Winning Space Race with Data Science

OSEI RICHARD SAYI

23 – 04 – 2022

Outline

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

Executive Summary

Summary of methodologies

- Data collection
- Data wrangling
- EDA with data visualization
- EDA with SQL
- Building an interactive map with Folium
- Building a Dashboard with Plotly Dash
- Predictive analysis (Classification)

Summary of all results

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

Introduction

- **Project background and context**

We predicted if the Falcon 9 first stage will land successfully. SpaceX 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 SpaceX 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

- **Common problems that needed solving.**
  - What influences if the rocket will land successfully?
  - The effect each relationship with certain rocket variables will impact in determining the success rate of a successful landing.
  - What conditions does SpaceX have to achieve to get the best results and ensure the best rocket success landing rate

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology
- Performed data wrangling
- Performed exploratory data analysis (EDA) using visualization and SQL
- Performed interactive visual analytics using Folium and Plotly Dash
- Performed predictive analysis using classification models

Data Collection

- We worked with SpaceX launch data that is gathered from the SpaceX REST API.
- Another popular data source for obtaining Falcon 9 Launch data was web scraping Wikipedia using BeautifulSoup

Data Collection – SpaceX API

Use SpaceX Rest API

Return as JSON file

Filter & Sort

Export (.csv)

Data Collection - Scraping

Get Response (wikipedia)

Extract data with Beautiful Soup

Parse table into Dict

Export (.csv)

```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"
page = requests.get(static_url)
soup = BeautifulSoup(page.text, 'html.parser')
```

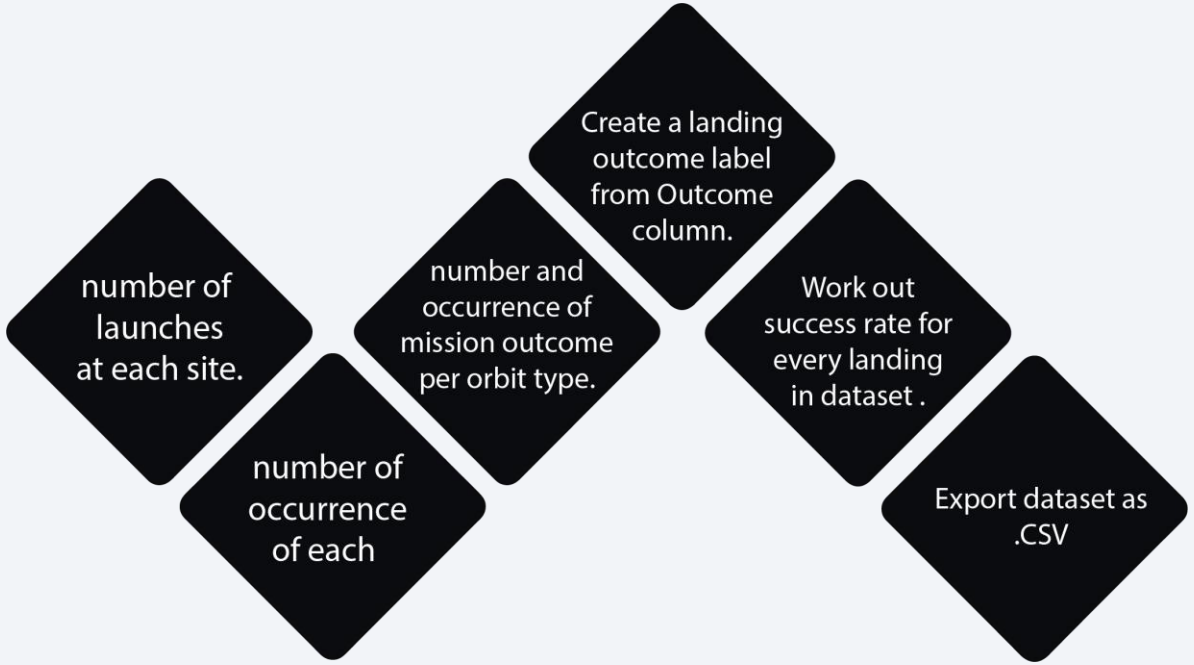
```
launch_dict= dict.fromkeys(column_names)
# Remove an irrelvant column
del launch_dict['Date and time ( )']
# Let's initial the launch_dict with each value to be an empty list
launch_dict['Flight No.'] = []
launch_dict['Launch site'] = []
launch_dict['Payload'] = []
launch_dict['Payload mass'] = []
launch_dict['Orbit'] = []
launch_dict['Customer'] = []
launch_dict['Launch outcome'] = []
# Added some new columns
launch_dict['Version Booster']=[]
launch_dict['Booster landing']=[]
launch_dict['Date']=[]
launch_dict['Time']=[]
```

```
headings = []
for key,values in dict(launch_dict).items():
    if key not in headings:
        headings.append(key)
    if values is None:
        del launch_dict[key]

def pad_dict_list(dict_list, padel):
    lmax = 0
    for lname in dict_list.keys():
        lmax = max(lmax, len(dict_list[lname]))
    for lname in dict_list.keys():
        ll = len(dict_list[lname])
        if  ll < lmax:
            dict_list[lname] += [padel] * (lmax - ll)
    return dict_list

pad_dict_list(launch_dict,0)

df = pd.DataFrame.from_dict(launch_dict)
df.to_csv('spacex_web_scraped.csv', index=False)
```

Data Wrangling

To find the successful and failed landing at each site, we performed the following exploratory data analysis on the dataset.

- Calculate the number of launches at each site.
- Calculate the number of occurrence of each orbit.
- Calculate the number and occurrence of mission outcome per orbit type.
- Create a landing outcome label from Outcome column
- Work out success rate for every landing in dataset .



EDA with Data Visualization

Scatter Graphs

- Flight Number VS. Payload Mass
- Flight Number VS. Launch Site
- Payload VS. Launch Site
- Orbit VS. Flight Number Payload
- Orbit Type Orbit VS. Payload

Scatter plots show how much one variable is affected by another. The relationship between two variables is called their correlation . Scatter plots usually consist of a large body of data.

Bar Chart

- Mean VS Orbit

A bar diagram makes it easy to compare sets of data between different groups at a glance. The graph represents categories on one axis and a discrete value in the other. The goal is to show the relationship between the two axes. Bar charts can also show big changes in data over time.

Line Graph

- Success Rate VS Year

Line graphs are useful in that they show data variables and trends very clearly and can help to make predictions about the results of data not yet recorded

EDA with SQL

For example of some questions we were asked about the data we needed information about. Which we are using SQL queries to get the answers in the dataset :

- Displaying the names of the unique launch sites in the space mission
- Displaying 5 records where launch sites begin with the string 'KSC'
- Displaying the total payload mass carried by boosters launched by NASA (CRS)
- Displaying average payload mass carried by booster version F9 v1.1
- Listing the date where the successful landing outcome in drone ship was achieved.
- Listing the names of the boosters which have success in ground pad and have payload mass greater than 4000 but less than 6000
- Listing the total number of successful and failure mission outcomes
- Listing the names of the booster versions which have carried the maximum payload mass.
- Listing the records which will display the month names, successful landing outcomes in ground pad ,booster versions, launch site for the months in year 2017
- Ranking the count of successful landing outcomes between the date 2010-06-04 and 2017-03-20 in descending order

Build an Interactive Map with Folium

To visualize the Launch Data into an interactive map.

- We took the Latitude and Longitude Coordinates at each launch site and added a Circle Marker around each launch site with a label of the name of the launch site.
- We assigned the dataframe launch outcomes(failures, successes) to classes 0 and 1 with Green and Red markers on the map in a Marker Cluster()

Using Haversine's formula we calculated the distance from the Launch Site to various landmarks to find various trends about what is around the Launch Site to measure patterns. Lines are drawn on the map to measure distance to landmarks

Example of some trends in which the Launch Site is situated in.

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

Build a Dashboard with Plotly Dash

The dashboard is built with Plotly and Dash web framework.

- Graph

1. **Pie Chart showing the total launches by a certain site/all sites**
   - *display relative proportions of multiple classes of data.*
   - size of the circle can be made proportional to the total quantity it represents.

2. **Scatter Graph showing the relationship with Outcome and Payload Mass (Kg) for the different Booster Versions**
   - It shows the relationship between two variables.
   - It is the best method to show you a non-linear pattern.
   - The range of data flow, i.e. maximum and minimum value, can be determined.
   - Observation and reading are straightforward.

Predictive Analysis (Classification)

BUILDING MODEL

- Load our dataset into NumPy and Pandas
- Transform Data
- Split our data into training and test data sets
- Check how many test samples we have
- Decide which type of machine learning algorithms we want to use
- Set our parameters and algorithms to GridSearchCV
- Fit our datasets into the GridSearchCV objects and train our dataset.

EVALUATING MODEL

- Check accuracy for each model
- Get tuned hyperparameters for each type of algorithms
- Plot Confusion Matrix

IMPROVING MODEL

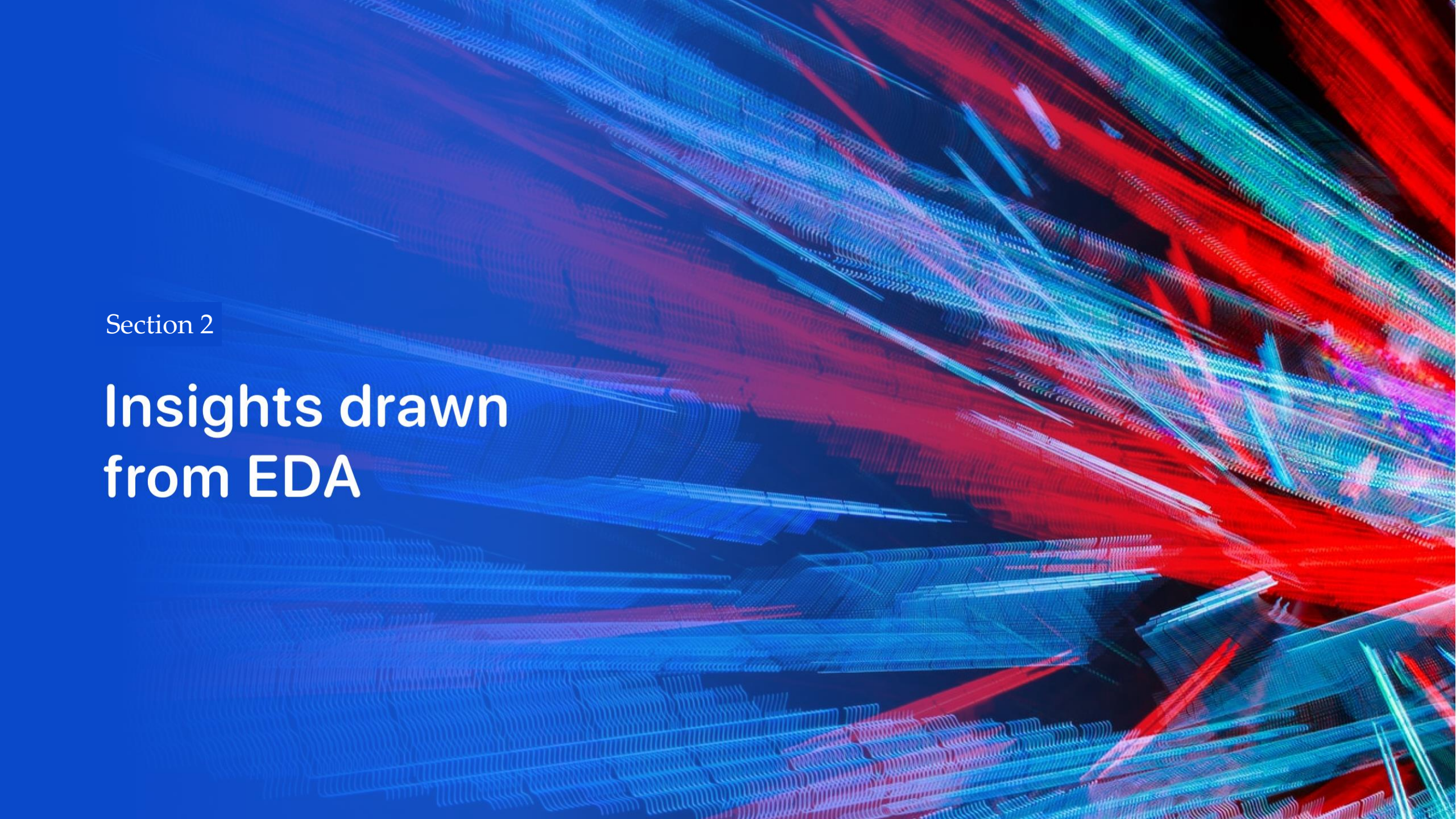
- Feature Engineering
- Algorithm Tuning

FINDING THE BEST PERFORMING CLASSIFICATION MODEL

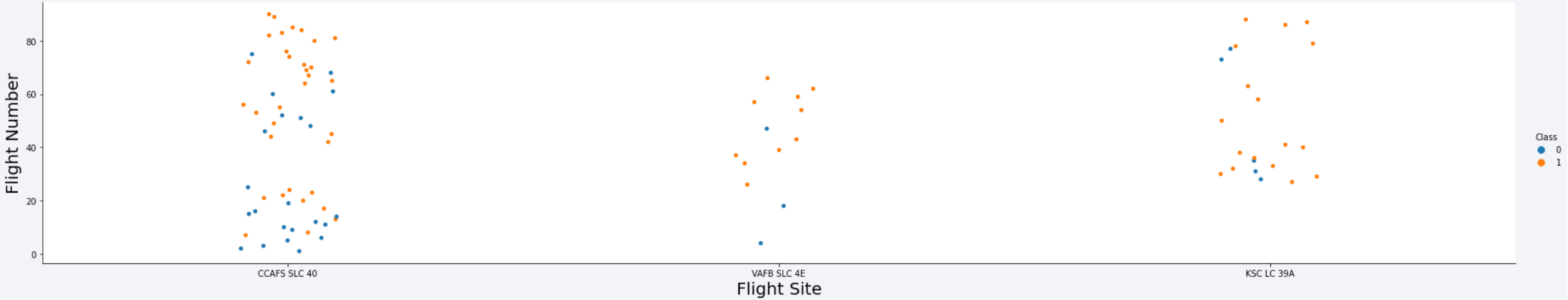
- The model with the best accuracy score wins the best performing model
- In the notebook there is a dictionary of algorithms with scores at the bottom of the notebook.

Results

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

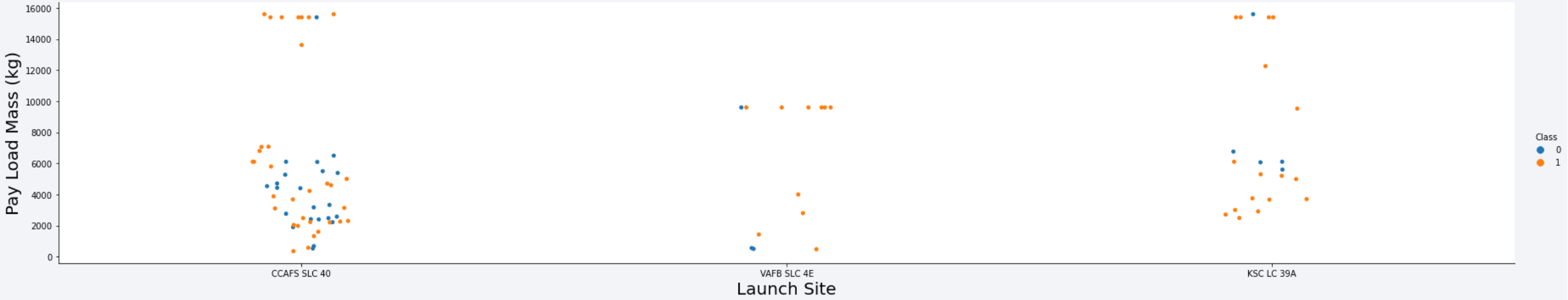

Section 2
Insights drawn from EDA

Flight Number vs. Launch Site



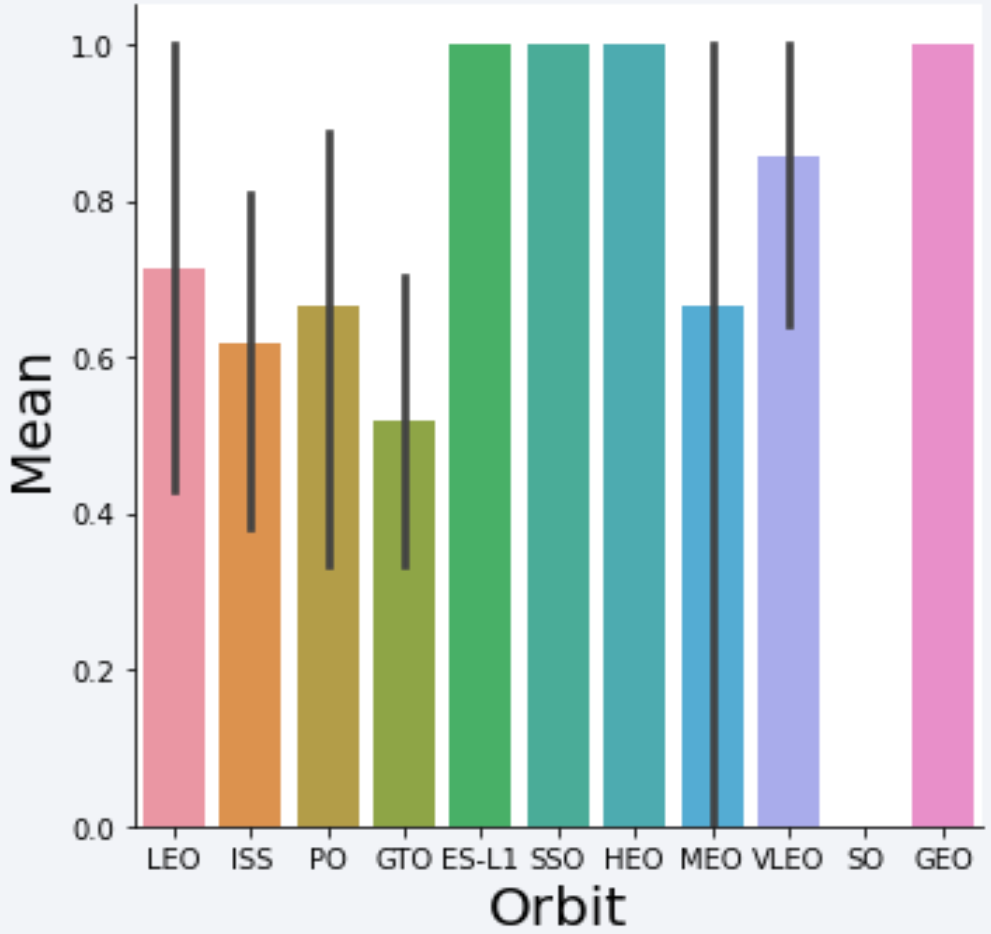
The more amount of flights at a launch site the greater the success rate at a launch site.

Payload vs. Launch Site



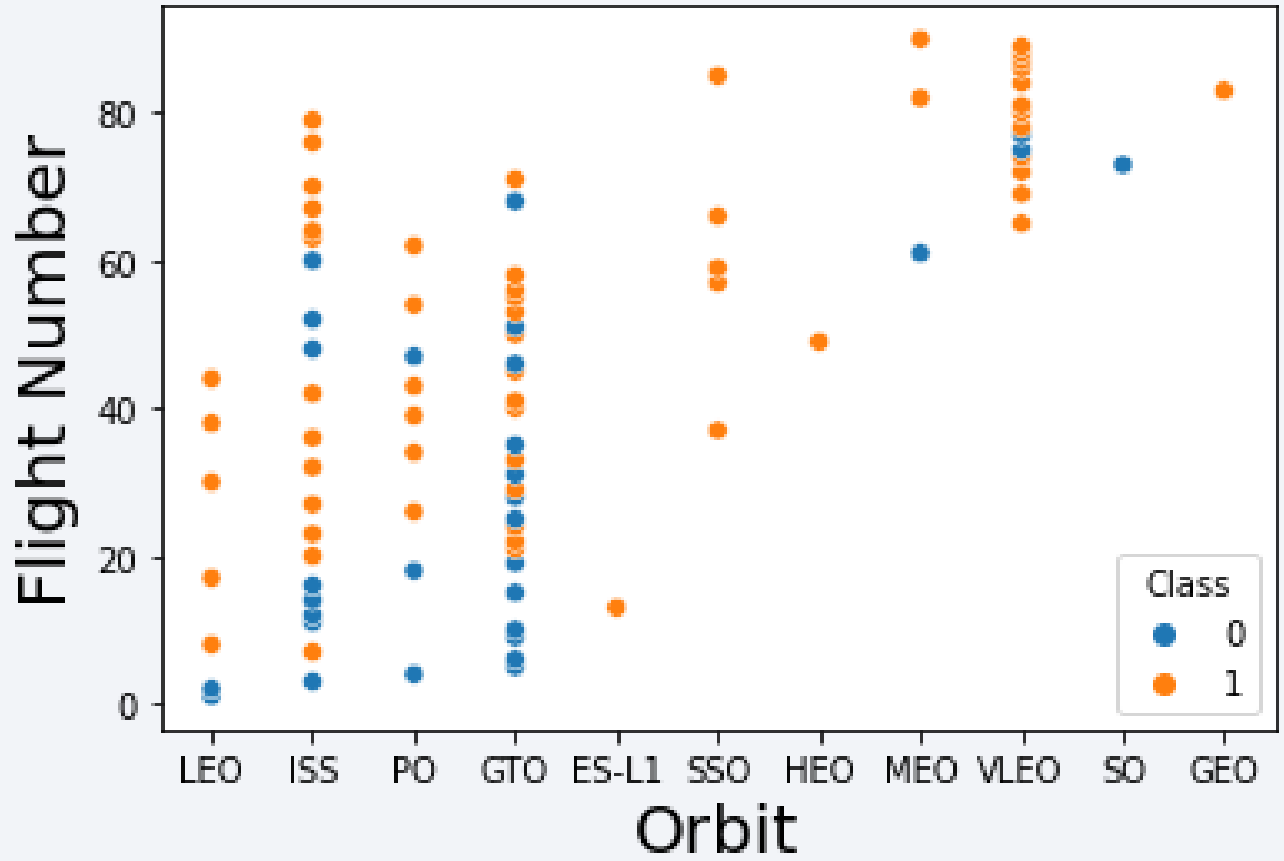
The greater the payload mass for Launch Site CCAFSSLC40 the higher the success rate for the Rocket. There is not quite a clear pattern to be found using this visualization to make decision if the Launch Site is dependant on PayLoad Mass for a success launch.

Success Rate vs. Orbit Type



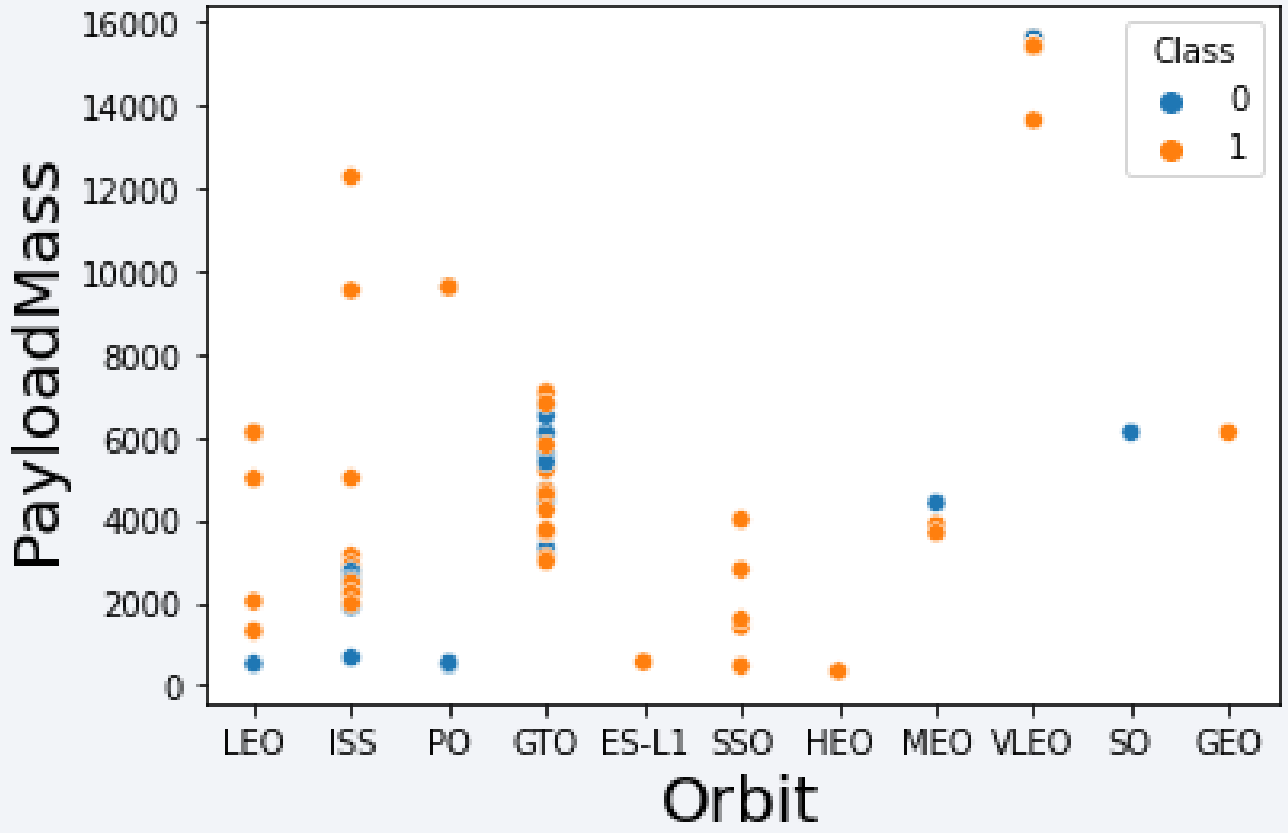
Orbit GEO,HEO,SSO,ES L1 has the best Success Rate

Flight Number vs. Orbit Type



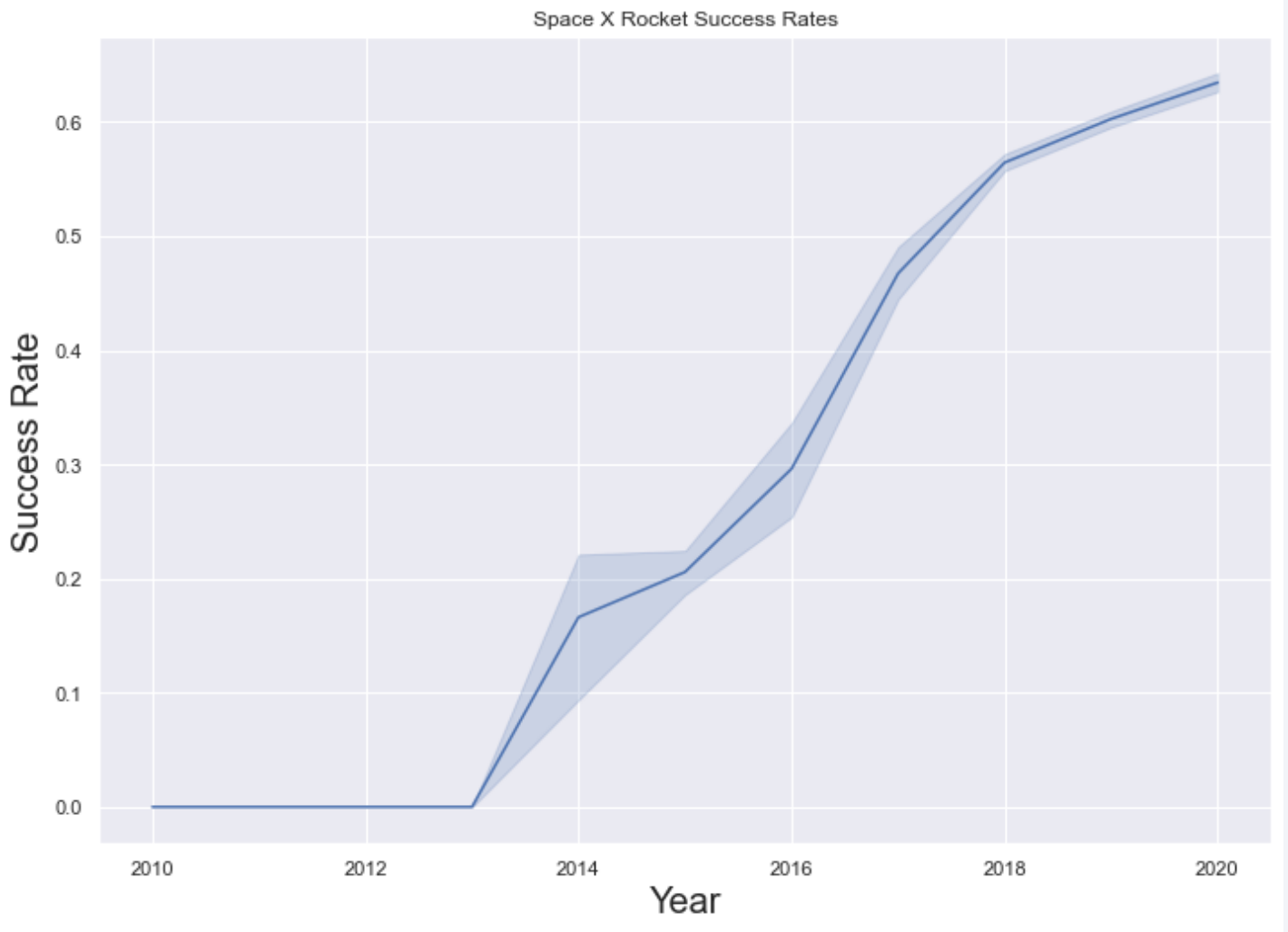
The LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.

Payload vs. Orbit Type



You should observe that Heavy payloads have a negative influence on GTO orbits and positive on GTO and Polar LEO (ISS) orbits.

Launch Success Yearly Trend



you can observe that the success rate since 2013kept increasing till 2020

All Launch Site Names

```
%%sql
SELECT DISTINCT Launch_Site from SPACEXTBL
```

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

Using the word ***DISTINCT*** in the query means that it will only show Unique values in the ***Launch_Site*** column from ***SPACEXTBL***

Launch Site Names Begin with 'CCA'

```
%%sql
SELECT * FROM SPACEXTBL
WHERE launch_site LIKE '%CCA%' LIMIT 5
```

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

QUERY EXPLAINATION

Using the word ***TOP 5*** in the query means that it will only show 5 records from ***SPACEXTBL*** and ***LIKE*** keyword has a wild card with the words ***'KSC%'*** the percentage in the end suggests that the Launch_Site name must start with KSC

Total Payload Mass

```
%%sql
SELECT SUM(payload_mass__kg_) AS TotalPayloadMass
FROM SPACEXTBL
```

totalpayloadmass
619967

QUERY EXPLAINATION

Using the function ***SUM*** summates the total in the column ***PAYLOAD_MASS_KG_***

Average Payload Mass by F9 v1.1

```
%%sql
SELECT AVG(payload_mass__kg_) AS F9V1
FROM SPACEXTBL
WHERE booster_version LIKE '%F9 v1.1%'
```

f9v1
2534

QUERY EXPLAINATION

Using the function ***AVG*** works out the average in the column ***PAYLOAD_MASS_KG_***

The ***WHERE*** clause filters the dataset to only perform calculations on ***Booster_version F9 v1.1***

First Successful Ground Landing Date

```
%%sql
SELECT min(DATE) AS FirstSuccessLanding
FROM SPACEXTBL
WHERE landing__outcome = 'Success (ground pad)'
```

firstsuccesslanding
2015-12-22

QUERY EXPLAINATION

Using the function ***MIN*** works out the minimum date in the column ***Date***

The ***WHERE*** clause filters the dataset to only perform calculations on ***Landing_Outcome Success (drone ship)***

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%%sql
SELECT payload
FROM SPACEXTBL
WHERE landing__outcome = 'Success (drone ship)' AND
payload_mass__kg_ BETWEEN 4000 AND 6000
```

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

QUERY EXPLAINATION

Selecting only ***Booster_Version***

The ***WHERE*** clause filters the dataset to ***Landing_Outcome = Success (drone ship)***

The ***AND*** clause specifies additional filter conditions ***Payload_MASS_KG_*** > 4000 AND ***Payload_MASS_KG_ < 6000***

Total Number of Successful and Failure Mission Outcomes

```
%%sql
SELECT mission_outcome,count(*)
FROM SPACEXTBL
GROUP by mission_outcome
```

mission_outcome	2
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

The use of count(*) sums the unique outcomes and GROUP is used to group the outcomes based on their differences

Boosters Carried Maximum Payload

Used a nexted querry, %%sql

```
select BOOSTER_VERSION as boosterversion,
PAYLOAD_MASS__KG_
from SPACEXTBL
where PAYLOAD_MASS__KG_=(select
max(PAYLOAD_MASS__KG_) from SPACEXTBL);
```

Used a nexted querry to retrieve the bosterversion that carried the maximum weight. Used the max function to find the maximum payload Which was 15600kg.

boosterversion	payload_mass__kg_
F9 B5 B1048.4	15600
F9 B5 B1049.4	15600
F9 B5 B1051.3	15600
F9 B5 B1056.4	15600
F9 B5 B1048.5	15600
F9 B5 B1051.4	15600
F9 B5 B1049.5	15600
F9 B5 B1060.2	15600
F9 B5 B1058.3	15600
F9 B5 B1051.6	15600
F9 B5 B1060.3	15600
F9 B5 B1049.7	15600

2015 Launch Records

```
%%sql SELECT MONTH(DATE),
MISSION_OUTCOME,
BOOSTER_VERSION,
LAUNCH_SITE
FROM SPACEXTBL
where EXTRACT(YEAR FROM DATE)='2015';
```

1	mission_outcome	booster_version	launch_site
1	Success	F9 v1.1 B1012	CCAFS LC-40
2	Success	F9 v1.1 B1013	CCAFS LC-40
3	Success	F9 v1.1 B1014	CCAFS LC-40
4	Success	F9 v1.1 B1015	CCAFS LC-40
4	Success	F9 v1.1 B1016	CCAFS LC-40
6	Failure (in flight)	F9 v1.1 B1018	CCAFS LC-40
12	Success	F9 FT B1019	CCAFS LC-40

WHERE clause filters ***Year*** to be 2015 from the extracted date data.

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

```
%%sql
SELECT LANDING__OUTCOME,count(*) as
TotalNumber
FROM SPACEXTBL
WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20'
GROUP by LANDING__OUTCOME
ORDER BY TotalNumber DESC
```

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

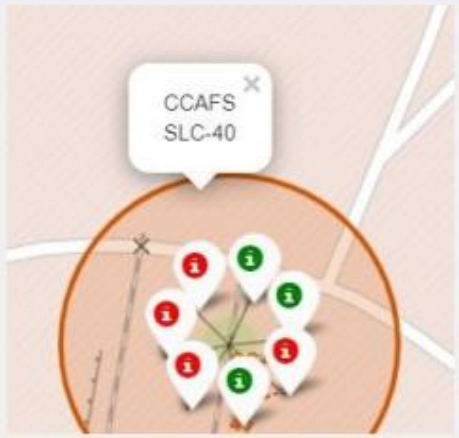
Section 3

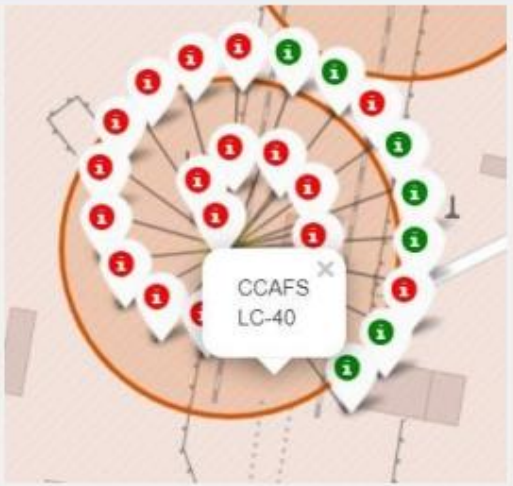
Launch Sites Proximities Analysis

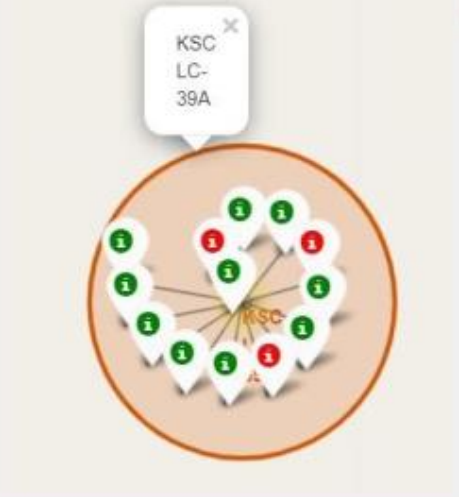
All launch sites global map markers

All launch sites global map markers

Color label Markers





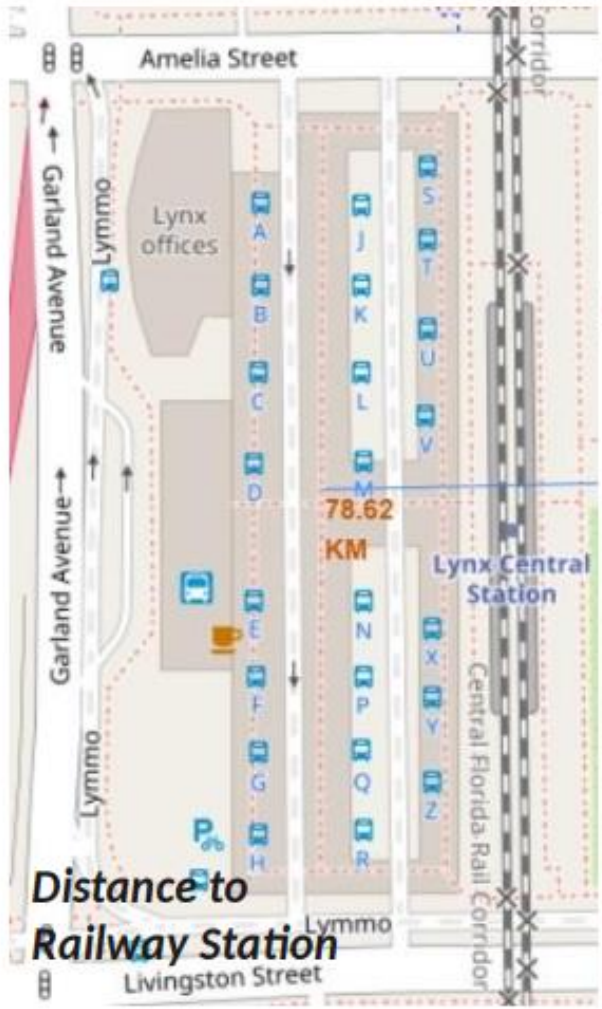


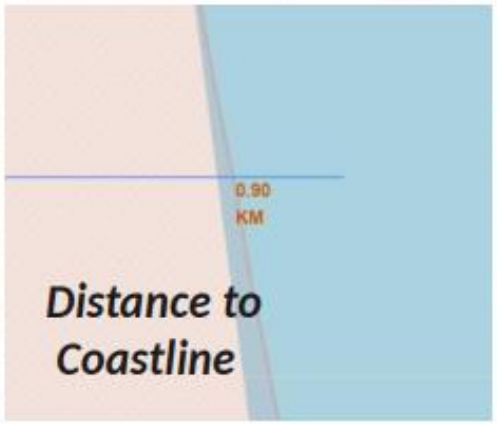
Florida Launch Sites

Green Marker shows successful Launches and Red Marker shows Failures

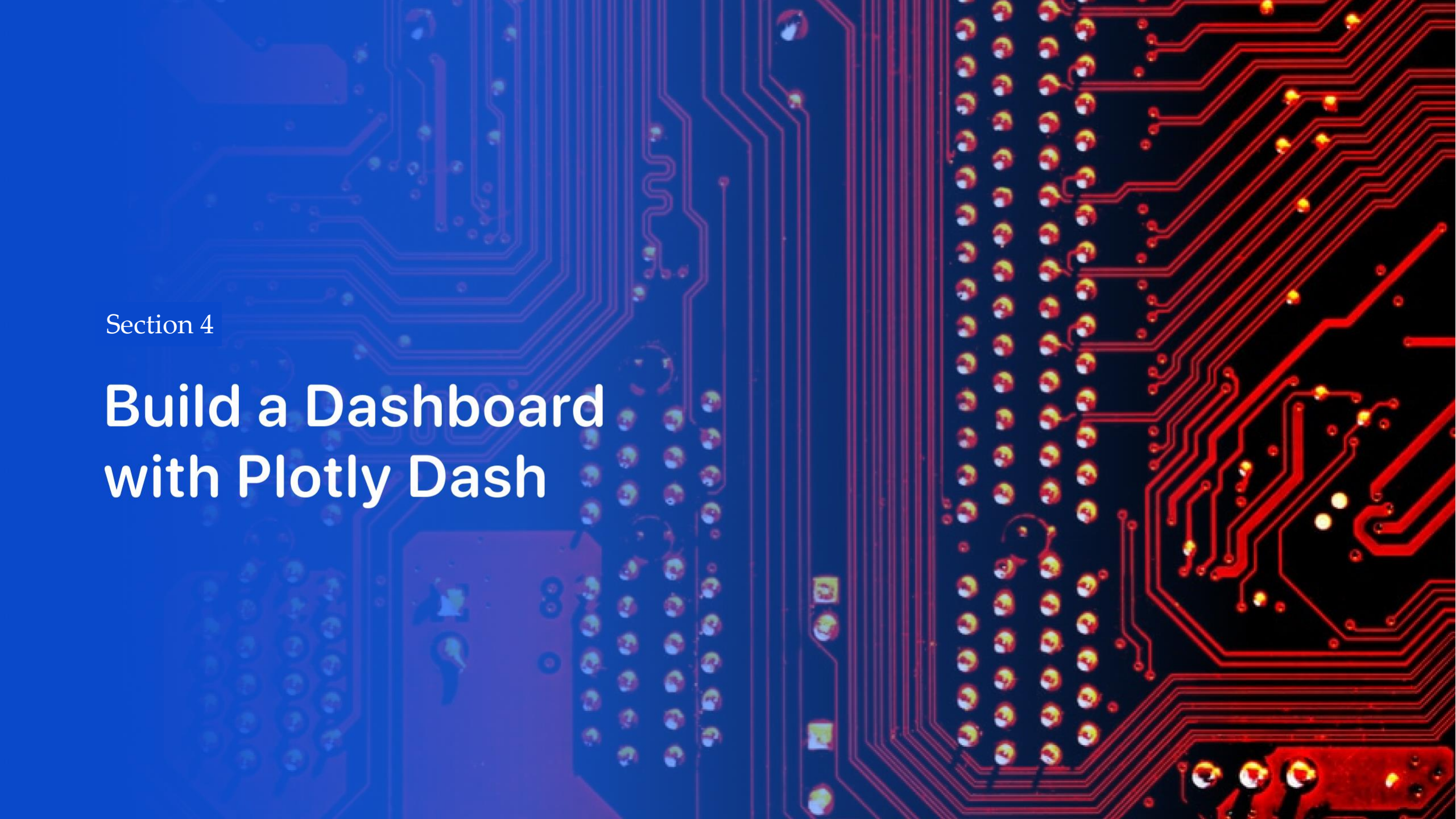
California Launch Site

Working out Launch Sites distance to landmarks to find trends with Haversine formula using CCAFS-SLC-40 as a reference



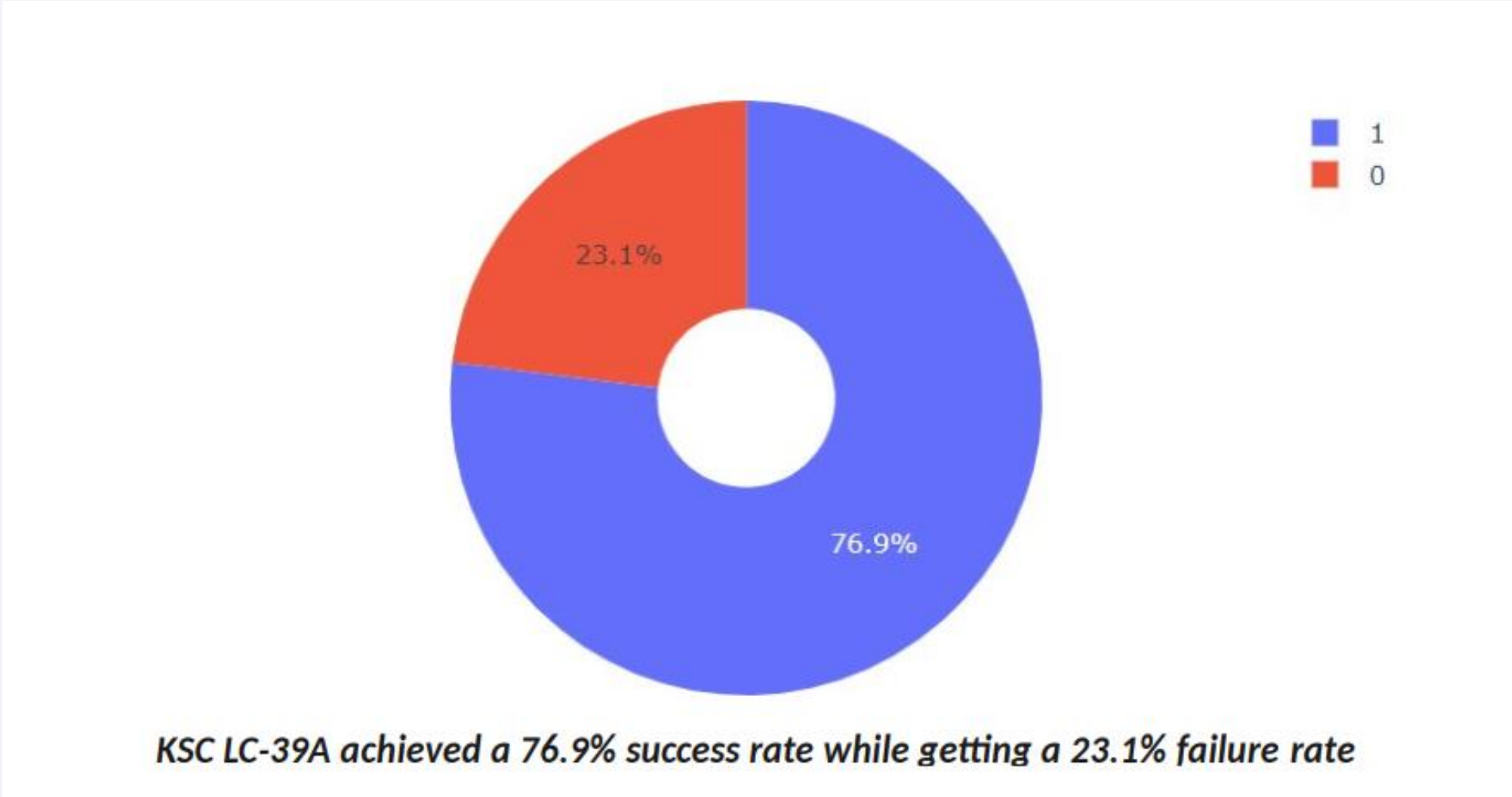


- Are launch sites in close proximity to railways? No
- Are launch sites in close proximity to highways? No
- Are launch sites in close proximity to coastline? Yes
- Do launch sites keep certain distance away from cities? Yes


Section 4
Build a Dashboard with Plotly Dash

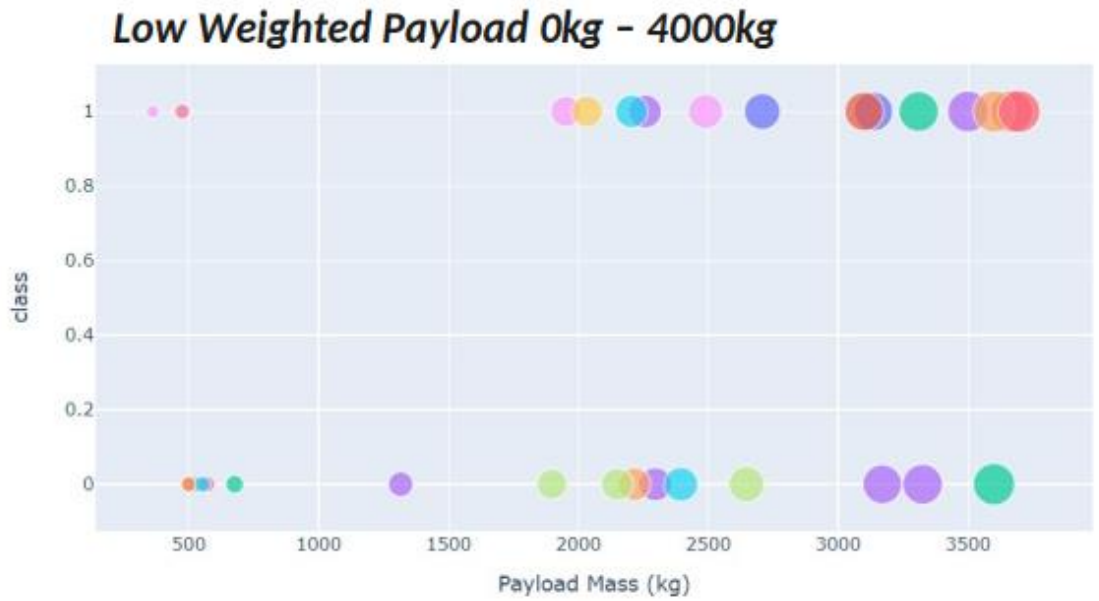
DASHBOARD – Pie chart showing the success percentage achieved by each launch site

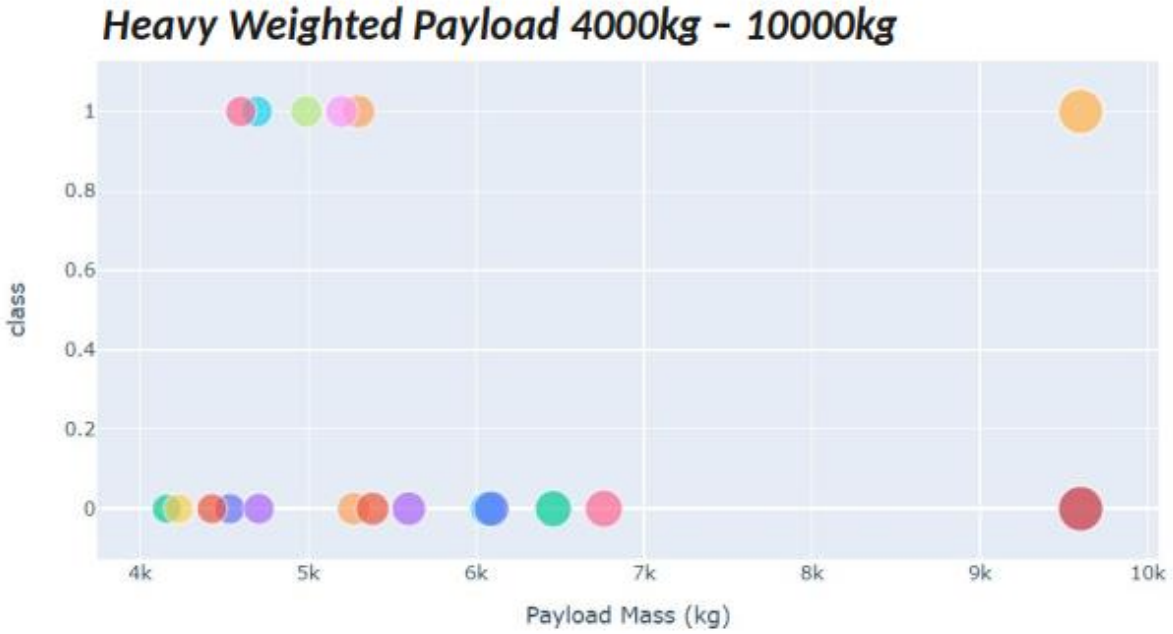
DASHBOARD – Pie chart for the launch site with highest launch success ratio



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

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



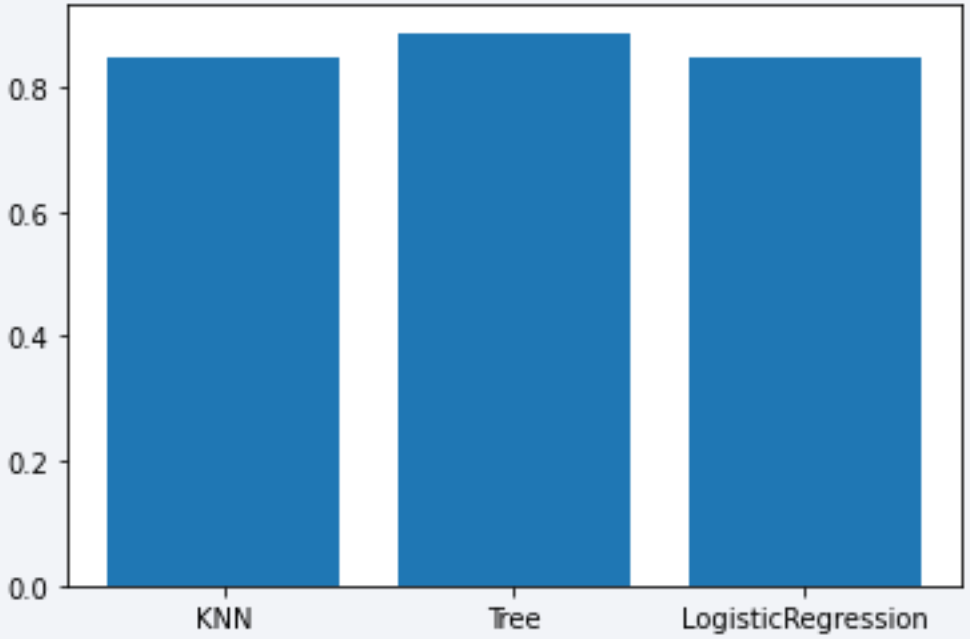


We can see the success rates for low weighted payloads is higher than the heavy weighted payloads

Section 5

Predictive Analysis (Classification)

Classification Accuracy

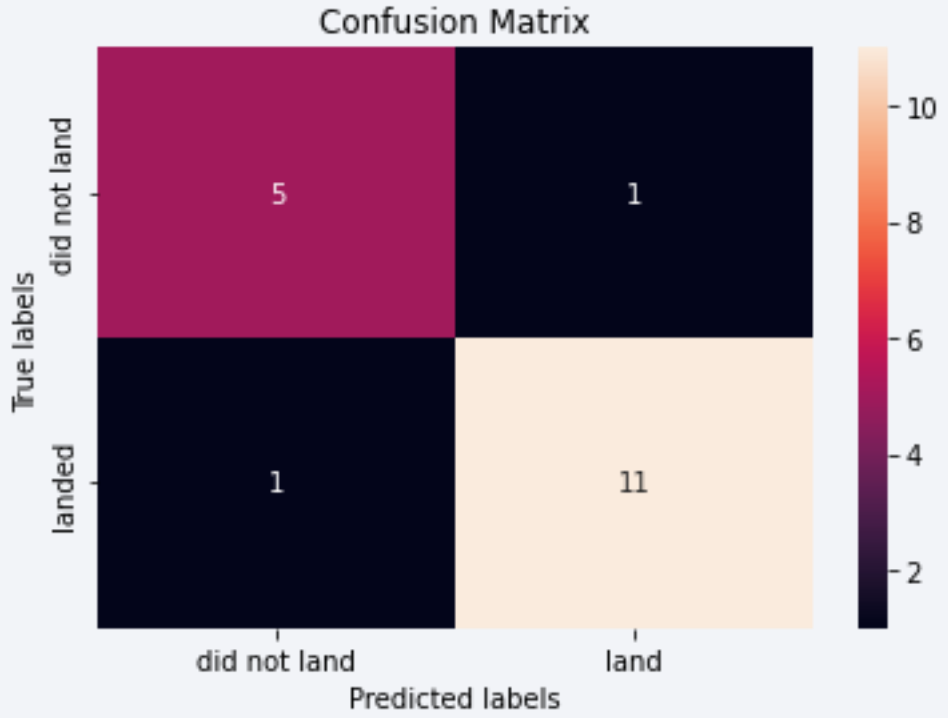


Knn	Tree	LogisticRegression
0.848214	0.875	0.846429

After selecting the best hyperparameters for the decision tree classifier using the validation data, we achieved 88.8% accuracy on the test data

Confusion Matrix

Examining the confusion matrix, we see that Tree can distinguish between the different classes. We see the model was able to predict a significant number of outcomes.



Conclusions

- The Tree Classifier Algorithm is the best for Machine Learning for this dataset
- Low weighted payloads perform better than the heavier payloads
- The success rates for SpaceX launches is directly proportional time in years they will eventually perfect the launches
- We can see that KSC LC-39A had the most successful launches from all the sites
- Orbit GEO,HEO,SSO,ES-L1 has the best Success Rate

Appendix

- Include any relevant assets like Python code snippets, SQL queries, charts, Notebook outputs, or data sets that you may have created during this project

Thank you!