IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Stuart Currie
18/11/2023

Outline

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

Executive Summary

Summary of Methodologies:

- SpaceX launch data was collected via the SpaceX REST API and webscraping Wikipedia
- Data was then cleaned (removing empty or non-values), and landing outcomes classified as a 1 (successful) or 0 (unsuccessful)
- Exploratory data analysis was performed using SQL and data visulaisation through an interactive Plotly Dashboard
- A Predicitve Analysis was then performed using machine learning to classify data using logistical regression, support vector machines, a decision tree classifier, and K nearest neighbour

Summary of Results:

- Launches with a low payload mass were more successful that those with a high payload mass
- Orbits ES-L1, GEO, HEO, and SSO have the highest success rate (100%)
- Booster B5 has the highest success rate of all boosters
- Most launch sites are near to the Equator, and all are close to the coast
- Landing success rate increased as years progressed
- Decision Tree Classifier is the best classification algorithm for this dataset

Introduction

Project Background:

In the commercial space age, SpaceX is one of the most successful companies – with accomplishments such as sending space craft to the International Space Station, Starlink satellites providing internet access around the world, and sending manned rockets into space. One of the reasons SpaceX has been so successful is by keeping costs down through a recoverable first stage – spending $62 million for a Falcon9 launch where it's competitors will spend around $165 million per launch. However, the first stage is not always recoverable or reused.

Our fictional client, SpaceY, would like to compete with SpaceX. To do so we will need to find out in which situations the SpaceX first stage is recoverable and which variables affect this.

Key Questions:

- How do variables such as payload mass, launch site, number of flights, and orbits affect the success rate of the first stage landing?
- Does the rate of successful landing increase over time?
- Which machine learning model can classify and predict the landing outcome with the highest accuracy?

Section 1

Methodology

Methodology

- Data collection methodology:
  - SpaceX REST API
  - Webscraping Wikipedia
- Perform data wrangling
  - Data was Filtered for Falcon9 launches only
  - Data was cleaned by replacing empty or NAN values with the average
  - Landing was classified as 1 (successful) or 0 (unsuccessful) using one hot encoding
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Building, tuning and evaluating classification modelling using logistical regression, SVM, decision tree, and k nearest neighbours

Data Collection

Data sets were collected using SpaceX REST API to send a GET request to obtain data, and webscraping the SpaceX Wikipedia page

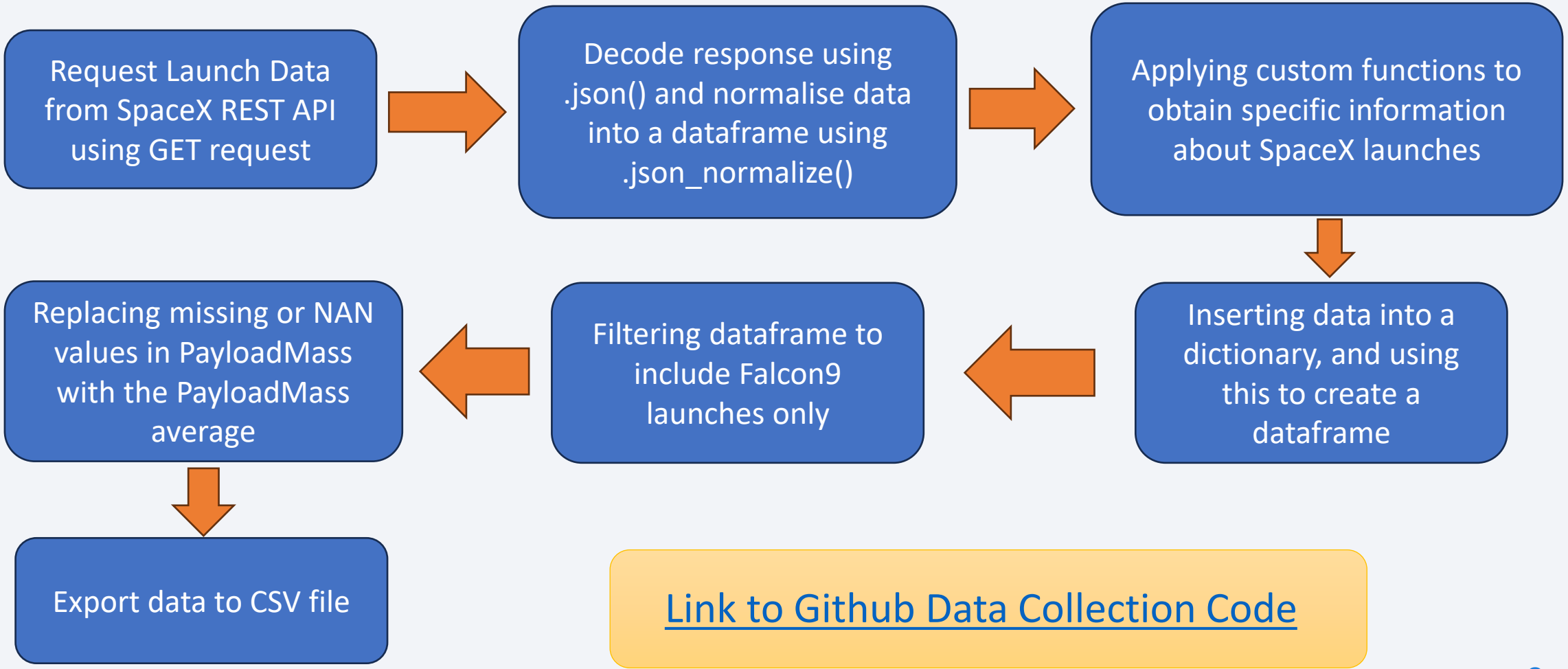
Fields Collected Through SpaceX REST API:

- FlightNumber
- Date
- BoosterVersion
- PayloadMass
- Orbit
- LaunchSite
- Outcome
- Flights
- GridFins
- Reused
- Legs
- LandingPad
- Block
- ReusedCount
- Serial
- Longitude and Latitiude

Fields Collected Through Wikipedia Webscraping:

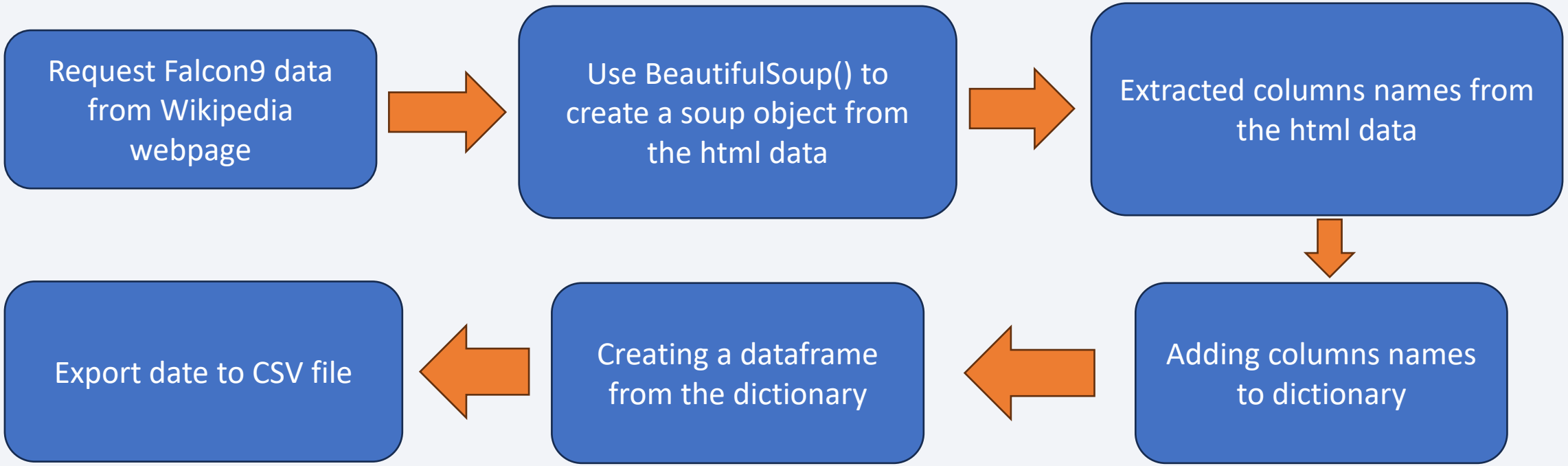
- FlightNo
- Launchsite
- Payload
- PayloadMass
- Orbit
- Customer
- Launch Outcome
- VersionBooster
- BoosterLanding
- Date
- Time

Data Collection – SpaceX API



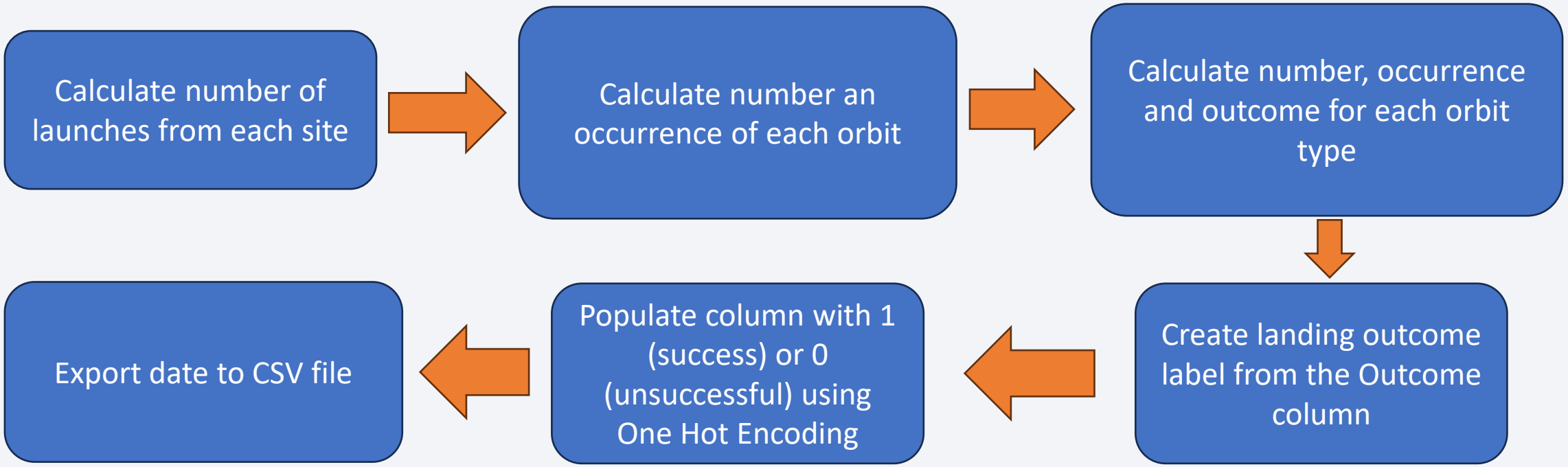
Link to Github Data Collection Code

Data Collection - Scraping



Link to Github Webscraping Code

Data Wrangling



Link to Github Data Wrangling Code

EDA with Data Visualization

Charts Plotted:

- Flight Number vs Payload Mass
- Flight Number vs Launch Site
- Payload Mass vs Launch Site
- Success Rate for each Orbit Type
- Flight Number vs Orbit Type
- Payload Mass vs Orbit Type
- Yearly Success Rate

- Line plots – used to show changes over time
- Bar plots – used to show categorical variable contribution to landing success
- Scatter plots – used to show relationship between variables

Link to Github EDA Data Visualisation Code

EDA with SQL

SQL Queries:

- Displaying unique Launch Site names
- Displaying 5 entries where the Launch Site begins with “CCA”
- Displaying the total payload mass carried by boosters launched by NASA (CRS)
- Displaying averagepayloadmass carried by booster version F9 v1.1
- Listing the date when the first succesful landingoutcome in ground pad was achieved
- Listing the names of the boosters which have success in drone ship and have payloadmass greater than 4000 but less than 6000
- Listing the total number of successful and failure mission outcomes
- Listing the names of the Booster Versions which have carried the maximum Payload Mass
- Listing the failed landing outcomes in drone ship, their booster versions and launch site names per month for the year 2015
- 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

Link to Github EDA SQL Code

Build an Interactive Map with Folium

Launch Site Markers:

- Circle markers for all launch sites (using longitutde and latitude data) and NASA Johnson Space Centre
- Coloured markers to denote if landing was successful (green) or unsuccessful (red)

Line Markers:

- Lines added to show distance between launch site KSC LC-39A and nearby coastline, rail line, highway and closest city

Link to Github Folium Code

Build a Dashboard with Plotly Dash

Launch Site Dropdown List:

- Used to select either all or a single Launch Site to display data for

Pie Chart of Successful Launches:

- Displays total count of successful landing for each Launch Site, or the specific Successful/Unsuccessful landing count if a specific Launch Site was selected from the dropdown list

Slider for Payload Mass:

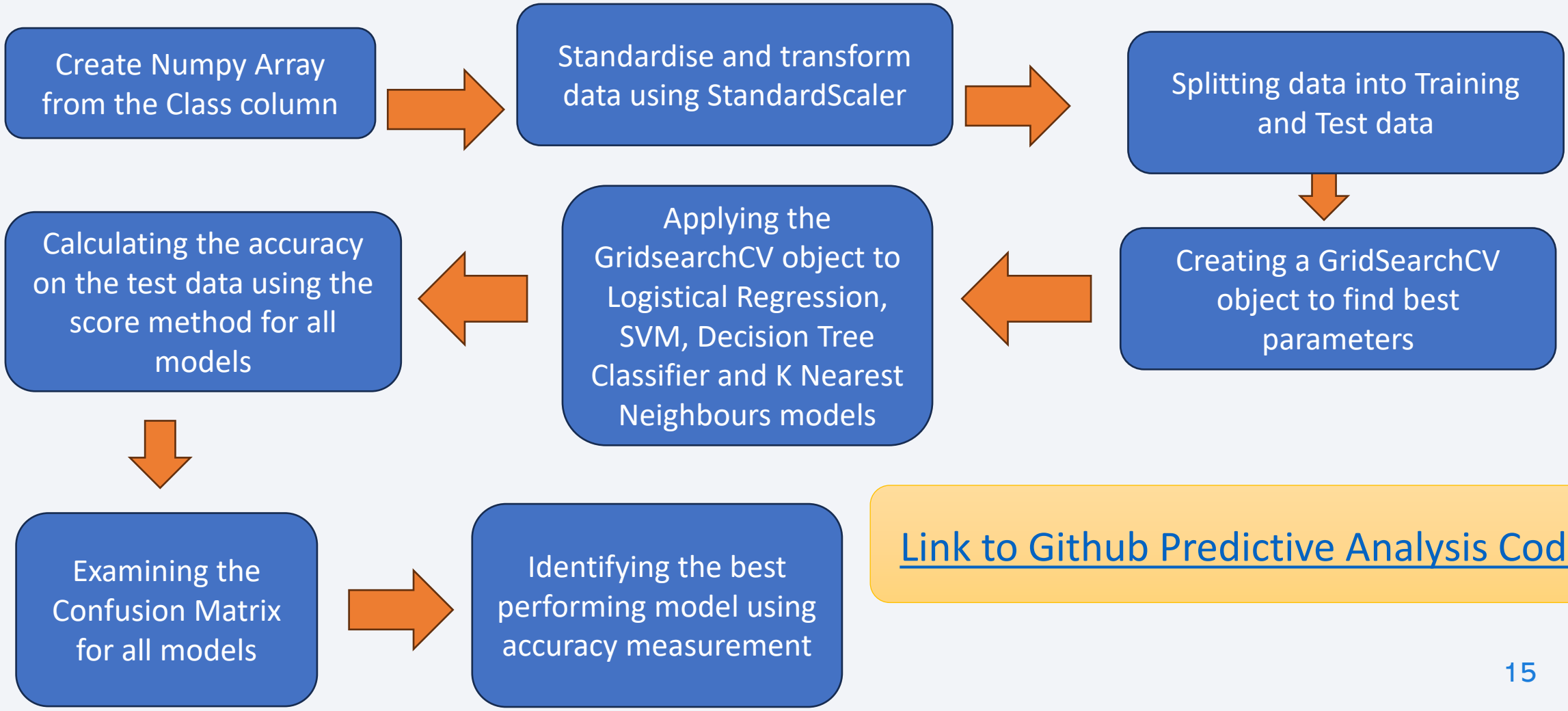
- Used to adjust which Payload Mass data is shown in visualisations (from 0 to 10000kg)

Scatter Plot of Payload Mass vs Success Rate for different Booster Versions:

- Displays relationship/correlations between Payload mass and Landing Success rate for different Booster Versions

Link to Github Plotly Dash Code

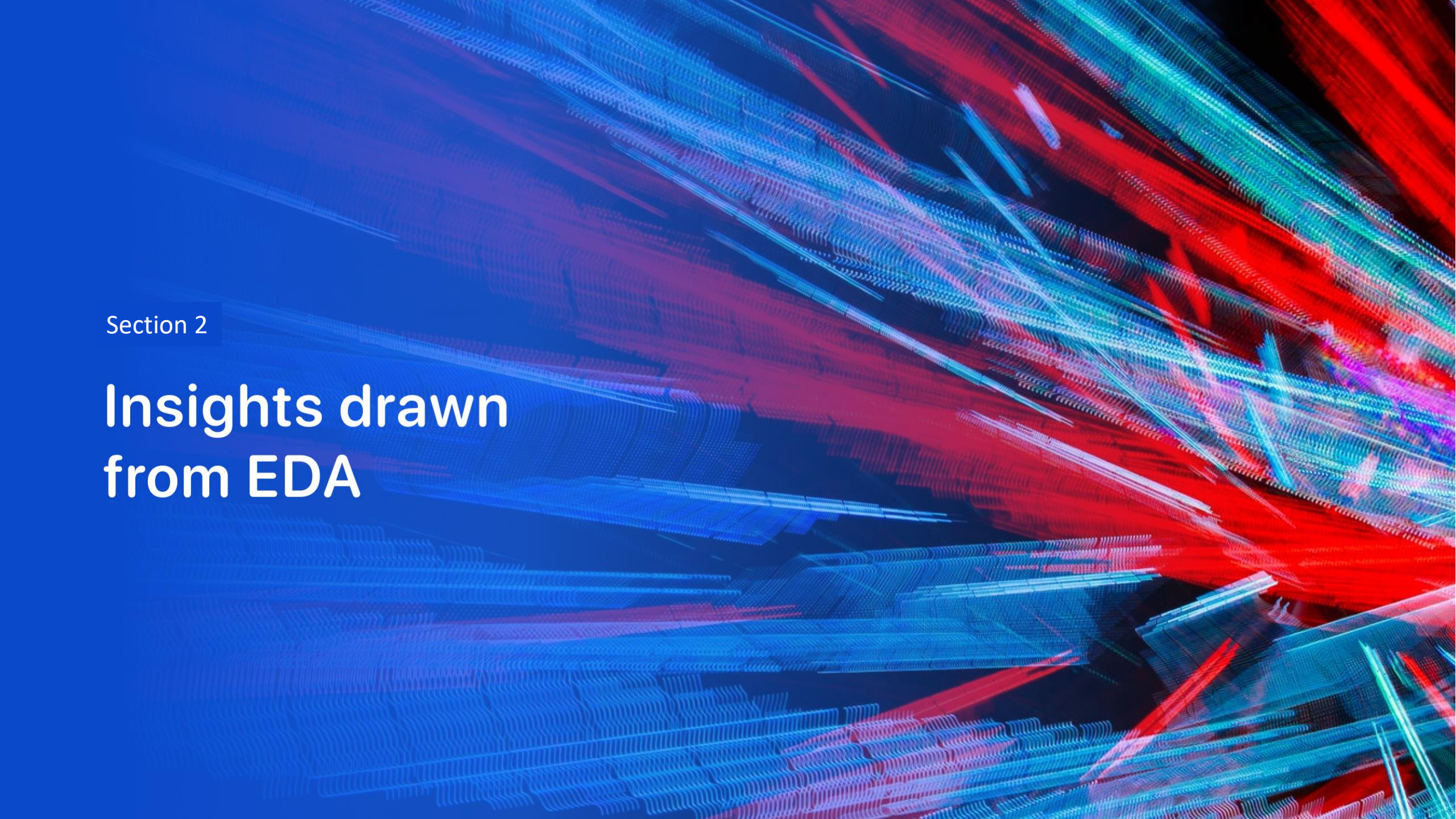
Predictive Analysis (Classification)



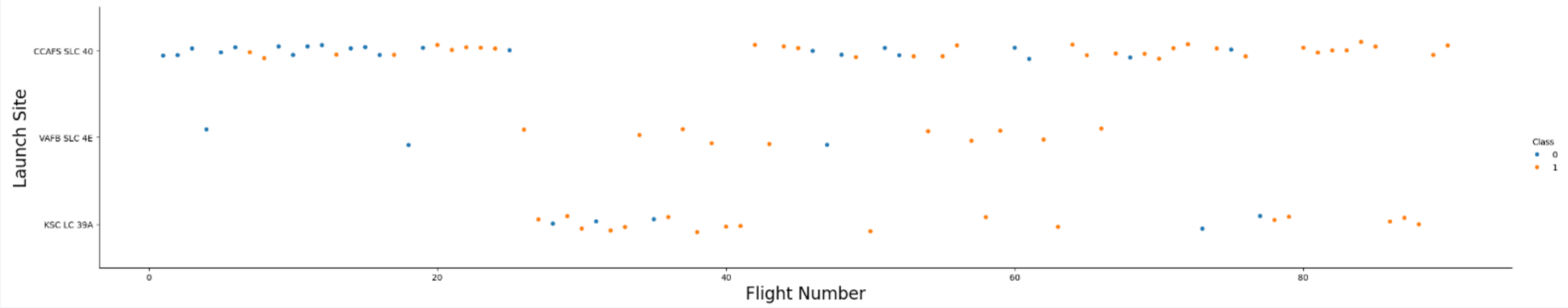
Link to Github Predictive Analysis Code

Results

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


Section 2
Insights drawn from EDA

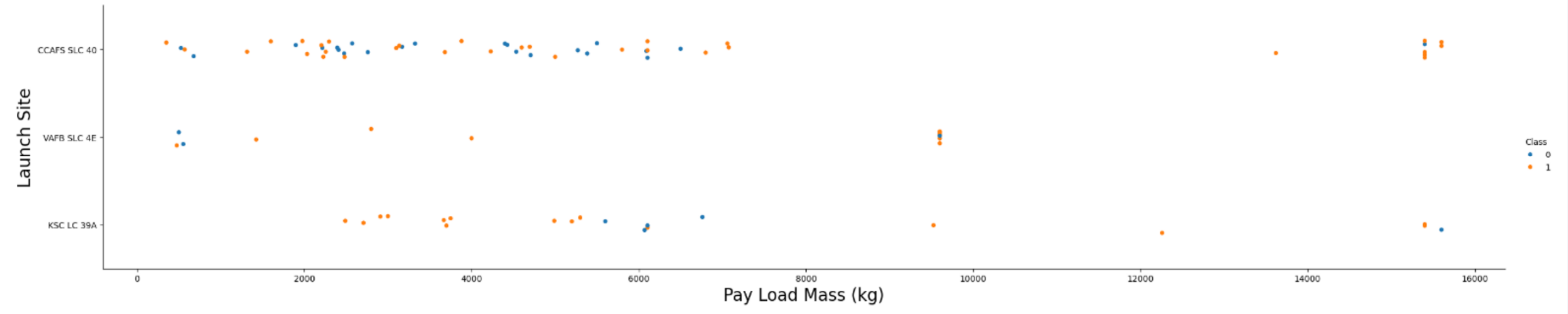
Flight Number vs. Launch Site



Interpretation:

- Earlier launches were less successful, later launches more successful
- Launch sites VAFB SLC 4E and KSC LE 39A had higher success rates
- Launch site CCAFS SLC 40 has roughly even split of successful and unsuccessful landings

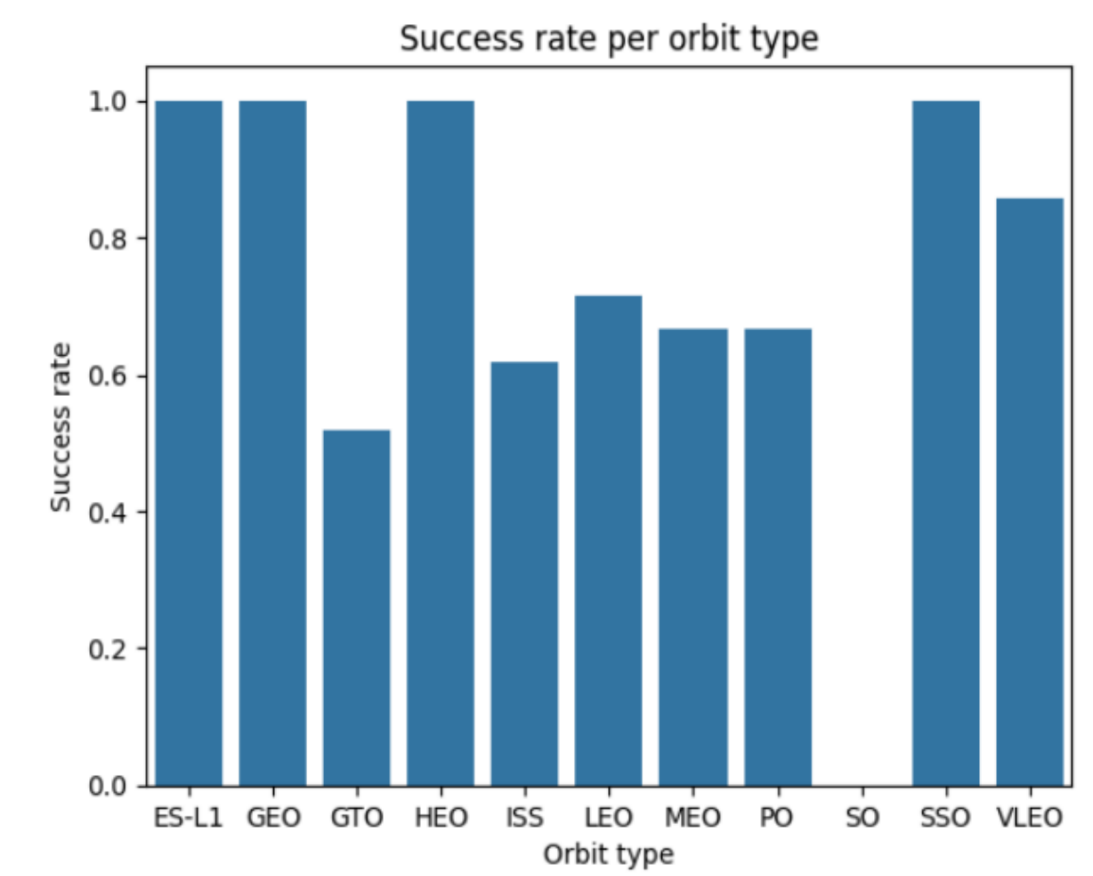
Payload vs. Launch Site



Interpretation:

- As the Payload Mass increased, so did the success rate
- Launch site KSC LC 39A had more success with lighter payloads (under 6000kg)

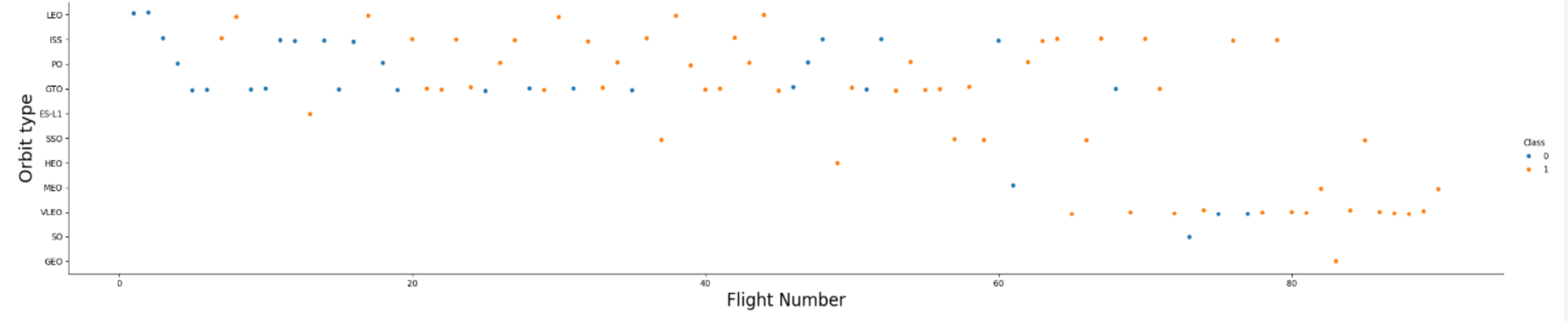
Success Rate vs. Orbit Type



Interpretation:

- Orbits ES-L1, GEO, HEO, and SSO have 100% success rate
- Orbits GTO, ISS, LEO, MEO, PO, and VLEO have a success rate between 50% and 80%
- Orbit SO had a 0% success rate

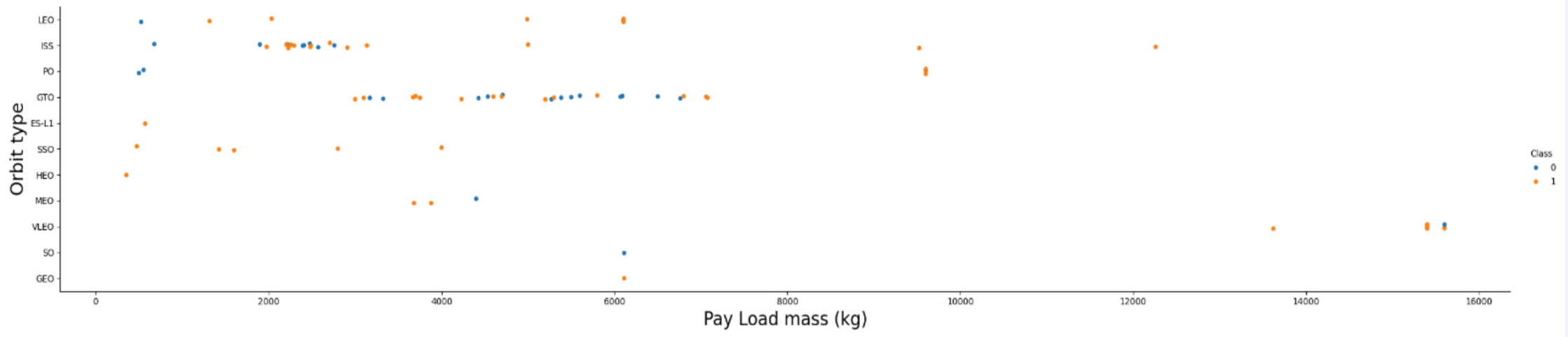
Flight Number vs. Orbit Type



Interpretation:

- As the Flight Number increased (or time progressed), so did the success rate
- As time progressed different Orbit types were performed

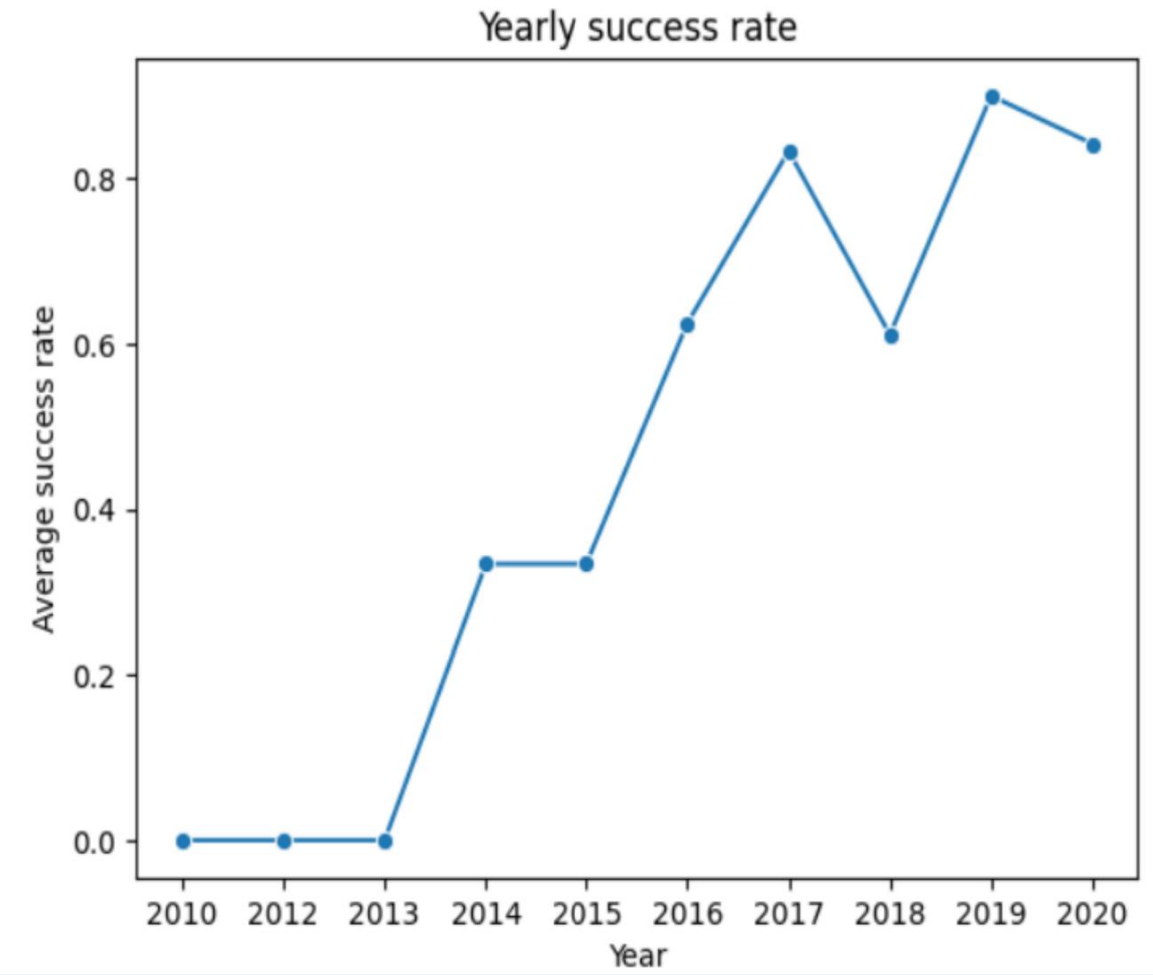
Payload vs. Orbit Type



Interpretation:

- As the Payload Mass increased, so did the success rate for most of the Orbit types
- Orbit type GTO had more of a mixture of successful and unsuccessful landings
- Orbit type ISS and SSO were used for lower Pay Load mass flights

Launch Success Yearly Trend



Interpretation:

- In the first 3 years (2010-2013) there was 0% success rate
- After 2013 the success rate steadily increased (despite slight dips in 2018 and 2020)

All Launch Site Names

```
In [8]: %%sql

        select distinct "Launch_Site" from SPACEXTABLE

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

Out[8]:

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

Interpretation:

All unique Launch Site names

Launch Site Names Begin with 'CCA'

In [9]:

```
%%sql

select * from SPACEXTABLE
where "Launch_Site" like 'CCA%' limit 5
```

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

Out[9]:

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

Interpretation:

First 5 records for entries with Launch site beginning with “CCA”

Total Payload Mass

In [10]:

```
%%sql

select Customer, sum("PAYLOAD_MASS__KG_") as Total_payload_mass
from SPACEXTABLE
where Customer = 'NASA (CRS)'
```

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

Out[10]:

Customer	Total_payload_mass
NASA (CRS)	45596

Interpretation:

Total Payload Mass from NASA (CRS) – 45,596 kg

Average Payload Mass by F9 v1.1

In [11]:
```
%%sql

select "Booster_version", avg("PAYLOAD_MASS__KG_") as Average_payload_mass
from SPACEXTABLE
where "Booster_version" = 'F9 v1.1'
```

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

Out[11]:

Booster_Version	Average_payload_mass
F9 v1.1	2928.4

Interpretation:

Average Payload mass by Booster Version – F9 v1.1 (2928.4kg)

First Successful Ground Landing Date

In [12]:

```
%%sql

select "Landing_Outcome", min(Date) as First_successful_landing
from SPACEXTABLE
where "Landing_Outcome" = 'Success (ground pad)'
```

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

Out[12]:

Landing_Outcome	First_successful_landing
Success (ground pad)	2015-12-22

Interpretation:

First successful ground pad landing (22nd Dec 2015)

Successful Drone Ship Landing with Payload between 4000 and 6000

In [13]:

```
%%sql

select distinct("Booster_Version") as Booster_Version_successful_in_drone_ship
from SPACEXTABLE
where "Landing_Outcome" = 'Success (drone ship)' and "PAYLOAD_MASS__KG_" between 4000 and 6000
```

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

Out[13]:

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

Interpretation:

Successful drone ship landings with payload mass between 4000 and 6000 kg

Total Number of Successful and Failure Mission Outcomes

In [14]:
```
%%sql

select "Mission_Outcome", count("Mission_Outcome")
from SPACEXTABLE
group by "Mission_Outcome"
```

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

Out[14]:

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

Interpretation:

Number of Success and Failed Mission Outcomes

Boosters Carried Maximum Payload

In [15]:

```
%%sql

select "Booster_Version" as max_payload_mass_booster_versions
from SPACEXTABLE
where "PAYLOAD_MASS__KG_" in
(select max("PAYLOAD_MASS__KG_")
from SPACEXTABLE)
```

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

Out[15]:

max_payload_mass_booster_versions
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

Interpretation:

Names of Booster Versions that have carried the Most Payload Mass

2015 Launch Records

In [16]:
```
%%sql

select substr(Date,6,2) as Month, "Landing_Outcome", "Booster_Version", "Launch_Site"
from SPACEXTABLE
where "Landing_Outcome" = 'Failure (drone ship)' and substr(Date,0,5) = '2015'
```

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

Out[16]:

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

Interpretation:

Displaying the Booster Version and Launch Site for unsuccessful landings on drone ships

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

In [17]:
```
%%sql

select Date, "Landing_Outcome", count("Landing_Outcome") as Landing_Outcome_count
from SPACEXTABLE
where Date between '2010-06-04' and '2017-03-20'
group by "Landing_Outcome"
order by Landing_Outcome_count desc
```

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

Out[17]:

Date	Landing_Outcome	Landing_Outcome_count
2012-05-22	No attempt	10
2016-04-08	Success (drone ship)	5
2015-01-10	Failure (drone ship)	5
2015-12-22	Success (ground pad)	3
2014-04-18	Controlled (ocean)	3
2013-09-29	Uncontrolled (ocean)	2
2010-06-04	Failure (parachute)	2
2015-06-28	Precluded (drone ship)	1

Interpretation:

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

Global Map of Launch Sites

In [8]:
```
# Initial the map
site_map = folium.Map(location=nasa_coordinate, zoom_start=5)
# For each launch site, add a Circle object based on its coordinate (Lat, Long) values. In addition, add Launch site name as a popup label
for index,row in launch_sites_df.iterrows():
    coordinate = [row['Lat'], row['Long']]
    circle = folium.Circle(coordinate, radius=1000, color='#d35400', fill=True).add_child(folium.Popup(row['Launch Site']))
    marker = folium.map.Marker(coordinate, icon=DivIcon(icon_size=(20,20),icon_anchor=(0,0),\
                                                        html='<div style="font-size: 12; color:#d35400;"><b>%s</b></div>' % row['Launch Site']))
    site_map.add_child(circle)
    site_map.add_child(marker)

site_map
```

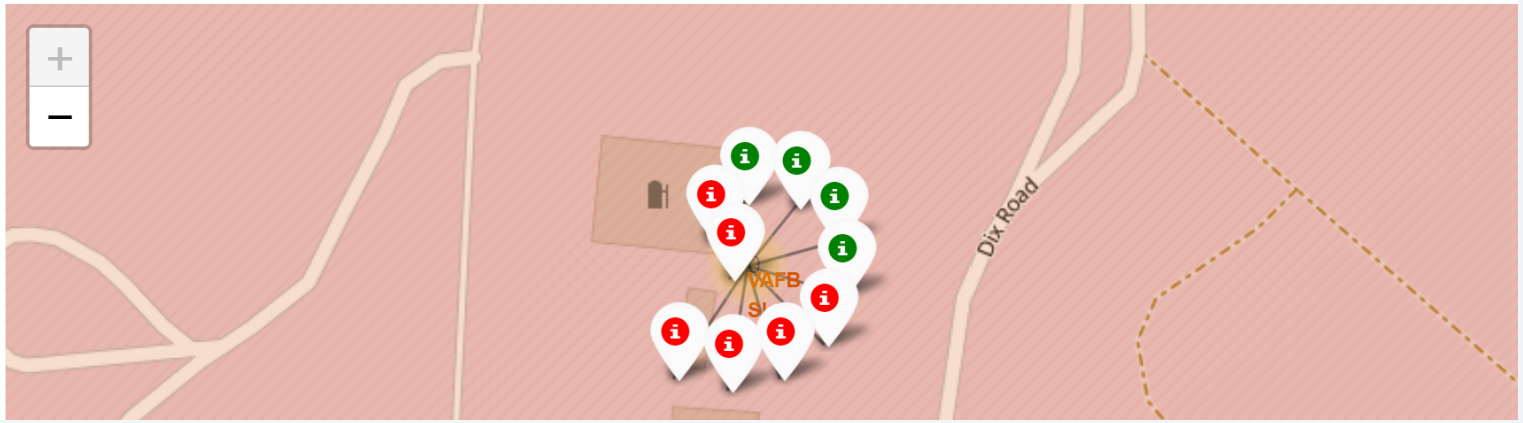
Out[8]:

- These are the locations of all launch sites
- They are all close to the coast and far away from major cities
- All sites appear close to the equator line

Colour Coded Launch Locations

In [12]:
```
# Add marker_cluster to current site_map
site_map.add_child(marker_cluster)

# for each row in spacex_df data frame
# create a Marker object with its coordinate
# and customize the Marker's icon property to indicate if this launch was successed or failed,
# e.g., icon=folium.Icon(color='white', icon_color=row['marker_color']
for index, record in spacex_df.iterrows():
    # TODO: Create and add a Marker cluster to the site map
    coord = [record['Lat'], record['Long']]
    marker = folium.Marker(coord, icon=folium.Icon(color='white', icon_color=record['marker_color']))
    marker_cluster.add_child(marker)

site_map
```

Out[12]:

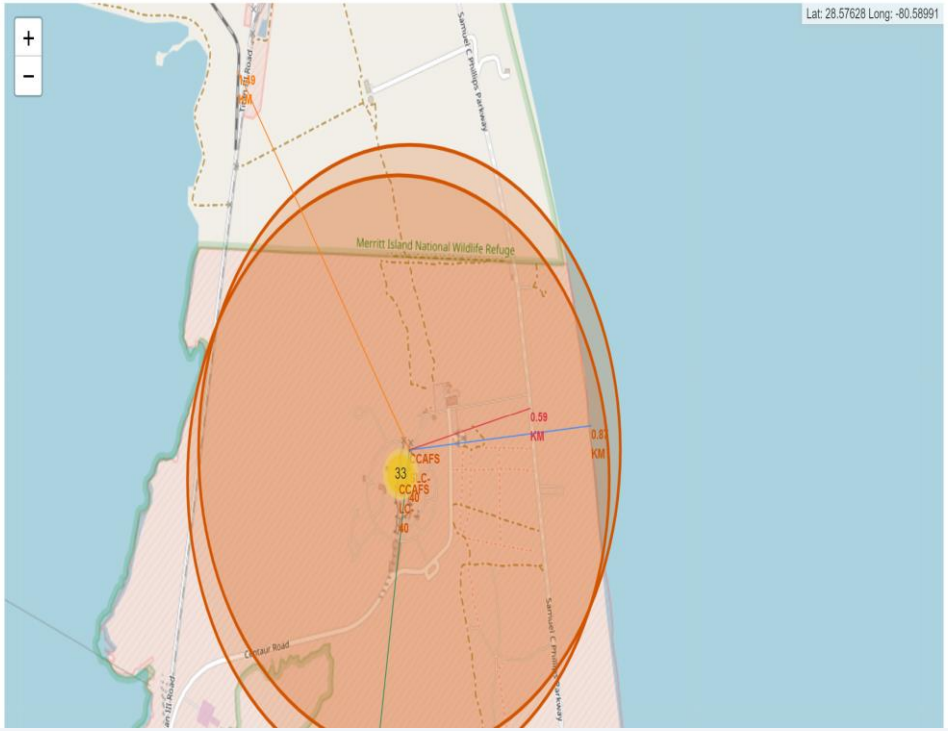


- Here we have markers for each launch location on the map – colour coded to show if the launch was successful (green) or unsuccessful (red)

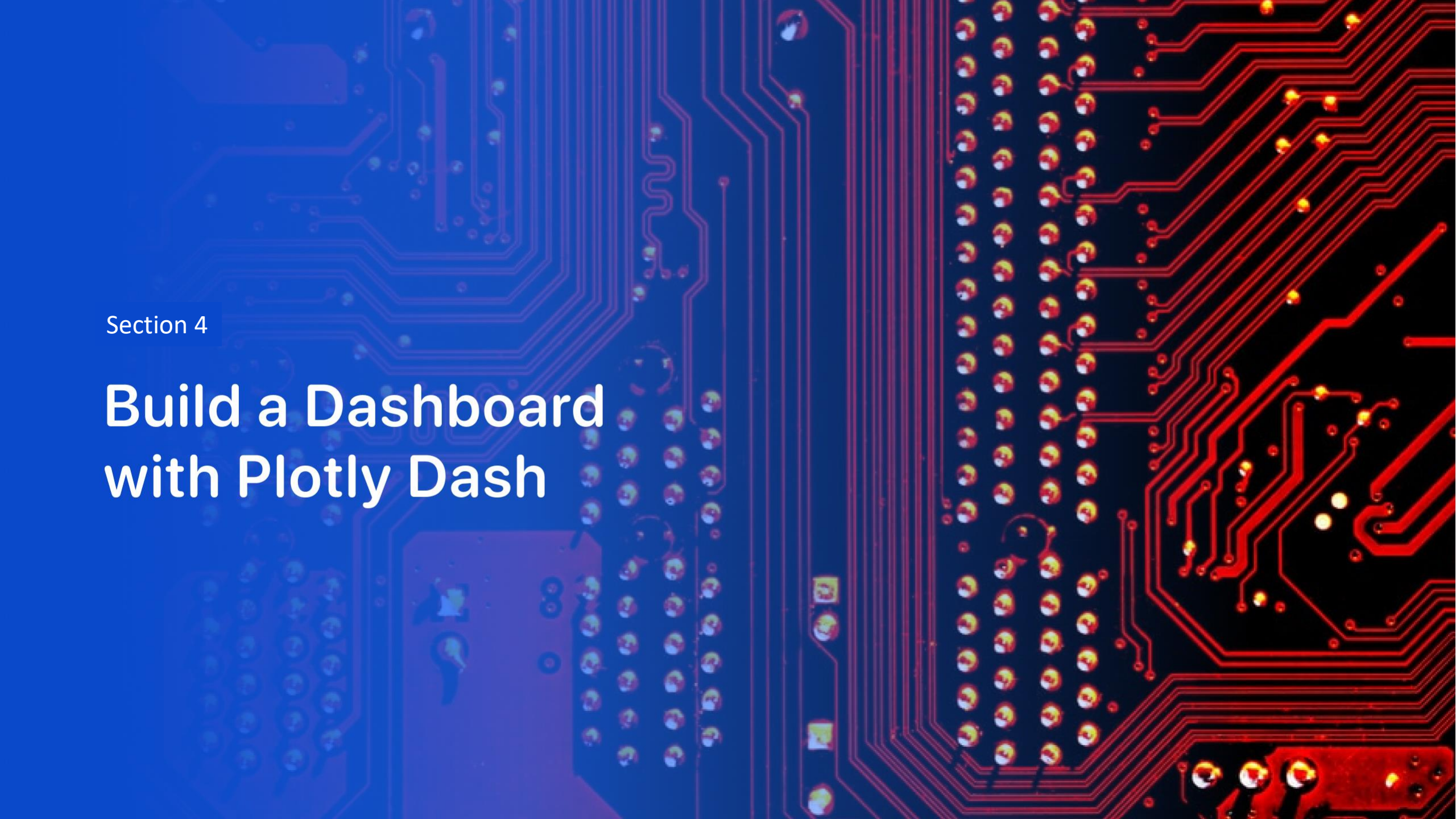
Launch Site CCAFS SLC40 and it's Proximities

```
In [19]: for coordinate,distance,html_color in zip([highway,railway,city],[highway_distance,railway_distance,city_distance],html_colors):
             distance_marker = folium.Marker(coordinate,\
                                             icon=DivIcon(icon_size=(20,20),icon_anchor=(0,0),\
                                                          html='<div style="font-size: 12; color:'+html_color+';"><b>%s</b></div>' % "{:.2f} KM".format(distan
             site_map.add_child(distance_marker)
             line = folium.PolyLine(locations=[coordinate,[launch_site_lat,launch_site_lon]], color=html_color, weight=1)
             site_map.add_child(line)

         site_map
```

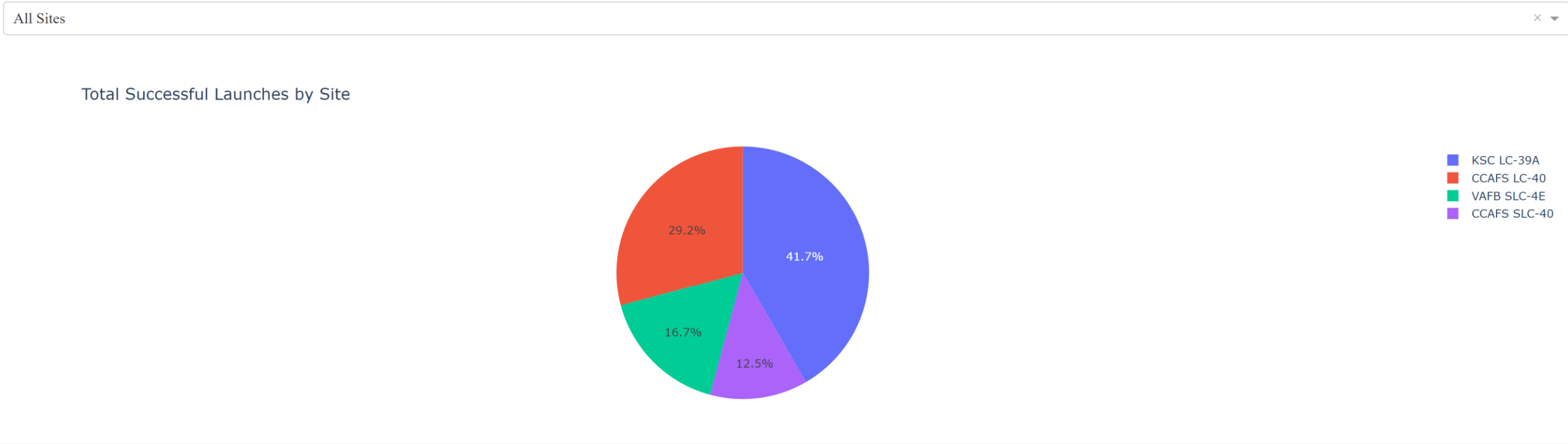
Out[19]:



- Lines on the map show the distance from launch site CCAF SLC40 to it's proximities, for example:
- 0.59km to the nearest highway (Samuel C Philips Highway)
- 0.87km to the coast
- 1.49km to the nearest train station (NASA Railway)
- 18.16km to the nearest city (Cape Canaveral)

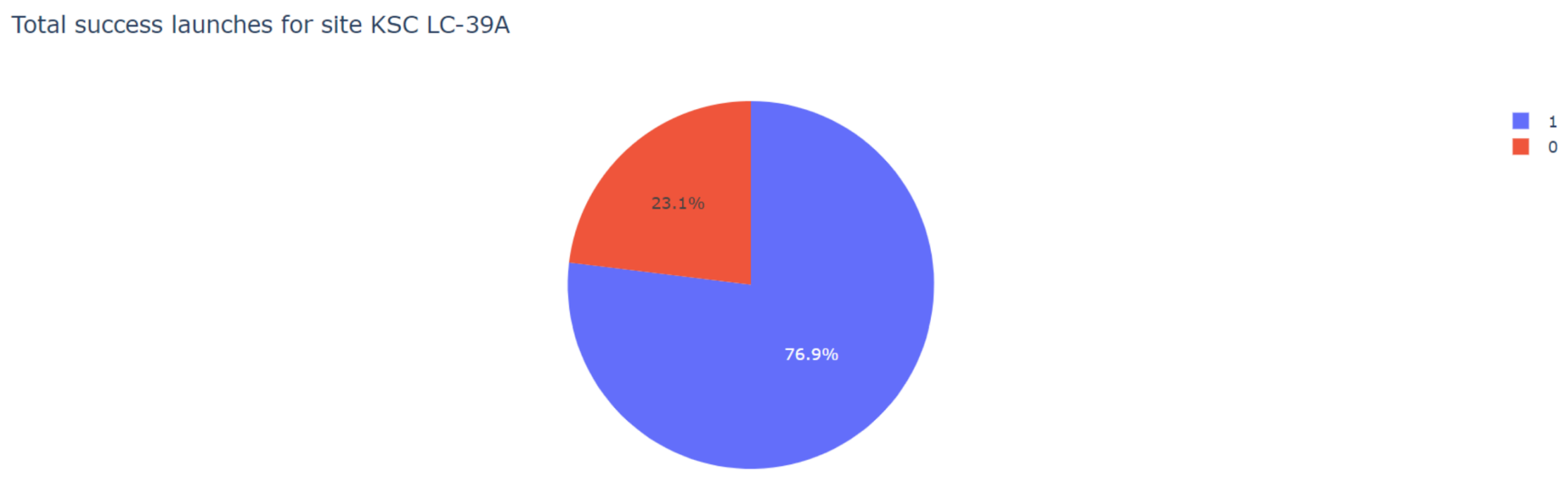

Section 4
Build a Dashboard with Plotly Dash

Total Success Rate by Launch Sites



- This pie chart displays the different sites as a proportion of the total success rate
- We can see that site KSC LC-39A had the greatest success rate of all the sites

Breakdown of Launch Site with Highest Success Rate



- This pie chart displays the success/failure rate for launch site KSC LC-39A
- The previous pie chart showed it's success rate as a proportion of all the successful launches (41.6%), but on closer inspection we can see it's individual success rate is higher than this (76.9%)

Payload vs Launch Outcome for All Sites

- From the scatter plot we can see payloads between 2000 and 5000kg have the highest success rate
- Payloads between 5000 and 10000kg have the lowest success rate
- Booster Version B5 had the highest success rate (100% with 1 launch), but as it only had one launch we could consider booster version PT having the highest success rate overall (as it was tested more than once, so we know it's not an outlier)

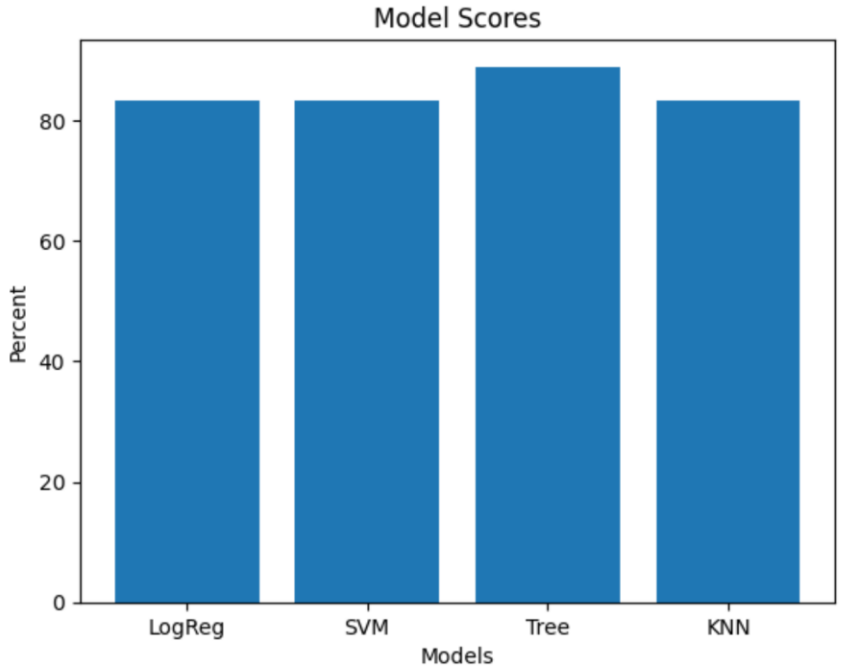
Section 5

Predictive Analysis (Classification)

Classification Accuracy

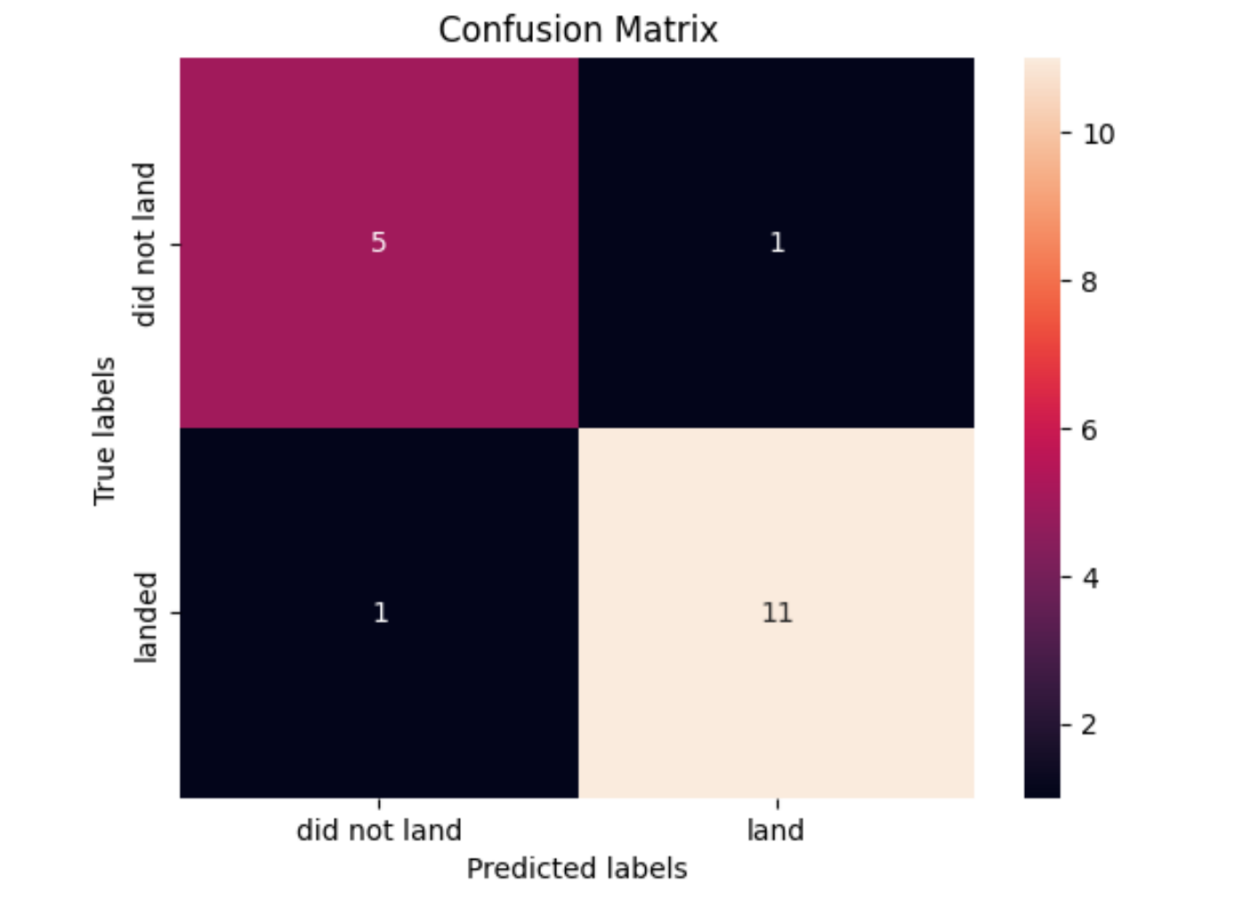
```
In [33]: scores = [(logreg_score*100), (svm_score*100), (tree_score*100), (knn_score*100)]
         models = ['LogReg', 'SVM', 'Tree', 'KNN']
         print(scores)
         print(models)

         plt.bar(models, scores)
         plt.xlabel('Models')
         plt.ylabel('Percent')
         plt.title('Model Scores')
         plt.show()
```

```
[83.33333333333334, 83.33333333333334, 88.88888888888889, 83.33333333333334]
['LogReg', 'SVM', 'Tree', 'KNN']
```



- Using a bar chart to display the accuracy scores for each model
- Comparing them we can see that all tests scored highly, but the Decision Tree test was the highest scoring overall and so the best model for this data set

Confusion Matrix



- In the decision tree confusion matrix we can see that the model only display 1 False Positive and 1 False Negative, all other predictions were accurate (11 True Positives and 5 True Negatives)

Conclusions

- Launches with a low payload mass were more successful that those with a high payload mass
- Orbits ES-L1, GEO, HEO, and SSO have the highest success rate (100%)
- Booster B5 has the highest success rate of all boosters
- Most launch sites are near to the Equator, and all are close to the coast
- Landing success rate increased as years progressed
- Decision Tree Classifier is the best classification algorithm for this dataset

Appendix

- Include any relevant assets like Python code snippets, SQL queries, charts, Notebook outputs, or data sets that you may have created during this project

Thank you!