IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Wojciech Sowiński
17.05.2022

Outline

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

Executive Summary

- Summary of methodologies
  - 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 all results
  - Exploratory Data Analysis result
  - Interactive analytics in screenshots
  - Predictive Analytics result

Introduction

- Project background and context

  The commercial space age is here, companies are making space travel affordable for everyone. The most successful company is SpaceX because, Falcon 9 rocket launch costs 62 million dollars, while other providers spend up to 165 million dollars each. Much of the savings are 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. Company Space Y run by Allon Musk hired data scientist for project to use machine learning for prediction of successful landing to compete with Space X. Goal of the project is to create complete machine learning pipeline that predicts outcome of rocket launching based on provided factors.

- Problems you want to find answers

- Finding determining factors of successful landing

- Determining what attributes are correlated with successful landings

Section 1

Methodology

Methodology

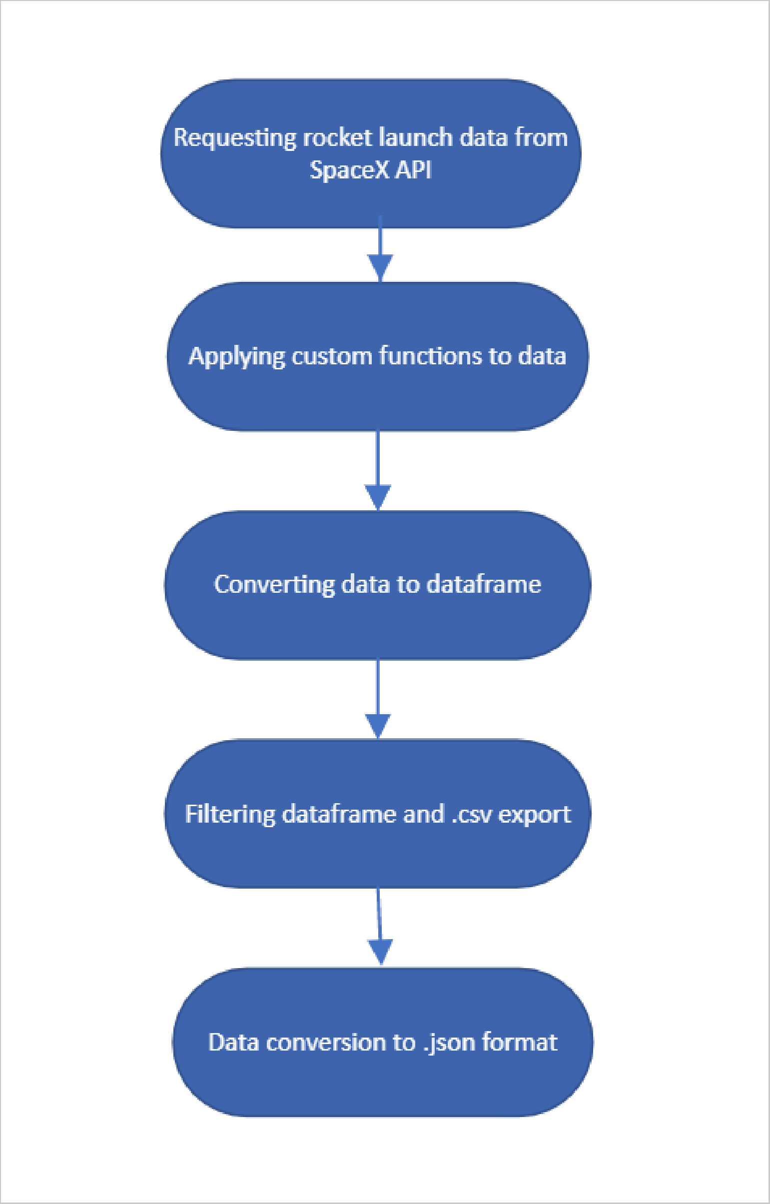
- Executive Summary
- Data collection methodology:
  - Data Collection using API
  - Data Collection with Webscraping from Wikipedia
- Perform data wrangling
  - One-hot encoding to create prediction class based on landing outcome feature
- Perform exploratory data analysis (EDA) using visualization 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

- The data collection was performer using various methods
  - Data collection using SpaceX API
  - Web scraping Falcon 9 launch records from Wikipedia with BeautifulSoup library
  - Wrangling Data using an API
  - Sampling Data to get records only for Falcon 9 from the provided dataset
  - Replacing None values with a mean of the specific columns
  - One-hot encoding for LandingPad column with NULL values

Data Collection – SpaceX API

- Flowchart of data collection using SpaceX API
- GitHub jupyter notebook for the data collection



Data Wrangling

- Code sample of data collection using SpaceX API
- GitHub jupyter notebook for the data collection

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

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

Check the content of the response

```
print(response.content)
```

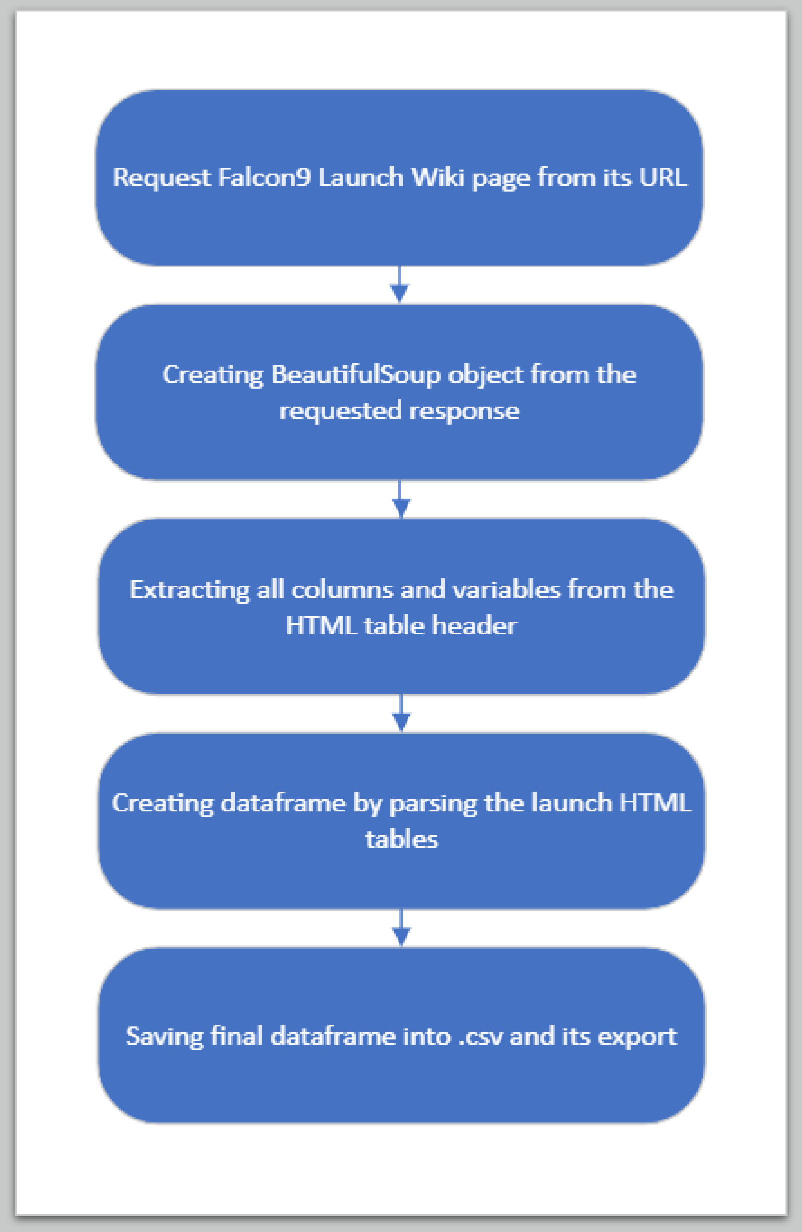
```
# Lets take a subset of our dataframe keeping only the features we want and the flight
data = data[['rocket', 'payloads', 'launchpad', 'cores', 'flight_number', 'date_utc']]
data.head()
```

```
# Calculate the mean value of PayloadMass column
payloadMean = data_falcon9['PayloadMass'].mean()
# Replace the np.nan values with its mean value
data_falcon9['PayloadMass'].fillna(payloadMean, inplace=True)
data_falcon9.head()
```

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	Longitude	Latitu
4	1	2010-06-04	Falcon 9	6123.547647	LEO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0003	-80.577366	28.561
5	2	2012-05-22	Falcon 9	525.000000	LEO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0005	-80.577366	28.561
6	3	2013-03-01	Falcon 9	677.000000	ISS	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0007	-80.577366	28.561
7	4	2013-09-29	Falcon 9	500.000000	PO	VAFB SLC 4E	False Ocean	1	False	False	False	None	1.0	0	B1003	-120.610829	34.632
8	5	2013-12-03	Falcon 9	3170.000000	GTO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B1004	-80.577366	28.561

Data Collection - Scraping

- Flowchart of web scraping
- Github jupyter notebook for web scraping



Data Wrangling

- Code sample of web scraping
- [Github jupyter notebook](#) for web scraping

```python
data = requests.get(static_url).text
```

```python
soup = BeautifulSoup(data, 'html5lib')
```

```python
column_names = []
for row in first_launch_table.find_all("th"):
    name = extract_column_from_header(row)
    if((name != None) and (len(name)) > 0):
        column_names.append(extract_column_from_header(row))
# 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
```

```python
df3.to_csv('spacex_web_scraped.csv', index=False)
```

	Flight No.	Launch site	Payload	Payload mass	Orbit	Customer	Launch outcome	Version Booster	Booster landing	Date	Time
110	111	KSC	Starlink	15600.0	LEO	SpaceX	Success\n	F9 v1.1	Success	14 March 2021	10:01
111	112	CCSFS	Starlink	15600.0	LEO	SpaceX	Success\n	F9 v1.1	Success	24 March 2021	08:28
112	113	CCSFS	Starlink	15600.0	LEO	SpaceX	Success\n	F9 v1.1	Success	7 April 2021	16:34
113	114	KSC	Crew-2	13000.0	LEO	NASA	Success\n	F9 v1.1	Success	23 April 2021	9:49
114	115	CCSFS	Starlink	15600.0	LEO	SpaceX	Success\n	F9 v1.1	Success	29 April 2021	03:44
115	116	KSC	Starlink	15600.0	LEO	SpaceX	Success\n	F9 v1.1	Success	4 May 2021	19:01
116	117	CCSFS	Starlink	15600.0	LEO	SpaceX	Success\n	F9 v1.1	Success	9 May 2021	06:42
117	118	KSC	Starlink	14000.0	LEO	SpaceX	Success\n	F9 v1.1	Success	15 May 2021	22:56
118	119	CCSFS	Starlink	15600.0	LEO	SpaceX	Success\n	F9 v1.1	Success	26 May 2021	18:59
119	120	KSC	SpaceX CRS-22	3328.0	LEO	NASA	Success\n	F9 v1.1	Success	3 June 2021	17:29
120	121	CCSFS	SXM-8	7000.0	GTO	Sirius XM	Success\n	F9 v1.1	Success	6 June 2021	04:26

Data Wrangling

- At first we had to calculate the numer and occurence on each site and each orbit and mission outcome per orbit type

Data Wrangling

- Then we had to create landing outcome label from Outcome column and export it as .csv file

```
# landing_class = 0 if bad_outcome
# landing_class = 1 otherwise
landing_class = []
for i in df['Outcome']:
    if(i is in b)
```

```
None None
None None
None None
False Ocean
```

```
for i,outcome in enumerate(landing_outcomes.keys()):
    print(i,outcome)
```

```
0 True ASDS
1 None None
2 True RTLS
3 False ASDS
4 True Ocean
5 False Ocean
6 None ASDS
7 False RTLS
```

We create a set of outcomes where the second stage did not land successfully:

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

```
{'False ASDS', 'False Ocean', 'False RTLS', 'None ASDS', 'None None'}
```

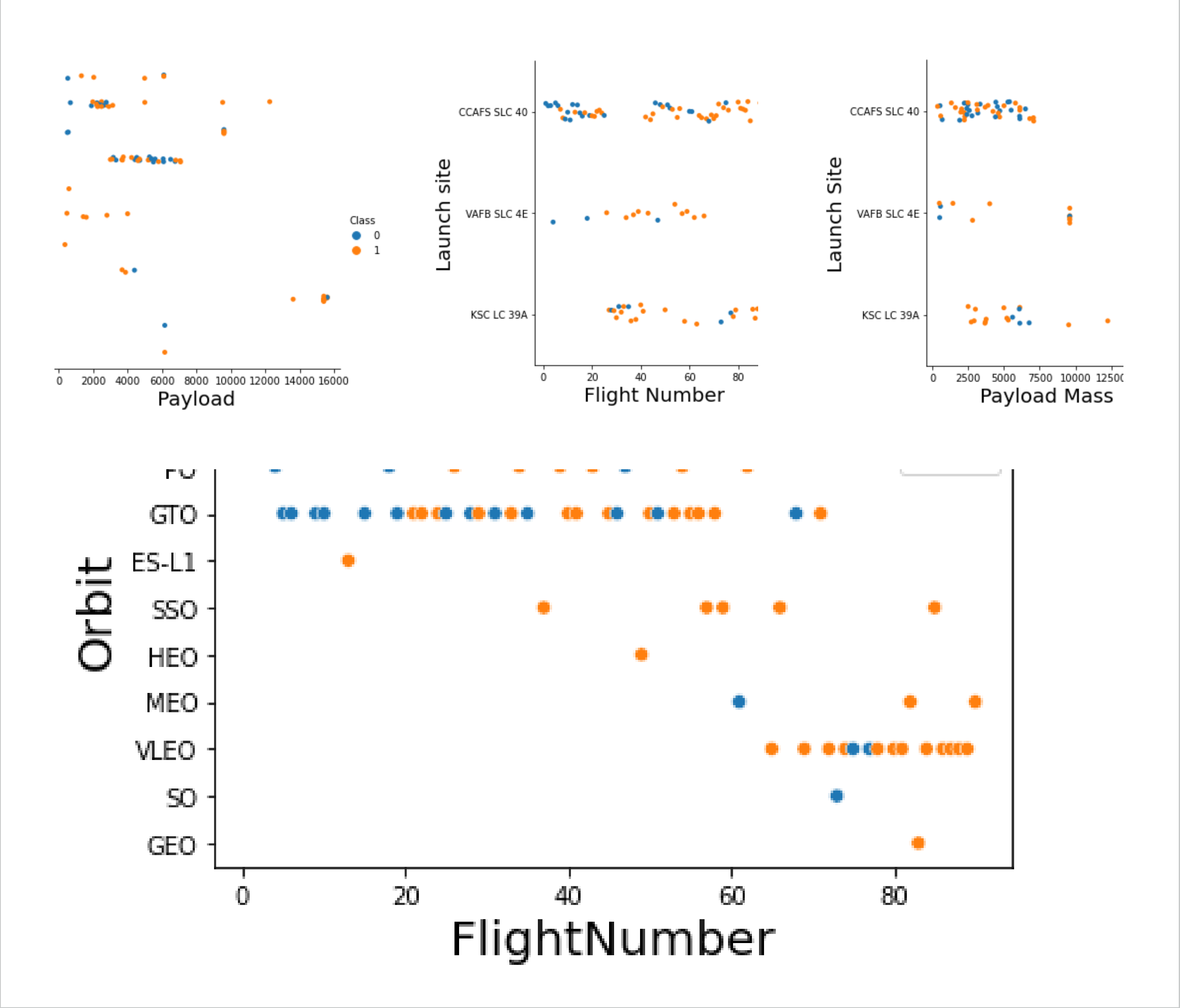
```
df["Class"].mean()
```

We can now export it to a CSV for the next section,but to make

```
df.to_csv("dataset_part\_2.csv", index=False)
```

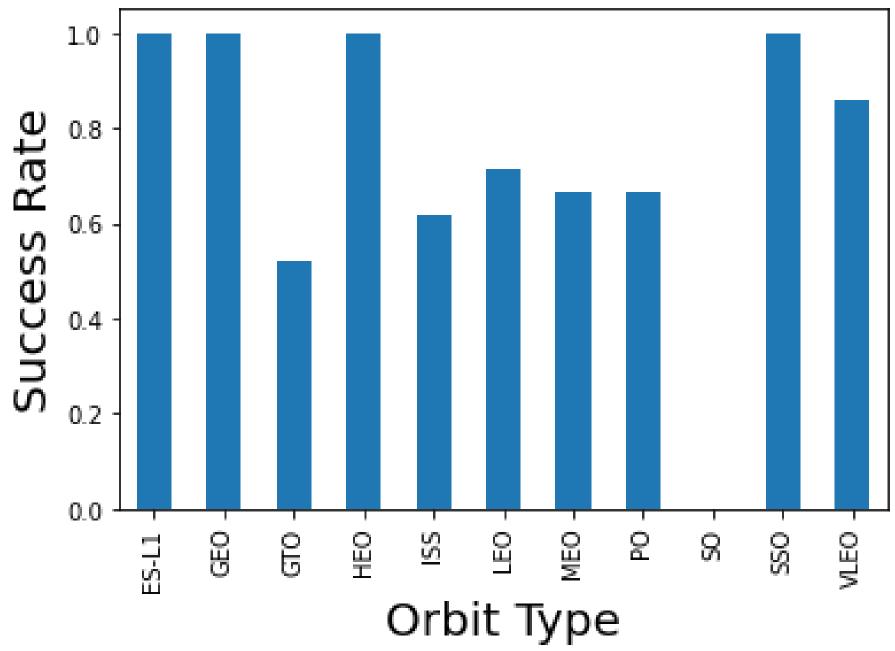
EDA with Data Visualization

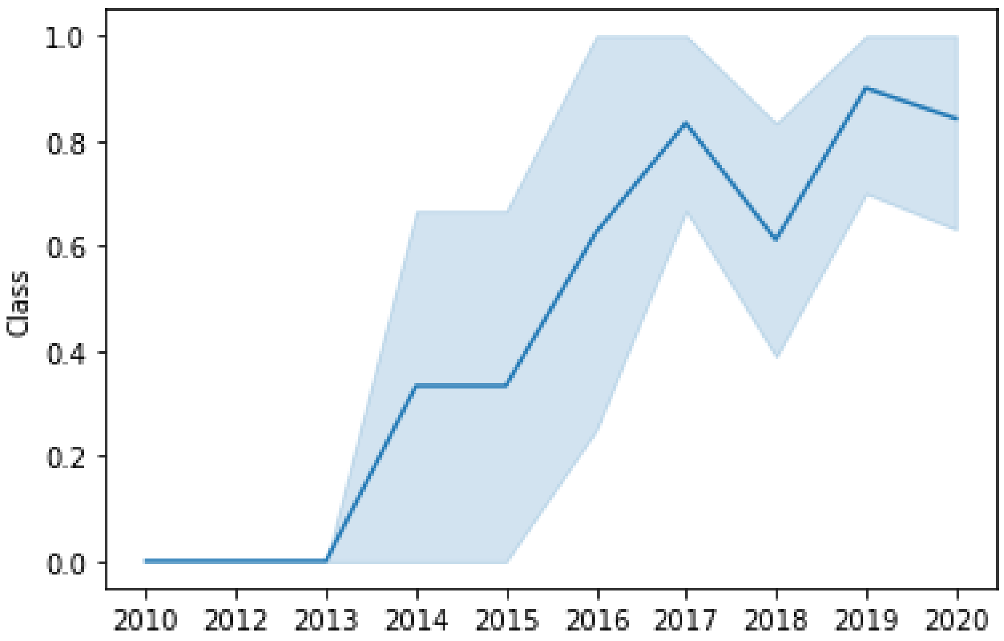
- To visualize the relationship between features we used **scatter plots** to obtain some preliminary insights about how each important feature would affect success rate. Scatter plots are used to see the correlation.



EDA with Data Visualization

- To visualize the relationship between the success rate of each orbit type, numerical columns and categorical column, we used a **bar chart** to easily differentiate success rate
- To visualize the launch success yearly trend we used **line plot** to notice that there's a trend of increasing success rate





EDA with Data Visualization

- Final step of EDA with Data Visualizaiton was performing one hot encoding for pre-selected features obtained by features engineering and exporting it to .csv file to use it in future machine learning pipeline

- Github notebook of EDA with Data Visualization

```
# HINT: use astype function
features_one_hot = features_one_hot.astype('float64')
features_one_hot.head()
```

ReusedCount	Orbit_ES-L1	Orbit_GEO	...	Serial_B1048	Serial_B1049	Serial_B1050	Serial_B1051	Serial_B1054	Serial_B1056	Serial_B1058	Serial_B1059	Serial_B1060	Serial_B1062
0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

We can now export it to a **CSV** for the next section,but to make the answers consistent, in the next lab we will provide data in a pre-selected date range.

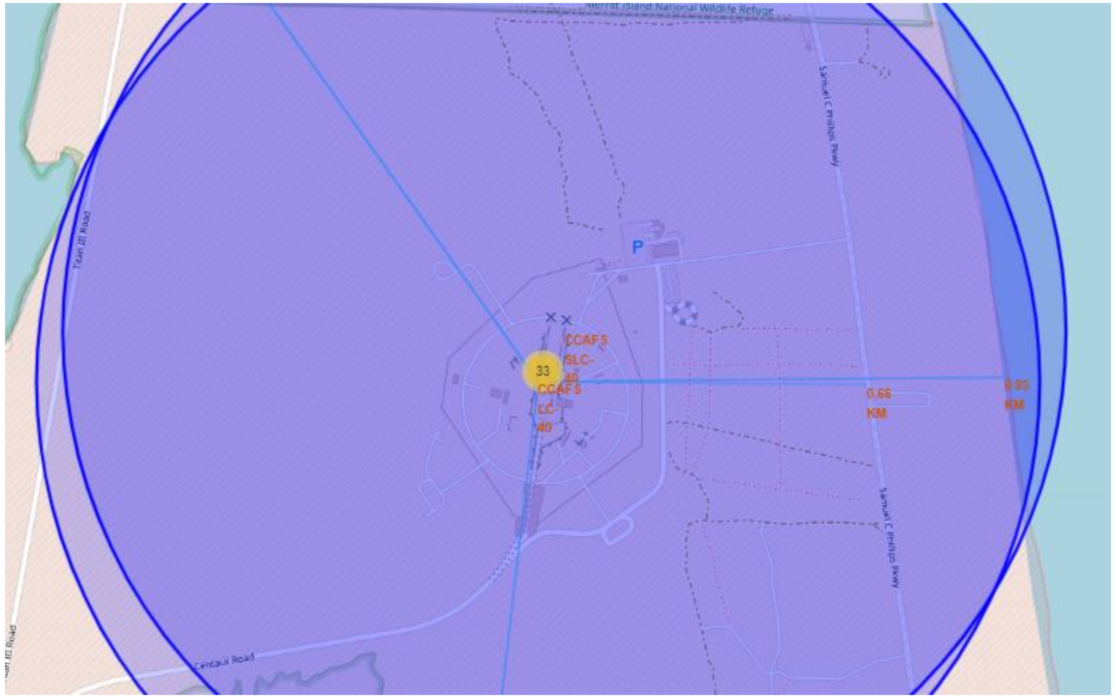
```
features_one_hot.to_csv('dataset_part\_3.csv', index=False)
```

EDA with SQL

- Displaying the names of the unique launch sites in the space mission
- Displaying records where launch sites begin with the string 'CCA,
- Displaying average payload mass carried by booster version F9 v1.1
- Listing the date when the first successful landing outcome in ground pad was achieved
- Listing the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
- Listing the total number of successful and failure mission outcomes
- Listing the names of the booster_versions which have carried the maximum payload mass. Using a subquery.
- Listing the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
- Ranking 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 URL of your completed EDA with SQL notebook

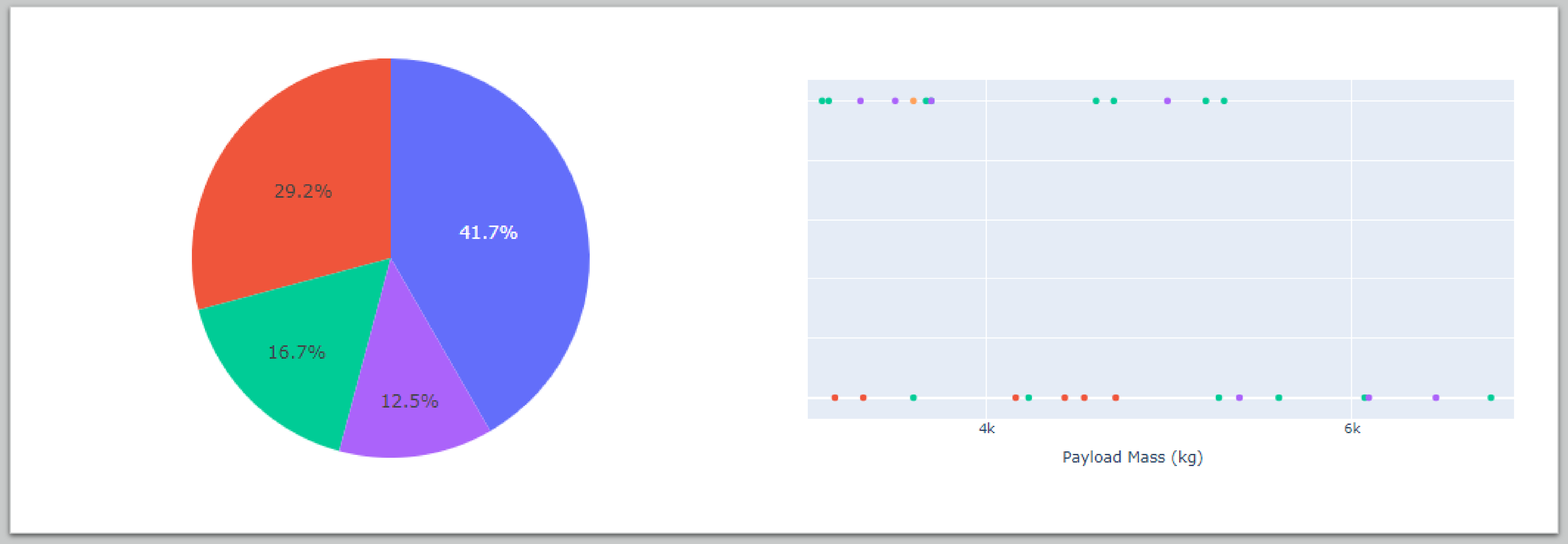
Build an Interactive Map with Folium

- We created markers of NASA and every launch site in our dataset, pop up labels indicating succeeded and failed launches of red and green color
- We created also markers for specific objects like railway, highway and city and drew lines indicating the distance from specific launch site
- We have added those objects to visualize the launch sites and try to obtain additional information for analysis such as distances from the Ocean, railways, cities etc. to get additional information
- GitHub URL of your completed interactive map with Folium map



Build a Dashboard with Plotly Dash

- We used pie charts and scatter plots and added dropdown list of launch sites and payload range slicer for user interaction
- We used those charts to visualize the correlation between launch site and mission success rate
- [GitHub URL of completed Plotly Dash lab]



Predictive Analysis

- Data pre-processing
  - Loading feature engineered data into dataframe
  - Casting data to NumPy series and performing pre-processing using StandardScaler
  - Test-train splitting of data
- Model Building
  - Creating 4 pairs of GridSearch and Model Objects
    - KNN model
    - SVM model
    - Decision Tree model
    - Logistic Regression model
- Model Fitting and Training of every created model
- Model Prediction
- Model Evaluation
  - Calculating the accuracy of every model
  - Creating Confusion Matrices for every model
- Finding the best classification model

Results

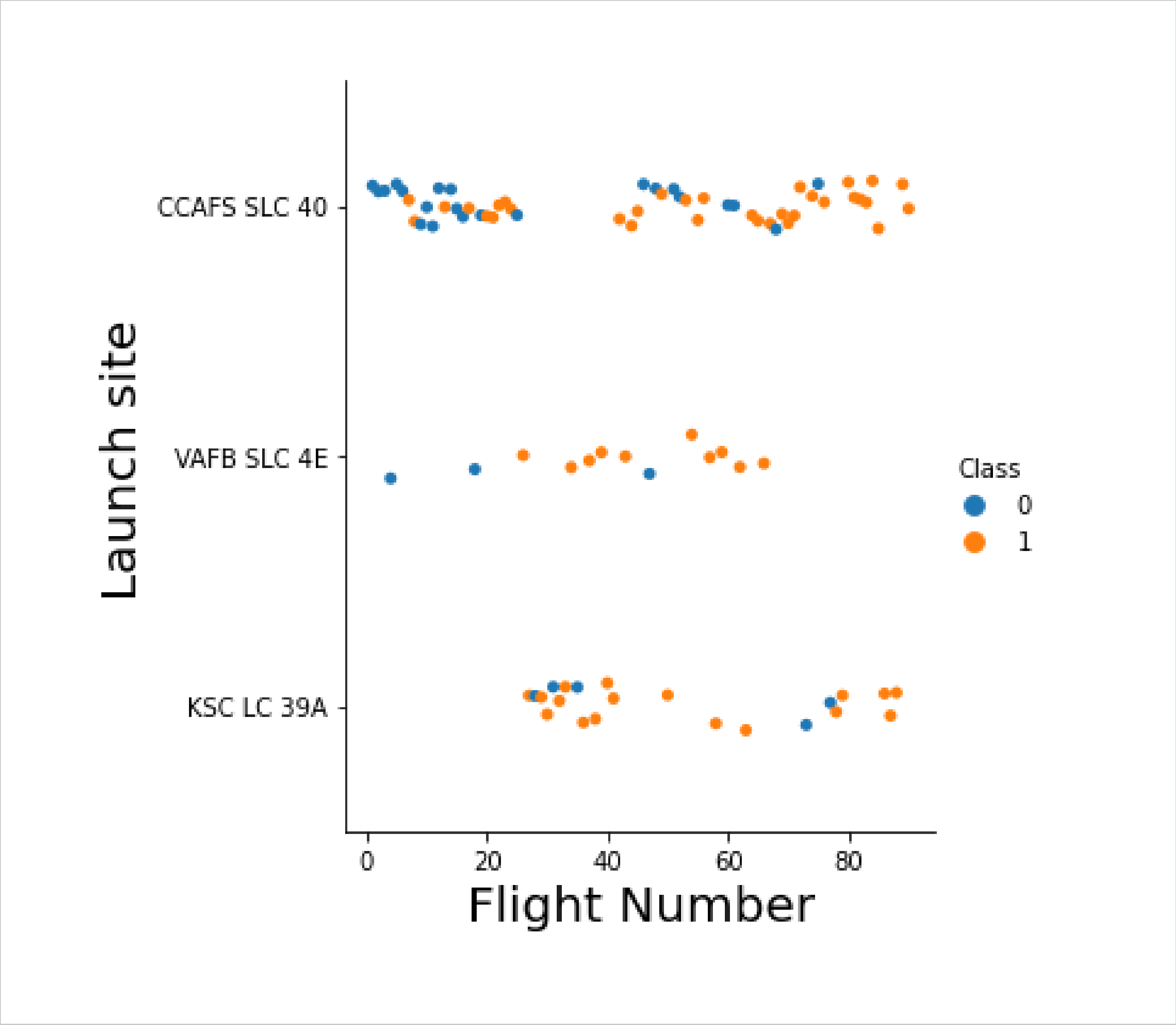
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

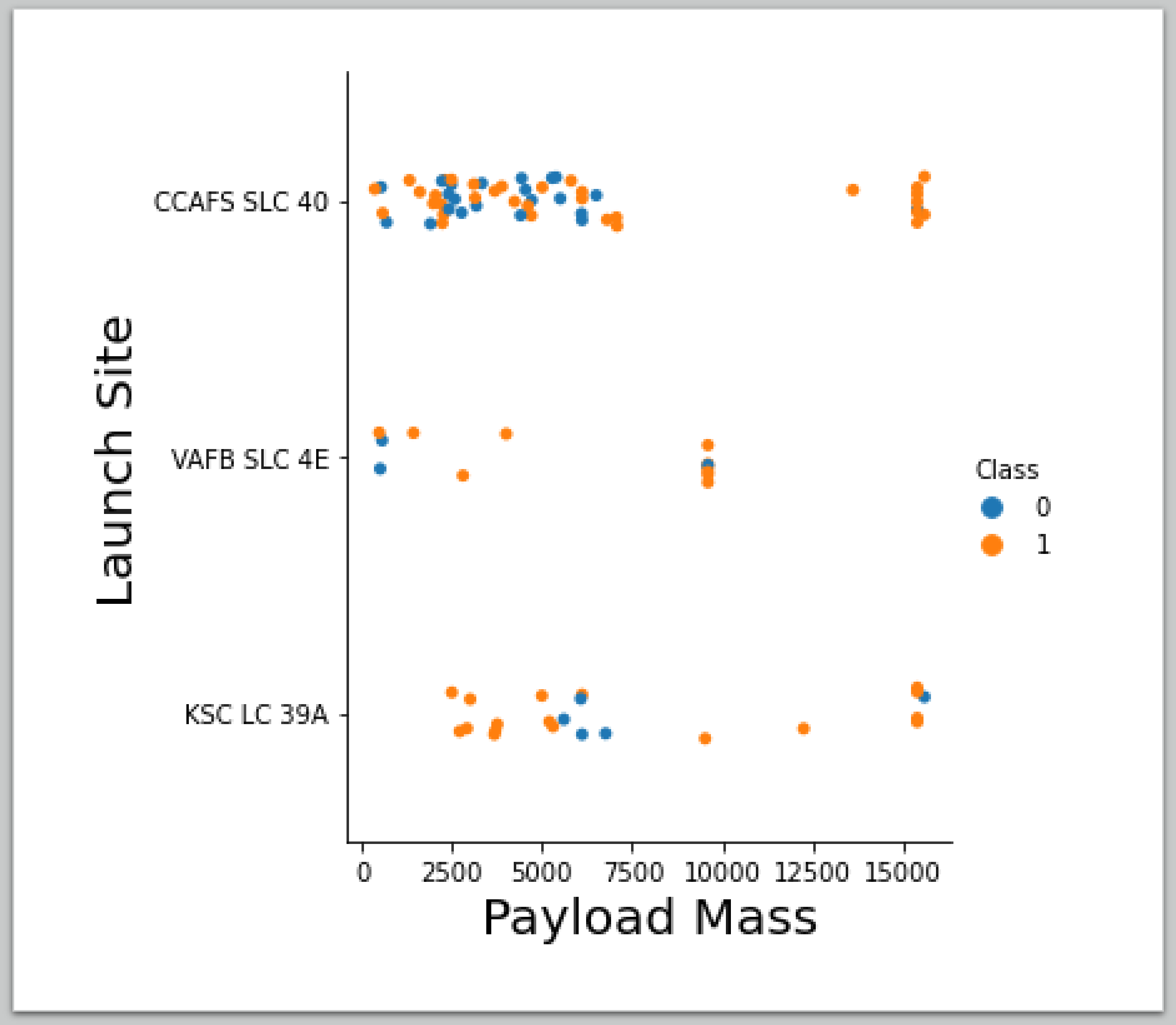
Flight Number vs. Launch Site

- Looking at this scatter plot of Flight Number vs. Launch Site we can notice that success rate for the each launch site was increasing while the flight number increasing too.



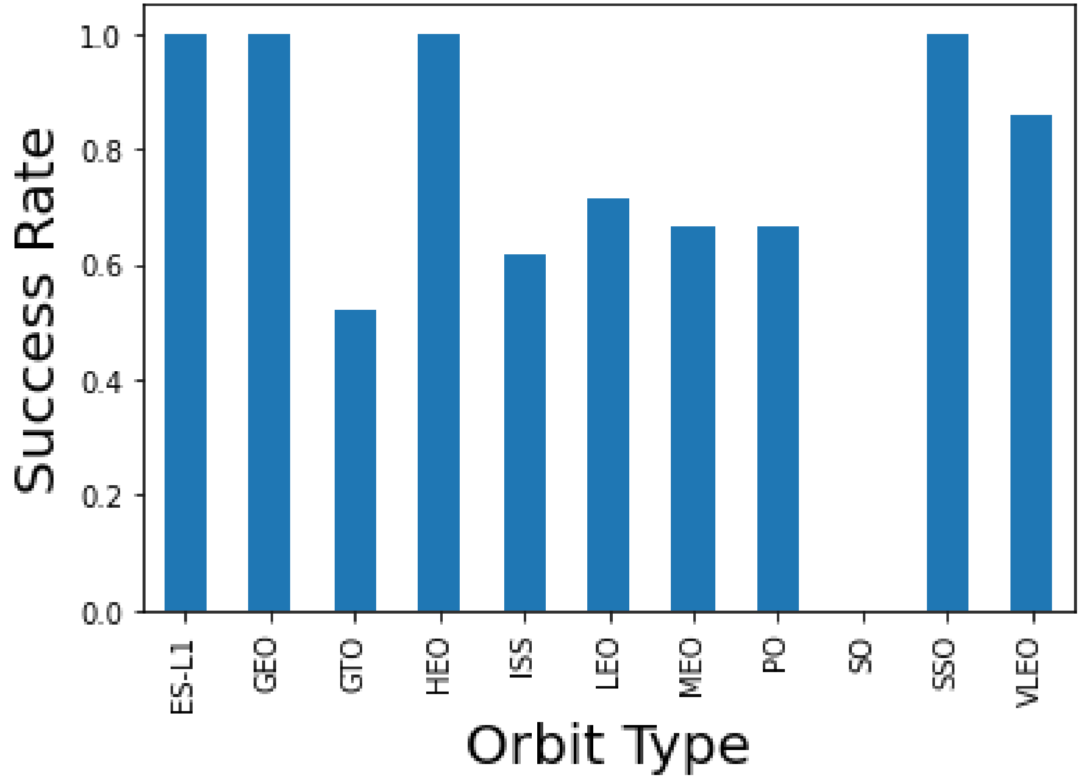
Payload vs. Launch Site

- By looking at this scatter plot of Payload vs. Launch Site we can see that
- Success launch for CCAFS SLC40 was only if the payload mass was above 12500. If payload mass was lower than 7500, the outcome was unsure.
- KSC LC 39A was pretty successful if payload was out of 5000-7500 range.
- VAFB SLC 4E doesn't have much data so we cannot say more that it didn't do good while payload was small.



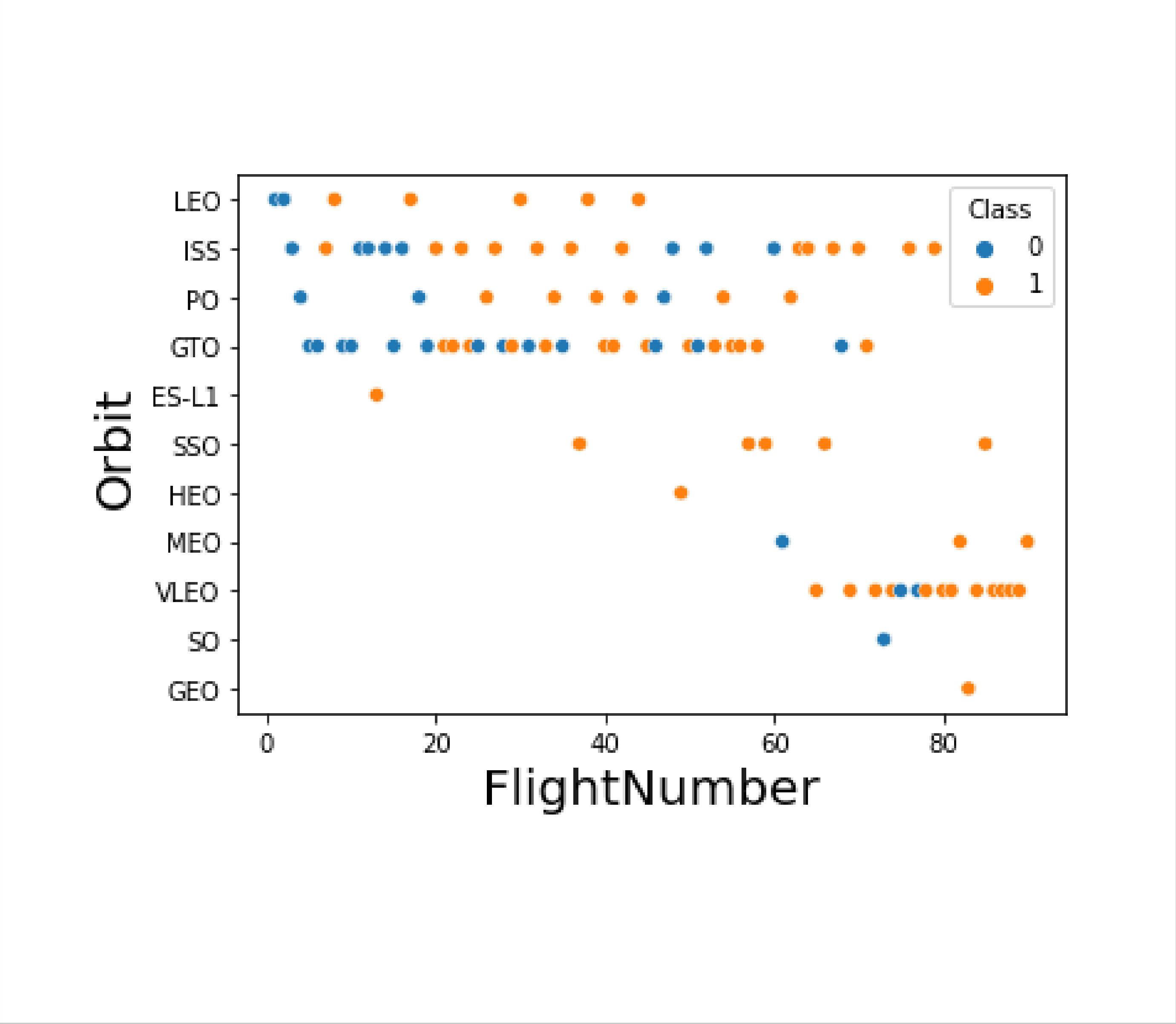
Success Rate vs. Orbit Type

- Looking at this bar chart for the success rate of each orbit type we can notice right away that 4 orbits – ES-L1, GEO, HEO, SSO have 100% success rate and one 90%.
- Orbits such as GTO, ISS, LEO, MEO, PO and SO achieved 70% down to 0% for the success rate thus we can use this info in planning future launches.



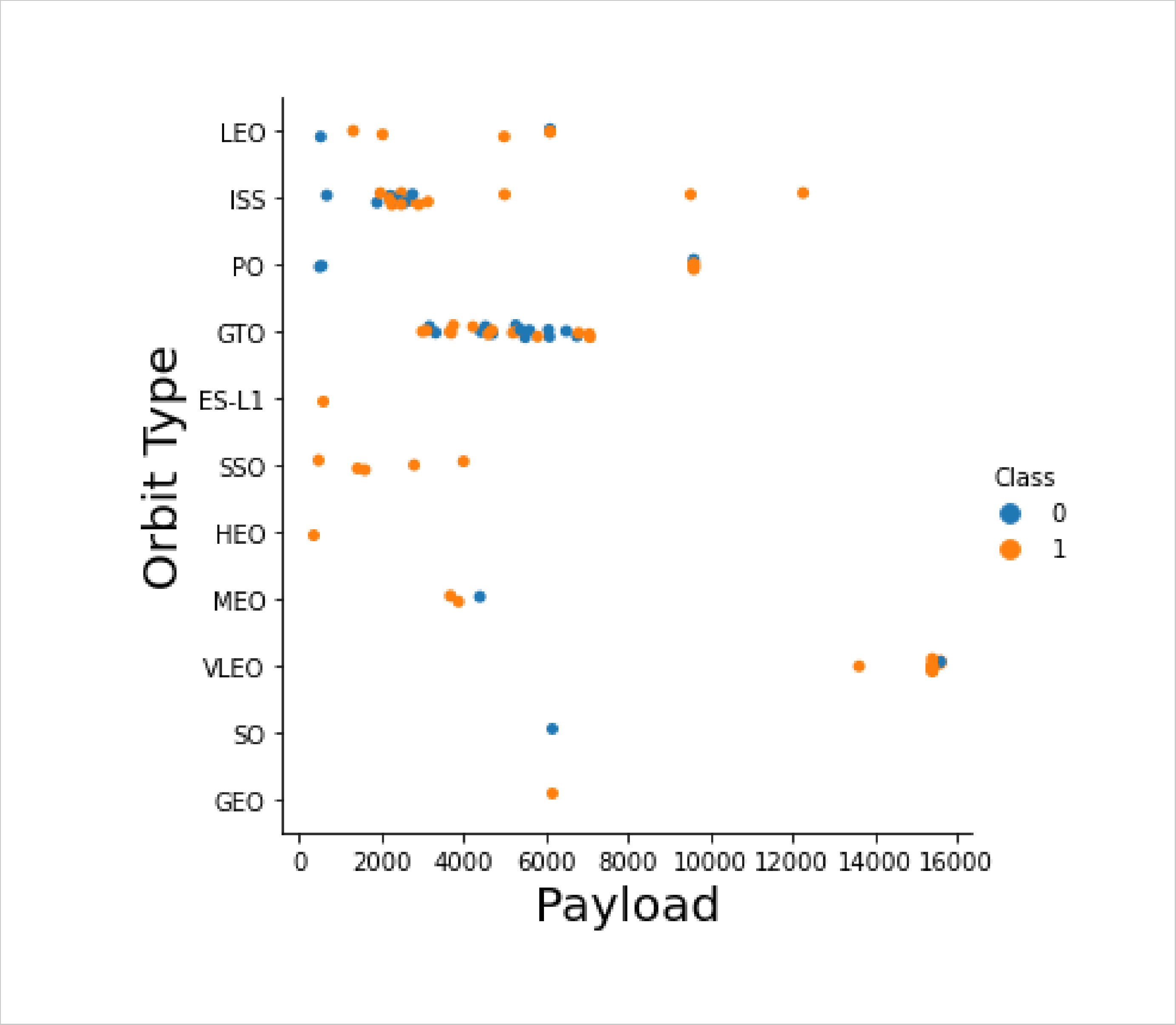
Flight Number vs. Orbit Type

- By looking at this scatter point of Flight number vs. Orbit type we can see that for some orbits like ISS, LEO or VLEO flight number has an importance, probably due to gained knowledge in those areas
- GTO although have many failed launches has an increasing rate of success while flight number increases



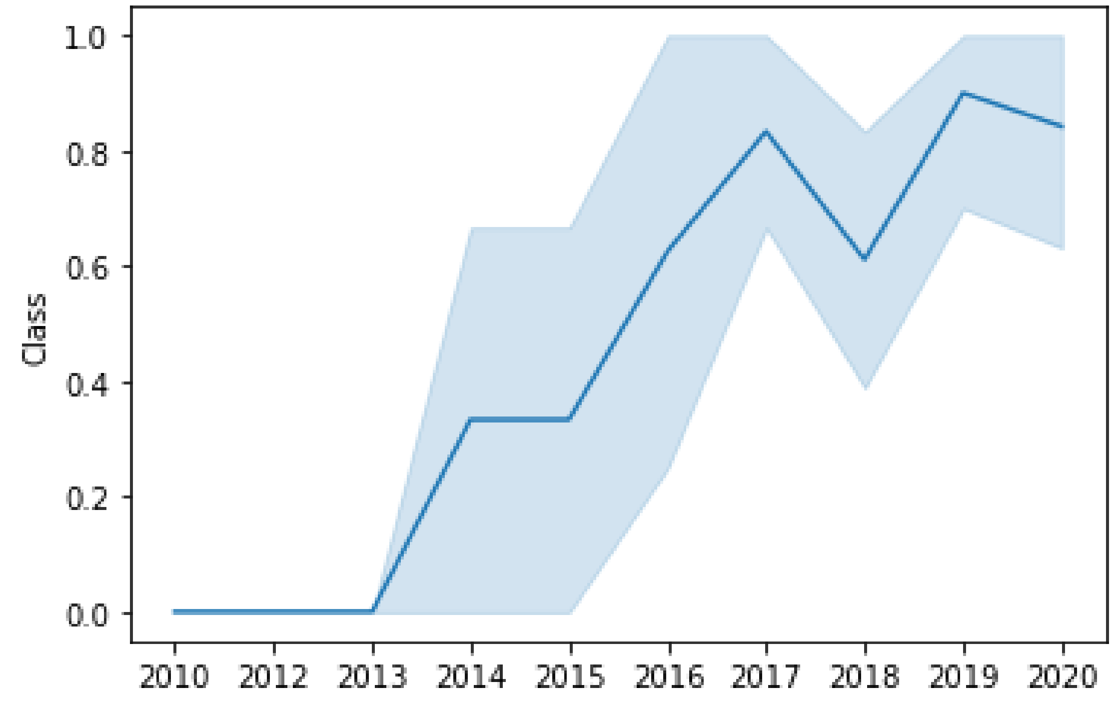
Payload vs. Orbit Type

- By looking at the scatter point of payload vs. orbit type we can notice that GTO orbit doesn't imply that correlation between those two features exists.
- Rest of the seen data is not enough to
- Show the screenshot of the scatter plot with explanations



Launch Success Yearly Trend

- By looking at the line chart of yearly average success rate we can easily notice that there is an increasing success trend. Launch success rate significantly increased over the last 10 years
- We can notice that there was a 20% decrease of success rate in 2018 but next year it got better



All Lanuch Site Names

EXPLANATION

Using UNIQUE we can ensure that all the values in a column are different. We got 4 rows with 4 different launch sites.

SQL QUERY

```
%%sql
SELECT UNIQUE(Launch_site) from spacex
```

SQL RESULT

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

Launch Site Names Begin with 'CCA'

EXPLANATION

Using SELECT statement with condition WHERE followed by LIKE that strarts with a substring 'CCA' and limiting it to 5 records using LIMIT 5. We got differents payload only for Lauch Site starting with CCA like CCAFS SLC-40.

SQL QUERY

```
%sql SELECT * FROM spacex where Launch_site like 'CCA%' LIMIT 5
```

SQL RESULT

DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0
2012-05-22	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525
2012-10-08	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677

Total payload mass

EXPLANATION

Using aggregate function SUM in SELECT statement we calculated Total Payload Mass carried by boosters launched by NASA (CRS). It returned the result of 45596 kg of total mass.

SQL QUERY

```
%sql SELECT SUM(payload_mass__kg_) FROM spacex where customer like 'NASA (CRS)'
```

SQL RESULT

1
45596

Average Payload Mass by F9 1.1V

EXPLANATION

Using aggregate function avg to get an average value of the column with payload mass but filtering the output only for booster F9 v1.1 by using WHERE clause.

SQL QUERY

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

SQL RESULT

1
2534

First Successful Ground Landing Date

EXPLANATION

The function MIN returns the smallest value from the column of filtered results only for succeed landing outcome. We got information that the first succeed landing had place on 2018-07-22.

SQL QUERY

```
%sql select MIN(DATE) from spacex where landing__outcome like 'Success'
```

SQL RESULT

1
2018-07-22

Successful Drone Ship Landing with Payload between 4000 and 6000

EXPLANATION

We limited our result by setting two condition – mission outcome had to be a 'Success' and payload mass of the booster had to be between 4000 kg and 6000kg. In a result we got a list of 22 payloads.

SQL QUERY

```
%sql select payload from spacex where mission_outcome like 'Success'
    and (payload_mass__kg_ > 4000 and payload_mass__kg_ < 6000)
```

SQL RESULT

payload
AsiaSat 8
AsiaSat 6
ABS-3A Eutelsat 115 West B
Turkmen 52 / MonacoSAT
SES-9
JCSAT-14
JCSAT-16
EchoStar 23
EchoStar 23
SES-10
NROL-76
Boeing X-37B OTV-5
SES-11 / EchoStar 105
GovSat-1 / SES-16
SES-12
Merah Putih
Merah Putih
Es hail 2
GPS III-01
Nusantara Satu, Beresheet Moon lander, S5
RADARSAT Constellation, SpaceX CRS-18
GPS III-03, ANASIS-II
ANASIS-II, Starlink 9 v1.0
GPS III-04 , Crew-1

Total Number of Successful and Failure Mission Outcomes

EXPLANATION

We used aggregate function COUNT to count total number of succeeded and failed missions and we grouped the results by their affiliation. We got a result of 99 mission with 'Success' outcome, 1 failure in flight and one extra success but data about payload status is unclear thus difference in labeling the mission outcome.

SQL QUERY

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

SQL RESULT

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

Boosters Carried Maximum Payload

EXPLANATION

We selected columns which we were interested in from a subquery where we used MAX function to get records of boosters only with maximum payload which as we can see is 15600 kg. In a result we got a list of 12 records of payloads with maximum payload mass.

SQL QUERY

```
%sql select payload, payload_mass__kg_ from spacex where payload_mass__kg_ =
(select MAX(payload_mass__kg_) from spacex)
```

SQL RESULT

payload	payload_mass__kg_
Starlink 1 v1.0, SpaceX CRS-19	15600
Starlink 2 v1.0, Crew Dragon in-flight abort test	15600
Starlink 3 v1.0, Starlink 4 v1.0	15600
Starlink 4 v1.0, SpaceX CRS-20	15600
Starlink 5 v1.0, Starlink 6 v1.0	15600
Starlink 6 v1.0, Crew Dragon Demo-2	15600
Starlink 7 v1.0, Starlink 8 v1.0	15600
Starlink 11 v1.0, Starlink 12 v1.0	15600
Starlink 12 v1.0, Starlink 13 v1.0	15600
Starlink 13 v1.0, Starlink 14 v1.0	15600
Starlink 14 v1.0, GPS III-04	15600
Starlink 15 v1.0, SpaceX CRS-21	15600

2015 Launch Records

EXPLANATION

We filtered our dataset to records of mission with failed landing outcome and limiting it only to the 2015 year by using WHERE clause.

SQL QUERY

```
%sql select landing__outcome, booster_version, launch_site, DATE

from spacex where landing__outcome like '%Failure%' and DATE like '%2015%'
```

SQL RESULT

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

Rank leading outcomes between 2010-06-04 and 2017-03-20

EXPLANATION

We filtered our dataset to records of mission to Succeeded landing outcome using ground pad and Failed landing outcome using drone ship in Date range between 2010-06-04 and 2017-03-20. Then we used aggregate function COUNT to count the sum of such events and grouped them by their affiliation and returned them in descending order.

SQL QUERY

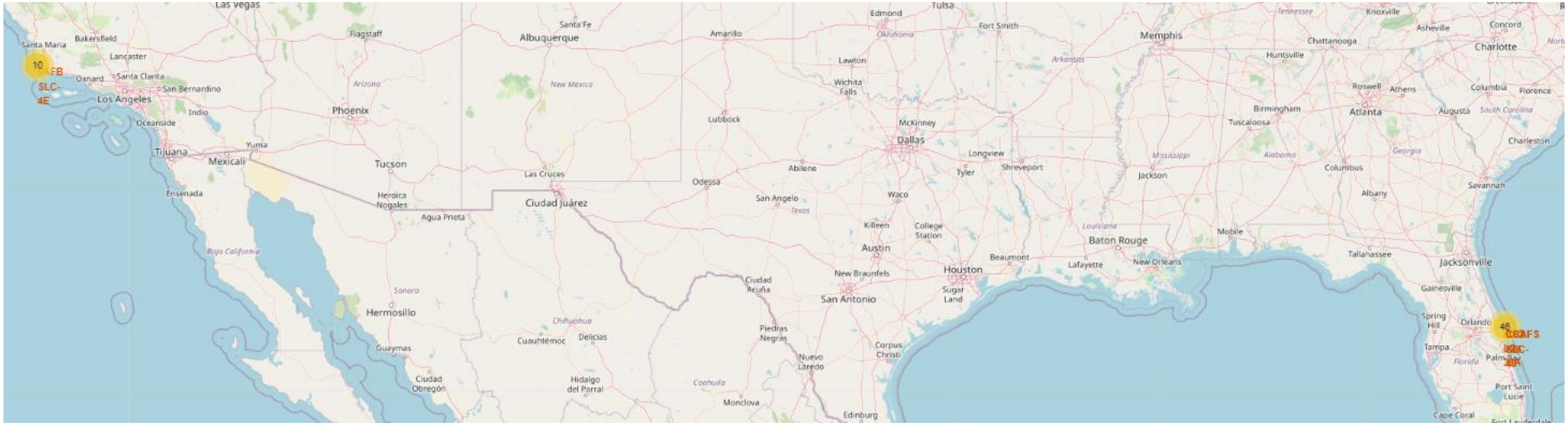
```
%%sql
select count(*) as counter, landing__outcome from spacex where landing__outcome like 'Success (ground pad)'
or landing__outcome like 'Failure (drone ship)' and date between '2010-06-04' and '2017-03-20' group by landing__outcome
order by counter desc
```

SQL RESULT

counter	landing__outcome
9	Success (ground pad)
5	Failure (drone ship)

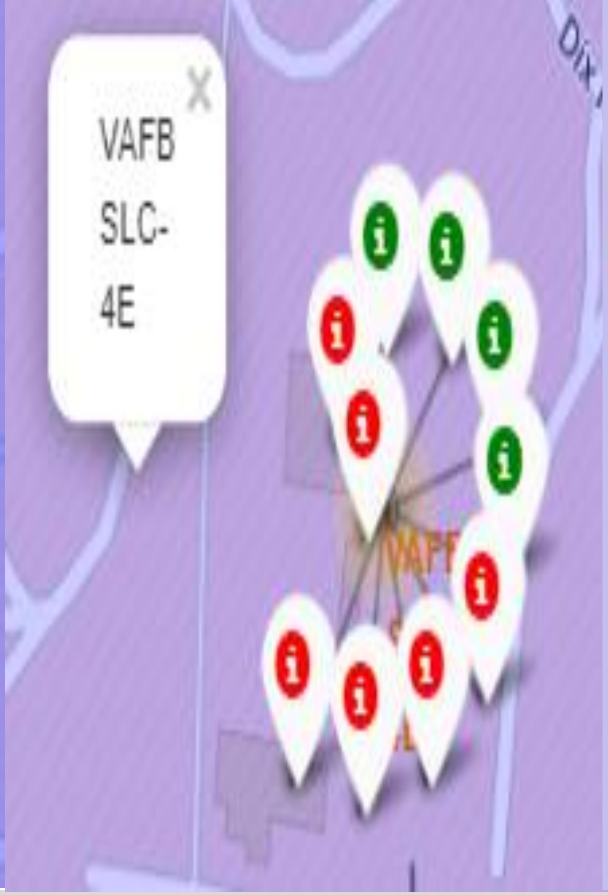
Section 3

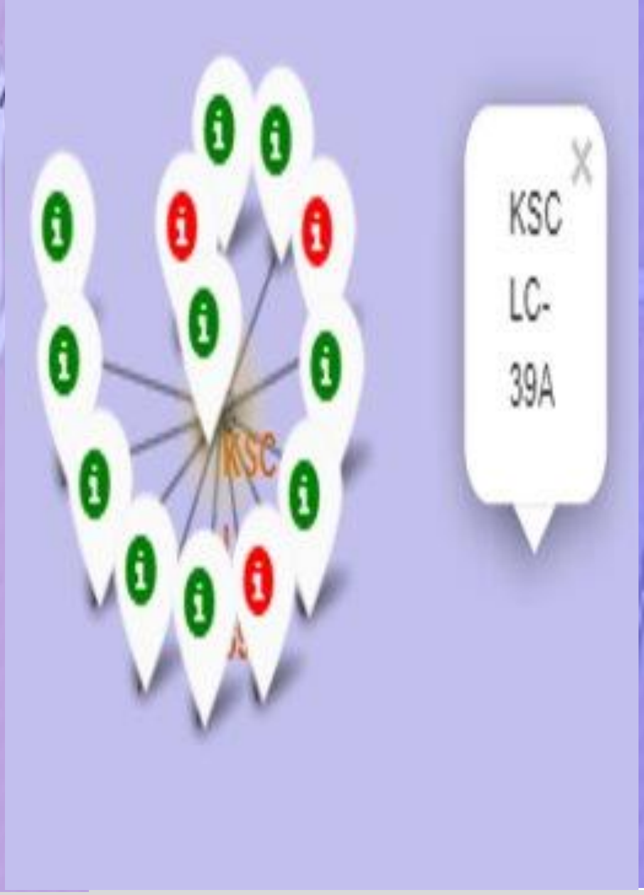
Launch Sites Proximities Analysis



Folium Map – ground stations

We can see on the global folium map that SpaceX has its ground stations only in the USA. Additional information we can gain from the folium map is that the Ground Stations are located on the coast of the county.

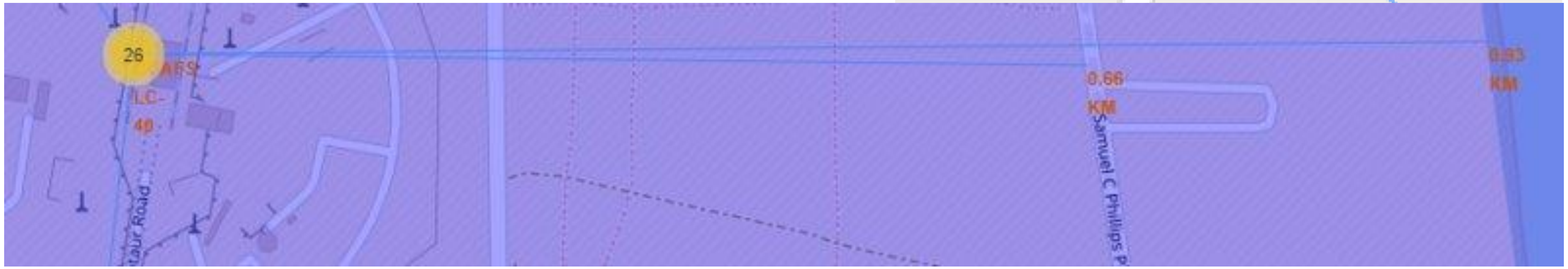




Folium Map – color labeled markers

We applied on the folium map a feature that allows seeing color-labeled markers by clicking on fic ground stations. Each marker indicates one mission with one of two outcomes – **green** indicates succeeded mission and **red** indicates failed mission.

On the screenshots we can see the distance from CCAFS LC-40 to Melbourne (51.08 km), Coast (0.93 km), highway (0.66 km) and railway (1.33 km). This information can provide us additional, important insight in the final output.

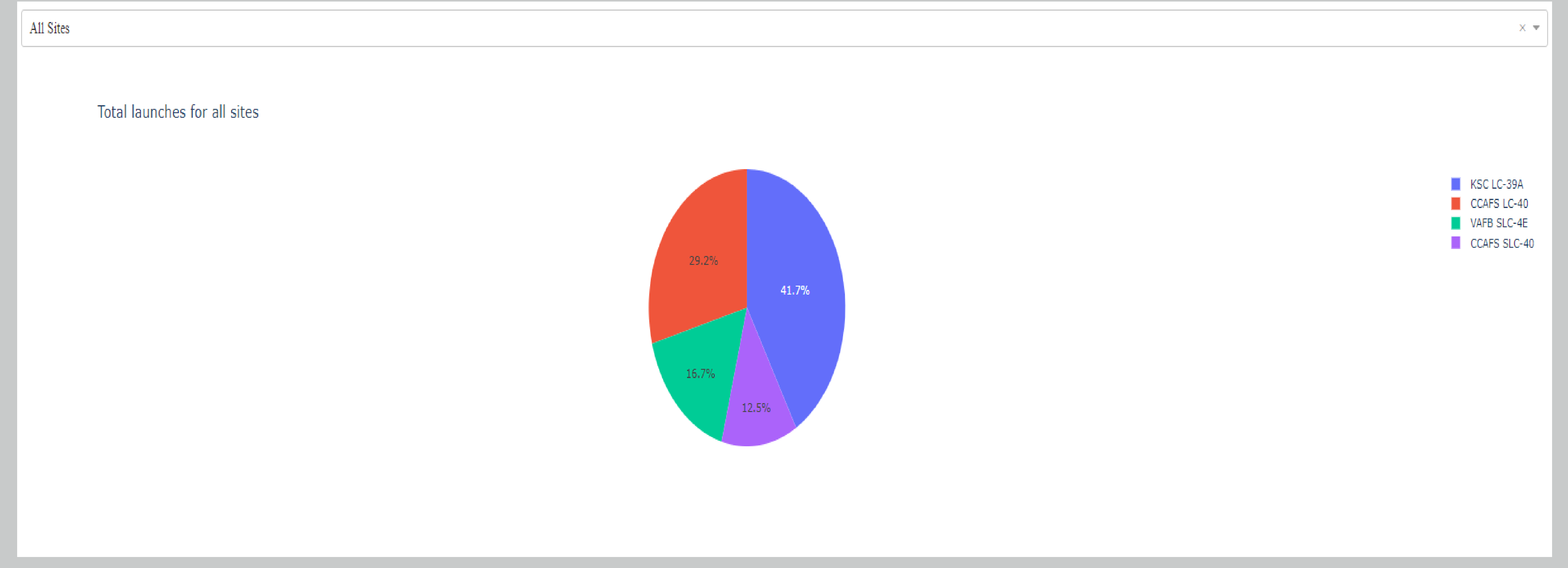


Folium Map – proximities measurements

We applied additional information on the folium map a feature that allows seeing launch site to its proximities such as railway, highway, coastline, with distance calculated and displayed.

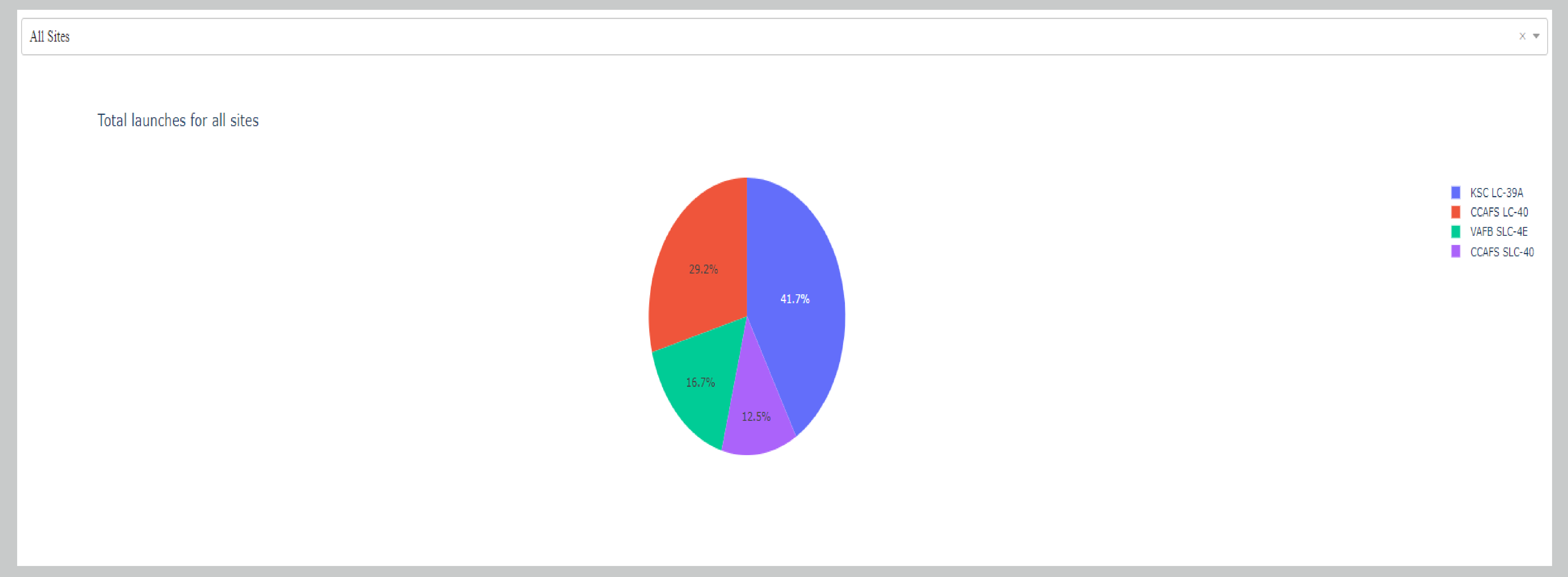
Section 4

Build a Dashboard with Plotly Dash



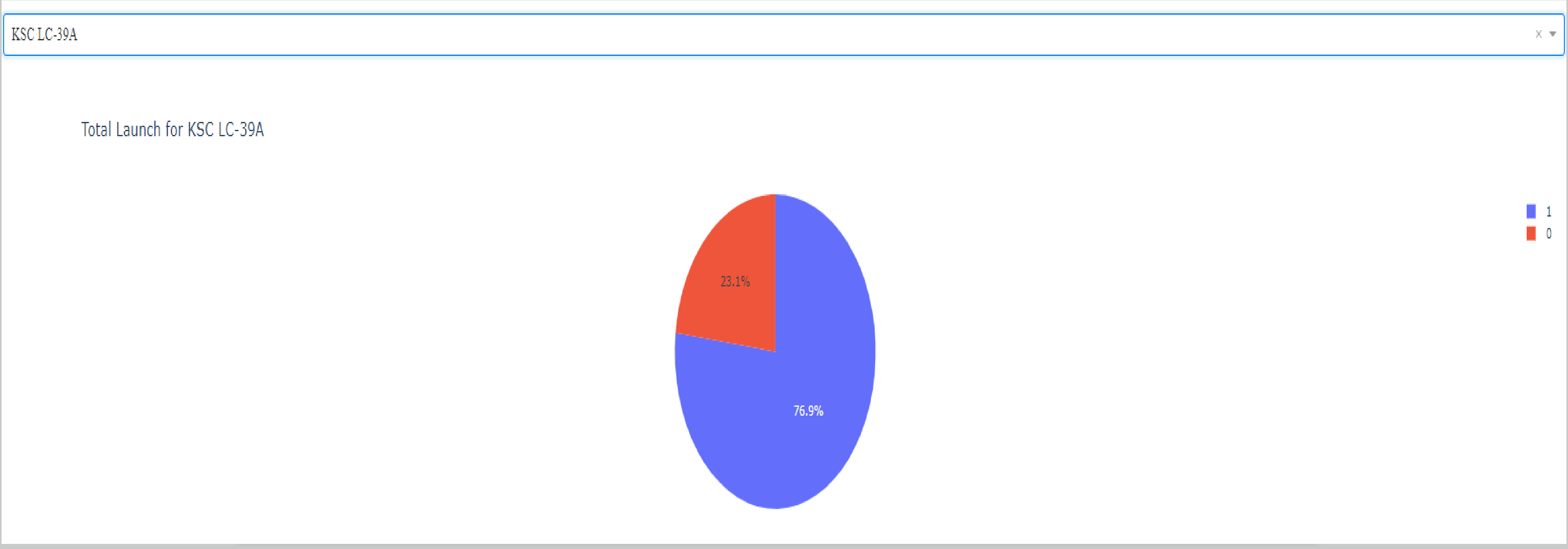
Dashboard of Total launches for all sites

The screenshot represents an interactive dashboard created using Plotly Dash library. This specific screenshot is displaying total launches for all sites. As we can easily notice thanks to the dashboard, KSC LC-39A has the highest of launches – 41.7%. Second top result is achieved by CCAFS LC-40 with 29.2%, third place is achieved by VAFB SLC-4E with 16.7% and at the fourth place is CCAFS SLC-40 with 12.5%.



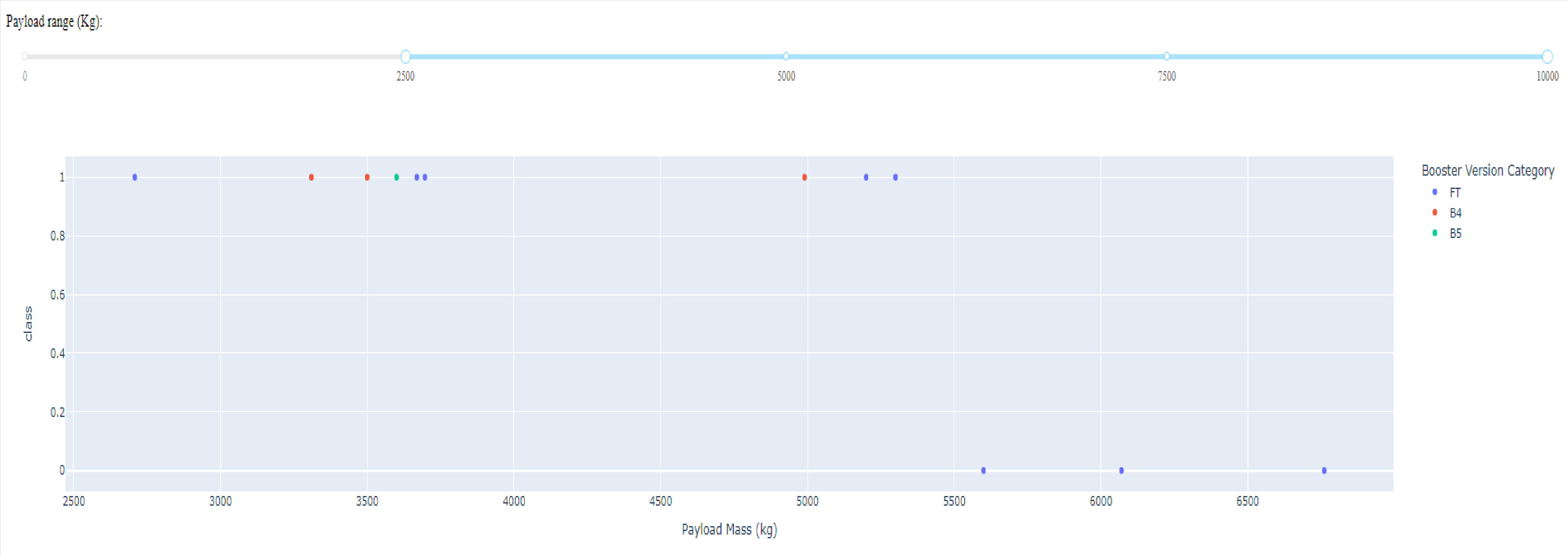
Dashboard of Total launches for all sites

The screenshot represents an interactive dashboard created using Plotly Dash library. This specific screenshot is displaying total launches for all sites. As we can easily notice thanks to the dashboard, KSC LC-39A has the highest of launches – 41.7%. Second top result is achieved by CCAFS LC-40 with 29.2%, third place is achieved by VAFB SLC-4E with 16.7% and at the fourth place is CCAFS SLC-40 with 12.5%.



Dashboard of Total launch for KSC LC-39A

The screenshot represents an interactive dashboard created using Plotly Dash library. This specific screenshot is displaying total launches for KSC LC-39A site that has the highest rate of total launches for all sites. The legend on the right indicates success mission outcome with number 1 and failed mission outcome with number 0. As we can see, 76.9% of total launch for this site has been successful while 23.1% has been failed.



Dashboard of Total launch for KSC LC-39A

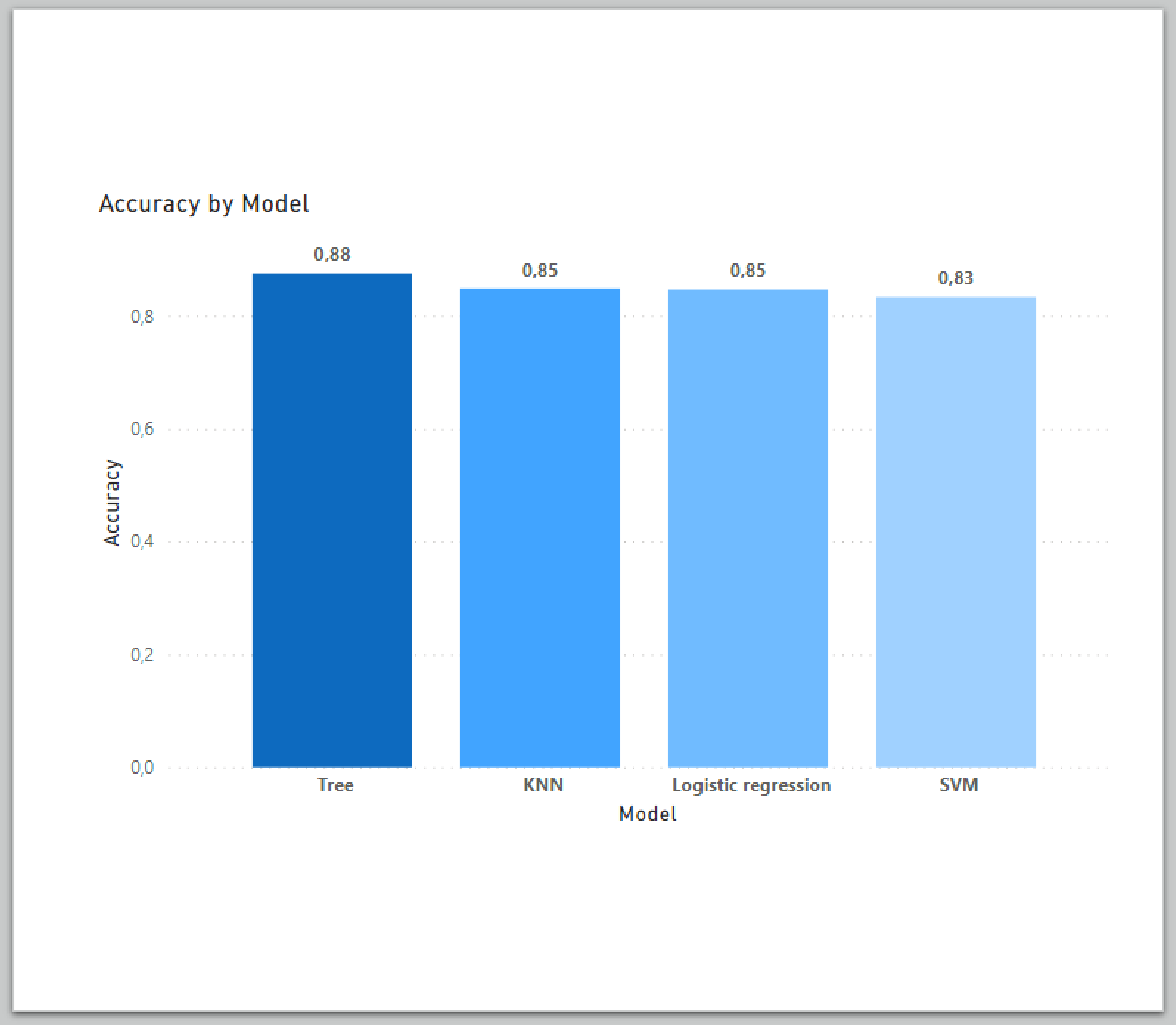
The screenshot represents an interactive dashboard created using Plotly Dash library. This specific screenshot is displaying a scatter plot for all sites, with different payload selected in the range slider. Dashboard allows users to select range of the payload range based on which scatter plots displays different output. In case of presented screenshot above, we can see that selected payload range is between 2500kg and 10000kg. In this range we see only three Booster Version Category – FT, B4 and B5. Class feature indicates failure and success mission outcome. We can see that only FT Booster Version with payload above 5500kg failed all mission, every booster with a payload below 5500 landed successfully.

Section 5

Predictive Analysis (Classification)

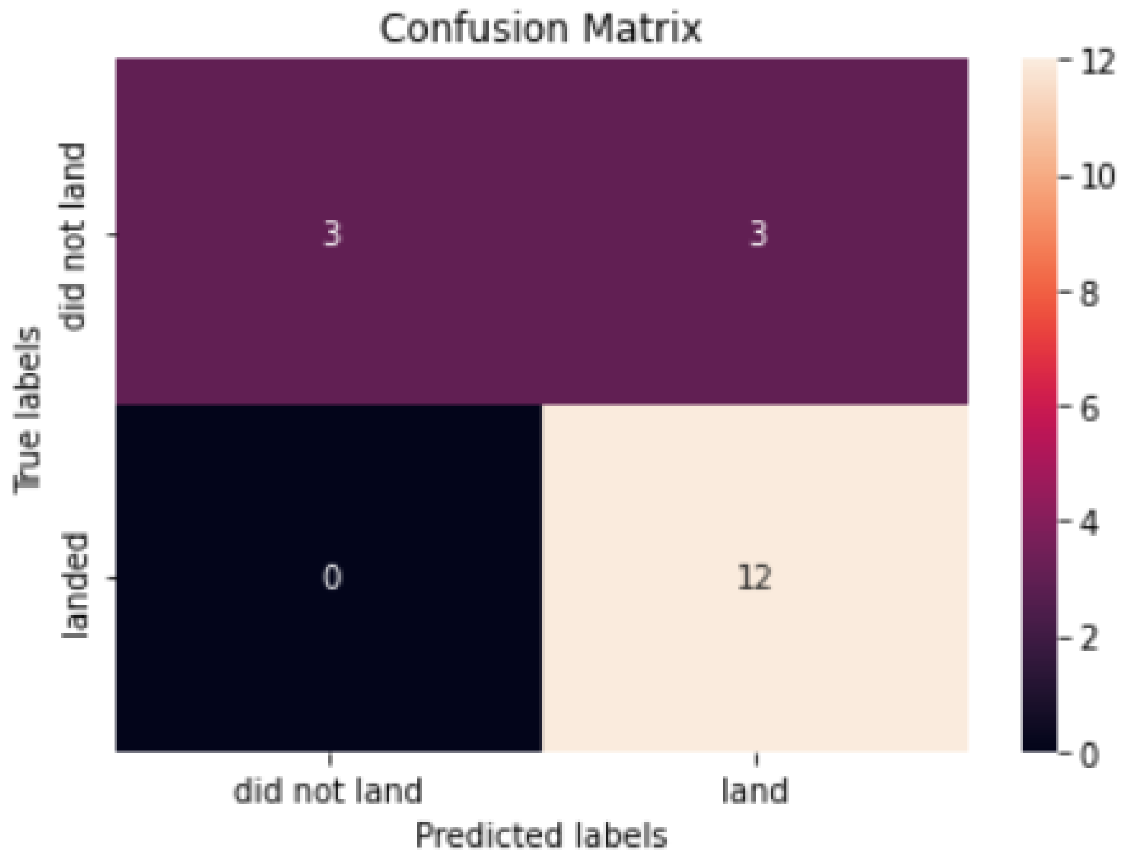
Classification Accuracy

- We have built four predictive models using different algorithms.
- The worst performance was achieved by SVM with accuracy of 83%. KNN and Logistic regression achieved the same accuracy 85%.
- Tree model has the highest classification accuracy of 88%.



Confusion Matrix

- Confusion Matrix on the right shows a matrix of true labels vs predicted labels achieved by best classification model which is Decision Tree
- We can see that out of 12 landing missions with a succeeded outcome, our model correctly predicted all landings, so called True Positives (TP)
- 3 missions out of 6 missions with a failed outcome (did not land) were classified as a successful landing, so called False Positives (FP)
- 3 mission out of 6 mission with a failed outcome (did not land) were correctly classified as a failed landing (TN)
- There were no landings with a succeeded landing classified by our model as a failure, so called False Negative (FN) which means good



Hyperparameters

```
tuned hpyerparameters :(best parameters)  {'criterion': 'gini', 'max_depth': 12, 'max_features': 'auto', 'min_samples_leaf': 4, 'min_samples_split': 2,
'splitter': 'random'}
accuracy : 0.875
```

Conclusions

- Increasing Flight Number has significant impact on mission outcome. The more experience launch site is, the higher chance of successful landing.
- The most successful launch site is KSC LC-39A (42% of total SpaceX launches) but its success rate was decreasing with payloads above 5500kg
- Orbits such as ES-L1, GEO, HEO and SSO achieved 100% success rate thus it might be good useful information in planning future launches ES-L1, GEO, HEO, SSO
- SpaceX has been more successful over last 10 years and are getting extremely close to hit the required target
- Decision Tree Clasifier achievied the best result of 88% among all trained models therefore is most suitable and effiecient for the dataset provided

Appendix

- Github repository contained all jupyter notebook and python files
- PowerBI
- GoogleColab
- IBM Cloud PAK

Thank you!