IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Nura Arifin Wong
25 October 2025

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:
  - Data can be collected from web API or web scraping of wiki page. For downstream analyses, data used was from Coursera provided csv files
- Perform data wrangling
  - Some parameters were converted to one-hot-encoding style (i.e., categorical variables were converted binary)
- 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

- **Method 1:** SpaceX API
  (https://api.spacexdata.com/v4)
- **Method 2:** Web scrapped Wiki page
  (https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922)

Flight No.	Version,Booster[b]	Launch site	Payload[c]	Payload mass	Orbit	Customer	Launchoutcome	Boosterlanding	1	...	4	5	6	7	Payload	Launch outcome	Version Booster	Booster landing	Date	Time
1	NaN	CCAFS	NaN	0	LEO	SpaceX	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	Dragon Spacecraft Qualification Unit	Success\n	F9 v1.07B0003.18	Failure	4 June 2010	18:45
2	NaN	CCAFS	NaN	0	LEO	NASA	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	Dragon	Success	F9 v1.07B0004.18	Failure	8 December 2010	15:43
3	NaN	CCAFS	NaN	525 kg	LEO	NASA	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	Dragon	Success	F9 v1.07B0005.18	No attempt\n	22 May 2012	07:44

Data Collection – SpaceX API

- Coursera provided auxiliary functions to get API data
- Parsing and cleaning json get requests
- GitHub URL: https://github.com/n-arifinwong/coursera_notes/blob/main/capstone/M1_1_DataWebAPI.ipynb

From `rocket`, learn booster name from the rocker column

In [2]:
```
# Takes the dataset and uses the rocket column to call the API and append the data to th
def getBoosterVersion(data):
    for x in data['rocket']:
        if x:
         response = requests.get("https://api.spacexdata.com/v4/rockets/"+str(x)).json()
         BoosterVersion.append(response['name'])
```

From `launchpad`, learn name of launchsite, longitude and latitude

In [3]:
```
# Takes the dataset and uses the launchpad column to call the API and append the data to
def getLaunchSite(data):
    for x in data['launchpad']:
        if x:
         response = requests.get("https://api.spacexdata.com/v4/launchpads/"+str(x)).jso
         Longitude.append(response['longitude'])
         Latitude.append(response['latitude'])
         LaunchSite.append(response['name'])
```

From the `payload`, learn the mass of the payload and its orbit

In [4]:
```
# Takes the dataset and uses the payloads column to call the API and append the data to
def getPayloadData(data):
    for load in data['payloads']:
        if load:
         response = requests.get("https://api.spacexdata.com/v4/payloads/"+load).json()
         PayloadMass.append(response['mass_kg'])
         Orbit.append(response['orbit'])
```

From `cores`, learn landing outcome, number of flights, etc.

In [5]:
```
# Takes the dataset and uses the cores column to call the API and append the data to the
def getCoreData(data):
    for core in data['cores']:
            if core['core'] != None:
                response = requests.get("https://api.spacexdata.com/v4/cores/"+core['cor
                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 Collection - Scraping

- Get request of Falcon9 Wiki page
- Parsing and clean the output using BeautifulSoup
- GitHub URL: https://github.com/n-arifinwong/coursera_notes/blob/main/capstone/M1_1_DataWebAPI.ipynb

Task 1: Request Falcon9 launch Wiki Page

```
# Import from specific date
from datetime import date
specific_date = date(2021,6,9)
params = {
    "date_string" : specific_date.strftime("%Y-%m-%d")
}

response = requests.get(static_url, headers = headers, params = params)
```

```
response.status_code # Status 200 is success
```

```
200
```

```
soup = BeautifulSoup(response.text, 'html.parser')
soup.title
```

```
<title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>
```

Task 2: Extract all column/variable names

```
# Find all table elements
html_tables = soup.find_all('table')

# Let's print the third table and check its content
first_launch_table = html_tables[2]
print(first_launch_table)
```

```
-bracket">]</span></a></sup>
</td>
```

Data Wrangling

- Verified the number of null values:
  `df.isnull().sum()/len(df)*100`
- Turned the landing outcome label into one-hot-encoding format (0 if failure, 1 if success)
- GitHub URL:
  https://github.com/n-arifinwong/coursera_notes/blob/main/capstone/M1_3_dataWrangling.ipynb

Task 4: Create landing outcome label

```
# landing_class = 0 if bad_outcome
# landing_class = 1 otherwise

outcomes = df['Outcome'].isin(bad_outcomes)
landing_class = outcomes
landing_class[outcomes] = 1
landing_class[outcomes == False] = 0
landing_class

```

```
0     1
1     1
2     1
3     1
4     1
     ..
85    0
86    0
87    0
88    0
89    0
Name: Outcome, Length: 90, dtype: object
```

EDA with SQL

- First, explored data using SQLite python magics

- GitHub URL:
  - https://github.com/n-arifinwong/coursera_notes/blob/main/capstone/M2_1_EDA_withSQL.ipynb

For example:

Task 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

```
%%sql

SELECT Landing_Outcome,COUNT(Landing_Outcome) FROM SPACEXTABLE GROUP BY Landing_Outcome
    HAVING Date BETWEEN '2010-06-04' AND '2017-03-20'
    ORDER BY COUNT(Landing_Outcome) DESC

```

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

Landing_Outcome	COUNT(Landing_Outcome)
No attempt	21
Success (drone ship)	14
Success (ground pad)	9
Failure (drone ship)	5
Controlled (ocean)	5
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

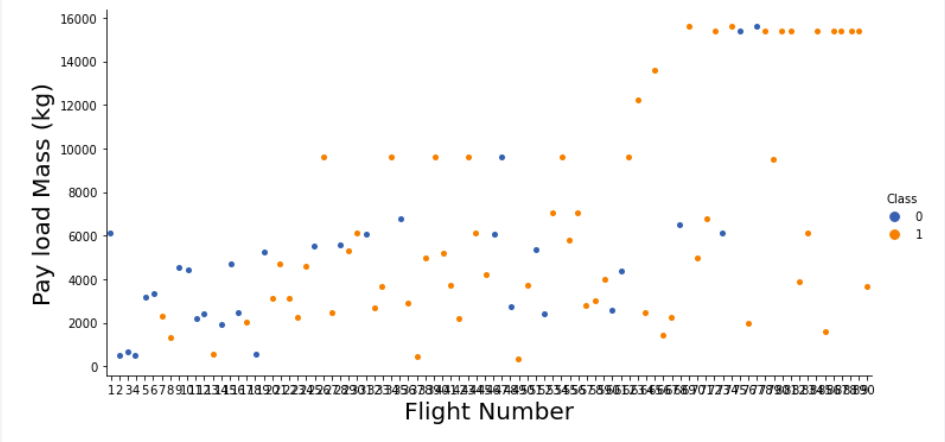
EDA with Data Visualization

- Second, visualized data using matplotlib and seaborne

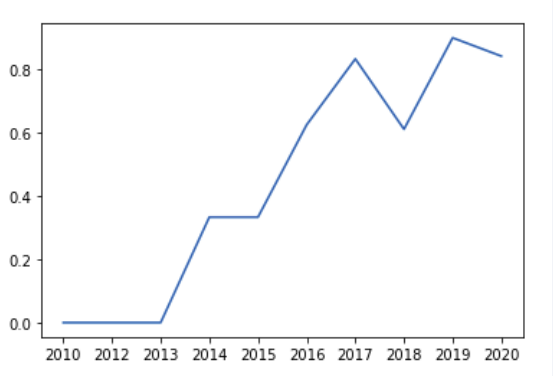
- GitHub URL:
  - https://github.com/n-arifinwong/coursera_notes/blob/main/capstone/M2_2_EDA_Visualization.ipynb

For example:

Pay load mass by flight number, coloured by success/failure



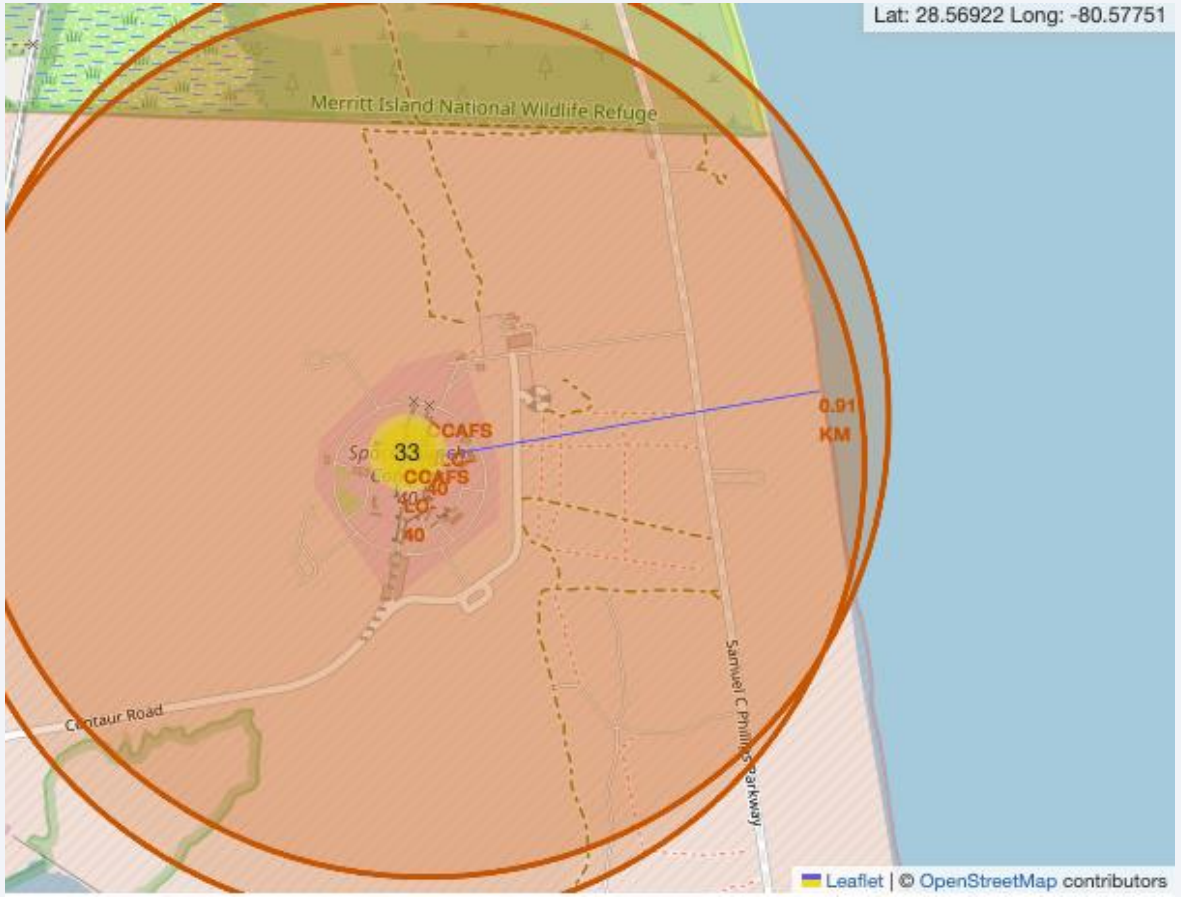
Yearly trend of average launch success:



Build an Interactive Map with Folium

- Used Folium package to visualize launch sites on a map, marked with circles/points
- Success/failures indicated
- Distance to closest coastline drawn using polyline (pictured)

- GitHub URL:
  - https://github.com/n-arifinwong/coursera_notes/blob/main/capstone/M3_1_foliumMaps.ipynb

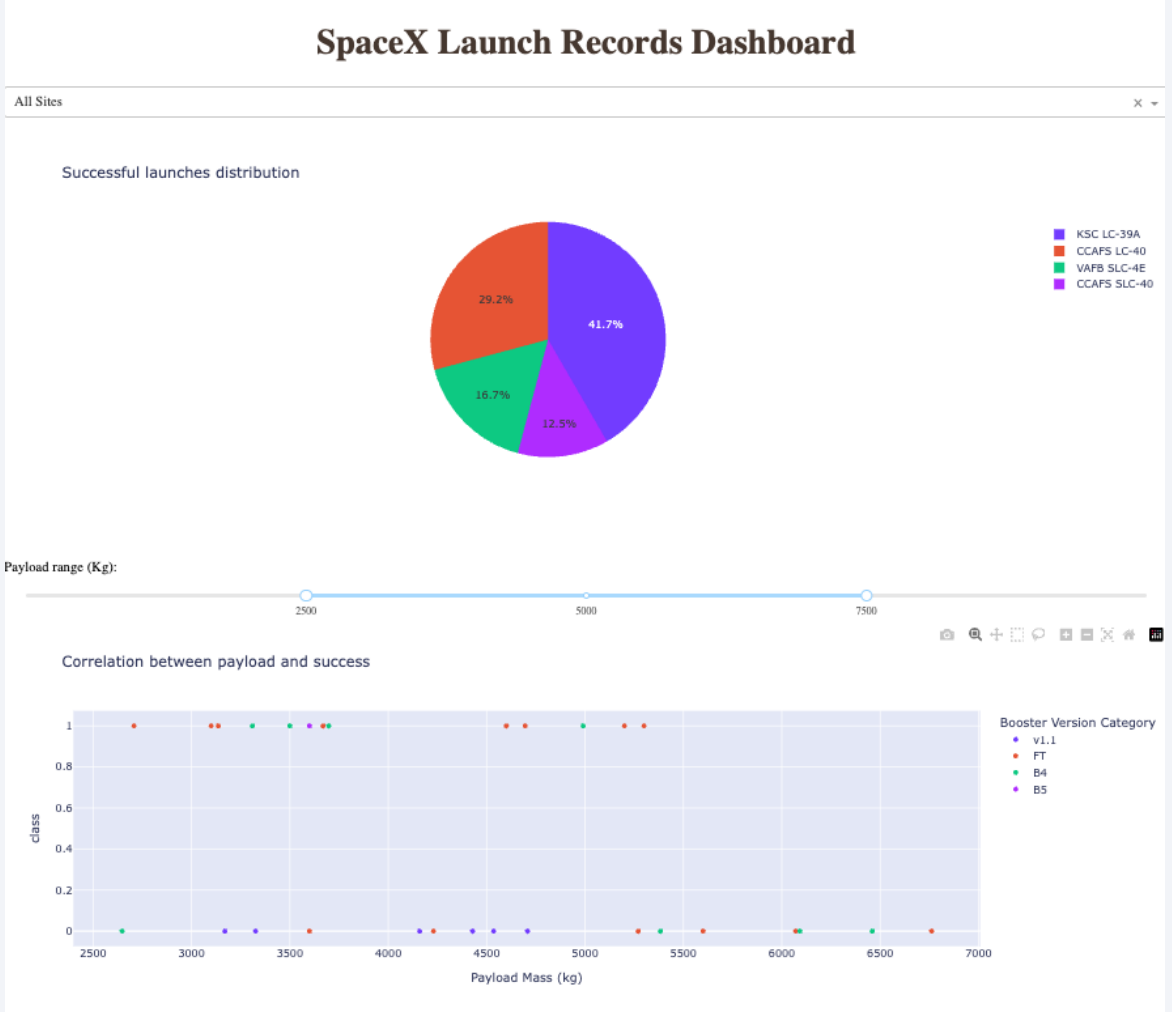


Build a Dashboard with Plotly Dash

- Interactive dashboard to visualize launch successes
- To run:

```
python3.11 spacex-dash-app.py
#Then open html link
```

- GitHub URL:
  https://github.com/n-arifinwong/coursera_notes/blob/main/capstone/1_data/dash/spacex-dash-app.py

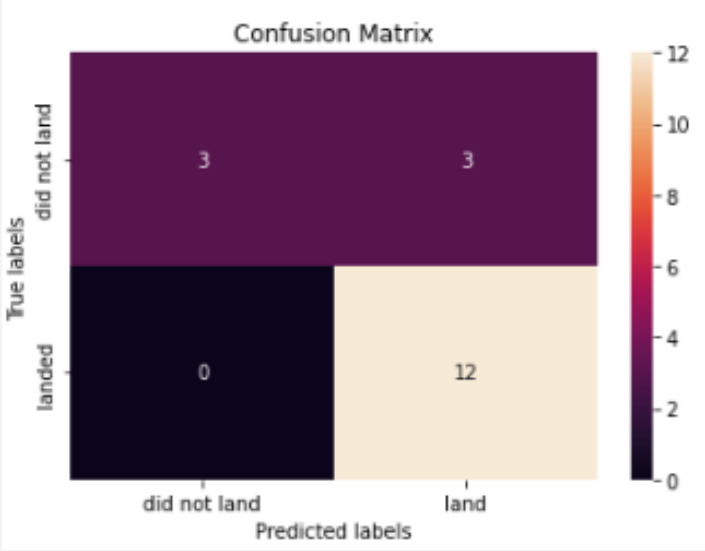


Predictive Analysis (Classification)

- Used GridSearch cross validation for hyperparameter tuning
- Used prediction models: logistic regression, svm, decision tree and KNN
- All models had similar accuracy (with decision tree sometimes performing worse, depending on random seed)

- GitHub URL:
  https://github.com/n-arifinwong/coursera_notes/blob/main/capstone/M4_predictiveAnalysis_classification.ipynb

Confusion matrix output from all models:

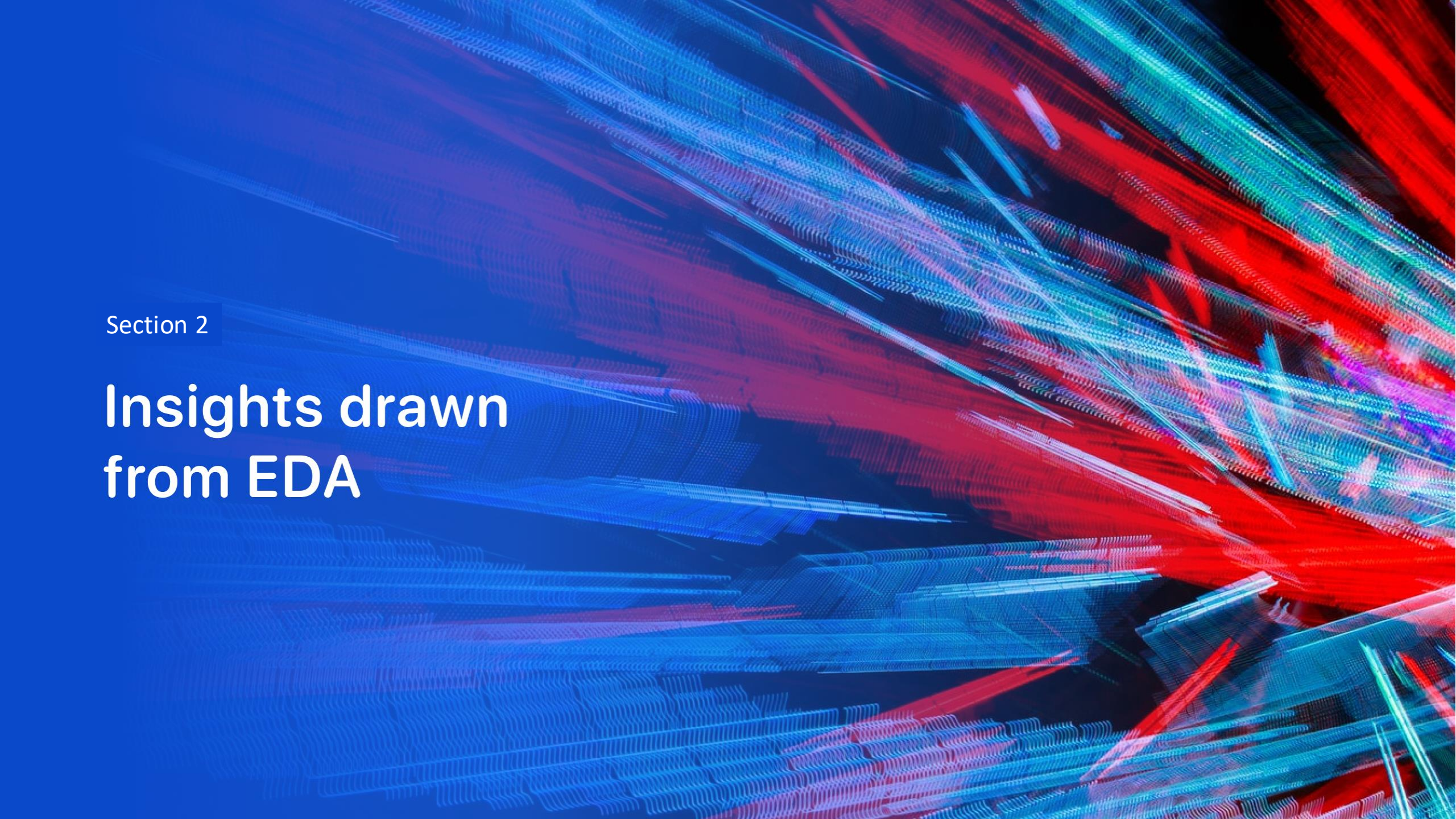


Accuracy score:

	Model	Score
0	logReg	0.833333
1	SVM	0.833333
2	Decision Tree	0.666667
3	KNN	0.833333

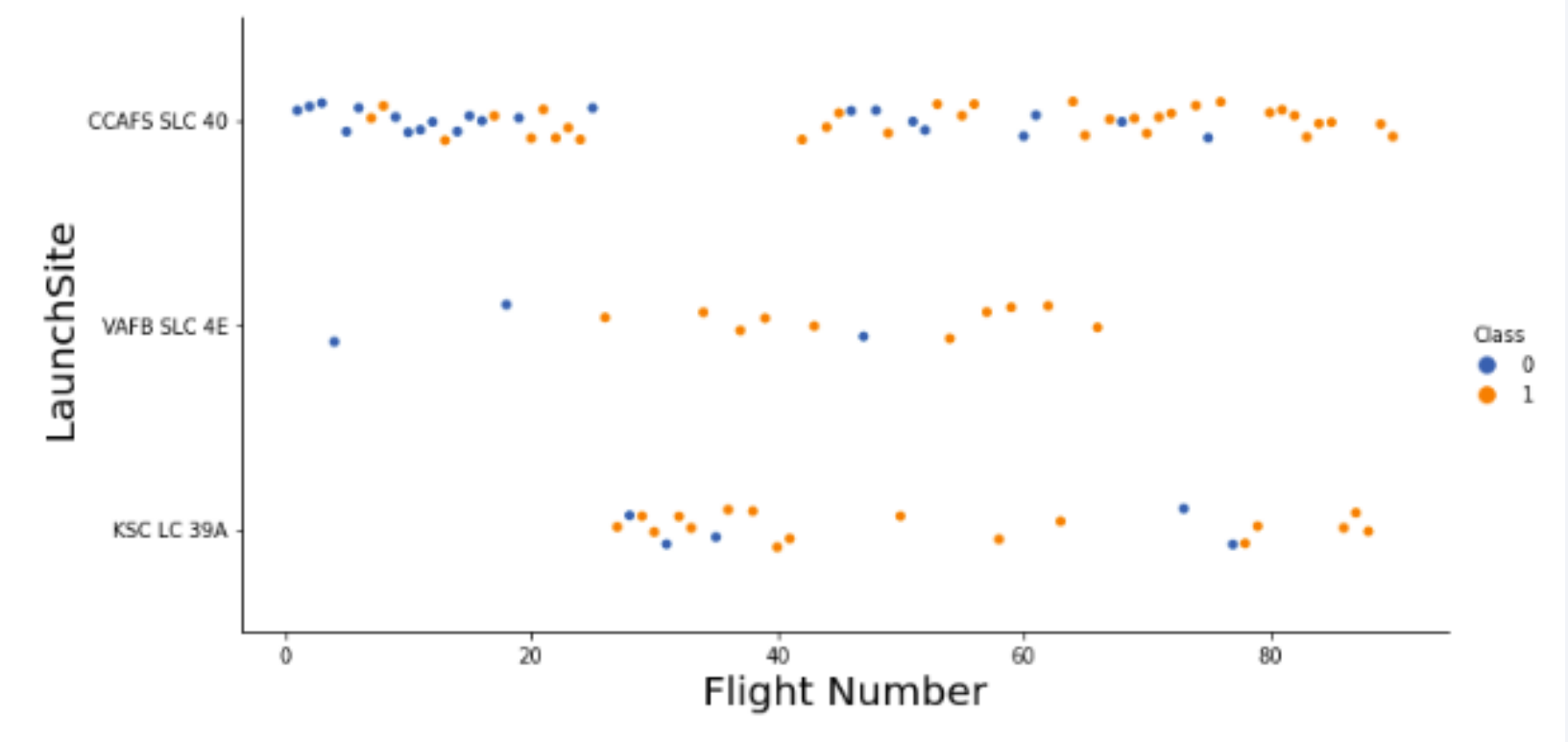
Results

- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results


Section 2
Insights drawn from EDA

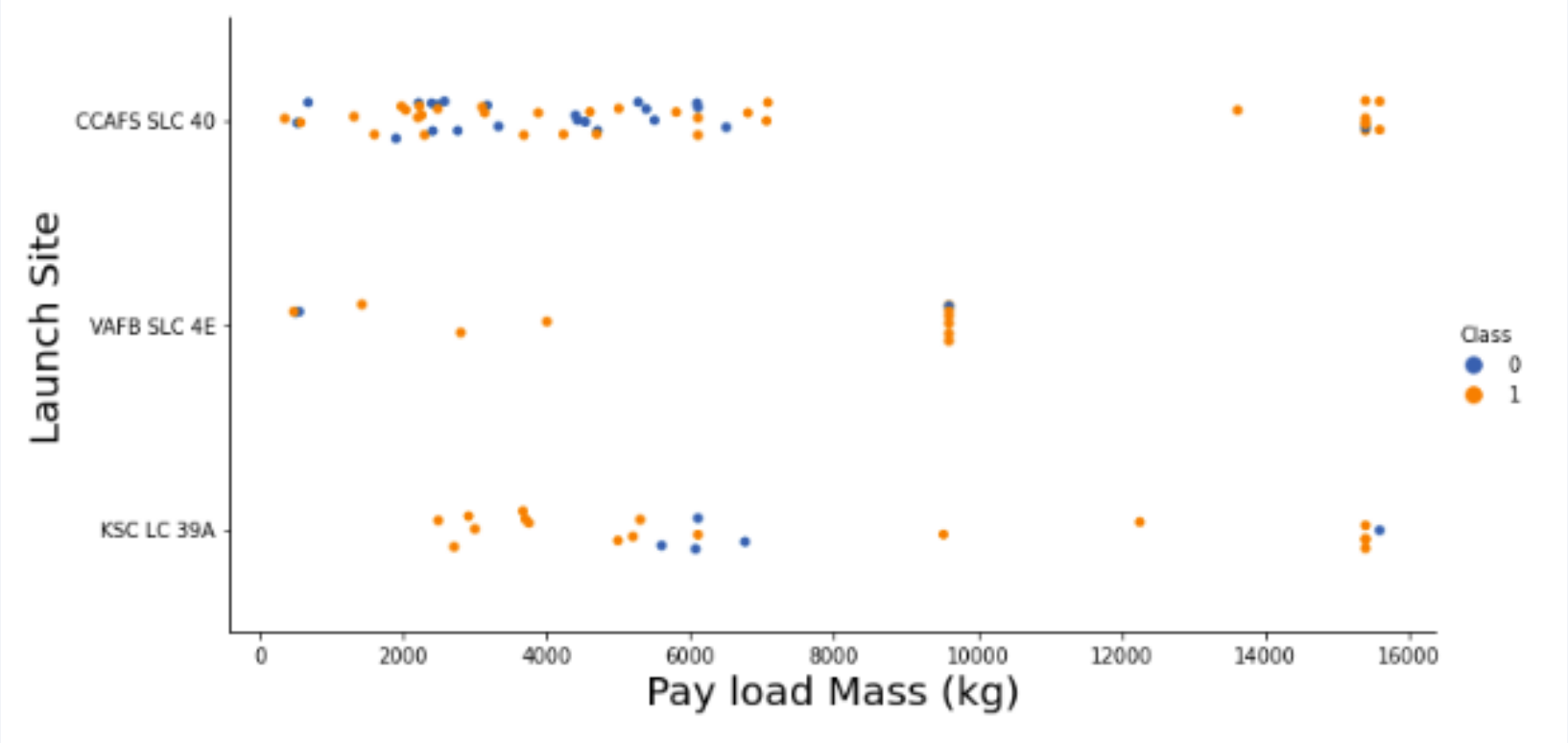
Flight Number vs. Launch Site

- VAFB SLC 4E had very few launches in comparison to other 2 sites
- CCAFS LC 40 Had the first few launches, which looked to have a few failures. But more recent launches were mostly successful



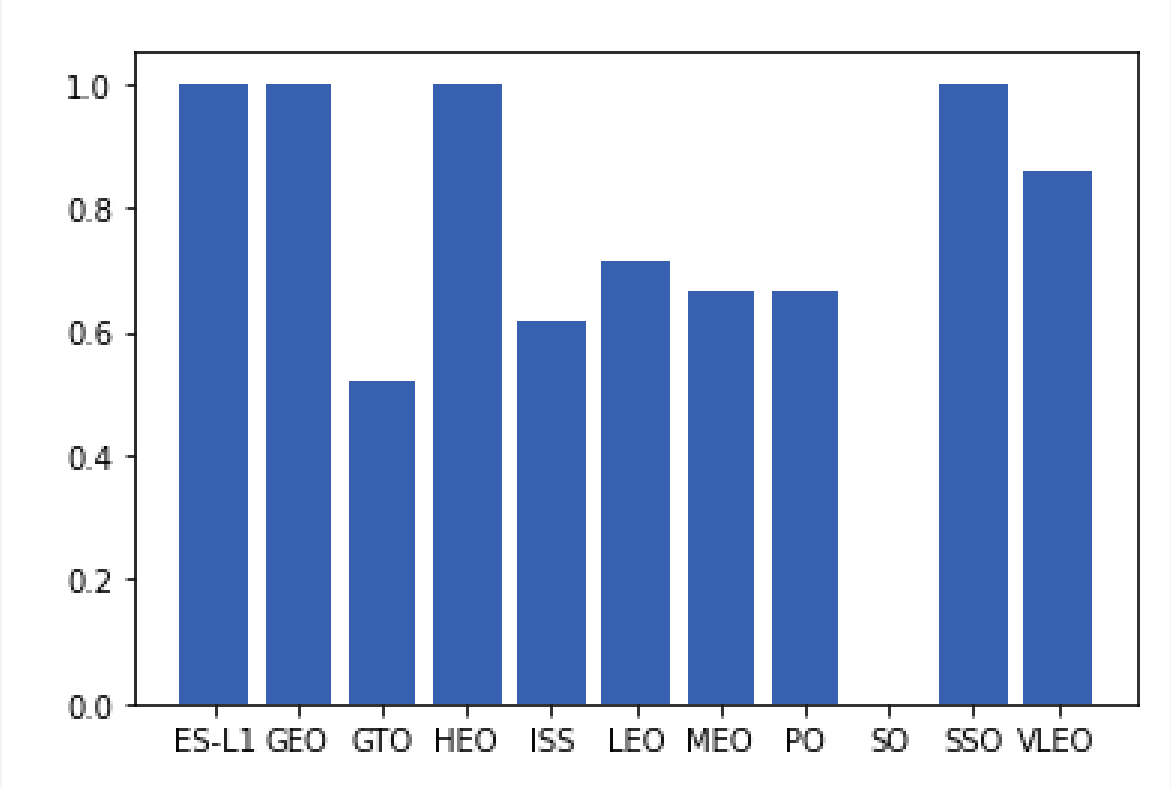
Payload vs. Launch Site

- High pay load mass seems to be rather successful
- But most of the launches have low pay load mass
- None of the launches from site VAFB SLC 4E were high pay load mass



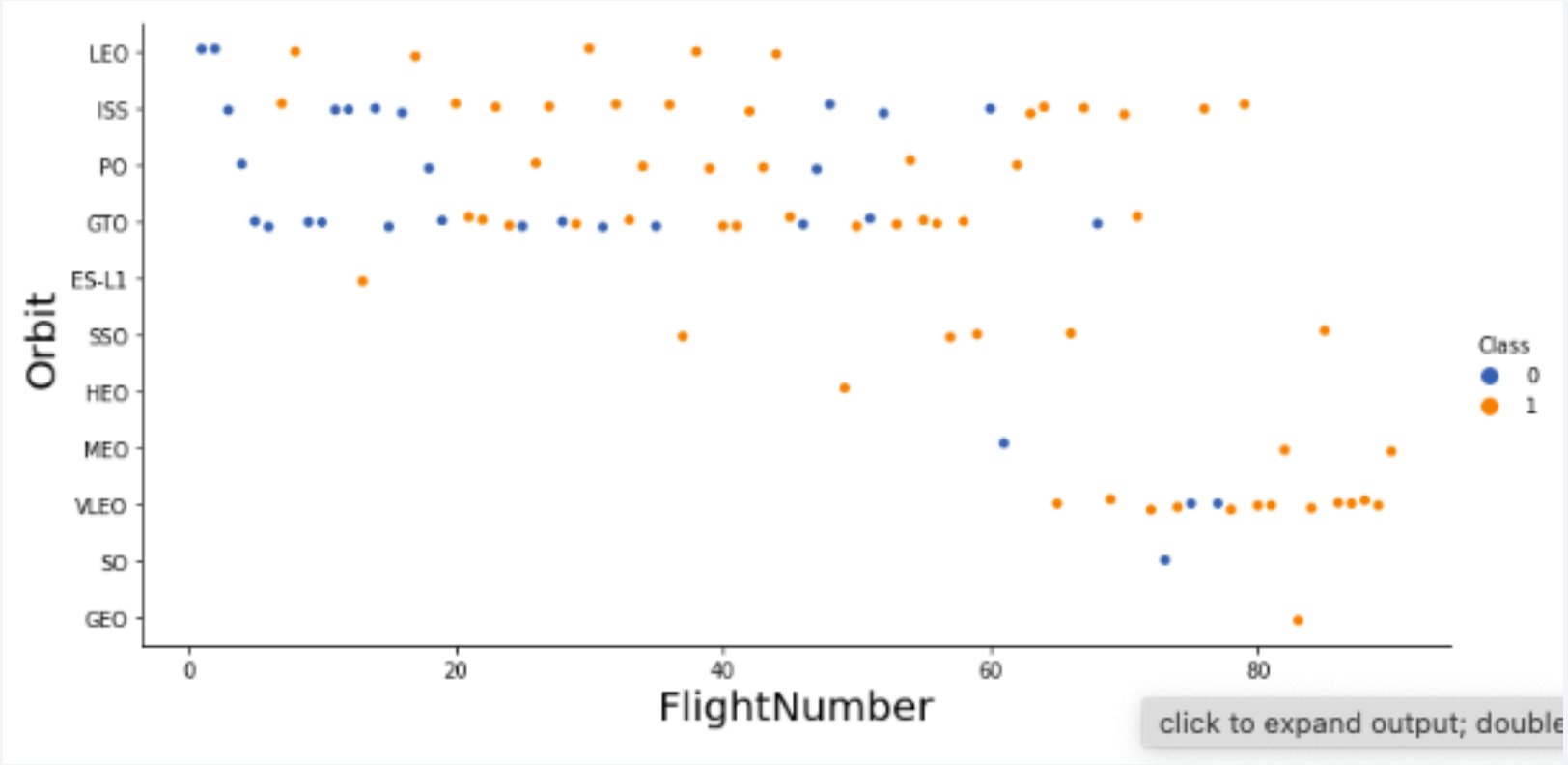
Success Rate vs. Orbit Type

- Orbit type ES-L1, GEO, HEO, and SSO have 100% success rates
- Orbit type SO has 0% success rate
- Other orbits have ~50-90% success rate



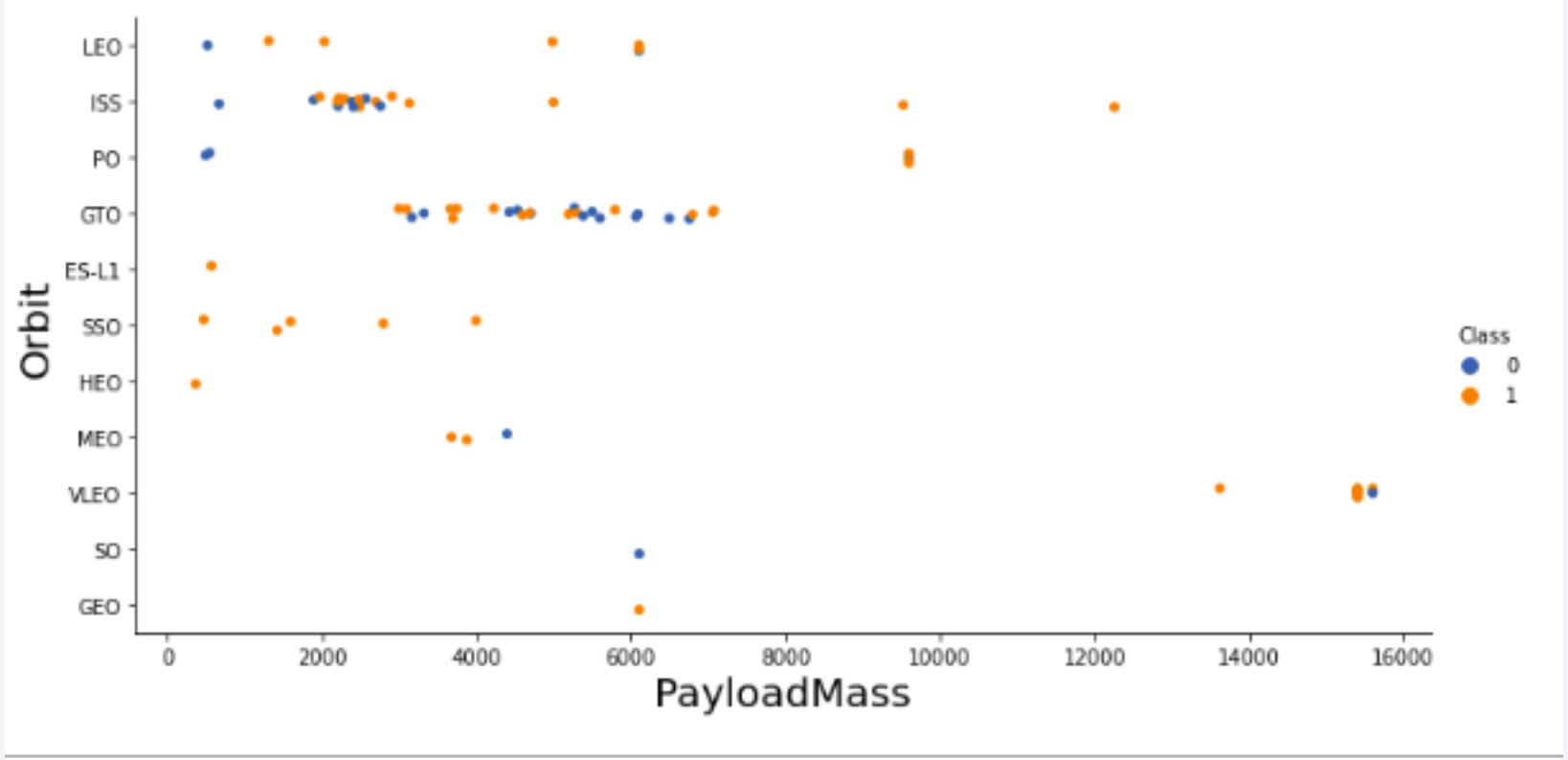
Flight Number vs. Orbit Type

- More recent flights go into VLEO orbit
- Earlier flights in LEO, ISS, PO, and GTO were likely failures



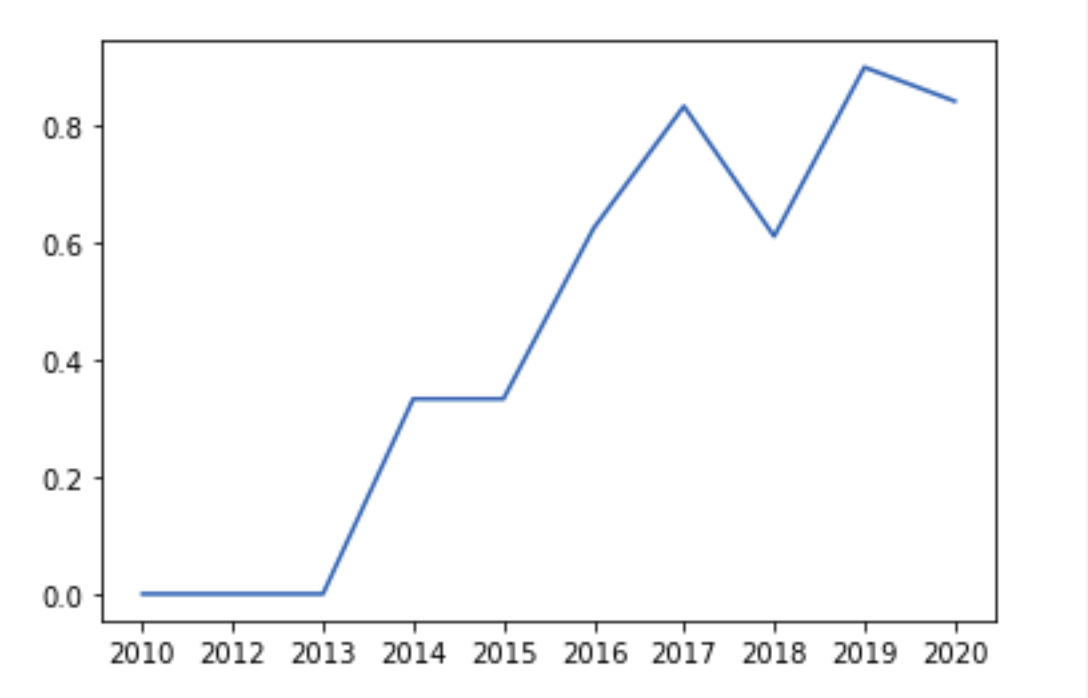
Payload vs. Orbit Type

- Higher pay load mass launches went into VLEO orbit



Launch Success Yearly Trend

- Success rate tends to be a rising trend, except for a dip in year 2018
- Success rate in earlier years (2010-2013) were 0%



All Launch Site Names

- There are 3 unique launch sites:
  - CCAFS SLC 40
  - VAFB SLC 4E
  - KSC LC 39A
  - CCAFS SLC-40

```
1 %sql SELECT DISTINCT Launch_Site FROM SPACEXTABLE
```

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

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

Launch Site Names Begin with 'CCA'

Task 2: Display 5 records where launch siters begin with CCA

```
1 %sql SELECT * FROM SPACEXTABLE WHERE Launch_Site LIKE 'CCA%' LIMIT 5
```

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

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	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	0:35:00	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

- Printed 5 records of launch sites starting with CCA

Total Payload Mass

Task 3: Display the total payload mass carried by boosters launched by NASA (CRS)

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE Payload LIKE '%CRS%'
```

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

SUM(PAYLOAD_MASS__KG_)
111268

- Total pay load mass = 111268 kg

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1
- Average pay load mass carried by booster version F9 v1.1 = 2534.97 kg

Task 4: Display average payload mass carried by booster version F9 v1.1

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

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

AVG(PAYLOAD_MASS__KG_)
2534.6666666666665

First Successful Ground Landing Date

- The first successful ground landing was on December 22nd 2015

Task 5: List the date when the first succesful landing outcome in ground pad was acheived

```
1 %sql SELECT MIN(Date) FROM SPACEXTABLE WHERE Landing_Outcome LIKE 'Success%'
```

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

MIN(Date)
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- All the booster versions in the table pictured

```
%%sql
SELECT DISTINCT Booster_Version,PAYLOAD_MASS__KG_ FROM SPACEXTABLE
WHERE PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 6000
```

Booster_Version	PAYLOAD_MASS__KG_
F9 v1.1	4535
F9 v1.1 B1011	4428
F9 v1.1 B1014	4159
F9 v1.1 B1016	4707
F9 FT B1020	5271
F9 FT B1022	4696
F9 FT B1026	4600
F9 FT B1030	5600
F9 FT B1021.2	5300
F9 FT B1032.1	5300
F9 B4 B1040.1	4990
F9 FT B1031.2	5200
F9 B4 B1043.1	5000
F9 FT B1032.2	4230
F9 B4 B1040.2	5384
F9 B5 B1046.2	5800
F9 B5 B1047.2	5300
F9 B5B1054	4400
F9 B5 B1048.3	4850
F9 B5 B1051.2	4200
F9 B5B1060.1	4311
F9 B5 B1058.2	5500
F9 B5B1062.1	4311

Total Number of Successful and Failure Mission Outcomes

- Mission outcomes were mostly successes

Task 7: List the total number of successful and failure mission outcomes

```
1 %sql SELECT Mission_Outcome,COUNT(Mission_Outcome) FROM SPACEXTABLE GROUP BY Mission_Outcome
```

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

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

Boosters Carried Maximum Payload

- The boosters are tabulated below:

Task 8: List all the booster_versions that have carried the maximum payload mass, using a subquery with a suitable aggregate function

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

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

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

- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015:

Task 9: 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

```
%%sql
SELECT substr(Date, 6,2),Landing_Outcome,Booster_Version,Launch_Site FROM SPACEXTABLE
WHERE Landing_Outcome LIKE 'Failure%drone%' AND substr(date,0,5)='2015'
```

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

substr(Date, 6,2)	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

- Ranked landing outcomes:

Task 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

```
%%sql

SELECT Landing_Outcome,COUNT(Landing_Outcome) FROM SPACEXTABLE GROUP BY Landing_Outcome
    HAVING Date BETWEEN '2010-06-04' AND '2017-03-20'
    ORDER BY COUNT(Landing_Outcome) DESC

```

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

Landing_Outcome	COUNT(Landing_Outcome)
No attempt	21
Success (drone ship)	14
Success (ground pad)	9
Failure (drone ship)	5
Controlled (ocean)	5
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

Section 3

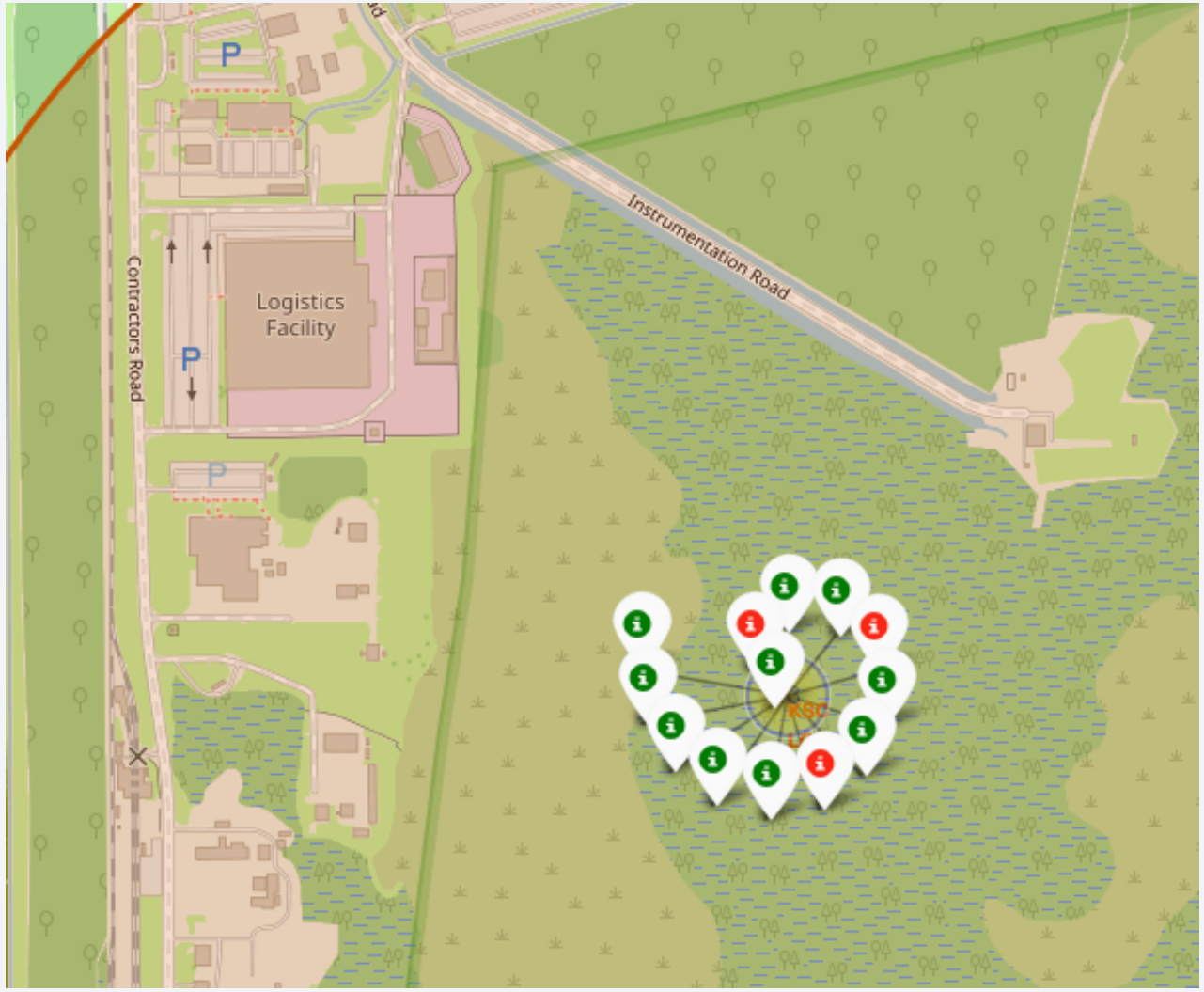
Launch Sites Proximities Analysis

All launch sites on map

- There were a total of 4 launch sites (pictured below)
- Launch sites KSC LC-39A, CCAFS LC-40, and CCAFS SLC-40 are very close to each other and appear to be overlapping
- VAFB SLC-4E is far away from the others

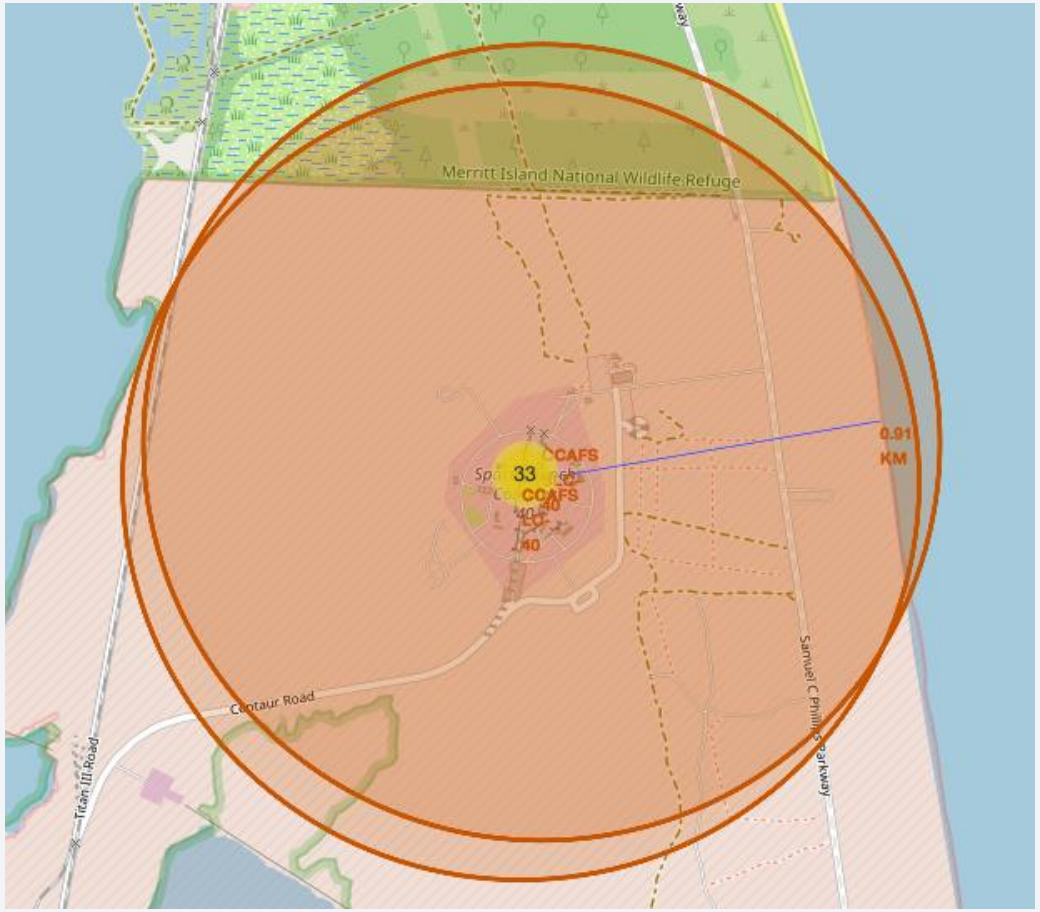
Launch successes from KSC LC-39A site

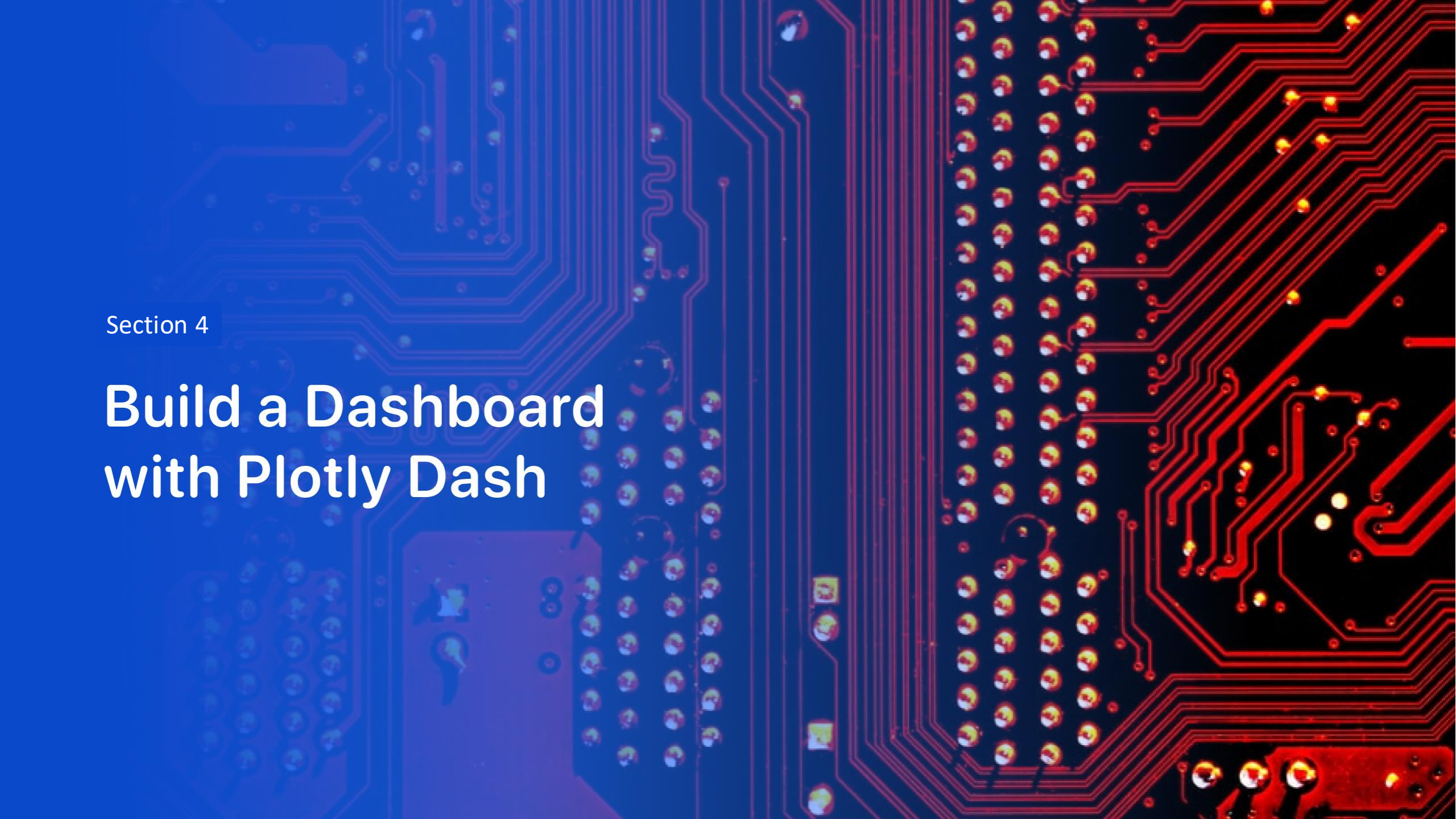
- Red = failure
- Green = success
- Total 13 launches and a ~78% success rate



<Folium Map Screenshot 3>

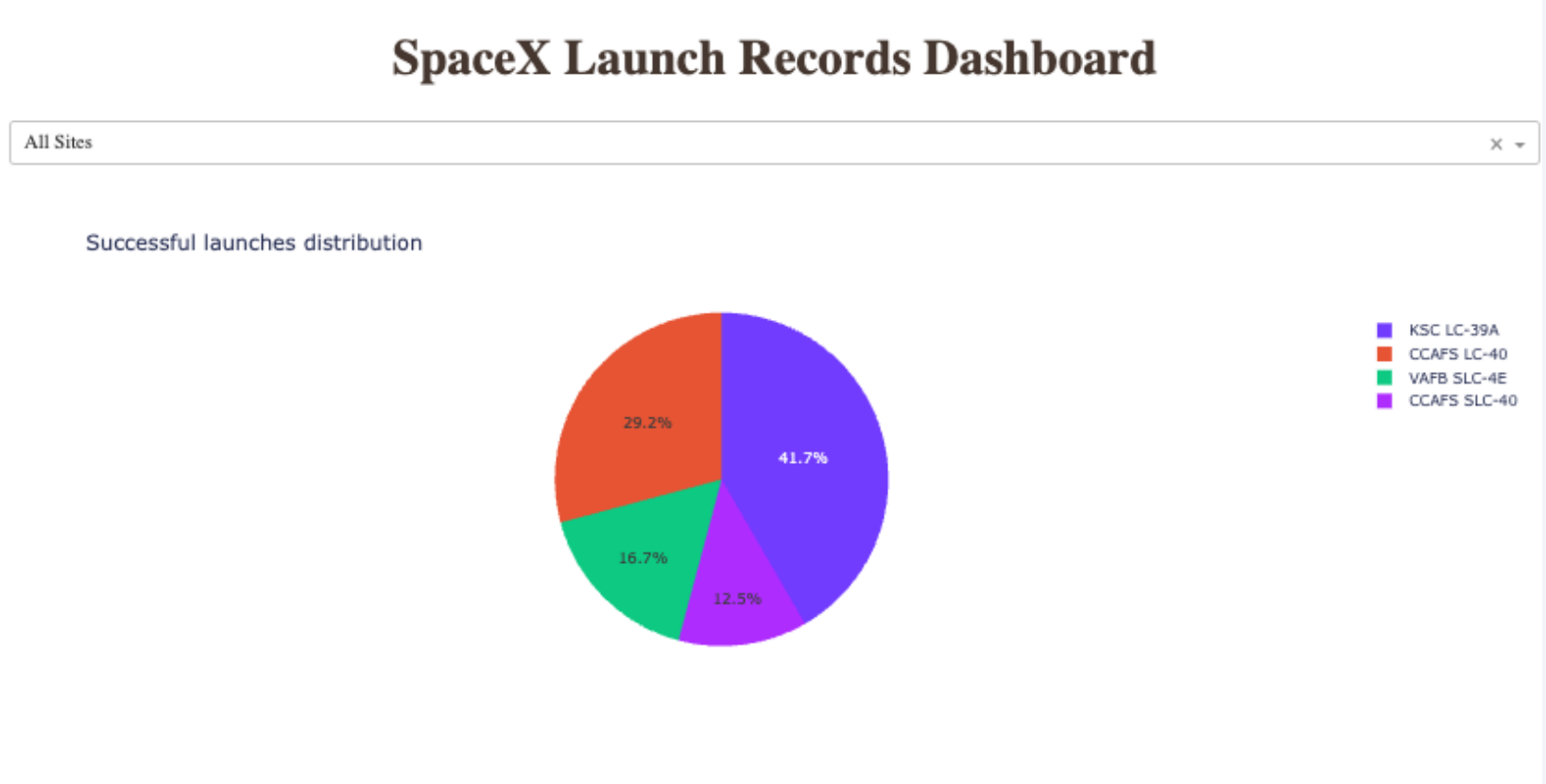
- CCAFS launch sites are about 0.91 KM away from the coastline
- But a wildlife refuge North from the launch site is closer
- The launch site must have disaster preparedness protocols to minimize impact to the ecology of the wildlife refuge, during a worst-case-scenario




Section 4
Build a Dashboard with Plotly Dash

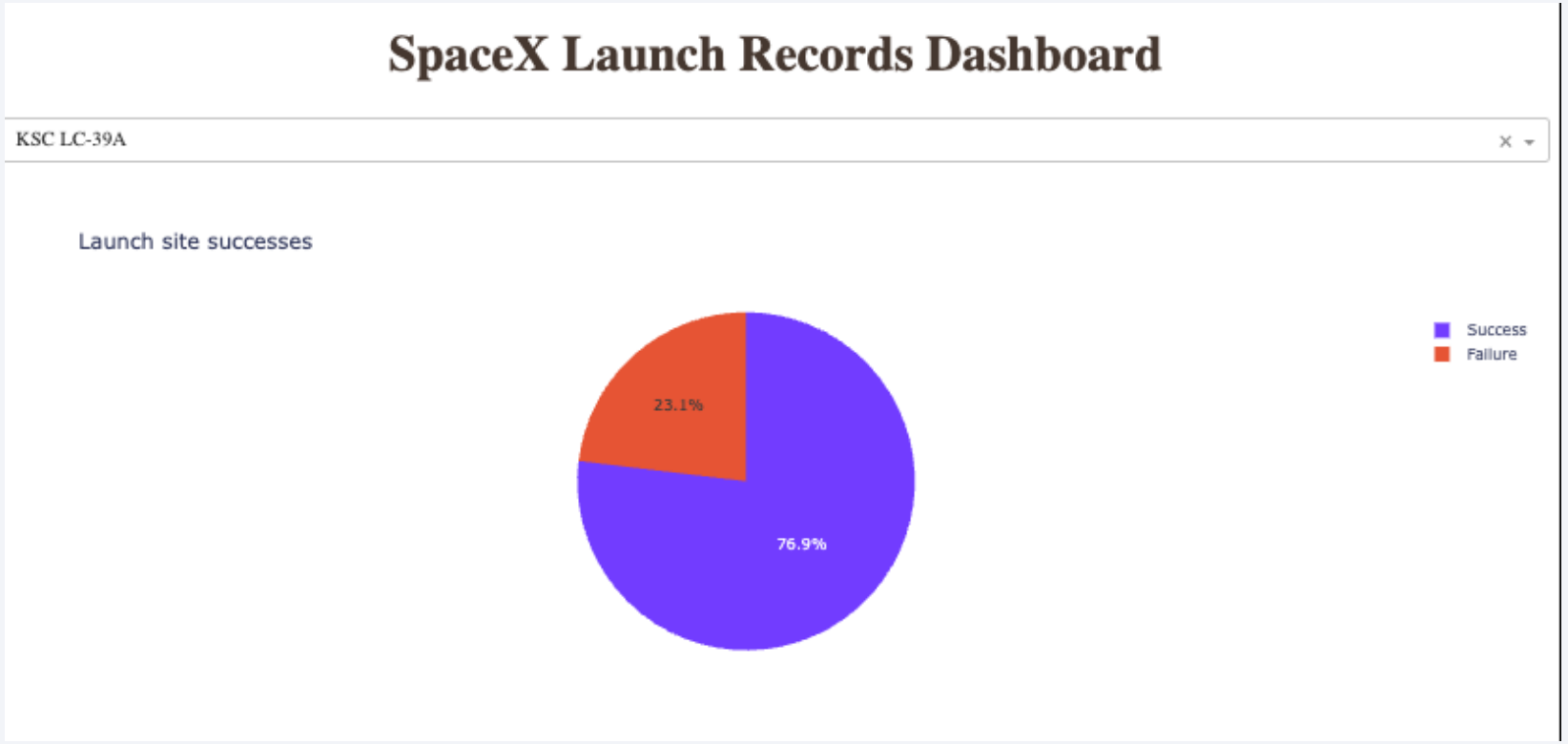
Launch Success of All Sites

- KSC LC-39A seems to have the greatest share of successful launches (41.7%), followed by CCAFS LC-40 (29.2%), VAFB SLC-4E (16.7%) and CCAFS SLC-40 (12.5%)



KSC LC-39A has the highest launch success rate

- KSC LC-39A has the highest successful launch rate (76.9% of the total launches)



Successful launches by payload range

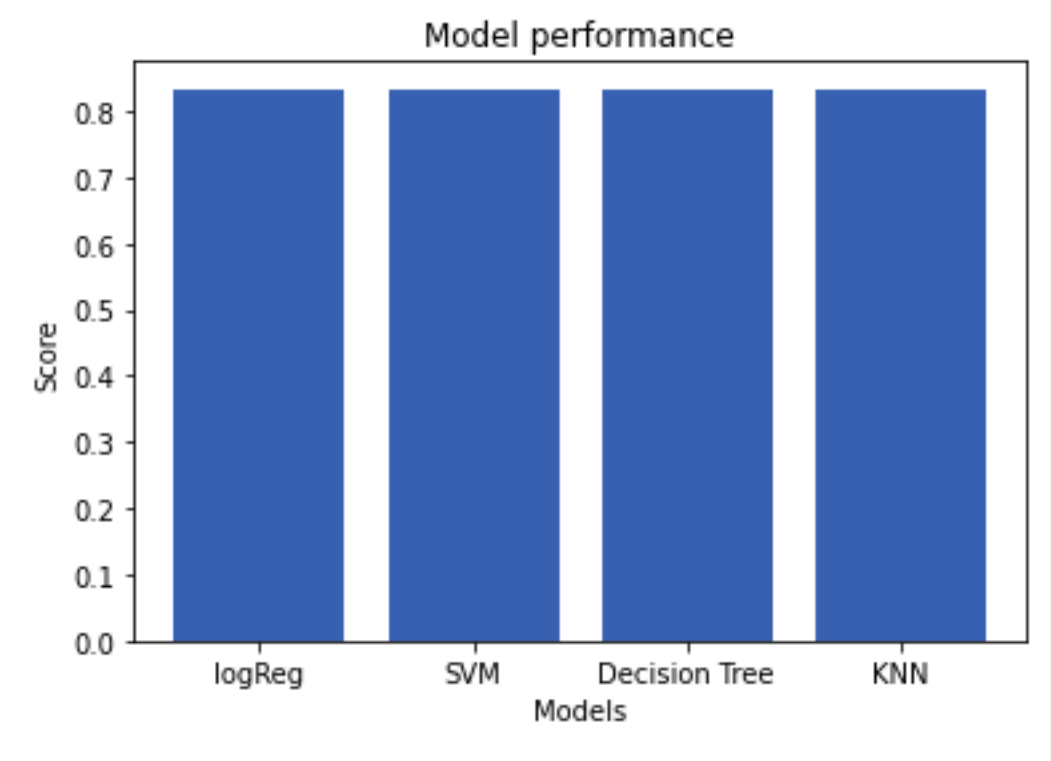
- Booster version category FT seems to have a lot of successful launches
- Booster version B4 had the highest payload mass, but resulted in one successful and one failure at that mass

Section 5

Predictive Analysis (Classification)

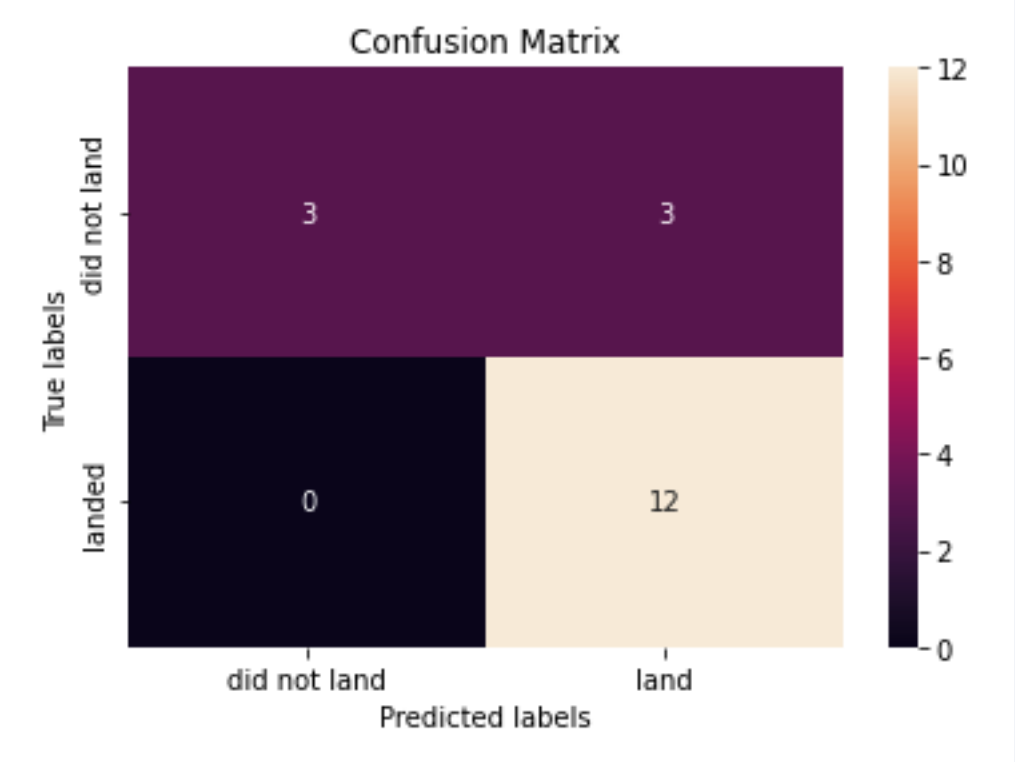
Classification Accuracy

- All the models performed equally well, with an accuracy of 0.84



Confusion Matrix

- Confusion matrix of all models were the same
- 0 false negatives (predicted no landing when actually landed)
- 3 false positives (predicted landed when actually no landing)



Conclusions

- The predictor models can estimate the landing success to quite a high level of accuracy
- Before any future launches, the rocket data can be input in the model to predict the outcome
- However, one caveat is that the dataset is quite heavily skewed to be mostly landed outcomes
- An improvement of the model could include stratified cross validation, leading to more reliable prediction

Appendix

- NA

Thank you!