IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Denys Marchenko
20.07.2022

Outline

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

Executive Summary

- Summary of methodologies:
  - Data was collected from SpaceX API and a wikipedia.com HTML table, labeled with success/failed labels, discovered with SQL queries, Folium visualization and a dashboard. Then the data was normalized, changed, fitted to SVM, KNN, LR and Decision Tree models using the method Grid Search. As a conclusion, accuracy was calculated and compared.
- Summary of all results:
  - Was defined that the models requires more data then 18 samples for dataset. However, the accuracy on a test set is equal to 83.33% for landing the 1st stage booster. Therefore, the best way is keep searching information and analyzing

Introduction

- Project background and context:
  - SpaceX 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 SpaceX 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 SpaceX for a rocket launch. Furthermore, trend for space travel is increasing.
- Data collection methodology:
  - Will the Falcon 9 first stage land successfully?
  - What factors do determine successful rocket landing
  - Relationships between variables to discover ones with the biggest impact

Section 1

Methodology

Methodology

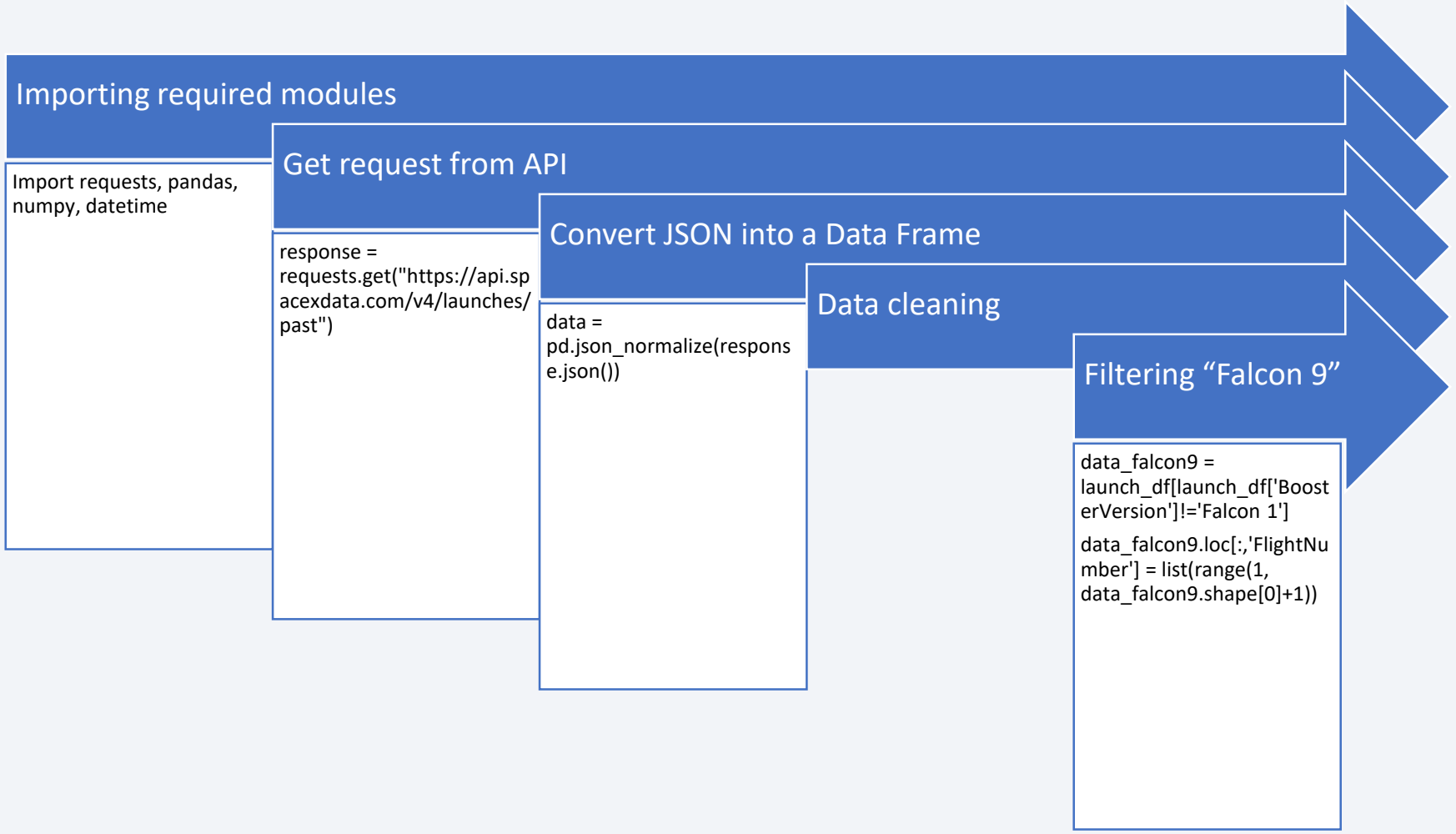
Executive Summary

- Data collection methodology:
  - Data was collected using a get request from a module “requests” to SpaceX API and from a wikipedia.com HTML table
- Perform data wrangling
  - Bad and good outcomes were classified and labeled in the data
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Scikit-learn library encomposes huge variety of classification models. 4 models were represented in the research: SVM, KNN, Decision Tree, Logistic Regression. Then models were trained hyperparameters were selected using method Grid Search. To determine the most efficient model, accuracy was calculated for each model

Data Collection

- Data collection methodology:
  - Get request to SpaceX API
  - Web scrapping from Wikipedia
- Data was obtained using “requests” module on SpaceX API. The obtained json file with raw data was converted into a Data Frame. Then the data was cleaned and missing values were replaced with mean. As the result, was obtained a Data Frame that contains 90 rows and 17 columns.
- Web scrapping allowed to retrieve historical launch records which are highly important. The data was collected using “requests” module, as mentioned before, and Beautiful Soup for retrieving data from the raw HTML code. Then was obtained the exact table as on wikipedia.com (121 rows, 10 columns) plus the 11th manual column named “Time”

Data Collection – SpaceX API



Data Collection - Scraping

Importing required modules

- Import requests, pandas, unicodedata, re, BeautifulSoup, sys

Request the Wiki page

- response = requests.get("https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922").text
- soup = BeautifulSoup(response, 'html5lib')

Extract all columns

- html_tables = soup.find_all('table')
- first_launch_table = html_tables[2]
- column_names = []
- ths = first_launch_table.find_all('th')
- for row in ths:
- header = extract_column_from_header(row)
- if header is not None and len(header) > 0:
- column_names.append(header)

Parsing HTML tables in a dictionary

Convert dictionary into a DF

- data_falcon9 = launch_df[launch_df['BoosterVersion']!='Falcon 1']
- data_falcon9.loc[:, 'FlightNumber'] = list(range(1, data_falcon9.shape[0]+1))

Data Wrangling

- The main goals of this section were: to receive data insides those are shown below, create labeled outcome column, determine the success rate which is equal to 66.67% (df["Class"].mean())

Number of launches on each site

- df['LaunchSite'].value_counts()

CCAFS SLC 40	55
KSC LC 39A	22
VAFB SLC 4E	13

Number and occurrence of each orbit

- df['Orbit'].value_counts()

GTO	27
ISS	21
VLEO	14
PO	9
LEO	7
SSO	5
MEO	3
GEO	1
ES-L1	1
HEO	1
SO	1

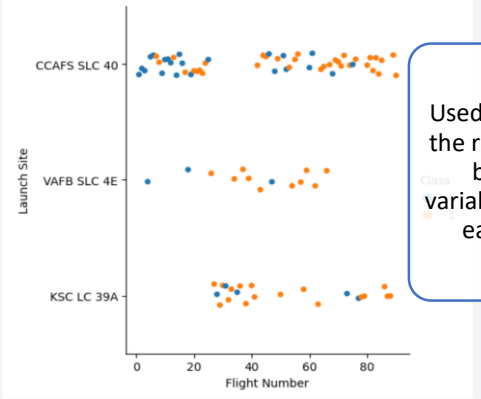
Number and occurrence of mission outcome per orbit type

- landing_outcomes = df['Outcome'].value_counts()

True ASDS	41
None None	19
True RTLS	14
False ASDS	6
True Ocean	5
None ASDS	2
False Ocean	2
False RTLS	1

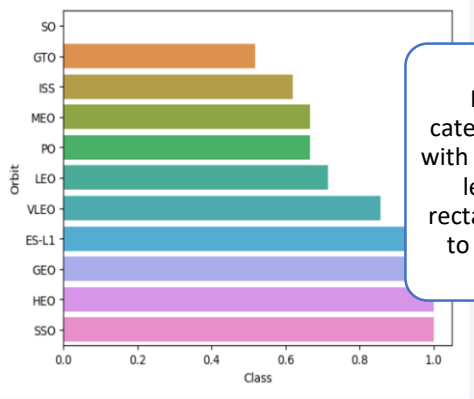
EDA with Data Visualization

Scatter plot



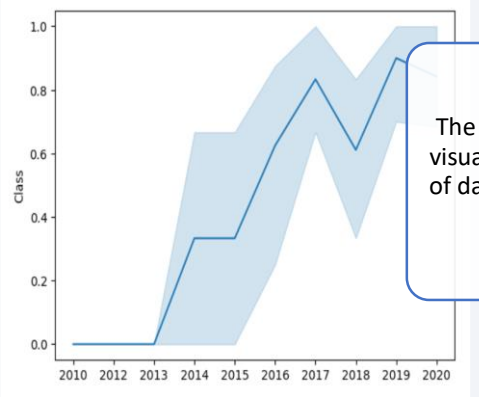
Used to visualize the relationships between variables through each value

Bar chart



Presents categorical data with proportional lengths of rectangular bars to the values

Line plot



The best fits for visualizing trend of data in a time period

EDA with SQL

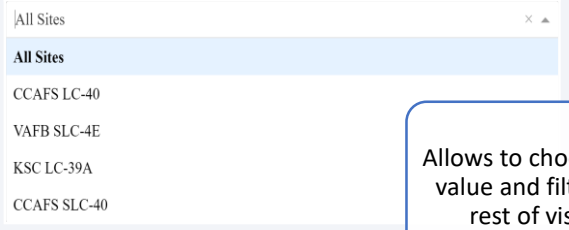
- In the research was used RDBMS SQLite in Python environment
- EDA was applied with SQL to get insights from the data. Next tasks were completed:
  - The names of the unique launch sites in the space mission
  - 5 records where launch sites begin with the string 'CCA'
  - The total payload mass carried by boosters launched by NASA (CRS)the names of the unique launch sites in the space mission
  - Average payload mass carried by booster version F9 v1.1
  - List of the date when the first successful landing outcome in ground pad was acheived
  - List of the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
  - List of the total number of successful and failure mission outcomes
  - List of the names of the booster_versions which have carried the maximum payload mass
  - List of the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015
  - Ranked count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order

Build an Interactive Map with Folium

- All launch sites were marked on a map (4 Launch Sites in Total = 4 marks)
- Color markers were added to the Data Frame for identifing launches' successful, therefore, markers of each launch was added on the map
- The distances were calculated between a launch site and some objects to answer the questions:
  - Are launch sites in close proximity to railways?
  - Are launch sites in close proximity to highways?
  - Are launch sites in close proximity to coastline?
  - Do launch sites keep certain distance away from cities?

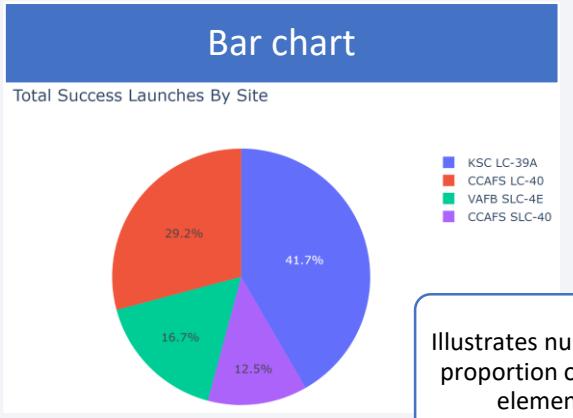
Build a Dashboard with Plotly Dash

Drop-down list



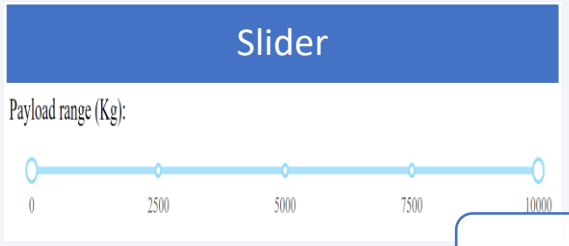
Allows to choose one value and filter the rest of visual elements

Bar chart



Illustrates numerical proportion of each element

Slider

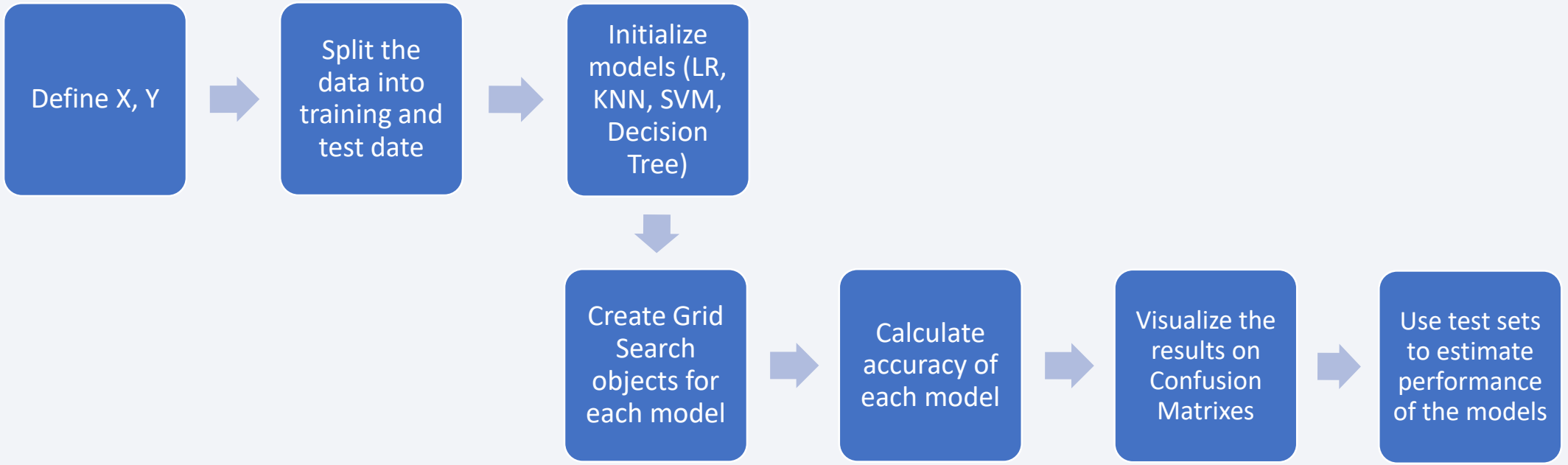


Filters values of the dataset by moving an indicator

Scatter plot

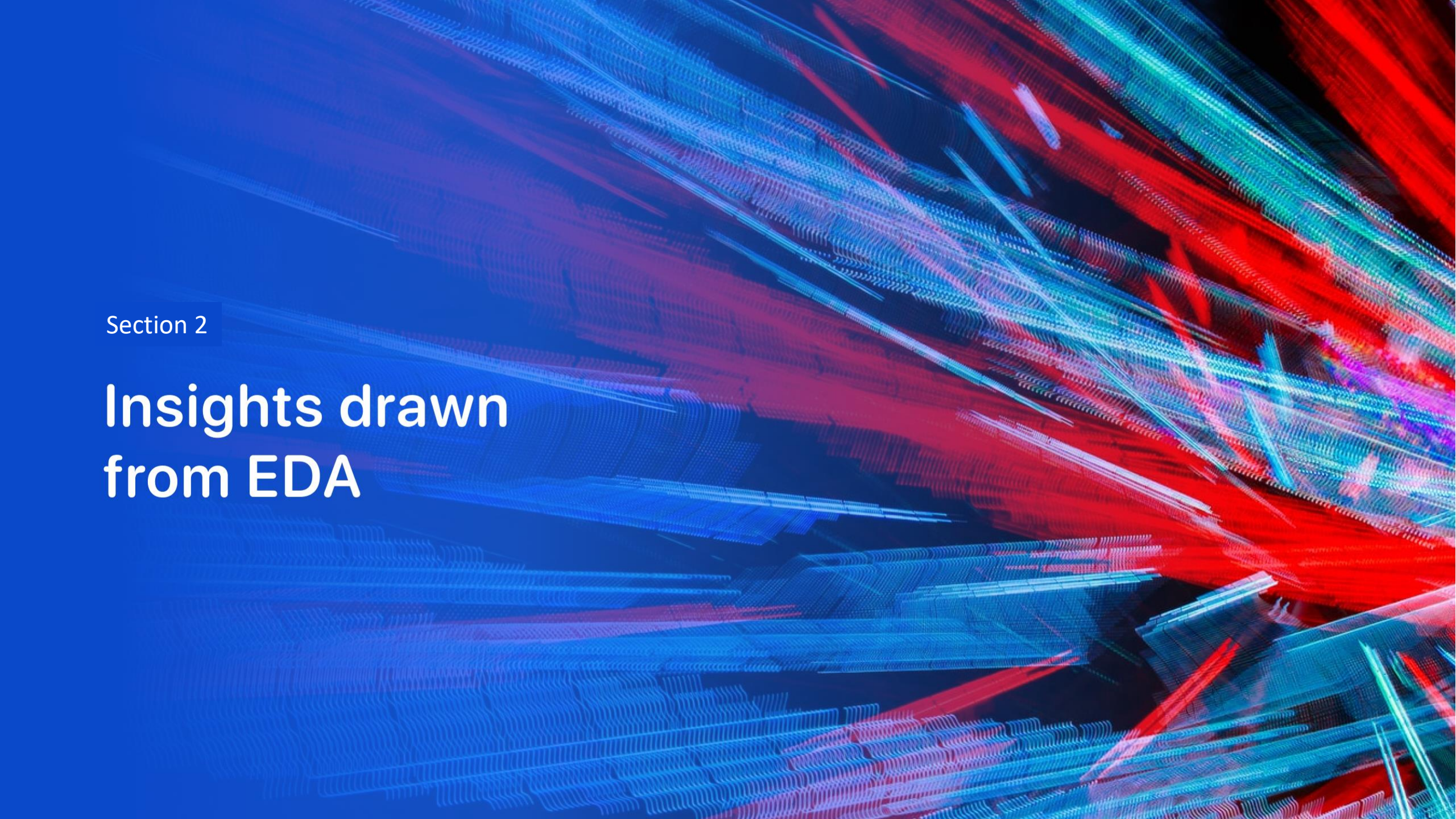
Demonstrates correlation between 2 variables

Predictive Analysis (Classification)



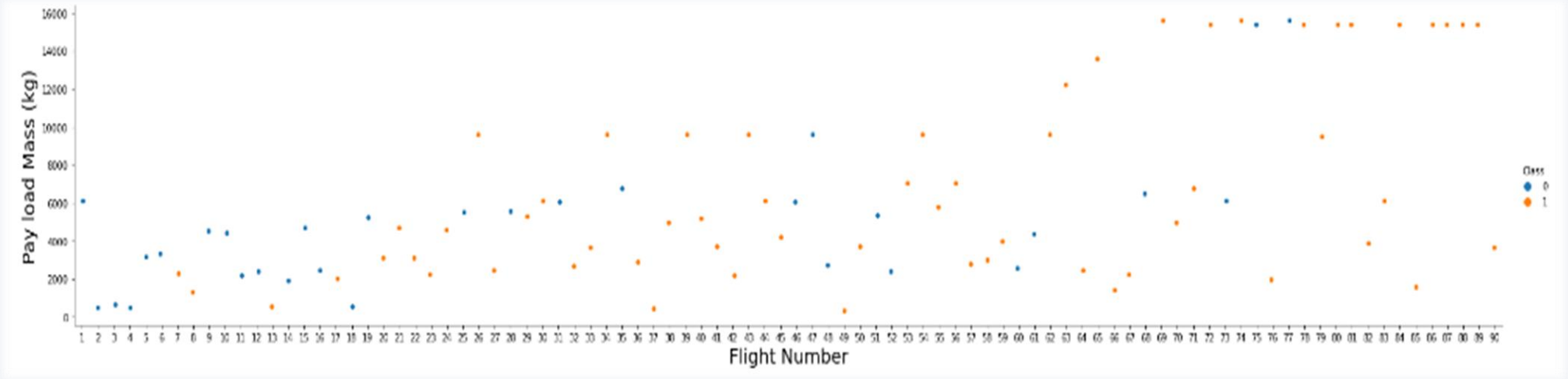
Results

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


Section 2
Insights drawn from EDA

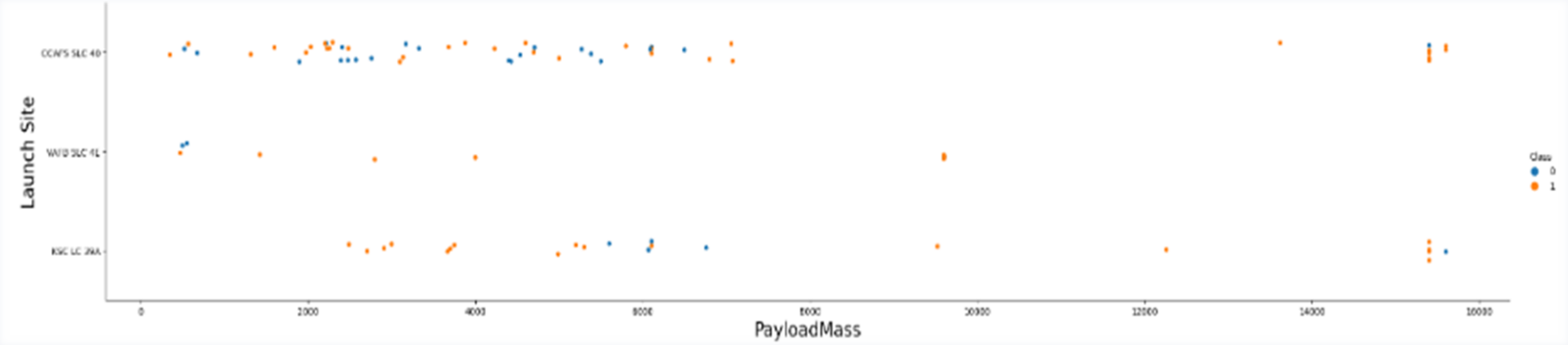
Flight Number vs. Launch Site

- As the flight number increases, the first stage is more likely to land successfully. Likewise, the more massive the payload, the less likely the first stage will return



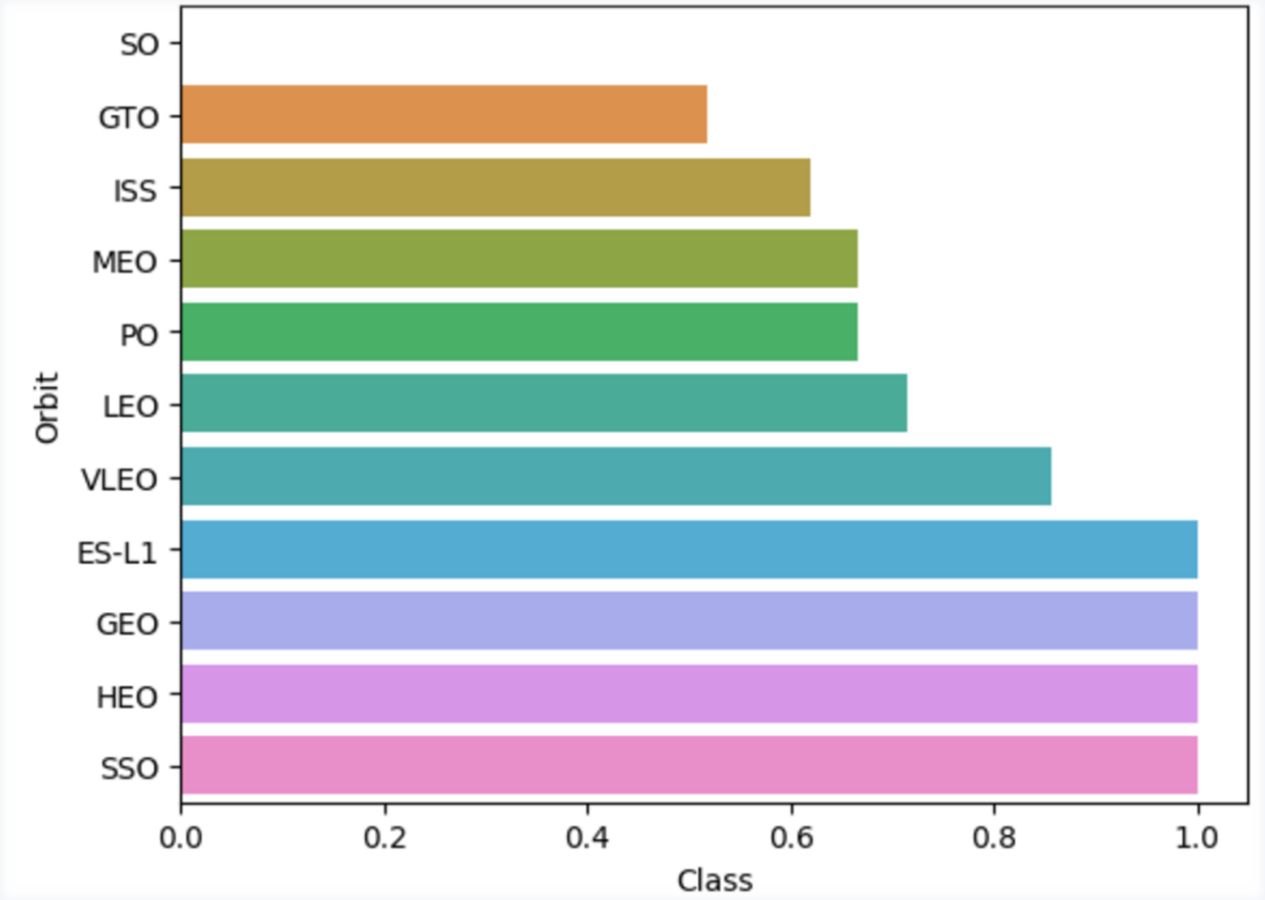
Payload vs. Launch Site

- The chart demonstrates that very high value of Payload Mass occurs higher success of a launch



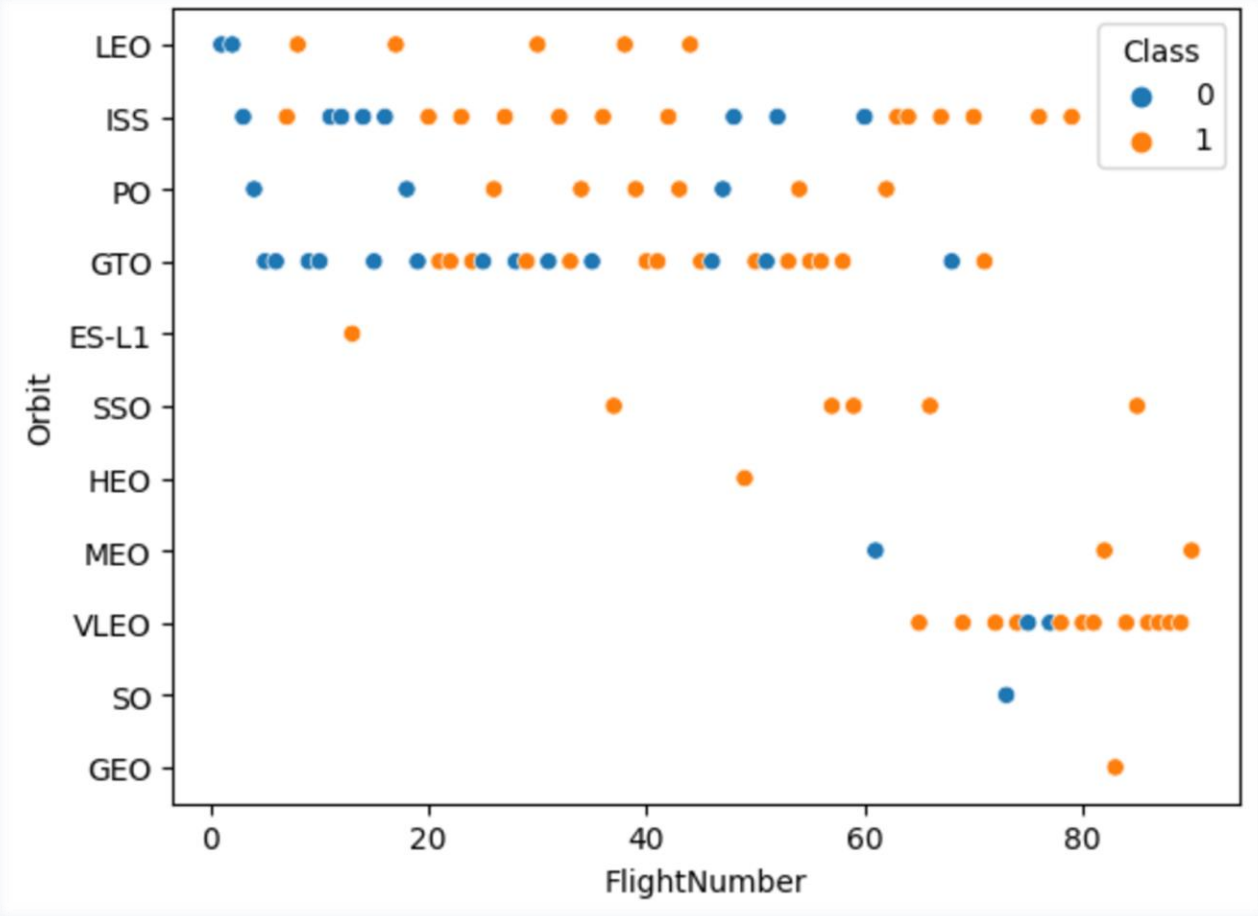
Success Rate vs. Orbit Type

- As we can see on the bar chart, SSO, HEO, GEO and ES-L1 orbits have the highest success rate
- On the other hand, GTO and ISS have the worst estimated success rate



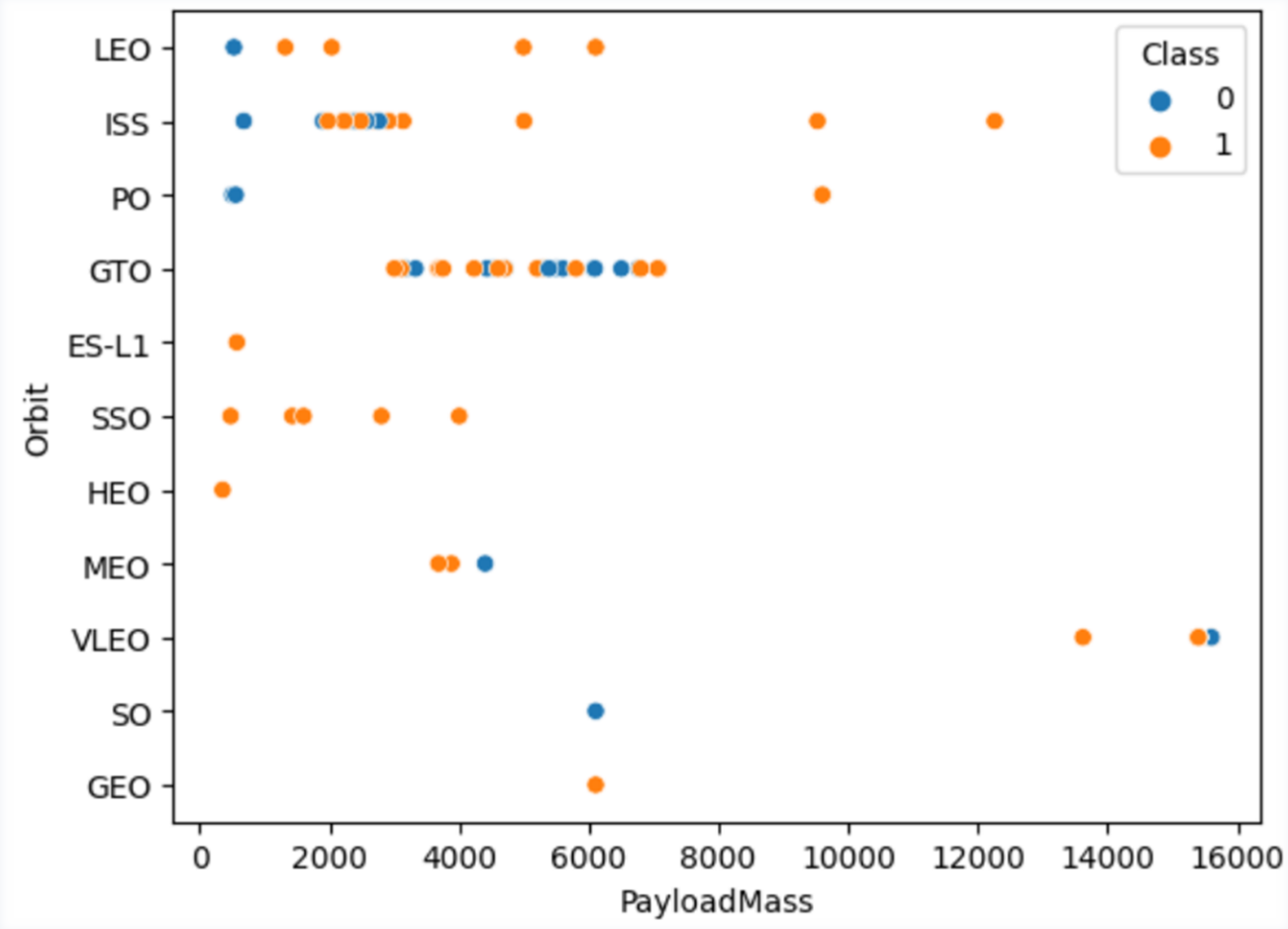
Flight Number vs. Orbit Type

- In the LEO orbit the success appears related to the number of flights
- Meanwhile, there seems to be no relationship between flight number when in GTO orbit



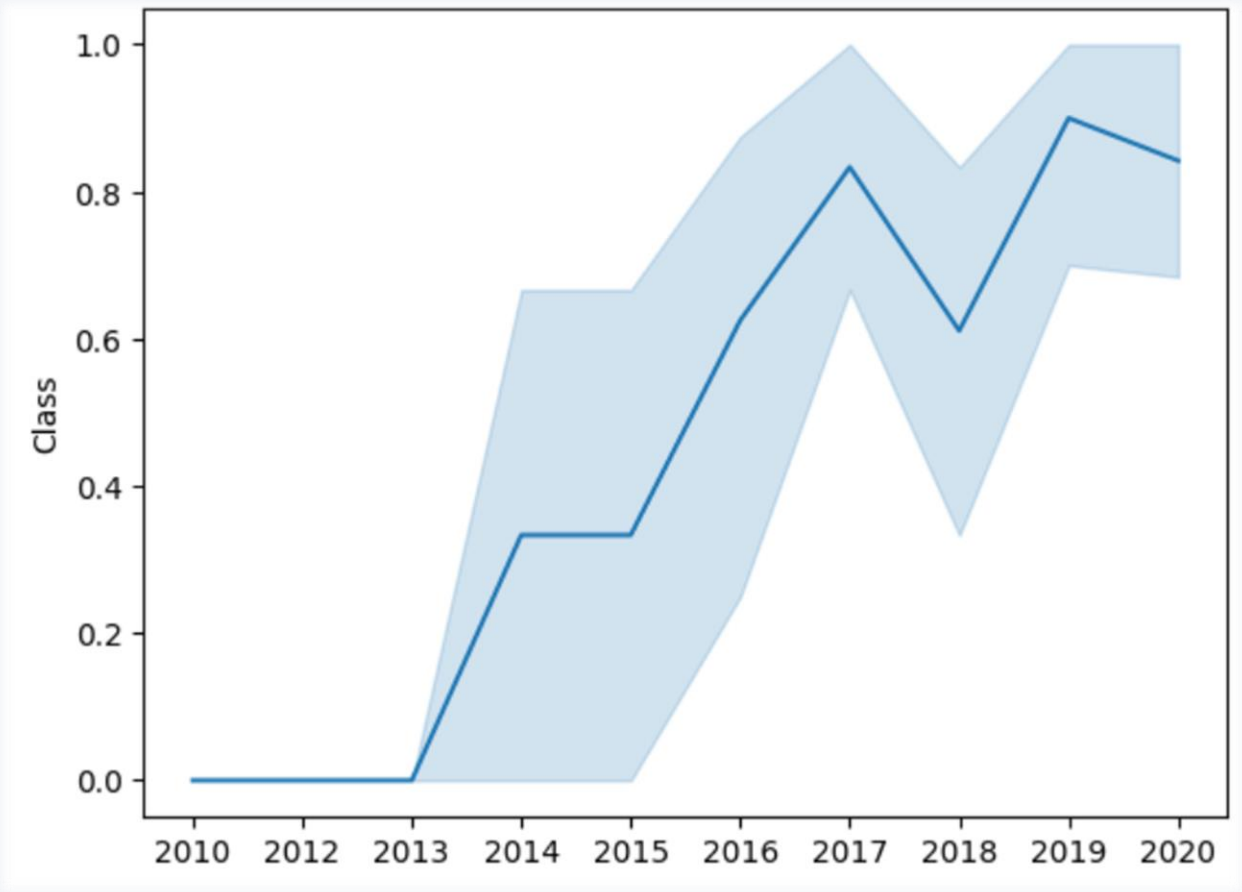
Payload vs. Orbit Type

- With heavy payloads the successful landing or positive landing rate are more for LEO and ISS.
- However for GTO we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission) are both there here.



Launch Success Yearly Trend

- The line chart demonstrates that the success rate have been increasing since 2013



All Launch Site Names

```
%%sql
SELECT DISTINCT(Landing_Outcome) as Unique_Landing_Outcomes
FROM SPACEXTBL
ORDER BY Unique_Landing_Outcomes ASC
```

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

Unique_Landing_Outcomes
Controlled (ocean)
Failure
Failure (drone ship)
Failure (parachute)
No attempt
No attempt
Precluded (drone ship)
Success
Success (drone ship)
Success (ground pad)
Uncontrolled (ocean)

- To display unique parameters of a column, DISTINCT() function is used. To do so, set column name as an argument

Launch Site Names Begin with 'CCA'

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

```
%%sql
SELECT *
FROM SPACEXTBL
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
04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX
08-12-2010	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
22-05-2012	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)
08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)
01-03-2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)

- LIMIT was used to retrieve TOP 5 records in SQLite code
- Data filtering is being done into the WHERE statement

Total Payload Mass

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

```
%%sql
SELECT SUM(PAYLOAD_MASS__KG_) as Total_Payload
FROM SPACEXTBL
WHERE Customer = 'NASA (CRS)'
```

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

Total_Payload
45596

- SUM() is an aggregation function that takes a numerical column as a parameter and returns sum of the column

Average Payload Mass by F9 v1.1

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

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

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

Average_Payload
2928.4

- The main idea behind AVG() is properly the same as for SUM(). It takes a numerical column as an argument and returns an average value

First Successful Ground Landing Date

- To retrieve the date when the first successful landing outcome in ground pad was achieved, a query have to filter date through a subquery in the WHERE statement. As SQLite doesn't provide date type, an interpretation with SUBSTR was used

```
%%sql
SELECT *
FROM SPACEXTBL
WHERE SUBSTR(Date,7)||SUBSTR(Date,4,2)||SUBSTR(Date,1,2) = (SELECT MIN(SUBSTR(Date,7)||SUBSTR(Date,4,2)||SUBSTR(Date,1,2) )
                FROM SPACEXTBL
                WHERE Landing_Outcome LIKE 'Success (ground pad)')
```

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
22-12-2015	01:29:00	F9 FT B1019	CCAFS LC-40	OG2 Mission 2 11 Orbcomm-OG2 satellites	2034	LEO	Orbcomm	Success	Success (ground pad)

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%%sql
SELECT Customer
FROM SPACEXTBL
WHERE Landing_Outcome = 'Success (drone ship)' AND
(PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 6000)
```

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

Customer
SKY Perfect JSAT Group
SKY Perfect JSAT Group
SES
SES EchoStar

- To filter data in this way, operator AND was used into the WHERE clause

Total Number of Successful and Failure Mission Outcomes

- Rows of the data base were countered and grouped to get a list of mission outcomes

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

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

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

Boosters Carried Maximum Payload

- In this case was used the similar approach as for date filtering. The subquery contains maximum value in WHERE clause

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

```
 * 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

2015 Launch Records

- For converting date into month names, a mapping table was created. Then the mapping table was joined to the core data base (LEFT JOIN) and filtered in the WHERE clause with operator AND

```
%%sql
CREATE TABLE Months_Mapping(
    Month_Number VARCHAR(20),
    Month_Name VARCHAR(20)
);
INSERT INTO Months_Mapping(Month_Number, Month_Name) VALUES
('01', 'January'),
('02', 'February'),
('03', 'March'),
('04', 'April'),
('05', 'May'),
('06', 'June'),
('07', 'July'),
('08', 'August'),
('09', 'September'),
('10', 'October'),
('11', 'November'),
('12', 'December');
```

```
 * sqlite:///my_data1.db
(sqlite3.OperationalError) table Months_Mapping already exists
[SQL: CREATE TABLE Months_Mapping(
    Month_Number VARCHAR(20),
    Month_Name VARCHAR(20)
);]
(Background on this error at: http://sqlalche.me/e/e3q8)
```

```
%%sql
SELECT Months_Mapping.Month_Name, SPACEXTBL.Landing_Outcome, SPACEXTBL.Booster_Version, SPACEXTBL.Launch_Site
FROM SPACEXTBL
LEFT JOIN Months_Mapping on substr(SPACEXTBL.Date, 4, 2) = Months_Mapping.Month_Number
WHERE Landing_Outcome = 'Failure (drone ship)' AND substr(Date,7,4)='2015'
```

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

Month_Name	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

- Ranking in SQLite provides through RANK () OVER (ORDER BY …) RANK
- Further code is similar to previous cases with WHERE, AND, GROUP BY

```
%%sql
SELECT Landing_Outcome, COUNT(*) as Boosts,
RANK () OVER (
ORDER BY COUNT(*) DESC
) Rank
FROM SPACEXTBL
WHERE Date BETWEEN '04-06-2010' AND '20-03-2017'
GROUP BY Landing_Outcome
```

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

Landing_Outcome	Boosts	Rank
Success	20	1
No attempt	10	2
Success (drone ship)	8	3
Success (ground pad)	6	4
Failure (drone ship)	4	5
Failure	3	6
Controlled (ocean)	3	6
Failure (parachute)	2	8
No attempt	1	9

Section 3

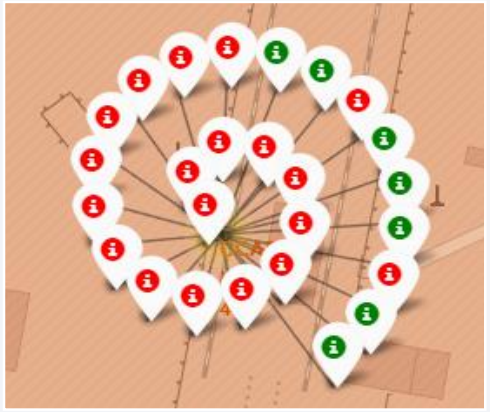
Launch Sites Proximities Analysis

Launch Site on a global map

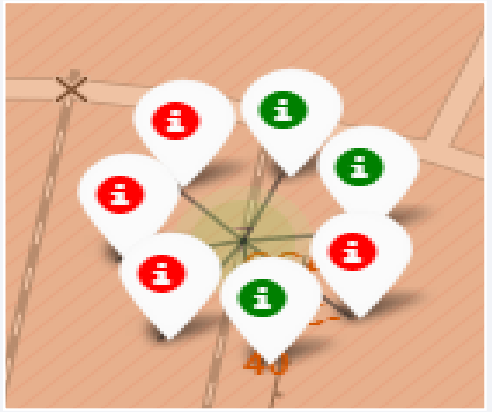
- All launch sites based in the USA coasts (Florida and California)

Launch Sites successful outcomes

CCAFS LC-40



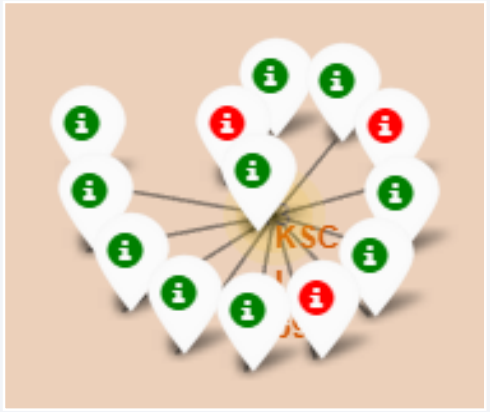
CCAFS SLC-40



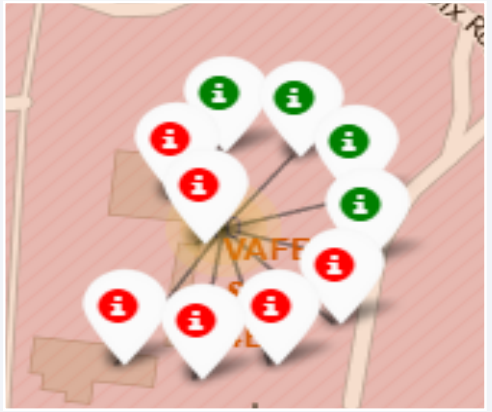
Successful launch

Failed launch

KSC LC-39A

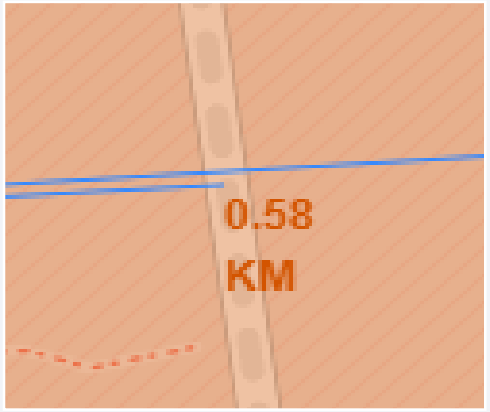


VAFB SLC-4E

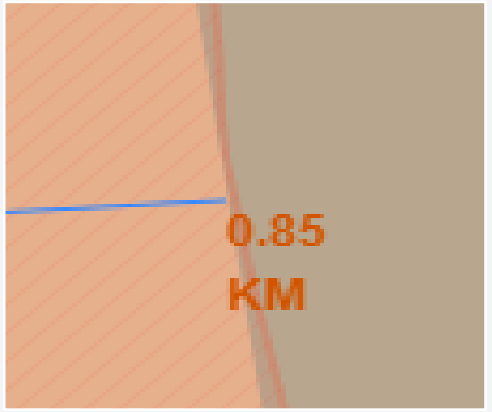


Distance from CCAFS SLC-40 to infrastructure elements

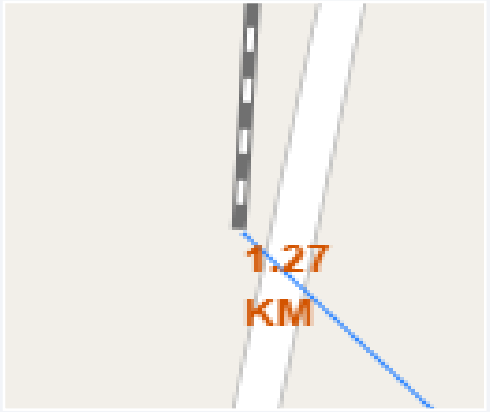
Highways (close proximity)



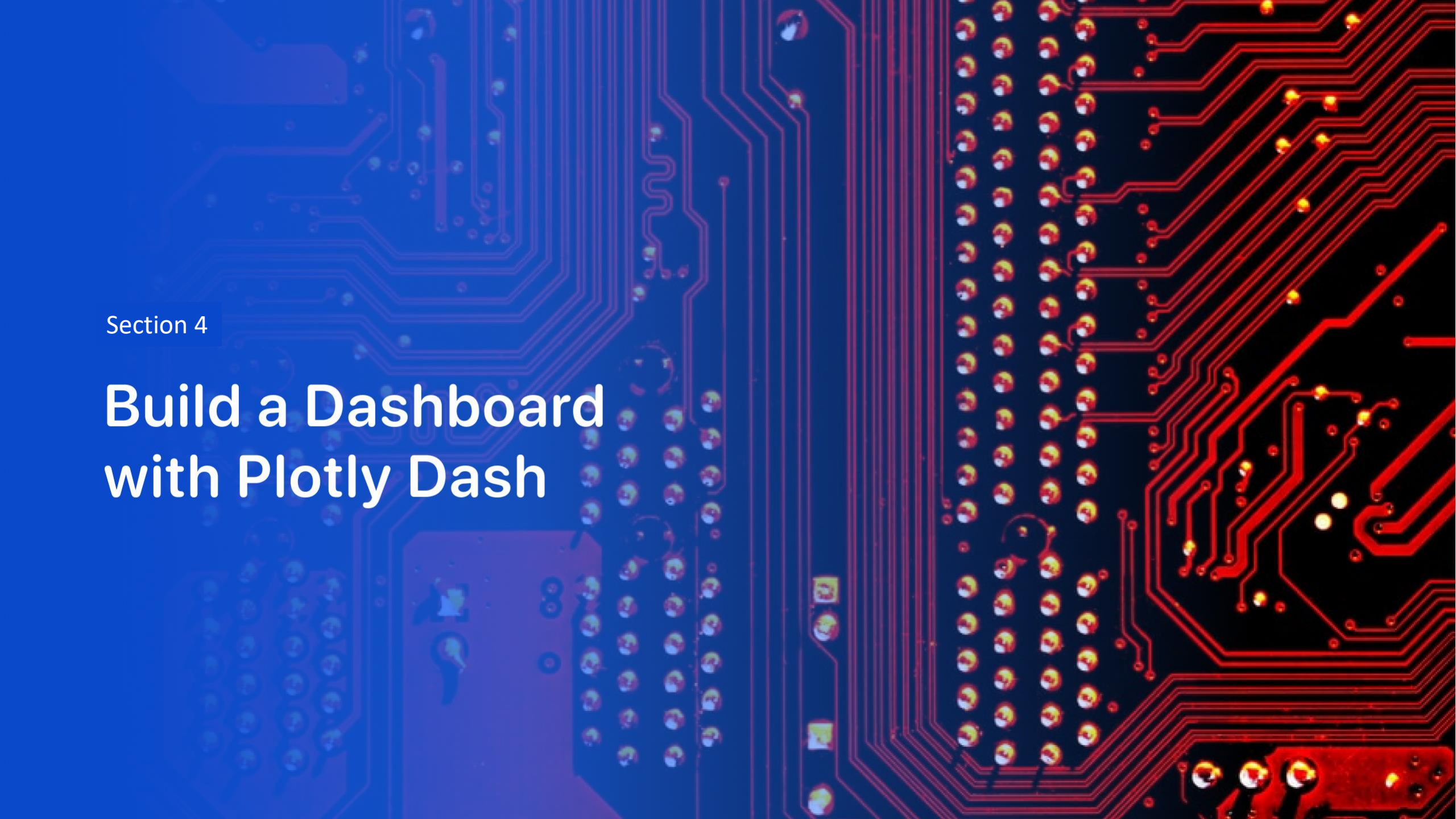
Coast (close proximity)



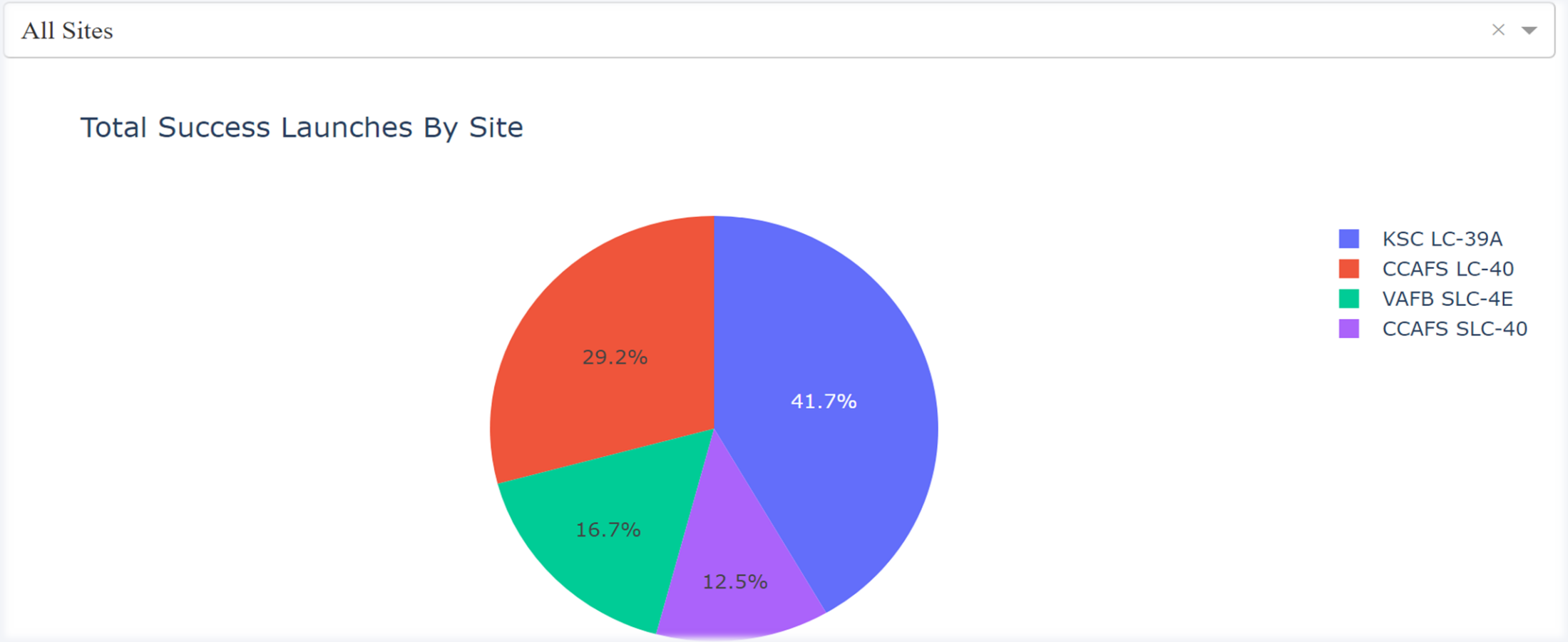
Railways (close proximity)



City (faraway)

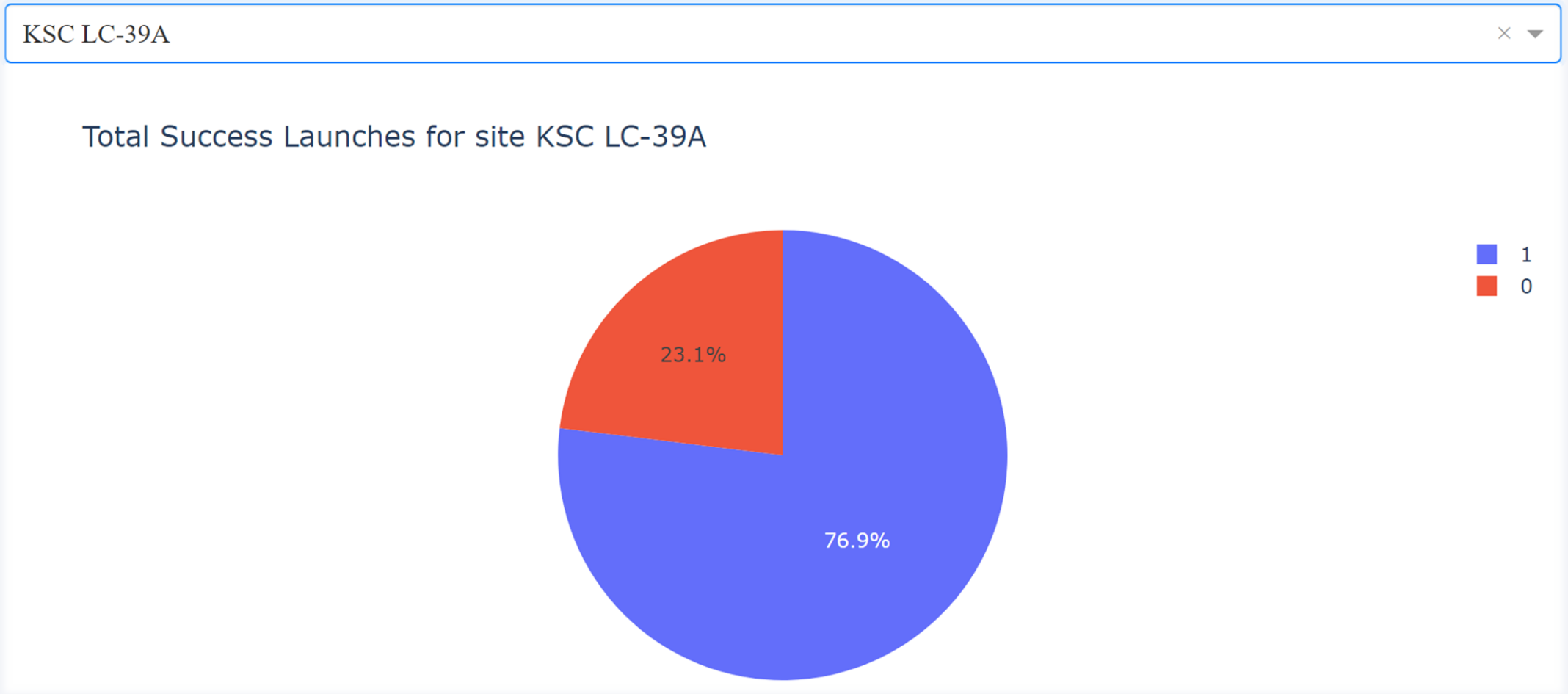

Section 4
Build a Dashboard with Plotly Dash

Success ratio in terms of launch sites



- Respectively to the pie chart above, KSC LC-39A has the greatest percentage of successful Launces

KSC LC-39A success/fail ratio



- KSC LC-39A achieved a 76.9% success rate which is a splendid work

Correlation between Payload and Success in terms of Booster Version

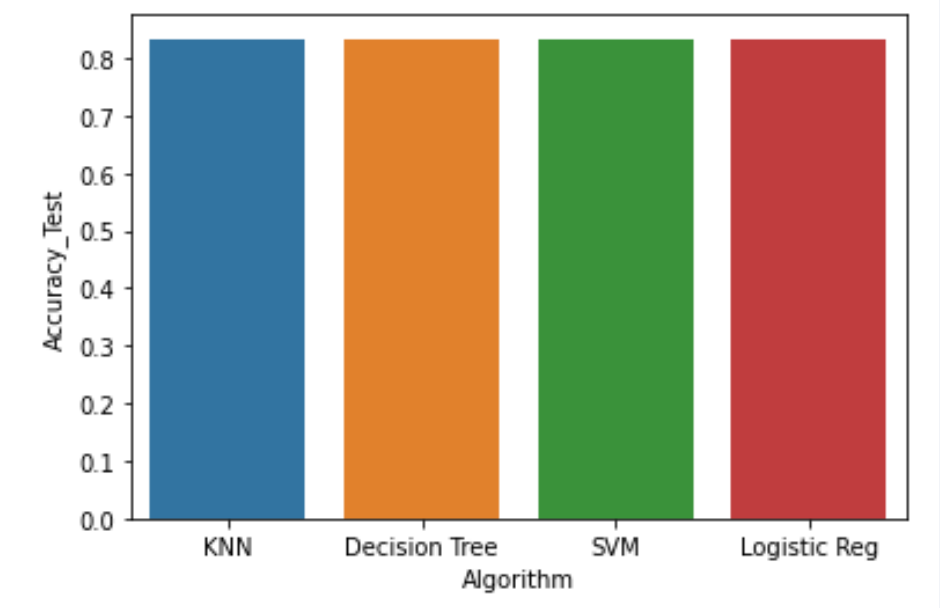
- The best performance is observed on the scatter plot with “FT” Booster. “v1.1” Booster has the lowest rate of success.

Section 5

Predictive Analysis (Classification)

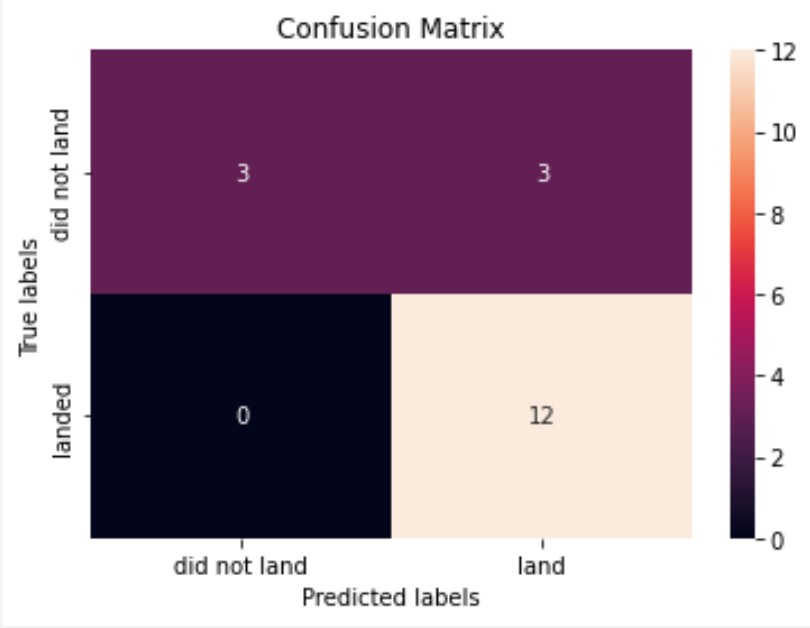
Classification Accuracy

- Given test data was fitted for the entire stack of algorithms. As shown on the bar chart, all of the algorithms have exactly the same value of accuracy.
- The phenomena was caused because of small sample of data.
- Besides, small sample of data negatively affects on further iterations, for instance, large variance in accuracy (as example, Decision Tree)



Confusion Matrix

- Confusion Matrix is undistinguished due to the same accuracy across all models
- Confusion Matrix insides:
  - 12 True Positives
  - 3 True Negatives
  - 3 False Negative (which are erroneous)



Conclusions

- Was defined that the models requires more data then 18 samples for dataset. However, the accuracy on a test set is equal to 83.33% for landing the 1st stage booster
- Launch success rate have been increasing since 2013
- KSC LC-39A is the most successful launce site and should be analyzed in more details
- The most successful launces were observed with higher Payload Mass and “FT” Booster Version
- Successful rate increases with number of flights

Thank you!