Winning Space Race with Data Science

Joshua Hill PhD.

14 September 2022

Outline

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

Executive Summary

- Summary of methodologies
  - Data collection using SpaceX API and Wikipedia
  - Data wrangling using python
  - Exploratory data analysis using python and SQL
  - Interactive maps using Folium
  - Interactive dashboards using Plotly
  - Machine learning to predict launch outcomes
- Summary of results
  - Exploratory data analysis
  - Screenshots if interactive maps and dashboards
  - Machine learning models

Introduction

- SpaceX launches reusable falcon9 rockets thereby bringing down the cost of each launch. Using machine learning we predict which launch variables are useful in predicting successful rocket launches and subsequent booster revoceries
- Problems you want to find answers
  - What influences if the booster will land successfully
  - Is one launch site better than another
  - Does orbit insertion matter

Section 1

Methodology

Methodology

Executive Summary

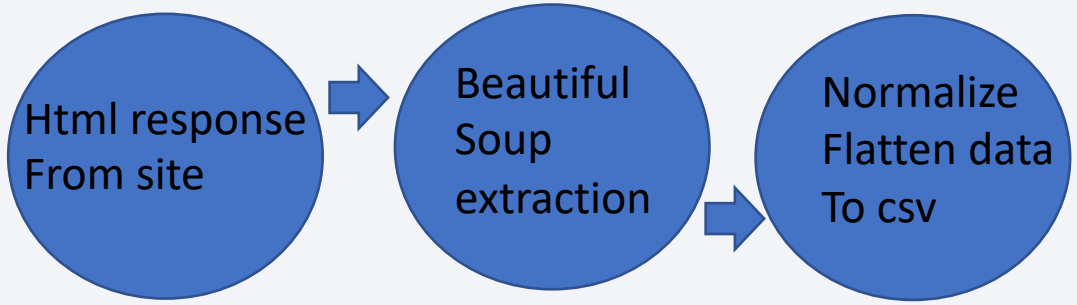
- Data collection methodology:
  - Data collected from the SpaceX website using their API and from the SpaceX Wikipedia page
- Perform data wrangling
  - Data was transformed using one-hot encoding for machine learning
  - Dates were made uniform
  - Nan data was made uniform
- Perform exploratory data analysis (EDA) using visualization and SQL
  - Used scatter plots and bar graphs to look at the data visually to find outliers and trrends
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Logistcal regression, SVM, Decision Tree and KNN machine learning models were used to, evaluate classification models

Data Collection

- Data sets were collected from the following sources.
  - SpaceX data was collected using the public SpaceX REST API
  - This data gives us dates, launches, payload mass, orbits and booster info
  - Web-scrapping was used to get data from the SpaceX Wikipedia page

SpaceX REST API

Web-scrapping



Data Collection – SpaceX API

- Complete code can be found on github at the address below

- https://github.com/jhill1440/IBM-applied-data-science-capstone/blob/8652b8be62f1617739e713d4e58277638039da4f/jupyter-labs-spacex-data-collection-api-complete.ipynb

Submit GET request and make a variable for response

```
spacex_url="https://api.spacexdata.com/v4/launches/past"
```

```
response = requests.get(spacex_url)
```

Make a dataframe from the json

```
# Use json_normalize meethod to convert the json result into a dataframe
response.json()
data = pd.json_normalize(response.json())
```

Apply custom functions and make a dictionary from the json

```
# Create a data from launch_dict
df = pd.DataFrame.from_dict(launch_dict)
```

Filter data based on wanted output

```
# Hint data['BoosterVersion']!='Falcon 1'
data_falcon9 = df[df['BoosterVersion'] != 'Falcon 1']
data_falcon9.head(5)
```

Data Collection - Scraping

- Complete code can be found on github at the address below
- https://github.com/jhill1440/IBM-applied-data-science-capstone/blob/8652b8be62f1617739e713d4e58277638039da4f/jupyter-labs-webscraping-complete.ipynb

Assign the target URL to a variable and use a GET response to Retrieve the data

Use Beautiful soup to parse the data

```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(page.text, 'html.parser')
```

Find all the tables

```
# Use the find_all function in the BeautifulSoup object, with element type `table`
# Assign the result to a list called `html_tables`
html_tables=soup.find_all('table')
```

Get the column names

```
column_names = []

# Apply find_all() function with `th` element on first_launch_table
# Iterate each th element and apply the provided extract_column_from_header() to get a column name
# Append the Non-empty column name (`if name is not None and len(name) > 0`) into a list called column_names
for i in first_launch_table.find_all('th'):
    if extract_column_from_header(i)!=None:
        if len(extract_column_from_header(i))>0:
            column_names.append(extract_column_from_header(i))
```

Create a dictionary, make it a dataframe and convert to csv

```
df=pd.DataFrame.from_dict(launch_dict, orient='index')
```

```
df.to_csv('spacex_web_scraped.csv', index=False)
```

Data Wrangling

- Complete code can be found on github at the address below
- https://github.com/jhill1440/IBM-applied-data-science-capstone/blob/8652b8be62f1617739e713d4e58277638039da4f/labs-jupyter-spacex-Data%20wrangling-complete.ipynb

The data was processed using a one-hot transformation on the launch outcome and was added as another column called Class. This was used to look at the different orbits and launch sites vs successful launch and booster recovery.

Preformed Exploratory Data Analysis

Calculated launch outcomes at different sites

Calculated orbit vs launch outcome

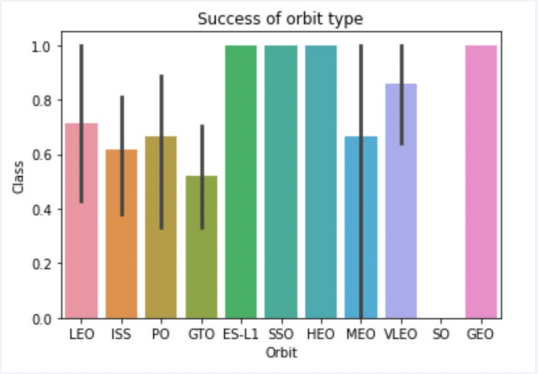
EDA with Data Visualization

- Complete code can be found on github at the address below
- https://github.com/jhill1440/IBM-applied-data-science-capstone/blob/8652b8be62f1617739e713d4e58277638039da4f/jupyter-labs-eda-dataviz-complete.ipynb

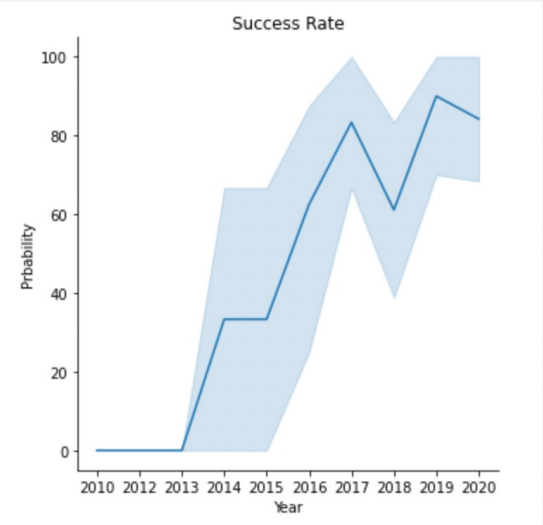
Scatter plots were made of:

Flight number vs Payload Mass
Flight Number vs Launch Site
Launch Site vs Payload Mass
Launch Site vs Orbit
Payload Mass vs Orbit

A bar graph was made of:
Orbit vs Successful booster recovery



A line graph was made of:
Successful booster recovery by year



EDA with SQL

- Complete code can be found on github at the address below
- https://github.com/jhill1440/IBM-applied-data-science-capstone/blob/8652b8be62f1617739e713d4e58277638039da4f/jupyter-labs-eda-sql-coursera_sqllite-complete.ipynb

SQL Queries were made to view the following:

- Unique launch sites
- Filtered to show only “CCA” launch sites
- Total payload mass launched
- Average payload mass launched
- Payloads between 4K kg and 9K kg that landed on droneship
- Number of missions that were successful
- Booster numbers that launched the most mass
- Failed outcomes between to dates that tried to land on a droneship
- Landing outcomes between two dates

Build an Interactive Map with Folium

- Complete code can be found on github at the address below
- https://github.com/jhill1440/IBM-applied-data-science-capstone/blob/8652b8be62f1617739e713d4e58277638039da4f/lab_jupyter_launch_site_location-complete.ipynb

Data was plotted on a folium map to visualize launch outcome on a launch site basis.
Data was visualized by adding red (failure) or green (success) circles to the map
We plotted the distance to the nearest city.
We plotted the distance to the nearest highway
We plotted the distance to the nearest rail tracks
We plotted the distance to the nearest coast

Build a Dashboard with Plotly Dash

- Complete code can be found on github at the address below
- https://github.com/jhill1440/IBM-applied-data-science-capstone/blob/8652b8be62f1617739e713d4e58277638039da4f/spacex-dashboard.py

A plotly dashboard was made so data could be visualized as the user wants to see it.
It has a drop down menu to select a launch site and this output is shown on a pie chart for launch success from the Selected launch site
A scatter plot can be made with a selector payload mass range and mission outcome

Predictive Analysis (Classification)

- Complete code can be found on github at the address below
- https://github.com/jhill1440/IBM-applied-data-science-capstone/blob/8652b8be62f1617739e713d4e58277638039da4f/SpaceX_Machine%20Learning%20Prediction_Part_5-complete.ipynb

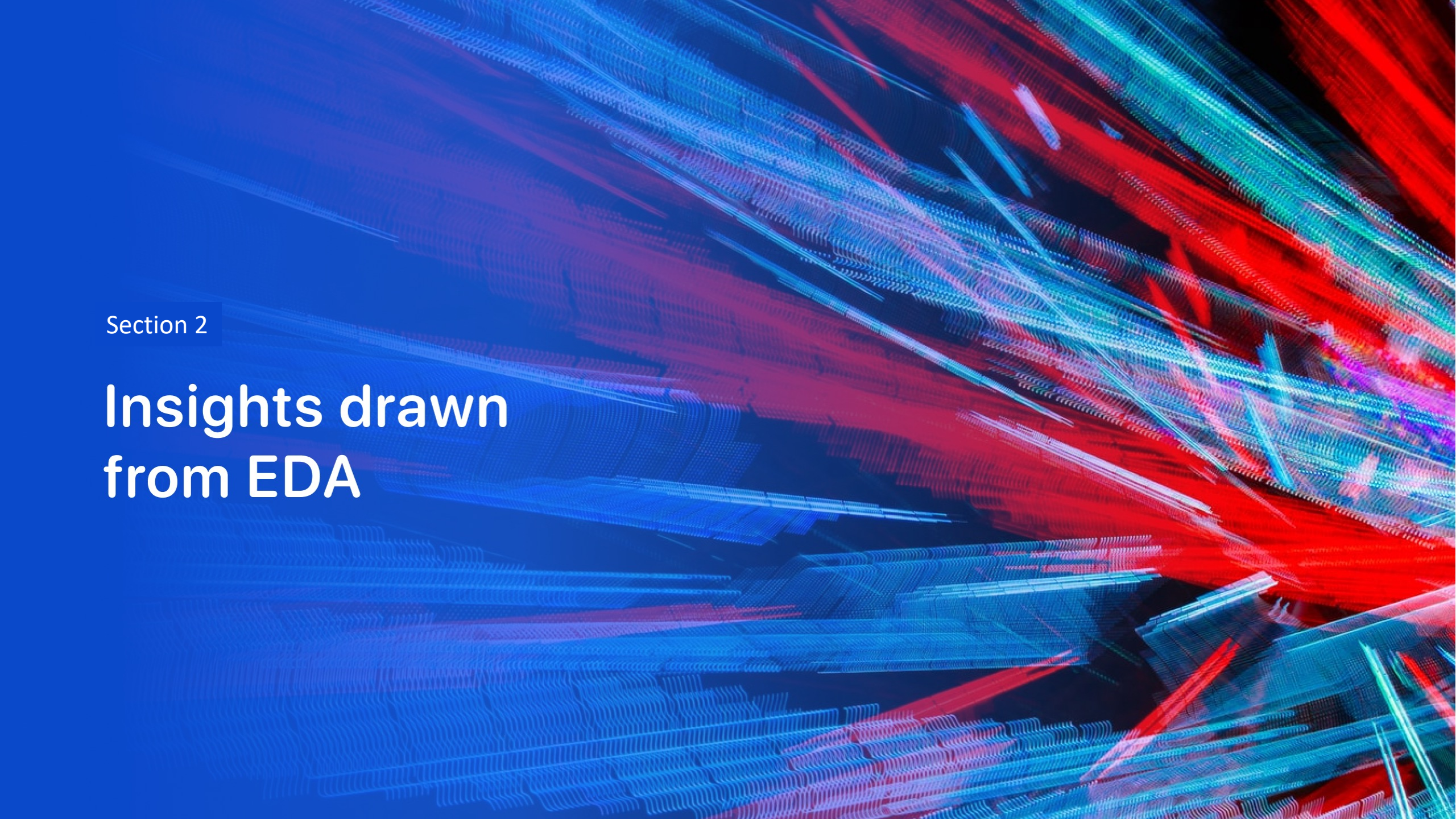
A machine learning model was made by loading data into a dataframe using pandas and numpy.
This was transformed to standard scaler and then split into a train and test set.

Four different models were created to test which was the best:

- Logitical regression
- SVM
- Decision Tree
- KNN

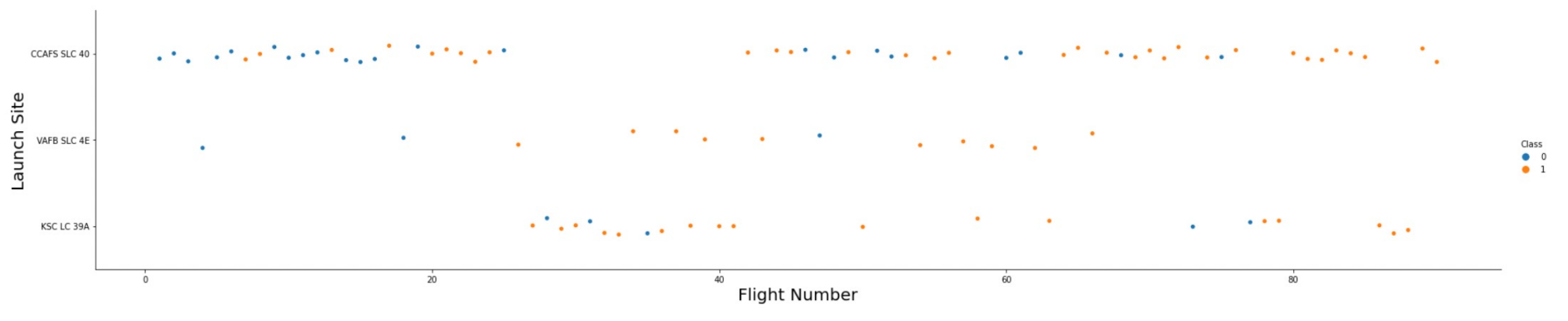
Results

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


Section 2
Insights drawn from EDA

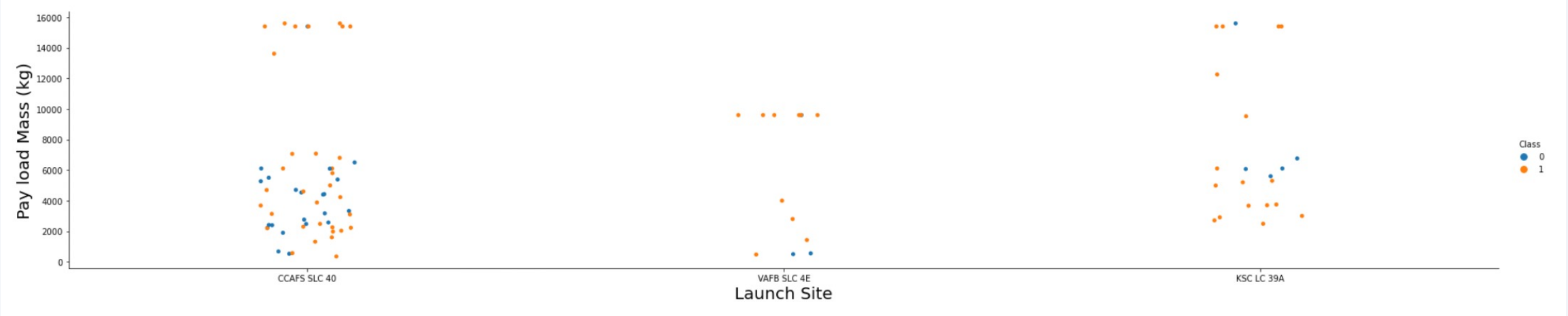
Flight Number vs. Launch Site

This shows where each flight number was launched from and if it was:
successful (class1, orange)
or
failure (class 2, blue)



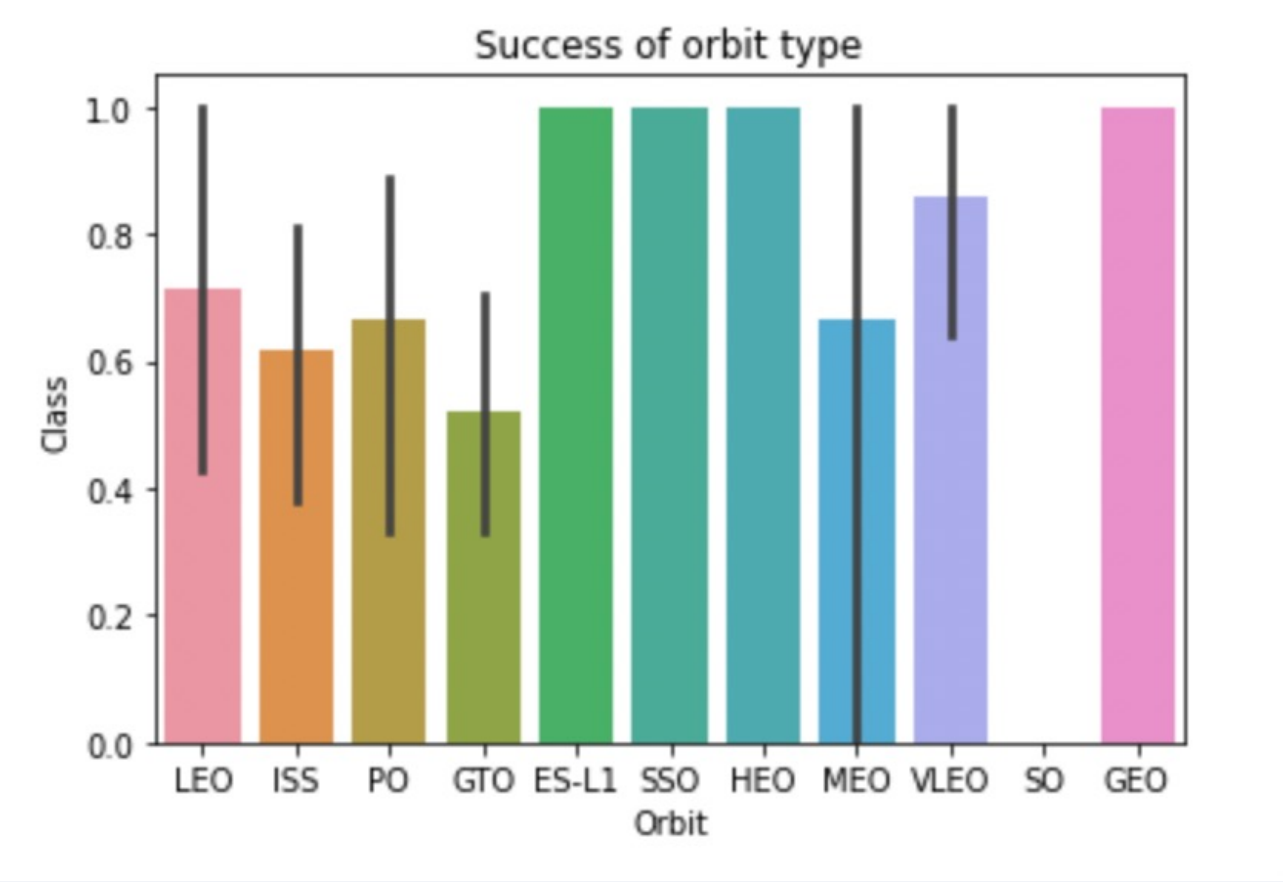
Payload vs. Launch Site

This shows the relationship between launch site and payload mass:
success (class 1, orange)
or
failure (class 0, blue)



Success Rate vs. Orbit Type

This shows the probability of success vs orbital destination



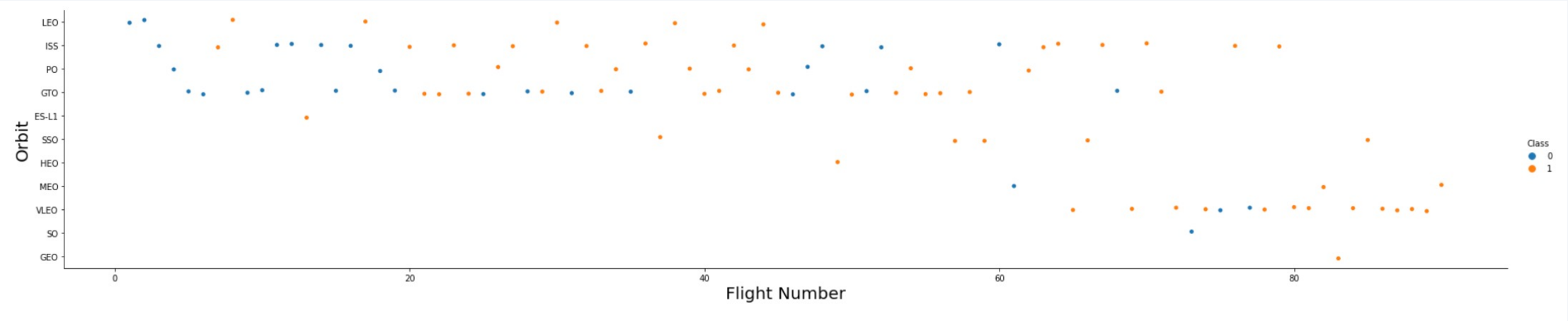
Flight Number vs. Orbit Type

This shows the relationship between the flight number and orbital destination:

success (class 1, orange)

or

failure (class 0, blue)



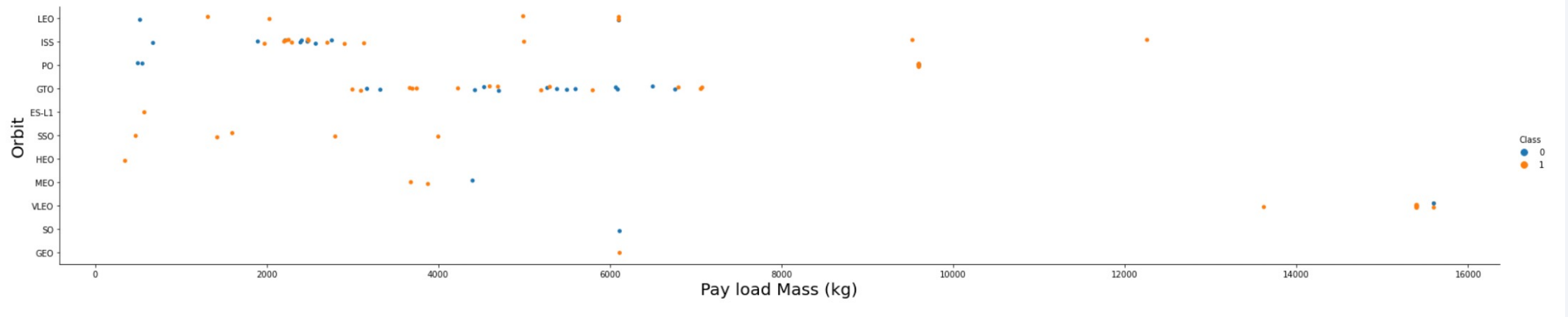
Payload vs. Orbit Type

This shows the relationship between the payload mass and orbital destination:

success (class 1, orange)

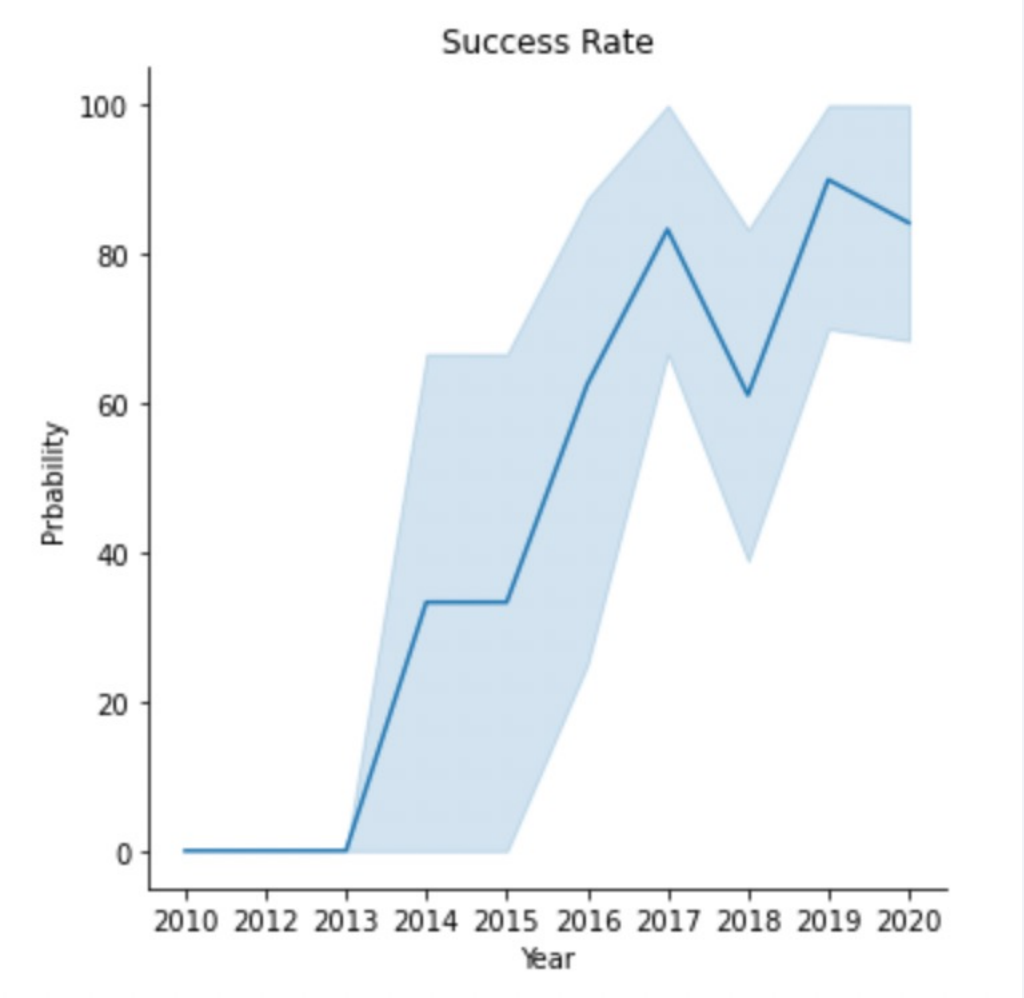
or

failure (class 0, blue)



Launch Success Yearly Trend

Probability of success on a year by year basis



All Launch Site Names

- Find the names of the unique launch sites
- sql select DISTINCT LAUNCH_SITE from SPACEXTBL

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

https://github.com/jhill1440/IBM-applied-data-science-capstone/blob/3d1ae8a97e33aaabd9ad5e3b8a65367d54df8db7/jupyter-labs-eda-sql-coursera_sqllite-complete.ipynb

Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites 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
04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
08-12-2010	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)
22-05-2012	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
01-03-2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

https://github.com/jhill1440/IBM-applied-data-science-capstone/blob/3d1ae8a97e33aaabd9ad5e3b8a65367d54df8db7/jupyter-labs-eda-sql-coursera_sqllite-complete.ipynb

Total Payload Mass

- Calculate the total payload carried by boosters from NASA
- sql select sum(payload_mass__kg_) as sum from SPACEXTBL where customer like 'NASA (CRS)'

sum
45596

https://github.com/jhill1440/IBM-applied-data-science-capstone/blob/3d1ae8a97e33aaabd9ad5e3b8a65367d54df8db7/jupyter-labs-eda-sql-coursera_sqllite-complete.ipynb

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1
- %sql select avg(payload_mass__kg_) as Average from SPACEXTBL where booster_version like 'F9 v1.1%'

Average
2534.6666666666665

https://github.com/jhill1440/IBM-applied-data-science-capstone/blob/3d1ae8a97e33aaabd9ad5e3b8a65367d54df8db7/jupyter-labs-eda-sql-coursera_sqllite-complete.ipynb

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad
- %sql select min(date) as Date from SPACEXTBL where mission_outcome like 'Success'

Date
01-03-2013

https://github.com/jhill1440/IBM-applied-data-science-capstone/blob/3d1ae8a97e33aaabd9ad5e3b8a65367d54df8db7/jupyter-labs-eda-sql-coursera_sqllite-complete.ipynb

Successful Drone Ship Landing with Payload between 4000 and 6000

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

- %sql select booster_version from SPACEXTBL where ("Landing _Outcome" like 'Success (drone ship)') AND (payload_mass__kg_ BETWEEN 4000 AND 6000) AND (mission_outcome like 'Success')

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

https://github.com/jhill1440/IBM-applied-data-science-capstone/blob/3d1ae8a97e33aaabd9ad5e3b8a65367d54df8db7/jupyter-labs-eda-sql-coursera_sqllite-complete.ipynb

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes
- %sql SELECT mission_outcome, count(*) as Count FROM SPACEXTBL GROUP by mission_outcome ORDER BY mission_outcome

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

https://github.com/jhill1440/IBM-applied-data-science-capstone/blob/3d1ae8a97e33aaabd9ad5e3b8a65367d54df8db7/jupyter-labs-eda-sql-coursera_sqllite-complete.ipynb

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass
- %sql select booster_version from SPACEXTBL where payload_mass__kg_=(select max(payload_mass__kg_) from SPACEXTBL)

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

https://github.com/jhill1440/IBM-applied-data-science-capstone/blob/3d1ae8a97e33aaabd9ad5e3b8a65367d54df8db7/jupyter-labs-eda-sql-coursera_sqllite-complete.ipynb

2015 Launch Records

- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
- %sql select substr (DATE, 4, 2) as Month, "Landing _Outcome", booster_version, launch_site from SPACEXTBL where substr(DATE,7,4) like '2015%' AND "Landing _Outcome" like 'Failure (drone ship)'

Month	Landing _Outcome	Booster_Version	Launch_Site
01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

https://github.com/jhill1440/IBM-applied-data-science-capstone/blob/3d1ae8a97e33aaabd9ad5e3b8a65367d54df8db7/jupyter-labs-eda-sql-coursera_sqllite-complete.ipynb

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

- Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order
- %sql select "Landing _Outcome", count (*) as count from SPACEXTBL where Date >= '04-06-2010' AND Date <= '20-03-2017' GROUP by "Landing _Outcome" ORDER BY count Desc

Landing _Outcome	count
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

https://github.com/jhill1440/IBM-applied-data-science-capstone/blob/3d1ae8a97e33aaabd9ad5e3b8a65367d54df8db7/jupyter-labs-eda-sql-coursera_sqllite-complete.ipynb


Section 3
Launch Sites
Proximities Analysis

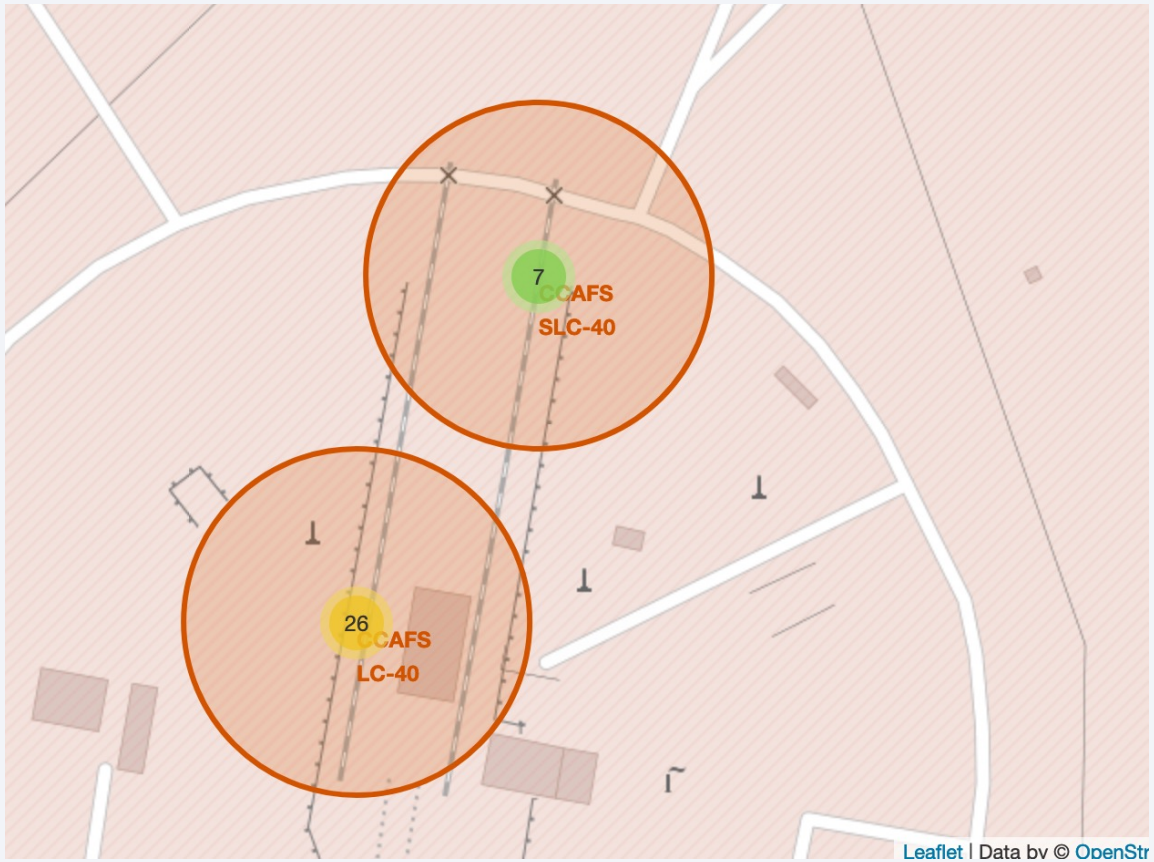
Folium Map of Launch Sites

- There are 3 different launch sites on two coasts

https://github.com/jhill1440/IBM-applied-data-science-capstone/blob/3d1ae8a97e33aaabd9ad5e3b8a65367d54df8db7/lab_jupyter_launch_site_location-complete.ipynb

Folium Map Launch site numbers

- Number of launches are each site

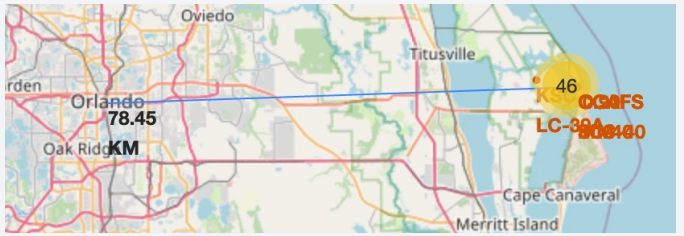


https://github.com/jhill1440/IBM-applied-data-science-capstone/blob/3d1ae8a97e33aaabd9ad5e3b8a65367d54df8db7/lab_jupyter_launch_site_location-complete.ipynb

Folium Map Distances to Major Sites

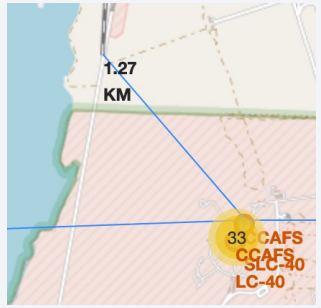
Distance to Coast

Distance to major city

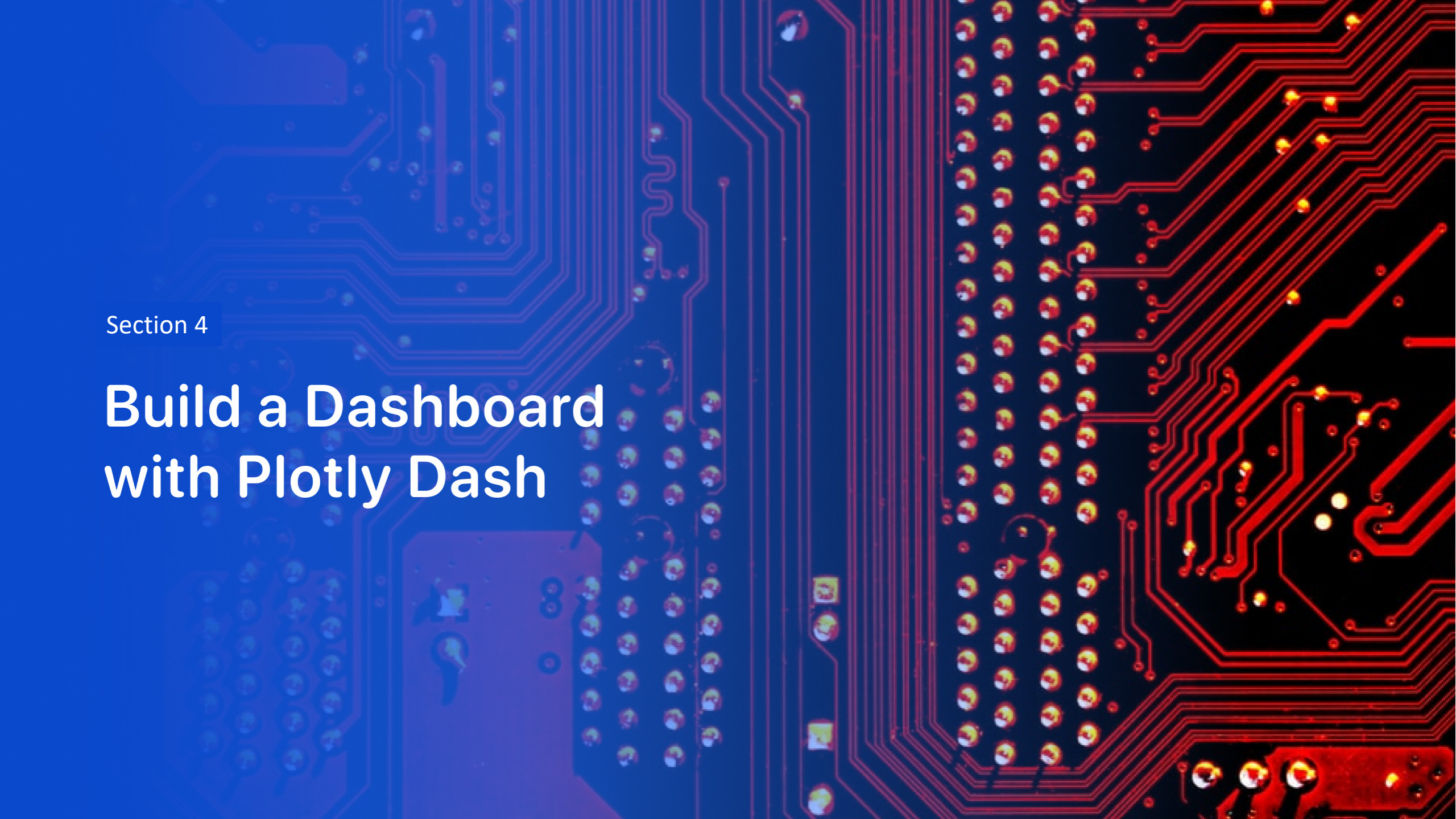


Distance to Highway

Distance to Railroad



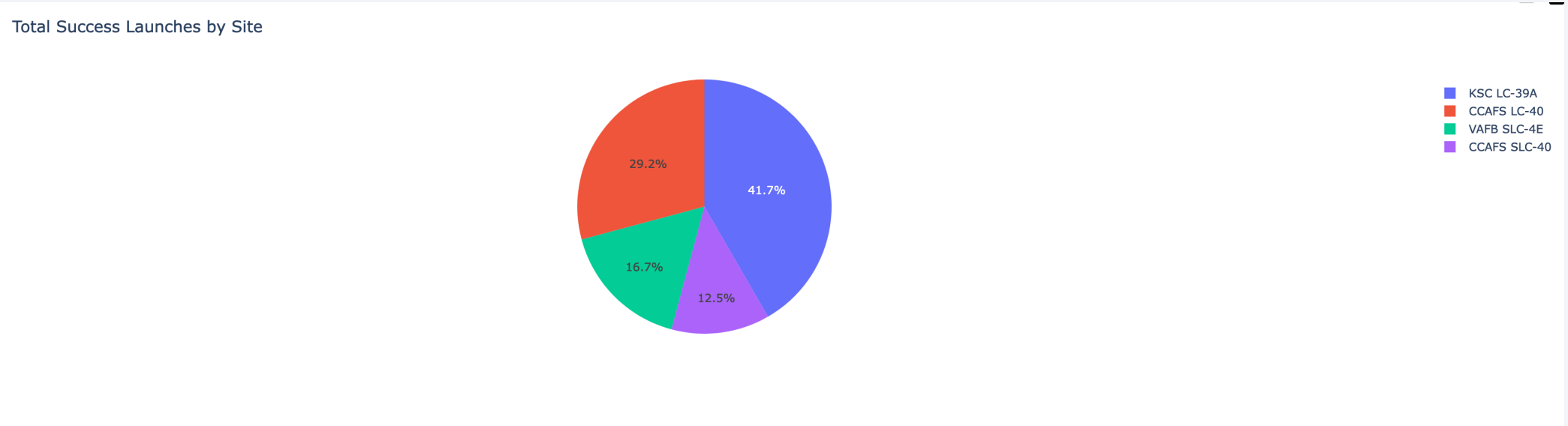
https://github.com/jhill1440/IBM-applied-data-science-capstone/blob/3d1ae8a97e33aaabd9ad5e3b8a65367d54df8db7/lab_jupyter_launch_site_location-complete.ipynb


Section 4
Build a Dashboard with Plotly Dash

Plotly Dashboard

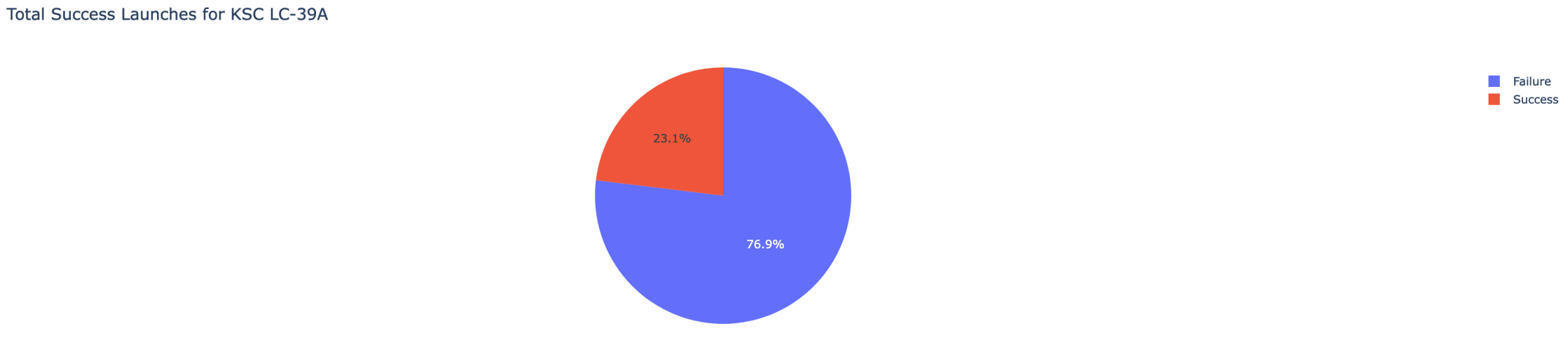
Screenshot of launch success count for all sites, in a piechart

- The best chance of launch success is from KSC LC-39A



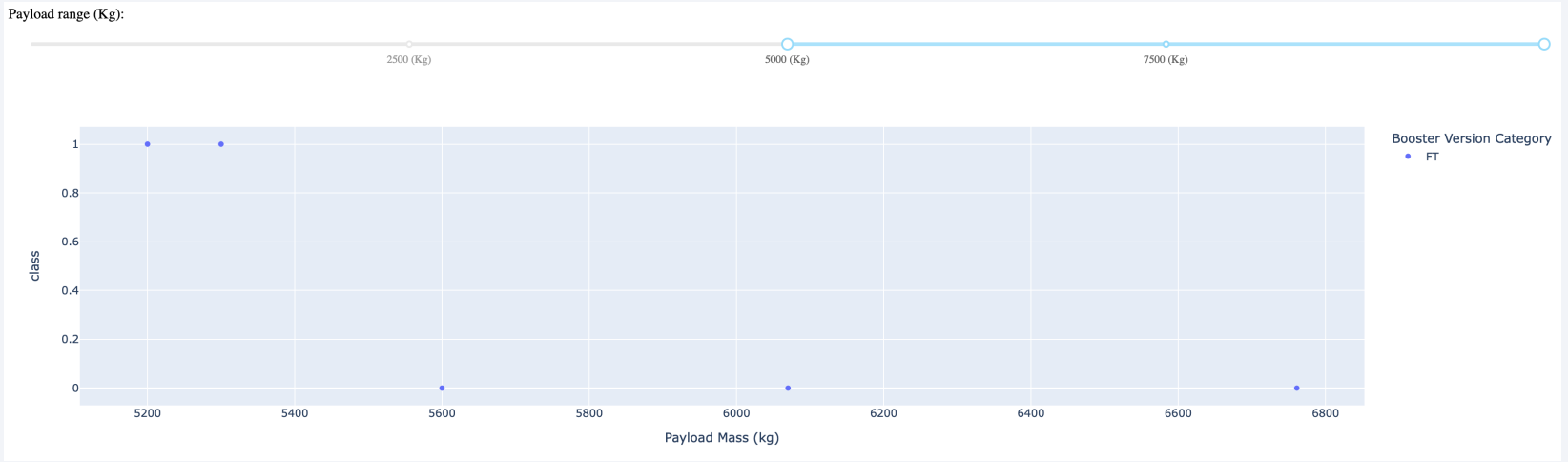
Plotly Dashboard

Screenshot of the piechart for the launch site with highest launch success ratio

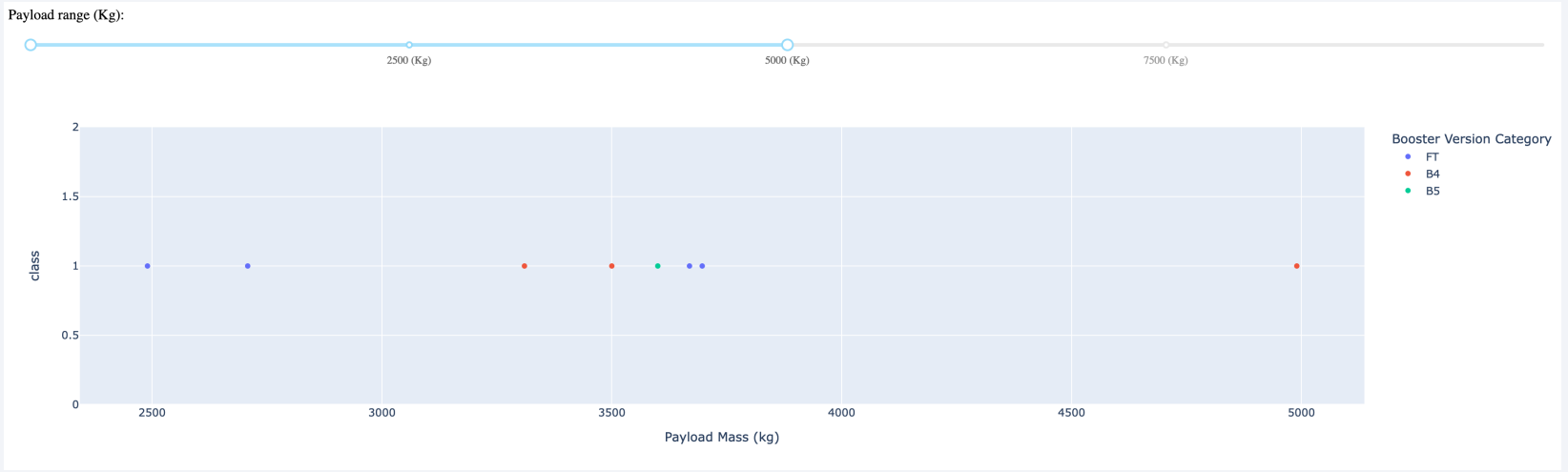


Plotly Dashboard

Success of launches with payloads above 5000kg



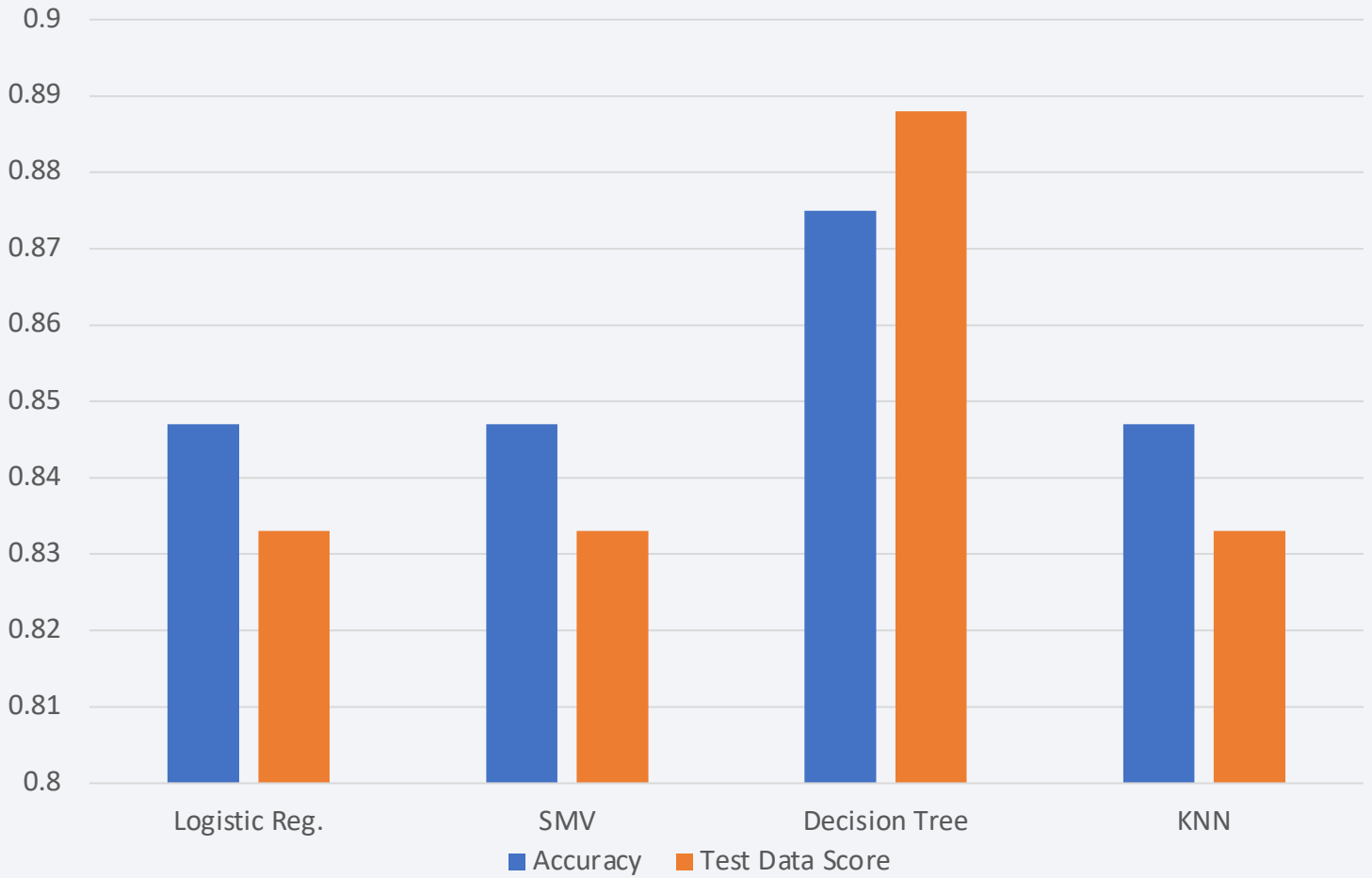
Success of launches with payloads below 5000kg



Section 5

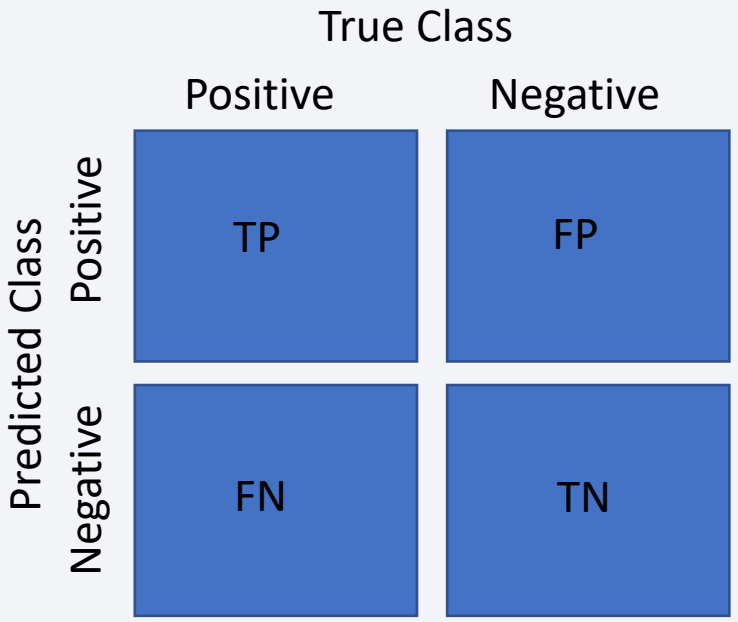
Predictive Analysis (Classification)

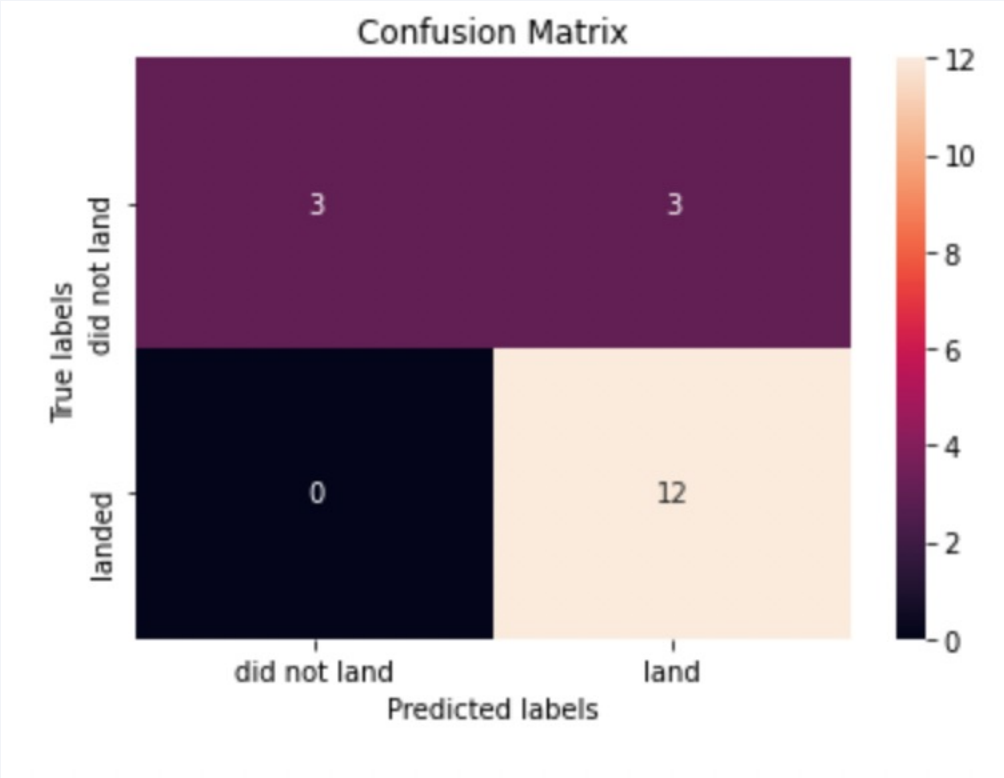
Classification Accuracy



Confusion Matrix for Decision Tree Analysis

Confusion matrix Key:





Conclusions

- The Decision tree classifier is the best with the data available. With more data and more variable a different model could be better
- Launches that are light (<5000kg) and go from KSC have the best chance of success
- SpaceX has consistantly improved their launch success rate over time

Thank you!