IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

<Name>
<Date>

Outline

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

Executive Summary

- Summary of methodologies

  Data Collection via SpaceX REST API;

  Data wrangling with Pandas

  Exploratory Data Analysis (EDA) with SQL

  Data visualization with Matplotlib, Seaborn, Folium, Dash

  Machine Learning, KNN, Logistic Regression, Decision Tree, SVM

- Summary of all results

  Investigative Exploratory Data Analysis and Data Visualizations

  Interactive maps and dashboard

  Best model accuracy

Introduction

- ## Background and Context

  The objective is to evaluate the viability of the new company Space Y to compete with Space X.

- ## Problems

  What are the characteristics of a successful landing

  What are the relationships of the various variables that go into a mission

  What will Space Y need in order to compete with Space X

Section 1

Methodology

Methodology

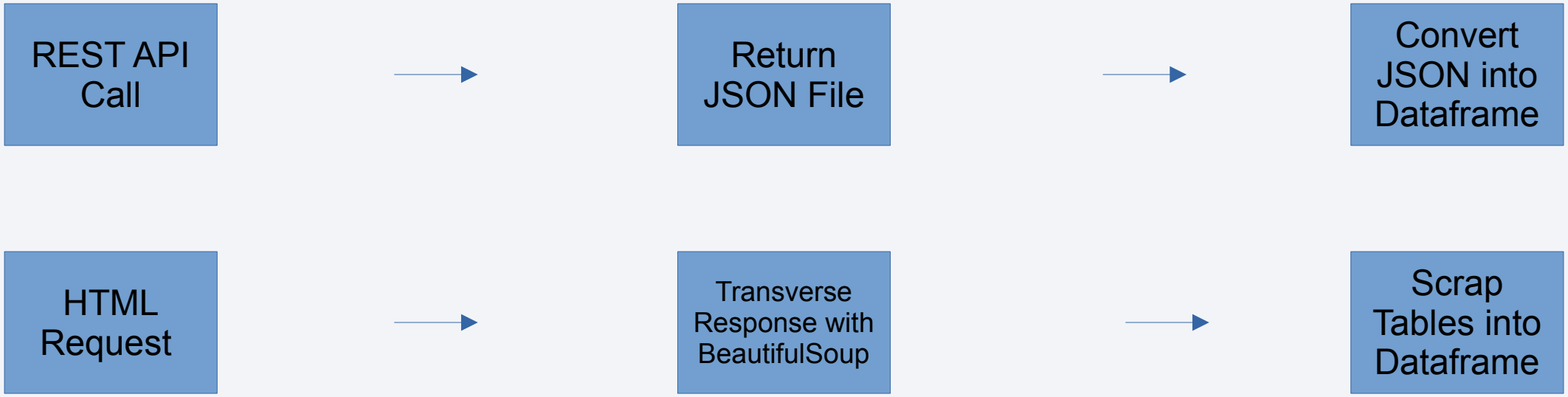
Executive Summary

- Data collection methodology:
  - Space X Rest API and scraping Falcon 9 wiki
- Perform data wrangling
  - Replacing null values, normalizing values, one hot encoding for classification
- 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 was requesting from Space X's REST API

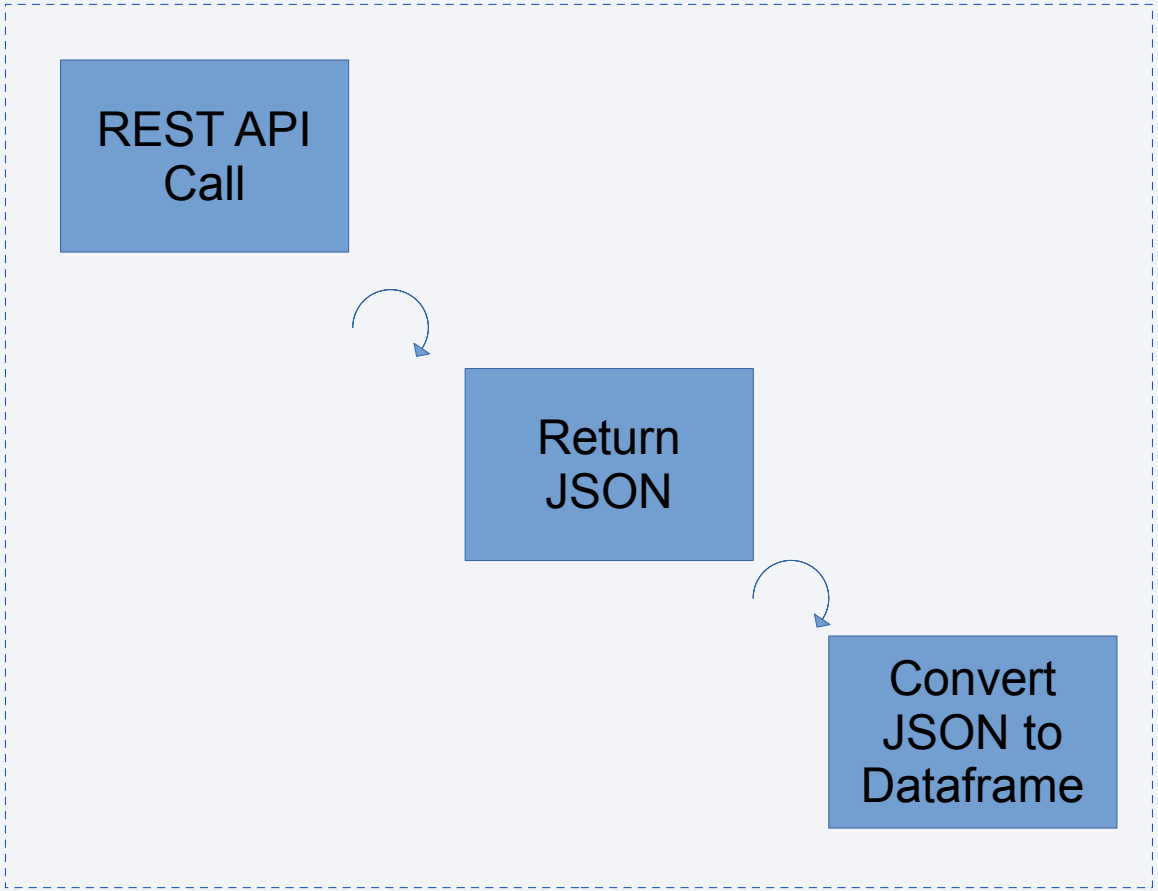
Data was scraped from https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches



Data Collection – SpaceX API

- Space X REST API URL: api.spacexdata.com/v4/
- Transformed JSON into Dataframe using Pandas Python Library

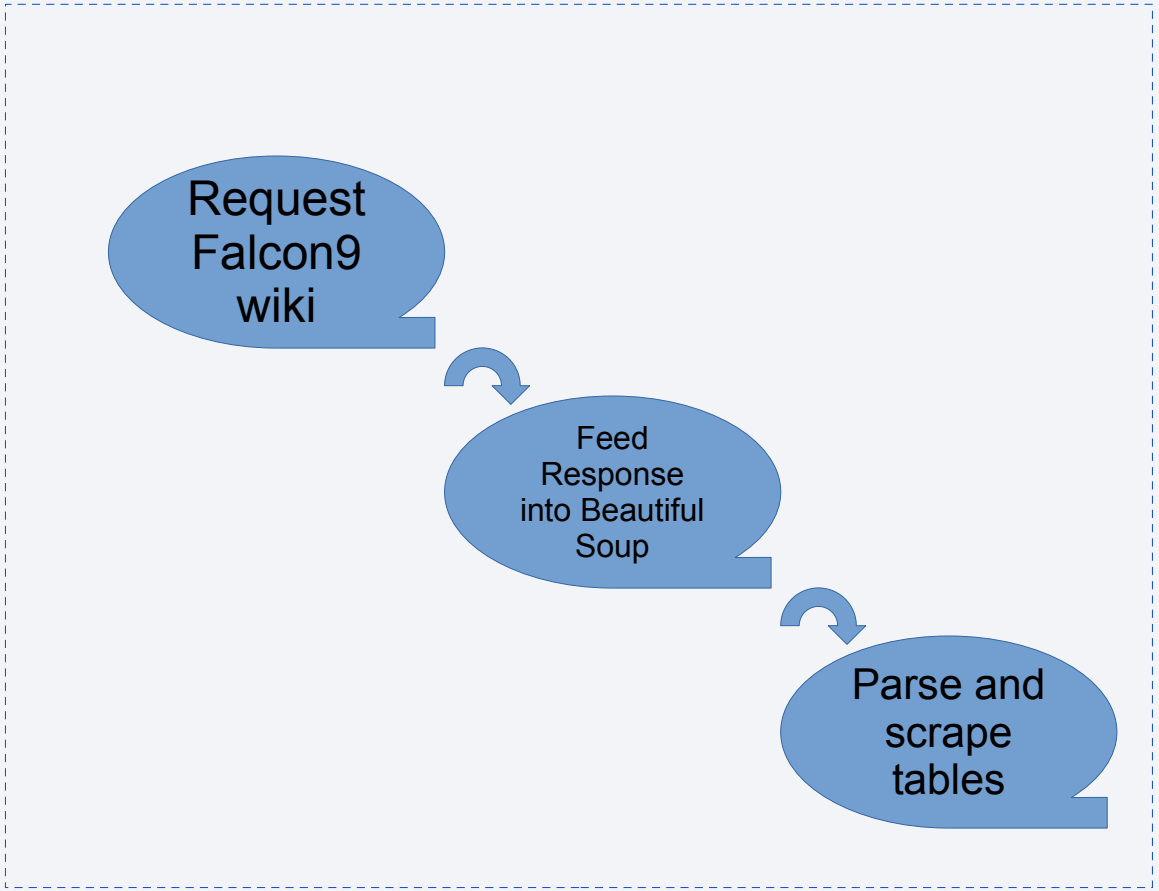
Link



Data Collection - Scraping

- Send Request to wiki
- Insert Response into BeautifulSoup
- Find all tables
- Convert Data into Dataframe

Link



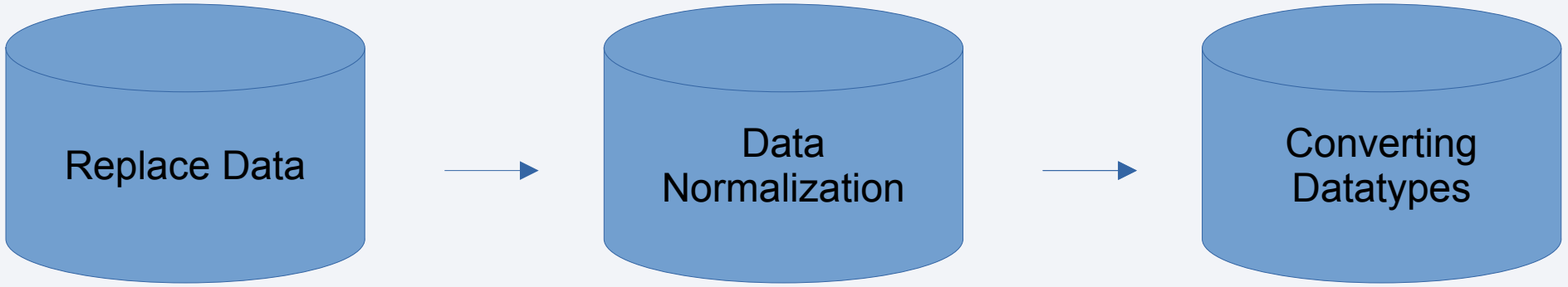
Data Wrangling

Missing values were replaced appropriately for analysis

Link

Data was Normalized for upcoming analysis

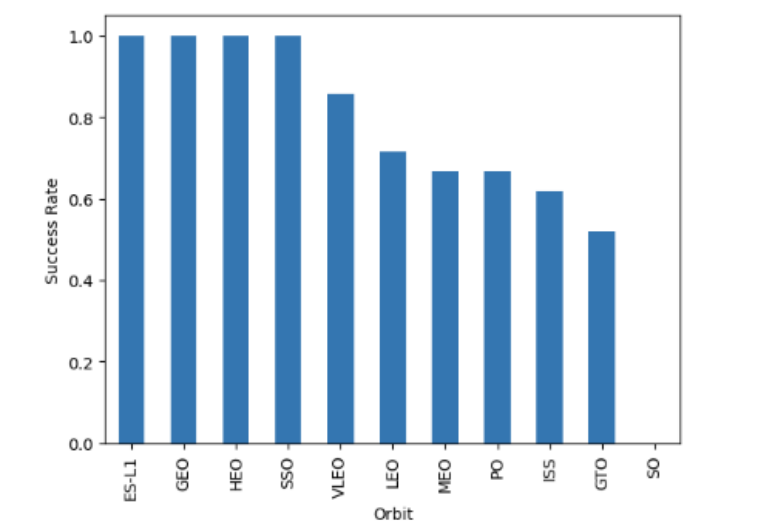
Categorical Attributes were converted to numeric for Classification

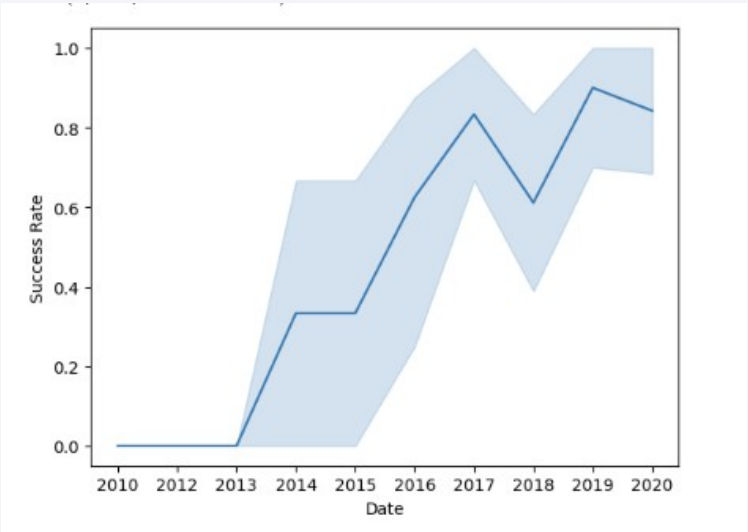


EDA with Data Visualization

In order to discover relationships scatterplots and barplots and lineplots were created

Link





EDA with SQL

- Retrieved unique Launch Site Names
- Retrieved 5 records where launch sites begin with the name 'CCA'
- Displayed the total payload carried by boosters for NASA CRS launches
- Displayed average payload carried by booster versions F9 v1.1
- Retrieved the date for the first successful landing
- Retrieved the names of the boosters for successes that had payload between 4000 and 6000
- Summarized successful and failed missions
- Retrieved the names of the booster version which had carried the maximum payload
- Retrieved the 2015 records for failed landing outcomes
- Sorted the number of successful landings between 04-06-2010 and 20-03-2017

Link

Build an Interactive Map with Folium

- Markers display launch sites
- Circles highlight areas around sites
- Marker Clusters depict groups of important sites
- Lines visualize distances between points.

Link

Build a Dashboard with Plotly Dash

Dashboard includes pie chart and scatter plot along with dropd down menu and range slider

Pie chart depicts the success vs failure for given launch site

Scatter chart indicates relationship between two variables

Drop down allows for visualizing specific metrics

Range slider allows for visualizing specific metrics

Link

Predictive Analysis (Classification)

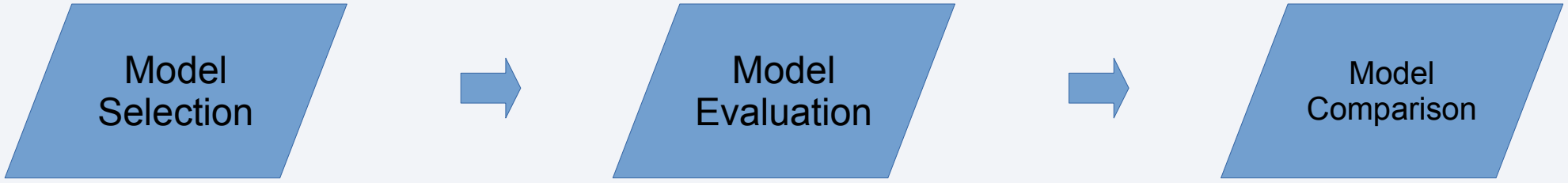
Split data into training and testing set

Select Model

Use Grid Search Cross Validation for parameters

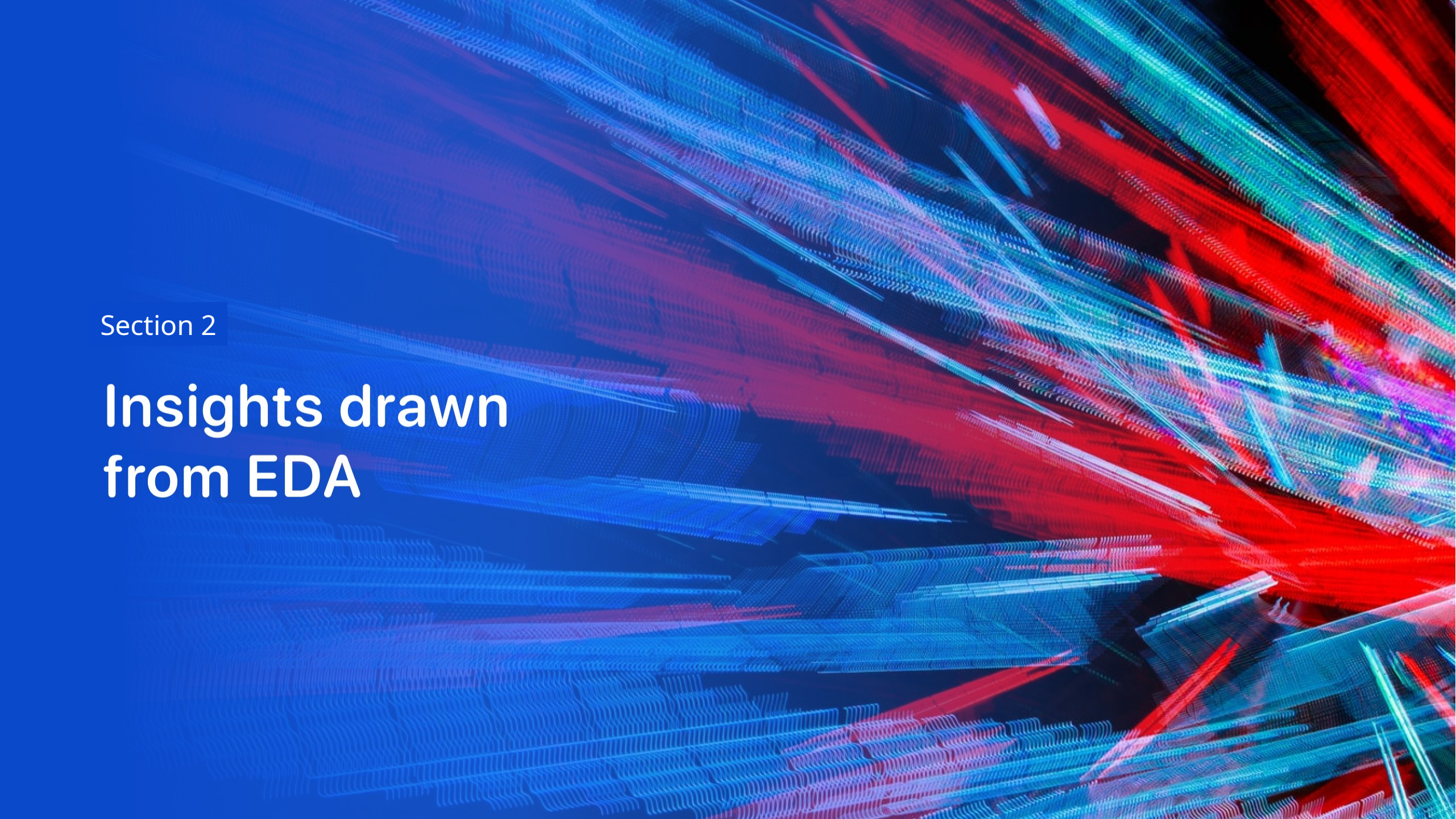
Compare Model Accuracy

URL of your completed predictive analysis lab, as an external reference and peer-review purpose



Results

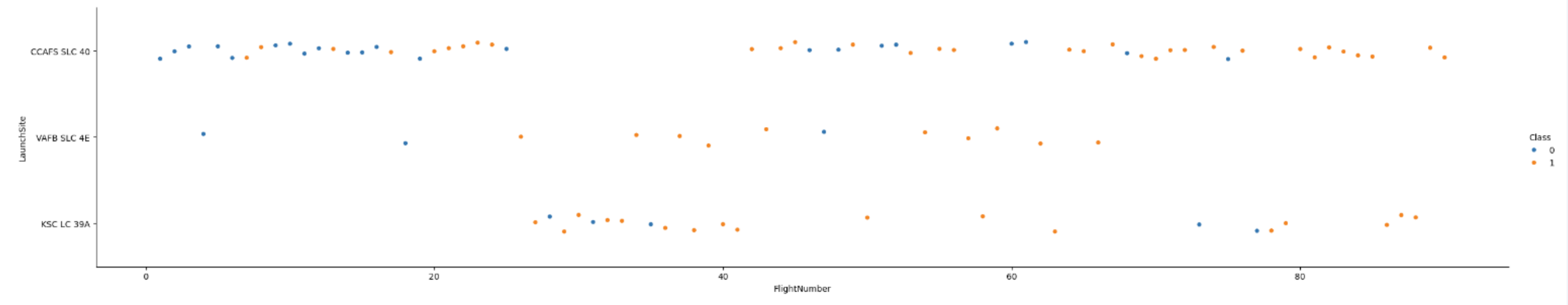
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results



Section 2

Insights drawn from EDA

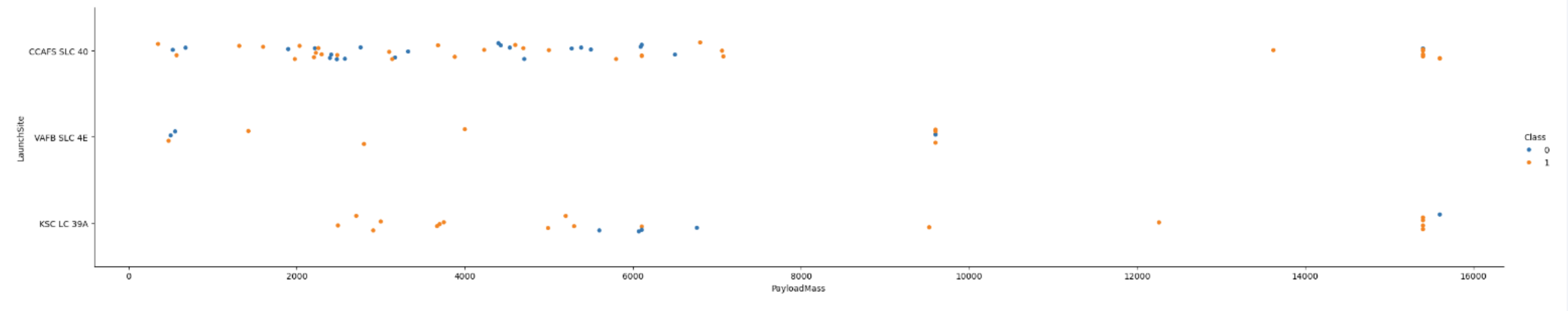
Flight Number vs. Launch Site



Scatter point depicts relationship between Flight number and Launch Site

CCAF5 SLC 40 has many flights

Payload vs. Launch Site



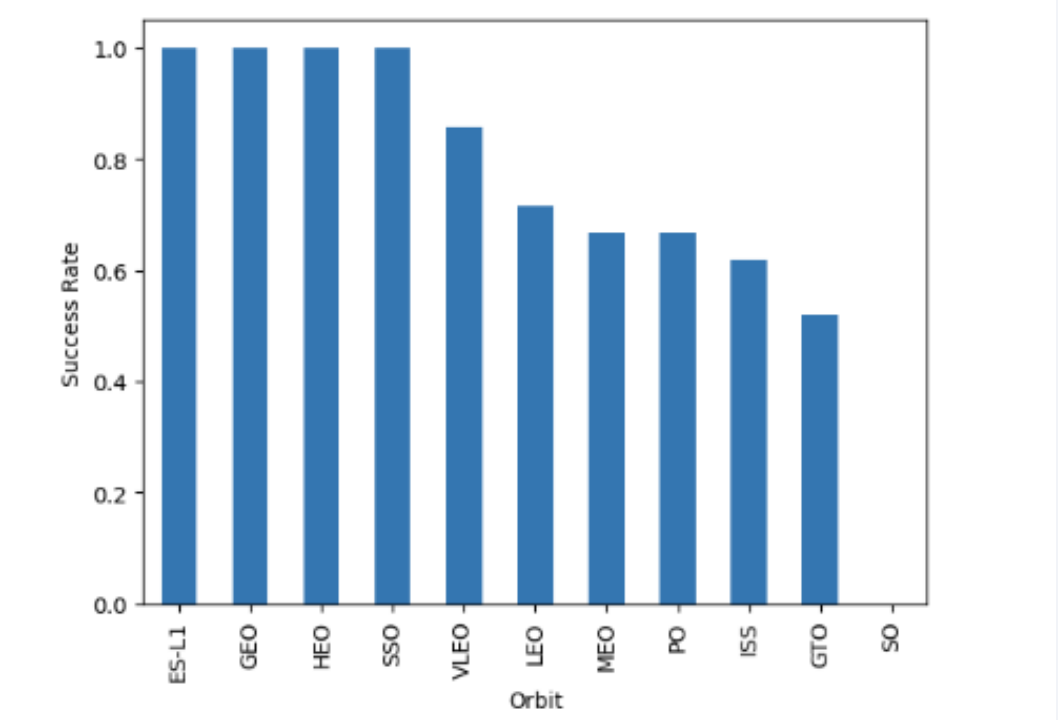
Some launch sites may have better odds depending on Payload

VA4B did not see any payload greater than 10000

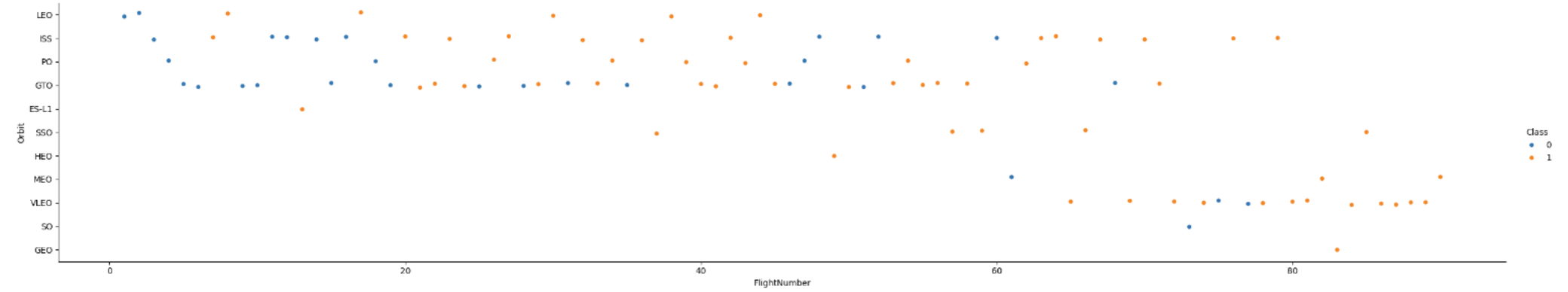
Success Rate vs. Orbit Type

Highest success rates:

- ES-L1
- GEO,
- HEO
- SSO



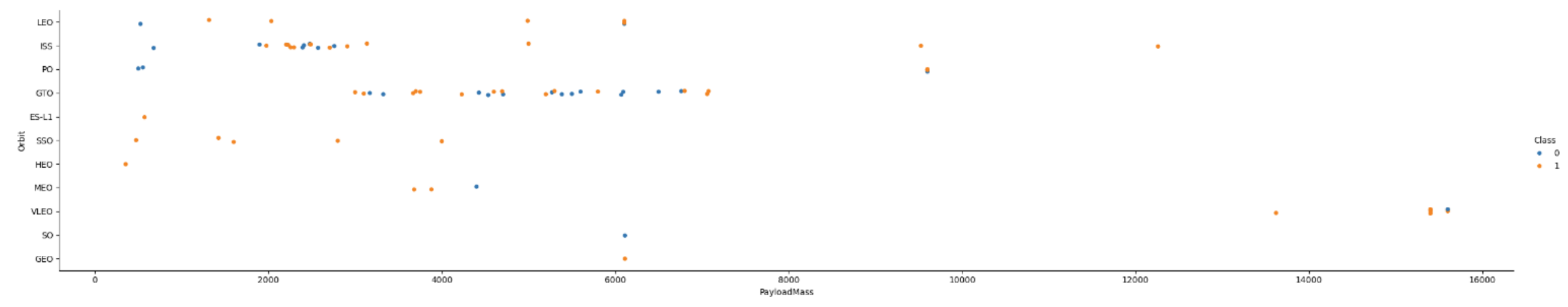
Flight Number vs. Orbit Type



Scatter point depicts relationship between Flight number and Orbit type

Some orbits see more flights compared to others

Payload vs. Orbit Type

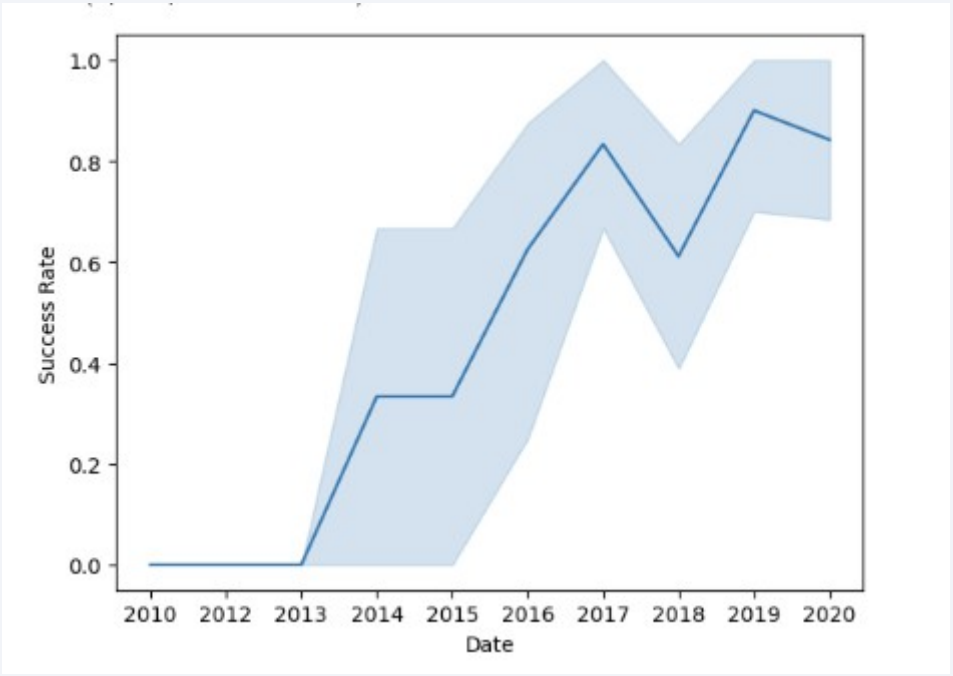


Scatter point depicts relationship between Payload and Orbit type

Most payloads are not near maximum

Launch Success Yearly Trend

- Success rates picks up in 2013
- First three years had saw no success but perhaps laid foundation for future missions



All Launch Site Names

```
%sql SELECT DISTINCT Launch_Site FROM SPACEXTABLE
```

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

Unique Launch Sites

Launch Site Names Begin with 'CCA'

```
%sql SELECT DISTINCT Launch_Site FROM SPACEXTABLE
```

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

Find 5 records where launch sites begin with `CCA`

Total Payload Mass

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE Customer = 'NASA (CRS)'
```

SUM(PAYLOAD_MASS__KG_)
45596

Total payload carried by boosters from NASA

Average Payload Mass by F9 v1.1

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE Booster_Version LIKE 'F9 v1.1%'
```

AVG(PAYLOAD_MASS__KG_)
2534.6666666666665

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE Booster_Version = 'F9 v1.1'
```

AVG(PAYLOAD_MASS__KG_)
2928.4

Average payload mass carried by booster version F9 v1.1 and

Average payload mass carried by booster version F9 v1.1.etc

First Successful Ground Landing Date

```
%sql SELECT MIN(Date) FROM SPACEXTABLE WHERE Landing_Outcome = 'Success (ground pad)'
```

MIN(Date)
2015-12-22

Dates of the first successful landing outcome on ground pad

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%sql SELECT Booster_Version FROM SPACEXTABLE WHERE Landing_Outcome = 'Success (ground pad)' AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000
```

Booster_Version
F9 FT B1032.1
F9 B4 B1040.1
F9 B4 B1043.1

Names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000

Total Number of Successful and Failure Mission Outcomes

```
%sql SELECT Mission_Outcome, COUNT(*) FROM SPACEXTABLE GROUP BY Mission_Outcome
```

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

Total number of successful and failure mission outcomes

Boosters Carried Maximum Payload

```
%%sql
    SELECT DISTINCT Booster_Version FROM SPACEXTABLE WHERE PAYLOAD_MASS__KG_ =
        (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE)
```

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

Names of the booster which have carried the maximum payload mass

2015 Launch Records

```
%%sql SELECT Booster_Version, Launch_Site, substr(Date, 6,2) FROM SPACEXTABLE
    WHERE substr(Date,0,5) = '2015' AND Landing_Outcome = 'Failure (drone ship)'
```

Booster_Version	Launch_Site	substr(Date, 6,2)
F9 v1.1 B1012	CCAFS LC-40	01
F9 v1.1 B1015	CCAFS LC-40	04

Failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

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

```
%%sql SELECT Landing_Outcome, COUNT(Landing_Outcome) FROM SPACEXTABLE
    WHERE Date BETWEEN '2010-06-04' AND '2017-03-20'
    GROUP BY Landing_Outcome ORDER BY COUNT(Landing_Outcome) DESC
```

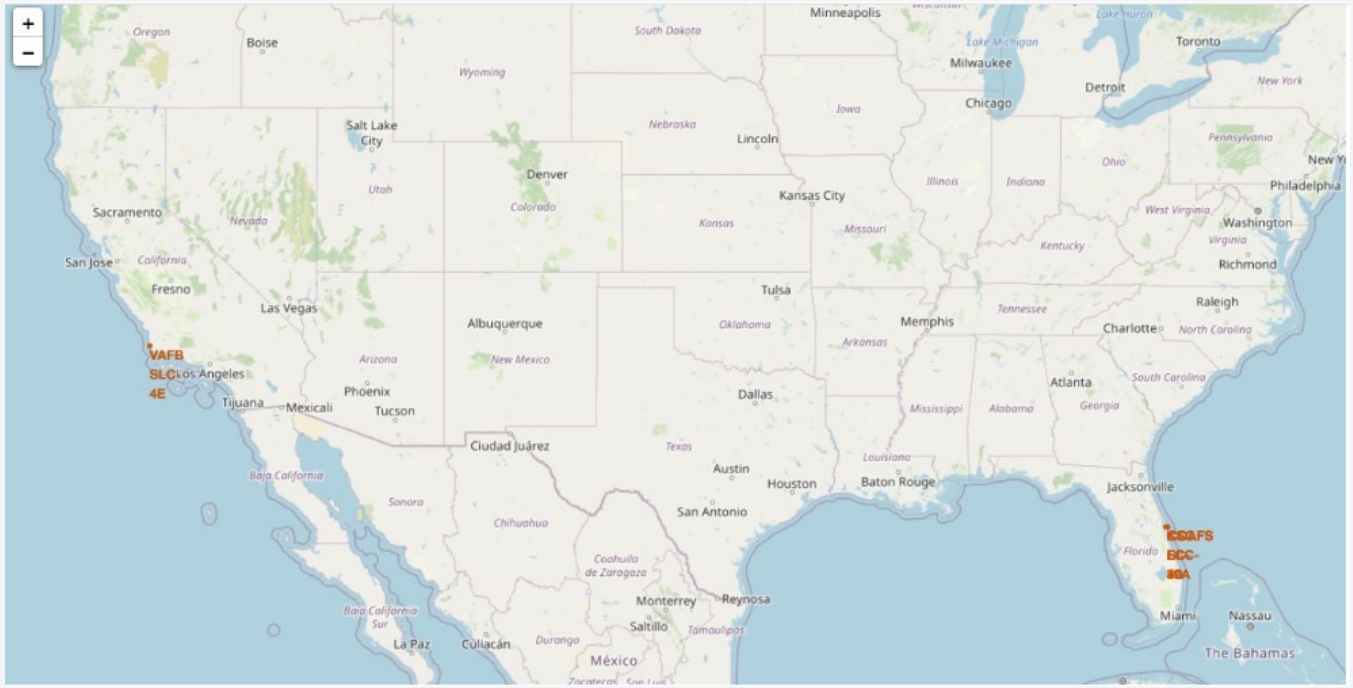
Landing_Outcome	COUNT(Landing_Outcome)
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

Ranking the 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

Section 3

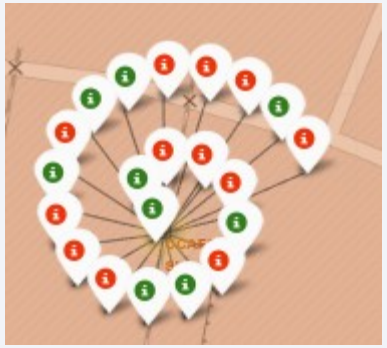
Launch Sites Proximities Analysis

Launch Sites



Launch sites locations in Florida and California

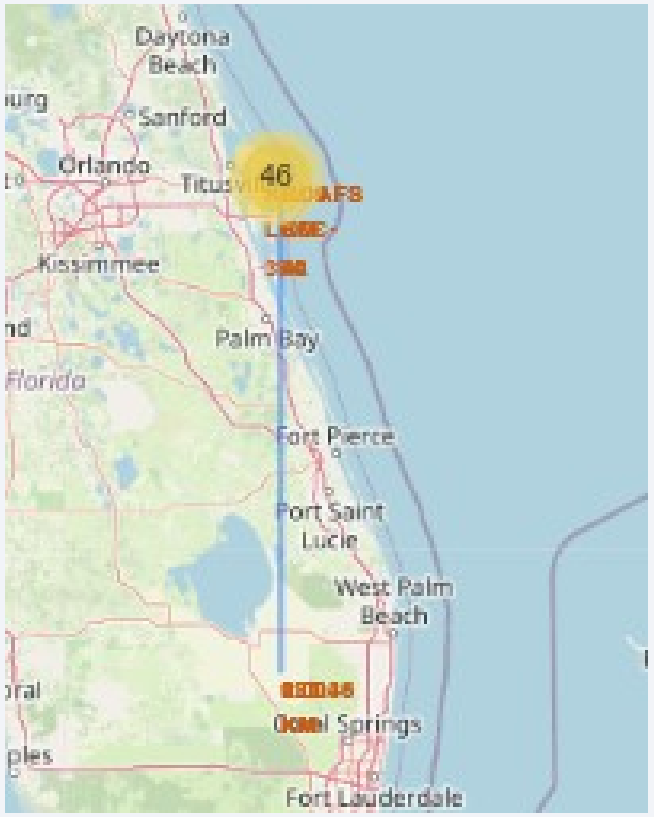
Launch Outcomes



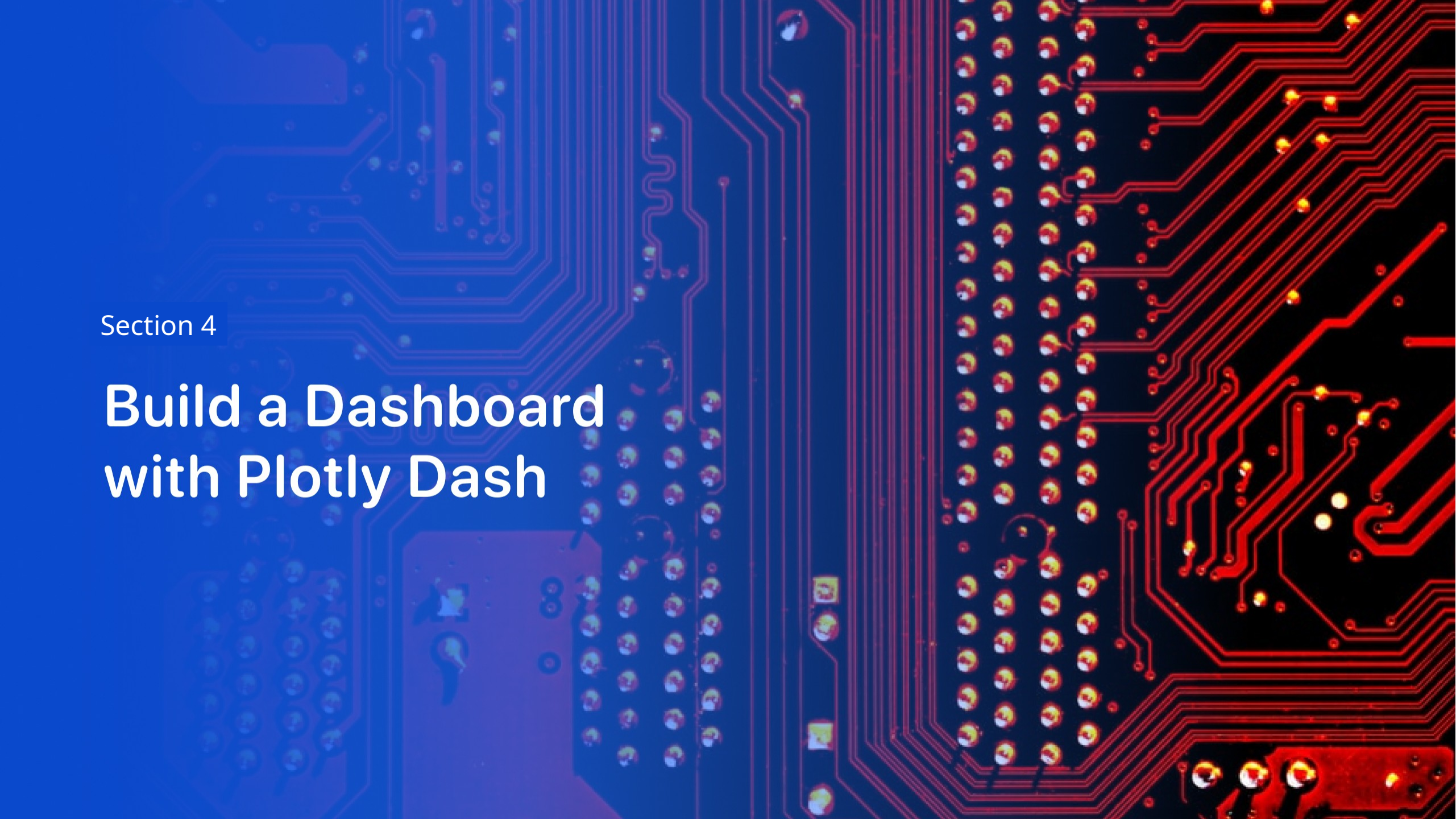
Green depicts success

Red shows failure

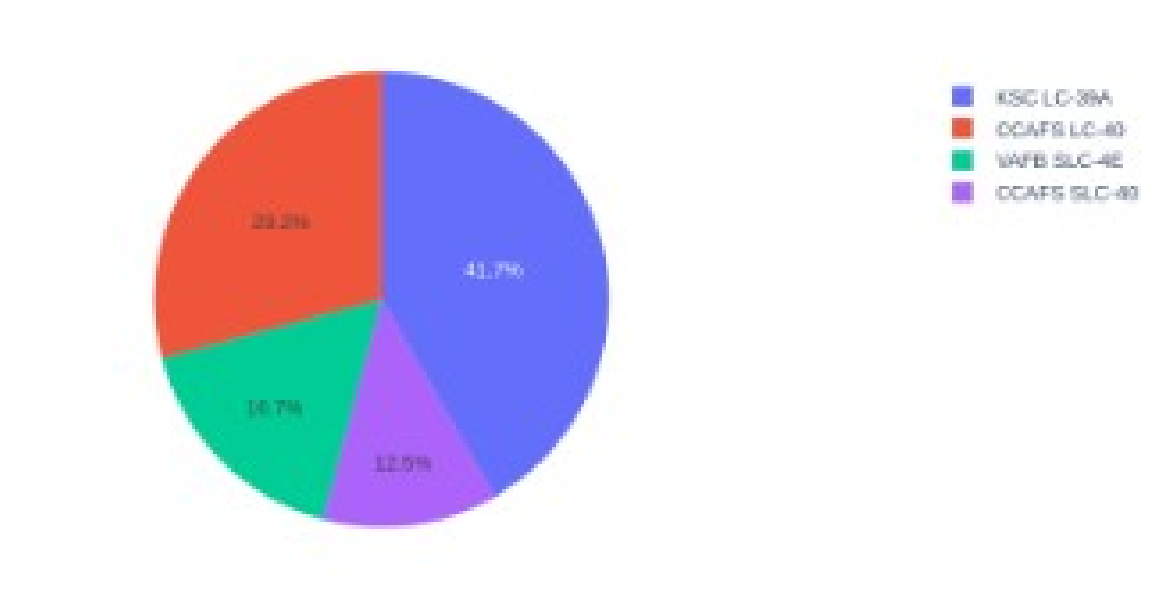
Launch Site Proximity



Launch site proximity, coastal and far from urban centres

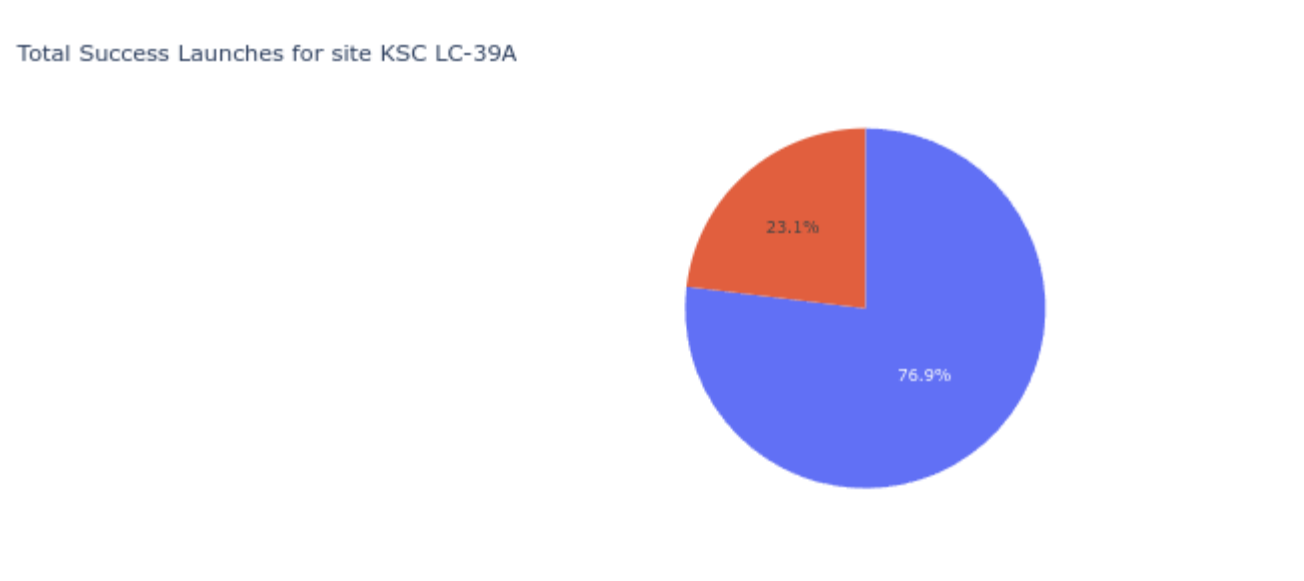

Section 4
Build a Dashboard with Plotly Dash

Total Successful Launches By Site



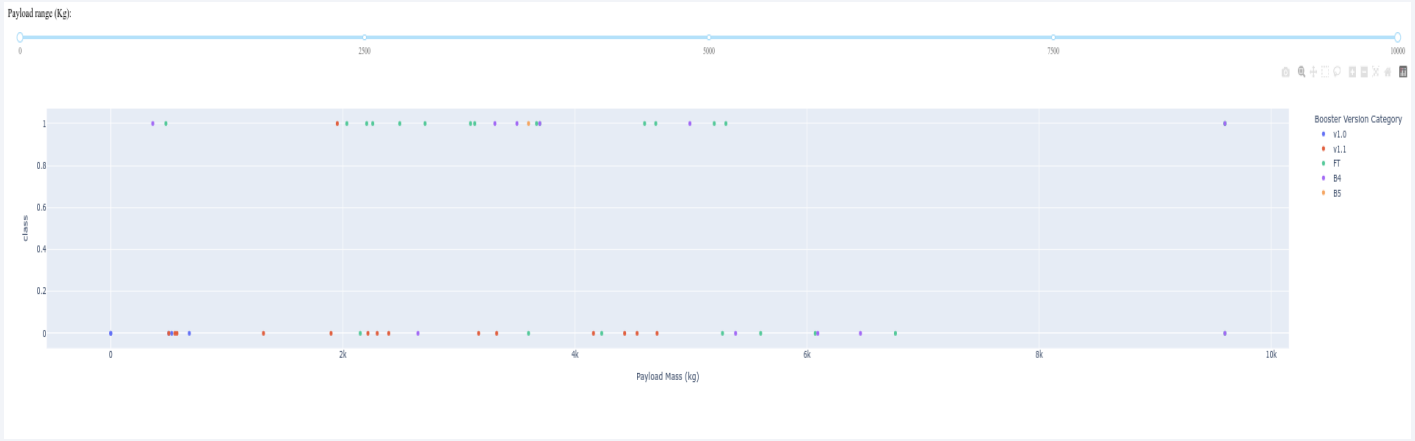
Launch Site may have relationship with success rate

Best Launch Success for KSC LC-39A



76.9% success rate was the highest

Payload vs Outcome



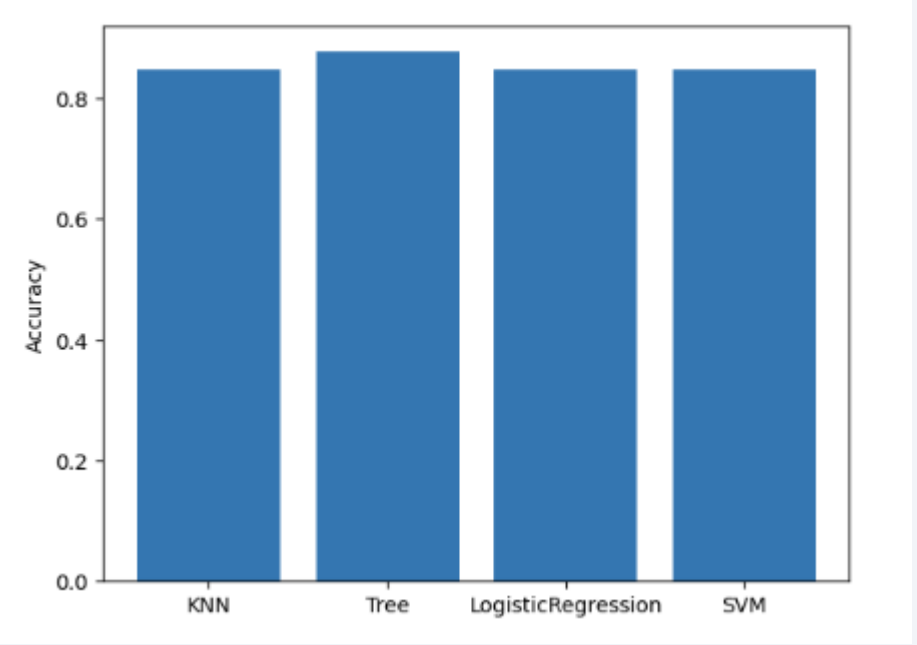
More lower payload outcomes

Section 5

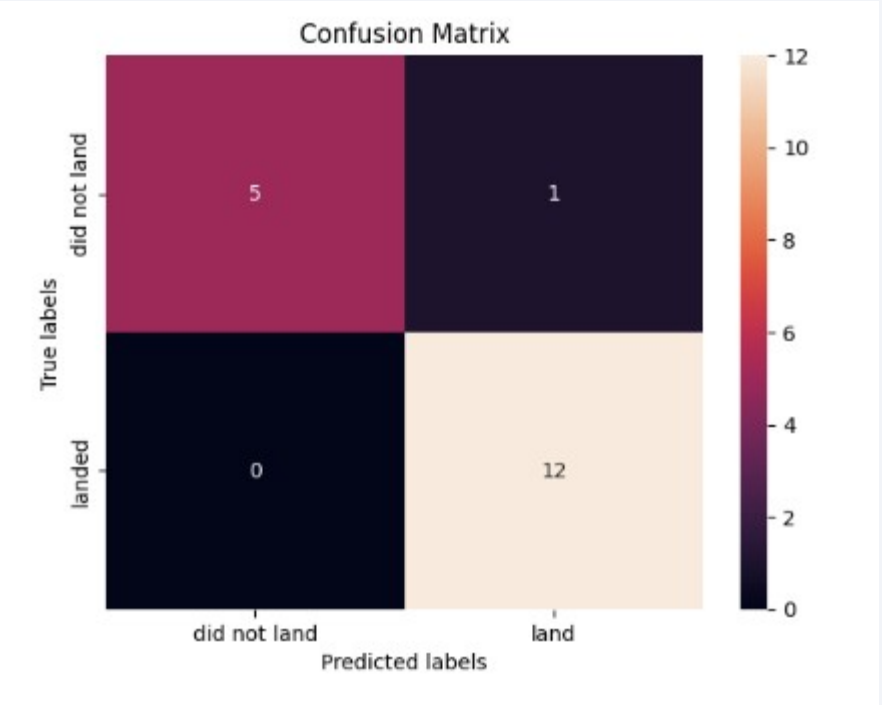
Predictive Analysis (Classification)

Classification Accuracy

- KKN, Decision Tree, Logistic Regression and SVM were used
- Decision Tree had the highest classification accuracy



Confusion Matrix



Confusion tree only shows 1 misclassified instance

Conclusions

Payload, launch site, orbit are all considerations for success

KSC LC-39A had many launches

Lower payloads are more tried

Decision Tree Classifier was best model

Appendix

See GitHub for links

TASK 8: Cast all numeric columns to `float64`

Now that our `features_one_hot` dataframe only contains numbers, cast the entire dataframe to variable type `float64`

```
# HINT: use astype function
features_one_hot.astype('float64')
```

	FlightNumber	PayloadMass	Flights	GridFins	Reused	Legs	Block	ReusedCount	Orbit_ES-L1	Orbit_GEO	...	Serial_B1048	Se
0	1.0	6104.959412	1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	...	0.0	
1	2.0	525.000000	1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	...	0.0	
2	3.0	677.000000	1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	...	0.0	
3	4.0	500.000000	1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	...	0.0	
4	5.0	3170.000000	1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	...	0.0	
...	...	...	...	...	...	...	...	...	...	...	...	...	
85	86.0	15400.000000	2.0	1.0	1.0	1.0	5.0	2.0	0.0	0.0	...	0.0	
86	87.0	15400.000000	3.0	1.0	1.0	1.0	5.0	2.0	0.0	0.0	...	0.0	
87	88.0	15400.000000	6.0	1.0	1.0	1.0	5.0	5.0	0.0	0.0	...	0.0	
88	89.0	15400.000000	3.0	1.0	1.0	1.0	5.0	2.0	0.0	0.0	...	0.0	
89	90.0	3681.000000	1.0	1.0	0.0	1.0	5.0	0.0	0.0	0.0	...	0.0	

90 rows × 80 columns

Thank you!