IBM SpaceX Capstone Project

Welcome to the IBM SpaceX Capstone Project! Join us on a journey to explore the goals, process, findings, and impact of this groundbreaking collaboration.

Saghar Ganji

E-Mail : sagharganji82@gmail.com

GitHub

sagharganji - Overview

sagharganji has 3 repositories available. Follow their code on GitHub.

Goals of the Project

Advancing Space Exploration

Develop innovative solutions to enhance space exploration technologies and capabilities.

Improving Data Analysis

Leverage IBM's expertise in data analytics to support SpaceX's mission in acquiring and interpreting crucial data.

Enhancing Collaboration

Foster collaboration between leading organizations in tech and space industries to push boundaries and drive progress.

Creating Commercial Opportunities

Identify potential commercial applications and spin-off technologies from the project.

Outline

1. Executive Summary
2. Introduction
3. Methodology
4. Results
5. Conclusion
6. Appendix

Executive Summary

- Collected data from public SpaceX API and SpaceX Wikipedia page. Created labels column ‘class’ which classifies successful landings. Explored data using SQL, visualization, folium maps, and dashboards. Gathered relevant columns to be used as features. Changed all categorical variables to binary using one hot encoding. Standardized data and used GridSearchCV to find best parameters for machine learning models. Visualize accuracy score of all models.
- Four machine learning models were produced: Logistic Regression, Support Vector Machine, Decision Tree Classifier, and K Nearest Neighbors. All produced similar results with accuracy rate of about 83.33%. All models over predicted successful landings. More data is needed for better model determination and accuracy.

Introduction to IBM and SpaceX

IBM

International Business Machines Corporation (IBM) is a global technology company focused on AI, cloud computing, and cybersecurity.

SpaceX

Space Exploration Technologies Corp. (SpaceX) is a private aerospace manufacturer and space transportation company led by Elon Musk.

In the backdrop of this project lies my affiliation with SpaceX, specifically focusing on predicting the outcomes of the Falcon 9 first stage. The significance of accurately forecasting whether the rockets will successfully land cannot be overstated, as failures would entail considerable resource expenditures for the company.

Key Questions to Address:

- 1.What factors contribute to the failure of the landing process?
- 2.Can we reliably predict the successful landing of the rockets?
- 3.What level of accuracy can be achieved in forecasting successful landings?

These pivotal questions underscore the project's mission to identify critical variables influencing landing outcomes, establish a robust predictive model, and quantify the accuracy of our predictions in ensuring the success of Falcon 9 first stage landings.

Methodology

Executive Summary

- Data collection methodology:
  - With Rest API and Web Scraping
- Perform data wrangling
  - Data were transformed and one hot encoded to be apply later on the Machine Learning models
- Perform exploratory data analysis (EDA) using visualization and SQL
  - Discovering new patterns in the data with visualization techniques such as scatter plots
- Perform interactive visual analytics using Folium and Plotly Dash
  - Dash and Folium were used to achieve this goal
- Perform predictive analysis using classification models
  - Classification machine learning models were built to achieve this goal

Data Collection

Data collection process involved a combination of API requests from Space X public API and web scraping data from a table in Space X's Wikipedia entry.

The next slide will show the flowchart of data collection from API and the one after will show the flowchart of data collection from webscraping.

Space X API Data Columns:

- FlightNumber, Date, BoosterVersion, PayloadMass, Orbit, LaunchSite, Outcome, Flights, GridFins,
- Reused, Legs, LandingPad, Block, ReusedCount, Serial, Longitude, Latitude

Wikipedia Webscrape Data Columns:

- Flight No., Launch site, Payload, PayloadMass, Orbit, Customer, Launch outcome, Version Booster, Booster landing, Date, Time

Data Collection – SpaceX API

1. Request (Space X APIs)
2. .JSON file + Lists(Launch Site, Booster Version, Payload Data)
3. Json_normalize to DataFrame data from JSON
4. Dictionary relevant data
5. Cast dictionary to a DataFrame
6. Filter data to only include Falcon 9 launches
7. Imputate missing PayloadMass values with mean

GitHub repo:

github.com

GitHub repo

Data Collection – Web Scraping

1. Request Wikipedia

   html

2. BeautifulSoup

   html5lib Parser

3. Find launch info html table

4. Create dictionary

5. Iterate through table cells to extract data to the dictionary

6. Cast dictionary to DataFrame

GitHub repo:

github.com

GitHub repo

Data Wrangling

Generate a training label for landing outcomes, assigning a value of 1 for success and 0 for failure. The Outcome column encompasses two elements: 'Mission Outcome' and 'Landing Location.'

Introduce a new training label column called 'class,' assigning a value of 1 if 'Mission Outcome' is True and 0 otherwise. Implement the following value mapping:

- True ASDS, True RTLS, & True Ocean → set to 1
- None None, False ASDS, None ASDS, False Ocean, False RTLS → set to 0

GitHub repo:

github.com

GitHub repo

EDA with Data Visualization

Conducted Exploratory Data Analysis (EDA) on key variables including Flight Number, Payload Mass, Launch Site, Orbit, Class, and Year.

Utilized the following plots for analysis:

- Flight Number vs. Payload Mass
- Flight Number vs. Launch Site
- Payload Mass vs. Launch Site
- Orbit vs. Success Rate
- Flight Number vs. Orbit
- Payload vs. Orbit
- Success Yearly Trend

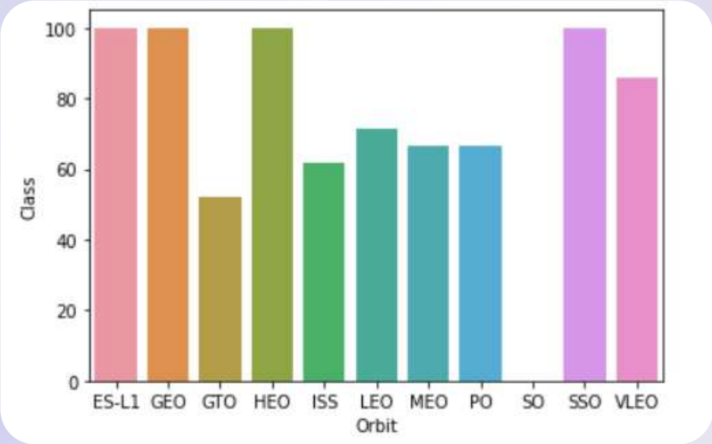
Employed scatter plots, line charts, and bar plots to discern relationships between variables. This analysis aimed to identify existing connections that could be utilized in training the machine learning model.

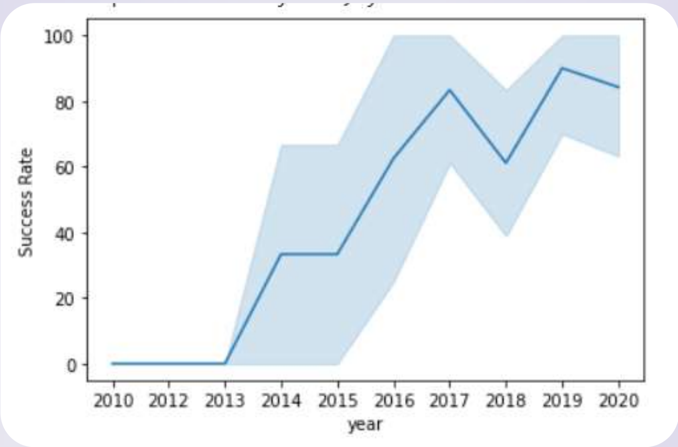
GitHub repo:

github.com

GitHub repo

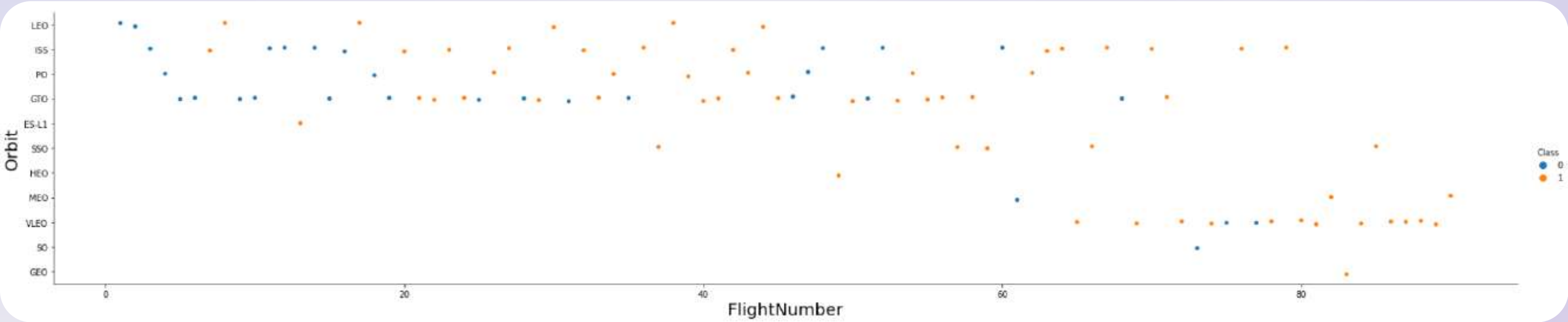
EDA with Data Visualization





Line chart between Year and Success rate

Line chart between Year and Success rate



Scatter plot between Orbit and FlightNumber

EDA with SQL

I employed SQL queries to address the following inquiries:

1. Retrieve the names of unique launch sites involved in the space missions.
2. Display 5 records where launch sites begin with the string 'CCA'.
3. Determine the total payload mass carried by boosters launched by NASA (CRS).
4. Calculate the average payload mass carried by booster version F9 v1.1.
5. Identify the date of the first successful landing outcome on an in-ground pad.
6. List the names of boosters achieving success on a drone ship with a payload mass greater than 4000 but less than 6000.
7. Provide the total number of successful and failed mission outcomes.
8. List the names of booster versions carrying the maximum payload mass, using a subquery.
9. Detail failed landing outcomes on drone ships, including their booster versions and launch site names for the year 2015.
10. Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the dates 2010-06-04 and 2017-03-20 in descending order.

GitHub repo:

github.com

GitHub repo

Build an interactive map with Folium

Utilized folium.Marker() to generate markers on the maps, enhancing visual representation. Employed folium.Circle() to create circles above markers, adding a dynamic layer to the map. Leveraged folium.Icon() to establish custom icons for specific locations on the map.

For creating connections, folium.PolyLine() was implemented to craft a polynomial line between points, while folium.plugins.AntPath() was applied to introduce an animated line, enhancing the visual appeal of the connection.

To streamline maps with multiple markers sharing identical coordinates, the markerCluster() function was employed, providing a more organized and simplified display.

GitHub repo:

github.com

GitHub repo

Build a Dashboard with Plotly Dash

Constructed a dynamic Dashboard using Plotly Dash:

- Employed Dash and HTML components as the backbone of the Dashboard, serving as the pivotal elements for integrating graphs, tables, dropdowns, and other essential features.
- Simplified data manipulation tasks by leveraging Pandas to create and manage the dataframe seamlessly.
- Utilized Plotly for graph plotting, with a focus on employing Pie charts and Scatter charts to enhance data visualization.
- Implemented a Rangeslider to facilitate convenient selection within the payload mass range.
- Incorporated a Dropdown feature for launch sites, enhancing user interactivity and providing a more tailored experience.

GitHub repo:

github.com

GitHub repo

Predictive analysis(Classification)

1

Building the model

1) Create column for the class.
2) Standardize the data.
3) Split the data info train and test sets.
4) Build GridSearchCV model and fit the data.

2

Evaluating the model

1) Calculating the accuracies.
2) Calculating the confusion matrixes.
3) Plot the results .

3

Finding the optimal model

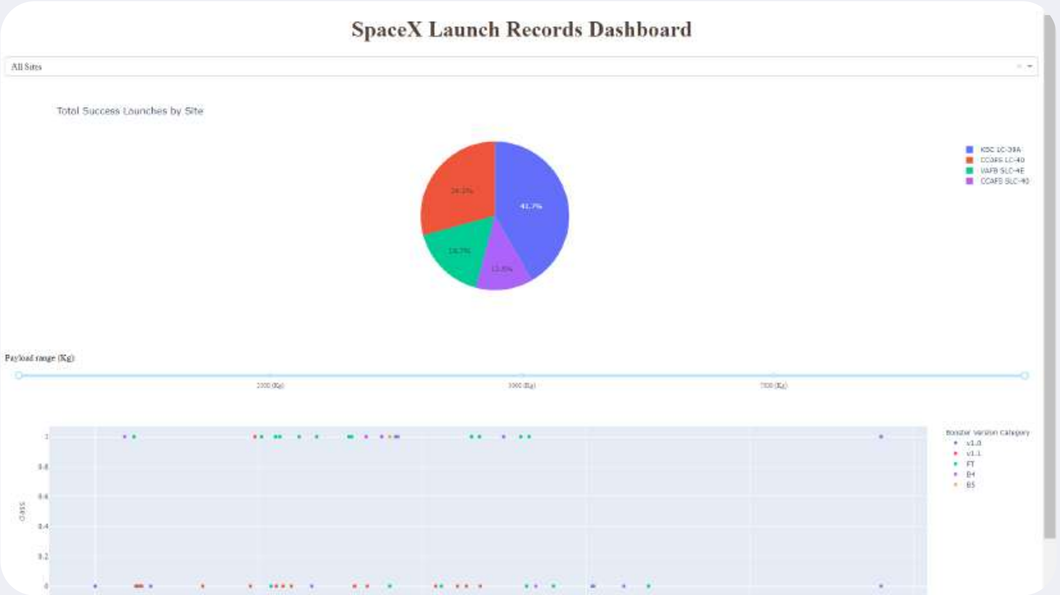
1) Find the best hyperparameters for the models.
2) Find the best model with highest accuracy

3) Confirm the optimal model.

github.com

GitHub repo

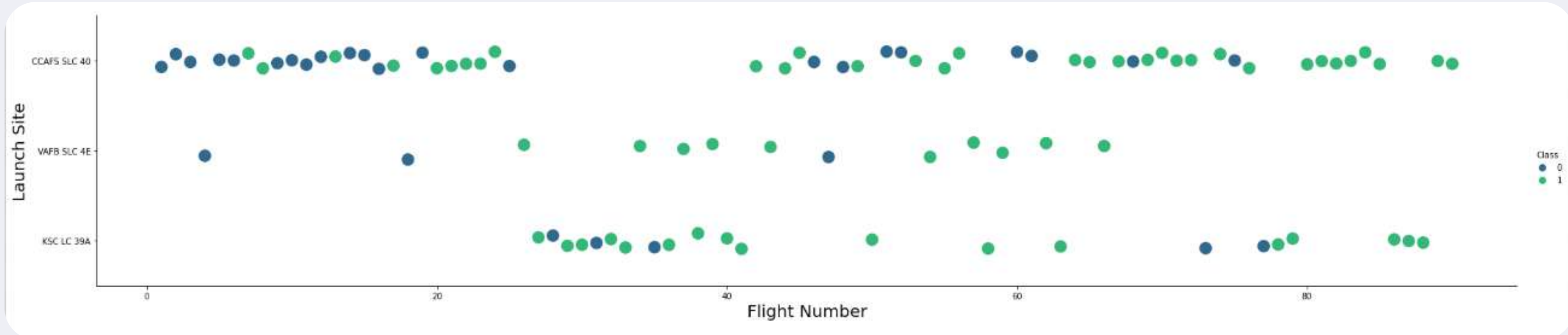
Results



This is a preview of the Plotly dashboard. The following sides will show the results of EDA with visualization, EDA with SQL, Interactive Map with Folium, and finally the results of our model with about 83% accuracy.

EDA with Visualization

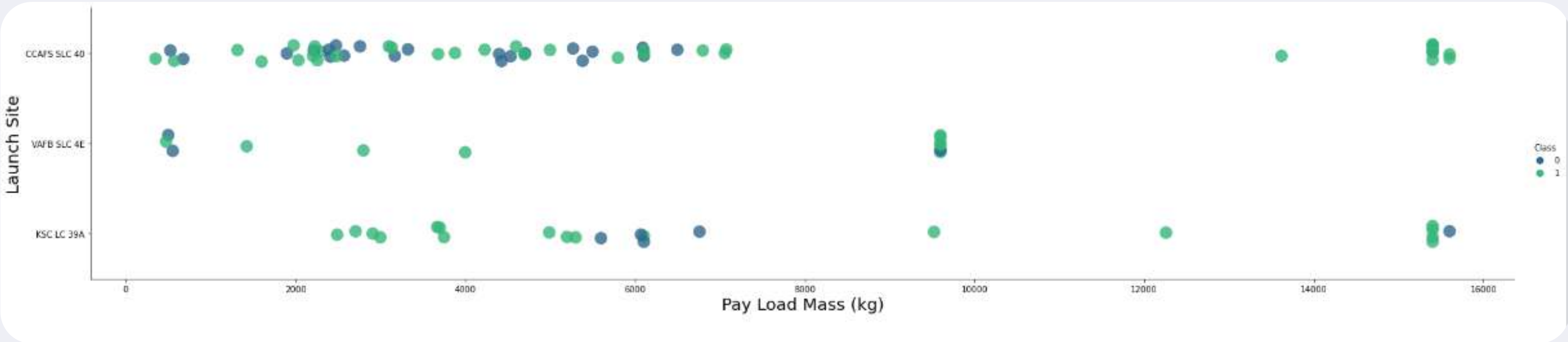
Flight Number vs. Launch Site



Green indicates successful launch; Blue 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

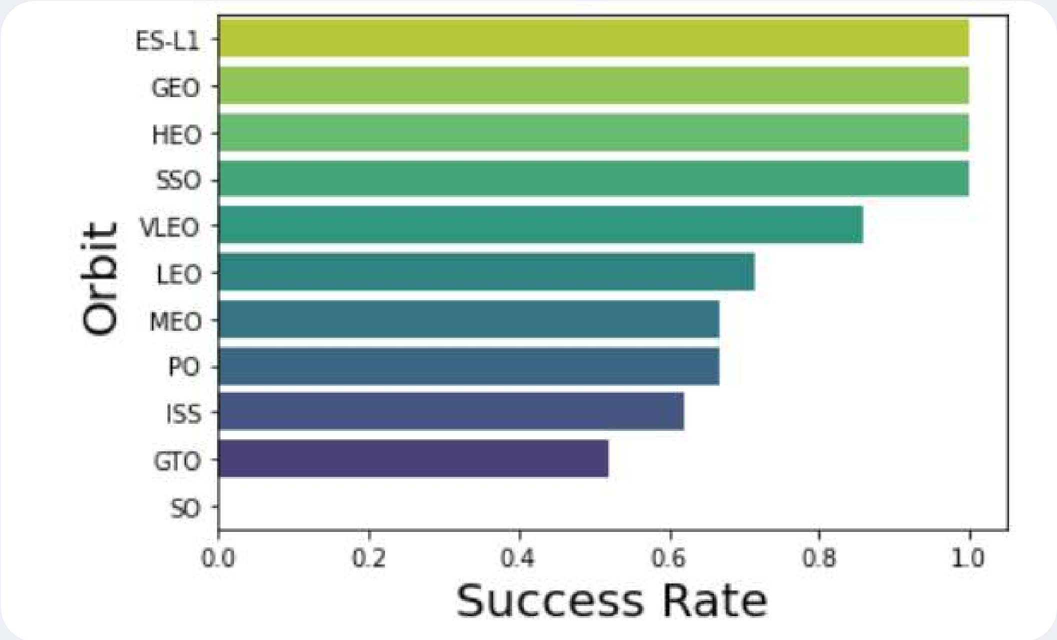


Green indicates successful launch; Purple indicates unsuccessful launch.

Payload mass appears to fall mostly between 0-6000 kg. Different launch sites also seem to use different payload mass.

Made with Gamma

Success rate vs. Orbit type



Success Rate Scale with 0 as 0%

0.6 as 60% 1 as 100%

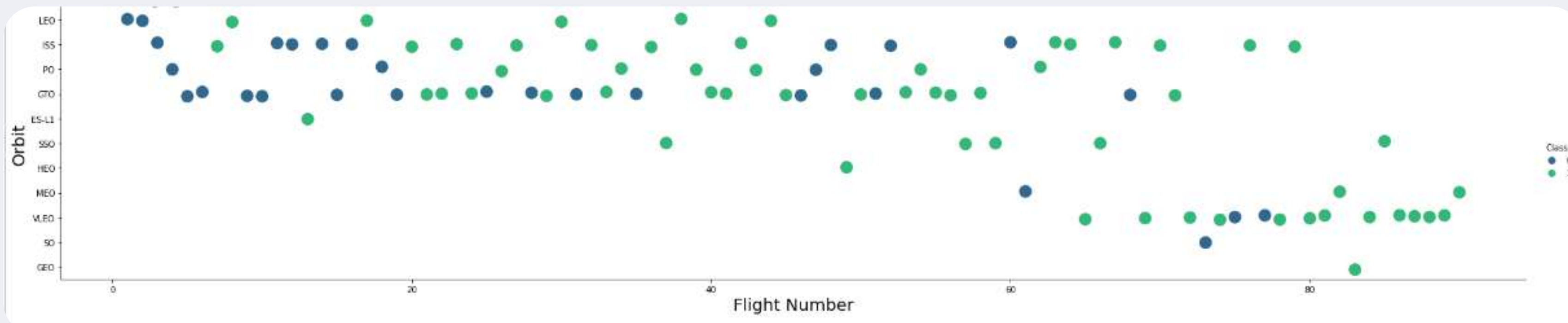
ES-L1 (1), GEO (1), HEO (1) have 100% success rate (sample sizes in parenthesis) SSO (5) has 100% success rate

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

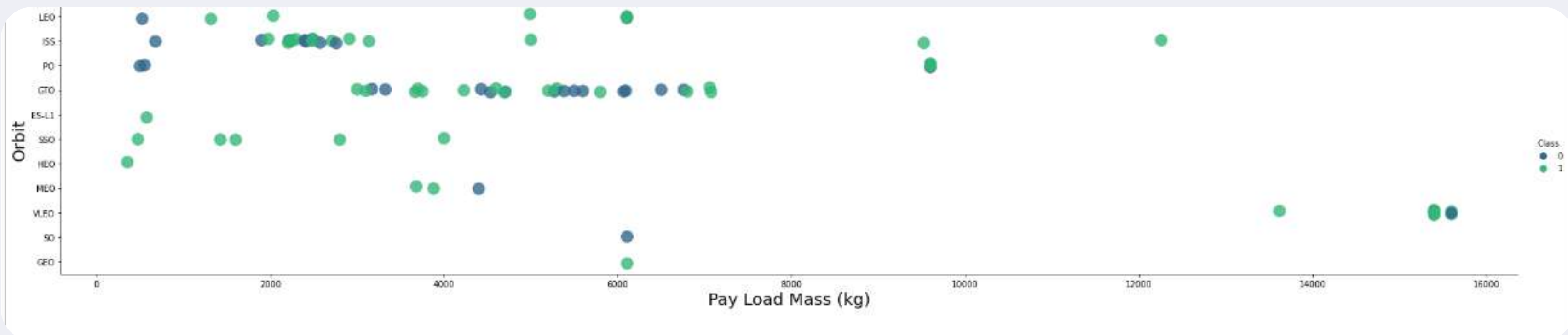


Green indicates successful launch; Blue indicates unsuccessful launch.

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



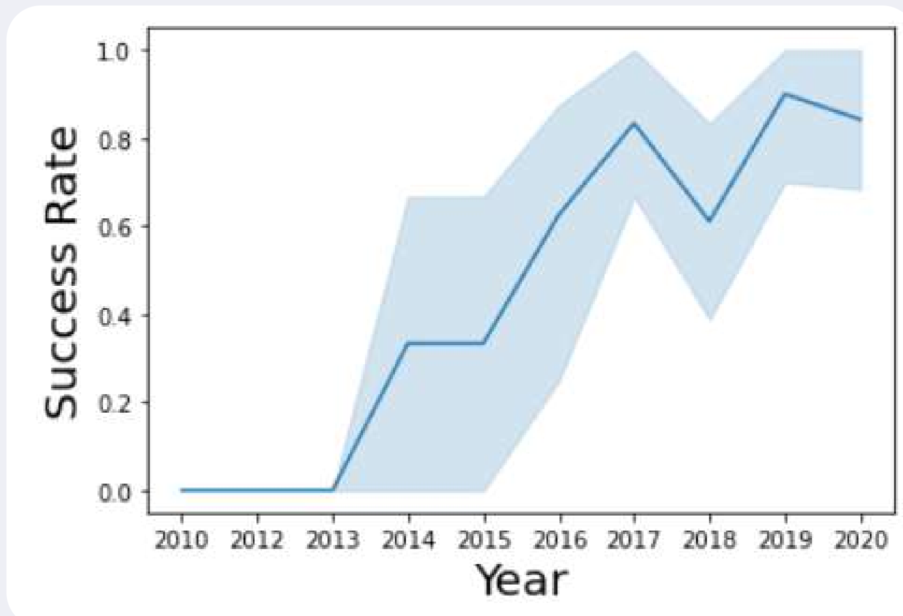
Green indicates successful launch; Purple indicates unsuccessful launch.

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



95% confidence interval (light blue shading)

Success generally increases over time since 2013 with a slight dip in 2018

Success in recent years at around 80%

EDA with SQL

EXPLORATORY DATA ANALYSIS WITH SQL DB2 INTEGRATED IN PYTHON WITH SQLALCHEMY

All Launch Site NamesTitle

In [4]:
```
%%sql
SELECT UNIQUE LAUNCH_SITE
FROM SPACEXDATASET;
```

```
 * ibm_db_sa://ftb12020:***@0c77d6f:
Done.
```

Out[4]:

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

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 Names Beginning with "CCA"

In [5]:
```
%%sql
SELECT *
FROM SPACEXDATASET
WHERE LAUNCH_SITE LIKE 'CCA%'
LIMIT 5;
```

```
 * ibm_db_sa://ftb12020:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/bludb
Done.
```

Out[5]:

DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-05-22	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

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

Total Payload Mass from NASA

```
%%sql
SELECT SUM(PAYLOAD_MASS__KG_) AS SUM_PAYLOAD_MASS_KG
FROM SPACEXDATASET
WHERE CUSTOMER = 'NASA (CRS)';
```

```
 * ibm_db_sa://ftb12020:***@0c77d6f2-5da9-48a9-81f8-86
Done.
```

sum_payload_mass_kg
45596

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).

Average Payload Mass by F9 v1.1

```
%%sql
SELECT AVG(PAYLOAD_MASS__KG_) AS AVG_PAYLOAD_MASS_KG
FROM SPACEXDATASET
WHERE booster_version = 'F9 v1.1'
```

```
 * ibm_db_sa://ftb12020:***@0c77d6f2-5da9-48a9-81f8-8
Done.
```

avg_payload_mass_kg
2928

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

First Successful Ground Pad Landing Date

```
%%sql
SELECT MIN(DATE) AS FIRST_SUCCESS
FROM SPACEXDATASET
WHERE landing__outcome = 'Success (ground pad)';
```

```
 * ibm_db_sa://ftb12020:***@0c77d6f2-5da9-48a9-81
Done.
```

first_success
2015-12-22

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.

Successful Drone Ship Landing with Payload Between 4000 and 6000

```
%%sql
SELECT booster_version
FROM SPACEXDATASET
WHERE landing__outcome = 'Success (drone ship)' AND payload_mass__kg_ BETWEEN 4001 AND 5999;
```

```
 * ibm_db_sa://ftb12020:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.database
Done.
```

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

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

Total Number of Each Mission Outcome

```
%%sql
SELECT mission_outcome, COUNT(*) AS no_outcome
FROM SPACEXDATASET
GROUP BY mission_outcome;
```

```
 * ibm_db_sa://ftb12020:***@0c77d6f2-5da9-48a9-
Done.
```

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

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.

Boosters that Carried Maximum Payload

```
%%sql
SELECT booster_version, PAYLOAD_MASS__KG_
FROM SPACEXDATASET
WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXDATASET);
```

```
 * ibm_db_sa://ftb12020:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1
Done.
```

booster_version	payload_mass__kg_
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

This query returns the booster versions that carried the highest payload mass of 15600 kg.

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.

2015 Failed Drone Ship Landing Records

```
%%sql
SELECT MONTHNAME(DATE) AS MONTH, landing__outcome, booster_version, PAYLOAD_MASS__KG_, launch_site
FROM SPACEXDATASET
WHERE landing__outcome = 'Failure (drone ship)' AND YEAR(DATE) = 2015;
```

```
 * ibm_db_sa://ftb12020:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.app
Done.
```

MONTH	landing__outcome	booster_version	payload_mass__kg_	launch_site
January	Failure (drone ship)	F9 v1.1 B1012	2395	CCAFS LC-40
April	Failure (drone ship)	F9 v1.1 B1015	1898	CCAFS LC-40

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.

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

```
%%sql
SELECT landing__outcome, COUNT(*) AS no_outcome
FROM SPACEXDATASET
WHERE landing__outcome LIKE 'Succes%' AND DATE BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY landing__outcome
ORDER BY no_outcome DESC;

 * ibm_db_sa://ftb12020:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg
Done.
```

landing__outcome	no_outcome
Success (drone ship)	5
Success (ground pad)	3

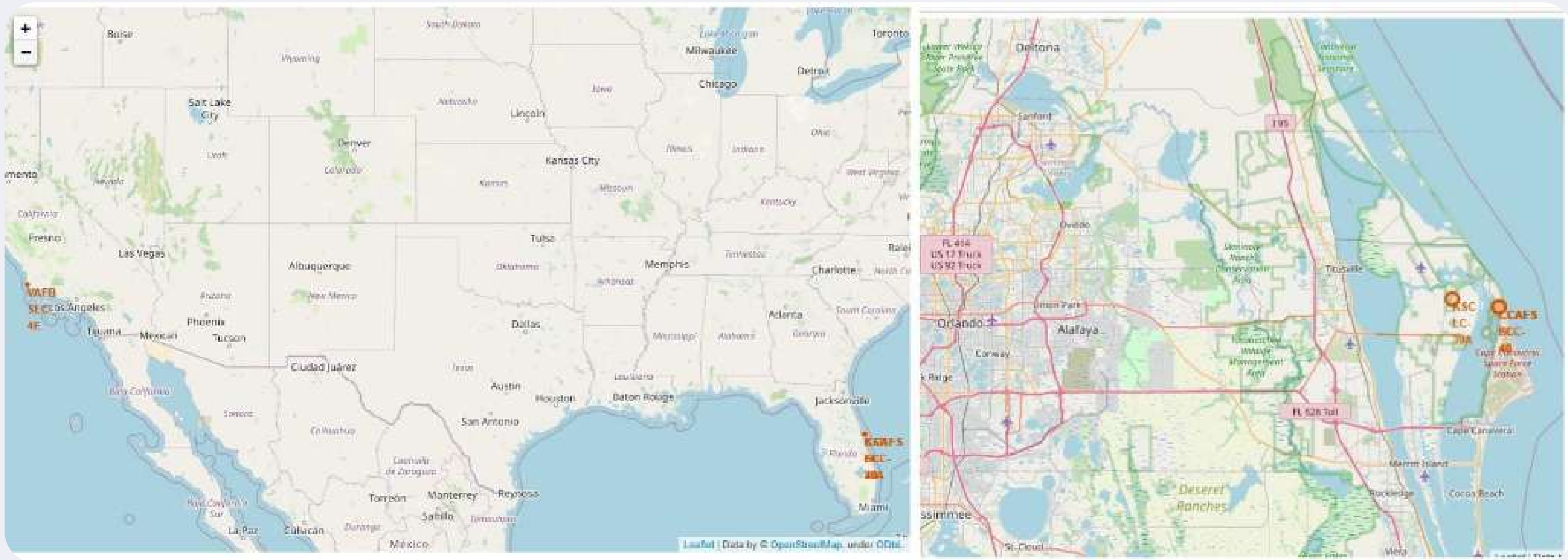
This query returns a list of successful landings and between 2010-06-04 and 2017-03-20 inclusively.

There are two types of successful landing outcomes: drone ship and ground pad landings.

There were 8 successful landings in total during this time period

Interactive Map with Folium

Launch Site Locations



The left map shows all launch sites relative US map. The right map shows the two Florida launch sites since they are very close to each other. All launch sites are near the ocean.

Color-Coded Launch Markers

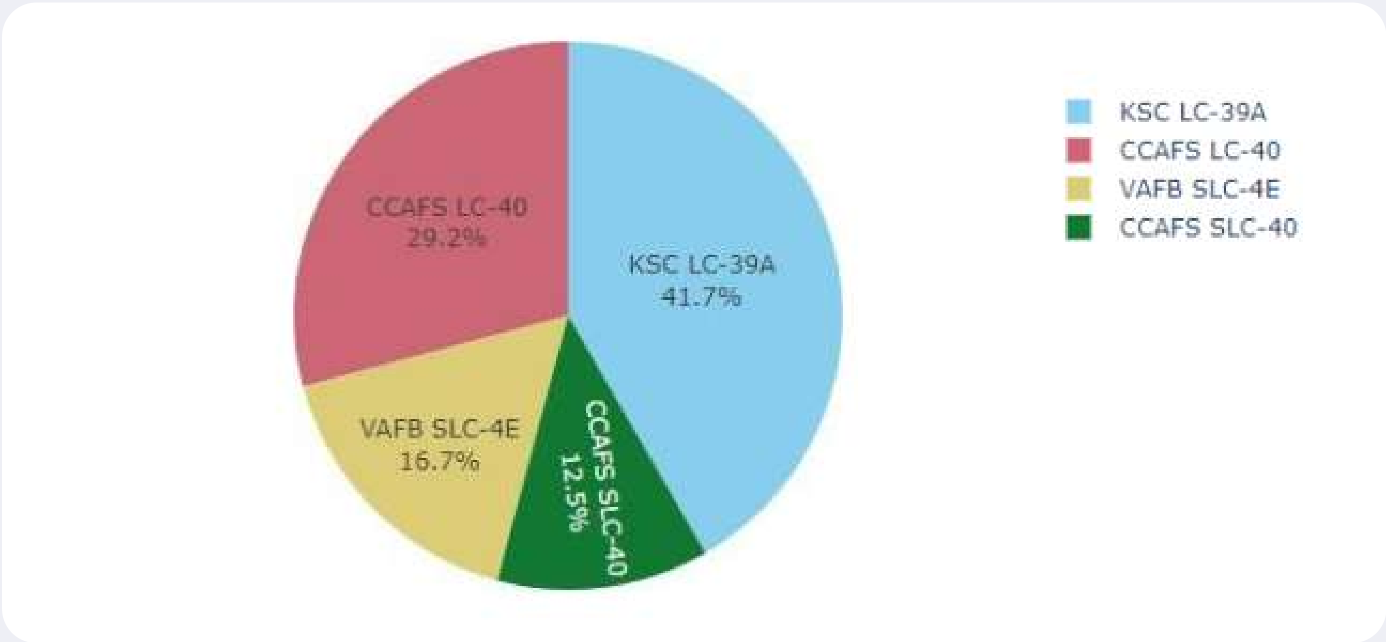
Clusters on Folium map can be clicked on to display each successful landing (green icon) and failed landing (red icon). In this example VAFB SLC-4E shows 4 successful landings and 6 failed landings.

Key Location Proximities

Using KSC LC-39A as an example, launch sites are very close to railways for large part and supply transportation. Launch sites are close to highways for human and supply transport. Launch sites are also close to coasts and relatively far from cities so that launch failures can land in the sea to avoid rockets falling on densely populated areas.

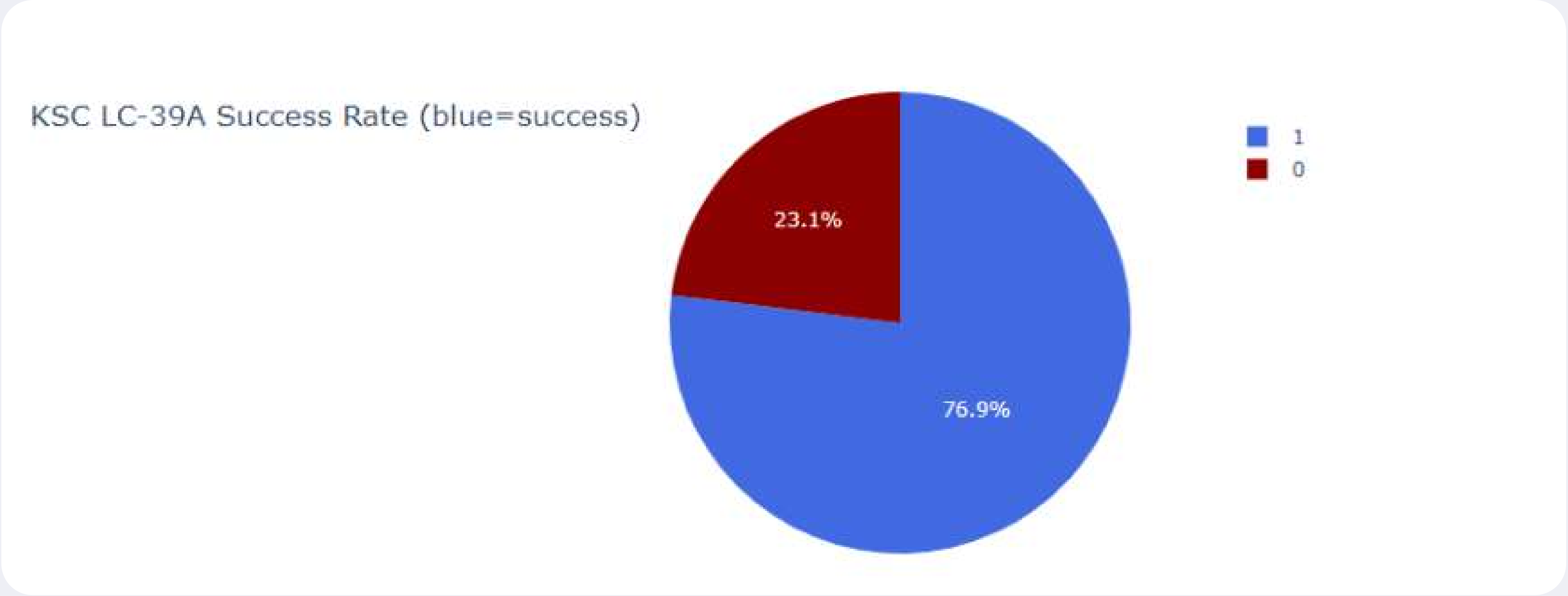
Build a Dashboard with Plotly Dash

Successful Launches Across Launch Sites



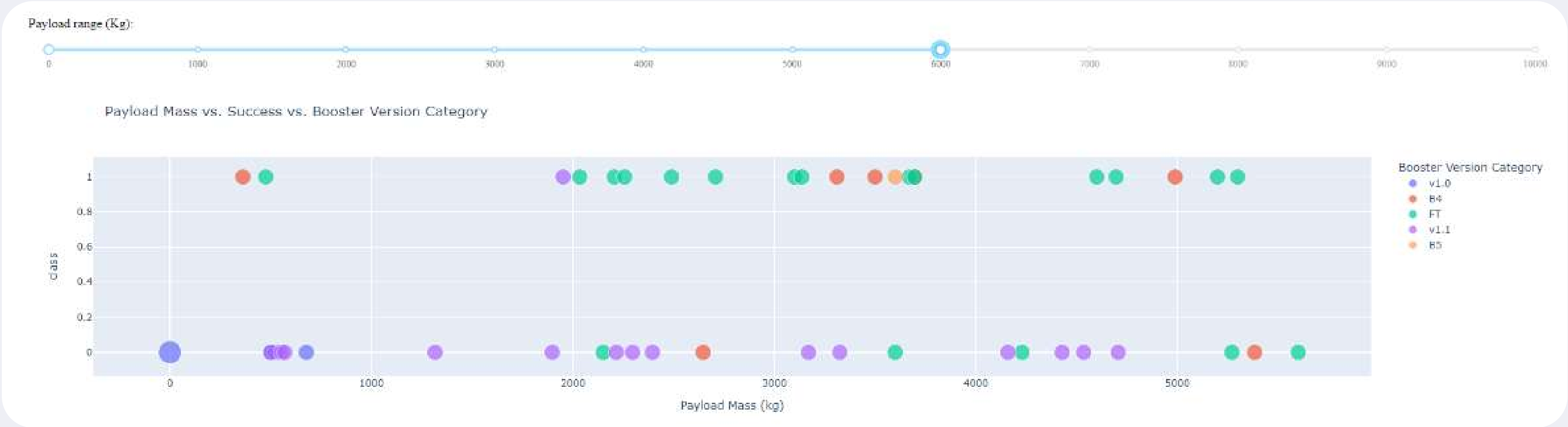
This is the distribution of successful landings across all launch sites. CCAFS LC-40 is the old name of CCAFS SLC-40 so CCAFS and KSC have the same amount of successful landings, but a majority of the successful landings where performed before the name change. VAFB has the smallest share of successful landings. This may be due to smaller sample and increase in difficulty of launching in the west coast.

Highest Success Rate Launch Site



KSC LC-39A has the highest success rate with 10 successful landings and 3 failed landings.

Payload Mass vs. Success vs. Booster Version Category



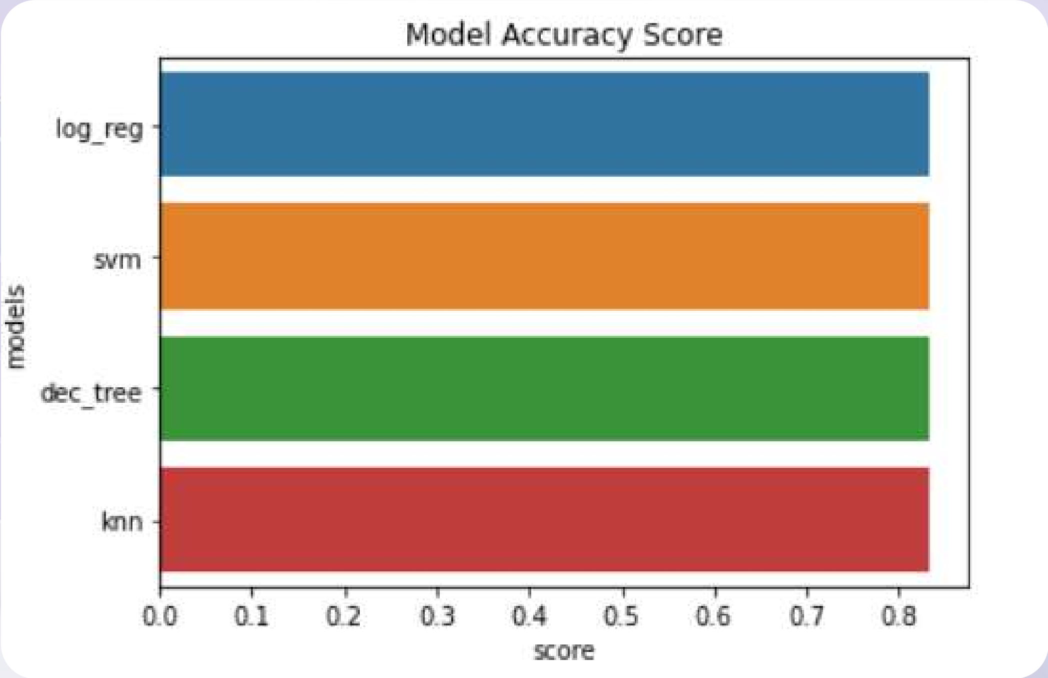
Plotly dashboard has a Payload range selector. However, this is set from 0-10000 instead of the max Payload of 15600. Class indicates 1 for successful landing and 0 for failure. Scatter plot also accounts for booster version category in color and number of launches in point size. In this particular range of 0-6000, interestingly there are two failed landings with payloads of zero kg.

Predictive Analysis (Classification)

GRIDSEARCHCV(CV=10) ON LOGISTIC REGRESSION, SVM, DECISION

TREE,ANDKNN

Classification Accuracy

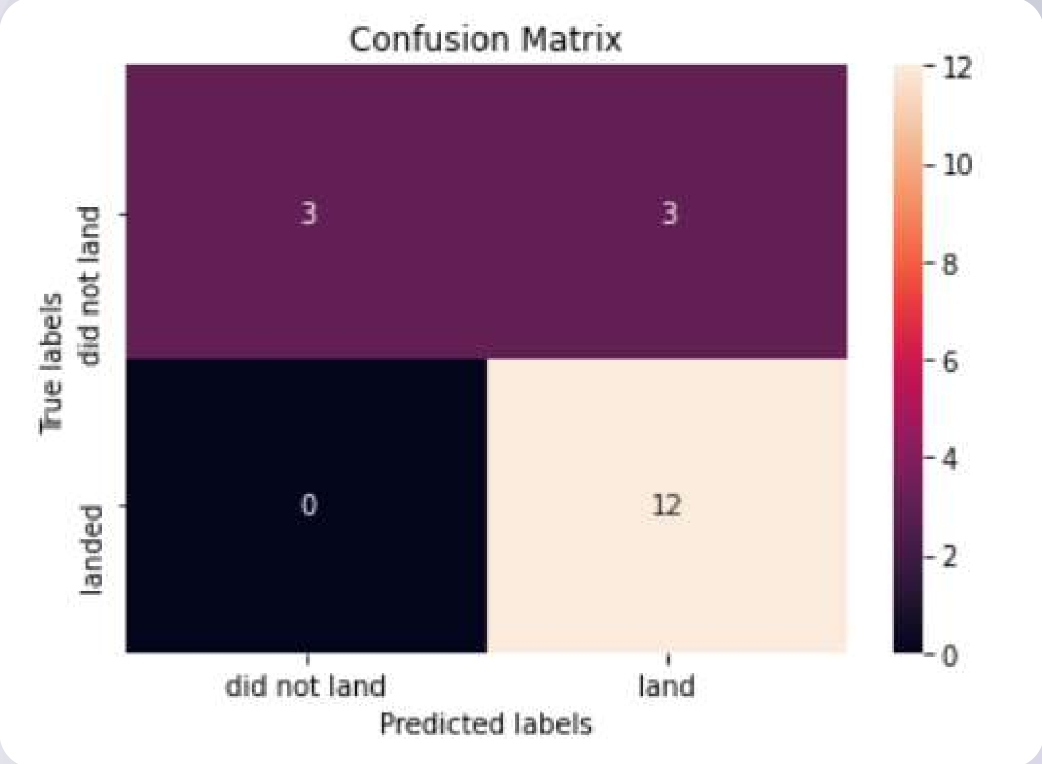


All models had virtually the same accuracy on the test set at 83.33% accuracy. It should be noted that test size is small at only sample size of 18.

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



Correct predictions are on a diagonal from top left to bottom right.

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.

Key Findings and Solutions

Quantum Computing

Exploring the potential of quantum computing to develop more sophisticated space mission simulations and optimization algorithms.

Reusable Rockets

Designing and refining reusable rocket technology to significantly reduce the cost of space missions.

AI-Driven Automation

Leveraging artificial intelligence to automate various processes involved in space exploration and data analysis.

Global Satellite Network

Building a global satellite network to enhance communication and data transmission capabilities.

CONCLUSION

Our objective was to develop a cutting-edge machine learning model for SpaceY, aiming to compete with SpaceX. The primary focus of the model is to forecast the successful landing of Stage 1 rockets, potentially saving up to $100 million USD per launch.

To achieve this, we leveraged data from a publicly accessible SpaceX API and conducted web scraping on the SpaceX Wikipedia page. This diverse dataset was meticulously labeled, and the information was stored in a robust DB2 SQL database. Additionally, we designed an intuitive dashboard for effective data visualization.

Our machine learning model, boasting an impressive accuracy rate of 83%, serves as a powerful tool for Allon Mask and the SpaceY team. With this model, they can make informed predictions regarding the likelihood of a successful Stage 1 landing before the launch, facilitating crucial decisions on whether to proceed with the launch.

While the current model delivers a commendable level of accuracy, we recommend further data collection efforts to refine and optimize the machine learning model. Gathering additional data will not only contribute to a more comprehensive understanding of the underlying patterns but also enhance the accuracy of predictions, ensuring SpaceY remains at the forefront of space exploration technology.

Conclusion and Next Steps

Successful Collaboration

The IBM SpaceX Capstone Project exemplifies the power of collaboration between tech giants and space pioneers.

Continued Innovation

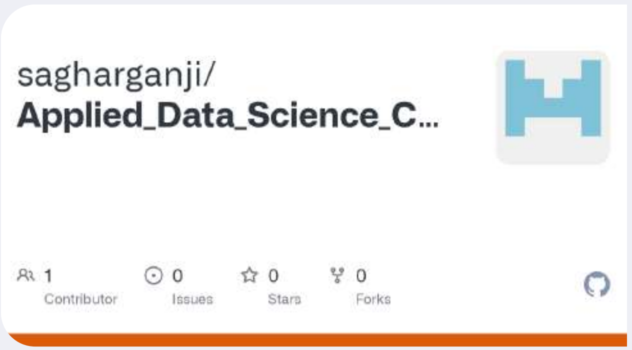
Continued research, development, and innovation will drive the future of space exploration and commercial space endeavors.

Inspiring Future Generations

This project inspires the next generation of scientists and engineers to push the boundaries of what's possible in space.

APPENDIX

GitHub repository url:



GitHub

GitHub - sagharganji/Applied_Data_Science_Capstone

Contribute to sagharganji/Applied_Data_Science_Capstone development by creating an account on GitHub.

Instructors:

Instructors: 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!

We appreciate your time and attention. It has been a pleasure sharing our insights and ideas with you. If you have any further questions or would like to connect, please don't hesitate to reach out. Have a wonderful day!