IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Karen J. Castro Serrato
October 20 / 2021

Outline

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

Executive Summary

For predict if the Falcon 9 first stage will land successfully we need to develop a series of methodologies, to obtain efficient and consistent results with expectations. In this project, I will provide with an overview of the problem and the tools and methodologies that will give solutions to our problem, such as:

- 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

The interaction between various variables can lead to a successful or unsuccessful landing of the Falcon 9. Each variable can be analyzed independently or classified together with other variables. The results will depend on how they are evaluated by our models. Especially variables like payload-mass are directly related to the successful landings of the first stage of the Flacon 9.

Introduction

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 it is can determine if the first stage will land, so it will be possible to determine the cost of a launch. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch.

Several examples of an unsuccessful and successful landing are shown in this project. Most unsuccessful landings are planned. SpaceX performs a controlled landing in the oceans.

The principal objective is predict if the Falcon 9 first stage will land successfully. If the first stage of the Falcon 9 manages to land successfully, it would mean significant cost savings.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:

  SpaceX has an API where the information of all the launches of its rockets is. The SpaceX REST API endpoints, or URL, starts with api.spacexdata.com/v4/. Make a call to the API, through the GET request function and add the obtained data to a list. This information can be converted into a Dataframe.

- Perform data wrangling

  The data must be analyzed both on its label and in its content. Initially each label can be reviewed and the respective modifications made, once the labels are ready is possible to analyze the data contained in each column and make the necessary adjustments.

- Perform exploratory data analysis (EDA) using visualization and SQL

Methodology

Executive Summary

- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - The first step is to build a machine learning pipeline to predict if the first stage of the Falcon 9 lands successfully. This will include: Preprocessing, allowing us to standardize our data, and Train test split, allowing us to split our data into training and testing data, the model need to be train and perform Grid Search, allowing us to find the hyperparameters that allow a given algorithm to perform best. Using the best hyperparameter values, it is possible to determine the model with the best accuracy using the training data. test Logistic Regression, Support Vector machines, Decision Tree Classifier, and K-nearest neighbors. Finally, the confusion matrix is output.

Data Collection

- The first step is work with the endpoint api.spacexdata.com/v4/launches/past. That URL target a specific endpoint of the API to get past launch data. Next, perform a get request using the requests library to obtain the launch data, which will be use to get the data from the API. This result can be viewed by calling the .json() method. The response will be in the form of a JSON, specifically a list of JSON objects. The get is a response in a form JSON. Specifically, a list of JSON objects which each represent a launch. To convert this JSON to a dataframe, we can use the json_normalize function. This function will allow us to “normalize” the structured json data into a flat table or dataframe.

Data Collection – SpaceX API

- Request and parse the SpaceX launch data using the GET request

- Github Link: https://github.com/karencastroserrato/Applied-Data-Science-Capstone-Project/blob/main/Data%20Collection%20API%20Lab.ipynb

```python
# Get the head of the dataframe
data.head()
```

	static_fire_date_utc	static_fire_date_unix	net	window	rocket	success	failures	details	crew	ships	capsules	payloads
0	2006-03-17T00:00:00.000Z	1.142554e+09	False	0.0	5e9d0d95eda69955f709d1eb	False	[{'time': 33, 'altitude': None, 'reason': 'merlin engine failure'}]	Engine failure at 33 seconds and loss of vehicle	[]	[]	[]	[5eb0e4b5b
1	None	NaN	False	0.0	5e9d0d95eda69955f709d1eb	False	[{'time': 301, 'altitude': 289, 'reason': 'harmonic oscillation leading to premature engine shutdown'}]	Successful first stage burn and transition to second stage, maximum altitude 289 km, Premature engine shutdown at T+7 min 30 s, Failed to reach orbit, Failed to recover first stage	[]	[]	[]	[5eb0e4b6b
							[{'time': 140, 'altitude': 35,	Residual stage 1				

Data Collection - Scraping

- Dataframe created with data obtained through web scraping

- Github Link: https://github.com/karencastroserrato/Applied-Data-Science-Capstone-Project/blob/main/Data%20Collection%20with%20Web%20Scraping.ipynb

After you have fill in the parsed launch record values into launch_dict, you can create a dataframe from it.

```
#df=pd.DataFrame(launch_dict)
df= pd.DataFrame({ key:pd.Series(value) for key, value in launch_dict.items() })
df
```

	Flight No.	Launch site	Payload	Payload mass	Orbit	Customer	Launch outcome	Version Booster	Booster landing	Date	Time
0	NaN	NaN	payload	payload_mass	orbit	customer	launch_outcome	bv	booster_landing	date	time
1	NaN	NaN	payload	payload_mass	orbit	customer	launch_outcome	bv	booster_landing	date	time
2	NaN	NaN	payload	payload_mass	orbit	customer	launch_outcome	bv	booster_landing	date	time
3	NaN	NaN	payload	payload_mass	orbit	customer	launch_outcome	bv	booster_landing	date	time
4	NaN	NaN	payload	payload_mass	orbit	customer	launch_outcome	bv	booster_landing	date	time
...	...	...	...	...	...	...	...	...	...	...	...
101	NaN	NaN	payload	payload_mass	orbit	customer	launch_outcome	NaN	booster_landing	date	time
102	NaN	NaN	payload	payload_mass	orbit	customer	launch_outcome	NaN	booster_landing	date	time
103	NaN	NaN	payload	payload_mass	orbit	customer	launch_outcome	NaN	booster_landing	date	time
104	NaN	NaN	payload	payload_mass	orbit	customer	launch_outcome	NaN	booster_landing	date	time
105	NaN	NaN	payload	payload_mass	orbit	NaN	NaN	NaN	NaN	date	time

106 rows × 11 columns

Data Wrangling

- It is important to transform raw data into a clean dataset which provides meaningful data on this project. Nulls and replace data, make filters to remove data are some of the tasks in wrangling data. After that, to further improve the data is good idea to dealt with using one hot encoding.

Data Wrangling

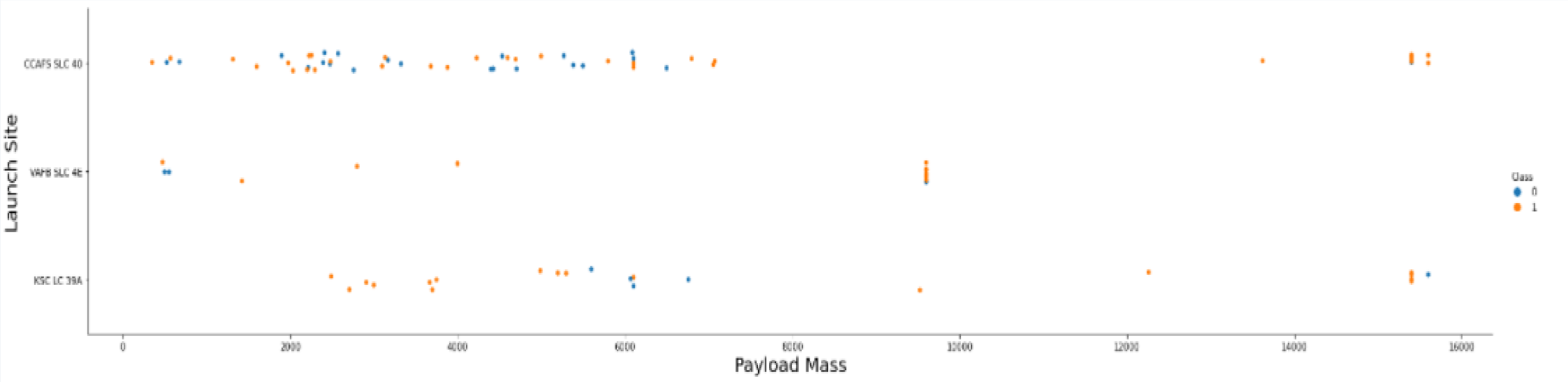
```
df.head(5)
```

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Seri
0	1	2010-06-04	Falcon 9	6104.959412	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B00
1	2	2012-05-22	Falcon 9	525.000000	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B00
2	3	2013-03-01	Falcon 9	677.000000	ISS	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B00
3	4	2013-09-29	Falcon 9	500.000000	PO	VAFB SLC 4E	False Ocean	1	False	False	False	NaN	1.0	0	B10
4	5	2013-12-03	Falcon 9	3170.000000	GTO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B10

Data analysis - wrangling

- Github Link: https://github.com/karencastroserrato/Applied-Data-Science-Capstone-Project/blob/main/Data%20wrangling.ipynb

EDA with Data Visualization



Most of the launches have been carried out with a payload mass no greater than 8000Kg. KSC LC 39A has been the Launch site with the best results so far. Almost all landings with more than 10,000kg have been successful.

- Github Link: https://github.com/karencastroserrato/Applied-Data-Science-Capstone-Project/blob/main/Exploratory%20Data%20Analysis%20with%20Visualization%20.ipynb

;

EDA with SQL

- %sql SELECT unique(launch_site) FROM SPACEXDATASET;
- %sql SELECT launch_site FROM SPACEXDATASET where launch_site LIKE'CCA%' limit 5;
- %sql SELECT customer, SUM(payload_mass__kg_) from SPACEXDATASET WHERE customer='NASA (CRS)' GROUP BY customer;
- %sql SELECT booster_version, AVG(payload_mass__kg_) from SPACEXDATASET where booster_version='F9 v1.1' GROUP BY booster_version;
- %sql select DATE, landing__outcome from SPACEXDATASET where landing__outcome='Success (ground pad)' limit 1;
- %%sql select booster_version, landing__outcome, payload_mass__kg_ from SPACEXDATASET WHERE landing__outcome='Success (drone ship)' and (payload_mass__kg_>4000) and (payload_mass__kg_<6000);

;

EDA with SQL

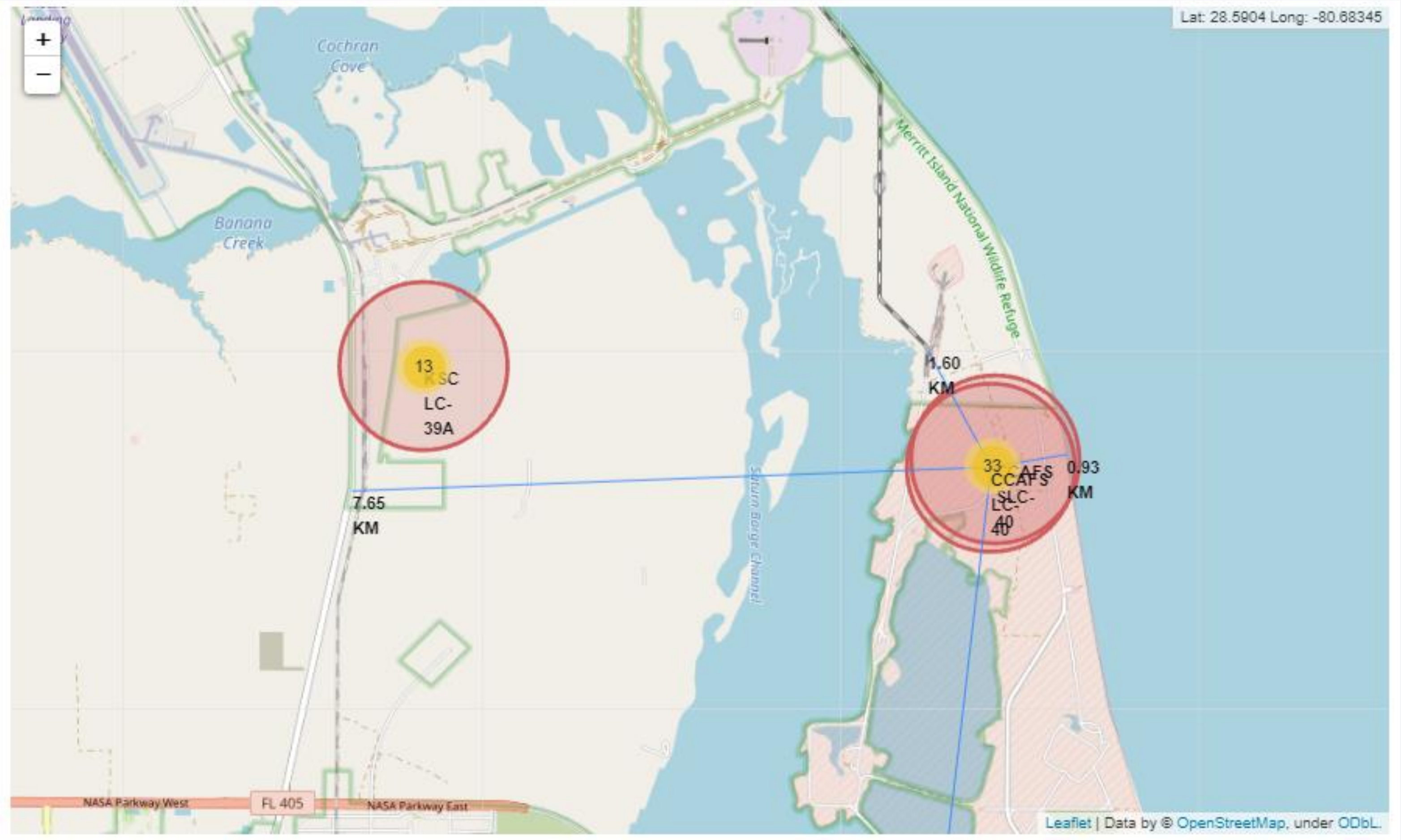
- %sql select mission_outcome, count(mission_outcome) from SPACEXDATASET group by mission_outcome;
- %sql select booster_version, (select max(payload_mass__kg_) as payload from SPACEXDATASET) from SPACEXDATASET;
- %sql select landing__outcome, booster_version, launch_site, DATE from SPACEXDATASET where landing__outcome='Failure (drone ship)' and YEAR(DATE)=2015;
- %%sql select landing__outcome, DATE from SPACEXDATASET WHERE landing__outcome='Failure (drone ship)' or landing__outcome='Success (ground pad)' and (DATE) between '2010-06-04' and '2017-03-20' order by DATE desc;

- Github Link: https://github.com/karencastroserrato/Applied-Data-Science-Capstone-Project/blob/main/Exploratory%20Data%20Analysis%20with%20SQL.ipynb

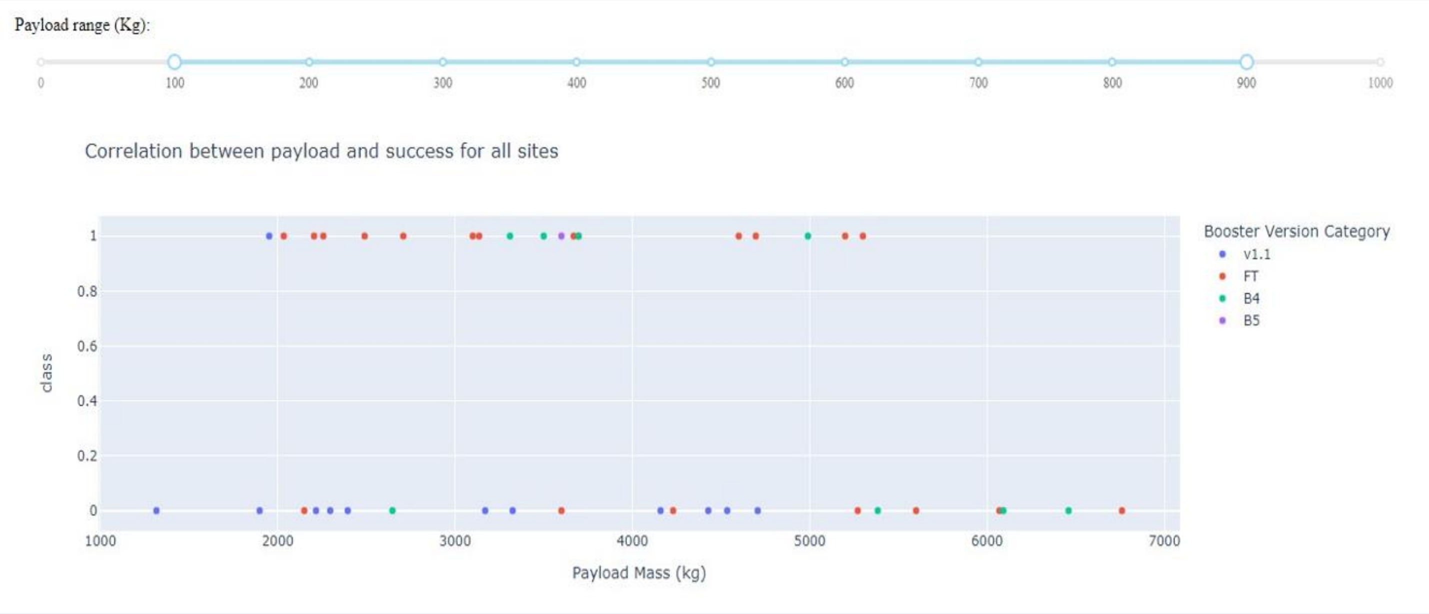
Build an Interactive Map with Folium

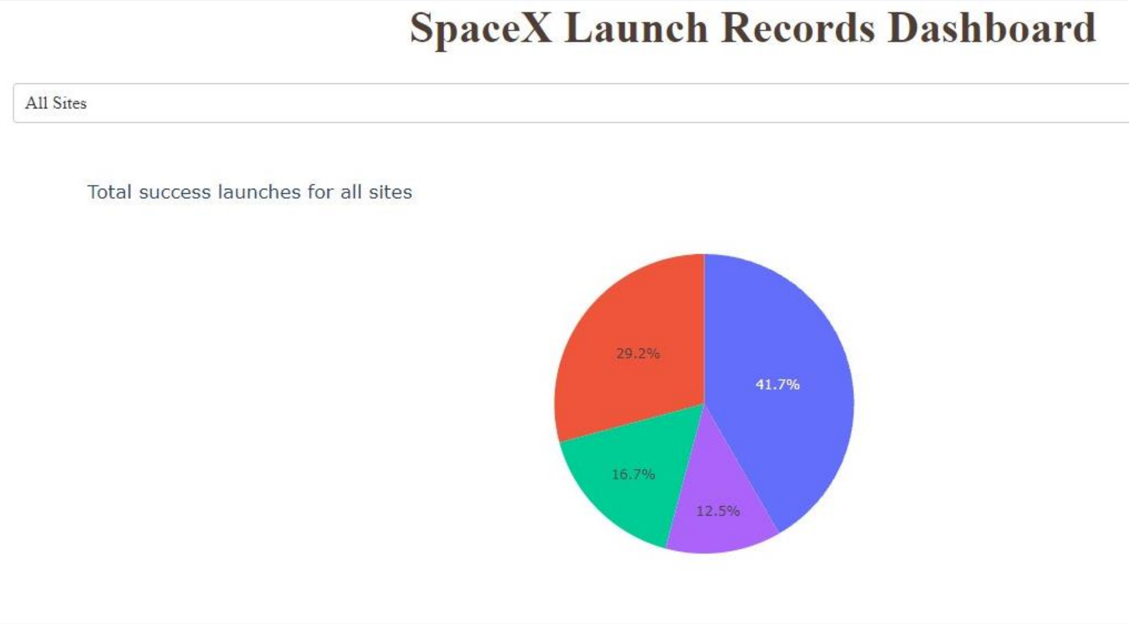
- In the image it is possible to see the distance between CCAFS LC-40 launch site and the nearest highway

- Github Link: https://github.com/karencastroserrato/Applied-Data-Science-Capstone-Project/blob/main/Interactive%20Visual%20Analytics%20with%20Folium%20.ipynb



Build a Dashboard with Plotly Dash





Pie chart shows all sites with a successful landing and the range slide shows the relationship between payload and successful landing for all sites.

- Github Images - Link: https://github.com/karencastroserrato/Applied-Data-Science-Capstone-Project/tree/main/Interactive%20Dashboard%20with%20Ploty%20Dash
- Github Code - Link: https://github.com/karencastroserrato/Applied-Data-Science-Capstone-Project/blob/main/Interactive%20Dashboard%20with%20Ploty%20Dash.py

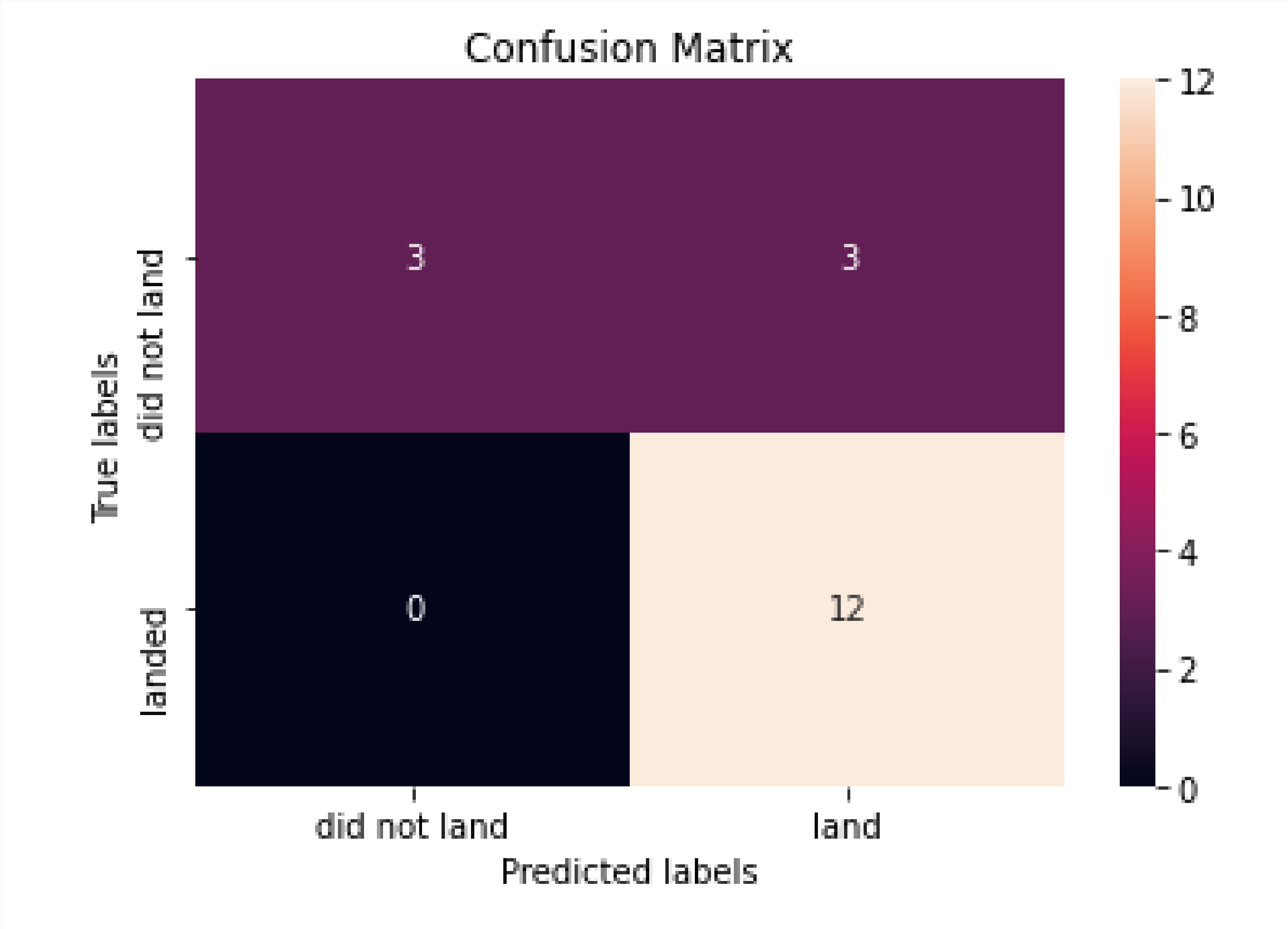
Predictive Analysis (Classification)

The first step is to build a machine learning pipeline to predict if the first stage of the Falcon 9 lands successfully. This will include: Preprocessing, allowing us to standardize our data, and Train test split, allowing us to split our data into training and testing data, the model need to be train and perform Grid Search, allowing us to find the hyperparameters that allow a given algorithm to perform best. Using the best hyperparameter values, it is possible to determine the model with the best accuracy using the training data. test Logistic Regression, Support Vector machines, Decision Tree Classifier, and K-nearest neighbors. Finally, the confusion matrix is output.

Predictive Analysis (Classification)

- A decision tree classifier was one of the best in the project. First use a GridSearchCV object tree_cv with cv = 10. Fit the object to find the best parameters from the dictionary parameters and finally calculate the accuracy of tree_cv on the test data using the method score.

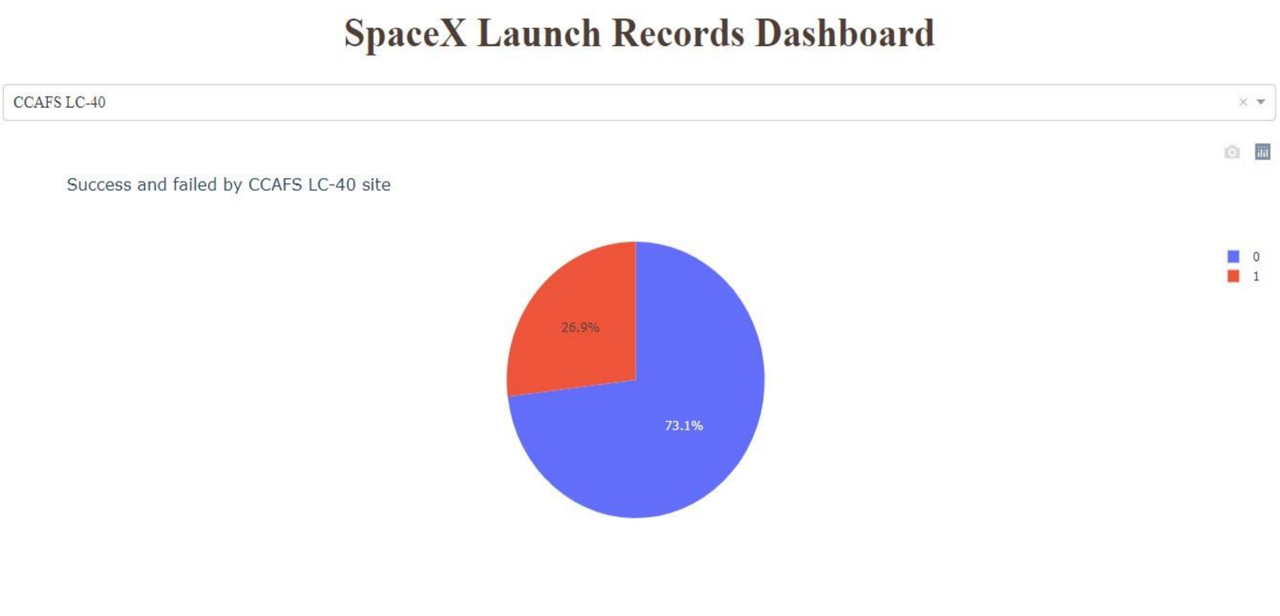
- Github Link: https://github.com/karencastroserrato/Applied-Data-Science-Capstone-Project/blob/main/Machine%20learning%20prediction.ipynb

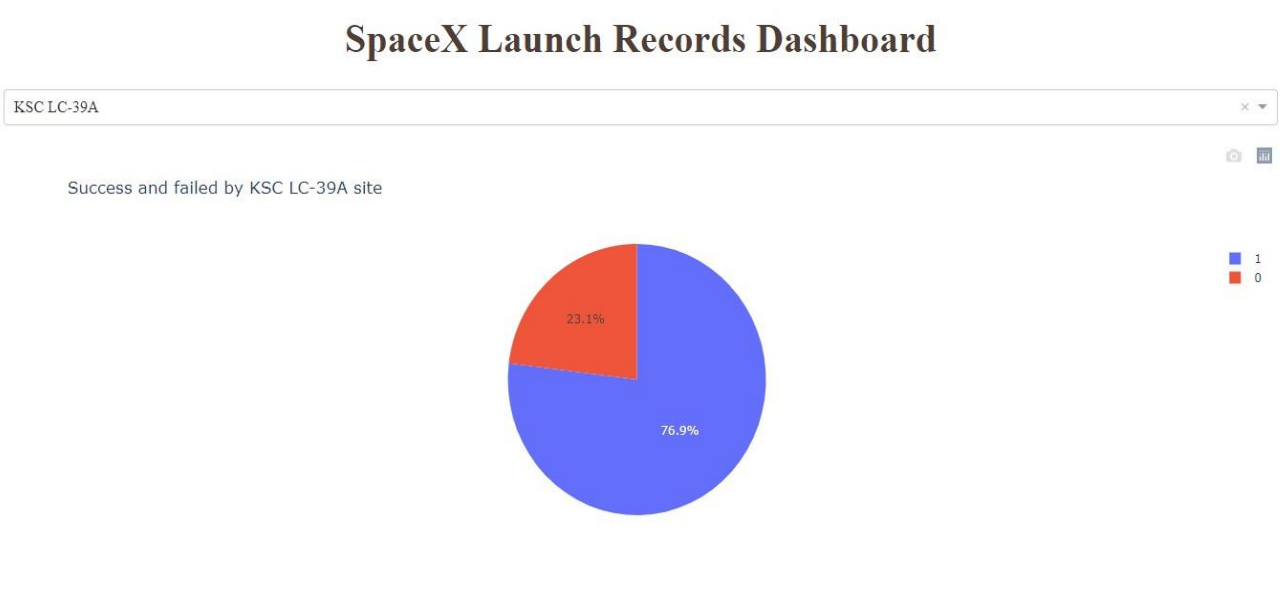


Results

- Different launch sites have different success rates. CCAFS LC-40, has a success rate of 60 %, while KSC LC-39A and VAFB SLC 4E has a success rate of 77%.
- Most of the launches have been carried out with a payload mass no greater than 8000Kg. KSC LC 39A has been the Launch site with the best results so far. Almost all landings with more than 10,000kg have been successful.
- In the LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit. Heavy payloads have a negative influence on GTO orbits and positive on GTO and Polar LEO (ISS) orbits. Success rate since 2013 kept increasing till 2020.
- There is some randomness to it: One thing to keep in mind is that our test is very small. It is only 18. In real situations, we likely won't be using a test size this small. So what is happening is that sometimes a group is split to a leaf that is labeled failure and sometimes it's split to a leaf that is labeled success. For a test sample size this small this is going to have huge effects on our accuracy .16/18 correct is 0.88888 accuracy 15/18 correct is 0.83333 accuracy 14/18 correct would be 0.7777 accuracy This is likely why your two scores are not equal. In fact, if you run any of your code enough times you'll see that the exact same code will produce different scores. If you want the randomness to stay consistent DecisionTree. Classifier takes a random_state argument that you can set. https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html

Results



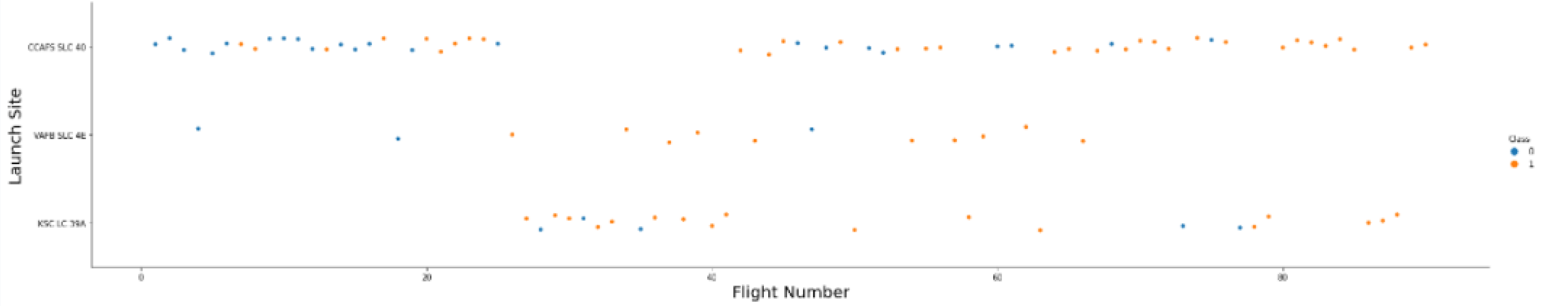


The previous graphs represent the minimum percentage of successful launches compared to the maximum percentage of successful launches that occurred in different launch sites.

Section 2

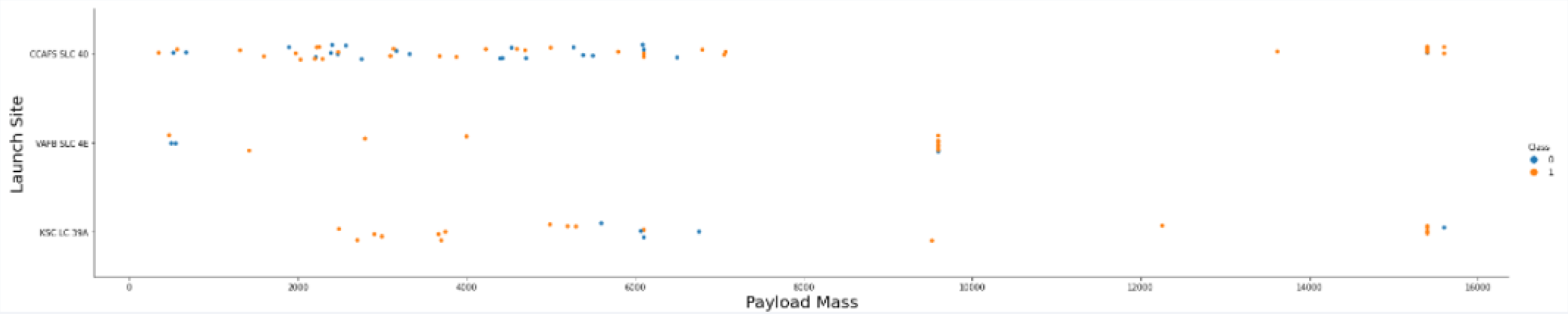
Insights drawn from EDA

Flight Number vs. Launch Site



- Most of the landings have been made on CCAFS SLC-40, followed by KSC LC-39A. Initially the landings were unsuccessful but after Flight 20, there has been a large majority of successful landings. VAFB SLC 4E had few landings but were generally successful.

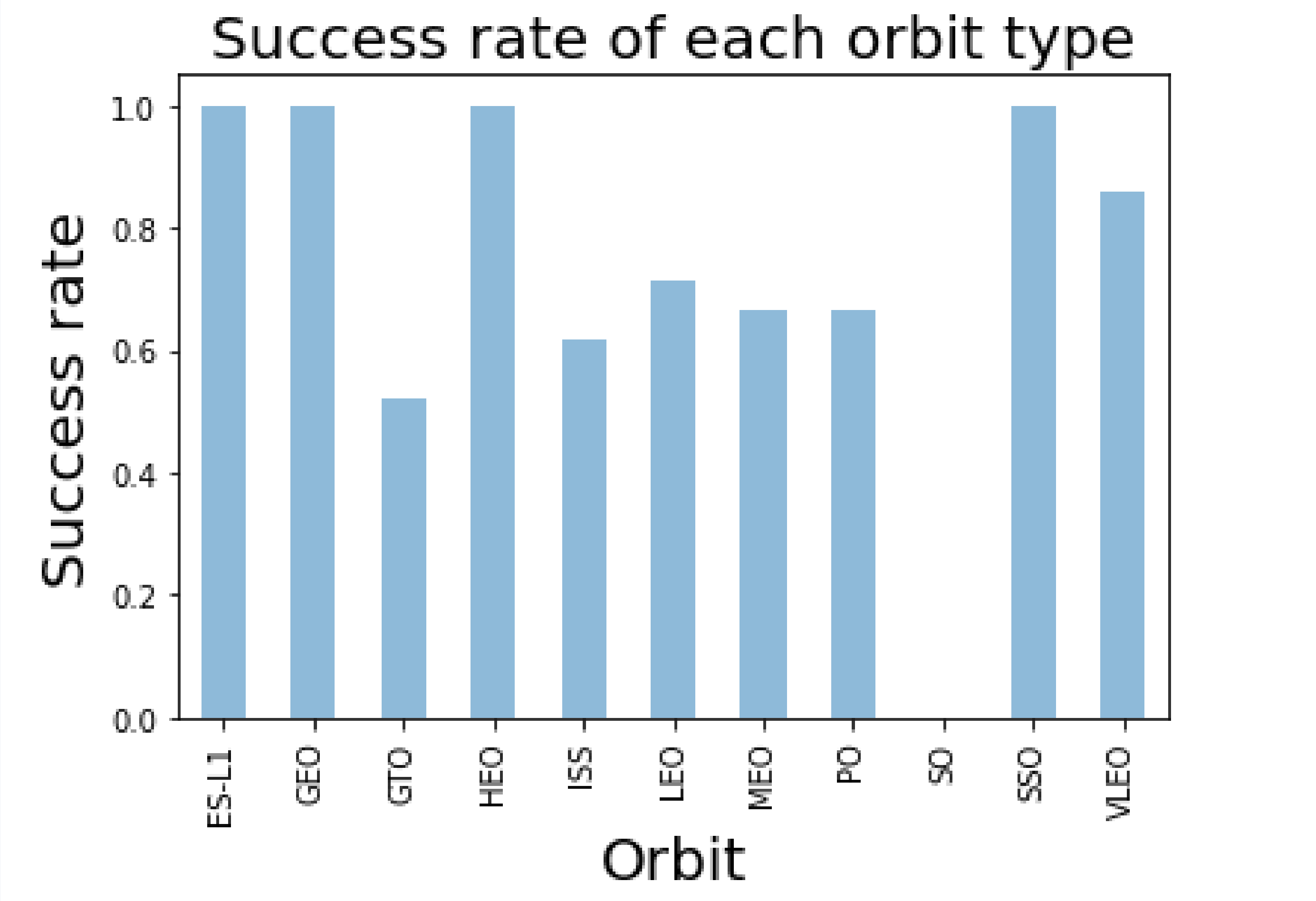
Payload vs. Launch Site



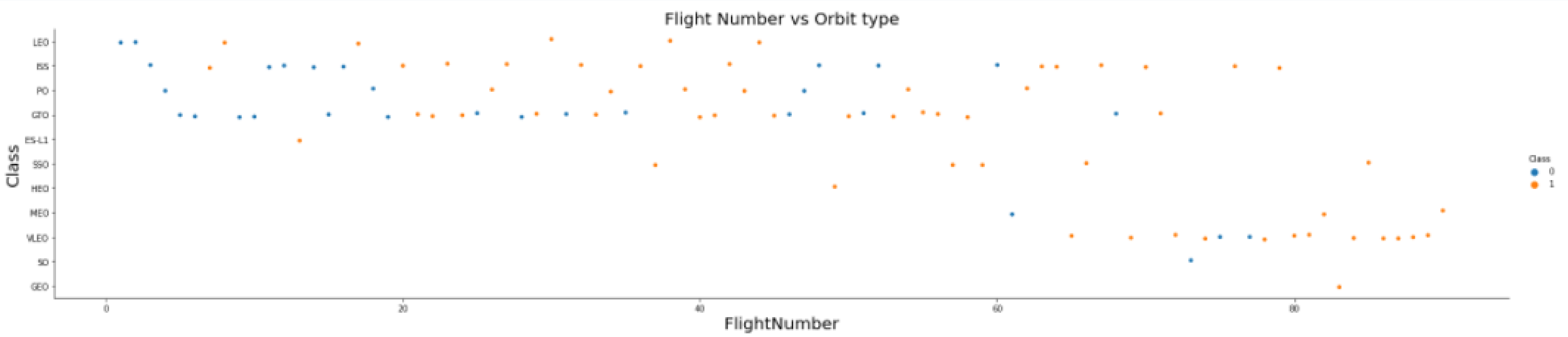
- There were no KSC LC-39A landings with payload mass greater than 2000 kg, most were between 2000kg and 6000kg which were successful. While the landings at the CCAFS SLC 40 site were mostly with a mass payload between 0kg and 8000kg, with varied results, successful landings and failures alike. Most of the launches have been carried out with a payload mass no greater than 8000Kg. KSC LC 39A has been the Launch site with the best results so far. Almost all landings with more than 10,000kg have been successful.

Success Rate vs. Orbit Type

- Four orbits had the best performance rate: ES-L1, GEO, HEO, SSO. While GTO had the lowest success rate. SO did not record any measurements.

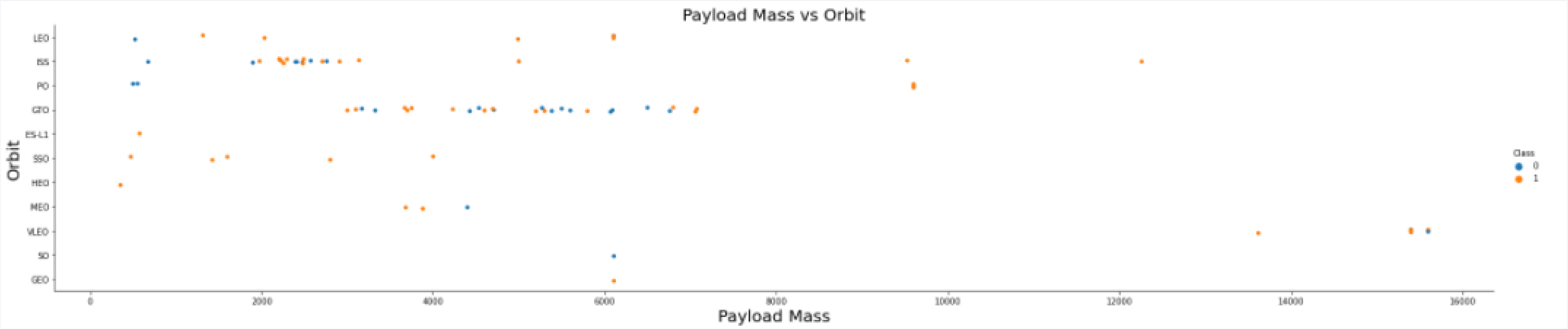


Flight Number vs. Orbit Type



- The first flights were only made in 4 orbits: LEO, ESS, PO and GTO. After flight number 60, flights were started in the rest of the orbits. In the LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.

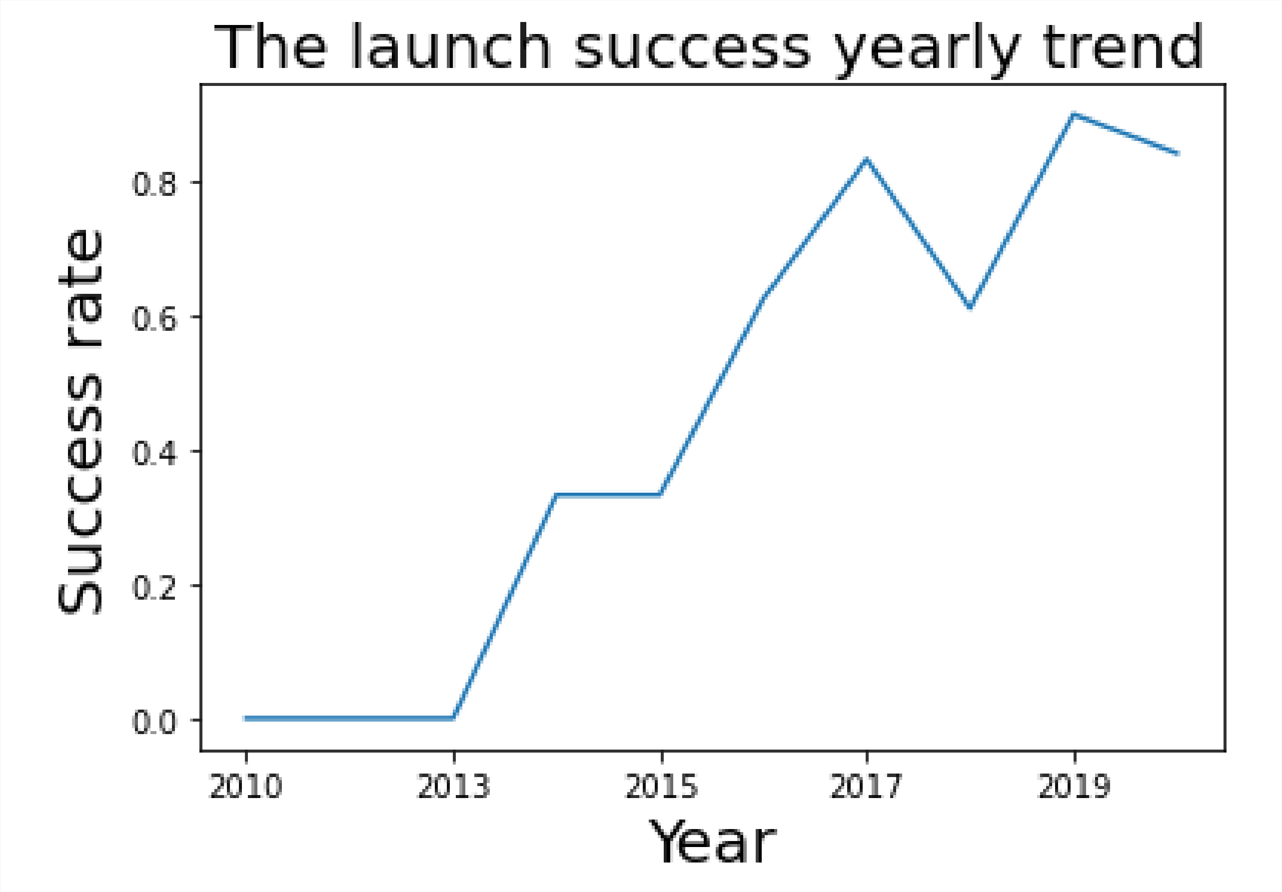
Payload vs. Orbit Type



- Very few landings were made with payload mass greater than 8000 kg. SSO performs well with light payloads, and LEO and ESS perform better with heavy payloads. Heavy payloads have a negative influence on GTO orbits and positive on GTO and Polar LEO (ISS) orbits.

Launch Success Yearly Trend

- The success rate since 2013 kept increasing till 2020. From 2014 to 2015 the same levels are maintained. After 2017 the rate of successful landings decreases and after 2018 they increase again.



All Launch Site Names

```
%sql SELECT unique(launch_site) FROM SPACEXDATASET;
```

```
 * ibm_db_sa://hfv84012:***@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

- There are only four launch sites. Select and unique were used.

Launch Site Names Begin with 'CCA'

```
%sql SELECT launch_site FROM SPACEXDATASET where launch_site LIKE'CCA%' limit 5;
```

```
 * ibm_db_sa://hfv84012:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:31321/bludb
Done.
```

launch_site
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40

- The above are the first five launch sites that start with the letters "CCA“. Select, where, like and limit were used to obtain the expected result.

Total Payload Mass

```
%sql SELECT customer, SUM(payload_mass__kg_) from SPACEXDATASET WHERE customer='NASA (CRS)' GROUP BY customer;
#%sql select * from SPACEXDATASET ;
```

```
 * ibm_db_sa://hfv84012:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:31321/bludb
Done.
```

customer	2
NASA (CRS)	45596

- The total payload mass carried by boosters launched by NASA (CRS) was 45596. Select, where, and group by were used to obtain the expected result.

Average Payload Mass by F9 v1.1

```
%sql SELECT booster_version, AVG(payload_mass__kg_) from SPACEXDATASET where booster_version='F9 v1.1' GROUP BY booster_version;
#%sql select * from SPACEXDATASET ;
```

```
 * ibm_db_sa://hfv84012:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:31321/bludb
Done.
```

booster_version	2
F9 v1.1	2928

- The average payload mass carried by booster version F9 v1.1 was 2928. Select, avg, where, and group by were used to obtain the expected result.

First Successful Ground Landing Date

```
%sql select DATE, landing__outcome from SPACEXDATASET where landing__outcome='Success (ground pad)' limit 1;
#%sql select * from SPACEXDATASET ;
```

```
 * ibm_db_sa://hfv84012:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:31321/bludb
Done.
```

DATE	landing__outcome
2015-12-22	Success (ground pad)

- This is the date when the first successful landing outcome in ground pad was achieved '2015-12-22'. Select and limit were used to obtain the expected result.

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%%sql select booster_version, landing__outcome, payload_mass__kg_
from SPACEXDATASET WHERE landing__outcome='Success (drone ship)' and (payload_mass__kg_>4000) and (payload_mass__kg_<6000);
```

```
 * ibm_db_sa://hfv84012:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:31321/bludb
Done.
```

booster_version	landing__outcome	payload_mass__kg_
F9 FT B1022	Success (drone ship)	4696
F9 FT B1026	Success (drone ship)	4600
F9 FT B1021.2	Success (drone ship)	5300
F9 FT B1031.2	Success (drone ship)	5200

- This is a list the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000. Select and comparatives were used to obtain the expected result.

Total Number of Successful and Failure Mission Outcomes

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

```
 * ibm_db_sa://hfv84012:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:31321/bludb
Done.
```

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

- The total number of successful and failure mission outcomes has 99 successful cases, one failed case and one unclear case. Select, count and group by were used to obtain the expected result.

Boosters Carried Maximum Payload

```
%sql select booster_version, (select max(payload_mass__kg_) as payload from SPACEXDATASET) from SPACEXDATASET ;
```

```
 * ibm_db_sa://hfv84012:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:31321/bludb
Done.
```

booster_version	payload
F9 v1.0 B0003	15600
F9 v1.0 B0004	15600
F9 v1.0 B0005	15600
F9 v1.0 B0006	15600
F9 v1.0 B0007	15600
F9 v1.1 B1003	15600
F9 v1.1	15600
F9 v1.1	15600
F9 v1.1	15600
F9 v1.1	15600
F9 v1.1	15600
F9 v1.1 B1011	15600
F9 v1.1 B1010	15600
F9 v1.1 B1012	15600
F9 v1.1 B1013	15600
F9 v1.1 B1014	15600
F9 v1.1 B1015	15600
F9 v1.1 B1016	15600
F9 v1.1 B1018	15600
F9 FT B1019	15600
F9 FT B1021.1	15600
F9 FT B1022	15600
F9 FT B1023.1	15600
F9 FT B1024	15600
F9 FT B1025.1	15600
F9 FT B1026	15600
F9 FT B1029.1	15600
F9 FT B1031.1	15600
F9 FT B1030	15600
F9 FT B1021.2	15600
F9 FT B1032.1	15600
F9 FT B1034	15600
F9 FT B1035.1	15600
F9 FT B1029.2	15600
F9 FT B1036.1	15600
F9 FT B1037	15600
F9 B4 B1039.1	15600
F9 FT B1038.1	15600
F9 B4 B1040.1	15600
F9 B4 B1041.1	15600
F9 FT B1031.2	15600
F9 B4 B1042.1	15600
F9 FT B1035.2	15600
F9 FT B1036.2	15600
F9 B4 B1043.1	15600
F9 FT B1032.2	15600
F9 FT B1038.2	15600
F9 B4 B1044	15600
F9 B4 B1041.2	15600
F9 B4 B1039.2	15600
F9 B4 B1045.1	15600
F9 B5 B1046.1	15600
F9 B4 B1043.2	15600
F9 B4 B1040.2	15600
F9 B4 B1045.2	15600
F9 B5B1047.1	15600
F9 B5B1048.1	15600
F9 B5 B1046.2	15600
F9 B5B1049.1	15600
F9 B5 B1048.2	15600
F9 B5 B1047.2	15600
F9 B5 B1046.3	15600
F9 B5B1050	15600
F9 B5B1054	15600
F9 B5 B1049.2	15600
F9 B5 B1048.3	15600
F9 B5B1051.1	15600
F9 B5B1056.1	15600
F9 B5 B1049.3	15600
F9 B5 B1051.2	15600
F9 B5 B1056.2	15600
F9 B5 B1047.3	15600
F9 B5 B1048.4	15600
F9 B5B1059.1	15600
F9 B5 B1056.3	15600
F9 B5 B1049.4	15600
F9 B5 B1046.4	15600
F9 B5 B1051.3	15600
F9 B5 B1056.4	15600
F9 B5 B1059.2	15600
F9 B5 B1048.5	15600
F9 B5 B1051.4	15600
F9 B5B1058.1	15600
F9 B5 B1049.5	15600
F9 B5 B1059.3	15600
F9 B5B1060.1	15600
F9 B5 B1058.2	15600
F9 B5 B1051.5	15600
F9 B5 B1049.6	15600
F9 B5 B1059.4	15600
F9 B5 B1060.2	15600
F9 B5 B1058.3	15600
F9 B5 B1051.6	15600
F9 B5 B1060.3	15600
F9 B5B1062.1	15600
F9 B5B1061.1	15600
F9 B5B1063.1	15600
F9 B5 B1049.7	15600
F9 B5 B1058.4	15600

- The names of the booster_versions which have carried the maximum payload mass are listed above. Select and subquery were used to obtain the expected result.

2015 Launch Records

```
%sql select landing__outcome, booster_version, launch_site, DATE from SPACEXDATASET where landing__outcome='Failure (drone ship)' and YEA
R(DATE)=2015;
```

```
 * ibm_db_sa://hfv84012:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:31321/bludb
Done.
```

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

- The failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015 are listed above. Select, where and year(date) were used to get the expected result.

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

```
%%sql select landing__outcome, DATE from SPACEXDATASET
WHERE landing__outcome='Failure (drone ship)' or landing__outcome='Success (ground pad)' and (DATE) between '2010-06-04' and '2017-03-20'
order by DATE desc;
```

```
 * ibm_db_sa://hfv84012:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:31321/bludb
Done.
```

landing__outcome	DATE
Success (ground pad)	2017-02-19
Success (ground pad)	2016-07-18
Failure (drone ship)	2016-06-15
Failure (drone ship)	2016-03-04
Failure (drone ship)	2016-01-17
Success (ground pad)	2015-12-22
Failure (drone ship)	2015-04-14
Failure (drone ship)	2015-01-10

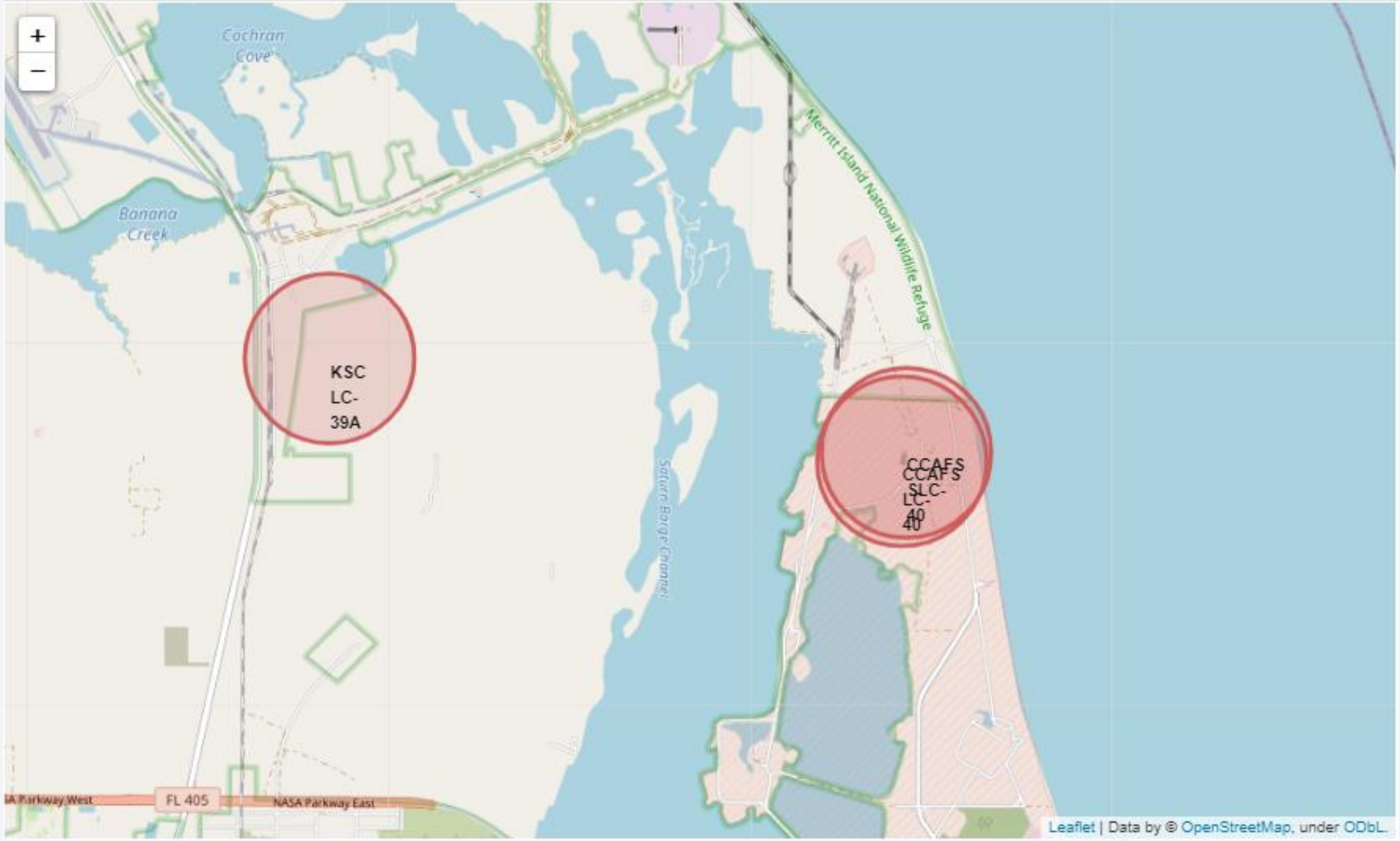
- Landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order are listed above. Select, where, order by, desc and comparative were used to obtain the expected result.

Section 4

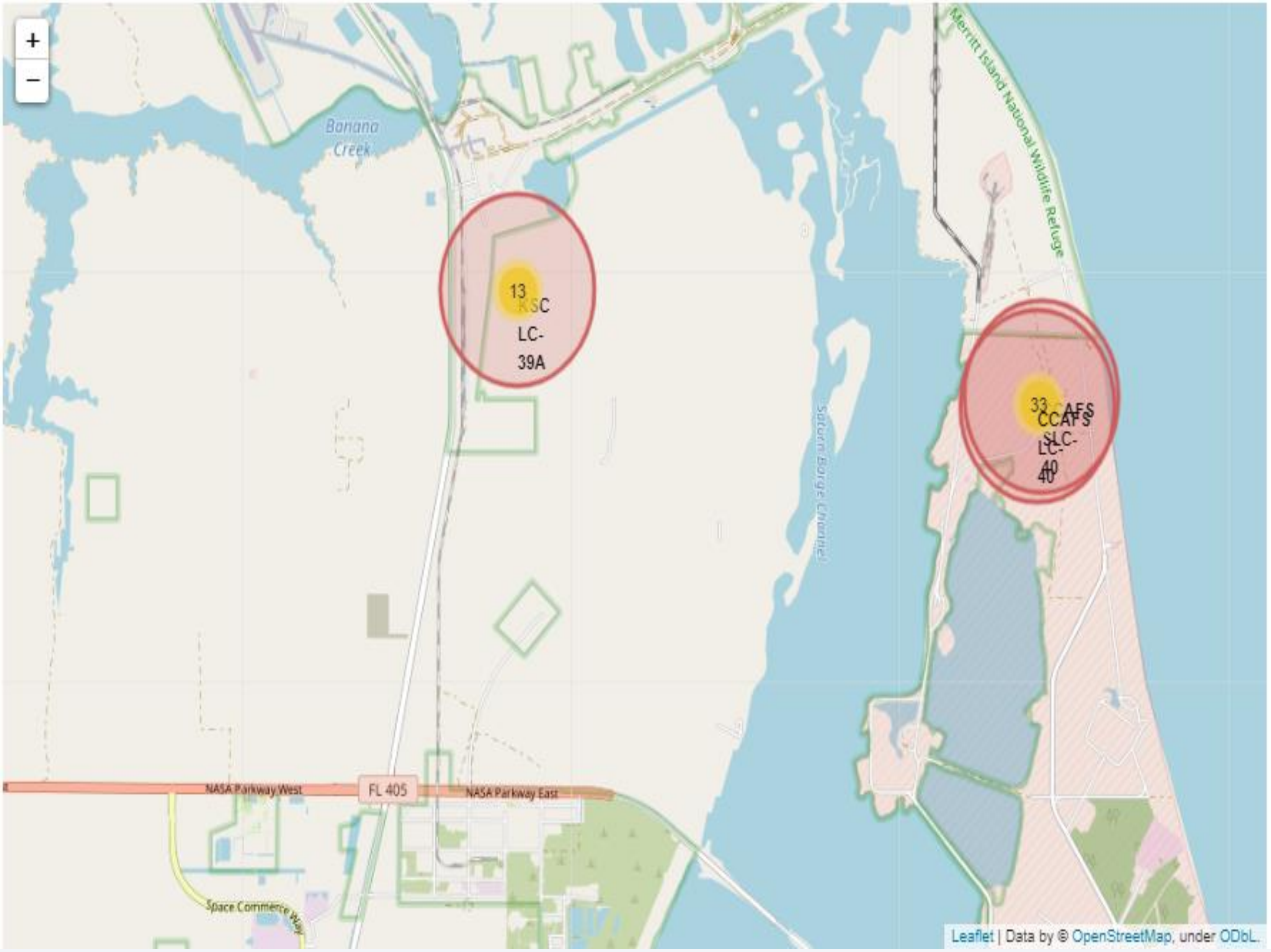
Launch Sites Proximities Analysis

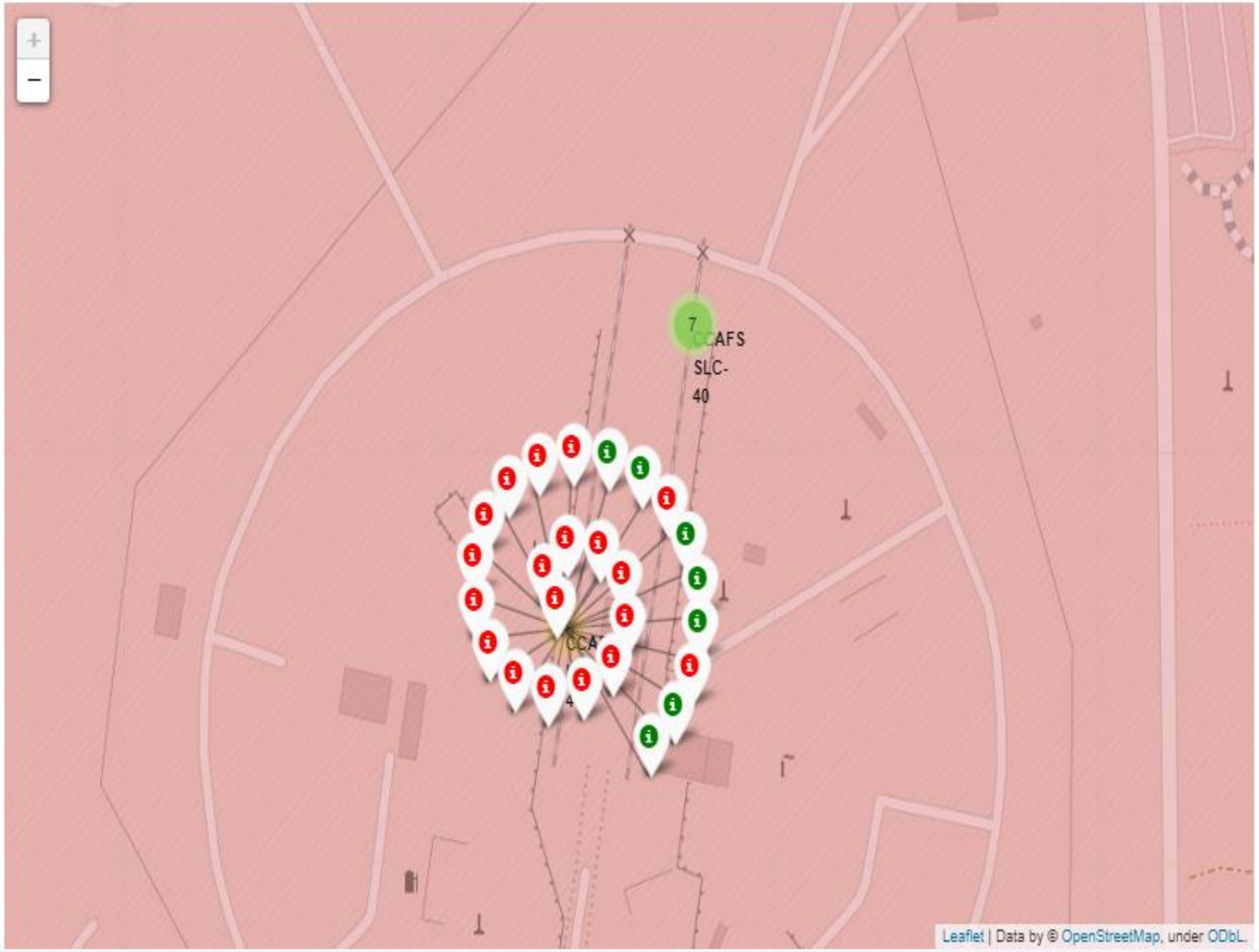
All launch sites on a map

- The launch sites are on the coast, close to the sea. There are two launch sites that are very close to each other, and a fourth that is in the state of California in the United States.



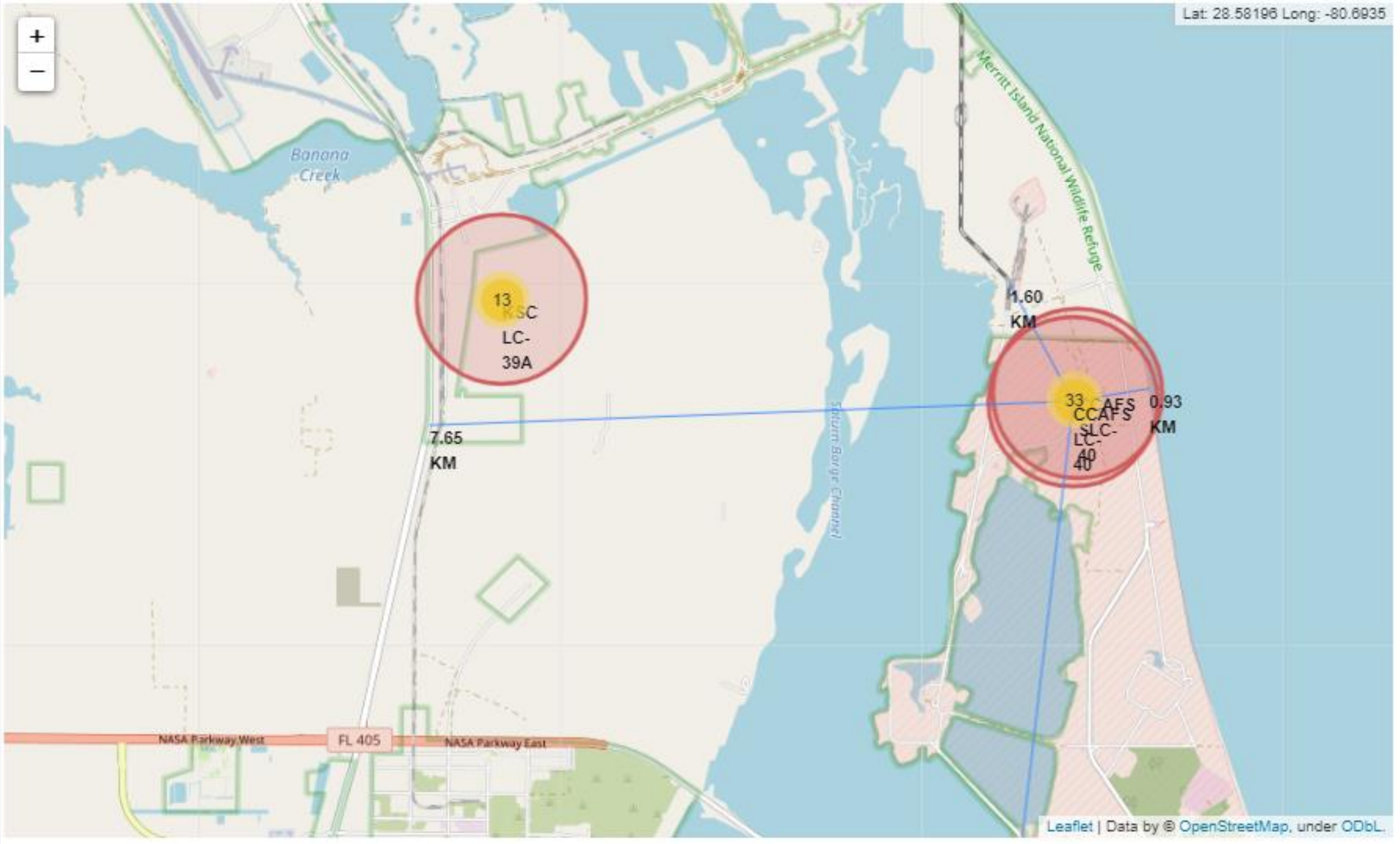
Success and failed launches for each site on the map





Distances between a launch site to its proximities

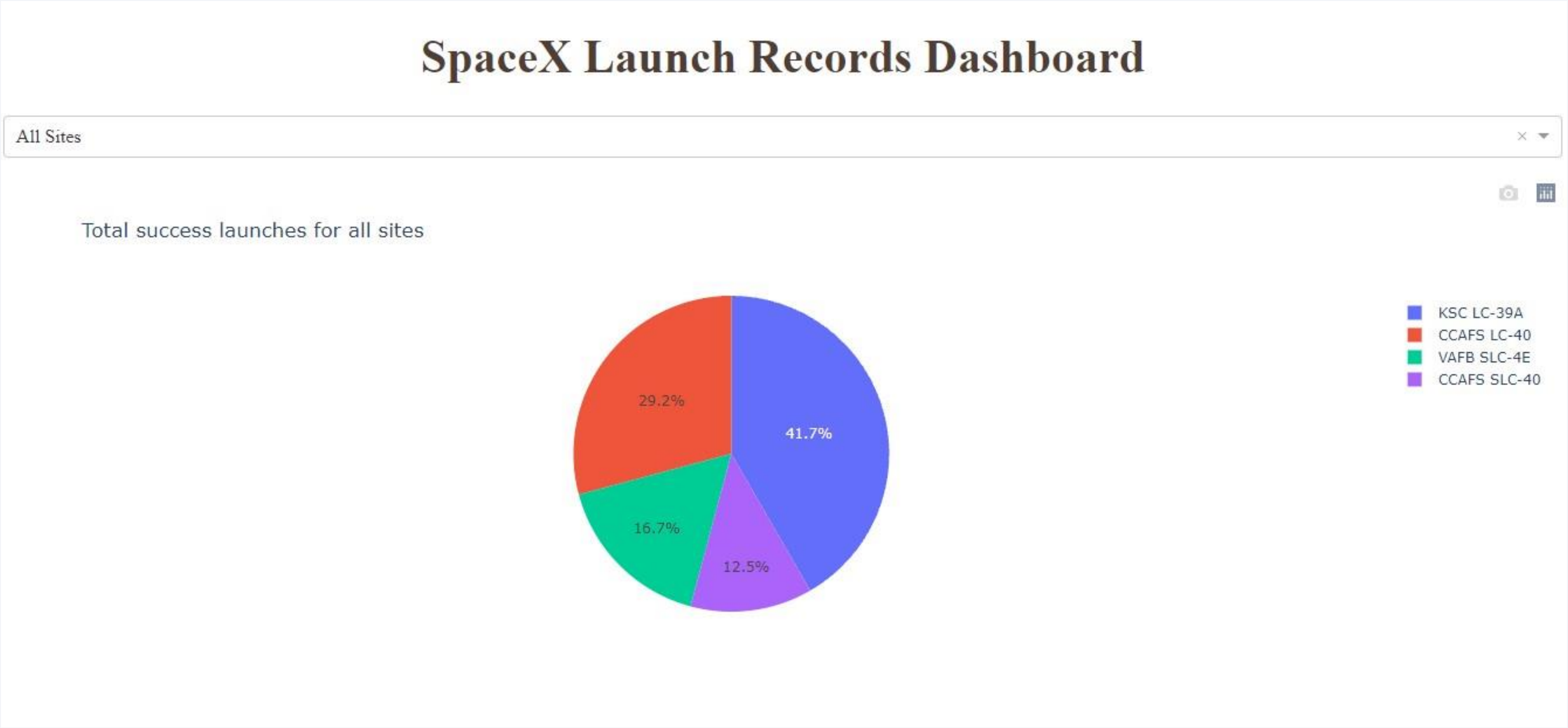
- The closest CCAFS LC-40 launch sites to the railways is approximately 1.60 km away. The closest highway is approximately 7.65 km from the CCAFS LC-40 launch site. Melbourne is one of the closest cities. This is located approximately 51.06 km from the CCAFS LC-40 launch site. The launch sites are almost on the coastline.



Section 5

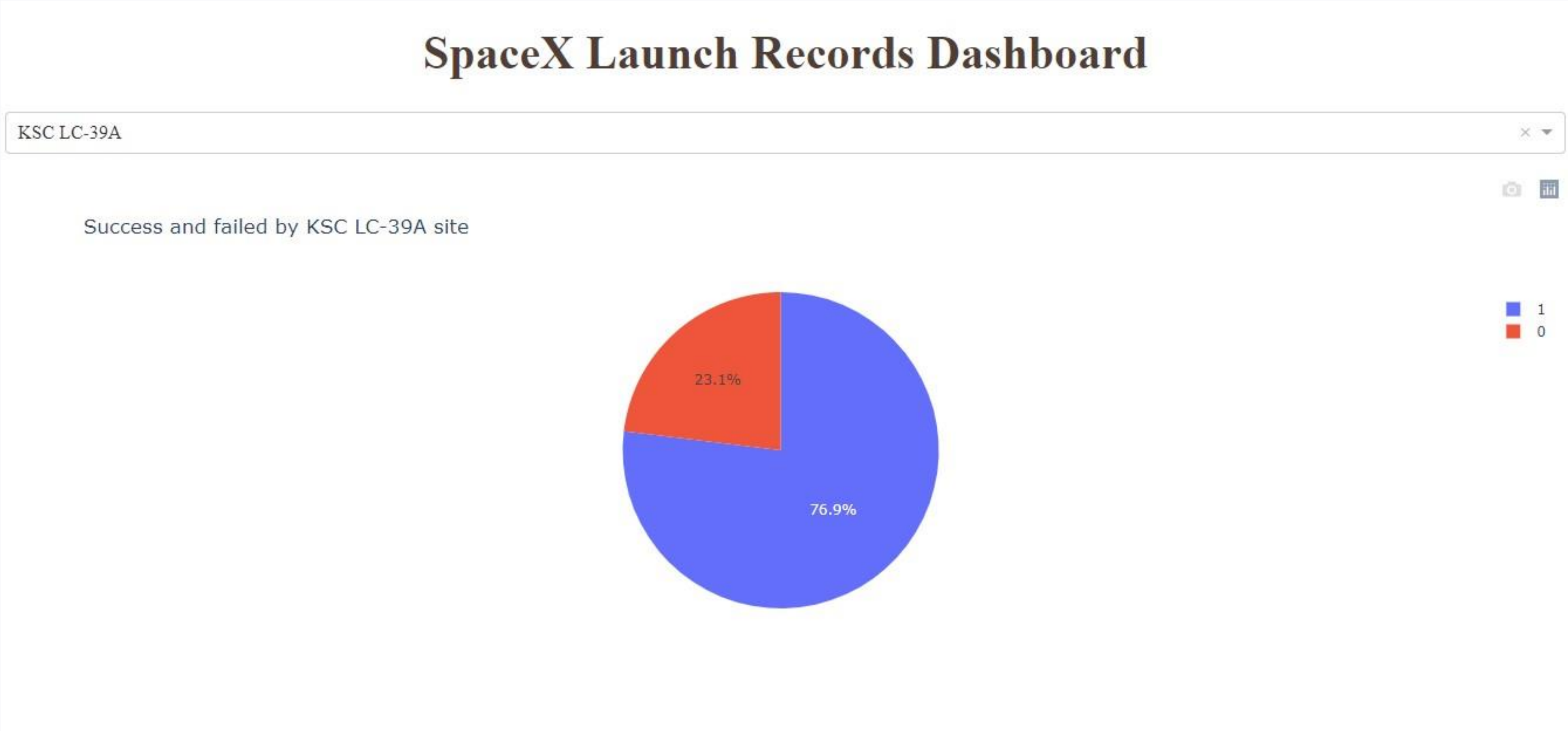
Build a Dashboard with Plotly Dash

Records dashboard for all sites



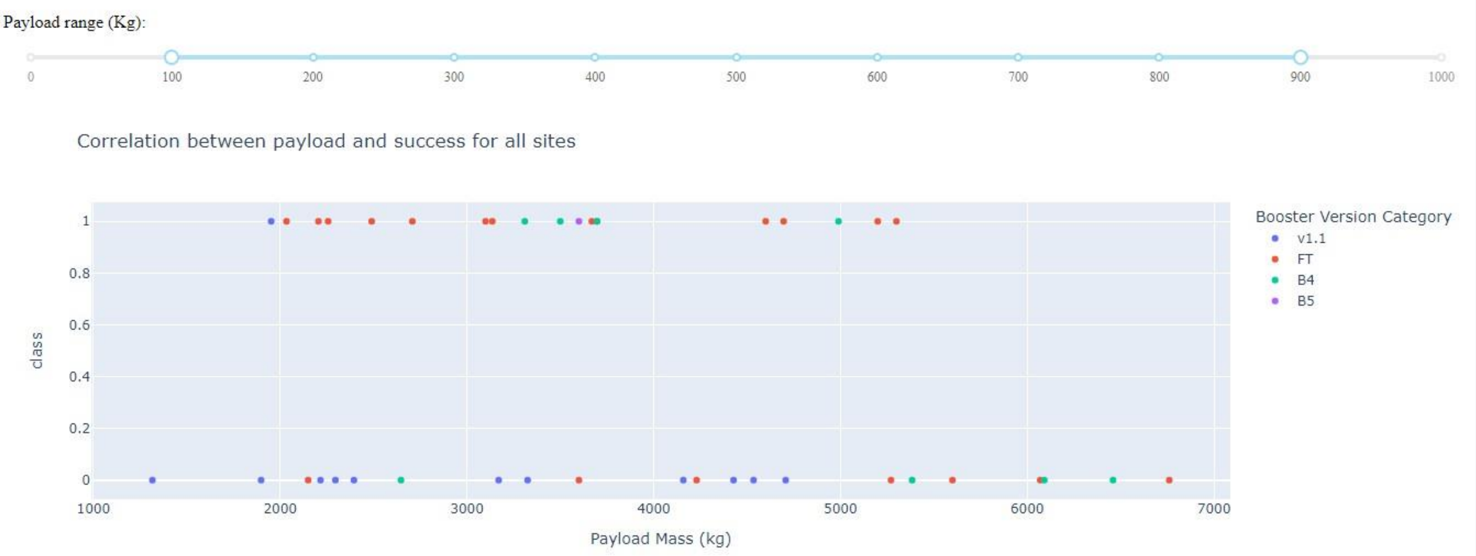
- In the previous pie chart we can see the different percentages of success stories for each launch site. KSC LC-39A registers the highest percentage with 41.7% and CCAFS SLC-40 with the lowest percentage being 12.5%.

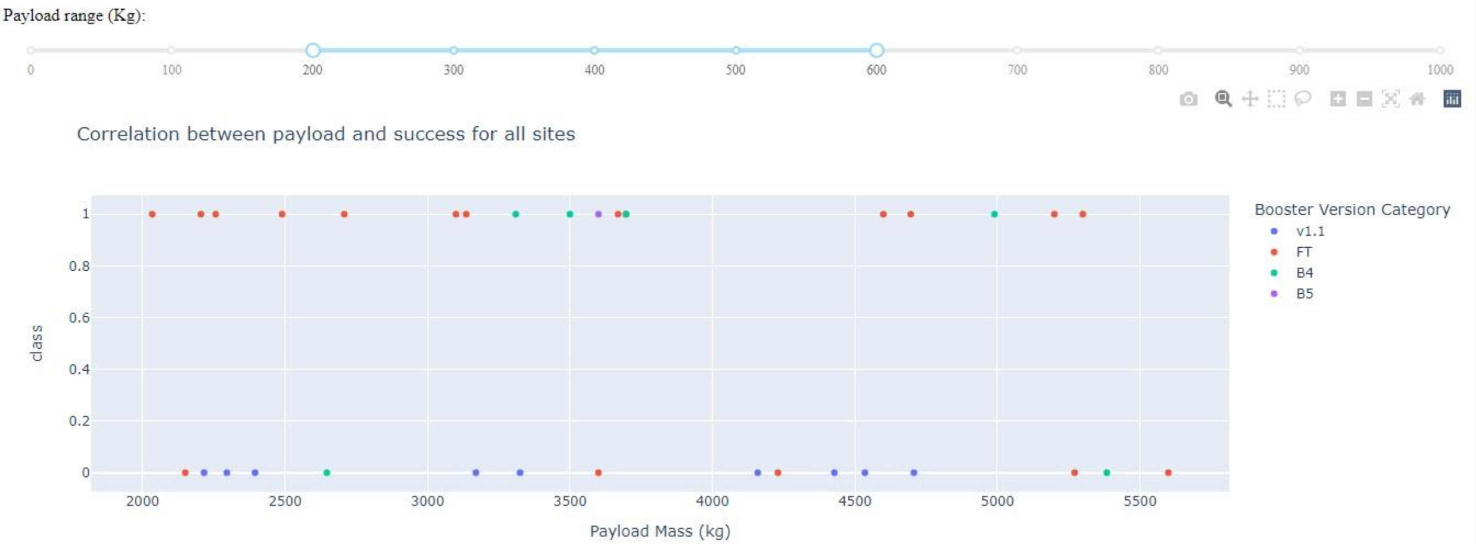
Records dashboard for KSC LC-39A



- KSC LC-39A has a 76.9% success rate and a 23.1% failure rate.

Records dashboard for correlations between payload and success for all sites



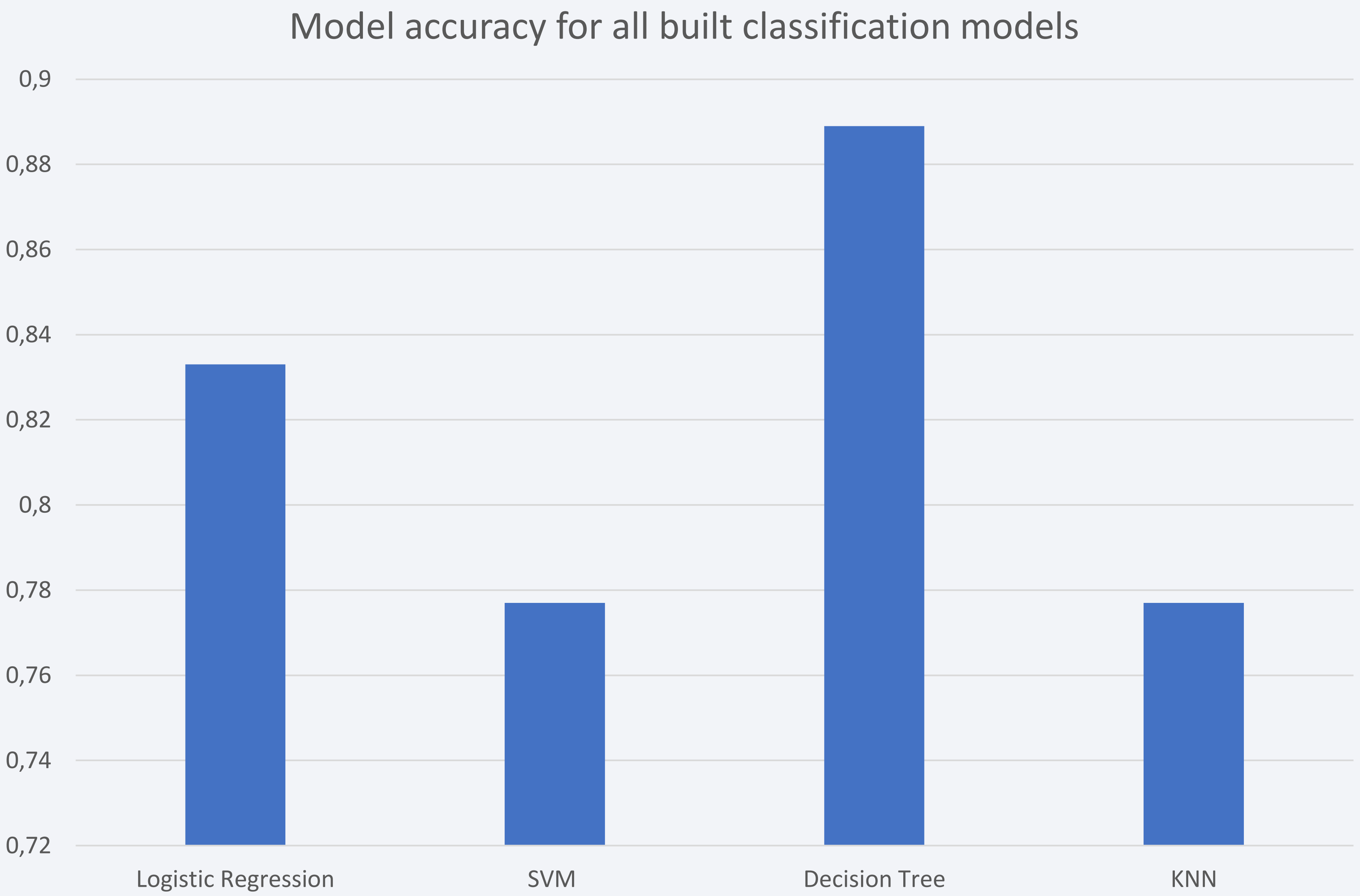


- Few landings with payload mass greater than 8000kg. The Booster version category with the highest number of launches were v1.1, the majority with successful landing cases and FT, the majority with unsuccessful landing cases.

Section 6

Predictive Analysis (Classification)

Classification Accuracy



Classification Accuracy

KEEP IN MIND!

There is some randomness to it: One thing to keep in mind is that our test is very small. It is only 18. In real situations, we likely won't be using a test size this small. So what is happening is that sometimes a group is split to a leaf that is labeled failure and sometimes it's split to a leaf that is labeled success.

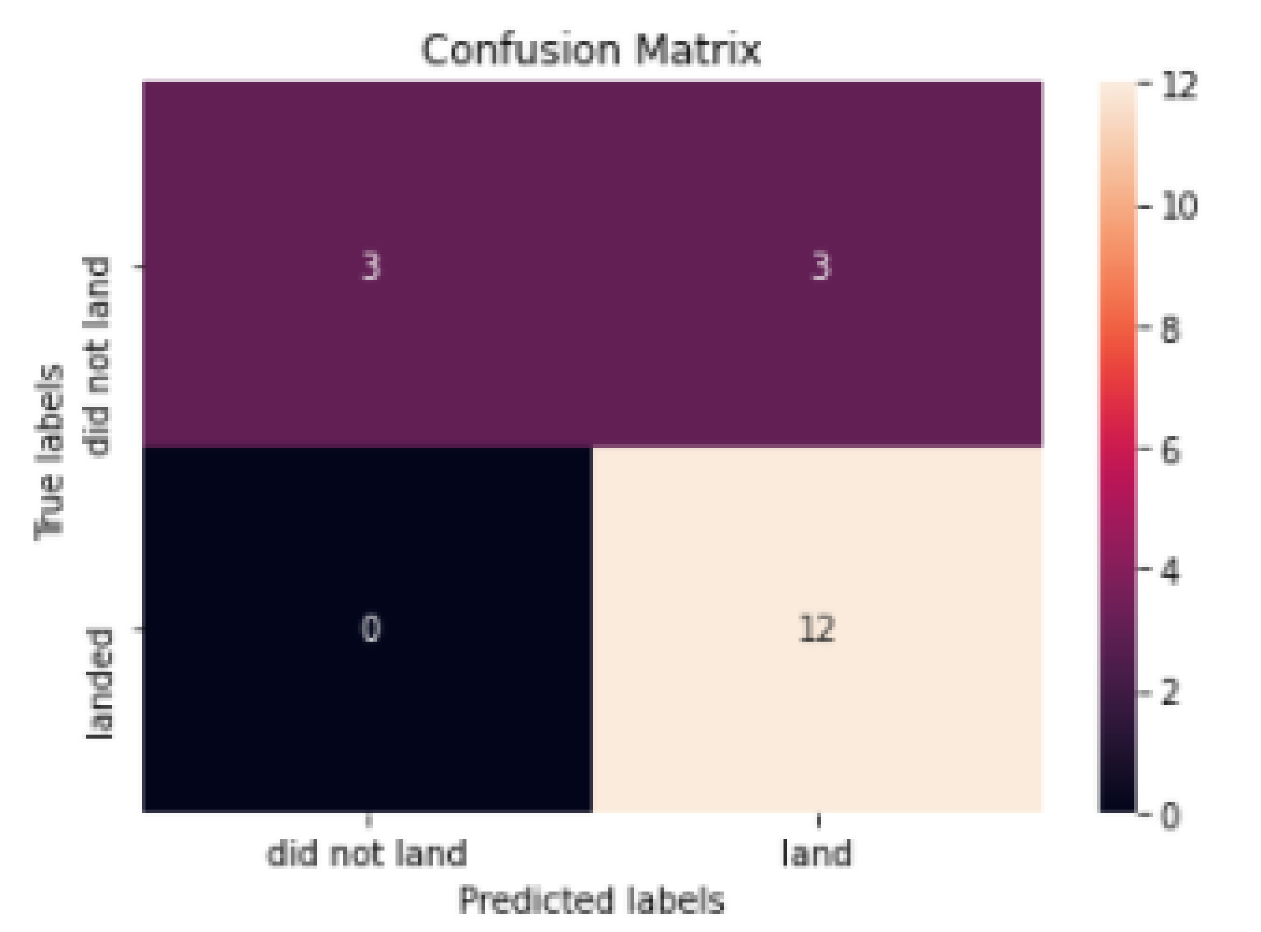
For a test sample size this small this is going to have huge effects on our accuracy:

16/18 correct is 0.88888 accuracy 15/18 correct is 0.83333 accuracy 14/18 correct would be 0.7777 accuracy

This is likely why your two scores are not equal. In fact, if you run any of your code enough times you'll see that the exact same code will produce different scores. If you want the randomness to stay consistent DecisionTreeClassifier takes a random_state argument that you can set.

Confusion Matrix

- Three labels were correctly predicted as missed landings. Three labels were incorrectly predicted indicating that the landing was successful, but it was actually successful. There was no case in which the landings were predicted to be unsuccessful, but in reality they were successful. Twelve labels were correctly predicted as successful landings.



Conclusions

- The data was collected by developing two methodologies, the first consists of connecting directly with the SpaceX API, and the second consists of performing Web Scraping on Wikipedia.
- The data is cleaned, data that can generate noise is modified. Initially the labels are modified as needed, followed by the data contained in each column. Finally the data is filtered if required.
- Queries can be made using SQL, to explore the data, during the queries no data modifications are normally made.
- Folium and Dash are tools of great help for the visualization and interaction of the data.
- Finally, the predictive models used in Machine Learning help determine if the first stage of Falcon 9 will land successfully. Each model performs its best and the accuracy of each model is evaluated. In this case, the decision tree was the model that best fit the data, and achieved the highest score.

Appendix

- https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html
- https://matplotlib.org/stable/gallery/pie_and_polar_charts/pie_features.html
- https://www.w3schools.com/python/matplotlib_scatter.asp
- https://dash.plotly.com/
- https://python-visualization.github.io/folium/

Thank you!