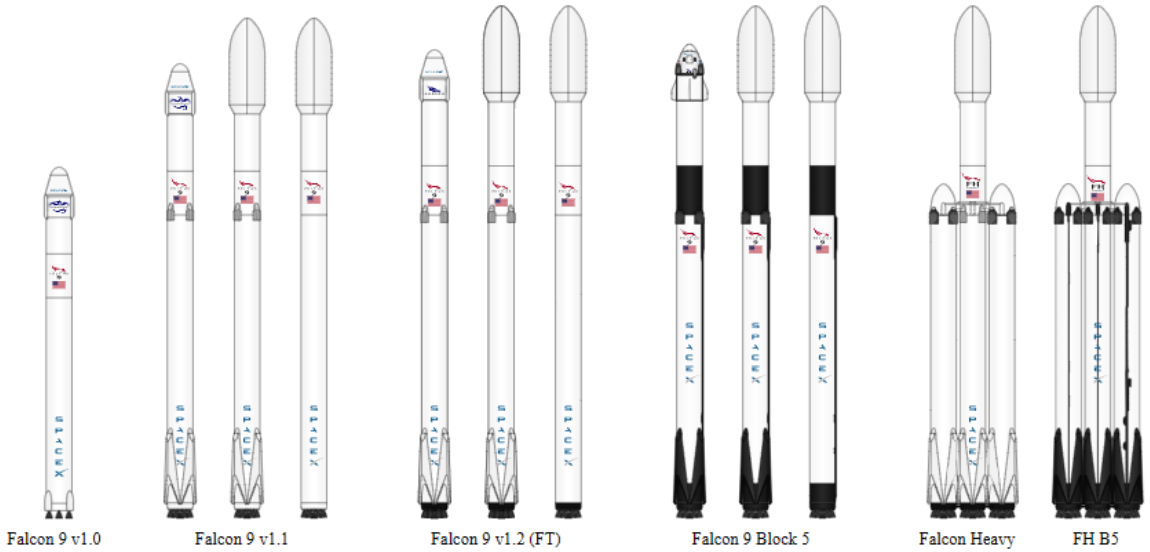


Predictive Analysis of First Stage Rocket Recovery from Falcon 9 Launches

Greg Michalak

6/20/2023

IBM Developer

SKILLS NETWORK

OUTLINE

- Executive Summary
- Introduction
- Methodology
- Results
  - Visualization – Charts
  - Dashboard
- Discussion
  - Findings & Implications
- Conclusion

EXECUTIVE SUMMARY

- Data Collection
  - Accessing the SpaceX API to collect launch data
  - Webscraping the SpaceX wikipedia page to collect launch data
- Data Wrangling/Exploration
  - Using the Pandas library and functions
  - Using the sqlalchemy library and sql queries
- Data Exploration/Visualization
  - Seaborn and Matplotlib libraries for visualization
- Data Visualization
  - Folium maps
  - Interactive Dashborad using Plotly
- Predictive Model Development and Testing

INTRODUCTION

- SpaceX has over 10 years of publicly available historical data about the rocket lunch missions it has performed
- They have been able to reduce the cost of a rocket launch by landing the first stage of their Falcon 9 rocket so it can be re-deployed in future missions
- Using data science tools, we can explore the launch data and predict the cost of a launch based on whether it is likely that the first stage will be recovered
- The data can be analyzed using the flexible programming platform, python, which has many built in data science tools and can interface with many other software platforms and languages
- The end deliverable of this project is a comprehensive predictive analysis using supervised learning models that can predict whether the first stage of a Falcon 9 rocket will land given the various launch conditions

METHODOLOGY – Data Collection

Define functions to extract relevant launch data

Obtain .json file from SpaceX API

Parse .json file and create Pandas dataframe with luanch ID's

Use functions to access the API and gather launch data for each ID

Create new Pandas dataframe with launch data and clean/format

- Using the SpaceX API
  - Defined python functions to populate lists of launch data
  - Downloaded the launch data from the API in .json format
  - Populated a dictionary with launch data and converted to a Pandas dataframe
  - Included launches with **only Falcon 9** boosters and filled in missing payload data with mean value of dataset

METHODOLOGY – Data Collection

Define functions to extract relevant launch data

Access data from SpaceX wikipedia url using BeautifulSoup library

Parse beautiful soup object with functions to create data dictionary

Create Pandas dataframe with dictionary

- Webscraping – practical use is for when API is unavailable
  - Again, python functions for extracting the data were defined
  - BeautifulSoup library used to read .HTML data
  - Extracted column names and table data using appropriate identifiers to populate dictionary
  - Write dictionary to Pandas dataframe

METHODOLOGY – Data Wrangling

Import data into Pandas dataframe

Check for null values and data types

Display all launch sites and frequency of orbit types

Assign outcome landing code based on outcome column

Calculate success rate for all launches

- Identifying key data components
  - Missing data
  - Data types
- Breakdown of the data
  - Lauches by site/orbit
  - Outcome of each mission
- Feature engineering
  - Identify "good" and "bad" outcomes
  - Apply a class variable to outcome types
  - Calculate success rate from class frequency

METHODOLOGY – Data Wrangling

Establish connection with sqlite database

Import data into Pandas and convert it into sql table

Perform queries using the %sql command and select statements

- sqlalchemy library was used to access the data
  - Establish database connection
  - Read in data as a Pandas dataframe
  - Convert Pandas dataframe to sql database
- Sql queries were then used to extract trends in the data
  - Lauch site breakdown
  - Average and total payload of the launces
  - Information on performance over time and based on booster type

METHODOLOGY – Data Exploration

Import data into Pandas dataframe

Perform categorical plots, bar plots, and line plots to see data trends

Use the get_dummies() function to map categorical variables to numeric

Cast all columns as float64

- Seaborn and Matplotlib libraries were used in conjuction with Pandas to create data visualizations
  - Catplots to see data trends based on lauch site, orbit, and payload mass
  - Bar/line plots were used to show success rate for each orbit type and over time
- Categorical variables were transformed with "one-hot" encoding and all features were set as numeric to prepare them for model generation

METHODOLOGY – Data Visualization

Import data into Pandas dataframe

Get launch site and NASA HQ coordinates and create markers on folium map

Mark all launches on map and color code by launch outcome

Find proximity of launch site(s) to relevant geographical markers using coordinates and mark on map

- Folium library was used to visualize launch site data
  - Map depicting all launches/outcomes was generated using geographic coordinates of launch sites
  - Each launch was color coded based on outcome
  - Proximity of relevant geographic markers (cities, coastlines, railways, etc.) to the launch sites was calculated and plotted

METHODOLOGY – Data Visualization

Import data into Pandas dataframe

Create Dash application

Create application layout to include dropdown menu, pie chart, slider bar to select payload range, and launch graph

Create callbacks for the pie chart and graph

Run application in web browser

- Python script accessing the Plotly library was used to create an interactive dashboard in a web browser
  - Pie chart of total percentage of successful flights by launch site and success rate per site
  - Scatterplot showing successful and failed launches based on booster version could be filtered by payload mass

METHODOLOGY – Model Development and Testing

Import data into Pandas dataframe

Create confusion matrix function

Create numpy arrays for input features and output responses

Scale and split the data into training and testing sets

Generate models on training data and test on testing data

Perform model evaluation with accuracy scores and confusion matrix

- Data preparation
  - Assigned the launch outcome as the target variable and the one-hot encoded launch conditions as input features
  - Data were cast as numpy arrays, then standardized and split into training and testing subsets
- Predictive models
  - Function to calculate confusion matrix was created
  - Chose 4 supervised predictive models
    - Logistic regression
    - Support vector machine
    - Decision tree
    - K-nearest neighbors
  - Hyperparameter tuning was performed using a grid search
  - Models were evaluated using the accuracy scores that resulted when using the training and testing input features, then applying a confusion matrix

RESULTS

```
In [5]: # Apply value_counts() on column LaunchSite
        df['LaunchSite'].value_counts()
```

```
Out[5]: CCAFS SLC 40    55
        KSC LC 39A      22
        VAFB SLC 4E     13
        Name: LaunchSite, dtype: int64
```

```
In [6]: # Apply value_counts on Orbit column
        df['Orbit'].value_counts()
```

```
Out[6]: GTO      27
        ISS      21
        VLEO     14
        PO        9
        LEO       7
        SSO       5
        MEO       3
        ES-L1     1
        HEO       1
        SO        1
        GEO       1
        Name: Orbit, dtype: int64
```

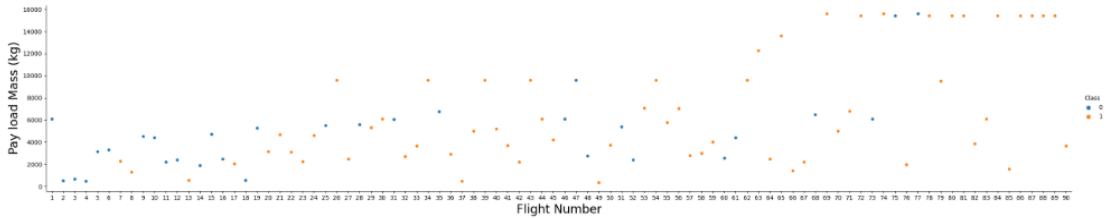
```
In [9]: bad_outcomes=set(landing_outcomes.keys()[[1,3,5,6,7]])
        bad_outcomes
```

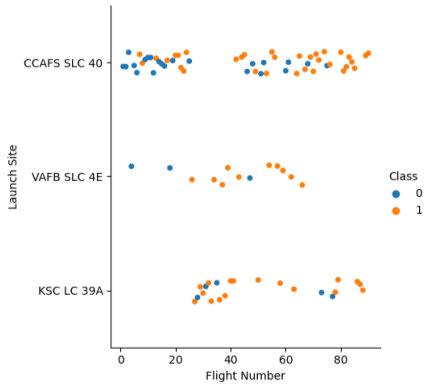
```
Out[9]: {'False ASDS', 'False Ocean', 'False RTLS', 'None ASDS', 'None None'}
```

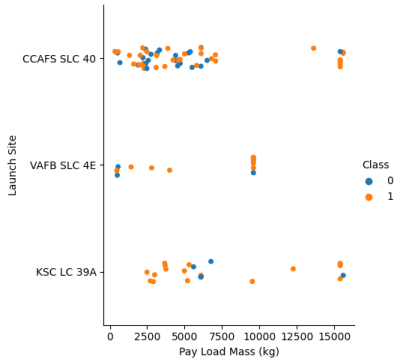
```
In [10]: # landing_class = 0 if bad_outcome
         # landing_class = 1 otherwise
         landing_class = []

         for outcome in df['Outcome']:
             if outcome in bad_outcomes:
                 landing_class.append(0)
             else:
                     landing_class.append(1)

         landing_class
```

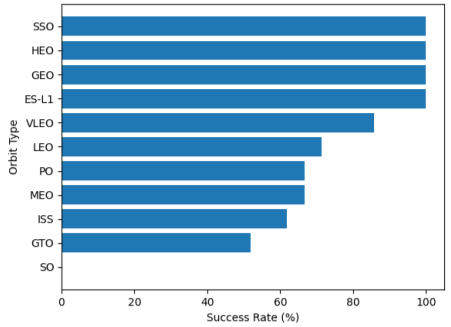
- Data Collection
  - SpaceX API was used to import the data
  - .json data object was parsed and cleaned to include relevant launch data and only missions using the Falcon 9 rocket booster
- Data Wrangling
  - 3 launch sites in total, the most frequent being the Cape Canaveral Space Launch Complex
  - Most frequent orbit type was geosynchronous high Earth orbit
  - Launch outcomes were classified and a success rate of 67% was determined

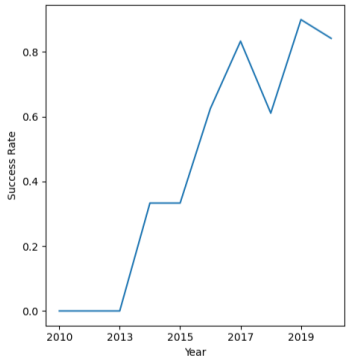
RESULTS





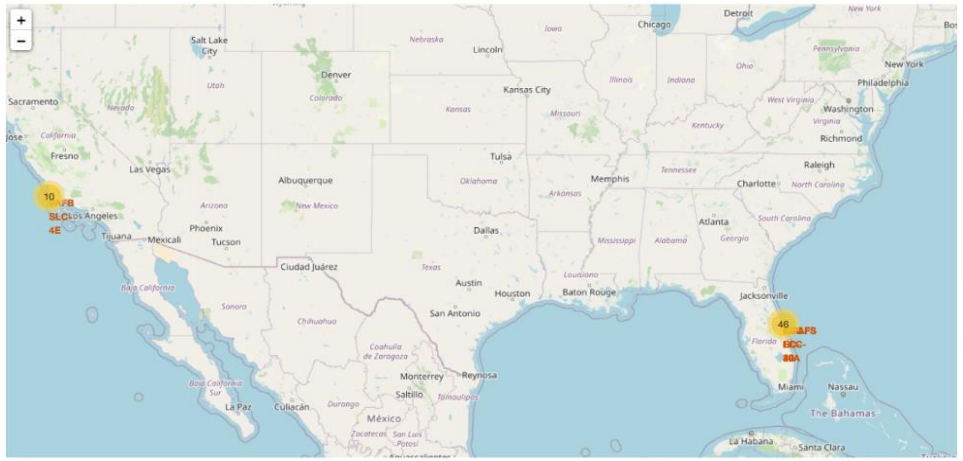


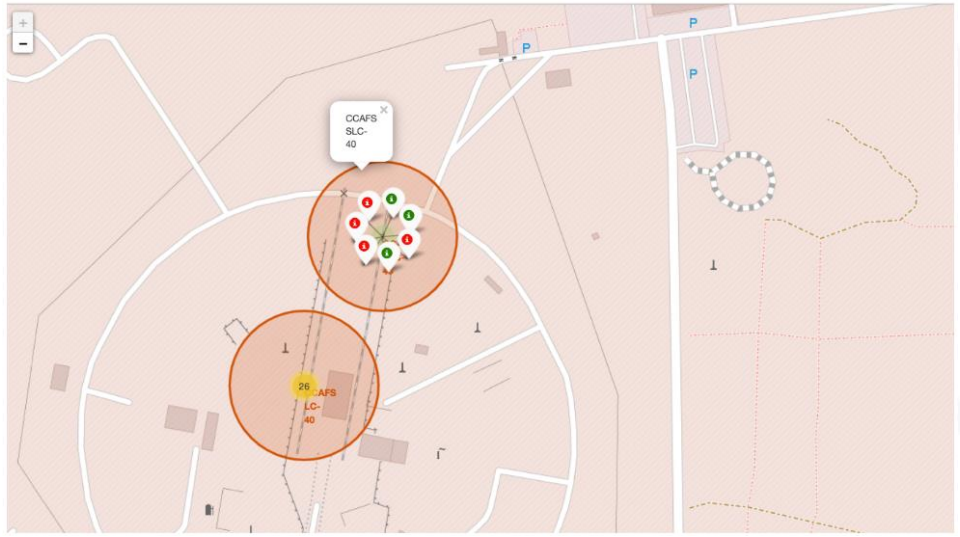




- Data Wrangling
  - Sqlalchemy library was used to create a sql object from a Pandas dataframe
  - Sql queries were performed on the launch data to produce a breakdown based on factors such as launch site, payload, and launch date
- Data Exploration
  - Launches were able to deliver a heavier payload mass more successfully over time
  - Cape Canaveral launch site had the most successful launches, but did not have the highest success rate
  - Heavy payloads (>10000 kg) were only delivered at CCAFS SLC 40 and KSC LC 39A
  - Launches with orbit types SSO, HEO, GEO, and ES-L1 had success rates of 100%, while the most frequent orbit type, GTO, had a success rate of about 50%
  - Launch success rate has steadily improved since 2010

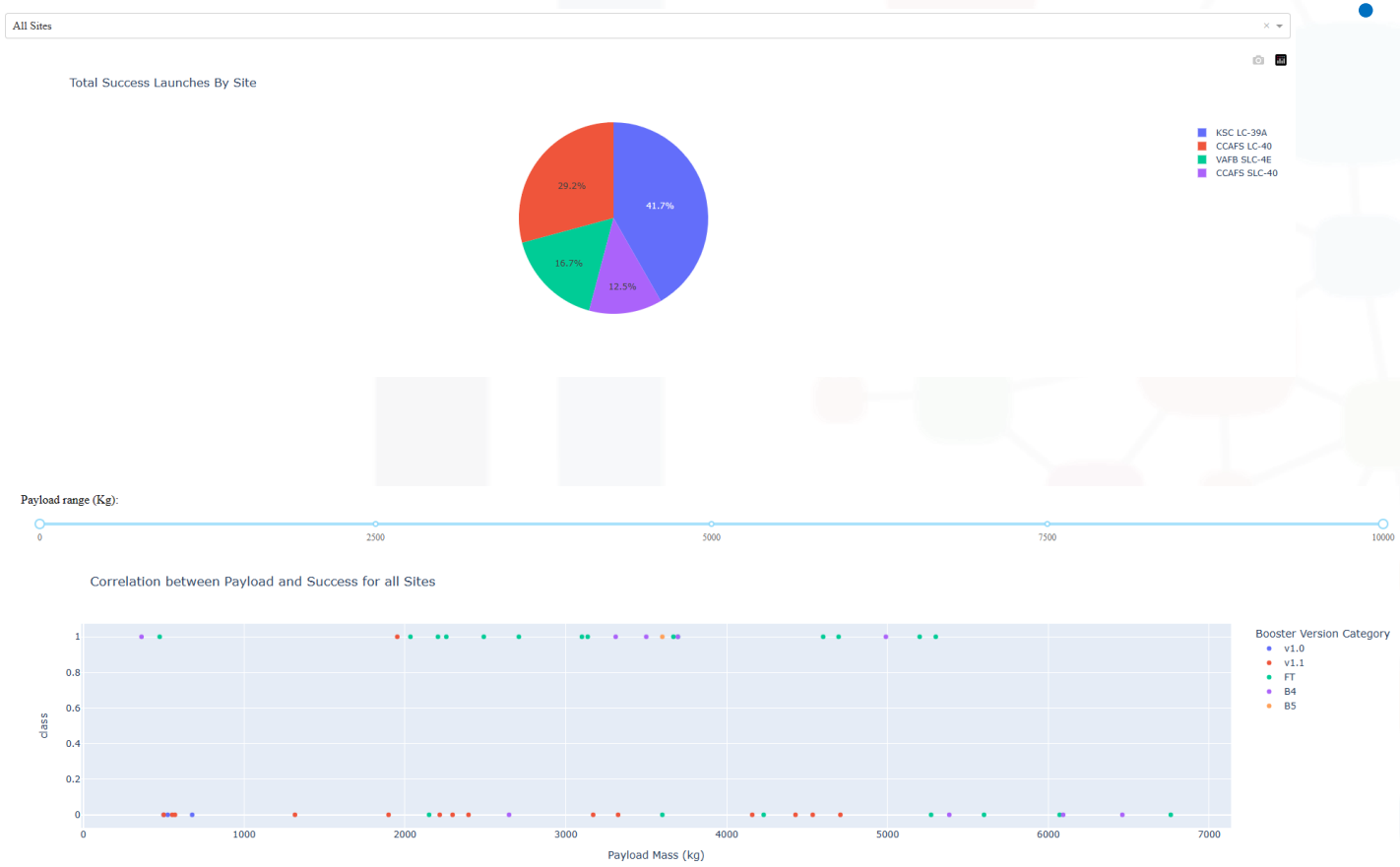
RESULTS





- Data Visualization with Folium Maps
  - All launch sites were on either the east (Florida) or west (southern California) coast
  - Each launch outcome could be seen for each as a color coded marker when zooming into each site
  - Folium map automatically shows latitude and longitude of the cursor, which allows for calculating proximities from launch site to relevant geographic points

RESULTS



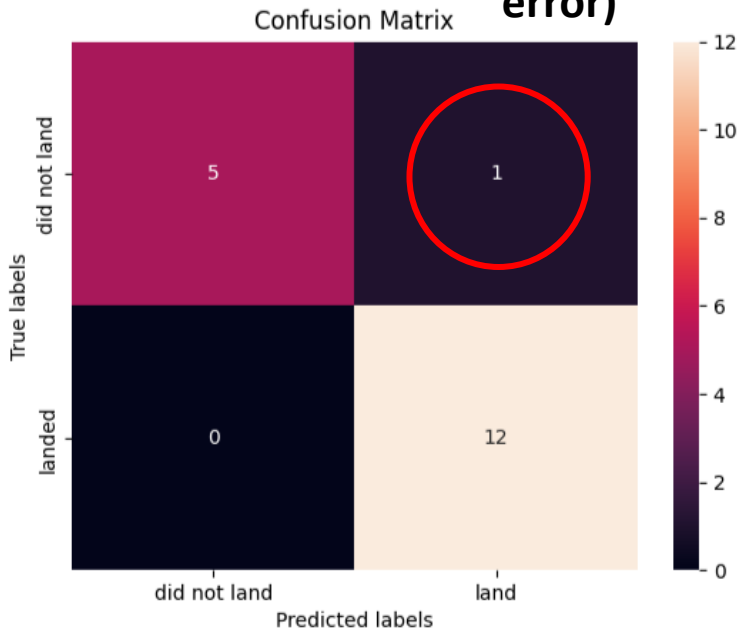
- Data Visualization with Plotly Dashboard
  - Implemented as a .py file that generates a dashboard in a web browser
  - User can select all sites or site-by-site analysis with dropdown menu
  - The user can also view the successful and failed launches broken down by booster version and payload range

RESULTS

Decision Tree had the best overall performance

Model	Training Accuracy (%)	Testing Accuracy (%)
Logistic Regression	0.85	0.83
Support Vector Machine	0.85	0.83
Decision Tree	0.9	0.94
K-nearest Neighbors	0.85	0.83

Only one false prediction (type I error)



Zero type II errors

OVERALL FINDINGS & IMPLICATIONS

Findings

- Overall success rate for SpaceX launches is 67%, which includes the early stages of the company
- Launch sites are located on the coasts and far from cities but close to transportation means
- The best predictive model is the decision tree

Implications

- We should be able to achieve a success rate of at least 70% given that we have predictive models
- We should utilize similar launch sites in order to prevent collateral damages from failed launches and allow access to roads, waterways, railways, etc. Transporting equipment/personnel
- We can develop a comprehensive flowchart to help predict our launch outcomes

CONCLUSION

- There is an abundance of launch data which is publicly accessible
- Often times, data are messy and need extensive data engineering to make them useable. The SpaceX launches seem to be well documented and are now available in a useable format.
- The data have been explored and visualized to gather trends in launch successes. We should use these data to guide the next steps in our business.
- We now have models that can predict launch outcomes with an accuracy of at least 83%. This should guide our decisions on establishing advantageous investments in technology and launch protocols.

Appendix

- Github link for project materials: https://github.com/swagert1/Coursera_IBM_Data_Science_Capstone.git