IBM Developer
SKILLS NETWORK

# Winning Space Race with Data Science

Tamzidul Matin

January 1, 2023

# Outline

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

# Executive Summary

- Summary of methodologies
  - Data Collection through API
  - Data Collection with Web scraping
  - Data Wrangling
  - Exploratory Data Analysis with SQL
  - Exploratory Data Analysis with Data Visualization
  - Interactive Visual Analytics with Folium
  - Machine Learning Prediction
- Summary of all results
  - Exploratory Data Analysis result
  - Interactive analytics in screenshots
  - Predictive analytics results

# 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.

- Problems you want to find answers
  - What factors determine if the rocket will land successfully?
  - The interaction amongst various features that determine the success rate of a successful landing
  - What operating conditions needs to be in place to ensure a successful landing program.

Section 1

# Methodology

# Methodology

## Executive Summary

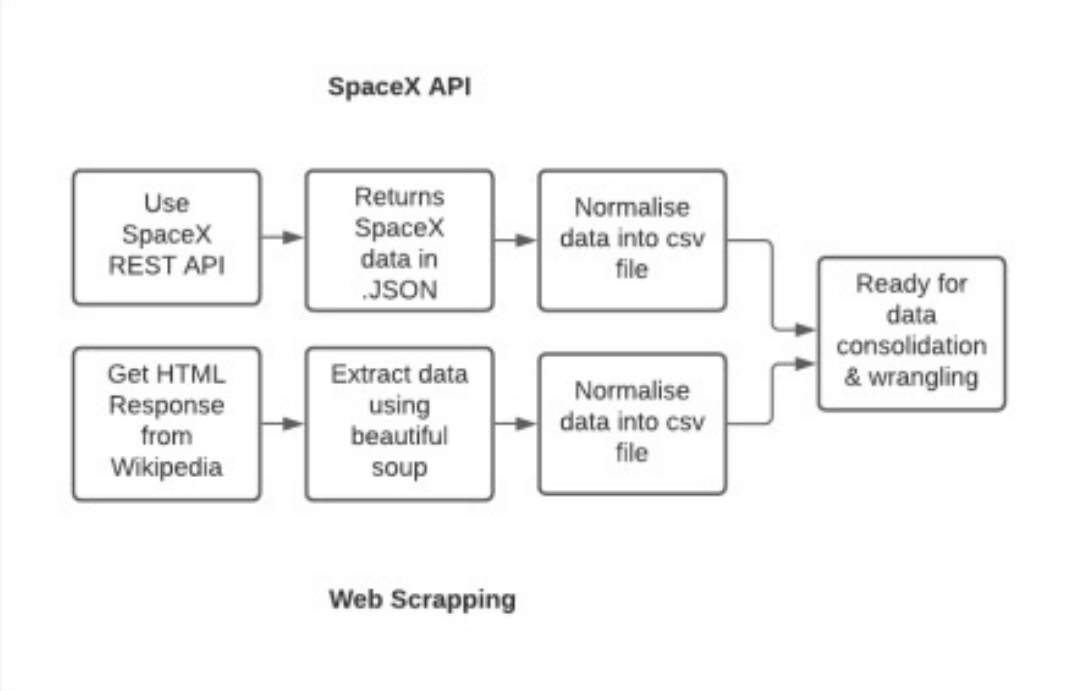
- Data collection methodology:
  - Data was collected using the SapceX API and web scraping form Wikipedia
- Perform data wrangling
  - One hot encoding was applied to categorical feature
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - LR, KNN, SVM, DT models have been built and evaluated for the best classifier

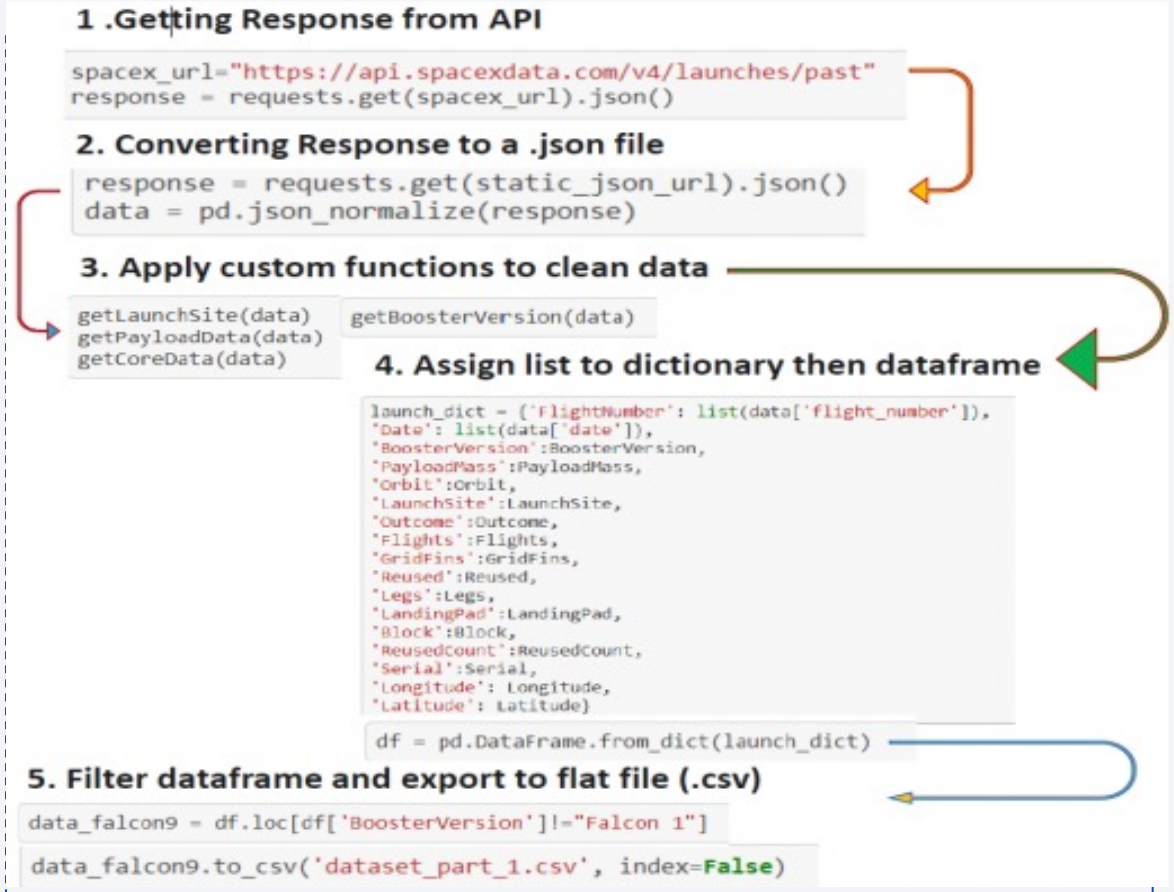
# Data Collection

- The following datasets was collected:
  - Data collection was done using get request to the SpaceX API.
  - Next, we decoded the response content as a Json using .json() function call and turn it into a pandas dataframe using .json_normalize().
  - We then cleaned the data, checked for missing values and fill in missing values where necessary.
  - In addition, we performed web scraping from Wikipedia for Falcon 9 launch records with BeautifulSoup.
  - The objective was to extract the launch records as HTML table, parse the table and convert it to a pandas dataframe for future analysis.

# Data Collection – SpaceX API

- Data collection with SpaceX REST calls

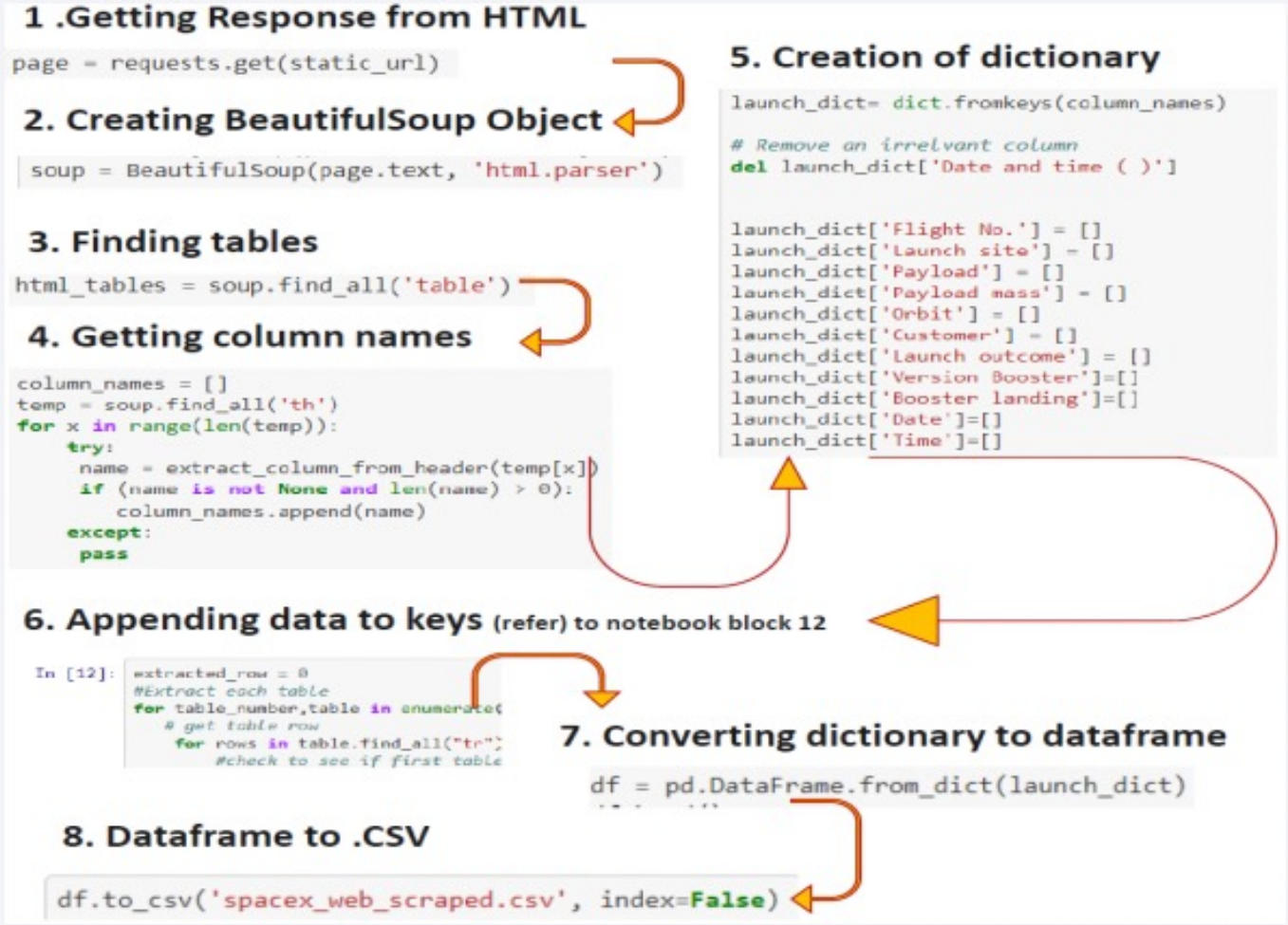



The link to the notebook is https://github.com/tmatin100/Applied-Data-Science-Capstone/blob/main/Lab-1.Collecting%20the%20data-api.ipynb

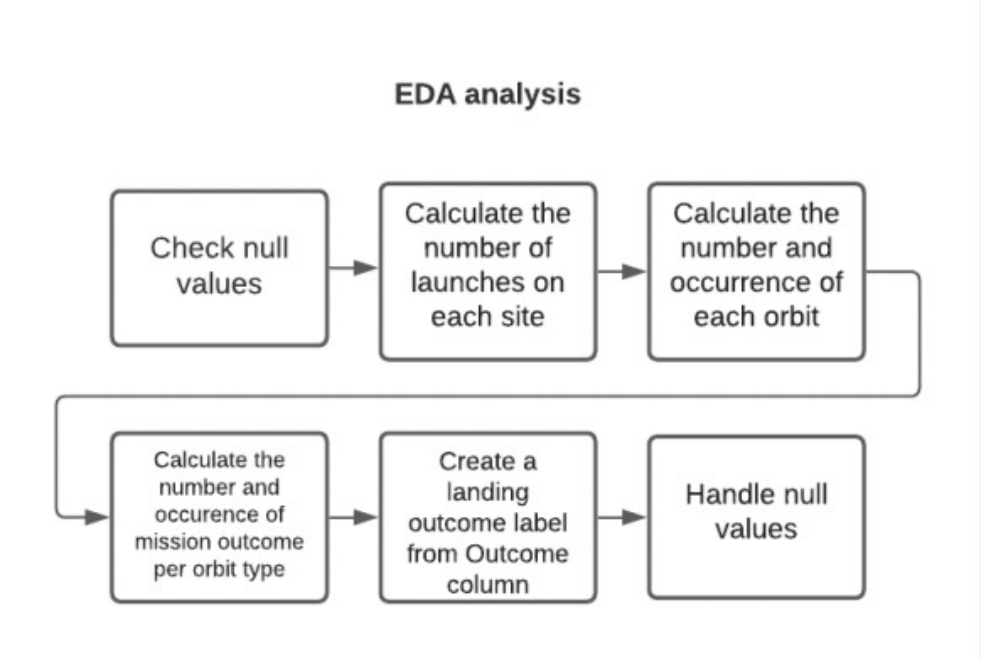
# Data Collection - Scraping

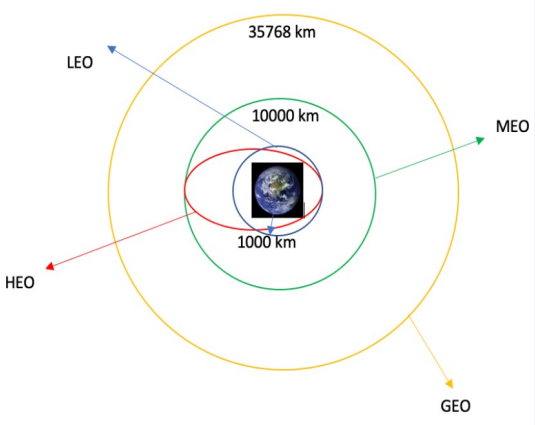
- We applied web scrapping to webscrap Falcon 9 launch records with BeautifulSoup
- We parsed the table and converted it into a pandas dataframe.
- The link to the notebook is
- https://github.com/tmatin100/Applied-Data-Science-Capstone/blob/main/Lab-2.Web%20scraping%20Falcon%209%20and%20Falcon%20Heavy%20Launches%20Records%20from%20Wikipedia.ipynb



# Data Wrangling

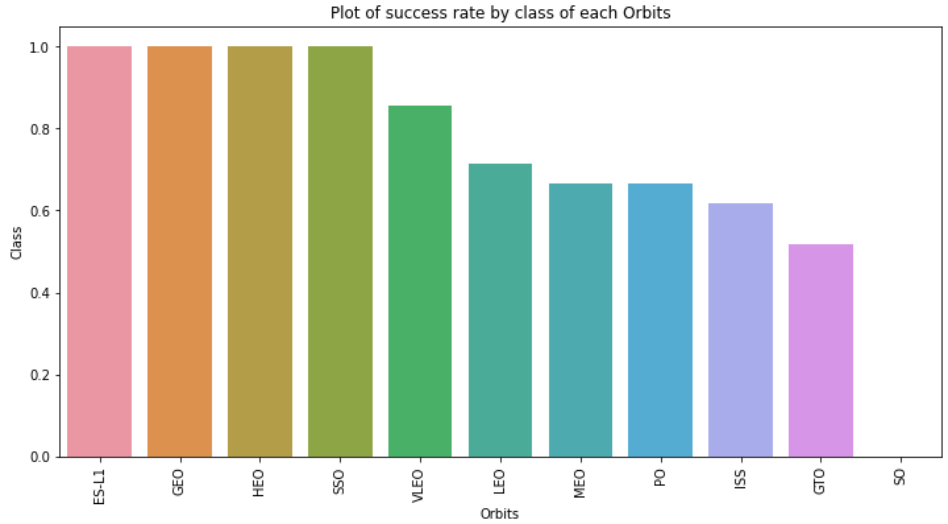
- We performed exploratory data analysis and determined the training labels.
- We calculated the number of launches at each site, and the number and occurrence of each orbits
- We created landing outcome label from outcome column and exported the results to csv.
- The link to the notebook is https://github.com/tmatin100/Applied-Data-Science-Capstone/blob/main/Lab-3.Data%20Wrangling.ipynb

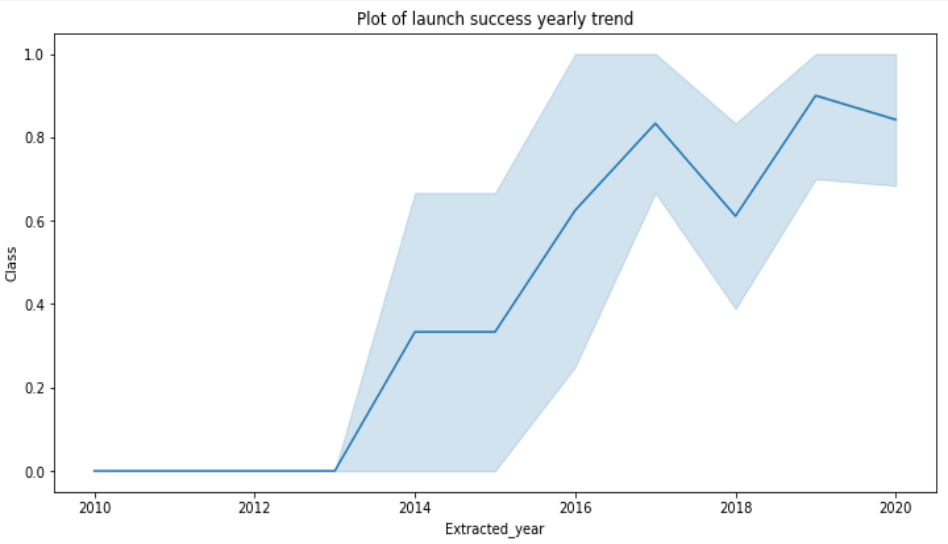


# EDA with Data Visualization

- 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.
- The link to the notebook is https://github.com/chuksoo/IBM-Data-Science-Capstone-SpaceX/blob/main/EDA%20with%20Data%20Visualization.ipynb



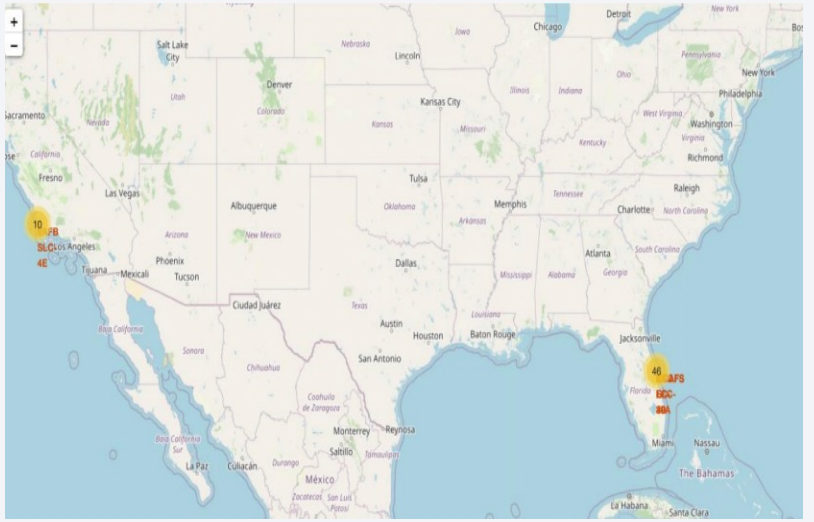



# EDA with SQL

- We loaded the SpaceX dataset into a PostgreSQL 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:
  - The names of unique launch sites in the space mission.
  - The total payload mass carried by boosters launched by NASA (CRS)
  - The average payload mass carried by booster version F9 v1.1
  - The total number of successful and failure mission outcomes
  - The failed landing outcomes in drone ship, their booster version and launch site names.
- The link to the notebook is https://github.com/tmatin100/Applied-Data-Science-Capstone/blob/main/Lab-4.EDA%20with%20SQL.ipynb

# Build an Interactive Map with Folium

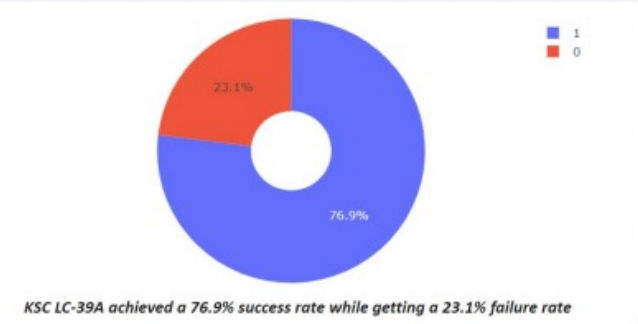
- We marked all launch sites, and added map objects such as markers, circles, lines to mark the success or failure of launches for each site on the folium map.
- We assigned the feature launch outcomes (failure or success) to class 0 and 1.i.e., 0 for failure, and 1 for success.
- Using the color-labeled marker clusters, we identified which launch sites have relatively high success rate.
- We calculated the distances between a launch site to its proximities. We answered some question for instance:
  - Are launch sites near railways, highways and coastlines.
  - Do launch sites keep certain distance away from cities.

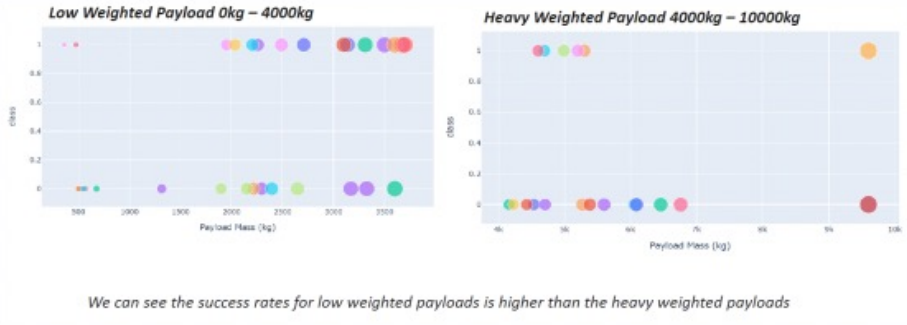



The link to the notebook is https://github.com/tmatin100/Applied-Data-Science-Capstone/blob/main/Lab-6.Launch%20Sites%20Locations%20Analysis%20with%20Folium.ipynb

# Build a Dashboard with Plotly Dash

- We built an interactive dashboard with Plotly dash
- 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.
- The link to the notebook https://github.com/tmatin100/Applied-Data-Science-Capstone/blob/main/Lab%206.a-plotly-dash-app.py

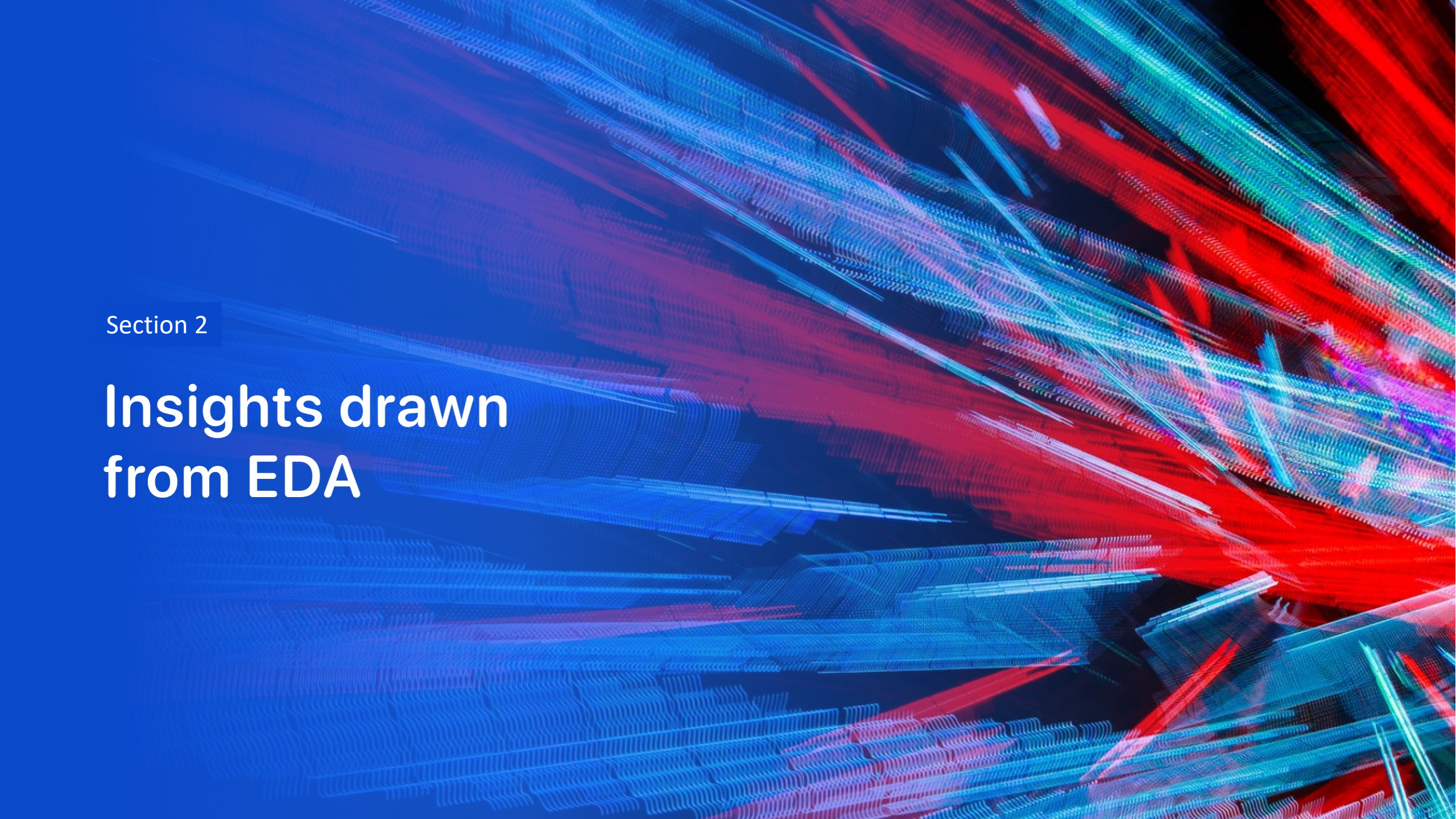



# Predictive Analysis (Classification)

- 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 model, improved the model using feature engineering and algorithm tuning.
- We found the best performing classification model.
- The link to the notebook is https://github.com/tmatin100/Applied-Data-Science-Capstone/blob/main/Lab-7.Machine%20Learning%20Prediction.ipynb

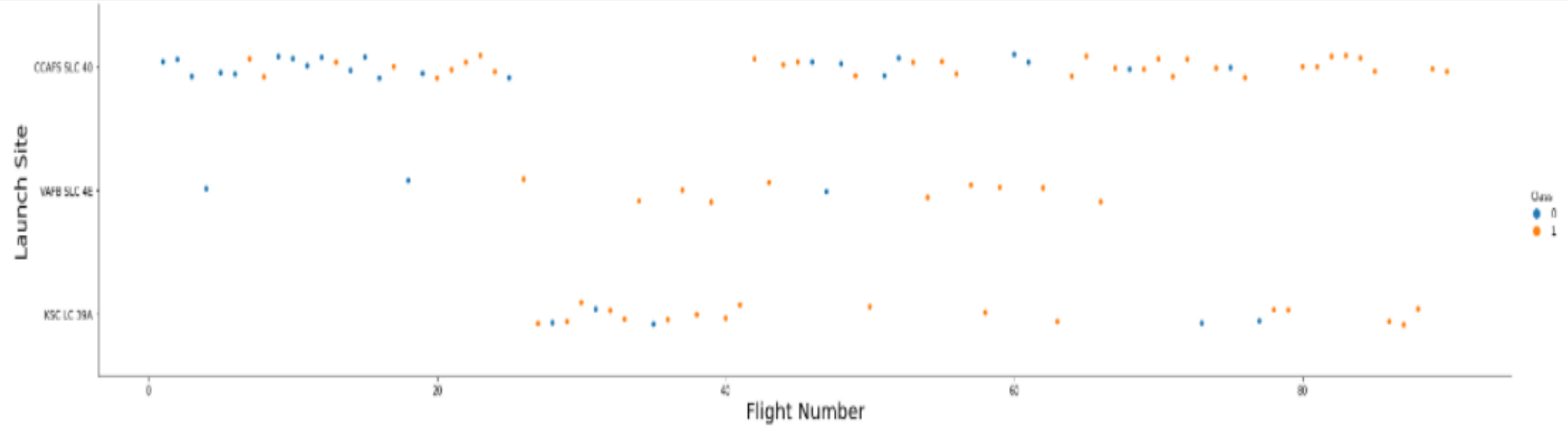
# Results

- The larger the flight amount at a launch site, the greater the success rate at a launch site
- Low weighted payloads perform better than the heavier payloads.
- Launch success rate started to increase in 2013 till 2020.
- KSC LC 39A had the most successful launches from all the sites.
- Orbits ES-L1, GEO, HEO, SSO, VLEO had the most success rate.
- The Decision tree classifier is the best machine learning algorithm for this task.


Section 2
Insights drawn from EDA

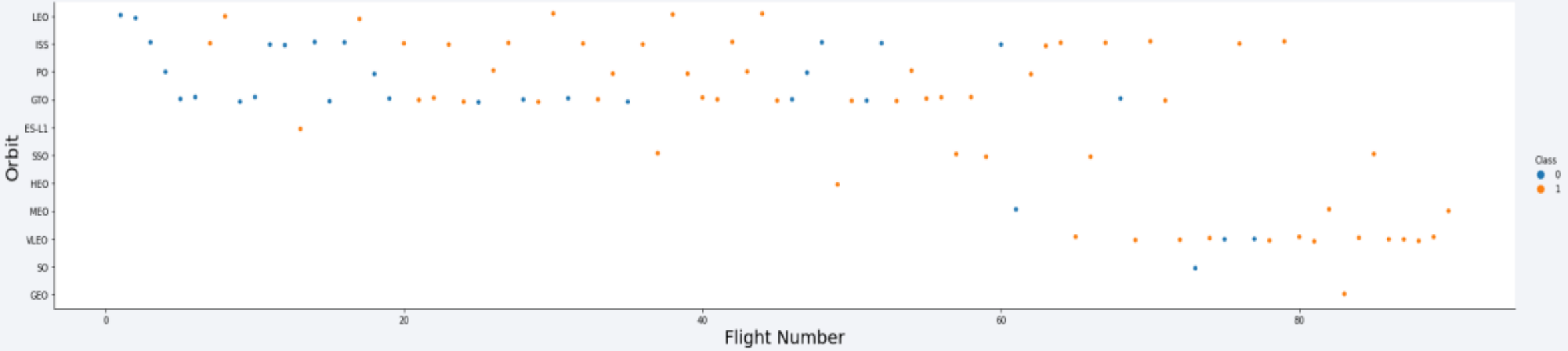
# Flight Number vs. Launch Site

- From the plot it can be seen that as the number of flights increases the success rate increases. It can also be observed that launch site CCAFS SLC 40 has the highest number of failures



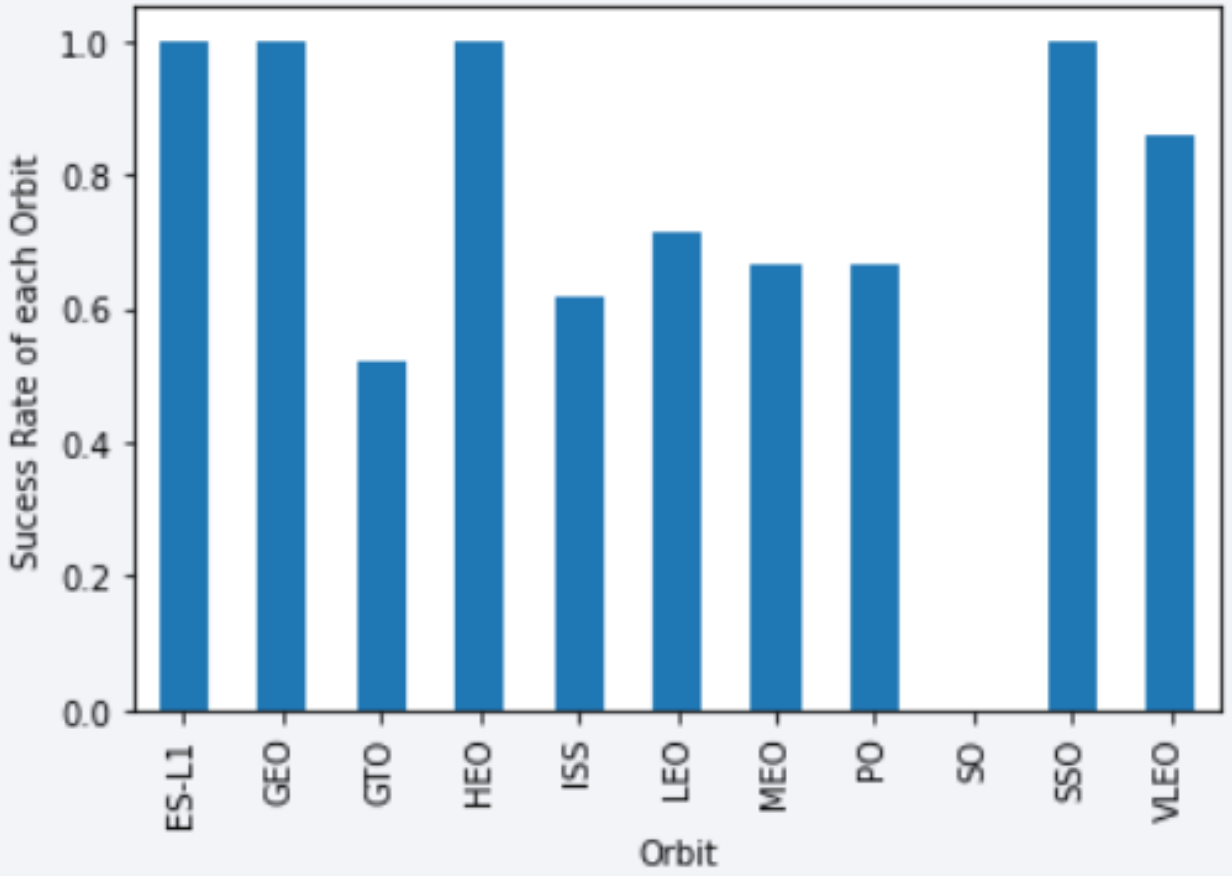
# Payload vs. Launch Site

- From the 3 launch sites present in the graph, it can be observed that the pay load mass mostly ranges from 0 to 8000 kg. There are only a few cases where the payload exceeds 8000 kg



# Success Rate vs. Orbit Type

- The success rate for each orbit can be observed from the bar chart. It can be observed that four orbits ( ES-L1, GEO, HEO and SSO) have a success rate of 1 while SO has a success rate of 0. The other orbits success rate are ranged



# Flight Number vs. Orbit Type

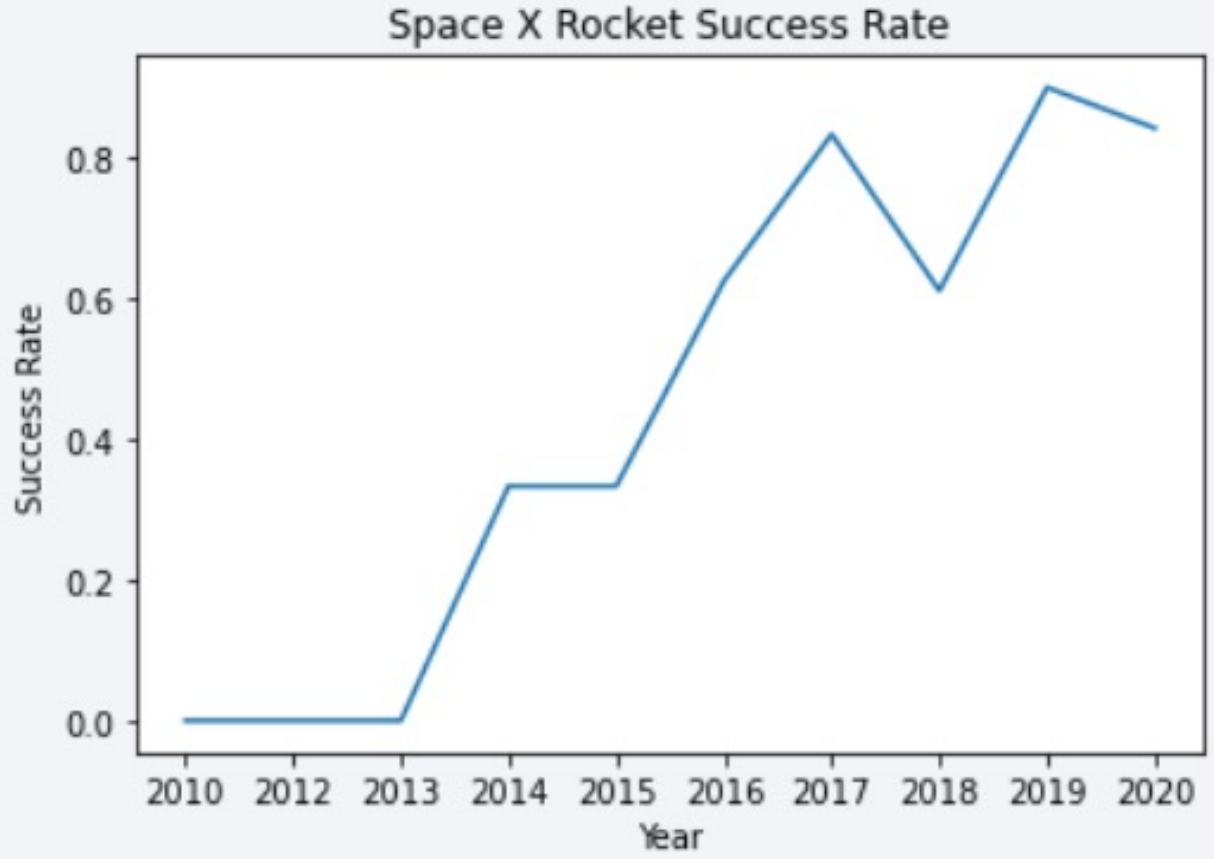
- From the scatter plot, it can be observed that as the number of flights increases the success rate also increases. Hence, we can say that the past

# Payload vs. Orbit Type

- It can be seen that for some orbit, a heavier pay load mass helps to improve the success rate, for example the LEO orbit. However, for some orbit such as GTO, increasing the pay load mass does not seem to improve the success rate.

# Launch Success Yearly Trend

- Launch success rate has increased significantly since 2013 and has stabilized since 2019, potentially due to advance in technology and lessons learned



# All Launch Site Names

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

```
%sql SELECT DISTINCT LAUNCH_SITE FROM SPACEXTBL;
```

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

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

# Launch Site Names Begin with 'CCA'

- We used the query bellow to display 5 records where launch sites begin with `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 | Mission_Outcome | Landing _Outcome |
|---|---|---|---|---|---|---|---|---|---|
| 04-06-2010 | 18:45:00 | F9 v1.0 B0003 | CCAFS LC-40 | Dragon Spacecraft Qualification Unit | 0 | LEO | SpaceX | Success | Failure (parachute) |
| 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 | Success | Failure (parachute) |
| 22-05-2012 | 07:44:00 | F9 v1.0 B0005 | CCAFS LC-40 | Dragon demo flight C2 | 525 | LEO (ISS) | NASA (COTS) | Success | No attempt |
| 08-10-2012 | 00:35:00 | F9 v1.0 B0006 | CCAFS LC-40 | SpaceX CRS-1 | 500 | LEO (ISS) | NASA (CRS) | Success | No attempt |
| 01-03-2013 | 15:10:00 | F9 v1.0 B0007 | CCAFS LC-40 | SpaceX CRS-2 | 677 | LEO (ISS) | NASA (CRS) | Success | No attempt |

# Total Payload Mass

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

```
%sql SELECT SUM(payload_mass__kg_) FROM SPACEXTBL WHERE customer = 'NASA (CRS)';
```

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

| SUM(payload_mass__kg_) |
|---|
| 45596 |

# Average Payload Mass by F9 v1.1

- We calculated the average payload mass carried by booster version F9 v1.1 as 2534.66

```
%sql SELECT AVG("PAYLOAD_MASS__KG_") FROM SPACEXTBL WHERE "BOOSTER_VERSION" LIKE 'F%9 v1.1%';
```

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

| AVG("PAYLOAD_MASS__KG_") |
| --- |
| 2534.6666666666665 |

# First Successful Ground Landing Date

- We observed that the dates of the first successful landing outcome on ground pad was January 3rd, 2013.

```
%sql SELECT MIN(DATE) FROM SPACEXTBL WHERE mission_outcome = 'Success';
```

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

| MIN(DATE) |
| --- |
| 01-03-2013 |

# Successful Drone Ship Landing with Payload between 4000 and 6000

- 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.

```
%sql SELECT booster_version FROM SPACEXTBL 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 |

# Total Number of Successful and Failure Mission Outcomes

- We used wildcard like '%' to filter for **WHERE** Mission Outcome was a success or a failure.

```
%sql SELECT (SELECT COUNT("MISSION_OUTCOME") FROM SPACEXTBL WHERE "MISSION_OUTCOME" LIKE '%Success%') AS SUCESS,\
(SELECT COUNT("MISSION_OUTCOME") FROM SPACEXTBL WHERE "MISSION_OUTCOME" LIKE '%FAILURE%') AS FAILURE
```

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

| SUCESS | FAILURE |
|---|---|
| 100 | 1 |

# Boosters Carried Maximum Payload

- These are the names of the booster which have carried the maximum payload mass

```
%sql SELECT "Booster_Version" FROM SPACEXTBL ORDER BY payload_mass__kg_ LIMIT 10
```

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

| Booster_Version |
|---|
| F9 v1.0 B0003 |
| F9 v1.0 B0004 |
| F9 B4 B1045.1 |
| F9 FT B1038.1 |
| F9 v1.0 B0006 |
| F9 v1.1 B1003 |
| F9 v1.0 B0005 |
| F9 v1.1 B1017 |
| F9 v1.1 B1013 |
| F9 v1.0 B0007 |

# 2015 Launch Records

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

```
%sql SELECT substr("DATE", 4, 2) AS MONTH, "BOOSTER_VERSION", "LAUNCH_SITE" FROM SPACEXTBL\
WHERE "LANDING _OUTCOME" = 'Failure (drone ship)' and substr("DATE",7,4) = '2015'
```

Output:

| MONTH | Booster_Version | Launch_Site |
|---|---|---|
| 01 | F9 v1.1 B1012 | CCAFS LC-40 |
| 04 | F9 v1.1 B1015 | CCAFS LC-40 |

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

- We selected Landing outcomes and the **COUNT** of landing outcomes from the data and used the **WHERE** clause to filter for landing outcomes **BETWEEN** 04-06-2010 to 20-03-2017
- 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.

SQL Query:

```
%sql SELECT "LANDING _OUTCOME", COUNT("LANDING _OUTCOME") FROM SPACEXTBL\
WHERE "DATE" >= '04-06-2010' and "DATE" <= '20-03-2017' and "LANDING _OUTCOME" LIKE '%Success%'\
GROUP BY "LANDING _OUTCOME" \
ORDER BY COUNT("LANDING _OUTCOME") DESC ;
```

Output:

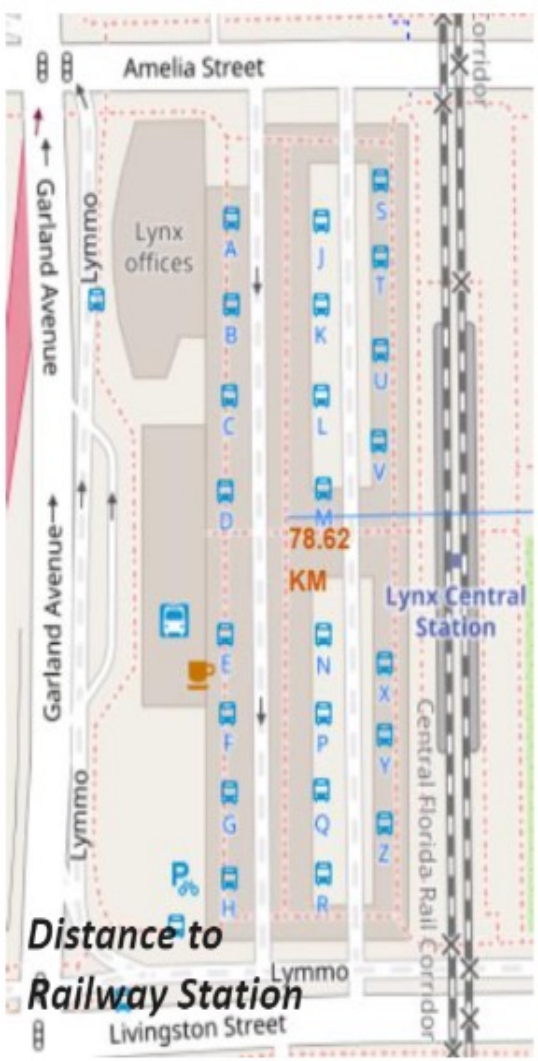
| Landing _Outcome | COUNT("LANDING _OUTCOME") |
|---|---|
| Success | 20 |
| Success (drone ship) | 8 |
| Success (ground pad) | 6 |


Section 3
Launch Sites
Proximities Analysis

# All launch sites global map markers

# Markers showing launch sites with color labels

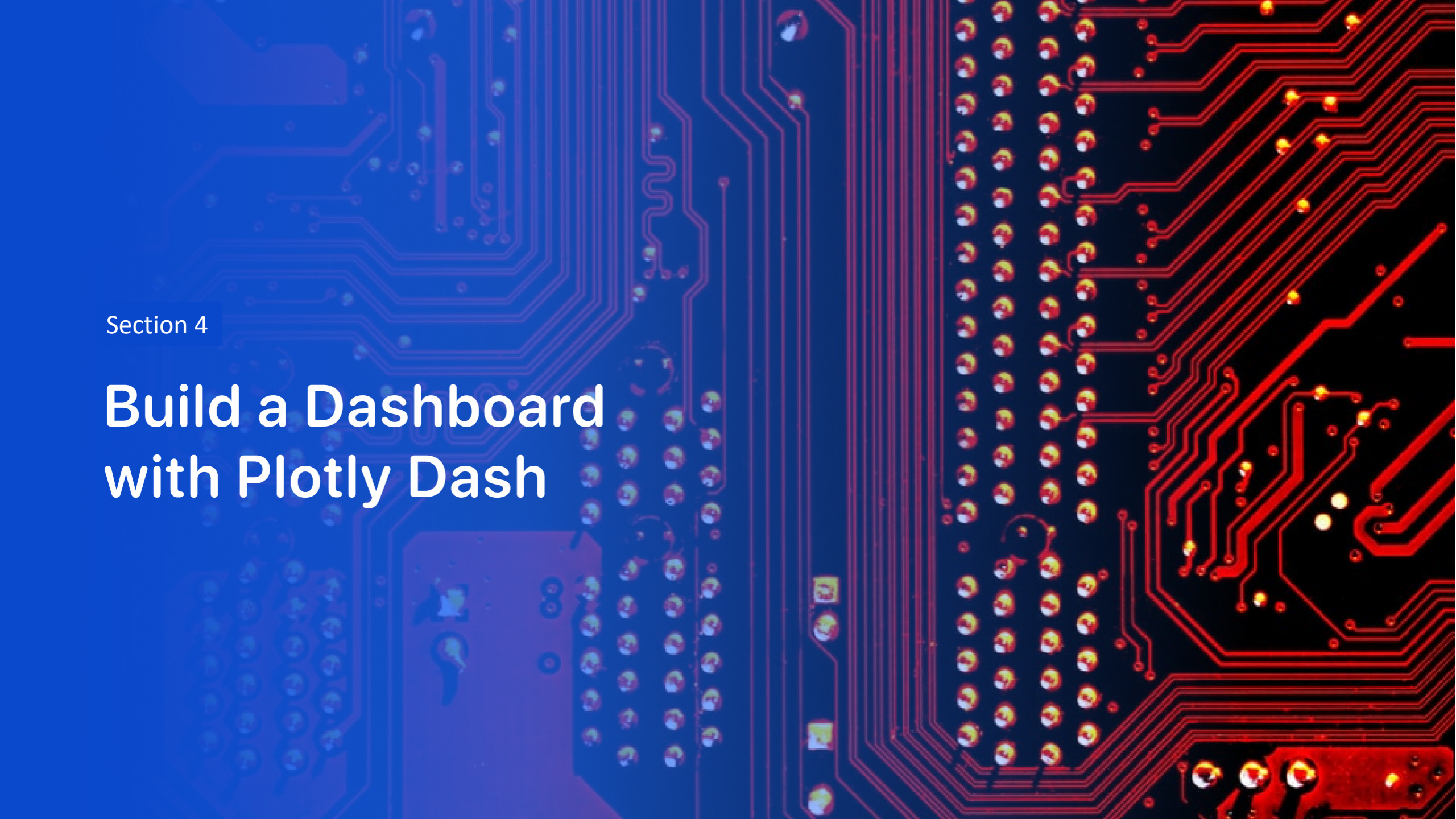
# Launch Site distance to landmarks



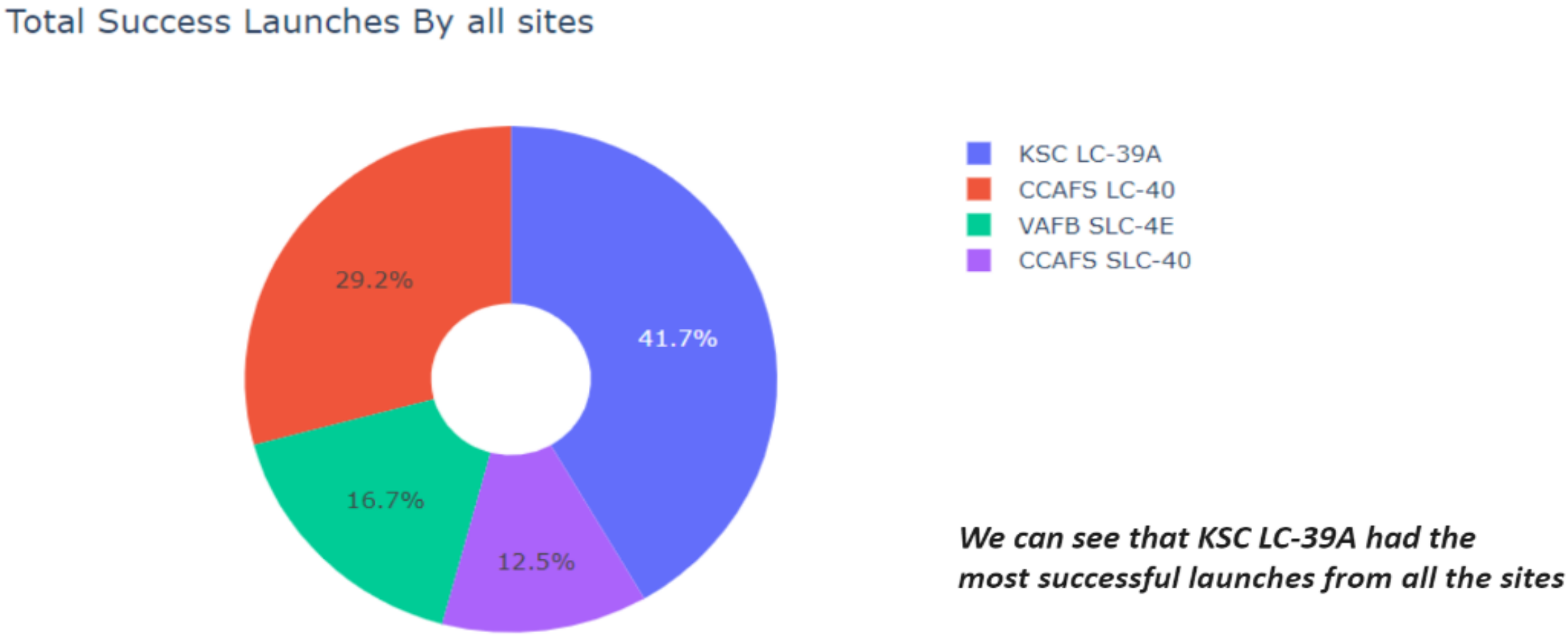





- Are launch sites in close proximity to railways? No
- Are launch sites in close proximity to highways? No
- Are launch sites in close proximity to coastline? Yes
- Do launch sites keep certain distance away from cities? Yes

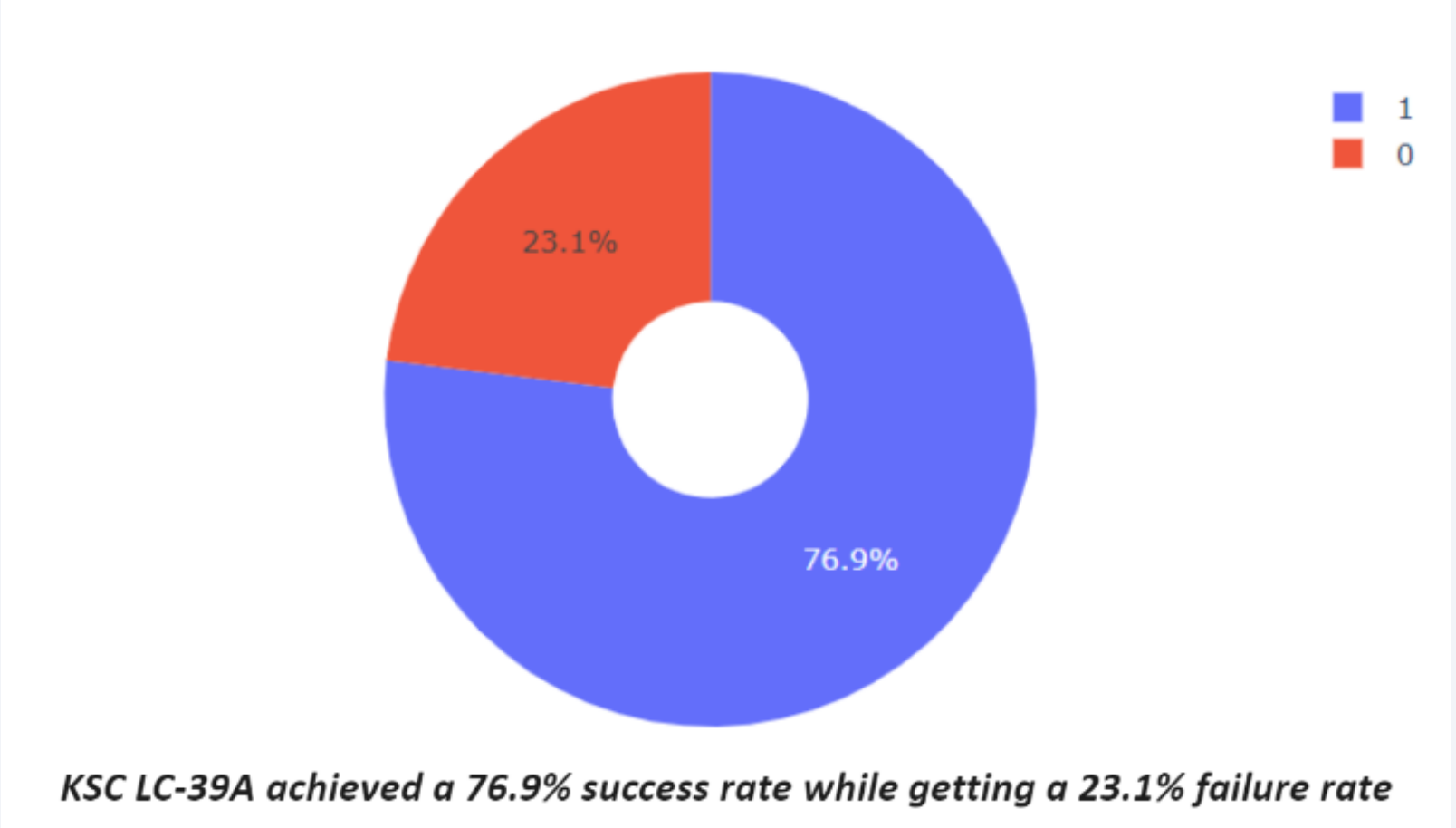

Section 4
Build a Dashboard with Plotly Dash

# Pie chart showing the success percentage achieved by each launch site



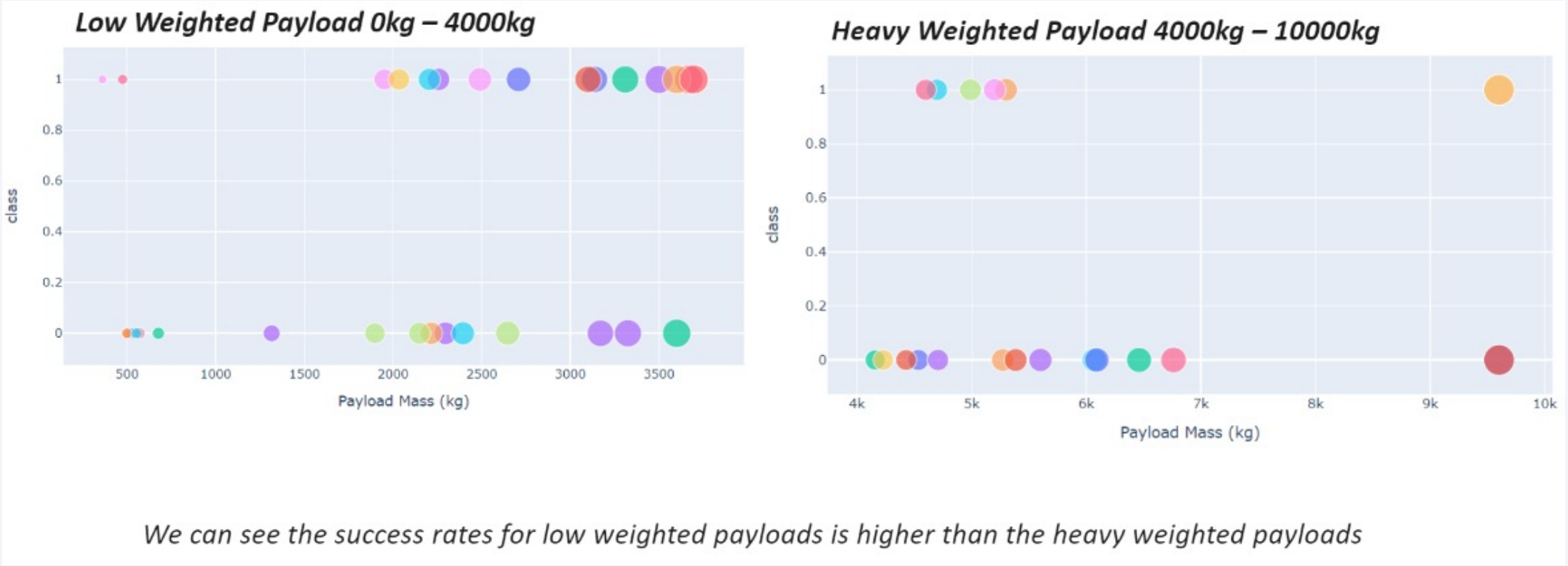

***We can see that KSC LC-39A had the most successful launches from all the sites***

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




*KSC LC-39A achieved a 76.9% success rate while getting a 23.1% failure rate*

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




*We can see the success rates for low weighted payloads is higher than the heavy weighted payloads*

Section 5

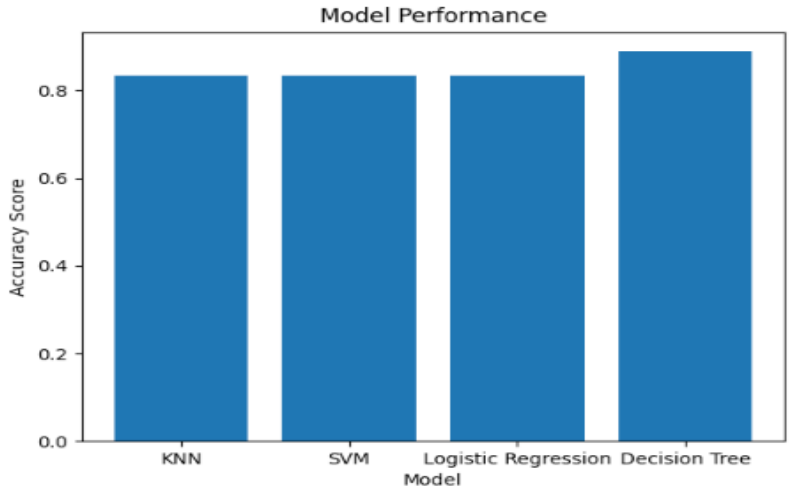
# Predictive Analysis (Classification)

# Classification Accuracy

- The decision tree classifier is the model with the highest classification accuracy with a 88.8%

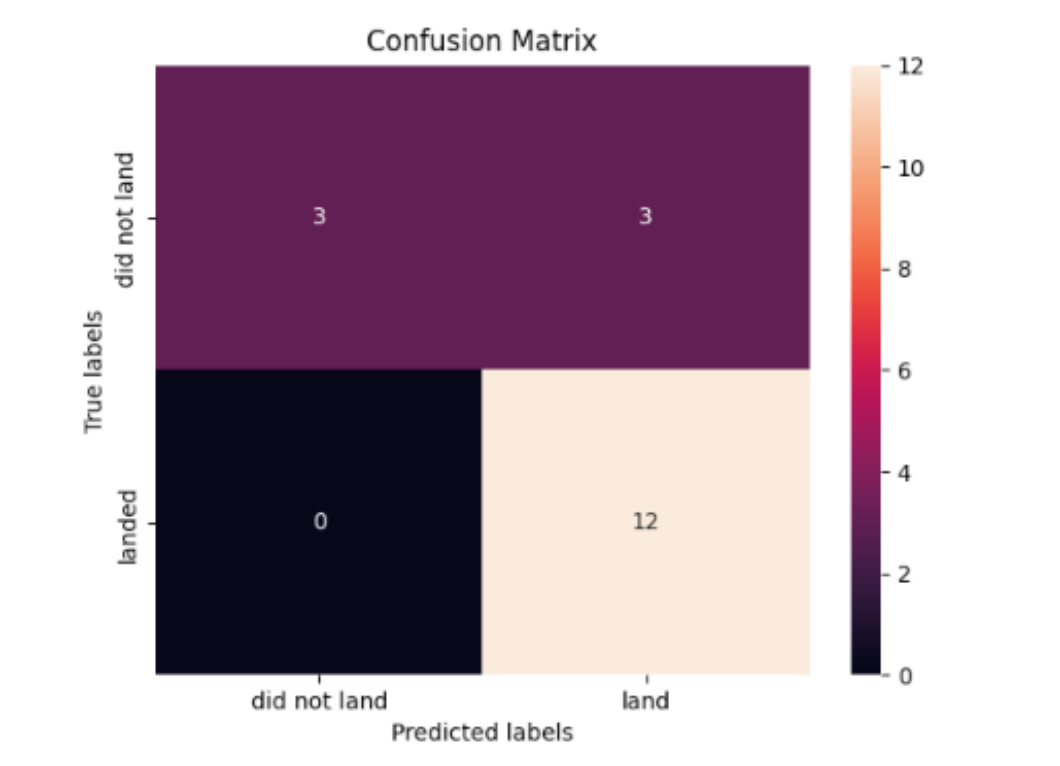
```
# To determine which model has the best accuracy score, we can compare the scores of the different models and keep track of the model wit

# Create lists of accuracy scores and predictor models
predictors = ["KNN", "SVM", "Logistic Regression", "Decision Tree"]
scores = [knn_cv.score(X_test, Y_test), svm_cv.score(X_test, Y_test), logreg_cv.score(X_test, Y_test), tree_cv.score(X_test, Y_test)]

# we can use the enumerate function to iterate through the scores list and print the index of each score along with the corresponding mod
for i, score in enumerate(scores):
    print(f"{predictors[i]}: {score:.4f}")



# Create bar chart
plt.bar(labels, scores)
plt.xlabel("Model")
plt.ylabel("Accuracy Score")
plt.title("Model Performance")

# Show plot
plt.show()
```

```
KNN: 0.8333
SVM: 0.8333
Logistic Regression: 0.8333
Decision Tree: 0.8889
```



# Confusion Matrix

- The confusion matrix for the decision tree classifier shows that the classifier can distinguish between the different classes. The major problem is the false positives .i.e., unsuccessful landing marked as successful landing by the classifier.



# Conclusions

We can conclude that:

- The larger the flight amount at a launch site, the greater the success rate at a launch site
- Low weighted payloads perform better than the heavier payloads.
- Launch success rate started to increase in 2013 till 2020.
- KSC LC 39A had the most successful launches from all the sites.
- Orbits ES-L1, GEO, HEO, SSO, VLEO had the most success rate.
- The Decision tree classifier is the best machine learning algorithm for this task.

# Appendix

- https://github.com/tmatin100/Applied-Data-Science-Capstone

# Thank you!