IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Thet Khine Lin
08/28/2023

Outline

- Executive Summary for the whole Project
- Introduction about the Project
- Methodology applied in the Project
- Results of the Project
- Conclusion
- Appendix

Executive Summary

Summary of Methodology

- The methodology of this project encompasses a comprehensive approach that combines data collection, preprocessing, exploratory data analysis (EDA), interactive visual analysis, and predictive modeling.
- The project aims to predict whether a rocket will land successfully or not, utilizing a classification model and visualizing the rate of successful and fail launching rate by creating interactive maps and dashboard.

Executive Summary

Summary of Results

- The results of the project reveal valuable insights into the factors influencing the success of rocket landings.
- Through a combination of exploratory data analysis (EDA), interactive visualizations, and predictive modeling, the project has provided a comprehensive understanding of rocket landing outcomes.
- The EDA phase, driven by visualizations and SQL queries, has uncovered significant patterns and trends within the dataset. Visualization techniques have illuminated correlations between features and outcomes. And SQL queries enabled efficient data exploration, revealing historical trends and finding data distributions.
- The integration of Folium and Plotly Dash has led to the creation of interactive maps and dashboards. These tools facilitate a deeper analysis of rocket launch site proximities and their successful launching rate. Folium-generated maps offer geographical insights into successful and unsuccessful launching sites, aiding in the identification of optimal launch locations. The Plotly Dash dashboard enhances user interaction, enabling dynamic exploration of Outcomes(success or fail launching) in each Launching Site by using Pie chart and, correlation between Payload mass (kg) and Outcomes(success or fail launching) by using scatter plot.
- Developing classification model leverages to anticipate rocket landing outcomes. By utilizing features derived from cleaned data, the model delivers predictions on the likelihood of successful landings. Model is evaluated by calculating accuracy and confusion matrix, which provides a clear depiction of predictive performance.

Introduction

Project Aim

- The aim of this project is to determine if the first stage of the rocket will land successfully or not.

Project Background

- To get the approximate cost of a launch, we have to determine the first stage of a rocket will land successfully or not.
- Space X advertises Falcon 9 rocket launches with a cost of 62 million dollars and other providers cost upwards of 165 million dollars each. This is because Space X reuse the first stage of a rocket. In Falcon 9 rocket, it has two stages, the second stage helps bring the payload to orbit but most of the work are done by the first stage. The first stage is comparatively larger than the second stage and also it is relatively more expensive than the second stage. Therefore, if we can determine if the first stage will land or not, then we can also determine the approximate cost of a launch.

Project Contents

- This project contains collecting data, cleaning data, exploring data by using visualization and SQL, creating interactive maps and dashboard , developing classification model , and evaluating the model.

Introduction

Problems that we want to find answers in this Project

1. Find optimal Launch Sites by extracting geographical insights into successful and unsuccessful landing sites based on public information of Space X?
2. Find the correlation between Payload mass (kg) and Outcomes of a rocket landing (Success or Fail landing) based on Booster Version?
3. Find if the rocket will land successfully or not by developing a machine learning model?

Section 1

Methodology

Contents of Methodology

(1) Data Collection

- Collect the data from API and Web scraping
- Convert the data to correct format to read in Python

(2) Data Wrangling

- Clean the data that are collected from API and Web scraping to perform further process

(3) Exploratory Data Analysis

- Explore the data
- Do Feature Engineering
- Find Insights

(4) Interactive Visual Analysis

- Perform Interactive Visual Analysis by creating maps and dashboard

(5) Building Machine Learning Model

- Create a machine learning model to predict if the first stage will land or not.

(1) Data Collection

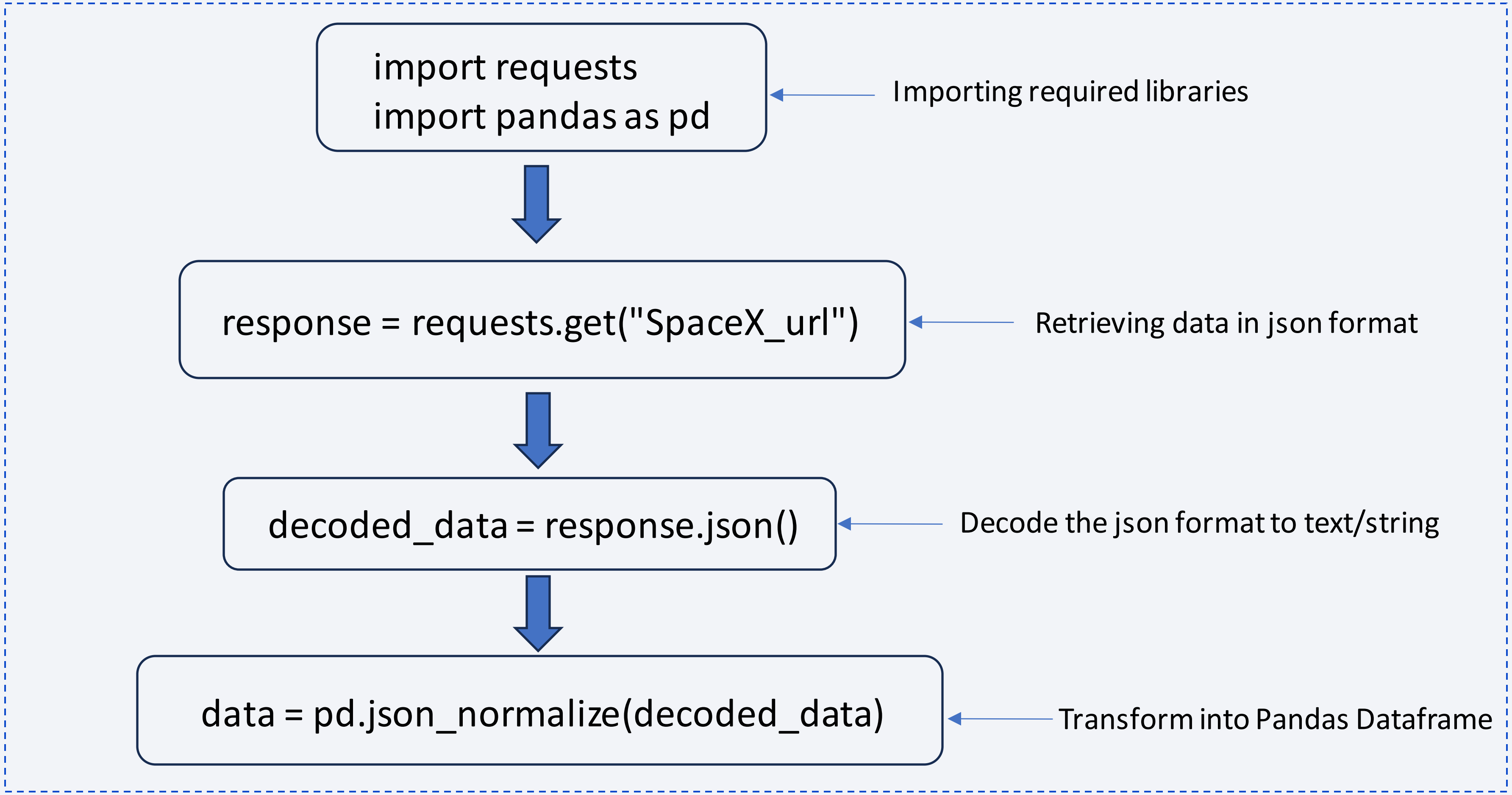
Step by step process of using SpaceX API

- To use SpaceX API in Python, first the "requests" library is required to import. It allows to make HTTP requests to get data from an API.
- Then the get function is called, which is supported by "requests" library. As an argument, the desired URL is passed, which is SpaceX API.
- The format of data, which are retrieved from "requests.get()" are in "json" format. So, they need to be decoded it into string/text by using "json()" function and then transform it into pandas dataframe by using "json_normalize()" function to be able to perform EDA and building ML models.

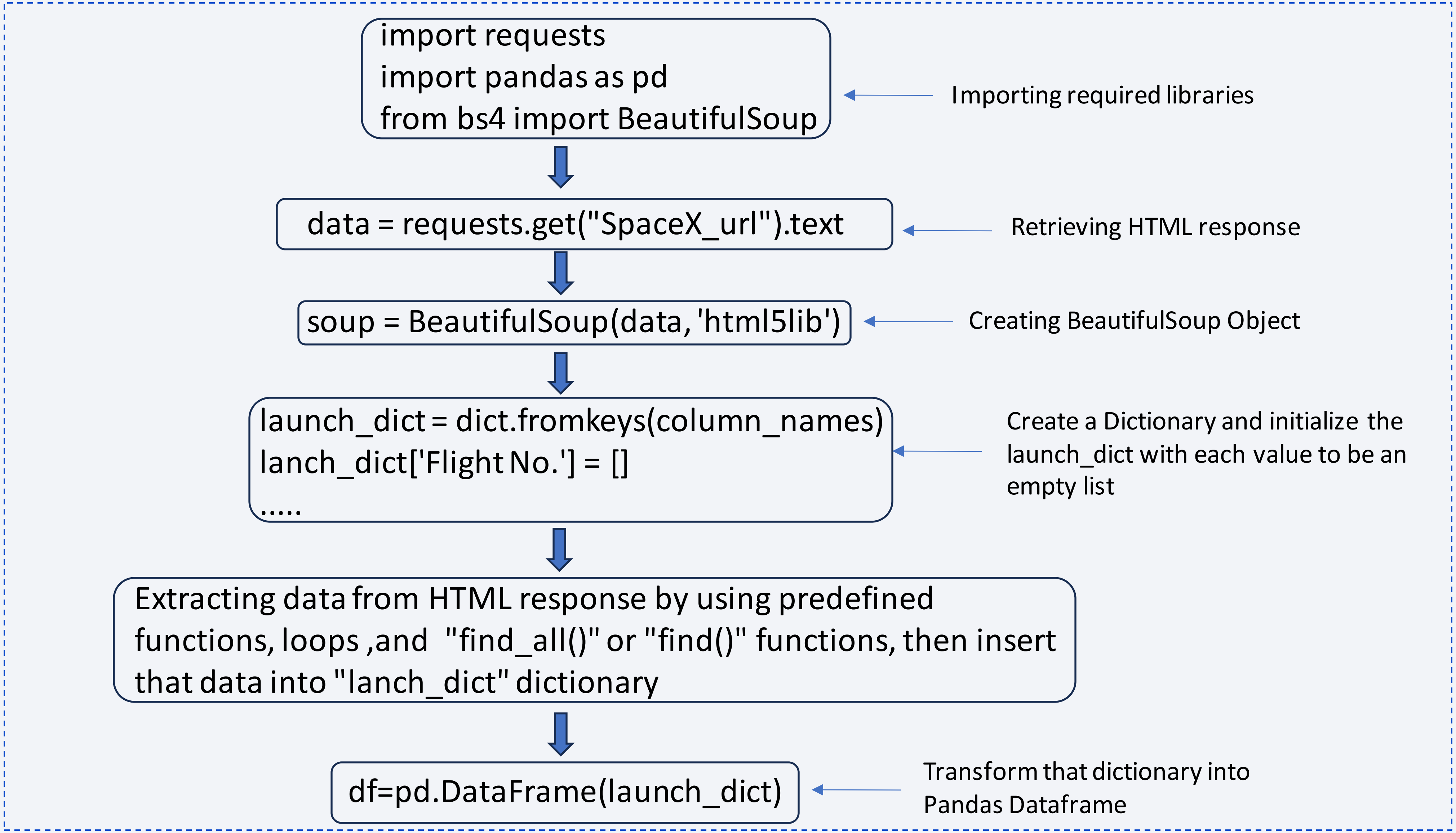
Step by step process of using Web Scraping

- To perform Web Scraping, "requests", and "BeautifulSoup" libraries are required to import for retrieving HTML response and extracting data from that HTML response, respectively.
- First, the url of whole Falcon9 Launch HYML page from wikipedia is passed to "request.get()" function and change the datatype to text/string.
- Then, "BeautifulSoup()" class is used to create a BeautifulSoup object from a response text content by giving the HTML response that has been transformed into text and "html5lib" third-party Python library that provides a pure-Python implementation for parsing and manipulation HTML documents.
- After applying "BeautifulSoup()", the data from the HTML response can be extracted by using "find" and "find_all" functions.
- These extracted data are then stored in a dictionary with respected keys to transform them into pandas dataframe.

Data Collection flow chart – SpaceX API



Data Collection flow chart – Web Scraping



GitHub URL for Data Collection

- GitHub url for Data Collection with SpaceX_API:
https://github.com/thetkhinelin25/SpaceX_Data_Collection_with_API.git

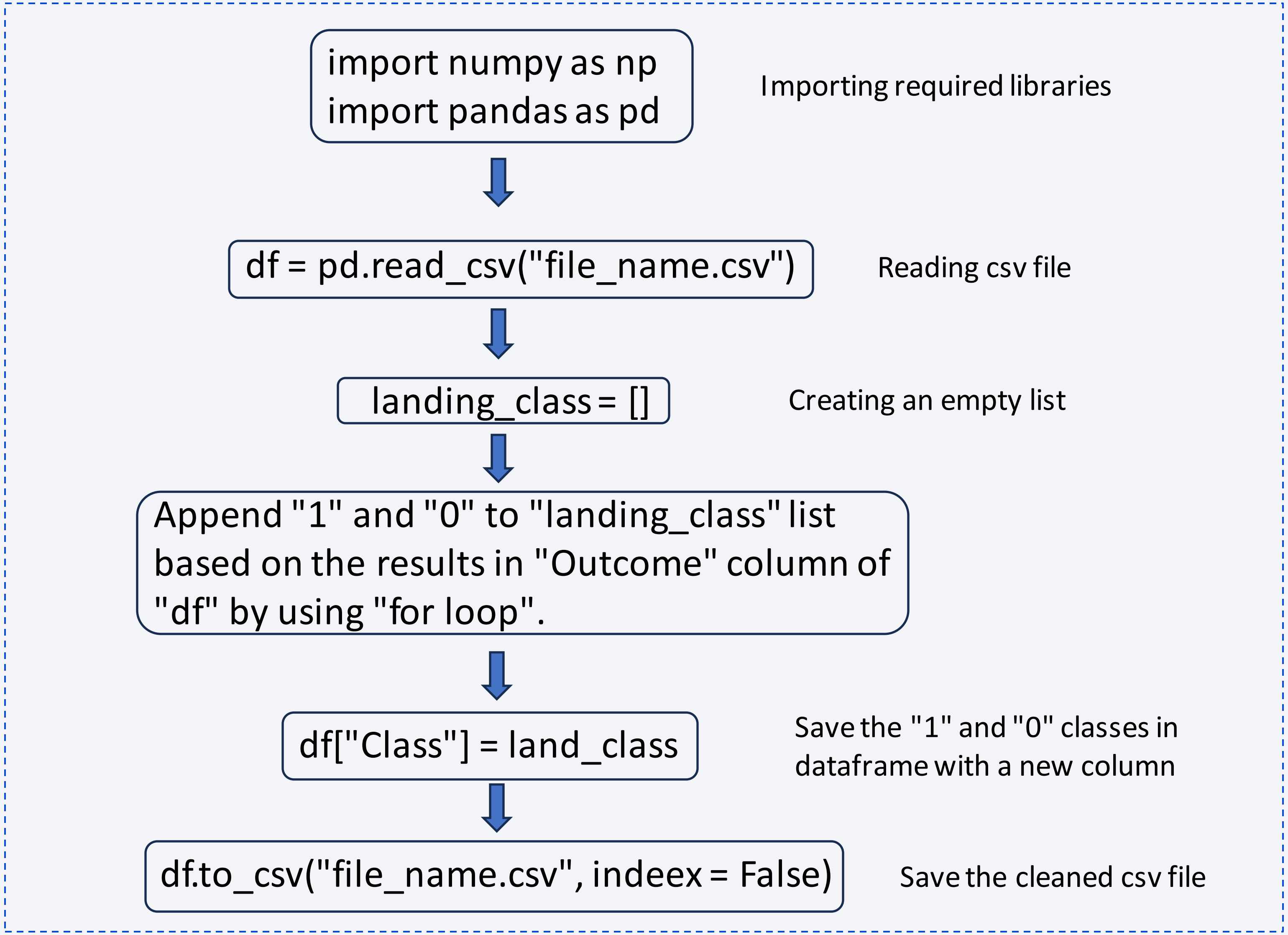
- GitHub url for Data Collection with Web Scraping:
https://github.com/thetkhinelin25/SpaceX_Data_Collection_with_Web_Scraping.git

(2) Data Wrangling

Step by step process of Data Wrangling

- The main task of data wrangling for this project is to convert the outcomes into Training Labels with 1 and 0. 1 means the booster successfully landed and 0 means it was unsuccessful.
- To do this, first, the necessary libraries, which are pandas and numpy, are imported.
- Then read the csv file that contains the data, which are collected in the previous Data Collecting stage.
- After the file has been read, an empty list, named as "landing_class", is created and then insert 1 and 0 based on the results in "Outcome" column of a dataframe by using "for loop".
- Once 1 and 0 have been inserted to the "landing_class" list, it is added to the original dataframe as new column, named "Class".

Data Wrangling flow chart



GitHub URL for Data Wrangling

- GitHub url for Data Wrangling:
https://github.com/thetkhinelin25/SpaceX-Data-Wrangling.git

(3) EDA with Data Visualization

No.	Chart Type	Applied columns	Purpose
1.	Scatter plot	X = "FlightNumber" Y = "PayloadMass"	To explore how FlightNumber and PayloadMass variables would affect the launch outcome.
2.	Scatter plot	X = "FlightNumber" Y = "LaunchSite"	To explore how FlightNumber and LaunchSite variables would affect the launch outcome.
3.	Scatter plot	X = "PayloadMass" Y = "LaunchSite"	To explore how PayloadMass and LaunchSite variables would affect the launch outcome.
4.	Bar chart	X = "Orbit" Y = "Success rate"	To visualize the relationship between success rate of each orbit type
5.	Scatter plot	X = "FlightNumber" Y = "Orbit"	To explore how FlightNumber and Orbit variables would affect the launch outcome.
6.	Scatter plot	X = "PayloadMass" Y = "Orbit"	To explore how PayloadMass and Orbit variables would affect the launch outcome.
7.	Line chart	X = "Year" Y = "Average success rate"	To visualize the launch success by yearly trend

GitHub URL for EDA with Visualization

- GitHub url for EDA with Visualization:
https://github.com/thetkhinelin25/SpaceX-EDA-with-Visualization.git

(3) SQL Queries in EDA

1. Display the names of the unique launch sites in the space mission
2. Display 5 records where launch sites begin with the string 'CCA'
3. Display the total payload mass carried by boosters launched by NASA (CRS)
4. Display average payload mass carried by booster version F9 v1.1
5. List the date when the first successful landing outcome in ground pad was achieved.
6. List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
7. List the total number of successful and failure mission outcomes
8. List the names of the booster_versions which have carried the maximum payload mass.
9. List records which display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.
10. Rank the count of each landing outcomes group between the date 2010-06-04 and 2017-03-20, in descending order.

GitHub URL for EDA with SQL

- GitHub url for EDA with SQL:
https://github.com/thetkhinelin25/SpaceX-EDA-with-SQL.git

(4) Build an Interactive Map with Folium

- The map objects that are used in interactive Folium map are Marker, Circle, and Polyline.

Map Objects	Their Usage
Marker	• To label the name of the launch side on the map based on the latitudes and longitudes of the launch sites. • To add the distance labels between launch site and its closet city, railway, highway, and selected coastline point.
Circle	• To circle around the location of the launch site to be able to visualize the launch site more easily on map.
Polyline	• To visualize the distance between launch site and its closet city, railway, highway, and selected coastline point.

GitHub URL for Interactive Map with Folium

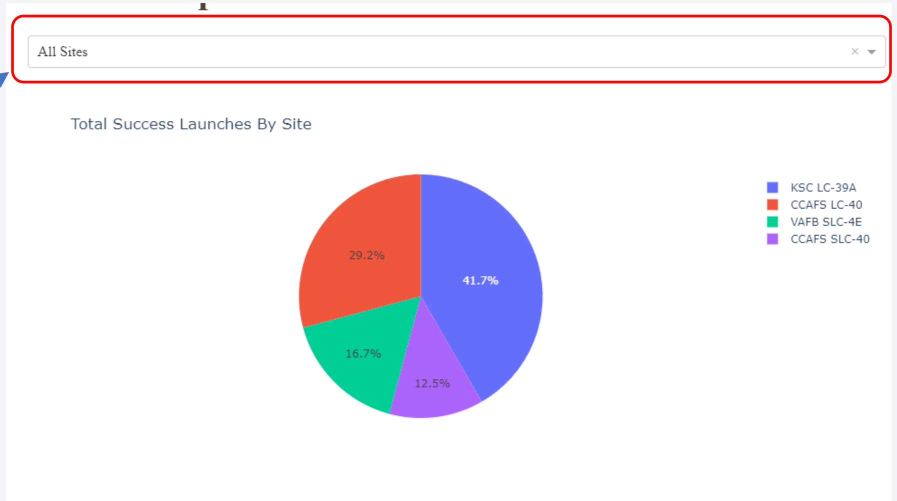
- GitHub url for Interactive Map with Folium: https://github.com/thetkhinelin25/SpaceX-Interactive-Map-with-Folium.git

(4) Build a Dashboard with Plotly Dash

Pie Chart

- It is used to show the total success/failure launching in percentage for each Launch Sites.
- For interaction, it contains a dropdown list to enable Launch Site selection.

Note: In the selection, if you select "All Sites", the pie chart will show the percentage of success launching for all sites.



Scatter Chart

- It is used to show the correlation between payload mass and Launch Success by coloring the scatters with Booster Version Category.
- For Interaction, it contains slider to select the payload mass range

GitHub URL for Dashboard with Plotly Dash

- GitHub url for Dashboard with Plotly Dash:
https://github.com/thetkhinelin25/SpaceX-Dashboard-with-Plotly-Dash.git

(5) Predictive Analysis (Classification)

Algorithms Selection

- Before the model is selected for the project, we built four different models by using four different algorithms based on the data that we collected.
- These algorithms are Logistic Regression, KNN, SVM and Decision Tree Classification.

Building Models

•The step by step process of building models is as follows:

1. Import the cleaned data
2. Define the features (independent variables) and target values (dependent variable) as X and Y
3. Standardize the features, not to bias on larger feature values while training the models
4. Split the features and target values into training set and testing set.
5. Build the models by finding the best hyperparameters (using GridSearchCV) for each models and training the models with train set.

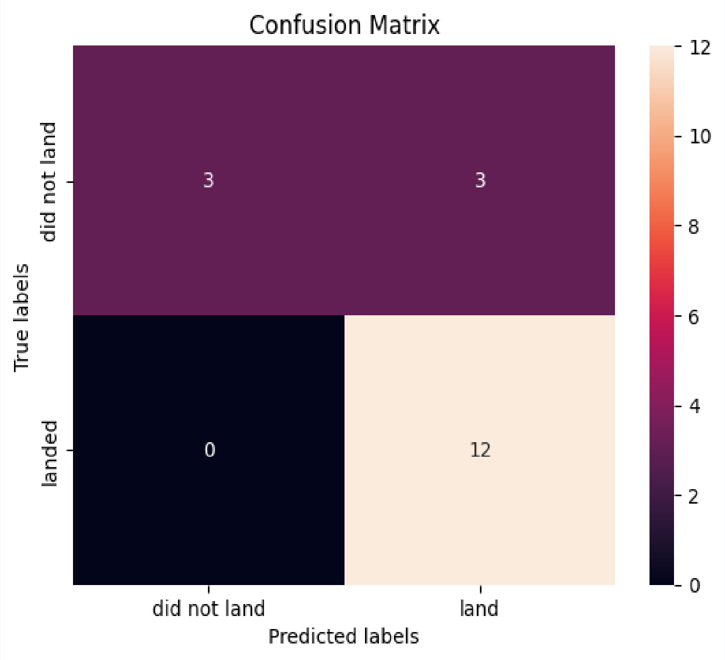
(5) Predictive Analysis (Classification)

Model Evaluation and Selecting best Model

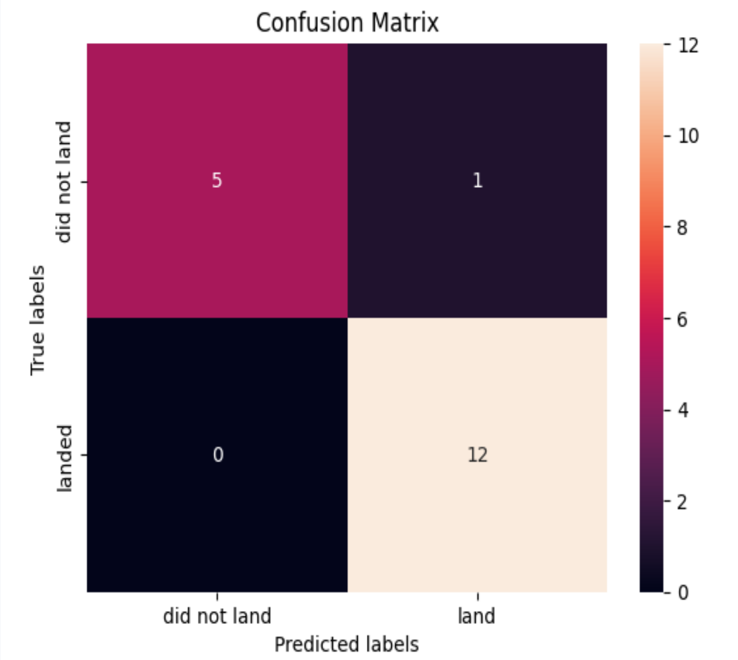
- After building the models, we selected the best model by testing with the test set and evaluate them by calculating the accuracy of test results and using Confusion Matrix to get clear depiction of predictive performance. Once we had evaluated the models, we chose Decision Tree Classification model.

- The evaluation results of the models are as follows:

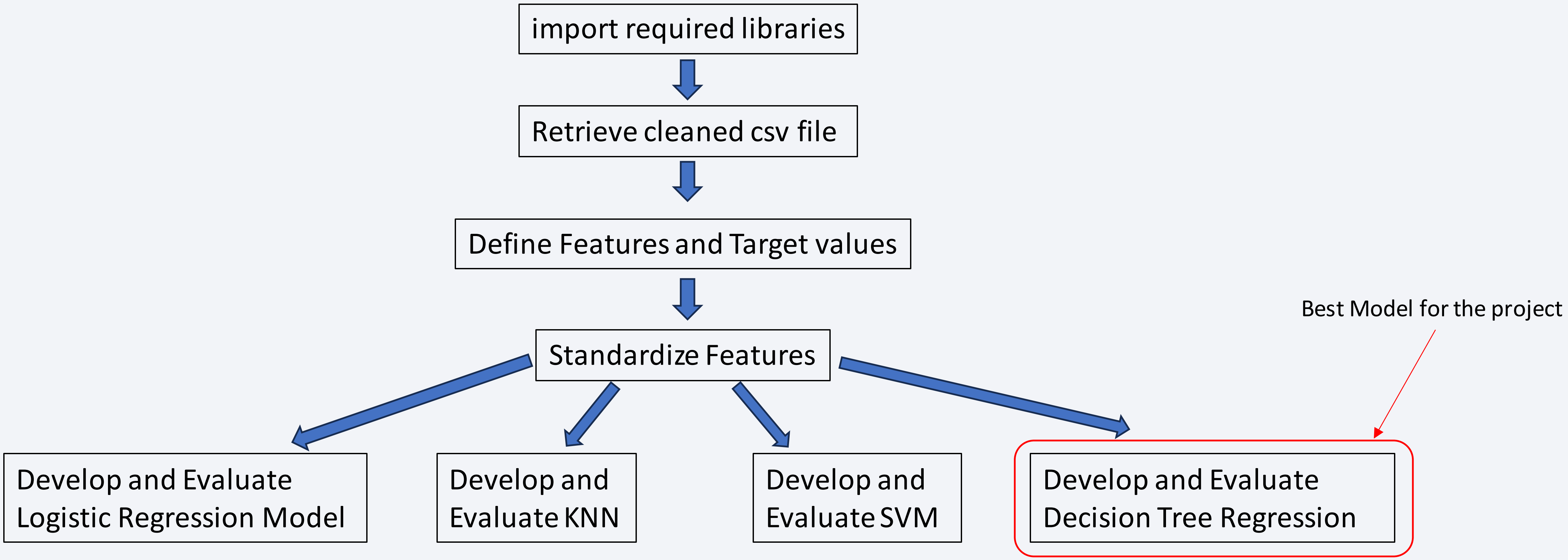
Logistic Regression, KNN, SVM
Test accuracy = `0.8333333333333334`



Decision Tree Classification
Test accuracy = `0.9444444444444444`



(5) Predictive Analysis (Classification)



GitHub URL for Predictive Analysis (Classification)

- GitHub url for Predictive Analysis (Classification):
https://github.com/thetkhinelin25/SpaceX-Predictive-Analysis-Classification-.git

Contents of Results

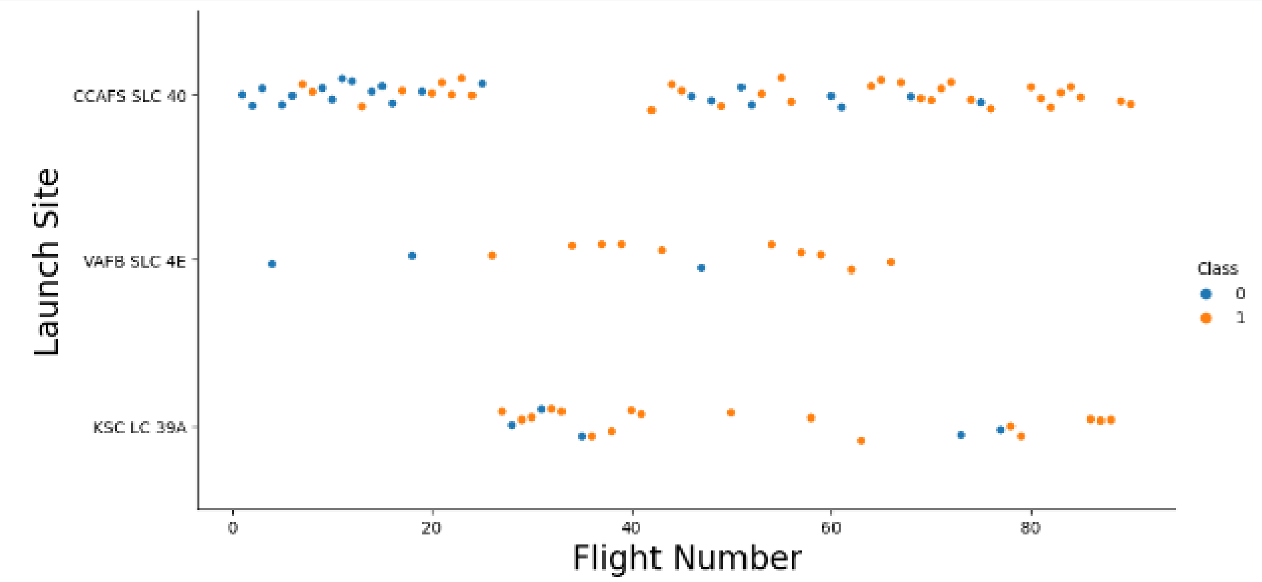
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

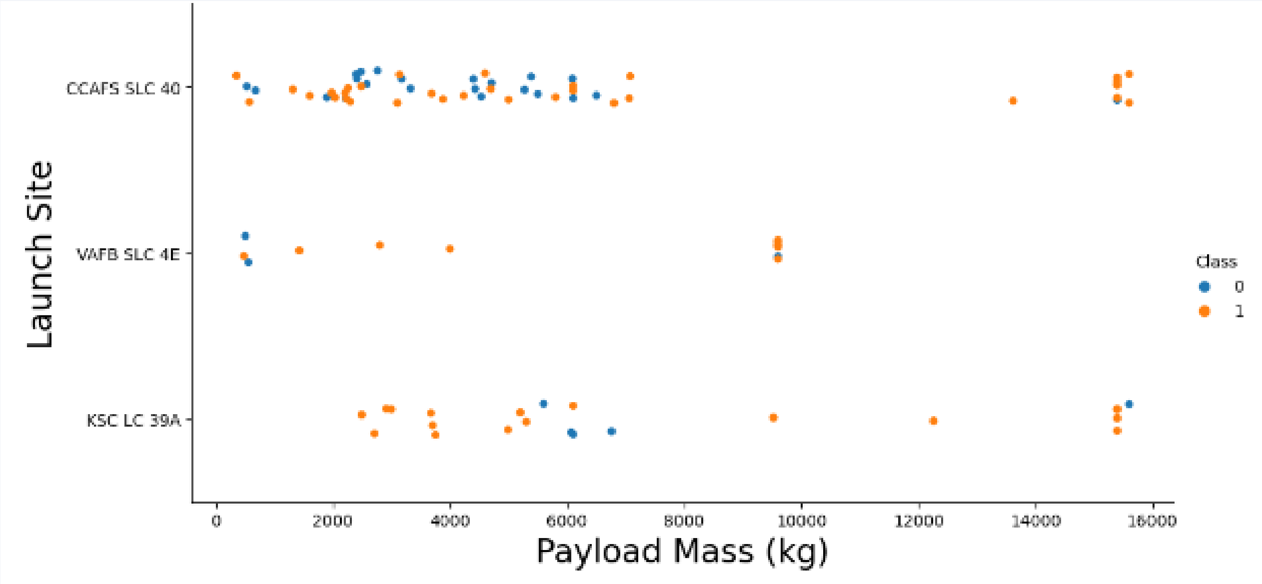
Flight Number vs. Launch Site

- Based on the scatter plot, we can say that most of the rocket are launched in "CCAFS SLC 40" launch site.
- The least launch are done in "VAFB SLC 4E" launch site.
- To talk about the success rate, we can say that in each launch site, the number of successful launches are more than the number of fail launches.



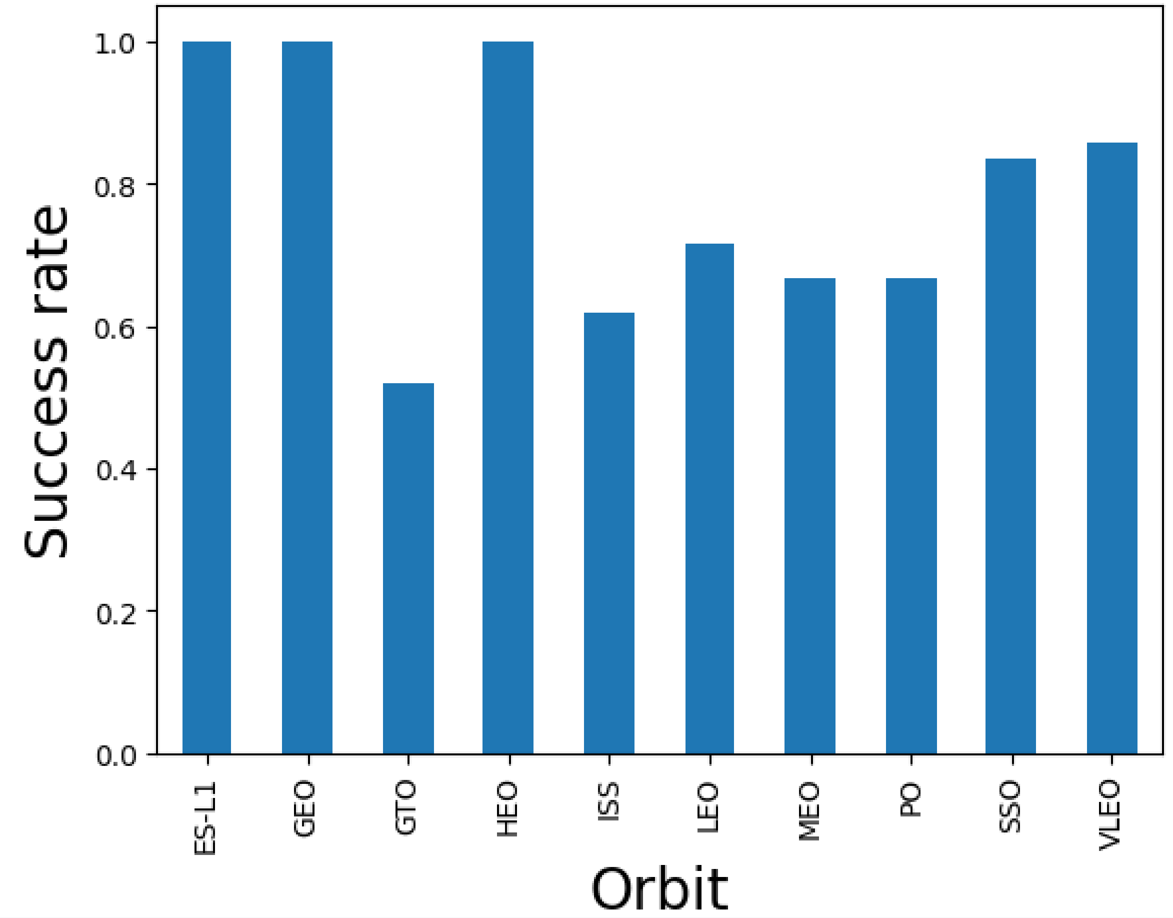
Payload vs. Launch Site

- In the range between 9000 to 15000 of Payload Mass, the launches are clearly more successful than in the range between 0 to 8000 of Payload Mass.
- So, we can say that in each launch site, increasing Payload Mass also increases the number of success launches.



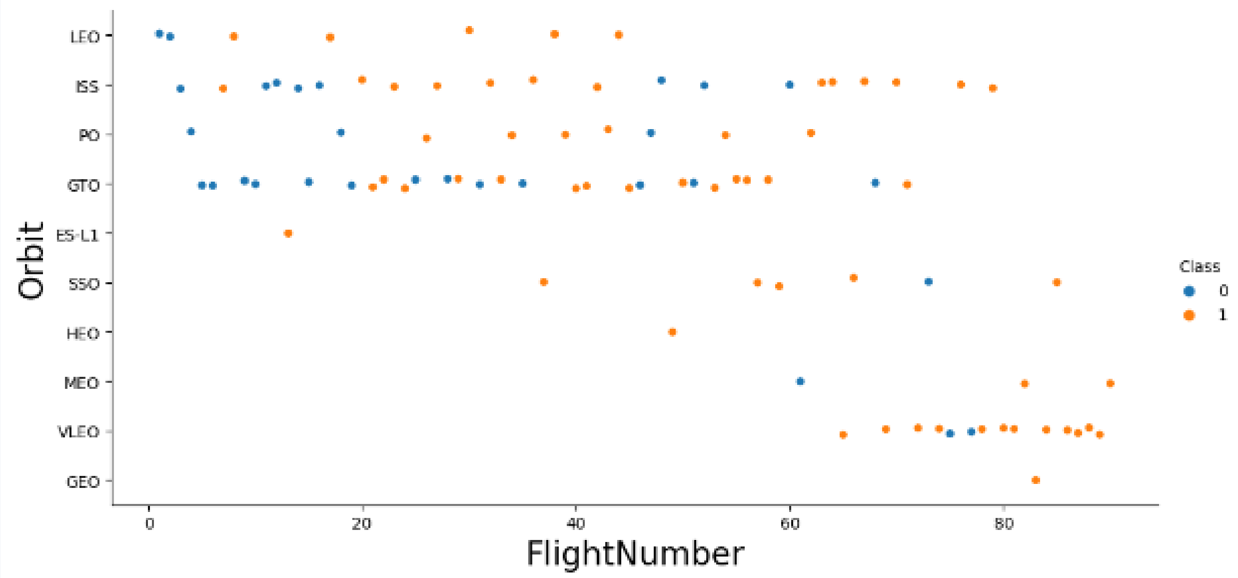
Success Rate vs. Orbit Type

- Based on the bar chart, we can say that landing the success rate of launching to ES-L1, GEO, and HEO orbits have highest success rate (100%) and the lowest success rates are found in GTO and ISS orbit launches (just above 50% and 60%, respectively).
- The success rate of the rest orbit launches are in the range between 65% to 85%.



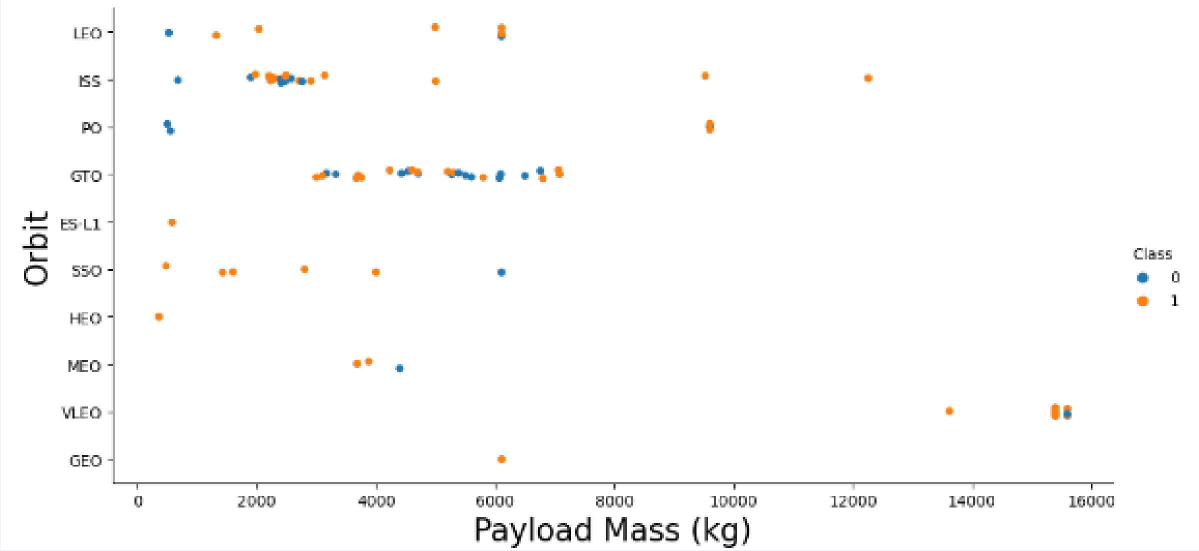
Flight Number vs. Orbit Type

- Based on the scatter plot, we can say that even ES-L1, GEO, and HEO have 100% success rate in landing, rockets are launched to these orbits just only a single time.
- On the other hand, most of the rockets are launched to GTO and ISS orbits.
- So, we can see that the reason why the landing success rate of launching to GTO and ISS orbits are at the lowest (just above 50% and 60%) is that they are mostly used orbits for launching rockets.



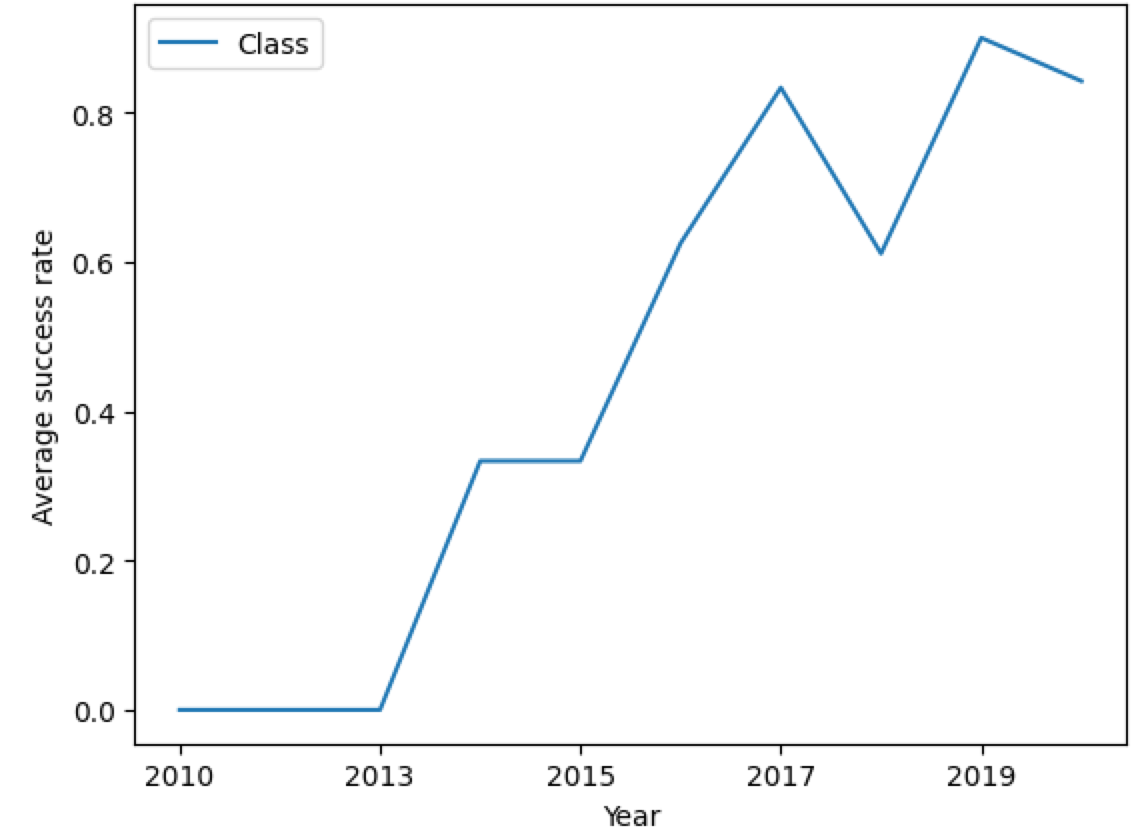
Payload vs. Orbit Type

- Based on the scatter plot, we can say, with heavy payloads the successful landing or positive landing rate are more for Polar (PO), LEO and ISS.
- However, for GTO we cannot distinguish the positive landing rate and negative landing rate, well because both are occurred unclearly in above 3000 and below 8000.



Launch Success Yearly Trend

- Based on the line chart, we can say that in the year between 2010 to 2013, there was no successful landing but since 2013, the successful landing rate kept increasing till to 2020.



All Launch Site Names

- There are four different Launch Sites in the data set that is used for EDA with SQL.
- There is an additional Launch Site in the dataset for EDA with SQL comparing with the dataset for Visualization using Matplotlib and Pandas and Building Predictive Model.
- Its name is "CCAFS LC-40".

```
In [14]: %%sql
SELECT DISTINCT(Launch_Site) FROM SPACEXTABLE;

 * sqlite:///my_data1.db
Done.

Out[14]:
```

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

Note:

- We did not consider "CCAFS LC-40" Launch Site, for EDA with Visualization using Matplotlib and Pandas and Building Predictive Model.
- The reason is that the locations of "CCAFS LC-40" and "CCAFS SLC-40" are so close and almost identical.
- So, considering the location difference between "CCAFS LC-40" and "CCAFS SLC-40" will not give any effect on predicting landing outcome.
- Thus why, we do not need to consider the data that occurred in "CCAFS LC-40" Launch Site in finding insights with Visualization and building model for predicting landing outcomes.

Total Payload Mass

- This query is performed to find the total payload mass carried by boosters from NASA.
- It shows there are 45596 kg for total payload mass carried by the boosters from NASA within 10 years (2010 to 2020).

```
In [13]: %%sql
SELECT Customer, SUM(PAYLOAD_MASS__KG_) FROM SPACEXTABLE
GROUP BY Customer
HAVING Customer = 'NASA (CRS)';

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

Out[13]:

Customer	SUM(PAYLOAD_MASS__KG_)
NASA (CRS)	45596

Launch Site Names Begin with 'CCA'

- This query is performed to get the data related with "CCAFS LC-40" and "CCAFS SLC-40" Launch Sites.

In [12]:
```
%%sql
SELECT * FROM SPACEXTABLE
WHERE Launch_Site LIKE 'CCA%'
limit 5;
```

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

Out[12]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-04-06	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-08-12	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
2012-08-10	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-01-03	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Average Payload Mass by F9 v1.1

- This query is performed to find the average payload mass carried by booster version F9 v1.1 .
- It shows the average payload mass carried by the booster version F9 v1.1 within 10 years (2010 to 2020) is 2928.4 kg.

```
In [14]: %%sql
SELECT Booster_Version, AVG(PAYLOAD_MASS__KG_) FROM SPACEXTABLE
GROUP BY Booster_Version
HAVING Booster_Version = 'F9 v1.1';

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

Out[14]:

Booster_Version	AVG(PAYLOAD_MASS__KG_)
F9 v1.1	2928.4

First Successful Ground Landing Date

- This query is performed to find the data of the first successful landing outcome was achieved in ground pad.

```
In [16]: %%sql
SELECT MIN(Date), Landing_Outcome FROM SPACEXTABLE
WHERE Landing_Outcome LIKE '%ground pad%';

 * sqlite:///my_data1.db
Done.

Out[16]:
```

MIN(Date)	Landing_Outcome
2015-12-22	Success (ground pad)

Successful Drone Ship Landing with Payload between 4000 and 6000

- This query is performed to find the names of the boosters which have success in drone ship but their payload mass must be between 4000 and 6000.

In [17]:
```
%%sql
SELECT Booster_Version, Landing_Outcome, PAYLOAD_MASS__KG_ FROM SPACEXTABLE
WHERE Landing_Outcome = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000;
```

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

Out[17]:

Booster_Version	Landing_Outcome	PAYLOAD_MASS__KG_
F9 FT B1022	Success (drone ship)	4696
F9 FT B1026	Success (drone ship)	4600
F9 FT B1021.2	Success (drone ship)	5300
F9 FT B1031.2	Success (drone ship)	5200

Total Number of Successful and Failure Mission Outcomes

- This query is performed to find the total number of successful and failure mission outcomes.

Note: There are two type of successful mission outcome, one is "Success" and the other one is "Success (payload status unclear)".

In [18]:
```
%%sql
UPDATE SPACEXTABLE
SET Mission_Outcome = TRIM(Mission_Outcome)
WHERE Mission_Outcome LIKE '%Success%';
```

```
 * sqlite:///my_data1.db
100 rows affected.
```

Out[18]: []

In [19]:
```
%%sql
SELECT Mission_Outcome, COUNT(Mission_Outcome) AS COUNT  FROM SPACEXTABLE
GROUP BY Mission_Outcome;
```

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

Out[19]:

Mission_Outcome	COUNT
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

Boosters Carried Maximum Payload

- This query is performed to find the names of the booster versions which have carried the maximum payload mass.

```
In [20]: %%sql
SELECT Booster_Version, PAYLOAD_MASS__KG_ FROM SPACEXTABLE
WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE);

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

Out[20]:

Booster_Version	PAYLOAD_MASS__KG_
F9 B5 B1048.4	15600
F9 B5 B1049.4	15600
F9 B5 B1051.3	15600
F9 B5 B1056.4	15600
F9 B5 B1048.5	15600
F9 B5 B1051.4	15600
F9 B5 B1049.5	15600
F9 B5 B1060.2	15600
F9 B5 B1058.3	15600
F9 B5 B1051.6	15600
F9 B5 B1060.3	15600
F9 B5 B1049.7	15600

2015 Launch Records

- This query is performed to find the records which contain the month names, booster version, landing outcomes in drone ship and launch site for the months in the year 2015.

```
In [21]: %%sql
SELECT substr(Date,6,2) AS Month, substr(Date,1,4) AS Year, Landing_Outcome, Booster_Version
FROM SPACEXTABLE
WHERE Year = '2015' AND Landing_Outcome = 'Failure (drone ship)';

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

Out[21]:

Month	Year	Landing_Outcome	Booster_Version
10	2015	Failure (drone ship)	F9 v1.1 B1012
04	2015	Failure (drone ship)	F9 v1.1 B1015

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

- This query is performed to find the count of each landing outcome between the date "2010-06-04" to "2017-03-20", in descending order.

In [22]:
```
%%sql
SELECT Landing_Outcome, COUNT(Landing_Outcome) AS Count, Date FROM SPACEXTABLE
GROUP BY Landing_Outcome
HAVING Date BETWEEN '2010-06-04' AND '2017-03-20'
ORDER BY Count DESC;
```

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

Out[22]:

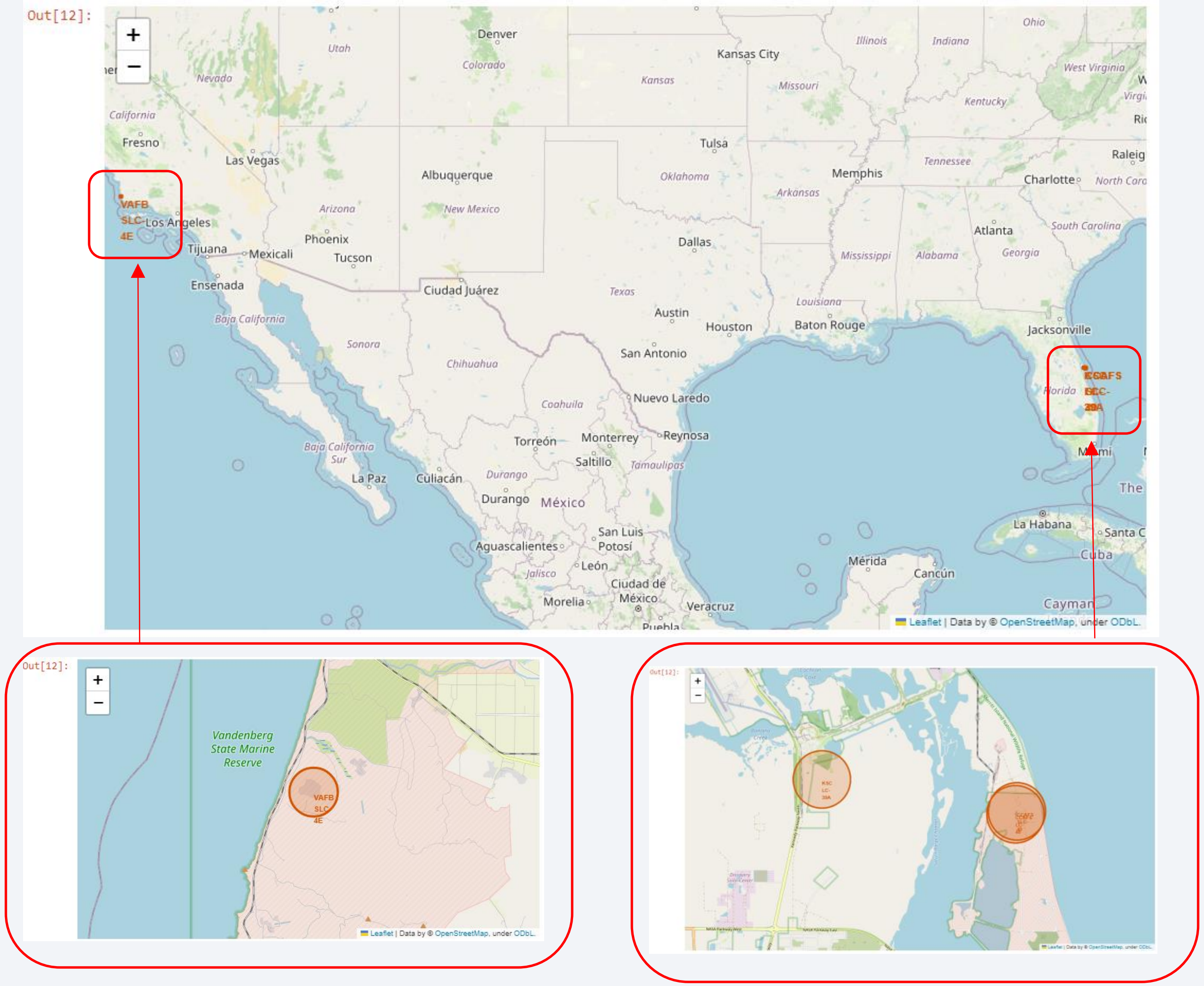
Landing_Outcome	Count	Date
No attempt	21	2012-05-22
Success (drone ship)	14	2016-08-04
Success (ground pad)	9	2015-12-22
Failure (drone ship)	5	2015-10-01
Controlled (ocean)	5	2014-04-18
Uncontrolled (ocean)	2	2013-09-29
Precluded (drone ship)	1	2015-06-28

Section 3

Launch Sites Proximities Analysis

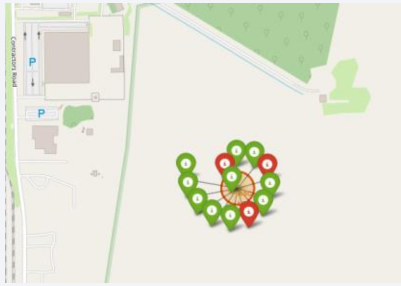
Map with all launch sites marks

- By visualizing the markers on the map, we can say that VAFB SLC 4E, CCAFS LC-40, and CCAFS SLC-40 are very close proximity to coasts but KSC LC-39A is not very close to coast. However, All of them are not in proximity to the Equator line.
- An interesting geographical observation arises from the SpaceX Falcon 9 launch sites: CCAFS LC-40, CCAFS SLC-40, and KSC LC-39A are situated in close proximity to each other. In contrast, VAFB SLC-4E stands notably apart from this them, exhibiting a significant separation in both latitude and longitude.



Map with success/failed launches marks for each site

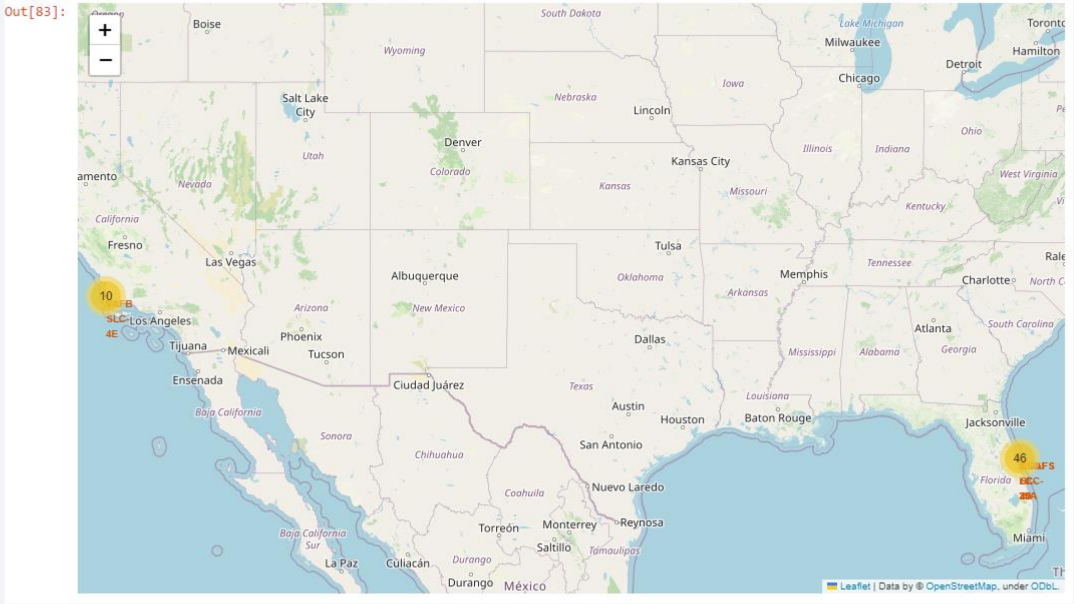
VAFB SLC 4E



KSC LC-39A

CCAFS LC-40

CCAFS SLC-40



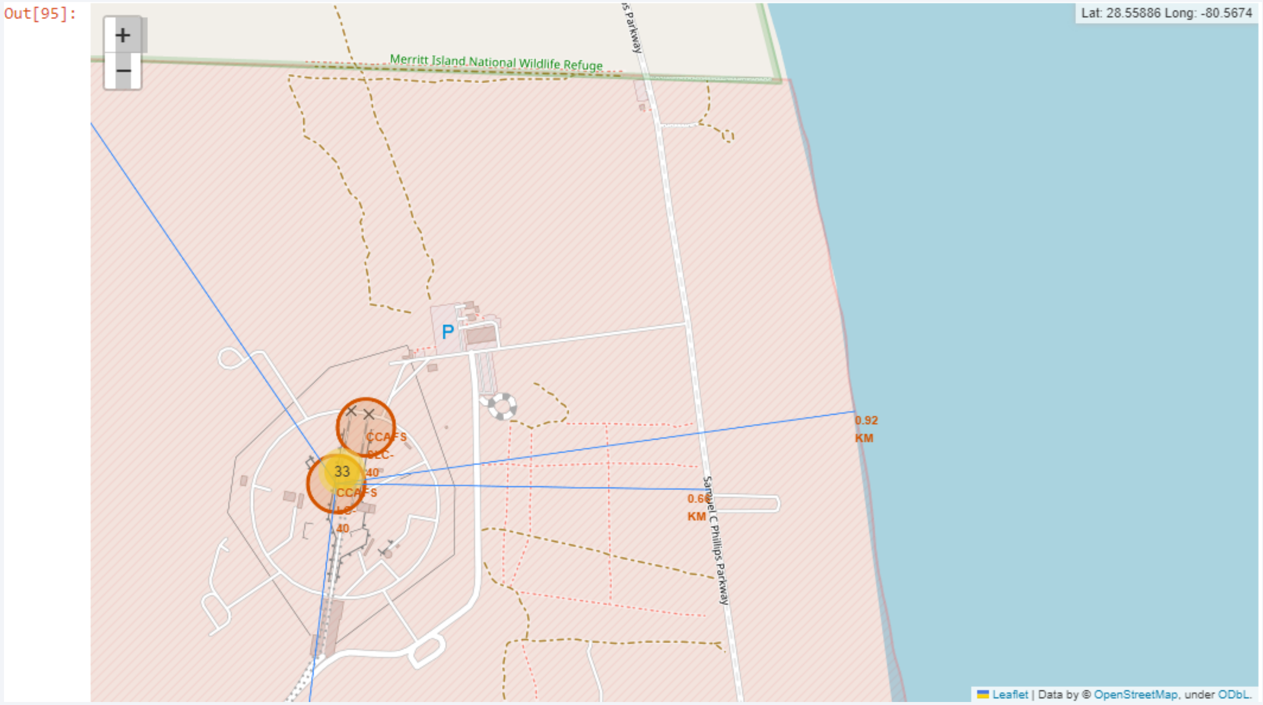
- Based on the marks' color, we can say that KSC LC-39A Launch Site has the most successful launching rate.
- But the most launch are done in CCAFS LC-40 Launch Site.

Green mark = Successful launch
Red mark = Fail launch or other condition

Distance between a selected launch site and its proximities

- We also like to know the distance between CCAFS LC-40 Launch Site and its closest coastline point, railway, highway, and city.
- This map shows the distance between:

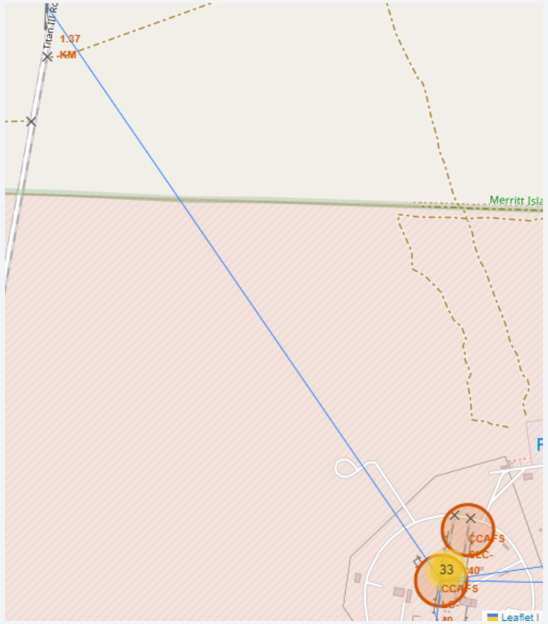
1. CCAFS LC-40 and closest coastline point (coordinate: [28.56345, -80.56798])
2. CCAFS LC-40 and NASA railway (coordinate: [28.57249, -80.58525])
3. CCAFS LC-40 and Samuel C Phillips Pkwy highway (coordinate: [28.56221, -80.57061])
4. CCAFS LC-40 and Melbourne (coordinate: [28.10605, -80.64016])



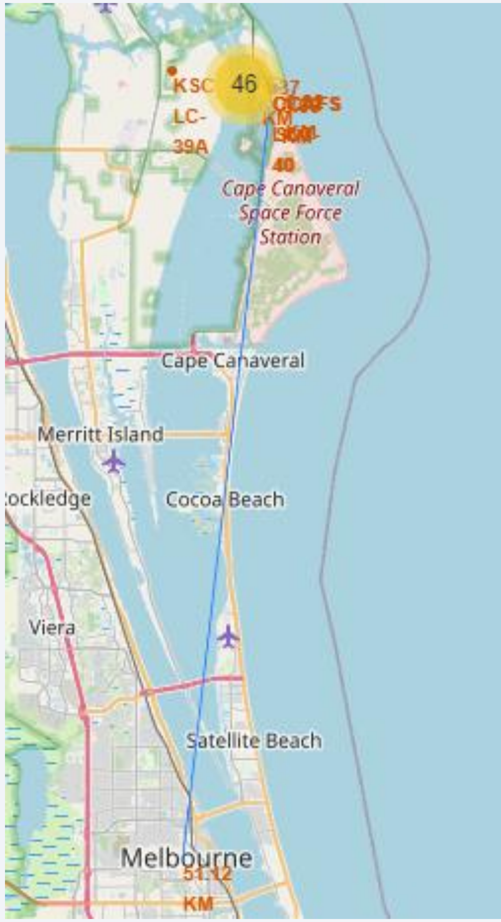
Distance between a selected launch site and its proximities

Distance between CCAFS LC-40 and closest coastline point = 0.92 km

Distance between CCAFS LC-40 and Samuel C Phillips Pkwy highway = 0.66 km



Distance between CCAFS LC-40 and NASA railway = 1.37 km



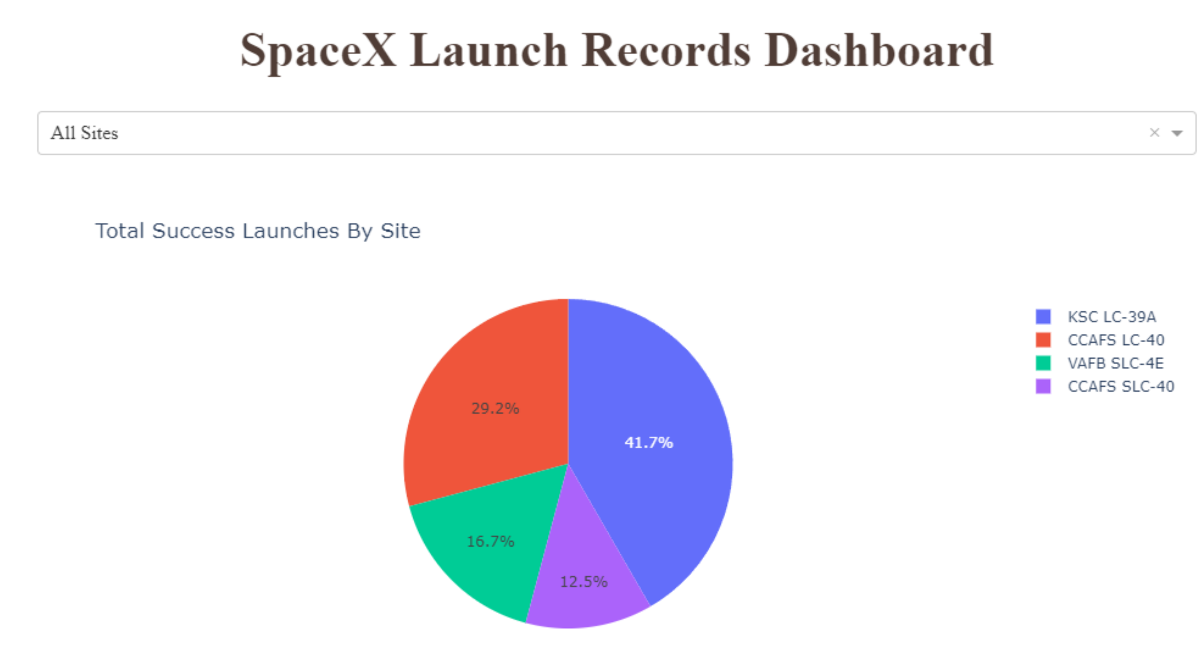
Distance between CCAFS LC-40 and Melbourne =51.12 km

Section 4

Build a Dashboard with Plotly Dash

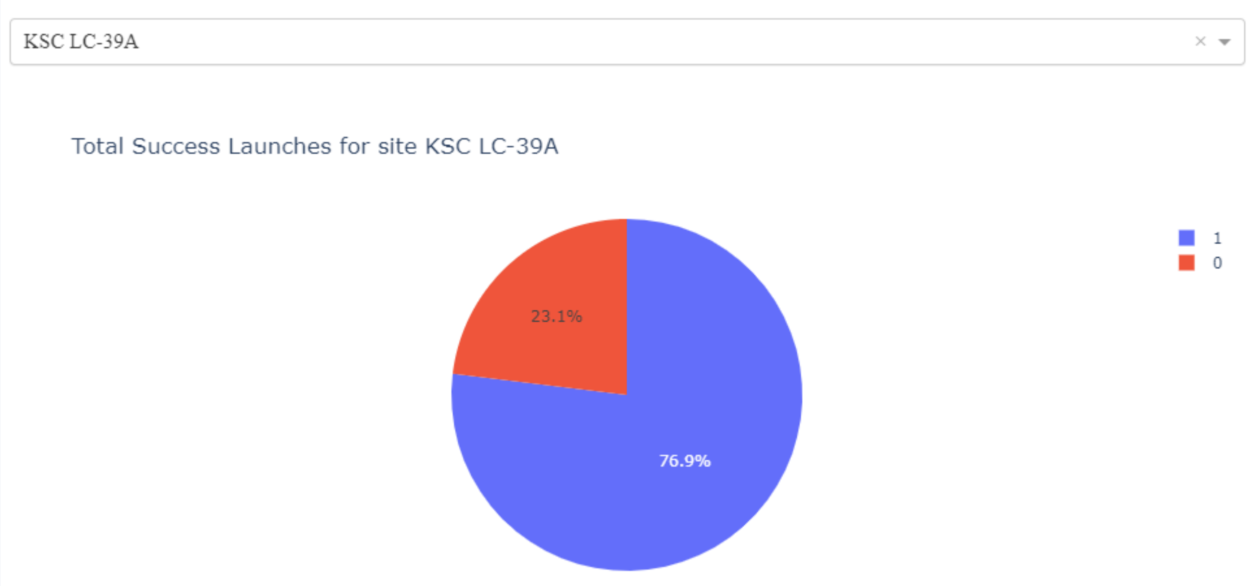
Launch Success Count in Percent for All Site

- Based on the Pie Chart, we can say that KSC LC-39A has the most successful launch count in the percentage with 41.7% .
- On the other hand, CCAFS SLC-40 has the least successful launch count in the percentage with 12.5%.



Launch Success Count in Percent for KSC LC-39A

- In the pie chart, the part in blue color shows the percentage of successful launches and the part in red color shows the percentage of failure launches.
- If we select KSC LC-39A, which has the highest success in launch count, we can see that more than three quarters of the launch counts are success launches.



Correlation Between Payload Mass and Success for All Sites

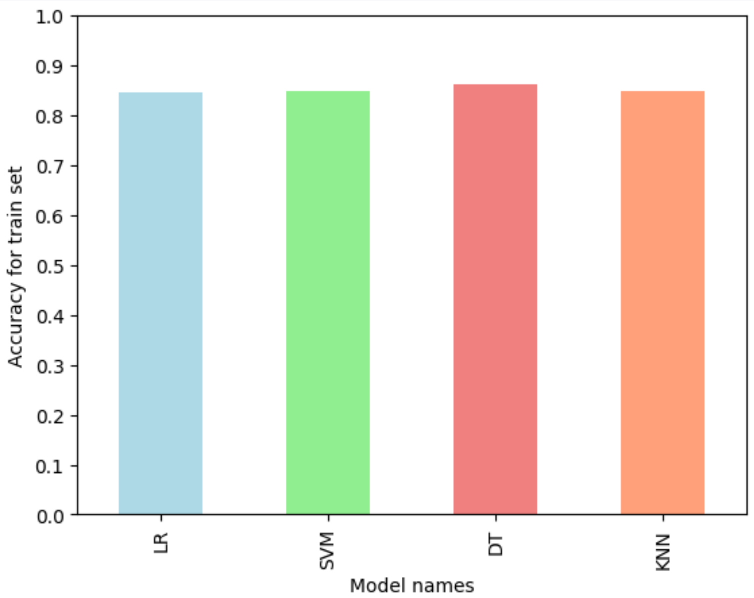
- Based on the scatter plot, we can say that in all of the launch sites, FT booster type has more successful launch than any other booster type.
- On the contract, least successful launch is occurred in v1.1 booster type.

Section 5

Predictive Analysis (Classification)

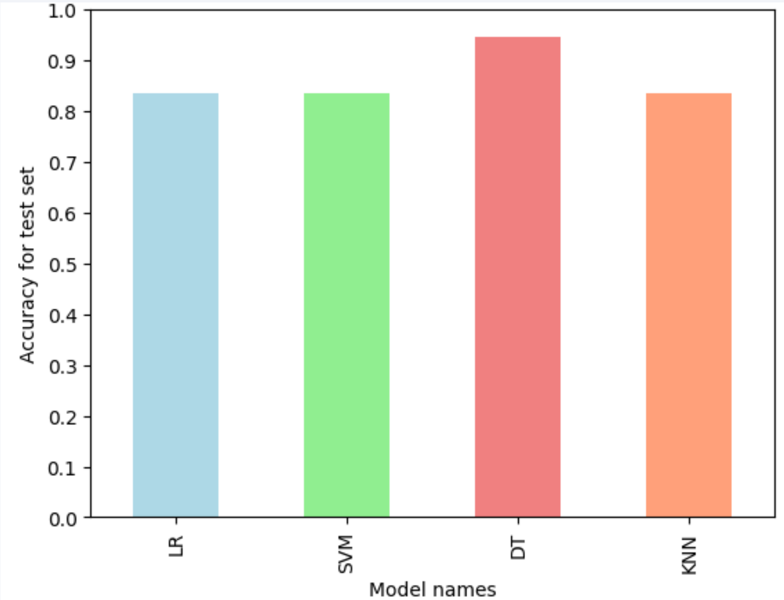
Classification Accuracy

Accuracy for Training data



	Model names	Accuracy_train
0	LR	0.846429
1	SVM	0.848214
2	DT	0.860714
3	KNN	0.848214

Accuracy for Testing data



	Model names	Accuracy_test
0	LR	0.833333
1	SVM	0.833333
2	DT	0.944444
3	KNN	0.833333

Models

LR = Logistic Regression

SVM = Support Vector Machine

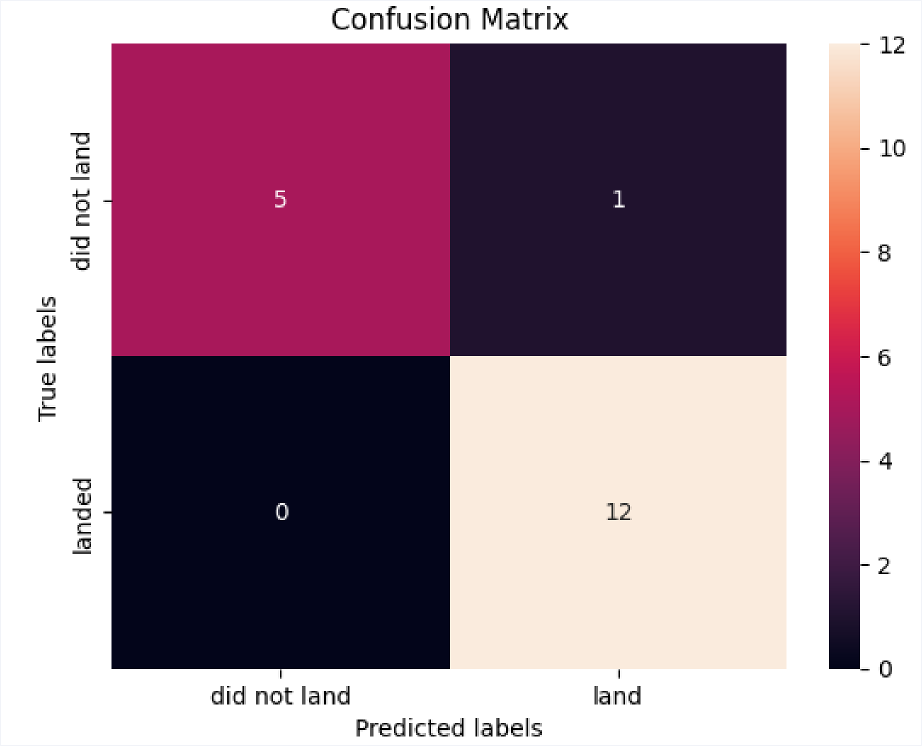
DT = Decision Tree

KNN = K nearest neighbors

For both train and test sets, we can see that Decision Tree Classification model gives the best accuracy result.

Confusion Matrix

The best performing model for this project is Decision Tree Classification model and its confusion matrix is as follows:



TP = 12, TN = 5, FP = 1, FN = 0

- Based on the confusion matrix, we can say that our Decision Tree Classification model did not produce any false negatives, meaning that it correctly classified all instances that belong to positive class in actual data set.
- For false positive, our model incorrectly predicted only one instance that belong to negative class in actual data set.
- In summary, this is a good outcome because our model give only one wrong prediction for negative class in actual data set.

Conclusions

Based on the above findings by using Falcon 9 data of SpaceX, we can summarize as followings:

For Orbit type and Payload Mass with respect to landing rate

- The orbits, which are mostly used to launch the rockets, are GTO and ISS with the landing success rate at just above 50% and 60%, respectively.
- For correlation between Payload Mass and Orbit Type, with heavy payloads, the successful landing or positive landing rate are increased for Polar(PO), LEO, and ISS orbits.

For Launch Site with respect to launching rate

- KSC LC-39A launch site has the most successful launching rate with the percentage of 41.7% in all of the Launch Sites' successful launching rate.
- For individual, KSC LC-39A has 76.9% successful launching rate and 23.1% failure launching rate.

Developing Model for landing outcomes

- Within attempted classification models, we select Decision Tree Classifier for predicting landing outcomes due to its higher accuracy value in training set and testing set.

Appendix

- The following GitHub URL contains all the tasks that I did for this project.
- It contains Python code in Jupyter notebooks and datasets that I used for this project.

GitHub URL:

https://github.com/thetkhinelin25/Data-Science-Capstone-Project.git

Thank you!