IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Tullio BALLARINO
12/07/2023

Outline

Executive Summary

- *Summary of methodologies*
  - Data collection methodology
  - Perform data wrangling
  - Perform exploratory data analysis (EDA) using visualization and SQL
  - Perform interactive visual analytics using Folium and Plotly Dash
  - Perform predictive analysis using classification models
- *Summary of all results*
  - Exploratory data analysis results
  - Interactive analytics demo in screenshots
  - Predictive analysis results

Introduction *(1)*

Project background and context

- The era of commercial spaceflight is here
  - Companies are making space travel affordable
    - Virgin Galactic, Rocket Lab, Blue Origin, SpaceX, …
  - SpaceX
    - sends spacecraft to the International Space Station
    - launches Starlink, a satellite internet constellation providing satellite Internet access
  - Each SpaceX launch is characterized by different attributes:
    - *Mass*, *Launch site*, *Orbit*, ...
  - SpaceX rocket launches are *relatively inexpensive*:
    - **62** million dollars vs upwards of **165** million dollars (other providers costs)
  - SpaceX (Falcon 9) savings due to SpaceX reuse the first stage (but *sometimes the first stage does not land without damage…*)

Introduction *(2)*

Problems to find answers

1. Find the actual price of each launch (strongly reliant on *reuse of rocket first stage*)
2. Determine any relationship between the launch parameters and the outcome of the launch (i.e., the correct or incorrect landing of the first stage)
3. If such a relationship *(point 2)* exists and is evident, predict the outcome of future launches

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology
- Perform data wrangling
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models

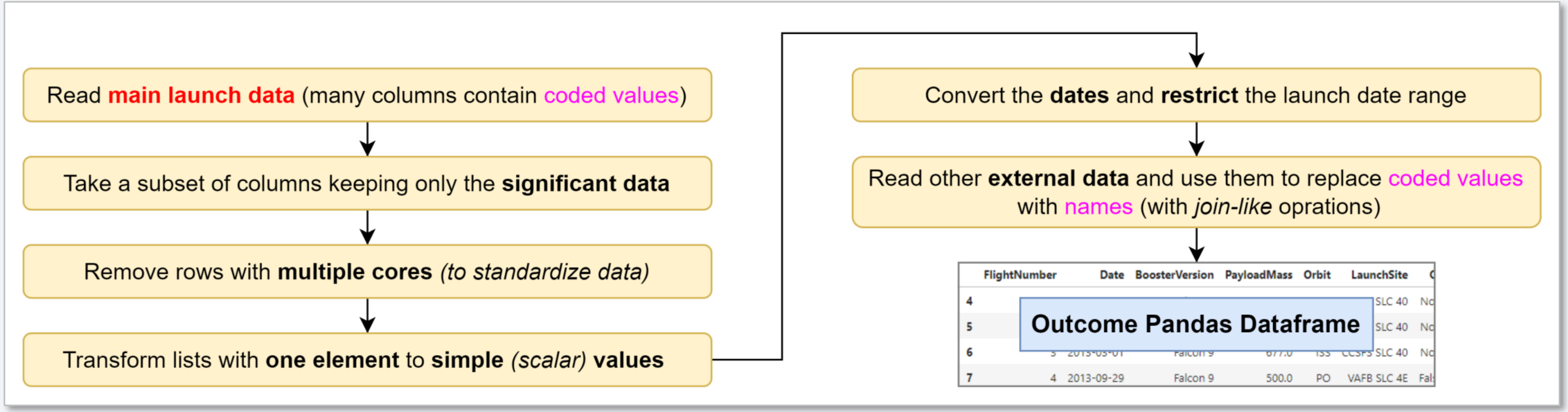
Data Collection

- *Data sets were collected / cleaned in 3 steps*

  1. Load launch raw data using SpaceX API and normalize/convert them to a manageable format
  2. Perform Web scraping to get more data directly from the web
  3. Perform exploratory Data Analysis and define Training Labels

Data Collection – SpaceX API

Data import and reorganisation using SpaceX API

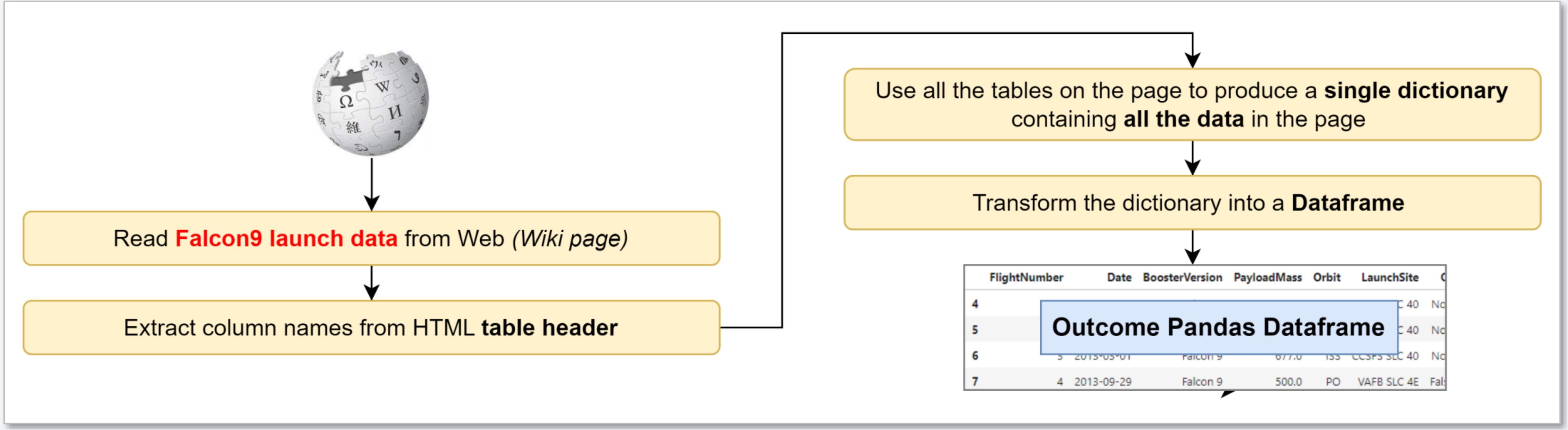


GitHub URL of the completed SpaceX API calls notebook:

https://github.com/tullioballarino/AppliedDataScienceCapstone/blob/main/1-1-jupyter-labs-spacex-data-collection-api.ipynb

Data Collection – Web scraping

Data import from the web using *BeautifulSoup*

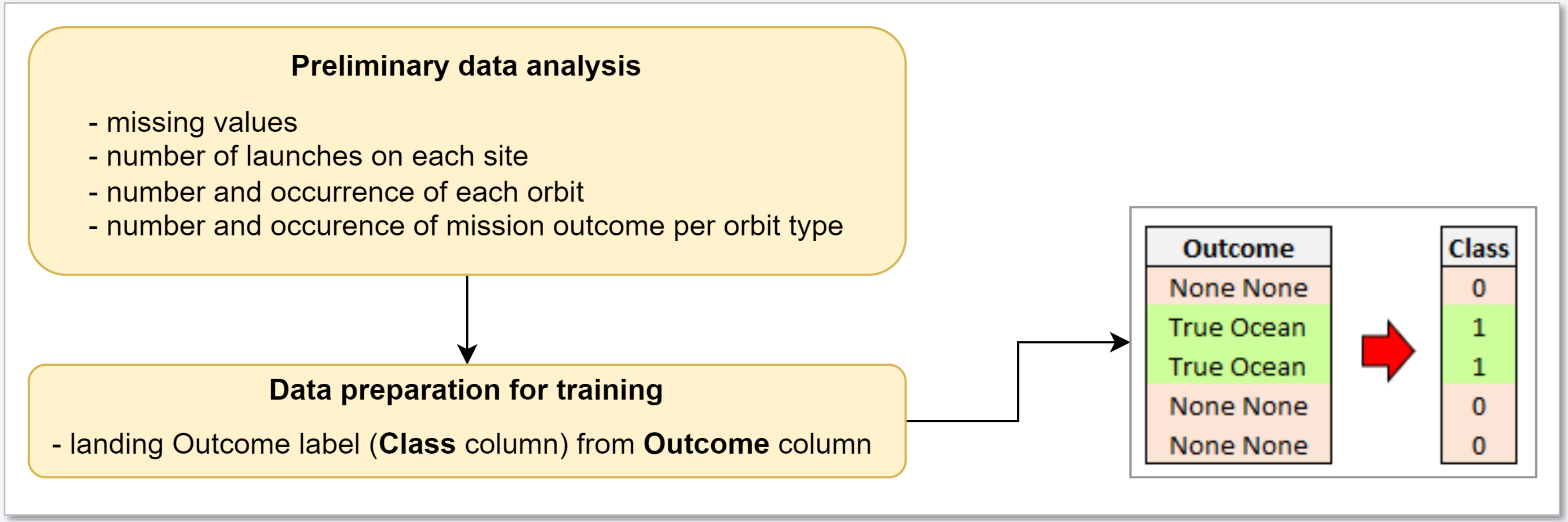


GitHub URL of the completed web scraping notebook:

https://github.com/tullioballarino/AppliedDataScienceCapstone/blob/main/1-2-jupyter-labs-webscraping.ipynb

Data Wrangling

Exploratory Data Analysis and definition of Training Labels



GitHub URL of the completed data wrangling notebook:

https://github.com/tullioballarino/AppliedDataScienceCapstone/blob/main/1-3-jupyter-labs-data_wrangling.ipynb

EDA with Data Visualization *(1)*

In order to identify possible relationships between the data of the launches, some graphs have been produced, listed in the following table together with the indicated results.

Plotted charts

x-axis	y-axis	Value	Chart kind	Question	Answer
Flight number	Pay load Mass	Class	Scatter	Flight Number and Pay load Mass affect the outcome?	**YES**: As the Flight number increases, the first stage is more likely to land successfully
Flight number	Launch site	Class	Scatter	Flight Number and Launch Site affect the outcome?	**YES**: different Launch sites have different success rates
Payload Mass	Launch site	Class	Scatter	Payload mass and Launch Site affect the outcome?	**NO**: no obvious relationship between Payload mass and Launch site
Orbit type	Success rate	Success rate	Bar	Any relationship between Orbit and Success rate?	**YES**: Orbit strongly influences Success rate

(continues on the next slide)

EDA with Data Visualization *(2)*

Plotted charts (follows)

x-axis	y-axis	Value	Chart kind	Question	Answer
Flight number	Orbit	Class	Scatter	Any relationship between Flight number and Orbit?	**PARTIAL**: in some Orbit Success is related on Flight number
Payload Mass	Orbit	Class	Scatter	Any relationship between Payload mass and Orbit?	**YES**: some Orbits (like *SSO*) in conjuncton with Payload Mass affect success rates
Year	Success rate	Success rate	Bar	Any relationship between Year and Success rate?	**YES**: the Success rate strongly increases as the Year increases

GitHub URL of the completed EDA with data visualization notebook:

https://github.com/tullioballarino/AppliedDataScienceCapstone/blob/main/2-2-jupyter-labs-eda-dataviz.ipynb

EDA with SQL *(1)*

SQL queries

1. Display the names of the unique launch sites in the space mission
```
SELECT DISTINCT Launch_Site FROM SPACEXTBL;
```
2. Display 5 records where launch sites begin with the string 'CCA'
```
SELECT * FROM SPACEXTBL
WHERE  Launch_Site LIKE 'CCA%' LIMIT 5;
```
3. Display the total payload mass carried by boosters launched by NASA (CRS)
```
SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL
WHERE  Customer = 'NASA (CRS)';
```
4. Display average payload mass carried by booster version F9 v1.1
```
SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTBL
WHERE  Booster_Version = 'F9 v1.1';
```
5. List the date when the first succesful landing outcome in ground pad was acheived
```
SELECT MIN(Date) FROM SPACEXTBL
WHERE  Landing_Outcome = 'Success (ground pad)';
```

(continues on the next slide)

EDA with SQL *(2)*

SQL queries (follows)

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

```
SELECT Booster_Version FROM SPACEXTBL
WHERE  (Landing_Outcome = 'Success (drone ship)') AND (PAYLOAD_MASS__KG_ > 4000)
       AND (PAYLOAD_MASS__KG_ < 6000);
```

7. List the total number of successful and failure mission outcomes

```
SELECT
(SELECT COUNT(*) FROM SPACEXTBL WHERE Mission_Outcome LIKE '%Success%')
   AS 'Successful outcomes',
(SELECT COUNT(*) FROM SPACEXTBL WHERE Mission_Outcome LIKE '%Failure%')
   AS 'Failure outcomes';
```

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

```
SELECT DISTINCT Booster_Version FROM SPACEXTBL
WHERE PAYLOAD_MASS__KG_ =
   (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL);
```

(continues on the next slide)

EDA with SQL *(3)*

SQL queries (follows)

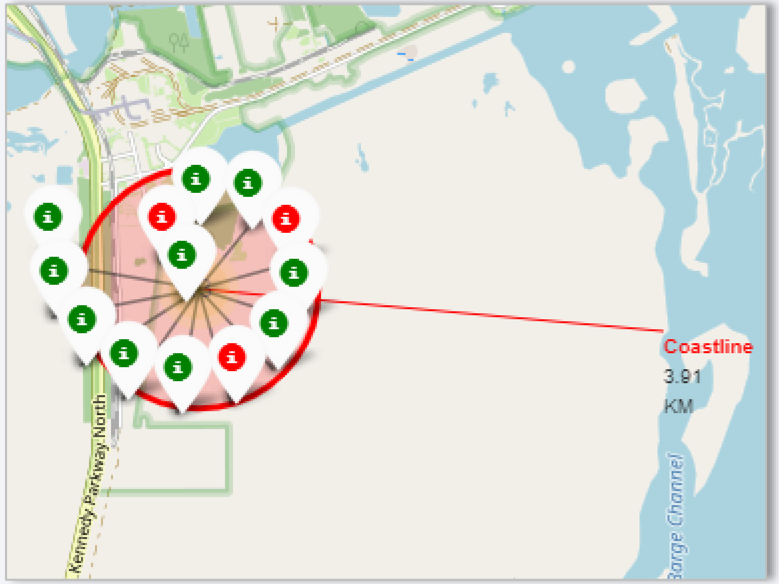
9. List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015
```
SELECT substr(Date, 4, 2) as Month, Landing_Outcome, Booster_Version, Launch_Site
FROM   SPACEXTBL
WHERE  (Landing_Outcome = 'Failure (drone ship)') AND (substr(Date,7,4) = '2015')
```
10. 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.
```
SELECT COUNT(*) AS 'Count', Landing_Outcome FROM SPACEXTBL
WHERE  ((substr(Date, 7, 4) || '-' || substr(Date, 4, 2) || '-' || substr(Date, 1, 2))
       BETWEEN '2010-06-04' AND '2017-03-20') AND (Landing_Outcome <> 'None')
GROUP  BY Landing_Outcome
ORDER  BY COUNT(*) DESC
```

GitHub URL of the completed EDA with SQL notebook:

https://github.com/tullioballarino/AppliedDataScienceCapstone/blob/main/2-1-jupyter-labs-eda-sql.ipynb

Build an Interactive Map with Folium

- *Classes of map objects added to a folium map*
  - Marker: to highlight a specific point on the map and/or associate it with a label
  - MarkerCluster: groups several contiguous markers under a single indicator, allowing them to be distinguished also if the zoom is inappropriate
  - Circle: to highlight a specific circular area of the map by associating it with a label
  - Line: to draw a line between 2 map points of given coordinates



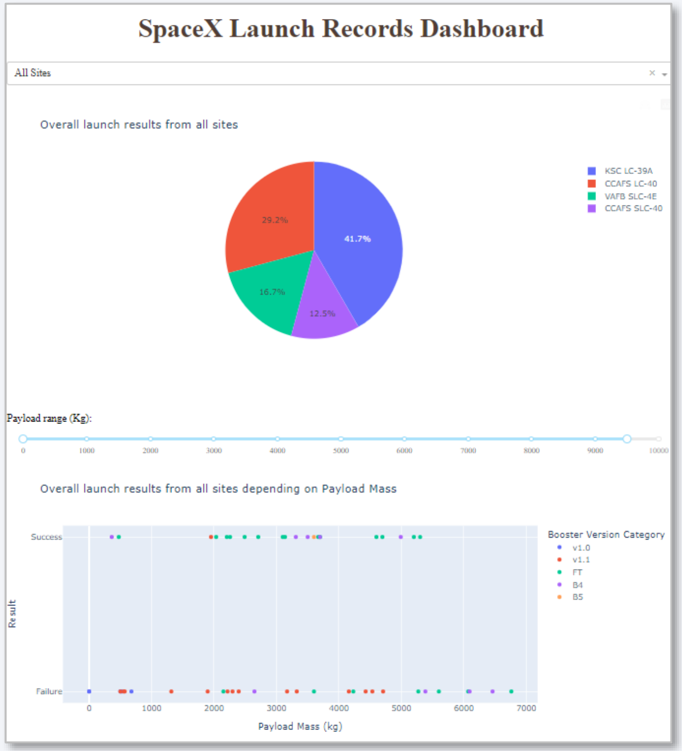
GitHub URL of the completed interactive map lab with Folium map:

https://github.com/tullioballarino/AppliedDataScienceCapstone/blob/main/3-1-jupyter-labs_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

- *Plots/graphs and interactions added to the dashboard*
  - Input Component: Launch Site Drop-down, to select ALL launch sites or just a specific site
  - Graph Component: Pie Chart graph, to show the success launches (*total count* of successes If *ALL* sites are selected, else *the only data for the selected site*)

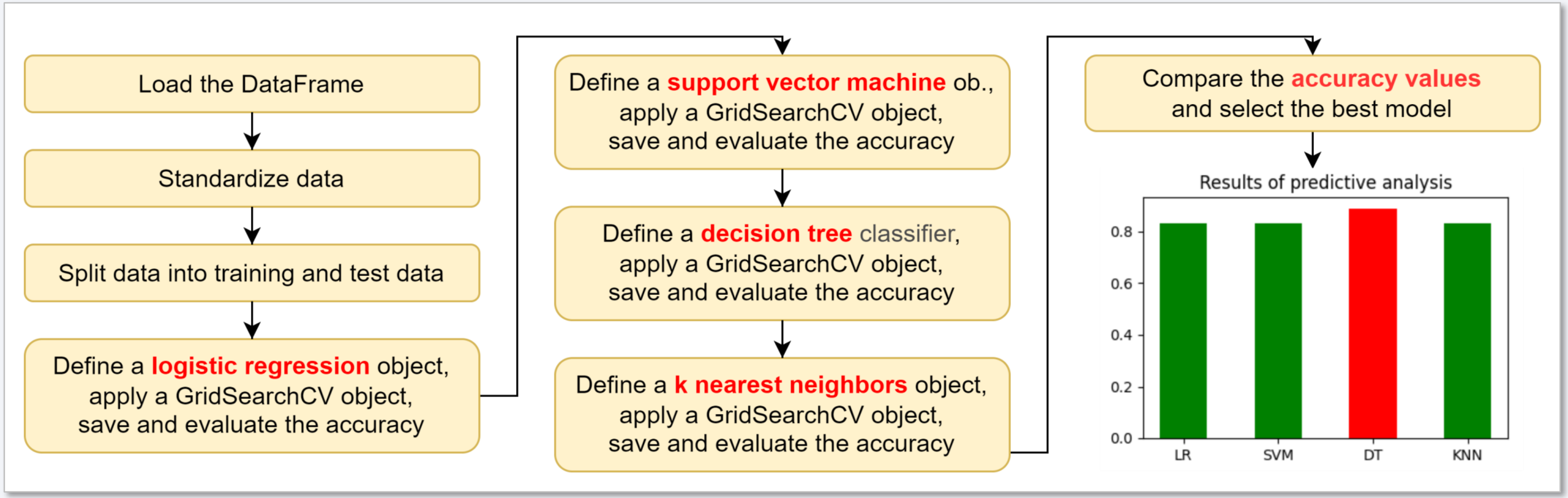
  - Input Component: Range Slider to define Payload range used in x-axis of following Scatter Plot
  - Graph Component: Scatter Plot with the payload on x-axis and the launch outcome on y-axis



GitHub URL of the completed EDA Plotly Dash lab:

https://github.com/tullioballarino/AppliedDataScienceCapstone/commit/3ca16306c1143715ba196661c92ebbc4b96279fb

Predictive Analysis (Classification)



GitHub URL of the completed predictive analysis lab:

https://github.com/tullioballarino/AppliedDataScienceCapstone/blob/main/4-1-jupyter-labs-MachineLearningPredictions.ipynb

Results

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

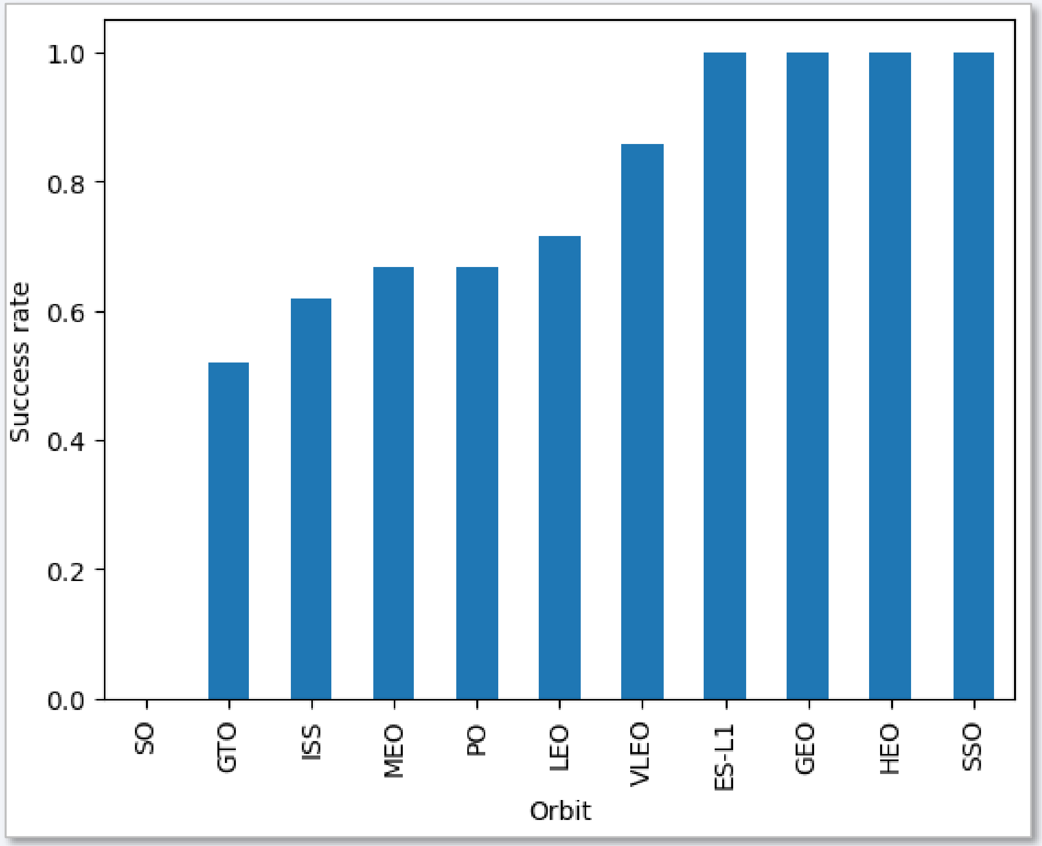
Exploratory data analysis results *(1)*

The data analysis highlighted several relationships between the various launch data, as summarized in the following table.

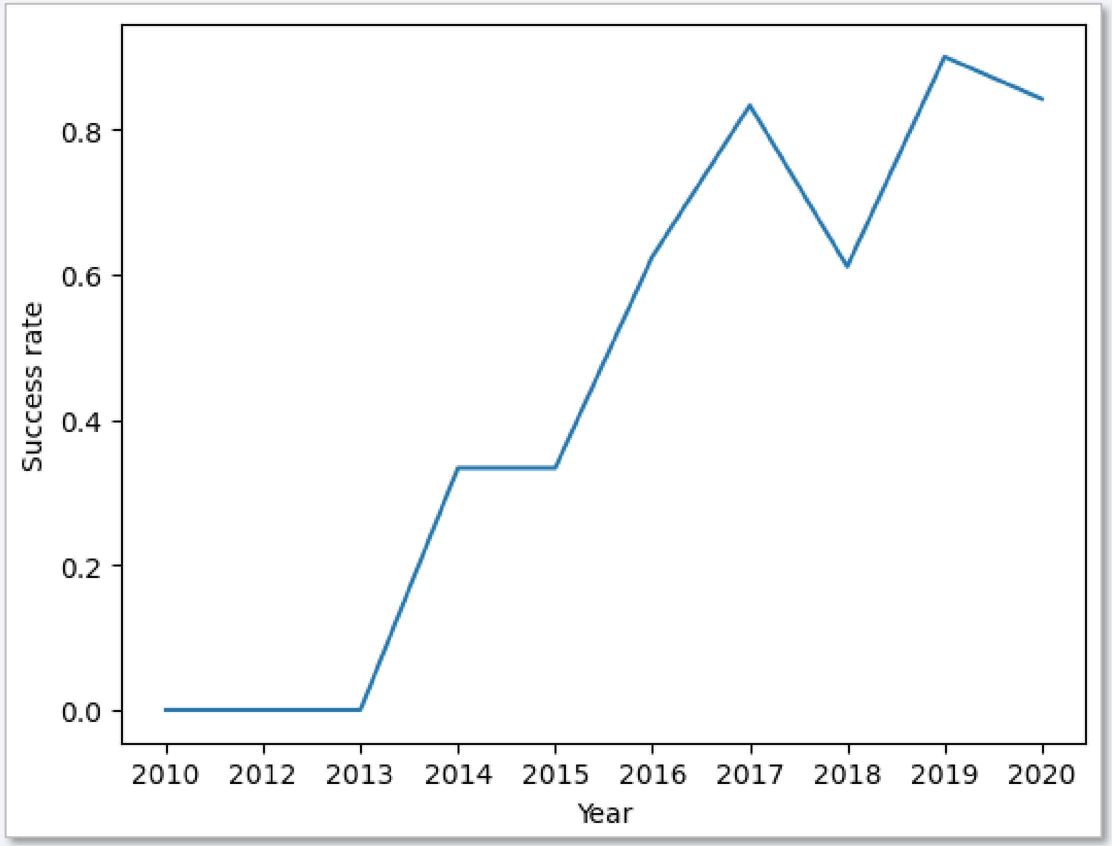
Value pair examined	Results
Flight number, Pay load mass	As the Flight number increases, the first stage is more likely to land successfully. The Payload mass is also important; it seems the more massive the payload, the less likely the first stage will return
Flight number, Launch site	Different Launch sites have different success rates. CCAFS LC-40 has a success rate of 60%, while KSC LC-39A and VAFB SLC 4E has a success rate of 77%
Payload mass, Launch site	There seems to be no obvious relationship between Payload mass and Launch site
Success rate, Orbit type	Orbit strongly influences Success rate (from 51% to 100% of successes depending on the Orbit travelled)
Flight Number, Orbit type	In some Orbits the Success rate appears related to the Flight Number; in other cases, there seems to be no relationship between Flight Number and Orbit
Payload mass, Orbit type	With heavy Payloads the successful landing or positive landing rate are more for Polar, LEO and ISS
Year of launch, Success rate	The Success rate since 2013 kept *increasing* till 2020

Exploratory data analysis results *(2)*

The following charts represent two interesting relationships between launch data pairs.

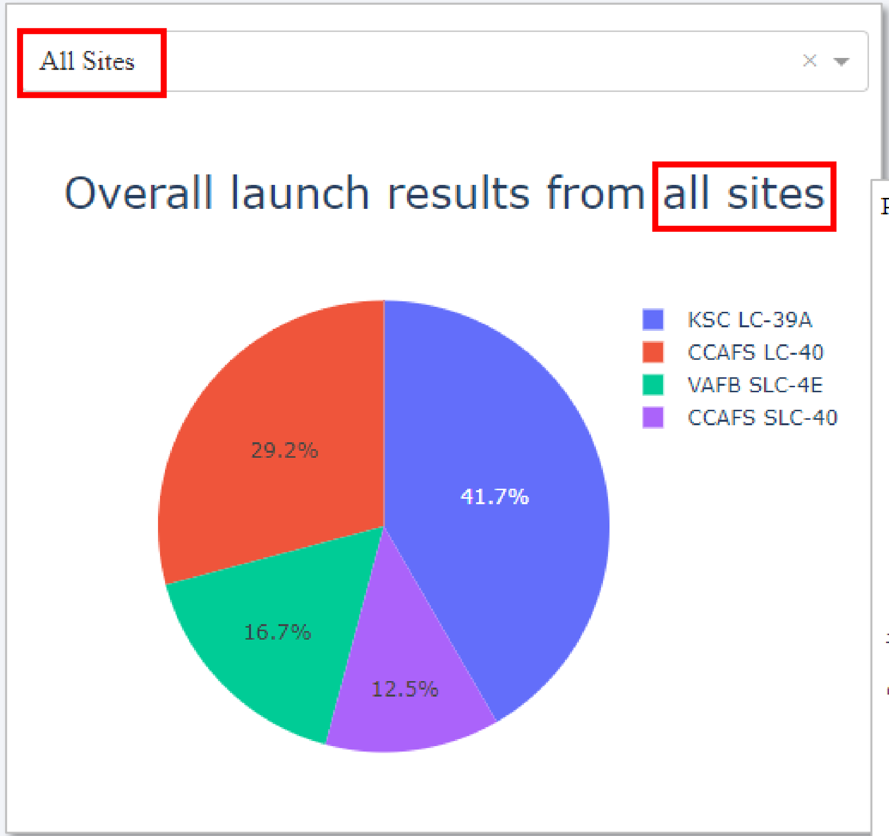


Orbit vs Success rate

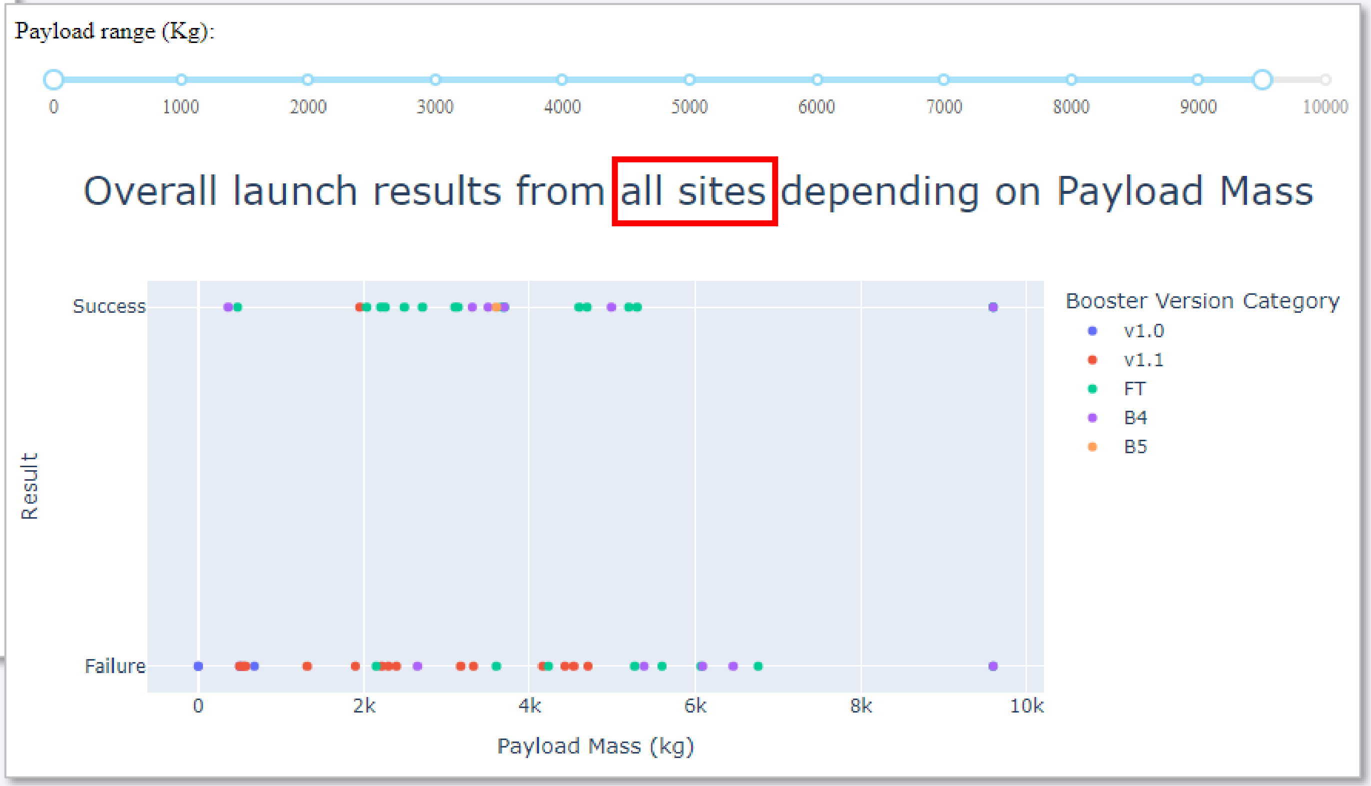


Launch Year vs Success rate

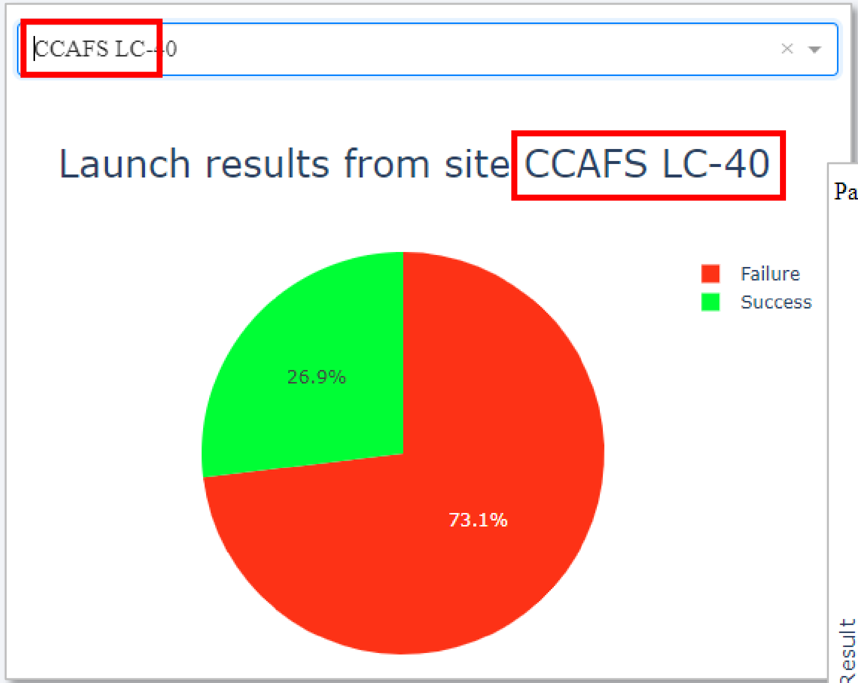
Interactive analytics demo in screenshots *(1)*



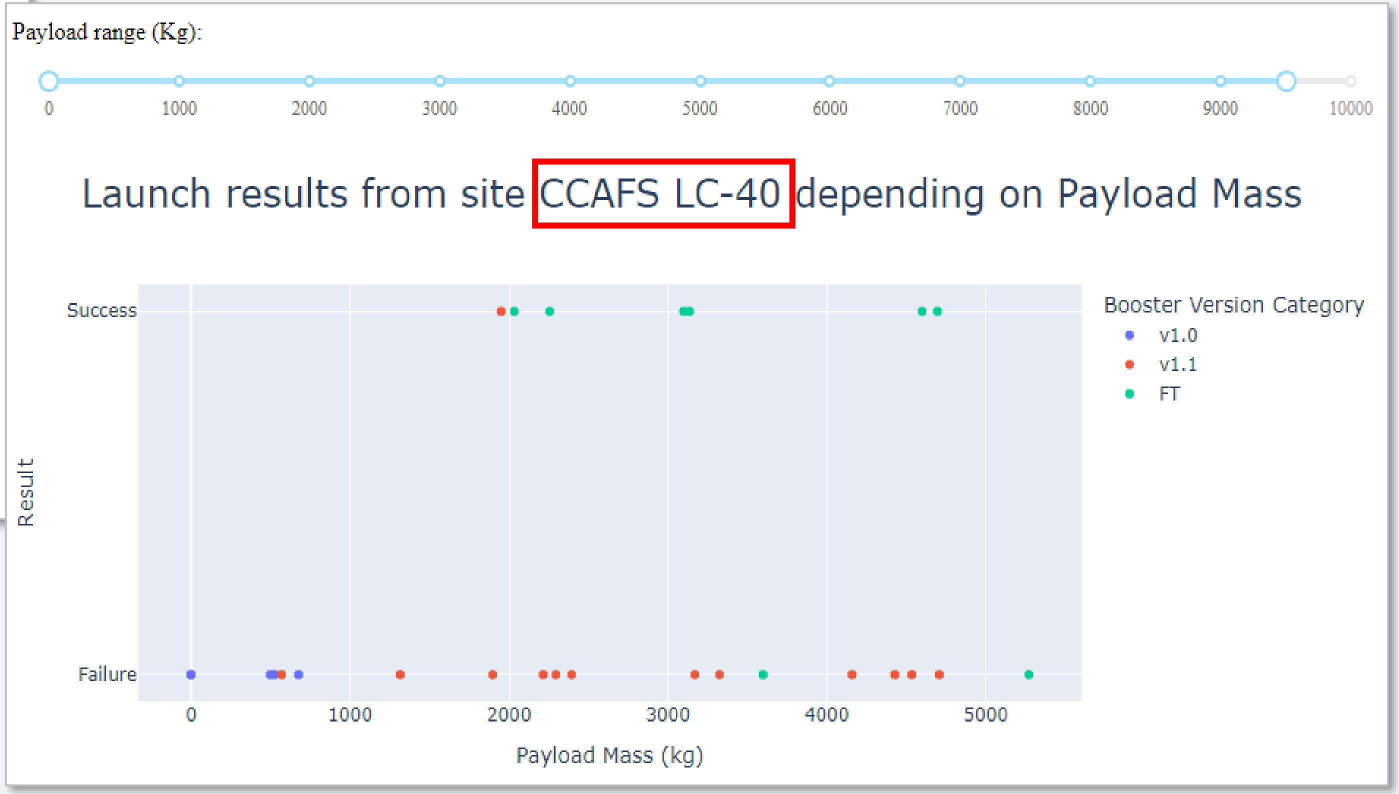
Overall launch results from *all sites*



Interactive analytics demo in screenshots *(2)*



Overall launch results from *a specific site*



Predictive analysis results

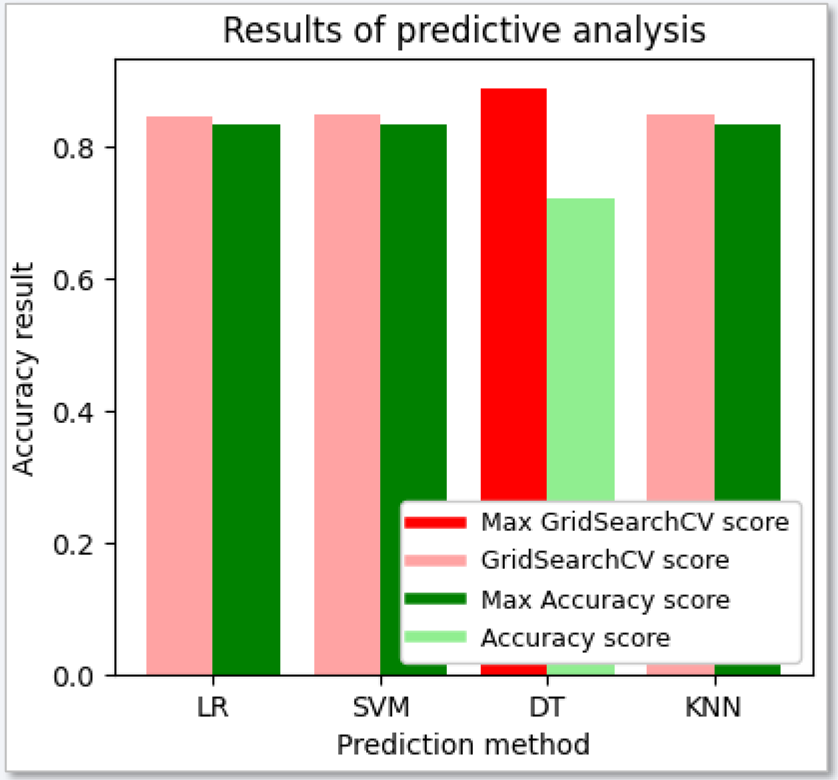
The predictive analysis was conducted using 4 different methods, as summarized in the table below; this table also shows the scores calculated with Accuracy score and GridSearchCV score methods.

From the results it can be deduced that the 4 methods, in terms of predictions, are substantially equivalent; it can also be assumed that the small differences in the scores depend more on the particular data analyzed than on the methods applied.

It must also be emphasized that the Accuracy score of the Decision tree varies according to the data returned from the data split function (in some cases it rises up to 8.7).

Overall list of methods and results

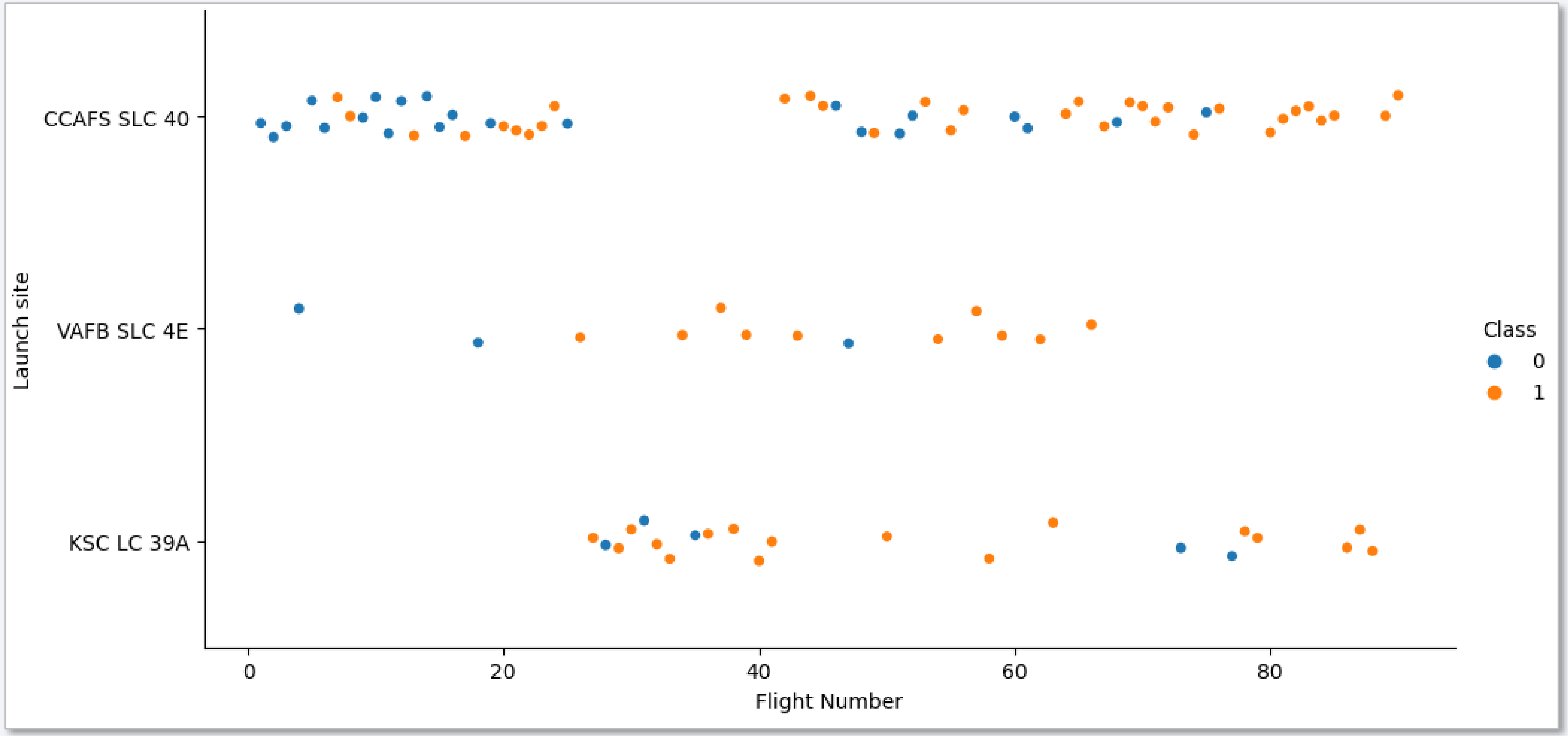
Method	CV best score	Accuracy score
Logistic regression	0.846	0.833
Support vector machine	0.848	0.833
Decision tree	0.889	0.722
K nearest neighbors	0.848	0.833



Section 2

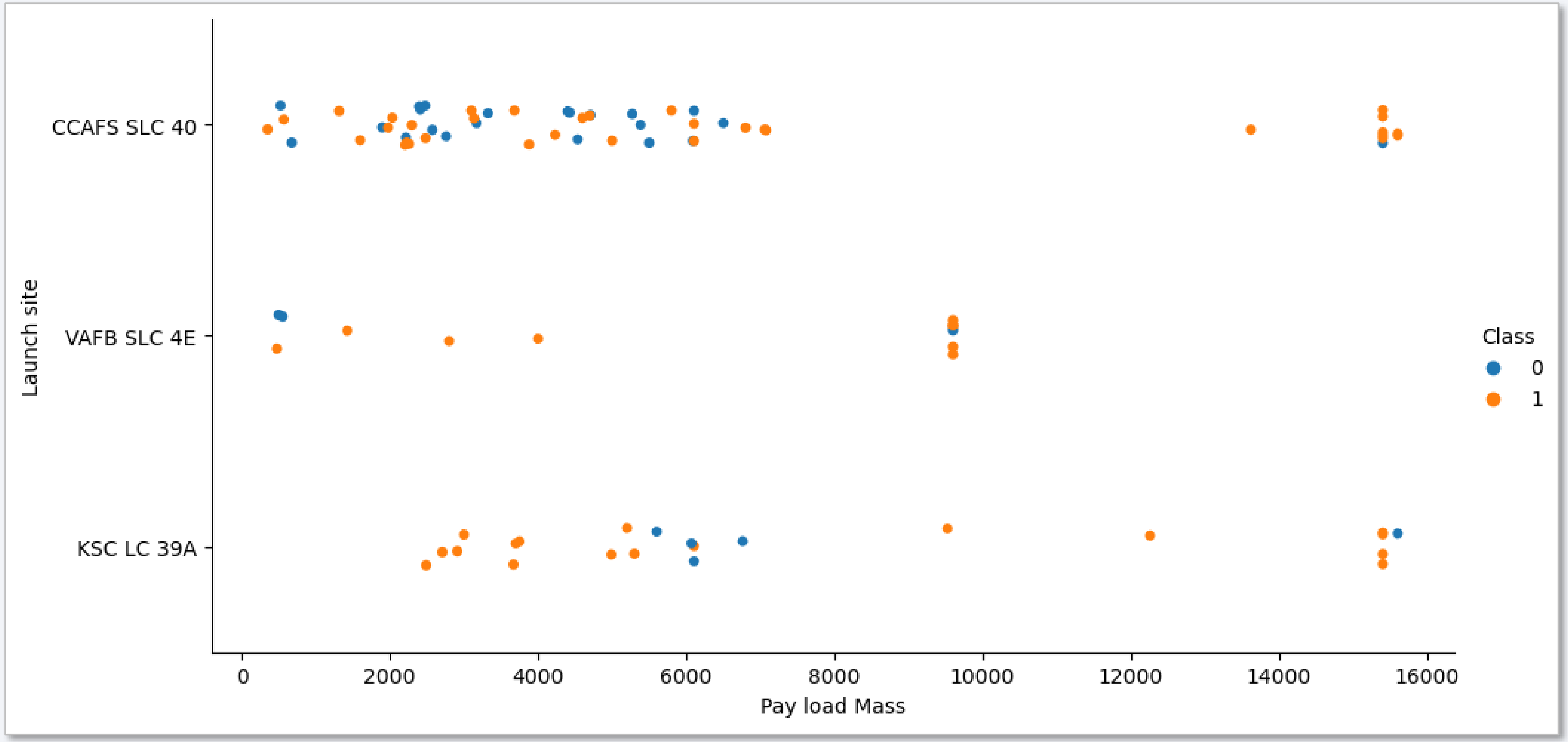
Insights drawn from EDA

Flight Number vs. Launch Site



Different Launch sites have different Success rates. CCAFS LC-40 has a success rate of 60%, while KSC LC-39A and VAFB SLC 4E has a success rate of 77%.

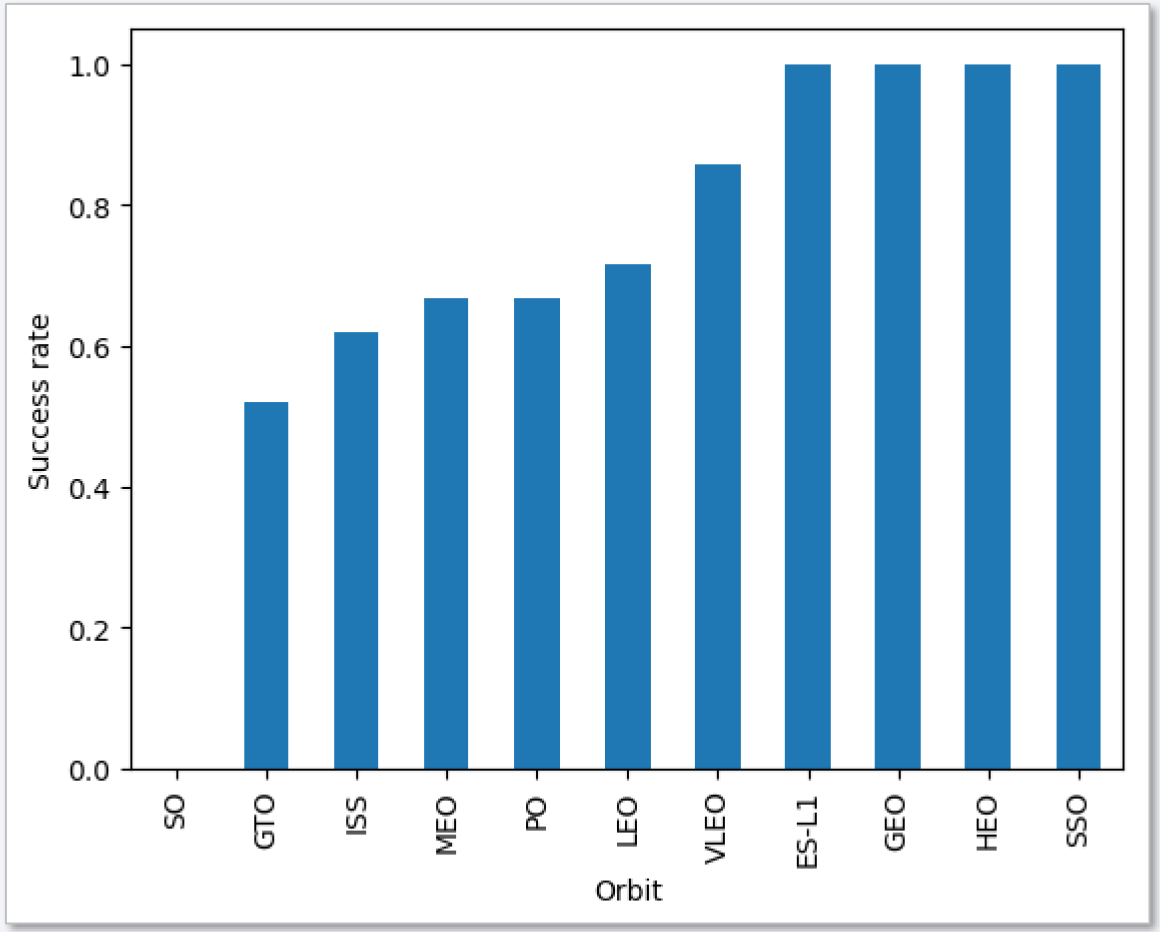
Payload vs. Launch Site



There seems to be no obvious relationship between Payload mass and Launch site.

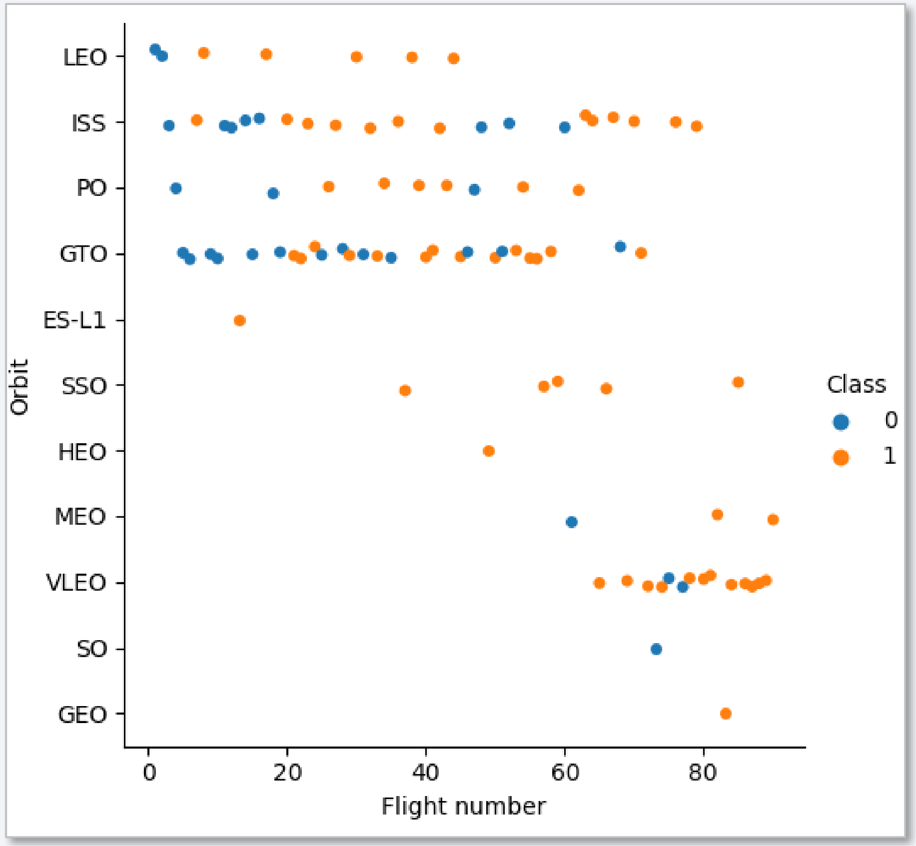
Note: for the VAFB-SLC Launch site there are no rockets launched for heavy Payload mass greater than 10,000.

Success Rate vs. Orbit Type



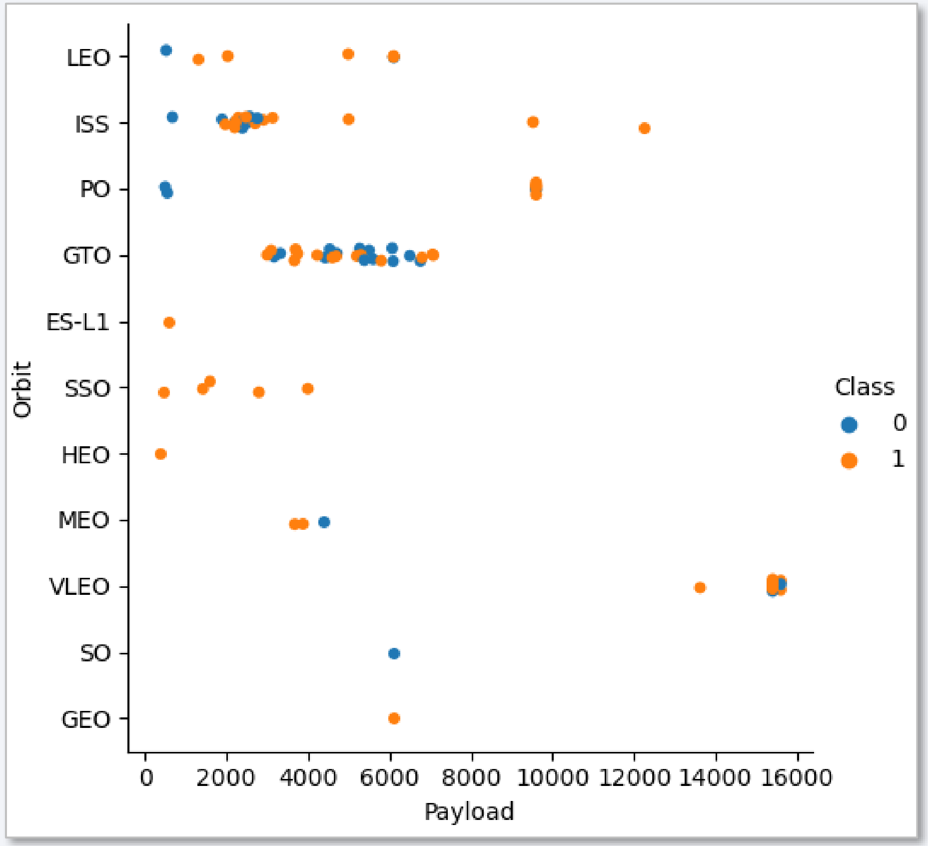
Orbit strongly influences Success rate (from 51% to 100% of successes depending on the orbit travelled).

Flight Number vs. Orbit Type



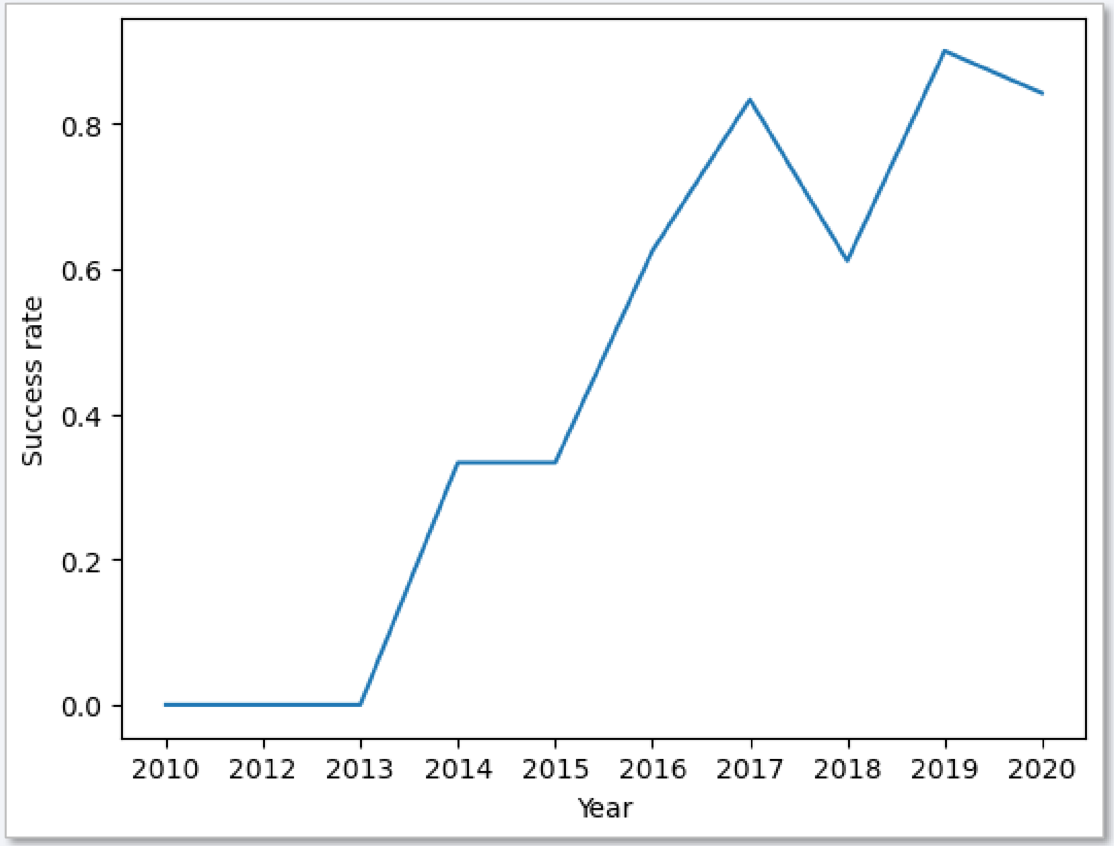
In some orbits the Success rate appears related to the number of flights; in other cases, there seems to be no relationship between flight number and orbit.

Payload vs. Orbit Type



With heavy payloads the successful landing or positive landing rate are more for Polar, LEO and ISS. For GTO we cannot distinguish this well as both positive landing rate and negative landing (unsuccessful mission) are both there here.

Launch Success Yearly Trend



The success rate since 2013 kept increasing till 2020.

1 - All Launch Site Names

- Find the names of the unique launch sites

```
%sql SELECT DISTINCT Launch_Site FROM SPACEXTBL;
```

```
 * sqlite:///my_data1.db
Done.
```

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

SELECT statement with DISTINCT keyword to display unique values

2 - Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with `CCA`

```
%sql SELECT * FROM SPACEXTBL WHERE Launch_Site LIKE 'CCA%' LIMIT 5;
```

```
 * sqlite:///my_data1.db
Done.
```

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit
06/04/2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0.0	LEO

SELECT statement with WHERE clause to select the rows and LIMIT clause to return only 5 records

3 - Total Payload Mass

- Calculate the total payload carried by boosters from NASA

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE Customer = 'NASA (CRS)'
```

```
 * sqlite:///my_data1.db
Done.
```

SUM(PAYLOAD_MASS__KG_)
45596.0

SELECT statement with WHERE clause to select the rows and the aggregate function SUM() to return the total payload

4 - Average Payload Mass by F9 v1.13232

- Calculate the average payload mass carried by booster version F9 v1.1

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE Booster_Version = 'F9 v1.1'
```

```
 * sqlite:///my_data1.db
Done.
```

AVG(PAYLOAD_MASS__KG_)
2928.4

SELECT statement with WHERE clause to select the rows and the aggregate function AVG() to return the average of payload mass

5 - First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad

```
%%sql SELECT MIN(Date) FROM SPACEXTBL
    WHERE Landing_Outcome = 'Success (ground pad)'
```

```
 * sqlite:///my_data1.db
Done.
```

MIN(Date)
01/08/2018

SELECT statement with WHERE clause to select the rows and the aggregate function MIN() to return the dates of the first successful landing outcome

6 - Successful Drone Ship Landing with Payload between 4000 and 6000

- List the names of boosters which have successfully landed on drone ship and had 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) AND (PAYLOAD_MASS__KG_ < 6000);
```

```
 * sqlite:///my_data1.db
Done.
```

Booster_Version
F9 FT B1022
F9 FT B1026
F9 FT B1021.2

SELECT statement with multiple WHERE clause to select the rows.

Note: the BETWEEN operator was not used as the required interval does not include extremes

7 - Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes

```
%%sql
SELECT
(SELECT COUNT(*) FROM SPACEXTBL
    WHERE Mission_Outcome LIKE '%Success%') AS 'Successful outcomes',
(SELECT COUNT(*) FROM SPACEXTBL
    WHERE Mission_Outcome LIKE '%Failure%') AS 'Failure outcomes';
```

```
 * sqlite:///my_data1.db
Done.
```

Successful outcomes	Failure outcomes
100	1

SELECT statement with two other nested SELECT statements, one for each required column

8 - Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass

```
%%sql
SELECT DISTINCT Booster_Version FROM SPACEXTBL
    WHERE PAYLOAD_MASS__KG_ =
        (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL);
```

```
 * sqlite:///my_data1.db
Done.
```

Booster_Version
F9 B5 B1048.4
F9 B5 B1049.4

SELECT statement with DISTINCT keyword to display unique values and including a nested SELECT statement to find the maximum payload mass

9 - 2015 Launch Records

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

```
%%sql
SELECT substr(Date, 4, 2) as Month, Landing_Outcome,
    Booster_Version, Launch_Site
    FROM SPACEXTBL
    WHERE (Landing_Outcome = 'Failure (drone ship)') AND (substr(Date,7,4) = '2015')
```

```
 * sqlite:///my_data1.db
Done.
```

Month	Landing_Outcome	Booster_Version	Launch_Site
10	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

SELECT statement with WHERE clause to select the rows.

Note: SQLite does not support month names, so month numbers were used instead of month names.

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

- 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

```
%%sql
SELECT COUNT(*) AS 'Count', Landing_Outcome FROM SPACEXTBL
    WHERE ((substr(Date, 7, 4) || '-' || substr(Date, 4, 2) || '-' || substr(Date, 1, 2))
    BETWEEN '2010-06-04' AND '2017-03-20') AND (Landing_Outcome <> 'None')
    GROUP BY Landing_Outcome
    ORDER BY COUNT(*) DESC
```

```
 * sqlite:///my_data1.db
Done.
```

Count	Landing_Outcome
10	No attempt
5	Success (ground pad)

SELECT statement with WHERE clause to select the rows, GROUP BY clause to group by Landing Outcome and ORDER BY clause to sort by Count of Landing Outcome.

Note: SQLite lacks many functions for converting dates, so the dates were converted using common string functions.

Section 3

Launch Sites Proximities Analysis

Folium Map: global map of all launch sites

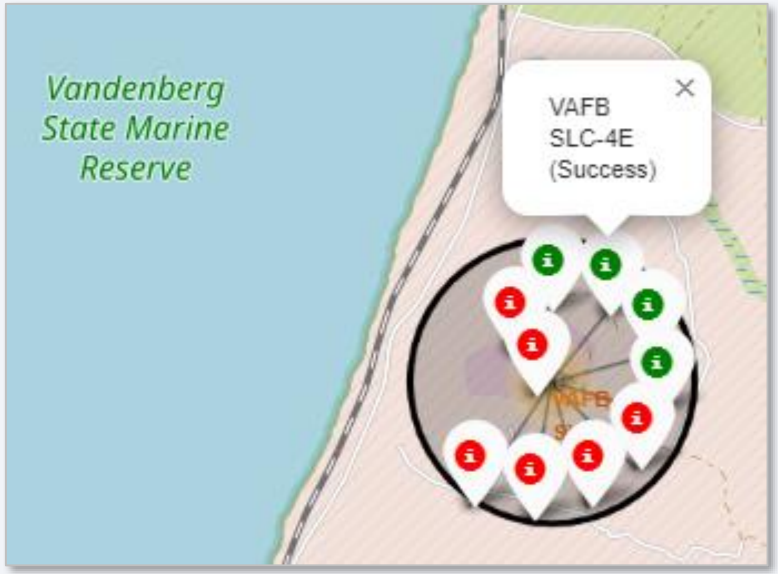
Zoom result

Zoom result

The following objects has been added on the map for each launch site:

- a marker with the site name
- a circle with a popup tooltip showing the site name

Folium Map: markers with site launch result

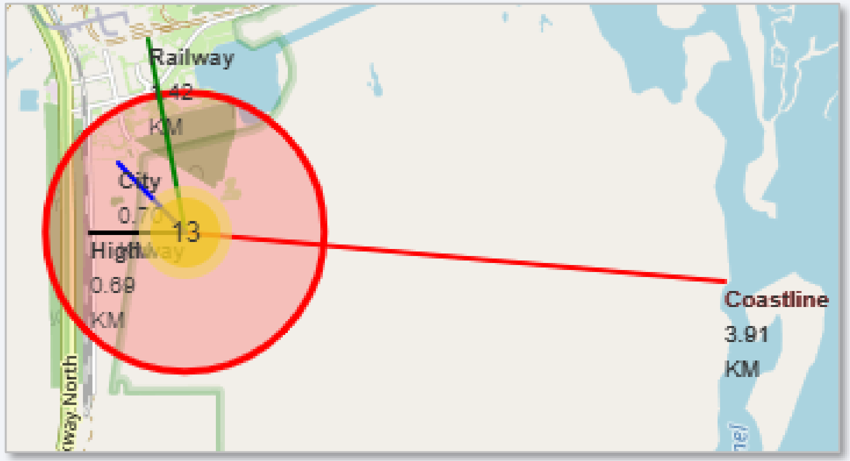


A cluster object was initially added to the map to collect colored icons indicating the launch results.

For each launch has been added on the map:

- a colored icon indicating the launch result (red = failure, green = success)
- a popup tooltip showing the site name and the launch result

Folium Map: markers with nearby significant points



For each significant point near a given launch site:

- the coordinates were found with the *MousePosition* tool
- the distance value from the neighbor site and the type of point have been drawn
- a colored line was drawn between the site and the significant point

df_map: KSC LC-39A site

Element	Lat	Lon	Color
Coastline	28.570	-80.607	red
Railway	28.586	-80.650	green
Highway	28.573	-80.654	black
City	28.578	-80.652	blue

```
launch_site = 'KSC LC-39A'
# for each row in df_map call 'distance_line' function
# to draw objects listed into the row
for index, row in df_map.iterrows():
    distance_line(launch_site, row['Lat'], row['Lon'], \
        row['Element'], row['Color'])
site_map
```

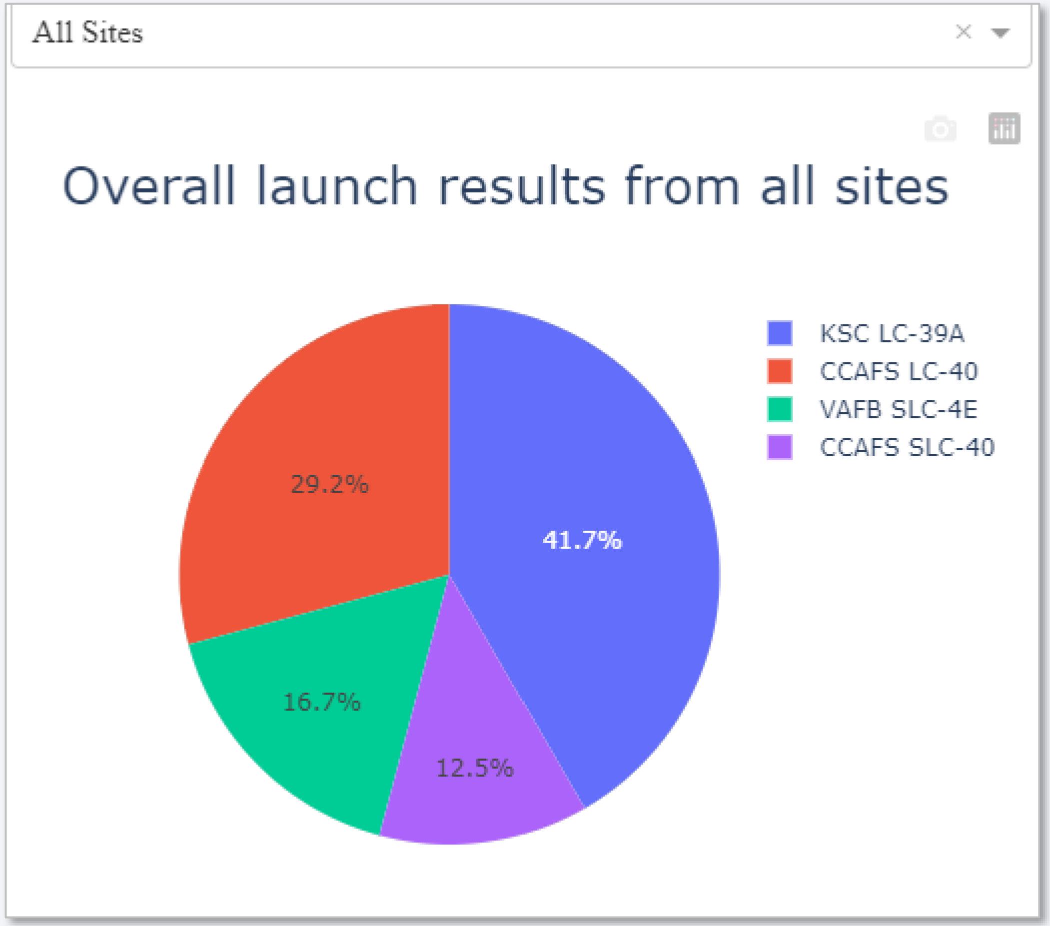
Section 4

Build a Dashboard with Plotly Dash

Launch success count for all sites

According to the assignment instructions, the chart shows, for each site, the percentage of successful launches of the site calculated with respect to the sum of successful launches of all the sites.

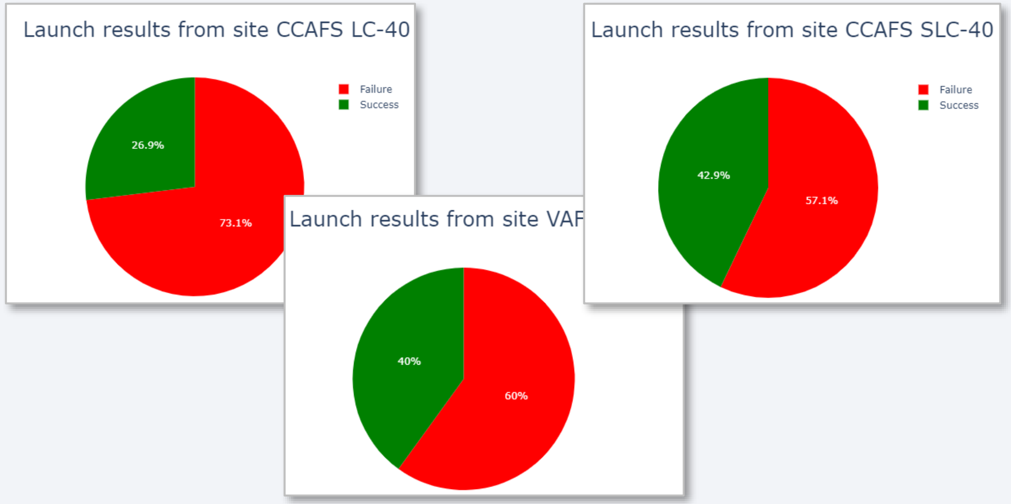
Warning: the data is not significant for calculating the percentage of successful launches of each site, as the value shown in the graph does not take into account the total number of launches of each site (successful or unsuccessful).
In other words, high percentages shown in the graph for a given site do not necessarily imply a high success rate for the site itself.

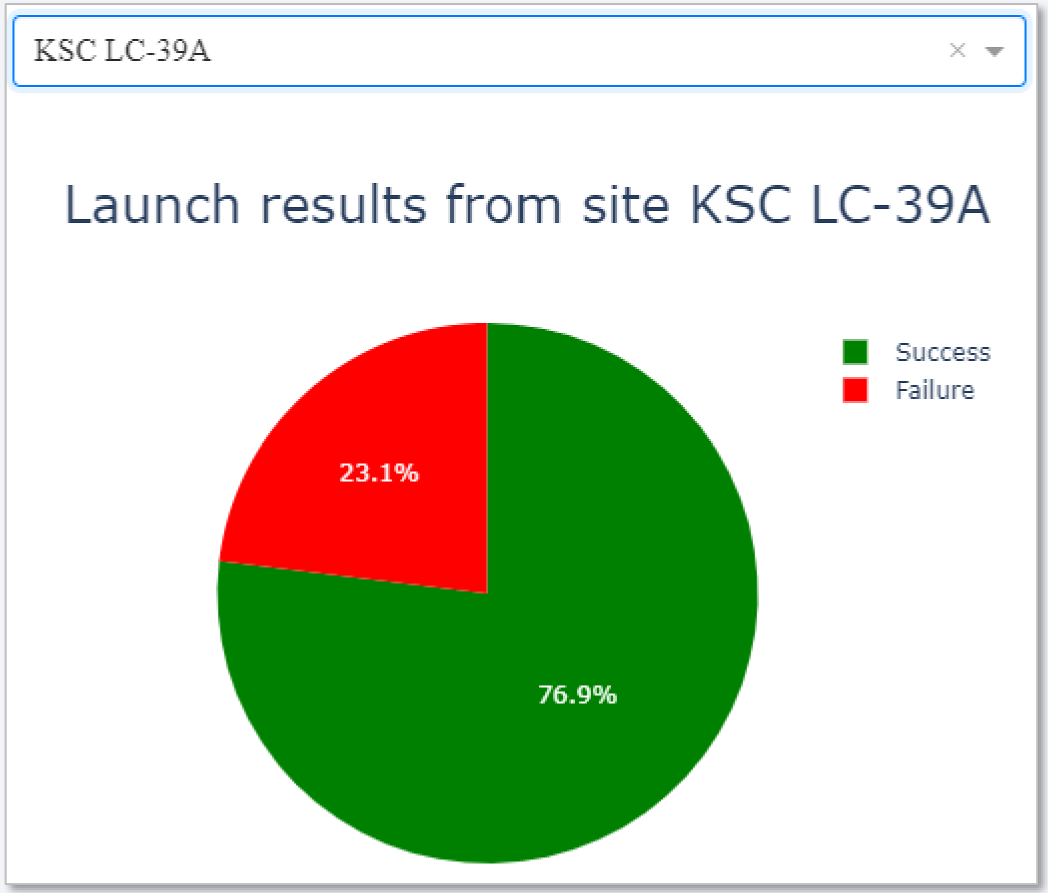


Launch success rate for KSC LC-39A site

The chart shows, for each site, the percentage of successful launches of the site KSC LC-39A calculated with respect to the total number of launches of the site.

It may be interesting to note how, for the other sites, the percentage of successes (green sector ■) is decidedly lower.





Payload vs. Launch Outcome relationship for all sites

It should be noted that the number of failures is, overall, higher than the number of successes,

However, examining the results of launches with the Booster FT (used only for a payload mass between 2k and 6k), it is found that the percentage of successes, in this case, exceeded the failures.

Section 5

Predictive Analysis (Classification)

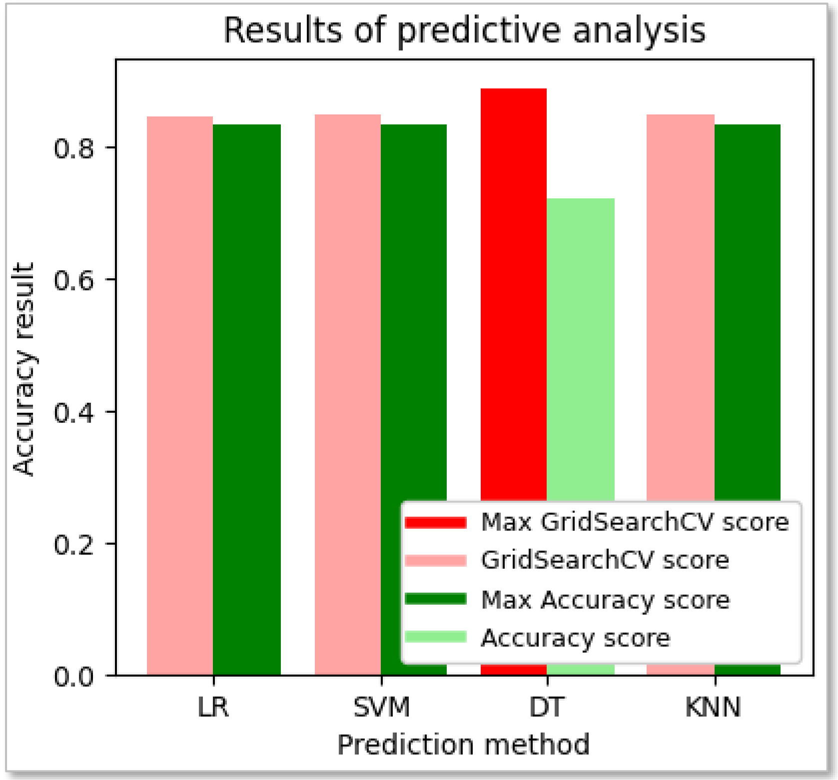
Classification Accuracy

For the final analysis of the results, it was considered significant to consider both Accuracy score and GridSearchCV score. The 4 classification methods, in terms of predictions, are substantially equivalent; it can also be assumed that the small differences in the scores depend more on the particular data analyzed than on the methods applied.

It must also be emphasized that the Accuracy score of the Decision tree varies according to the random data returned from the data split function (in some cases it rises up to 8.7).

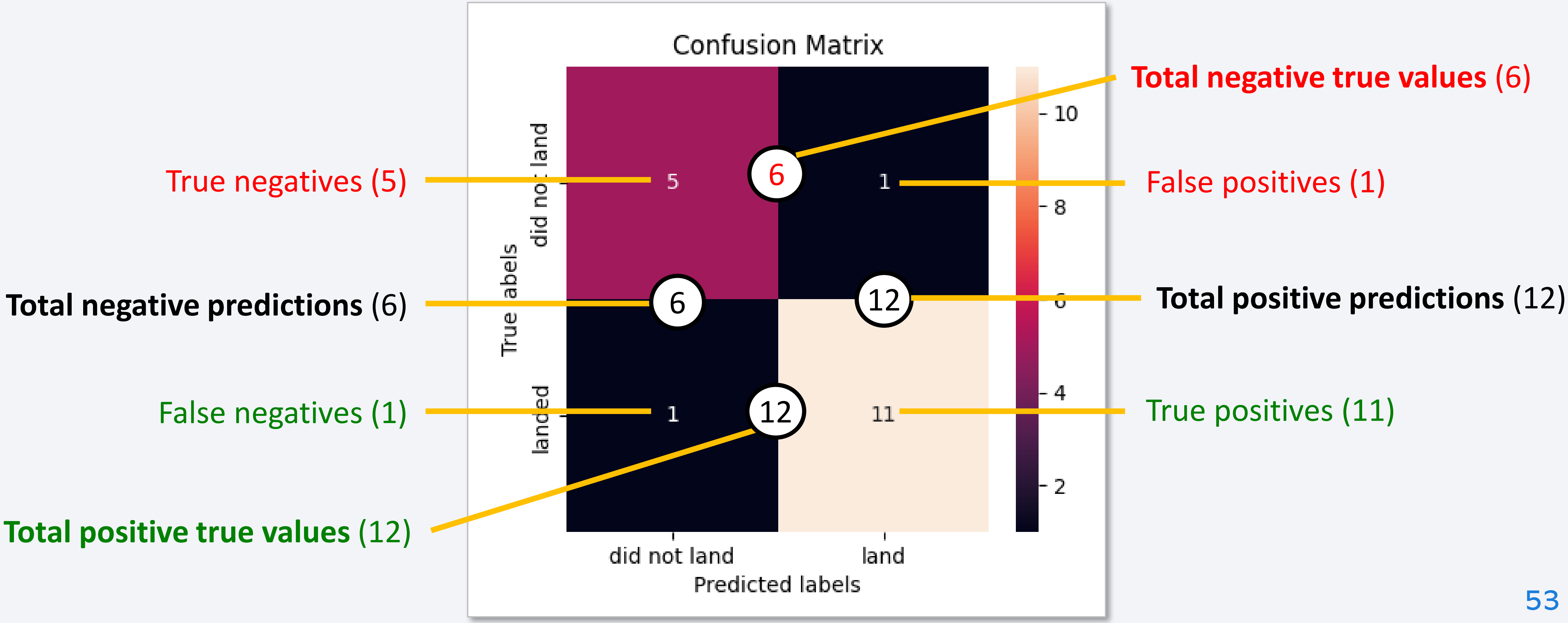
Overall list of methods and results

Method	CV best score	Accuracy score
Logistic regression	0.846	0.833
Support vector machine	0.848	0.833
Decision tree	0.889	0.722
K nearest neighbors	0.848	0.833



Confusion Matrix

Confusion matrix of the decision tree classifier



Conclusions

1. Examining the results of the application of the classification methods, it can be stated that the launch parameters considered in the calculation partially affect the recovery probability of the first stage of the rocket.
   In particular, the parameters that appear to have the most influence on the outcome are:
   - the launch site
   - the orbit
   - the year of the launch.
2. With regards to the possibility of intervening on these parameters, it can be noted that
   - the launch site can likely be selected based on the frequency of successes related to the site itself and seems therefore an element on which action can be taken;
   - the orbit depends on the type of mission requested by the customer and can hardly be changed;
   - the improvement linked to the progress of the launch year indicates that the experience of previous launches and the improvement of technologies over time are factors to be taken into great consideration as they substantially modify the outcome of the missions.

Personal considerations

Regarding predictive methods analysis, I would like to examine the following aspect: check to what extent these methods take into consideration overall values of groups of variables (let's say *clusters of variables*) rather than vectors of variables in which each component counts mainly *for itself* but not it is considered in combination with others.

In other words, it would be interesting to consider, in the analysis, pairs, triads, ..., tuples of variables exactly as if they were simple variables. To clarify this aspect, the following 2 examples may be useful, referring to the entire dataframe of the launches.

1. The pair of the two values
   [LaunchSite = 'KSC LC 39A', Orbit = 'ISS']
   is always associated with a Class value equal to 1, regardless of the value of all other variables.
2. Similarly, the pair
   [LaunchSite = 'VAFB SLC 4E', Orbit = 'SSO']
   also has the same characteristic.

It would therefore seem that, in some cases, the prediction of test values can be based precisely on the presence of a given *pattern* within the vector of independent variables rather than on the whole vector.

Of course, this consideration is only a starting point for further study, to be carried out also using the contents of the next modules of this course.

Appendix *(1)*

Example of python code used in conjunction with folium library to highlight significant points closest to launch sites:

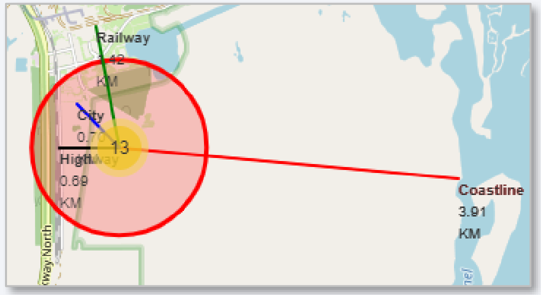
```
# Return the coordinates of a launch site given its name [launch_site]
def site_coordinates(launch_site):
    launch_site_row = launch_sites_df[launch_sites_df['Launch Site'] == launch_site]
    return launch_site_row['Lat'].values[0], launch_site_row['Long'].values[0]

# Calculate the distance between the launch site of given name [launch_site]
# and the point of coordinates [point_lat], [point_lon]
def site_distance(launch_site, point_lat, point_lon):
    launch_site_lat, launch_site_lon = site_coordinates(launch_site)
    return calculate_distance(launch_site_lat, launch_site_lon, point_lat, point_lon)
```

Appendix *(2)*

Example of python code used in conjunction with folium library to highlight significant points closest to launch sites:

```
# - Display the distance between the launch site named [launch_site]
#   and the point of coordinates [point_lat], [point_lon] labeled with [label];
# - Create a `folium.PolyLine` object using [point_lat], [point_lon] and [launch_site] coordinates
#   with color [linecolor]
def distance_line(launch_site, point_lat, point_lon, label, linecolor):
    # ---------
    # display the distance
    distance = site_distance(launch_site, point_lat, point_lon)
    distance_marker = folium.Marker(
            [point_lat, point_lon],
            icon=DivIcon(
                icon_size=(30,30),
                icon_anchor=(0,0),
                html='<div style="font-size: 14"><b>%s</b></div>' % label + "{:10.2f} KM".format(distance),
            )
        )
    site_map.add_child(distance_marker)
    # ---------
    # create a `folium.PolyLine` object
    launch_site_lat, launch_site_lon = site_coordinates(launch_site)
    coordinates = [[launch_site_lat, launch_site_lon], [point_lat, point_lon]]
    lines = folium.PolyLine(locations=coordinates, weight=2, color=linecolor)
    site_map.add_child(lines)
```



Thank you!