IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Amruth Pai Thukaram

https://github.com/amruthpai123/IBMdatascience

09/05/2024

Outline

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

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

Section 1

Methodology

Methodology

Executive Summary

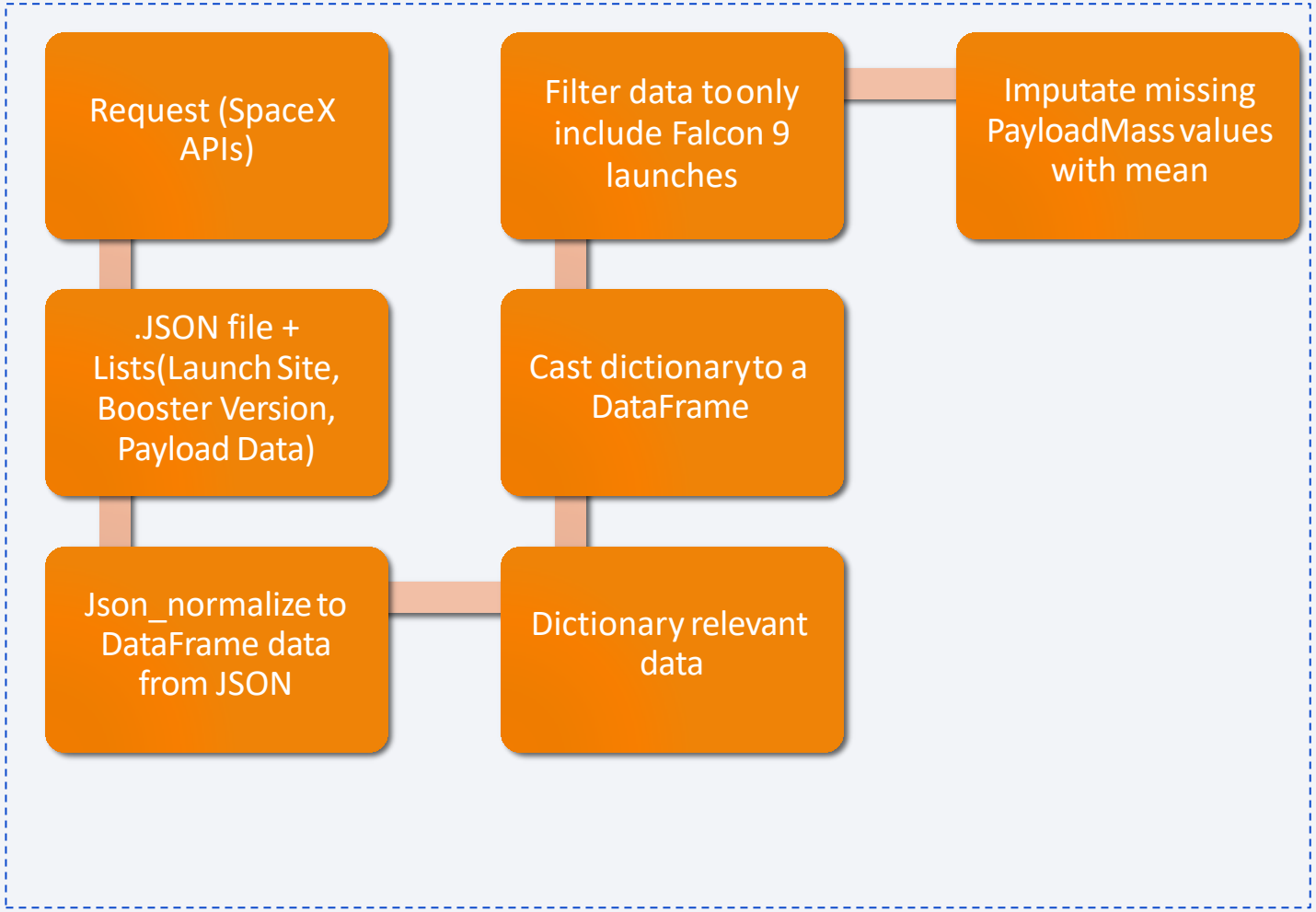
- Data collection methodology:
  - Combined data from SpaceX public API and SpaceX Wikipedia page
- Perform data wrangling
  - Classifying true landings as successful and unsuccessful otherwise
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Tuned models using GridSearchCV

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

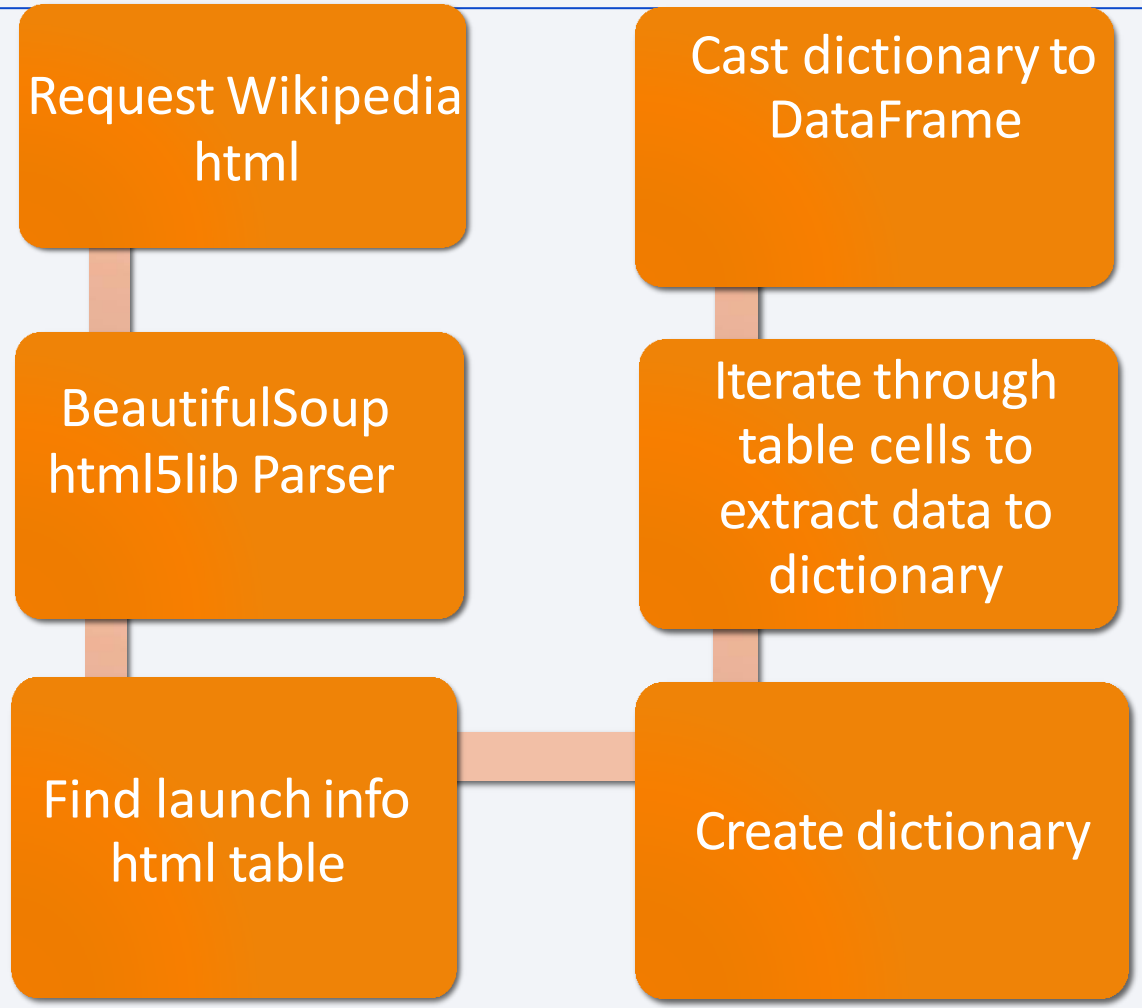
Data Collection – SpaceX API

- GitHub URL of the completed SpaceX API calls notebook:
- https://github.com/amruthpai123/IBMdatascience/blob/main/CapstoneProject/SpaceXFalcon9DataCollection.ipynb



Data Collection - Scraping

- GitHub URL of the completed web scraping notebook:
- https://github.com/amruthpai123/IBMdatascience/blob/main/CapstoneProject/spaceXWebScrapping.ipynb



Data Wrangling

- Create a training label with landing outcomes where successful = 1 & failure = 0.
- Outcome column has two components: ‘Mission Outcome’ ‘Landing Location’
- New training label column ‘class’ with a value of 1 if ‘Mission Outcome’ is True and 0 otherwise.

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 URL:

- https://github.com/amruthpai123/IBMdatascience/blob/main/CapstoneProject/DataWrangling.ipynb

EDA with Data Visualization

- Exploratory Data Analysis performed on variables Flight Number, Payload Mass, Launch Site, Orbit, Class and Year.

Plots Used:

- 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, and Success Yearly Trend

- Scatter plots, line charts, and bar plots were used to compare relationships between variables to

- decide if a relationship exists so that they could be used in training the machine learning model

GitHub url:
https://github.com/navassherif98/IBM_Data_Science_Professional_Certification/blob/master/10.Applied_Data_Science_Capstone/Week%202%20EDA/EDA%20with%20Visualization.ipynb

EDA with SQL

- Loaded data set into IBM DB2 Database.
- Queried using SQL Python integration.
- Queries were made to get a better understanding of the dataset.
- Queried information about launch site names, mission outcomes, various pay load sizes of customers and booster versions, and landing outcomes

GitHub url:

- https://github.com/amruthpai123/IBMdatascience/blob/main/CapstoneProject/SQLCapstone.ipynb

Build an Interactive Map with Folium

- Folium maps mark Launch Sites, successful and unsuccessful landings, and a proximity example to key locations: Railway, Highway, Coast, and City.
- This allows us to understand why launch sites may be located where they are. Also visualizes successful landings relative to location.
- GitHub url:
- https://github.com/amruthpai123/IBMdatascience/blob/main/CapstoneProject/FoliumCapstone.ipynb

Build a Dashboard with Plotly Dash

Summarize what plots/graphs and interactions you have added to a dashboard

Explain why you added those plots and interactions

Add the Dashboard includes a pie chart and a scatter plot.

Pie chart can be selected to show distribution of successful landings across all launch sites and can be selected to show individual launch site success rates.

Scatter plot takes two inputs: All sites or individual site and payload mass on a slider between 0 and 10000 kg.

The pie chart is used to visualize launch site success rate.

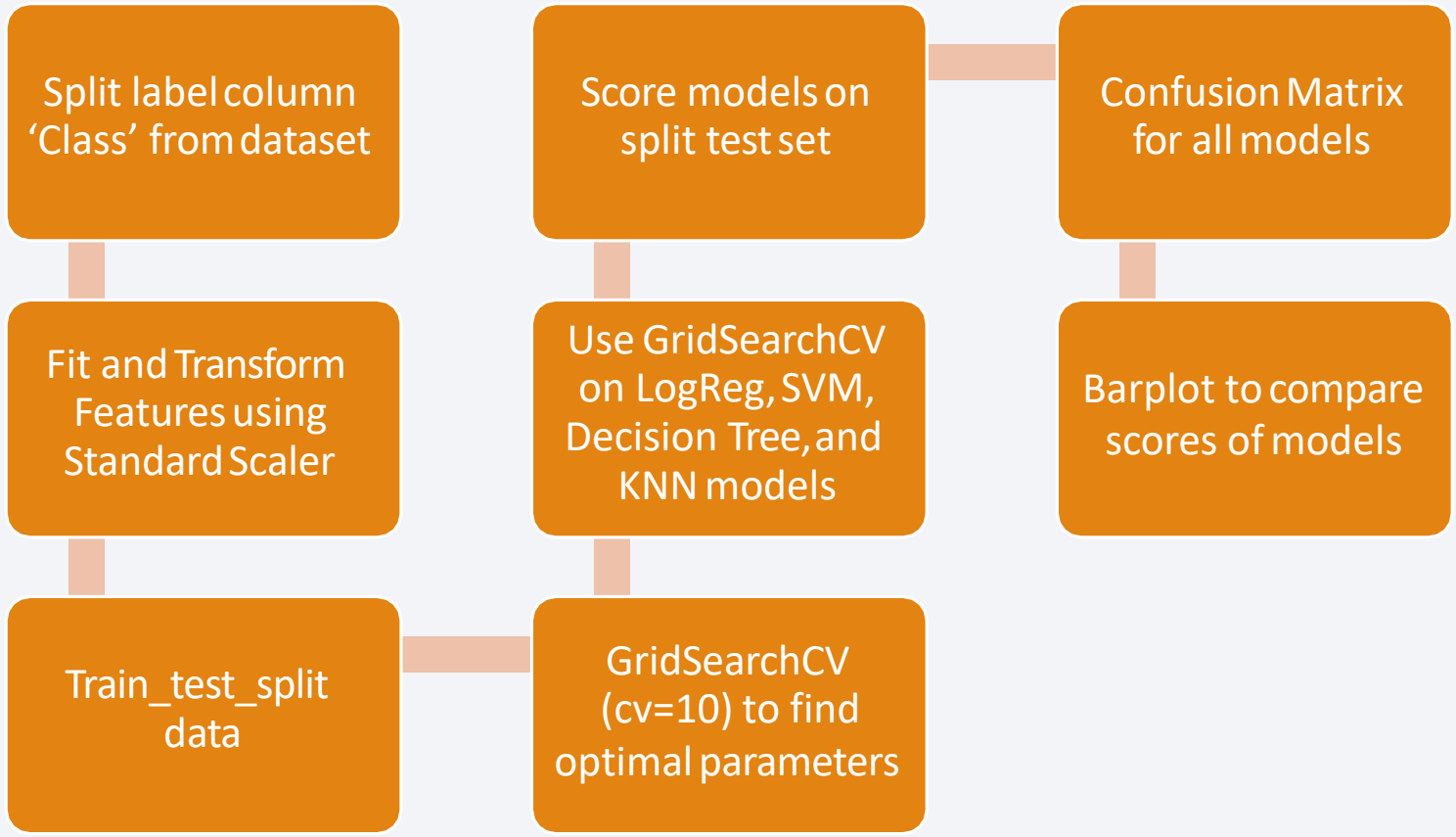
The scatter plot can help us see how success varies across launch sites, payload mass, and

booster version category.

- GitHub url:
- https://github.com/amruthpai123/IBMdatascience/blob/main/CapstoneProject/PlotlyDashboardCapstone.ipynb

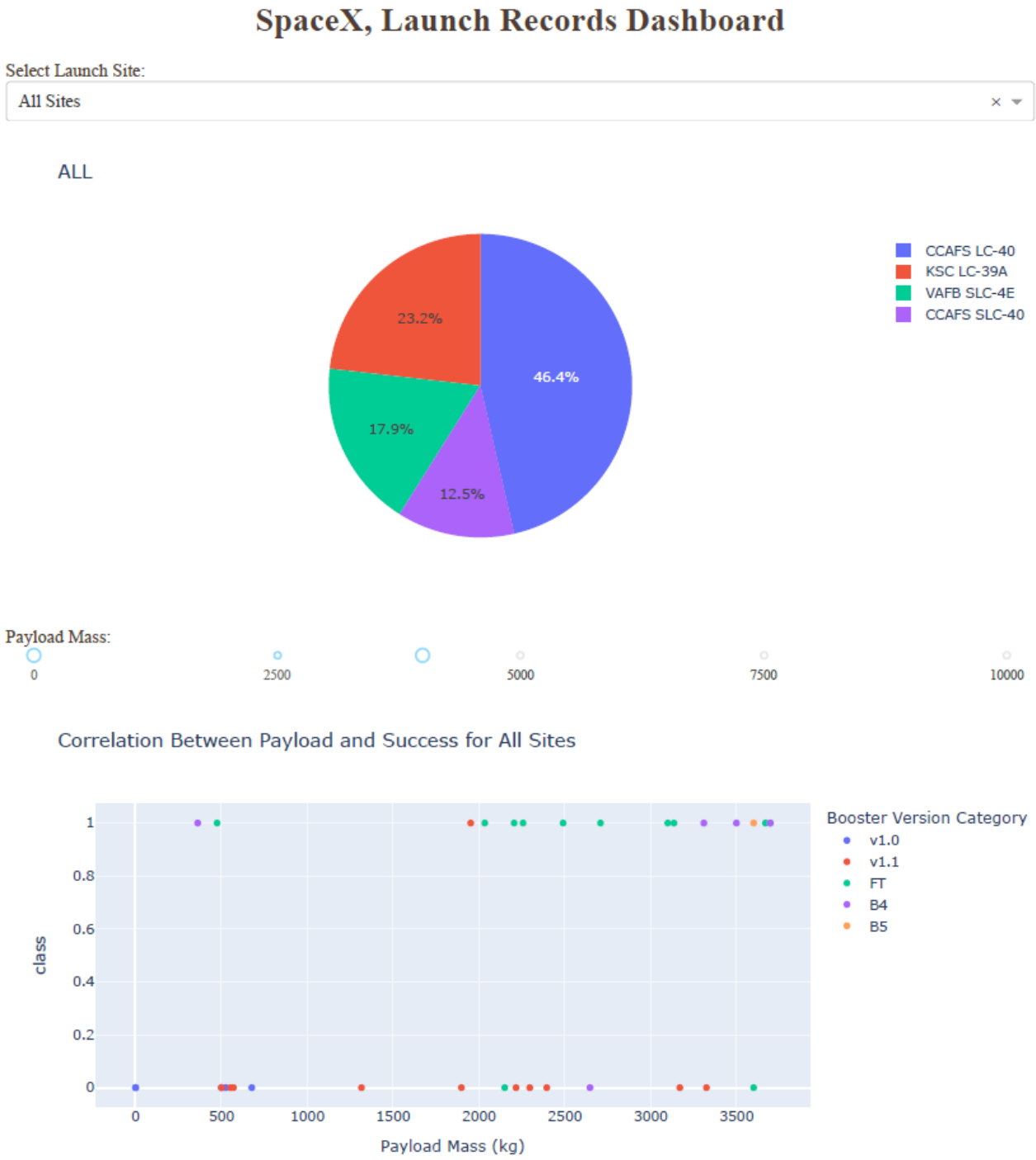
Predictive Analysis (Classification)

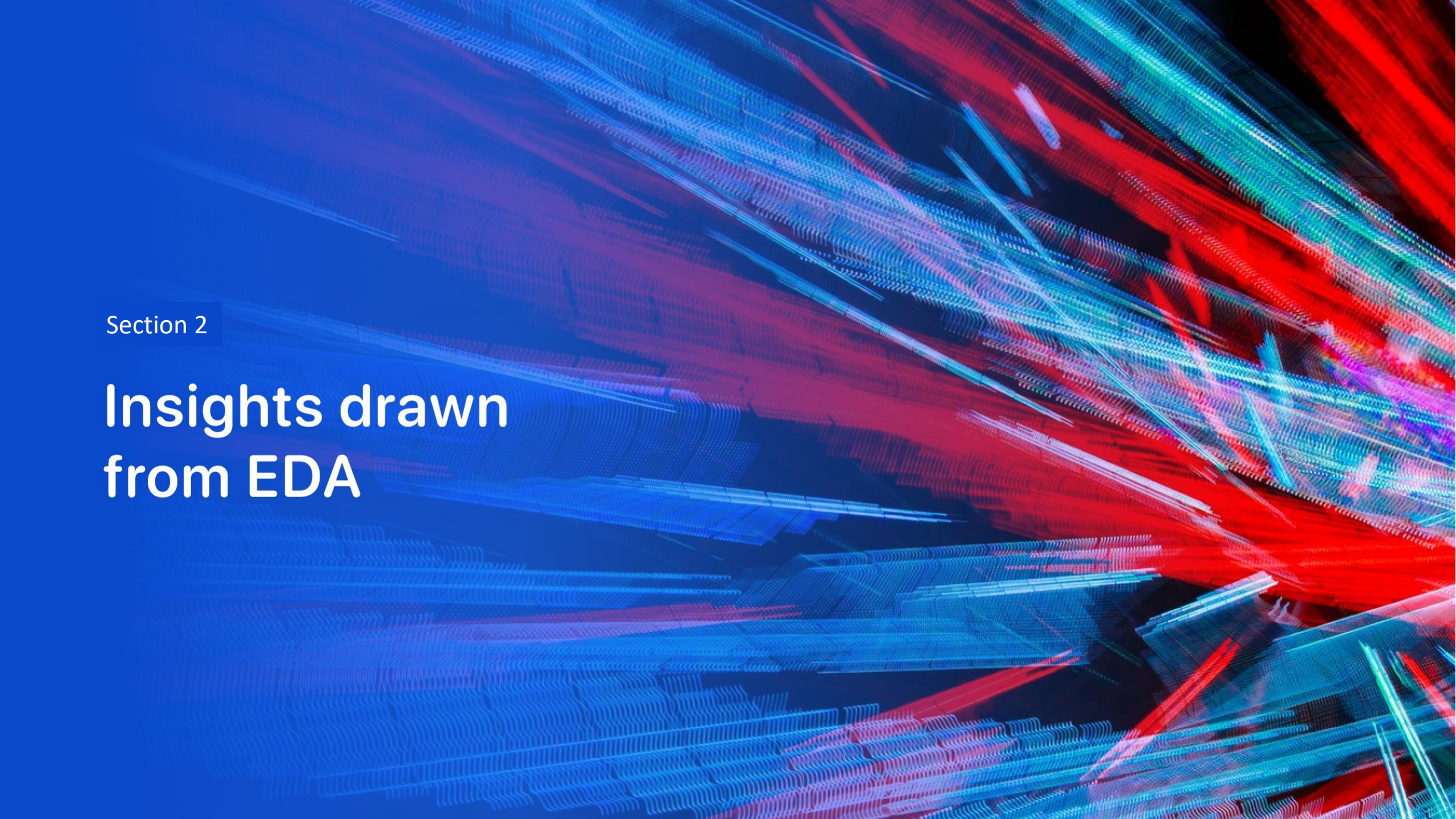
GitHub url:
https://github.com/amruthpai123/IBMdatascience/blob/main/CapstoneProject/MLCapstone.ipynb



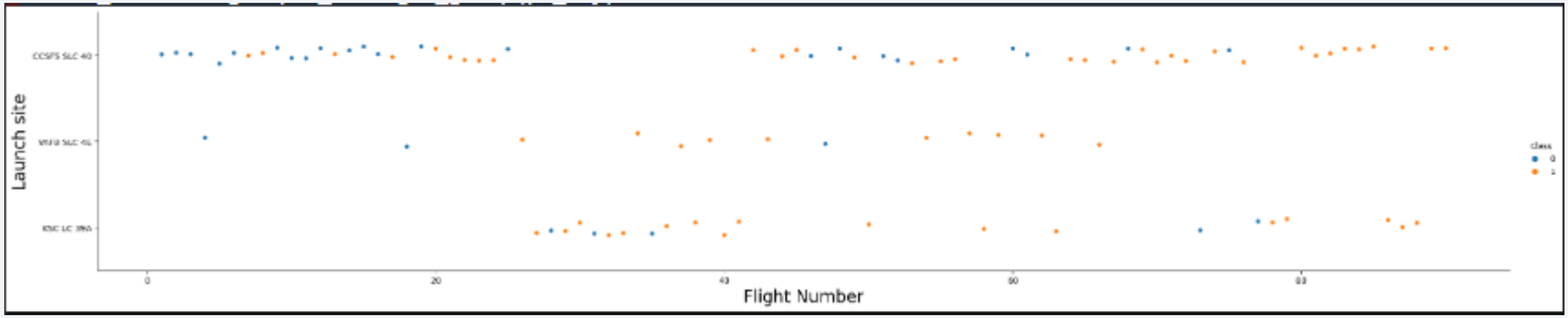
Results

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.




Section 2
Insights drawn from EDA

Flight Number vs. Launch Site



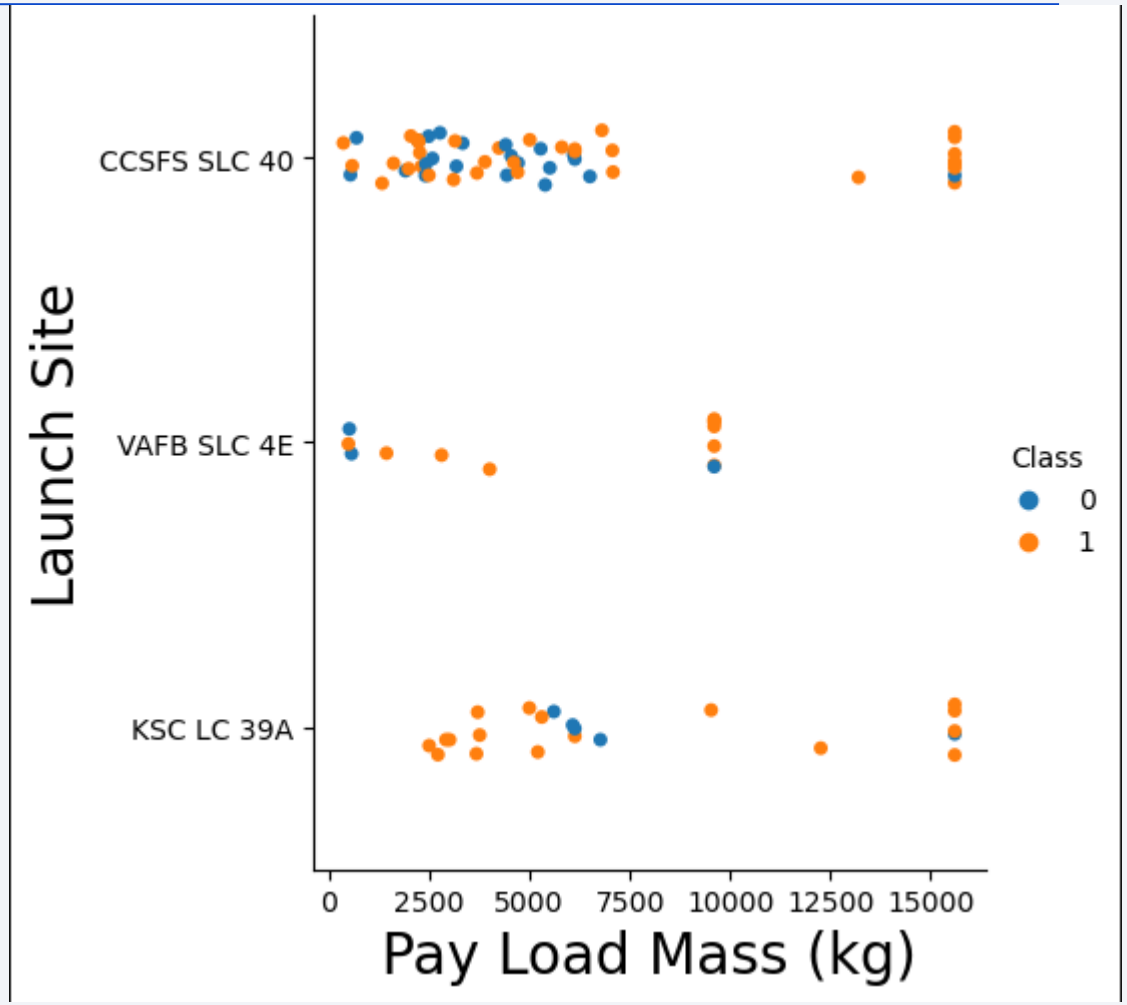
Orange indicates successful launch; Blue indicates unsuccessful launch

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

Orange indicates successful launch; Blue indicates unsuccessful launch

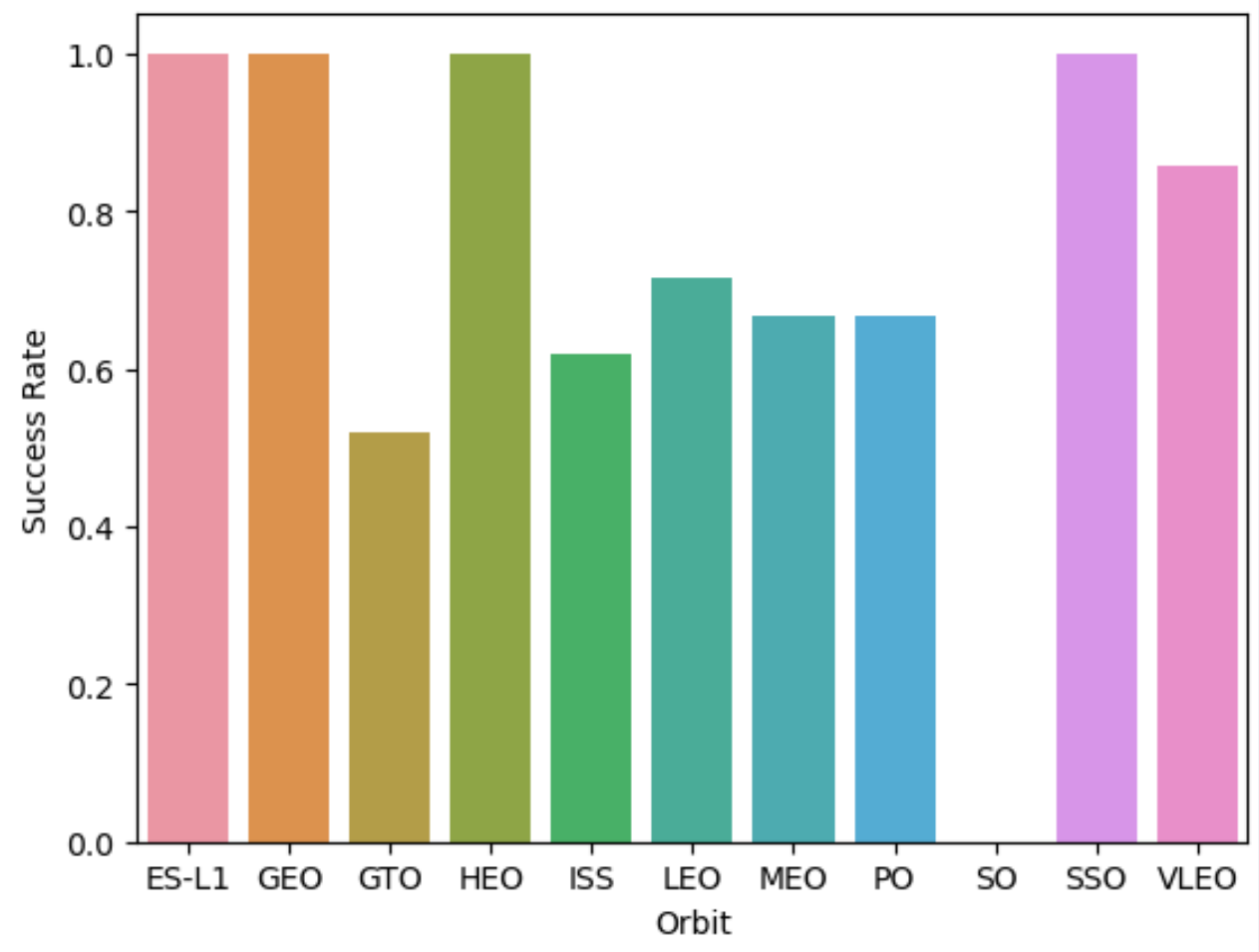
Payload mass appears to fall mostly between 0-6000 kg. Different launch sites also seem to use different payload mass



Success Rate vs. Orbit Type

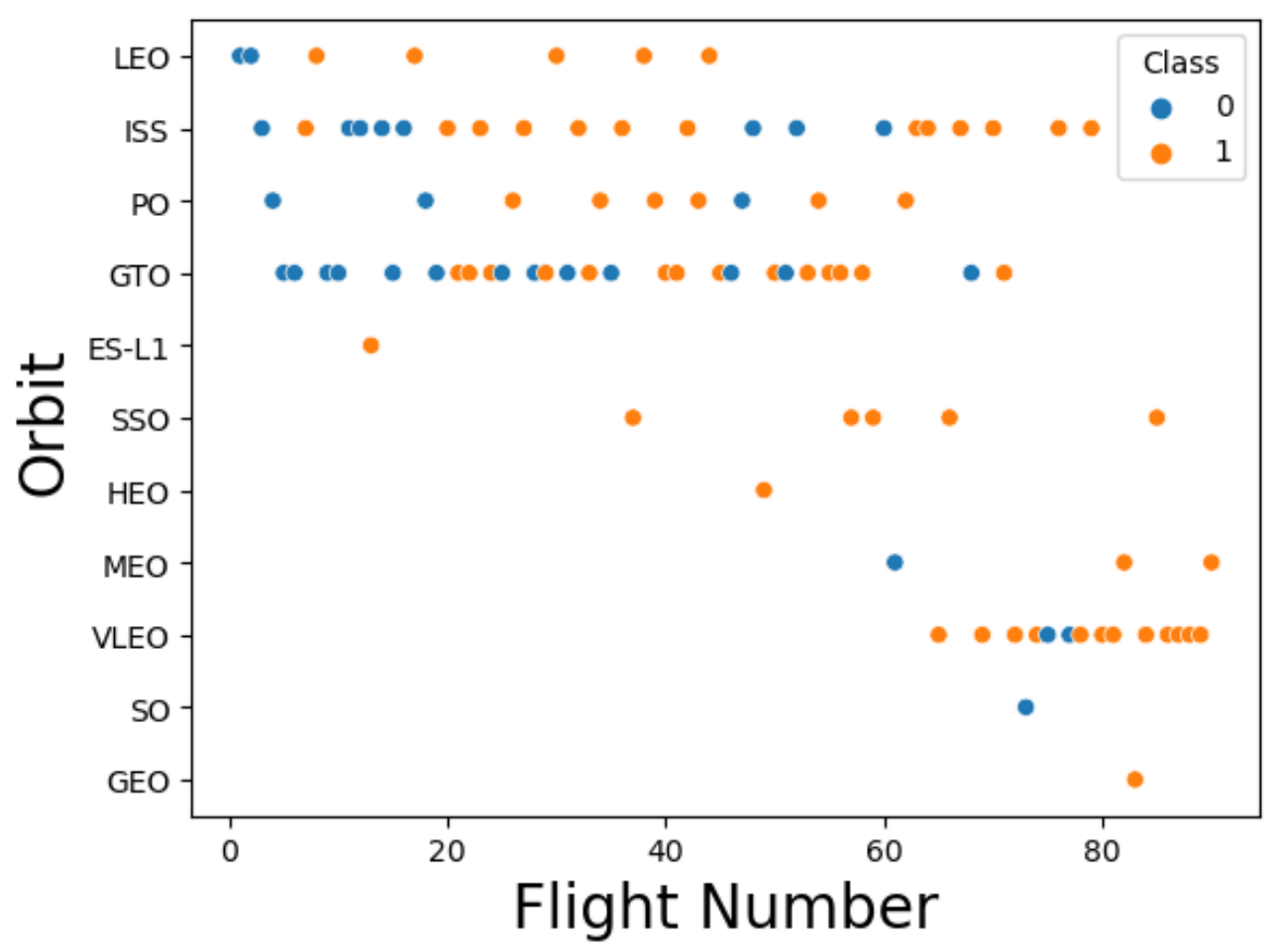
Success Rate Scale with
0 as 0%
0.6 as 60%
1 as 100%

- 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



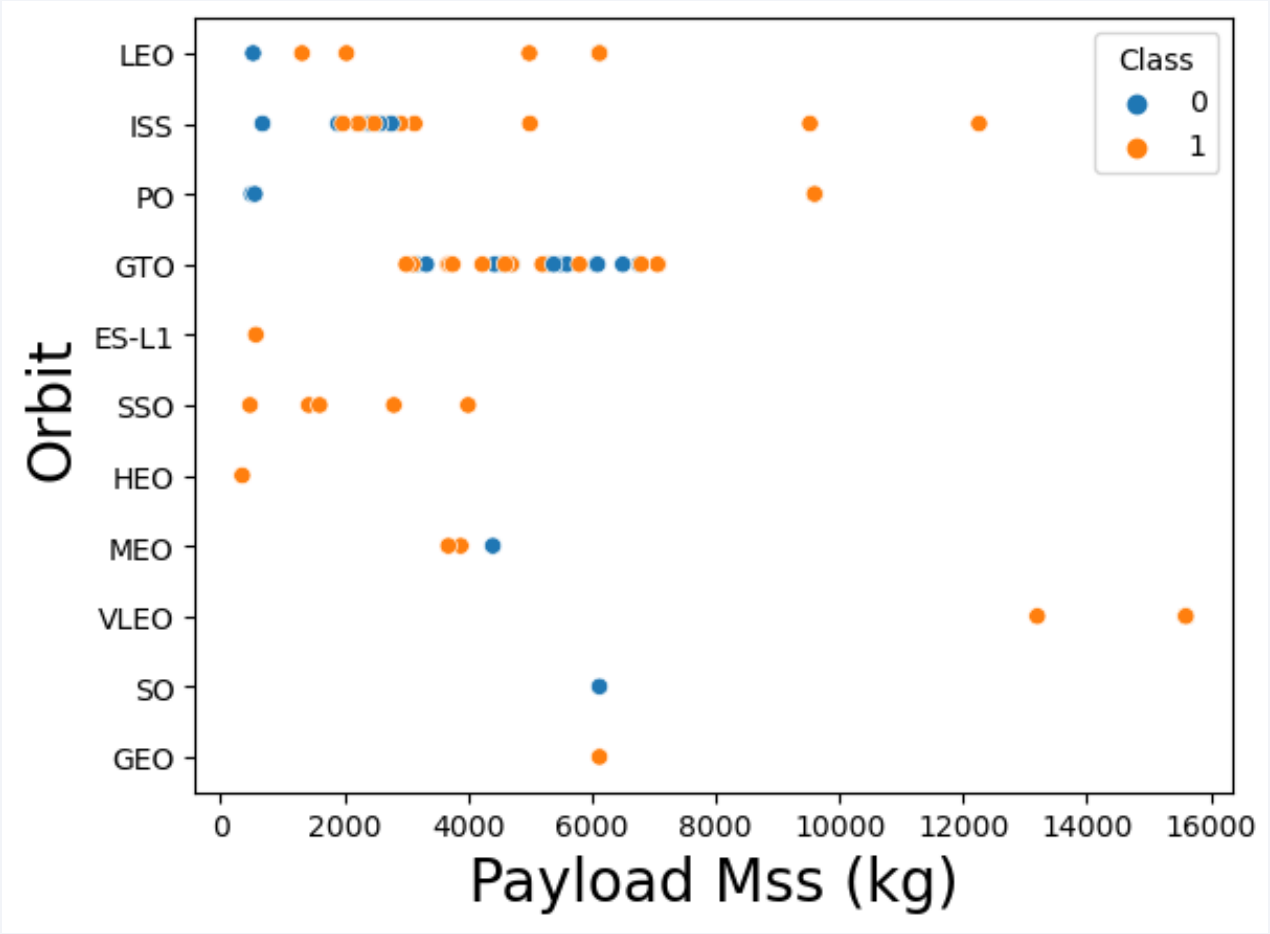
Flight Number vs. Orbit Type

- Orange indicates successful launch; Blue indicates unsuccessful launch
- Show a scatter point of Flight number vs. Orbit type
- Show the screenshot of the scatter plot 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

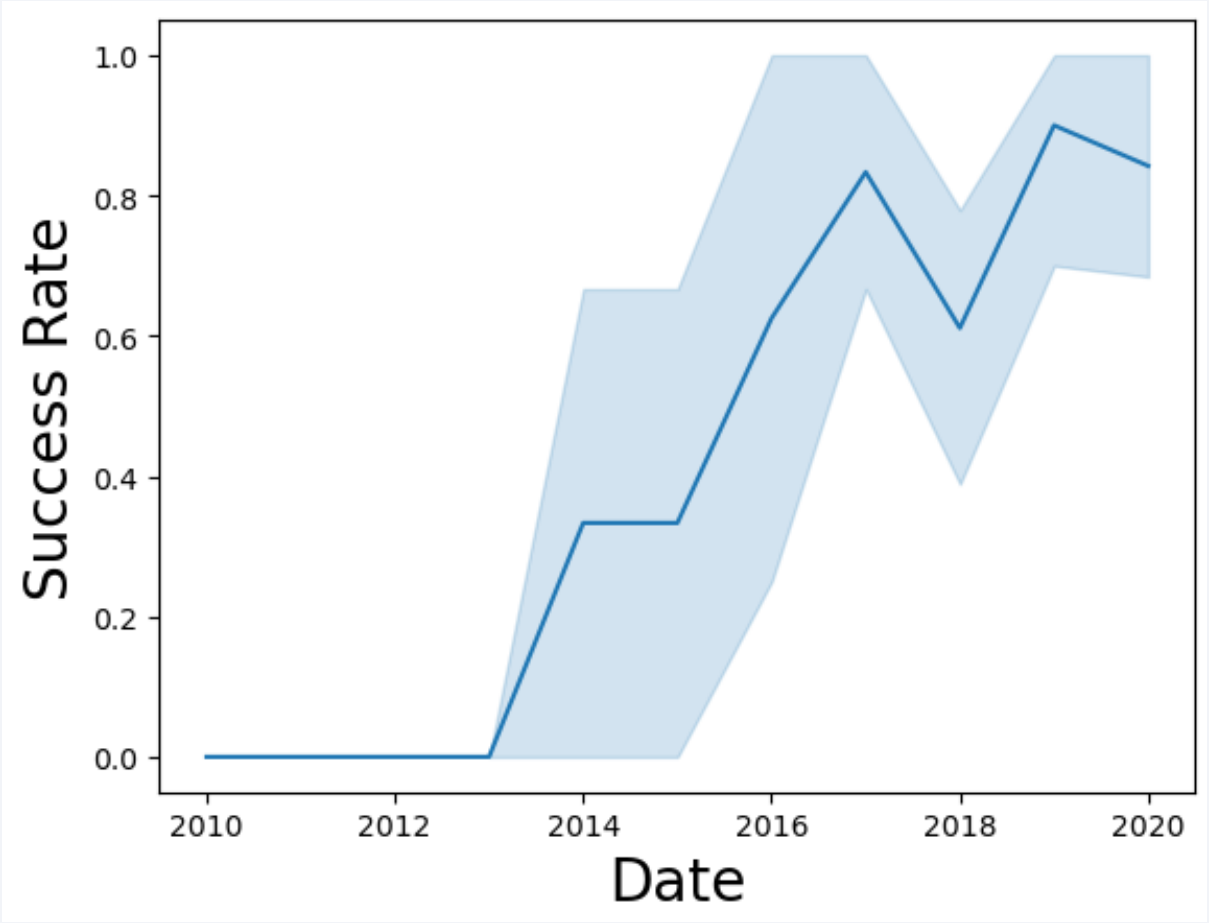
Orange indicates successful launch;
Blue indicates unsuccessful launch.



- 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
- the screenshot of the scatter plot with explanations

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

EXPLORATORY DATA ANALYSIS WITH SQL DB2

INTEGRATED IN PYTHON WITH SQLALCHEMY

All Launch Site Names

Present Likely only 4 unique launch_site values:

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

Task 1

Display the names of the unique launch sites in the space mission

In [28]:
```
%%sql
select distinct(Launch_Site) from spacex
```

```
 * sqlite:///my_data1.db
   sqlite:///mydata1.db
Done.
```

Out[28]:

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

Launch Site Names Begin with 'CCA'

```
[10]: %%sql
select * from spacex where Launch_Site like "CCA%" limit 5
```

```
 * sqlite:///my_data1.db
Done.
```

[10]:

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

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

Total Payload Mass

```
[50]: %%sql
      select Customer, sum(PAYLOAD_MASS__KG_) as 'TOTAL PAYLOAD' from spacex where Customer like '%NASA%(CRS)%'
       * sqlite:///my_data1.db
         sqlite:///mydata1.db
      Done.
```

[50]:

Customer	TOTAL PAYLOAD
NASA (CRS)	48213

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

```
[49]: %%sql
      select Booster_Version, avg(PAYLOAD_MASS__KG_) as 'AVERAGE PAYLOAD' from spacex where Booster_Version like '%F9 v1.1%'
```

```
 * sqlite:///my_data1.db
   sqlite:///mydata1.db
Done.
```

[49]:

Booster_Version	AVERAGE PAYLOAD
F9 v1.1 B1003	2534.6666666666665

- 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 Landing Date

```
[11]: %%sql
      select min(Date) as first_success from spacex where Landing_Outcome = 'Success (ground pad)'

 * sqlite:///my_data1.db
Done.
[11]: 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

```
•[58]: %%sql
select Booster_Version  from spacex where Landing_Outcome = 'Success (drone ship)' and
PAYLOAD_MASS__KG_ >4000 and PAYLOAD_MASS__KG_<6000

 * sqlite:///my_data1.db
   sqlite:///mydata1.db
Done.

[58]:
```

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 Successful and Failure Mission Outcomes

```
[80]: %%sql
select Mission_Outcome, count(Mission_Outcome) as outcome from spacex group by Mission_Outcome
 * sqlite:///my_data1.db
   sqlite:///mydata1.db
Done.
```

[80]:

Mission_Outcome	outcome
Failure (in flight)	1
Success	98
Success	1
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 Carried Maximum Payload

```
[67]: %%sql
select Booster_Version, PAYLOAD_MASS__KG_ from spacex where
PAYLOAD_MASS__KG_ =(select max(PAYLOAD_MASS__KG_) from spacex)

 * sqlite:///my_data1.db
   sqlite:///mydata1.db
Done.
```

[67]:

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 Launch Records

```
[70]: %%sql
select substr(Date,6,2) as Month, Landing_Outcome, Booster_Version, Launch_Site
from spacex where substr(Date,0,5)='2015' and Landing_Outcome like '%failure%'

 * sqlite:///my_data1.db
   sqlite:///mydata1.db
Done.
```

[70]:

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

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

```
[13]: %%sql
      select Landing_Outcome, count(*) as Count_Launches from spacex where
      landing_Outcome like '%Success%' and
      Date>='2010-06-04' and Date<='2017-03-20'
      group by Landing_Outcome order by Count_Launches desc

       * sqlite:///my_data1.db
      Done.
```

[13]:

Landing_Outcome	Count_Launches
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.

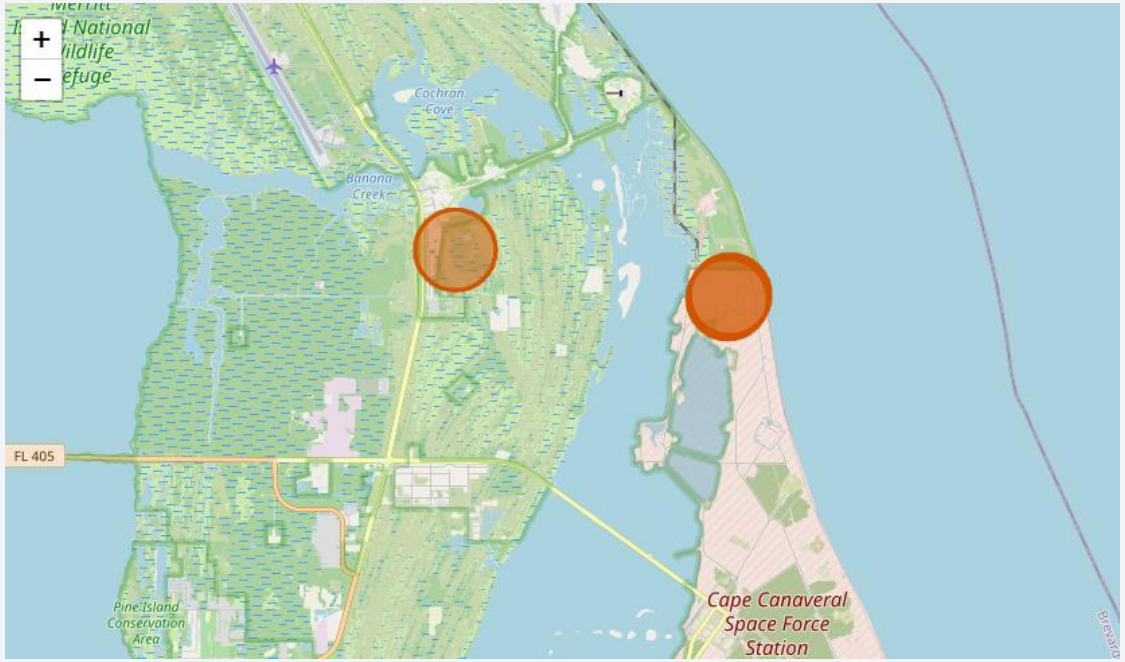
There are two types of successful landing outcomes: drone ship and ground pad landings.

There were 8 successful landings in total during this time period

Section 3

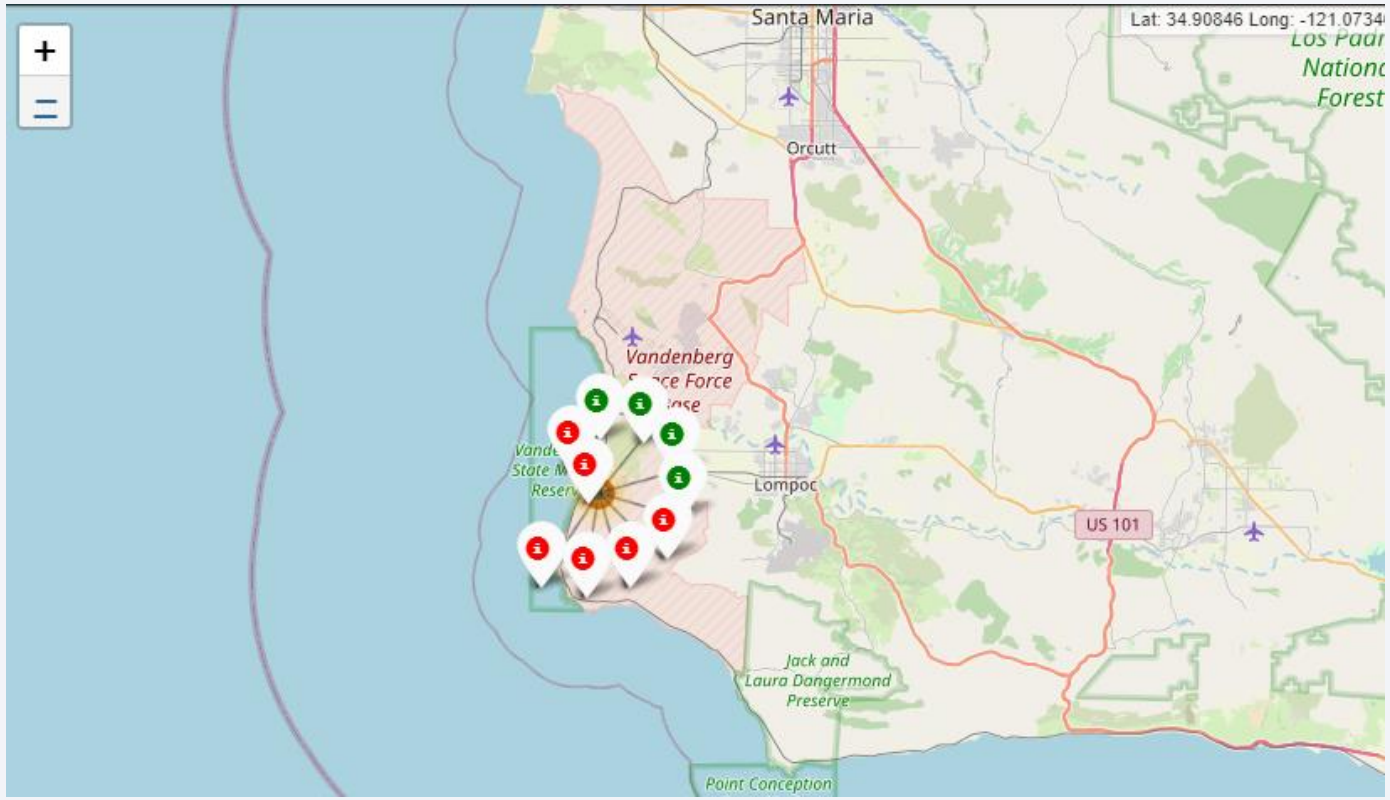
Launch Sites Proximities Analysis

Launch Site Locations



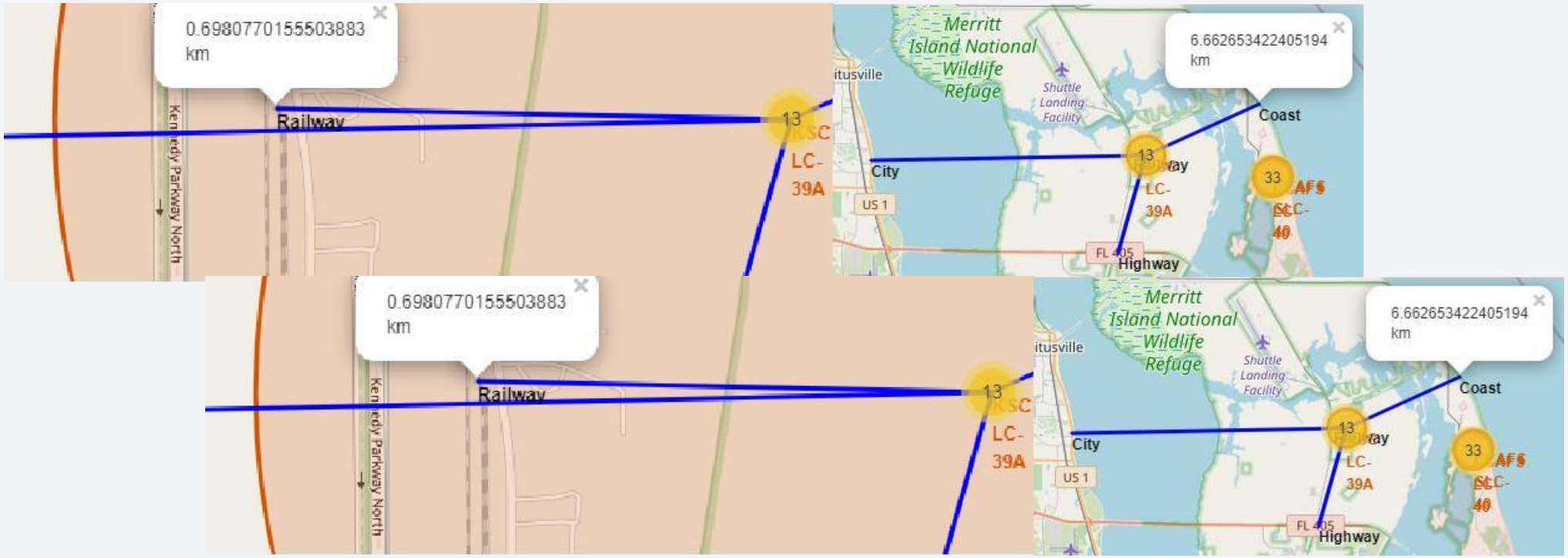
- The left map shows all launch sites relative US map. The right map shows the two Florida launch sites since they are very close to each other. All launch sites are near the ocean.

Color-Coded Launch Markers

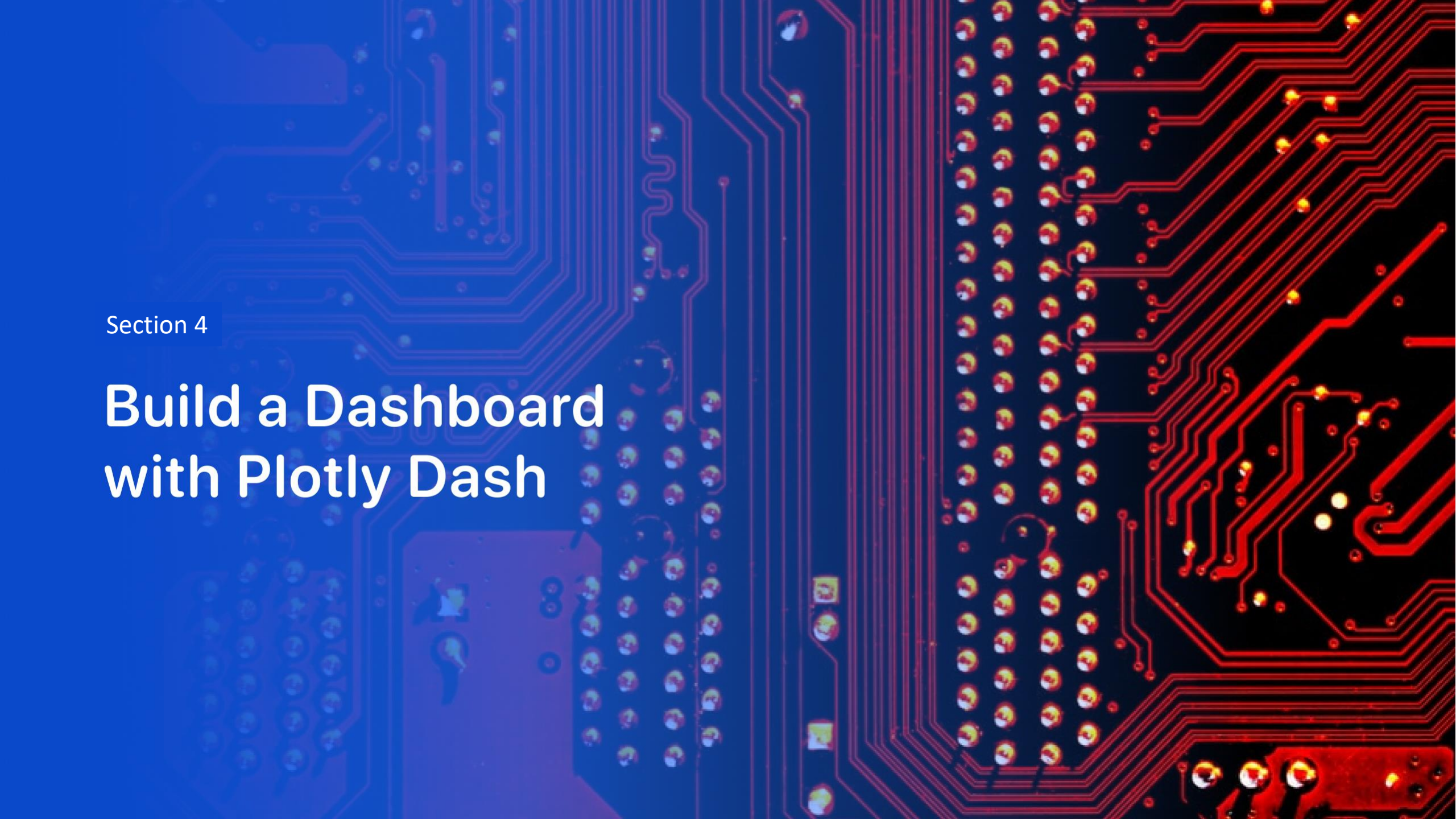


Clusters on Folium map can be clicked on to display each successful landing (green icon) and failed landing (red icon). In this example VAFB SLC-4E shows 4 successful landings and 6 failed landings.

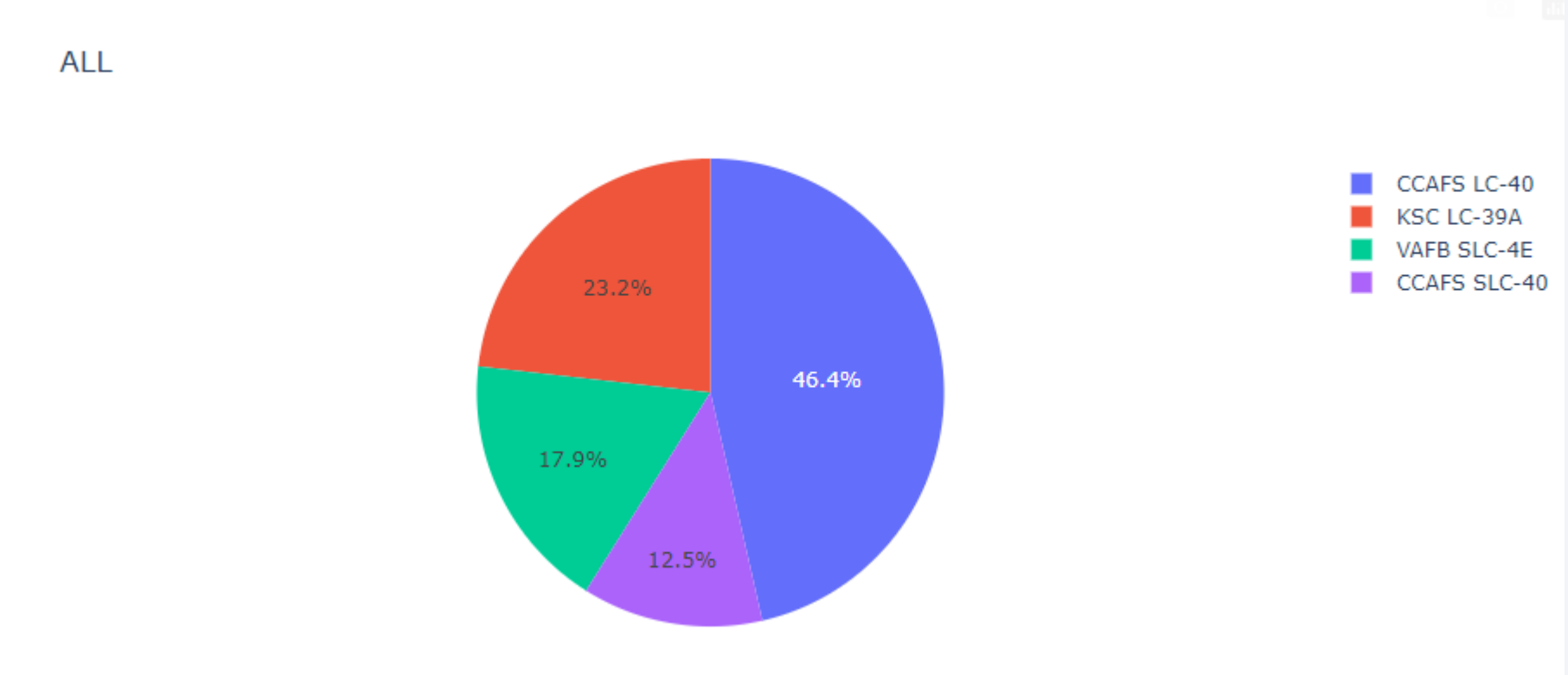
Key Location Proximities



- 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.

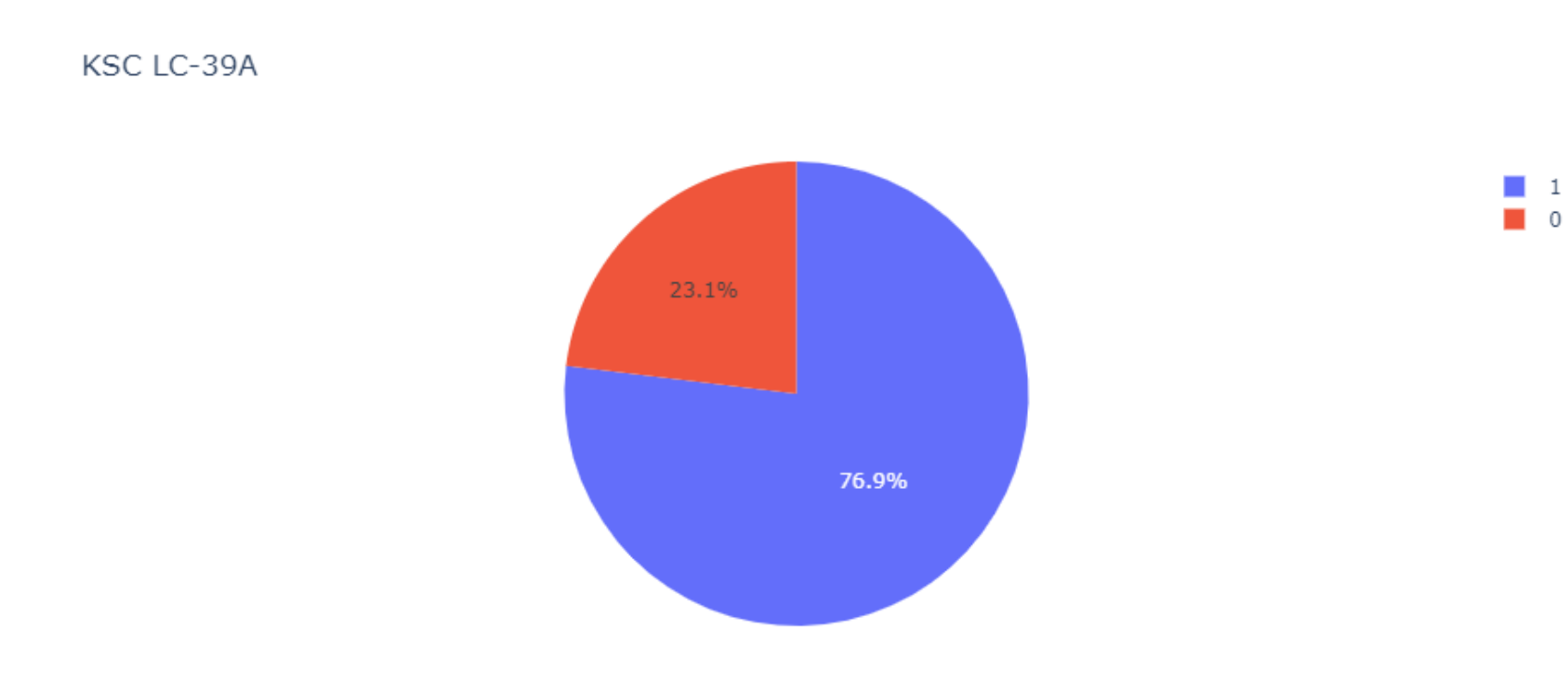

Section 4
Build a Dashboard with Plotly Dash

Successful Launches Across Launch Sites



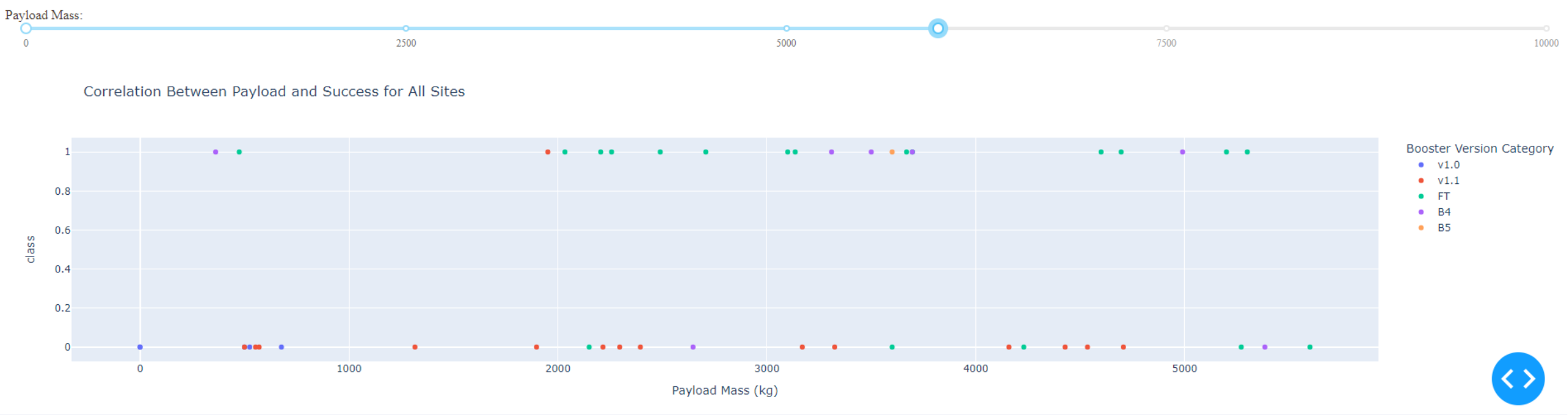
- This is the distribution of successful landings across all launch sites. CCAFS LC-40 is the old name of CCAFS SLC-40 so CCAFS and KSC have the same amount of successful landings, but a majority of the successful landings where performed before the name change. VAFB has the smallest share of successful 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

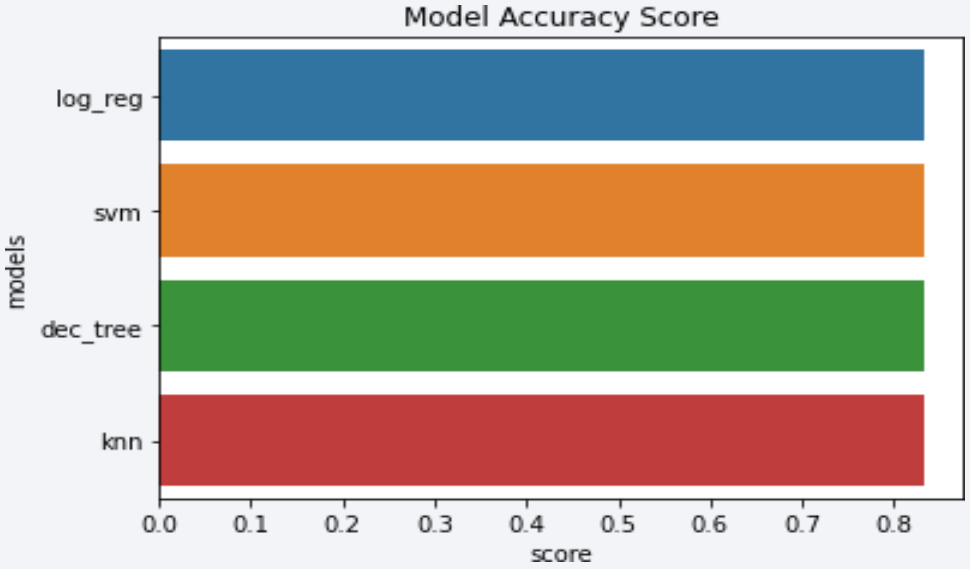


- Plotly dashboard has a Payload range selector. However, this is set from 0-10000 instead of the max Payload of 15600. Class indicates 1 for successful landing and 0 for failure. Scatter plot also accounts for booster version category in color and number of launches in point size. In this particular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

Section 5

Predictive Analysis (Classification)

Classification Accuracy



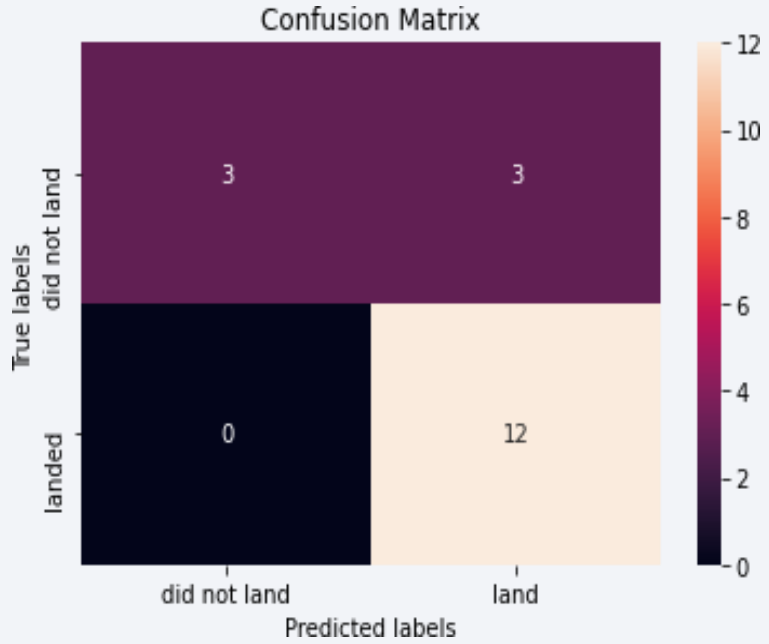
All models had virtually the same accuracy on the test set at 83.33% accuracy.

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.

Conclusions

- Our task: to develop a machine learning model for Space Y who wants to bid against SpaceX
- The goal of model is to predict when Stage 1 will successfully land to save ~$100 million USD
- Used data from a public SpaceX API and web scraping SpaceX Wikipedia page
- Created data labels and stored data into a DB2 SQL database
- Created a dashboard for visualization
- We created a machine learning model with an accuracy of 83%
- Allon Mask of SpaceY can use this model to predict with relatively high accuracy whether a launch will have a successful Stage 1 landing before launch to determine whether the launch should be made or not
- If possible more data should be collected to better determine the best machine learning model and improve accuracy

Appendix

GitHub repository url:

https://github.com/amruthpai123/IBMdatascience/tree/main

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!