IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Franzisca Gorniacyzk
19.09.2022

Outline

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

Outline

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

Executive Summary

- Topic

  Evaluate SpaceX's success in launching their Falcon 9 rockets and then returning the booster unharmed, to be reused for further launches.

- Approach

  Gather data on previous Falcon 9 missions, analyze results and build predictive models.

- Methodology

  Gather data from remote sources/Internet, perform data wrangling, visualize using tables, graphs and maps, and use classification models to predict landing outcomes.

- Results

  Find how launch site, payload, booster version and Orbit type affect the landing outcome.

- Conclusion

  Classification models have an 80% accuracy in prediction the success of future landing attempts.

Outline

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

Introduction

- SpaceX:

  American aerospace company building rockets to transport cargo and humans (payload) into orbits around earth or to other planetoids (moons and planets)

- Falcon 9 rocket:

  SpaceX's partially reusable rocket type, consisting of booster (responsible for lift-off) and second stage (responsible for carrying payload to final destination).

- Booster:

  Falcon 9's booster have the ability to land again after separating from the second stage. They can be reused for further launches. This results in a significant reduction of expenses.

- Question:

  Which parameters are essential for a successful landing of the Falcon 9 boosters and can we predict the outcomes of future landing attempts?

Outline

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

Outline

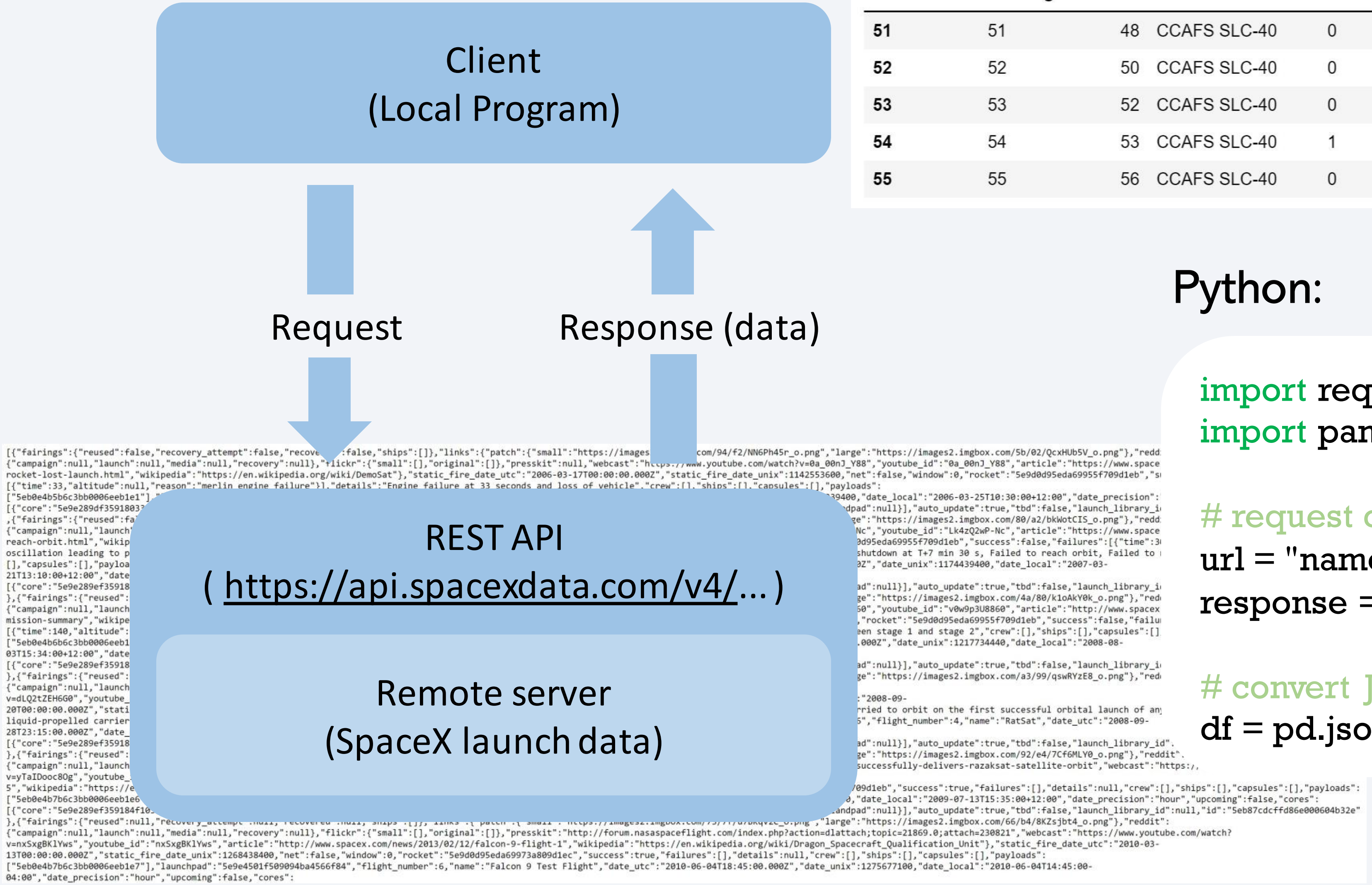
- Executive Summary
- Introduction
- Methodology
  - Data collection
  - Data wrangling
  - Exploratory data analysis (EDA)
  - Interactive visual analytics
  - Predictive analysis
- Results
- Conclusion
- Appendix

Outline

- Executive Summary
- Introduction
- Methodology
    - Data collection
    - Data wrangling
    - Exploratory data analysis (EDA)
    - Interactive visual analytics
    - Predictive analysis
- Results
- Conclusion
- Appendix

Data Collection – SpaceX API

Github:
https://github.com/franziscaG/Coursera_Capstone_Project/blob/master/jupyter-labs-spacex-data-collection-api.ipynb



	Unnamed: 0	Flight Number	Launch Site	class	Payload Mass (kg)	Booster Version	Booster Version Category
51	51	48	CCAFS SLC-40	0	4230.0	F9 FT B1032.2	FT
52	52	50	CCAFS SLC-40	0	6092.0	F9 B4 B1044	B4
53	53	52	CCAFS SLC-40	0	2647.0	F9 B4 B1039.2	B4
54	54	53	CCAFS SLC-40	1	362.0	F9 B4 B1045.1	B4
55	55	56	CCAFS SLC-40	0	5384.0	F9 B4 B1040.2	B4

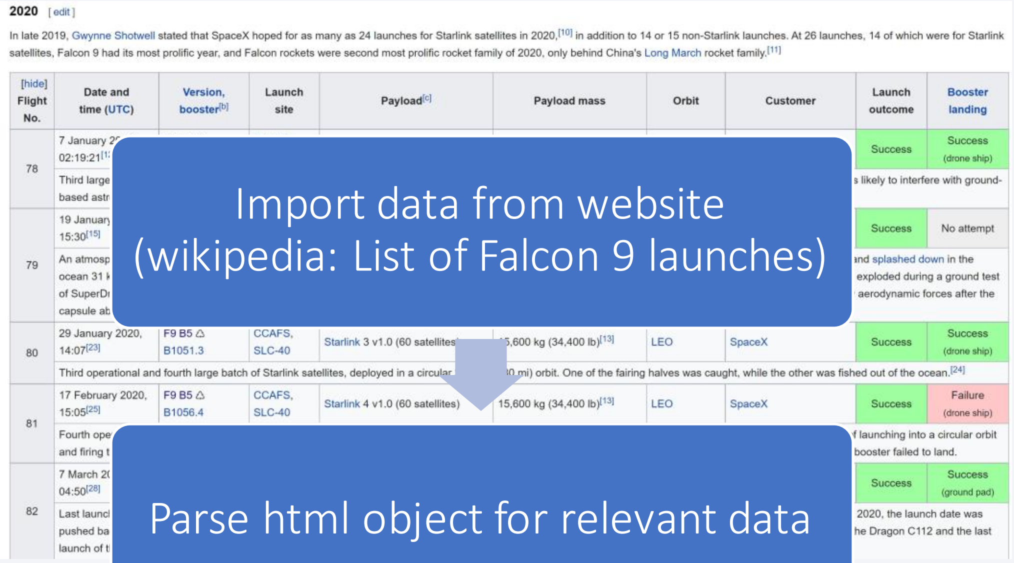
Python:

```
import requests
import pandas as pd

# request data from API
url = "name_of_REST_API"
response = requests.get(url)

# convert JSON to pandas dataframe
df = pd.json_normalize(response)
```

Github:

https://github.com/franziscaG/Coursera_Capstone_Project/blob/master/jupyter-labs-spacex-data-wrangling.ipynb

Data Collection – Web Scraping



Import data from website (wikipedia: List of Falcon 9 launches)

Parse html object for relevant data

Convert to readable dataframe

	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	SpaceX	Success	F9 v1.0	Failure	4 June 2010	18:45
1	2	CCAFS	Dragon	0	LEO	NASA	Success	F9 v1.0	Failure	8 December 2010	15:43
2	3	CCAFS	Dragon	525 kg	LEO	NASA	Success	F9 v1.0	No attempt	22 May 2012	07:44
3	4	CCAFS	SpaceX CRS-1	4,700 kg	LEO	NASA	Success	F9 v1.0	No attempt	8 October 2012	00:35
4	5	CCAFS	SpaceX CRS-2	4,877 kg	LEO	NASA	Success	F9 v1.0	No attempt	1 March 2013	15:10

Python:

```
from bs4 import BeautifulSoup

# request data from API
url = "name_of_website"
response = requests.get(url)

# convert  to BeautifulSoup object
soup = BeautifulSoup(response.text)

# find tables
html_tables  = Soup.find_all('table')

# iterate  through  table  elements  and  add  to  new
pandas  dataframe
df = ...
```

Outline

- Executive Summary
- Introduction
- Methodology
    - Data collection
    - Data wrangling
    - Exploratory data analysis (EDA)
    - Interactive visual analytics
    - Predictive analysis
- Results
- Conclusion
- Appendix

Data Wrangling

Github:

https://github.com/franziscaG/Coursera_Capstone_Project/blob/master/jupyter-labs-spacex-data-wrangling.ipynb

- Data type and information

```
df.info()
```

```
Data columns (total 17 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   FlightNumber    90 non-null     int64
 1   Date            90 non-null     object
```

- Data statistics

```
df.describe()
```

	FlightNumber	PayloadMass	Flights
count	90.000000	90.000000	90.000000
mean	45.500000	6104.959412	1.788889

- Data grouping

```
df['column'].value_counts()
```

	Count
CCAFS SLC 40	55
KSC LC 39A	22
VAFB SLC 4E	13

```
df.groupby('column').mean()
```

LaunchSite	PayloadMass
CCAFS SLC 40	5548.207786
KSC LC 39A	7606.450856
VAFB SLC 4E	5919.461538

```
# classify outcome into good (1) and bad (0) into column 'class'
```

4	5	None None	0
5	6	None None	0
6	7	True Ocean	1
7	8	True Ocean	1

Target variable CLASS

Outline

- Executive Summary
- Introduction
- Methodology
    - Data collection
    - Data wrangling
    - Exploratory data analysis (EDA)
    - Interactive visual analytics
    - Predictive analysis
- Results
- Conclusion
- Appendix

Github:

https://github.com/franziscaG/Coursera_Capstone_Project/blob/master/jupyter-labs-eda-dataviz.ipynb

EDA with Data Visualization

```
import seaborn as sns
```

- Scatter plots: *CLASS vs Payload mass, Flight number, Launch Site and Orbit*
  → Answers which variable affects the landing outcome

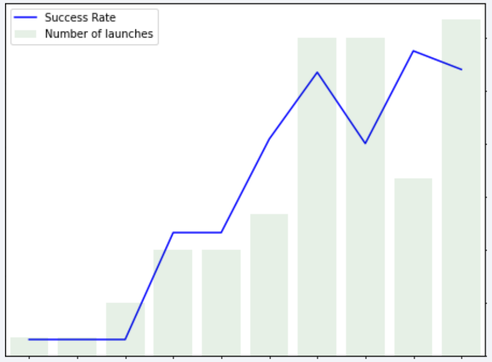
```
sns.catplot(x='FlightNumber',y='LaunchSite',data=df,hue='Class')
```

- Bar plot: *Success rate vs Orbit*
  → Shows which orbit leads to the most successful landings

```
sns.catplot(x='Orbit',y='SuccessRate',data=bar_df,kind='bar')
```

- Line plot: *Success rate vs Year*
  → Shows the improvements over the years

```
sns.lineplot(x='Year',y='SuccessRate',data=line_df)
```



EDA with SQL

Github:

https://github.com/franziscaG/Coursera_Capstone_Project/blob/master/jupyter-labs-eda-sql-coursera_sqllite.ipynb

```
import sqlite3

# create local database and load data into it
connection = sqlite3.connect('database_name')
df = read.csv('data_repository')
df.to_sql('SPACEXTBL',connection)
```

```
# sql magic
%load_ext sql
%sql lite:///database_name.db
```

Ex. Find number of successful landings

```
%%sql
SELECT
    CASE
        WHEN [Landing _Outcome] LIKE '%Success%' THEN 'Success'
        ELSE 'Failure'
    END AS landing_outcome,
    COUNT(Date)  AS count
    FROM SPACEXTBL GROUP BY landing_outcome
```

landing_outcome	count
Failure	40
Success	61

Outline

- Executive Summary
- Introduction
- Methodology
    - Data collection
    - Data wrangling
    - Exploratory data analysis (EDA)
    - Interactive visual analytics
    - Predictive analysis
- Results
- Conclusion
- Appendix

Github:

https://github.com/franziscaG/Coursera_Capstone_Project/blob/master/lab_jupyter_launch_site_location.ipynb

Build an Interactive Map with Folium

```
import folium
from folium.plugins import MarkerCluster,MousePosition
from folium.features import DivIcon

site_map = folium.Map(location, zoom_start)
```

- Circles and markers: to mark launch sites

```
circle = folium.Circle(coordinates).add_child(folium.Popup(label))
```

- Marker-cluster: to mark launches (green – successful landing, red – failure)

```
marker=folium.Marker(coordinates, icon)
```

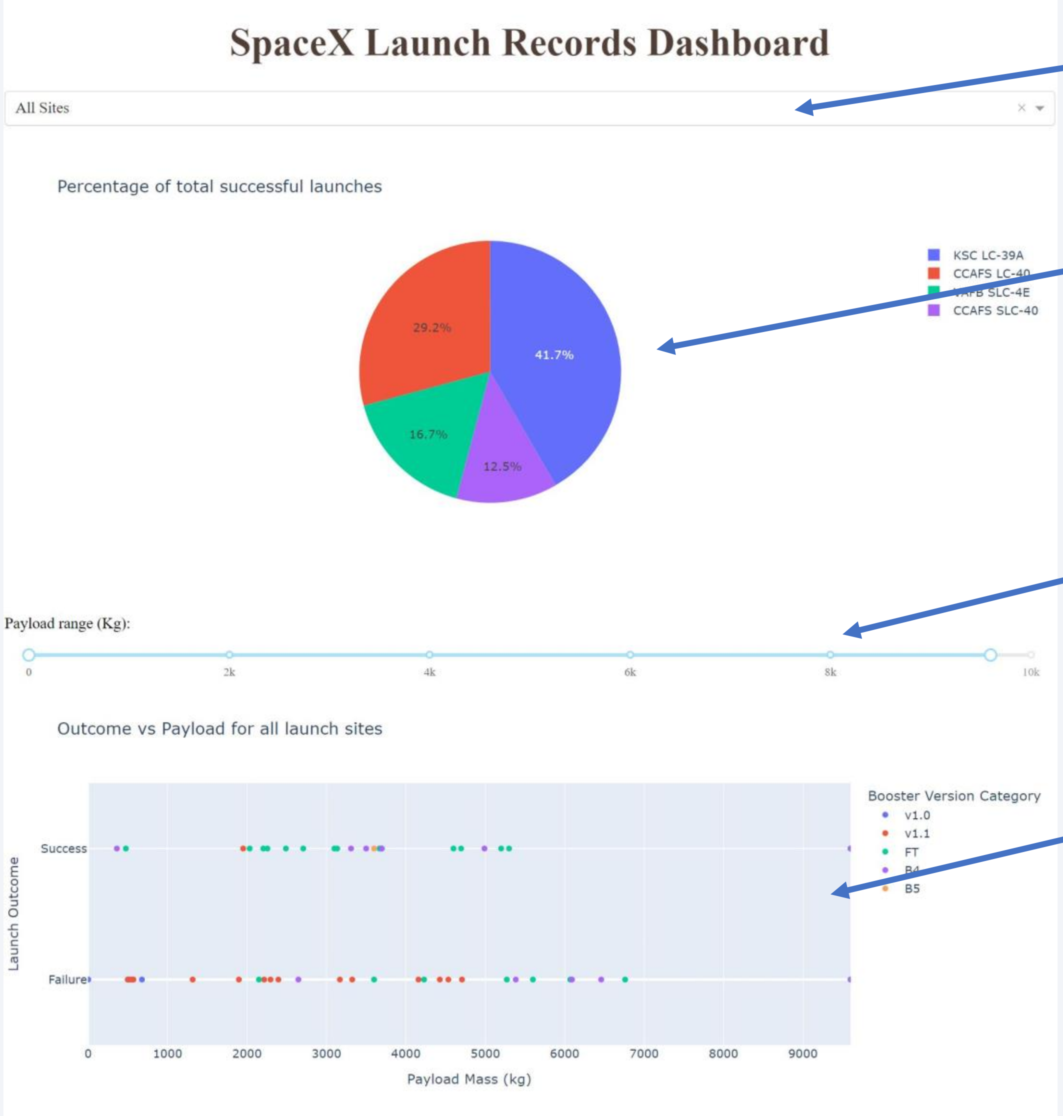
- Lines and markers: to mark closest coast, railway, highway and city

```
line=folium.PolyLine(locations=[start,end])
```

Github:

https://github.com/franziscaG/Coursera_Capstone_Project/blob/master/Dash_application.ipynb

Build a Dashboard with Plotly Dash



Dropdown menu:
to choose Launch Site

refer to notebook on how to build a dash application in Python

Pie chart:
to display the percentage of successful landings

Range Slider:
to choose a range of Payload Mass

Scatter Plot:
to display Landing Outcome vs Payload Mass

Outline

- Executive Summary
- Introduction
- Methodology
  - Data collection
  - Data wrangling
  - Exploratory data analysis (EDA)
  - Interactive visual analytics
  - Predictive analysis
- Results
- Conclusion
- Appendix

Predictive Analysis (Classification)

Github: https://github.com/franziscaG/Coursera_Capstone_Project/blob/master/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Define target variable Y and dependent variables X

Normalize X

Split X and Y in training and testing sets

Grid search on training set

- Split into n folds
- Choose model parameters
- Choose n-1 folds
- Fit model
- Calculate accuracy for nth fold
- Average over n-accuracies

Iterate over choice of n-1 folds and model para meters

Best model parameters

Calculate accuracy on testing set

```
from sklearn import preprocessing
from sklearn.model_selection import train_test_split, GridSearchCV

transform = preprocessing.StandardScaler()
X = transform.fit_transform(X)

X_train,Y_train,X_test, Y_test = train_test_split(X,Y,test_size)

model=GridSearchCV(model_object,parameters, n_folds)
```


```
accuracy = model.fit(X_train,Y_train).score(X_test,Y_test)
```

Outline

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

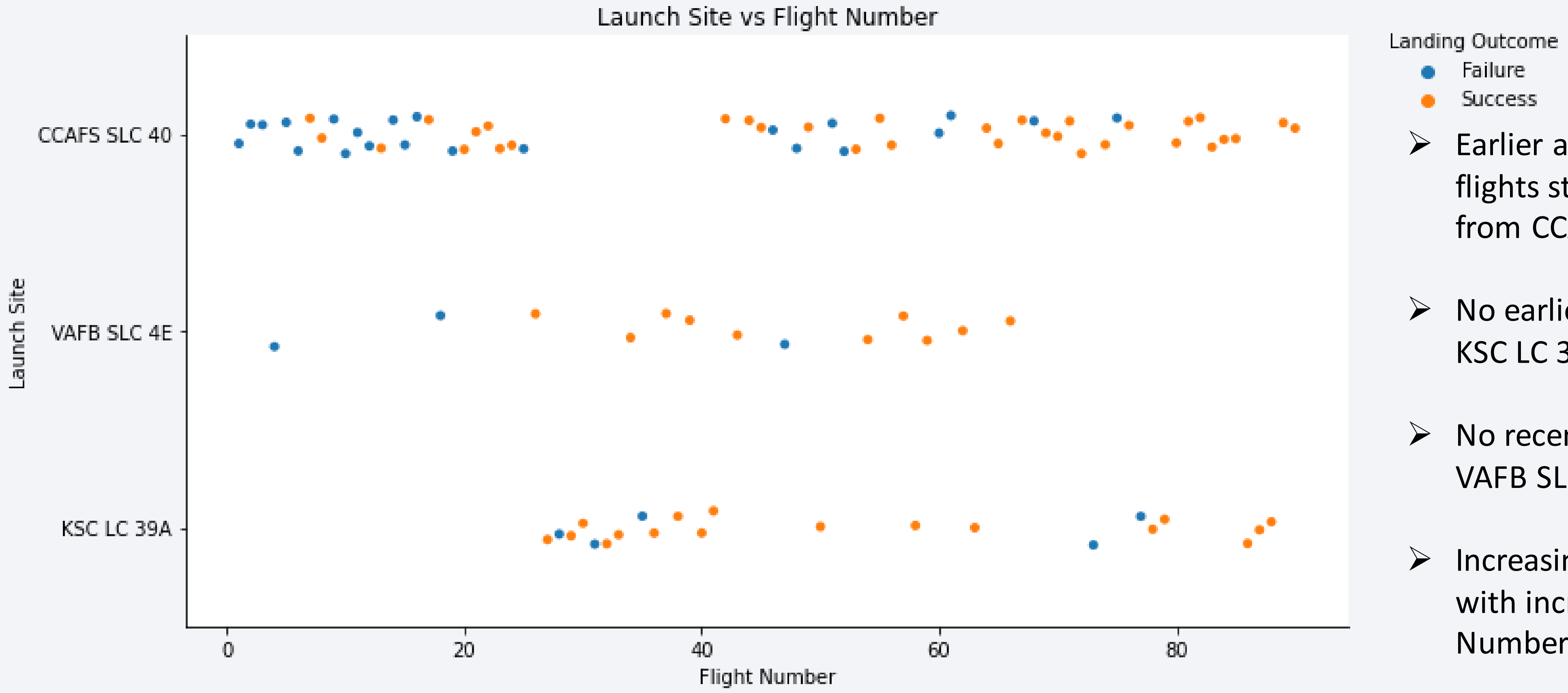
Outline

- Executive Summary
- Introduction
- Methodology
- Results
    - Exploratory data analysis (EDA)
    - Interactive visual analytics
    - Predictive analysis
- Conclusion
- Appendix

Outline

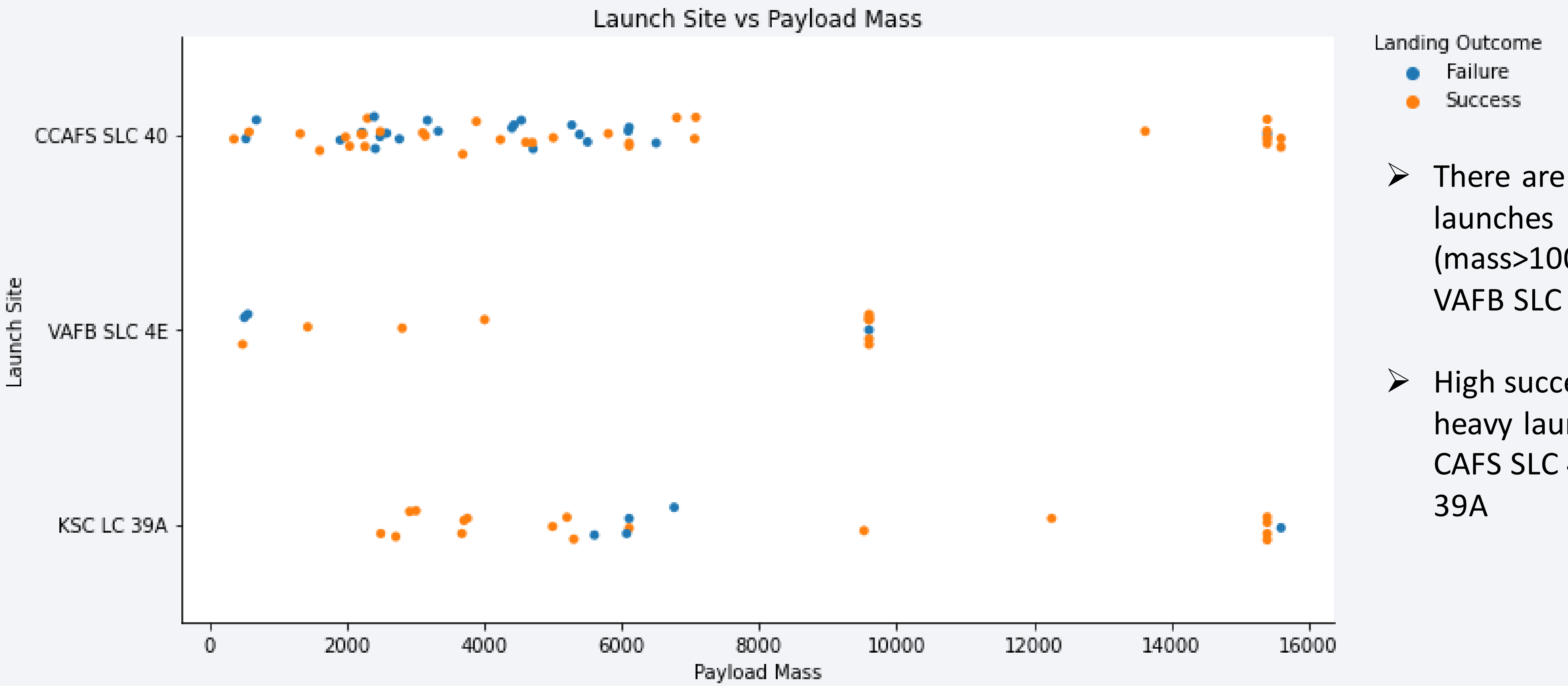
- Executive Summary
- Introduction
- Methodology
- Results
    - Exploratory data analysis (EDA)
    - Interactive visual analytics
    - Predictive analysis
- Conclusion
- Appendix

Flight Number vs. Launch Site



- Earlier and most recent flights starting mostly from CCAFS SLC 40
- No earlier flights from KSC LC 39A
- No recent flights from VAFB SLC 4E
- Increasing success rate with increasing Flight Number

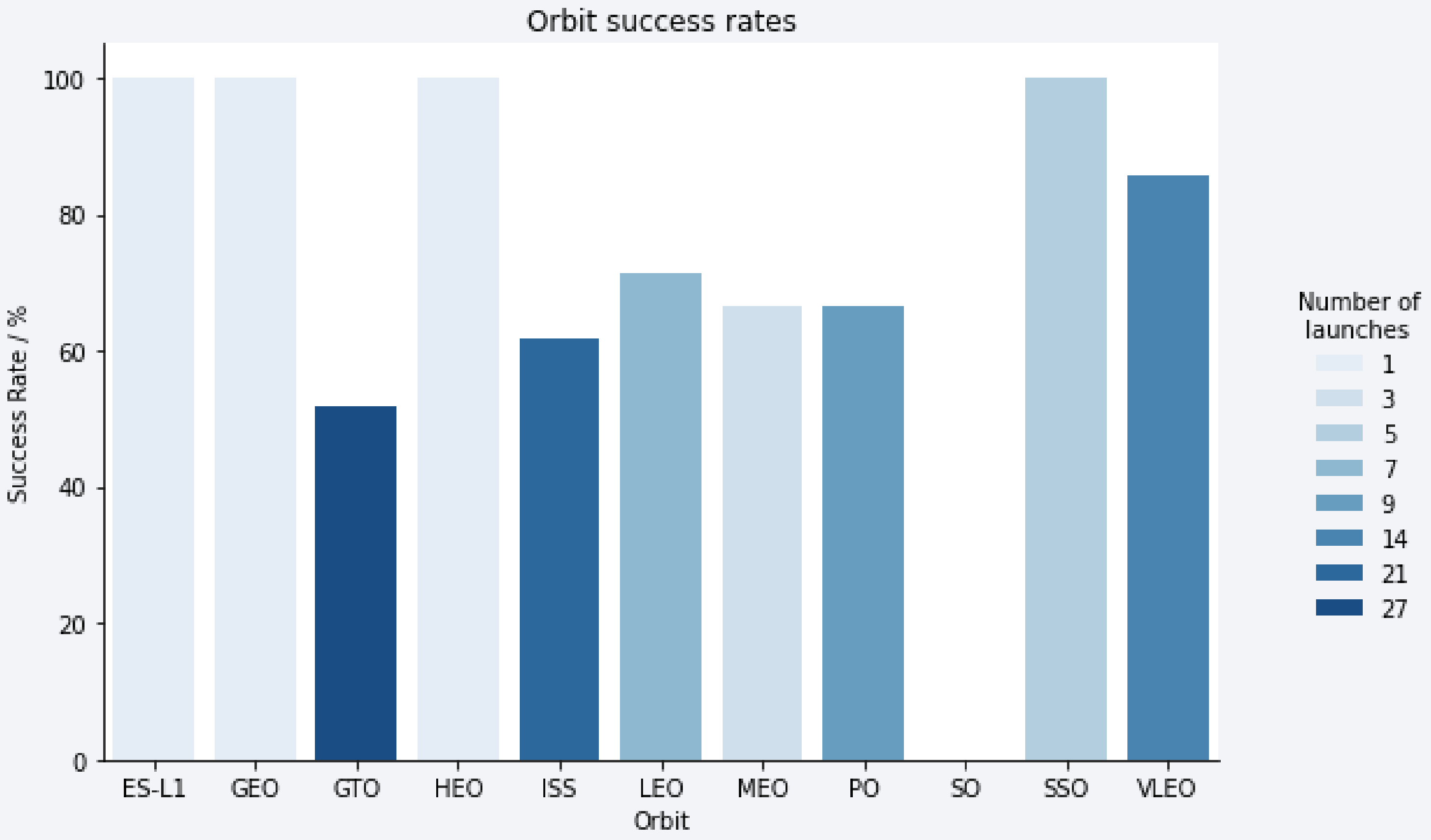
Payload vs. Launch Site



- There are no heavy launches (mass>10000kg) from VAFB SLC 4E

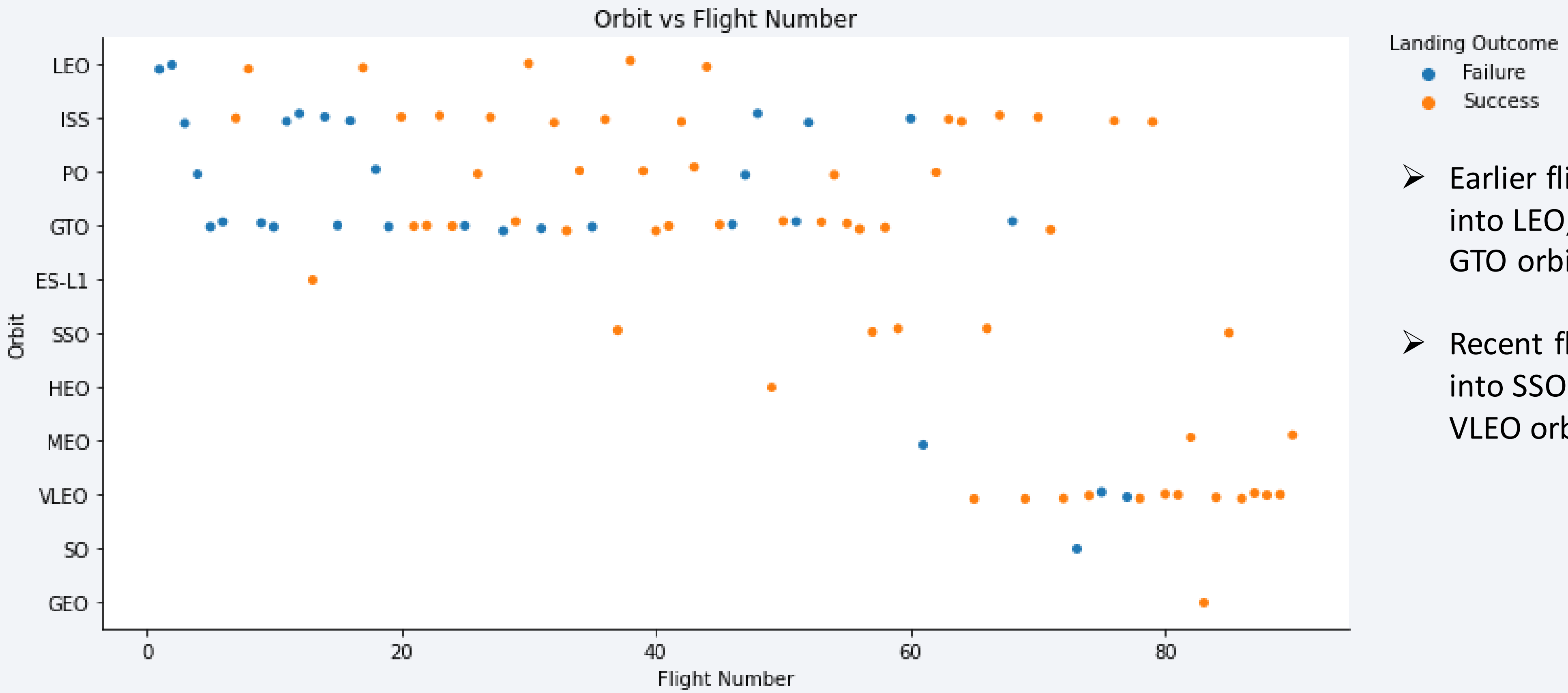
- High success rate for heavy launches from CAFS SLC 40 and KSC LC 39A

Success Rate vs. Orbit Type



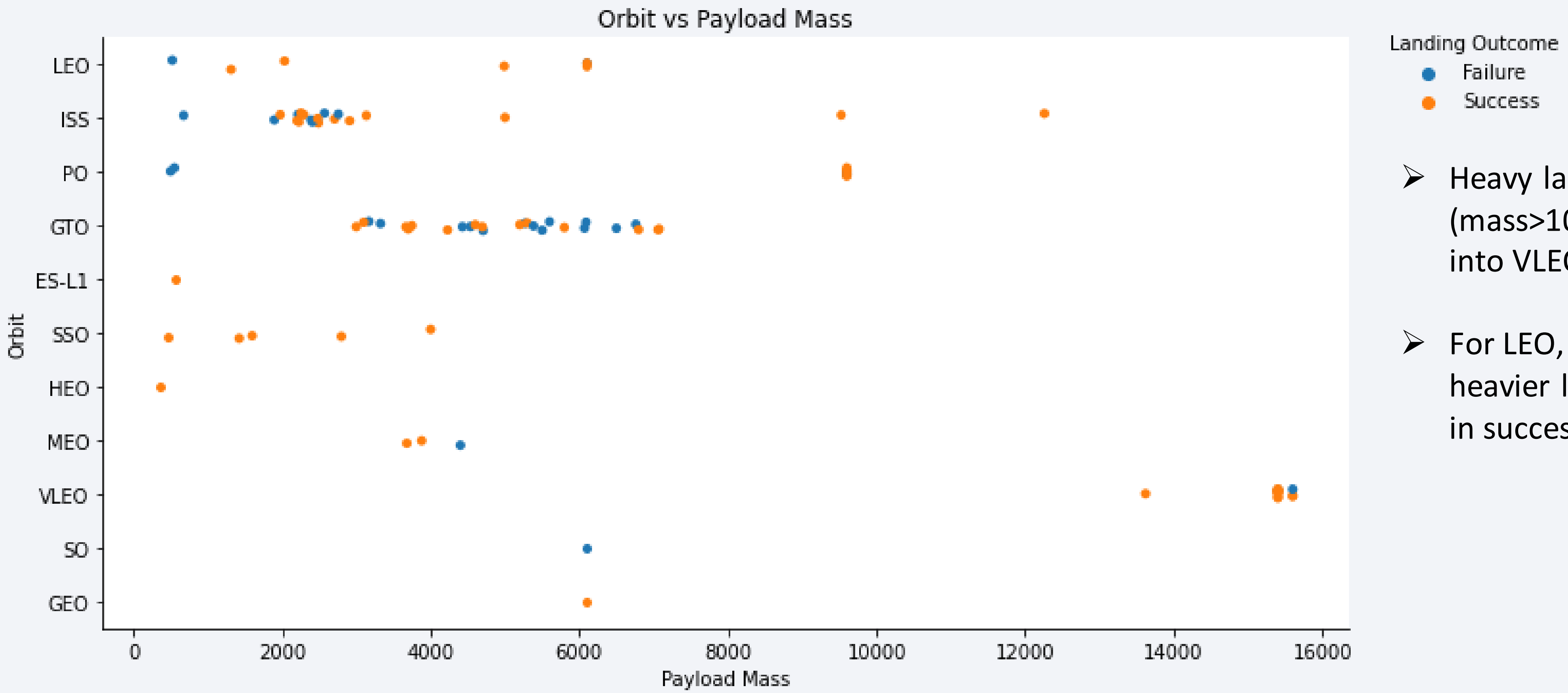
- High success rates for ES-L1, GEO, HEO and SSO orbits, but only few launches in these
- Most launches into GTO orbit, but only about 50% success rate
- Most promising orbit is VLEO with 14 launches and about 85% success rate

Flight Number vs. Orbit Type



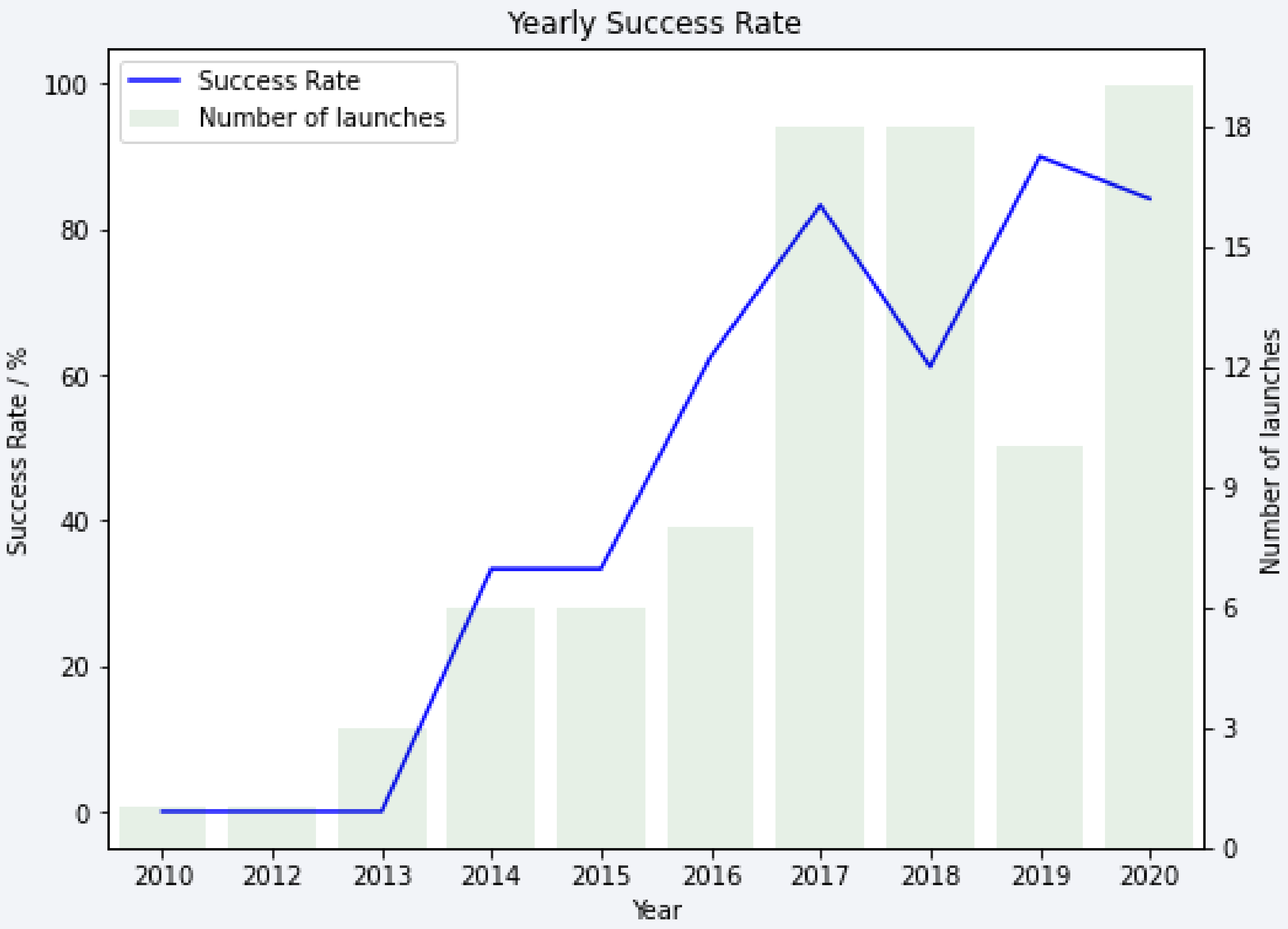
- ➢ Earlier flights mostly into LEO, ISS, PO and GTO orbits
- ➢ Recent flights mostly into SSO, MEO, and VLEO orbits

Payload vs. Orbit Type



- Heavy launches (mass>10000kg) only into VLEO orbit
- For LEO, ISS and PO heavier launches result in successful landings

Launch Success Yearly Trend



- Non-monotonous increase in success rate between the years of 2013 to 2020

Launch Site Names

All unique launch sites

Launch_Site	count
CCAFS LC-40	26
CCAFS SLC-40	34
KSC LC-39A	25
VAFB SLC-4E	16

Five launches from CCAFS location

Date	Launch_Site	Landing _Outcome
04-06-2010	CCAFS LC-40	Failure (parachute)
08-12-2010	CCAFS LC-40	Failure (parachute)
22-05-2012	CCAFS LC-40	No attempt
08-10-2012	CCAFS LC-40	No attempt
01-03-2013	CCAFS LC-40	No attempt

Payload Mass

Booster_Version	payload_kg
F9 B5 B1048.4	15600
F9 B5 B1048.5	15600
F9 B5 B1049.4	15600
F9 B5 B1049.5	15600
F9 B5 B1049.7	15600
F9 B5 B1051.3	15600
F9 B5 B1051.4	15600
F9 B5 B1051.6	15600
F9 B5 B1056.4	15600
F9 B5 B1058.3	15600
F9 B5 B1060.2	15600
F9 B5 B1060.3	15600

Booster versions that carry maximum payload

NASA	total_payload_kg
no	519987
yes	99980

Total payload carried by all the rockets launched for NASA

F9_v1_1	avg_payload_kg
no	6766.8
yes	2534.7

Average payload carried by the F9 v1.1 booster versions

Successful Landings

landing_outcome	count
Failure	40
Success	61

- Total number of successful and failed landings

Date	Landing _Outcome
01-05-2017	Success (ground pad)

- First successful landing on a ground pad

Booster_Version	payload_kg	Landing _Outcome
F9 FT B1022	4696	Success (drone ship)
F9 FT B1026	4600	Success (drone ship)
F9 FT B1021.2	5300	Success (drone ship)
F9 FT B1031.2	5200	Success (drone ship)

- Successful landings on a drone ship of rockets carrying a payload between 4000 and 6000 kg

Launch Records

year	month	landing	Booster_Version	Launch_Site
2015	01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
2015	04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40
2015	06	Precluded (drone ship)	F9 v1.1 B1018	CCAFS LC-40

Failed landings in 2015

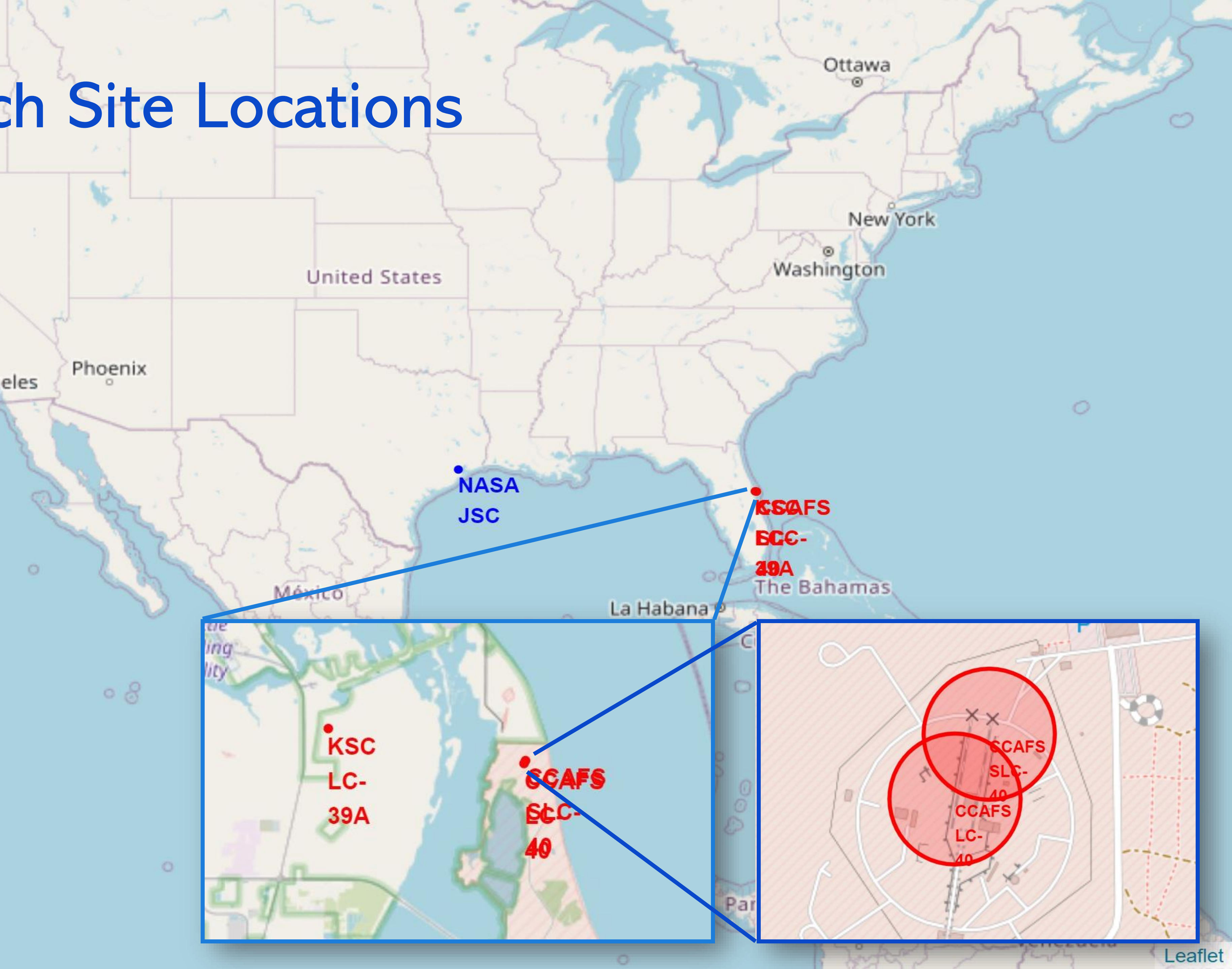
landing_outcome	count
Success	38
No attempt	21
Success (drone ship)	14
Success (ground pad)	9
Failure (drone ship)	5
Controlled (ocean)	5
Failure	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1
No attempt	1

Number of occurrences for each type of landing outcome

Outline

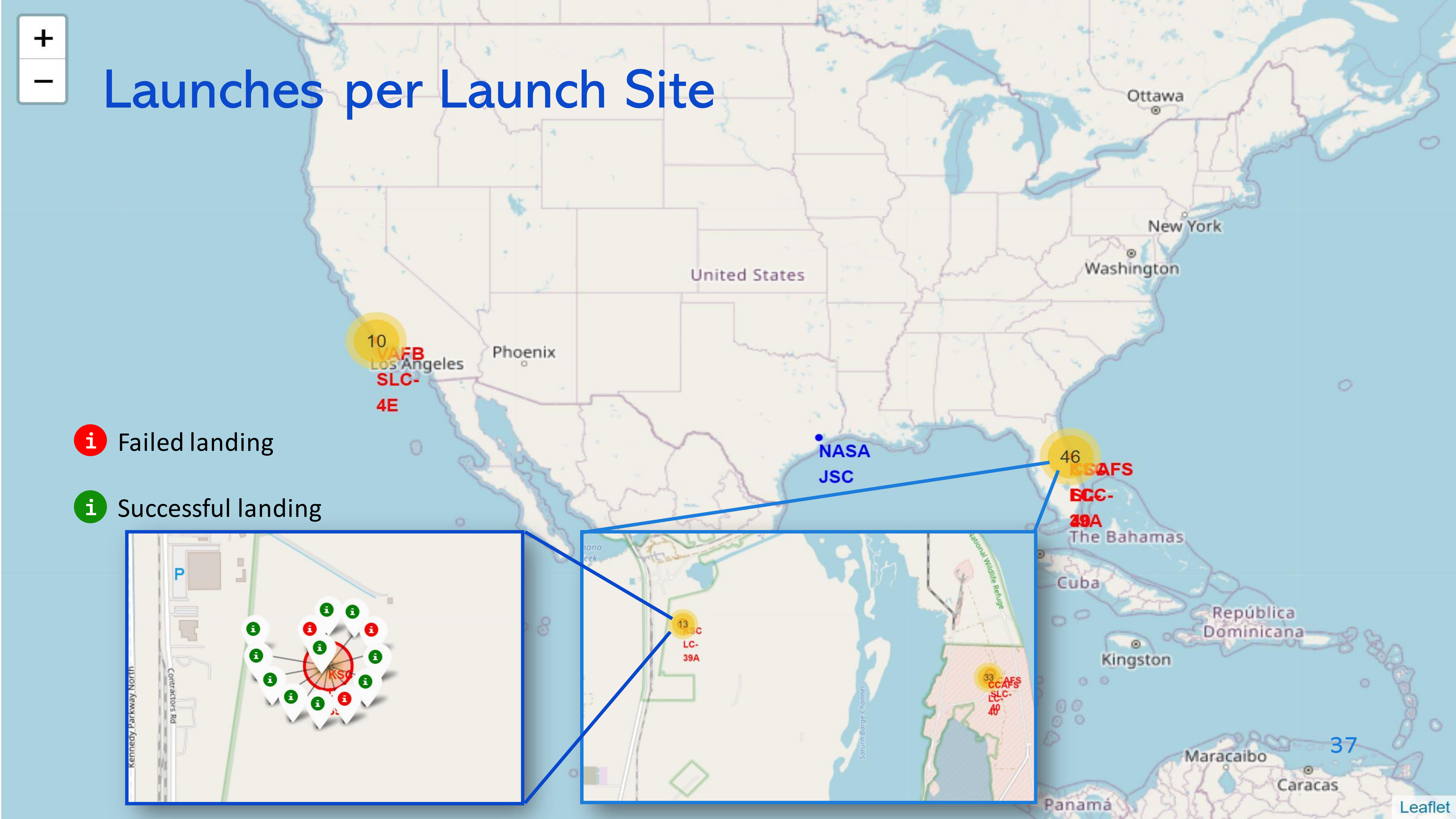
- Executive Summary
- Introduction
- Methodology
- Results
    - Exploratory data analysis (EDA)
    - Interactive visual analytics
    - Predictive analysis
- Conclusion
- Appendix

Map of Launch Site Locations



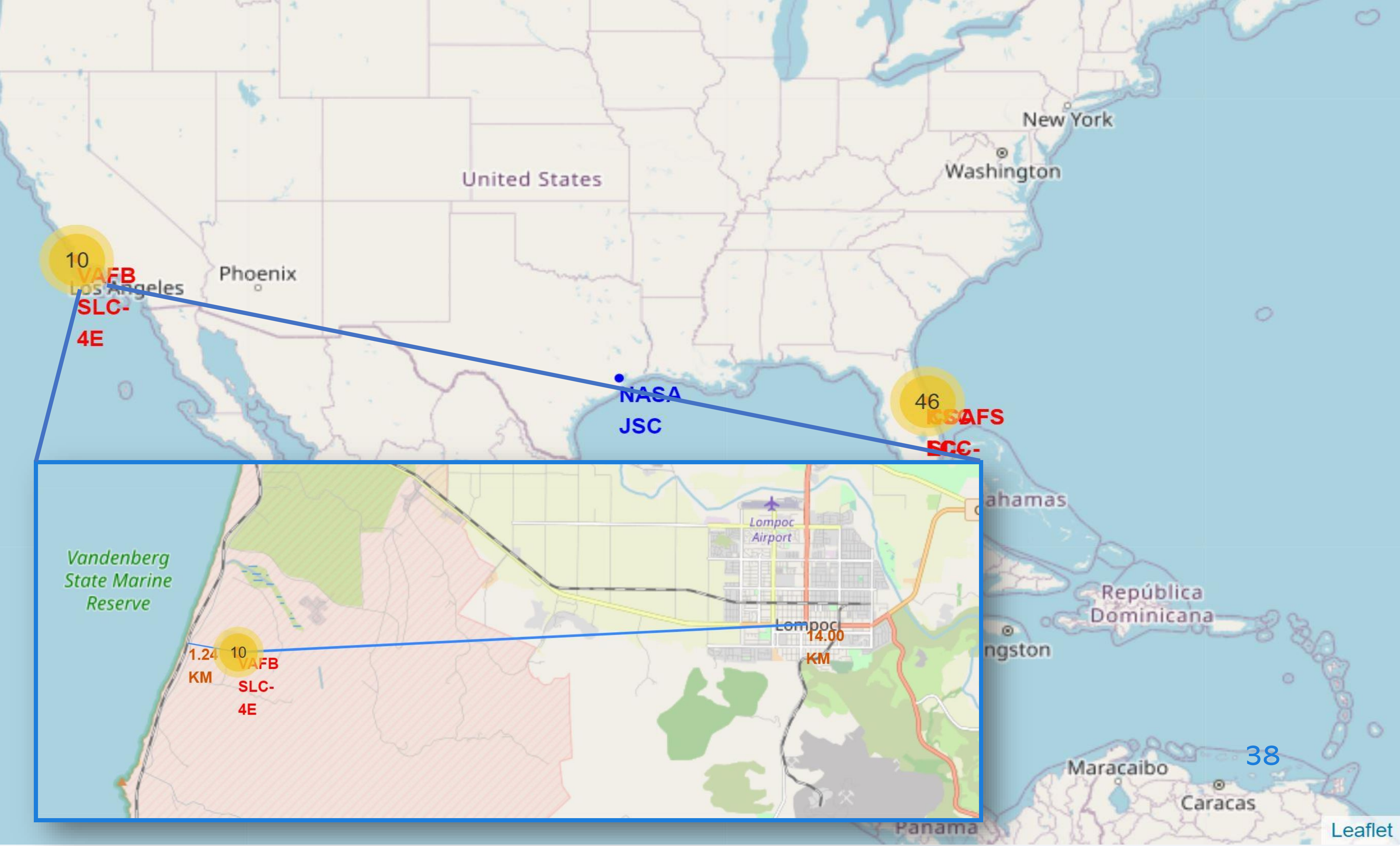
- Close proximity to the coast
  - Large uninhabited area
- Mainly eastern coast
  - Launches towards east, 'help' due to Earth rotation
- Close to equator
  - More 'help'

Launches per Launch Site



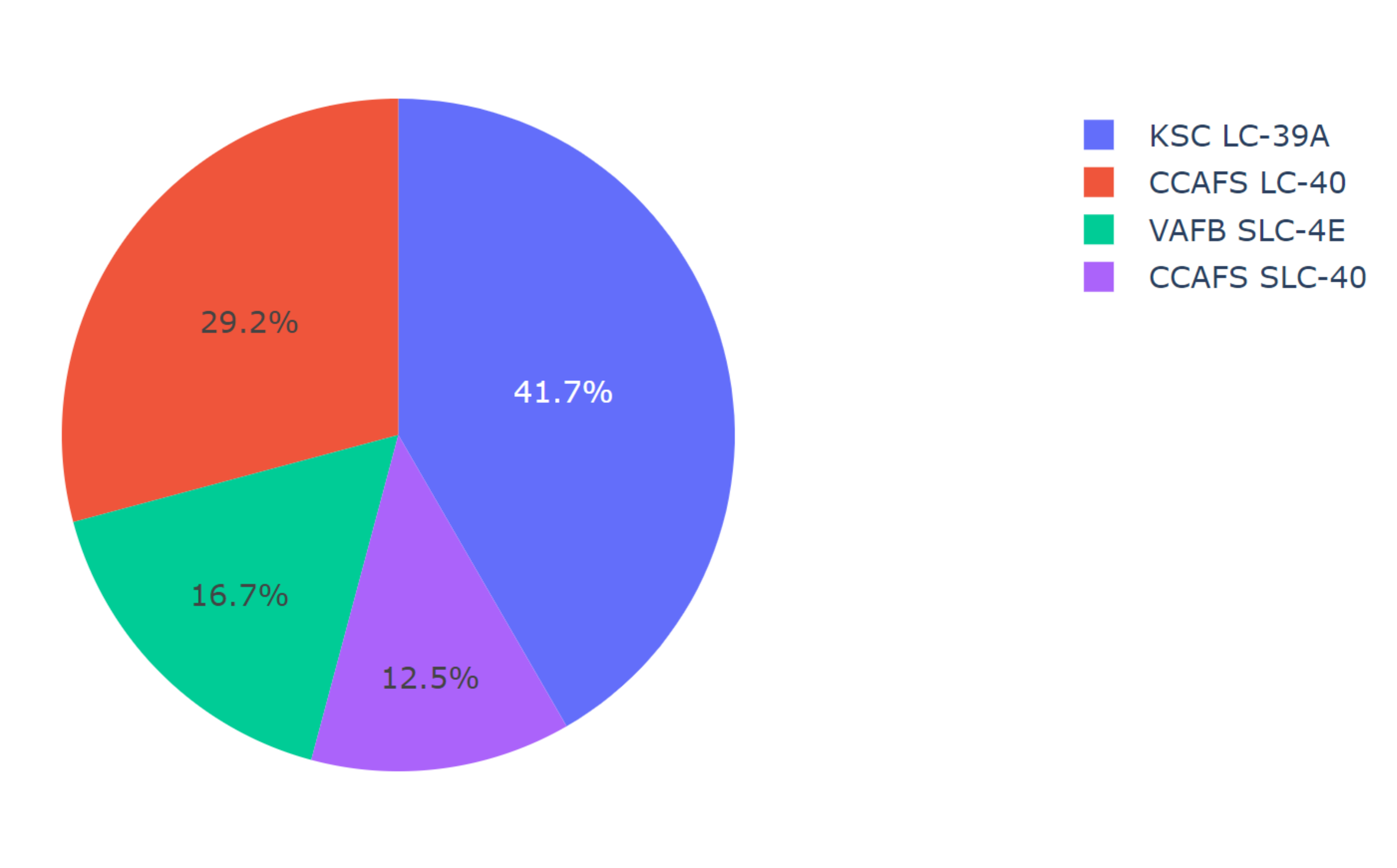
Launch Sites proximity to infrastructure

- ➢ close proximity to highways and railways
  - ➢ Bring people and material
- ➢ Close proximity to coastline
  - ➢ Ocean safe for falling parts
- ➢ Safe distance from settlements
  - ➢ Avoid damage due to failures



Percentage of total successful landings

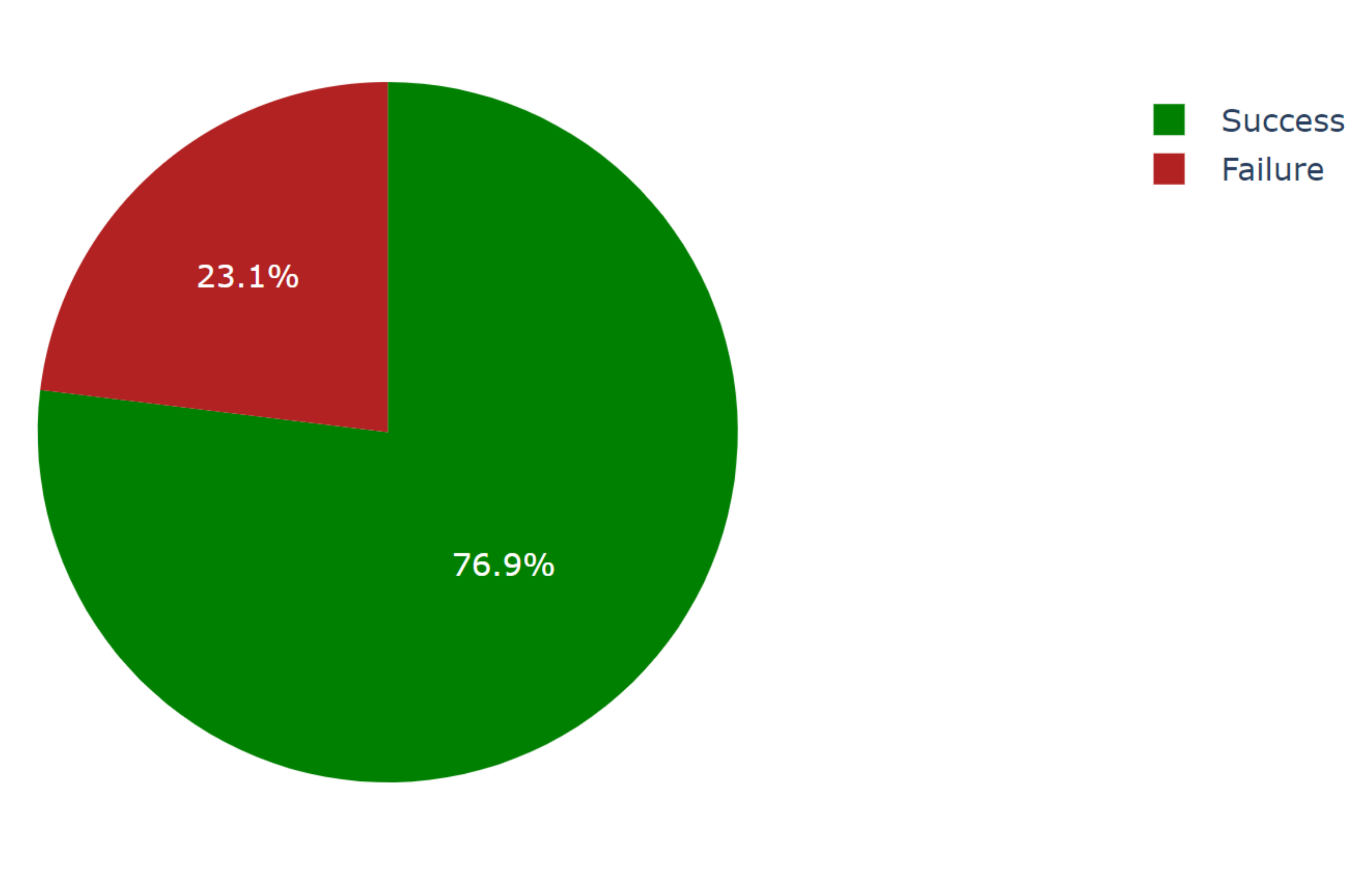
- The launch site KSC LC-39A has the most successful landings (41.7% of the overall successful landings)
- The fewest successful landings are recorded at launch site CCAFS SLC-40 (12.5%)
- These numbers are absolute and not weighed against the total number of landings attempted at each site!



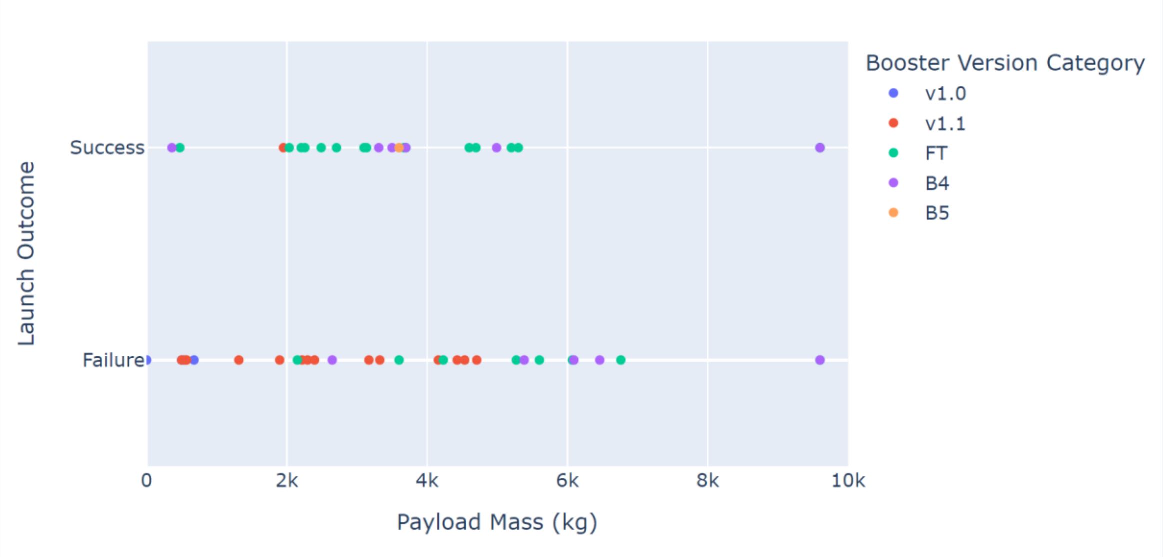
Landing outcome at launch site KSC LC-39A

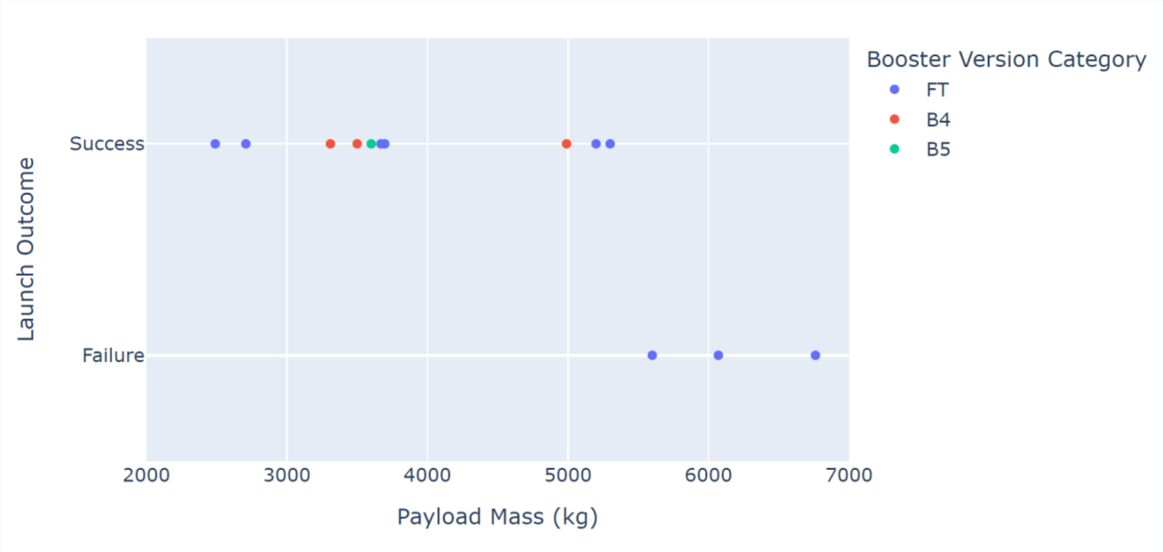
KSC LC-39A has also the highest landing success rate

- Of all the attempted landings at this site 76.9% were successful
- While only 23.1% failed



Lunch Outcome vs Payload





For all launch sites:

- A high percentage of the v1.1 Booster Version resulted in failed landings, independently of payload

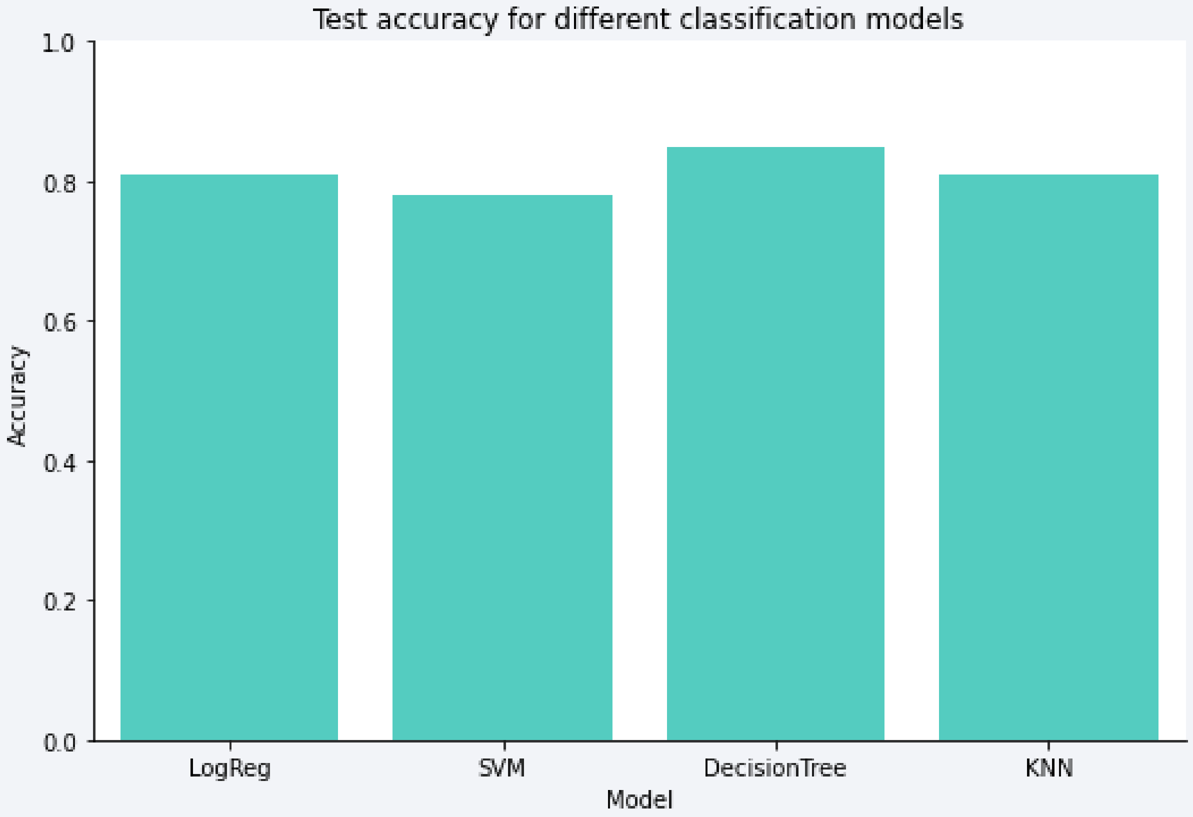
For KSC LC-39A:

- Launches with payload smaller than 5500kg resulted in successful landings
- While heavier launches failed to land successfuly

Outline

- Executive Summary
- Introduction
- Methodology
- Results
  - Exploratory data analysis (EDA)
  - Interactive visual analytics
  - Predictive analysis
- Conclusion
- Appendix

Classification Accuracy



- All classification models have a prediction accuracy of around 0.8+- 0.5
- The highest accuracy has the Decision Tree model (although it also has the highest variance in accuracy – not seen here)

Confusion Matrix – Decision Tree

- For a selected range of launches the model predicts 9 failed landings and 18 successful ones
- Almost all failure predictions (8 out of 9) were indeed failed landings
- And only 3 of the success predictions actually resulted in failures
- Total model accuracy: 0.85

Outline

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

Conclusions

- Most promising orbit is VLEO with 14 launches and about 85% success rate
- Non-monotonous increase in success rate between the years of 2013 to 2020
- Launch Sites: mainly eastern coast; close proximity to transport infrastructure; far from settlements; KSC LC-39A has the most successful landings
- A high percentage of the v1.1 Booster Version resulted in failed landings, independently of payload
- All classification models have a prediction accuracy of around 0.8+- 0.5

Outline

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

Appendix

- Include any relevant assets like Python code snippets, SQL queries, charts, Notebook outputs, or data sets that you may have created during this project

  Yeah – no! Way too many, people would just get confused...

Thank you!