Winning Space Race with Data Science

Kien Ng

11-October-2021

Outline

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

Executive Summary

- List of project methodologies:
    - Data Collection
    - Data Wrangling
    - Exploratory Data Analysis with Data Visualization
    - Exploratory Data Analysis with SQL
    - Building an Interactive Map with Folium
    - Building a Dashboard with Plotly Dash
    - Predictive Analysis (Classification)

- Results
    - Logistic Regression, SVM, and KNN classification models output similar accuracy score of 83%
    - Decision Tree algorithm accuracy score is 78%

Introduction

- SpaceX advertises Falcon 9 rocket launches on its website with a $62 million price tag per launch compare to typical cost of $165 million and above per launch for other spaceflight companies. Much of the savings is because SpaceX re-uses parts of its rocket.
- If we can accurately predict the likelihood of SpaceX first stage rocket landing successfully, we can determine the cost of a launch.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - SpaceX API
  - Web scraping from Wikipedia
- Perform data wrangling
  - Replace missing and null values with mean value
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Algorithms: Logistic Regression, Support Vector Machines, Decision Tree, K-Nearest Neighbors
  - Tune: use GridSearchCV to tune and select the best hyperparameters of each model
  - Evaluate the model: use score method to evaluate accuracy of each model

Data Collection

- SpaceX API
  - Request SpaceX launch data using GET request
  - Convert the requested JSON results into a Pandas dataframe using json.normalize()
- Web Scraping Wikipedia
  - Perform HTTP GET method to request Falcon9 Launch html page
  - Pull the data out of html and create a Pandas dataframe by parsing the html tables

Data Collection – SpaceX API

Use GET request to request rocket launch data from API

Decode the content response as a json using .json()

Turn it into a Pandas dataframe using .json_normalize()

Request information of selected features from API again

Construct the obtained data into a dictionary

Create a Pandas dataframe from the dictionary

Filter the dataframe to only include Falcon 9 launches

Replace the missing and null values with mean value

Export the data to a csv file

[link to github]

Data Collection – Scraping

Use http GET method to request Falcon 9 launch html page

Create a BeautifulSoup object from html response

Extract all column names from html table header

Create a dictionary with extracted column names

Get launch records from table rows

Fill up the dictionary with parsed launch records

Create a Pandas dataframe from the dictionary

Export the data to a csv file

[link to github]

Data Wrangling

Objective:

- perform Exploratory Data Analysis and determine training labels
- convert successful landing to 1 and unsuccessful landing to 0

Data Analysis:

- Calculate the number of launches on each launching site
- Calculate the number and occurrence of each orbit
- Calculate the number and occurrence of landing outcome per orbit type
- Create a landing outcome label from **Outcome** column
- Create a new **Class** column from landing outcome label on the dataframe, with:

  1 means successful landing

  0 means unsuccessful landing

[link to github]

EDA with Data Visualization

Plotted charts:

- Scatter plot - to observe and show relationships between two numeric variables
    a. Relationship between Flight Number and Launch Site
    b. Relationship between Payload Mass and Launch Site
    c. Relationship between Flight Number and Orbit type
    d. Relationship between Payload Mass and Orbit type
- Bar Chart – to compare things between different groups or categories
    a. Relationship between Flight Number and Launch Site
- Line Plot – shows frequency of data along a number line
    a. Visualize the launch success yearly trend

link to github

EDA with SQL

SQL queries performed:

- Display the names of unique launch sites
- Display five records where launch site names begin with "CCA"
- Display total payload mass carried by boosters launched by NASA (CRS)
- Display payload average mass carried by booster version F9 v1.1
- Get the date of first successful landing outcome in ground pad
- List boosters with success in drone ship and payload mass between 4000 and 6000
- List total number of successful and failed landing outcomes
- List booster versions which have carried maximum payload mass
- List failed landing outcomes in drone ship, their booster versions and launch sites for year 2015
- Rank count of landing outcomes between 2010-06-04 and 2017-03-20 in descending order

[link to github]

Build an Interactive Map with Folium

Map objects added to folium map:

- Circle – using folium.Circle()
  - to circle the coordinates and to draw attention to larger areas
- Marker – using folium.Marker() and MarkerCluster()
  - folium.Marker() - to plot markers on the map
  - MarkerCluster() - combine markers with similar coordinates into one cluster on the map
- Distance – using MousePosition() and folium.Polyline()
  - MousePosition() - to get coordinates for a mouse over a point on the map
  - folium.PolyLine() - to draw a line between coordinates

link to github

Build a Dashboard with Plotly Dash

- Plots and Graphs added to dashboard:
  - Pie chart:
    a. For all launch sites - to show total successful launches count
    b. For a specific launch site – to show Success vs Failed launch count for that site
  - Scatter plot – to show Payload and Launch Success correlation by booster version
- Interactions added to dashboard:
  - Dropdown menu - to enable launch site selection
  - Slider – to enable payload range selection

link to github

Predictive Analysis (Classification)

Create a numpy array from column "Class" - assign it to variable Y

Standardize variable X data with StandardScaler()

Split X and Y data into training and test dataset

Create an object for KNN, SVM, Logistic Regression and Decision Tree

Create GridSearchCV object for each model and find the best parameters

Calculate accuracy on test data for each model using score() method

Plot confusion matrix for each model to evaluate their performance

Compare all models and find the model with the best performance

link to github

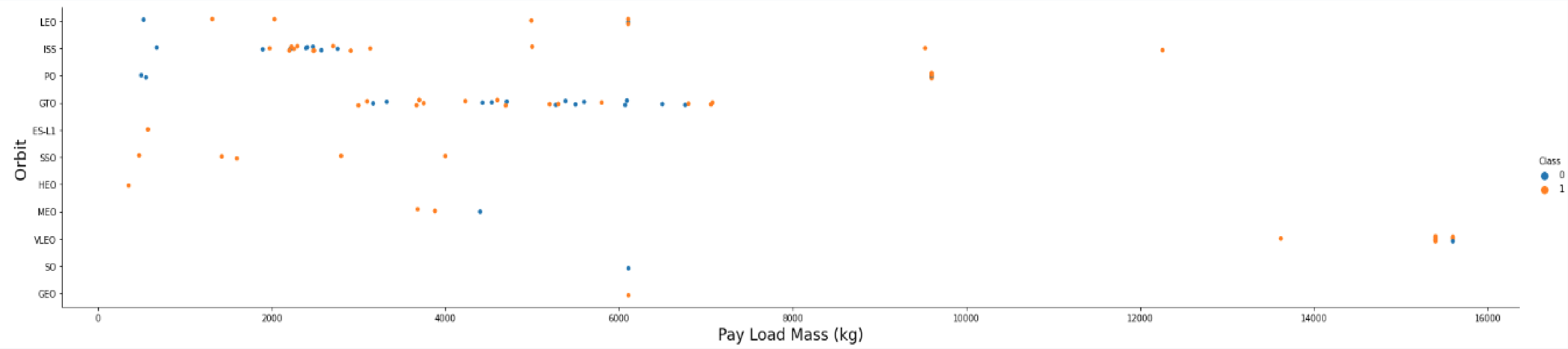
Results

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

Section 2

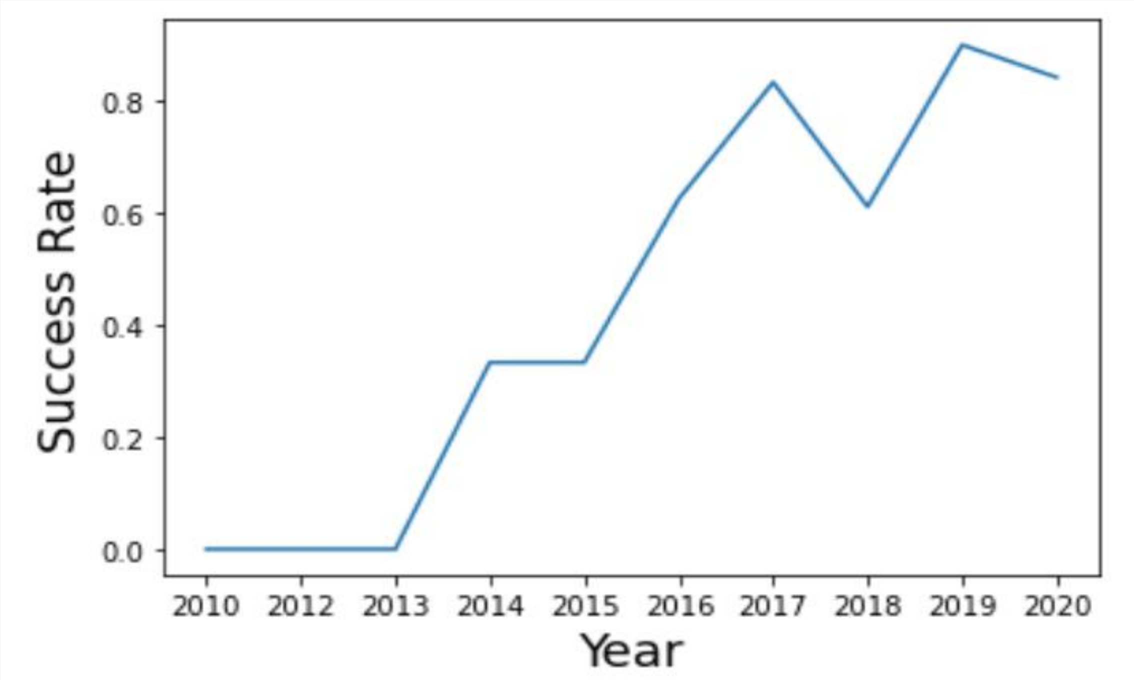
Insights drawn from EDA

Payload vs. Orbit Type



- Heavier payloads have negative influence on GTO orbits and positive influence on Polar LEO orbits.

Launch Success Yearly Trend



Launch success rate has been increasing since 2013. It dropped in 2018 and increase again in 2019. It dipped slightly in 2020.

All Launch Site Names

```
In [6]: %sql SELECT DISTINCT launch_site FROM SPACEX;

         * ibm_db_sa://jjq73128:***@764264db-9824-4
        Done.

Out[6]:
```

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

- Use DISTINCT clause to eliminate all duplicates and fetch only unique records

Launch Site Names Begin with 'CCA'

```
In [7]: %sql SELECT * FROM SPACEX WHERE launch_site LIKE 'CCA%' LIMIT 5;

 * ibm_db_sa://jjq73128:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/bludb
Done.
```

Out[7]:

DATE	Time (UTC)	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	Landing _Outcome
04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
08-12-2010	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)
22-05-2012	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
01-03-2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

- Use LIKE command in a WHERE clause to search for launch sites begin with `CCA`
- Use % as a wildcard and limit the result to five records

Total Payload Mass

```
In [8]: %%sql SELECT sum(payload_mass__kg_) as " Nasa (CRS) Total Payload Mass" FROM SPACEX
        WHERE customer='NASA (CRS)';

         * ibm_db_sa://jjq73128:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqblod
        Done.
```

Out[8]:

Nasa (CRS) Total Payload Mass
45596

- Use SUM() to get total payload mass
- Use WHERE clause to select only booster launched by NASA

Average Payload Mass by F9 v1.1

```
In [9]: %%sql SELECT AVG(payload_mass__kg_) as "Average Payload Mass" FROM SPACEX
        WHERE booster_version='F9 v1.1';

         * ibm_db_sa://jjq73128:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io9(
        Done.
```

Out[9]:

Average Payload Mass
2928

- Use AVG() to get average payload mass
- Use WHERE clause to select only booster version F9 v1.1

First Successful Ground Landing Date

```
In [10]: %%sql SELECT DATE as "First Successful Landing (ground pad)" FROM SPACEX
         WHERE "Landing _Outcome"='Success (ground pad)'
         ORDER BY "Landing _Outcome"='Success (ground pad)' LIMIT 1;

          * ibm_db_sa://jjq73128:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io9
         Done.
```

Out[10]:

First Successful Landing (ground pad)
22-12-2015

- Use WHERE clause to select all successful landing outcome on ground pad and their dates
- Use ORDER BY clause to sort the result in ascending order
- Select the first record using LIMIT clause

Successful Drone Ship Landing with Payload between 4000 and 6000

```
In [11]: %%sql SELECT booster_version FROM SPACEX
         WHERE "Landing _Outcome"='Success (drone ship)'
         AND payload_mass__kg_ > 4000 AND payload_mass__kg_ < 6000;

          * ibm_db_sa://jjq73128:***@764264db-9824-4b7c-82df-40d1l
         Done.

Out[11]:
```

booster_version
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
F9 FT B1031.2

- Use WHERE clause to select boosters successfully landed on drone ship
- Use AND clause to select boosters with payload mass > 4000 and < 6000

Total Number of Successful and Failure Mission Outcomes

```
In [12]: %%sql SELECT mission_outcome, COUNT(*) as "Mission Outcomes" FROM SPACEX
         GROUP BY mission_outcome
         ORDER BY mission_outcome;

 * ibm_db_sa://jjq73128:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io
Done.
```

Out[12]:

mission_outcome	Mission Outcomes
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

- Use COUNT() to calculate mission outcome records
- Use GROUP BY clause to group rows or records with same values
- Use ORDER BY clause to sort the result with mission outcome column in ascending order

Boosters Carried Maximum Payload

```
In [13]: %%sql SELECT booster_version, payload_mass__kg_ FROM SPACEX
         WHERE payload_mass__kg_ =(SELECT MAX(payload_mass__kg_) FROM SPACEX);

 * ibm_db_sa://jjq73128:***@764264db-9824-4b7c-82df-40d1b13897c2.b
Done.
```

Out[13]:

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

- Use WHERE clause to select boosters which have carried the maximum payload mass

2015 Launch Records

```
In [14]: %%sql SELECT DATE, "Landing _Outcome", booster_version, launch_site FROM SPACEX
         WHERE "Landing _Outcome"='Failure (drone ship)' AND DATE LIKE '%2015%';

          * ibm_db_sa://jjq73128:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kql
         Done.
```

Out[14]:

DATE	Landing _Outcome	booster_version	launch_site
10-01-2015	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
14-04-2015	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

- Use WHERE clause to select the failed landing outcomes in drone ship, their booster versions, and launch site names
- Use LIKE command to filter the result for year 2015 only

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

In [15]:
```
%%sql SELECT "Landing _Outcome", COUNT("Landing _Outcome") as "Count" FROM SPACEX
WHERE DATE BETWEEN '04-06-2010' AND '20-03-2017'
GROUP BY "Landing _Outcome"
ORDER BY Count DESC;
```

```
 * ibm_db_sa://jjq73128:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1
Done.
```

Out[15]:

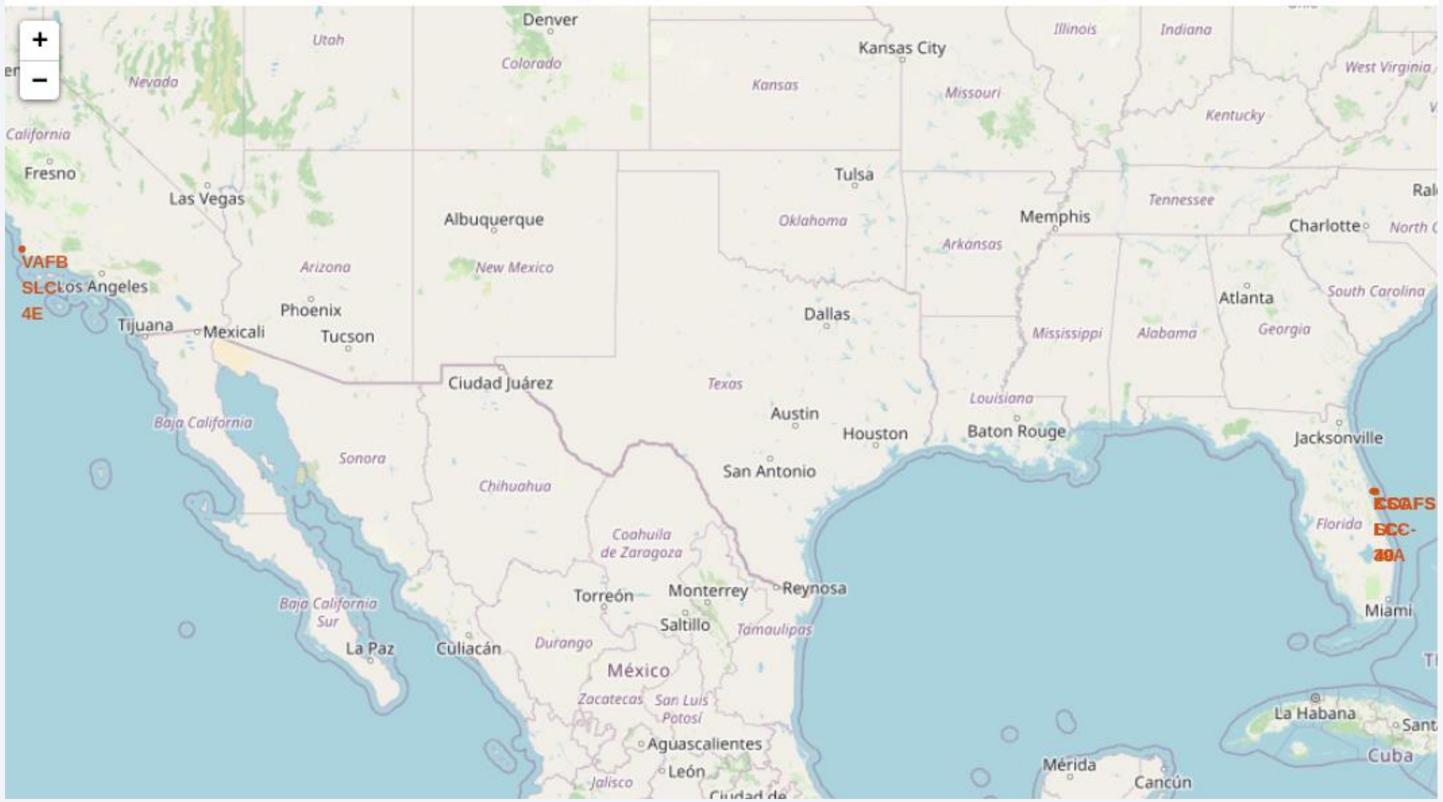
Landing _Outcome	Count
Success	20
No attempt	11
Success (drone ship)	8
Success (ground pad)	6
Failure (drone ship)	4
Controlled (ocean)	3
Failure	3
Failure (parachute)	2

- Use COUNT() to calculate all landing outcome records
- Use WHERE clause to select dates between 2010-06-04 and 2017-03-20
- Use GROUP BY clause to group rows or records with same values
- Use ORDER BY clause to sort the result with new Count column in descending order

Section 4

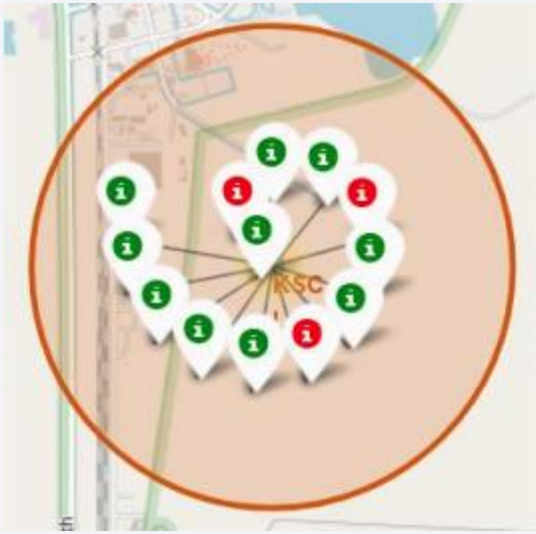
Launch Sites Proximities Analysis

All Falcon 9 Launch Sites Location

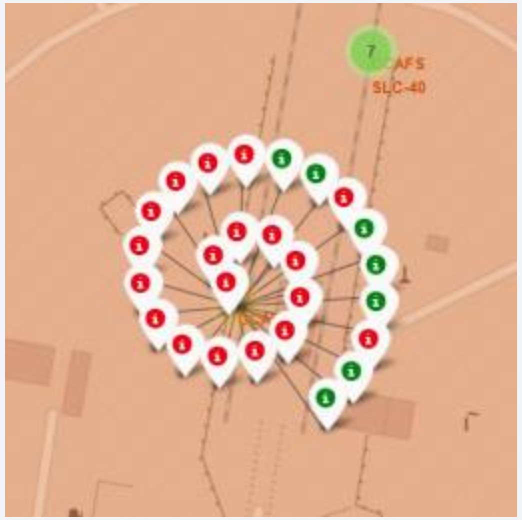


- Most of the launch sites are located along the ocean so the rockets can be safely launched and travel over the open water.

Landing Outcome of Launch Sites



KSC LC-39A



CCAFS SLC-40

- Green label is for successful launching and red label is failed launching
- KSC LC-49A site have more successful launching rate compare to CCAFS LC-40

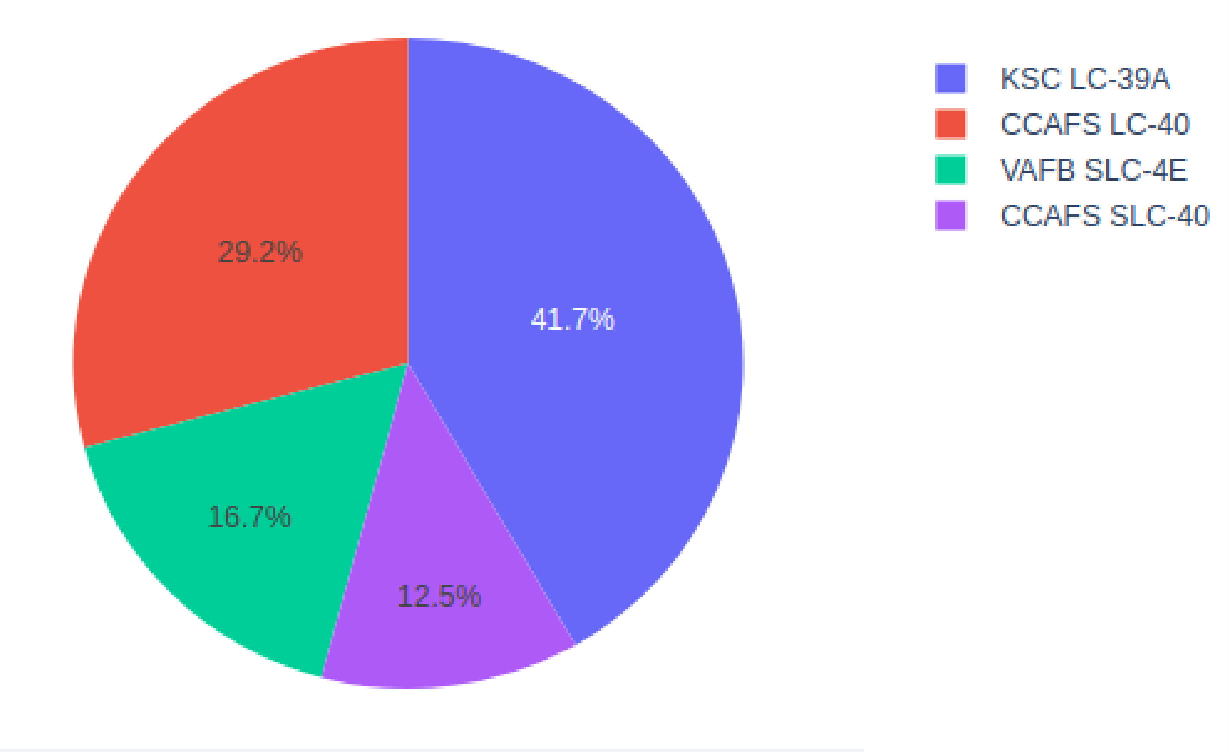
Distance of Launch Site to its Proximities

- CCAFS LC-40 launch site distance from the coastline is about 0.88km

Section 5

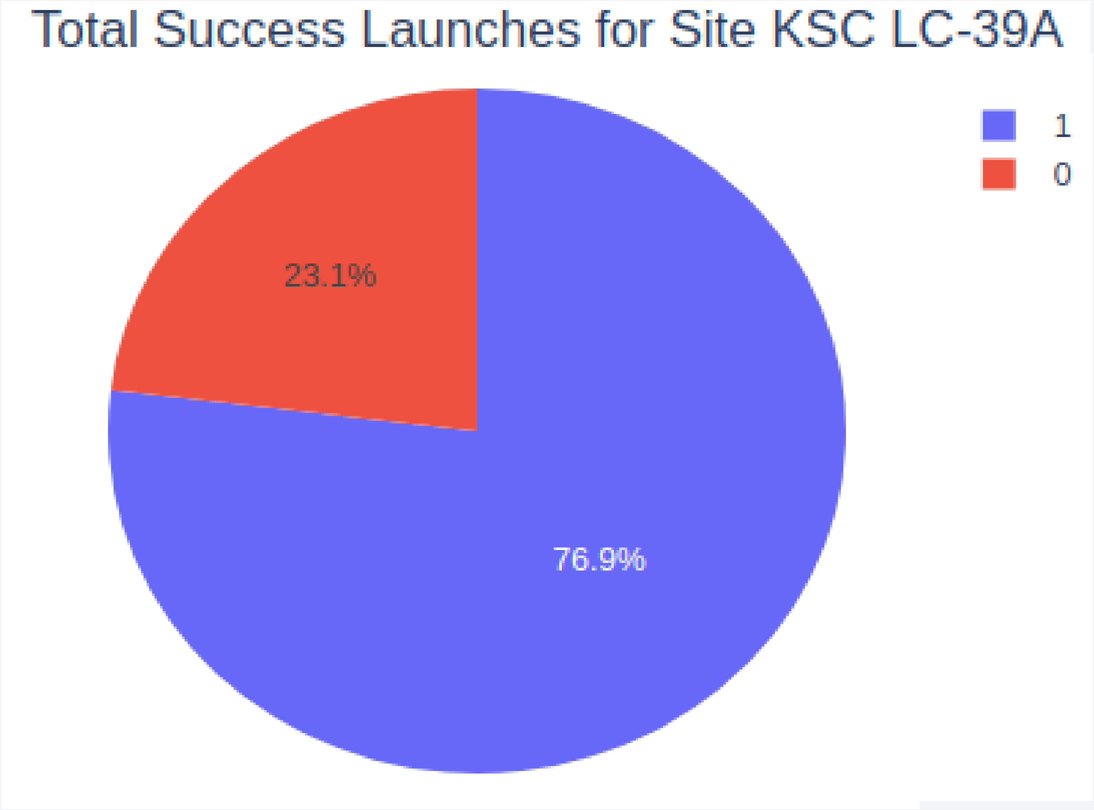
Build a Dashboard with Plotly Dash

Total Success Launches for All Sites



- KSC LC-39A is the site with the most successful launches
- CCAFS SLC-40 is the site with the lowest successful launches

Launch Site with Highest Success Rate



- KSC LC-39A launches ratio with successful launches of 76.9%

Payload vs Launch Outcome for All Sites

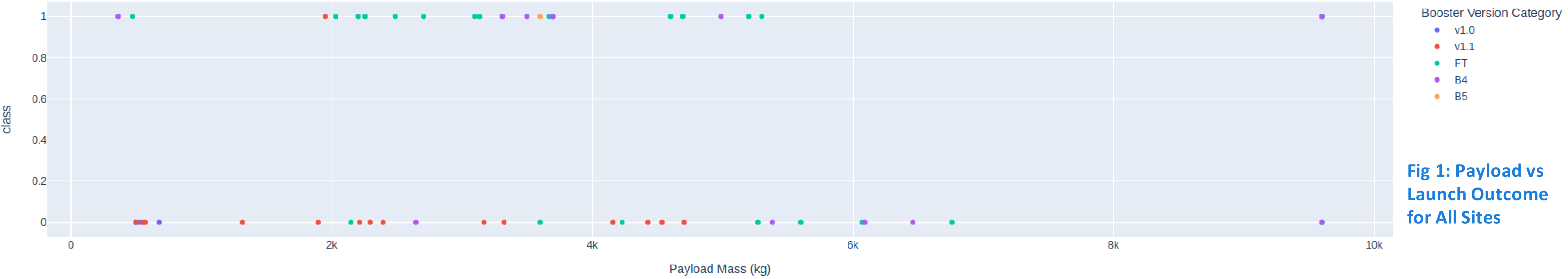


Fig 1: Payload vs Launch Outcome for All Sites

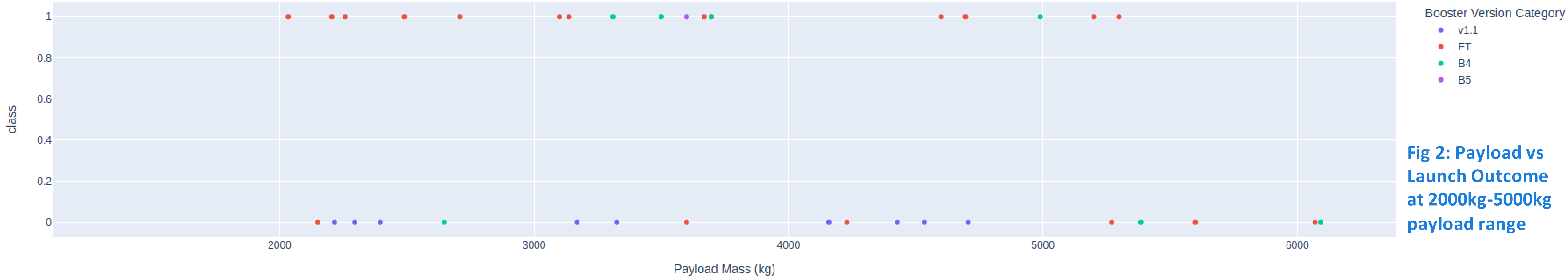


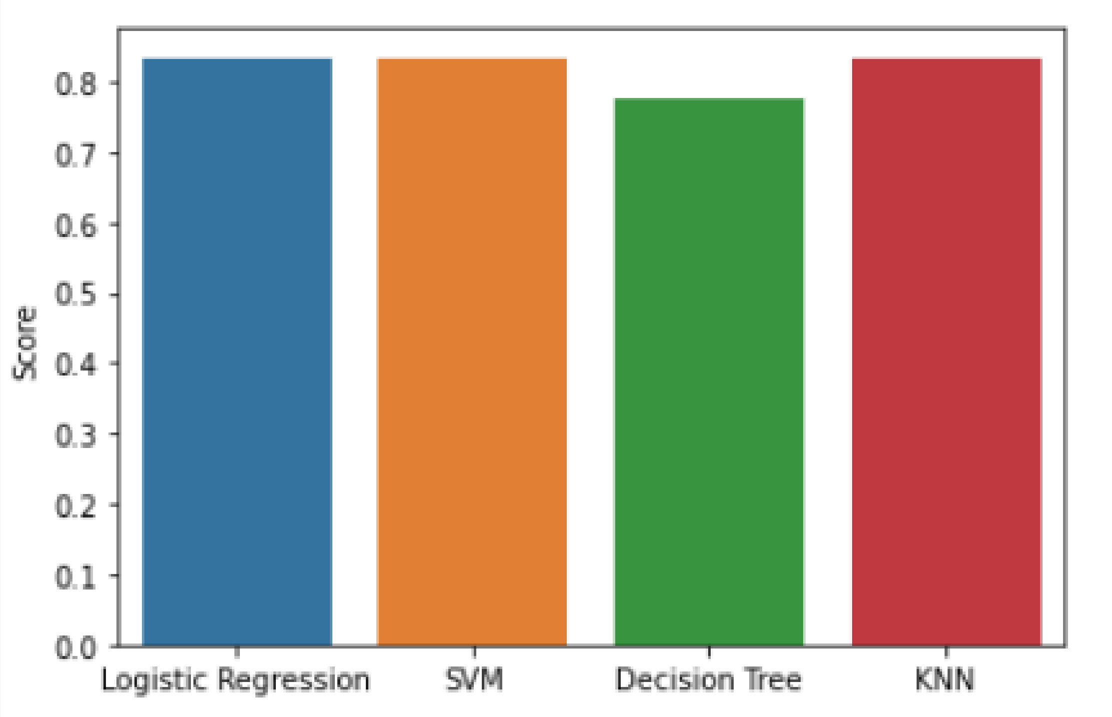
Fig 2: Payload vs Launch Outcome at 2000kg-5000kg payload range

- Majority of successful launches are within the 2000kg-5000kg payload mass range
- FT booster have the most successful launches within this payload range

Section 6

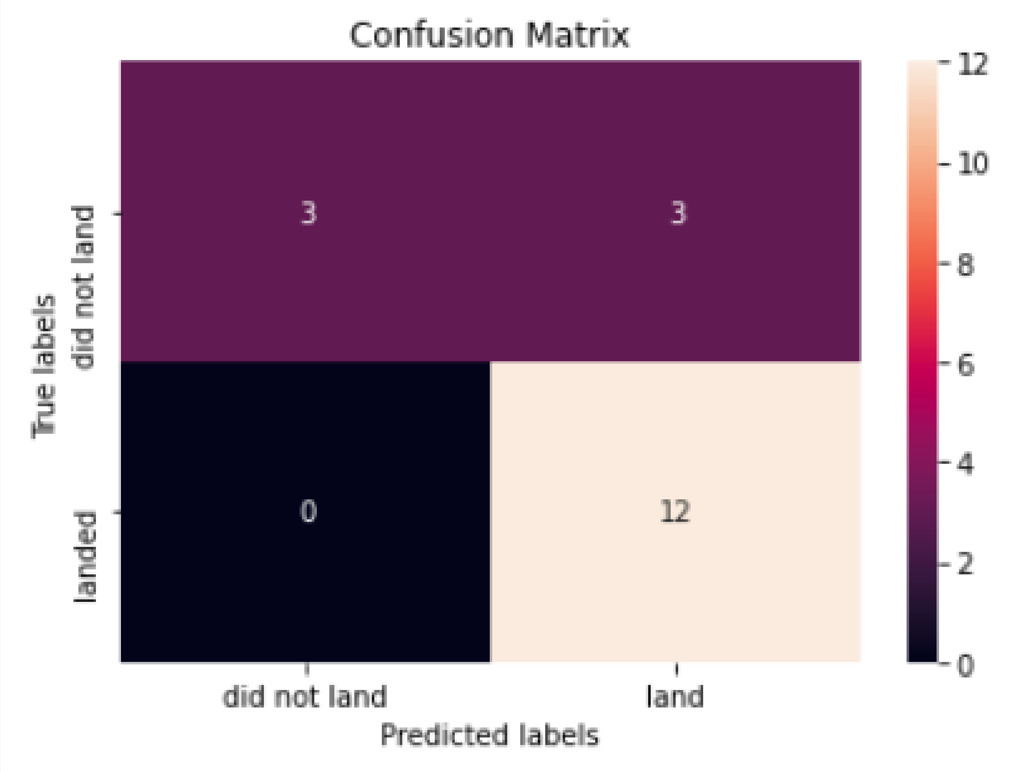
Predictive Analysis (Classification)

Classification Accuracy



- Logistic Regression, SVM and KNN models have similar accuracy score of 83%
- Even though accuracy of Decision Tree algorithm on validation dataset is 90%, but its accuracy score on test dataset is lower at 78%

Confusion Matrix



- Logistic Regression, SVM, and KNN models have similar confusion matrix results
- All three models generate three incorrect predictions of false positives (Type I error)

Conclusions

- Logistic Regression is the recommended algorithm to predict the success of first stage rocket landing, due to:
  - good accuracy for this simple dataset
  - easier implementation, compare to other classification models, interpretation and very efficient to train
  - reasonable computational requirements

Appendix

- Data Collection – SpaceX API
  - Rocket launch data https://api.spacexdata.com/v4/launches/past

- Data Collection – Web Scraping from Wikipedia
  - Falcon 9 historical launch records

  https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches

Thank you!