



IBM Developer
SKILLS NETWORK



APPLIED DATA SCIENCE CAPSTONE

ANAS BELAYDI

OUTLINE

1. EXECUTIVE SUMMARY
2. INTRODUCTION
3. METHODOLOGY
4. RESULTS
5. CONCLUSION
6. APPENDIX

1. EXECUTIVE SUMMARY

In this capstone project, we will predict if the SpaceX Falcon 9 first stage will land successfully using several machine learning classification algorithms. The main steps in this project include:

- Data collection, wrangling, and formatting
- Exploratory data analysis
- Interactive data visualization
- Machine learning prediction

In the visualization phase, graphs shows that some features of the rocket launches have strong correlation than others with the outcome of the launches success or failure.

For the prediction, It's concluded that **decision tree** may be the best machine learning algorithm to predict if the Falcon 9 first stage will land successfully.

2. INTRODUCTION

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

Most unsuccessful landings are planned. Sometimes, SpaceX will perform a controlled landing in the ocean.

The main question that we are trying to answer is :

For a given set of features about a Falcon 9 rocket launch, will the first stage of the rocket land successfully?

METHODOLOGY

3. METHODOLOGY

Executive Summary :

- **Data collection methodology:**

There are two source that we will use to collect data needed :

- **Source 1** : Make a get request to the SpaceX public API, to collect data
- **Source 2** : Performing web scraping to collect Falcon 9 historical launch records from a Wikipedia page titled `List of Falcon 9 and Falcon Heavy launches`

- **Perform data wrangling :**

- o Delete unneeded columns
- o Dealing with missing values
- o Feature Ingenieuring

- **Perform exploratory data analysis (EDA) using visualization and SQL**

- **Perform interactive visual analytics using Folium and Plotly Dash**

- **Perform predictive analysis using classification models**

Tuned models using GridSearchCV to get the best performance

Data Collection Overview

Data collection process involved a combination of two sources :

- ▶ Api requests from Space X public API
- ▶ web scraping data from a table in Space X's Wikipedia entry.

Space X API Data Columns	Wikipedia Webscraping Columns
FlightNumber , Date , BoosterVersion , PayloadMass , Orbit , LaunchSite , Outcome , Flight , GriFins , Reused , Legs , LandingPad , Block , ReusedCount , Serial , Longitude , Latitude	Flight No , Launch Site , PayloadMass , Orbit , Customer , Launch outcome , Version , Booster , Booster landing , Date , Time

Data Collection– SpaceX API

Steps followed to extract data using SpaceX API

STEP	EXPLANATION
1 Request (SpaceX APIs)	Send an HTTP request to the SpaceX API to retrieve data about launches. The API provides JSON data containing information like launch site, booster version, and payload data.
2 .JSON file + Lists(Launch Site, Booster Version, Payload Data)	Parse the JSON response and extract the relevant data, which may include lists for launch sites, booster versions, and payload details.
3 Json_normalize to DataFrame data from JSON	Convert the JSON data into a pandas DataFrame using the json_normalize function. This transforms the nested structure of the JSON data into a more structured DataFrame.
4 Dictionary relevant data	Select the specific columns or data points from the DataFrame that are relevant for your analysis.
5 Cast dictionary to a DataFrame	Convert the dictionary containing the relevant data into a new DataFrame. This ensures that the data is in a consistent format for further analysis.
6 Filter data to only include Falcon 9 launches	Apply filters to the DataFrame to select only the rows corresponding to Falcon 9 launches.
7 Replace missing PayloadMass values with mean	Handle missing values in the PayloadMass column by replacing them with the mean value of that column. This helps ensure data consistency and completeness.

[Link to the NoteBook on GitHub](#)

Data Collection– Web Scraping

Steps followed to extract data using Web Scraping

STEP	EXPLANATION	
1	Request Wikipedia HTML	Send an HTTP request to the SpaceX Wikipedia page to retrieve the HTML content.
2	BeautifulSoup HTML5lib Parser	Parse the HTML content using the BeautifulSoup library with the HTML5lib parser. This converts the HTML into a structured format that can be easily manipulated.
3	Find launch info HTML table	Use BeautifulSoup to locate the HTML table containing the launch information. This might involve searching for specific tags or attributes within the HTML structure.
4	Cast dictionary to DataFrame	Convert the extracted data into a pandas DataFrame. This provides a structured and efficient way to work with the data.
5	Iterate through table cells to extract data to dictionary	Iterate through the table cells to extract the desired data and store it in a dictionary. This dictionary will represent a single row of the DataFrame.
6	Create dictionary	Create a list of dictionaries, where each dictionary represents a row of data extracted from the table. This list can then be converted into a pandas DataFrame.

[Link to the NoteBook on GitHub](#)

Data wrangling

In this step, we'll create a new target variable by categorizing the 'outcome' values into two groups.

The first group of outcomes means that the landing was successful (with a value is 1), and the second group means that the landing failed (with a value 0)

While the dataset has eight distinct outcomes, we'll simplify this to a binary classification problem.

OUTCOME VALUE	CLASS
True ASDS True RTLS True Ocean	1
None None False ASDS False Ocean None ASDS False RTLS	0

EDA with Data Visualization

Exploratory Data Analysis performed on variables Flight Number, Payload Mass, Launch Site, Orbit, Class and year. Visualizations help to compare relationships between variables to decide if a relationship exists so that they could be used in training the machine learning model

PLOT TYPE	FEATURES	TITLE
Scatter plot	Flight number , PayloadMass ,Class	Flight Number vs Payload Mass
Scatter plot	Launch site, Flight number, Class	Relationship between launch sites and their Flight number
Scatter plot	Payload Mass, Launch site, Class	Relationship between launch sites and their payload Mass
bar plot	Orbit, Success rate (calculated)	Success rate by Orbit
Scatter plot	Orbit type, Flight Number, Class	Relationship between Flight Number and orbit type
Scatter plot	PayloadMass, orbit type, Class	Relationship between Payload Mass and orbit type
Lineplot	Years, Success rate(calculated)	Success rate over years

[The link to the Notebook on GitHub](#)

EDA with SQL

Performed SQL queries:

- Displaying the names of the unique launch sites in the space mission
- Displaying 5 records where launch sites begin with the string 'CCA'
- Displaying the total payload mass carried by boosters launched by NASA (CRS)
- Displaying average payload mass carried by booster version F9 v1.1
- Listing the date when the first successful landing outcome in ground pad was achieved
- Listing the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
- Listing the total number of successful and failure mission outcomes
- Listing the names of the booster versions which have carried the maximum payload mass
- Listing the failed landing outcomes in drone ship, their booster versions and launch site names for the months in year 2015
- Ranking the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20 in descending ord

Build an Interactive Map with Folium

Markers of all Launch Sites

- Added Marker with Circle, Popup Label and Text Label of NASA Johnson Space Center using its latitude and longitude coordinates as a start location.
- Added Markers with Circle, Popup Label and Text Label of all Launch Sites using their latitude and longitude coordinates to show their geographical locations and proximity to Equator and coasts.

Coloured Markers of the launch outcomes for each Launch Site

Added coloured Markers of success (Green) and failed (Red) launches using Marker Cluster to identify which launch sites have relatively high success rates.

Distances between a Launch Site to its proximities

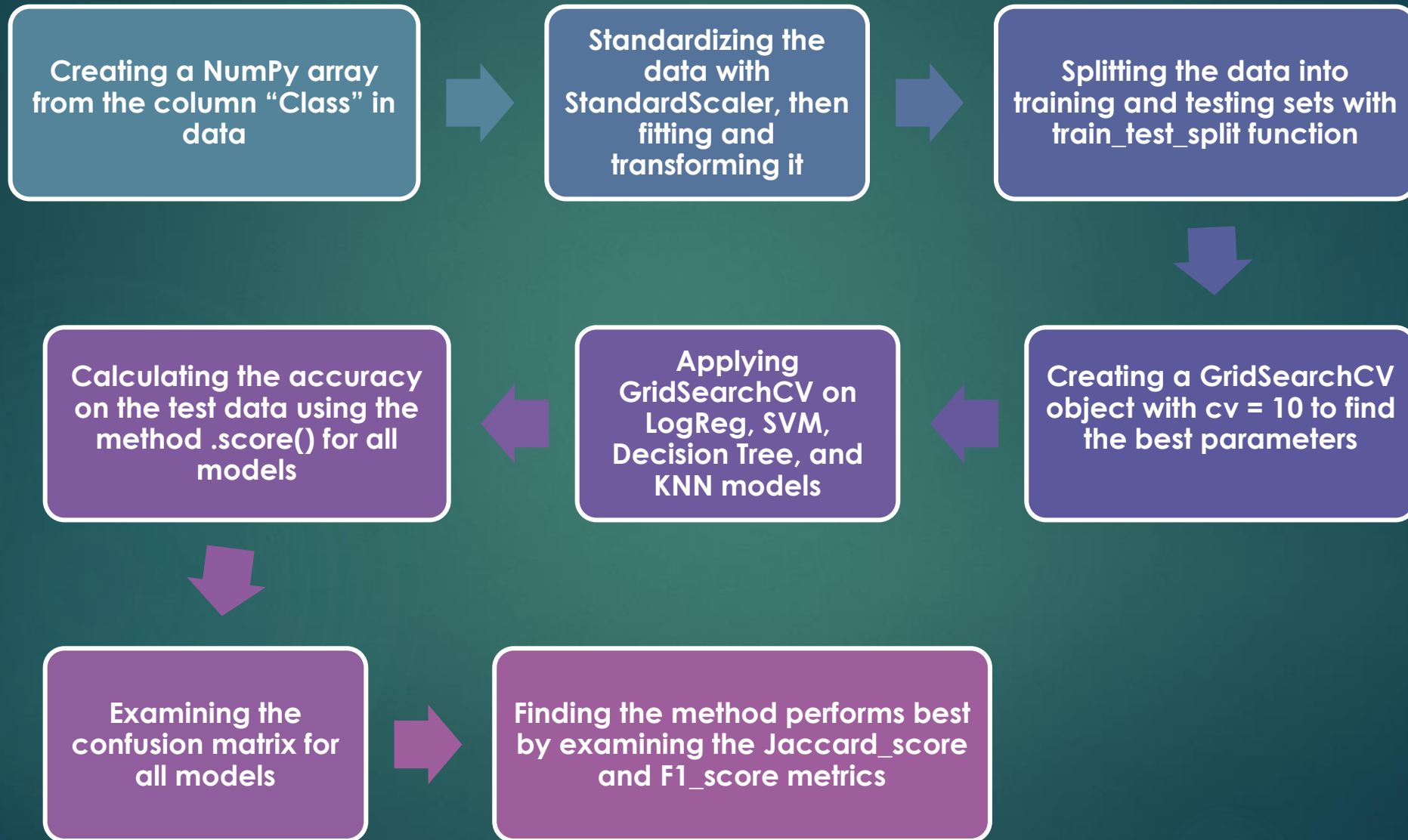
- Added coloured Lines to show distances between the Launch Site KSC LC-39A (as an example) and its proximities like Railway, Highway, Coastline and Closest City.

Build a Dashboard with Plotly Dash

Elements used in the Dashboard :

ELEMENT	EXPLANATION
Launch Sites Dropdown List:	Enable Launch Site selection
Pie Chart showing Success Launches (All Sites/Certain Site	Added a pie chart to show the total successful launches count for all sites and the Success vs. Failed counts for the site, if a specific Launch Site was selected.
Slider of Payload Mass Range:	Added a slider to select Payload range.
Scatter Chart of Payload Mass vs. Success Rate for the different Booster Versions	Added a scatter chart to show the correlation between Payload and Launch Success.

Predictive analysis (Classification)



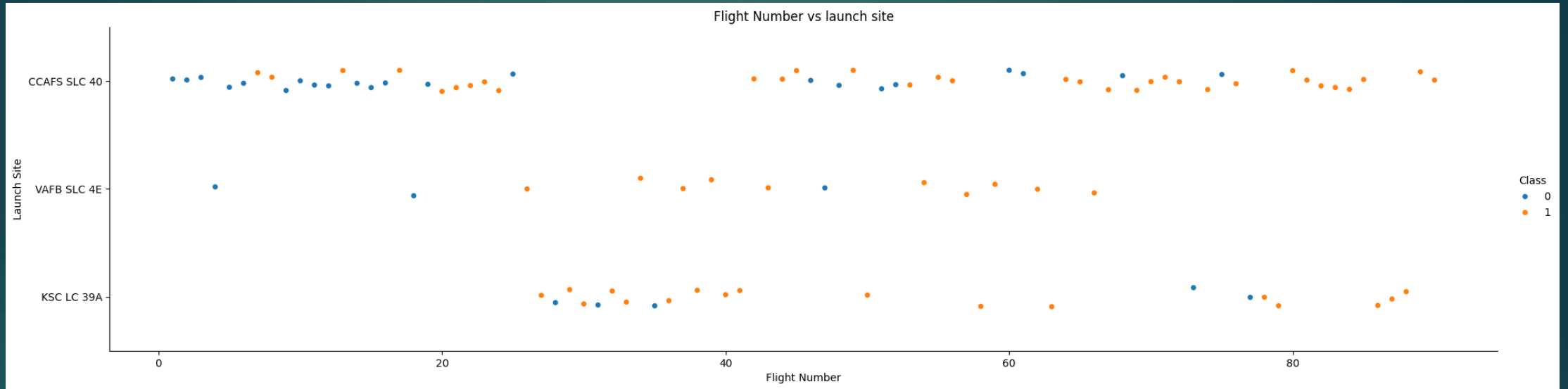
Results

- ▶ Exploratory data analysis results
- ▶ Interactive analytics demo on screenshots
- ▶ Predictive analysis results

EDA with Visualization

EDA with Data Visualization

The relationship between FlightNumber and launch site to define the success of landing



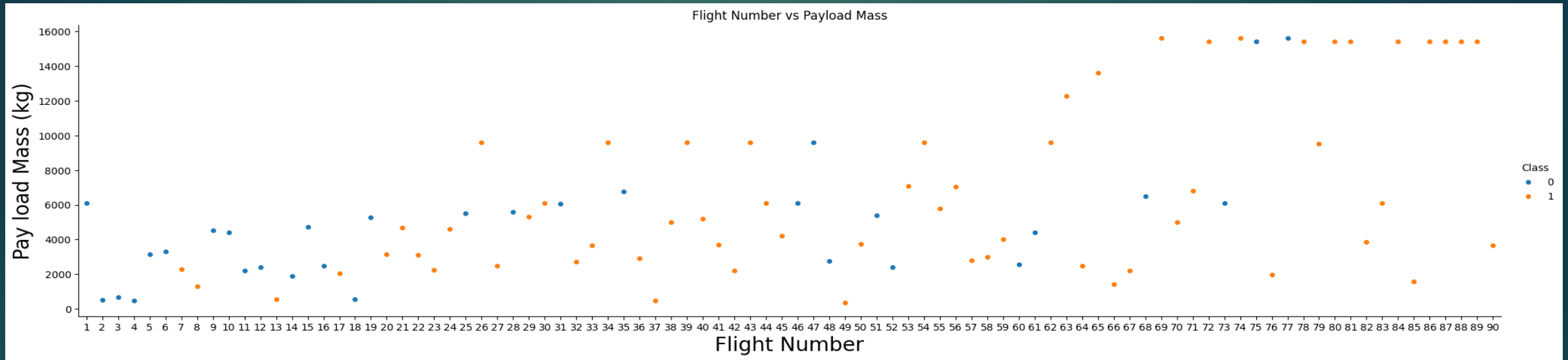
Launch Site Frequency: "CCAFS SLC 40" is the most frequently used launch site, followed by "KSC LC 39A" and "VAFB SLC 4E".

Launch Success: There seems to be a general trend where later flight numbers from "KSC LC 39A" and "VAFB SLC 4E" have a higher success rate (indicated by the blue dots). However, this trend is less clear for "CCAFS SLC 40".

Launch Site Variation: The spread of data points for "CCAFS SLC 40" suggests more variation in launch outcomes compared to the other two sites.

EDA with Data Visualization

The relationship between FlightNumber and PayloadMass to define the success of landing



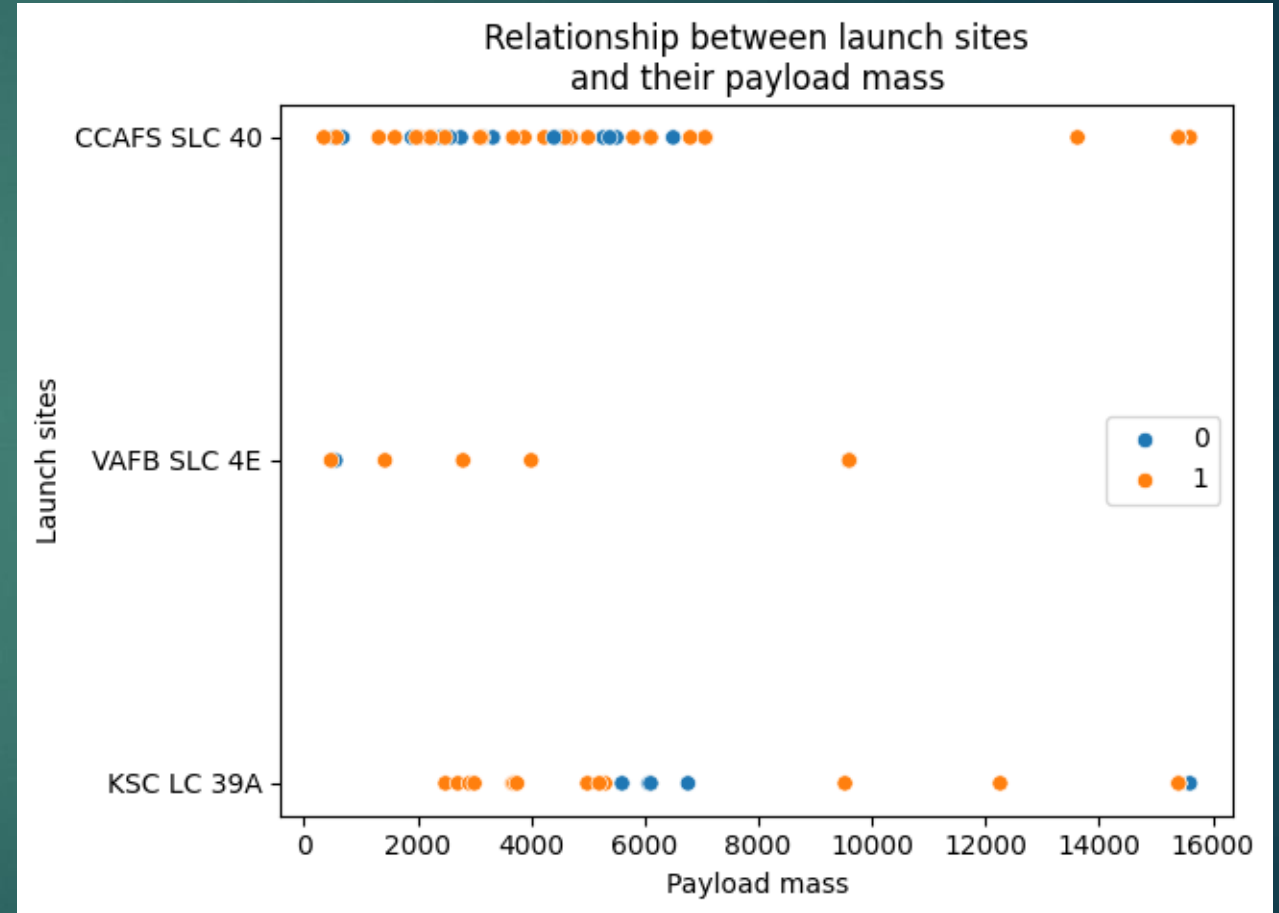
By overlaying the launch outcome on the plot of FlightNumber vs. PayloadMass, a pattern emerges. As the FlightNumber increases, the likelihood of the first stage landing successfully also increases. Conversely, as the PayloadMass increases, the likelihood of the first stage returning successfully decreases.

This suggests that both the number of launch attempts and the mass of the payload play significant roles in determining the success of the first stage's return.

EDA with Data Visualization

The relationship between Payload and Launch Site

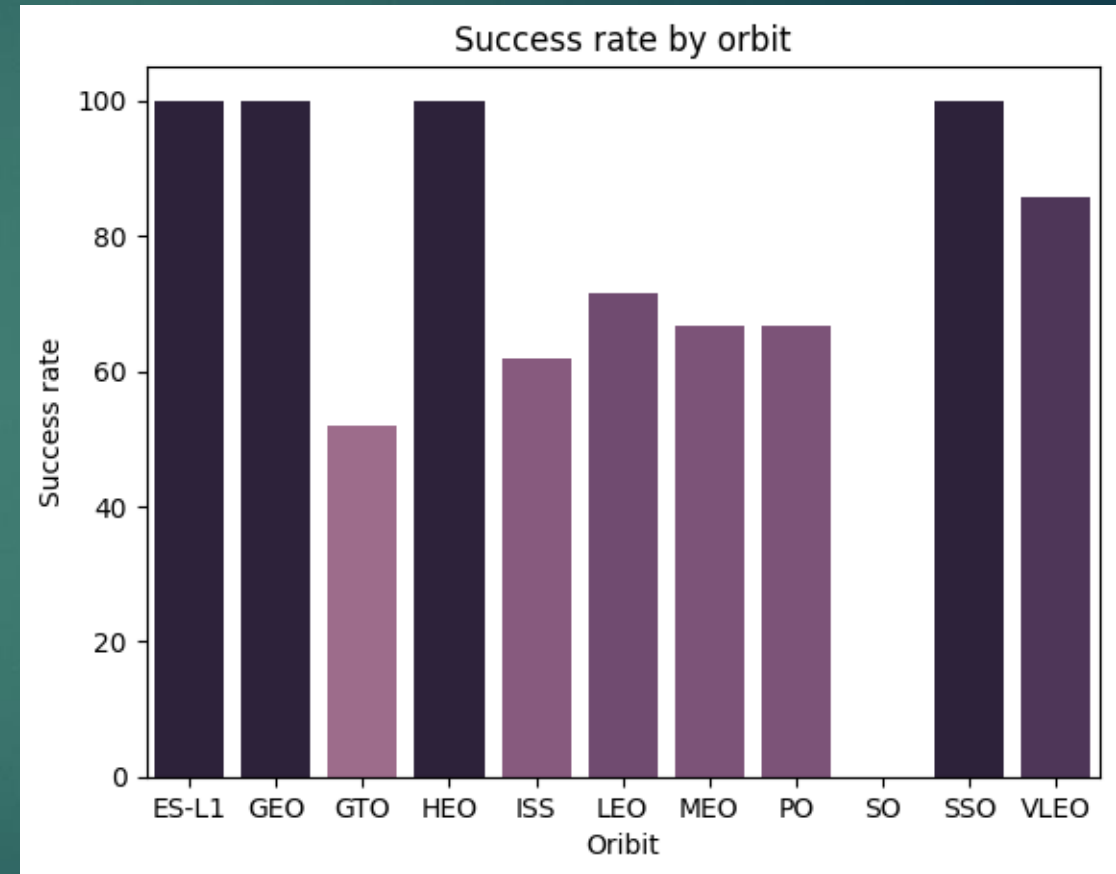
Observing Payload Vs. Launch Site scatter point chart shows that in the **VAFB-SLC 4E** Launch-site, there are no rockets launched for heavy payload Mass (greater than 10000).



EDA with Data Visualization

Visualize the relationship between success rate of each orbit type

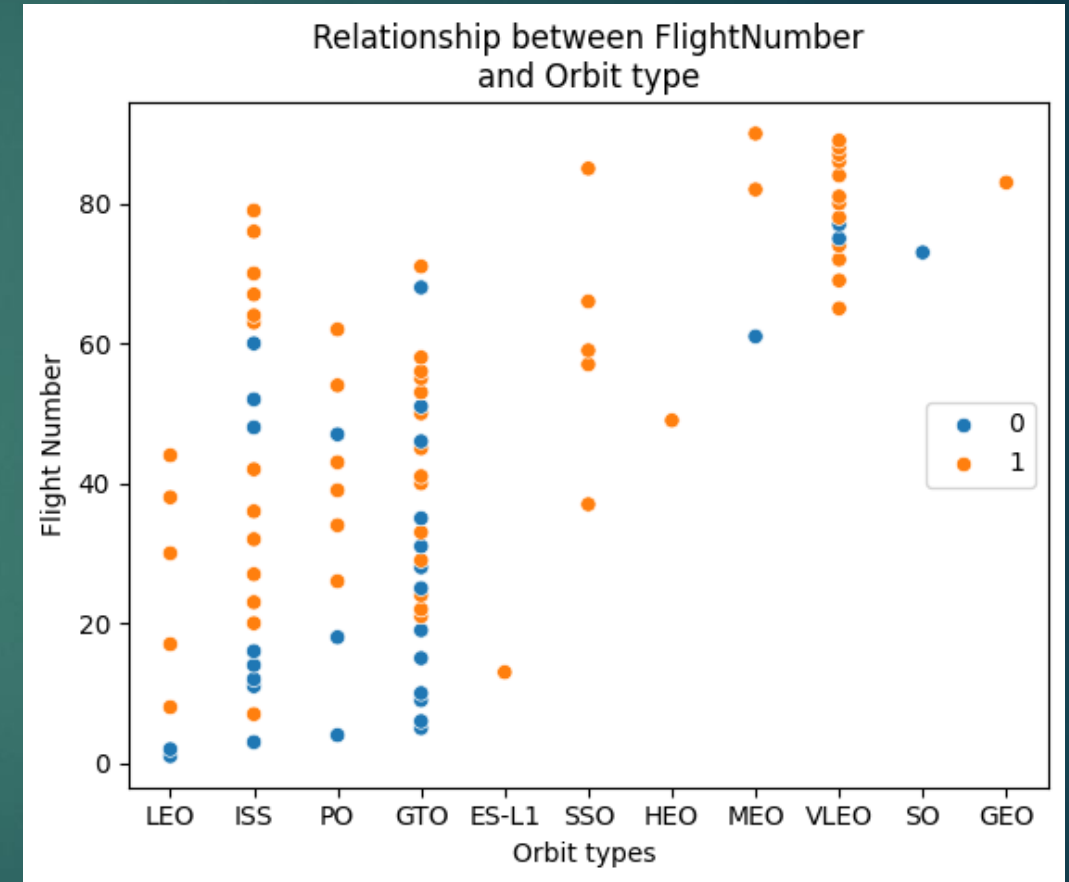
Analyzing the plotted bar chart indicates that **ES-L1** , **GEO** , **HEO** , and **SSD**, orbits have the highest success rate (100%)



EDA with Data Visualization

The relationship between FlightNumber and Orbit type.

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.

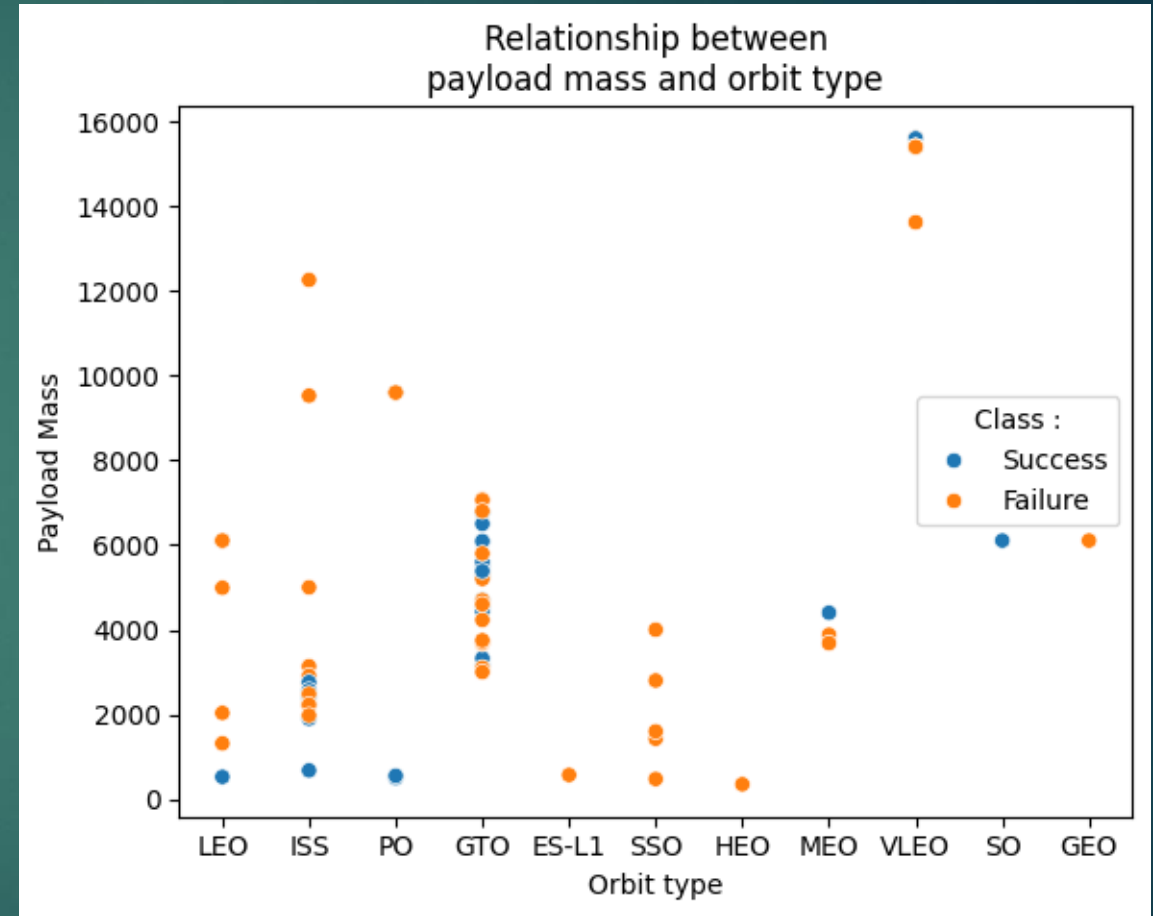


EDA with Data Visualization

The relationship between Payload mass and Orbit type

With heavy payloads the successful landing or positive landing rate are more for Polar, LEO and ISS.

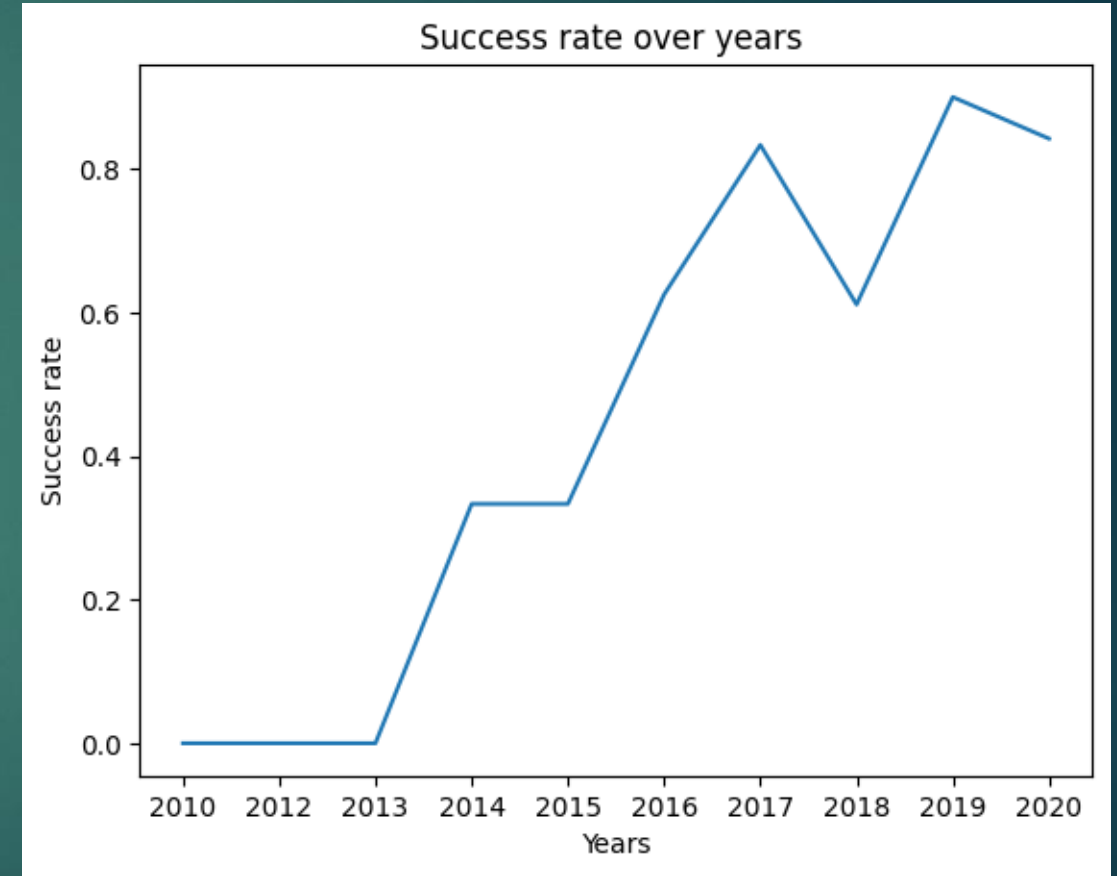
However for GTO we cannot distinguish this well as both positive landing rate and negative landing(unsuccesful mission) are both there here



EDA with Data Visualization

Success rate over years

We can observe that the success rate since 2013 kept increasing till 2017 (stable in 2014) and after 2015 it started increasing.



EDA with SQL

All launch site names

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

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

```
%sql select distinct Launch_Site from SPACEXTABLE ;
```

[11]

... * [sqlite:///my_data1.db](#)

Done.

...

Launch_Site

CCAFS LC-40

VAFB SLC-4E

KSC LC-39A

CCAFS SLC-40

Launch site names begin with 'CCA'

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

```
%sql select * from SPACEXTABLE where Launch_Site like 'CCA%' LIMIT 5 ;
```

Python

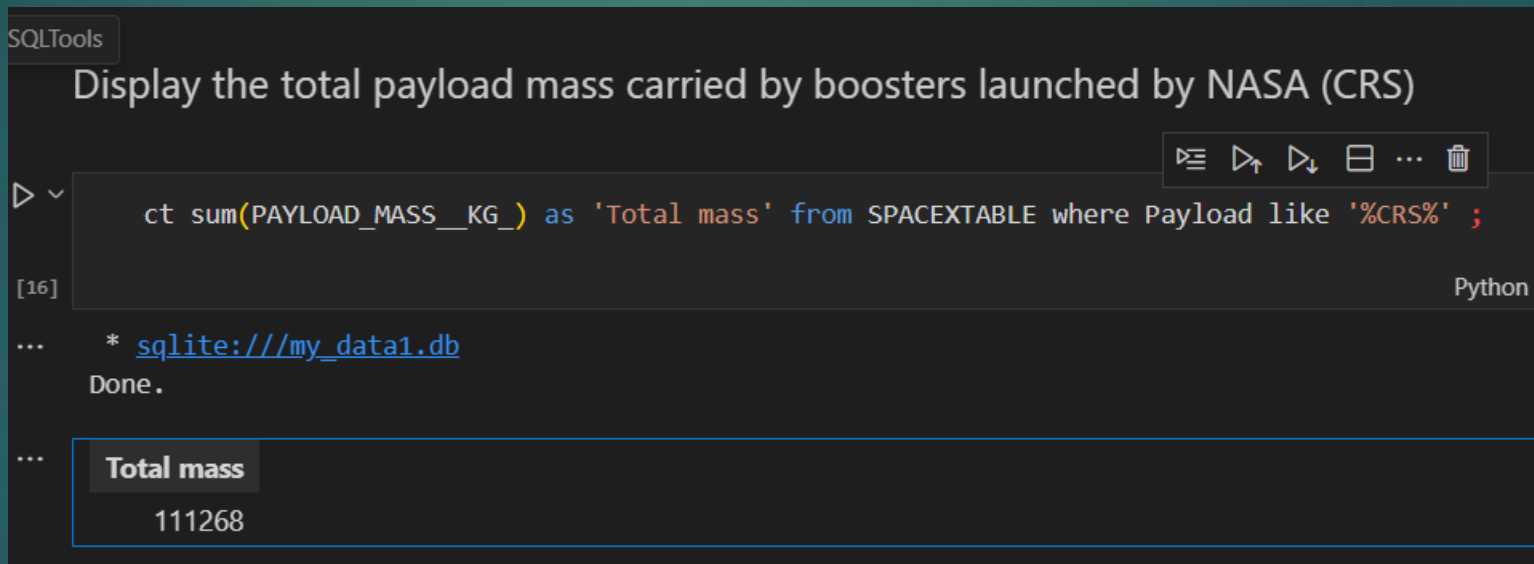
```
* sqlite:///my\_data1.db
```

Done.

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS_KG_	Orbit	Customer
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX
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
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)

Total payload mass

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



The screenshot shows a Jupyter Notebook interface with a dark theme. The top of the cell contains the text "SQLTools" and "Display the total payload mass carried by boosters launched by NASA (CRS)". Below this is a toolbar with icons for running, stepping through, and other actions. The main area of the cell contains an SQL query: `ct sum(PAYLOAD_MASS_KG_) as 'Total mass' from SPACEXTABLE where Payload like '%CRS%' ;`. Below the query, the output is displayed as a table with two columns: "Total mass" and "111268".

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

ct sum(PAYLOAD_MASS_KG_) as 'Total mass' from SPACEXTABLE where Payload like '%CRS%' ;

[16] Python

* sqlite:///my_data1.db
Done.

Total mass
111268
```

Average payload mass by F9 v1.1

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

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

%sql select avg(PAYLOAD_MASS_KG_) as 'AVG payload mass(booster F9 v1.1)'
|from SPACEXTABLE WHERE Booster Version = 'F9 v1.1' ;

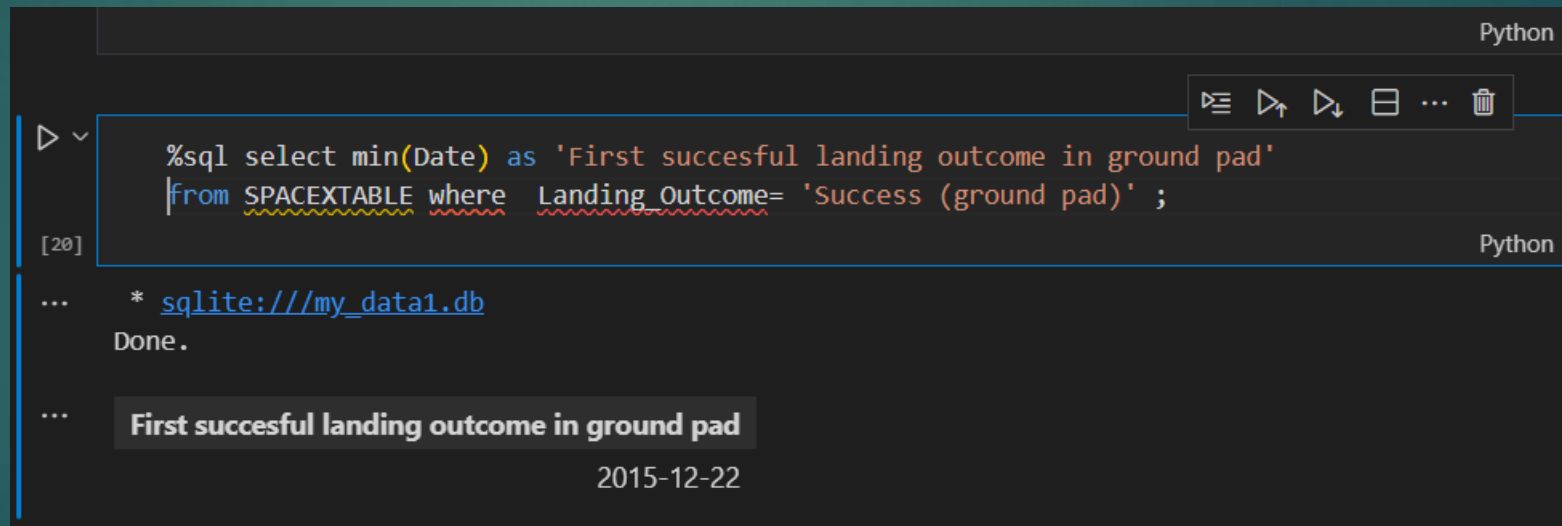
[17] Python

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

... AVG payload mass(booster F9 v1.1)
2928.4
```

First successful ground landing date

Listing the date when the first successful landing outcome in ground pad was achieved



```
Python

%sql select min(Date) as 'First succesful landing outcome in ground pad'
from SPACEXTABLE where Landing_Outcome= 'Success (ground pad)' ;

[20]

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

... First succesful landing outcome in ground pad
2015-12-22
```

The screenshot shows a Jupyter Notebook interface with a dark theme. At the top right, there's a tab labeled 'Python'. Below it, a toolbar contains icons for running, stepping through, and other code execution functions. The main area shows a code cell with an SQL query. The query is: `%sql select min(Date) as 'First succesful landing outcome in ground pad' from SPACEXTABLE where Landing_Outcome= 'Success (ground pad)' ;`. Below the code, the output is displayed. It starts with `[20]`, followed by a line `* sqlite:///my_data1.db` and `Done.`. Then, there's a header `First succesful landing outcome in ground pad` and a single data point `2015-12-22`.

Successful drone landing with payload between 4000 and 6000

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

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

```
[21]: %sql select Booster_Version from SPACEXTABLE 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   |
| F9 FT B1031.2   |


```

Total number of successful and failure mission outcomes

Listing the total number of successful and failure mission outcomes

Total successful mission outcomes	Total failure mission outcomes
61	10

```
%sql select count(*) as 'Total SUCCESS' from SPACEXTABLE
WHERE Landing_Outcome like '%Success%';
[23]
* sqlite:///my_data1.db
Done.

Total SUCCESS
61

%sql select count(*) as 'Total FAILURE' from SPACEXTABLE
WHERE Landing_Outcome like '%Failure%';
[24]
* sqlite:///my_data1.db
Done.

Total FAILURE
10
```


Boosters carried maximum payload

Listing the names of the booster versions which have carried the maximum payload mass

```
Task 8

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

PACEXTABLE WHERE PAYLOAD_MASS_KG_ = (select max(PAYLOAD_MASS_KG_) from SPACEXTABLE ) ;

[18] ✓ 0.0s Python

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

Listing the failed landing outcomes in drone ship, their booster versions and launch site names for the months in year 2015

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.

Note: SQLite does not support monthnames. So you need to use substr(Date, 6,2) as month to get the months and substr(Date,0,5)='2015' for year.

```
#Landing_Outcome : Failure (drone ship)
%sql select  substr(Date , 6,2) as month , Landing_Outcome , Booster_Version , Launch_Site
from SPACEXTABLE group by month , Landing_Outcome, Booster_Version, Launch_Site
HAVING Landing_Outcome = 'Failure (drone ship)';
```

[26]

Python

... * [sqlite:///my_data1.db](#)

Done.

...

month	Landing_Outcome	Booster_Version	Launch_Site
01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
01	Failure (drone ship)	F9 v1.1 B1017	VAFB SLC-4E
03	Failure (drone ship)	F9 FT B1020	CCAFS LC-40
04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40
06	Failure (drone ship)	F9 FT B1024	CCAFS LC-40

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

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

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 count (Landing_Outcome) as counts, Landing_Outcome
from SPACEXTABLE
group by Landing_Outcome
having Date > '2010-06-04' AND Date < '2017-03-20'
order by counts desc ;
```

Python

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

Done.

counts	Landing_Outcome
21	No attempt
14	Success (drone ship)
9	Success (ground pad)
5	Failure (drone ship)
5	Controlled (ocean)
2	Uncontrolled (ocean)
1	Precluded (drone ship)

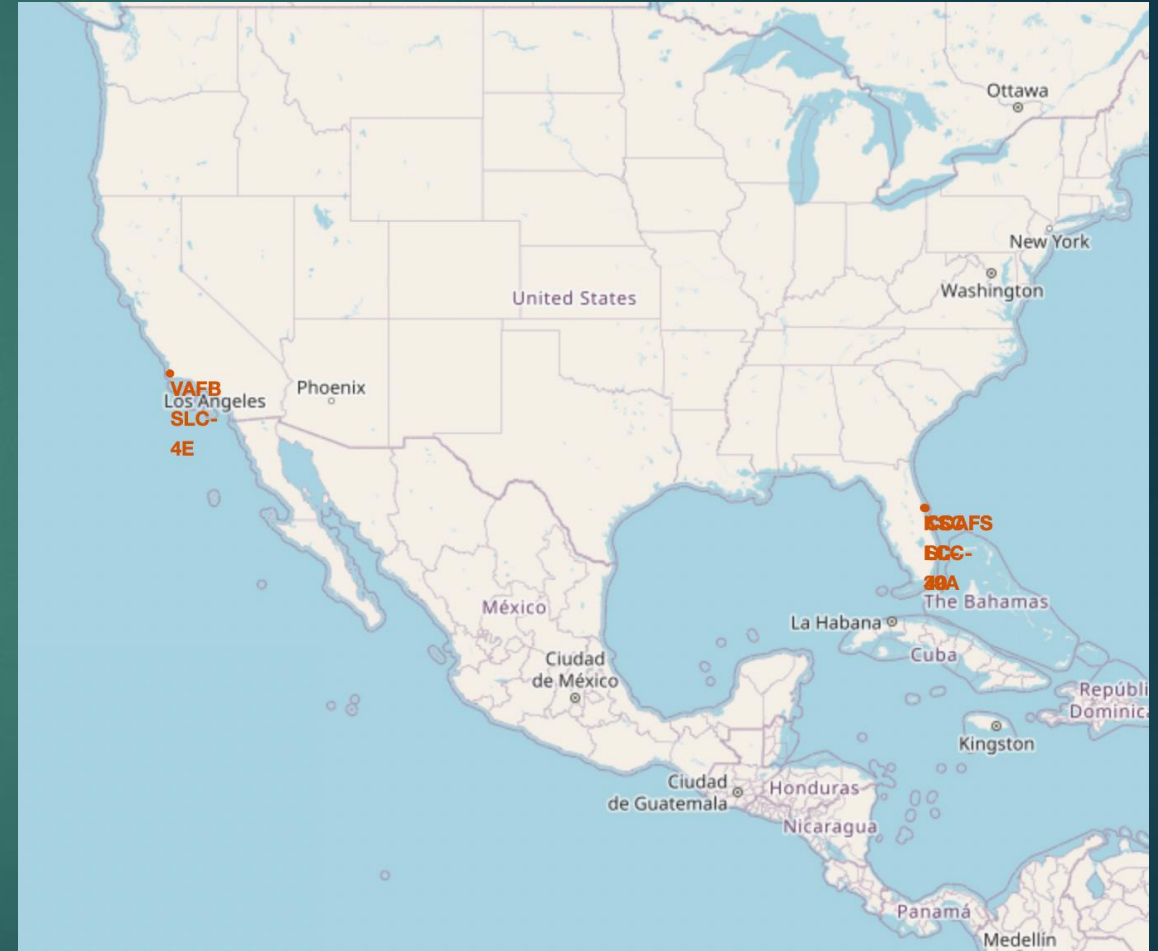
Interactive map with Folium

All launch sites location markers on a global map

Most of Launch sites are in proximity to the Equator line. The land is moving faster at the equator than any other place on the surface of the Earth. Anything on the surface of the Earth at the equator is already moving at 1670 km/hour. If a ship is launched from the equator it goes up into space, and it is also moving around the Earth at the same speed it was moving before launching. This is because of inertia.

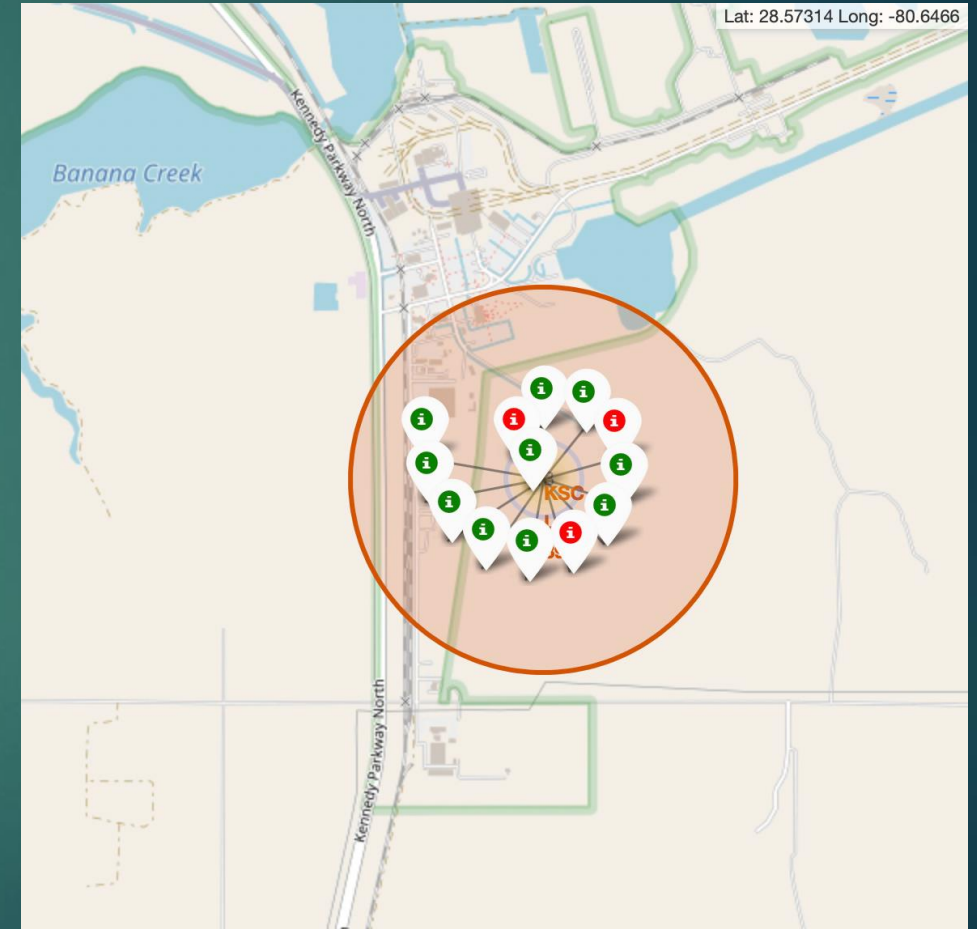
This speed will help the spacecraft keep up a good enough speed to stay in orbit.

All launch sites are in very close proximity to the coast, while launching rockets towards the ocean it minimises the risk of having any debris dropping or exploding near people.



Colour-labeled launch records on the map

- From the colour-labeled markers we should be able to easily identify which launch sites have relatively high success rates.
- **Green** Marker = Successful Launch
- **Red** Marker = Failed Launch
- Launch Site KSC LC-39A has a very high Success Rate



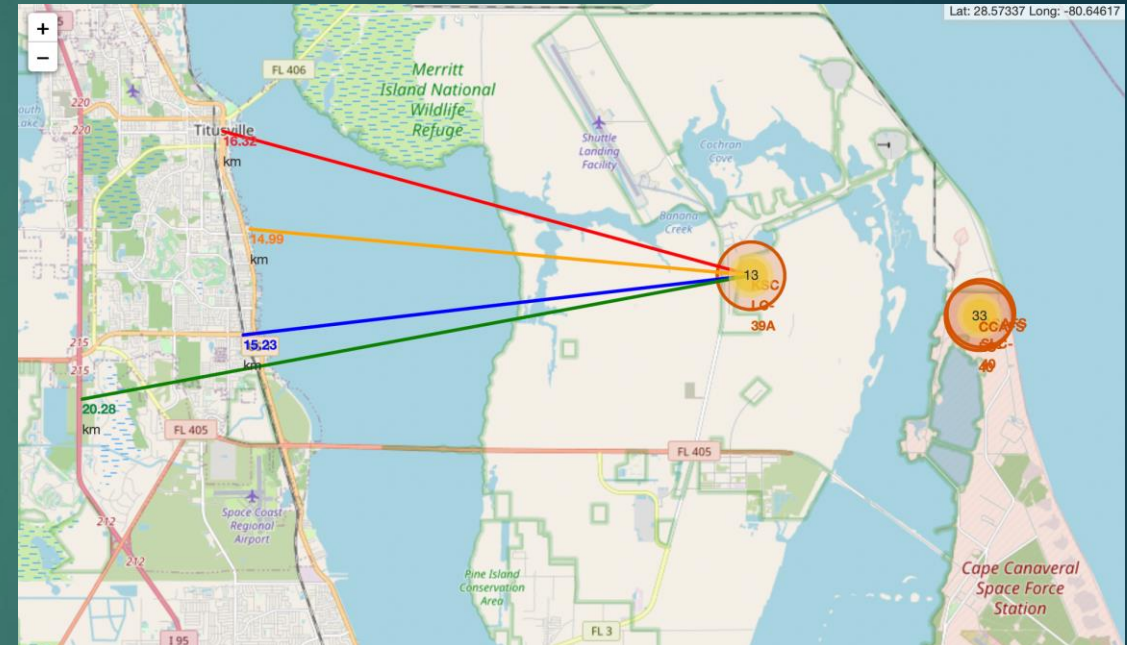
Distance from the launch site KSC LC-39A to its proximities

From the visual analysis of the launch site KSC LC-39A we can clearly see that it is:

- relative close to railway (15.23 km)
- relative close to highway (20.28 km)
- relative close to coastline (14.99 km)

- Also the launch site KSC LC-39A is relative close to its closest city Titusville (16.32 km).

- Failed rocket with its high speed can cover distances like 15-20 km in few seconds. It could be potentially dangerous to populated areas





Build a Dashboard with Plotly Dash

Launch success count for all sites

Total Success Launches by Site



The chart clearly shows that from all the sites, KSC LC 39A has the most successful launches

Launch site with highest launch success ratio

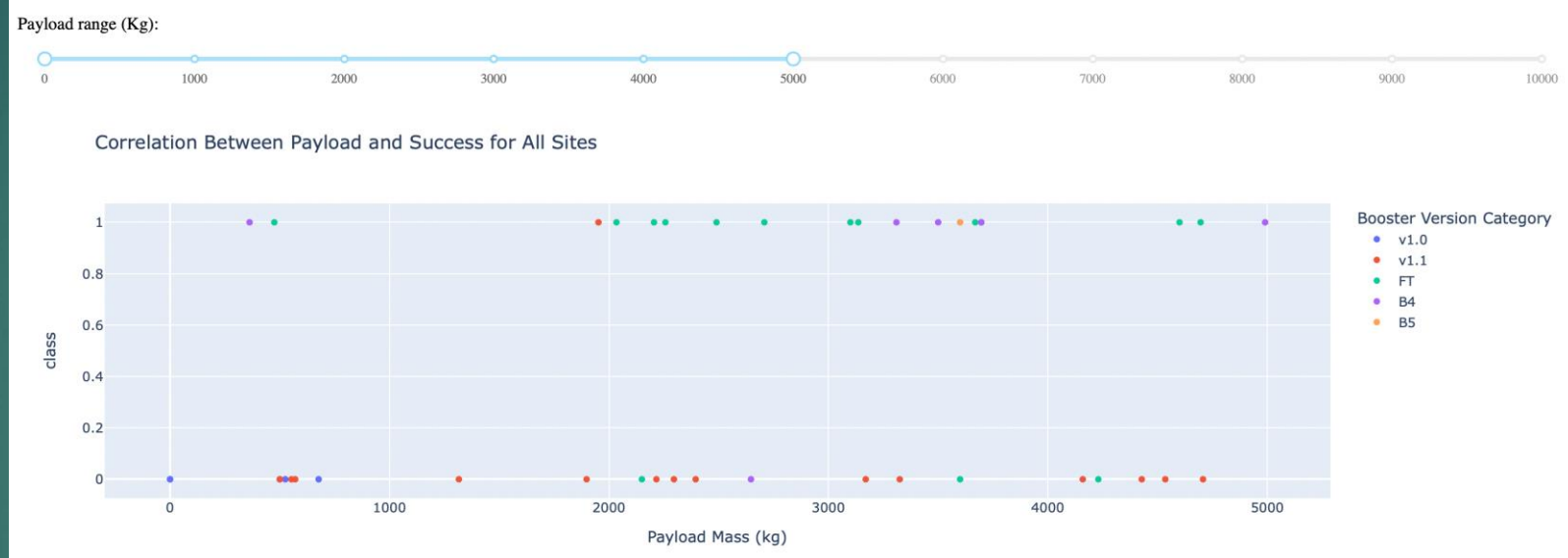
Total Success Launches for Site KSC LC-39A



KSC LC-39A has the highest launch success rate (76.9%) with 10 successful and only 3 failed landings

Payload Mass vs. Launch Outcome for all sites

The charts show that payloads between 2000 and 5500 kg have the highest success rate.





Predictive analysis (Classification)

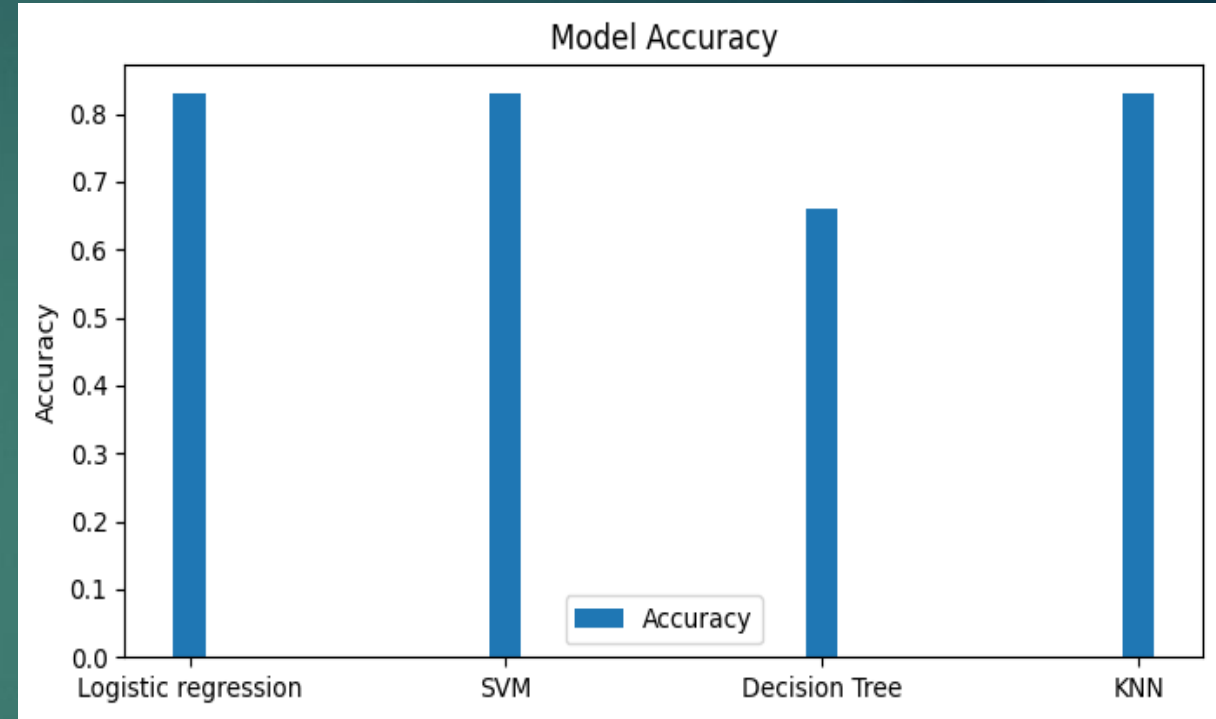
Classification Accuracy

The models had virtually the same accuracy on the test set at 83.33% accuracy, except the decision tree classifier with 66%.

It should be noted that test size is small at only sample size of 18.

This can cause large variance in accuracy results, such as those in Decision Tree Classifier model

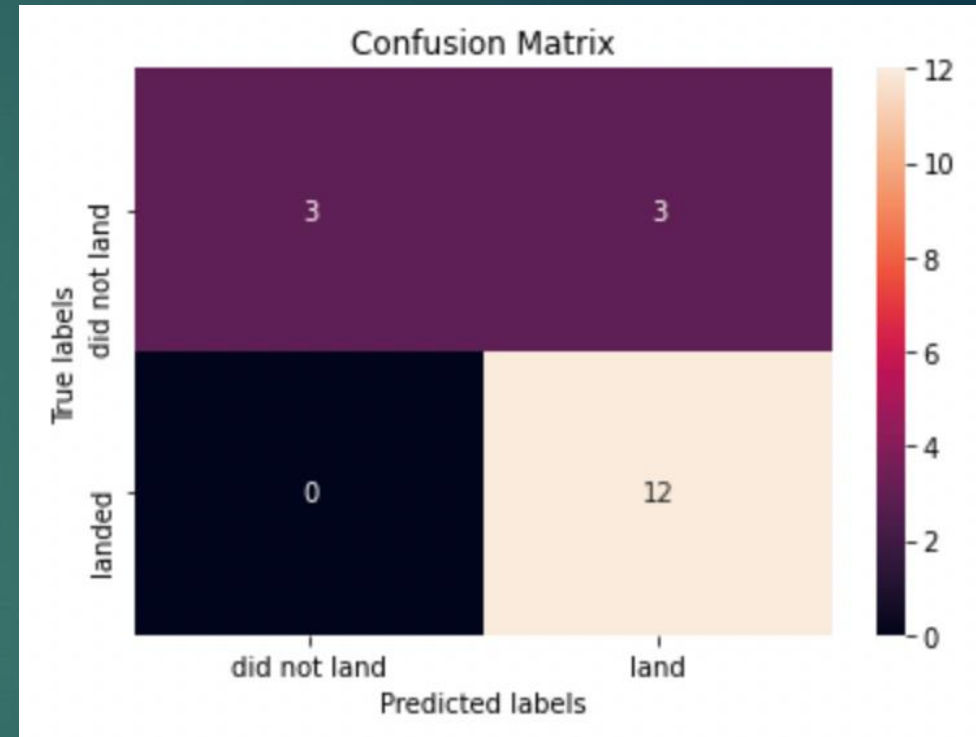
We likely need more data to determine the best model.



	accuracy
Logistic regression	0.83
SVM	0.83
Decision Tree	0.66
KNN	0.83

Confusion Matrix

- Since all models performed the same for the test set, the confusion matrix is the same across all models. (*except Decision Tree*)
- The models predicted 12 successful landings when the true label was successful landing
- The models predicted 3 unsuccessful landings when the true label was unsuccessful landing.
- The models predicted 3 successful landings when the true label was unsuccessful landings (false positives).
- Our models make good predictions of successful landings.



Conclusions

- Decision Tree, KNN and SVM Models are the best algorithms for this dataset.
- Launches with a low payload mass show better results than launches with a larger payload mass.
- Most of launch sites are in proximity to the Equator line and all the sites are in very close proximity to the coast.
- The success rate of launches increases over the years.
- KSC LC-39A has the highest success rate of the launches from all the sites.
- Orbits ES-L1, GEO, HEO and SSO have 100% success rate

Appendix

GitHub repository url :

https://github.com/anasheros/Applied_data_science

Special Thanks to All Instructors:

[IBM Data Science Professional Certificate | Coursera](#)