IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Sebastián Vilchis
September 18, 2025

Outline

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

Executive Summary

- This project aimed **to predict the successful landing of SpaceX rockets**, the hallmark of SpaceX is to reuse the first stage, presumably the main part of rocket launching, to cut the flight cost by 50%. For that purpose we prepare data about the technical specificities of every launch, analyze it and find the best model for it to make our predictions.
- The data collected is ranging from the date, target orbit to the booster version, grid fins, the launching pad and the rocket leg usage to attain 17 fields to aid us answer our questions about the launch. One of the key results in the modeling is the most important feature in the modeling is the landing leg usage for a successful launch.

Introduction

- SpaceX, is leading the market of private space missions, given that many companies are trying to replicate its success and for SpaceY to drive the best strategy possible, it is fundamental that we answer **the main question of this project, can we predict the landing of the first stage?** reusing it accounts for the economic advantage SpaceX has over its competitors. Other commercial companies offer their flights for 143 million USD, while SpaceX reusable rocket mission is offered for 64 million USD.
- Whereas the data collected can be used to answer questions related other technical important details about missions, such as the orbit selection for a given mission, we must focus on questions about the landing itself, **which are the features that predict the given outcome of the first stage landing? what kind of data we need to answer this? Is the landing location a good predictor?** Is the problem with SpaceX purely technical? From a fundamental point of view it is and we can circumvent it by analyzing the technical aspects that vary from one launch to the next such as payload mass and booster version.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - The dataset was obtained using different entry points in the SpaceX Rest API and performing WebScraping on the dedicated Wikipedia page.
- Perform data wrangling
  - The processes on data achieve a useful and consistent format, removing missing data and assessing the distribution of its different categories as well as transforming the data about the landing outcome, encoding it in binary system, 1 for successful, 0 for failure.
- Perform exploratory data analysis (EDA) using visualization and SQL
  - The Exploratory Data Analysis focused on analyzing the different payloads and booster versions on the Falcon 9, the current and main rocket, as analyzing the landing sites
- Perform interactive visual analytics using Folium and Plotly Dash.
  - Various important graphs to asses the selected features and their relationship, as well as some displayed in an interactive web based dashboard, relating the outcome to different quantities.
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models to predict the outcome.

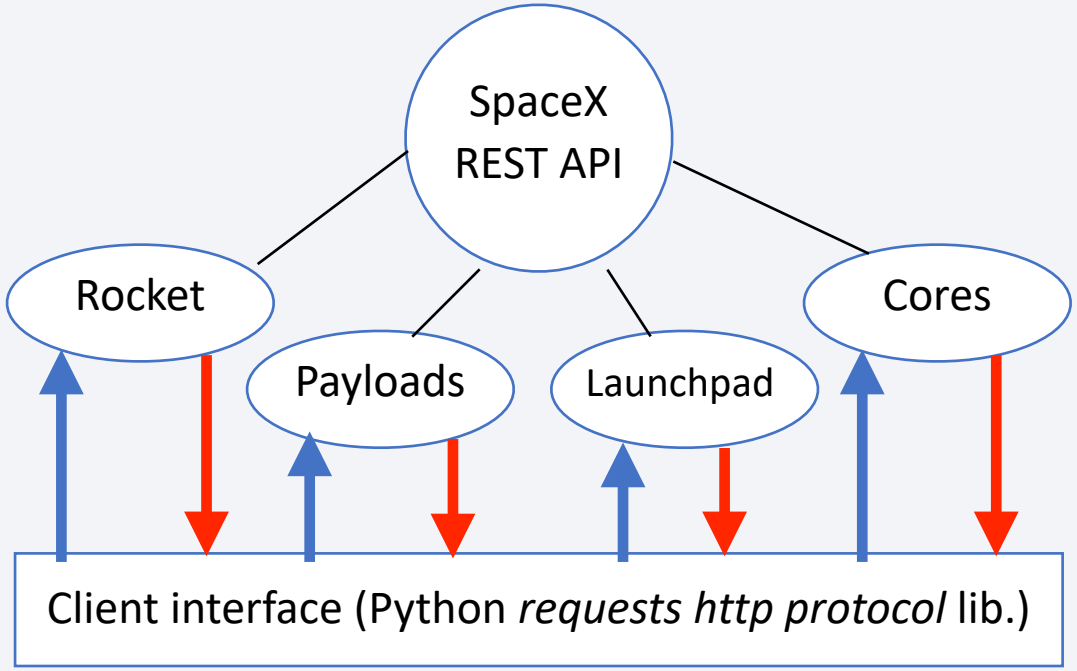
Data Collection

- Datasets were drawn from two main sources: SpaceX API, which consist in the following entrypoints: *rocket, payloads, launchpad* and *cores.*
  - From rocket we would like to learn the booster name.
  - From payload we would like to learn the mass of the payload and the orbit that it is going to.
  - From launchpad we would like to know the name of the launch site being used, the longitude, and the latitude.
  - 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, whether the core is reused, whether 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.
- The general picture is that a *Client* (user) HTTP requests (via python in this case) information in an specific entry of an API, the response is codified in a JSON file which we then filtered and concatenated in a pandas DataFrame to be usable.
- Some data was corroborated and added from the dedicated Wikipedia page, using a WebScraping python library called Beautiful Soup, **the following are the fields selected**:
  - FlightNumber, Date, BoosterVersion, PayloadMass, Orbit, LaunchSite, Outcome, Flights, GridFins, Reused, Legs, LandingPad, Block, ReusedCount, Serial, Longitude, Latitude

Data Collection – SpaceX API

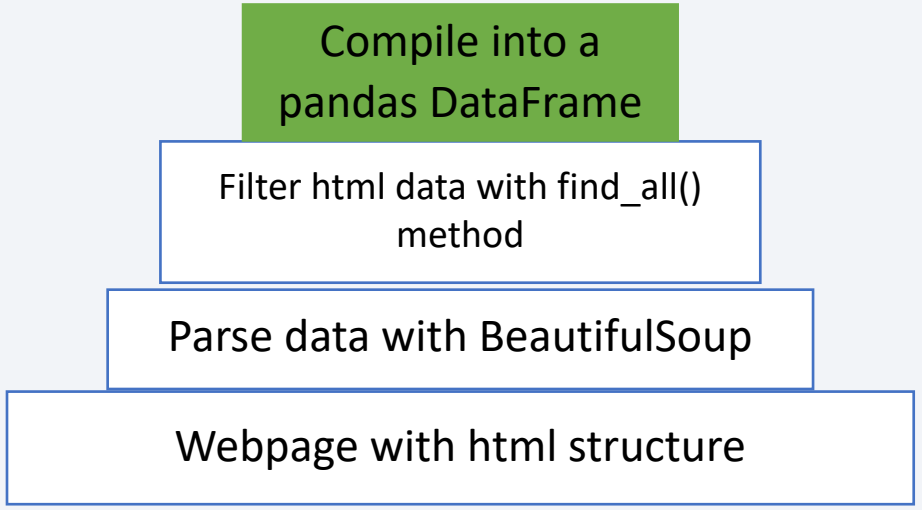
- The flowchart on the left, represent requests with blue arrows and responses with red arrows

- https://github.com/rsevp/IBM/blob/main/jupyter-labs-spacex-data-collection-api.ipynb



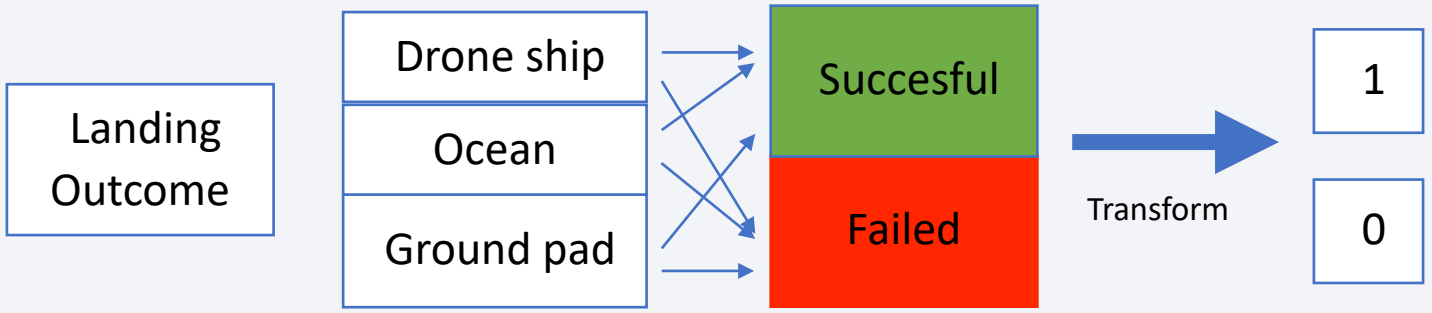
Data Collection - Scraping

- Using the Beautiful Soup library you can identify the components of an html webpage. In this case, we used the table tag of the html structure of Wikipedia.org, then implemented functions to extract the structured data from the desired table.
- https://github.com/rsevp/IBM/blob/main/jupyter-labs-webscraping.ipynb



Data Wrangling

- Once the data is available in a pandas dataframe, we learn the datatypes with the df.dtypes attribute. We asses the missing values with df.isnull() method. We check the count of the variables Orbit, LaunchSite and **LandingOutcome (our target variable)**, the last is transformed from a categorical nature to a binary system, as it is more useful for modeling to encode it in a numerical format (as it is illustrated by the diagram) that we call **Class**.
- https://github.com/rsevp/IBM/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb



EDA with Data Visualization

1. First, the f**light number and payload mass scatterplot**, each point gets colored with its landing outcome (*Class*). This was done to analyze historically the mass payload successful landings, overall the success rate has increased over the years.
2. Then, **flight number and landing site scatterplot,** to look for a trend on the top successful landing sites.
3. The scatterplot **between Payload Mass and Launch Site,** to analyze the usage of the different landing sites, it turns out heavy payloads only have 2 landing sites
4. We used groupby method on Orbit column and get the mean of Class column, then plotted a **bar chart to visualize the success rate on each target orbit.**
5. After, we used a **scatterplot for FlightNumber and Orbit type**, to analyze the different orbits reached in different periods of SpaceX,
6. The last **Orbit scatterplot was made with the Payload Mass** variable, in that plot we saw only VLEO (very low earth orbit) orbit was targeted by heavy flights.
7. Finally, we **plotted a line with the rise in the success rate over the years**, which is over 80% since 2019.

- https://github.com/rsevp/IBM/blob/main/edadataviz.ipynb

EDA with SQL

The following queries in SQL with the sqlite3 library were performed due to the connection of the appropriate database:

- Display the names of the unique launch sites in the space mission
- Display 5 records where launch sites begin with the string 'CCA'
- Display the total payload mass carried by boosters launched by NASA (CRS)
- Display average payload mass carried by booster version F9 v1.1
- List the date when the first succesful landing outcome in ground pad was acheived.
- List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
- List the total number of successful and failure mission outcomes
- List all the booster_versions that have carried the maximum payload mass, using a subquery with a suitable aggregate function.
- List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.
- 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.

- https://github.com/rsevp/IBM/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- Circles are labeled locations in a map, while Markers present messages upon clicking a map site, we created the following:
- Circle and Marker on the NASA Johnson Space Center.
- Circle and marker to every unique launch site (4), labels were provided by the previous pandas data frame.
- Green marker in every success launch, Red marker on every unsuccessful launch within Market Cluster object which facilitates in-map counting.
- Using the polyline() object, we created lines from the CCAFS SLC-40 site to the nearest city, coastline, highway and railroad, then calculated their length.
- A line to the equator was added.
- Every object was added to answer specific questions about the success rate and their spatial relationship. Also to ask ourselves about the selection of the launching sites.
- https://github.com/rsevp/IBM/blob/main/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

- The Dashboard contains two main categories: All the bases or Each specific base, for all the bases we present each base **share of successful landings in a pie plot**. While selecting a specific base presents a pie p[lot of the successful vs. the unsuccessful landings.
- Then for each category we present **a scatterplot with the selected base(s) and the payload range successfu**l and unsuccessful landings, as well as the booster version used in the color of each point.
- This was the best way to present key findings in the variables like the successful landing sites.
- https://github.com/rsevp/IBM/blob/main/spacex-dash-app.py

Predictive Analysis (Classification)

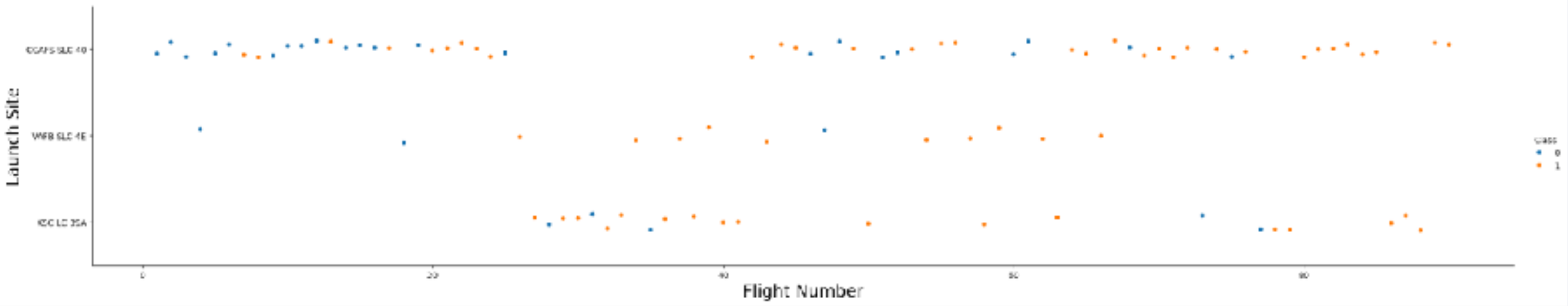
- All the records compiled (90) in the data frame were splitted into target data to predict and evaluate with and predictor features, then in a randomly selected test data set and a train data set (20/80).
- The predictor features X were scaled, so the model didn't overestimate the importance of a feature because of its range.
- The categorical nature of our analysis gave us reason to construct the following models: LogisticRegression, DecisionTreeClassifier, K-Nearest Neighbor and Support Vector Machine.
- Every model was then optimized with the relevant parameters, cross-validated and scored with the GridSearchCV procedure, they they were scored for their accuracy in predictions and plotted their confusion matrices, i.e. the models result on the test data.
- https://github.com/rsevp/IBM/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5(2).ipynb

Section 2

Insights drawn from EDA

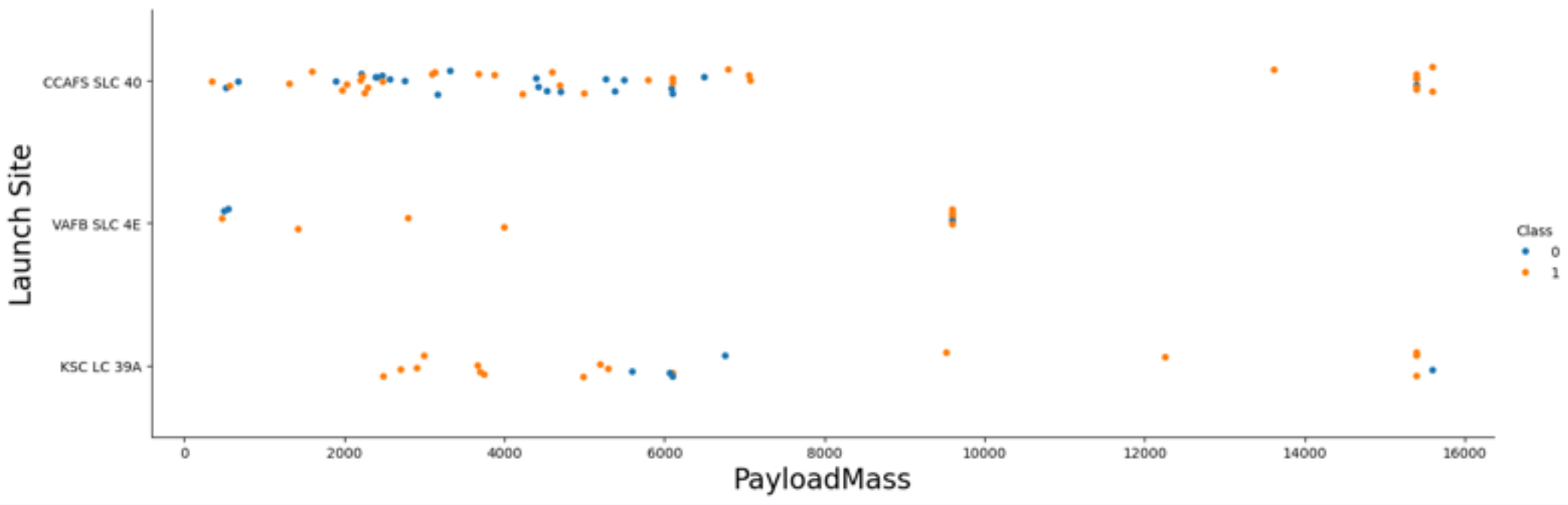
Flight Number vs. Launch Site

- Scatter plot of Flight Number vs. Launch Site: we can see that successful flight rate increase in the latter flights, in the CAAFS site, which was unused during a period (20 to 40), outcome class is encoded in the colors.



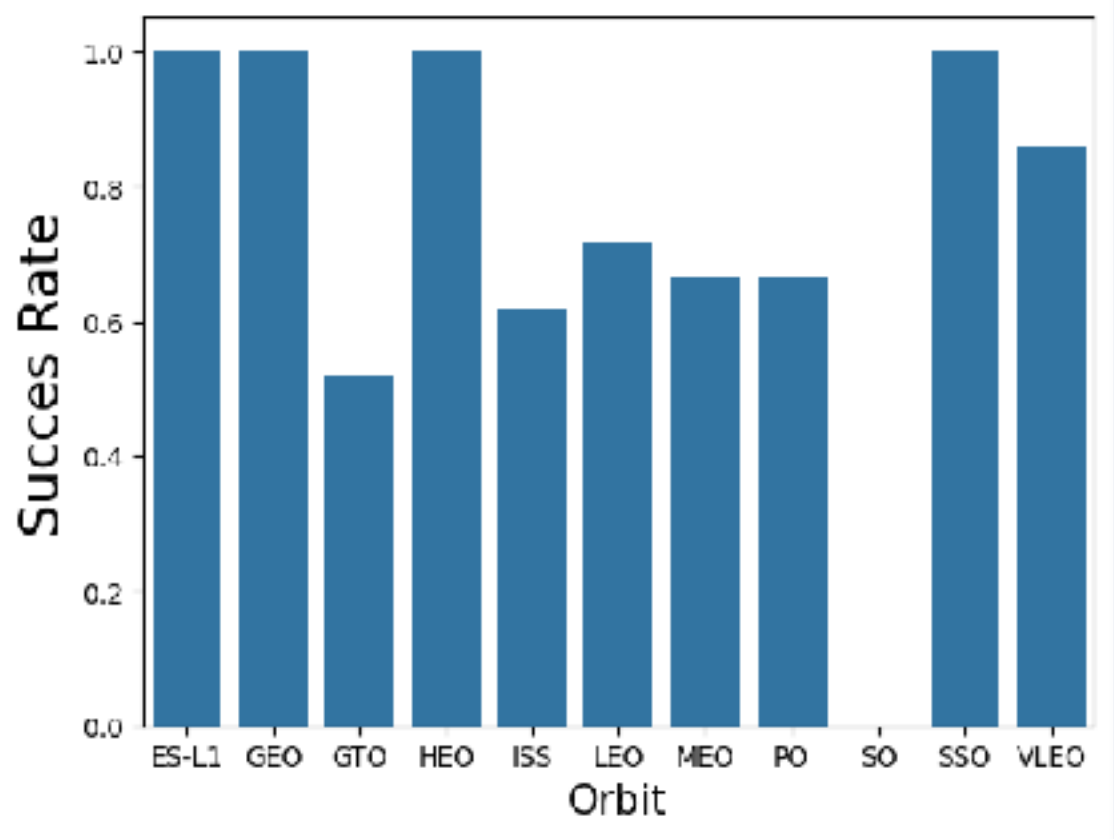
Payload vs. Launch Site

- Scatter plot of Payload vs. Launch Site: we can see VAFB is not suited for heavy flights, outcome class is encoded in the colors..



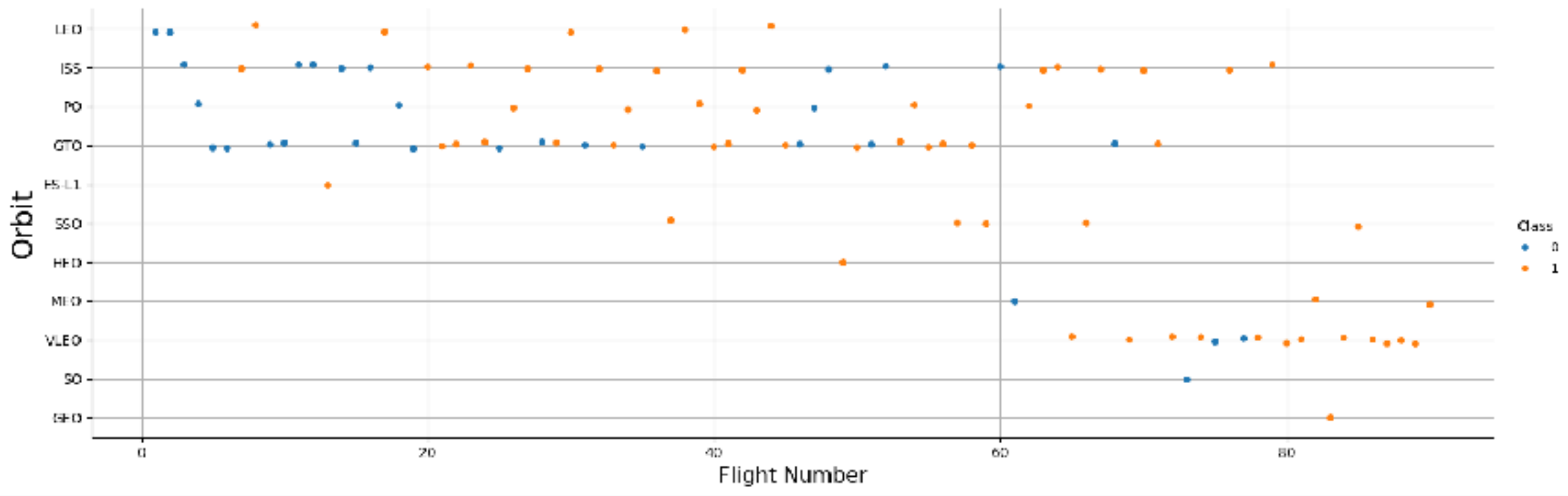
Success Rate vs. Orbit Type

- Bar chart for the success rate of each orbit type, we see the GTO orbit, while heavily demanded its rather unsuccessful.



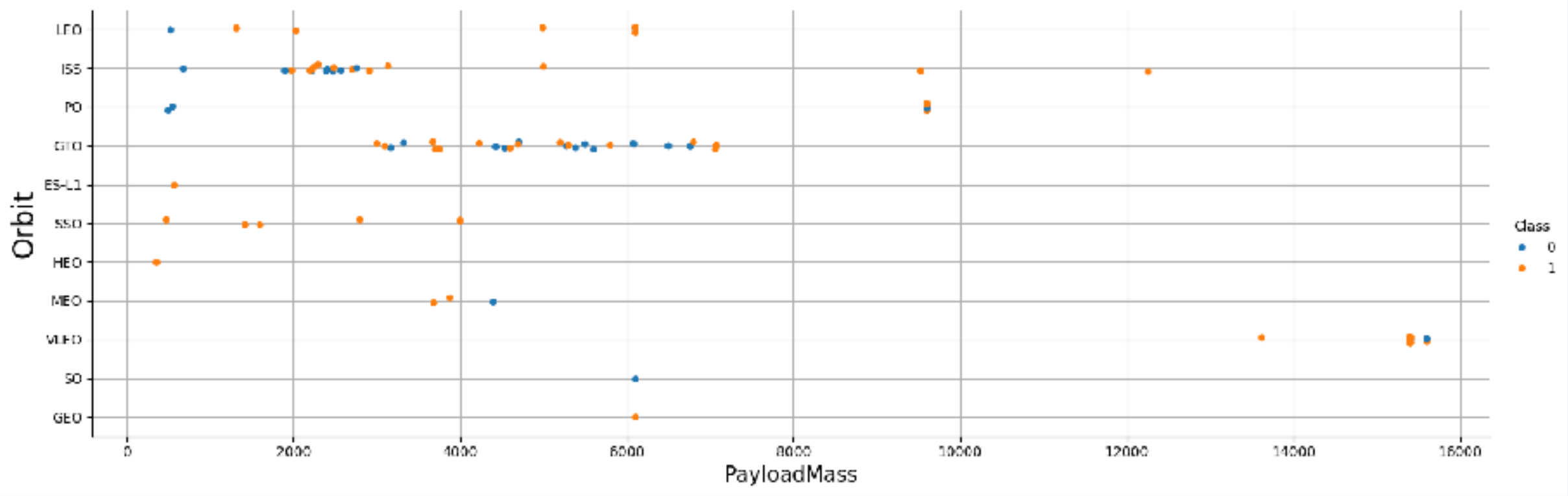
Flight Number vs. Orbit Type

- Show a scatter point of Flight number vs. Orbit type: Heavy flights go mostly to VLEO, LEO & ISS orbit flights are successful, unlike GTO. Outcome class is encoded in the colors.



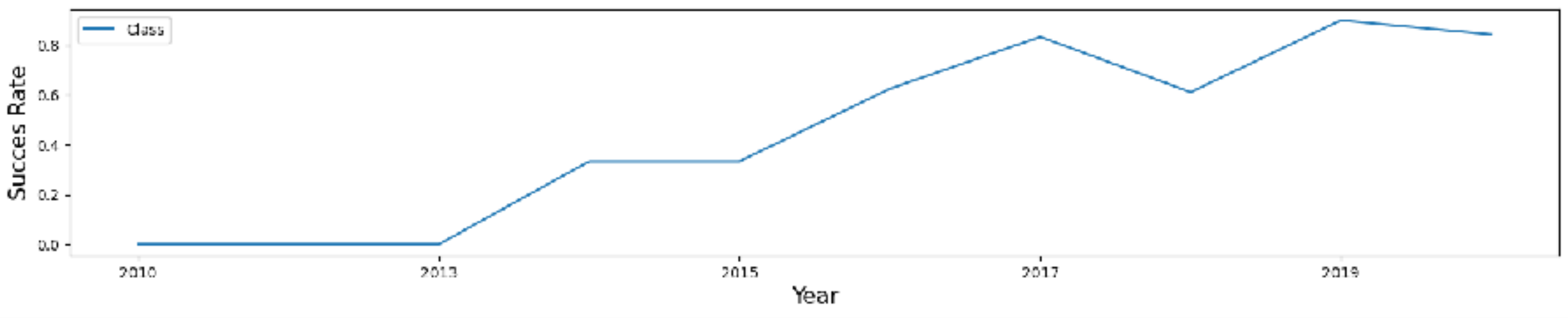
Payload vs. Orbit Type

- Scatter point of payload vs. orbit type: Heavy flights are unusual, around 5 ton there are only GTO, 2 ton are mostly ISS.



Launch Success Yearly Trend

- Line chart of yearly average success rate: the rise is success rate is undeniable but fluctuates greatly every even year (mostly local minimums)



All Launch Site Names

- Find the names of the unique launch sites: All are in coast states.

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

Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with *CCA, as there are two sites contiguous.*

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

- Calculate the total payload carried by boosters from NASA, a high number that needs explanation.

SUM(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1, light flights.

AVG(PAYLOAD_MASS__KG_)
2534.6666666666665

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad, very early and groundbreaking.

MIN("Date")	Landing_Outcome
2015-12-22	Success (ground pad)

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.

Booster_Version	Landing_Outcome
F9 FT B1022	Success (drone ship)
F9 FT B1026	Success (drone ship)
F9 FT B1021.2	Success (drone ship)
F9 FT B1031.2	Success (drone ship)

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes, important in space exploration, that while the landing may fail, the mission outcome is spotless.

COUNT(*)	Mission_Outcome
1	Failure (in flight)

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass. We can see every version is different.

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

2015 Launch Records

- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015, including the month, all records are first semester launches..

substr("Date",6,2)	Landing_Outcome	Booster_Version	Launch_Site
01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40
01	Failure (drone ship)	F9 v1.1 B1017	VAFB SLC-4E
03	Failure (drone ship)	F9 FT B1020	CCAFS LC-40
06	Failure (drone ship)	F9 FT B1024	CCAFS LC-40

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

- 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. Every landing outcome seem to have the same count for failures and success.

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

Section 3

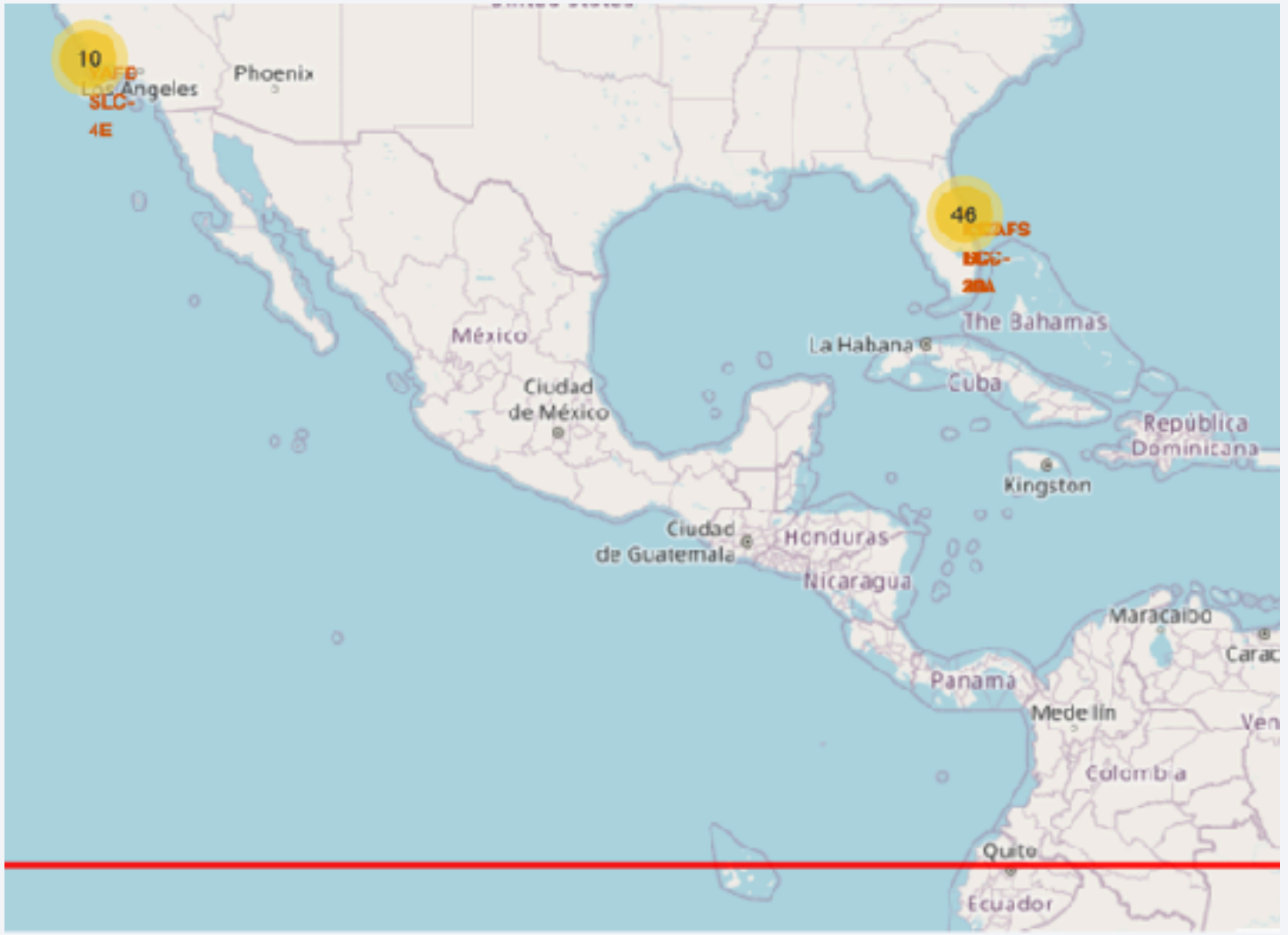
Launch Sites Proximities Analysis

Launch Sites Map Screenshot

- Marked in Red Circles, both on the southmost coastal states. The florida Sites are three, near from each other.

Marker Cluster with every landing outcome recorded.

- The cluster of every launch was added, colored by their class: green is successful, red is failure. An equator line was added.



Lines to the closest city railroad highway and coastline

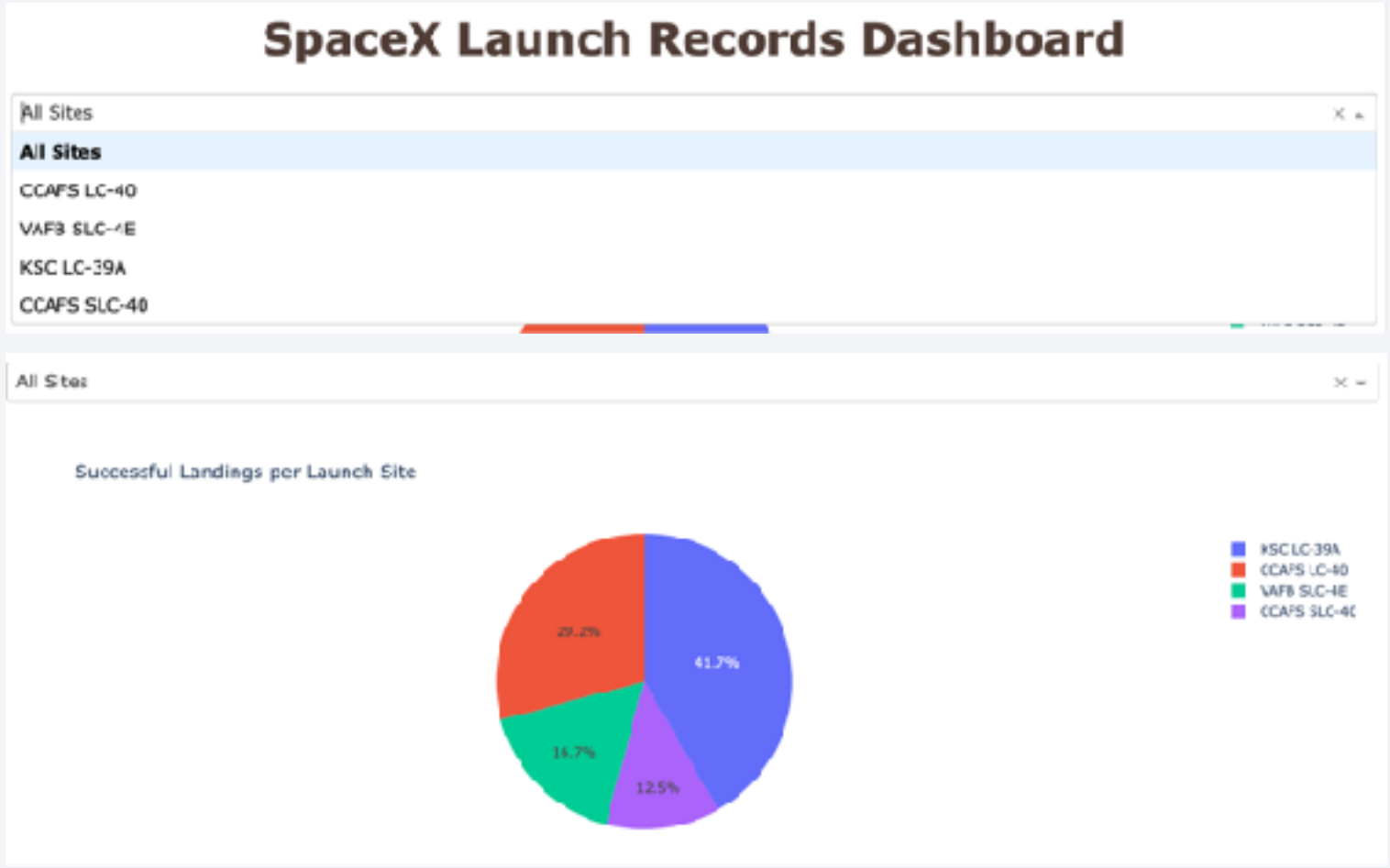
- From the CCAFS SLC-40 site, only cities are away.

Section 4

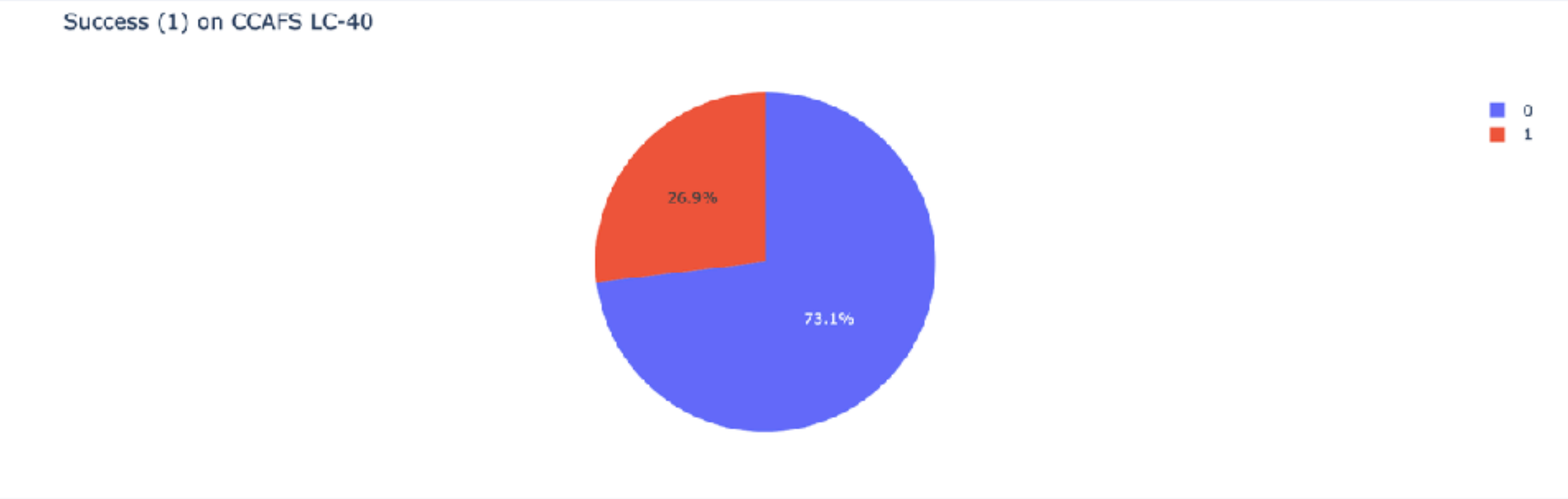
Build a Dashboard with Plotly Dash

Dashboard title and all-site pie chart.

- KSC LC-39A is clearly the most successful launch site. Also, in the title screenshot is included the dropdown menu for clarification.

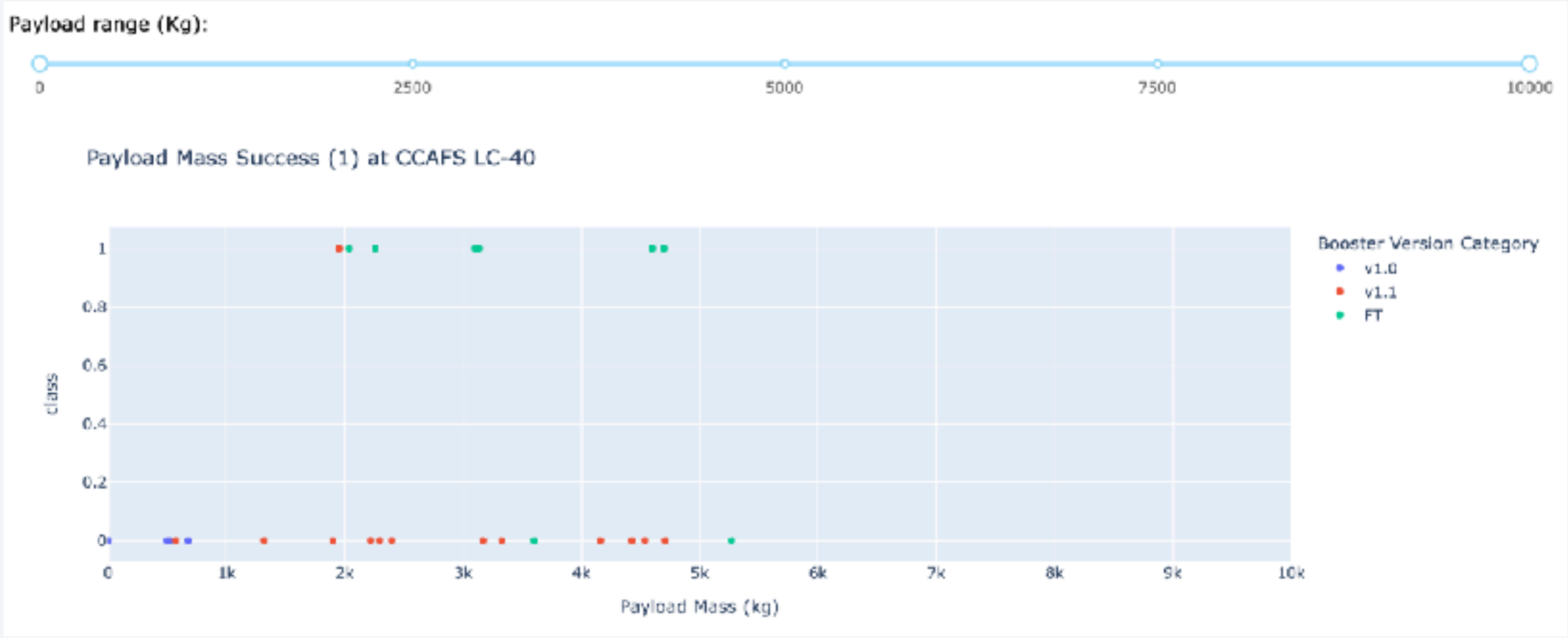


Dashboard pie chart for the most successful site



Dashboard scatterplot

- The plot showing the relationship between the payload mass and the landing outcome, its interactive element is the payload range scrollbar on the top.

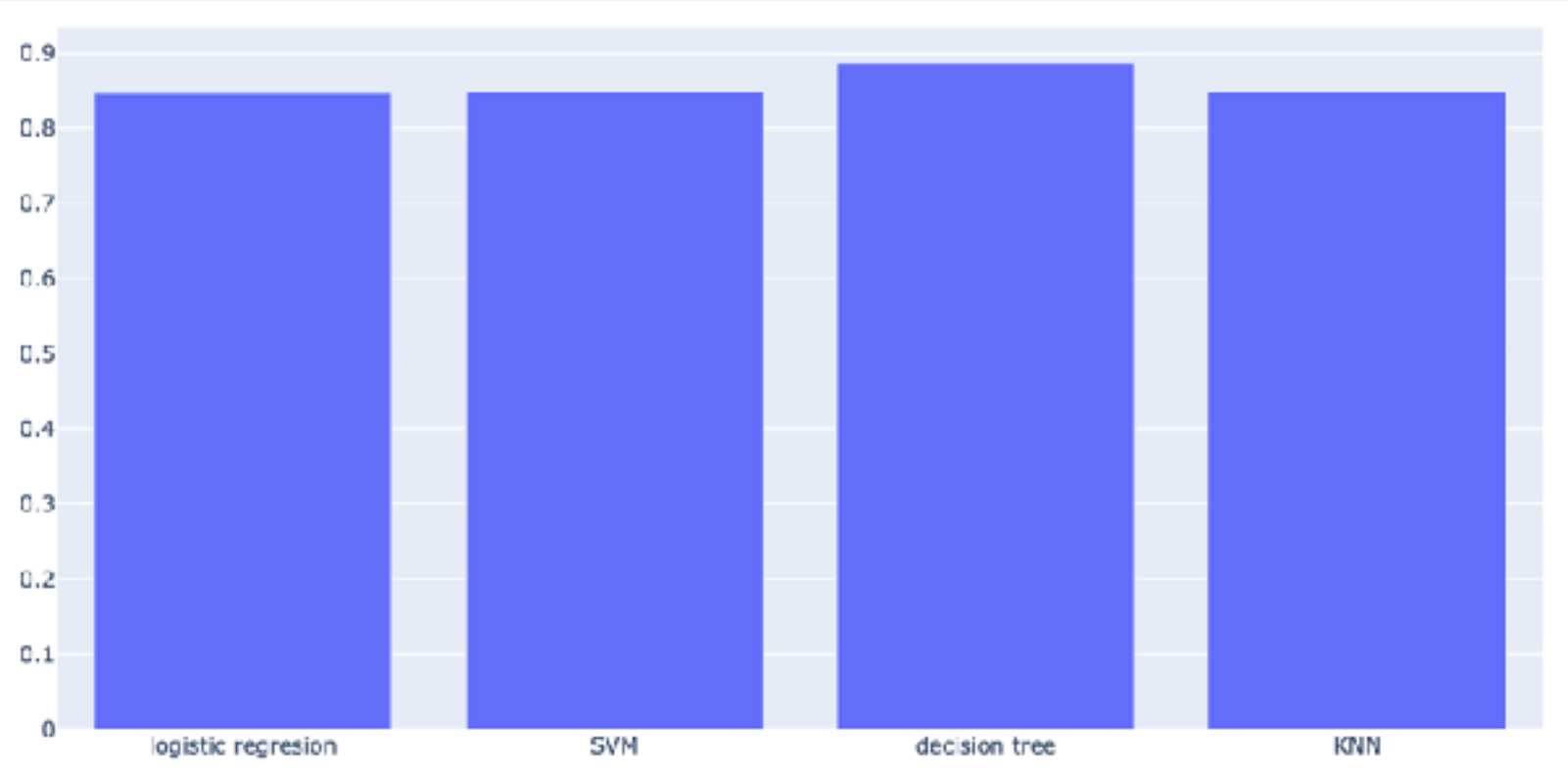


Section 5

Predictive Analysis (Classification)

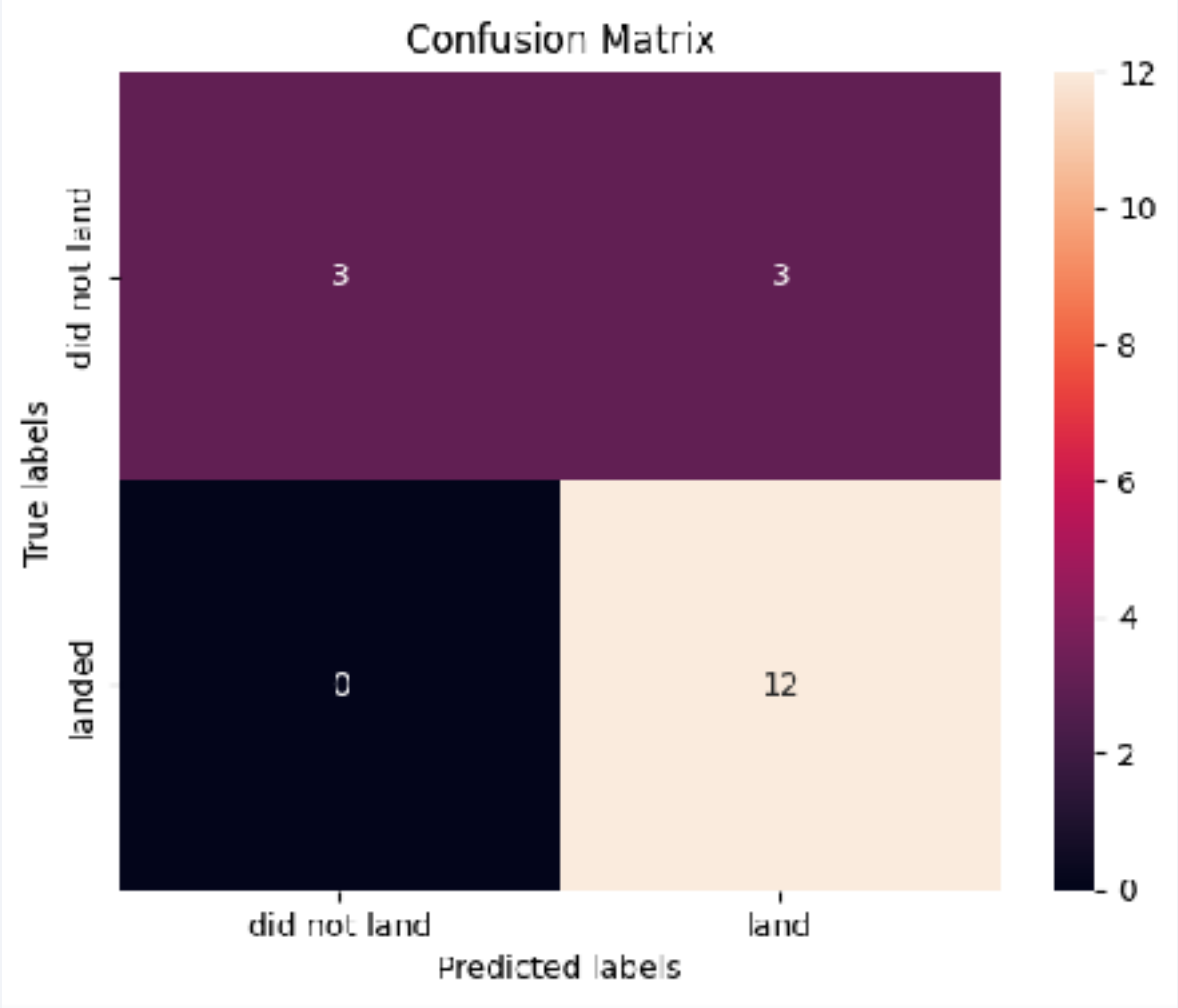
Classification Accuracy

- The accuracy in every model was nearly identical (83% on test data). this can be stated showing the outputted confusion matrices is the same for every model. But the score on the training data is from the Decision Tree Classifier, although every model's score is successful.

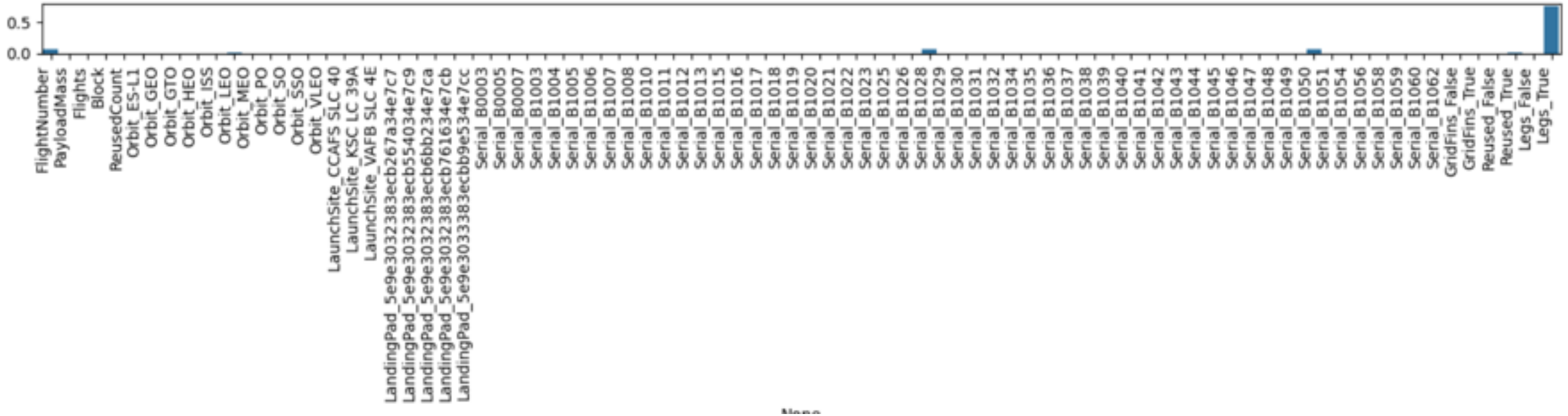


Confusion Matrix

- The diagonal elements are the true negatives and positives, while the best possible scenario is that the off-diagonal elements are cero.
- The negative point is that the model doesn't predict 50 % of the cases where it didn't.



- Furthermore, the most important features were extracted for this model, where the total variance explained is accounted for each feature and the models total. It was seen that surprisingly a technical detail was thrown as the top responsible for the successful prediction: Landing legs.



Conclusions

- The conclusion for the fundamental question Can we predict the landing outcome? Is Yes, given the data described before, a model 83% accurate is able to answer the question not certainly but with high odds.
- The model can improve with a larger sample, perhaps we can include not only spaceX data in the future.
- Other question solved:
  - Which site has the largest successful launches? **KSC-LC 39A**
  - Which site has the highest launch success rate? **CCAFS SLAC-40**
  - Which payload range(s) has the highest launch success rate? **3000 to 4000 kg.**
  - Which payload range(s) has the lowest launch success rate? **4000 ti 7000 kg**
  - Which F9 Booster version (v1.0, v1.1, FT, B4, B5, etc.) has the highest launch success rate? **FT**
  - Are launch sites in close proximity to railways? **Yes, 7.4 KM (from CCAFC SLC-40)**
  - Are launch sites in close proximity to highways? **Yes, 0.5 KM (from CCAFC SLC-40)**
  - Are launch sites in close proximity to coastline? **No, 20 KM to Titusville (from CCAFC SLC-40)**
  - Do launch sites keep certain distance away from cities? **Yes**

Appendix

- The decision tree parameters were optimized further for the gridsearchcv to not produce any errors.

```
parameters = {'criterion': ['gini', 'entropy'],
     'splitter': ['best', 'random'],
     'max_depth': [2*n for n in range(1,10)],
     'max_features': ['sqrt'],
     'min_samples_leaf': [1, 2, 4],
     'min_samples_split': [2, 5, 10]}

tree = DecisionTreeClassifier()
```

```
tree_cv = GridSearchCV(estimator=tree, param_grid=parameters, cv=10)
tree_cv.fit(X_train,Y_train)
```

```
GridSearchCV(cv=10, estimator=DecisionTreeClassifier(),
             param_grid={'criterion': ['gini', 'entropy'],
                         'max_depth': [2, 4, 6, 8, 10, 12, 14, 16, 18],
                         'max_features': ['sqrt'],
                         'min_samples_leaf': [1, 2, 4],
                         'min_samples_split': [2, 5, 10],
                         'splitter': ['best', 'random']})
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

```
print("tuned hpyerparameters :(best parameters) ",tree_cv.best_params_)
print("accuracy :",tree_cv.best_score_)
```

```
tuned hpyerparameters :(best parameters)  {'criterion': 'gini', 'max_depth': 4, 'max_features': 'sqrt', 'min_sample
s_leaf': 1, 'min_samples_split': 2, 'splitter': 'random'}
accuracy : 0.8875
```

Thank you!