IBM Developer
SKILLS NETWORK

# Winning Space Race with Data Science


**Nianci Ma**


**31 Jan 2023**

# Outline

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

# Executive Summary

- **Summary of methodologies**
  - ❑Data Collection
  - ❑Data Wrangling
  - ❑Exploratory Data Analysis
  - ❑Interactive Visual Analytics
  - ❑Predictive Analysis
- **Summary of all results**
  - ❑Exploratory Data Analysis(EDA)
  - ❑Geospatial analytics
  - ❑Interactive dashboard
  - ❑Predictive analysis of classification models

# Introduction

- Project background and context
    - ❑SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars, which is much saving then others because of the reuse first stage in the rocket
    - ❑Whether Falcon can successfully land? That’s what we need to predict.
- Problems you want to find answers
    - ❑Correlations between rocket variables and successful landing rate
    - ❑Different conditions to get the best results and ensure the best successful landing rate

Section 1

# Methodology

# Methodology

## Executive Summary

- **Data collection methodology:**
  - Making GET requests to the SpaceX REST API
  - Web Scraping
- **Perform data wrangling**
  - Using .fillna() mehtod to remove NAN
  - Using .value_counts() to determine :
    - Number of launches on each site
    - Nmber and occurrence of each orbit
    - Number and occurrence of mission outcome per orbit typ
- **Creating a landing** outcome label that shows the following (0 unsuccessful,  1 successful)

# Methodology

- **Perform exploratory data analysis (EDA) using visualization and SQL**
  - Using SQL to manipulate and evaluate the SpaceX dataset
  - Using Pandas and Matplotlib to visualize relationships between variables and determine patterns
- **Perform interactive visual analytics using Folium and Plotly Dash**
  - Geospatial analytics using Folium
  - Creating an interactive dashboard using Plotyly Dash
- **Perform predictive analysis using classification models**
  - Using Ski-learn
    - pre-process(standardize) the data; Split the data into training and testing set; Train different classsification models; Find hyperparameters using GridSearchCV
  - Plotting confusion matrices for each classification model
  - Assessing the acuracy of each classification model

# Data Collection

1.Lunch data from SpaceX API

This image cannot currently be displayed.

2. Convert response to JSON file

```
data = pd.json_normalize(response.json())
```

3. Use custom functions to clean data

```
# Call getBoosterVersion
getBoosterVersion(data)
```

```
# Call getPayloadData
getPayloadData(data)
```

```
# Call getLaunchSite
getLaunchSite(data)
```

```
# Call getCoreData
getCoreData(data)
```

# Data Collection

**4. Combine columns**

```
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}
```

```
launch_df = pd.DataFrame.from_dict(launch_dict)
```

**5. Filter dataframe and exporting to CSV**

```
data_falcon9 = launch_df[launch_df['BoosterVersion'] == 'Falcon 9']
```

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

# Data Collection – SpaceX API

## 1. Load rocket data from SpaceX API

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

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

## 2. Convert response to JSON

```
data = pd.json_normalize(response.json())
```

## 3. Clean data using custom functions

```
# Call getBoosterVersion
getBoosterVersion(data)
```

```
# Call getPayloadData
getPayloadData(data)
```

```
# Call getLaunchSite
getLaunchSite(data)
```

```
# Call getCoreData
getCoreData(data)
```

# Data Collection – SpaceX API

**4. Create data frame**

```
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}
```

```
launch_df = pd.DataFrame.from_dict(launch_dict)
```

**5. Filter data and export a CSV**

```
data_falcon9 = launch_df[launch_df['BoosterVersion'] == 'Falcon 9']
```

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

# Data Collection - Scraping

**1. Request HTML page**

```
html_data = requests.get(static_url).text
```

**2. Create a BeautifulSoup**

```
soup = BeautifulSoup(html_data, 'html5lib')
```

**3. Assign the all table result to a list**

```
html_tables = soup.find_all('table')
```

**4. Extract columns name**

```
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)
```

# Data Collection - Scraping

**5. Create a dictionary for combine data**

**6. Fill  up the data in the dictionary**

**7. Create a new dataframe and export to CSV**

```
launch_dict= dict.fromkeys(column_names)
# Remove an irrelvant column
del launch_dict['Date and time ( )']
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']=[]
```

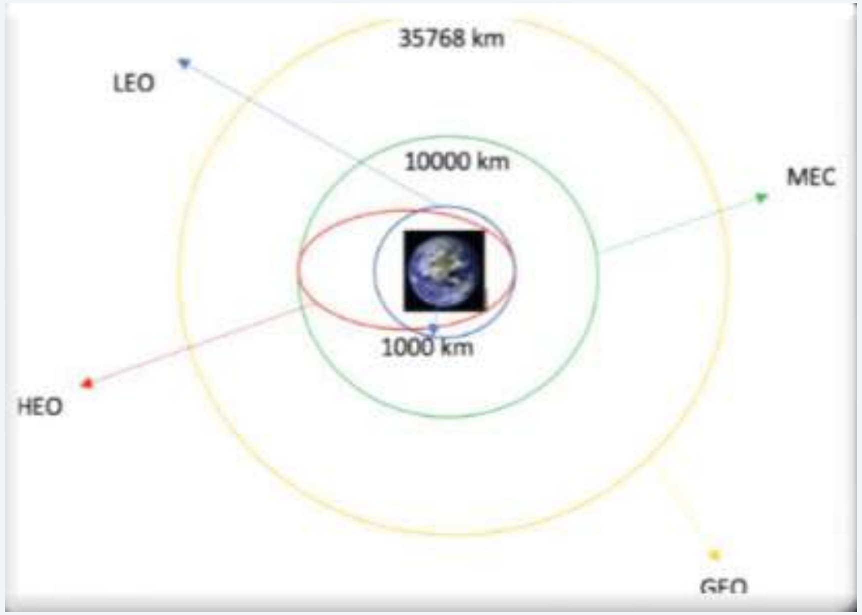
```
df=pd.DataFrame(launch_dict)
```

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

# Data Wrangling

The SpaceX dataset contains several SpaceX launch facilities, and all this data in column

Every launch aims in order to a dedicated orbit, and some of common orbit types are shown in the picture below.




Data Exploration with .value_counts()

# Data Wrangling

True Ocean – the mission result has successfully landed in a specific area of ocean.
False Ocean – the mission result has unsuccessfully landed in a specific area of the ocean
True RTLS - the mission result has successfully landed on the ground pad
False RTLS – the mission result has unsuccessfully landed on the ground pad
True ASDS – the mission result has successfully landed on the drone ship
False ASDS – the mission result has not landed on the drone ship

1 = successful 0 = failure

# EDA with Data Visualization

## Scatter Charts

Scatter charts were produced relationships in:

- Flight number and Launch site
- Payload and Launch site
- Orbit type and Flight number
- Payload and Orbit type

## Bar Chart

Bar chart was produced to visualize
The relationship between:

Success rate and Orbit type

## Line Charts

Line charts were produced to
Visualize the relationship between:

Success rate and Year

# EDA with SQL

Displaying the names of the unique launch sites in the space mission
Displaying 5 records where launch sites begin with the string ’CCA’
Displaying the total payload mass carried by boosters launched by NASA(CRS)
Displaying ave payload mass carried by booster version F9 v1.1
Listing the date when the first successful landing outcome in ground pad was achieved
Listing the names of the boosters which habe success in drone ship and have payload mass greater then 4000 but less Than 6000
Lising 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 failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
Rainking the count  of landing outcomes between te date 4$^{th}$ Jun 2010 and 20$^{th}$ Mar 2017

# Build an Interactive Map with Folium

1. Mark all launch sites on the map
   1. Initialise the map using a folium map object
   2. Add a folium.circle and olium.Marker for each launch site on the map
2. Mark the success, failed launches for each site on the map
   1. As many launches have the same coordinates, it makes sense to cluster together
   2. Before clustering them, assign a marker color of successful = 1 failed = 0
   3. To put the launches into clusters, for each launch, add a folium.Marker to the MarkerCluster() object
   4. Create an Icon as a text label., assigning the icon_color as the marker_colour determined previously.
3. Calculate the distances between a launch site to its proximities
   1. To explore the proximities of launch sites, calculation of distances between points can be mmade using the Lat and Long values
   2. After marking a point using the Lat and Long values, create a folium.Marker object to show the distance
   3. To display the distance line betwween two points, draw a folium.Polyline and add this to the map

# Build a Dashboard with Plotly Dash

- Pie chart
  - For shiowing total success launches by sites
  - This chart can be selected to indicate a successful landing distribution across all launch sites or to indicate the successrate of individual launch sites.
- Scatter chart
  - For showing the relationship between Outcomes and Payload mass by different boosters
  - 2 inputs: all sites,individual site & Payload mass on a slider in 0 – 10000kg
  - This chart helps determine how success depends on the launch point, payload mass, and booster version categories.

# Predictive Analysis (Classification)

## Model Development

- To prepare the dataset
    - Load dataset
    - Perform necessary data transformations
    - Split data into  traning and testing set
    - Decide which type of ML algorithms are fit
- For eachchosen
    - Create a GridSearchCV and dictionary of
  parameters
    - Fit the object to the parameters
    - Use the braining data set to train model

## Model Evaluation

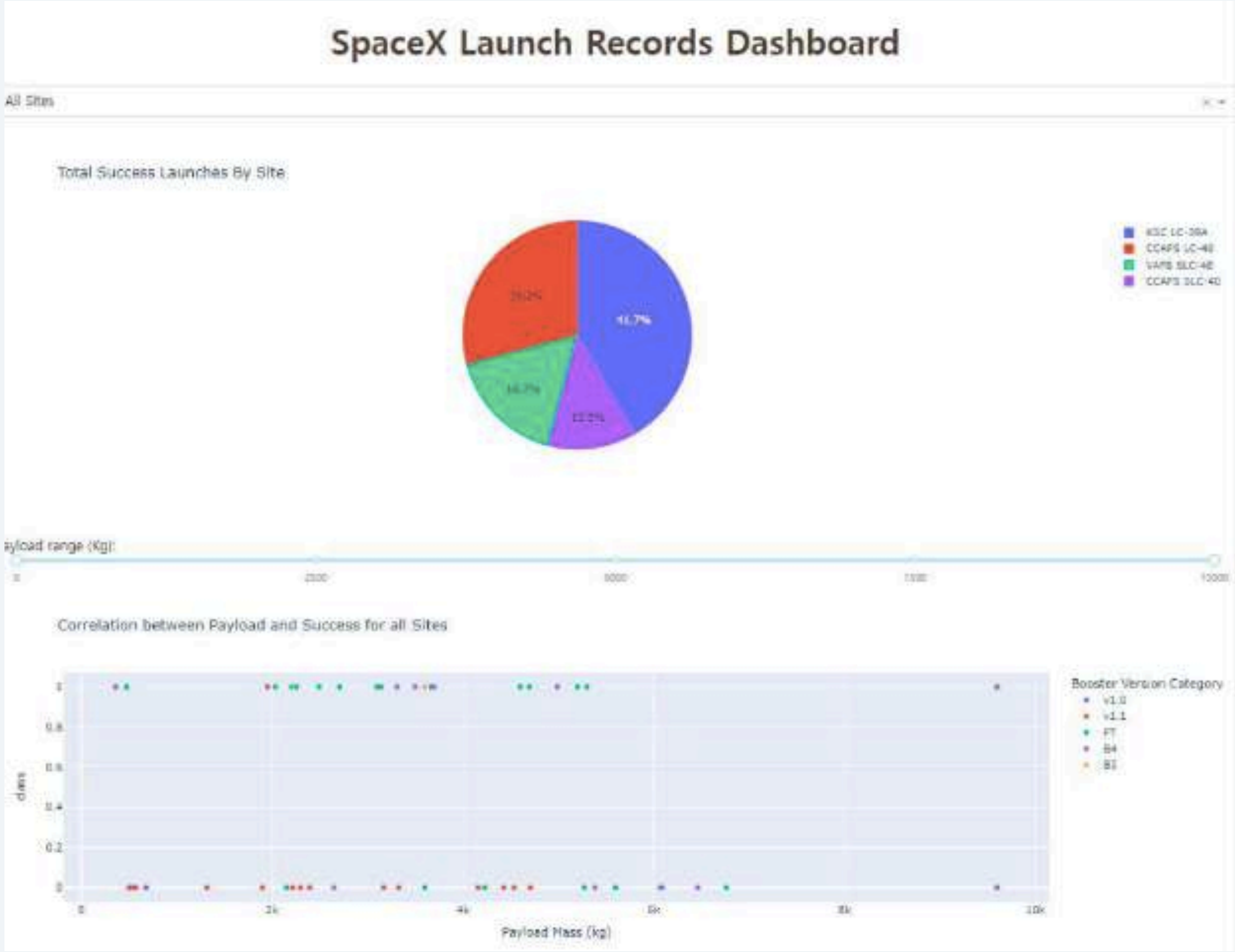
- Using the output GridSearchCV
- Check the turned hyperparameters
- Check the accuracy
- Plot and examine the Confusion Matrix

## Finding the best fit Model

- Review the accuracy for all chosen algorithms
- The model with the highest acuracy score is determined a the best performing model

# Results

- A preview of the Dashboard with Plotly Dash
- The results of EDA with visualization, EDA with SQL, interactive Map with Folium and Interactive Dashboard will be shown in the next slides
- Comparing the accuracy of the four methods, we can see, all return the same accuracy of about 83% for test data

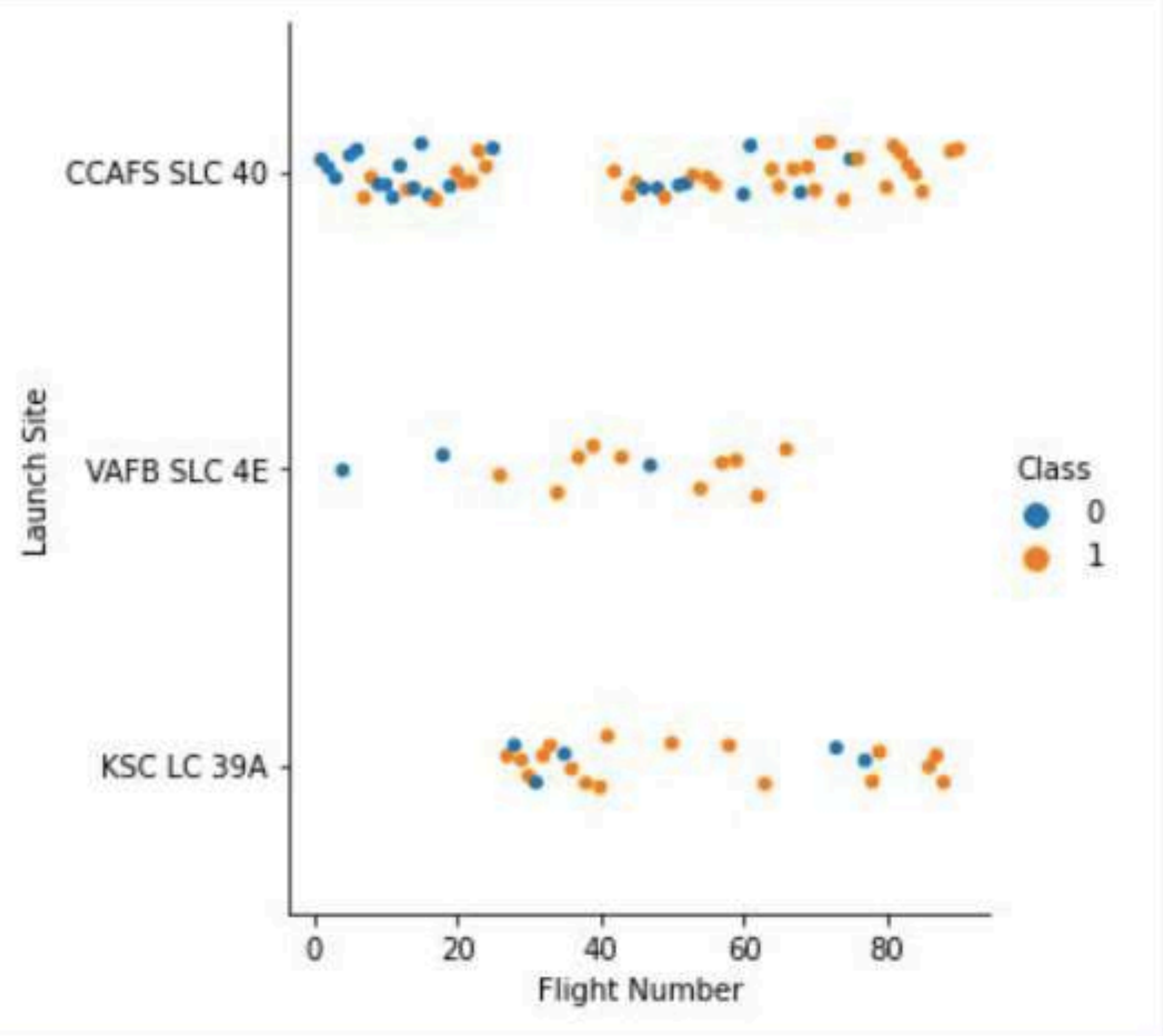


Section 2

# Insights drawn from EDA

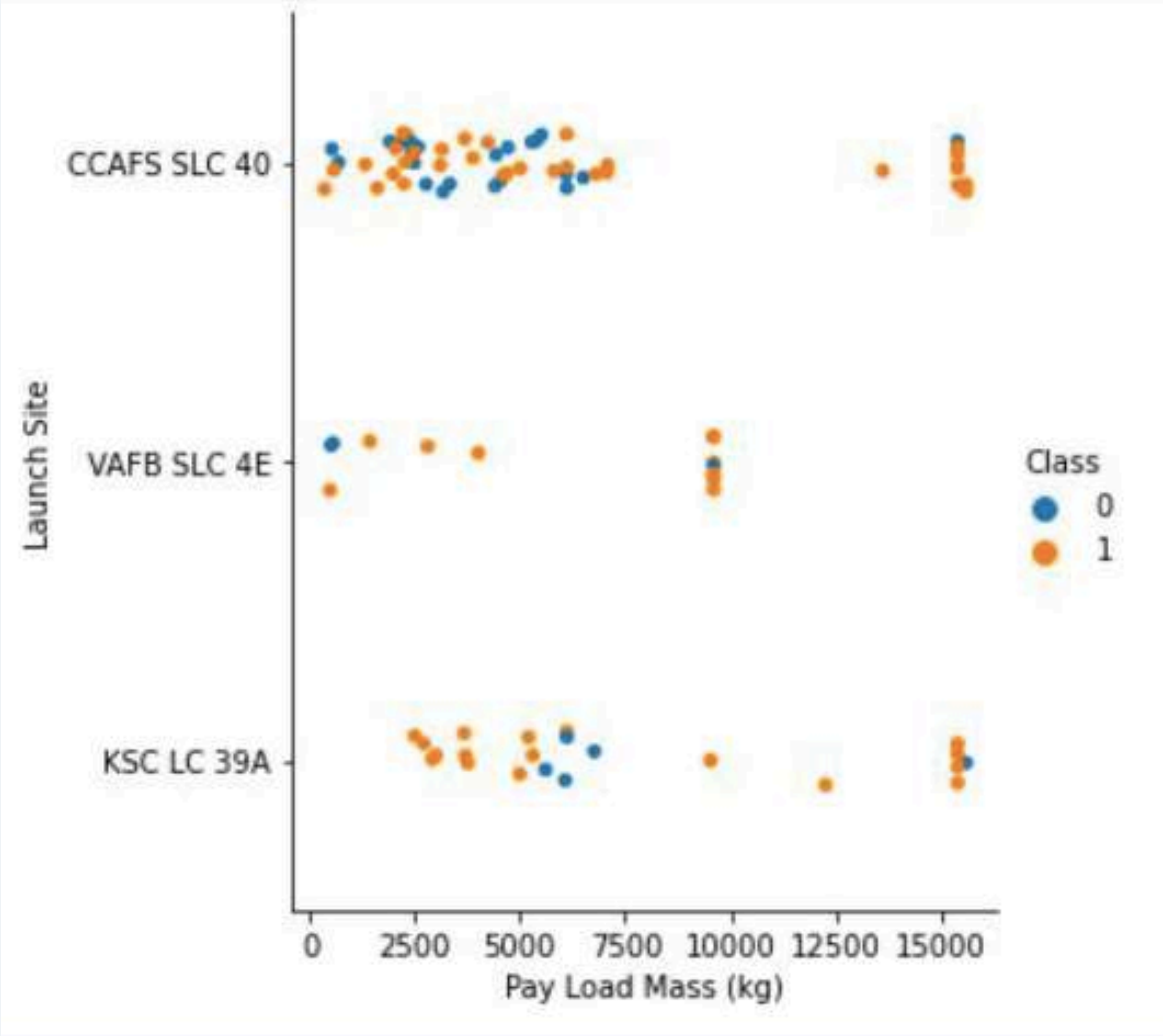
# Flight Number vs. Launch Site

- As the number of flights increases, the rate of success at a launch site increases.
- Most of the early flights <30 were launched from CCAFS SLC 40, and were generally unsuccessful
- The flights from VAFB SLC 4E shows the trend is earlier flights were less successful
- No early flights were launched from KSC LC 39A, so the launches from this site are more successful
- Above a flight number of around 30, there are significantly more successful landings



# Payload vs. Launch Site

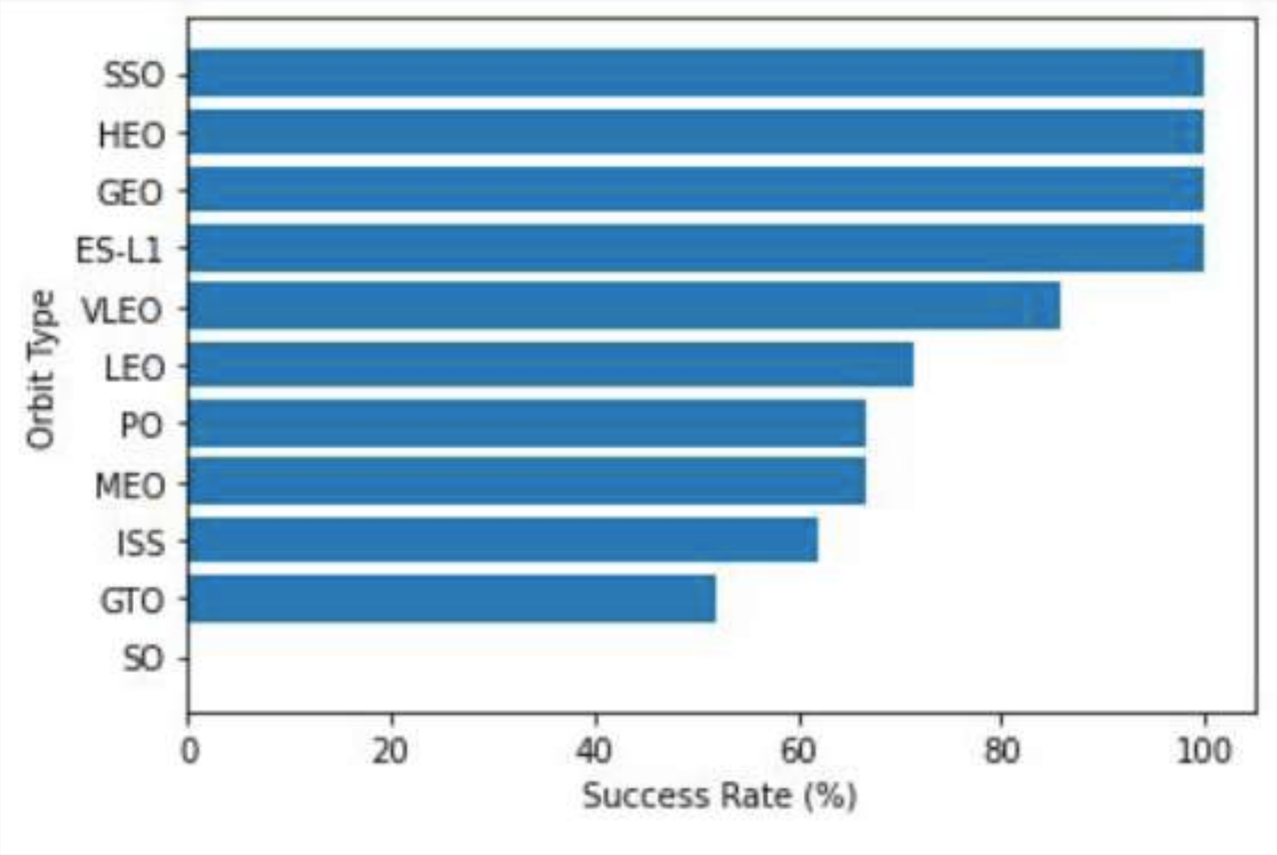
- Above a payload mass of around 7000kg, there are very few unsuccessful landings, but it also far less data for these heavier launches.
- There is no clear correlation between payload mass and success rate for a given launch site.
- All sites launched a bariety of payload masses, with most of the launches from CCAFS SLC 40



# Success Rate vs. Orbit Type

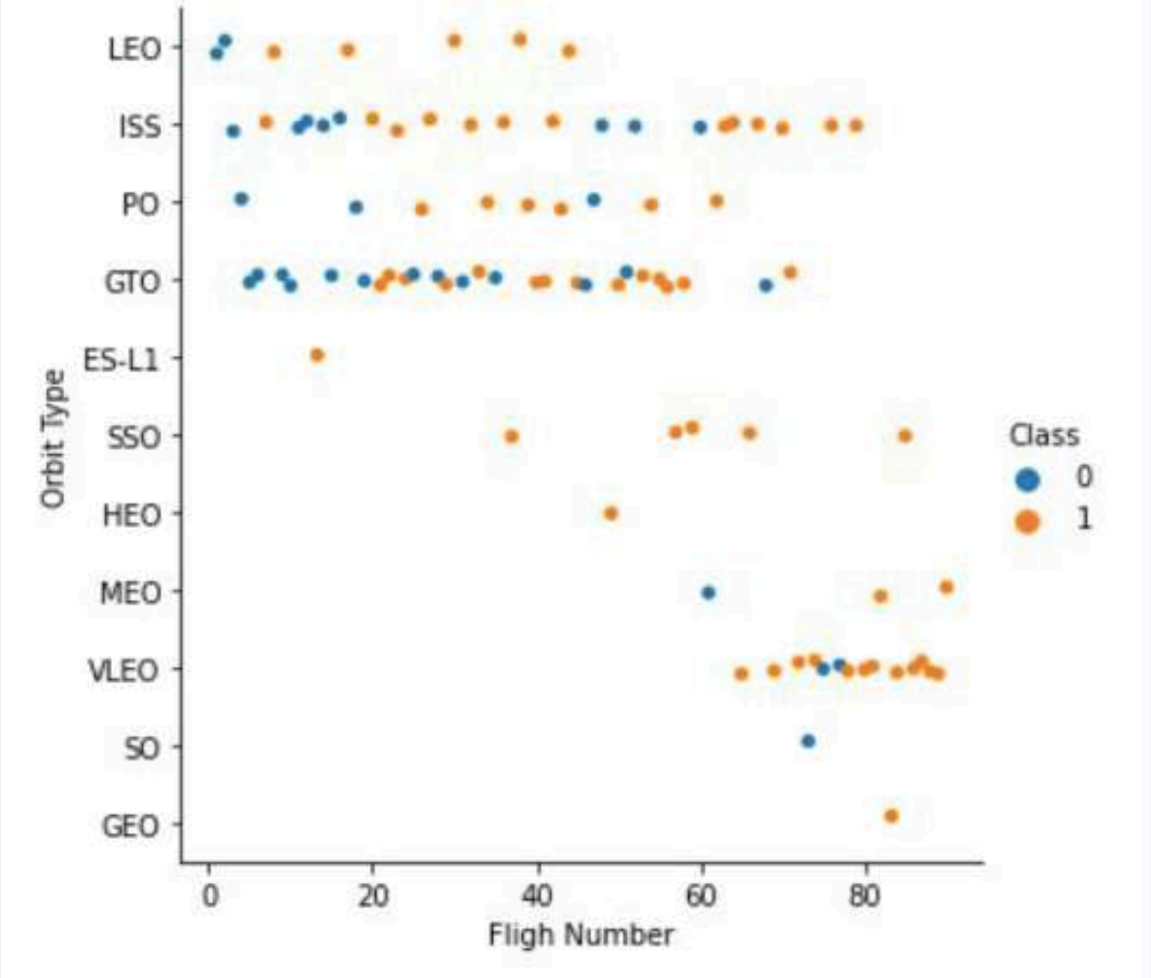
- Orbits has 100% success rate
    - ES-L1
    - GEO
    - HO
    - SSO

- The orbit wit hthe lowest success rate
    - SO



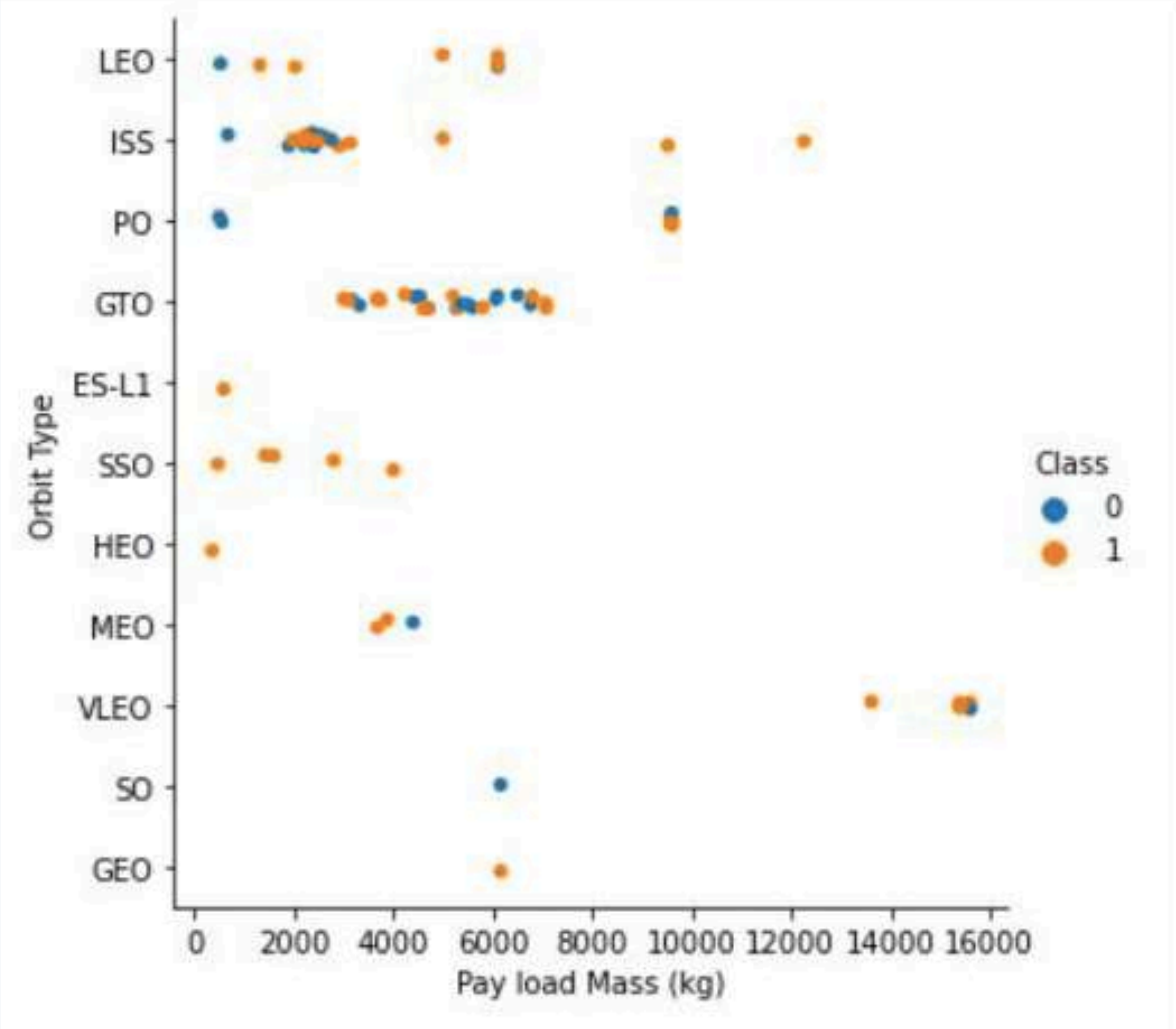
# Flight Number vs. Orbit Type

- 100% success rate of GEO, HEO, and ES-L1 orbits can be explained by only having 1 flight into the respective orbits.
- The 100% success rate in SSO is more impressive, with 5 successful flights.
- Weak relationship in Flight number and success rate for GTO
- Flight number increases, success rate increases. This is most extreme for LEO, where unsuccessful landings only occurred for the low flight numbers



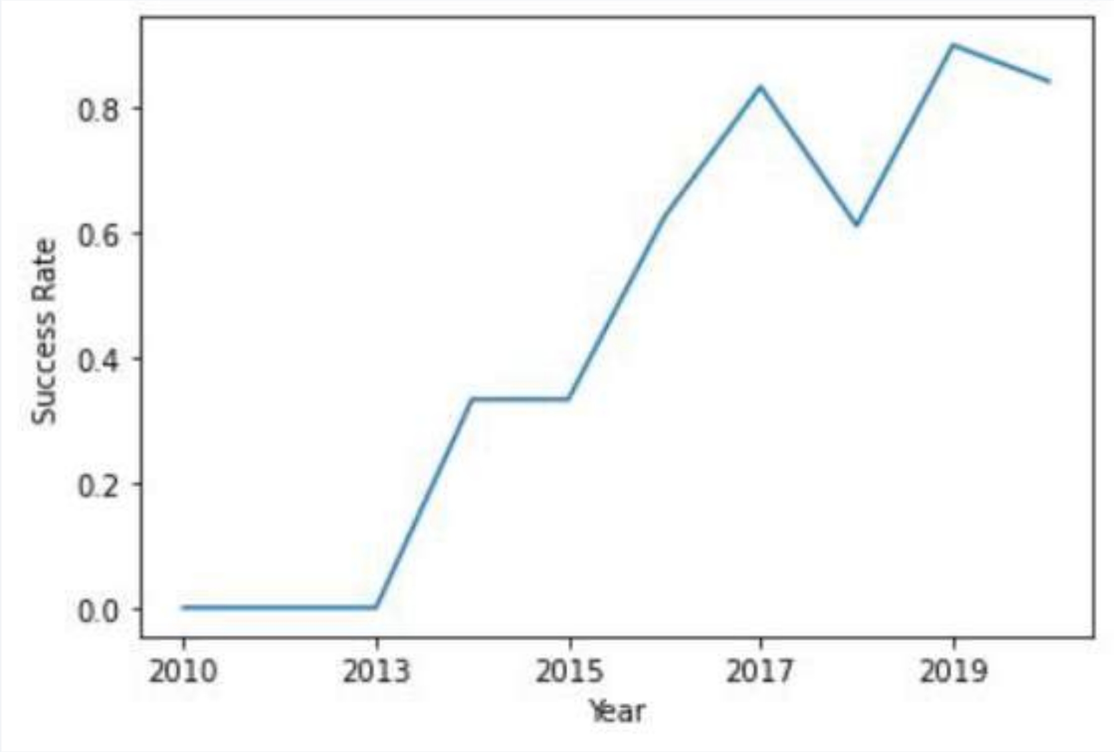
# Payload vs. Orbit Type

- The following orbit types have more success with heavy payloads
  - PO
  - ISS
    LEO
- For GTO, the relationship between payload mass and success rate is unclear
- VLEO launches are associated with heavier payloads, which make intuitive sense.



# Launch Success Yearly Trend

- Between 2010 and 2013, all landingswere unsuccessful
- After 2013, the success rate increased, despite small dips in 2018 and 2020
- After 2016, there was always a greater than 50% chance of success



# All Launch Site Names

```
SELECT DISTINCT LAUNCH_SITE
FROM SPACEXTBL
```

Result

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

# Launch Site Names Begin with 'CCA'

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

## Result

| DATE | time__utc_ | booster_version | launch_site | payload | payload_mass__kg_ | orbit | customer | mission_outcome | landing__outcom |
|---|---|---|---|---|---|---|---|---|---|
| 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 attem |
| 2012-10-08 | 00:35:00 | F9 v1.0 B0006 | CCAFS LC-40 | SpaceX CRS-1 | 500 | LEO (ISS) | NASA (CRS) | Success | No attem |
| 2013-03-01 | 15:10:00 | F9 v1.0 B0007 | CCAFS LC-40 | SpaceX CRS-2 | 677 | LEO (ISS) | NASA (CRS) | Success | No attem |

# Total Payload Mass

```
SELECT SUM(PAYLOAD_MASS__KG_)
       AS total_payload_mass_kg
FROM SPACEXTBL
WHERE CUSTOMER = 'NASA (CRS)'
```

Result

| total_payload_mass_kg |
|---|
| 45596 |

# Average Payload Mass by F9 v1.1

```
SELECT AVG(PAYLOAD_MASS__KG_)
       AS avg_payload_mass_kg
FROM SPACEXTBL
WHERE BOOSTER_VERSION = 'F9 v1.1'
```

Result

| avg_payload_mass_kg |
|---|
| 2928 |

# First Successful Ground Landing Date

```
SELECT MIN(DATE)
  AS first_successful_landing_date
FROM SPACEXTBL
WHERE LANDING__OUTCOME
      = 'Success (ground pad)'
```

Result

| first_successful_landing_date |
|---|
| 2015-12-22 |

# Successful Drone Ship Landing with Payload between 4000 and 6000

```
SELECT BOOSTER_VERSION
FROM SPACEXTBL
WHERE LANDING__OUTCOME = 'Success (drone ship)'
   AND (PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000)
```

Result

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

# Total Number of Successful and Failure Mission Outcomes

```
SELECT MISSION_OUTCOME,
       COUNT(*) AS total_number
FROM SPACEXTBL
GROUP BY MISSION_OUTCOME
```

Result

| mission_outcome | total_number |
|---:|---:|
| Failure (in flight) | 1 |
| Success | 99 |
| Success (payload status unclear) | 1 |

# Boosters Carried Maximum Payload

```
SELECT DISTINCT BOOSTER_VERSION,
       PAYLOAD_MASS__KG_
FROM SPACEXTBL
WHERE PAYLOAD_MASS__KG_ = (
       SELECT MAX(PAYLOAD_MASS__KG_)
       FROM SPACEXTBL)
```

Result

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

# 2015 Launch Records

```
SELECT LANDING__OUTCOME,
       BOOSTER_VERSION,
       LAUNCH_SITE
FROM SPACEXTBL
WHERE LANDING__OUTCOME
          = 'Failure (drone ship)'
       AND YEAR(DATE) = '2015'
```

Result

| landing__outcome | booster_version | launch_site |
|---|---|---|
| Failure (drone ship) | F9 v1.1 B1012 | CCAFS LC-40 |
| Failure (drone ship) | F9 v1.1 B1015 | CCAFS LC-40 |

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

```
SELECT LANDING__OUTCOME,
       COUNT(LANDING__OUTCOME) AS total_number
FROM SPACEXTBL
WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY LANDING__OUTCOME
ORDER BY total_number DESC
```

Result

| landing__outcome | total_number |
|---|---|
| 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

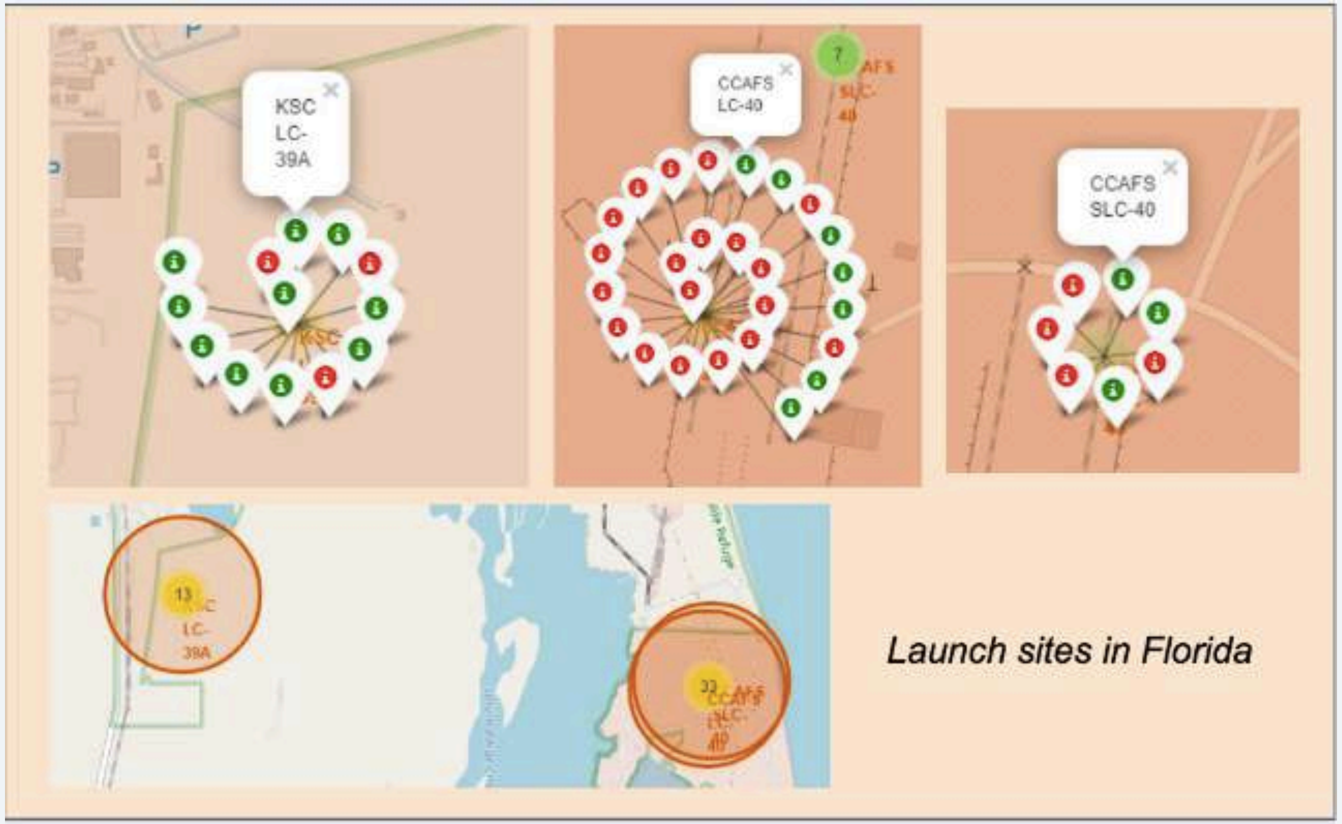
# All Locations of Launch SItes


All SpaceX launch sites are on coasts of the United States of America, specifically Florida and California.

# <Folium Map Screenshot 2>

- Replace <Folium map screenshot 2> title with an appropriate title
- Explore the folium map and make a proper screenshot to show the color-labeled launch outcomes on the map
- Explain the important elements and findings on the screenshot

# Color-labeled Launch Outcomes



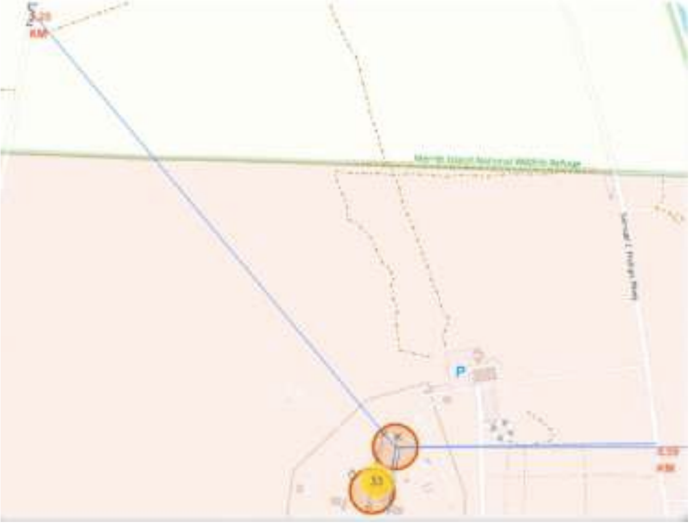
Launch sites in Florida

Launch site in California

Launches have been grouped into clusters, and annotated with green icons for successful launches, and red icons for failed launches.

# Proximity of Launch Sites Of Other Points Of Interest


Using CCAFS SLC-40 launch site as example, we can understand more about the placement of launch sites




Are launch sites in close prximity to railways?

Yes, the costline is only 0.87km due East.

Are launch sites in close proximity to highways?

Yes, the nearest highwya is only 0.59km away.

Are launch sites in close proximity to railways?

Yes, the nearest railway is only 1.29km away

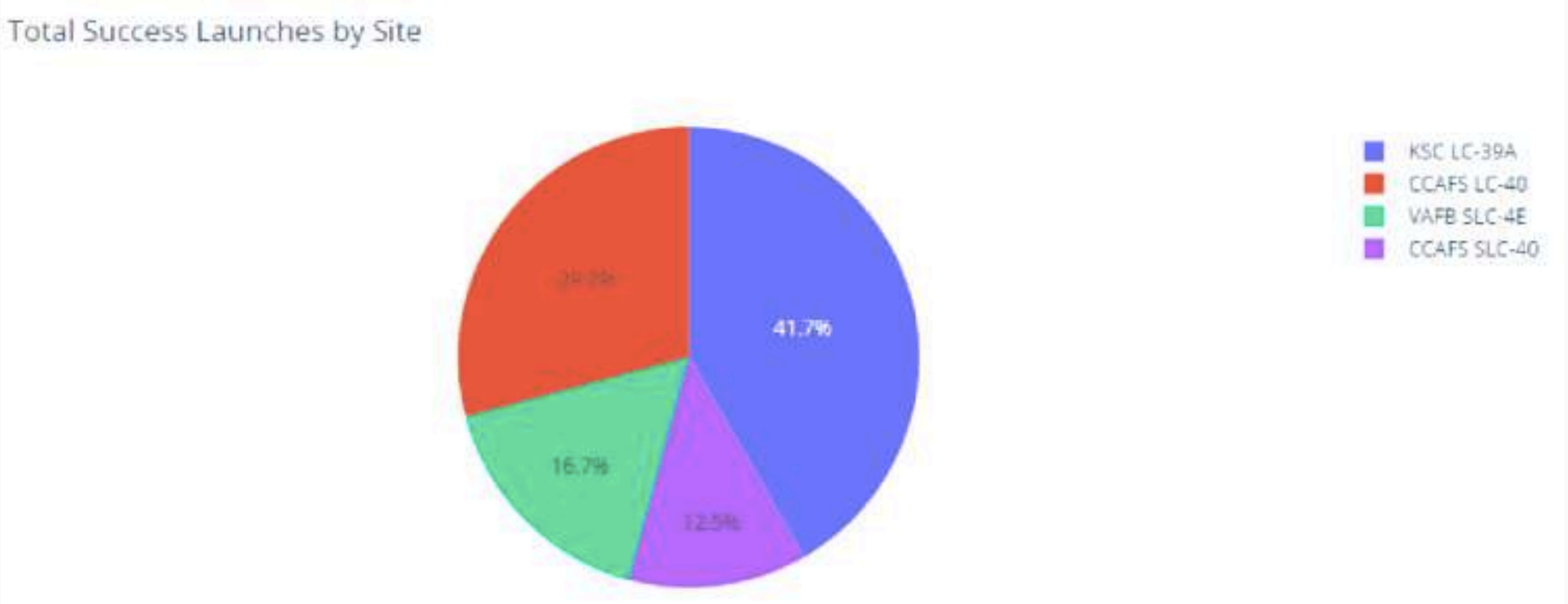
Do launch sites keep certain distance away from cities?

Yes, the nearest city is 51.74km away

Section 4

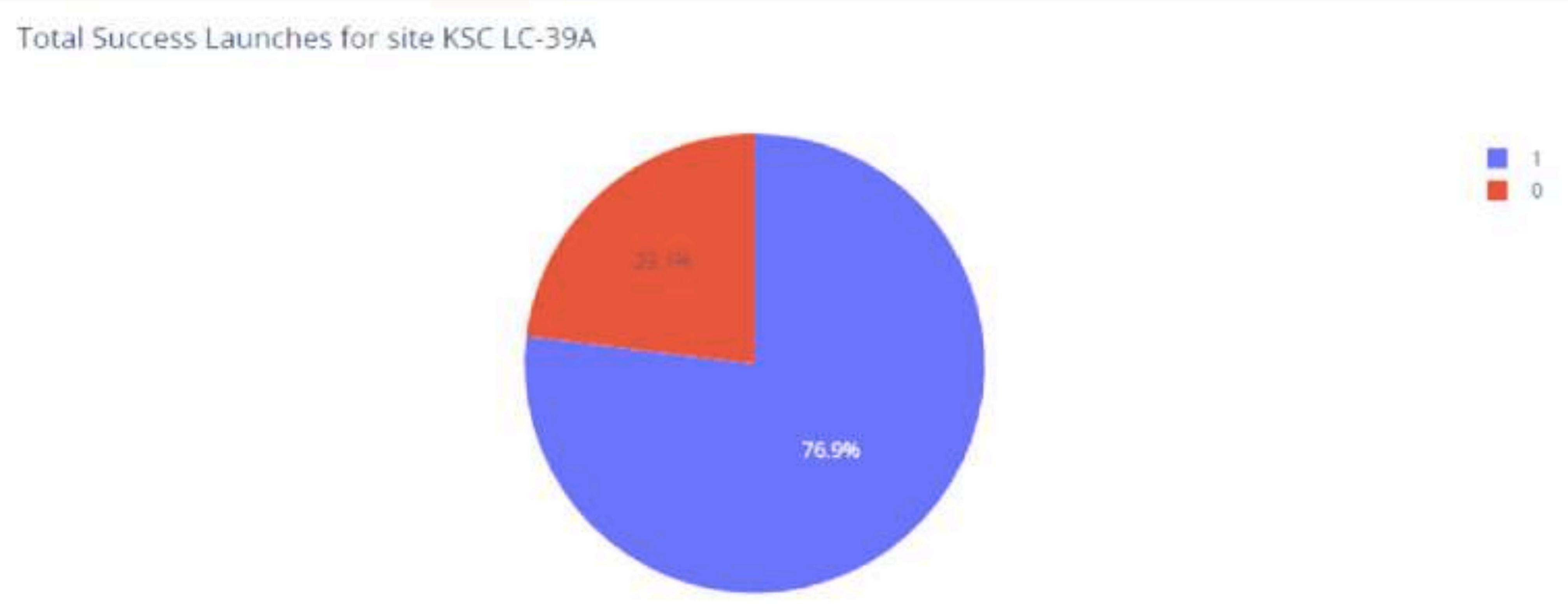
# Build a Dashboard with Plotly Dash

# Success Launch by Sites




The launch site KSCLC-39 A had the most successful launches, with 41.7% of the total successful launches.

# Launch Site with Highest Launch Success Ratio




KSLC– 39A has the highest success rate with 10 landing successes 76.9% and 3 landing failures 23.1%

# Payload vs Launch Outcome Scatter Plot

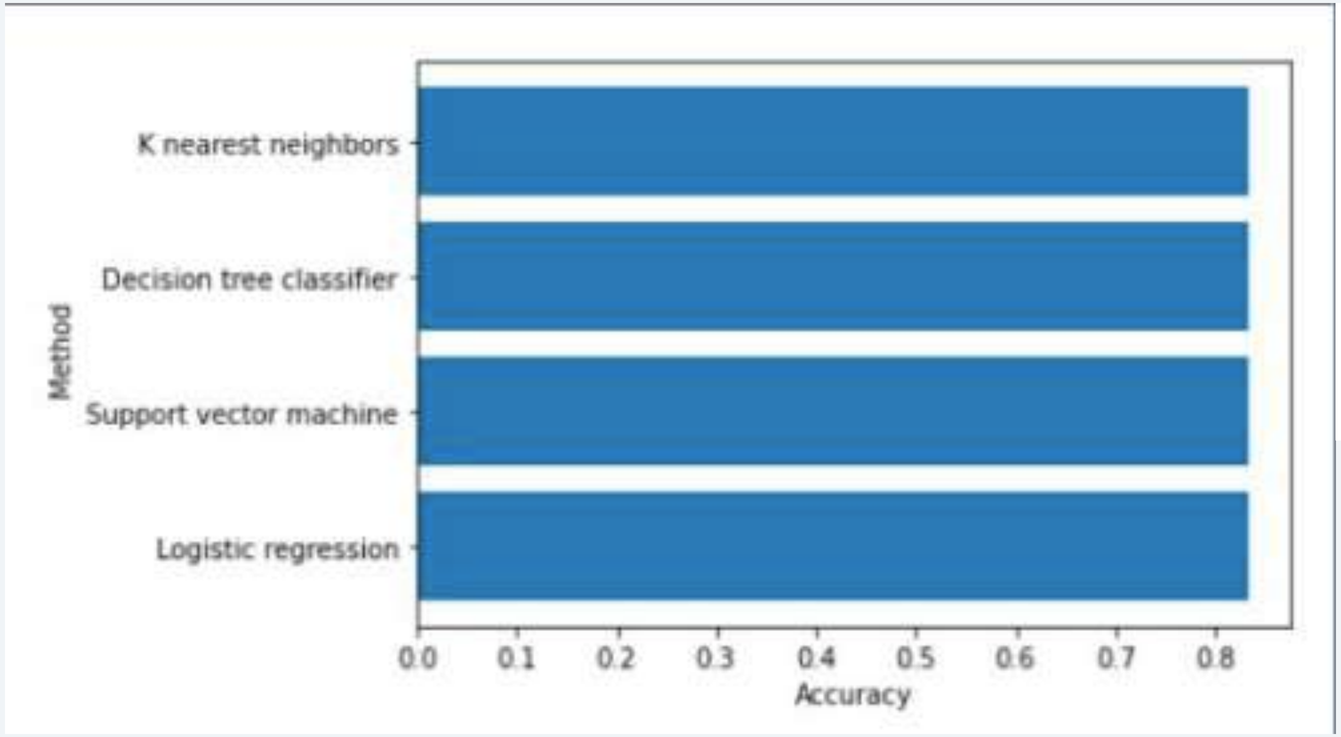


These figures show that the Launch success rate for low weighted payloads is higher than that of heavy weighted payloads.

Section 5

# Predictive Analysis (Classification)

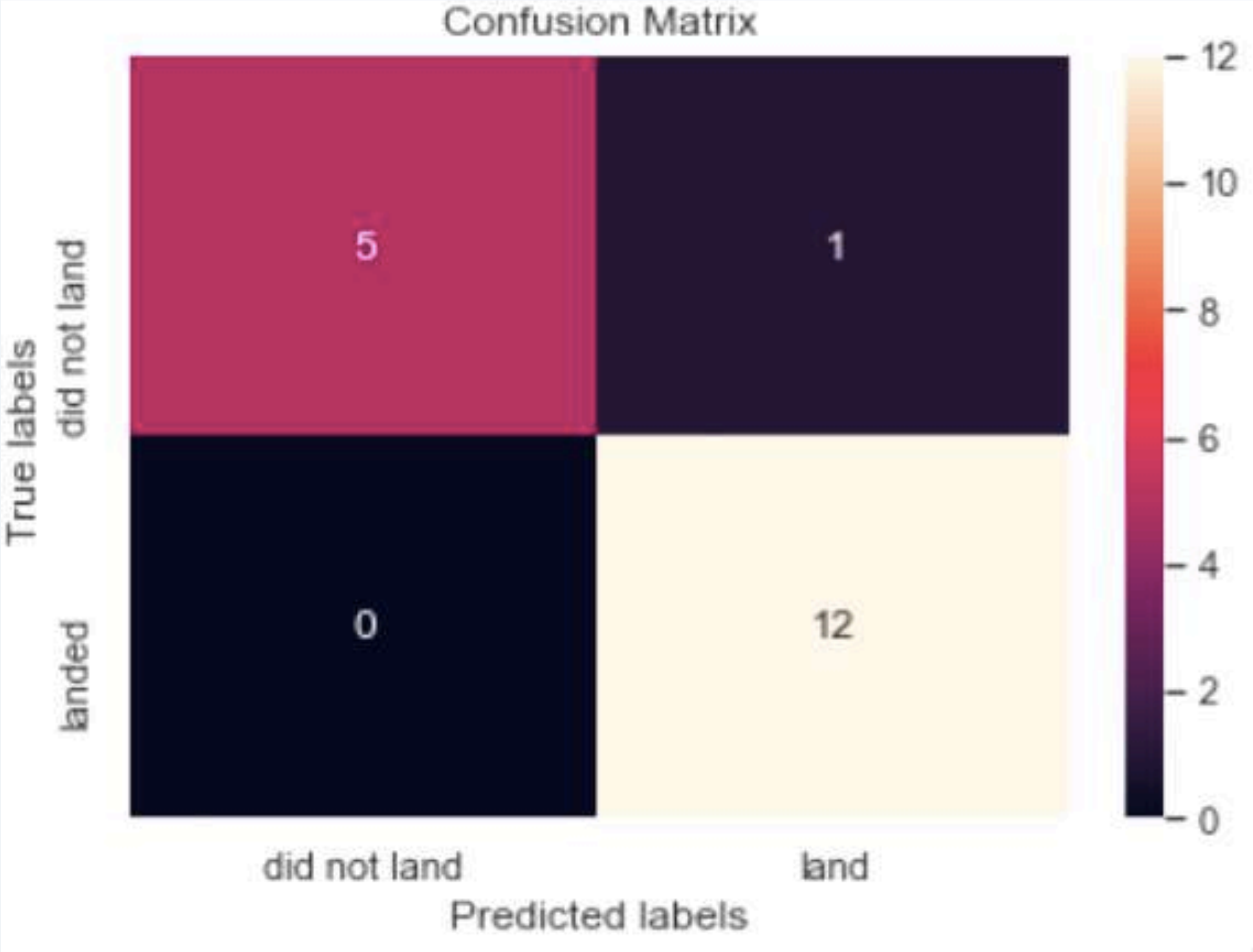
# Classification Accuracy




| | Method | Accuracy |
|---|---|---|
| **0** | Logistic regression | 0.833333 |
| **1** | Support vector machine | 0.833333 |
| **2** | Decision tree classifier | 0.833333 |
| **3** | K nearest neighbors | 0.833333 |

- In the test set, the accuracy of all models was virtually the same at 83.33%
- The test size was small at 18
- That means more data is needed to determine the optimal model

# Confusion Matrix




- The best performing classification model is the DecisionTree
- This explain the confusion matrix shos only 1 out of 18 results classified incorrectly
- The other 17 results are correctly classified

# Conclusions

- As the number of flights increases, the rate of success at a launch site increases, with most early flights beging unsuccessful.
- Orbital types SSO, HEO, GEO, and ES-L1 have the highest success rate 100%
- KSLC-39A has the highest number of launch successes and the highest success rate among all sites
- The success for massive payloads is lower than that for low payloads.
- In this dataset, all models have the same accuracy, ubt it seems more data is needed to determine the optimal model due to the small data size.

# Appendix

**Coursera Applied Data Science Capstone Course**

# Thank you!