IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Roman MURZAC
30.12.2023

Outline

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

Executive Summary

- Collected data from public SpaceX API and from SpaceX Wikipedia page.
- Created labels column ‘class’ which classifies successful landings.
- Explored data using SQL, visualization, folium maps, and dashboards.
- Gathered relevant columns to be used as features.
- Changed all categorical variables to binary using one hot encoding.
- Standardized data and used GridSearchCV to find best parameters for machine learning models.
- Visualize accuracy score of all models.
- Four machine learning models were produced: Logistic Regression, Support Vector Machine, Decision Tree Classifier, and K Nearest Neighbors. Best method is Tree Classifier with accuracy rate of about 94.44%. All models over predicted successful landings.

Introduction

- Background:
- Commercial Space Age is Here
- Space X has best pricing ($62 million vs. $165 million USD)
- Largely due to ability to recover part of rocket (Stage 1)
- Space Y wants to compete with Space X

- Problem:
- Space Y tasks us to train a machine learning model to predict successful Stage 1 recovery

Section 1

Methodology

Methodology

Executive Summary

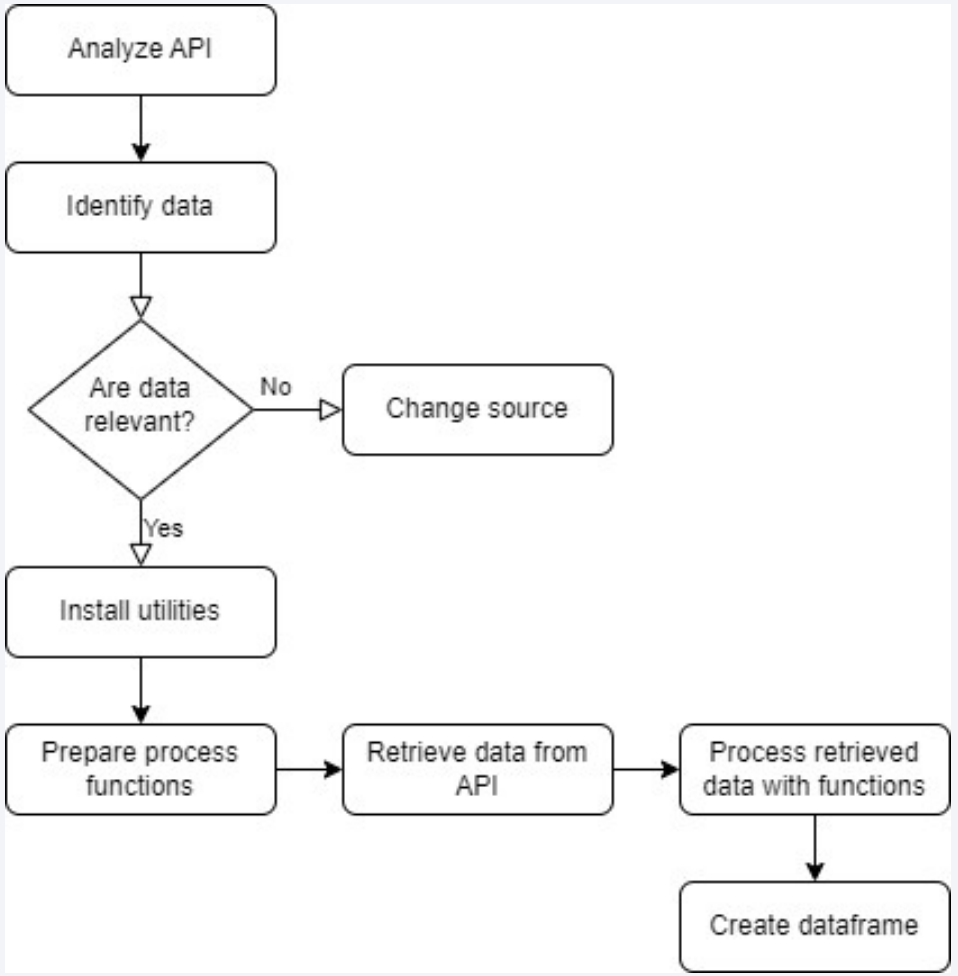
- Data collection methodology:
  - Combined data from SpaceX public API and SpaceX Wikipedia page
- Perform data wrangling
  - Classifying true landings as successful and unsuccessful otherwise
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Tuned models using GridSearchCV

Data Collection

- Data collection process involved a combination of API requests from Space X public API and web scraping data from a table in Space X's Wikipedia entry.
- The next slide will show the flowchart of data collection from API and the one after will show the flowchart of data collection from webscraping.
- Space X API Data Columns:
- FlightNumber, Date, BoosterVersion, PayloadMass, Orbit, LaunchSite, Outcome, Flights, GridFins,
- Reused, Legs, LandingPad, Block, ReusedCount, Serial, Longitude, Latitude
- Wikipedia Webscrape Data Columns:
- Flight No., Launch site, Payload, PayloadMass, Orbit, Customer, Launch outcome, Version Booster, Booster landing, Date, Time

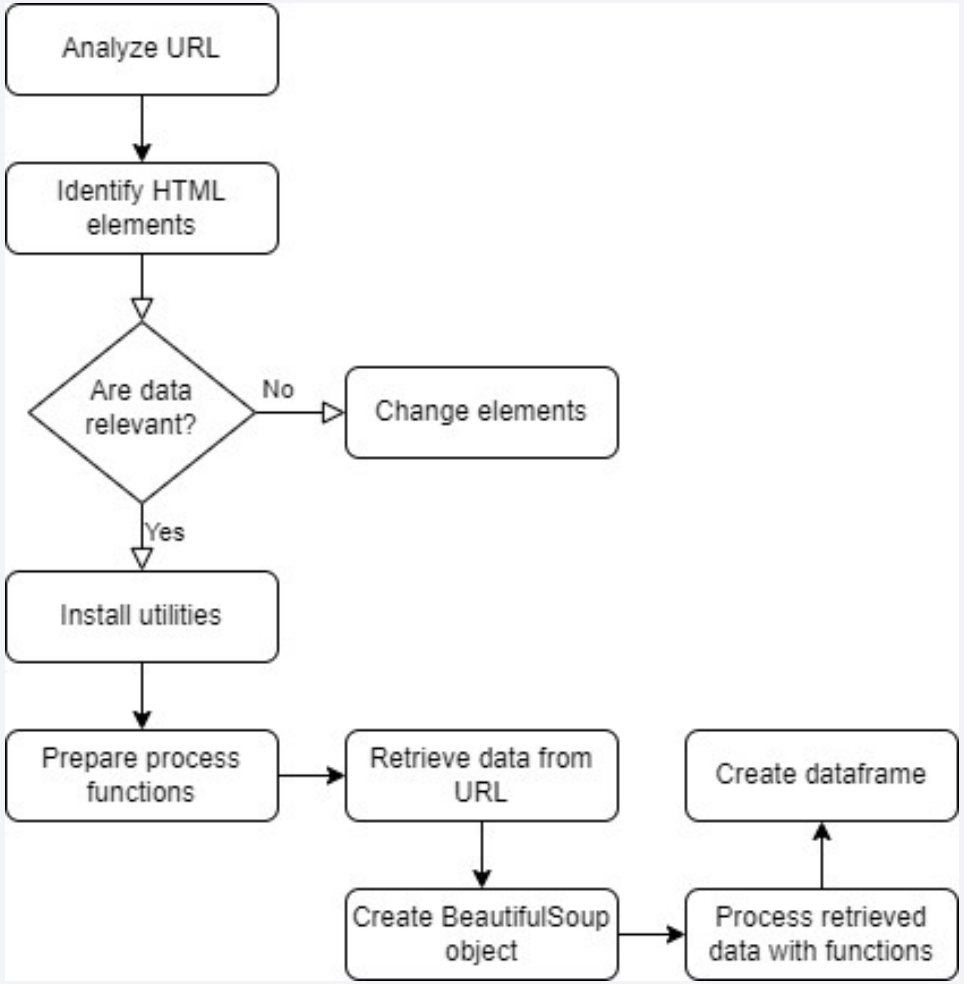
Data Collection – SpaceX API

- SpaceX data was retrieved using REST API calls.
- Used Endpoint is: https://api.spacexdata.com/v4/.
- Were created auxiliary functions for raw data processing and imported necessary packages.
- Retrieved content in JSON format was normalized and processed with auxiliary functions.
- Processed content was transposed in a Dataframe.
- GitHub repository URL of the SpaceX API calls notebook: https://github.com/romanmurzac/ibm-ds-certification/blob/main/Final_Exam/jupyter-labs-spacex-data-collection-api.ipynb



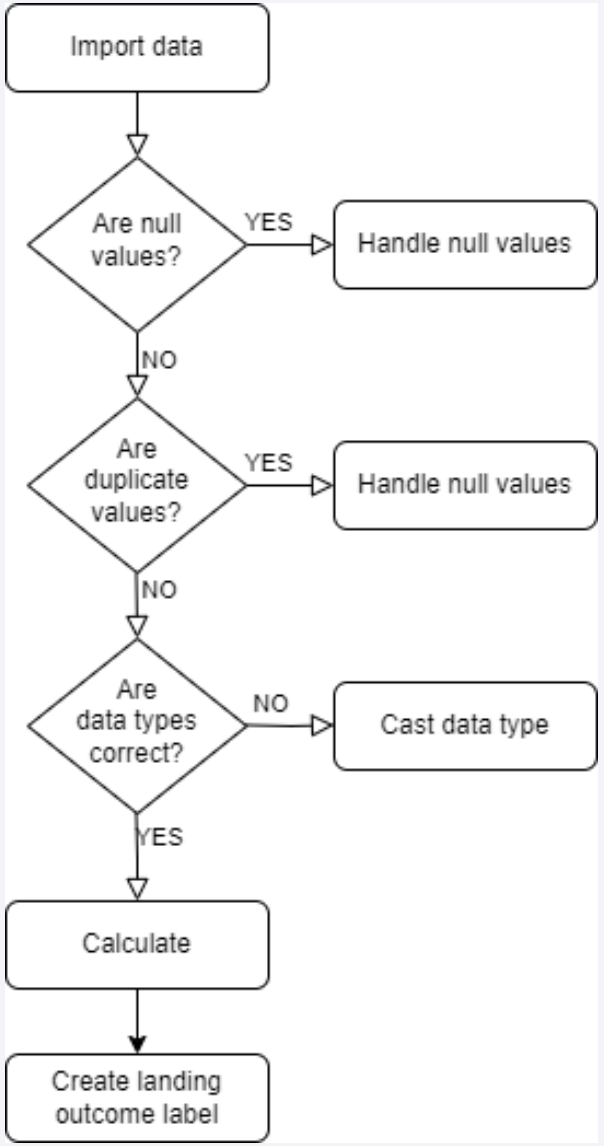
Data Collection – Scraping

- SpaceX data was retrieved using Wikipedia data.
- Used page is: https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches
- Were created auxiliary functions for raw data processing and imported necessary packages.
- Retrieved content in HTML format from provided URL.
- Processed content with Beautiful Soup object, retrieved necessary elements and transposed in a Dataframe.
- GitHub repository URL of the web scraping notebook: https://github.com/romanmurzac/ibm-ds-certification/blob/main/Final_Exam/jupyter-labs-webscraping.ipynb



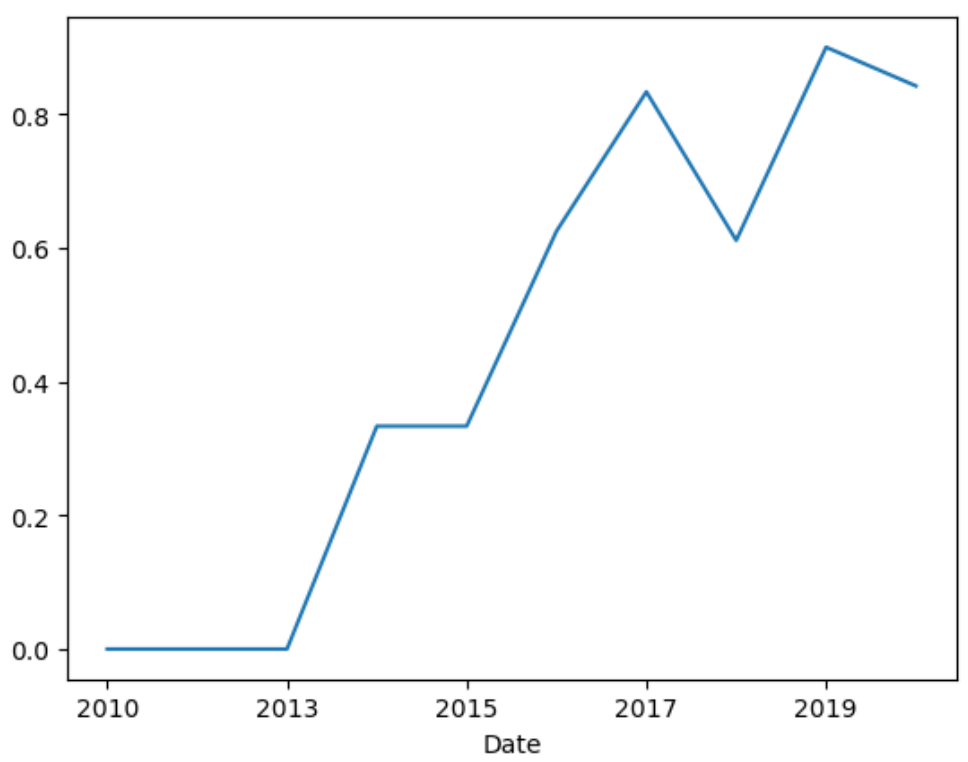
Data Wrangling

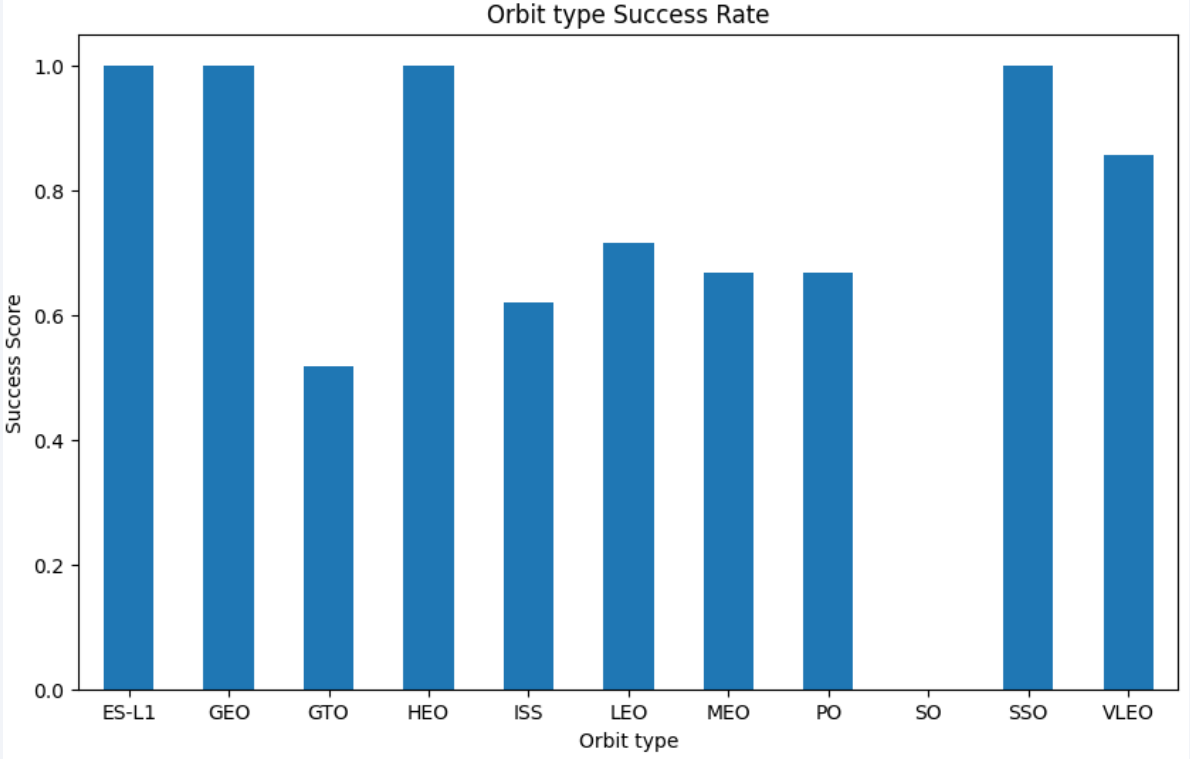
- Analyzed a part of data.
- Checked if null values.
- Checked data type of each column.
- Calculated: number of launches on each site, number and occurrence of each orbit, number and occurrence of mission outcome of the orbits
- Created a landing outcome label from Outcome column.
- GitHub repository URL of data wrangling related notebooks: https://github.com/romanmurzac/ibm-ds-certification/blob/main/Final_Exam/labs-jupyter-spacex-Data%20wrangling.ipynb



EDA with Data Visualization

- Exploratory Data Analysis performed on variables Flight Number, Payload Mass, Launch Site, Orbit, Class and Year.
- GitHub repository URL of EDA with data visualization notebook: https://github.com/romanmurzac/ibm-ds-certification/blob/main/Final_Exam/jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb





EDA with SQL

- Loaded data set into IBM DB2 Database.
- Queried using SQL Python integration.
- Queries were made to get a better understanding of the dataset.
- Queried information about launch site names, mission outcomes, various pay load sizes of customers and booster versions, and landing outcomes
- GitHub repository URL of EDA with SQL notebook: https://github.com/romanmurzac/ibm-ds-certification/blob/main/Final_Exam/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- Folium maps mark Launch Sites, successful and unsuccessful landings, and a proximity example to key locations: Railway, Highway, Coast, and City.
- This allows us to understand why launch sites may be located where they are. Also visualizes successful landings relative to location.
- GitHub URL of interactive map with Folium map: https://github.com/romanmurzac/ibm-ds-certification/blob/main/Final_Exam/lab_jupyter_launch_site_location.jupyterlite.ipynb

Build a Dashboard with Plotly Dash

- Dashboard includes a pie chart and a scatter plot.
- Pie chart can be selected to show distribution of successful landings across all launch sites and can be selected to show individual launch site success rates.
- Scatter plot takes two inputs: All sites or individual site and payload mass on a slider between 0 and 10000 kg.
- The pie chart is used to visualize launch site success rate.
- The scatter plot can help us see how success varies across launch sites, payload mass, and
- booster version category.
- GitHub URL of Plotly Dash lab: https://github.com/romanmurzac/ibm-ds-certification/blob/main/Final_Exam/spacex_dash_app.py

Predictive Analysis (Classification)

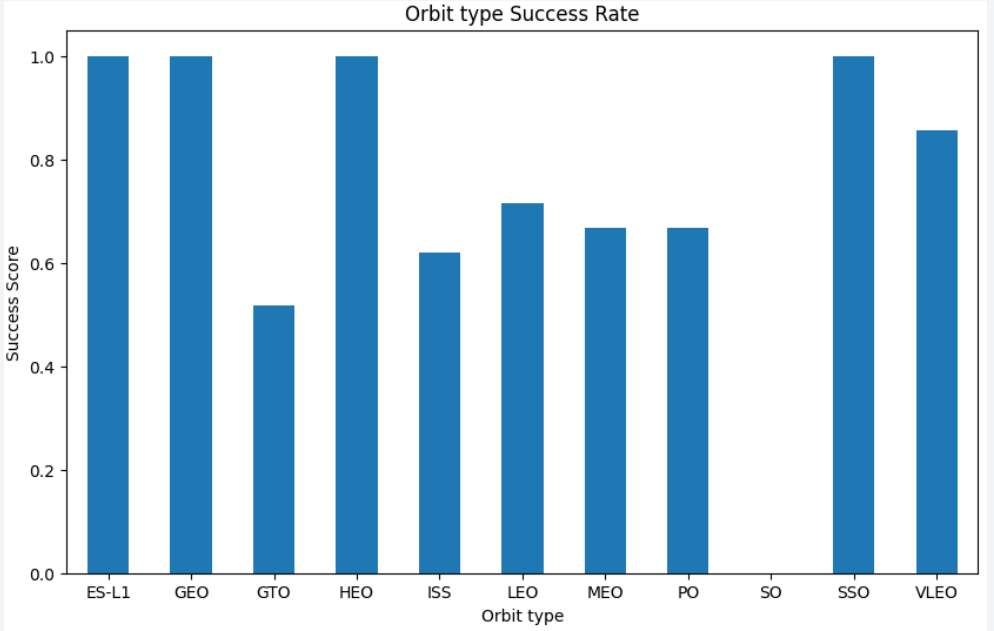
- Load SpaceX dataset (csv) in to a Dataframe and create NumPy array from the column class in data.
- Standardize data in X then reassign to variable X using transform.
- Train/test/split X and Y in to training and test data sets.
- Create and refine Models based on classification algorithms.
- Find the best performing model
- GitHub URL of predictive analysis lab: https://github.com/romanmurzac/ibm-ds-certification/blob/main/Final_Exam/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

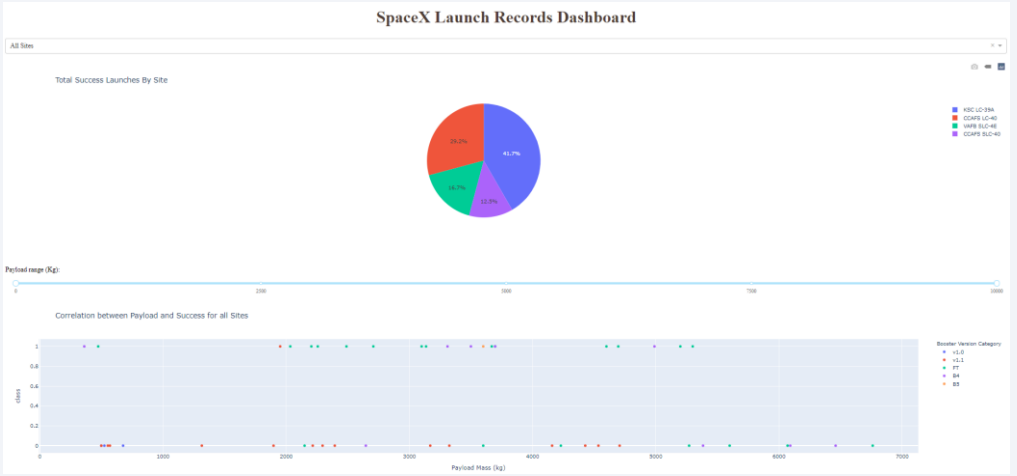
Results

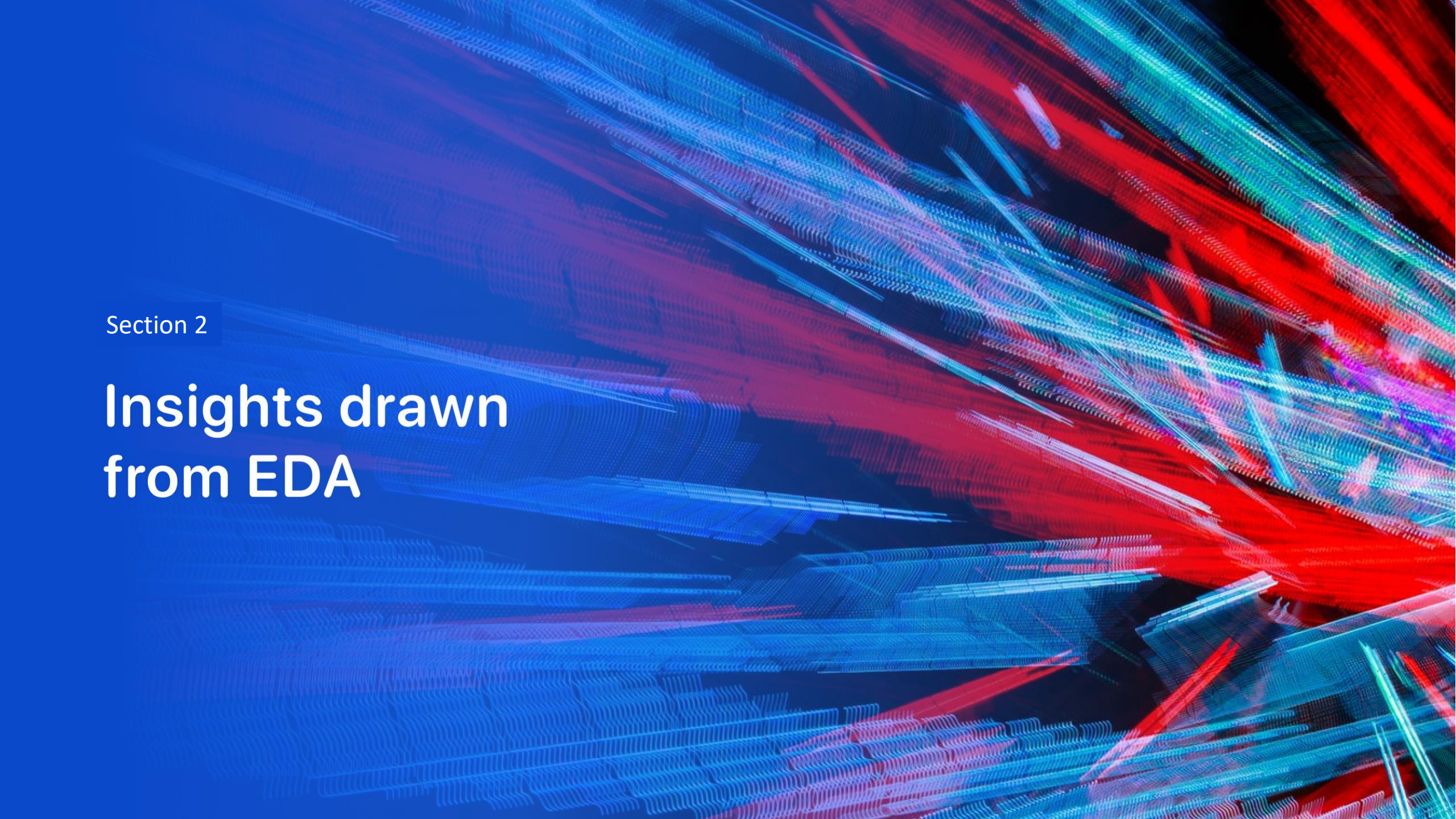
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

```
LogReg = logreg_cv.score(X_test,Y_test)
SVM = svm_cv.score(X_test,Y_test)
Tree = tree_cv.score(X_test,Y_test)
KNN = knn_cv.score(X_test,Y_test)

best_method = max(LogReg, SVM, Tree, KNN)

if best_method == LogReg:
    print(f"LogReg method performs best with final score: {round(LogReg, 2)}")
elif best_method == SVM:
    print(f"SVM method performs best with final score: {round(SVM, 2)}")
elif best_method == Tree:
    print(f"Tree method performs best with final score: {round(Tree, 2)}")
elif best_method == KNN:
    print(f"KNN method performs best with final score: {round(KNN, 2)}")
```

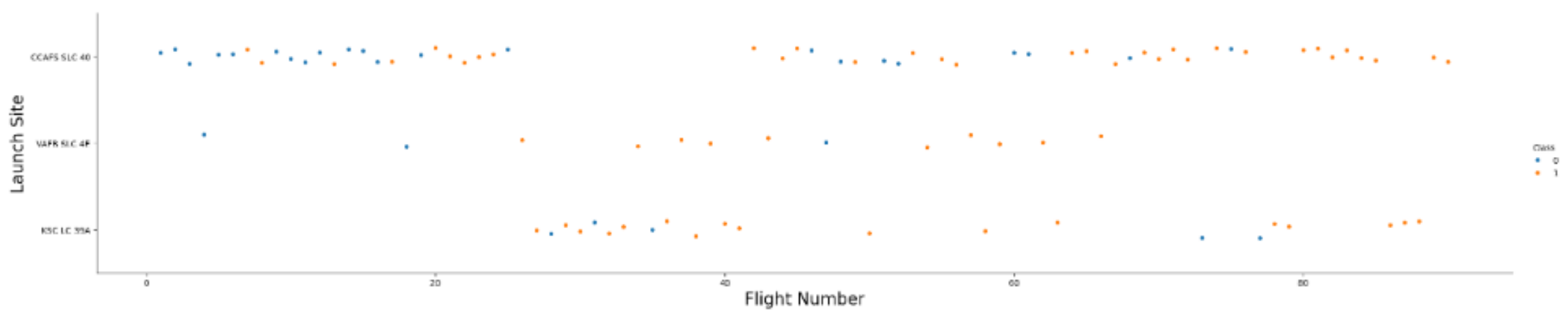
```
Tree method performs best with final score: 0.94
```





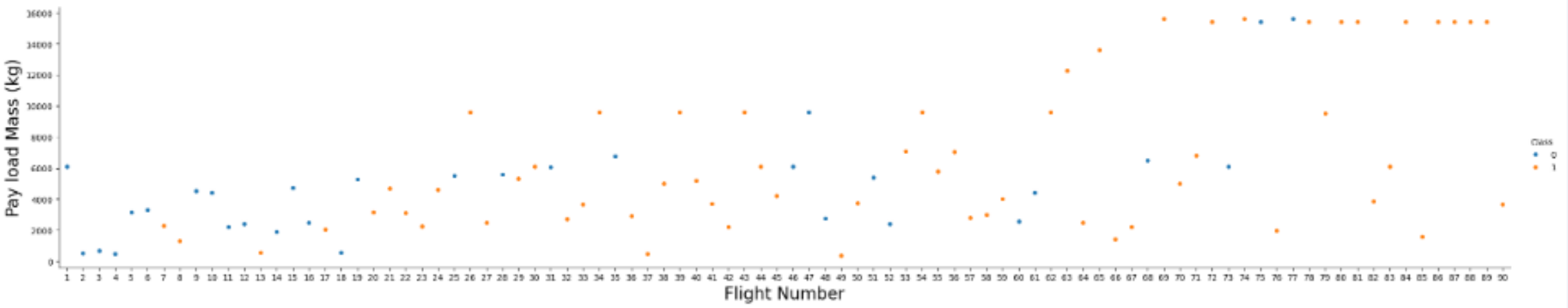

Section 2
Insights drawn from EDA

Flight Number vs. Launch Site



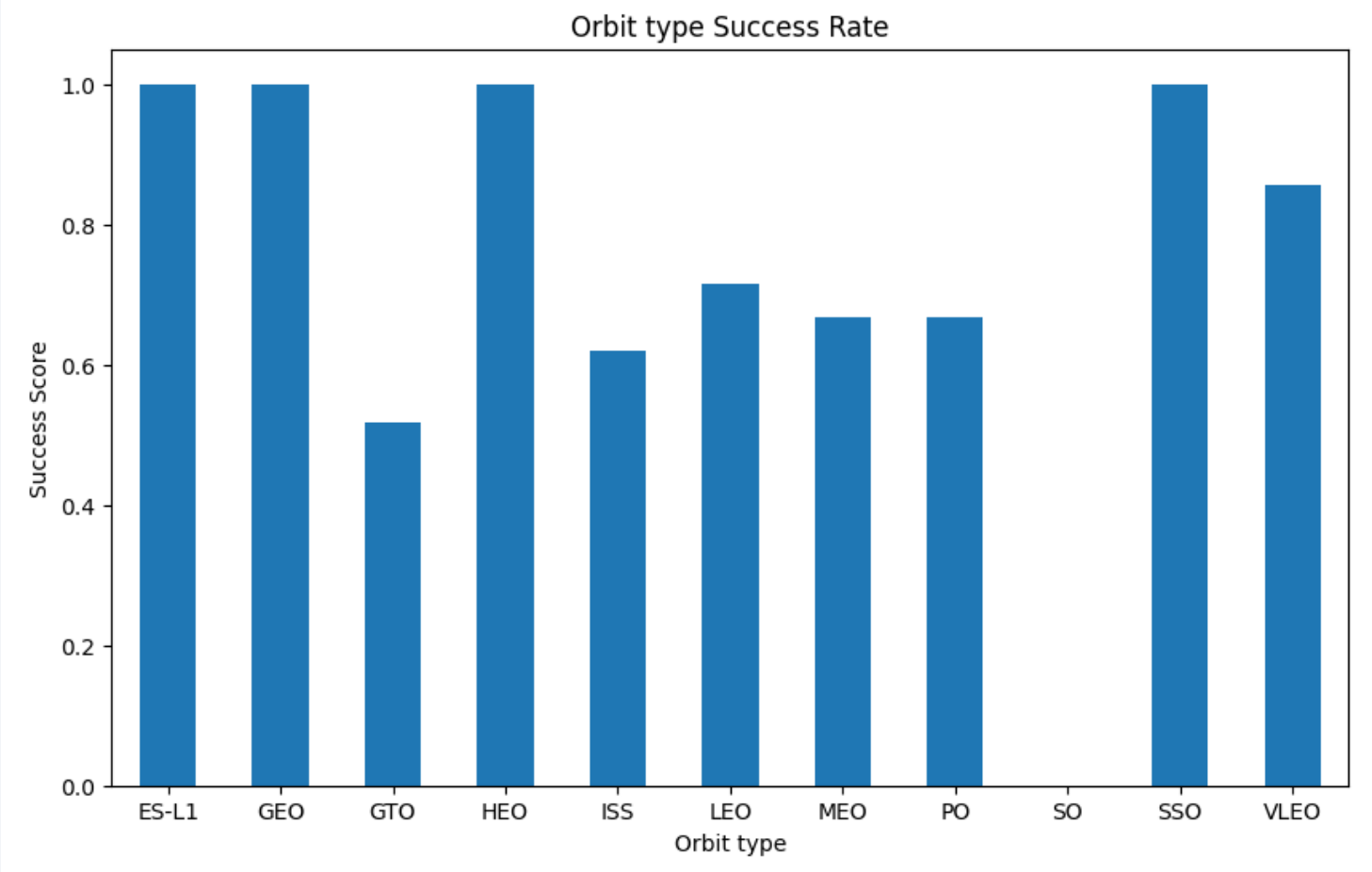
- Success rates (Class=1) increases as the number of flights increase
- For launch site ‘KSC LC 39A’, it takes at least around 25 launches before a first successful launch

Payload vs. Launch Site



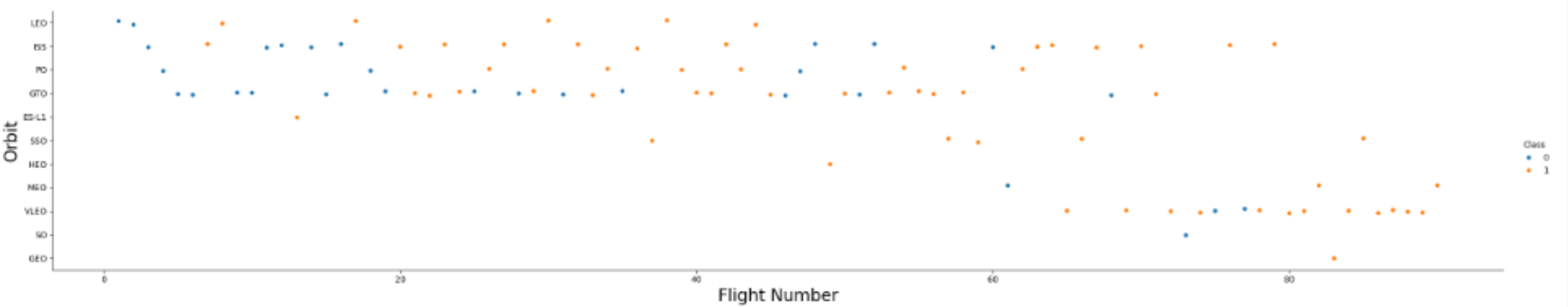
- For launch site ‘VAFB SLC 4E’, there are no rockets launched for payload greater than 10,000 kg
- Percentage of successful launch (Class=1) increases for launch site ‘VAFB SLC 4E’ as the payload mass increases
- There is no clear correlation or pattern between launch site and payload mass

Success Rate vs. Orbit Type



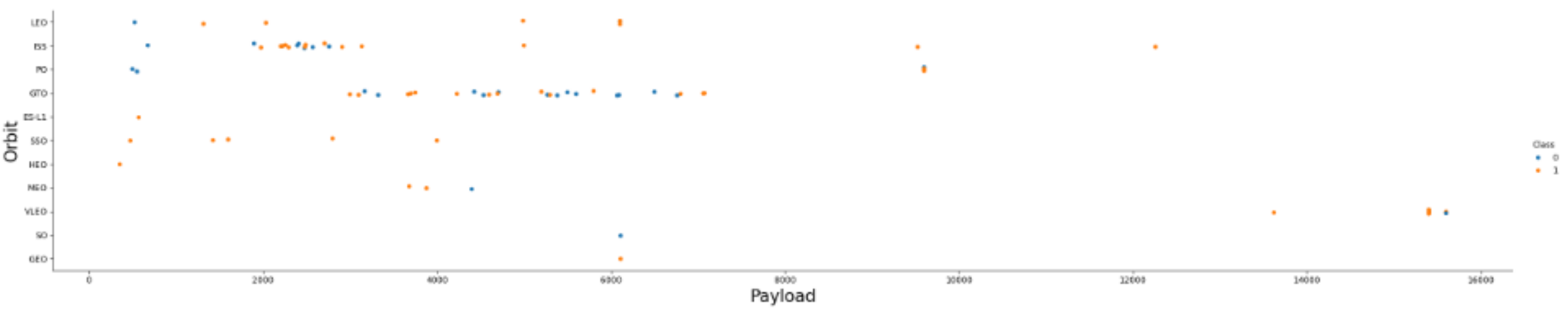
- Orbits ES-LI, GEO, HEO, and SSO have the highest success rates
- GTO orbit has the lowest success rate

Flight Number vs. Orbit Type



- For orbit VLEO, first successful landing (class=1) doesn't occur until 60+ number of flights
- For most orbits (LEO, ISS, PO, SSO, MEO, VLEO) successful landing rates appear to increase with flight numbers
- There is no relationship between flight number and orbit for GTO

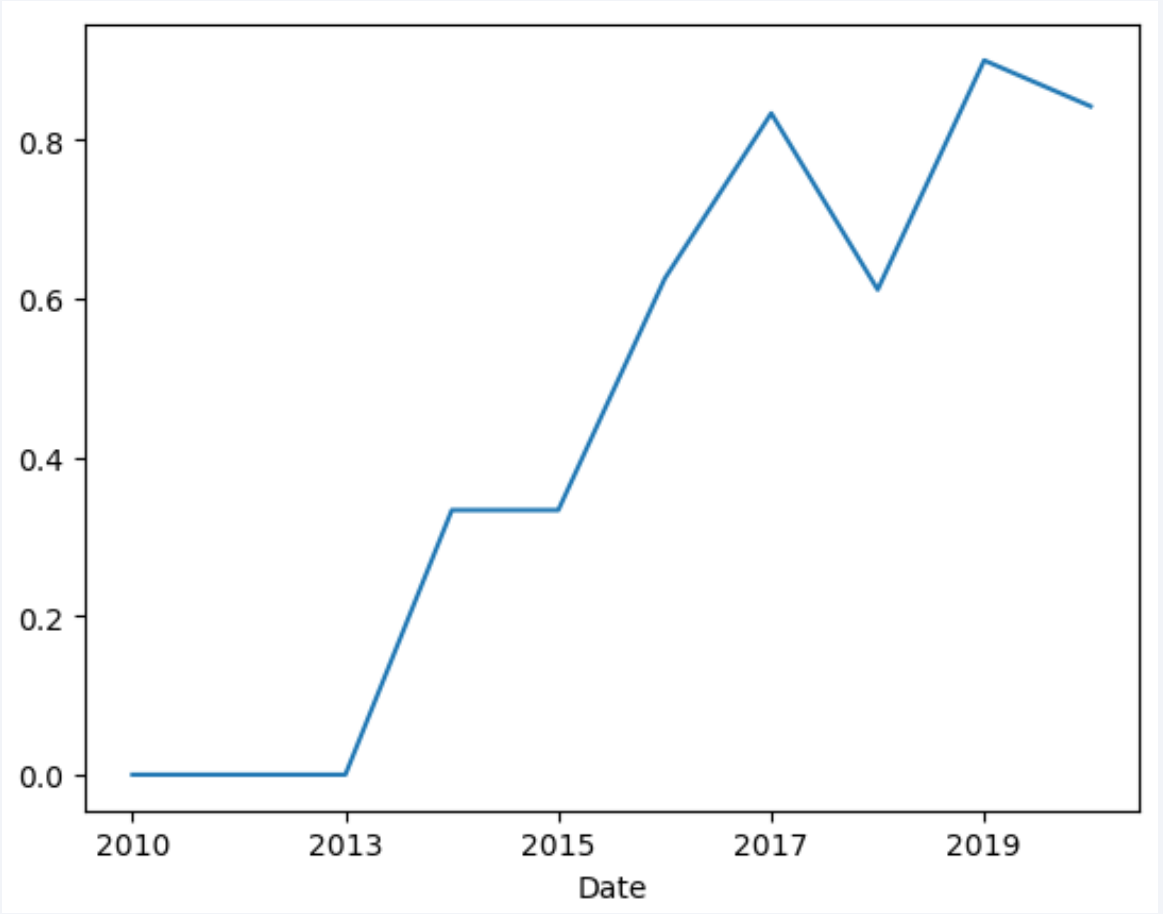
Payload vs. Orbit Type



- Successful landing rates (Class=1) appear to increase with pay load for orbits LEO, ISS, PO, and SSO
- For GEO orbit, there is not clear pattern between payload and orbit for successful or unsuccessful landing

Launch Success Yearly Trend

- Success rate (Class=1) increased by about 80% between 2013 and 2020
- Success rates remained the same between 2010 and 2013 and between 2014 and 2015
- Success rates decreased between 2017 and 2018 and between 2019 and 2020



All Launch Site Names

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

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

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

- ‘distinct’ returns only unique values from the queries column (Launch_Site)
- There are 4 unique launch sites

Launch Site Names Begin with 'CCA'

```
%sql SELECT * FROM SPACEXTABLE WHERE "Launch_Site" LIKE '%CCA%' LIMIT 5
```

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2

- Using keyword ‘Like’ and format ‘CCA%’, returns records where ‘Launch_Site’ column starts with “CCA”.
- Limit 5, limits the number of returned records to 5

Total Payload Mass

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

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

Total_Payload_Mass
45596

- ‘sum’ adds column ‘PAYLOAD_MASS_KG’ and returns total payload mass for customers named ‘NASA (CRS)’

Average Payload Mass by F9 v1.1

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

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

AVG("PAYLOAD_MASS__KG_")
2928.4

- avg' keyword returns the average of payload mass in 'PAYLOAD_MASS_KG' column where booster version is 'F9 v1.1'

First Successful Ground Landing Date

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

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

MIN("Date")
2015-12-22

- ‘min(Date)’ selects the first or the oldest date from the ‘Date’ column where first successful landing on group pad was achieved
- Where clause defines the criteria to return date for scenarios where ‘Landing_Outcome’ value is equal to ‘Success (ground pad)’

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%sql SELECT DISTINCT "Booster_Version" FROM SPACEXTABLE WHERE ("PAYLOAD_MASS__KG_" BETWEEN 4000 AND 6000) AND ("Landing_Outcome" = 'Success (drone ship)')
```

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

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

- The query finds the booster version where payload mass is greater than 4000 but less than 6000 and the landing outcome is success in drone ship
- The ‘and’ operator in the where clause returns booster versions where both conditions in the where clause are true

Total Number of Successful and Failure Mission Outcomes

```
%sql SELECT "Mission_Outcome", COUNT("Mission_Outcome") AS Mission FROM SPACEXTABLE GROUP BY "Mission_Outcome"
```

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

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

- The ‘group by’ keyword arranges identical data in a column in to group
- In this case, number of mission outcomes by types of outcomes are grouped in column ‘counts’

Boosters Carried Maximum Payload

```
%sql SELECT "Booster_Version", "PAYLOAD_MASS__KG_" FROM SPACEXTABLE WHERE "PAYLOAD_MASS__KG_" = (SELECT MAX("PAYLOAD_MASS__KG_") FROM SPACEXTABLE)
 * sqlite:///my_data1.db
Done.
```

Booster_Version	PAYLOAD_MASS__KG_
F9 B5 B1048.4	15600
F9 B5 B1049.4	15600
F9 B5 B1051.3	15600
F9 B5 B1056.4	15600
F9 B5 B1048.5	15600
F9 B5 B1051.4	15600
F9 B5 B1049.5	15600
F9 B5 B1060.2	15600
F9 B5 B1058.3	15600
F9 B5 B1051.6	15600
F9 B5 B1060.3	15600
F9 B5 B1049.7	15600

- The sub query returns the maximum payload mass by using keywork ‘max’ on the pay load mass column
- The main query returns booster versions and respective payload mass where payload mass is maximum with value of 15600

2015 Launch Records

```
%sql SELECT SUBSTR(Date, 6, 2) AS Month, "Landing_Outcome", "Booster_Version", "Launch_Site" FROM SPACEXTABLE WHERE "Landing_Outcome" LIKE "%Fail%" AND SUBSTR(Date, 0, 5) = '2015'
 * sqlite:///my_data1.db
Done.
```

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

- The query lists landing outcome, booster version, and the launch site where landing outcome is failed in drone ship and the year is 2015
- The ‘and’ operator in the where clause returns booster versions where both conditions in the where clause are true
- The ‘year’ keywork extracts the year from column ‘Date
- The results identify launch site as ‘CCAFS LC-40’ and booster version as F9 v1.1 B1012 and B1015 that had failed landing outcomes in drop ship in the year 2015

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

```
%sql SELECT "Landing_Outcome", COUNT(*) AS "Landing_Counts" FROM SPACEXTABLE WHERE "Date" BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY "Landing_Outcome" ORDER BY COUNT(*) DESC
 * sqlite:///my_data1.db
Done.
```

Landing_Outcome	Landing_Counts
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

- The ‘group by’ key word arranges data in column ‘Landing__Outcome’ into groups
- The ‘between’ and ‘and’ keywords return data that is between 2010-06-04 and 2017-03-20
- The ‘order by’ keyword arranges the counts column in descending order
- The result of the query is a ranked list of landing outcome counts per the specified date range

Section 3

Launch Sites Proximities Analysis

SpaceX Falcon9 – Launch Sites Map

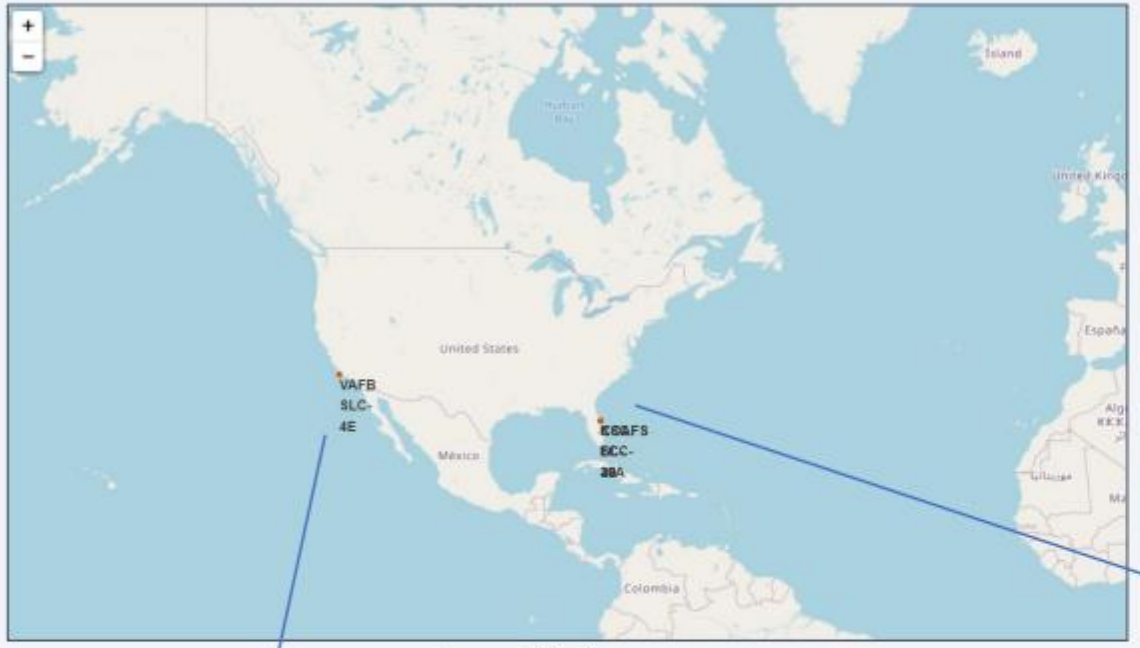


Fig 1 – Global Map

Figure 1 on left displays the Global map with Falcon 9 launch sites that are located in the United States (in California and Florida). Each launch site contains a circle, label, and a popup to highlight the location and the name of the launch site. It is also evident that all launch sites are near the coast.

Figure 2 and Figure 3 zoom in to the launch sites to display 4 launch sites:

- VAFB SLC-4E (CA)
- CCAFS LC-40 (FL)
- KSC LC-39A (FL)
- CCAFS SLC-40 (FL)

Fig 2 – Zoom 1

Fig 3 – Zoom 2

SpaceX Falcon9 – Success / Failed Launch Map

Fig 1 – US map with all Launch Sites

- Figure 1 is the US map with all the Launch Sites. The numbers on each site depict the total number of successful and failed launches
- Figure 2, 3, 4, and 5 zoom in to each site and displays the success/fail markers with green as success and red as failed
- By looking at each site map, KSC LC-39A Launch Site has the greatest number of successful launches

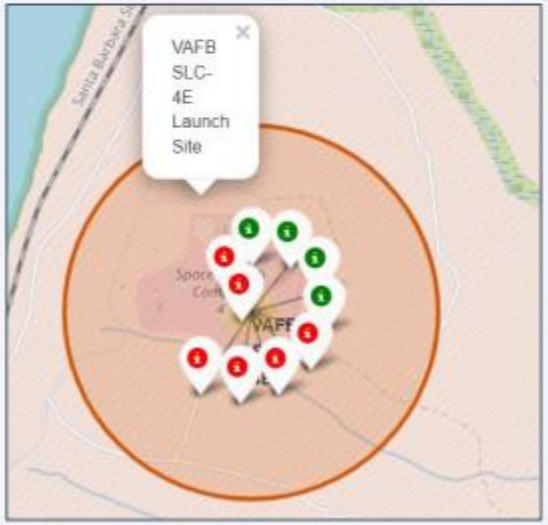


Fig 2 – VAFB Launch Site with success/failed markers

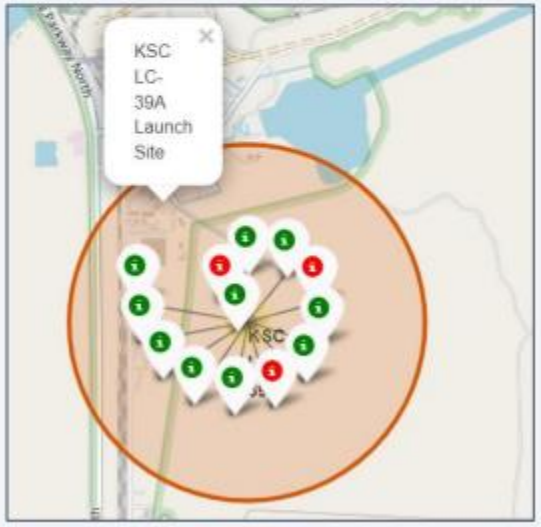


Fig 3 – KSC LC-39A success/failed markers

Fig 4 – CCAFS SLC-40 success/failed markers

Fig 5 – CCAFS SLC-40 success/failed markers

SpaceX Falcon9 – Proximity Distance Map

Fig 1 – Proximity site map for VAFB SLC-4E

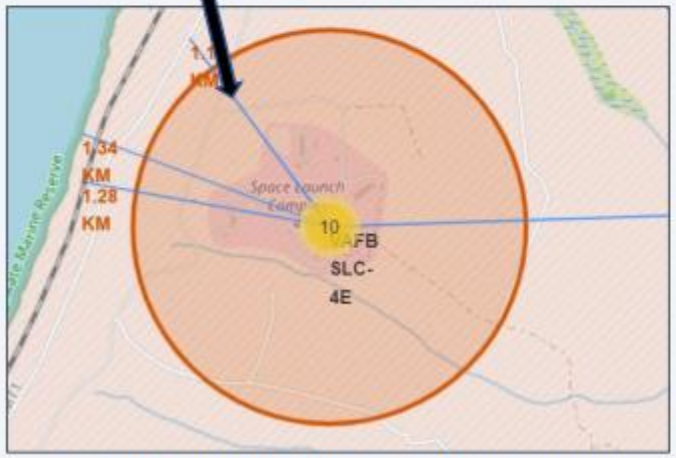
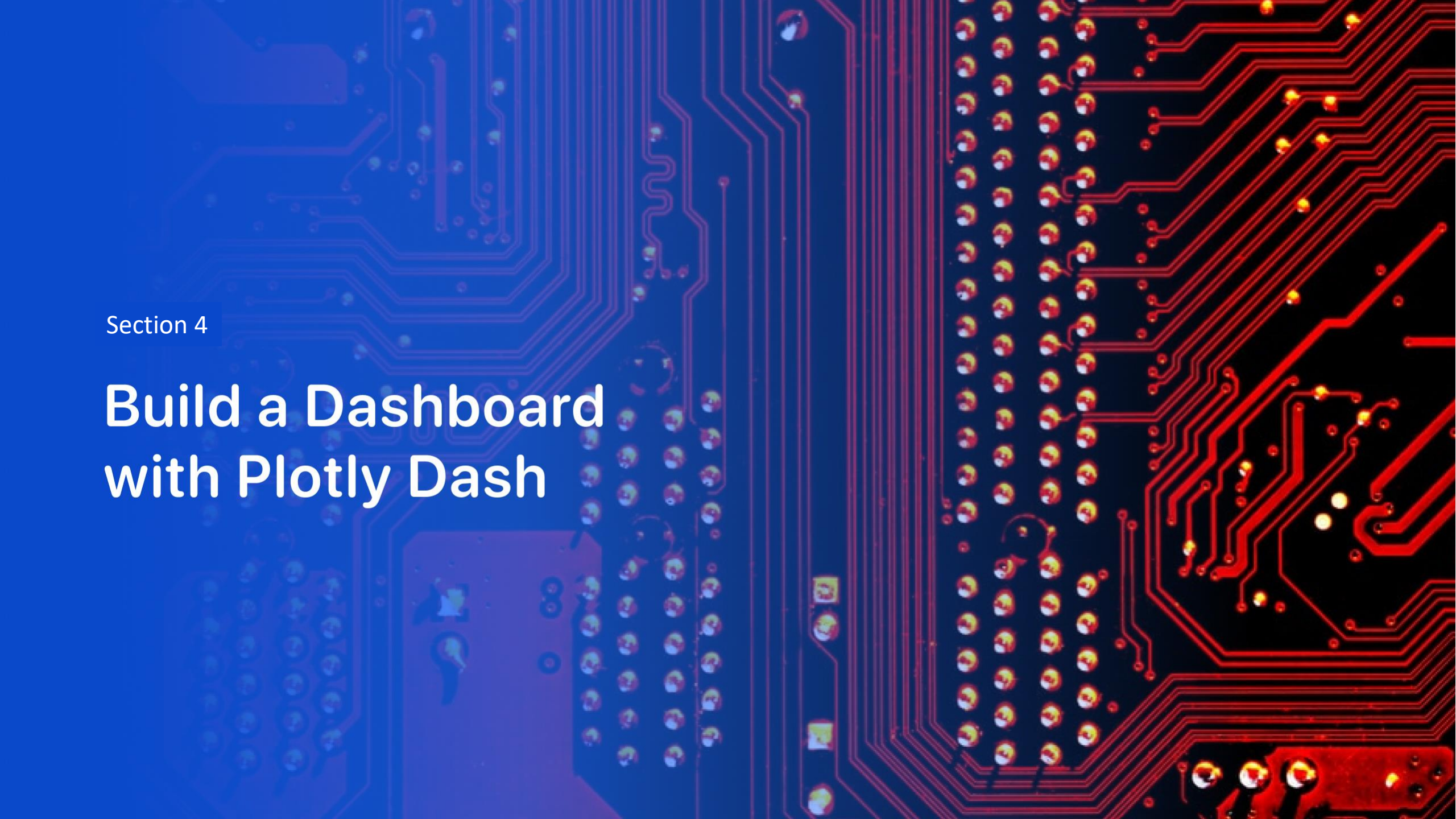


Fig 2 – Zoom in for sites – coastline, railroad, and highway

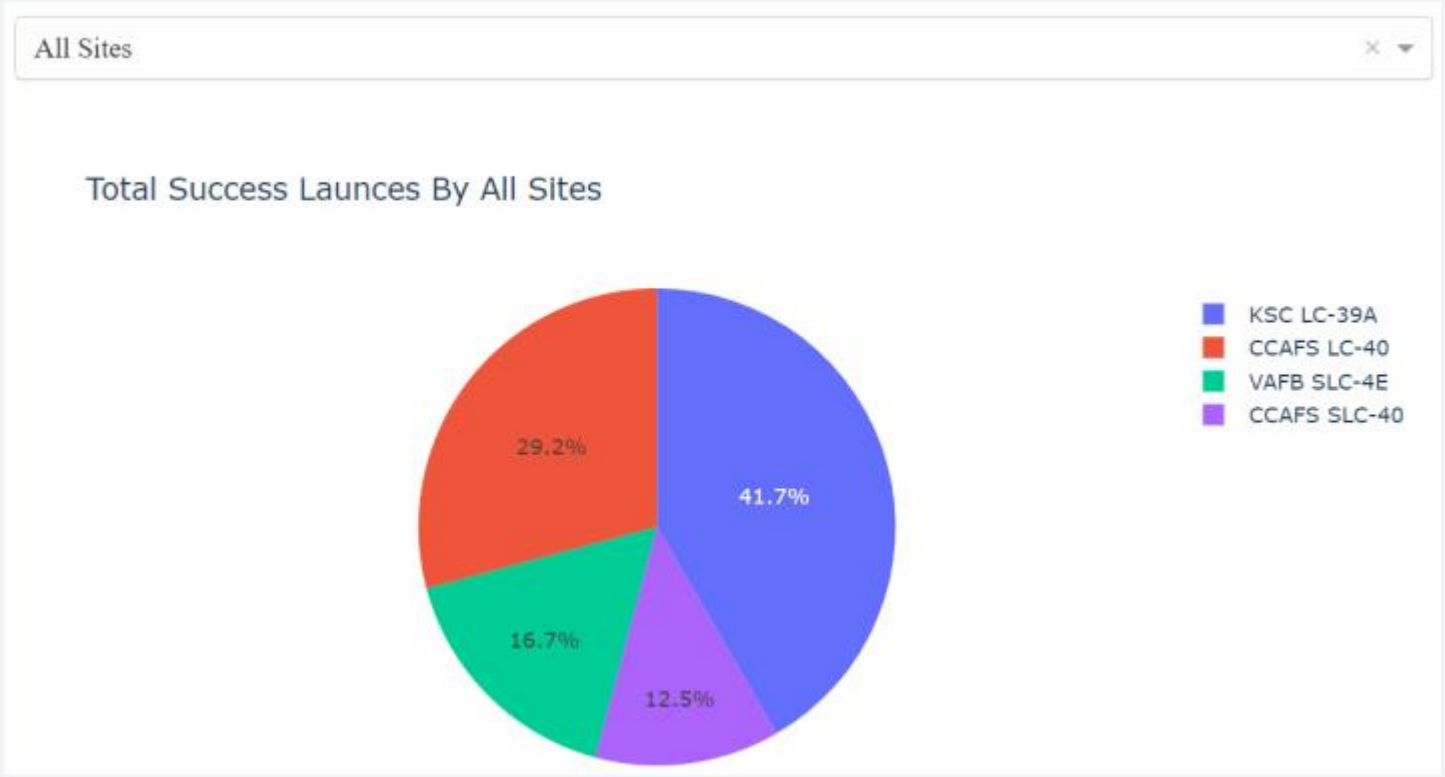
Figure 1 displays all the proximity sites marked on the map for Launch Site VAFB SLC-4E. City Lompoc is located further away from Launch Site compared to other proximities such as coastline, railroad, highway, etc. The map also displays a marker with city distance from the Launch Site (14.09 km)

Figure 2 provides a zoom in view into other proximities such as coastline, railroad, and highway with respective distances from the Launch Site

In general, cities are located away from the Launch Sites to minimize impacts of any accidental impacts to the general public and infrastructure. Launch Sites are strategically located near the coastline, railroad, and highways to provide easy access to resources.

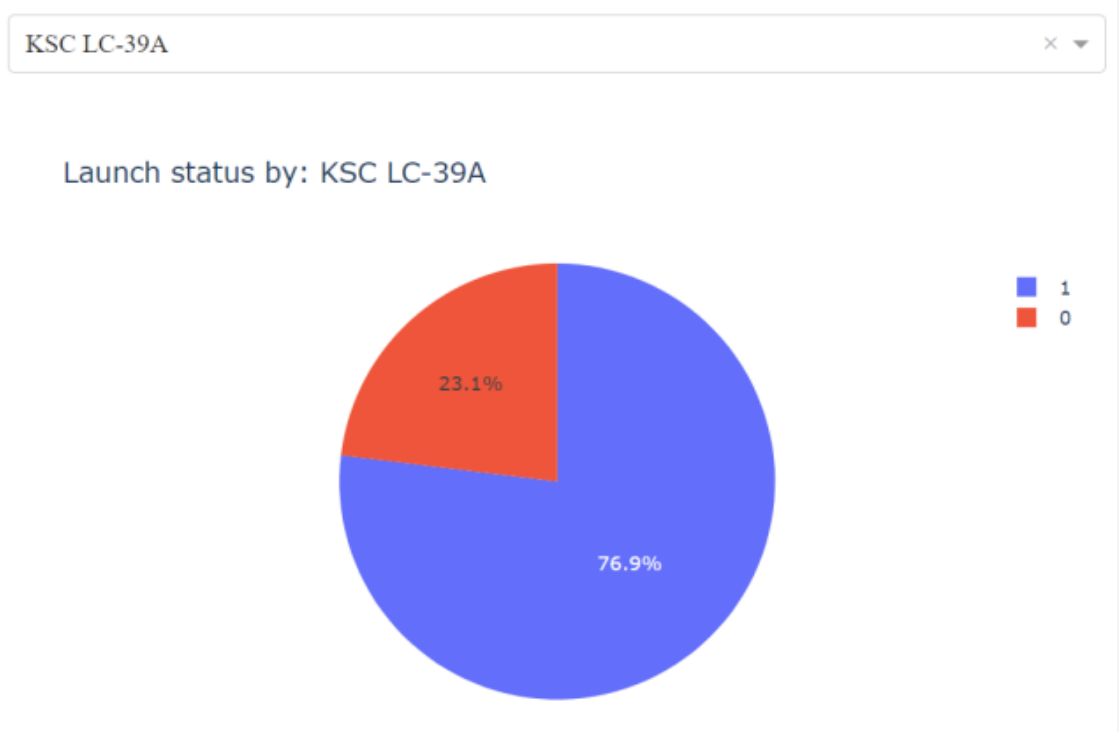

Section 4
Build a Dashboard with Plotly Dash

Launch Success Counts



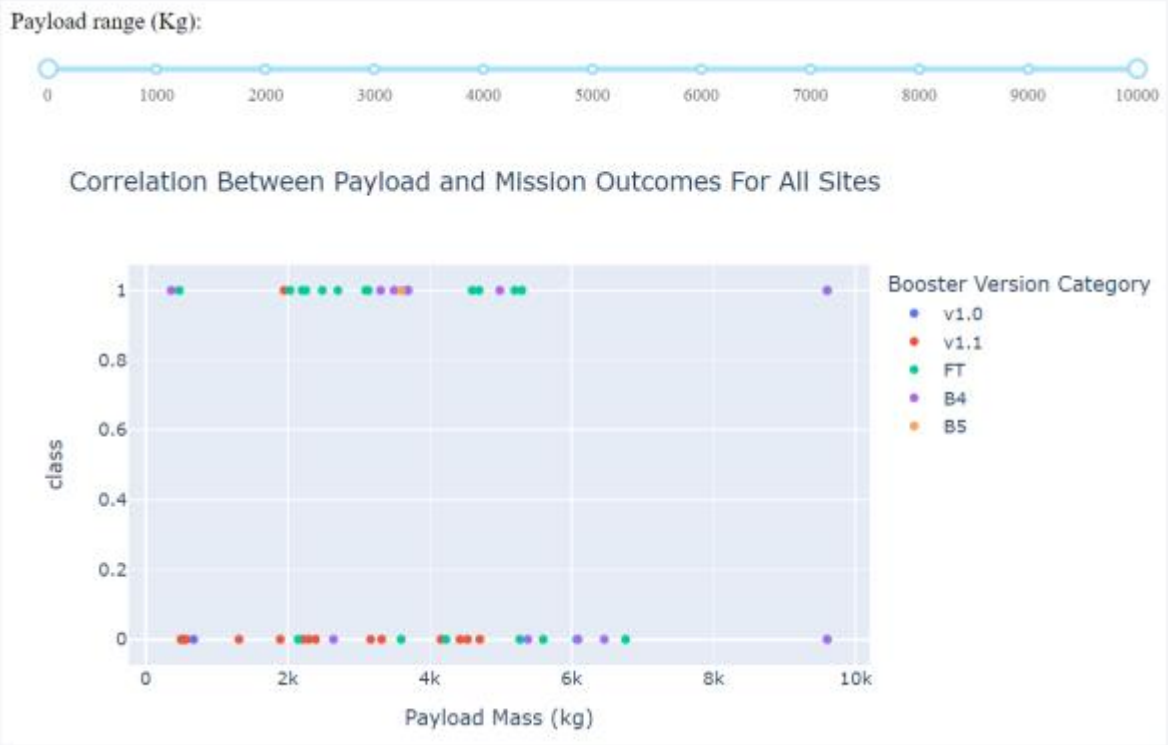
- Launch Site ‘KSC LC-39A’ has the highest launch success rate
- Launch Site ‘CCAFS SLC-40’ has the lowest launch success rate

Highest Launch Success Ratio



- KSC LC-39A Launch Site has the highest launch success rate and count
- Launch success rate is 76.9%
- Launch success failure rate is 23.1%

Payload vs Launch Outcome

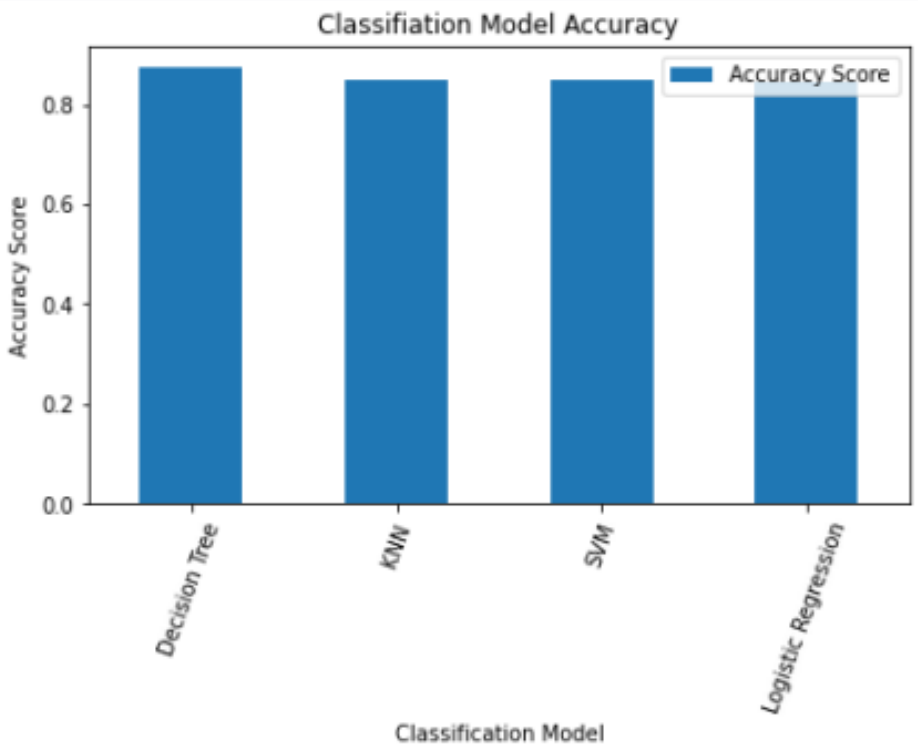


- Most successful launches are in the payload range from 2000 to about 5500
- Booster version category ‘FT’ has the most successful launches
- Only booster with a success launch when payload is greater than 6k is ‘B4’

Section 5

Predictive Analysis (Classification)

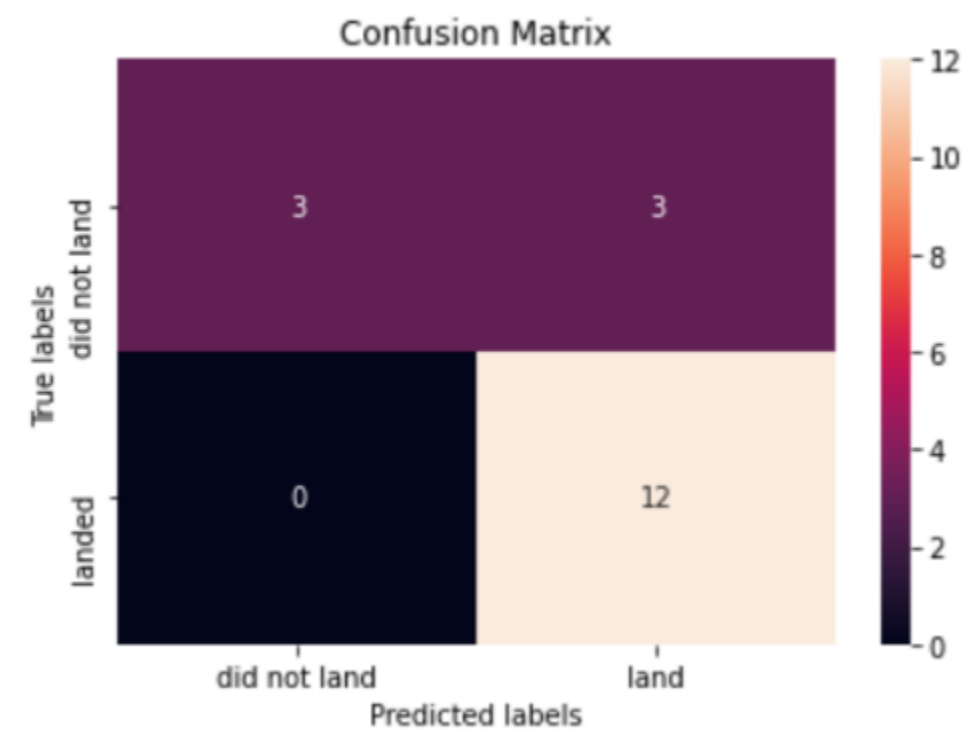
Classification Accuracy



	Algo Type	Accuracy Score	Test Data Accuracy Score
2	Decision Tree	0.875000	0.833333
3	KNN	0.848214	0.833333
1	SVM	0.848214	0.833333
0	Logistic Regression	0.846429	0.833333

- Based on the Accuracy scores and as also evident from the bar chart, Decision Tree algorithm has the highest classification score with a value of .8750
- Accuracy Score on the test data is the same for all the classification algorithms based on the data set with a value of .8333
- Given that the Accuracy scores for Classfication algorithms are very close and the test scores are the same, we may need a broader data set to further tune the models

Confusion Matrix



- The confusion matrix is same for all the models (LR, SVM, Decision Tree, KNN)
- Per the confusion matrix, the classifier made 18 predictions
- 12 scenarios were predicted Yes for landing, and they did land successfully (True positive)
- 3 scenarios (top left) were predicted No for landing, and they did not land (True negative)
- 3 scenarios (top right) were predicted Yes for landing, but they did not land successfully (False positive)
- Overall, the classifier is correct about 83% of the time ((TP + TN) / Total) with a misclassification or error rate ((FP + FN) / Total) of about 16.5%

Conclusions

- As the numbers of flights increase, the first stage is more likely to land successfully
- Success rates appear go up as Payload increases but there is no clear correlation between Payload mass and success rates
- Launch success rate increased by about 80% from 2013 to 2020
- Launch Site ‘KSC LC-39A’ has the highest launch success rate and Launch Site ‘CCAFS SLC 40’ has the lowest launch success rate
- Orbits ES-L1, GEO, HEO, and SSO have the highest launch success rates and orbit GTO the lowest
- Lunch sites are located strategically away from the cities and closer to coastline, railroads, and highways
- The best performing Machine Learning Classfication Model is the Decision Tree with an accuracy of about 87.5%. When the models were scored on the test data, the accuracy score was about 83% for all models. More data may be needed to further tune the models and find a potential better fit.

Appendix

```
LogReg = logreg_cv.score(X_test,Y_test)
SVM = svm_cv.score(X_test,Y_test)
Tree = tree_cv.score(X_test,Y_test)
KNN = knn_cv.score(X_test,Y_test)

best_method = max(LogReg, SVM, Tree, KNN)

if best_method == LogReg:
    print(f"LogReg method performs best with final score: {round(LogReg, 2)}")
elif best_method == SVM:
    print(f"SVM method performs best with final score: {round(SVM, 2)}")
elif best_method == Tree:
    print(f"Tree method performs best with final score: {round(Tree, 2)}")
elif best_method == KNN:
    print(f"KNN method performs best with final score: {round(KNN, 2)}")
```

```
LogReg method performs best with final score: 0.83
```

Thank you!