IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Martin Hucík
17.1.2022

Outline

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

Executive Summary

- Summary of methodologies
  - Data collection
  - Data wrangling
  - EDA with SQL and data visualization
  - Visualizations with interactive maps and dashboards
  - Building classification predictive models
- Summary of all results
  - Identified site with highest success rate
  - Orbit, orbit vs payload success rate
  - Best boosters' options
  - Launch sites similarities

Introduction

Project background and context

This project is the final part of IBM Data Science Professional Course. Which aims to demonstrate knowledge of fundamental data science concepts like data gathering, manipulation and visualization and ability to formulate outcomes from given data.

In this capstone project, we will try to predict if the Falcon 9 first stage will land successfully.
SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars what is well bellow other providers cost upward of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage.
Therefore, if we can determine if the first stage will land, we can determine the cost of a launch.

Problems we want to find answers for...

What are the factors increasing probability of successful first stage landing?

Can we build a predictive model to help us predict if the landing will be successful?

Are there features that currently used launch sites have in common?

Section 1

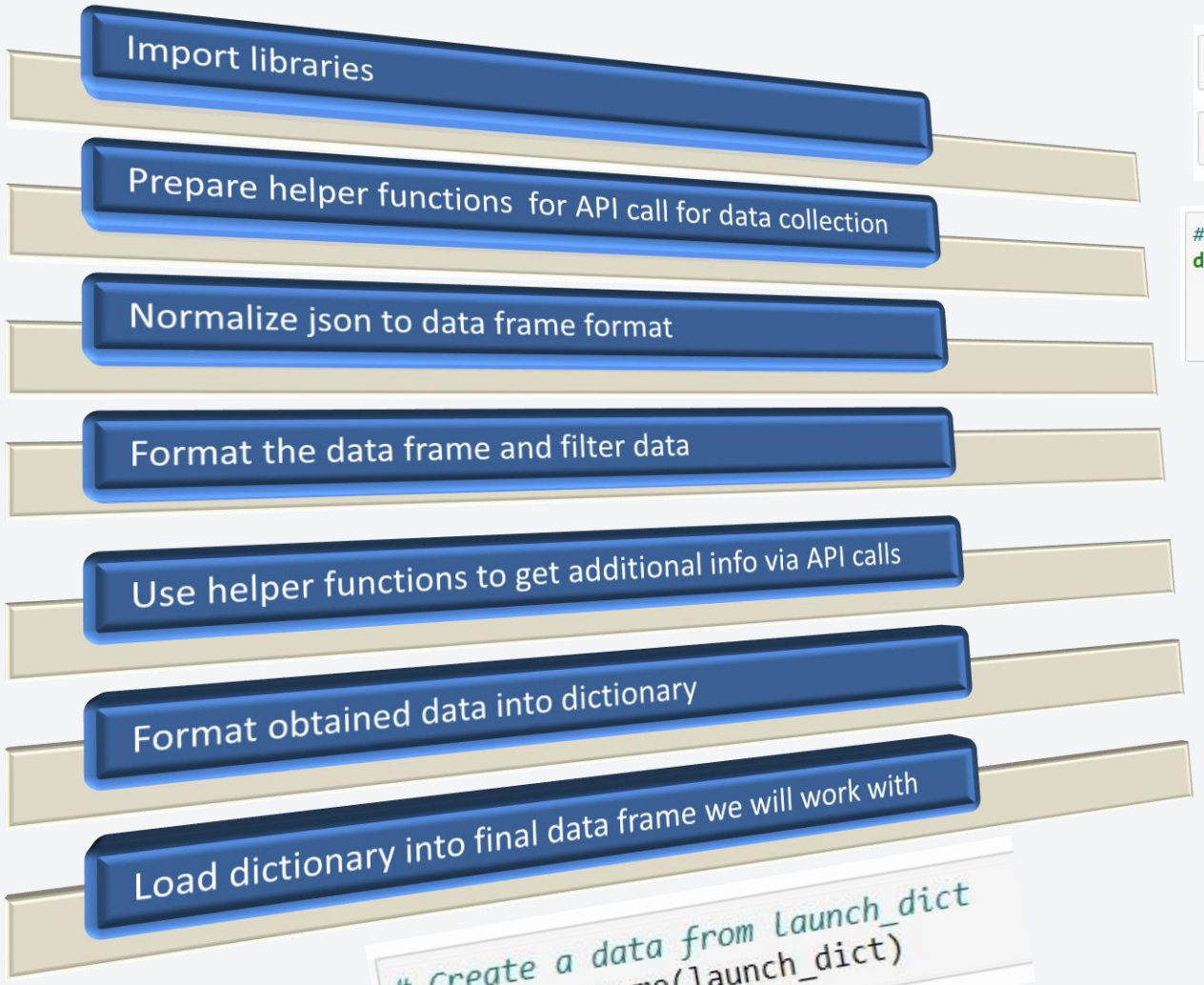
Methodology

Methodology

Executive Summary

- Data collection methodology:
  Data was collected by:
  - Web scraping of historical launch records from Wikipedia
  - REST API call of json format data file
- Data wrangling
  - The data was filter to include only relevant information about Falcon 9 launches.
  - The data frame was checked, missing data was replaced by average value where appropriate
  - Categorical variables were transformed by „One Hot Encoding“
- Data analysis (EDA) through visualization and SQL
  - Prepared scatter and bar plots, and interactive maps and dashboard (via Folium and Plotly Dash)
  - Executed several SQL queries to extract answers for given questions from data files in database
- Performed predictive analysis using classification models
  - Prepared LOGREG, SVM, KNN, DECISION TREE
  - Optimized by GridSearch method

Data Collection




```
spacex_url="https://api.spacexdata.com/v4/launches/past"
```

```
response = requests.get(spacex_url)
```

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

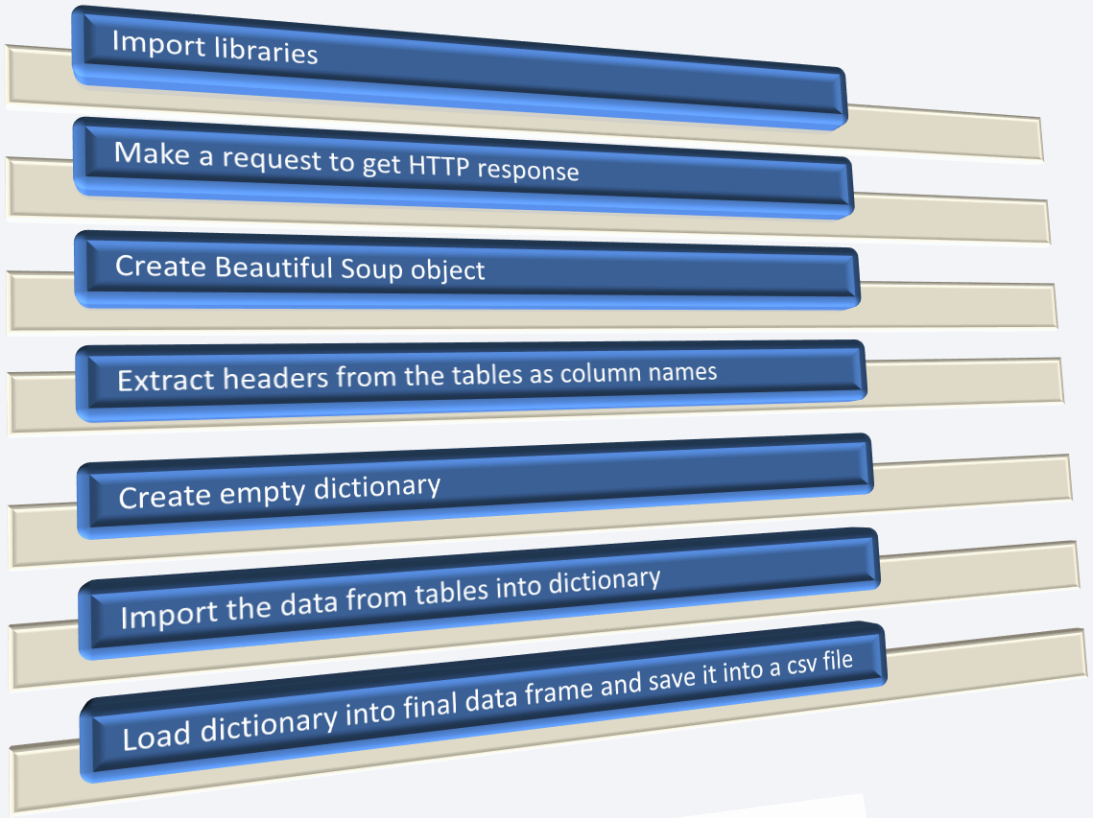
```
# Use json_normalize meethod to convert the json result into a dataframe
datfram = response.json()
data = pd.json_normalize(datfram)
```

```
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}
```

```
# Create a data from launch_dict
df=pd.DataFrame(launch_dict)
```


GitHub Link

Data Collection - Scraping




```
df=pd.DataFrame(launch_dict)

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

```
# use requests.get() method with the provided static_url
# assign the response to a object
response=requests.get(url=static_url).text
```

Create a `BeautifulSoup` object from the HTML `response`

```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(response,"html.parser")
```

```
column_names = []

# Apply find_all() function with `th` element on first_launch_table
# Iterate each th element and apply the provided extract_column_from_header() to get a column name
# Append the Non-empty column name (`if name is not None and len(name) > 0`) into a list called column_names

columns= html_tables[2].find_all('th')

for i in columns:
    name=extract_column_from_header(i)
    if(name!=None and len(name)>0):
        column_names.append(name)
```

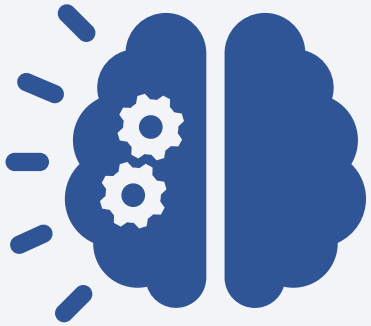
```
launch_dict= dict.fromkeys(column_names)

# Remove an irrelvant column
del launch_dict['Date and time ( )']

# Let's initial the launch_dict with each value to be an empty list
launch_dict['Flight No.'] = []
launch_dict['Launch site'] = []
launch_dict['Payload'] = []
launch_dict['Payload mass'] = []
launch_dict['Orbit'] = []
launch_dict['Customer'] = []
launch_dict['Launch outcome'] = []
# Added some new columns
launch_dict['Version Booster']=[]
launch_dict['Booster landing']=[]
launch_dict['Date']=[]
launch_dict['Time']=[]
```

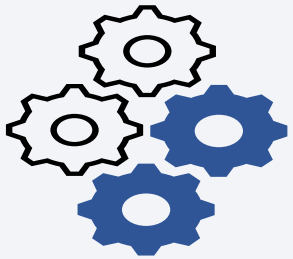
```
extracted_row = 0
#Extract each table
for table_number,table in enumerate(soup.find_all('table',"wikitable plainrowheaders collapsible")):
    # get table row
    for rows in table.find_all("tr"):
        #check to see if first table heading is as number corresponding to launch a number
        if rows.th:
            if rows.th.string:
                flight_number=rows.th.string.strip()
                flag=flight_number.isdigit()
        else:
            flag=False
        #get table element
        row=rows.find_all('td')
        #if it is number save cells in a dictonary
        if flag:
            extracted_row += 1
            # Flight Number value
            # TODO: Append the flight_number into launch_dict with key `Flight No.`
            #print(flight_number)
            datatimelist=date_time(row[0])
```


GitHub Link

Data Wrangling

- Filter the data frame to include only data relevant for analysis
- Check in data frame for missing values with isnull method
- Consider appropriate manipulation (keep empty, transform, drop)
- We apply replacement by average value and keep empty approach





GitHub Link

EDA with Data Visualization

Several graphical analysis considering relationship to landing success were prepared

Scatter plots:

- Pay load mass vs. Flight number
- Flight number vs. Launch site
- Pay load mass vs. Launch site
- Flight number vs. Orbit
- Pay load mass vs. Orbit

Bar charts:

Landing success rate per orbit

Line charts:

Success rate yearly trend

GitHub Link

EDA with SQL

Following list of task/question was answered via queries to database

- Display the names of the unique launch sites in the space mission.
- Display 5 records where launch sites begin with the string 'CCA‘.
- Display the total payload mass carried by boosters launched by NASA (CRS).
- Display average payload mass carried by booster version F9 v1.1.
- List the date when the first successful landing outcome in ground pad was achieved.
- List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000.
- List the total number of successful and failure mission outcomes.
- List the names of the booster_versions which have carried the maximum payload mass. Use a subquery.
- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015.
- 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.

GitHub Link

Build an Interactive Map with Folium

- Pop up markers with cycles were prepared to indicate Launch site's location.
- Marker clusters were created to indicate successful/failed landings in each site.
- Lines were drawn to indicate distance between Sites and some important locations like near cities, rails and coast lines...

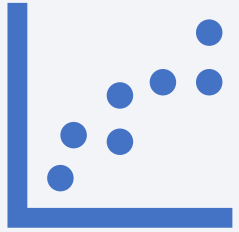
GitHub Link

Build a Dashboard with Plotly Dash

In Plotly Dash were prepared interactive dashboard consisting of:

Pie charts representing:

- Success landing count per Sites (in percentage).
- Success rate per chosen sites.



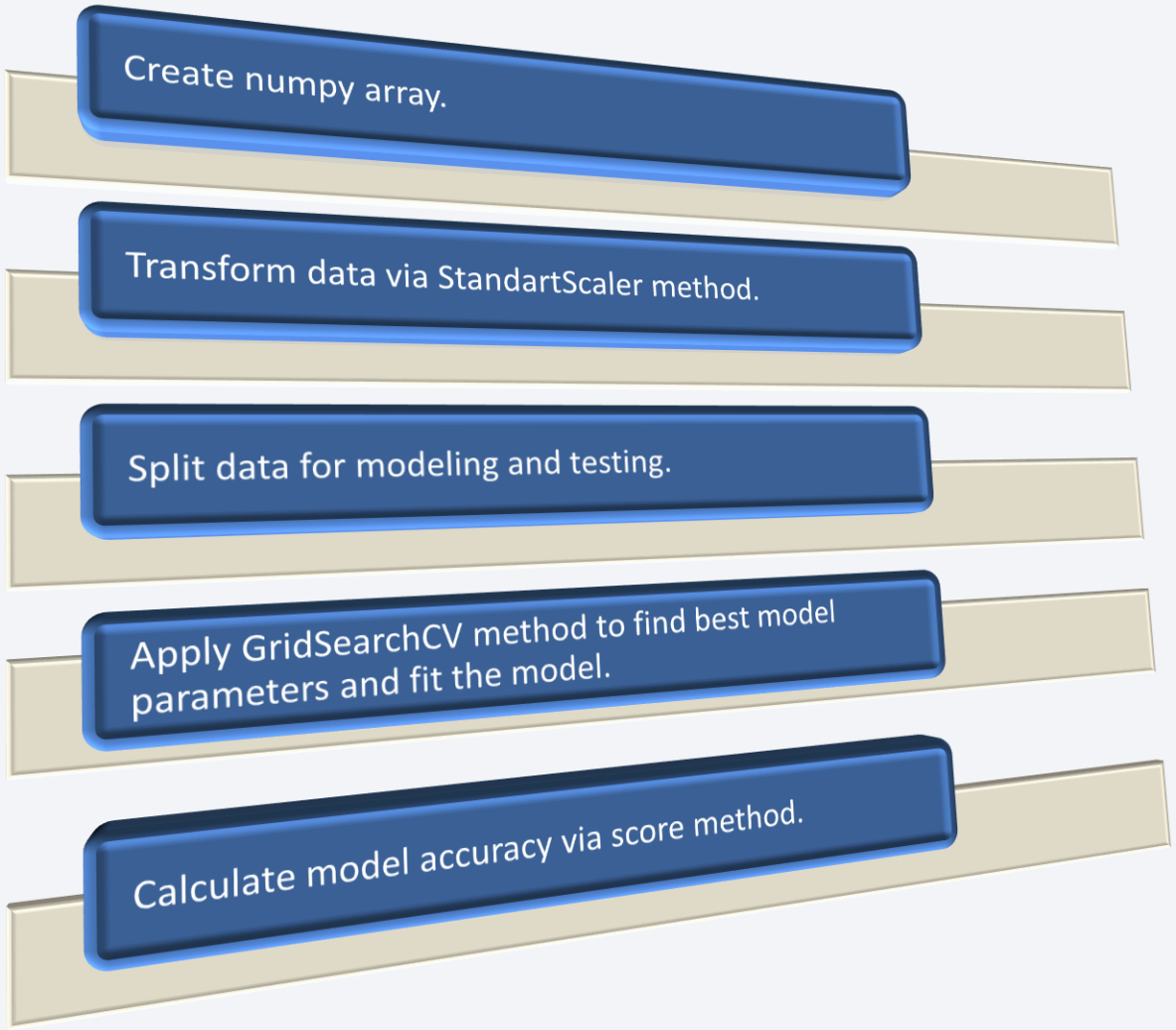
Scatter plot representing:

Total launch success on pay load mass for all sites or per selected site.

Including slider to specify range of pay load mass.

GitHub Link

Predictive Analysis (Classification)




```
Y = data.Class.to_numpy()
```

```
# students get this
transform = preprocessing.StandardScaler().fit(X).transform(X.astype(float))
X=transform
```

```
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)
```

```
parameters ={'C':[0.01,0.1,1],
             'penalty':['l2'],
             'solver':['lbfgs']}
LR_object=LogisticRegression()
logreg_cv=GridSearchCV(LR_object,parameters,cv=10)
out=logreg_cv.fit(X_train,Y_train)
```

```
logreg_accuracy_score=logreg_cv.score(X_test,Y_test)
```


This approach was applied during Logistics regression, SVM, Decision tree and KNN models preparation.

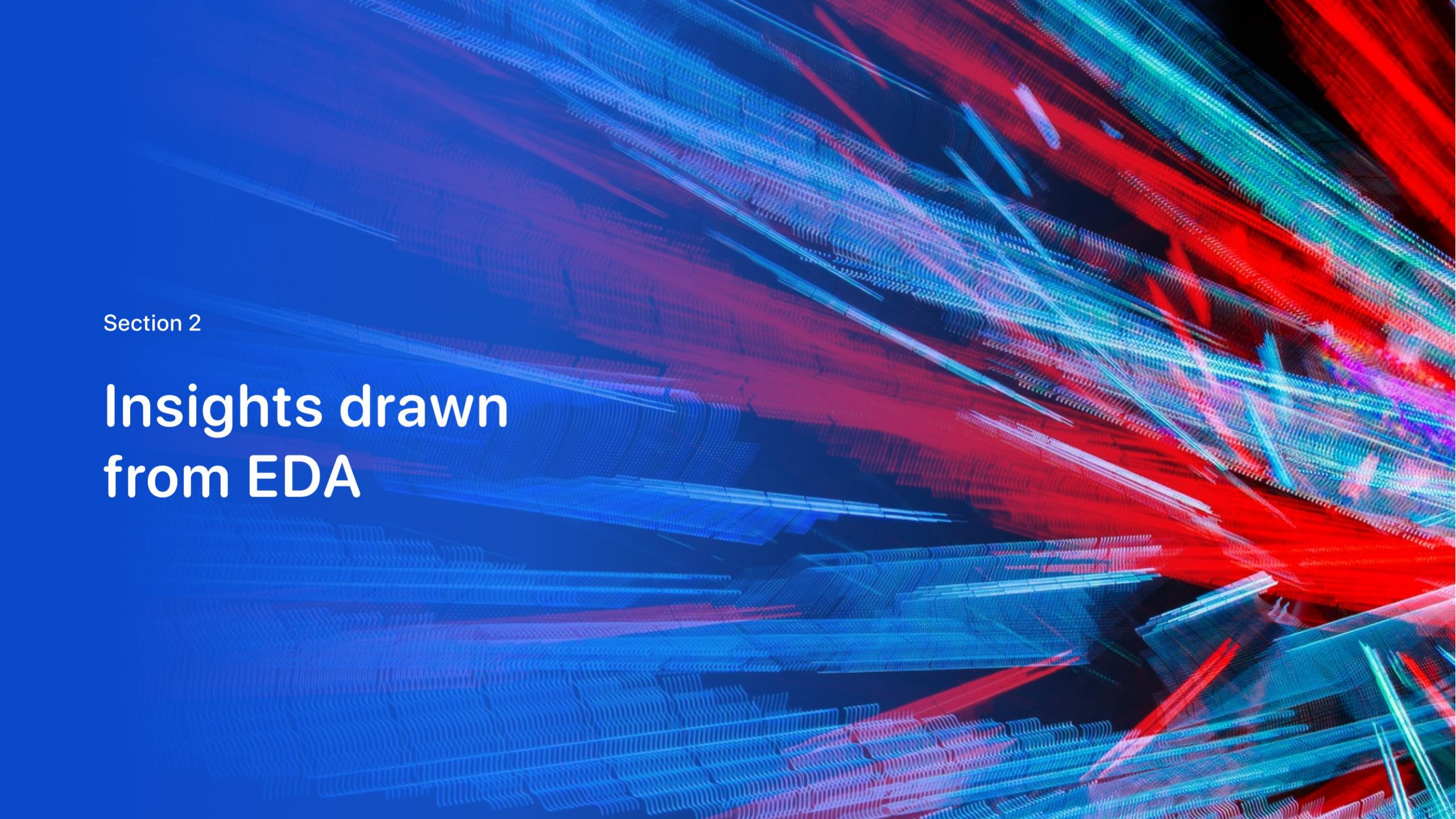
GitHub Link

Results

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

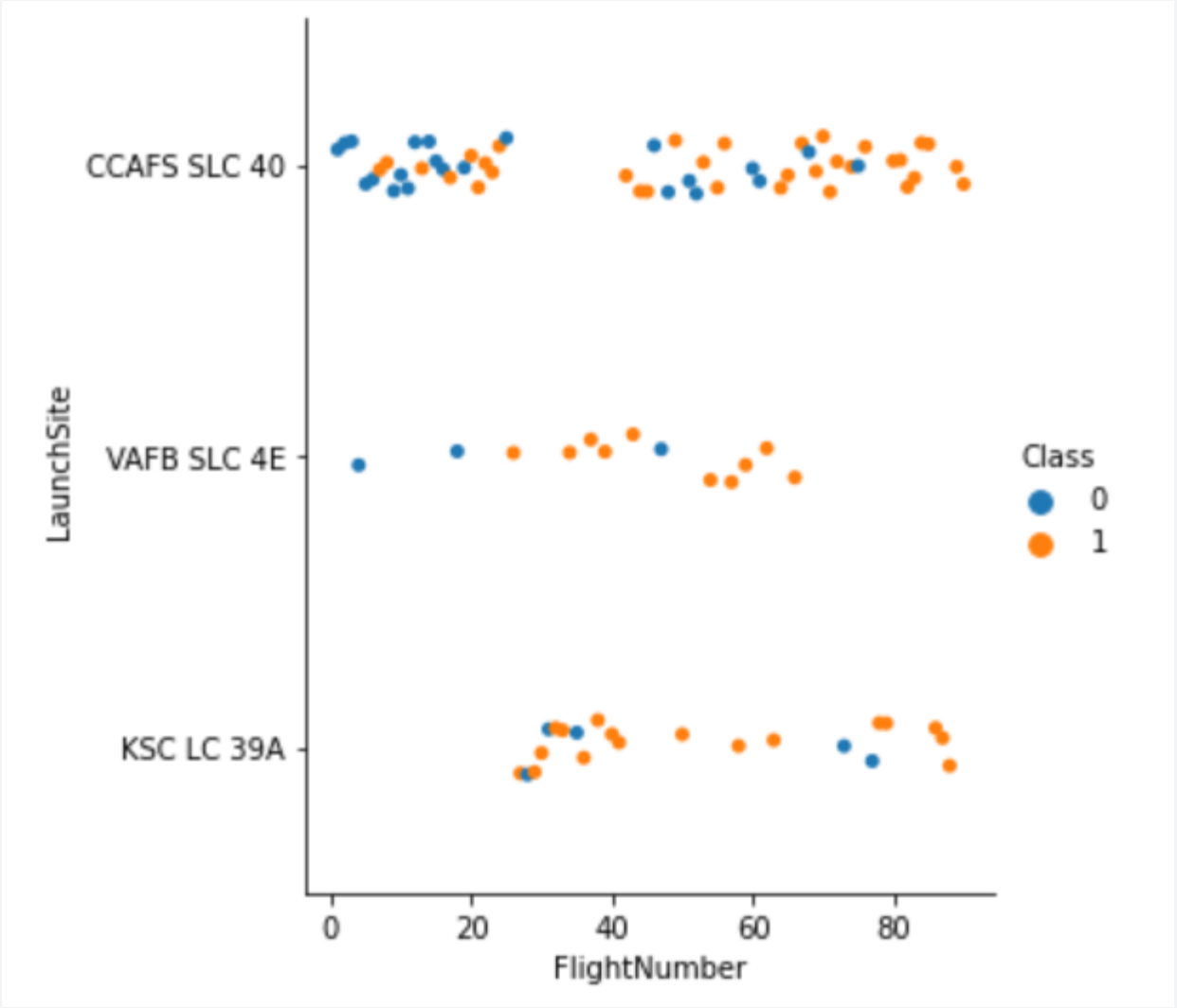
Section 2

Insights drawn from EDA



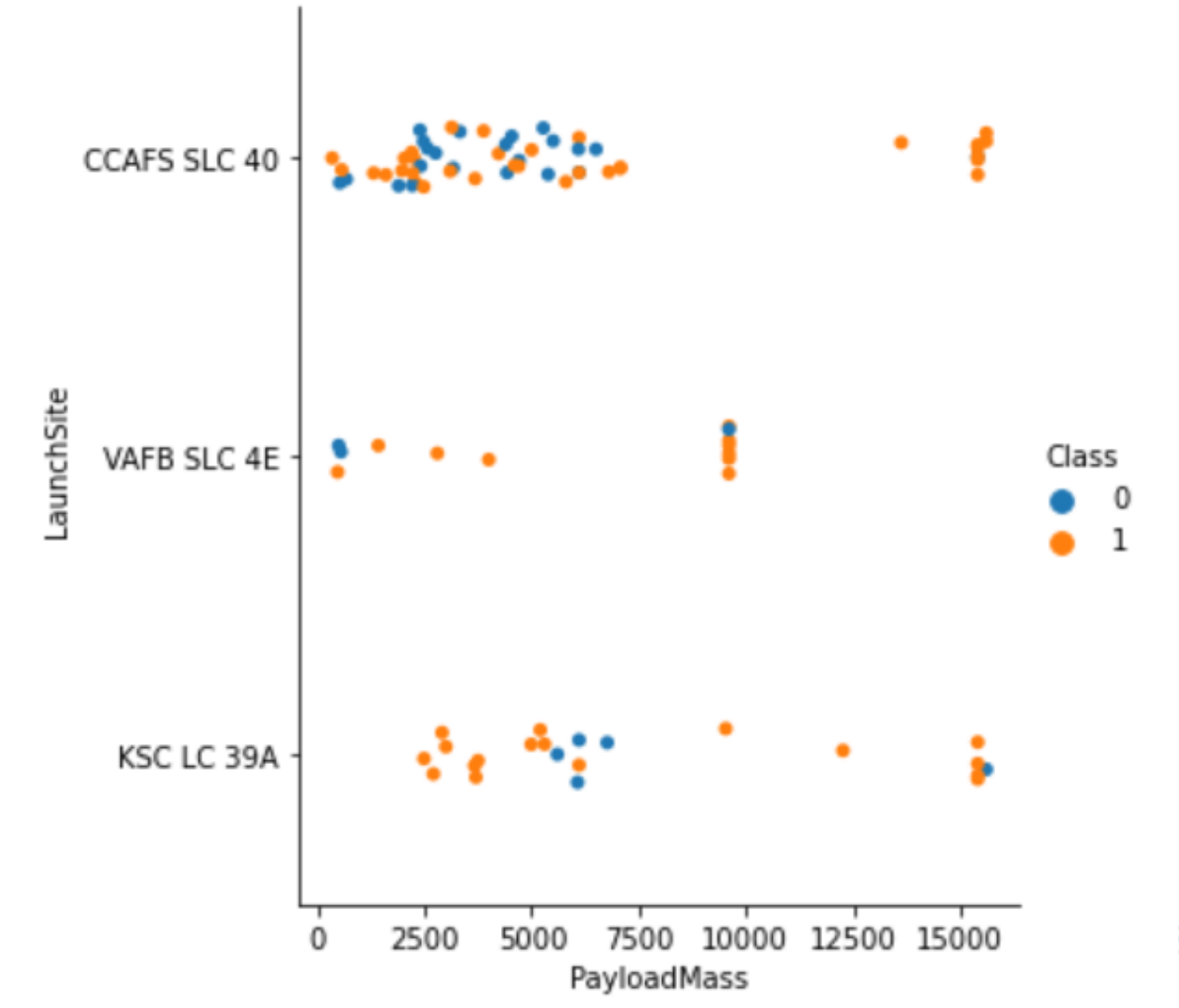
Flight Number vs. Launch Site

- Show a scatter plot of Flight Number vs. Launch Site
- Show the screenshot of the scatter plot with explanations



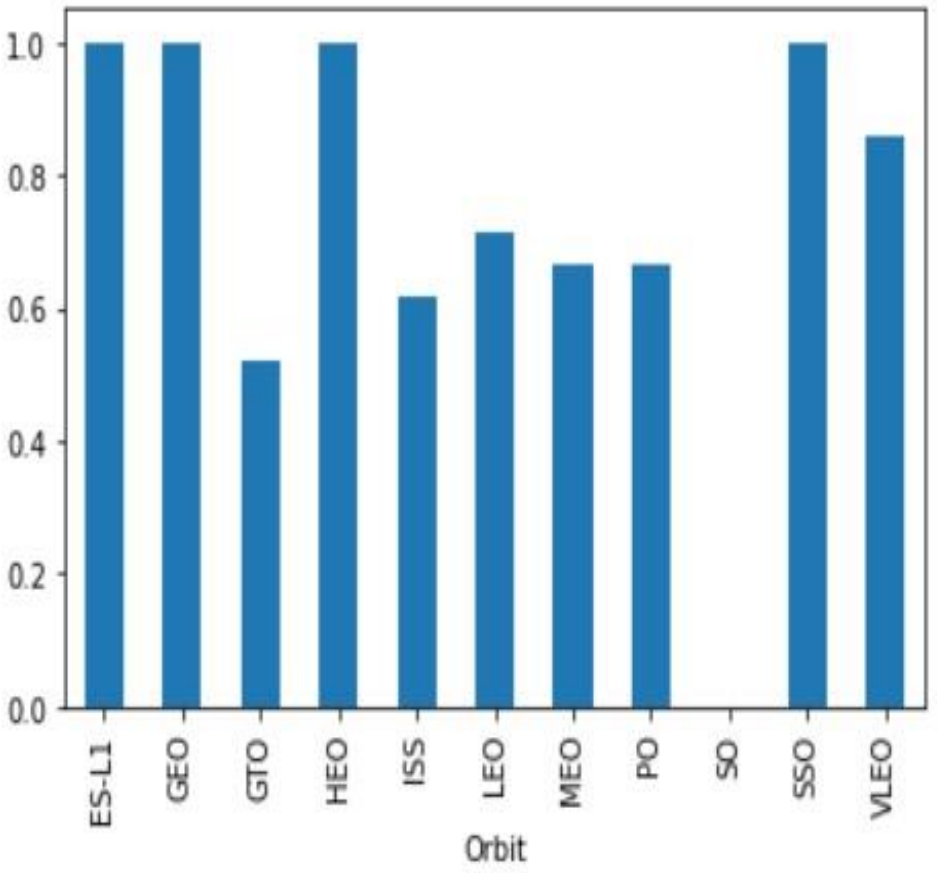
Payload vs. Launch Site

- Show a scatter plot of Payload vs. Launch Site
- Show the screenshot of the scatter plot with explanations



Success Rate vs. Orbit Type

- **GEO, SSO, HEO and ES-L1 orbits with 100% success rate are good OPTIONS**
- VLEO with more than 80% success rate could be still a good option.
- **GTO and SO Orbits are the risky** ones.



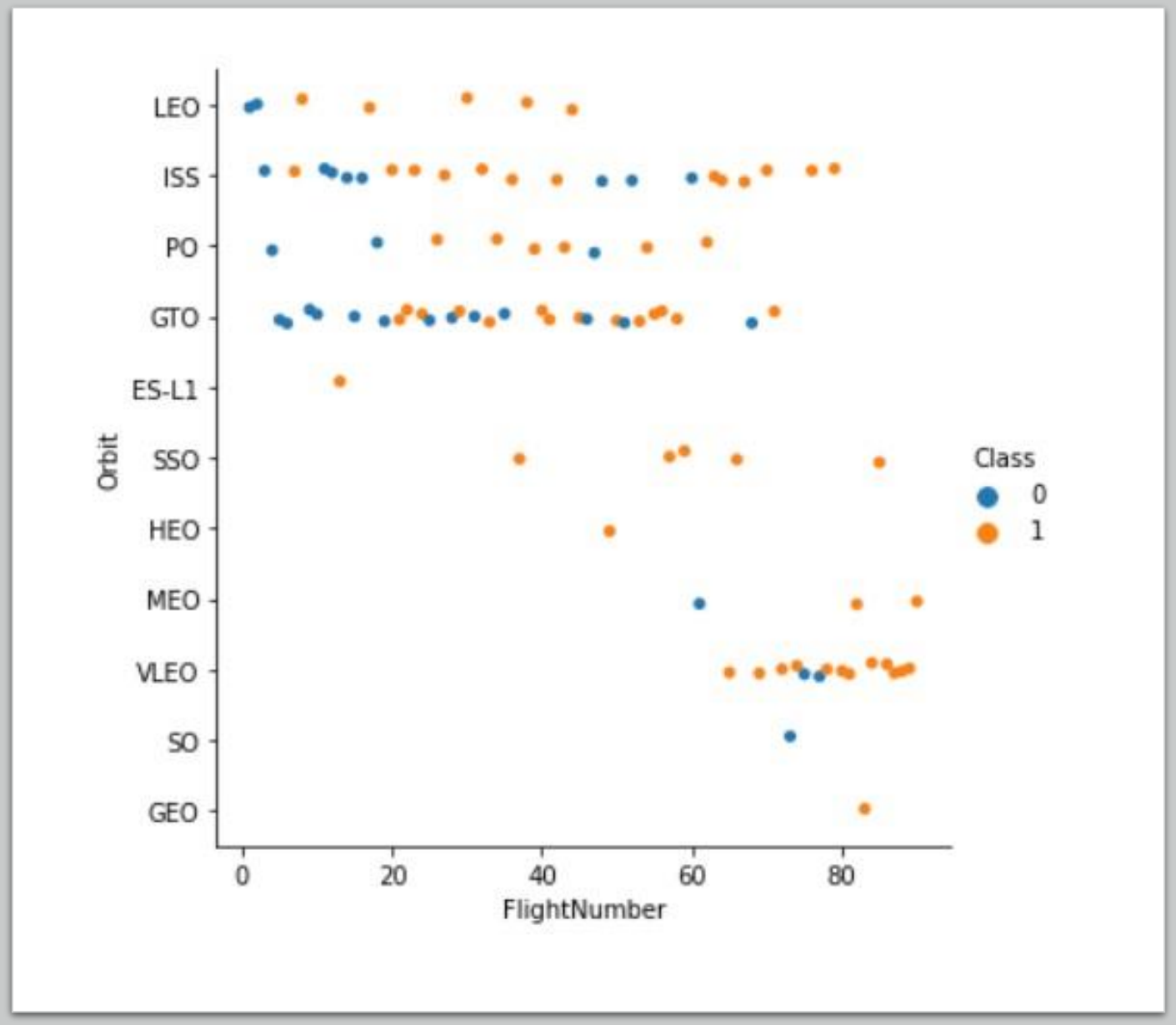
Orbit	mean	count
SSO	1.000000	5
ES-L1	1.000000	1
GEO	1.000000	1
HEO	1.000000	1
VLEO	0.857143	14
LEO	0.714286	7
PO	0.666667	9
MEO	0.666667	3
ISS	0.619048	21
GTO	0.518519	27
SO	0.000000	1

It is important to take into consideration also number of flights to certain orbit.

For example, ES-L1, HEO and GEO have 100% success rate, but there was only 1 flight for each.

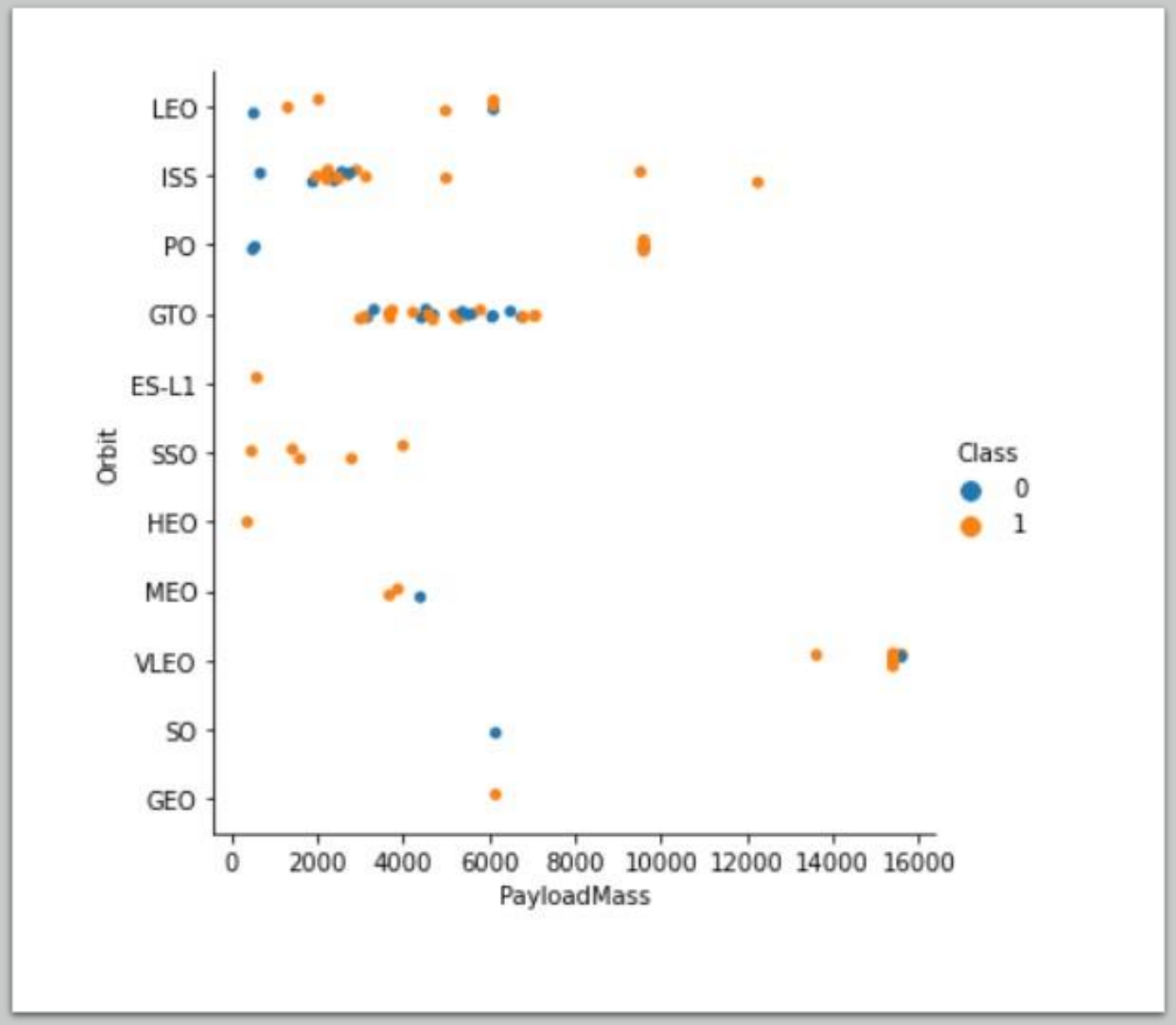
Flight Number vs. Orbit Type

- Number per Orbit significantly vary.
- Success rate seems to improve with increasing number of flights.



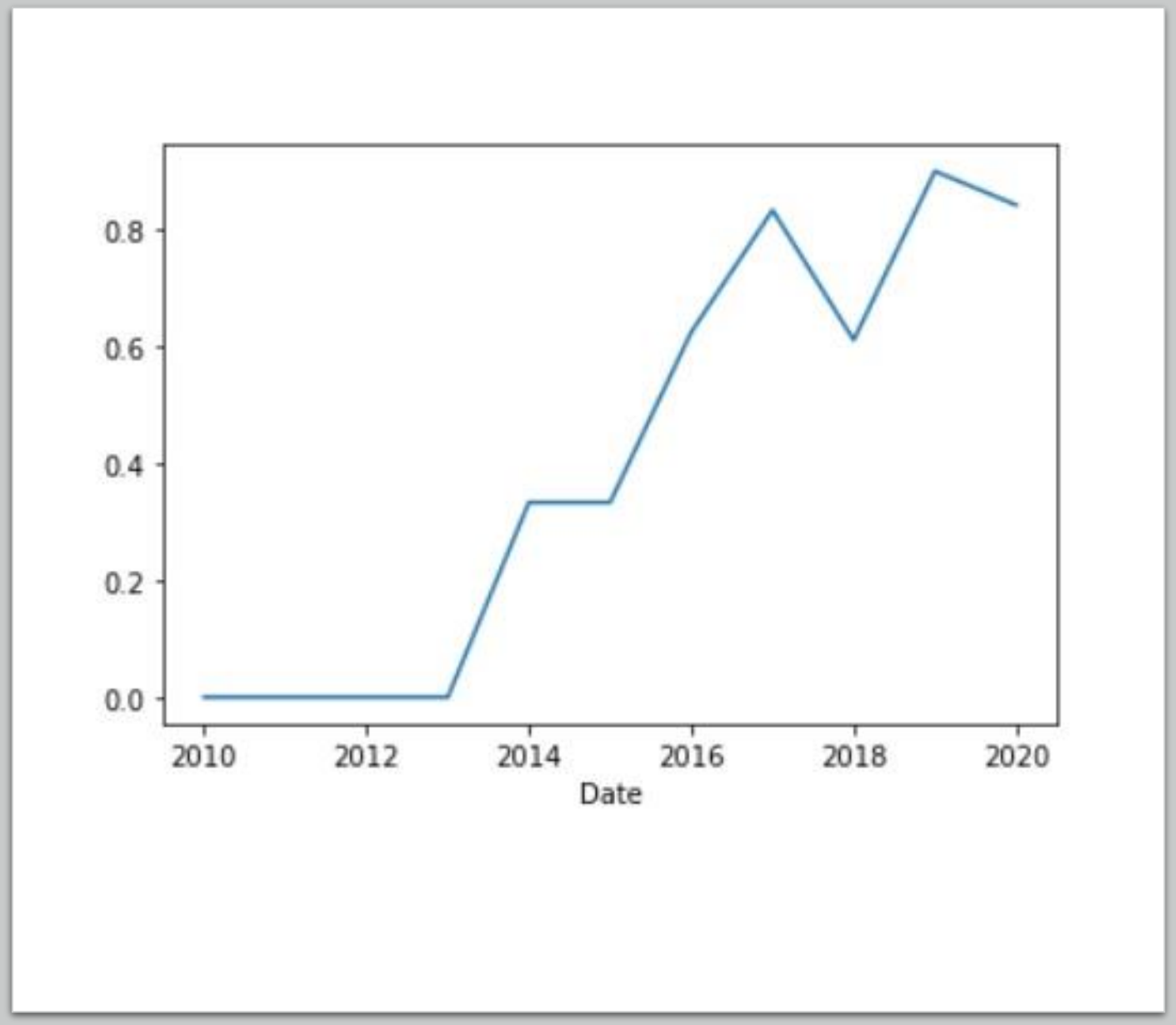
Payload vs. Orbit Type

- Most of the flights were executed with Payload under 8 000 kg.
- There is not clear payload - orbit relation to success rate.



Launch Success Yearly Trend

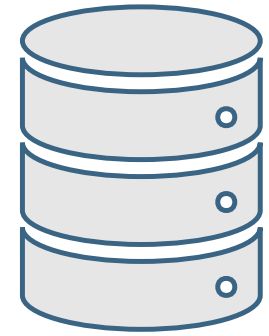
From 2013 the launch success rate gradually increase up to over 80% in 2020.



SQL

All Launch Site Names

Following query was used to extract Launch Site Names from the database:



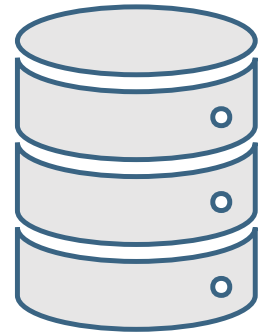
```
%sql select distinct(launch_site) from SPACEXTBL;
```

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

SQL

Launch Site Names Begin with 'CCA'

Following query was used to extract required output from the database:



```
%sql select * from SPACEXTBL where launch_site like 'CCA%' LIMIT 5;
```

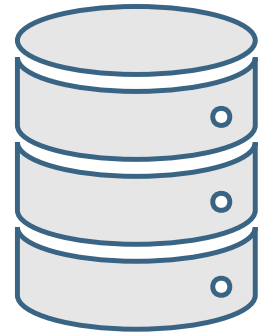
DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-05-22	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	00:35:00	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

SQL

Total Payload Mass

Average Payload Mass by F9 v1.1

Following queries were used to extract answers from the database:



```
%sql select sum(payload_mass__kg_) from SPACEXTBL where customer='NASA (CRS)';
```

1
45596

```
%sql select avg(payload_mass__kg_) from SPACEXTBL where booster_version like'F9 v1.1%';
```

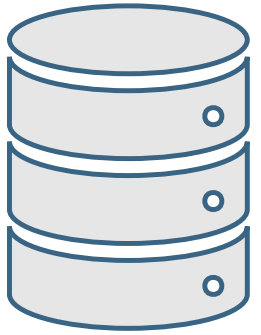
1
2534

SQL

First Successful Ground Landing Date

Successful Drone Ship Landing with Payload between 4000 and 6000

Following queries were used to extract answers from the database:



```
%sql select min(DATE) from SPACEXTBL where landing__outcome ='Success';
```

1
2018-07-22

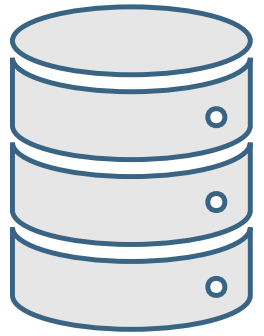
```
%%sql
select booster_version from SPACEXTBL
where landing__outcome ='Success'
and payload_mass__kg_ between 4000 and 6000;
```

booster_version
F9 B5 B1046.2
F9 B5 B1047.2
F9 B5 B1046.3
F9 B5 B1048.3
F9 B5 B1051.2
F9 B5B1060.1
F9 B5 B1058.2
F9 B5B1062.1

SQL

Total Number of Successful and Failure Mission Outcomes

Following query was used to extract answer from the database:



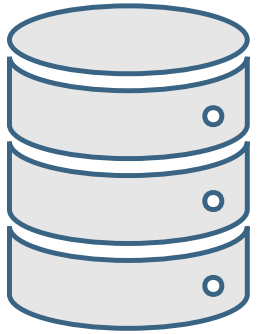
```
%%sql
select mission_outcome, count(mission_outcome) from SPACEXTBL
group by mission_outcome;
```

mission_outcome	2
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

SQL

Boosters Carried Maximum Payload

Following query was used to extract answer from the database:



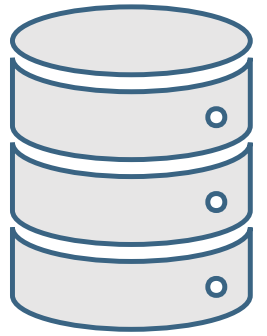
```
%%sql
select DISTINCT(booster_version) from SPACEXTBL
where (select max(payload_mass__kg_) from SPACEXTBL);
```

booster_version
F9 B4 B1039.2
F9 B4 B1040.2
F9 B4 B1041.2
F9 B4 B1043.2
F9 B4 B1039.1
F9 B4 B1040.1
F9 B4 B1041.1
F9 B4 B1042.1
F9 B4 B1043.1
F9 B4 B1044

SQL

2015 Launch Records

Following query was used to extract 2015 launch records from the database:



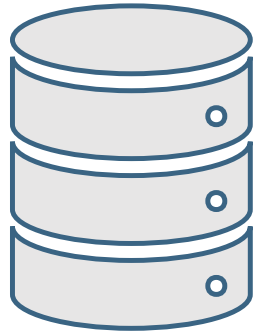
```
%%sql
SELECT DATE,landing__outcome,booster_version,orbit from SPACEXTBL
WHERE landing__outcome = 'Failure (drone ship)'
AND DATE LIKE '2015%';
```

DATE	landing__outcome	booster_version	orbit
2015-01-10	Failure (drone ship)	F9 v1.1 B1012	LEO (ISS)
2015-04-14	Failure (drone ship)	F9 v1.1 B1015	LEO (ISS)

SQL

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

Following query was used to extract Outcomes Between 2010-06-04 and 2017-03-20 from the database:



```
%%sql
select landing__outcome,count(landing__outcome) from SPACEXTBL
where day(DATE) between '2010-06-04' and '2017-03-20'
group by landing__outcome
order by count(landing__outcome)
```

Section 4

Launch Sites Proximities Analysis

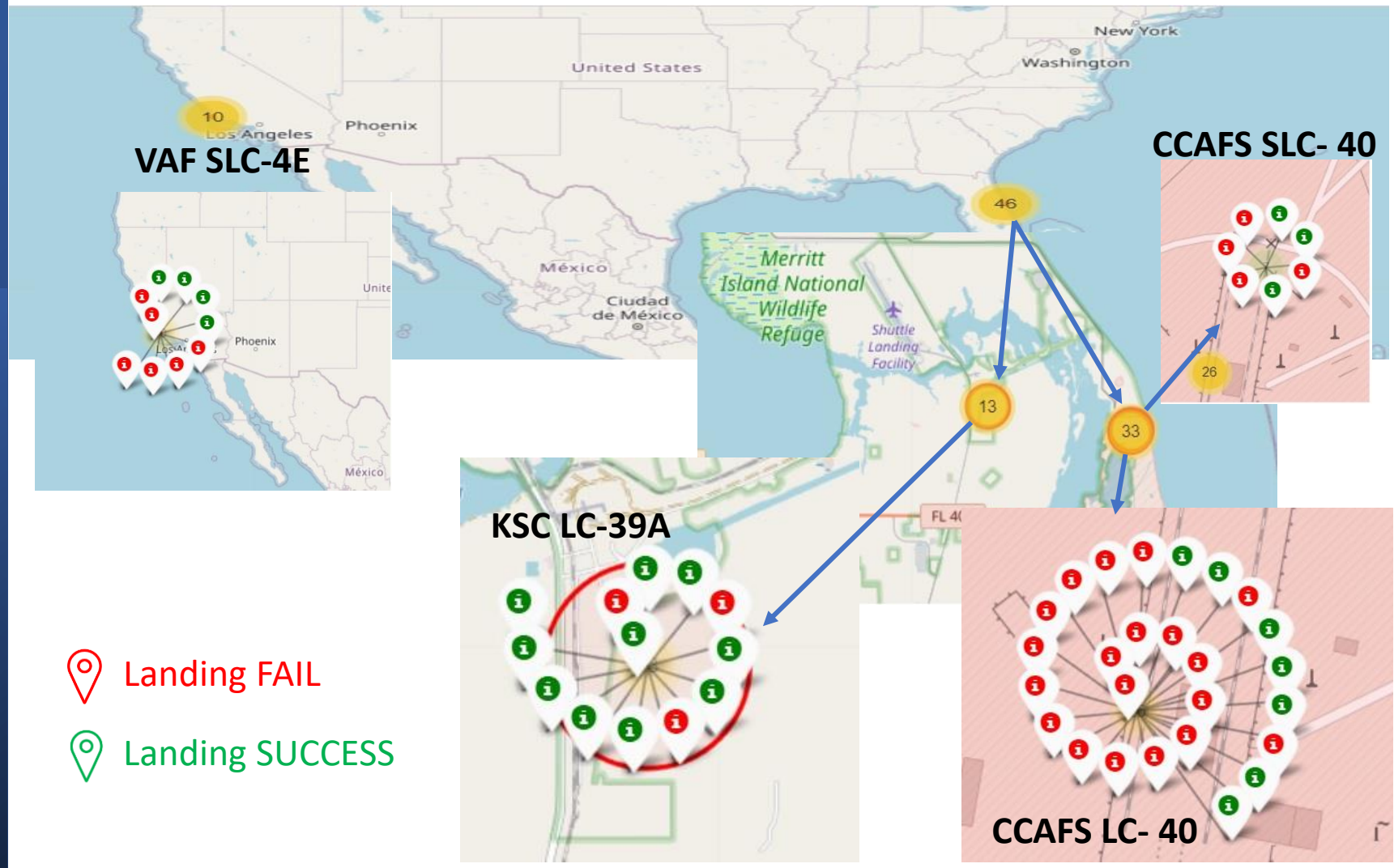
Launch Sites Locations and Number of Flights

Launch Sites are located at coastline of both Pacific and Atlantic ocean with similar latitudes.

Launch Sites located at coast of Atlantic ocean however executed four times more flights than the ones at the Pacific coast.

Launch Sites Mission Success Rate Visualization

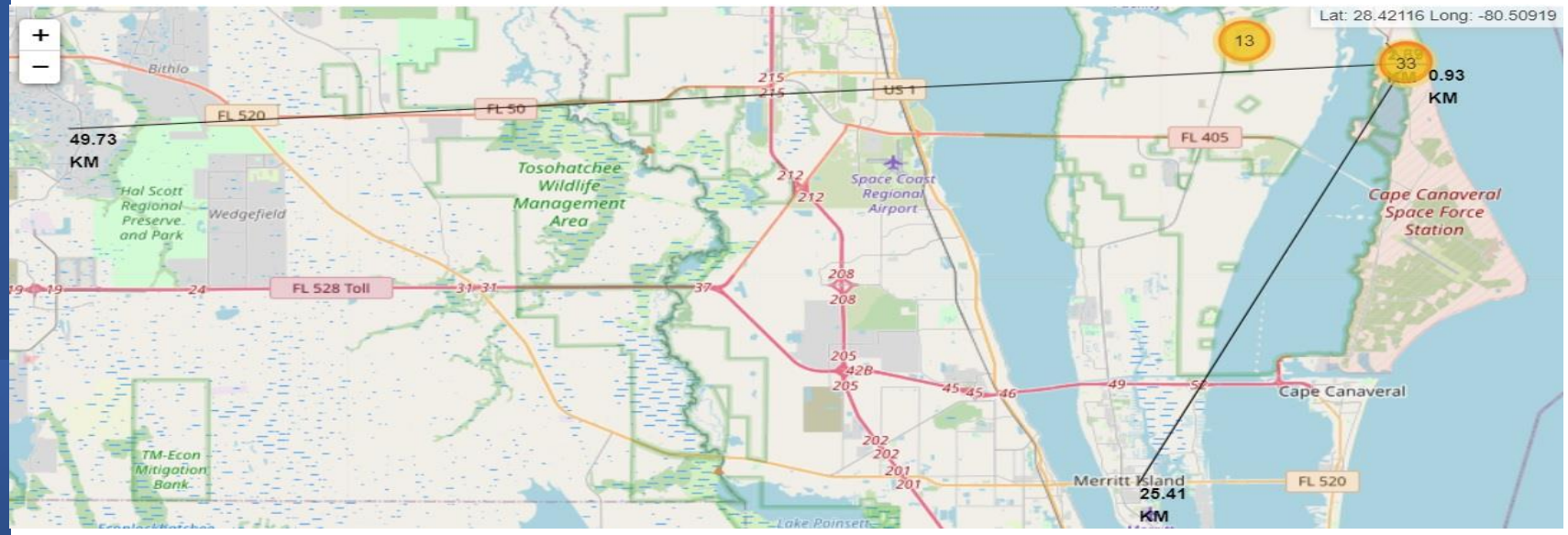
From visualization is clear that site **KSC LC -39A has the highest success rate**.



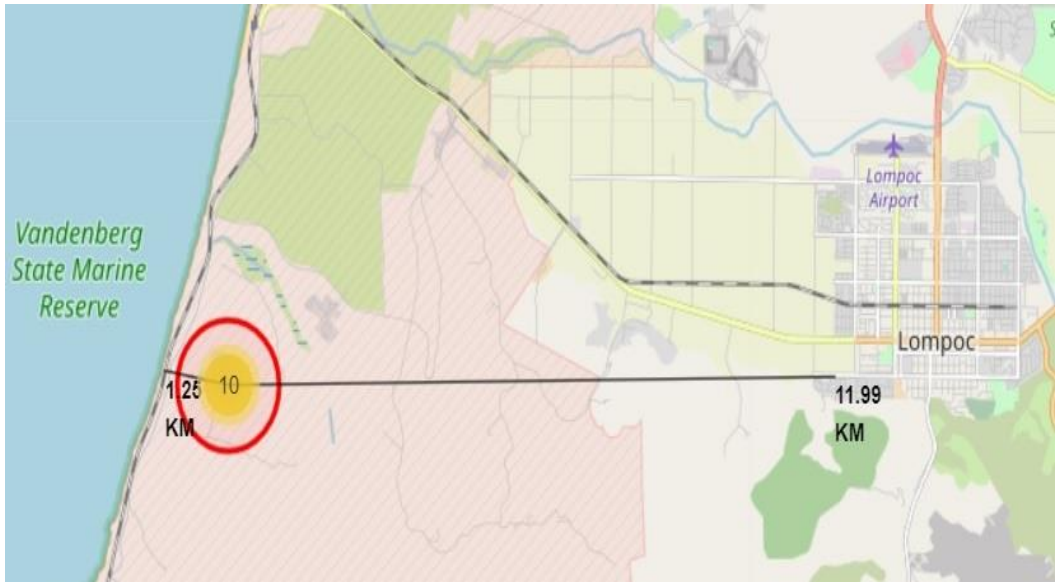
On the other hand, despite of high number of flights has the site **CCAFS LS - 40 the lowest number of successful landing**.

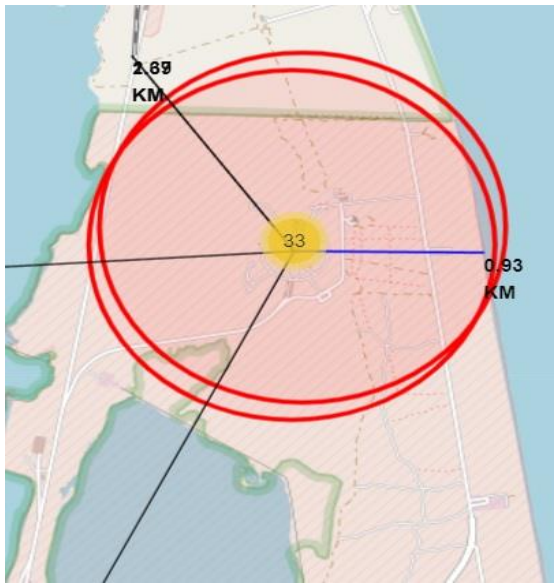
Launch Sites Similarities

Launch site locations share several similarities like almost identical distance to coastline and railways.



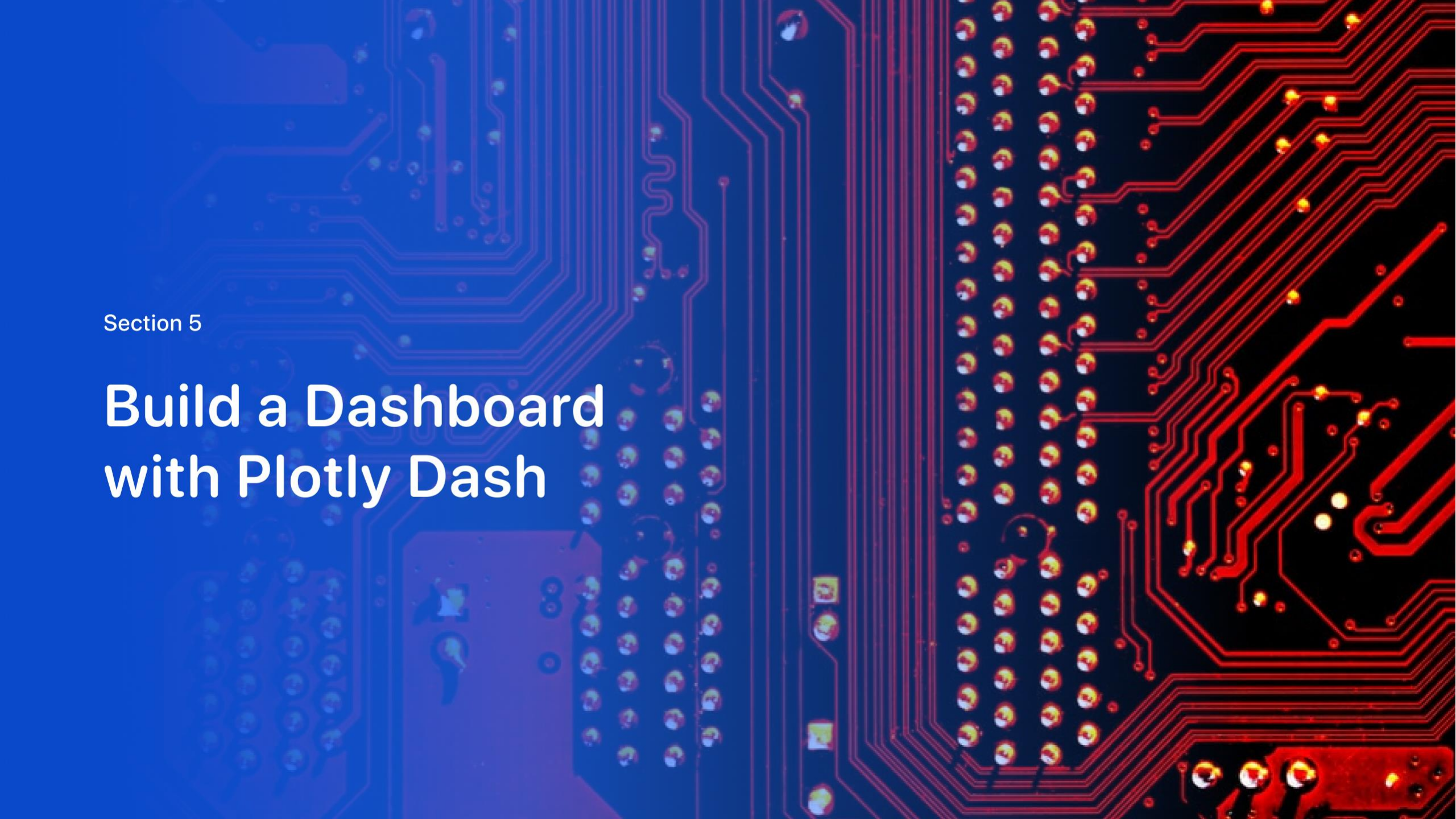
On the other hand, distance to the nearest cities use to be over 10 km for each site.





Section 5

Build a Dashboard with Plotly Dash

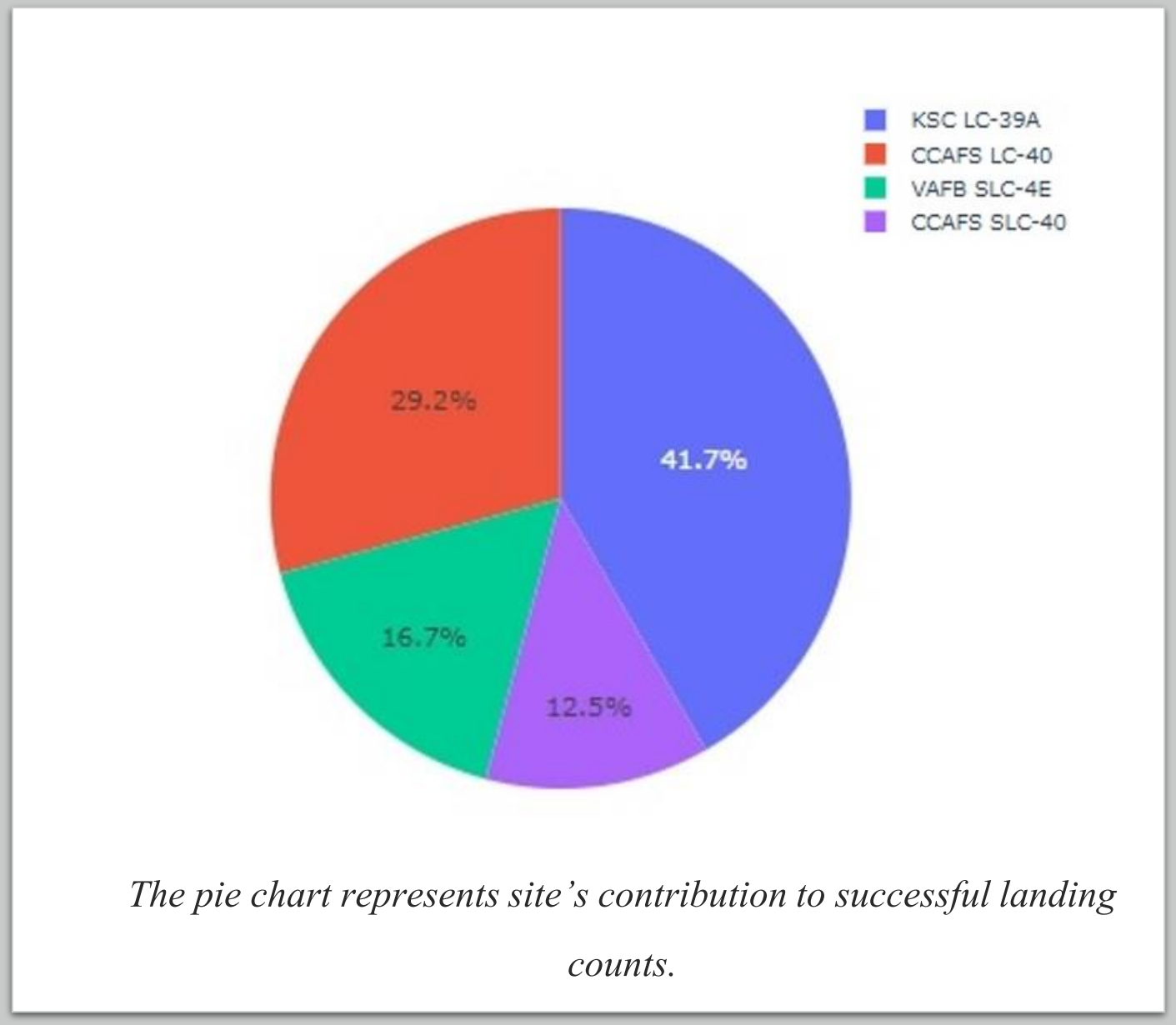


Success landing count per Sites
(in percentage)

- Almoust half of all successful landings are due to KSC LC3A site
(In other words, 41,7% of all successful landing were at KSC LC3A).

- The remainig „half“ is splited between the other three sites.

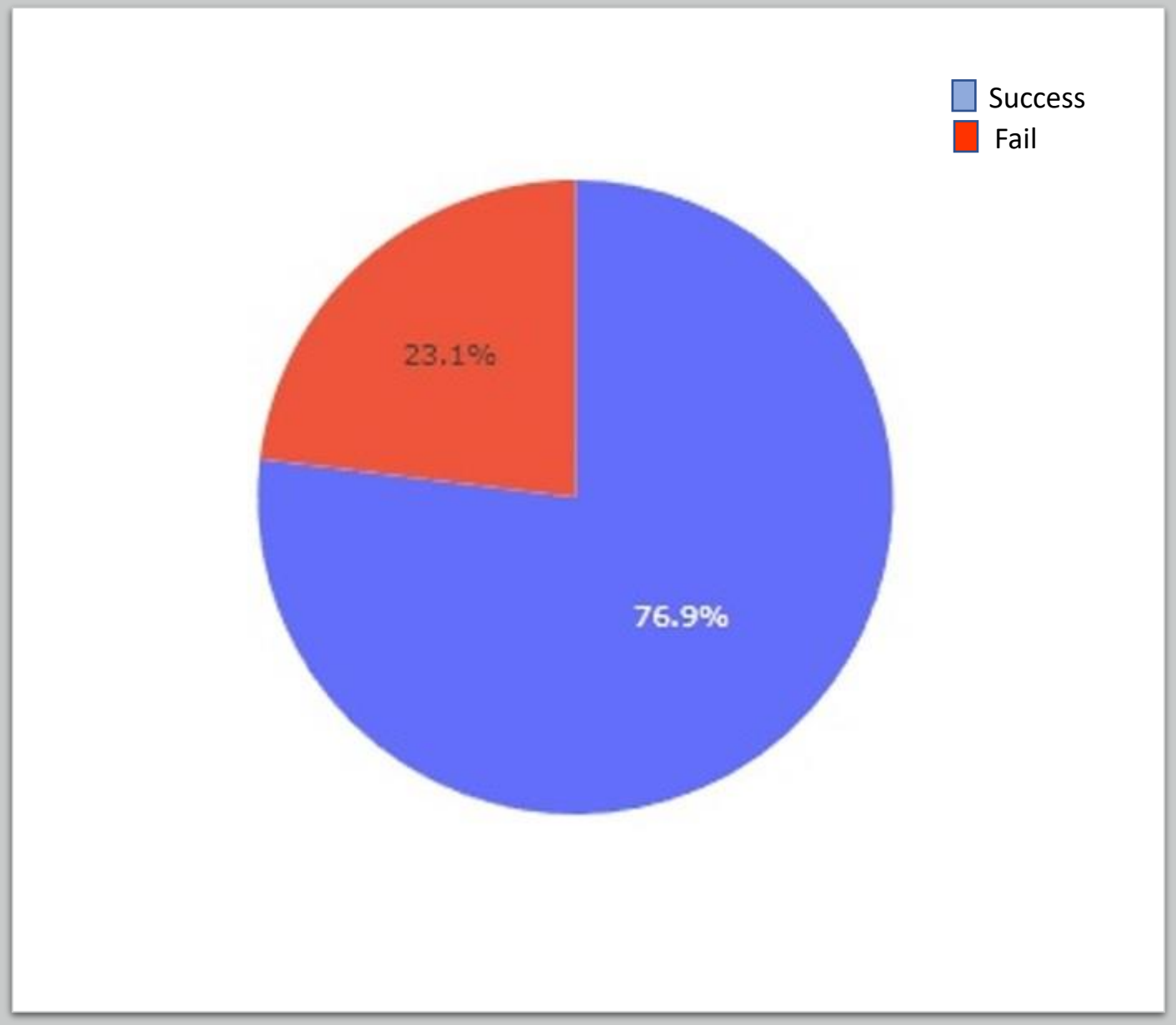
To understand it better have a look at success rates per site.



The pie chart represents site’s contribution to successful landing counts.

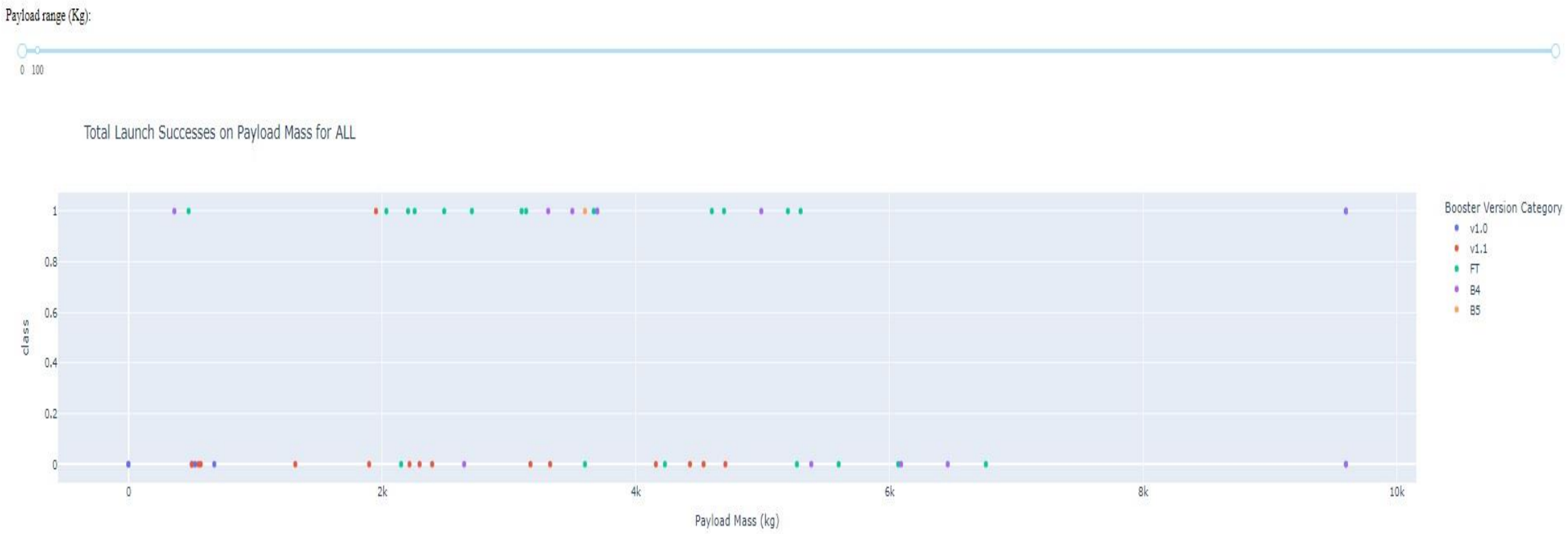
KSC LC3A has the highest landing success rate from all sites.

- The success rate of KSC LC3A site is 76,9%.
- From 13 landings 10 were successful.



Overview of launch success pre booster version

- **Booster version FT contributes the most to launch success in Payload range up to 6 000 kg.**
- **However, the only successful attempt with payload over 8 000 kg were done with booster B4.**

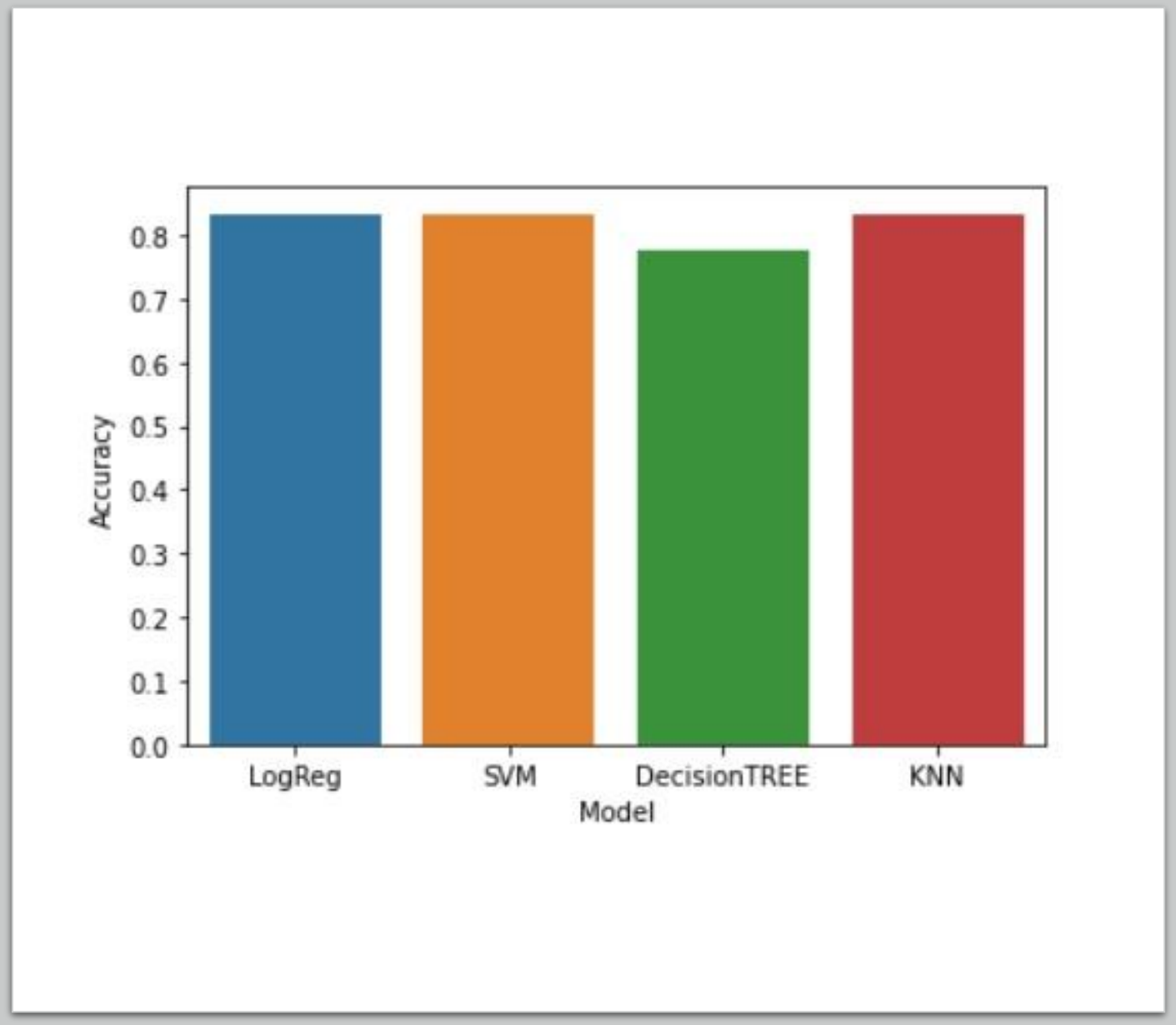


Section 6

Predictive Analysis (Classification)

Classification Accuracy

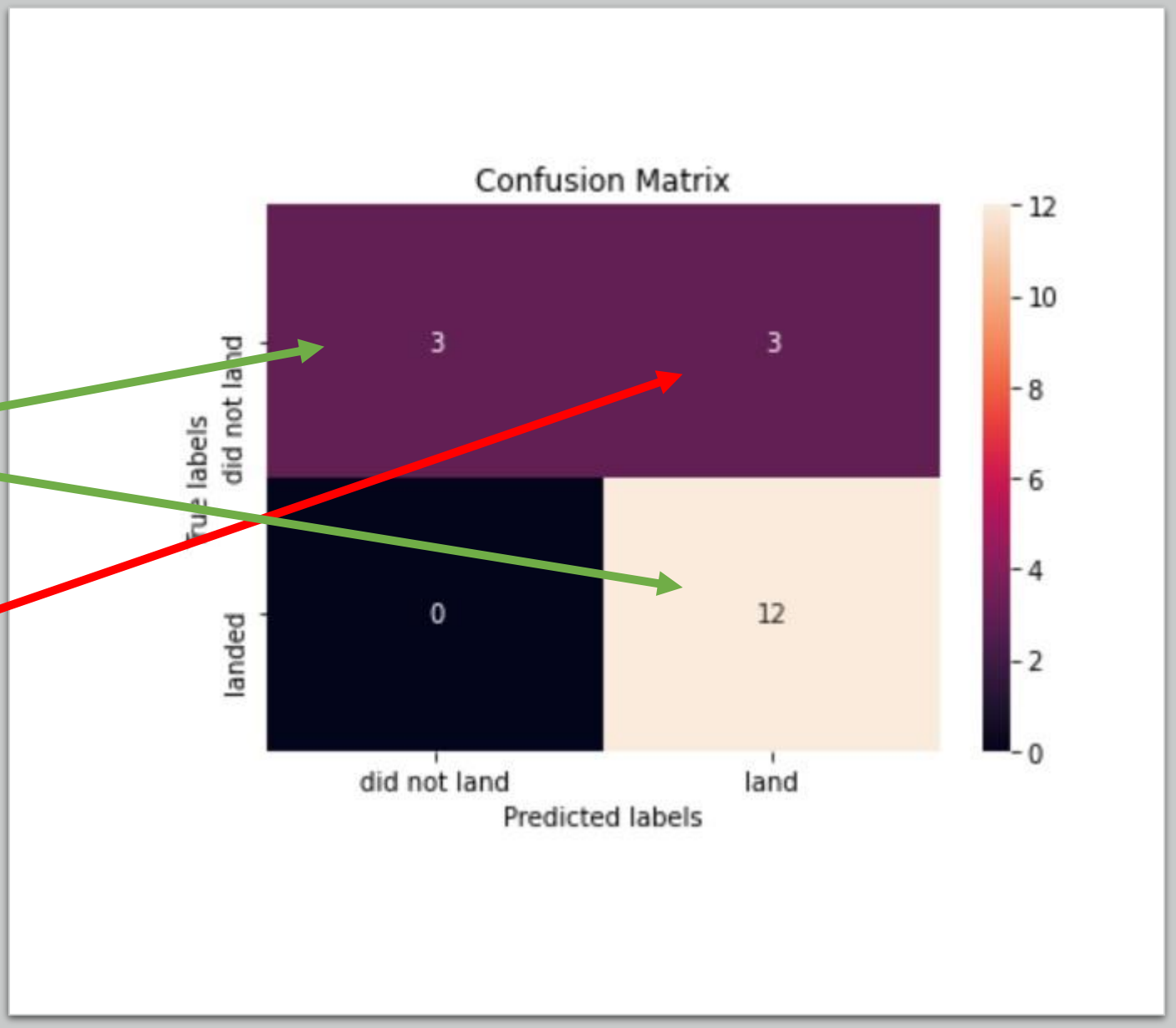
Logistic regression, **SVM** and **KNN** models have the same model **accuracy** of **83%.**



Confusion Matrix

Confusion matrix shows that prepared classification models* correctly predicts 15 of 18 launches success or failure.

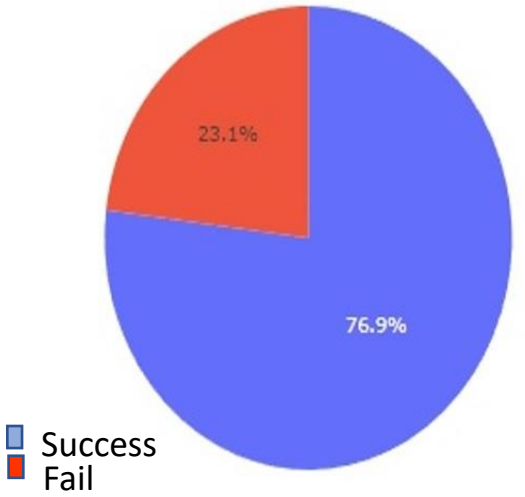
Only in 3 cases model prediction was not correct when 3 failures predicts as successful landing.



*Logistic regression, SVM and KNN

Summary: KSC LC3A has the highest landing success rate from all sites.

SUCCESS RATE:



ORBIT:

LaunchSite	Orbit	mean	count
KSC LC 39A	GTO	0.666667	9
	ISS	1.000000	5
	LEO	1.000000	2
	SO	0.000000	1
	VLEO	0.800000	5

KSC LC3A site has relatively lover success rate at GTO orbit but is successful to ISS a VLEO with 100% resp. 80% success rates.

ORBIT vs. PAYLOAD:

LaunchSite	Orbit	PayloadMassCategory*	mean	count
KSC LC 39A	GTO	Low PayloadMass	0.666667	9
		High PayloadMass	NaN	0
	ISS	Low PayloadMass	1.000000	3
		High PayloadMass	1.000000	2
	LEO	Low PayloadMass	1.000000	2
		High PayloadMass	NaN	0
	VLEO	Low PayloadMass	NaN	0
		High PayloadMass	0.800000	5

- Launches to ISS orbit was successful regardless the payload.
- Interestingly VLEO orbit reached 80% success rates even with payload over 7 500 kg.
- On the other hand, GTO orbit has success rate less than 70% with payload under 7 500 kg.

BOOSTER VERSION:

- Booster version FT contributes the most to launch success in Payload range up to 6 000 kg.
- However, the only successful attempt with payload over 8 000 kg were done with booster B4.

* High payload was defined as payload over 7 500 kg.

Conclusions

To maximize first stage landing success rate:

- KSC LC3A is the launch site with the highest success rate (76,9%).
- Almost half (41.7%) of all successful landing were executed at this site.
- If possible, utilize ISS and LEO orbits and FT boosters for payloads under 7500 kg.
- For payloads over 7 500 kg is preferred VLEO orbit and B4 booster.

Good luck!

Appendix

- Space X Wikipedia web page
- Data collection – jupyter notebook
- Web scraping – jupyter notebook
- Data wrangling – jupyter notebook
- EDA – Data visualization
- EDA – SQL
- Interactive map – Folium
- Interactive dashboard – Plotly
- Predictive analysis – jupyter notebook

Thank you!