IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

<Name>
<Date>

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
  - Data exploration and analysis
  - Interactive visualizations and analysis
  - Predictive analytics results

Introduction

- Background

  SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upwards of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage resulting in significant savings and therefore significantly lower prices.

- Problem statement

  - Is it possible to to use use publicly available Space X data and machine learning to predict the likelihood Space X successfully landing their first stage rockets and therefore confirm their costs savings.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Space X public API and Webscrape of Wikipedia
- Perform data wrangling
  - One-hot encoding of 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 – SpaceX API

1. Ran data collection via GET request from SpaceX API
2. Decoded response as a Json using .json() function then converted to pandas dataframe using .json_normalize()
3. Cleaned data, checked for missing values and filled in missing values and exported to csv

- GitHub URL to API data collection notebook: https://github.com/brainardp/ds-capstone/blob/main/1-Data%20Collection%20API.ipynb

Data Collection – Web Scraping

1. Web scraped Wikipedia Falcon 9 launch records with BeautifulSoup to extract the launch records as an HTML table
2. Converted table to a pandas dataframe
3. Exported results to csv

- GitHub URL to web scraping notebook: https://github.com/brainardp/ds-capstone/blob/main/2-Data%20Collection%20with%20Web%20Scraping.ipynb

Data Wrangling

1. Performed exploratory data analysis of SpaceX data and identified training labels
2. Calculated number of launches at each site and orbit
3. Created landing outcome label from outcome column and exported the results to csv

- GitHub URL of data wrangling notebook: https://github.com/brainardp/ds-capstone/blob/main/2-Data%20Collection%20with%20Web%20Scraping.ipynb

EDA with Data Visualization

1. Visualized the relationship between flight number and launch site;
2. Payload and launch site;
3. Success rate of each orbit type;
4. Flight number and orbit type;
5. Payload and orbit type;
6. And launch success yearly trend

GitHub URL of completed EDA with data visualization notebook:
https://github.com/brainardp/ds-capstone/blob/main/4-jupyter-labs-eda-dataviz.ipynb

EDA with SQL

- Pulled SpaceX data residing in DB2 database into Jupyter notebook using Db2 Magic to conduct exploratory data analysis with SQL:
    1. The names of unique launch sites in the space mission
    2. The total payload mass carried by boosters launched by NASA (CRS)
    3. The average payload mass carried by booster version F9 v1.1
    4. When the first successful landing outcome in ground pad was achieved
    5. Listed names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
    6. Listed total number of successful and failure mission outcomes
    7. Ranked the count of landing outcomes.

- GitHub URL of completed EDA with SQL notebook: https://github.com/brainardp/ds-capstone/blob/main/5-EDA%20with%20SQL.ipynb

Build an Interactive Map with Folium

1. All launch sites were marked using folium map objects (markers, circles, and lines) to mark the success or failure of launches for each site on the folium map.
2. Feature launch outcomes (failure or success) were assigned to class 0 (failure) and 1 (success).
3. Used the color-labeled marker clusters to identify launch sites with high success rates.
4. Calculated the distances between a launch site to railways and coast lines

- GitHub URL of your completed interactive folium map: https://github.com/brainardp/ds-capstone/blob/main/6-Interactive%20Visual%20Analytics%20with%20Folium.ipynb

Build a Dashboard with Plotly Dash

- Created interactive dashboard with Plotly dash with the following charts:
  - Pie chart depicting launch success by site
  - Scatter chart to correlate success rate between payload mass and booster version

- GitHub URL of your completed Plotly Dash lab: https://github.com/brainardp/ds-capstone/blob/main/7-spacex_dash_app.py

Predictive Analysis (Classification)

1. Loaded data using numpy and pandas, transformed the data, and split our data into training and testing.
2. Built different machine learning models and tuned different hyperparameters using GridSearchCV.
3. Calculated model accuracy using score method
4. Compared scores to identify the best performing model

- GitHub URL of your completed predictive analysis lab: https://github.com/brainardp/ds-capstone/blob/main/8-SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results

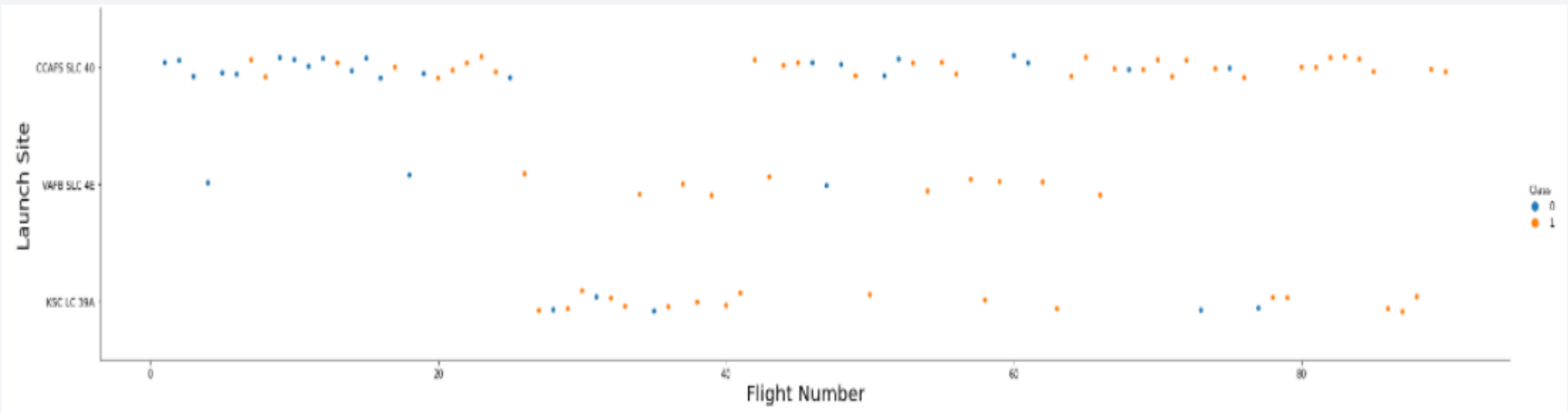
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

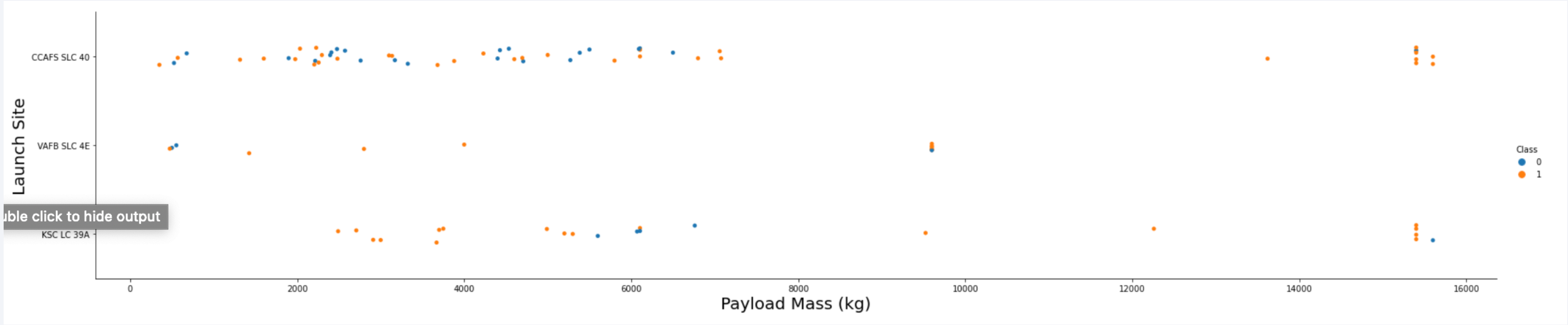
Flight Number vs. Launch Site

The more flights, the greater the success rate at a launch site.



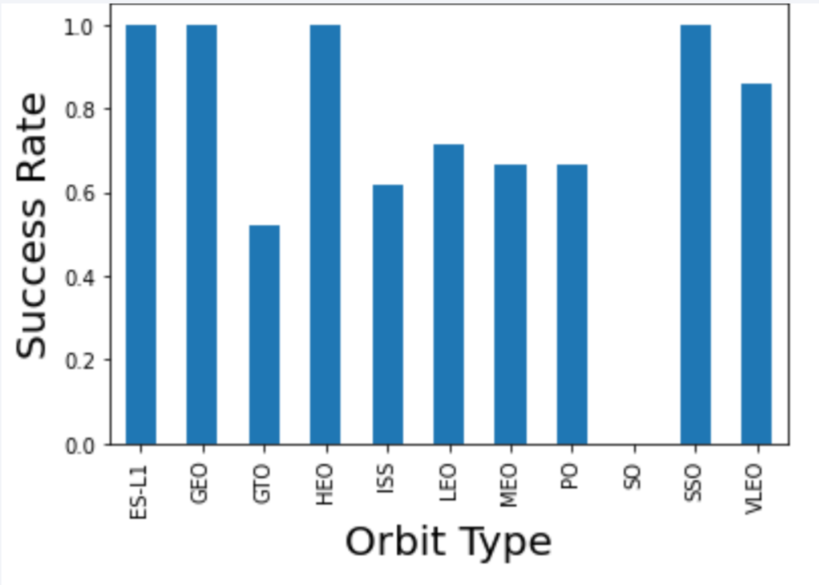
Payload vs. Launch Site

- CCAFS SLC 40 had the higher success rate at higher payload weight



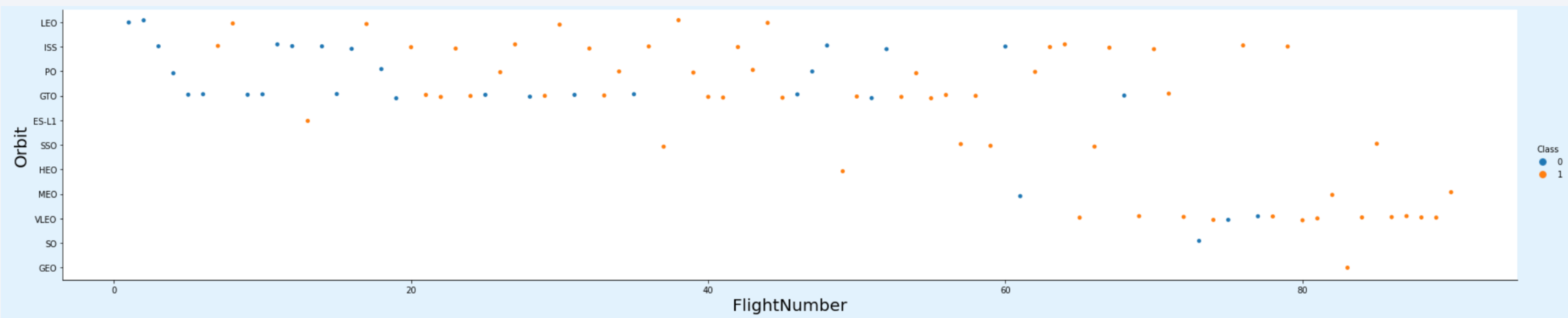
Success Rate vs. Orbit Type

- ES-L1, GEO, HEO and SSO had the highest success rate



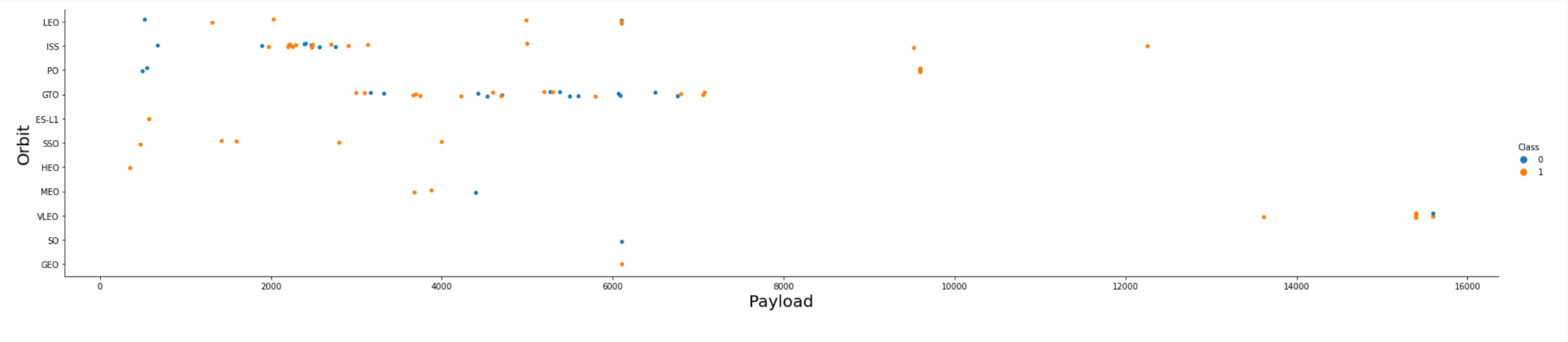
Flight Number vs. Orbit Type

- Success rate improves at higher flight numbers for all orbit types



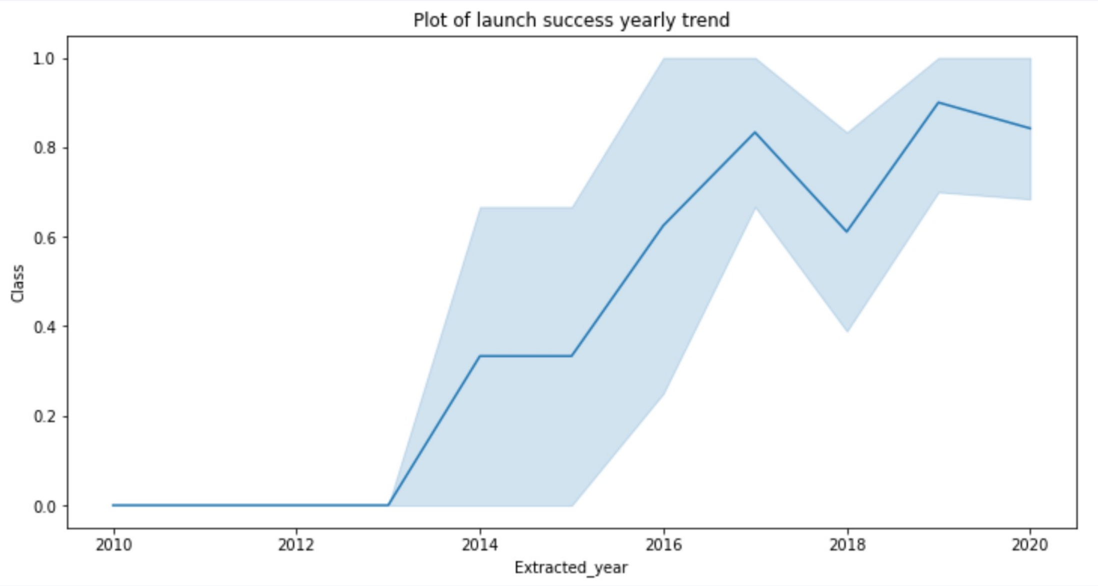
Payload vs. Orbit Type

- At heavier payloads, there are more successful landings for PO, LEO and ISS orbits



Launch Success Yearly Trend

- The success rate significantly increasing from 2013 until 2020



All Launch Site Names

- Used select distinct from SpaceX dataset to display list of their launch sites

```
%sql select distinct launch_site FROM SPACEXTBL
```

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

Launch Site Names Begin with 'CCA'

- Queried data to find 5 records where launch sites begin with `CCA`

```
%sql select launch_site as cca_launchsite from SPACEXTBL where launch_site like 'CCA%' order by DATE asc limit 5;
```

Out[4]:

cca_launchsite
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40

Total Payload Mass

- Calculated the total payload carried by boosters from NASA as 45596

In [5]:

```
%sql select sum(PAYLOAD_MASS__KG_) as sum_nasa_payload from SPACEXTBL where CUSTOMER = 'NASA (CRS)'
```

Out[5]:

sum_nasa_payload
45596

Average Payload Mass by F9 v1.1

- Calculated the average payload mass carried by booster version F9 v1.1 as 2928

In [6]:

```
%sql select avg(PAYLOAD_MASS__KG_) as avg_F9_payload_mass from SPACEXTBL where BOOSTER_VERSION = 'F9 v1.1'
```

Out[6]:

avg_f9_payload_mass
2928

First Successful Ground Landing Date

- Date of the first successful landing outcome on ground pad

In [7]:

```
%sql select min(DATE) as first_sucessfull_ground_landing from SPACEXTBL where LANDING__OUTCOME = 'Success (ground pad)'
```

Out[7]:

first_sucessfull_ground_landing
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

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

In [8]:

```
%sql select BOOSTER_VERSION from SPACEXTBL where LANDING__OUTCOME = 'Success (drone ship)' and PAYLOAD_MASS__KG_ > 4000 and PAYLOAD_MASS_
```

Out[8]:

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

Total Number of Successful and Failure Mission Outcomes

- Total number of successful and failure mission outcomes

In [9]:

```
%sql select MISSION_OUTCOME, count(*) count from SPACEXTBL group by MISSION_OUTCOME
```

Out[9]:

mission_outcome	COUNT
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass

In [10]:

```
%sql select BOOSTER_VERSION as maxed_boosters from SPACEXTBL where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEXTBL)
```

Out[10]:

maxed_boosters
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

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

In [12]:

```
%sql select date, LANDING__OUTCOME as failed_landings, BOOSTER_VERSION as booster, launch_site from SPACEXTBL \
where LANDING__OUTCOME = 'Failure (drone ship)' AND date like '2015%'
```

```
 * ibm_db_sa://zzr40941:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31
Done.
```

Out[12]:

DATE	failed_landings	booster	launch_site
2015-01-10	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
2015-04-14	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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

- Landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, ranked in descending order

In [30]:

```
%sql SELECT LANDING__OUTCOME, COUNT(*) AS OUTCOMES FROM SPACEXTBL WHERE date > '2010-06-04' AND date < '2017-03-20' GROUP BY LANDING__OUTC
```

```
 * ibm_db_sa://zzr40941:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/bludb
Done.
```

Out[30]:

landing__outcome	outcomes
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Uncontrolled (ocean)	2
Failure (parachute)	1
Precluded (drone ship)	1

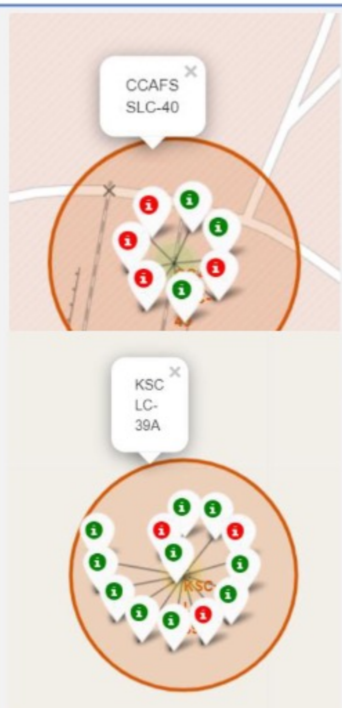
Section 3

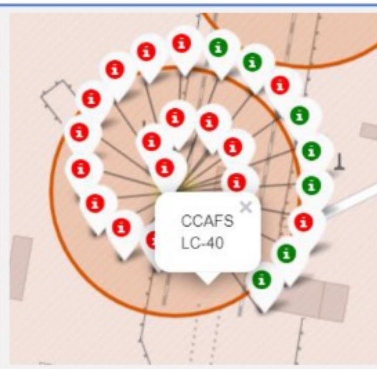
Launch Sites Proximities Analysis

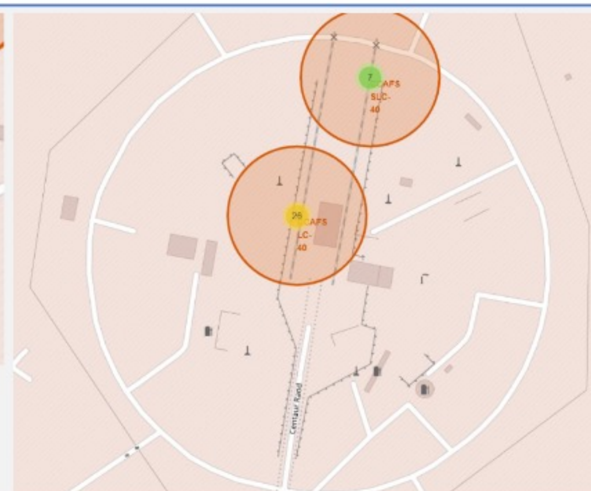
SpaceX Launch Sites

- VAFB SLC 4E is located in California and KSC LC, CCASF SLC-40 and CCASF LC-40 are in Florida

Markers showing SpaceX Launches by Site







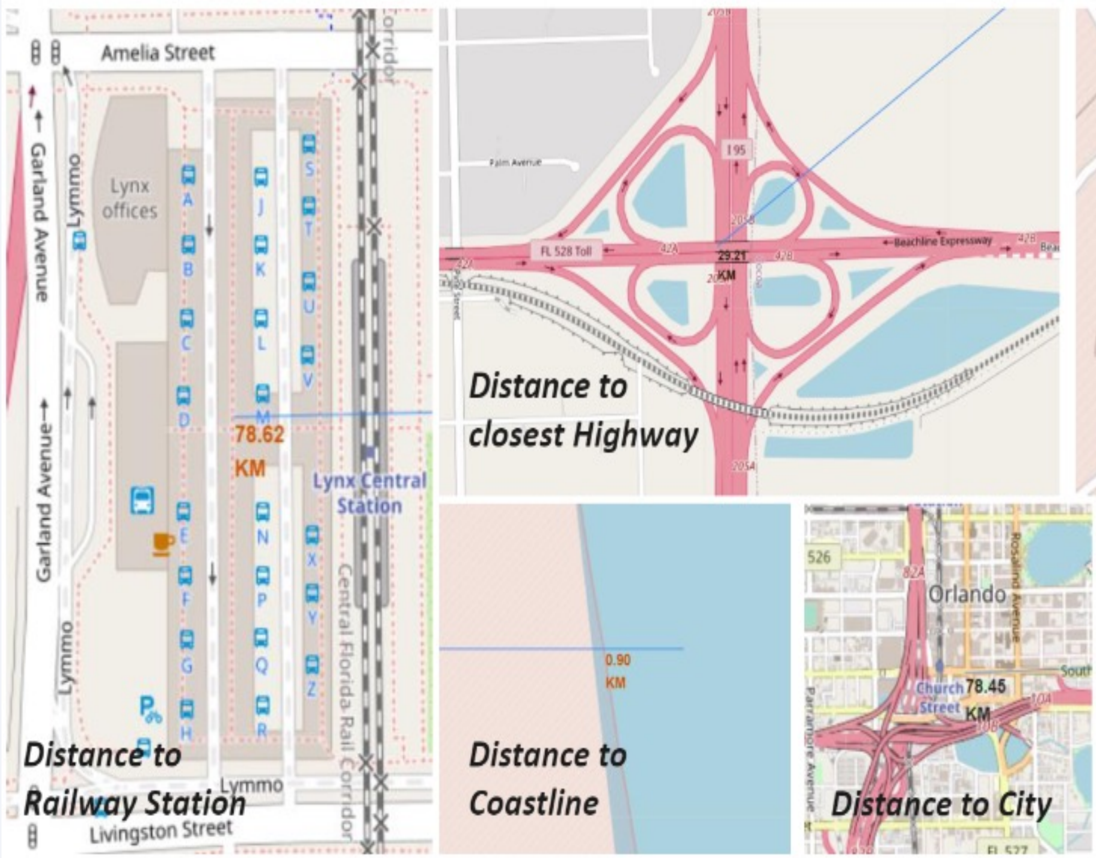
Florida Launch Sites

Green Marker shows successful Launches and Red Marker shows Failures

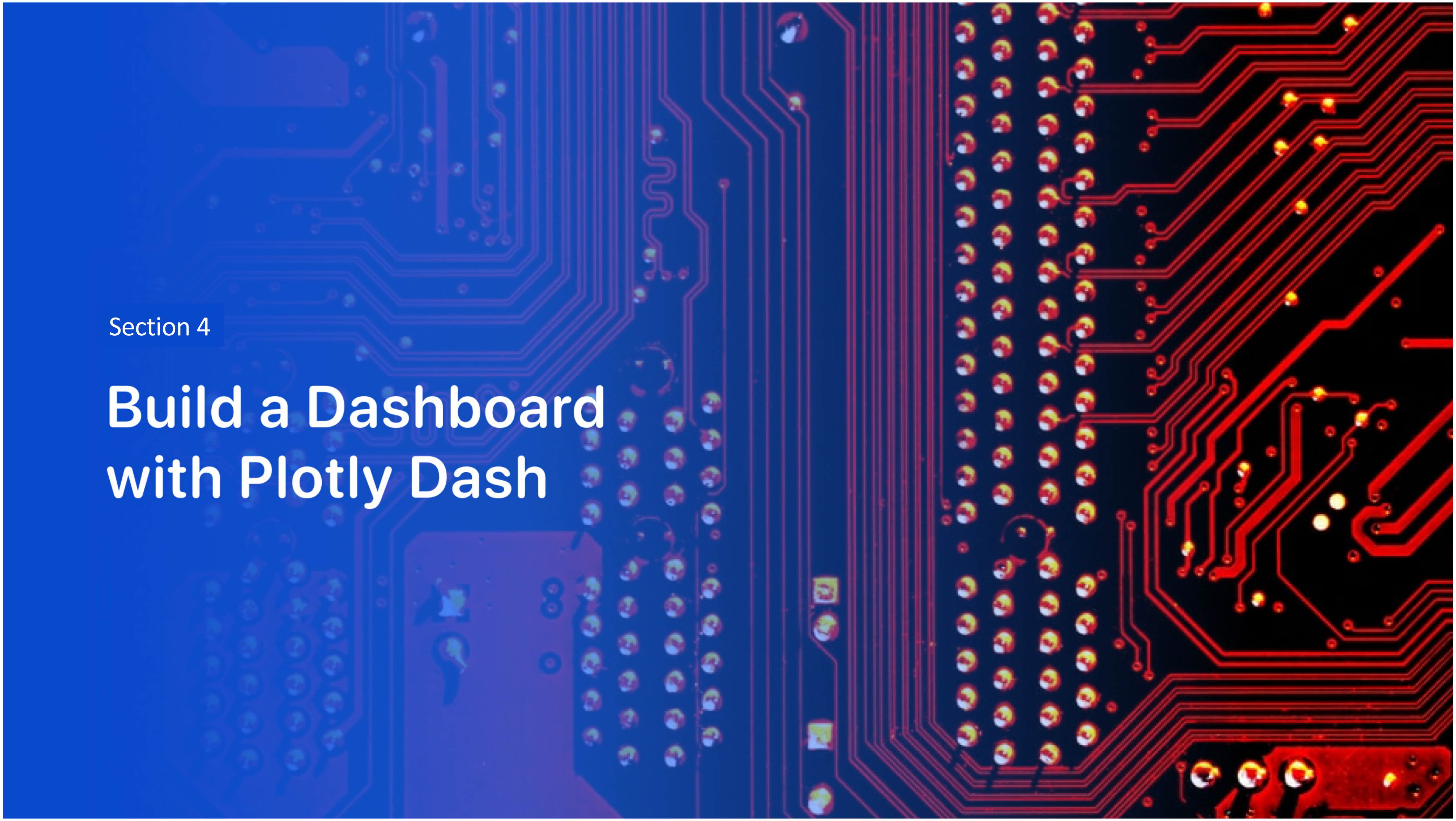
California Launch Site

37

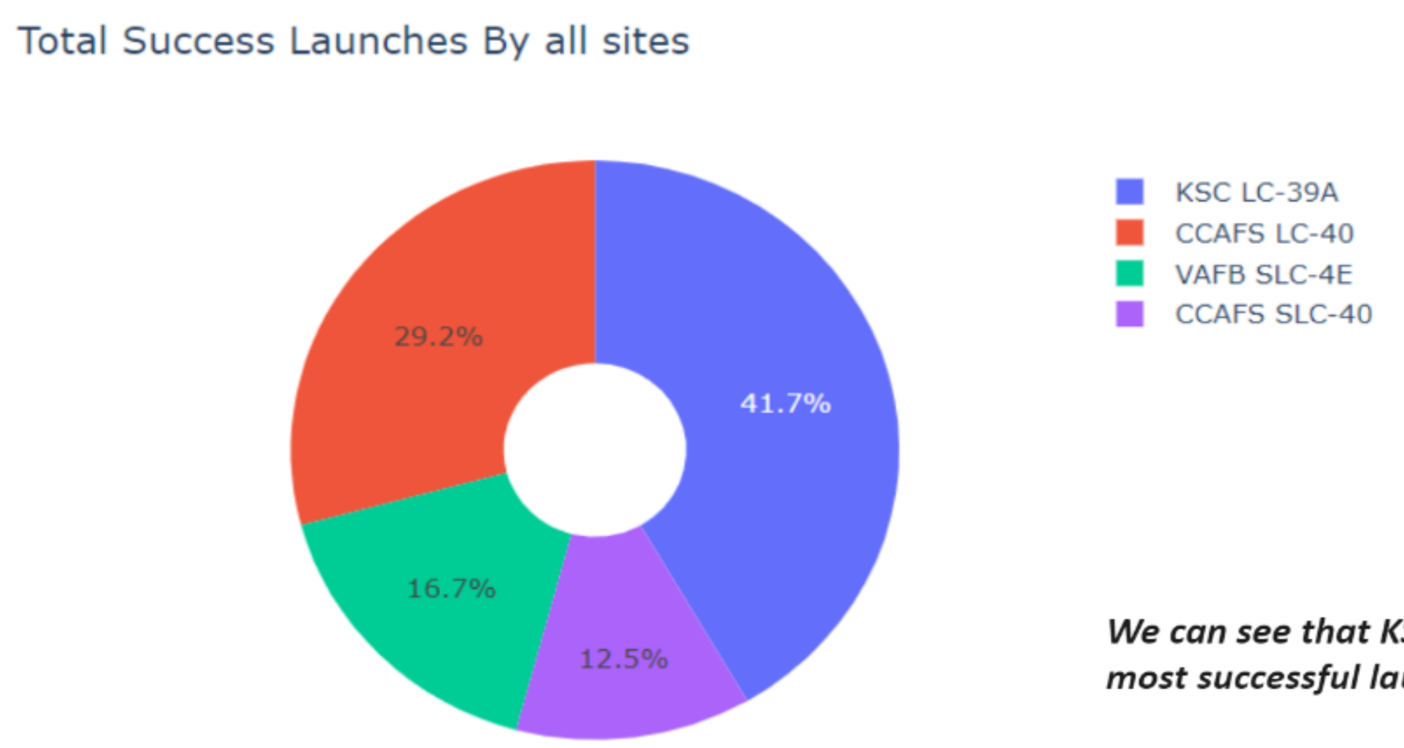
Launch Site Distance from Landmarks



•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

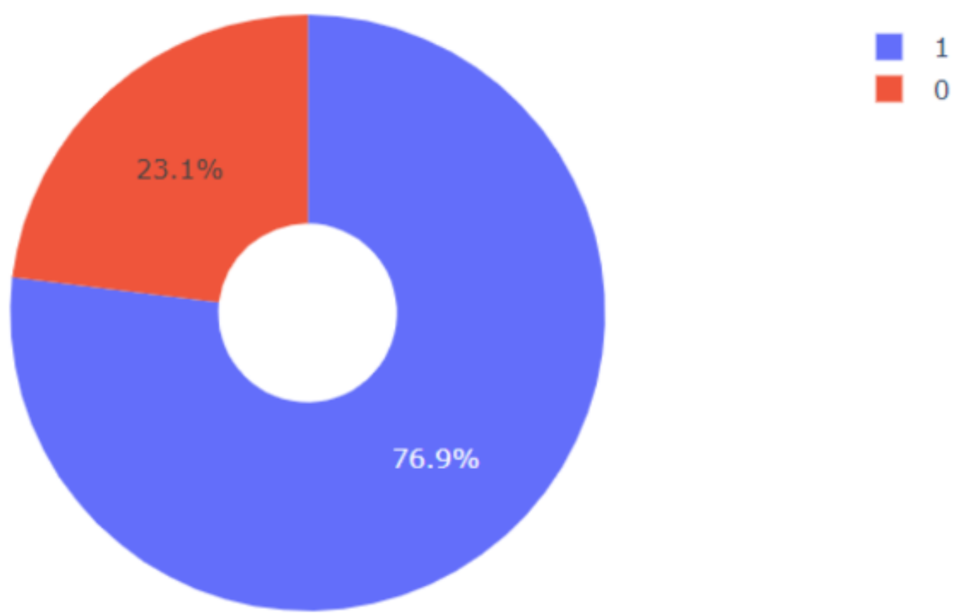

Section 4
Build a Dashboard
with Plotly Dash

Launch Success Rate by Site



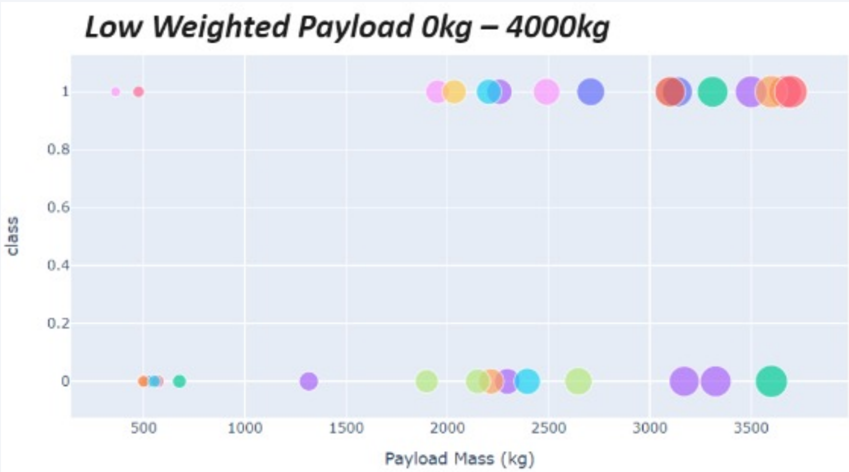
We can see that KSC LC-39A had the most successful launches from all the sites

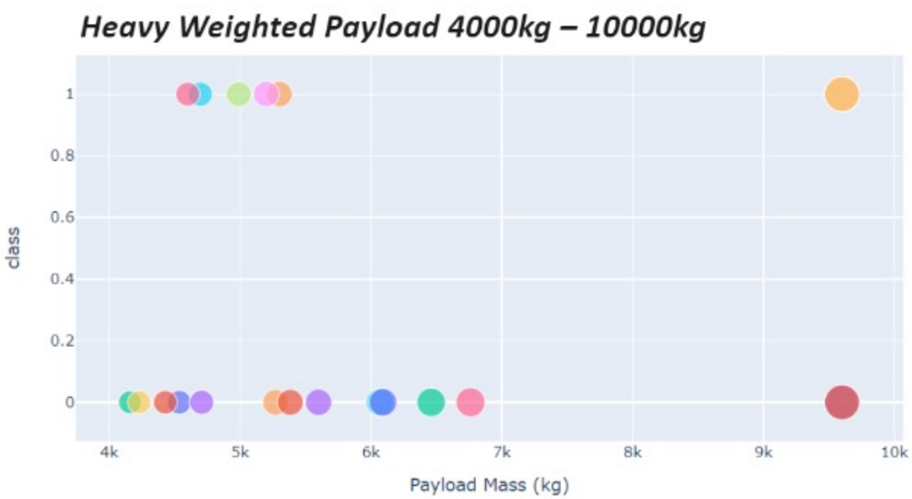
Site with the Highest Launch Success Rate



KSC LC-39A achieved a 76.9% success rate while getting a 23.1% failure rate

Payload impact on Launch Outcomes





We can see the success rates for low weighted payloads is higher than the heavy weighted payloads

Section 5

Predictive Analysis (Classification)

Classification Accuracy

- The decision tree classifier is the model with the highest classification accuracy

In [34]:

```
# After comparing accuracy of above methods, they all preformed practically
# the same, except for tree which fit train data slightly better but test data worse.
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_)
```

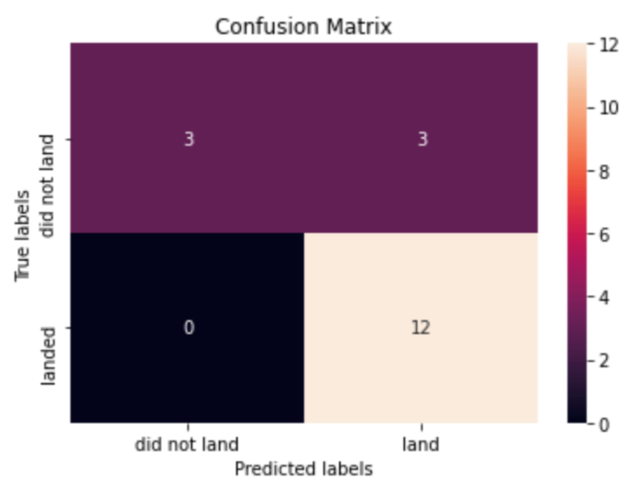
```
Best model is DecisionTree with a score of 0.8767857142857143
Best params is : {'criterion': 'entropy', 'max_depth': 12, 'max_features': 'sqrt', 'min_samples_leaf': 4, 'min_samples_split': 2, 'splitte
r': 'random'}
```

Confusion Matrix

- The confusion matrix for the decision tree classifier. Some unsuccessful landings are marked as false negatives.

In [32]:

```
yhat = knn_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
```



Conclusions

- The more launches at a launch site, the greater the success.
- 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 predicting success rate.


Thank you!