IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Parag K Harnaikar
2024 April 9th

Purpose of this presentation

- ✓This report records how landing outcome of booster and its reuse reduces the price of a launch by Space X.
- ✓It explores, using Data Science, whether landing outcome of booster used in launch can be predicted.
- ✓The prediction is important because it determines Space X launch price. This will help Space Y understand launch prices for them to be competitive ahead of the launch.
- ✓The report includes details of data sources, data wrangling, exploration, insights drawn, interactive analytics, predictive model analysis, evaluation and conclusions.

Outline

- Executive Summary
- Introduction
- Methodology
  - *Results of Exploratory data analysis*
  - *Results of Interactive data analytics*
  - *Results of Predictive analysis*
- Insights drawn from EDA
- Launch Sites Proximities Analysis
- Build a Dash with Plotly Dashboard
- Predictive Analysis (Classification)
  - *Conclusions*
  - *Appendix*
  - *Abbreviations and Acronyms*

Executive Summary

- Summary of methodologies
  - Space X API and Web scraping for collecting data, Falcon 9 launches up to 13-Nov-2020 only considered and empty data points replaced with averages.
  - Data explored using Structured Query and Visualizations, Interactive analytics using Folium maps and Plotly Dashboard, Relationship between features and landing outcome identified.
  - Evaluation of classification models, using cross validation train and test data set folds, to predict booster landing outcome of launch.
- Summary of all results
  - Number of flights, payload mass, orbit type, booster version all affect landing outcome of launch booster and hence are used to predict the outcome.
  - Decision tree model provides the most accurate prediction, with accuracy score of 89%.

Introduction

- Project background and context
  - ✓ ‘Space Y’ to enter the industry of commercial rocket launch.
  - ✓ The first stage is the biggest piece, does most of the work and hence decides the price of a launch.
  - ✓ The main competition is ‘Space X’, because ‘Space X’ can ‘reuse’ their Falcon 9 first stage.
  - ✓ As a result, Space X has been able to reduce the price of launch to $ 62 million compared to upwards of $ 165 million by others.
  - ✓ To be competitive, Space Y needs to determine or predict the price of each launch.
- Problems you want to find answers
  - ✓ *Will the first stage of next Space X launch land successfully?*
  - ✓ *Will Space X reuse the first stage?*
  - ✓ *Can we predict the first stage landing outcome using the historical details?*

Section 1

Methodology

Methodology

Summary

- Data collection methodology:
  - Space X interface.
  - Scraping of the Wikipedia Space X Falcon web pages.
- Perform data wrangling
  - Falcon 9 only details up to 13-Nov-2020 are considered.
  - Empty Payload Mass are replaced with their average.
  - Additional features from other Space X interface end points are added.
  - Landing outcome is normalized for analysis.

Methodology…continued

Summary

- Perform exploratory data analysis (EDA) using visualization and Structured Query
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Categorical features are hot-encoded.
  - Data are split into 80% Train and 20% Test sets.
  - All features are standardized in a cross-validation grid with different Machine Learning methods for Predictive analysis.
  - Different parameters are used in the pipeline for fine tuning the models.
  - The model are trained using Train dataset and tested for predicting the outcome of Test dataset.
  - The models with the best accuracy score are selected.
  - Finally the model performance is evaluated using confusion matrix and accuracy scores.

Data Collection

- Data are collected using GET request from SpaceX API: `https://api.spacexdata.com/v4/`
- The GET request response is decoded and normalized using response json method and panda's json normalize method respectively and loaded into a panda data frame.
- Data are restricted to up to the date of 13-Nov-2020.
- Data in panda data frame is supplemented with details from additional Space X API end points:
  - '**rockets**' end point: booster name.
  - '**payloads**' end point: payload mass and orbit it is going to.
  - '**launchpads**' end point: Name of Launch Site used, its longitude, latitude.
  - '**cores**' end point: block, core is reused or not, reuse count, serial, flight, grid fins used nor not, legs are used or not, landing success, landing type, landing pad used.

Data Collection…continued

Data collection using additional Space X API end points:

- ‘rockets’ end point: https://api.spacexdata.com/v4/rockets/<rocket>
- ‘payloads’ end point: https://api.spacexdata.com/v4/payloads/<payload>
- ‘launchpads’ end point: https://api.spacexdata.com/v4/launchpads/<launchpad>
- ‘cores’ end point: https://api.spacexdata.com/v4/cores/<core>

Collected data are saved in *dataset_part_1.csv*.

Data Collection…continued

Data collection by web scraping a Space X Wikipedia page using Python BeautifulSoup module:

- Wikipedia static page: https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922
- All html tables with an attribute filter of ‘wikitable plainrowheaders collapsible’ in the page are located using BeautifulSoup find_all method.
- From the table header row using find_all method, first these column names are extracted, using a helper function ‘extract_column_from_header’: Flight No, Launch Site, Payload, Payload Mass, Orbit, Customer, Launch Outcome.
- From the table record header row cells using find_all method, these data values are extracted using helper functions ‘date_time’, ‘booster_version’, ‘landing_status’, and ‘get_mass’: Flight No, Launch Site, Payload, Payload Mass, Orbit, Customer, Launch Outcome, Date, Time, Booster Landing, Booster Version.
- The data scraped from the web are saved in *spacex_web_scraped.csv*.

Data Collection – Space X API

Please refer to the flow chart for the order of data collection steps:

- Calls to five Space X API end points are made.
- GET request is used.
- json methods on the response are used to decode and normalize data.
- Data is restricted to 13-Nov-2020.
- GET requests to four additional end points are made to collect additional features.

GitHub URL of the completed Space X API calls notebook:
https://github.com/ParagHarnai/Care/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

Space X API URL:
https://api.spacexdata.com/v4/rockets

data = response of REST API GET request

Decode the data using response json method

Normalize data using panda jason normalize method

Restrict the data to 13-Nov-2020 in panda data frame

Supplement the data with features from other Space X API end points:
--> ‘rockets’ end point: booster name
--> ‘payloads’ end point: payload mass and orbit it is going to
--> ‘launchpads’ end point: Name of Launch Site used, its longitude, latitude
--> ‘cores’ end point: block, reused, reuse count, serial, flight, gridfins, legs, landing success, landing type, landpad

Data Collection – Web Scraping

Please refer to the flow chart for the order of data collection steps:

- Calls to Space X Wikipedia static page are made first to extract column names and again to extract data values.
- GET request is used.
- Column names and Data values are processed using helper functions.

GitHub URL of the completed Web Scraping notebook:
https://github.com/ParagHarnai/Care/blob/main/jupyter-labs-webscraping.ipynb

Space X Wikipedia page:
https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922

↓

data = response of GET request

↓

Decode the data using BeautifulSoup module

↓

Extract the tables, iterate through one header row first to extract column nams using helper function extract_column_from_header

↓

Extract the tables, iterate through all table record header rows, to extract data values using helper functions ‘date_time’, ‘booster_version’, ‘landing_status’, and ‘get_mass’

Data Wrangling

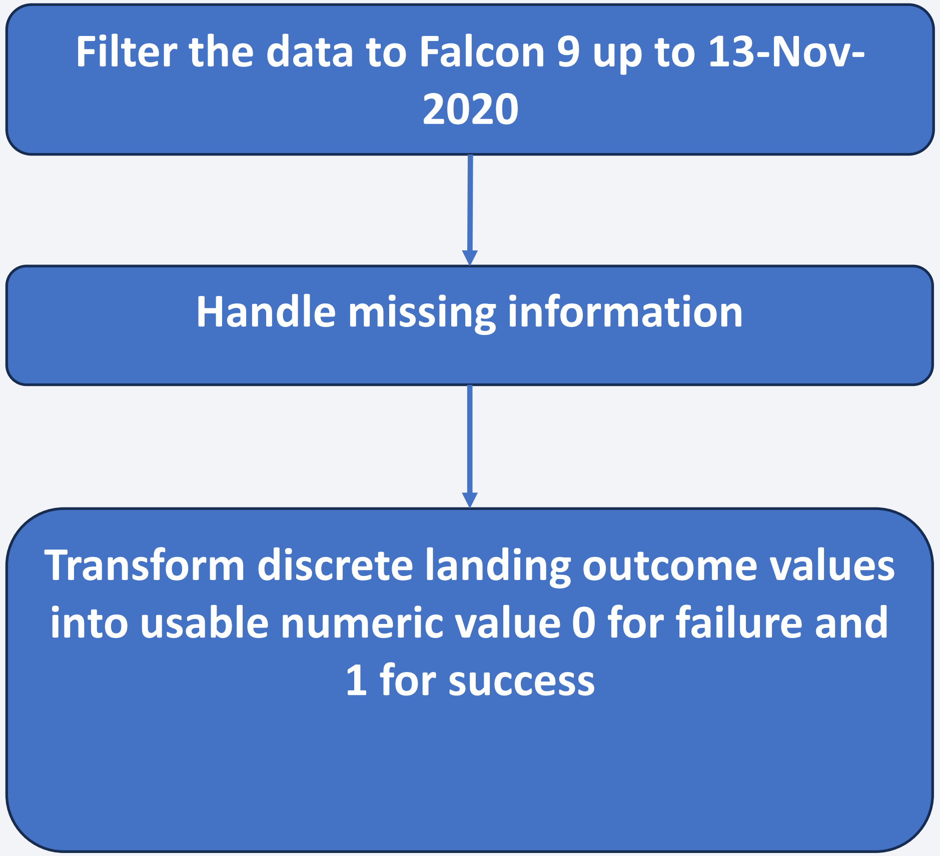
- Falcon 9 Booster only considered and its details up to 13-Nov-2020 only are considered.
- Empty Payload Mass are replaced with their average.
- Additional features from other Space X interface end points are added.
- Discrete Landing Outcome values are transformed into 0 for failure and 1 for success.

Data Wrangling…continued

Please refer to the flow chart for the order of data wrangling steps:

- Filter the data to Falcon 9 up to 13-Nov-2020.
- Handle missing information.
- Transform discrete outcome values into usable numeric values 0 and 1.

GitHub URL of the completed Data Wrangling notebook: https://github.com/ParagHarnai/Care/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb



EDA with Data Visualization

Chart\Plot	Features\Outcome	Purpose
Categorical scatter	Flight, Payload Mass, Outcome overlay	Over a range of flights how does Payload Mass affect the landing outcome?
Categorical scatter	Flight, Launch Site, Landing Outcome overlay	Is there a relationship between Launch Site and landing outcome with respect to number of flights?
Categorical scatter	Payload Mass, Launch Site, Landing Outcome overlay	Does Payload Mass have impact on the landing outcome by Launch Sites?
Bar	Orbit type, Landing Outcome success rate	Does the type of Orbit decide the landing success rate?
Categorical scatter	Flight, Orbit type, Landing Outcome overlay	For various Orbit types does the landing outcome change as the number of flights increases?
Categorical scatter	Flight, Orbit type, Landing Outcome overlay	How is the landing outcome based on mass of payload within various orbit types?
Line	Average landing success rate over the years between 2013 and 2020	What is the trend of landing success rate over the years of launches between 2013 and 2020?

EDA with Data Visualization…continued

GitHub URL of the completed EDA with Data Visualization notebook:

https://github.com/ParagHarnai/Care/blob/main/jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb

EDA with SQL

Summary of scripts run in sql database for exploratory analysis:

- Unique launch sites.
- Sample records for Launch Site that begins with ‘CCA’ in its name.
- Total payload mass carried by boosters launched for the customer NASA (CRS).
- Average payload mass carried by booster version F9 v1.1.
- The date when the first successful landing outcome on ground pad was achieved.
- Names of boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000.

EDA with SQL…continued

Summary of scripts run in sql database for exploratory analysis

- Total number of successful and failed mission outcomes.
- Names of booster versions which carried the maximum payload mass (using a subquery).
- Month names, failure outcomes of landing on drone ship, booster version, and launch site in the months of year 2015.
- Aggregate and sort the count of landing outcomes between 4-Jun-2010 and 20-Mar-2017 in descending order.

GitHub URL of the completed EDA with SQL notebook:
https://github.com/ParagHarnai/Care/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Interactive Map with Folium

Map annotations	Why did we add the map annotations?
Circle markers with site name pop-up and icon	To locate launch sites on map, to understand the factors affecting selection of launch site location and to understand any relationship between launch site and the mission, landing outcomes.
Marker clusters	To easily identify rates of successful and failed landings by launch sites and understand their relationship with landing success.
Mouse position	To mark down proximity location coordinates around launch sites, their distances from the sites and to understand the characteristics of launch sites.
Polyline	To easily identify and compare relative distances between proximity locations and launch sites.

GitHub URL of the completed interactive map with Folium notebook:
https://github.com/ParagHarnai/Care/blob/main/lab_jupyter_launch_site_location.jupyterlite.ipynb

Dashboard with Plotly Dash

Interactive visual analytics real time, on Space X launch data related to landing

Plot\chart and Interactions	**Why did we create them?**
Drop-down with Launch sites	For user, to select: -- Individual launch site. with default selection of ALL sites.
Pie chart	Selected in drop-down: -- With ALL sites, to visualize the share of successful landings by site. -- With individual site, to visualize the share of successful and failed landings.
Range slider	For user, to select a range of Payload Mass (kg)
Scatter plot	For sites selected in drop-down: -- With ALL sites, to visualize the correlation between Payload Mass and landing success. -- With individual site and payload mass range, to visualize the correlation between the selected range of Payload Mass and landing success for the selected launch site.

Dashboard with Plotly Dash…continued

GitHub URL of the completed Interactive Dashboard using Plotly Dash lab

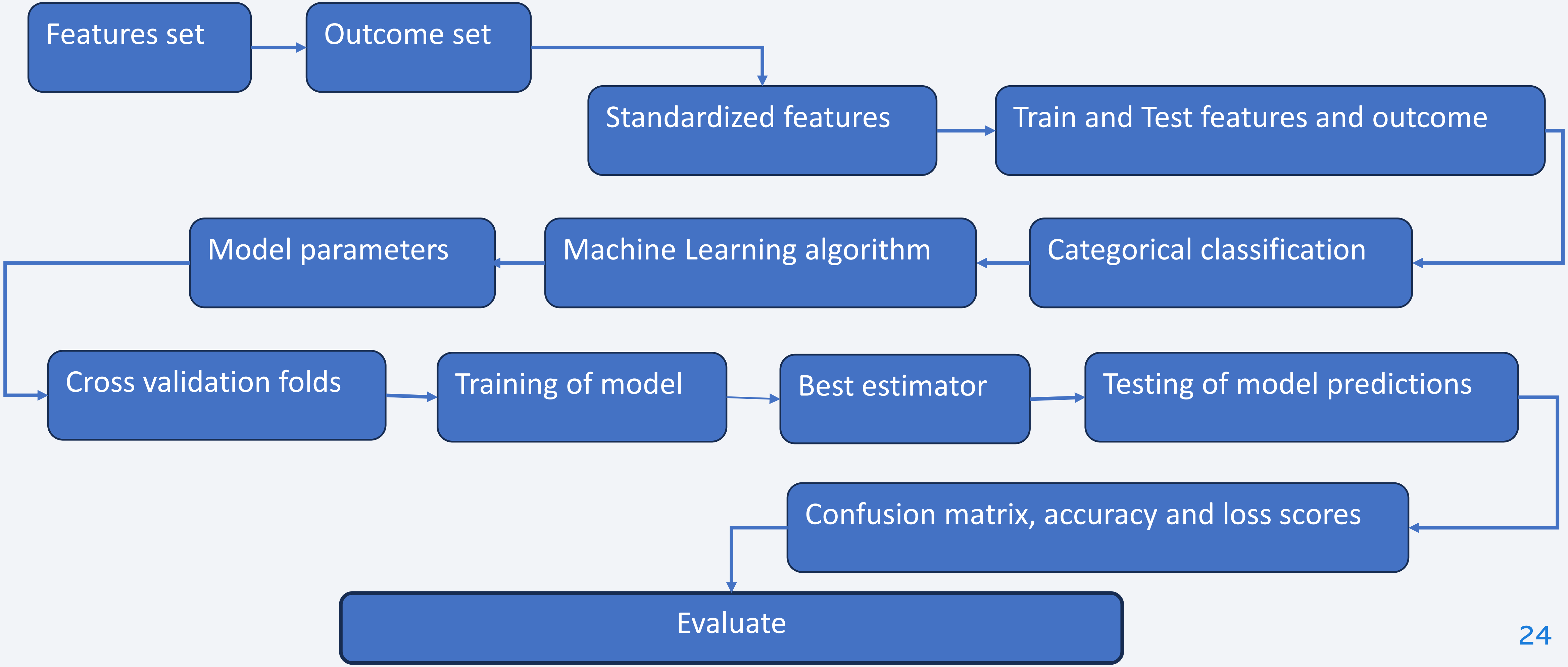
https://github.com/ParagHarnai/Care/blob/main/spacex_dash_app.py

Predictive Analysis (Classification)

- Separate fields into ‘features’ and ‘landing outcome’ data sets.
- Preprocess the data sets by transforming them using Standard Scaler.
- Split the features and outcome data sets into 80% train and 20% test.
- Select machine learning algorithm for categorical classification.
- Select the algorithm specific parameters, to fine tune it.
- For better accuracy with limited amount of data, use cross validation folds in data sets.
- Train the model for different algorithms such as Logistic regression, Support Vector Machine, Decision Tree, and K nearest neighbor, with cross validation folds of 10 and using Train features data set.
- Using the best estimator (parameters), predict the outcome using Test features set.
- Create confusion matrix, to evaluate the accuracy performance of different models.
- Calculate and compare accuracy and loss scores of different models and select the one with best performance.

Predictive Analysis (Classification)…continued

Process Flow chart

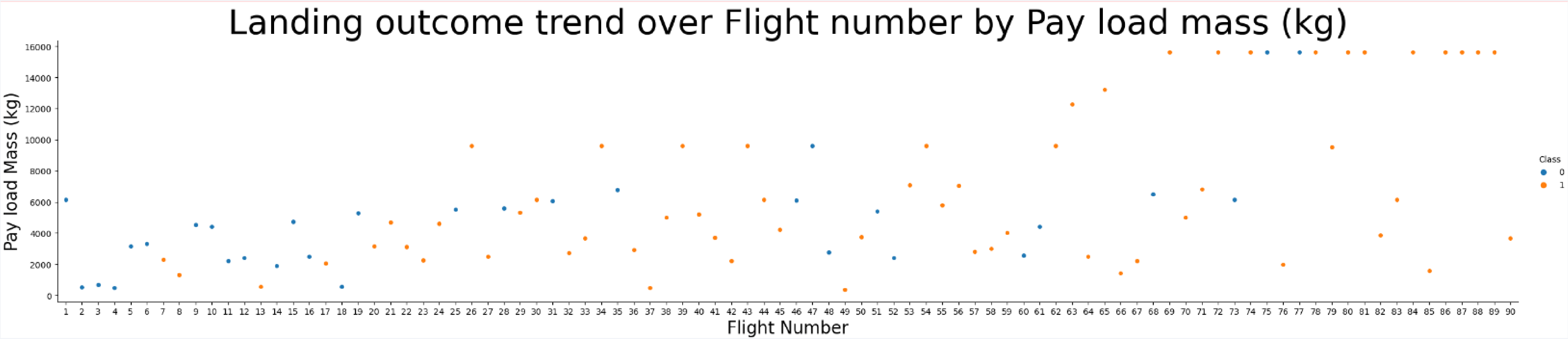


Predictive Analysis (Classification)…continued

GitHub URL of the completed Predictive Analysis lab

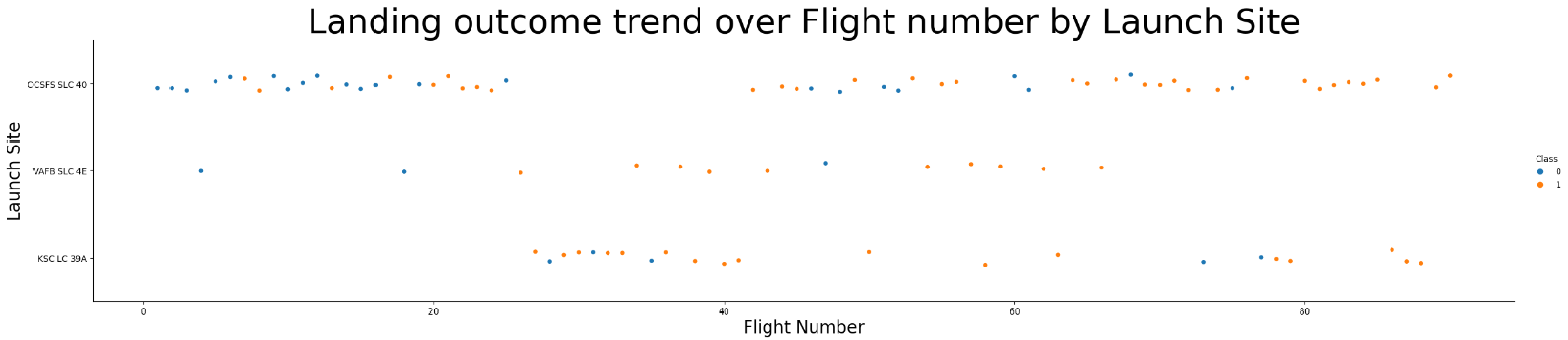
https://github.com/ParagHarnai/Care/blob/main/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

Results – Exploratory data analysis



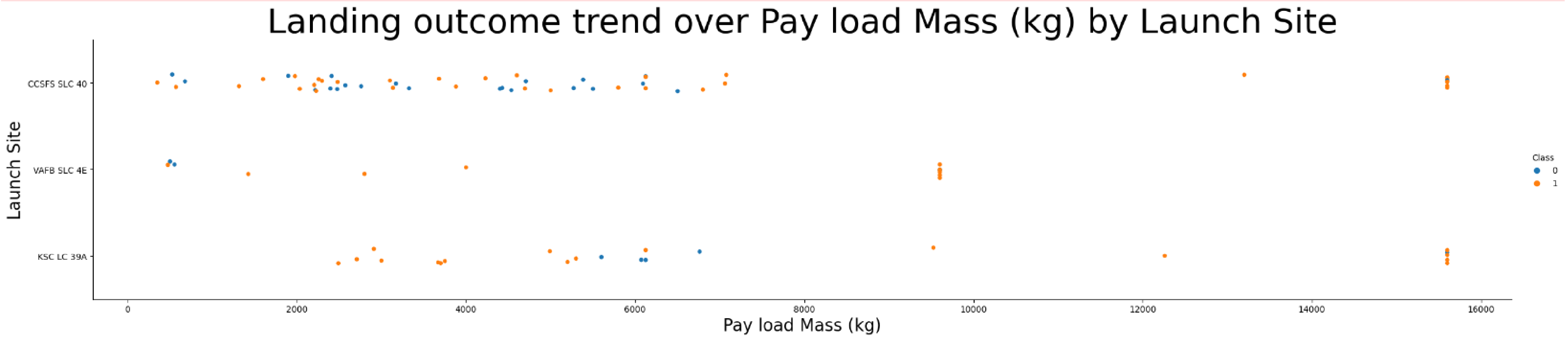
Correlation between Flights over time, Payload mass and landing outcome.

Results – Exploratory data analysis



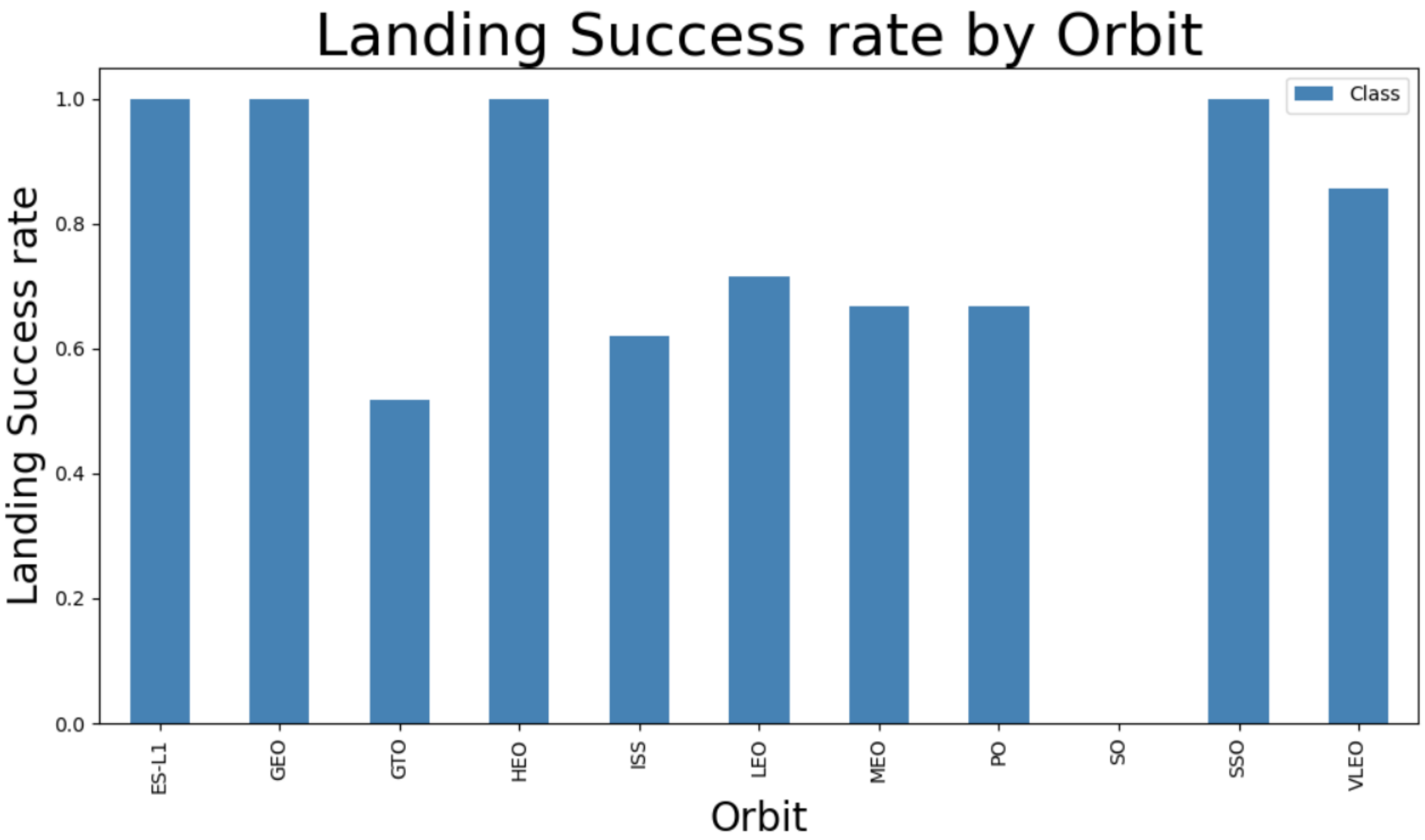
Correlation between Flights over time, Launch sites, and landing outcome.

Results – Exploratory data analysis



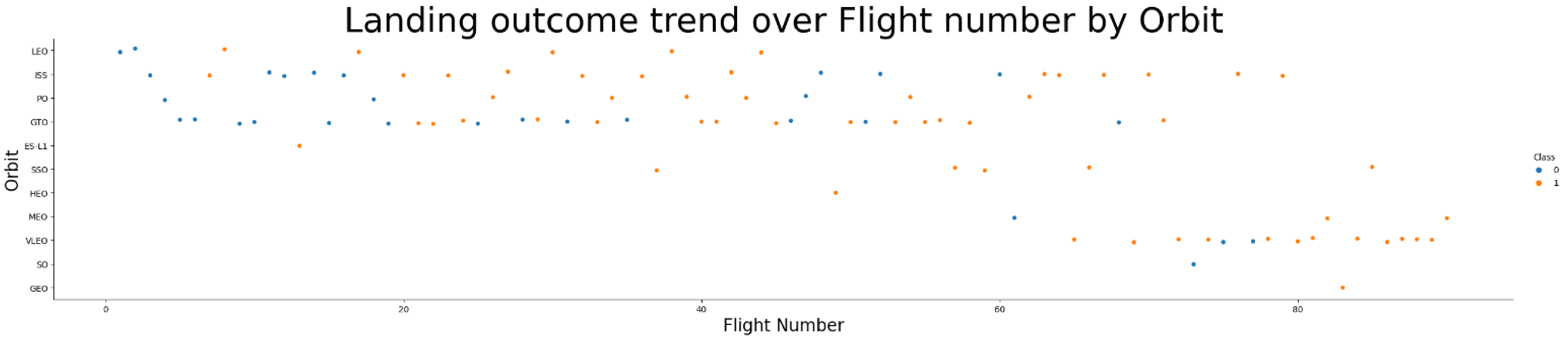
Correlation between Payload mass, Launch sites, and landing outcome.

Results – Exploratory data analysis



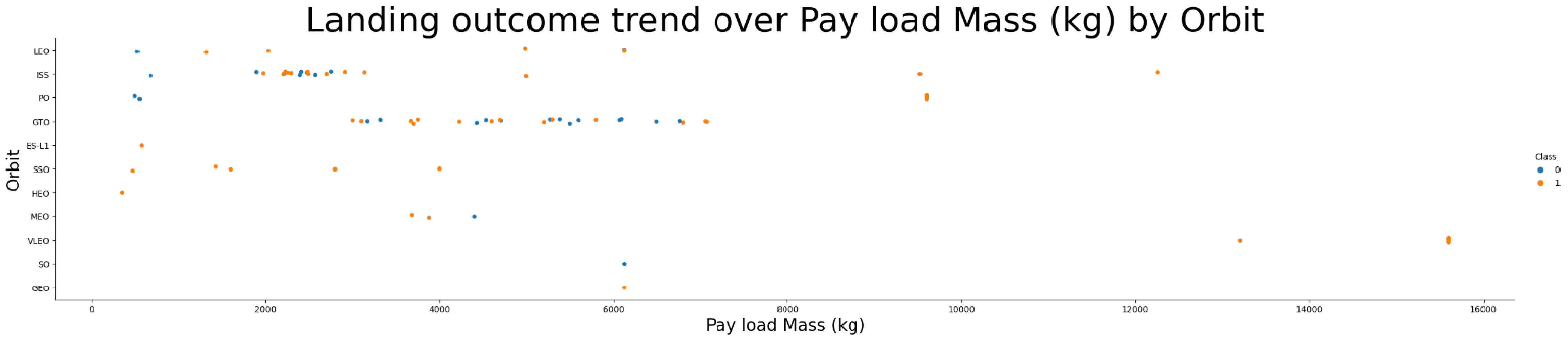
Correlation between Orbit type and Landing success rate.

Results – Exploratory data analysis



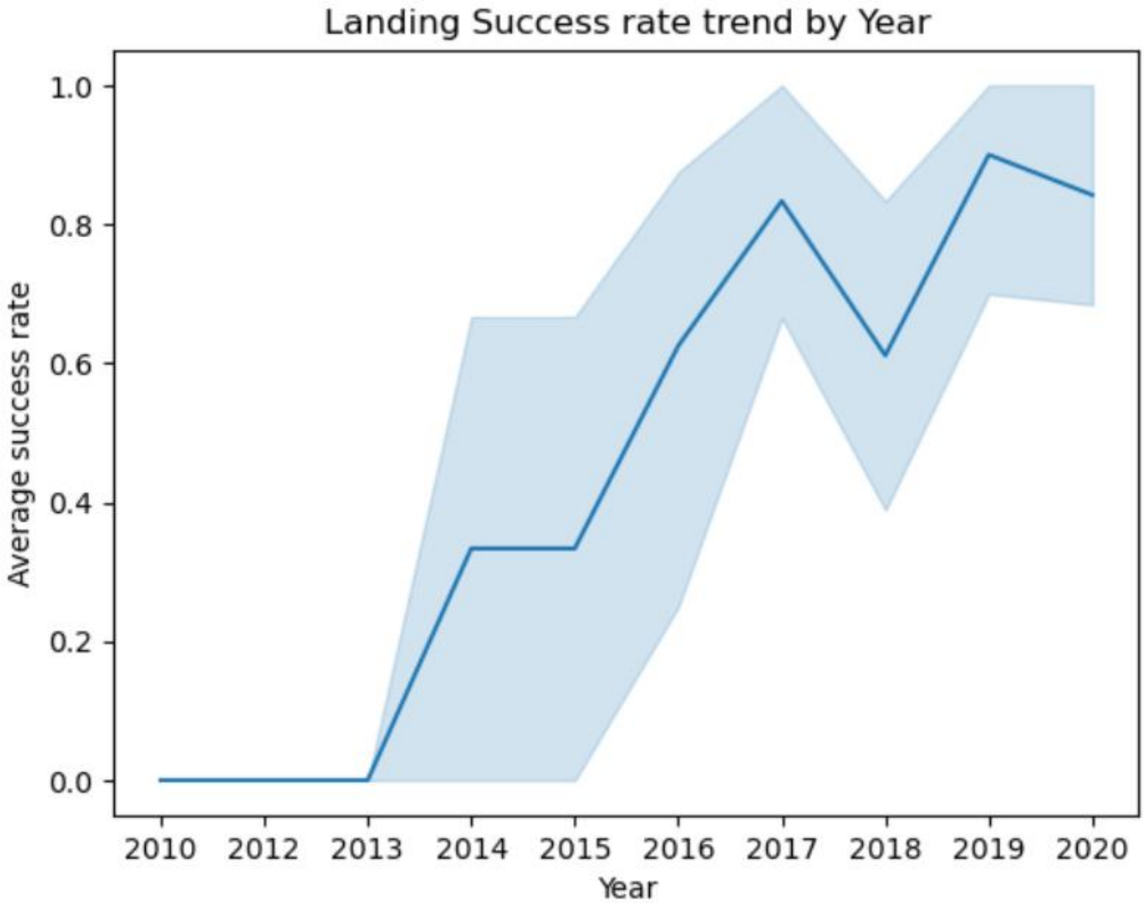
Correlation between Flights over time, Orbit type, and landing outcome.

Results – Exploratory data analysis



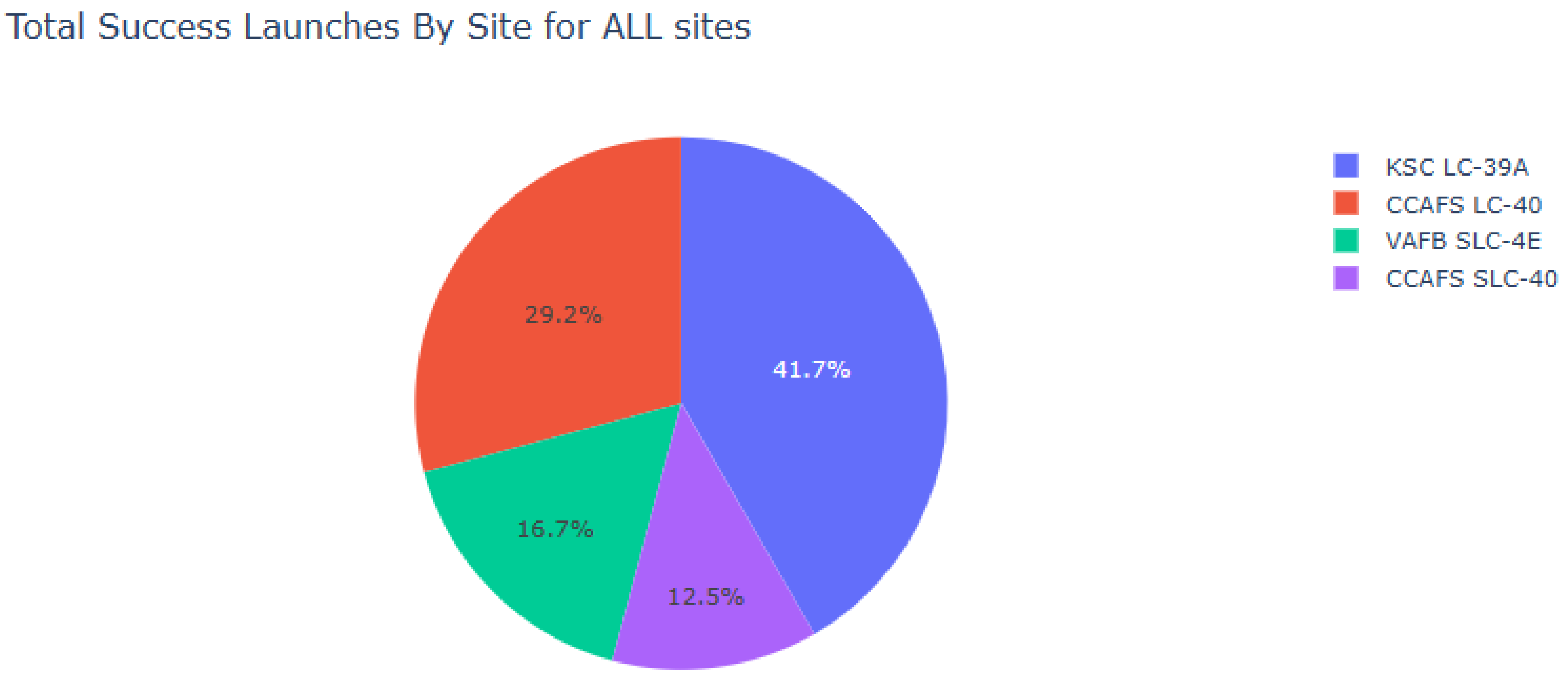
Correlation between Payload mass, Orbit type, and landing outcome.

Results – Exploratory data analysis

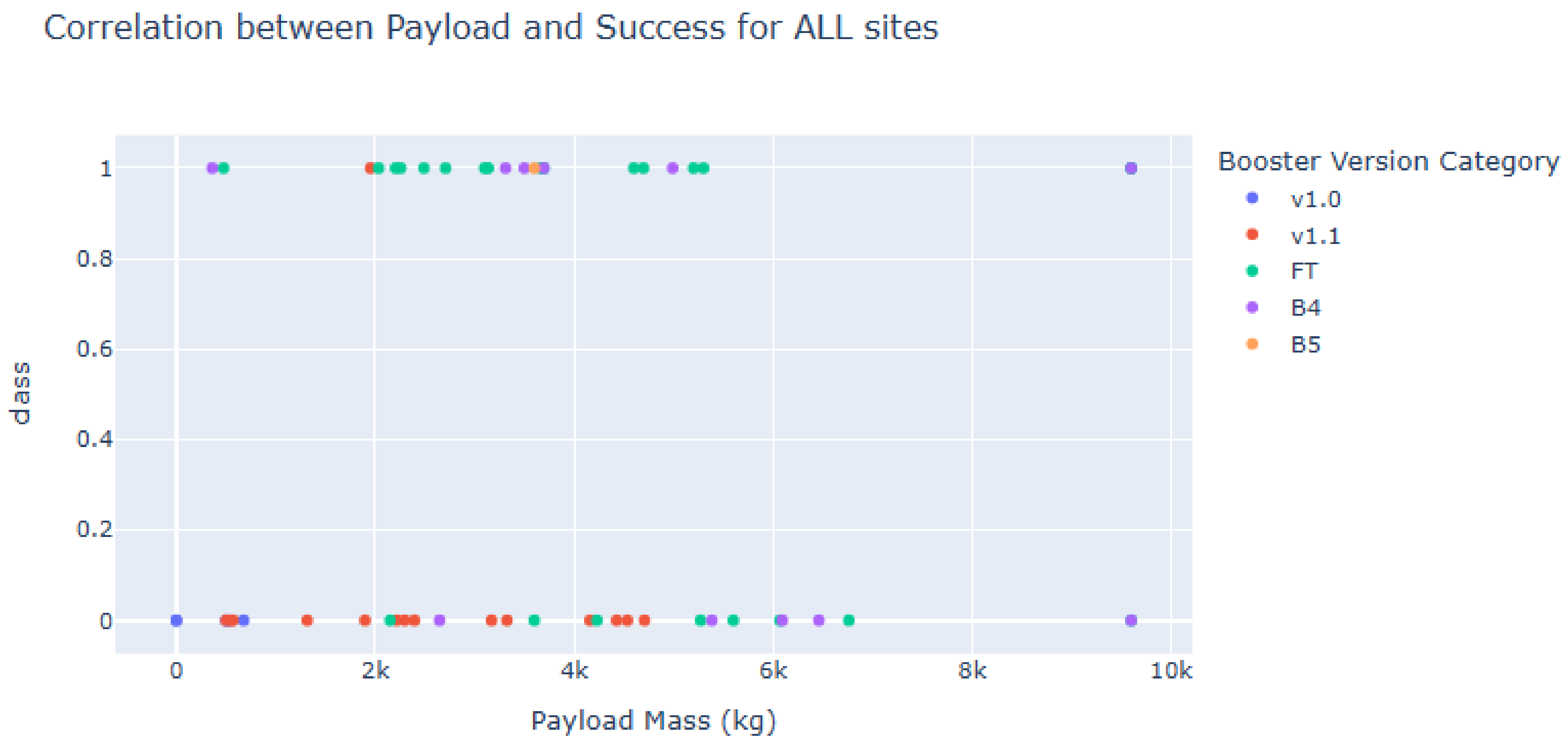


Trend of Average landing success rate between years 2010 and 2020.

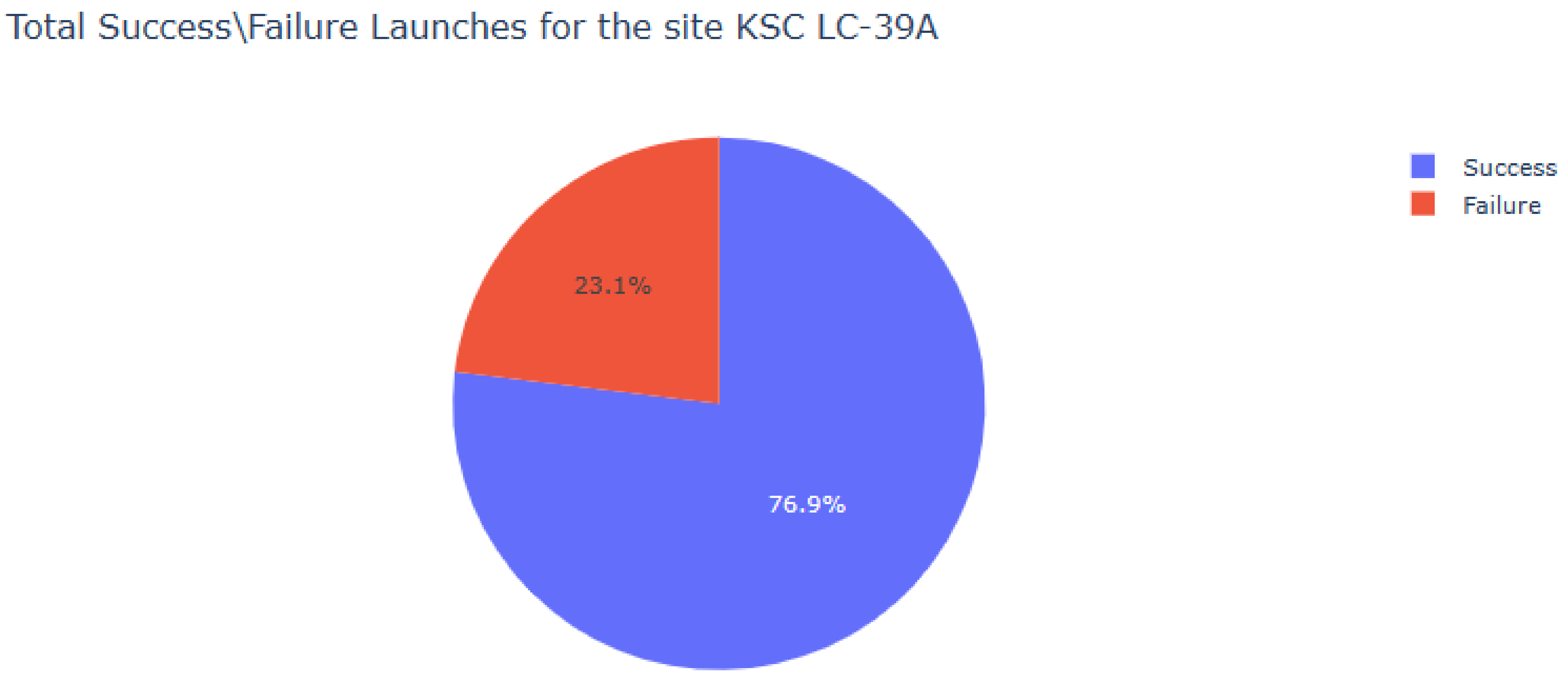
Results – Interactive data analytics - all sites together



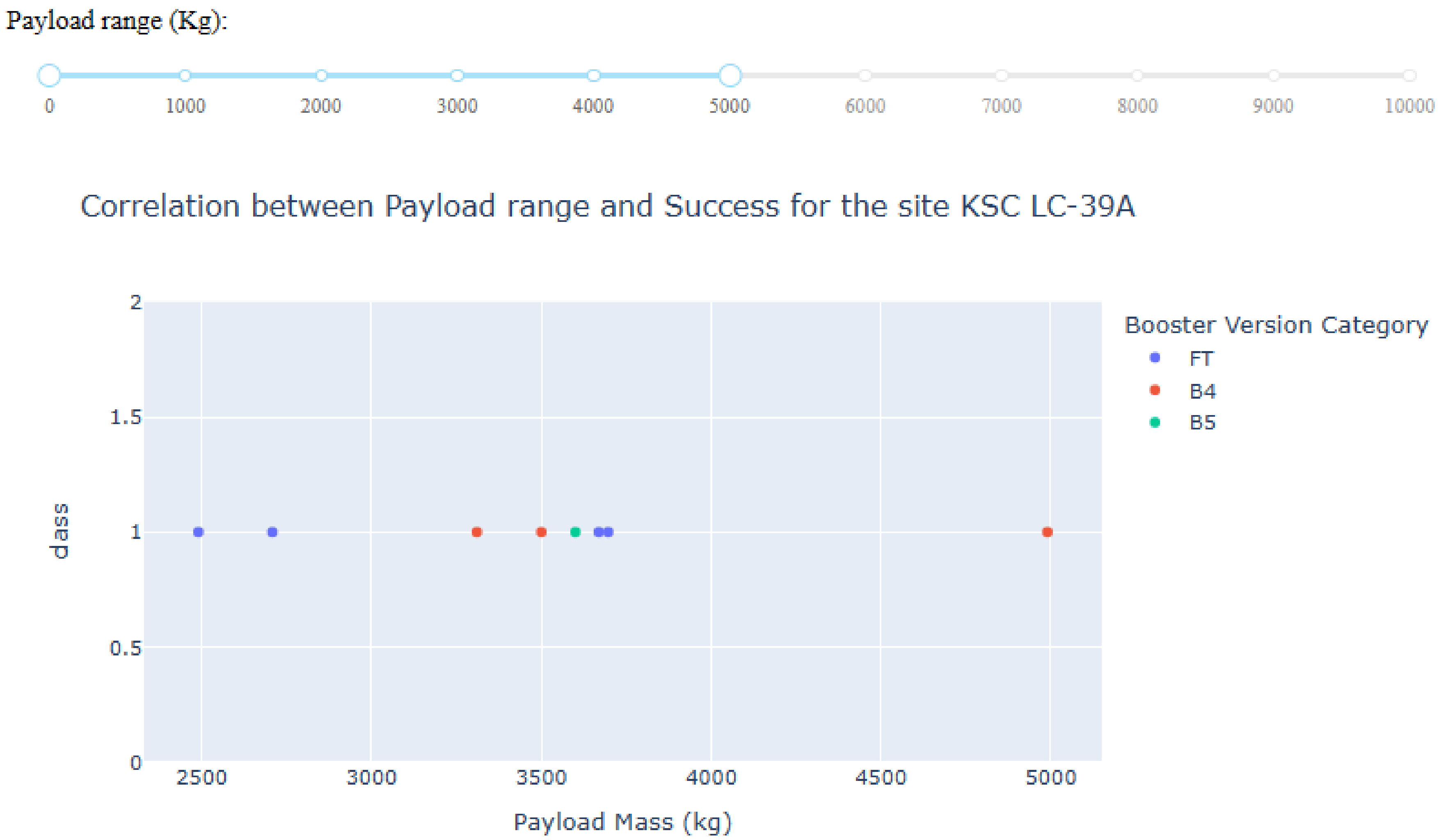
Results – Interactive data analytics - all sites together



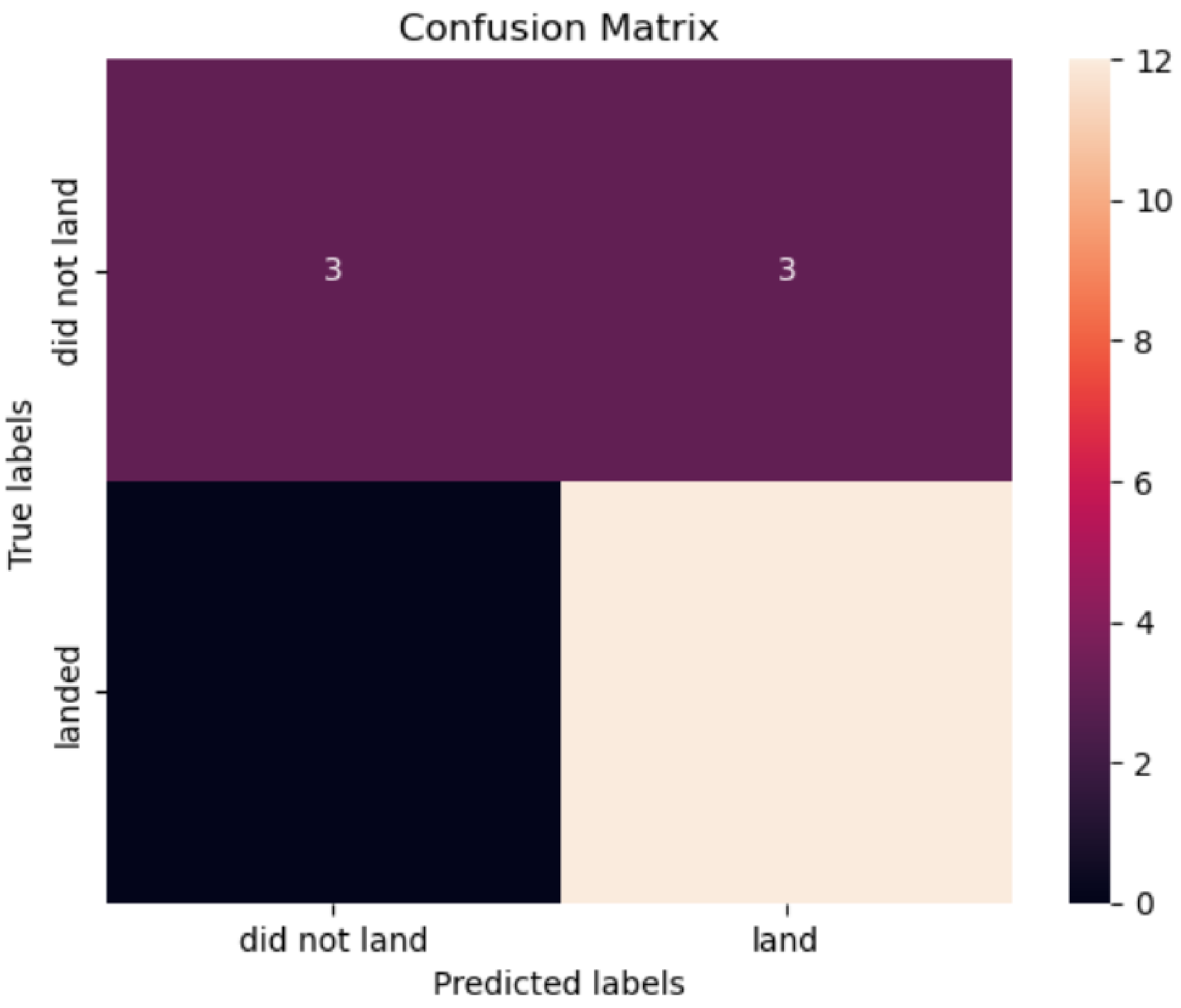
Results – Interactive data analytics – by a site



Results – Interactive data analytics – by a site and payload



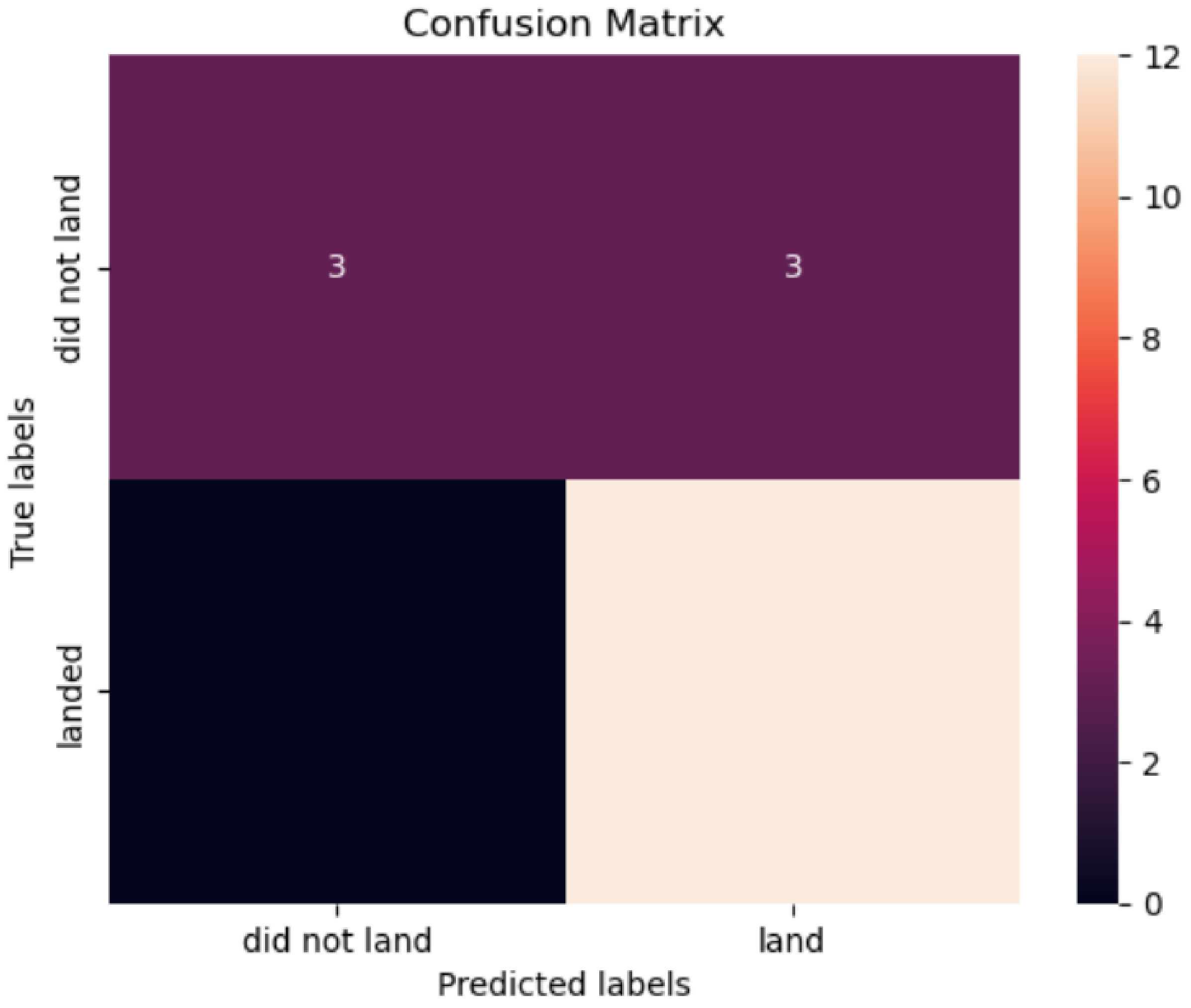
Results – Predictive Analysis with Logistic Regression



Accuracy Score: 0.83

False positives are a concern

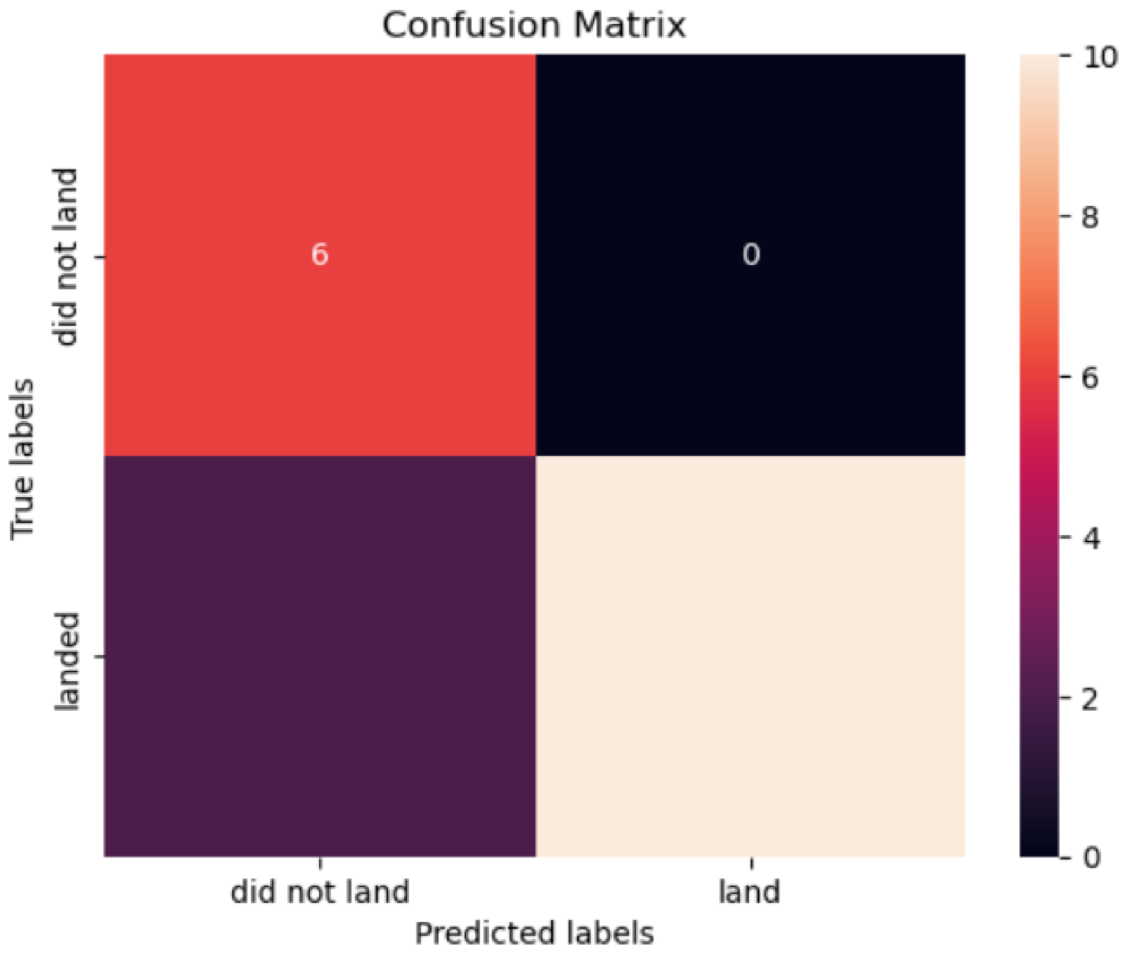
Results – Predictive Analysis with Support Vector Machine



Accuracy Score: 0.83

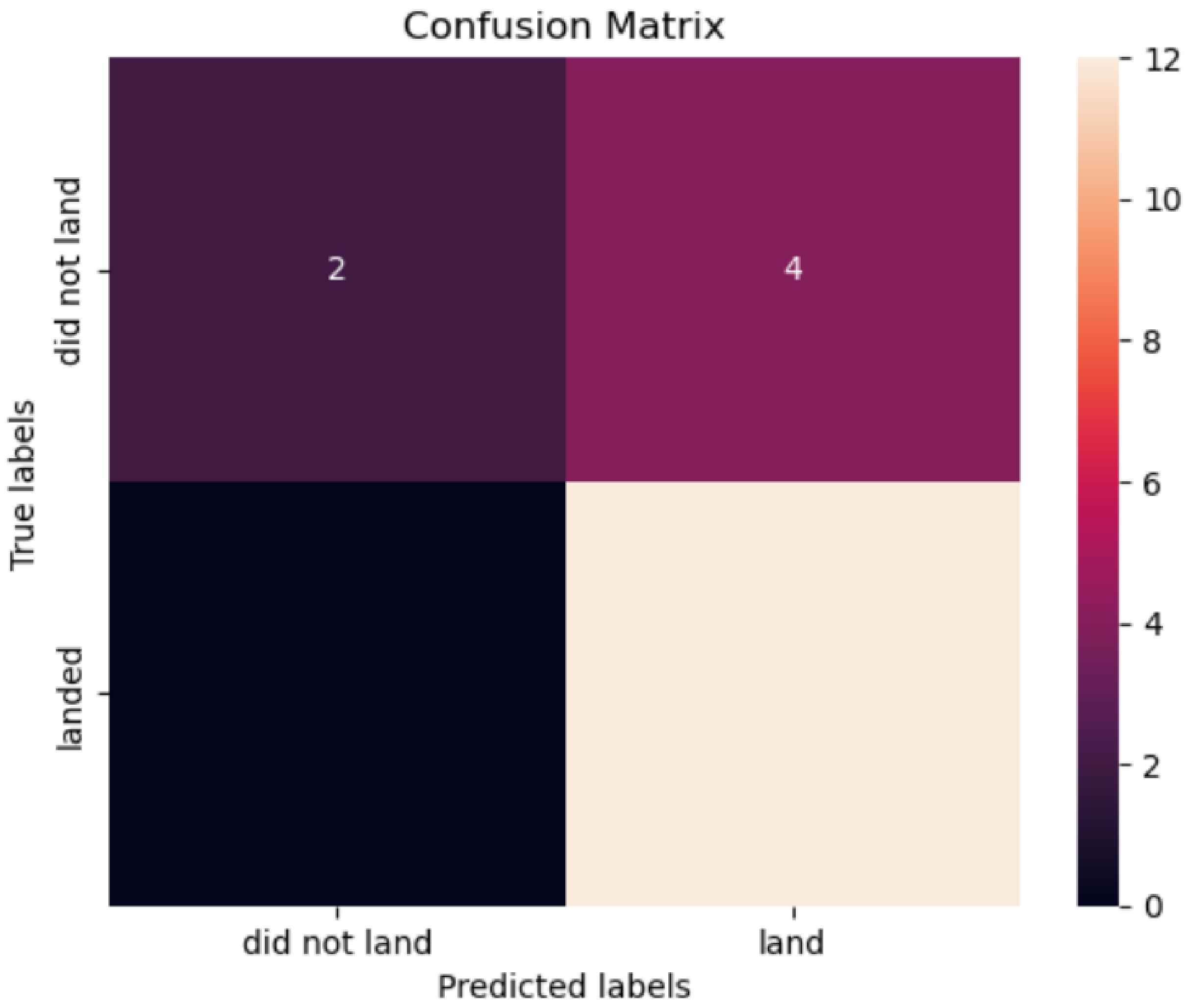
False positives are a concern

Results – Predictive Analysis with Decision Tree



Accuracy Score: 0.89

Results – Predictive Analysis with K Nearest Neighbor



Accuracy Score: 0.78

False positives are a concern

Results – Predictive Analysis comparison

	KNN_Metrics	Tree_Metrics	LR_Metrics	SVM_Metrics
Accuracy_Score	0.777778	0.888889	0.833333	0.833333
Jaccard_Index	0.333333	0.750000	0.500000	0.500000
F1_Score	0.738095	0.891775	0.814815	0.814815
Log_Loss	0.000000	0.000000	0.394355	0.000000

K Nearest Neighbour

Decision Tree

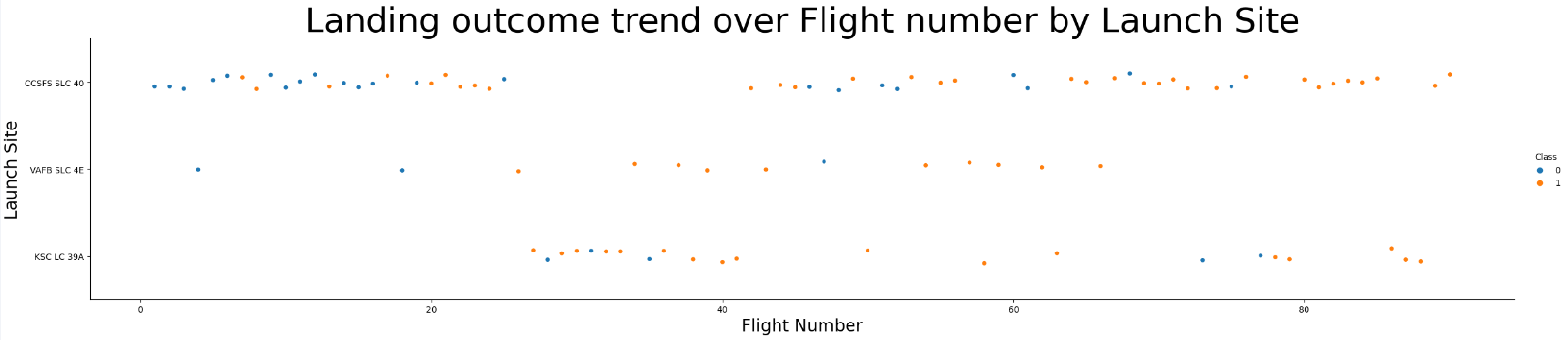
Logistic Regression

Support Vector Machine

Section 2

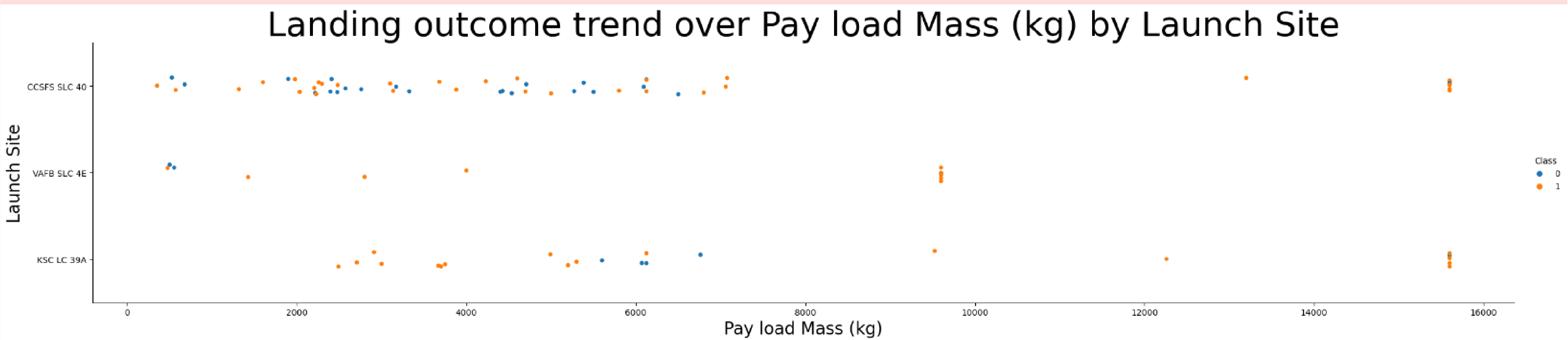
Insights drawn from EDA

Flight Number vs. Launch Site



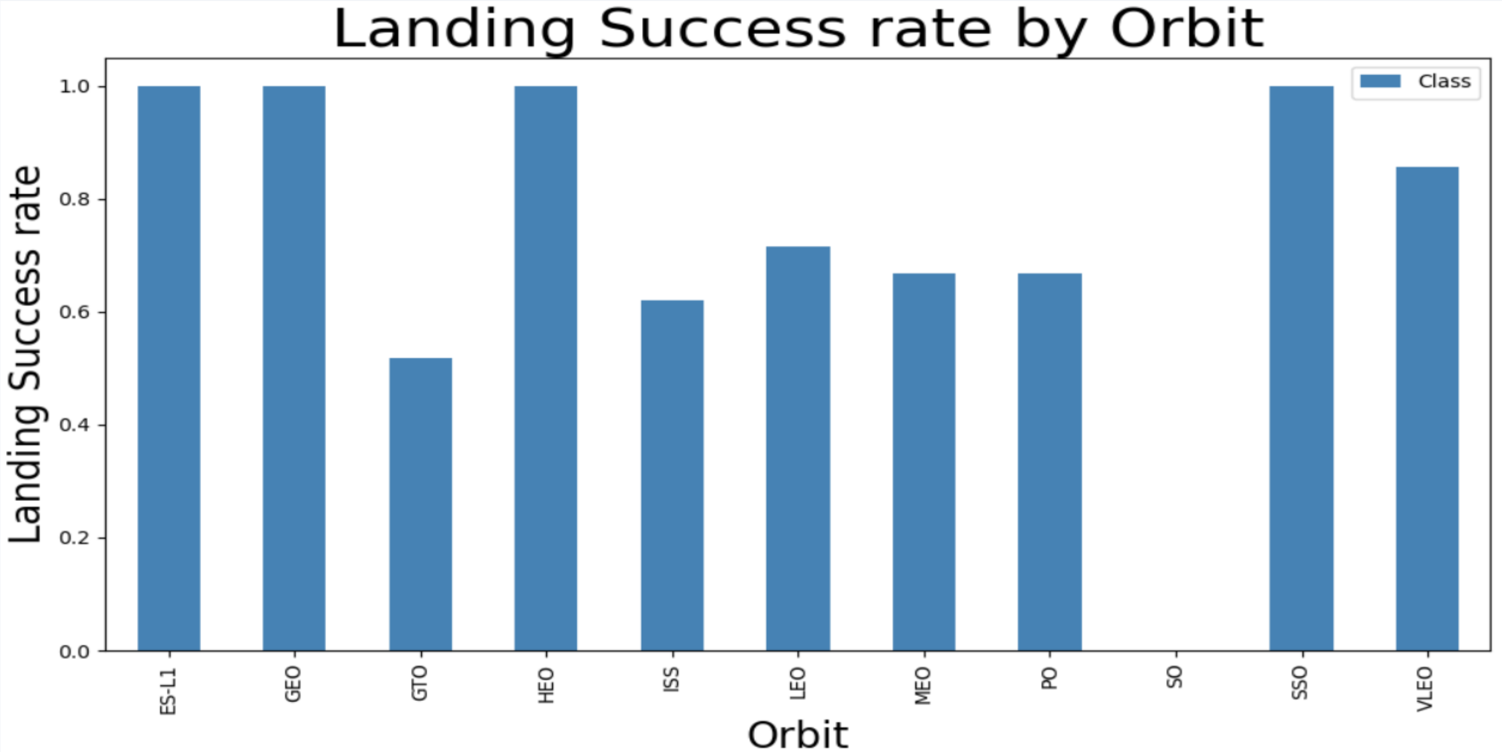
- Launch site 'CCAFS LC-40' has highest absolute number of flights.
- Followed by launch site 'KSC LC-39A'.
- Launch site 'VAFB SLC 4E' has lowest absolute number of flights.
- Launch site CCAFS LC-40 has lower rate of landing success compared to the other two.

Payload vs. Launch Site



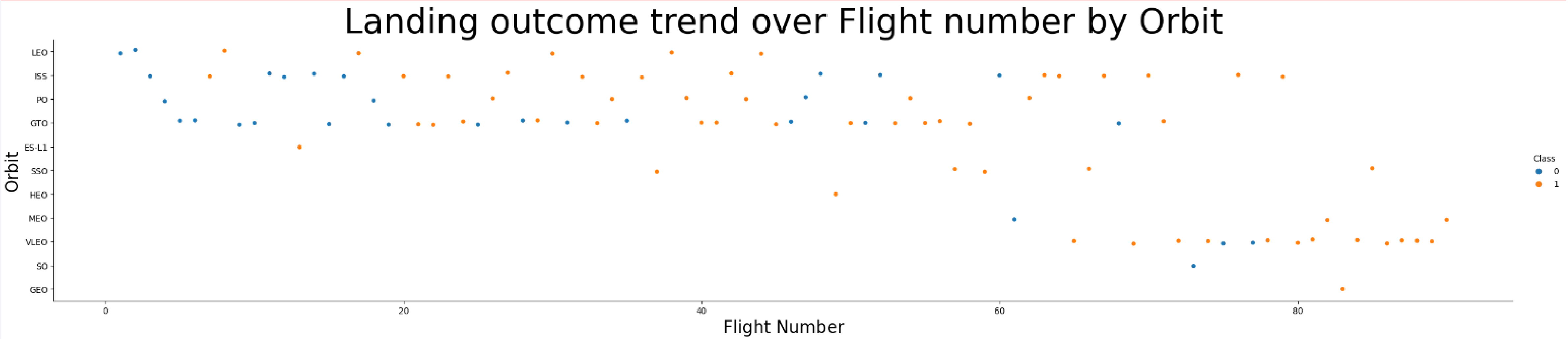
- ➢ Launch site VAFB-SLC has no rockets launched for heavy payload mass (greater than 10000).
- ➢ A very small percentage of rockets launched heavy payload mass for the other two launch sites.

Success Rate vs. Orbit Type



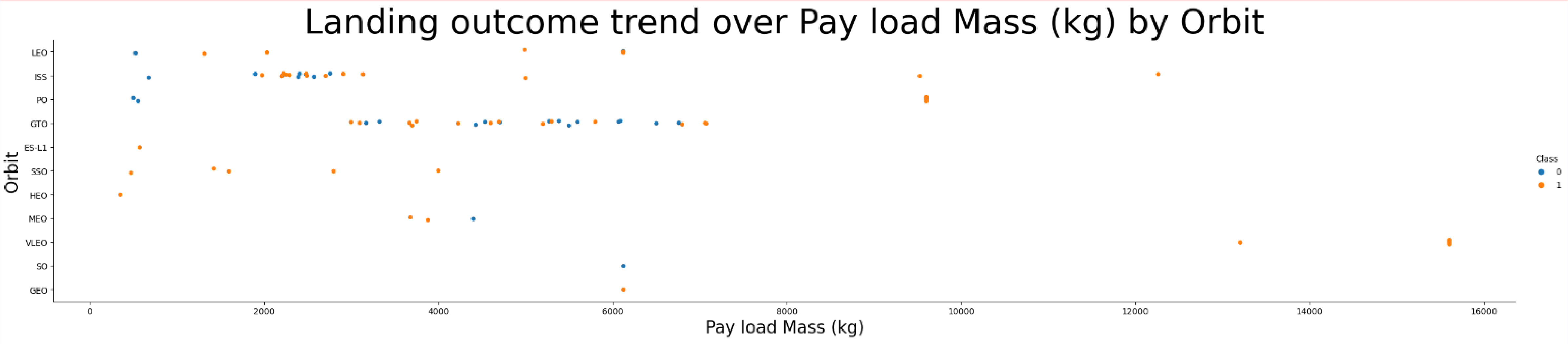
- Orbit types ESL1, GEO, HEO, and SSO have the highest landing success rate.
- Followed by orbit types VLEO and LEO.

Flight Number vs. Orbit Type



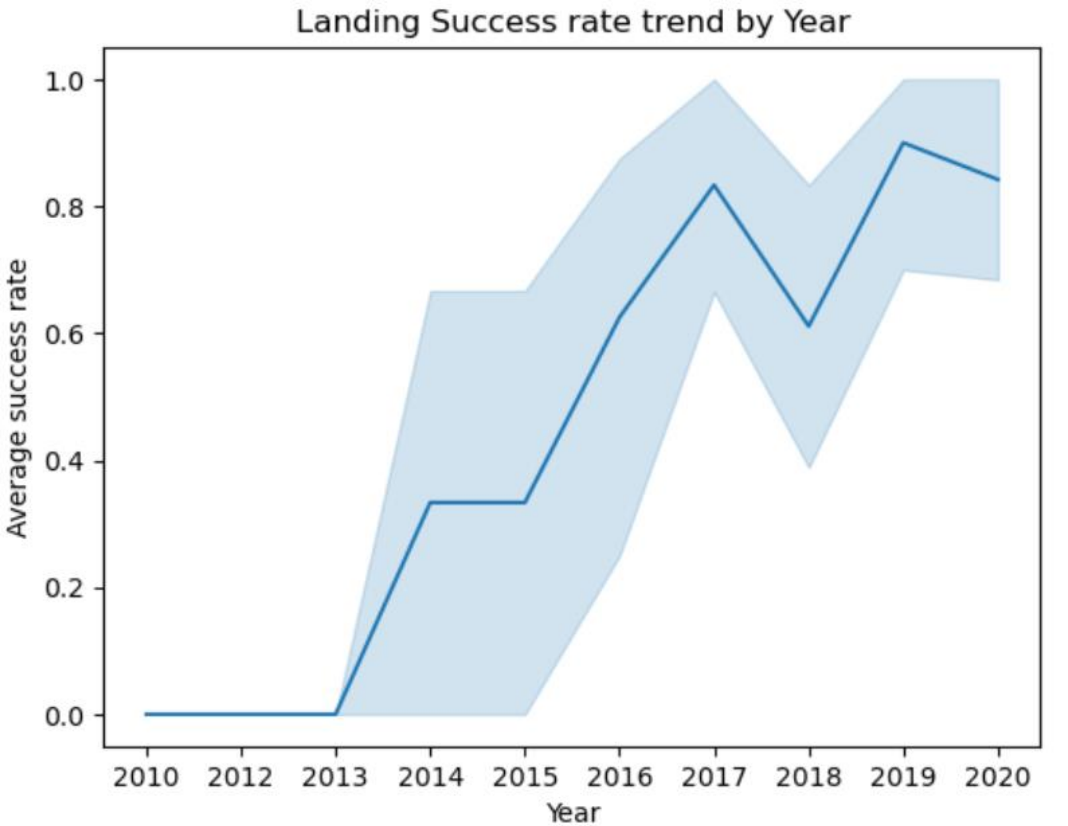
- ➢In the LEO orbit type, the landing success appears to related to the number of flights.
- ➢On the other hand, there seems to be no relationship between flight number when the flight is in GTO and all other orbit types.

Payload vs. Orbit Type



- ➢ With heavy payloads, successful landing or positive landing rates are more for Polar, LEO and ISS orbit types.
- ➢ However for GTO, we cannot distinguish this well as both positive landing rate and negative landing (unsuccessful mission) are both present here.

Launch Success Yearly Trend



Average success rate of landing has been going up since 2013 till 2020.

All Launch Site Names

```
%%sql
select "Launch_Site" from SPACEXTABLE group by "Launch_Site"
```

Running query in 'sqlite:///my_data1.db'

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

Falcon 9 launches have used the four unique sites between years 2013 and 2020:

CCAFS LC-40, CCAFS SLC-40, KSC LC-39A, and VAFB SLC-4E.

Launch Site Names Begin with 'CCA'

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

Running query in 'sqlite:///my_data1.db'

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

Five launches from the site beginning with ‘CCA’ in the name, between 2010 and 2013, have no landing success.

Total Payload Mass

```
%%sql
select "Customer", sum("PAYLOAD_MASS__KG_") as "total_payload_mass" from SPACEXTABLE where "Customer" = 'NASA (CRS)' group by "Customer"
```

Running query in 'sqlite:///my_data1.db'

Customer	total_payload_mass
NASA (CRS)	45596

Total of Payload Mass of all the Falcon 9 launches for NASA (CRS) (one of the customers) is 45596 kg.

Average Payload Mass by F9 v1.1

```
%%sql
select "Booster_Version", avg("PAYLOAD_MASS__KG_") as "average_payload_mass (kg)" from SPACEXTABLE where "Booster_Version" = 'F9 v1.1' group by "Booster
```

Running query in 'sqlite:///my_data1.db'

Booster_Version	average_payload_mass (kg)
F9 v1.1	2928.4

Average of all the Payload Mass launched by Falcon 9 booster version F9 v1.1 is 2928.4 kg.

First Successful Ground Landing Date

```
%%sql
select "Landing_Outcome", min("Date") as "date_first_successful_landing (CCYY-MM-DD)" from SPACEXTABLE where "Landing_Outcome" like '%ground pad%' group
```

Running query in 'sqlite:///my_data1.db'

Landing_Outcome	date_first_successful_landing (CCYY-MM-DD)
Success (ground pad)	2015-12-22

The date when Falcon 9 booster first landed back on a ground pad successfully is 2015, December 22nd.

Successful Drone Ship Landing with Payload mass between 4000 and 6000 kg

```
%%sql
select "Booster_Version" from SPACEXTABLE where "Landing_Outcome" = 'Success (drone ship)' and ("PAYLOAD_MASS__KG_" > 4000 and "PAYLOAD_MASS__KG_" < 6000
```

Running query in 'sqlite:///my_data1.db'

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

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

F9 FT 81022, F9 FT 81026, F9 FT 81021.2, F9 FT 81031.2

Total Number of Successful and Failure Mission Outcomes

```
%%sql
select Case when "Mission_Outcome" like '%Success%' then 'Success' else 'Failure' end as "Mission_Outcome", Count(*) as "total_number" from SPACEXTABLE g
```

Running query in 'sqlite:///my_data1.db'

Mission_Outcome	total_number
Failure	1
Success	100

Total number of successful (100) and failed (1) mission outcomes.

Boosters Carried Maximum Payload

```
%%sql
select "Booster_Version", "PAYLOAD_MASS__KG_" from SPACEXTABLE where "PAYLOAD_MASS__KG_" = (select max("PAYLOAD_MASS__KG_") from SPACEXTABLE) group by "E
```

Running query in 'sqlite:///my_data1.db'

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

The list of names of the booster_versions which have carried the maximum payload mass, which is 15600 kg.

2015 Launch Records

```
%%sql
select Case substr("Date", 6, 2) When '01' then 'Jan' When '02' then 'Feb' When '03' then 'Mar' When '04' then 'Apr' When '05' then 'May' When '06' then
```

Running query in 'sqlite:///my_data1.db'

Month	Year	Landing_Outcome	Booster_Version	Launch_Site
Jan	2015	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
Apr	2015	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

List of the failed outcomes of landing on drone ship, their booster versions, and launch site names from the months in year 2015.

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

```
%%sql
select "Landing_Outcome", count(*) as "count_of_landing_outcome" from SPACEXTABLE where "Date" between '2010-06-04' and '2017-03-20' group by "Landing_Ou
```

Running query in 'sqlite:///my_data1.db'

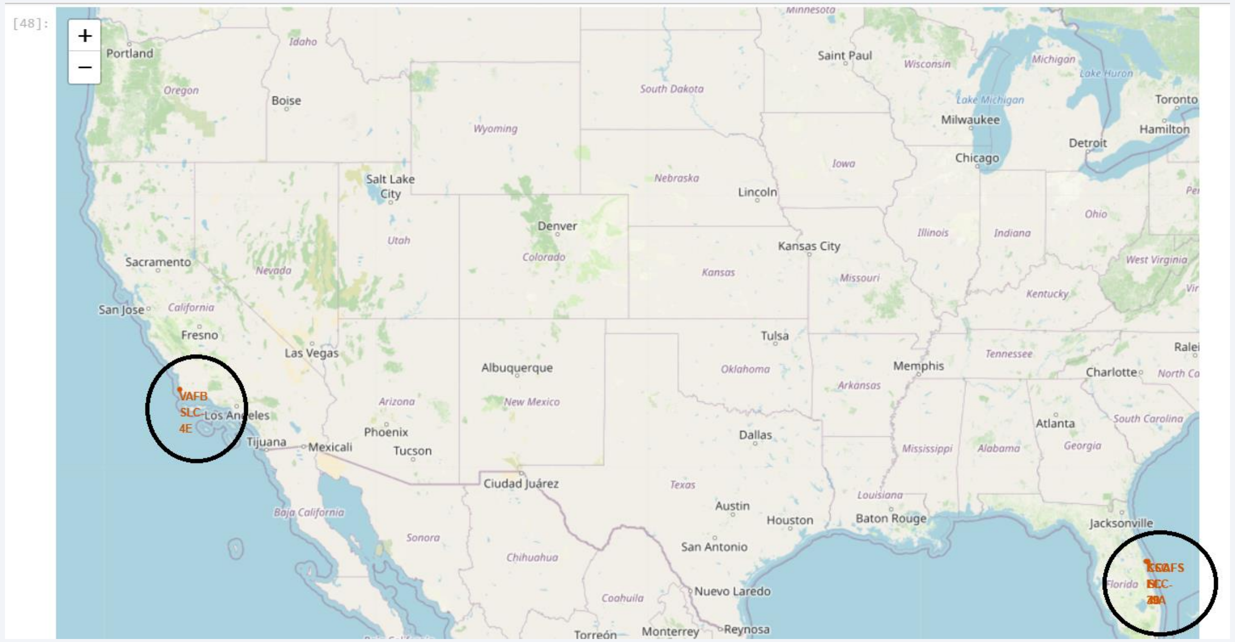
Landing_Outcome	count_of_landing_outcome
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

Count of unique landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the dates 2010-06-04 and 2017-03-20, sorted in descending order.

Section 3

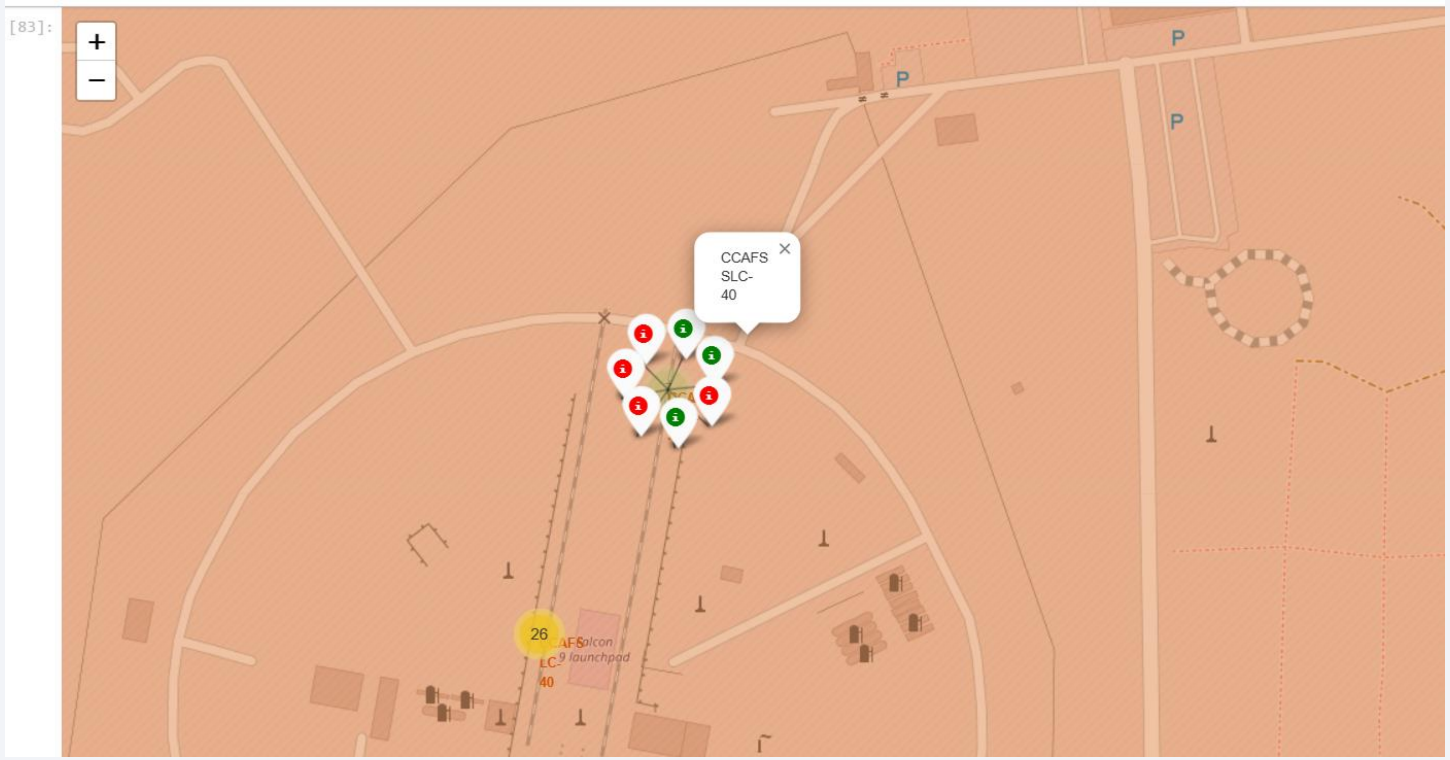
Launch Sites Proximities Analysis

Launch sites on a map – discover and analyze proximity factors



- ✓ Location of site plays an important role in the success of launch. Initial position and trajectory of the launch, transport requirements, surroundings and safety are among many factors that drive the selection of launch site.
- ✓ Launch sites are closer to Equator line because the launch then can take the advantage of slightly less gravitational force. It can carry the launch into the orbit with less fuel.
- ✓ Launch sites are also in close proximity to coast area because the launch is then closer to water area. If the launch fails it is safer near water with fire and debris.

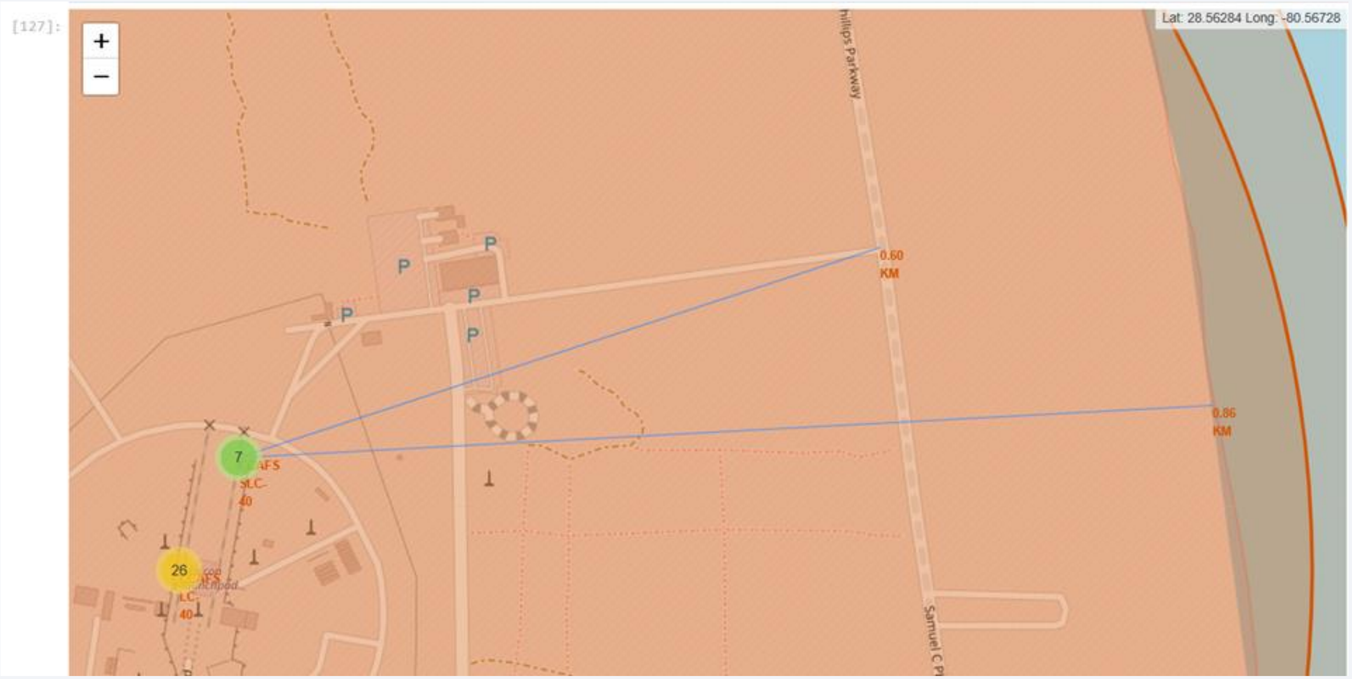
Interactive success rate visualization of launch sites



✓Launches at each site are marked in a cluster using color codes.

✓It simplifies the visualization of success rate by site, right on the map itself.

Discovery of proximities and the distances from launch sites



✓ Mouse Position is added to mark down the coordinates of proximity location points.

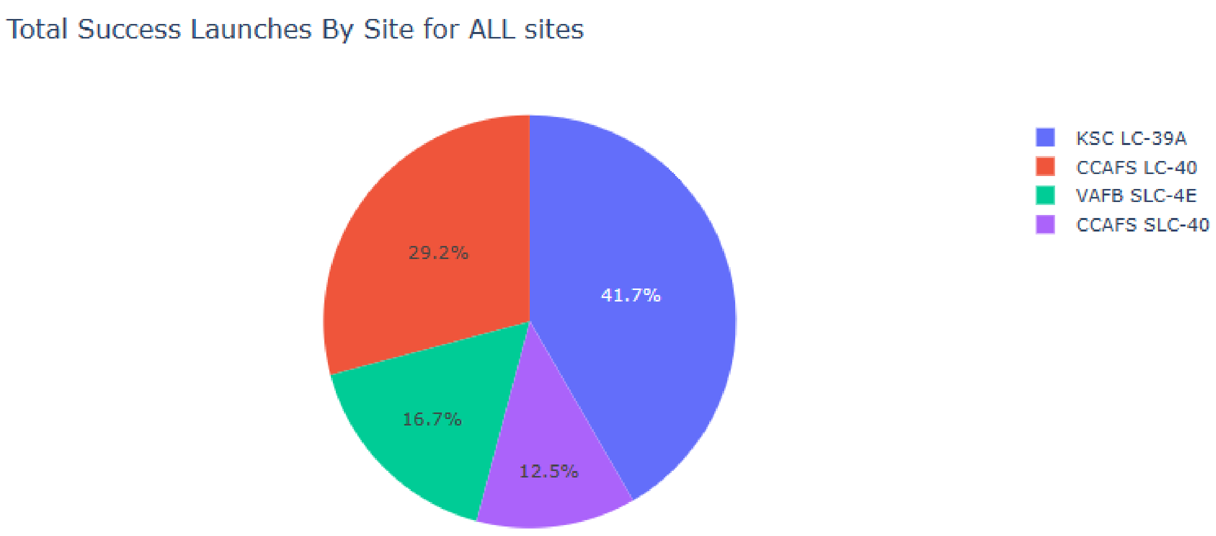
✓ Distance marker helps easily see the distance between launch site and proximity point.

✓ Polyline helps easily identify proximity points which are being analyzed.

Section 4

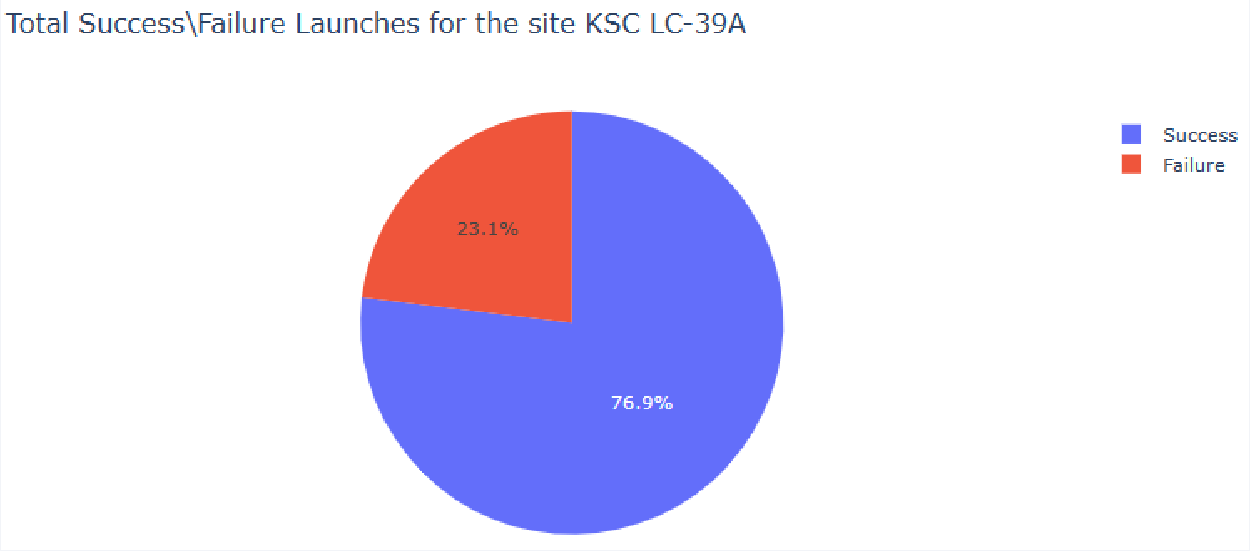
Build a Dashboard with Plotly Dash

Interactive proportion of total success by sites



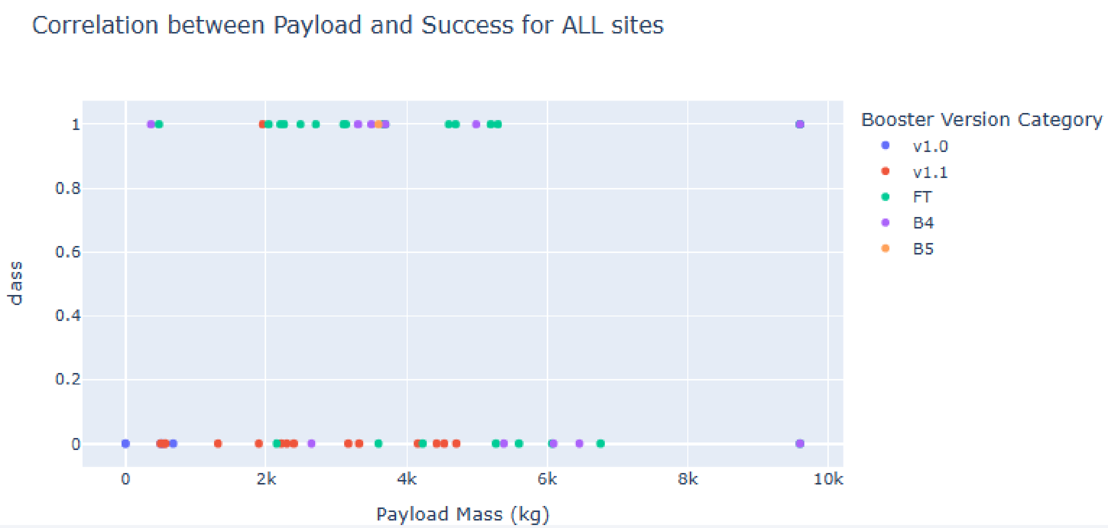
- Pie chart displays proportion of total success by sites.
- In top-right corner, legends show the names of launch sites.
- The order of legends follows proportions from highest to lowest.
- KSC LC-39A site has the highest success portion of 41.7%.
- CCAFS SLC-40 site has the lowest success portion of 12.5%.

Interactive proportion success and failure for a selected site



- Pie chart displays proportion of success and failure for the site (KSC LC-39A) with highest share of total success.
- In top-right corner, legends show the colors of Success and Failure portions.
- The order of legends follows proportions from highest to lowest.
- KSC LC-39A site has success and failure portions of 76.9% and 23.1% respectively.

Rate of Booster Version success over the Payload

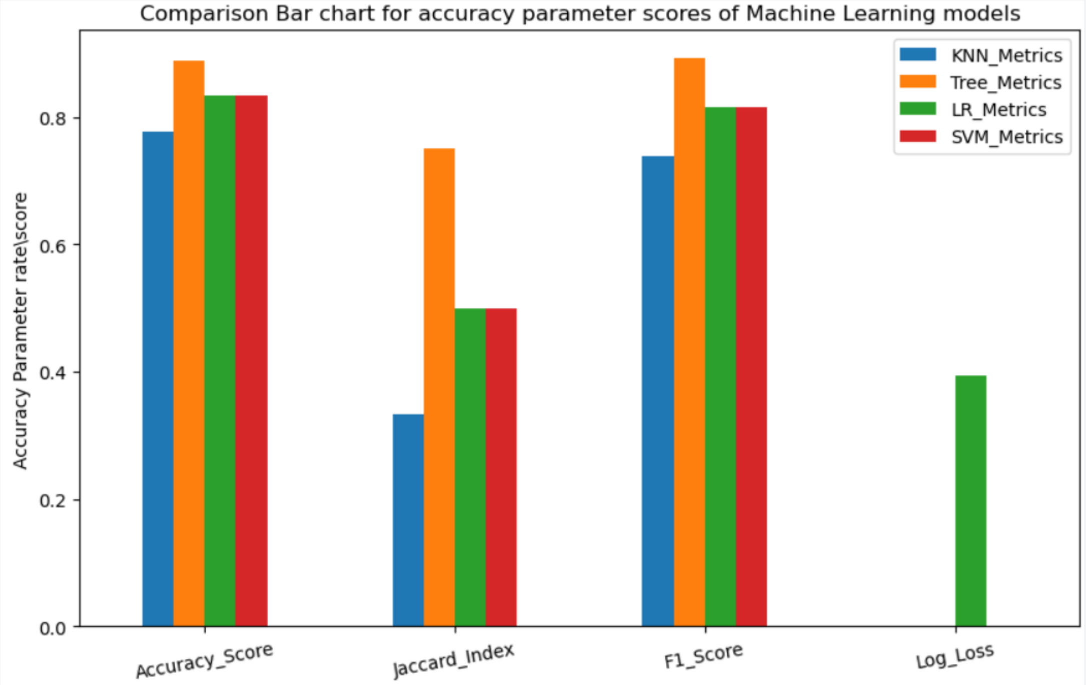


- ➢ Launch success of different Booster versions over Payload mass, across all sites. Here, entire Payload range is considered for all sites.
- ➢ Top right corner shows Booster version categories.
- ➢ Booster Version B5 has one success and no failure and hence has 100% success rate which is the highest.
- ➢ The next one in order is Booster version FT with success rate of 67%. FT is also the version used in highest number of launches, that is, 24.
- ➢ Version 1.0 had no success, so has the lowest success rate of 0%.

Section 5

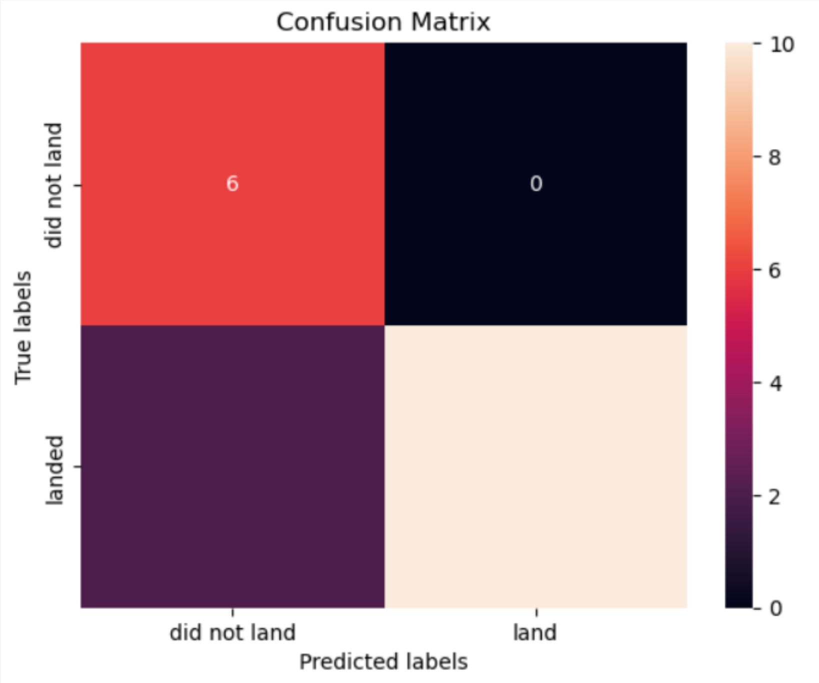
Predictive Analysis (Classification)

Classification Accuracy



- Logistic Regression and SVM methods provide similar accuracy, followed by KNN method.
- Decision Tree provides the highest of accuracy with least of confusion when classifying test set outcomes.
- Since the volume is not big in this case, Decision Tree is the best method.

Confusion Matrix



- ➢ Decision Tree has highest precision with zero False Positives (black square).
- ➢ There are no test data records for True Positives (peach square) and False Negatives (purple square).
- ➢ With 100% True Negatives (red square), Decision Tree also has highest accuracy.

Conclusions

- Decision tree has higher accuracy with both train as well as test data sets, compared to other models – logistic regression, support vector metrics, and K nearest neighbor.
- With small size of data set, Decision Tree performance should not be affected.
- Additional fine-tuning of the Decision Tree model could be explored through different parameters and Grid Search folds.
- New dummy Test Data set could be generated to test the accuracy of the model and to fine-tune it further.
- The deployed model can predict the landing outcome and price of the launch for Space Y to be competitive against Space X.

Future developments

- ❑This work could be further enhanced to predict the reuse of boosters.
- ❑Factors, affecting the reuse of successfully landed boosters, could be explored.
- ❑Factors, affecting the decision to abandon boosters which could actually be reused, could be explored.
- ❑Landing outcome combined with confirmed reuse of boosters could be used to more accurately predict the price of a launch.

Appendix

Data created out of Data Collection: https://github.com/ParagHarnai/Care/blob/main/dataset_part_1.csv

Data created out of Web Scraping: https://github.com/ParagHarnai/Care/blob/main/spacex_web_scraped.csv

Data created out of Data Wrangling: https://github.com/ParagHarnai/Care/blob/main/dataset_part_2.csv

Data created out Exploratory Data Visualization Analysis:
https://github.com/ParagHarnai/Care/blob/main/dataset_part_3.csv

Data used for Exploratory Data sql Analysis: https://github.com/ParagHarnai/Care/blob/main/Spacex.csv

Data created out of Exploratory Data sql Analysis:
https://github.com/ParagHarnai/Care/blob/main/my_data1.db

Data used for Interactive launch site Proximity Analysis:
https://github.com/ParagHarnai/Care/blob/main/spacex_launch_geo.csv

Plotly Dashboard analysis in Jupyter Notebook:
https://github.com/ParagHarnai/Care/blob/main/spacex_dash_app.ipynb

Flow-charts in MS spread sheet: https://github.com/ParagHarnai/Care/blob/main/Flow-chart.xlsx

Abbreviations and Acronyms

SQL → Structured Query Language

sql → Structured Query Language

EDA → Exploratory Data Analysis

csv → Comma Separated Values

URL → Uniform Resource Locator

API → Application Programming Interface

REST → REpresentational State Transfer

Thank you!