IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

CHIEN, CHUN-YEH
2022/11/12

Outline

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

Executive Summary

- **Summary of methodologies**
- Data Collecting
- Data Wrangling
- EDA with visualization and SQL
- Build an Interactive Map with Folium
- Build a Dashboard Application with Plotly Dash
- Predictive Analysis (Classification)
- **Summary of all results**
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Introduction

- 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.

- We will predict if the Falcon 9 first stage will land successfully.

Section 1

Methodology

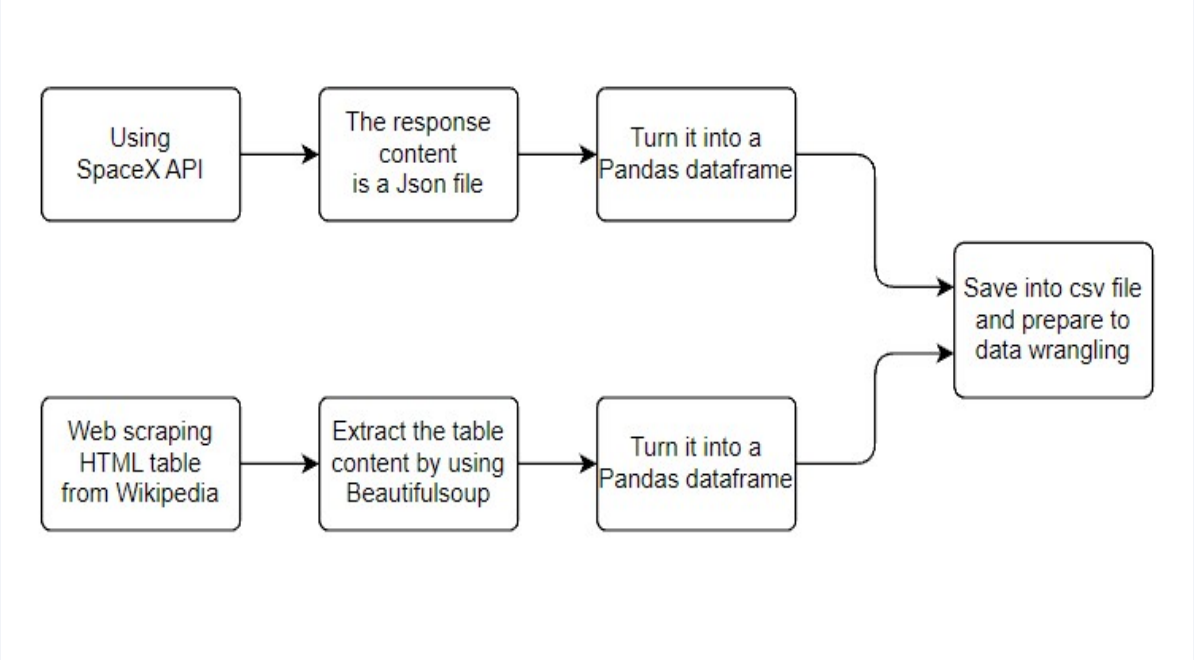
Methodology

Executive Summary

- Data collection methodology
  - Request to the SpaceX API
  - Web scraping Falcon 9 and Falcon Heavy Launches Records from Wikipedia
- Perform data wrangling
  - Dealing with Missing Values ; Determine Training Labels
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Find best Hyperparameter for SVM, KNN, Classification Trees and Logistic Regression
  - Check each of R2_score and confusion matrix to evaluation

Data Collection

- Using SpaceX API
- API URL : https://api.spacexdata.com/v4/launches/past
- This API will give us data about launches, including information about the rocket used, payload delivered, launch specifications, landing specifications, and landing outcome.



- Another data source is Wikipedia page
- Extract all column/variable names from the HTML table header

Data Collection – SpaceX API

- Data collection with SpaceX REST API

- https://github.com/Joe-Chien/Pythons-Basics-for-Data-Science-Project/blob/master/jupyter-labs-spacex-data-collection-api.ipynb

Request and parse the SpaceX launch data using the GET request

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

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

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

Applying 4 functions help us use the API to extract information

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

Turn into pandas DF

```
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}
data_falcon = pd.DataFrame(launch_dict)
```

Filter the dataframe to only include `Falcon 9` launches

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

Export it to CSV file

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

Data Collection - Scraping

- Web scraping from Wikipedia

- https://github.com/Joe-Chien/Pythons-Basics-for-Data-Science-Project/blob/master/jupyter-labs-webscraping.ipynb

Request the Falcon9 Launch Wiki page from its URL

```
response = requests.get(static_url)
soup = BeautifulSoup(response.text)
```

Extract all column/variable names from the HTML table header

```
html_tables = soup.find_all(name = 'tr')
```

```
column_names = []
first_launch_table_th = first_launch_table.find_all('th')
print(first_launch_table_th)
for row in first_launch_table_th:
    col = extract_column_from_header(row)
    if len(col)>0:
        column_names.append(col)
```

Create a data frame by parsing the launch HTML tables

```
launch_dict= dict.fromkeys(column_names)
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'] = []
launch_dict['Version Booster']=[]
launch_dict['Booster landing']=[]
launch_dict['Date']=[]
launch_dict['Time']=[]
```

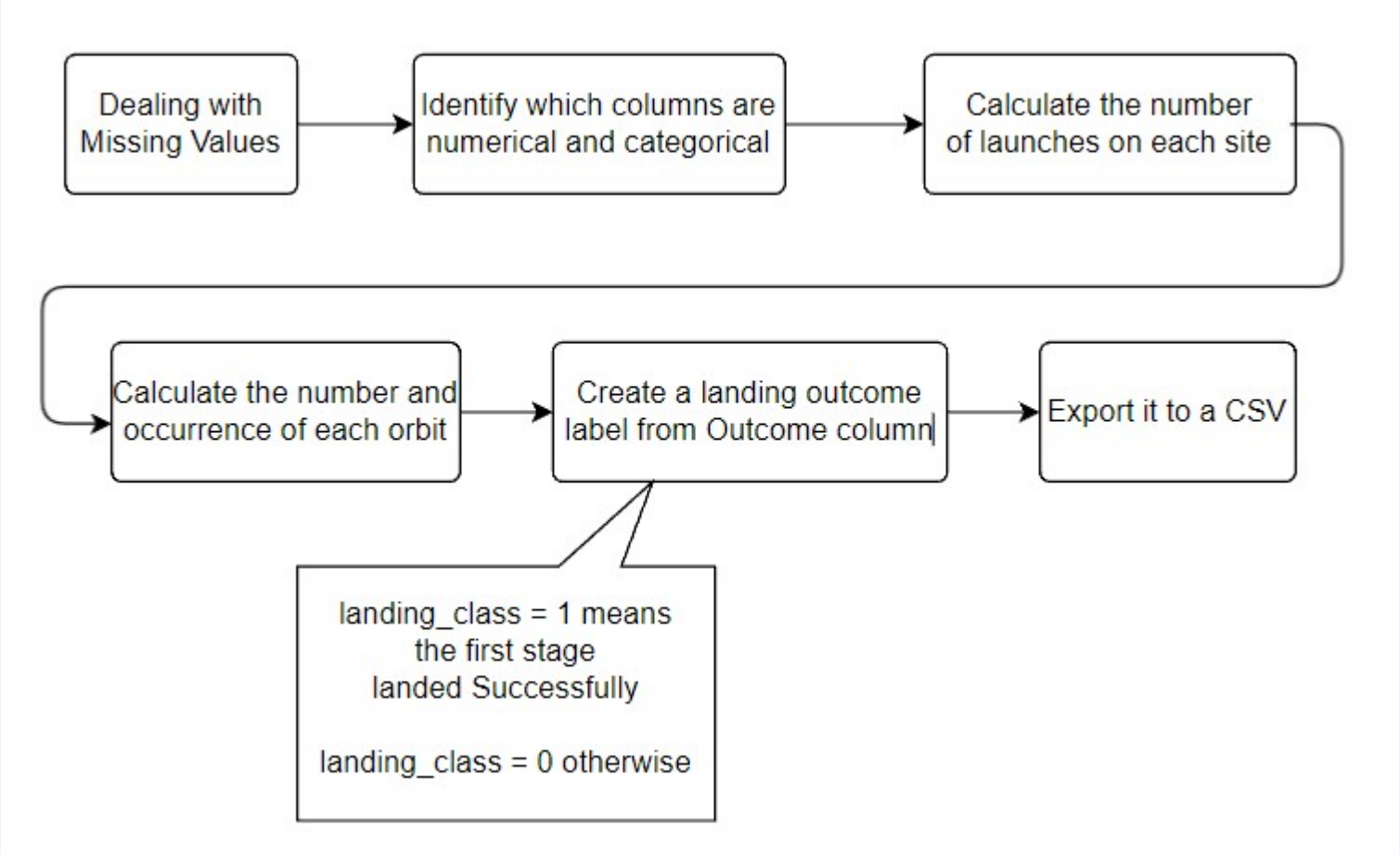
```
extracted_row = 0
#Extract each table
for table_number,table in enumerate(soup.find_all('table','
   # get table row
    for rows in table.find_all("tr"):
        #check to see if first table heading is as number
        if rows.th:
            if rows.th.string:
                flight_number=rows.th.string.strip()
                flag=flight_number.isdigit()
```

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

Export it to CSV file

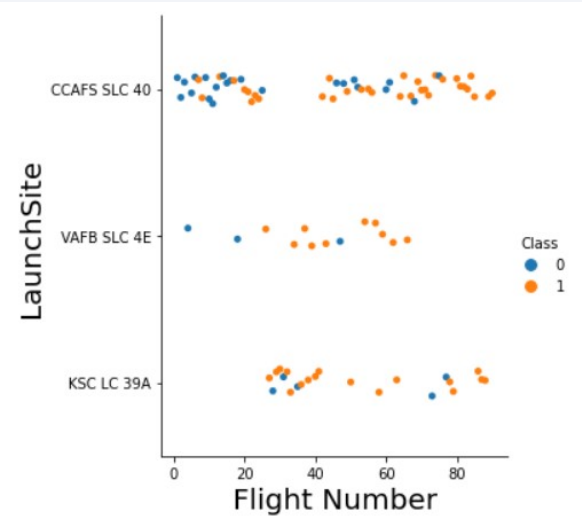
```
df.to_csv('spacex_web_scraped.csv', index=False)
```

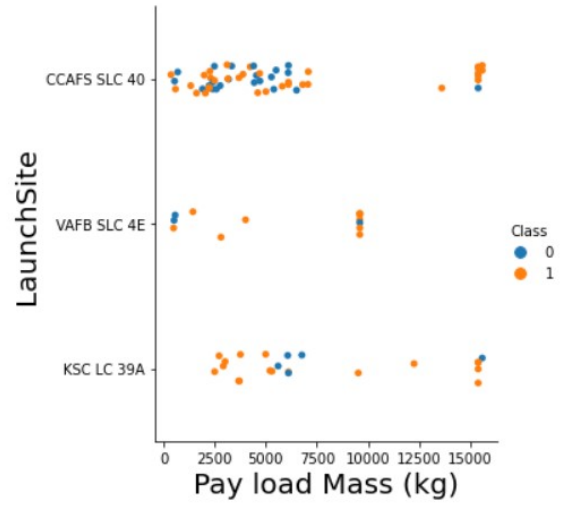
Data Wrangling

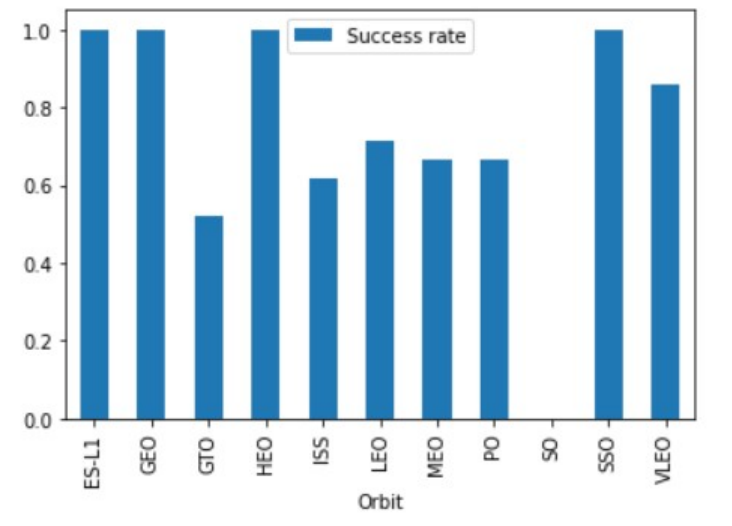


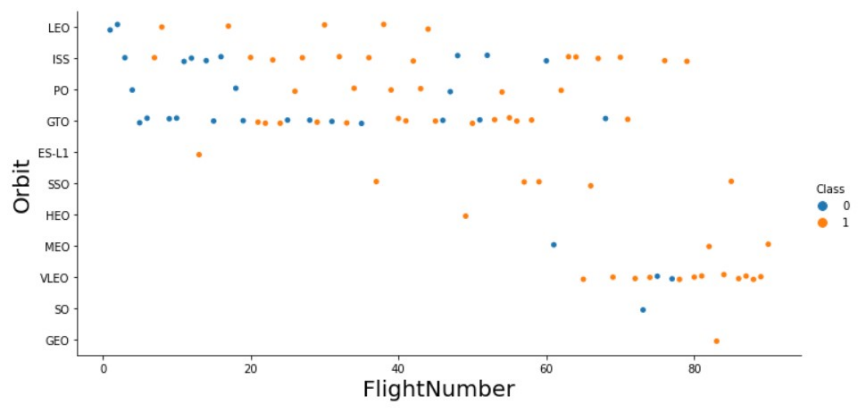
https://github.com/Joe-Chien/Pythons-Basics-for-Data-Science-Project/blob/master/labs-jupyter-spacex-Data%20wrangling.ipynb

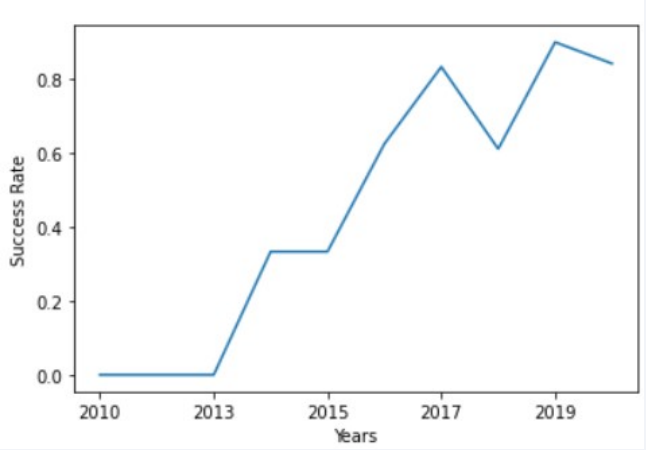
EDA with Data Visualization











https://github.com/Joe-Chien/Pythons-Basics-for-Data-Science-Project/blob/master/jupyter-labs-eda-dataviz.ipynb

EDA with SQL

- SQL quries performed include:
  - Display 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 acheived
  - 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 failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
  - 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
- https://github.com/Joe-Chien/Pythons-Basics-for-Data-Science-Project/blob/master/jupyter-labs-eda-sql-coursera.ipynb

Build an Interactive Map with Folium

- Mark all launch sites on a map (orange circle area)
- Mark the success/failed launches for each site on the map
- If a launch was successful (class=1), then we use a green marker and if a launch was failed, we use a red marker (class=0)
- Calculate the distances between a launch site to its proximities

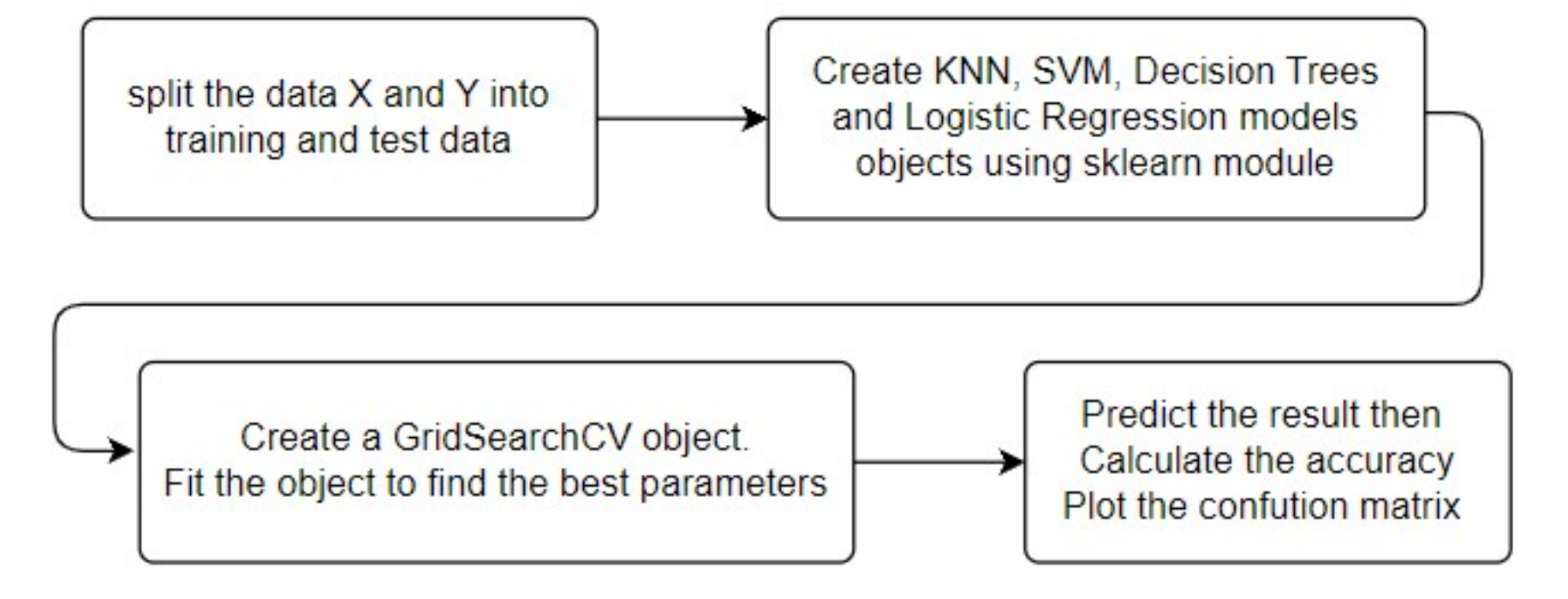
- We found that each launch station is close to the railway, the coastline, the highway, and some distance from the city
- https://github.com/Joe-Chien/Pythons-Basics-for-Data-Science-Project/blob/master/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

- Create a pie chart to show Total Success Launches for site (All sites or select ones)
- Create a scatter plot to show success count on Payload mass for site (All sites or select ones) , we also create a range slider to select payload range (Kg)

- Dashboards can produce real-time visuals.
- It will help business make informed decisions, thereby improving performance.
- https://github.com/Joe-Chien/Pythons-Basics-for-Data-Science-Project/blob/master/spacex_dash_app.ipynb

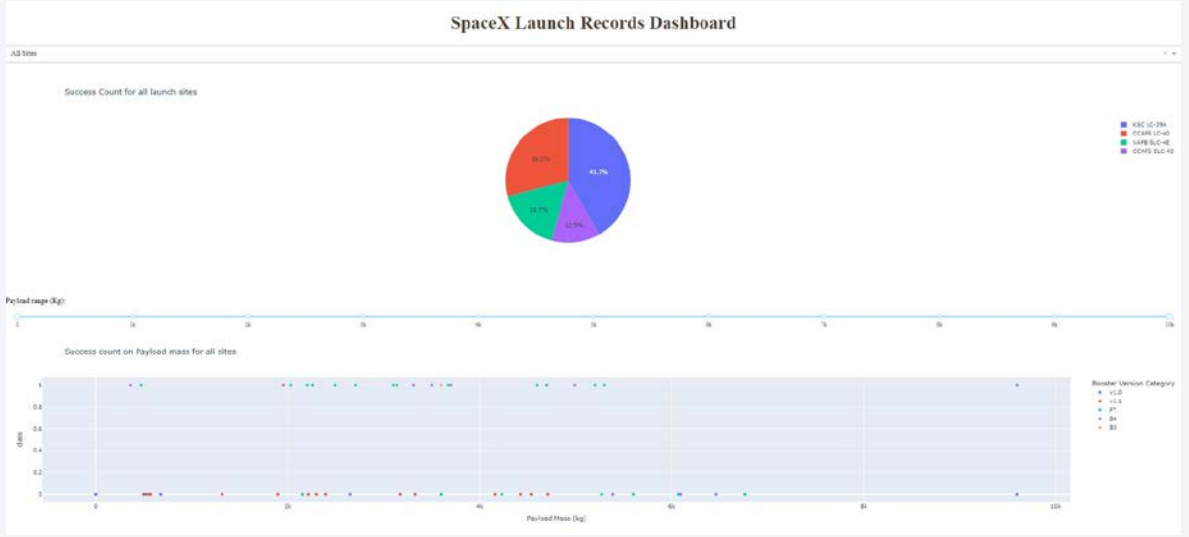
Predictive Analysis (Classification)



- https://github.com/Joe-Chien/Pythons-Basics-for-Data-Science-Project/blob/master/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results

- The success rate increases with each passing year
- ES-L1, GEO, HEO and SSO orbit type has highest sucess rate
- Interactive analytics

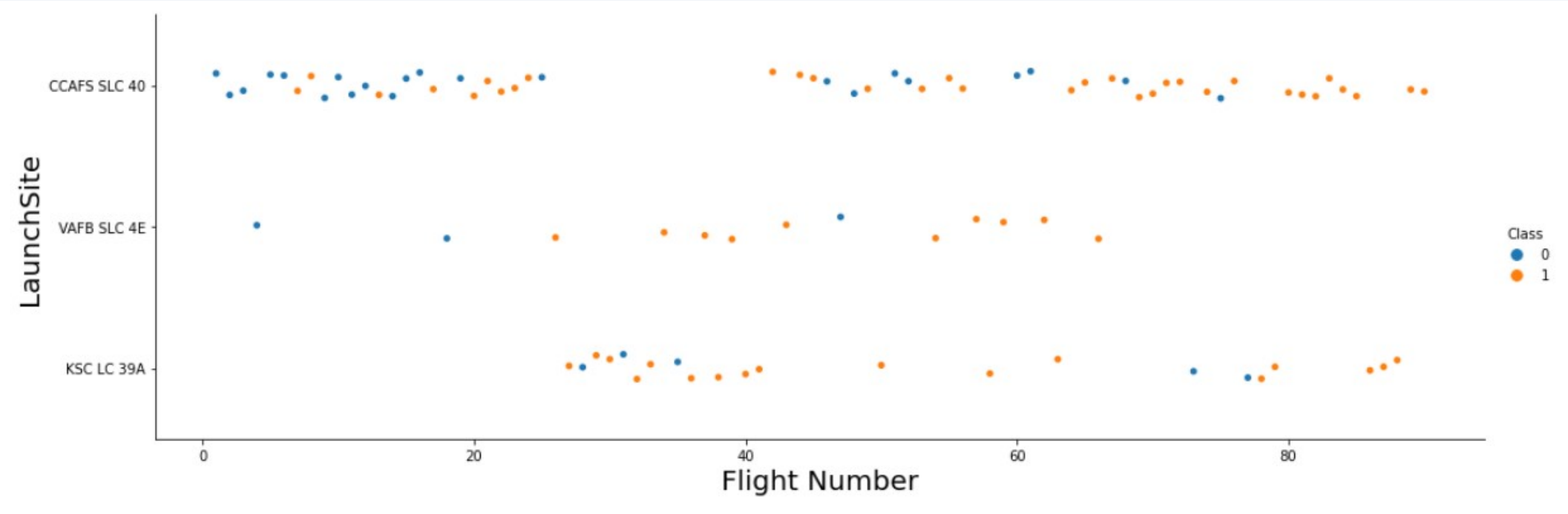


- Practically all predicted algorithms give the same result
- Accuracy score close to 83%

Section 2

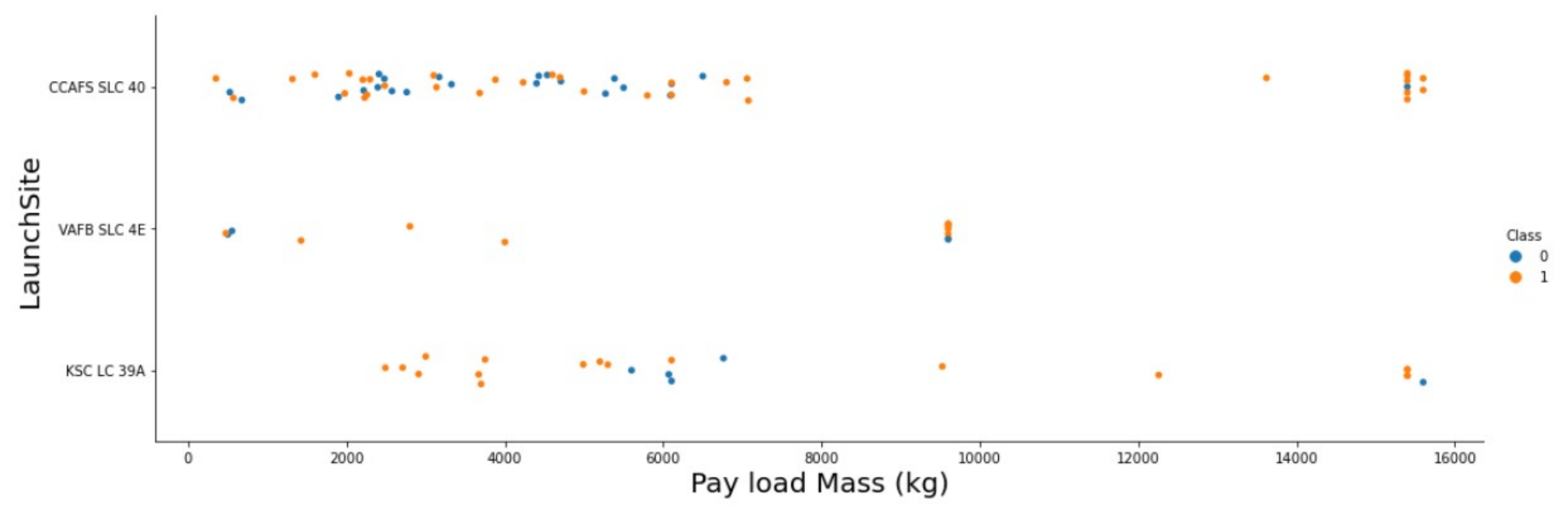
Insights drawn from EDA

Flight Number vs. Launch Site



- As the number of launches from the site of CCAFS SLC 40 increases, the launch success rate also increases

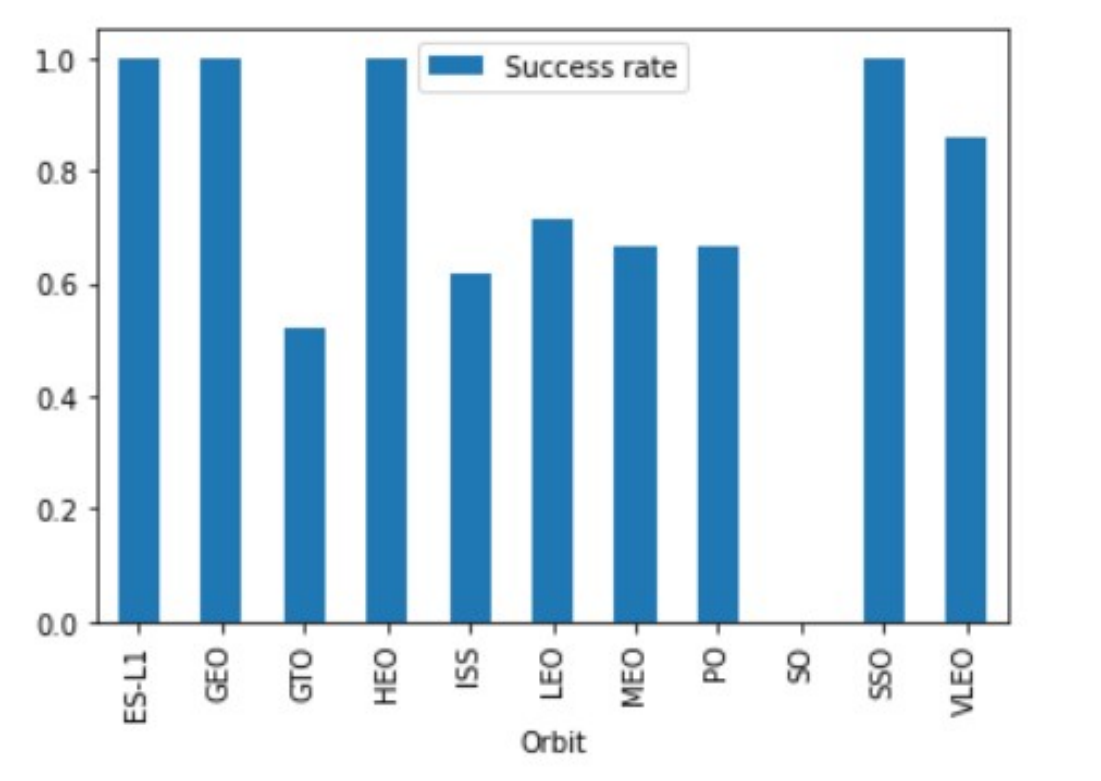
Payload vs. Launch Site



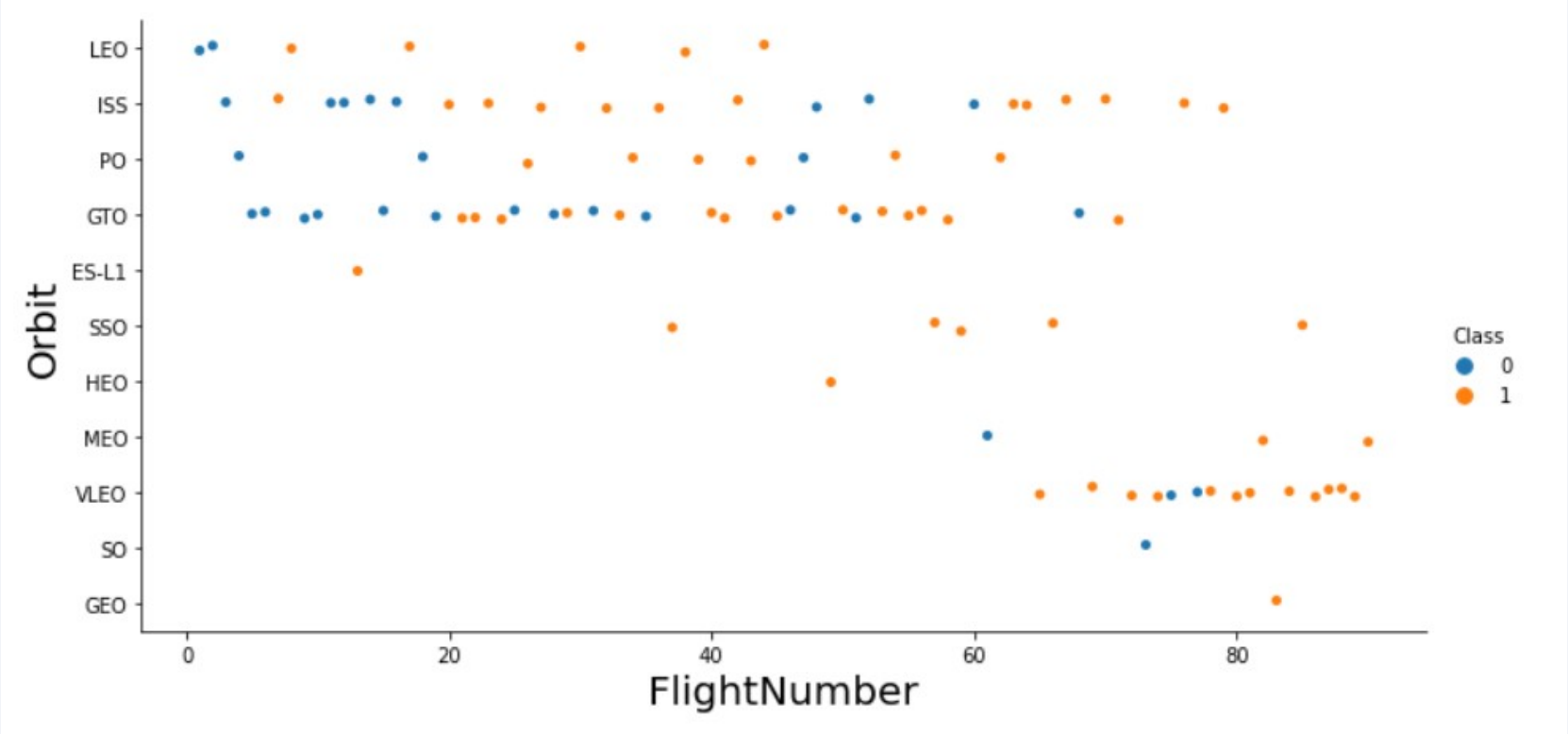
- Launches from the site of CCAFS SLC 40 are much higher usage than launches from other sites

Success Rate vs. Orbit Type

- The orbit type of ES-L1, GEO, HEO, SSO are among the highest success rate

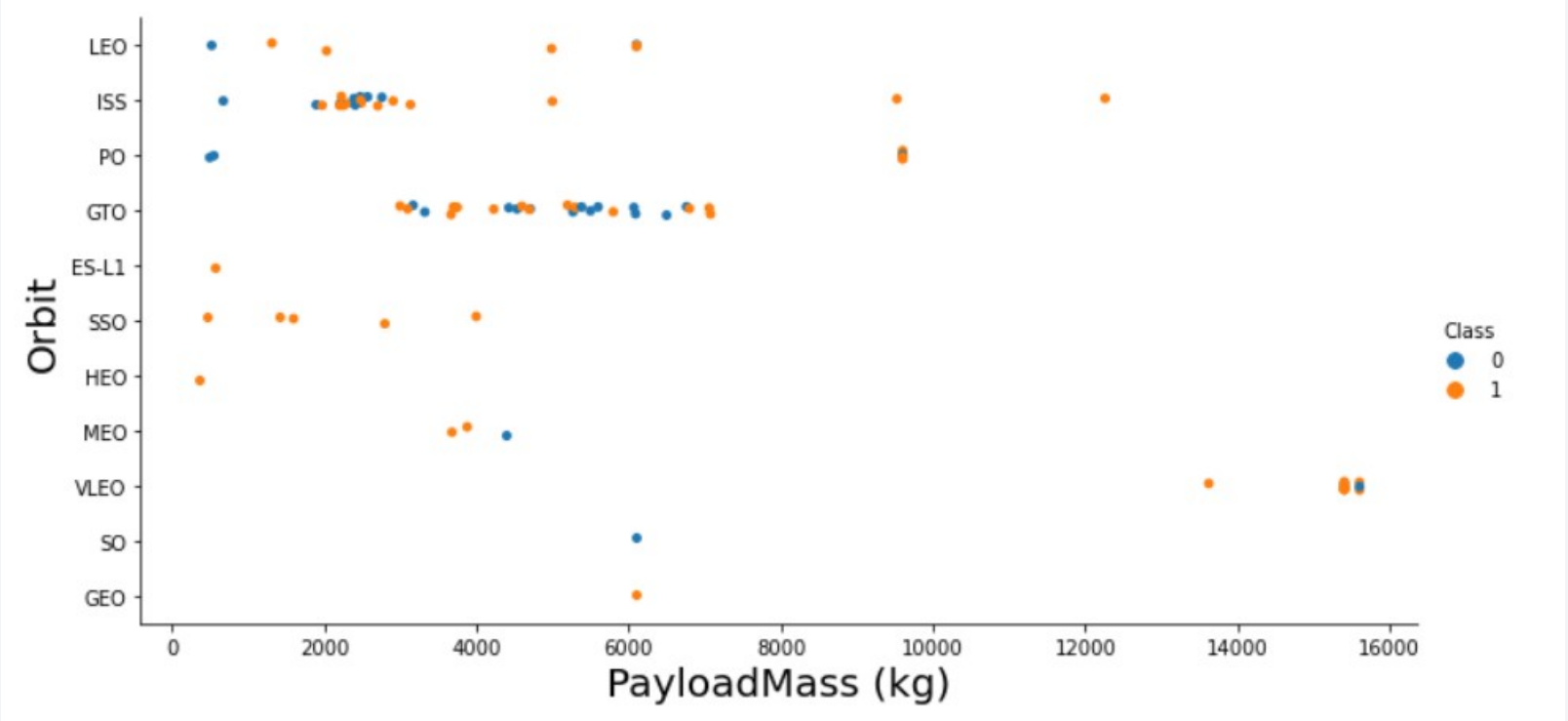


Flight Number vs. Orbit Type



- The Success appears of LEO orbit related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.

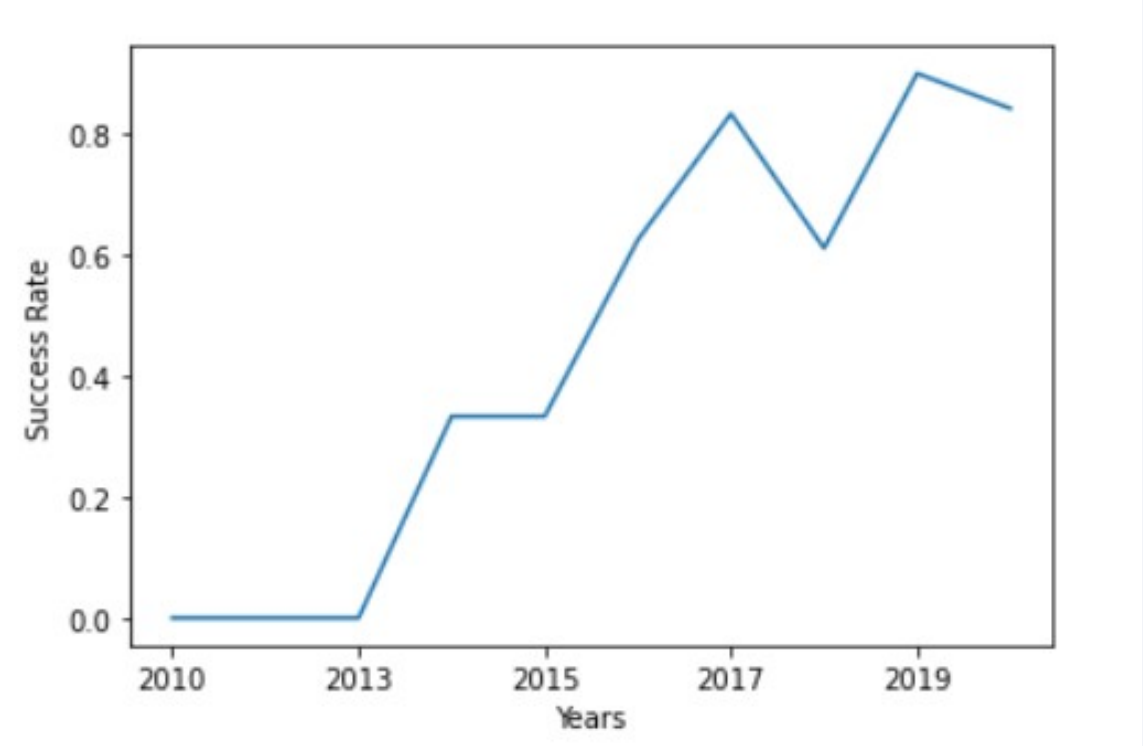
Payload vs. Orbit Type



- 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 sucess rate since 2013 kept increasing till 2020



All Launch Site Names

- %sql SELECT DISTINCT Launch_Site FROM SPACEX

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

Launch Site Names Begin with 'CCA'

- %sql SELECT * FROM SPACEX WHERE Launch_Site LIKE 'CCA%' LIMIT 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

Total Payload Mass

- %sql SELECT SUM(payload_mass__kg_) FROM SPACEX WHERE CUSTOMER='NASA (CRS)'

45596

Average Payload Mass by F9 v1.1

- %sql SELECT AVG(payload_mass__kg_) FROM SPACEX WHERE BOOSTER_VERSION='F9 v1.1'

```
2928
```

First Successful Ground Landing Date

- %sql SELECT MIN(DATE) FROM SPACEX WHERE landing__outcome='Success (ground pad)'

2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- %sql SELECT Booster_Version, PAYLOAD_MASS__KG_ FROM SPACEX WHERE landing__outcome='Success (drone ship)' and PAYLOAD_MASS__KG_ BETWEEN 4000 and 6000

booster_version	payload_mass__kg_
F9 FT B1022	4696
F9 FT B1026	4600
F9 FT B1021.2	5300
F9 FT B1031.2	5200

Total Number of Successful and Failure Mission Outcomes

- %sql SELECT mission_outcome , COUNT(*) as total_number FROM SPACEX GROUP BY mission_outcome

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

Boosters Carried Maximum Payload

- %sql SELECT BOOSTER_VERSION , PAYLOAD_MASS__KG_ FROM SPACEX WHERE PAYLOAD_MASS__KG_=(SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEX)

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

2015 Launch Records

- %sql SELECT DATE, BOOSTER_VERSION , Launch_Site, LANDING__OUTCOME FROM SPACEX WHERE (Date LIKE '%2015%') and Landing__Outcome='Failure (drone ship)'

DATE	booster_version	launch_site	landing__outcome
2015-01-10	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
2015-04-14	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

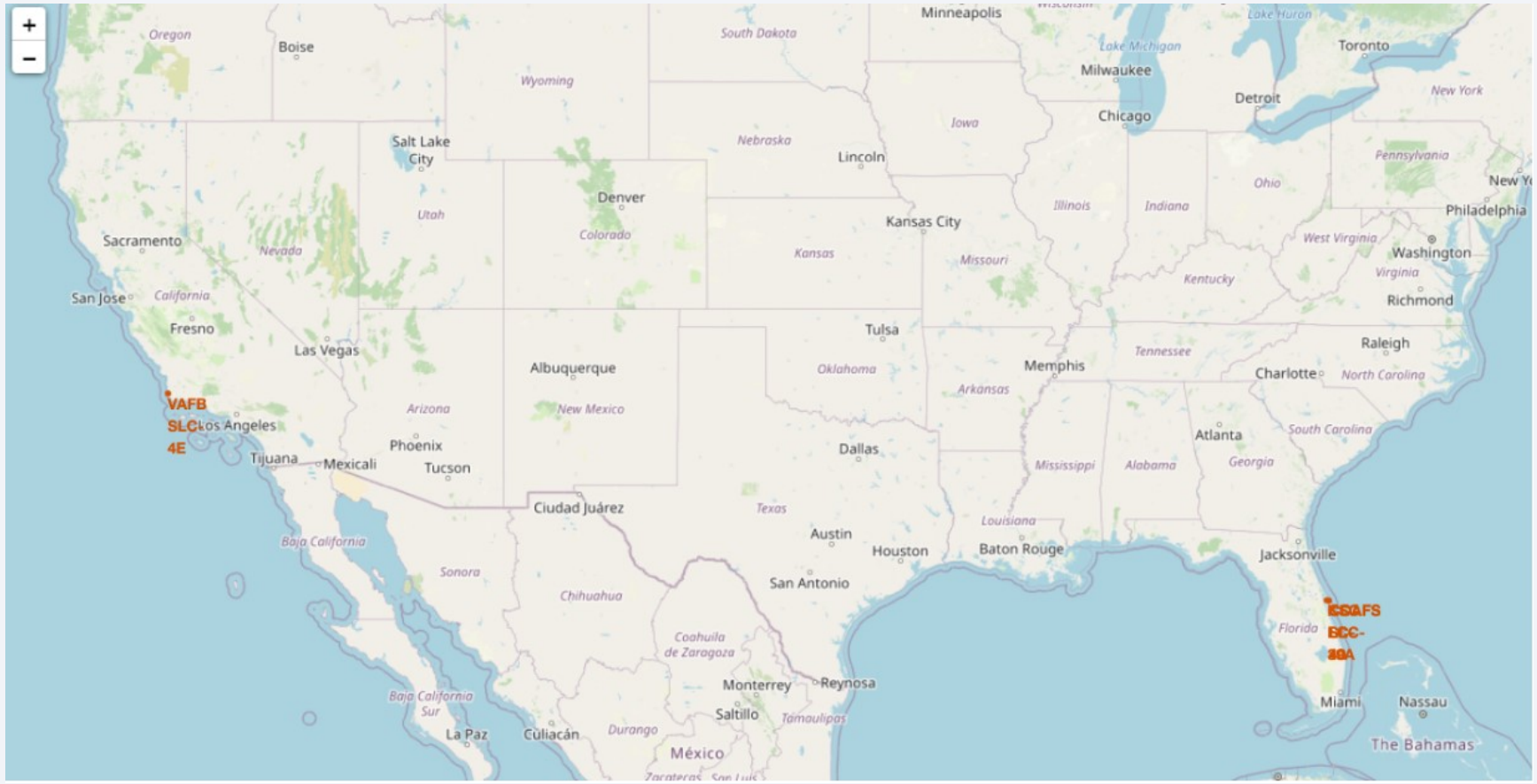
Rank Landing Outcomes Between 2010-06-04 and 2017-03-20

- %sql SELECT LANDING__OUTCOME ,COUNT(*) AS COUNT_LAUNCHES FROM SPACEX WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY LANDING__OUTCOME ORDER BY COUNT(*) DESC

landing__outcome	count_launches
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

Mark all launch sites on a map

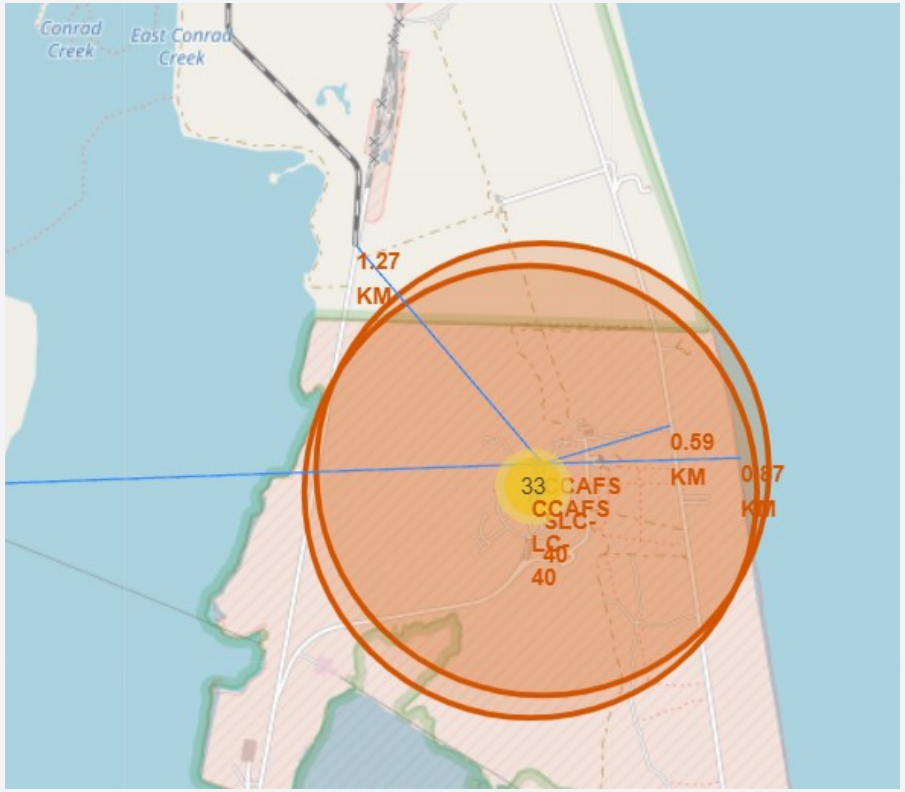


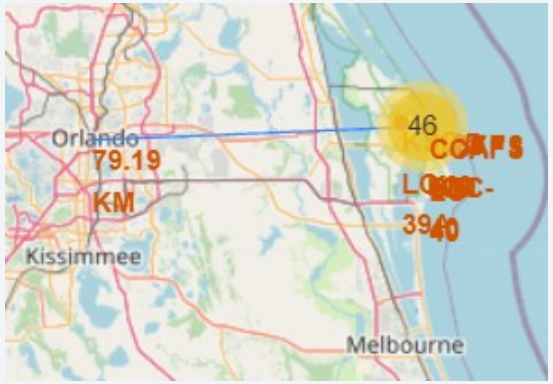
- All launch sites are very close proximity to the coast

Mark the success/failed launches for each site on the map

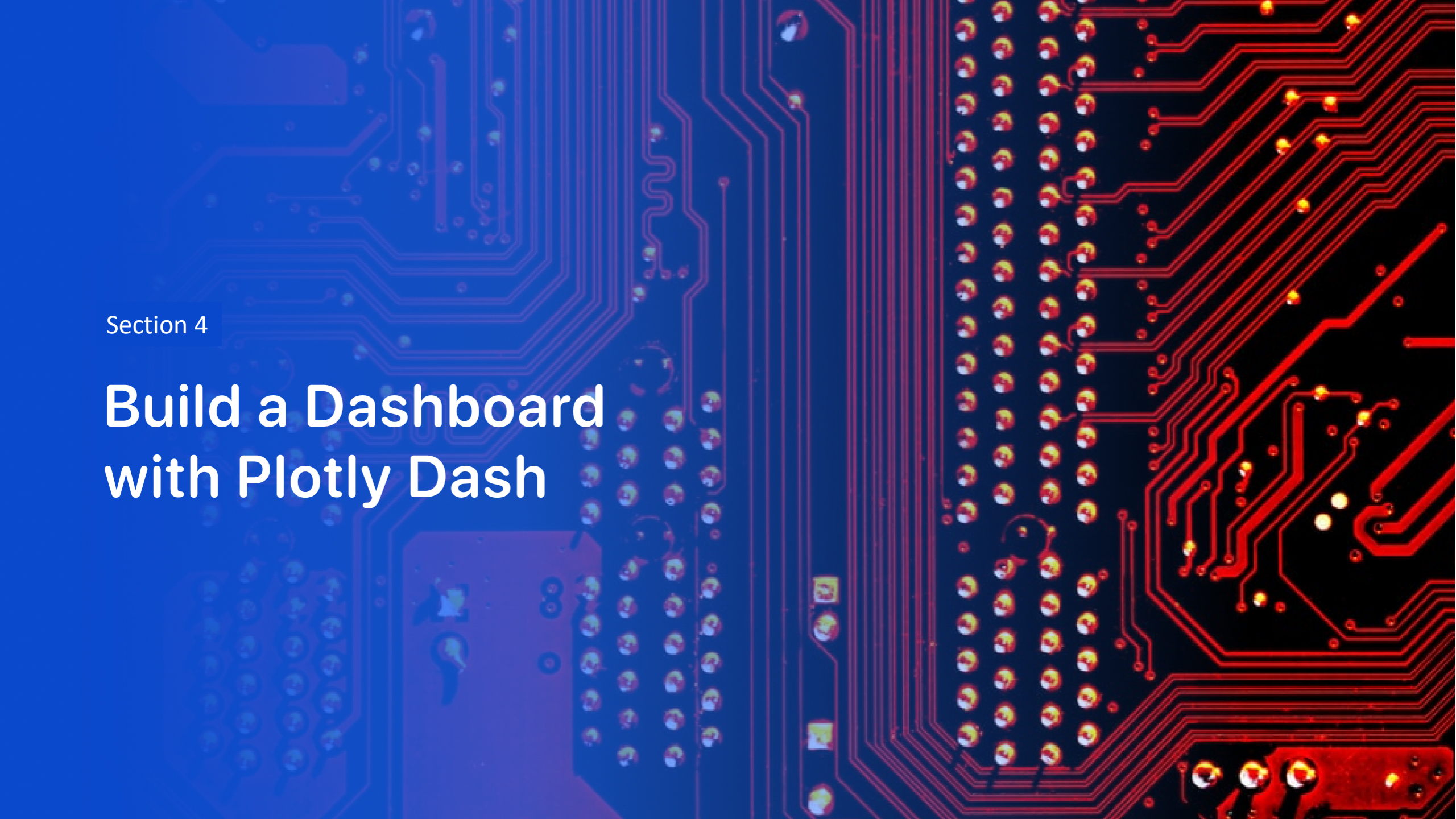
- We can be able to easily identify which launch sites have relatively high success rates.

Calculate the distances between a launch site to its proximities

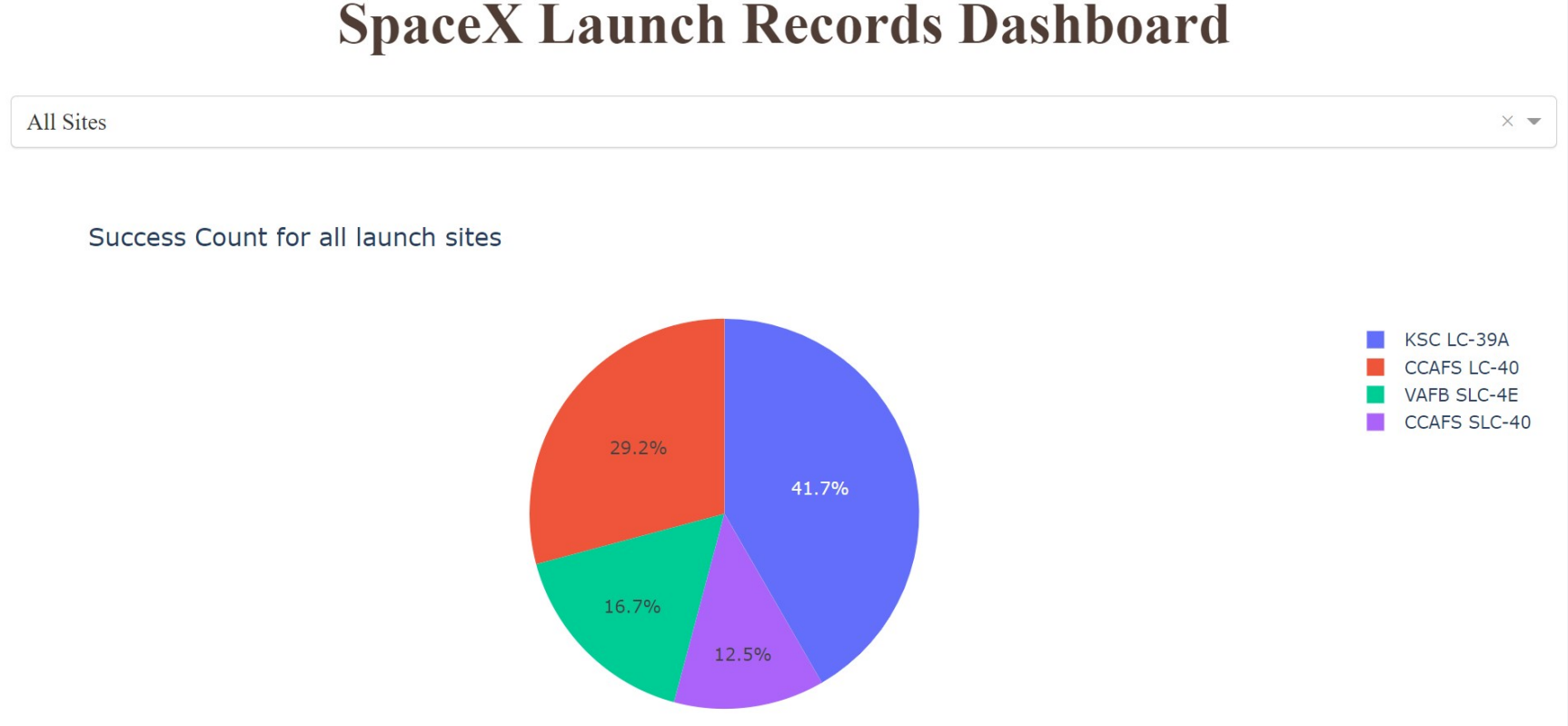




- Launch sites are in close proximity to railways, railway and coastline
- Launch sites keep certain distance away from cities

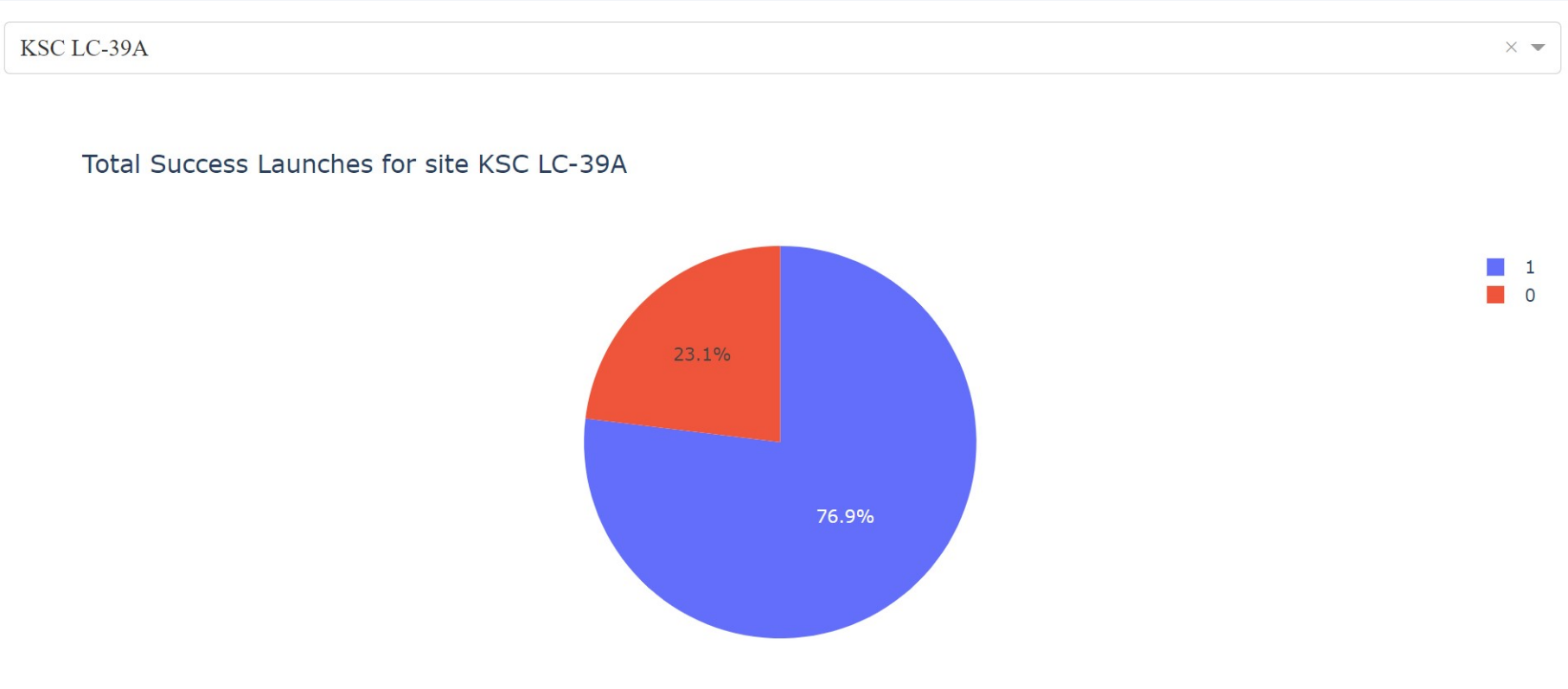

Section 4
Build a Dashboard with Plotly Dash

Launch success count for all sites



- KSC LC-39A has the largest successful launches from all the sites

Which site has the highest launch success rate



- KSC LC-39A has the higest successful rate (76.9%)

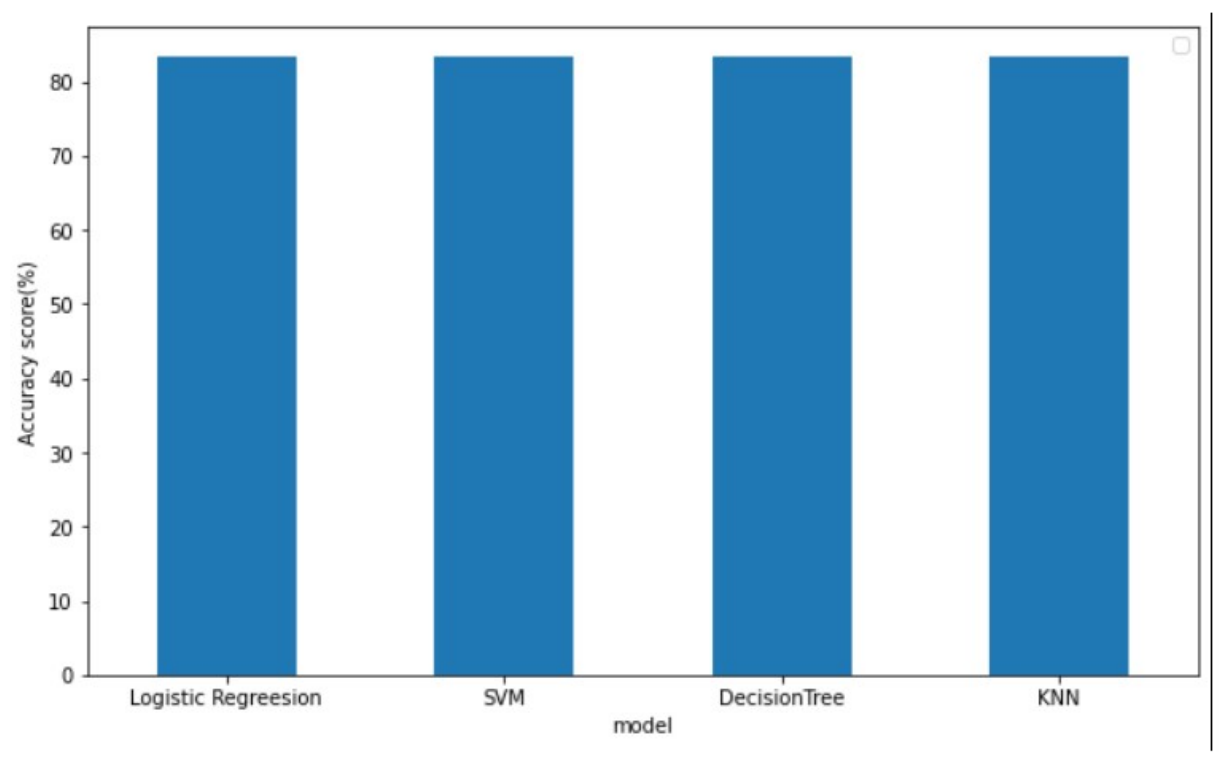
payload range(s) vs launch outcome

- We can see the booster version called FT has the highest success rate in range 2000~4000kg

Section 5

Predictive Analysis (Classification)

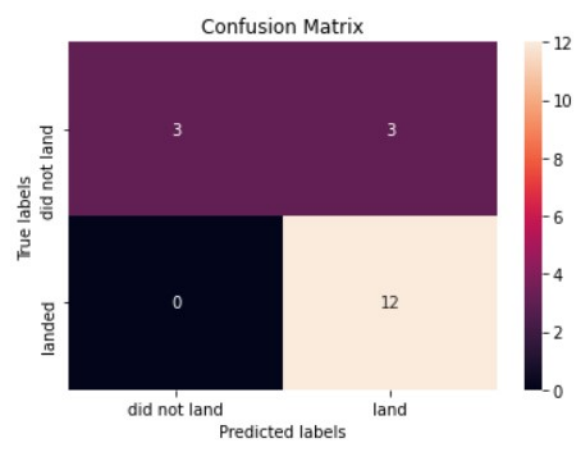
Classification Accuracy



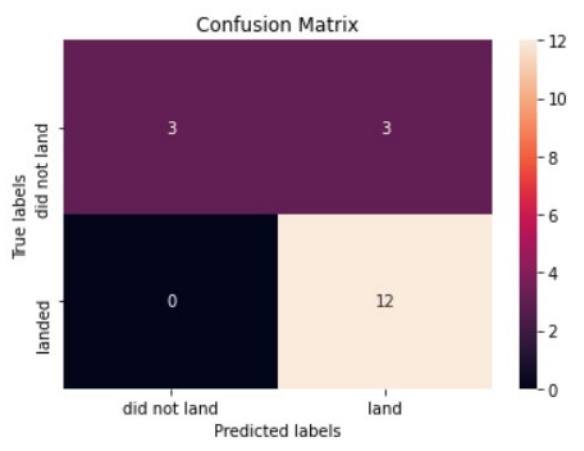
- All these algorithms give the same result(83.3333%)

Confusion Matrix

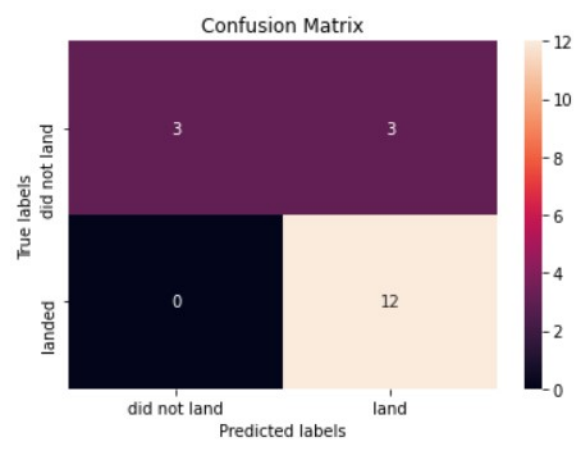
- Logisic Regression



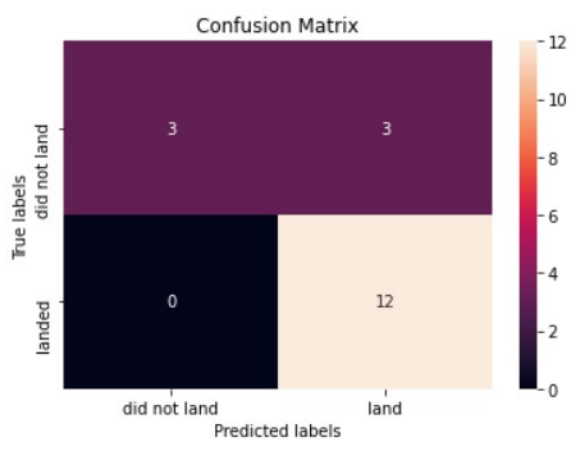
SVM



- Decision Tree



KNN



Conclusions

- The orbit type of ES-L1, GEO, HEO, SSO are among the highest success rate
- The sucess rate increases with the years
- KSC LC-39A has the higest successful rate
- All these algorithms give the same result (some false positive)

Appendix

- GitHub repository url:
- https://github.com/Joe-Chien/Pythons-Basics-for-Data-Science-Project

- SpaceX data
- Wikipedia

Thank you!