IBM Data Science SpaceX Capstone Project

Nermeen Nabegh Ahmed

Sep 7th, 2021

OUTLINE

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

EXECUTIVE SUMMARY

- Business Problem
  - Space Rockets launch costing (based on stage 1 landing) challenge
- Data requirements
  - Historical launches data of landing success
- Methodology
  - Data collection
  - Data wrangling
  - EDA with data visualization
  - EDA with SQL
  - Folium interactive geographical map
  - Plotly Dash live dashboard
  - Stage 1 landing predictive classifier
- Results
  - EDA insights
  - Interactive map and dashboard demo
  - Predictive classifier outcome

INTRODUCTION

- Project Background:
  - SpaceX advertises Falcon 9 rocket launches at a cost of 62 million dollars versus 165 million dollars launches by other providers
  - The primary cost saving agent is the high success rate of stage 1 landing and thus its reusability in future launches
  - The challenge here is to set a right costing forecast of the rocket launches through predicting its potential to land stage 1 successfully

- Key questions to answer:
  - What are the most influential features of a rocket launch that control the probability of stage 1 landing successfully?
  - Given the compound effect of different features (such as site location, payload mass, etc.), how can we predict the potential of having a successful launch?
  - What conditions shall we recommend for a new rocket launch to increase its landing success probability?

METHODOLOGY

Link to Github Repository

- Data collection
  - SpaceX REST API
  - Wikipedia Web scrapping
- Data wrangling
- EDA
  - Data Viz (Correlation studies of different features and landing success)
  - SQL (Data population statistics understanding)
- Interactive visuals
  - Geographical relations study through Folium maps
  - Live Dashboard statistical analysis
- Best predictive classifier model building
  - Experiment usability and compatibility of SVM, Tree maps, KNN, Logistic Regression optimizing parameters
  - Select the best test data performing classifier

METHODOLOGY – DATA COLLECTION

[Link to Github Repository]

SpaceX REST API

Extract from API data

```
spacex_url="https://api.spacexdata.com/v4/launches/past"
```

```
response = requests.get(spacex_url)
```

Convert to .json static file

```
r = response.json()
data = pd.json_normalize(r)
```

Clean data for relevance in dict format

```
getBoosterVersion(data)
```

```
getLaunchSite(data)
```

```
# Call getPayloadData
getPayloadData(data)
```

```
# Call getCoreData
getCoreData(data)
```

```
launch_dict = {'FlightNumber': list(data['flight_number']),
'Date': list(data['date']),
'BoosterVersion':BoosterVersion,
'PayloadMass':PayloadMass,
'Orbit':Orbit,
'LaunchSite':LaunchSite,
'Outcome':Outcome,
'Flights':Flights,
'GridFins':GridFins,
'Reused':Reused,
'Legs':Legs,
'LandingPad':LandingPad,
'Block':Block,
'ReusedCount':ReusedCount,
'Serial':Serial,
'Longitude': Longitude,
'Latitude': Latitude}
```

```
data = data[data['BoosterVersion']!='Falcon 1']
```

Export .csv format for further analysis

```
df = pd.DataFrame(launch_dict)
```

```
data_falcon9.to_csv('dataset_part\_1.csv', index=False)
```

Wikipedia Page - Web scrapping

Extract from HTML

```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=102768692
```

```
response = requests.get(static_url)
```

Convert to beautiful soup parse-able format

```
soup = BeautifulSoup(response.text)
```

Clean data for relevance in dict format

```
html_tables=soup.find_all('table')
```

```
temp = first_launch_table.find_all('th')
for t in temp:
    n = extract_column_from_header(t)
    if( n is not None and len(n) > 0):
        column_names.append(n)
```

```
launch_dict= dict.fromkeys(column_names)

# Remove an irrelvant column
del launch_dict['Date and time ( )']

launch_dict['Flight No.'] = []
launch_dict['Launch site'] = []
launch_dict['Payload'] = []
launch_dict['Payload mass'] = []
launch_dict['Orbit'] = []
launch_dict['Customer'] = []
launch_dict['Launch outcome'] = []
# Added some new columns
launch_dict['Version Booster']=[]
launch_dict['Booster landing']=[]
launch_dict['Date']=[]
launch_dict['Time']=[]
```

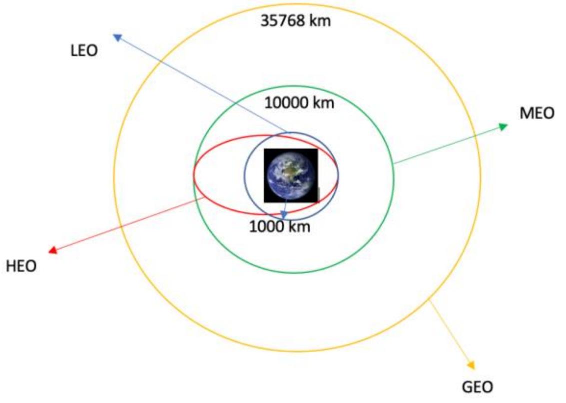
```
df = pd.DataFrame(launch_dict)
```

Export .csv format for further analysis

```
df.to_csv('spacex_web_scraped.csv', index=False)
```

METHODOLOGY – DATA WRANGLING

[Link to Github Repository]



For Data fairness check, basic grouping statistics, and ML data preparation:

1. Calculate the number of launches per site:

```
# Apply value_counts() on column LaunchSite
df.value_counts('LaunchSite')
```

```
LaunchSite
CCAFS SLC 40    55
KSC LC 39A      22
VAFB SLC 4E     13
dtype: int64
```

2. Calculate the # and occurrence of each orbit:

```
df.value_counts('Orbit')
```

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

3. Calculate the # and occurrence of outcome per orbit:

```
landing_outcomes=df.value_counts('Outcome')
landing_outcomes
```

```
Outcome
True ASDS      41
None None      19
True RTLS      14
False ASDS      6
True Ocean      5
False Ocean     2
None ASDS       2
False RTLS      1
dtype: int64
```

```
for i,outcome in enumerate(landing_outcomes.keys()):
    print(i,outcome)
```

```
0 True ASDS
1 None None
2 True RTLS
3 False ASDS
4 True Ocean
5 False Ocean
6 None ASDS
7 False RTLS
```

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

```
{'False ASDS', 'False Ocean', 'False RTLS', 'None ASDS', 'None None'}
```

4. Create a landing outcome class

```
landing_class = []
for x in df['Outcome']:
    #print(x)
    if x in bad_outcomes:
        landing_class.append(0)
        #print(0)
    else:
        landing_class.append(1)
        #print(1)
```

```
df['Class']=landing_class
```

METHODOLOGY – EDA with Visualization

Scatter Plots:

- Flight # vs site
- Flight # vs payload mass
- Payload vs orbit
- Flight # vs Orbit

These enable showing correlation between different features

Bar Chart:

- Success rate vs orbit

This shows a categorical comparison of diff. orbit types

Link to Github Repository

Line Graph:

- Success rate vs dates

This shows a progression and behavior over time proving a growth (improvement) or decaying trend

METHODOLOGY – EDA with SQL

SQL Queries run to perform exploratory statistical and categorical analysis of: Link to Github Repository

- The unique launch sites names in the space mission
- First 5 records sample of 'KSC' launch sites
- The total payload mass carried by boosters launched by NASA (CRS)
- Average payload mass carried by booster version F9 v1.1
- Dates of successful landing outcome in drone ship
- Names of the boosters with successful ground pad landing at payload mass > 4000 and <6000
- Total number of successful and failed mission outcomes
- The booster_versions names which have carried the maximum payload mass
- Records of month names, successful ground pad landing_outcomes, booster versions, launch_site for the year 2017
- Ranking the count of successful landing_outcomes from 2010-06-04 to 2017-03-20 descendingly

```
df['Class']=landing_class
```

METHODOLOGY – Folium map

[Link to Github Repository]

Visualize the launch data map:

- Utilizing the lat, long coordinates of each site
- Describing the site features wth a circular marker
- Segregating failed and successful launches with a Red and Green color coding respectively
- Exploring different geographical features impact on success vs failure potential such as the proximity to:
  (utilizing Haversine's formula for distance calculation)
  - Highways
  - Railways
  - Coastal areas
  - Urban regions

```
df['Class']=landing_class
```

METHODOLOGY – Plotly Dash

Link to Github Repository

Develop an interactive live dashboard providing insights through:

Pie chart of total launches for a selected site or the total sites collection

- Shows relative proportions of different sites successful landing distribution
- Shows % of success vs failure for a given site

Scatter Plot showing the correlation between Outcome and Payload Mass (Kg) for different Booster Versions with freedom of selection of the range of payload mass of interest

```
df['Class']=landing_class
```

METHODOLOGY – Predictive Classifier

Link to Github Repository

Classification model development:

- Split our data into training and test data sets
- Decide on machine learning algorithms to use
- Optimize our parameters for each algorithm by GridSearchCV
- Fit our training datasets into the GridSearchCV objects

Classification models evaluation:

- Check accuracy for each model
- Get tuned hyperparameters for each type of algorithms
- Plot Confusion Matrix

Improving the model:

- Feature Engineering
- Algorithm Tuning

Selecting the best model and parameters:

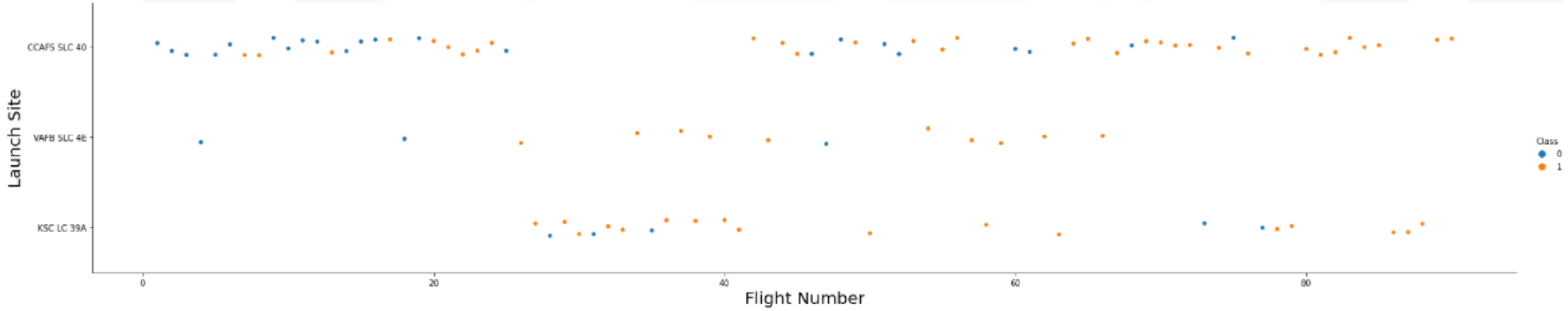
- The model with the best accuracy score is selected

```
df['Class']=landing_class
```

RESULTS – EDA with Visualization

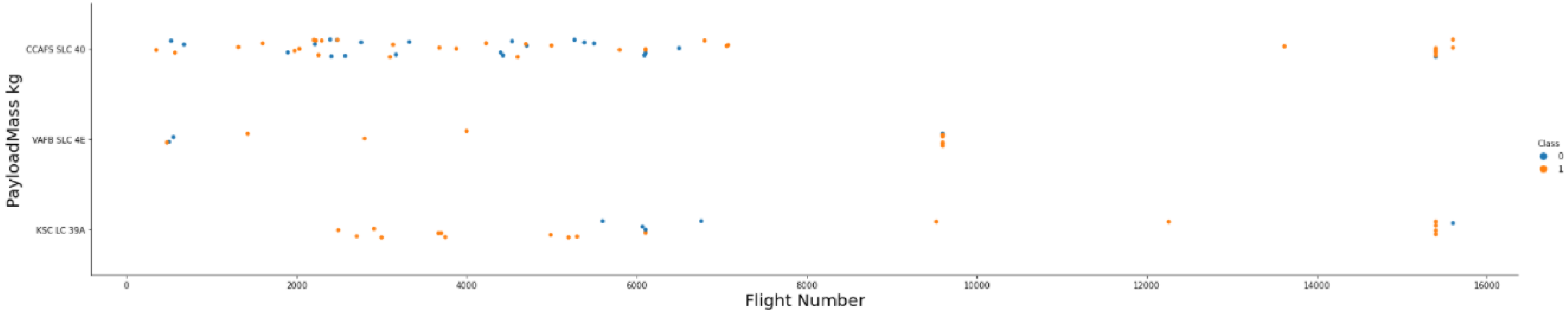
[Link to Github Repository]

Flight number vs Flight Site:



The more orange dots in each horizontal distribution the higher the success rate at a launch site

Payload Mass vs Flight number:



IBM Developer

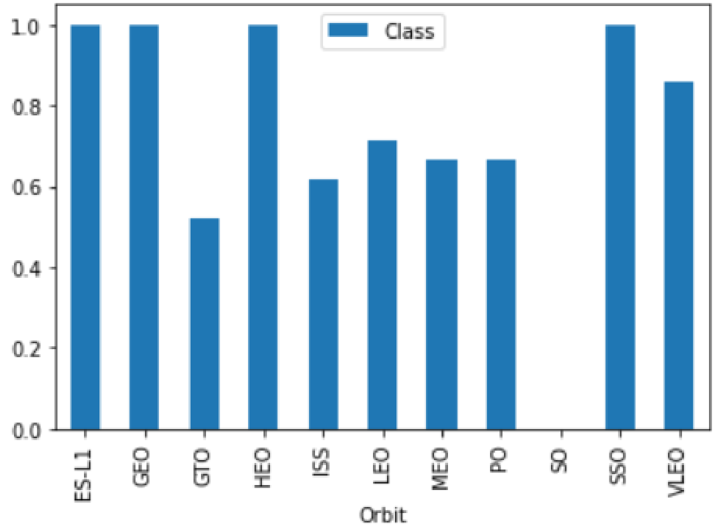
Looks like the higher the payload mass the greater the orange dots (success) representation vs total

RESULTS – EDA with Visualization

[Link to Github Repository]()

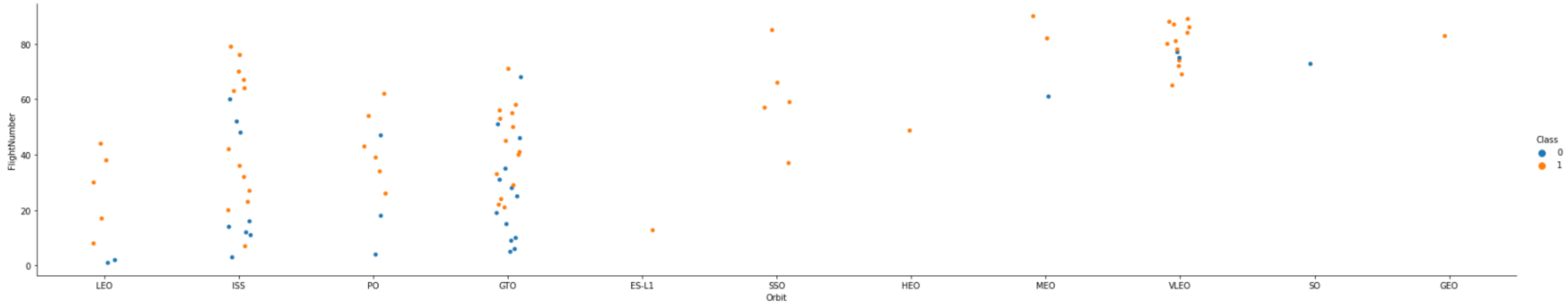
Success rate vs. Orbit type:

Orbit ES-L1, GEO,HEO,SSO, have the highest success Rate



Flight Number vs. Orbit type:

Generally but specifically for LEO orbit success seems related to high # of flights except for GTO orbit (For SO the sample is not representative)

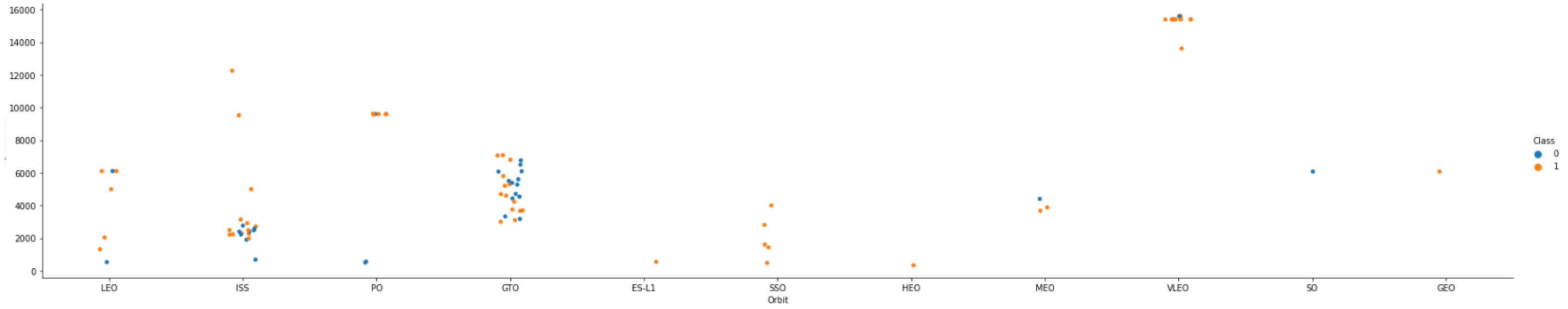


RESULTS – EDA with Visualization

Link to Github Repository

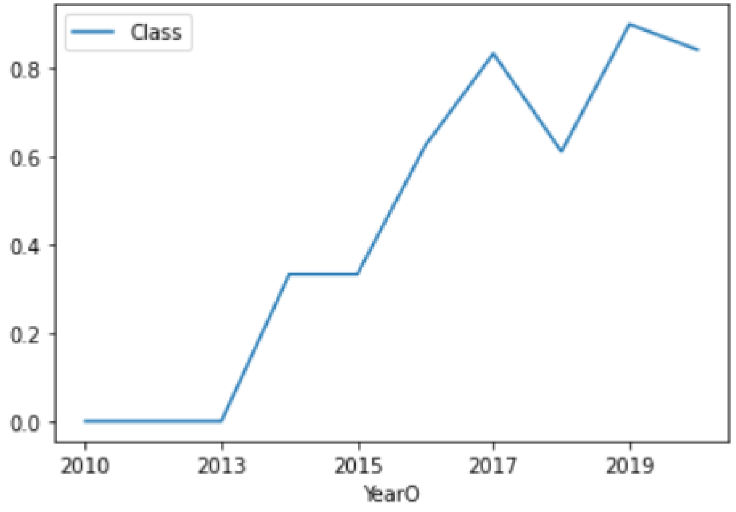
Payload vs. Orbit type

Heavy payload mass seem inconclusive for in terms of success for LEO,GTO yet imperfectly positive for ISS, PO and negative for MEO, VLEO



Launch success yearly trend

Success rate improves over time specifically since 2013



RESULTS – EDA with SQL

[Link to Github Repository]

Unique Launch sites

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

Records of ‘KSC’ Launch sites

DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__outcome
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
2014-04-18	19:25:00	F9 v1.1	CCAFS LC-40	SpaceX CRS-3	2296	LEO (ISS)	NASA (CRS)	Success	Controlled (ocean)
2014-07-14	15:15:00	F9 v1.1	CCAFS LC-40	OG2 Mission 1 6 Orbcomm-OG2 satellites	1316	LEO	Orbcomm	Success	Controlled (ocean)
2014-09-21	5:52:00	F9 v1.1 B1010	CCAFS LC-40	SpaceX CRS-4	2216	LEO (ISS)	NASA (CRS)	Success	Uncontrolled (ocean)
2015-04-14	20:10:00	F9 v1.1 B1015	CCAFS LC-40	SpaceX CRS-6	1898	LEO (ISS)	NASA (CRS)	Success	Failure (drone ship)

Total payload of NASA (CRS) boosters

sum_payyload_kg
23589

Avg. payload of F9 v1.1 booster

avg_payload_kg
1806

First date of successful drone ship landing

DATE
2012-05-22

RESULTS – EDA with SQL

Link to Github Repository

Boosters with ground success & 4000<payload<6000

booster_version
F9 FT B1021.2
F9 FT B1026

# of successful and failed landings

mission_outcome	2
Failure (in flight)	1
Success	55

Booster versions w/ max. payload mass carried

booster_version
F9 B5 B1048.5
F9 B5 B1049.7
F9 B5 B1051.3
F9 B5 B1051.4
F9 B5 B1051.6
F9 B5 B1056.4
F9 B5 B1060.3

Records of ground successes in 2017

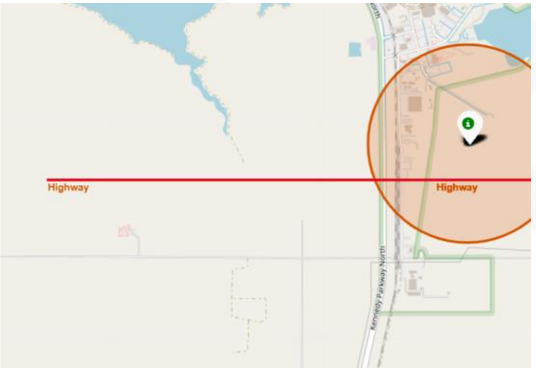
landing__outcome	booster_version	launch_site
Precluded (drone ship)	F9 v1.1 B1018	CCAFS LC-40

# of success. landing from 4/6/2010 to 20/3/2017

landing__outcome	2
Failure (drone ship)	3
No attempt	3
Success (drone ship)	3
Success (ground pad)	3
Controlled (ocean)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

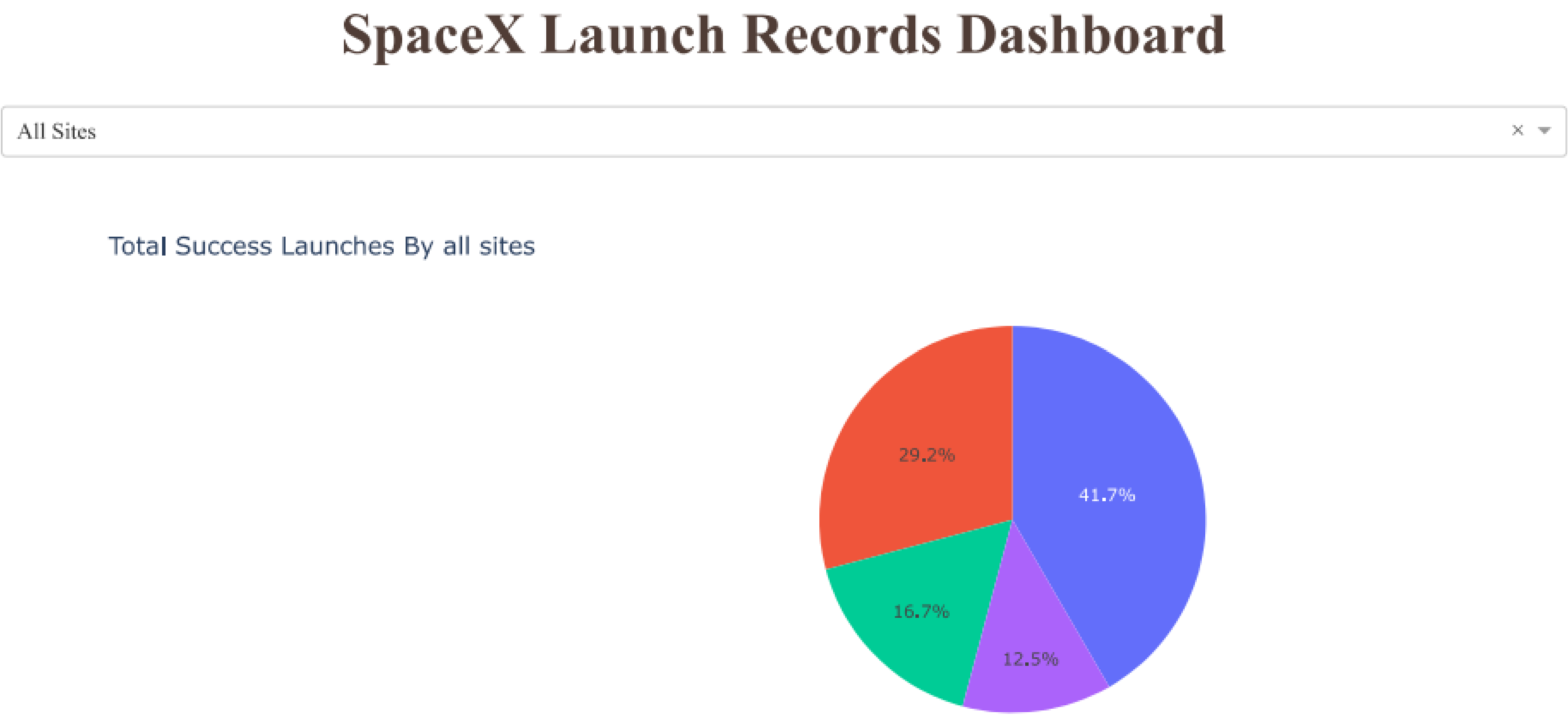
RESULTS – Folium map

Link to Github Repository



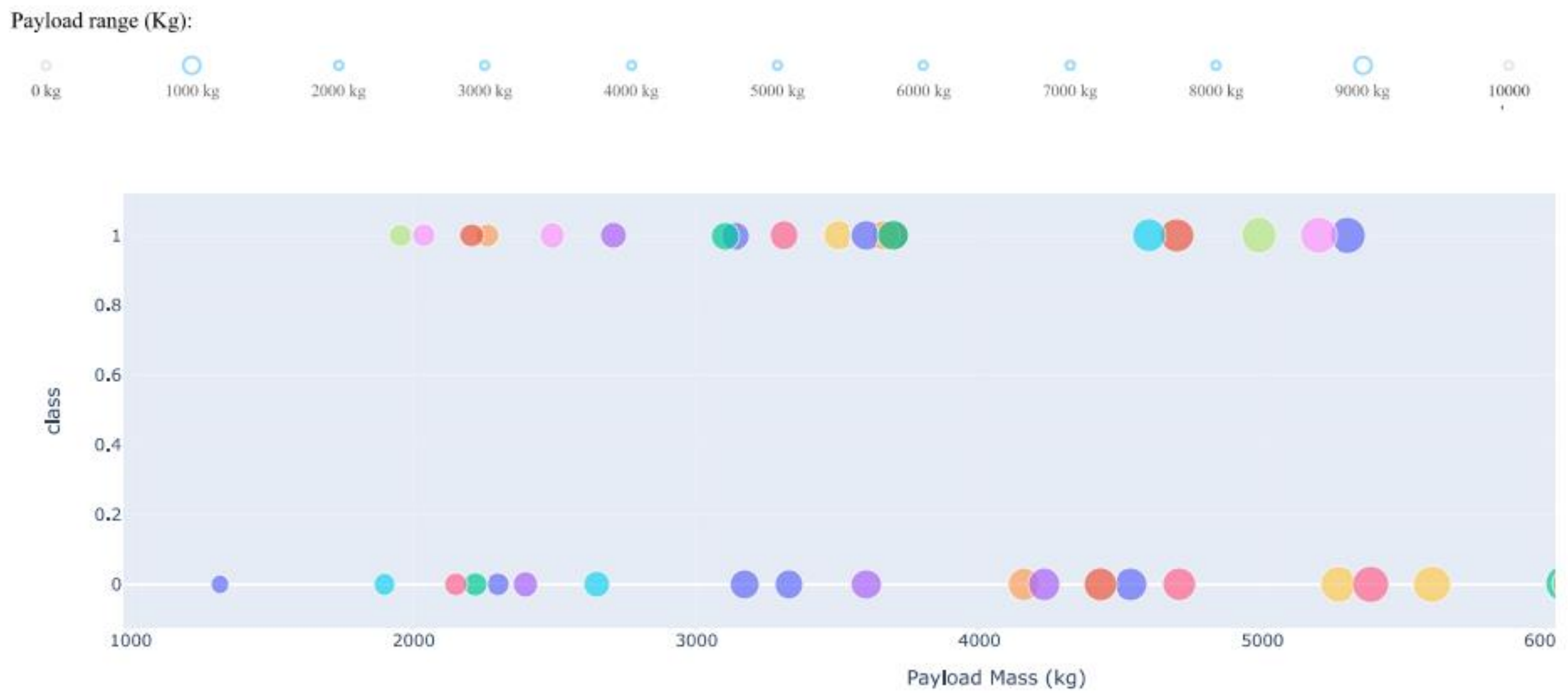
RESULTS – Plotly Dashboard

[Link to Github Repository]



RESULTS – Plotly Dashboard

[Link to Github Repository]

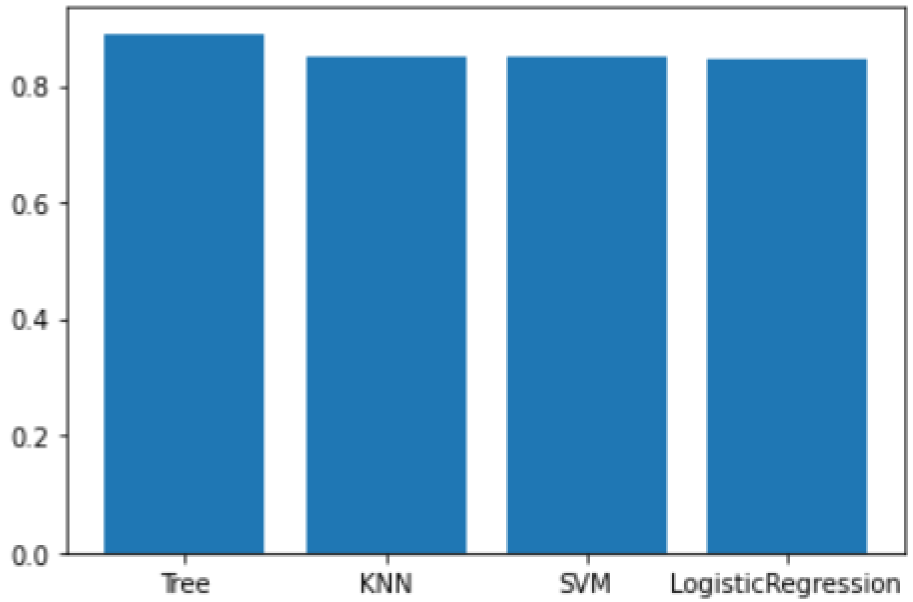


RESULTS – ML Predictive Classification

[Link to Github Repository]

Best Classifier:

Tree algorithm performing slightly better than the rest when tested and refined to parameters:




```
bestalgorithm = 'KNN'
bestalgorithm_score = knn_cv.best_score_
tree_score = tree_cv.best_score_
logreg_score = logreg_cv.best_score_
svm_score = svm_cv.best_score_
if tree_score > bestalgorithm_score:
    bestalgorithm = 'Tree'
    bestalgorithm_score = tree_score
if logreg_score > bestalgorithm_score:
    bestalgorithm = 'LogisticRegression'
    bestalgorithm_score = logreg_score
if svm_score > bestalgorithm_score:
    bestalgorithm = 'SVM'
    bestalgorithm_score = svm_score
print('Best Algorithm is',bestalgorithm,'with a score of',bestalgorithm_score)
if bestalgorithm == 'Tree':
    print('Best Params is :',tree_cv.best_params_)
if bestalgorithm == 'KNN':
    print('Best Params is :',knn_cv.best_params_)
if bestalgorithm == 'LogisticRegression':
    print('Best Params is :',logreg_cv.best_params_)
if bestalgorithm == 'SVM':
    print('Best Params is :',svm_cv.best_params_)
```

```
Best Algorithm is Tree with a score of 0.8892857142857142
Best Params is : {'criterion': 'gini', 'max_depth': 6, 'max_features': 'auto', 'min_samples_leaf': 1, 'min_samples_split': 5, 'splitter': 'random'}
```

DISCUSSION

Link to Github Repository

CONCLUSION

[Link to Github Repository]

- The Tree Classifier Algorithm is the best for
- The success rates for SpaceX launches increases over time (+ve learning curve)
- KSC LC-39A had the highest # of successful launches from all the sites
- Orbit ES-L1, GEO,HEO,SSO have the best success rate

GRADING SYSTEM (for Peer Graded Assignment)

In the next exercise, you can find a provided PowerPoint template to help you get started. However, you are free to add additional slides, charts, and tables. There are a total of 40 points possible for the final assessment, and you will be graded by your peers, who are also completing this assignment. The main grading criteria will be:

- Uploaded the URL of your GitHub repository including all the completed notebooks and Python files (1 pt) – Done
- Uploaded your completed presentation in PDF format (1 pt) – Done
- Completed the required Executive Summary slide (1 pt) – Done
- Completed the required Introduction slide (1 pt) – Done
- Completed the required data collection and data wrangling methodology related slides (1 pt) – Done
- Completed the required EDA and interactive visual analytics methodology related slides (3 pts)
- Completed the required predictive analysis methodology related slides (1 pt)
- Completed the required EDA with visualization results slides (6 pts)
- Completed the required EDA with SQL results slides (10 pts)
- Completed the required interactive map with Folium results slides (3 pts)
- Completed the required Plotly Dash dashboard results slides (3 pts)
- Completed the required predictive analysis (classification) results slides (6 pts)
- Completed the required Conclusion slide (1 pts)
- Applied your creativity to improve the presentation beyond the template (1 pts)
- Displayed any innovative insights (1 pts)