Winning Space Race with Data Science

Carol Xiao
May 20, 2023

Outline

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

Executive Summary

Summary of methodologies:

- Data collection using SpaceX REST API and Web scape Wikipedia pages
- Data wrangling
- EDA with data visualization
- EDA with SQL
- Building an interactive map with Folium
- Building a Dashboard with Plotly Dash
- Predictive analysis (Classification)

Summary of all results

- EDA with data visualization results
- EDA with SQL results
- Interactive analytics
- Predictive analysis

Introduction

- Project background and context
  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.

- Problems you want to find answers
  The project task is to predicting if the first stage of the SpaceX Falcon 9 rocket will land successfully.

Section 1

Methodology

Methodology

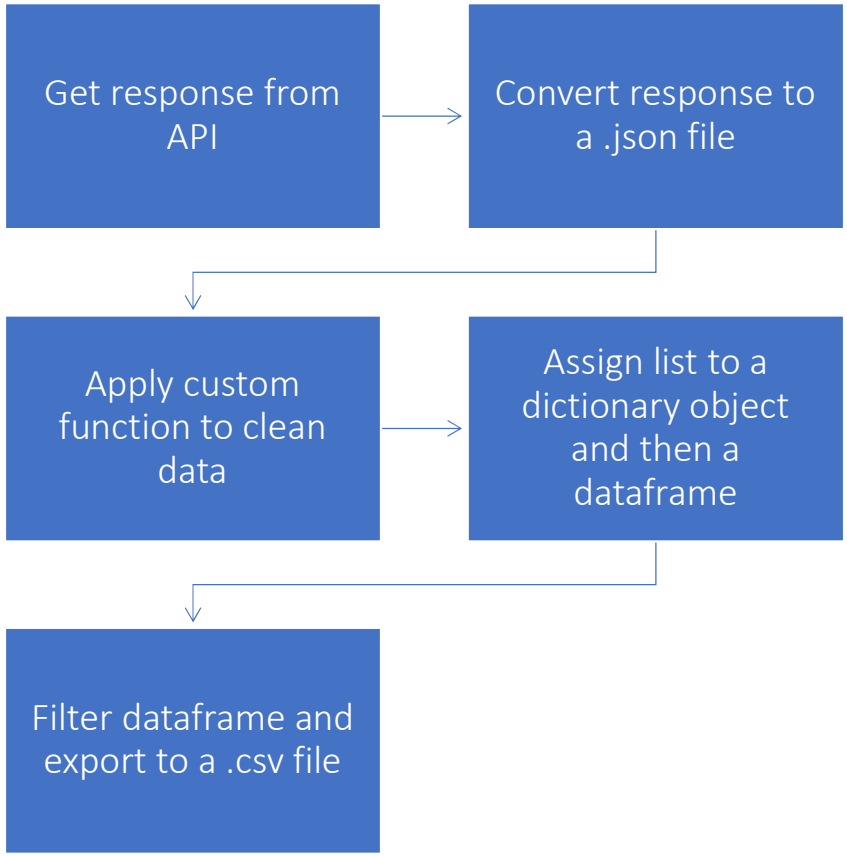
Executive Summary

- Data collection methodology:
  - SpaceX Rest API
  - Web Scrapping from Wikipedia
- Perform data wrangling
  - Check null values
  - Add “class” column to indicate landing successful or not
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - LR, KNN, SVM, DT models have been built and evaluated for the best classifier

Data Collection – SpaceX REST API

SpaceX REST API was used to collect data:

- The SpaceX REST API endpoints used is: api.spacexdata.com/v4/
- This API will give us data about launches, including information about the rocket used, payload delivered, launch specifications, landing specifications, and landing outcome.



Link to the Jupiter Notebook on Github:
https://github.com/weicai2015/DataScience/blob/master/Data%20Collection.ipynb

Data Collection – SpaceX API (Code)

1. Get response from requesting the URL

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

2. Convert response to a .json file

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

3. Apply custom function to clean data

```
getBoosterVersion(data)
getLaunchSite(data)
getPayloadData(data)
getCoreData(data)
```

4. Assign list to dictionary the dataframe

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

5. Export the dataframe to .CSV file

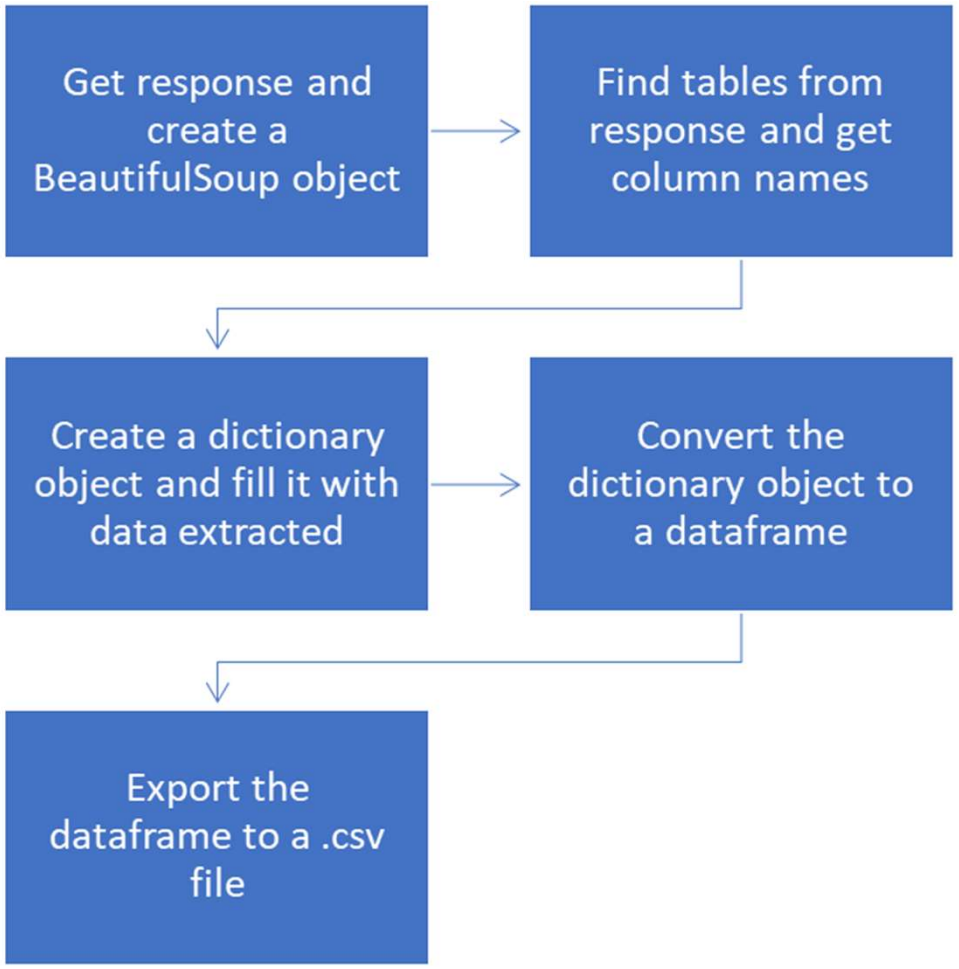
```
data_falcon9.to_csv('dataset_part_1.csv', index=False)
```

Data Collection – Web Scraping

Obtain Falcon 9 Launch data from web scraping Wikipedia :

- The url used is: https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922

- This Wiki page will give us data like flight number, launch site, payload, payload mass, obit, customer, launch outcome, booster version, and booster landing and etc.



Link to the Jupiter Notebook on github:
https://github.com/weicai2015/DataScience/blob/master/Data%20Collection%20with%20Web%20Scrapping.ipynb

Data Collection – Web Scraping (Code)

1. Get response, and create a BeautifulSoup Object

```
resp = requests.get( static_url )
bs = BeautifulSoup(resp.content, 'html.parser')
```

2. Find tables from response

```
html_tables = bs.find_all( 'table' )
first_launch_table = html_tables[2]
```

3. Get column names

```
column_names = []
ths = first_launch_table.find_all( 'th', {"scope" : "col"} )
for i in ths:
    i_name = extract_column_from_header( i )
    if( i_name is not None and len( i_name ) > 0 ):
        column_names.append( i_name )
```

4. Creating a dictionary object and initializing it to empty list (part of the code shown here)

```
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'] = []
```

5. Fill the dictionary object with data extracted (part of the code shown here)

```
for table_number,table in enumerate(bs.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)
            datatimelist=date_time(row[0])
```

6. Convert the dictionary object to a dataframe

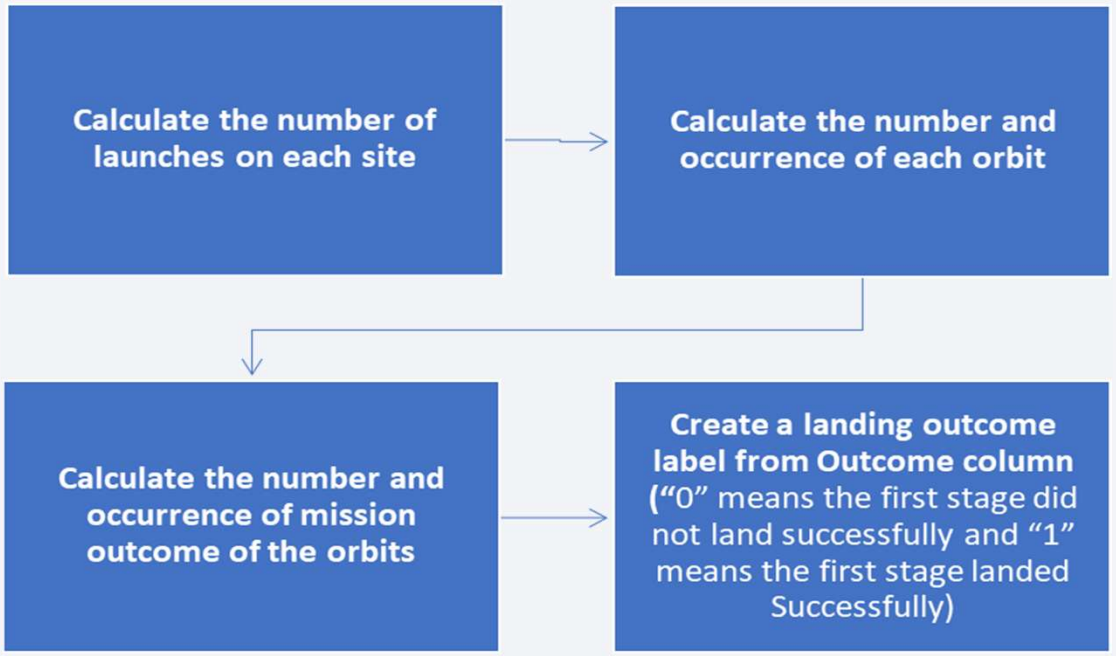
```
df=pd.DataFrame(launch_dict)
```

7. Export the dataframe to .CSV file

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

Link to the Jupiter Notebook on github:
https://github.com/weicai2015/DataScience/blob/master/Data%20Collection%20with%20Web%20Scrapping.ipynb

Data Wrangling



Link to Jupiter Notebook on Github:
https://github.com/weicai2015/DataScience/blob/master/Data%20Wrangling1.ipynb

EDA with Data Visualization

- Scatter chart
  - Flight Number vs. Launch Site
  - Payload mess vs. Launch Site
  - Flight Numbers vs. Orbit
  - Payload vs. Orbit
- Bar chart
  - Success Rate vs. Orbit
- Line chart
  - Launch Success Yearly Trend

Link to the Jupiter Notebook on Github:
https://github.com/weicai2015/DataScience/blob/master/EDA%20with%20Data%20Visualization.ipynb

EDA with SQL

SQL queries performed include:

1. Display the names of the unique launch sites in the pace mission
2. Display 5 records where launch sites begin with the string 'CCA'
3. Display the total payload mass carried by boosters launched by NASA(CRS)
4. Display average payload mass carried by booster version F9 v 1.1
5. List the data where the successful landing outcome in ground pad was achieved.
6. List the names of the boosters which have success in drone ship and have payload mass greater then 4000 but less then 6000
7. List the total number of successful and failure mission outcomes
8. List the names of the booster_versions which have carried the maximum payload mass.
9. List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
10. 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

The link to the Jupiter notebook in Github
https://github.com/weicai2015/DataScience/blob/master/EDA%20with%20SQL.ipynb

Build an Interactive Map with Folium

A Folium map object is built, it centered on NASA Space Center at Houston, TX. On this map:

- Mark each launch site with a red circle together with a marker shown its name
- Mark the success and failed launches for each site on the map. **Green** for successful landing and **red** for failed landing.
- Calculate the distances between a launch site to the key locations around it.

The link to the Jupiter notebook in Github
https://github.com/weicai2015/DataScience/blob/master/EDA%20with%20SQL.ipynb

Build a Dashboard with Plotly Dash

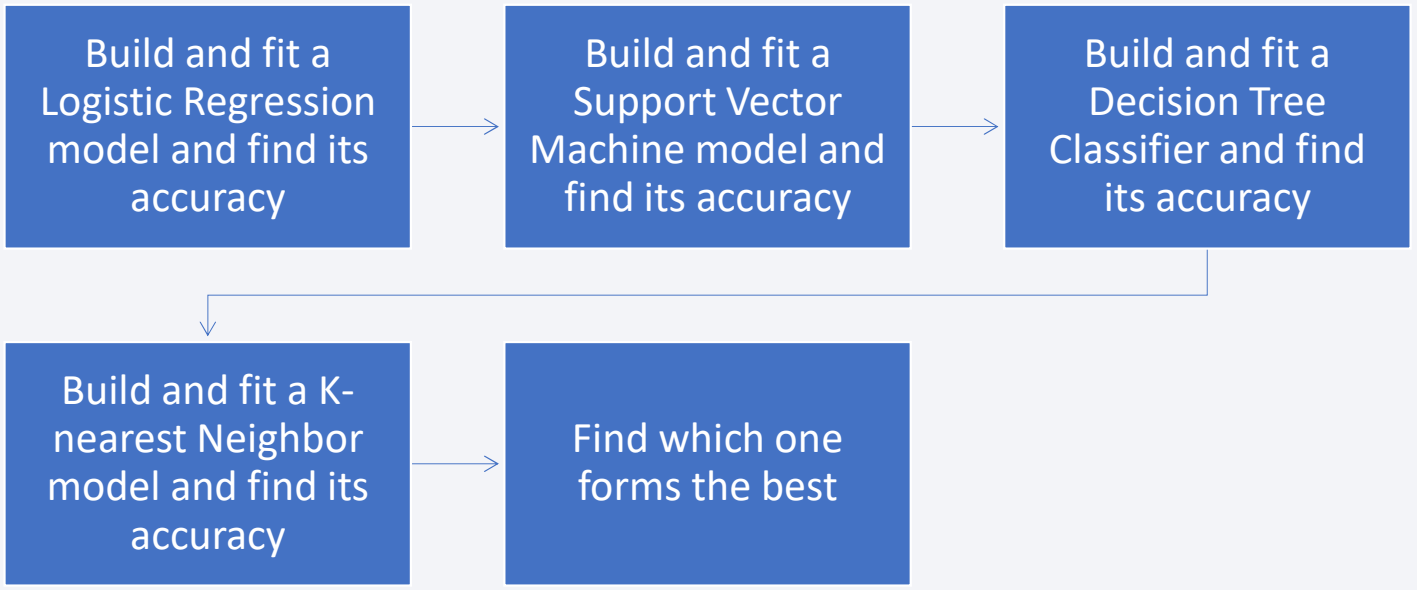
- A dropdown list of all launch sites is added to find out:
  - Total successful launches of all sites
  - Success rate of each site

- A sliding bar of payload mass is added to find out:
  - Payload range that has the highest launch success rate
  - Payload range that has the lowest launch success rate
  - F9 Booster version that has the highest launch success rate

Link to Python code on Github:
https://github.com/weicai2015/DataScience/blob/master/spacex_dash_app.py

Predictive Analysis (Classification)

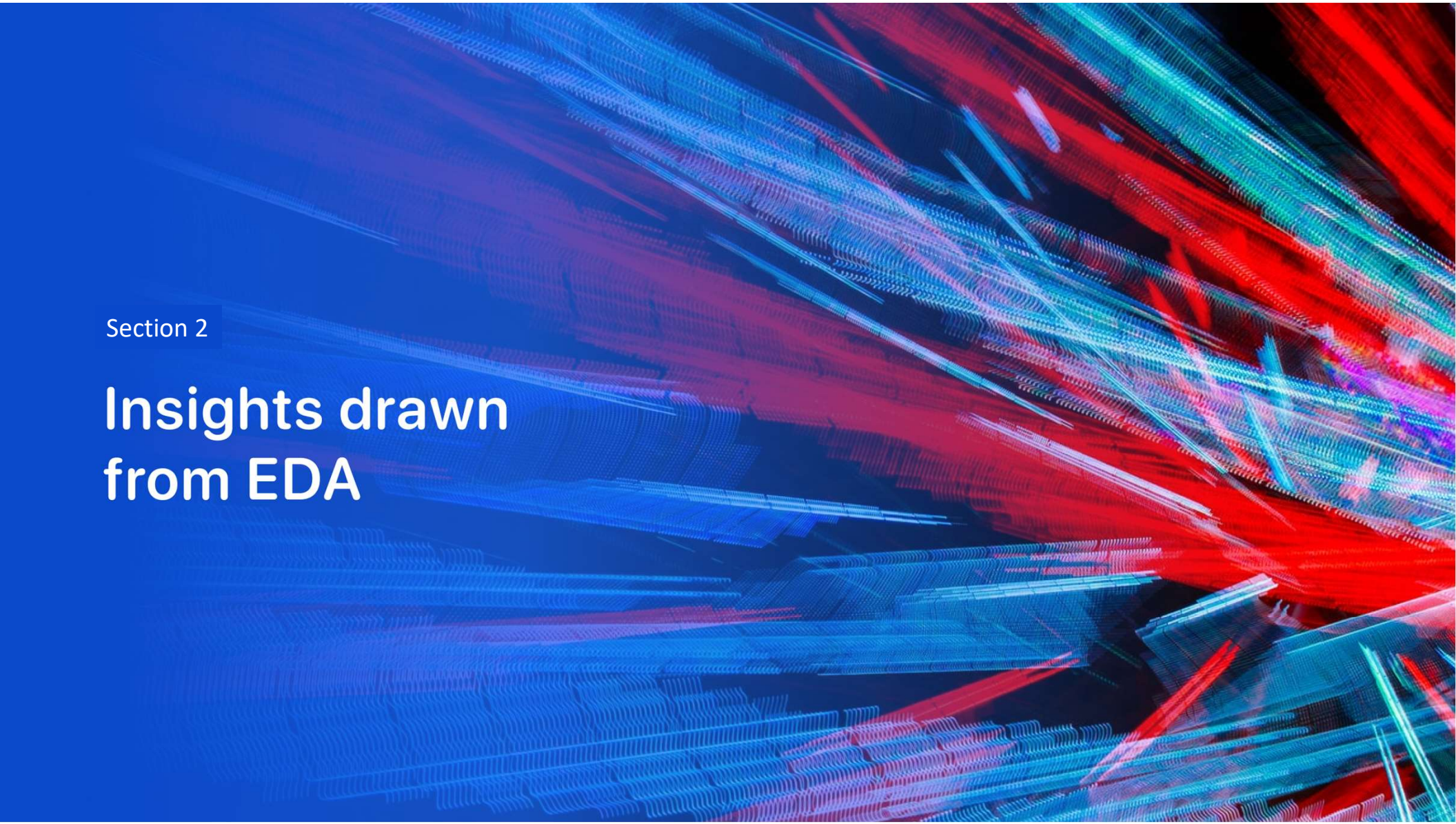
I built logistic regression, support Vector Machines, Decision Tree Classifier and K-nearest Neighbors models, analyzed their accuracy.



Link to the Jupiter Notebook on Github:
https://github.com/weicai2015/DataScience/blob/master/Landing%20Prediction.ipynb

Results

- EDA with data visualization
- EDA with SQL
- interactive analysis with Folium
- Interactive analysis with dashboard
- Prediction analysis



Section 2

Insights drawn from EDA

EDA with Visualization

Flight Number vs. Launch Site

Payload mess vs. Launch Site

Success Rate vs. Orbit

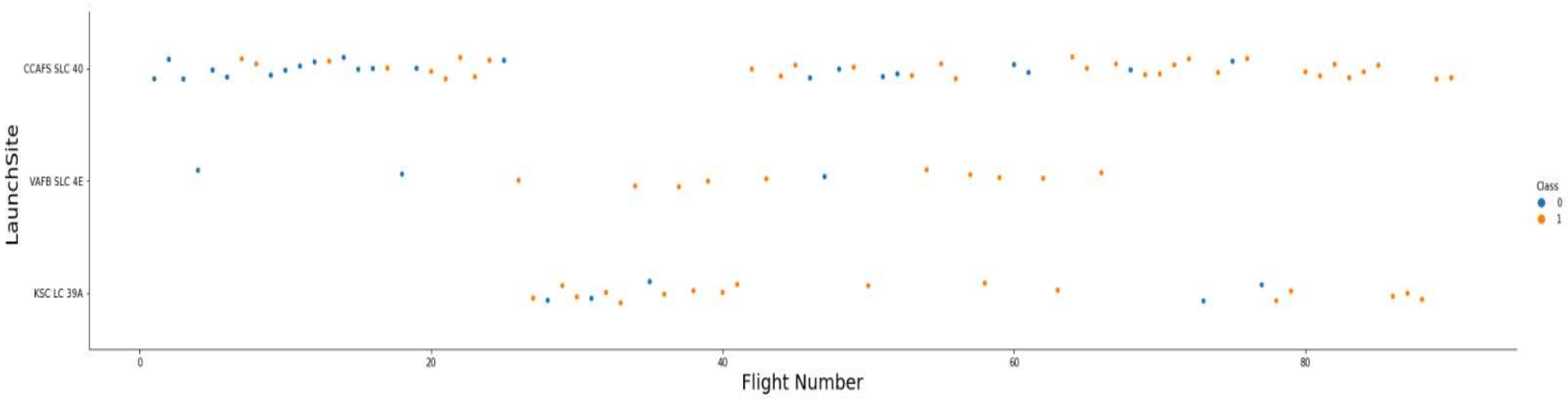
Flight Numbers vs. Orbit

Payload vs. Orbit

Launch Success Yearly Trend

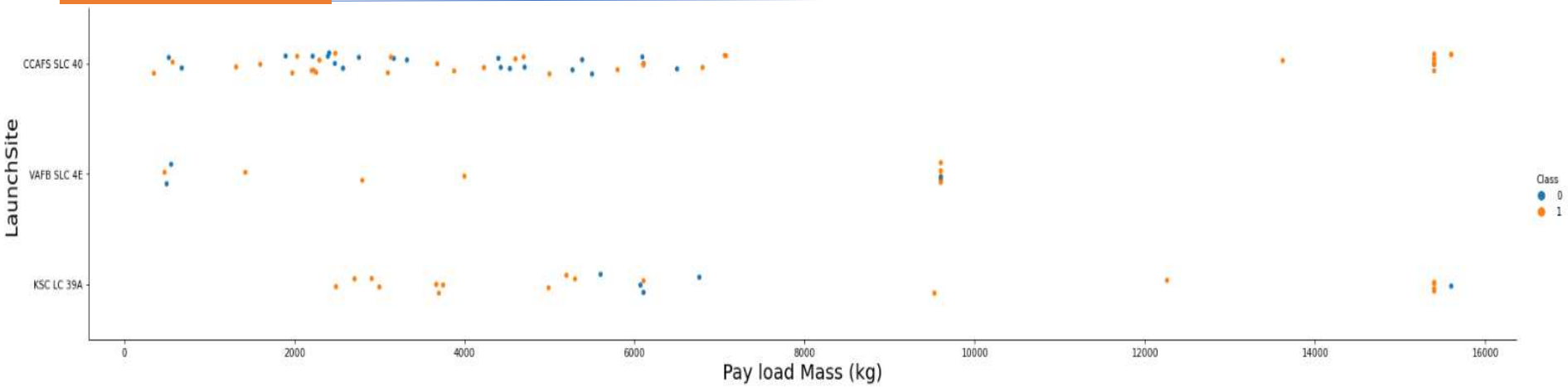
Link to the Jupiter Notebook on Github:
https://github.com/weicai2015/DataScience/blob/master/EDA%20with%20Data%20Visualization.ipynb

Flight Number vs. Launch Site



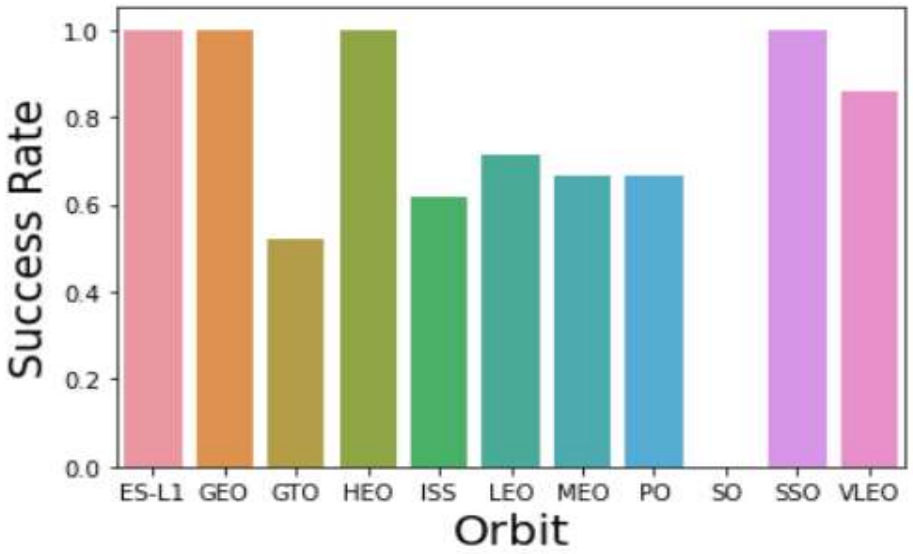
Launches from the site of CCAFS SLC 40 are significantly more than launches from other sites.

Payload Mass vs. Launch Site



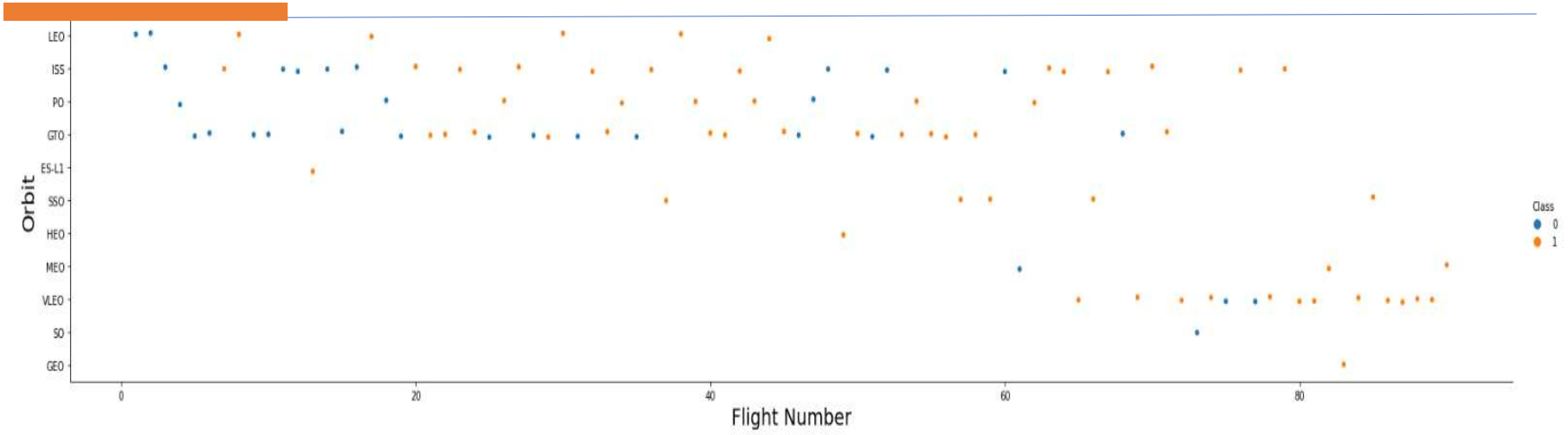
- The majority of Pay Loads with lower Mass (less than 10000kg) have been launched from site CCAFS SLC 40
- Launch site VAFB-SLC has no rockets launched for heavy payload mass(greater than 10000).

Success Rate vs. Orbit Type



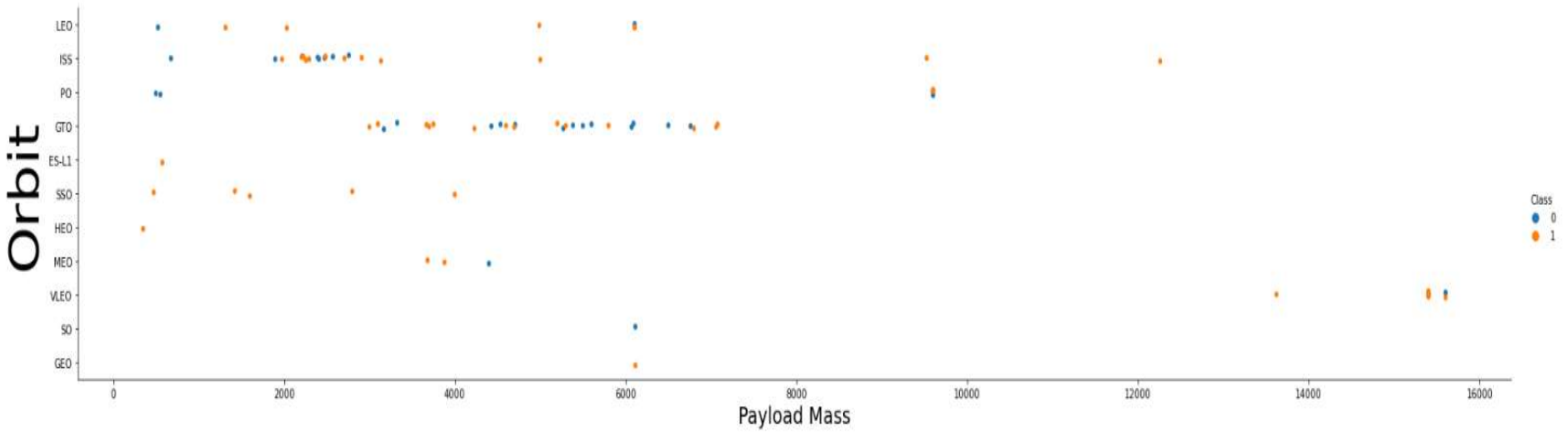
Orbit types of ES-L1, GEO, HEO, SSO have the highest success rate

Flight Number vs. Orbit Type



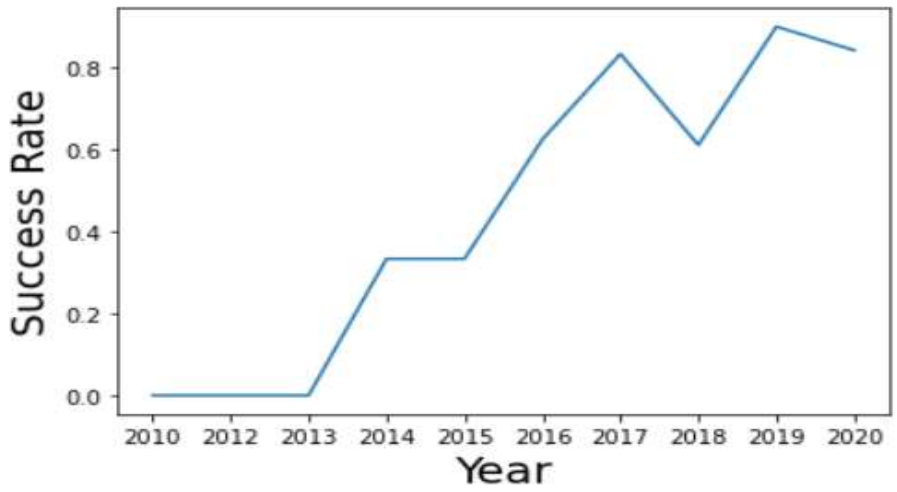
- The LEO orbit the success appears related to the number of flights;
- there seems to be no relationship between flight number and success when in GTO orbit.

Payload vs. Orbit Type



- With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS.
- However for GTO we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission) are both there here.

Launch Success Yearly Trend



The success rate has increased significantly since 2013 and then has been stable since 2019

EDA with SQL

SQL queries performed include:

1. Display the names of the unique launch sites in the pace mission
2. Display 5 records where launch sites begin with the string 'CCA'
3. Display the total payload mass carried by boosters launched by NASA(CRS)
4. Display average payload mass carried by booster version F9 v 1.1
5. List the data where the successful landing outcome in ground pad was achieved.
6. List the names of the boosters which have success in drone ship and have payload mass greater then 4000 but less then 6000
7. List the total number of successful and failure mission outcomes
8. List the names of the booster_versions which have carried the maximum payload mass.
9. List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
10. 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

The link to the Jupiter notebook in Github

https://github.com/weicai2015/DataScience/blob/master/EDA%20with%20SQL.ipynb

All Launch Site Names

SQL: select distinct("Launch_Site") from SPACEXTBL

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

Launch Site Names Begin with 'CCA'

SQL: select * from SPACEXTBL where Launch_Site like 'CCA%' limit 5

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing _Outcome
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

Total Payload Mass

SQL: select sum(PAYLOAD_MASS__KG_), Customer FROM SPACEXTBL where Customer='NASA (CRS)'

sum(PAYLOAD_MASS__KG_)	Customer
45596	NASA (CRS)

Average Payload Mass by F9 v1.1

SQL: select avg(PAYLOAD_MASS__KG_) from SPACEXTBL where Booster_Version like 'F9 v1.1%'

avg(PAYLOAD_MASS__KG_)
2534.6666666666665

First Successful Ground Landing Date

SQL: select "Date" from SPACEXTBL
where "Landing _Outcome" =
"Success (ground pad)"

Date
22-12-2015
18-07-2016
19-02-2017
01-05-2017
03-06-2017
14-08-2017
07-09-2017
15-12-2017
08-01-2018

Successful Drone Ship Landing with Payload between 4000 and 6000

sQL: select Booster_Version from SPACEXTBL
where Landing_Outcome = "Success (drone ship)" and PAYLOAD_MASS__KG_ between 4000 and 6000

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

Total Number of Successful and Failure Mission Outcomes

SQL: select count(*), Mission_Outcome
from SPACEXTBL
where Mission_Outcome like 'Failure%'
or Mission_Outcome like 'Success%'
group by Mission_Outcome

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

Boosters Carried Maximum Payload

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

2015 Launch Records

SQL: select Date, booster_version, launch_site, landing_outcome
from SPACEXTBL where substr(Date, 7, 4) = '2015' and landing_outcome = "Failure (drone ship)"

Date	Booster_Version	Launch_Site	Landing_Outcome
01/10/2015	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
14/04/2015	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

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

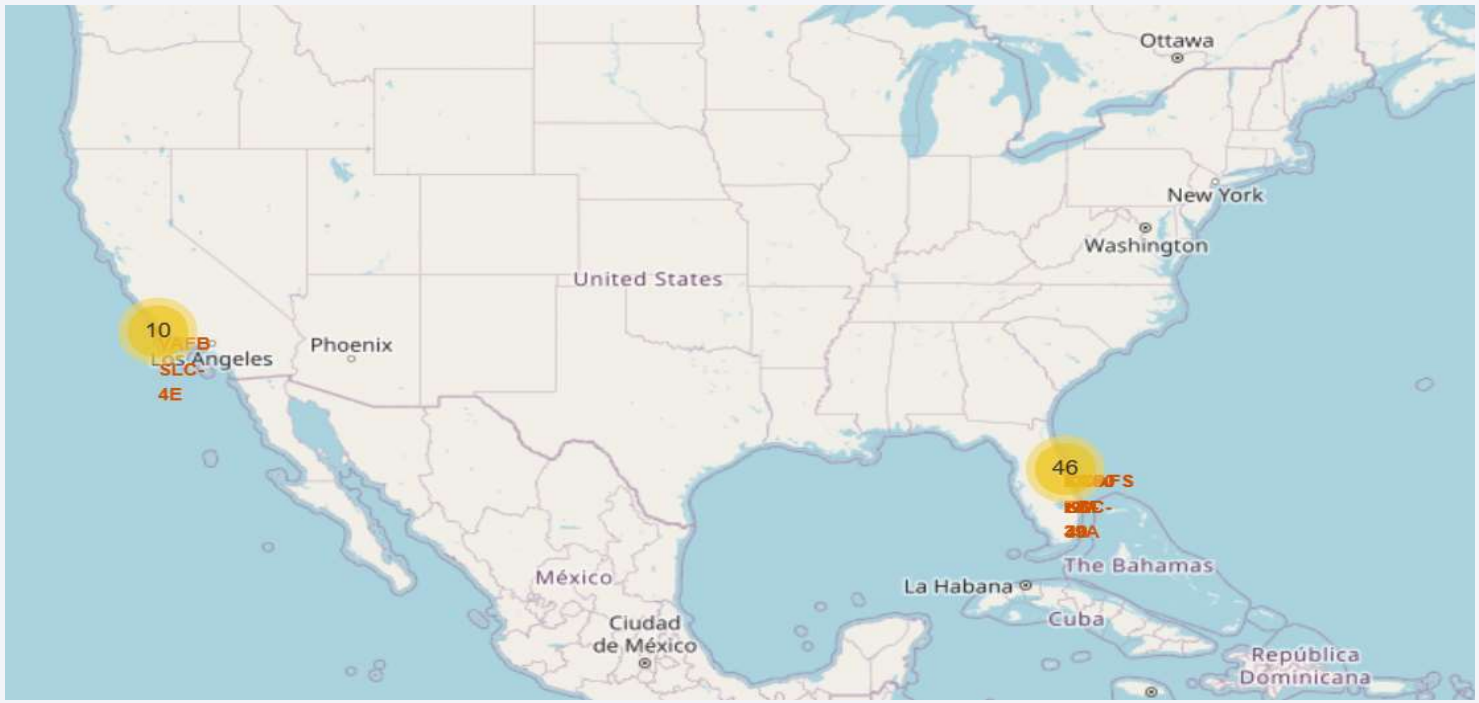
SQL: select count(*) c, Landing_Outcome
from SPACEXTBL
where Date between "04-06-2010" and "20-03-2017" group by Landing_Outcome order by c desc

c	Landing_Outcome
20	Success
10	No attempt
8	Success (drone ship)
7	Success (ground pad)
3	Failure (drone ship)
3	Failure
2	Failure (parachute)
2	Controlled (ocean)
1	No attempt

Section 3

Launch Sites Proximities Analysis

Map with All Launch Sites Marked Out

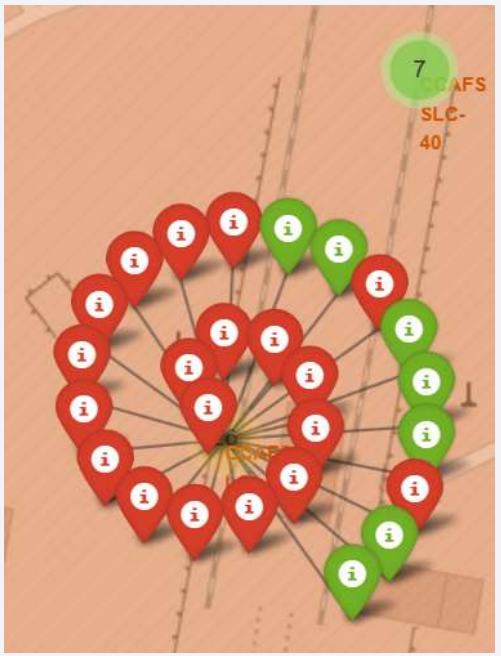


Site VAFB SLC-4E is on the west coast, and the other 3 sites are on the east coast, and they are pretty close.

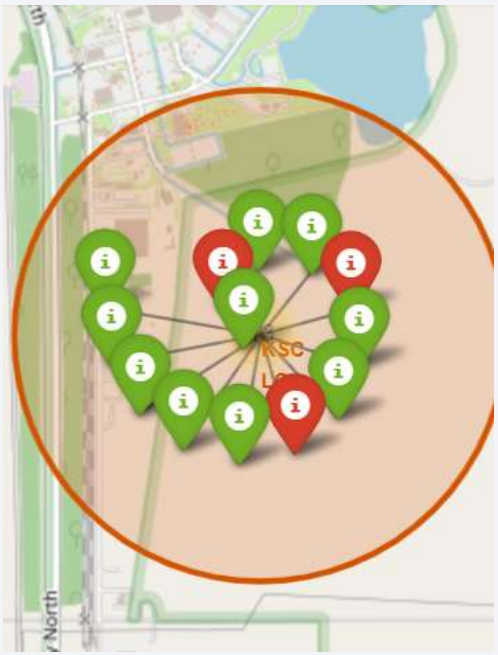
Map with markers of success/failed launches for each site

CCAFS SLC-40

VSFB SLC-4E



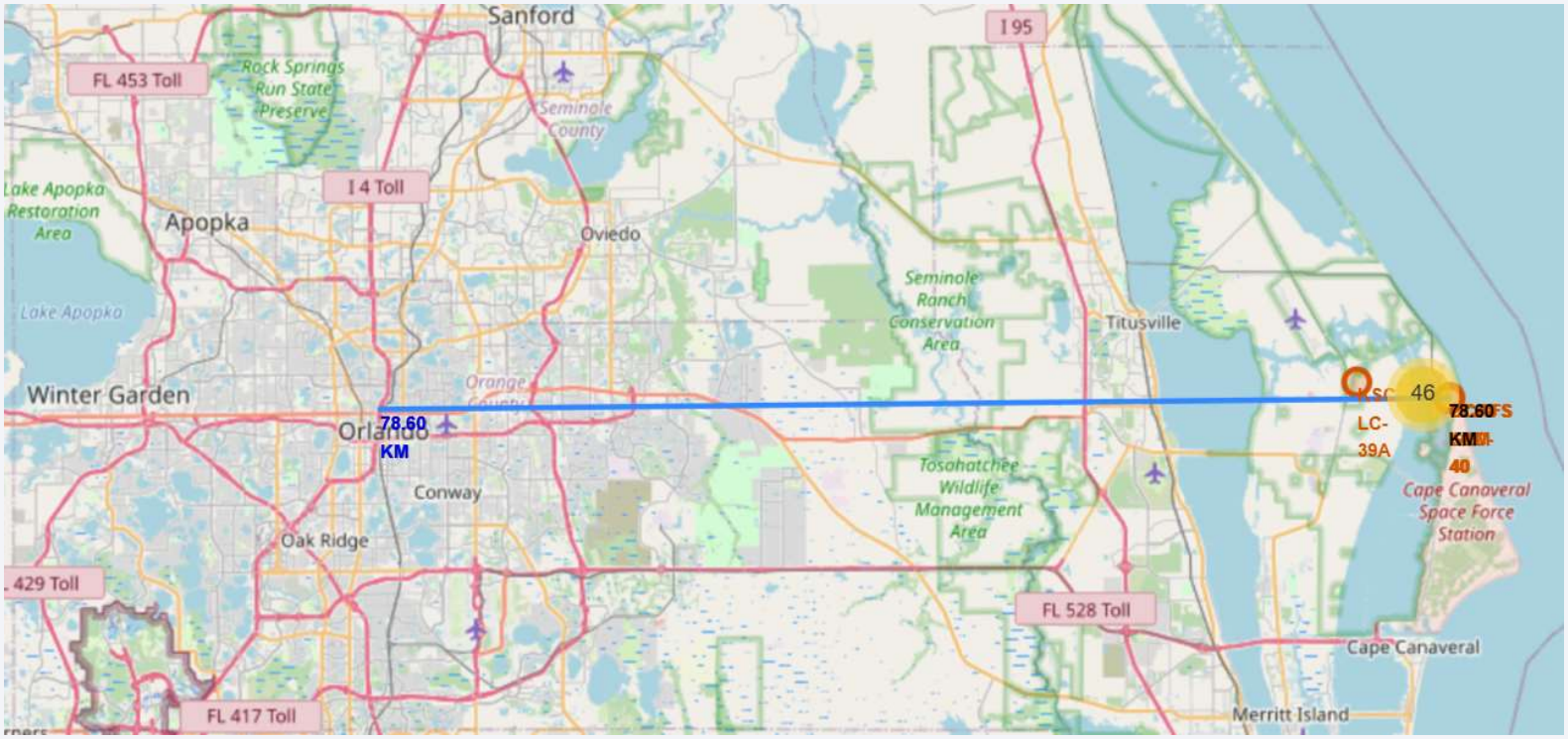
CCSFS LC-40



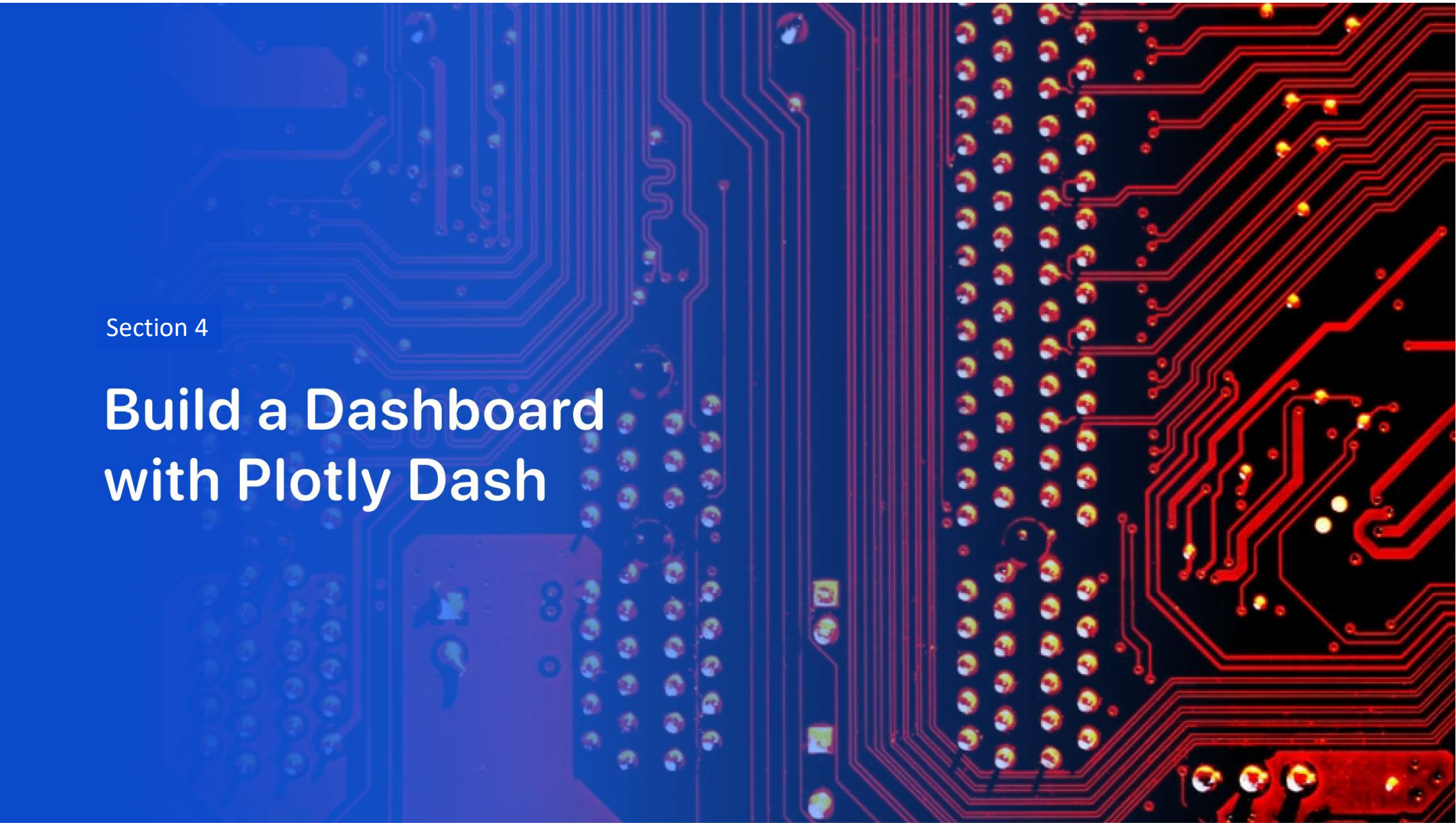
KSC LC-39A

Four parts of the whole map is shown here. Green marker means success landings, and red means failed landings.

Map with marker of distance between launch sites and key locations around them



Above is a map shows distance between site CCAFS SLC-40 and city Orlando, the distance is 78.598km.



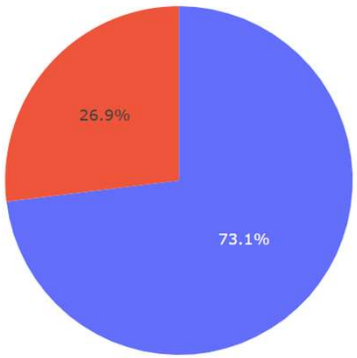
Section 4

Build a Dashboard with Plotly Dash

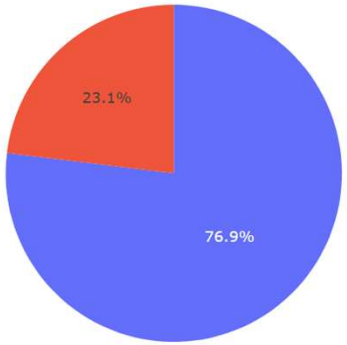
Total successful launches of all sites

We can see site KSC LC-39A has the most successful launches, and the second one is CCAFS LC-40.

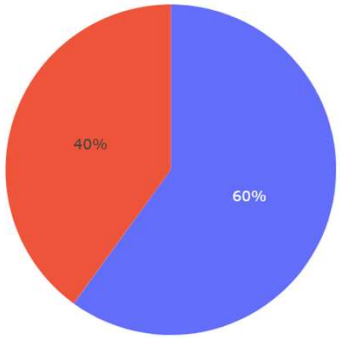
Success rate of each site



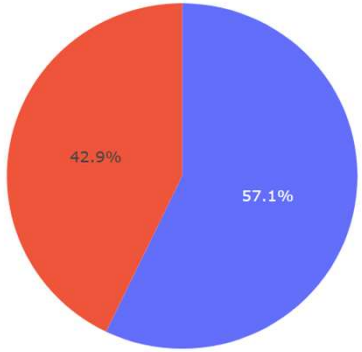
CCSFS LC-40



KSC LC-39A



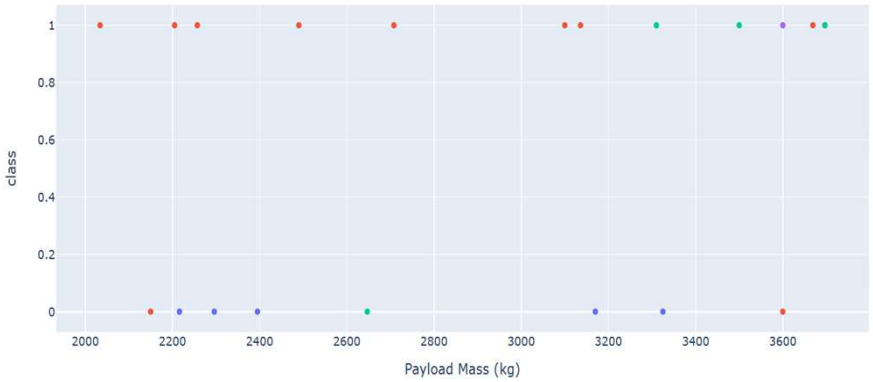
VSFB SLC-4E



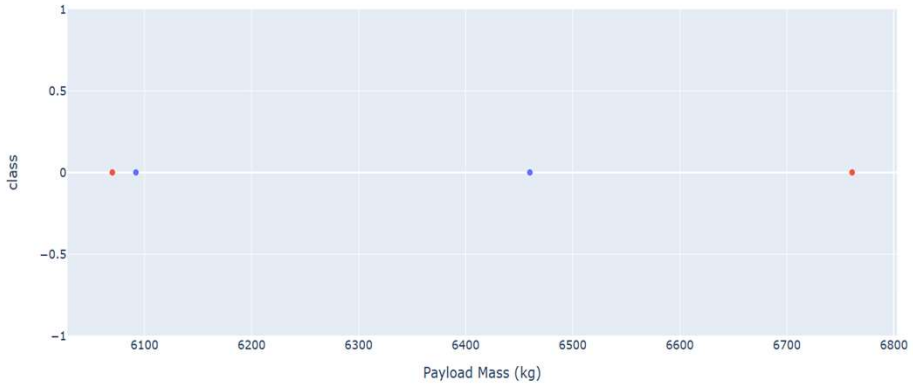
CCAFS SLC-40

We can see site KSC LC-39A has the largest successful launch rate, which is 76.9% and CCAFS SLC-40 has the least, which is 57.1%

Payload range vs. launch outcomes (1)



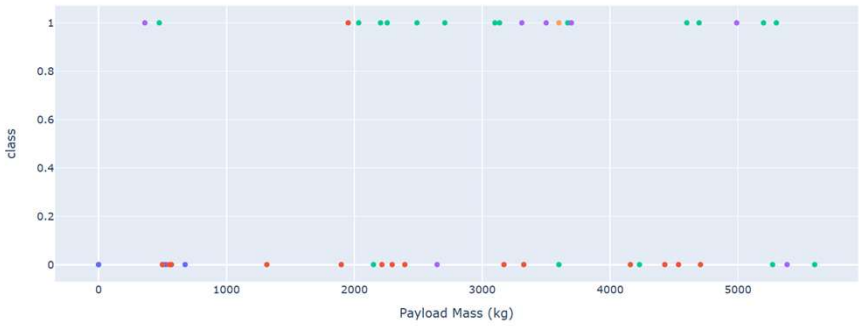
Payload 2000-4000kg



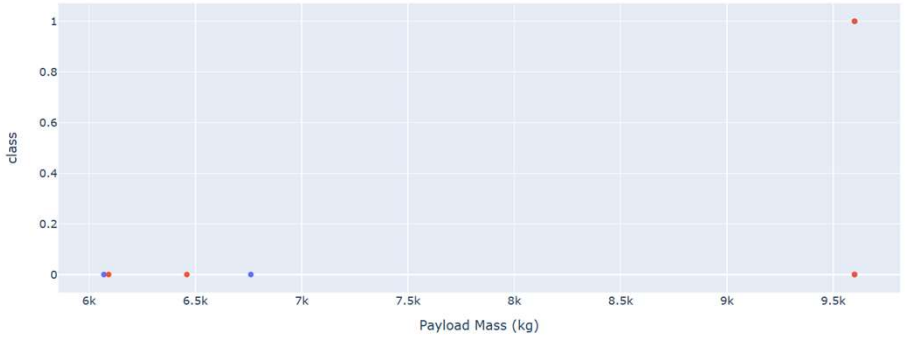
Payload 6000-8000kg

We can see payload 2000-4000kg has the highest launch success rate and payload 6000-8000kg has the lowest launch success rate.

Payload range vs. launch outcomes (2)



Payload 0-6000kg



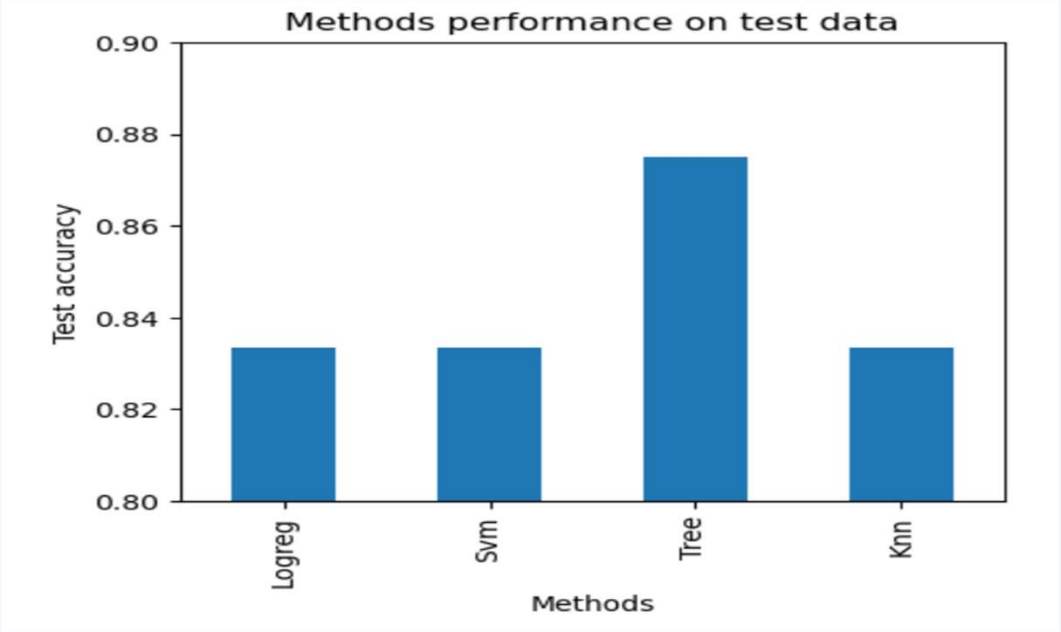
Payload 6000-10000kg

We can also see payload 0-6000kg has much higher launch success rate than payload 6000-10000kg, booster version FT has the highest launch success rate.

Section 5

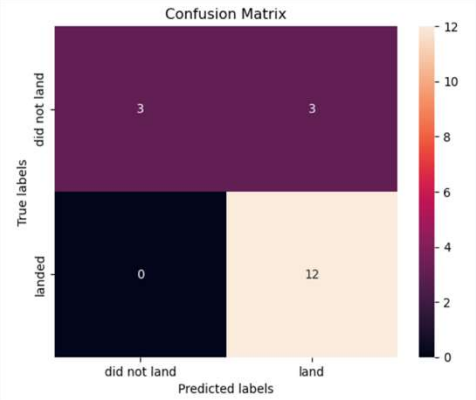
Predictive Analysis (Classification)

Classification Accuracy

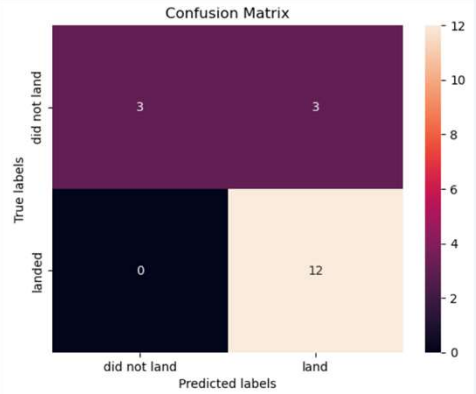


We can see Decision Tree classifier has the most accuracy.

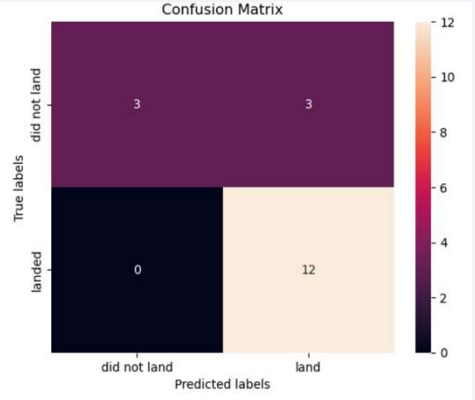
Confusion Matrix



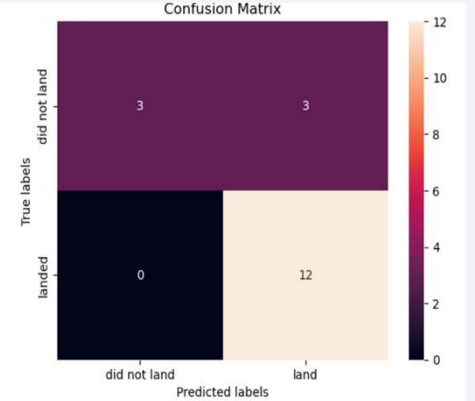
Logistic Regression



Support Vector Machine



Decision Tree



K-nearest Neighbors

We can see the confusion Matrixes of all 4 methods are pretty much the same.

Conclusions

- The success rate has increased significantly since 2013 and then has been stable since 2019
- Orbit ES-L1 GEO, HEO, SSO, have the best Success Rate.
- Site KSC LC-39A has the most successful launches and it also has the largest successful launch rate
- Payload 2000-4000kg has the highest launch success rate and payload 6000-8000kg has the lowest launch success rate. In a broader range, payload 0-6000kg has much higher launch success rate than payload 6000-10000kg.
- Booster version FT has the highest launch success rate.
- The decision tree model is the best in terms of prediction accuracy for this dataset.
- The success rates for SpaceX launches is directly proportional time in years they will eventually perfect the launches

Appendix – Links on Github for reference

- Data collection using SpaceX REST API:
  https://github.com/weicai2015/DataScience/blob/master/Data%20Collection.ipynb
- Data collection using Web Scrapping:
  https://github.com/weicai2015/DataScience/blob/master/Data%20Collection%20with%20Web%20Scrapping.ipynb
- Data Wrangling:
  https://github.com/weicai2015/DataScience/blob/master/Data%20Wrangling1.ipynb
- EDA with data visualization:
  https://github.com/weicai2015/DataScience/blob/master/EDA%20with%20Data%20Visualization.ipynb
- EDA with SQL:
  https://github.com/weicai2015/DataScience/blob/master/EDA%20with%20SQL.ipynb
- Python code to build Plotly Dash Dashboard on Github:
  https://github.com/weicai2015/DataScience/blob/master/spacex_dash_app.py
- Interactive analysis with Folium:
  https://github.com/weicai2015/DataScience/blob/master/Launch%20site%20analysis%20with%20Folium.ipynb
- Predictive analysis:
  https://github.com/weicai2015/DataScience/blob/master/Landing%20Prediction.ipynb

Thank you!