IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Andreas Formella
07-Oct-2024

Outline

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

Executive Summary

- Summary of methodologies
- Summary of all results

Introduction

- Project background and context
- Problems you want to find answers

Section 1

Methodology

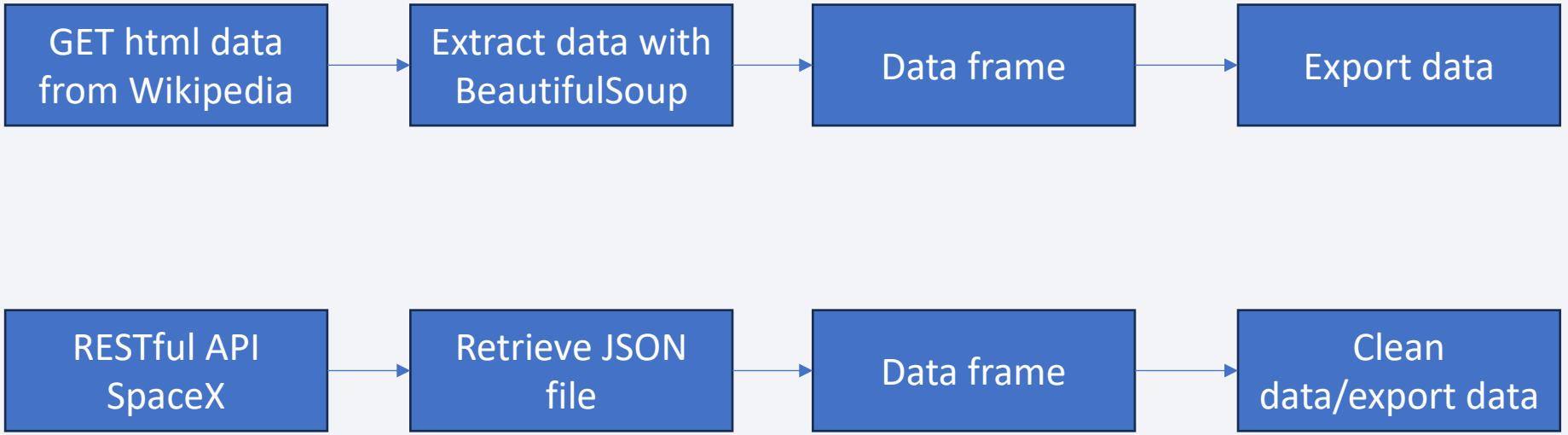
Methodology

Executive Summary

- Data collection methodology:
  - Web scrapping (Wikipedia)
  - RESTful API SpaceX
- Perform data wrangling
  - Dropping unnecessary columns
  - Adding means to cells without value
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection

- Describe how data sets were collected.
- You need to present your data collection process use key phrases and flowcharts



Data Collection – SpaceX API

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

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

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

[50]:
```
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}
```

Then, we need to create a Pandas data frame from the dictionary launch_dict

[51]:
```
# Create a data from launch_dict
data = pd.DataFrame(launch_dict)
```

Data Collection - Scraping

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

```
# Use BeautifulSoup() to create a Bea
soup = BeautifulSoup(data)
```

```
# Use the find_all function in the BeautifulS
# Assign the result to a list called `html_ta
html_tables = soup.find_all('table')
```

```
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)
```

```
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
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']=[]
```

Data Wrangling

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

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

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

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

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

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

EDA with Data Visualization

- Scatter Plots
  - Flight number vs payload mass
  - Flight number vs launch site
  - Payload vs launch site
  - Orbit vs flight number
  - Payload vs orbit type
  - Orbit vs payload mass
- Bar Graph
  - Success rate vs orbit
- Line Graph
  - Success rate vs year

EDA with SQL

- SQL queries for
  - Launch site
  - Total payload
  - Average payload
  - First time landing outcome was good
  - All boosters used for payloads of 4000 till 6000
  - Total number of starts / success / failure

Build an Interactive Map with Folium

- Red circle at launch site
- Names of launch site
- Markers to show success and failure landings

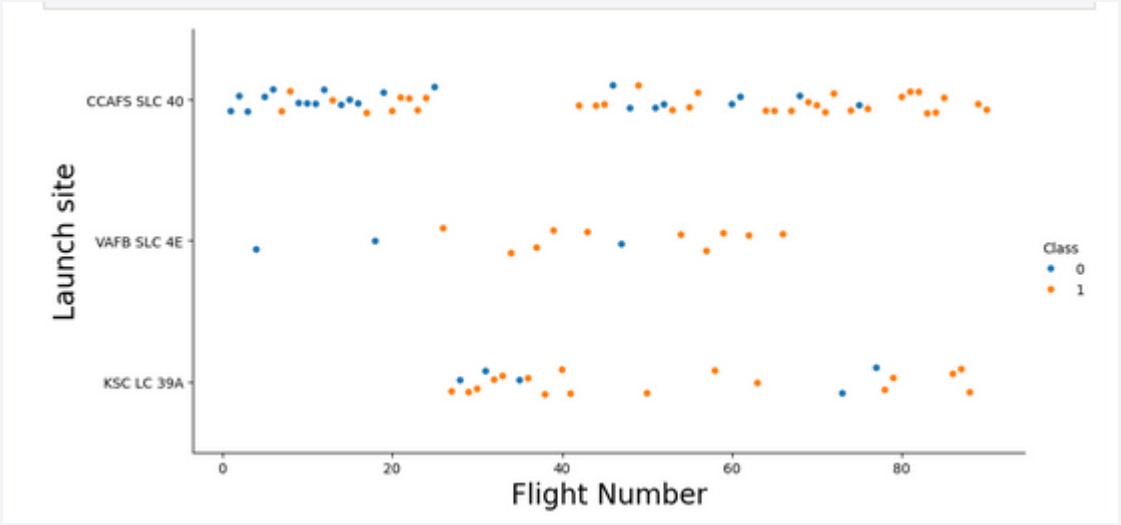
Build a Dashboard with Plotly Dash

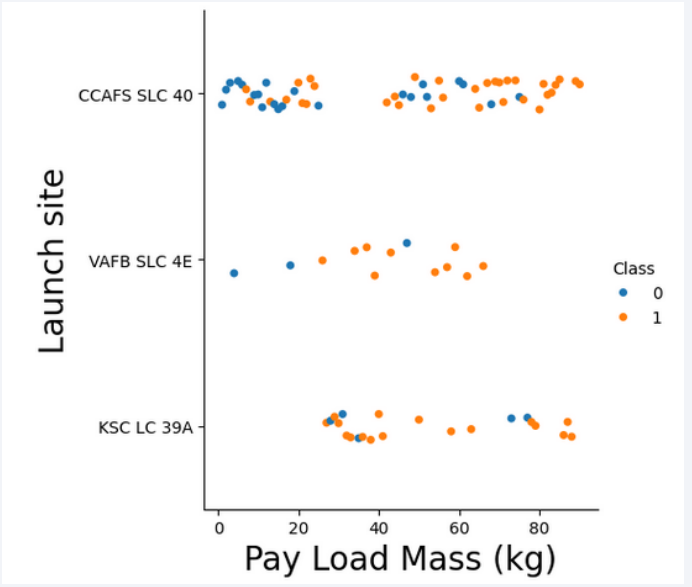
- Dropdown menu shows launch site
- Pie chart shows success and failure
- Range slider shows payload
- Scatter plot shows relation between variables

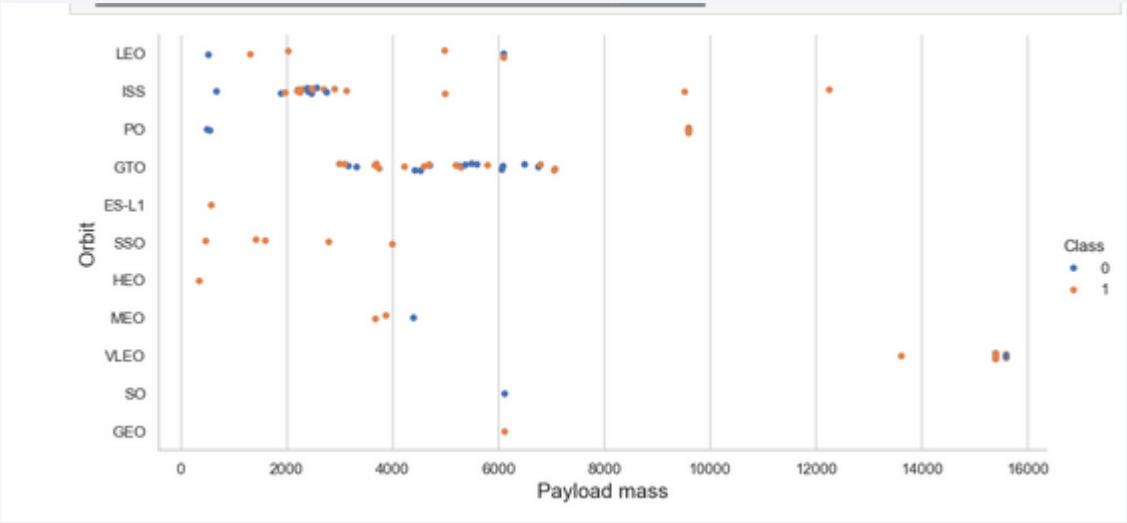
Predictive Analysis (Classification)

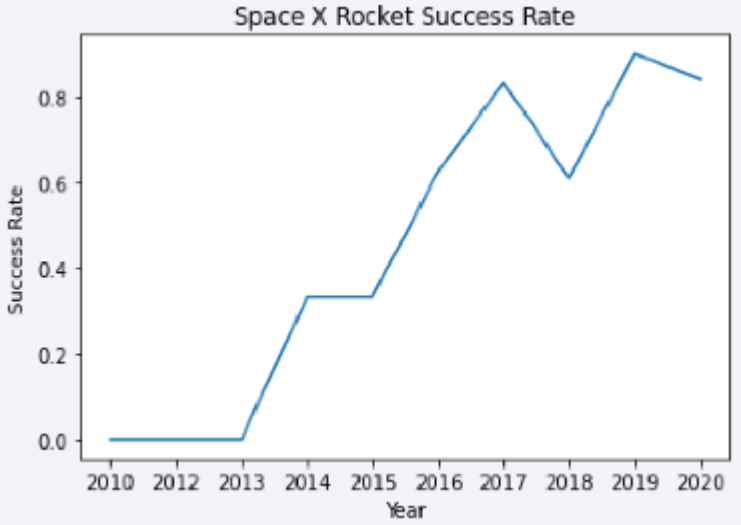
- Prepare Data
- Prepare Models
- Evaluation of Models
- Compare different models

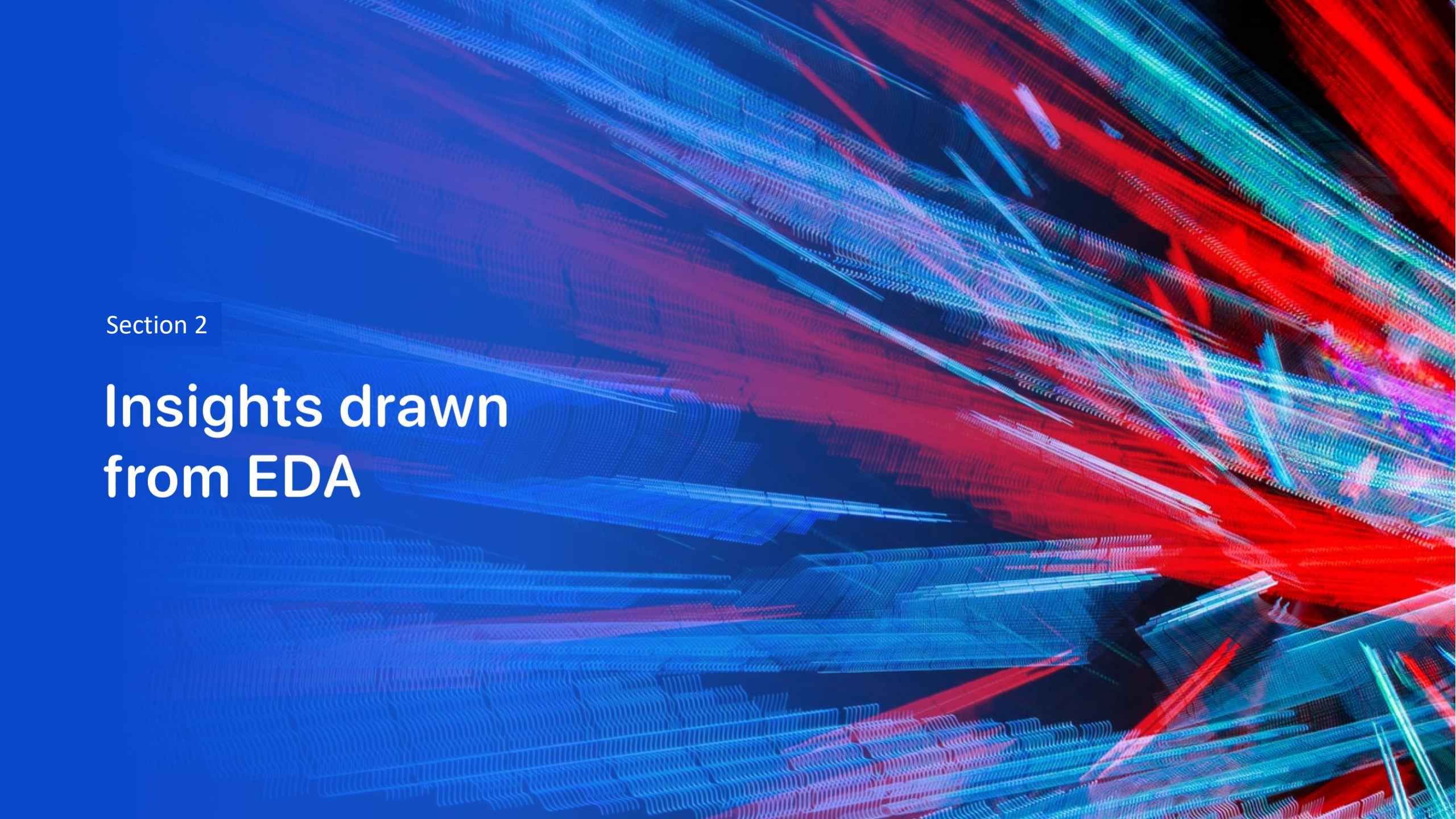
Results (I)


CCAFS SLC 40
VAFB SLC 4E
KSC LC 39A
Launch site
0
20
40
60
80
Flight Number
Class
0
1

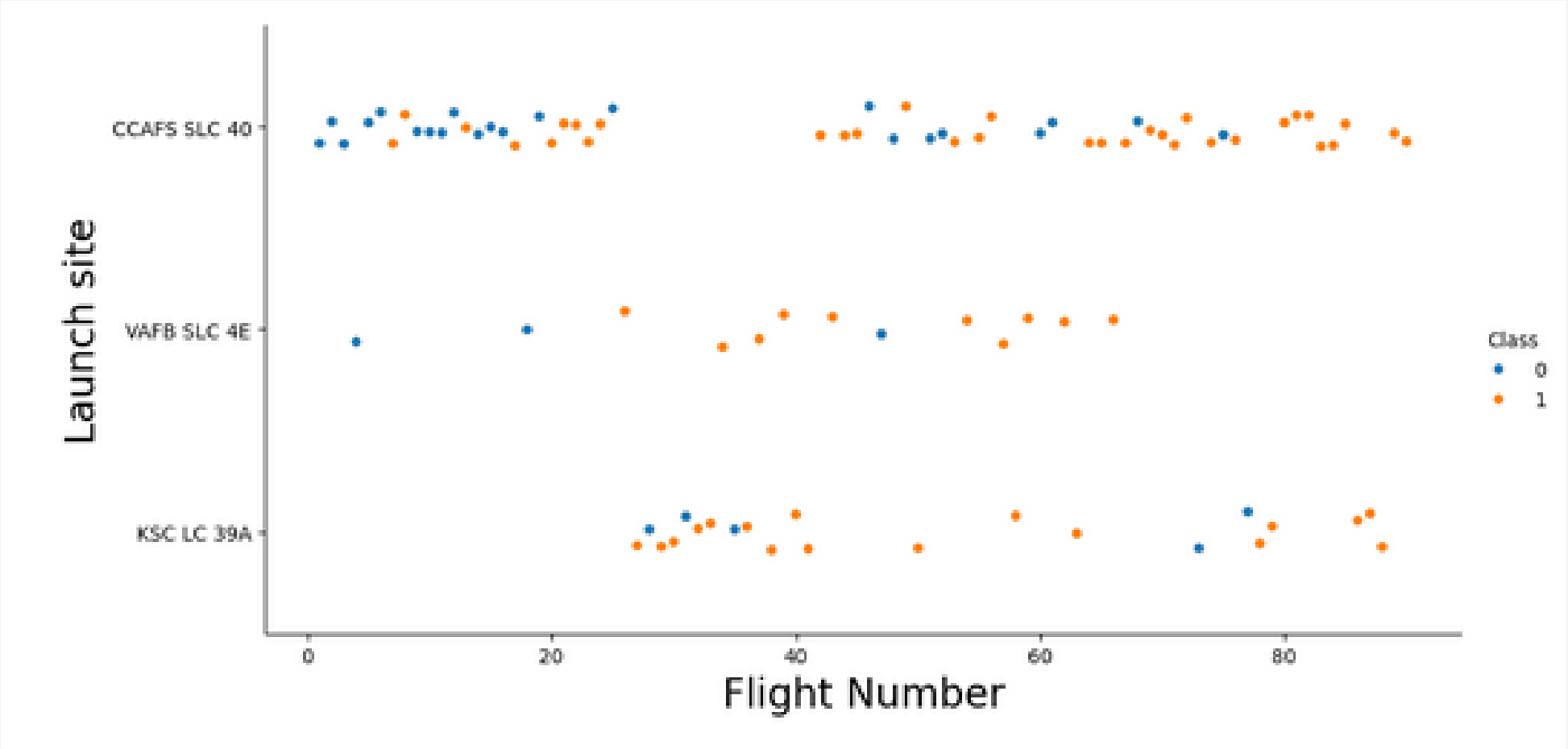

CCAFS SLC 40
VAFB SLC 4E
KSC LC 39A
Launch site
0
20
40
60
80
Pay Load Mass (kg)
Class
0
1


LEO
ISS
PO
GTO
ES-L1
SSO
HEO
MEO
VLEO
SO
GEO
Orbit
0
2000
4000
6000
8000
10000
12000
14000
16000
Payload mass
Class
0
1

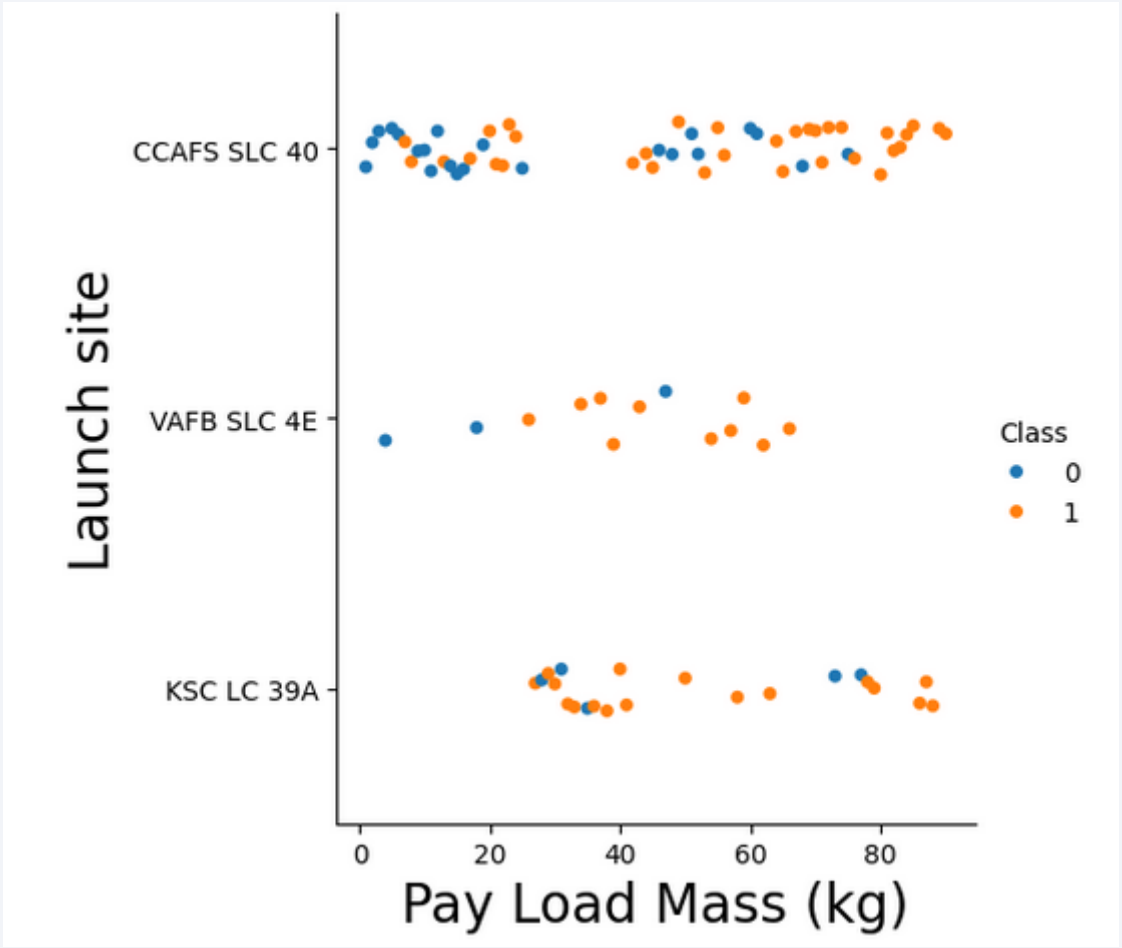

Space X Rocket Success Rate
Success Rate
0.0
0.2
0.4
0.6
0.8
2010
2012
2013
2014
2015
2016
2017
2018
2019
2020
Year


Section 2
Insights drawn from EDA

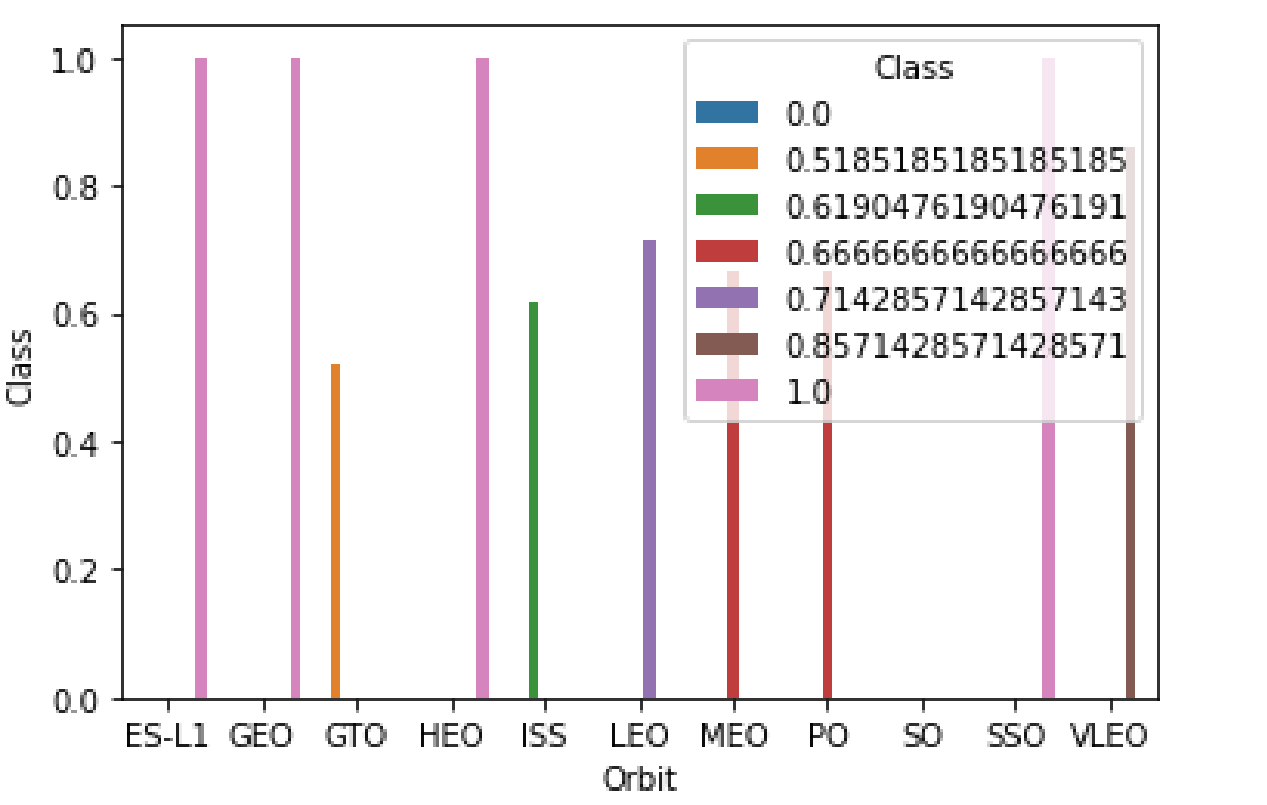
Flight Number vs. Launch Site



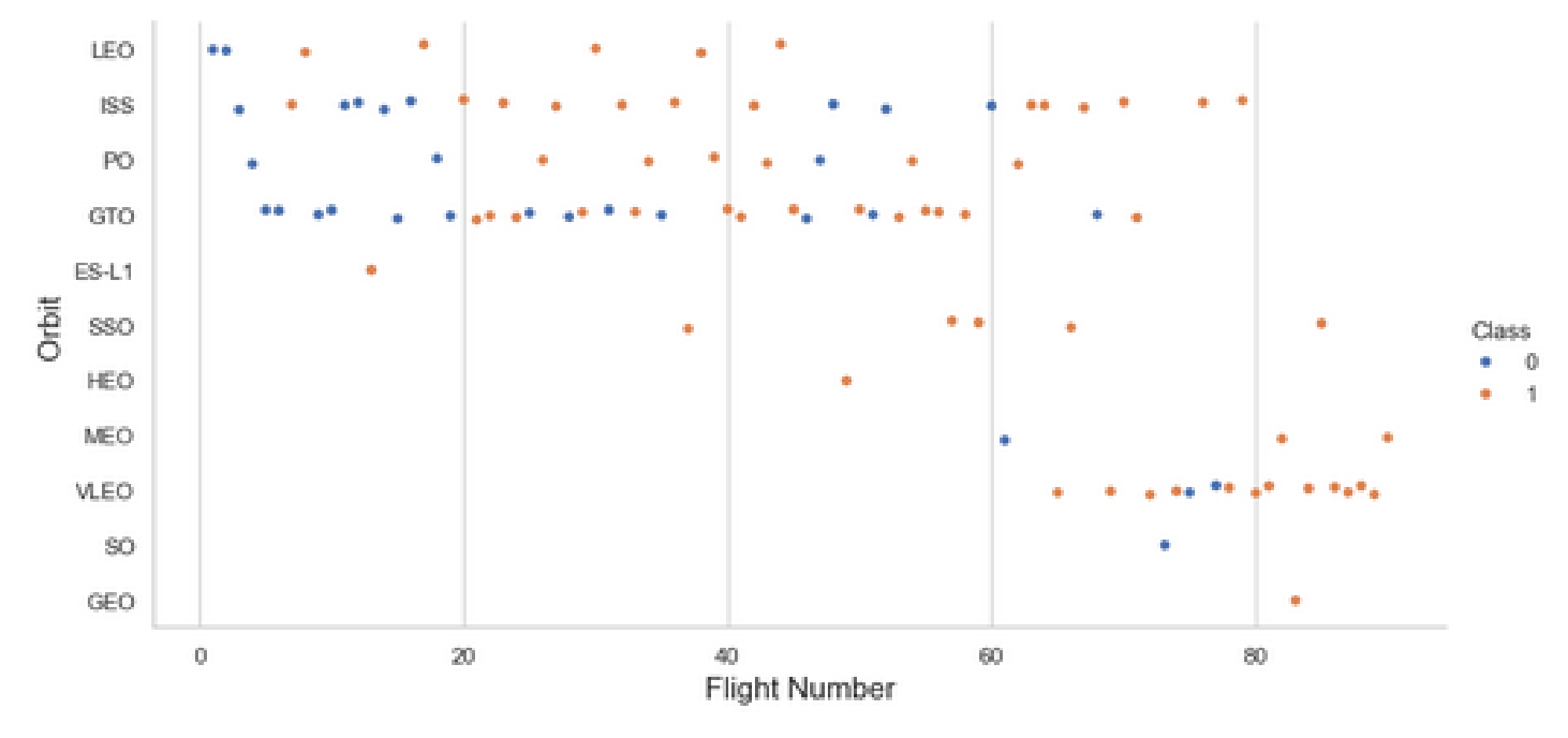
Payload vs. Launch Site



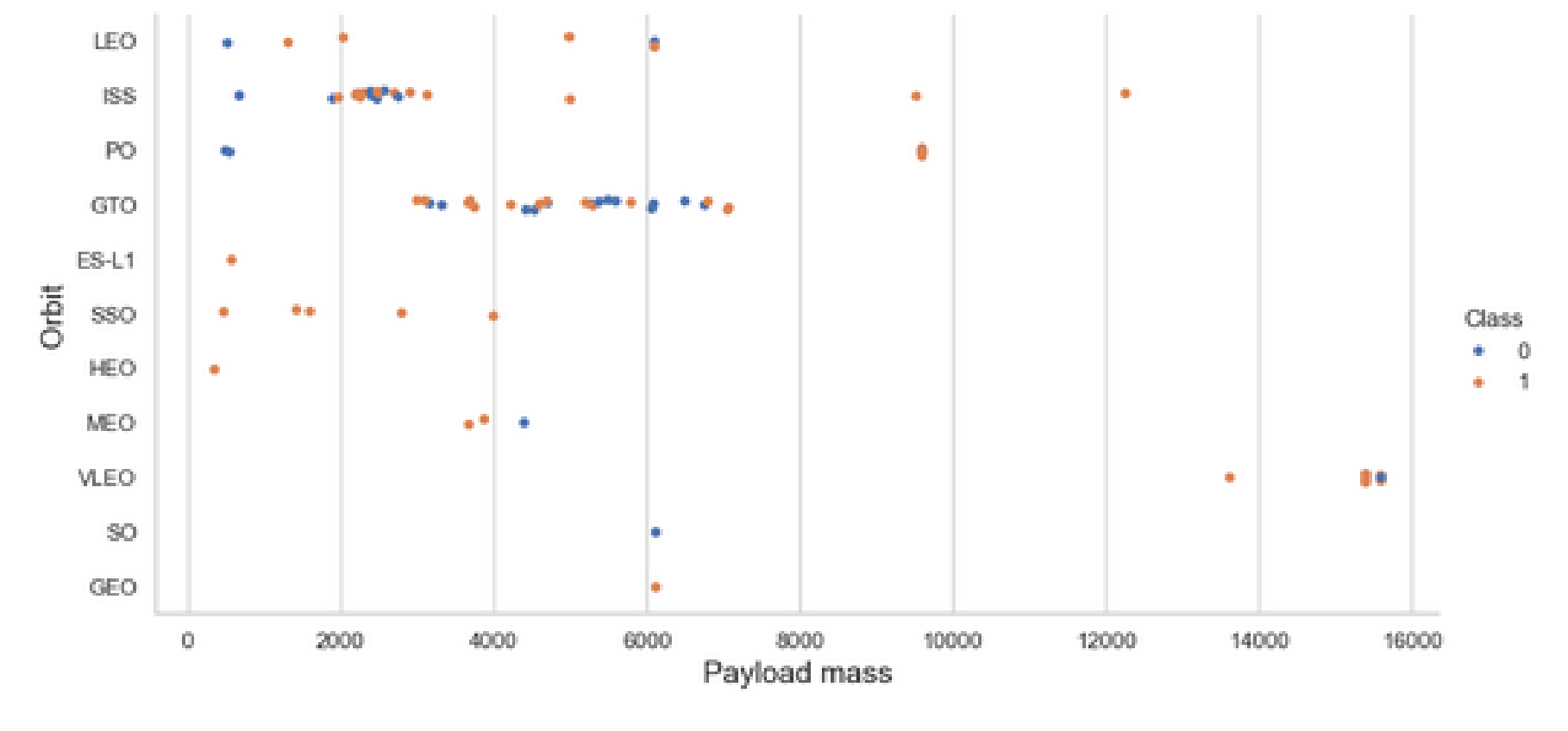
Success Rate vs. Orbit Type



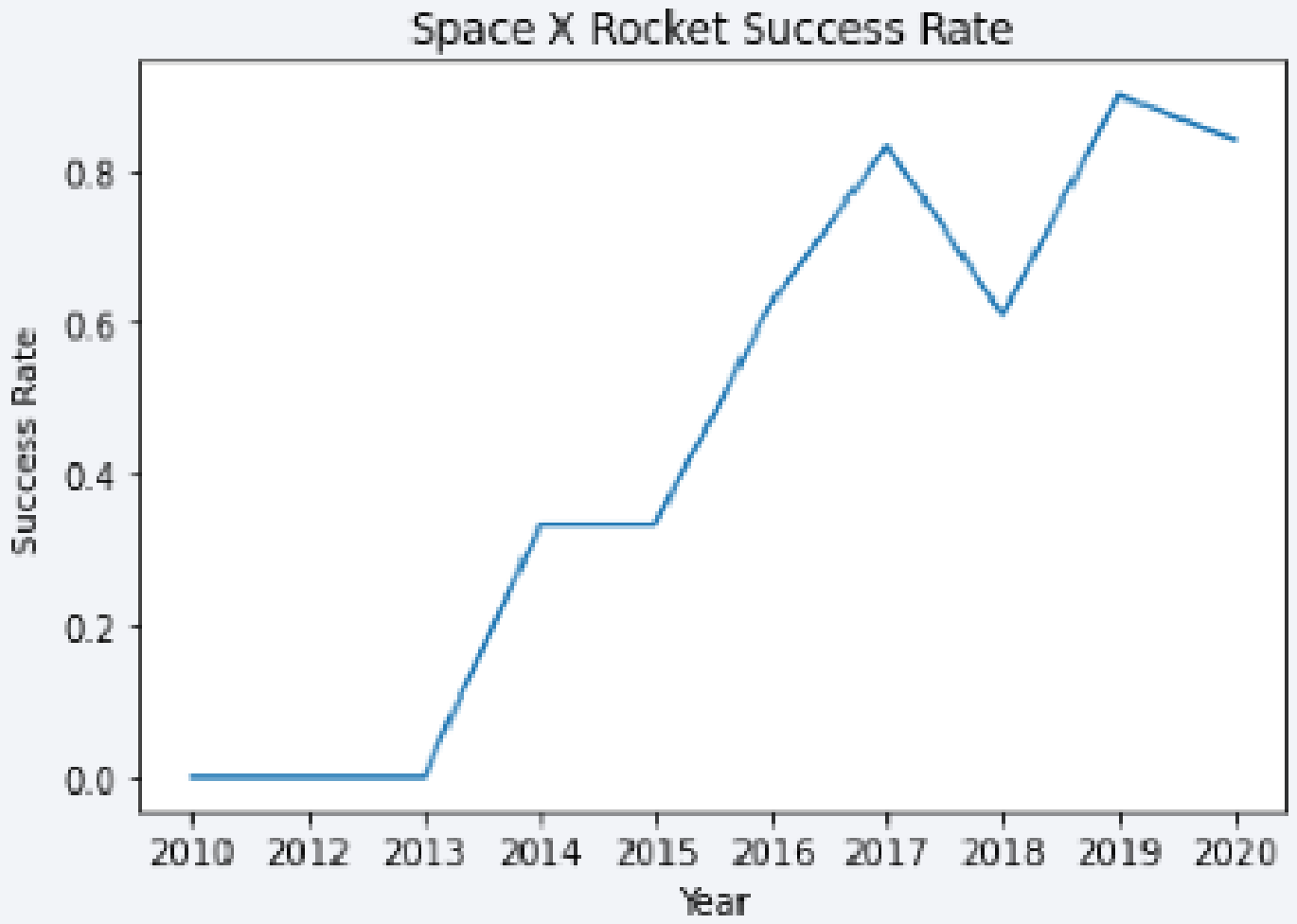
Flight Number vs. Orbit Type



Payload vs. Orbit Type



Launch Success Yearly Trend



All Launch Site Names

```
[12]: %sql select distinct (LAUNCH_SITE) from SPACEXTBL;
```

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

[12]:

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

Launch Site Names Begin with 'CCA'

Display 5 records where launch sites begin with the string 'CCA'

[26]:
```
%sql SELECT * from SPACEXTBL where (LAUNCH_SITE) LIKE 'CCA%' LIMIT 5;
```

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

[26]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	

Total Payload Mass

```
[15]: %sql select sum(PAYLOAD_MASS__KG_) as payloadmass from SPACEXTBL;
```

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

[15]:

payloadmass
619967

Average Payload Mass by F9 v1.1

Task 4

Display average payload mass carried by booster version F9 v1.1

```
[6]: %sql select avg(PAYLOAD_MASS__KG_) as payloadmass from SPACEXTBL;
```

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

[6]:

payloadmass
6138.287128712871

First Successful Ground Landing Date

Hint:Use min function

```
[27]: %sql select min(DATE) from SPACEXTBL where LANDING_OUTCOME='Success (groun
```

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

[27]:

min(DATE)
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

payload mass greater than 4000 but less than 6000

```
[9]: %sql select BOOSTER_VERSION from SPACEXTBL where LANDING_OUTCOME='Success
```

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

[9]:

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

Total Number of Successful and Failure Mission Outcomes

```
9]: %sql select Mission_outcome, count(MISSION_OUTCOME) as missionoutcomes fro
```

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

9]:

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

Boosters Carried Maximum Payload

```
0]: %sql select BOOSTER_VERSION as boosterversion from SPACEXTBL where PAYLOAD
 * sqlite:///my_data1.db
Done.
```

0]:

boosterversion
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

[32]:
```
%sql SELECT substr(Date, 6,2),MISSION_OUTCOME,BOOSTER_VERSION,LAUNCH_SITE
```

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

[32]:

substr(Date, 6,2)	Mission_Outcome	Booster_Version	Launch_Site
01	Success	F9 v1.1 B1012	CCAFS LC-40
02	Success	F9 v1.1 B1013	CCAFS LC-40
03	Success	F9 v1.1 B1014	CCAFS LC-40
04	Success	F9 v1.1 B1015	CCAFS LC-40
04	Success	F9 v1.1 B1016	CCAFS LC-40
06	Failure (in flight)	F9 v1.1 B1018	CCAFS LC-40
12	Success	F9 FT B1019	CCAFS LC-40

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

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

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

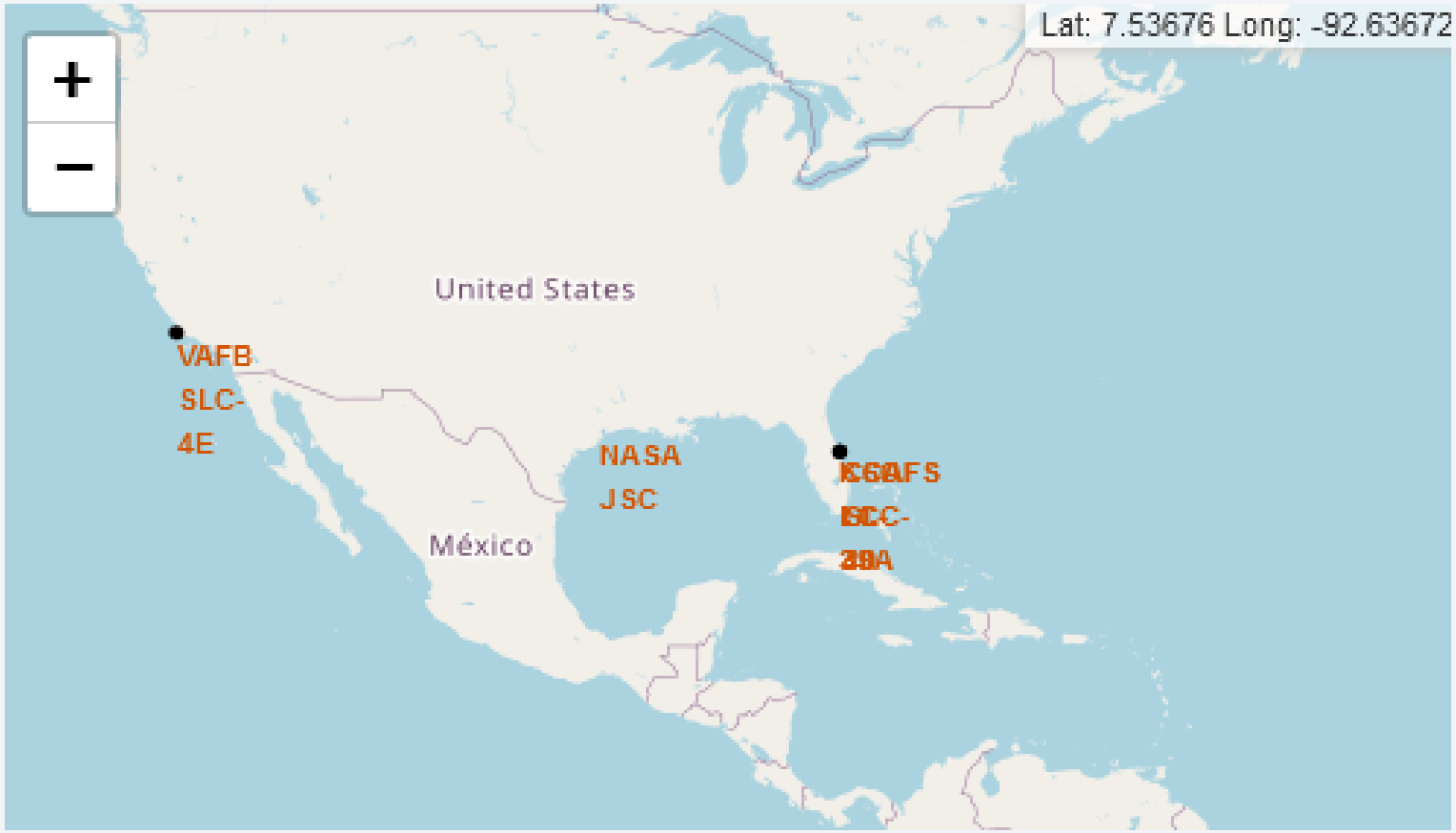
Section 3

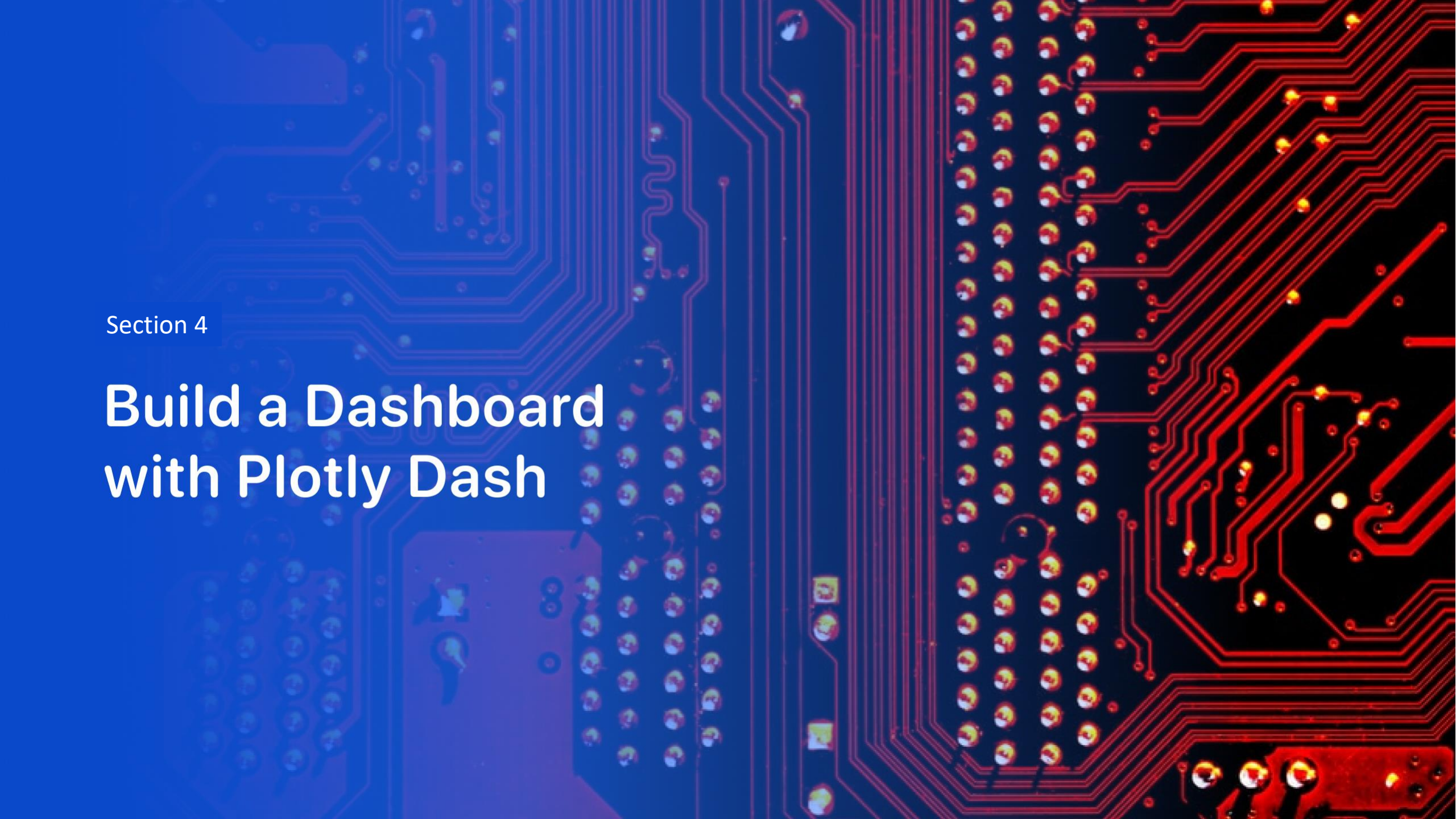
Launch Sites Proximities Analysis

<Folium Map Screenshot 1>

<Folium Map Screenshot 2>

<Folium Map Screenshot 3>




Section 4
Build a Dashboard with Plotly Dash

<Dashboard Screenshot 1>

- Replace <Dashboard screenshot 1> title with an appropriate title
- Show the screenshot of launch success count for all sites, in a piechart
- Explain the important elements and findings on the screenshot

<Dashboard Screenshot 2>

- Replace <Dashboard screenshot 2> title with an appropriate title
- Show the screenshot of the piechart for the launch site with highest launch success ratio
- Explain the important elements and findings on the screenshot

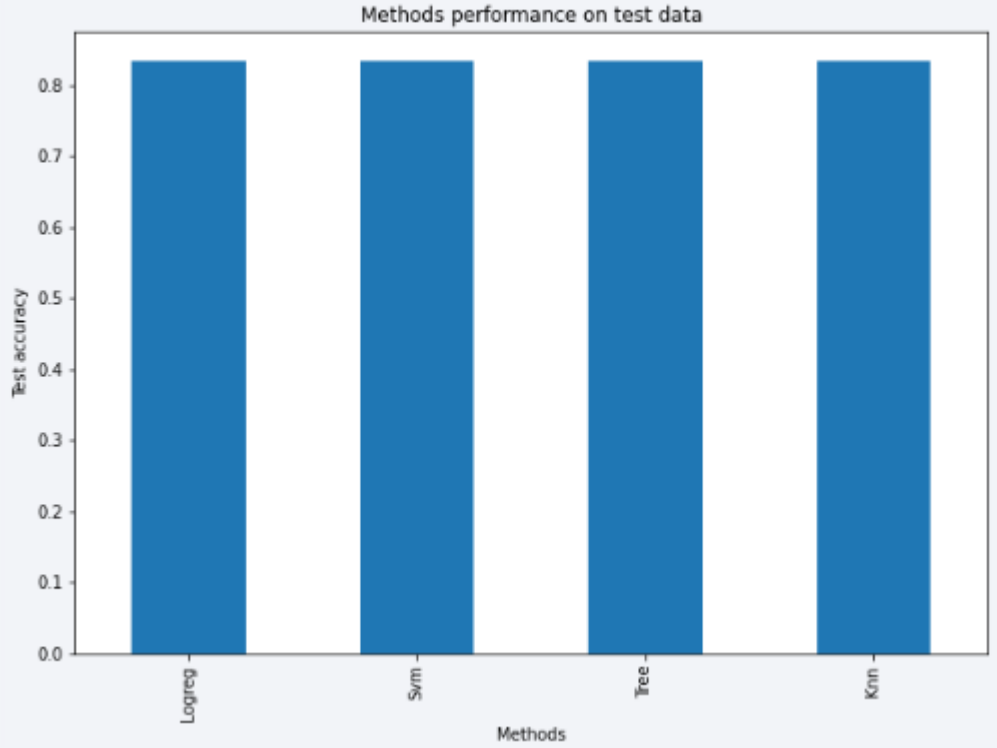
<Dashboard Screenshot 3>

- Replace <Dashboard screenshot 3> title with an appropriate title
- Show screenshots of Payload vs. Launch Outcome scatter plot for all sites, with different payload selected in the range slider
- Explain the important elements and findings on the screenshot, such as which payload range or booster version have the largest success rate, etc.

Section 5

Predictive Analysis (Classification)

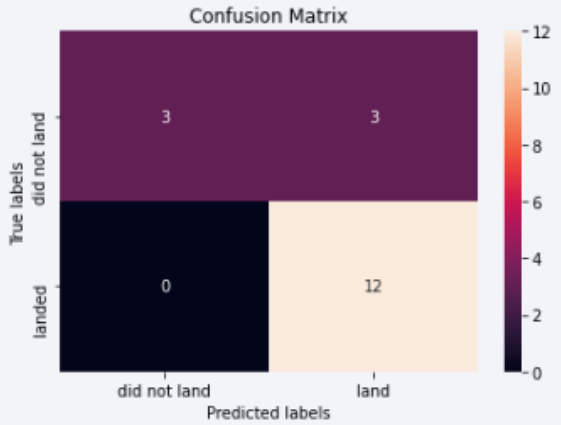
Classification Accuracy



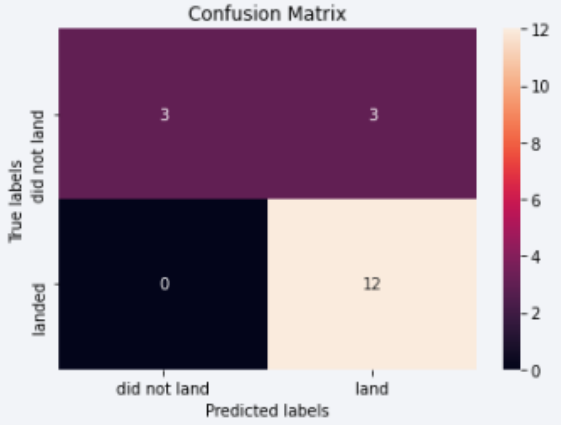
Confusion Matrix

Logistic regression

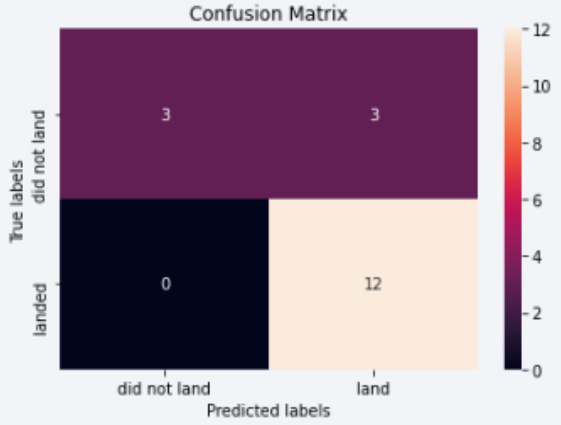
Decision Tree



kNN



SVM



Conclusions

- Success rate is dependent from launch site, orbit, payload and number of previous launches
- Over time, the success rate is improving significantly

Thank you!