IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Alejandra Ramos
24-01-2022

Outline

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

Executive Summary

- In terms of methodology, data collection was performed, the tables of information were processed and "cleaned". One proceeded to graph different relationships between features. Interactive maps and graphs were also made to be able to vary and compare more easily each feature of the launches and finally a prediction of the outcome of the launches was made with different classification algorithms.

- In terms of results, Space Y first endeavor should focus on launches using a booster with similar specs as the F9 FT Booster, from a site with resources and conditions like the KSC LC39A and centered in orbits like LEO to increase the potential profit.

Introduction

Over the years private companies have taken an interest in making space travel accessible to all, SpaceX has been one of the companies with more recognition in this subject, since its foundation in 2002, having achieved great success with their use of a "two stage rocket", which as the name imply are space vehicles that use two separate stages that provide propulsion consecutively to achieve orbital velocity, Space X can reduce the cost of each rocket from 165 million (from other suppliers) to 62 million thanks to this recovery of the first stage, which is the most expensive stage. For this project we will using data science to work for a new rocket company: Space Y founded by Allon Musk, we gather the information about Space X and determine if the first stage will land successfully and the characteristics for it to see how one can replicate the achievement from Space X.

Section 1

Methodology

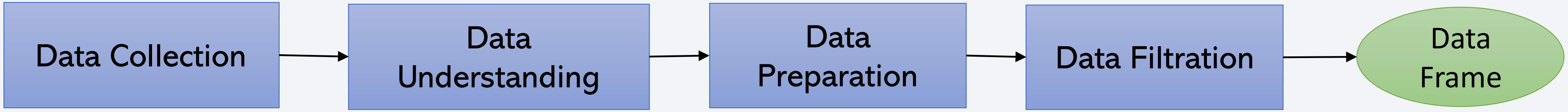
Methodology

Executive Summary

- Data collection methodology:
  - The data resources are identified and collected, the information is saved in variables, which are then reviewed and cleaned up. To finally be stored in data frames.
- Perform data wrangling
  - NaN values are detected in the table, which will then be replaced (e.g. with the average), values with different formats will be overwritten and unnecessary values will be deleted.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Create the variable with the features (X) and the vector with the variable to be predicted (Y), apply the function “train_test_split", apply the algorithm of each classification method (with the set "train" and finally proceed to calculate the efficiency of the method with the set "test".

Data Collection

- The data resources are identified and gathered from the SPACEX API and the SPACEX Wiki page. For both cases we uses the function .get() to extract the content of the link to a variable, that later we will “clean”: remove the duplicates, review the data format, delete/replace the missing data and the invalid values. After that, we will filter the information that we require, in this case only the data from the Falcon 9, and finally, all the relevant information will be save in a data frame.

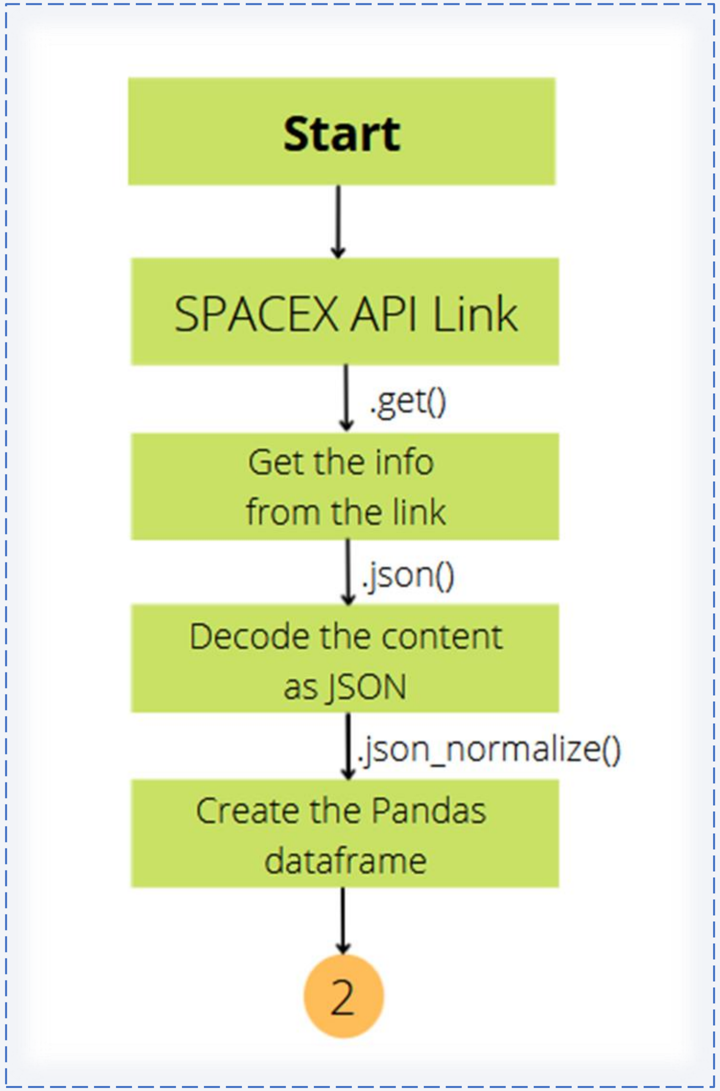


Data Collection – SpaceX API

Link to Github: SpaceX API

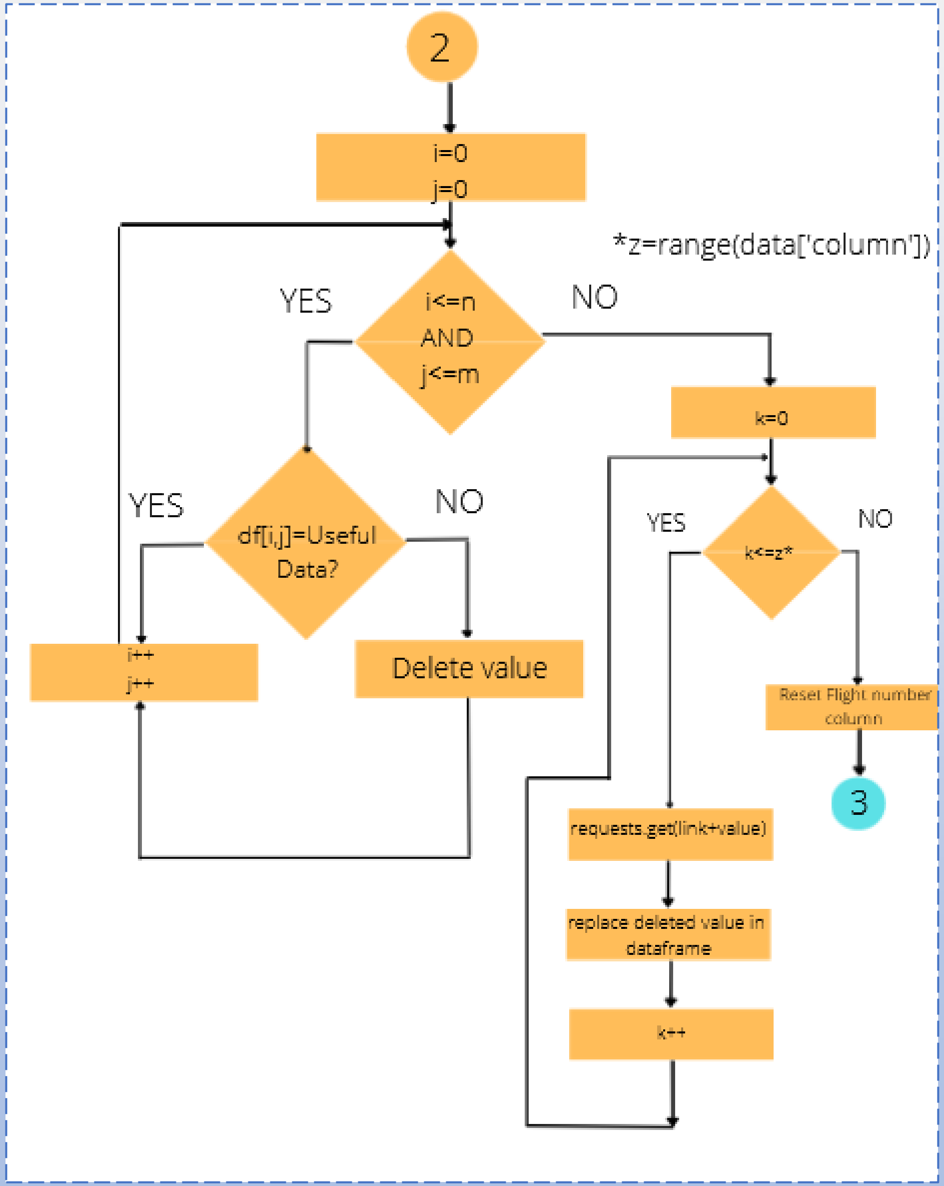
The data collection with SpaceX REST could be separated in four (4) differences phases, in the following slides show the separate flowcharts for each of the phases

- Phase 1 describes how the **data is collected from the source used** (with the GET() function) and **how the information is transformed** into a format/object useful for processing (JSON).



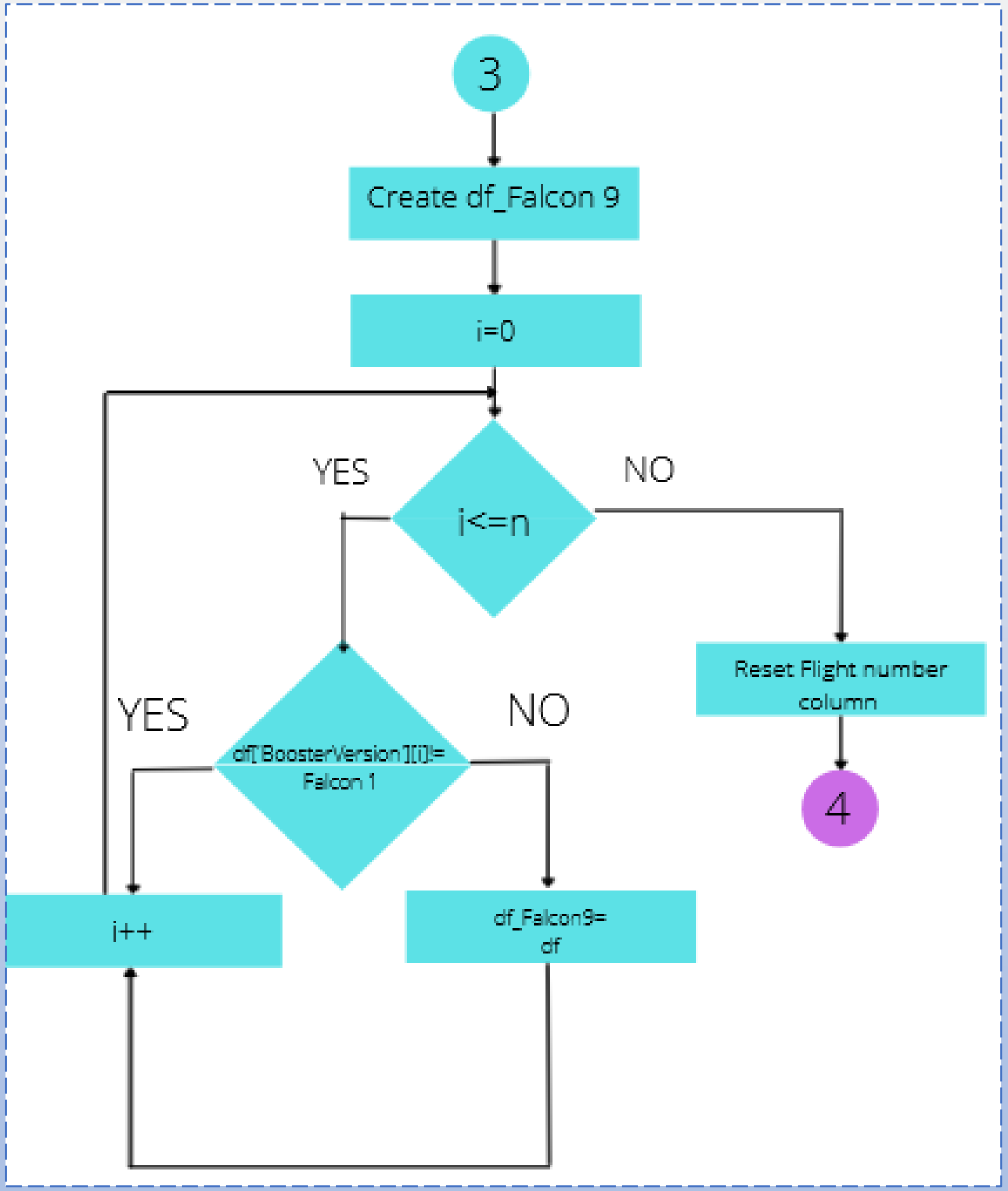
Data Collection – SpaceX API

- Phase 2 describes how the data obtained are **identified and cleaned**, deleting useless data for the study, in this case ID values that only work for SPACE X internal usage; because they do not reveal any information relevant for external observers.
- These deleted values will be **replaced by values extracted** directly from the columns in the tables found in the SPACE X API link with the use once again of the **GET() function**, the name of the column and the value to extract.



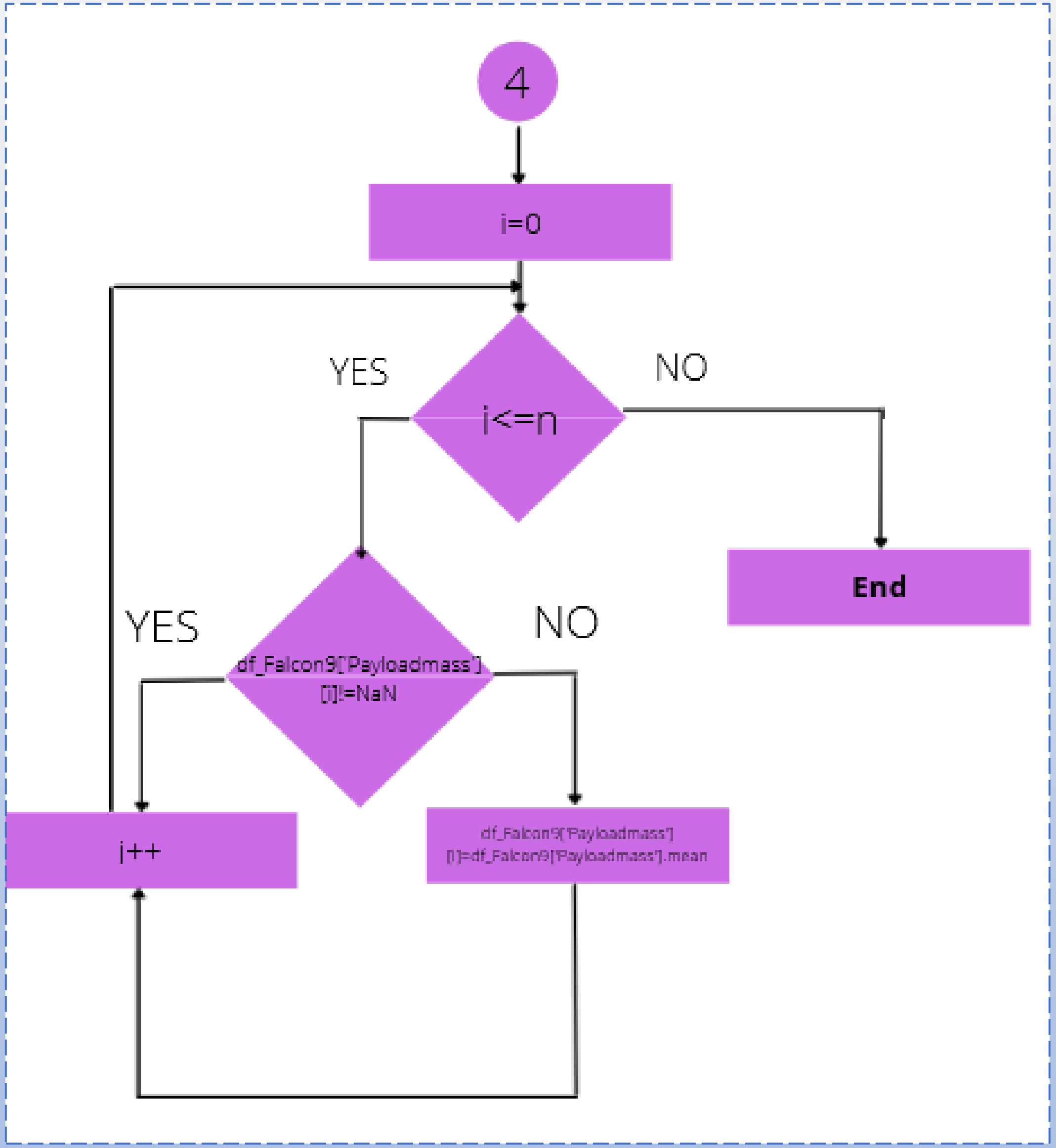
Data Collection – SpaceX API

- Phase 3 describes how the previously cleaned data is **filtered** to leave in the data frame only that which will be useful for the study, in this case values that correspond to the **Falcon 9** version of the boosters.



Data Collection – SpaceX API

- Finally, phase 4 describes how to **prepare** the data by identifying the unknown values in the data frame **(NaN)**, specifically from the booster payload column, and replacing them with the **average** value of the entire value.

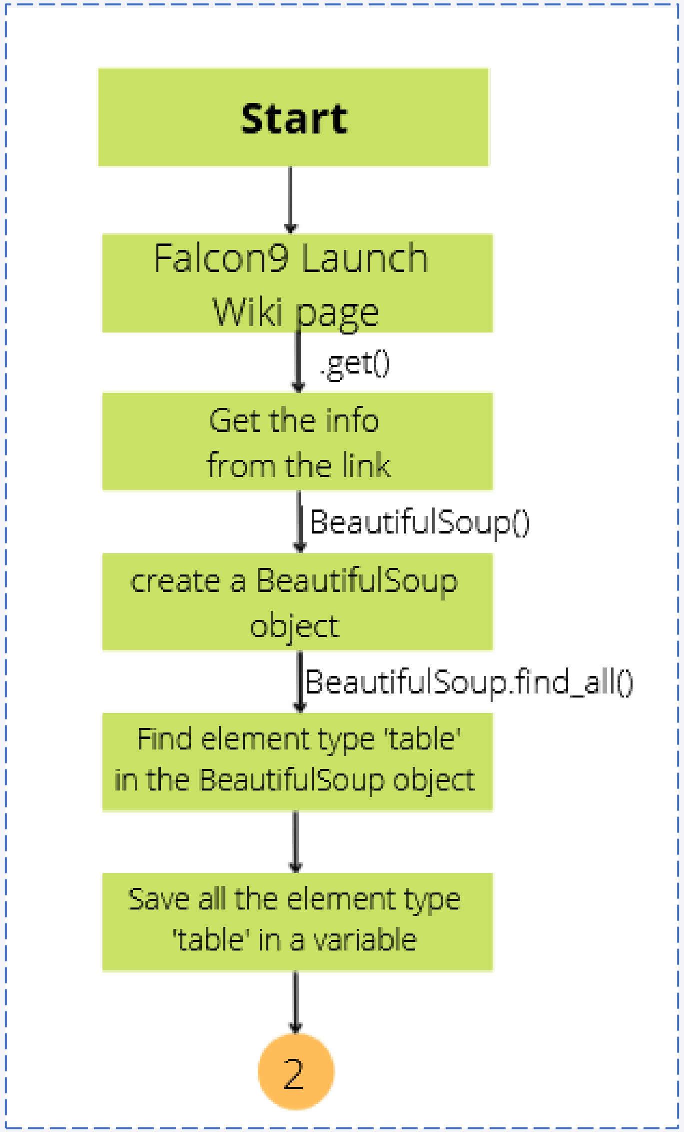


Data Collection - Scraping

Link to Github: [Scraping]

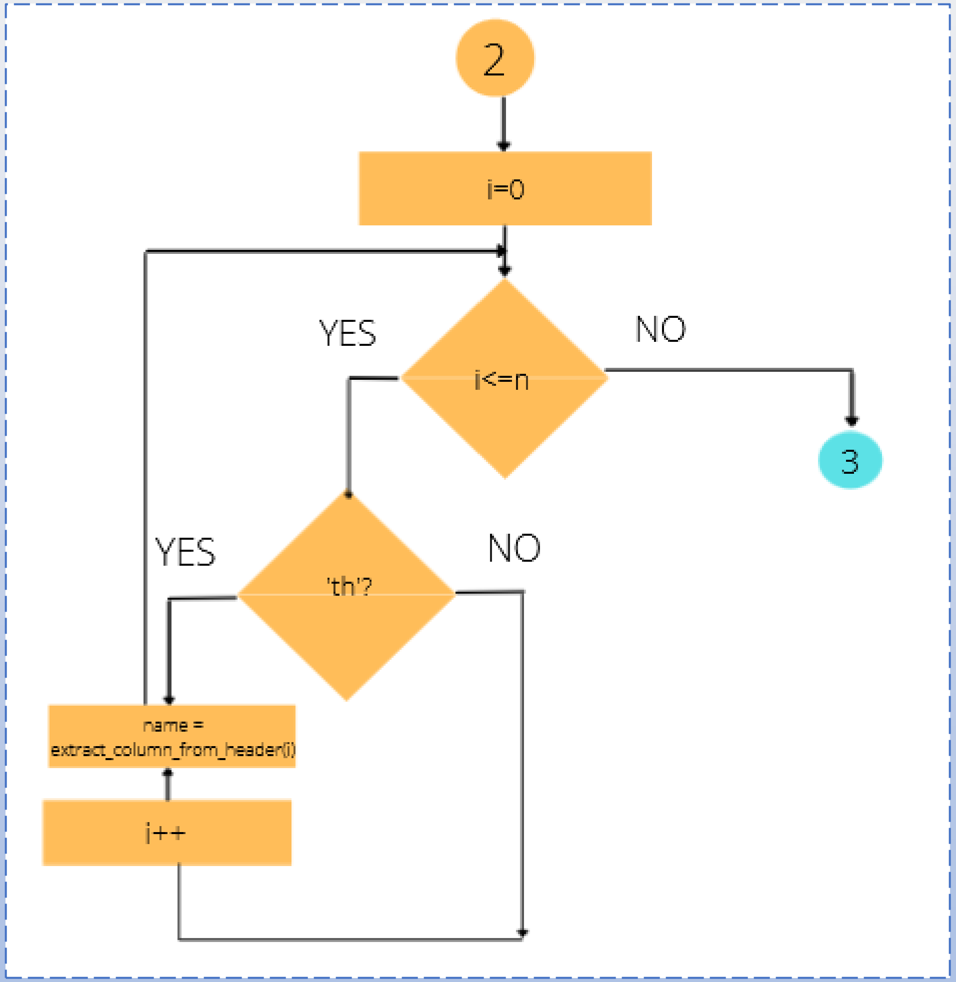
The data collection with web scraping could be separated in four (4) differences phases, in the following slides show the separate flowcharts for each of the phases.

- Phase 1 describes how the **data is collected from the source used** (with the GET() function) and **how the information is transformed** into a format/object useful for processing (BeautifulSoup).



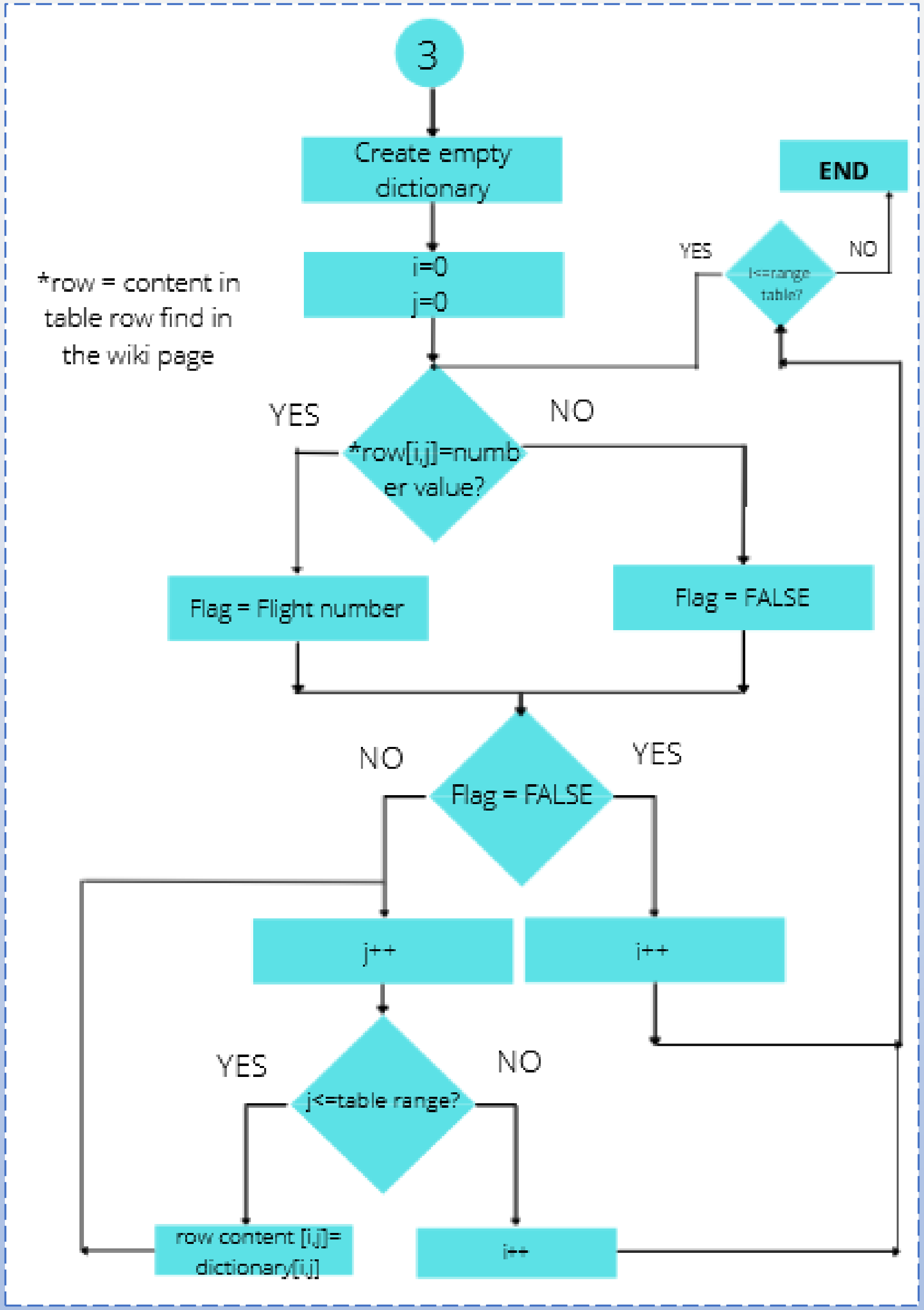
Data Collection - Scraping

- Phase 2 describes how the column names (to be used in the data frame as well) are **obtained from the table headers** found on the web page. The **<th>** tag is used to recognize the header elements. Then they are stored in a list.



Data Collection - Scraping

- Phase 3 describes how the content of the tables is **extracted to fill the data frame**; the first step is to determine that the first value of the row is indeed the number of a flight (with the first conditional). After that, the data is stored in the first cell of the **data frame and then the table is scrolled horizontally** (i.e., the same row but the next column) to obtain the next value of that flight (e.g., the launch date, then the Booster number and so on).
- This process continues until all the data of the row has been saved and then moves to the next row, and so on until all the values of the table are complete.



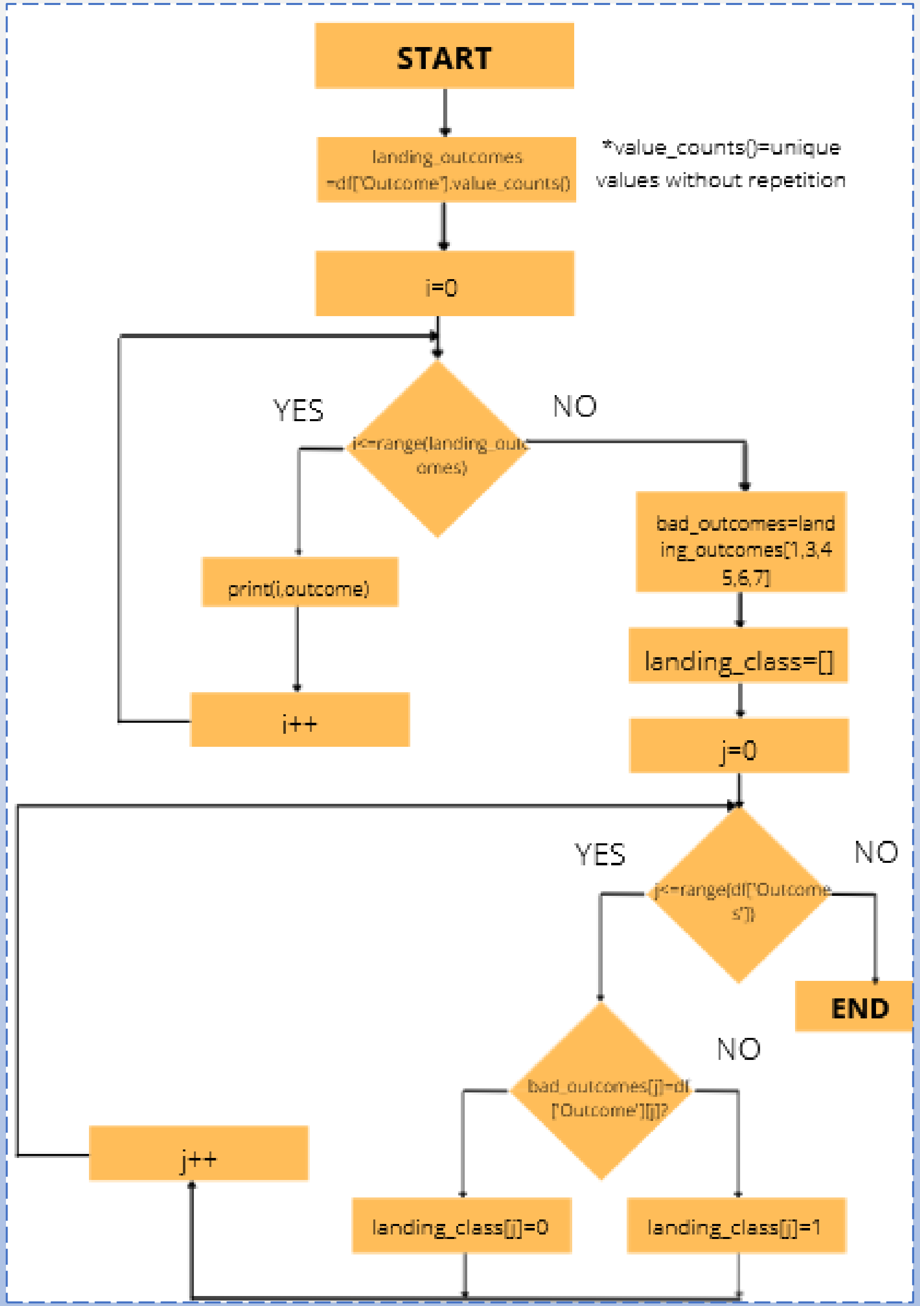
Data Wrangling

- Data Wrangling is the process of "cleaning" and transforming values, so that they can be processed and joined to other databases, for example, using the same nomenclature or blending data for easier reading. In both above data collection processes, we performed data wrangling too.
- In this laboratory ONLY the data processing was performed on the results of the landings (TRUE OCEAN, FALSE OCEAN, TRUE RTLS and so on) were transformed into binary results (1 for all successful landings and 0 for all failed landings).

Link to Github: Wrangling

Data Wrangling

- In the present flowchart, 3 loops are highlighted, the first one is to obtain all the results (unique without repetitions) of the landings and number them, thus creating a variable (called bad_outcomes) with only the failed outcomes. This variable is then compared with the original database and when the value of the table corresponds to any of the content of "bad_outcomes", the result is 0, otherwise it is 1, both cases will be stored in a new variable called "Landing_class".



EDA with Data Visualization

Link to Github: EDA with Data Visualization

- The charts that were plotted:
  - Flight Number vs Launch Site: to determine the success rate of each Launch Site.
  - Payload vs Launch Site: to determine which payload was used in each Launch Site.
  - Success Rate vs Orbit type: to determine the success rate of each Orbit type.
  - Flight Number vs Orbit type: to observe the evolution of the change (in time) of the Orbit type used.
  - Payload vs Orbit type: to determine which payload was used for each Orbit type.
  - Orbit type vs Launch Site: to determine what size booster can be launched at each launch site (with the orbit type)
  - Launch Success yearly trend: to determine the year(s) when the SpaceX began to be successful in order to study more specifically what changed.

EDA with SQL

Link to Github: [EDA with SQL]

- SQL queries used:
  - Queries to collect and choice the data we want it and from desired table as: SELECT and FROM
  - Conditional queries to restrict the needed data as: WHERE, LIMIT and LIKE.
  - Queries to perfom unions or intersections of data: GROUP BY and UNIQUE()
  - Queries to conduct mathematical operations just like: SUM(), AVG(), MIN() and MAX()
  - Queries to perform logical operations like: AND.

Build an Interactive Map with Folium

- Map objects created and added to a folium map:
  - Circles: mostly used to mark and evaluated the position of the launch sites in the map.
  - Markers with color code: to visualized easily successful and failed outcomes (on each launch site).
  - Lines and text: to pictured and compared effectively the distances between a specific launch site and other important places just as highways, cities, line coast and railroad.

Link to Github: Interactive Map with Folium

Build a Dashboard with Plotly Dash

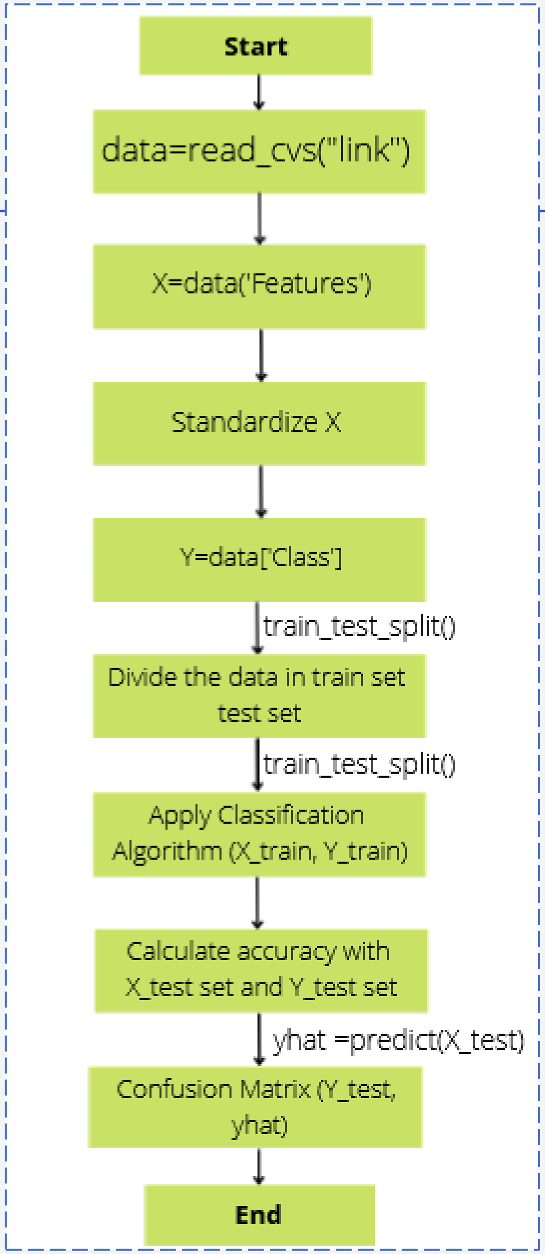
- Plots, graphs and interactions added to a dashboard:
    - Dropdown list with all launch sites: to change easy the launch site to compare the success rate between then.
    - Pie chart with total success launches by sites: to compare graphically all the success rate of each site, to determine faster the most successful place.
    - Slider to select payload range: for easy variation between all the payload range, to compare the success rate between them.
    - Scatter plot with the correlation between payload and success for each launch site: for easy understanding how, the payload affects the successful rate between all the launch sites.

Link to Github: Dashboard with Plotly Dash

Predictive Analysis (Classification)

- The predictive analysis starts after obtaining the table with the data (which is in .cvs format) and saving it in a variable (called data). All the features of each flight will be stored in the variable X and in the variable Y will be stored the vector "Class" which corresponds to the column with the same name of the data frame that contains the outcome of the phase 1 landings of each booster. The function **"train_test_split()"** will then be applied to the mentioned variables. The algorithms for each **classification method** (Logistic Regression, Support Vector Machine, Decision Tree and K Nearest Neighbors) are applied to the **train set**, then the accuracy of respectively method is calculated with the **test set** and finally the confusion matrix is obtained with the “**y_test**” and the “**yhat**” (the “**yhat**” is the y value obtained using the “**x_test**” with the predicted function of each classification algorithm).

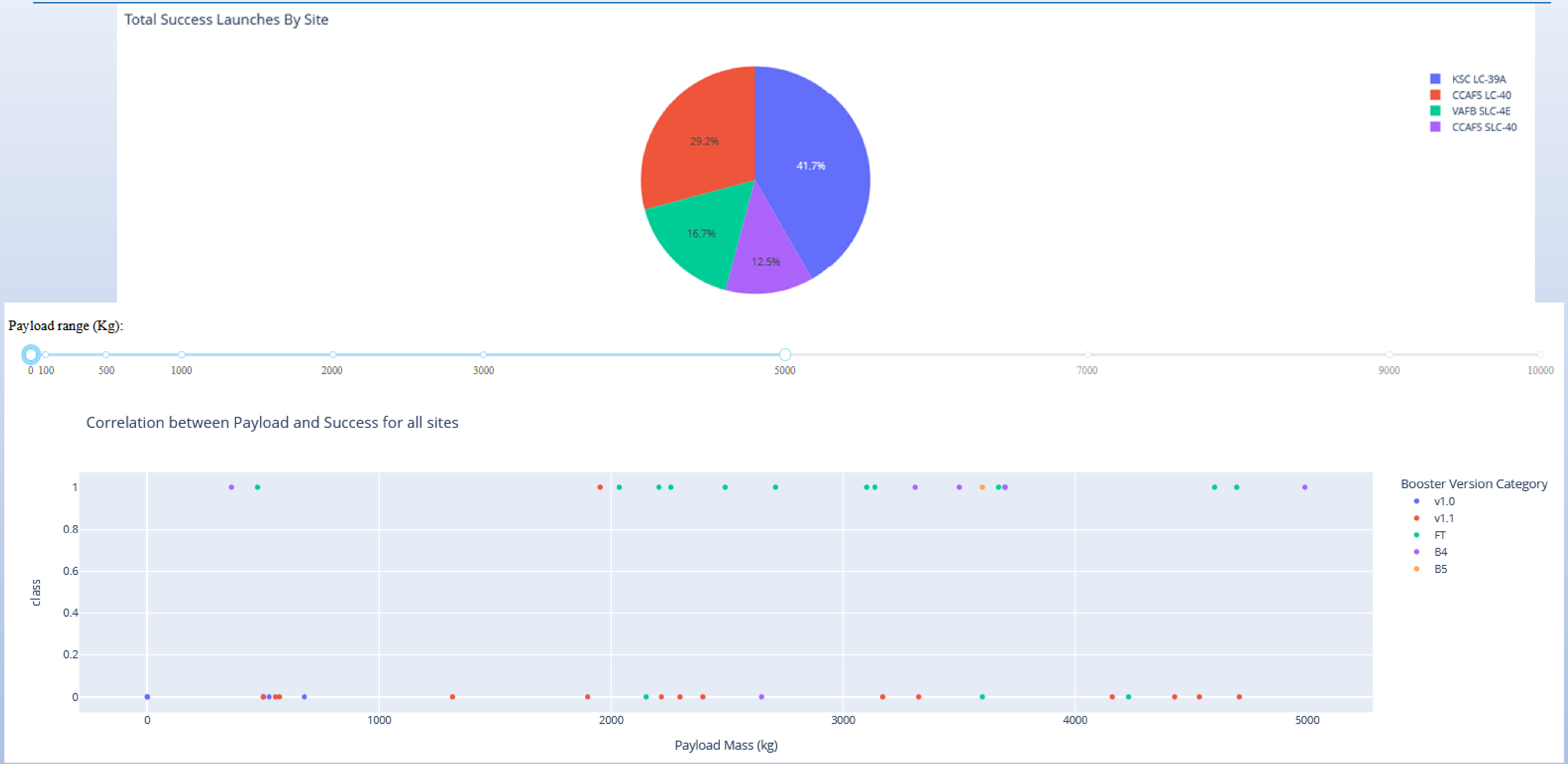
Link to Github: [Predictive Analysis]



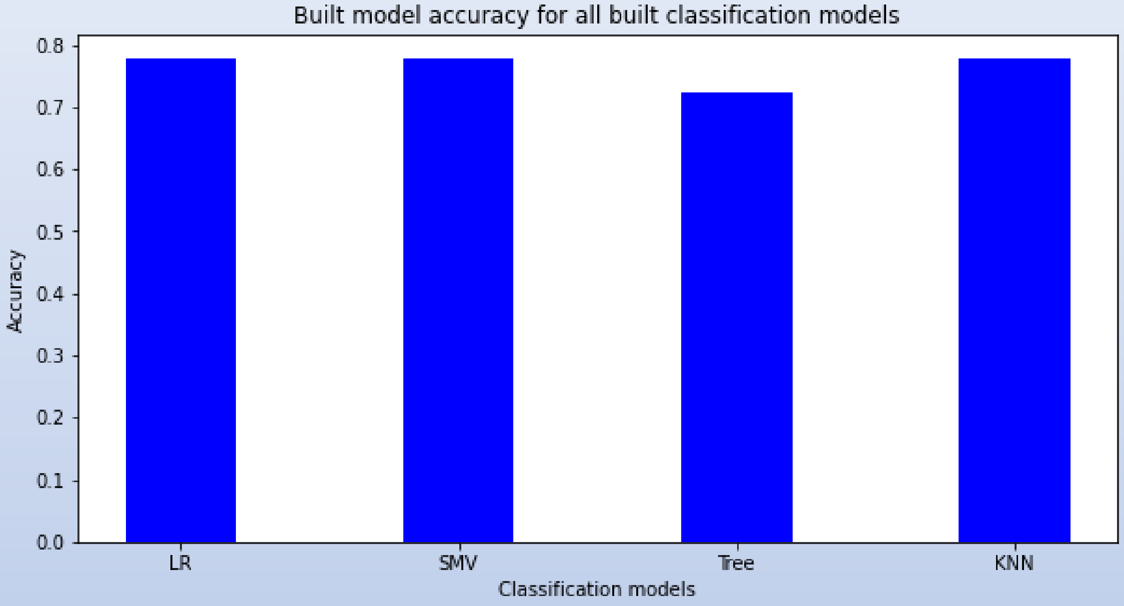
Results (from EDA)

- In the VAFB-SLC launch site there are no rockets launched for heavy payload mass(greater than 10000).
- The highest success rate are for the ES-L1, GEO, HEO, SSO and VLEO orbits.
- We only have the orbit type PO and SSO in the launch site VAFB SLC 4E.
- We only have the orbit type LEO, ISS, GTO, VLEO and SO in the launch site KSC LC 39A.
- The site with the highest success rate is KSC LC-39A.
- The only booster version with the Payload between 5000 and 10000 kg are the FT and the B4
- The B4 booster version is the booster with the highest payload.
- The booster with the highest success rate is the Booster F9 FT (Falcon 9 Full Thrust)

Results (Interactive analytics)



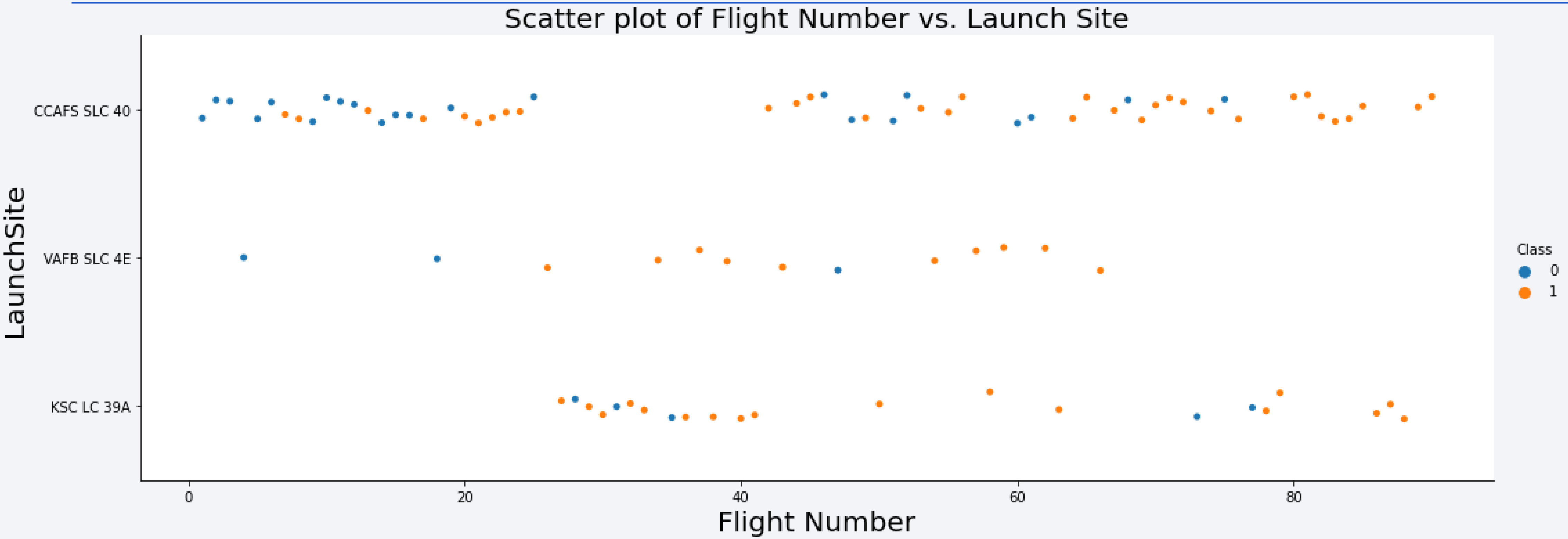
Results (Predictive Analysis)



Section 2

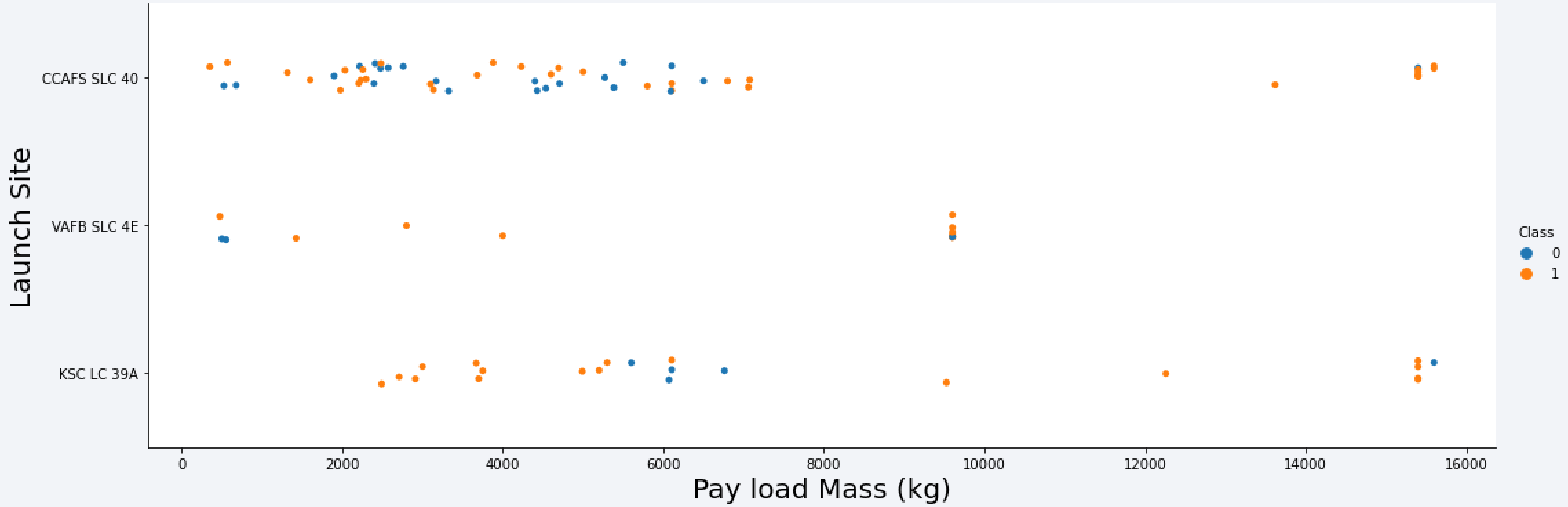
Insights drawn from EDA

Flight Number vs. Launch Site



- The success rate of CCAFS SLC 40 is **60%**
- The success rate of KSC LC-39A is **77%**
- The success rate of VAFB SLC 4E is **77%**

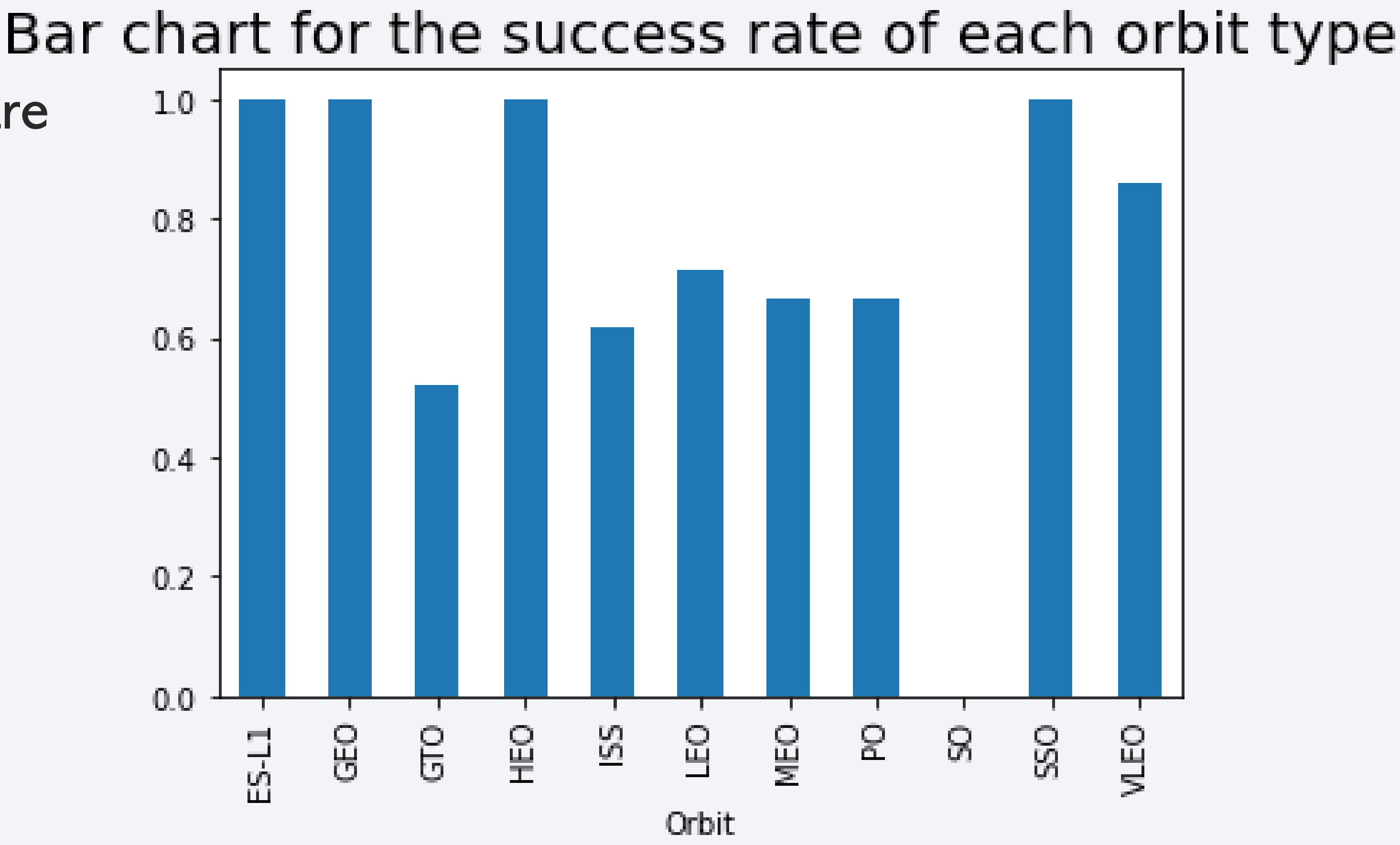
Payload vs. Launch Site



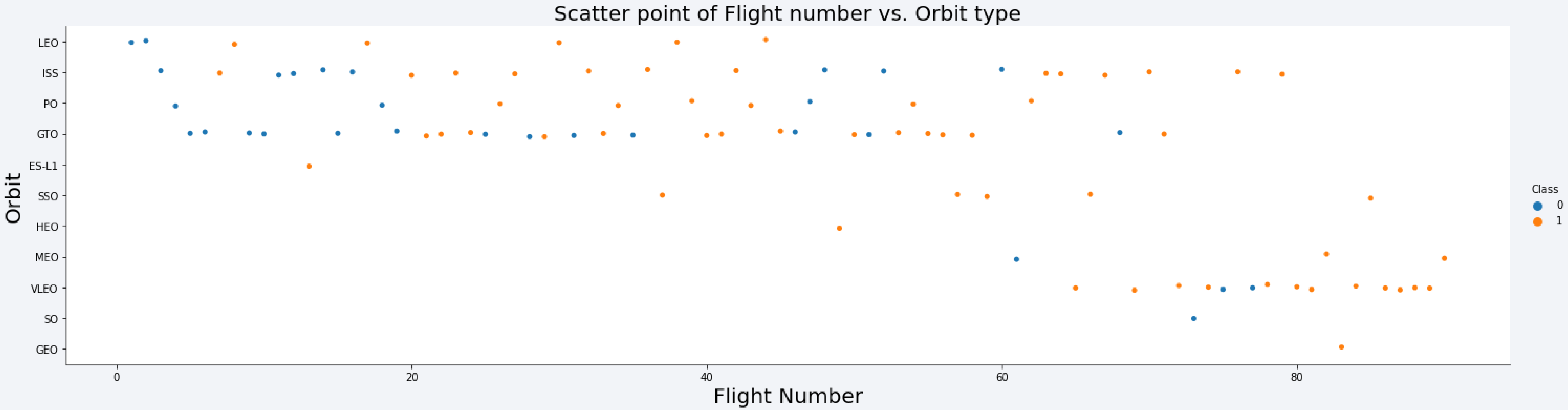
- In the VAFB-SLC launch site there are no rockets launched for heavy payload mass(greater than 10000).

Success Rate vs. Orbit Type

- The highest success rate are for the ES-L1, GEO, HEO, SSO and VLEO orbits.
- The lowest success rate is the GTO orbit.

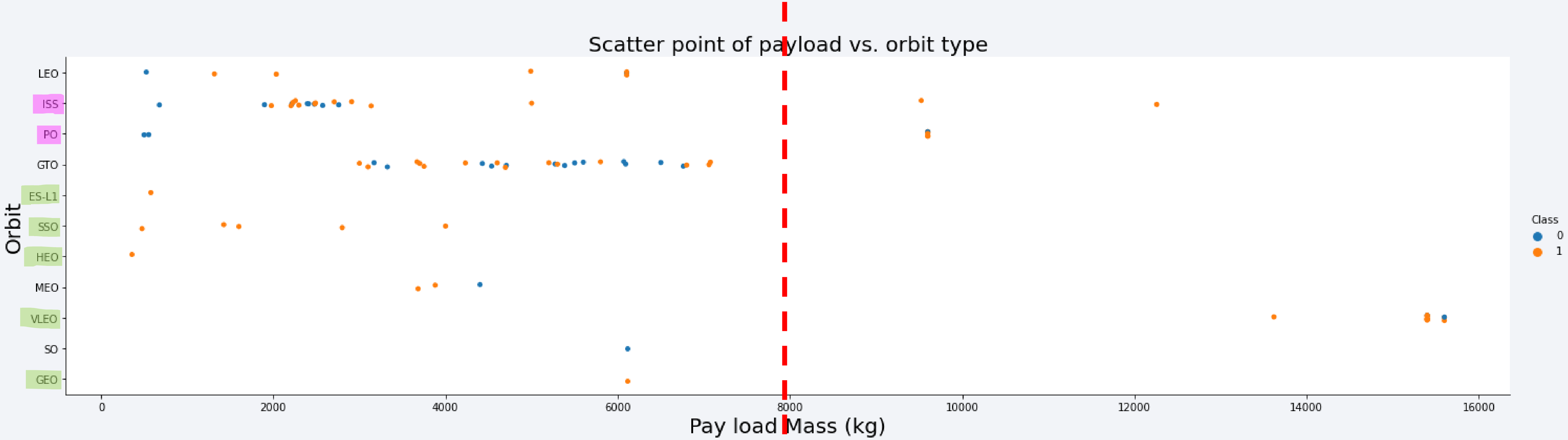


Flight Number vs. Orbit Type



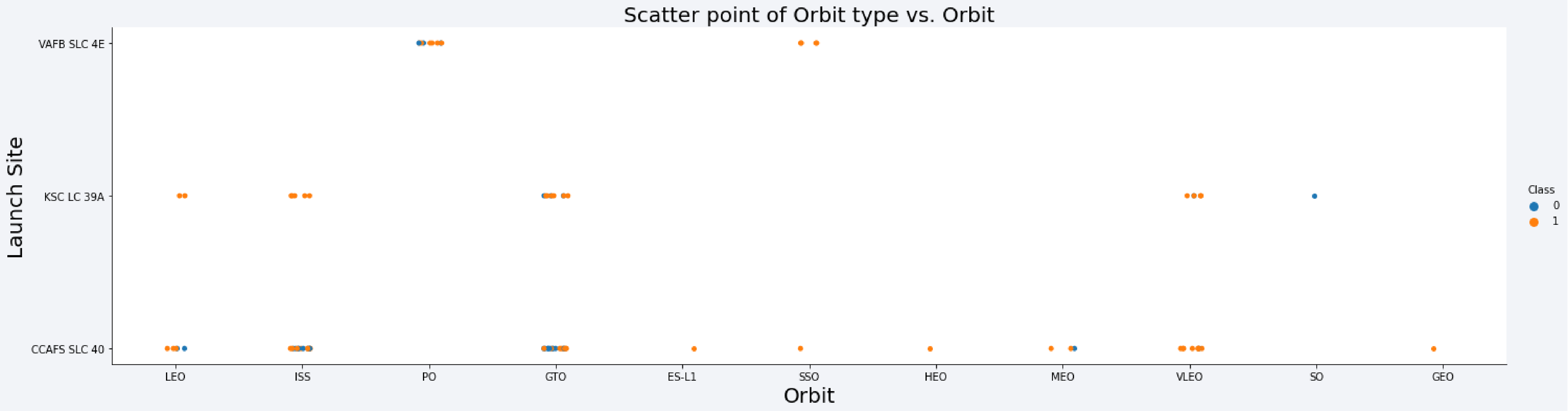
- The first orbit performed was LEO.
- The VLEO orbit started to be used after the 60th flight.
- The last orbit performed is MEO.
- One of the last orbits performed is GEO.

Payload vs. Orbit Type



- The orbits with the highest efficiency (ES-L1, GEO, HEO, SSO and VLEO), have that success rate only with payloads less than 8000 Kg.
- The orbits with the highest efficiency ratio for payloads above 8000 Kg are ISS and PO.
- for GTO we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission) are both there here.

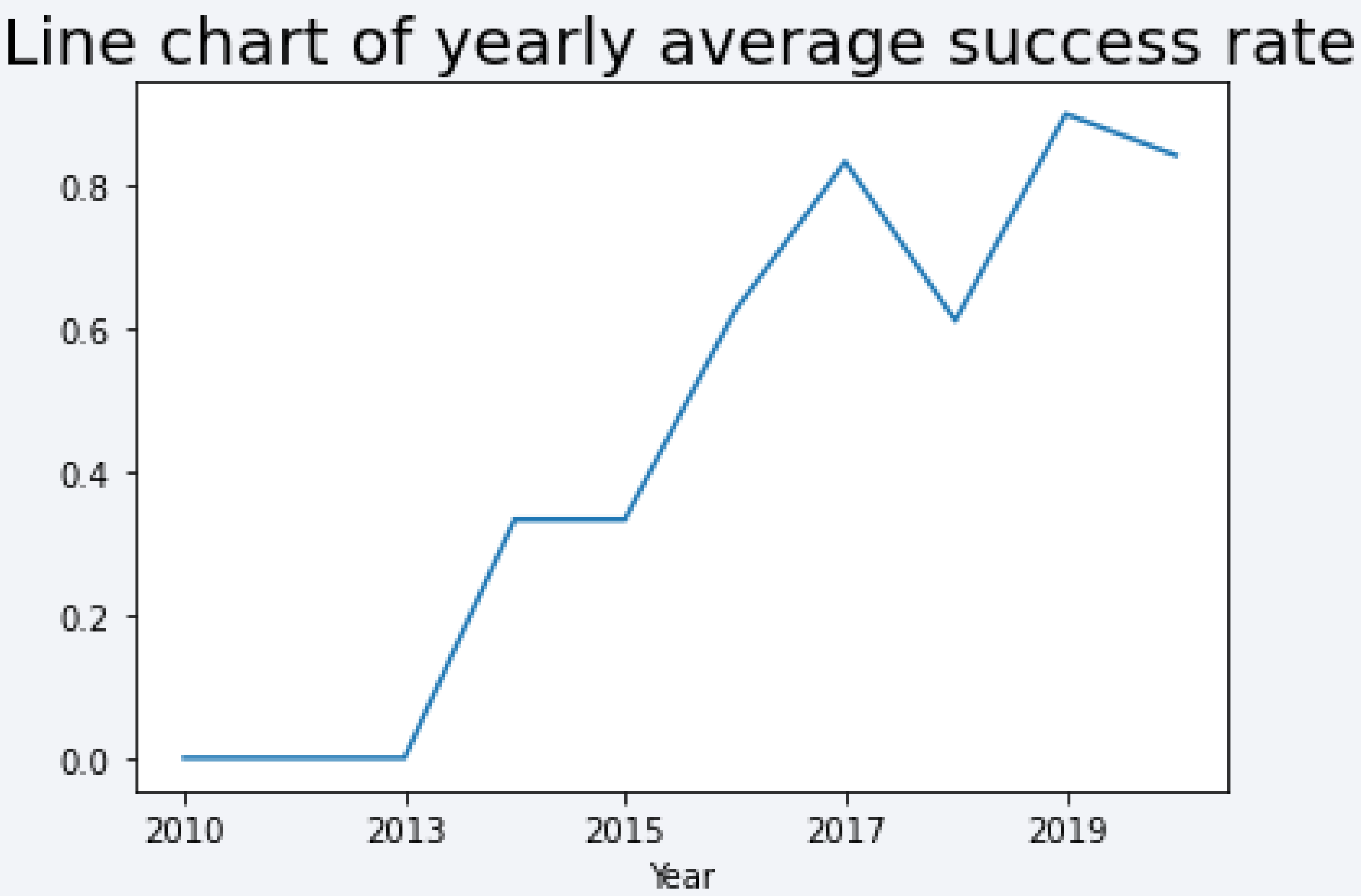
Orbit type vs. Launch Site



- We only have the orbit type PO and SSO in the launch site VAFB SLC 4E.
- We only have the orbit type LEO, ISS, GTO, VLEO and SO in the launch site KSC LC 39A.
- We don't have the orbit type PO and SO in the launch site CCAFS SLC 40.

Launch Success Yearly Trend

- The success ratio started to increase in 2013.
- The peak of success was in 2019
- In 2015 happened the first successful landing outcome on ground pad



All Launch Site Names

- To select the names of the launch sites without repetitions, the function "unique()" is used, placing the name of the column to be filtered between parentheses.

Display the names of the unique launch sites in the space mission

```
%sql select unique(LAUNCH_SITE) from SPACEXTBL
```

```
 * ibm_db_sa://znw88391:***@b0aebb68-94fa-46ec-a1fc-1c999edb6187.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:31249
/bludb
Done.
```

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

Launch Site Names Begin with 'CCA'

- The function "where" is a conditional that allows to filter more easily the information, when you do not have a complete text by which you want to limit the search, you can use the function "like" and place "%" at the beginning or at the end to mark the exact text you are looking for and then the rest does not matter. The Limit function only reduces the amount of data to show, in this case 5.

Display 5 records where launch sites begin with the string 'CCA'

```
%sql select * from SPACEXTBL where LAUNCH_SITE like 'CCA%' LIMIT 5
```

```
 * ibm_db_sa://ryb37360:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/bludb
Done.
```

DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-05-22	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

- To obtain the total number of payload mass carried by boosters launched by NASA, we use the function "sum()" between parenthesis placing the name of the column to be summed. Here we also use the function "where" to filter the searches in the customer column, in this case by the full name "NASA (CRS)".

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

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

```
 * ibm_db_sa://ryb37360:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/bludb
Done.
```

]:

1
45596

Average Payload Mass by F9 v1.1

- To obtain the average weight of the F9 v1.1 version boosters, we use the function "avg()" placing, again, inside the parenthesis the column of interest and with the function "where" we filter the types of boosters to "F9 v1.1".

Display average payload mass carried by booster version F9 v1.1

```
%sql select avg(payload_mass__kg_) from SPACEXTBL where booster_version='F9 v1.1'
```

```
 * ibm_db_sa://znw88391:***@b0aebb68-94fa-46ec-a1fc-1c999edb6187.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:31249
/bludb
Done.
```

1
2928

First Successful Ground Landing Date

- To obtain the date of the first successful landing of the first phase, the min() function is used, placing between brackets the column containing the dates ("date") to obtain the minimum value between them.

List the date when the first successful landing outcome in ground pad was acheived.

Hint:Use min function

```
%sql select min(date) from SPACEXTBL where landing__outcome = 'Success (ground pad)'
```

```
 * ibm_db_sa://znw88391:***@b0aebb68-94fa-46ec-a1fc-1c999edb6187.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:31249
/bludb
Done.
```

1
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- To filter the results by more than one criterion, one can use in addition to the "where" function, the "AND" function, that allows you to join each separately criteria.

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

```
%sql select booster_version from SPACEXTBL where landing__outcome = 'Success (drone ship)' and payload_mass__kg_>4000 a
nd payload_mass__kg_<6000
```

```
 * ibm_db_sa://znw88391:***@b0aebb68-94fa-46ec-a1fc-1c999edb6187.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:31249
/bludb
Done.
```

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

Total Number of Successful and Failure Mission Outcomes

- The "count()" function is used to enumerate the amount of data within the "landing_outcome" column that meets each condition in each case.

List the total number of successful and failure mission outcomes

```
%sql select count(landing__outcome) from SPACEXTBL where landing__outcome like 'Success%'
```

```
 * ibm_db_sa://znw88391:***@b0aebb68-94fa-46ec-a1fc-1c999edb6187.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:31249
/bludb
Done.
```

1
61

```
%sql select count(landing__outcome) from SPACEXTBL where landing__outcome like 'Failure%'
```

```
 * ibm_db_sa://znw88391:***@b0aebb68-94fa-46ec-a1fc-1c999edb6187.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:31249
/bludb
Done.
```

1
10

Boosters Carried Maximum Payload

- For the selection of the maximum weight of the payload it is necessary to use a subquery, which is a query that is inside another query, in this one the syntaxes “SELECT” and “FROM” are valid.

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

```
%sql select booster_version from SPACEXTBL where payload_mass__kg_ in (select max(payload_mass__kg_) from SPACEXTBL)
```

```
 * ibm_db_sa://znw88391:***@b0aebb68-94fa-46ec-a1fc-1c999edb6187.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:31249
/bludb
Done.
```

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

2015 Launch Records

- The same as in the past slides one used the function "where" to filter the data, in addition one can use the function "year()" to obtain the other condition (the year 2015), this works only if the column "date" has the correct format (DD-MM-YYYYY).

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

```
%sql select landing__outcome, booster_version, launch_site, date from SPACEXTBL where landing__outcome = 'Failure (dron
e ship)' and year(date)=2015
```

```
 * ibm_db_sa://znw88391:***@b0aebb68-94fa-46ec-a1fc-1c999edb6187.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:31249
/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

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

- To present the results collected by "Success (drone ship)" and "Success (ground pad)" one uses the function "group by" followed by the name of the column with the desired groups, for this case are only 2 groups because the data was already filtered with the successful results.

Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order

```
%sql select landing__outcome, count (landing__outcome) from SPACEXTBL where date between '2010-06-04' and '2017-03-20'
and landing__outcome like 'Success%' group by landing__outcome
```

```
 * ibm_db_sa://znw88391:***@b0aebb68-94fa-46ec-a1fc-1c999edb6187.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:31249
/bludb
Done.
```

landing__outcome	**2**
Success (drone ship)	5
Success (ground pad)	3

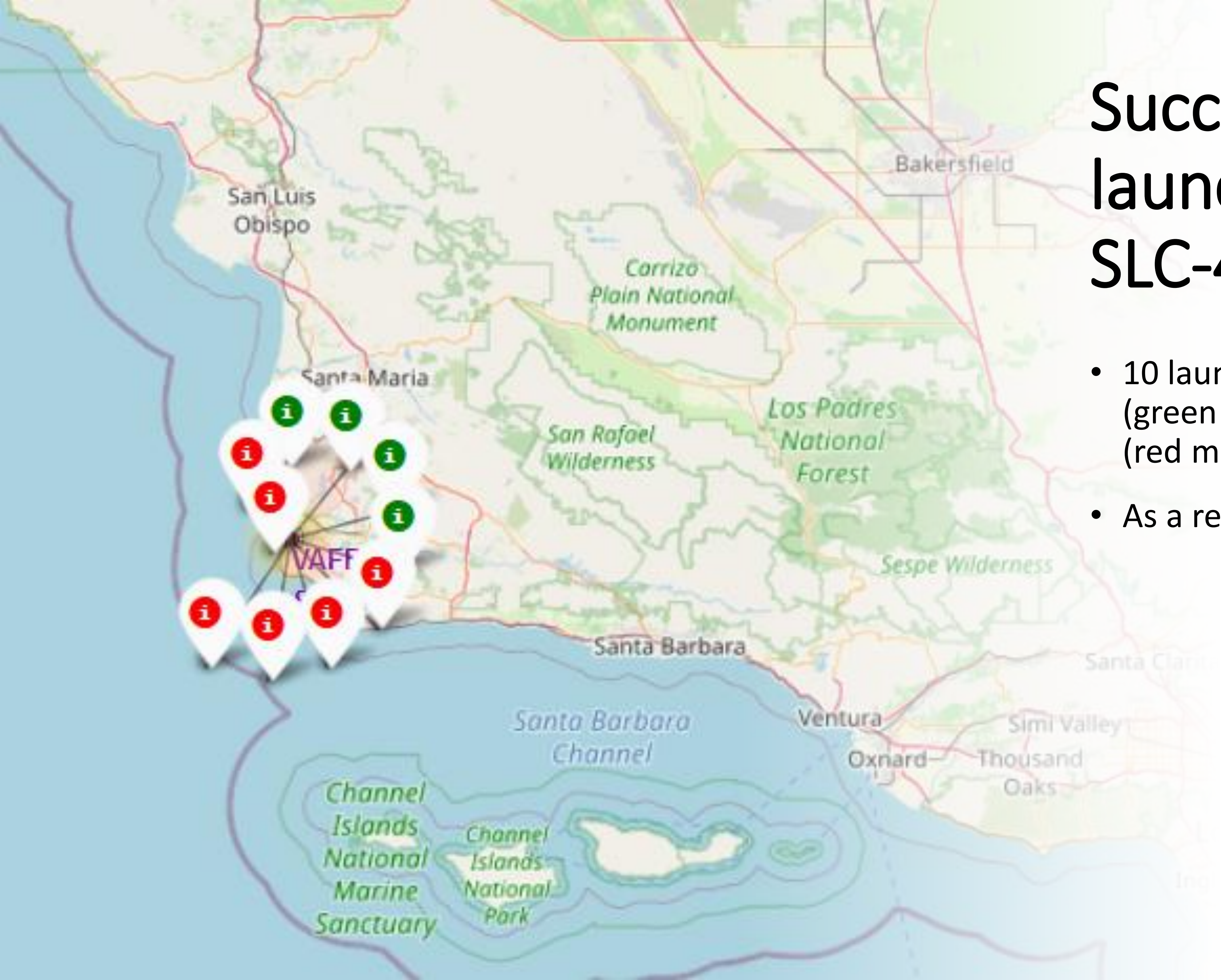
Section 4

Launch Sites Proximities Analysis

Launch Sites Locations with Folium

- The launch sites should be far distant from highly populated areas such as cities.
- The launch sites should be preferably **near the coast**

- Red Area: Cape Canaveral Space Launch Complex 40 (CCAFS LC-40)
- Blue Area: Cape Canaveral Air Force Station Space Launch Complex 40 (CCAFS SCL-40)

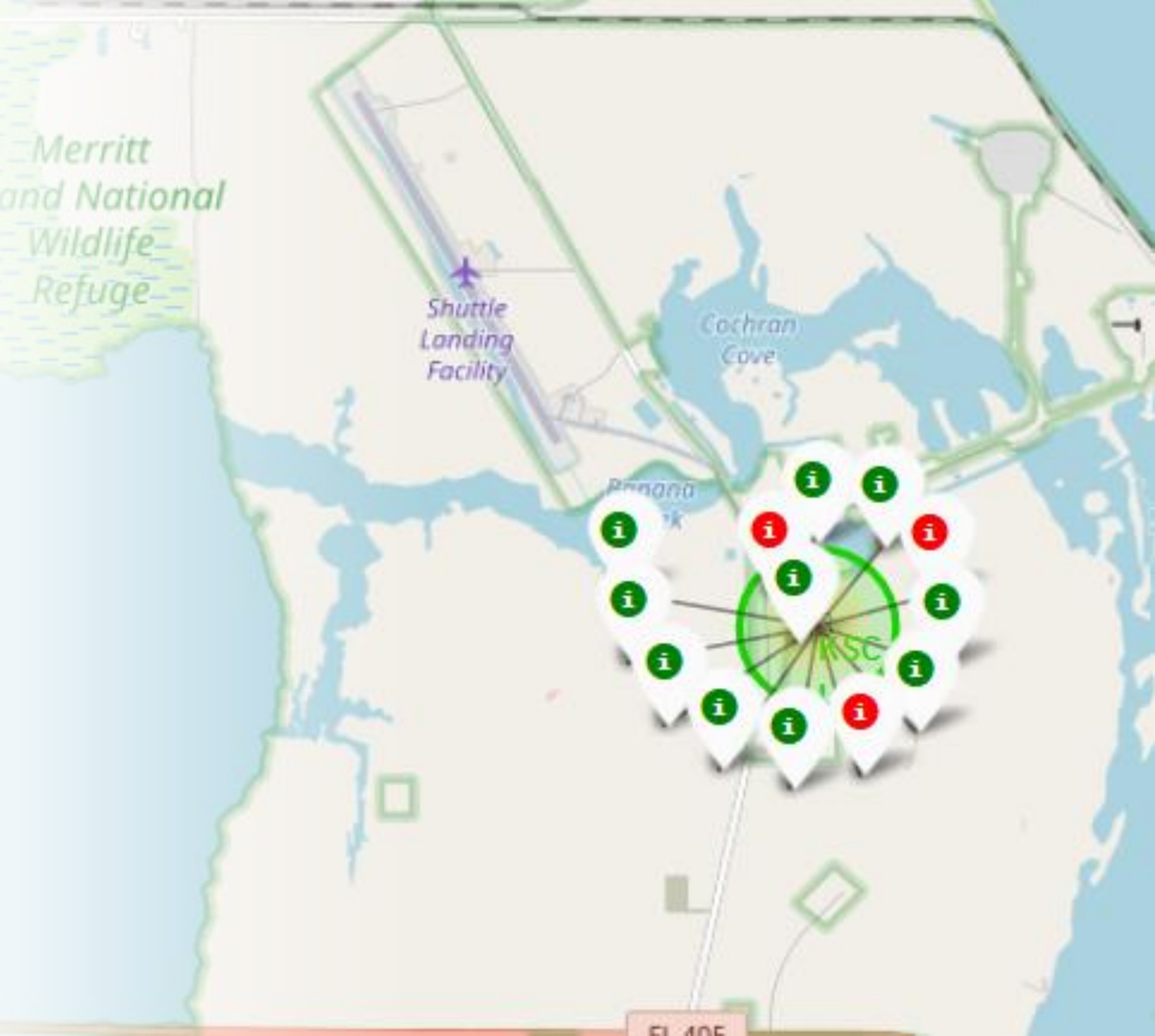


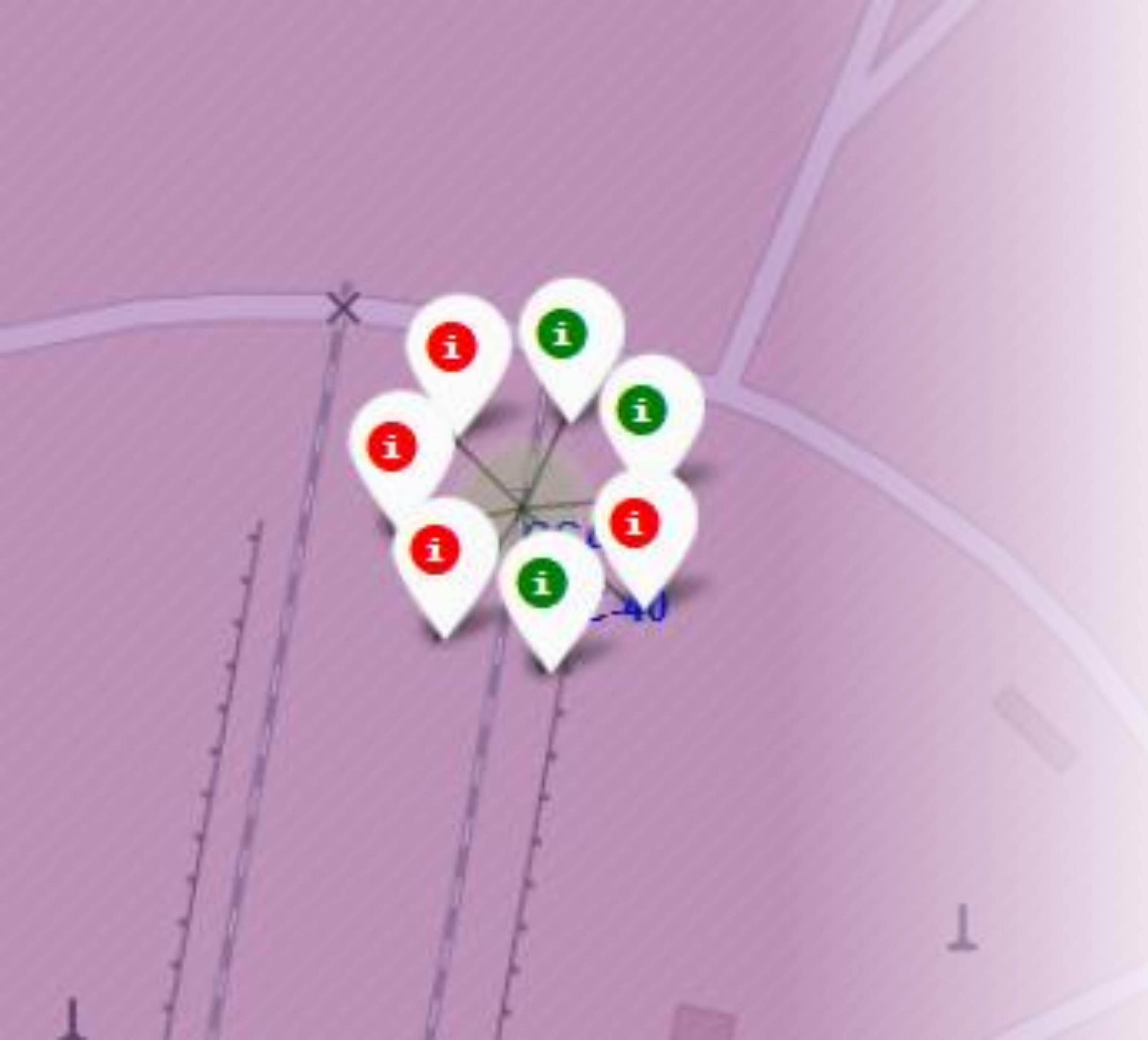
Success and failed launches for VAFB SLC-4E

- 10 launches in total, 4 successful (green markers) and 6 failures (red markers).
- As a result, a **40%** success rate

Success and failed launches for KSC-LC-39A

- 13 launches in total, 10 successful (green markers) and 3 failures (red markers).
- As a result, a **77%** success rate



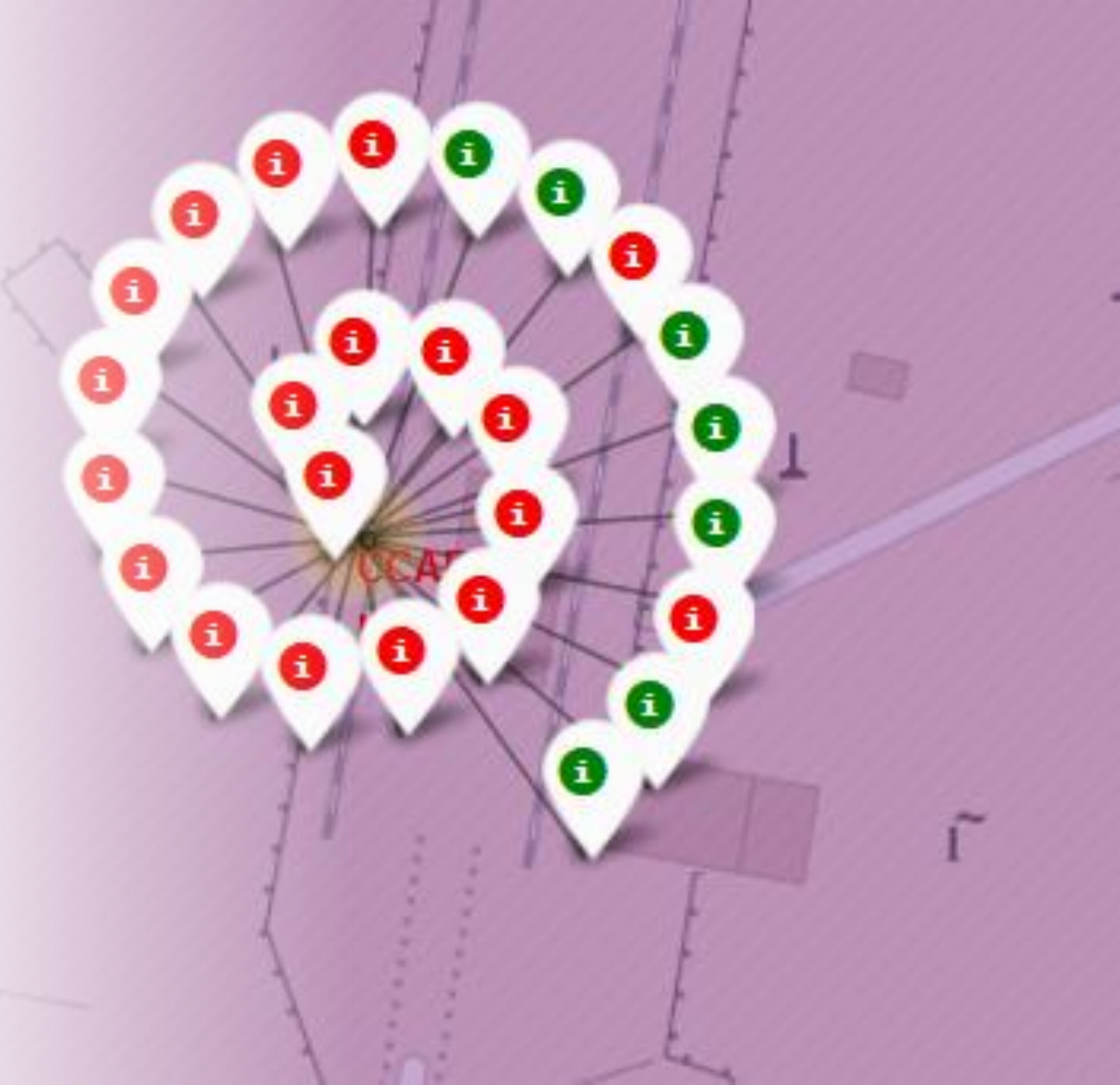


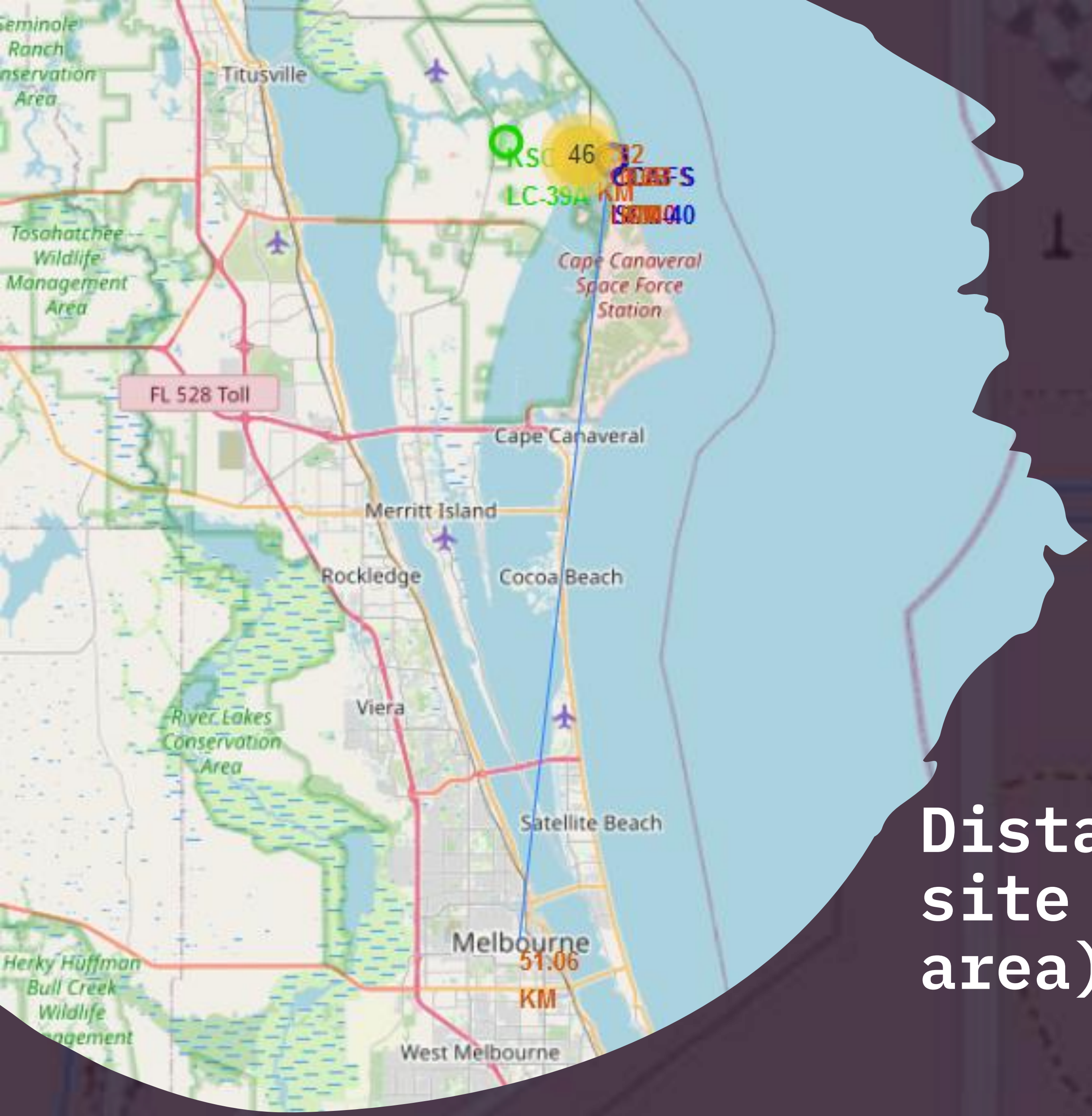
Success and failed launches for CCAFS SLC-40

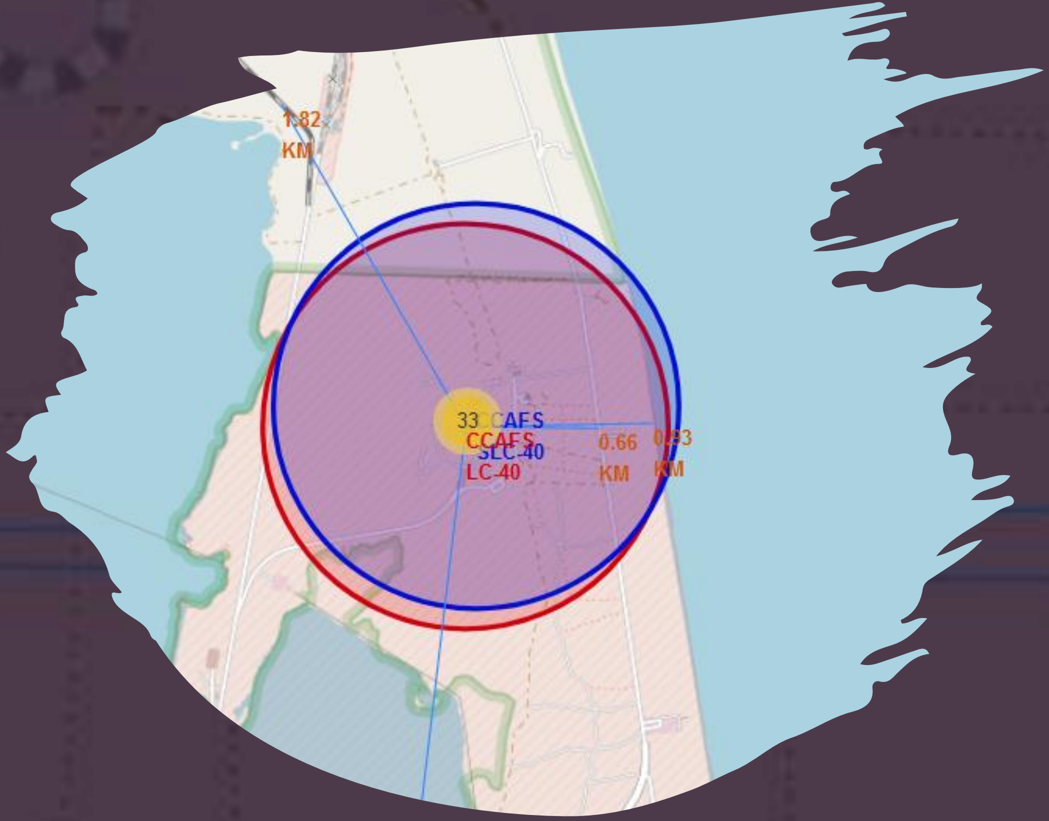
- 7 launches in total, 3 successful (green markers) and 4 failures (red markers).
- As a result, a **43%** success rate

Success and failed launches for CCAFS LC-40

- 26 launches in total, 7 successful (green markers) and 19 failures (red markers).
- As a result, a **27%** success rate



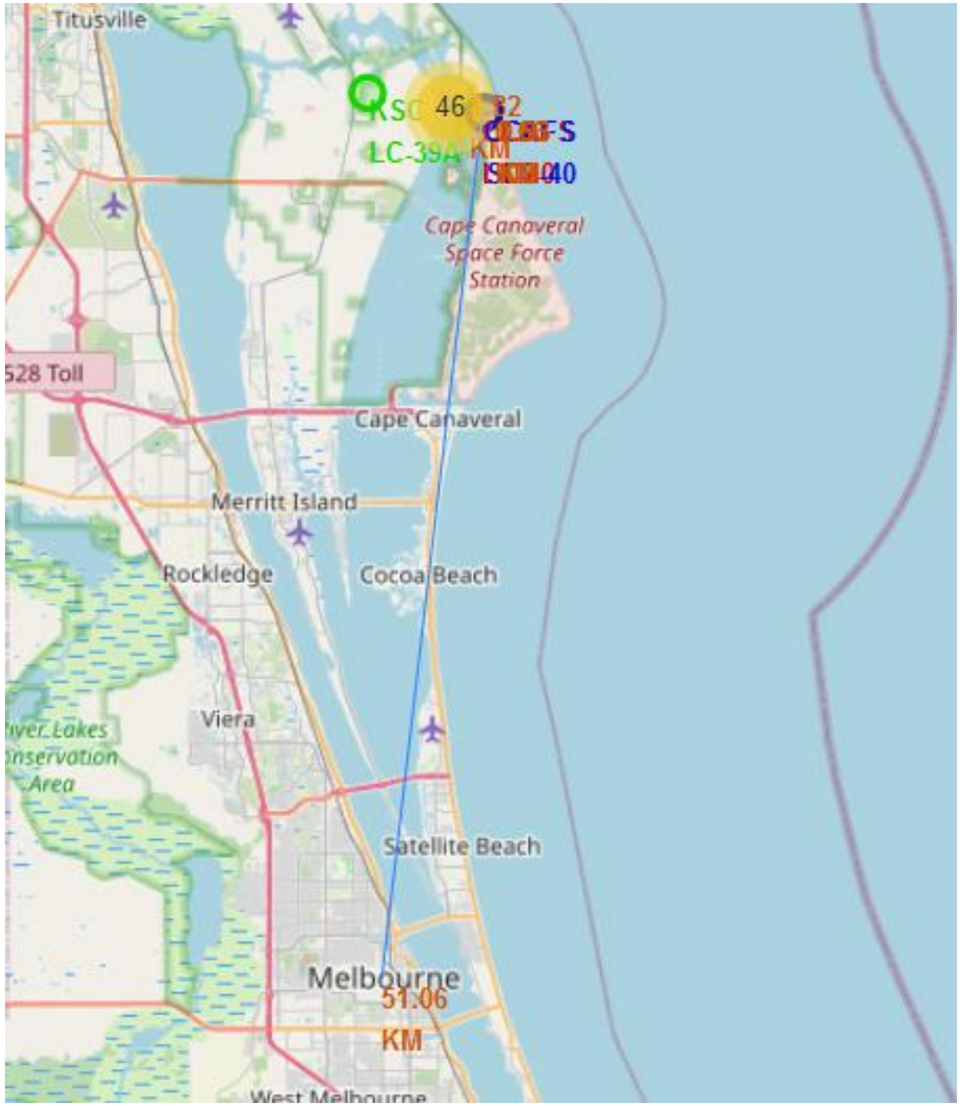




Distances between the launch site CCAFS LC-40 (the red area) to its proximities

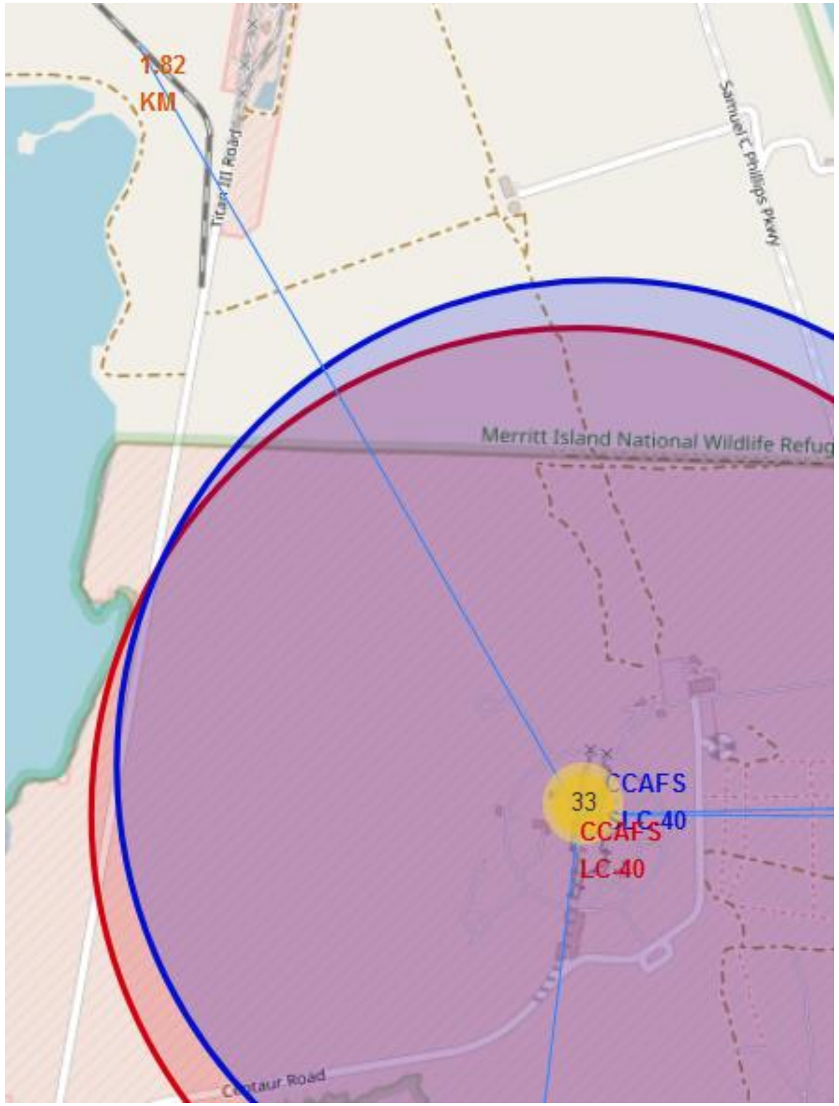
Distance between the launch site CCAFS LC-40 (the red area) to Melbourne

- The distance to the city of Melbourne is 51.06 km.
- Between the distance to the coastline, highway, railway, this is the greatest distance among all.
- The launch sites should be far distant from highly populated areas such as cities.



Distance between the launch site CCAFS LC-40 (the red area) to the NASA Railroad

- The distance to the NASA Railroad is 1.82 km.
- Between the distance to the coastline, highway, railway, this is the second greatest distance among all.
- The NASA Railroad is an industrial short-line railroad is not a passenger train.



Distance between the launch site CCAFS LC-40 (the red area) to Samuel C Phillips Pkwy and the coastline

- The distance to the Samuel C Phillips Pkwy is 0.66 km.
- The distance to the coastline is 0.93 km.
- Between all the distances this two are the shortest ones.
- The launch site is close to the coast because in case of an accident the remains would fall into the sea.
- The road runs the full length of the Cape and extends into Kennedy Space Center. Outside the Cape south gate, the road is known as Florida State Road 401. Between the Cape south gate and the industrial area, it is a four-lane divided highway with a spacious grass median.

Section 5

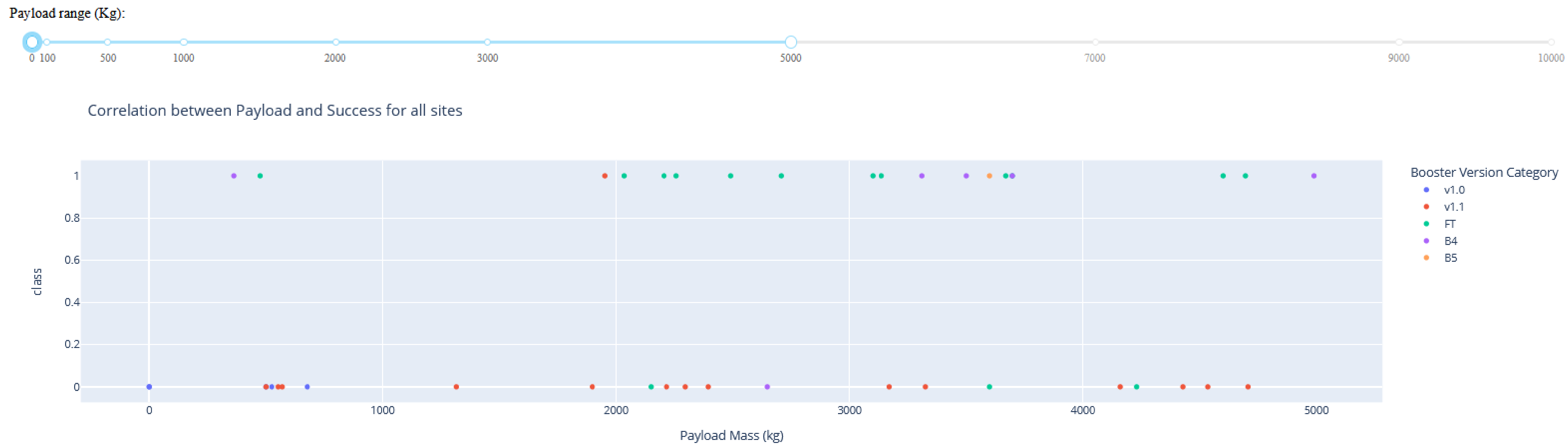
Build a Dashboard with Plotly Dash

Total success launches by site

- The site with the highest success rate is KSC LC-39A
- The site with the lowest success rate is CCAFS SLC-40

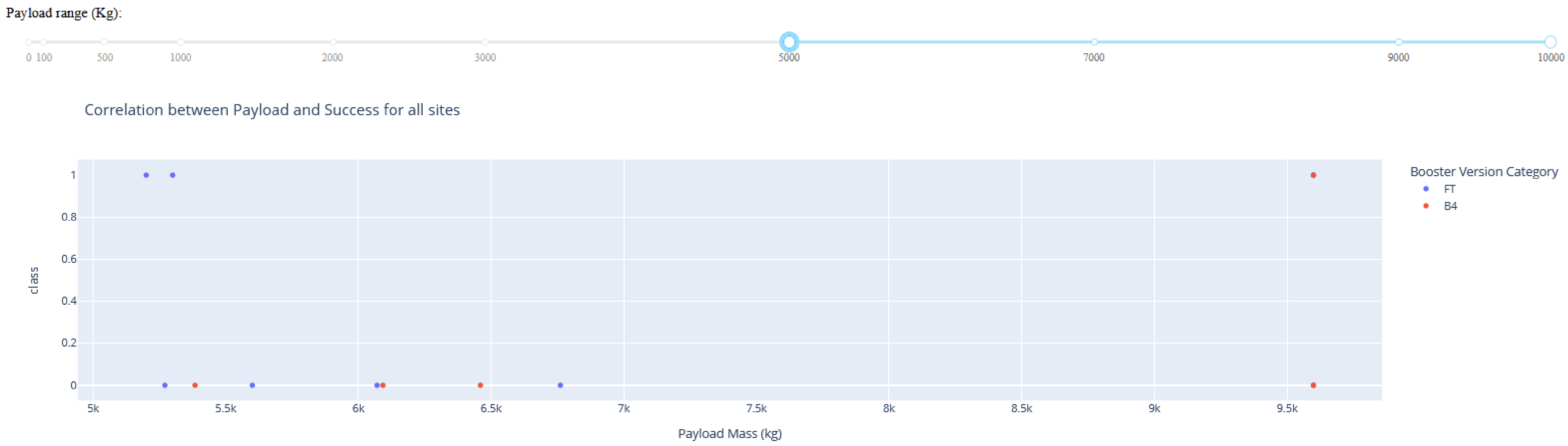
Total success launches for site KSC LC-39A

- The success rate for site KSC LC-39A is **76.9%**



Correlation between Payload (0 to 5000 kg) and success for all sites.

- The success rate of the Booster v1.0 is **0% (with 3 launches)**
- The success rate of the Booster v1.1 is **7% (with 15 launches)**
- The success rate of the Booster FT is **79% (with 14 launches)**
- The success rate of the Booster B4 is **67% (with 5 launches)**
- The success rate of the Booster B5 is **100% (with only one launch)**



Correlation between Payload (5000 to 10000 kg) and success for all sites.

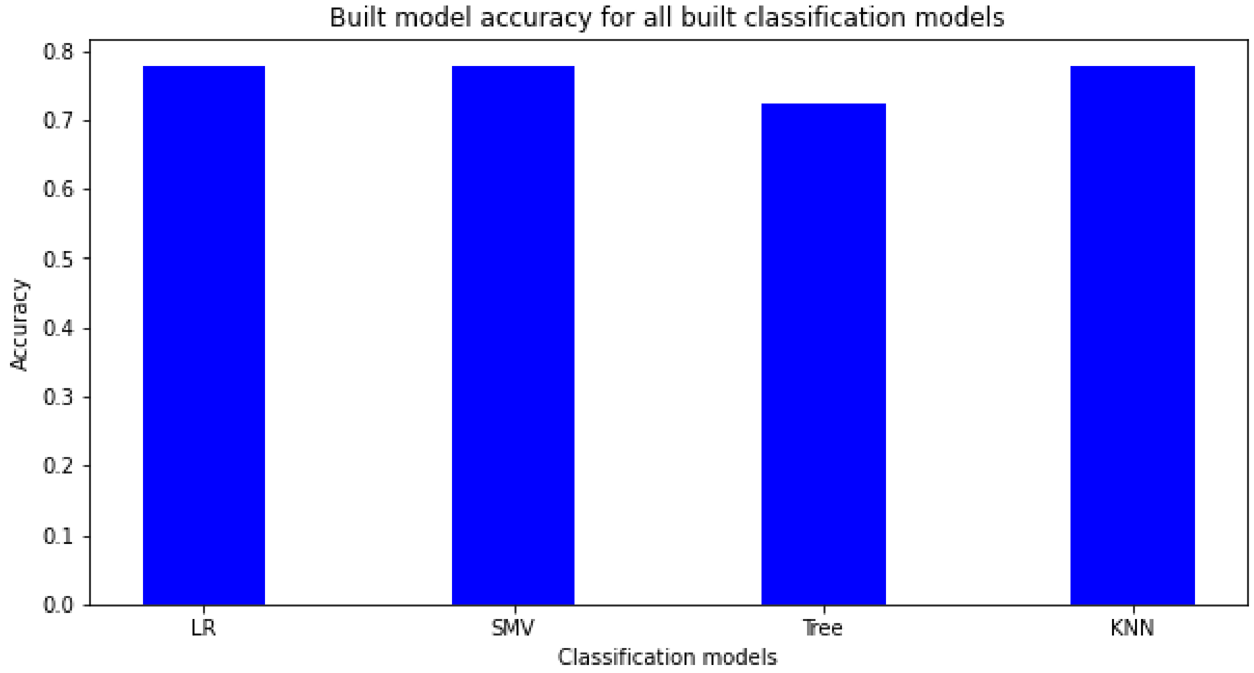
- The only booster version with the Payload between 5000 and 10000 kg are the FT and the B4
- The B4 booster version is the booster with the highest payload.
- The success rate of the Booster B4 is **20%** (with 5 launches)
- The success rate of the Booster FT is **33%** (with 6 launches)

Section 6

Predictive Analysis (Classification)

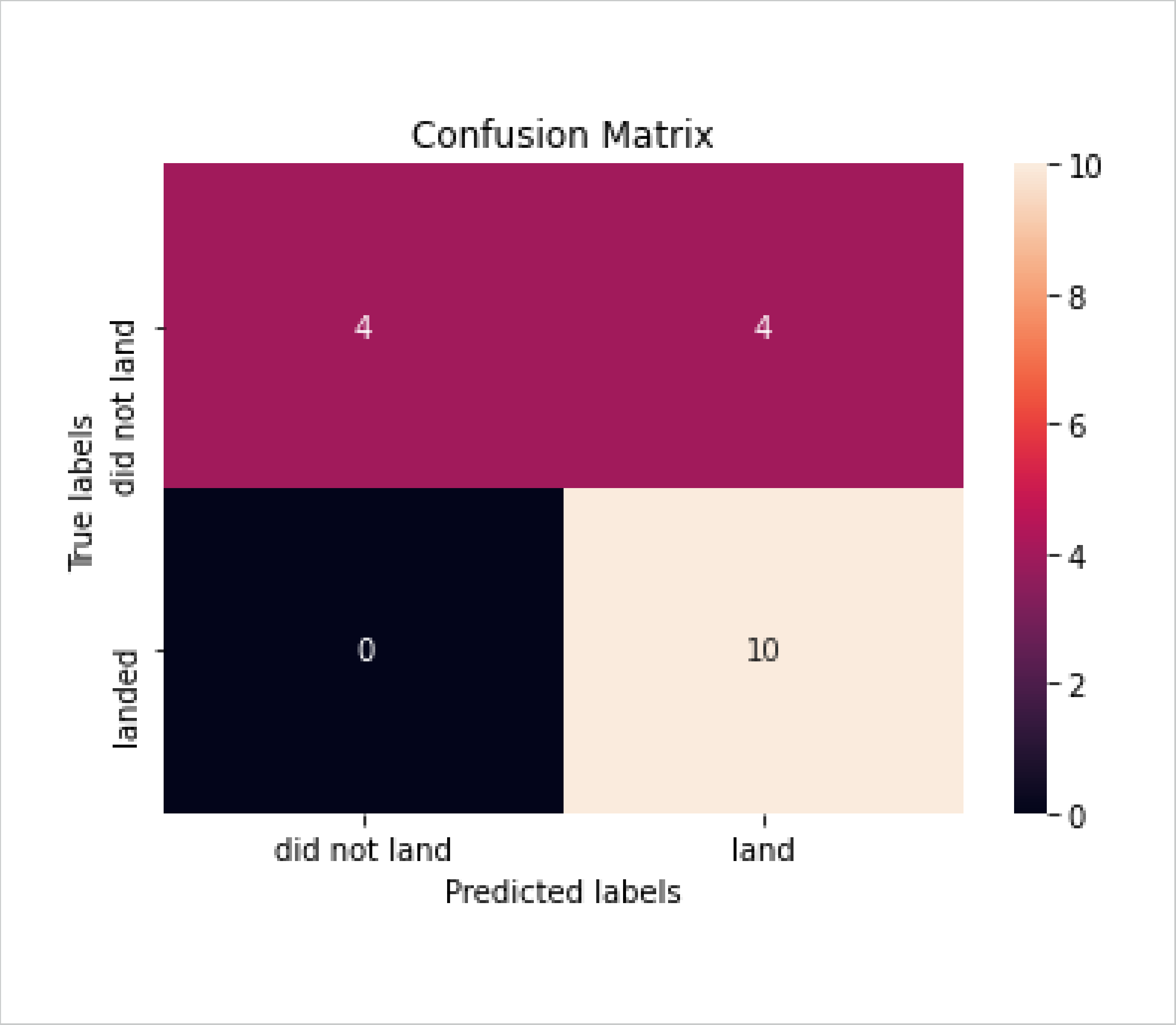
Classification Accuracy

- The accuracy of the classification models LR, SMV and KNN is **78%**
- The accuracy of the classification model Tree is **72%**
- 3 of the 4 models have the highest accuracy
- 3 of the 4 models have the same accuracy



Confusion Matrix

- We see that the major problem is false positives.
- The results are practically the same, because the dataset is small and having lesser values.
- When separating the data set for the "train and test" method, the size of the test set has only 18 values.



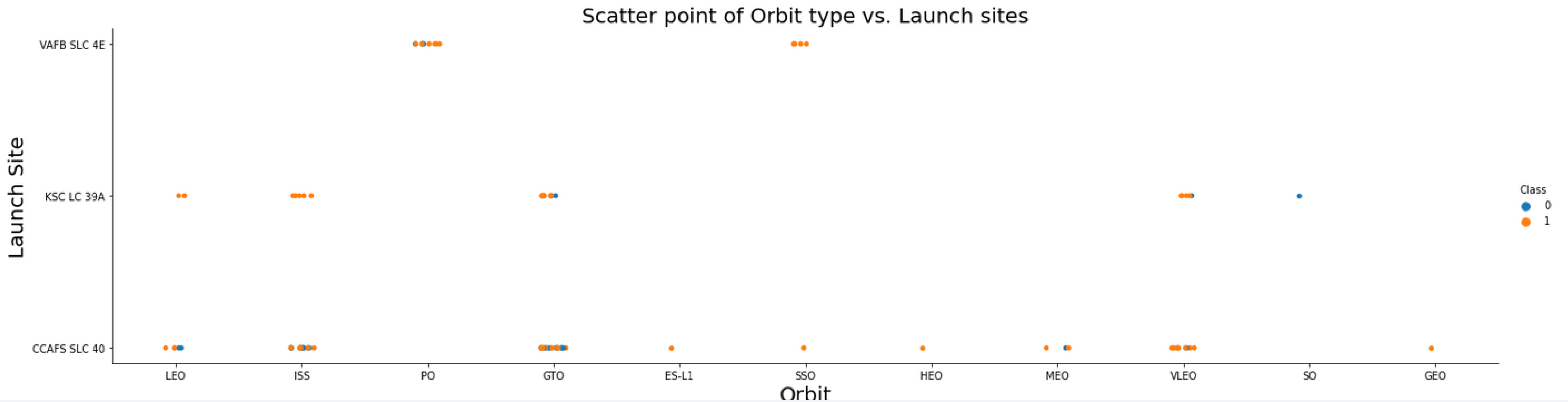
Conclusions

- The booster with highest success rate is the Booster F9 FT (Falcon 9 Full Thrust) was used for the first time in December 2015 (the year in which the success rate of the missions started to increase considerably) *[slide 32]* with this booster more than 100 successful launches have been performed.
- The launch sites should be far distant from highly populated areas such as cities, preferably near the coast, where it is possible to perform two types of landings for the recovery of the booster (on land or at sea), being the second one the one with better results based on the data obtained *[slide 42]*.
- The relationship between booster weight, orbit type and mission effectiveness should be emphasized, for example, for PO orbit, the successful missions were those where boosters weighing more than 8000 Kg were used *[slide 30]*, thus increasing considerably the cost, unlike in the case of other orbits such as HEO, GEO or LEO which have a higher success rate using smaller boosters (less than 8000 Kg) *[slide 30]*. But the type of orbit also varies the possible launch site to choose for each mission, according to the results obtained, it can be observed that in the VAFB SLC 4E site were performed only launches for PO and ISS orbits *[slide 27]*, while in the KSC LC 39A (which is the most successful launch site among all) *[slide 54]* were performed launches for LEO, ISS, GTO, VLEO and SO orbits.
- It is important to highlight the consequences of using different databases for different studies, for example, the dataset with more values resulted in the VAFB SLC site having a success rate of 77% *[slide 26]* in contrast to the second one with less values which resulted in a success rate of 40%. *[slide 45]*.
- Considering the mentioned points, Space Y first endeavor should focus on launches using a booster with similar specs as the F9 FT Booster, from a site with resources and conditions like the KSC LC39A and centered in orbits like LEO to increase the potential profit.

Appendix [A]

- Code and result of plotting "Orbit type vs. Launch sites".

```
# Plot a scatter point chart with x axis to be the Orbit and y axis to be Launch sites.
sns.catplot(y="LaunchSite", x="Orbit", hue="Class", data=df, aspect = 4)
plt.xlabel("Orbit",fontsize=20)
plt.ylabel("Launch Site",fontsize=20)
plt.title("Scatter point of Orbit type vs. Launch sites",fontsize=20)
plt.show()
```



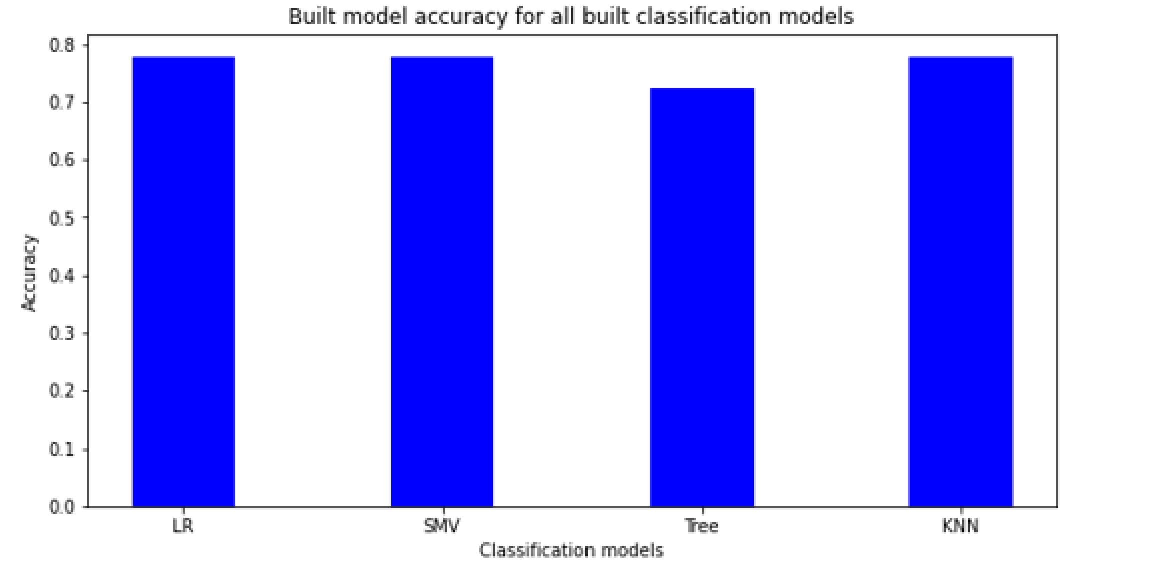
Appendix [B]

```
data = {'LR':acc_lr, 'SMV':acc_svm, 'Tree':acc_tree,
        'KNN':acc_knn}
courses = list(data.keys())
values = list(data.values())

fig = plt.figure(figsize = (10, 5))

# creating the bar plot
plt.bar(courses, values, color ='blue',
        width = 0.4)

plt.xlabel("Classification models")
plt.ylabel("Accuracy")
plt.title("Built model accuracy for all built classification models")
plt.show()
```



- Code and result of plotting the accuracy of each classification model.

Appendix [C.1]

Orbit type	Name	Distance (Km)	Use	Notes
LEO	**Low Earth Orbit**	1000	communication and remote sensing satellite systems	the International Space Station (ISS) and Hubble Space Telescope use this orbit
MEO	**Medium Earth Orbit**	10000	commonly used for navigation systems	The GPS use this orbit
GSO	**Geosynchronous Orbit**	35786	telecommunications and Earth observation	speed that matches the Earth's rotation
GEO	**Geostationary Orbit**	35786	telecommunications and Earth observation	speed that matches the Earth's rotation but only orbit Earth's equator

Appendix [C.2]

Orbit type	Name	Distance (Km)	Use	Notes
PO	**Polar Orbit**	between 200 to 1000	for satellites providing reconnaissance, weather tracking, measuring atmospheric conditions, and long-term Earth observation.	Within 30 degrees of the Earth’s poles
SSO	**Sun-Synchronous Orbit**	between 600 to 800	to monitor an area because the SSO objects pass over an Earth region at the same local time every day	A type of polar orbit, but synchronous with the sun

Appendix [C.3]

Orbit type	Name	Distance (Km)	Use	Notes
HEO	**Highly Elliptical Orbit**	Lower point under 1,000 km and a high **peak** (the point farthest from the earth) altitude of over 35,756 km.	for communications, satellite radio, remote sensing and other applications	An HEO is oblong, with one end nearer the Earth and other more distant
GTO	**Geostationary Transfer Orbit**	Elliptical orbit with an apoapsis altitude about 37,000 km	an orbit where, by using relatively little energy from built-in motors, the satellite or spacecraft can move from one orbit to another.	Elliptical orbit

Appendix [C.4]

Orbit type	Name	Distance (Km)	Use	Notes
ES-L1	**Lagrangian Point**	1.5 million kilometres inside the Earth's orbit, partway between the Sun and the Earth	space-based observatories and telescopes whose mission is to photograph deep, dark space	The most used L-points are L1 and L2
VLEO	**Very low Earth orbits**	below about 450	Earth observation.	Start in 2017
ISS	**International Space Station**	Between 360 and 440	station that serves as a space environment research laboratory	LEO orbit

Thank you!