SpaceX Project

Gavriushkin Egor

01.12.2022

IBM Developer

SKILLS NETWORK

Outline

- Executive Summary
- Introduction
- Methodology
- Results
- Discussion
- Conclusion

EXECUTIVE SUMMARY

- Summary of methodologies
  - Data collection
  - Data wrangling
  - Features Selection
  - EDA with data visualization
  - EDA with SQL
  - Interactive map with Folium
  - Dashboard with Plotly Dash
  - Predictive analysis (Classification)
- Summary of all results
  - EDA results
  - Interactive analytics
  - Predictive analysis

INTRODUCTION

Project background and context

SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; 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 I want to find answers

The project task is to predict if the stage of the SpaceX Falcon 9 rocket will land successfully

METHODOLOGY

- Data collection methodology:
  - SpaceX Rest API
  - Web Scraping from Wikipedia
- Data wrangling
  - One hot encoding fields for Machine Learning , cleaning data from null values and irrelevant data
- Perform exploratory data analysis using visualizations and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Building machine learning algorithms using classification methods such as:
  - Logistic Regression
  - K- Nearest Neighbours
  - Support Vector Machine
  - Decision Trees

Data collection

The SpaceX launches data was collected from SpaceX API and Wikipedia

- SpaceX API was used to get data about launches, including information about the rocket used, payload delivered, launch specifications, landing specifications, and landing outcome.

- Wikipedia Web Scraping with BeautifulSoup parsing library gave us more information about launch dates, launch outcomes and payload mass and etc.

SpaceX API Data collection

Interface example

Using SpaceX API interface via requests lib as follows:

"response = requests.get("https://api.spacexdata.com/v4/rockets/"+str(x)).json()"

This function gave us accsess to Booster version of the launches

Data attained

Using the API data access method we got the following data, which was placed into a dataframe:

- Booster Version
- Launchpad longitude and latitude
- Payload Mass (kg)
- Core data, including:
  - Reuse count
  - Serial number
  - Grid Fins and Legs
  - Flight number
  - Etc.

Github publicly opened code with comments:
https://github.com/gavriushkinegor/IBM-Capstone/blob/main/1%20SpaceX%20API.ipynb

Wikipedia Data collection with BeautifulSoup library

```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"
# use requests.get() method with the provided static_url
# assign the response to a object
response = requests.get(static_url)
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']=[]
column_names = []
first_launch_table.find_all('th')
for name in first_launch_table.find_all('th'):
    if name is not None and len(name) > 0 :
        column_names.append(extract_column_from_header(name))
extracted_row = 0
```

Using requests lib we were able to:

1. Fetch data from Wikipedia tables
2. Find and process desired tables via BS4 lib using final_all()
3. Using python code we managed to fill our dataframe with desired information

Github publicly opened code with comments:
https://github.com/gavriushkinegor/IBM-Capstone/blob/main/2%20SpaceX%20Data%20Collection.ipynb

Data Wrangling
EDA Algorithm

Check null values → Calculate the number of launches on each site → Calculate the number and occurrence of each orbit → Calculate the number and occurrence of mission outcome per orbit type → Create a landing outcome label from Outcome column → Reviewing the data as a SQL database to get new insights
(more on the next slide)

Github publicly opened code with comments:
https://github.com/gavriushkinegor/IBM-Capstone/blob/main/2%20SpaceX%20Data%20Collection.ipynb

SQL Database Findings & Implications

Short overview of the process

- SQLite used to work in Jupyter NB environment

1. Identifying unique launch sites in the space mission:

→ %sql SELECT DISTINCT LAUNCH_SITE   FROM SPACEXTBL

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

→ %sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE CUSTOMER = 'NASA (CRS)'

3. Rank the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017

→
```
%%sql SELECT "DATE", COUNT("LANDING _OUTCOME") AS Successfull_outcomes_count FROM SPACEXTBL
 WHERE substr(Date,7,4) || substr(Date,4,2) || substr(Date,1,2) between '20100604'
and '20170320'AND "LANDING _OUTCOME" LIKE '%Success%'
GROUP BY "DATE"
ORDER BY COUNT("LANDING _OUTCOME") DESC
```

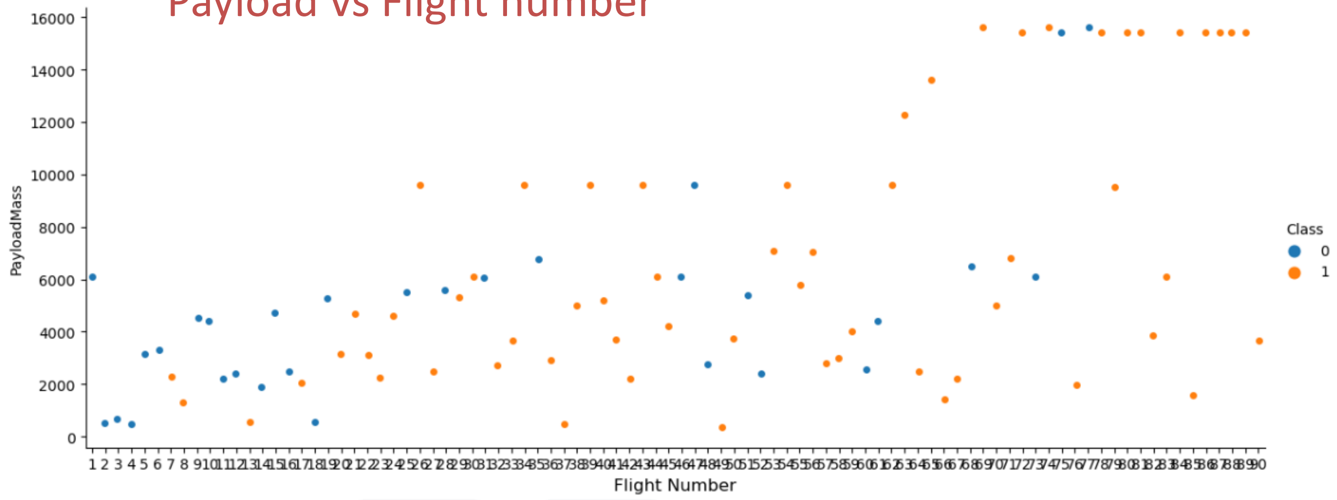
Github publicly opened code with comments:
https://github.com/gavriushkinegor/IBM-Capstone/blob/main/4%20jupyter-labs-eda-sql-coursera_sqllite.ipynb

EDA With Data Visualization

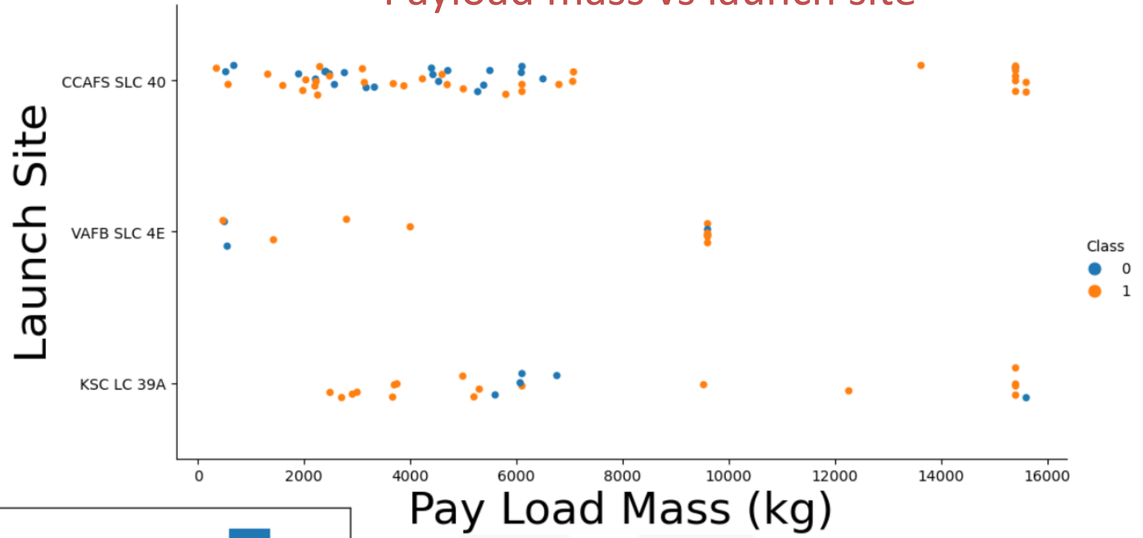
Launch outcome class 0 (blue) – Failure

Launch outcome class 1 (orange) - Success

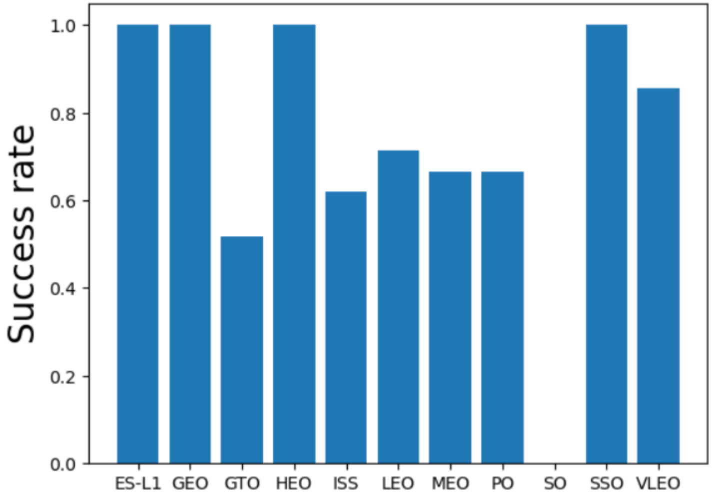
Payload vs Flignt number



Payload mass vs launch site



Success rate evaluated to Orbit



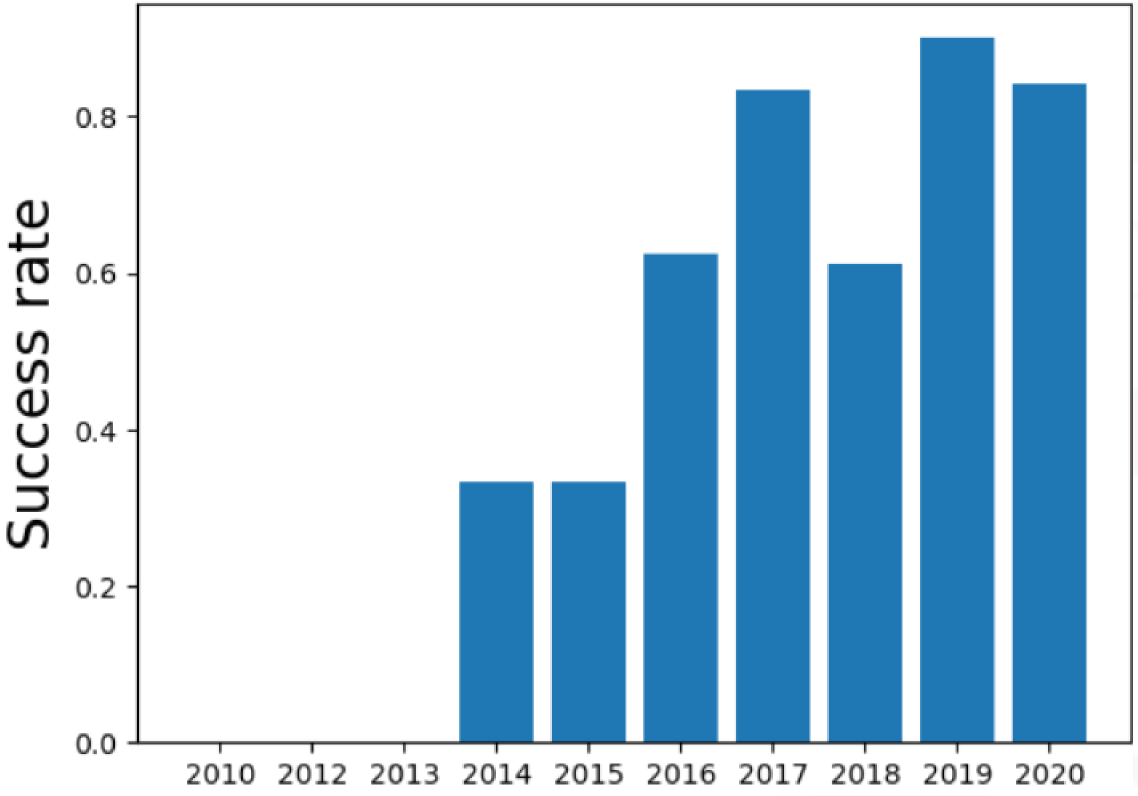
Github publicly opened code with comments:
https://github.com/gavriushkinegor/IBM-Capstone/blob/main/5%20EDA%20with%20Visualization%20Lab.ipynb

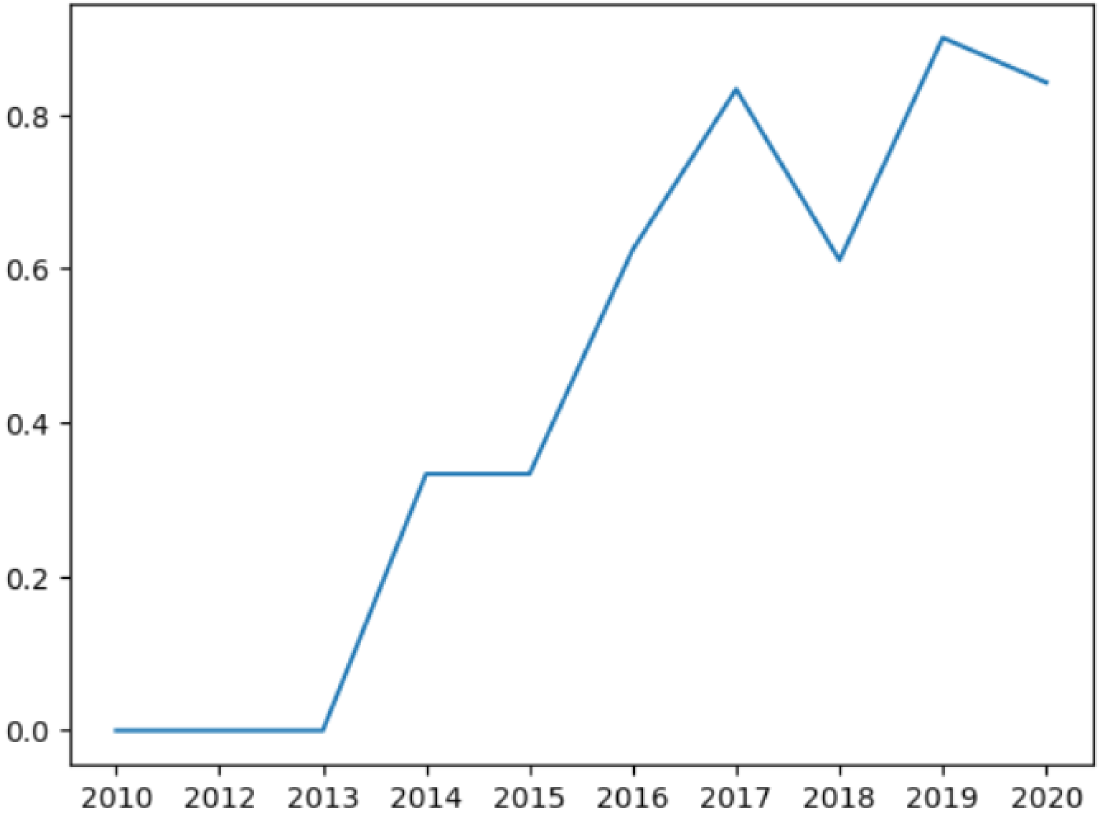
IBM Developer

SKILLS NETWORK

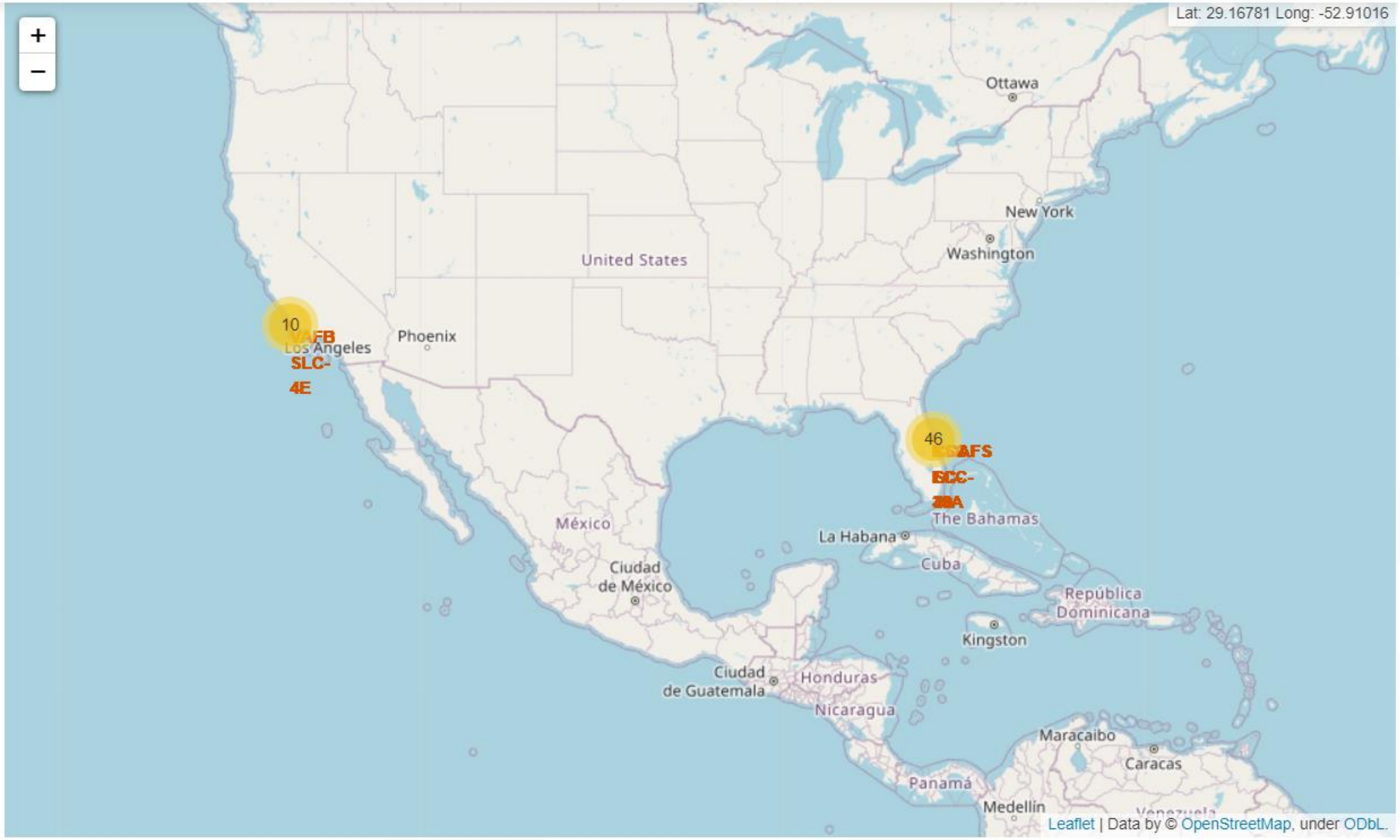
EDA With Data Visualization pt.2

Success rate evaluated to years of launches





Build Interactive Map with Folium



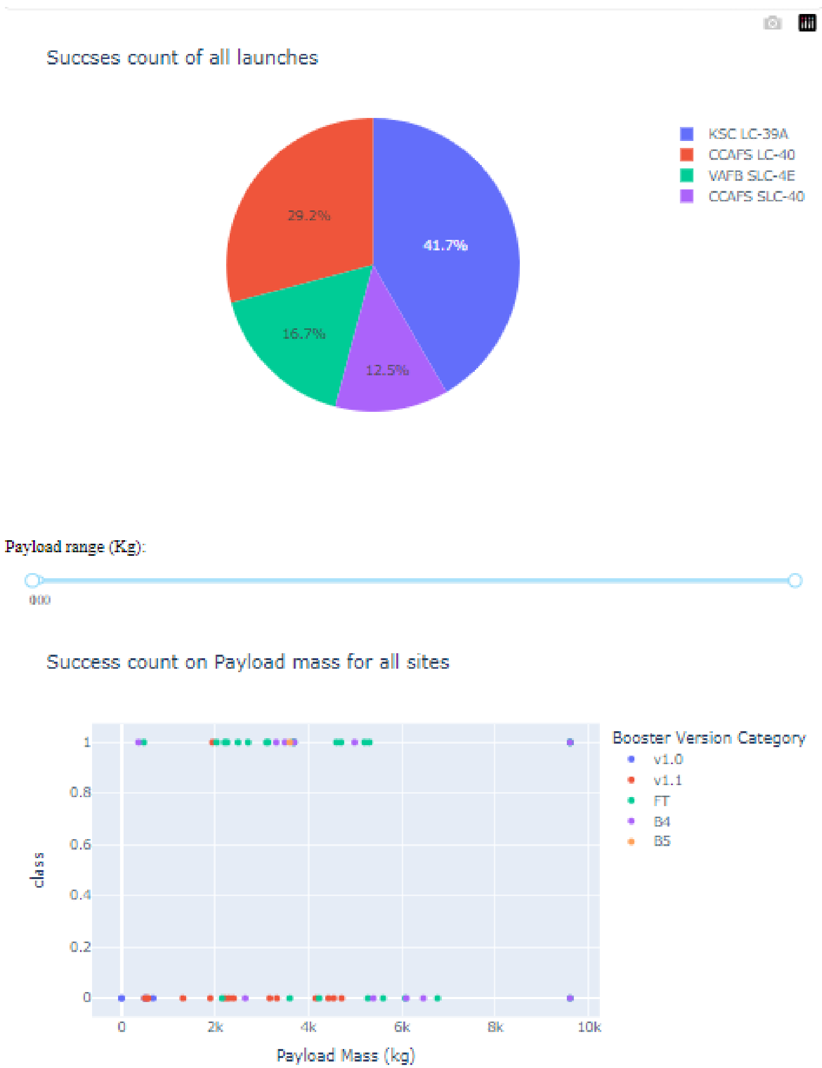
Key findings

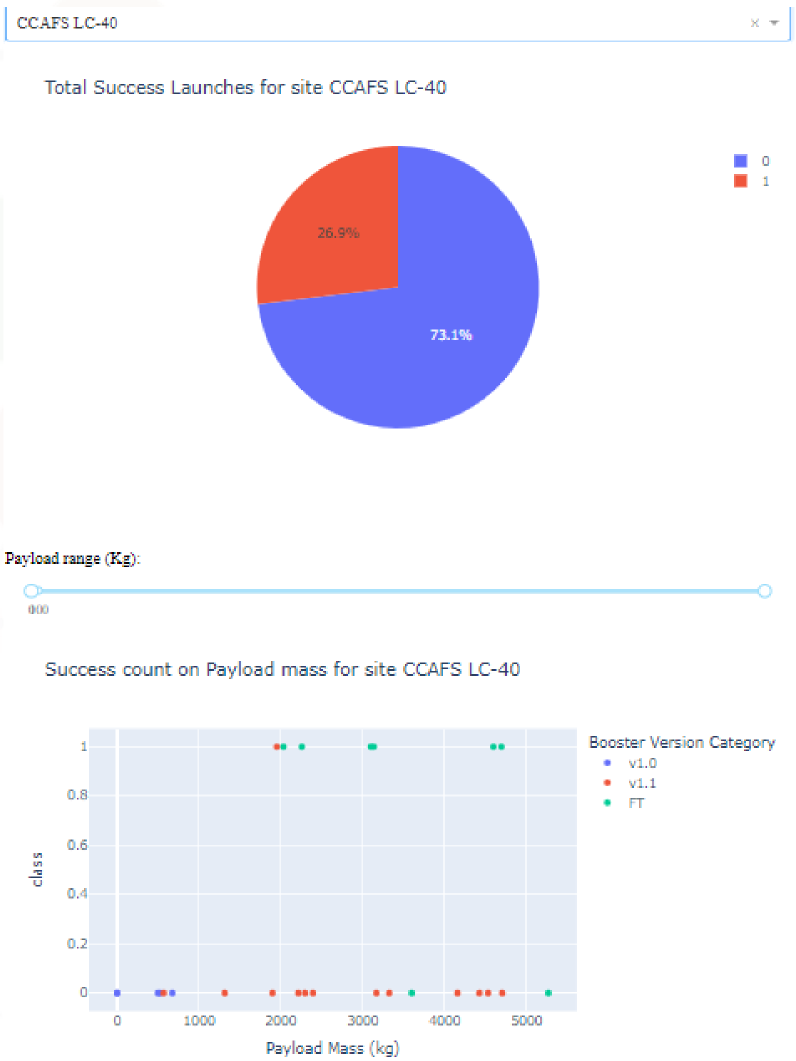
1) Launch sites are quite close to railways (1.3km)
2) Launch sites are quite close to highways (0.6km)
3) Launch sites are quite close to coastlines (0.86km)
4) Launch sites are pretty far from cities (51.4km)

Github publicly opened code with comments:
https://github.com/gavriushkinegor/IBM-Capstone/blob/main/6%20Interactive%20Visual%20Analytics%20with%20Folium%20lab%20lab_jupyter_launch_site_location.ipynb
Alternative link with NBViewer(many browsers don't work with Folium)
https://nbviewer.org/github/gavriushkinegor/IBM-Capstone/blob/main/6%20Interactive%20Visual%20Analytics%20with%20Folium%20lab%20lab_jupyter_launch_site_location.ipynb

Map markers added and clustered in order to visualize launch sites, outcome of the launch.
Folium map also allowed us to calculate distance between Launch site and vital infrastructure such as highways, railways, cities.

Build a dashboard with Plotly Dash





Github publicly opened code with comments:
https://github.com/gavriushkinegor/IBM-Capstone/blob/main/7%20spacex_dash_app%20completed.py

Feature Selection and evaluation

As a part of a research, I wanted to evaluate the weight and the importance of the features selected.

I used 3 methods of estimating feature importance. Let me also give you a short summary about each of them.

- **Univariate selection** - Statistical tests can be used to select those features that have the strongest relationship with the output variable. The scikit-learn library provides the SelectKBest class that can be used with a suite of different statistical tests to select a specific number of features. This method is based on chi2 evaluation for each feature in our dataset towards the independent feature, which is a class (or launch outcome, where class 0 is failure and class 1 is success) in our case.

  This method was the method of choice when building our model.

- **Feature Importance-** You can get the feature importance of each feature of your dataset by using the feature importance property of the model. Feature importance gives you a score for each feature of your data, the higher the score more important or relevant is the feature towards your output variable.

- **Correlation Matrix with Heatmap** (more information in discussion section)

  Interesing results were also achieved using Correlation Matrix with Heatmap.

  Let me tell you a bit more about this method.

  - Correlation states how the features are related to each other or the target variable.
  - Correlation can be positive (increase in one value of feature increases the value of the target variable) or negative (increase in one value of feature decreases the value of the target variable)
  - Heatmap makes it easy to identify which features are most related to the target variable, we will plot heatmap of correlated features using the seaborn library.

Github publicly opened code with comments:
https://github.com/gavriushkinegor/IBM-Capstone/blob/main/Feature%20Importance%20Evaluation.ipynb

Predictive Analysis (Classification)

Builidng a model is crucial and will help us predict if the stage of the SpaceX Falcon 9 rocket will land successfully.

Four classification models were used.

Here is a list of each of them with performance tests.

The metrics chosen for each model are R2 (determination coefficient), Jaccard Score and F1 Score.

Classification model	R2	Jaccard Score	F1 Score
Logistic Regression	0.83	0.8	0.8
SVM	0.84	0.8	0.89
Decision Trees	0.89	0.85	0.91
KNN	0.84	0.8	0.89

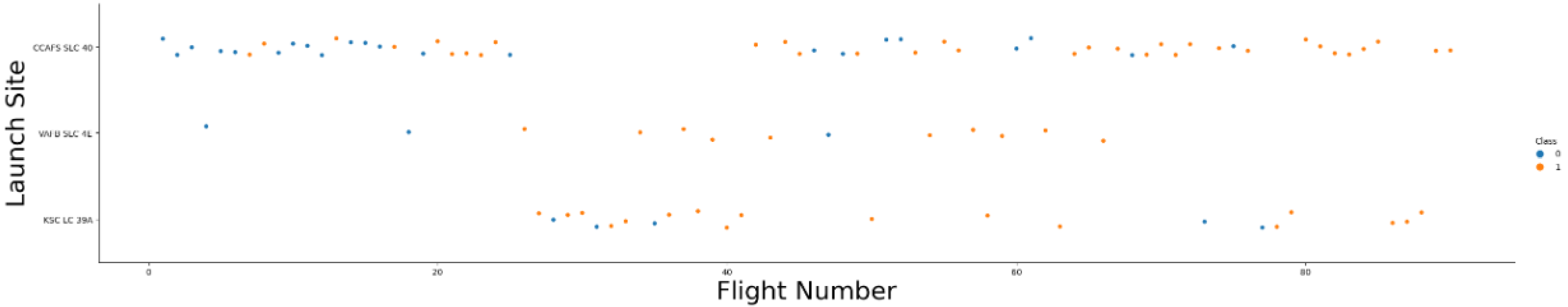
Results

- Decision Tree model worked best and outperformed other models, judging by all the metrics used:
  - R2 = 0.89
  - F1 score = 0.91
  - Jaccard score = 0.85
- Low weighted payload launches perform better than heavy weighted payload launches
- The success rate of SpaceX Launches is positively correlated with number of years of which they launch their rockets
- KSC LC39A had the most successful launches comparing to all other sites
- Orbit HEO, LEO, SSO, ES L1 had the highest mission success rate

Insights Drawn From EDA

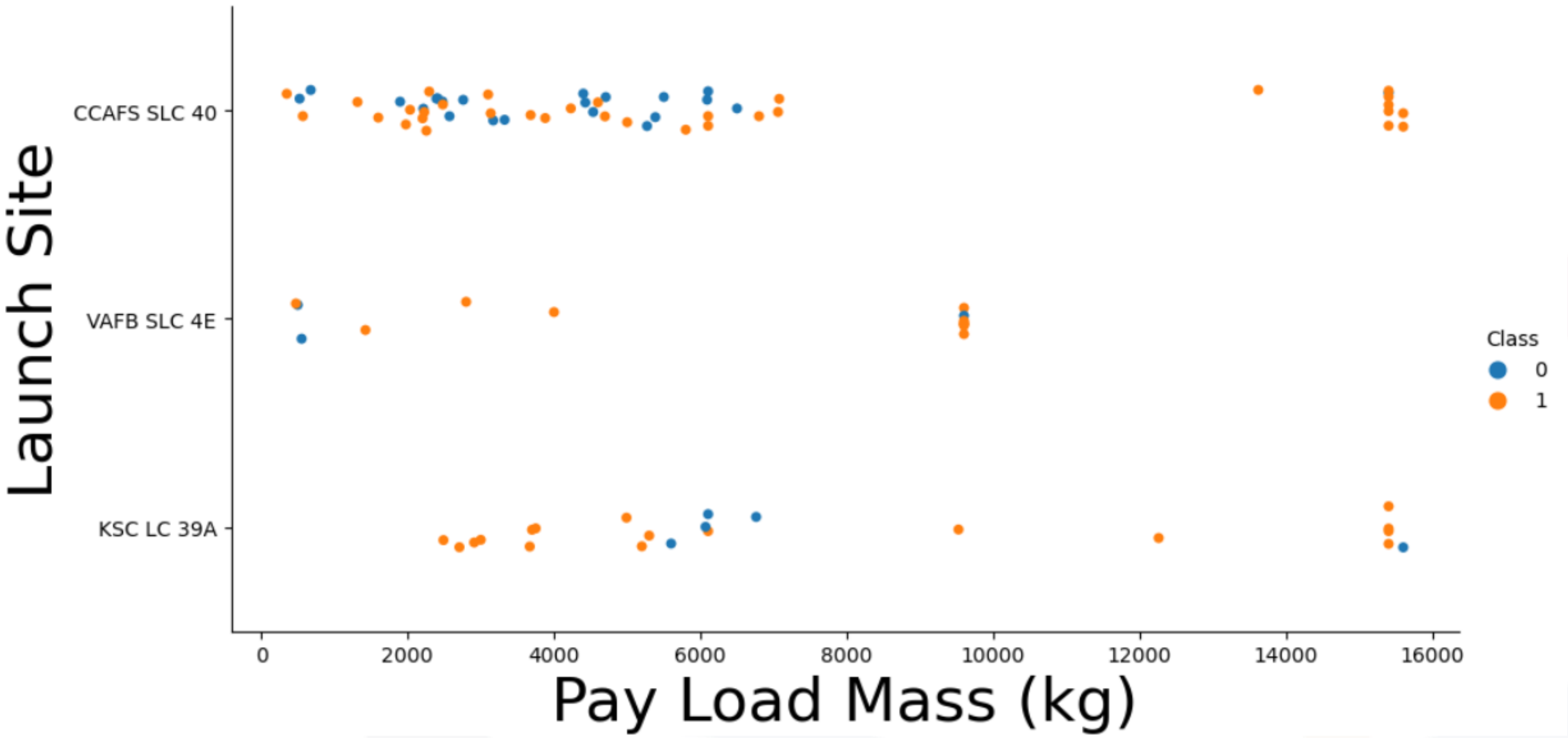
Discussion section

Launch Site vs Flight Number



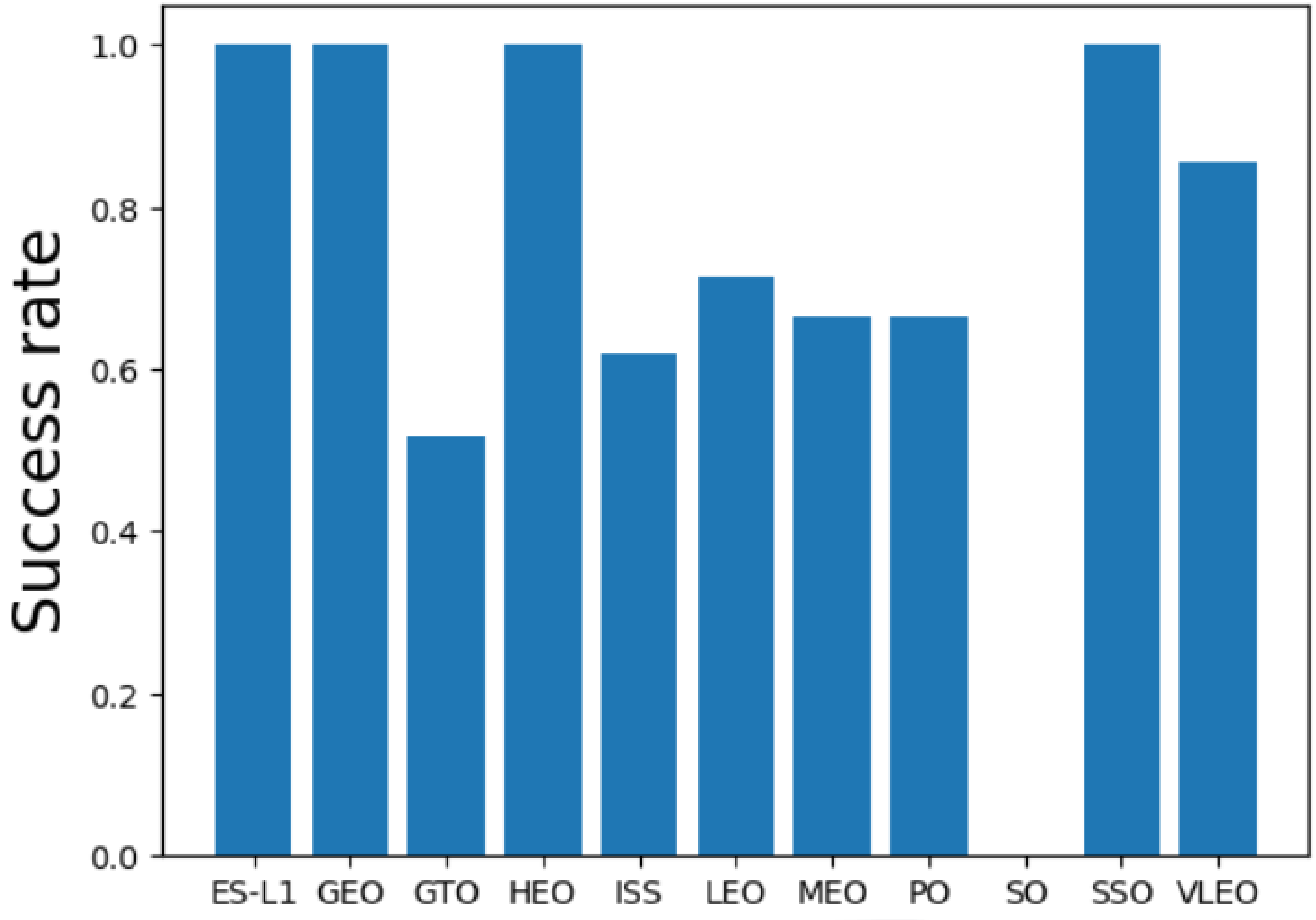
The amount of launches from CCAFS SLC 40 is significantly higher than from other sites

Payload mass vs Launch site



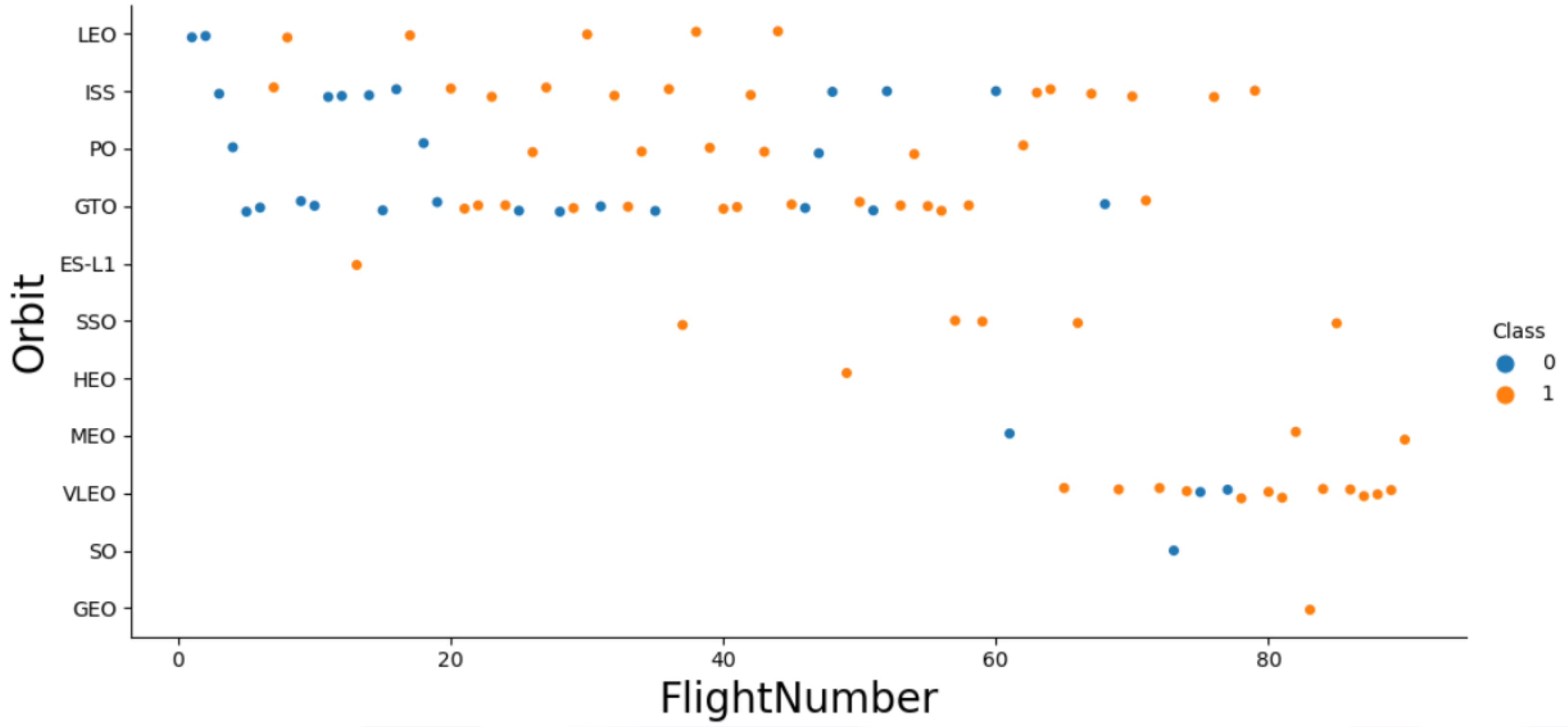
The majority of launches with payload mass from 0 to 7500kg were launched from CCAFS SLC 40 site.

Orbit vs Launch outcome



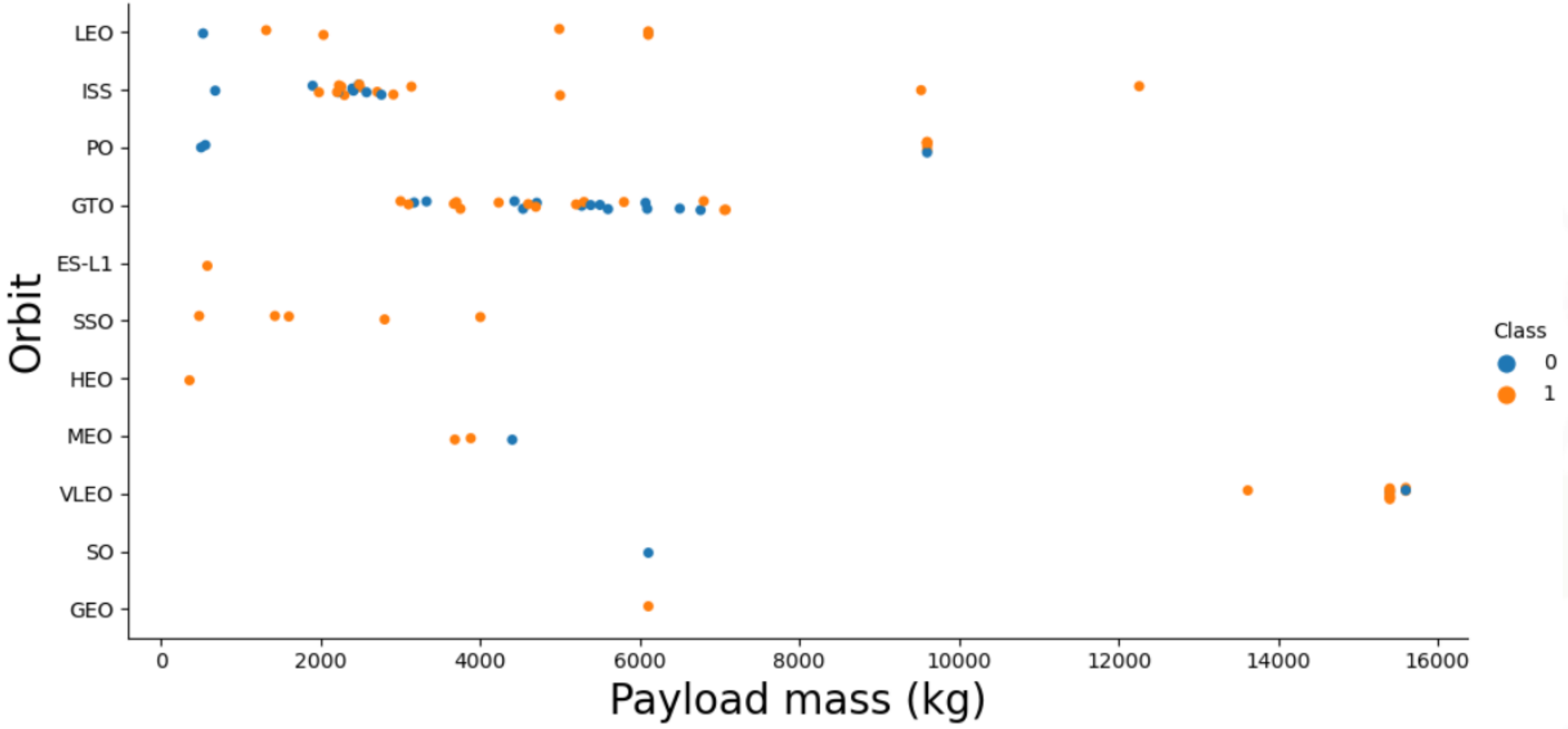
Launches to orbits ES-L1, GEO, HEO, SSO and VLEO are more likely to have successful outcomes than others.

Orbit vs Flight Number



- The amount of launches to VLEO increased over the years and the flight numbers count.
- GTO, PO, ISS AND LEO were the most frequent launches among others

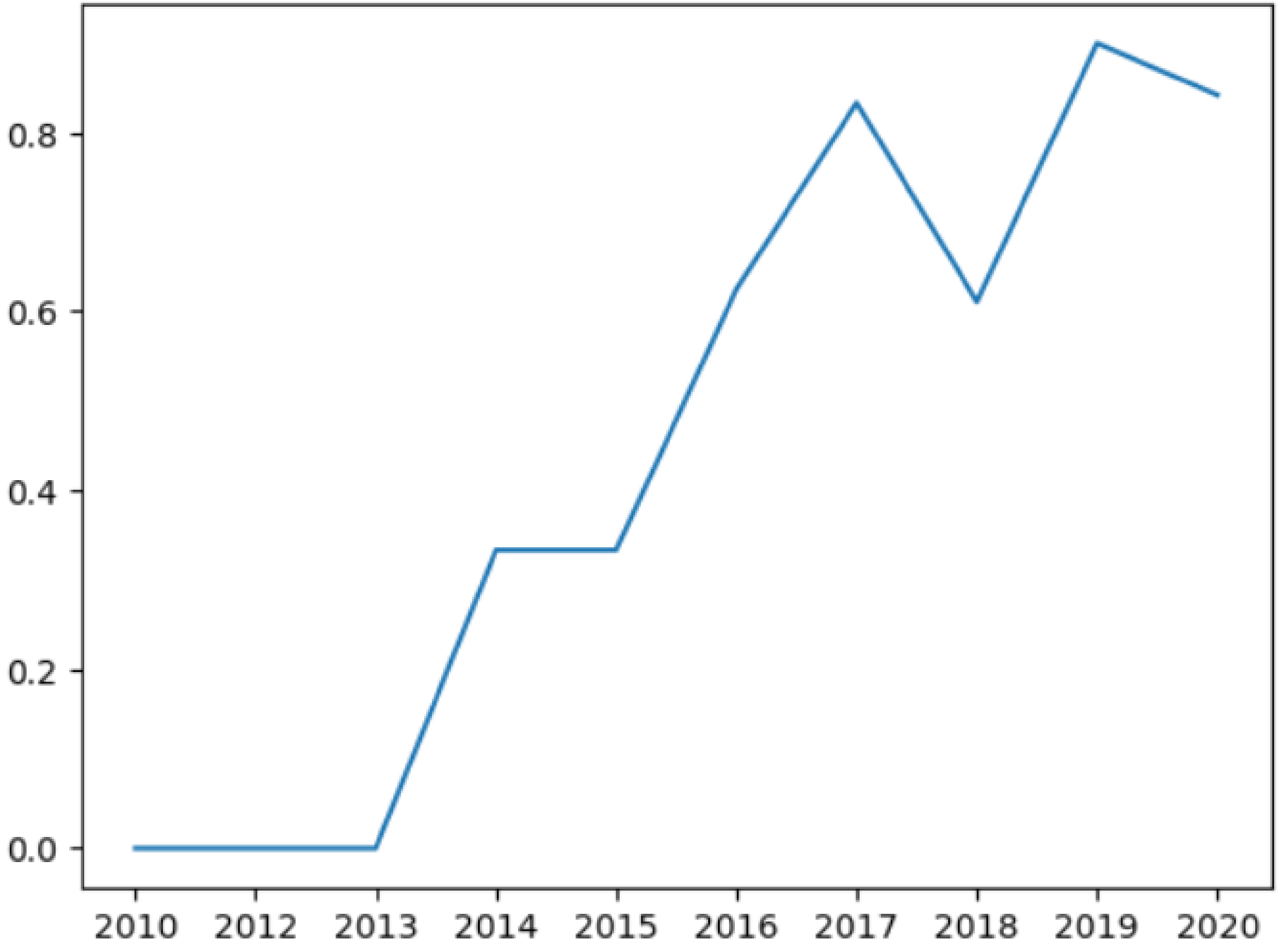
Payload vs Orbit type



- The most popular payload mass for ISS orbit is from 2000kg to 4000kg

- The most popular payload mass for GTO orbit ranges from 2500kg to 8000kg

Launch success trend over the years



- After year 2013 the amount of successful launches have grown dramatically
- From year 2014 to 2015 success rate was stabilized
- Year 2015 and 2016 was very successful in terms of launch outcomes for SpaceX
- The success rate peaked in year 2019 and is stabilized since then

Unique launch sites in the space mission

```
%sql SELECT DISTINCT LAUNCH_SITE FROM SPACEXTBL
```

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

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

Launch sites begin with the string 'CCA'

```
# %sql select * FROM SPACEXTBL WHERE 'LAUNCH_SITE' LIKE 'CCA%' LIMIT 5
%sql SELECT * FROM SPACEXTBL WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5
```

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing _Outcome
04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
08-12-2010	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
22-05-2012	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
01-03-2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total payload mass carried by boosters launched by NASA (CRS)

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE CUSTOMER = 'NASA (CRS)'
#%sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE CUSTOMER LIKE 'NASA%'
```

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

SUM(PAYLOAD_MASS__KG_)
45596

Average payload mass carried by booster version F9 v1.1

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE BOOSTER_VERSION = 'F9 v1.1'
```

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

AVG(PAYLOAD_MASS__KG_)
2928.4

The date when the first succesful landing outcome in ground pad was acheived

```
#%sql SELECT MIN(DATE) FROM SPACEXTBL WHERE Landing _Outcome = 'Success (ground pad)'
%sql SELECT min(DATE) FROM SPACEXTBL WHERE "Landing _Outcome" ='Success (ground pad)'
```

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

min(DATE)
01-05-2017

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

```
%sql SELECT "Booster_Version" FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ > '4000' AND PAYLOAD_MASS__KG_ < '6000' AND "LANDING _OUTCO
```

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

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

The total number of successful and failure mission outcomes

```
%sql SELECT COUNT(*) FROM SPACEXTBL WHERE Mission_Outcome LIKE '%Success%' OR Mission_Outcome LIKE '%Failure%'
```

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

COUNT(*)
101

The names of the booster versions which have carried the maximum payload mass

```
# %sql SELECT "BOOSTER_VERSION" FROM SPACEXTBL WHERE (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL)
%sql SELECT "BOOSTER_VERSION" FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL)
```

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

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

Month names, outcomes in drone ship ,booster versions, launch site for the months in year 2015

```
%%sql SELECT substr(Date, 4, 2) as month,booster_version,"Landing _Outcome"
from SPACEXTBL where "Landing _Outcome"
='Failure (drone ship)' and substr(Date,7,4)='2015'
```

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

month	Booster_Version	Landing _Outcome
01	F9 v1.1 B1012	Failure (drone ship)
04	F9 v1.1 B1015	Failure (drone ship)

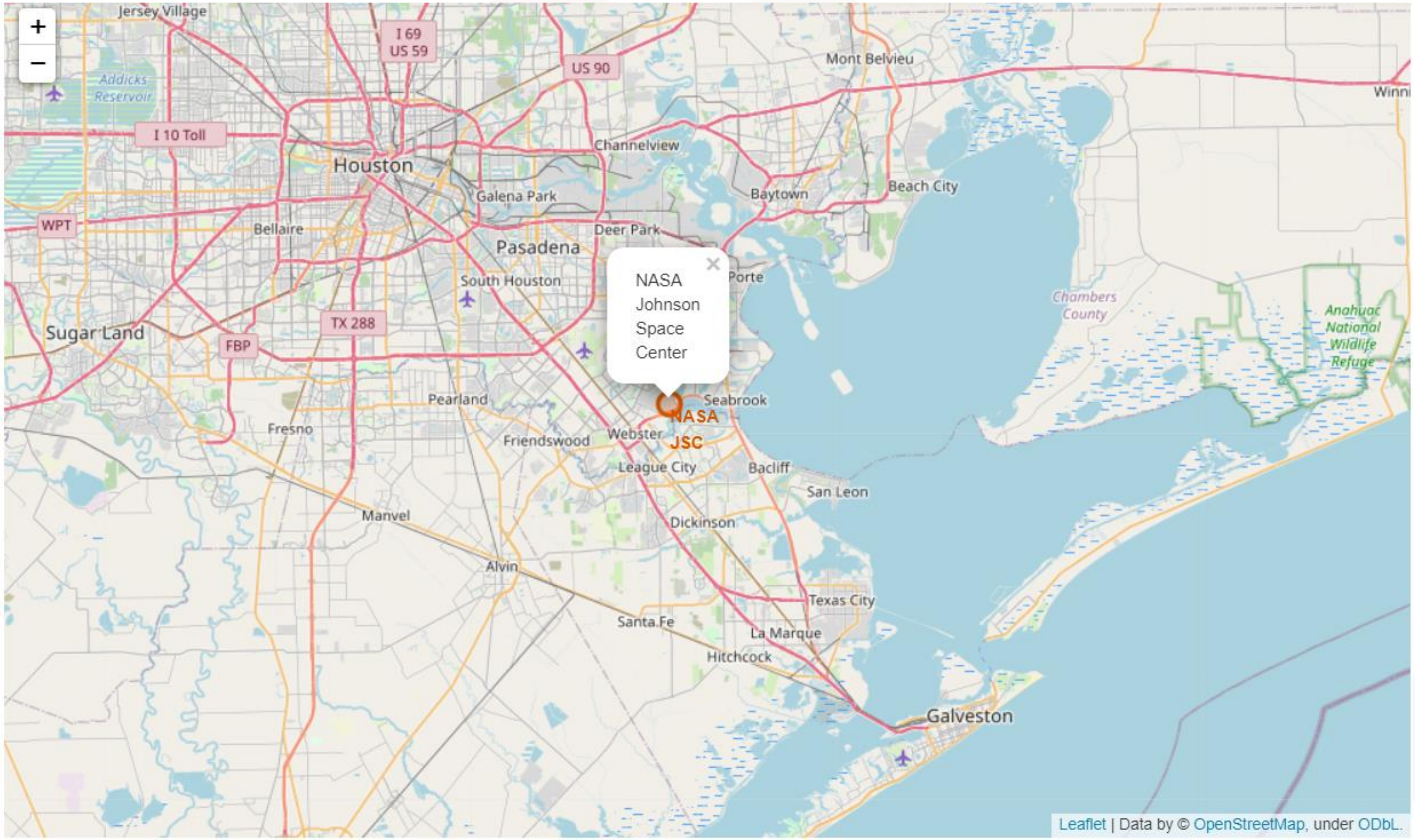
Ranking the count of successful landing outcomes between the date 04-06-2010 and 20-03-2017 in descending order

```
%%sql SELECT "DATE", COUNT("LANDING _OUTCOME") AS Successfull_outcomes_count FROM SPACEXTBL
     WHERE substr(Date,7,4) || substr(Date,4,2) || substr(Date,1,2) between '20100604'
and '20170320'AND "LANDING _OUTCOME" LIKE '%Success%'
    GROUP BY "DATE"
    ORDER BY COUNT("LANDING _OUTCOME") DESC
```

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

Date	Successfull_outcomes_count
27-05-2016	1
22-12-2015	1
19-02-2017	1
18-07-2016	1
14-08-2016	1
14-01-2017	1
08-04-2016	1
06-05-2016	1

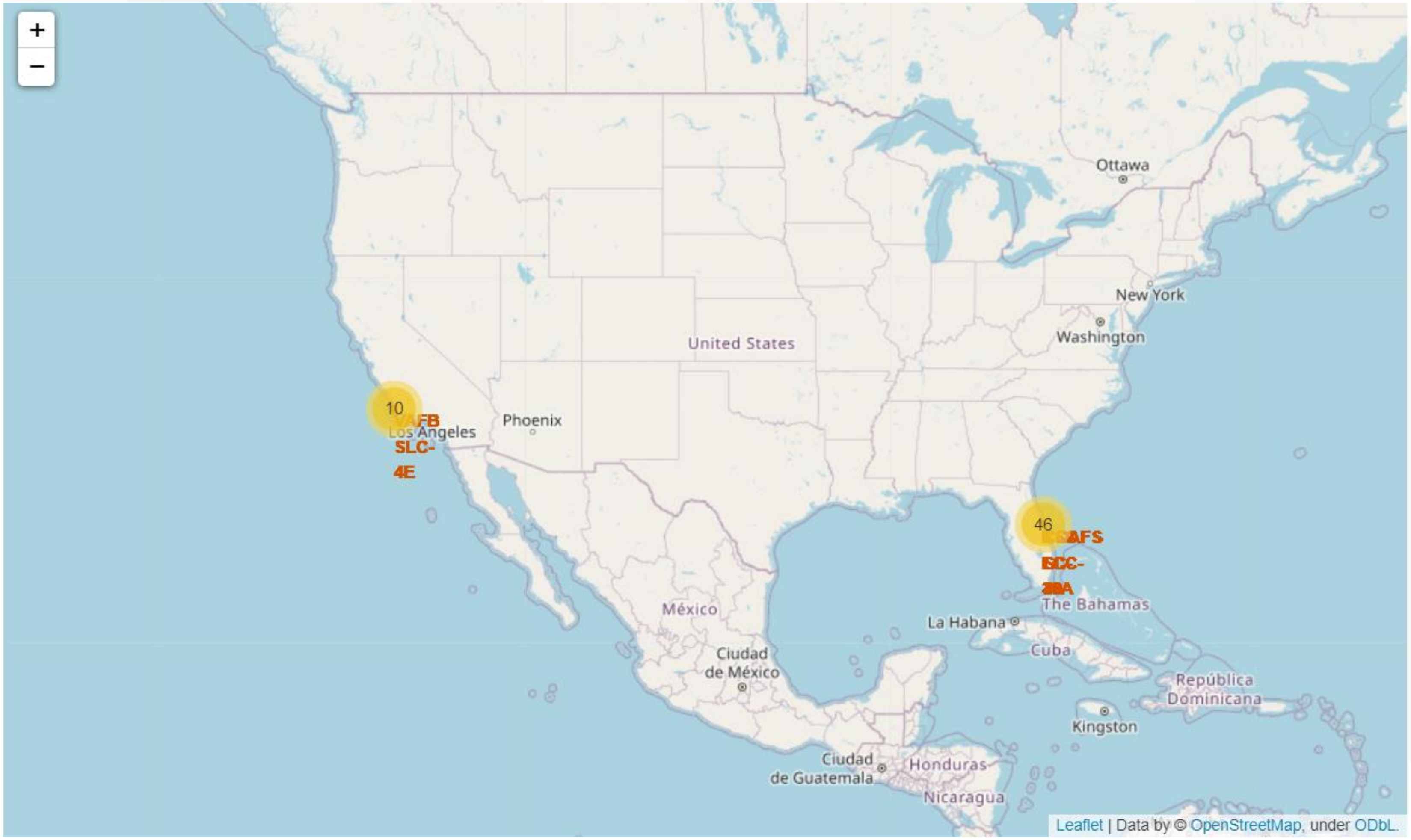
All launches sites marked on a map



IBM Developer

SKILLS NETWORK

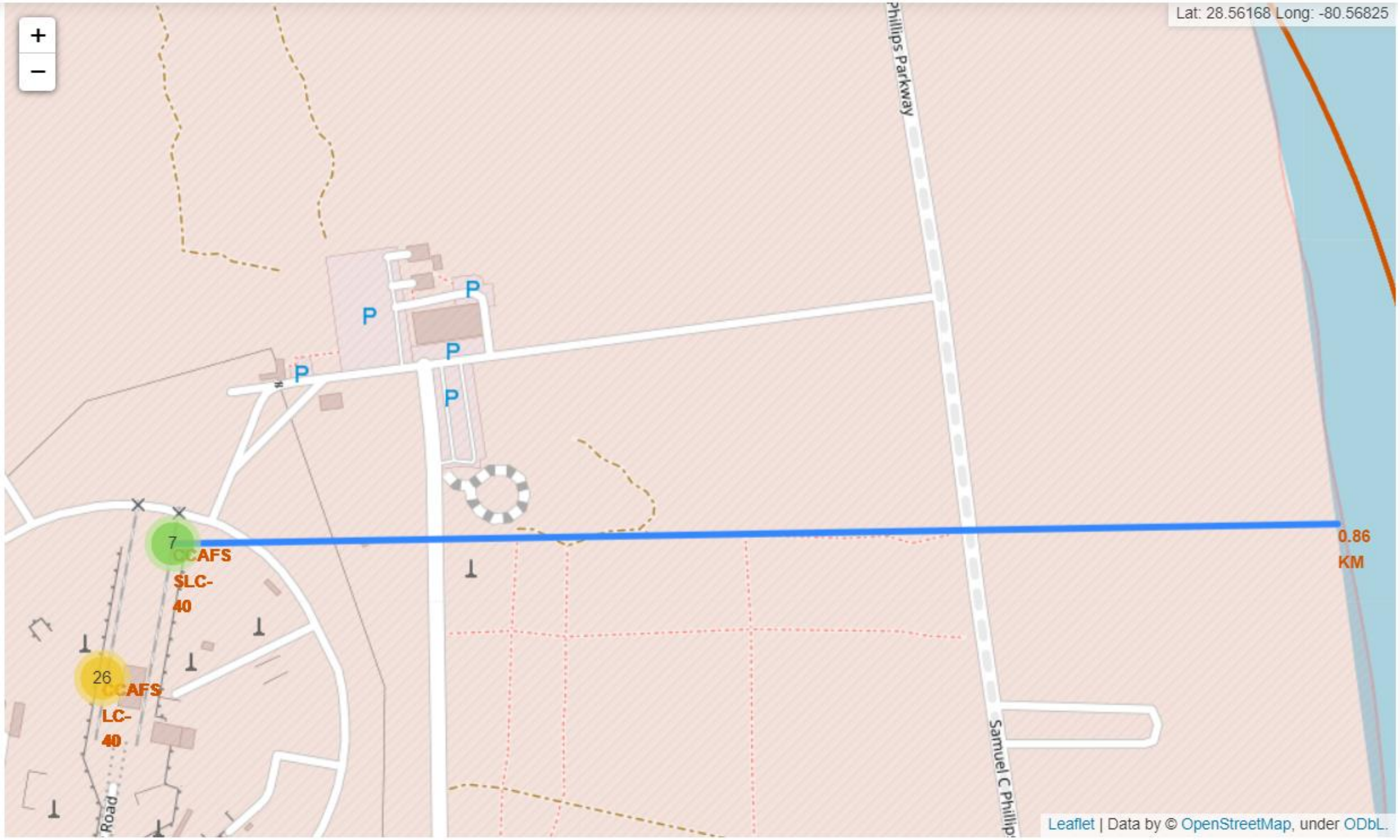
Marking the success/failed launches for each site on the map



IBM Developer

SKILLS NETWORK

Distances between a launch site to the coast



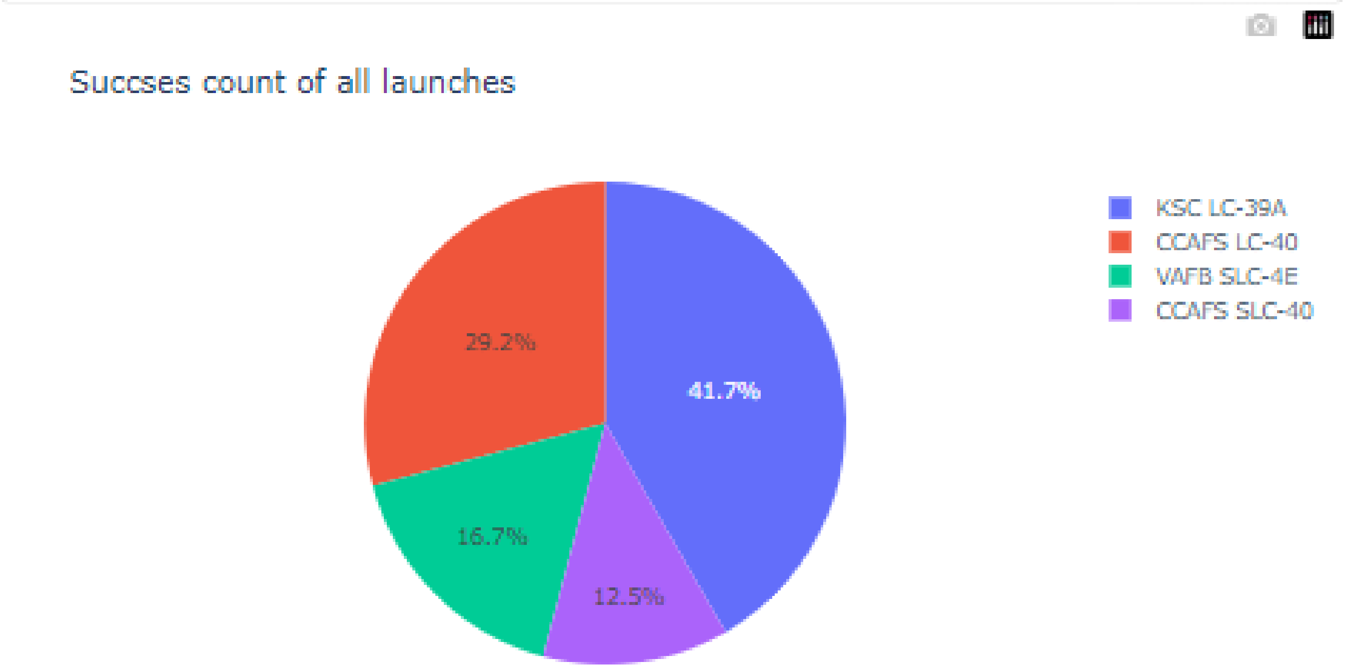
IBM Developer

SKILLS NETWORK

SEGA™

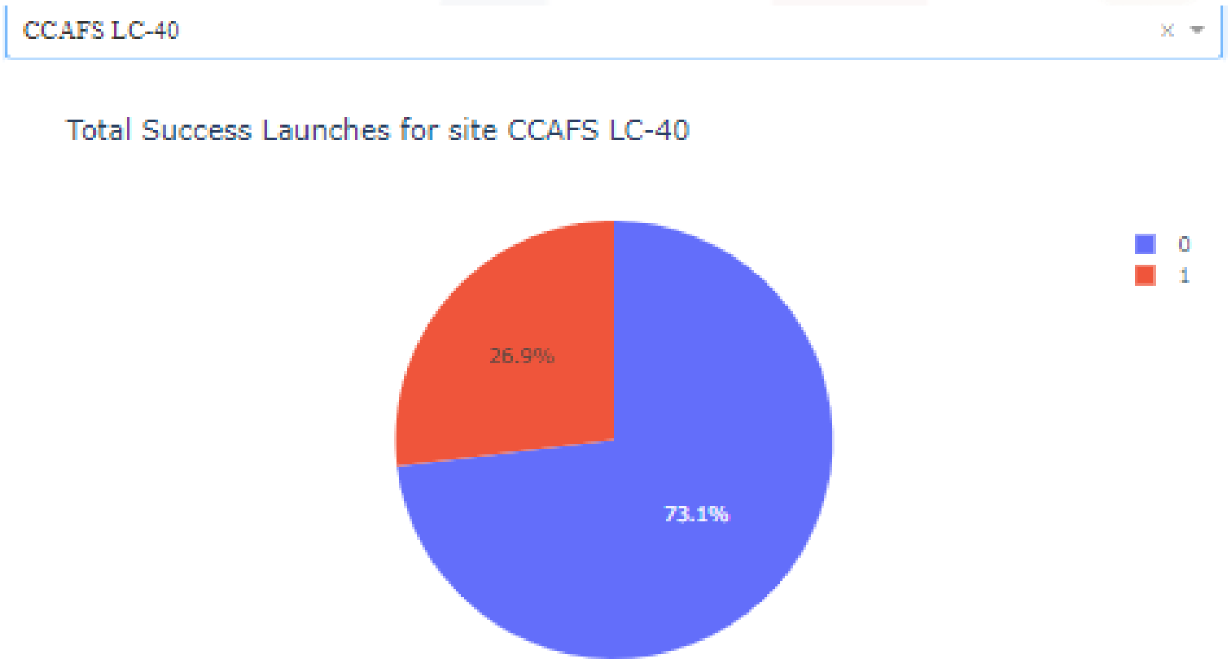
Sega? Not really. Interactive plots with Plotly Dash

Total Success by all launches



We can see that KSC LC-39A has the highest success rate

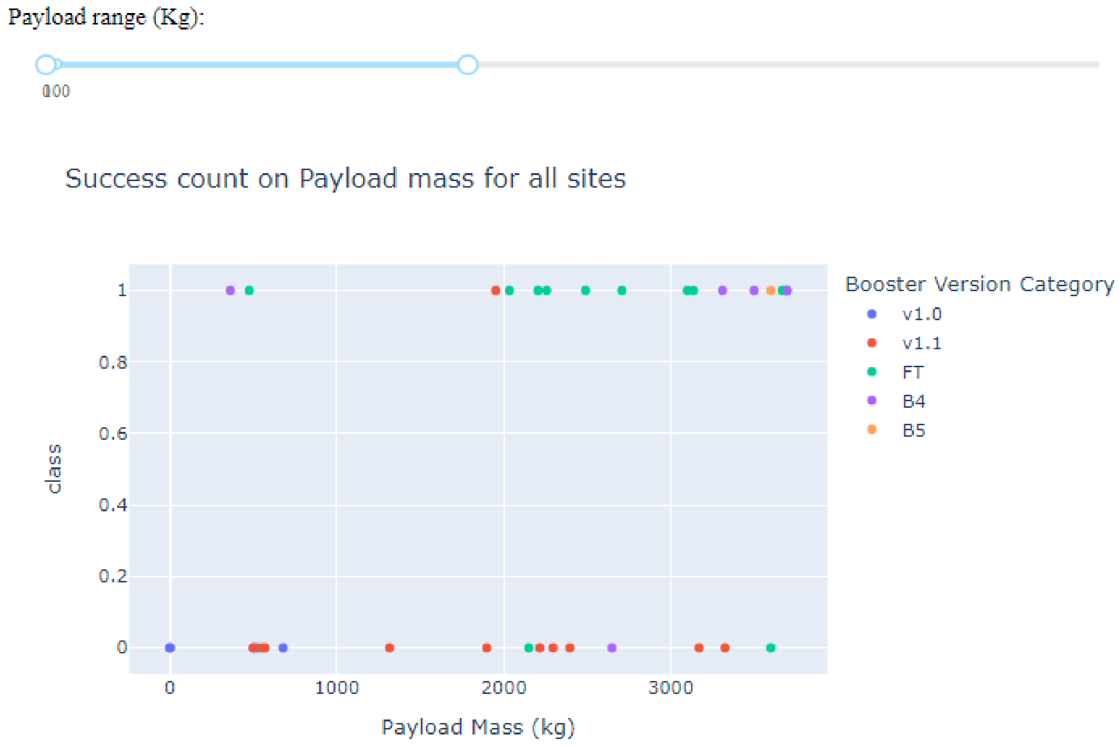
Success rate by site



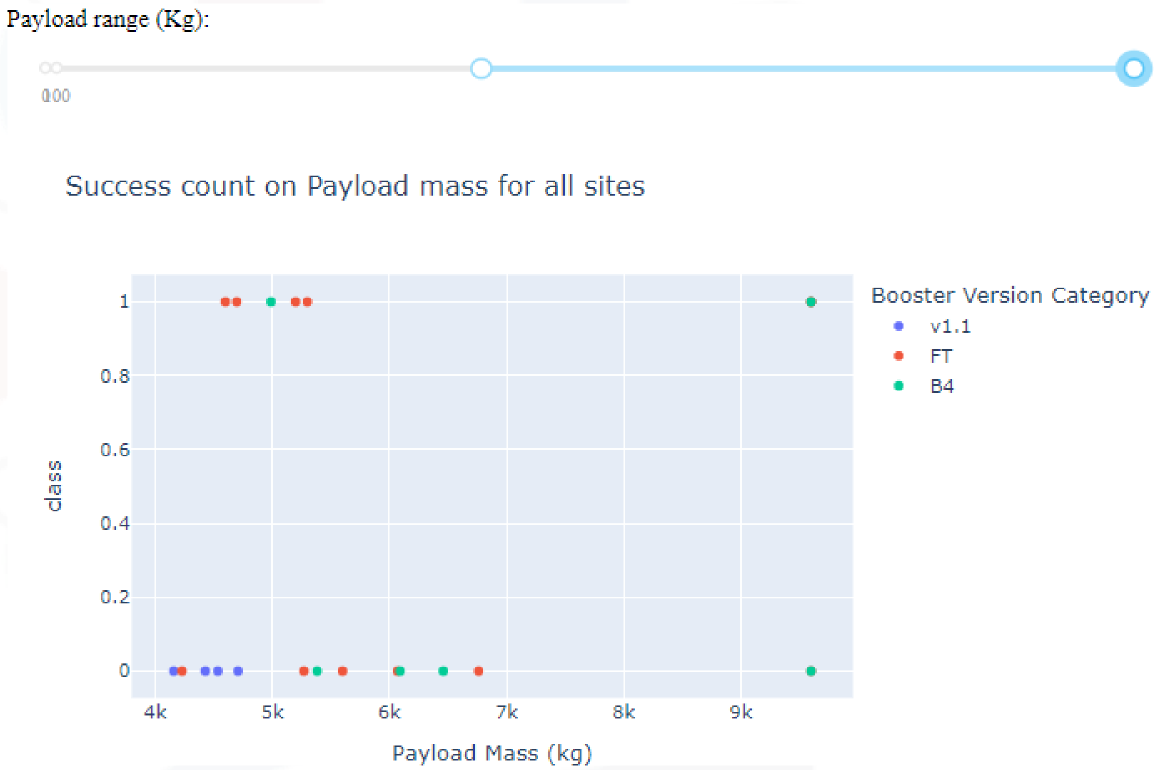
CCAFS LC-40 has the highest success rate among all launch sites

Success rate based on payload mass

Low weighted payload 0 – 4000kg



Heavy weighted payload 4000-10000kg



We can observe that low weighted payload lunches have a higher success rate

IBM Developer

SKILLS NETWORK

Feature Selection and evaluation

As a part of a research, I wanted to evaluate the weight and the importance of the features selected.

I used 3 methods of estimating feature importance. Let me also give you a short summary about each of them.

- **Univariate selection** - Statistical tests can be used to select those features that have the strongest relationship with the output variable. The scikit-learn library provides the SelectKBest class that can be used with a suite of different statistical tests to select a specific number of features. This method is based on chi2 evaluation for each feature in our dataset towards the independent feature, which is a class (or launch outcome, where class 0 is failure and class 1 is success) in our case.

This method was the method of choice when building our model.

- **Feature Importance-** You can get the feature importance of each feature of your dataset by using the feature importance property of the model. Feature importance gives you a score for each feature of your data, the higher the score more important or relevant is the feature towards your output variable.

- **Correlation Matrix with Heatmap** (more information below)

Interesing results were also achieved using Correlation Matrix with Heatmap.

Let me tell you a bit more about this method.

- Correlation states how the features are related to each other or the target variable.
- Correlation can be positive (increase in one value of feature increases the value of the target variable) or negative (increase in one value of feature decreases the value of the target variable)
- Heatmap makes it easy to identify which features are most related to the target variable, we will plot heatmap of correlated features using the seaborn library.

Univariative selection

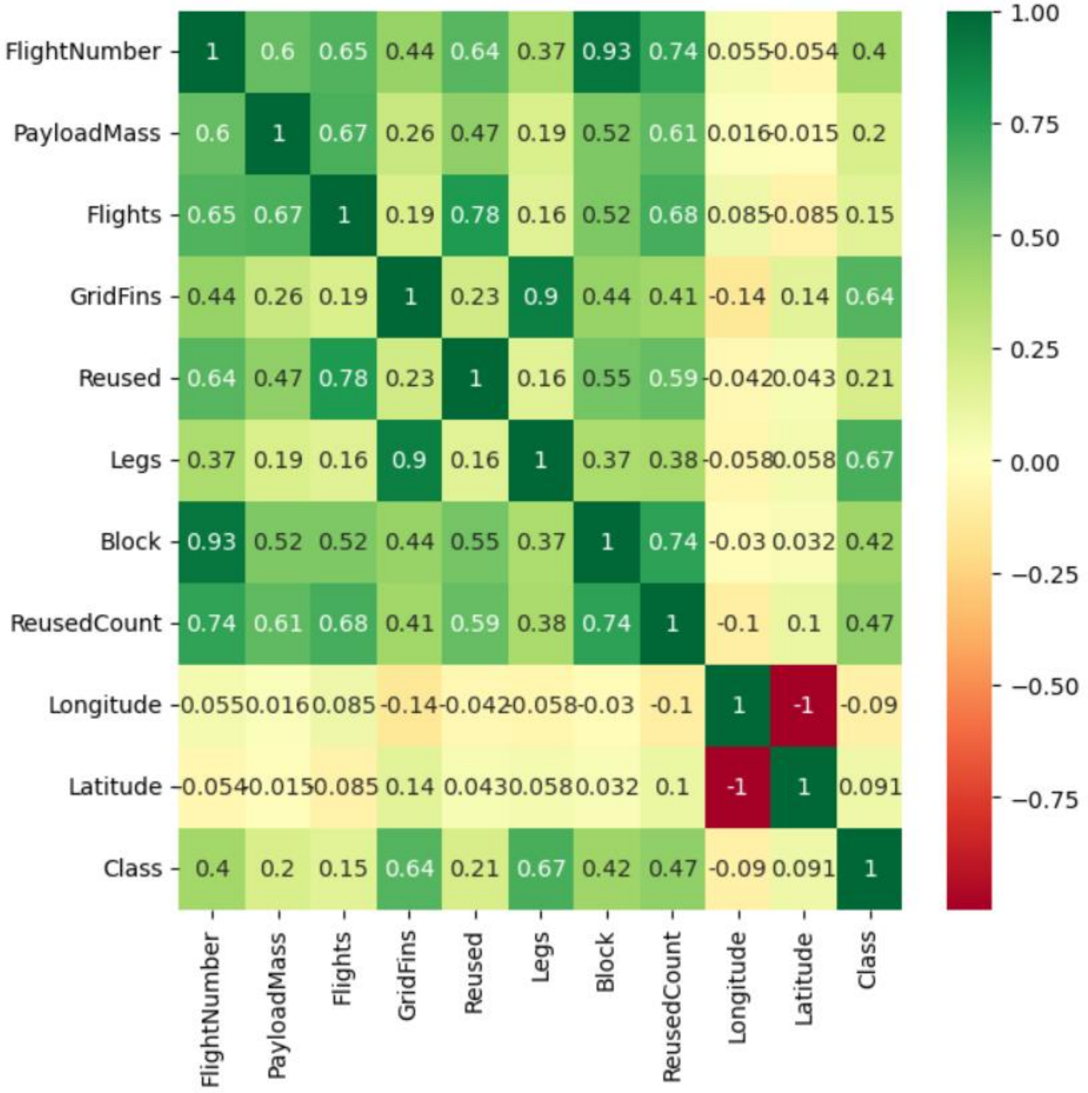
Finally, let's take a look of a top 10 most important feature scores

```
print(featurescores.nlargest(10,'Score'))
```

```
                               Feature         Score
1                          PayloadMass  12851.122424
0                         FlightNumber    215.658242
4                          ReusedCount     34.231544
81                          Legs_False     32.236842
77                      GridFins_False     28.900000
3                                Block     11.200000
82                           Legs_True      8.626761
78                       GridFins_True      8.257143
21  LandingPad_5e9e3032383ecb6bb234e7ca     5.714286
22  LandingPad_5e9e3032383ecb761634e7cb     4.000000
```

This are the features that will help us develop the most optimal model

Correlation Matrix with Heatmap



•Using Heatmap correlation matrix we can observe features that have a correlation with class feature.

•Class is a feature that predicts if the outcome will be successful or not.

•Every Feature that has an index higher that 0.2 is considered to have string correlation with launch outcome.

•This analysis was done in order to optimize our future classification models, which will give us a prediction, whether a launch outcome of SpaceX rocket will be successful

Predictive Analysis (Classification)

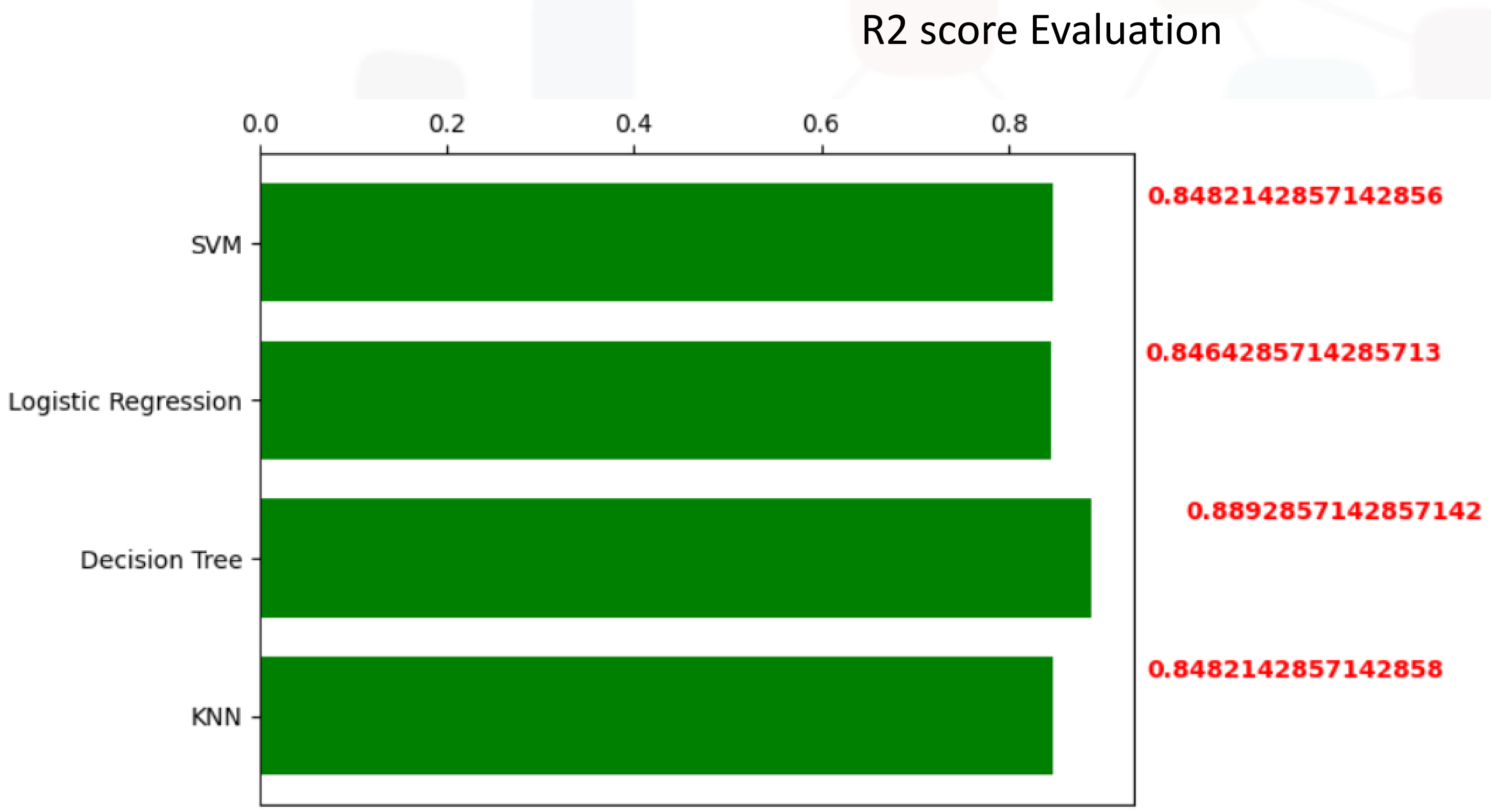
Four classification models were used.
Here is a list of each of them with performance tests.
The metrics chosen for each model are R2 (determination coefficient), Jaccard Score and F1 Score.

Classification model	R2	Jaccard Score	F1 Score
Logistic Regression	0.83	0.8	0.8
SVM	0.84	0.8	0.89
Decision Trees	0.89	0.85	0.91
KNN	0.84	0.8	0.89

Let's take a closer look with matplotlib visualizations on each metric of each model compared to others

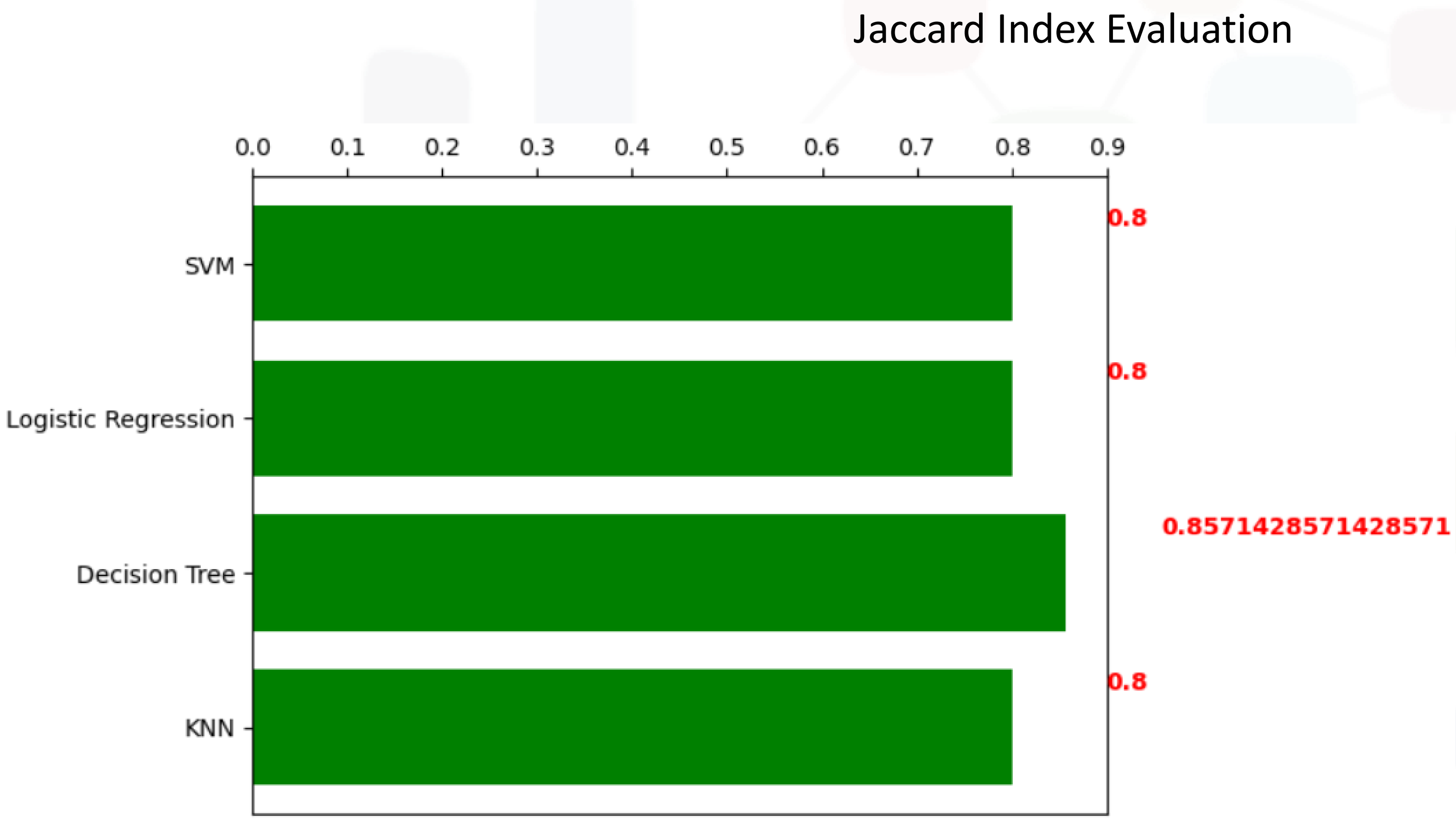
(see next slides)

R2 score Evaluation of the model



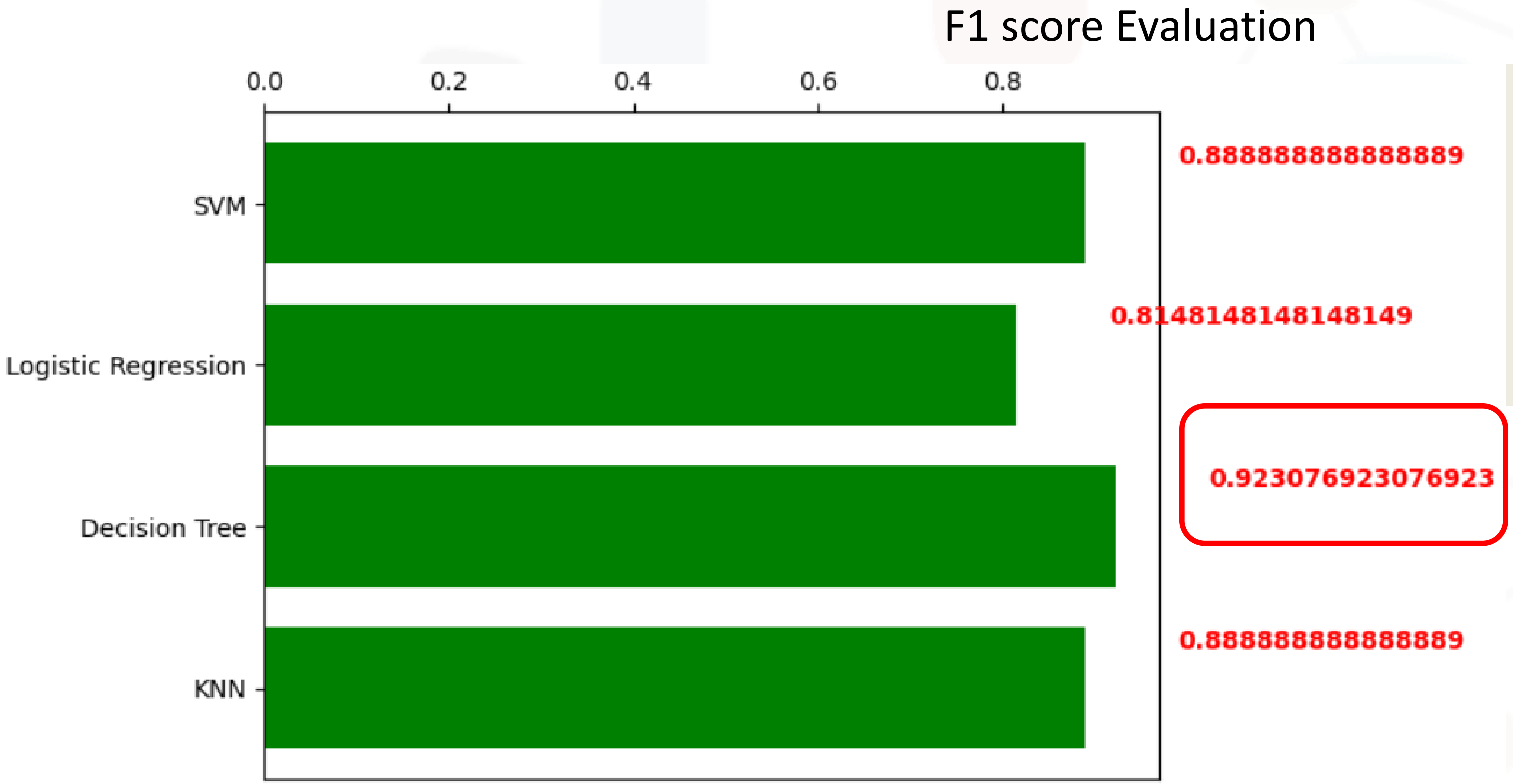
R2 Score for Decision trees beats all other models using this metric

Jaccard index Evaluation of the model



Jaccard Index for Decision tree beats all other models using this metric as well

F1 score Evaluation of the model



F1 Score for Decision trees beats all other models using this metric **by a margin!**

Conclusion

- Decision Tree model worked best and outperformed other models, judging by all the metrics used:
  - R2 = 0.89
  - F1 score = 0.91
  - Jaccard score = 0.85
- Low weighted payload launches perform better than heavy weighted payload launches
- The success rate of SpaceX Launches is positively correlated with number of years of which they launch their rockets
- KSC LC39A had the most successful launches comparing to all other sites
- Orbit HEO, LEO, SSO, ES L1 had the highest mission success rate

Thanks for watching!

Gavriushkin Egor

01.12.2022