IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Esteban J. Chinni
February 2023

Outline

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

Executive Summary

- Summary of methodologies

Data collection, data wrangling, exploratory data analysis (SQL), data visualization (Folium, Plotly) , model development, model evaluation, and reporting

- Summary of all results

Predict if the first stage of the SpaceX Falcon 9 rocket will land successfully. All models over predicted successful landings. Additional data is needed to improve model determination and accuracy

Different machine learning models produced: Logistic Regression, Support Vector Machine, Decision Tree Classifier, and K Nearest Neighbors. All produced similar results with accuracy ~ 83.33%.

Introduction

- Project background and context

Predict 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.

- Problems you want to find answers

Which are the variables and their relationship that influence on a successful rocket landing?

Section 1

Methodology

Methodology

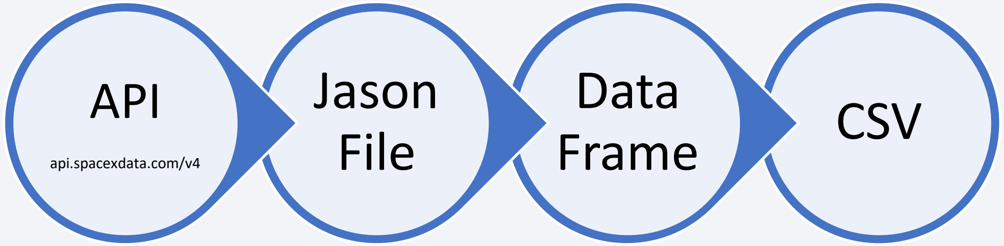
Executive Summary

- Data collection methodology:
  - Retrieved from SpaceX API and web scraped to collect Falcon 9 historical launch records from a Wikipedia page.
- Perform data wrangling
  - Replace missing values with mean and other appropriate methods. Encoding data fields for Machine Learning and dropping irrelevant columns
    - Classifying true landings as successful and unsuccessful otherwise
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Different models were trained. Model with highest accuracy with tuned hyper-parameters was selected.

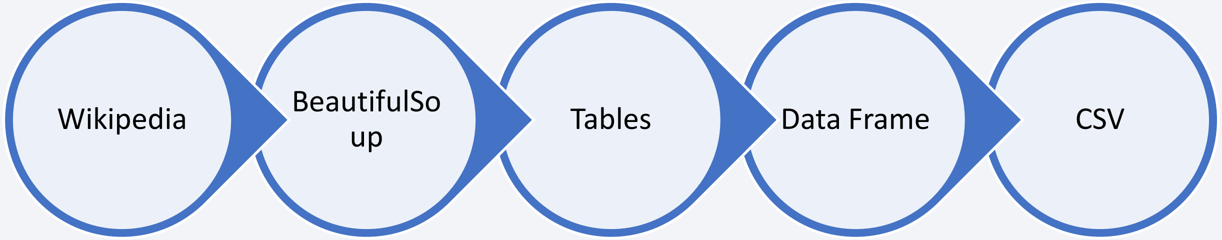
Data Collection

- Describe how data sets were collected

Option 1: Space X API



Option 2: Webscraping



Deal with Nulls values!*

*replace the null values with the mean

Data Collection – SpaceX API

- Present your data collection with SpaceX REST calls using key phrases and flowcharts

1. Getting response from SpaceX-API
2. Converting response to json-file and transferring into a Data-Frame
3. Clean up the data using prepared functions
4. Creating a final dataset with the columns of interest
5. Filtering dataset for Falcon 9 Launches
6. Dealing with Missing Values
7. Export to CSV

GitHub https://github.com/echinni/IBM-Data-Science-Final-Project/blob/299616c7750133b55a37bff9bc523137a74efd3f/Week%201.1%20Data%20Collection%20API%20Lab.ipynb

1
```
spacex_url="https://api.spacexdata.com/v4/launches/past"
```
```
response = requests.get(spacex_url)
```

2
```
# Use json_normalize meethod to convert the json result into a dataframe
data = pd.json_normalize(response.json())
```

3
```
# Call getBoosterVersion
getBoosterVersion(data)

# Call getLaunchSite
getLaunchSite(data)

# Call getPayloadData
getPayloadData(data)

# Call getCoreData
getCoreData(data)
```

4
```
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
```
# Hint data['BoosterVersion']!='Falcon 1'
data_falcon9 = df1[df1['BoosterVersion']!='Falcon 1']
```

6
```
# Replace the np.nan values with its mean value
data_falcon9['PayloadMass'] = data_falcon9['PayloadMass'].replace(np.nan, av_PayloadMass)
```

7
```
data_falcon9.to_csv('dataset_part_1.csv', index=False)
```

Data Collection - Scraping

- Present your web scraping process using key phrases and flowcharts

1. Request the Falcon9 Launch Wiki page from its URL
2. Create Beautiful-soup object from HTML response
3. Extract all column/variable names from the HTML table header
4. Create a data frame by parsing the launch HTML tables
5. Exporting data to CSV-file

GitHub https://github.com/echinni/IBM-Data-Science-Final-Project/blob/27f6af94f87a0723c6205f574710349e367e0a51/Week%201.2%20Data%20Collection%20with%20Web%20Scraping.ipynb

1.

```
4]: static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_
# assign the response to a object
html_data=requests.get(static_url).text
```

2.

```
[21]: # Use BeautifulSoup() to create a BeautifulSoup
soup = BeautifulSoup(html_data,'html.parser')
```

3.

```
# Assign the result to a list calle
html_tables=soup.find_all('table')
```

4.

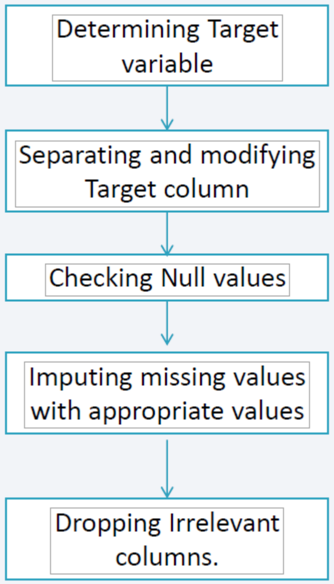
```
[49]: extracted_row = 0
#Extract each table
for table_number,table in enumerate(soup.find_all('table',"wikitable plainrowheaders collapsible")):
   # get table row
    for rows in table.find_all("tr"):
        #check to see if first table heading is as number corresponding to launch a number
        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)
            #print(flight_number)
```

5.

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

Data Wrangling

- Perform some Exploratory Data Analysis (EDA) to find some patterns in the data and determine what would be the label for training supervised models.
- **Identify and calculate the percentage of the missing values in each attribute**
- **Calculate the number of launches on each site**
- **Calculate the number and occurrence of each orbit**
- **Calculate the number and occurence of mission outcome per orbit type**
- **Create a landing outcome label from Outcome column:** 1 means the booster successfully landed 0 means it was unsuccessful.



https://github.com/echinni/IBM-Data-Science-Final-Project/blob/c2bec7687974aa338e71b890a31787c1f98ef875/Week%201.3%20Data%20Wrangling.ipynb

EDA with Data Visualization

- Summarize what charts were plotted and why you used those charts
    - Flight Number vs. Payload Mass
    - Flight Number vs. Launch Site
    - Payload Mass vs. Launch Site
    - Orbit vs. Success Rate
    - Flight Number vs. Orbit
    - Payload vs Orbit
    - Success Yearly Trend

- **Scatterplots** are useful to show relationships between variables
- **Bar charts** are suitable for comparing the ratio of a variable in discrete classes with one another, if necessary, grouping them as well
- **Line plots** show the progression of variables over time

GitHub [IBM-Data-Science-Final-Project/Week 2.2 EDA with Visualization Lab.ipynb at main · echinni/IBM-Data-Science-Final-Project (github.com)]

EDA with SQL

- Using bullet point format, summarize the SQL queries you performed
- Loaded data set into IBM DB2 Database.
- Queried using SQL Python integration.
- Queries were made to get a better understanding of the dataset.
- Queried information about launch site names, mission outcomes, various pay load sizes of customers and booster versions, and landing outcomes

GitHub [IBM-Data-Science-Final-Project/Week 2.1 Complete the EDA with SQL lab2.ipynb at main · echinni/IBM-Data-Science-Final-Project (github.com)]

Build an Interactive Map with Folium

- Summarize what map objects such as markers, circles, lines, etc. you created and added to a folium map. Explain why you added those objects
- Folium maps mark Launch Sites. Given a text label, and its latitude and longitude, successful (green) and unsuccessful (red) landings, and a proximity example to key locations: Railway, Highway, Coast, and City.
- This allows us to understand why launch sites may be located where they are. Also visualizes successful landings relative to location.

GitHub [IBM-Data-Science-Final-Project/Week 3.1 Data Visualization with Folium.ipynb at main · echinni/IBM-Data-Science-Final-Project (github.com)]

Build a Dashboard with Plotly Dash

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.*

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.

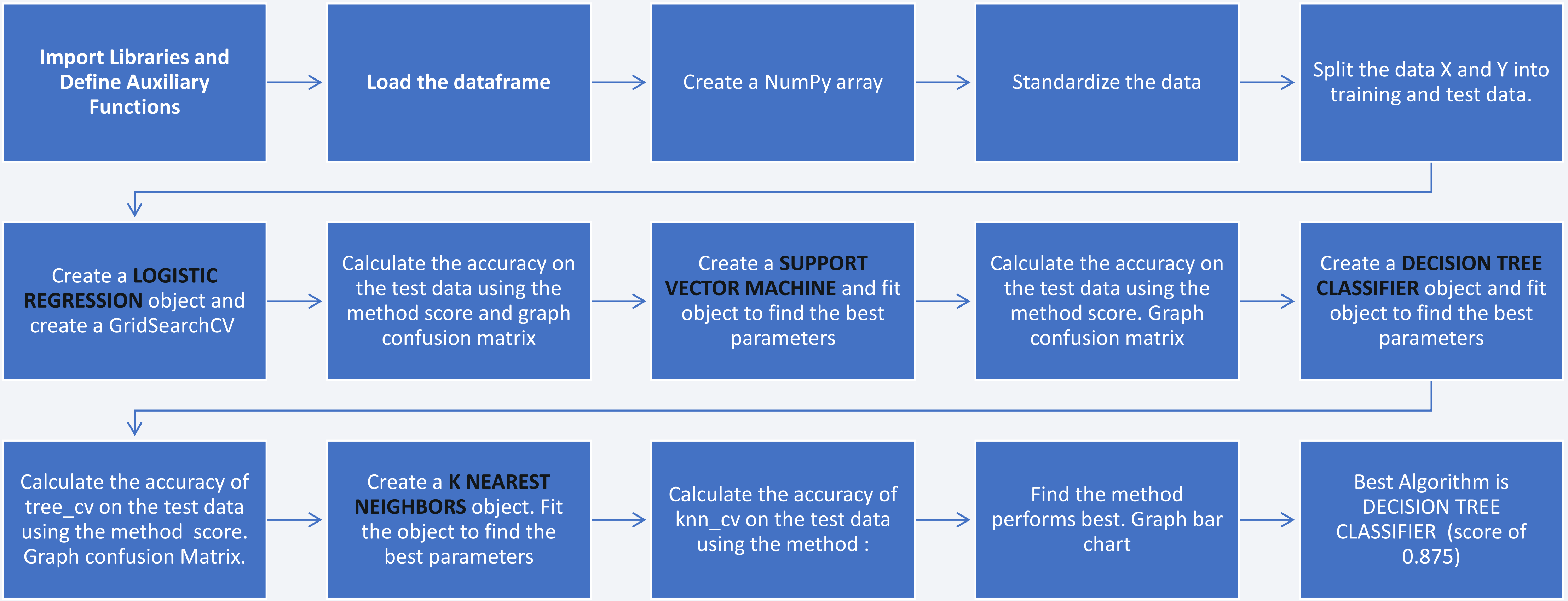
Observation and reading are straightforward.

GitHub https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/labs/module_3/lab_theia_plotly_dash.md.html

Predictive Analysis (Classification)

- Summarize how you built, evaluated, improved, and found the best performing classification model
- You need present your model development process using key phrases and flowchart
- Add the GitHub URL of your completed predictive analysis lab, as an external reference and peer-review purpose

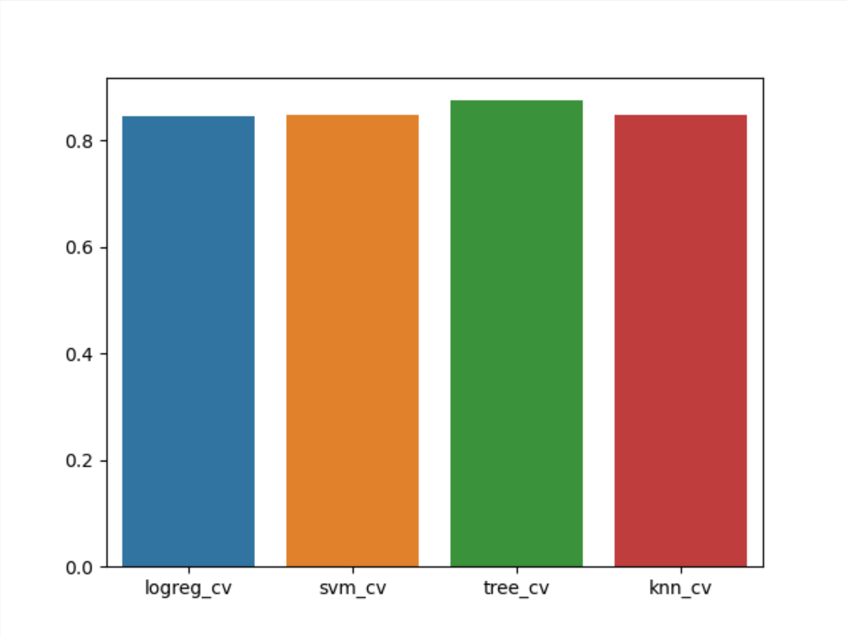
Predictive Analysis (Classification)



GitHub [IBM-Data-Science-Final-Project/Week 4 Machine Learning Prediction.ipynb at main · echinni/IBM-Data-Science-Final-Project (github.com)]

Results

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

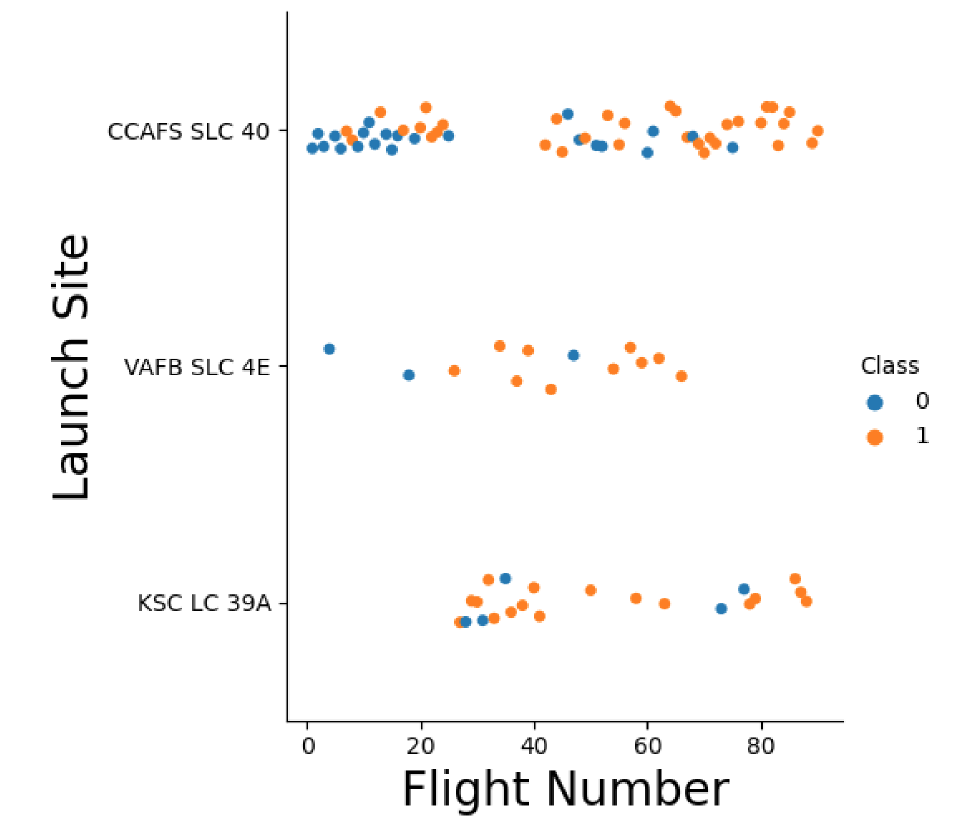


Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

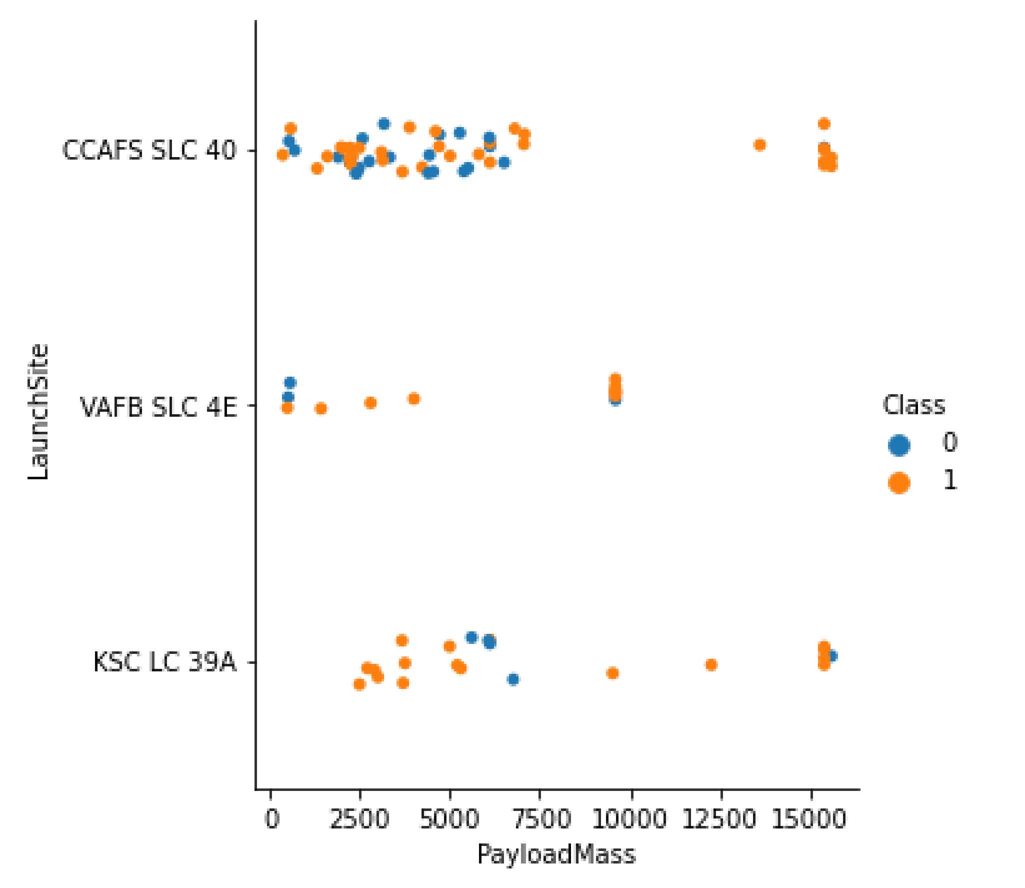
- Launch success rate increased over the time (from 0% to 100%)
- Most part of the launches were performed from CCAFS
- VAFB SLC 4E and KSC LC 39A have the highest success rates for starting positions.



Payload vs. Launch Site

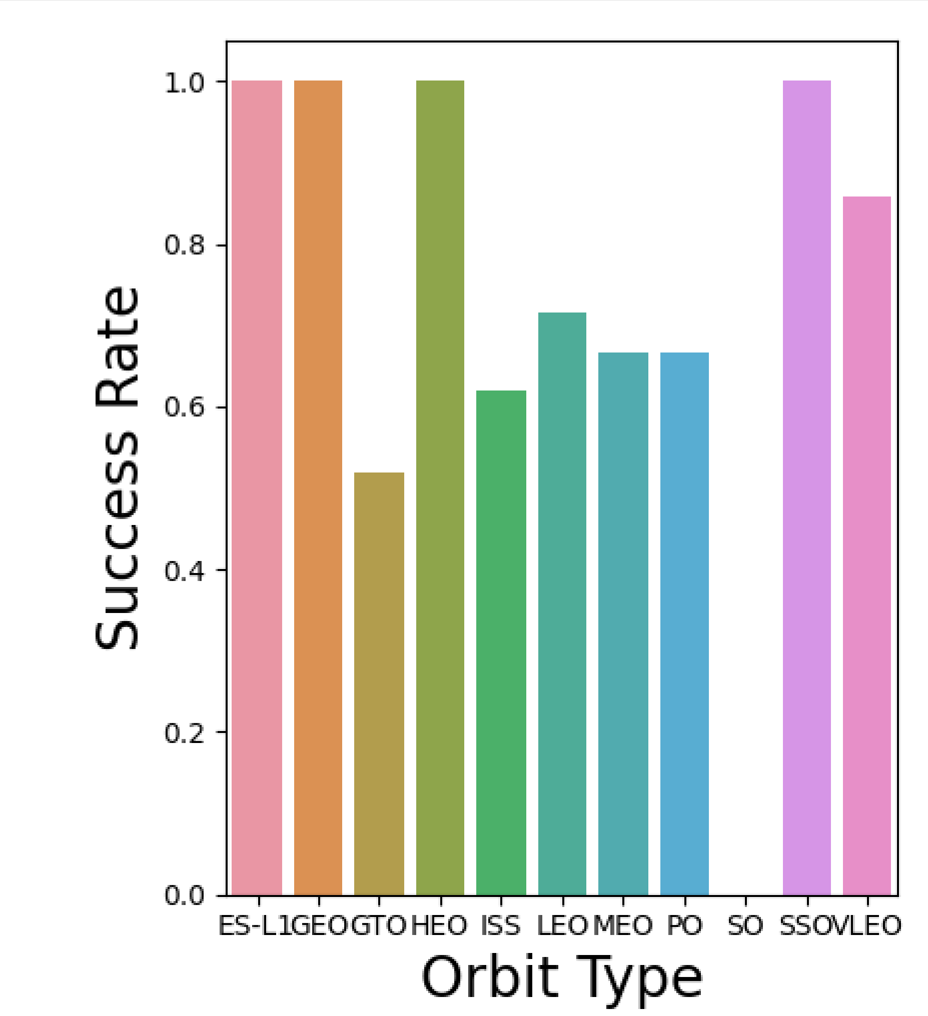
- The larger the payload, the better the success rate, except for KSC.

- The VAFB SLC 4E launch site appears inadequate for launches with heavy payloads (>10.000)



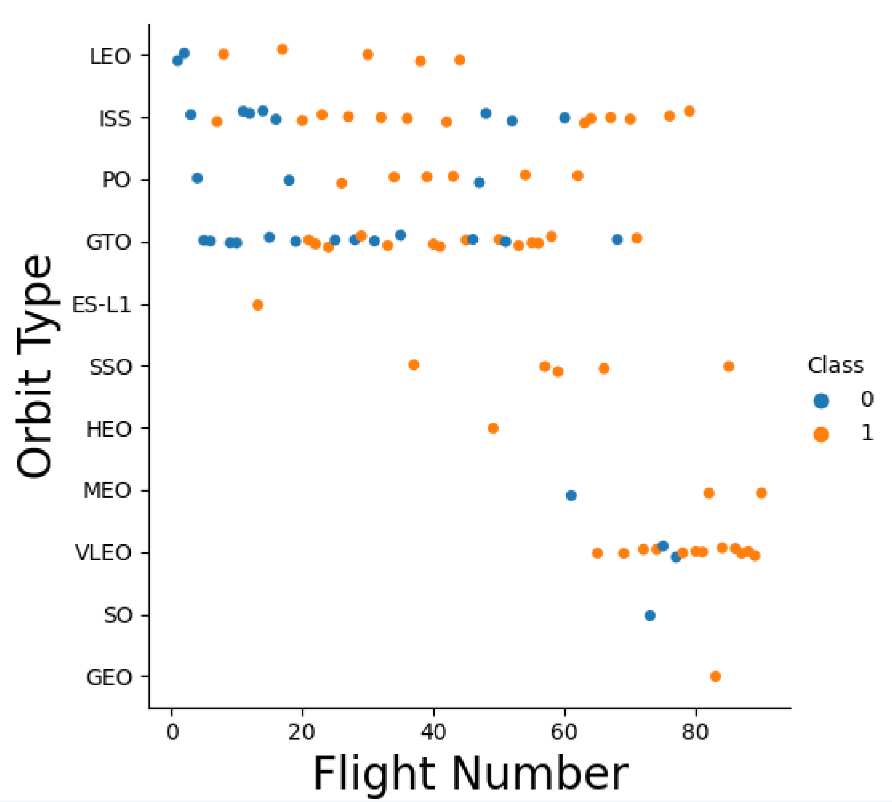
Success Rate vs. Orbit Type

- The orbits ES-L1, GEO, HEO and SSO have highest 100% sucess rate.
- No successful launch found fororbit type SO.
- The rest of orbit types present a success rate of around 60%.



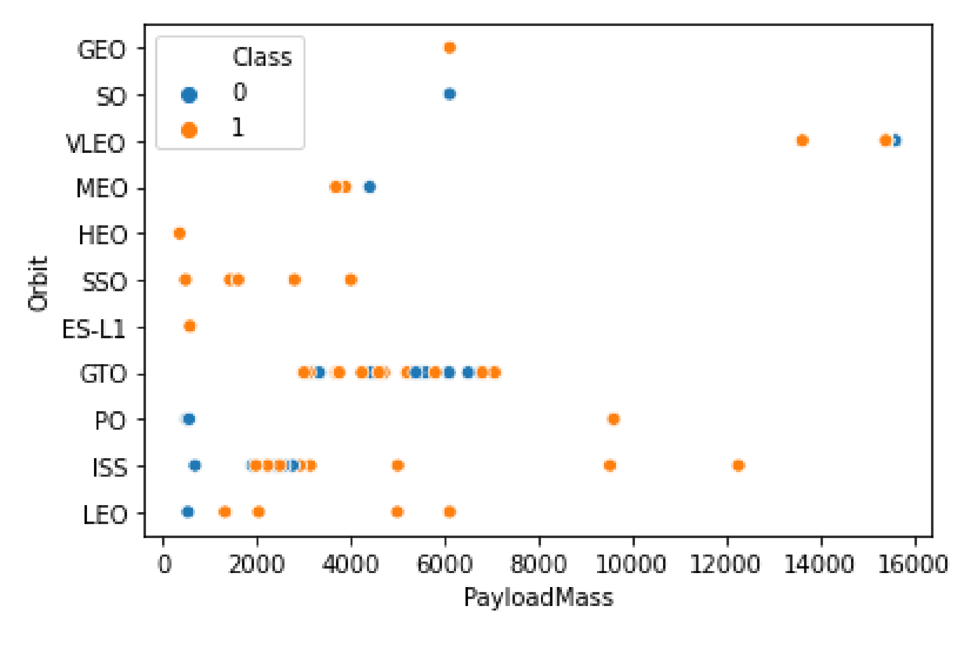
Flight Number vs. Orbit Type

- VLEO orbit was selected for the most recent launches (from flight 65 on). Present a high successful rate of 86%.
- Higher number of launches: ISS (21), GTO (27) and VLEO (14).
- No relationship between orbit type and flight number for GTO and ISS orbits
- SpaceX appears to perform better in lower orbits or Sun-synchronous orbits



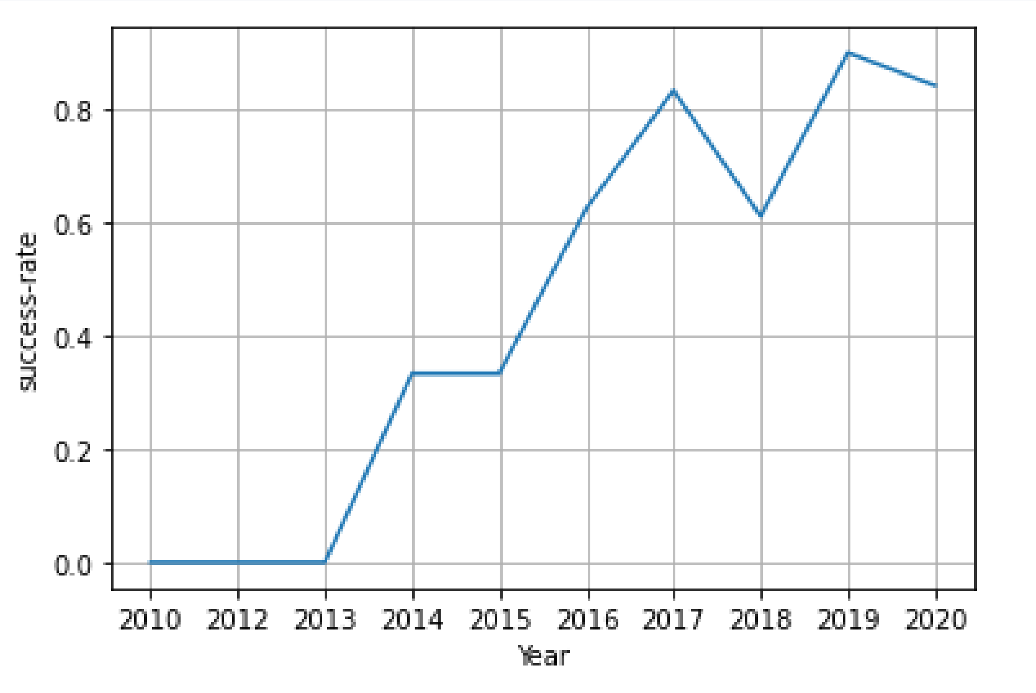
Payload vs. Orbit Type

- Low payloads and low orbits indicate a low success rate, but this could also have been the result of the early launch attempts.
- Higher launches: orbits GTO and ISS.
- ISS have the most diverse array of payloads



Launch Success Yearly Trend

- Success generally increases over time since 2013 with a slight dip in 2018
- Success in recent years at around 80%



All Launch Site Names

In [4]:
```
%%sql
SELECT UNIQUE LAUNCH_SITE
FROM SPACEXDATASET;
```

```
 * ibm_db_sa://ftb12020:***@0c77d6f
Done.
```

Out[4]:

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

Query unique launch site names from database.

CCAFS SLC-40 and CCAFSSLC-40 likely all represent the same launch site with data entry errors.

CCAFS LC-40 was the previous name.

Likely only 3 unique launch_site values:

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

Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with `CCA`

In [5]:
```
%%sql
SELECT *
FROM SPACEXDATASET
WHERE LAUNCH_SITE LIKE 'CCA%'
LIMIT 5;
```

```
 * ibm_db_sa://ftb12020:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/bludb
Done.
```

Out[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

First five entries in database with Launch Site name beginning with CCA.

Total Payload Mass

- Calculate the total payload carried by boosters from NASA

Display the total payload mass carried by boosters launched by NASA (CRS)

```
]: %sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE CUSTOMER LIKE 'NASA (CRS)'
```

```
 * sqlite:///my_data1.db
Done.
```

SUM(PAYLOAD_MASS__KG_)
45596

- The result returns the total payload of all boosters launched by NASA.

Average Payload Mass by F9 v1.1

```
%%sql
SELECT AVG(PAYLOAD_MASS__KG_) AS AVG_PAYLOAD_MASS_KG
FROM SPACEXDATASET
WHERE booster_version = 'F9 v1.1'
```

```
 * ibm_db_sa://ftb12020:***@0c77d6f2-5da9-48a9-81f8-86
Done.
```

avg_payload_mass_kg
2928

- This query calculates the average payload mass or launches which used booster version F9 v1.1
- Average payload mass of F9 1.1 is on the low end of our payload mass range

First Successful Ground Landing Date

```
%%sql
SELECT MIN(DATE) AS FIRST_SUCCESS
FROM SPACEXDATASET
WHERE landing__outcome = 'Success (ground pad)';
```

```
 * ibm_db_sa://ftb12020:***@0c77d6f2-5da9-48a9-81
Done.
```

first_success
2015-12-22

- This query returns the first successful ground pad landing date.
- First ground pad landing wasn't
- until the end of 2015.
- Successful landings in general
- appear starting 2014.

Successful Drone Ship Landing with Payload between 4000 and 6000

List the names of the boosters which have success in drone ship and have 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;
```

```
 * sqlite:///my_data1.db
Done.
```

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

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

Total Number of Successful and Failure Mission Outcomes

```
%%sql SELECT
    sum(CASE WHEN "Mission_Outcome" LIKE 'Success%' THEN 1 ELSE 0 END) AS 'Success',
    sum(CASE WHEN "Mission_Outcome" LIKE 'Failure%' THEN 1 ELSE 0 END) AS 'Failure'
    from SPACEXTBL
```

```
 * sqlite:///my_data1.db
Done.
```

Success	Failure
100	1

Results display the total number of successful and failed mission outcomes. It states that the failed are minimal.

Boosters Carried Maximum Payload

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

List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

In [32]:

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

```
 * sqlite:///my_data1.db
Done.
```

Out[32]:

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

Query returns the names of the booster versions that carried the maximum payload, a total of 12 booster versions

2015 Launch Records

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

```
%%sql
SELECT MONTHNAME(DATE) AS MONTH, landing__outcome, booster_version, PAYLOAD_MASS__KG_, launch_site
FROM SPACEXDATASET
WHERE landing__outcome = 'Failure (drone ship)' AND YEAR(DATE) = 2015;
```

```
 * ibm_db_sa://ftb12020:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.app
Done.
```

MONTH	landing__outcome	booster_version	payload_mass__kg_	launch_site
January	Failure (drone ship)	F9 v1.1 B1012	2395	CCAFS LC-40
April	Failure (drone ship)	F9 v1.1 B1015	1898	CCAFS LC-40

This query returns the Month, Landing Outcome, Booster Version, Payload Mass (kg), and Launch site of 2015 launches where stage 1 failed to land on a drone ship.

There were two such occurrences.

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 no_outcome
FROM SPACEXDATASET
WHERE landing__outcome LIKE 'Succes%' AND DATE BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY landing__outcome
ORDER BY no_outcome DESC;
```

```
 * ibm_db_sa://ftb12020:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg
Done.
```

landing__outcome	no_outcome
Success (drone ship)	5
Success (ground pad)	3

This query returns a list of successful landings and between 2010-06-04 and 2017-03-20 inclusively.

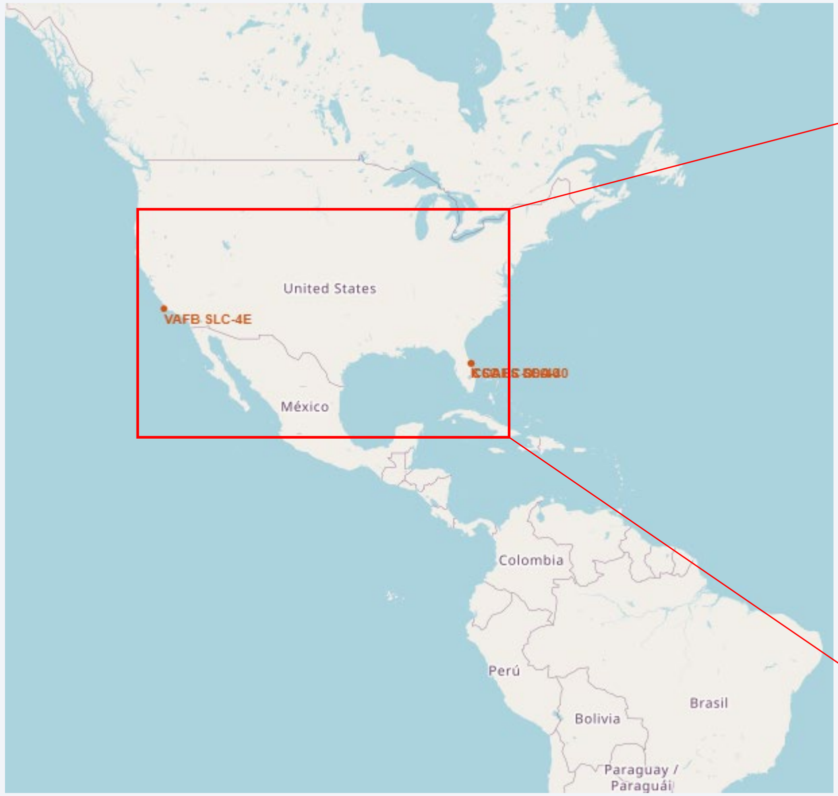
There are two types of successful landing outcomes: drone ship and ground pad landings.

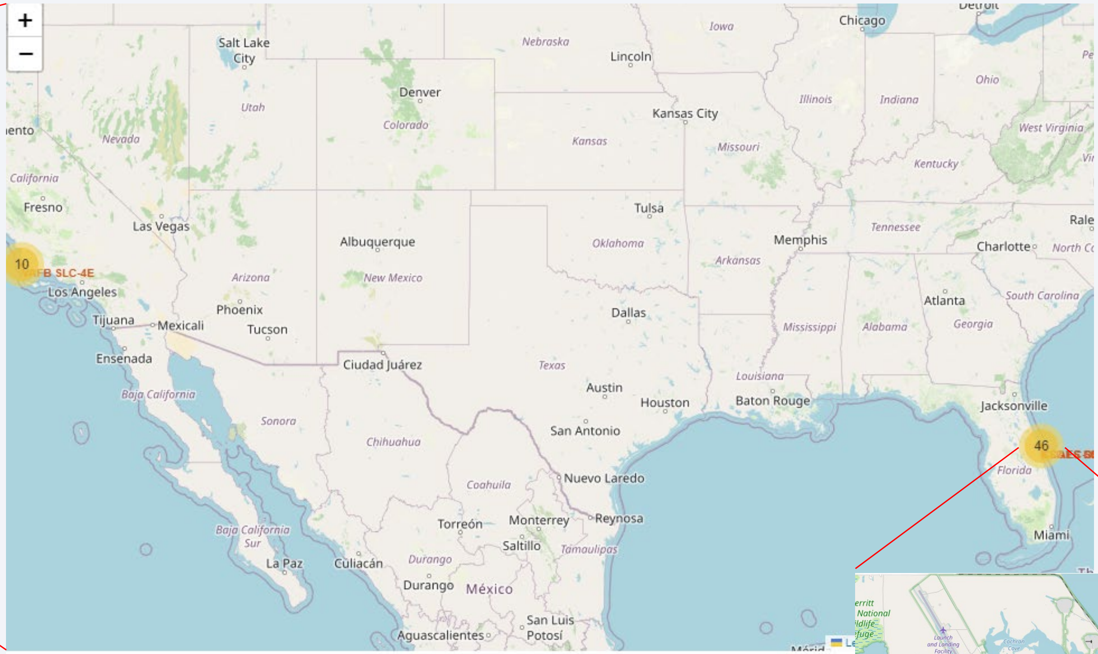
There were 8 successful landings in total during this time period

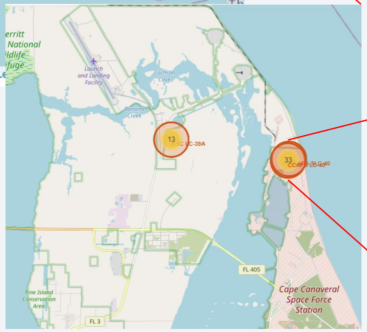
Section 3

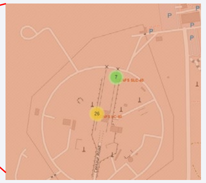
Launch Sites Proximities Analysis

Space X Launch Sites







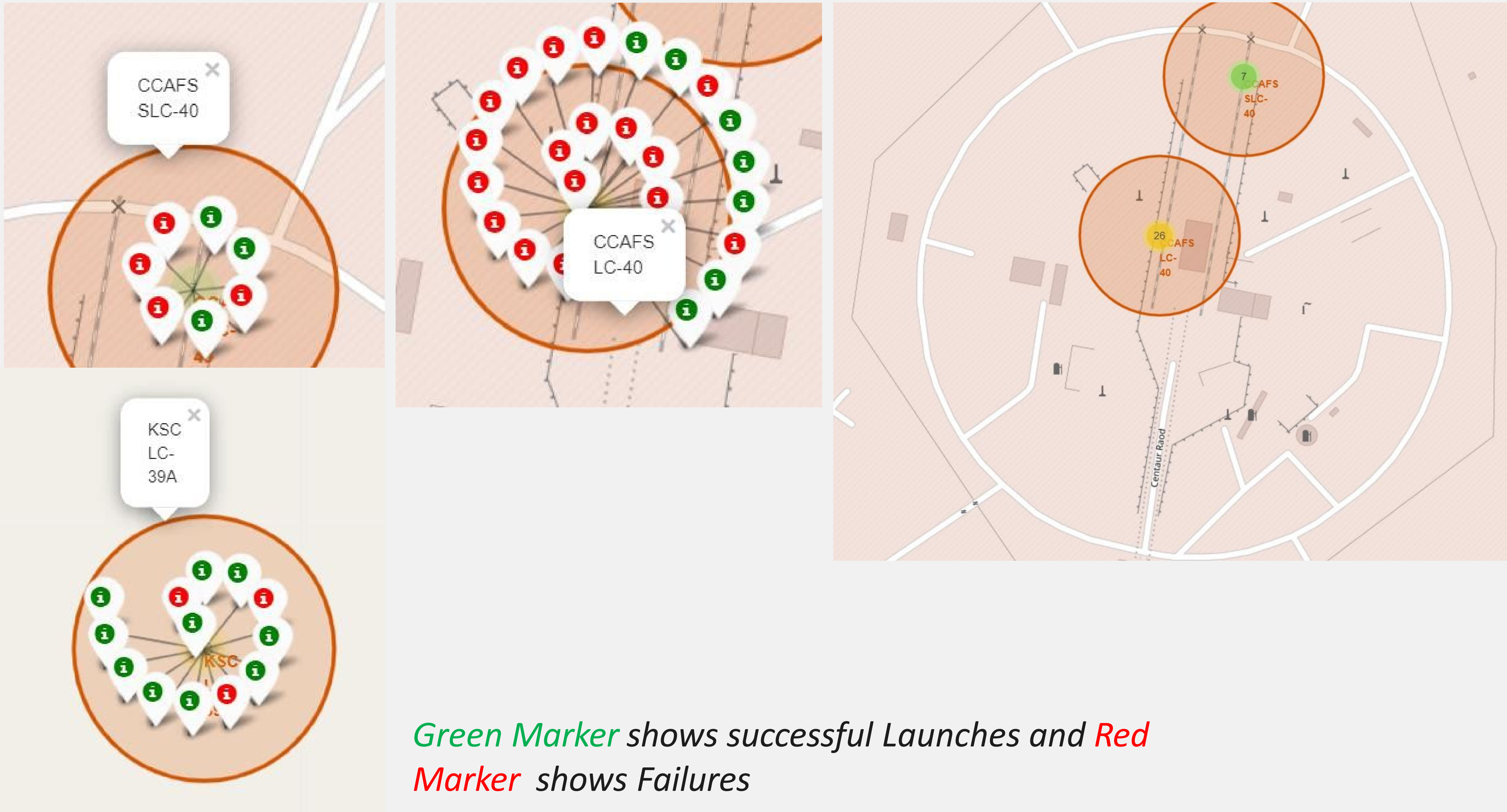


- Three launch sites are close to each other in Orlando, Florida.
- One launch site is in California.
- All launch sites are close to the coast so that first stage can be thrown to the sea after the take-off.

Success Launches

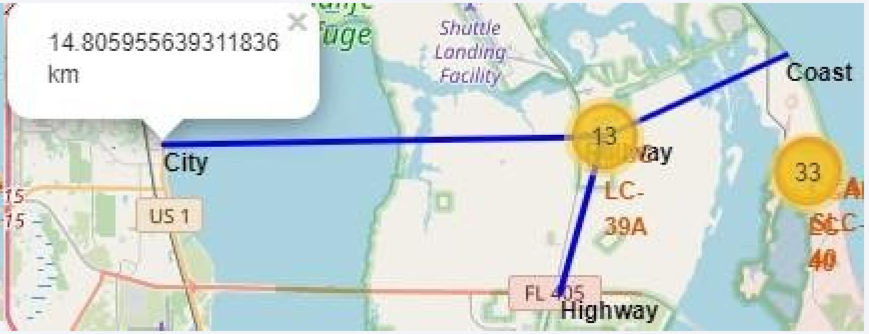
CALIFORNIA

FLORIDA



Green Marker shows successful Launches and Red Marker shows Failures

Strategic Location



Close to railways for large part and supply transportation

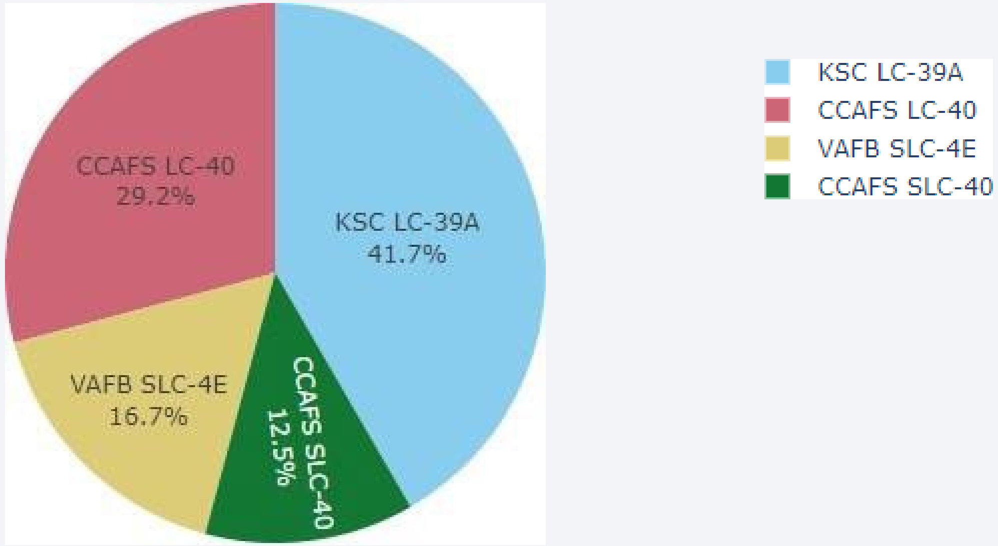
Close to highways for human and supply transport

Close to coasts and relatively far from cities so that launch failures can land in the sea and avoiding rockets falling populated areas.

Section 4

Build a Dashboard with Plotly Dash

Launch success count for all sites



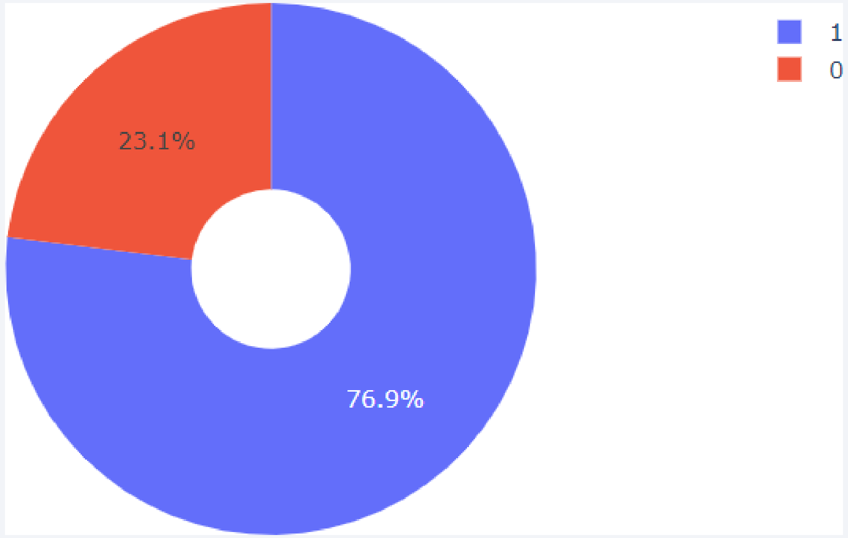
Most successful launch attempts: "KSC LC-39A“

2/3 of the total successful missions: "KSC LC-39A" and "CCAFS LC-40“

VAFB has the smallest share of successful landings. This may be due to smaller sample and increase in difficulty of launching in the west coast..

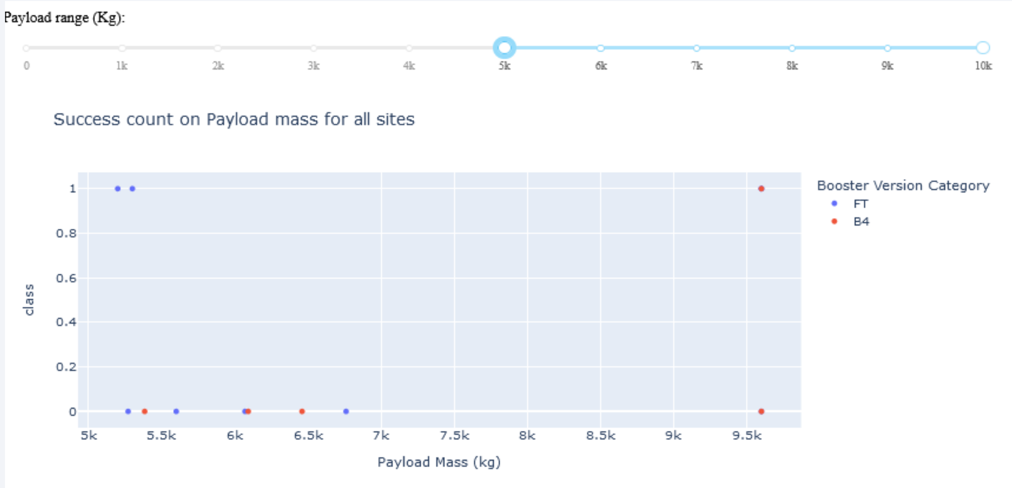
Launch site with highest launch success ratio

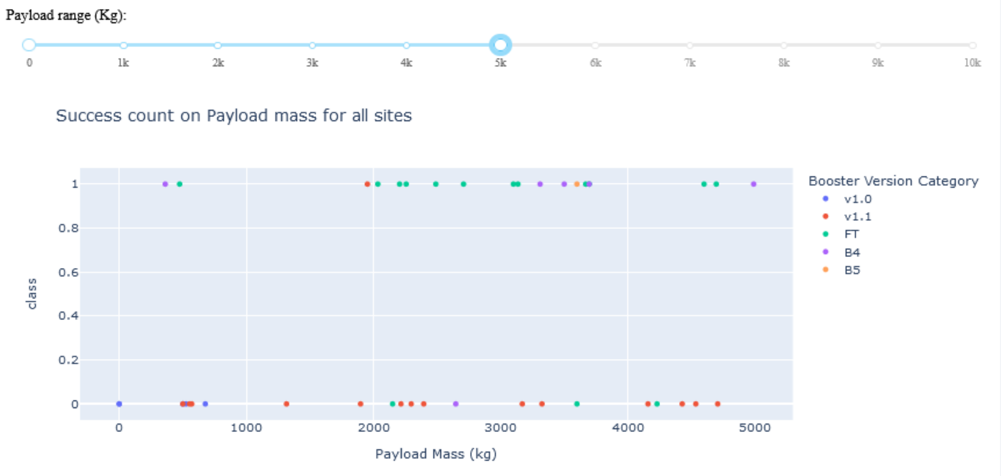
KSC LC-39A Success Rate (blue=success)



KSC LC-39A has the highest success rate with 10 successful landings and 3 failed landings.

Payload vs. Launch Outcome





Plotly dashboard has a Payload range selector.

Class indicates 1 for successful landing and 0 for failure.

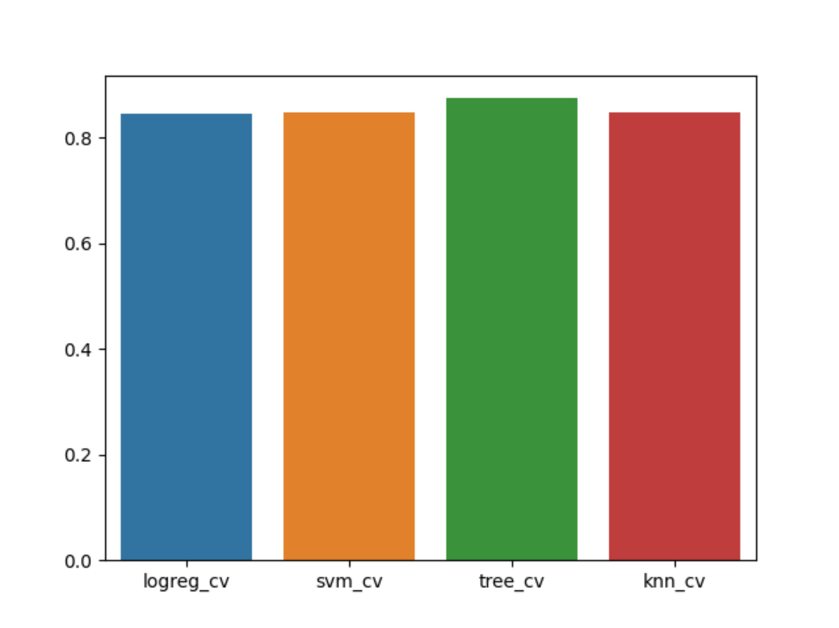
Scatter plot also accounts for booster version category in color and number of launches in point size.

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

Section 5

Predictive Analysis (Classification)

Classification Accuracy



After selecting the best hyperparameters for the decision tree classifier using the validation data, we achieved 83.33% accuracy on the test data.

Decision Tress has the highest classification accuracy

Confusion Matrix

The major problem is false positives.

Conclusions

- It is observed that Decision tree has the highest classification accuracy with accuracy score of 88.8%.
- 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
- Launch sitres are located close to coast, eailways and highways mainting safety distance from cities
- Allon Mask of SpaceY can use this model to predict with relatively high accuracy whether a launch will have a successful Stage 1 landing before launch to determine whether the launch should be made or not

Appendix

Github url for Capstone Project

[echinni/IBM-Data-Science-Final-Project: Python Basics for Data Science Project (github.com)]

Thank you for taking time and evaluate the project!

Thank you!