IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Christoph Promberger
July 4, 2022

Outline

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

Executive Summary

- Methodology (see methodology section for more details)
  - Data collection methodology:
  - Perform data wrangling
  - Perform exploratory data analysis (EDA) using visualization and SQL
  - Perform interactive visual analytics using Folium and Plotly Dash
  - Perform predictive analysis using classification models
- Key results (see results section for more details):
  - The average success rate for a SpaceX launch is 66%, but it increased significantly over the years (from 33% in 2014/2015 to 90% in 2019)
  - Our best model (Decision Tree classifier) can correctly predict the outcome in 17 out of 18 launches contained in the test set, yielding an accuracy of 94.4%.

Introduction

- Project background and context
  - In order to compete against Space X, we have to understand better their cost of each launch.
  - The key determinant of the launch price is whether the first stage does land (and can be reused) or not.
  - Instead of using rocket science, we want to approach this question the "data science way", by using publicly available information to train a machine learning model to predict the launch success.
- Problems you want to find answers
  - For a given launch (with given parameters), will the first stage land or not?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Webscraping
  - SpaceX's public API
- Perform data wrangling
  - Calculate number of launches from each site, occurrence of each orbit and mission outcomes
  - Create landing outcome label, save it to dataframe and calculate average success rate
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Evaluate 4 different models (Logistic Regression, SVM, Decision Tree and KNN)
  - Optimize parameters for each of these models with the help of GridSearchCV

Data Collection

Data was collected in 2 ways:

1. Accessing SpaceX's public API [https://api.spacexdata.com/v4/launches/past, https://api.spacexdata.com/v4/rockets/, https://api.spacexdata.com/v4/launchpads/, https://api.spacexdata.com/v4/payloads/, https://api.spacexdata.com/v4/cores/]

2. Web Scraping from Wikipedia Article on List of Falcon 9 and Falcon Heavy launches [https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches]

Data Collection – SpaceX API

Find me on github:
https://github.com/prombi/IBM-Capstone-Project/blob/f0b3313e1e06b2a0c6ace4088414642cf7a5be99/01%20-%20Data%20Collection%20API.ipynb

https://api.spacexdata.com/v4/launches/past

```
response = requests.get(spacex_url)

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

data

Retrieve main data from SpaceX API

https://api.spacexdata.com/v4/rockets/, https://api.spacexdata.com/v4/launchpads/, https://api.spacexdata.com/v4/payloads/, https://api.spacexdata.com/v4/cores/

```
data = data[['rocket', 'payloads', 'launchpad', 'cores', 'flight_number', 'date_utc']]
…
getBoosterVersion(data)
getLaunchSite(data)
…
data_new = pd.DataFrame(launch_dict)
```

data_new

Clean data, translate ids into meaningful names with the help of further SpaceX APIs

```
data_falcon9 = data_new[data_new['BoosterVersion']!='Falcon 1'].copy()
```

data_falcon9

Filter for falcon9 launches

```
mean_payload = data_falcon9['PayloadMass'].mean()

data_falcon9['PayloadMass'].replace(np.nan, mean_payload, inplace=True)
```

Data Wrangling: Deal with missing values

```
data_falcon9.to_csv('dataset_part_1.csv', index=False)
```

Export to .csv

Data Collection - Scraping

Find me on github:
https://github.com/prombi/IBM-Capstone-Project/blob/f0b3313e1e06b2a0c6ace4088414642cf7a5be99/02%20-%20Data%20Collection%20webscraping.ipynb

https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922

```
response =
requests.get(static_url)

soup =
BeautifulSoup(response.content)
```

Retrieve soup object from Wikipedia Site

soup

```
html_tables =
soup.find_all('table')
first_launch_table =
html_tables[2]
…
```

Get html table, extract column names

data_new

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

Parse html table to create launch-dict

launch_dict

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

Create dataframe from launch_dict

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

Export to .csv

Data Wrangling

Find me on github:
https://github.com/prombi/IBM-Capstone-Project/blob/f0b3313e1e06b2a0c6ace4088414642cf7a5be99/03%20-%20Data%20Wrangling.ipynb

Import launch data from saved csv

```
df= pd.read_csv("(...)/dataset_part_1.csv")
```

→ *df* →

Task 1: Calculate number of launches from each site

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

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

Task 2: Calculate the number and occurrence of each orbit

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

```
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
```

Task 3: Calculate number of mission outcomes

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

Task 4: Create landing outcome label and save it to df

```
bad_outcomes=set(landing_outcomes.keys()[[1,3,5,6,7]])

landing_class = [0 if (outcome in bad_outcomes) else 1 for outcome in df['Outcome']]

df['Class']=landing_class
```

Calculate average success rate

```
df["Class"].mean()
```

```
0.6666666666666666
```

Export to .csv

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

EDA with Data Visualization

Find me on github:
https://github.com/prombi/IBM-Capstone-Project/blob/f0b3313e1e06b2a0c6ace4088414642cf7a5be99/05%20-%20EDA%20with%20Visualization.ipynb

Plotted Charts:

1. Relationship between Flight Number and Launch Site (Catplot FlightNumber vs LaunchSite): Check whether the usage of different launch sites over time has an impact on the success rate of these sites

2. Relationship between Payload and Launch Site (Catplot Payload vs. LaunchSite): Check whether certain payload mass ranges are (not) launched from certain sites.

3. Relationship between Orbit type and success rate (Bar plot Payload vs. LaunchSite): See how the orbit type impacts success rate.

4. Relationship between Flight Number and Orbit Type (Catplot FlightNumber vs LaunchSite): Check whether the for different orbits the success depends on flight number. This is the case for LEO orbit, whereas for GTO orbit there is no such relationship.

5. Relationship between Payload and Orbit Type (Catplot Payload vs. Orbit Type): Check dependence of success rate on payload & orbit.

6. Launch Success yearly trend (Line Plot Years vs. Success Rate): See how success rate develops over time. Clear upward trend from 2010 to 2020!

EDA with SQL

Find me on github:
https://github.com/prombi/IBM-Capstone-Project/blob/f0b3313e1e06b2a0c6ace4088414642cf7a5be99/04%20-%20EDA%20with%20SQL.ipynb

SQL queries performed:

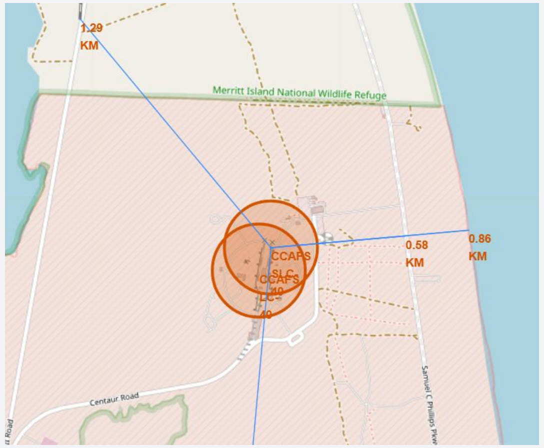
1. Display the names of the unique launch sites in the space mission:
   %sql SELECT DISTINCT launch_site FROM SPACEXDATASET
2. Display 5 records where launch sites begin with the string 'CCA':
   %sql SELECT * FROM SPACEXDATASET WHERE launch_site LIKE 'CCA%' LIMIT 5
3. Display the total payload mass carried by boosters launched by NASA (CRS):
   %sql SELECT COUNT(DATE), SUM(payload_mass__kg_) FROM SPACEXDATASET WHERE customer = 'NASA (CRS)'
4. Display average payload mass carried by booster version F9 v1.1:
   %sql SELECT COUNT(DATE) AS Count, AVG(payload_mass__kg_) AS Avg_Payload_Mass FROM SPACEXDATASET WHERE booster_version = 'F9 v1.1'
5. List the date when the first successful landing outcome in ground pad was achieved:
   %sql SELECT MIN(Date) FROM SPACEXDATASET WHERE landing__outcome = 'Success (ground pad)'
6. List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000:
   %sql SELECT DISTINCT(booster_version) FROM SPACEXDATASET WHERE landing__outcome = 'Success (drone ship)' AND payload_mass__kg_ BETWEEN 4000 AND 6000
7. List the total number of successful and failure mission outcomes:
   %sql SELECT mission_outcome, COUNT(Date) AS COUNT FROM SPACEXDATASET GROUP BY mission_outcome
8. List the names of the booster_versions which have carried the maximum payload mass:
   %sql SELECT DISTINCT(booster_version), payload_mass__kg_ FROM SPACEXDATASET WHERE payload_mass__kg_ = (SELECT MAX(payload_mass__kg_) FROM SPACEXDATASET)
9. List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015:
   %sql SELECT Date, landing__outcome, booster_version, launch_site FROM SPACEXDATASET WHERE landing__outcome = 'Failure (drone ship)' AND YEAR(Date) = '2015'
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) AS Count FROM SPACEXDATASET WHERE Date BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY landing__outcome ORDER BY Count DESC

Build an Interactive Map with Folium

Find me on github:
https://github.com/prombi/IBM-Capstone-Project/blob/f0b3313e1e06b2a0c6ace4088414642cf7a5be99/06%20-%20lab_jupyter_launch_site_location.ipynb

Map objects created:

- Per each launch site (4 total)
    - Circle around launch site → to find the sites on the map:
      circle = folium.Circle(coordinate, radius=200, color='#d35400', fill=True).add_child(folium.Popup(name))
    - Marker with launch site name → to identify the site names:
      marker = folium.map.Marker(coordinate, icon=DivIcon(ion_size=(20,20), icon_anchor=(0,0), html='<div style="font-size: 12; color:#d35400;">%s</div>' % name,))
    - Marker cluster with success/failed launches → to see the number of and success of all launches from this site:
      marker = folium.Marker(location = [record['Lat'], record['Long']], icon = folium.Icon(color='white', icon_color=record['marker_color']))
- For CCAFS SLC 40:
    - Marker and line to closest coast
    - Marker and line to closest city
    - Marker and line to closest highway
    - Marker and line to closest railroad



Build a Dashboard with Plotly Dash

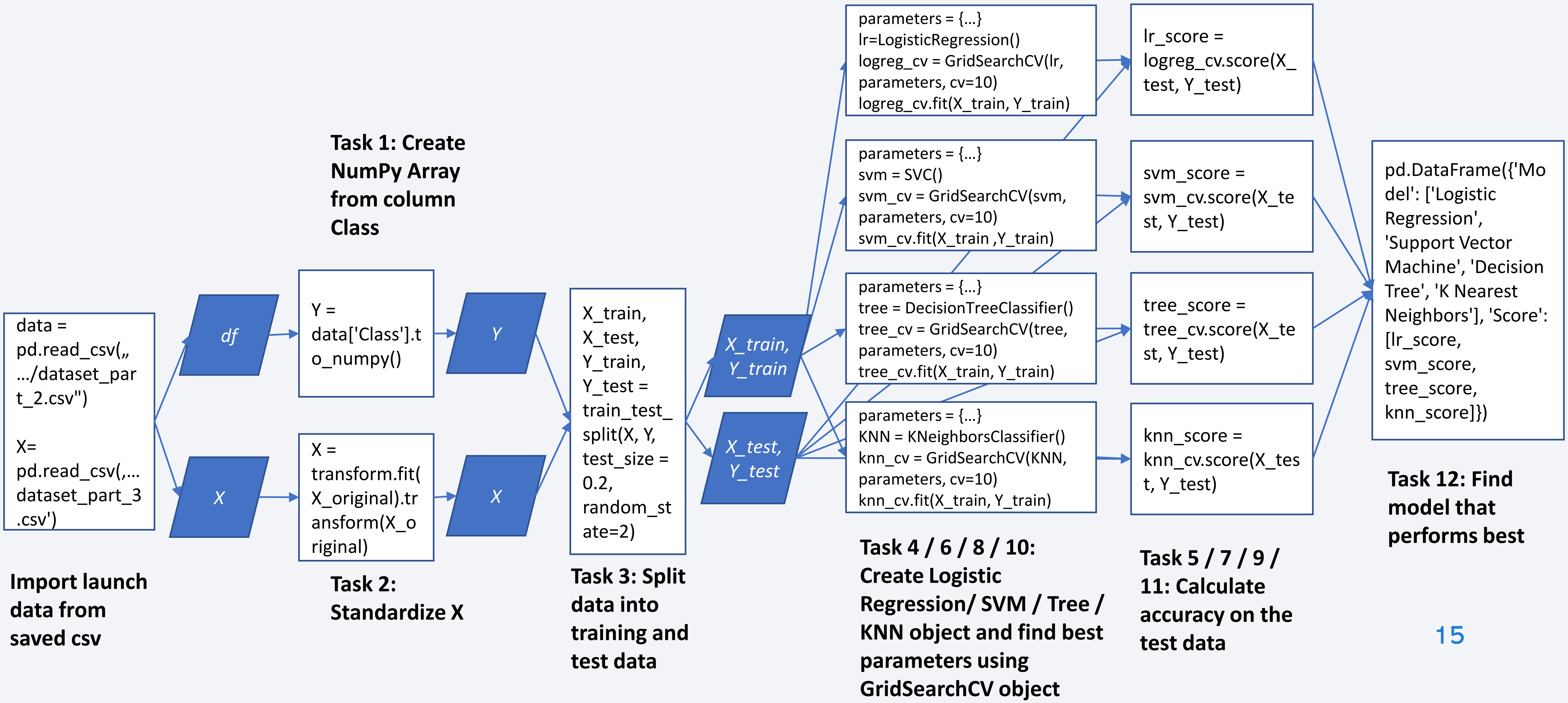
Find me on github:
https://github.com/prombi/IBM-Capstone-Project/blob/f0b3313e1e06b2a0c6ace4088414642cf7a5be99/07%20-%20spacex_dash_app.py

Plots/graphs and interactions added to the dashboard:

1. Dropdown site selection
   → to enable the user to drill down the following anylses into specific sites
2. Pie chart showing successful launches (if ‘all sites’ selected), respectively success vs. fail counts for the selected site
   → giving an overview of success rates
3. Slider to select payload range
   → allow user to drill down analysis by payload ranges
4. Scatter chart to show correlation btw. Payload and launch success
   → overview of success for different payloads

Predictive Analysis (Classification)

Find me on github:
https://github.com/prombi/IBM-Capstone-Project/blob/f0b3313e1e06b2a0c6ace4088414642cf7a5be99/08%20-%20SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb



Results

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

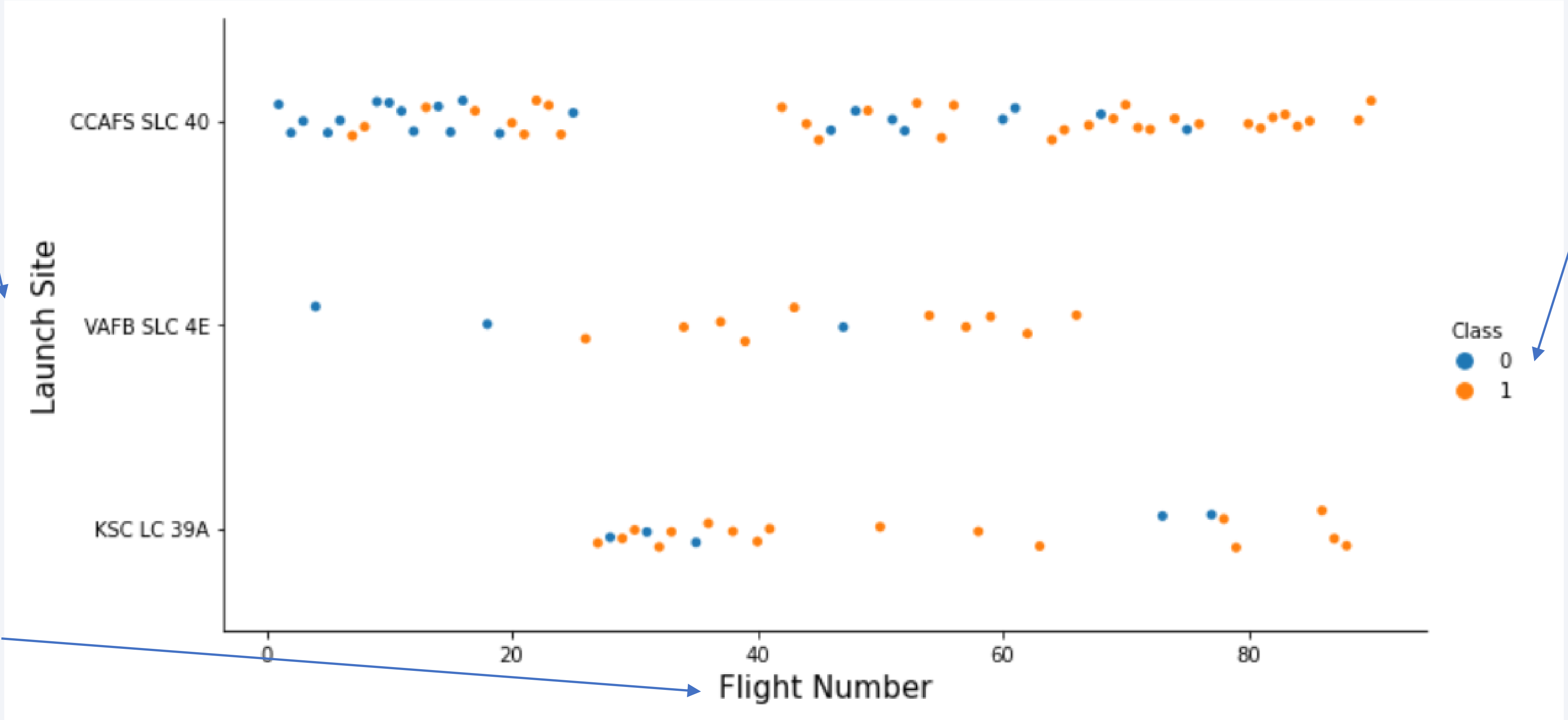
→ See next slides

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

Launch site



Class:
0=fail,
1=success

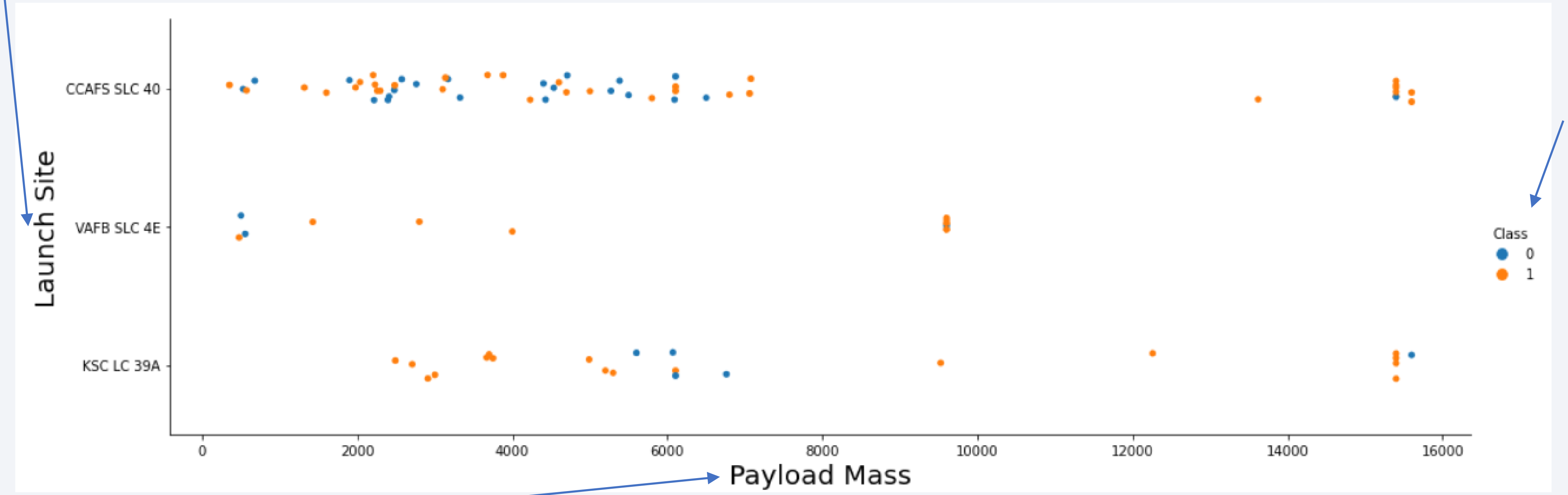
Flight Number

Findings:

1. CCAFS SLC40 is the Launch site with the most starts (55) as compared to KSC (22) and VAFB (13)
2. Particularly in the beginning (first 25 starts) when the success rate was still quite low, CCAFS was used almost exclusively (only 2 starts from VAFB, 23 starts from CCAFS). Counting only starts from 26 onwards, success rate for CCAFS is also at 75%

Payload vs. Launch Site

Launch site



Class:
0=fail,
1=success

Payload mass

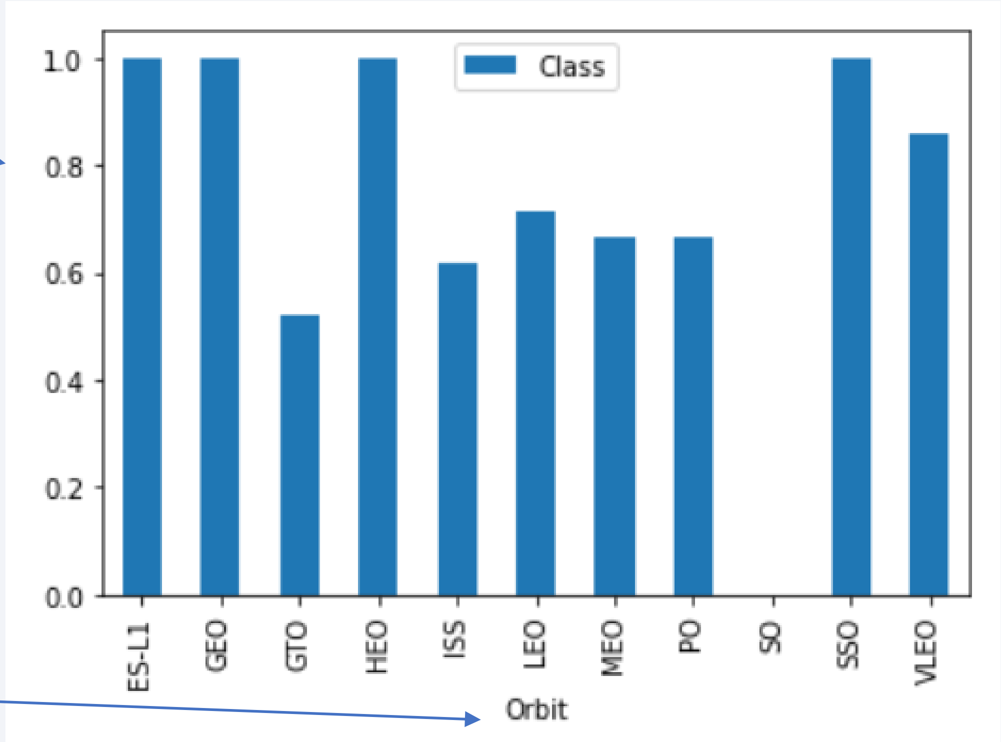
Findings:

1. Lighter payloads (<8000kg) are predominantly performed from CCAFS SLC 40
2. Heavy payloads (<10000kg) are only done from KSC and CCAFS, not VAFB

Success Rate vs. Orbit Type

Success rate

Orbit

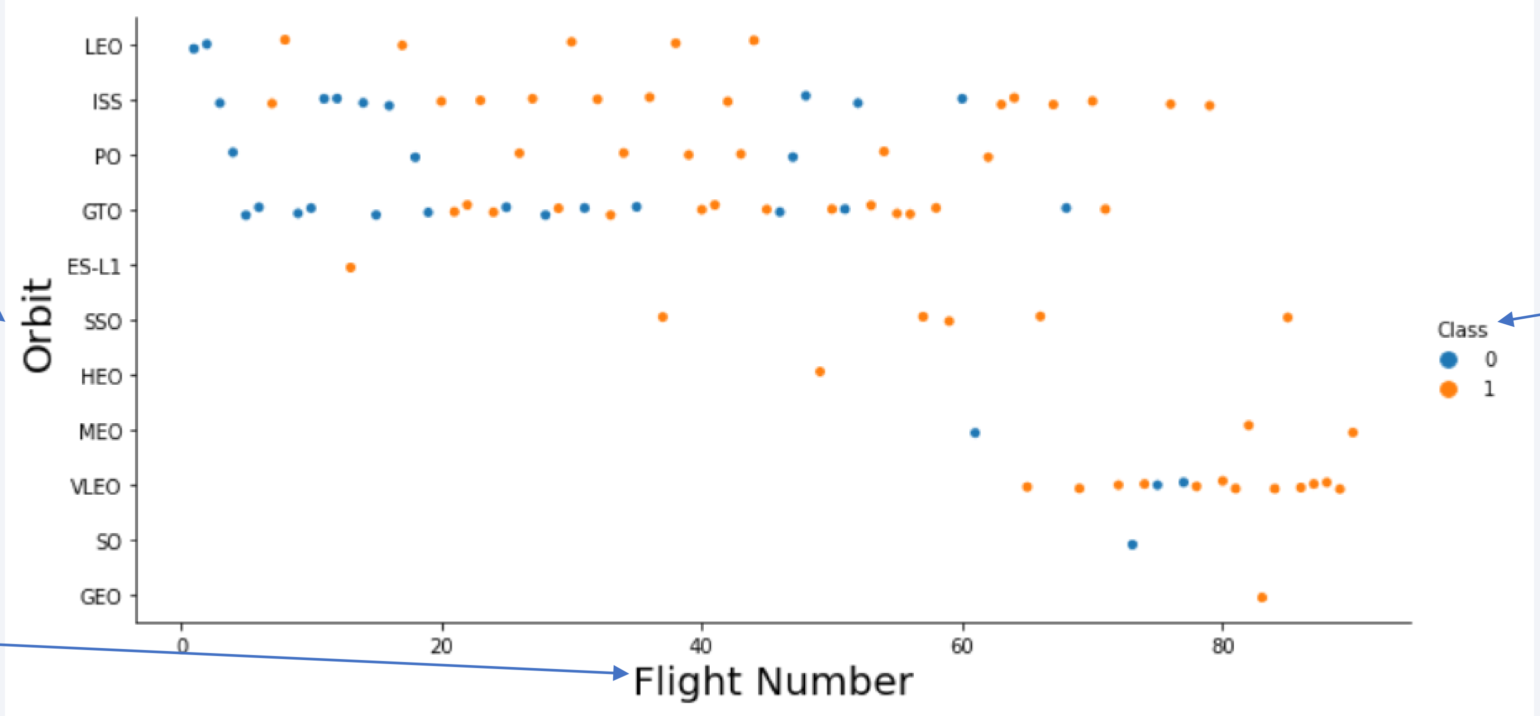


Finding: Orbits with highest success rate (counting only orbits with 5 or more launches):

1. SSO: 100% @ 5 launches
2. VLEO: 86% @ 14 launches
3. LEO: 71% @ 7 launches
4. PO: 67% @ 9 launches
5. ISS: 62% @ 21 launches
6. GTO: 52% @ 27 launches

Flight Number vs. Orbit Type

Orbit



Flight no.

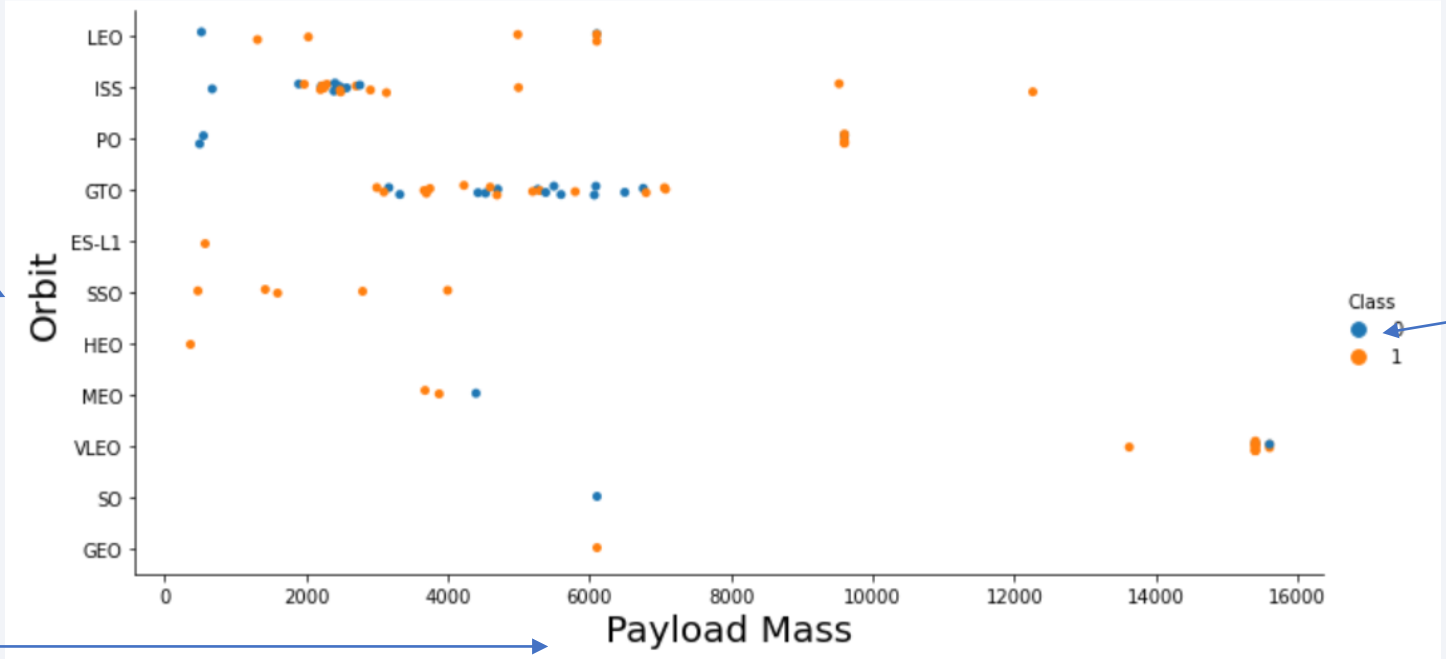
Class:
0=fail,
1=success

Findings:

1. For LEO orbit, the success increases with number of flights;
2. No such relationship for GTO orbit.

Payload vs. Orbit Type

Orbit



Class:
0=fail,
1=success

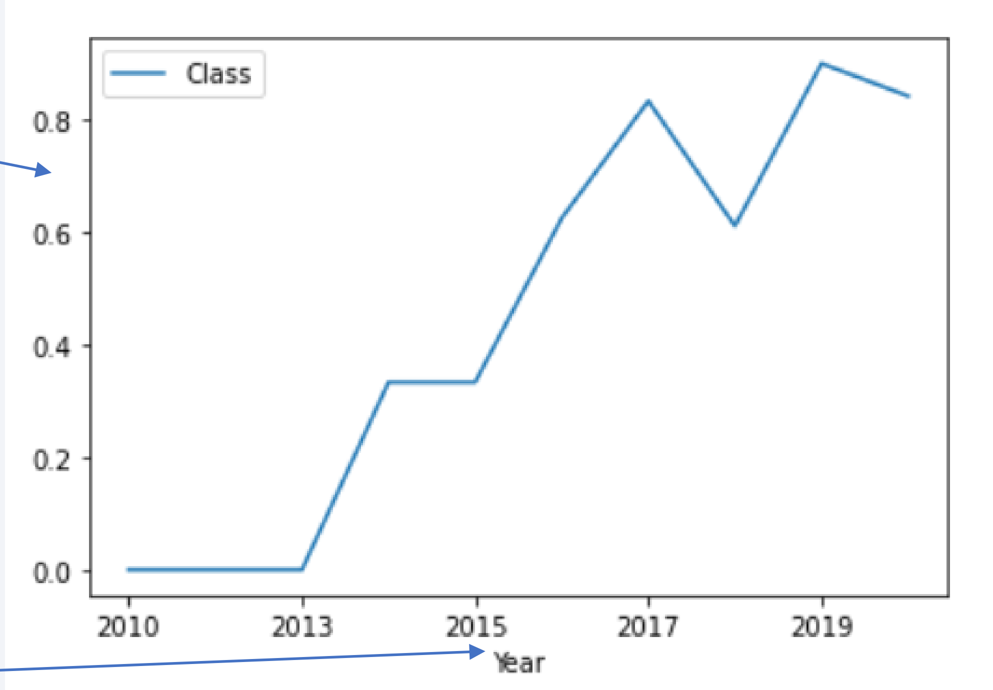
Payload mass

Findings:

1. Heavy payloads (>= 10000kg) are only launched to ISS, PO and VLEO, with a very high success rate (only 1 fail in VLEO)
2. For GTO, the payload is btw. ~3500 and ~7000 kg, with successful and failed landing outcomes evenly distributed

Launch Success Yearly Trend

Success rate



Year

Findings:

1. There us a very clear upward trend (learning curve) btw. 2013 and 2017
2. After 2017, the learning curve seems to flatten, with drops in 2018 and 2020 (but still 2020 is on the level of 2017)
3. The highest success rate (90%) was achieved in 2019

All Launch Site Names

Display the names of the unique launch sites in the space mission

In [10]:
```
%sql SELECT DISTINCT launch_site FROM SPACEXDATASET
```

```
 * ibm_db_sa://hdw01742:***@ea286ace-86c7-4d5b-8580-3fbfa46
Done.
```

Out[10]:

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

There are 4 distinct launch sites

Launch Site Names Begin with 'CCA'

Display 5 records where launch sites begin with the string 'CCA'

```
%sql SELECT *  FROM SPACEXDATASET WHERE launch_site LIKE 'CCA%' LIMIT 5
```

```
 * ibm_db_sa://hdw01742:***@ea286ace-86c7-4d5b-8580-3fbfa46b1c66.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31505/bludb
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	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

… 5 examples of records where launch site name begins with „CCA“

Total Payload Mass

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

```
%sql SELECT COUNT(DATE), SUM(payload_mass__kg_) FROM SPACEXDATASET WHERE customer = 'NASA (CRS)'
```

```
 * ibm_db_sa://hdw01742:***@ea286ace-86c7-4d5b-8580-3fbfa46b1c66.bs2io90l08kqb1od8lcg.databases.appdo
Done.
```

1	2
20	45596

There are in total 20 launches by NASA (CRS), with a total payload mass of 45596 kg

Average Payload Mass by F9 v1.1

Display average payload mass carried by booster version F9 v1.1 ¶

```
%sql SELECT COUNT(DATE) AS Count, AVG(payload_mass__kg_) AS Avg_Payload_Mass  FROM SPACEXDATASET
WHERE booster_version = 'F9 v1.1'
```

```
 * ibm_db_sa://hdw01742:***@ea286ace-86c7-4d5b-8580-3fbfa46b1c66.bs2io90l08kqb1od8lcg.databases.appdom
Done.
```

COUNT	avg_payload_mass
5	2928

There are in total 5 carried by booster version F9 v1.1, with an average payload mass of 2928 kg

First Successful Ground Landing Date

List the date when the first successful landing outcome in ground pad was acheived.

Hint:Use min function

```
%sql SELECT MIN(Date) FROM SPACEXDATASET WHERE landing__outcome = 'Success (ground pad)'
```

```
 * ibm_db_sa://hdw01742:***@ea286ace-86c7-4d5b-8580-3fbfa46b1c66.bs2io90l08kqb1od8lcg.databas
Done.
```

1
2015-12-22

The first successful landing outcome in ground pad was on 12/22, 2015

Successful Drone Ship Landing with Payload between 4000 and 6000

List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

```
%sql SELECT DISTINCT(booster_version) FROM SPACEXDATASET
WHERE landing__outcome = 'Success (drone ship)'
AND payload_mass__kg_ BETWEEN 4000 AND 6000
```

```
 * ibm_db_sa://hdw01742:***@ea286ace-86c7-4d5b-8580-3fbfa46b1c66.bs2io90l08kqb1od8lcg.databases.appdomain.clou
Done.
```

booster_version
F9 FT B1021.2
F9 FT B1031.2
F9 FT B1022
F9 FT B1026

There were 4 boosters which had success in drone ship and payload btw. 4000 and 6000 kg

Total Number of Successful and Failure Mission Outcomes

List the total number of successful and failure mission outcomes

```
%sql SELECT mission_outcome, COUNT(Date) AS COUNT FROM SPACEXDATASET GROUP BY mission_outcome
```

```
 * ibm_db_sa://hdw01742:***@ea286ace-86c7-4d5b-8580-3fbfa46b1c66.bs2io90l08kqb1od8lcg.databases.app
Done.
```

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

Of 101 flights, 99 had successful missions (which does not mean that the landing was also successful)

Boosters Carried Maximum Payload

List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

```
%%sql SELECT DISTINCT(booster_version), payload_mass__kg_ FROM SPACEXDATASET WHERE
 payload_mass__kg_ =
    (SELECT MAX(payload_mass__kg_) FROM SPACEXDATASET)
```

booster_version	payload_mass__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

12 distinct booster versions carried maximum payload

2015 Launch Records

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

```
%%sql SELECT Date, landing__outcome, booster_version, launch_site FROM SPACEXDATASET
WHERE landing__outcome = 'Failure (drone ship)' AND YEAR(Date) = '2015'
```

```
 * ibm_db_sa://hdw01742:***@ea286ace-86c7-4d5b-8580-3fbfa46b1c66.bs2io90l08kqb1od8lcg.databases.a
Done.
```

DATE	landing__outcome	booster_version	launch_site
2015-01-10	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
2015-04-14	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

2 failed landings in 2015

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

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) AS Count FROM SPACEXDATASET
WHERE Date BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY landing__outcome ORDER BY Count DESC
```

```
 * ibm_db_sa://hdw01742:***@ea286ace-86c7-4d5b-8580-3fbfa46b1c66.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31505/bluc
Done.
```

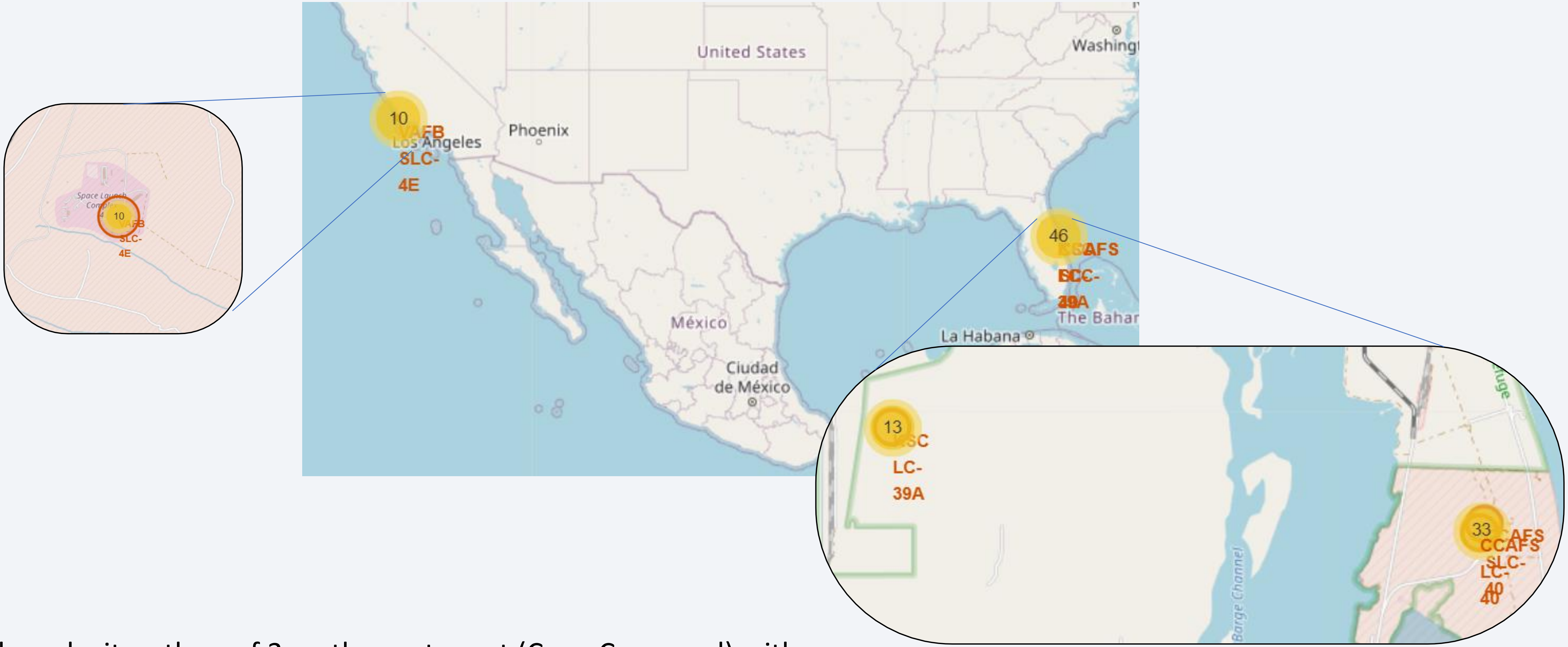
landing__outcome	COUNT
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

Most frequent landing outcomes in given timeframe were ‚No attempt‘ (10x), ‚Failure (drone ship)‘ (5x) and ‚Success (drone ship)‘ (5x)

Section 3

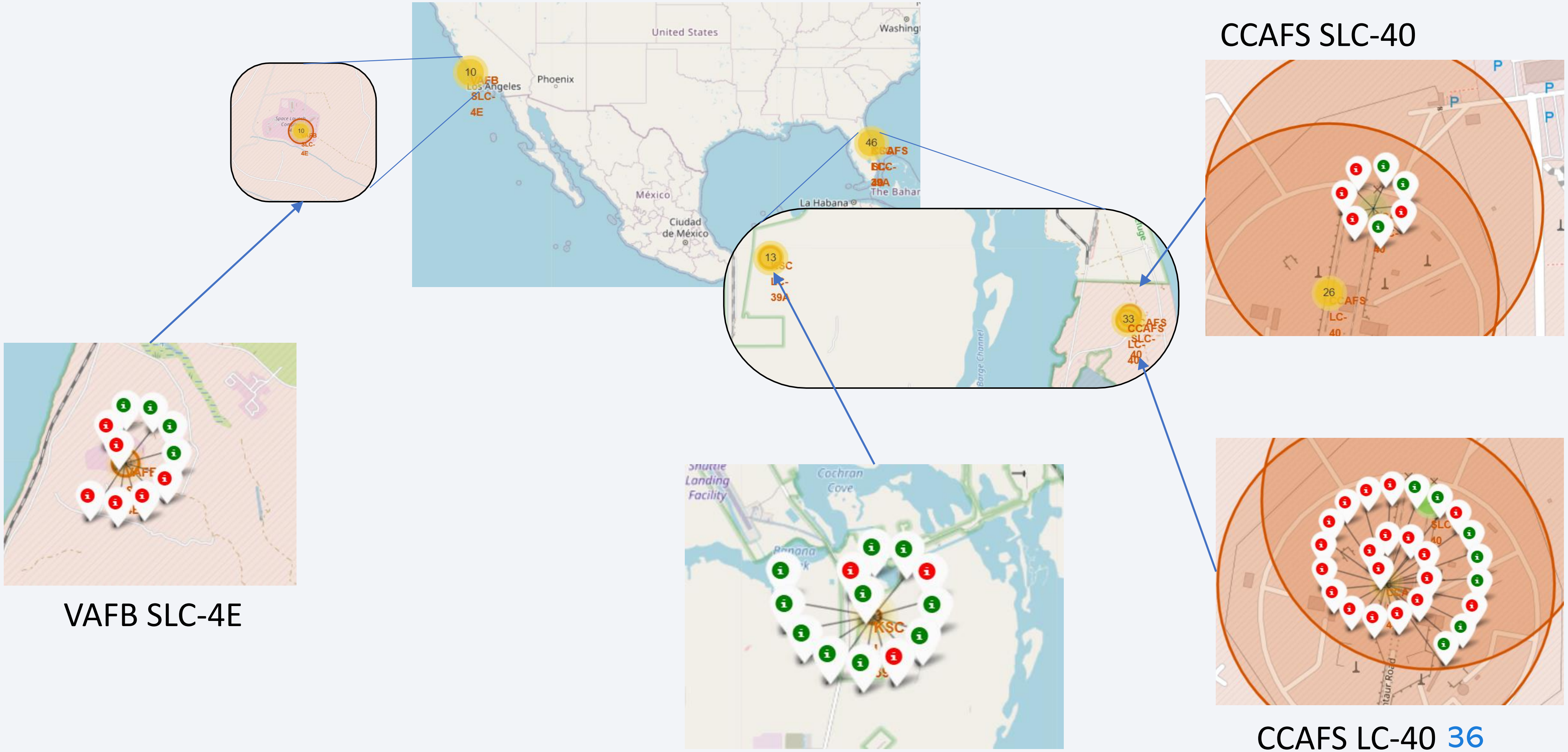
Launch Sites Proximities Analysis

Global map of all launch sites



4 launch sites, thereof 3 on the east coast (Cape Canaveral) with a total of 46 launches, and one site on the west coast with 10 launches

Launch sites deep-dive: Launch outcomes



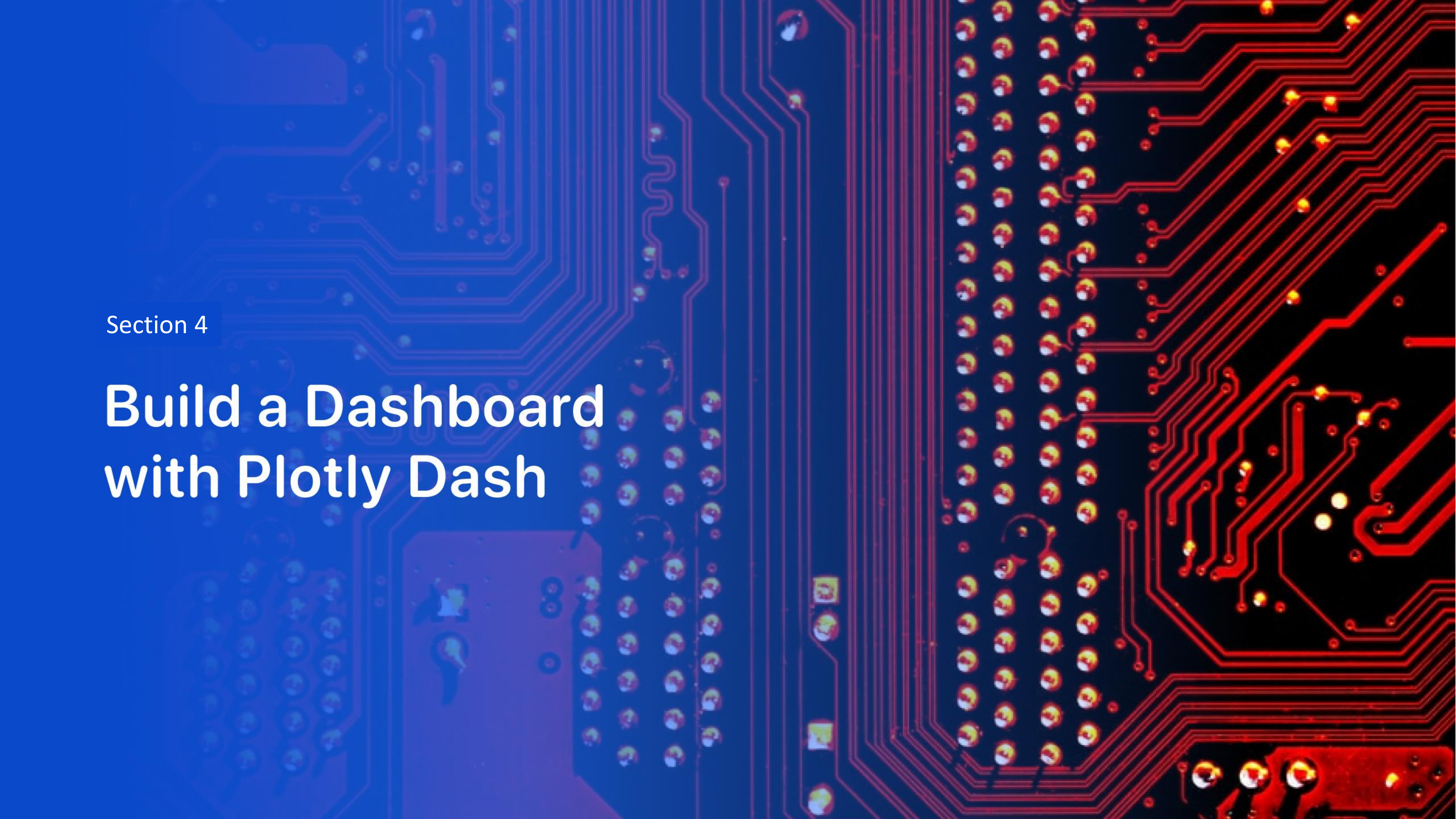
Deep dive: CCAFS SLC-40 proximities

Next railway: ~1.3km

Next coast: ~0.9km

Next highway: ~0.6km

Next city: ~17km



Section 4

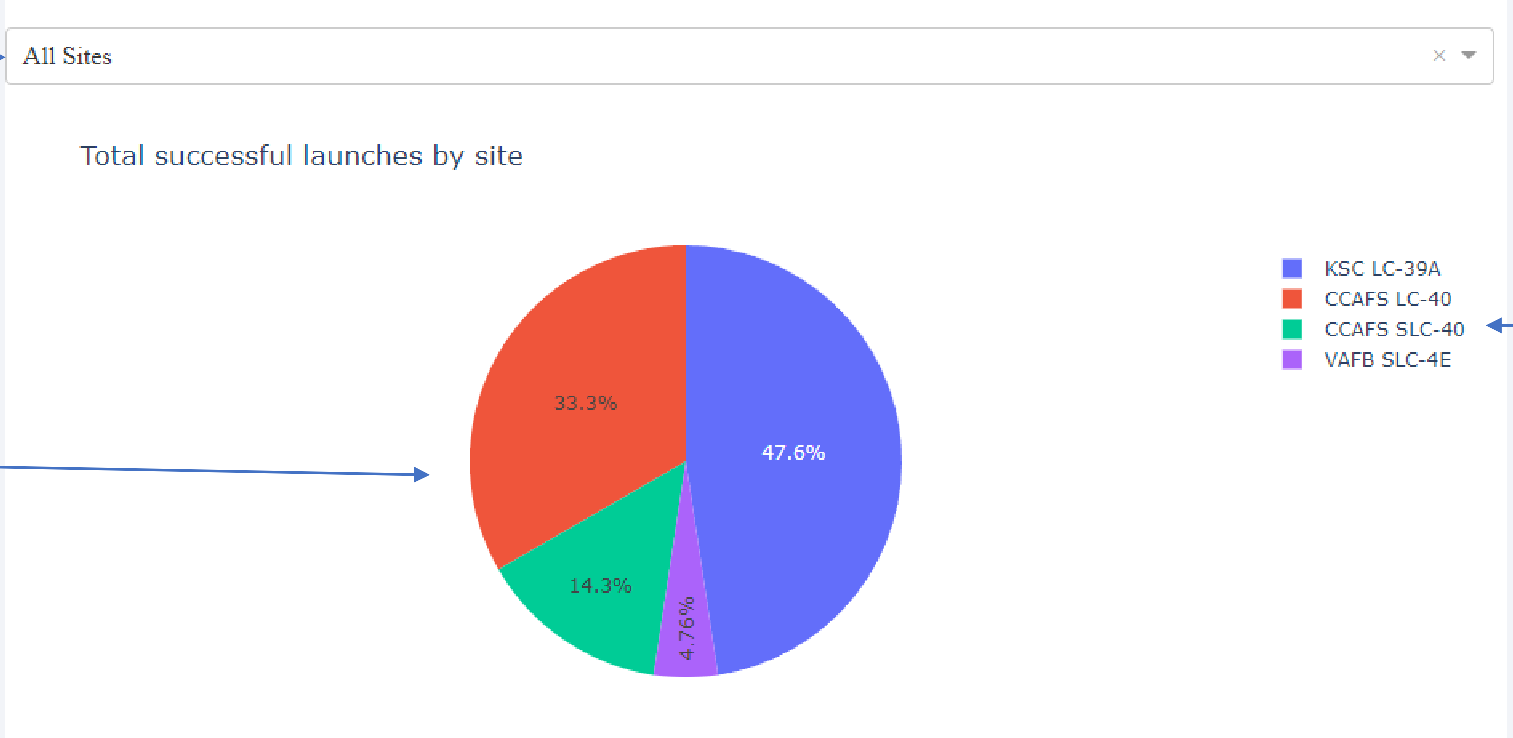
Build a Dashboard with Plotly Dash

Successful launches across all sites

Dropdown

Legend

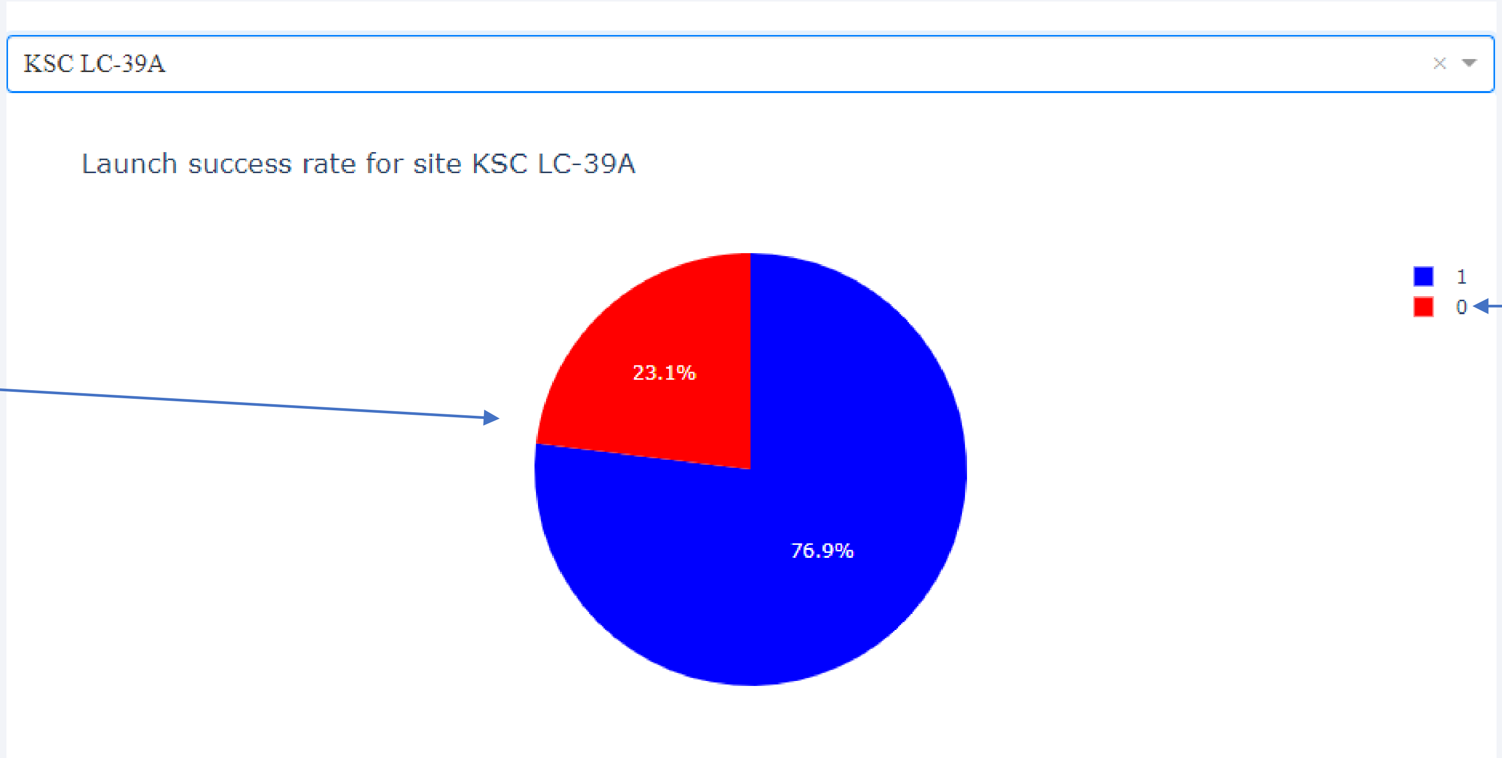
Pie chart



Finding: Most successful launches from KSC LC-39A

Launch success for site with highest success ratio

Dropdown



Pie chart

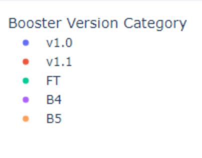
Legend
(0=fail, 1=success)

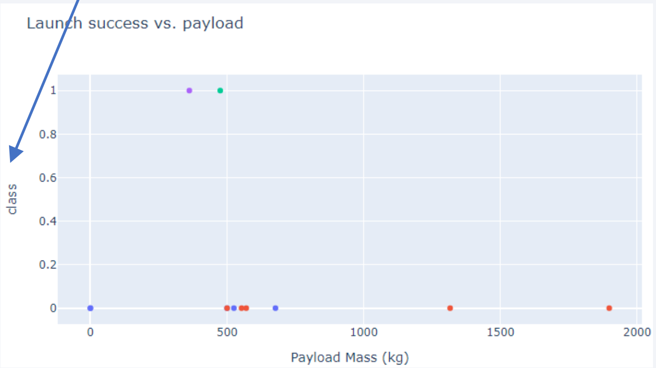
Finding: Success rate at KSC LC-39A is 76.9%

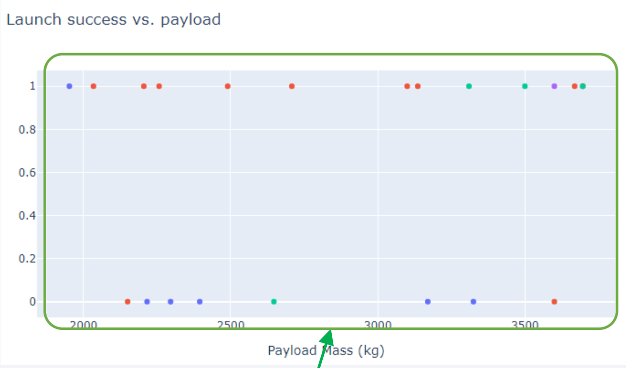
Payload vs. Launch Outcome for all sites

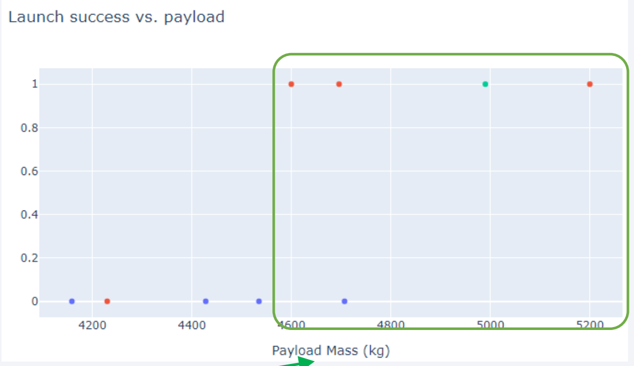
Launch outcome (1=successful, 0 = fail)

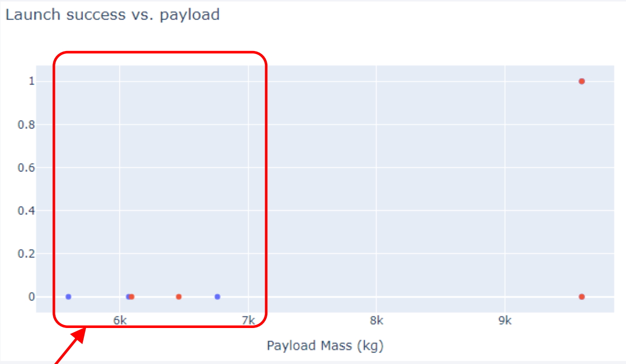
Booster version











Payload ranges with highest success rate: 1950-3700 kg, 4600-5300kg

Payload range with lowest success rate: 5600-6800 kg

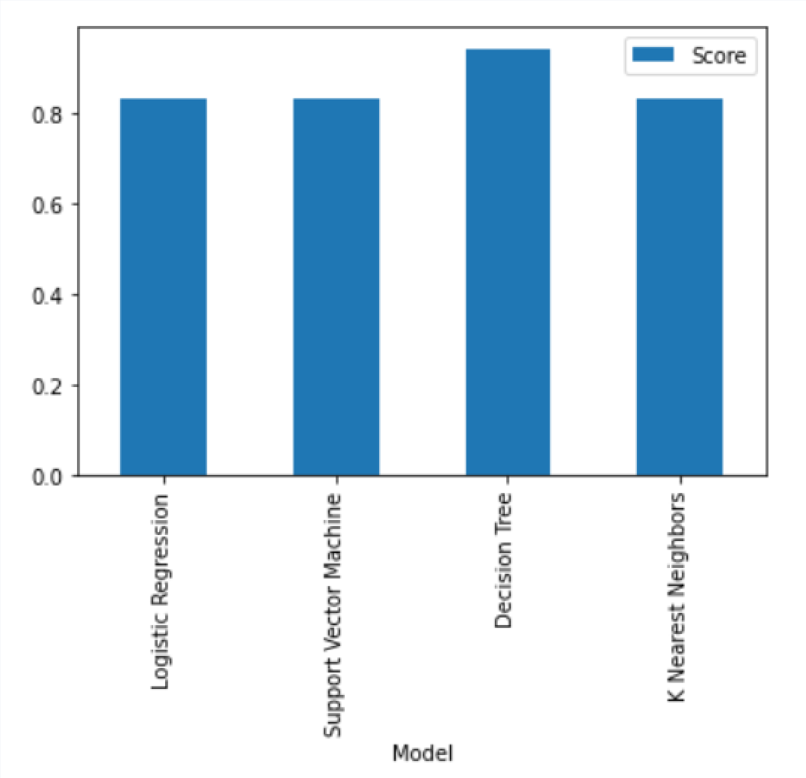
Booster versions with highest success rate:

- B5 (1/1 = 100%)
- FT (16/24 = 66%)

Section 5

Predictive Analysis (Classification)

Classification Accuracy

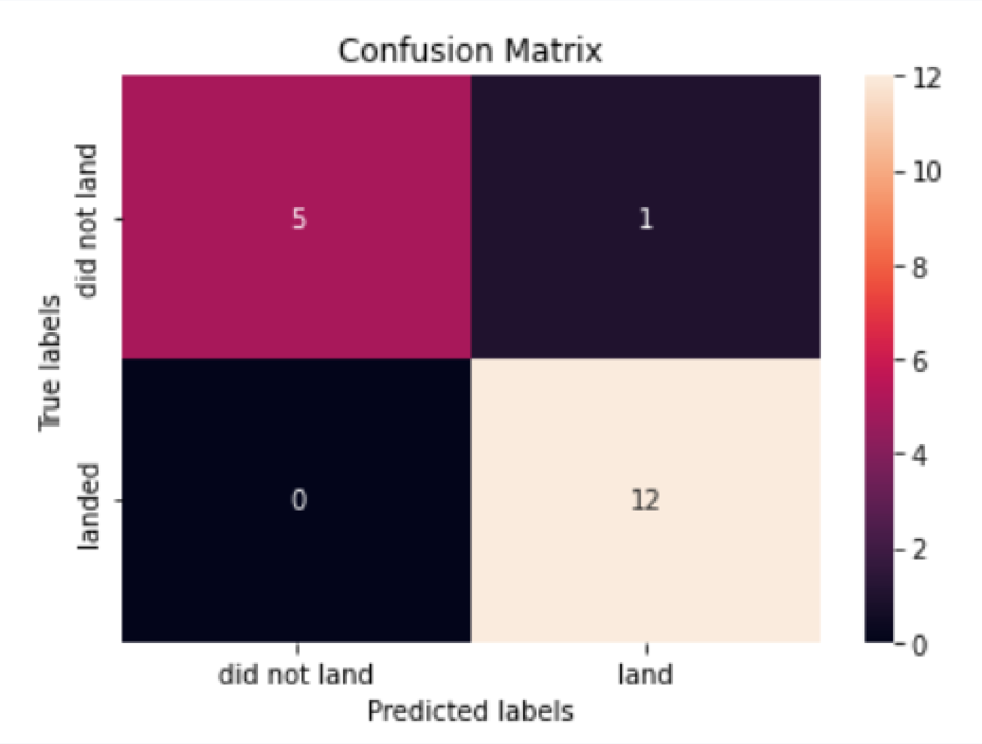


Decision tree has the highest classification accuracy on the test set (94.4%)*!

**Note: From run to run, the best parameters for the decision tree model can change and its performance can also vary btw. ~66% and 94%. We chose one of the best performing configurations here*

(parameters = {'criterion': 'gini', 'max_depth': 14, 'max_features': 'auto', 'min_samples_leaf': 1, 'min_samples_split': 10, 'splitter': 'best'})

Confusion Matrix for best model (decision tree)*



Note: From run to run, the best parameters for the decision tree model can change and its performance can also vary btw. ~66% and 94%. We chose one of the best performing configurations here

(*parameters = {'criterion': 'gini', 'max_depth': 14, 'max_features': 'auto', 'min_samples_leaf': 1, 'min_samples_split': 10, 'splitter': 'best'}*)

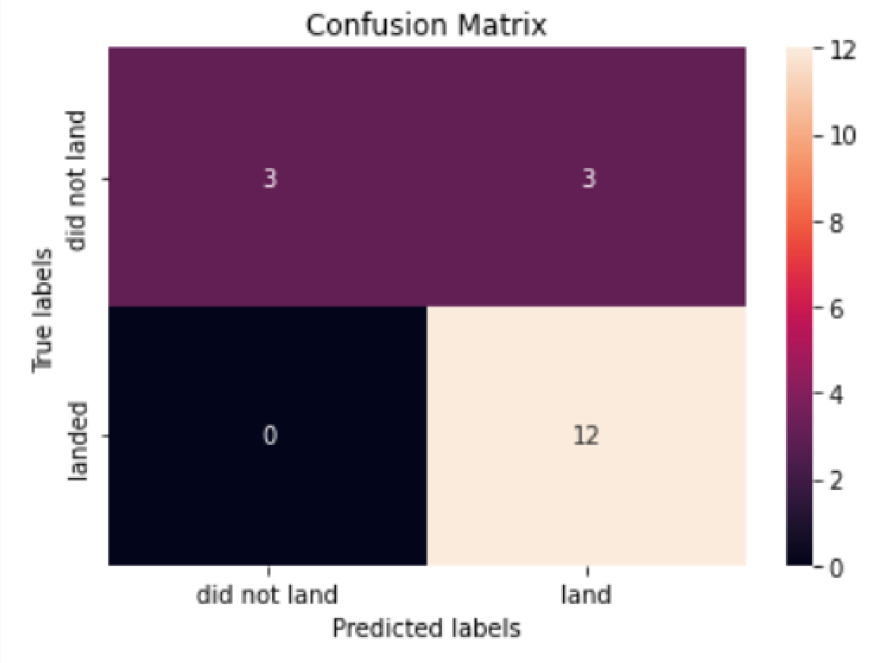
Conclusions

- LogReg, SVM and KNN yield the same outcomes:
  Accuracy = 83.3%, Confusion Matrix see Appendix
- For Decision Tree model, the best parameters change from run to run:
  - In the best case, we find 94.4% accuracy and a confusion matrix as shown in the previous slides. This is the best model that we can find, with only one false positive and no false negatives.
  - In the worst case, we find 66.6% accuracy
  - There seems to be some randomness involved in the GridSearchCV optimization algorithm. Further analysis of this algorithm is in place (but goes beyond the scope of this assignment)

Appendix

Confusion Matrix for typical model

All models apart from decision tree, i.e. LogReg, SVM, KNN, have the same performance (83.3%), and the same confusion matrix shown here:



Thank you!