IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Paul Webb
2024-05-22

Outline

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

Executive Summary

- **Summary of methodologies**
  - Data Collection was performed both via an API and Web scraping techniques
  - Data Wrangling was performed to standardize data and replace missing values
  - Exploratory Data Analysis using SQL queries was performed (SQLite)
  - Exploratory Data Analysis via Visualization was performed (Seaborn)
  - Visualization via interactive maps marking key data was performed (Folium)
  - An interactive dashboard was created to explore data (Plotly Dash)
- **Summary of all results**
  - Landing sites should be place near coastlines and the equator
  - Kennedy Space Center (KSC LC-39A) launch site offers the highest chance of a successful first stage recovery
  - The highest chance of a first stage recovery success are for ES-L1, GEO, HEO and SSO orbits
  - Success rates increase with the flight number (time/experience)

Introduction

- **Project background and context**

Commercial space launch providers have been entering the market and ramping up launch capabilities over the past 2-3 decades. One of the most successful is SpaceX who have reduced launch costs by focusing on reusability.

The aim of this project is to use machine learning and visualization techniques to predict the probability of a successful first stage landing. Successfully landing the first stage provides the ability to reuse and to reduce the overall launch cost. This will provide data for the business case for an alternate entrant into the commercial space launch sector.

- **Problems you want to find answers**
  - What factors determine the successful landing and reuse of the first stage of a launch?
  - Provide dashboards and visualizations to allow exploration of launch data
  - Provide a predictive model of successful first stage recovery based on relevant factors

Section 1

Methodology

Methodology

Executive Summary

- **Data collection methodology:**
  - Publicly available launch data was collected
  - Data was collected through a combination of API calls to SpaceX and web-scraping of Wikipedia pages

- **Perform data wrangling**
  - Data retrieved was converted to Pandas data frames
  - Data was filtered as appropriate (Falcon9 launch data only)
  - Data was explored and additional columns created as required to assist with analysis

Methodology

Executive Summary

- **Perform exploratory data analysis (EDA) using visualization and SQL**
  - Visualizations of key features and their relationships were created using the Seaborn library
  - Data was loaded into an SQLite database for exploratory analysis of locations, payloads, launch sites and how each affect first stage recovery
- **Perform interactive visual analytics using Folium and Plotly Dash**
  - Folium was used to generate interactive maps exploring launch sites and nearby facilities
  - A Plotly Dash application was created to explore successful launches by site, payload mass and booster version

Methodology

Executive Summary

- **Perform predictive analysis using classification models**
    - Data related to launch and recovery attempts were loaded
    - Data were split into test and train data sets
    - Alternative prediction models were trained and evaluated:
        - Logistic Regression
        - Support Vector Machine
        - Decision Tree
        - K Nearest Neighbors
    - The alternatives were scored and the most accurate was chosen

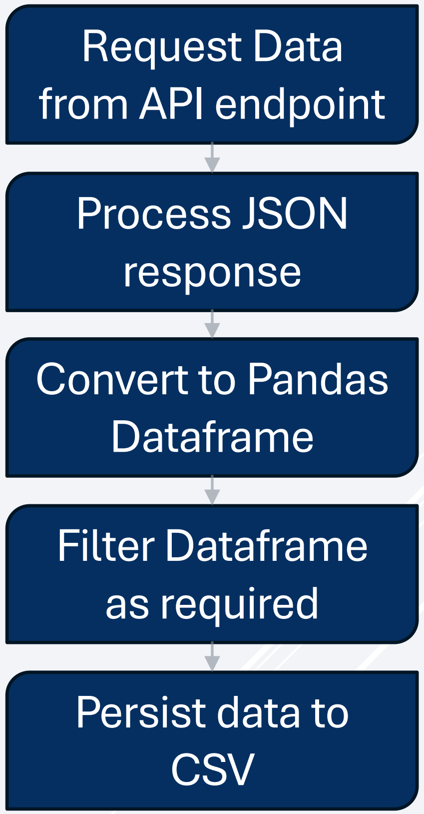
Data Collection

- **How data sets were collected:**
  - Data sets were collected using a combination of SpaceX API access and web-scraping techniques
  - The “requests” library was used to retrieve the data in both cases
  - API data was read in JSON format and converted to Pandas dataframes
  - Web data was processed with “BeautifulSoup” to extract relevant data to load into Pandas

- **The following slides detail the collection process with flow charts and further details about the location of the collected data**

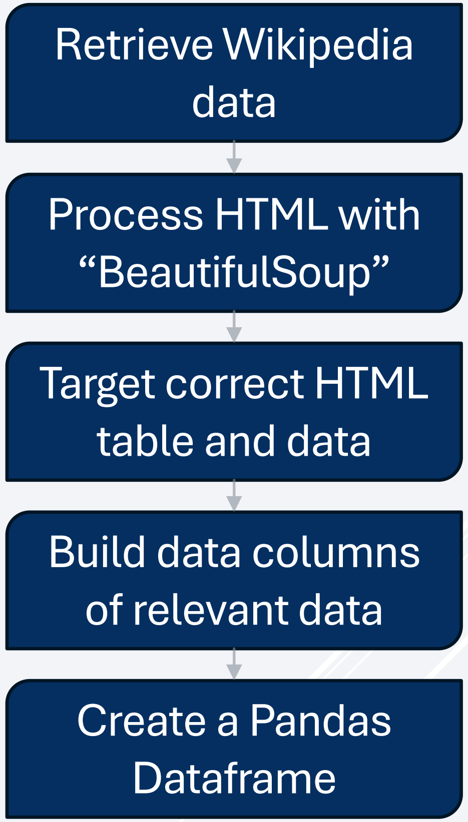
Data Collection – SpaceX API

- Standard REST GET calls to the SpaceX API endpoints were made using the “requests” library to retrieve JSON data
- The JSON data was processed and converted into Pandas Dataframes
- Dataframes were filtered to only include Falcon 9 Launches
- Resulting data was persisted as CSV for further processing in later stages
- Jupyter notebook detailing processing and endpoints used is available from: https://github.com/technobok/ibm_data_science_professional_certificate/blob/main/applied_datascience_capstone(10)/module1/1%20Collecting%20the%20Data/jupyter-labs-spacex-data-collection-api.ipynb



Data Collection – Scraping

- “requests” was used to retrieve data from Wikipedia detailing the launch details of SpaceX launches
- Retrieved HTML data was processes with “BeautifulSoup”
- The correct HTML table was selected
- Columns of relevant data were built by iterating through the table
- The dictionary of columns was converted to a Pandas Dataframe
- Jupyter notebook detailing processing and URL’s used is available from: https://github.com/technobok/ibm_data_science_professional_certificate/blob/main/applied_datascience_capstone(10)/module1/1%20Collecting%20the%20Data/jupyter-labs-webscraping.ipynb



Data Wrangling

- Previously retrieved CSV data was loaded
- Fields with missing data were identified
- Numerical vs Categorical fields were identified
- Types of unsuccessful first stage landings with identified
- A simplified "Outcome" column was created showing success/failure of first stage recovery
- Jupyter notebook detailing the data wrangling steps is available at: https://github.com/technobok/ibm_data_science_professional_certificate/blob/main/applied_datascience_capstone(10)/module1/2%20Data%20Wrangling/labs-jupyter-spacex-Data%20wrangling.ipynb

Retrieve CSV data as Pandas Dataframe

↓

Explore data using isnull, dtypes, value_counts etc.

↓

Identify unsuccessful first stage landing types

↓

Build a new "Outcomes" column as a list

↓

Add "Outcomes" columns to the Pandas Dataframe

↓

Persist data to CSV

EDA with Data Visualization

- **The Seaborn visualization library was used to create plots**
- **Key features that were expected to be related to successful first stage recovery were:**
  - Flight number
  - Payload mass
  - Launch site
- **To visualize, the following plots of these features were created:**
  - Payload mass per Flight number, colored by success of first stage recovery
  - Launch site per Flight number, colored by success of first stage recovery
  - Launch site per Payload mass, colored by success of first stage recovery
  - Orbit type per Flight number, colored by success of first stage recovery

EDA with Data Visualization

- The Jupyter notebook detailing visualizations created is available at:

https://github.com/technobok/ibm_data_science_professional_certificate/blob/main/applied_datascience_capstone(10)/module2/2%20Exploratory%20Analysis%20Using%20Pandas%20and%20Matplotlib%20(Visualisation)/edadataviz.ipynb

EDA with SQL

- **Launch data was loaded into an SQLite database for exploration**
- **The following queries were run:**
  - Listing of distinct Launch Sites
  - Listing of launches from sites beginning with CCA
  - Aggregate sum of mass launched for a customer (NASA)
  - Average payload for the F9 v1.1 rocket
  - Date of the first successful ground landing
  - Names of booster versions successfully landing on a drone ship with payload mass between 4000 and 6000 KG.
  - Total number of missions grouped by mission outcome (success, failure)

EDA with SQL

- **The following queries were run (continued):**
  - List of booster versions that have carried a mass equal to the maximum payload mass
  - List of records, with month names, that failed landing on drone ships within 2015
  - Ranked summary of number of missions for each landing outcome type
- **The Jupyter notebook containing the database created and the SQL queries run is available at:** https://github.com/technobok/ibm_data_science_professional_certificate/blob/main/applied_datascience_capstone(10)/module2/1%20Exploratory%20Data%20Analysis%20Using%20SQL/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- **Folium was used to generate interactive maps showing:**
  - Locations of launch attempts
  - Locations annotated with number of launch attempts and with success/failure markers
  - Markers show distance and lines to nearby facilities: Nearest Coastline, Nearest city, Nearest Railway line, Nearest Highway
- **These markers were added to determine the factors used for a launch site: location for rocket trajectories, local available facilities etc.**
- **The Jupyter notebook containing the Folium maps generated is available at:** https://github.com/technobok/ibm_data_science_professional_certificate/blob/main/applied_datascience_capstone(10)/module3/1%20Interactive%20Visual%20Analytics%20and%20Dashboard%20(Folium%2BPlotly)/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

Plots/graphs and interactions you have added to a dashboard:

- A Plotly Dash application was built to explore the relationships between launch site, payload mass and booster version and how they relate to successful first stage recovery
- Graphs added to the application were:
  - Pie chart of distribution of total successful launches by site…
  - …with drill down to see total launches for a site by success/failure
  - Scatter chart of Success/Failure by Payload mass, colored by booster version
- Interactions added were:
  - A dropdown list to select launch site, with an option for ‘All’ to visualize all launches from any site
  - A payload range selector to explore the first stage recovery as related to ranges of payload mass
- The source code for the Plotly Dash application can be found at: https://github.com/technobok/ibm_data_science_professional_certificate/blob/main/applied_datascience_capstone(10)/module3/1%20Interactive%20Visual%20Analytics%20and%20Dashboard%20(Folium%2BPlotly)/spacex_dash_app.py

Predictive Analysis (Classification)

A machine learning pipeline was created to predict the successful landing and reuse of the first stage

- The features were standardized using the StandardScaler
- The independent and dependent features were isolated
- The dataset was split into training and testing sets
- Several predictive methods were evaluated (Logistic Regression, Support Vector Machine, Decision Tree classifier, K Nearest Neighbours)
- For each method, Grid Search was used to determine the best parameters on the ‘train’ data
- The result of each method was evaluated on the ‘test’ data using its score() and a confusion matrix
- The best performing model was chosen
- The Jupyter notebook detailing the predictive analysis models tested is available at: https://github.com/technobok/ibm_data_science_professional_certificate/blob/main/applied_datascience_capstone(10)/module4/1%20Predictive%20Analysis%20(Classification)/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

1. Standardize dataframe features
2. Split independent (X) and dependent (Y) features
3. Split data into ‘test’ and ‘train’ sets
4. Per model, use GridSearch to find best parameters
5. Per model, evaluate using score() + confusion matrix
6. Select best performing model

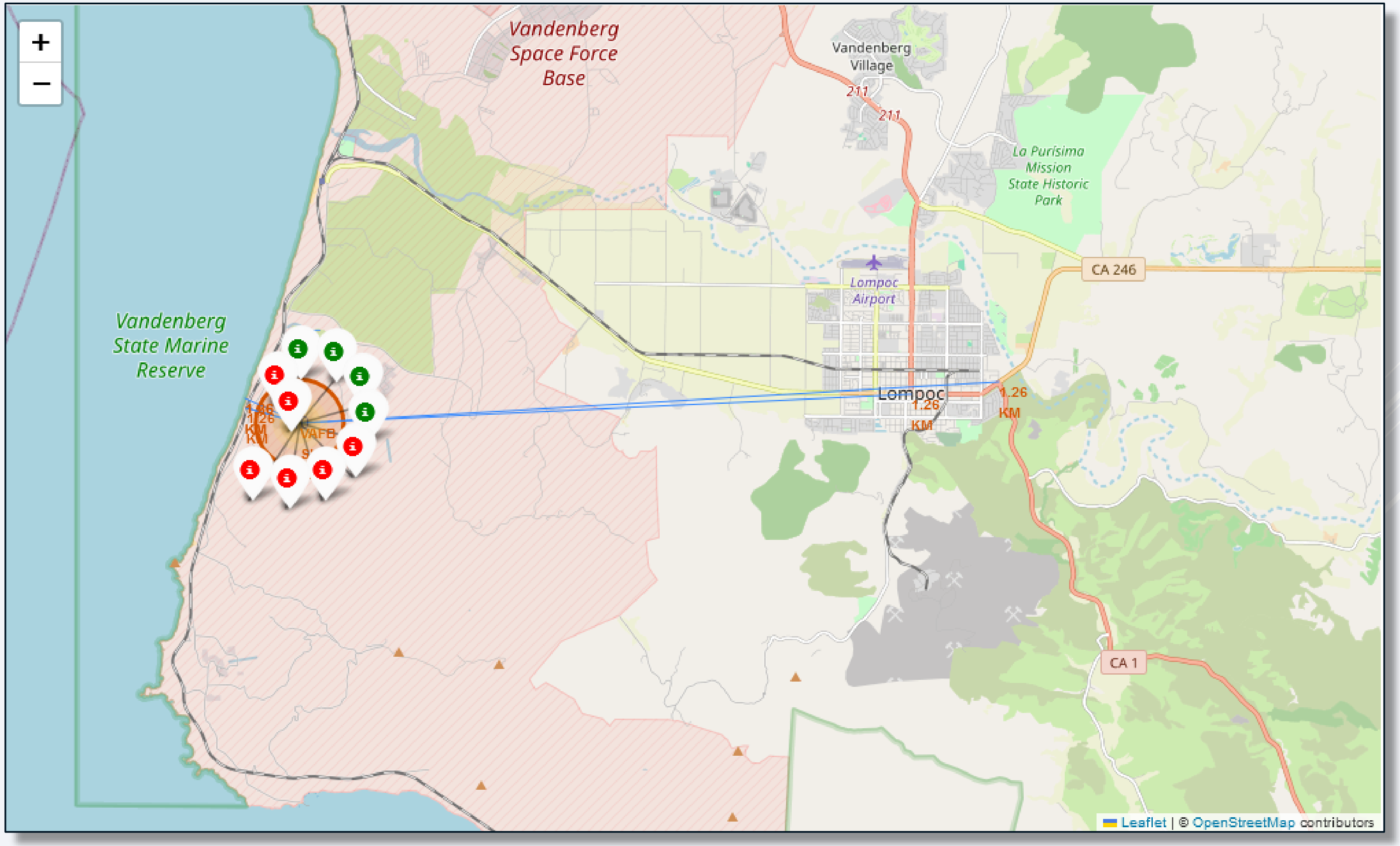
Results

- Exploratory data analysis results
    - The 4 launch sites used by SpaceX were determined
    - The total payload mass delivered for NASA is approaching 100 metric tons
    - The first successful ground pad landing was December 2015
    - There are 4 versions of Falcon 9 boosters that have delivered payloads from 4,000 to 6,000 kg
    - There were 2 failed drone ship attempted landings in 2015
    - The percentage of successful first stage landings are increasing over time
    - The chance of a successful first stage landing increases with payload mass

Results

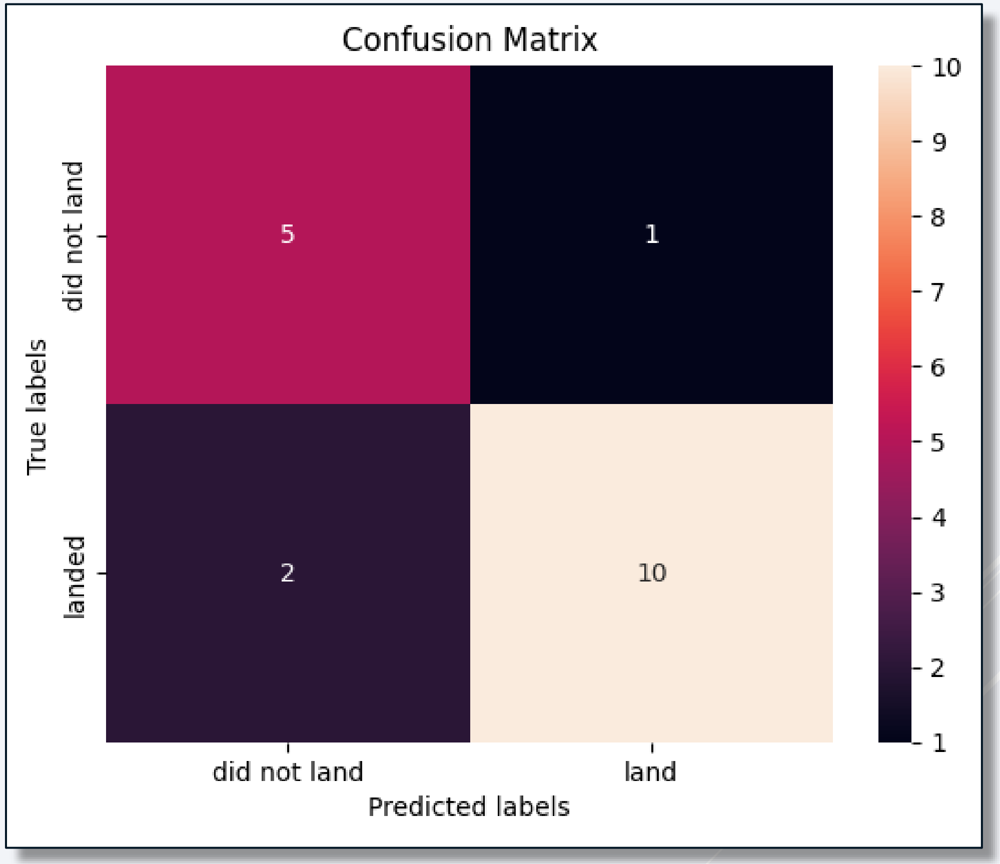
Interactive analytics demo in screenshots:

Demo screenshot of interactive Folium map generated showing number of successful/failed first stage recoveries launched from Vandenberg AFB, with markers to nearest facilities (coastline, rail line, city, highway)



Results

- Predictive analysis results
  - Decision Tree performed the best of the predictive models based on its scores in both the training and test data sets
  - A Confusion matrix on the training data is presented

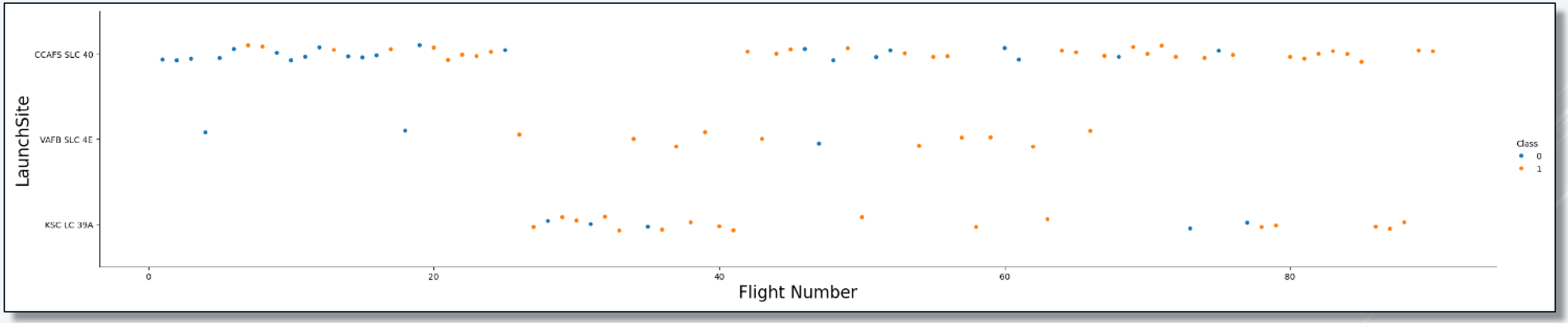


Section 2

Insights drawn from EDA

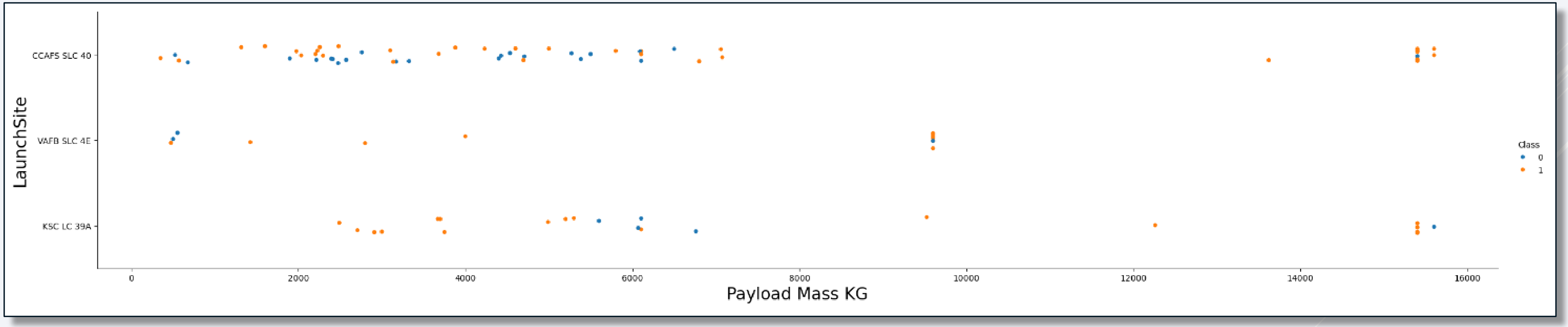
Flight Number vs. Launch Site

- A scatter plot of Flight Number vs. Launch Site
- Reliability of first stage recovery has increased at all launch sites over time



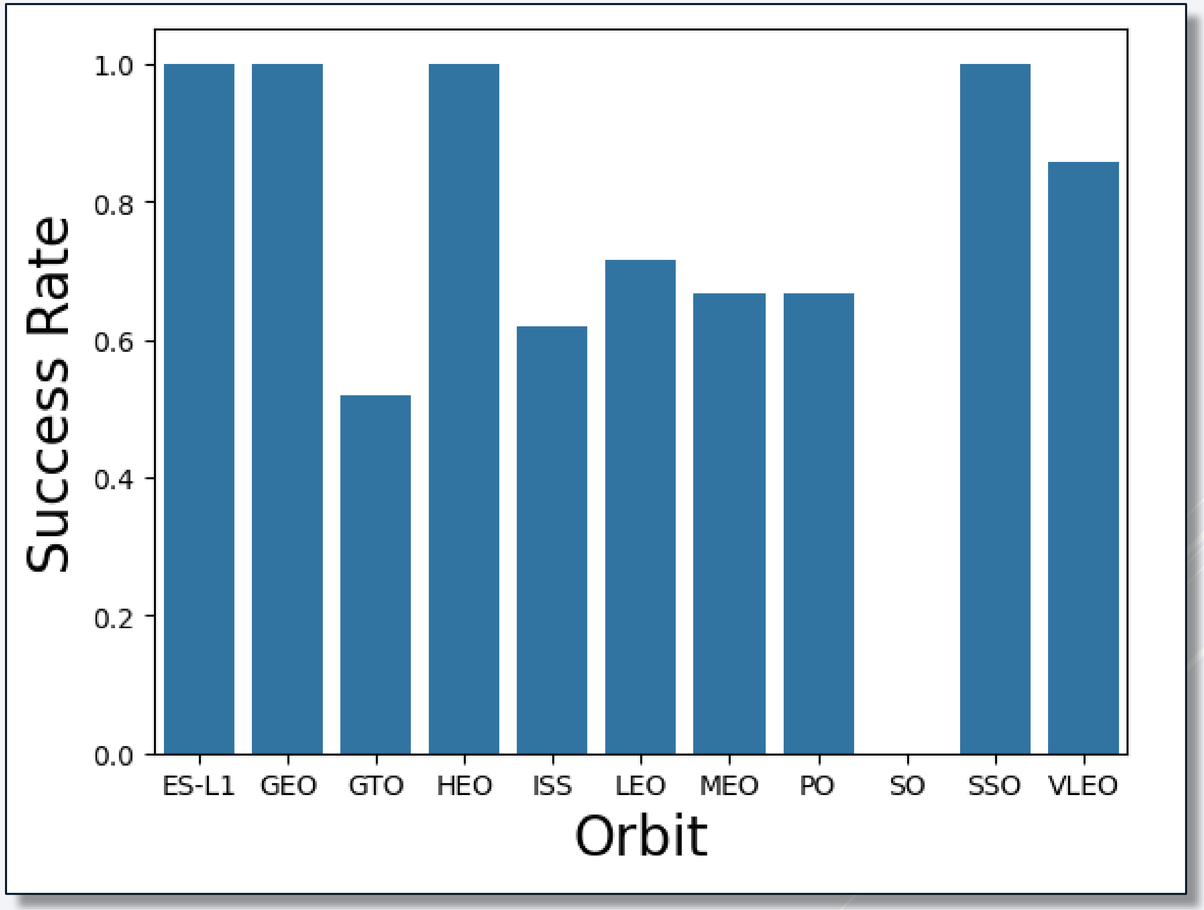
Payload vs. Launch Site

- A scatter plot of Payload vs. Launch Site
- VAFB launch site does not host missions with a payload mass over 10,000kg
- There are only 2 missions with over 10,000 kg payload where the first stage was not recovered



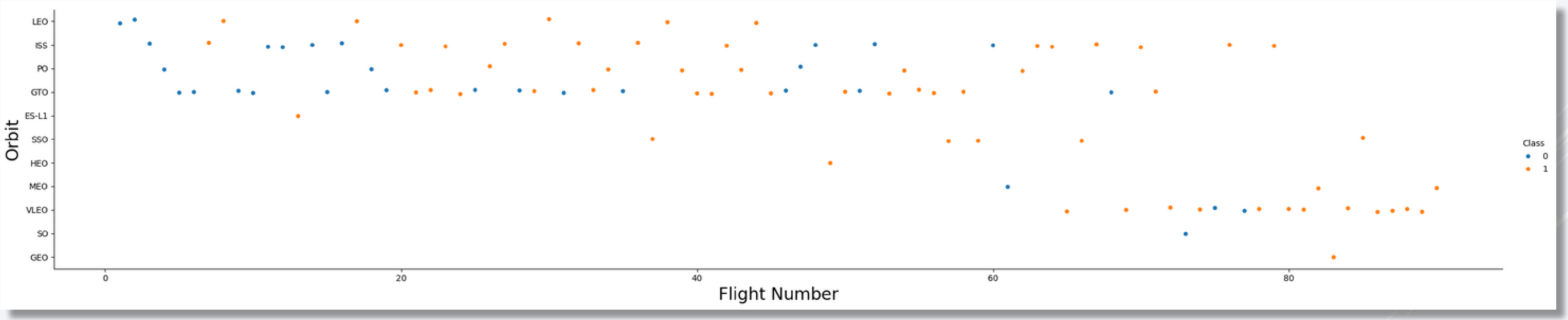
Success Rate vs. Orbit Type

- A bar chart for the success rate of each orbit type is presented
- There is a 100% first stage recovery success rate for ES-L1, GEO, HEO and SSO orbits



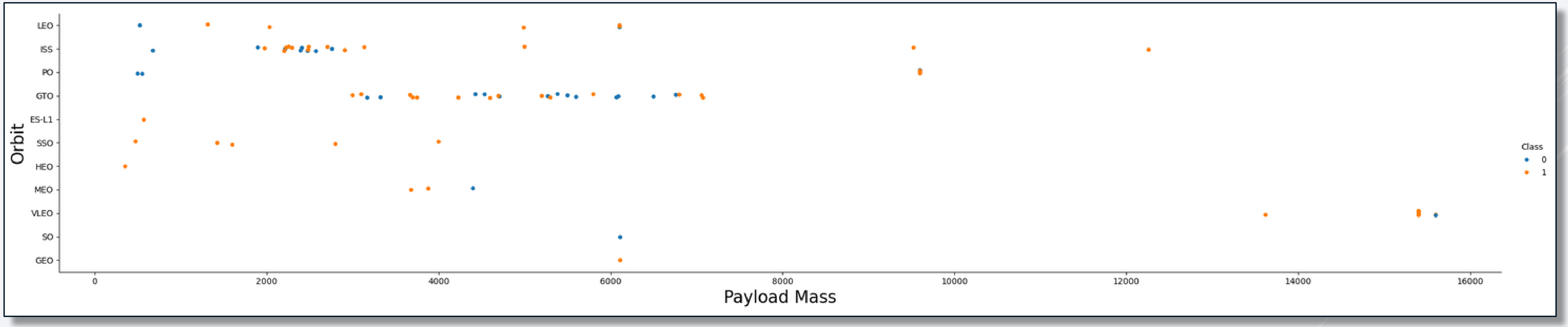
Flight Number vs. Orbit Type

- A scatter plot of Flight number vs. Orbit type
- Chance of first stage recovery is increasing for all orbit types over time (flight number)
- Recently there are fewer GTO missions instead VLEO missions are increasing



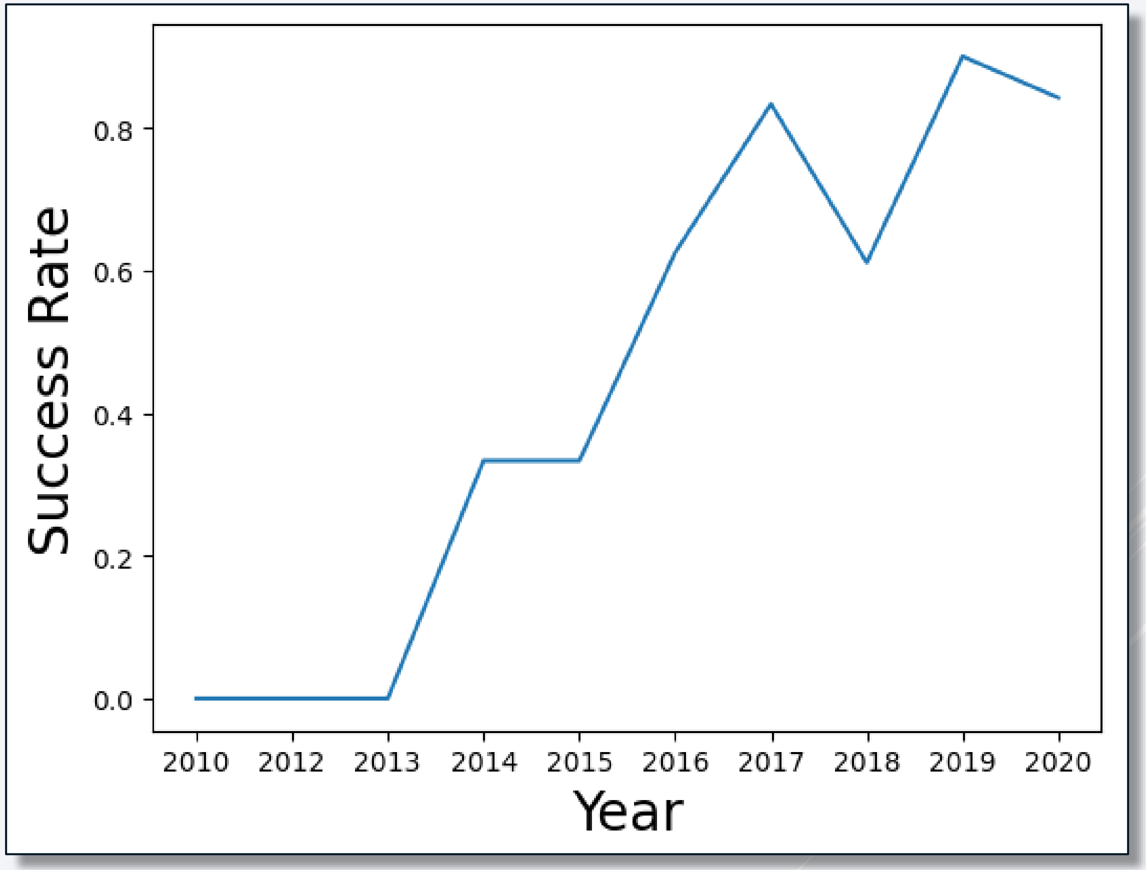
Payload vs. Orbit Type

- A scatter point of payload vs. orbit type
- Higher payload masses (over 8,000 kg) are only launched to ISS, PO and VLEO orbits and these appear to have a higher success rate of recovering the first stage



Launch Success Yearly Trend

- A line chart of yearly average success rate
- We observe a clear upward trend for the success of first stage recovery over time since 2013 to the present



All Launch Site Names

- A list of unique launch sites are presented
- ‘select distinct’ was used to create the list

Display the names of the unique launch sites in the space mission

```
%sql select distinct "Launch_Site" from SPACEXTBL
```

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

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

Launch Site Names Begin with 'CCA'

- Listing of 5 records where launch sites begin with `CCA` is presented
- The 'like' operator using a pattern with the '%' wildcard character matched the correct records
- 'limit' was used to return a maximum of 5 records only

Display 5 records where launch sites begin with the string 'CCA'

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

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
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

- Calculation for the total payload carried by boosters from NASA is presented
- The aggregate ‘sum’ function was used to total payload mass for all matching records
- The ‘like’ operator was used to match NASA launches

Display the total payload mass carried by boosters launched by NASA (CRS)

```
%sql select sum("PAYLOAD_MASS__KG_") as PayloadMass from SPACEXTBL where Customer like 'NASA%'
```

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

PayloadMass
99980

Average Payload Mass by F9 v1.1

- Calculation the average payload mass carried by booster version F9 v1.1 is presented
- The ‘avg’ aggregate function was used to calculate the average payload mass for the matching records

Display average payload mass carried by booster version F9 v1.1

```
%sql select avg("PAYLOAD_MASS__KG_") as PayloadMass from SPACEXTBL where Booster_Version like 'F9 v1.1%'
```

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

PayloadMass
2534.6666666666665

First Successful Ground Landing Date

- The date of the first successful landing outcome on ground pad was determined
- The aggregate ‘min’ function was used to find the earliest date of the matching successful ground pad records

```
%sql select min(Date) mindate from SPACEXTBL where "Landing_Outcome" = 'Success (ground pad)'
```

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

mindate
2015-12-22

Successful Drone Ship Landing with 4,000-6,000kg Payload

- A list of the names of boosters which have successfully landed on drone ship and had payload mass greater than 4,000 but less than 6,000 kg is presented
- Multiple expressions in the ‘where’ clause were used to apply the criteria
- ‘distinct’ is used to show unique Booster versions

List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

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

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

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

Total Number of Successful and Failure Mission Outcomes

- A calculation the total number of successful and failure mission outcomes is presented
- 'group by' was used to aggregate the result set by Mission_Outcome
- The mission outcome itself was listed followed by a column showing the 'count' of that mission type

List the total number of successful and failure mission outcomes

```
%sql select "Mission_Outcome", count("Mission_Outcome") as "Mission count" from SPACEXTBL group by "Mission_Outcome"
```

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

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

Boosters Carried Maximum Payload

- A list of the names of boosters which have carried the maximum payload mass is presented
- A subquery was used to determine the maximum payload mass
- The ‘where’ clause selects all missions having that maximum payload mass
- ‘distinct’ is used to show unique booster versions

```
%sql select distinct Booster_Version from SPACEXTBL where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEXTBL)
```

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

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

- A list of the failed landing outcomes on drone ships, their booster versions, and launch site names for in year 2015 is presented
- The ‘substr’ function was used to extract the year and month from the date field for presentation and comparison

```
%sql select substr(Date,0,5) as year, substr(Date, 6, 2) as month, * from SPACEXTBL where substr(Date,0,5)='2015' and Landing_Outcome = 'Failure (drone ship)'
```

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

year	month	Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS_KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2015	01	2015-01-10	9:47:00	F9 v1.1 B1012	CCAFS LC-40	SpaceX CRS-5	2395	LEO (ISS)	NASA (CRS)	Success	Failure (drone ship)
2015	04	2015-04-14	20:10:00	F9 v1.1 B1015	CCAFS LC-40	SpaceX CRS-6	1898	LEO (ISS)	NASA (CRS)	Success	Failure (drone ship)

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

- A rank of the count of landing outcomes between the date 2010-06-04 and 2017-03-20, in descending order is presented
- A grouped subquery is used to calculate the count of each Landing Outcome
- The outer query is used to sort the summarized subquery in descending order

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

```
%sql select Landing_Outcome, Landing_Count from (\
    select Landing_Outcome, count(Landing_Outcome) as Landing_Count from SPACEXTBL \
    where Date between '2010-06-04' and '2017-03-20' group by Landing_Outcome\
) t \
order by t.Landing_Count desc
```

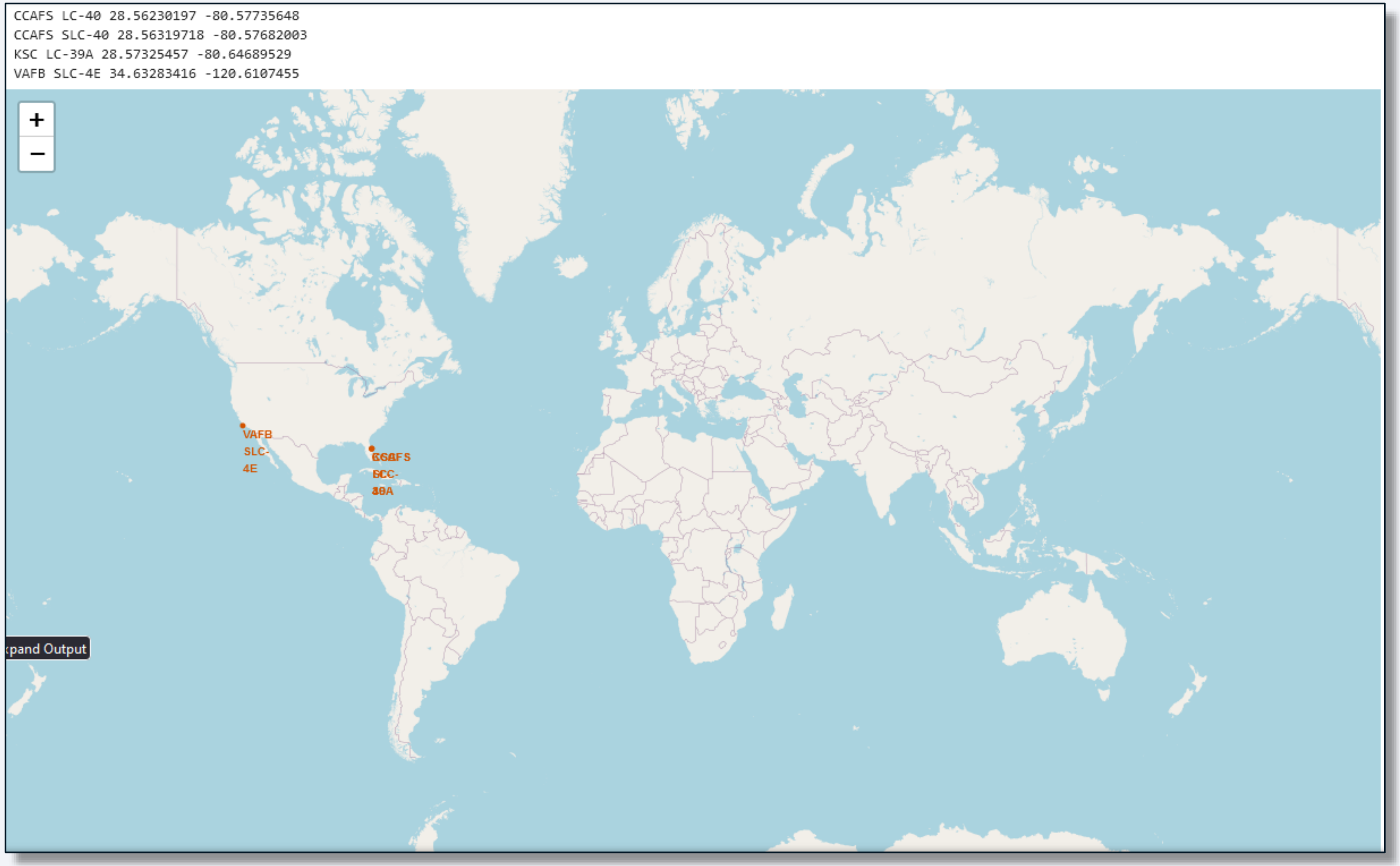
```
 * sqlite:///my_data1.db
Done.
```

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

Section 3

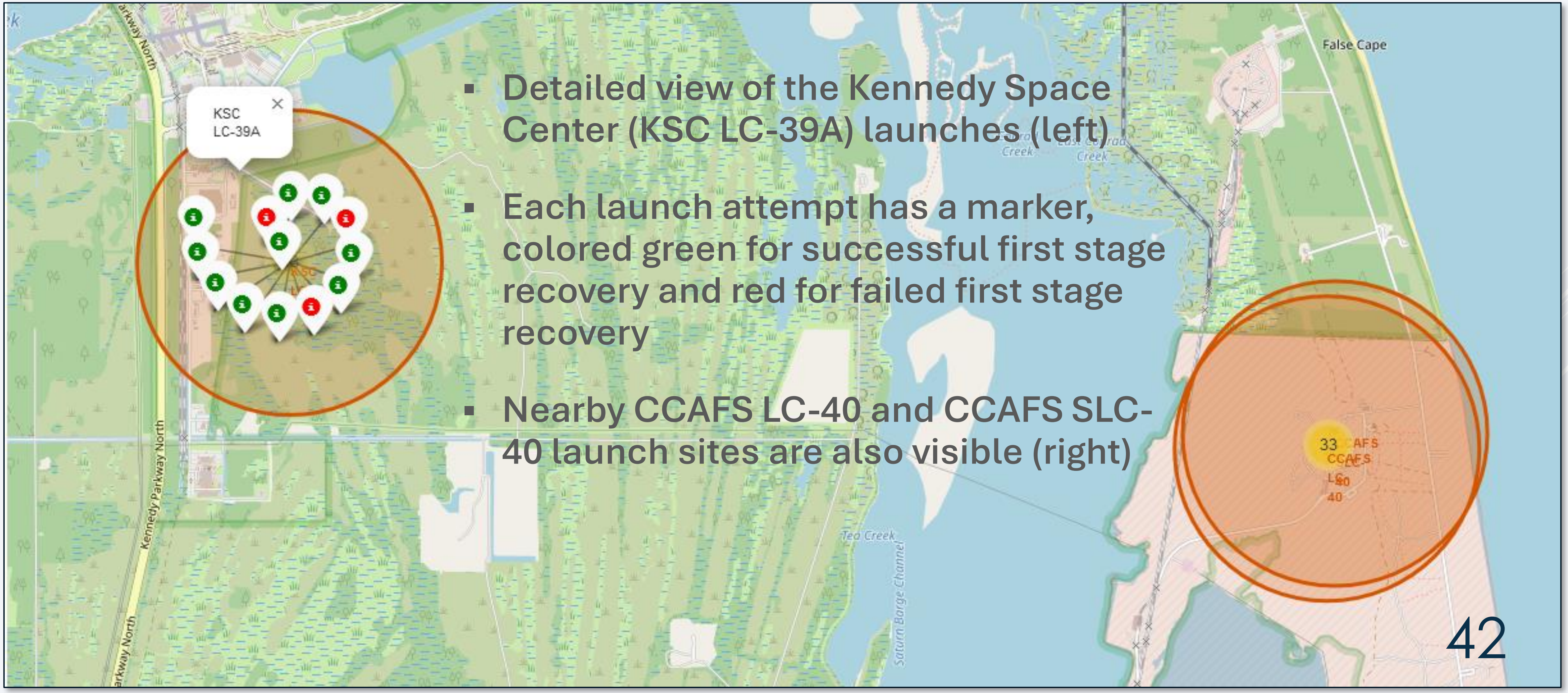
Launch Sites Proximities Analysis

Global Map of SpaceX Launch Sites



- SpaceX uses 4 launch sites indicated here on a global map
- All are in the USA
- All are located near coastlines
- All are relatively near to the equator

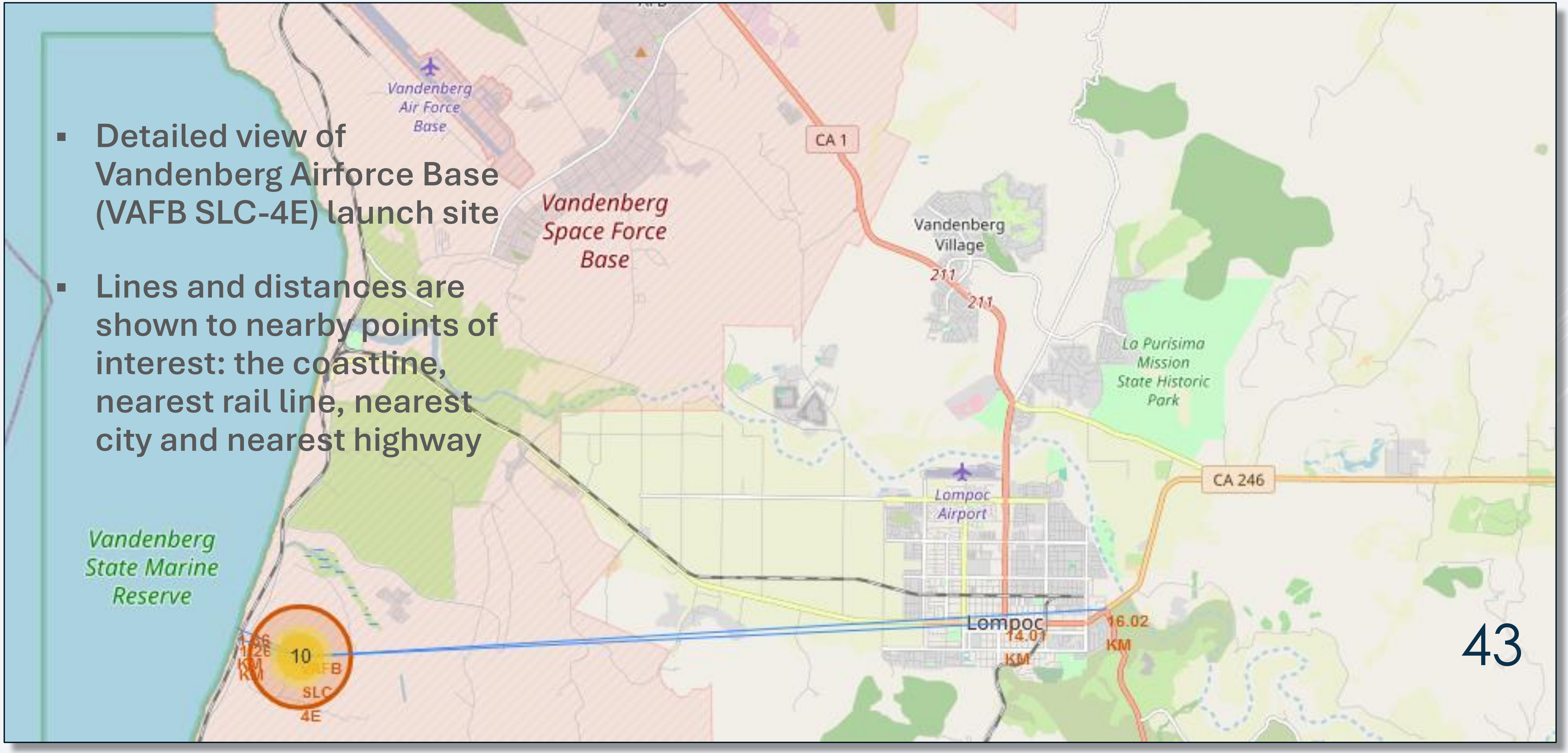
Launch site with color coded launch attempt markers



- Detailed view of the Kennedy Space Center (KSC LC-39A) launches (left)
- Each launch attempt has a marker, colored green for successful first stage recovery and red for failed first stage recovery
- Nearby CCAFS LC-40 and CCAFS SLC-40 launch sites are also visible (right)

Launch site with proximities to selected features

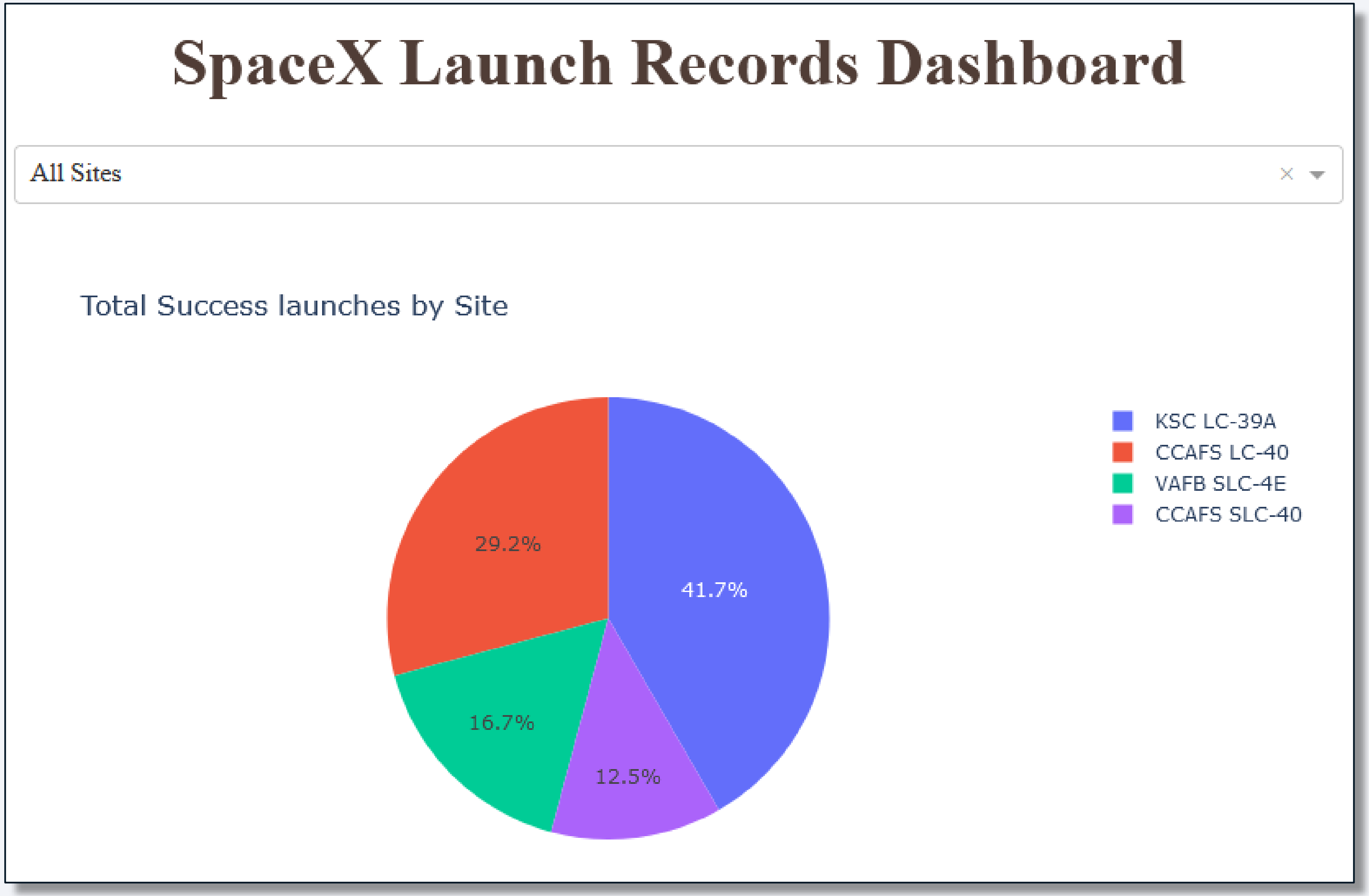
- Detailed view of Vandenberg Airforce Base (VAFB SLC-4E) launch site
- Lines and distances are shown to nearby points of interest: the coastline, nearest rail line, nearest city and nearest highway



Section 4

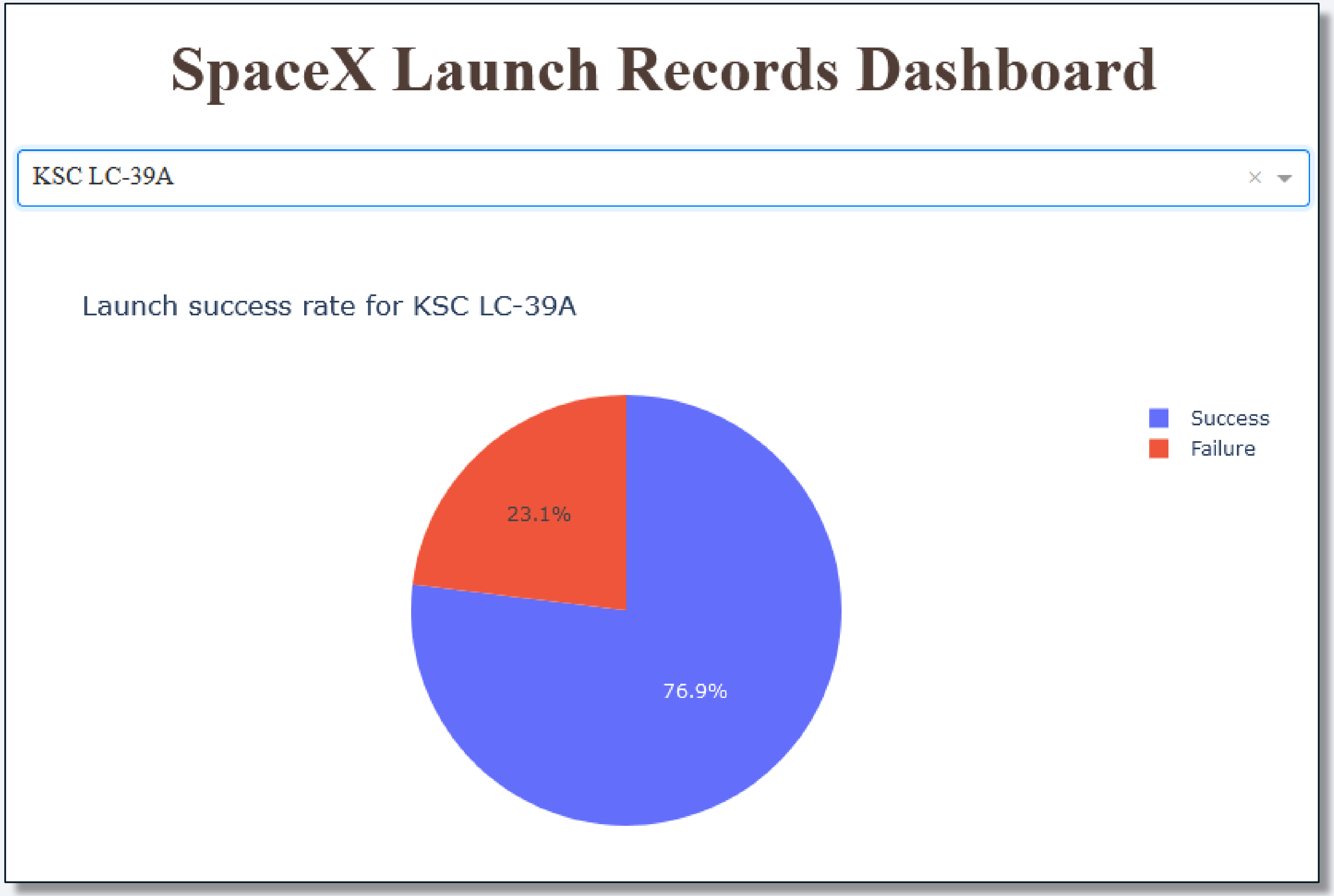
Build a Dashboard with Plotly Dash

Dashboard: Total Successful launches by Site



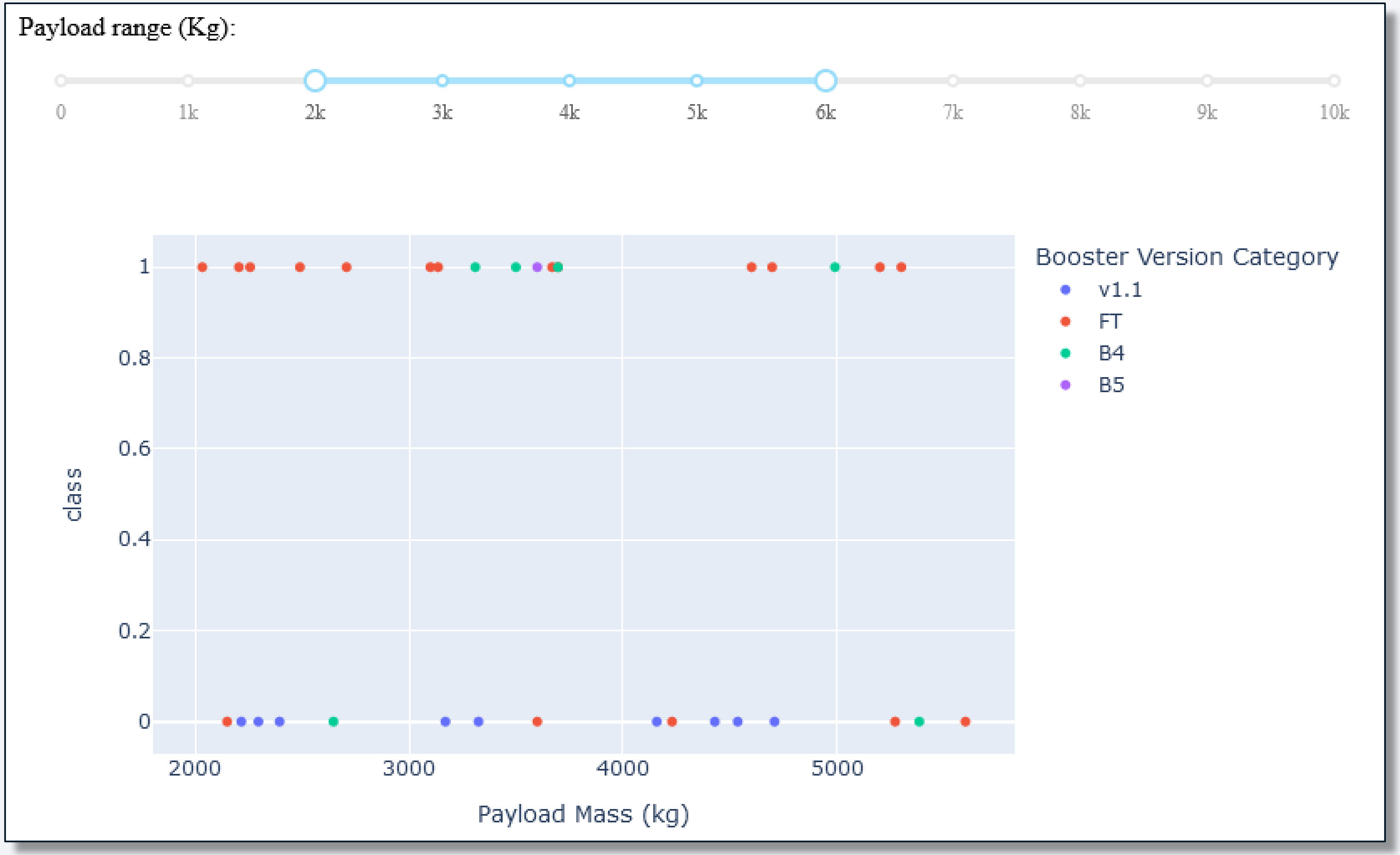
- Dashboard screenshot showing a pie chart of launch percentages by launch site where the first stage was successfully recovered
- Dashboard provides a drop-down to select the launch site (All Sites selected).
- KSC LC-39A has the greatest percentage of launches with successful first stage recoveries

Dashboard: Launch success rate for KSC LC-39A



- Dashboard screenshot showing the site with the highest first stage recovery success rate (KSC LC-39A)
- KSC LC-39A achieved a 76.9% first stage recovery success rate

Dashboard: Success rate per Payload mass by Booster Version

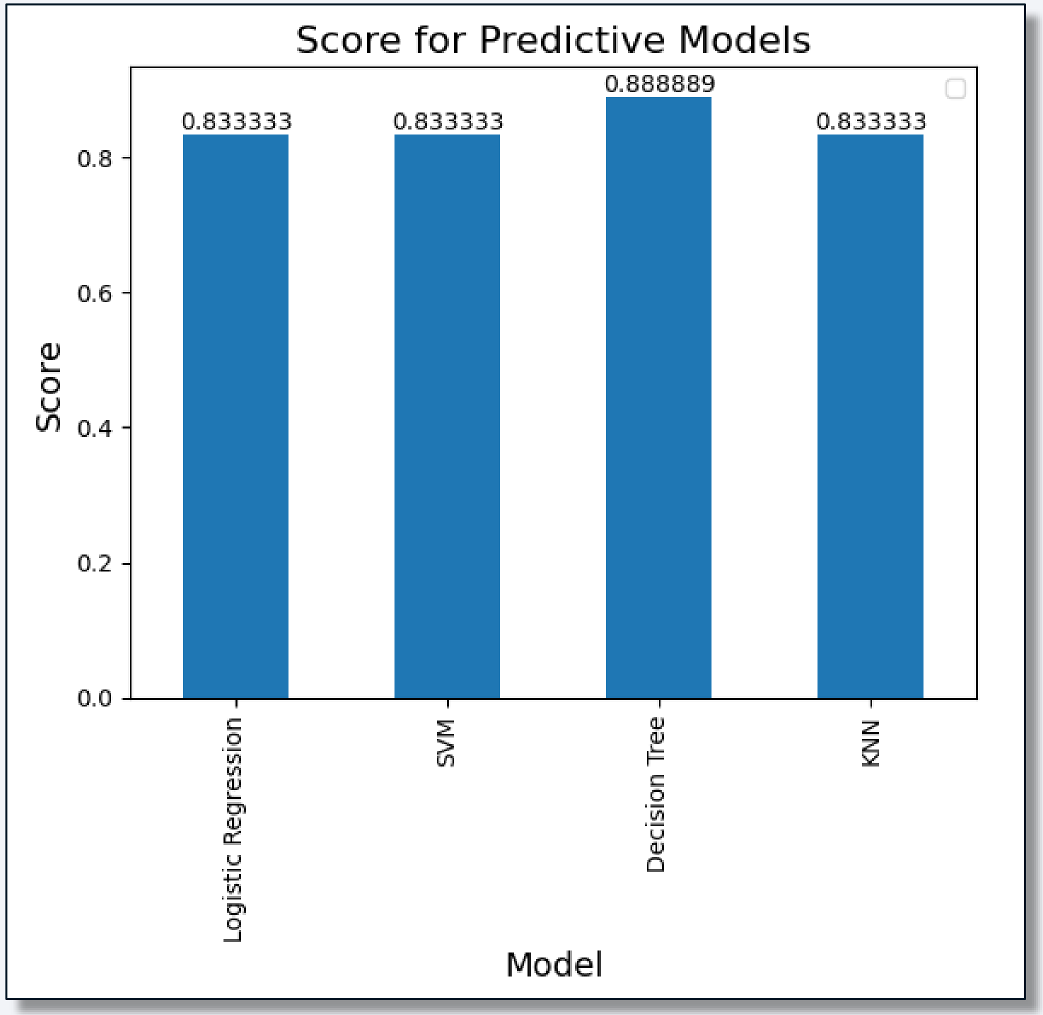


- Screenshot showing success(1)/failure(0) class of first stage recovery for all launch sites
- The range slider has been used to restrict the chart to launches with a payload mass between 2,000 kg and 6,000 kg
- The scatter markers are colored by the Booster Version Category
- The "FT" version of the booster has had the largest percentage of successful recoveries within this payload range

Section 5

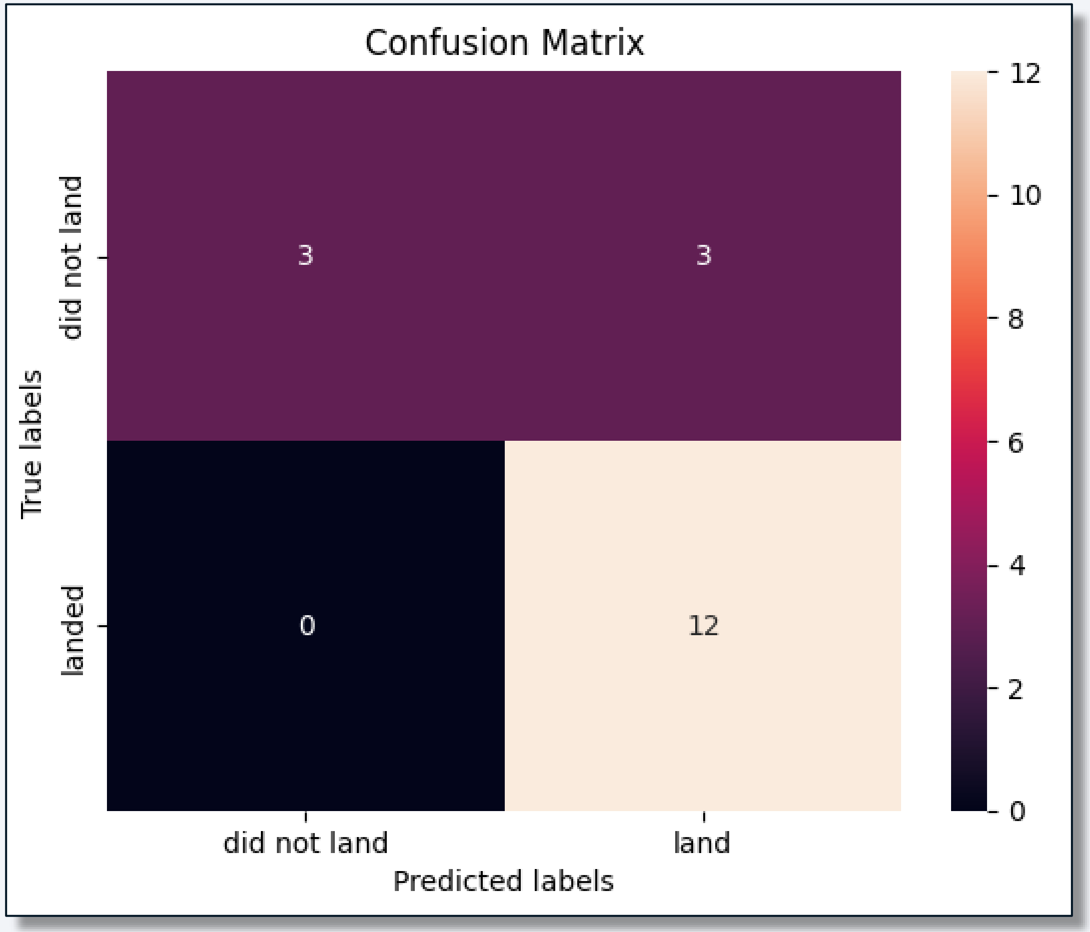
Predictive Analysis (Classification)

Classification Accuracy



- A bar chart showing the score for each model after training based on the 'test' data set is presented
- The Decision Tree was found to have the highest score on the test data

Confusion Matrix



- A confusion matrix for the Decision Tree model is presented
- The model correctly predicted all successful landings
- The model correctly predicted half of the unsuccessful landings

Conclusions

We have shown that:

- Landing sites should be place near coastlines and the equator
- Kennedy Space Center (KSC LC-39A) launch site offers the highest chance of a successful first stage recovery
- The highest chance of a first stage recovery success are for ES-L1, GEO, HEO and SSO orbits
- Success rates increase with the flight number (time/experience)

Appendix

All analyzed data, Python code, SQL data and Jupyter notebooks used throughout this analysis are available in the repository at:

https://github.com/technobok/ibm_data_science_professional_certificate/tree/main/applied_datascience_capstone(10)

Thank you!