IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Cuong, Nguyen Manh
10-Sep-2021

Outline

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

Executive Summary

In this capston, I will crawl the data thourgh SpaceX api and webscraping. Using python and Sql to clean data and perform Exploratory Data analysis (EDA) to get the general information about the data. And using mathplotlib, seaborn, and dash_plotly to visualize the data to find out useful information. Lastly, using some of methodology approaches to figure out the best fit model predict landing results such as Logistic Regression, Support Vector Machine, Decision Tree classification, K Nearest Neighbors classification.

After valuation, I find out that Decision Tree classification is the best method for my dataset.

Introduction

- **Project background and context**

The commercial space age is here, companies are making space travel affordable for everyone. Virgin Galactic is providing suborbital spaceflights. Rocket Lab is a small satellite provider. Blue Origin manufactures sub-orbital and orbital reusable rockets. Perhaps the most successful is SpaceX. SpaceX’s accomplishments include: Sending spacecraft to the International Space Station. Starlink, a satellite internet constellation providing satellite Internet access. Sending manned missions to Space. One reason SpaceX can do this is the rocket launches are relatively inexpensive. SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upwards of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage.

- **The purpose of this project**

To predict if the Falcon 9 first stage will land successfully or not by training a machine learning model and use public information. Therefore if we can determine if the first stage will land, we can determine the cost of a launch.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Using requests library to get information about Falcon 9 Rocket from SpaceX API
  - Using BeautifulSoup to crawl information about Falcon 9 Launces from wikipedia.org
- Perform data wrangling
  - Process the "outcome" to lable the Outcome column
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Standardize the data and split to trainset-testset then using GridSearchCV to tunning model

Data Collection

Requests data from SpaceX API

- Create user define function to get data from SpaceX API
- Parsing data and convert to DataFrame by using pandas function: json() and json_normalize()

Preprocessing data from API

- Filter data related to falcon 9 only
- Dealing with missing values
- Export data to CSV file

Web scraping Falcon 9 and Falcon Heavy Launches Records from Wikipedia

- Using BeautifulSoup to get data from Wikipedia
- Extract rows and columns
- Create DataFrame
- Export data to CSV file

Data Collection - SpaceX API

Requests data from SpaceX API

- Create user define function to get data from SpaceX API
- Parsing data and convert to DataFrame by using pandas function: json() and json_normalize()

Preprocessing data from API

- Filter data related to falcon 9 only
- Dealing with missing values
- Export data to CSV file

GitHub URL:
https://github.com/bnbbdz/Data_sience_ibm_course_capstone/blob/master/02_Notebook/jupyter-labs-spacex-data-collection-api.ipynb

```
data_falcon9
```

]:

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	Longitude	Latitude
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	28.561857
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	28.561857
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	28.561857
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	34.632093
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	28.561857
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
89	86	2020-09-03	Falcon 9	15600.000000	VLEO	KSC LC 39A	True ASDS	2	True	True	True	5e9e3032383ecb6bb234e7ca	5.0	7	B1060	-80.603956	28.608058
90	87	2020-10-06	Falcon 9	15600.000000	VLEO	KSC LC 39A	True ASDS	3	True	True	True	5e9e3032383ecb6bb234e7ca	5.0	7	B1058	-80.603956	28.608058
91	88	2020-10-18	Falcon 9	15600.000000	VLEO	KSC LC 39A	True ASDS	6	True	True	True	5e9e3032383ecb6bb234e7ca	5.0	9	B1051	-80.603956	28.608058
92	89	2020-10-24	Falcon 9	15600.000000	VLEO	CCSFS SLC 40	True ASDS	3	True	True	True	5e9e3033383ecbb9e534e7cc	5.0	7	B1060	-80.577366	28.561857
93	90	2020-11-05	Falcon 9	3681.000000	MEO	CCSFS SLC 40	True ASDS	1	True	False	True	5e9e3032383ecb6bb234e7ca	5.0	1	B1062	-80.577366	28.561857

90 rows × 17 columns

Data Collection - Scraping

- Using BeautifulSoup to get data from Wikipedia
- Extract rows and columns
- Create DataFrame
- Export data to CSV file

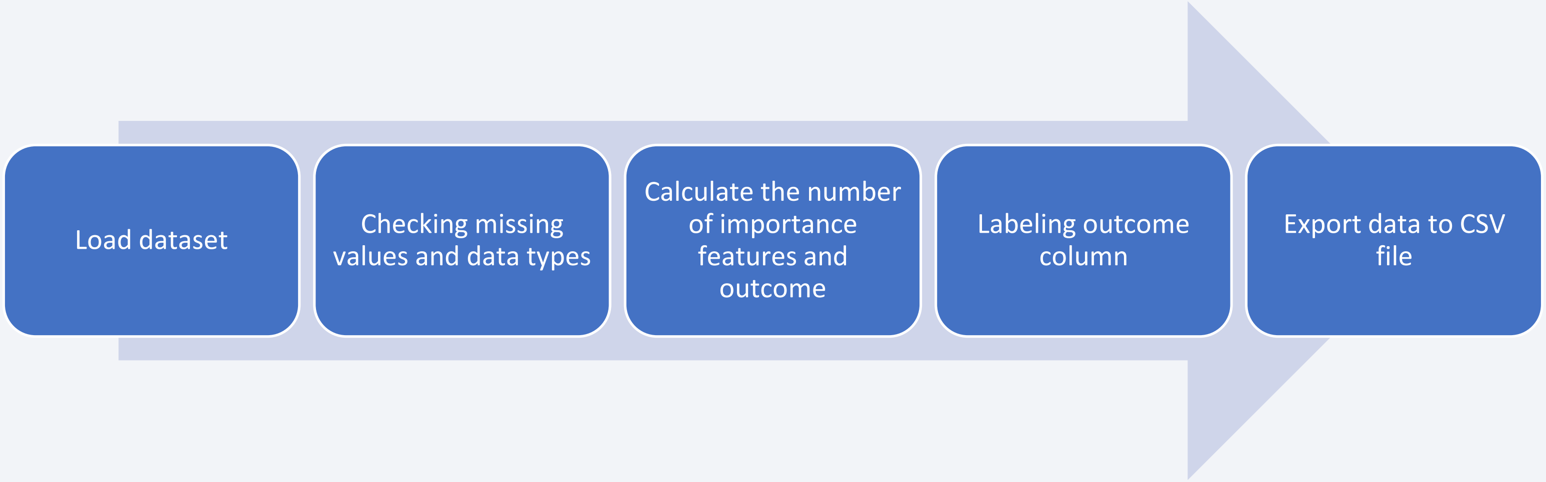
GitHub URL :
https://github.com/bnbbdz/Data_sience_ibm_course_capstone/blob/master/02_Notebook/jupyter-labs-webscraping.ipynb

```
df.head()
```

3]:

	Flight No.	Launch site	Payload	Payload mass	Orbit	Customer	Launch outcome	Version Booster	Booster landing	Date	Time
0	1	CCAFS	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success\n	F9 v1.0B0003.1	Failure	4 June 2010	18:45
1	2	CCAFS	Dragon	0	LEO	NASA	Success	F9 v1.0B0004.1	Failure	8 December 2010	15:43
2	3	CCAFS	Dragon	525 kg	LEO	NASA	Success	F9 v1.0B0005.1	No attempt\n	22 May 2012	07:44
3	4	CCAFS	SpaceX CRS-1	4,700 kg	LEO	NASA	Success\n	F9 v1.0B0006.1	No attempt	8 October 2012	00:35
4	5	CCAFS	SpaceX CRS-2	4,877 kg	LEO	NASA	Success\n	F9 v1.0B0007.1	No attempt\n	1 March 2013	15:10

Data Wrangling



GitHub URL :
https://github.com/bnbbdz/Data_sience_ibm_course_capstone/blob/master/02_Notebook/labs-jupyter-spacex-Data%20wrangling.ipynb

```
df.head(5)
```

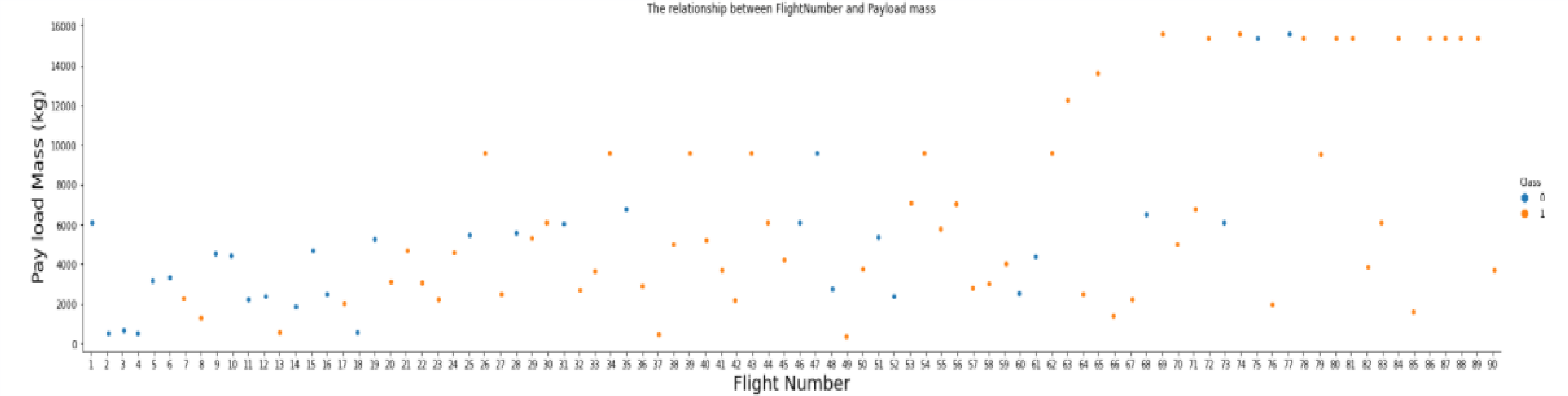
	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	Longitude	Latitude	Class
0	1	2010-06-04	Falcon 9	6104.959412	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0003	-80.577366	28.561857	0
1	2	2012-05-22	Falcon 9	525.000000	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0005	-80.577366	28.561857	0
2	3	2013-03-01	Falcon 9	677.000000	ISS	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0007	-80.577366	28.561857	0
3	4	2013-09-29	Falcon 9	500.000000	PO	VAFB SLC 4E	False Ocean	1	False	False	False	NaN	1.0	0	B1003	-120.610829	34.632093	0
4	5	2013-12-03	Falcon 9	3170.000000	GTO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B1004	-80.577366	28.561857	0

EDA with Data Visualization

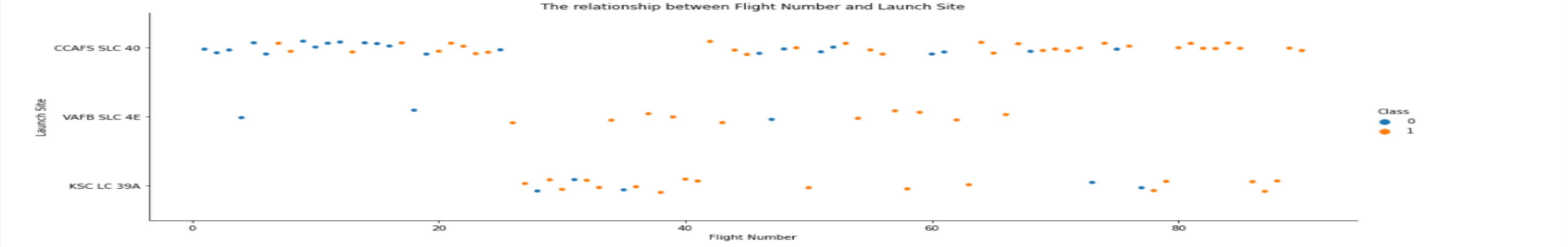
- I used some kind of charts like scatter point chart, line chart, bar chart.
- Through catter point chart I can see the relationship between features such as The relationship between FlightNumber and Payload mass , The relationship between Flight Number and Launch Site, The relationship between Payload mass and Launch Site , etc
- Based on bar chart I can see the success rate in different Orbit type,
- And finally I used line chart to see the success trend line during the period 2010 to 2020.

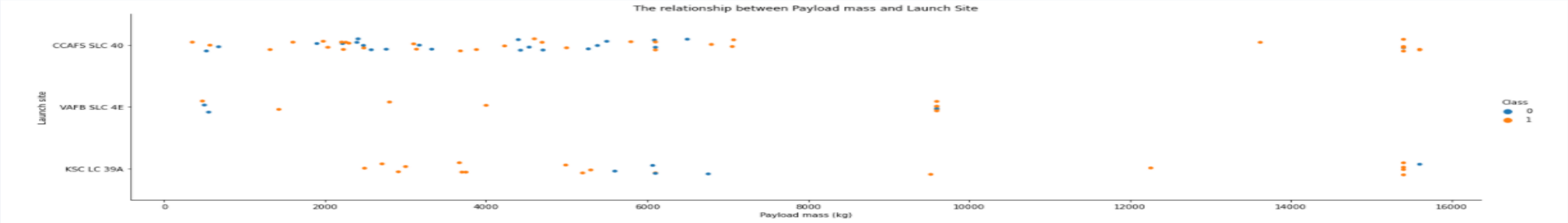
GitHub URL : https://github.com/bnbbdz/Data_sience_ibm_course_capstone/blob/master/02_Notebook/jupyter-labs-eda-dataviz.ipynb

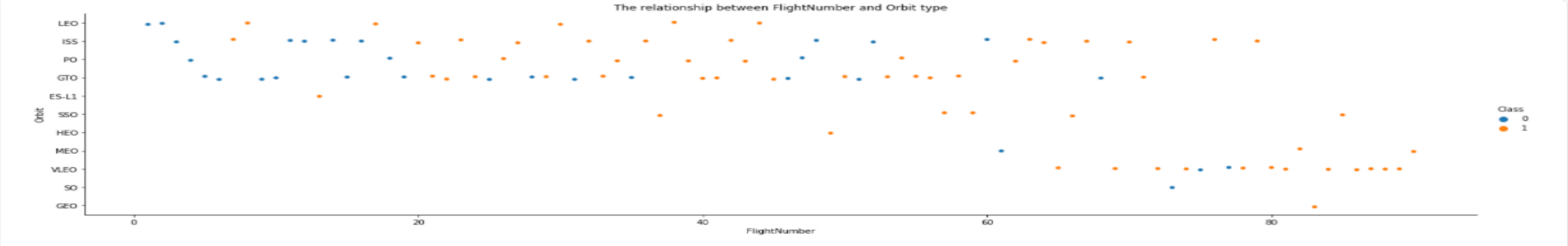
For example you can see on the chart “The relationship between FlightNumber and Payload mass” that as the flight number increases, the first stage is more likely to land successfully. The payload mass is also important; it seems the more massive the payload, the less likely the first stage will return



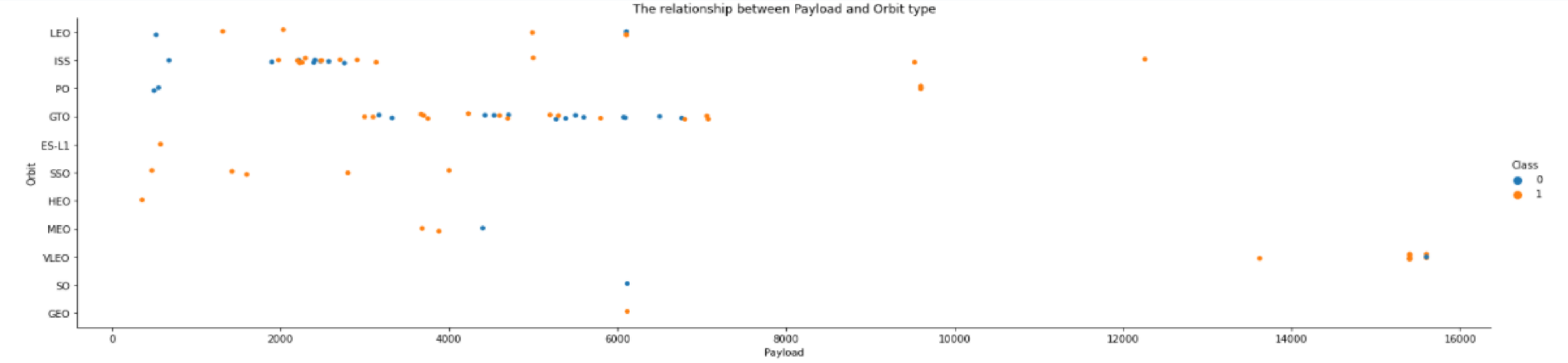
EDA with Data Visualization

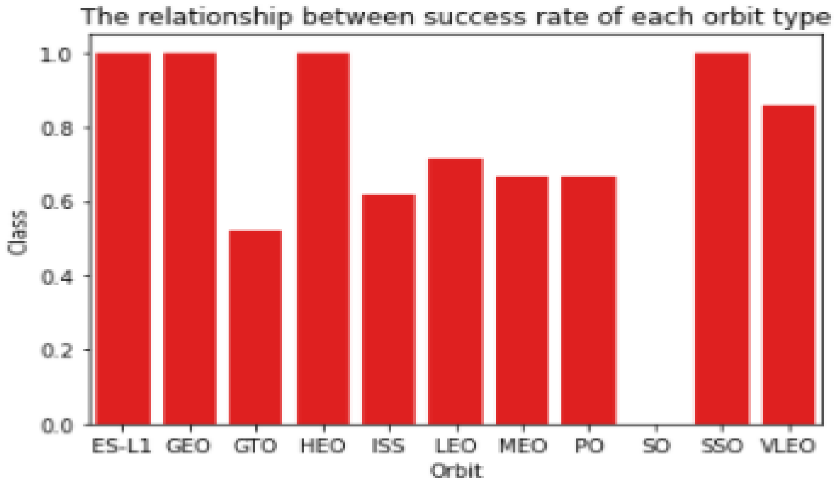






EDA with Data Visualization





EDA with SQL

- Connect to the database using SQL magic commands
- Using distinct to Display the names of the unique launch sites in the space mission
- Using Like and limit to Display 5 records where launch sites begin with the string 'CCA‘
- Using aggregate function sum() to calculate the total payload mass carried by boosters launched by NASA (CRS)
- Using aggregate function avg () and like to calculate average payload mass carried by booster version F9 v1.1
- Using aggregate function min () and like to get the first successful landing outcome in ground pad was achieved
- Using between to filter the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
- Using group by to List the total number of successful and failure mission outcomes
- Using subquery to the names of the booster_versions which have carried the maximum payload mass
- Using group by and order by to 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

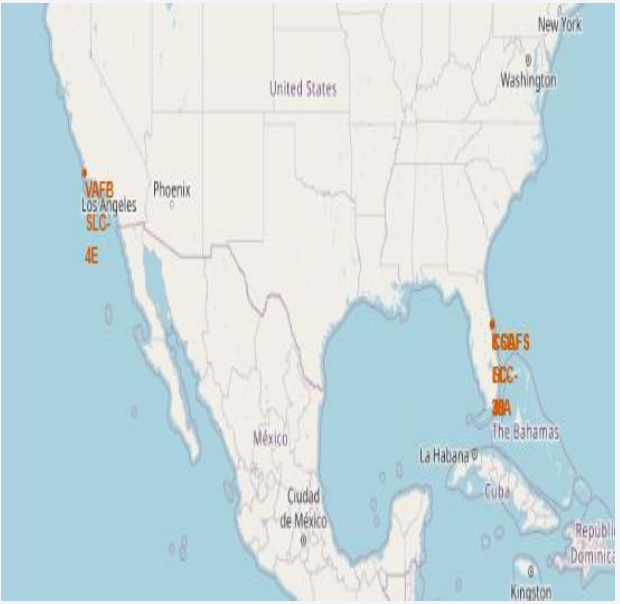
GitHub URL : https://github.com/bnbbdz/Data_sience_ibm_course_capstone/blob/master/02_Notebook/jupyter-labs-eda-sql-coursera.ipynb

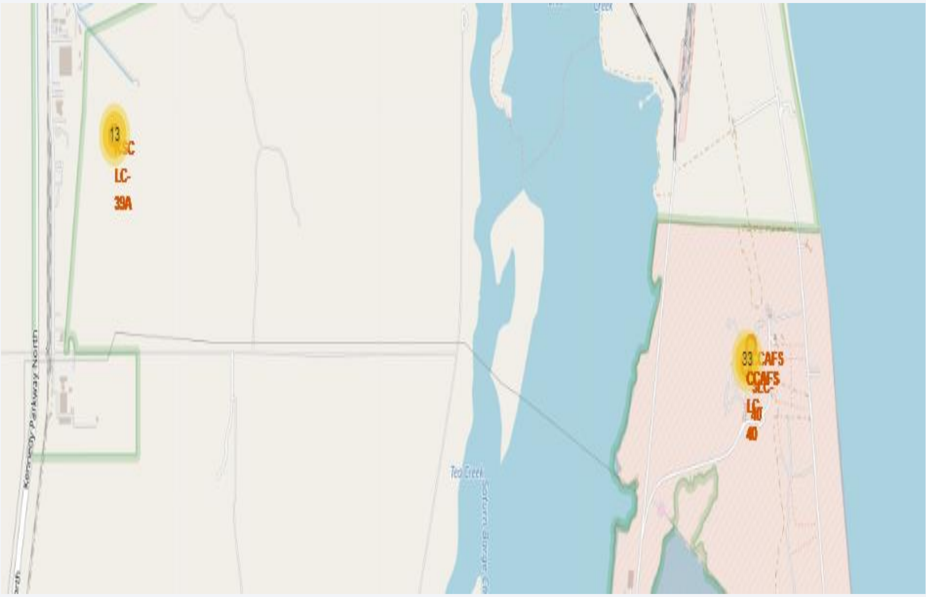
Build an Interactive Map with Folium

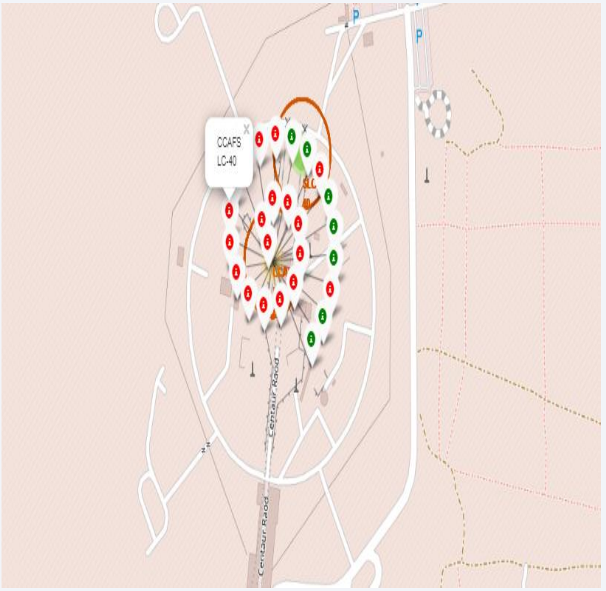
- The launch success rate may depend on many factors such as payload mass, orbit type, and so on. It may also depend on the location and proximities of a launch site, i.e., the initial position of rocket trajectories. Finding an optimal location for building a launch site certainly involves many factors and hopefully it could discover some of the factors by analyzing the existing launch site locations.

- I used Map with Folium to visualize and mark all the launch sites on the map, then I marked the success/failed launches for each site on the map.

- GitHub URL : https://github.com/bnbbdz/Data_sience_ibm_course_capstone/blob/master/02_Notebook/lab_jupyter_launch_site_location.ipynb



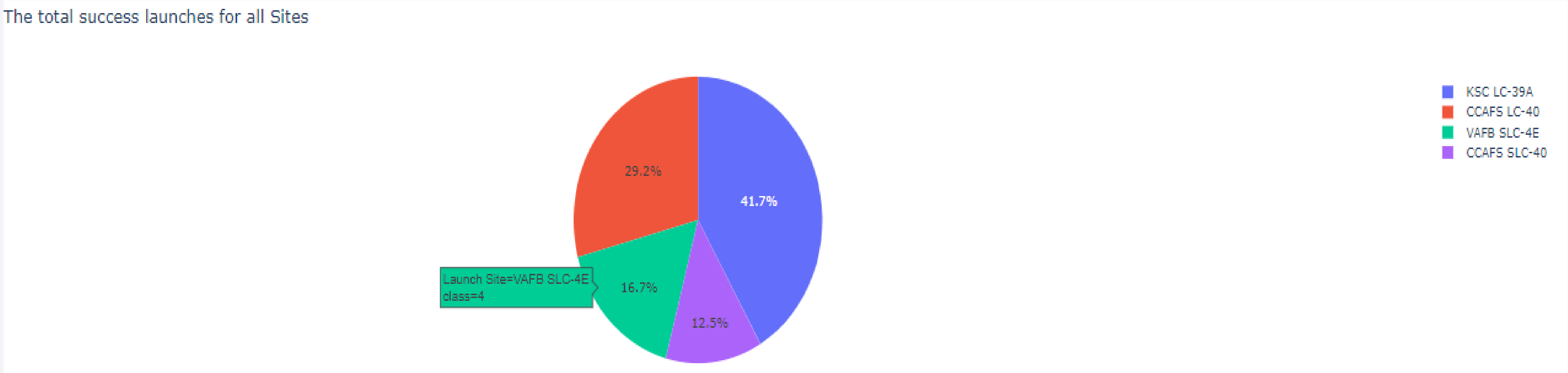


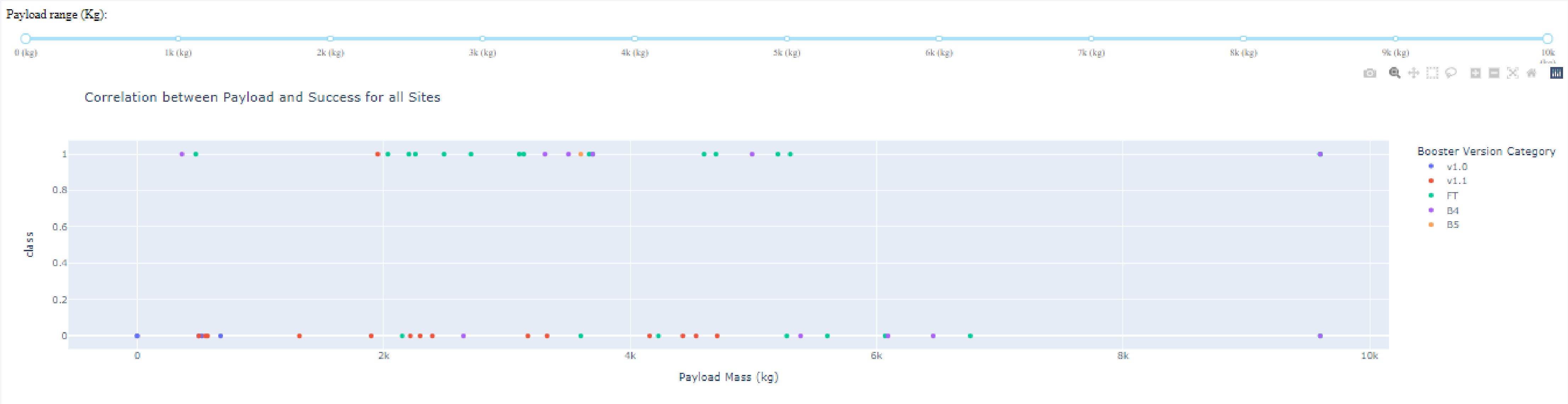


Build a Dashboard with Plotly Dash

- I created plots and interactions to have more understanding about the dataset through answering some questions
    1. Which site has the largest successful launches?
    2. Which site has the highest launch success rate?
    3. Which payload range(s) has the highest launch success rate?
    4. Which payload range(s) has the lowest launch success rate?
    5. Which F9 Booster version (v1.0, v1.1, FT, B4, B5, etc.) has the highest
- I used a pie chart to check the proportion of each site in the total success launches for all sites and the proportion of success or failed launches in each site. And using the scatter plot to examine the changes in the correlation between Payload and Success in each site when I change the Payload
- GitHub URL : https://github.com/bnbbdz/Data_sience_ibm_course_capstone/blob/master/03_Py_file/spacex_dash_app.py

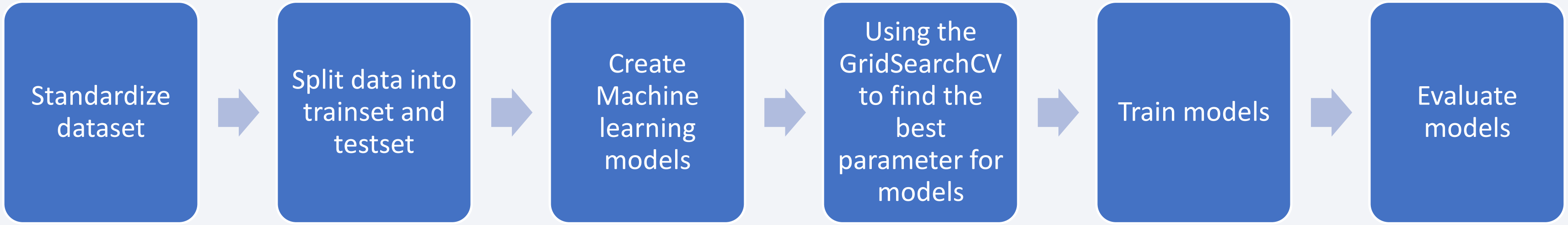
Build a Dashboard with Plotly Dash





Predictive Analysis (Classification)

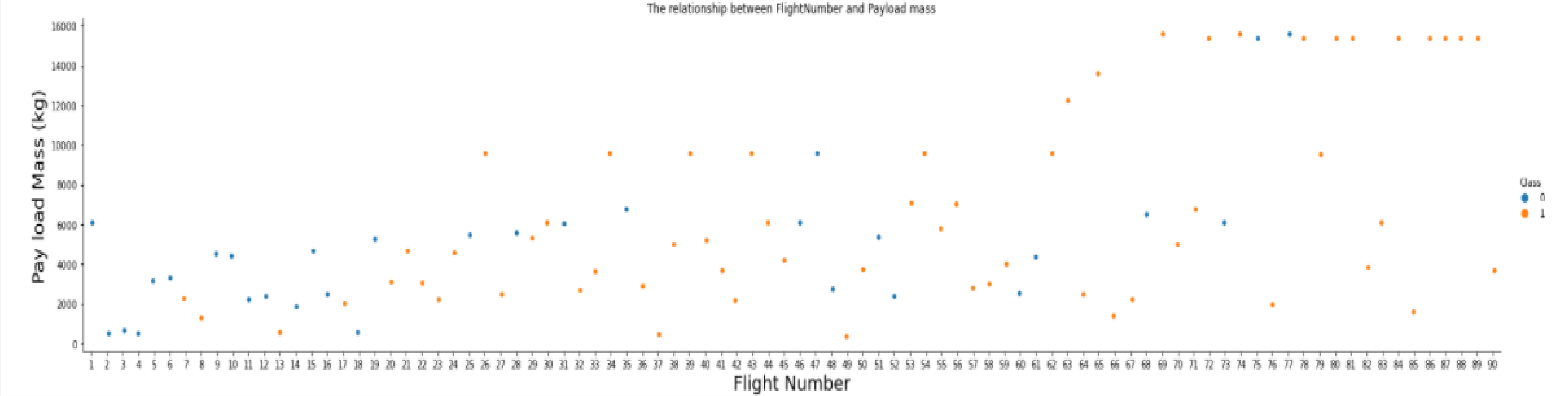
- I used the sklearn library to do some of methodology approaches to figure out the best fit model predict landing results such as Logistic Regression, Support Vector Machine, Decision Tree classification, K Nearest Neighbors classification. Standardize the data and split to trainset-testset then using GridSearchCV to tunning model. As the result, the Decision Tree classification give the best performance in these models with my data set

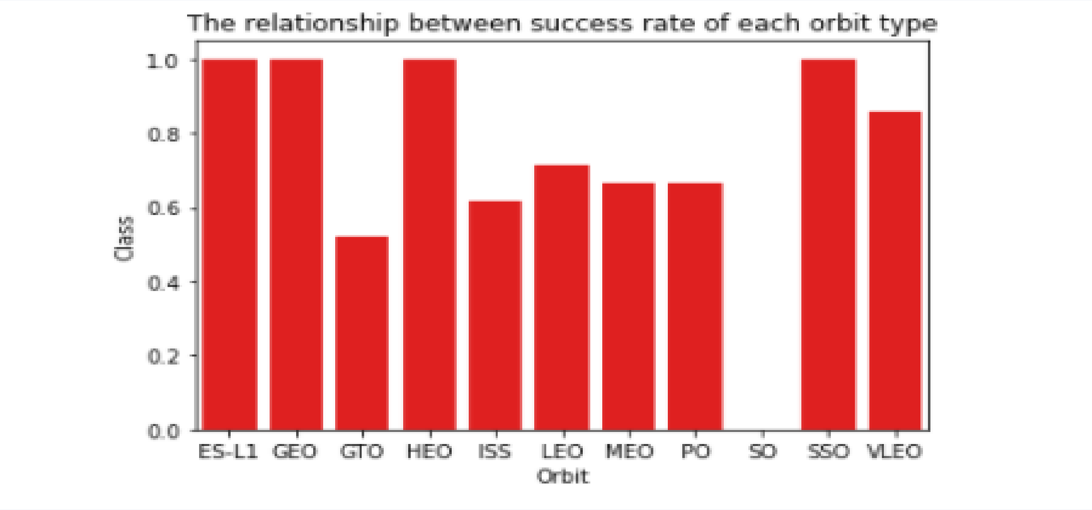


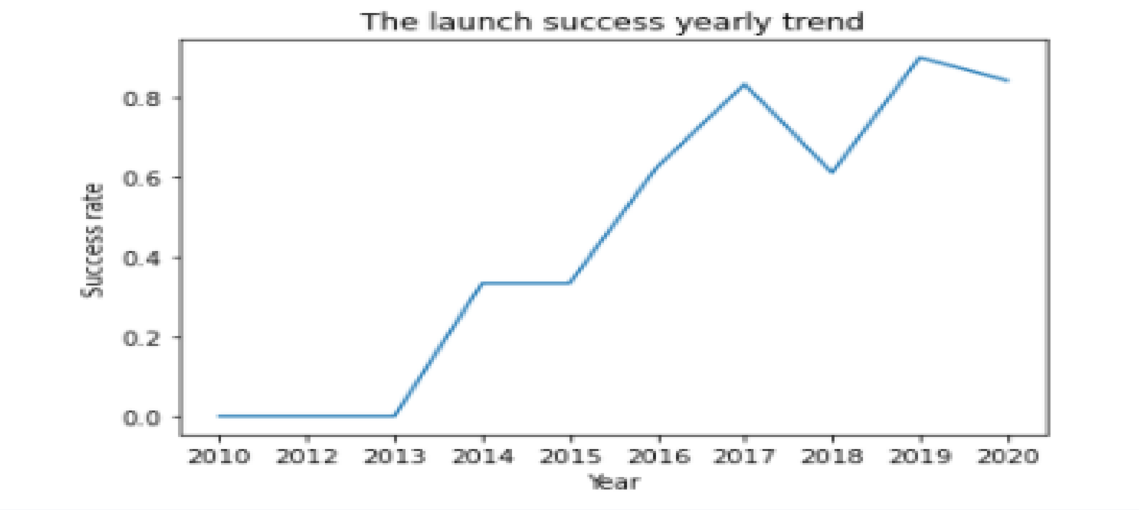
GitHub URL :
https://github.com/bnbbdz/Data_sience_ibm_course_capstone/blob/master/02_Notebook/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results

- Exploratory data analysis results

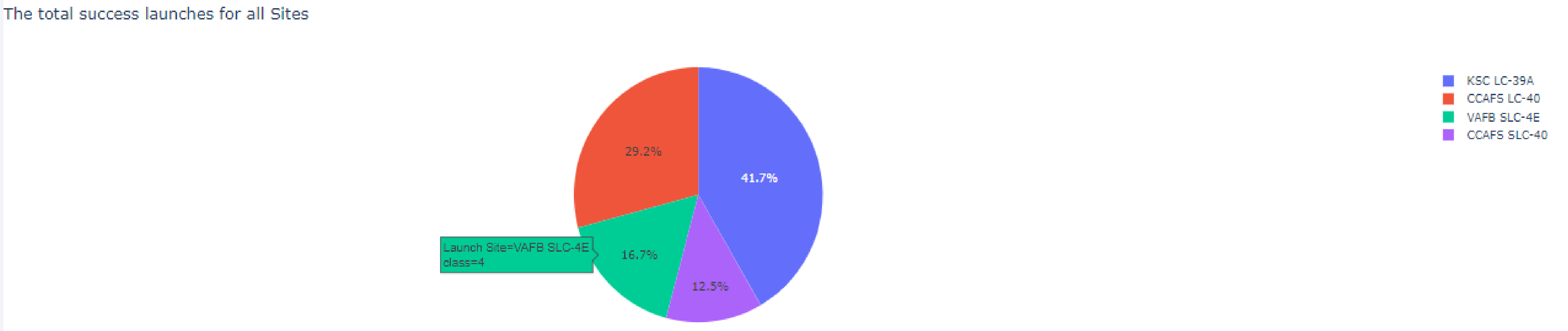


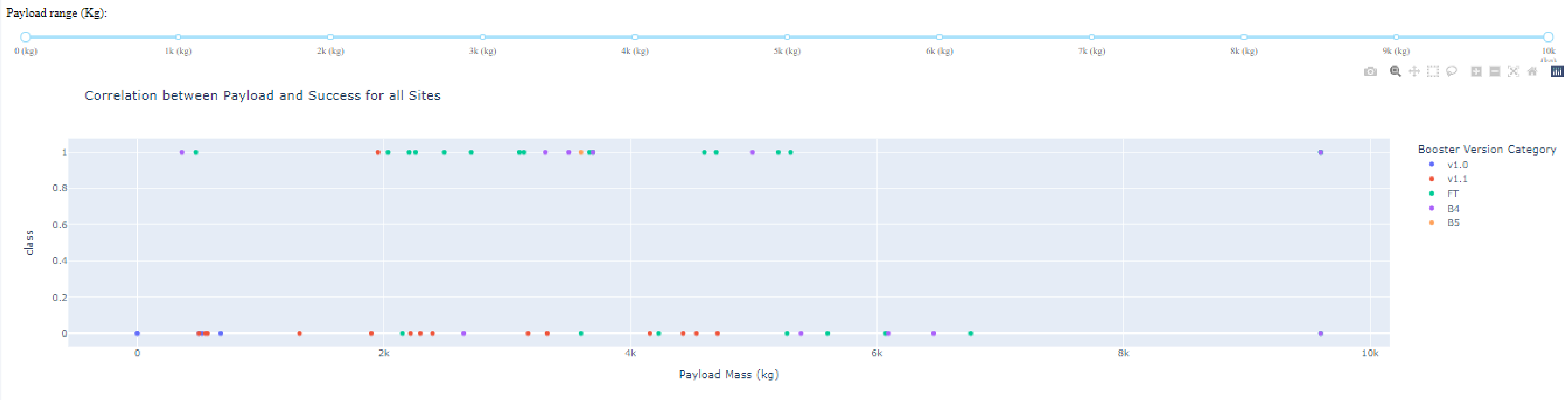




Results

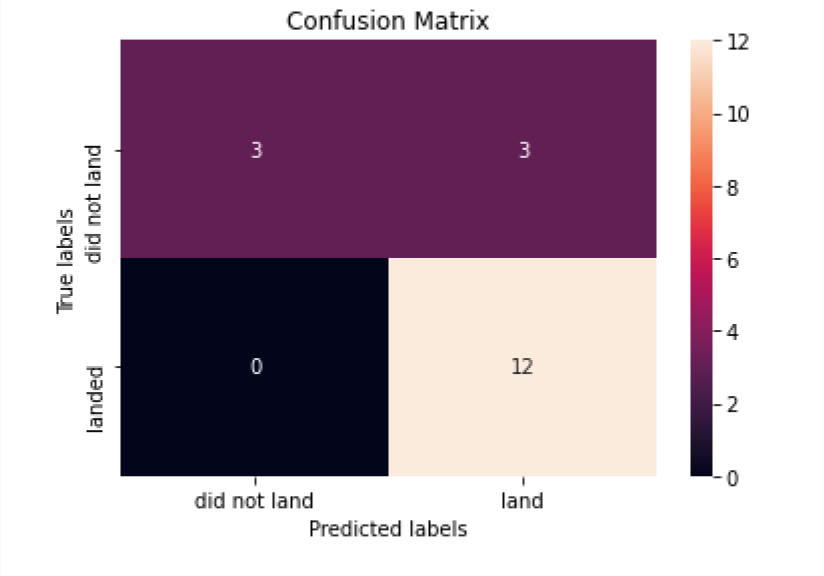
Interactive analytics demo in screenshots

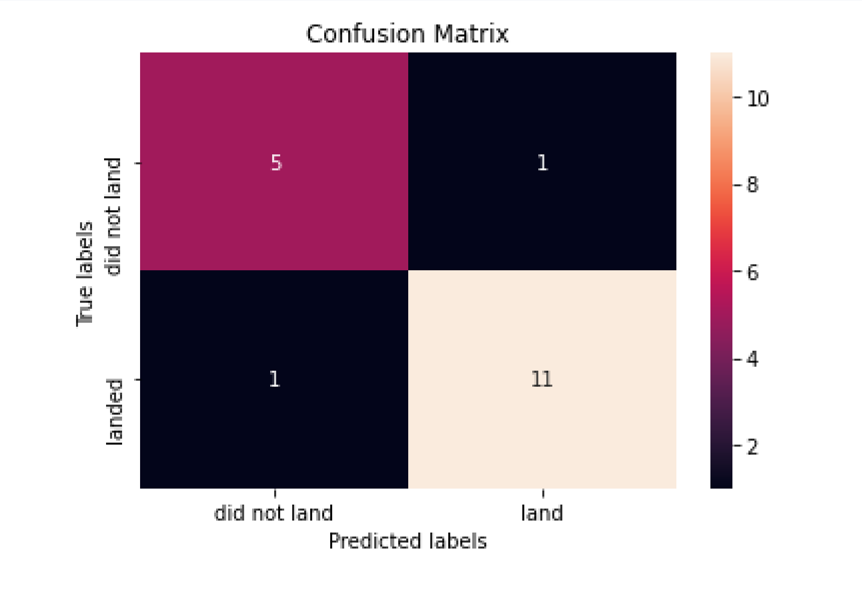




Results

Predictive analysis results

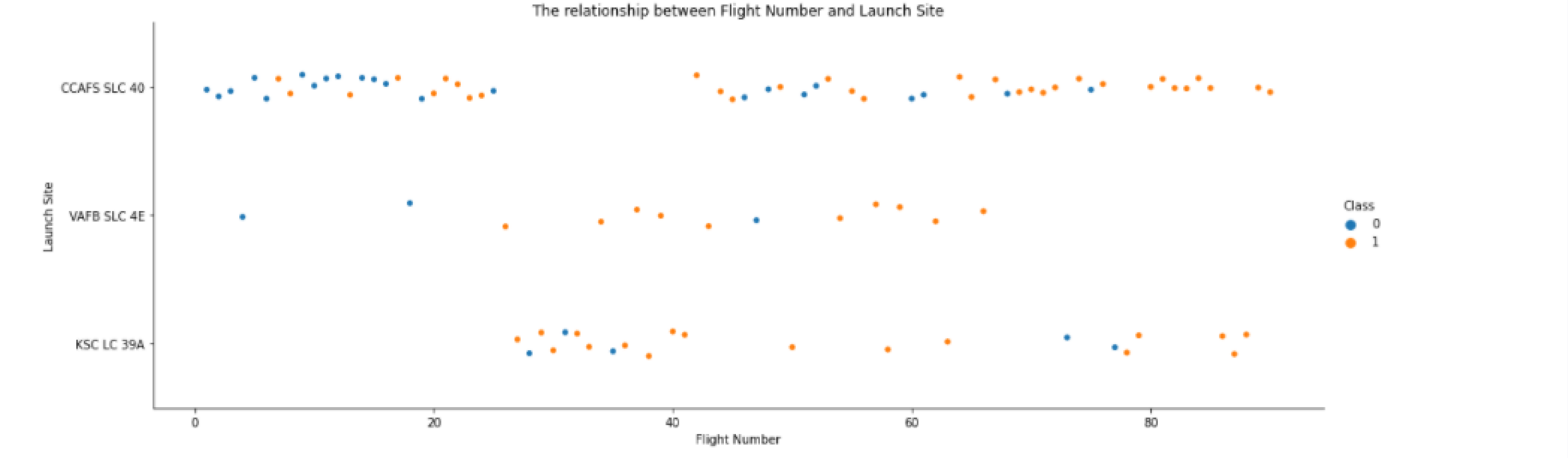




Section 2

Insights drawn from EDA

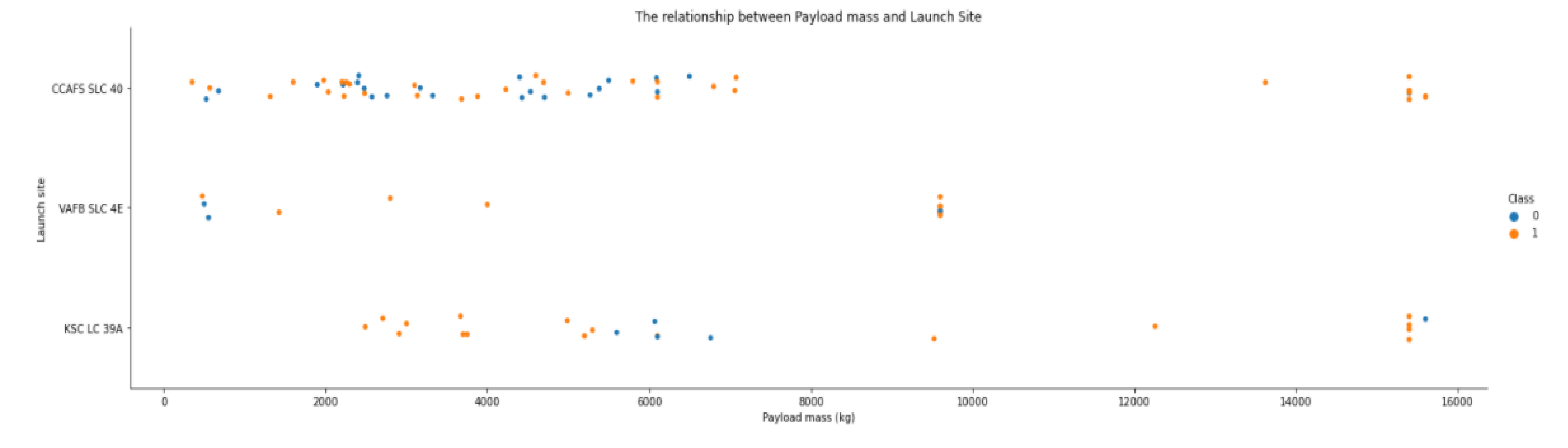
Flight Number vs. Launch Site



Scatter plot of Flight Number vs. Launch Site

We see that as the flight number increases, the first stage is more likely to land successfully for all sites. The Launch site is also important; KSC LC-39A and VAFB SLC 4E have higher success rates than CCAFS LC-40.

Payload vs. Launch Site

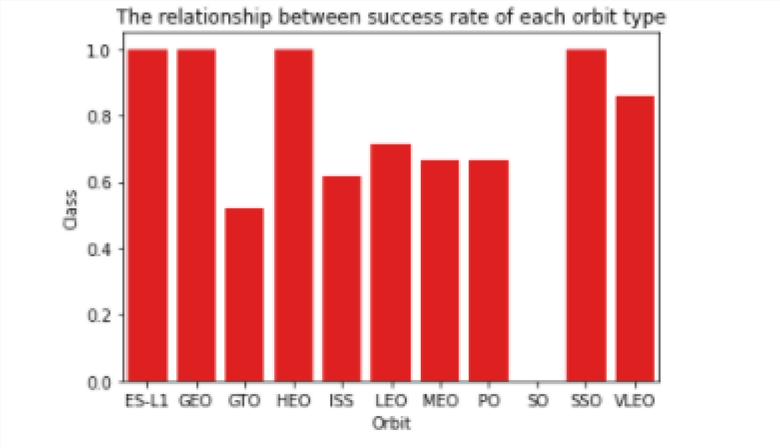


Scatter plot of Payload vs. Launch Site

We see that as the Payload increases, the first stage is more likely to land successfully for all sites.
But when we see the Payload mass between 0 and 8000 (kg), there is a different trend between the launch sites:

- CCAFS SLC 40: there is no difference in the Payload mass
- VAFB SLC 4E: The Payload mass increases, the success rate increases respectively
- KSC LC 39A: The Payload mass increases, the success rate decreases respectively

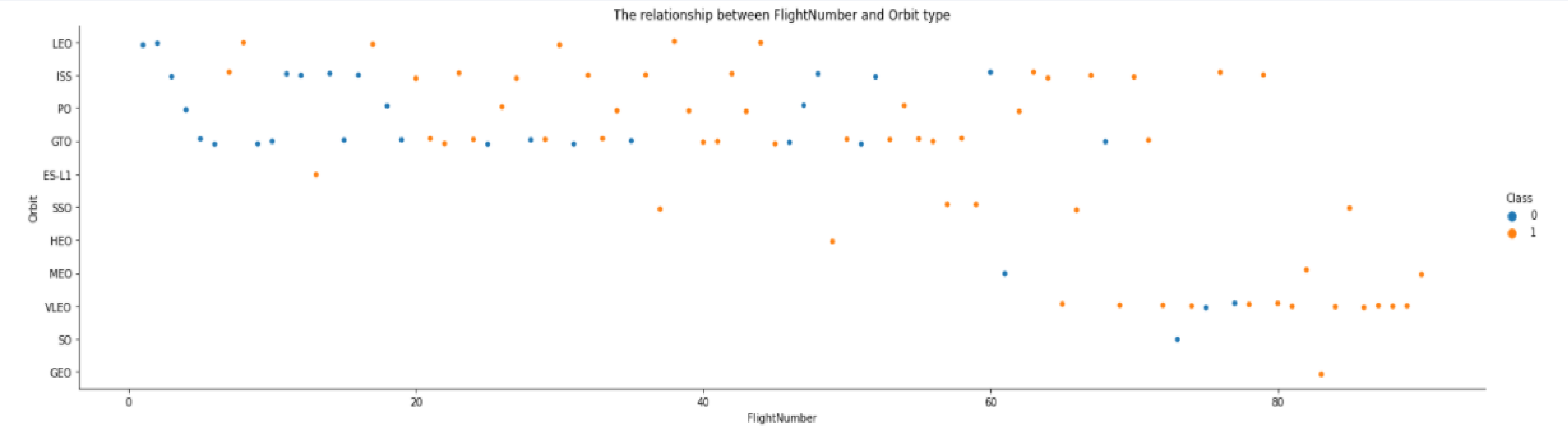
Success Rate vs. Orbit Type



Bar chart for the success rate of each orbit type

We see that ES-L1, GEO, HEO, SSO orbit types have the highest success rate achieve nearly 100%, while SO orbit type has the lowest success rate, and other orbit types have the success rate between 50% and 80%.

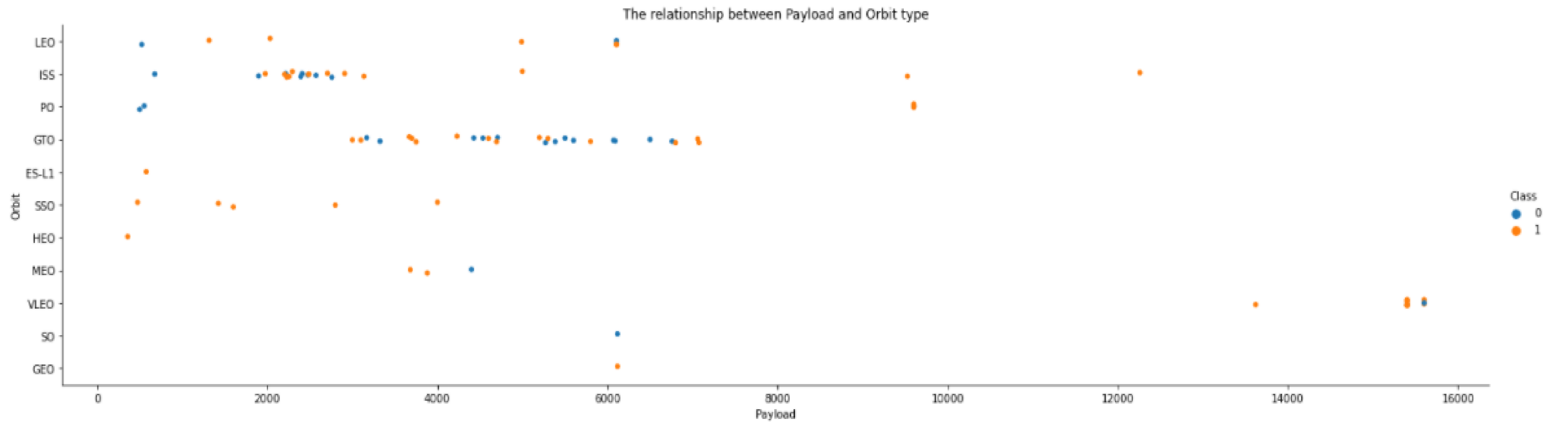
Flight Number vs. Orbit Type



Scatter plot of Flight number vs. Orbit type

We see that as the Flight number increases, the first stage is more likely to land successfully for all Orbit types.
But the SO orbit type and GTO have the lowest success rate, however GTO success rate is gradually improved

Payload vs. Orbit Type

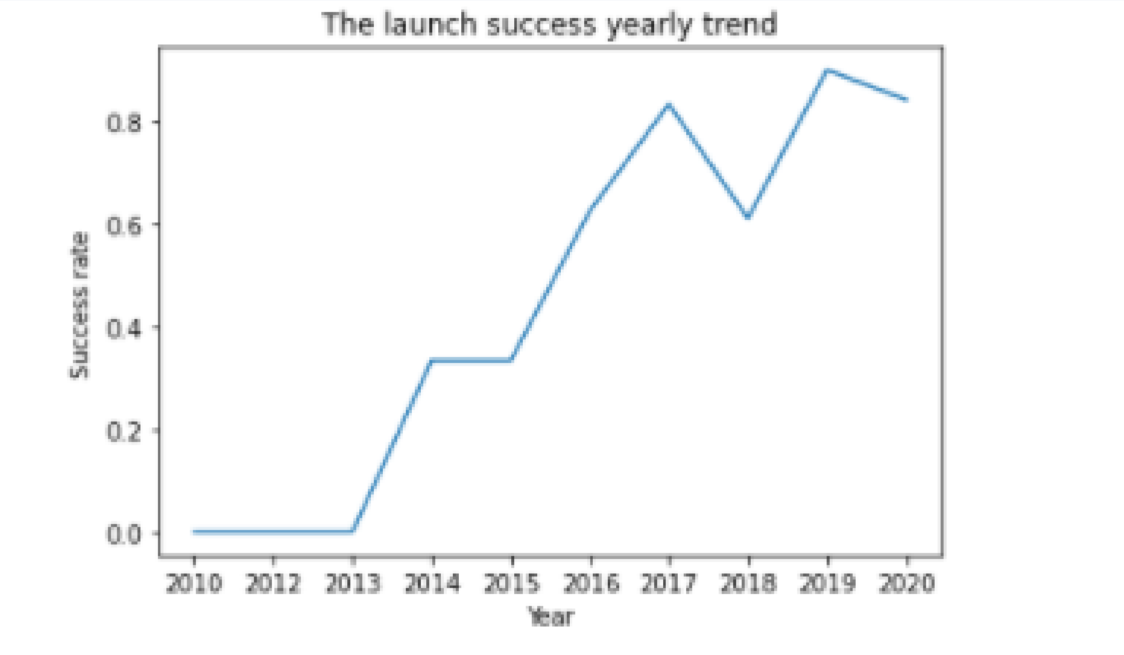


Scatter plot of Payload vs. Orbit type

We see that as the Payload increases, the first stage is more likely to land successfully for almost Orbit types.
But there is a different trend between Orbit types:

- GTO: there is no difference in the Payload mass
- MEO, LEO and VLEO: The Payload mass increases, the success rate decreases respectively

Launch Success Yearly Trend



Line chart of yearly average success rate

We see that as the success rate increase over the year. but 2018 and 2020 have a slight decrease in the success rate

All Launch Site Names

Using distinct to display the names of the unique launch sites. There is 4 launch sites in the data set

```
%%sql
select distinct launch_site from spacextbl
```

```
 * ibm_db_sa://vmp47428:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:31321/BLUDB
Done.
```

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

Launch Site Names Begin with 'CCA'

Using limit 5 to display 5 records where launch sites begin with `CCA`

```
%%sql
select * from spacextbl
where launch_site like 'CCA%'
limit 5
```

```
 * ibm_db_sa://vmp47428:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:31321/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

Total Payload Mass

Using sum() function and where clause to calculate the total payload carried by boosters from NASA. The total Payload from NASA is 45596 kg

```
%sql select sum(payload_mass__kg_) total_payload_mass_kg from spacextbl where customer = 'NASA (CRS)'
```

```
 * ibm_db_sa://vmp47428:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:31321/BLUDB
Done.
```

total_payload_mass_kg
45596

Average Payload Mass by F9 v1.1

Using avg() function alculate the average payload mass carried by booster version F9 v1.1. The result is 2534 kg

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

```
 * ibm_db_sa://vmp47428:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:31321/BLUDB
Done.
```

avg_payload_mass_kg
2534

First Successful Ground Landing Date

Using min() function to find the dates of the first successful landing outcome on ground pad. The first date is 22/12/2015

```
%%sql
select min(DATE) first_date_successful_landing
from spacextbl
where landing__outcome = 'Success (ground pad)'
```

```
 * ibm_db_sa://vmp47428:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:31321/BLUDB
Done.
```

first_date_successful_landing
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- Using between in where clause to list the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000

```
%%sql
select booster_version
from spacextbl
where landing__outcome = 'Success (drone ship)'
and payload_mass__kg_ between 4000 and 6000
```

```
 * ibm_db_sa://vmp47428:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:31321/BLUDB
Done.
```

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

Total Number of Successful and Failure Mission Outcomes

Using group by and count to calculate the total number of successful and failure mission outcomes

```
%%sql
select mission_outcome, count(1) no_outcome
from spacextbl
group by mission_outcome
```

```
 * ibm_db_sa://vmp47428:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:31321/BLUDB
Done.
```

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

Boosters Carried Maximum Payload

Using max() function and subquery to list the names of the booster which have carried the maximum payload mass

```
%%sql
select booster_version
from spacextbl
where payload_mass__kg_ = (select max(payload_mass__kg_) from spacextbl)
```

```
 * ibm_db_sa://vmp47428:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:31321/BLUDB
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

2015 Launch Records

Using year() function to extract year to list the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

```
%%sql
select landing__outcome , booster_version ,launch_site
from spacextbl
where year(DATE) = 2015
and landing__outcome = 'Failure (drone ship)'
```

```
 * ibm_db_sa://vmp47428:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:31321/BLUDB
Done.
```

landing__outcome	booster_version	launch_site
Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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

Using group by, count() function and order by to 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(1) no_landing_outcome
from spacextbl
group by landing__outcome
order by count(1) desc
```

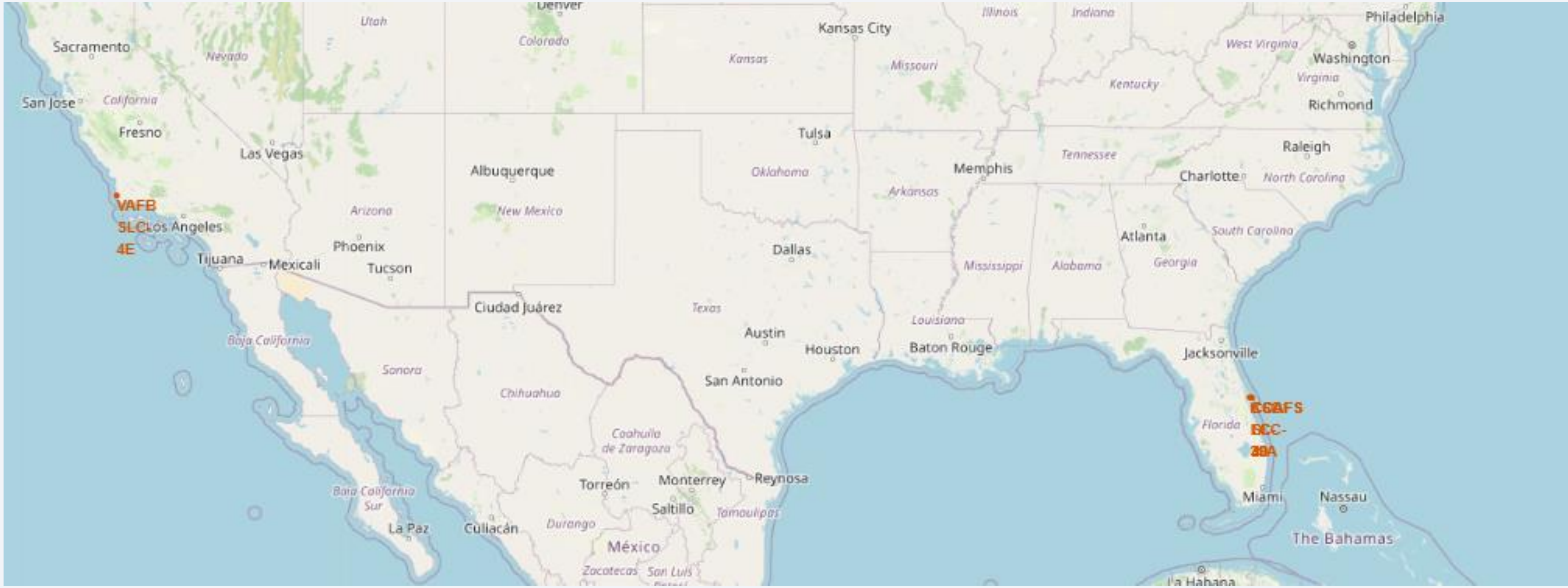
```
 * ibm_db_sa://vmp47428:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:31321/BLUDB
Done.
```

landing__outcome	no_landing_outcome
Success	38
No attempt	22
Success (drone ship)	14
Success (ground pad)	9
Controlled (ocean)	5
Failure (drone ship)	5
Failure	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

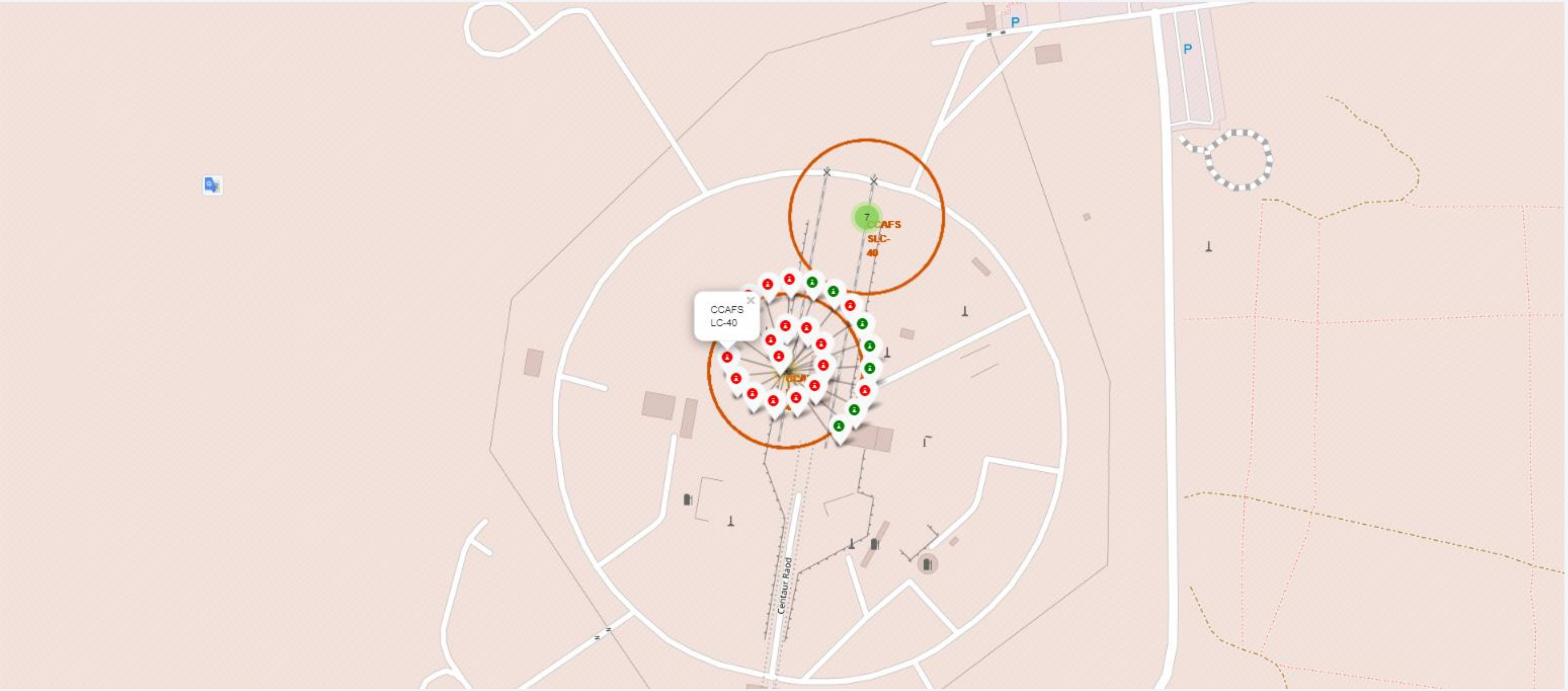
Section 4

Launch Sites Proximities Analysis

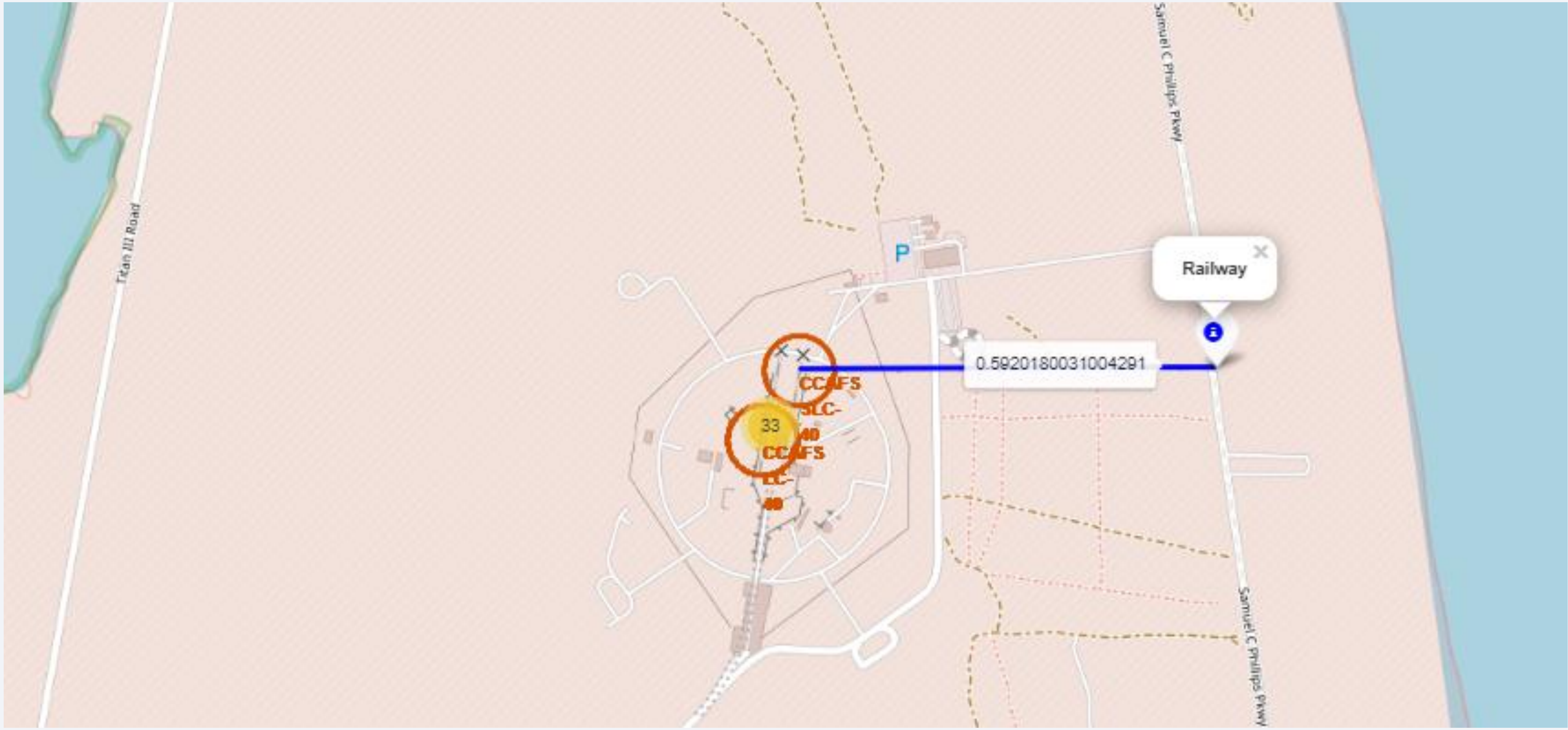
Launch sites location



Outcomes in each Launch sites location



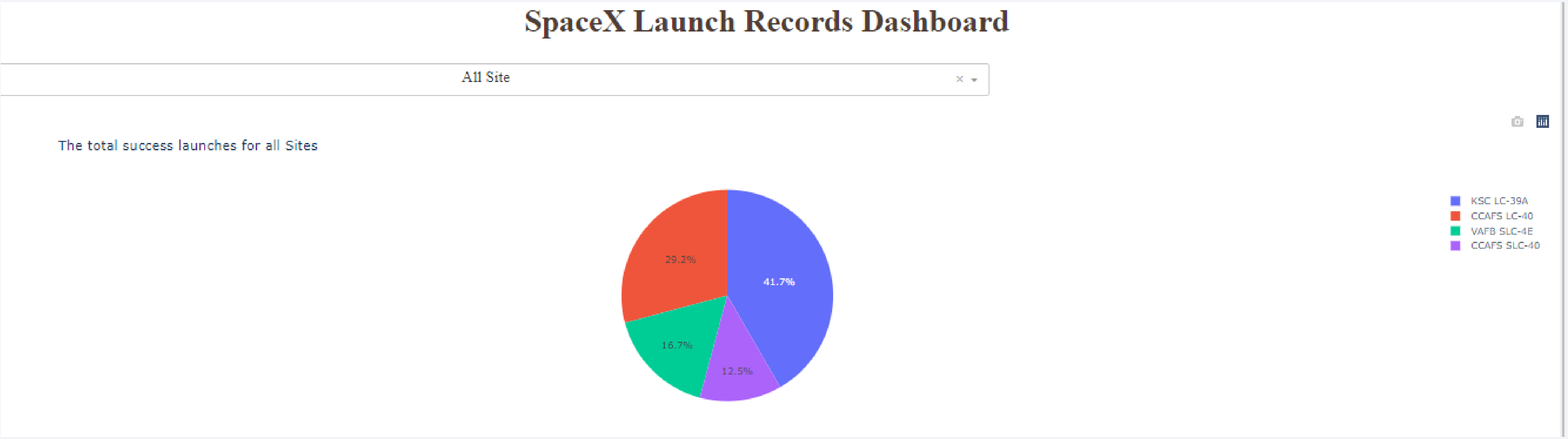
Distance between launch site to its proximities railway



Section 5

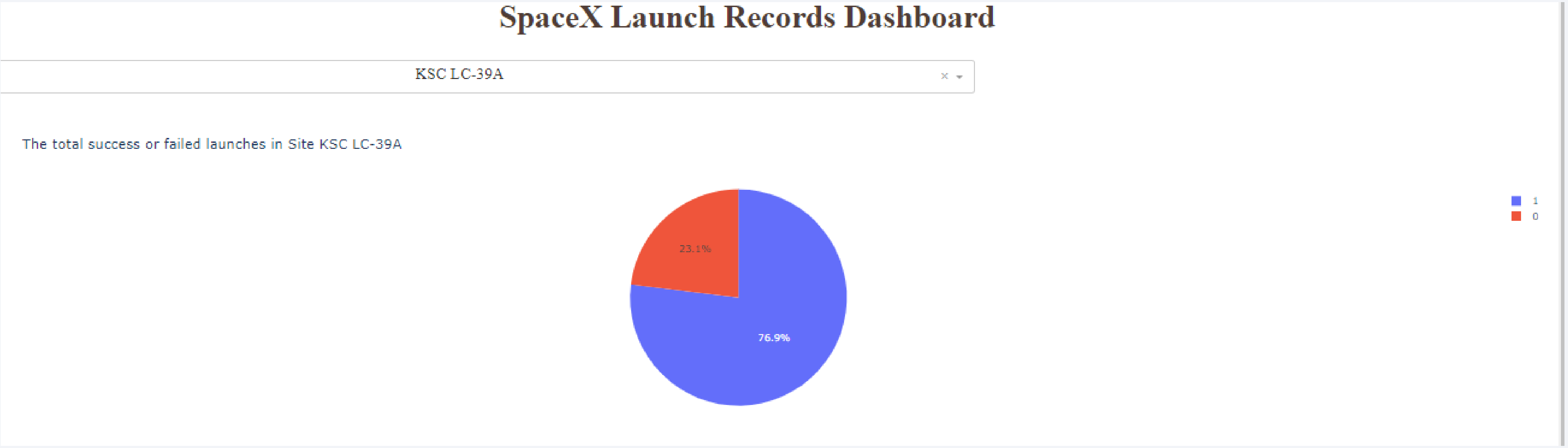
Build a Dashboard with Plotly Dash

Success rate for all sites



This pie chart to check the proportion of each site in the total success launches for all sites. As we can see, KSC LC-39A has the highest proportion in the total of success launches.

The ratio of success/failed in the site KSC LC-39A



KSC LC-39A has the highest proportion in the total of success launches. Also it has the highest success rate is 76.9%

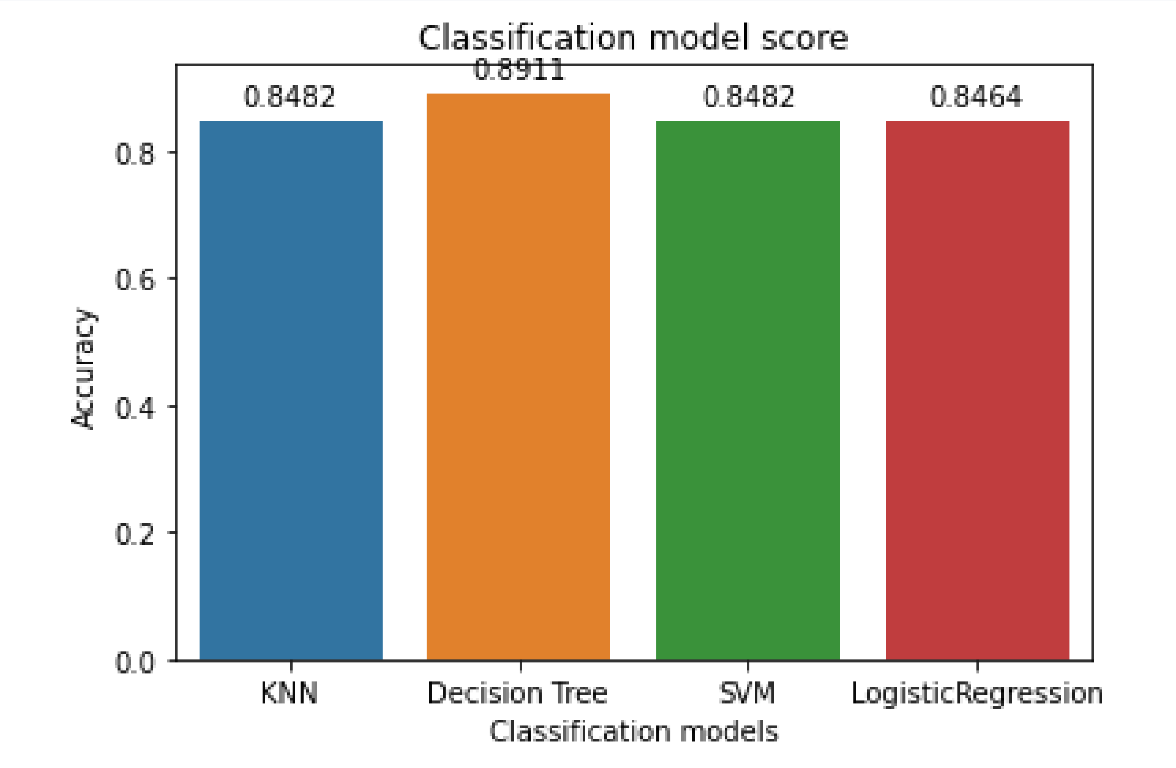
Correlation between Payload and Success for all sites

As we can see on the Payload range between 1000 kg and 8000 kg, Booster version B5 has the highest success rate, after that is FT and B4 version. And v1.1 version has lowest success rate. And as the Payload increase, the percentage of a success rate decrease

Section 6

Predictive Analysis (Classification)

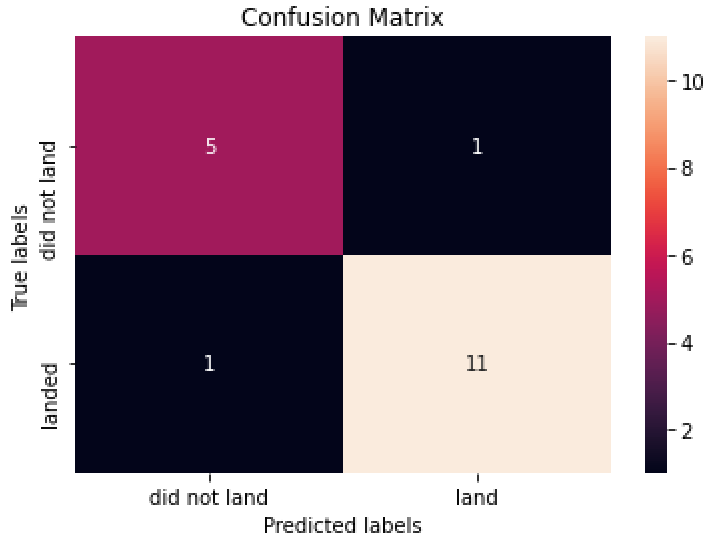
Classification Accuracy



As we can see on the bar chart above, The Decision tree model has the highest accuracy with a score is 0.8911. Therefore, the Decision tree model is the best performing model in 4 models that were trained.

Confusion Matrix

```
yhat = tree_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
```




```
from sklearn.metrics import classification_report
print(classification_report(Y_test,tree_cv.predict(X_test)))
```

```
              precision    recall  f1-score   support

           0       0.83      0.83      0.83         6
           1       0.92      0.92      0.92        12

    accuracy                           0.89        18
   macro avg       0.88      0.88      0.88        18
weighted avg       0.89      0.89      0.89        18
```

As we can see on the onfusion matrix of the Decision tree model above, there is 16 cases were predicted correctly and 2 cases were given wrong (1 case Type 1 error and 1 case Type 2 error).

In first row, the actual “did not land” is 6 cases, in test set predict “did not land” is 5 cases, predict “land” is 1 case, this mean the classifier predict correctly 5/6 cases

In second row, the actual “land” is 12 cases, in test set predict “did not land” is 1 cases, predict “land” is 11 case, this mean the classifier predict correctly 11/12 cases

Conclusions

SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upwards of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage. Through the processing such as Data collection, data wrangling, exploratory data analysis (EDA) using visualization and SQL, interactive visual analytics, and predict if the Falcon 9 first stage will land successfully or not using classification models like KNN, Decision tree, SVM, Logistic regression,...

Throughout the EDA process, I found out that there are various features that are correlated with the Landing successfully such as : the Orbit type, the Payload mass, The flight number, Launch site, etc.

And I do a lot of ML models and evaluated these models to find the best-performing model. As the result, I discovered that the Decision tree model fit with the dataset I was collected and give 89,11% accuracy.

Appendix

- Capstone github repository: https://github.com/bnbbdz/Data_sience_ibm_course_capstone.git

Thank you!