Winning Space Race with Data Science

Nimisha Agrawal

15th October 2023

Outline

Executive Summary

- Summary of methodologies
  - Data Collection through SpaceX REST API and web scraping
  - Data Wrangling to create success/ fail outcome variable
  - Explore data with data visualization techniques, considering the following factors: payload, launch site, flight number and yearly trend
  - Analyze the data with SQL, calculating the following statistics: total payload, payload range for successful launches, and total # of successful and failed outcomes
  - Explore launch site success rates and proximity to geographical markers
  - Visualize the launch sites with the most success and successful payload ranges
  - Build Models to predict landing outcomes using logistic regression, support vector machine (SVM), decision tree and K-nearest neighbor (KNN)
- Summary of all results
  - Exploratory Data Analysis : Launch success rate has improved over time.
  - Interactive dashboard/visualization : Most launch sites are near the equator, all are close to the coast.
  - Predictive Analytics results : All models performed similarly on the test set. The decision tree model slightly outperformed

Introduction

- Project background and context

Space X 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. This goal of the project is to create a machine learning pipeline to predict if the first stage will land successfully.

- Problems you want to find answers
  - How payload mass, launch site, number of flights, and orbits affect first-stage landing success
  - Rate of successful landings over time
  - 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:

  Data was split into training data and test data to find the best Hyperparameter for SVM, Classification Trees, and Logistic Regression. Then the method that performs best using test data was determined using confusion matrix.

Data Collection

- Describe how data sets were collected.
  - Data collection was done using get request to the SpaceX API.
  - Next, we decoded the response content as a Json using .json() function call and turn it into a pandas dataframe using .json_normalize().
  - We then cleaned the data, checked for missing values and fill in missing values where necessary.
  - In addition, we performed web scraping from Wikipedia for Falcon 9 launch records with BeautifulSoup.
  - The objective was to extract the launch records as HTML table, parse the table and convert it to a pandas dataframe for future analysis.

Data Collection – SpaceX API

- We used the get request to the SpaceX API to collect data, clean the requested data and did some basic data wrangling and formatting.
- The link to the notebook :

IBM-Data-Science-Capstone/01_NA-spacex-data-collection-api (1).ipynb at main · NimishaAgra/IBM-Data-Science-Capstone (github.com)

Request data from SpaceX API (rocket launch data)

Decode response using .json() and convert to a dataframeusing .json_normalize()

Request information about the launches from SpaceX API using custom functions

Create dictionary from the data

Create dataframe from the dictionary

Filter dataframe to contain only Falcon 9 launches

Replace missing values of Payload Mass with calculated .mean()

Export data to csv file

Data Collection - Scraping

- We applied web scrapping to webscrap Falcon 9 launch records with BeautifulSoup
- We parsed the table and converted it into a pandas dataframe.
- The link to the notebook :

IBM-Data-Science-Capstone/02_NA-webscraping.ipynb at main · NimishaAgra/IBM-Data-Science-Capstone (github.com)

Request data(Falcon 9 launch data) from Wikipedia)

Create BeautifulSoup object from HTML response

Extract column names from HTML table header

Collect data from parsing HTML tables

Create dictionary from the data

Create dataframe from the dictionary

Export data to csv file

Data Wrangling

- We performed exploratory data analysis and determined the training labels.
- We calculated the number of launches at each site, and the number and occurrence of each orbits
- We created landing outcome label from outcome column and exported the results to csv.
- The link to the notebook:

IBM-Data-Science-Capstone/03_NA-spacex-Data wrangling (1).ipynb at main · NimishaAgra/IBM-Data-Science-Capstone (github.com)

EDA with Data Visualization

- We explored the 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.
- The link to the notebook:

IBM-Data-Science-Capstone/05_NA-eda-dataviz.ipynb.jupyterlite.ipynb at main · NimishaAgra/IBM-Data-Science-Capstone (github.com)

EDA with SQL

- Using bullet point format, summarize the SQL queries you performed
- Add the GitHub URL of your completed EDA with SQL notebook, as an external reference and peer-review purpose
- The link to the notebook:
- IBM-Data-Science-Capstone/04_SpaceX_EDA_SQL.ipynb at main · NimishaAgra/IBM-Data-Science-Capstone (github.com)

Build an Interactive Map with Folium

- We 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.
- We 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.
- We calculated the distances between a launch site to its proximities. We answered some question for instance:
  - Are launch sites near railways, highways and coastlines.
  - Do launch sites keep certain distance away from cities.
- Link to the notebook :

IBM-Data-Science-Capstone/06_NA_launch_site_location.jupyterlite.ipynb at main · NimishaAgra/IBM-Data-Science-Capstone (github.com)

Build a Dashboard with Plotly Dash

- We built an interactive dashboard with Plotly dash
- We plotted pie charts showing the total launches by a certain sites
- We plotted scatter graph showing the relationship with Outcome and Payload Mass (Kg) for the different booster version.
- Link to the notebook :

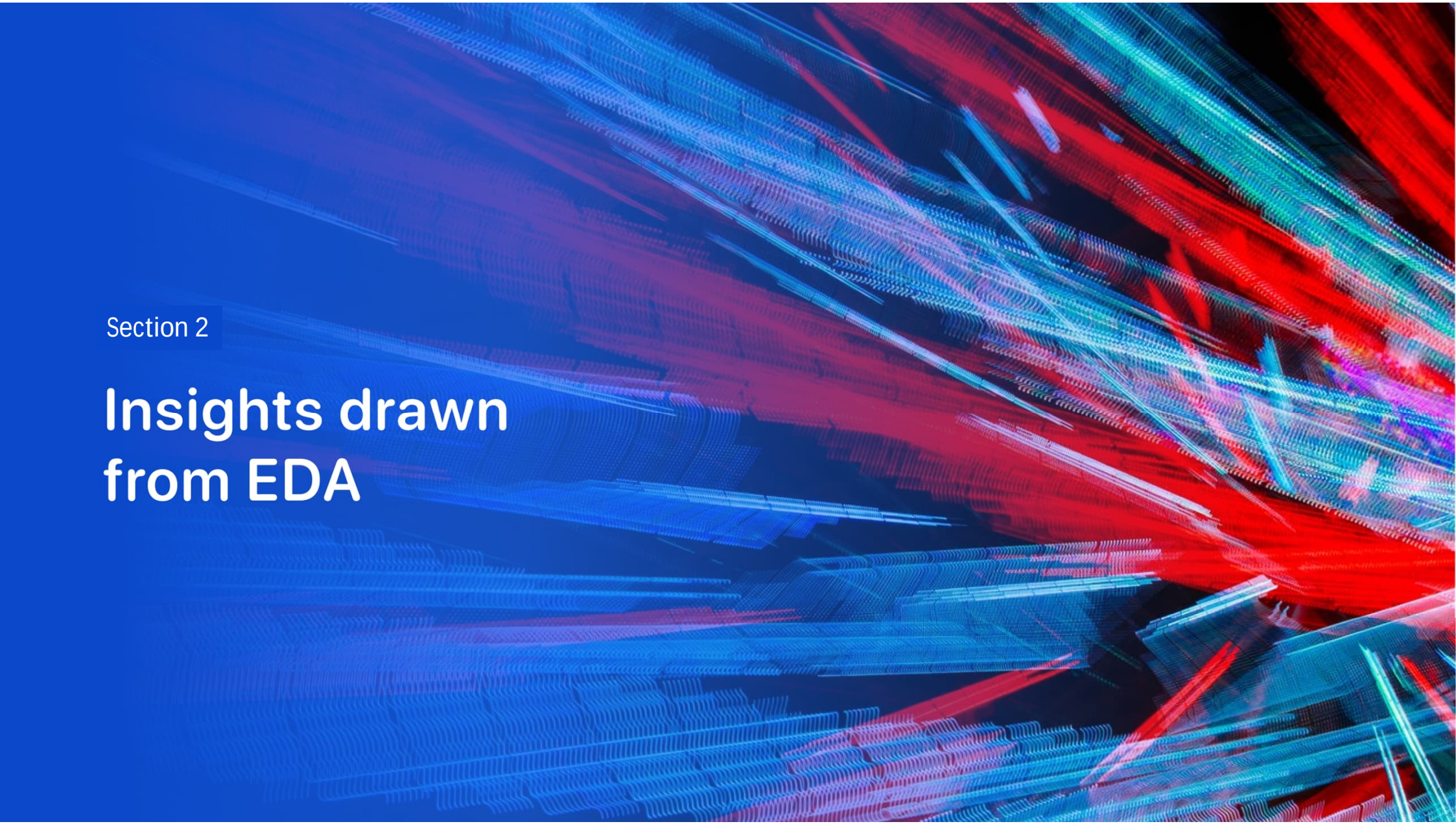
IBM-Data-Science-Capstone/07_SpaceX_Interactive_Visual_Analytics_Plotly.py at main · NimishaAgra/IBM-Data-Science-Capstone (github.com)

Predictive Analysis (Classification)

- We loaded the data using numpy and pandas, transformed the data, split our data into training and testing.
- We built different machine learning models and tune different hyperparameters using GridSearchCV.
- We used accuracy as the metric for our model, improved the model using feature engineering and algorithm tuning.
- We found the best performing classification model.
- Link to the notebook :

IBM-Data-Science-Capstone/08_NA_SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb at main · NimishaAgra/IBM-Data-Science-Capstone (github.com)

Results

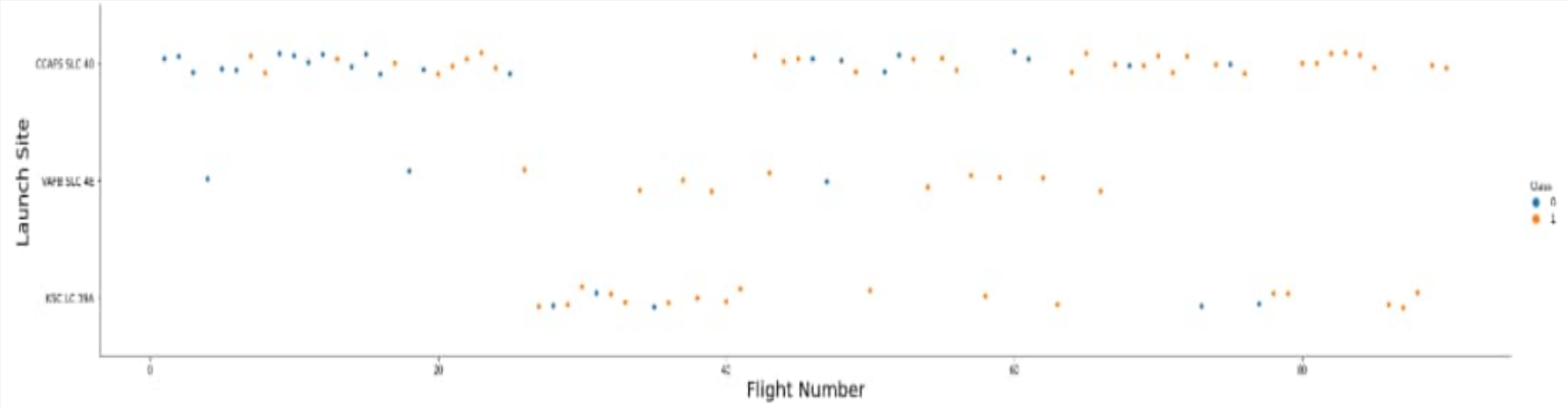


Section 2

Insights drawn from EDA

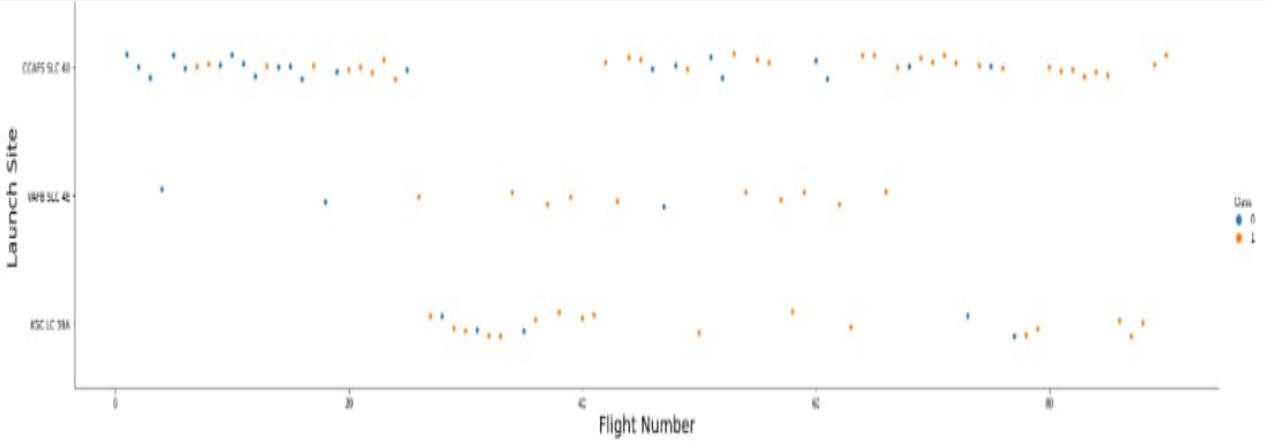
Flight Number vs. Launch Site

- **Earlier flights** had a **lower success rate** (**blue = fail**)
- **Later flights** had a **higher success rate** (**orange = success**)
- Around half of launches were from CCAFS SLC 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.



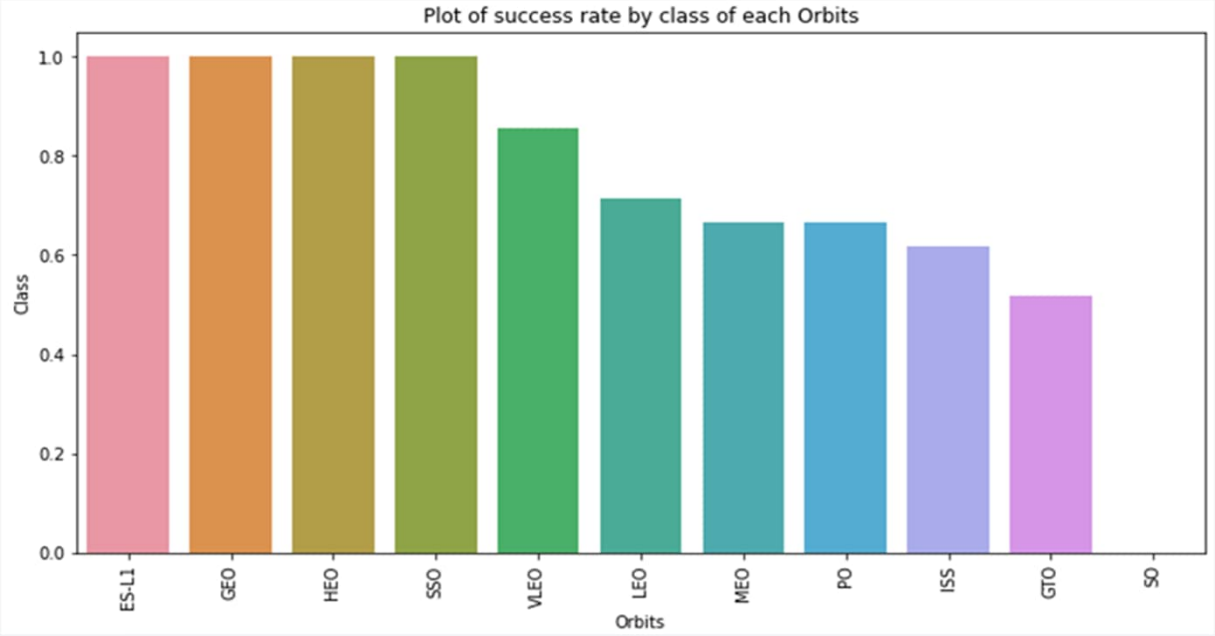
Payload vs. Launch Site

- Typically, the **higher**the **payload mass** (kg), the **higher** the **success rate**
- Most launces with a payload greater than 7,000 kg 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 ~10,000 kg



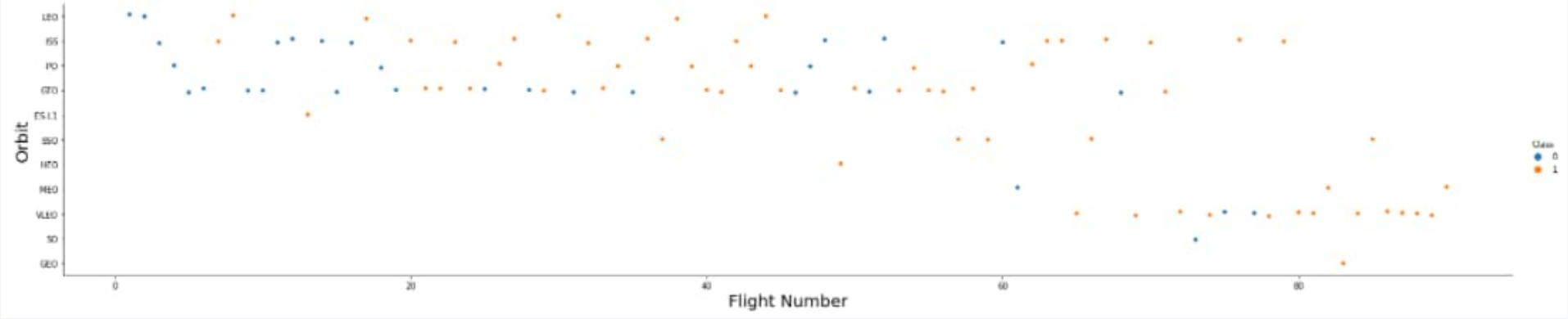
Success Rate vs. Orbit Type

- From the plot, we can see that ES-L1, GEO, HEO, SSO, VLEO had the most success rate.



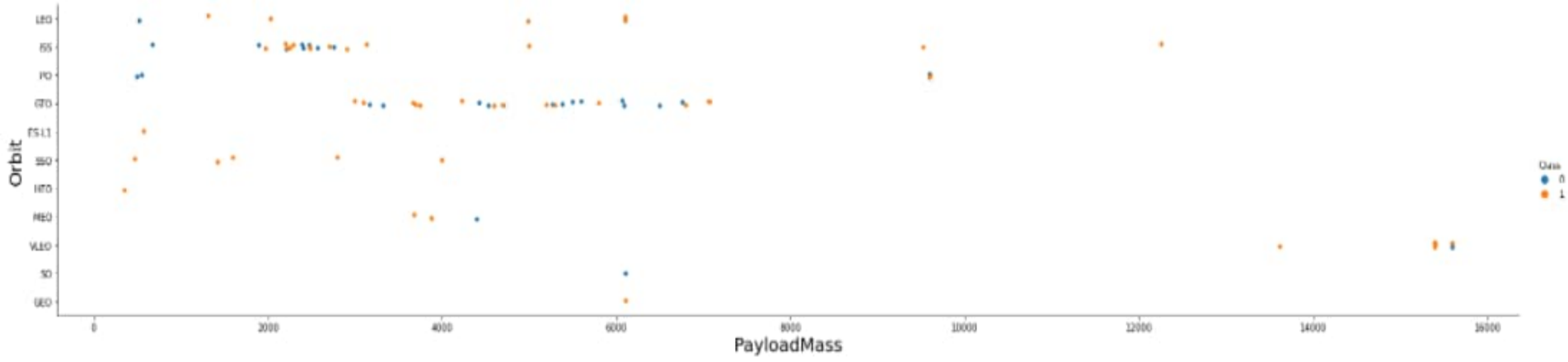
Flight Number vs. Orbit Type

- The success rate typically increases with the number of flights for each orbit
- •This relationship is highly apparent for the LEO orbit
- •The GTO orbit, however, does not follow this trend



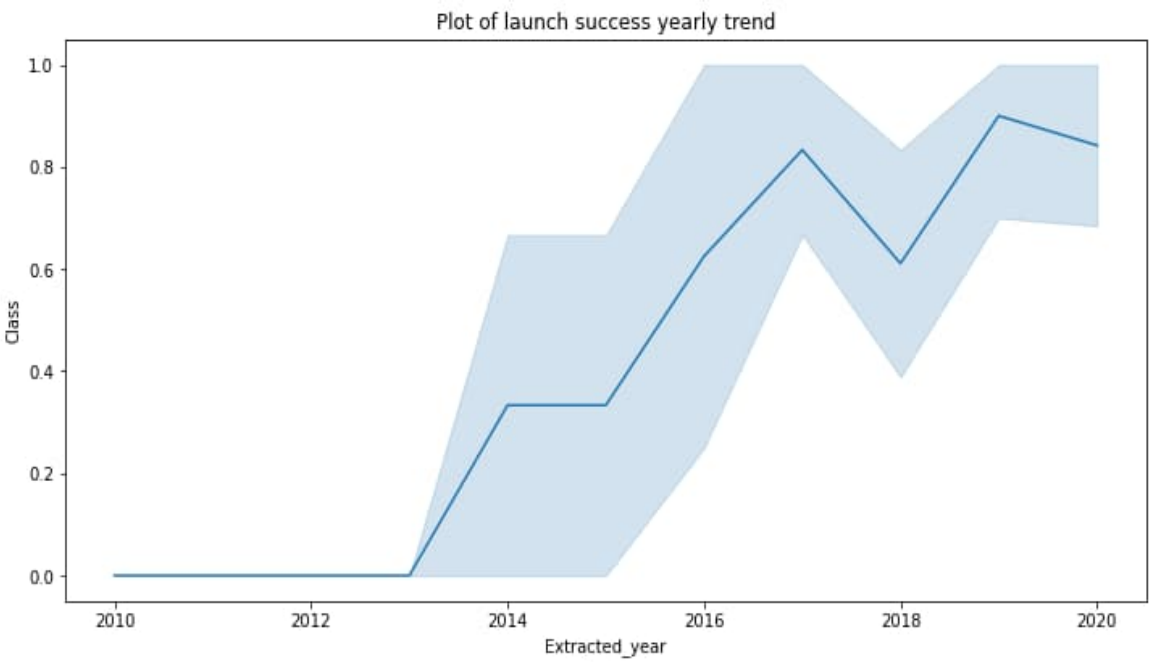
Payload vs. Orbit Type

- Heavy payloads are better with LEO, ISS and PO orbits
- •The GTO orbit has mixed success with heavier payloads



Launch Success Yearly Trend

- The success rate improved from 2013-2017 and 2018-2019
- •The success rate decreased from 2017-2018 and from 2019-2020
- •Overall, the success rate has improved since 2013



All Launch Site Names

- We used the key word **DISTINCT** to show only unique launch sites from the SpaceX data.

Display the names of the unique launch sites in the space mission

In [10]:
```
task_1 = '''
        SELECT DISTINCT LaunchSite
        FROM SpaceX
'''
create_pandas_df(task_1, database=conn)
```

Out[10]:

	launchsite
0	KSC LC-39A
1	CCAFS LC-40
2	CCAFS SLC-40
3	VAFB SLC-4E

Launch Site Names Begin with 'CCA'

- We used the query below to display 5 records where launch sites begin with `CCA`

```
%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

Total 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 by F9 v1.1

- We calculated the average payload mass carried by booster version F9 v1.1 as 2928.4

```
%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

- We observed that 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

- We used the **WHERE** clause to filter for boosters which have successfully landed on drone ship 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

- 1 Failure in Flight
- 99 Success
- 1 Success (payload status unclear)

```
%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

- We determined the booster that have carried the maximum payload using a subquery in the **WHERE** clause and the **MAX()** function.

```
%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

- Showing month, date, booster version, launch site and landing outcome 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

- We 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.
- We 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;
```

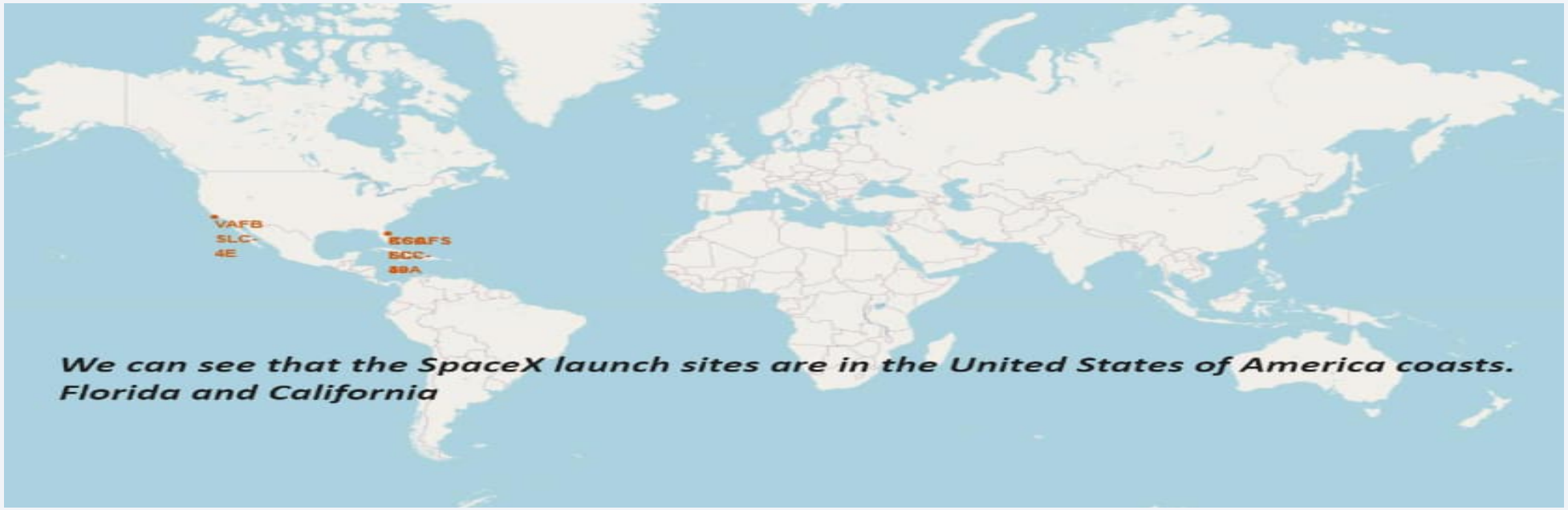
```
 * sqlite:///my_data1.db
Done.
```

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 near the equator get an **additional natural boost**-due to the rotational speed of earth -that **helps save the cost**of putting in extra fuel and boosters.

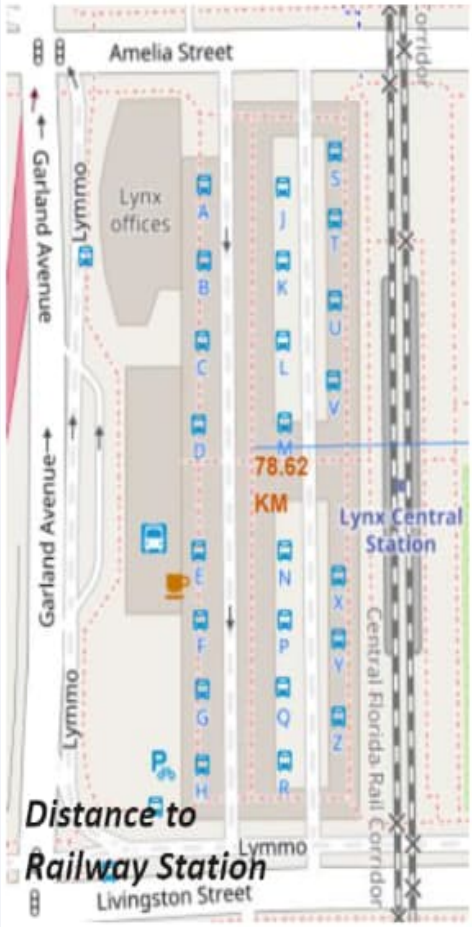


Launch outcomes

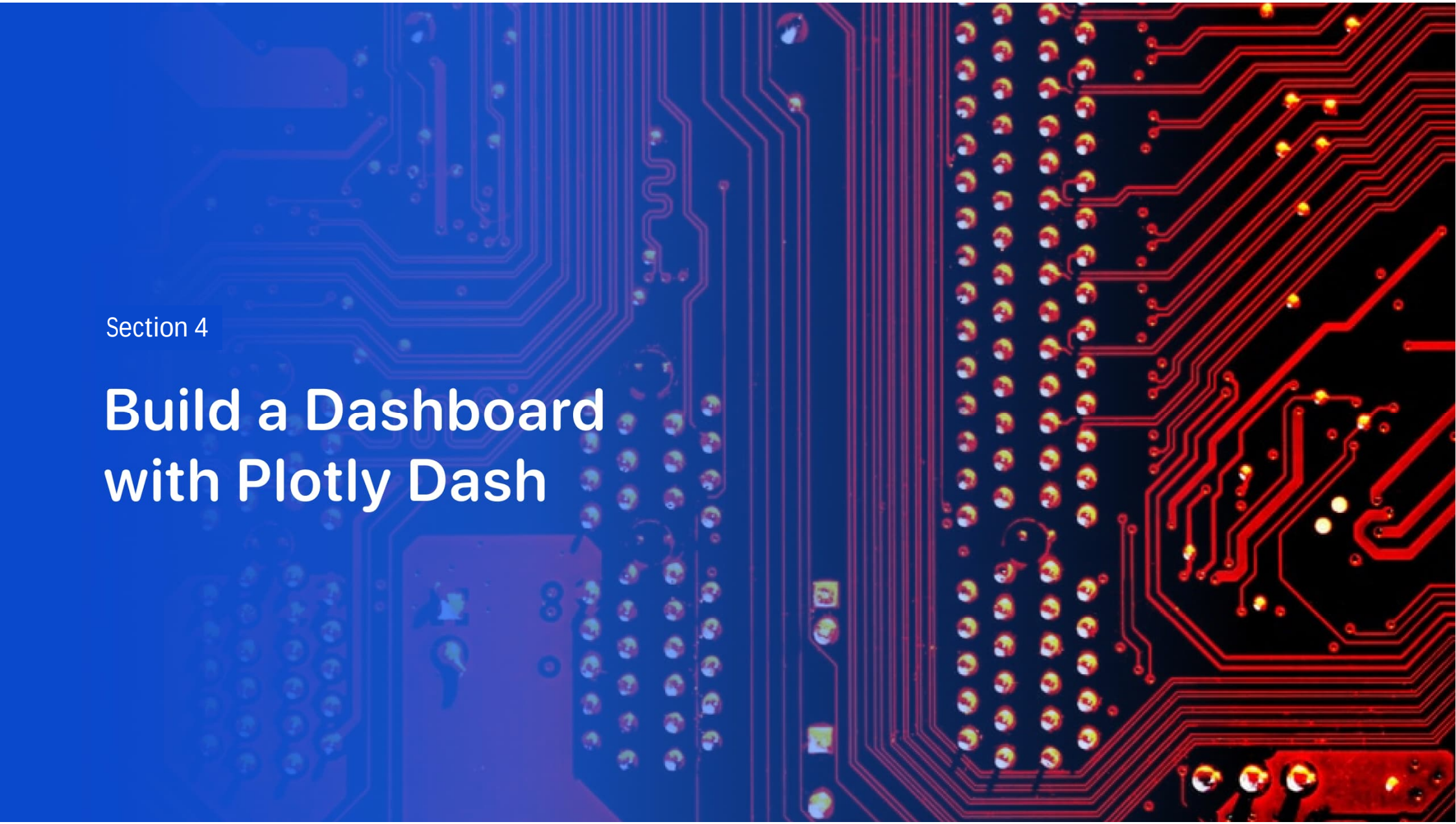
At Each Launch Site

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

Launch Site distance to landmarks

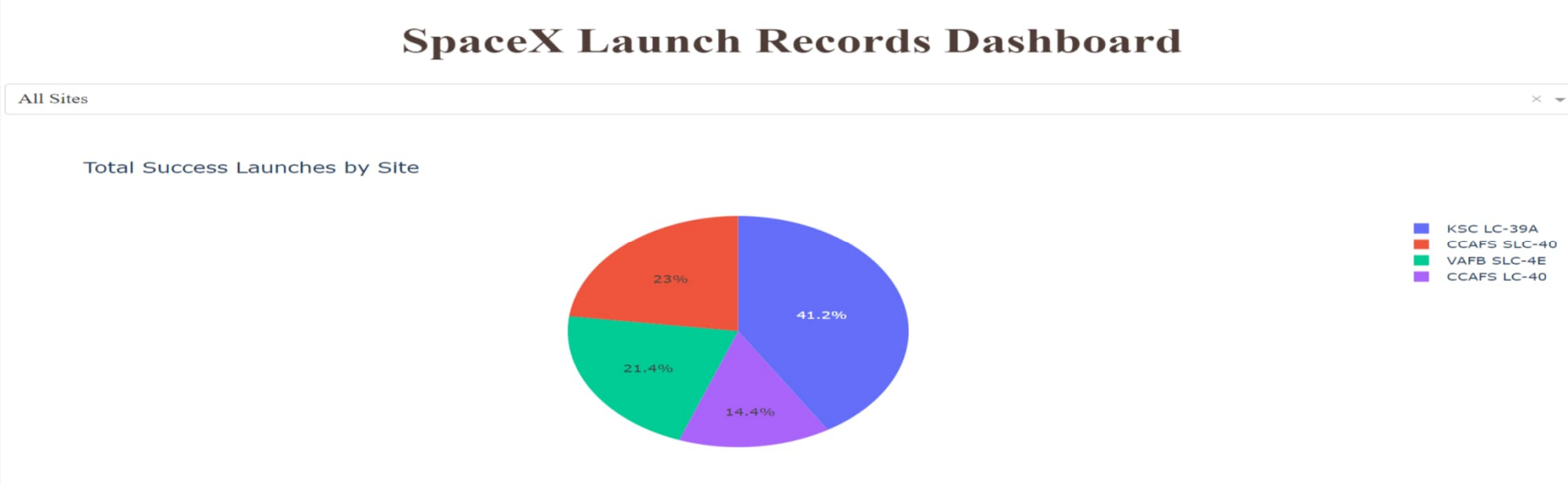


- Are launch sites in close proximity to railways? No
- Are launch sites in close proximity to highways? No
- Are launch sites in close proximity to coastline? Yes
- Do launch sites keep certain distance away from cities? Yes


Section 4
Build a Dashboard with Plotly Dash

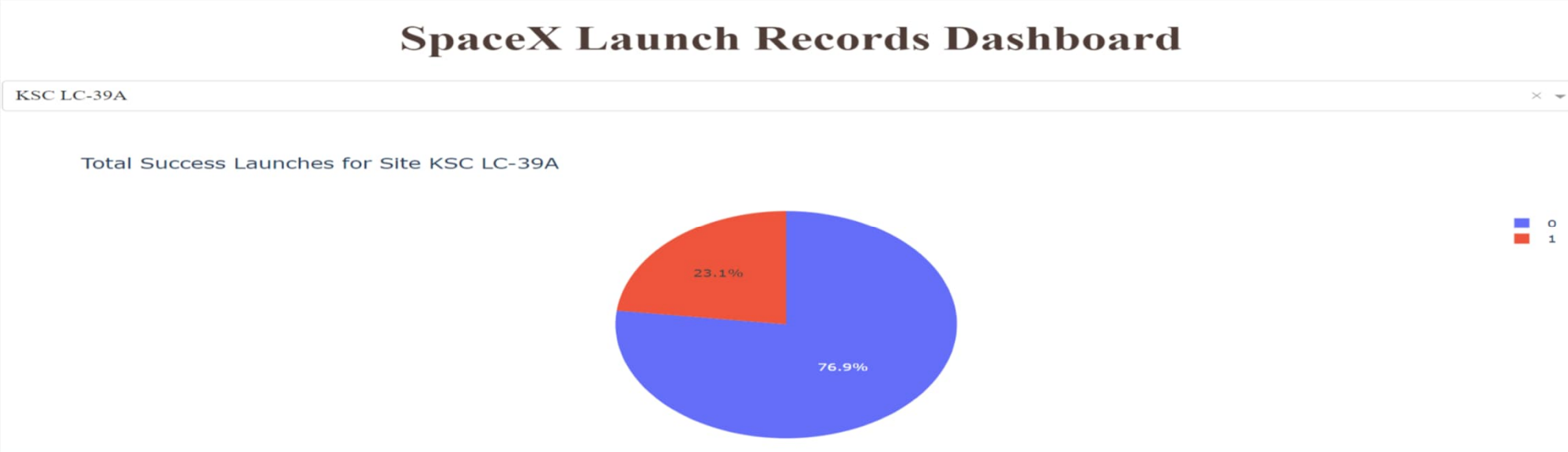
Launch success by site

- **Success as Percent of Total**
- **KSC LC-39A** has the **most successful launches** amongst launch sites (**41.2%**)



Launch Success (KSC LC-29A)

- **Success as Percent of Total**
- **KSC LC-39A** has the **highest success rate** amongst launch sites (**76.9%**)
- 10 successful launches and 3 failed launches
- Explain the important elements and findings on the screenshot



Payload mass and success

By Booster Version

- **Payloads between 2,000 kg** and **5,000 kg** have the **highest success rate**
- 1 indicating successful outcome and 0 indicating an unsuccessful outcome

Section 5

Predictive Analysis (Classification)

Classification Accuracy

Accuracy

- **All** the **models** performed at about the same level and had the **same scores** and **accuracy**. This is likely due to the **small dataset**. The **Decision Tree model slightly outperformed** the restwhenlookingat.best_score_
- .best_score_ is the average of all cv folds for a single combination of the parameters

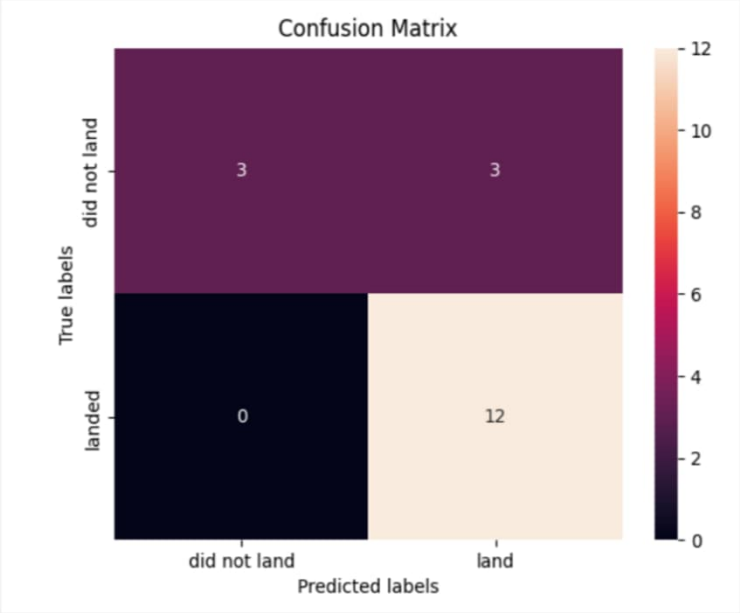
```
models = {'KNeighbors':knn_cv.best_score_,
          'DecisionTree':tree_cv.best_score_,
          'LogisticRegression':logreg_cv.best_score_,
          'SupportVector': svm_cv.best_score_}

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'}
```

	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

Confusion Matrix

- **Performance Summary**
- A confusion matrix summarizes the performance of a classification algorithm
- All the confusion matrices were identical
- The fact that there are false positives (Type 1 error) is not good
- Confusion Matrix Outputs:12 True positive
- 3 True negative
- **3 False positive**
- 0 False Negative



- **Precision**= TP / (TP + FP)12 / 15 = .80
- **Recall**= TP / (TP + FN)12 / 12 = 1
- **F1 Score**= 2 * (Precision * Recall) / (Precision + Recall)2 * (.8 * 1) / (.8 + 1) = .89
- **Accuracy**= (TP + TN) / (TP + TN + FP + FN) = .833

Conclusions

We can conclude that:

- 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.

Thank you!