IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Pierre Delzers
02/01/2023

Outline

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

Executive Summary

- Summary of methodologies
  - Data collection
  - Data Wrangling
  - EDA with Visualization
  - EDA with SQL
  - Building interactive maps
  - Building dashboards
  - Predictive Analysis

- Summary of all results
  - First analysis based on EDA
  - Interactive maps and dashboards
  - Predictive results

Introduction

- Project background and context

  This project aims to predict if the Falcon 9 first stage can land successfully to reuse it and significantly reduce the cost of rocket launches. Saving and reusing the first stage can save tens of millions for each launch. Therefore, determining if the first stage can land successfully and be reused can help determine the cost of a launch for the company which is an important strategic factor.

- Problems you want to find answers
  - What are the conditions for a successful landing?
  - What are the links between the variables and the success rate?
  - How can SpaceY maximize its landing success rate?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Using SpaceX REST API
  - WebScrapping from Wikipedia
- Perform data wrangling
  - Irrelevant features were cleaned, and the remaining data were transformed
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Predicting results with machine learning while testing and selecting the best parameters

Data Collection

I – Data was gathered using SpaceX REST API

This API provides data about launches, their success, the rocket used, the payload, and other specifications

This data helps gather information about the success and failures of the first-stage landing

II – WebScrapping from Wikipedia

This website provides a lot of data about Falcon 9 Launcher

Data Collection – SpaceX API

```
launch_dict = {'FlightNumber': list(data['flight_number']),
'Date': list(data['date']),
'BoosterVersion':BoosterVersion,
'PayloadMass':PayloadMass,
'Orbit':Orbit,
```

```
getBoosterVersion(data)
getLaunchSite(data)
getPayloadData(data)
getCoreData(data)
```

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


```
data_falcon9 = df[df["BoosterVersion"] == "Falcon 9"]
data_falcon9.to_csv('dataset_part_1.csv', index=False)
```

```
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)
response.status_code
```


GitHub

Data Collection - Scraping

```
column_names = []
elements = soup.find_all("table")
for i in range(len(elements)):
    name = extract_column_from_header(element[i])
    if name is not  None and len(name)>0 :
        column_names.append(name)
    else:
        pass
```

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

```
launch_dict= dict.fromkeys(column_names)
```


```
html_tables = soup.find_all("table")
```

```
soup = BeautifulSoup(response.content)
```

```
launch_dict['Flight No.'] = []
launch_dict['Launch site'] = []
launch_dict['Payload'] = []
launch_dict['Payload mass'] = []
launch_dict['Orbit'] = []
```

```
df.to_csv('spacex_web_scraped.csv', index=False)
```

```
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)
```


GitHub

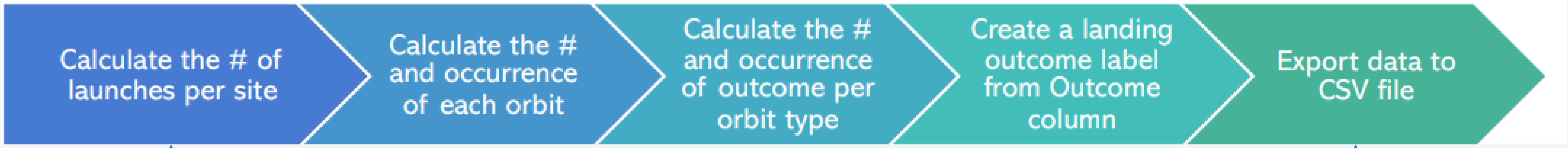
Data Wrangling

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

```
Orbit
GTO      27
ISS      21
VLEO     14
PO        9
LEO       7
```

```
landing_class = []

for i in range(0,df.shape[0]):
    if df["Outcome"][i] in bad_outcomes == False:
        landing_class.append(0)
    else:
        landing_class.append(1)
df['Class']=landing_class
```




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

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

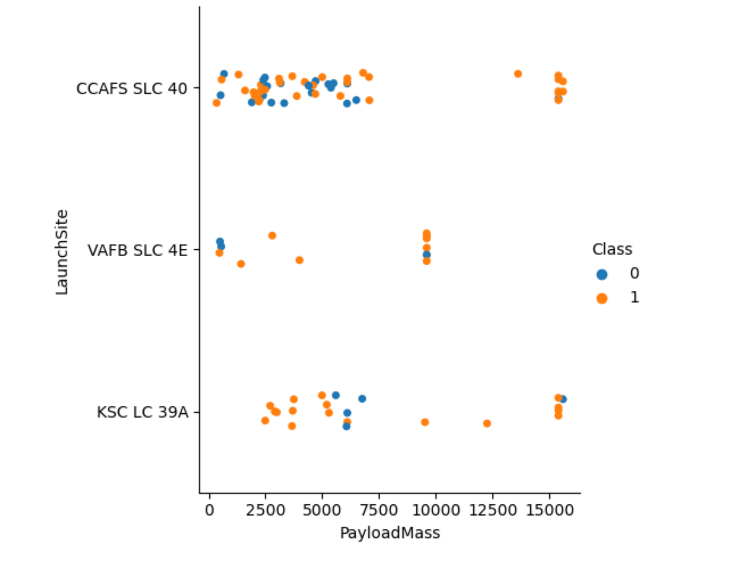
```
landing_outcomes = df[["Outcome"]].value_counts()
landing_outcomes
```

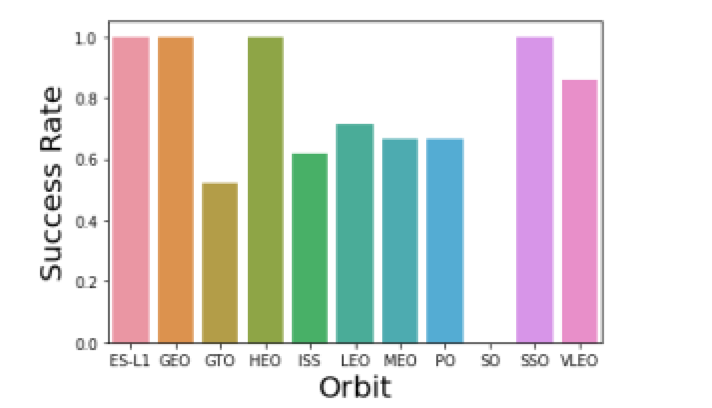
```
Outcome
True ASDS      41
None None      19
True RTLS      14
False ASDS      6
```

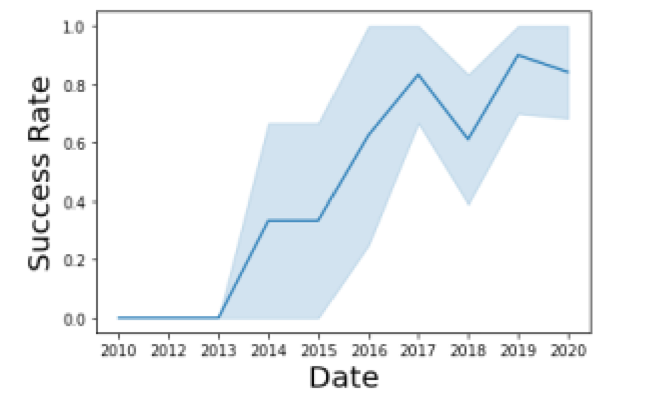
```
df.to_csv("dataset_part_2.csv", index=False)
```


GitHub

EDA with Data Visualization







Scatter plot to show correlation between:

- Flight Number & Payload Mass
- Flight Number & Launch Site
- Flight Number & Orbit Type
- Payload Mass & Launch Site
- Payload Mass & Orbit Type

Bar graph to compare Data from different groups, here:

- Success Rate & Orbit

Line plot to show trends, here:

- Success Rate & Date

GitHub

EDA with SQL

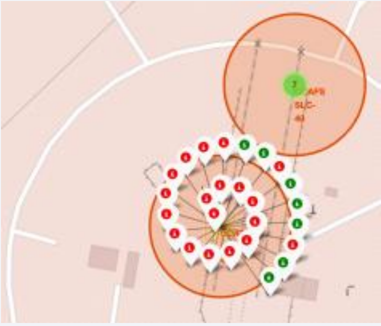
Using SQL queries we gathered from the dataset:

- The name of each unique launch site
- 5 records where launch sites begin with CCA
- The total payload mass carried
- The average payload carried by booster F9 v1.1
- The date of the first successful landing in ground pad
- List of booster with a payload between 4000 and 6000
- The total number of successful and failure mission
- The name of the booster which have carried the maximum payload mass
- The failed landing outcomes in 2015
- Ranking the count of landing outcomes between 04-06-2010 and 03-20-2017 in descending order

GitHub

Build an Interactive Map with Folium

We used coordinates for each launch site



We used a green marker for successful launches

We used lines to indicate distances between the launch site and near structures

GitHub

Build a Dashboard with Plotly Dash

Pie Chart

- Shows the success rate of all launch sites
- Display the proportion between success and fails

Scatter Plot

- Show correlation between Mission Outcome & Pauload Mass for different Booster Version

GitHub

Predictive Analysis (Classification)

Building Model

- Preprocessing and standardizing data
- Creating test and train datasets
- Testing and optimizing various parameters with GridSearch

Evaluating Model

- Checking accuracy of each model
- Optimizing hyperparameters
- Plotting confusion matrix

Selecting the best Model

- Choosing the model with the best accuracy

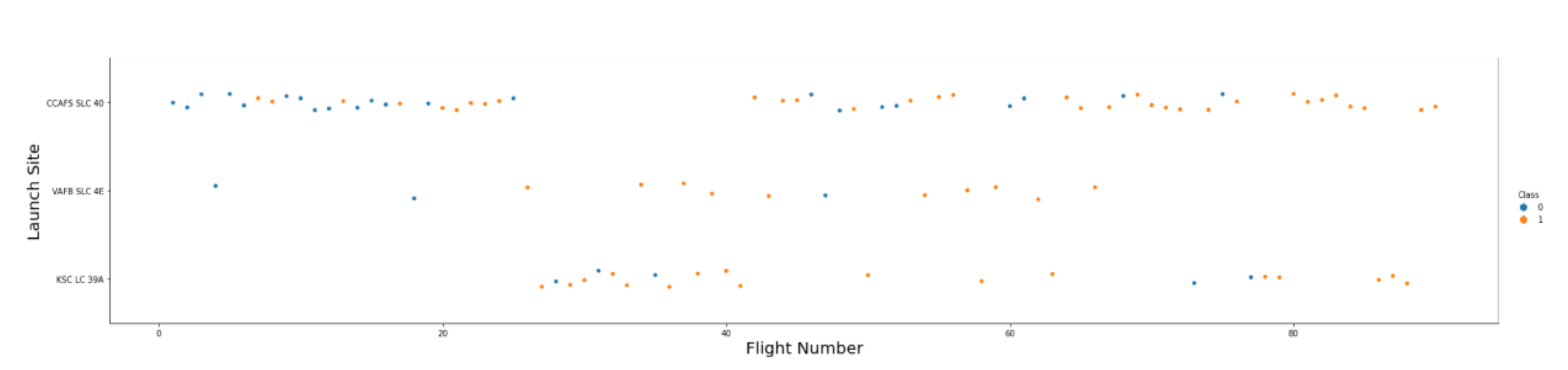
Results

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

Section 2

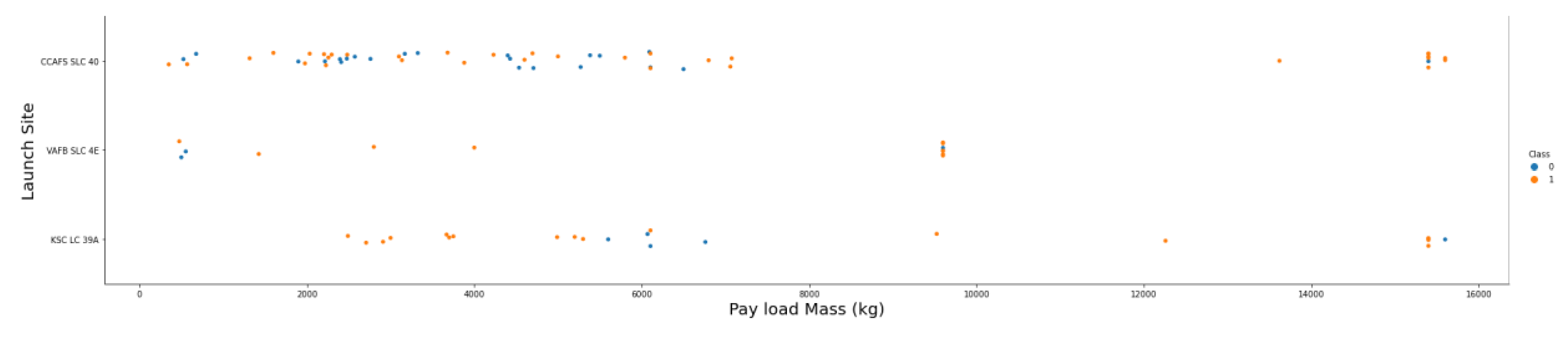
Insights drawn from EDA

Flight Number vs. Launch Site



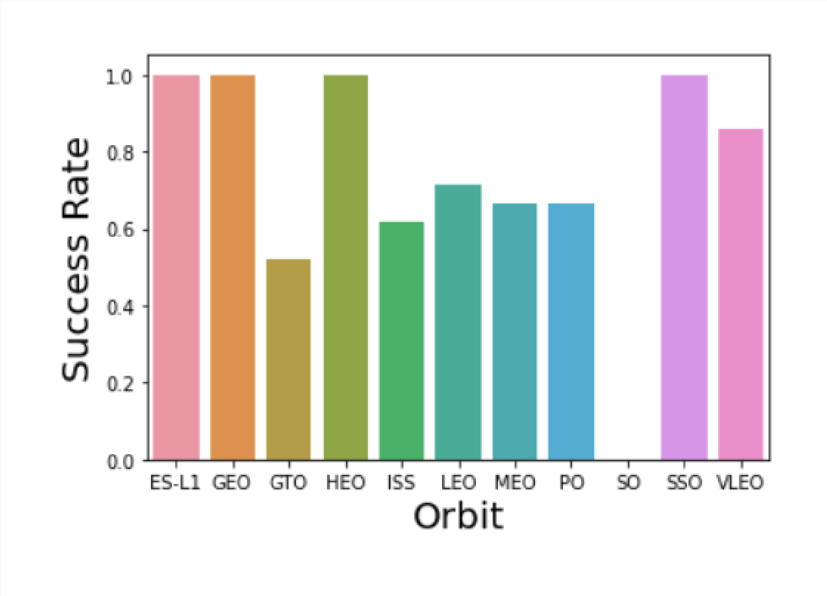
- More recent mission have a higher success rate
- The first flights were unsuccessful

Payload vs. Launch Site



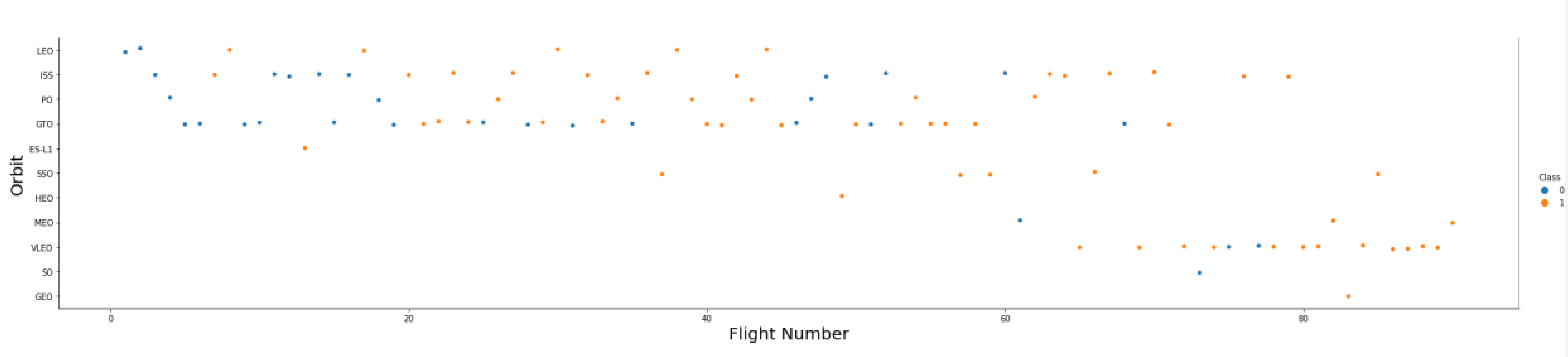
- No clear patterns
- Higher payload mass seems to have a higher success rate

Success Rate vs. Orbit Type



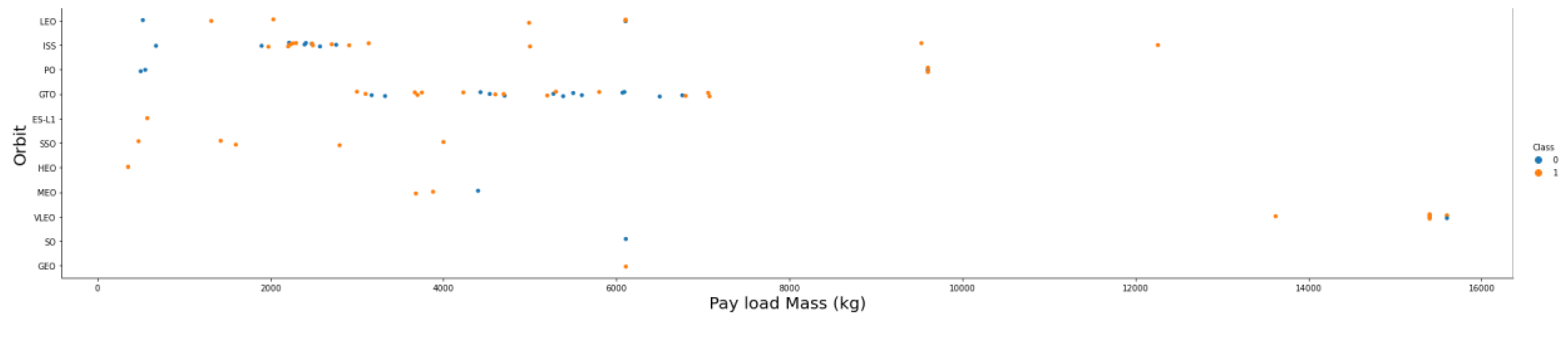
- ES-L1, GEO, HEO, SSO orbits always successfully landed
- SO never successfully landed
- Other landing success rate were between 50% and 90%

Flight Number vs. Orbit Type



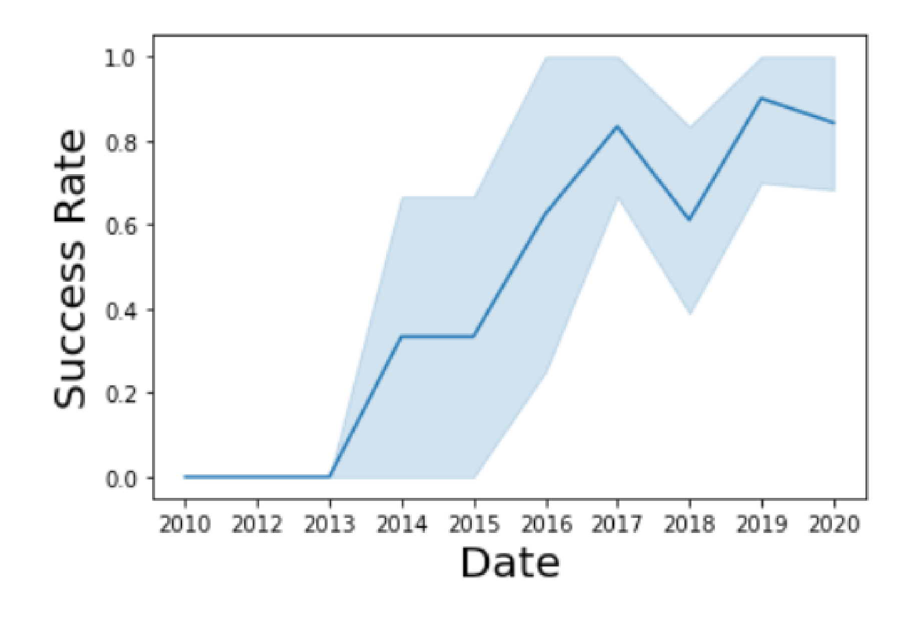
- No correlation with GTO orbit
- High flight numbers mainly go to VLEO orbit
- High flight numbers have a higher success rate
- VLEO has a high success rate

Payload vs. Orbit Type



- ISS, LEO, SSO have a high success rate
- Higher payload masses have a higher success rate for ISS
- No correlation between success rate and payload masses

Launch Success Yearly Trend



- There is no successful landing before 2014
- Success rate increase each year since 2014

All Launch Site Names

```
%sql select distinct(Launch_Site) from SPACEXTBL
```

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

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

Name of the unique launch site

Launch Site Names Begin with 'CCA'

```
%%sql
select * from SPACEXTBL 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
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

5 records where launch sites begin with CCA

Total Payload Mass

```
%sql select sum(PAYLOAD_MASS__KG_) from SPACEXTBL where Customer like "%NASA%"
```

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

sum(PAYLOAD_MASS__KG_)
107010

Total payload mass carried by boosters launched by NASA

Average Payload Mass by F9 v1.1

```
%sql select avg(PAYLOAD_MASS__KG_) from SPACEXTBL where Booster_Version like "%F9 v1.1%"

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

avg(PAYLOAD_MASS__KG_)
2534.6666666666665

Average payload mass carried by booster version F9 v1.1

First Successful Ground Landing Date

```
%sql select min(Date) from SPACEXTBL where "Landing _Outcome" like "%Success (ground_pad)%"
```

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

min(Date)
01-05-2017

Date of the first successful landing outcome in ground pad

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%sql select Booster_Version from SPACEXTBL where Mission_Outcome == "Success" and PAYLOAD_MASS__KG_ between 4000 and 6000
```

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

Booster_Version
F9 v1.1
F9 v1.1 B1011
F9 v1.1 B1014
F9 v1.1 B1016
F9 FT B1020
F9 FT B1022
F9 FT B1026

Names of boosters which have had success in drone ships and have a payload between 4000 and 6000

Total Number of Successful and Failure Mission Outcomes

```
%%sql select (case when mission_outcome like '%Success%' then 'Success' else 'Failure' end) as mission_outcomes,count(*) as qty
from SPACEXTBL group by (case when mission_outcome like '%Success%' then 'Success' else 'Failure' end)
```

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

mission_outcomes	qty
Failure	1
Success	100

Number of successes and failures mission outcomes

Boosters Carried Maximum Payload

```
%sql select Booster_Version,  max(PAYLOAD_MASS__KG_)  from SPACEXTBL
```

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

Booster_Version	max(PAYLOAD_MASS__KG_)
F9 B5 B1048.4	15600

Booster carrying the maximum payload mass

2015 Launch Records

```
%sql select "Date", "Landing _Outcome", Booster_Version, Launch_Site from SPACEXTBL where Date like "%2015%" and "Landing _Outcome" = "Failure (drone ship)"
```

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

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

Launch site of failed launch in 2015

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

```
%sql select "Landing _Outcome" ,count("Landing _Outcome") as qty from SPACEXTBL where (date between '04-06-2010' and '20-03-2017') group by "Landing _Outcome" order by 2 desc
```

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

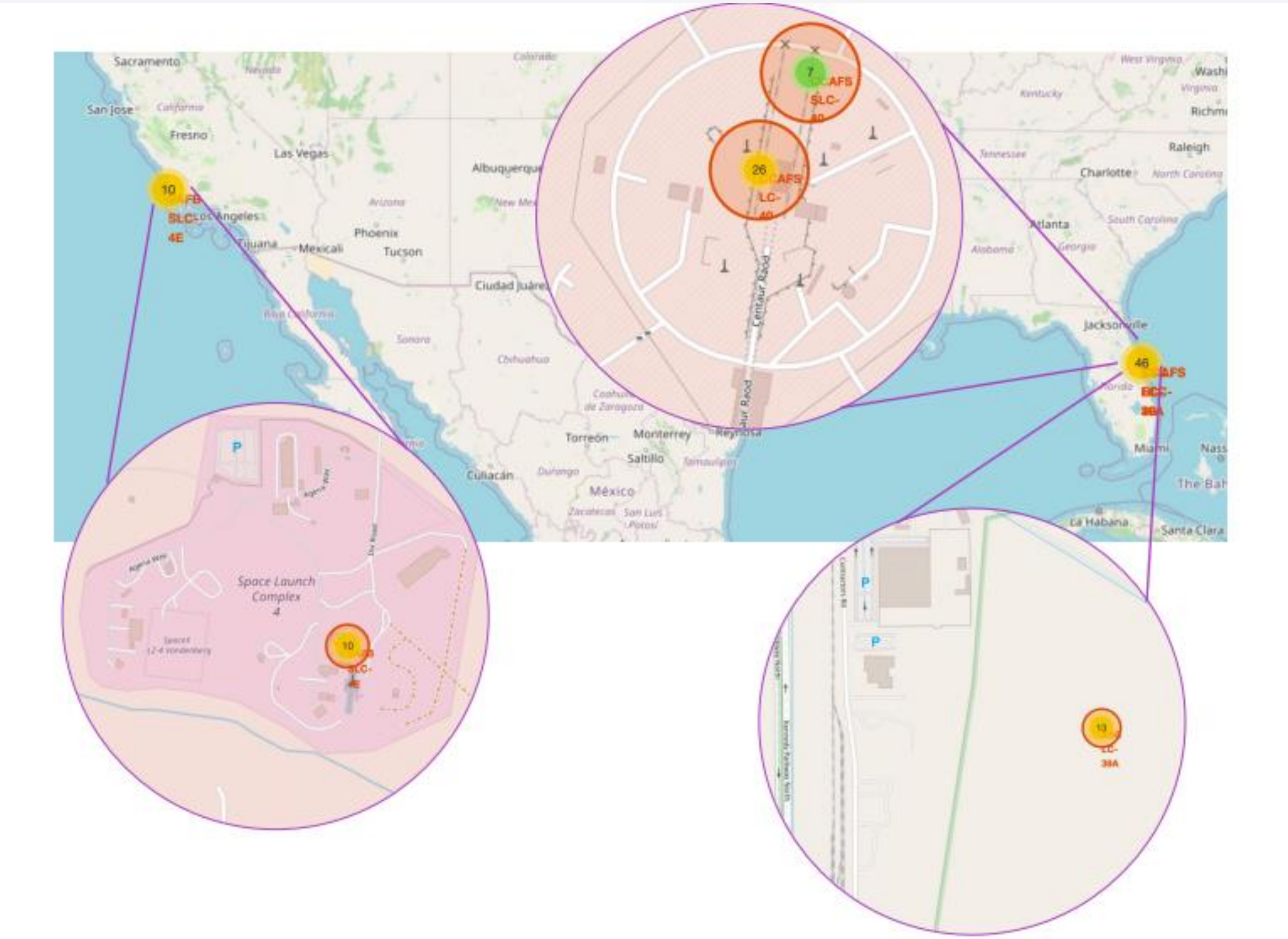
Landing _Outcome	qty
Success	20
No attempt	10
Success (drone ship)	8
Success (ground pad)	6
Failure (drone ship)	4
Failure	3
Controlled (ocean)	3
Failure (parachute)	2
No attempt	1

Ranking landing outcomes between 2010-06-04 and 2017-03-20

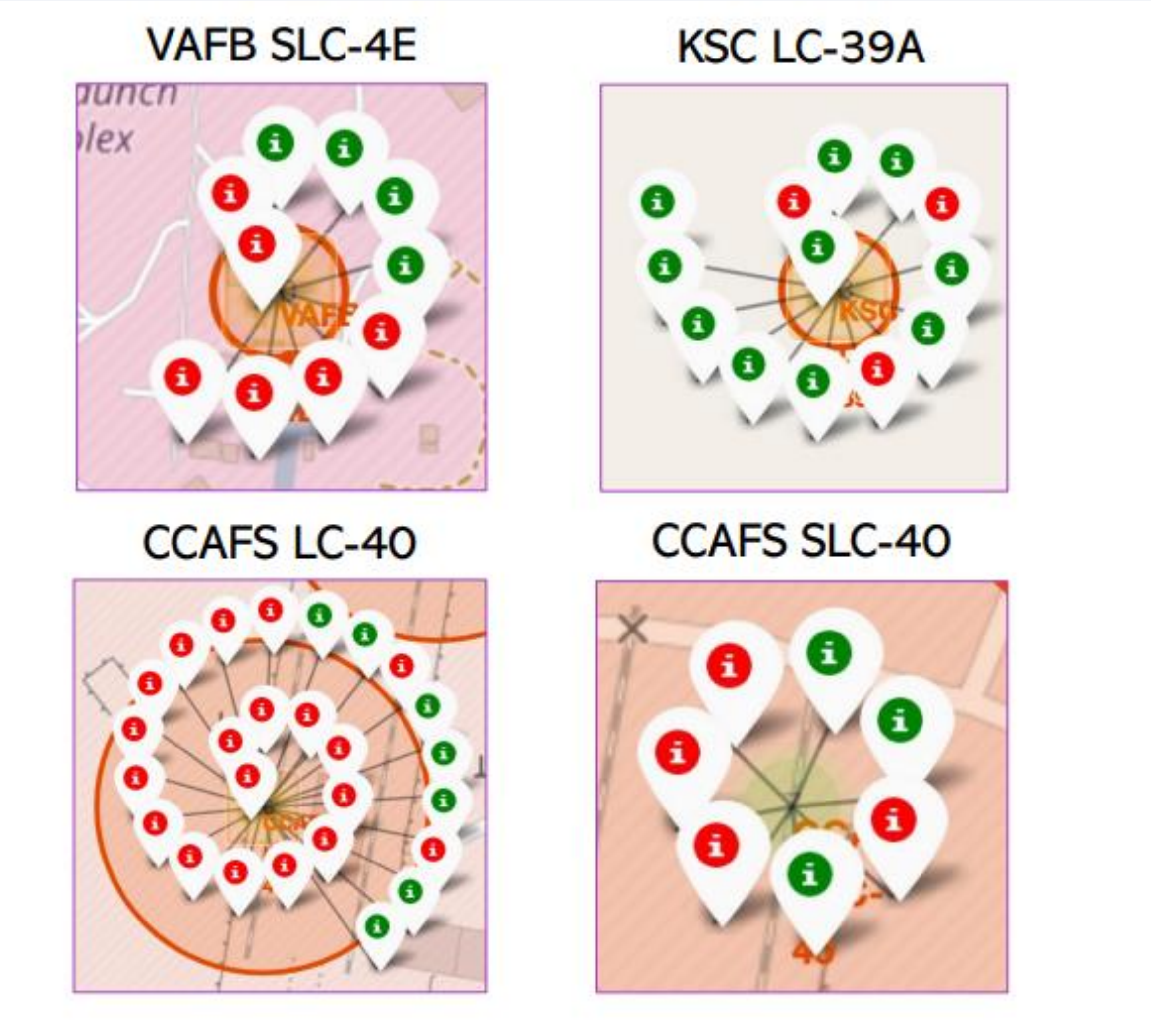
Section 3

Launch Sites Proximities Analysis

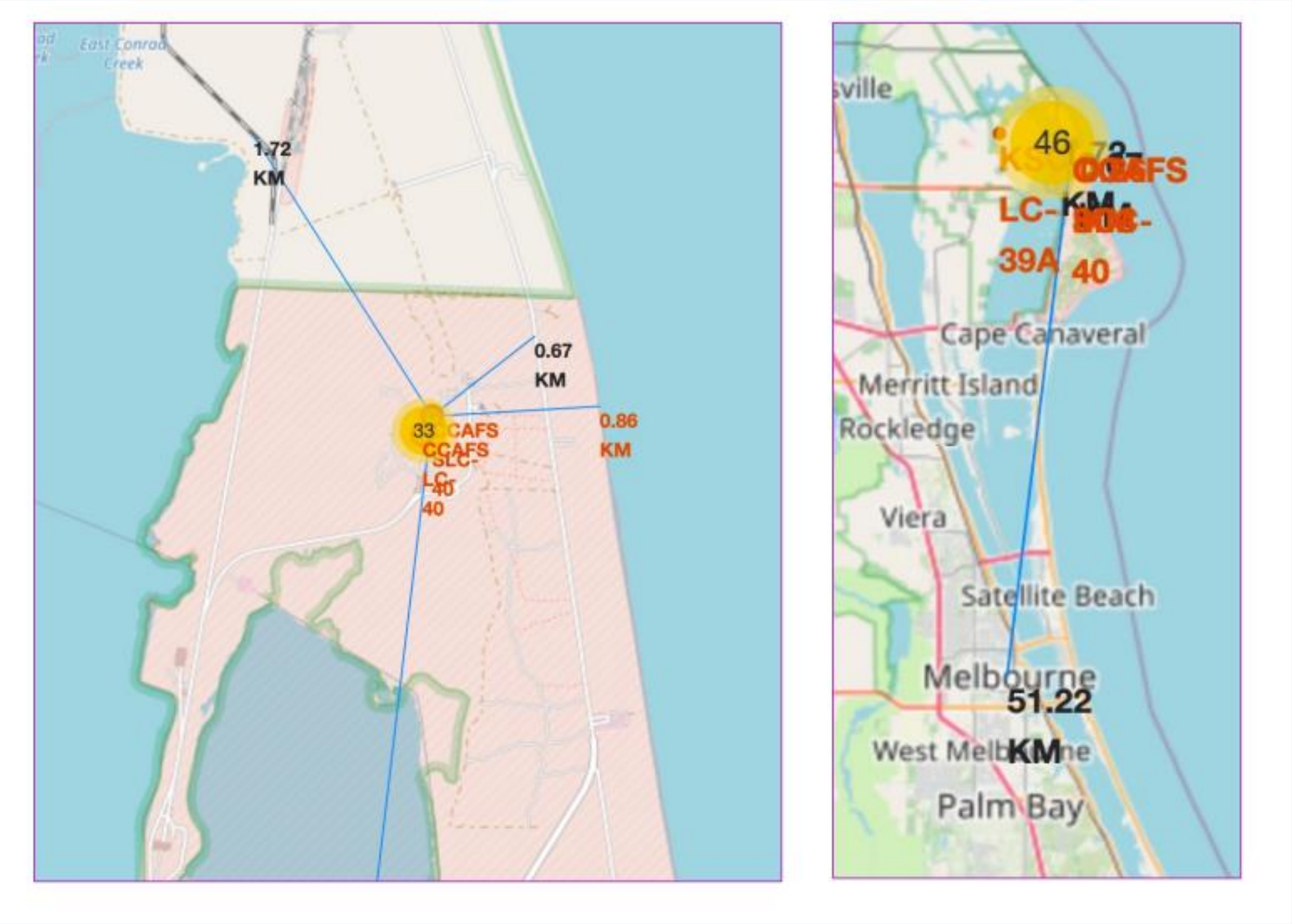
SpaceX launch sites



SpaceX launch site success rate



SpaceX launch site distances from nearby structures



Section 4

Predictive Analysis (Classification)

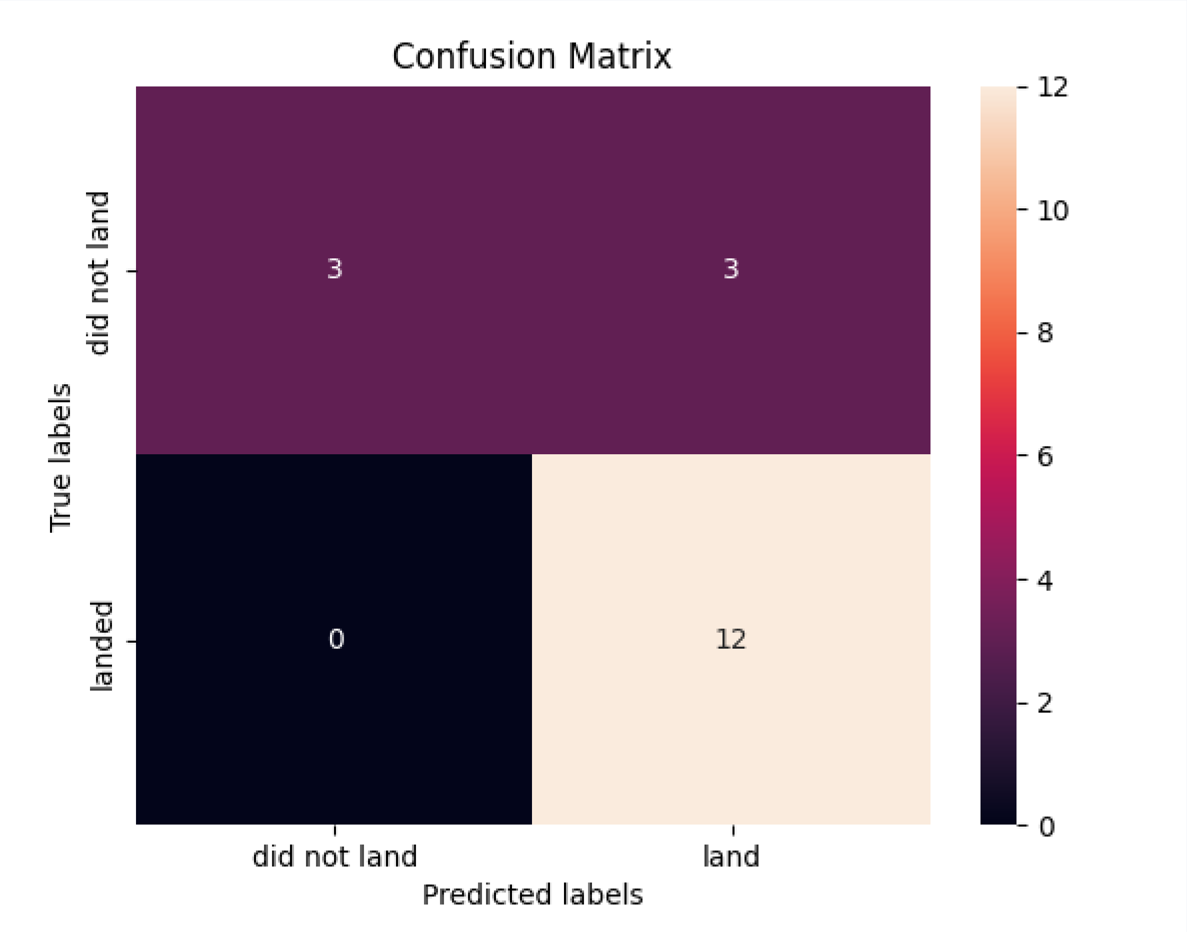
Classification Accuracy

```
df = pd.DataFrame([logreg_cv.best_score_,svm_cv.best_score_,knn_cv.best_score_,tree_cv.best_score_])
index = ["logreg_cv","svm_cv","knn_cv","tree_cv"]
df.index = index
df
```

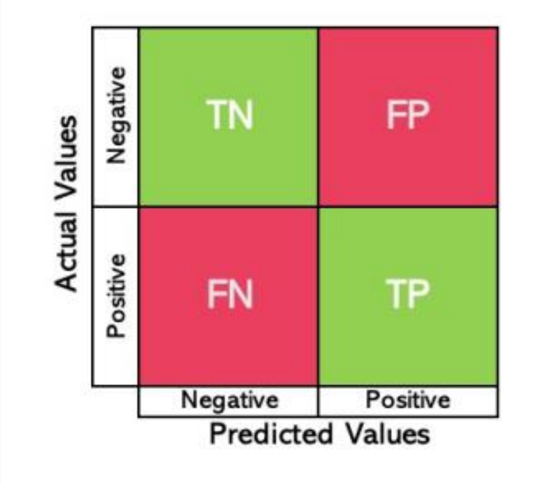
	0
logreg_cv	0.846429
svm_cv	0.848214
knn_cv	0.848214
tree_cv	0.889286

All models have qualitative and similar accuracy but the Decision Tree model performs the best

Confusion Matrix

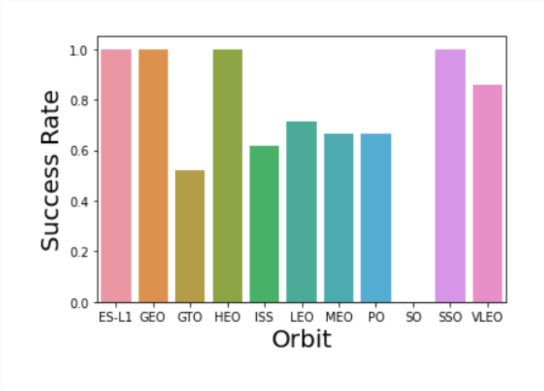


Confusion matrix of the Decision Tree model

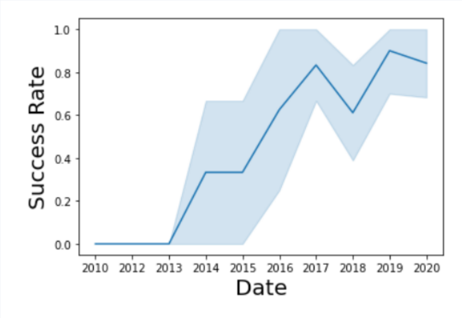


Conclusions

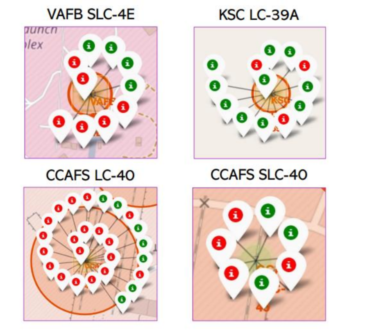
Aiming for ES-L1, GEO, HEO, SSO ensure better success rate

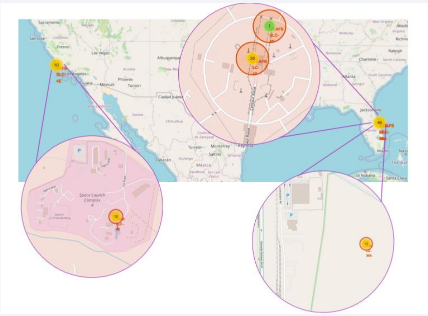


Launch success has improved over time



KSC LC-39A is the launch site that ensures the higher success rate





Decision Tree model performs the best to analyse the data

logreg_cv	0.846429
svm_cv	0.848214
knn_cv	0.848214
tree_cv	0.889286

Thank you!