Winning Space Race with Data Science

Cid C. de Souza
February 5, 2023

Outline

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

Executive Summary

- Project Goal
  - Predict if the SpaceX Falcon 9 first stage will land successfully
- Summary of methodologies
  - **Data collection**: SpaceX API, BeatifulSoup (webscrapping)
  - **Data wrangling**: Pandas, NumPy
  - **Exploratory Data Analysis**:
    - Data visualization: Pandas, Matplotlib, Seaborn
    - Database queries: IBM DB2, SQL scripts
  - **Geographic Analysis through Interactive Maps**: Folium
  - **Dashboard for Data Analysis**: Plotly Dash
  - **Predictive Analysis (Classification)**:
    - Logistic Regression, SVM, Classification Trees, KNN: Sklearn
    - Hyperparameter tuning: GridSearchCV
- Summary of results
  - All tested models performed equally well attaining an accuracy score of 83% and an F1 score of 89%

Introduction

- Project background and context

  Space X 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 Space X can reuse the first stage*.

  Project Goal: predict if the first stage will land as this can determine the cost of a launch.

  This information can be used if an alternate company wants to bid against Space X for a rocket launch.

- Questions to be answered
  - *General:* which mission data affect most the landing success rate?
    - What is the impact of the orbit in the landing success rate ?
    - What is the impact of the launch site in the landing success rate ?
    - What is the impact of the payload mass in the landing success rate ?

Section 1

Methodology

Methodology

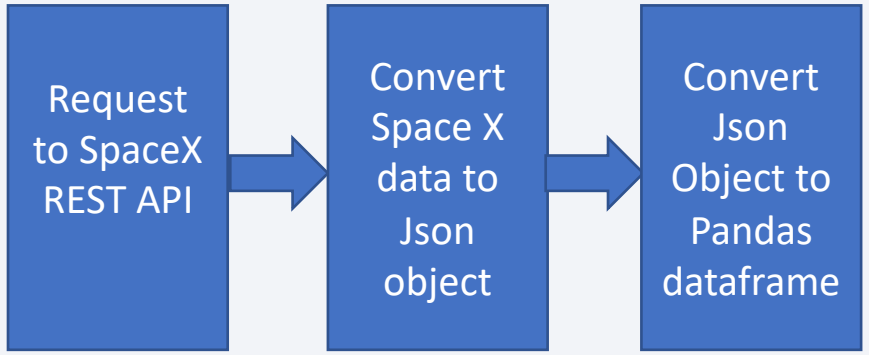
Executive Summary

- Data collection methodology:
  - Space X API: build Pandas Dataframe to ease the treatment of the data
  - Wikipedia List of Falcon 9 and Falcon Heavy launches: webscraping with BeautifulSoup
- Perform data wrangling
  - Pandas' functions used for cleaning the data and to do One Hot Encoding to apply ML models
- Perform exploratory data analysis (EDA) using visualization and SQL
  - Scatterplots, lineplots, bar graphs: MatPlotlib and Pandas
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - build, tune and evaluate classification models using the sklearn library

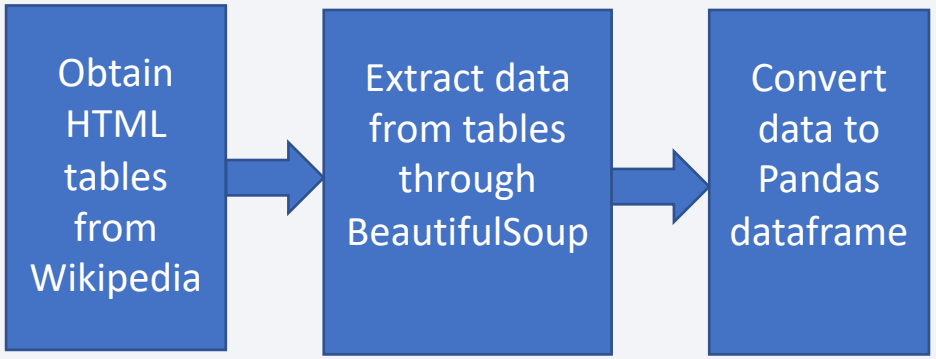
Data Collection

- Data was collected through two methods
  1. Using requests.get function from the Space X API:
     - Decode response content as a Json: .json( ) function
     - Convert Json object to pandas dataframe: .json_normalize( ) function
  2. Using webscraping from Wikipedia List of Falcon 9 and Falcon Heavy launches:
     - BeautifulSoup Python library

1. Space X API flowchart

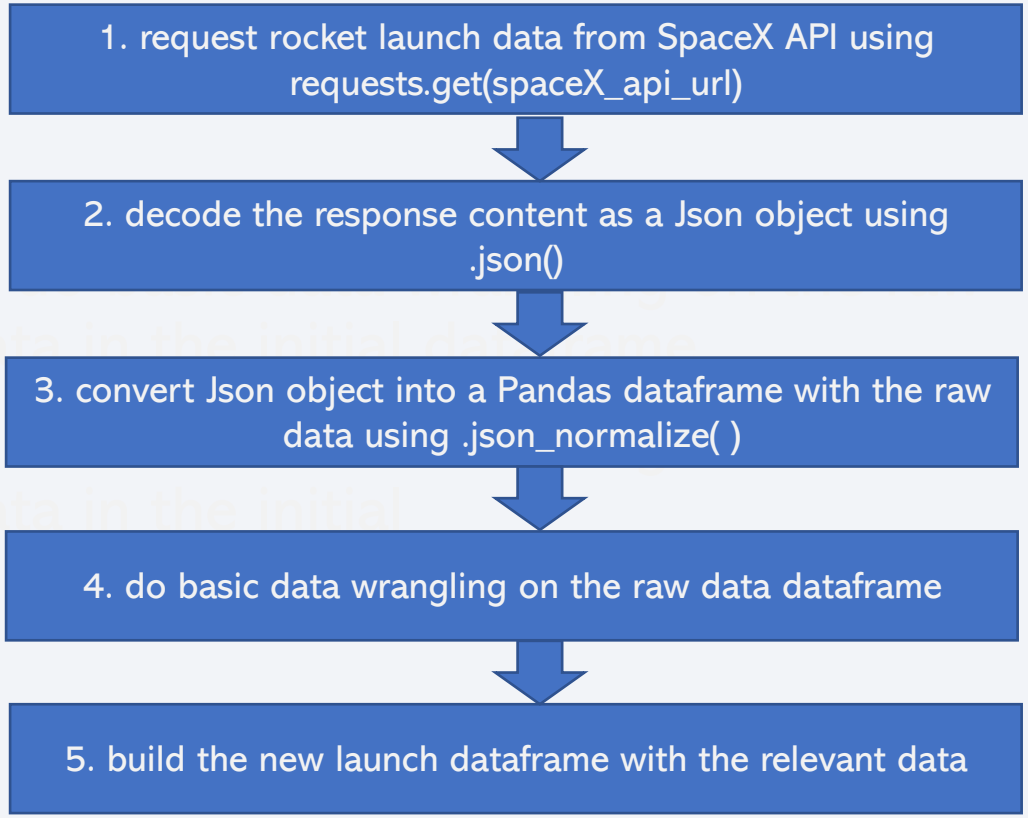


2. Webscrapping flowchart



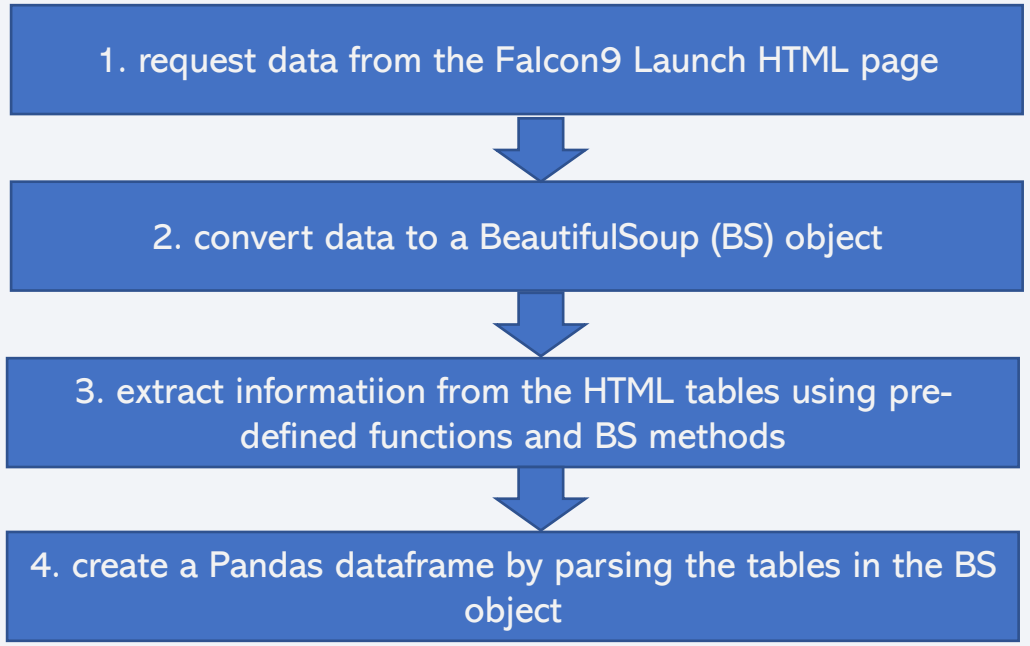
Data Collection – SpaceX API

- Basic Steps:
  1. request rocket launch data from SpaceX API using requests.get(spaceX_api_url)
  2. decode the response content as a Json object using .json( )
  3. convert Json object into a Pandas dataframe with the raw data using .json_normalize( )
  4. do basic data wrangling on the raw data in the initial dataframe
  5. build the new launch dataframe with the relevant data
- Jupyter Notebook on GitHub



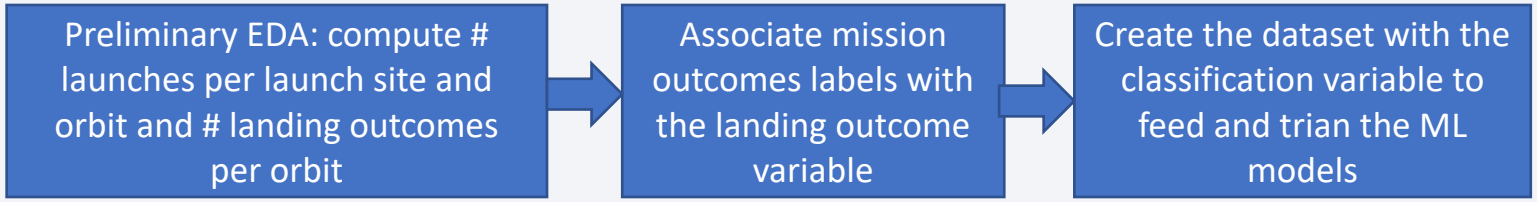
Data Collection - Webscraping

- Basic Steps:
    1. request data from the Falcon9 Launch HTML page
    2. convert data to a BeautifulSoup (BS) object
    3. extract informatiion from the HTML tables using pre-defined functions and BS methods
    4. create a Pandas dataframe by parsing the tables in the BS object
- Jupyter Notebook on GitHub



Data Wrangling

- Data processing tasks:
  - Perform preliminary Exploratory Data Analysis
  - Create classification variable to represent the outcome of a launch
- Flowchart:



- Definition of the classification variable:
  - The original dataset has several different label outcomes corresponding to cases where the booster did not land successfully. These label outcomes are processed to compute the value of the classification variable where 1 means the booster successfully landed and 0 means it was unsuccessful.
- Jupyter Notebook on GitHub

EDA with Data Visualization

- Several charts were plotted to help understanding the relation between several features and also how some features are related to the landing success rate
  - Features considered: Flight Number, Year, Payload Mass, Launch Site, Orbit and Year
- To analyze how two features are related to one another the following scatter plots were used: *Flight Number vs. Payload Mass*, *Flight Number vs. Launch Site*, *Payload vs. Launch Site*, *Orbit vs. Flight Number* and *Payload vs. Orbit*
- To show trends over time, line graphs were plotted: *Success Rate and Total Launches vs. Year* and *Total (Cumulative) Landing Failures vs. Year*
- To assess the influence of orbits in landing outcomes, the following bar graphs were plotted: *Success vs. Orbit* and *Weighted Success Rate vs. Orbit*
- Jupyter Notebook on GitHub

EDA with SQL

- The Space X launch database was uploaded to IBM DB2
- Libraries used:
  - SQLAlchemy: object-relational mapper to facilitate the communication between Python code and databases
  - ipython-sql: %sql magic for Python which allows to connect to any database supported by SQLAlchemy such as DB2
- SQL queries performed:
  - names of the unique launch sites in the space mission
  - five records where launch sites begin with the string 'KSC'
  - total payload mass carried by boosters launched by NASA (CRS)
  - average payload mass carried by booster version F9 v1.1
  - dates where the successful landing outcome in drone ship was achieved.
  - names of the boosters which have success in ground pad and have payload mass between 4000 and 6000
  - total number of successful and failure mission outcomes
  - names of the booster versions which have carried the maximum payload mass.
  - records which will display the month names, successful landing outcomes in ground pad ,booster versions, launch site for the months in year 2017
  - count of successful landing outcomes between 2010-06-04 and 2017-03-20 in descending order
- Jupyter Notebook on GitHub

Build an Interactive Map with Folium

- Launch sites were identified on the map using Folium circle object and latitude / longitude information
  - circle marks permit to easily locate the launch sites in the US territory
- Success/failed launches for each site were identified through Folium green/red marker objects on the map
  - color scheme of these markers allows a quick visual assessment of the sites' success rate
- Distances between launch sites and relevant geographic locations (such as railways, highways, coastline and cities) were computed and identified through Folium line objects joining the two locations of interest
- Jupyter Notebook on GitHub

Build a Dashboard with Plotly Dash

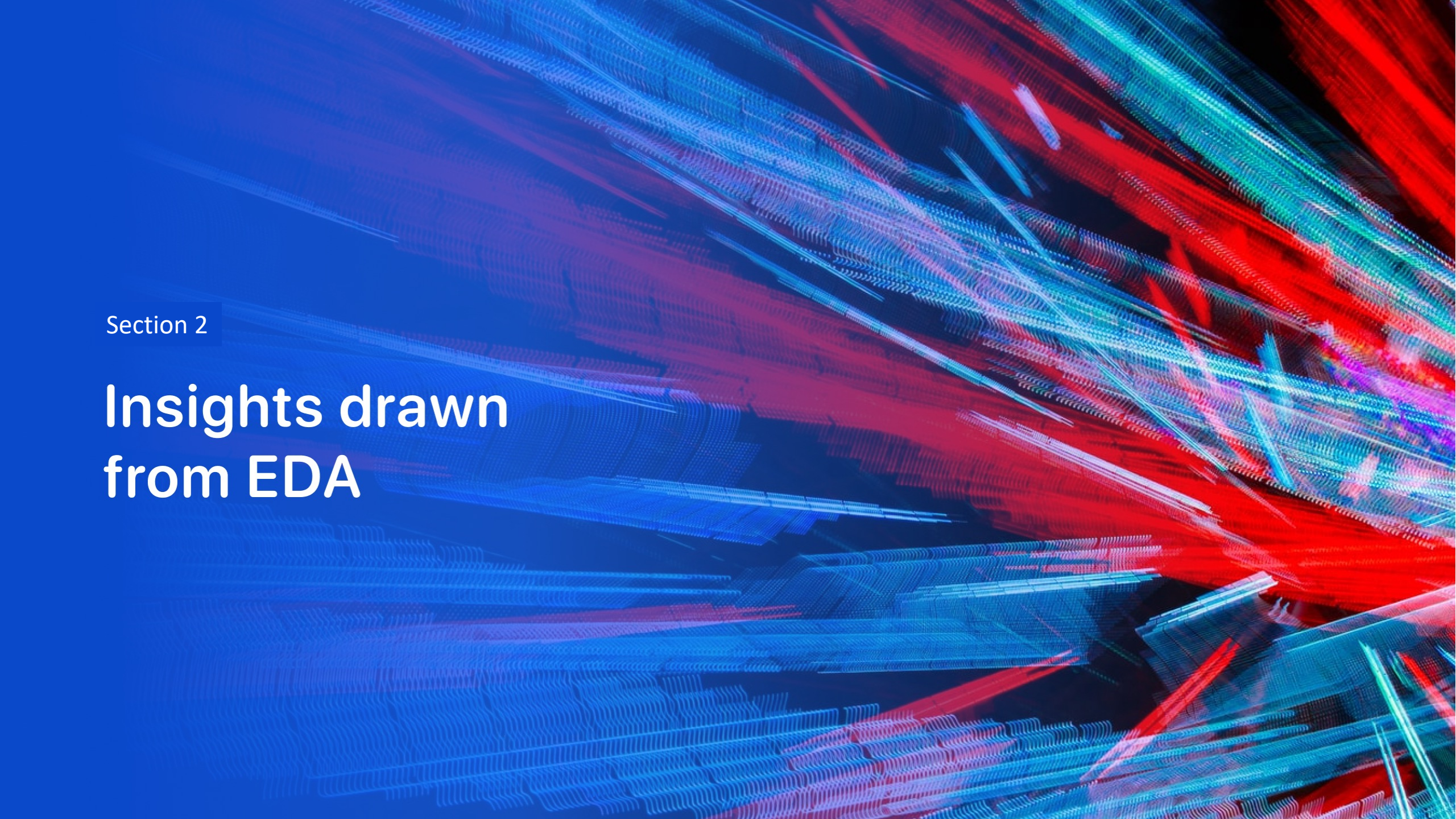
- The dashboard contains two main charts:
  - A pie chart representing the percentage of successes and failures for a selected site or for all sites according to the options offered by a dropdown menu
  - A scatter plot to visualize the correlation between Payload and Success for the selected site(s) and payload range determined through a RangeSlider object available in the dashboard
- Motivation for charts choice:
  - Pie charts are ideal to quickly evaluate the magnitude and difference of percentages for a small number of classes (two in this case, success and failure)
  - Scatter plots provide an easy way to visually check the dependency between variables
- Jupyter Notebook on GitHub

Predictive Analysis (Classification)

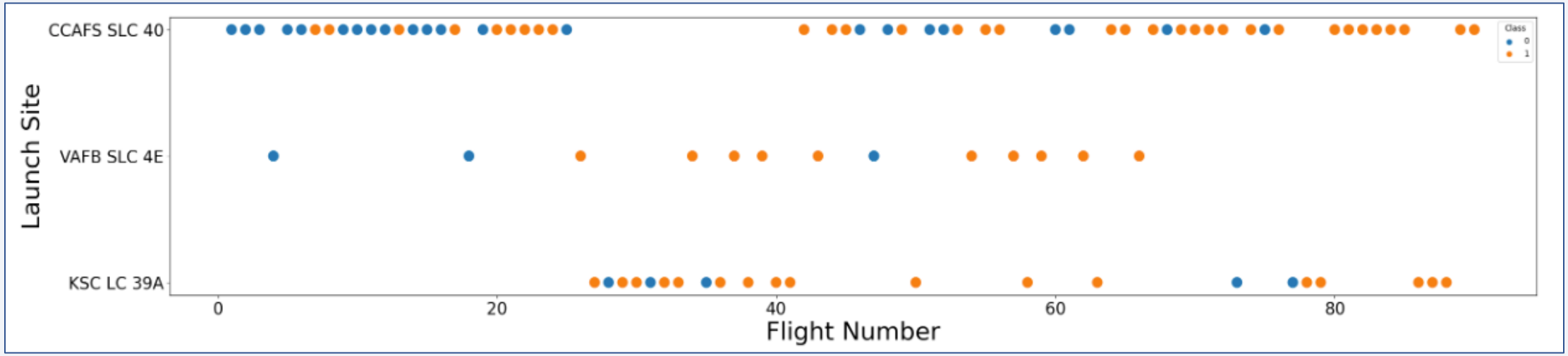
- **Main libraries used:** NumPy, Pandas and Sklearn
- **Model Building:**
  - Split dataset into training and test data sets
  - Select ML models to test: Logistic Regression, SVM, Tree and KNN
  - For each model, perform hyperparameter tuning using GridSearchCV and fit the models with the training dataset
  - Obtain the predictions of the fitted models for the test dataset
- **Model Evaluation:**
  - Check Jaccard, accuracy and F1 scores for each model
  - Plot Confusion Matrix
  - Decide which model performs best according to the scores above
- Jupyter Notebook on GitHub

Results

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

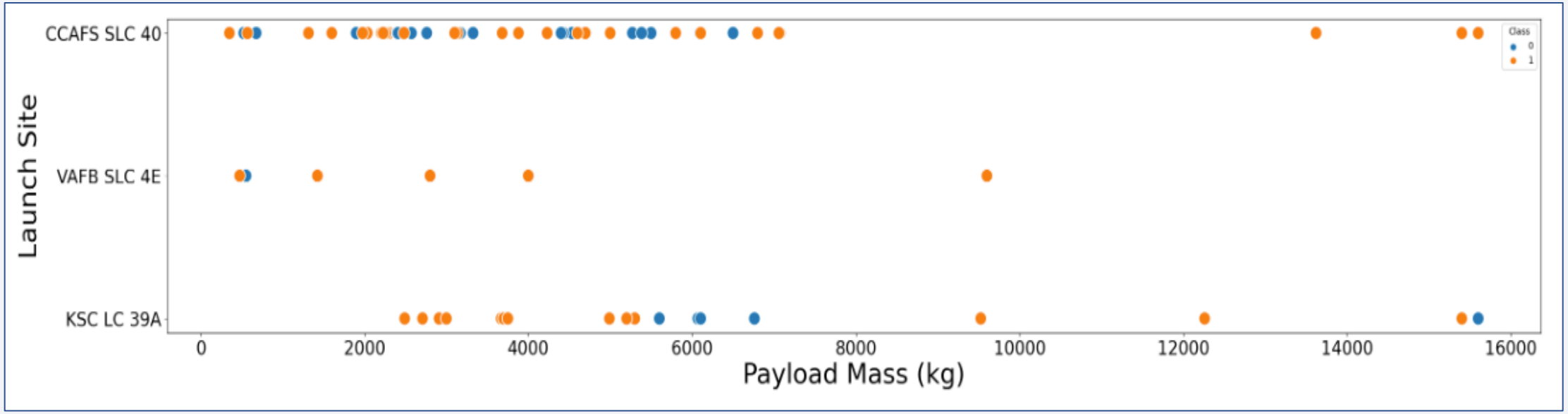

Section 2
Insights drawn from EDA

Flight Number vs. Launch Site



The success rate of site CCAFS SLC 40 is 60%, below the 77% verified for KSC LC 39A and VAFB SLC 4E. However, the scatter plot reveals that about half of the failures occurred on missions with the lowest 30 flight numbers, i.e., in the first third of total flights. The graph shows that the large majority of these initial missions, when the rocket technology was less developed, were launched from CCAFS SLC 40. This disproportionately penalizes the success rate of this site.

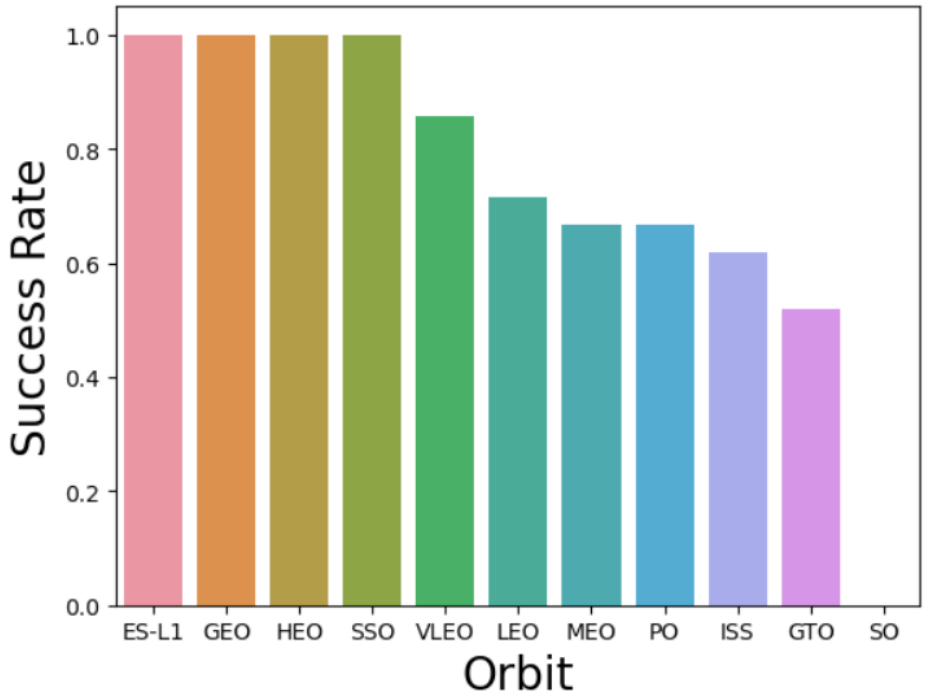
Payload vs. Launch Site

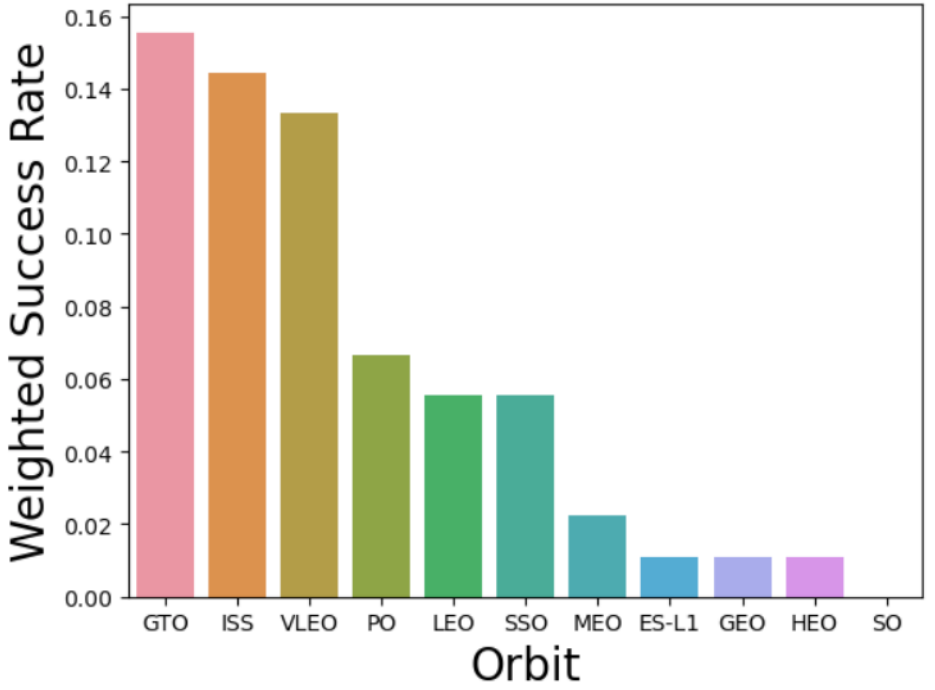


The scatterplot shows that there are no rockets launched from VAFB-SLC with heavy payload mass (greater than 10000) while the range of payload mass is much wider for the two other launch sites.

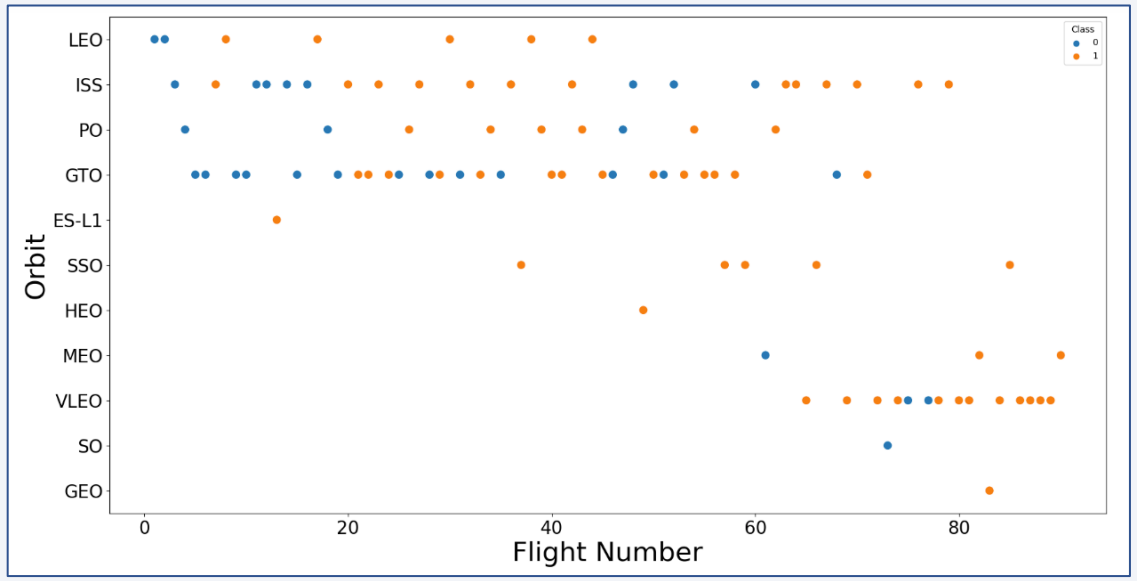
Success Rate vs. Orbit Type

The success rate is 100% for ES-L1, GEO, HEO and SSO. However, the number of launches for the 3 first orbits are is just ONE! Consequently, the success rate has small statistical significance. Among the orbits with 100% success rate, SSO is the only one whose number of launches is >= 5, the median. To mitigate the problem we define the weighted success rate of site i as: WSR(i) = SR(i) * Num_Launches(i) / Total_Launches. This takes into account the number of launches for each orbit (note that the sum of the weighted success rates (WSR) is equal to the overall success rate of the dataset). Following this analysis, we can see that GTO, ISS and VLEO (that have 27, 21 and 14 launches respectively) have the highest weighted success rates and they are more than twice the value obtained for PO.





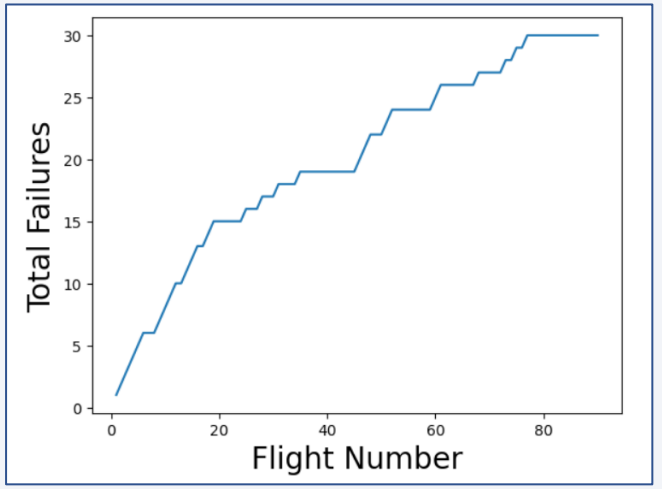
Flight Number vs. Orbit Type



From the graph it is unclear how success relates to the orbits and the flight numbers. 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.

Inspecting the next graph we confirm the intuition that there are more failures in early flights (the ones with smaller numbers) possibly due to the continuous technological improvements.

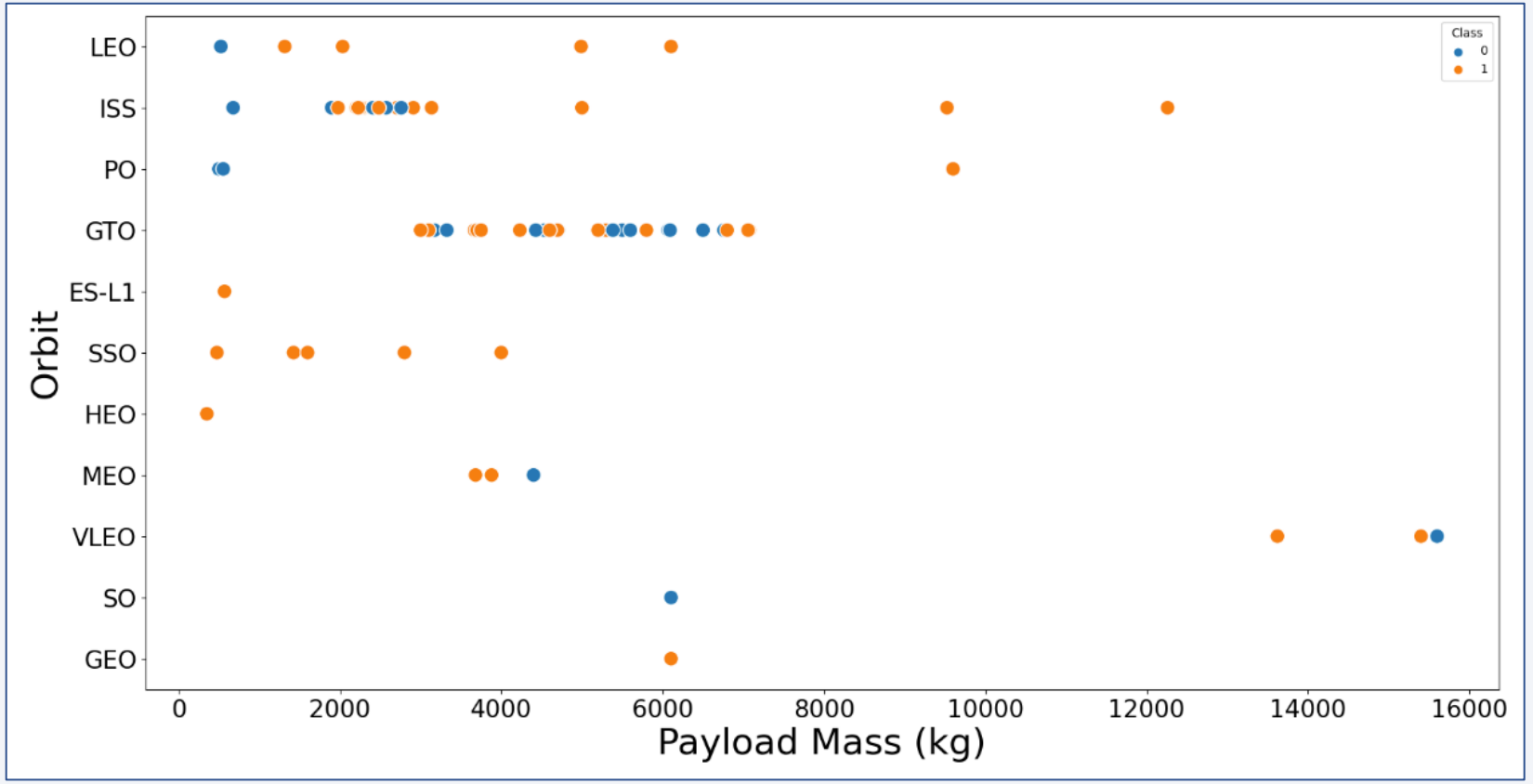
For instance, in the first 30 flights the number of failures was 17, in the next 30 ones the failures dropped to 8 and, finally, in the last 30 flights only 5 failures occurred.



Since LEO, ISS, PO and GTO concentrate all but one of the 30 first flights, this reduces the success rates of these orbits. On the other hand, for SSO, all flight numbers are larger than 30 and they all succeeded. This is an indication that the success rate of an orbit is less dependent on the orbit itself and more related to the improvements of the technology over time.

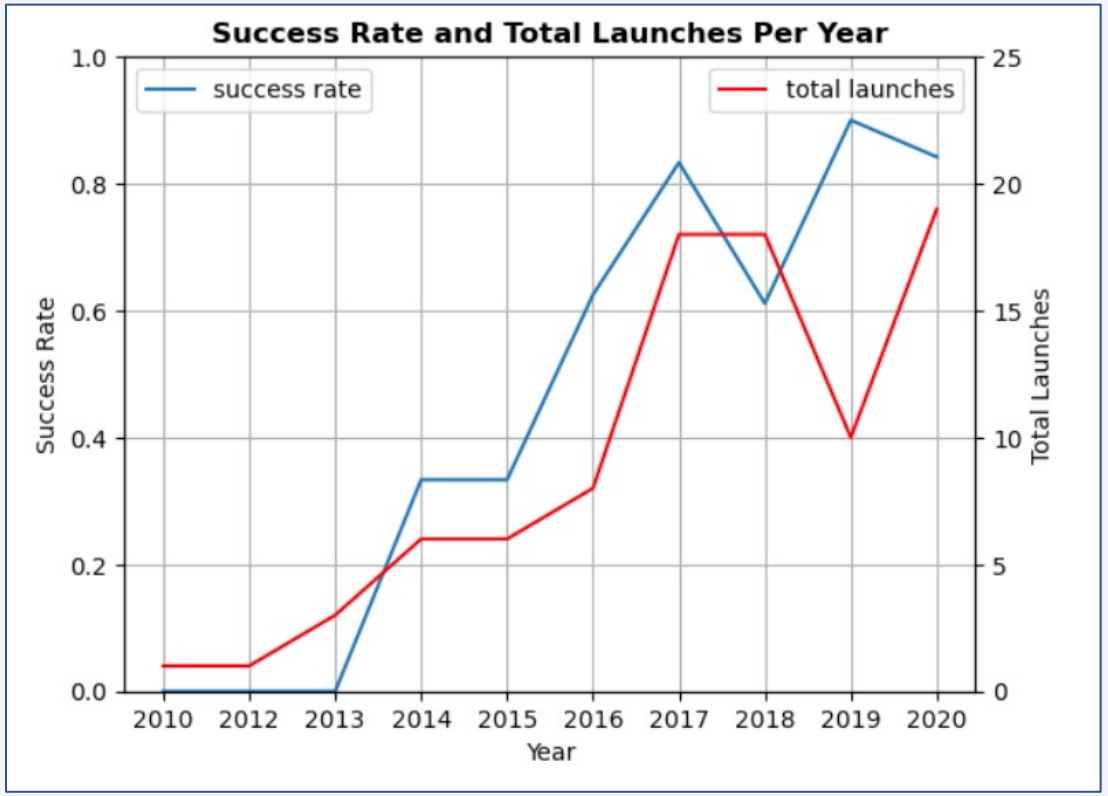
Payload vs. Orbit Type

- For some orbits like LEO and ISS the success rate increases with payload mass. However, for some others, payload mass seems to have little or no influence in the landing outcome, at it is the case for GTO.



Launch Success Yearly Trend

The graph gives an interesting insight. Both the success rate and the number of launches increased (monotonically) from 2010 to 2017. In 2018, the success rate reduced to the same level as in 2016, although the number of launches remained the same as in the previous year. This loss in performance led to a reduction in the number of launches in 2019 (from 18 to only 10) which suggests that Space X must have slowed down the launches in search for the causes of the failures. If this was indeed the case, the effect was positive as the success rate rose again in 2019, slightly decreasing in 2020 when the number of launches attained its peak in the period of this analysis.



All Launch Site Names

```
In [4]: %sql SELECT DISTINCT launch_site FROM spacex2;
 * ibm_db_sa://tht47923:***@125f9f61-9715-46f9-9399-c8177b21803b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:3042
6/bludb
Done.
```

Out[4]:

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

Explanation: the reserved word DISTINCT in the SQL query is used to filter the unique values in the launch_site column from spacex2

Launch Site Names Begin with 'CCA'

```
In [5]: %sql SELECT *, launch_site FROM spacex2 \
        WHERE lcase(launch_site) like 'cca%' \
        FETCH first 5 rows only;
```

```
 * ibm_db_sa://tht47923:***@125f9f61-9715-46f9-9399-c8177b21803b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:3042
6/bludb
Done.
```

Out[5]:

DATE	time__utc_	booster_version	launch_site	payload	payload_mass_kg_	orbit	customer	mission_outcome	landing_outcome	launch_site_1
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)	CCAFS LC-40
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)	CCAFS LC-40
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	CCAFS LC-40
2012-10-08	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt	CCAFS LC-40
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt	CCAFS LC-40

Explanation: the WHERE clause in the SQL query selects all rows whose launch_site column has the prefix `cca`, regardless if these three letters are written in capitals or not. The FETCH clause ensures that only the first five rows satisfying the previous condition are kept

Total Payload Mass

```
In [6]: %sql SELECT sum(PAYLOAD_MASS_KG_) FROM SPACEX2 \
        WHERE lcase(customer) like '%nasa (crs)%';

         * ibm_db_sa://tht47923:***@125f9f61-9715-46f9-9399-c8177b21803b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:3042
        6/bludb
        Done.

Out[6]:  1
         48213
```

Explanation: the WHERE clause in the SQL query selects all rows whose customer column contains the substring `nasa (crs)`, regardless if the letters in the substring are written in capitals or not. The sum( ) function in the SELECT call returns the total for column PAYLOAD_MASS_KG_

Average Payload Mass by F9 v1.1

```
In [7]: %sql SELECT AVG(PAYLOAD_MASS_KG_) FROM SPACEX2 \
        WHERE lcase(Booster_Version) like '%f9 v1.1%';
         * ibm_db_sa://tht47923:***@125f9f61-9715-46f9-9399-c8177b21803b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:3042
        6/bludb
        Done.
Out[7]:  1
         2534
```

Explanation: the WHERE clause in the SQL query selects all rows whose Booster_Version column contains the substring `f9 v1.1`, regardless if the letters in the substring are written in capitals or not. The avg( ) function in the SELECT call returns the average for column PAYLOAD_MASS_KG_

First Successful Ground Landing Date

```
In [8]: %sql SELECT MIN(DATE) FROM SPACEX2 \
        WHERE lcase(Landing_Outcome) like '%success%' \
              AND lcase(Landing_Outcome) like '%ground pad%';

         * ibm_db_sa://tht47923:***@125f9f61-9715-46f9-9399-c8177b21803b.clogj3sd0tgtu0lqde00.databases.appdomain.cloud:3042
        6/bludb
        Done.

Out[8]:
          1
        2015-12-22
```

Explanation: the WHERE clause in the SQL query selects all rows whose Landing_Outcome column contains the substrings `success` and `ground pad`, regardless if the letters in the substrings are written in capitals or not. The min( ) function in the SELECT call returns the minimum for column DATE for all selected rows

Successful Drone Ship Landing with Payload between 4000 and 6000

```
In [8]: %sql SELECT MIN(DATE) FROM SPACEX2 \
        WHERE lcase(Landing_Outcome) like '%success%' \
              AND lcase(Landing_Outcome) like '%ground pad%';

         * ibm_db_sa://tht47923:***@125f9f61-9715-46f9-9399-c8177b21803b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:3042
        6/bludb
        Done.

Out[8]:
          1
        2015-12-22
```

Explanation: the WHERE clause in the SQL query selects all rows whose Landing_Outcome column contains the substrings `success` and `drone ship`, regardless if the letters in the substrings are written in capitals or not. Besides, after the last AND, the WHERE keeps only these rows where the value in column PAYLOAD_MASS_KG_ is between 4000 AND 6000. The SELECT command specifies that only the columns Booster_Version, PAYLOAD_MASS_KG_ and Landing_Outcome of the selected rows are reported by the query

Total Number of Successful and Failure Mission Outcomes

```
In [10]: %sql SELECT Mission_Outcome, COUNT(*) FROM SPACEX2 \
         GROUP BY Mission_Outcome;

          * ibm_db_sa://tht47923:***@125f9f61-9715-46f9-9399-c8177b21803b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:3042
         6/bludb
         Done.
```

Out[10]:

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

Explanation: the GROUPBY clause in the query forces the reported results to be aggregated according to the unique values in column Mission_Outcome, which define the different groups for which the outputs are calculated. The final values reported at the output of the SQL script are the number of occurrences of rows in each of these groups, as required by the function COUNT.

Boosters Carried Maximum Payload

```
In [11]: %sql SELECT DAte, Booster_Version, PAYLOAD_MASS_KG_ FROM SPACEX2 \
         WHERE PAYLOAD_MASS_KG_ = (SELECT MAX(PAYLOAD_MASS_KG_) from SPACEX2);

          * ibm_db_sa://tht47923:***@125f9f61-9715-46f9-9399-c8177b21803b.clogj3sd0tgtu0lqde00.databases.appdomain.cloud:3042
         6/bludb
         Done.
```

Out[11]:

DATE	booster_version	payload_mass_kg_
2019-11-11	F9 B5 B1048.4	15600
2020-01-07	F9 B5 B1049.4	15600
2020-01-29	F9 B5 B1051.3	15600
2020-02-17	F9 B5 B1056.4	15600
2020-03-18	F9 B5 B1048.5	15600
2020-04-22	F9 B5 B1051.4	15600
2020-06-04	F9 B5 B1049.5	15600
2020-09-03	F9 B5 B1060.2	15600
2020-10-06	F9 B5 B1058.3	15600
2020-10-18	F9 B5 B1051.6	15600
2020-10-24	F9 B5 B1060.3	15600
2020-11-25	F9 B5 B1049.7	15600

Explanation: in the subquery (last SELECT command) the function MAX returns the maximum value in the column PAYLOAD_MASS_KG_. Then, the WHERE clause selects all rows in the table SPACEX2 for which the payload mass is equal to the computed maximum. Finally, the SELECT command specifies that only the columns Date, Booster_Version and PAYLOAD_MASS_KG_ of the selected rows are reported by the query

2015 Launch Records

```
In [12]: %sql SELECT date, Landing_Outcome, Booster_Version, Launch_site FROM SPACEX2 \
         WHERE (lcase(Landing_Outcome) like '%failure%' AND \
               lcase(Landing_Outcome) like '%drone ship%') \
               AND YEAR(DATE) = 2015;

 * ibm_db_sa://tht47923:***@125f9f61-9715-46f9-9399-c8177b21803b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:3042
6/bludb
Done.
```

Out[12]:

DATE	landing_outcome	booster_version	launch_site
2015-01-10	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
2015-04-14	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

Explanation: the WHERE clause in the SQL query selects all rows whose Landing_Outcome column contains the substrings `failure` and `drone ship`, regardless if the letters in the substrings are written in capitals or not. Besides, after the last AND, it selects only the rows where the value in column DATE whose YEAR, returned by the function with this same name, returns 2015. The SELECT command specifies that only the columns DATE, Landing_Outcome, Booster_Version, Launch_site of the selected rows are reported by the query

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

```
In [13]: %sql SELECT Landing_Outcome, COUNT(Landing_outcome) as CNT_LAND FROM SPACEX2 \
WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' \
GROUP BY Landing_Outcome \
ORDER BY CNT_LAND DESC;
```

```
 * ibm_db_sa://tht47923:***@125f9f61-9715-46f9-9399-c8177b21803b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:3042
6/bludb
Done.
```

Out[13]:

landing_outcome	cnt_land
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

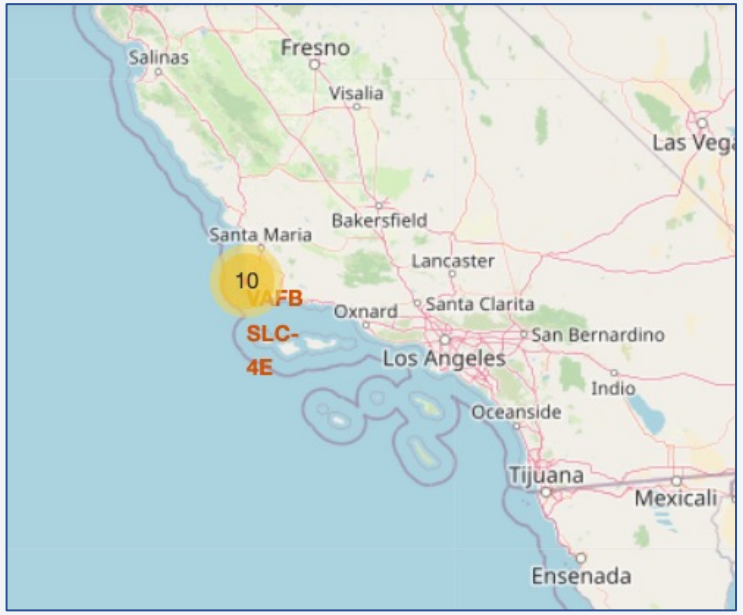
Explanation: the WHERE clause selects rows where the value in the DATE columns is in the required time interval. The GROUPBY clause forces the reported results to be aggregated according to the unique values in column Landing_Outcome and the ORDER clause forces them to appear in decreasing order of the counting values computed by the COUNT function in the SELECT command.

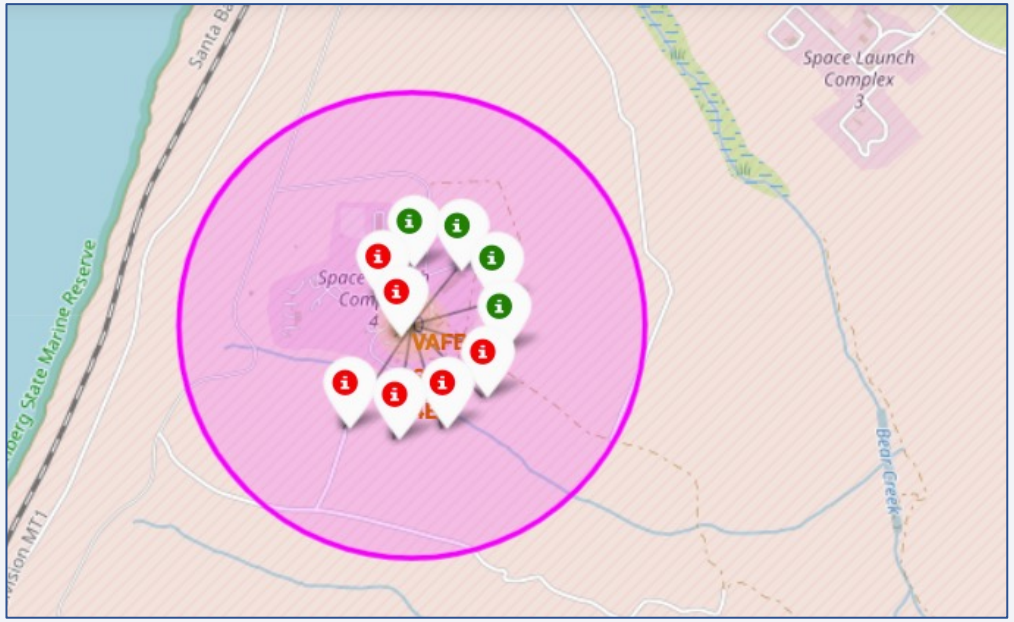

Section 3
Launch Sites
Proximities Analysis

Launch Site Locations

- Space X launch sites are all located in the United States and close to the coastline, either in Florida or in California
- Folium Circle Markers are used to identify these sites in the world map

Landing Outcome per Site with Folium Markers



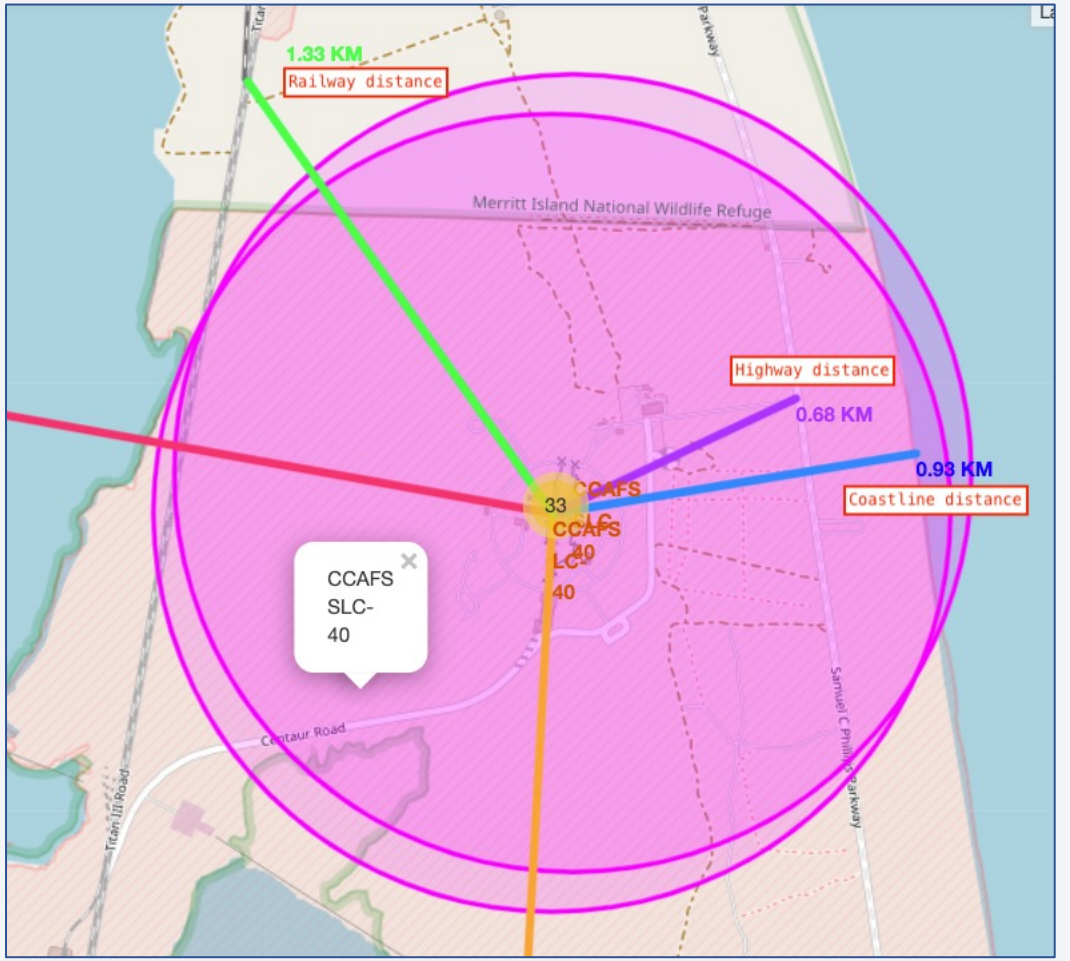


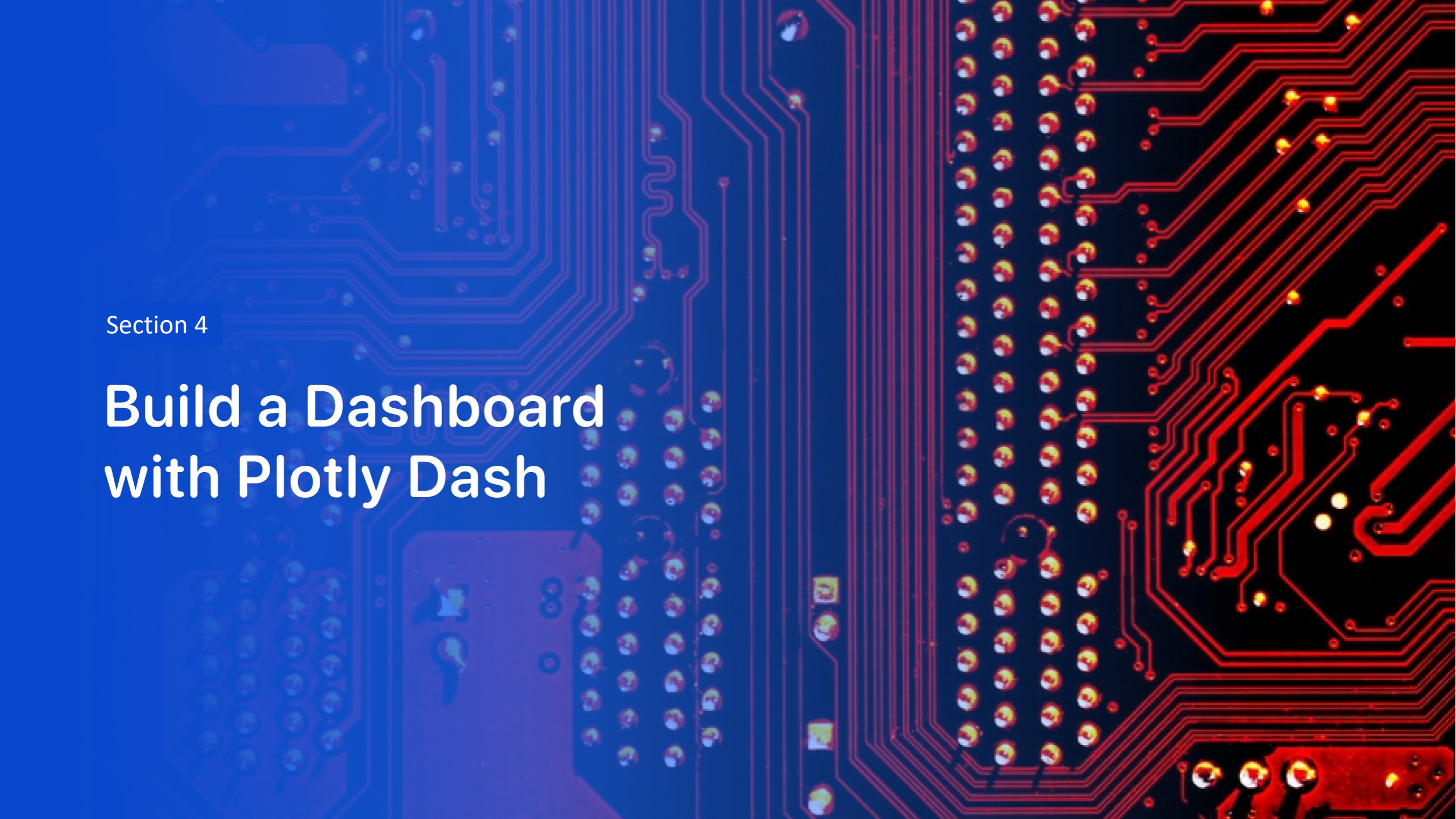
- A Marker Cluster is created for each launch site represented by a yellow circle. Inside the circle is the number of launches from the location
- Clicking on one such marker, we can see small green and red markers corresponding, respectively, to a successful or to a failed landing outcome
- The screenshots displayed correspond to the VAFB SLC 4E launch site

Launch Site distance to landmarks

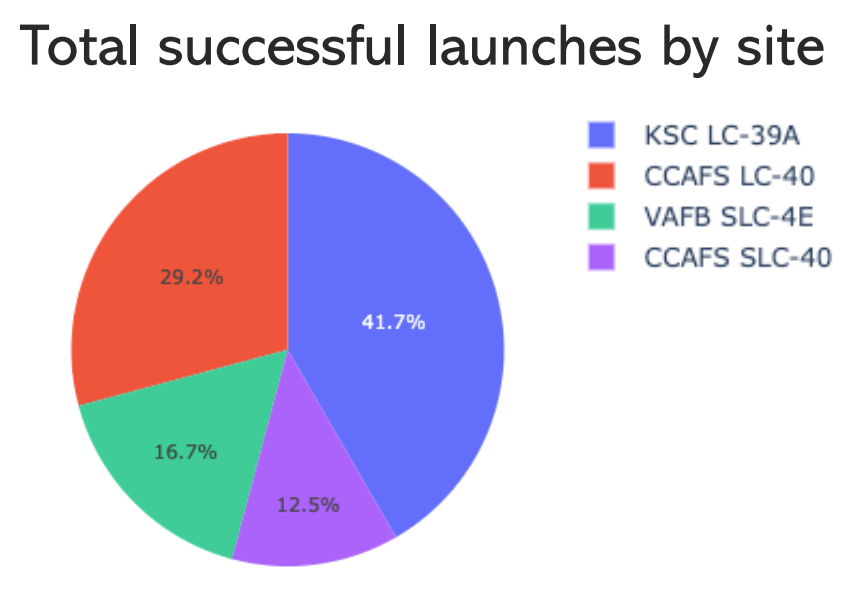
- **Questions analyzed:**
  - Are launch sites in close proximity to railways? Yes
  - Are launch sites in close proximity to highways? Yes
  - Are launch sites in close proximity to coastline? Yes
  - Do launch sites keep certain distance away from cities? Yes
- Conclusions:

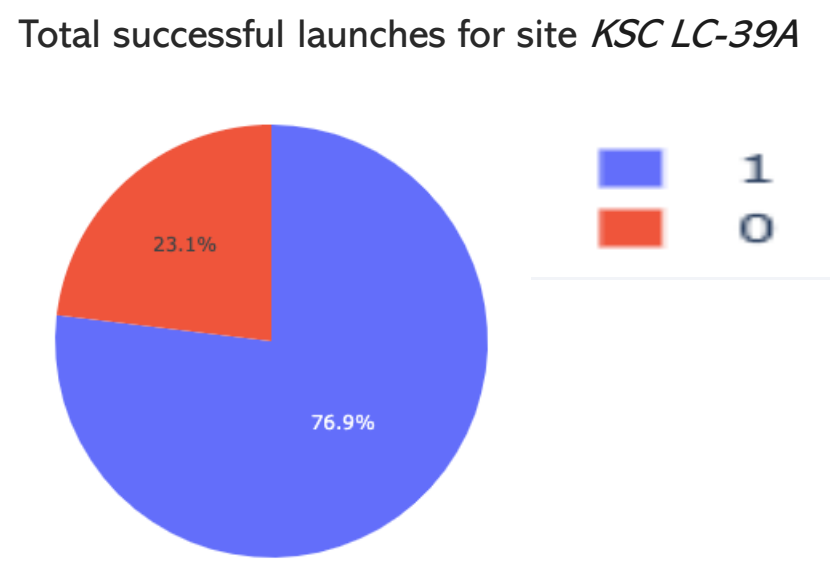
  Proximity to highways and railways ease the logistics, for instance, to bring parts of the rockets to the launch site. On the other hand, being away for urban centers reduces the chances of life losses in case of an accident during launching




Section 4
Build a Dashboard with Plotly Dash

Dashboard – Pie Charts and Successful Outcomes



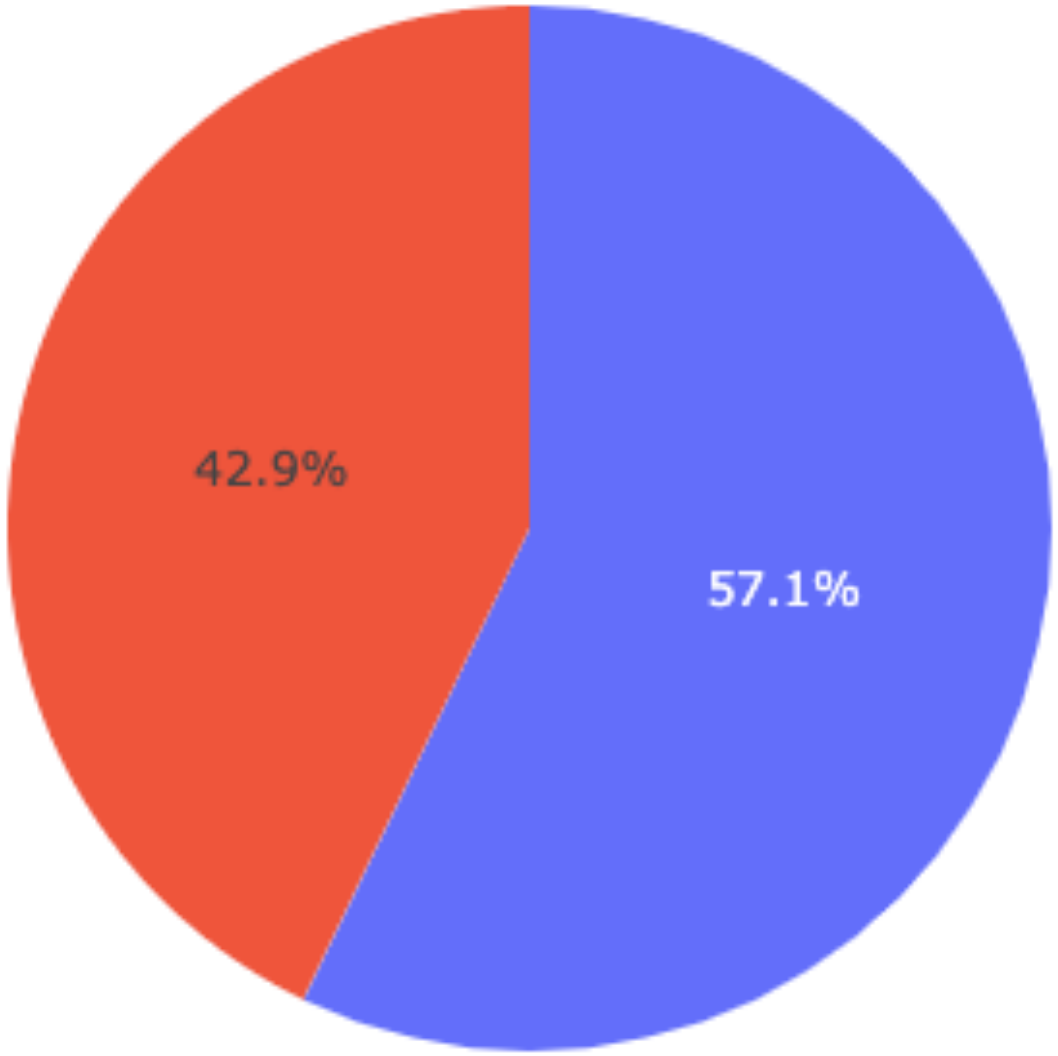


KSC LC-39A *is the launch site with the highest number of successful missions*

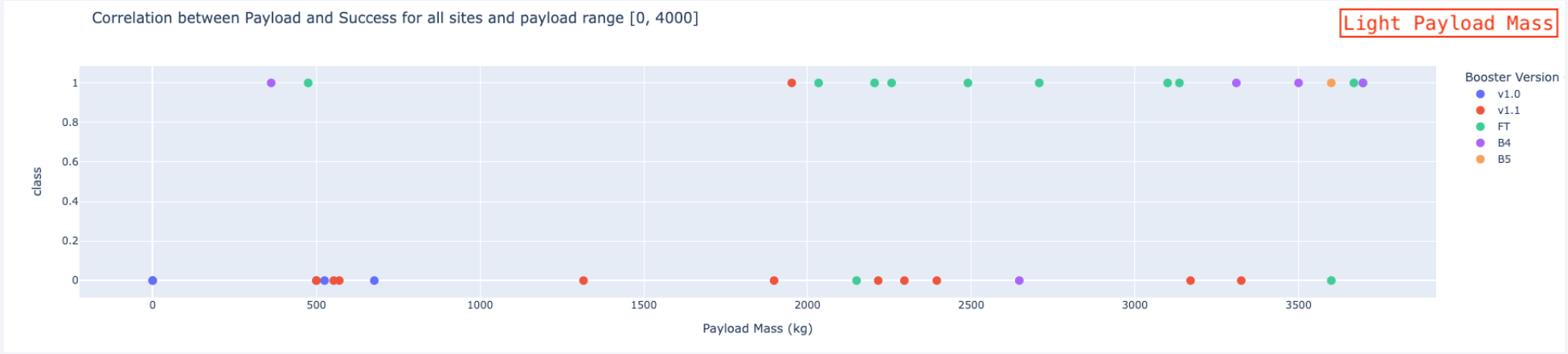
In KSC LC-39A 76.9% of the missions had successful outcomes with only 23.1% failing

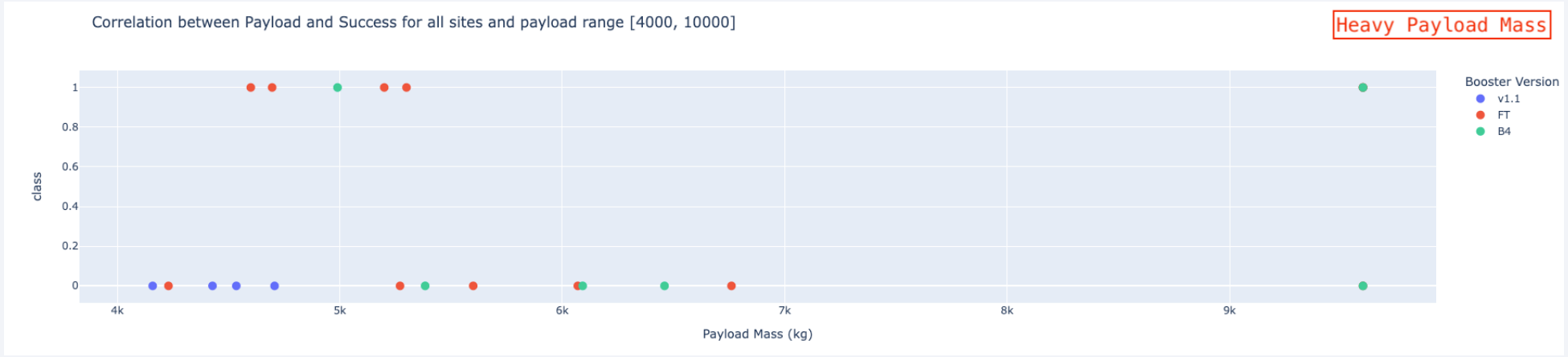
Dashboard – Pie Chart of Most and Least Successful Sites

The previous slide shows the pie chart containing the percentage of successful missions for KSC LC-39A, the site with highest success ratio. On the right is the equivalent chart for CCAFS SLC-40, the site with the lowest ratio. As seen, the ratio is not homogeneous for different sites



Dashboard – Payload Mass and Success Rates





Analysis: the success rates for light payloads is higher than the one for heavy payloads

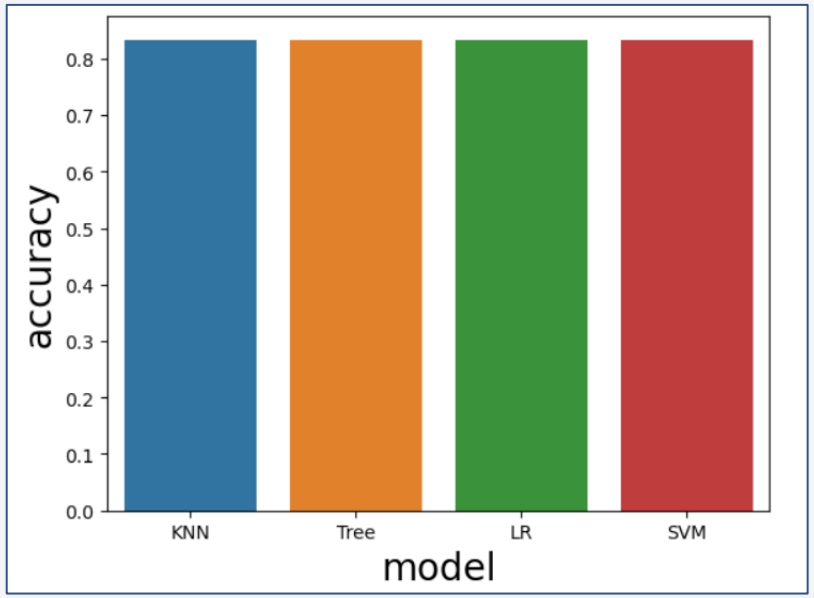
Section 5

Predictive Analysis (Classification)

Classification Accuracy

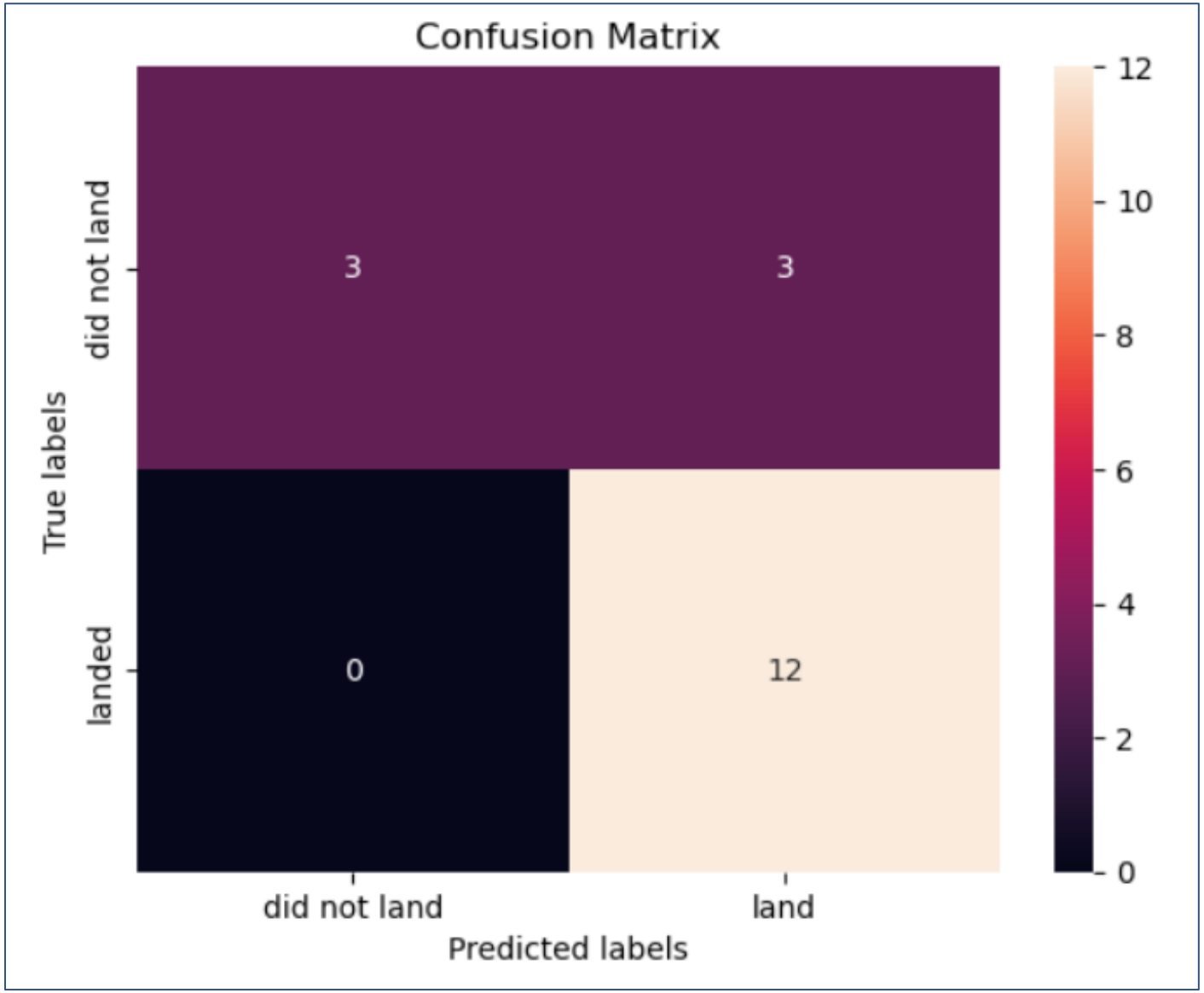
- Four ML models were tested: Logistic Regression (LR), SVM, Tree and KNN
- Besides accuracy, the Jaccard and the F1 scores were computed through Sklearn methods
- The table below shows all models performed equally well possibly due to the fact that the dataset is very small and the test dataset consisted of only 18 samples
- Of course, in the bar chart comparing accuracies of the different models, all bars have the same height

model	jaccard index	f1_score	accuracy
KNN	0.800000	0.888889	0.833333
Tree	0.800000	0.888889	0.833333
LR	0.800000	0.888889	0.833333
SVM	0.800000	0.888889	0.833333



Confusion Matrix

- The confusion matrix is the same for all models.
- As seen in the plot below, out of the 18 samples in the test dataset, there were 3 False Positives, no False Negatives, 3 True Negatives and 12 True Positives



Conclusions

- LR, SVM, Tree and KNN models performed equally well attaining high accuracy and F1 scores of 83 and 89%, respectively
- *Caveat:* the original dataset, and consequently the test dataset, is too small to permit a more robust conclusion about the models
- Despite some small fluctuations, landing success rates for Space X missions have increased over time, indicating that there has been considerable progress in rocket technology
- KSC LC-39A had the most successful launches among all sites
- Success rate is higher for missions with light payloads than for missions with heavy payloads
- The use of Weighted Success Rates (WSR) is more appropriate for assessing the impact of orbits on mission outcomes than the flat success rate, as it increases the level of statistical significance of the analysis. Using WSR, the four most successful orbits in decreasing order, are GTO, ISSV, LEO and PO

Appendix

- Examples of improvements in the presentation beyond the template:
  - Bar plot of the Weighted Success Rate [slide 20]
  - Joint plot of the lines representing the yearly success rate and the yearly total launches [slide 23]
  - Use of different performance scores (Jaccard and F1) to assess the ML models' performance [slide 43]
- Example of Innovative Insights:
  - Increase statistical significance of analysis of orbits influence on mission outcomes through the introduction of the concept of Weighted Success Rates
  - Joint analysis of yearly success rate and total launches to look for explanations on why the success rate fluctuated in some years
- Final consideration:
  - A more in-depth study is needed to understand the influence of different variables on the outcome of missions that gives greater weight to launch dates and thus somehow takes into account technological advances

Thank you!