Data Science Capstone project

<José Cáceres>

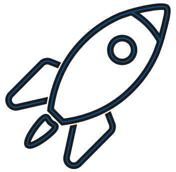
<27/08/2021>

Outline

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

Executive Summary

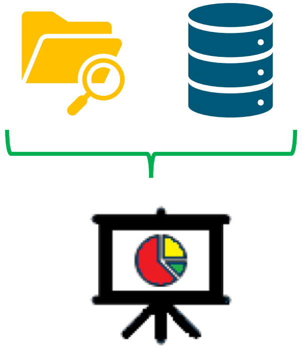
This project consists of the prediction of the successful landing of stage 1 on the Falcon 9 rocket.



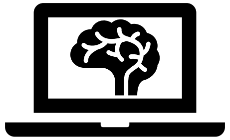
One of the main reasons why the launch costs of the Space X company are lower than the rest is due to the correct landing of stage 1 of its rockets. This strategy allows a cost reduction of up to 100 million dollars.

That is why it is of the utmost importance for the company to be able to determine whether the landing of phase one of the rockets will end in success or failure.

In order to predict the successful landing of stage 1 of the falcon 9 rocket, the first step is the collection, cleaning and organization of data, followed by different visualizations that allow a deeper understanding of the data.



Finally, a machine learning model is developed capable of performing the task of predicting the success or otherwise of the landing of stage 1 of the rockets.



Introduction

- The costs of rocket launches are extremely high in part due to the loss of the stage 1 of the rockets.
- The costs of the spaceX Falcon 9 rocket launches are considerably lower than the rest because they can recover stage 1 of the rockets.
- In order to keep the costs of rocket launches low, it is important to be able to predict the success of the Falcon 9 stage 1 rocket landing.

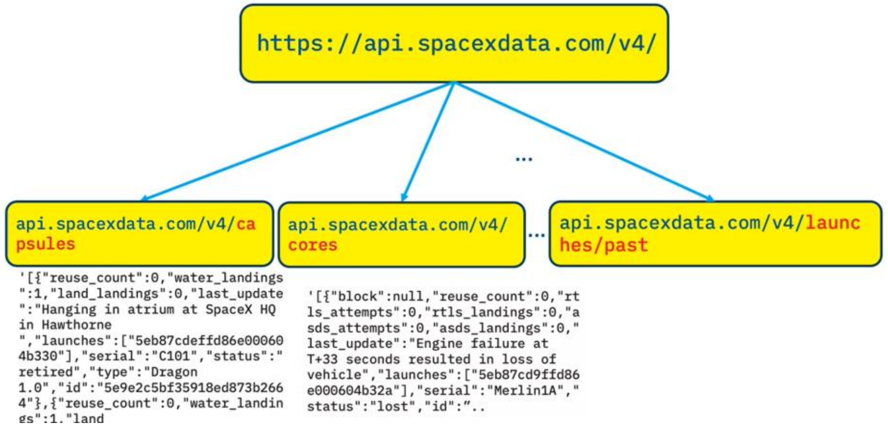
Methodology

- Perform data wrangling
  - Describe how data were processed
- 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

Methodology

Data collection

The information used for this project corresponds to the data obtained from the **SpaceX API** together with the **Wikipedia** data on the falcon 9 launches collected using the WebScrapping method.



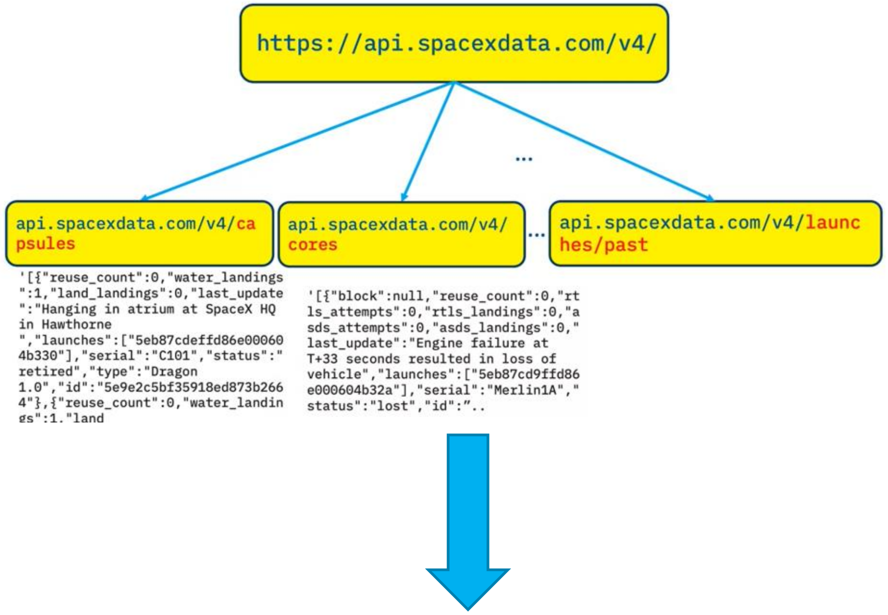
[hide] Flight No.	Date and time (UTC)	Version, Booster [b]	Launch site	Payload[c]	Payload mass	Orbit	Customer	Launch outcome	Booster landing
1	4 June 2010, 18:45	F9 v1.0[7] B0003.1[8]	CCAFS, SLC-40	Dragon Spacecraft Qualification Unit		LEO	SpaceX	Success	Failure[9][10] (parachute)
	First flight of Falcon 9 v1.0.[11] Used a boilerplate version of Dragon capsule which was not designed to separate from the second stage.(more details below) Attempted to recover the first stage by parachuting it into the ocean, but it burned up on reentry, before the parachutes even deployed.[12]								
2	8 December 2010, 15:43[13]	F9 v1.0[7] B0004.1[8]	CCAFS, SLC-40	Dragon demo flight C1 (Dragon C101)		LEO (ISS)	NASA (COTS) NRO	Success[9]	Failure[9][14] (parachute)
	Maiden flight of Dragon capsule, consisting of over 3 hours of testing thruster maneuvering and reentry.[15] Attempted to recover the first stage by parachuting it into the ocean, but it disintegrated upon reentry, before the parachutes were deployed.[12] (more details below) It also included two CubeSats,[16] and a wheel of Brouère cheese. Before launch, SpaceX found there was a crack in the nozzle of the 2nd stage engine. So Elon just had them cut off the end of the nozzle with a pair of shears and launched the rocket a few days later. After SpaceX trimmed the nozzle, NASA was notified of the change and they agreed to it.[17]								
3	22 May 2012, 07:44[18]	F9 v1.0[7] B0005.1[8]	CCAFS, SLC-40	Dragon demo flight C2+[19] (Dragon C102)	525 kg (1,157 lb)[20]	LEO (ISS)	NASA (COTS)	Success[21]	No attempt
	Dragon spacecraft demonstrated a series of tests before it was allowed to approach the International Space Station. Two days later, it became the first commercial spacecraft to board the ISS.[18] (more details below)								
4	8 October 2012, 00:35[22]	F9 v1.0[7] B0006.1[8]	CCAFS, SLC-40	SpaceX CRS-1[23] (Dragon C103)	4,700 kg (10,400 lb)	LEO (ISS)	NASA (CRS)	Success	No attempt
				Orbcomm-OG2[24]	172 kg (379 lb)[25]	LEO	Orbcomm	Partial failure[26]	
	CRS-1 was successful, but the secondary payload was inserted into an abnormally low orbit and subsequently lost. This was due to one of the nine Merlin engines shutting down during the launch, and NASA declining a second reignition, as per ISS visiting vehicle safety rules, the primary payload owner is contractually allowed to decline a second reignition. NASA stated that this was because SpaceX could not guarantee a high enough likelihood of the second stage completing the second burn successfully which was required to avoid any risk of secondary payload's collision with the ISS.[27][28][29]								
5	1 March 2013, 15:10	F9 v1.0[7] B0007.1[8]	CCAFS, SLC-40	SpaceX CRS-2[23] (Dragon C104)	4,877 kg (10,752 lb)	LEO (ISS)	NASA (CRS)	Success	No attempt
	Last launch of the original Falcon 9 v1.0 launch vehicle, first use of the unpressurized trunk section of Dragon.[30]								
6	29 September 2013, 16:00[31]	F9 v1.1[7] B1003[8]	VAFB, SLC-4E	CASSIOPE[23][32]	500 kg (1,100 lb)	Polar orbit LEO	MDA	Success[31]	Uncontrolled (ocean)[d]
	First commercial mission with a private customer, first launch from Vandenberg, and demonstration flight of Falcon 9 v1.1 with an improved 13-tonne to LEO capacity.[30] After separation from the second stage carrying Canadian commercial and scientific satellites, the first stage booster								

Data collection – SpaceX API

Information and details on Falcon 9 rocket launches are obtained from the SpaceX API.

As seen in the figure, this api contains multiple endpoints that provide different information about Falcon 9 launches, using the beautifulsoup and python requests libraries it is possible to extract this data and structure it for later analysis.

Finally, the data obtained from the API corresponds to:
booster name, the mass of the payload, the orbit that it is going to, the name of the launch site and its position, the outcome of the landing, the type of the landing, number of flights and the core information.



	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	Longitude	Latitude
4	1	2010-06-04	Falcon 9	NaN	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	525.0	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	677.0	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	500.0	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	3170.0	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	15600.0	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	15600.0	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	15600.0	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	15600.0	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	3681.0	MEO	CCSFS SLC 40	True ASDS	1	True	False	True	5e9e3032383ecb6bb234e7ca	5.0	1	B1062	-80.577366	28.561857

https://github.com/joseMCV/space-y/blob/e289e099c77fd5a5aba82732517da419df227843/space_y_api.ipynb

Data collection – Web scraping

To extract the data from wikipedia, first the raw page must be read, for this the python requests library is used, once the page has been read, BeautifulSoup is used to find the specific information that is required from this page.

requested corresponds to the following: Flight number, date and time, place of launch, payload, orbit to which the launch is directed, client and final status of the operation (success or failure)

[hide] Flight No.	Date and time (UTC)	Version, Booster [b]	Launch site	Payload[c]	Payload mass	Orbit	Customer	Launch outcome	Booster landing
1	4 June 2010, 18:45	F9 v1.0[7] B0003.1[8]	CCAFS, SLC-40	Dragon Spacecraft Qualification Unit		LEO	SpaceX	Success	Failure[9][10] (parachute)
	First flight of Falcon 9 v1.0.[11] Used a boilerplate version of Dragon capsule which was not designed to separate from the second stage.(more details below) Attempted to recover the first stage by parachuting it into the ocean, but it burned up on reentry, before the parachutes even deployed.[12]								
2	8 December 2010, 15:43[13]	F9 v1.0[7] B0004.1[8]	CCAFS, SLC-40	Dragon demo flight C1 (Dragon C101)		LEO (ISS)	NASA (COTS) NRO	Success[9]	Failure[9][14] (parachute)
	Maiden flight of Dragon capsule, consisting of over 3 hours of testing thruster maneuvering and reentry.[15] Attempted to recover the first stage by parachuting it into the ocean, but it disintegrated upon reentry, before the parachutes were deployed.[12] (more details below) It also included two CubeSats,[16] and a wheel of Brouère cheese. Before launch, SpaceX found there was a crack in the nozzle of the 2nd stage engine. So Elon just had them cut off the end of the nozzle with a pair of shears and launched the rocket a few days later. After SpaceX trimmed the nozzle, NASA was notified of the change and they agreed to it.[17]								
3	22 May 2012, 07:44[18]	F9 v1.0[7] B0005.1[8]	CCAFS, SLC-40	Dragon demo flight C2+[19] (Dragon C102)	525 kg (1,157 lb)[20]	LEO (ISS)	NASA (COTS)	Success[21]	No attempt
	Dragon spacecraft demonstrated a series of tests before it was allowed to approach the International Space Station. Two days later, it became the first commercial spacecraft to board the ISS.[18] (more details below)								
4	8 October 2012, 00:35[22]	F9 v1.0[7] B0006.1[8]	CCAFS, SLC-40	SpaceX CRS-1[23] (Dragon C103)	4,700 kg (10,400 lb)	LEO (ISS)	NASA (CRS)	Success	No attempt
				Orbcomm-OG2[24]	172 kg (379 lb)[25]	LEO	Orbcomm	Partial failure[26]	
	CRS-1 was successful, but the secondary payload was inserted into an abnormally low orbit and subsequently lost. This was due to one of the nine Merlin engines shutting down during the launch, and NASA declining a second reignition, as per ISS visiting vehicle safety rules, the primary payload owner is contractually allowed to decline a second reignition. NASA stated that this was because SpaceX could not guarantee a high enough likelihood of the second stage completing the second burn successfully which was required to avoid any risk of secondary payload's collision with the ISS.[27][28][29]								
5	1 March 2013, 15:10	F9 v1.0[7] B0007.1[8]	CCAFS, SLC-40	SpaceX CRS-2[23] (Dragon C104)	4,877 kg (10,752 lb)	LEO (ISS)	NASA (CRS)	Success	No attempt
	Last launch of the original Falcon 9 v1.0 launch vehicle, first use of the unpressurized trunk section of Dragon.[30]								
6	29 September 2013, 16:00[31]	F9 v1.1[7] B1003[8]	VAFB, SLC-4E	CASSIOPE[23][32]	500 kg (1,100 lb)	Polar orbit LEO	MDA	Success[31]	Uncontrolled (ocean)[d]
	First commercial mission with a private customer, first launch from Vandenberg, and demonstration flight of Falcon 9 v1.1 with an improved 13-tonne to LEO capacity.[30] After separation from the second stage carrying Canadian commercial and scientific satellites, the first stage booster								

	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
...	...	...	...	...	...	...	...	...	...	...	...
116	117	CCSFS	Starlink	15,600 kg	LEO	SpaceX	Success\n	F9 B5B1051.10	Success	9 May 2021	06:42
117	118	KSC	Starlink	~14,000 kg	LEO	SpaceX	Success\n	F9 B5B1058.8	Success	15 May 2021	22:56
118	119	CCSFS	Starlink	15,600 kg	LEO	SpaceX	Success\n	F9 B5B1063.2	Success	26 May 2021	18:59
119	120	KSC	SpaceX CRS-22	3,328 kg	LEO	NASA	Success\n	F9 B5B1067.1	Success	3 June 2021	17:29
120	121	CCSFS	SXM-8	7,000 kg	GTO	Sirius XM	Success\n	F9 B5	Success	6 June 2021	04:26

https://github.com/joseMCV/space-y/blob/e289e099c77fd5a5aba82732517da419df227843/space_y_webscrapping.ipynb

Data wrangling

- To make the prediction about the success or failure of the Falcon 9 rocket stage 1 landing, it is important to label the data, that is, to be able to determine if the rocket stage 1 landed successfully or not.
- In this stage, different columns of the data were analyzed to understand the relevance of each one with respect to the final objective of the mission, that is, how each column affects the success in the landing of stage 1.

https://github.com/joseMCV/space-y/blob/e289e099c77fd5a5aba82732517da419df227843/space_y_wrangling.ipynb

EDA with data visualization

- The first visualizations generated correspond to scatter graphics, this is to visualize the relationships between the different variables, the relationships plotted were:
  - FlightNumber vs. PayloadMass.
  - FlightNumber vs LaunchSite.
  - LaunchSites vd PayloadMass.
  - FlightNumber and Orbit type.
  - Payload vs. Orbit type.

- It should be noted that all the visualizations are contrasted with the success or not of the landing, so thanks to these it can be determined whether the graphed variables influence the success of the launch.

https://github.com/joseMCV/space-y/blob/e289e099c77fd5a5aba82732517da419df227843/space_y_visualizationseda.ipynb

EDA with data visualization

- After making the scatter graphs, a line graph was made to determine the behavior of the success of the launches over time, from this graph it appears that since 2013, the success in the landings of stage 1 of the Falcon 9 have been increasing.

https://github.com/joseMCV/space-y/blob/e289e099c77fd5a5aba82732517da419df227843/space_y_visualizationseda.ipynb

EDA with SQL

To obtain relevant information on the data, various SQL queries listed below were performed:

- SELECT DISTINCT LAUNCH_SITE FROM SPACEXTBL
- SELECT * FROM SPACEXTBL WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5
- SELECT customer, SUM(payload_mass__kg_) as total_payload
  FROM SPACEXTBL
  WHERE customer = 'NASA (CRS)' GROUP BY customer
- SELECT BOOSTER_VERSION, AVG(payload_mass__kg_) as avg_payload
  FROM SPACEXTBL WHERE BOOSTER_VERSION = 'F9 v1.1'
  GROUP BY BOOSTER_VERSION
- SELECT MIN(DATE) FROM SPACEXTBL
  WHERE MISSION_OUTCOME LIKE '%Succ%'
- SELECT DISTINCT BOOSTER_VERSION
  FROM SPACEXTBL
  WHERE MISSION_OUTCOME LIKE '%Succ%'
  AND payload_mass__kg_ >6000
- SELECT MISSION_OUTCOME, COUNT(*)
  FROM SPACEXTBL
  GROUP BY MISSION_OUTCOME
- SELECT DISTINCT BOOSTER_VERSION
  FROM SPACEXTBL
  WHERE payload_mass__kg_ = (SELECT MAX(payload_mass__kg_) FROM SPACEXTBL)
- SELECT DATE, BOOSTER_VERSION, LAUNCH_SITE
  FROM SPACEXTBL
  WHERE LANDING__OUTCOME = 'Failure (drone ship)'
  AND YEAR(DATE) = 2015
- SELECT LANDING__OUTCOME, COUNT(*) AS outcome_counts
  FROM SPACEXTBL
  WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20'
  GROUP BY LANDING__OUTCOME
  ORDER BY count(*) DESC

https://github.com/joseMCV/space-y/blob/e289e099c77fd5a5aba82732517da419df227843/space_y_sqleda.ipynb

Build an interactive map with Folium

- To better understand the location of the launches, the visualizations of the python folium library are used, in which markers are used to understand where the launches occurred, these in turn are decorated with an icon that indicates whether or not that launch was successful. It is important to note that the markers were clustered so that the visualization was not overloaded, finally, lines were added to calculate the distances between the launches and the nearest cities.

https://github.com/joseMCV/space-y/blob/e289e099c77fd5a5aba82732517da419df227843/space_y_dash.ipynb

Build a Dashboard with Plotly Dash

- For the designed dashboard, two interactions and two different graphics were implemented, the interactions consisted of selecting a launch site through a dropdown and selecting the payload at launch through a range slider.
- The graphs designed were linked to the previously mentioned interactions, the pie graph when all the launch sites were selected, showed the total of successful launches for each launch site, however, if a specific launch site was selected, the graph it showed the number of times it had successful and failed landings. On the other hand, the Scatter chart was linked to both interactions where both were used as filters for the chart.
- These visualizations were added in order to improve the understanding of the data and the relationships that showed a greater correlation with the output of each landing of each launch.

https://github.com/joseMCV/space-y/blob/1b40332acda4b2b22836d24fb7303c06f0ccbda9/spacey_dash_app.py

Predictive analysis (Classification)

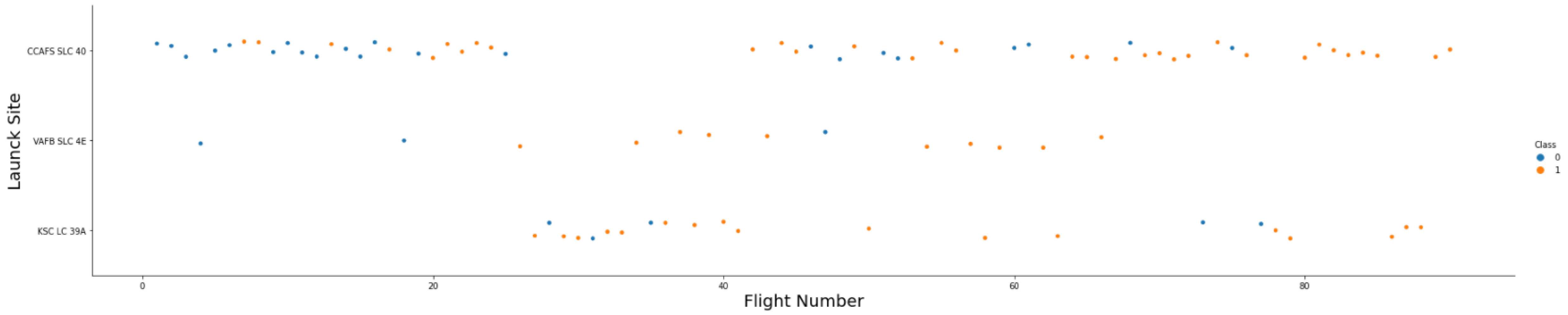
- The sklearn library was used to create 4 different models that would allow to perform the designated classification task, that is, to predict whether the Falcon 9 rocket phase one landing status.
- The four selected models were KNN, Logistic Regression, SVM and decision tree, for these four models a gridsearch method was used to find the hyperparameters that could deliver the best results over the training set, 10 iterations of cross validation were used to find the best parameters.
- Finally, the models were tested on the test set to determine their performance on data they had never "seen" before.
- The models used present an identical behavior with respect to the general precision, that is, the ability to predict whether stage 1 of the rocket will successfully land or not.

Results

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

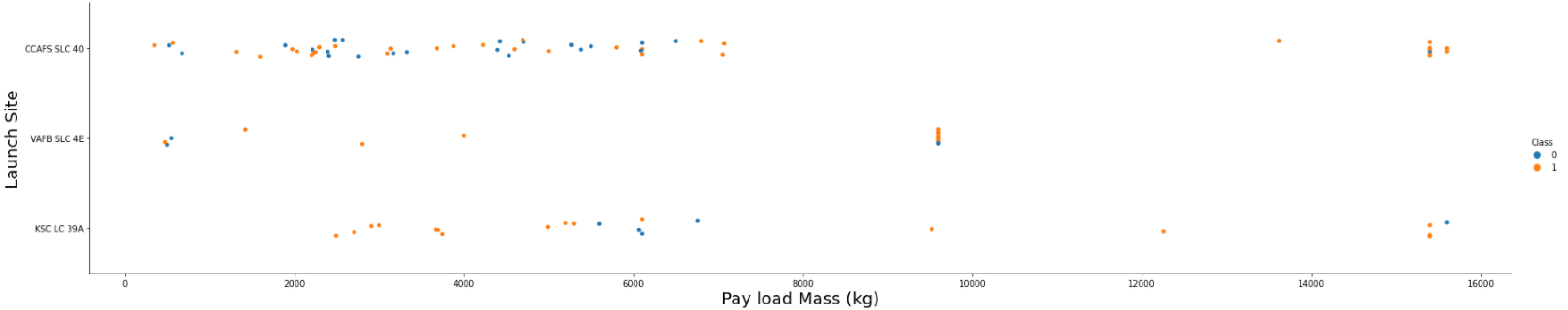
EDA with Visualization

Flight Number vs. Launch Site



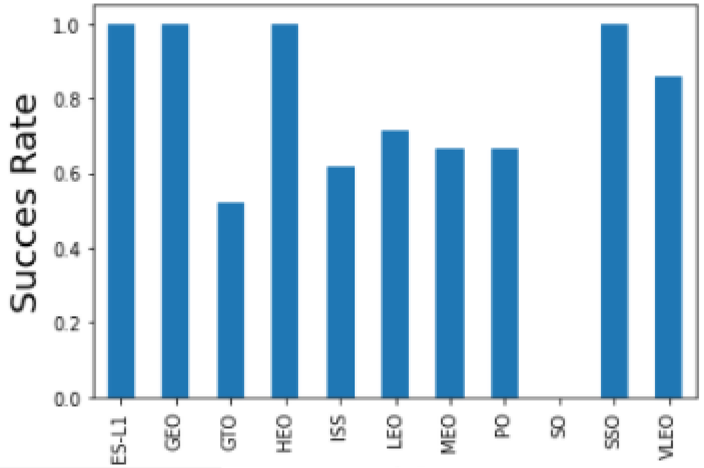
From the illustrated graph it can be seen that as the number of flights increases, so does the number of successful attempts, that is, the greater the number of flights for each of the launch sites, the greater the probability that the stage 1 of the rocket achieves a successful landing.

Payload vs. Launch Site



From the graph it can be seen that when the payload mass is greater than 8000, most of the flights end in a success, in addition to the KSC LC 39A launch site all launches with a payload mass less than 4000 end in success. CCAFS SLC 40 launches with a payload mass less than 800 have the most diverse behavior within the chart.

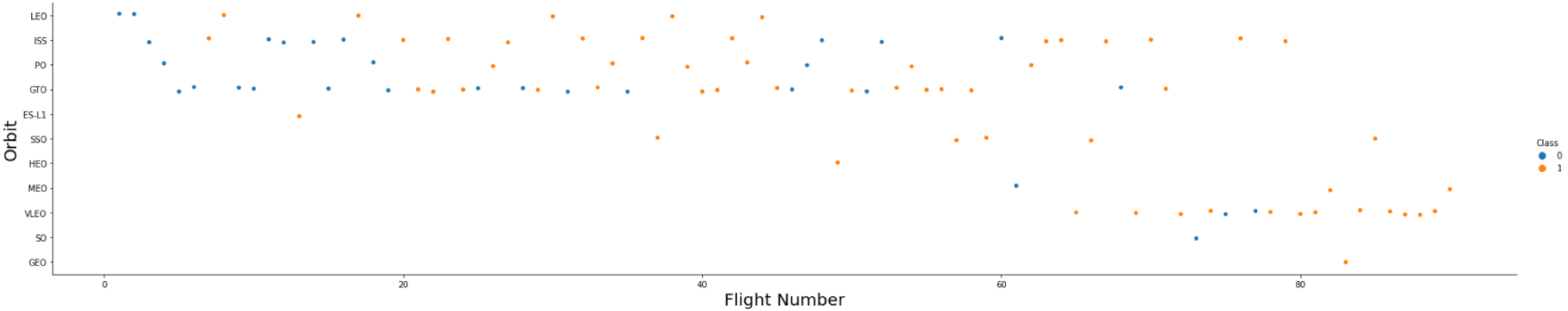
Success rate vs. Orbit type



Orbit	Success Rate	count
ES-L1	1.000000	1
GEO	1.000000	1
HEO	1.000000	1
SSO	1.000000	5
VLEO	0.857143	14
LEO	0.714286	7
MEO	0.666667	3
PO	0.666667	9
ISS	0.619048	21
GTO	0.518519	27
SO	0.000000	1

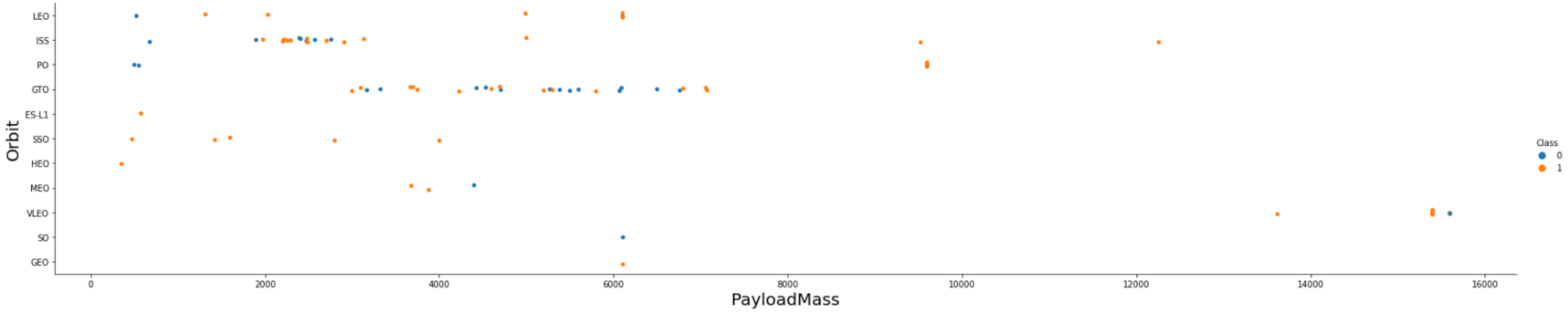
At first glance, one might think that the orbits with the highest success rate are ES-L1, GEO and HEO, however, only one launch has been made to each of these, which does not give us the necessary information to verify this statement. On the other hand, SSO and VLEO have success rates higher than 85%, already having a considerable number of flights, so they could be identified as the orbits with the highest success rate

Flight Number vs. Orbit type



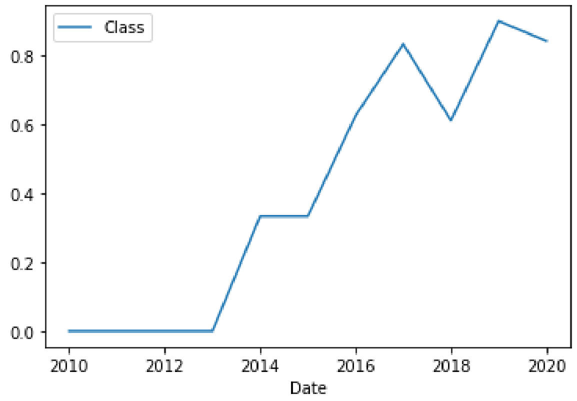
Nothing can be concluded from this graph at first glance, beyond the aforementioned relationship between the increase in the number of flights being proportional to the success rate.

Payload vs. Orbit type



For the LEO, ISS, PO orbits, the higher the payload, the higher the success rate, for the SSO orbit, all attempts have been successful, as has the ES-L1 orbit, so a relationship cannot be appreciated. with the payloadmass.

Launch success yearly trend



It can be seen that since 2013, the number of successful landings has increased, this is related to the experience and learnings obtained from previous missions, this supports the theory that the greater the number of flights, the greater the probability of success.

EDA with SQL

All launch site names

- The "DISTINCT" function is used to find the unique values in the requested column

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

```
 * ibm_db_sa://wfd01464:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/bludb
Done.
```

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

Launch site names begin with `CCA`

- The combination of the "WHERE" and "LIKE" functions is used to make sure we obtain the records of the release sites that begin with CCA, also the "LIMIT" function is used to only obtain 5 records.

In [14]:
```
%%sql SELECT * FROM SPACEXTBL
WHERE LAUNCH_SITE LIKE 'CCA%'
LIMIT 5
```

```
 * ibm_db_sa://wfd01464:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/bludb
Done.
```

Out[14]:

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

- The "SUM" function is used to obtain the sum of the payload, and the "WHERE" function is used to specify the customer. The "GROUP BY" function is used to make the SUM function work.

In [18]:
```
%%sql SELECT customer, SUM(payload_mass__kg_) as total_payload
FROM SPACEXTBL
WHERE customer = 'NASA (CRS)'
GROUP BY customer
```

```
 * ibm_db_sa://wfd01464:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/bludb
Done.
```

Out[18]:

customer	total_payload
NASA (CRS)	45596

Average payload mass by F9 v1.1

- The "AVG" function is used to obtain the average of the payload and the "WHERE" function is used to specify the customer, the "GROUP BY" function is used to make the "AVG" function work.

In [23]:
```
%%sql SELECT BOOSTER_VERSION, AVG(payload_mass__kg_) as avg_payload
FROM SPACEXTBL
WHERE BOOSTER_VERSION = 'F9 v1.1'
GROUP BY BOOSTER_VERSION
```

```
 * ibm_db_sa://wfd01464:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/bludb
Done.
```

Out[23]:

booster_version	avg_payload
F9 v1.1	2928

First successful ground landing date

- The "MIN" function is used to find the minimum date, and the where function to fulfill the condition of successful landing.

In [13]:
```
%%sql SELECT MIN(DATE) FROM SPACEXTBL
WHERE landing__outcome = 'Success (ground pad)'
```

```
 * ibm_db_sa://wfd01464:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/bludb
Done.
```

Out[13]:

1
2015-12-22

Successful drone ship landing with payload between 4000 and 6000

The "DISTINCT" function is used to find the unique values in the requested column, along with this the "WHERE" function is used both to ensure the specified landing condition and to find the payload within the requested range, it should be noted that this is the same could be achieved using the "BETWEEN" function

```
In [16]: %%sql SELECT DISTINCT BOOSTER_VERSION
FROM SPACEXTBL
WHERE landing__outcome = 'Success (drone ship)'
AND payload_mass__kg_ >4000
AND payload_mass__kg_ <6000
```

```
 * ibm_db_sa://wfd01464:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/bludb
Done.
```

Out[16]:

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

Total number of successful and failure mission outcomes

- The "COUNT" function is used to count the number of landings in each category (successful or failure)

```
%%sql SELECT  MISSION_OUTCOME, COUNT(*)
FROM SPACEXTBL
GROUP BY MISSION_OUTCOME
```

```
 * ibm_db_sa://wfd01464:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/bludb
Done.
```

]:

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

Boosters carried maximum **payload**

- In this case, a sub-query must be used to find the maximum payload condition, then the "WHERE" function is used to find the BOOSTER_VERSIONS that meet the maximum payload condition.

Task 8

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

In [36]:

```
%%sql SELECT DISTINCT BOOSTER_VERSION
FROM SPACEXTBL
WHERE payload_mass__kg_ = (SELECT MAX(payload_mass__kg_) FROM SPACEXTBL)
```

```
 * ibm_db_sa://wfd01464:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/bludb
Done.
```

Out[36]:

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

2015 launch records

- In this query the "WHERE" function is used to fulfill the landing_outcome condition, and the "WHERE" and "YEAR" function for the condition about the year 2015

Task 9

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

In [41]:
```
%%sql SELECT DATE, BOOSTER_VERSION, LAUNCH_SITE
FROM SPACEXTBL
WHERE LANDING__OUTCOME = 'Failure (drone ship)'
AND YEAR(DATE) = 2015
```

```
 * ibm_db_sa://wfd01464:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/bludb
Done.
```

Out[41]:

DATE	booster_version	launch_site
2015-01-10	F9 v1.1 B1012	CCAFS LC-40
2015-04-14	F9 v1.1 B1015	CCAFS LC-40

Rank success count between 2010-06-04 and 2017-03-20

- The "COUNT" function is used to be able to do a proper ranking, together with a "WHERE" and "BETWEEN" function to meet the established requirements.

In [44]:
```
%%sql SELECT LANDING__OUTCOME, COUNT(*) AS outcome_counts
FROM SPACEXTBL
WHERE DATE BETWEEN  '2010-06-04' AND '2017-03-20'
GROUP BY LANDING__OUTCOME
ORDER BY count(*) DESC
```

```
 * ibm_db_sa://wfd01464:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/bludb
Done.
```

Out[44]:

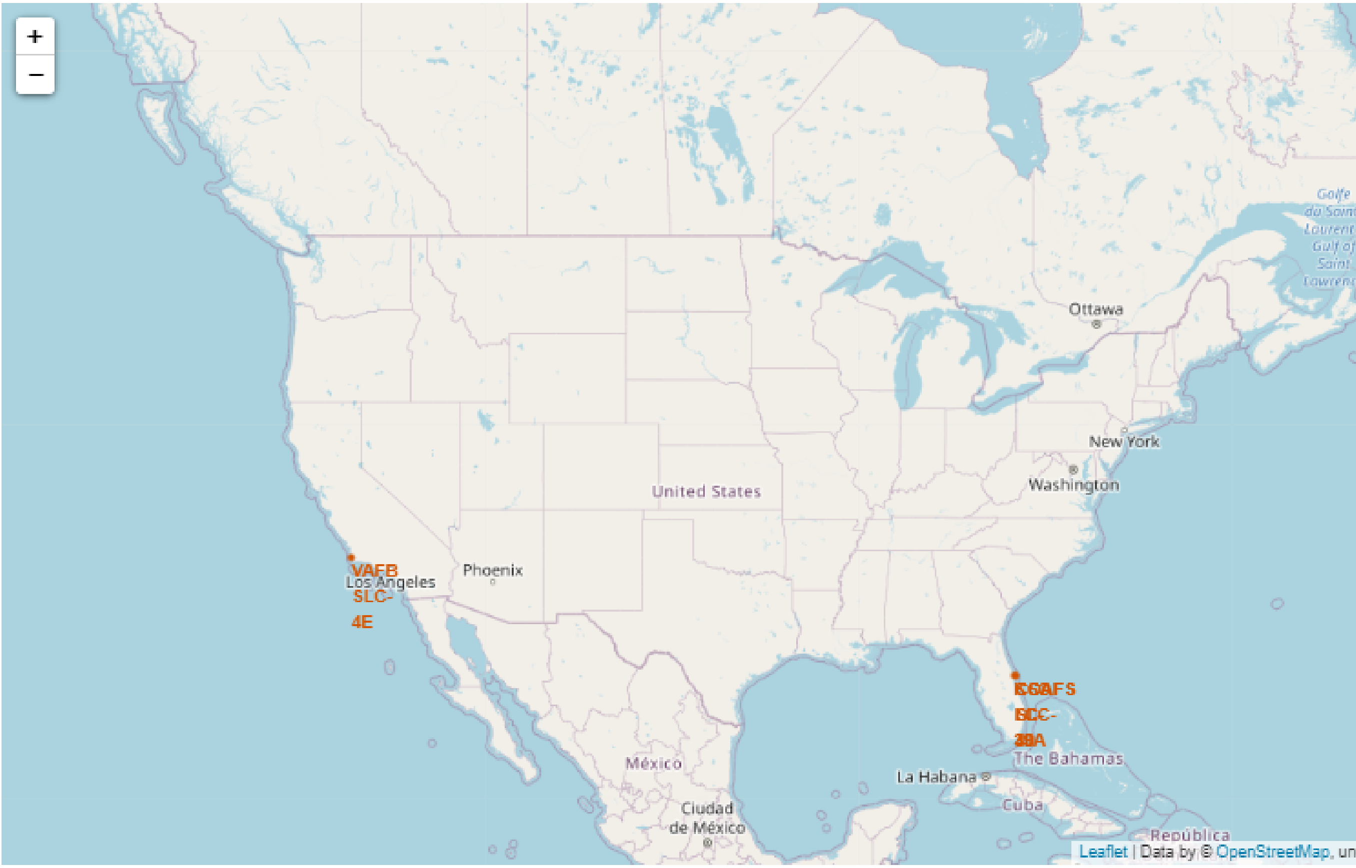
landing__outcome	outcome_counts
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

Interactive map with Folium

Launch Sites

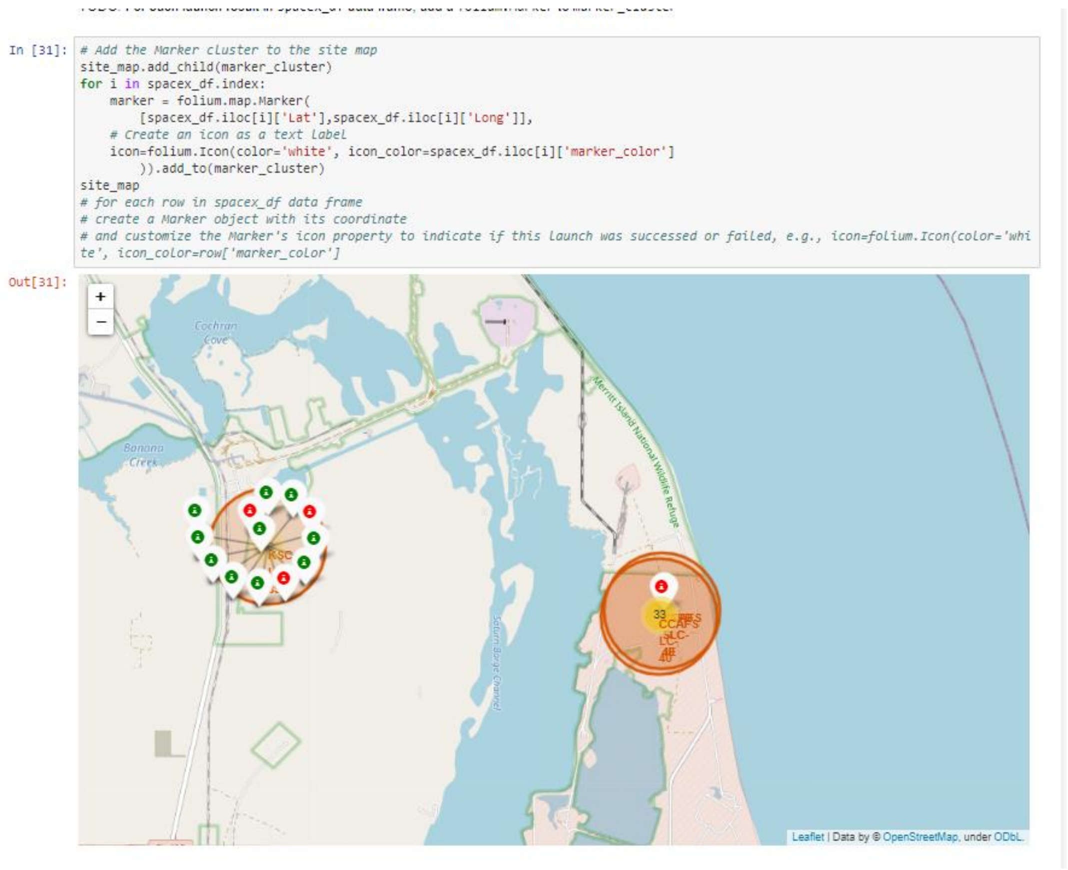
- In this map you can see the launch sites studied and their geographic distributions.
- As can be seen, they are all found in the United States, specifically in Los Angeles and Florida, in both cases areas that allow landings in the ocean.

```
# Initial the map
site_map = folium.Map(location=nasa_coordinate, zoom_start=5)
# For each launch site, add a Circle object based on its coordinate (Lat, Long) values. In addition, add Launch site nam
opup label
for lon,lat,site in zip(launch_sites_df.Long, launch_sites_df.Lat, launch_sites_df['Launch Site']):
    # Create a blue circle at NASA Johnson Space Center's coordinate with a popup label showing its name
    circle = folium.Circle([lat,lon], radius=1000, color='#d35400', fill=True).add_child(folium.Popup(site))
    # Create a blue circle at NASA Johnson Space Center's coordinate with a icon showing its name
    marker = folium.map.Marker(
        [lat,lon],
    # Create an icon as a text label
    icon=DivIcon(
        icon_size=(20,20),
        icon_anchor=(0,0),
        html='<div style="font-size: 12; color:#d35400;"><b>%s</b></div>' % site,
        )
    )
    site_map.add_child(circle)
    site_map.add_child(marker)
site_map
```



Launch Site Outcomes

- In this map, in addition to being able to see the geographical distribution of the launch sites, it is also possible to observe the number of successful and failed launches in each one, these are observed in the colors of the icons indicated in each launch site, where the icons Green symbolizes a successful launch, and a red icon symbolizes a failed launch.

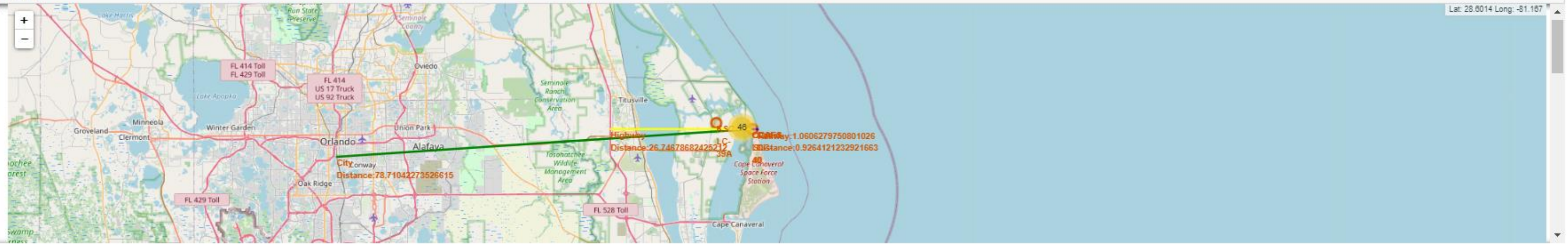


<Folium map screenshot 3>

- In the graphical map you can see relevant areas near the launch site, where it is identified that the populated areas are far from it due to security issues, also it is distinguished that the launch sites are in coastal areas so that the remnants rockets can land or fall directly into the ocean and cause no collateral damage.

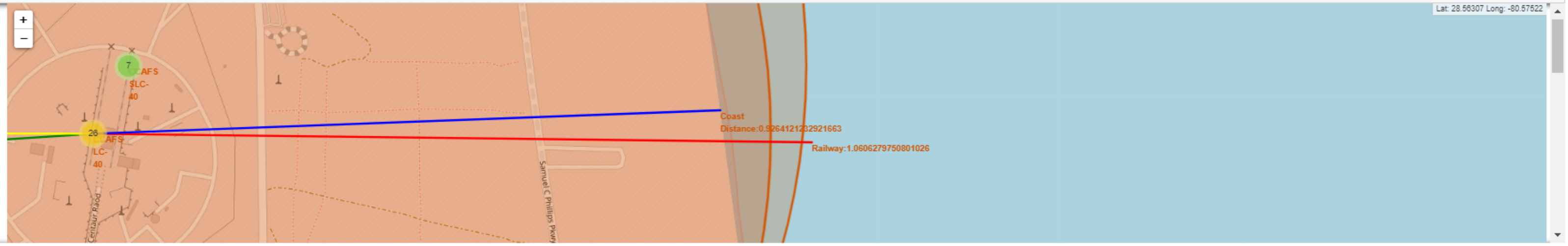
```
    icon_size=(20,20),
    icon_anchor=(0,0),
    html='<div style="font-size: 12; color:#d35400;"><b>%s</b></div>' % 'Coast Distance:{}'.format(distance_airport),
    )
)
site_map.add_child(marker)
folium.PolyLine([launch_site,airport], color='blue').add_to(site_map)
site_map
```

Out[90]:



```
    icon_size=(20,20),
    icon_anchor=(0,0),
    html='<div style="font-size: 12; color:#d35400;"><b>%s</b></div>' % 'Coast Distance:{}'.format(distance_airport),
    )
)
site_map.add_child(marker)
folium.PolyLine([launch_site,airport], color='blue').add_to(site_map)
site_map
```

Out[90]:



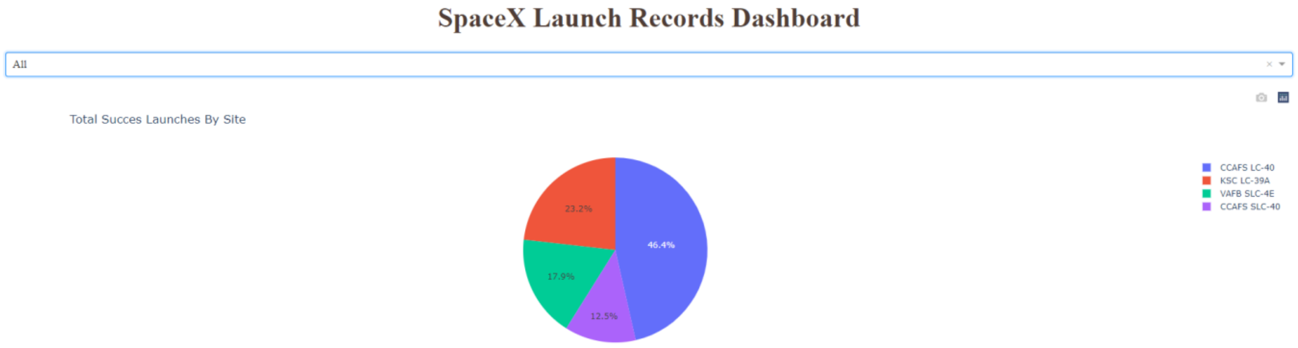

```
After you plot distance lines to the proximities, you can answer the following questions easily:

*   Are launch sites in close proximity to railways?
*   Are launch sites in close proximity to highways?
*   Are launch sites in close proximity to coastline?
*   Do launch sites keep certain distance away from cities?

Also please try to explain your findings.
```

As can be seen on the map, the launch centers are located near the coastlines and railways, this is due to the need to launch in areas where the rocket stages can fall into the sea and not alter the Nobody's safety, it can also be seen that the launch sites are close to the railways so that the parts and materials needed for the launches can be delivered efficiently. Finally, on the map it can be seen that the launch sites are far from populated areas such as large cities, that is, again, due to security issues.

Build a Dashboard with Plotly Dash

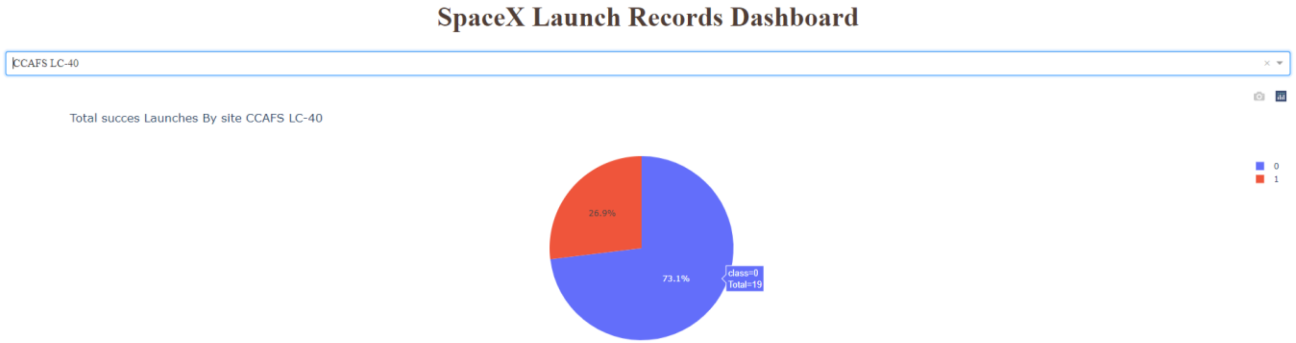
SPACEX Launch Dashboard – Pie Chart

This graph tells us directly which are the launch sites that show the highest number of successful launches, and their proportion with respect to the total number of successful launches.



SPACEX Launch Dashboard – Most Successful Launch Site

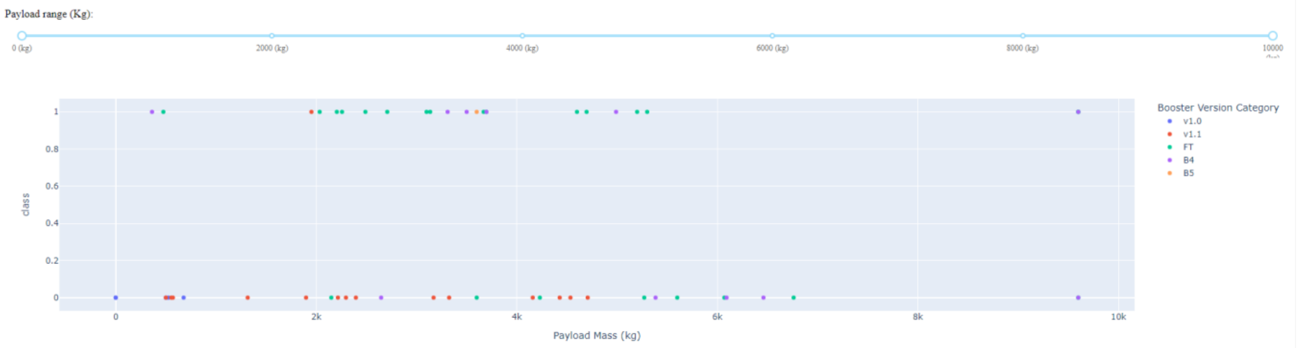
From the illustrated figure it is distinguished that the CCAFS LC-40 launch site has a success rate of 73.1% with a total of 19 launches with successful results.

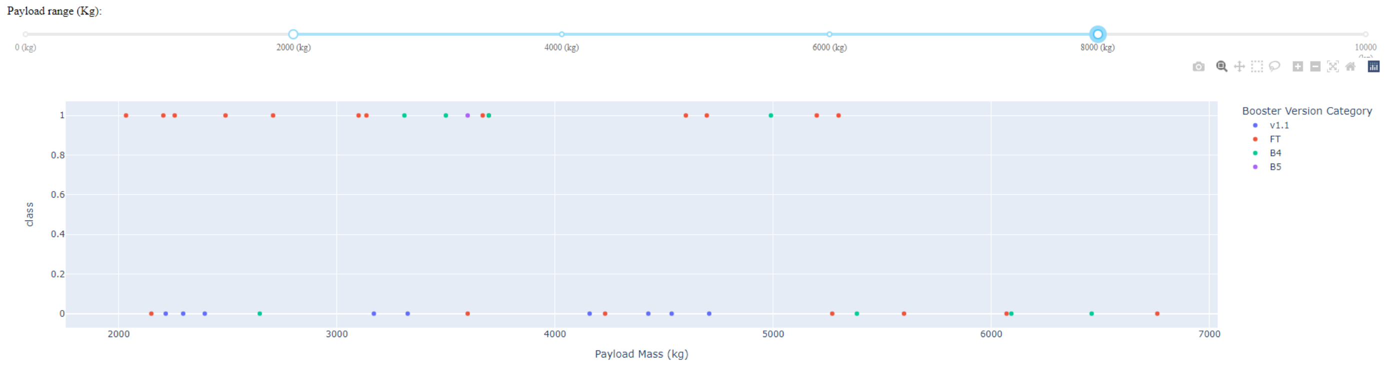


SPACEX Launch Dashboard – Scatter Plot

It can be determined that for all launch sites, if the payload is between 6000 and 9000 kg, the mission is very likely to be a success.

For the versions of Booster v1.0 and v1.1, it is distinguished that most missions were a failure, regardless of the mission payload, for the FT version, most missions with payloads between 2000 and 4000 kg were successful , while when the payload was outside these ranges, its behavior was varied. For the version of Booster B4, it is observed that it has a favorable behavior between 3000 and 5000 kg of payload, while for the rest of the range of payloads, it shows a rather negative behavior, finally, the version of booster B5 has only one launch, so not much can be concluded from its behavior.

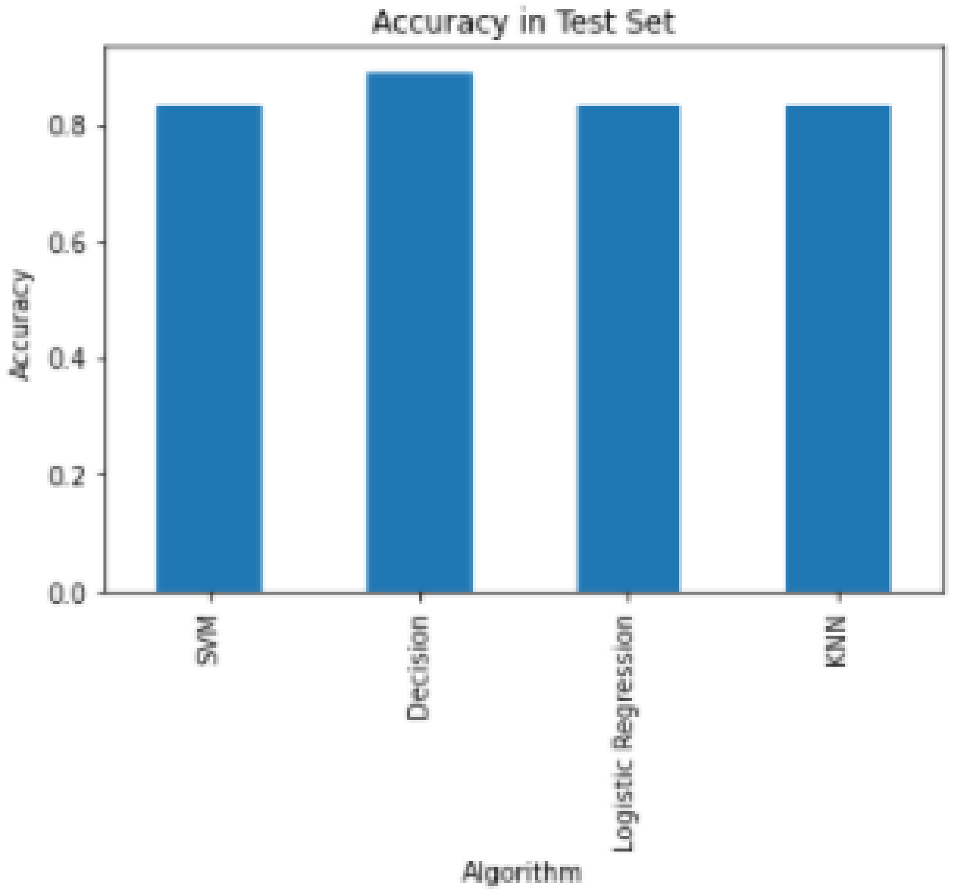




Predictive analysis (Classification)

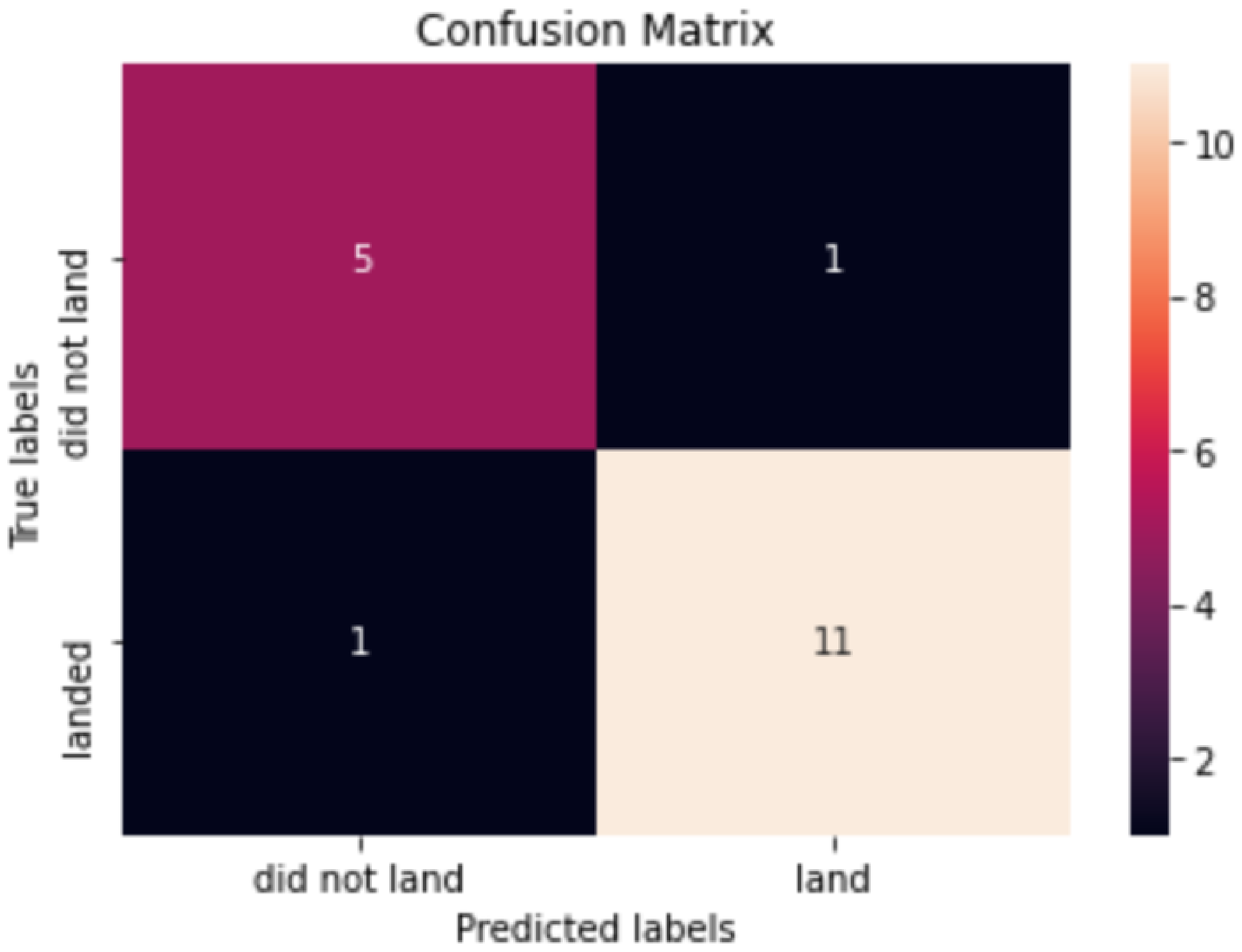
Classification Accuracy

As seen in the image, the model with the best performance in the test set corresponds to a decision tree, so this is the selected model.



Confusion Matrix

As can be seen in the confusion matrix, the algorithm is able to correctly determine the classes 88% of the time, however, in the cases where it is wrong (2), it shows a false positive and a false negative, where the The first of these is the most dangerous case, since the algorithm indicates that the landing would be successful, and finally it is not, which would cause millions of dollars in losses.



CONCLUSION

- It is possible to determine whether the Falcon 9 phase 1 landing will end successfully with reasonably high precision.
- Many of the variables obtained from various data sources were used to make the prediction of the selected model more accurate.
- Although the developed model presents good results, these could always be improved by adding various variables and sources of information, such as the weather on the day the launch was made, or the wind speed.
- The visualizations made were of the utmost importance to understand the behavior of the variables and their relationship with the dependent variable to be predicted.

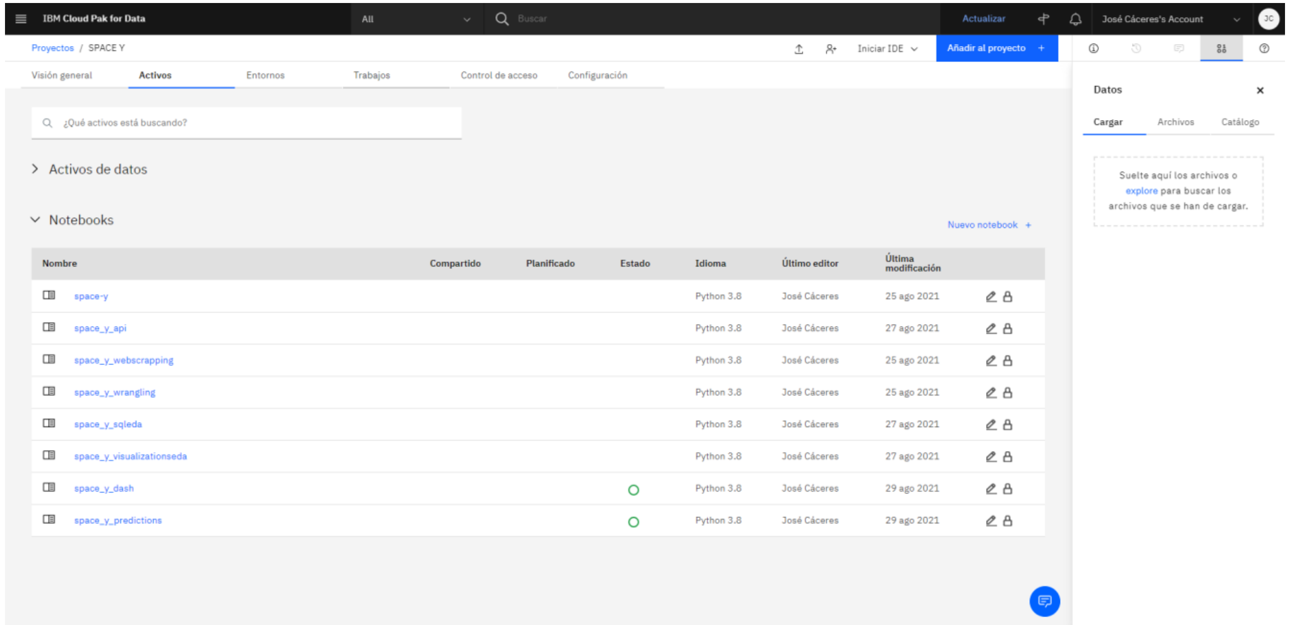
APPENDIX: Git Repository

- The link to the repository is at the link below.

https://github.com/joseMCV/space-y

APPENDIX: Plataform

- All the codes created for this work were developed on the IBM platform, in the Watson Studio tool.



APPENDIX: Libraries

- The python libraries used for this project are listed below:
  - Pandas
  - Numpy
  - Sklearn
  - Folium
  - Dash
  - Requests
  - BeautifullSoup
  - Matplotlib
  - Plotly