sites. Additionally, a slider is provided to filter payload mass values between 0 and 10,000 kg. The scatter plot's primary purpose is to explore how success rates vary across different launch sites, payload masses, and booster version categories.

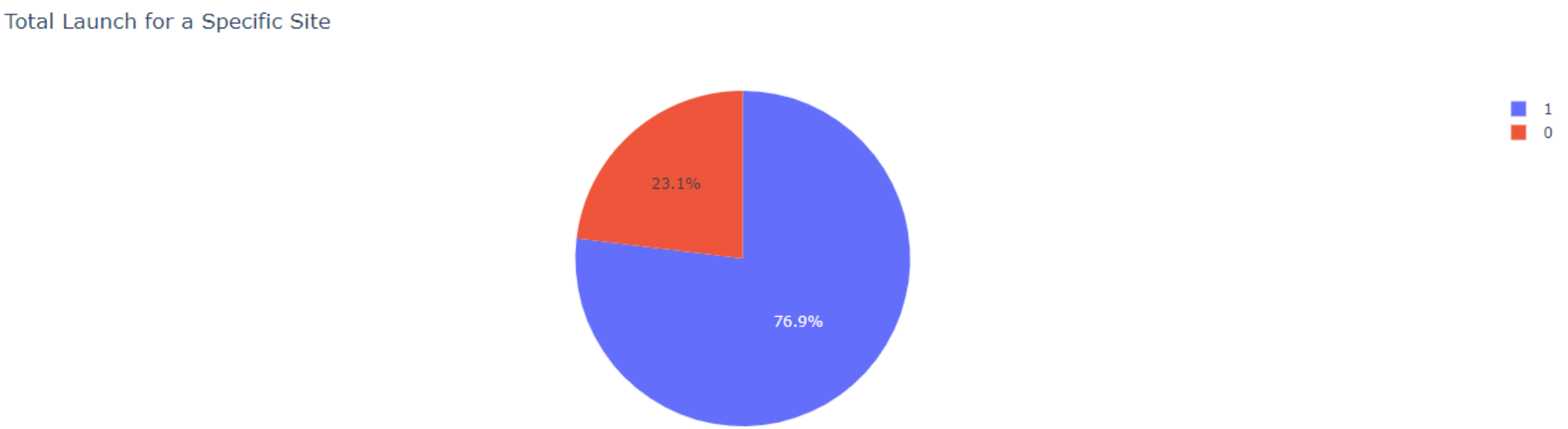
In summary, the pie chart presents a macro-level perspective of launch site success rates, while the interactive scatter plot allows for in-depth analysis of success variations with respect to launch sites, payload masses, and booster version categories.

https://github.com/job-nattawat/IBM_Data_Science/blob/main/Applied%20Data%20Science%20Capstone/spacex_dash_app.py

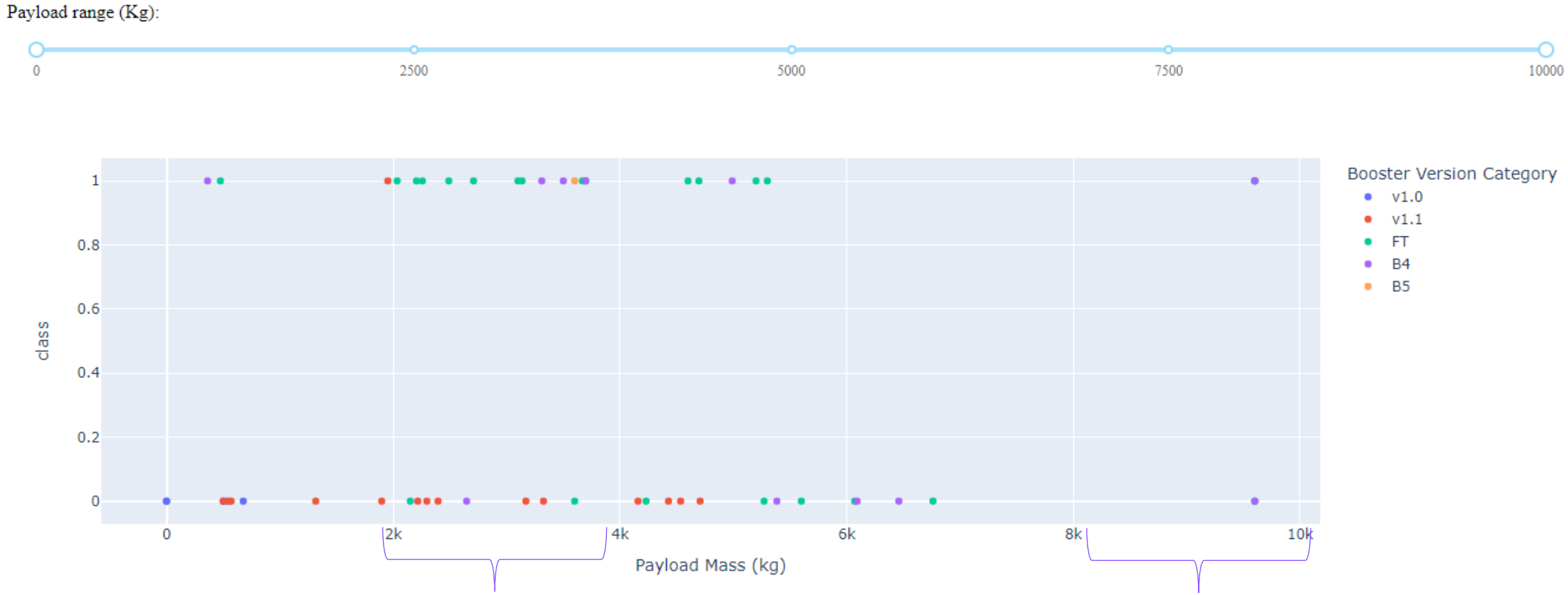
Largest Successful Launches

Highest Success Rate Launch Site

KSC LC-39A



Payload Mass vs. Success vs. Booster Version Category

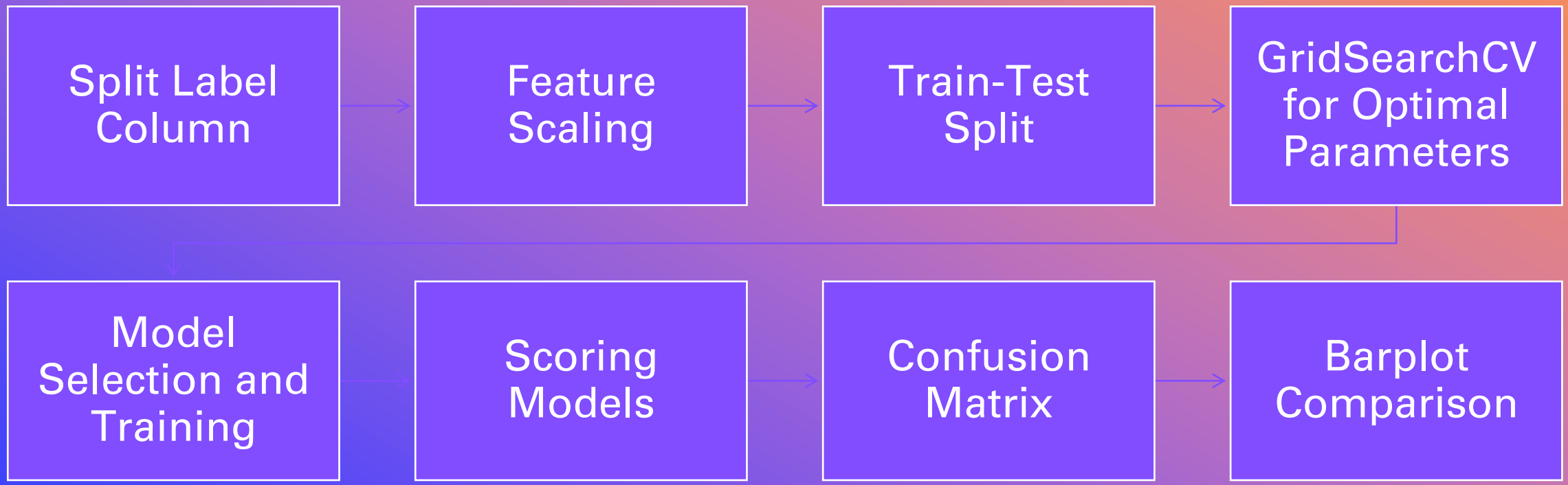


Highest Success Rate

Lowest Success Rate

PREDICTIVE ANALYSIS (CLASSIFICATION)

https://github.com/job-nattawat/IBM_Data_Science/blob/main/Applied%20Data%20Science%20Capstone/IBM-DS0321EN-SkillsNetwork_labs_module_4_SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

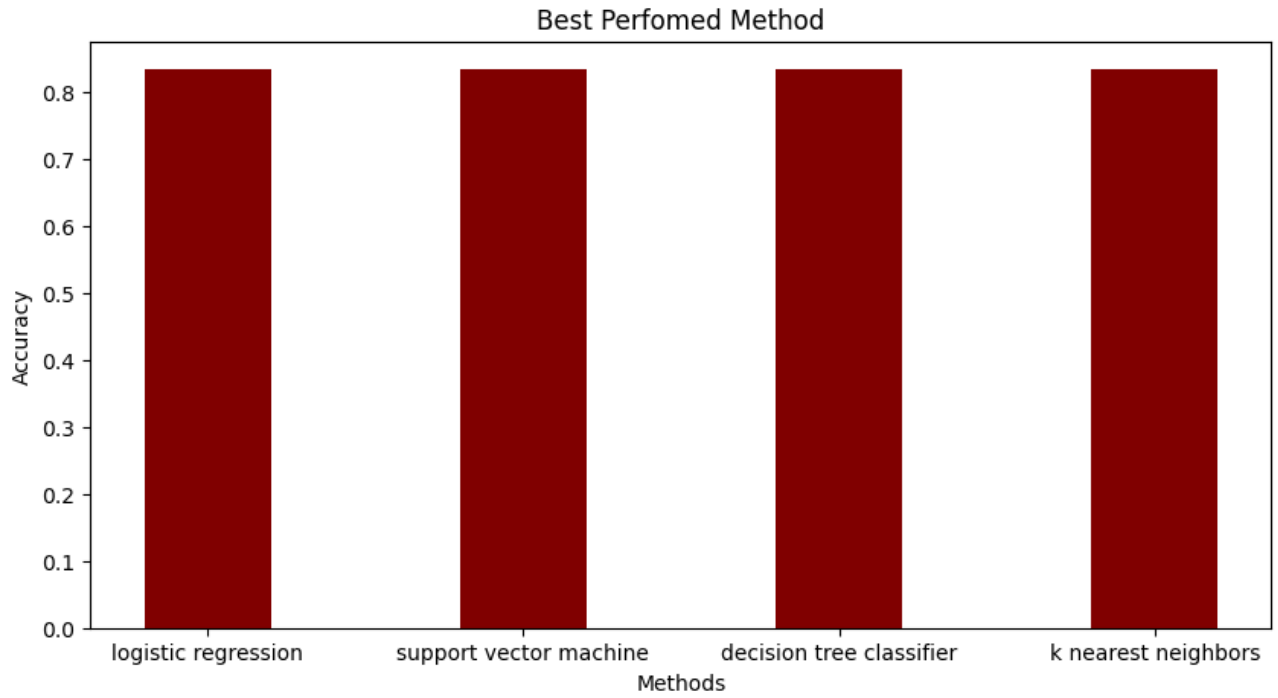


PREDICTIVE ANALYSIS (CLASSIFICATION)

- Split Label Column:
  The 'Class' column was separated from the dataset, serving as the target variable for the machine learning models.
- Feature Scaling:
  The features were scaled using Standard Scaler to ensure that all input variables were on the same scale, facilitating better model training.
- Train-Test Split:
  The data was split into training and testing sets to evaluate model performance on unseen data effectively.
- GridSearchCV for Optimal Parameters:
  GridSearchCV with cross-validation (cv=10) was utilized to find the optimal hyperparameters for each machine learning model, enhancing their predictive performance.
- Model Selection and Training:
  GridSearchCV was applied to train Logistic Regression, Support Vector Machine (SVM), Decision Tree, and K Nearest Neighbors (KNN) models.
- Scoring Models:
  The trained models were scored on the split test set to evaluate their performance using suitable metrics.
- Confusion Matrix:
  Confusion matrices were generated for all models, providing insights into their true positive, true negative, false positive, and false negative predictions.
- Barplot Comparison:
  A barplot was constructed to compare the performance scores (accuracy) of the different models, facilitating an easy assessment of their relative strengths.

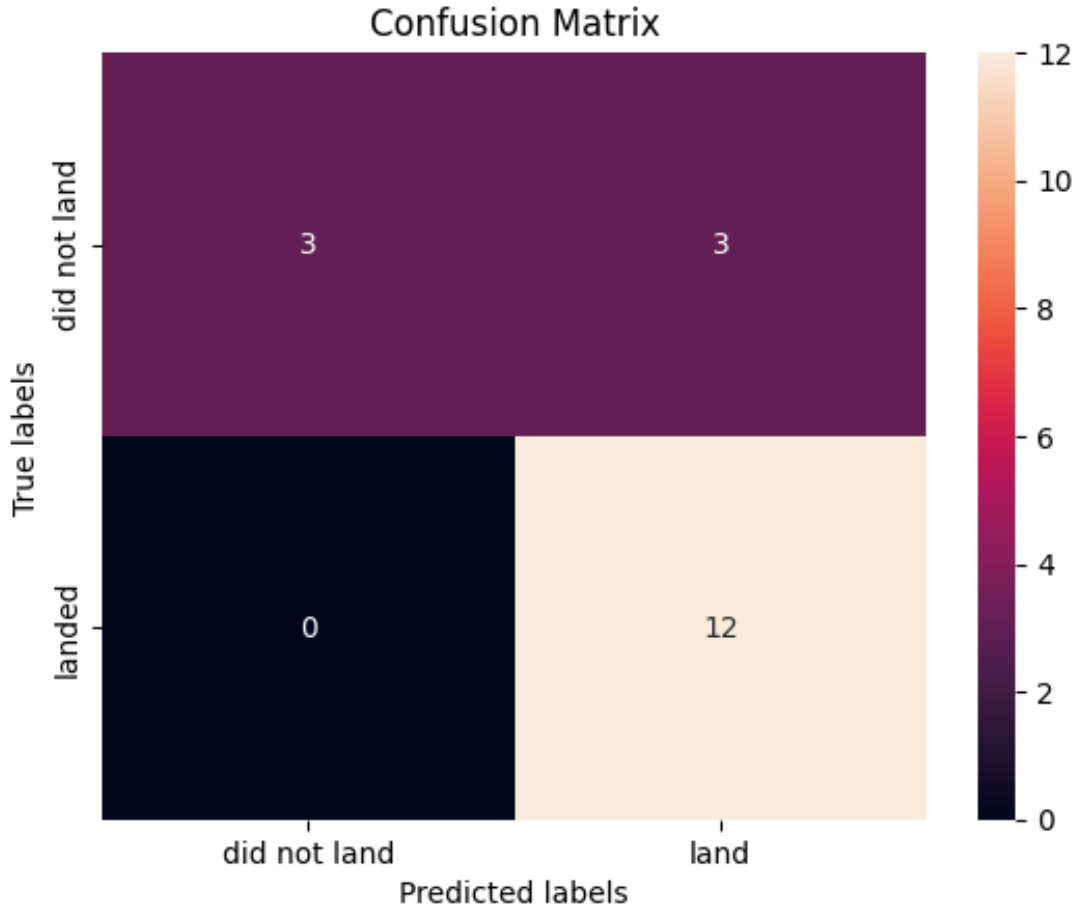
Classification Accuracy

All models had virtually the same accuracy on the test set at 83.33% accuracy. It should be noted that test size is small at only sample size of 18. This can cause large variance in accuracy results, such as those in Decision Tree Classifier model in repeated runs. We likely need more data to determine the best model.



Confusion Matrix

Since all models performed the same for the test set, the confusion matrix is the same across all models. The models predicted 12 successful landings when the true label was successful landing. The models predicted 3 unsuccessful landings when the true label was unsuccessful landing. The models predicted 3 successful landings when the true label was unsuccessful landings (false positives). Our models over predict successful landings.



CONCLUSION

Our task was to develop a machine learning model for Space Y, aiming to compete with SpaceX. The primary goal of the model was to predict whether Stage 1 of the rocket would successfully land, thereby saving approximately $100 million USD per launch.

To accomplish this, we collected data from both a public SpaceX API and web scraping the SpaceX Wikipedia page. The data was labeled and stored into a DB2 SQL database for efficient management.

For effective data visualization, we created a dashboard to explore and analyze the data. Using this dataset, we developed a machine learning model that achieved an accuracy of 83%.

This model can be utilized by Allon Mask of SpaceY to predict with a relatively high accuracy whether a launch will have a successful Stage 1 landing before the launch itself. By having this predictive capability, SpaceY can make informed decisions about whether to proceed with the launch or not.

As a recommendation, we suggest collecting more data to further enhance the model's performance and accuracy. More data will allow us to determine the best machine learning model and fine-tune its parameters, ultimately improving its predictive capabilities.

APPENDIX

GitHub repository url:

https://github.com/job-nattawat/IBM_Data_Science/tree/main/Applied%20Data%20Science%20Capstone

Rav Ahuja, Alex Aklson, Aije Egwaikhide, Svetlana Levitan, Romeo Kienzler, Polong Lin, Joseph Santarcangelo, Azim Hirjani, Hima Vasudevan, Saishruthi Swaminathan, Saeed Aghabozorgi, Yan Luo

Special Thanks to All Instructors:

https://www.coursera.org/professional-certificates/ibm-data-science?#instructors

THANK YOU

Nattawat Tanalurkmongkol

https://github.com/job-nattawat