IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Craig M. Rash Jr

2023.02.04

Outline

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

Executive Summary

- Summary of methodologies:
  - Data Collection
  - Data Wrangling
  - EDA with Visualizations
  - Predictive Analysis
- Summary of all results
  - Matplotlib graphs
  - Interactive visualizations
  - Categorical Models

Introduction

- Background: To determine our competitiveness against SpaceX, we must determine the likelihood of SpaceX rocket success landing.
- Can we predict the success of a given launch?
    - Launch site,
    - Payload mass,
    - Rocket type,
    - Landing legs,
    - Grid fins,
    - Mission type
    - Proximity to other infrastructure.

Section 1

Methodology

Methodology

Executive Summary

- Data collection
- Performed data wrangling
- Performed exploratory data analysis (EDA) using visualization and SQL
- Performed interactive visual analytics using Folium and Plotly Dash
- Performed predictive analysis using classification models

Data Collection

- Data Requirements
  - Source: https://api.spacexdata.com/v4/launches/past
  - Source: https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922

Data Collection – SpaceX API

Defining the URL:
spacex_url="https://api.spacexdata.com/v4/launches/past"

Requesting the URL:
response = requests.get(spacex_url)

print(response.content)

=

b'[{"fairings":{"reused":false,"recovery_attempt":false,"recovered":false,"ships":[]},"lir
mages2.imgbox.com/94/f2/NN6Ph45r_o.png","large":"https://images2.imgbox.com/5b/02/QcxHUb5\
ll,"launch":null,"media":null,"recovery":null},"flickr":{"small":[],"original":[]},"pressk
youtube.com/watch?v=0a_00nJ_Y88","youtube_id":"0a_00nJ_Y88","article":"https://www.space.c
1-rocket-lost-launch.html","wikipedia":"https://en.wikipedia.org/wiki/DemoSat"},"static_fi
0.000Z","static_fire_date_unix":1142553600,"net":false,"window":0,"rocket":"5e9d0d95eda699
lures":[{"time":33,"altitude":null,"reason":"merlin engine failure"}],"details":"Engine fa
vehicle","crew":[],"ships":[],"capsules":[],"payloads":["5eb0e4b5b6c3bb0006eeb1e1"],"launc
6","flight_number":1,"name":"FalconSat","date_utc":"2006-03-24T22:30:00.000Z","date_unix":
-25T10:30:00+12:00","date_precision":"hour","upcoming":false,"cores":[{"core":"5e9e289df35
ins":false,"legs":false,"reused":false,"landing_attempt":false,"landing_success":null,"lar
l}],"auto_update":true,"tbd":false,"launch_library_id":null,"id":"5eb87cd9ffd86e000604b32a
e,"recovery_attempt":false,"recovered":false,"ships":[]},"links":{"patch":{"small":"https:
eNb_o.png","large":"https://images2.imgbox.com/80/a2/bkWotCIS_o.png"},"reddit":{"campaign"
l,"recovery":null},"flickr":{"small":[],"original":[]},"presskit":null,"webcast":"https://
-Nc","youtube_id":"Lk4zQ2wP-Nc","article":"https://www.space.com/3590-spacex-falcon-1-rock

https://github.com/GitCraigRash/SpaceX_Presentation/blob/main/2023.02.04.SpaceX_Presentation%2CData_Collection.md

Data Wrangling

Raw Data

static_json_url= 'https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_call_spacex_api.json'

Convert to DataFrame.

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

Isolate Relevant Data

Isolate important information
data =
data[['rocket', 'payloads', 'launchpad', 'cores', 'flight_number', 'date_utc']]

https://github.com/GitCraigRash/SpaceX_Presentation/blob/main/2023.02.04.SpaceX_Presentation%2CData_Collection.md

Data Wrangling

Removed Multiple Cores and multiple payloads

```
data = data[data['cores'].map(len)==1]
data = data[data['payloads'].map(len)==1]
```

Changed data type of ['cores'] and ['payloads']

```
data['cores'] = data['cores'].map(lambda x : x[0])
data['payloads'] = data['payloads'].map(lambda x : x[0])
```

Converted data to datetime and extracting the data.

```
data['date'] = pd.to_datetime(data['date_utc']).dt.date
```


https://github.com/GitCraigRash/SpaceX_Presentation/blob/main/2023.02.04.SpaceX_Presentation%2CData_Collection.md

Data Wrangling

Restricted the dates of launches to before November 13th 2020.

```
data = data[data['date'] <= datetime.date(2020, 11, 13)]
```

Created empty lists of desired columns.

```
BoosterVersion = []      Legs = []
PayloadMass = []         LandingPad = []
Orbit = []               Block = []
LaunchSite = []          ReusedCount = []
Outcome = []             Serial = []
Flights = []             Longitude = []
GridFins = []            Latitude = []
Reused = []
```

https://github.com/GitCraigRash/SpaceX_Presentation/blob/main/2023.02.04.SpaceX_Presentation%2CData_Collection.md

Data Wrangling

Used predefined methods to extract from data and populate lists.

```
def getBoosterVersion(data):
    for x in data['rocket']:
        response = requests.get("https://api.spacexdata.com/v4/rockets/"+str(x)).json()
        BoosterVersion.append(response['name'])
```

https://github.com/GitCraigRash/SpaceX_Presentation/blob/main/2023.02.04.SpaceX_Presentation%2CData_Collection.md

Data Wrangling

Used predefined methods to extract from data and populate lists.

```
def getLaunchSite(data):
    for x in data['launchpad']:
        response = requests.get("https://api.spacexdata.com/v4/launchpads/"+str(x)).json()
        Longitude.append(response['longitude'])
        Latitude.append(response['latitude'])
        LaunchSite.append(response['name'])
```

https://github.com/GitCraigRash/SpaceX_Presentation/blob/main/2023.02.04.SpaceX_Presentation%2CData_Collection.md

Data Wrangling

Used predefined methods to extract from data and populate lists.

```
def getPayloadData(data):
    for load in data['payloads']:
        response = requests.get("https://api.spacexdata.com/v4/payloads/"+load).json()
        PayloadMass.append(response['mass_kg'])
        Orbit.append(response['orbit'])
```


https://github.com/GitCraigRash/SpaceX_Presentation/blob/main/2023.02.04.SpaceX_Presentation%2CData_Collection.md

Data Wrangling

```
def getCoreData(data):
    for core in data['cores']:
            if core['core'] != None:
                response = requests.get("https://api.spacexdata.com/v4/cores/"+core['core']).json()
                Block.append(response['block'])
                ReusedCount.append(response['reuse_count'])
                Serial.append(response['serial'])
            else:
                Block.append(None)
                ReusedCount.append(None)
                Serial.append(None)
            Outcome.append(str(core['landing_success'])+' '+str(core['landing_type']))
            Flights.append(core['flight'])
            GridFins.append(core['gridfins'])
            Reused.append(core['reused'])
            Legs.append(core['legs'])
            LandingPad.append(core['landpad'])
```

Data Wrangling

Created a Dictionary and changed into DataFrame.

```
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}
```

https://github.com/GitCraigRash/ThirdGit/blob/main/2023.02.04.SpaceX_Presentation%2CData_Collection.md

Data Wrangling

Resulting DataFrame:

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	Landing
0	1	2006-03-24	Falcon 1	20.0	LEO	Kwajalein Atoll	None None	1	False	False	False	N
1	2	2007-03-21	Falcon 1	NaN	LEO	Kwajalein Atoll	None None	1	False	False	False	N
2	4	2008-09-28	Falcon 1	165.0	LEO	Kwajalein Atoll	None None	1	False	False	False	N
3	5	2009-07-13	Falcon 1	200.0	LEO	Kwajalein Atoll	None None	1	False	False	False	N
4	6	2010-06-04	Falcon 9	NaN	LEO	CCSFS SLC 40	None None	1	False	False	False	N

https://github.com/GitCraigRash/SpaceX_Presentation/blob/main/2023.02.04.SpaceX_Presentation%2CData_Collection.md

Data Wrangling

- Removing and Changing Data
  - Filter to include only Falcon 9 launches
  - Reset flight numbers
  - Replace Nan values with the mean

https://github.com/GitCraigRash/SpaceX_Presentation/blob/main/2023.02.04.SpaceX_Presentation%2CData_Collection.md

Data Wrangling

- Removing and Changing Data

- Filter to include only Falcon 9 launches

```
data_falcon9 = data.loc[data['BoosterVersion']!='Falcon 1']
```

- Reset flight numbers

```
data_falcon9.loc[:,'FlightNumber'] = list(range(1, data_falcon9.shape[0]+1))
```

- Replace Nan values with the mean.

```
Payload_mean=data_falcon9["PayloadMass"].mean()
data_falcon9.fillna(value=Payload_mean)
```


https://github.com/GitCraigRash/SpaceX_Presentation/blob/main/2023.02.04.SpaceX_Presentation%2CData_Collection.md

EDA with Data Visualization

- Visualizations
  - Payload Mass vs Flight number
  - Launch Site vs Payload Mass
  - Orbit type vs Number of Successful Missions
  - Orbit type vs Flight number
  - Orbit type to Payload mass
  - Number of Successful Flights over Time

https://github.com/GitCraigRash/ThirdGit/blob/main/IBM-DS0321EN-SkillsNetwork_labs_module_2_jupyter-labs-eda-dataviz.ipynb.jupyterlite%20(1).ipynb

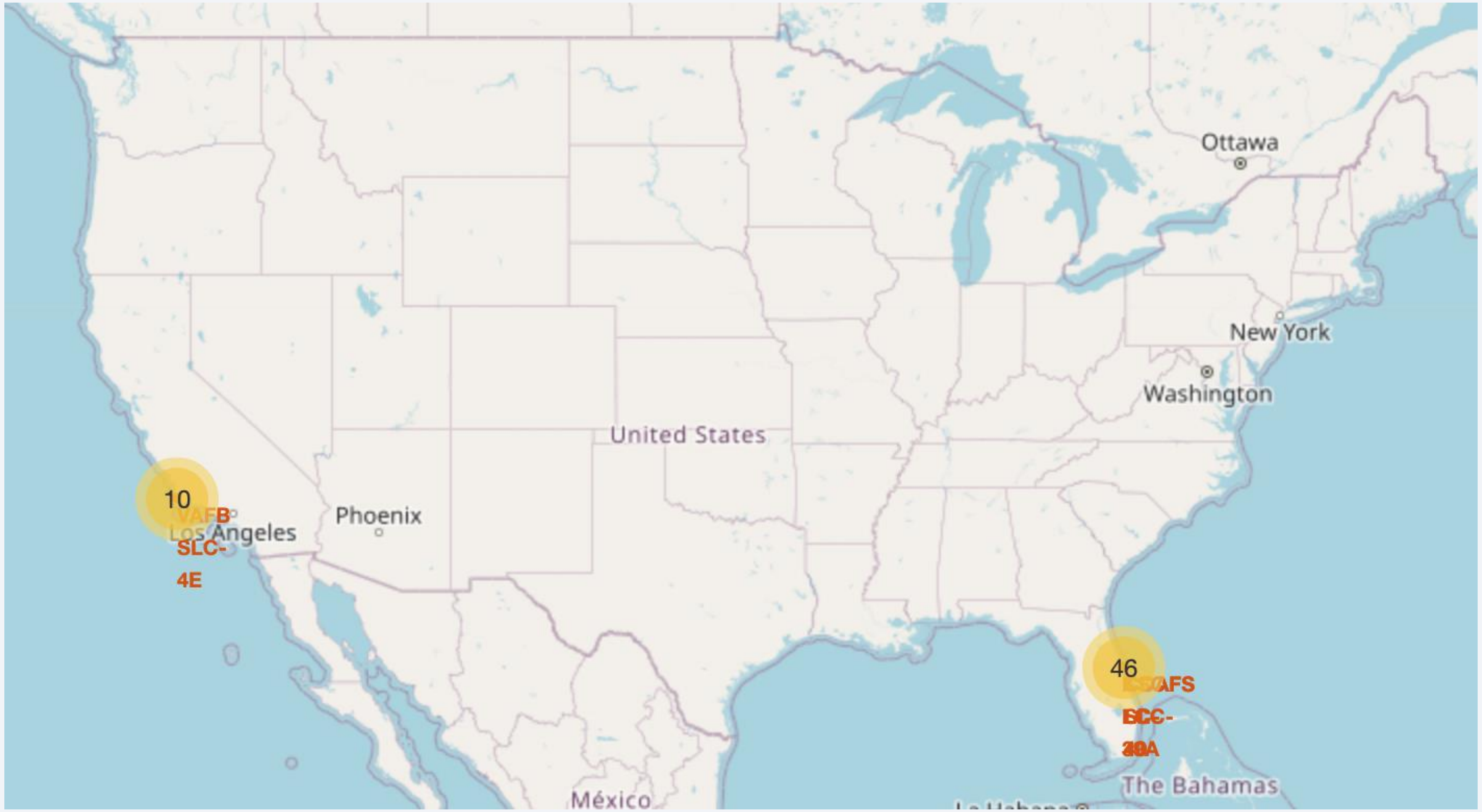
EDA with SQL

- SQL queries
  - Display the names of the unique launch sites in the space mission.
  - Total payload mass carried by boosters launched by NASA (CRS)
  - Display average payload mass carried by booster version F9 v1.1
  - The date when the first successful landing outcome in ground pad was achieved.
  - 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 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/GitCraigRash/SpaceX_Presentation/blob/main/2023.02.04%2CSpaceX_Presentation%2CSQL.md

Build an Interactive Map with Folium

- Folium Map

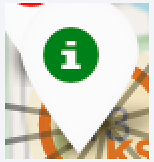


https://github.com/GitCraigRash/SpaceX_Presentation/blob/main/2020.02.04%2CSpaceX_Presentation%2CFolium.md

Build an Interactive Map with Folium

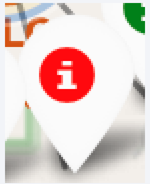
- Labels, Markers and Clusters

Launch Site



Successful

Cluster

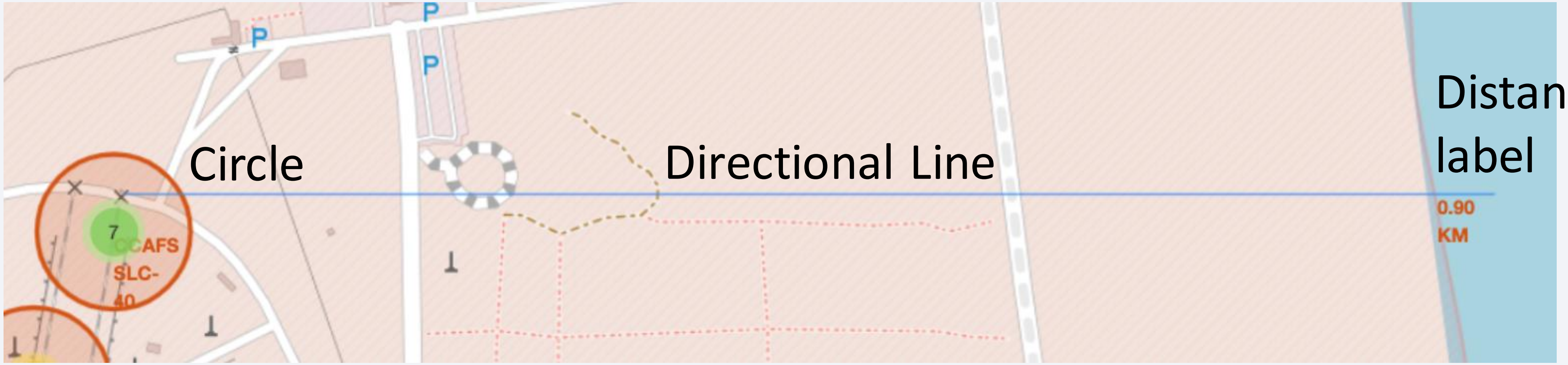


Failure

https://github.com/GitCraigRash/SpaceX_Presentation/blob/main/2020.02.04%2CSpaceX_Presentation%2CFolium.md

Build an Interactive Map with Folium

- Circles, Lines, and Labels



Build a Dashboard with Plotly Dash

- Plots/graphs on the dashboard
  - Pie chart of successful launches by launch site.
  - Dropdown displays pie chart for all launches at a particular site.
  - Scatter plot displays the payload weight, booster version, and successfulness.
  - Slider filters scatter plot by payload weight.

https://github.com/GitCraigRash/SpaceX_Presentation/blob/main/Plotly_Dash_Code

Build a Dashboard with Plotly Dash

- The pie chart of all sites indicates the proportions of successful launches SpaceX has experienced at each site.
- Dropdown- To see the proportion of missions were successful for that site.
- Scatterplot- shows the relationship of the booster version and payload mass to successful landings.
- Slider -To focus on a specific range of payload mass. To identify the probability of future successful launches in that range.

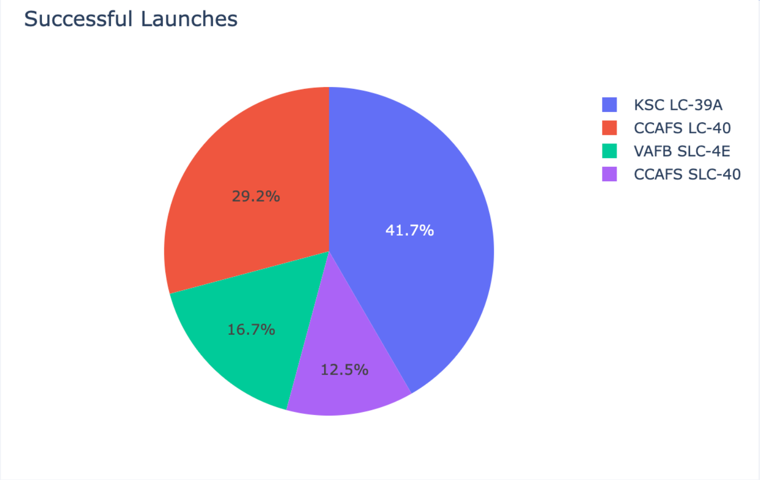
https://github.com/GitCraigRash/SpaceX_Presentation/blob/main/Plotly_Dash_Code

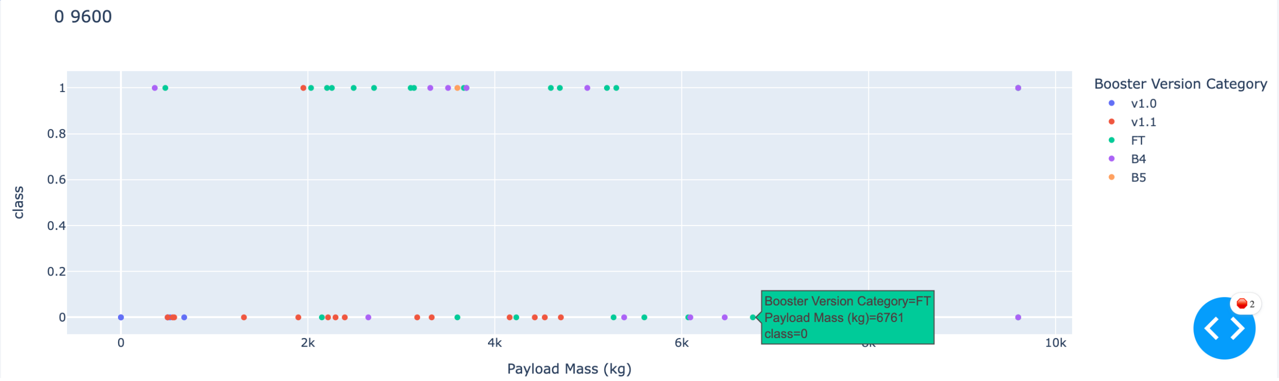
Build a Dashboard with Plotly Dash

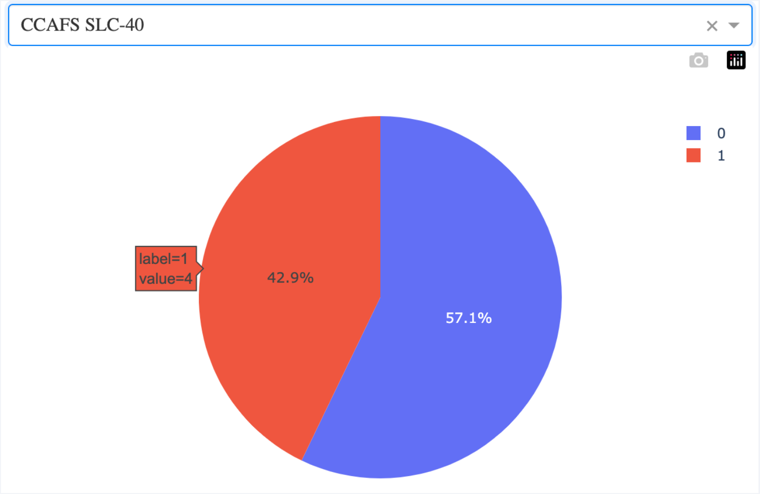
- Visualizing these facts will alert the investor that SpaceX's risk of failure is too high and SpaceY is a better investment.

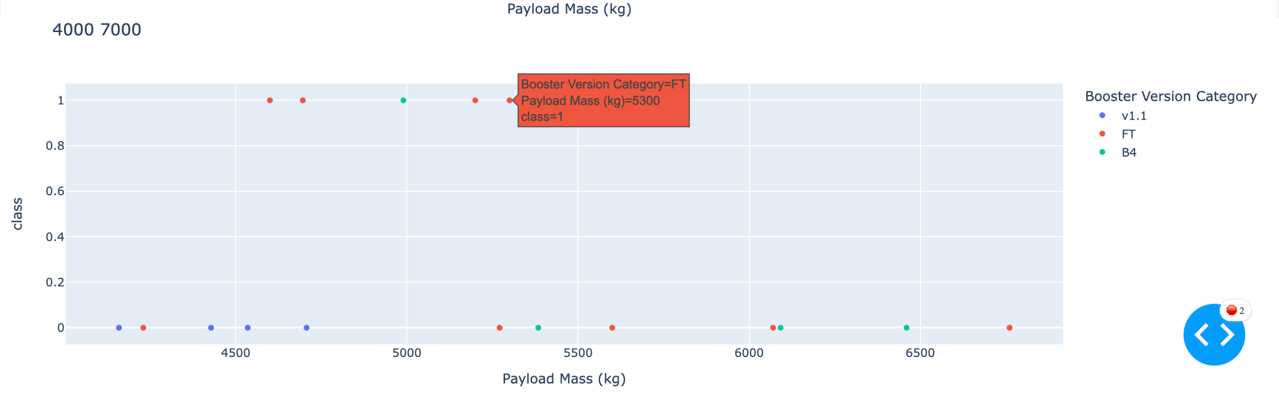
https://github.com/GitCraigRash/SpaceX_Presentation/blob/main/Plotly_Dash_Code

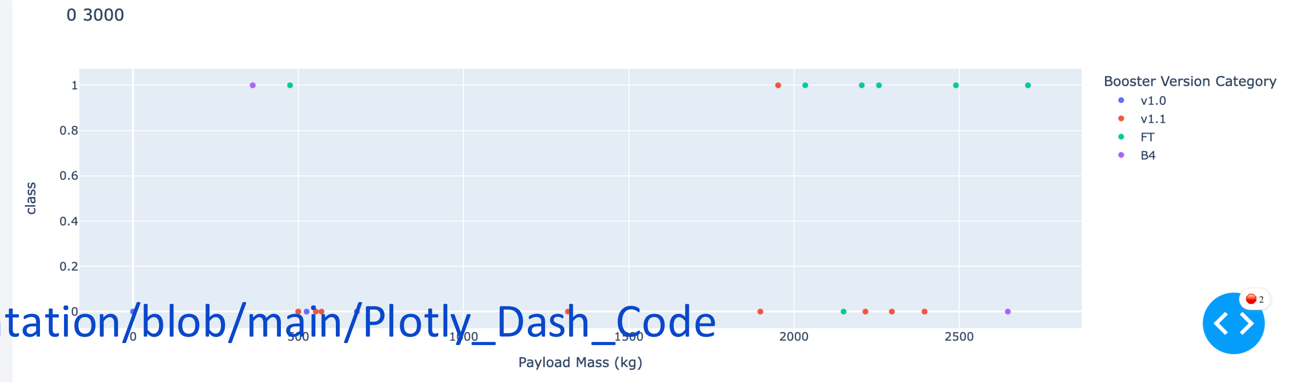
Build a Dashboard with Plotly Dash







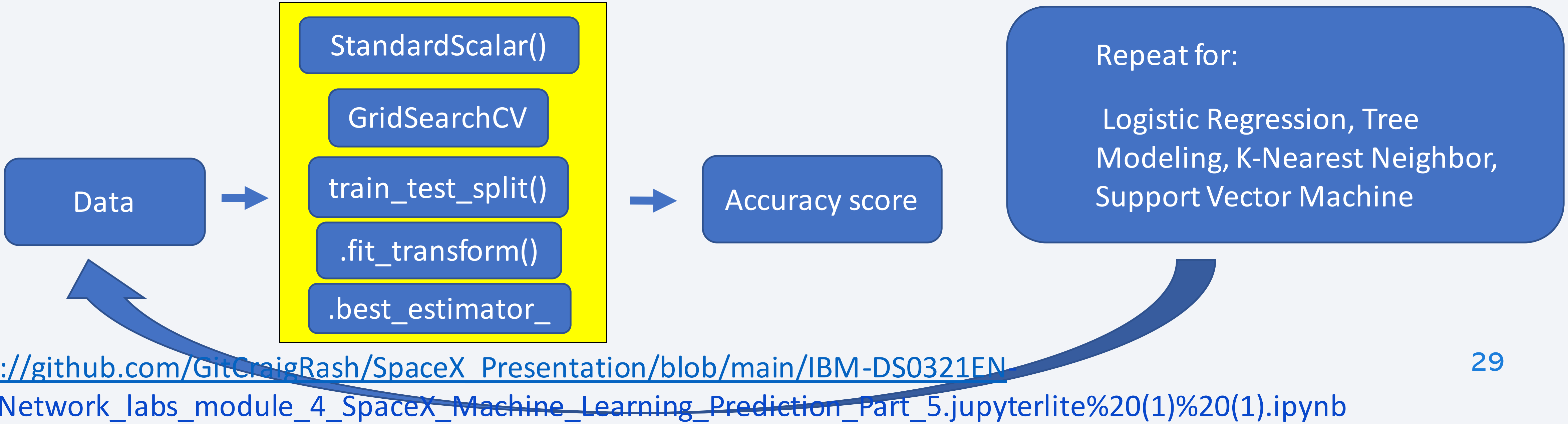




https://github.com/GitCraigRash/SpaceX_Presentation/blob/main/Plotly_Dash_Code

Predictive Analysis (Classification)

- I took the information, scaled it using .StandardScalar(), and fit the data using .fit_transform() and split it into A/B 80/20%. Used GridSearchCV on the parameters and fit the model with the data. Finally, used .best_estimator_ to find the model's accuracy.



https://github.com/GitCraigRash/SpaceX_Presentation/blob/main/IBM-DS0321EN-SkillsNetwork_labs_module_4_SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite%20(1)%20(1).ipynb

Results

Predictive analysis results

Logistic Regression: 0.8333333333333334

Support Vector Machine : 0.8333333333333334

Decision Tree Classifier: 0.9444444444444445

K-Nearest Neighbors: 0.8333333333333334

All models obtained: 0.9444444444444445

*Results slightly differ between runs because of randomness in training/test split and automatic parameter changes.

https://github.com/GitCraigRash/SpaceX_Presentation/blob/main/IBM-DS0321EN-SkillsNetwork_labs_module_4_SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite%20(1)%20(1).ipynb

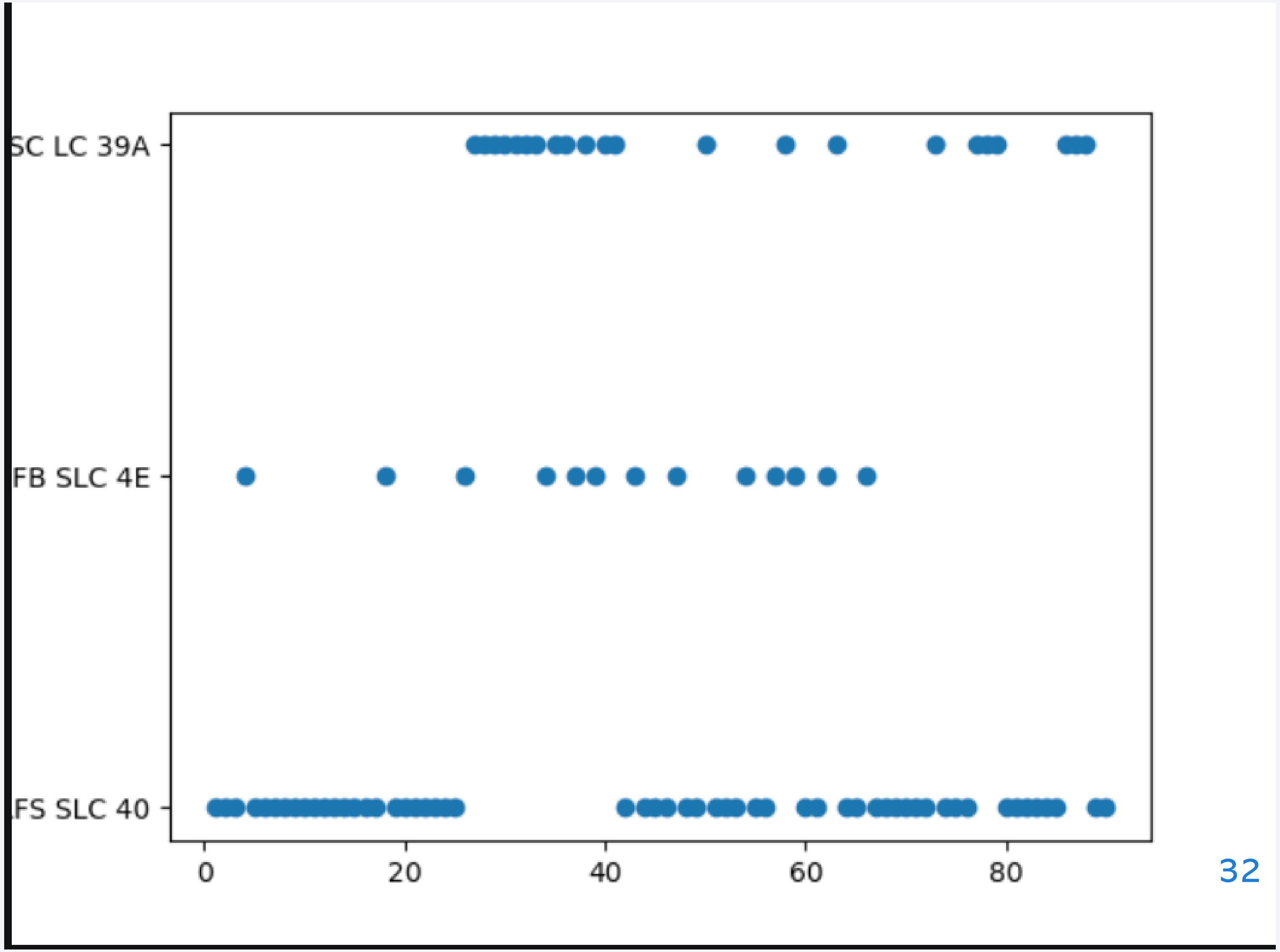
Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

- Flight Number vs. Launch Site

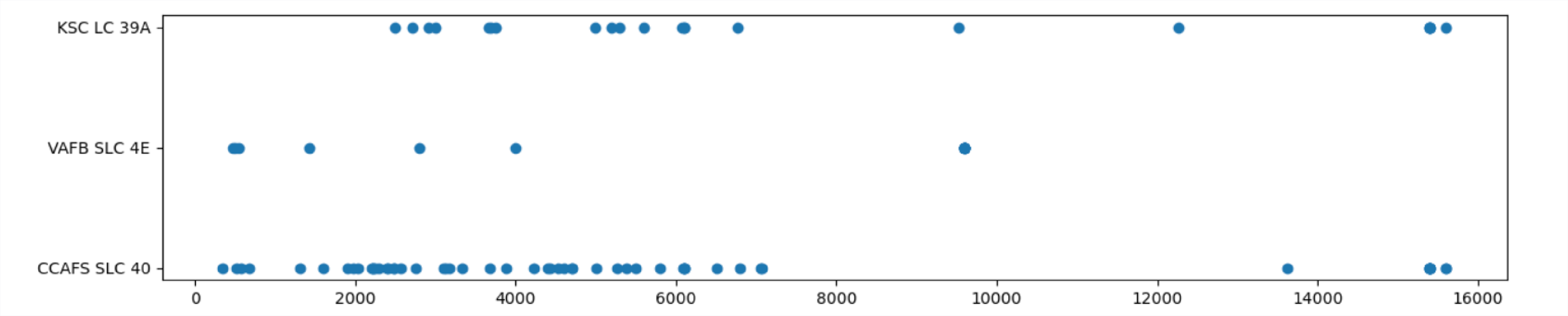
Each row represents launch sites in the East and West coast. It is noteworthy that there was a hard shift to cite SC LC 39A (California) after the 1st of September 2016 explosion on the launch pad.
15 months later, flights at CCSFS SLC 40 (Florida) continued on the 15th of December 2017.



Payload vs. Launch Site

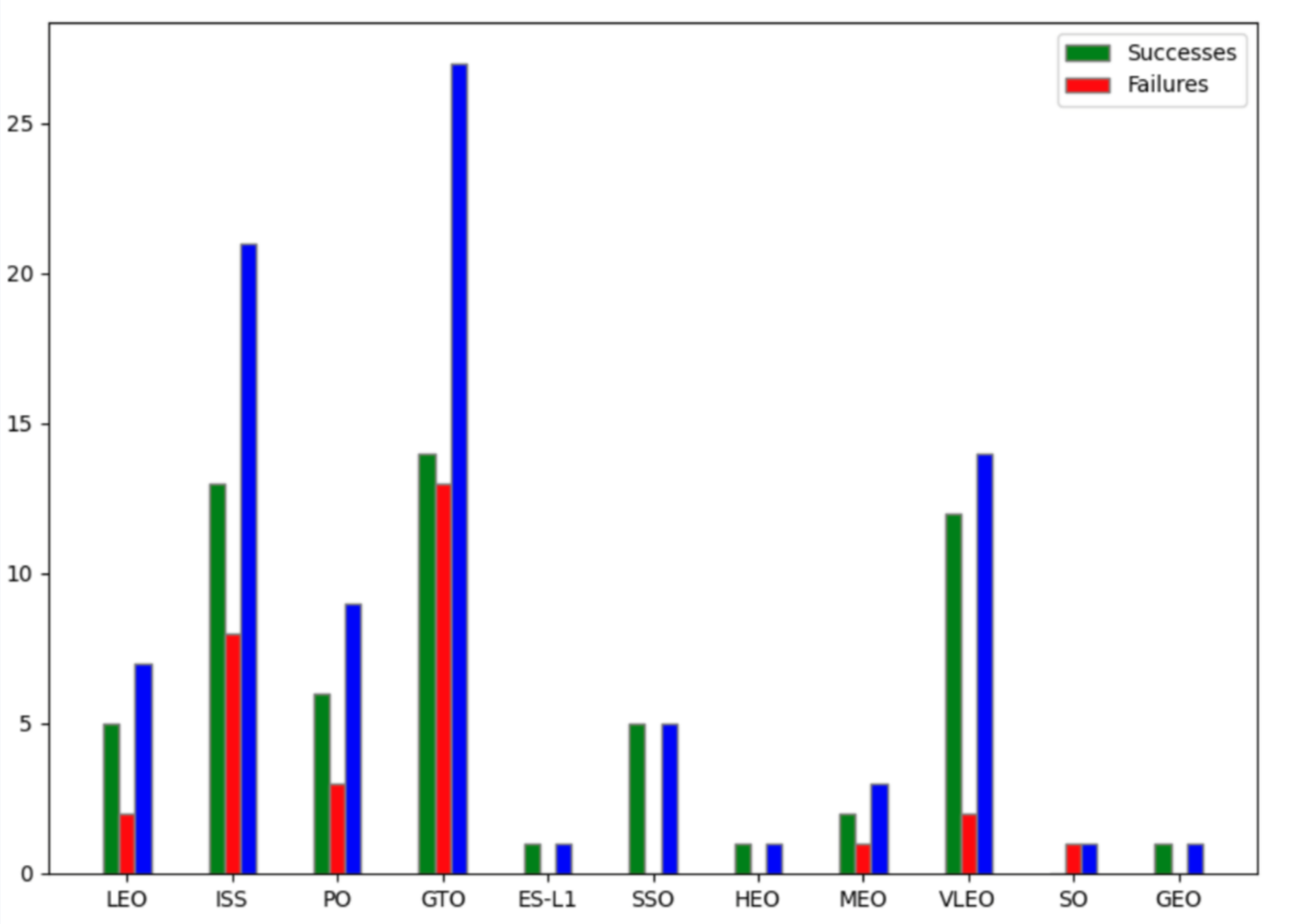
- Payload vs. Launch Site

Each dot is a launch located at the intersection of payload and launch site.



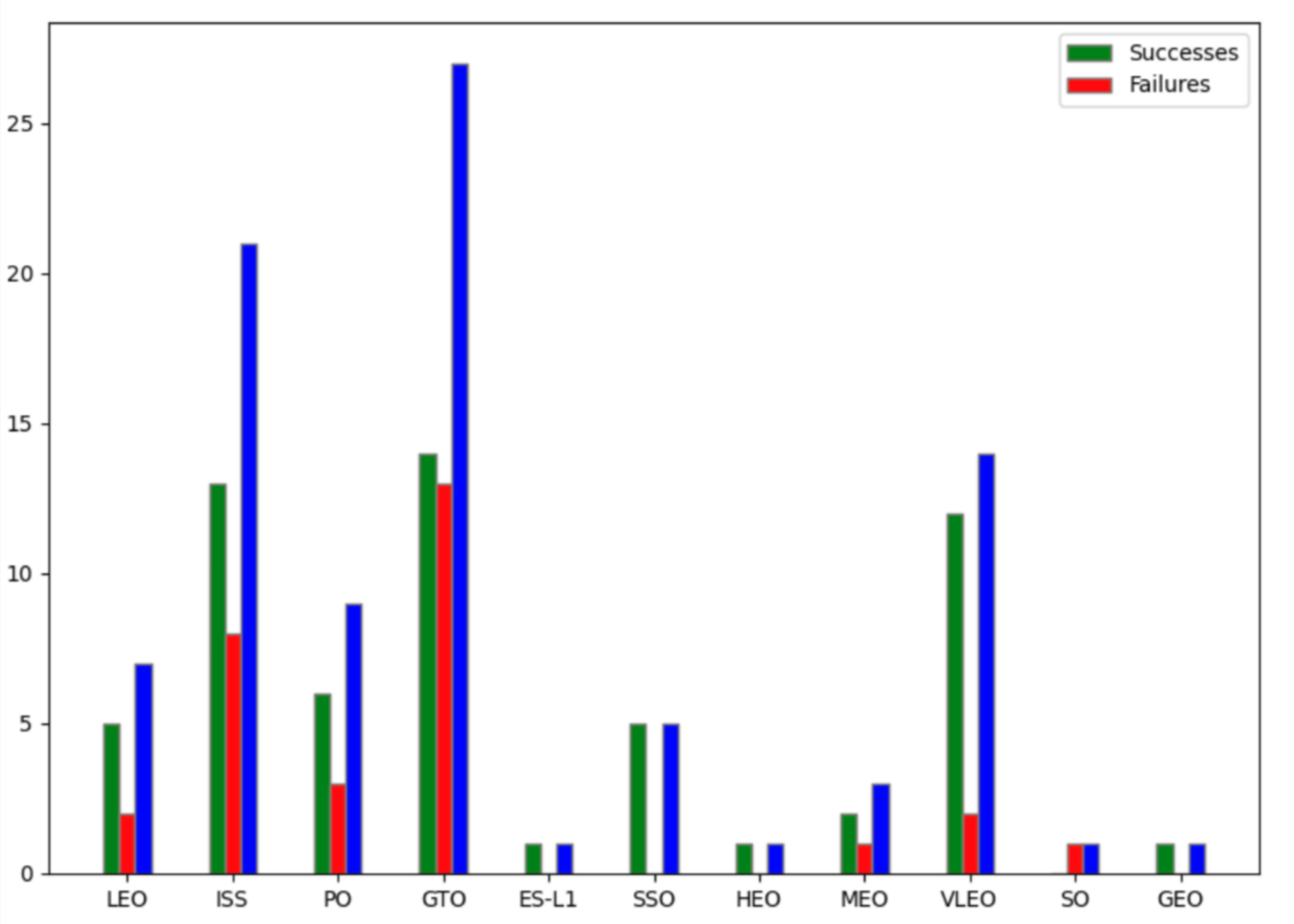
Success Rate vs. Orbit Type

- Green=Success
- Red=Failure
- Blue=Total



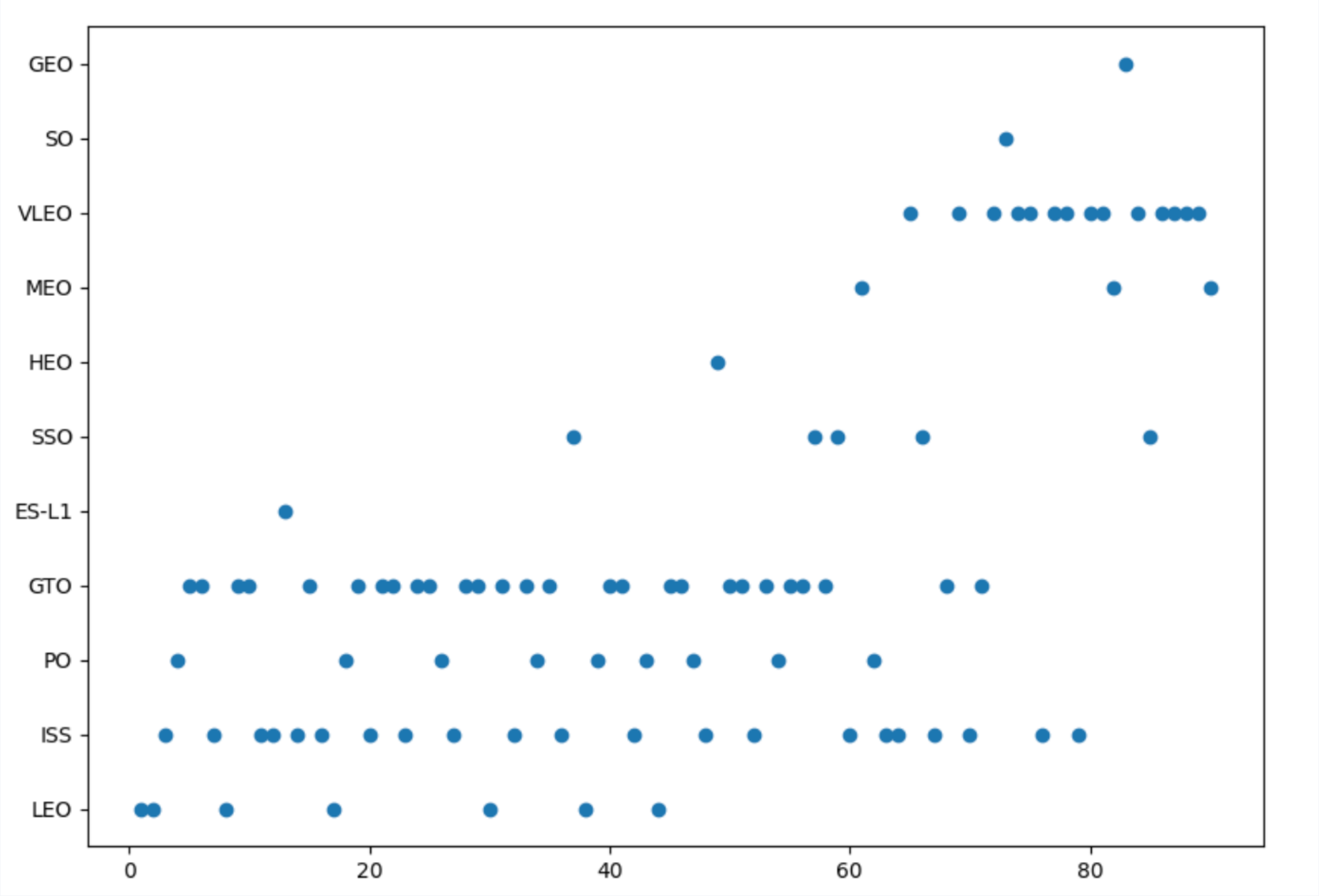
Success Rate vs. Orbit Type

- GTO: 0.518
- ISS: 0.619
- PO: 0.666
- LEO: 0.714
- VLEO:0.857



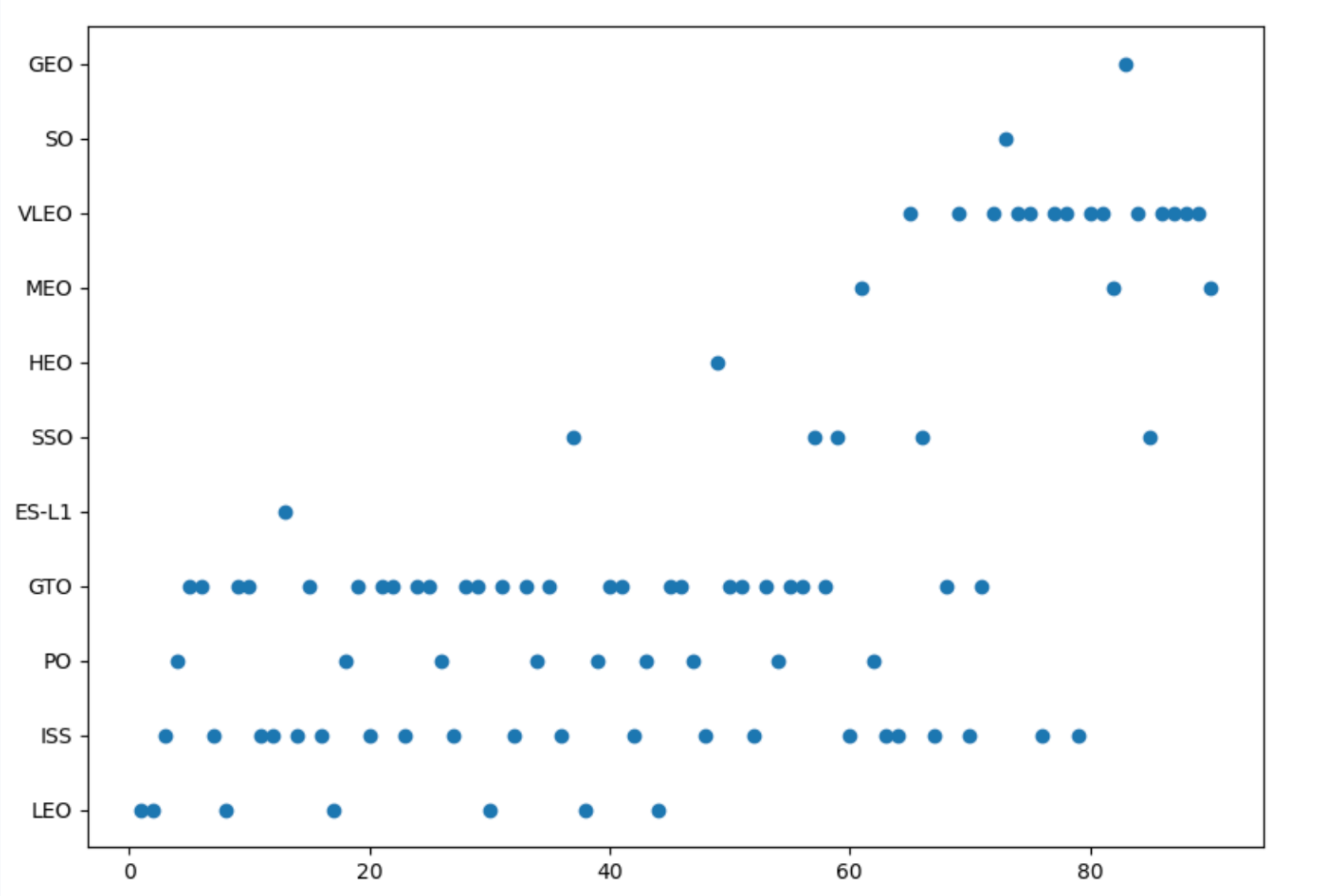
Flight Number vs. Orbit Type

- VLEO missions have been recent and boast the highest success rate.



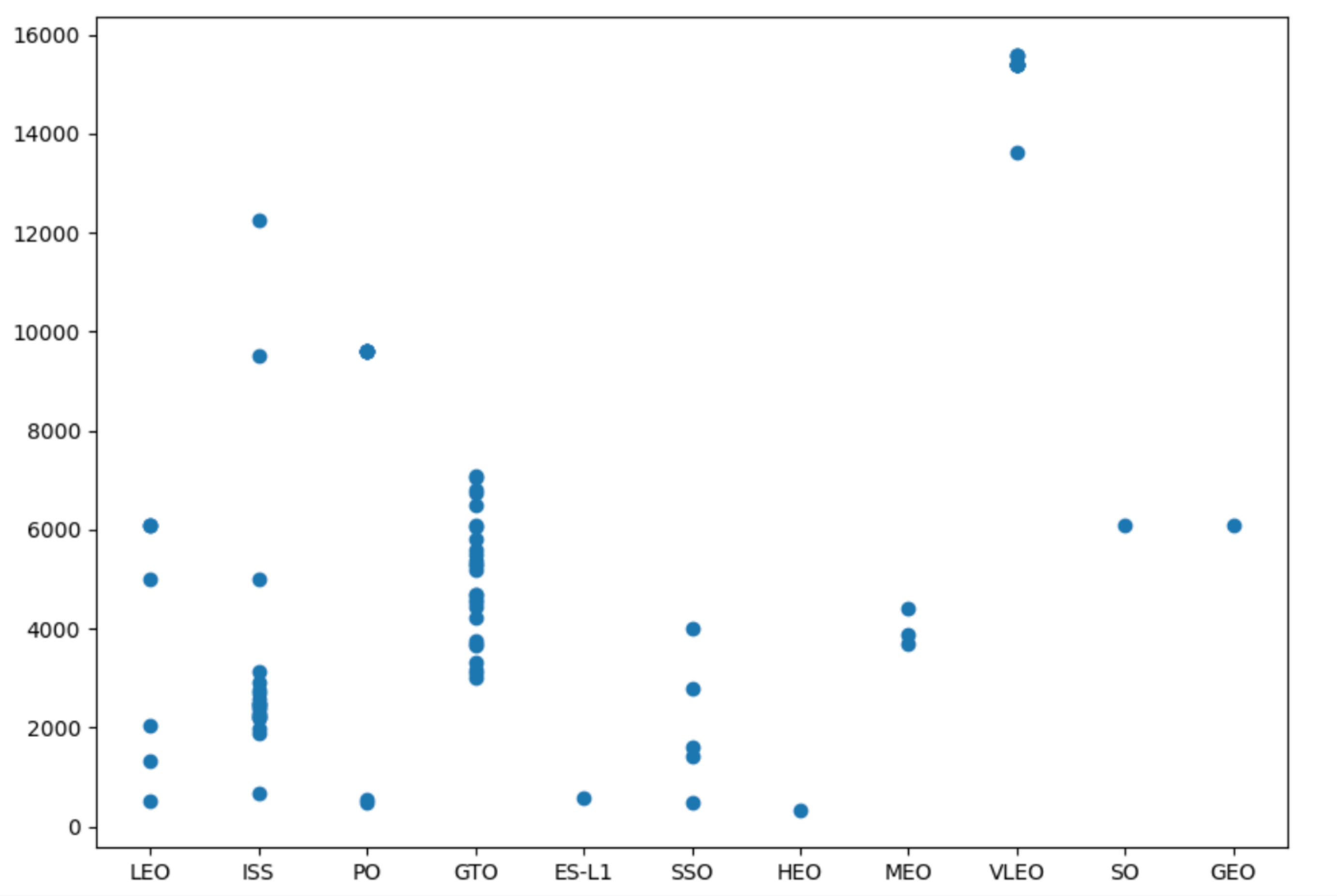
Flight Number vs. Orbit Type

- LEO have not been recent, yet they have the second highest success rate.



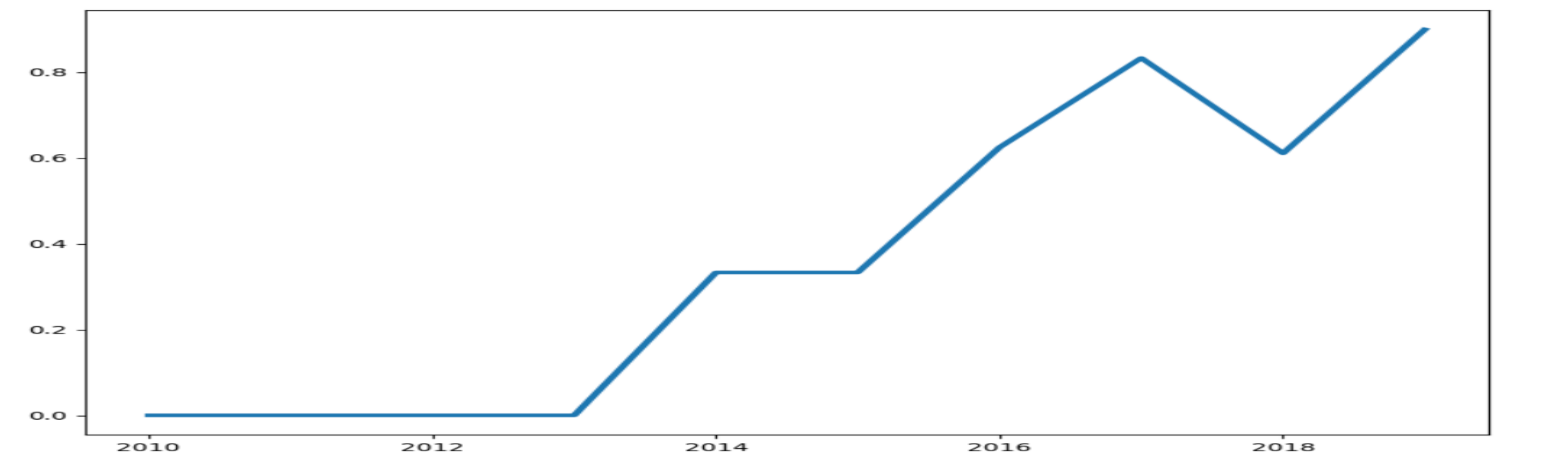
Payload vs. Orbit Type

- Most Mission types have a narrow range of payload mass, the exception being ISS.



Launch Success Yearly Trend

- There is a positive correlation between years and success rate.



All Launch Site Names

```
Select DISTINCT(launch_site) FROM SPACEXTBL
```

- All launch sites used by SpaceX

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;
```

- Five records where launch sites begin with `CCA`

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 LIKE "%(CRS)%"
```

- Total payload carried by boosters from NASA(CRS).

SUM(PAYLOAD_MASS__KG_)
48213

Average Payload Mass by F9 v1.1

```sql
SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTBL
WHERE Booster_Version LIKE '%F9 v1.1%';
```

- Calculated the average payload mass carried by booster version F9 v1.1

AVG(PAYLOAD_MASS__KG_)
2534.6666666666665

First Successful Ground Landing Date

```
SELECT DATE, [Landing _Outcome] FROM SPACEXTBL
where (substr(DATE,7)||substr(DATE,4,2)||substr(DATE,1,2)
      between '20100604' and '20171130') AND [Landing _Outcome]  LIKE "%Success (ground pad)%"
      LIMIT 1;
```

- The first successful landing outcome on the ground pad

Date	Landing _Outcome
22-12-2015	Success (ground pad)

Successful Drone Ship Landing with Payload between 4000 and 6000

```sql
%%sql
SELECT Booster_Version FROM SPACEXTBL
Where PAYLOAD_MASS__KG_ >4000 AND PAYLOAD_MASS__KG_ <6000
AND"Landing _Outcome" LIKE '%Success (drone ship)%';
```

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

```
SELECT Mission_Outcome,COUNT(*) FROM SPACEXTBL Group by Mission_Outcome;
```

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

- Total number of successful and failure mission outcomes

Boosters Carried Maximum Payload

```
SELECT Distinct(Booster_version),(PAYLOAD_MASS__KG_) as "Payload" FROM SPACEXTBL
Where PAYLOAD_MASS__KG_=(Select MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL);
```

- Names of the booster which have carried the maximum payload mass

Booster_Version	Payload
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

```
SELECT CASE cast(SUBSTR(DATE,4,2) as integer)
        WHEN "01" THEN "January"
        WHEN "02" THEN "February"
        WHEN "03" THEN "March"
        WHEN "04" THEN "April"
        WHEN "05" THEN "May"
        WHEN "06" THEN "June"
        WHEN "07" THEN "July"
        WHEN "08" THEN "August"
        WHEN "09" THEN "September"
        WHEN "10" THEN "October"
        WHEN "11" THEN "November"
        WHEN "12" THEN "December"
    END as "month",
    Date,Launch_site, Booster_Version,[Landing _Outcome]
    FROM SPACEXTBL
    WHERE [Landing _Outcome] = "Failure (drone ship)" AND substr(Date,7,4)='2015'
```

month	Date	Launch_Site	Booster_Version	Landing _Outcome
January	10-01-2015	CCAFS LC-40	F9 v1.1 B1012	Failure (drone ship)
April	14-04-2015	CCAFS LC-40	F9 v1.1 B1015	Failure (drone ship)

- The failed landing outcomes in drone ship, their booster versions, and launch site names for in year 2015.

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

```
SELECT [Landing _Outcome],COUNT([Landing _Outcome]) "Missions", RANK() OVER( ORDER BY COUNT([Landing _Outcome]) DESC) Rank FROM SPACEXTBL
where (substr(DATE,7)||substr(DATE,4,2)||substr(DATE,1,2)
    between '20100604' and '20171130') AND [Landing _Outcome] LIKE "%Success%"
Group by [Landing _Outcome];
```

- Ranked 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.

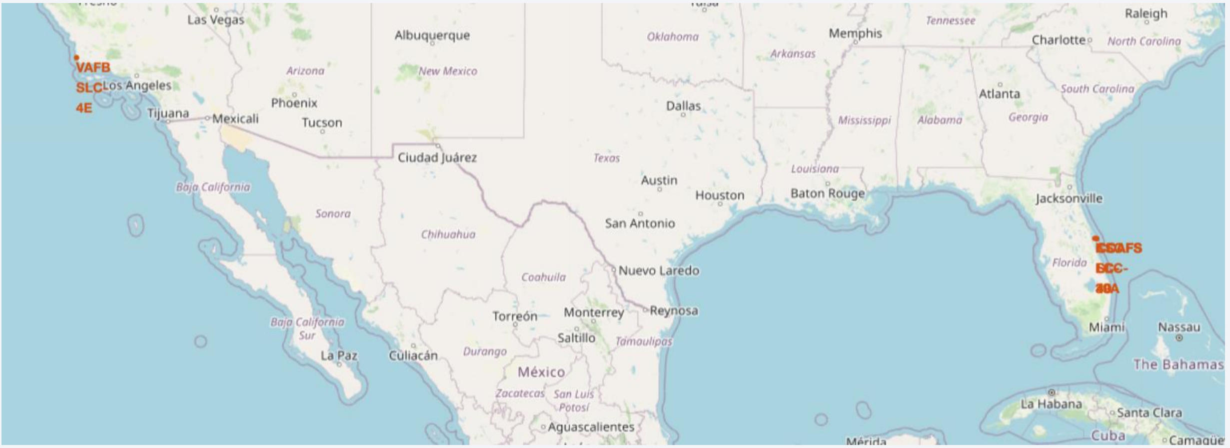
Landing _Outcome	Missions	Rank
Success (drone ship)	5	1
Success (ground pad)	3	2

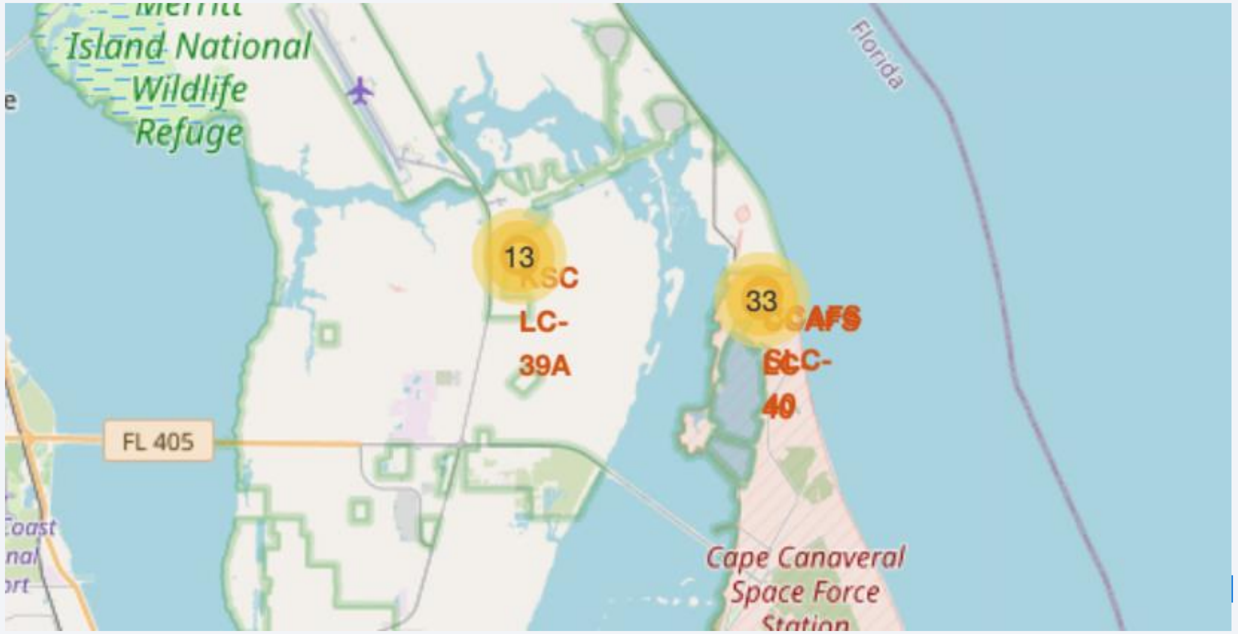
Section 3

Launch Sites Proximities Analysis

Launch Site Locations

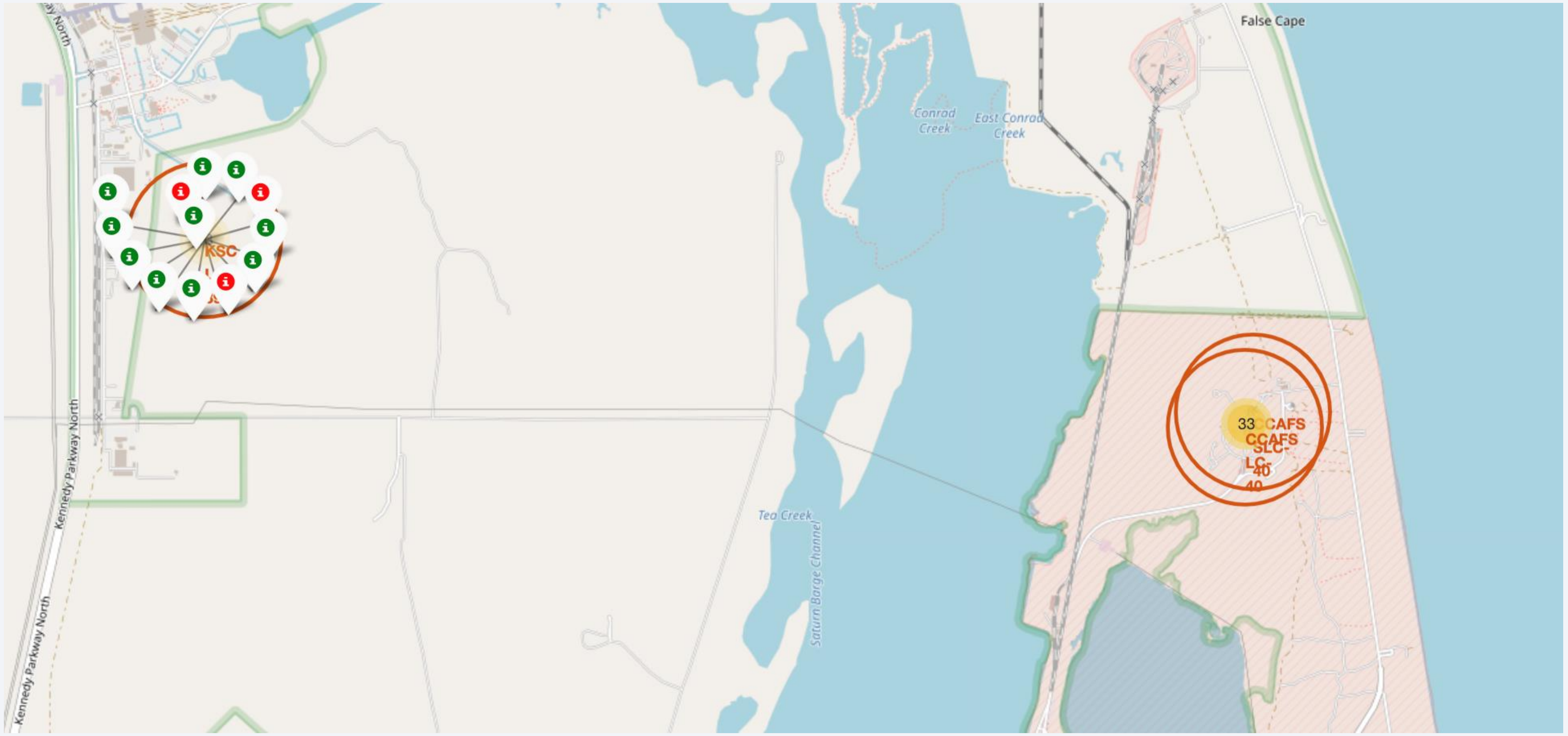
Most launches take place on the East Coast where the rockets can launch eastward to take advantage of the earth's spin while avoiding population centers.





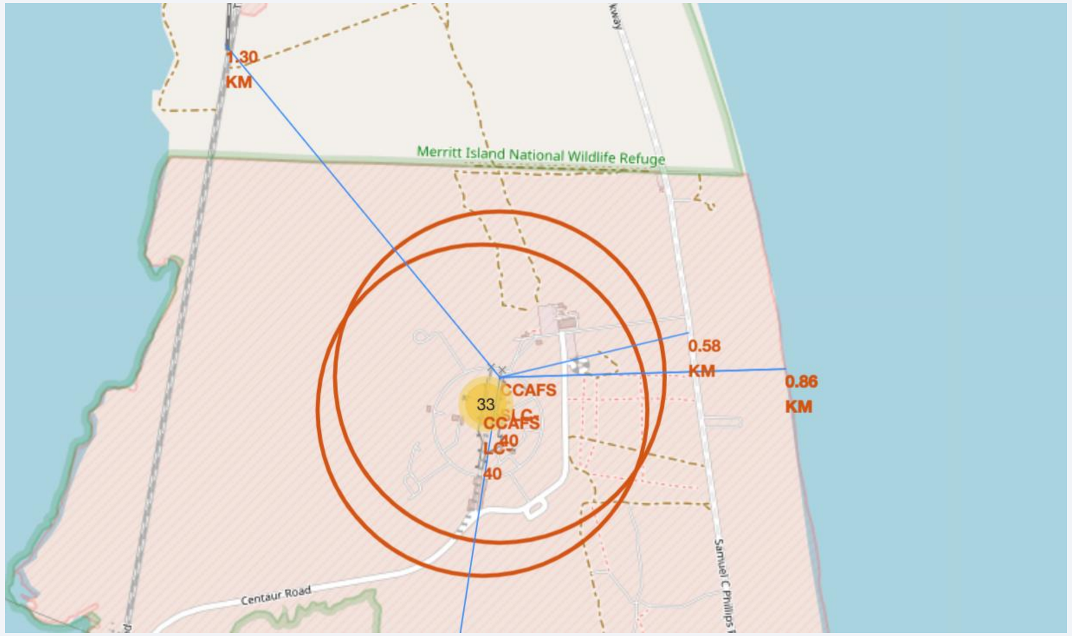
Launch Markers in Florida

- Markers indicate the number of Launches, color indicates the success or failure.



Key Infrastructure

- Proximities such as railway, highway, coastline, with distance calculated and displayed

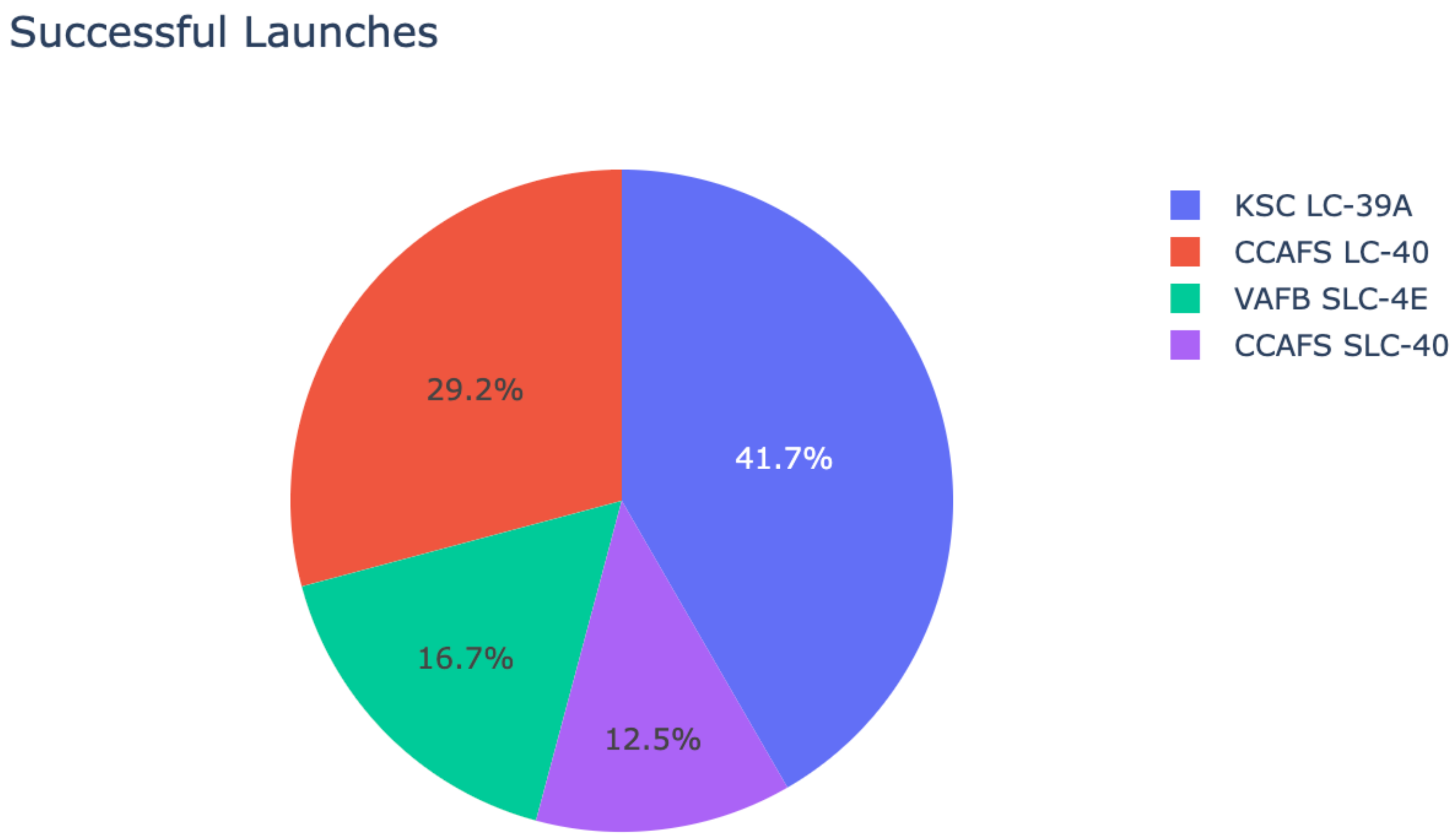


Section 4

Build a Dashboard with Plotly Dash

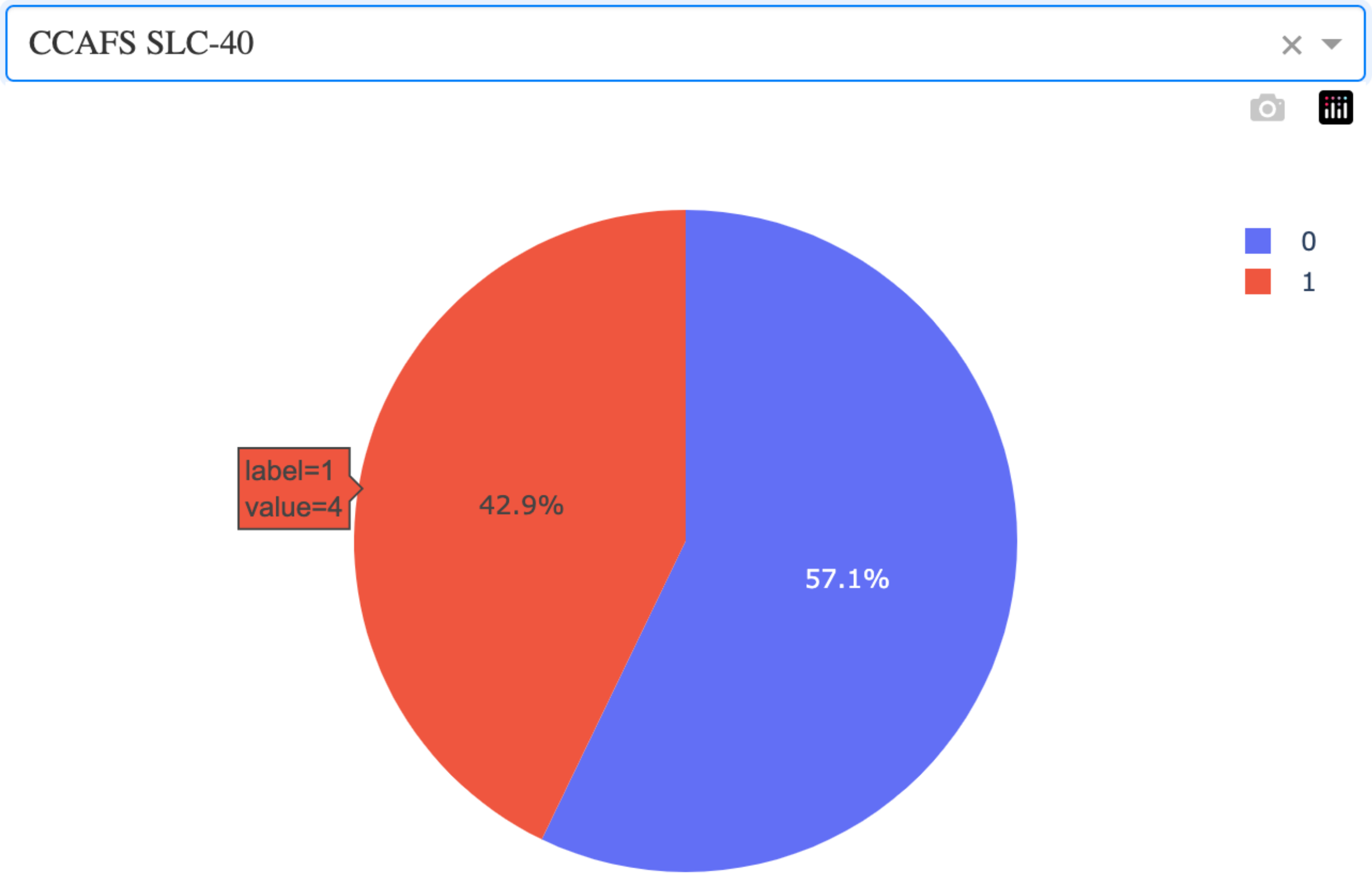
Total Successful Launches

- Facilities on the East coast, particularly KSC LC-39A, are preferred.



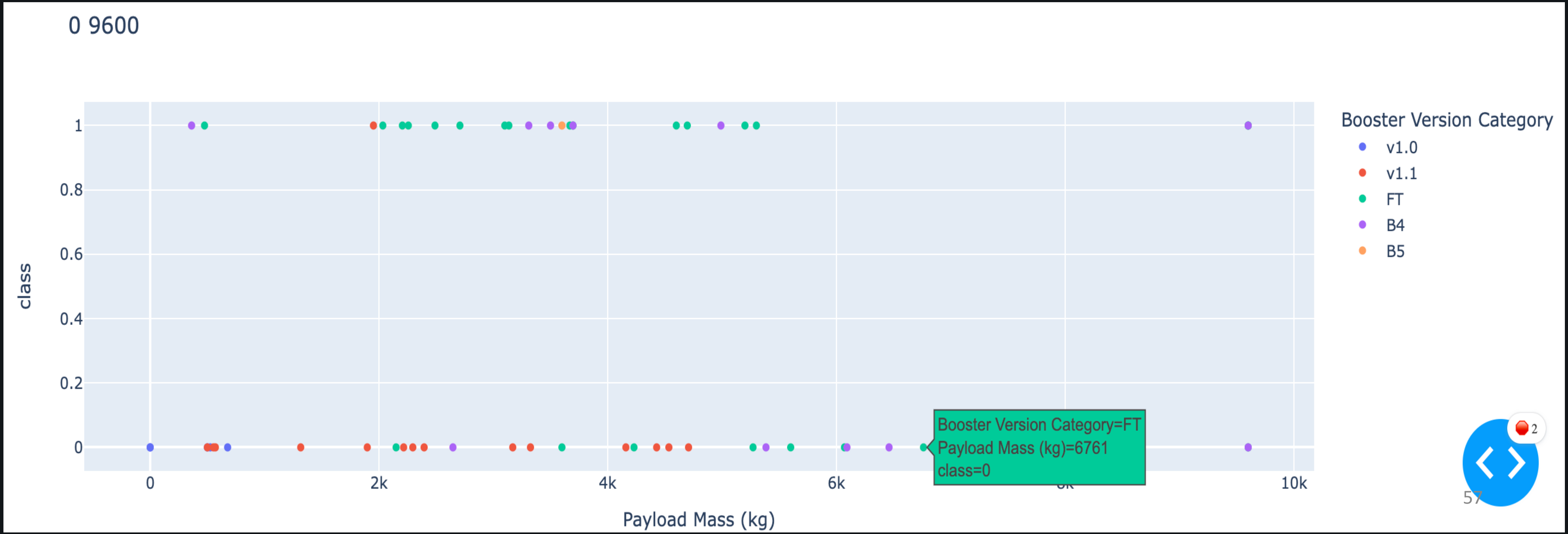
Total Launches per Site

- The best launch success rates are at CCAFS SLC-40 .



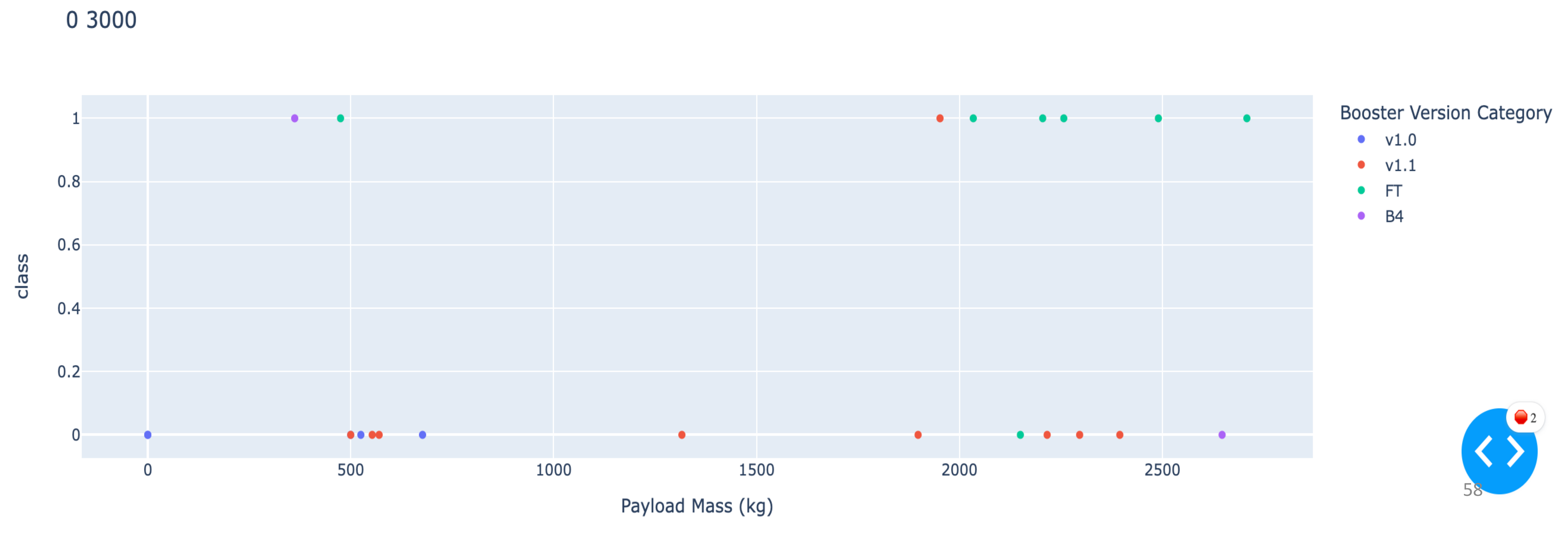
Scatterplots

- Most successful launches have been with Booster Version FT
- Filter 0 – 9600 kg



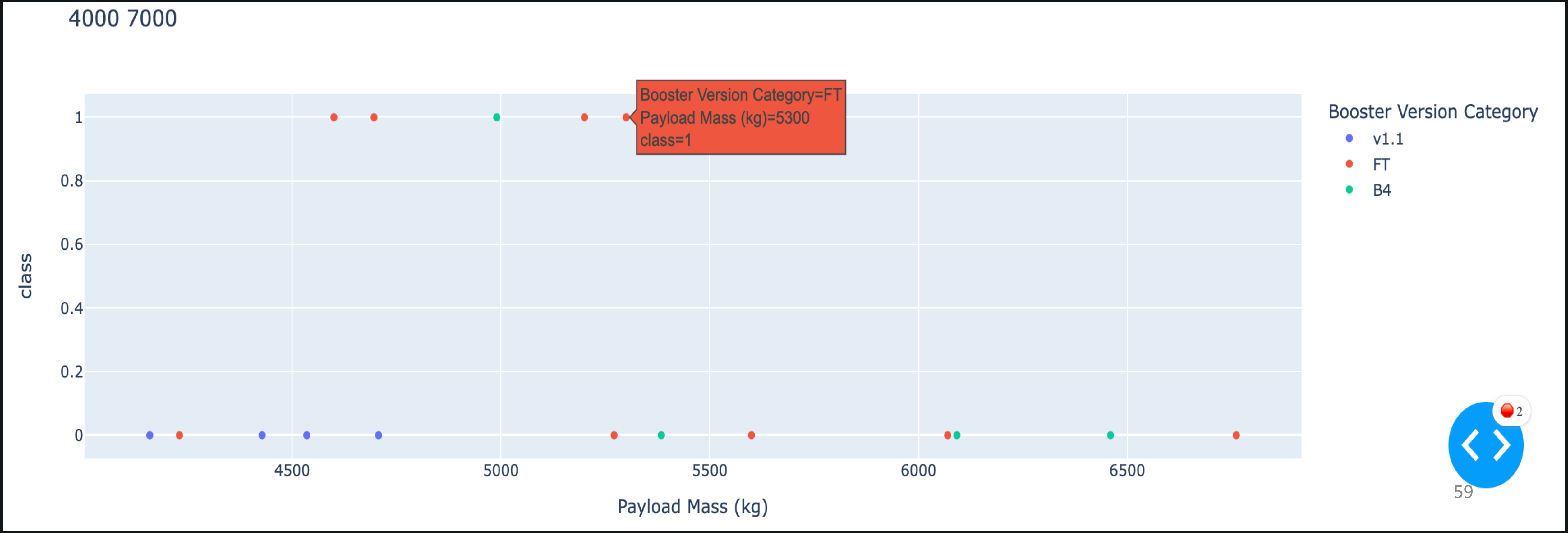
Scatterplots

- Most FT flights are between 2000kg and 4000kg
- Filter 0 – 3000 kg



Scatterplots

- Least successful payload range for landing.
- Filter 4000 – 7000 kg



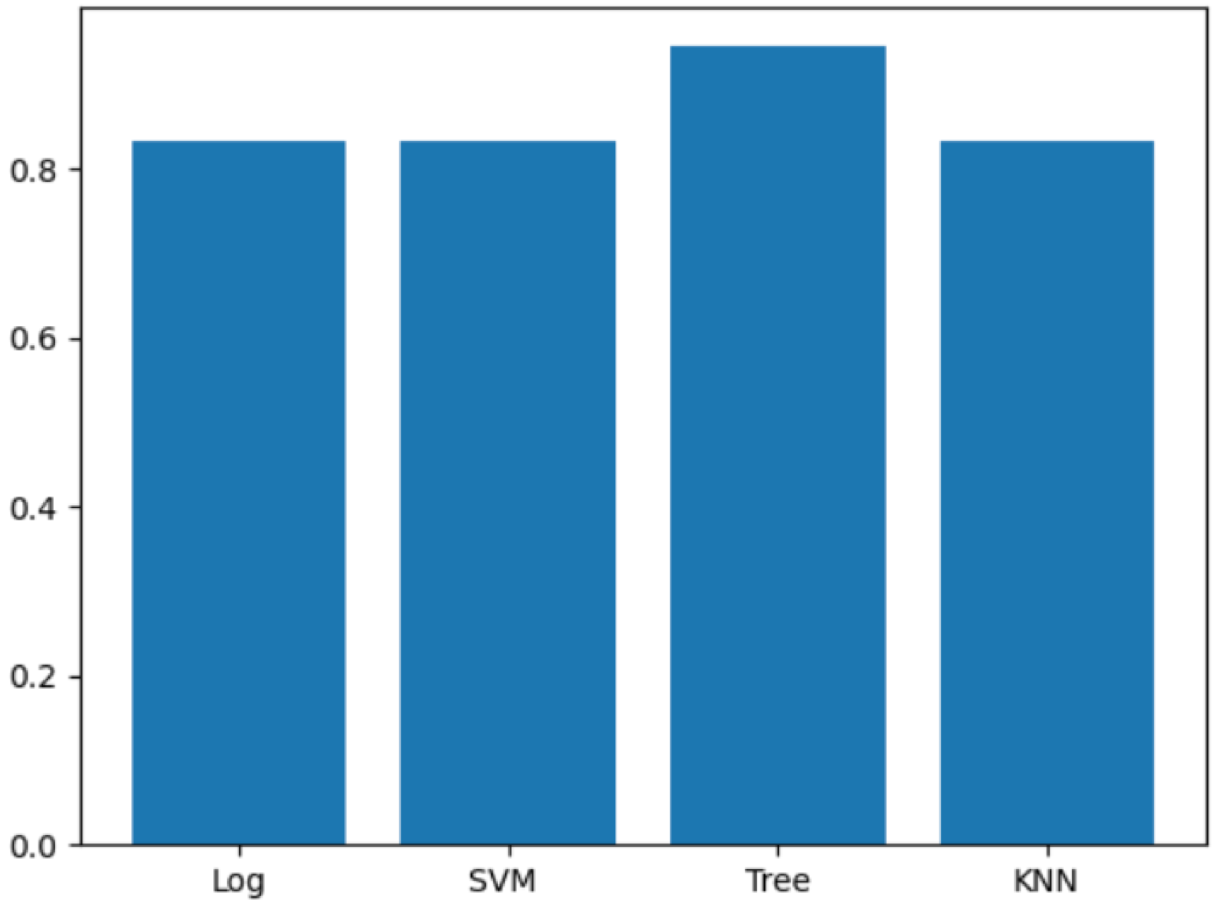
Section 5

Predictive Analysis (Classification)

Accuracy of Models

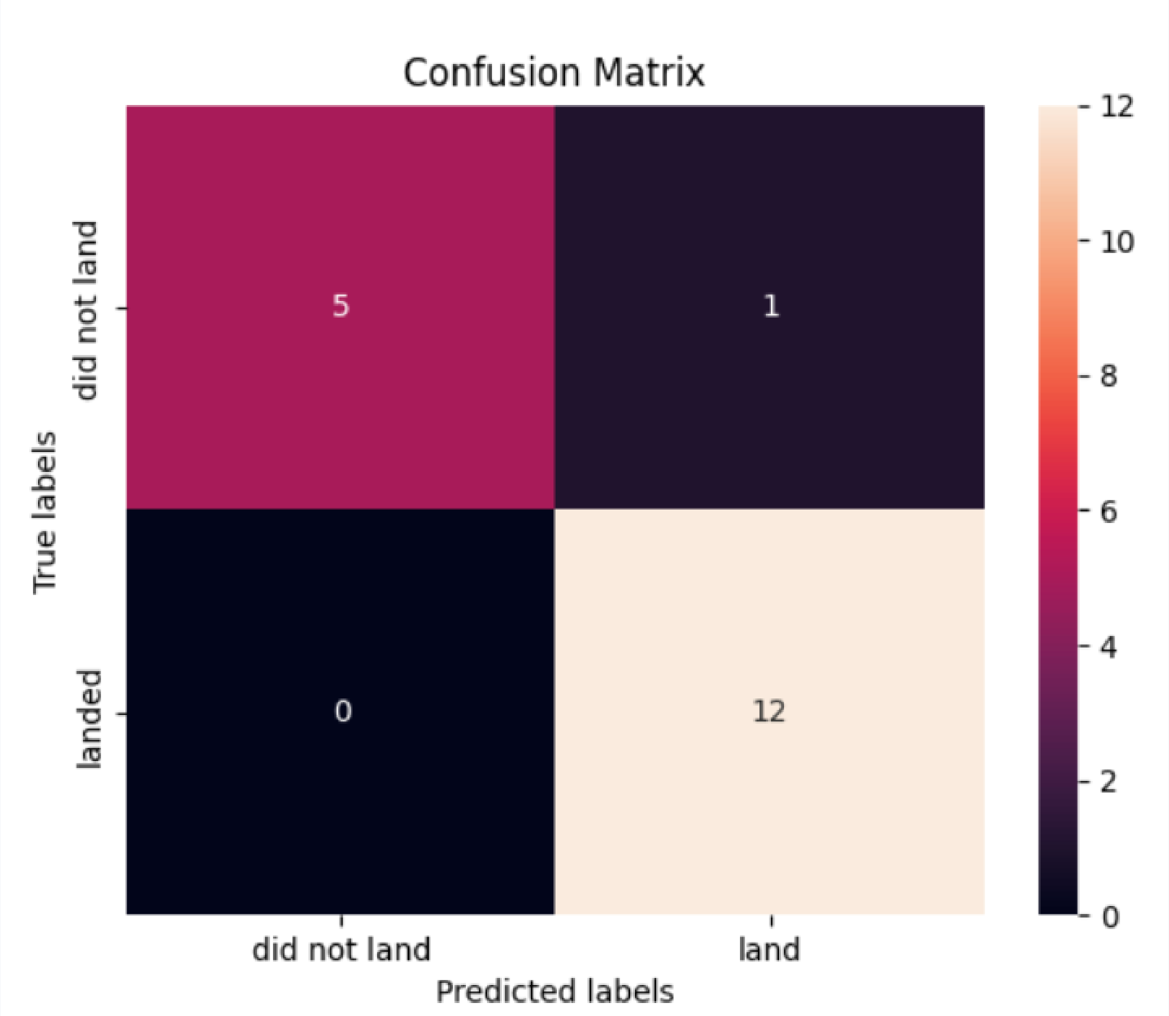
- Uniformity is common when running test.*

*Tree models vary significantly 60-95% between runs.



Confusion Matrix

- Tree model greatly varies between runs (60%-95%) this could be because it does not use any distance formula for calculating error.



Conclusions

- Positive relationship between time and launch success rates.
- Negative relationship between height of the orbit and chance of success.
- Newer booster versions have a higher success rate.
- There is not enough data for us to conclude which categorical model would be best.

Appendix

Webscrapping: https://github.com/GitCraigRash/SpaceX_Presentation/blob/main/jupyter-labs-webscraping.md

Machine Learning Models: https://github.com/GitCraigRash/SpaceX_Presentation/blob/main/IBM-DS0321EN-SkillsNetwork_labs_module_4_SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite%20(1)%20(1).ipynb

EDA with Visualizationhttps://github.com/GitCraigRash/ThirdGit/blob/main/IBM-DS0321EN-SkillsNetwork_labs_module_2_jupyter-labs-eda-dataviz.ipynb.jupyterlite%20(1).ipynb

Plotly Dash: https://github.com/GitCraigRash/SpaceX_Presentation/blob/main/Plotly_Dash_Code

Folium:
https://github.com/GitCraigRash/SpaceX_Presentation/blob/main/2020.02.04%2CSpaceX_Presentation%2CFolium.md

Data Collection:
https://github.com/GitCraigRash/SpaceX_Presentation/blob/main/2023.02.04.SpaceX_Presentation%2CData_Collection.md

Thank you!