Data Science Capstone Project

Ronald Tekenya

16/11/2021

OUTLINE

- Executive Summary
- Introduction
- Methodology
- Results
  - Visualization – Charts
  - Dashboard
- Discussion
  - Findings & Implications
- Conclusion
- Appendix

EXECUTIVE SUMMARY

- Data Collection
- Data wrangling
- EDA
  - With data visualization
  - With SQL
- Building an interactive map with Folium
- Building a Dashboard with Plotly Dash
- Predictive analysis (Classification)

INTRODUCTION

Project Background

- New Era of space exploration
- SpaceX Falcon 9 rocket cost is $62 m , while its competitors cost $165m and more
- SpaceX reusing of its Falcon 9 rockets contributes to its low cost, however, the landing needs to be successful.
- SpaceY wants to compete with SpaceX using this information

Problem

- SpaceY tasked us to use machine learning models to predict successful stage 1 recovery using SpaceX data

METHODOLOGY

- Data collection methodology:
  - Combined data from SpaceX public API
  - SpaceX Wikipedia page (Web Scrapping)
- Perform data wrangling
- Perform data exploratory analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using
  - Folium
  - Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, and evaluate classification models

Methodology

Overview of data collection, wrangling, visualization, dashboard and model training

Data Collection Overview

- Data for the project was collected by the combination of SpaceX REST API and web scrapping SpaceX's Wikipedia entry using python BeautifulSoup library into a workable data-frame as follows;

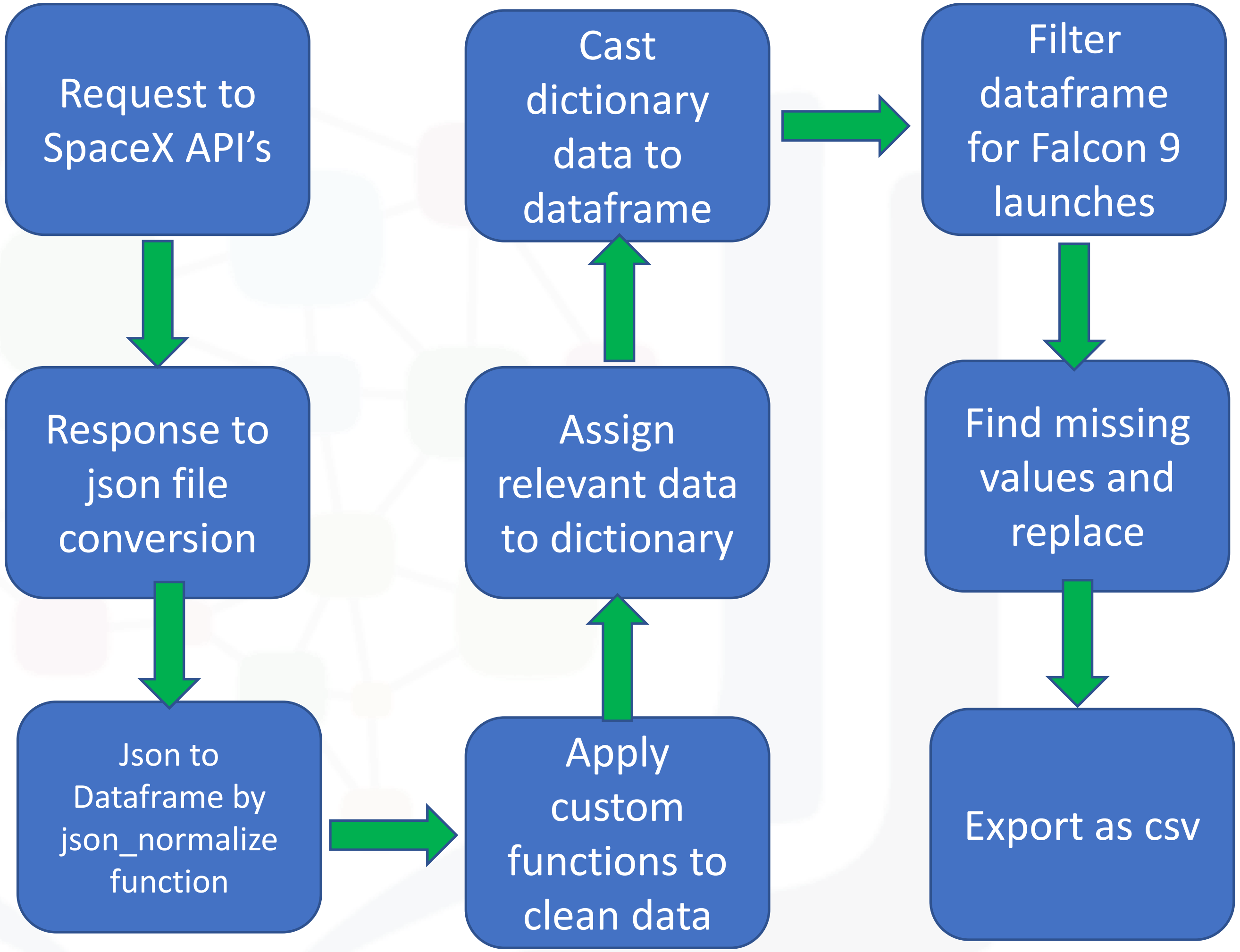
1. SpaceX API

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

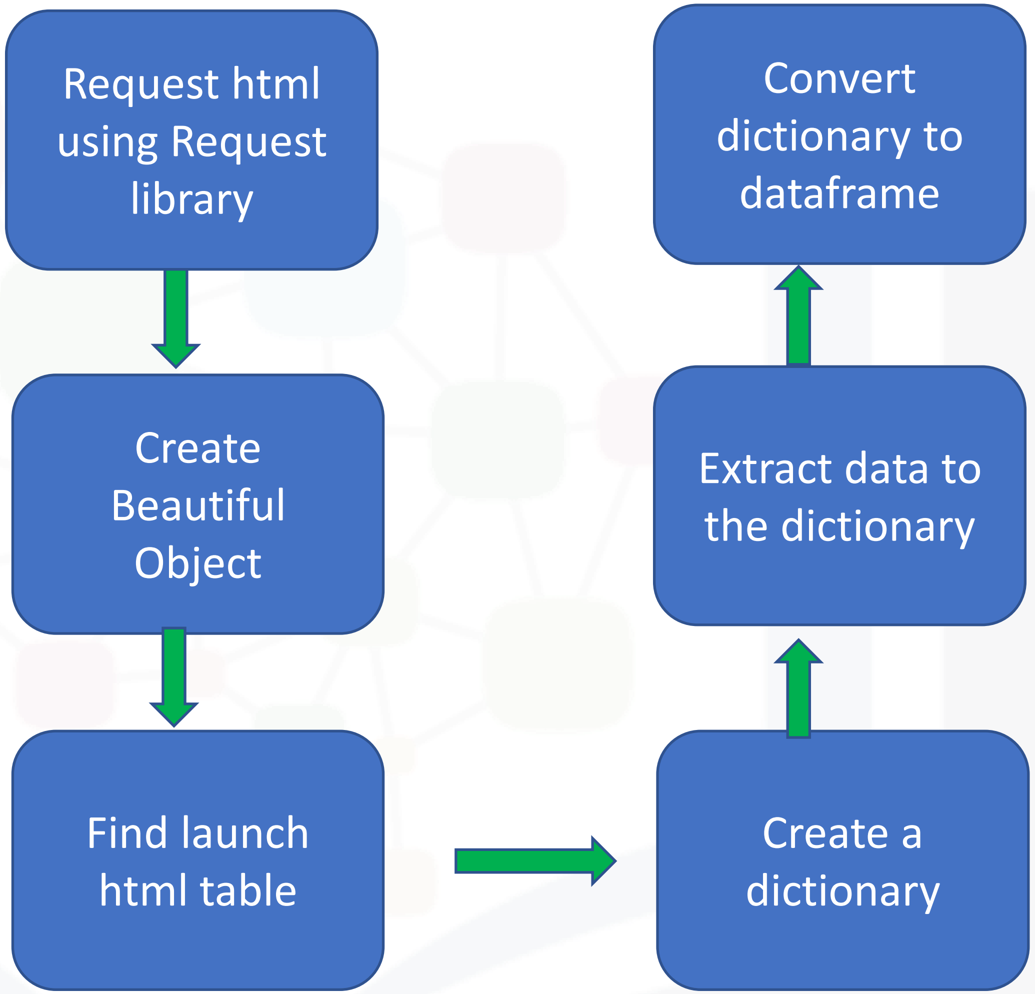
2. Wikipedia Web scrapped data

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

Data Collection – SpaceX API



Data Collection – Web Scrapping



Data Wrangling

- Tuning data to numeric values(Training labels) for success and failure to land, as such a training model can be created from such:

Mapping:

- True ASDS, True RTLS & True Ocean -> as ‘1’
- None None, False ASDS, None ASD, False Ocean, False RTLS -> as ‘0’

EDA with Data Visualization

Scatter Graphs Drawn:

- Flight Number vs Payload Mass
- Flight Number vs Launch Site
- Payload vs Launch Site
- Orbit vs Flight Number
- Payload vs Orbit Type
- Orbit vs Payload Mass

Line Graphs Drawn:

Success Rate vs Year

Bar Graphs Drawn:

Mean vs Orbit

Graphs Purposes:

- To compare relationships between variables and decide which regression to be used (linear or non-linear)

EDA with SQL

- Use IBM Db2 database to load out dataset, thereafter integrated with Python
- Queries were made to filter, organize and sort our data for better understanding
- Queried Info:
  - Unique launch site names, mission outcomes, payload mass by boosters, total number of successful landing outcomes, booster versions etc

Building an interactive map with Folium

- Folium was used to visualize the launch data into an interactive map
- Latitude and longitude coordinates of each launch site were used to add Circle Marker around each launch site
- Green -> successful launch_outcomes -> class 1
- Red -> failure launch_outcomes -> class 0
- This allows us to understand why launch sites may be located where they are.
- Enable us to visualize successful landings relative to the locations

Build interactive dashboard with Plotly Dash

- This dashboard was built with Flask and Dash for easy interactive
- The Dashboard includes a scatter plot and pie chart

Scatter Plot:

- To show the relationship between two variables
- To show relationship between Outcome and Payload Mass and booster version category

Pie Chart

- To show distribution of successful landings across all launch sites and can be selected to show individual launch site success rates.
- To display relative proportions of multiple classes of data

Predictive Analysis(Classification)

Building Model:

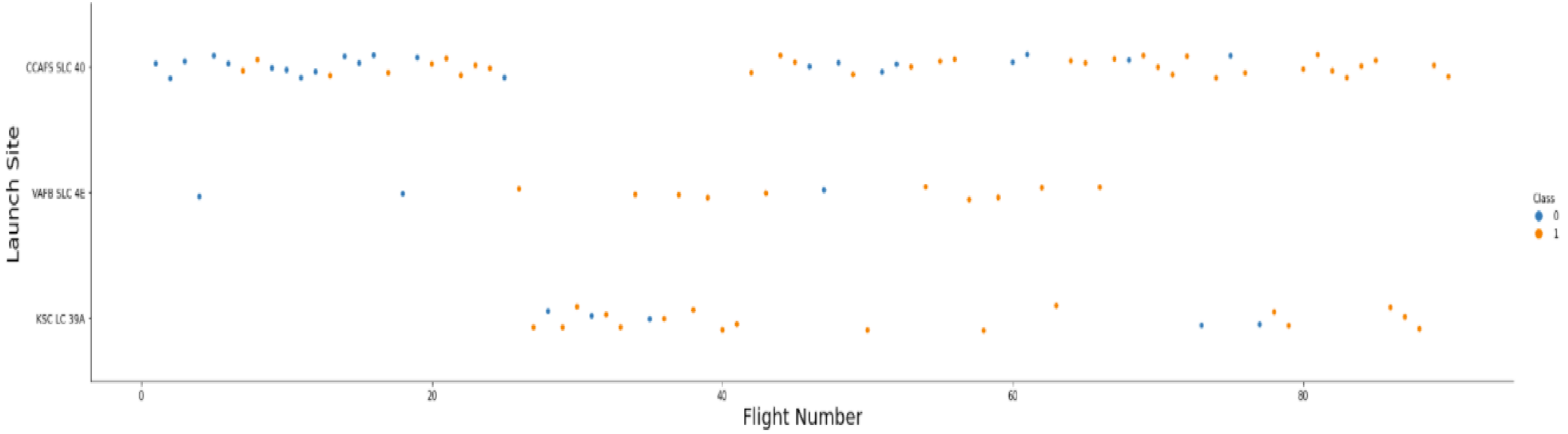
1. Split label column ‘Class’ from dataset
2. Fit and Transform Features using Standard Scaler
3. Train_test_split data
4. GridSearchCV (cv=10) to find optimal parameters
5. Use GridSearchCV on LogReg, SVM, Decision Tree, and KNN models
6. Score models on split test set
7. Confusion Matrix for all models
8. Barplot to compare scores of models

Results

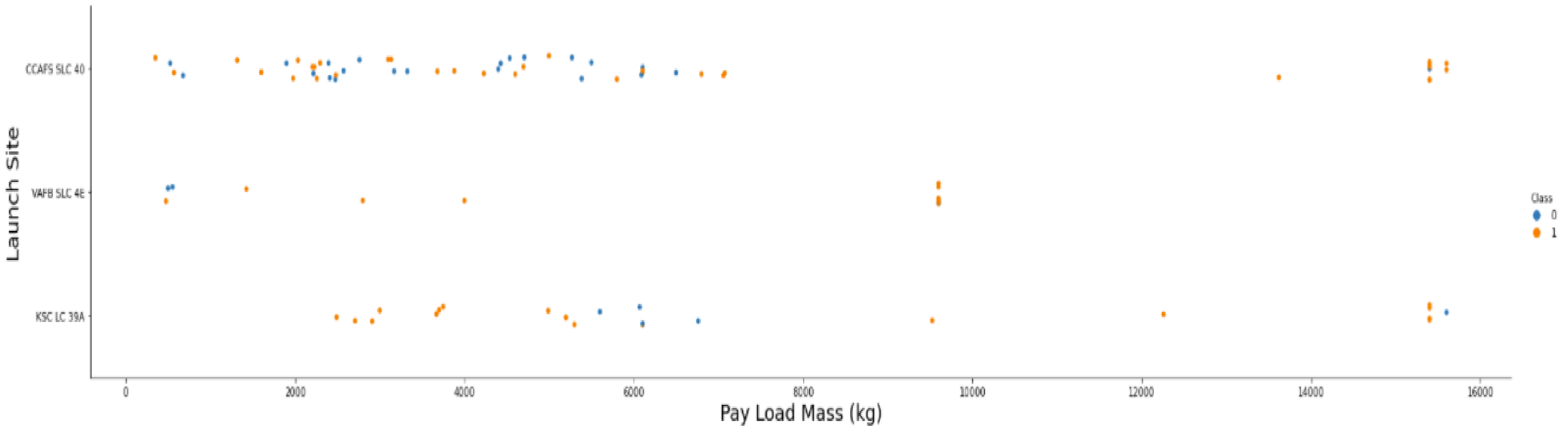
- Exploratory Data Analysis Results
- Interactive analytics demo in screenshoots
- Predictive analysis results

EDA with Visualization

Flight number vs Launch Site

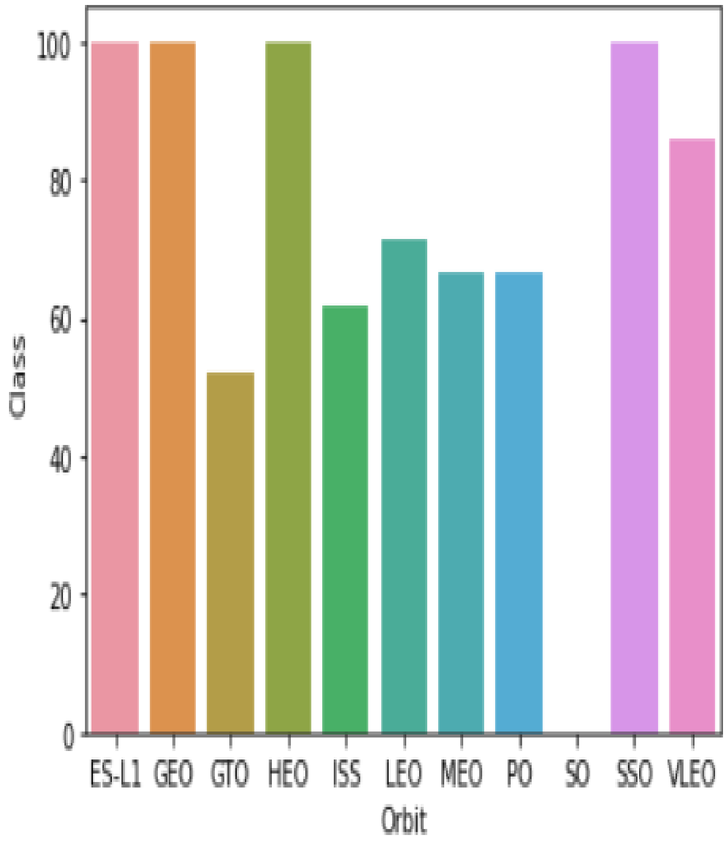


Payload vs. Launch Site



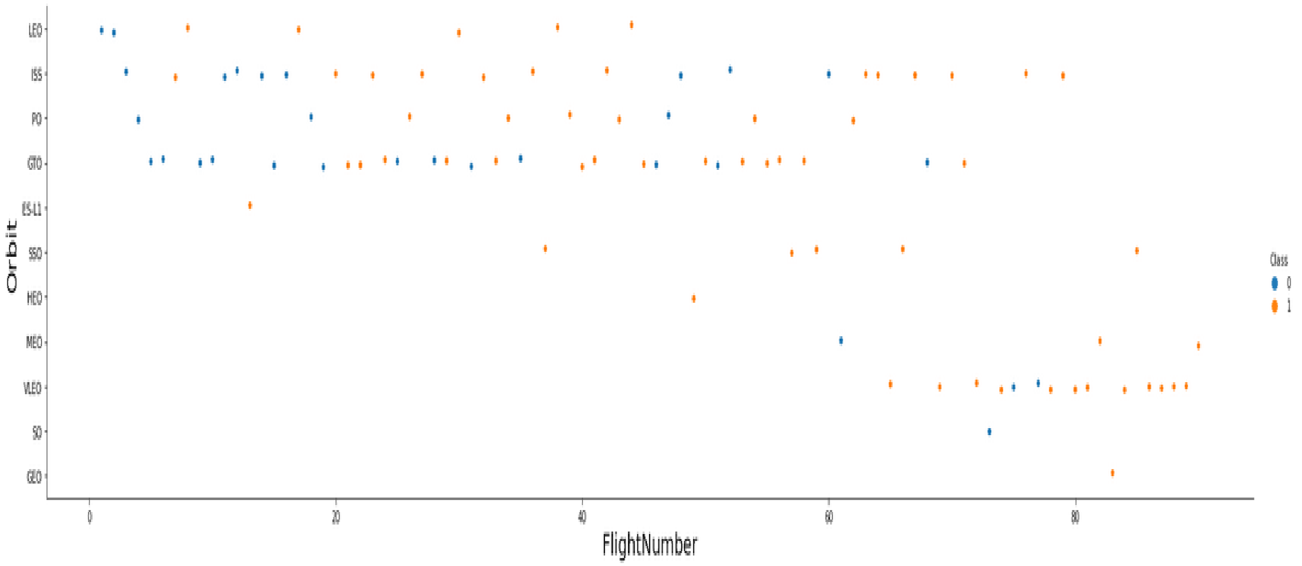
- Payload mass appears to fall mostly between 0-6000kg
- The greater the Pay Load Mass, the success the rate increase

Success Rate vs. Orbit Type



Orbits ES-L1, GEO, HEO and SSO have 100% success rate
 VLEO914) has decent success rate and attemps

Flight Number vs. Orbit Type

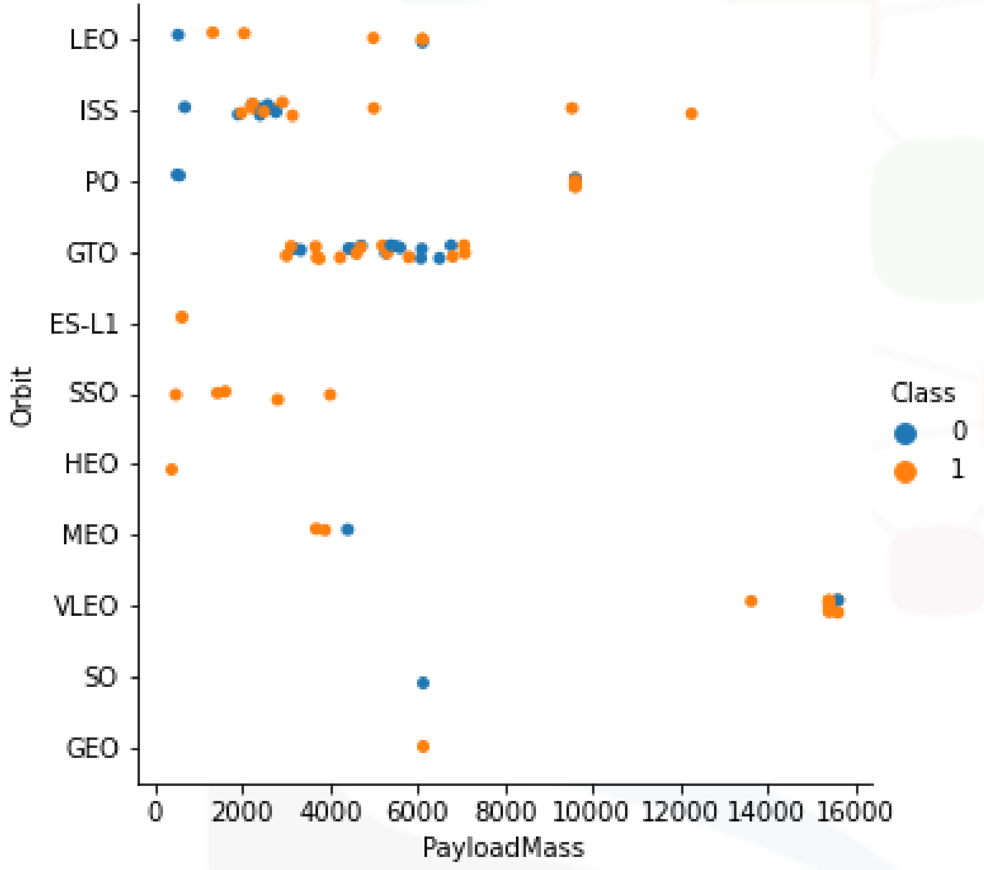


In the LEO orbit, the success is related to the number of flights

Launch Orbit preferences changed over Flight Number

SpaceX appears to perform better in lower orbits or Sun-synchronous orbits

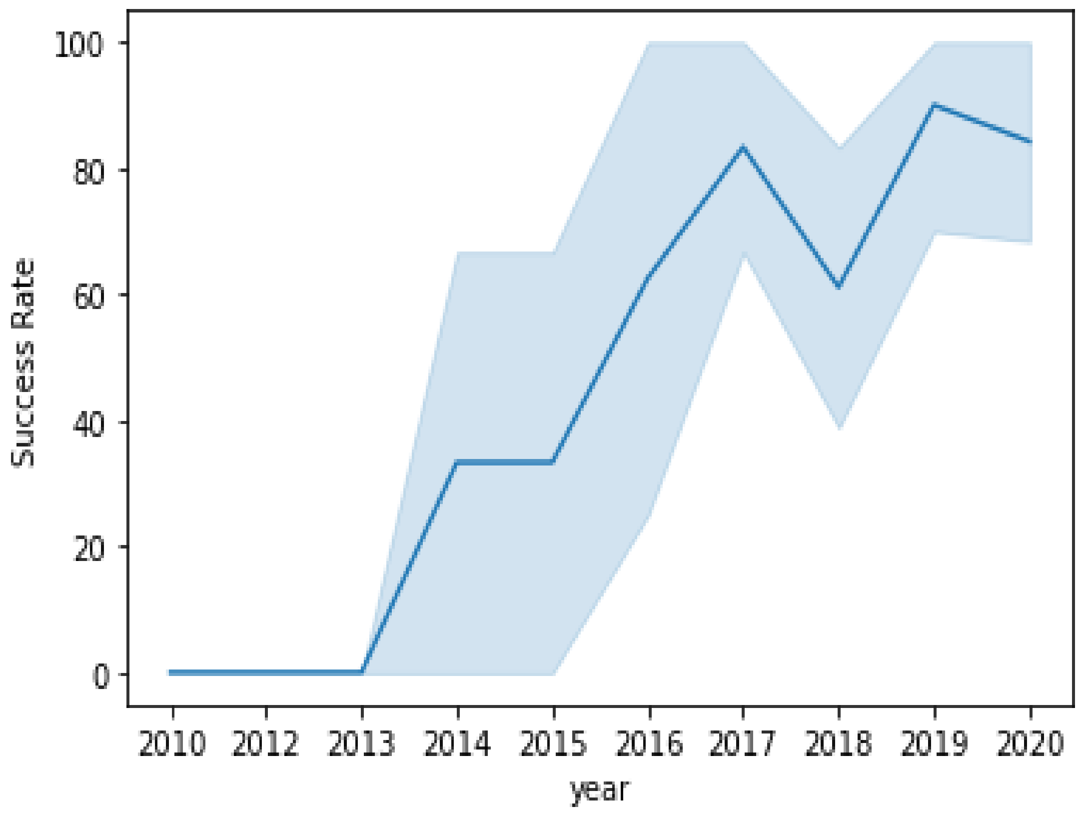
Payload vs. Orbit Type



- The heavy payloads have a negative influence on GTO orbits. Instead, for LEO and ISS orbits have a positive influence
- LEO and SSO seem to have relatively low payload mass

Launch Success Yearly Trend

<AxesSubplot:xlabel='year', ylabel='Success Rate'>



- High success rate since 2013 as indicated by the blue shading
- Slight dip in success rate in 2018

EDA with SQL

Exploratory data analysis with SQL DB2 integrated in Python with SQLachemy

IBM Developer

SKILLS NETWORK

All Launch Site Names

- Unique Launch sites
  - CCAFS LC-40
  - CCAFS SLC-40
  - KSC LC-39A
  - VAFB SCL-4E
- SQL quesry
  - SELECT DISTINCT LAUNCH_SITE FROM SPACEXTBL
  - All unique values to be returned

Launch Site Names Begin with ‘CCA’

- SELECT*FROM SPACEXTBL WHERE LAUNCH_SITE LIKE ‘CCA’ LIMIT 5;
- First 5 entries will be return as per above query

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

Total Payload Mass

- This query sums total payload mass in kg where NASA was the customer
  - SELECT SUM(payload_mass__kg_)
    FROM SPACEXTBL
    WHERE CUSTOMER = 'NASA (CRS)';
  - 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'
```

- Average payload mass carried by booster version F9 V1.1
- This query calculates the average payload mass or launches which used booster version F9 v1.1

First Successful Ground Landing Date

- First successful landing outcome on ground pad occurred on 2015-12-22
- SQL query:
  - SELECT MIN(DATE)
    FROM SPACEXTBL
    WHERE LANDING__OUTCOME = 'Success (ground pad)’;
  - The MIN function returns the minimum value (in this case for DATE column) whereas the WHERE predicate filters by LANDING__OUTCOME.

Successful Drone Ship Landing with Payload

- SQL query
  - SELECT BOOSTER_VERSION FROM SPACEXTBL WHERE LANDING__OUTCOME = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000;
  - The BETWEEN operator selects values withing a given range.

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

Total Number of Each Mission Outcome

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

* ibm_db_sa:[redacted]

Done.

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

- The COUNT function counts the rows. In this case it will count the rows that satisfy certain condition, which is specified in the WHERE predicate
- 99 success vs 1 failure

Boosters Carried Maximum Payload

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

- SQL query:
  ```
  SELECT BOOSTER_VERSION
  FROM SPACEXTBL
  WHERE PAYLOAD_MASS__KG_ = (SELECT
  MAX(PAYLOAD_MASS_KG_)
  FROM SPACEXDATASET);
  ```
- This query returns the booster versions that carried the highest payload mass of 15600 kg.

2015 Launch 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;
```

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

- Two such occurrences appeared

Rank Landing Oucomes 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;
```

*
Done.

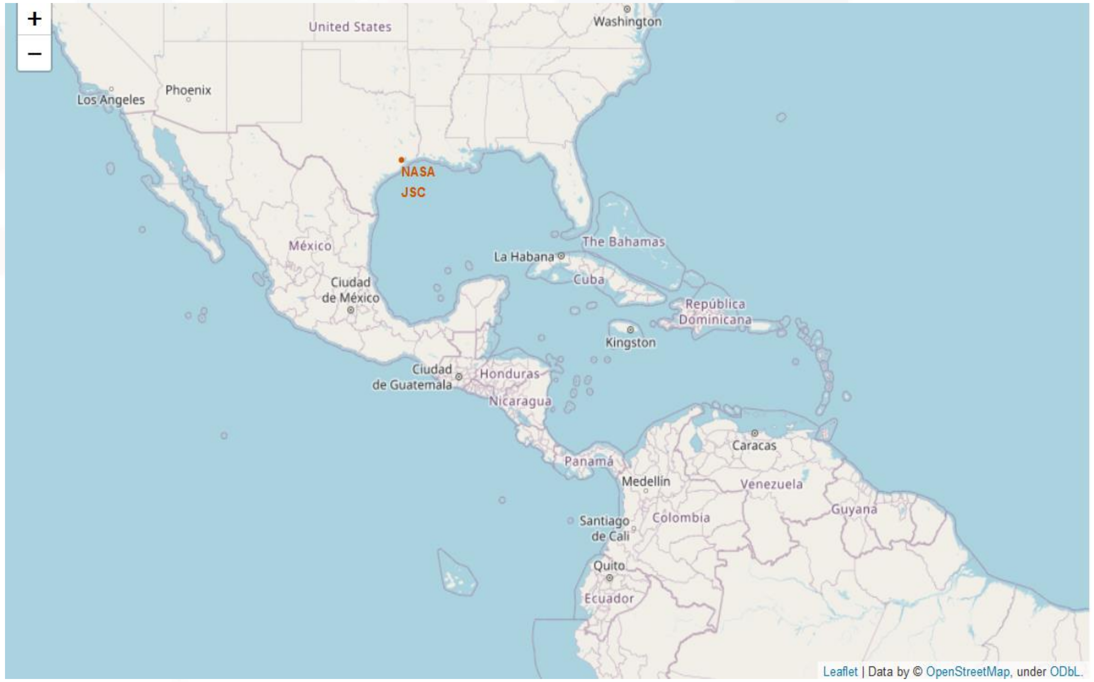
landing__outcome	no_outcome
Success (drone ship)	5
Success (ground pad)	3

- 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
- 8 successful landings in total during this time period

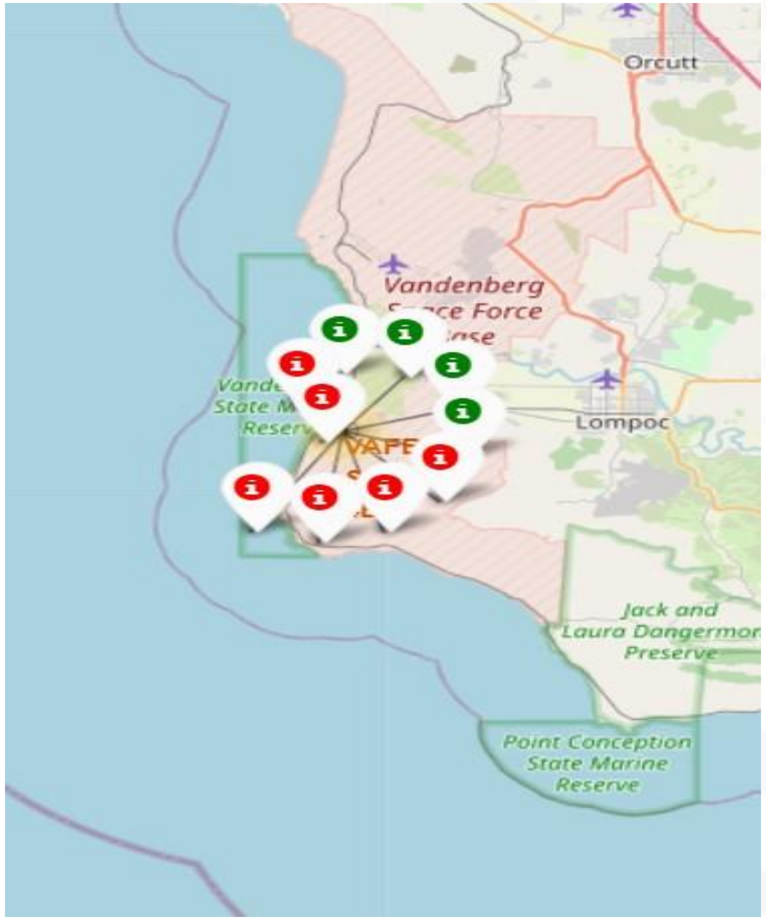
Interactive Map with Folium

Launch Sites

- A map showing all Launch sites
- Left Map-> world relative - Right Map -> US relative



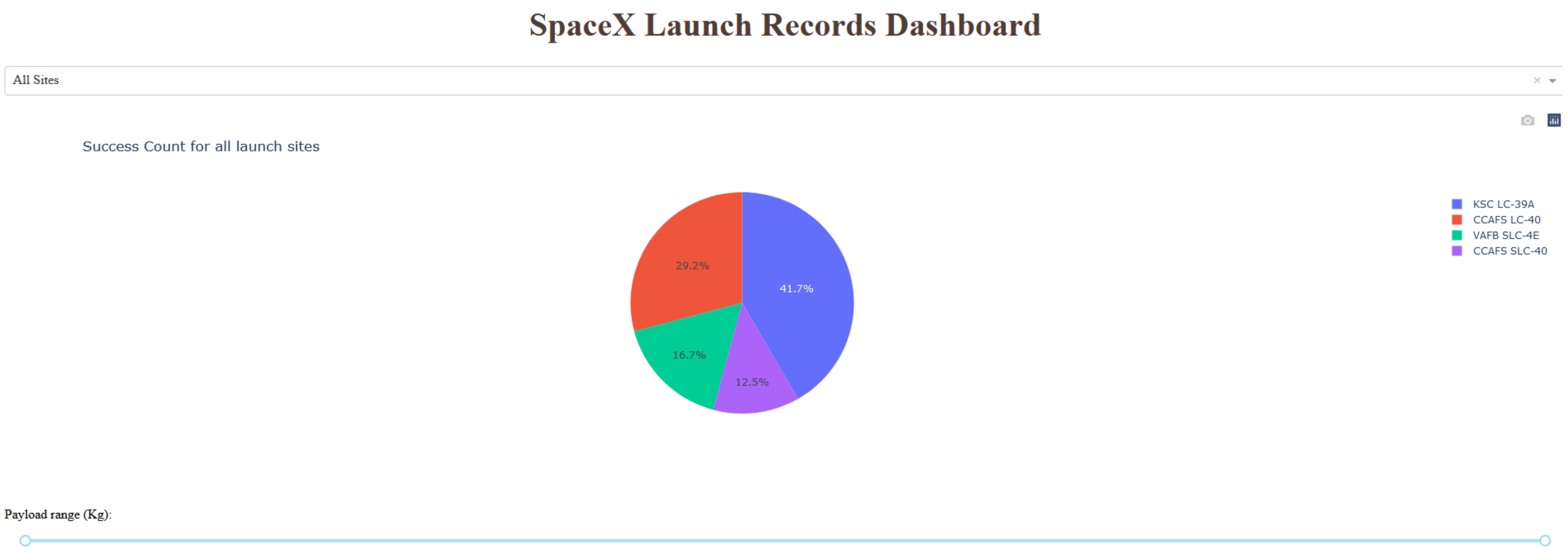
Color-Coded Launch Markers



- 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.

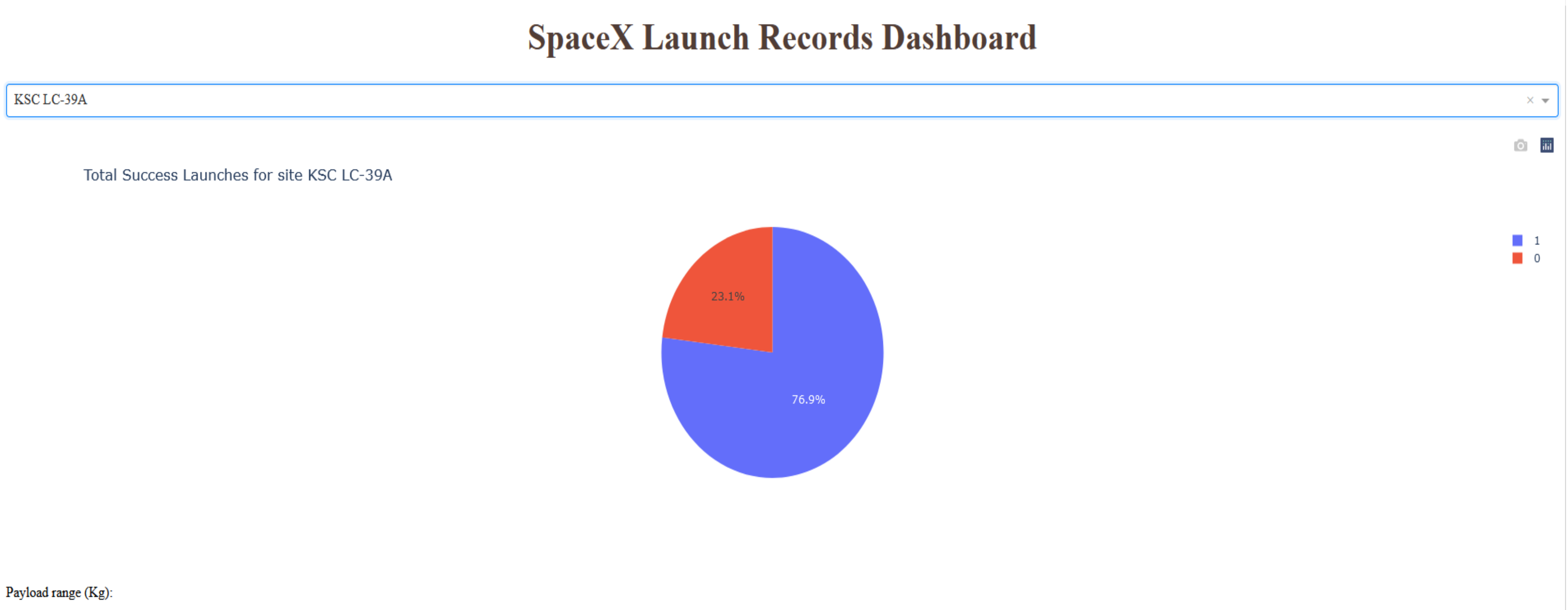
Dashboard with Plotly Dash

Total Success Launches Across all sites



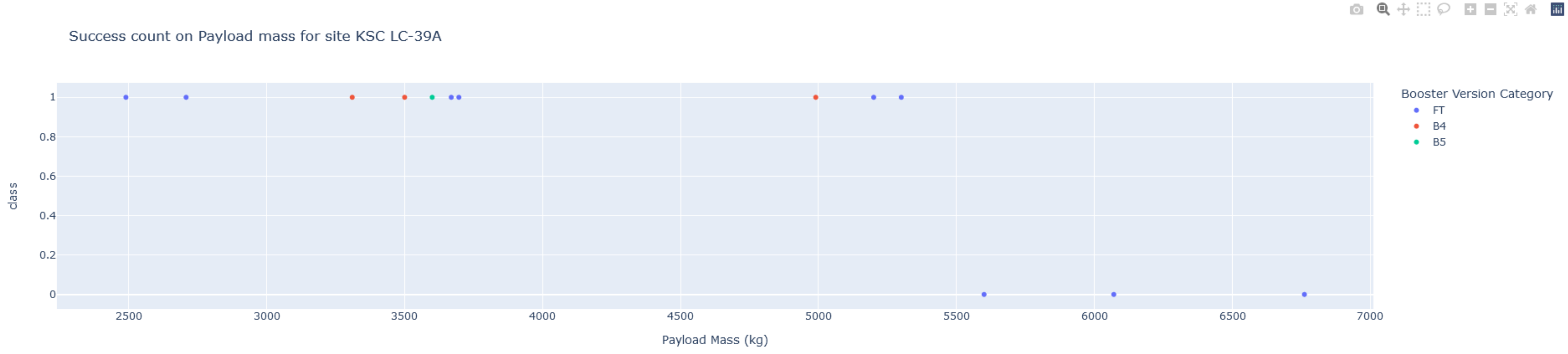
- This is the distribution of successful landings across all launch sites
- KSC LC-39A has the most successful launches from all sites

Highest success rate launch site



KSC LC-39A with 76,9% succuss rate

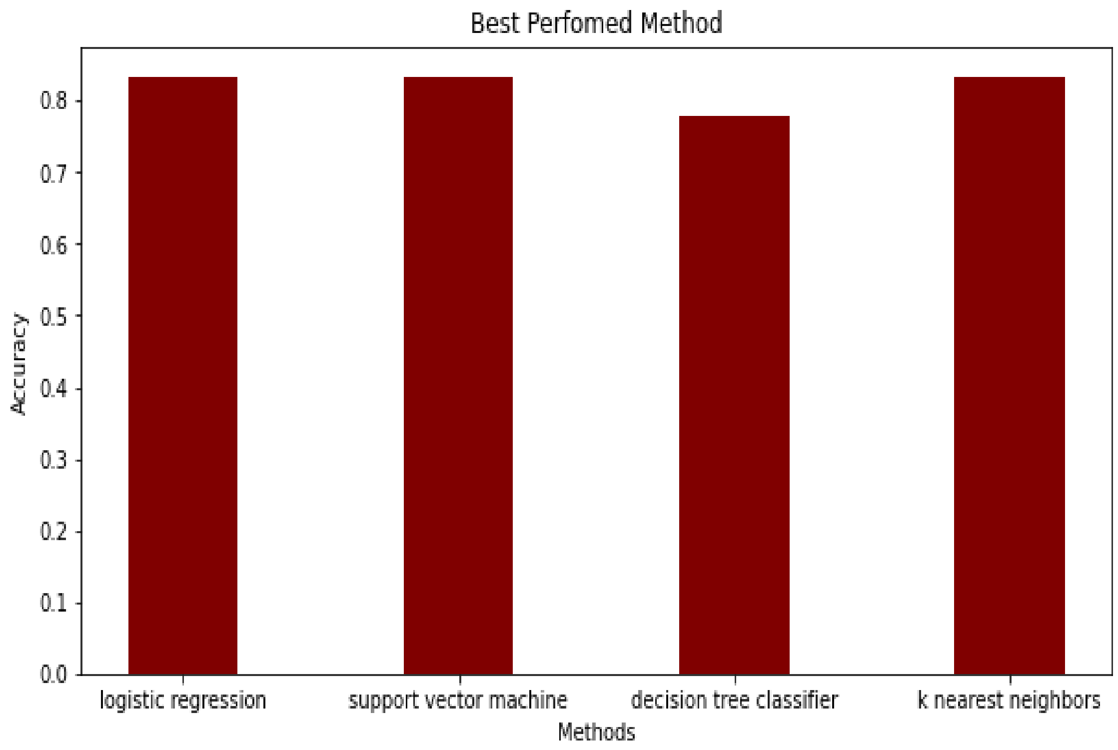
Payload Mass vs Success vs Booster version



- 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.

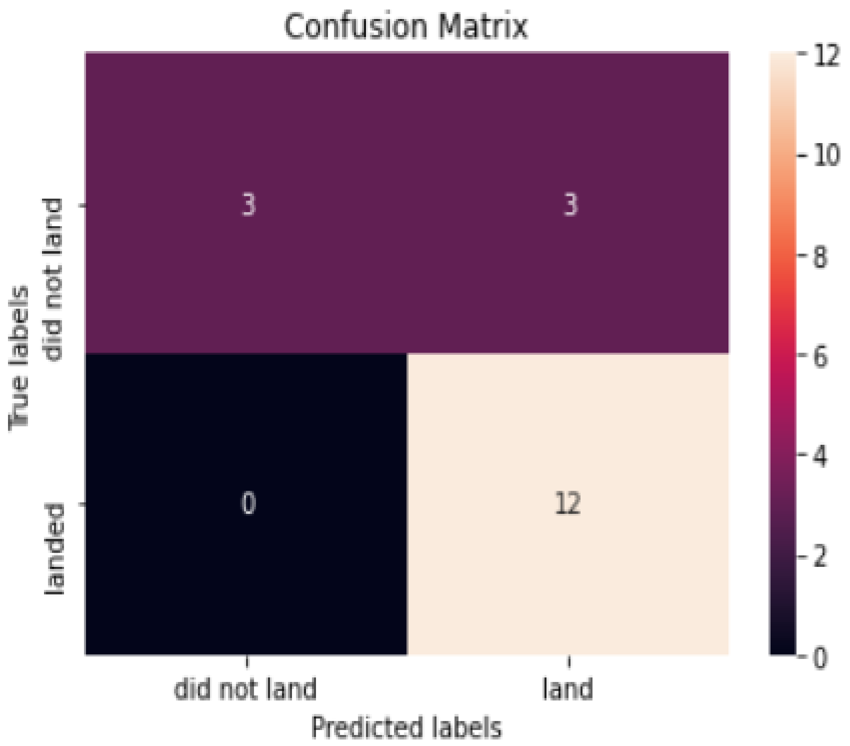
Predictive Analysis (Classification)

Classification Accuracy



- Accuracy among all is very close to each other
- Might be because of the small sample size

Confusion Matrix



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

- We were successful on developing a machine learning model for SpaceY
- Better prediction model will contribute to SpaceY saving lots of money
- The accuracy for our model was over 83%
- Of the SpaceX version, KSC LC-39A had the most successful launches from all sites
- More data will make our model better and more accurate

APPENDIX

- https://github.com/rtekenya/Applied-Data-Science-Capstone.git.