Winning Space Race with Data Science

Prisha Agarwal
12/17/2024

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection through API
  - Data Collection with Web Scraping
  - Data Wrangling
  - Exploratory Data Analysis with SQL
  - Exploratory Data Analysis with Data Visualization
  - Interactive Visual Analytics with Folium
  - Machine Learning Prediction
- Summary of all results
  - Exploratory Data Analytics result
  - Interactive analytics in screenshots
  - Predictive Analytics result

Introduction

Background

Space X, a leader in the space research industry, aims to make space travel affordable and accessible. It 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 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 space X for a rocket launch. The goal of this project is to create a machine learning pipeline to predict if the first stage will land successfully.

- Problems you want to find answers
  - What factors determine if the rocket will land successfully?
  - The interaction amongst various features that determine the success rate of a successful landing
  - Is the best predictive model for successful landing (binary classification)?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was collected using SpaceX API and web scraping from Wikipedia.
- Perform data wrangling
  - One-hot encoding was applied to categorical features
- 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

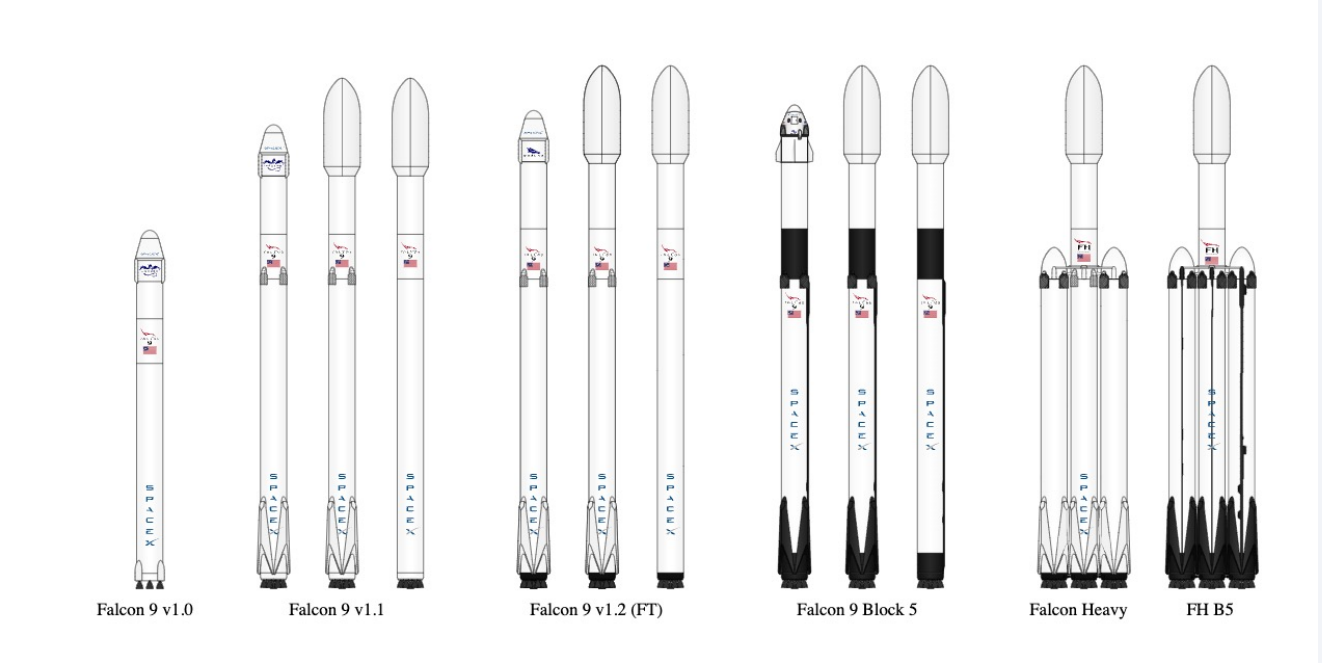
- The data was collected using various methods
  - Get request to the SpaceX API
    - Decode response content as a Json using .json() function call and turn it into a pandas dataframe using .json_normalize().
    - Clean the data, check for missing values and fill in missing values where necessary
  - Perform webscraping from Wikipedia for Falcon 9 launch recors with BeautifulSoup
    - Extract launch records as an HTML table
    - Parse table
    - Convert table to pandas dataframe for future analysis

Data Collection – SpaceX API

1. Request data (SpaceX API rocket launch data)
2. Decode response using .json() and convert to a dataframe using .json_normalize()
3. Request information about the launches from SpaceX API using custim functions
4. Create dictionary from the data
5. Create data frame from the dictionary
6. Filter dataframe to contain only Falcon 9 launches
7. Replace missing values to Payload Mass with calculated .mean()
8. Export data to csv file

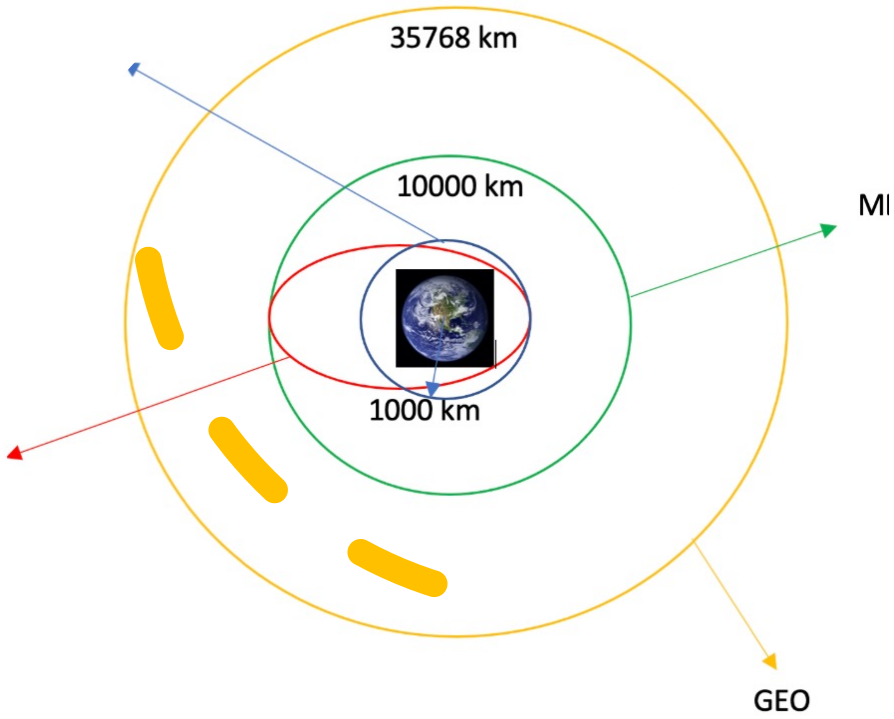
Data Collection - Scraping

- Applied web scrapping to webscrap Falcon 9 launch records with BeautifulSoup
- Parsed the table and converted it into a pandas dataframe.
- Create dictionary from data
- Create dataframe from dictionary
- Export data to csv file



Data Wrangling

- I performed exploratory data analysis (EDA) and determined the training labels
- Calculated the number of launches at each site, and the number and occurrence of each orbits
- Created landing outcome label from outcome column and exported the results of csv. = Landing was not always successful!!



Use the method `.value_counts()` on the column `Outcome` to determine the number of `landing_outcomes` .Then assign it to a variable landing_outcomes.

In [8]:
```
# landing_outcomes = values on Outcome column
landing_outcomes = df['Outcome'].value_counts()
landing_outcomes
```

Out[8]:
```
True ASDS      41
None None      19
True RTLS      14
False ASDS      6
True Ocean      5
False Ocean     2
None ASDS       2
False RTLS      1
Name: Outcome, dtype: int64
```

EDA with Data Visualization

- Explored data by visualizing the relationship between flight number and launch site, payload and launch site, success rate of each orbit type, flight number and orbit type, the launch success yearly trend.
- Charts:
  - Flight Number vs. Payload
  - Flight Number vs. Launch Site
  - Payload Mass (kg) vs. Launch Site
  - Payload Mass (kg) vs. Orbit type

EDA with SQL

- Loaded the SpaceX dataset into a PostgreSQL database without leaving the jupyter notebook
- Applied EDA with SQL to get insight from the data. We wrote queries to find out for instance:
  - The names of unique launch sites in the space mission
  - The total payload mass carried by boosters launched by NASA (CRS)
  - The average payload mass carried by booster version F9 v1.1
  - The total number of successful and failure mission outcomes
  - The failed landing outcomes in drone ship, their booster version and launch site names.

Build an Interactive Map with Folium

- Marked all launch sites, and added map objects such as markers, circles, lines to mark the success or failure of launches for each site on the folium map
- Assigned the feature launch outcomes (failure or success) to class 0 and 1 .i.e., 0 for failure, and 1 for success
- Using the color-labeled marker clusters, we identified which launch sites have relatively high success rate
- Calculated the distances between a launch site to its proximities:
    - Are launch sites near railways, haighways and coastlines?
    - Do launch sites keep certain distance away from cities?

Build a Dashboard with Plotly Dash

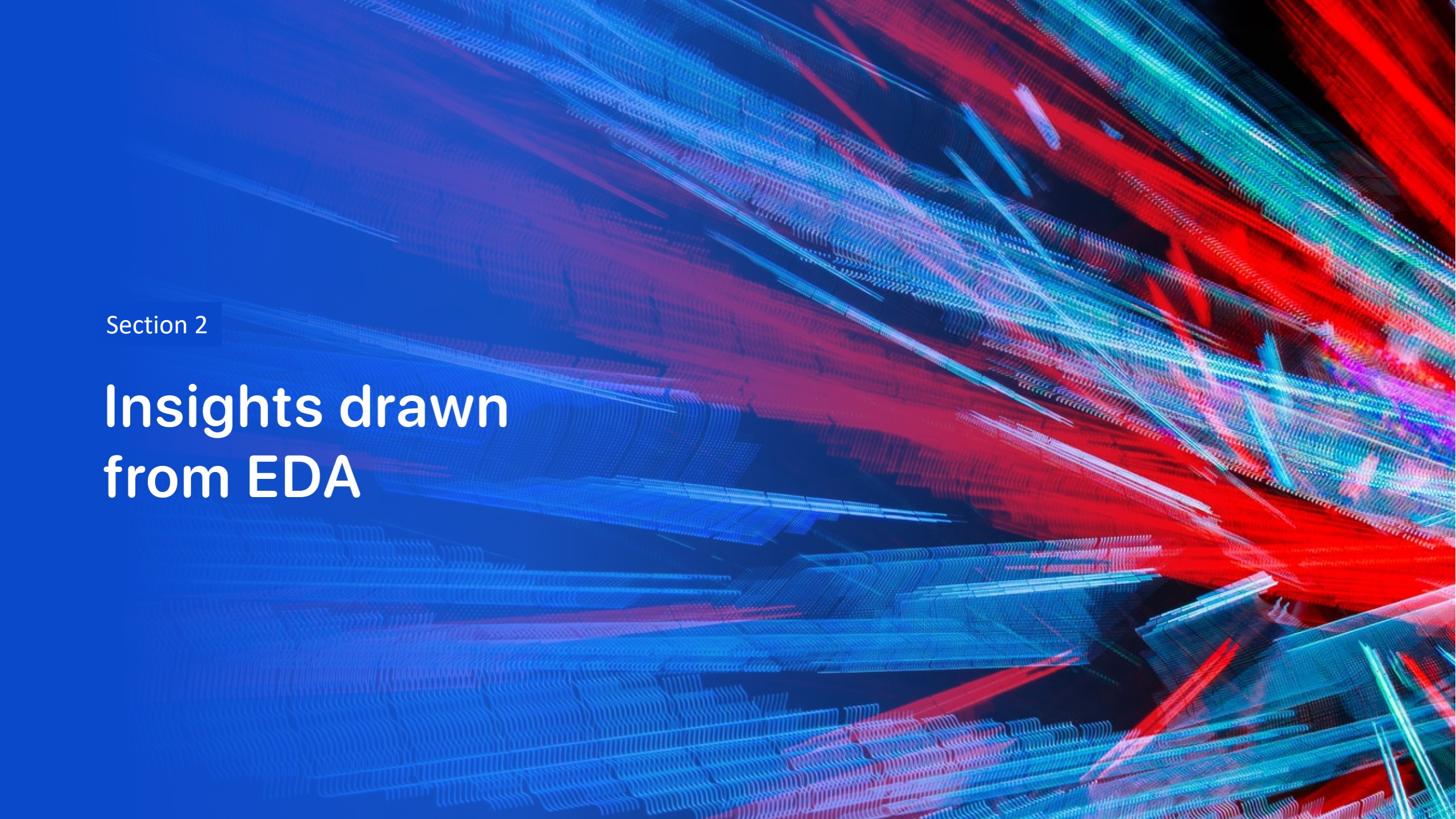
- Built an interactive dashboard with Plotly dash
    - Dropdown list with Launch Sites --> allow user to select all launch sites or a certain one
- Plotted pie charts showing the total launches by certain sites
- Plotted scatter graph showing the relationship with Outcome and Payload Mass (Kg) for the different booster version
- Slider of Payload Mass Range
    - Allow user to select payload mass range

Predictive Analysis (Classification)

- Loaded the data using numpy and pandas, transformed the data, split our data into training and testing
- Built different machine learning models and tune different hyperparameters using GridSearchCV.
- Used accuracy as the metric for our model, improved the model using feature engineering the algorithm tuning
- Found the best performing classification model using Jaccard_Score, F1_Score and Accuracy

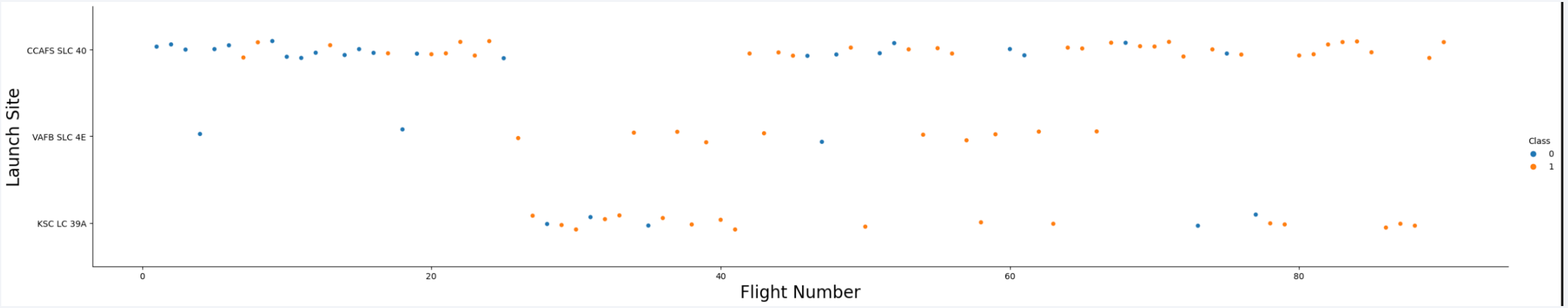
Results

- **Exploratory data analysis results**
  - Launch success has improved over time
  - KSC LC-39A has the highest success rate among landing sites
  - Orbits ES-L1, GEO, HEO and SSO have a 100% success rate
- **Interactive analytics demo in screenshots**
  - Most launch sites are near the equator, and all are close to the coast
  - Launch sites are far away from anything a failed launch can damage (city, highway, railway) while still close enough to bring people and material to support launch activities
- **Predictive analysis results**
  - Decision Tree model is the best predictive model for the dataset


Section 2
Insights drawn from EDA

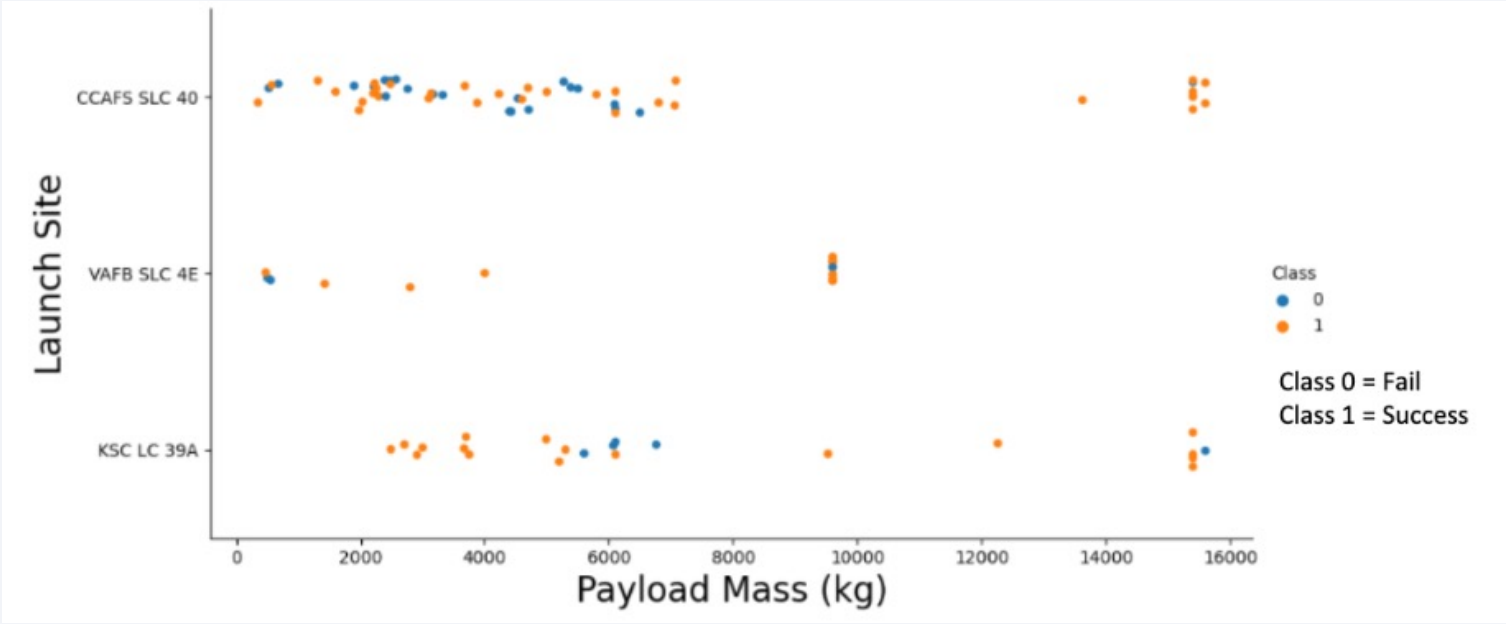
Flight Number vs. Launch Site

- Earlier flights had a lower success rate (blue = fail)
- Later flights had a higher success rate (orange = success)
- Around half of launches were from CCAFSSLC 40 launch site
- VAFB SLC 4E and KSC LC 39A have higher success rates
- We can infer that new launches have a higher success rate



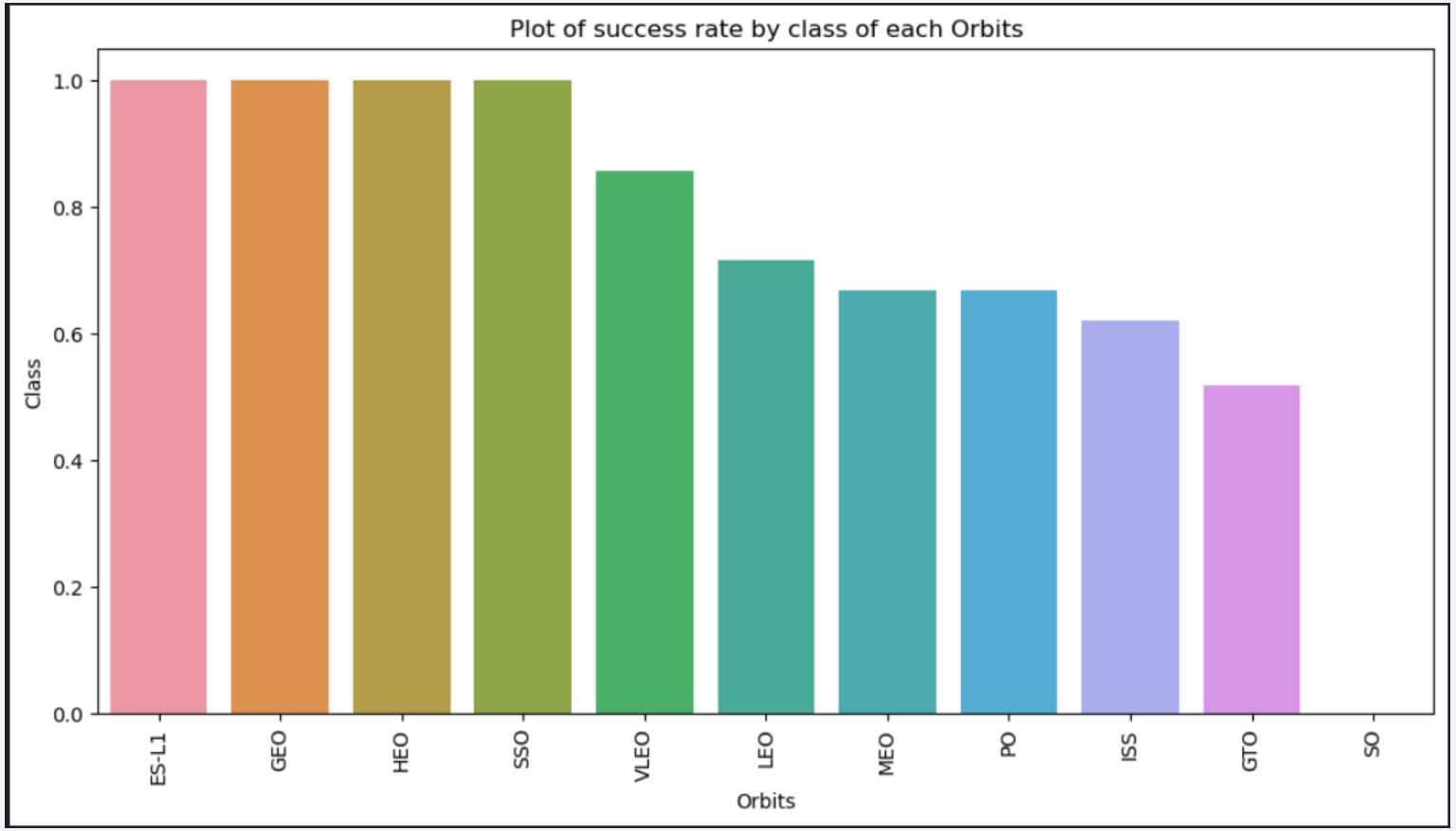
Payload vs. Launch Site

- The greater the payload mass (kg), the higher the success rate
- Most launched with a payload greater than 7000kg were successful
- KSC LC 39A has a 100% success rate for launches less than 5,500 kg
- VAFB SKC 4E has not launched anything greater than around 10K kg



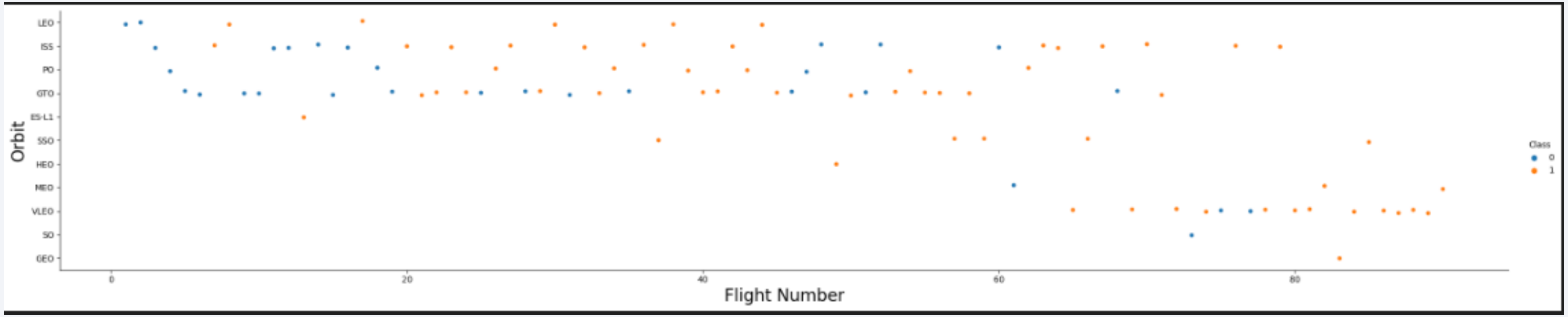
Success Rate vs. Orbit Type

- **100% success rate:** ES-L1, GEO, HEO, SSO
- **50%-80% success rate:** LEO, MEO, PO, ISS, GTO
- **0% success rate:** SO



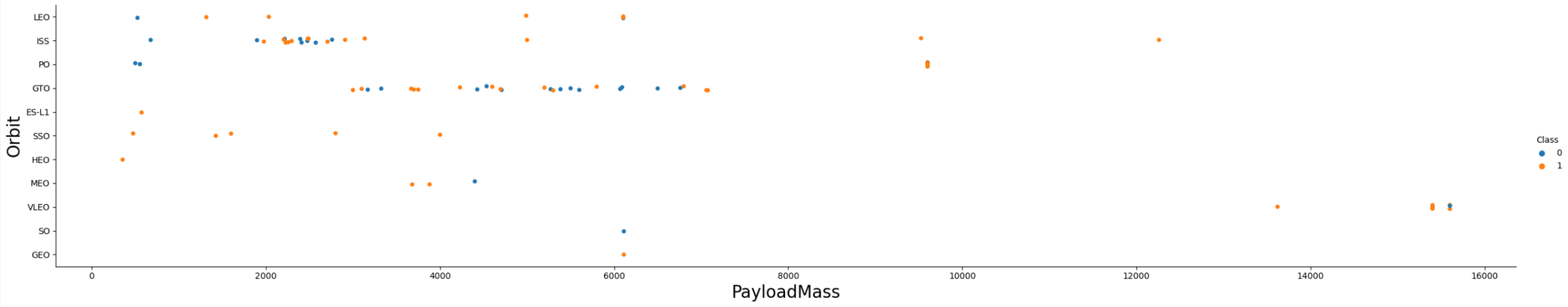
Flight Number vs. Orbit Type

- The success rate typically increases with number of flights for each orbit
- In LEO orbit, success is related to the number of flights whereas in the GTO orbit, there is no relationship between flight number and orbit.



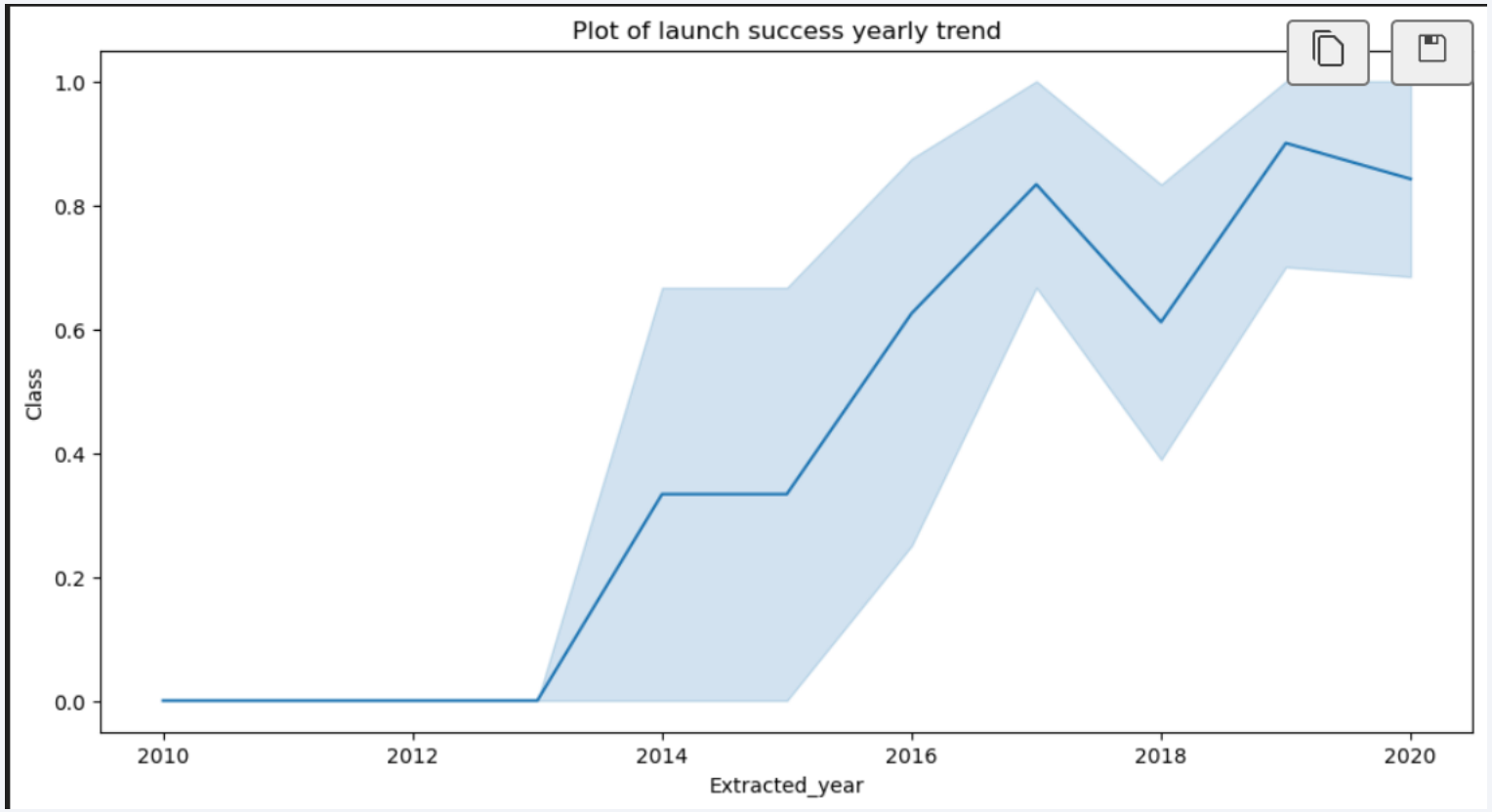
Payload vs. Orbit Type

- We can observe that with heavy payloads, the successful landing are more for PO, LEO, and ISS orbits
- The GTO orbit has mixed success with heavier payloads



Launch Success Yearly Trend

- Success rate improved from 2013-2017 and 2018-2019
- Success rate decreased from 2017-2018 and from 2019-2020
- Overall, success rate improved since 2013



All Launch Site Names

- Use the key word DISTINCT to show only unique launch sites from SpaceX data

Launch Site Names

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

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

```
 * ibm_db_sa://yyy33800:***@1bbf73c5-d84a-4bb0-85b9-ab1a4348f4a4.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:32286/BLUDB
   sqlite:///my_data1.db
Done.
```

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

Launch Site Names Begin with 'CCA'

- Here are the 5 beginning launch site names with ‘CCA’ and the inputed query

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

```
 * ibm_db_sa://yyy33800:***@1bbf73c5-d84a-4bb0-85b9-ab1a4348f4a4.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:32286/BLUDB
   sqlite:///my_data1.db
Done.
```

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

Payload Mass

`

- **Total Payload Mass = 45,596 kg** (total) carried by boosters launched by NASA (CRS)

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) \
    FROM SPACEXTBL \
    WHERE CUSTOMER = 'NASA (CRS)';
 * ibm_db_sa://yyy33800:***@1bbf73c5-d84a-4
   sqlite:///my_data1.db
Done.
    1
45596
```

- **Average Payload Mass = 2,928 kg** (average) carried by booster version F9 v1.1

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) \
    FROM SPACEXTBL \
    WHERE BOOSTER_VERSION = 'F9 v1.1';
 * ibm_db_sa://yyy33800:***@1bbf73c5-d84a-4
   sqlite:///my_data1.db
Done.
    1
2928
```

First Successful Ground Landing Date

The dates of the first successful landing outcome on ground pad was 22nd December 2015

```
%sql SELECT MIN(DATE) \
FROM SPACEXTBL \
WHERE LANDING__OUTCOME = 'Success (ground pad)'
```

```
 * ibm_db_sa://yyy33800:***@1bbf73c5-d84a-4bb0-85b
   sqlite:///my_data1.db
Done.
```

1
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- JSCAT-14, JSCAT-16, SES-10, SES-11/EchoStar 105
- Used the WHERE clause to filter for boosters, which have successfully landed on droneship and applied the AND condition to determine successful landing with payload mass greater than 4000 but less than 6000.

```
%sql SELECT PAYLOAD \
FROM SPACEXTBL \
WHERE LANDING__OUTCOME = 'Success (drone ship)' \
AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000;
```

```
 * ibm_db_sa://yyy33800:***@1bbf73c5-d84a-4bb0-85b9
   sqlite:///my_data1.db
Done.
```

payload
JCSAT-14
JCSAT-16
SES-10
SES-11 / EchoStar 105

Total Number of Successful and Failure Mission Outcomes

- 99 total successful mission outcomes
- 1 failure mission outcome
- Used wildcard like '%' to filter for WHERE MissionOutcome was a success or a failure

```
%sql SELECT MISSION_OUTCOME, COUNT(*) as total_number \
FROM SPACEXTBL \
GROUP BY MISSION_OUTCOME;
```

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

Mission_Outcome	total_number
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

Boosters Carried Maximum Payload

- Determined the booster that have carried the maximum payload using a subquery in the WHERE clause and the MAX() function.
  - F9 B5 V1048.4
  - F9 B5 V1049.4
  - F9 B5 V1051.3
  - F9 B5 V1056.4
  - F9 B5 V1048.5
  - F9 B5 V1051.4
  - F9 B5 V1049.5
  - F9 B5 V1060.2
  - F9 B5 V1058.3
  - F9 B5 V1051.6
  - F9 B5 V1060.3
  - F9 B5 V1049.7

```
%sql SELECT BOOSTER_VERSION \
FROM SPACEXTBL \
WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL);
```

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

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

2015 Launch Records

Used a combination of the WHERE clause, LIKE, AND, and BETWEEN conditions to filter for failed landing outcomes in drone ship, their booster versions, and launch site names for year 2015

```
%sql SELECT substr(Date,4,2) as month, DATE,BOOSTER_VERSION, LAUNCH_SITE, [Landing _Outcome] \
FROM SPACEXTBL \
where [Landing _Outcome] = 'Failure (drone ship)' and substr(Date,7,4)='2015';
```

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

month	Date	Booster_Version	Launch_Site	Landing _Outcome
01	10-01-2015	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
04	14-04-2015	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

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

- Selected Landing outcomes and the COUNT of landing outcomes from the data and used the WHERE clause to filter for landing outcomes BETWEEN 2010-06-04 to 2010-03-20
- Applied the GROUP BY clause to group the landing outcomes and the ORDER BY clause to order the grouped landing outcome in descending order.

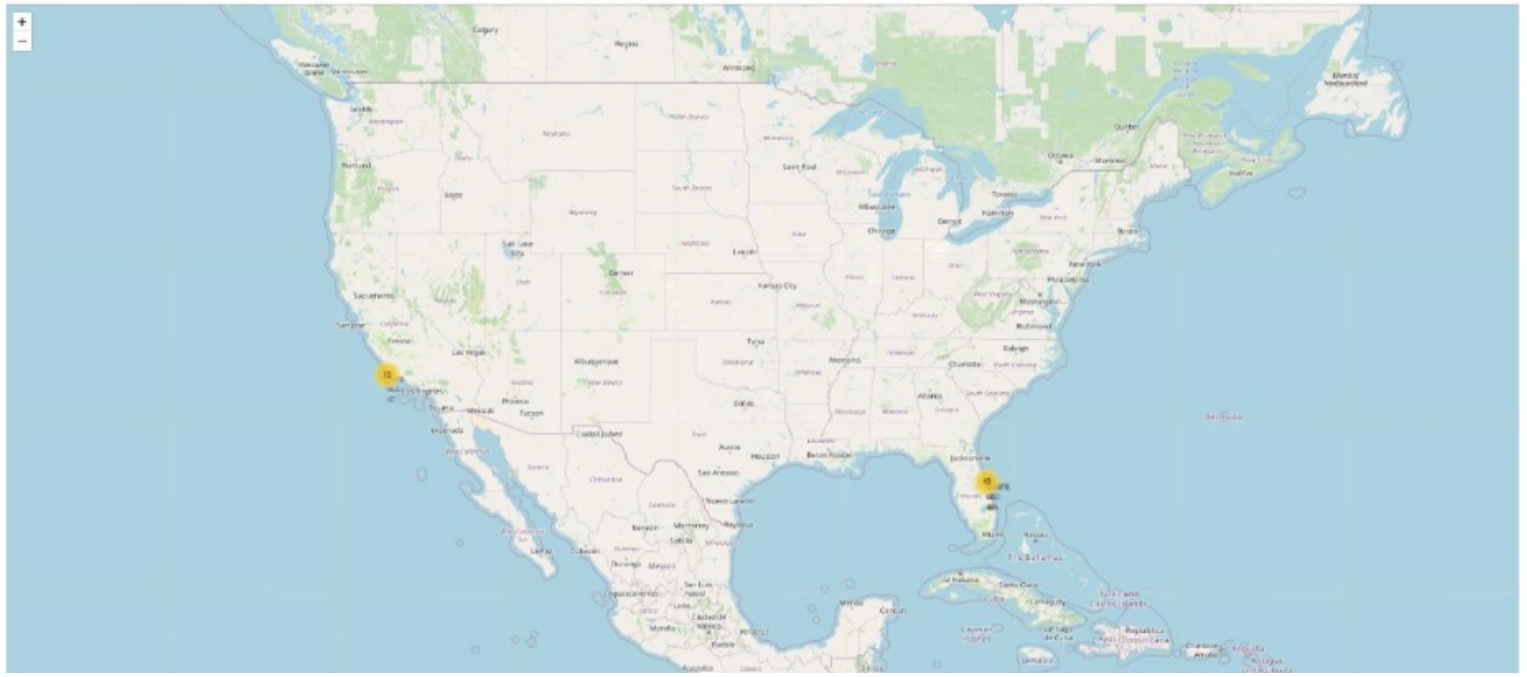
```
%sql SELECT [Landing _Outcome], count(*) as count_outcomes \
FROM SPACEXTBL \
WHERE DATE between '04-06-2010' and '20-03-2017' group by [Landing _Outcome] order by count_outcomes DESC;
```

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

Landing _Outcome	count_outcomes
Success	20
No attempt	10
Success (drone ship)	8
Success (ground pad)	6
Failure (drone ship)	4
Failure	3
Controlled (ocean)	3
Failure (parachute)	2
No attempt	1

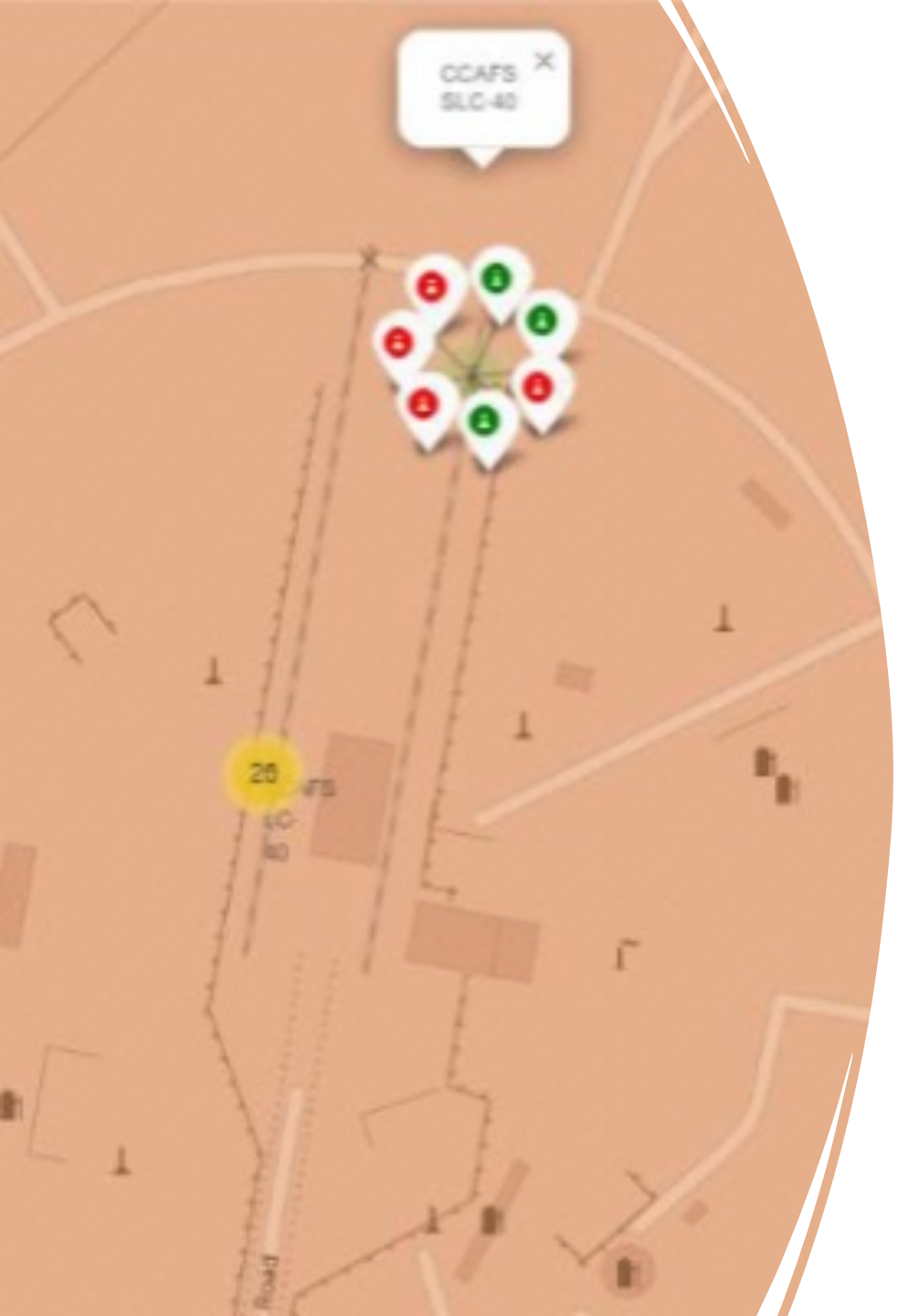

Section 3
Launch Sites
Proximities Analysis

All launch sites global map markers



Near Equator: the closer the launch site to the equator the easier it is to launch to equatorial orbit, and the more help you get from Earth's rotation for a prograde orbit.

**Rockets launched from sites nearby get an additional natural boost – due to rotational speed of Earth – that helps save cost of putting in extra fuel.



Launch Outcomes

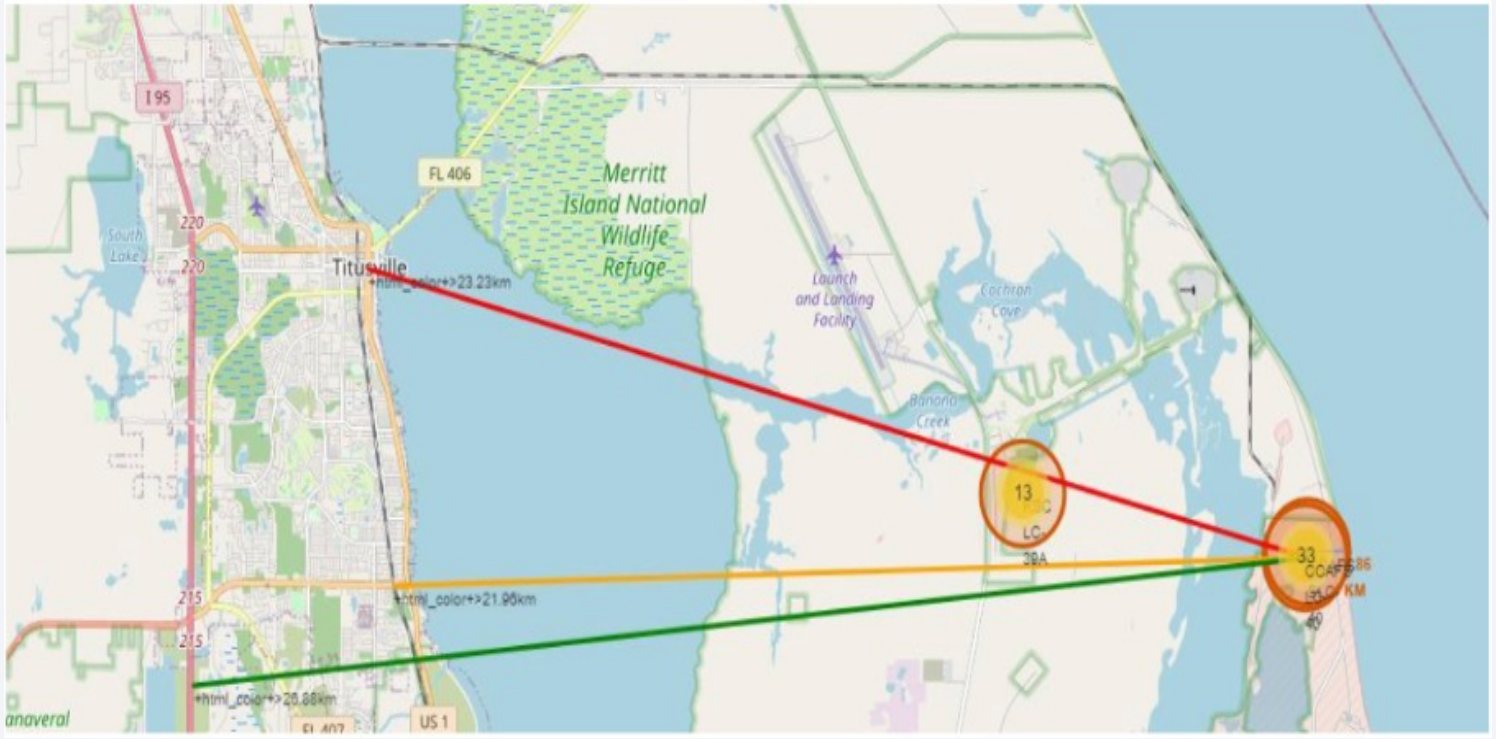
At Each Launch Site: -

- **Green** markers for successful launches
- **Red** markers for unsuccessful launches
- Launch site CCAF SLC-40 has a 3/7 success rate (42.9%)

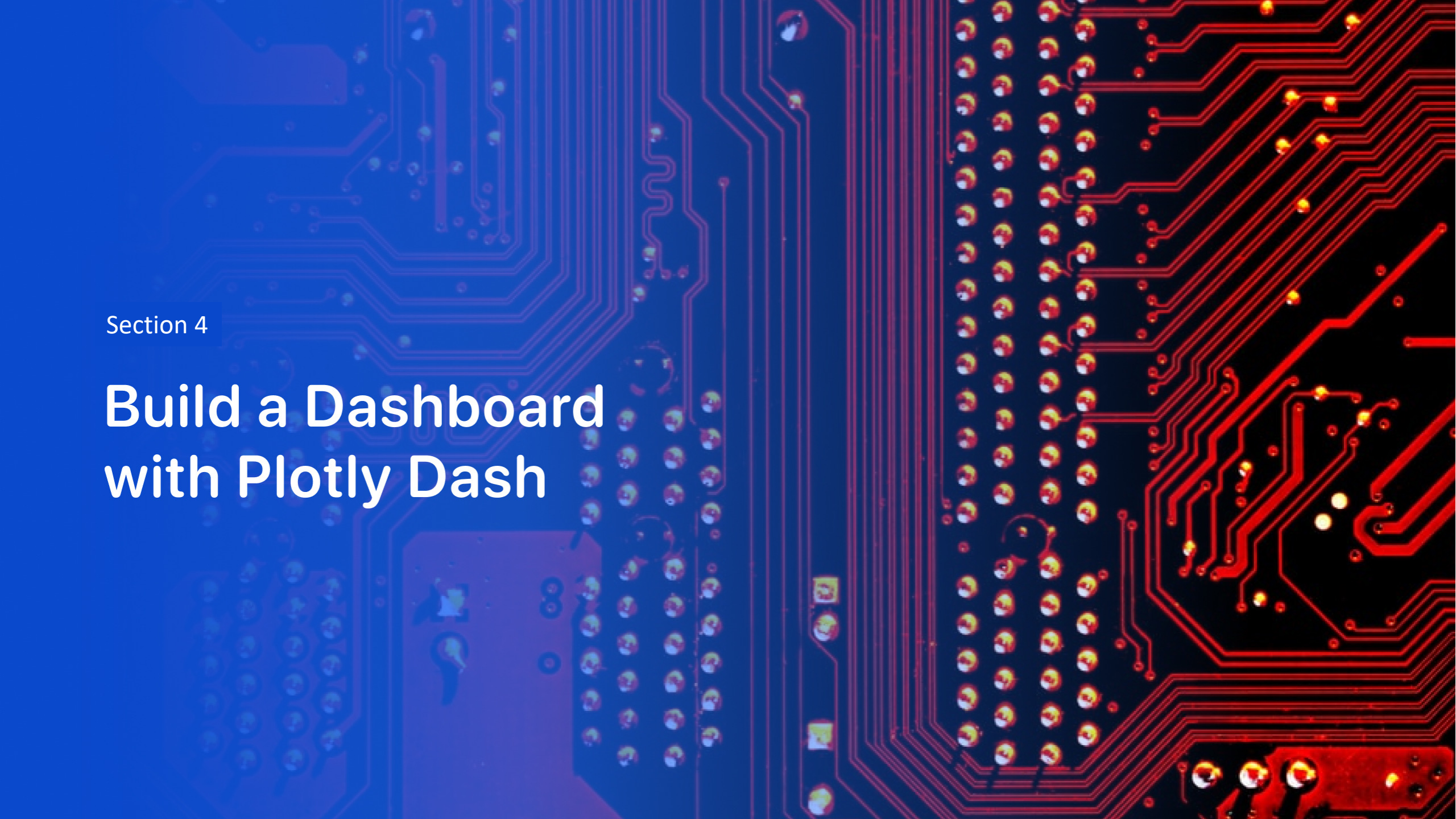
Distance to Proximities

CCAFS SLC-40

- .86 km from nearest coastline
- 21.96 km from nearest railway
- 23.23 km from nearest city
- 26.88 km from nearest highway

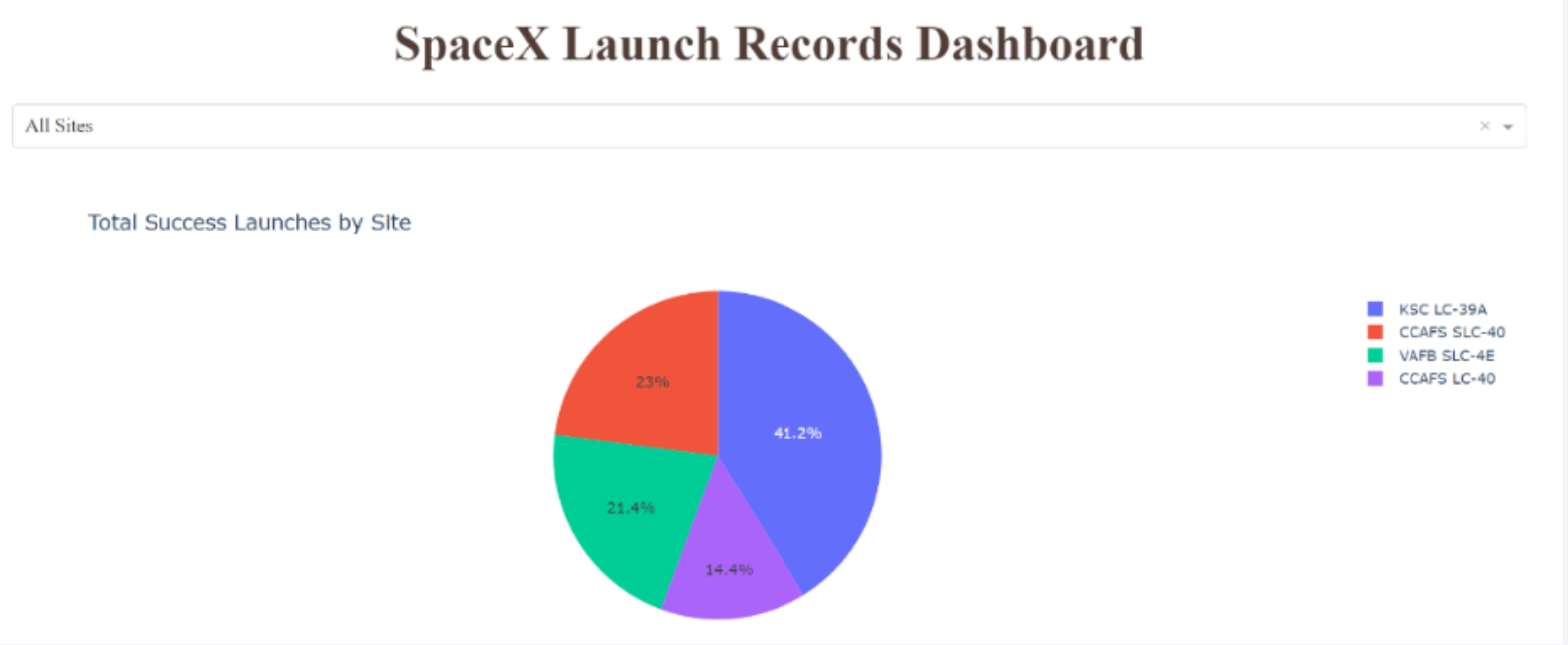


- **Coasts:** help ensure that spent stages dropped along the launch path/failed launches don't fall on people/property
- **Safety/security:** exclusion zone around launch site needed
- **Transportation/Infrastructure and Cities:** away from anything failed launch can damage but close enough to bring roads/rails/docks for support of launch activities


Section 4
Build a Dashboard with Plotly Dash

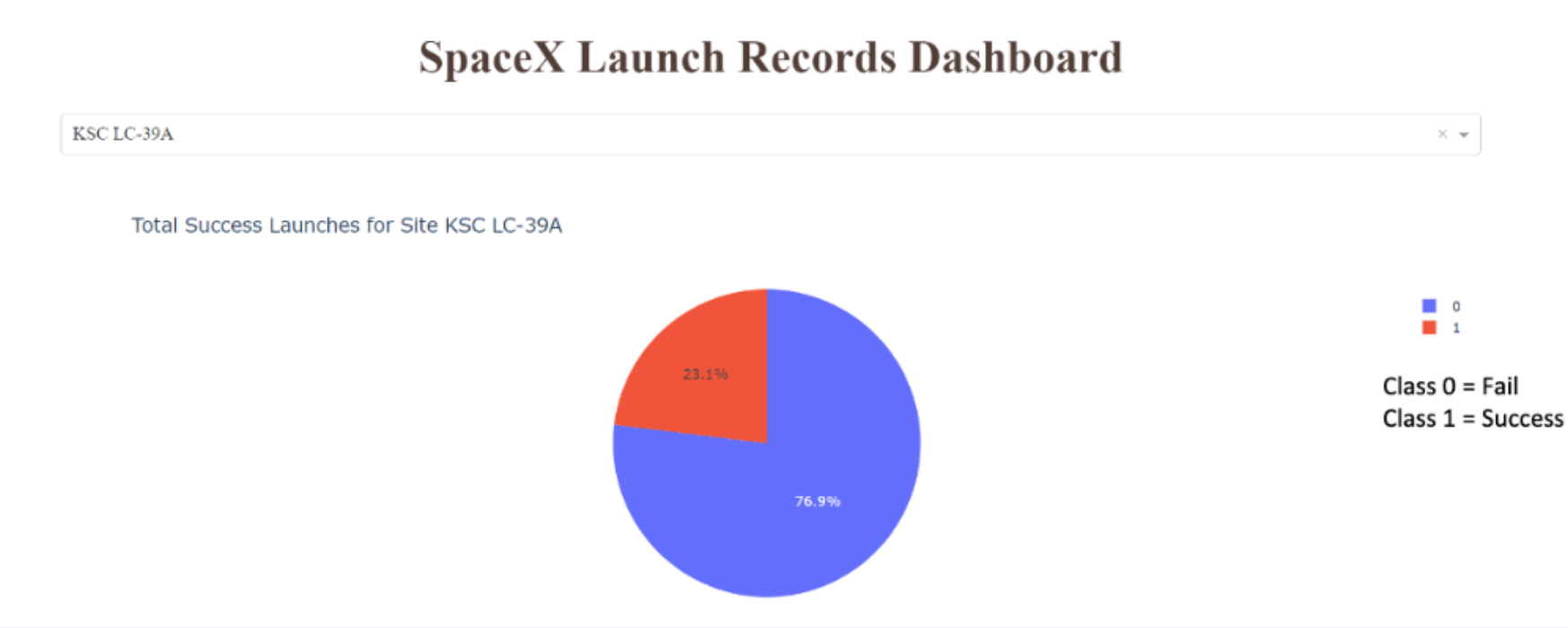
Pie chart of Total Success Launches For All Sites (%)

- KSC LC-39A has the most successful launches among launch sites (41.2%)



Pie chart of Launch site with Highest Launch Success Ratio

- KSC LC-39A has the highest success rate amongst launch sites (76.9%)
- 10 successful launches and 3 failed launches



Scatter Plot of Payload vs. Launch Outcome for All Sites

- Success rates for low weighted payloads is higher than the heavy weighted payloads
- Payloads between 2000 kg – 5000 kg have the highest success rate
- 1 indicating successful outcome and 0 indicating an unsuccessful outcome

Section 5

Predictive Analysis (Classification)

Classification Accuracy

- .best_score_ is the average of all cv folds for a single combination of the parameters
- The **decision tree** classifier is the model with the highest classification accuracy

	LogReg	SVM	Tree	KNN
Jaccard_Score	0.800000	0.800000	0.800000	0.800000
F1_Score	0.888889	0.888889	0.888889	0.888889
Accuracy	0.833333	0.833333	0.833333	0.833333

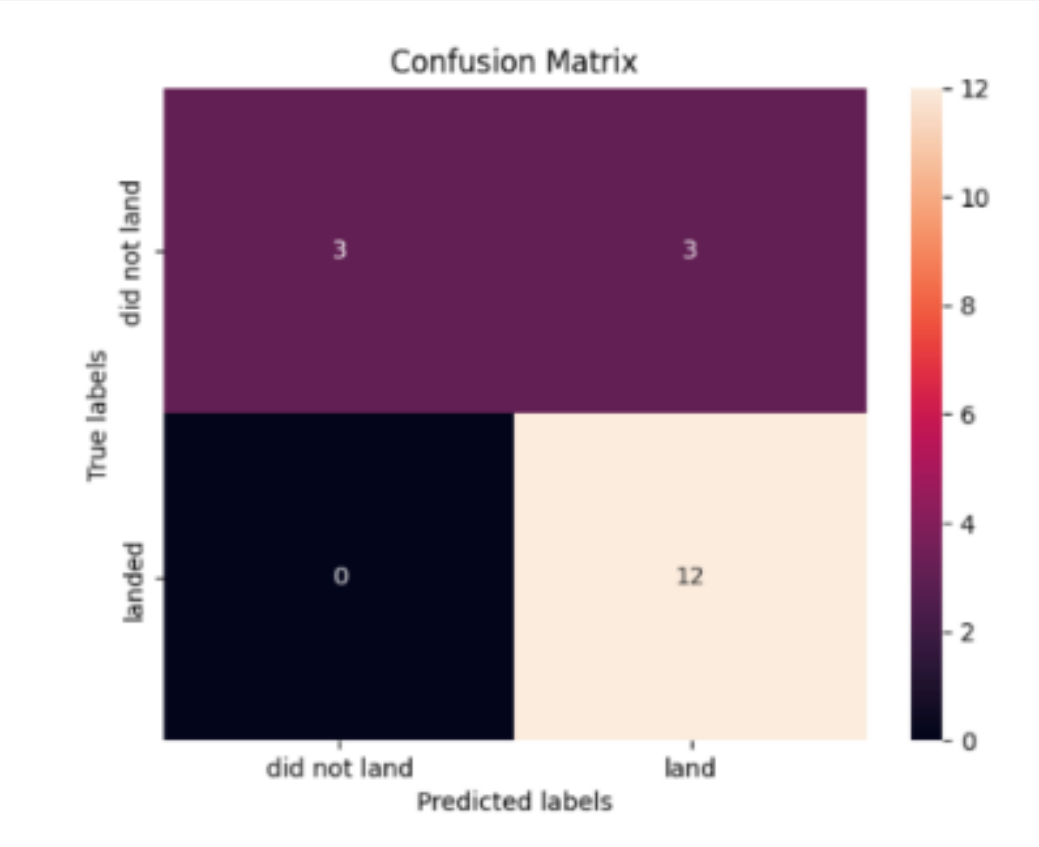
```
models = {'KNeighbors':knn_cv.best_score_,
          'DecisionTree':tree_cv.best_score_,
          'LogisticRegression':logreg_cv.best_score_,
          'SupportVector': svm_cv.best_score_}

bestalgorithm = max(models, key=models.get)
print('Best model is', bestalgorithm,'with a score of', models[bestalgorithm])
if bestalgorithm == 'DecisionTree':
    print('Best params is :', tree_cv.best_params_)
if bestalgorithm == 'KNeighbors':
    print('Best params is :', knn_cv.best_params_)
if bestalgorithm == 'LogisticRegression':
    print('Best params is :', logreg_cv.best_params_)
if bestalgorithm == 'SupportVector':
    print('Best params is :', svm_cv.best_params_)
```

```
Best model is DecisionTree with a score of 0.9017857142857142
Best params is : {'criterion': 'gini', 'max_depth': 16, 'max_features': 'auto', 'min_samples_leaf': 4, 'min_samples_split': 10, 'splitter': 'random'}
```

Confusion Matrix

- The confusion matrix for the decision tree classifer shows that the classifier can distinguish between the different classes. The major problem is the false positives i.e. unsuccessful landing marked as successful landing by the classifier
- Outputs:
  - 12 True positive
  - 3 True negative
  - 3 False positive
  - 0 False Negative



Conclusions

- The larger the flight amount at a launch site, the greater the success rate at a launch site
- Launch success rate started to increase in 2013 till 2020
- Orbits ES-L1, GEO, HEO, SSO, VLEO had the most success rate
- KSC LC-39A had the most successful launches of any sites
- The Decision tree classifier is the best machine learning algorithm for this task

Appendix

- Github repository link: https://github.com/pagarwal1501/datasciencecertification

Thank you!