IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

<Manuel Henkes>
<17/04/2023>

Outline

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

Executive Summary

- Summary of methodologies

To maximize the company's profit, SpaceY hires data scientists to analyze successful rocket landings as they offer potential to be reused. Methods used are

- Collection of data using SpaceX REST API & web scraping techniques
- Wrangling of data to determine a landing outcome variable for each landing
- Exploratory data analysis using SQL, Pandas, and Matplotlib
- Creation of an interactive dashboard with folium and visualization of data using ploty dash to understand the importance of other factors such as payload, launch sites, and more
- Machine learning algorithms to determine the best model

- Summary of all results

- Launch success has improved over time and is highest when the landing site is close to the equator and the coast. KSC LC-39A has the highest launch success rate. Orbits ES-L1, GEO, HEO, and SSO never had an unsuccessful landing.

Introduction

- Project background and context
- SpaceX is a leader in the space industry, striving to make travelling through space affordable for individuals. The company found the can reuse the falcon 9 rocket launch's first stage to decrease costs to over one hundred million dollars per rocket launch. Therefore, data scientists are hired to maximize the successful landing outcome of a rocket's first stage.
- Challenges
- Where can reliable data be retrieved?
- Which parameters are the most relevant to determine successful first stage landing?
- Which predictive model serves the scenario best?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was collected from online sources using REST API and web scraping techniques.
- Perform data wrangling
  - Filtering data, handling missing values, applying one-hot encoding, dropping columns and analyzing data frames to prepare for further analysis and modelling.
- 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

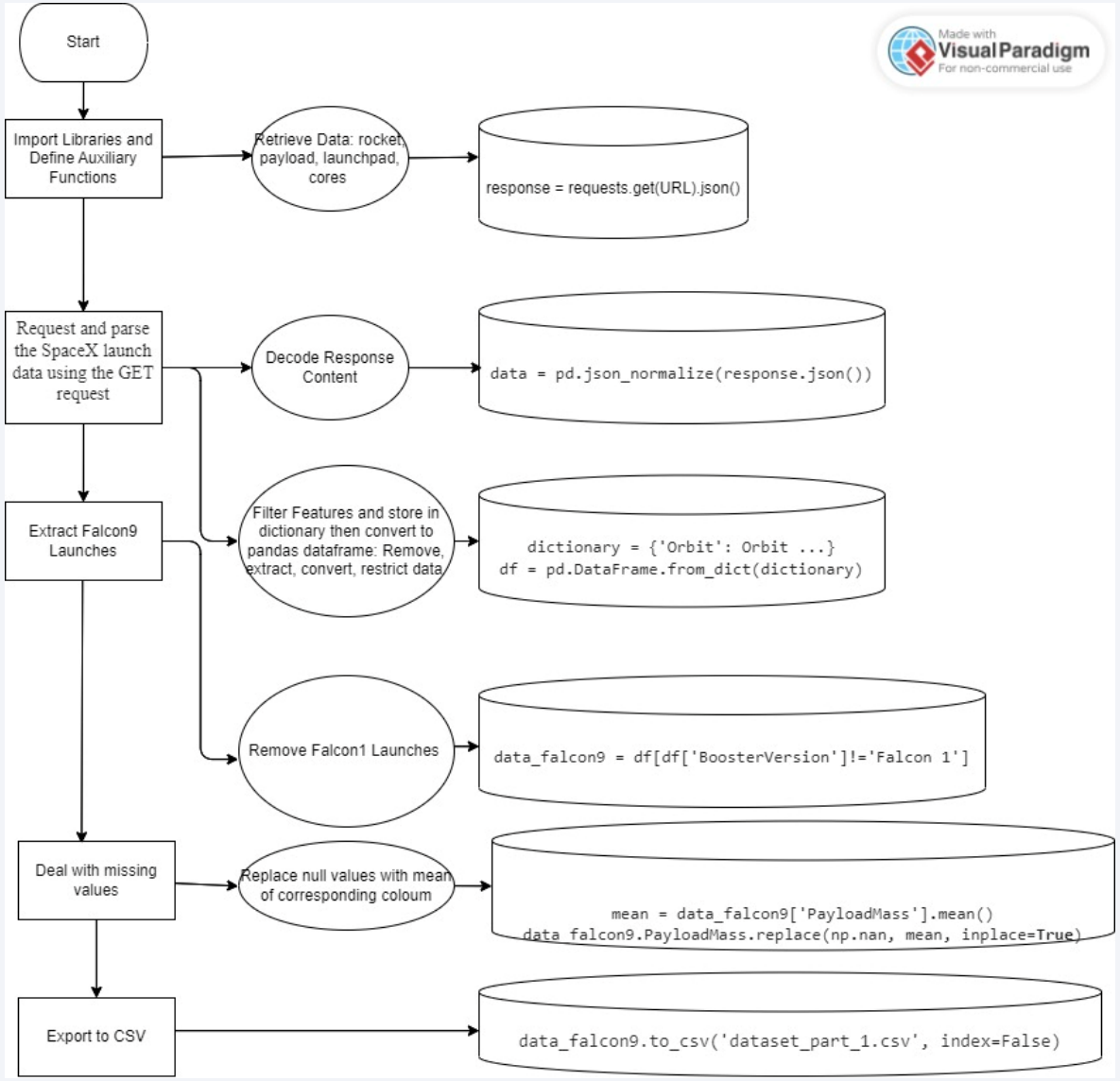
- Steps how data sets were collected:

1) Request and parse the SpaceX launch data using the GET request
    1) Convert json result to Pandas dataframe
    2) extract relevant features into a dictionary (flight number, date, booster version, payload mass, orbit, launch site, outcome etc)
2) Filter the dataframe to contain only Falcon 9 launches
3) Deal with missing values
    1) Replace empty values with mean values
4) Export data to CSV

Data Collection – SpaceX API

GitHub Notebook:

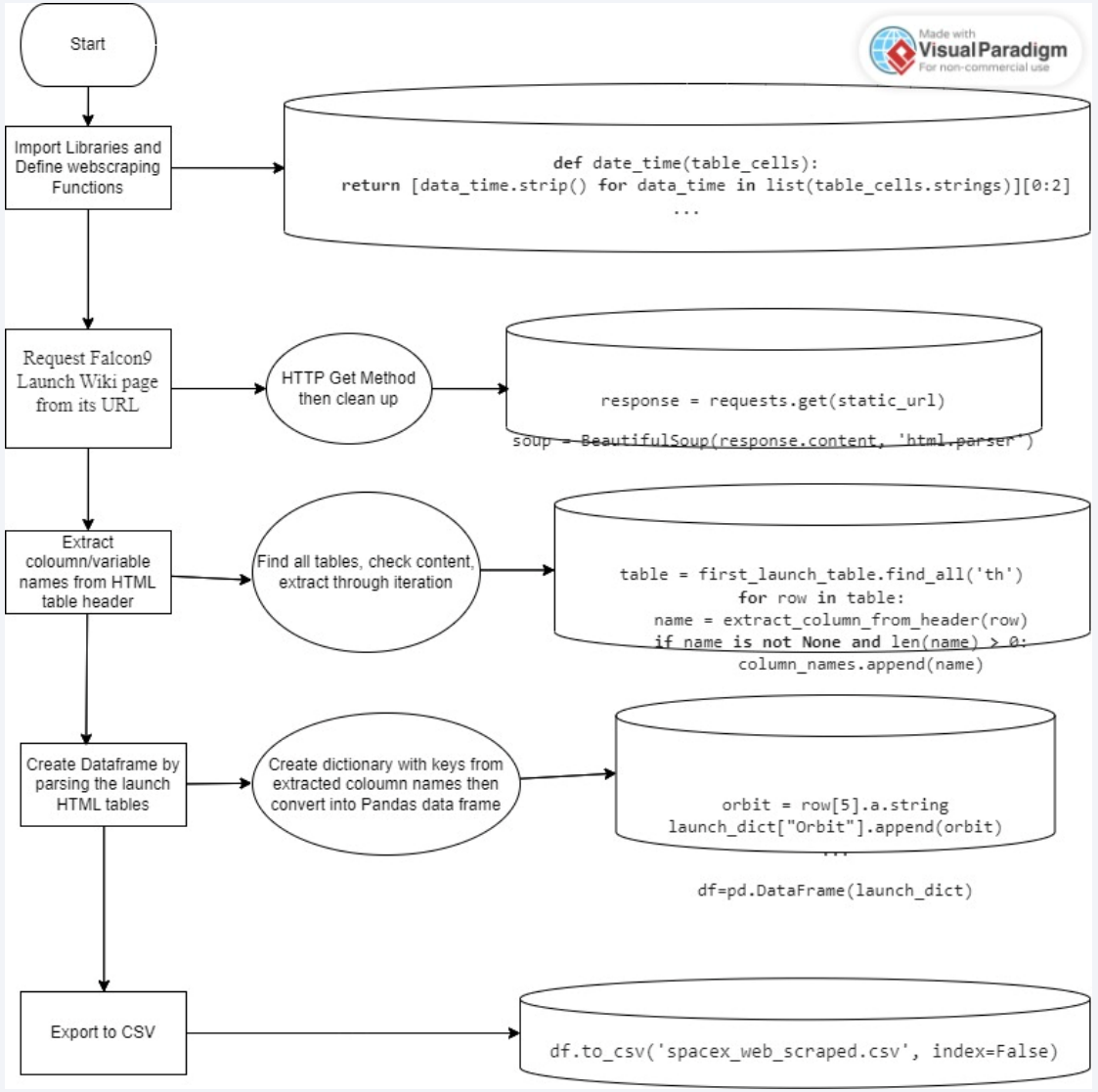
https://github.com/mhenkes92/IBM-Data-Science-Professional/blob/332cea1e420f0e9d25c24f678e7aab759f803f60/Data%20Science%20Capstone%2001%20-%20Data%20Collection%20API%20Lab.ipynb



Data Collection - Scraping

- GitHub Notebook:

https://github.com/mhenkes92/IBM-Data-Science-Professional/blob/332cea1e420f0e9d25c24f678e7aab759f803f60/Data%20Science%20Capstone%2002%20-%20Webscraping.ipynb



Data Wrangling

Exploratory Data Analysis & Training Label determination:

GitHub: https://github.com/mhenkes92/IBM-Data-Science-Professional/blob/332cea1e420f0e9d25c24f678e7aab759f803f60/Data%20Science%20Capstone%2003%20-%20Data%20Wrangling.ipynb

Instructions:

- Import libraries and define auxiliary functions
- Fetch dataset & calculate % missing values per attribute
- Calculate the number of launches on each site
- Calculate number of occurrence of each orbit
- Calculate number and occurrence of mission outcome per orbit type
- Create landing outcome label from Outcome
- Determine Success Rate

Corresponding Code:

- Import … ; def…
- df=pd.read_csv(dataset_part_1_csv) ; df.isnull().sum()/df.shape[0]*100
- df.LaunchSite.value_counts()
- df.Orbit.value_counts()
- landing_outcomes = df.Outcome.value_counts()
- for i,outcome in enumerate(landing_outcomes.keys()): print(i,outcome) ; bad_outcomes=set(landing_outcomes.keys()[[1,3,5,6,7]]) ; landing_class = []
- for j in df['Outcome']:

```
if j in set(bad_outcomes):
    landing_class.append(0)
else:
    landing_class.append(1)
```

- df["Class"].mean()

EDA with Data Visualization

GitHub Notebook: https://github.com/mhenkes92/IBM-Data-Science-Professional/blob/332cea1e420f0e9d25c24f678e7aab759f803f60/Data%20Science%20Capstone%2005%20-%20EDA%20with%20Visualization.ipynb

- To better understand the relationship between features and to determine the best landing outcome, following charts were plotted:
  - Flight Number vs Payload scatter plot showing an increase in flight numbers and lighter payload mass resulting in higher success rates
  - Flight number vs launch sites scatter plot showing different launch sites have different success rates with KSC LC-39A and VAFB SLC 4E having the highest success rates of 77%
  - Launch site vs Payload Mass scatter plot showing the launch site VAFB-SLC doesn't have payloads greater than 10000
  - Class vs orbit bar chart showing that some orbits have a higher success rate than others while an orbit vs flight number and orbit vs payload scatter plot showing that heavy payloads have higher success rates for Polar, LEO, and ISS while flight numbers are unrelated to orbits when in orbit GTO
  - Success Rate vs Date showing success rates increase when date increases

EDA with SQL

- GitHub Notebook: https://github.com/mhenkes92/IBM-Data-Science-Professional/blob/332cea1e420f0e9d25c24f678e7aab759f803f60/Data%20Science%20Capstone%2006%20-%20Data%20Visualization%20with%20Folium.ipynb

Instructions	SQL Queries
Display names of unique launch sites	%sql select distinct Launch_Site from spacextbl;
Display 5 records where launch sites begin with “CCA”	%sql select * from spacextbl where Launch_Site like 'CCA%' limit 5;
Display total payload mass carried by boosterslaunched by NASA CRS	%sql SELECT SUM(PAYLOAD_MASS__KG_) AS total_payload FROM SPACEXTBL WHERE Customer LIKE 'NASA (CRS)';
Display average payload mass carried by booster version F9v1.1	%sql SELECT SUM(PAYLOAD_MASS__KG_) AS total_payload FROM SPACEXTBL WHERE Customer LIKE 'F9 v1.1';
List date when first successful landing outcome in ground pad achieved	%sql SELECT min(date) AS Early_Date from SPACEXTBL where Landing_Outcome LIKE 'Success (ground pad)'
List names of booster with success in drone ship and payload between 4000 and 6000	%sql SELECT DISTINCT Customer, Landing_Outcome,PAYLOAD_MASS__KG_ FROM SPACEXTBL WHERE Landing_Outcome ='Success (drone ship)' AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000;

Build an Interactive Map with Folium

- Markers, circles, and lines were created to understand geographical influence on the landing outcome.
- Circles were used to visualize the coordinate of launch sites, success/failed launches were marked for each launch site, and lines were used to calculate the distance between a launch site to its proximities
- GitHub Notebook: https://github.com/mhenkes92/IBM-Data-Science-Professional/blob/332cea1e420f0e9d25c24f678e7aab759f803f60/Data%20Science%20Capstone%2006%20-%20Data%20Visualization%20with%20Folium.ipynb

Build a Dashboard with Plotly Dash

- Plots and Graphs added to the interactive dashboard are a pie chart and scatter point chart. The interactivity consists of a drop down input component, callback function to render the success vs pie chart based on selected launch sites from the dropdown, range slider to select the payload mass, and callback function to render the success vs payload scatter chart
- The dashboard was created to easily retrieve information about which site has the largest successful launches, highest launch success rate, optimal payload mass, lowest launch success and which booster version has the highest success rate.
- GitHub Notebook: https://github.com/mhenkes92/IBM-Data-Science-Professional/blob/main/Data%20Science%20Capstone%2007%20-%20Interactive%20Dashboard%20Ploty%20Dash.py

Predictive Analysis (Classification)

- After importing libraries and auxiliary functions, dataframe was loaded (data = pd.read_csv(spaceX_dataset.csv)), a NumPy array created from the Class variable in the data and assigned to the target variable Y (Y = data["Class"].to_numpy()). Data is standardized, then reassigned to variable X (X = preprocessing.StandardScaler().fit_transform(X)). Using the train_test_split function the data was split into training (80%) and testing data (20%). Several models were trained, tested,and fitted using GridSearchCV and a dictionary to find the best parameters. Each model was evaluated using the score method.
- The best model to used for this problem is the Decision Tree model because the calculated accuracy is about 83%.
- GitHub Notebook: https://github.com/mhenkes92/IBM-Data-Science-Professional/blob/main/Data%20Science%20Capstone%2008%20-%20Machine%20Learning%20Prediction.ipynb

Results

- Exploratory data analysis results:
  - Launch success rate improved over time
  - KSC LC-39A has the highest success rate
  - Orbits ES-L1, GEO, HEO, SSO have a 100% success rate
  - The lighter the payload the higher the probability of successful landing outcome
- Interactive analytics demo in screenshots
  - Most launch sites are close to the equator and the cost
  - Folium map shows launch sites are isolated enough so no damage within proximity can occur
- Predictive analysis results
  - Decision Tree model is the best predictive model for this dataset

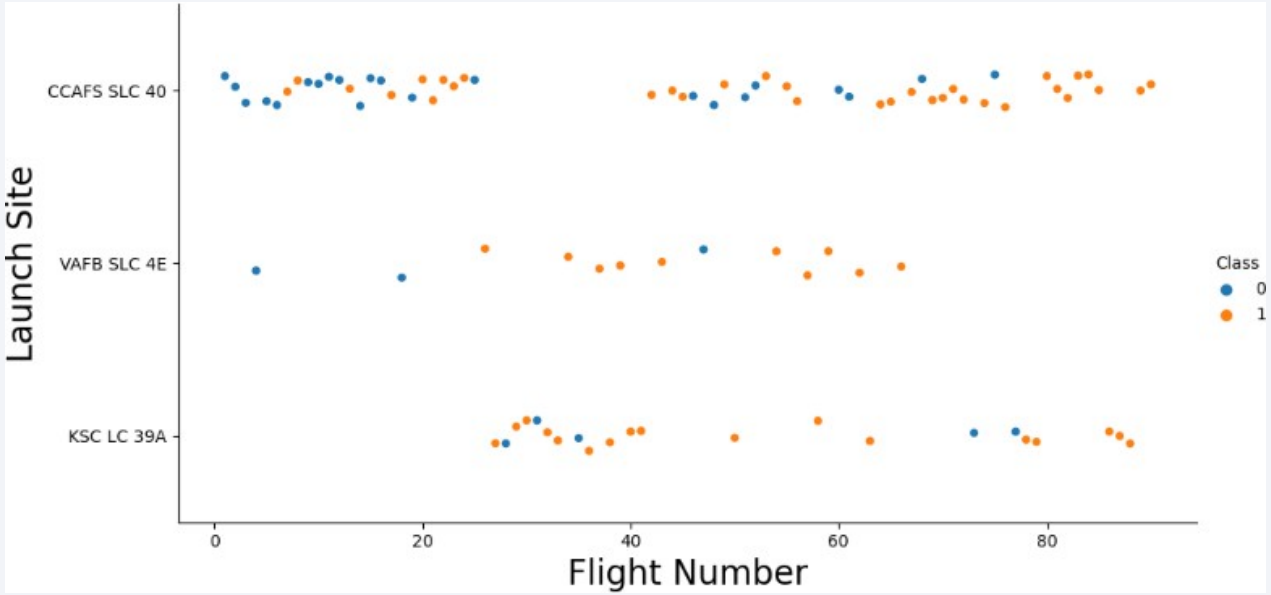
Section 2

Insights drawn from EDA

Flight Number vs. LaunchSite

Exploratory Data Analysis

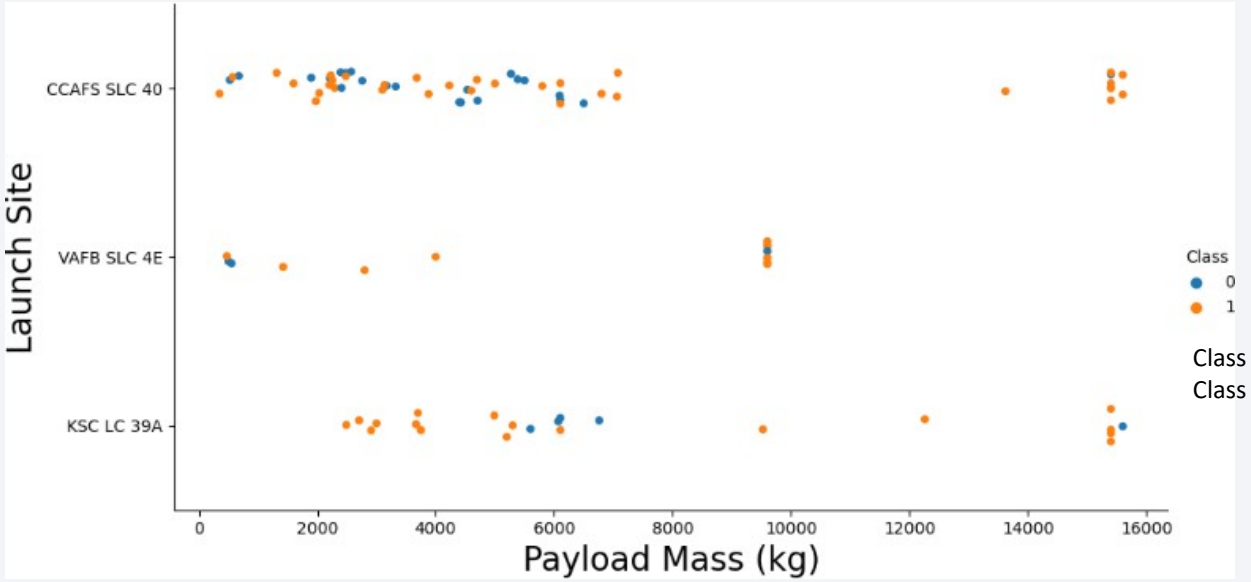
- **Earlier flights** had a **lower success rate** (**blue = fail**)
- **Later flights** had a **higher success rate** (**orange = success**)
- Around half of launches were from CCAFS SLC 40 launch site
- VAFB SLC 4E and KSC LC 39A have higher success rates
- We can infer that new launches have a higher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



Payload vs. LaunchSite

Exploratory Data Analysis

- 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

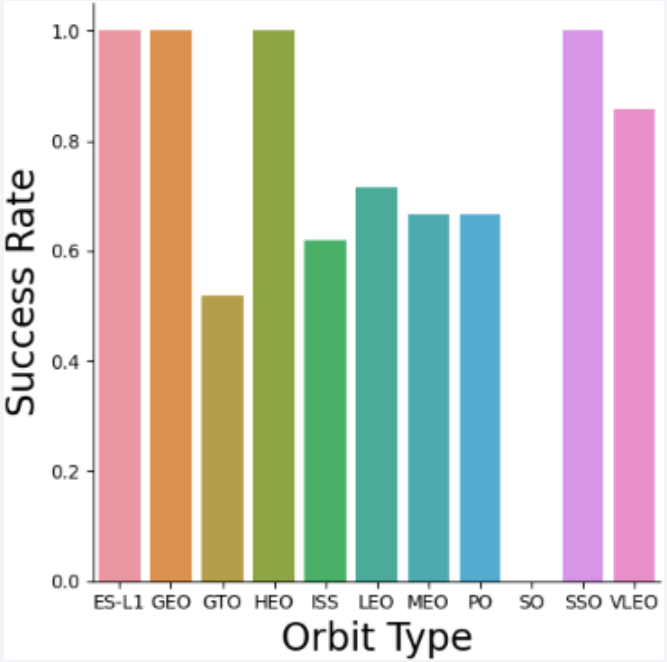


Class 0 = Fail
Class 1 = Success

Success Rate by Orbit

Exploratory Data Analysis

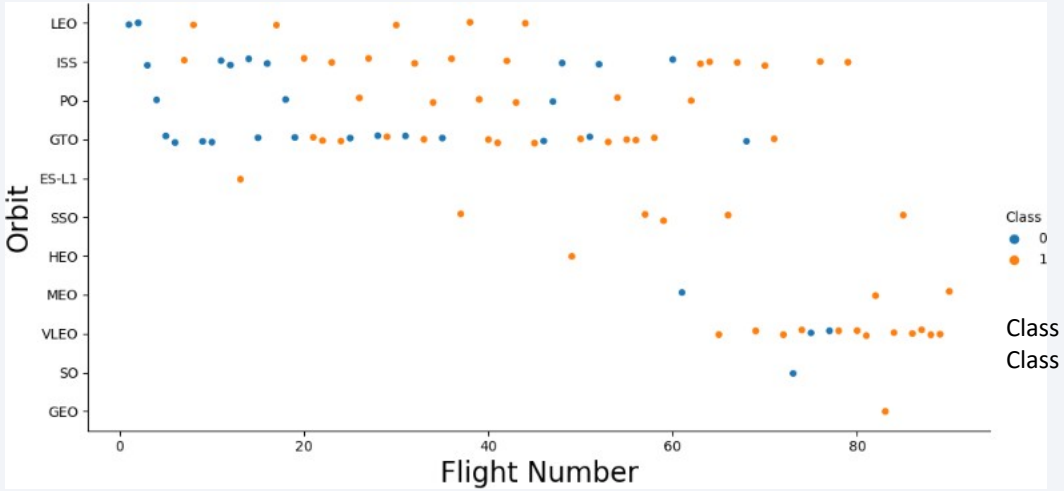
- **100% Success Rate**: ES-L1, GEO, HEO and SSO
- **50%-80% Success Rate**: GTO, ISS, LEO, MEO, PO
- **0% Success Rate**: SO



Flight Number vs. Orbit

Exploratory Data Analysis

- 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

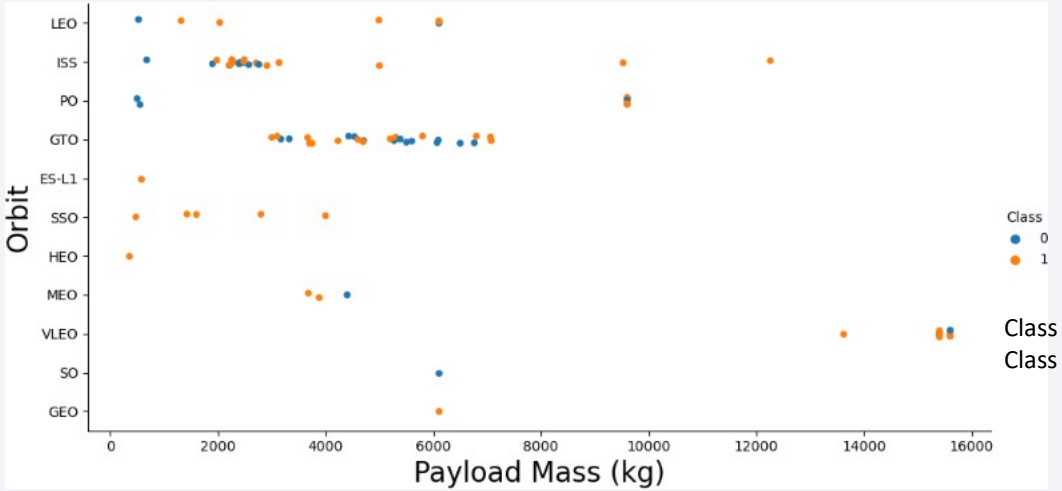


Class 0 = Fail
Class 1 = Success

Payload vs. Orbit

Exploratory Data Analysis

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

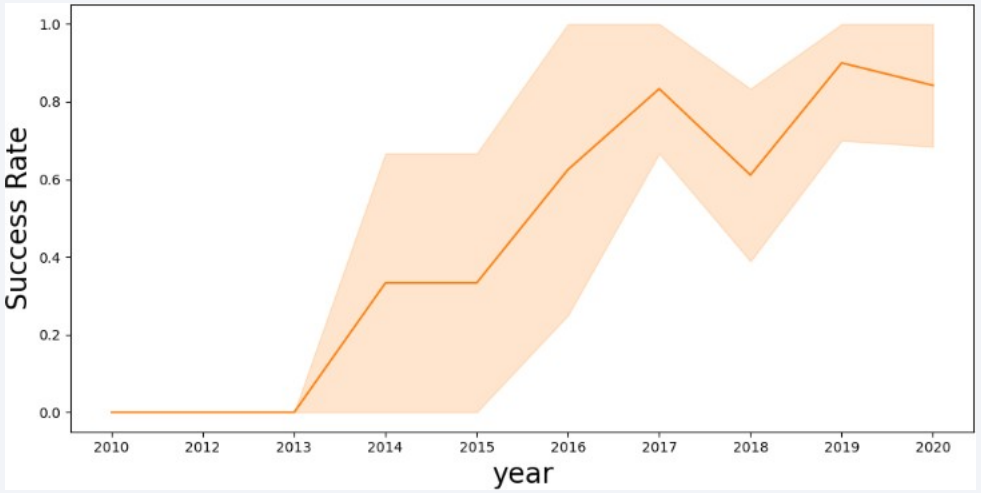


Class 0 = Fail
Class 1 = Success

Launch Success over Time

Exploratory Data Analysis

- 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



Launch Site Information

Launch Site Names

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

Landing Outcome Cont.

```
[30]: %sql ibm_db_sa://yyy33800:dwNKg8J3L0I8d6CP@1bbf73c5
      %sql SELECT Unique(LAUNCH_SITE) FROM SPACEXTBL;

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

[30]:

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

Records with Launch Site Starting with CCA

- Displaying 5 records below

```
%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-
   sqlite:///my_data1.db
Done.
```

1
2928

Landing & MissionInfo

1st Successful Landing in Ground Pad

- 12/22/2015

```
%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

Booster Drone Ship Landing

- Booster mass greater than 4,000 but less than 6,000
- JSCAT-14, JSCAT-16, SES-10, SES-11 / EchoStar 105

```
%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 Failed 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

Carrying Max Payload

- 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

```
%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

Failed Landings on DroneShip

In 2015

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

```
%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)

Count of Successful Landings

Ranked Descending

- Count of landing outcomes between 2010-06-04 and 2017-03-20 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.
```

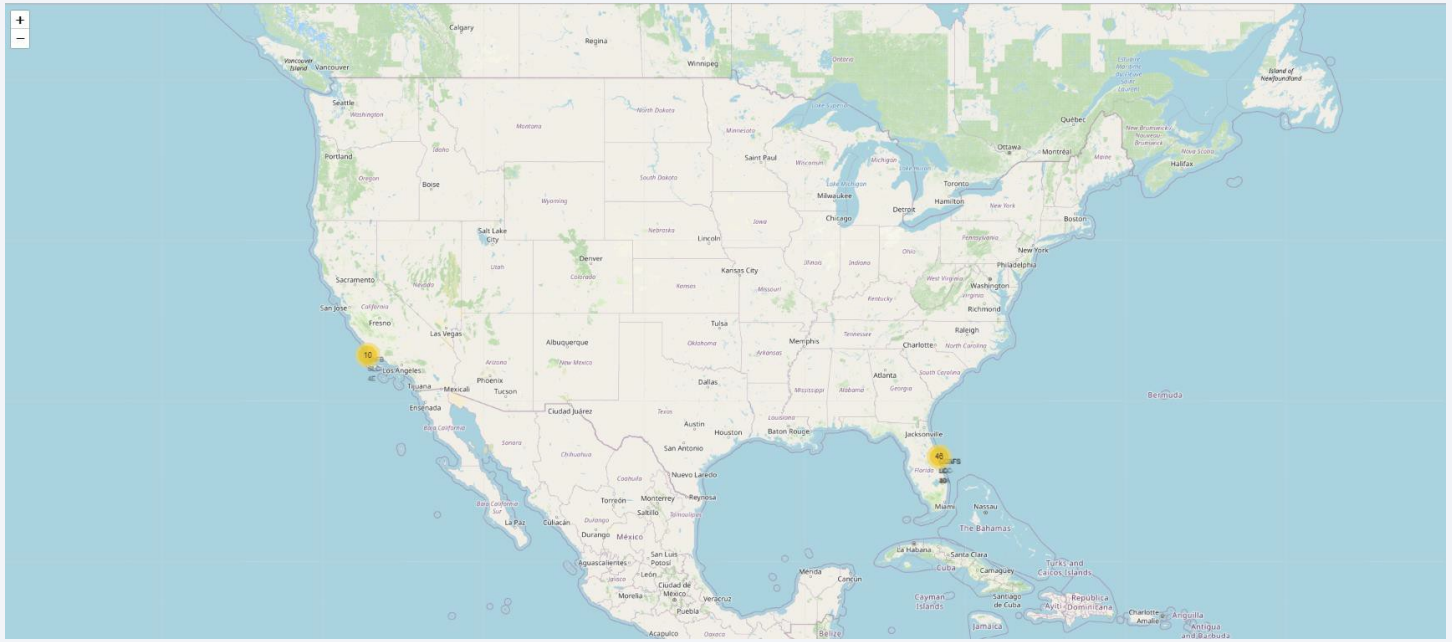
Landing _Outcome	count_outcomes
Success	20
No attempt	10
Success (drone ship)	8
Success (ground pad)	6
Failure (drone ship)	4
Failure	3
Controlled (ocean)	3
Failure (parachute)	2
No attempt	1


Section 3
Launch Sites
Proximities Analysis

Launch Sites

With 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

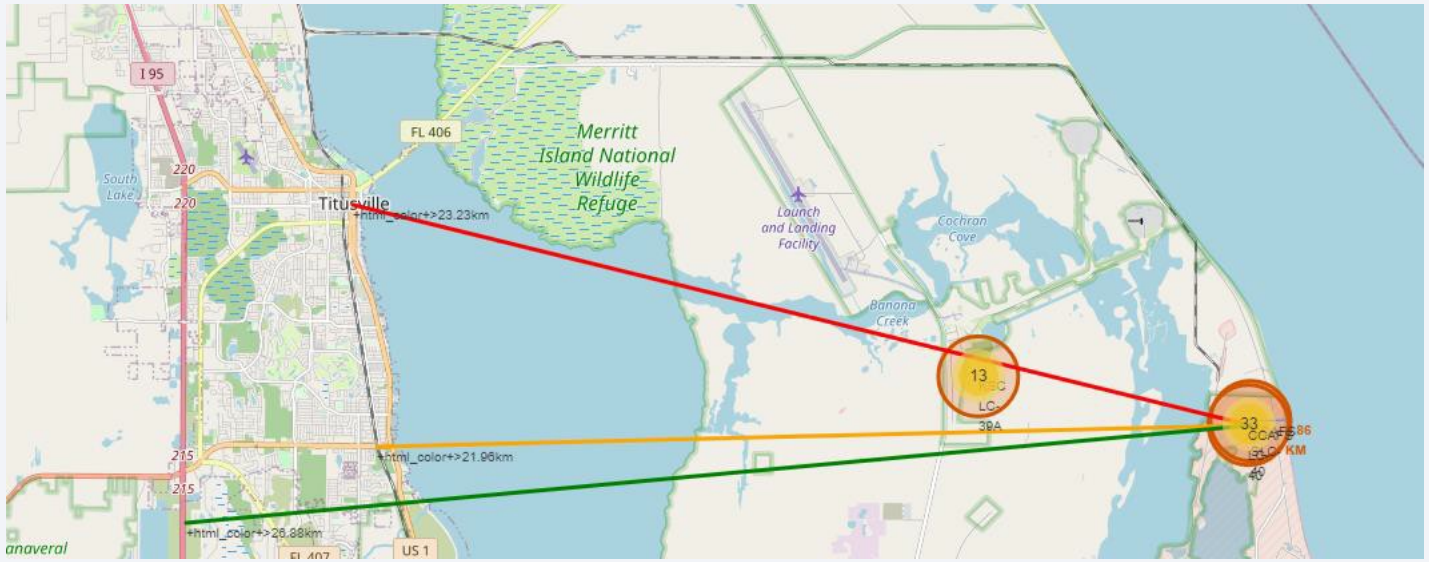
At Each Launch Site

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

Distance to Proximities

CCAFS SLC-40

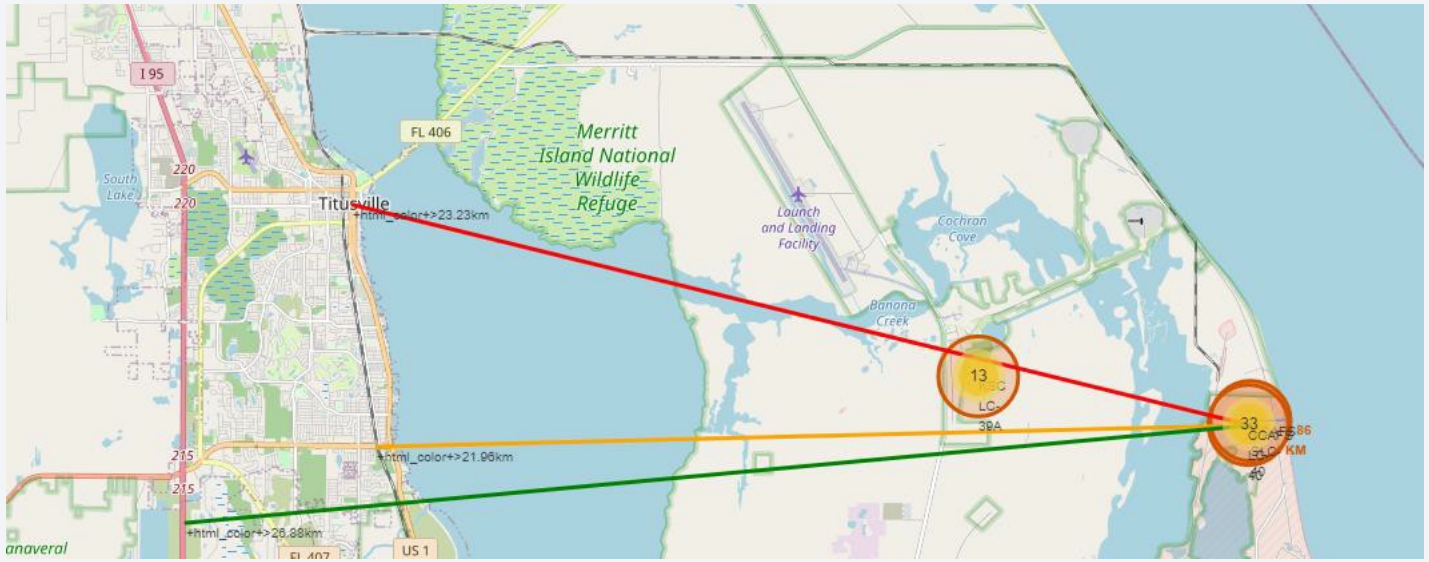
- **.86 km** from nearest coastline
- **21.96 km** from nearest railway
- **23.23 km** from nearest city
- **26.88 km** from nearest highway

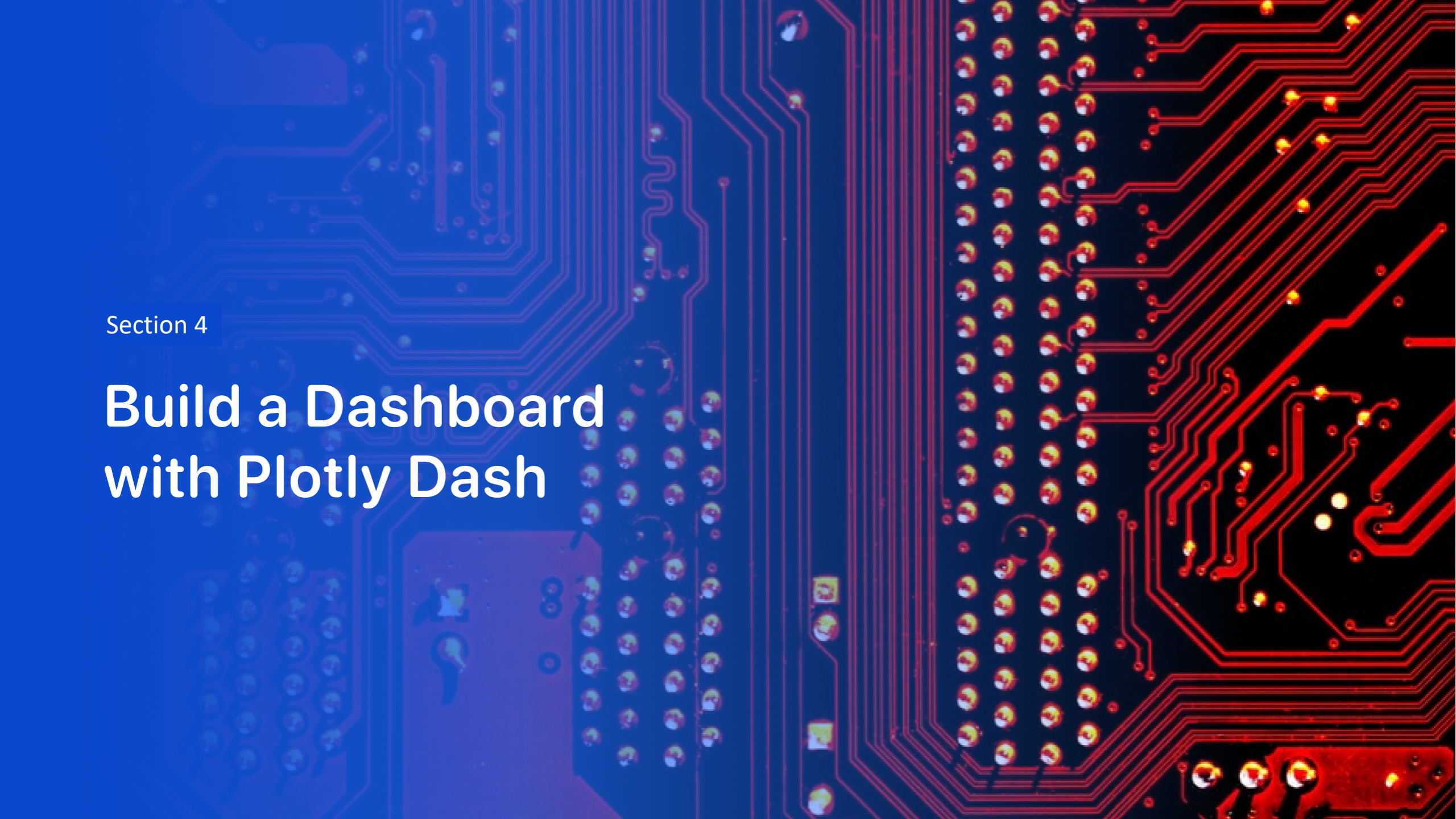


Distance to Proximities

CCAFS SLC-40

- **Coasts**: help ensure that spent stages dropped along the launch path or failed launches don't fall on people or property.
- **Safety / Security:** needs to be an exclusion zone around the launch site to keep unauthorized people away and keep people safe.
- **Transportation/Infrastructure and Cities**: need to be away from anything a failed launch can damage, but still close enough to roads/rails/docks to be able to bring people and material to or from it in support of launch activities.

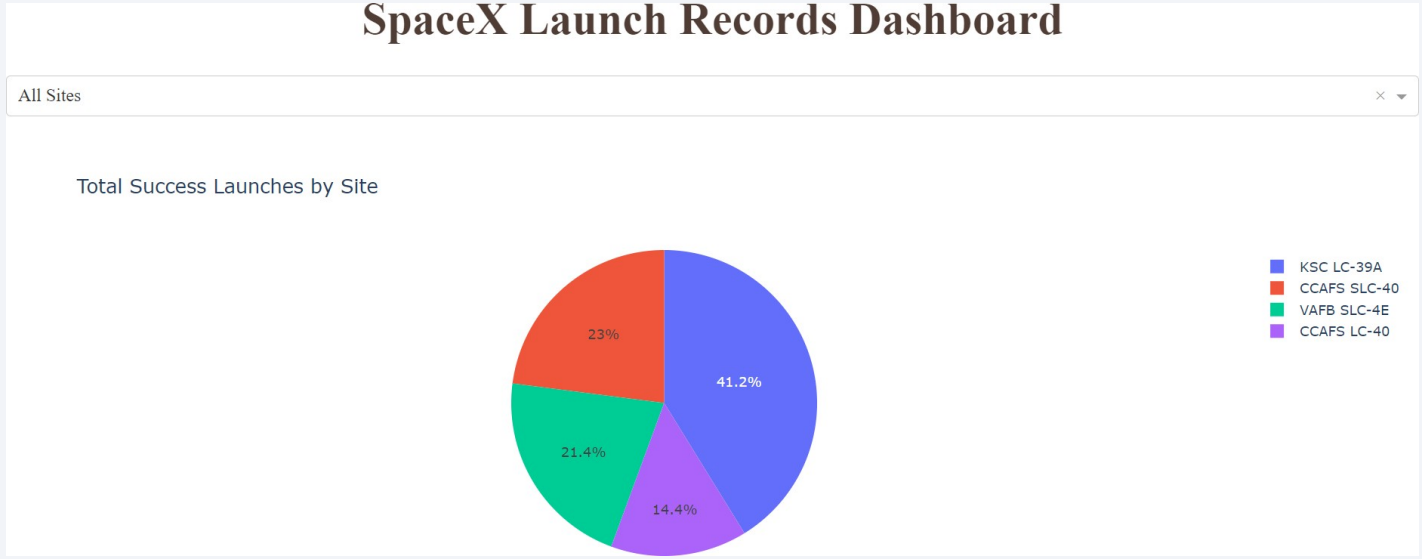



Section 4
Build a Dashboard with Plotly Dash

Launch Success by Site

Success as Percent of Total

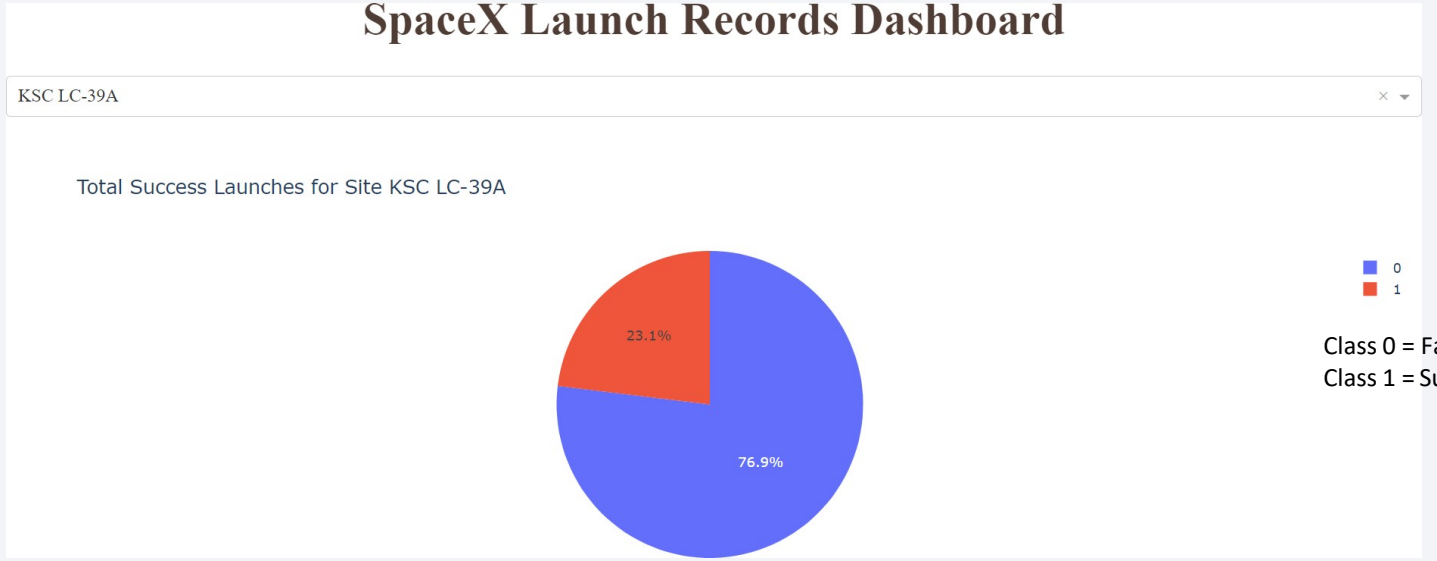
- **KSC LC-39A** has the **most successful launches** amongst launch sites (**41.2%**)



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



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

- **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 rest when looking at .best_score_
- .best_score_ is the average of all cv folds for a single combination of the parameters

	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

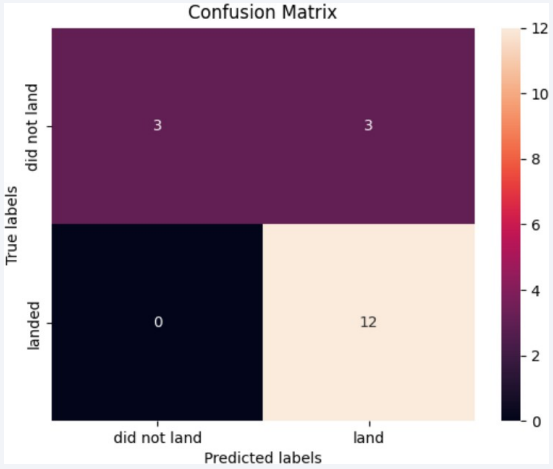
```
models = {'KNeighbors':knn_cv.best_score_,
              'DecisionTree':tree_cv.best_score_,
              'LogisticRegression':logreg_cv.best_score_,
              'SupportVector': svm_cv.best_score_}

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 Matrices

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



Conclusion

Research

- **Model Performance**: The models performed similarly on the test set with the decision tree model slightly outperforming
- **Equator**: Most of the launch sites are near the equator for an additional natural boost - due to the rotational speed of earth - which helps save the cost of putting in extra fuel and boosters
- **Coast**: All the launch sites are close to the coast
- **Launch Success**: Increases over time
- **KSC LC-39A**: Has the highest success rate among launch sites. Has a 100% success rate for launches less than 5,500 kg
- **Orbits**: ES-L1, GEO, HEO, and SSO have a 100% success rate
- **Payload Mass**: Across all launch sites, the higher the payload mass (kg), the higher the successrate

Conclusion

Things to Consider

- **Dataset**: A larger dataset will help build on the predictive analytics results to help understand if the findings can be generalizable to a larger data set
- **Feature Analysis / PCA**: Additional feature analysis or principal component analysis should be conducted to see if it can help improve accuracy
- **XGBoost**: Is a powerful model which was not utilized in this study. It would be interesting to see if it outperforms the other classification models

Thank you!