IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Name: Wael Chmaisani
Date: 11-March-2023

Outline

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

Executive Summary

Summary of methodologies

- Data collection
- Data wrangling
- EDA using SQL
- EDA using Pandas and Matplotlib
- Interactive visual analytics and dashboard
- Predictive analysis (classification)

Summary of all results

- Determination of the features that affect on Space X Falcon 9 landing outcomes
- Identification the best classification model that accurately predict landing outcome of Falcon 9

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

- We will predict if the Falcon 9 first stage will land successfully
- We will predict the cost of a launch

Section 1

Methodology

Methodology

Executive Summary

- **Data collection methodology**
  - Data Collection – SpaceX API
  - Web scraping Falcon 9 and Falcon Heavy Launches Records from Wikipedia
- **Perform data wrangling**
  - Dealing with Missing Values
  - Determine what would be the label for training supervised models
- **Perform exploratory data analysis (EDA) using visualization and SQL**
- **Perform interactive visual analytics using Folium and Plotly Dash**
- **Perform predictive analysis using classification model**
  - Standardize the data and split into training data and test data
  - Build LR, SVM, DT and KNN models using GridSeachCV method and evaluate them by using the method score

Data Collection

Request and parse the rocket launch data from **SpaceX API** using the GET request

Request the Falcon9 Launch Wiki HTML page using **web scraping**

Data collected
Ready as a data frame

Data Collection – SpaceX API

1 Request the JSON of the SpaceX launch data from API using the GET request

```
static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_call_spacex_api.json'
response = requests.get(static_json_url).json()
```

2 Use json normalize method to convert the JSON result into a data frame

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

	static_fire_date_utc	static_fire_date_unix	tbd	net	window	rocket	success	details	crew	ships	capsules	payloads	launchpad
0	2006-03-17T00:00:00.000Z	1.142554e+09	False	False	0.0	5e9d0d95eda69955f709d1eb	False	Engine failure at 33 seconds and loss of vehicle	[]	[]	[]	[5eb0e4b5b6c3bb0006eeb1e1]	5e9e4502f5090995de566f86

3 Combine the columns into a dictionary and create a Pandas **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}

df=pd.DataFrame(launch_dict)
```

FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	Longitude	Latitude
1	2006-03-24	Falcon 1	20.0	LEO	Kwajalein Atoll	None None	1	False	False	False	None	NaN	0	Merlin1A	167.743129	9.047721
2	2007-03-21	Falcon 1	NaN	LEO	Kwajalein Atoll	None None	1	False	False	False	None	NaN	0	Merlin2A	167.743129	9.047721
4	2008-09-28	Falcon 1	165.0	LEO	Kwajalein Atoll	None None	1	False	False	False	None	NaN	0	Merlin2C	167.743129	9.047721

https://github.com/WaelChmaisani/Applied-Data-Science-Project/blob/main/jupyter-labs-Collecting.ipynb

Data Collection - Scraping

1 **Perform an HTTP GET method to request the Falcon9 Launch HTML page, as an HTTP response**

```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"
response = requests.get(static_url).text
```

2 **Use BeautifulSoup() to create a BeautifulSoup object from a response text content**

```
soup = BeautifulSoup(response,"html.parser")
```

3 **Find all tables on the wiki page using the find_all function in the BeautifulSoup object and print the third table where is our target table contains the actual launch records**

```
html_tables = soup.find_all("table")
first_launch_table = html_tables[2]
```

4 **Apply the provided extract_column_from_header() to extract column name**

```
column_names = []
cells = first_launch_table.find_all("th")
for i,cell in enumerate(cells):
    column_name = extract_column_from_header(cell)
    if (column_name != None) and (column_name != '') :
        column_names.append(column_name)
```

```
['Flight No.', 'Date and time ( )', 'Launch site', 'Payload', 'Payload mass', 'Orbit', 'Customer', 'Launch outcome']
```

5 **Create an empty dictionary with keys from the extracted column names and convert it into a Pandas data frame**

```
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'] = []
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)
```

	Flight No.	Launch site	Payload	Payload mass	Orbit	Customer	Launch outcome	Version Booster	Booster landing	Date	Time
0	121	CCSFS	SXM-8	7,000 kg	GTO	[Sirius XM]	Success\n	F9 B5	Success	6 June 2021	04:26

https://github.com/WaelChmaisani/Applied-Data-Science-Project/blob/main/jupyter-labs-webscraping.ipynb

Data Wrangling

Data frame

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	Longitude	Latitude
0	1	2010-06-04	Falcon 9	6104.959412	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0003	-80.577366	28.561857
1	2	2012-05-22	Falcon 9	525.000000	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0005	-80.577366	28.561857
2	3	2013-03-01	Falcon 9	677.000000	ISS	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0007	-80.577366	28.561857
3	4	2013-09-29	Falcon 9	500.000000	PO	VAFB SLC 4E	False Ocean	1	False	False	False	NaN	1.0	0	B1003	-120.610829	34.632093

Identify and calculate the percentage of the missing values in each attribute

```
df.isnull().sum()/df.shape[0]*100
```

```
FlightNumber       0.000000
Date               0.000000
BoosterVersion     0.000000
PayloadMass        0.000000
Orbit              0.000000
LaunchSite         0.000000
Outcome            0.000000
Flights            0.000000
GridFins           0.000000
Reused             0.000000
Legs               0.000000
LandingPad        28.888889
Block              0.000000
ReusedCount        0.000000
Serial             0.000000
Longitude          0.000000
Latitude           0.000000
dtype: float64
```

Apply value_counts() on columns

```
df["LaunchSite"].value_counts()
```

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

```
df["Orbit"].value_counts()
```

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

```
df["Outcome"].value_counts()
```

```
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: Outcome, dtype: int64
```

Create a landing outcome label from Outcome column

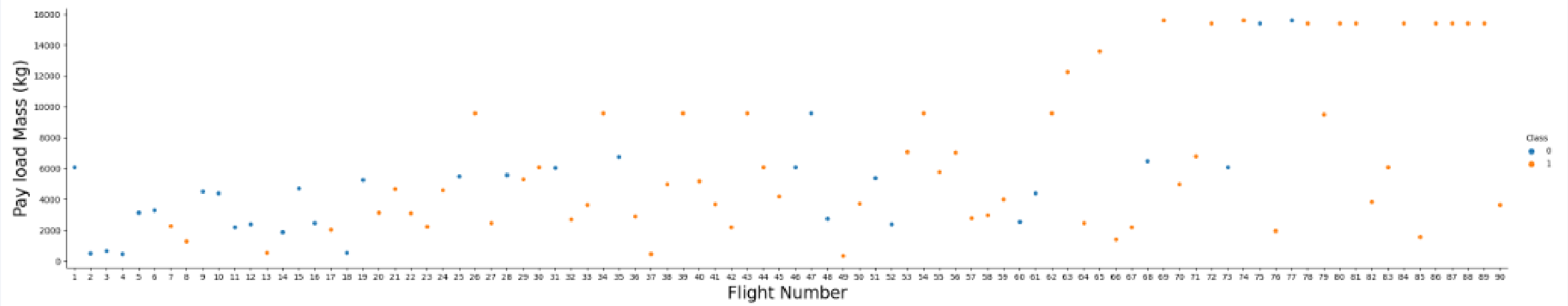
```
landing_outcomes = df["Outcome"].value_counts()
for i,outcome in enumerate(landing_outcomes.keys()):
    print(i,outcome)

bad_outcomes = set(landing_outcomes.keys()[[1,3,5,6,7]])

landing_class=[]
for outcome in df["Outcome"]:
    if outcome in bad_outcomes:
        landing_class.append(0)
    else:
        landing_class.append(1)

df['Class'] = landing_class
```

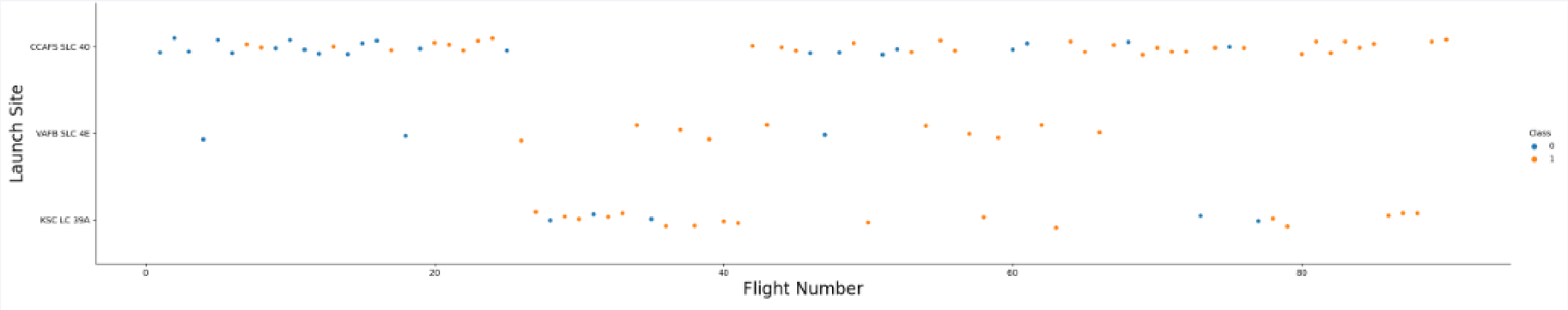
Class
0
0
0
0
0

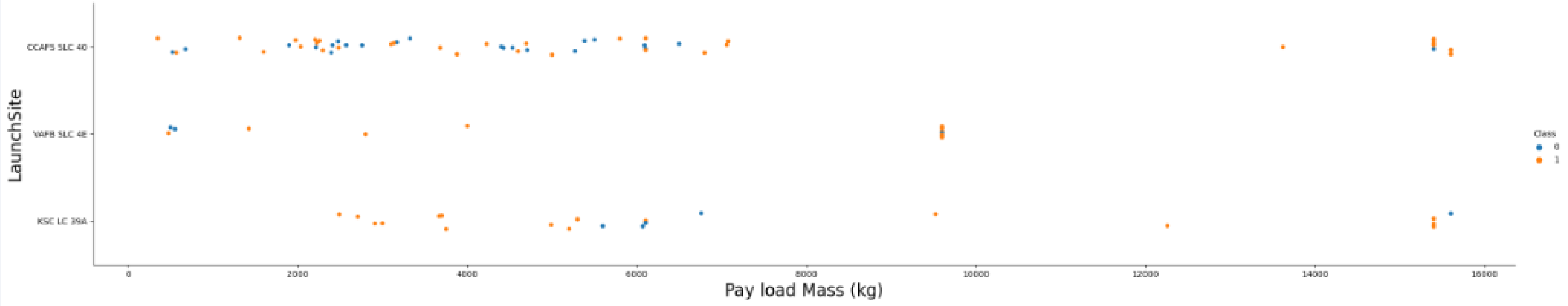
https://github.com/WaelChmaisani/Applied-Data-Science-Project/blob/main/labs-jupyter-spacex-data_wrangling.ipynb

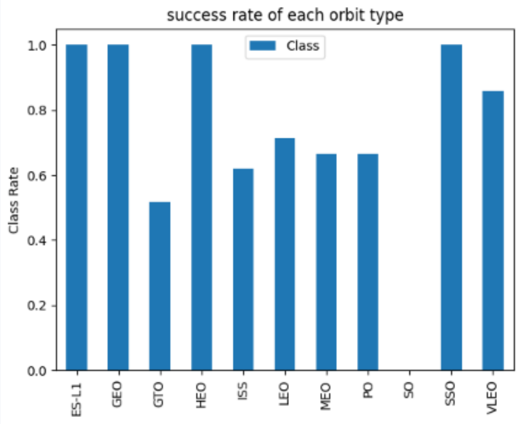
EDA with Data Visualization

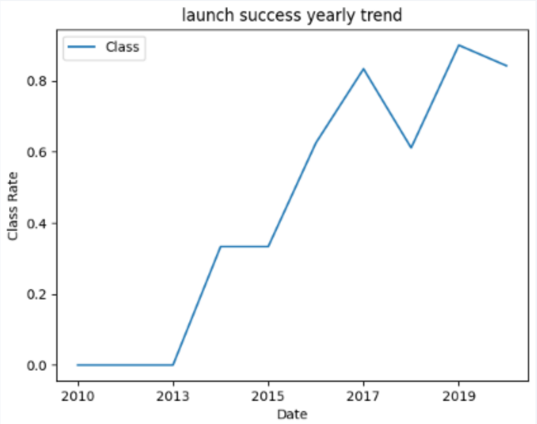
Various plots show how the variable features would affect the launch outcome











https://github.com/WaelChmaisani/Applied-Data-Science-Project/blob/main/jupyter-labs-eda-dataviz.ipynb

EDA with SQL

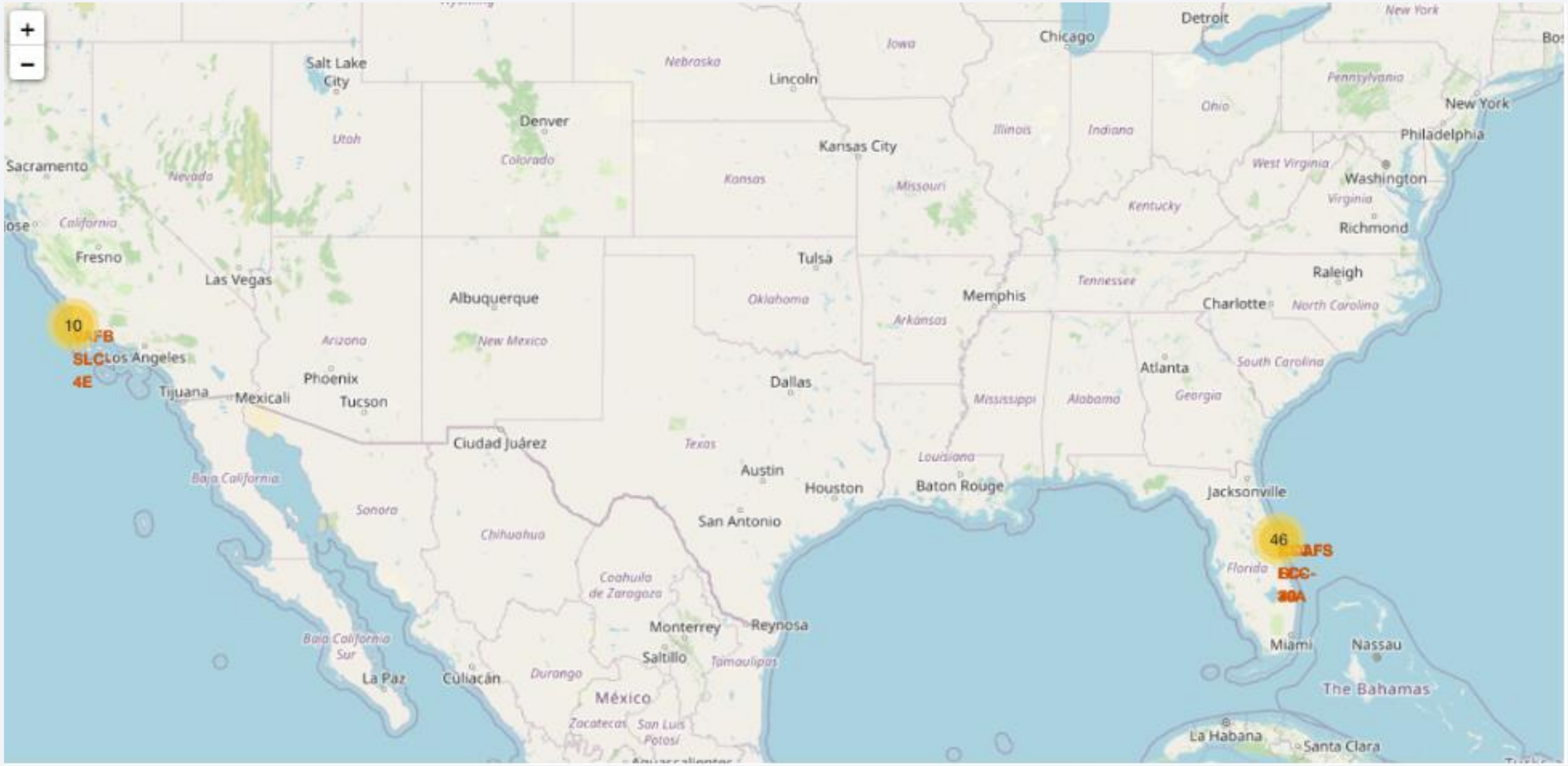
SQL carried out include:

- 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 achieved
- 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. Use a subquery
- List the records which will display the month names, failure landing outcomes in drone ship ,booster versions, launch site for the months in year 2015
- Rank the count of successful landing outcomes between the date 04-06-2010 and 20-03-2017 in descending order

https://github.com/WaelChmaisani/Applied-Data-Science-Project/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

Map with markers shows the land outcomes of launch sites

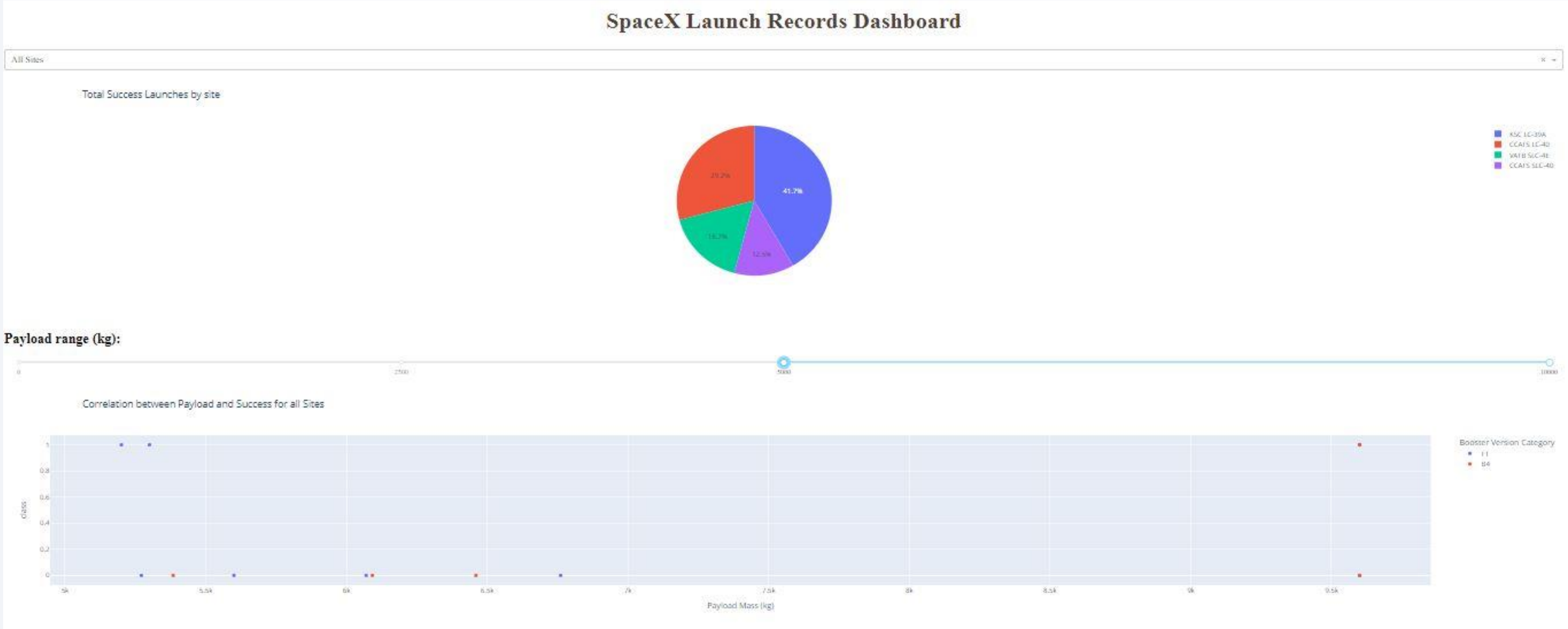


https://github.com/WaelChmaisani/Applied-Data-Science-Project/blob/main/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

Dashboard consists:

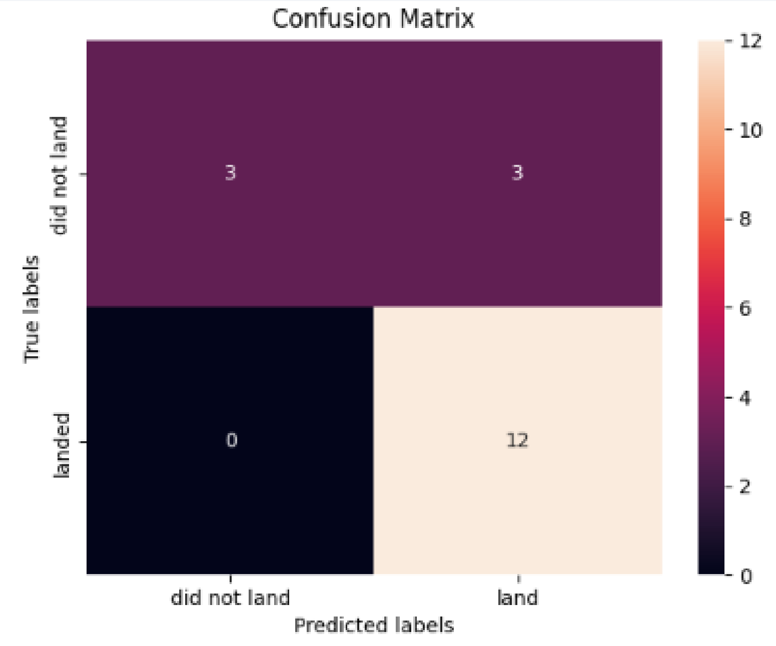
- ➢ Pie chart along with dropdown to show the land outcome rates for each site
- ➢ Scatter plot along with range slider to show the affect of pay load mass on the land outcomes



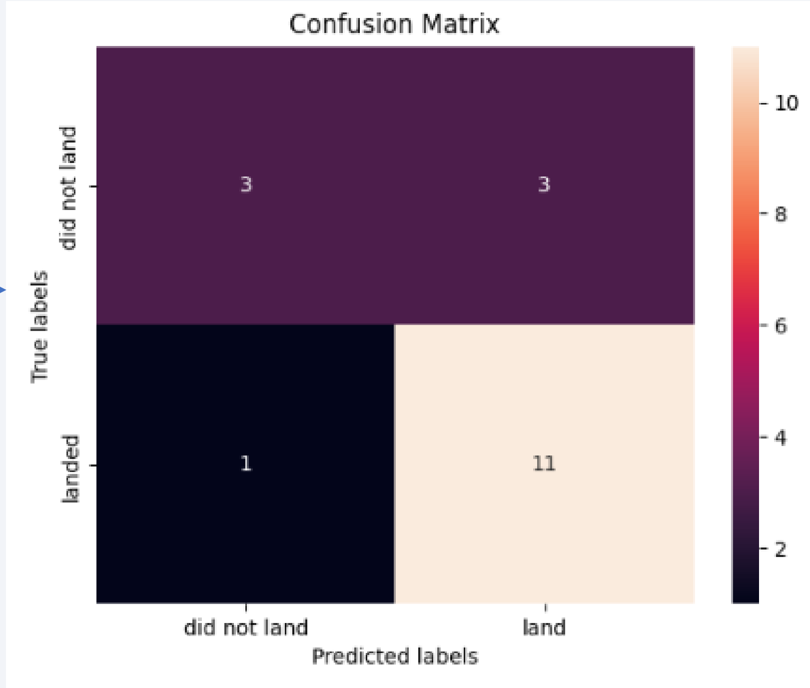
https://github.com/WaelChmaisani/Applied-Data-Science-Project/blob/main/Spacex_Falcon9_Dash.py

Predictive Analysis (Classification)

- Four classification models LR, Tree, SVM and KNN are built to predict the land outcome
- Their evaluation show best accuracy for LR, SVM and KNN through the score method along with confusion matrix



Model	Accuracy
KNN	0.833333
Tree	0.777778
LR	0.833333
SVM	0.833333



https://github.com/WaelChmaisani/Applied-Data-Science-Project/blob/main/SpaceX_Machine_Learning_Prediction.ipynb

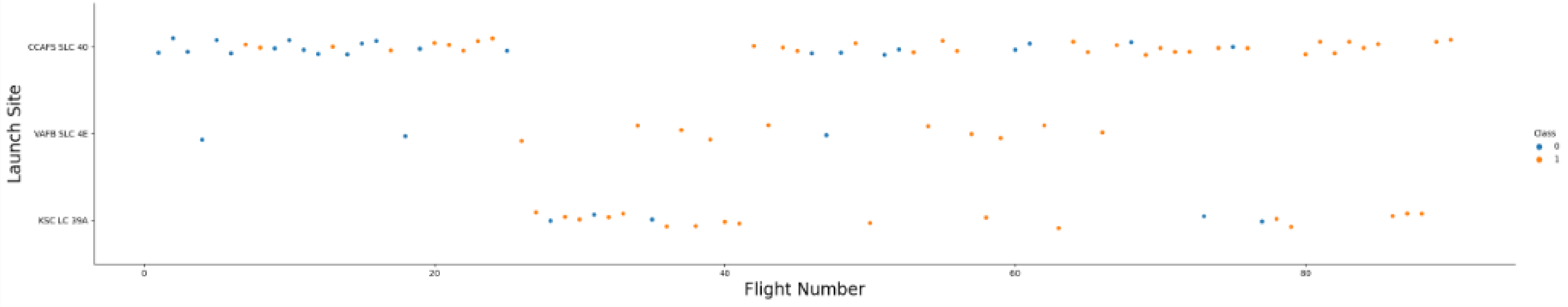
Results

- ✓ The more massive the payload, the less likely the first stage will return
- ✓ The orbits ES-L1, GEO, HEO and SSO have the highest launch success rate
- ✓ The launch site KSC LC 39A has the highest launch success rate
- ✓ FT Booster version has the highest launch success rate
- ✓ The launch success rate is increase from 2013 to 2020
- ✓ The evaluation of LR, SVM and KNN classification models results best accuracy

Section 2

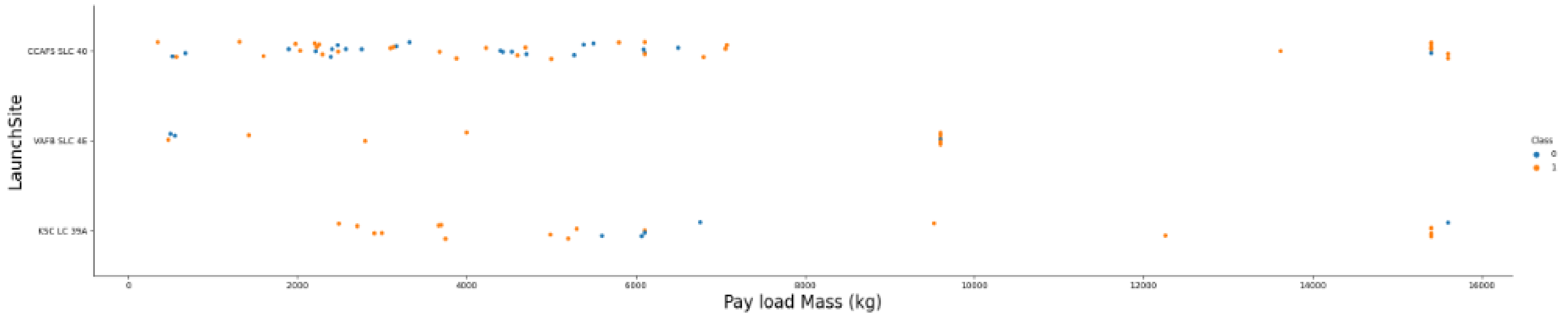
Insights drawn from EDA

Flight Number vs. Launch Site



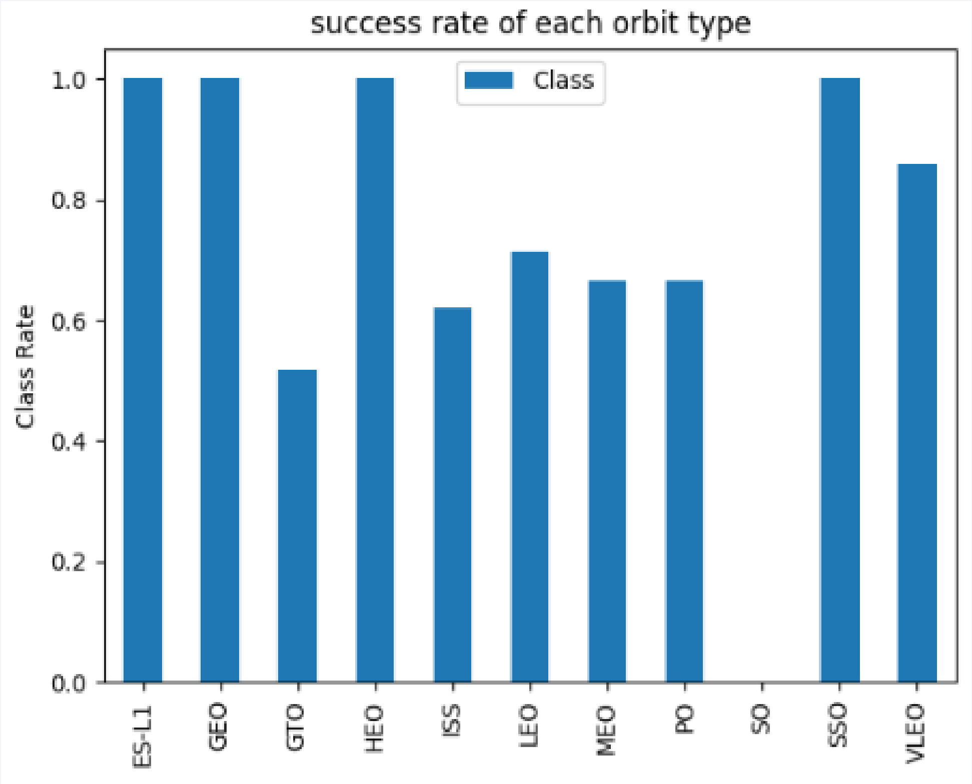
- The flight number is high for CCAFS SLC 40 launch site
- The launch site KSC LC 39A has the highest launch success rate

Payload vs. Launch Site



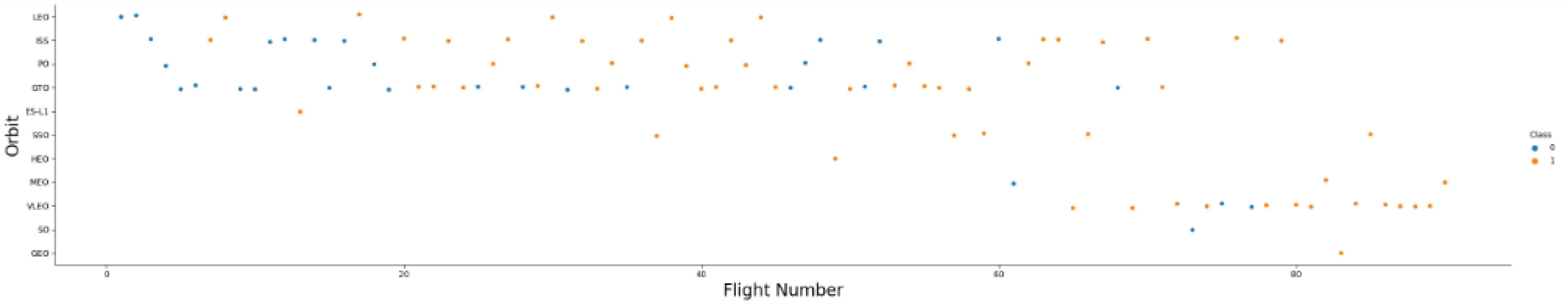
- The most launches correspond to pay load mass range between 350 kg and 7060 kg
- The launches of KSC LC 39A site, have a pay load mass range between 2490 kg and 5300 kg, are all successful
- The VAFB-SLC launch site there are no rockets launched for heavy payload mass greater than 10000

Success Rate vs. Orbit Type



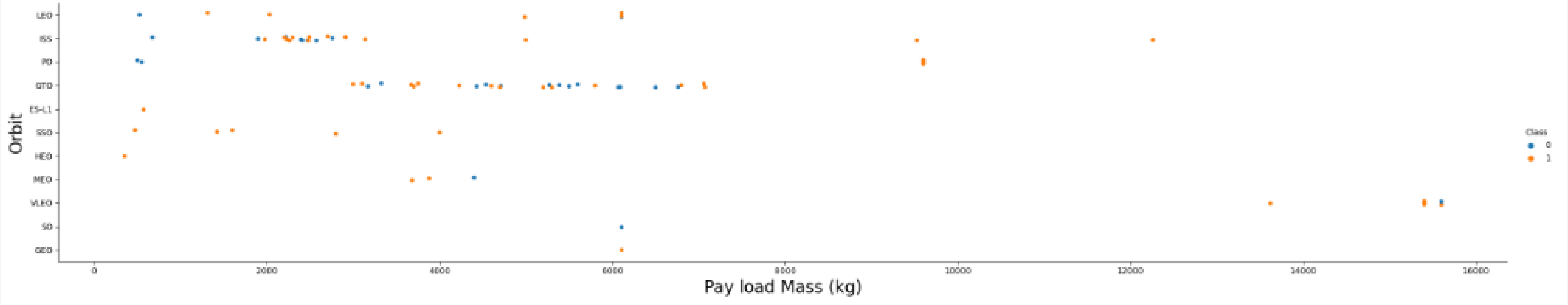
- ES-L1, GEO, HEO, and SSO orbits have the highest launch success rate
- GTO orbit has the lowest launch success rate

Flight Number vs. Orbit Type



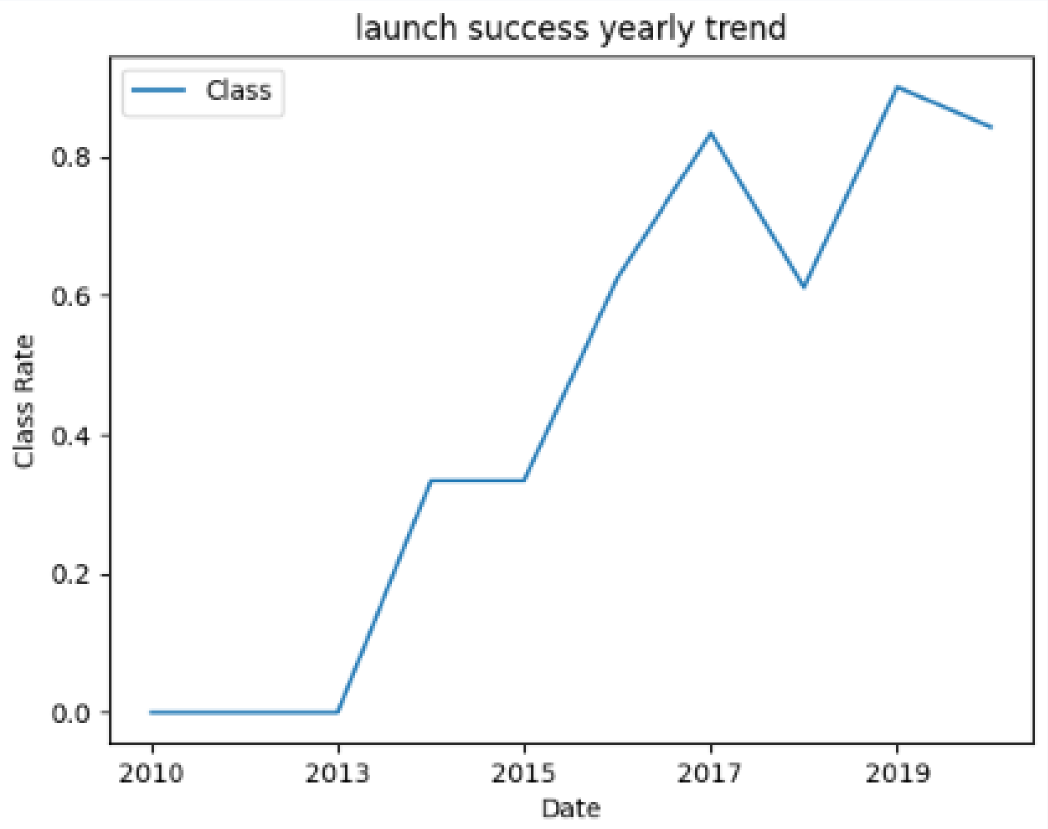
In the LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit

Payload vs. Orbit Type



- With heavy payloads the successful landing or positive landing rate are more for Polar, LEO and ISS
- 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 since 2013 kept increasing till 2020
- 2019 has the maximum success rate

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_) FROM SPACEXTBL WHERE Customer = "NASA (CRS)"
```

SUM(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE Booster_Version = "F9 v1.1"
```

AVG(PAYLOAD_MASS__KG_)
2928.4

First Successful Ground Landing Date

```
%sql SELECT Date,MIN(substr(Date,7,4)) FROM SPACEXTBL WHERE "Landing _Outcome" = "Success (ground pad)"
```

Date	MIN(substr(Date,7,4))
22-12-2015	2015

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 "Mission_Outcome",COUNT("Mission_Outcome") FROM SPACEXTBL GROUP BY "Mission_Outcome"
```

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

Boosters Carried Maximum Payload

```
%sql SELECT Booster_Version,PAYLOAD_MASS__KG_ FROM SPACEXTBL WHERE "PAYLOAD_MASS__KG_" = (SELECT MAX("PAYLOAD_MASS__KG_") FROM SPACEXTBL)
```

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 substr(Date, 4, 2) AS MONTH,"Landing _Outcome","Booster_Version","Launch_Site"
FROM SPACEXTBL WHERE "Landing _Outcome" = "Failure (drone ship)" AND substr(Date,7,4) = "2015"
```

MONTH	Landing _Outcome	Booster_Version	Launch_Site
01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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

```
%sql SELECT "Landing _Outcome",COUNT(*) AS COUNT_LAUNCHES FROM SPACEXTBL \
WHERE ("Landing _Outcome" LIKE "Success%") AND (Date BETWEEN "04-06-2010" and "20-03-2017") \
GROUP BY "Landing _Outcome" \
ORDER BY COUNT_LAUNCHES DESC
```

Landing _Outcome	COUNT_LAUNCHES
Success	20
Success (drone ship)	8
Success (ground pad)	6

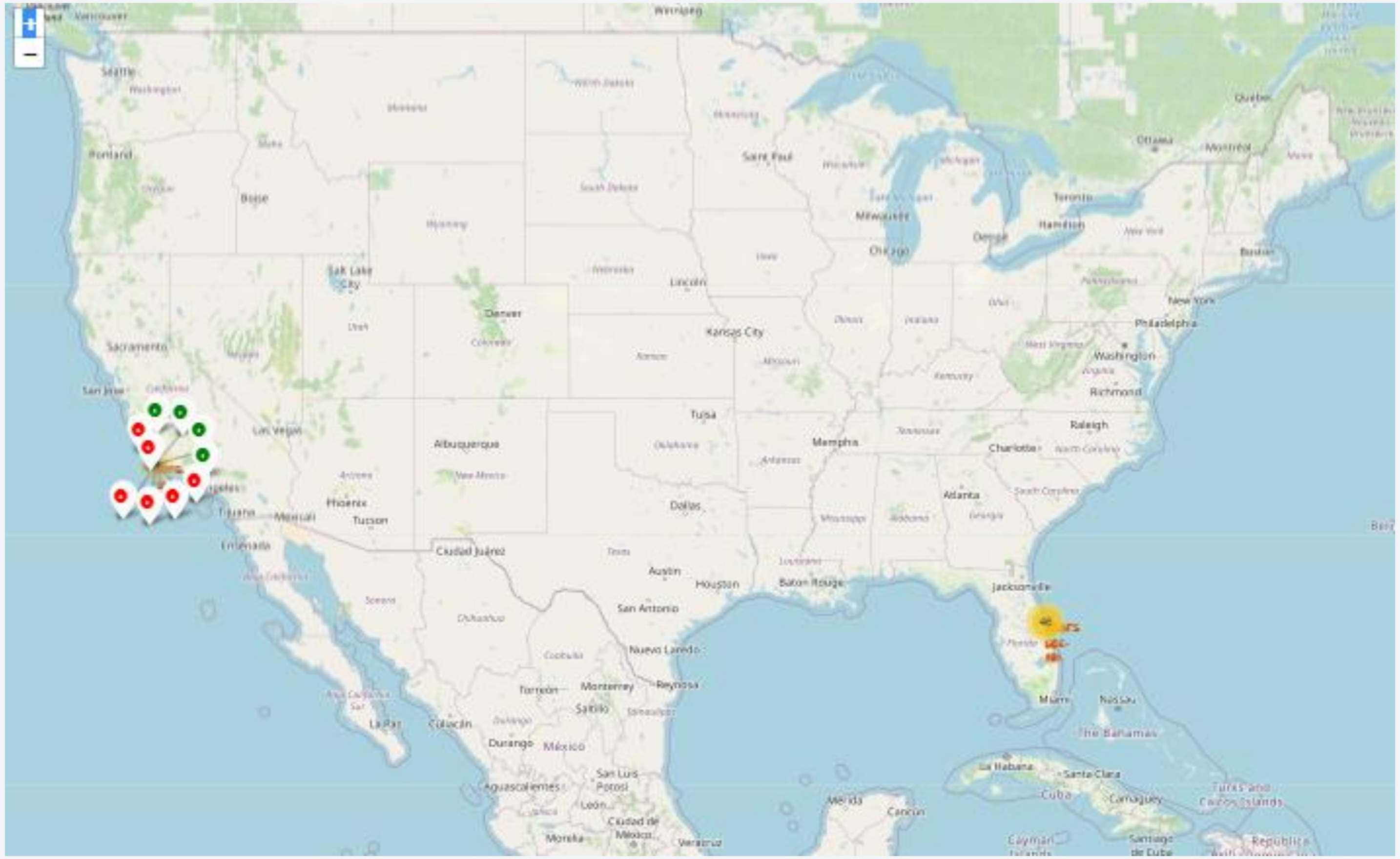
Section 3

Launch Sites Proximities Analysis

Map Of Marked Launch Sites

All launch sites are marked by a small yellow circle and their label names

Map Of Color-Labeled Launch Success/Failed Outcomes



Green mark: Successful launch
Red mark: Failed launch

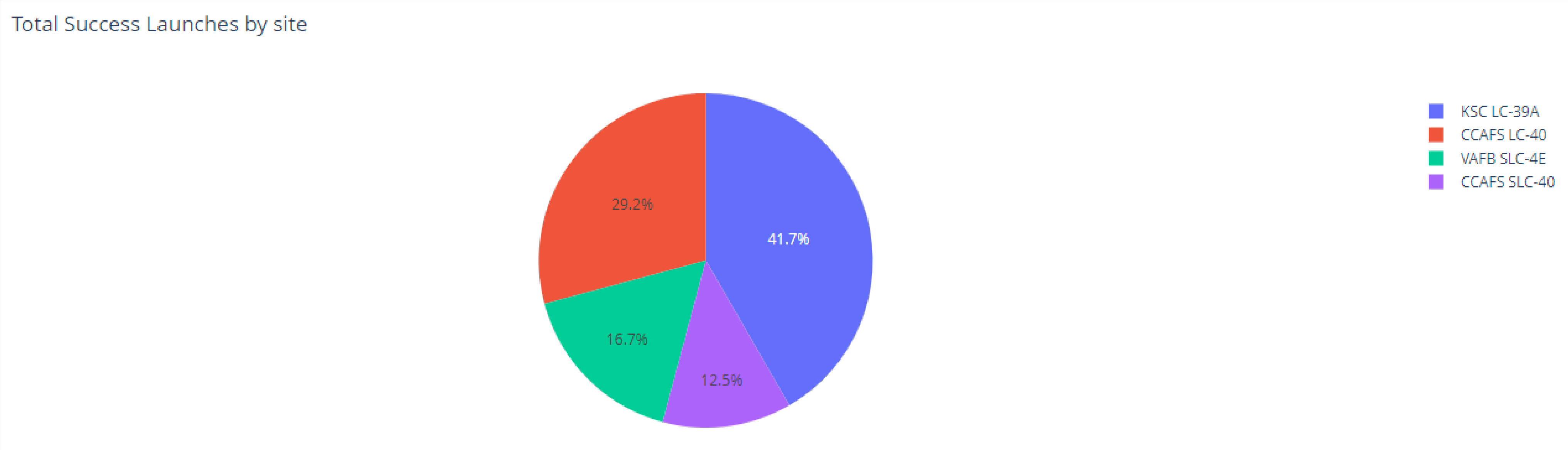
Launch Sites Proximities

Blue line: distance between the launch site and its proximities

Section 4

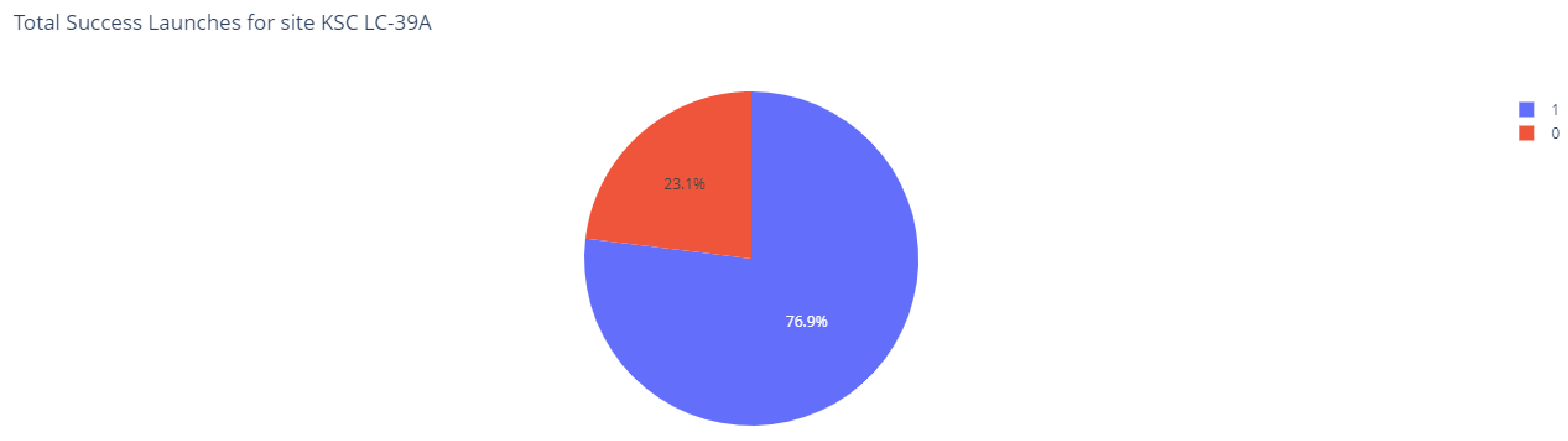
Build a Dashboard with Plotly Dash

Launch Success Count For All Sites



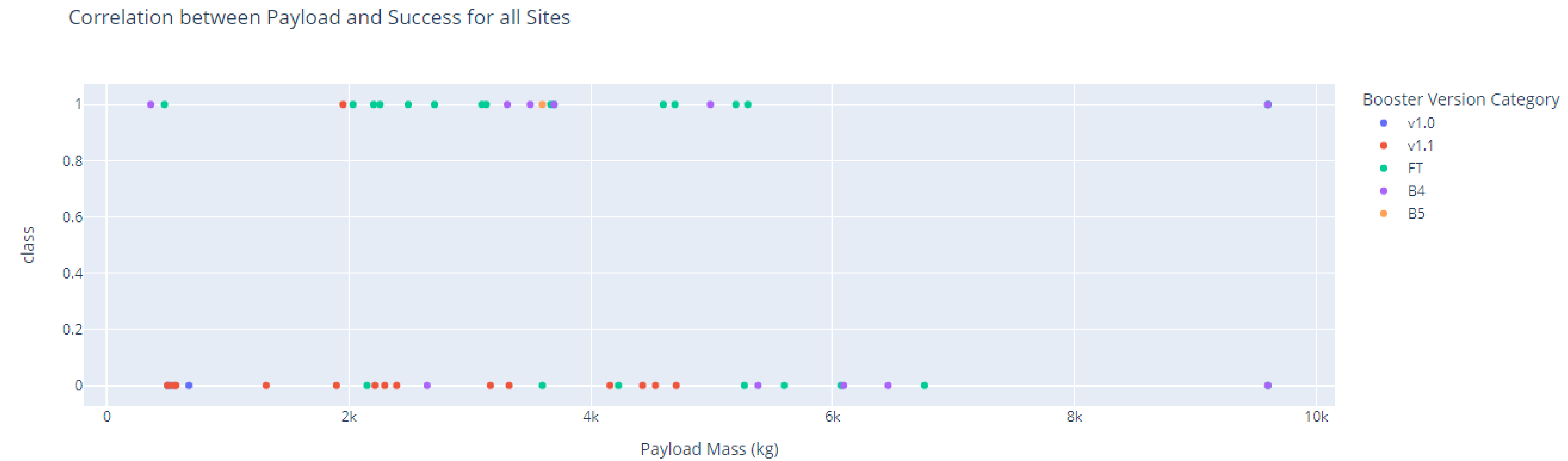
- The launch site KSC LC-39A has the highest launch success rate
- CCAFS SLC-40 has the lowest success rate

Launch Site With Highest Launch Success Ratio



77% of KSC LC-39A launches were successfully landing

Payload vs. Launch Outcome Scatter Plot For All Sites

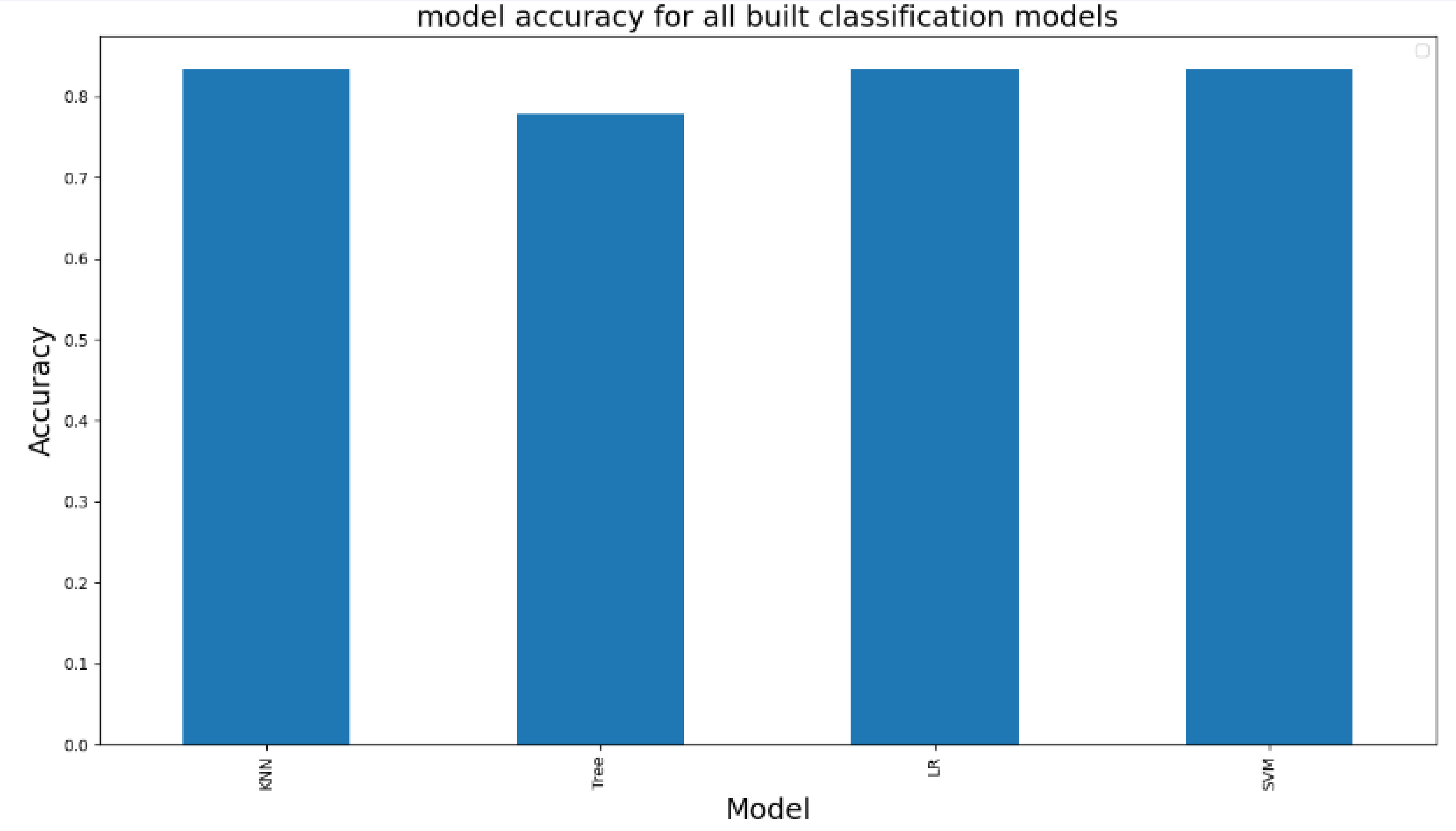


FT Booster version has the highest launch success rate with payload range between 2034 kg and 5300 kg

Section 5

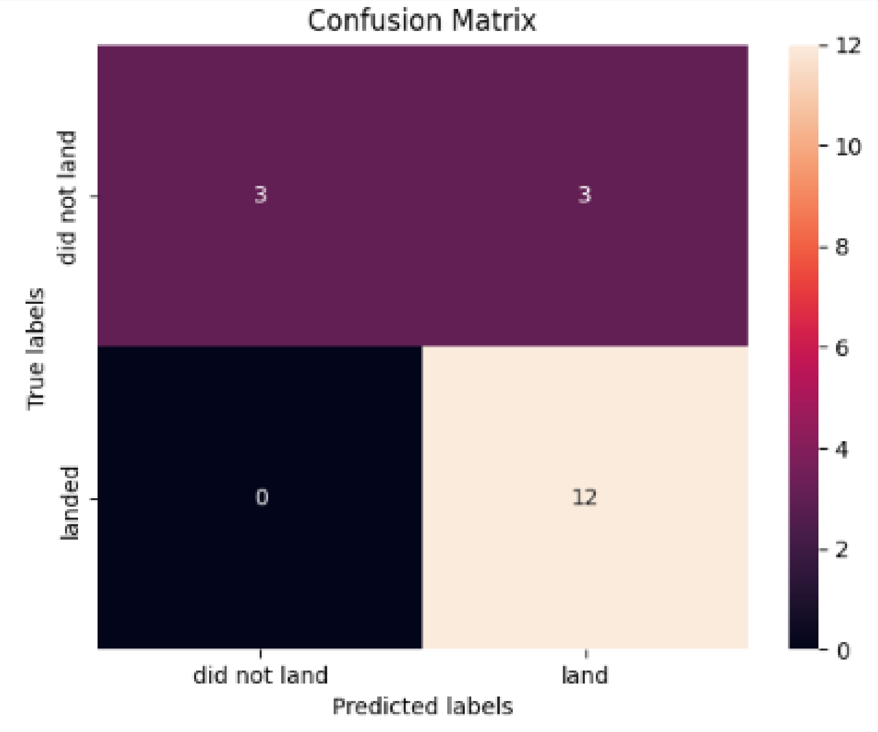
Predictive Analysis (Classification)

Classification Accuracy

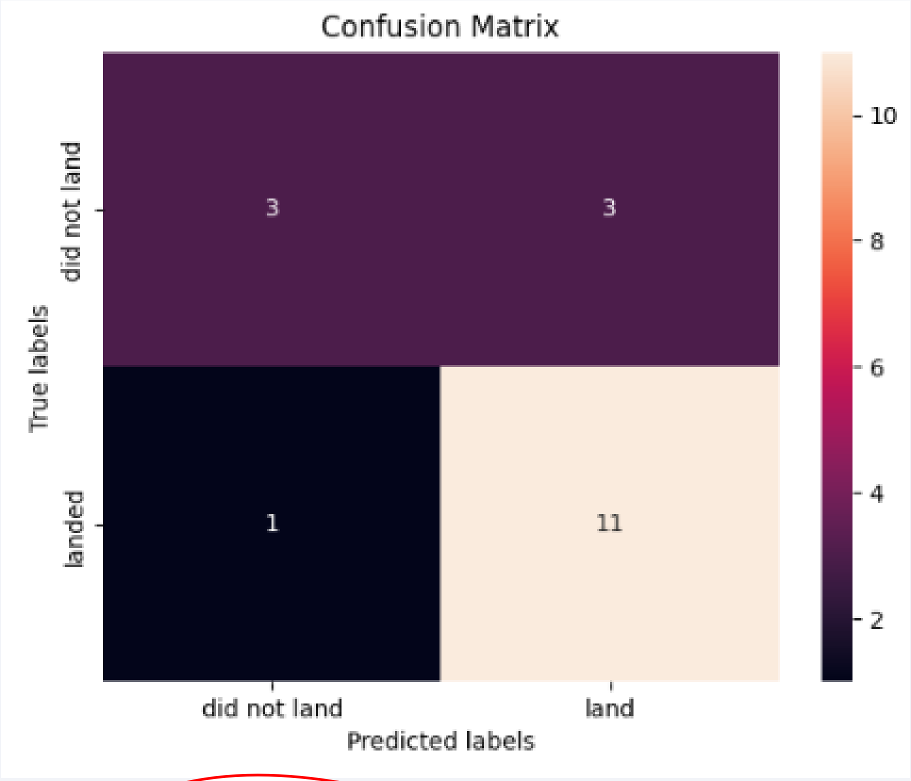


LR, SVM and KNN models have the highest classification accuracy

Confusion Matrix



LR, SVM, KNN



Tree

Conclusions

✓ The highest launch success rate of space X Falcon 9 first stage landing corresponds to:

- ❑ ES-L1, GEO, HEO and SSO orbits
- ❑ KSC LC 39A launch site
- ❑ FT Booster version
- ❑ Low pay load mass

✓ Classification models show accuracy of Space X Falcon 9 first stage landing Prediction for:

- ❑ Logistic Regression LR
- ❑ Support Vector Machine SVM
- ❑ K-Nearest Neighbors KNN

Thank you!