Winning Space Race with Data Science

Mike P
2/17/22

Outline

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

Executive Summary

- Summary of methodologies
- Summary of all results

Introduction

- Project background and context
- Problems you want to find answers

spacex can save money by reusing the first stage of its rocket launches. If we determine if a first stage will land, we can determine the cost of a launch. most of the work is done by the first stage. Sometimes it lands, sometimes it crashes, or sometimes it is sacrificed for mission parameters like payload, orbit, and customer.

In this scenario, I work for spaceY founded by Allon Musk who is a competitor to SpaceX.

My job is to determine the price of each launch. I will gather information on spaceX and create dashboards for my team.

I will determine if spaceX will reuse the first stage. Instead of rocket science, I will use data science to predict if spacex will reuse the first stage.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Collect data from SpaceX api
  - Webscraped Wikipedia page on List of Falcon 9 and Falcon Heavy launches
- Perform data wrangling
  - We read in the data and checked the values of the columns (nulls, etc.)
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection – SpaceX API

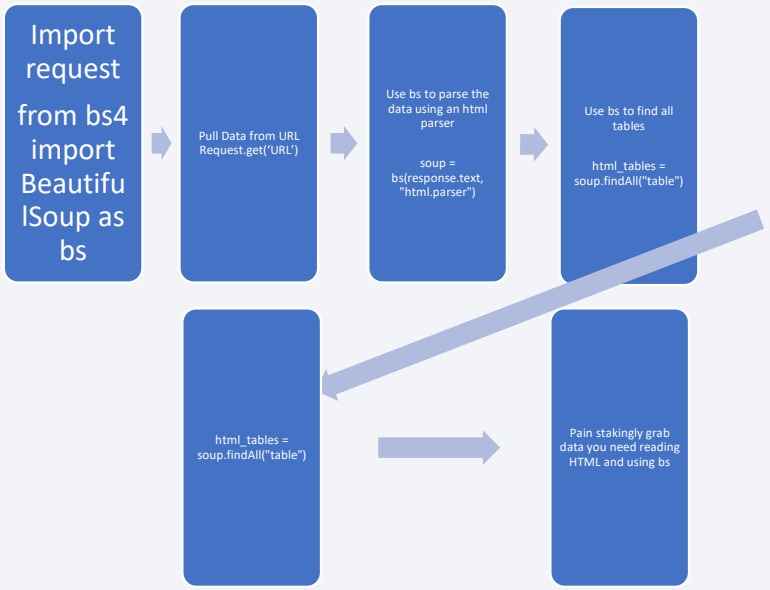
- Present your data collection with SpaceX REST calls using key phrases and flowcharts

- Add the GitHub URL of the completed SpaceX API calls notebook (must include completed code cell and outcome cell), as an external reference and peer-review purpose

Data Collection - Scraping

- Present your web scraping process using key phrases and flowcharts

- Add the GitHub URL of the completed web scraping notebook, as an external reference and peer-review purpose



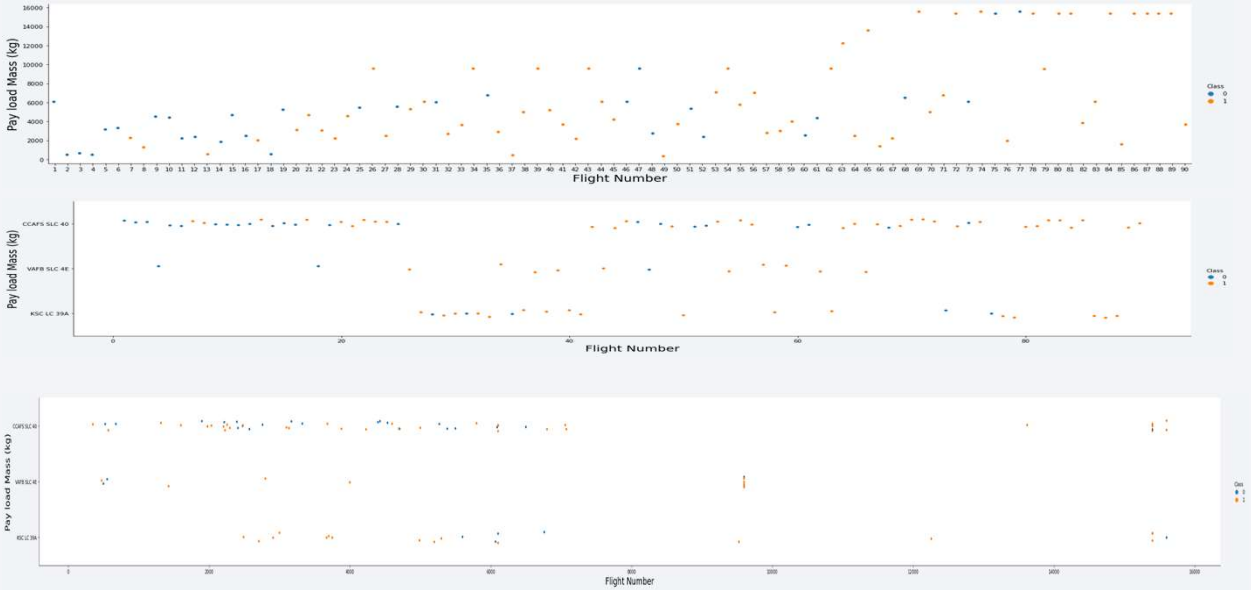
Data Wrangling

- Use pandas and numpy
- Go through dataframe with
- df.head(10)
- df.isnull().sum()/df.count()*100
- df.dtypes
- df['LaunchSite'].value_counts()
- df["Class"].mean()

- https://github.com/mikepalms/coursERA_repo

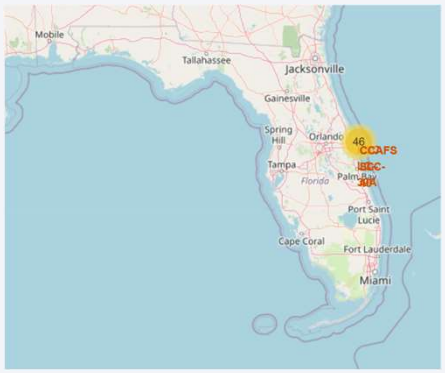
EDA with Data Visualization

- Class = 1 is not a bad outcome, aka a success
- We see that as the flight number increases, the first stage is more likely to land successfully. The payload mass is also important; it seems the more massive the payload, the less likely the first stage will return.
- you can observe that the success rate since 2013 kept increasing till 2020 on the bottom right
- https://github.com/mikepalms/coursERA_repo



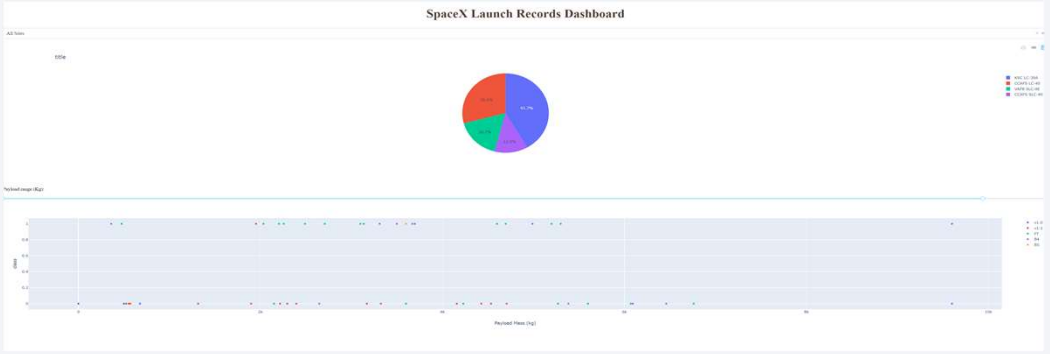
Build an Interactive Map with Folium

- I added a circles for the nasa coordinate in Houston
- Then I added circles for the launch sites in Florida and California
- I added green markers for successful launch outcomes and red markers for unsuccessful outcomes
- I added a line between the coastline and the closest launch site
- https://github.com/mikepalms/coursERA_repo



Build a Dashboard with Plotly Dash

- Added a drop down list to enable launch site selection
- Added a pie chart to show the total successful launches count for all sites
- Added a callback function for `site-dropdown` as input, `success-pie-chart` as output
- Added a callback function for `site-dropdown` and `payload-slider` as inputs, `success-payload-scatter-chart` as output
- https://github.com/mikepalms/coursERA_repo



Predictive Analysis (Classification)

- Goal: If we can determine if the first stage will land, we can determine the cost of a launch
- Perform exploratory Data Analysis and determine Training Labels with .describe()
- create a column for the class with data.Class.to_numpy()
- Standardize the data with preprocessing.StandardScaler().fit(X).transform(X)
- Split into training data and test data with

  X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=4)
- Find best Hyperparameter for SVM, Classification Trees and Logistic Regression with

  lr=LogisticRegression(), svm = SVC(), tree = DecisionTreeClassifier(), & KNN = KNeighborsClassifier()

  grid_search = GridSearchCV(lr, parameters, cv=10)

  logreg_cv = grid_search.fit(X_train, Y_train)
- Find the method performs best using test data with

  print("tuned hpyerparameters :(best parameters) ",tree_cv.best_params_)
  print("accuracy :",tree_cv.best_score_)
  Accuracy: ~.9196
- https://github.com/mikepalms/coursERA_repo

Results

- Exploratory data analysis results

  We see that as the flight number increases, the first stage is more likely to land successfully. The payload mass is also important; it seems the more massive the payload, the less likely the first stage will return.

- Predictive analysis result:

  Our decision tree model performed the best.