IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

David Church
May 12, 2022

Outline

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

Executive Summary

- Data on SpaceX launch results was obtained through use of the SpaceX API.
- Additional data was web scrapped from SpaceX Wikipedia tables to from the full data set of SpaceX launch results.
- The retrieved data was processed and then used to preform exploratory data analysis (EDA) using SQL and various visualization techniques.
- Several classification models where trained and optimized using GridSearchCV allowing for predictive analysis of whether a rocket launch will have a successful landing.
- The predictive model was developed with an accuracy of 83%.

Introduction

- Private space company SpaceX has found tremendous success in reducing cost per rocket launch by landing the first stage of their Falcon9 rockets allowing for re-use of the most expensive portion of the launch.
- Using data science techniques, we aim to analyze past SpaceX launch data to gain insights and build a model to determine whether or not a rocket will land successfully.
- Using these models SpaceY will be able to accelerate the success of our own rocket re-use program making us a viable competitor to SpaceX.

Section 1

Methodology

Methodology

- Data collection methodology:
  - Data was collected via SpaceX API and Data wrangling from Wikipedia tables
- Perform data wrangling:
  - The different cases for the “landing_outcomes” column were parsed into success or failure
  - A new binary column called “Class” was then created with 0 representing a failure and 1 representing a success.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash

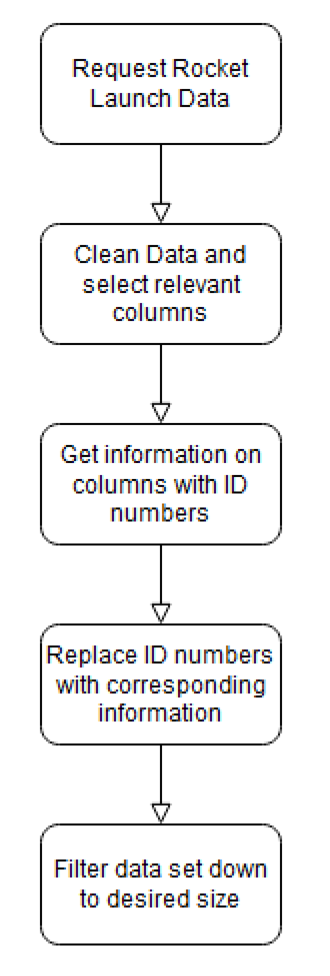
Methodology

- Perform predictive analysis using classification models:
  - The “Class” column was broken out from the data set as the dependent variable
  - The remaining columns were prepared for the analysis using the preprocessing function StandardScalar
  - The Data was broken into 2 groups for training and testing using the train_test_split function
  - Classification models were created using SVM, Classification Tree, and logistic regression with optimized parameters from the GridSearchCV function
  - The accuracy was determined for all classification models for both the training and test dataset

Data Collection

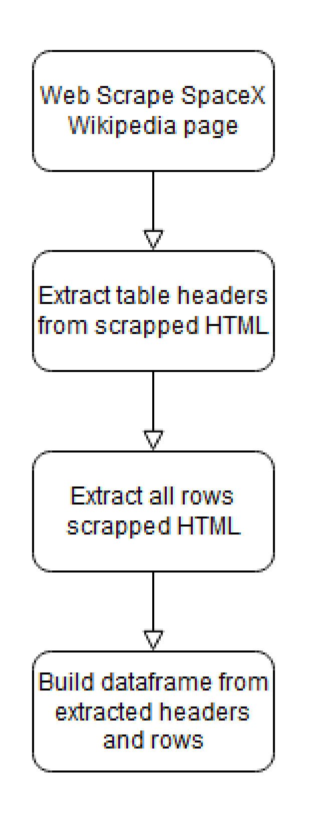
Method 1: API Collection

- Collect basic data from SpaceX API "launches/past" endpoint
- Normalize the data and create a dataframe
- Remove un-needed rows (multi payload launches)
- Access rockets, launchpads, payloads, cores endpoints
- replace ID numbers from basic dataframe with new data
- Clean and filter final dataframe



Method 2: Web Scrapping

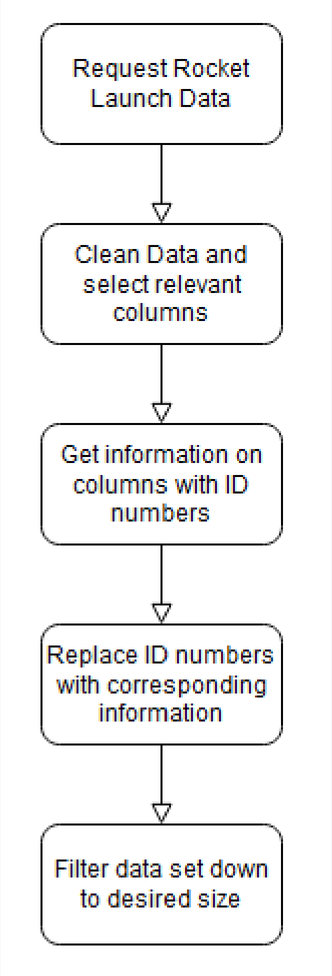
- Table is extracted from Wikipedia page
- Column headers are pulled from the HTML to create a new list
- New lists for each column are created for every row in the table
- The header list and the lists for all columns are combined into a final dataframe



Data Collection – SpaceX API

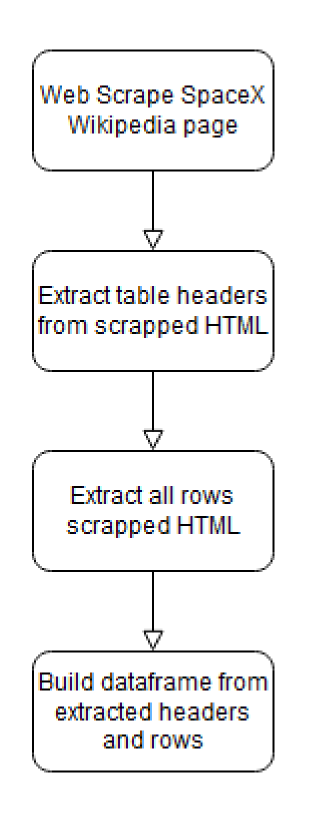
- API Endpoints called:
  - https://api.spacexdata.com/v4/launches/past
  - https://api.spacexdata.com/v4/rockets/
  - https://api.spacexdata.com/v4/launchpads/
  - https://api.spacexdata.com/v4/payloads/
  - https://api.spacexdata.com/v4/cores/
- API Data was consolidated to a single dataframe
- Unwanted rows (i.e. multiple payload and launches with more than 1 booster) were removed from the data set

Link for peer review:
https://github.com/lavaxv123/IBM-Data-Science-Capstone/blob/main/jupyter-labs-spacex-data-collection-api.ipynb



Data Collection - Scraping

- Python requests module was used to get the HTML code for the SpaceX launches Wikipedia page.
- Using the HTML code, a Beautiful Soup object was created.
- Using the “find_all” function, the tables were broken out into a list called html_tables.
- From this new list we select the table to be extracted and assign it to a new variable.
- The headers are then extracted into a new list, followed by extracting each column into its own list.
- These lists are then combined into the final dataframe for the scraped data.



Link for peer review:

https://github.com/lavaxv123/IBM-Data-Science-Capstone/blob/main/jupyter-labs-webscraping.ipynb

Data Wrangling

- The sum of the null values in each column was calculated and showed 5 nulls in the PayloadMass column and 26 null values in the LandingPad column.
- We replace the null values in the PayloadMass column with the average payload mass for the entire dataset leaving the dataframe with only null values for the LandingPad column.
- The unique Outcomes are found and then looped through with a 'for' loop in order to separate the bad outcomes from the good ones.
- A new list called landing_class is created and for every outcome in the dataframe, if the "Outcome" field exists in the list of bad outcomes, a value of 0 will be appended to the new landing_class list. Otherwise a 1 will be appended.
- This new list is then added as a new column to the dataframe called "Class".
- This allows the data to easily show weather one of 8 possible outcomes is a success or failure with a binary column.

Link for peer review:

https://github.com/lavaxv123/IBM-Data-Science-Capstone/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

- To compare various variables and their effects on success rate of the rocket landings various charts were created.
- Five scatterplots were created:
    - Payload Mass vs. Flight Number
    - Launch Site vs Flight Number
    - Launch Site vs Payload Mass
    - Orbit vs Flight Number
    - Orbit vs Payload Mass

- One bar graph was created:
    - Landing Success Rate vs Orbit

Link for peer review:

https://github.com/lavaxv123/IBM-Data-Science-Capstone/blob/main/jupyter-labs-eda-dataviz.ipynb

EDA with SQL

- Task 1: Uses a Select distinct query to get the unique set of launch sites
- Task 2: Query uses select * with a where clause for strings beginning with ‘CCA’
- Task 3: Select query for the sum of ‘Payload_Mass__KG_’ where customer is ‘NASA (CRS)’
- Task 4: Query uses select avg, with a where clause for the desired booster version
- Task 5: Select query on the min(date), with a where clause on the desired landing outcome
- Task 6: Select query with a where clause using an “and” clause to filter on 2 criteria
- Task 7: Select query using ‘group by’ clause to sort on mission_outcome
- Task 8: Use a sub query filter the where clause on payloads that equal the maximum payload
- Task 9: Use a select query with a where clause and 3 criteria to filter on
- Task 10: Select query with a where clause, group by clause, and order by DESC clause

Link for peer review:

https://github.com/lavaxv123/IBM-Data-Science-Capstone/blob/main/jupyter-labs-eda-sql-coursera.ipynb

Build an Interactive Map with Folium

- Each of the four launch sites have several markers indicating them:
    - A large circle surrounding the immediate area of the launch site
    - If the large circle is clicked, on an annotation with the launch site name will appear
    - A cluster marker to group all nearby markers together when the map is sufficiently zoomed out
    - Each cluster marker has a number of colored markers either red or green depending on whether the landing attempt is a success or a failure
    - There are 2 polylines, one from a launch site to the nearest coast, and another from a launch site to the nearest railroad

Link for peer review:

https://github.com/lavaxv123/IBM-Data-Science-Capstone/blob/main/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

- The interactive Plotly dashboard utilizes 2 charts and 2 filters.
- An interactive Pie chart was added that shows the percentage of successes that occur at a given launch site. A dropdown filter for the launch site can be used to get the success rate for a given launch site.
- This Pie chart helps to determine if certain launch site have greater success of landing rockets than others.
- A scatterplot comparing the successes against the payload mass, with the data points being colored according to the booster version. There is a payload mass slider to filter the data in the scatterplot to different payload ranges. The launch site filter also applies to the scatterplot.

Link for peer review:

https://github.com/lavaxv123/IBM-Data-Science-Capstone/blob/main/spacex_dash_app.py

Predictive Analysis (Classification)

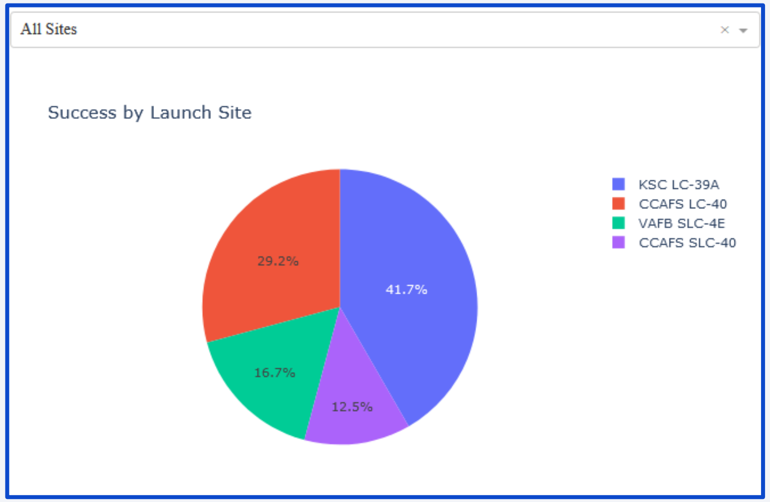
- The data was broken into feature set X and the target variable 'Class' as Y.
- The feature set is standardized using the preprocessing.StandardScaler function.
- The data set is then split into a training set and a test set using train_test_split with a test size of 0.2.
- The GridSearchCV function was used to find the optimal parameters and fit for 4 different classification models (logistic regression, SVC, decision tree, KNN).
- The accuracy was determined for each of the models, using both the training data and the test data.
- A confusion matrix was created using the test data for each of the models.

Link for peer review:

https://github.com/lavaxv123/IBM-Data-Science-Capstone/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results

- Through the use of exploratory data analysis, it was found that as the flight number increases the success rate of landing the rocket does as well.
- Launch site KSC LC-39A has the highest success rate of all launch sites at 76.9%.
- All four classification models that were trained had the same test accuracy at 83.33%, and the best model taking into account the training accuracy is a decision tree with an accuracy of 87.5%.

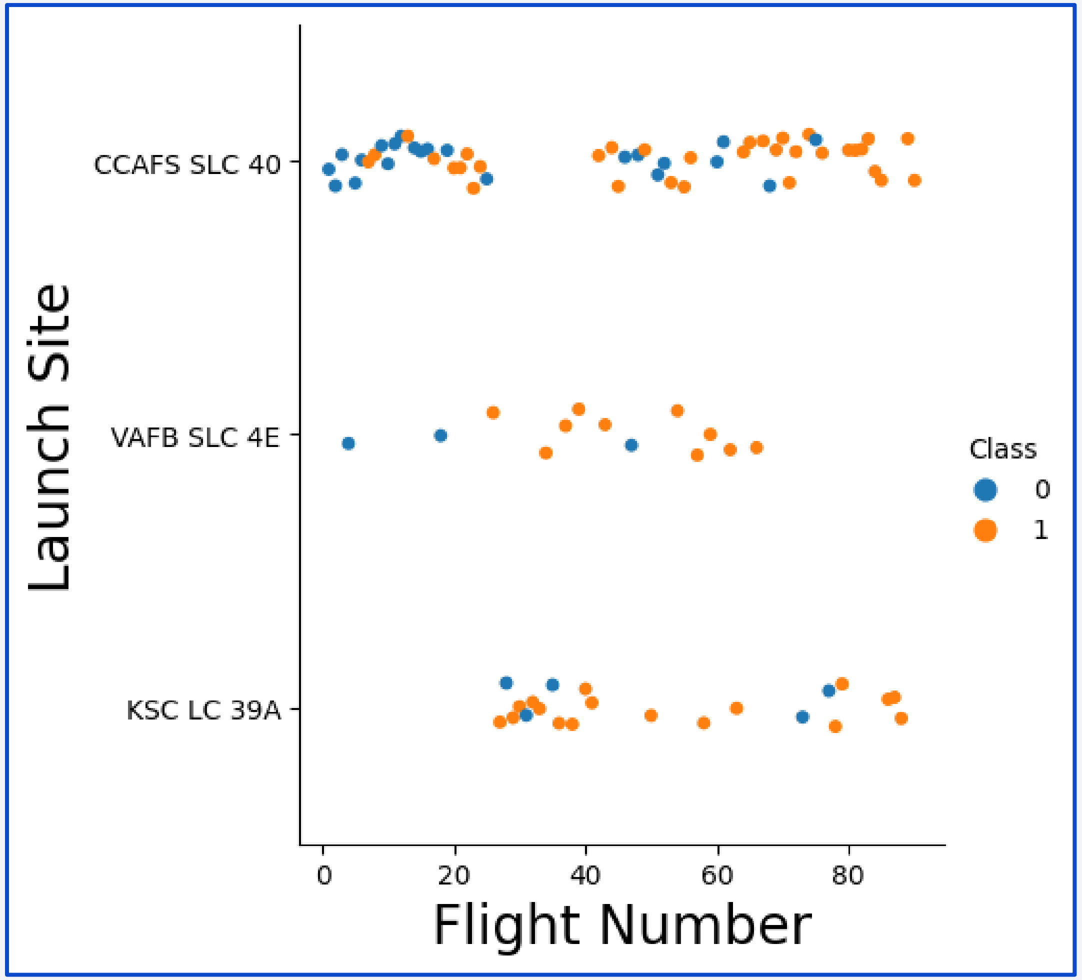


Section 2

Insights drawn from EDA

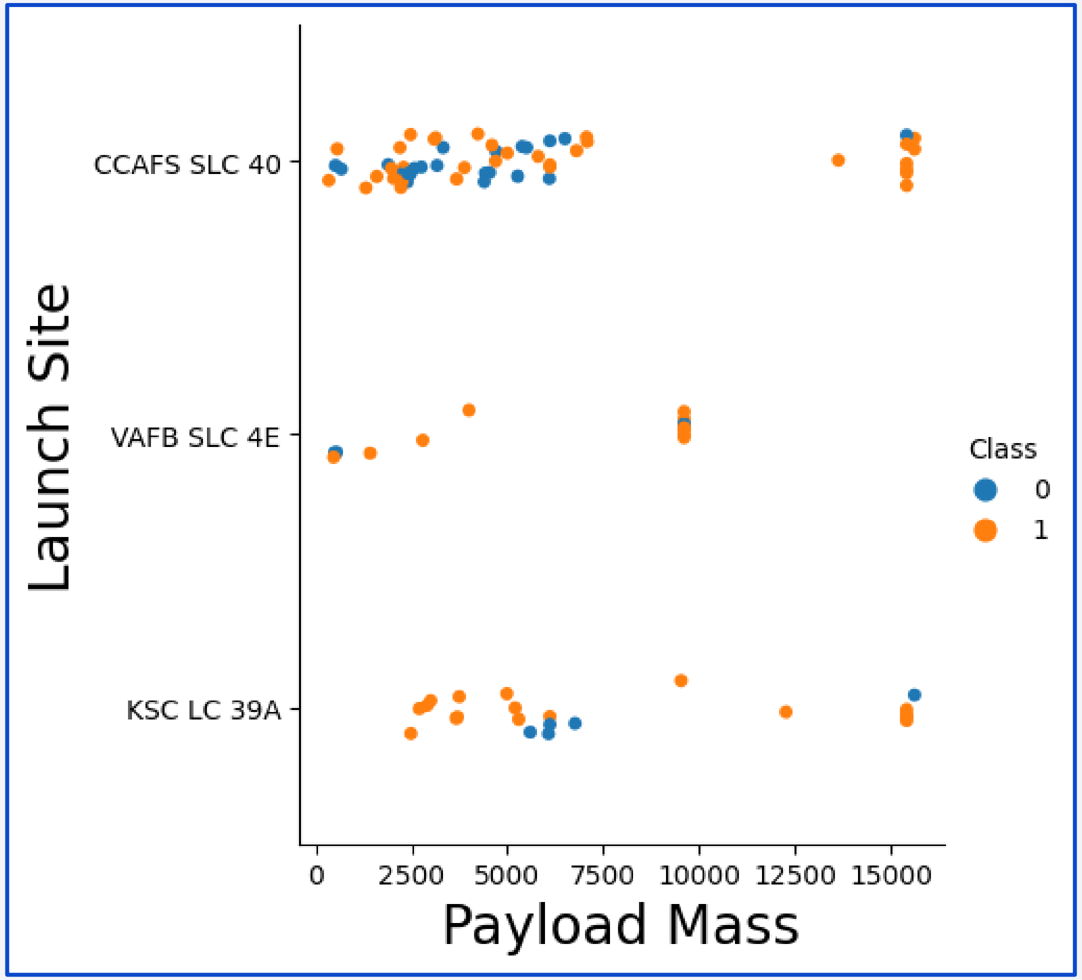
Flight Number vs. Launch Site

- Launch site CCAFS SLC 40 is the most commonly used launch site.
- Launch site KSC LC 39A didn't begin use until after the 20th flight.
- The majority of the earlier flights had landing failures, whereas the majority of the later flights ended with landing success.



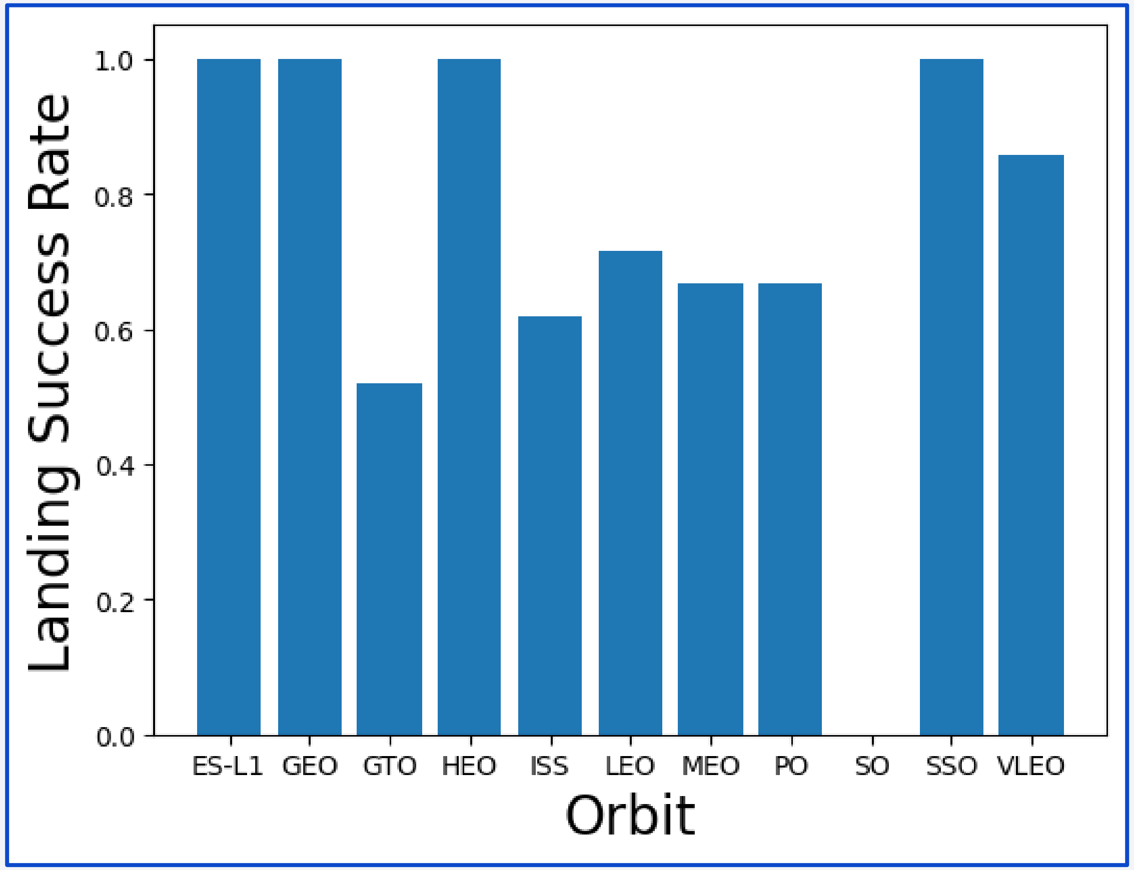
Payload vs. Launch Site

- The CCAFS SLC 40 launch site has a large gap in between its low payload launches and high payload launches.
- Larger payload masses seem to have higher chances for successful landing.
- VAFB SLC 4E doesn't launch rockets with payload mass of greater than 10,000 kg.



Success Rate vs. Orbit Type

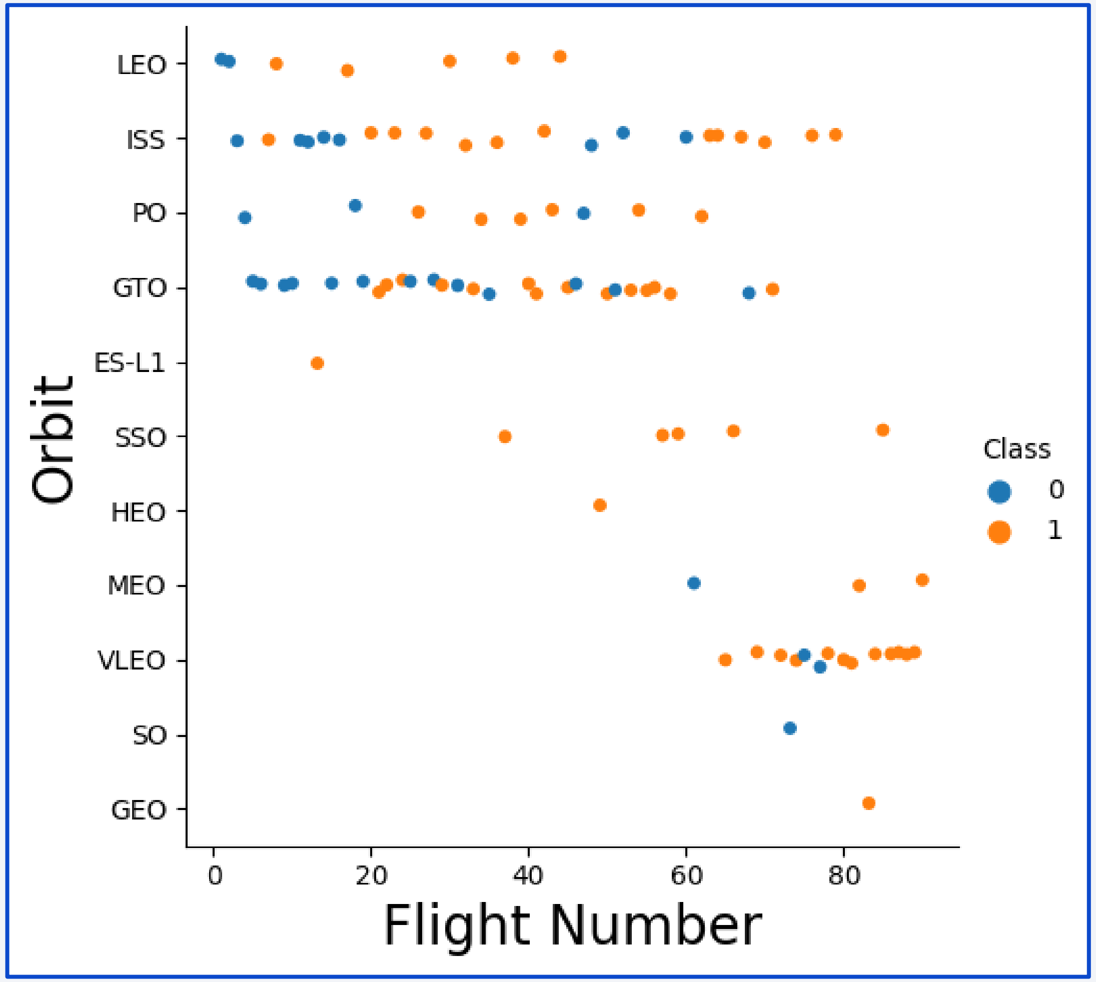
- Orbits have either a high success rate near 100% or a medium level success rate near 60%.
- SO orbit is an outlier with 0% success rate.



Flight Number vs. Orbit Type

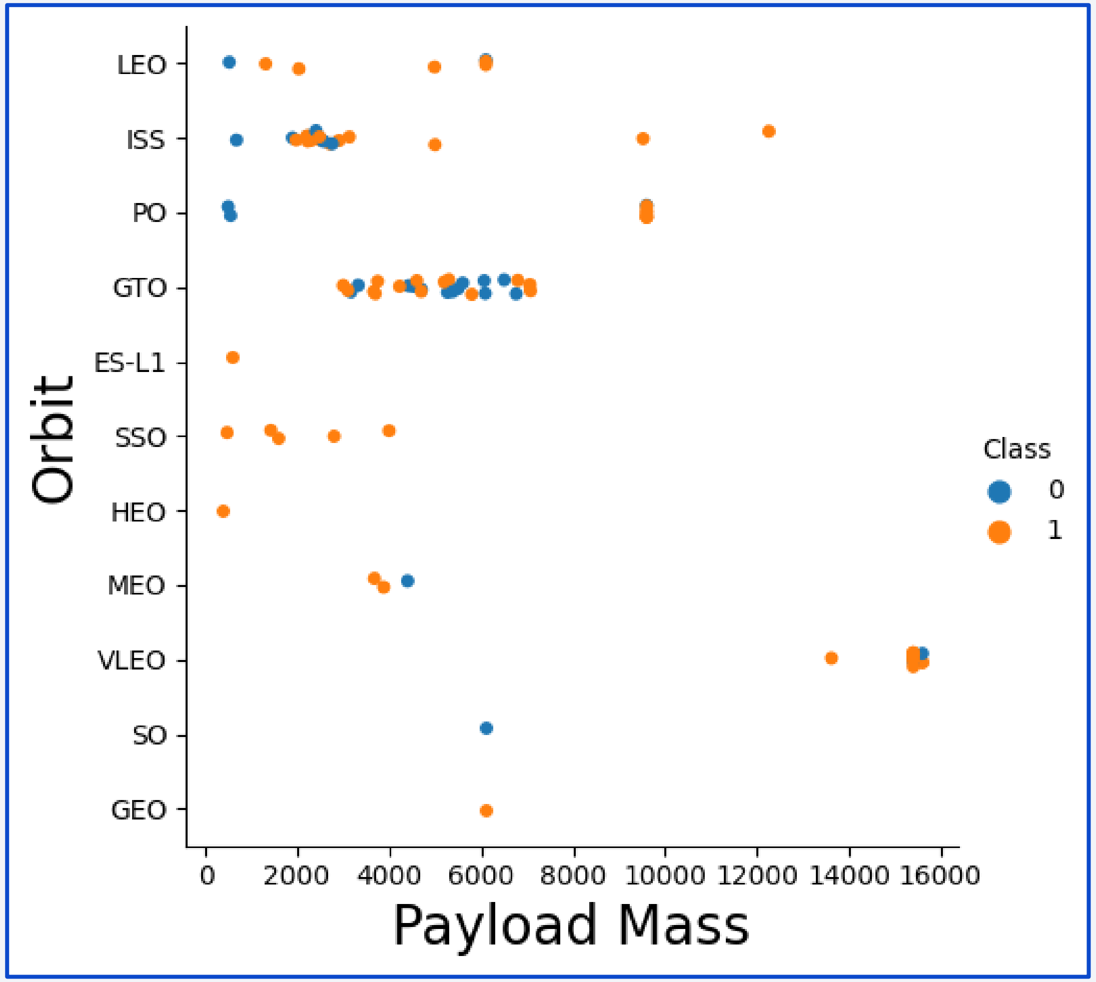
- The most common orbit types are (LEO, ISS, PO, GTO, VLEO), and other orbit types don't have a significant enough number of launches to determine success rate.

- Launches to VLEO only occurred after 60 flights meaning success rate for this orbit type may be skewed towards success.



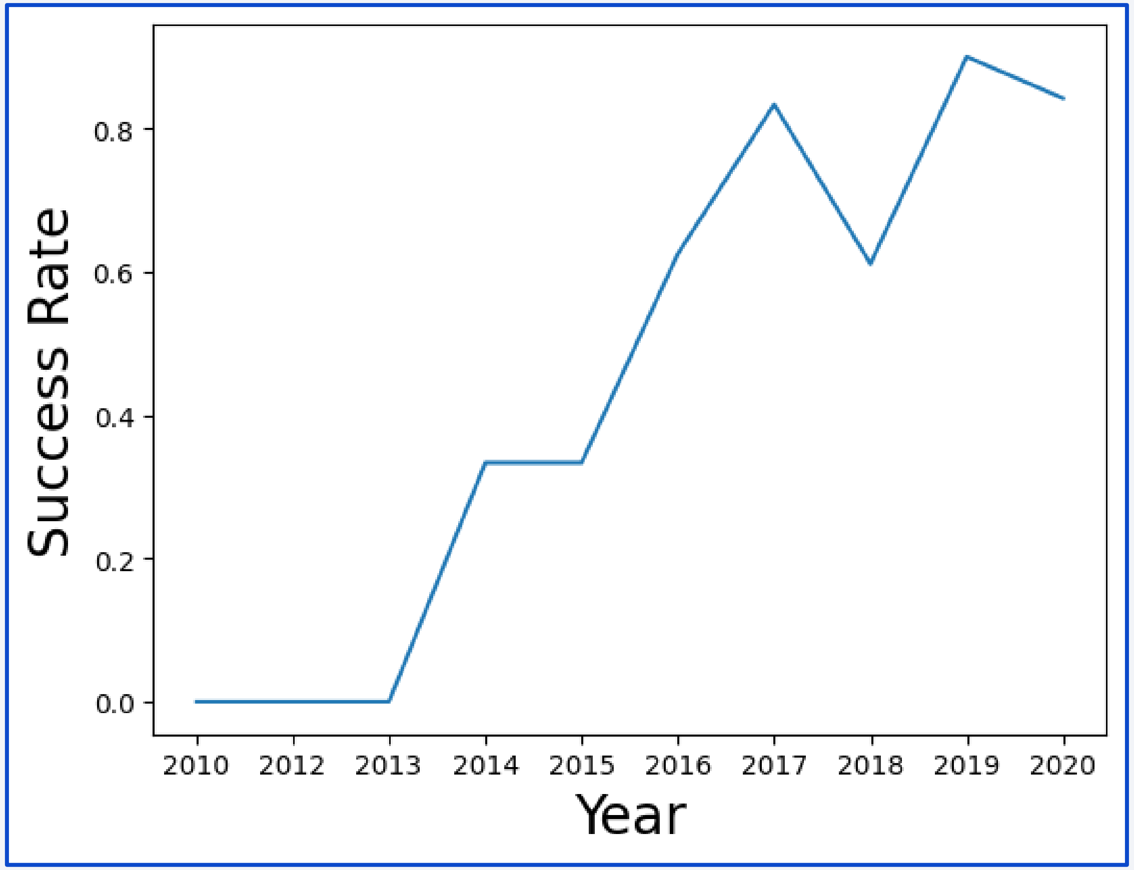
Payload vs. Orbit Type

- Typical rocket launches carry payloads with under 8,000 kg mass.
- VLEO is an outlier with all of the payloads sitting around 16,000 kg mass



Launch Success Yearly Trend

- It took several years before landing a rocket became successful. Years 2010 through 2013 had a 0% success rate.
- The general trend of the launch success rate is positive. This seems to agree with the earlier observation that success rate increases with flight number.



All Launch Site Names

- The table to pull the unique launch site values was pulled from the MySQL database with the following query:
  - select distinct Launch_site from spacex

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

Launch Site Names Begin with 'CCA'

- The following query uses the wildcard operator ‘%’ to pull records where launch site name begins with ‘CCA’:
  - select * from spacex where Launch_Site like 'CCA%' limit 5;

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

- Using the aggregate function SUM to sum all of the payload masses and a where function to filter the following query produces a table with the total sum of payload mass 45,596 kg:
  - select SUM(Payload_Mass__KG_) from spacex where Customer ='NASA (CRS)';

Average Payload Mass by F9 v1.1

- Using the average function AVG to average all of the payload masses and a where function to filter the following query produces a table with the average payload mass 2,928 kg:
  - select avg(Payload_Mass__KG_) from spacex where booster_version = 'F9 v1.1';

First Successful Ground Landing Date

- The Min function was used to get the earliest date alongside the where function to filter on the desired landing outcome to find 2015-12-22 as the earliest successful ground pad landing.
    - select min(date) from spacex where Landing_Outcome = 'Success (ground pad)';

Successful Drone Ship Landing with Payload between 4000 and 6000

- A where function was used to filter the query down, using the ‘and’ statement to have 2 criteria to filter on.
  - select booster_version from spacex where landing_outcome = 'Success (drone ship)' and (4000 < payload_mass__kg_ < 6000);

booster_version
F9 FT B1021.1
F9 FT B1022
F9 FT B1023.1
F9 FT B1026
F9 FT B1029.1
F9 FT B1021.2
F9 FT B1029.2
F9 FT B1036.1
F9 FT B1038.1
F9 B4 B1041.1
F9 FT B1031.2
F9 B4 B1042.1
F9 B4 B1045.1
F9 B5 B1046.1

Total Number of Successful and Failure Mission Outcomes

- This query uses the count function to return the number of rows, using ‘group by’ to group the distinct mission_outcome’s together.
  - select mission_outcome, count(*) as NUM from spacex group by mission_outcome;

mission_outcome	NUM
Success	98
Failure (in flight)	1
Success (payload status unclear)	1
Success	1

Boosters Carried Maximum Payload

- The distinct function is used to get only the unique boosters and then the query is filtered with a where clause. In order to get maximum payload mass for the where clause, a sub query using the max function is used.
  - select distinct(Booster_version) from spacex where payload_mass__kg_ = (select max(payload_mass__kg_) from spacex);

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

- This query uses a where clause with double sided wild cards for 2 of the criteria. The third criteria uses the ‘year’ function to pull the year from the date column.
  - select landing_outcome, booster_version, launch_site from spacex where year(date)=2015 and mission_outcome like '%Failure%' and landing_outcome like '%drone%';

landing_outcome	booster_version	launch_site
Precluded (drone ship)	F9 v1.1 B1018	CCAFS LC-40

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

- This query uses the count function to count all the records and a where clause is used to specify the date range. The records are then grouped by the landing outcome and ordered by the total counts in descending order.
  - select landing_outcome, count(*) as NUM from spacex where '2010-06-04'< Date < '2017-03-20' group by landing_outcome order by NUM DESC;

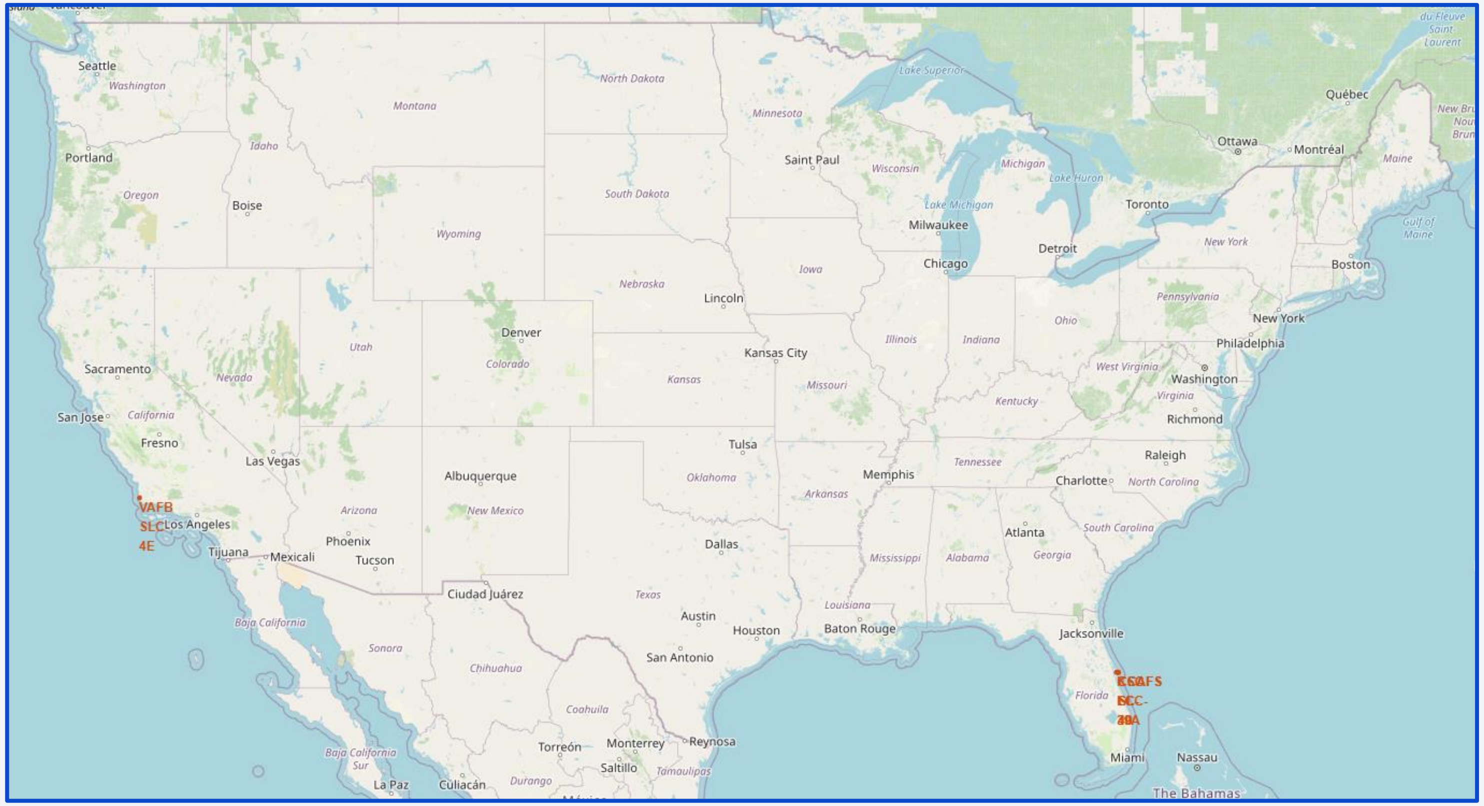
landing_outcome	NUM
Success	38
No attempt	21
Success (drone ship)	14
Success (ground pad)	9
Controlled (ocean)	5
Failure (drone ship)	5
Failure	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1
No attempt	1

Section 3

Launch Sites Proximities Analysis

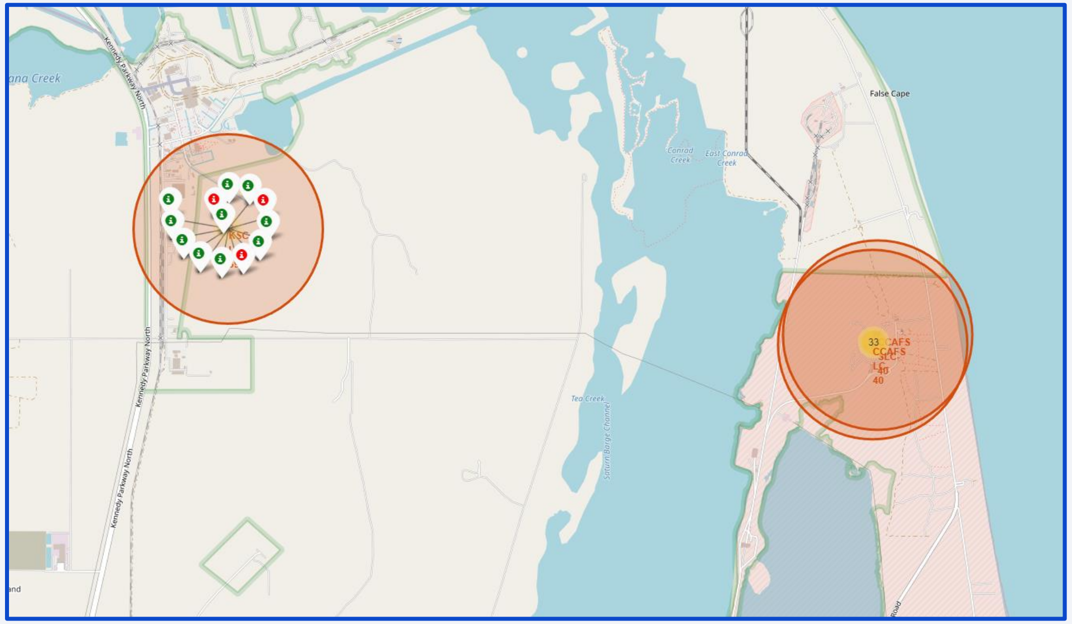
Launch Site Locations

- All launch site locations are shown on this map of the United States.
- Launch site's are only present in California and Florida.
- All launch sites are by the coast.



Launch Outcome Markers

- The 3 launch sites in Florida are show on the map to the right.
- In the interactive map, there are colored markers to help visualize the launch outcomes at each launch site.
- As you click on the launch site the markers will appear, and as you click off the disappear.



Folium's Polylines

- Launch site KSC LC-39A has a polyline going from the launch site to the shore.

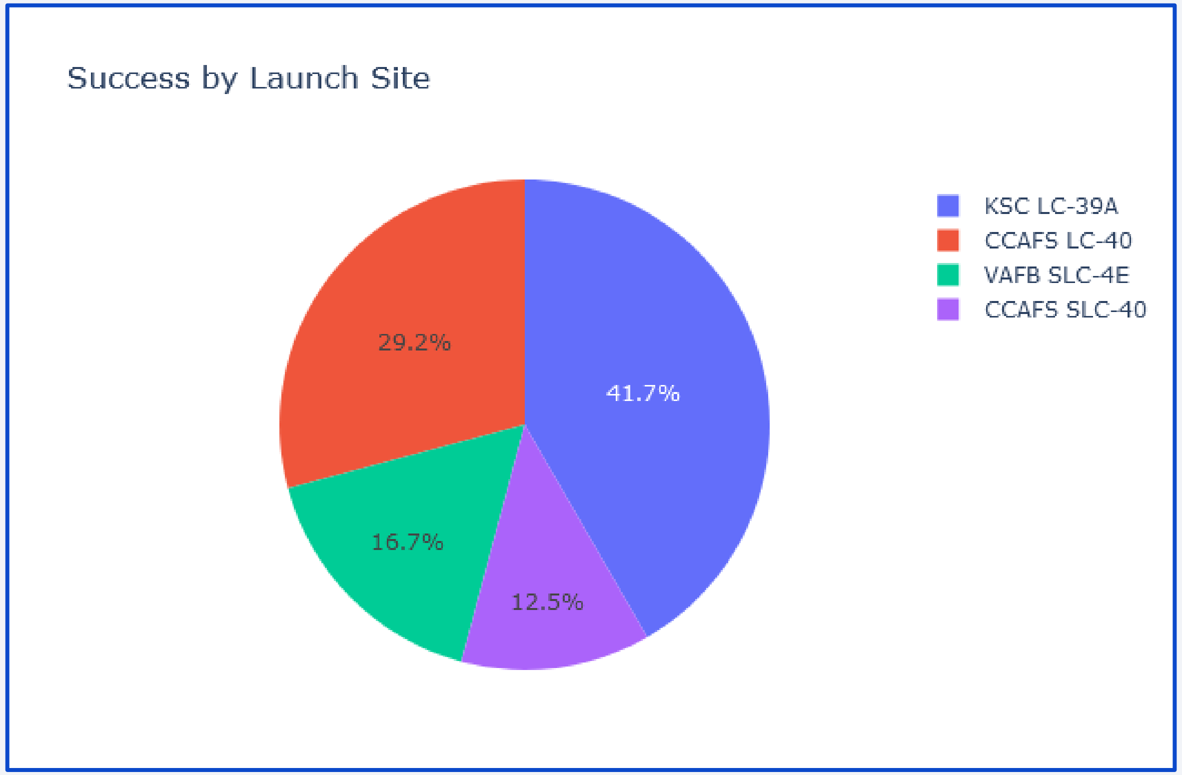
- Launch site CCAFS SLC-40 has a polyline going from the launch site to the nearby railroad.

Section 4

Build a Dashboard with Plotly Dash

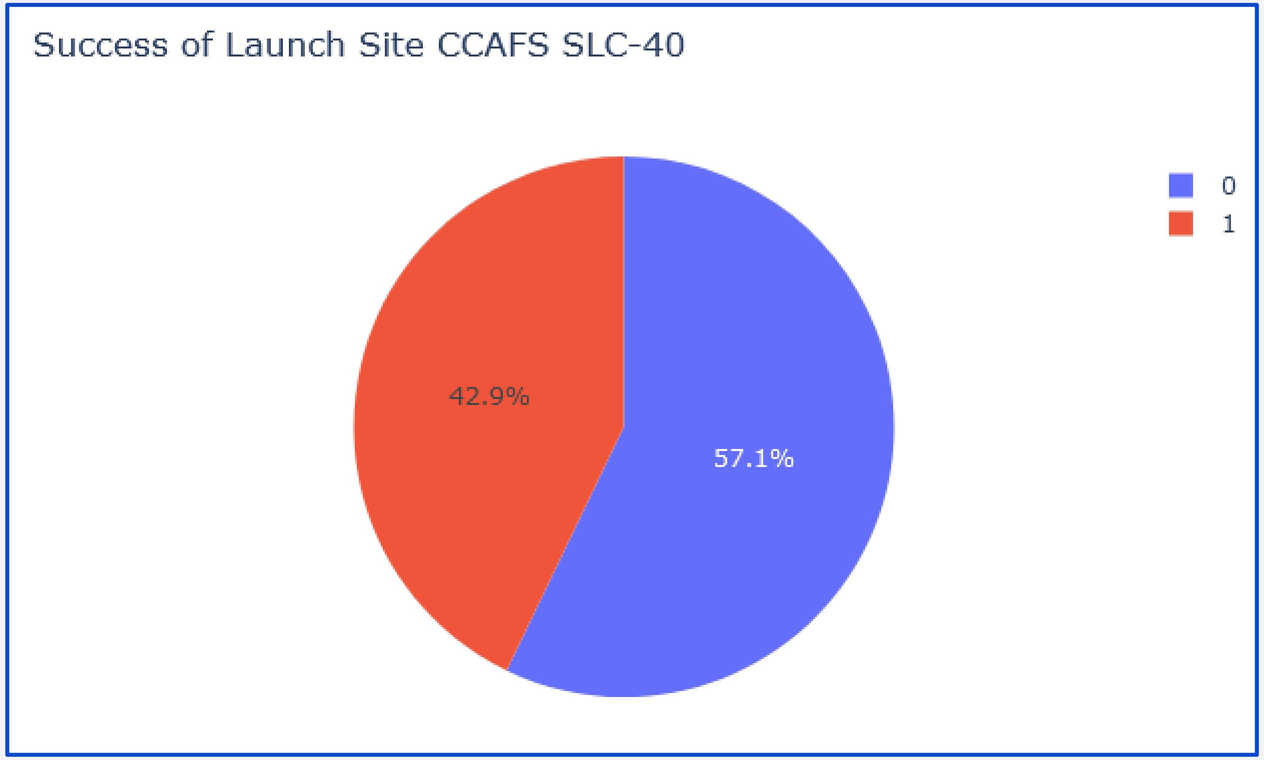
Launch Site Success Rates

- Out of all successful landing the vast majority are split between KSC LC-39A and CCAFS LC-40.



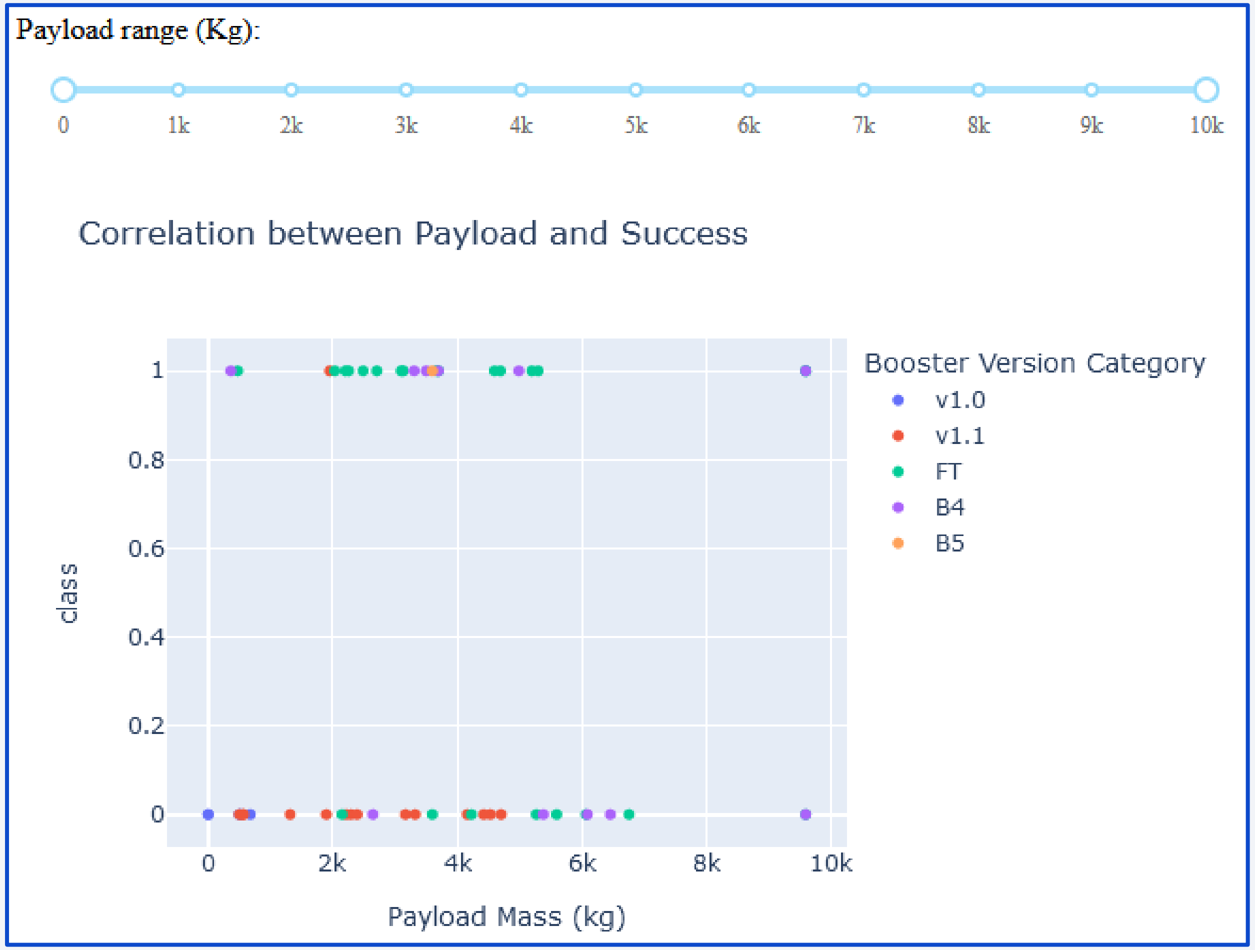
Highest Launch Site Landing Percent

- The launch site had the highest landing success rate.
- The success rate of this launch site is 42.9%.



Success by Payload Mass

- The slider on the top can filter the visual based on the payload mass.
- The visual shows the successes for different payload mass, color coded based on the booster version.



Section 5

Predictive Analysis (Classification)

Classification Accuracy

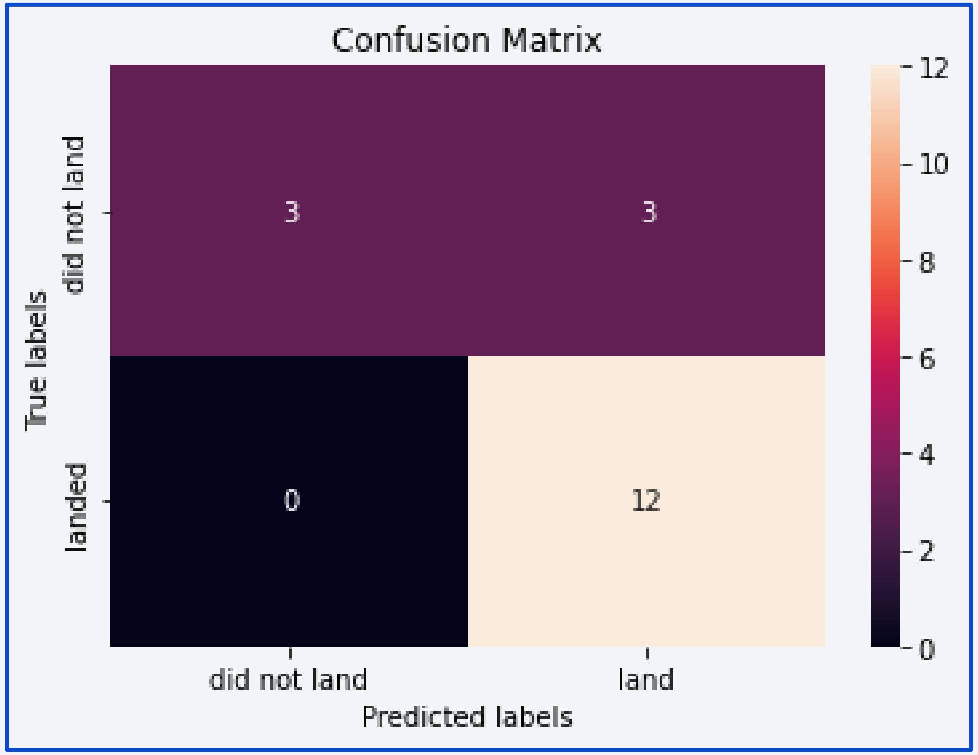
- Each model had the same test accuracy at 83.3 %.

- The Decision Tree Model had the highest training accuracy at 87.5%

Model	Data	Accuracy
Logistic Regression	Training	84.6%
Logistic Regression	Test	83.3%
SVM	Training	84.8%
SVM	Test	83.3%
Decision Tree	Training	87.5%
Decision Tree	Test	83.3%
K Nearest Neighbor	Training	84.8%
K Nearest Neighbor	Test	83.3%

Confusion Matrix

- This model had a problem with false positives.
- Out of all the cases in the test set, there where 3 false positives and the rest of the results were true positives.



Conclusions

- Every since 2013 SpaceX has had a positive trend for launch success rate peaking in 2019 at 90%.
- The majority of the launches occur with a payload mass of under 10,000 kg and an average launch mass of 2,928 kg.
- Launch site KSC LC-39A has the highest number of successful landings, however the site with the highest success rate is CCAFS SLC-40 with 42.9%.
- With the optimized parameters that were tested, all 4 of the models that were trained had the same test accuracy at 83.3%. The Decision Tree model had the best training accuracy at 87.5%.
- All models had an bias with false positives and no false negatives.

Appendix

- GitHub repository link:
  - https://github.com/lavaxv123/IBM-Data-Science-Capstone

Thank you!