IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Rachael Hazell
26 March 2024

Outline

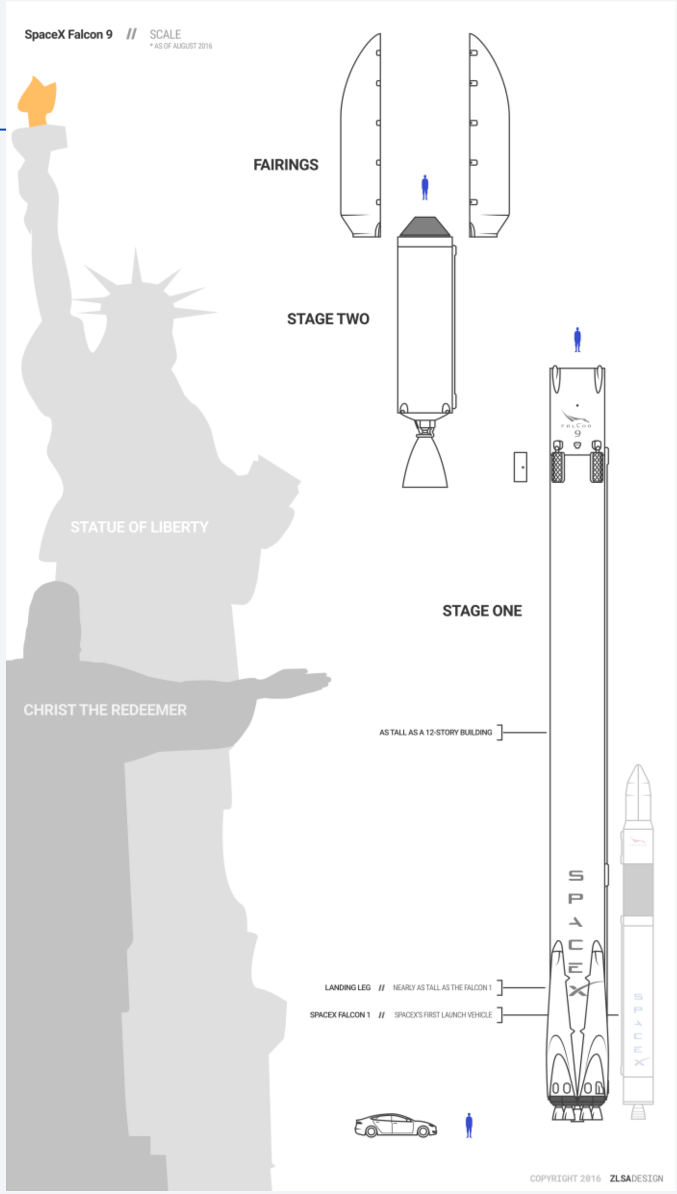
Executive Summary

- SpaceY has conducted an analysis of the launch results of the SpaceX Falcon 9 rocket and their ability to re-land the first stage.
- Re-landing the first stage of the rocket is critical for reducing costs and keeping a competitive advantage.
- Data was gathered from the official SpaceX API and Wikipedia on launches from 2020 and earlier.
- The success with which SpaceX has landed their first stage has improved dramatically over time and they now have very high success rates including for payloads up to 15,000kg. The use of landing legs and choice of very low earth orbits are important factors determining success.
- SpaceY is encouraged to choose launch sites that are proximal to the coastline and major transport infrastructure such as railways and highways when choosing their launch locations.

Introduction

- SpaceX provides a rocket launch service that is relatively cheap compared to the competition.
- A major reason for this is their ability to re-land the first stage of the rocket in many instances. The first stage of the Falcon 9 rocket is the largest stage and carries the rocket payload.
- Re-landing the first stage enables SpaceX to re-use this large and expensive part, providing significant cost savings.
- Start-up company SpaceY wants to be able to compete with SpaceX.
- Our aim is to understand what factors contribute to the likelihood of a successful re-landing of the SpaceX first stage and provide this information to SpaceY so that they can offer competitive pricing and use these learnings to enhance their own launches.

SpaceX Falcon 9 Scale



Source: ZLSA Design[1]

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was collected from the SpaceX API and scraped from Wikipedia. However, most analysis was performed on datasets downloaded from the Skills Network Labs for consistency.
- Perform data wrangling
  - Data were limited to launches from 2020 and earlier for the Falcon 9 rocket using only one core. Records with missing data were also excluded. The success of the first stage landing was inferred from the mission outcomes supplied.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Four alternative classification models were compared in Python on a subset of training data utilising a GridSearch to find the optimum parameters for each model. The accuracy of the models was compared by making predictions on a 20% test dataset and comparing with the true outcomes.

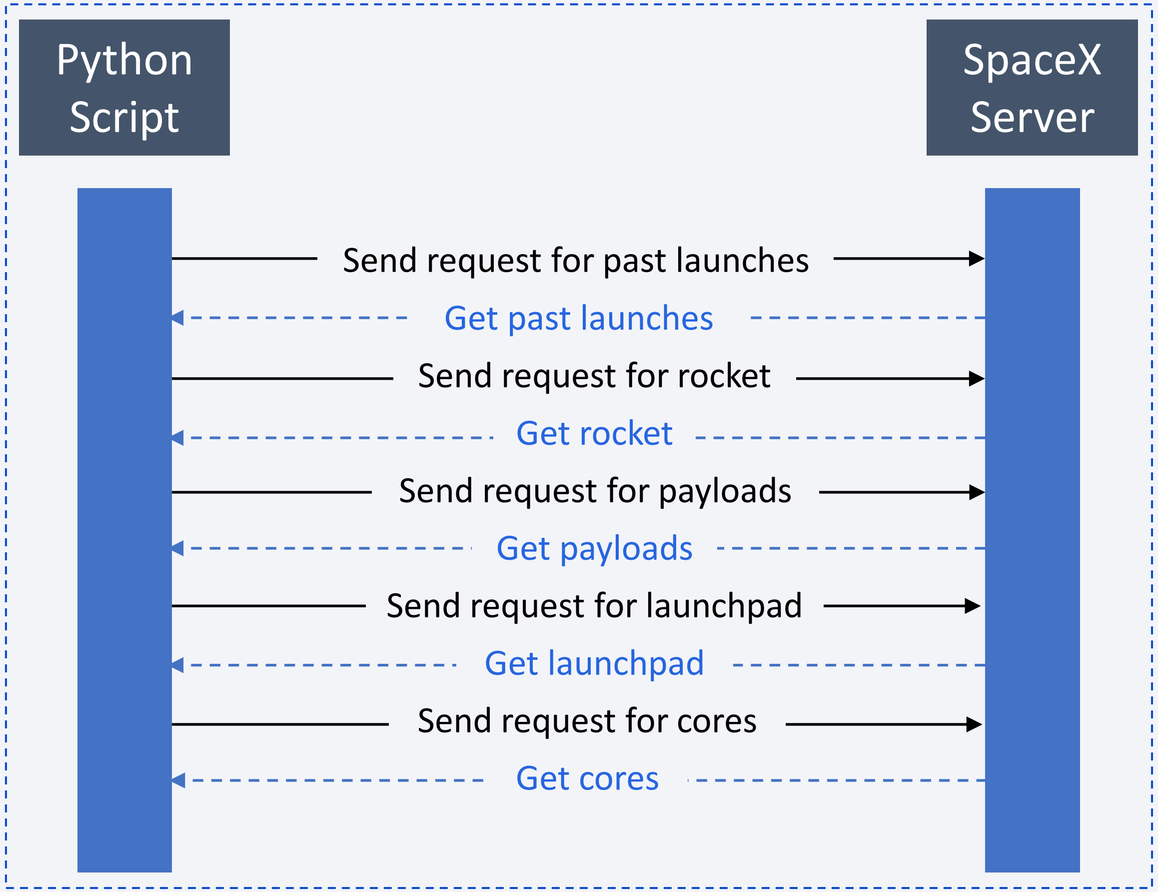
Data Collection

- Data was collected in Python 3.8 using two sources of SpaceX launch data:
  - 1. SpaceX API
    - Detailed data on all Falcon 9 launches prior to 14 November 2020 was extracted using calls to the official SpaceX APIs.
    - Data from launches that involved multiple cores, or with missing values (other than for landing pad) were excluded.
  - 2. Wikipedia
    - Data on all launches from 2020 was collected from the Wikipedia page for Falcon 9 and Falcon Heavy launches (https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches). This data contained fewer variables but includes sufficient variables for exploratory analysis of launch outcomes.
- For purposes of consistency, all actual analyses were completed using the datasets provided for this course by IBM Skills Network:

Dataset No.	URL	Tasks Used	No. Launches
1	https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/dataset_part_1.csv	Data Wrangling	90
2	https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/dataset_part_2.csv	EDA Data Visualisation, Classification	90
3	https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/dataset_part_3.csv	Classification	90
4	https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/labs/module_2/data/Spacex.csv	EDA SQL	101
5	https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/spacex_launch_geo.csv	Folium	56
6	https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/spacex_launch_dash.csv	Plotly	56

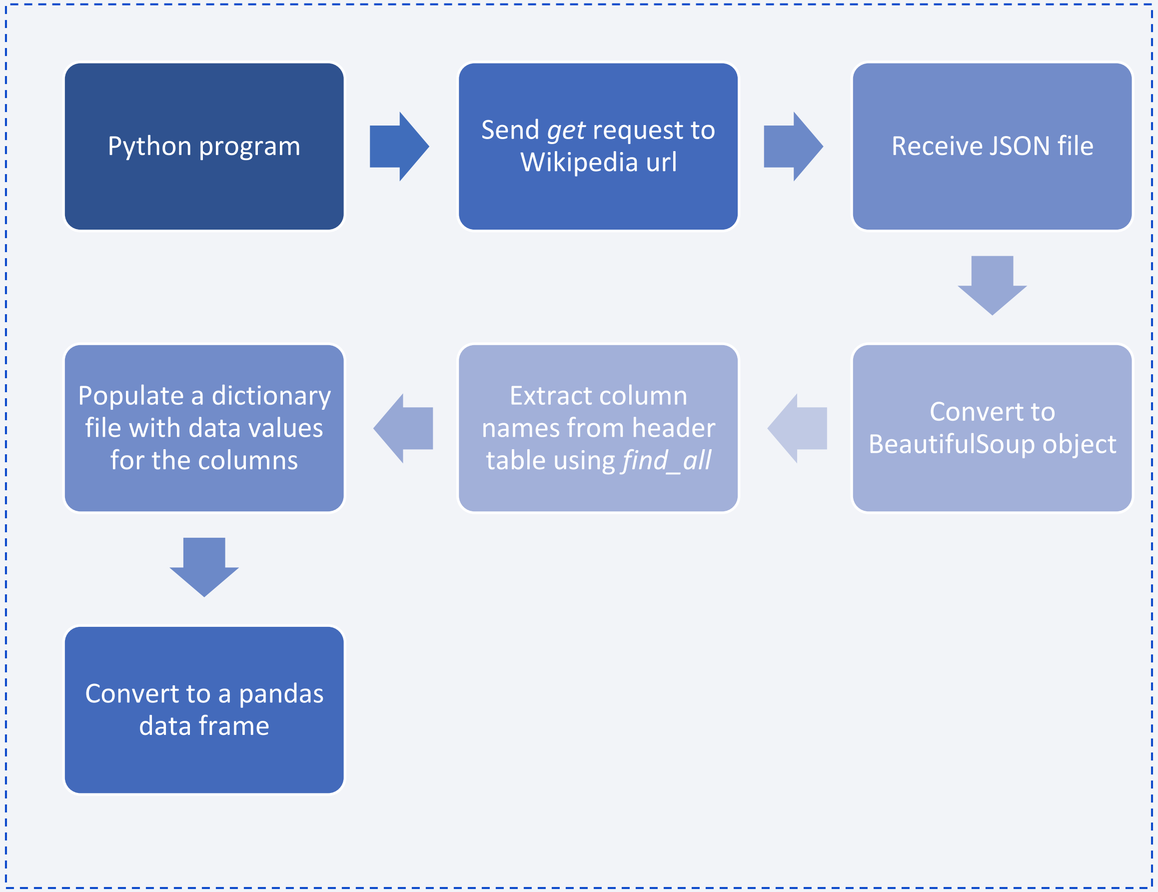
Data Collection – SpaceX API

- Data was collected from the Past Launches SpaceX API (https://api.spacexdata.com/v4/launches/past) using the *get* method in the *requests* library.
- Further API calls were made to retrieve the following attributes for IDs returned by the API.
  - `rocket`: Booster Name
  - `payloads`: Payload mass, Intended orbit
  - `launchpad`: Name of launch site, Latitude, Longitude
  - `cores`: Landing outcome and type, landing pad used, number of flights with that core, whether the core was reused (& no. times reused), core block (version number), core serial number, whether gridfins used, whether legs used
- Helper functions and API addresses for these calls are provided in Appendix 1.
- Code available here: https://github.com/razzleh/IBMDataSciencePublic/blob/main/CapstoneProject/01-spacex-data-collection-api.ipynb



Data Collection - Scraping

- Data was extracted using the *requests* and *BeautifulSoup* libraries.
- The resulting JSON file was parsed to extract column headers and data values and convert them into a pandas data frame.
- Several helper functions were used which can be found in the notebook available on GitHub.
- Code available here: https://github.com/razzleh/IBMDataSciencePublic/blob/main/CapstoneProject/02-jupyter-labs-webscraping.ipynb



Data Wrangling

- A single outcome variable ('Class') was created to classify launches as successful (1) or unsuccessful (0) depending on the value of the mission outcome.
- Launches were considered unsuccessful if they had any of the following mission outcomes:
    - False ASDS
    - False Ocean
    - False RTLS
    - None ASDS
    - None None
- All other mission outcomes were considered successful
- A description of mission outcomes is provided in Appendix 2.
- Code available here: https://github.com/razzleh/IBMDataSciencePublic/blob/main/CapstoneProject/03-labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

- The following charts were plotted and presented:
  - Flight number x launch site – Scatter
  - Flight number x Payload mass – Scatter
  - Payload mass x launch site – Scatter
  - Success Rate x orbit type – Bar
  - Flight number x orbit type - Scatter
  - Payload x orbit type – Scatter
  - Success Rate x year – Line
- Scatter plots were used to show distribution of individual flights, whilst bar and line charts were more effective for illustrating differences in calculated measures.

- Code available here: https://github.com/razzleh/IBMDataSciencePublic/blob/main/CapstoneProject/05-jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb

EDA with SQL

- A synopsis of the SQL queries performed for the EDA:
    1. Selected distinct launch sites
    2. Selected 5 records with a launch site commencing with 'CCA'
    3. Calculated the total payload mass (sum) for the 'NASA (CRS)' customer
    4. Calculated the average payload mass where the booster version was 'F9 v1.1'
    5. Selected the minimum date with a landing outcome of 'Success (ground pad)'
    6. Selected distinct booster versions with a landing outcome of 'Success (drone ship)' and a payload mass greater than 4000 and less than 6000 kg.
    7. Counted the number of records for each mission outcome
    8. Used a subquery in the where clause to selected the distinct booster versions with a payload mass equal to the maximum payload mass in the data
    9. Selected the month name, landing outcome, booster version and launch site for all launches in 2015 with a landing outcome of 'Failure (drone ship)'
    10. Selected the record count for each landing outcome between 2010-06-04 and 2017-03-20 sorted in descending order.
- Code available here: https://github.com/razzleh/IBMDataSciencePublic/blob/main/CapstoneProject/04-jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- Maps were created using the folium library in Python to display:
  - Launch sites (circles and name markers)
  - Individual launches at each site (marker cluster pins) with coloured icons indicating success/failure of first stage re-landing
  - Direction to nearest railway, coastline and highway (polyline objects)
  - Distance to nearest railway, coastline and highway (invisible icons displaying formatted distance)
- Code available here: https://github.com/razzleh/IBMDataSciencePublic/blob/main/CapstoneProject/06-lab_jupyter_launch_site_location.jupyterlite.ipynb

Build a Dashboard with Plotly Dash

- A dashboard was created using Plotly Dash with two interactive chart zones.
- Zone 1 displays a pie chart that can either display:
    - The distribution of successful flights by launch site (when all sites are selected); or
    - The breakdown of successful-unsuccessful flights for an individually selected launch site.
- Zone 2 displays a scatter plot of payload mass by outcome and is overlaid with a colour representing the booster version. This chart can be controlled by a range slider allowing the user to inspect various payload mass ranges in more detail.
- Code available here: https://github.com/razzleh/IBMDataSciencePublic/blob/main/CapstoneProject/07-spacex_dash_app.py

Predictive Analysis (Classification)

- The data was split into training (80%) and test (20%) datasets.
- Four different classification models were applied to the data: Logistic Regression, Support Vector Machines, Decision Tree, K-Nearest Neighbours.
- For each method, a grid search was performed to find the parameters that would deliver the best model for that method on the training data.
- The best parameters are then applied to the test data to calculate an accuracy score and confusion matrix for each method.
- The grid search parameters used for each method are shown in Appendix 3.
- Code available here: https://github.com/razzleh/IBMDataSciencePublic/blob/main/CapstoneProject/08-SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

Section 2

Results

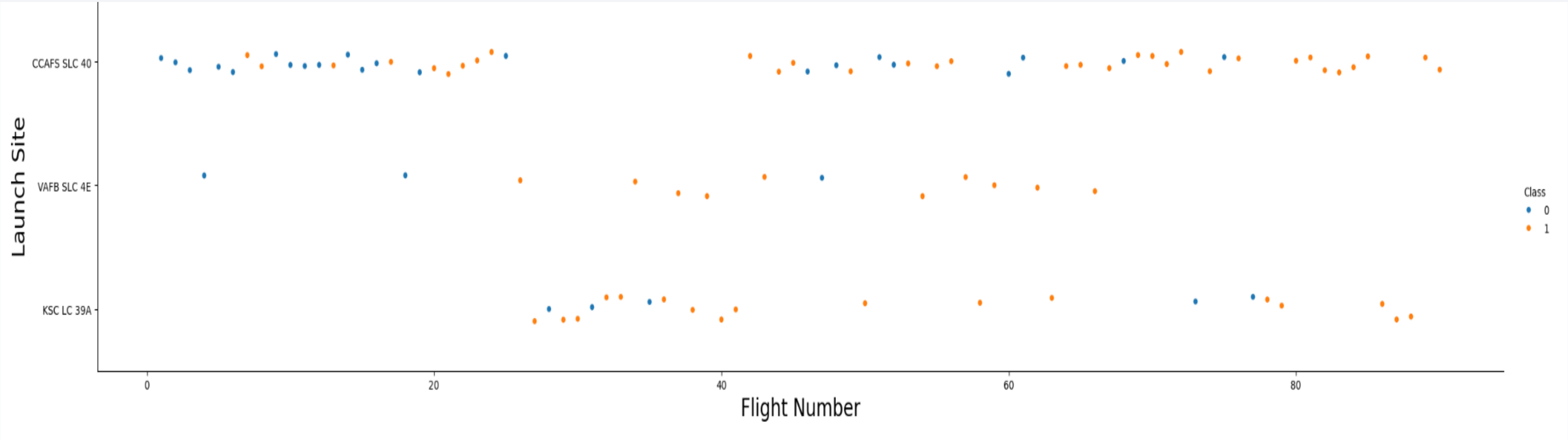
- Insights drawn from Exploratory Data Analysis
- Launch Sites Proximities Analysis
- Build a Dashboard with Plotly Dash
- Predictive Analysis (Classification)

Section 2A

Insights drawn from EDA

Flight Number vs. Launch Site

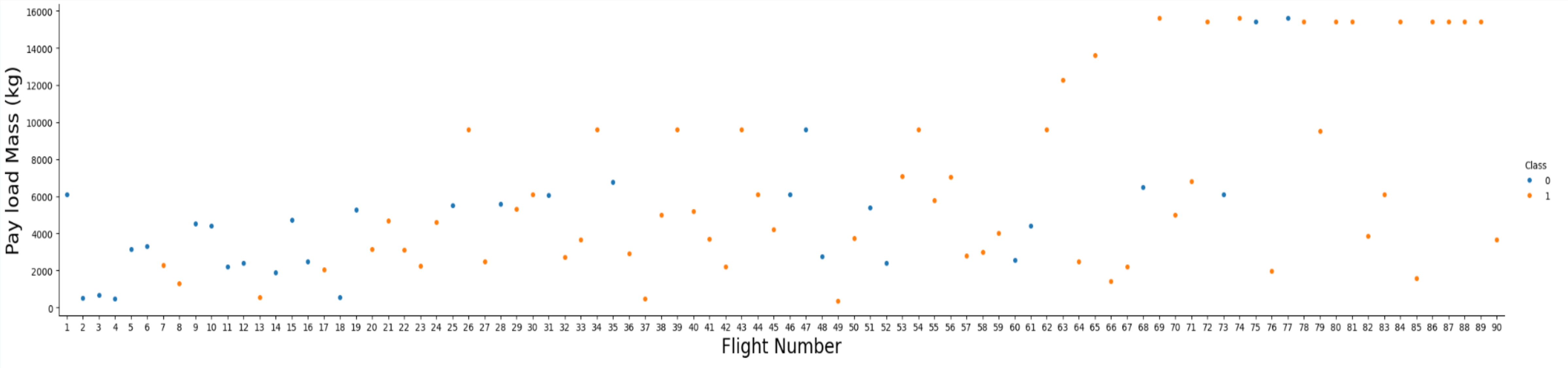
- The scatter plot below shows flight number by launch site with orange indicating that the first stage landed successfully and blue that the first stage did not successfully land.
- This chart shows that the success rate has improved over time.



Payload vs. Flight Number

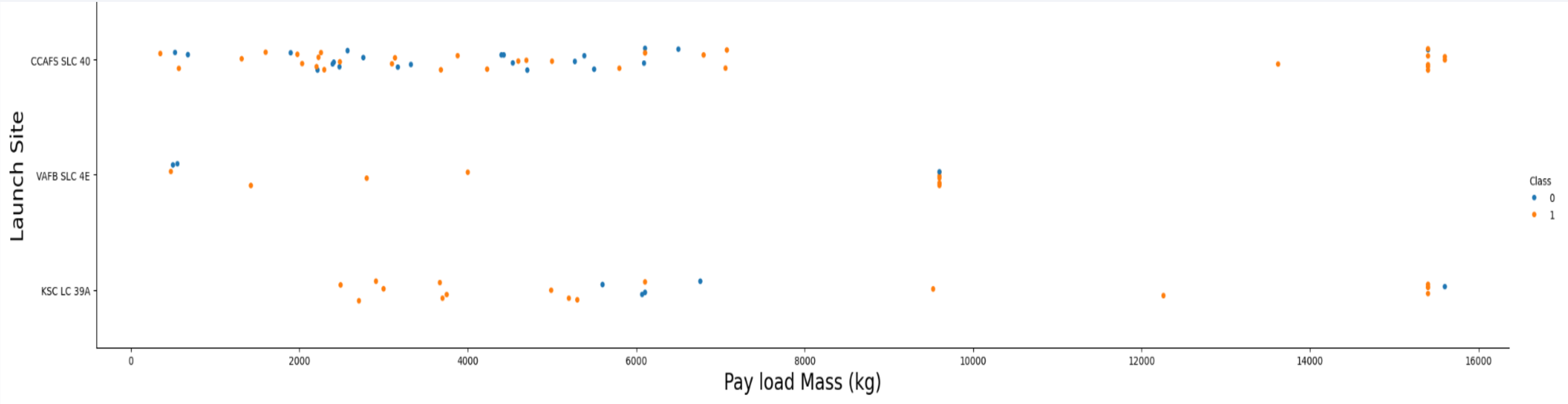
- The scatter plot below shows the payload carried by each flight with orange indicating that the first stage landed successfully and blue that the first stage did not successfully land.

- This chart shows that the payloads that were able to be successfully carried increased over the first 70 flights, but since then the maximum payload has remained roughly constant.



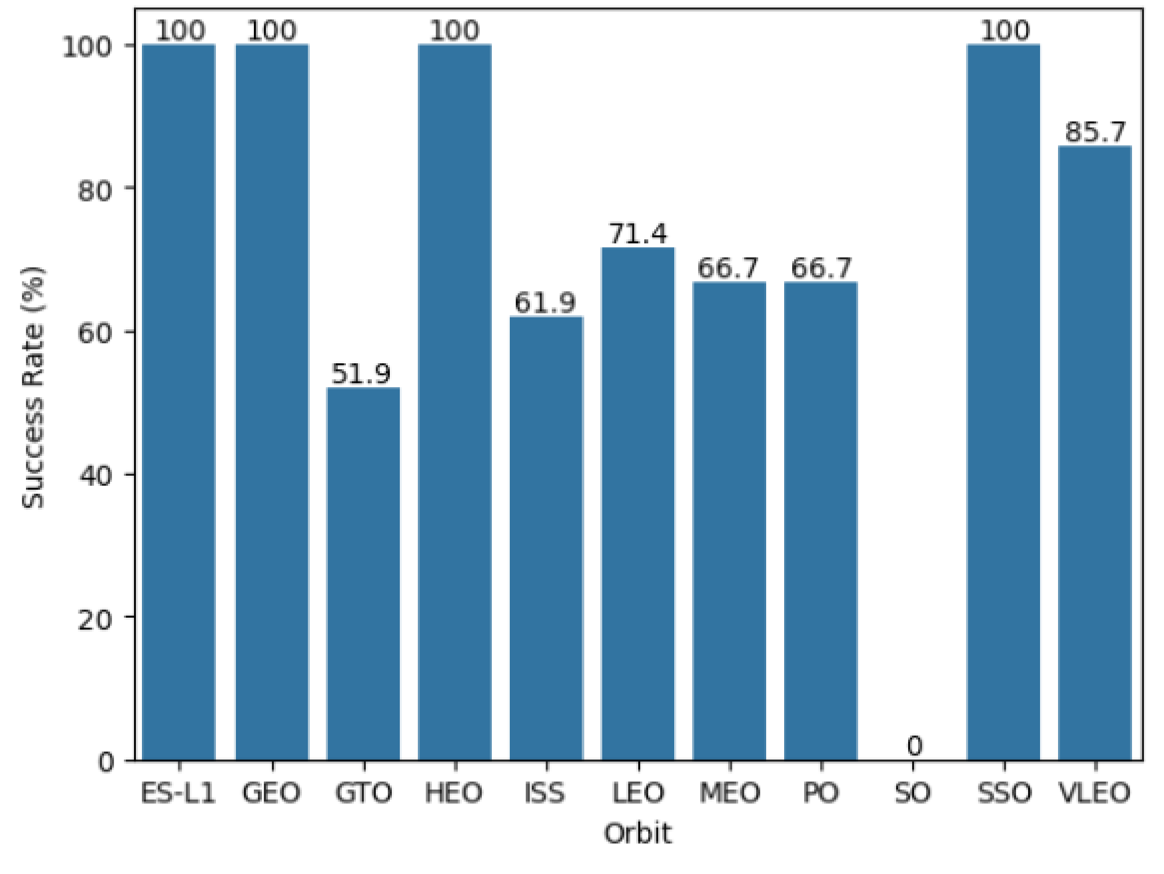
Payload vs. Launch Site

- The scatter plot below shows how the payloads and success of first stage landing have varied across the launch sites. CCAFS has been the most frequent launch site. Both it and KSC have successfully launched high payload flights and landed the first stage. VAFB has not been used on payloads above 10,000 kg.



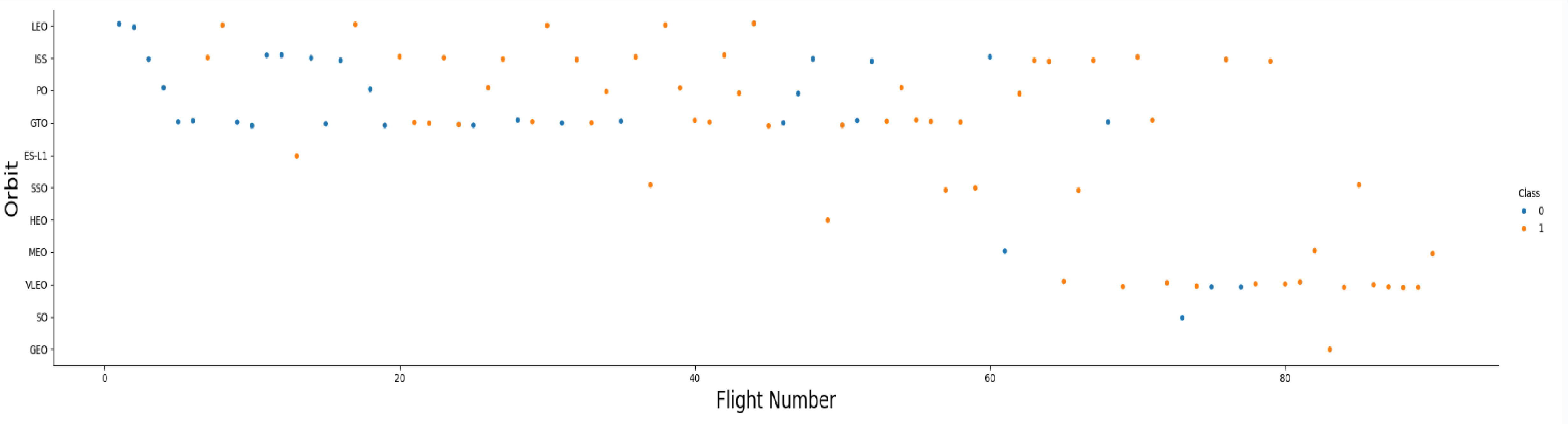
Success Rate vs. Orbit Type

- The chart at right shows the success rate of landing the first stage by intended orbit.
- Four orbit types have had success rates of 100%. Of these however, only the SSO orbit has been used more than once.
- Of the three most commonly used orbits, VLEO has the highest success rate (86%), followed by ISS (62%) and GTO (52%).



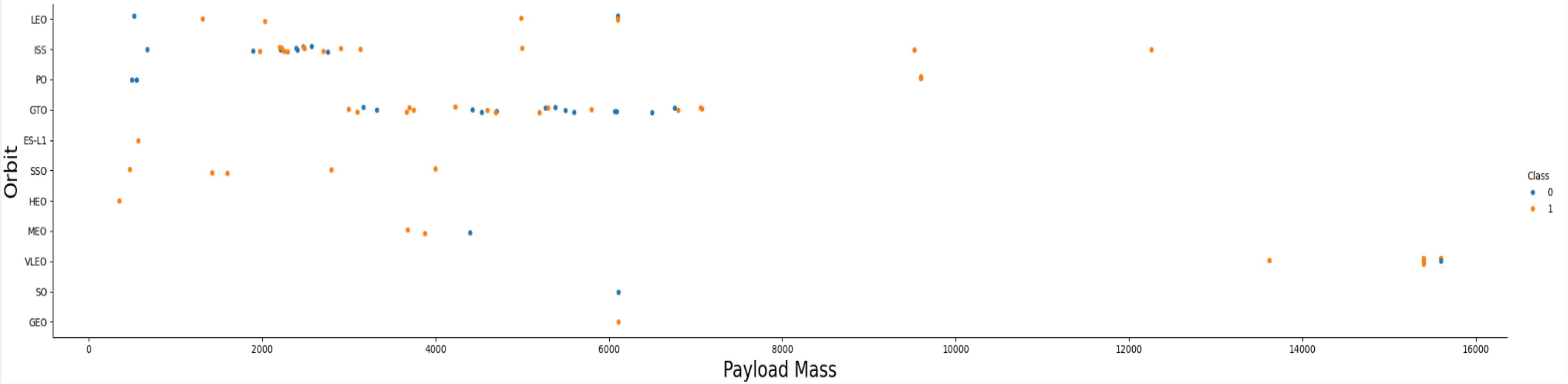
Flight Number vs. Orbit Type

- The scatter plot below shows the orbits used by flight number and success of first stage landing (success=orange, failure=blue). This shows that over time, the type of orbit used has progressed in favour of VLEO more than previously popular GTO, PO and LEO orbits. This may be as it has had a high success rate. The ISS orbit continues to be used.



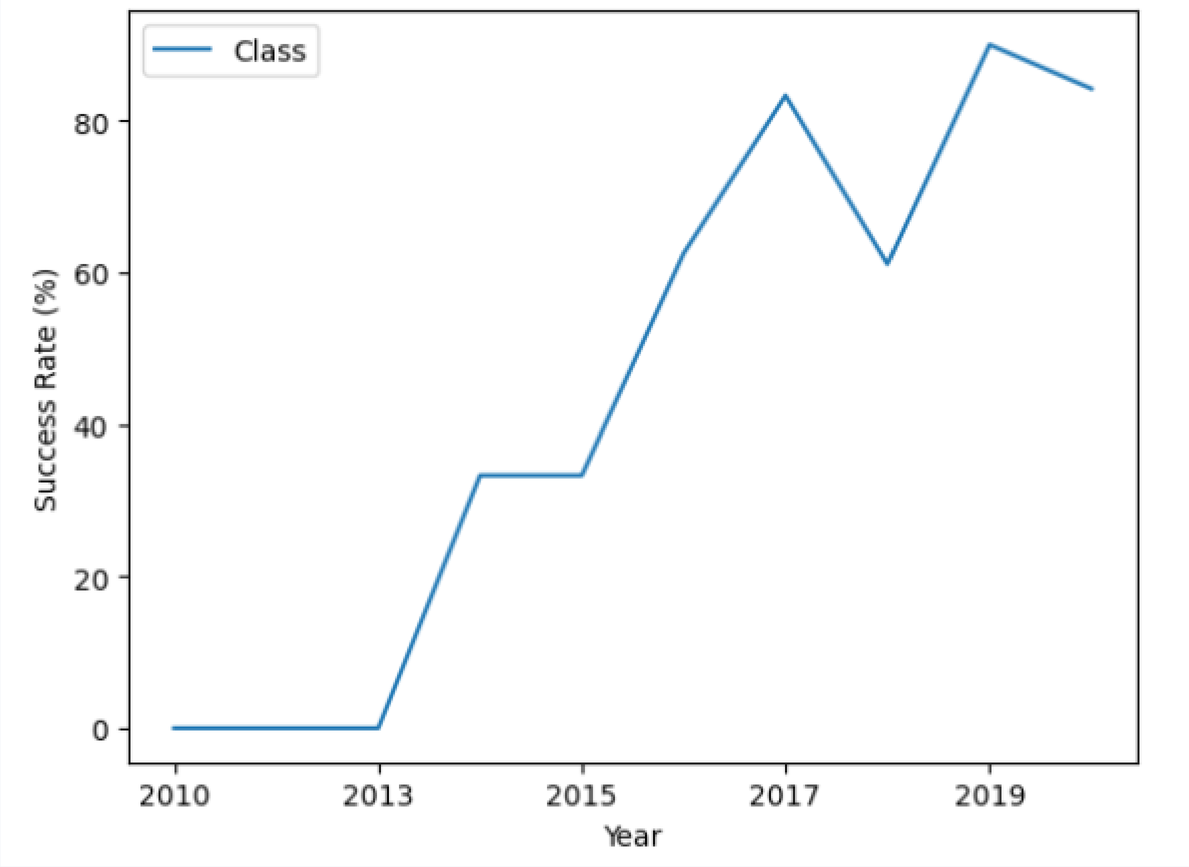
Payload vs. Orbit Type

- The scatter plot below shows the spread of payloads by orbit type and success of landing the first stage (orange=success, blue=failure).
- The highest payloads have all been launched into VLEO (very low earth orbit). The GTO orbit has largely been unsuccessful at the highest payloads at which it has been used and these remain well below 8000kg (less than half the maximum payload mass for which success has been observed across all orbits).



Launch Success Yearly Trend

- The line chart at right confirms the observations from the scatter plots that the success rate of landing the first stage has increased over time.
- Prior to 2014 there was a 0% success rate, but in 2020 the success rate was 84%



All Launch Site Names

- Four unique launch sites were found in the data. These data codes and their interpretation are shown below:

Site Code (in data)	Site Description
CCAFS LC-40	Cape Canaveral Space Force Station, Launch Complex 40
CCAFS SLC-40	Cape Canaveral Space Force Station, Space Launch Complex 40
KSC LC-39A	Kennedy Space Center, Launch Complex 39A
VAFB SLC-4E	Vandenberg Space Force Base, Space Launch Complex 4E

- CCAFS SLC-40 was previously known as CCAFS LC-40 but the two sites are treated independently in this analysis. In some datasets analysed, CCAFS LC-40 has been recoded to CCAFS SLC-40.

KSC LC-39A. Source: National Air and Space Museum [2]

Launch Site Names Begin with 'CCA'

- A selection of five records from Cape Canaveral (launch sites beginning with CCA) are shown below.
- Four of these flights were for NASA and one for SpaceX. All were considered successful missions although none of them had successful landings.

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

Total Payload Mass

- In IBM dataset 4, there was a total payload mass across all flights for NASA of 45,596 kg (45.6 tonnes).
- This is similar to the weight of a single Boeing 737-MAX aircraft[3].

Alaska Airlines Boeing 737 Max 9. Source: Wikipedia[4]

Average Payload Mass by F9 v1.1

- In IBM dataset 4, the average payload mass carried by Falcon 9 flights using booster version 1.1 was 2,928kg.
- This is similar to the weight of a RAM 1500[5,6].

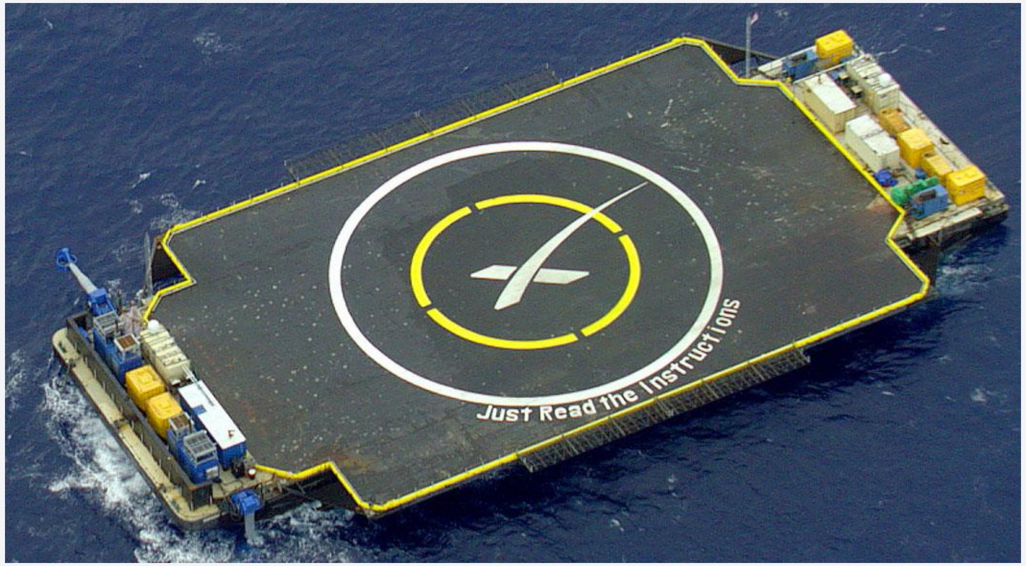
2023 RAM 1500 Laramie Sport. Source: Carsales.com.au[7]

First Successful Ground Landing Date

- The first successful landing on a ground pad was achieved on 22 December 2015.

Successful Drone Ship Landing with Payload between 4000 and 6000

- The following boosters have successfully landed on a drone ship having a payload mass greater than 4000 but less than 6000kg:
  - F9 FT B1022
  - F9 FT B1026
  - F9 FT B1021.2
  - F9 FT B1031.2



SpaceX drone ship. Source: SpaceNews.com[8]

Total Number of Successful and Failure Mission Outcomes

- Based on IBM dataset 4, the Falcon 9 has had 100 successful missions and one unsuccessful mission that failed in flight.
- The unsuccessful mission, SpaceX CRS-7, was launched on 28 June 2015. It was a private resupply mission to the International Space Station contracted by NASA which failed less than three minutes into the flight.[9]

Boosters Carried Maximum Payload

- The following boosters have all carried the maximum payload mass of 15,600 kg.

Booster Version	Booster Version
F9 B5 B1048.4	F9 B5 B1051.4
F9 B5 B1048.5	F9 B5 B1051.6
F9 B5 B1049.4	F9 B5 B1056.4
F9 B5 B1049.5	F9 B5 B1058.3
F9 B5 B1049.7	F9 B5 B1060.2
F9 B5 B1051.3	F9 B5 B1060.3

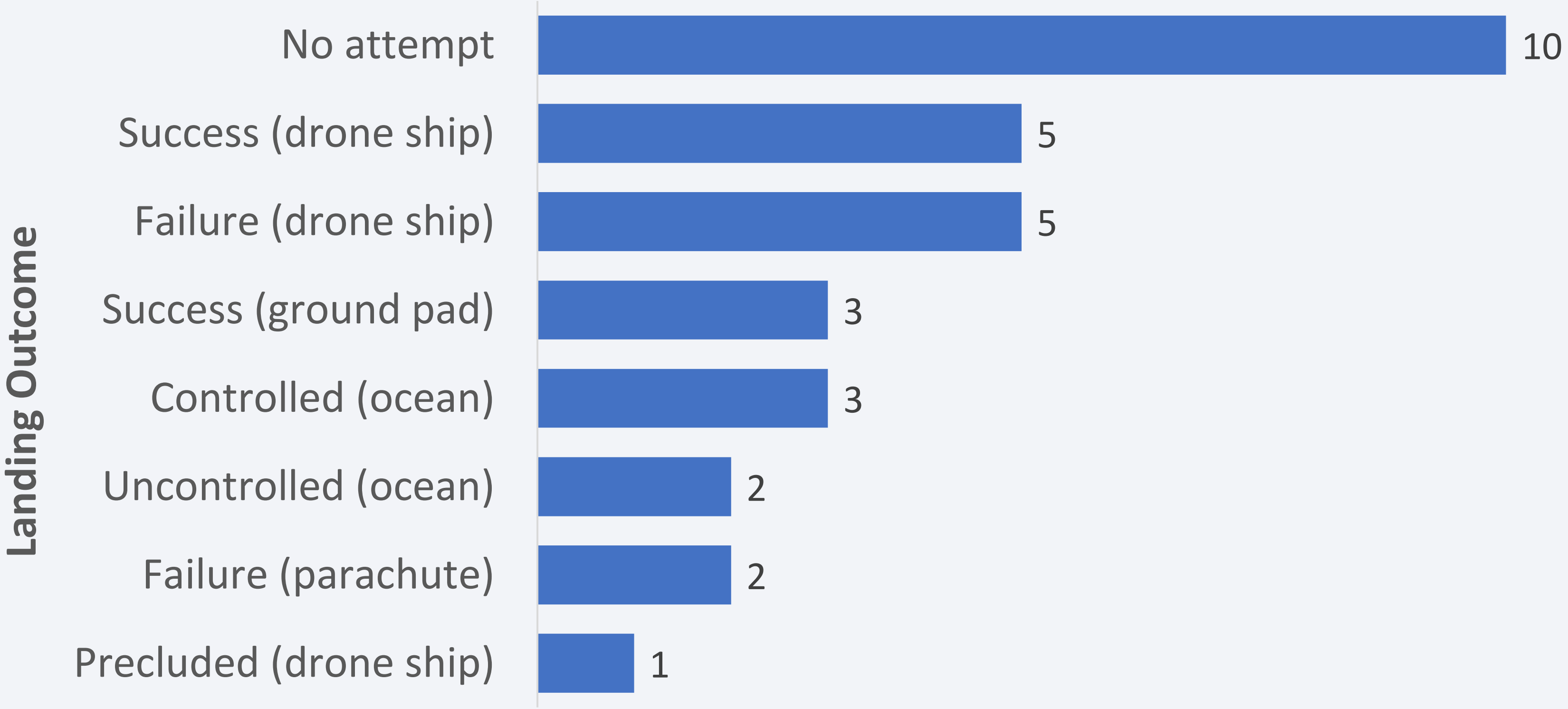
2015 Launch Records

- There were two missions in 2015 that had failed drone ship landings. The booster versions and launch sites for these were as shown below:

Month	Landing Outcome	Booster Version	Launch Site
January	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
April	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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

- The chart below shows the distribution of landing outcomes between the dates 2010-06-04 and 2017-03-20 in descending order.

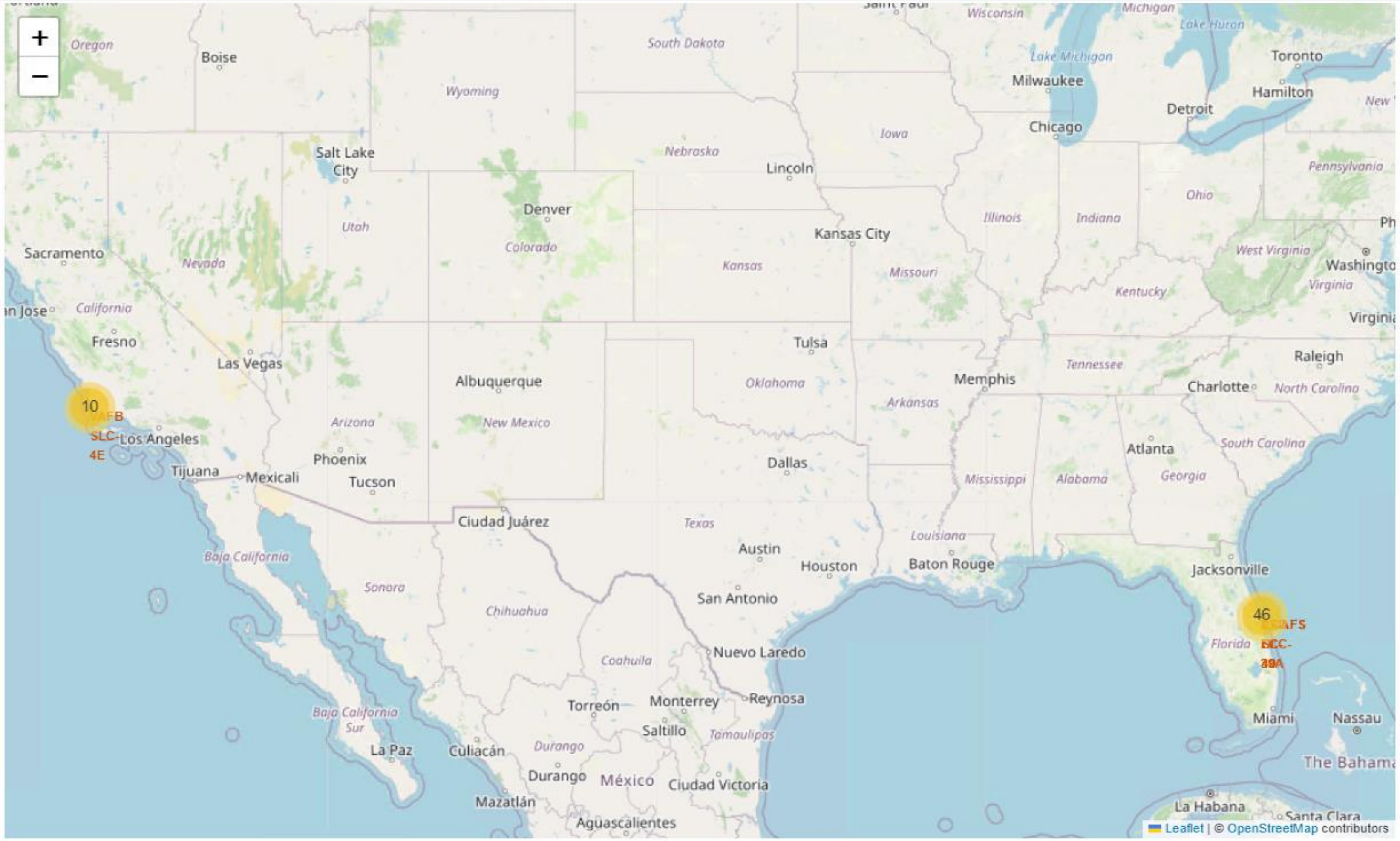


Section 2B

Launch Sites Proximities Analysis

Location of SpaceX Launches

- This map shows the location of each launch recorded in the data. It shows that the majority of launches have occurred in Florida and a smaller number on the West Coast of the USA.

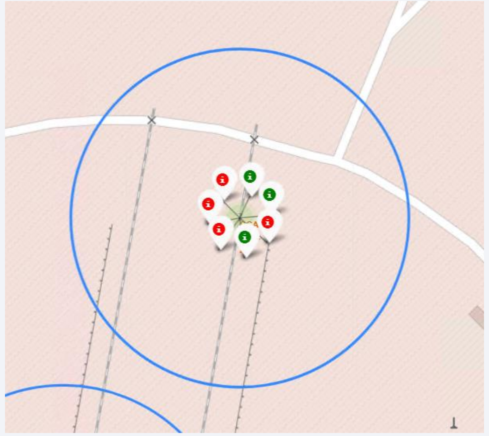


First stage re-landing success by Launch Site

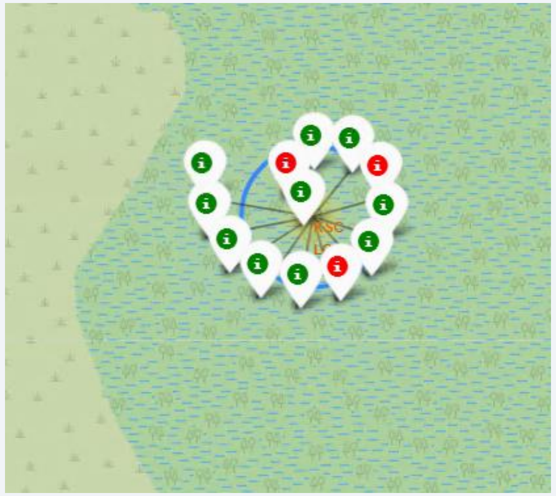
- The images at right show colour coded pins for each launch at the four launch sites, with green pins indicating successful re-landing of the first stage and red pins indicating unsuccessful landing of the first stage.

CCAFS LC-40

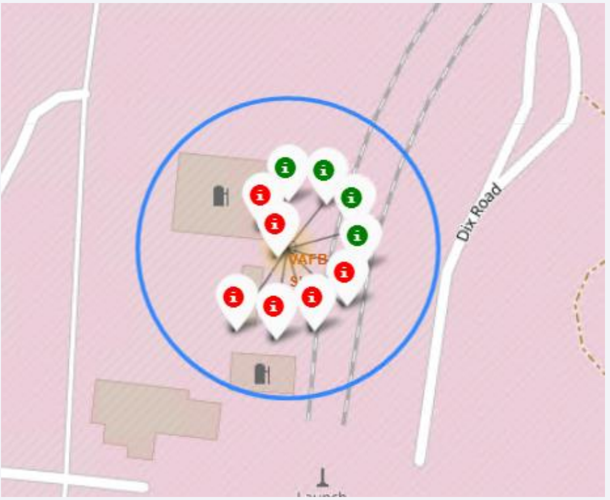
CCAFS SLC-40



KSC LC-39A

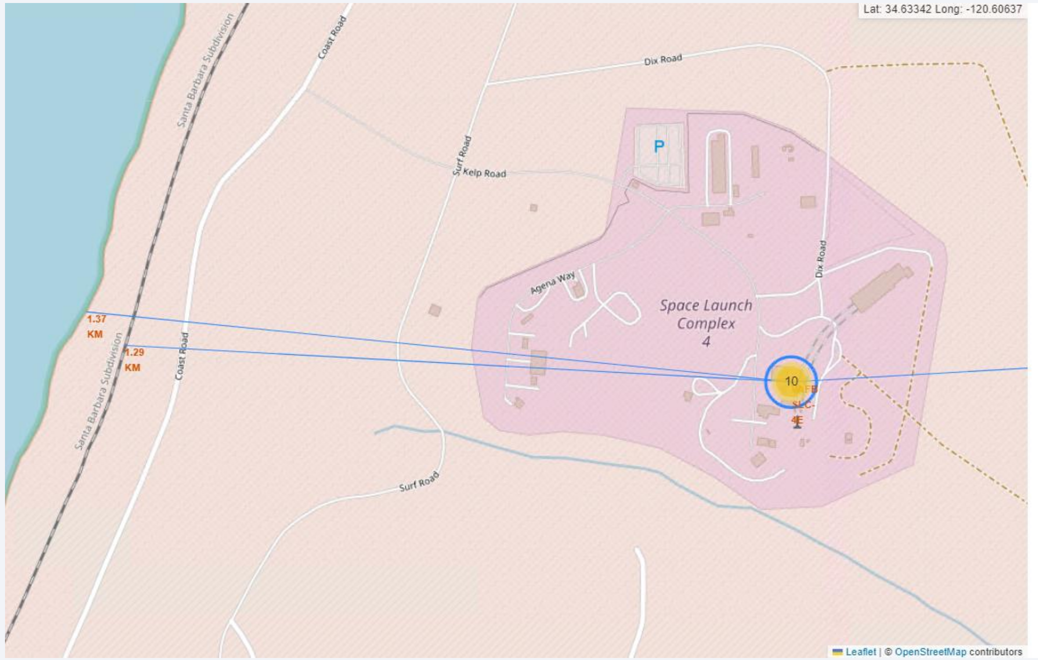


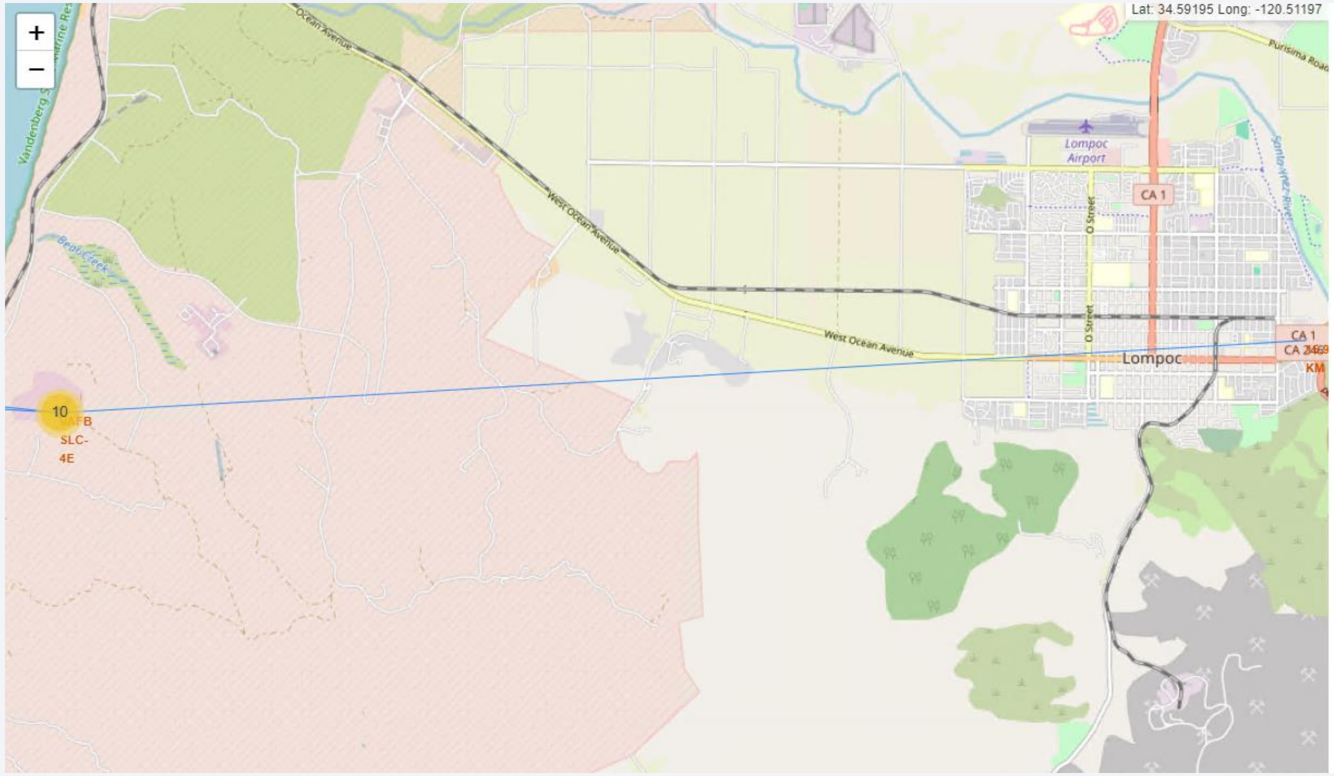
VAFB SLC-4E



Proximity of VAFB to coast and transport

- The Vandenberg Space Force Base is very close to rail (1.29km) and the coast (1.37km) but is somewhat farther from the highway (marked as where East Ocean Avenue becomes Hwy 246; 15.99km)



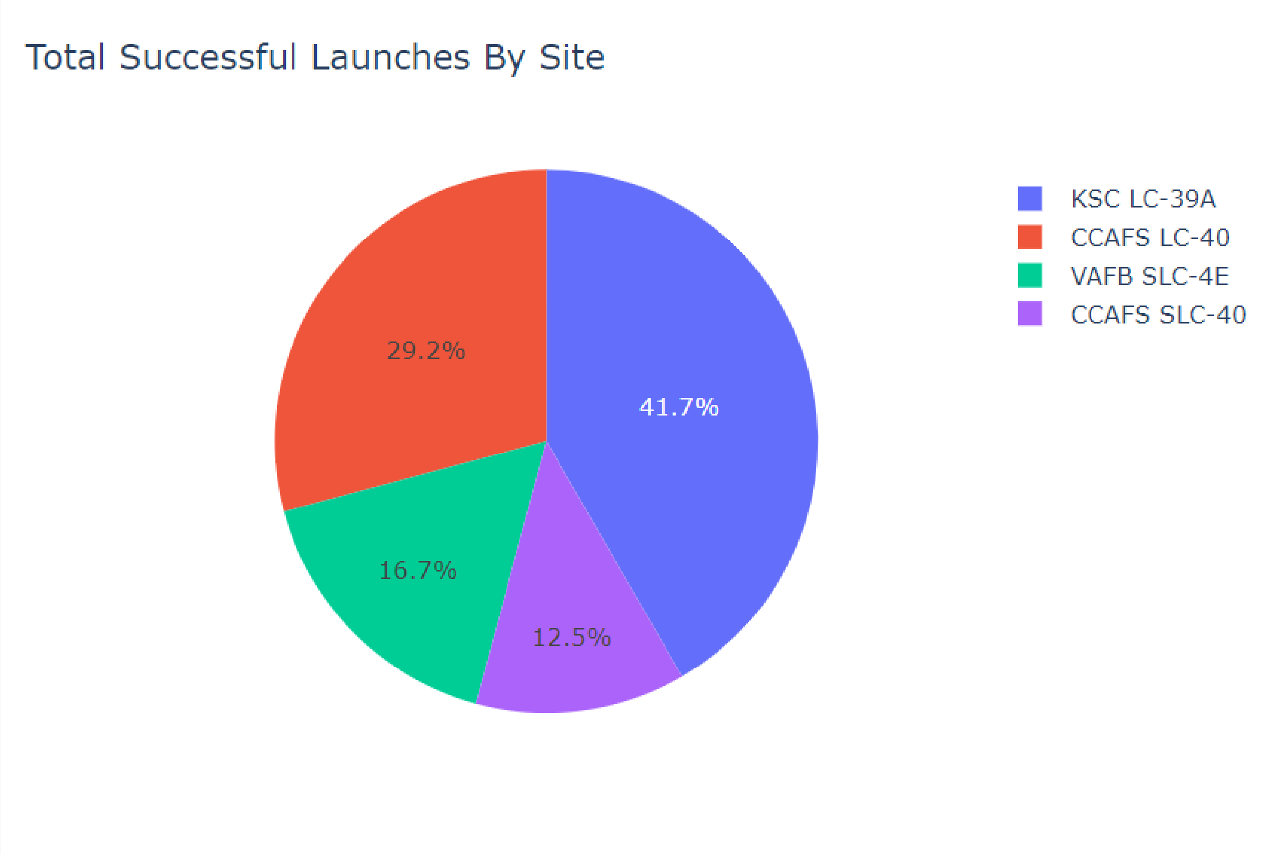


Section 2C

Build a Dashboard with Plotly Dash

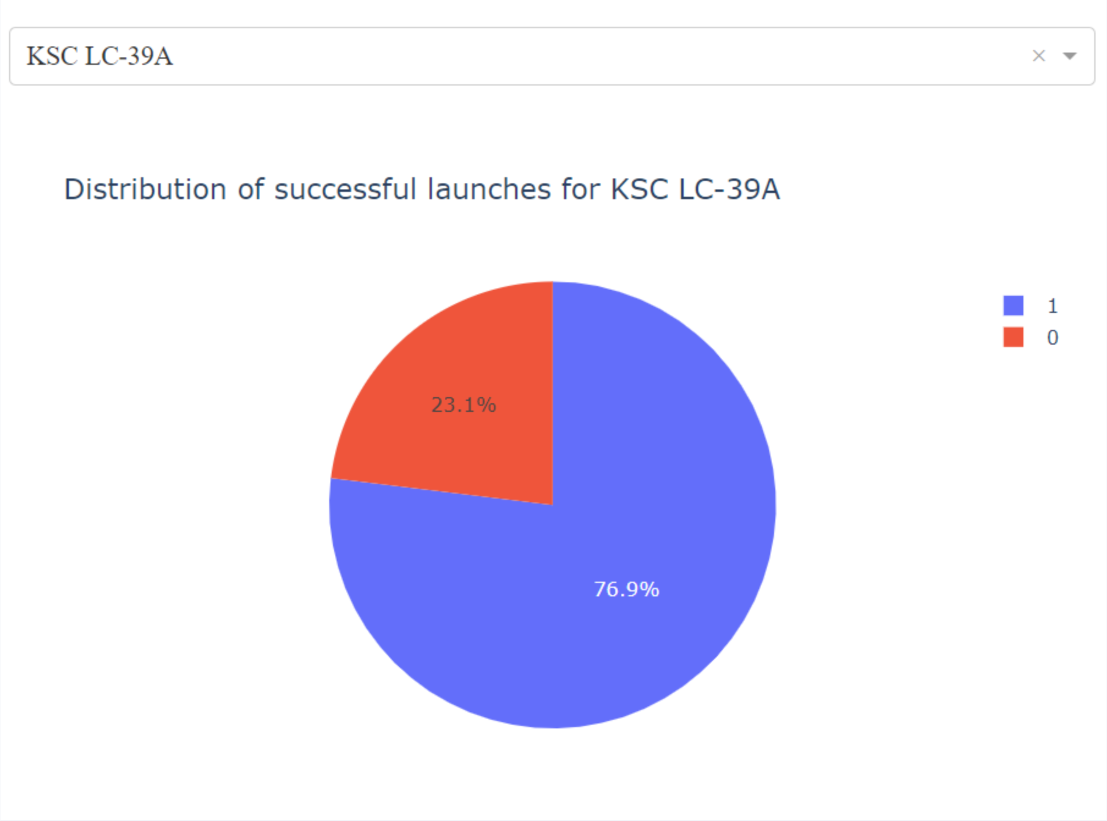
Successful Launch Sites

- The first chart on the dashboard shows the distribution of “successful launches” by Launch site.
- Although Kennedy Space Centre had the most successful launches, it is in fact equal to the Cape Canaveral Space Force Station when the two CCAFS sites are combined.



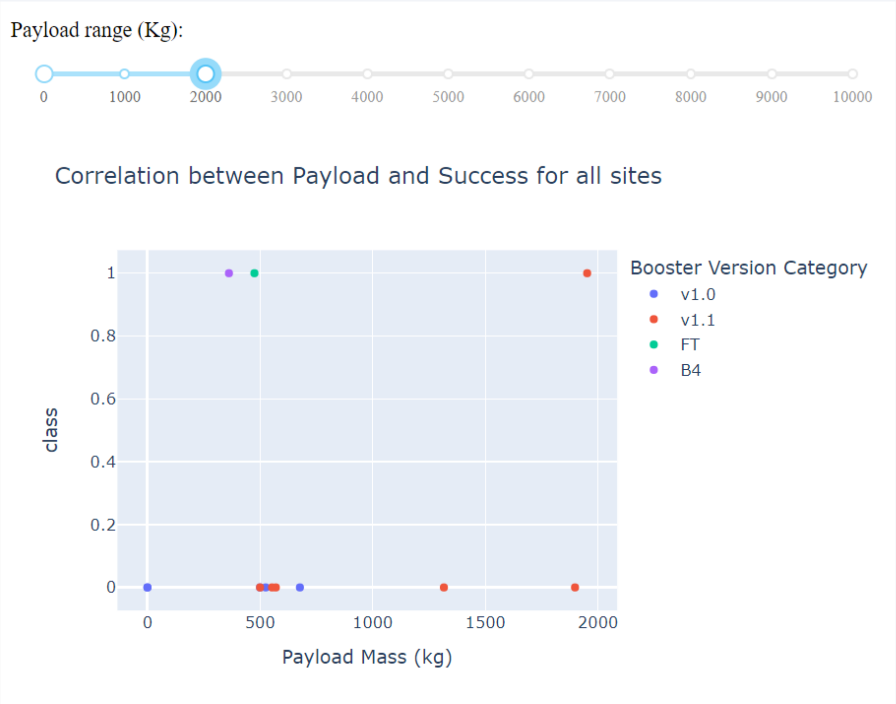
Launch Outcomes at KSC

- The dashboard can be filtered to show the individual launch outcomes at any site.
- This chart shows the outcomes at Kennedy Space Centre, which was successful for almost 77% of launches.



Success at Various Payloads

- Finally, the dashboard has an interactive visualisation of success by payload mass and booster version. The range slider allows the chart to be filtered by payload mass.
- These two images show that at very low payload mass, most boosters had infrequent success, but at a higher range, the FT booster had more successful launches. V1.1 booster had no success in either of these ranges.

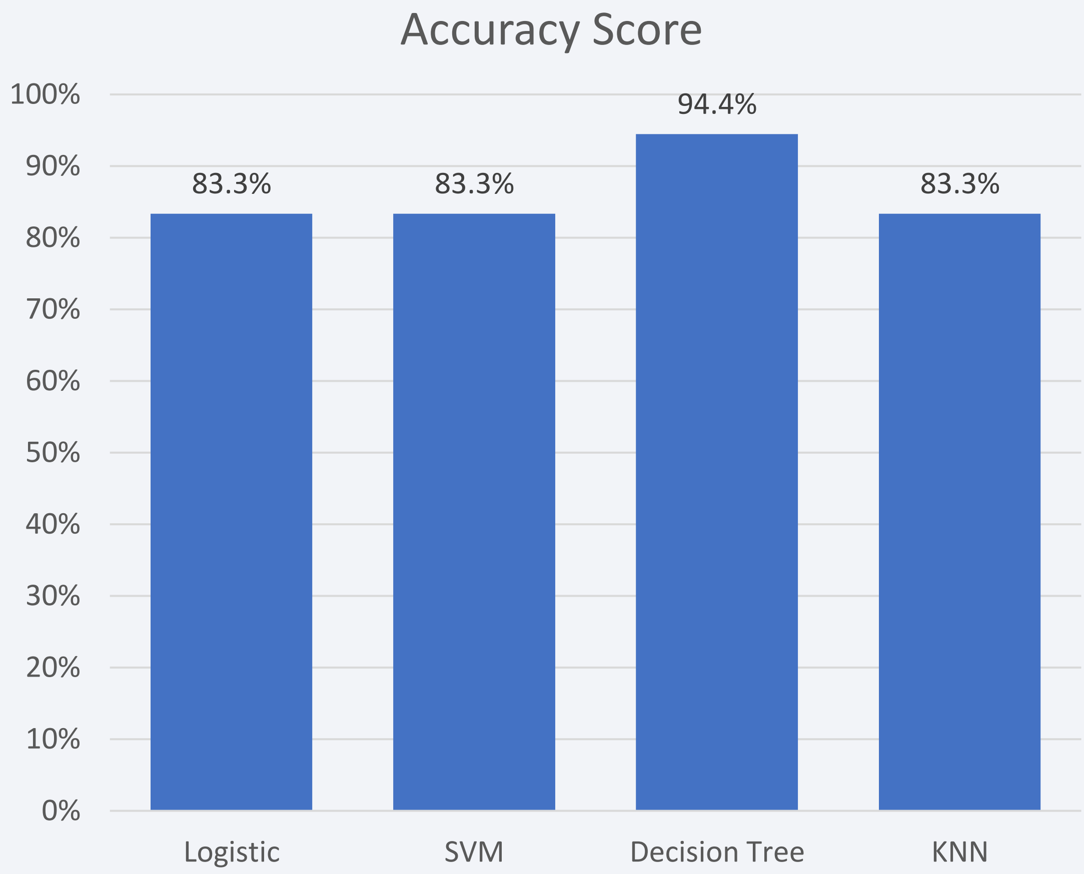


Section 2D

Predictive Analysis (Classification)

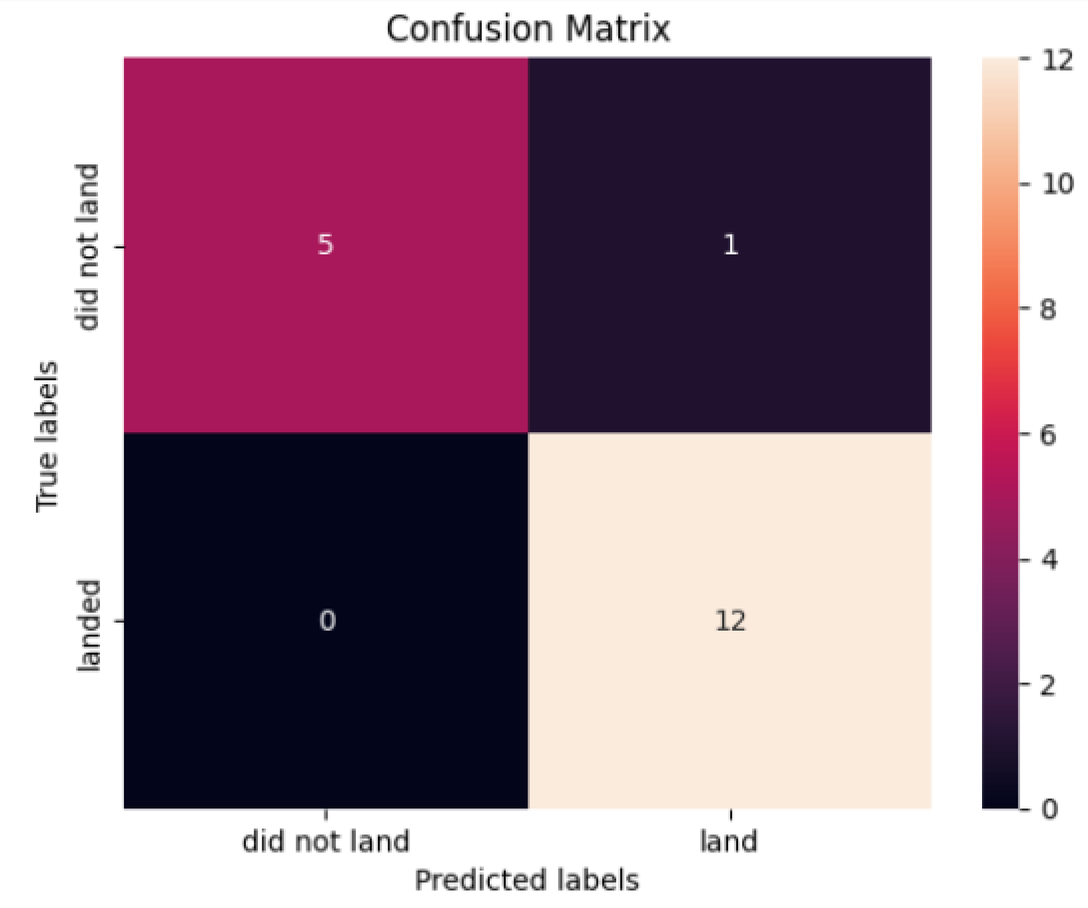
Classification Accuracy

- Four classification models were applied to the data with their accuracy scores shown at right.
- The decision tree had the highest accuracy (94%).
- All other models had the same level of accuracy (83%).



Confusion Matrix

- The confusion matrix for the decision tree model (shown at right) predicted all but one record correctly in the test data.
- The incorrect prediction was a false positive. The model predicted that the first stage would land successfully when in fact it did not.
- This gave a false positive rate of 17%.
- There were no false negatives in this test sample.



Decision Tree Summary

- The top insights from the decision tree model were as follows:
  - The most important feature was the use of landing legs
    - Flights without landing legs almost exclusively failed
  - The next most important feature was very low earth orbit (VLEO)
    - VLEO orbits are associated with much higher rates of landing success
  - Other important features were: Block, No. Flights and the orbits LEO and GTO

SpaceX landing legs. Source: StackExchange[10]

Section 3

Conclusions

Conclusions

- The success rate of SpaceX landing the first stage of the Falcon 9 has dramatically increased over the 10 years from 2010 to 2020.
- Success rates have been high for a few years now.
- Over time, SpaceX have been able to dramatically increase the payload on the first stage and successfully land it, with a number of successful payloads around 15,000 kg. These payloads have had particular success when launched into very low earth orbits.
- The use of landing legs was found to be the overall most important predictor of success in re-landing the first stage. However, as SpaceX sometimes intentionally sacrifices the first stage, more work is required to unpick the interplay between use of landing legs and intention to land the mission.
- Analysis of SpaceX launch site locations has found that they are very close to the coastline, railways, highways and major cities and SpaceY should also factor these features into their choice of launch pad locations.

Section 4

References

References

[1] ZLSA Design 2019, *SpaceX Falcon 9 Scale,* accessed 26 March 2024, <https://zlsadesign.com/infographic/vehicle/spacex-falcon9-scale/>.

[2] Neufeld M 2020, *Launch Complex 39: From Saturn to Shuttle to SpaceX and SLS,* National Air and Space Museum – Smithsonian, accessed 25 March 2024, < https://airandspace.si.edu/stories/editorial/launch-complex-39-saturn-shuttle-spacex-and-sls>.

[3] Wikipedia 2024, Table: Boeing 737 characteristics, *Boeing 737*, accessed 25 March 2024, <https://en.wikipedia.org/wiki/Boeing_737>.

[4] KirkXWB 2023, "Alaska Airlines Boeing 737 MAX 9, registered N960AK, over Seattle-Tacoma Int'l Airport", digital photograph, Wikipedia, accessed 25 March 2024, < https://en.wikipedia.org/wiki/Boeing_737_MAX#/media/File:Alaska_737_Max_9.jpg>

[5] TacomaDodge 2023, *RAM 1500 Weight: GVWR, Curb Weight & More*, accessed 25 March 2024 <https://www.tacomadodge.com/manufacturer-information/ram-1500-weight/>.

[6] Google 2024, *Convert pounds to kg*, accessed 25 March 2024 <https://www.google.com.au/search?q=convert+pounds+to+kg>.

[7] Carsales 2023, "RAM 1500 Laramie Sport 01", digital photograph, Carsales.com.au, accessed 25 March 2024, < https://editorial.pxcrush.net/carsales/general/editorial/ram-1500-laramie-sport-01.jpg?width=1024&height=682>.

[8] Foust J 2016, *SpaceX to Land at Sea after Launching Jason-3*, SpaceNews.com, accessed 25 March 2024, <https://spacenews.com/spacex-to-land-at-sea-after-launching-jason-3/>.

[9] Wikipedia 2024, *SpaceX CRS-7,* accessed 25 March 2024 <https://en.wikipedia.org/wiki/SpaceX_CRS-7>.

[10] Machavity 2019, *Why doesn't the Falcon-9 first stage use three legs to land?,* StackExchange, accessed 25 March 2024 < https://space.stackexchange.com/questions/38224/why-doesnt-the-falcon-9-first-stage-use-three-legs-to-land>.

Section 5

Appendix

Appendix 1 – API Helper functions (1/2)

From the `rocket` column we would like to learn the booster name.

```
# Takes the dataset and uses the rocket column to call the API and append the data to the list
def getBoosterVersion(data):
    for x in data['rocket']:
       if x:
        response = requests.get("https://api.spacexdata.com/v4/rockets/"+str(x)).json()
        BoosterVersion.append(response['name'])
```

From the `launchpad` we would like to know the name of the launch site being used, the logitude, and the latitude.

```
# Takes the dataset and uses the launchpad column to call the API and append the data to the list
def getLaunchSite(data):
    for x in data['launchpad']:
       if x:
         response = requests.get("https://api.spacexdata.com/v4/launchpads/"+str(x)).json()
         Longitude.append(response['longitude'])
         Latitude.append(response['latitude'])
         LaunchSite.append(response['name'])
```

From the `payload` we would like to learn the mass of the payload and the orbit that it is going to.

```
# Takes the dataset and uses the payloads column to call the API and append the data to the lists
def getPayloadData(data):
    for load in data['payloads']:
       if load:
        response = requests.get("https://api.spacexdata.com/v4/payloads/"+load).json()
        PayloadMass.append(response['mass_kg'])
        Orbit.append(response['orbit'])
```

Appendix 1 – API Helper functions (2/2)

From `cores` we would like to learn the outcome of the landing, the type of the landing, number of flights with that core, whether gridfins were used, wheter the core is reused, wheter legs were used, the landing pad used, the block of the core which is a number used to seperate version of cores, the number of times this specific core has been reused, and the serial of the core.

```
# Takes the dataset and uses the cores column to call the API and append the data to the lists
def getCoreData(data):
    for core in data['cores']:
            if core['core'] != None:
                response = requests.get("https://api.spacexdata.com/v4/cores/"+core['core']).json()
                Block.append(response['block'])
                ReusedCount.append(response['reuse_count'])
                Serial.append(response['serial'])
            else:
                Block.append(None)
                ReusedCount.append(None)
                Serial.append(None)
            Outcome.append(str(core['landing_success'])+' '+str(core['landing_type']))
            Flights.append(core['flight'])
            GridFins.append(core['gridfins'])
            Reused.append(core['reused'])
            Legs.append(core['legs'])
            LandingPad.append(core['landpad'])
```

Appendix 2 – Mission Outcomes

The table below describes the mission outcomes used to classify missions as successfully or unsuccessfully landed.

Code	Description	Successful Landing?
False ASDS	Unsuccessfully landed to drone ship	No
False Ocean	Unsuccessfully landed to specific region of ocean	No
False RTLS	Unsuccessfully landed to ground pad	No
None ASDS	Failure to land	No
None None	Failure to Land	No
True ASDS	Successfully landed to drone ship	Yes
True Ocean	Successfully landed to specific region of ocean	Yes
True RTLS	Successfully landed to ground pad	Yes

Appendix 3: Grid Search Parameters for Predictive Analysis

- **Logistic Regression:**
  - C: {0.1, 0.1, 1}
  - penalty='l2'
  - solver='lbfgs'
  - cv=10

- **Support Vector Machines:**
  - C: np.logspace(-3,3,5)
  - gamma: np.logspace(-3,3,5)
  - kernel: linear, rbf, poly, sigmoid
  - cv=10

- **Decision Tree**
  - criterion: gini, entropy
  - splitter: best, random
  - max_depth: 2*n for n in range(1,10)
  - max_features: auto, sqrt
  - min_samples_leaf: 1,2,4
  - min_samples_split: 2,5,10
  - cv=10

- **K-Nearest Neighbours**
  - n_neighbours: 1,11,1
  - algorithm: auto, ball_tree, kd_tree, brute
  - p: 1,2
  - cv=10

Thank you!