Winning Space Race with Data Science

Maor Assa
March 15th 2023

Outline

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

Executive Summary

Summary of methodologies used

- Data Collection through API
- Data Collection with Web Scraping
- Data Wrangling
- Exploratory Data Analysis with SQL
- Exploratory Data Analysis with Data Visualization
- Interactive Visual Analytics with Folium
- Machine Learning Prediction

Summary of the results

Success rate is dependent on booster maturity (especially since 2016)

Orbit matters: Some orbits are showing very high success rate

Launch site matters: East coast launch sites are showing greater success rate

Payload matters: some payload ranges show very high success rate while others show very low rate

Introduction: spaceX commercial launches

- spaceX is the largest commercial space launches provider
- space X began commercial launches in 2013
- spaceX unique commercial advantage is based on the ability to re-use the first stage Booster
- space X launch cost is 40% lower than traditional methods due to the above capability
- space X is operating from 4 sites and launch various payloads to various orbits

The questions we are researching in this report:

- What parameters impact a successful landing of the 1st stage Booster
- How can this knowledge be leveraged for commercial launch companies

Section 1

Methodology

Methodology

Executive Summary

1

Data collection

spaceX open database: obtained via spaceX API

Wikipedia data: obtained via web scrapping

2

Data Processing

Missing values were identified

Data types were validated

Clear launch outcome label was added

Feature analysis and one-hot encoding was performed to allow application of prediction models

3

Data Analysis

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

2 methods were used to collect the data for analysis:

1

spaceX data via API

1. get request: SpaceX API
2. .json() function: decoded the response as Json
3. .json_normalize(): Data as pandas dataframe
4. Check & append missing values

2

Wikipedia data via web scraping

1. Scraping data toa BeautifulSoup() object
2. extract data as HTML tables
3. convert to a pandas dataframe

Data Collection – SpaceX API

spaceX data via API

get request: SpaceX API

```
In [6]:  spacex_url="https://api.spacexdata.com/v4/launches/past"

In [7]:  response = requests.get(spacex_url)
```

.json() function: decoded the response as Json

```
In [9]:  static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_c

In [10]: response.status_code
Out[10]: 200
```

.json_normalize(): Data as pandas dataframe

```
In [11]: # Use json_normalize meethod to convert the json result into a dataframe
         data=pd.json_normalize(response.json())

In [12]: # Get the head of the dataframe
         data.head()
```

Out[12]:

	static_fire_date_utc	static_fire_date_unix	net	window	rocket	success	failures	details	crew	ships	capsules	
0	2006-03-17T00:00:00.000Z	1.142554e+09	False	0.0	5e9d0d95eda69955f709d1eb	False	[{'time': 33, 'altitude': None, 'reason': 'merlin engine failure'}]	Engine failure at 33 seconds and loss of vehicle	[]	[]	[]	[5eb0e4b5b

Check & append missing values

```
In [26]: # Calculate the mean value of PayloadMass column
         mean= data_falcon9["PayloadMass"].mean()

         # Replace the np.nan values with its mean value
         data_falcon9['PayloadMass'].fillna(value=mean,inplace=True)

         data_falcon9.isnull().sum()
```

- GitHub URL of the completed SpaceX API calls notebook
  https://github.com/maorassa/Finalcapstonetest/blob/master/Final%20capstone%20API%20data%20collection.ipynb

Data Collection - Scraping

Wikipedia data via web scraping

Scraping data toa BeautifulSoup() object

```
In [6]:
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(response, 'html.parser')
```

Print the page title to verify if the `BeautifulSoup` object was created properly

```
In [7]:
# Use soup.title attribute
print(soup.title)
```

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

extract data as HTML tables

```
In [8]:
# Use the find_all function in the BeautifulSoup object, with element type `table`
# Assign the result to a list called `html_tables`
html_tables = soup.find_all("table")
print(html_tables)
```

```
[<table class="multicol" role="presentation" style="border-collapse: collapse; padding: 0; border: 0; background:transparent; width:100%;">
<tbody><tr>
<td style="text-align: left; vertical-align: top;">
<h3><span class="mw-headline" id="Rocket_configurations">Rocket configurations</span></h3>
<div class="chart noresize" style="margin-top:1em;max-width:420px;">
<div style="position:relative;min-height:320px;min-width:420px;max-width:420px;">
<div style="float:right;position:relative;min-height:240px;min-width:320px;max-width:320px;border-left:1px black solid;border-bottom:1px black s
d;">
```

convert to a pandas dataframe

```
In [15]:
headings = []
for key,values in dict(launch_dict).items():
    if key not in headings:
        headings.append(key)
    if values is None:
        del launch_dict[key]

def pad_dict_list(dict_list, padel):
    lmax = 0
    for lname in dict_list.keys():
        lmax = max(lmax, len(dict_list[lname]))
    for lname in dict_list.keys():
        ll = len(dict_list[lname])
        if  ll < lmax:
            dict_list[lname] += [padel] * (lmax - ll)
    return dict_list

pad_dict_list(launch_dict,0)

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

Out[15]:

	Flight No.	Launch site	Payload	Payload mass	Orbit	Customer	Launch outcome	Version Booster	Booster landing	Date	Time
0	1	CCAFS	Dragon Spacecraft Qualification Unit	0	LEO	<generator object Tag._all_strings at 0x000001...	Success\n	F9 v1.0B0003.1	Failure	4 June 2010	18:45

GitHub URL of the completed SpaceX API calls notebook

https://github.com/maorassa/Finalcapstonetest/blob/master/Final_capstone_webscraping.ipynb

Data Wrangling

2 major tasks were done at this stage:

1 Identify missing values and data types

```
In [3]: df.isnull().sum()/df.count()*100

Out[3]: FlightNumber       0.000
        Date               0.000
        BoosterVersion     0.000
```

```
In [4]: df.dtypes

Out[4]: FlightNumber         int64
        Date                object
        BoosterVersion      object
        PayloadMass        float64
```

2 Analyze training labels

Launch sites:

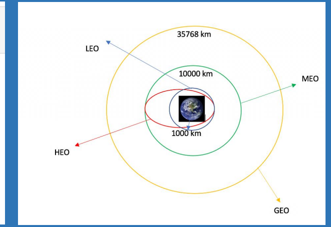
```
In [5]: # Apply value_counts() on column LaunchSite
        df['LaunchSite'].value_counts()

Out[5]: CCAFS SLC 40    55
        KSC LC 39A      22
        VAFB SLC 4E     13
        Name: LaunchSite, dtype: int64
```

Orbits:

```
In [6]: # Apply value_counts on Orbit column
        df['Orbit'].value_counts()

Out[6]: 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
```



Outcome Label:

```
In [10]: # landing_class = 0 if bad_outcome
         # 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)
```

GitHub URL of the completed SpaceX data wrangling notebook

https://github.com/maorassa/Finalcapstonetest/blob/master/Final%20capstone%20Data%20wrangling%20.ipynb

EDA with Data Visualization

We have plotted few charts in order to decide on the most relevant features for explaining the variables of a successful 1st stage re-landing:

Variables analyzed	Chart type
FlightNumber vs. PayloadMass	Scatter plot
FlightNumber vs LaunchSite	Scatter plot
launch Vs payload mass.	Scatter plot
success rate of each orbit type	Bar chart
FlightNumber and Orbit type	Scatter plot
Payload and Orbit type	Scatter plot
launch success yearly trend	Line chart

GitHub URL of the completed SpaceX data visualization notebook

https://github.com/maorassa/Finalcapstonetest/blob/master/Final%20capstone%20EDA%20with%20Pandas%20and%20Matplotlib.ipynb

EDA with SQL

We applied EDA with SQLite to get insight from the data. We wrote queries to find out for instance:

- Unique launch sites
- Sum of payload mass launched by NASA (CRS)
- Avg. payload mass carried by F9 v1.1 booster
- The total number of successful and failure mission outcomes
- Boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
- Booster versions which have carried the maximum payload mass

```
In [4]: !pip install sqlalchemy==1.3.9

Requirement already satisfied: sqlalchemy

In [5]: !pip3 install ipython-sql
```


GitHub URL of the completed SpaceX EDA notebook

https://github.com/maorassa/Finalcapstonetest/blob/master/Final%20capstone-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

We used few geographical analysis tools based on Folium map analysis:

Folium object	Purpose
folium.Circle	Highlight launch sites
folium.Marker	Mark the launch sites
MarkerCluster	Mark multiple launches outcomes per site
MousePosition	Mark specific relevant points of interest
PolyLine	Calculate distance (line of site vector)

GitHub URL of the completed SpaceX Folium notebook

https://github.com/maorassa/Finalcapstonetest/blob/master/Final%20capstone%20Folium%20maps%20analysis.ipynb

Build a Dashboard with Plotly Dash

We have used an interactive dashboard to analyze the follwoing:

Chart	Analysis
Success rate per sites selected	Analyze the difference between sites
Outcome per payload mass	Analyze the payload impact on outcomes
Payload slider	Analyze the payload impact for specific weight ranges
Site drop down list	Analyze specific sites

GitHub URL of the completed SpaceX Plotly notebook

https://github.com/maorassa/Finalcapstonetest/blob/master/Plotly_dash_code

Create a NumPy array from the column Class in data

Predictive Analysis (Classification)

spaceX prediction analysis stages

Create NumPy array from the launch outputs & Standardize the data

```
In [5]: Y = data['Class'].to_numpy()
```

```
In [9]: # students get this
        transform = preprocessing.StandardScaler().fit(X)
In [10]: X=transform.transform(X)
In [11]: X.std(axis=0)
Out[11]: array([1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1.,
               1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1.,
               1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1.,
               1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1.,
               1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1.,
               1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1.])
```

split the data X and Y into training and test data (20% test)

```
X_train, X_test, Y_train, Y_test

X_train, X_test, Y_train, Y_test = train_test_split(X,Y,test_size=0.2,random_state=2)
```

Create models objects

Model
Logistic regression
SVM
Decision tree classifier
KNN

Gridsearch objects to optimize parameters for each model

Calculate accuracy for each model

```
In [20]: yhat=logreg_cv.predict(X_test)
In [21]: logreg_cv.score(X_test, Y_test)
```


GitHub URL of the completed SpaceX Prediction notebook

https://eu-de.dataplatform.cloud.ibm.com/analytics/notebooks/v2/8dc75764-98b2-4870-a025-a80e637650dd/view?access_token=044fda62e289bfb083aa6ab9b89b235a363aaceb383bee62c28cf7945dc423c0

Results

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

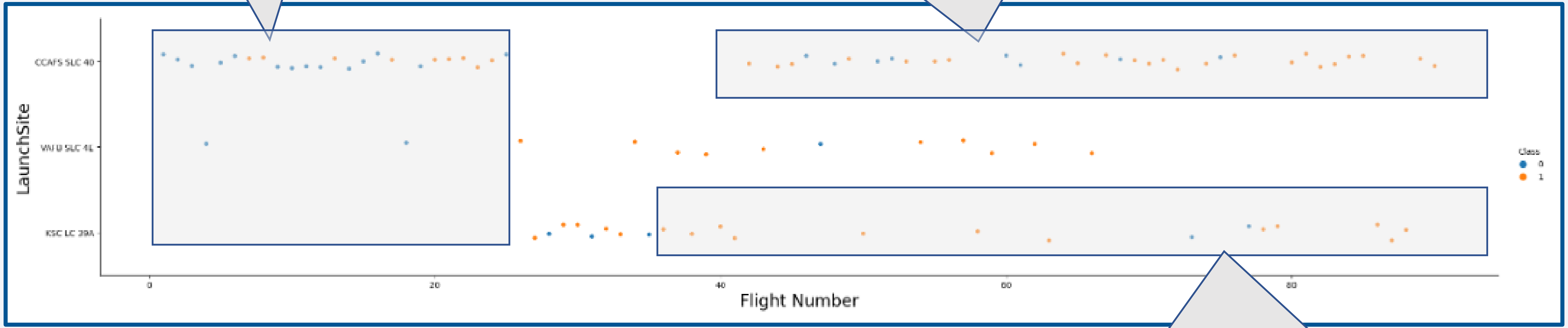
Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

First launches (0-25) show higher failure rate

CCAFS SLC 40:
launch site with many launches and success rate that grows with time

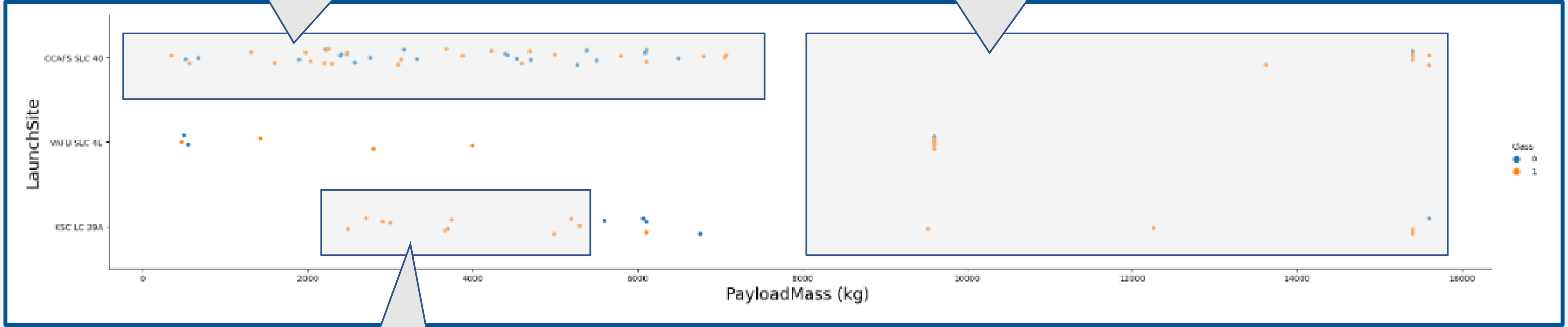


KSC LC 39A:
launch site with many launches and success rate that grows with time

Payload vs. Launch Site

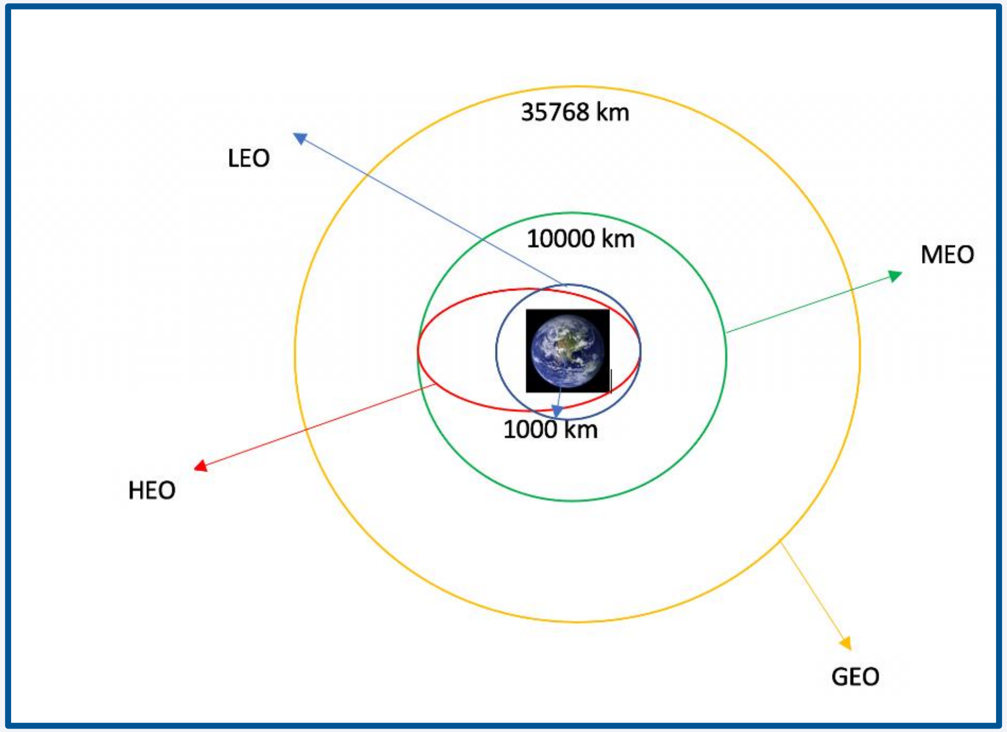
CCAFS SLC 40:
Success rate will be influenced from a relative higher number of launches

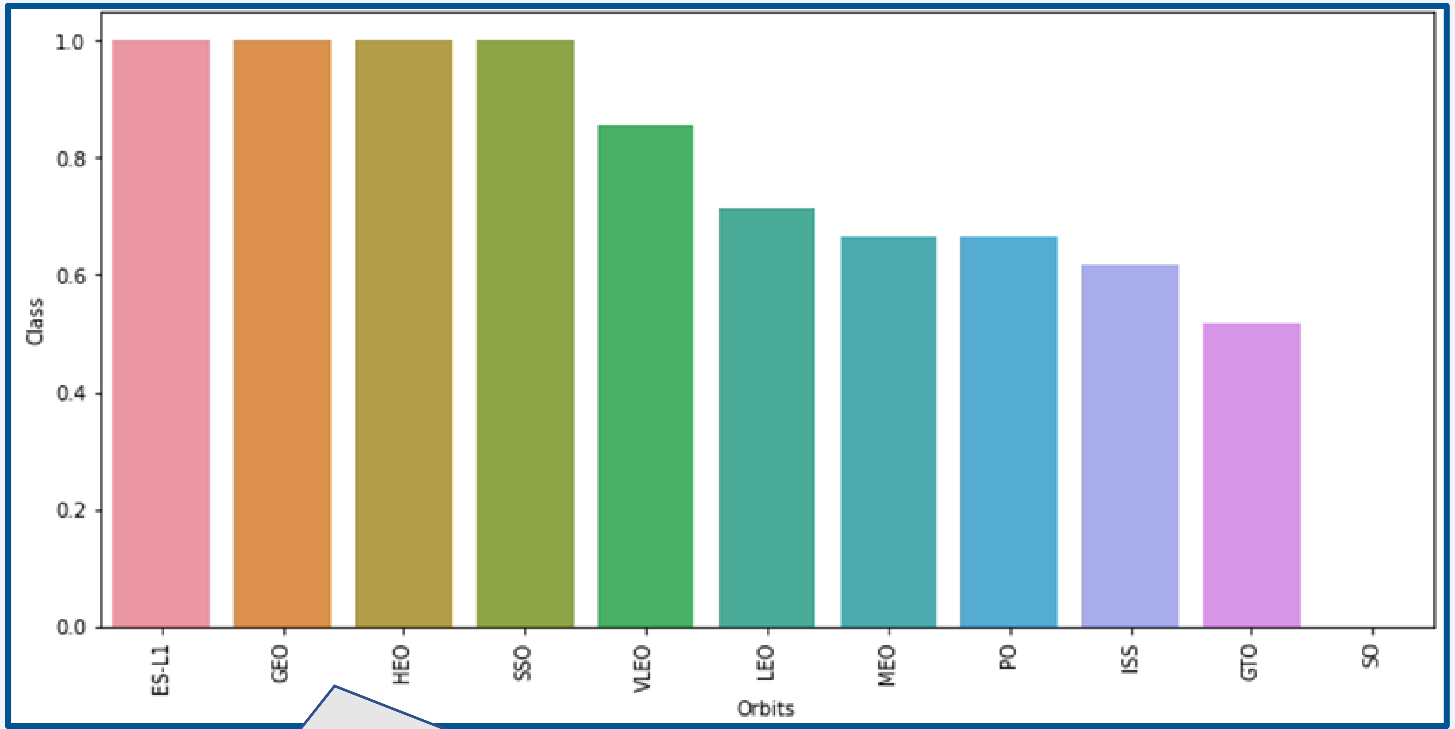
Heavy payloads (+10K kg)
Success rate seems relatively higher across sites



KSC LC 39A:
Success rate seems relatively higher for small payloads

Success Rate vs. Orbit Type



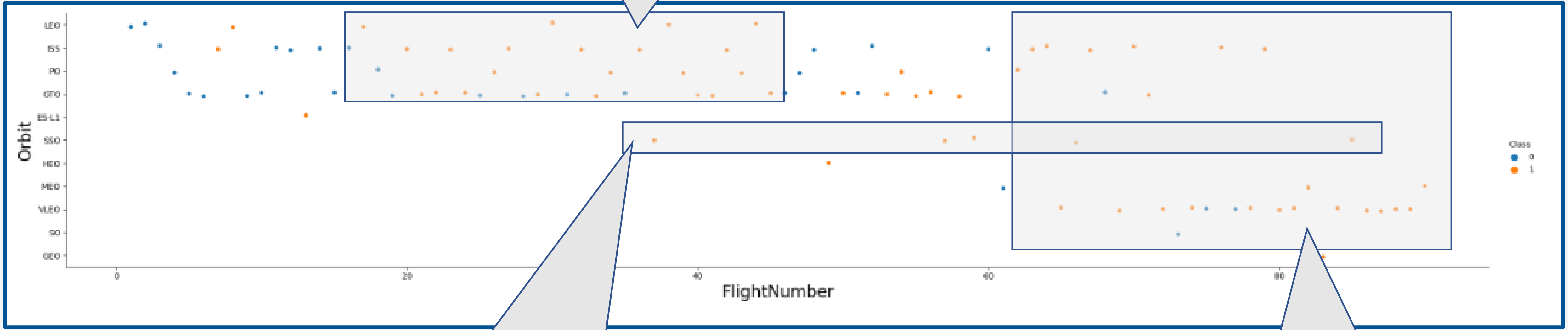


Higher orbits have a higher prevalence of high success rate

Flight Number vs. Orbit Type

Flights 20 - 50

LEO, ISS, PO orbits are showing better success rate



SSO

100% success rate increases across timeline

Flights # 60 and above

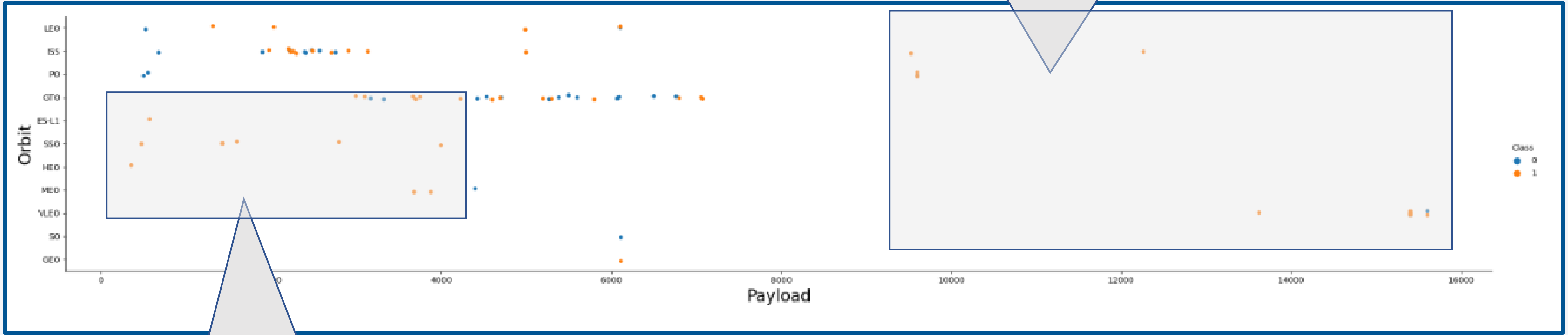
Success rate increases across orbits

Payload vs. Orbit Type

Heavy payloads (+10K kg)

3 orbits: ISS, PO, VLEO

Success rate seems relatively higher

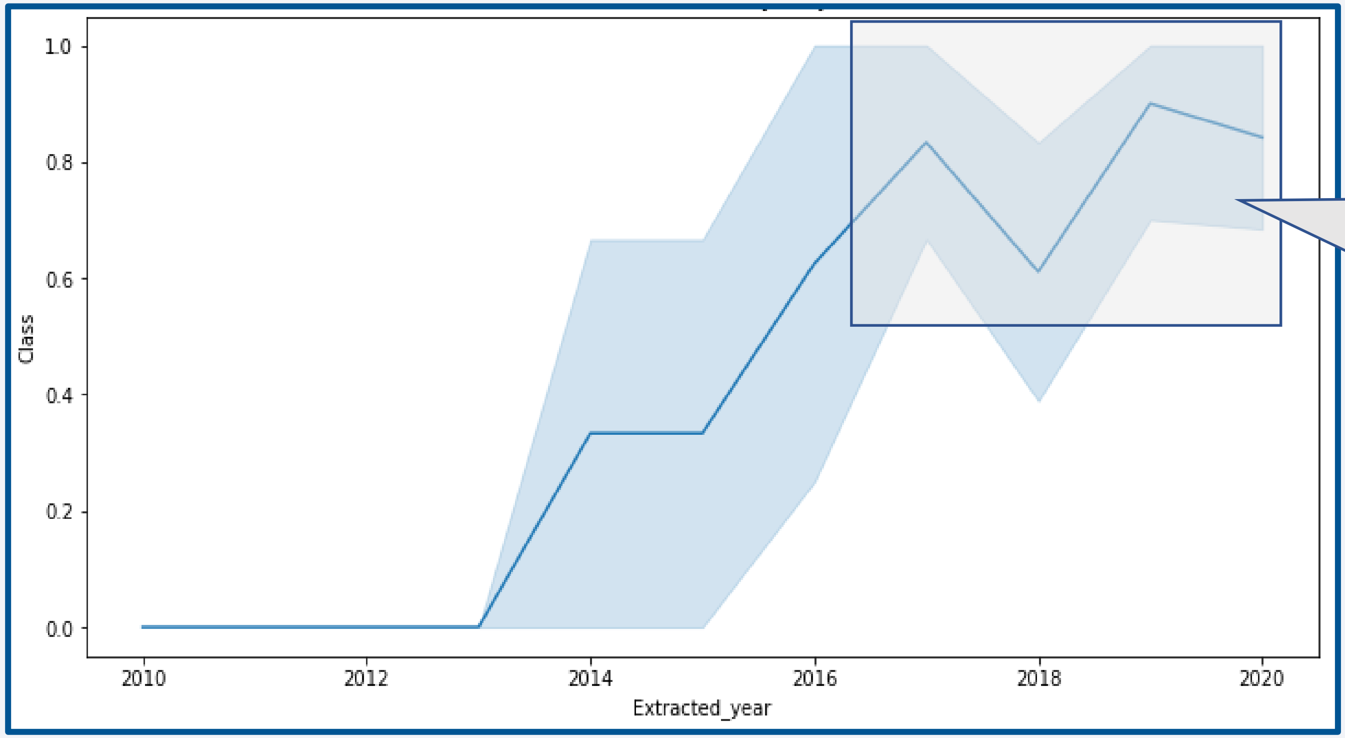


Small payloads

Success rate seems relatively higher for orbits:

SSO, MEO & GFO

Launch Success Yearly Trend



Booster maturity

- Since 2017 the launching system is showing maturity trend
- Success mean rate for the mature system is approx. 80% (compared with 66% general success rate)

All Launch Site Names

In [13]:

```
%sql select distinct LAUNCH_SITE from SPACEXTBL
```

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

Out[13]:

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

DISTINCT command query the launch sites column for specific launch sites

Launch Site Names Begin with 'CCA'

LIKE command query the launch sites with specific names

In [23]:
```
%sql select * from SPACEXTBL where "Launch_Site" like 'CCA%' limit 5
```

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

Out[23]:

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

SUM command totals the payload values

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

In [25]:
```
%sql select sum(PAYLOAD_MASS__KG_) from SPACEXTBL where Customer='NASA (CRS)'
```

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

Out[25]:

sum(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

AVG command clacs the avg for payload values

```
In [27]: %sql select avg(PAYLOAD_MASS__KG_) from SPACEXTBL where Booster_Version='F9 v1.1'

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

Out[27]:

avg(PAYLOAD_MASS__KG_)
2928.4

First Successful Ground Landing Date

MIN command clacs the min date

```
In [58]: %sql select min(date) from SPACEXTBL where 'LANDING__OUTCOME' = 'Success (ground pad)'

 * sqlite:///my_data1.db
Done.

Out[58]:
min(date)
None
```

Successful Drone Ship Landing with Payload between 4000 and 6000

AND command for a query with several conditions

```
In [48]: %sql SELECT * FROM SPACEXTBL WHERE 'Landing _Outcome' = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000

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

Out[48]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing _Outcome

	boosterversion
0	F9 FT B1022
1	F9 FT B1026
2	F9 FT B1021.2
3	F9 FT B1031.2

Total Number of Successful and Failure Mission Outcomes

COUNT + GROUPBY commands for summing outcomes

In [41]:
```
%sql Select MISSION_OUTCOME,count(MISSION_OUTCOME) from SPACEXTBL GROUP BY MISSION_OUTCOME
```

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

Out[41]:

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

Boosters Carried Maximum Payload

Sub query to define the max payload and use it as a criteria

```
In [55]: %sql select distinct BOOSTER_VERSION,PAYLOAD_MASS__KG_ from SPACEXTBL where PAYLOAD_MASS__KG_ = (select MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL)

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

Out[55]:

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

In [68]:
```
%sql select substr(Date, 4, 2) as month,substr(Date, 7, 4)as year,LANDING__OUTCOME,BOOSTER_VERSION,LAUNCH_SITE from SPACEXTBL
where 'year'=2015 and 'LANDING__OUTCOME' = 'Failure (drone ship)'
```

Out[18]:

	boosterversion	launchsite	landingoutcome
0	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
1	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

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

In [77]:

```
%sql select 'Landing _Outcome', count(*) as count, Date from SPACEXTBL where date between '04-06-2010' and '20-03-2017' GROUP BY 'Landing _Outcome'
ORDERBY count DESC
```

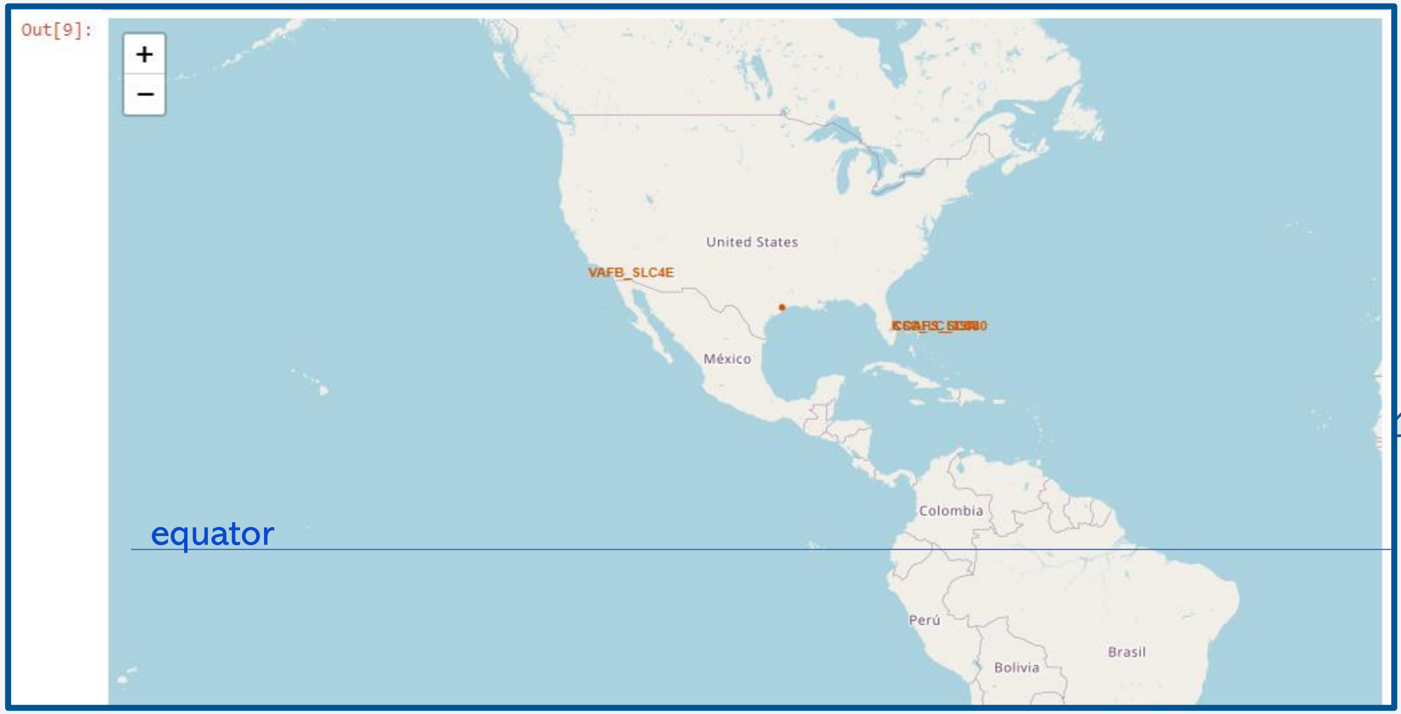
Out[19]:

	landingoutcome	count
0	No attempt	10
1	Success (drone ship)	6
2	Failure (drone ship)	5
3	Success (ground pad)	5
4	Controlled (ocean)	3
5	Uncontrolled (ocean)	2
6	Precluded (drone ship)	1
7	Failure (parachute)	1

Section 3

Launch Sites Proximities Analysis

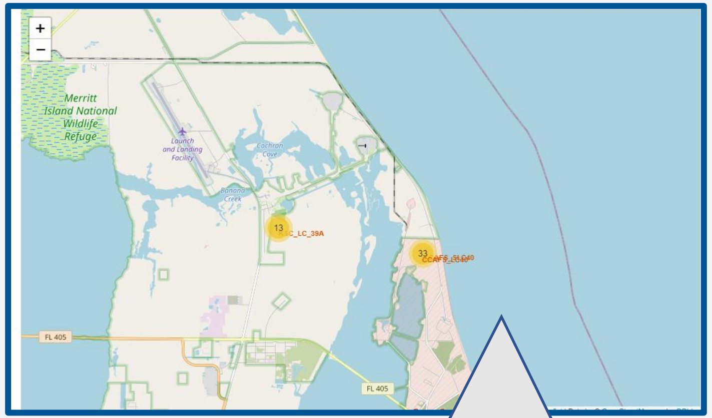
Launch sites locations



There are 2 areas of launching.

- 1 in west coat
- 1 in the east coast
- The east coast sites are closer to the equator

Variation between east coast sites

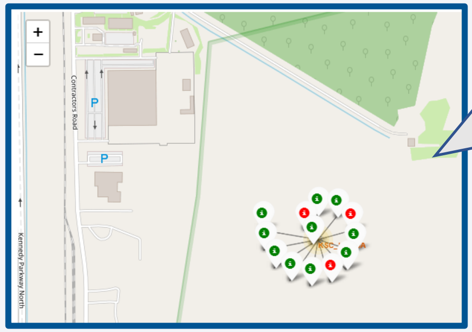


Zoom on 3 east coast sites:

KSC LXC 39A

CCA FS LC 40

CAFS SLC 40



KSC LSC 39A

Success rate relative higher

CCAFS SLC 40:

Success rate relative lower

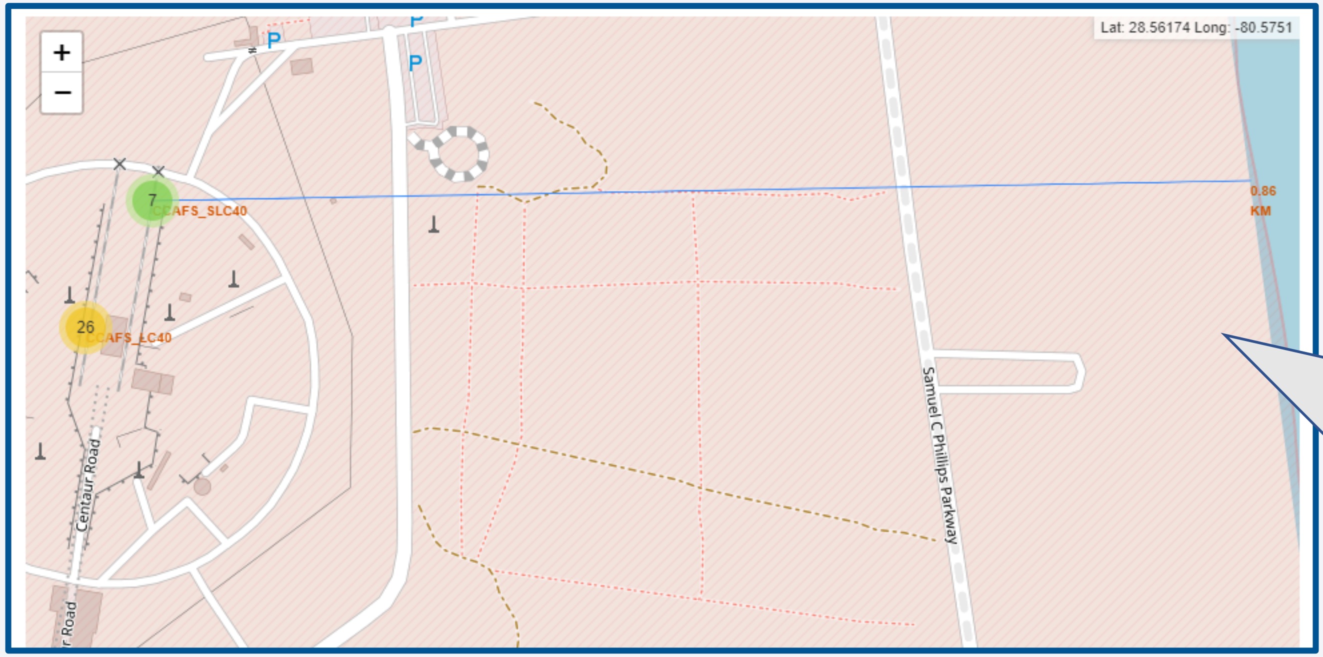
High number of launches

CCAFS LC 40:

Low number of launches

Low success rate

<Folium Map Screenshot 3>



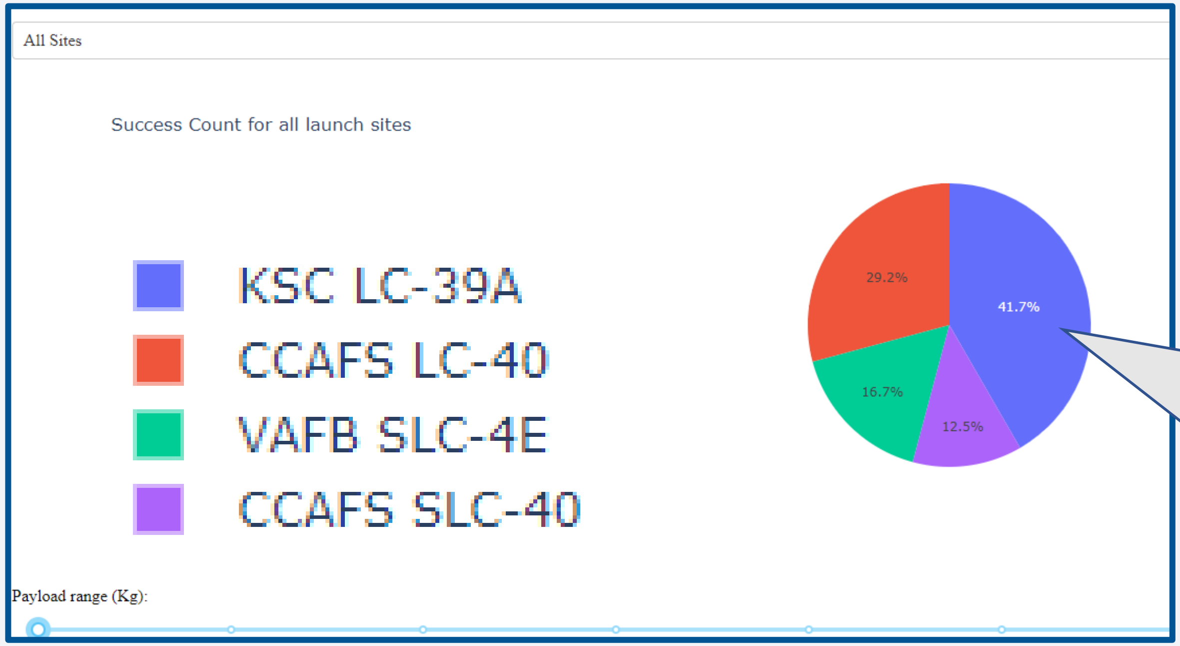
Proximity to sea may affect the success rate for 1^{st} stage landing.

1^{st} stage may land on sea platform or land

Section 4

Build a Dashboard with Plotly Dash

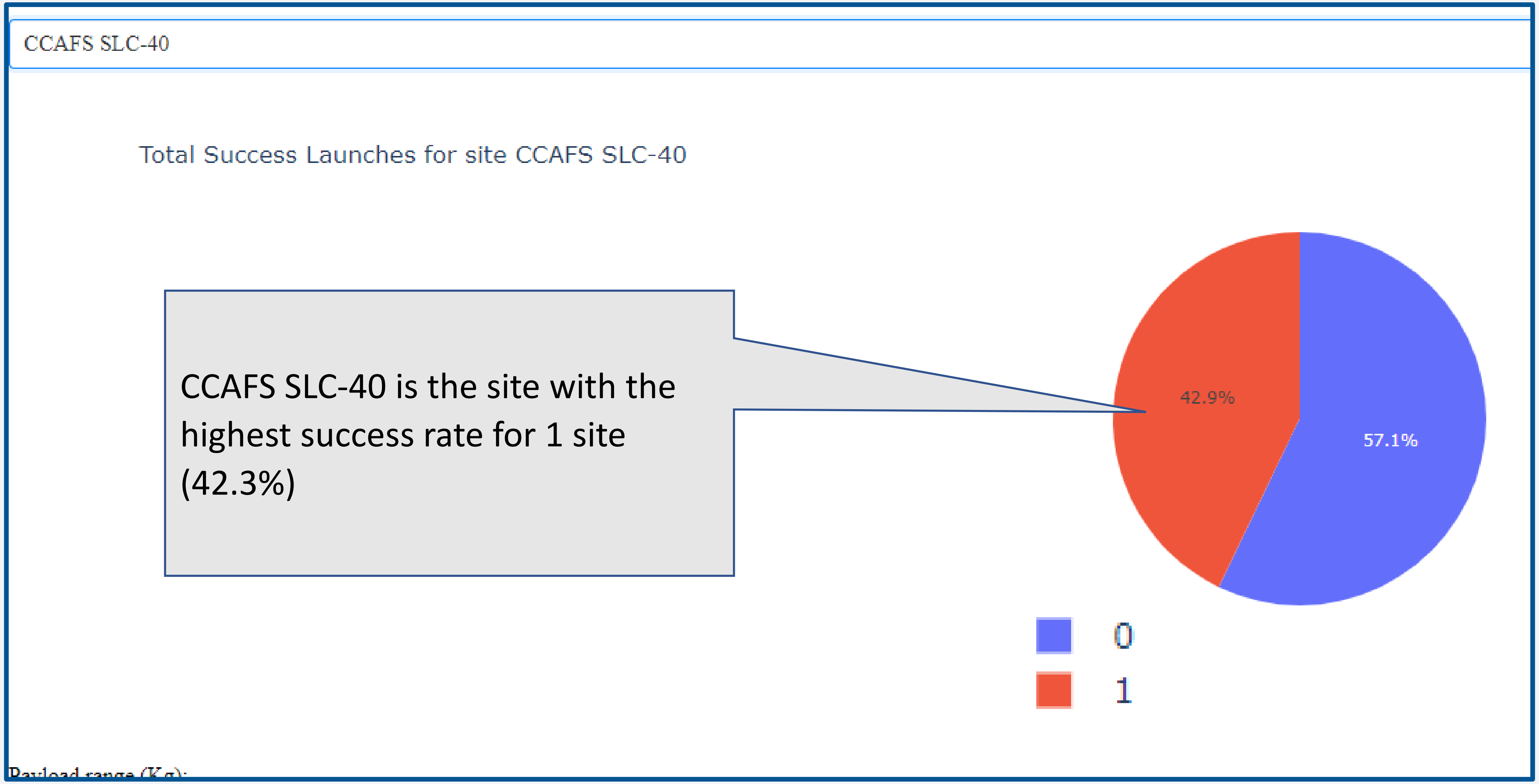
Which site has the highest success rate



KSC LC 39A

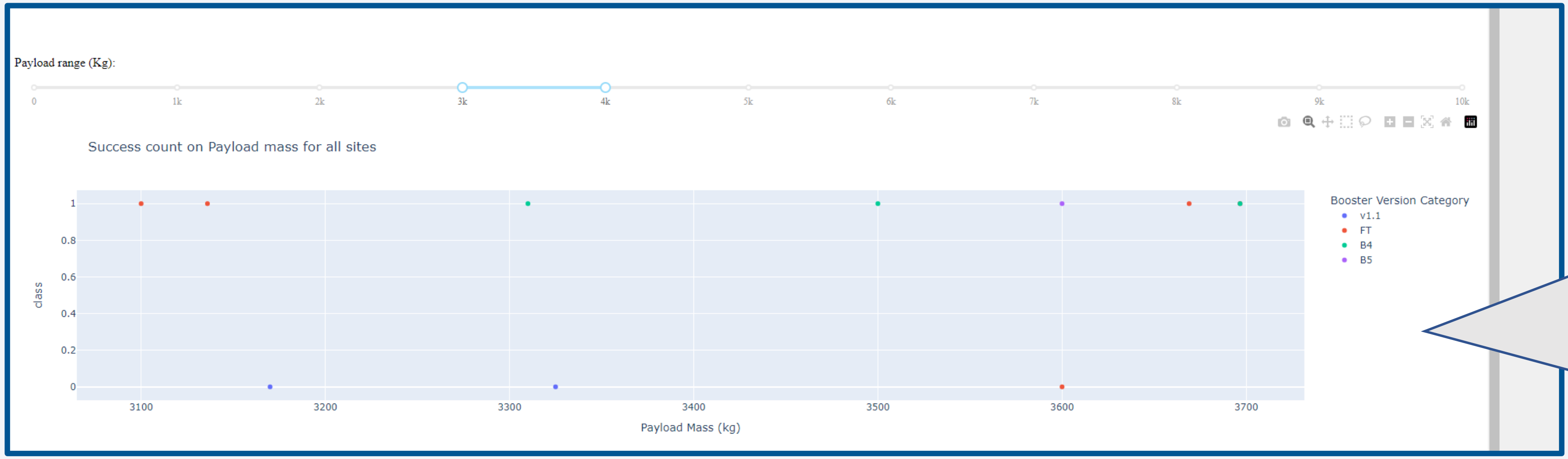
41% of all successful launches

<Dashboard Screenshot 2>

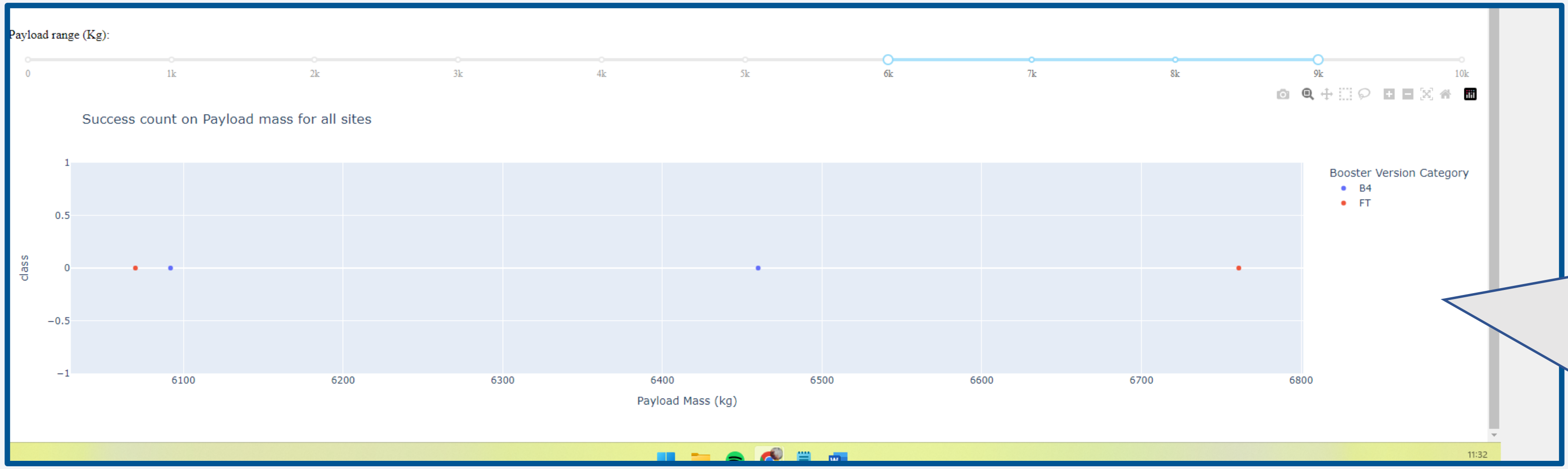


CCAFS SLC-40 is the site with the highest success rate for 1 site (42.3%)

High & Low success per payload (kg) ranges



3K-4K kg =(7/10) 70%



6K – 9K kg =(0/4) = 0%

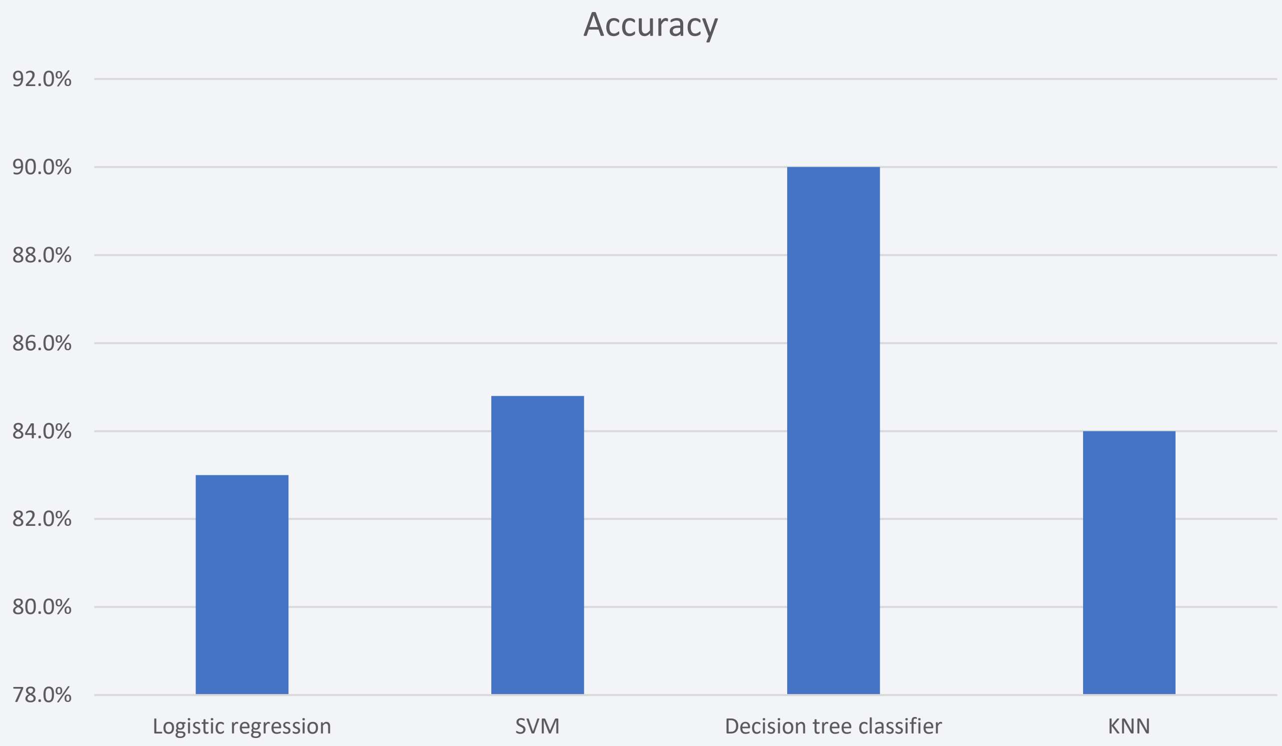
Section 5

Predictive Analysis (Classification)

Classification Accuracy

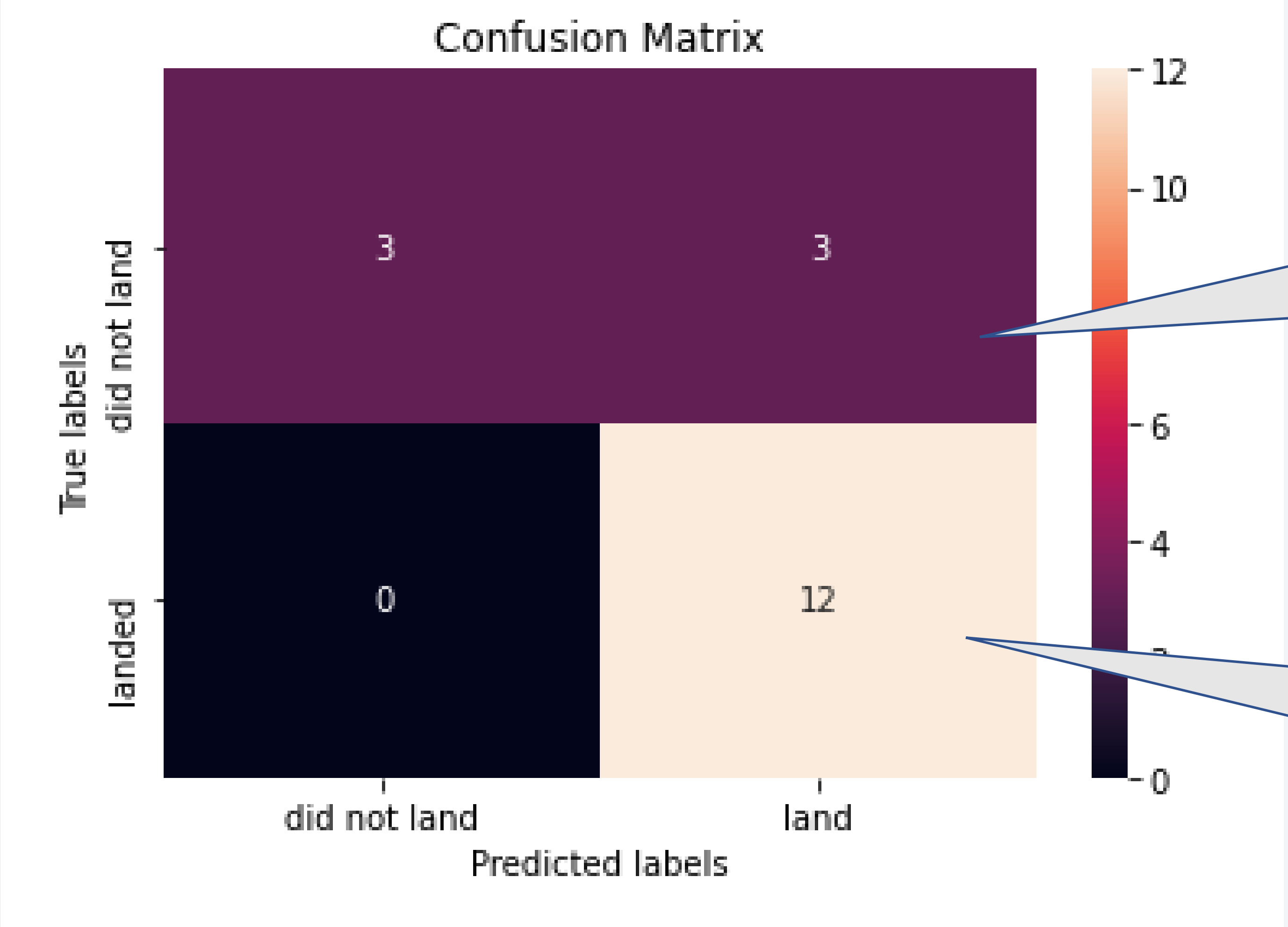
In [18]:
```
features = df[['FlightNumber', 'PayloadMass', 'Orbit', 'LaunchSite', 'Flights', 'GridFins', 'Reused', 'Legs', 'LandingPad', 'Block', 'ReusedCount',
features.head()
```

The features used



Decision tree classifier yielded the highest accuracy

Confusion Matrix



The over predicted success, with 50% accuracy on failure prediction

The model predicted successful 1st stage landing very well

Conclusions

- Success rate is dependent on booster maturity (especially since 2016)
- Orbit matters: Some orbits are showing very high success rate
- Launch site matters: East coast launch sites are showing greater success rate
- Payload matters: some payload ranges show very high success rate while others show very low rate

Profile of exemplary successful launch	
Launch date	2020 and onwards
Launch site	KSC LC-39A
Payload (kg)	3K – 4K
Orbit	ES-L1, GEO, HEO, SSO, VLEO

Thank you!