Winning Space Race with Data Science

Barnaby Donohew

12th November 2021

Outline

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

- GitHub repo here: https://github.com/delphinusuk/ibm-ds-pro-capstone

Executive Summary

- The aim was to investigate the factors that affected the success of SpaceX missions, such as the rocket design, payload, launch site and landing method.
- SpaceX mission data was obtained from the SpaceX public API and from SpaceX launch data on Wikipedia. The desired data was extracted and transformed for Exploratory Data Analysis (EDA) and visualization (including mapping and interactive dashboards).
- Various machine learning classification models (logistic regression, support vector machines, decision trees, and k-nearest-neighbors) were trained and tested (and the hyperparameters tuned) to find the most accurate predictor of launch success.
- The decision tree model was the most accurate predictor of launch success.
- https://github.com/delphinusuk/ibm-ds-pro-capstone

Introduction

- In this capstone, we will predict if the Falcon 9 first stage will land successfully. SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage.
- Therefore if we can determine if the first stage will land, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch.

Section 1

Methodology

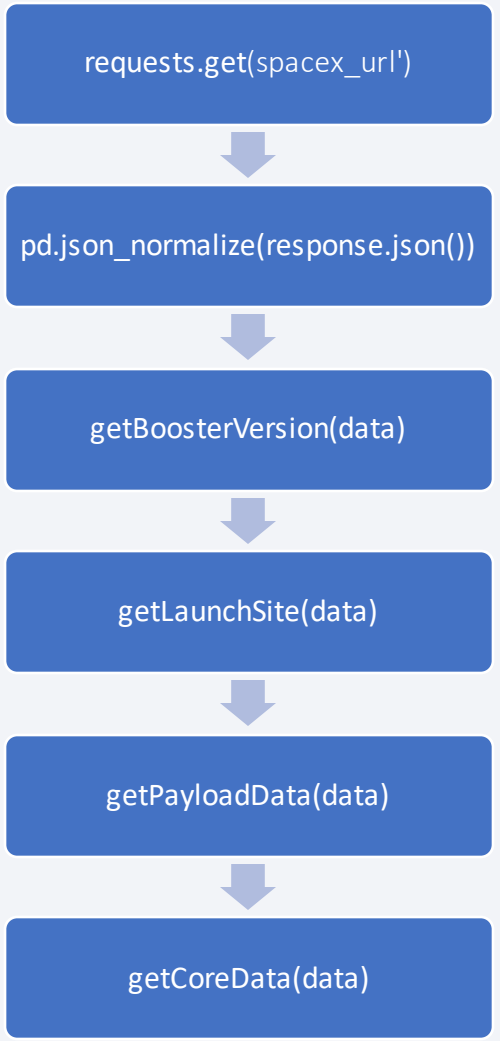
Methodology

Executive Summary

- Data collection methodology:
  - SpaceX API and web scraping Wikipedia
- Perform data wrangling
  - Dealt with missing values and created landing category variable
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Grid search to train, test and tune classification models

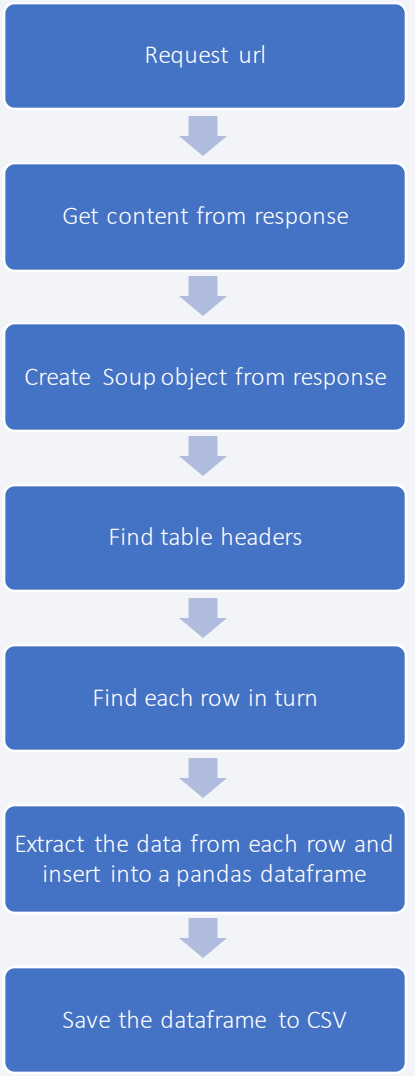
Data Collection – SpaceX API

- SpaceX REST API requests are as follows:
  - Past launch data is requested from:
    - https://api.spacexdata.com/v4/launches/past
  - Booster data is gained from a rocket type request:
    - https://api.spacexdata.com/v4/rockets/
  - Launchpad name, longitude and latitude comes from a launch site request:
    - https://api.spacexdata.com/v4/launchpads/
  - Payload data comes from a payloads request:
    - https://api.spacexdata.com/v4/payloads/
  - Specific rocket core information comes from a core request:
    - https://api.spacexdata.com/v4/cores/
- Falcon 9 data was added to a pandas dataframe, then, after missing values dealt with, it was exported to a CSV file
- https://github.com/delphinusuk/ibm-ds-pro-capstone/blob/9499761c563116123f02c582cf491713aa542fab/Data%20Collection%20API%20Lab.ipynb



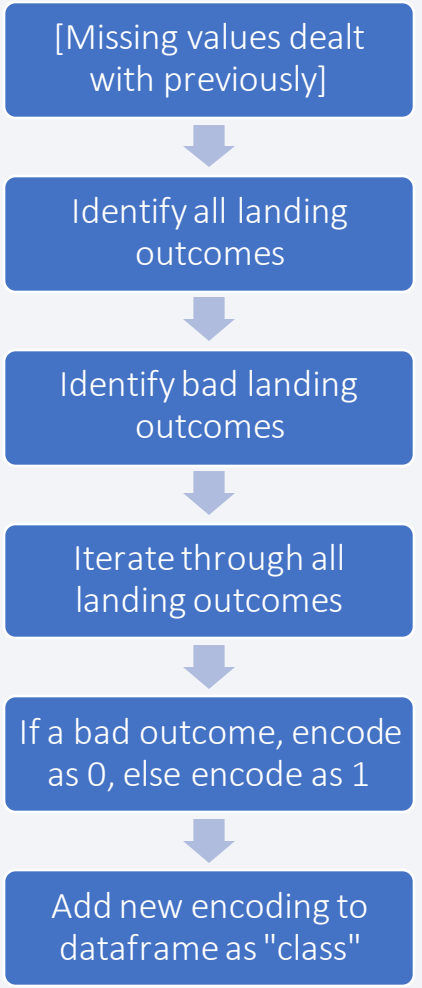
Data Collection - Scraping

- Used the requests library to scrape data from https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches
- Used BeautifulSoup to parse the content returned in the response
- The parsed data was added to a pandas dataframe and then exported to a CSV file
- https://github.com/delphinusuk/ibm-ds-pro-capstone/blob/9499761c563116123f02c582cf491713aa542fab/Data%20Collection%20with%20Web%20Scraping.ipynb



Data Wrangling

- A classification variable ("class") was created to encode the landing outcome as either 0 (bad) or 1 (good). The variable is the target variable that the classification algorithm will need to predict.
- https://github.com/delphinusuk/ibm-ds-pro-capstone/blob/9499761c563116123f02c582cf491713aa542fab/Data%20Wrangling.ipynb



EDA with Data Visualization

- Here we perform Exploratory Data Analysis using Matplotlib visualization techniques and Feature Engineering using Pandas.
- Specifically, the relationships between Flight number, Launch site, Payload mass and orbit type were investigated using a variety of graphs types, such as scatter plots, bar graphs and line graphs.
- All visualizations were color-coded by "class" so the effects of the variables and their relationships on the launch outcome were visible.
- The features were engineered using one-hot-encoding (which turns categorical data into numeric data, suitable for the classification algorithms) and then all data was cast to a float64 type.
- https://github.com/delphinusuk/ibm-ds-pro-capstone/blob/9499761c563116123f02c582cf491713aa542fab/EDA%20with%20visualization.ipynb

EDA with SQL

- The following queries were performed:
  - SELECT DISTINCT Launch_Site **from SPACEXTBL**
  - SELECT * FROM SPACEXTBL WHERE Launch_Site LIKE 'CCA%' LIMIT 5
  - SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE Customer = 'NASA (CRS)'
  - SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE Booster_Version = 'F9 v1.1'
  - SELECT MIN(substr(Date,7,4)||'-'||substr(Date,4,2)||'-'||substr(Date,1,2)) AS FIRST_DATE FROM SPACEXTBL WHERE "Landing _Outcome" = 'Success (ground pad)'
  - SELECT Booster_Version FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000 AND "Landing _Outcome" = 'Success (drone ship)'
  - SELECT Mission_Outcome, COUNT(*) FROM SPACEXTBL GROUP BY Mission_Outcome
  - SELECT DISTINCT Booster_Version FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL)
  - SELECT substr(Date,7,4) AS YEAR, substr(Date,4,2) AS MONTH, "Landing _Outcome", Booster_Version, Launch_Site FROM SPACEXTBL WHERE substr(Date,7,4)='2015' AND "Landing _Outcome" = 'Failure (drone ship)'
  - SELECT "Landing _Outcome", Count(*) FROM SPACEXTBL WHERE "Landing _Outcome" IN ('Success','Success (drone ship)','Success (ground pad)') AND substr(Date,7,4)||substr(Date,4,2)||substr(Date,1,2) BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY "Landing _Outcome" ORDER BY (Count(*)) DESC;

- https://github.com/delphinusuk/ibm-ds-pro-capstone/blob/9499761c563116123f02c582cf491713aa542fab/EDA%20(SQLLite%20version).ipynb

Build an Interactive Map with Folium

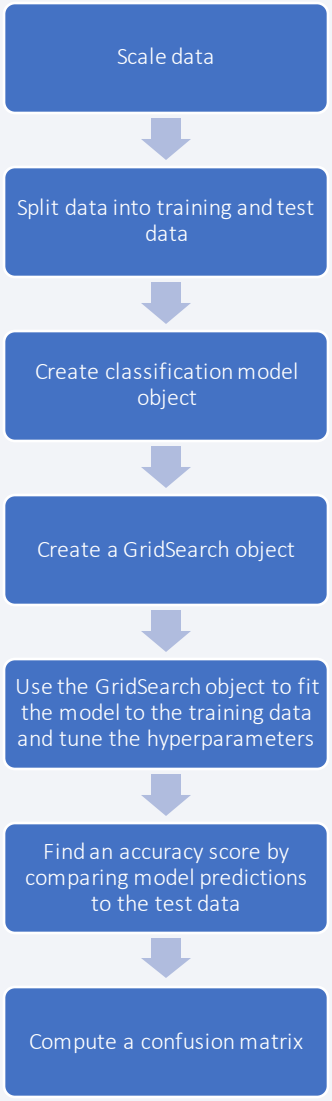
- Circles and Markers were added to an interactive Folium map to indicate launch locations.
- A MarkerCluster was added for each site to visualize the launch outcomes at each site.
- A line was drawn between a launch site and the coast and label with the distance was added to show the proximity of the two. Additional lines help users visualize distances to other important features that you want to avoid, such as roads, railways and towns/cities etc.
- https://github.com/delphinusuk/ibm-ds-pro-capstone/blob/9499761c563116123f02c582cf491713aa542fab/Interactive%20visual%20analytics.ipynb

Build a Dashboard with Plotly Dash

- An interactive dashboard was created to allow users to investigate the effects of launch site, payload mass and booster type on the launch outcome (good or bad).
- Launch site was selectable from a drop down menu and a range of payload masses could be selected using a slider control.
- A pie chart showed either the successful outcome for all launch sites or the proportion of good and bad outcomes for any one selected launch site.
- A scatter chart showed how the launch outcome varied by the selected site and payload range and the data points were color-coded by booster type.
- https://github.com/delphinusuk/ibm-ds-pro-capstone/blob/9499761c563116123f02c582cf491713aa542fab/Interactive%20visual%20analytics%20and%20dashboard.ipynb

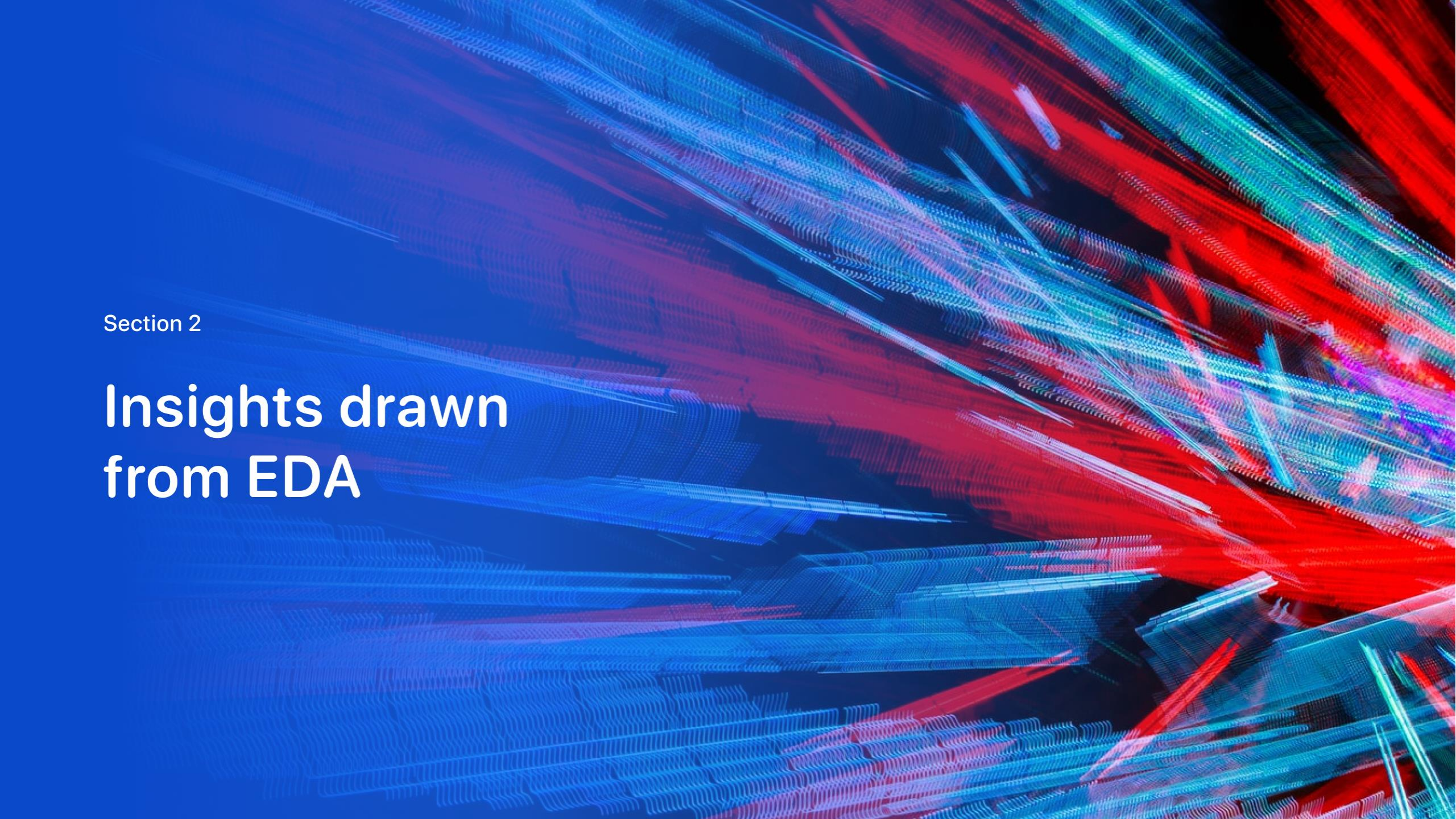
Predictive Analysis (Classification)

- Follow the process in the flowchart for the following classification algorithms:
  - Logistic regression
  - Support vector machines
  - Decision tree
  - K-nearest neighbors
- https://github.com/delphinusuk/ibm-ds-pro-capstone/blob/2fcb9ac40a446101c05917d663b4e9b491de79fb/Machine%20Learning%20Prediction.ipynb



Results

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



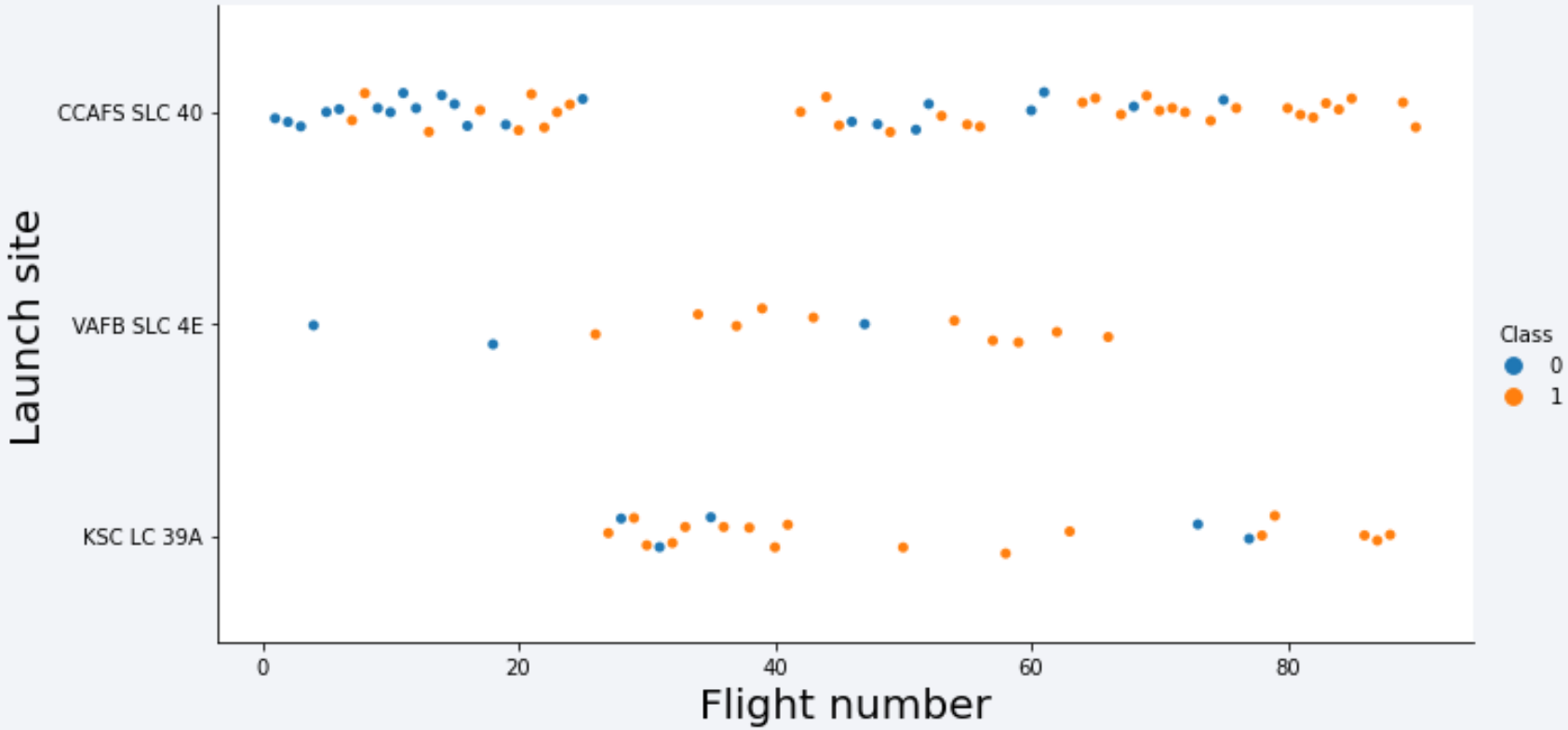
Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

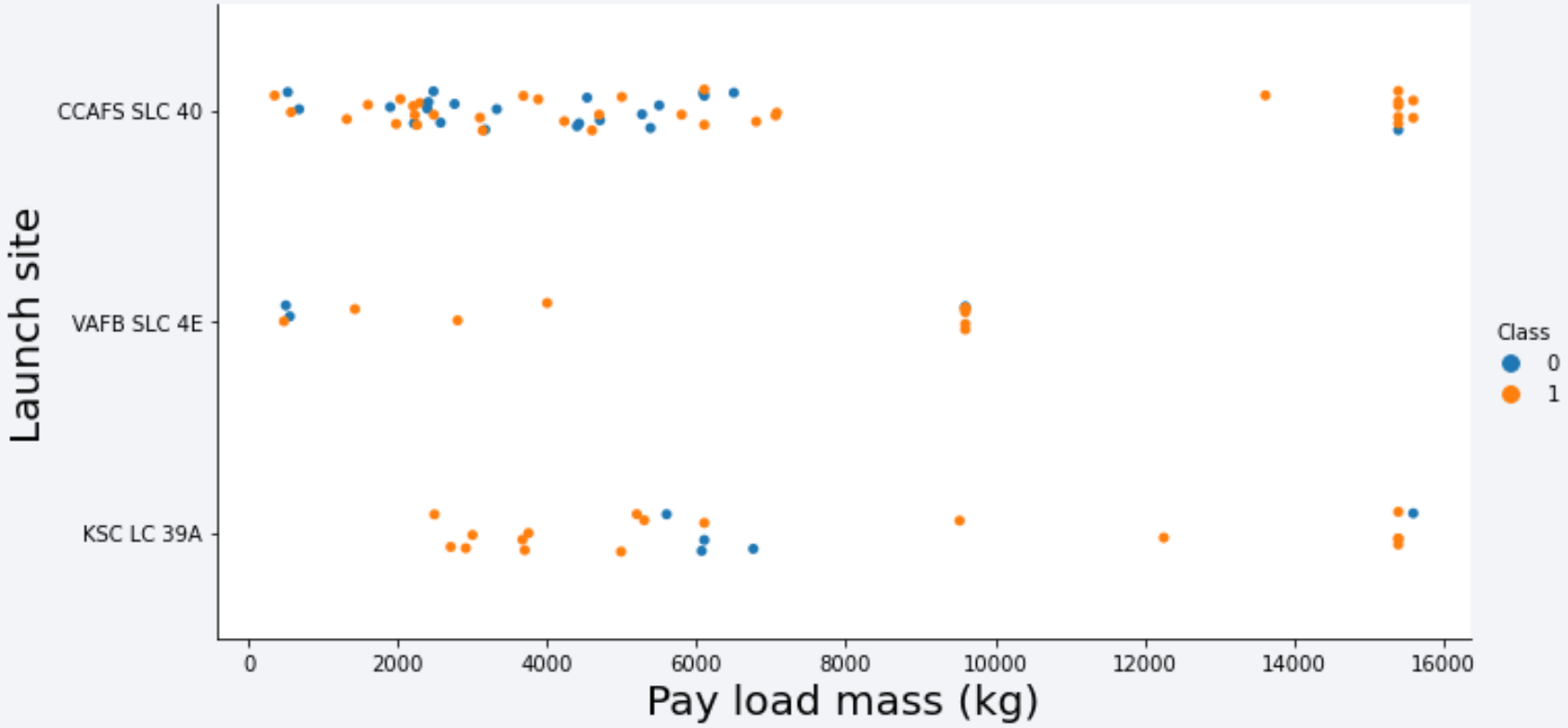
- Scatter plot of Flight Number vs. Launch Site, colored by outcome (blue = bad, orange = good)

- CCAFS… has had the most tests but VAFB… and KSC have the higher success rate (probably because they were used for many of the later flights where reliability might have improved).



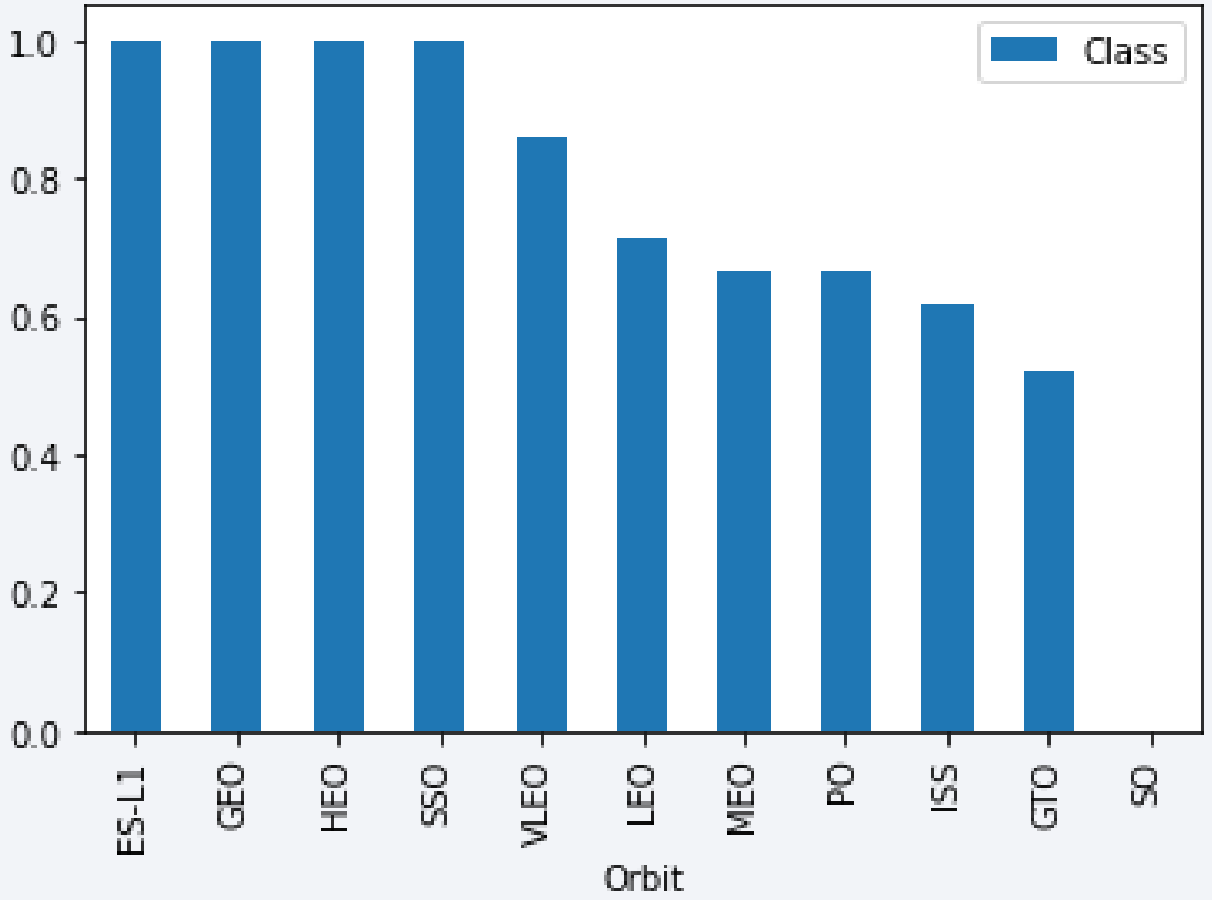
Payload vs. Launch Site

- Scatter plot of Payload vs. Launch Site, colored by outcome (blue = bad, orange = good)
- VAFB… has not had many heavy flights.
- Launches with higher payloads were more successful (possibly because they were only carried out when the engineers knew that the reliability had improved and the extra cost of the payload would be safer).



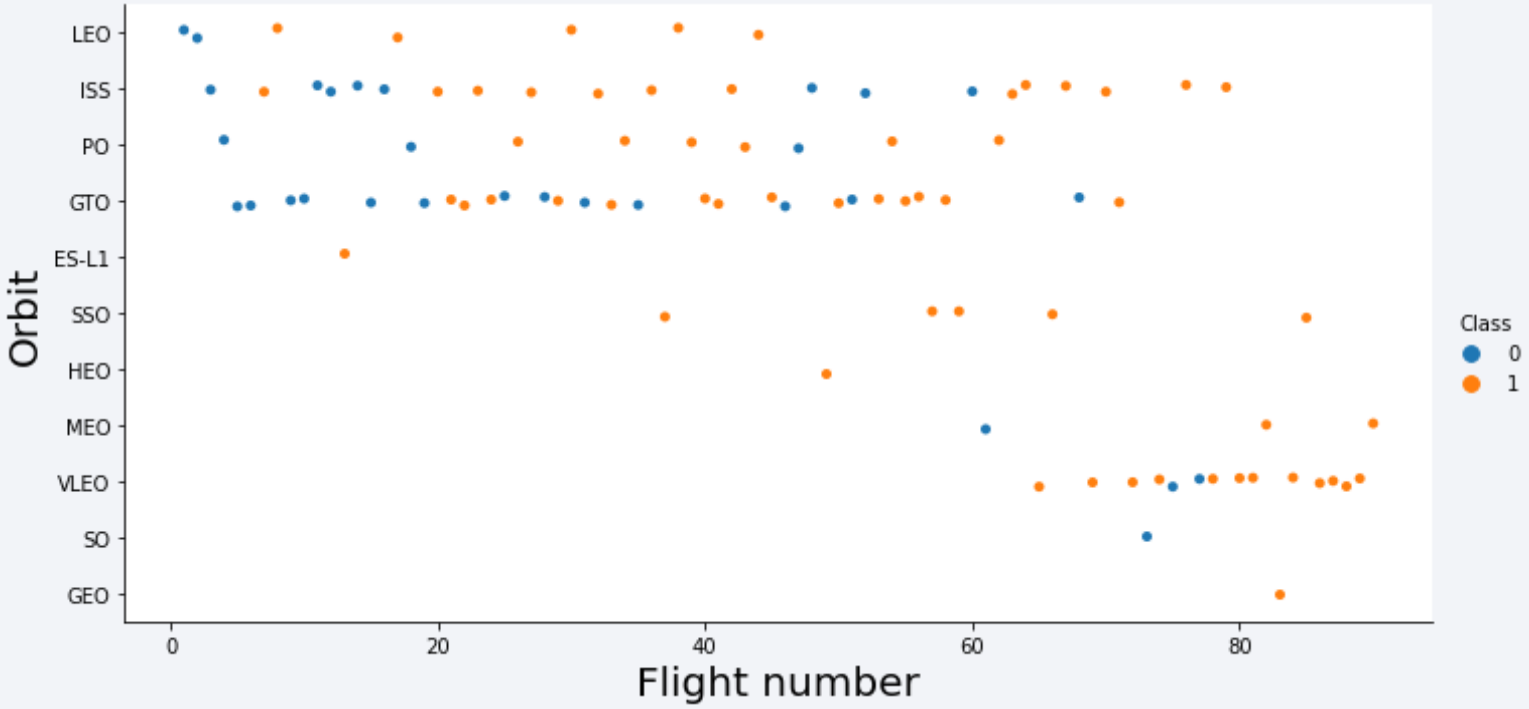
Success Rate vs. Orbit Type

- Bar chart for the success rate of each orbit type
- Higher orbits (GEO, HEO) seem to have higher rates of success.
- Note that SSO = SO so actually that orbit isn't 100% successful



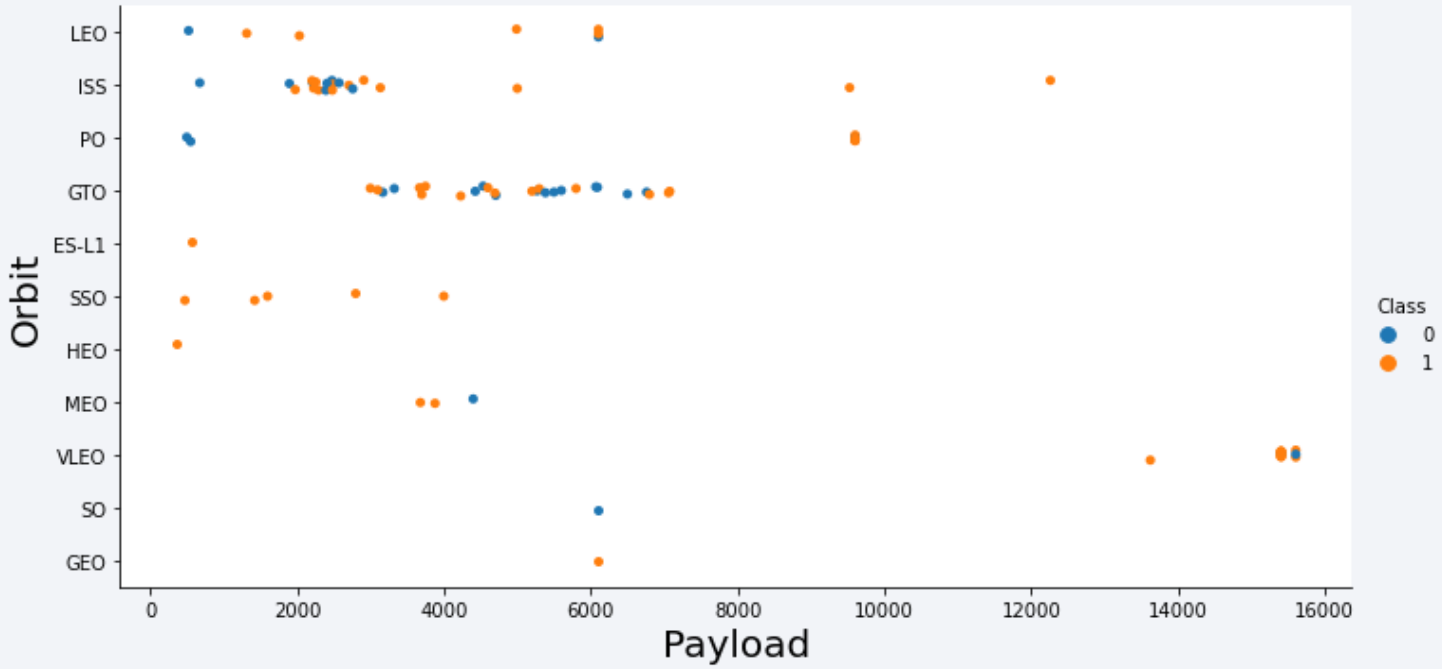
Flight Number vs. Orbit Type

- Scatter point of Flight number vs. Orbit type, colored by outcome (blue = bad, orange = good)
- The outcomes have tended to improve across all orbits as the number of flights increased.



Payload vs. Orbit Type

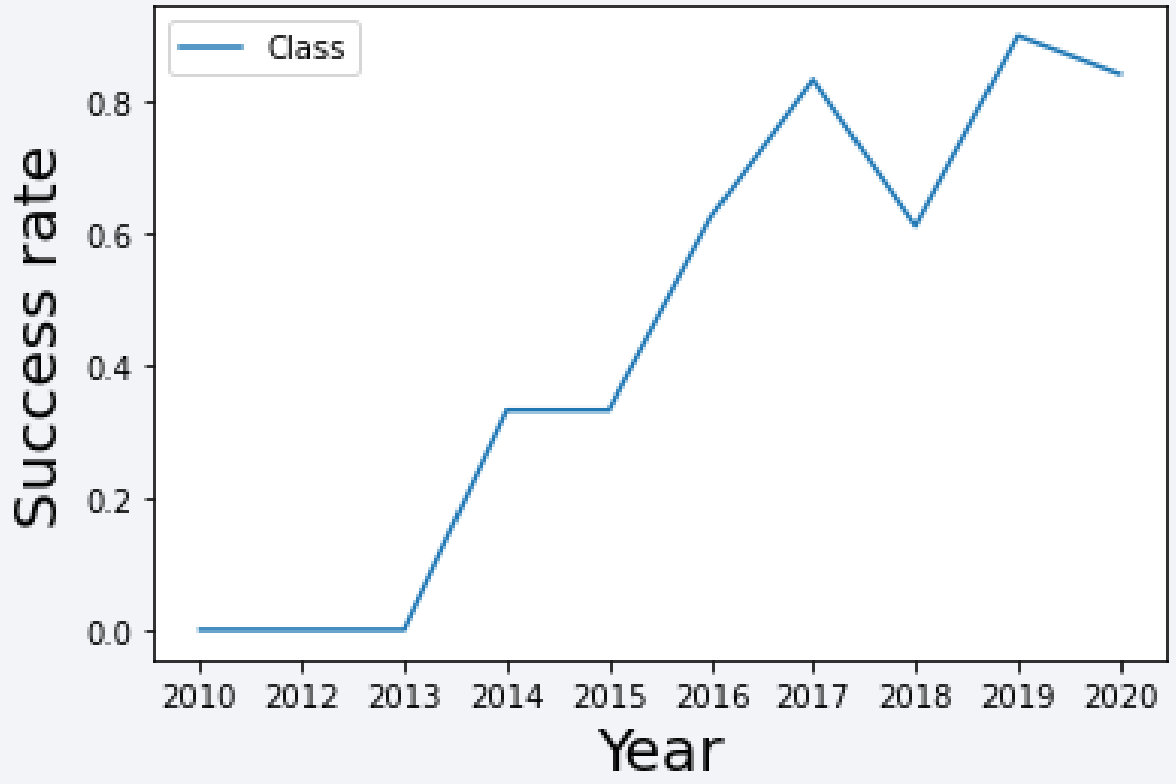
- Scatter point of payload vs. orbit type, colored by outcome (blue = bad, orange = good)
- There are no clear patterns of increased success with increased payload for any given orbit.
- SSO orbit is appears consistently successful (note that SSO = SO so actually it isn't 100% successful).



Launch Success Yearly Trend

- Line chart of yearly average success rate

- The success rate has been steadily increasing since 2013, although there was a dip around 2018.



All Launch Site Names

- The names of the launch sites
  - SELECT DISTINCT Launch_Site FROM **SPACEXTBL**
- SELECT DISTINCT selects the unique values from the specified column.

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

Launch Site Names Begin with 'CCA'

- 5 x records where launch sites begin with CCA
  - SELECT * FROM SPACEXTBL WHERE Launch_Site LIKE 'CCA%' LIMIT 5
- *, selects all; WHERE filters the data based on the pattern specified by the LIKE command; and LIMIT limits the output to 5 records.

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing _Outcome
04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
08-12-2010	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
22-05-2012	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
01-03-2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

- The total payload carried by boosters from NASA
  - SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE Customer = 'NASA (CRS)'
- SUM sums the payload values in the set of records filtered by Customer according to the condition specified in the WHERE command.

SUM(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1
  - SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE Booster_Version = 'F9 v1.1'
- Similar method to the last slide but using AVG and a different filter condition.

AVG(PAYLOAD_MASS__KG_)
2928.4

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad
  - SELECT MIN(substr(Date,7,4)||'-'||substr(Date,4,2)||'-'||substr(Date,1,2)) AS FIRST_DATE FROM SPACEXTBL WHERE "Landing _Outcome" = 'Success (ground pad)'
- With SQLLite there is no date type so dates are strings and you have to use the above substr command and patterns to correctly order the date returned by each record so that you can used the MIN command to find the first one.

FIRST_DATE
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- List the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000
  - SELECT Booster_Version FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000 AND "Landing_Outcome" = 'Success (drone ship)'
- The BETWEEN command is useful for filtering records by some range of values within a column.

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

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes
  - SELECT Mission_Outcome, COUNT(*) FROM SPACEXTBL GROUP BY Mission_Outcome
- COUNT(*) returns the count, GROUP BY aggregates the data according to the specified variable.

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

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass
  - SELECT DISTINCT Booster_Version FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL)
- A subquery is used to find the maximum payload mass across all records [using MAX()] and that returned value then becomes the filter with which to select all booster versions that have carried that mass.

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

2015 Launch Records

- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
  - SELECT substr(Date,7,4) AS YEAR, substr(Date,4,2) AS MONTH, "Landing _Outcome", Booster_Version, Launch_Site FROM SPACEXTBL WHERE substr(Date,7,4)='2015' AND "Landing _Outcome" = 'Failure (drone ship)'
- More SQLLite date handling. The AND condition is used to create two filters with each one working one value in a specific column.

YEAR	MONTH	Landing _Outcome	Booster_Version	Launch_Site
2015	01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
2015	04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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

- Rank the count of **successful** landing outcomes between the date 2010-06-04 and 2017-03-20, in descending order
  - SELECT "Landing _Outcome", Count(*) FROM SPACEXTBL WHERE "Landing _Outcome" IN ('Success','Success (drone ship)','Success (ground pad)') AND substr(Date,7,4)||substr(Date,4,2)||substr(Date,1,2) BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY "Landing _Outcome" ORDER BY (Count(*)) DESC;
- More SQLLite date handling, plus DISTINCT plus a filter range using BETWEEN and aggregation of outcomes using GROUP. [Note my assignment clearly asked for successful outcomes only.]

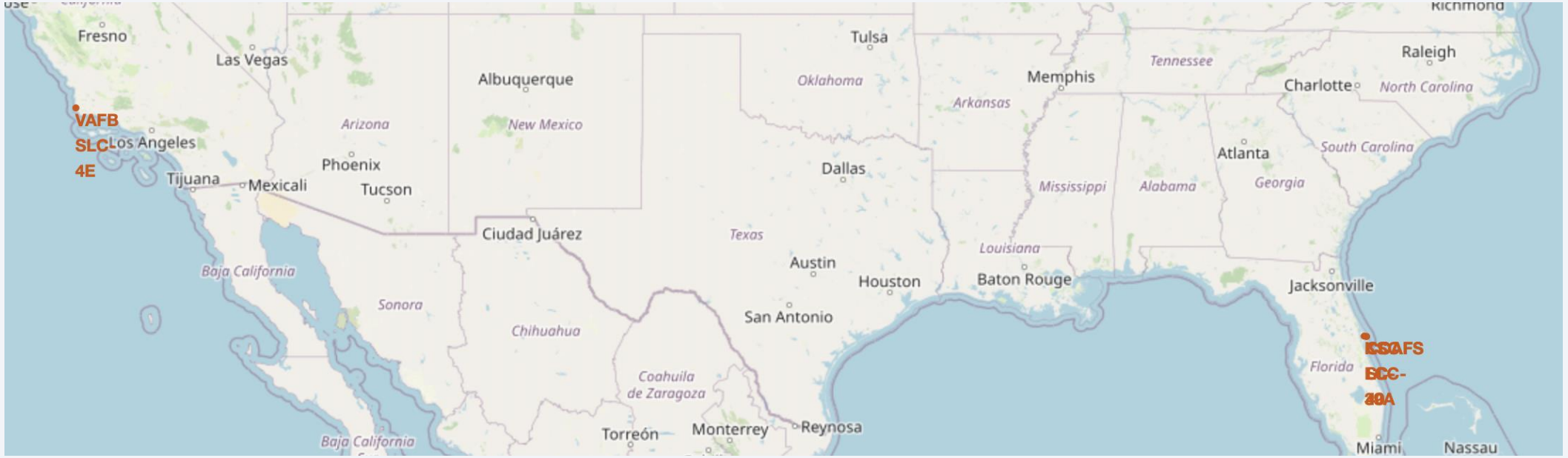
Landing _Outcome	Count(*)
Success (drone ship)	4
Success (ground pad)	2

Section 4

Launch Sites Proximities Analysis

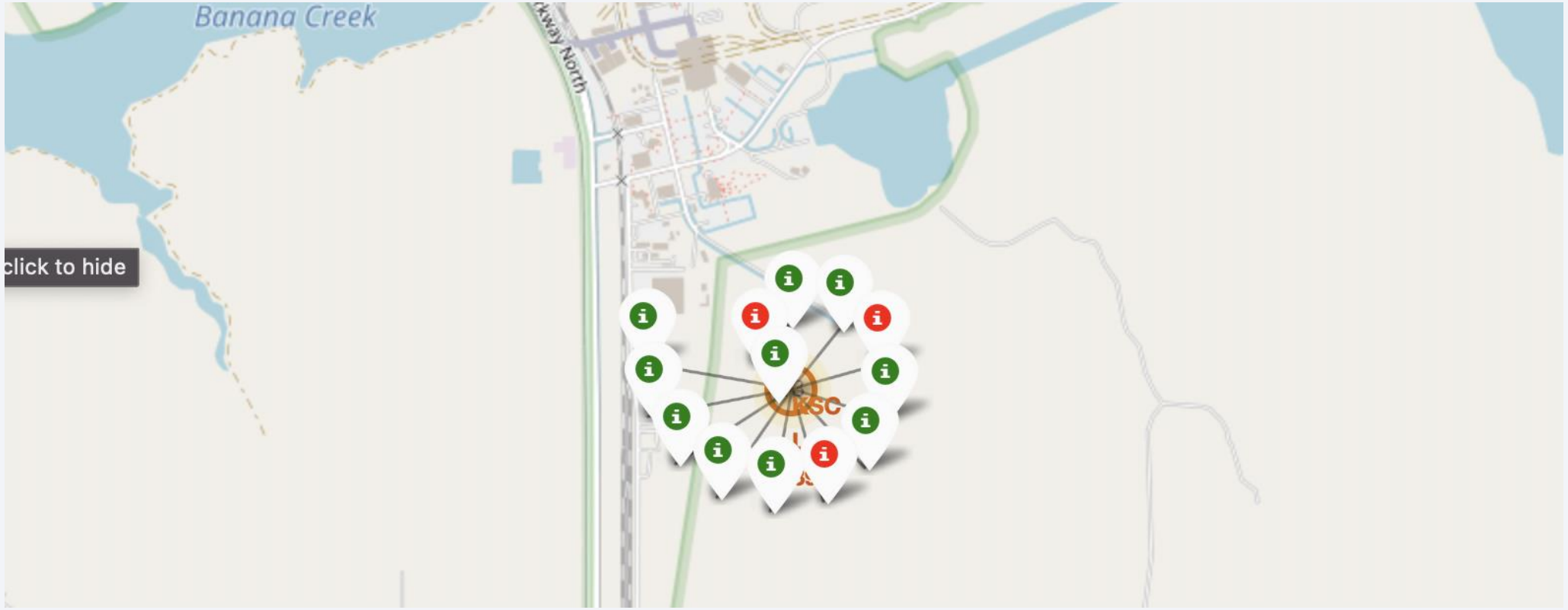
Launch site locations

The launch sites are next to the coast and major oceans.



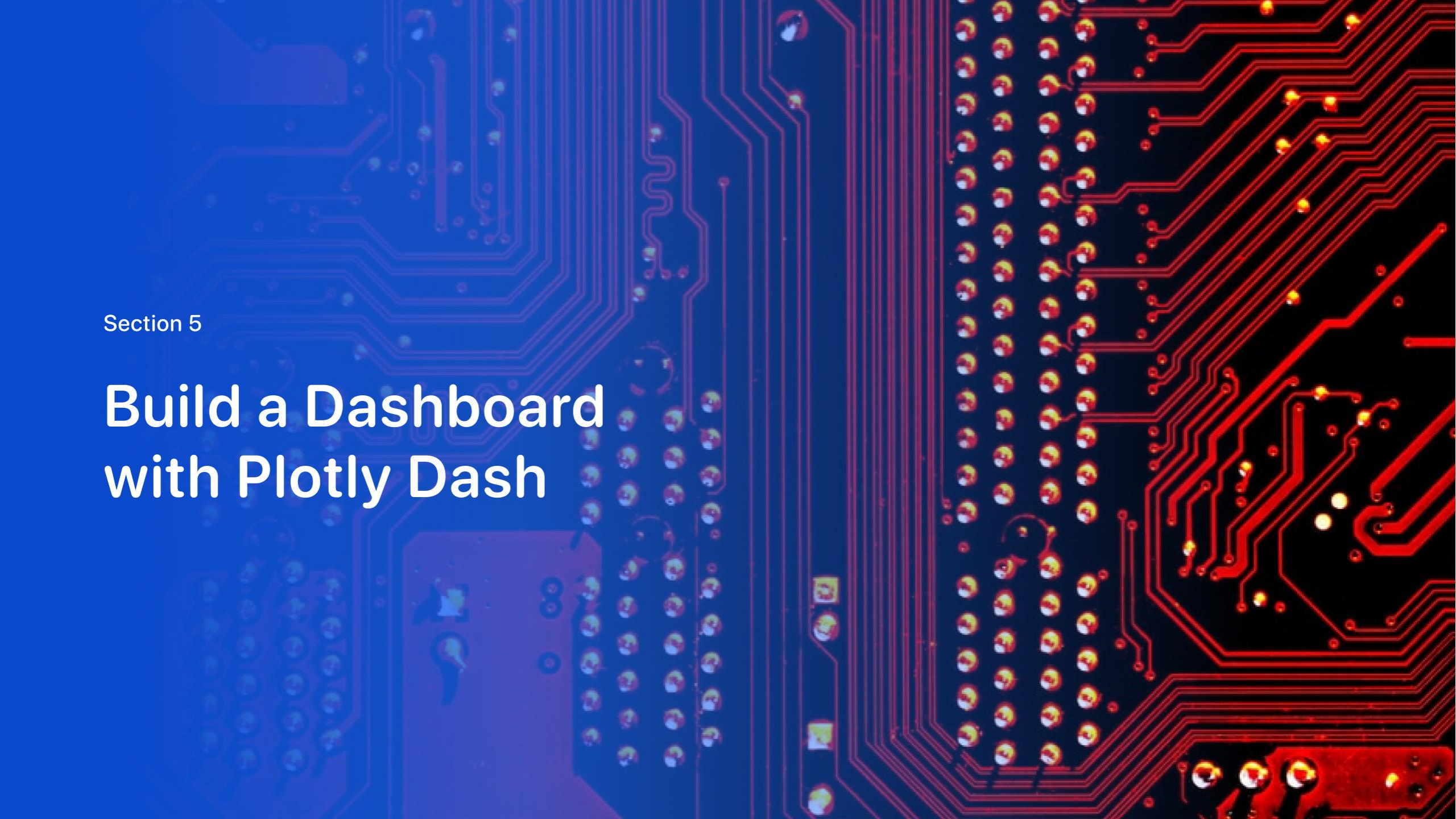
Launch site outcomes

Clustered set of markers showing launch outcomes for launch site KCS.



Proximity of launch site to other land/coastal features

It could be important to know how far a launch site is from other land/coastal features. [Honestly, this was an exceptionally dull task. I hated it. So there was no way I was going to waste my time drawing other lines to other features. It has nothing to do with predicting launch outcomes so I didn't bother.]

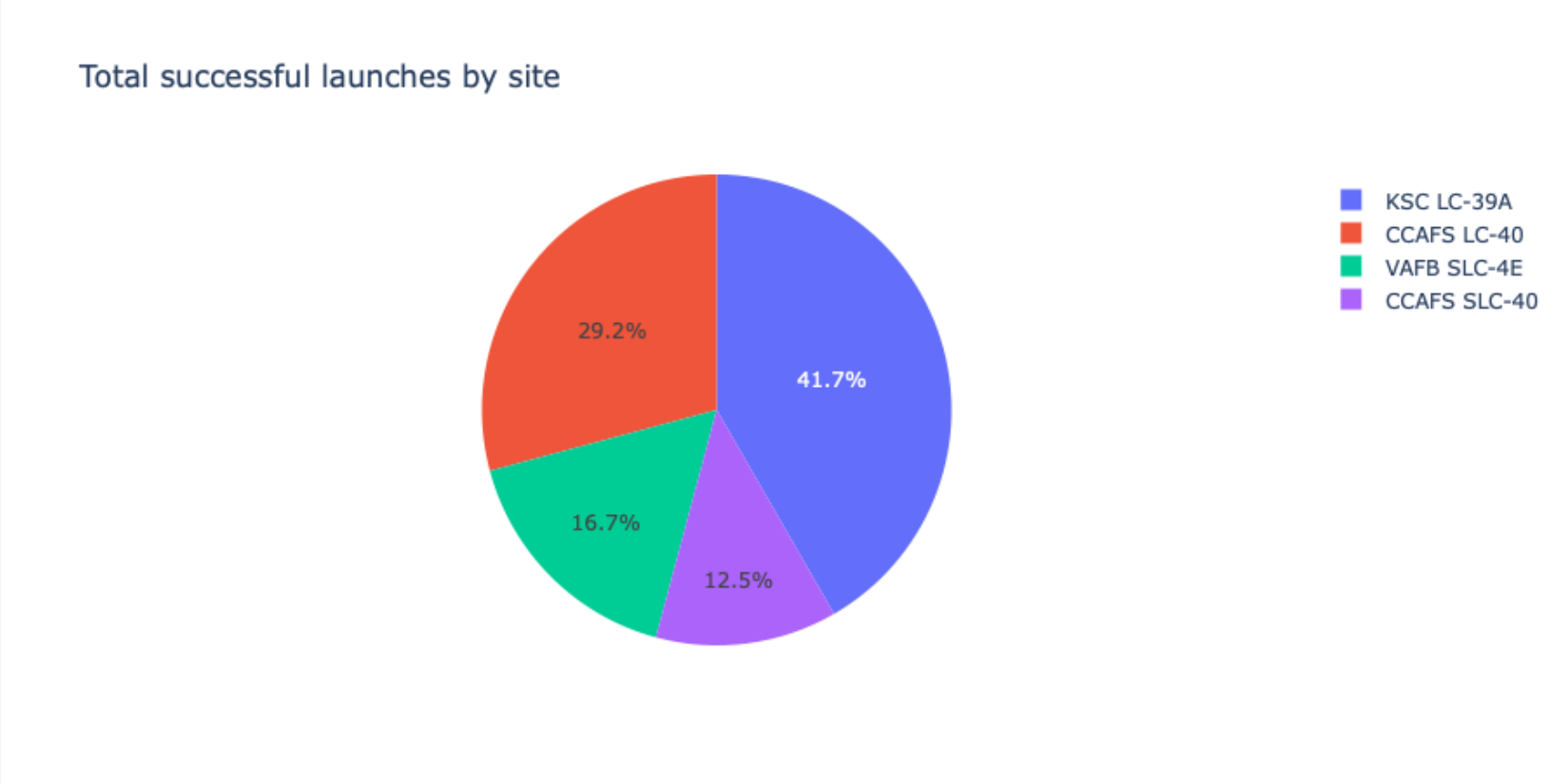


Section 5

Build a Dashboard with Plotly Dash

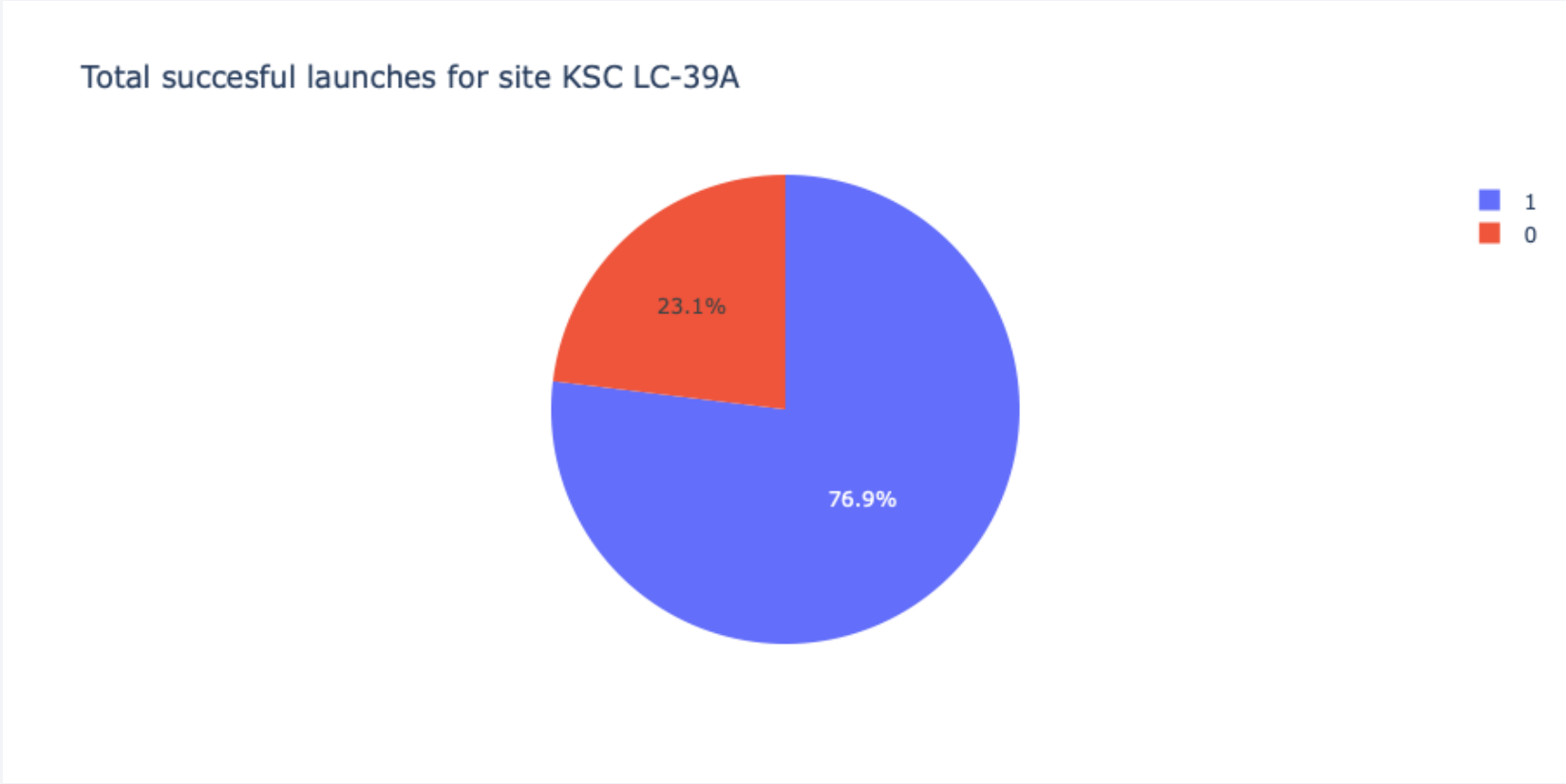
Launch successes across all sites

- More successful launches have taken place at KSC.



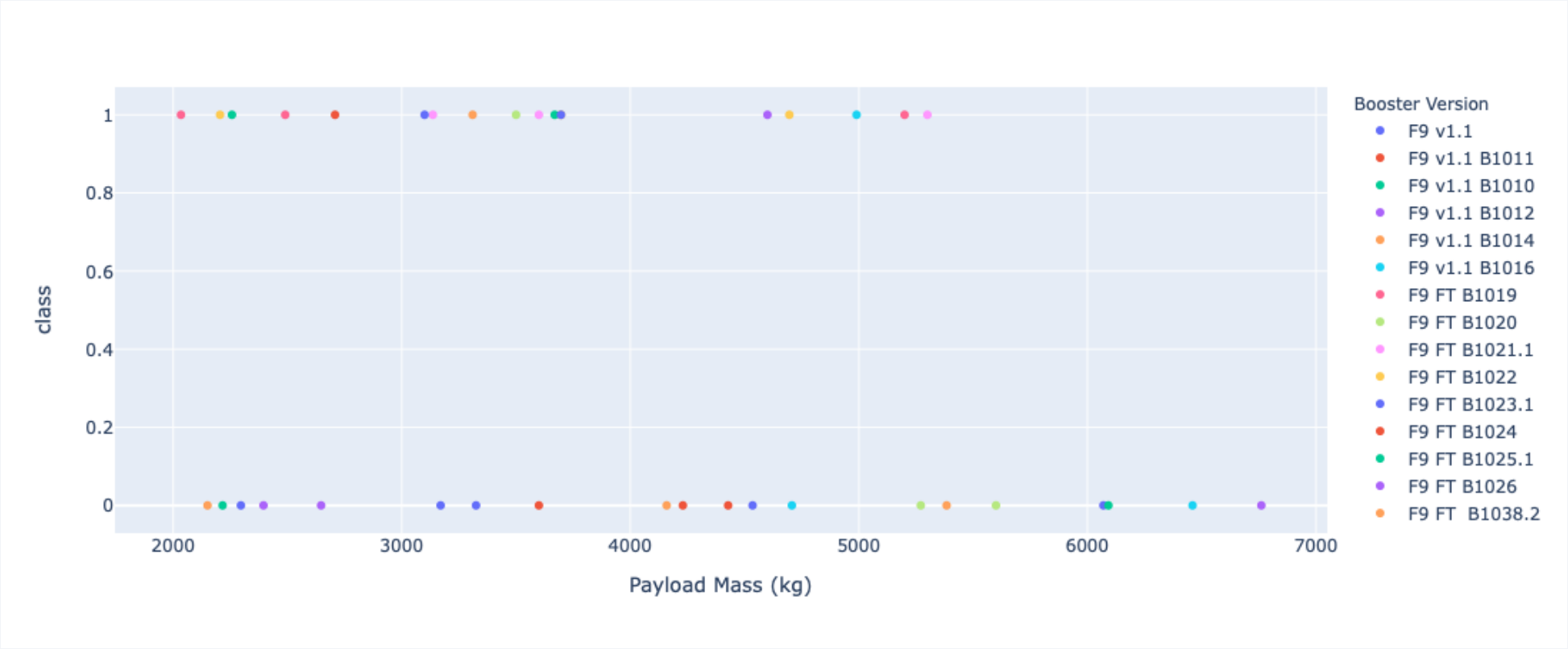
Launch site with the highest success rate

- The KSC launch site has the highest success rate so it isn't just the overall number of launches that gives it the highest proportion of successful launches in the previous graph.



Payload (2000 – 7000 kg) vs. Launch Outcome for all sites

- Need to drill down further (e.g. may be recategorize booster types into v1.1 and FT) to make sense of this data.

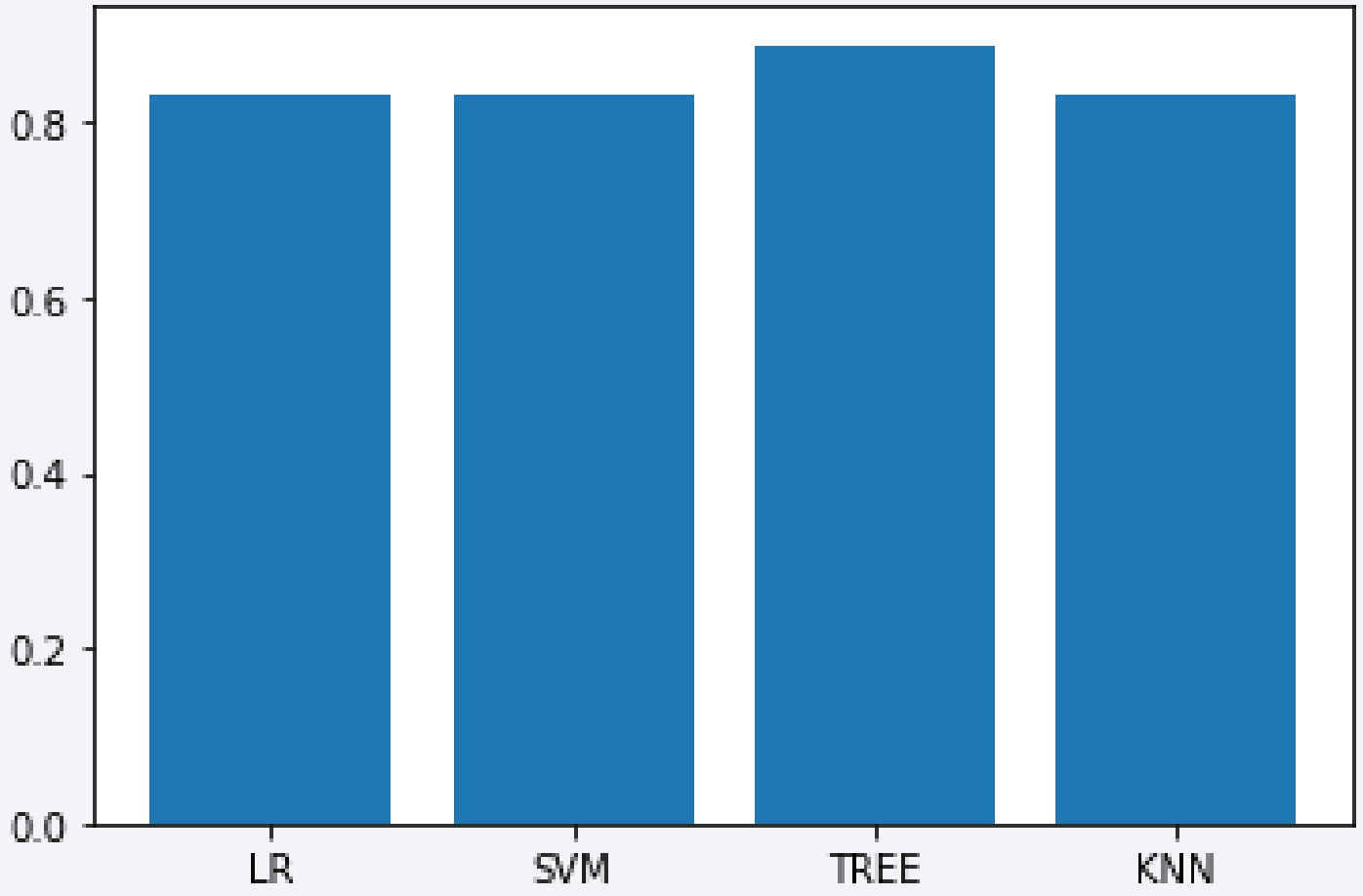


Section 6

Predictive Analysis (Classification)

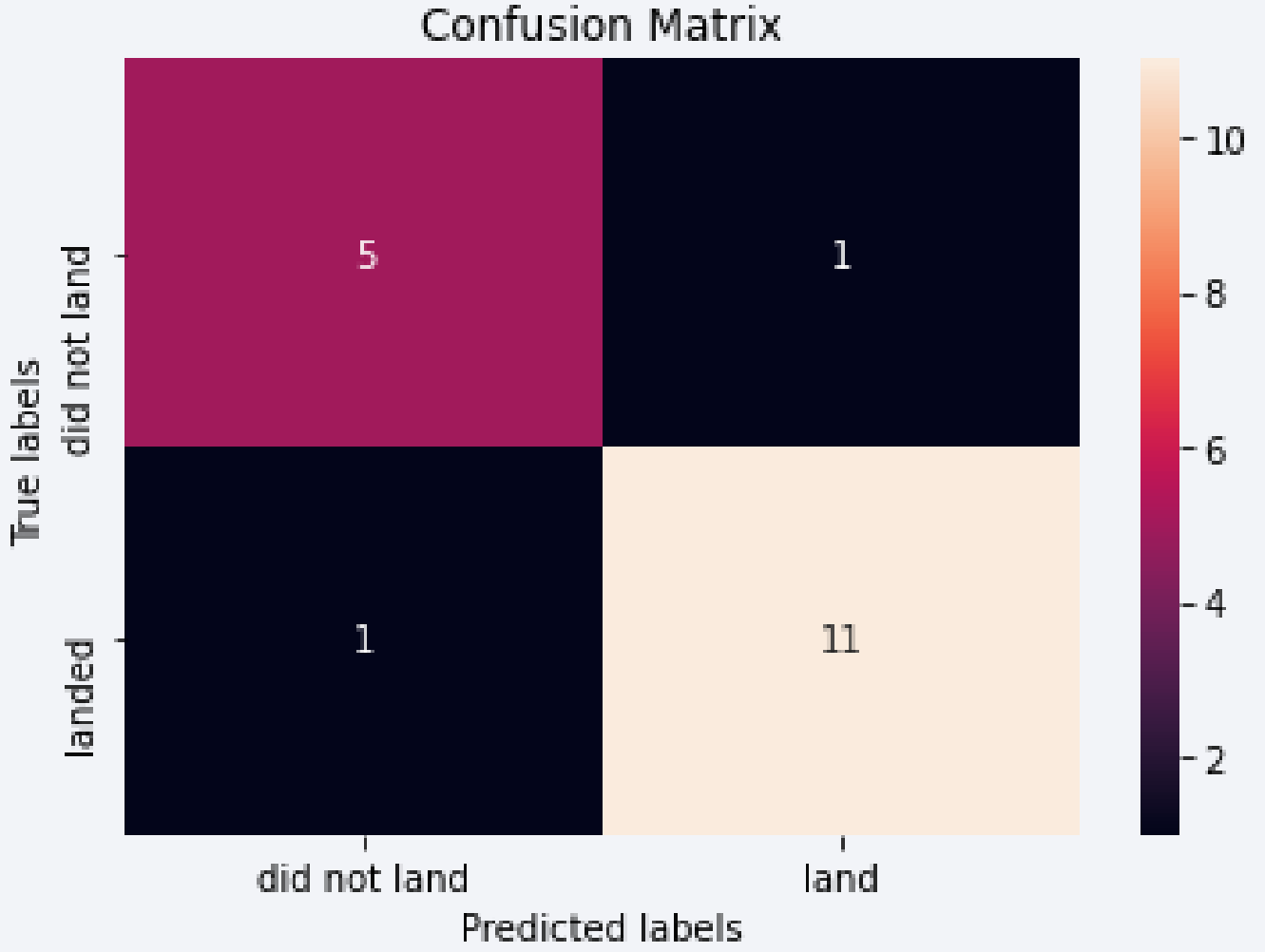
Classification Accuracy

- Model accuracy for each classification models
- The decision tree (TREE) classification model showed the highest accuracy at around 0.89.
- The tuned hyperparameters were: {'criterion': 'entropy', 'max_depth': 10, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 10, 'splitter': 'random'}



Confusion Matrix

- The decision tree showed the fewest combined false positives and false negatives (one of each, so two in total) so, in that respect, was the best model.



Conclusions

- Launch success rates increase with time.
- The KSC launch site has the best launch outcomes.
- A decision tree is the best classifier for predicting launch outcomes based on information such as the launch site, payload, orbit and booster types etc.

Appendix

- Nothing to add here, other than I'd be massively criticized at work if I produced a presentation like this with over forty slides. A handful of slides is plenty, even to those wanting a "deep dive".
- Also, we could have put all our links to our notebooks here, not in the main presentation (the appendix is a really valid place to put them).

Thank you!