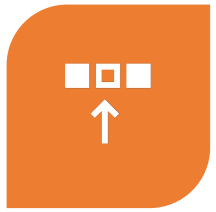
Winning Space Race with Data Science

Jenna H C

22nd September 2023

Presented by Jenna H C

Outline



EXECUTIVE SUMMARY

INTRODUCTION

METHODOLOGY

RESULTS

CONCLUSION

APPENDIX

Executive Summary

Summary of methodologies

Introduction

Project background and context	Space X advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because Space X can reuse the first stage. Therefore, if we can determine if the first stage will land, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against space X for a rocket launch. This goal of the project is to create a machine learning pipeline to predict if the first stage will land successfully.

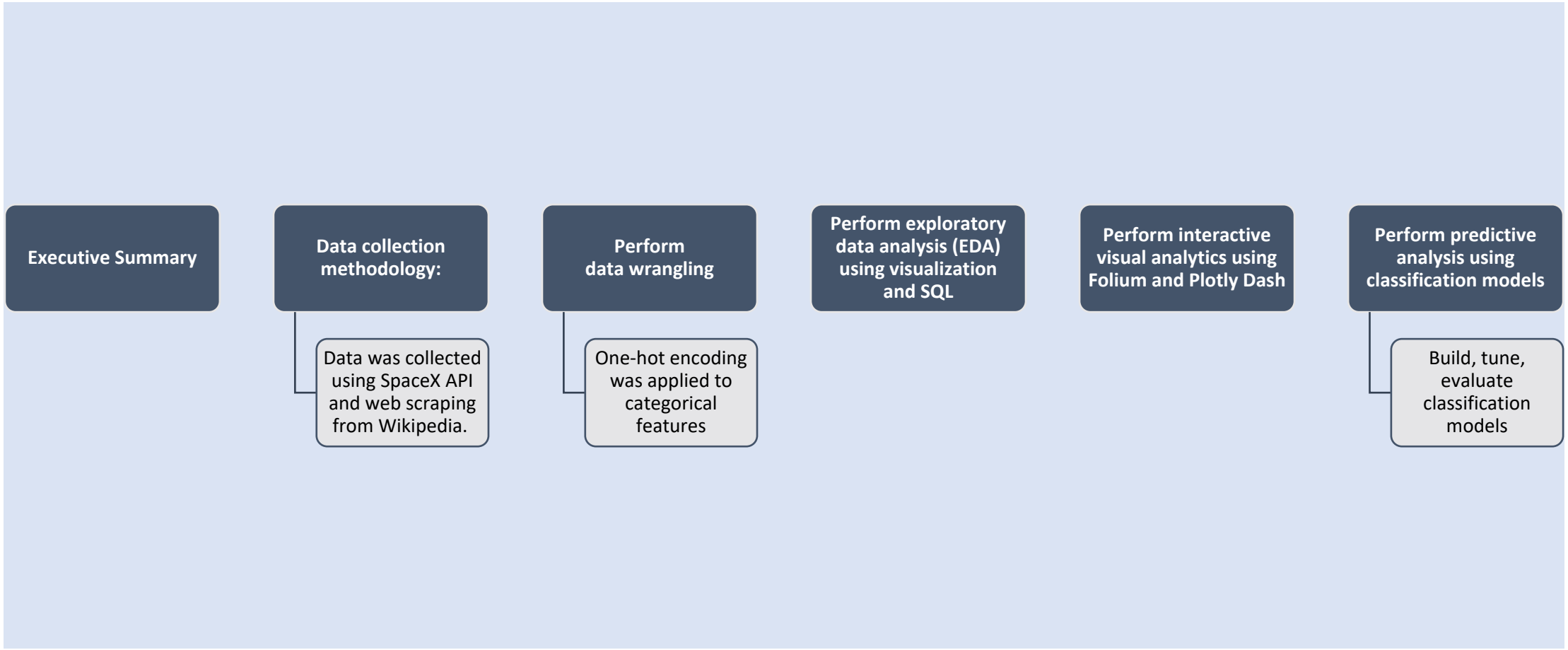
Section 1

Methodology

9/23/23

Presented by Jenna H C

Methodology



Data Collection

Data collection process involved a combination of API requests from SpaceX REST API and Web Scraping data from a table in SpaceX's Wikipedia entry.

We had to use both of these data collection methods in order to get complete information about the launches for a more detailed analysis.

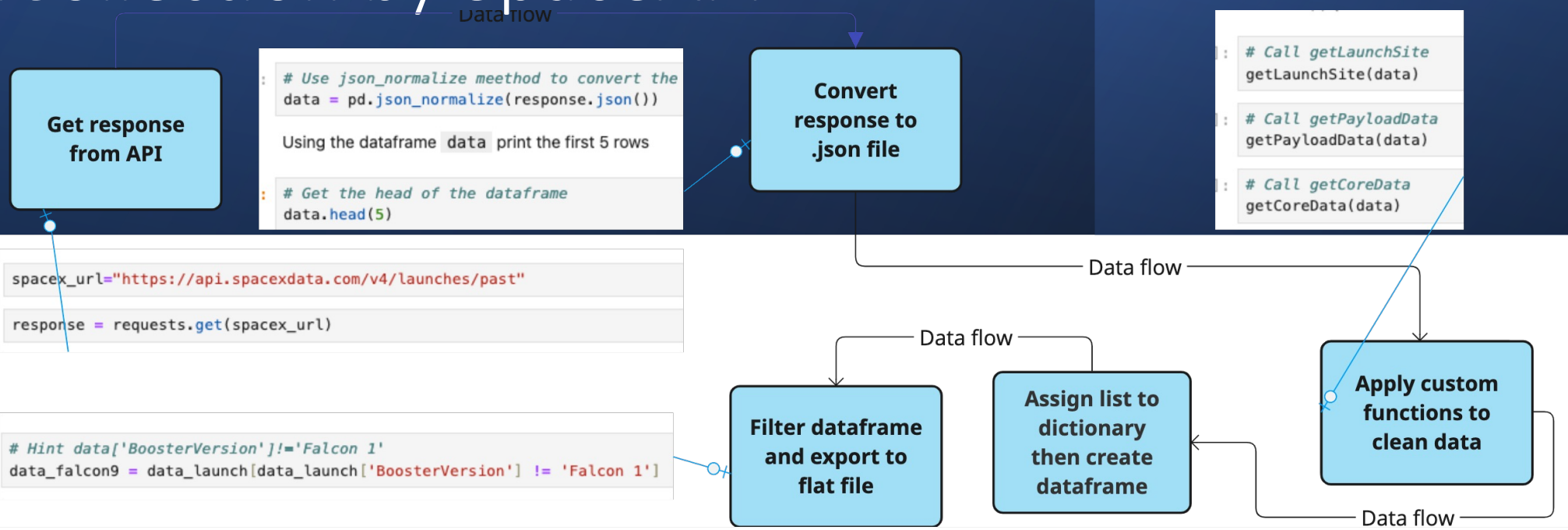
Data Columns are obtained by using SpaceX REST API:

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

Data Columns are obtained by using Wikipedia Web Scraping:

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

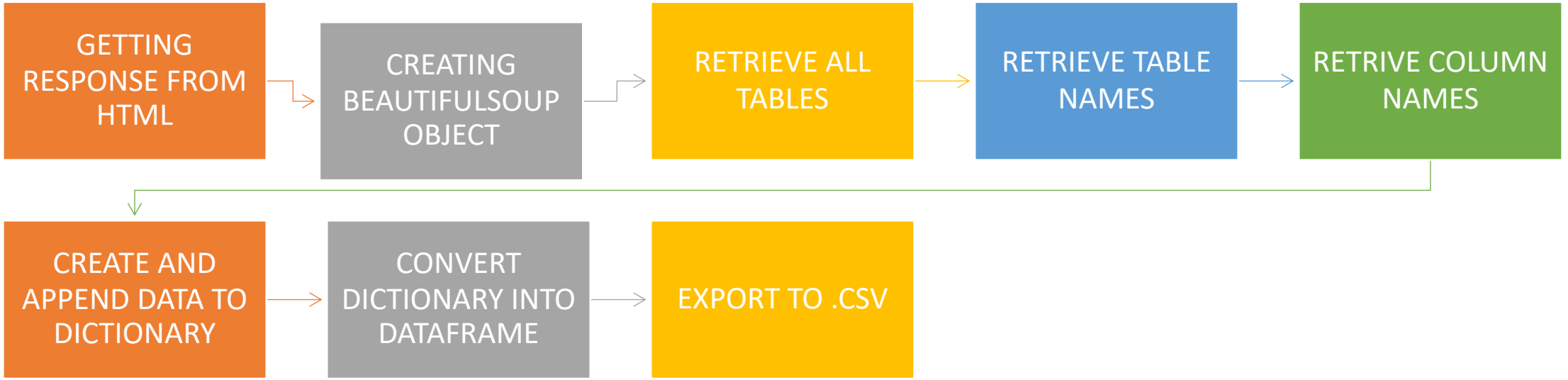
DataCollection by SpaceAPI




```
data_falcon9 = data_falcon9.copy()
data_falcon9['FlightNumber'] = list(range(1, data_falcon9.shape[0] + 1))
data_falcon9
```

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	Longitude	Latitu
4	1	2010-06-04	Falcon 9	6123.547647	LEO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0003	-80.577366	28.5618
5	2	2012-05-22	Falcon 9	525.000000	LEO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0005	-80.577366	28.5618
6	3	2013-03-01	Falcon 9	677.000000	ISS	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0007	-80.577366	28.5618
7	4	2013-09-29	Falcon 9	500.000000	PO	VAFB SLC 4E	False Ocean	1	False	False	False	None	1.0	0	B1003	-120.610829	34.6320
8	5	2013-12-03	Falcon 9	3170.000000	GTO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B1004	-80.577366	28.5618
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	

Data Collection by web scraping




```
=pd.DataFrame(launch_dict)
```

```
.head()
```

Flight No.	Launch site	Payload	Payload mass	Orbit	Customer	Launch outcome	Version Booster	Booster landing	Date	T
1	CCAFS	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success\n	F9 v1.0B0003.1	Failure	4 June 2010	18
2	CCAFS	Dragon	0	LEO	NASA	Success	F9 v1.0B0004.1	Failure	8 December 2010	15
3	CCAFS	Dragon	525 kg	LEO	NASA	Success	F9 v1.0B0005.1	No attempt\n	22 May 2012	07
4	CCAFS	SpaceX CRS-1	4,700 kg	LEO	NASA	Success\n	F9 v1.0B0006.1	No attempt	8 October 2012	00
5	CCAFS	SpaceX CRS-2	4,877 kg	LEO	NASA	Success\n	F9 v1.0B0007.1	No attempt\n	1 March 2013	15

Data Wrangling

We performed exploratory data analysis and determined the training labels.

Data wrangling is the process of cleaning and unifying messy and complex data sets for easy access and analysis.

We created landing outcome label from outcome column and exported the results to csv.

```
[2]: df.head(5)
```

[2]:

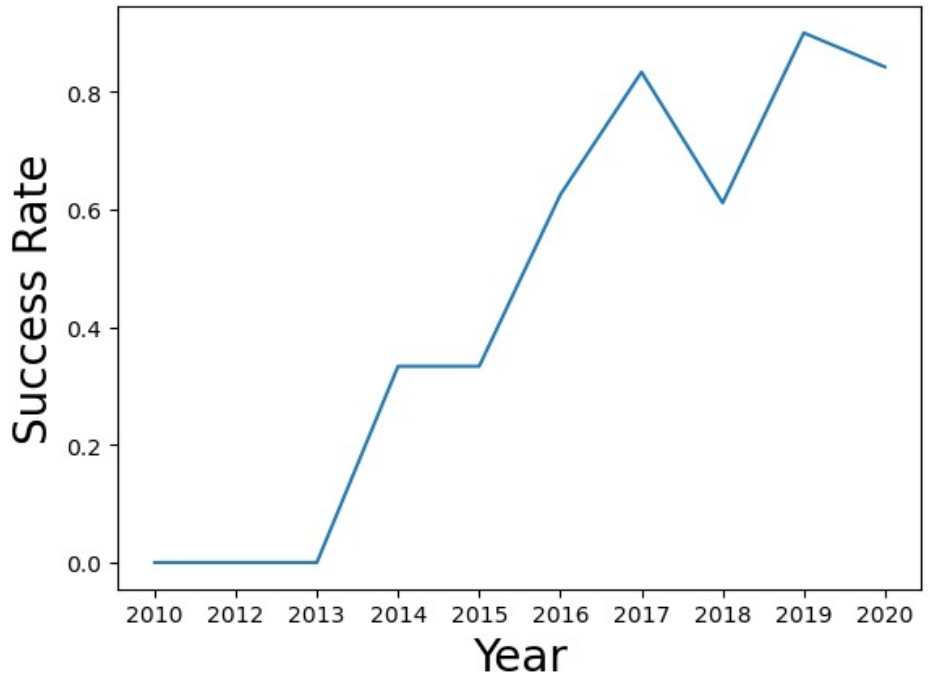
	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block
0	1	2010-06-04	Falcon 9	6104.959412	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0
1	2	2012-05-22	Falcon 9	525.000000	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0
2	3	2013-03-01	Falcon 9	677.000000	ISS	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0
3	4	2013-09-29	Falcon 9	500.000000	PO	VAFB SLC 4E	False Ocean	1	False	False	False	NaN	1.0
4	5	2013-12-03	Falcon 9	3170.000000	GTO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0

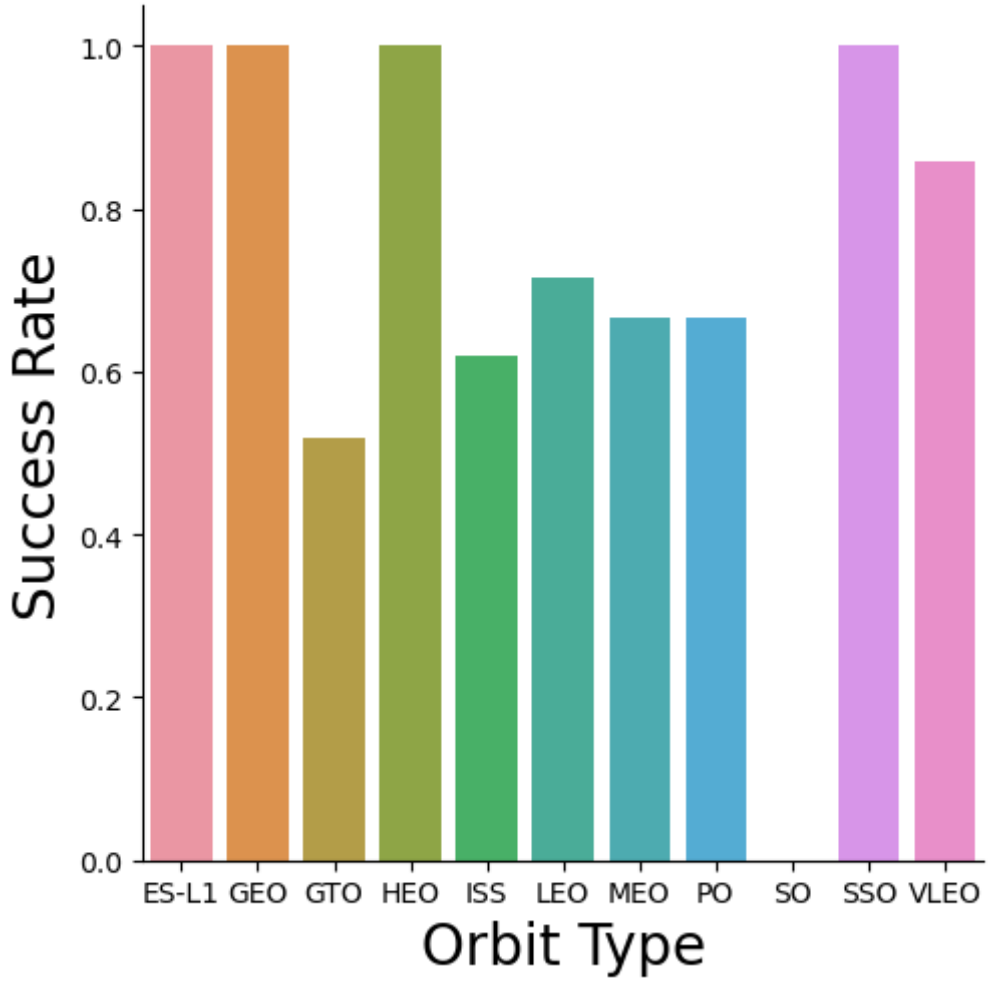
We calculated the number of launches at each site, and the number and occurrence of each orbits

We can use the following line of code to determine the success rate:

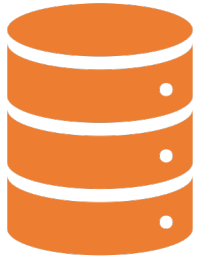
EDA with Data Visualization

- We explored the data by visualizing the relationship between flight number and launch Site, payload and launch site, success rate of each orbit type, flight number and orbit type, the launch success yearly trend.

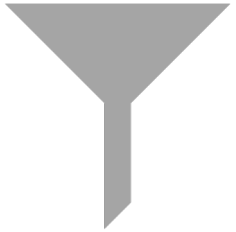




EDA with SQL



SOL is an indispensable tool for data scientists and analysts as most of the real-world data is stored in databases. It's not only the standard language for relational database operations, but also an incredibly powerful tool for analyzing data and drawing useful insights from. Here we loaded the SpaceX dataset into a SQlite database without leaving the Jupyter notebook.



We applied EDA with SQL to get insight from the data. We wrote queries to find out for instance:

Build an Interactive Map with Folium

Folium makes it easy to visualize data on an interactive leaflet map.

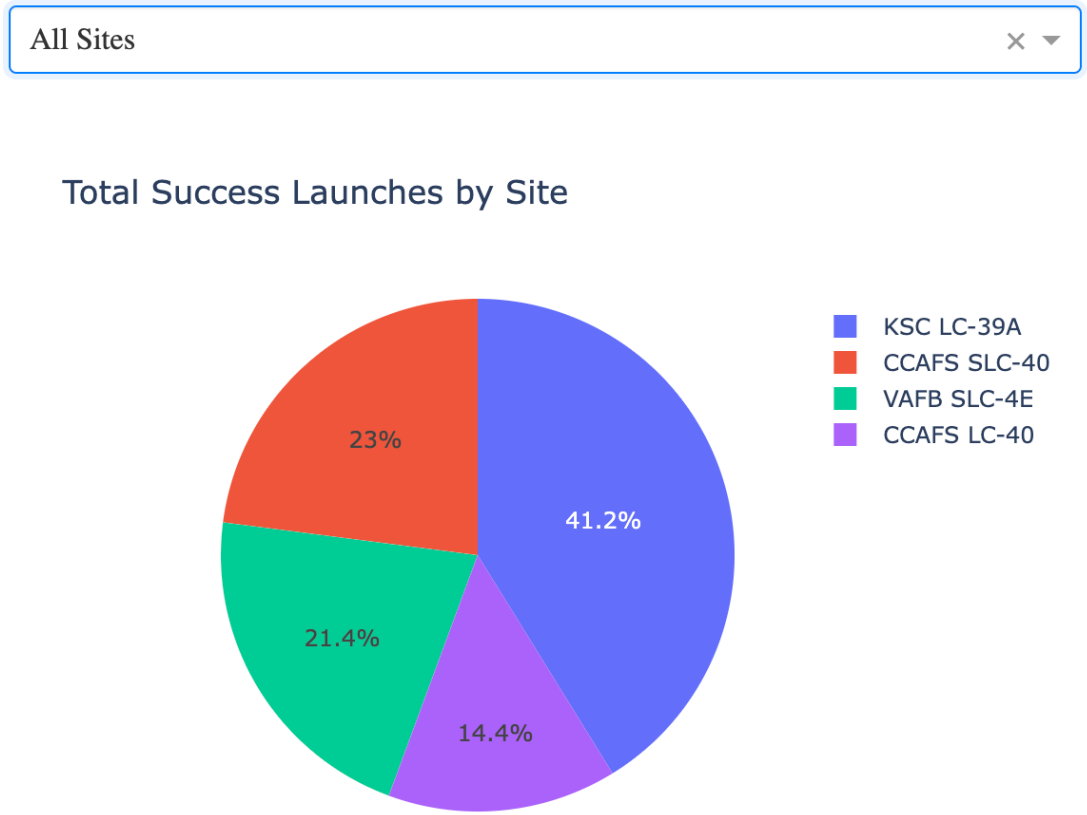
We marked all launch sites, and added map objects such as markers, circles, lines to mark the success or failure of launches for each site on the folium map.

We assigned the feature launch outcomes (failure or success) to class 0 and 1.i.e., 0 for failure, and 1 for success.

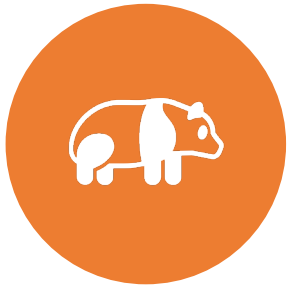
We use the color-labeled marker clusters, we identified which launch sites have relatively high success rate.

Build an Interactive Map with Plotly Dash

- We built an interactive dashboard with Plotly dash
- We created drop down menu for launch sites
- We created a rangeslider for Payload Mass range selection
- We plotted pie charts showing the total launches by a certain sites
- We plotted scatter graph showing the relationship with Outcome and Payload Mass (Kg) for the different booster version.

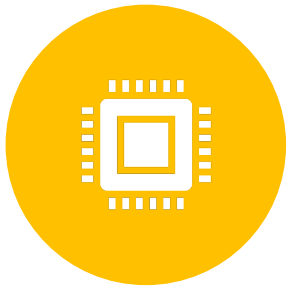


Predictive Analysis (Classification)

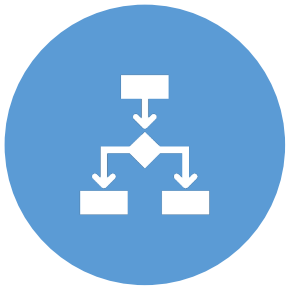


We loaded the data using numpy and pandas, transformed the data, split our data into training and testing.

We built different machine learning models and tune different hyperparameters using GridSearchCV.



We used accuracy as the metric for our models, improved the models using feature engineering and algorithm tuning.



We found the best performing classification model to the decision tree.

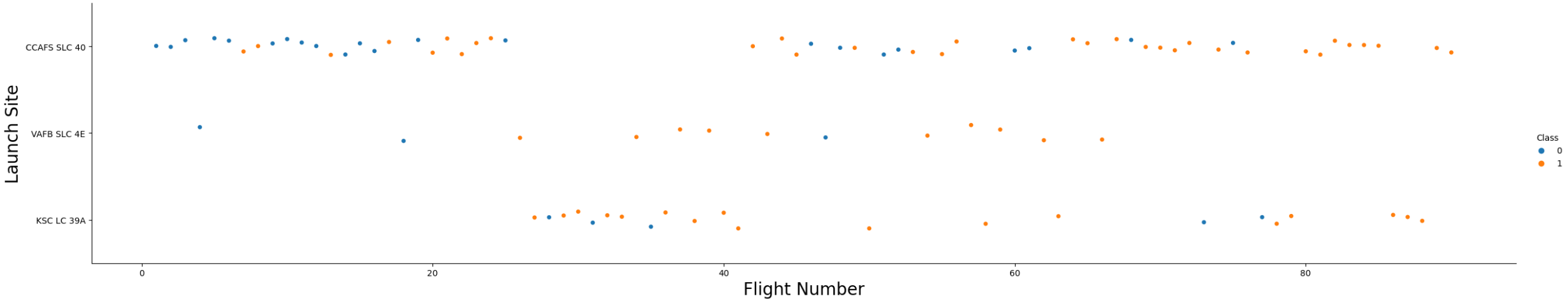
Results

- Exploratory data analysis results
- Interactive analytic results
- Predictive analysis results

Section 2

Insights drawn from EDA

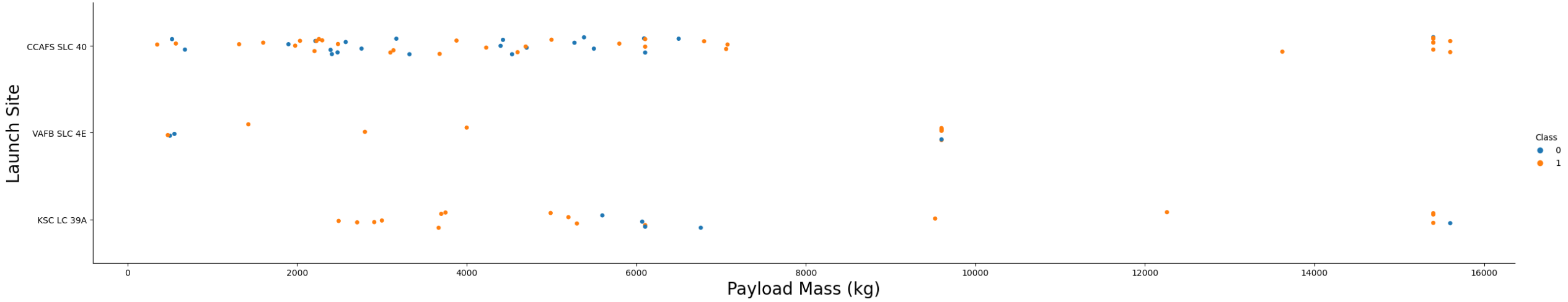
Flight Number vs. Launch Site



- **Explanation:**
- The earliest flights all failed while the latest flights all succeeded.
- The CCAFS SLC 40 launch site has about a half of all launches.
- VAFB SLC 4E and KSC LC 39A have higher success rates.
- It can be assumed that each new launch has a higher rate of success.

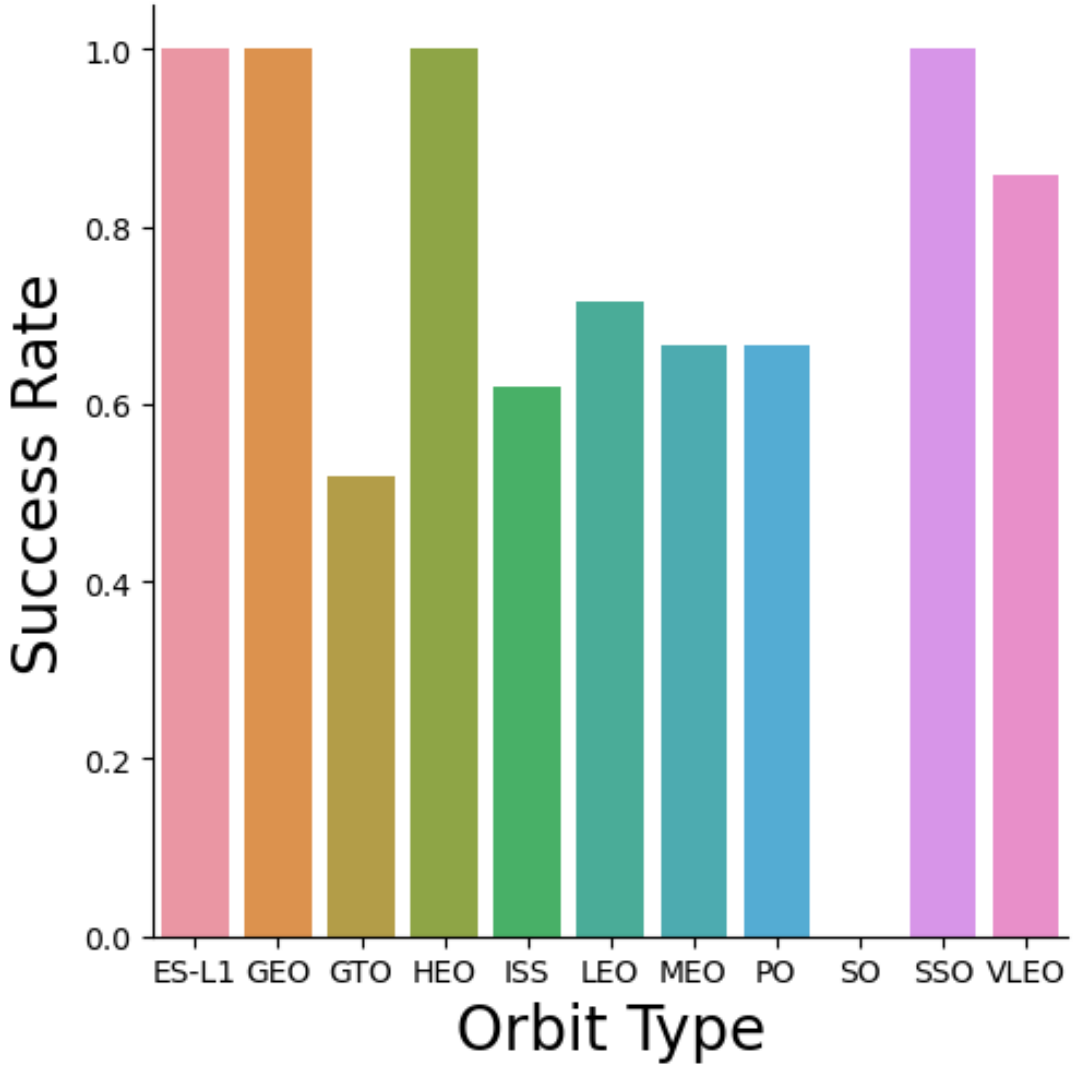
Payload vs. Launch Site

- **Explanation:**
- For every launch site the higher the payload mass, the higher the success rate.
- Most of the launches with payload mass over 7000 kg were successfull.
- KSC LC 39A has a 100% success rate for payload mass under 5500 kg too.

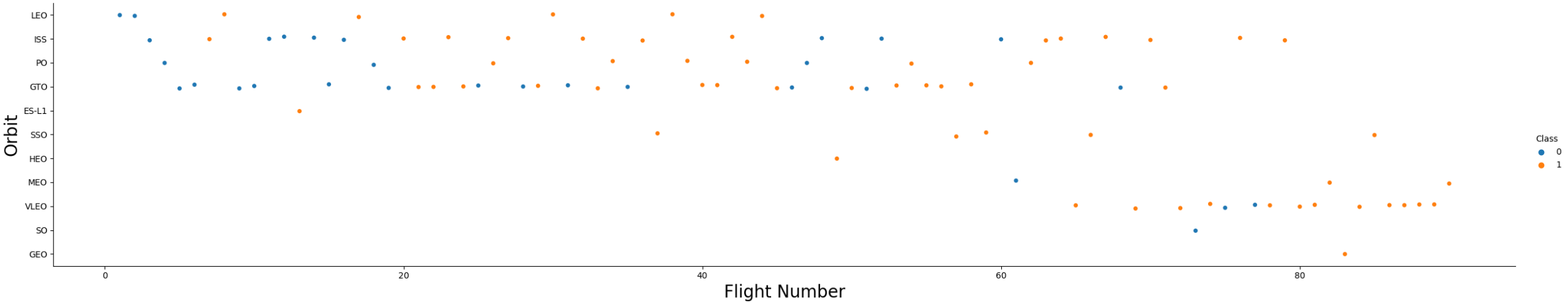


Success Rate vs. Orbit Type

- **Explanation:**
- Orbits with 100% success rate are:
  - ES-L1
  - GEO
  - HEO
  - SSO
- Orbits with 0% success rate are:
  - SO
- Orbits with success rate between 50% and 85%:
  - GTO
  - ISS
  - LEO
  - MEO
  - PO



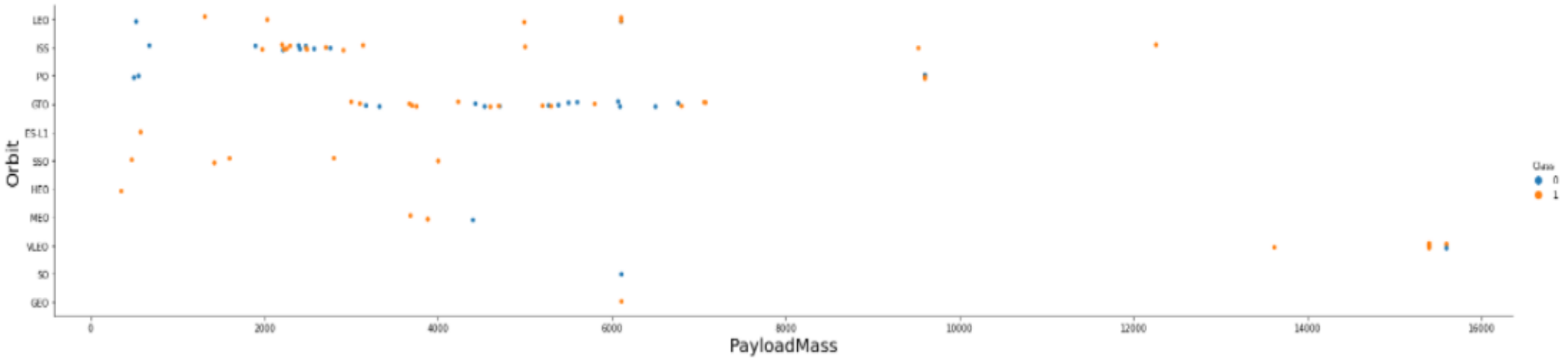
Flight Number vs. Orbit Type



- The plot below shows the Flight Number vs. Orbit type. We observe that in the LEO orbit, success is related to the number of flights whereas in the GTO orbit, there is no relationship between flight number and the orbit.

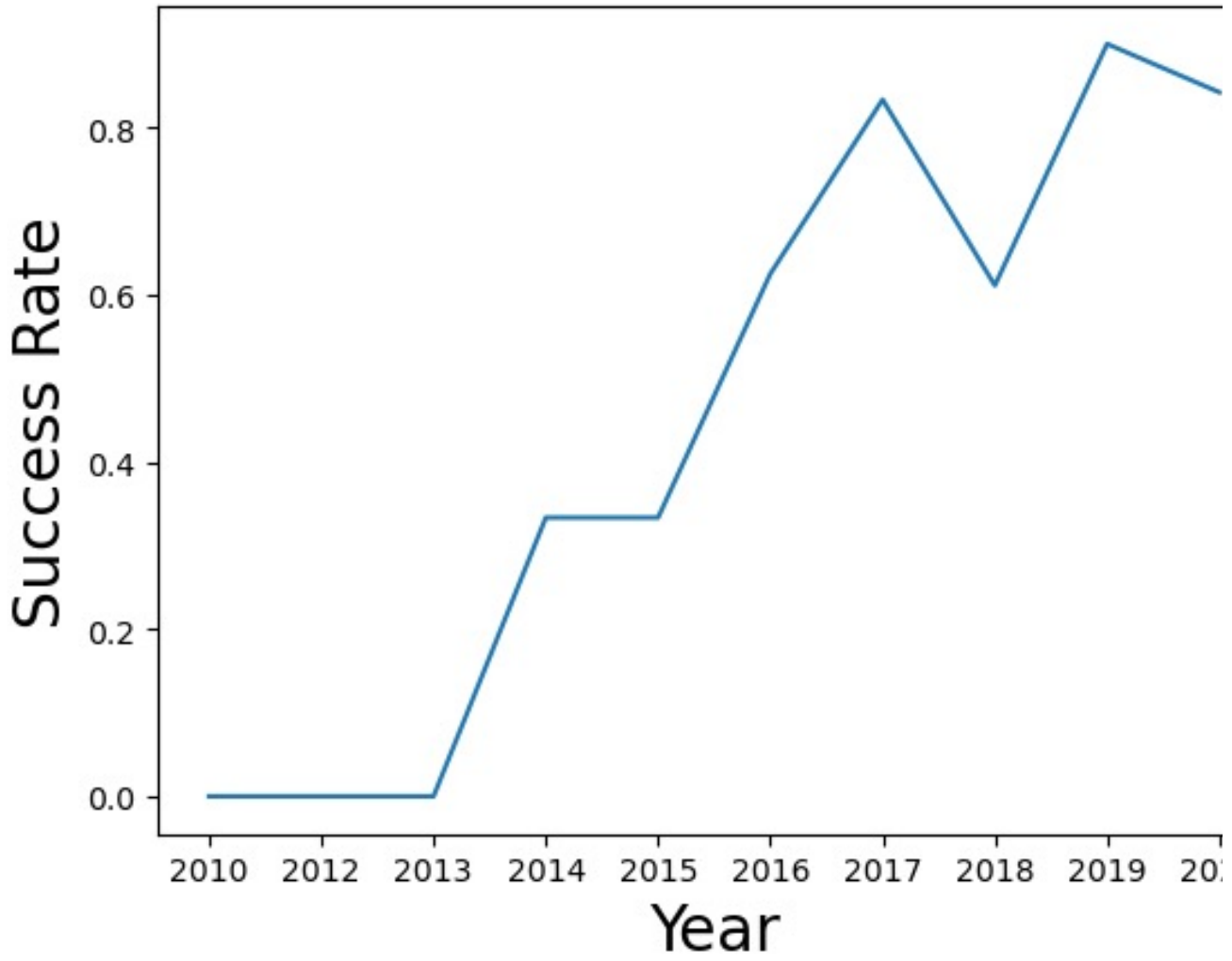
Payload vs. Orbit Type

- We can observe that with heavy payloads, the successful landing are more for PO, LEO and ISS orbits.



Launch Success Yearly Trend

- From the plot, we can observe that success rate since 2013 kept on increasing till 2020.



All Launch Site Names

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

Description:

We used the key word **DISTINCT** to show only unique launch sites from the SpaceX data.

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

```
%sql select distinct launch_site from SPACEXDATASE;
```

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

Launch Site Names Begin with 'CCA'

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

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

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-04-06	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-08-12	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
2012-08-10	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-01-03	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Description:

We used the query above to display 5 records where launch sites begin with `CCA`

Total Payload Mass

- **Description:**

We calculated the total payload carried by boosters from NASA as 45596 using the query below

Display the total payload mass carried by boosters launched by NASA (CRS)

```
%sql select sum(payload_mass__kg_) as total_payload_mass from SPACEXDATASET where customer = 'NASA (CRS)';
```

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

total_payload_mass
45596

Average Payload Mass by F9 v1.1

Description:

We calculated the average payload mass carried by booster version F9 v1.1 as 2928.4

```
%sql select avg(payload_mass__kg_) as average_payload_mass from SPACEXDATASET where booster_version like '%F9 v1.1%';
```

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

average_payload_mass
2534.6666666666665

First Successful Ground Landing Date

Description

- We observed that the dates of the first successful landing outcome on ground pad was 22nd December 2015

```
%sql select min(date) as first_successful_landing from SPACEXDATASET where landing_outcome = 'Success (ground pad)';
```

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

first_successful_landing
2015-12-22

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 4000 and 6000;
```

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

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

Description

- We used the **WHERE** clause to filter for boosters which have successfully landed on drone ship and applied the **AND** condition to determine successful landing with payload mass greater than 4000 but less than 6000

Total Number of Successful and Failure Mission Outcomes

```
%sql select mission_outcome, count(*) as total_number from SPACEXDATASET group by mission_outcome;
```

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

Mission_Outcome	total_number
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

Description

- We used count(*) to retrieve total number of successful and failure mission outcomes

Boosters Carried Maximum Payload

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

```
 * sqlite:///my_data1.db
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
F9 B5 B1058.3
F9 B5 B1051.6
F9 B5 B1060.3
F9 B5 B1049.7

Description

- We determined the booster that have carried the maximum payload using a subquery in the **WHERE** clause and the **MAX()** function.

2015 Launch Records

Note: SQLLite does not support monthnames. So you need to use substr(Date, 4, 2) as month to get the months and substr(Date,7,4)='2015' for year.

```
%%sql select date, booster_version, launch_site, landing_outcome from SPACEXDATASET
where landing_outcome = 'Failure (drone ship)' and strftime('%Y', date) = '2015';
```

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

Date	Booster_Version	Launch_Site	Landing_Outcome
2015-10-01	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
2015-04-14	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

Description

- We used a combinations of the **WHERE** clause, **LIKE**, **AND**, and **BETWEEN** conditions to filter for failed landing outcomes in drone ship, their booster versions, and launch site names for year 2015

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

Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017

```
%%sql select landing_outcome, count(*) as count_outcomes from SPACEXDATASET
      where date between '2010-06-04' and '2017-03-20'
      group by landing_outcome
      order by count_outcomes desc;
```

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

Landing_Outcome	count_outcomes
No attempt	10
Success (ground pad)	5
Success (drone ship)	5
Failure (drone ship)	5
Controlled (ocean)	3
Uncontrolled (ocean)	2
Precluded (drone ship)	1
Failure (parachute)	1

Description:

We selected Landing outcomes and the **COUNT** of landing outcomes from the data and used the **WHERE** clause to filter for landing outcomes **BETWEEN** 2010-06-04 to 2010-03-20.

We applied the **GROUP BY** clause to group the landing outcomes and the **ORDER BY** clause to order the grouped landing outcome in descending order.

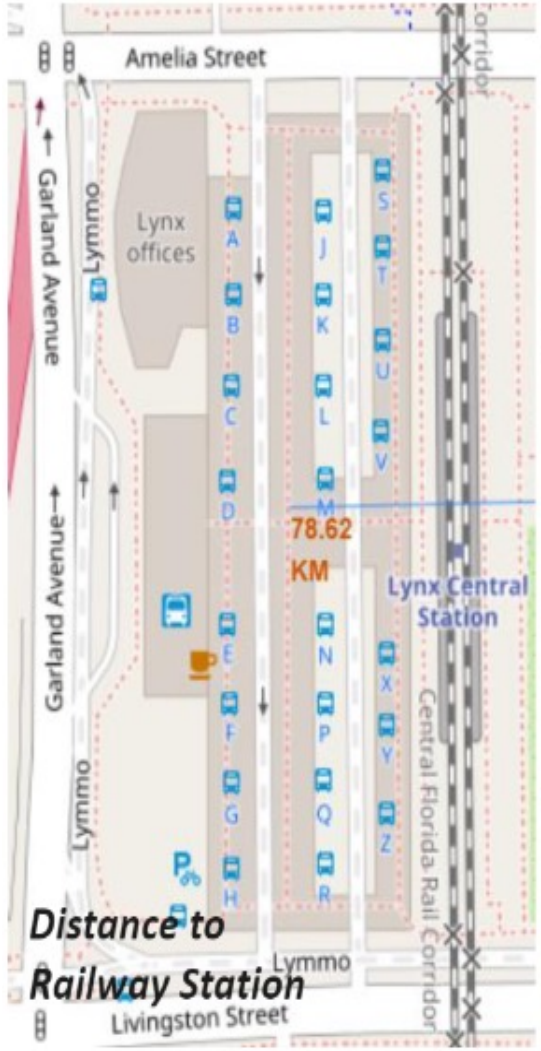

Section 3
Launch Sites
Proximities Analysis

All launch sites global map markers

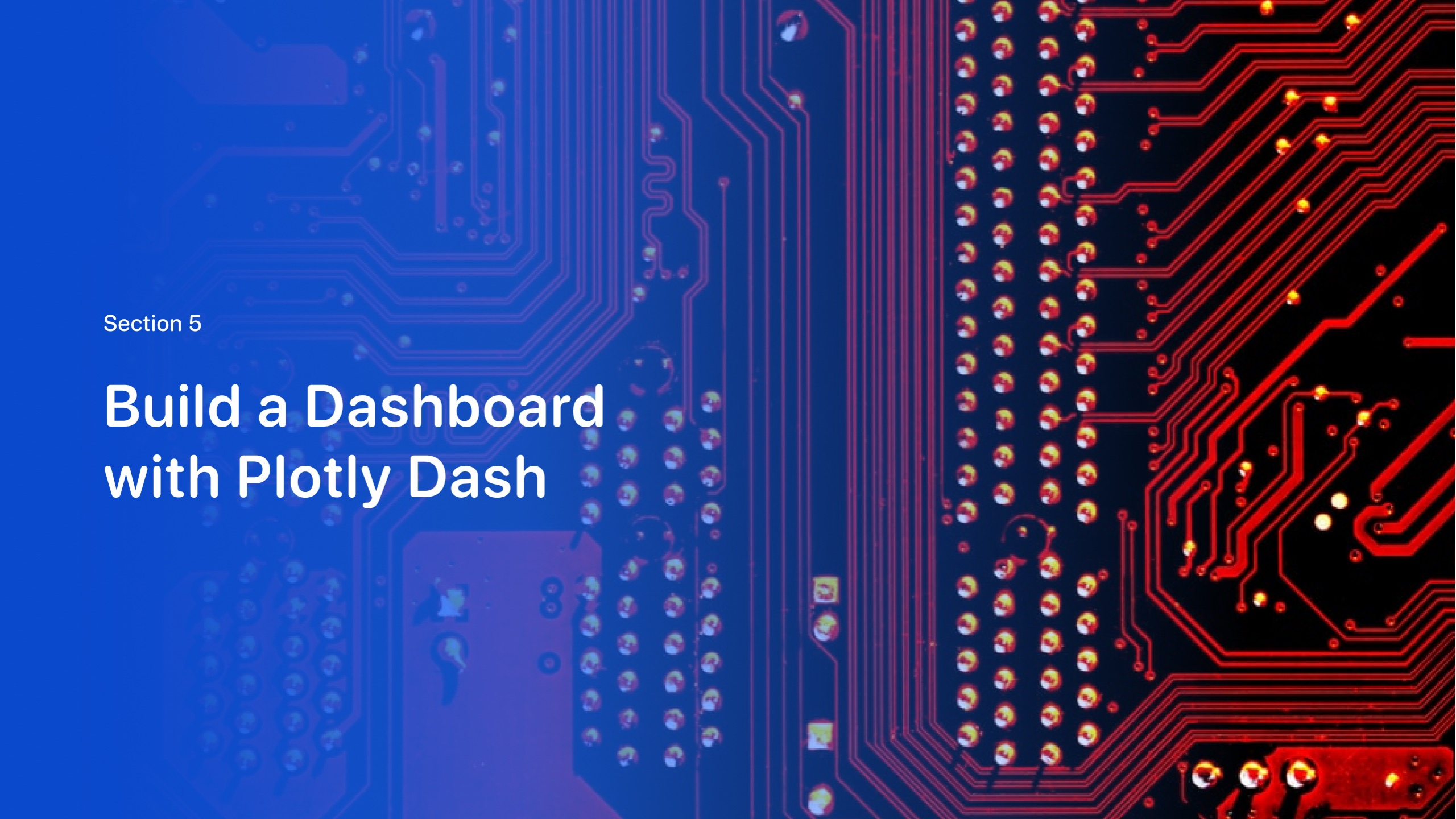
We can see that the SpaceX launch sites are in the United States of America coasts. Florida and California

Markers showing launch sites with color labels

Launch Site distance to landmarks



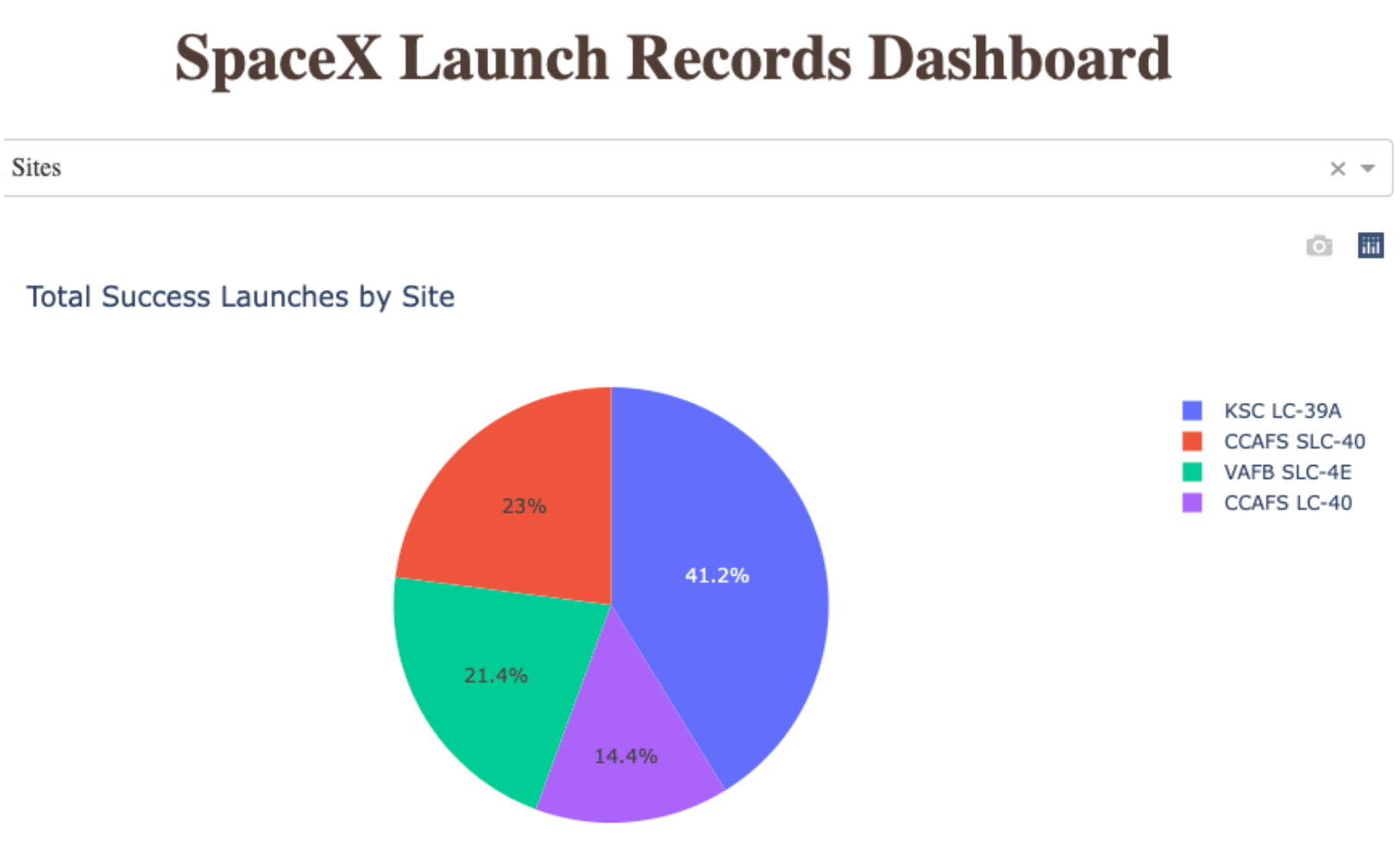
- Are launch sites in close proximity to railways? No
- Are launch sites in close proximity to highways? No
- Are launch sites in close proximity to coastline? Yes
- Do launch sites keep certain distance away from cities? Yes



Section 5

Build a Dashboard with Plotly Dash

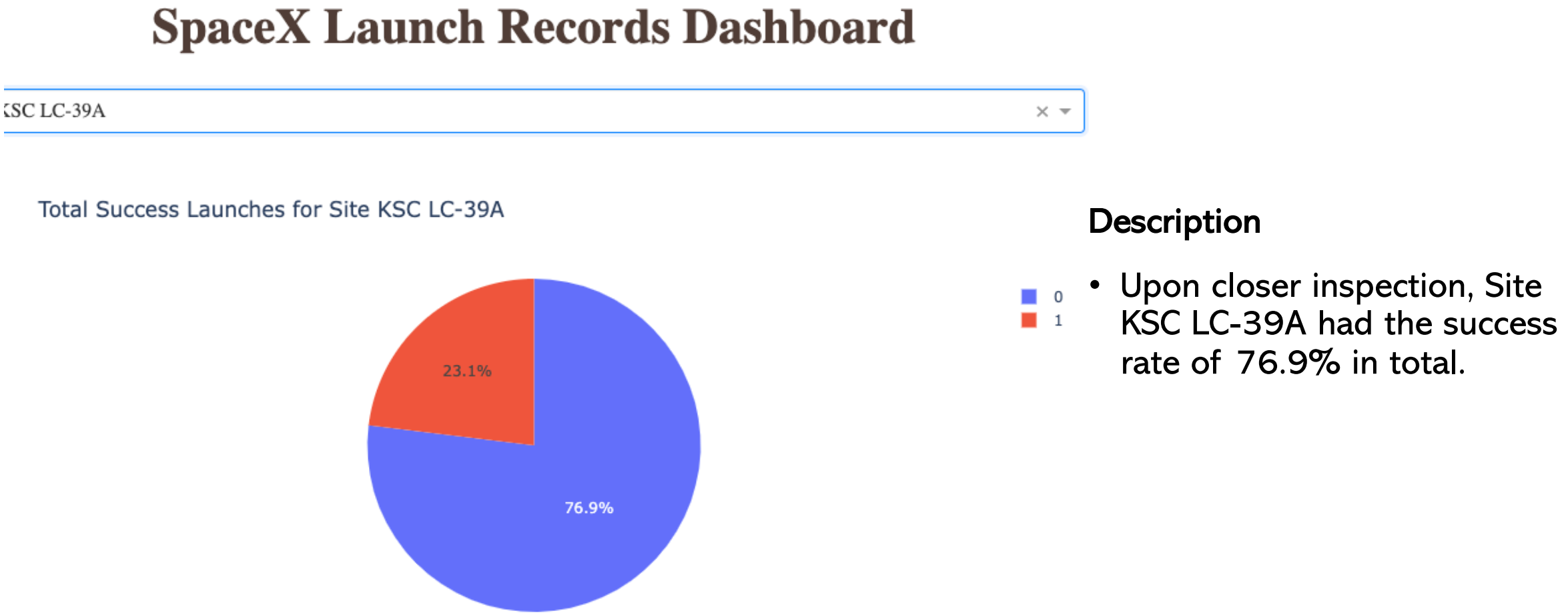
Pie chart showing the success percentage achieved by each launch site



Description

- KSC LC-30A site had the most successful launch rate of all sites.

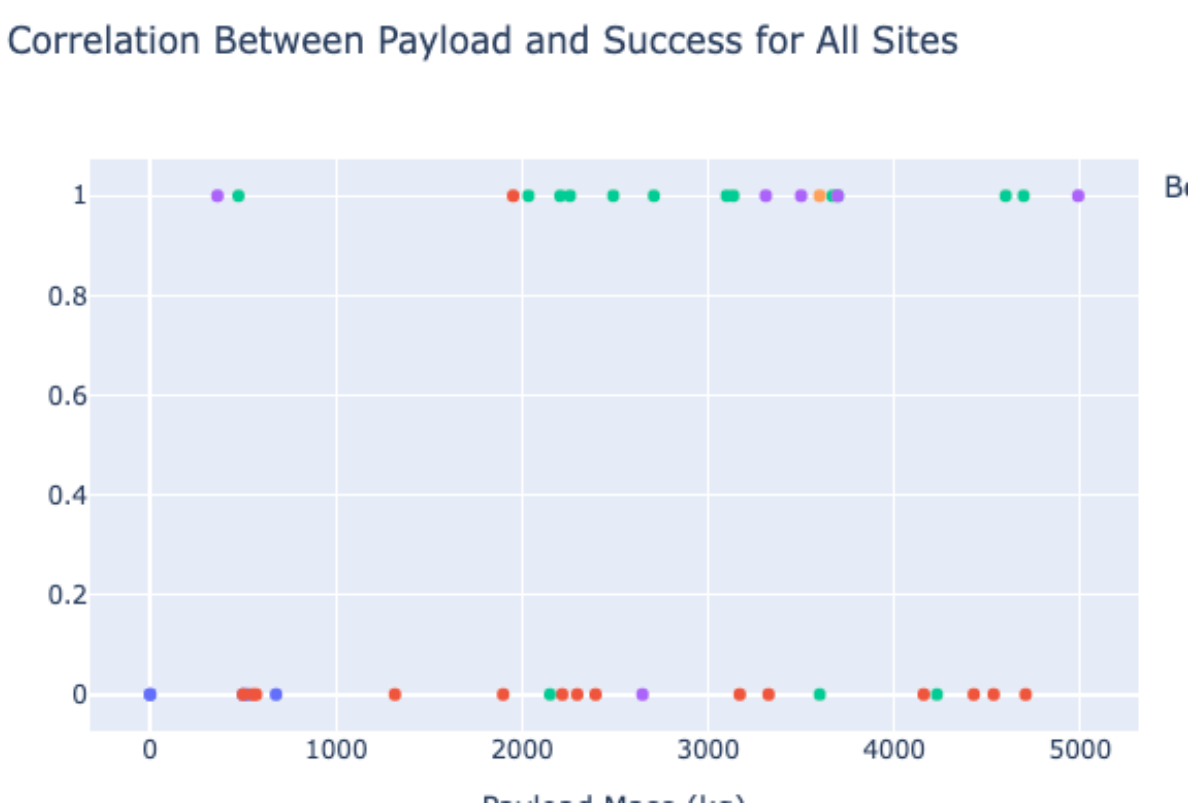
Pie chart showing the Launch site with the highest launch success ratio

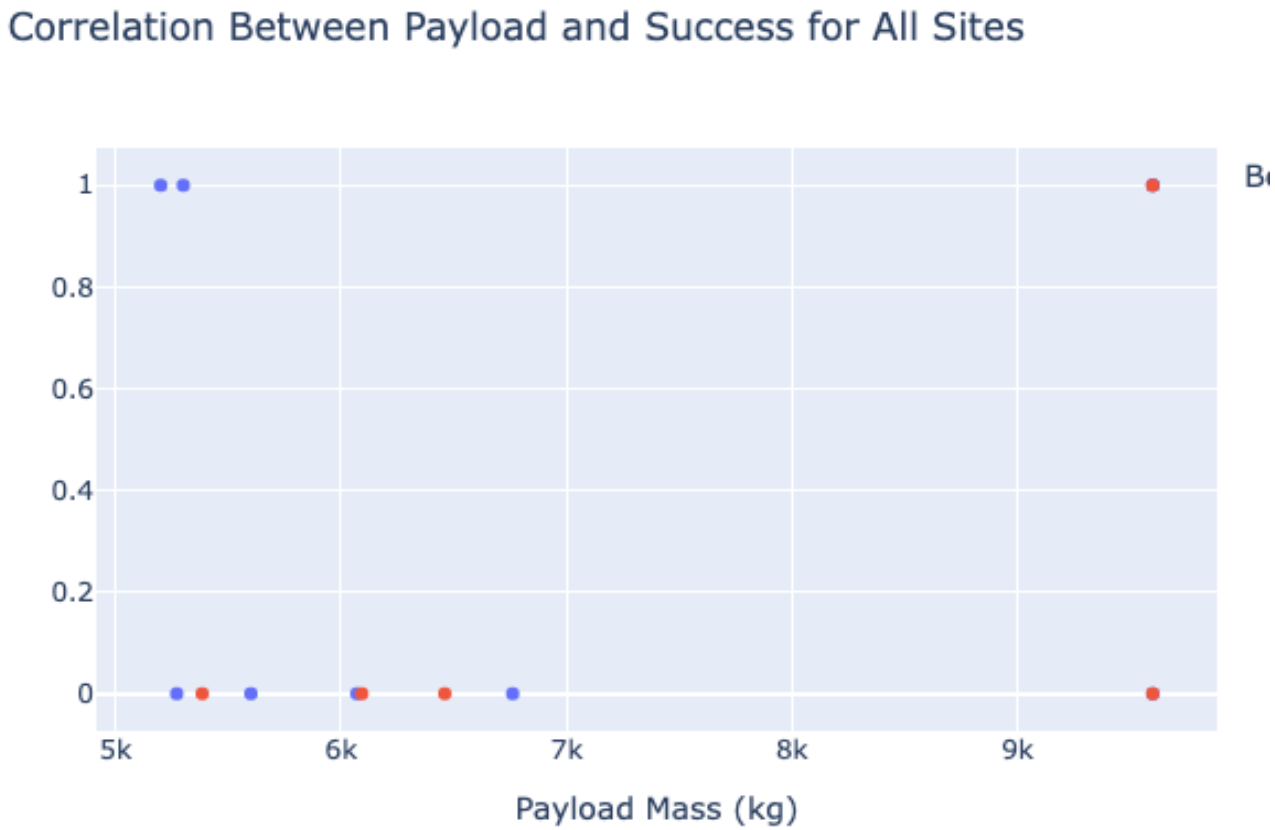


Description

- Upon closer inspection, Site KSC LC-39A had the success rate of 76.9% in total.

Scatter plot of Payload vs Launch Outcome for all sites, with different payload selected in the range slider





Description:

- The side-by-side comparison shows us that success rates for low weighted payloads is higher than the heavy weighted payloads..

Section 6

Predictive Analysis (Classification)

Classification Accuracy

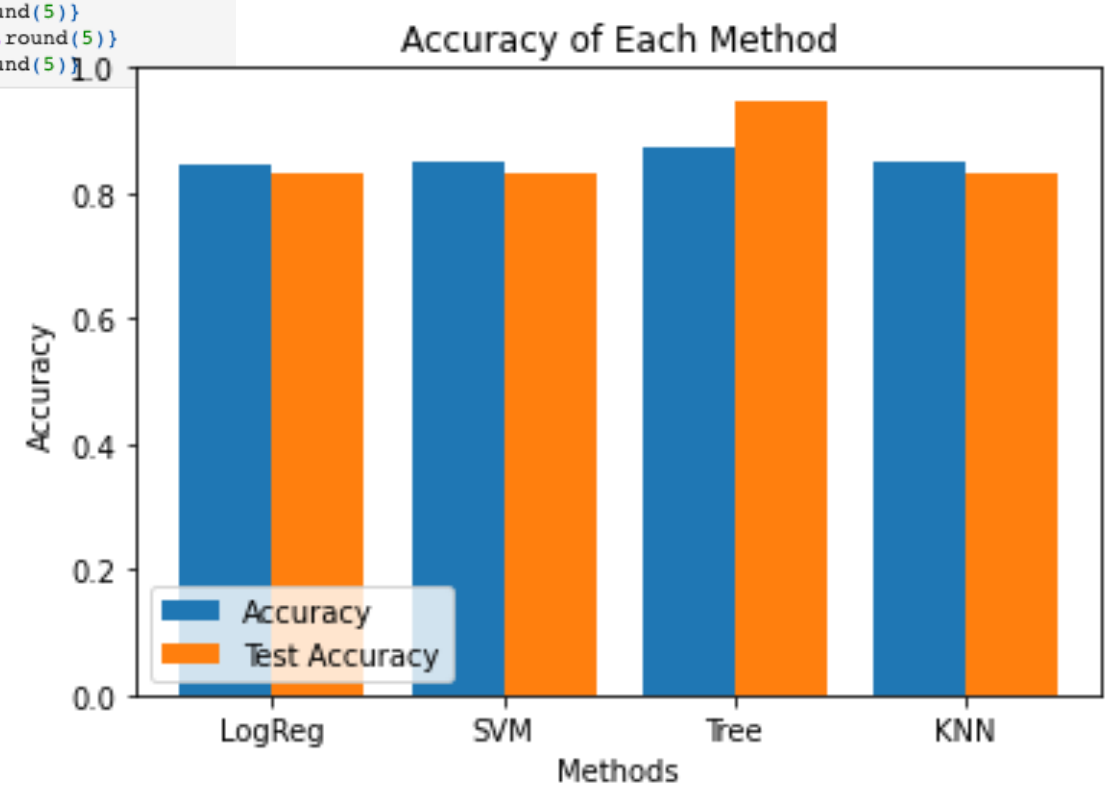
Find the method performs best:

```
print("Model\t\tAccuracy\tTestAccuracy")#,logreg_cv.best_score_)
print("LogReg\t\t{}\t\t{}".format((logreg_cv.best_score_).round(5), logreg_cv.score(X_test, Y_test).round(5)))
print("SVM\t\t{}\t\t{}".format((svm_cv.best_score_).round(5), svm_cv.score(X_test, Y_test).round(5)))
print("Tree\t\t{}\t\t{}".format((tree_cv.best_score_).round(5), tree_cv.score(X_test, Y_test).round(5)))
print("KNN\t\t{}\t\t{}".format((knn_cv.best_score_).round(5), knn_cv.score(X_test, Y_test).round(5)))

comparison = {}

comparison['LogReg'] = {'Accuracy': logreg_cv.best_score_.round(5), 'TestAccuracy': logreg_cv.score(X_test, Y_test).round(5)}
comparison['SVM'] = {'Accuracy': svm_cv.best_score_.round(5), 'TestAccuracy': svm_cv.score(X_test, Y_test).round(5)}
comparison['Tree'] = {'Accuracy': tree_cv.best_score_.round(5), 'TestAccuracy': tree_cv.score(X_test, Y_test).round(5)}
comparison['KNN'] = {'Accuracy': knn_cv.best_score_.round(5), 'TestAccuracy': knn_cv.score(X_test, Y_test).round(5)}
```

Model	Accuracy	TestAccuracy
LogReg	0.84643	0.83333
SVM	0.84821	0.83333
Tree	0.875	0.94444
KNN	0.84821	0.83333

Conclusion:

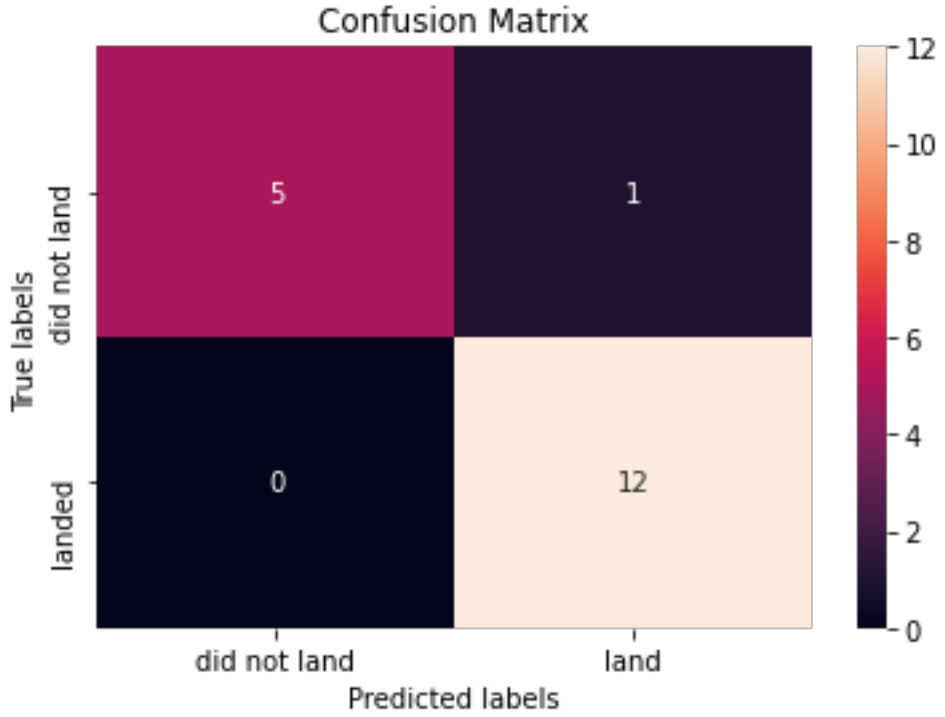
- The decision tree classifier is the model with the highest classification accuracy and the highest test accuracy.



Conclusions

Conclusion:

- The decision tree classifier's confusion matrix indicates its ability to differentiate between various classes. However, a significant issue arises in the form of false positives, wherein the classifier incorrectly identifies unsuccessful landings as successful ones.



Confusion Matrix

The bigger the flight amount at a launch site, the greater the success rate was at launch sites.

Launch success rate has been steadily increasing with time and it looks like they would reach the required target eventually.

Increasing payload mass seems to have negative impact on success launch rate.

Orbits ES-L1, GEO, HEO, SSO, VLEO had the most success rate.

Site KSC LC-39A had the most successful launches of any sites.

Decision tree classifier is the best machine learning algorithm for this task.

Thank you!

Presented by Jenna H C