IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Billy Koech
27th March 2033

Outline

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

Executive Summary

SpaceX’s success is mainly attributed to it’s ability to reuse first stage parts of it’s rockets. Some of it’s attempts to land the first stage have been both a success and a failure. In this project we use machine learning to try and determine whether the success of landing can be predicted using rocket launch conditions and some other features.

The data for this project was collected through the SpaceX API and by scraping Wikipedia. The data was cleaned and features to be used in the models selected.

Classification models were trained and the best model had an accuracy of 0.833. The models did a good job in predicting rockets that successfully landed and a bad job in predicting rockets that failed to land.

Introduction

SpaceX has gained worldwide attention for a series of historic milestones. It is the only private company ever to return a spacecraft from low-earth orbit, which it first accomplished in December 2010.

SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars whereas other providers cost upward of 165 million dollars each, much of the savings is because Space X 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.

Section 1

Methodology

Methodology

Methodology Outline

- Data collection using:
  - SpaceX API
  - Web Scrapping
- Data wrangling
- Exploratory data analysis (EDA) using visualization and SQL
- Interactive visual analytics using:
  - Folium
  - Plotly Dash
- Predictive analysis using classification models

Data Collection

The data for the project was collected using two main methods:

1. Using the spaceX API. The rocket launch data from SpaceX was requested from the following url https://api.spacexdata.com/v4/launches/past

2. By performing web scraping to collect Falcon 9 history records from a Wikipedia page titled *List of Falcon 9 and Falcon Heavy Launches.* https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches

Data Collection – SpaceX API

Request made to collect rocket launch data.

requests.get(="https://api.spacexdata.com/v4/launches/past")

Jupyter Notebook

SpaceX rocket launch data.

Response received having the rocket launch data.

response.status_code()

--> 200

- GitHub link:

https://github.com/billykoech137/spacex_falcon9_project/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

Data Collection - Scraping

Wikipedia:
List of Falcon 9 and Falcon Heavy Launches

Using Beautiful Soup to extract the relevant tables form the Wikipedia page:

```
response = requests.get(static_url).text
soup = BeautifulSoup(response)
soup.find_all('table',"wikitable plainrowheaders collapsible")
df=pd.DataFrame(launch_data)
```

Structured Data in a pandas Data Frame

GitHub link:

https://github.com/billykoech137/spacex_falcon9_project/blob/main/jupyter-labs-webscraping.ipynb

Data Wrangling

- The following were performed:
  - Calculating the total launches on each site.
  - Calculating the number of occurrence of each orbit
  - Calculating the number of occurrence of mission outcome per orbit type
  - Creating landing outcome label from the outcome column

GitHub URL:

https://github.com/billykoech137/spacex_falcon9_project/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

- The following charts were plotted.
  - Scatter plots: Used to show relationship between continuous variables.
  - Categorical plots: Used to show relationship between categorical and continuous variables.
  - Bar Chats: Used to show representation of categorical variables
  - Line graphs : Used to show trend line.

- GitHub URL:

https://github.com/billykoech137/spacex_falcon9_project/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with SQL

- The following SQL queries were performed
  - Displayed the names of the unique launch sites in the space mission
  - Displayed 5 records where launch sites begin with the string 'CCA'
  - Displayed the total payload mass carried by boosters launched by NASA (CRS)
  - Displayed average payload mass carried by booster version F9 v1.1
  - Displayed a list of the dates when the first successful landing outcome in ground pad was achieved.
  - Displayed a list of the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
  - Displayed a list of the total number of successful and failure mission outcomes
  - Displayed a list the names of the booster versions which have carried the maximum payload mass. Use a subquery
  - Displayed a list of the records which will display the month names, failure landing outcomes in drone ship ,booster versions, launch site for the months in year 2015.
  - Ranked the count of successful landing outcomes between the date 04-06-2010 and 20-03-2017 in descending order.

GitHub URL:

https://github.com/billykoech137/spacex_falcon9_project/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- The following map objects were used:
  - Markers: Used to specific location.
  - Circles: used to add highlighted circle area with a text label on a specific launch site coordinates.
  - Lines: draw a line from the different site location to the shoreline.
  - MarkerCluster: used to add many markers having similar coordinates.

GitHub URL:

https://github.com/billykoech137/spacex_falcon9_project/blob/main/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

- The following plots/graphs and interactions were added to a dashboard:
  - A scatter plot showing the correlation between payload mass and success
  - A pie chart showing Total Success Launches By Site

- GitHub URL:
- https://github.com/billykoech137/spacex_falcon9_project/blob/main/Dashboard/spacex_dash_app.py

Predictive Analysis (Classification)

- The following steps were taking during the predictive analysis using classification models:
  - Features X and target Y were determined.
  - The data was then split 80-20. 80% for training and 20% for validation.
  - The following models were then trained on the data and the best performing model was selected.
    - Logistic Regression
    - Support Vector Machine
    - K Nearest Neighbor
    - Decision Tree Classifier

- GitHub URL:

https://github.com/billykoech137/spacex_falcon9_project/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

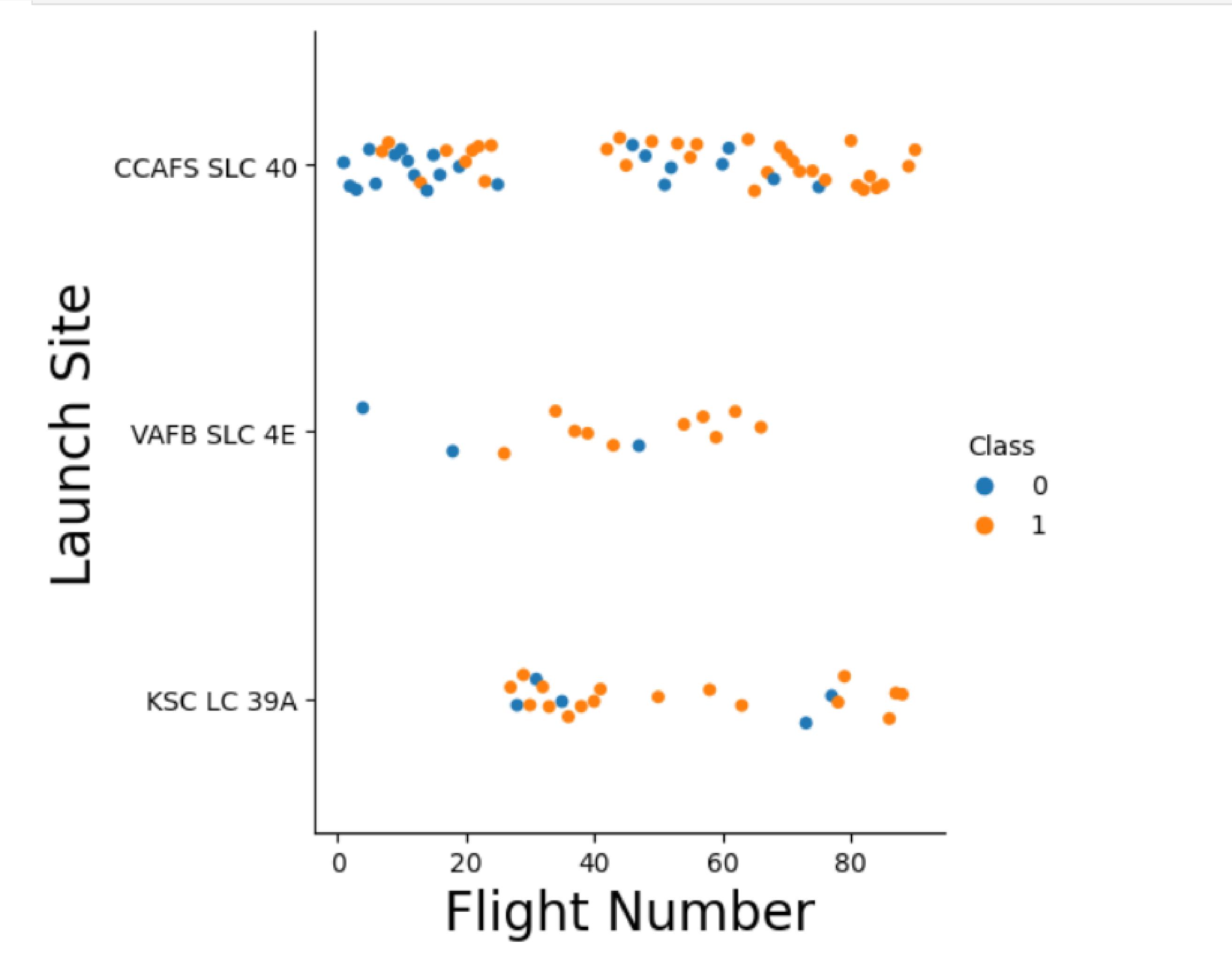
Results

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

Section 2

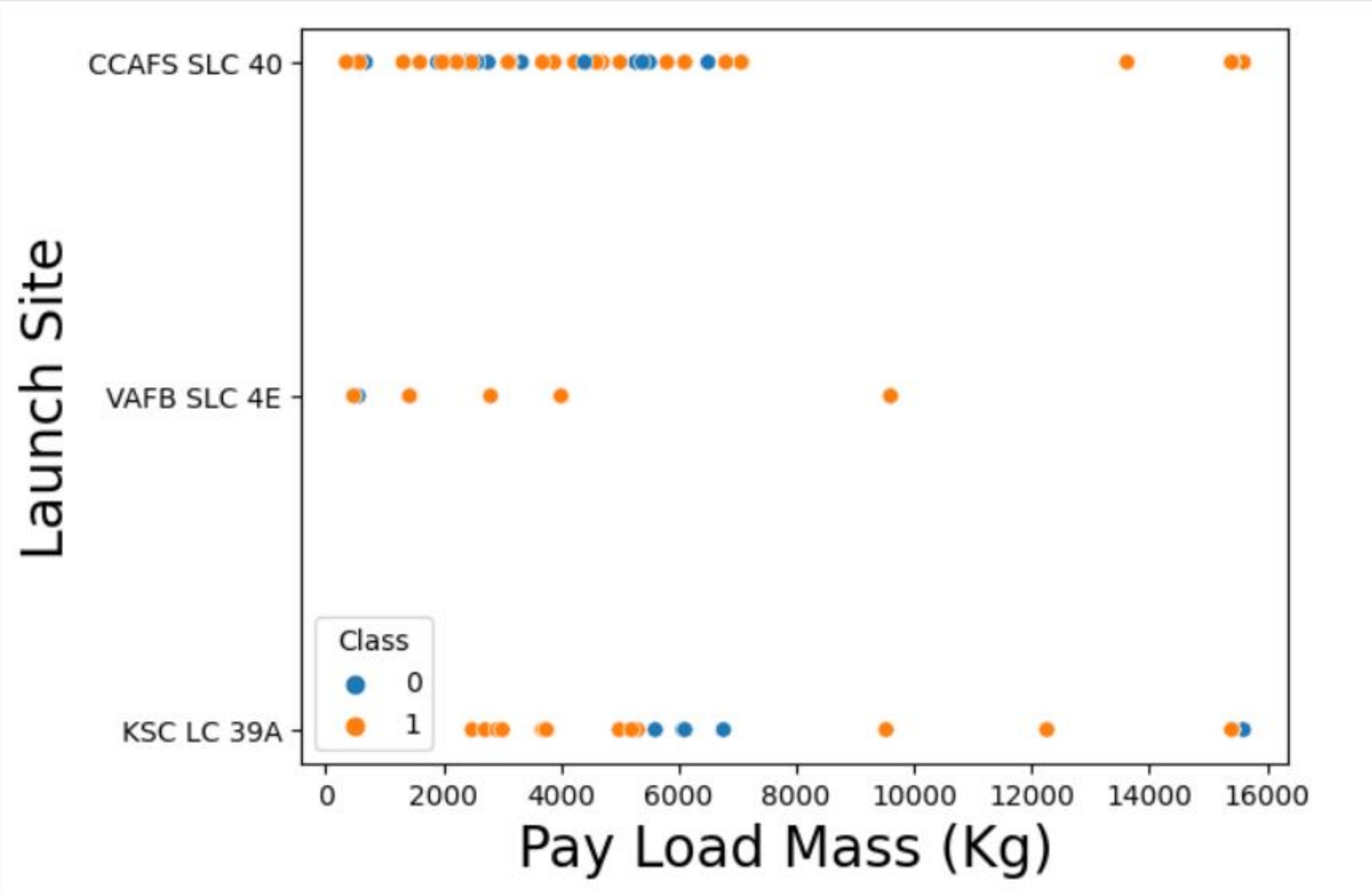
Insights drawn from EDA

Flight Number vs. Launch Site



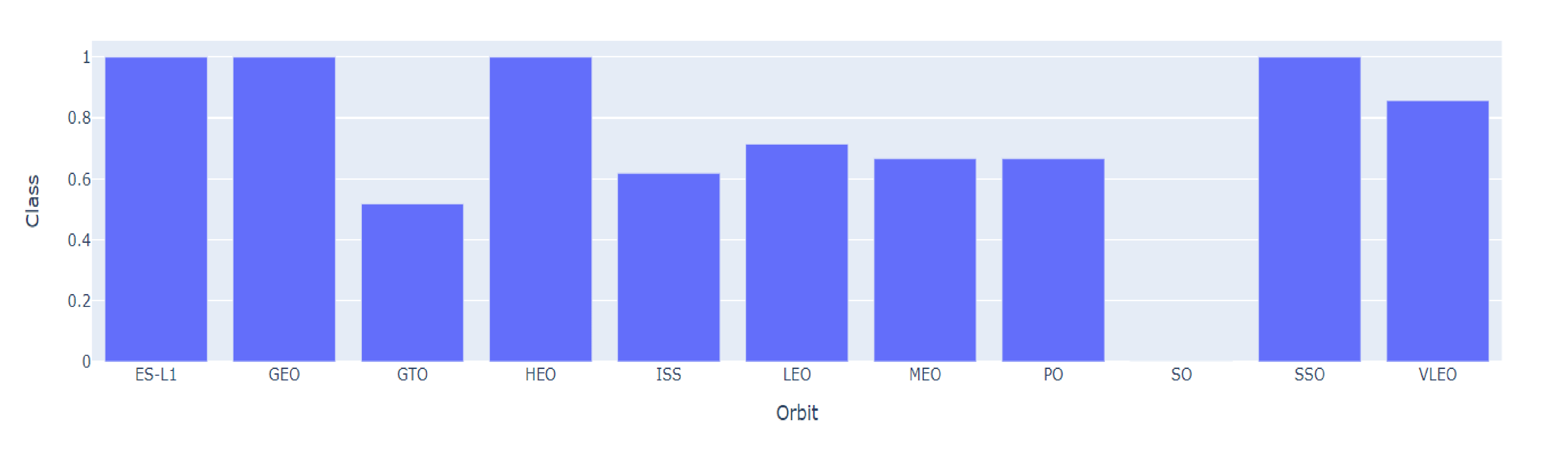
VAFB SLC has the least number of flights while CCAFS has the highest number of flights.

Payload vs. Launch Site



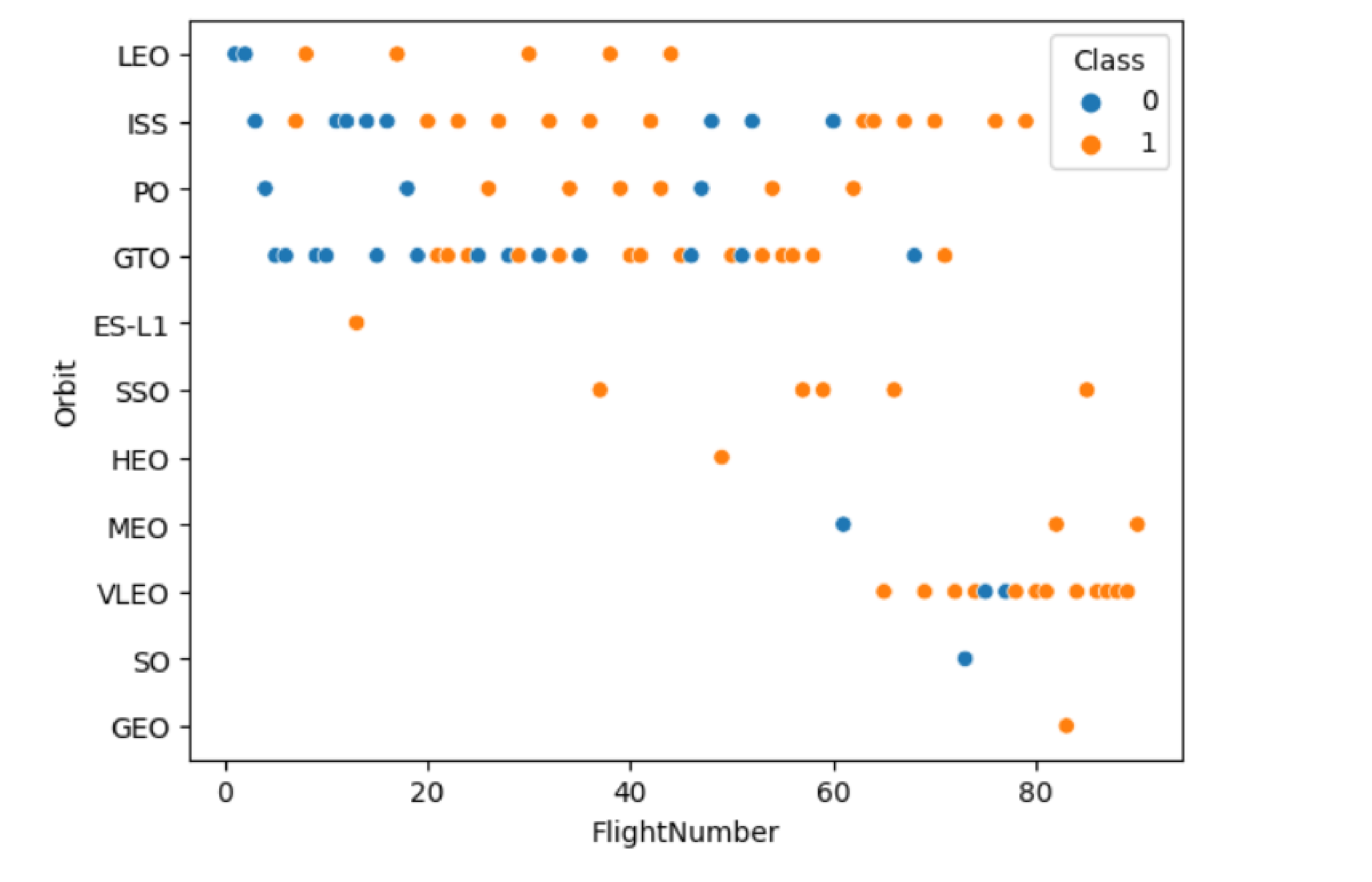
- For the VAFB-SLC launch site there are no rockets launched for heavy payload mass(greater than 10000)

Success Rate vs. Orbit Type



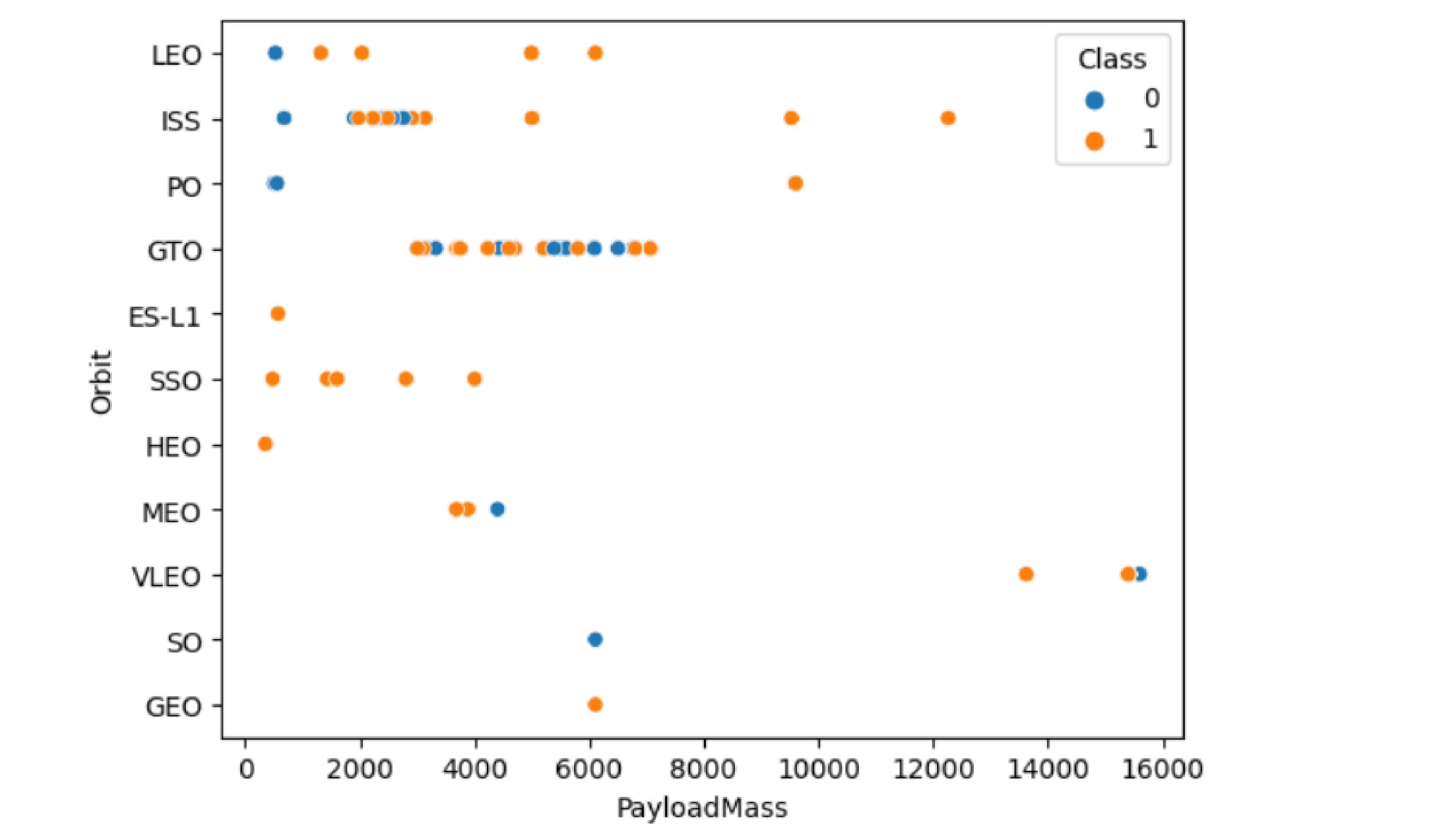
- ES-L1, GEO, LEO, SSO have the highest success rates.
- SO has the lowest success rate.

Flight Number vs. Orbit Type



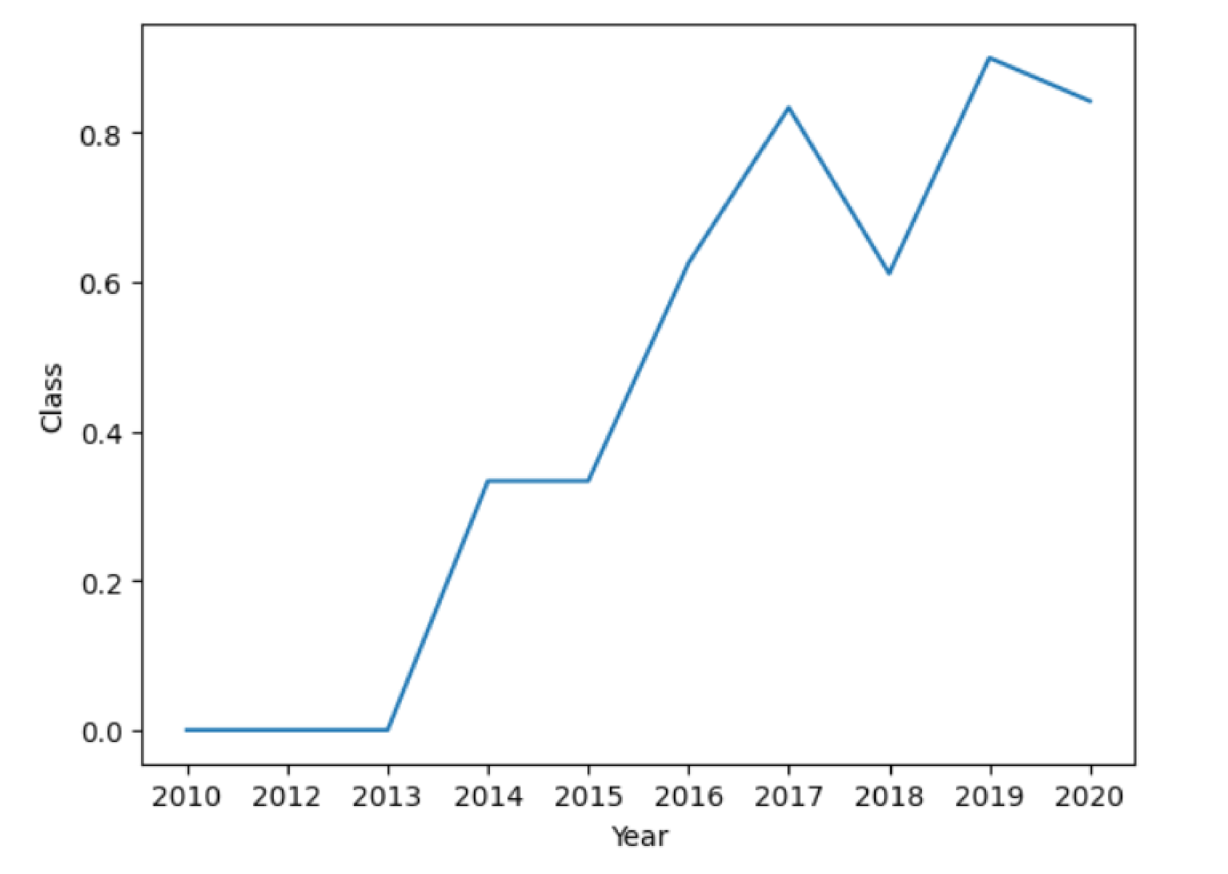
- For the LEO orbit the Success increases with number of flights.
- There is no relation between flight number and success for the other orbits.

Payload vs. Orbit Type



- With heavy payloads the successful landing rate are more for LEO, ISS and Polar.
- For GTO, there is no relationship between payload mass and success rate.

Launch Success Yearly Trend



- The since 2013 kept increasing till 2020

All Launch Site Names

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

- CCAFS LC-40: Cape Canaveral Air Force Station Launch Complex 40
- VAFB SLC-4E: Vandenberg Space Force Base Space Launch Complex 4E
- KSC LC-39A: Kennedy Space Center Launch Complex 39A
- CCAFS SLC-40: Cape Canaveral Air Force Station Space Launch Complex 40

Launch Site Names Begin with 'CCA'

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

- These are launches that happened at Cape Canaveral Air Force Station

Total Payload Mass

Total_Payload
45596

- Total payload carried by boosters from NASA

Average Payload Mass by F9 v1.1

Average_payload
2928.4

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

First Successful Ground Landing Date

Date
01-05-2017
03-06-2017
07-09-2017
08-01-2018
14-08-2017
15-12-2017
18-07-2016
19-02-2017
22-12-2015

- Dates of the first successful landing outcome on ground pad

Successful Drone Ship Landing with Payload between 4000 and 6000

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

- Names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000

Total Number of Successful and Failure Mission Outcomes

Sucess	Failure
100	1

- Total number of successful and failure mission outcomes

Boosters Carried Maximum Payload

Booster_Version
F9 B5 B1048.4
F9 B5 B1049.4
F9 B5 B1051.3
F9 B5 B1056.4
F9 B5 B1048.5
F9 B5 B1051.4
F9 B5 B1049.5
F9 B5 B1060.2
F9 B5 B1058.3
F9 B5 B1051.6
F9 B5 B1060.3
F9 B5 B1049.7

- Names of the booster which have carried the maximum payload mass

2015 Launch Records

Month	Landing _Outcome	Booster_Version	Launch_Site
01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

- Failed landing outcomes in drone ship, their booster versions, and launch site names for in year 2015

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

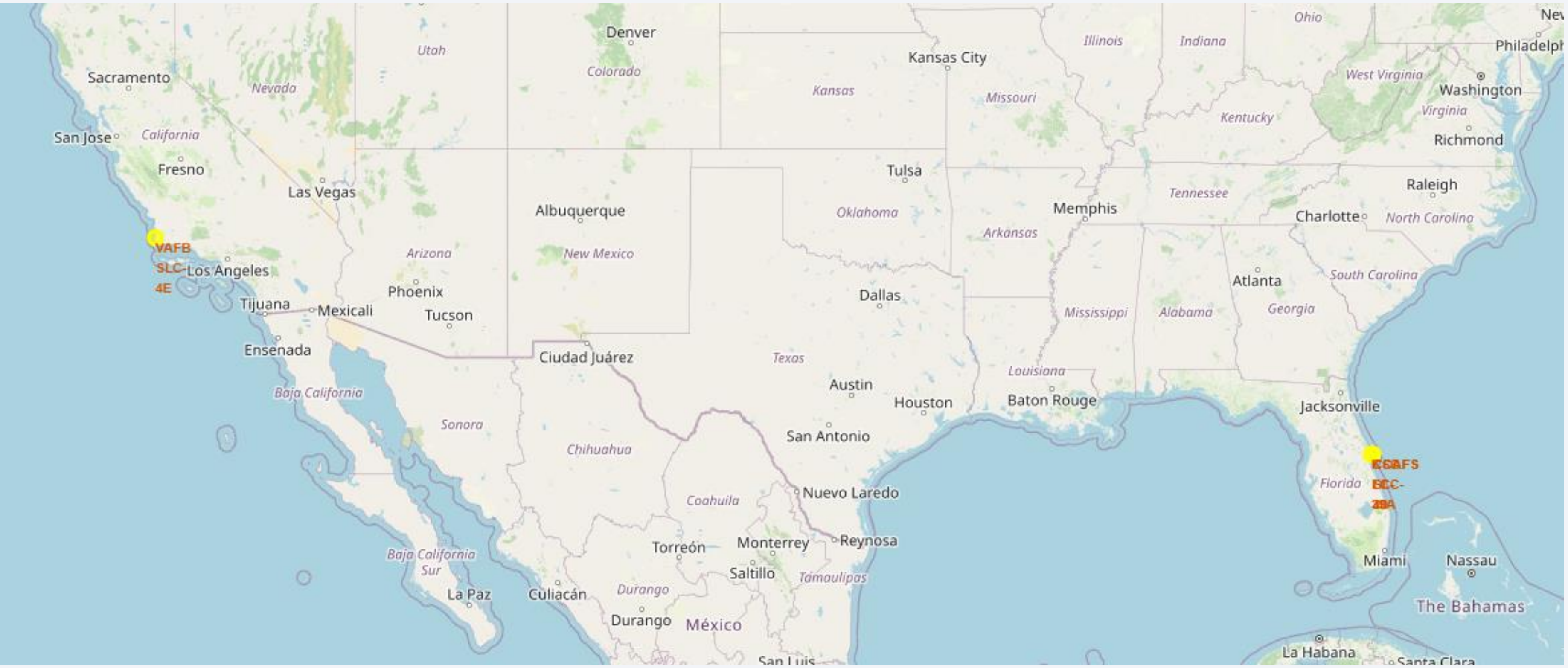
name	value
success	20
success_drone_ship	8
success_ground_pad	6
failure_drone_ship	4
failure	3
controlled_ocean	3
failure_parachute	1
uncontrolled_ocean	0
precluded_drone_ship	0

- Count ranking of landing outcomes between the date 2010-06-04 and 2017-03-20, in descending order

Section 3

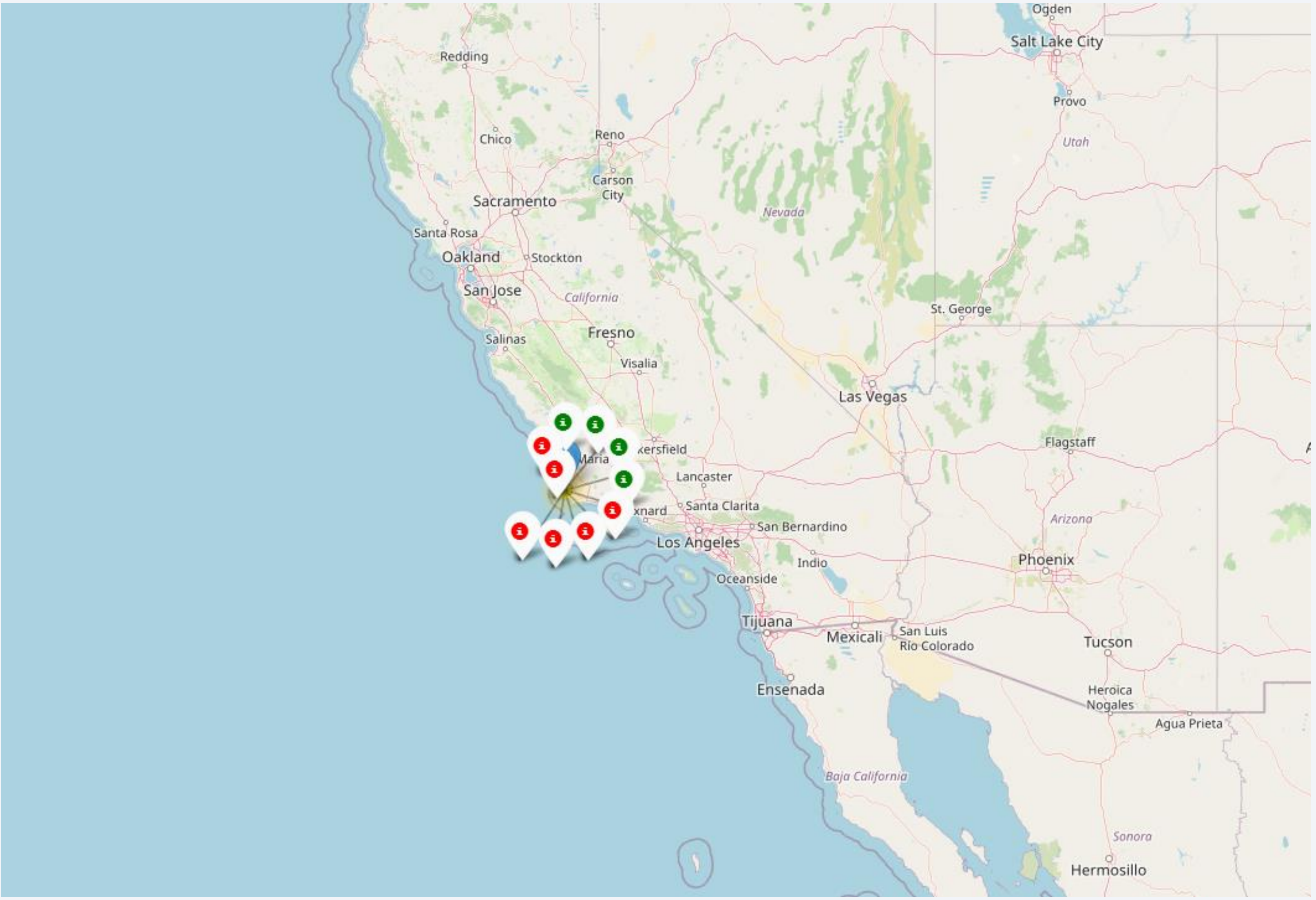
Launch Sites Proximities Analysis

Map Showing Locations of the Launch Sites



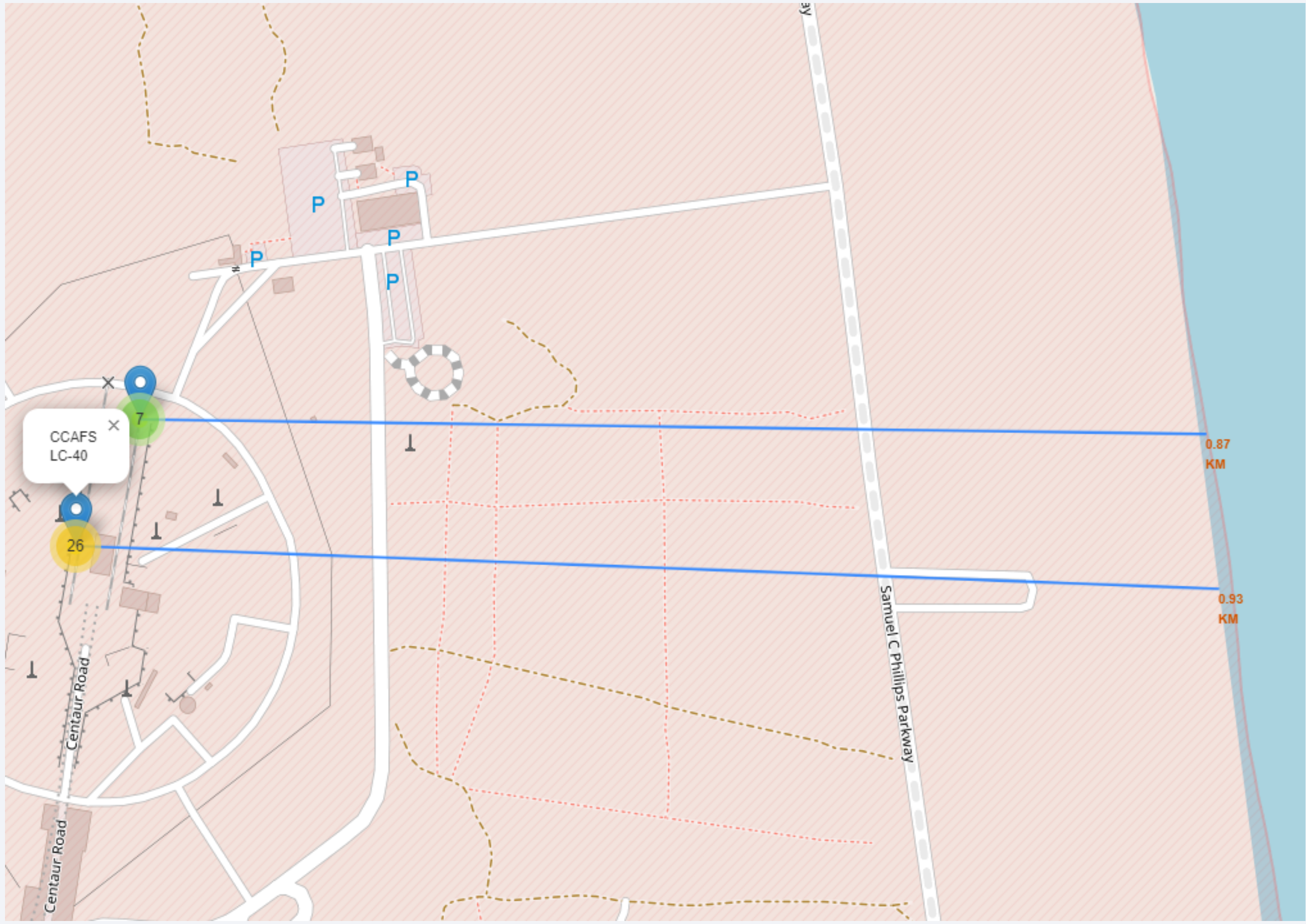
- Launch sites’ location markers on a global map
- Shows the different coordinates of the launch site locations on the global map.

Color-labeled Launch Outcomes



- Color-labeled launch outcomes on the map

Distance to the Coastline

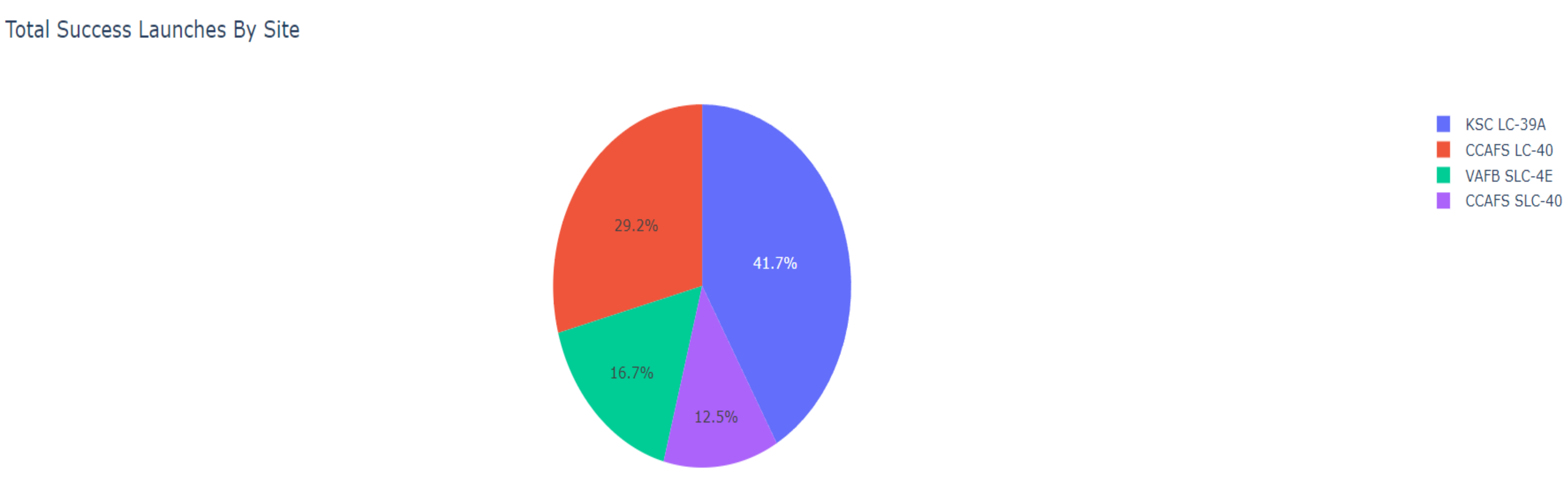


- Distance of Cape Canaveral Air Force Stations to the coastline.

Section 4

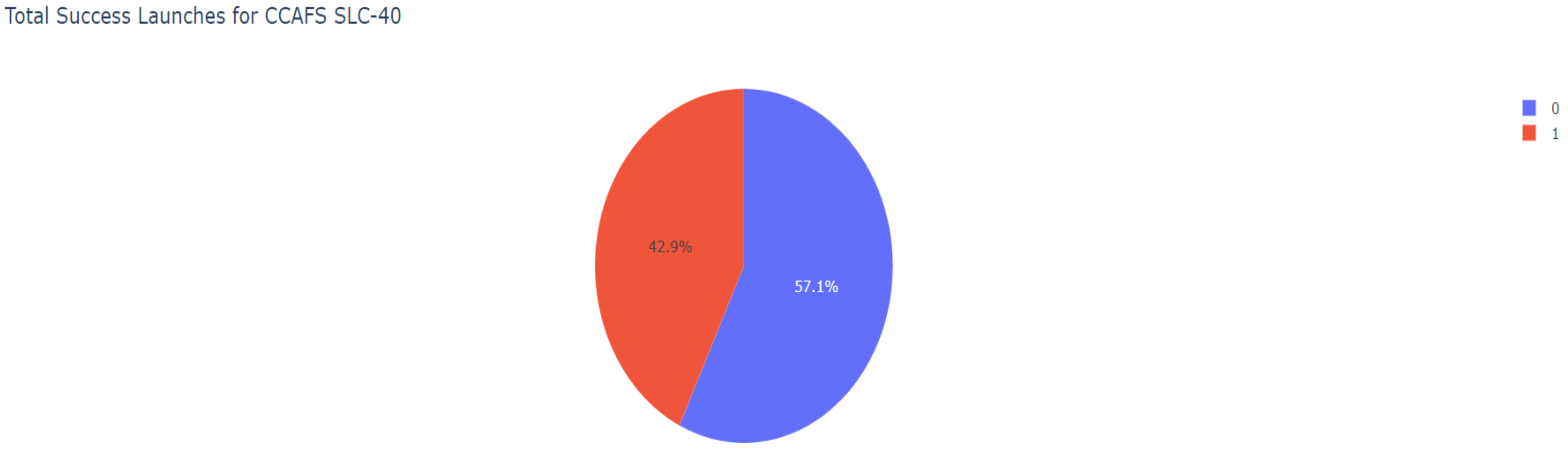
Build a Dashboard with Plotly Dash

Launch Success Count for all Sites



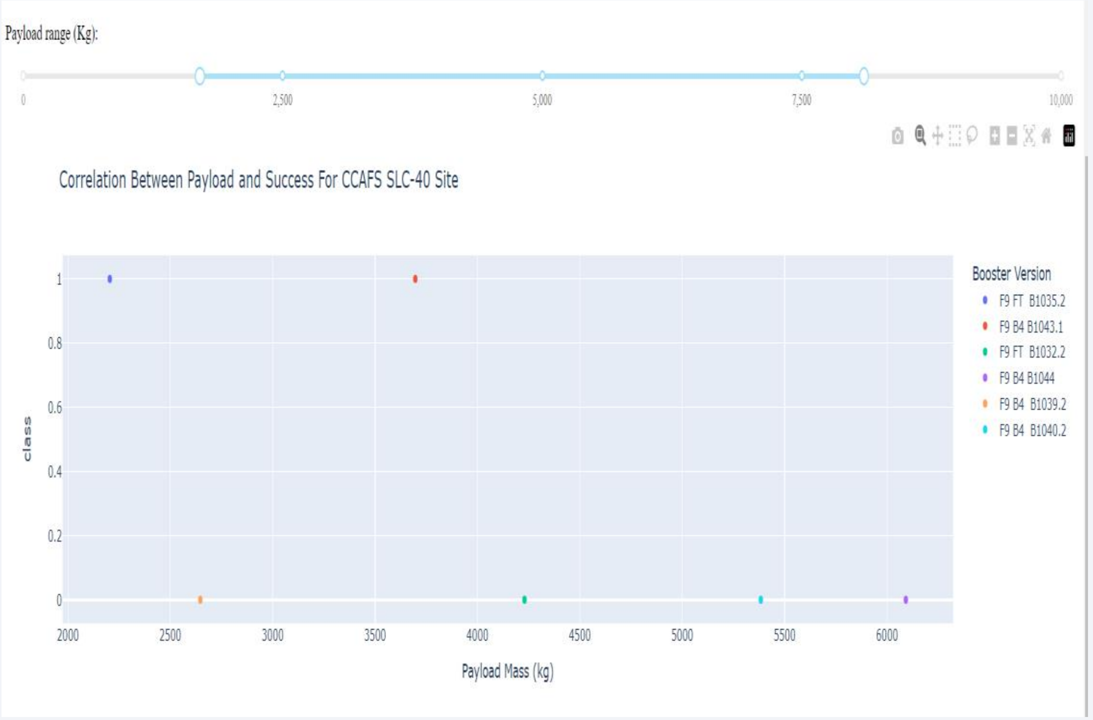
Pie chart showing launch success count for all sites.

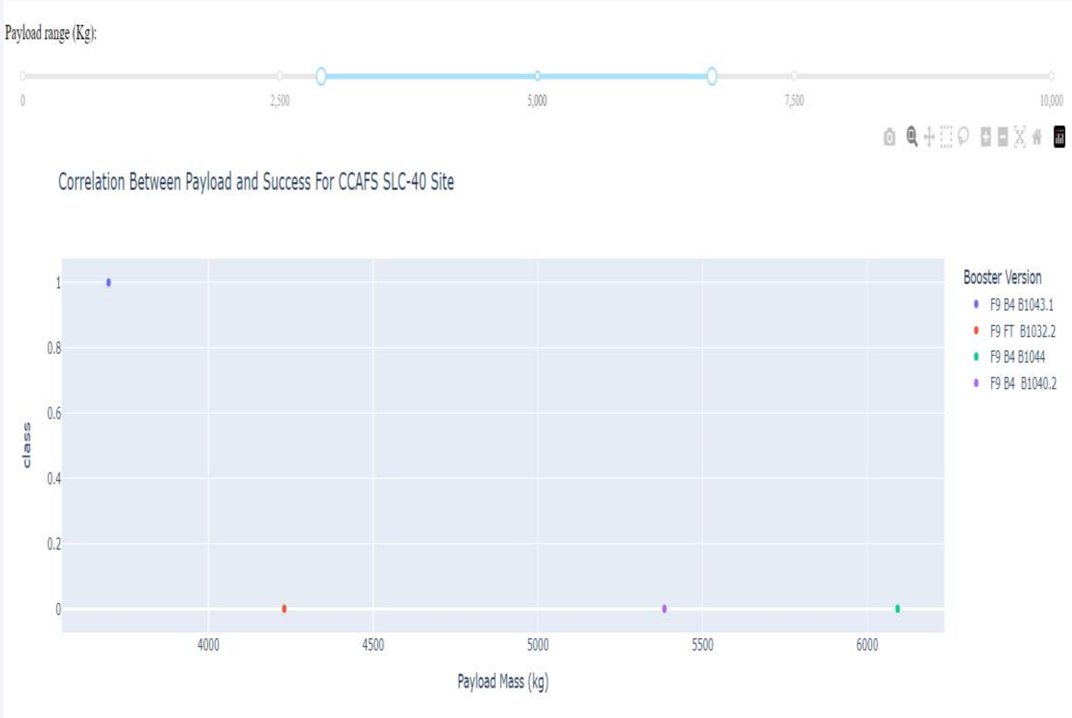
Pie chart Showing Success Launches



Pie chart showing the success ratio for the Cape Canaveral Air Force Station Space Launch Complex 40

Payload Vs. Launch Outcome



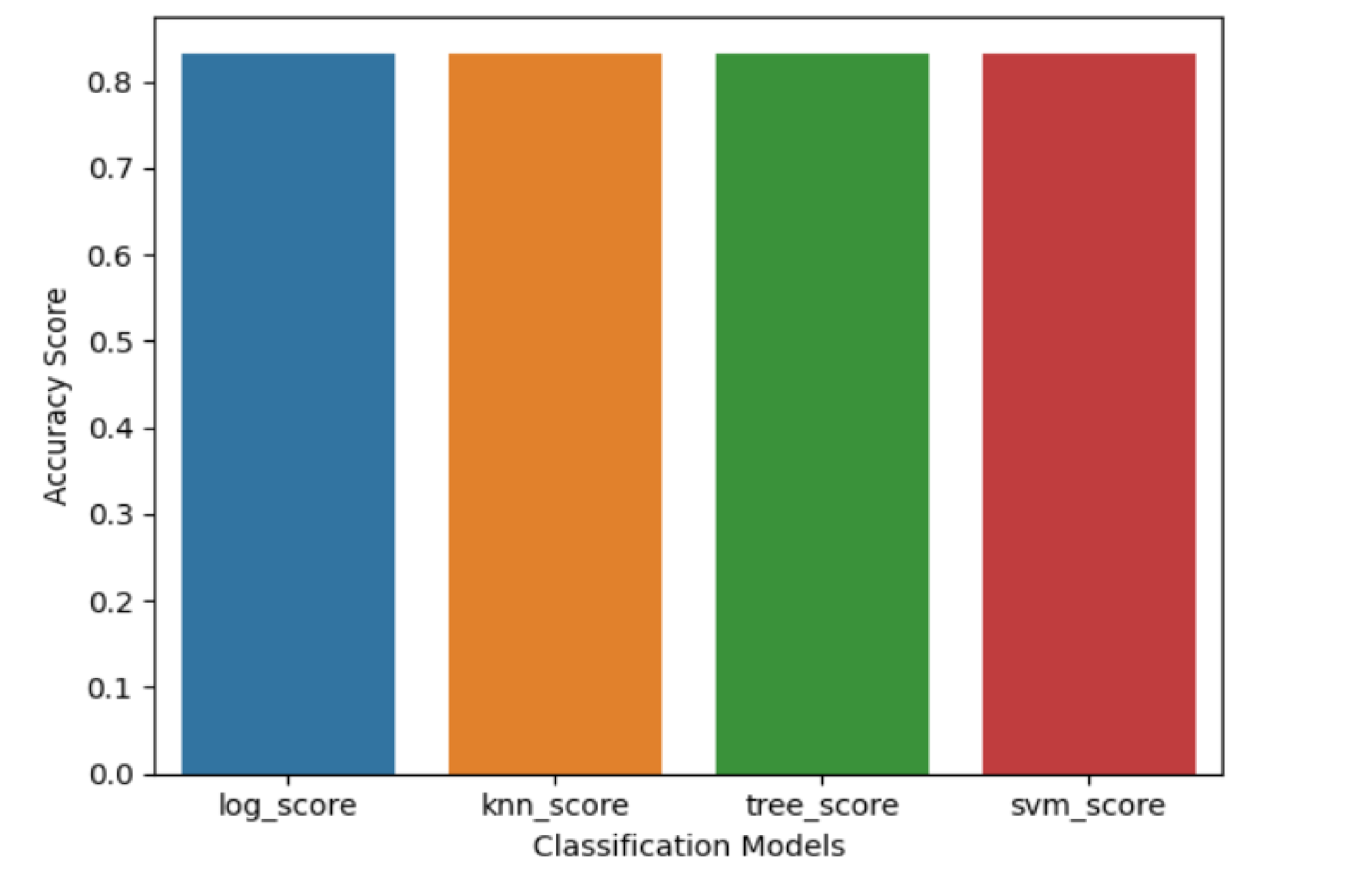


Payload vs. Launch Outcome scatter plot for all sites, with different payload selected in the range slider

Section 5

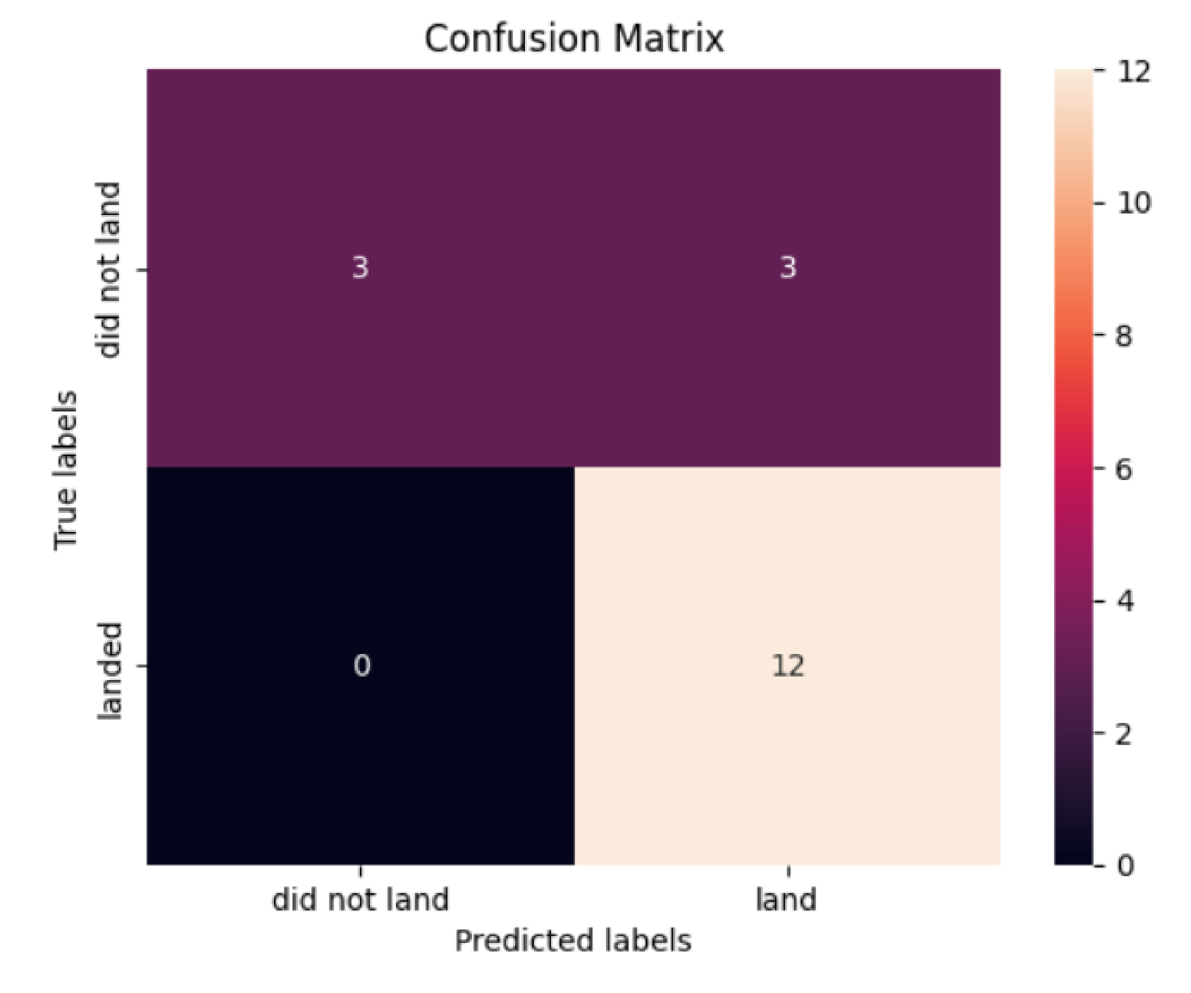
Predictive Analysis (Classification)

Classification Accuracy



All the models have the same classification accuracy of 83.33%

Confusion Matrix



- True Positives: 12
- False Positives: 0

- True Negatives: 3
- False Negative: 3

- The models did a good job predicting rockets that landed.

Conclusions

- The classification models did well predicting the rockets that landed. More tuning of the models should be done to improve it’s accuracy of predicting rockets that did not land.

Appendix

```python
parameters ={'C':[0.01,0.1,1],
             'penalty':['l2'],
             'solver':['lbfgs']}
```

```python
parameters ={"C":[0.01,0.1,1],'penalty':['l2'], 'solver':['lbfgs']}# l1 lasso l2 ridge
lr=LogisticRegression()

logreg_cv = GridSearchCV(estimator=lr, param_grid=parameters, cv=10)
logreg_cv.fit(X_train, Y_train)
```

```
GridSearchCV
estimator: LogisticRegression
LogisticRegression
```

Code showing model training

```sql
%%sql

select "Date"
from spacextbl
where "Landing _Outcome" = "Success (ground pad)"
order by "Date"
```

```
 * sqlite:///my_data1.db
Done.
```

SQL code

Thank you!