IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Meet Shah
23-02-2022

Outline

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

Executive Summary

- Summary of methodologies
  - Collected data from public Space X API and Space X 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.
- Summary of all results
  - 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 company's are booming now days.
- Space X has best pricing ($62 million vs. $165 million USD)
- Largely due to ability to recover part of rocket (Stage 1)

Problem

- Space Y company is new in the market
- They want to achieve same recover part of rocket as Space X company.
- Space Y tasks us to train a machine learning model to predict successful Stage 1 recovery

Section 1

Methodology

Methodology

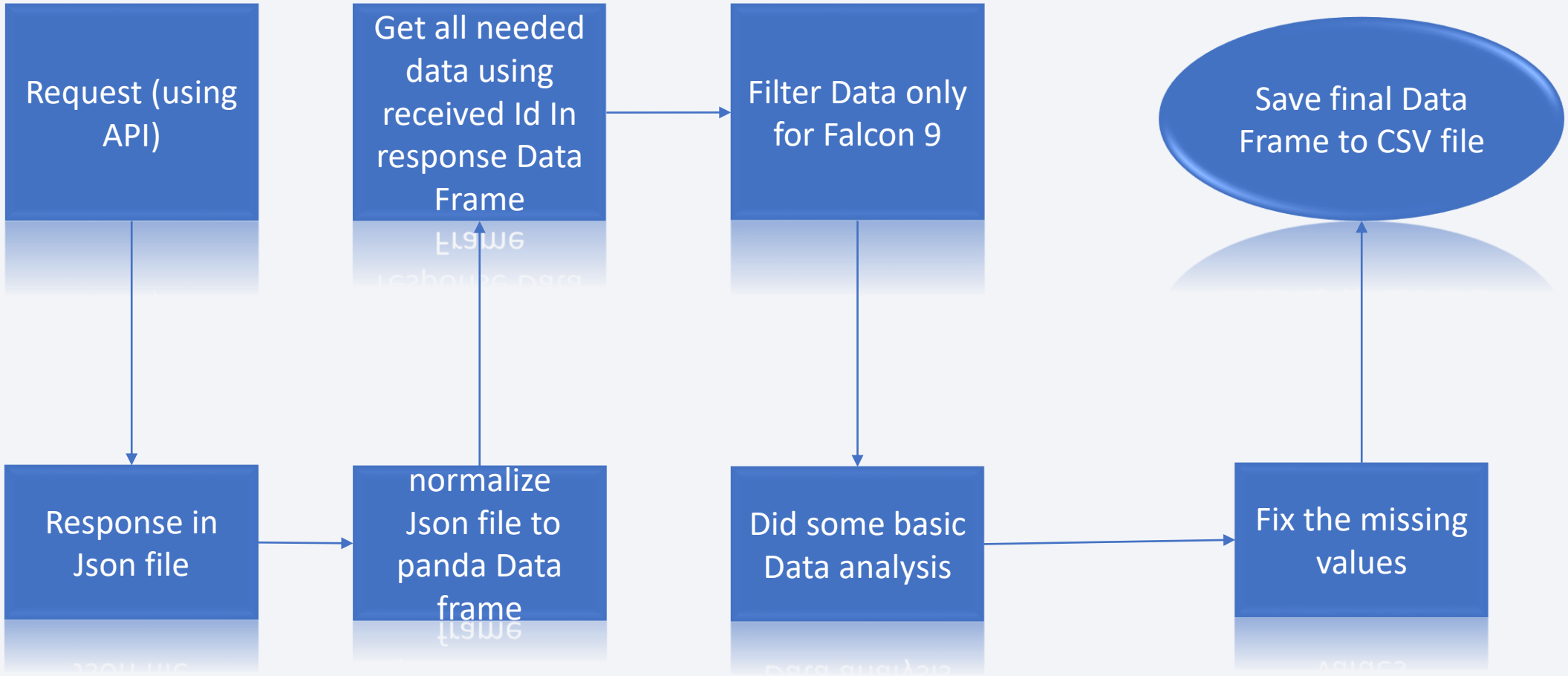
Executive Summary

- Data collection methodology:
  - Describe how data was collected
- Perform data wrangling
  - Describe how data was processed
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection

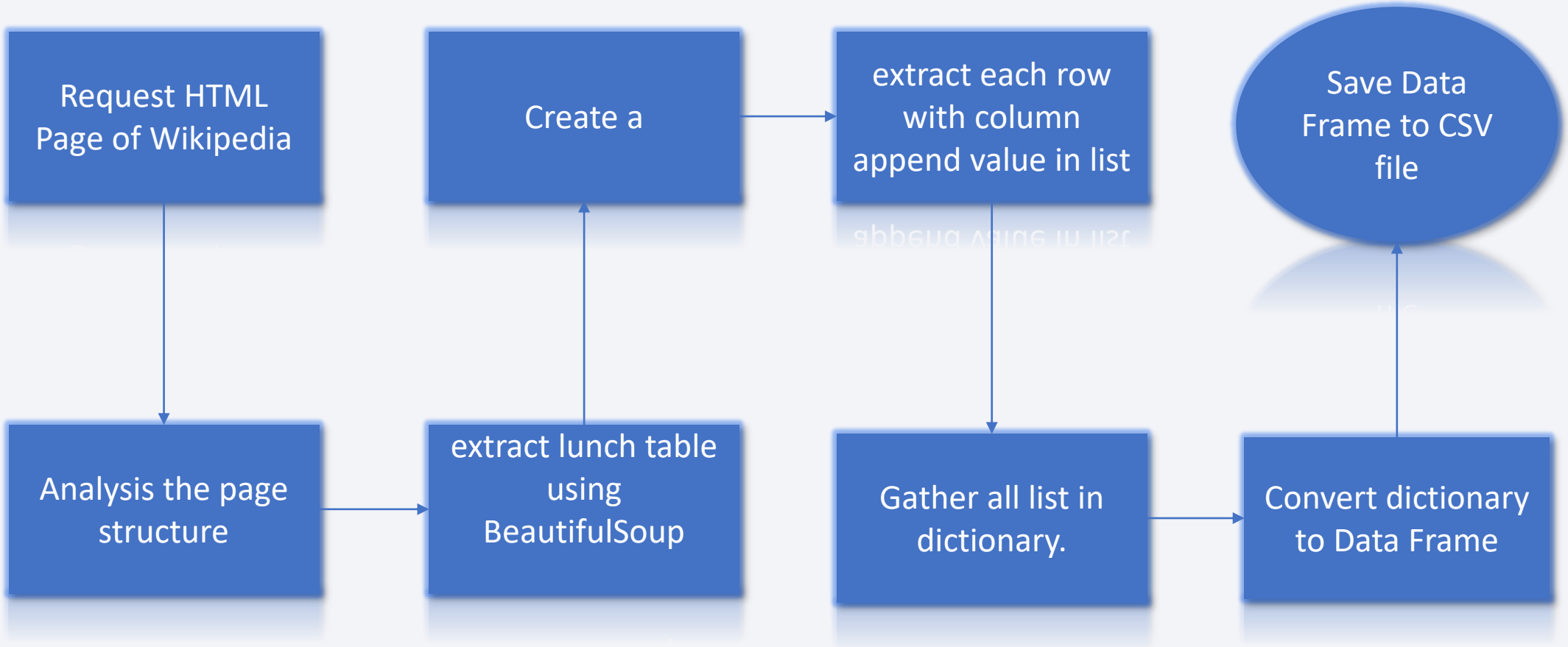
- Data collection process involved a combination of API requests from Space X public API and web scraping data from a table in Space X's Wikipedia entry.
- The next slide will show the flowchart of data collection from API and the one after will show the flowchart of data collection from web scraping.
- Space X API Data Columns:
  - Flight Number, Date, Booster Version, Payload Mass, Orbit, Launch Site, Outcome, Flights, GridFins,
  - Reused, Legs, Landing Pad, Block, Reused Count, Serial, Longitude, Latitude
- Wikipedia Web-scrape Data Columns:
  - Flight No., Launch site, Payload, Payload Mass, Orbit, Customer, Launch outcome, Version Booster, Booster landing, Date, Time

Data Collection - SpaceX API



- Jupiter notebook :- GIT LINK

Data Collection - Scraping



- Jupiter notebook :-GIT LINK

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 , False ASDS, None ASDS, False Ocean, False RTLS - set to -> 0

- Jupiter notebook :-GIT LINK

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.

- Jupiter notebook :-GIt LINK

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

- Jupiter notebook :-GIT LINK

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.

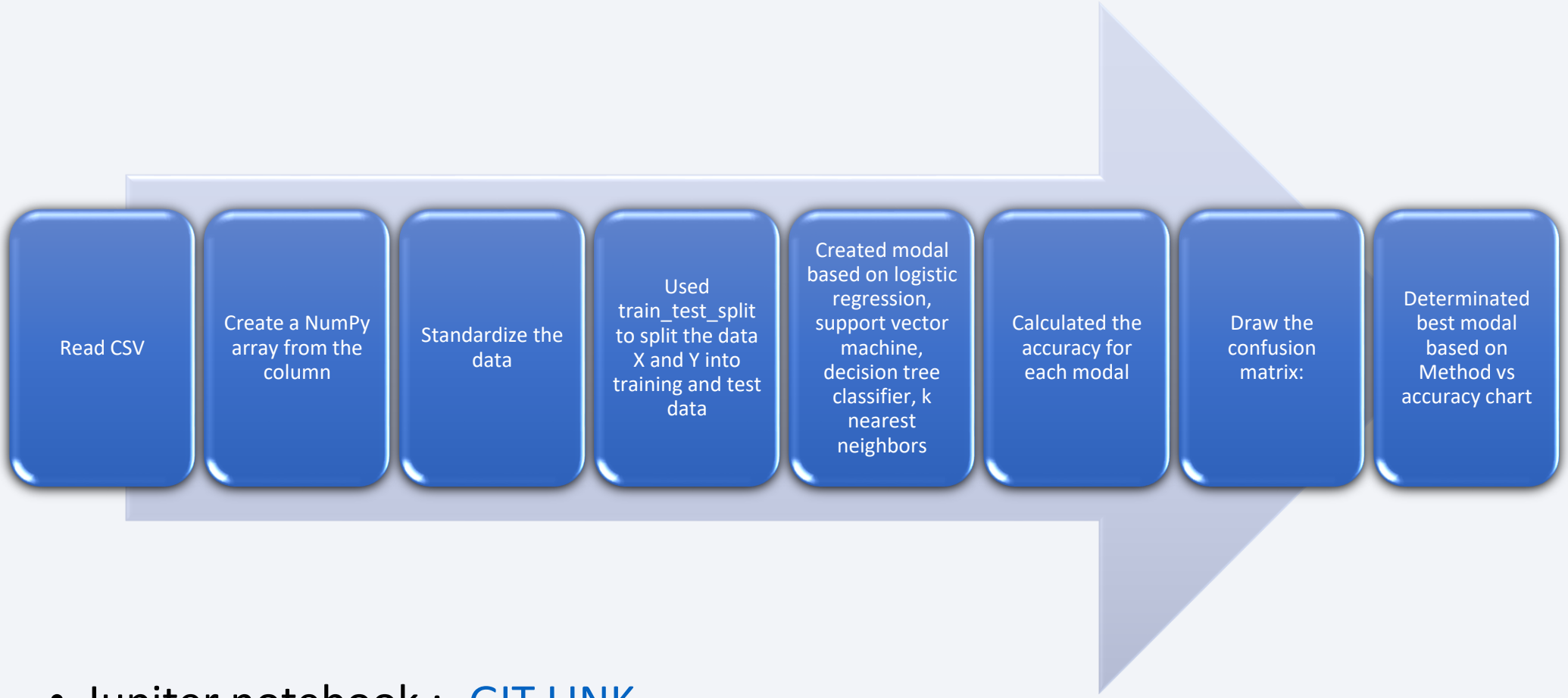
- Jupiter notebook :- GIT LINK

Build a Dashboard with Plotly Dash

- 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.

- Jupiter notebook :- GIT LINK

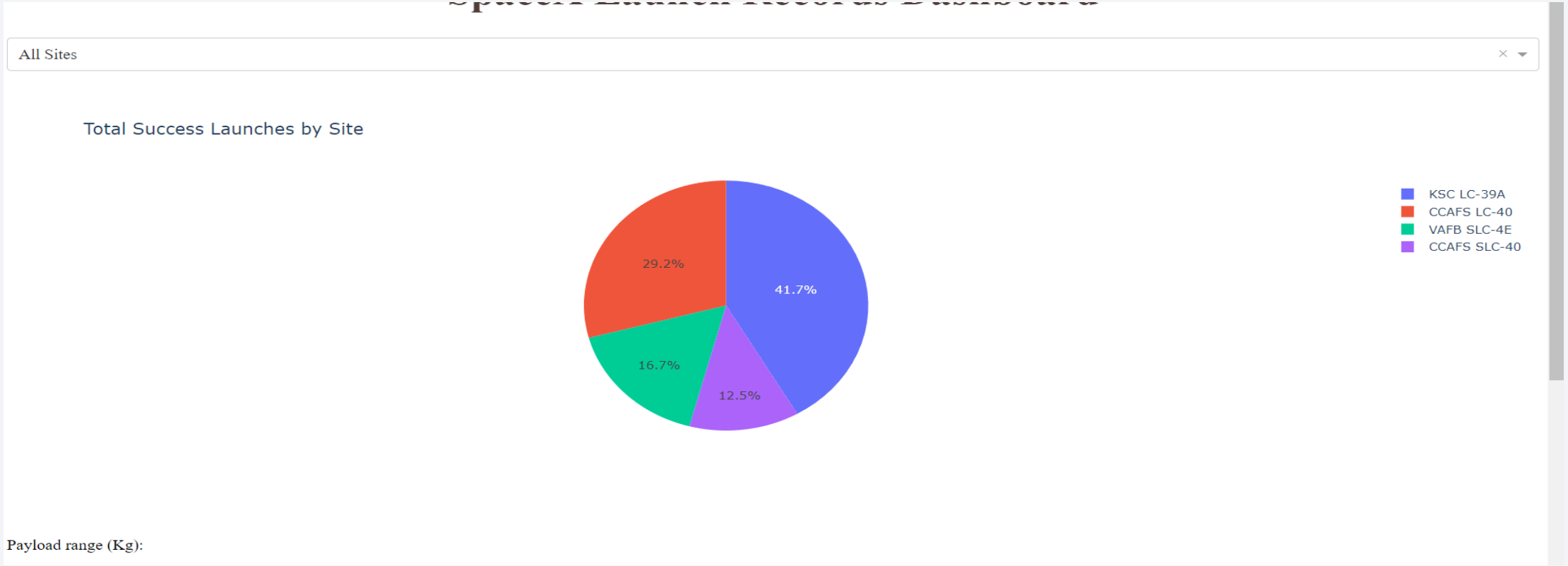
Predictive Analysis (Classification)



- Jupiter notebook :- GIT LINK

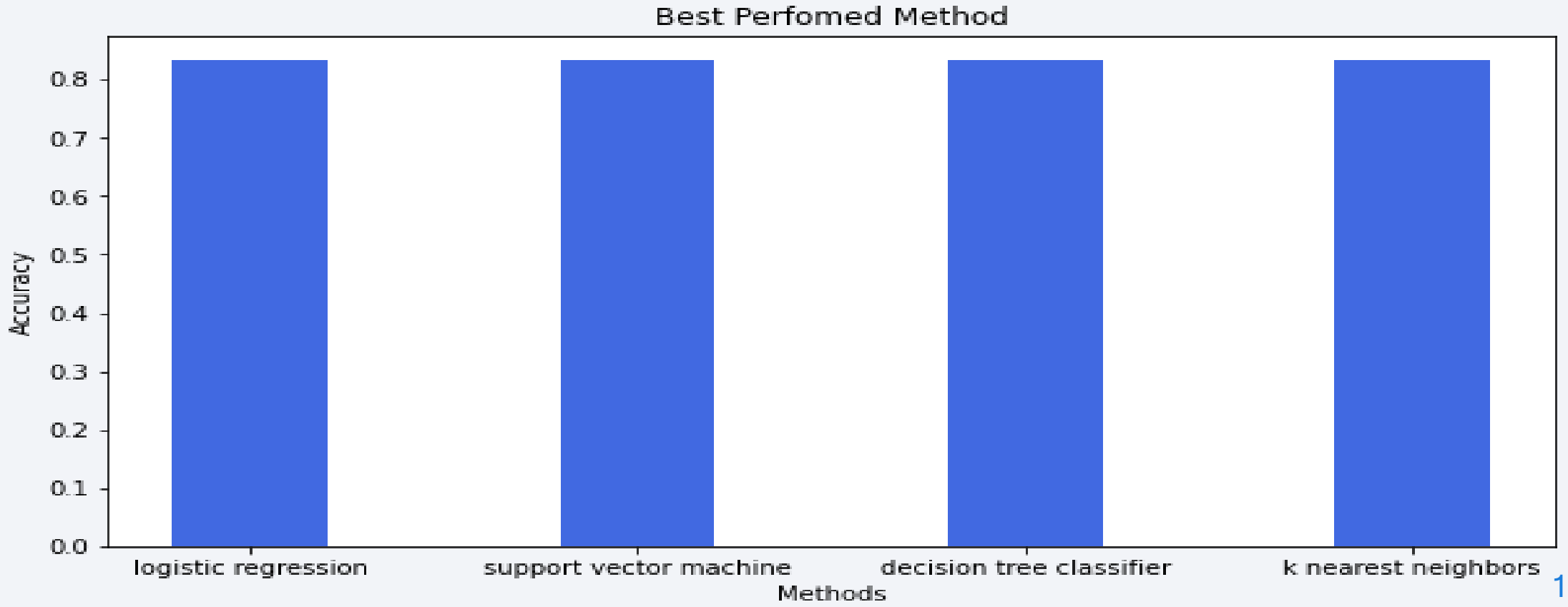
Results of Exploratory data analysis

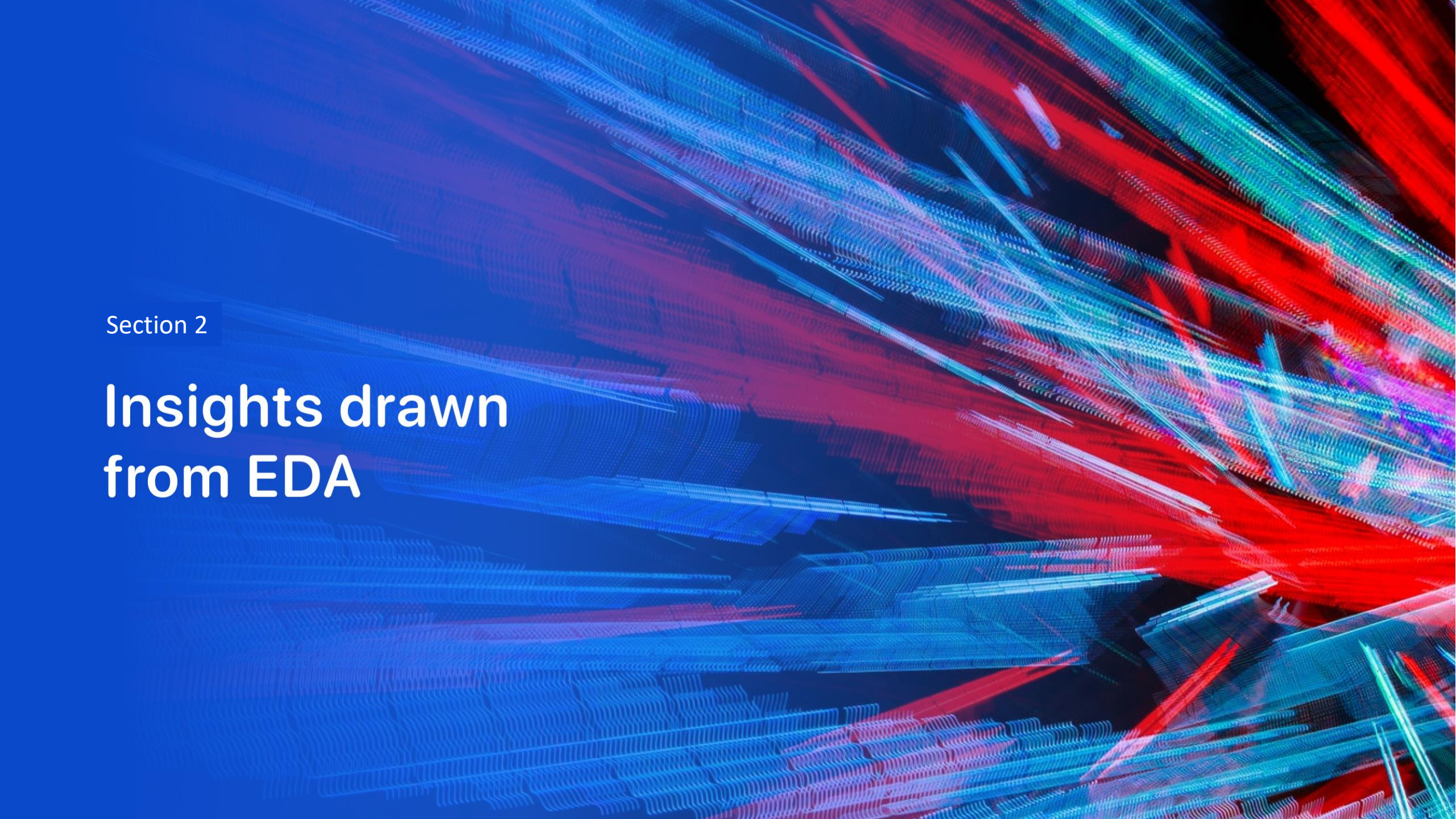
- This is a preview of the Plotly dashboard which indicates success lunches by percentage per location.



Results of Predictive analysis

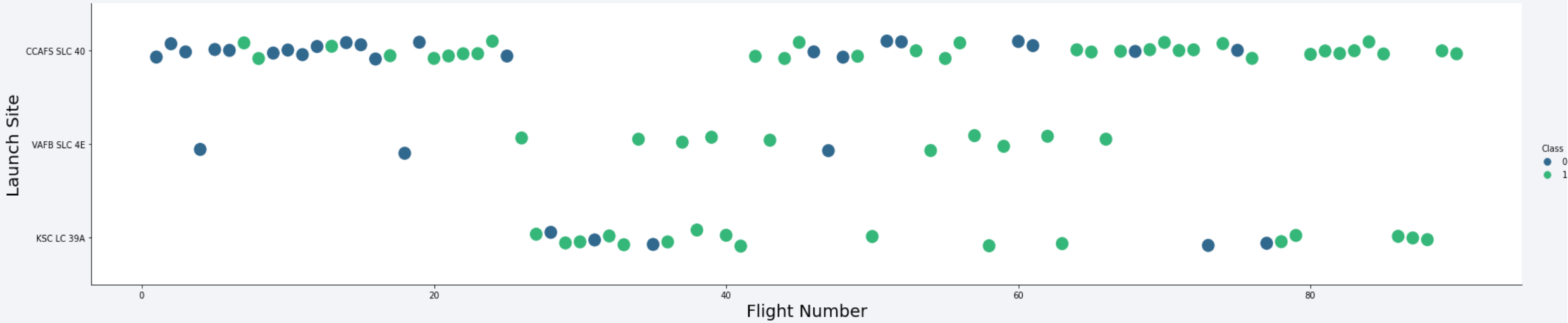
- Below image is output of Model based on available Data. In that we can see all 4 methods () are having more than 80% success ration in prediction.
- We can use any of them model for prediction.




Section 2
Insights drawn from EDA

Flight Number vs. Launch Site

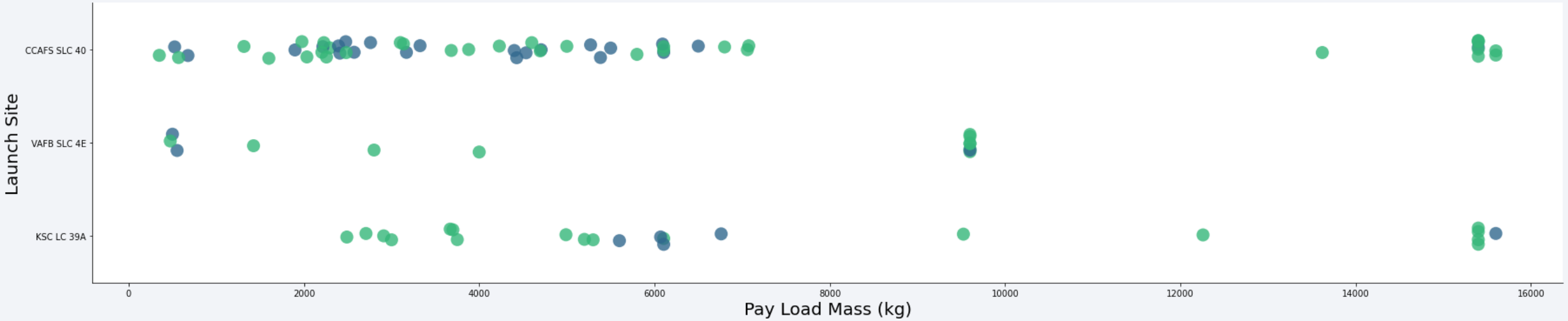
Green indicates successful launch; Purple indicates unsuccessful launch.



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.

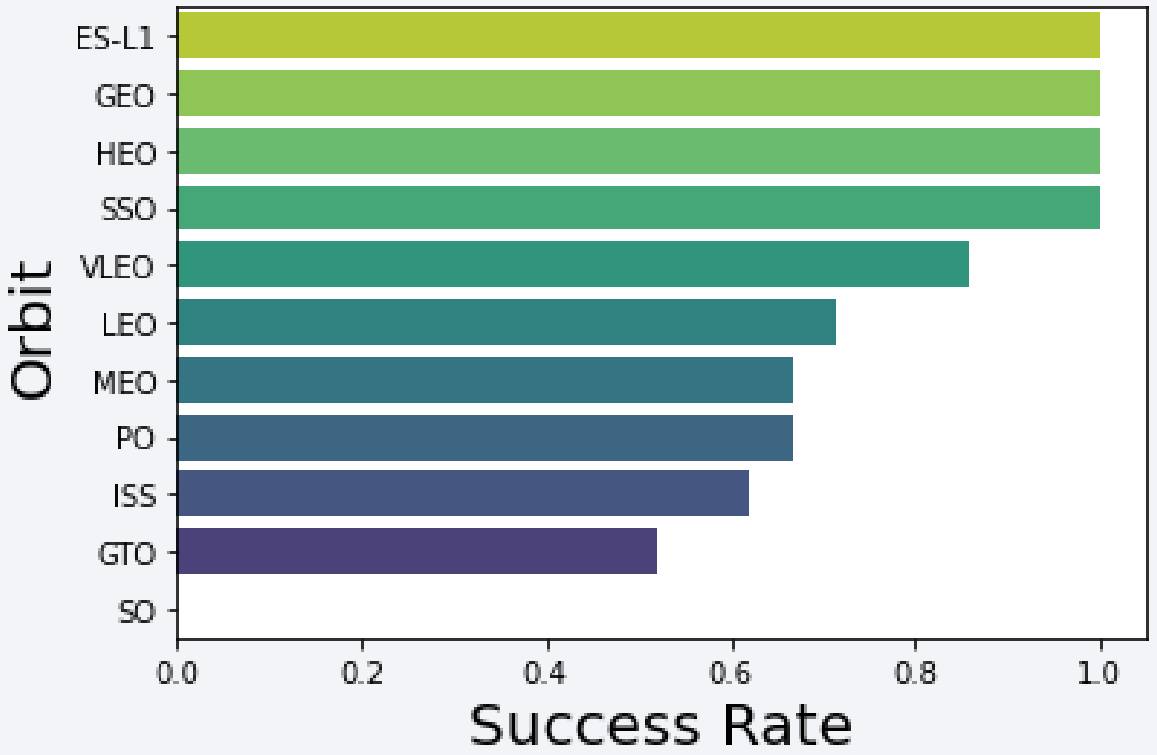


Payload mass appears to fall mostly between 0-6000 kg. Different launch sites also seem to use different payload mass.

Success Rate vs. Orbit Type

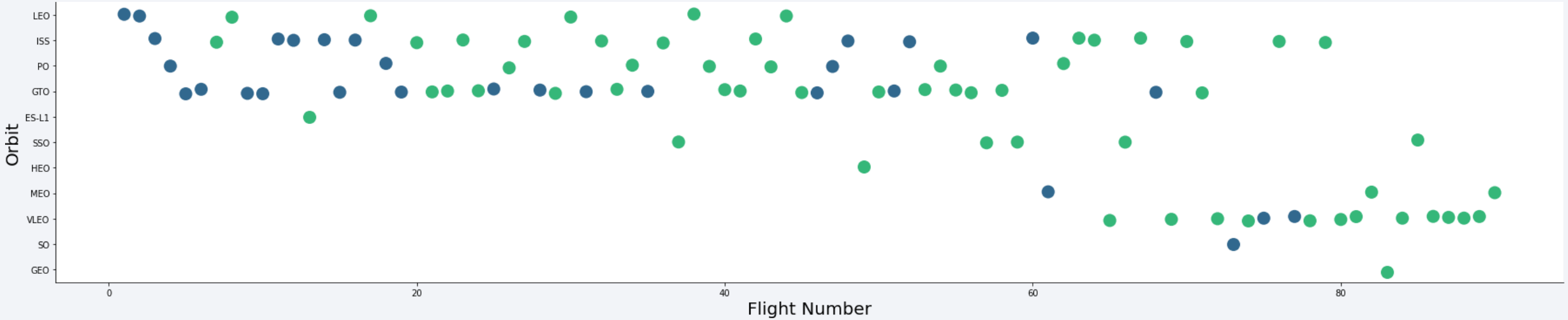
- Success Rate Scale with 0 as 0%
- 0.6 as 60% 1 as 100%

- 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

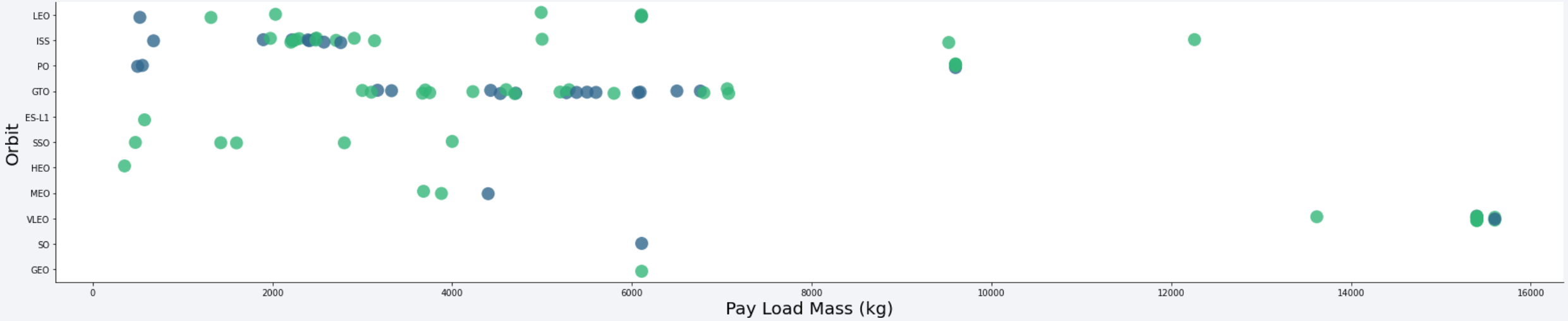
Green indicates successful launch; Purple indicates unsuccessful launch.



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

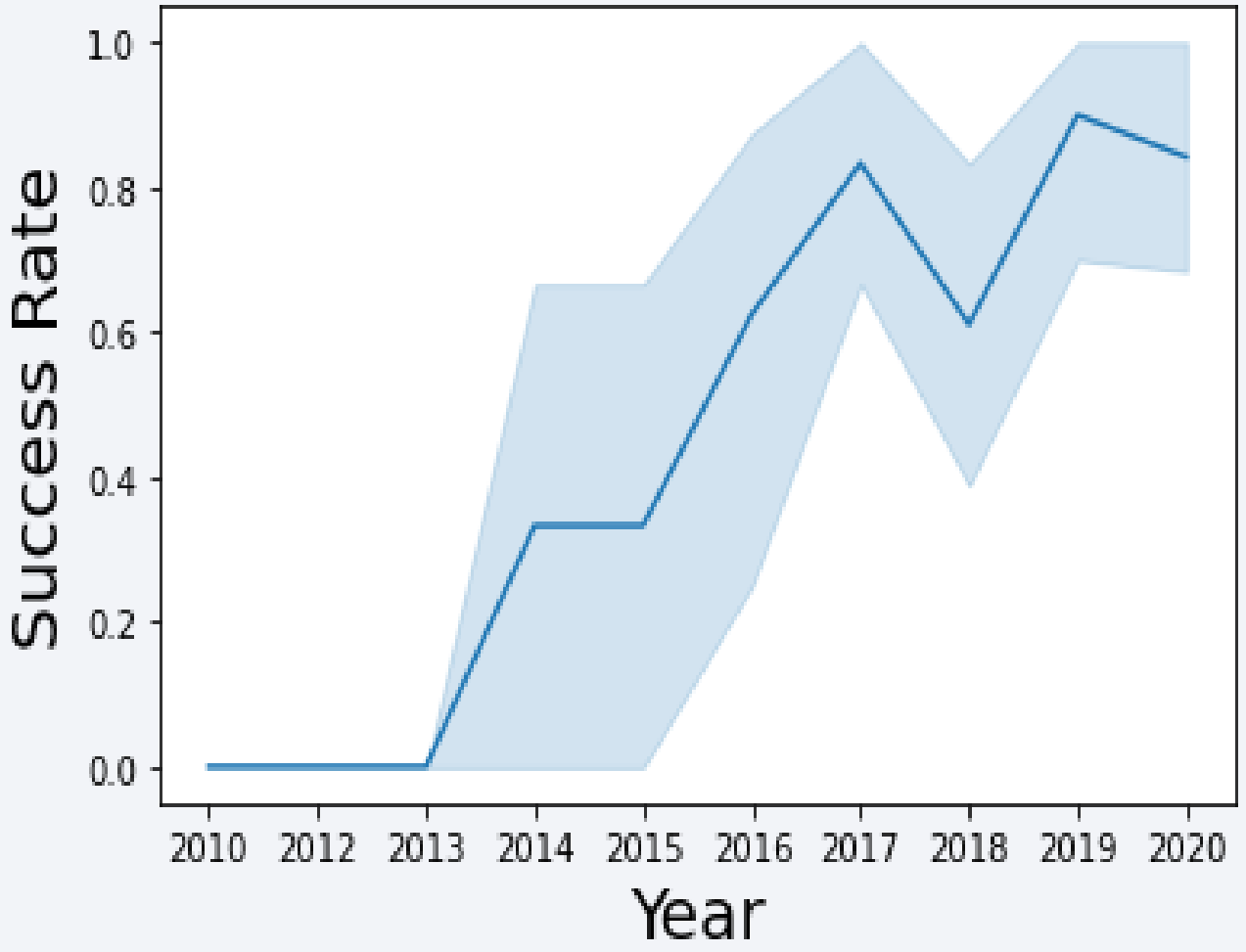
Green indicates successful launch; Purple indicates unsuccessful launch.



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

- 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%



All Launch Site Names

- CCAFS SLC-40 and CCAFSSLC-40 likely all represent the same
- CCAFS LC-40 was the previous name. Likely only 3 unique launch_site values:
  - CCAFS SLC-40,
  - KSC LC-39A,
  - VAFB SLC-4E

```
%%sql
select Distinct launch_site from SPACX
```

✓ 0.7s

```
 * ibm_db_sa://gtc93081:***@764264db-9824-4b7c-82df-4
Done.
```

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

Launch Site Names Begin with 'CCA'

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

Display 5 records where launch sites begin with the string 'CCA'

```
%%sql
select * from SPACX where launch_site like 'CCA%' limit 5
```

Python

```
 * ibm_db_sa://gtc93081:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/bludb
Done.
```

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 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).

```
[34]
%%sql
select sum(payload_mass__kg_) as sum from SPACX where customer like 'NASA (CRS)'
```

```
 * ibm_db_sa://gtc93081:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud
Done.
```

SUM
45596

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

Display average payload mass carried by booster version F9 v1.1

```
%%sql
select avg(payload_mass__kg_) as Average from SPACX where booster_version like 'F9 v1.1%'
```

Python

```
 * ibm_db_sa://gtc93081:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/bludb
Done.
```

average
2534

First Successful Ground 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.

```
%%sql
select min(date) as Date from SPACX where mission_outcome like 'Success'
```

```
[36]
...  * ibm_db_sa://gtc93081:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb
Done.
```

DATE
2010-06-04

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.

List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

```
%%sql
select booster_version from SPACX where (mission_outcome like 'Success')
AND (payload_mass__kg_ BETWEEN 4000 AND 6000) AND (landing__outcome like 'Success (drone ship)')
```

Python

```
 * ibm_db_sa://gtc93081:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/bludb
Done.
```

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

Total Number of Successful and Failure Mission Outcomes

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

List the total number of successful and failure mission outcomes

```
%%sql
SELECT mission_outcome, count(*) as Count FROM SPACX GROUP by mission_outcome ORDER BY mission_outcome
```

```
 * ibm_db_sa://gtc93081:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/bludb
Done.
```

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

Boosters Carried Maximum Payload

```
%%sql
select booster_version from SPACX where
payload_mass__kg_=(select max(payload_mass__kg_) from SPACX)
```

```
 * ibm_db_sa://gtc93081:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/bludb
Done.
```

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

- 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

- 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. One happened in January another one in April both has same booster version and same launch site.

List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

```
%%sql
select MONTHNAME(DATE) as Month, landing__outcome, booster_version, launch_site
from SPACX where DATE like '2015%' AND landing__outcome like 'Failure (drone ship)'
```

```
 * ibm_db_sa://gtc93081:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/bludb
Done.
```

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

Rank Landing Outcomes 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 are two types of successful landing outcomes: drone ship and ground pad landings.
- There were 8 successful landings in total during this time period

Rank the count of landing outcomes (such as Failure (drone ship) or Su
03-20, in descending order

```
%%sql
select landing__outcome, count(*) as count from SPACX
where Date >= '2010-06-04' AND Date <= '2017-03-20'
GROUP by landing__outcome ORDER BY count Desc
```

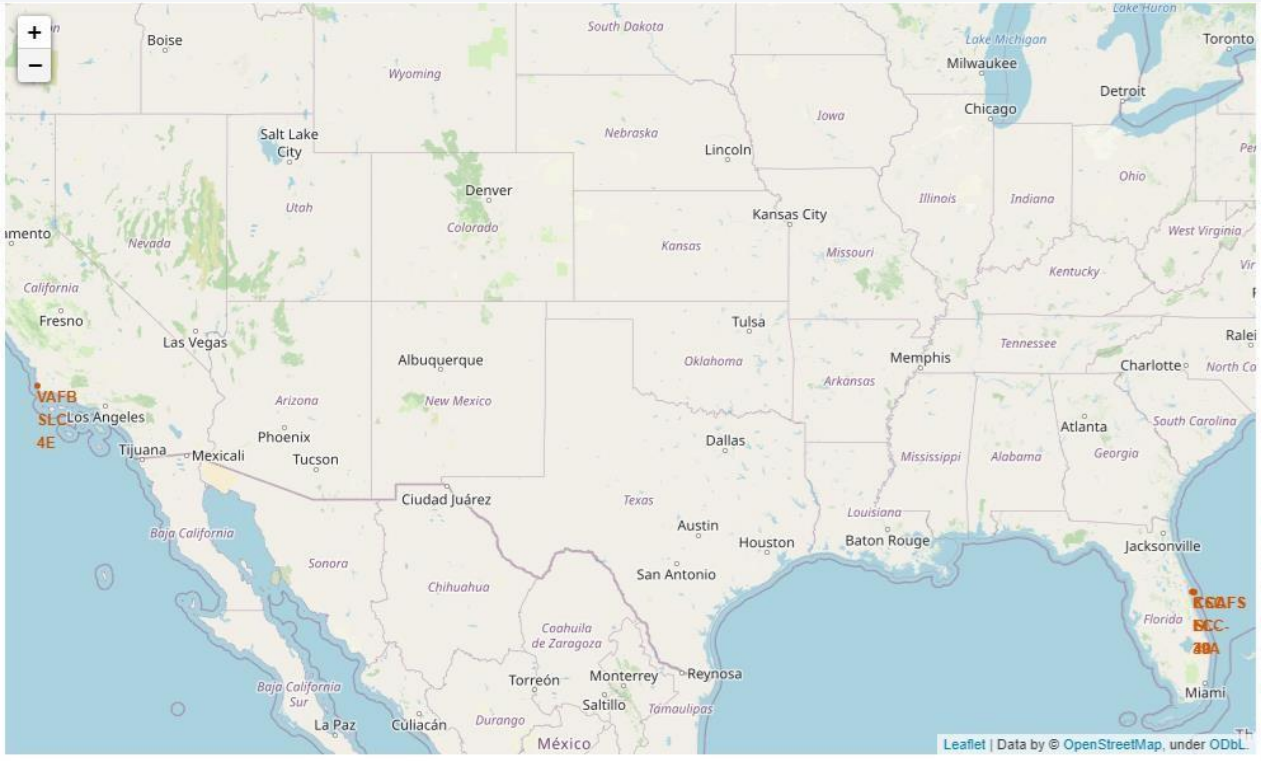
```
* ibm_db_sa://gtc93081:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1c
Done.
```

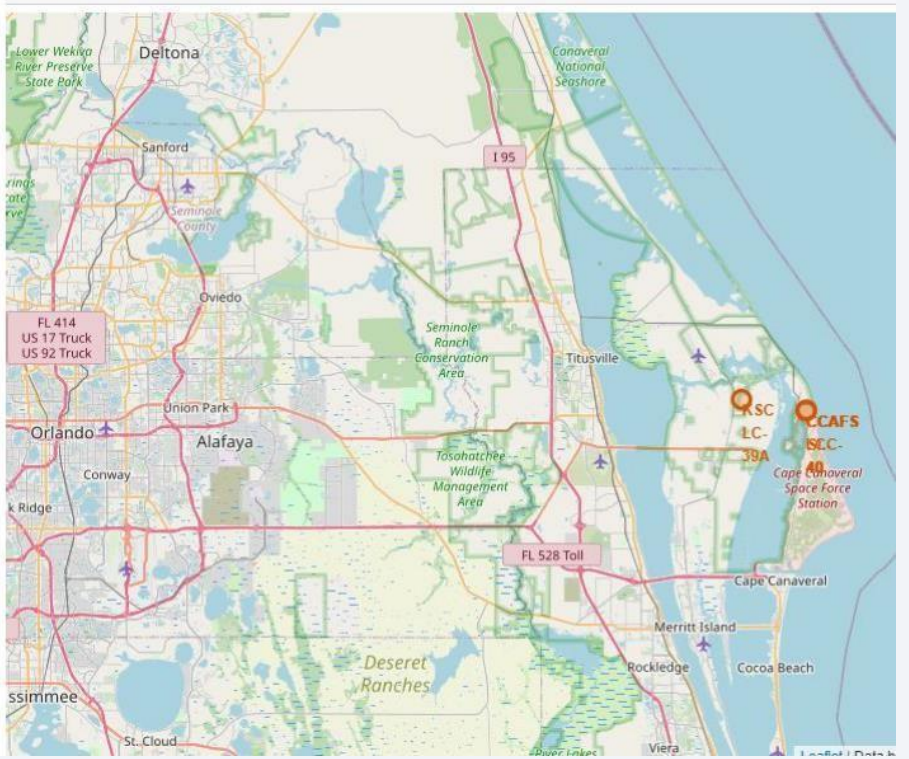
landing__outcome	COUNT
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3

Section 3

Launch Sites Proximities Analysis

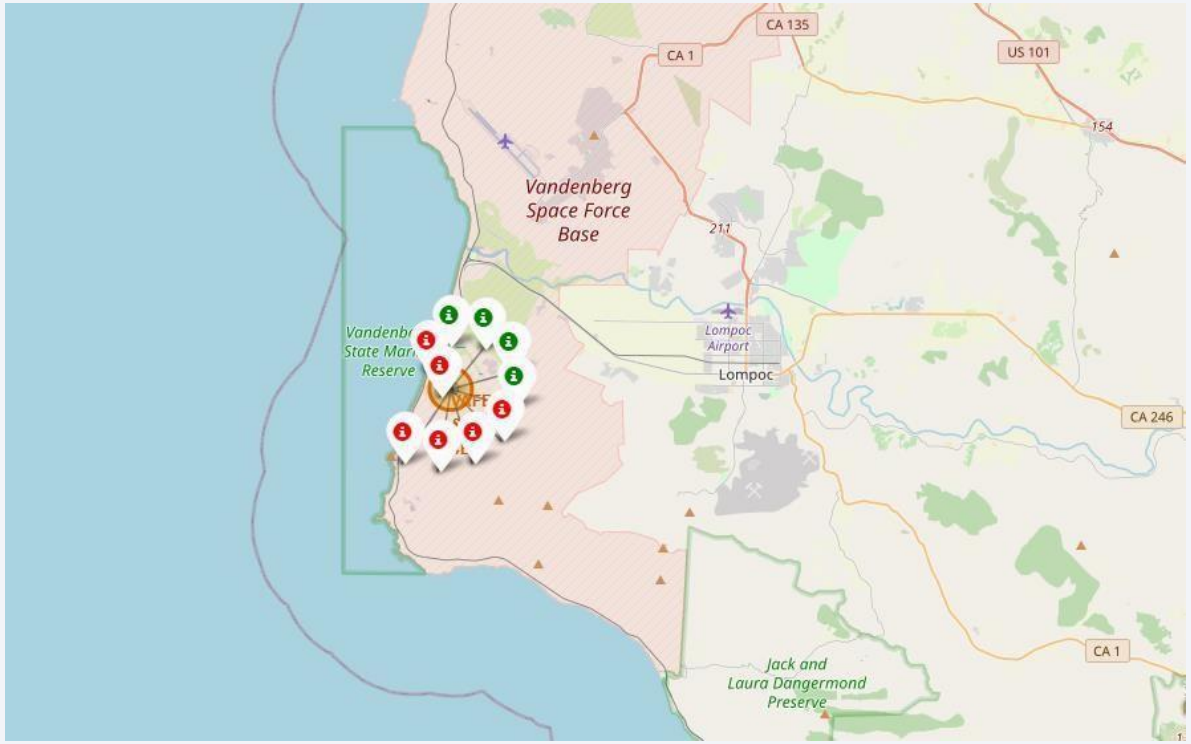
Launch Site Locations





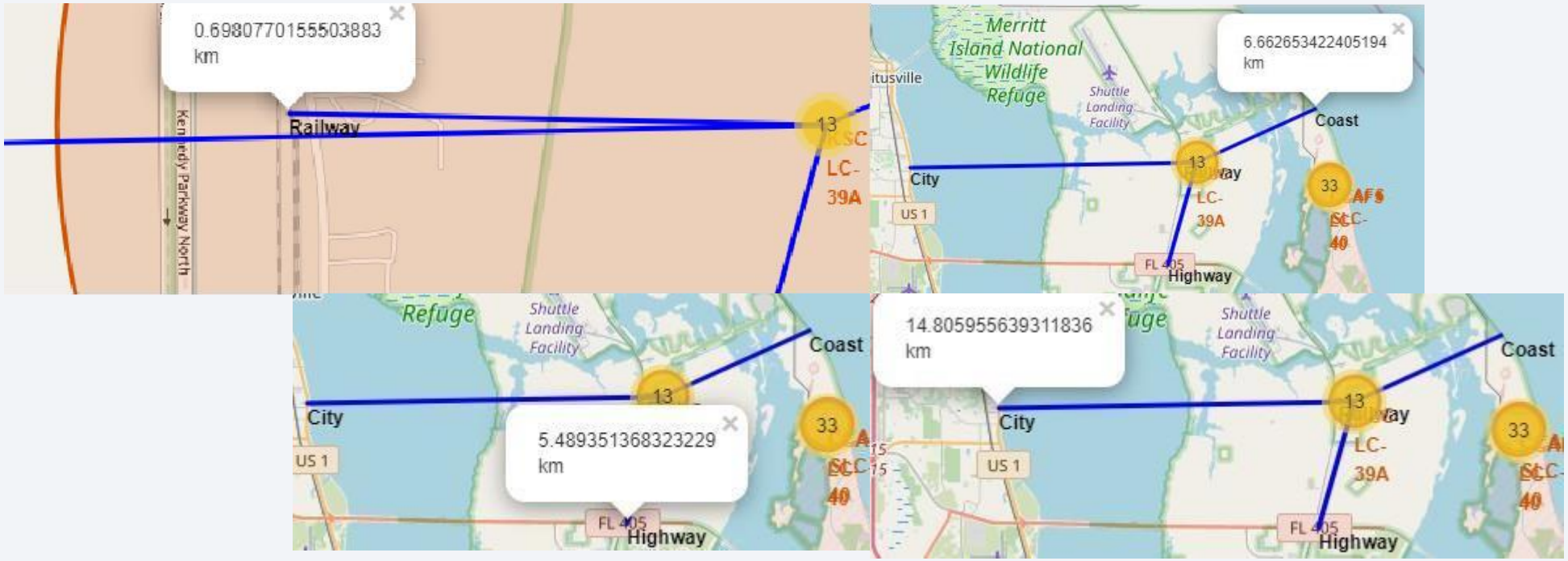
The left map shows all launch sites relative US map. The right map shows the two Florida launch sites since they are very close to each other. All launch sites are near the ocean.

Color-Coded Launch Markers

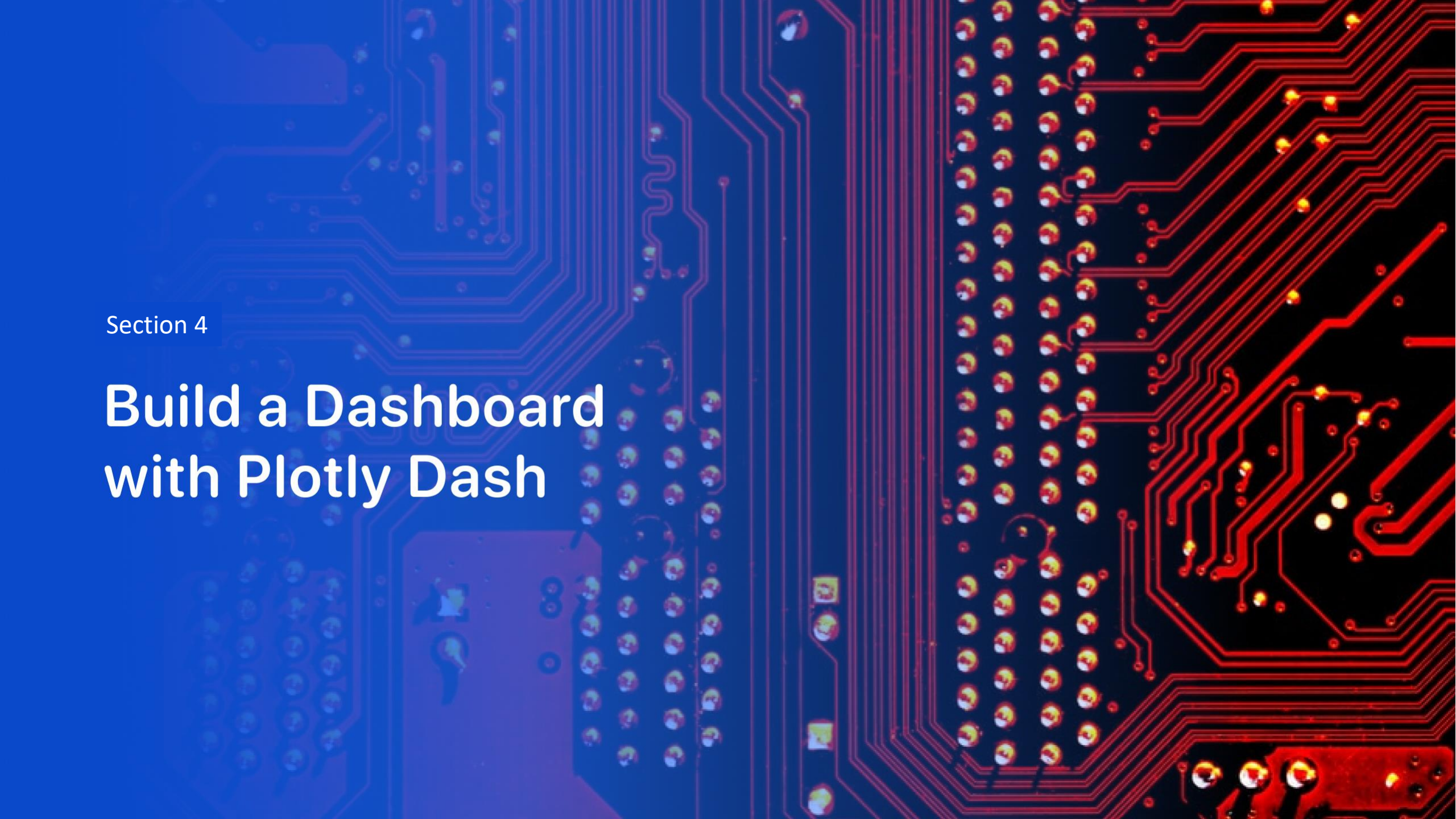


Clusters on Folium map can be clicked on to display each successful landing (green icon) and failed
landing (red icon). In this example VAFB SLC-4E shows 4 successful landings and 6 failed landings.

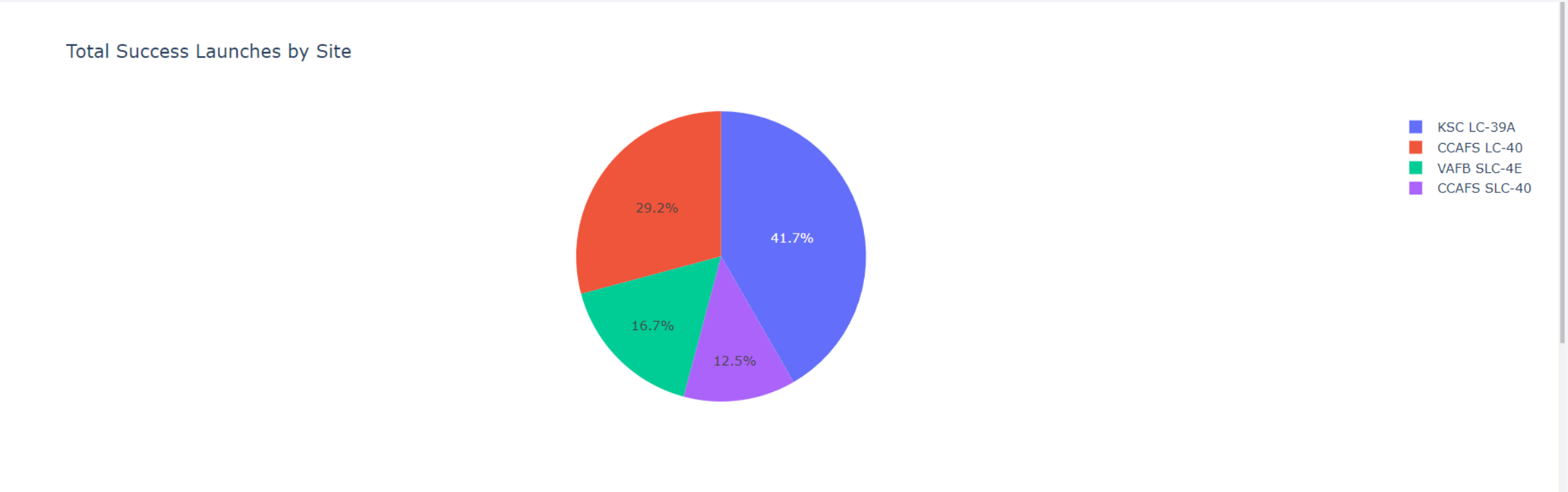
Key Location Proximities



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.

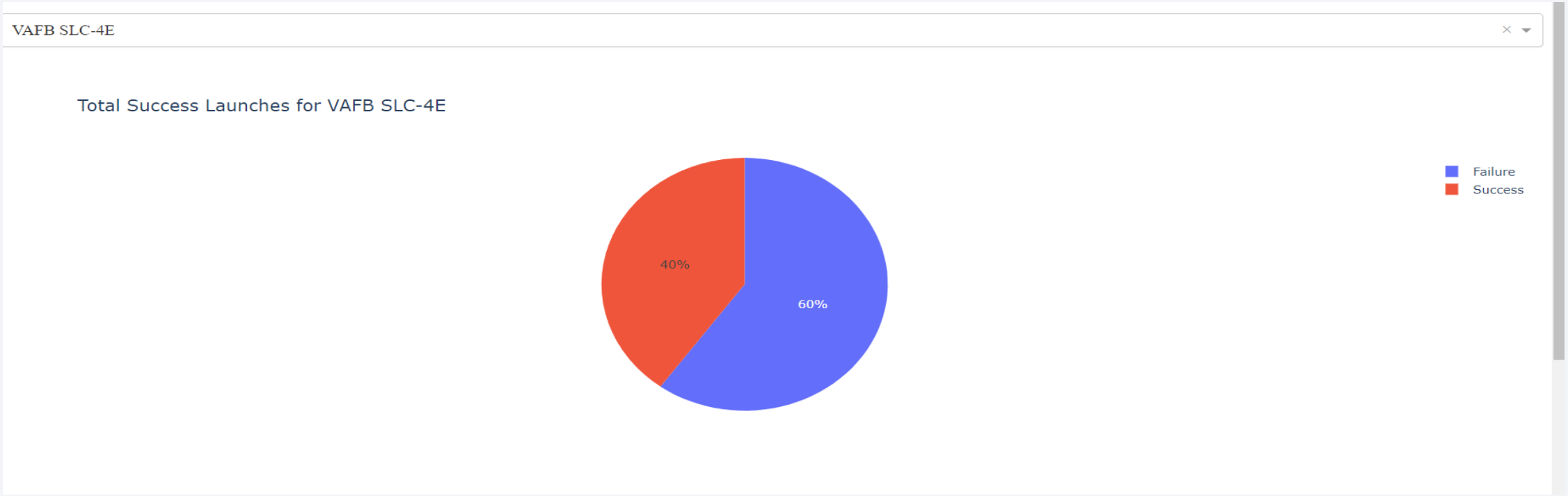

Section 4
Build a Dashboard with Plotly Dash

Successful Launches Across Launch Sites



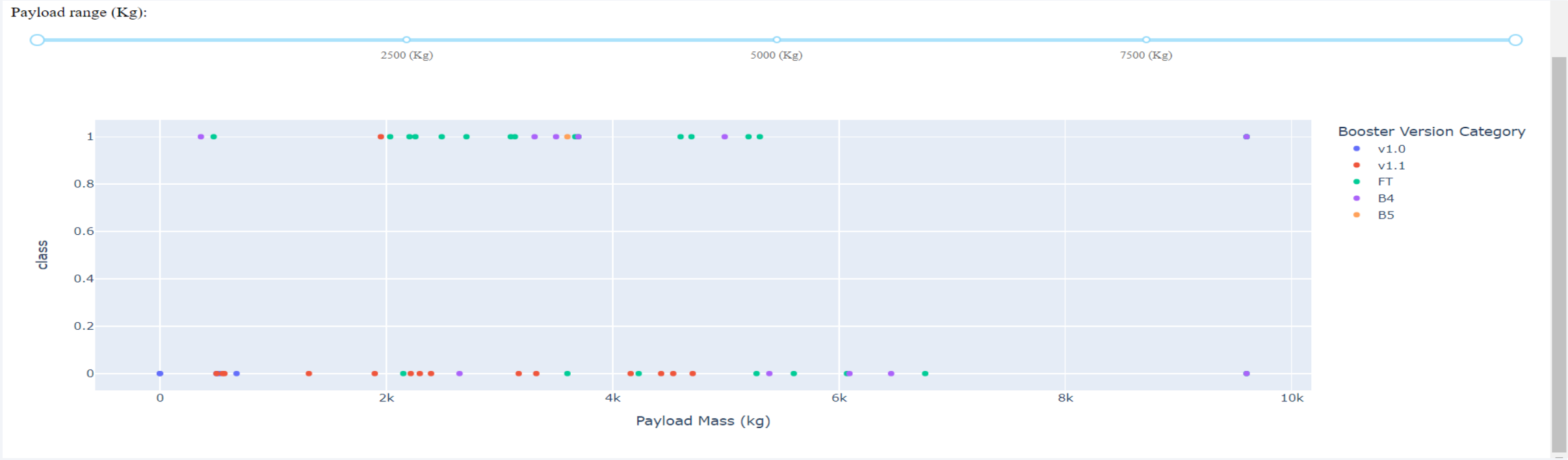
This is the distribution of successful landings across all launch sites

Total Success Rate by VAFB SLC -4E site



KSC LC-39A has the highest success rate with 4 successful landings out of 10..

Payload Mass vs. Success vs. Booster Version Category



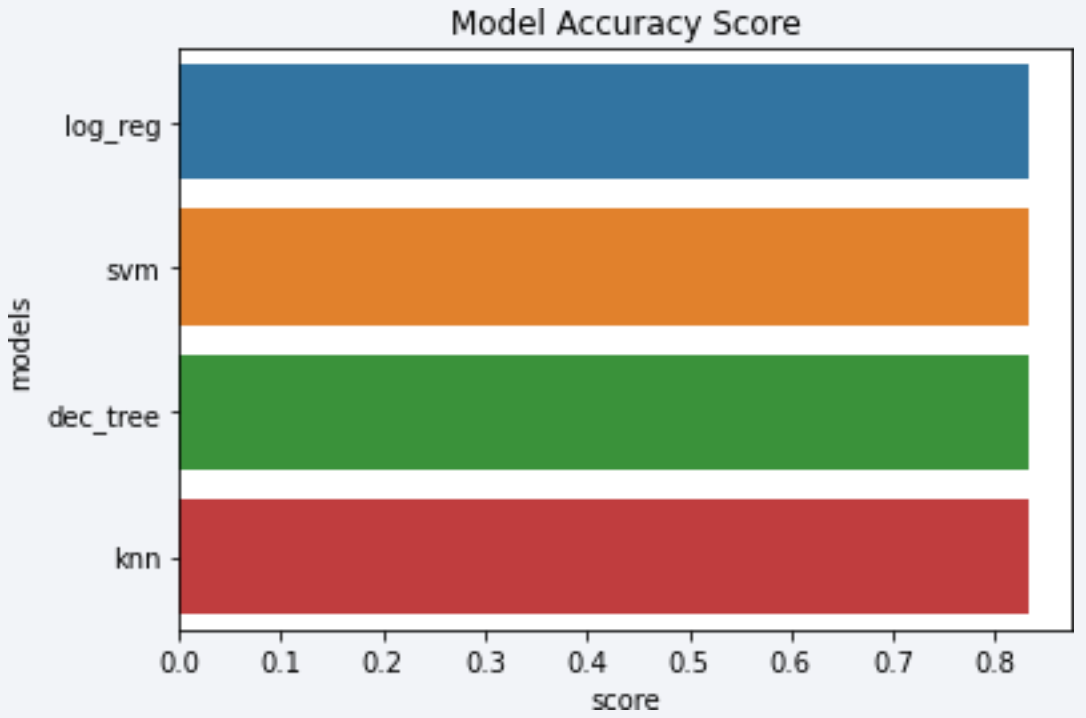
Plotly dashboard has a Payload range selector. However, this is set from 0-10000 instead of the max Payload of 15600. Class indicates 1 for successful landing and 0 for failure. Scatter plot also accounts for booster version category in color and number of launches in point size. In this particular range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

Section 5

Predictive Analysis (Classification)

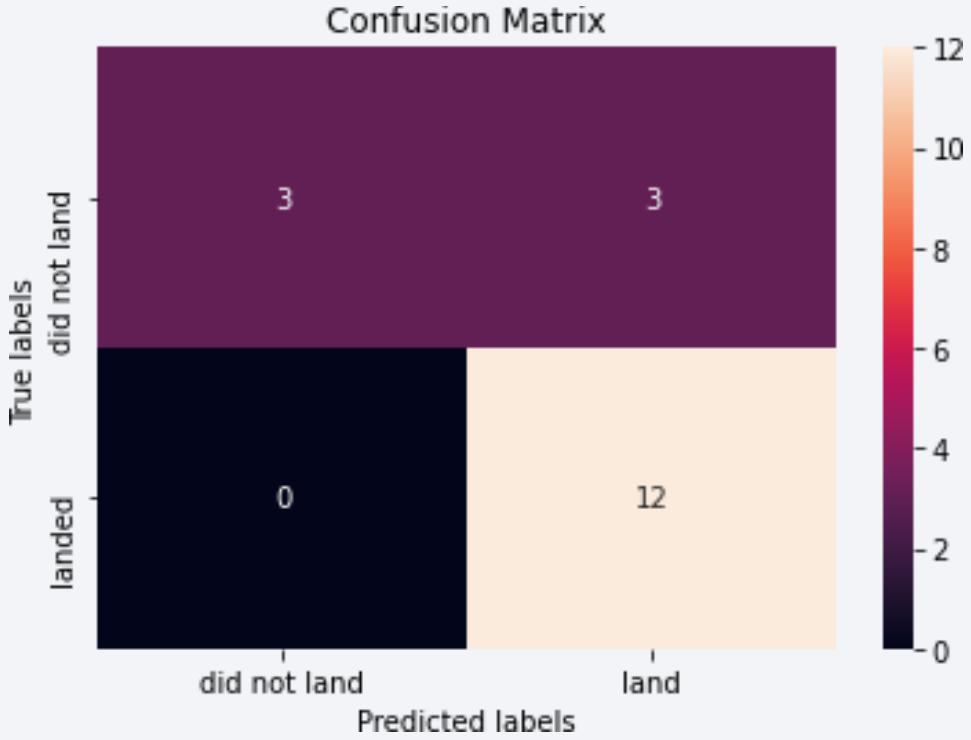
Classification Accuracy

- 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

- Show the confusion matrix of the best performing model 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

- Wikipedia page of Data :
- Thanks to IBM for Giving notebook and source of Data with Space-X API
- Thanks to Coursera and all instructors
- My git repo :- https://github.com/meetshahcode/Capstone

Thank you!