IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Pedro Barcellos
03/04/2022

Outline

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

Executive Summary

- Summary of methodologies
  - Usage of python programming to create a Data Collection through API
  - Web Scrapping with python programming adding to the created Data Collection
  - Data Wrangling (clean-up \ standardization)
  - Exploratory Data Analysis (EDA) using Structured Query Language (SQL)
  - EDA using various forms of Visualization with python programming (plotly)
  - Interactive Visual Analytics using Folium (python programming)
  - Machine Learning Prediction
- Summary of all results
  - EDA results
  - Screenshots from the Interactive Visual Analysis
  - Predictive Analytics result

Introduction

- Project background and context

  SpaceX has gained worldwide attention for a series of historic milestones.

  It is the only private company ever to return a spacecraft from low-earth orbit, which it first accomplished in December 2010.

  SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage. Therefore, if we can determine if the first stage will land, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch.

- Problems you want to find answers

  - What factors are decisive / key for a successful landing?
  - Is there any relation between this factors that determines a successful landing?
  - If so, what is the combination with higher chance for a successful landing?

Section 1

Methodology

Methodology

Executive Summary

- Data collections were created by using python programming to retrieve data from SpaceX API and via web scraping Wikipedia (List of Falcon 9 and Falcon Heavy launches)
- Data collected from the API was then "processed" using python Pandas data frames removing the unwanted columns and removing null values – Wangling
- Initial EDA using SQL has paved the way to a more in-depth analysis since some relations with the data were already showing up.
- These relations were then visually checked in an interactive dashboard using Folium and Plotly Dash
- Finally predictive analysis using classification models corroborate to identify and suggest the relations we look for

Data Collection

Data collection was done using various methods via python programming. Utilizing requests library, we retrieved data from SpaceX API in json format, which was then “translated” into Panda data frames creating a “table” from the attributes and values presented on the json response.

Then we narrowed the table contents by excluding unwanted columns, and by calculating some of the values using given premade functions.

Web scraping was performed to gather data from Falcon 9 launch using python BeautifulSoup which were then converted also into Panda data frames to complement the analysis.

Data wrangling was next, when null values were replaced on the dataset by its column mean value (PayloadMass), and initial counts on the grouping of values per Launch Site, Orbit and Outcome, enabled us to classify the data adding a column with the successful (1) of failed (0) outcome.

Data Collection – SpaceX API

- Using request method and creating a normalized Panda data frame from SpaceX API json response.

- GitHub link: [SpaceX Collecting Data]()

Task 1: Request and parse the SpaceX launch data using the GET request

To make the requested JSON results more consistent, we will use the following static response object for this project:

In [40]:
```
static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_call_spacex_api.json'
```

We should see that the request was successfull with the 200 status response code

In [41]:
```
response.status_code
```
Out[41]: 200

Now we decode the response content as a Json using `.json()` and turn it into a Pandas dataframe using `.json_normalize()`

In [42]:
```
# Use json_normalize meethod to convert the json result into a dataframe
#response.json()
data=pd.json_normalize(response.json())
```

Using the dataframe `data` print the first 5 rows

In [43]:
```
# Get the head of the dataframe
data.head(5)
```

Out[43]:

	static_fire_date_utc	static_fire_date_unix	net	window	rocket	success	failures	details	crew	ships	capsules	
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	[]	[]	[]	[

Data Collection - Scraping

- Using python BeautifulSoup library we performed webscraping of the Falcon 9 launch records from Wikipedia, parsing the html format into a Panda data frame.
- GitHub link: Web Scrapping

	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 (COTS)\nNRO	Success	F9 v1.0B0004.1	Failure	8 December 2010	15:43
2	3	CCAFS	Dragon	525 kg	LEO	NASA (COTS)	Success	F9 v1.0B0005.1	No attempt\n	22 May 2012	07:44
3	4	CCAFS	SpaceX CRS-1	4,700 kg	LEO	NASA (CRS)	Success\n	F9 v1.0B0006.1	No attempt	8 October 2012	00:35
4	5	CCAFS	SpaceX CRS-2	4,877 kg	LEO	NASA (CRS)	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 Capella Space and Tyvak	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 (CRS)	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

121 rows × 11 columns

Data Wrangling

EDA was done on top of the data collection of the API combined with the scraped from the Wiki, on form of Panda data frames where null values for PayloadMass were replaced by its mean value, and classification of the Outcome was turned into a new column / attribute called Class where 0 denotes a failed result and 1 a successful one. That allowed us to initiate quantitative analysis of number of successful and unsuccessful launches per Launch Site, and its occurrences per Orbit.

- GitHub link: [Data Wrangling]

This variable will represent the classification variable that represents the outcome of each launch. If the value is zero, the first stage did not land successfully; one means the first stage landed Successfully

In [62]:
```
df['Class']=landing_class
df[['Class']].head(8)
```

Out[62]:

	Class
0	0
1	0
2	0
3	0
4	0
5	0
6	1
7	1

In [64]:
```
df.head(8)
```

Out[64]:

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	Reused(
0	1	2010-06-04	Falcon 9	6104.959412	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0
1	2	2012-05-22	Falcon 9	525.000000	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0
2	3	2013-03-01	Falcon 9	677.000000	ISS	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0
3	4	2013-09-29	Falcon 9	500.000000	PO	VAFB SLC 4E	False Ocean	1	False	False	False	NaN	1.0	0
4	5	2013-12-03	Falcon 9	3170.000000	GTO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0
5	6	2014-01-06	Falcon 9	3325.000000	GTO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0
6	7	2014-04-18	Falcon 9	2296.000000	ISS	CCAFS SLC 40	True Ocean	1	False	False	True	NaN	1.0	0
7	8	2014-07-14	Falcon 9	1316.000000	LEO	CCAFS SLC 40	True Ocean	1	False	False	True	NaN	1.0	0

We can use the following line of code to determine the success rate:

In [65]:
```
df["Class"].mean()
```

Out[65]:
```
0.6666666666666666
```

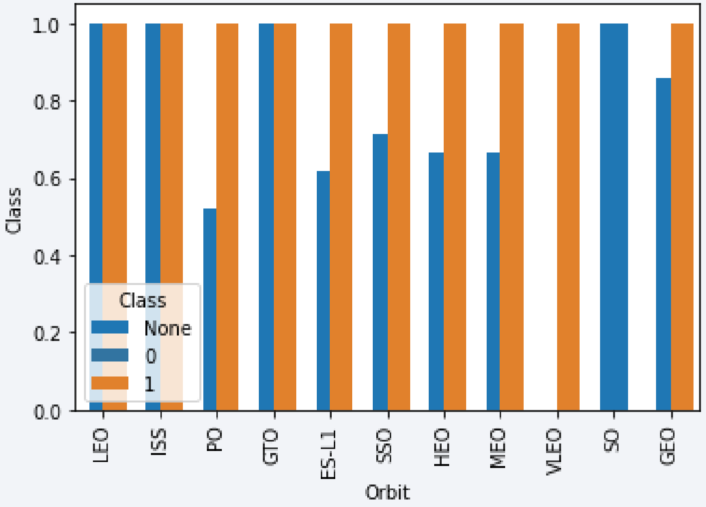
EDA with Data Visualization

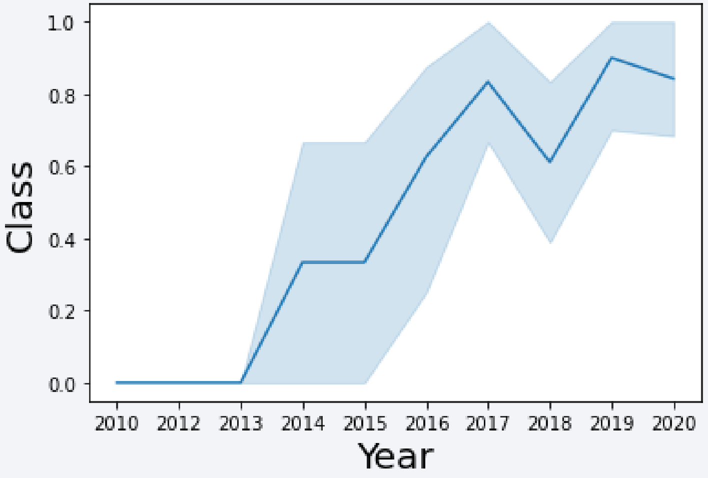
Here are some of the findings possible with the visual analysis of the linear, bar and scatter plots. By relating Flight number and Launch Site, Orbit and Success ratio (Class).

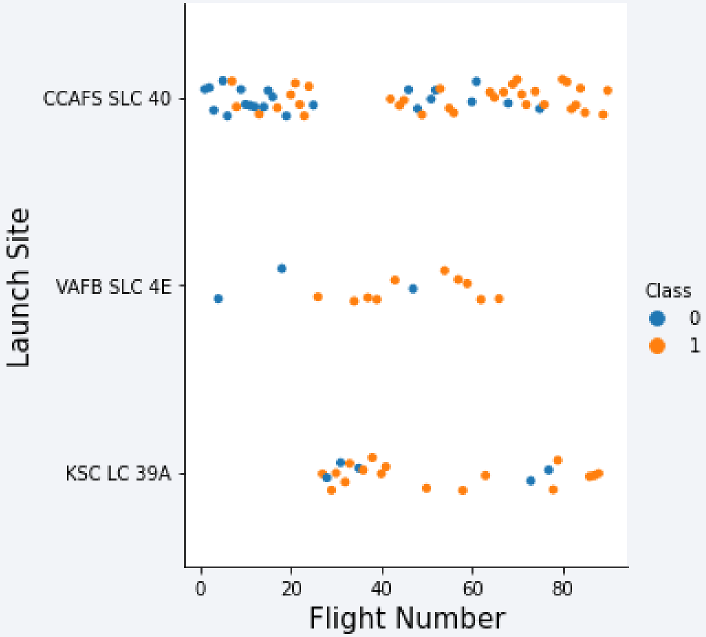
As a summary we could say that:

- The success ratio of SpaceX launches increased in time. (Class x Year graph on next slide)
- Recent launches (flight numbers > 40) were more prone to success on CCAFS SLC 40 Launch Site;
- Early launches (flight numbers 0 to 25) were more prone to fail and were also launched from CCAFS SLC 40;
- Fewer launches were done on VAFB SLC 4E than in any other launch site, which proportionally gives it a better success ratio;
- KSC LC 39A launch site was only used after the last “twentieths” flight number, failed on CCAFS SLC 40;
- Activities returned to launch site CCAFS SLC 40 after the flight number 40 and a good success ratio started after flight number 60.
- Orbit PO has higher success rate, followed by ES-L1 then HEO and MEO to SSO then GEO.
- LEO, ISS, GTO tie on success rate.
- VLEO orbit has only success.
- SO orbit has only failed.
- GitHub link: [Data Visualization]

EDA with Data Visualization







EDA with SQL

- We loaded the SpaceX csv dataset into IBM PostgreSQL database; GitHub link: [EAD with SQL]
- We used SQL to get some insights from the data:
  - The names of the unique launch sites in the space mission
  - First 5 records where launch sites beginning with the string 'CCA'
  - Total payload mass carried by boosters launched by NASA (CRS)
  - Average payload mass carried by booster version F9 v1.1
  - The date when the first successful landing outcome in ground pad was achieved.
  - Names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
  - Total number of successful and failure mission outcomes
  - Names of the booster versions which have carried the maximum payload mass.
  - The failed landing outcomes in drone ship, their booster versions, and launch site names for in year 2015
  - 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

Build an Interactive Map with Folium

- We plot all Launch Sites on a map using circles, and by using map objects we added markers for success or failed launches, (based on the classification of the Outcome: Class where 0 is fail and 1 success), and lines denoting the calculated distance from launch sites to highways, railroads and closest city. And by using clusters, we shown which launch site had higher success rate.
- The goal was to answer some of these questions (where we found it is true):
- Are launch sites near railways, highways and / or coastline?
- Do launch sites keep certain distance away from cities?
- GitHub link: [Interactive Map with Folium]

Build a Dashboard with Plotly Dash

- Dropdown list of Launch Sites for selection and input for the other components that were filtered dynamically
- Pie chart to show the total successful launches count for all sites and if / when a specific launch site was selected, show the Success vs. Failed counts for the site
- Horizontal slider to select payload range
- Scatter chart to show the correlation between payload and launch success
- The main goal was to visualize the correlation between Launch site, Payload and Success rate for the SpaceX launches.
- GitHub link: [Dashboard app]

Build a Dashboard with Plotly Dash

Predictive Analysis (Classification)

- Data was loaded to data frames using Panda and Numpy, then the data frame was split into test and training data.
- A few different machine learning models were built and parametrized using GridSearchCV
- The score and accuracy of the models was calculated, and the models improved by it with the usage of Logistics Regression followed by Support Vector machine, Decision Tree, Confusion matrix, Nearst Neighbors
- Github link: Predictive Analysis Classification

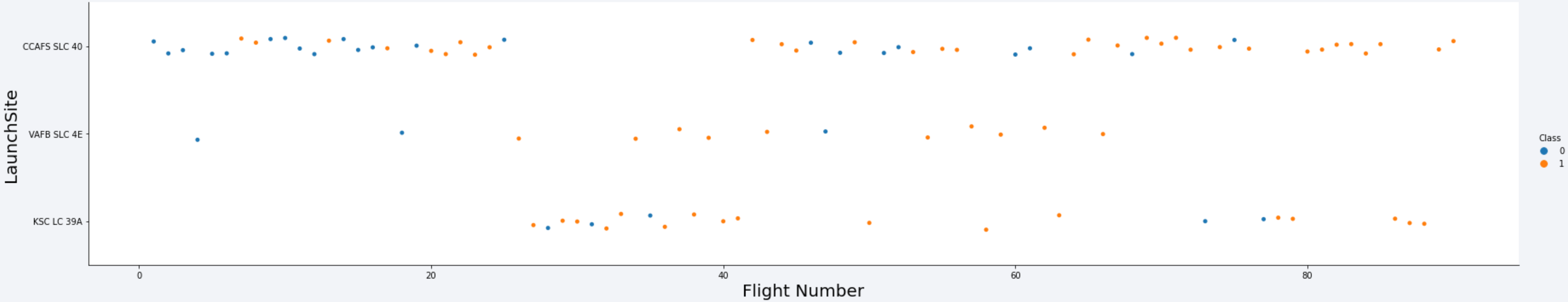
Results

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

Section 2

Insights drawn from EDA

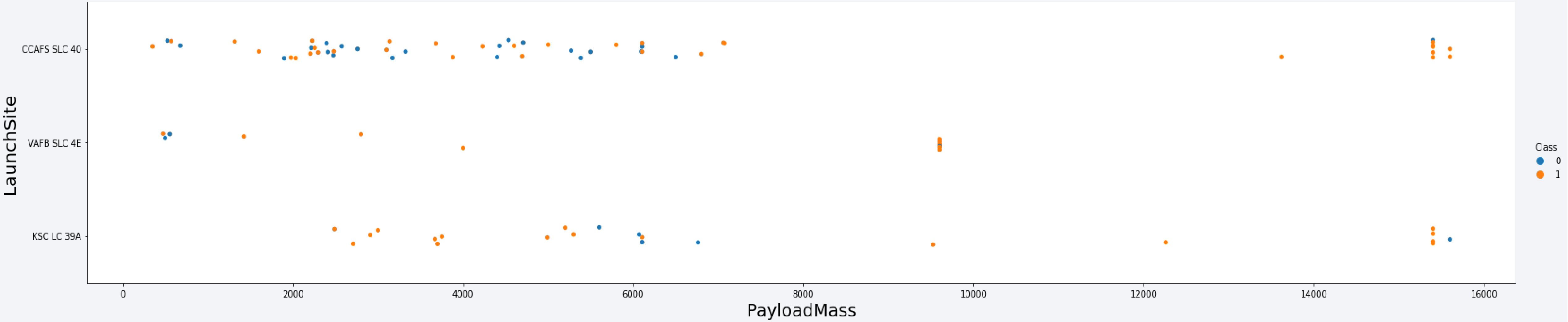
Flight Number vs. Launch Site



Scatter plot of Flight Number vs. Launch Site

- Recent launches (flight numbers > 40) were more prone to success on CCAFS SLC 40 Launch Site;
- Early launches (flight numbers 0 to 25) were more prone to fail and were also launched from CCAFS SLC 40;
- Fewer launches were done on VAFB SLC 4E than in any other launch site, which proportionally gives it a better success ratio;
- KSC LC 39A launch site was only used or started its activities after the last "twentieths" flight number failed on CCAFS SLC 40 site;
- Activities returned to launch site CCAFS SLC 40 after the flight number 40 and a good success ratio started after flight number 60.

Payload vs. Launch Site



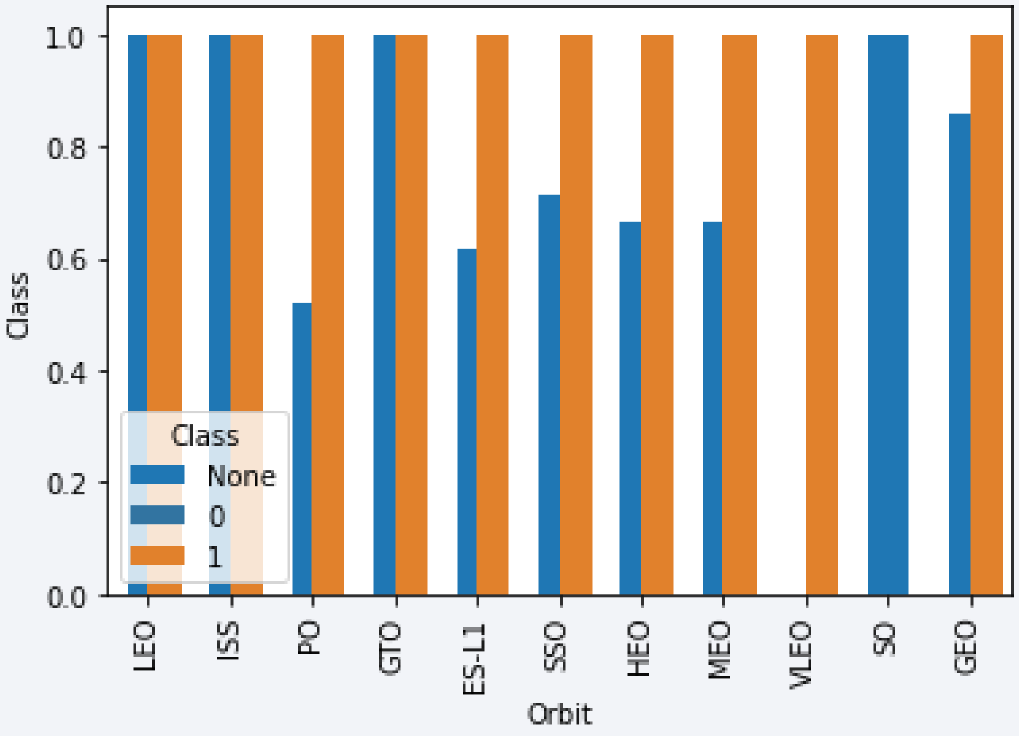
Scatter plot of Payload vs. Launch Site

- For the VAFB-SLC launch site there are no rockets launched for heavy payload mass(greater than 10000)
- The greater the payload for CCAFS SLC 40 higher the success rate.

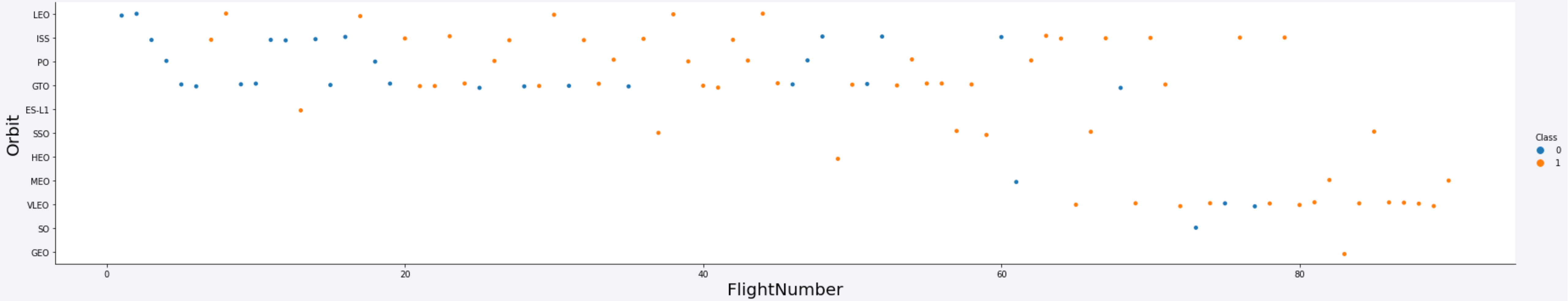
Success Rate vs. Orbit Type

Bar chart for the success rate of each orbit type

- Orbit PO has higher success rate, followed by ES-L1 then HEO and MEO to SSO then GEO.
- LEO, ISS, GTO tie on success rate.
- VLEO orbit has only success.
- SO orbit has only failed.



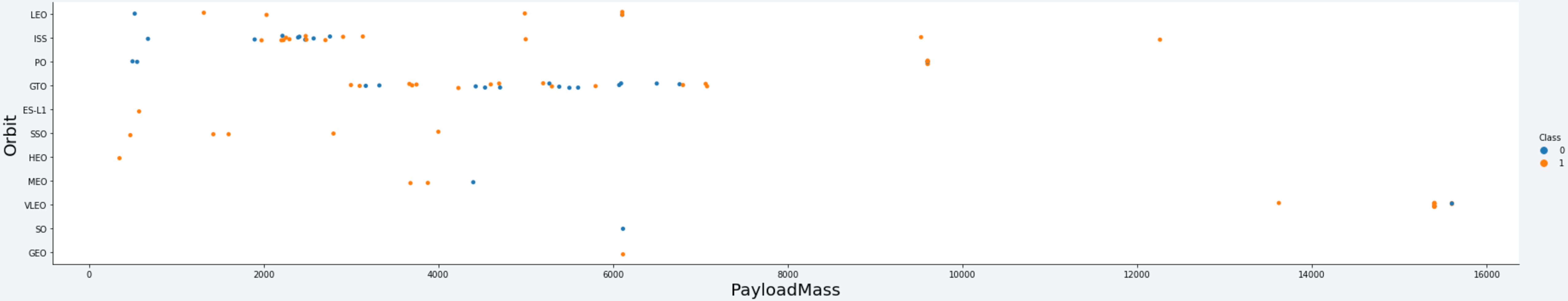
Flight Number vs. Orbit Type



Scatter plot of Flight number vs. Orbit type

- 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



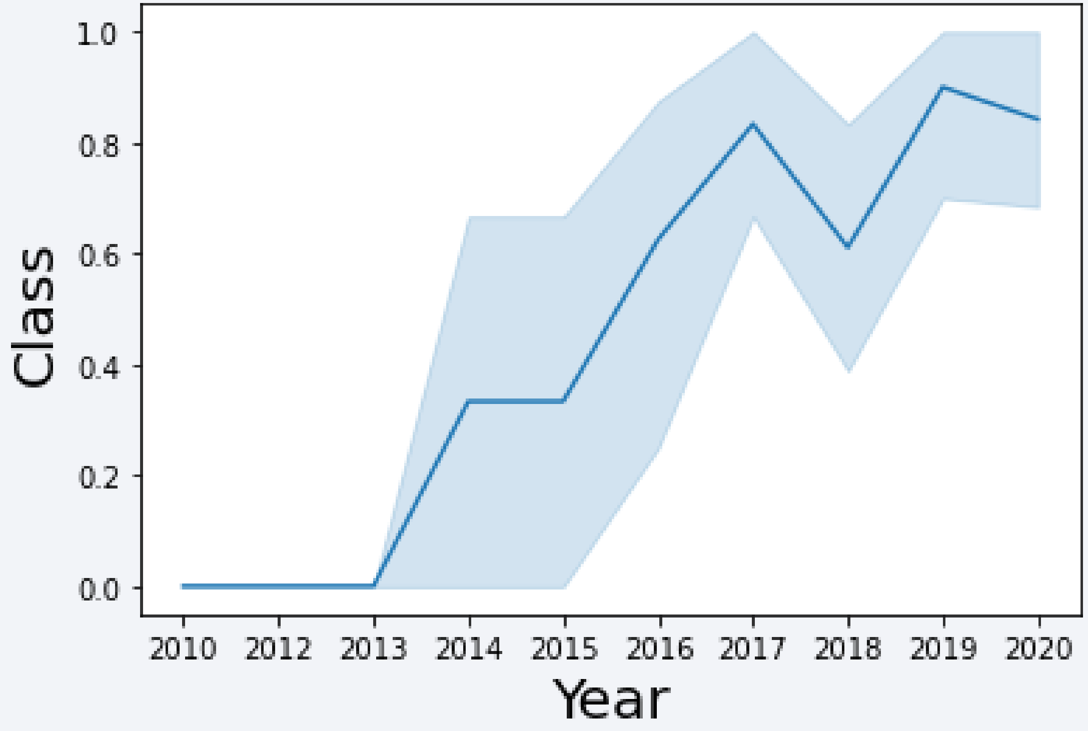
Scatter plot of payload vs. orbit type

There is an increase of the successful landing accompanied by an increase on the payload for the orbits PO, LEO and ISS

Launch Success Yearly Trend

Line chart of yearly average success rate

- From 2013 onwards there was a constantly increase on the success rate for SpaceX rocket launches with a slight decline in 2018 which was soon recovered in 2019



All Launch Site Names

- Data selection using SQL using the function DISTINCT

```
In [6]: %sql select distinct(LAUNCH_SITE) from spacextbl;

 * ibm_db_sa://wlw41726:***@824dfd4d-99de-440d-9991-629c01b3832d.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:30119/bludb
Done.

Out[6]:
```

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

Launch Site Names Begin with 'CCA'

- Data selection using SQL using the function LIMIT and the operator LIKE returning the first 5 records of launch sites beginning with `CCA`

In [7]:
```
%sql select * from spacextbl where LAUNCH_SITE like 'CCA%' limit 5;
```

```
 * ibm_db_sa://wlw41726:***@824dfd4d-99de-440d-9991-629c01b3832d.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:30119/bludb
Done.
```

Out[7]:

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

- Data selection using SQL to calculate the total payload carried by boosters from NASA using function SUM and where clause customer equal ‘NASA (CRS)’

In [15]:
```
%sql select sum(PAYLOAD_MASS__KG_) as total_payload_mass_kg from spacextbl where customer='NASA (CRS)';
```

```
 * ibm_db_sa://wlw41726:***@824dfd4d-99de-440d-9991-629c01b3832d.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:30119/bludb
Done.
```

Out[15]:

total_payload_mass_kg
45596

Average Payload Mass by F9 v1.1

- Data selection using SQL to calculate the average payload mass carried by booster version F9 v1.1, using function AVG and where clause booster version equal 'F9 v1.1'

```
In [16]: %sql select AVG(PAYLOAD_MASS__KG_) as avg_payload_mass_kg from spacextbl where booster_version='F9 v1.1';
 * ibm_db_sa://wlw41726:***@824dfd4d-99de-440d-9991-629c01b3832d.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:30119/bludb
Done.
Out[16]:
```

avg_payload_mass_kg
2928

First Successful Ground Landing Date

- Data selection using SQL to find the dates of the first successful landing outcome on ground pad (2015-12-22), utilizing a simple where clause on the attribute / column landing__outcome and the function MIN on date

```
In [20]: %sql select min(date) from spacextbl where landing__outcome='Success (ground pad)';

 * ibm_db_sa://wlw41726:***@824dfd4d-99de-440d-9991-629c01b3832d.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:30119/bludb
Done.

Out[20]:
```

1
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- Data selection using SQL to list the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000, utilizing function DISTINCT on booster_version and BETWEEN for the desired values of payload_mass__kg

```
In [21]: %sql select distinct(booster_version) from spacextbl where landing__outcome='Success (drone ship)' and PAYLOAD_MASS__KG_ BETWEEN
4000 AND 6000 ;

 * ibm_db_sa://wlw41726:***@824dfd4d-99de-440d-9991-629c01b3832d.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:30119/bludb
Done.

Out[21]:
```

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

Total Number of Successful and Failure Mission Outcomes

- Data selection using SQL to calculate the total number of successful and failure mission outcomes, via the function COUNT, creating a new column and GROUP BY on the mission_outcome
- A total of 100 success if you disconsider the payload status and a single (1) failure in flight

```
In [23]: %sql select mission_outcome, count(*) counter from spacextbl group by mission_outcome;

 * ibm_db_sa://wlw41726:***@824dfd4d-99de-440d-9991-629c01b3832d.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:30119/bludb
Done.

Out[23]:
```

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

Boosters Carried Maximum Payload

- Data selection with SQL to list the names of the booster which have carried the maximum payload mass, using DISTINCT function and a subquery to return the MAX payload.

In [26]:
```
%sql select distinct(booster_version) from (SELECT booster_version, PAYLOAD_MASS__KG_ FROM spacextbl WHERE PAYLOAD_MASS__KG_ = (
SELECT MAX(PAYLOAD_MASS__KG_) PAYLOAD_MASS__KG_ FROM spacextbl));
```

```
 * ibm_db_sa://wlw41726:***@824dfd4d-99de-440d-9991-629c01b3832d.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:30119/bludb
Done.
```

Out[26]:

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

- Data selection with SQL to list the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015, using a simple where clause stating the values for each column (landing__outcome and date), we also used the function / format YEAR for the date column.

```
In [32]: %sql select landing__outcome,booster_version,launch_site from spacextbl where landing__outcome='Failure (drone ship)' and YEAR(d
ate)='2015';

 * ibm_db_sa://wlw41726:***@824dfd4d-99de-440d-9991-629c01b3832d.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:30119/bludb
Done.
```

Out[32]:

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

- Data selection with SQL ranking the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order, by using the function COUNT and creating a column counter which returns the count, the function IN to specify the outcomes desired and the function BETWEEN to filter the date field, grouping it using the function GROUP BY on the outcome field, and ordering it with ORDER BY function.
- Please note the "such as" values described were used as filters on the data selection

```
In [39]: %sql select landing__outcome, count(*) counter from spacextbl where landing__outcome in ('Failure (drone ship)','Success (ground
pad)') and date BETWEEN '2010-06-04' and '2017-03-20' group by landing__outcome order by counter desc;

 * ibm_db_sa://wlw41726:***@824dfd4d-99de-440d-9991-629c01b3832d.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:30119/bludb
Done.

Out[39]:
```

landing__outcome	counter
Failure (drone ship)	5
Success (ground pad)	3

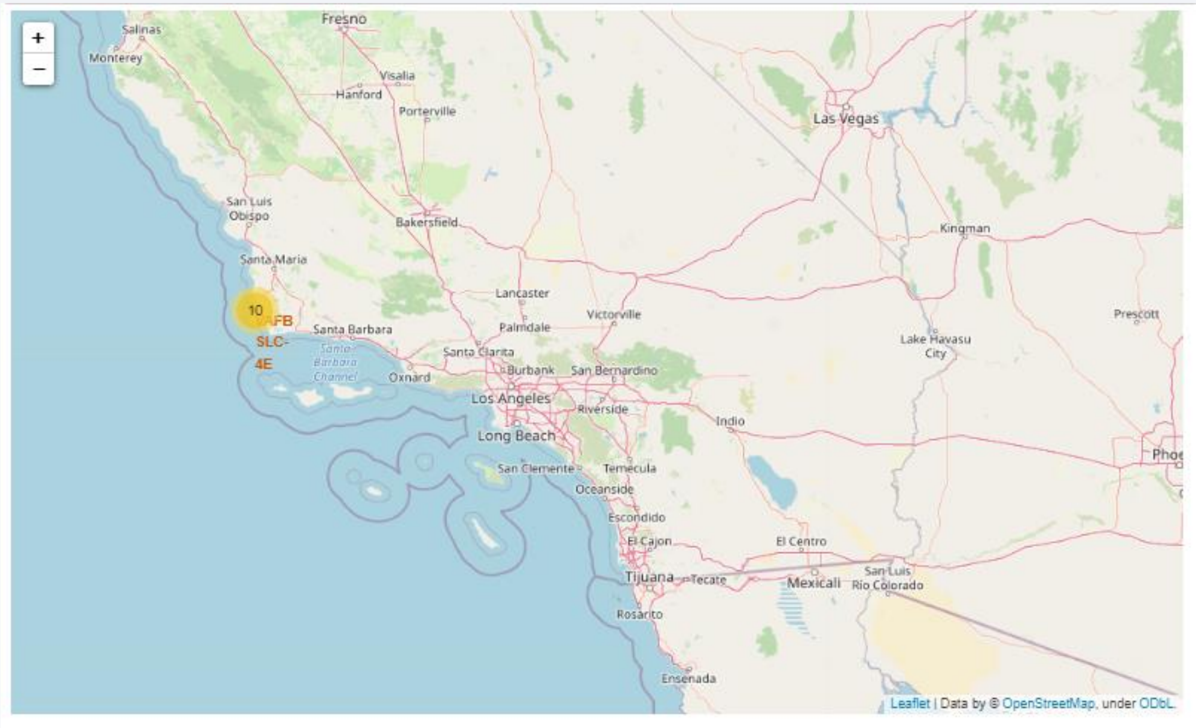
Section 3

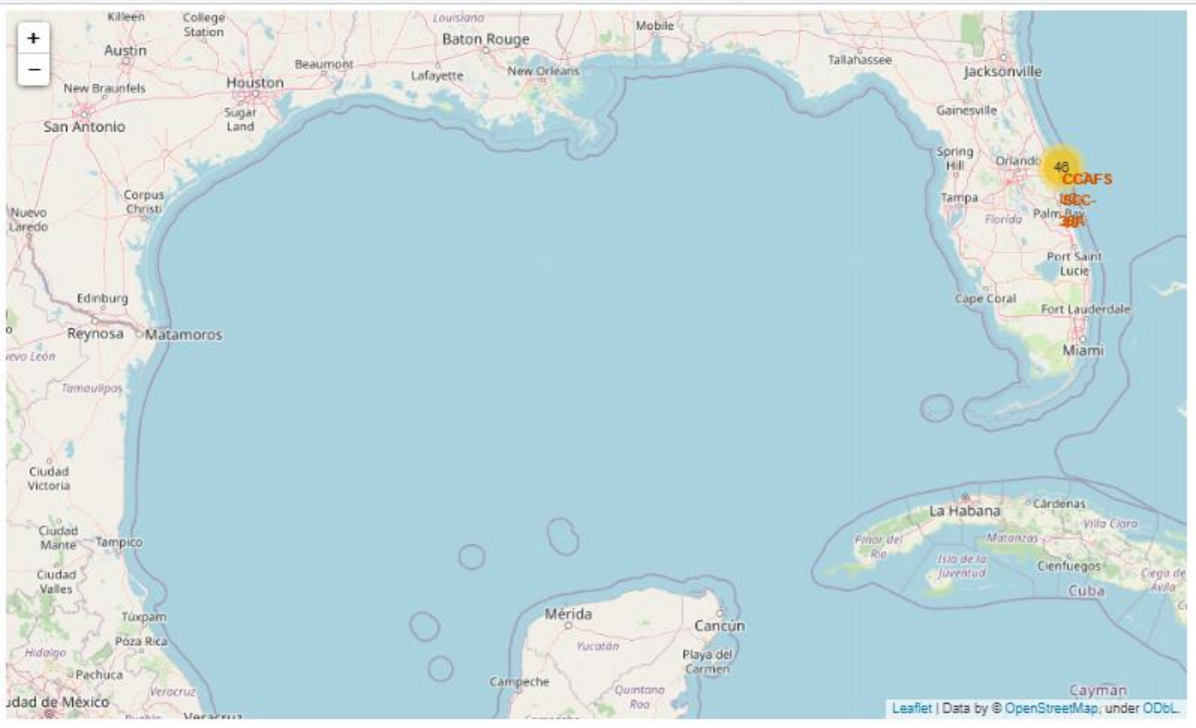
Launch Sites Proximities Analysis

Launch Sites Geographical analysis

- All SpaceX launch sites are situated on the northern hemisphere at the United States of America on its west coast in California and on its east coast in Florida. There is a higher success rate for the launches in Florida Cape Canaveral sites.
- Which also shows that the closer to the Equator the better the result of the launch.
- And the proximity analysis shows that the launch sites rely on infrastructure such as railroads and highways to operate and that for safety reasons they are far from overpopulated areas like cities

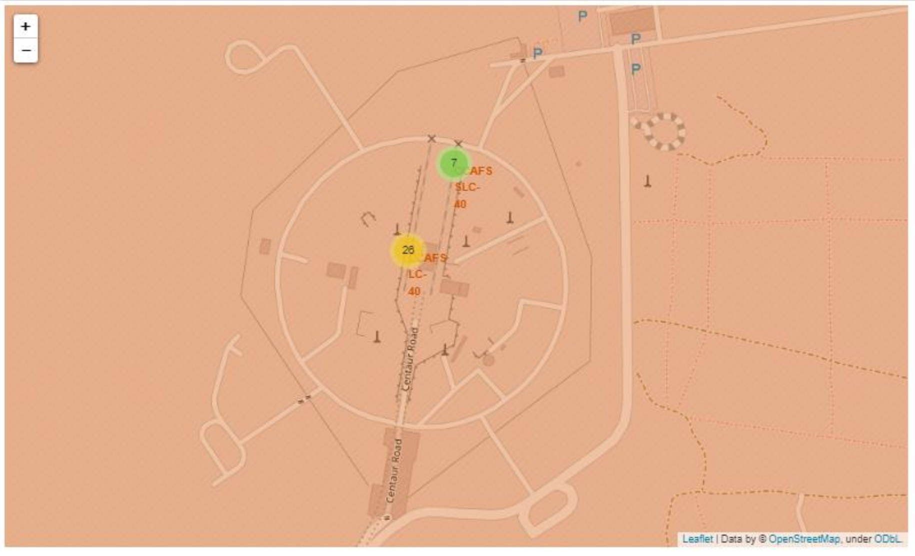
Launch Sites Geographical analysis





Launch Sites Geographical analysis

- For some reason the icons for the colors did not render in the map, I believe it might be related to the zoom scale or the background tiles ; but the date frame was updated



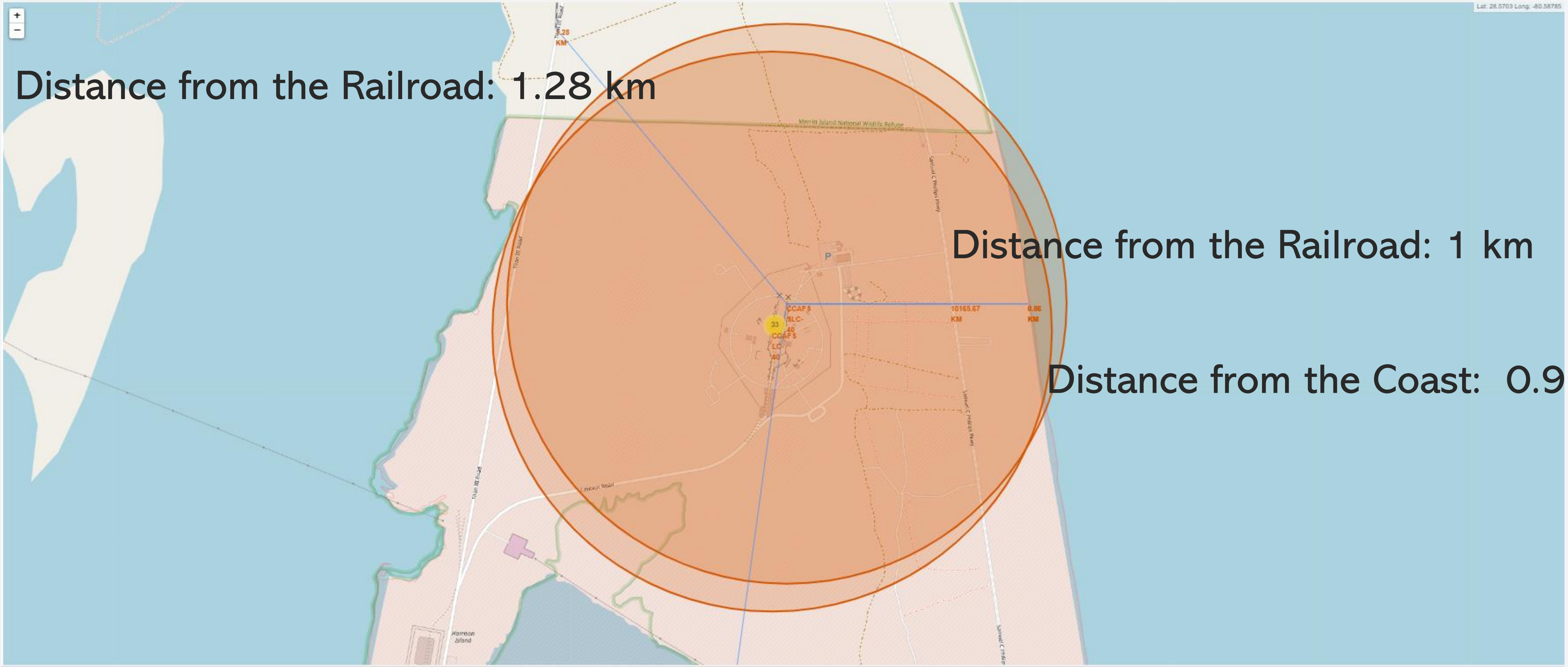
In [105]:
```python
# Function to assign color to launch outcome
def assign_marker_color(launch_outcome):
    if launch_outcome == 1:
        return 'green'
    else:
        return 'red'

spacex_df['marker_color'] = spacex_df['class'].apply(assign_marker_color)
spacex_df.tail(10)
```

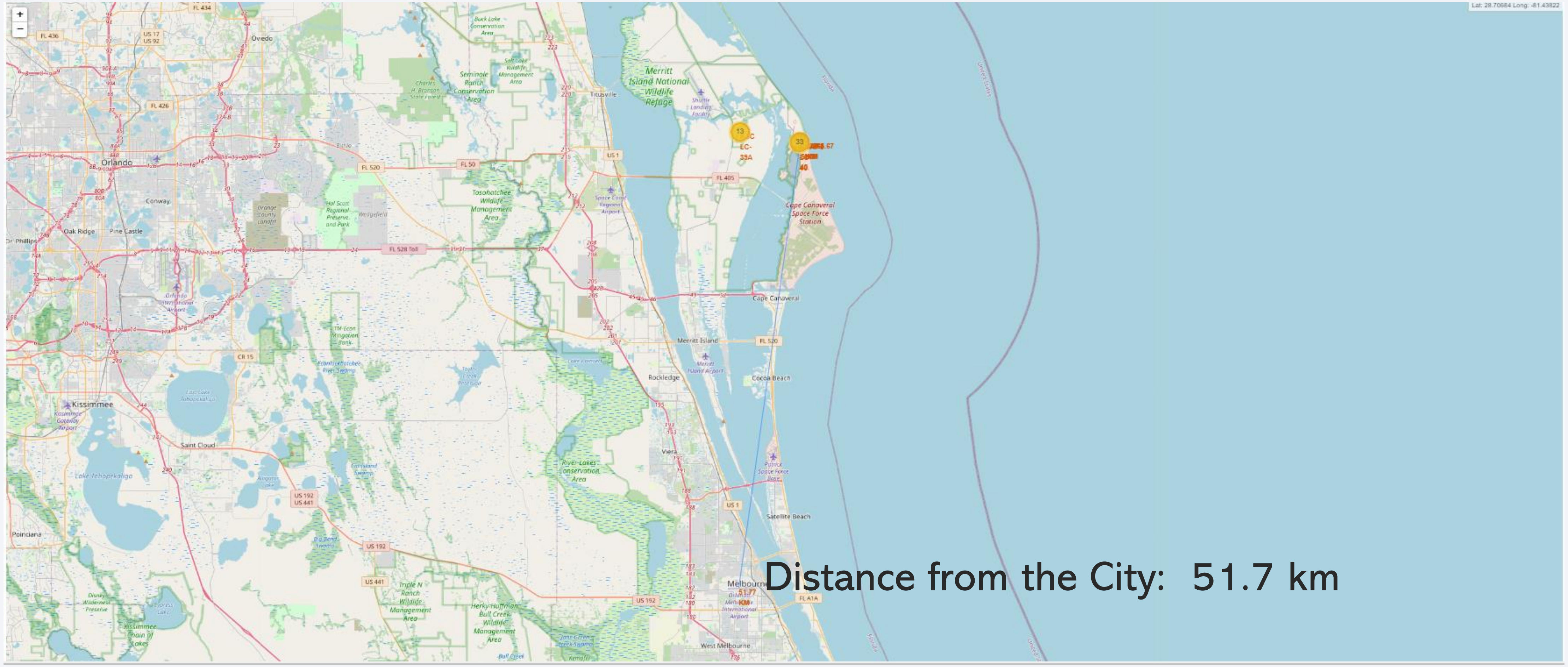
Out[105]:

	Launch Site	Lat	Long	class	marker_color
46	KSC LC-39A	28.573255	-80.646895	1	green
47	KSC LC-39A	28.573255	-80.646895	1	green
48	KSC LC-39A	28.573255	-80.646895	1	green
49	CCAFS SLC-40	28.563197	-80.576820	1	green
50	CCAFS SLC-40	28.563197	-80.576820	1	green
51	CCAFS SLC-40	28.563197	-80.576820	0	red
52	CCAFS SLC-40	28.563197	-80.576820	0	red
53	CCAFS SLC-40	28.563197	-80.576820	0	red
54	CCAFS SLC-40	28.563197	-80.576820	1	green
55	CCAFS SLC-40	28.563197	-80.576820	0	red

Launch Sites Geographical analysis



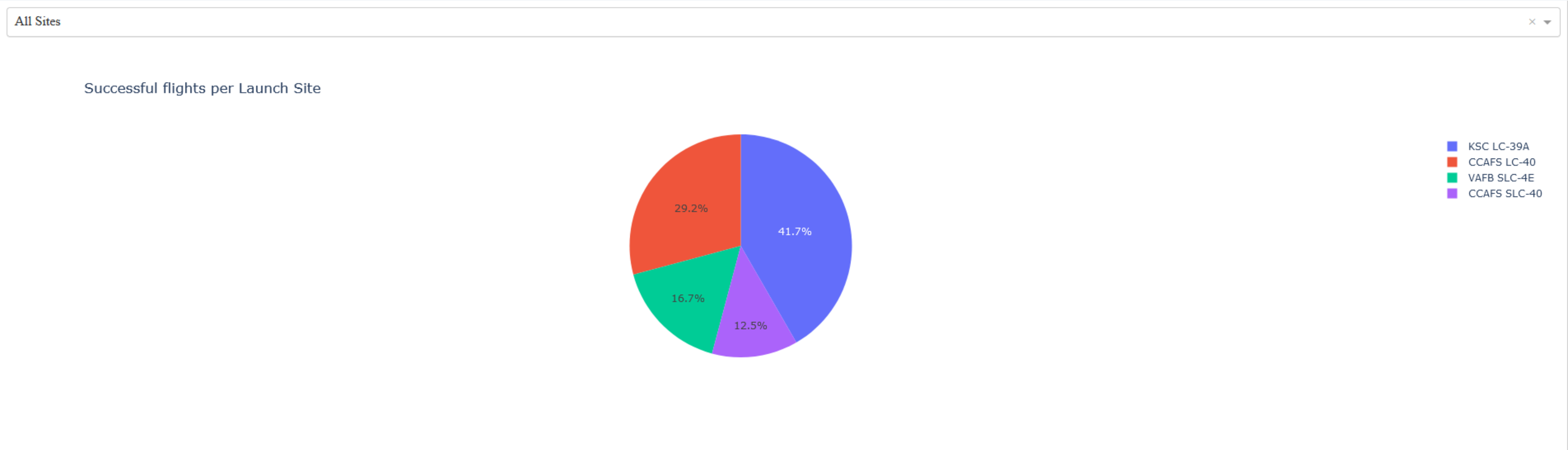
Launch Sites Geographical analysis



Section 4

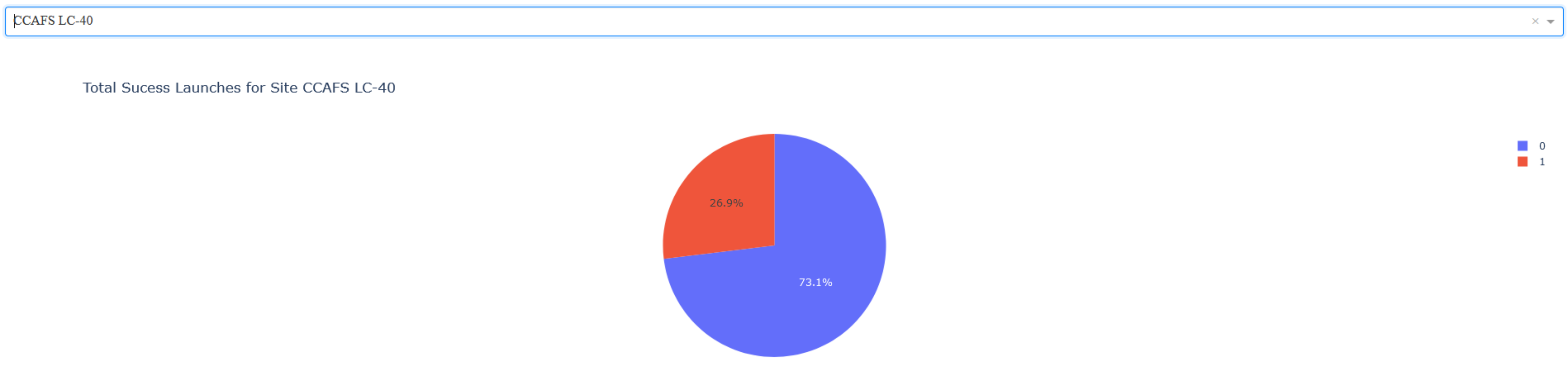
Build a Dashboard with Plotly Dash

SpaceX Launch Records Dashboard



- Dropdown component selection = ALL (label: “All sites”), used as input on callback app pie chart showing the Successful flights per Launch Site where KSC LC-39A has the best success rate. The pie chart allows us to quickly visualize the comparison of the success rate between launch sites
- Which site has the largest successful launches? KSC LC-39A with 41.7%

SpaceX Launch Records Dashboard

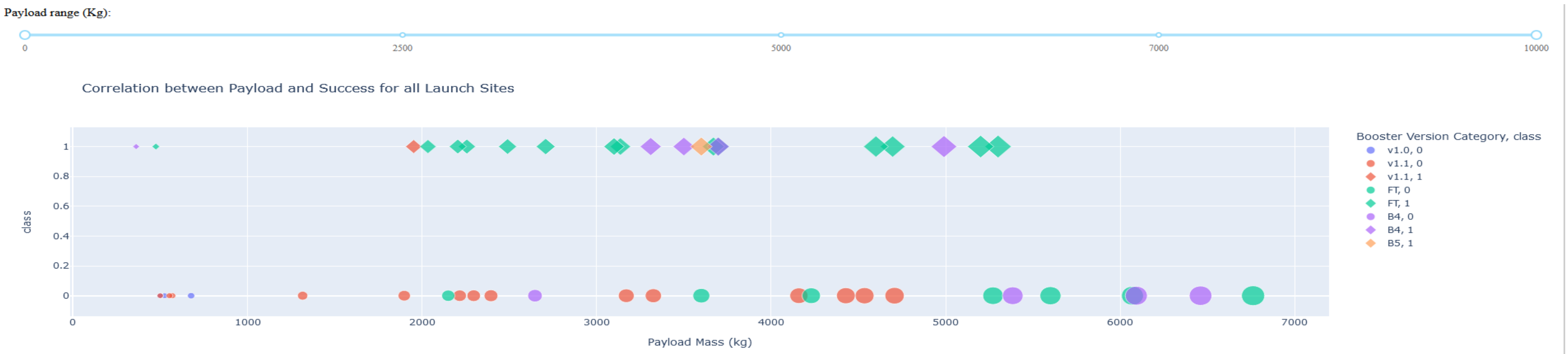


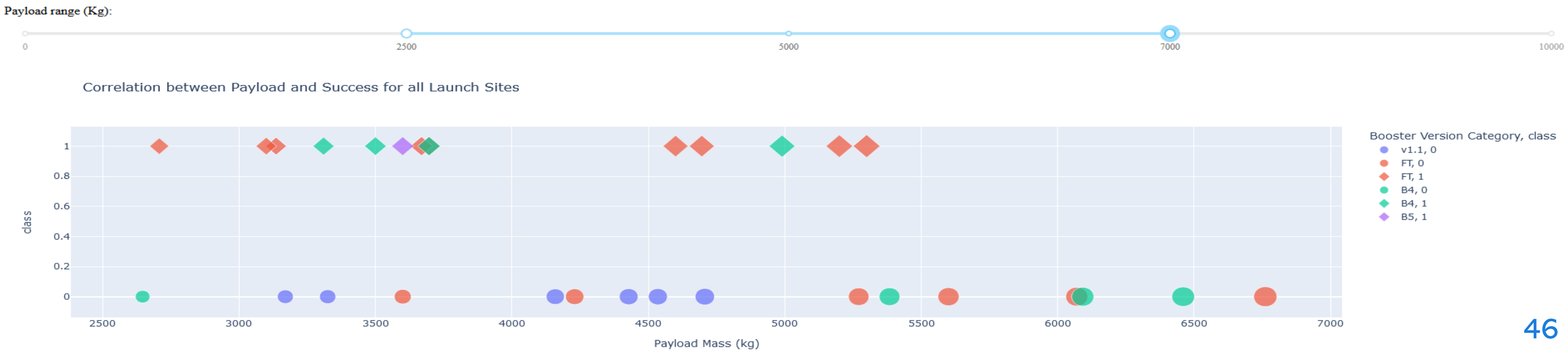
- Dropdown component selection = CCAFS LC-40, used as input on callback app pie chart showing the Successful flights for Launch Site. Where we can see a 73.1% fail rate and 26.9% success. The pie chart allows us to quickly visualize the the success ratio on a launch sites
- Which site has the highest launch success rate? CCAFS SLC-40 with 42.9 %

SpaceX Launch Records Dashboard

- On the next slide we have two pictures of the scatter graph generated by the selection on a horizontal slider component showing the full Payload. The first range from 0 to 10000 and the second from 2500 to 7000, on a "All sites" selection from the dropdown component.
- Range and selection were used as input on callback app showing the Correlation between Payload Mass (kg) and Success rate for all Launch Sites, on the scatter graph. Where we can identify:
  - Which payload range(s) has the highest launch success rate? 2500 - 5500
  - Which payload range(s) has the lowest launch success rate? 2000 - 3000
  - Which F9 Booster version (v1.0, v1.1, FT, B4, B5, etc.) has the highest launch success rate? FT

SpaceX Launch Records Dashboard





Section 5

Predictive Analysis (Classification)

Classification Accuracy

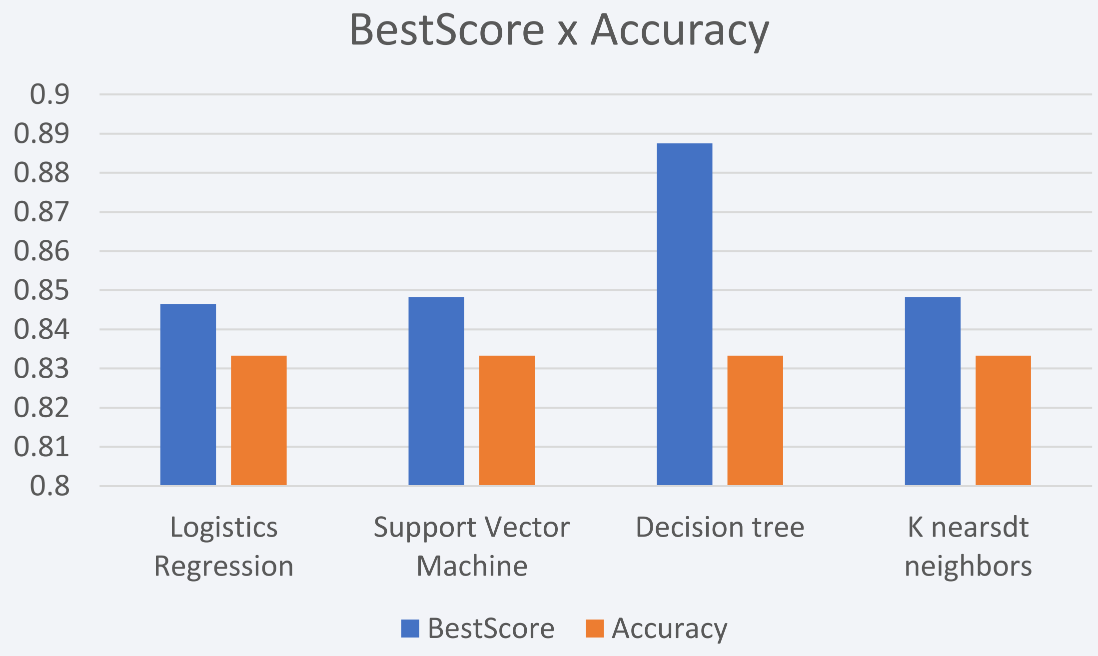
- The model with best score is Decision Tree, with 0.8875
- The accuracy for all models was: 0.8333

```
In [34]: print('Accuracy for Logistics Regression method:', logreg_cv.score(X_test, Y_test))
         print( 'Accuracy for Support Vector Machine method:', svm_cv.score(X_test, Y_test))
         print('Accuracy for Decision tree method:', tree_cv.score(X_test, Y_test))
         print('Accuracy for K nearsdt neighbors method:', knn_cv.score(X_test, Y_test))

    Accuracy for Logistics Regression method: 0.8333333333333334
    Accuracy for Support Vector Machine method: 0.8333333333333334
    Accuracy for Decision tree method: 0.8333333333333334
    Accuracy for K nearsdt neighbors method: 0.8333333333333334

In [37]: print('Best score for Logistics Regression method:', logreg_cv.best_score_)
         print( 'Best score for Support Vector Machine method:', svm_cv.best_score_)
         print('Best score for Decision tree method:', tree_cv.best_score_)
         print('Best score for K nearsdt neighbors method:', knn_cv.best_score_)

    Best score for Logistics Regression method: 0.8464285714285713
    Best score for Support Vector Machine method: 0.8482142857142856
    Best score for Decision tree method: 0.8875
    Best score for K nearsdt neighbors method: 0.8482142857142858
```



Confusion Matrix

- As Decision Tree was the best performing model its Confusion Matrix shows two classes very distinguished. But we also see that the major problem is false positives, where failed landings are defined as successful by the classifier.

Conclusions

- The greater the number of launches on a site, greater is also its success rate.
- From 2013 onwards there was a constantly increase on the success rate for SpaceX rocket launches with a slight decline in 2018 which was soon recovered in 2019
- Orbit PO has higher success rate, followed by ES-L1 then HEO and MEO to SSO then GEO.
- Largest successful launches were found at KSC LC-39A site (41.7%)
- The highest launch success rate was found at CCAFS SLC-40 site (42.9 %)
- The payload range(s) with the highest launch success rate is 2500 - 5500
- The payload range(s) with the lowest launch success rate 2000 - 3000
- The F9 Booster version FT has the highest launch success rate
- The model with best score is Decision Tree, with 0.8875
- The accuracy for all models was: 0.8333
- The average payload mass carried by booster version F9 v1.1 was 2928 kg

Appendix

- GitHub link to all datasets used and notebooks created:

  https://github.com/p00d33m/Coursera-IBM-DS-Capstone-2022

Thank you!