IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Aaron Tan
1 June 2022

Outline

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

https://github.com/ahron111/Coursera-IBM-Applied-Data-Science-Capstone

Executive Summary

Summary of methodologies

- Data Collection via API, Web Scraping
- Exploratory Data Analysis (EDA) with Data Visualization
- EDA with SQL
- Interactive Map with Folium
- Dashboards with Plotly Dash
- Predictive Analysis

Summary of all results

- Exploratory Data Analysis results
- Interactive maps and dashboard
- Predictive results

https://github.com/ahron111/Coursera-IBM-Applied-Data-Science-Capstone

Introduction

IBM Developer SKILLS NETWORK

Project Background and Context

This project is to predict if the Falcon 9 first stage will successfully land. From its website, the Falcon 9 rocket launch cost 62 million dollars. Other providers cost upward of 165 million dollars. The price difference is explained by SpaceX being able to reuse the first stage. By determining if the stage will land, we can determine the cost of a launch. This information is relevant for other companies if it wants to compete with SpaceX for a rocket launch.

Key Problems to Answer

- Main characteristics of a successful or failed landing
- Effect of the relationships of the variables to the success or failure of the landing
- Conditions that allow SpaceX to achieve the best landing success rate

https://github.com/ahron111/Coursera-IBM-Applied-Data-Science-Capstone

Section 1
Methodology

Methodology

Executive Summary

- Data collection methodology:
  - SpaceX REST API
  - Web scraping on Wikipedia
- Perform data wrangling
  - Removing redundant columns
  - One Hot Encoding for Classification models
- 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

https://github.com/ahron111/Coursera-IBM-Applied-Data-Science-Capstone

Data Collection

IBM Developer SKILLS NETWORK

Datasets collected from REST SpaceX API and web scrapping Wikipedia

Information obtained: Rocket, Launches, Payload

REST SpaceX API

SpaceX REST API call → JSON file returns → Dataframe from JSON → Clean data for export

Web Scrapping Wikipedia

Get HTML response (Wikipedia) → Extract data With BeautifulSoup → Create Dataframe → Export

https://github.com/ahron111/Coursera-IBM-Applied-Data-Science-Capstone

Data Collection – SpaceX API

1. API Response

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

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

2. JSON file Conversion

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

3. Transform Data

```
# Lets take a subset of our dataframe keeping only the features we want and the flight number, and date_utc.
data = data[['rocket', 'payloads', 'launchpad', 'cores', 'flight_number', 'date_utc']]

# We will remove rows with multiple cores because those are falcon rockets with 2 extra rocket boosters and rows that have multiple payloads in a single rocket.
data = data[data['cores'].map(len)==1]
data = data[data['payloads'].map(len)==1]

# Since payloads and cores are lists of size 1 we will also extract the single value in the list and replace the feature.
data['cores'] = data['cores'].map(lambda x : x[0])
data['payloads'] = data['payloads'].map(lambda x : x[0])

# We also want to convert the date_utc to a datetime datatype and then extracting the date leaving the time
data['date'] = pd.to_datetime(data['date_utc']).dt.date

# Using the date we will restrict the dates of the launches
data = data[data['date'] <= datetime.date(2020, 11, 13)]
```


https://github.com/ahron111/Coursera-IBM-Applied-Data-Science-Capstone

Data Collection – SpaceX API

IBM Developer SKILLS NETWORK

4. Create dictionary

```
launch_dict = {'FlightNumber': list(data['flight_number']),
'Date': list(data['date']),
'BoosterVersion':BoosterVersion,
'PayloadMass':PayloadMass,
'Orbit':Orbit,
'LaunchSite':LaunchSite,
'Outcome':Outcome,
'Flights':Flights,
'GridFins':GridFins,
'Reused':Reused,
'Legs':Legs,
'LandingPad':LandingPad,
'Block':Block,
'ReusedCount':ReusedCount,
'Serial':Serial,
'Longitude': Longitude,
'Latitude': Latitude}
```

5. Create dataframe

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

6. Filter dataframe

```
# Hint data['BoosterVersion']!='Falcon 1'
data_falcon9 = df[df['BoosterVersion']!='Falcon 1']
```

```
data_falcon9.loc[:,'FlightNumber'] = list(range(1, data_falcon9.shape[0]+1))
data_falcon9
```

```
# Calculate the mean value of PayloadMass column
payloadmassavg = data_falcon9['PayloadMass'].mean()
# Replace the np.nan values with its mean value
data_falcon9['PayloadMass'].replace(np.nan, payloadmassavg, inplace=True)
```

7. Export

```
data_falcon9.to_csv('dataset_part_1.csv', index=False)
```


https://github.com/ahron111/Coursera-IBM-Applied-Data-Science-Capstone

Data Collection - Scraping

IBM Developer
SKILLS NETWORK

1. HTML Response

```
# use requests.get() method with the provided static_url
# assign the response to a object
data = requests.get(static_url).text
```

2. Create BeautifulSoup

```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(data, 'html5lib')
```

3. Find all 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')
```

4. Get 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 row in first_launch_table.find_all('th'):
    name = extract_column_from_header(row)
    if (name != None and len(name) > 0):
        column_names.append(name)
```


https://github.com/ahron111/Coursera-IBM-Applied-Data-Science-Capstone

Data Collection - Scraping

5. Create Dictionary

```
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']=[]
```

6. Add data to keys

```
extracted_row = 0
#Extract each table
for table_number,table in enumerate(soup.find_all('table',"wikitable plainrowheaders collapsible")):
   # get table row
    for rows in table.find_all("tr"):
        #check to see if first table heading is as number corresponding to launch a number
        if rows.th:
            if rows.th.string:
                flight_number=rows.th.string.strip()
                flag=flight_number.isdigit()
        else:
            flag=False
        #get table element
        row=rows.find_all('td')
        #if it is number save cells in a dictonary
        if flag:
            extracted_row += 1
            # Flight Number value
            # TODO: Append the flight_number into launch_dict with key `Flight No.`
            launch_dict['Flight No.'].append(flight_number) #TODO-1
            #print(flight_number)
            datatimelist=date_time(row[0])
```

Full code on github

7. Create dataframe from dictionary

```
df=pd.DataFrame(launch_dict)
```

8. Export

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


https://github.com/ahron111/Coursera-IBM-Applied-Data-Science-Capstone

Data Wrangling

MISSION SUCCESS	True Ocean	True RTLS	True ASDS
MISSION FAILURE	False Ocean	False RTLS	False ASDS

1. Launches for each site

```
# Apply value_counts() on column LaunchSite
df['LaunchSite'].value_counts()
```

```
CCAFS SLC 40    55
KSC LC 39A      22
VAFB SLC 4E     13
Name: LaunchSite, dtype: int64
```

2. Number and occurrence for each orbit

```
# Apply value_counts on Orbit column
df['Orbit'].value_counts()
```

```
GTO      27
ISS      21
VLEO     14
PO        9
LEO       7
SSO       5
MEO       3
HEO       1
SO        1
GEO       1
ES-L1     1
Name: Orbit, dtype: int64
```


https://github.com/ahron111/Coursera-IBM-Applied-Data-Science-Capstone

Data Wrangling

3. Number and occurrence of outcome per orbit type

```
# landing_outcomes = values on Outcome column
landing_outcomes = df['Outcome'].value_counts()
landing_outcomes
```

```
True ASDS      41
None None      19
True RTLS      14
False ASDS      6
True Ocean      5
None ASDS       2
False Ocean     2
False RTLS      1
Name: Outcome, dtype: int64
```

4. Landing outcome label

```
# landing_class = 0 if bad_outcome
# landing_class = 1 otherwise

landing_class = []
for outcome in df['Outcome']:
    if outcome in bad_outcomes:
        landing_class.append(0)
    else:
        landing_class.append(1)
```

5. Export

```
df.to_csv("dataset_part_2.csv", index=False)
```


https://github.com/ahron111/Coursera-IBM-Applied-Data-Science-Capstone

EDA with Data Visualization

Scatter Plots

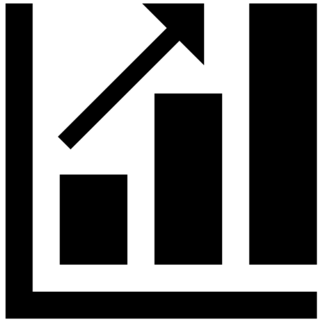
- Flight Number vs. Payload Mass
- Flight Number vs. Launch Site
- Payload vs. Launch Site
- Orbit vs. Flight Number
- Payload vs. Orbit Type
- Orbit vs. Payload Mass

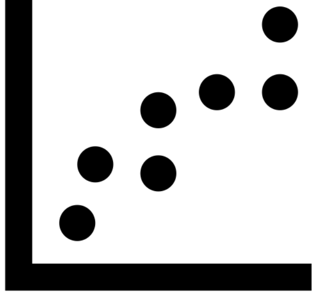
Bar Graphs

- Success Rate vs Orbit

Line Graphs

- Success Rate vs Year





https://github.com/ahron111/Coursera-IBM-Applied-Data-Science-Capstone

EDA with SQL

- Displaying the names of the unique launch sites in the space mission.
- Display 5 records where launch sites begin with the string 'CCA'
- Display the total payload mass carried by boosters launched by NASA (CRS).
- Display average payload mass carried by booster version F9 v1.1.
- List the date when the first successful landing outcome in ground pad was achieved.
- List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000.
- List the total number of successful and failure mission outcomes.
- List the names of the booster versions which have carried the maximum payload mass.
- List the records which will display the month names, failure landing outcomes in drone ship, booster versions, launch site for the months in year 2015.
- Rank the count of successful landing outcomes between the date 04-06-2010 and 20-03-2017 in descending order.

https://github.com/ahron111/Coursera-IBM-Applied-Data-Science-Capstone

Build an Interactive Map with Folium

Folium map object is a map centered on NASA Johnson Space Center at Houson, Texas

- Red circle at NASA Johnson Space Center's coordinate with label showing its name (*folium.Circle, folium.map.Marker*).
- Red circles at each launch site coordinates with label showing launch site name (*folium.Circle, folium.map.Marker, folium.features.DivIcon*).
- The grouping of points in a cluster to display multiple and different information for the same coordinates (*folium.plugins.MarkerCluster*).
- Markers to show successful and unsuccessful landings. Green for successful landing and Red for unsuccessful landing. (*folium.map.Marker, folium.Icon*).
- Markers to show distance between launch site to key locations (*railway, highway, coastway, city*) and plot a line between them. (*folium.map.Marker, folium.PolyLine, folium.features.DivIcon*)

Objects are created to understand the problem and data better. We can show easily all launch sites, their surroundings and the number of successful and unsuccessful landings.

https://github.com/ahron111/Coursera-IBM-Applied-Data-Science-Capstone

Build a Dashboard with Plotly Dash

Dashboard has dropdown, pie chart, rangeslider and scatter plots

- Dropdown allows a user to choose the launch site or all launch sites (*dash_core_components.Dropdown*).
- Pie chart shows the total success and the total failure for the launch site chosen with the dropdown component (*plotly.express.pie*).
- Rangeslider allows a user to select a payload mass in a fixed range (*dash_core_components.RangeSlider*).
- Scatter chart shows the relationship between two variables, in particular Success vs Payload Mass (*plotly.express.scatter*).

https://github.com/ahron111/Coursera-IBM-Applied-Data-Science-Capstone

Predictive Analysis (Classification)

1 Data Preparation

- Load dataset
- Normalize fata
- Perform train-test split

2 Model Preparation

- Select machine learning algorithms
- Set parameters for each algorithm to GridSearchCV
- Train GridSearchModel models with training dataset

3 Model Evaluation

- Get best hyperparameters for each model
- Compute accuracy for each model with test dataset
- Plot Confusion Matrix

4 Model Comparison

- Comparison of models based on their accuracy
- The model with the best accuracy will be chosen

https://github.com/ahron111/Coursera-IBM-Applied-Data-Science-Capstone

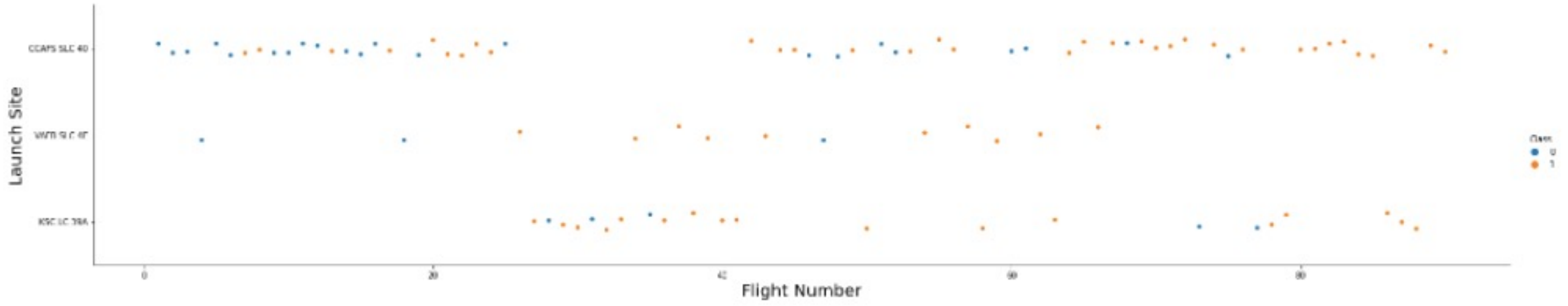
Results

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

Section 2

Insights drawn from EDA

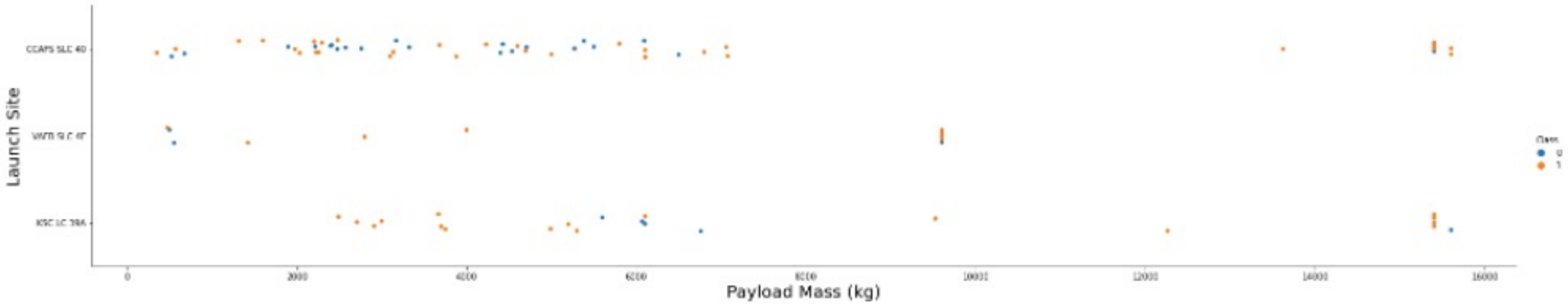
Flight Number vs. Launch Site



Success rate increases over time for each site

https://github.com/ahron111/Coursera-IBM-Applied-Data-Science-Capstone

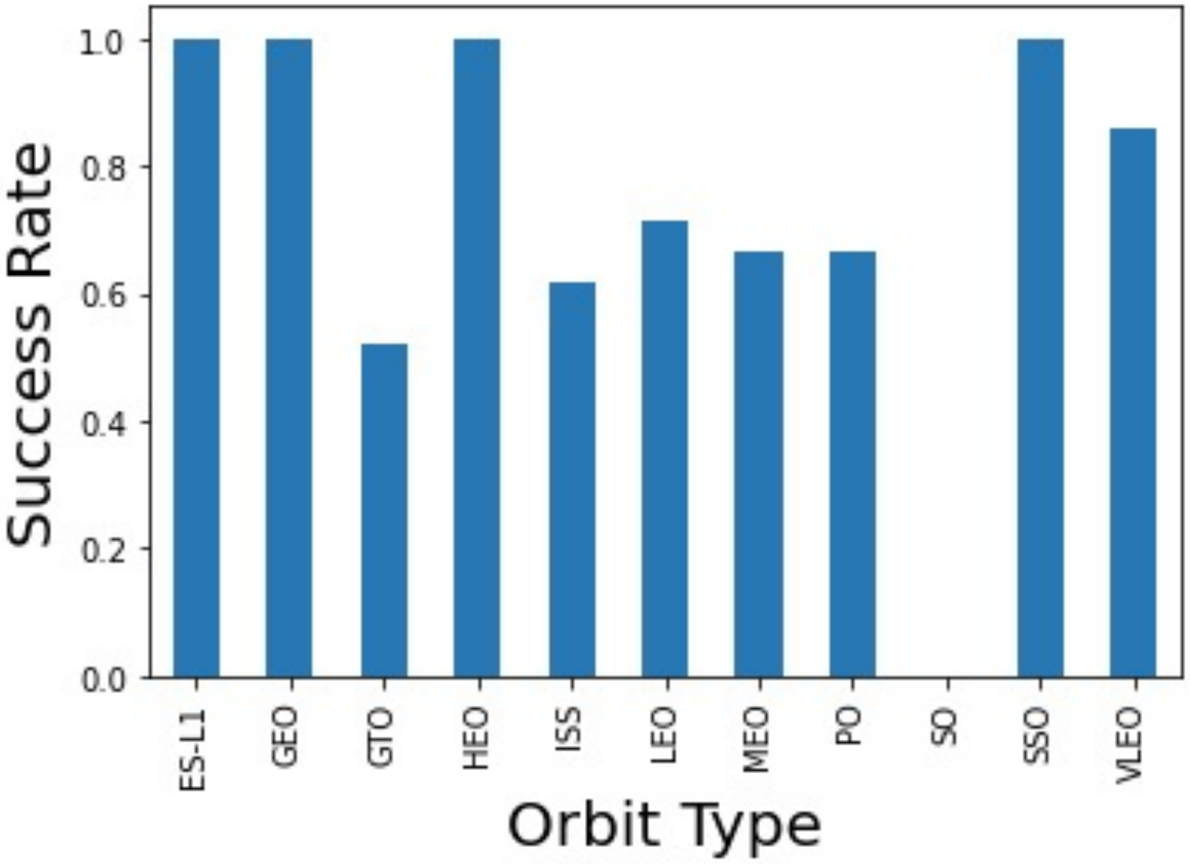
Payload vs. Launch Site



Different launch sites have their own payload constraints. The payload can also affect the probability of a successful landing.

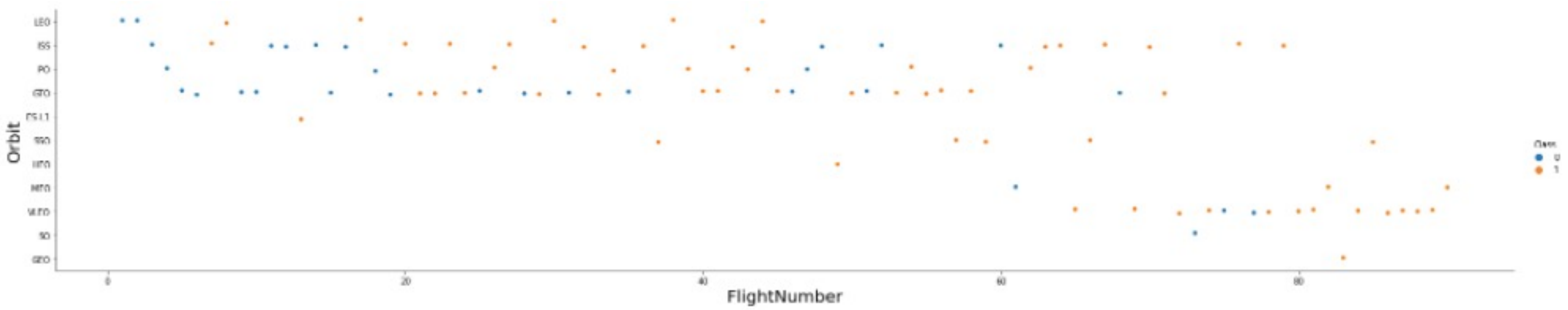
https://github.com/ahron111/Coursera-IBM-Applied-Data-Science-Capstone

Success Rate vs. Orbit Type



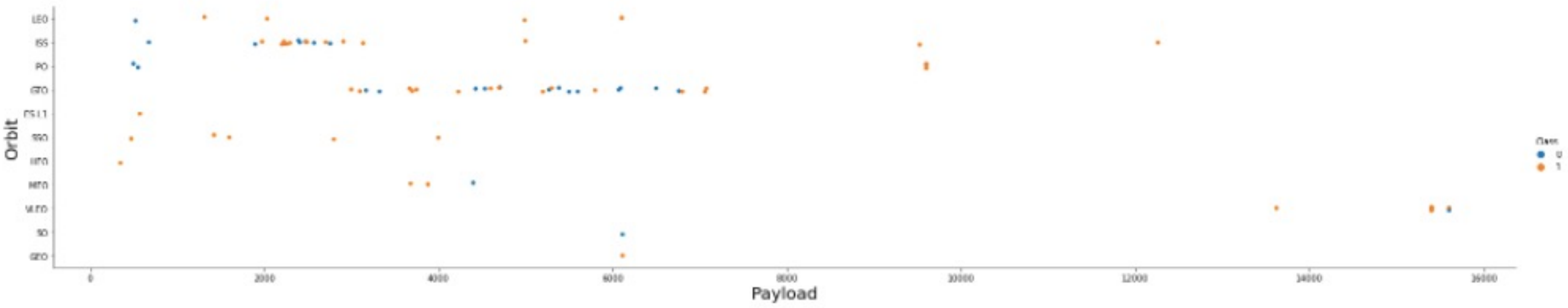
https://github.com/ahron111/Coursera-IBM-Applied-Data-Science-Capstone

Flight Number vs. Orbit Type



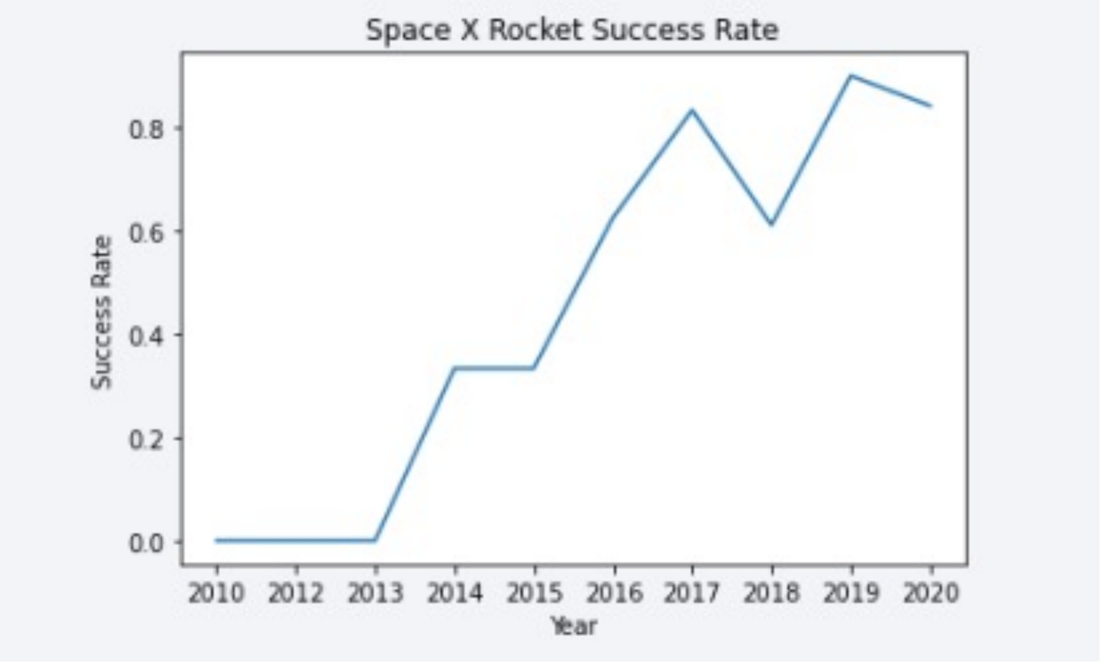
https://github.com/ahron111/Coursera-IBM-Applied-Data-Science-Capstone

Payload vs. Orbit Type



https://github.com/ahron111/Coursera-IBM-Applied-Data-Science-Capstone

Launch Success Yearly Trend



From 2013, the success rate of the SpaceX Rocket has been increasing.

https://github.com/ahron111/Coursera-IBM-Applied-Data-Science-Capstone

All Launch Site Names

SQL Query

```
%sql SELECT DISTINCT "LAUNCH_SITE" FROM SPACEXTBL
```

Explanation

DISTINCT is used to remove duplicate LAUNCH_SITE in the data

Results

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

https://github.com/ahron111/Coursera-IBM-Applied-Data-Science-Capstone

Launch Site Names Begin with 'CCA'

IBM Developer
SKILLS NETWORK

SQL Query

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

Results

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

Explanation

WHERE clause followed by LIKE clause filters launch sites that contain the substring CCA. LIMIT 5 shows 5 records from filtering.

https://github.com/ahron111/Coursera-IBM-Applied-Data-Science-Capstone

Total Payload Mass

SQL Query

```
%sql SELECT SUM("PAYLOAD_MASS__KG_") FROM SPACEXTBL WHERE "CUSTOMER" = 'NASA (CRS)'
```

Results

SUM("PAYLOAD_MASS__KG_")
45596

Explanation

The query returns the sum of all payload masses where the customer is NASA (CRS)

https://github.com/ahron111/Coursera-IBM-Applied-Data-Science-Capstone

Average Payload Mass by F9 v1.1

SQL Query

```
%sql SELECT AVG("PAYLOAD_MASS__KG_") FROM SPACEXTBL WHERE "BOOSTER_VERSION" LIKE '%F9 v1.1%'
```

Results

AVG("PAYLOAD_MASS__KG_")
2534.6666666666665

Explanation

The query returns the average of all payload masses where the booster version contains the substring "F9 v1.1"

https://github.com/ahron111/Coursera-IBM-Applied-Data-Science-Capstone

First Successful Ground Landing Date

SQL Query

```
%sql SELECT MIN("DATE") FROM SPACEXTBL WHERE "Landing _Outcome" LIKE '%Success%'
```

Results

MIN("DATE")
01-05-2017

Explanation

This query selects the oldest successful landing. The WHERE clause filters the dataset to keep only records where there was a successful landing. The MIN function selects the oldest record.

https://github.com/ahron111/Coursera-IBM-Applied-Data-Science-Capstone

Successful Drone Ship Landing with Payload between 4000 and 6000

IBM Developer
SKILLS NETWORK

SQL Query

```
%sql SELECT "BOOSTER_VERSION" FROM SPACEXTBL WHERE "LANDING _OUTCOME" = 'Success (drone ship)' \
AND "PAYLOAD_MASS__KG_" > 4000 AND "PAYLOAD_MASS__KG_" < 6000;
```

Results

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

Explanation

This query returns the booster version where landing was successful and payload mass is between 4000 and 6000 kg. The WHERE and AND clauses filter the dataset.

https://github.com/ahron111/Coursera-IBM-Applied-Data-Science-Capstone

Total Number of Successful and Failure Mission Outcomes

IBM Developer SKILLS NETWORK

SQL Query

```
%sql SELECT (SELECT COUNT("MISSION_OUTCOME") FROM SPACEXTBL WHERE "MISSION_OUTCOME" LIKE '%Success%') AS SUCCESS, \
(SELECT COUNT("MISSION_OUTCOME") FROM SPACEXTBL WHERE "MISSION_OUTCOME" LIKE '%Failure%') AS FAILURE
```

Results

SUCCESS	FAILURE
100	1

Explanation

With the first SELECT, we show the subqueries that return results. The first subquery counts the successful mission. The second subquery counts the unsuccessful mission. The WHERE clause followed by LIKE clause filters mission outcome. The COUNT function counts records filtered.

https://github.com/ahron111/Coursera-IBM-Applied-Data-Science-Capstone

Boosters Carried Maximum Payload

SQL Query

```
%sql SELECT DISTINCT "BOOSTER_VERSION" FROM SPACEXTBL \
WHERE "PAYLOAD_MASS__KG_" = (SELECT max("PAYLOAD_MASS__KG_") FROM SPACEXTBL)
```

Results

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

Explanation

We used a subquery to filter data which returns only the heaviest payload mass with MAX function. The main query uses subquery results and returns unique booster version (SELECT DISTINCT) with the heaviest payload mass.

https://github.com/ahron111/Coursera-IBM-Applied-Data-Science-Capstone

2015 Launch Records

SQL Query

```
%sql SELECT substr("DATE", 4, 2) AS MONTH, "BOOSTER_VERSION", "LAUNCH_SITE" FROM SPACEXTBL\
WHERE "LANDING _OUTCOME" = 'Failure (drone ship)' and substr("DATE",7,4) = '2015'
```

Results

MONTH	Booster_Version	Launch_Site
01	F9 v1.1 B1012	CCAFS LC-40
04	F9 v1.1 B1015	CCAFS LC-40

Explanation

The query returns month, booster version, launch site where landing was unsuccessful and landing date took place in 2015. The Substr function processes date in order to take month or year. Substr(DATE, 4, 2) shows month. Substr(DATE,7, 4) shows year.

https://github.com/ahron111/Coursera-IBM-Applied-Data-Science-Capstone

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

IBM Developer
SKILLS NETWORK

SQL Query

```
%sql SELECT "LANDING _OUTCOME", COUNT("LANDING _OUTCOME") FROM SPACEXTBL\
WHERE "DATE" >= '04-06-2010' and "DATE" <= '20-03-2017' and "LANDING _OUTCOME" LIKE '%Success%'\
GROUP BY "LANDING _OUTCOME" \
ORDER BY COUNT("LANDING _OUTCOME") DESC ;
```

Results

Landing _Outcome	COUNT("LANDING _OUTCOME")
Success	20
Success (drone ship)	8
Success (ground pad)	6

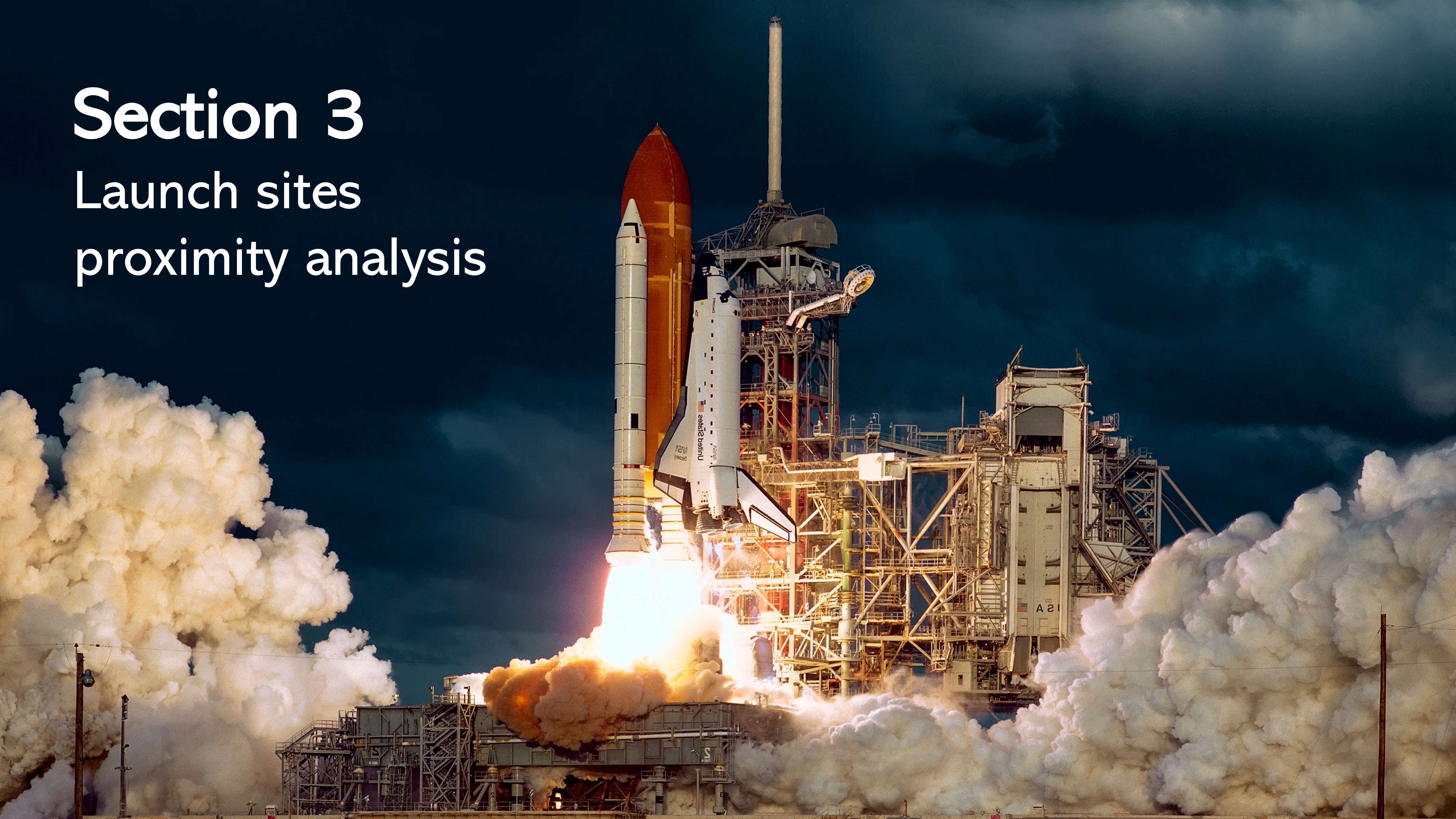
Explanation

The query returns landing outcomes and their count where mission was successful and date is between 04/06/2010 and 20/03/2017. The GROUP BY clause groups results by landing outcome and ORDER BY COUNT DESC shows results in decreasing order.

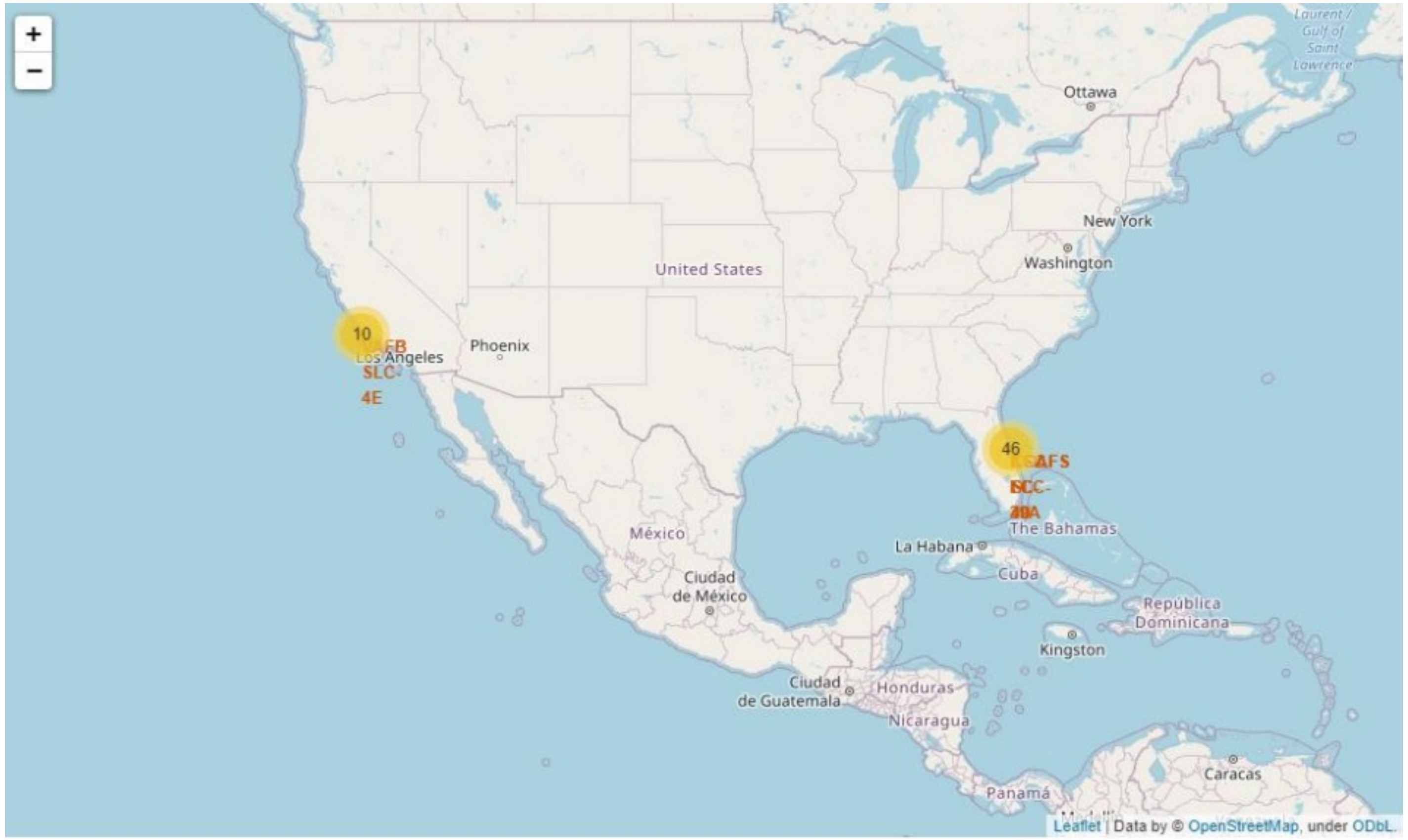
https://github.com/ahron111/Coursera-IBM-Applied-Data-Science-Capstone

Section 3

Launch sites proximity analysis



Folium – Ground stations



Launch sites are located along the coast of the United states

https://github.com/ahron111/Coursera-IBM-Applied-Data-Science-Capstone

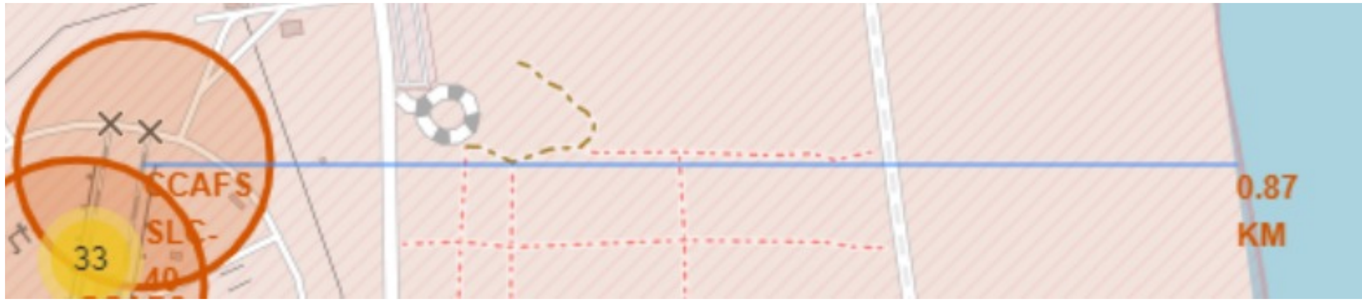
Folium – Colored Labelled Markers

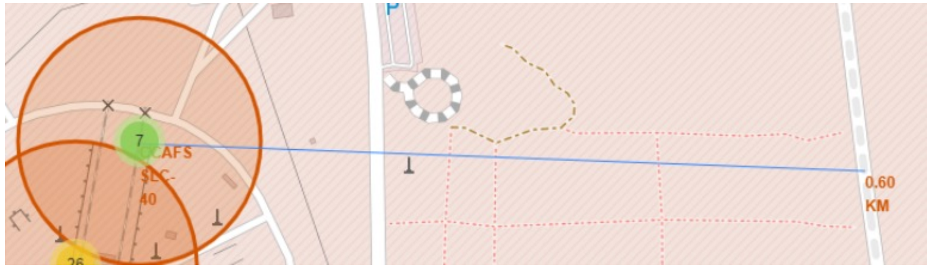
IBM Developer
SKILLS NETWORK

KSC LC-39A has the highest launch success rate

https://github.com/ahron111/Coursera-IBM-Applied-Data-Science-Capstone

Folium – CCAFS SLC-40 Proximities





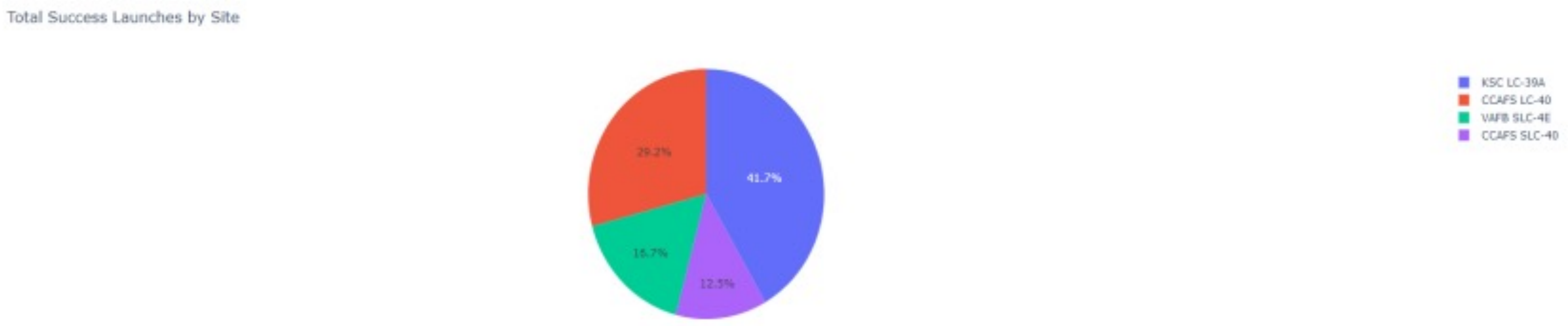
CCAFS SLC-40 is close to railways, highways, and coastlines. ***BUT*** it is not close to cities

https://github.com/ahron111/Coursera-IBM-Applied-Data-Science-Capstone

Section 4

Build a dashboard with Plotly Dash

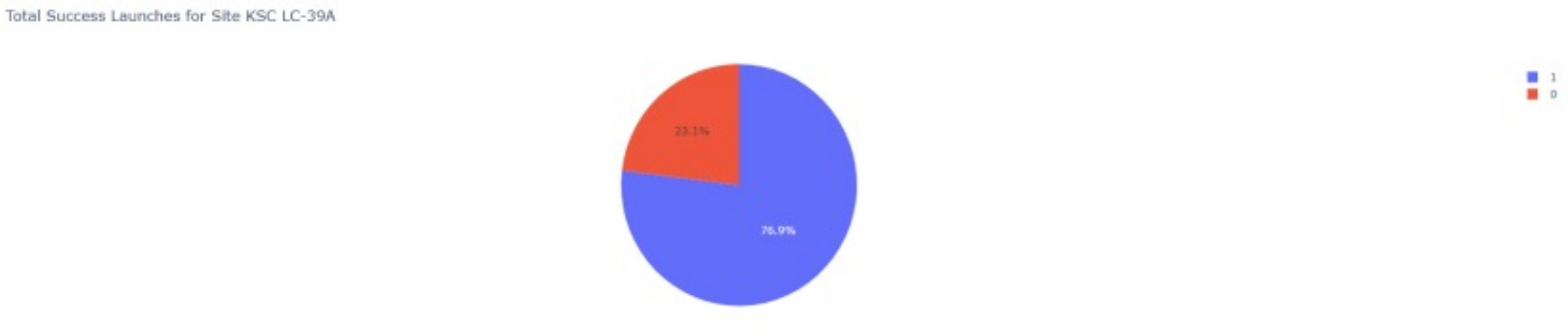
Dashboard – Total Success Launches (Site)



KSC LC-39A has the highest success for its launches

https://github.com/ahron111/Coursera-IBM-Applied-Data-Science-Capstone

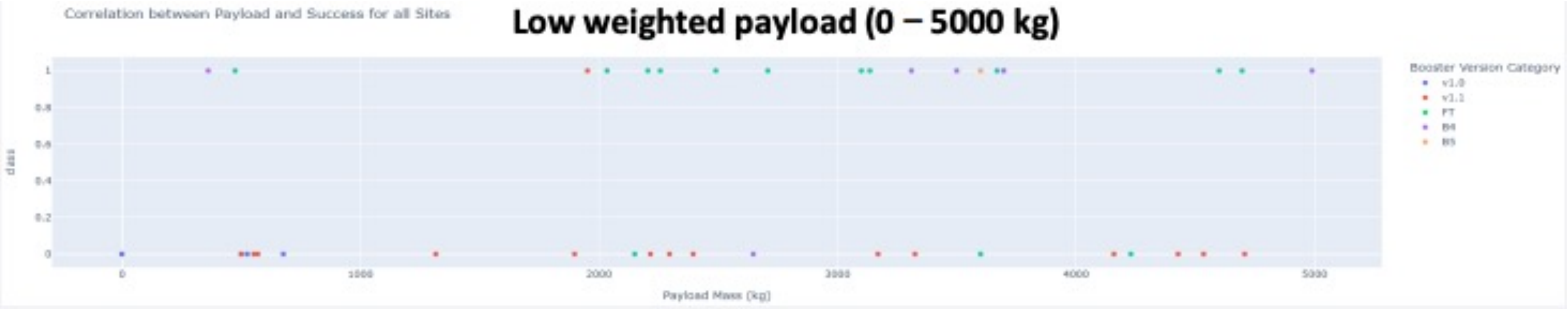
Dashboard – Total Success Launches (KSC LC-39A)

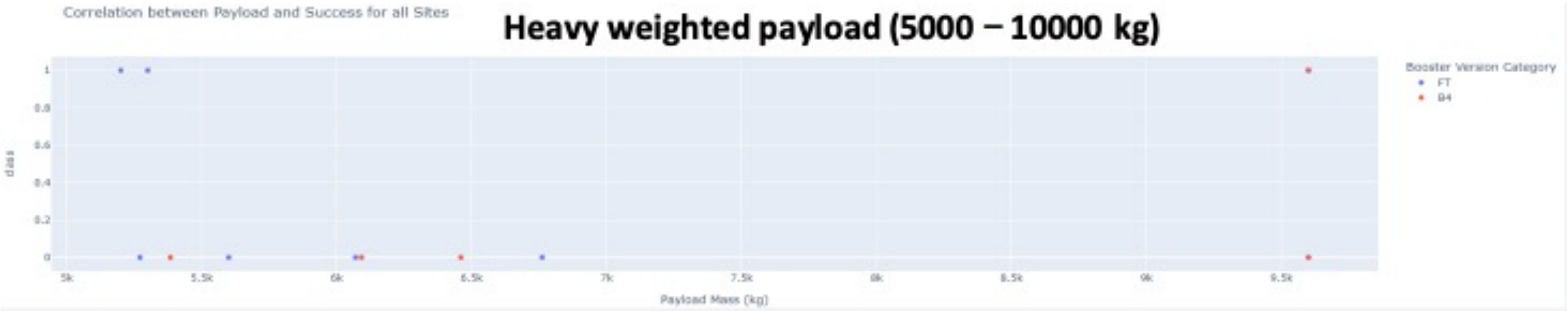


KSC LC-39A has a success rate of 76.9%.

https://github.com/ahron111/Coursera-IBM-Applied-Data-Science-Capstone

Dashboard – Payload mass v. Outcome for all sites and different payloads





Lower weighted payloads have a better success rate compared to heavy weighted payloads.

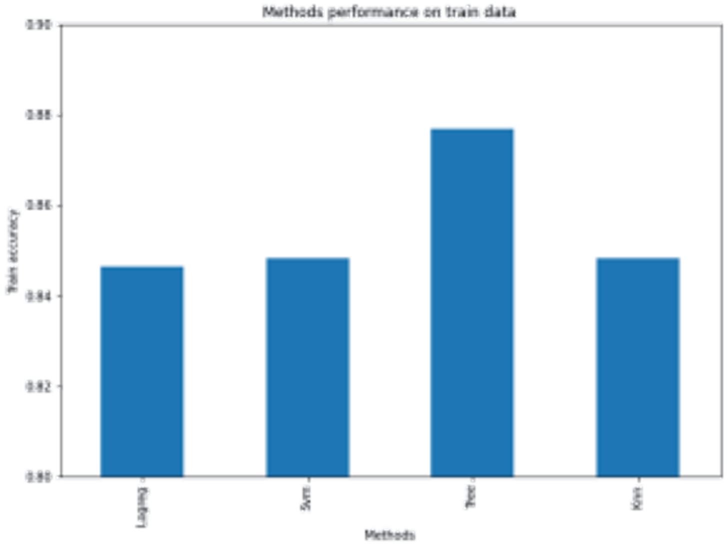
https://github.com/ahron111/Coursera-IBM-Applied-Data-Science-Capstone

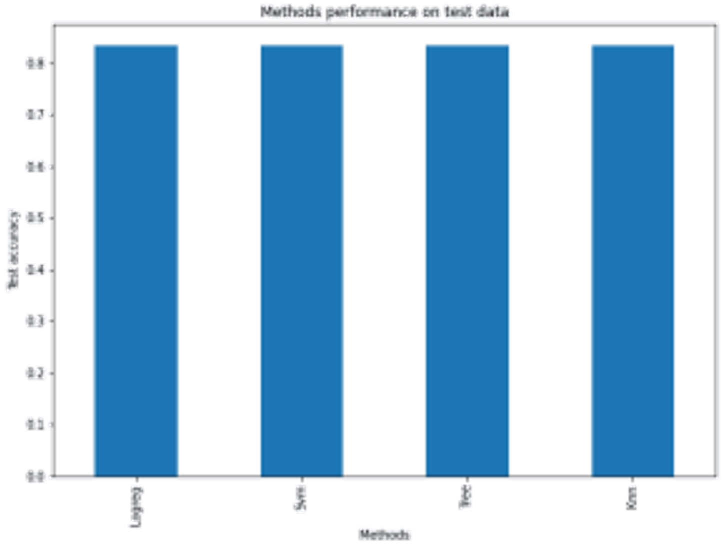
Section 5

Predictive analysis (Classification)

Classification Accuracy

Bar Charts





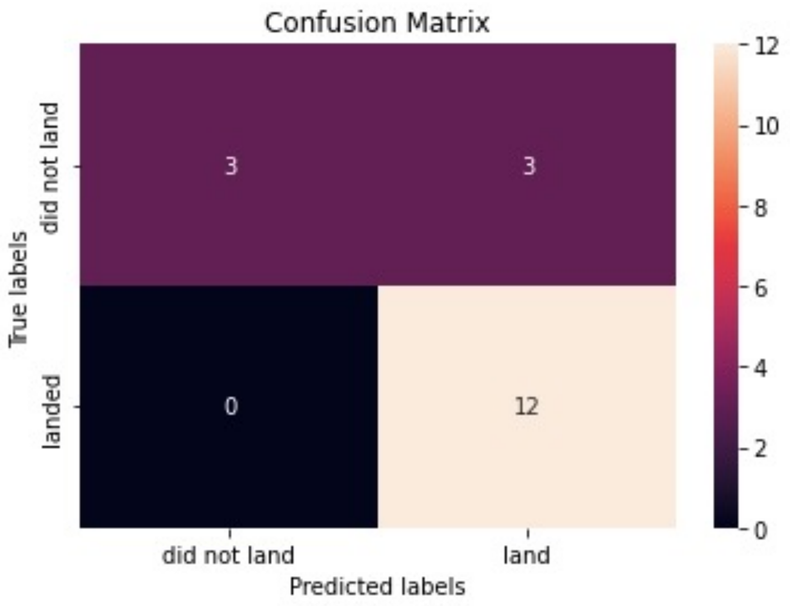
Results Table

	Accuracy Train	Accuracy Test
Tree	0.876786	0.833333
Knn	0.848214	0.833333
Svm	0.848214	0.833333
Logreg	0.846429	0.833333

All machine learning methods produced similar accuracy on the test data

https://github.com/ahron111/Coursera-IBM-Applied-Data-Science-Capstone

Confusion Matrix



The confusion matrix for all the models are the same making it difficult to choose the optimal one

The key problem highlighted from the matrix is the high false positive rate

https://github.com/ahron111/Coursera-IBM-Applied-Data-Science-Capstone

Conclusions

- Several key factors can help explain a successful mission. These are the launch site, orbit, and the number of previous launches.
- Orbits which have the best success rate are: GEO, HEO, SSO, ES-L1
- Depending on the orbits, the payload mass can be a critical for the success of a mission. Different orbits require a light or heavy payload mass. Generally, low weighted payloads perform better than the heavy weighted payloads.

Decision Tree Algorithm is chosen as the desired model as it has the best train accuracy, given all models have identical test accuracies.

Appendix

Github Link:

https://github.com/ahron111/Coursera-IBM-Applied-Data-Science-Capstone

https://github.com/ahron111/Coursera-IBM-Applied-Data-Science-Capstone

Thank you!