

Winning Space Race with Data Science

Cleverlano Gomes

24 May 2023



OUTLINE



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

EXECUTIVE SUMMARY



- Summary of Methodologies
 - SpaceX data collection api
 - SpaceX webscraping Falcon 9
 - SpaceX data wrangling
 - SpaceX EDA using SQL
 - SpaceX EDA data visualization using Pandas
 - SpaceX launch site locations with Folium interactive visual
 - Interactive Dashboard Plotly Dash
 - SpaceX Machine learning prediction
- Summary of all results
 - EDA results
 - Interactive Visual Analytics and Dashboards
 - Predictive Analysis

INTRODUCTION



- Project Background and Context
 - 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.
- Problems you want to find answers
 - In this capstone, we will predict if the Falcon 9 first stage will land successfully using data from Falcon 9 rocket launches advertised on its website.

METHODOLOGY



- Executive Summary
- Data Collection methodology:
 - Describes how data sets were collected
- Perform data wrangling
 - Describes how data were processed
- 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

- Description of how SpaceX Falcon9 data was collected.
 - Data was first collected using SpaceX API (a RESTful API) by making a get request to the SpaceX API. This was done by first defining a series helper functions that would help in the use of the API to extract information using identification numbers in the launch data and then requesting rocket launch data from the SpaceX API url.
 - Finally to make the requested JSON results more consistent, the SpaceX launch data was requested and parsed using the GET request and then decoded the response content as a JsonResult which was then converted into a Pandas data frame.
 - Also performed web scraping to collect Falcon 9 historical launch records from a Wikipedia page titled List of Falcon 9 and Falcon Heavy launches of the launch records are stored in a HTML. Using BeautifulSoup and request Libraries, I extract the Falcon 9 launch HTML table records from the Wikipedia page, Parsed the table and converted it into a Pandas data frame

Data Collection – SpaceX API

- Data collected using SpaceX API (a RESTful API) by making a get request to the SpaceX API then requested and parsed the SpaceX launch data using the GET request and decoded the response content as a JsonResult which was then converted into a Pandas data frame



Here is the GitHub URL of the completed SpaceX API calls notebook
(<https://github.com/cleverlanio/Space-Y/blob/master/1.spacex-data-collection-api.ipynb>)

Data Collection - WebScraping

- Here is the GitHub URL of the completed web scraping notebook.
 - <https://github.com/cleverlanio/Space-Y/blob/master/2.spacex-web scraping%20Falcon%209.ipynb>
- Performed web scraping to collect Falcon 9 historical launch records from a Wikipedia using BeautifulSoup and request, to extract the Falcon 9 launch records from HTML table of the Wikipedia page, then created a data frame by parsing the launch HTML.

EDA with Data Visualization

- Performed data Analysis and Feature Engineering using Pandas and Matplotlib.i.e.
 - Exploratory Data Analysis
 - Preparing Data Feature Engineerin
- Used scatter plots to Visualize the relationship between Flight Number and Launch Site, Payload and Launch Site, FlightNumberand Orbit type, Payload and Orbit type.
- Used Bar chart to Visualize the relationship between success rate of each orbit type
- Line plot to Visualize the launch success yearly trend.
- Here is the GitHub URL of your completed EDA with data visualization notebook:
 - <https://github.com/cleverlanio/Space-Y/blob/master/5.spacex-EDA-dataviz%20using%20Pandas.ipynb>

EDA with SQL

- The following SQL queries were performed for EDA
 - Display the names of the unique launch sites in the space mission

```
%sql SELECT DISTINCT LAUNCH_SITE as "Launch_Sites" FROM SPACEXTBL;
```

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

```
%sql SELECT * FROM 'SPACEXTBL' WHERE Launch_Site LIKE 'CCA%' LIMIT 5;
```

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

```
%sql SELECT DISTINCT Booster_Version, Payload FROM SPACEXTBL WHERE "Landing_Outcome" = "Success (drone ship)" AND PA
```

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

```
%sql SELECT AVG(PAYLOAD_MASS_KG_) as "Payload Mass kgs", Customer, Booster_Version From 'SPACEXTBL' WHERE Booster_Ve
```

EDA with SQL (Cont...)

- List the date when the first successful landing outcome in ground pad was achieved

```
%sql SELECT MIN(DATE) FROM 'SPACEXTBL' WHERE "Landing _Outcome" = "Success (ground pad)";
```

- List the total number of successful and failure mission outcomes

```
%sql SELECT "Mission_Outcome", COUNT("Mission_Outcome") as Total FROM SPACEXTBL GROUP BY "Mission_Outcome";
```

- Here is the GitHub URL of your completed EDA with SQL notebook:
 - <https://github.com/cleverlanio/Space-Y/blob/master/4.spacex-EDA-using-SQL.ipynb>

Build an Interactive Map with Folium

- Created folium map to marked all the launch sites, and created map objects such as markers, circles, lines to mark the success or failure of launches for each launch site.
- Created a launch set outcomes (failure=0 or success=1)
- Here is the GitHub URL of the completed interactive map with Folium map, as an external reference and peer-review purpose:
 - <https://github.com/cleverlanio/Space-Y/blob/master/6.spacex%20Launch%20Site%20Locations%20with%20Folium-Interactive%20Visual.ipynb>

Build a Dashboard with Plotly Dash

- Built an interactive dashboard application with Plotly dash by:
 - Adding a Launch Site Drop-down Input Component
 - Adding a callback function to render success-pie-chart based on selected site dropdown
 - Adding a Range Slider to Select Payload
 - Addeng a callback function to render the success-payload-scatter-chart scatter plot
- Here is the GitHub URL of your completed Plotly Dash lab:
 - https://github.com/cleverlanio/Space-Y/blob/master/7.Interactive%20Dashboard%20Ploty%20Dash%20-%20spacex_dash_app.py

Predictive Analysis (Classification)

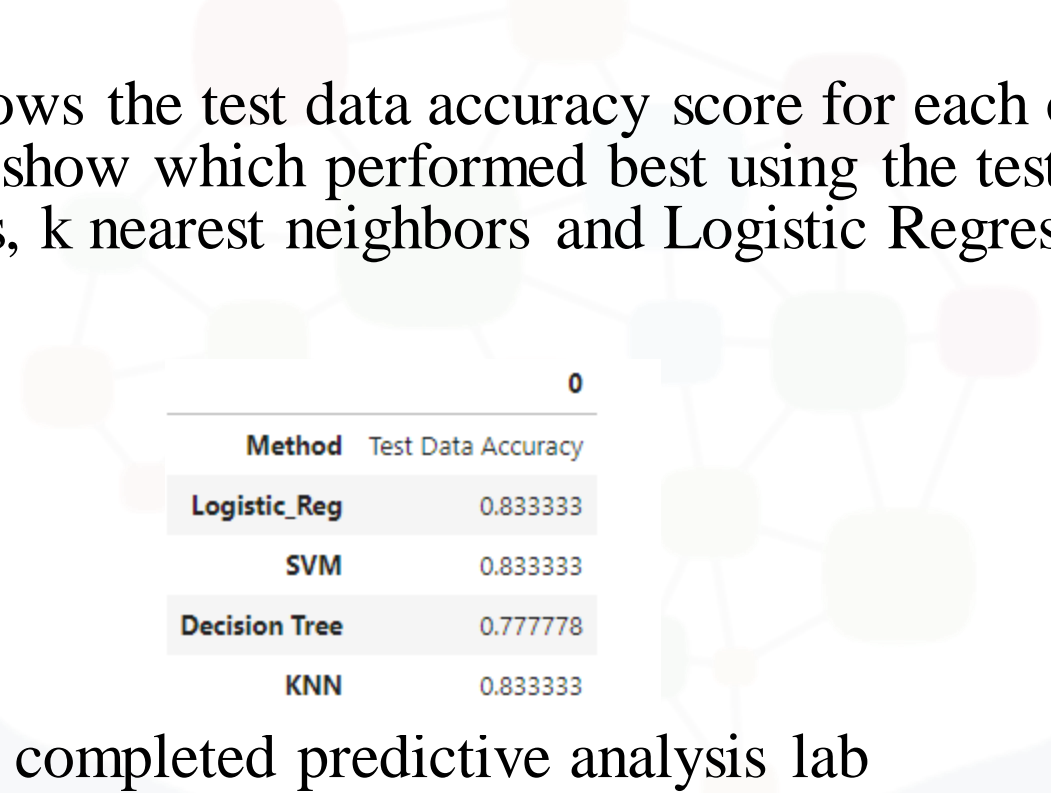
- Summary of how I built, evaluated, improved, and found the best performing classification model
- After loading the data as a Pandas Dataframe, I set out to perform exploratory Data Analysis and determine Training Labels by;
 - creating a NumPy array from the column Class in data, by applying the method `to_numpy()` then assigned it to the variable Y as the outcome variable
 - Then standardized the feature dataset (x) by transforming it using `preprocessing.StandardScaler()` function from Sklearn
 - After which the data was split into training and testing sets using the function `train_test_split` from `sklearn.model_selection` with the `test_size` parameter set to 0.2 and `random_state` to 2.

Predictive Analysis (Classification)

- In order to find the best ML model/ method that would performs best using the test data between SVM, Classification Trees, k nearest neighbors and Logistic Regression;
 - First created an object for each of the algorithms then created a GridSearchCV object and assigned them a set of parameters for each model.
 - For each of the models under evaluation, the GridsearchCV object was created with cv=10, then fit the training data into the GridSearch object for each to Find best Hyperparameter.
 - After fitting the training set, we output GridSearchCV object for each of the models, then displayed the best parameters using the data attribute best_params_ and the accuracy on the validation data using the data attribute best_score_.
 - Finally using the method score to calculate the accuracy on the test data for each model and plotted a confussion matrix for each using the test and predicted outcomes.

Predictive Analysis (Classification)

- The table below shows the test data accuracy score for each of the methods comparing them to show which performed best using the test data between SVM, Classification Trees, k nearest neighbors and Logistic Regression;



0	
Method	Test Data Accuracy
Logistic_Reg	0.833333
SVM	0.833333
Decision Tree	0.777778
KNN	0.833333

- GitHub URL of the completed predictive analysis lab
 - https://github.com/cleverlanio/Space-Y/blob/master/8.%20SpaceX_Machine_Learning_Prediction.ipynb

RESULTS

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

PROGRAMMING LANGUAGE TRENDS

Current Year

<Bar chart of top 10 programming languages for the current year goes here.>

Next Year

< Bar chart of top 10 programming languages for the next year goes here.>

PROGRAMMING LANGUAGE TRENDS - FINDINGS & IMPLICATIONS

Findings

- Finding 1
- Finding 2
- Finding 3

Implications

- Implication 1
- Implication 2
- Implication 3



DATABASE TRENDS

Current Year

< Bar chart of top 10 databases for the current year goes here >

Next Year

< Bar chart of top 10 databases for the next year goes here.>

DATABASE TRENDS - FINDINGS & IMPLICATIONS

Findings

- Finding 1
- Finding 2
- Finding 3

Implications

- Implication 1
- Implication 2
- Implication 3

DASHBOARD



<The permanent link of the read-only view of the Cognos dashboard goes here.>

DASHBOARD TAB 1

Screenshot of dashboard tab 1 goes here

DASHBOARD TAB 2

Screenshot of dashboard tab 2 goes here

DASHBOARD TAB 3

Screenshot of dashboard tab 3 goes here

DISCUSSION



OVERALL FINDINGS & IMPLICATIONS

Findings

- Finding 1
- Finding 2
- Finding 3

Implications

- Implication 1
- Implication 2
- Implication 3

CONCLUSION



- Point 1
- Point 2
- Point 3
- Point 4

APPENDIX



- Include any relevant additional charts, or tables that you may have created during the analysis phase.

JOB POSTINGS

In Module 1 you have collected the job posting data using Job API in a file named “job-postings.xlsx”. Present that data using a bar chart here. Order the bar chart in the descending order of the number of job postings.

POPULAR LANGUAGES

In Module 1 you have collected the job postings data using web scraping in a file named “popular-languages.csv”. Present that data using a bar chart here. Order the bar chart in the descending order of salary.