IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Zihao Gu

2025.01.09

Outline

1. Executive Summary
2. Introduction
3. Methodology
4. Results
5. Conclusion
6. Appendix

1. Executive Summary

1. Executive Summary

- Summary of methodologies
  - 1. Data Collection
  - 2. Data Webscraping
  - 3. Data Wrangling
  - 4. Exploratory Data Analysis with SQL
  - 5. Exploratory Data Analysis with Data Visualization
  - 6. Building an interactive map with Folium
  - 7. Building a Dashboard with Plotly Dash
  - 8.Predictive Classification Analysis

- Summary of all results
  - Exploratory Data Analysis results
  - Interactive analytics demo in screenshots
  - Predictive analysis results

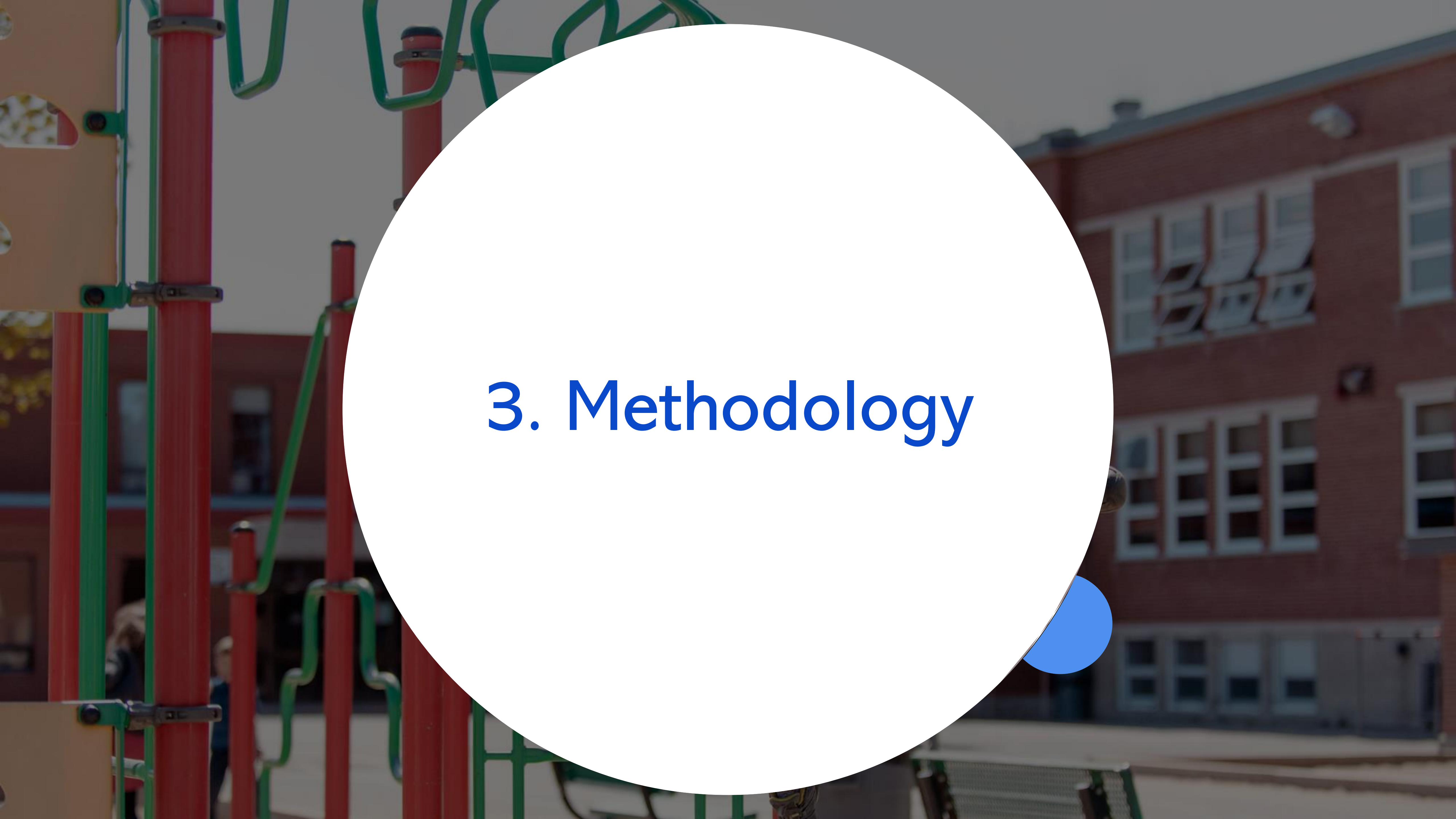
2. Introduction

2. Introduction

- background and context
  - SpaceX is the most successful company of the commercial space age, making space travel affordable.
  - The company advertises Falcon9 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. Based on public information and machine learning models, we are going to predict if SpaceX will reuse the first stage.

- Questions to be answered
  - How do variables such as payload mass, launch site, number off lights, and orbits affect the success of the first stage landing?
  - Does the rate of successful landings increase over the years?
  - What is the best algorithm that can be used for binary classification in this case?

3. Methodology



3. Methodology – 1.Data Collection API

In this capstone project, we aim to predict whether the Falcon 9 first stage will land successfully.

SpaceX promotes its Falcon 9 rocket launches at a cost of $62 million, which is significantly lower than other providers, whose prices start at over $165 million per launch. This cost savings is largely due to SpaceX's ability to reuse the first stage.

We can estimate the overall cost of a launch by predicting whether the first stage will land successfully. This information could be valuable for competing companies looking to bid against SpaceX for rocket launches.

I gathered and formatted the data appropriately using an API.

3. Methodology – 1.Data Collection API

Requesting rocket launch data from **SpaceX API**;

Decoding the response content using **.json()**

Using **.json_normalize()** to turn it into a dataframe

Requesting information about the launches from **SpaceX API** by applying custom functions

Constructing data we have obtained into a dictionary using **dict.fromkeys()**

Creating a dataframe from the dictionary using **DataFrame()**

Filtering the dataframe to only include Falcon 9 launches

Replacing missing values of Payload Mass column with calculated **.mean()** for this column

Exporting the data to CSV file, using **.to_csv()**

GitHub URL: 1-SpaceX-data-collection-api.ipynb

3. Methodology – 2.Data Webscraping

Requesting Falcon 9 launch data from Wikipedia

Creating a BeautifulSoup object from the HTML response

Extracting all column names from the HTML table header

Collecting the data by parsing HTML tables

Constructing data we have obtained into a dictionary

Creating a dataframefrom the dictionary

Exporting the data to CSV file, using **.to_csv()**

GitHub URL: 2-SpaceX-webscraping.ipynb

3. Methodology – 3.Data Wrangling

Perform exploratory data analysis and determine Training Labels

Calculate the number of launches on each site

Calculate the number and the occurrence of each orbit

Calculate the number and occurrence of mission outcome per orbit type

Create a landing outcome label from Outcome column

Exporting the data to CSV file, using **.to_csv()**

GitHub URL: 3-SpaceX-DataWrangling.ipynb

3. Methodology – 4.EDA SQL

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 for the months in year 2015

Ranking the count of landing outcomes (such as Failure or Success) between the date2010-06-04 and 2017-03-20 in descending order

GitHub URL: 4-SpaceX-EDA-sqllite.ipynb

3. Methodology – 5.EDA Visualization

Charts plotted:

Flight Number vs. Payload Mass, Flight Number vs. Launch Site, Payload Mass vs. Launch Site, Orbit Type vs. Success Rate, Flight Number vs. Orbit Type, Payload Mass vs Orbit Type and Success Rate Yearly Trend

Scatter plots show the relationship between variables.

If any kind of relationship exist, they could be used in machine learning model.

Bar charts show comparisons among discrete categories.

The goal is to show the relationship between the specific categories being compared and a measured value.

Line charts show trends in data over time (time series)

You can plot a line chart with x axis to be Year and y axis to be average success rate, to get the average launch success trend.

To select the features that be used in success prediction:

Apply OneHotEncoder to the designated columns. Assign the value to the variable features_one_hot , display the results using the method head.

GitHub URL: 5-SpaceX-EDA-DataVisualization.ipynb

3. Methodology – 6.Build an interactive map with Folium

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.

Colored Markers of the launch outcomes for each Launch Site:

Added colored 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 colored Lines to show distances between the Launch Site KSC LC-39A and its proximities like Railway, Highway, Coastline and Closest City.

Draw a PolyLine between a launch site to the selected

GitHub URL: 6-SpaceX_launch_site_location.ipynb

3. Methodology – 7. Building a Dashboard with Plotly Dash

TASK 1: Add a dropdown list to enable Launch Site selection.

Using **dcc.Dropdown()** to add the dropdown list.

Using "options" to list all values

TASK 2: Add a pie chart to show the total successful launches count.

Using the **dcc.Graph()** to show the Success vs. Failed counts.

Using **@app.callback** for 'site-dropdown' as input, 'success-pie-chart' as output.

TASK 3: Add a slider to select payload range.

Using **dcc.RangeSlider()** to set up all the attributes for the payload in the range.

TASK 4: Add a scatter chart to show correlation between payload and launch success.

Using **html.Div()** to add the scatter chart.

Using **@app.callback** to set up inputs and outputs of payload slider.

Using **app.run_server()** to run the above tasks on the allocated web.

GitHub URL: 7-SpaceX_dash_app.py

3. Methodology – 8.ML Prediction

Creating a NumPy array from the column "Class" in data.

Standardizing the data with Standard Scaler, then fitting and transforming it.

Splitting the data into training and testing sets with train_test_split function.

Creating a GridSearchCV object with cv = 10 to find the best parameters.

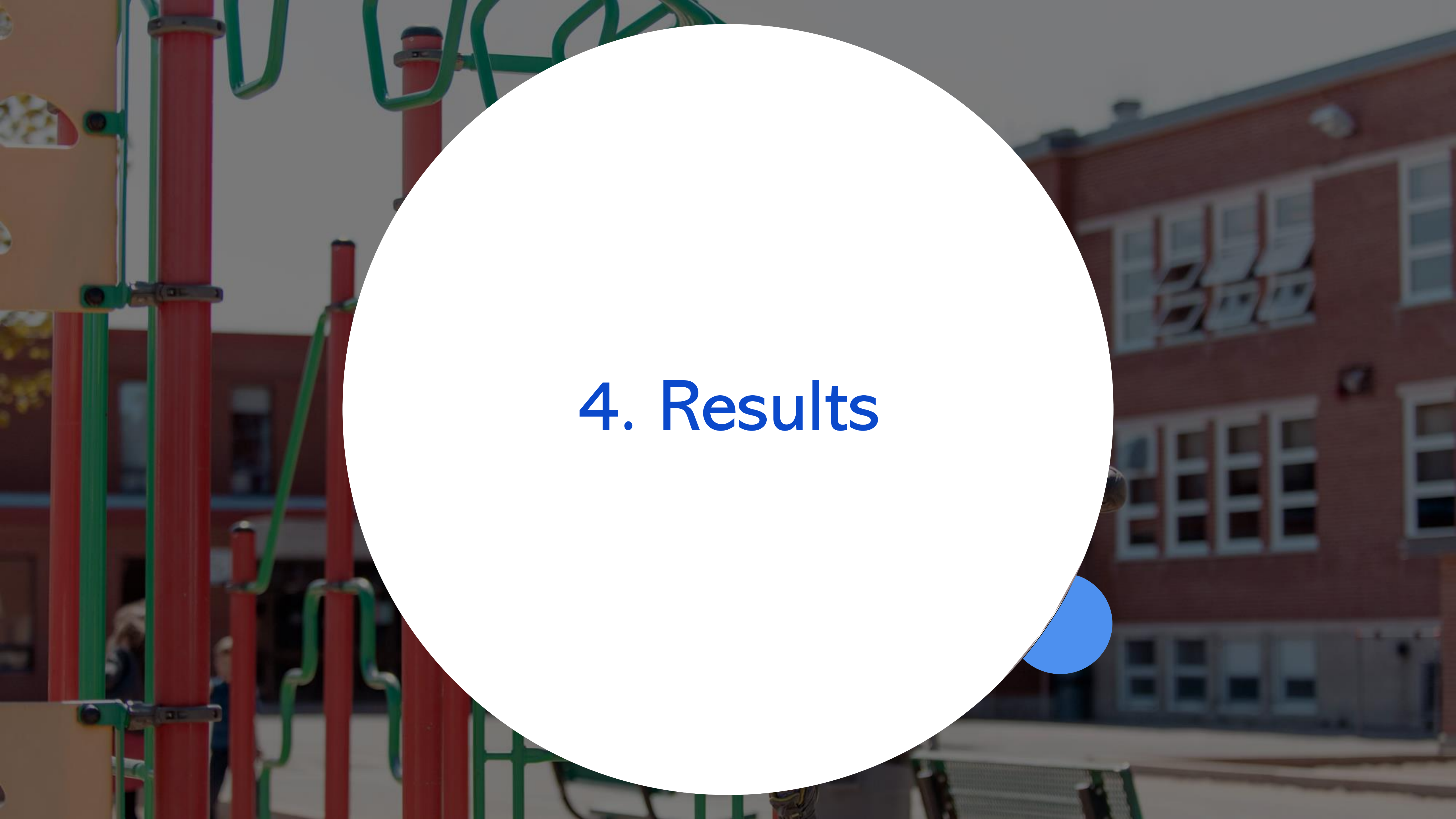
Applying GridSearchCV on LogReg, SVM, Decision Tree, and KNN models.

Calculating the accuracy on the test data using the method .score() for all models

Examining the confusion matrix for all models

Finding the method performs best by examining the Jaccard_score and F1_score metrics

GitHub URL: 8-SpaceX-ML_Prediction.ipynb



4. Results

4. Results

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

4. Results – 4.EDA with SQL

TASK 1: **Display the names of the unique launch sites in the space mission**

Launch_Site
1. CCAFS LC-40
2. VAFB SLC-4E
3. KSC LC-39A
4. CCAFS SLC-40

Task 1

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

```
%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

4. Results – 4.EDA with SQL

Task 2
Display 5 records where launch sites begin with the string 'CCA'

Task 2

Display 5 records where launch sites begin with the string '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
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO
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)
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)

4. Results – 4.EDA with SQL

Task 3
Display the total payload mass carried by boosters launched by NASA (CRS)

1. The sum of total payload mass carried by boosters which were launched by NASA is: 619967.

Task 3

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

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL;
```

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

SUM(PAYLOAD_MASS__KG_)
619967

4. Results – 4.EDA with SQL

Task 4
Display average payload mass carried by booster version F9 v1.1

The average payload mass carried by booster version which like "F9 v1.1" is: 2928.4

Task 4

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

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE Booster_Version LIKE "F9
```

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

AVG(PAYLOAD_MASS__KG_)
2928.4

4. Results – 4.EDA with SQL

Task 5
List the date when the first succesful landing outcome in ground pad was acheived.

Task 5

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

Hint:Use min function

```
%sql SELECT MIN(DATE) FROM SPACEXTBL WHERE LANDING_OUTCOME="Success (ground pad)";
```

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

MIN(DATE)
2015-12-22

4. Results – 4.EDA with SQL

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

Task 6

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

```
%sql select Booster_Version from SPACEXTBL where "Landing_Outcome" = 'Success (drone ship)' and
```

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

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

4. Results – 4.EDA with SQL

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

Task 7

List the total number of successful and failure mission outcomes

```
%sql select mission_outcome, count(*) as total_number from SPACEXTBL group by mission_outcome;
```

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

Mission_Outcome	total_number
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

4. Results – 4.EDA with SQL

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

Task 8

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

```
%sql select booster_version from SPACEXTBL where payload_mass__kg_ = (select max(payload_mass_
```

```
 * 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

4. Results – 4.EDA with SQL

Task 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.

Task 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.

Note: SQLLite 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.

```
%%sql select substr(Date, 6,2) as month, Date, Booster_Version, launch_site, Landing_Outcome fı
      where "Landing_Outcome" = 'Failure (drone ship)' and  substr(Date,0,5)='2015';
```

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

month	Date	Booster_Version	Launch_Site	Landing_Outcome
01	2015-01-10	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
04	2015-04-14	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

4. Results – 4.EDA with SQL

Task 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.

Task 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.

```
%%sql select "Landing_Outcome", count(*) as count_outcomes from SPACEXTBL
      where date between '2010-06-04' and '2017-03-20'
      group by "Landing_Outcome"
      order by count_outcomes desc;
```

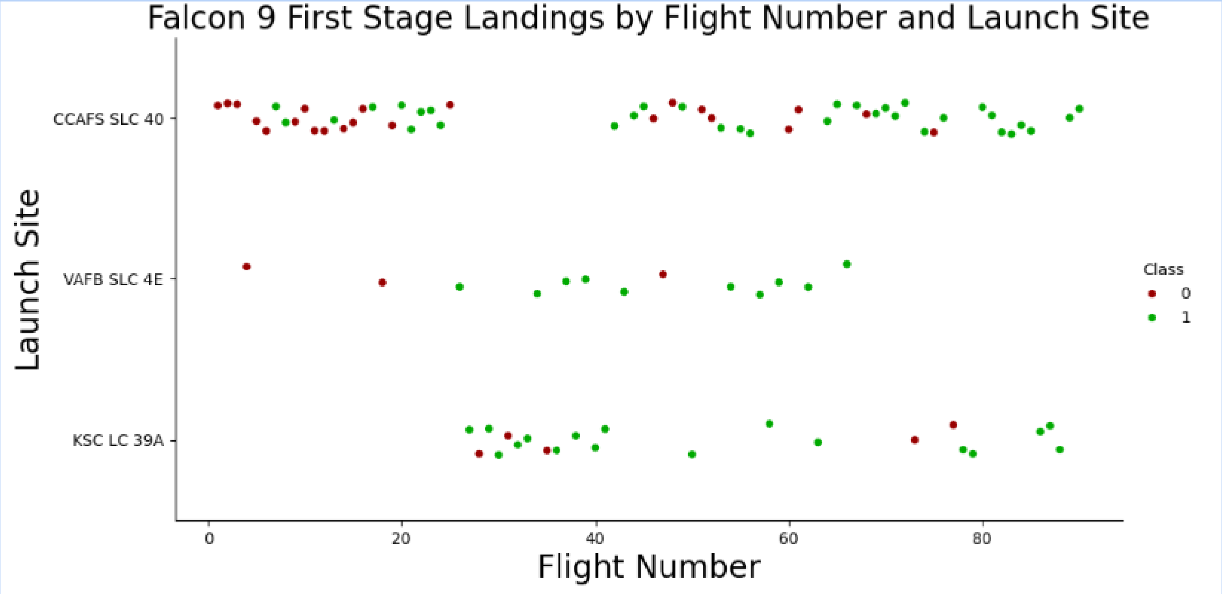
```
 * sqlite:///my_data1.db
Done.
```

Landing_Outcome	count_outcomes
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

4. Results – 5.EDA Visualization

TASK 1: Visualize the relationship between **Flight Number and Launch Site**

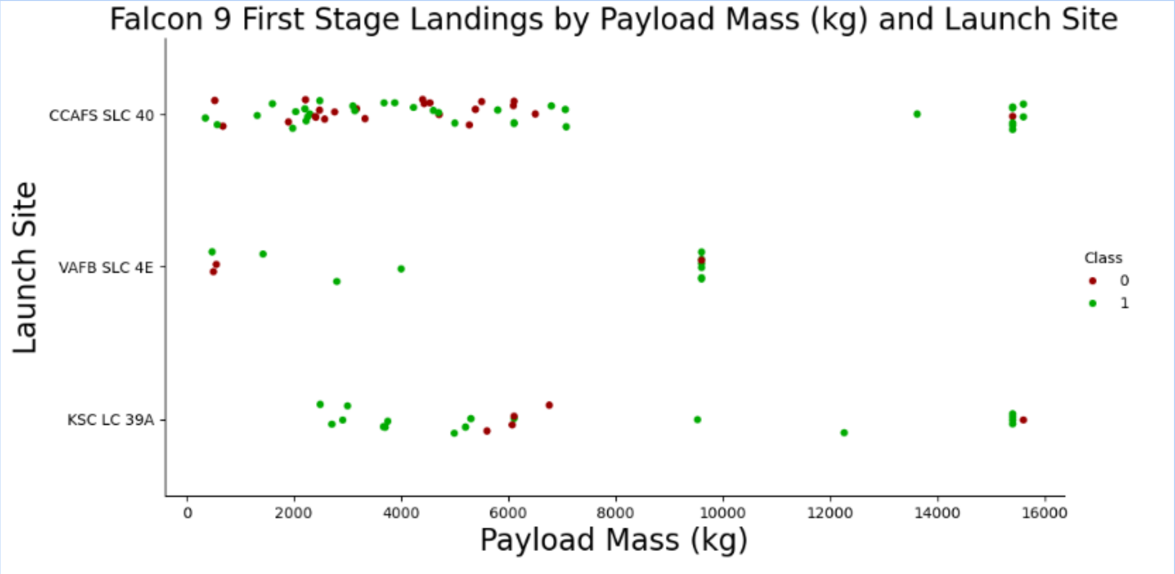
1. The earliest lights all failed while the latest lights all succeeded.
2. The CCAFS SLC 40 launch site has about a half of all launches.
3. VAFB SLC 4E and KSC LC 39A have higher success rates.
4. It can be assumed that each new launch has a higher rate of success.



4. Results – 5.EDA Visualization

TASK 2: Visualize the relationship between **Payload Mass and Launch Site**

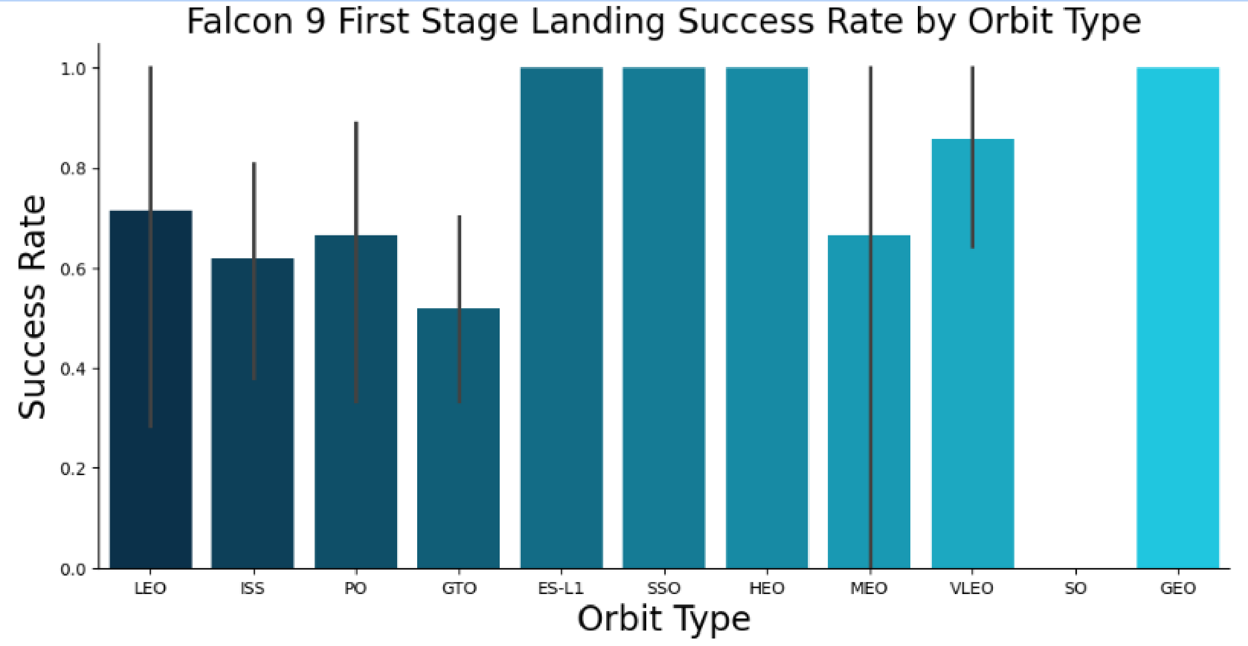
1. For every launch site the higher the payload mass, the higher the success rate.
2. Most of the launches with payload mass over 7000 kg were successful.
3. KSC LC 39A has a 100% success rate for payload mass under 5500 kg too.



4. Results – 5.EDA Visualization

Task3: Visualize the relationship between **success rate of each orbit type**

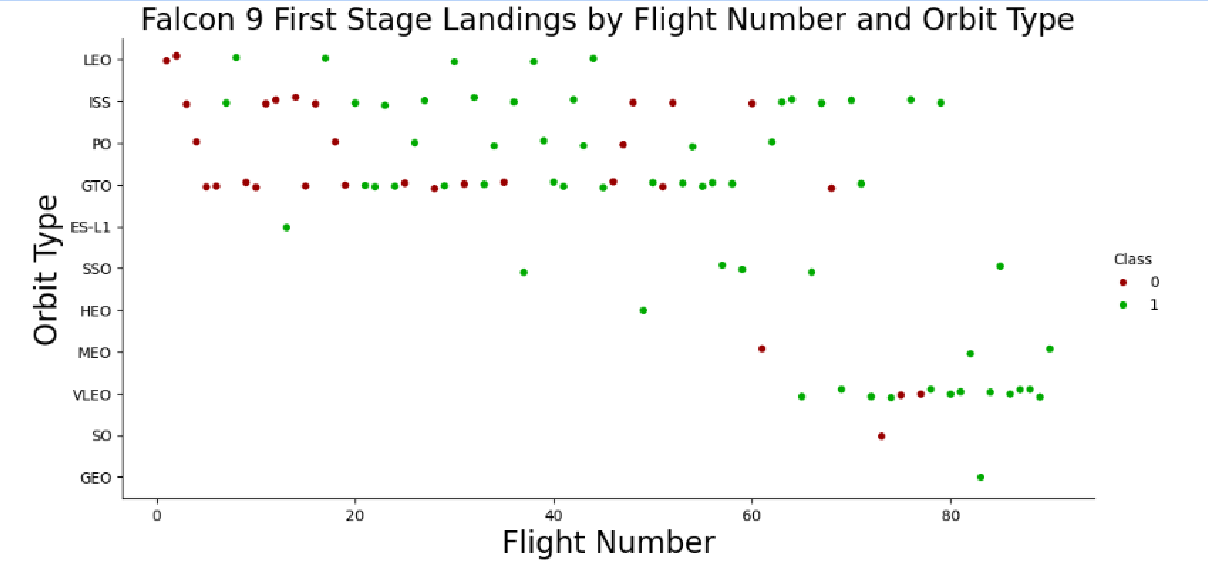
1. Orbits with 100% success rate: - ES-L1, GEO, HEO, SSO
2. Orbits with 0% success rate: - SO
3. Orbits with success rate between 50% and 85%: - GTO, ISS, LEO, MEO, PO



4. Results – 5.EDA Visualization

Task4: Visualize the relationship between **Flight Number and Orbit type**

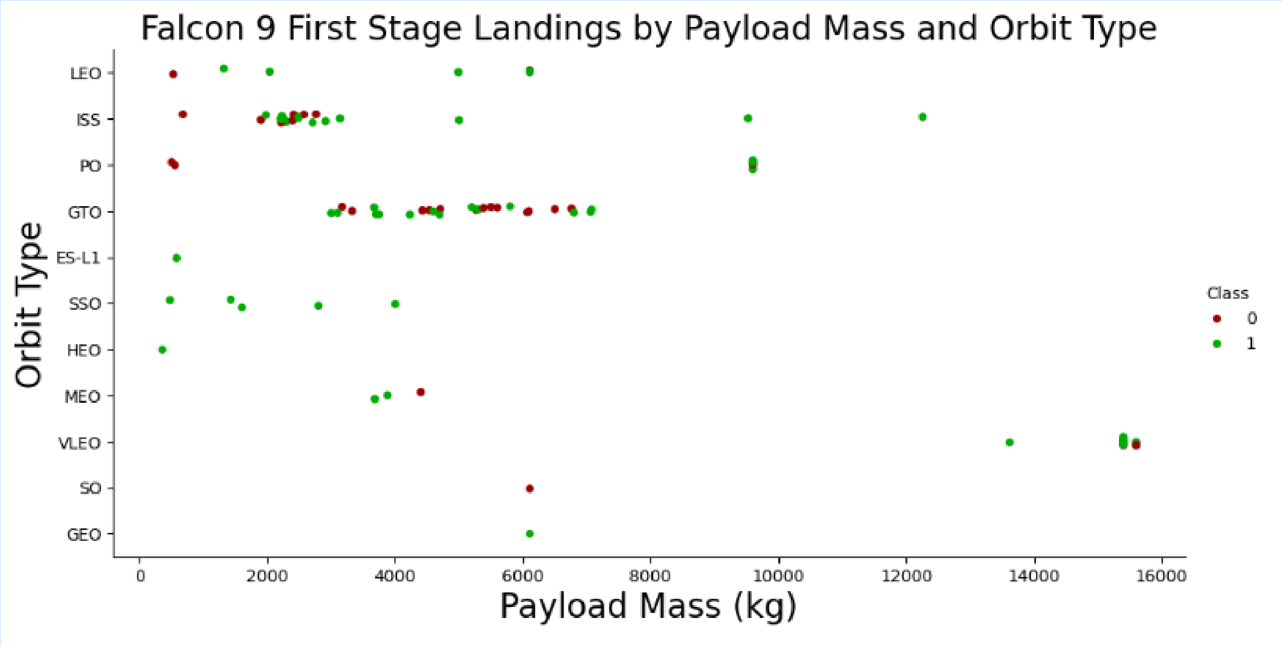
1. Orbits with 100% success rate: - ES-L1, GEO, HEO, SSO
2. Orbits with 0% success rate: - SO
3. Orbits with success rate between 50% and 85%: - GTO, ISS, LEO, MEO, PO



4. Results – 5.EDA Visualization

TASK 5: Visualize the relationship between **Payload Mass and Orbit type**

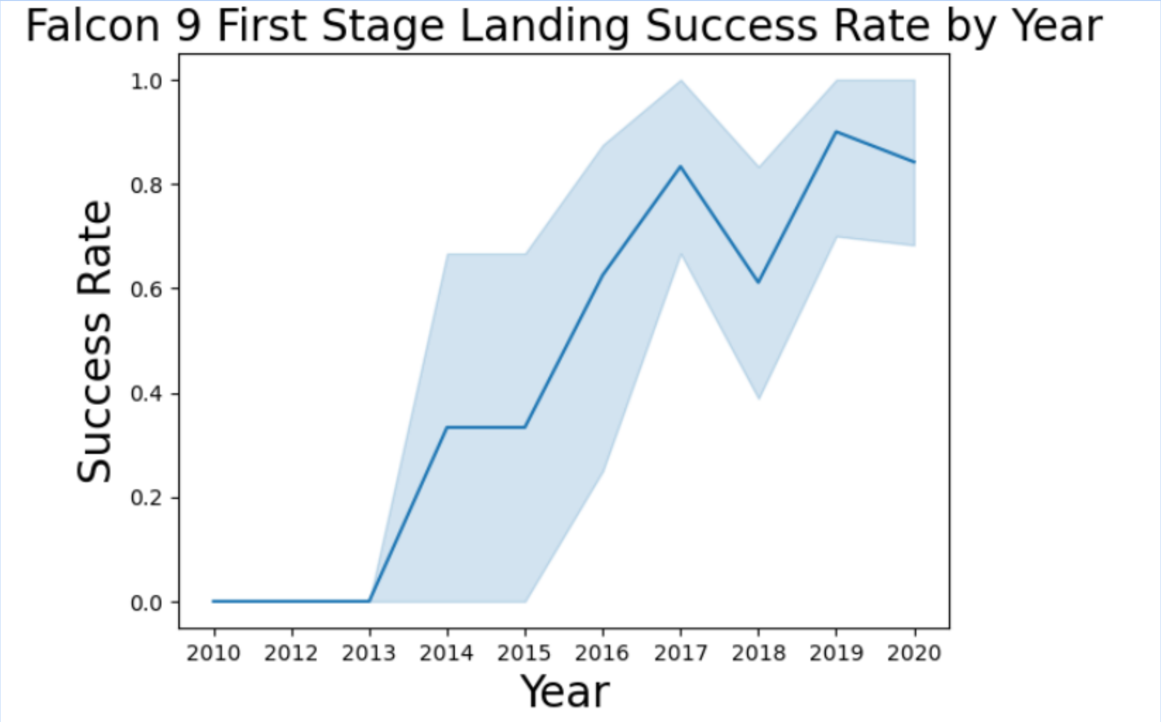
1. Heavy payloads have a negative influence on GTO orbits
2. positive on GTO and Polar LEO (ISS) orbits.



4. Results – 5.EDA Visualization

TASK 6: Visualize the launch success yearly trend

1. The success rate since 2013 kept increasing till 2020.
2. The success rate has a significant decrease on 2018.



4. Results – 6. Sites Locations Analysis with Folium

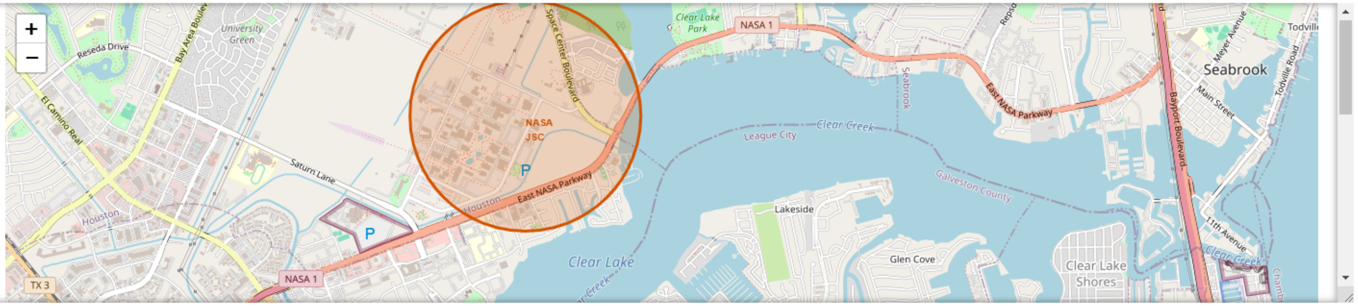
Task 1: Mark all launch sites on a map

1. From the given .csv file, we can find the latitude and longitude for each site.
2. Using folium.Circle() to create the area on the displayed map.

We could use `folium.Circle` to add a highlighted circle area with a text label on a specific coordinate. For example,

```
[12]: # Create a blue circle at NASA Johnson Space Center's coordinate with a popup label showing its name
circle = folium.Circle(nasa_coordinate, radius=1000, color='#d35400', fill=True).add_child(folium.Popup('NASA Johnson Space Center'))
# Create a blue circle at NASA Johnson Space Center's coordinate with a icon showing its name
marker = folium.map.Marker(
    nasa_coordinate,
    # Create an icon as a text label
    icon=DivIcon(
        icon_size=(20,20),
        icon_anchor=(0,0),
        html='<div style="font-size: 12; color:#d35400;"><b>%s</b></div>' % 'NASA JSC',
        )
    )
site_map.add_child(circle)
site_map.add_child(marker)
```

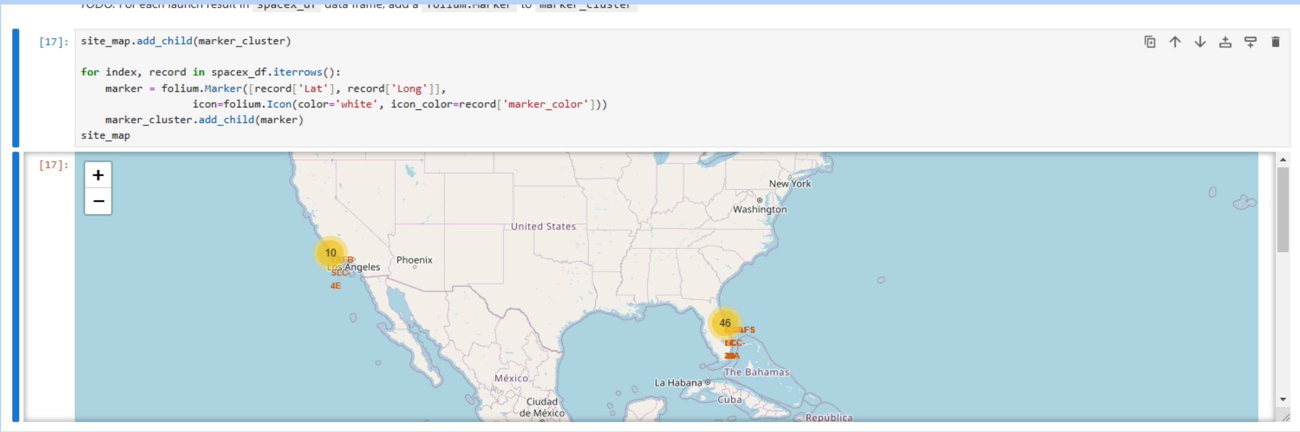
[12]:

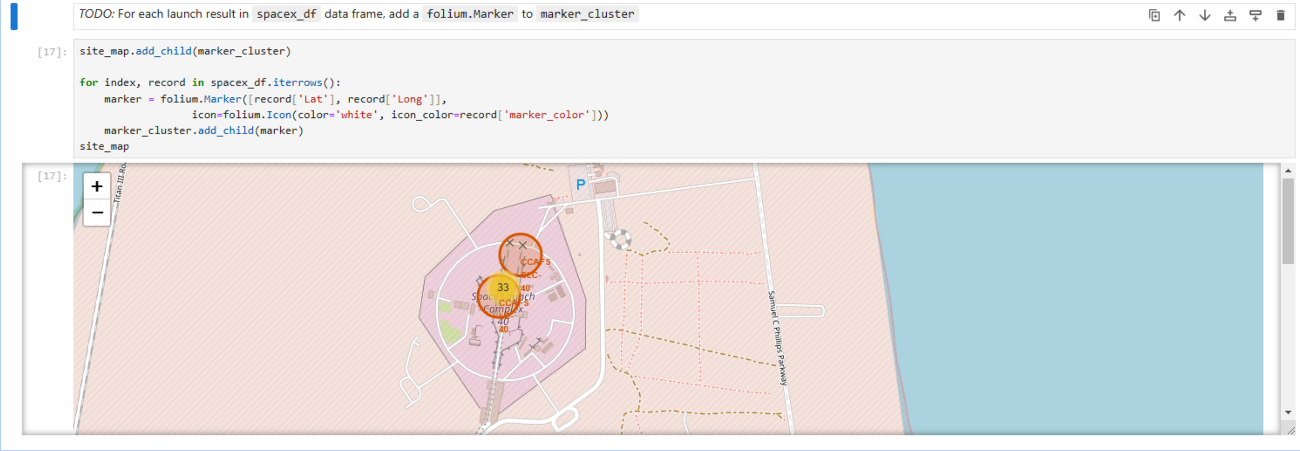


4. Results – 6. Sites Locations Analysis with Folium

Task 2: Mark the success/failed launches for each site on the map

1. Create a new column in the dataframe called marker_color to store the marker colors based on the class value.
2. For each launch result in spacex_df data frame, add a folium.Marker to marker_cluster.

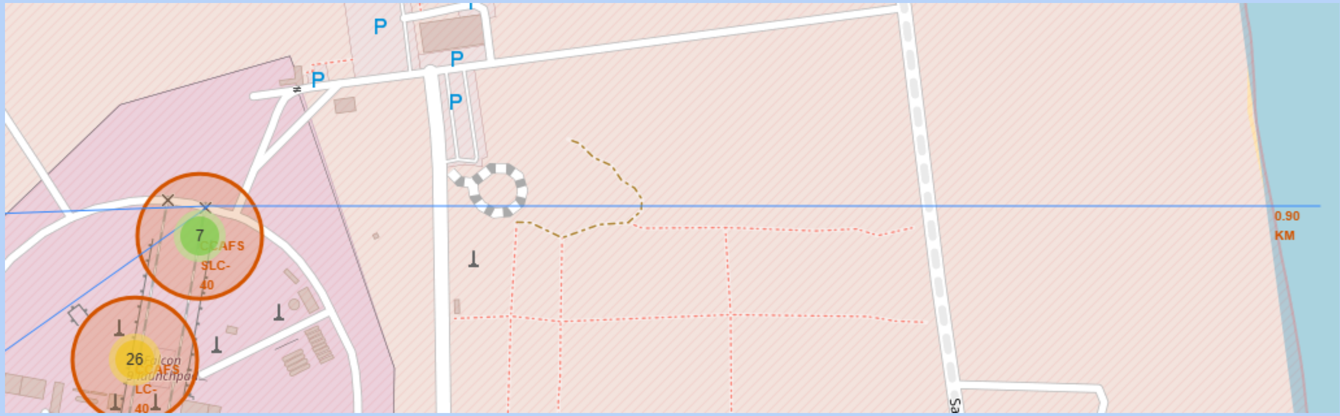


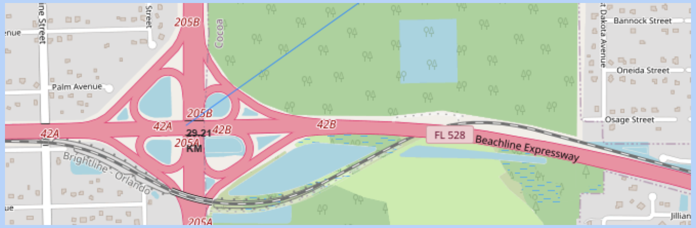


4. Results – 6. Sites Locations Analysis with Folium

TASK 3: Calculate the distances between a launch site to its proximities

1. Add a MousePosition on the map to get coordinate for a mouse over a point on the map.
2. Draw a PolyLine between a launch site to the selected coastline point

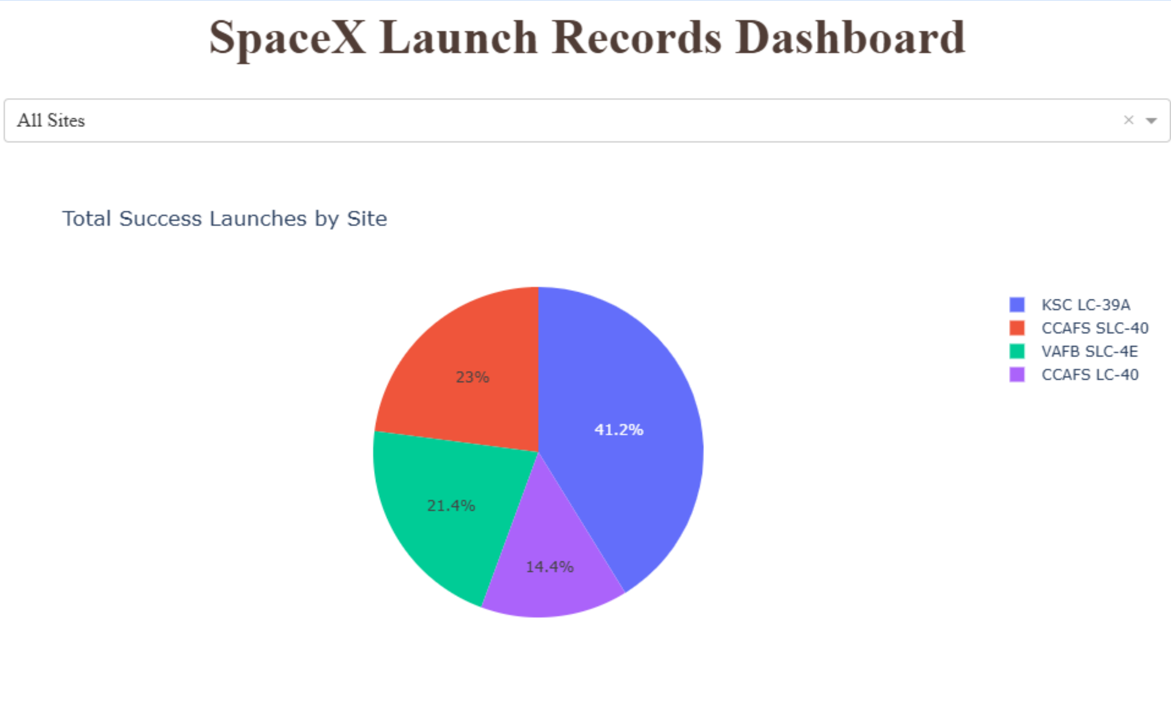




4. Results – 7.Dashboard with Ploty Dash

TASK 1: Add a Launch Site Drop-down Input Component

1. For 4 different launch sites, this task implement a dropdown menu.
2. The options is the list for all 4 launch sites.
3. There are also some other attributes for the plots.



4. Results – 7.Dashboard with Ploty Dash

TASK 2: Add a callback function to render success-pie-chart based on selected site dropdown

1. For 4 different launch sites, this task implement a dropdown menu.
2. The options is the list for all 4 launch sites.
3. There are also some other attributes for the plots.

4. Results – 7.Dashboard with Ploty Dash

TASK 3: Add a Range Slider to Select Payload
TASK 4: Add a callback function to render the success-payload-scatter-chart scatter plot

1. The range slider is to find if variable payload is correlated to mission outcome.
2. It is also to color-label the Booster version on each scatter point so that we may observe mission outcomes with different boosters.

4. Results – 8. ML Prediction

TASK 1: Create a NumPy array from the column Class in data.

1. By applying the method to_numpy() then assign it to the variable Y.
2. The output is a Pandas series (only one bracket df['name of column']).

TASK 1

Create a NumPy array from the column `Class` in `data`, by applying the method `to_numpy()` then assign it to the variable `Y`,make sure the output is a Pandas series (only one bracket df['name of column']).

```
Y = pd.Series(data['Class'].to_numpy())
Y.head(10)
```

```
0    0
1    0
2    0
3    0
4    0
5    0
6    1
7    1
8    0
9    0
dtype: int64
```

4. Results – 8. ML Prediction

TASK 1: Create a NumPy array from the column Class in data.

1. By applying the method to_numpy() then assign it to the variable Y.
2. The output is a Pandas series (only one bracket df['name of column']).

TASK 1

Create a NumPy array from the column `Class` in `data`, by applying the method `to_numpy()` then assign it to the variable `Y`,make sure the output is a Pandas series (only one bracket df['name of column']).

```
Y = pd.Series(data['Class'].to_numpy())
Y.head(10)
```

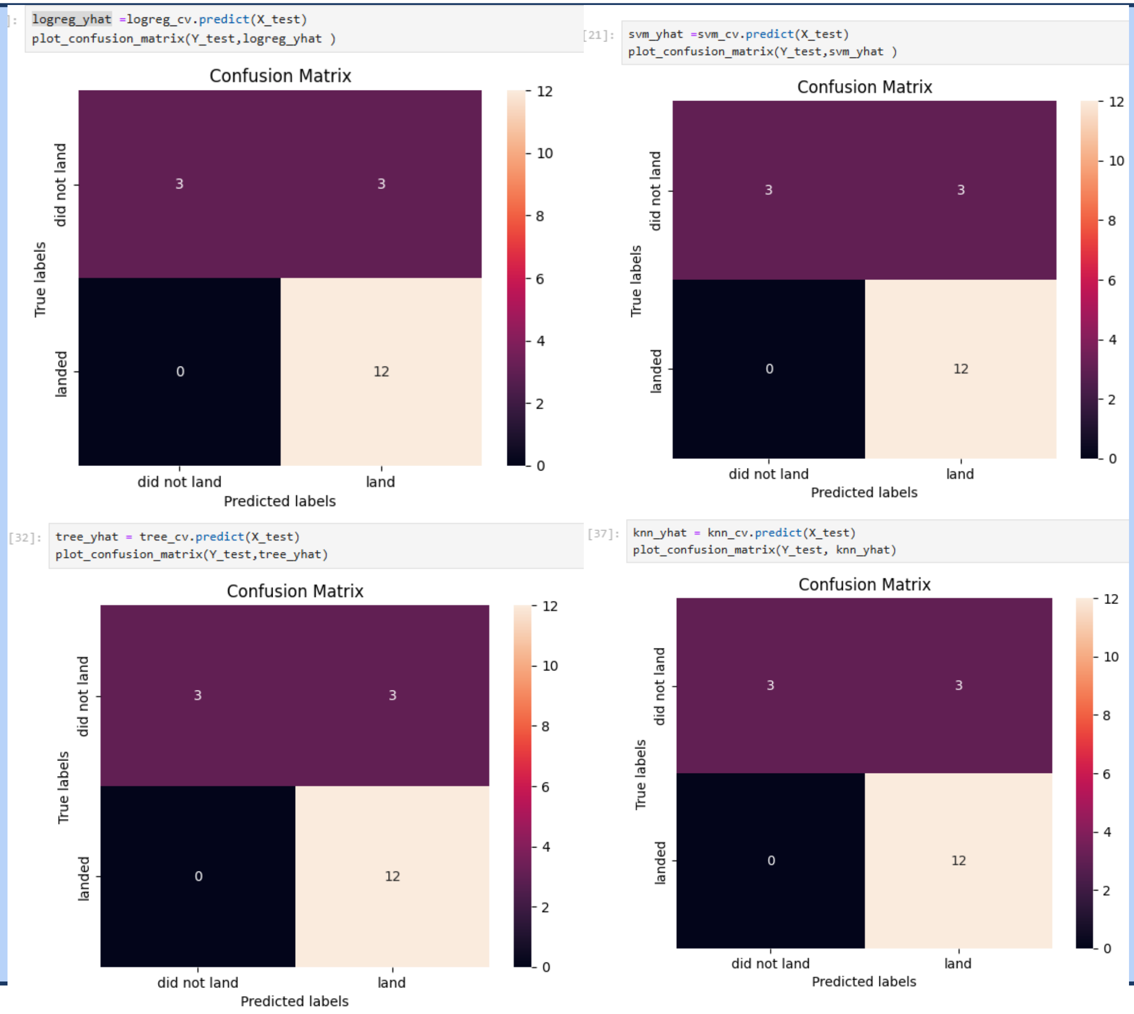
```
0    0
1    0
2    0
3    0
4    0
5    0
6    1
7    1
8    0
9    0
dtype: int64
```

4. Results – 8. ML Prediction

TASK 5-11: Calculate the accuracy of 4 methods and plot their confusion matrix.

1. Confusion Matrix: Examining the confusion matrix, we see that logistic regression can distinguish between the different classes. We see that the major problem is false positives.

2. Using 4 methods(Logistic Regression, SVM, Decision Tree, KNN), first train their best parameters. Plot the confusion matrix using plot_confusion_matrix().
3. We can see that these 4 methods has no difference under these comparison.



4. Results – 8. ML Prediction

TASK 12-1: Find the method performs best:

1. We plot the scores from test sets based on the 4 methods.
2. We can see from the results that these 4 methods have no difference on Training set. So we can try the methods on Whole data.

TASK 12

Find the method performs best:

```
from sklearn.metrics import jaccard_score, f1_score

# Examining the scores from Test sets
jaccard_scores = [
                  jaccard_score(Y_test, logreg_yhat, average='binary'),
                  jaccard_score(Y_test, svm_yhat, average='binary'),
                  jaccard_score(Y_test, tree_yhat, average='binary'),
                  jaccard_score(Y_test, knn_yhat, average='binary'),
                 ]

f1_scores = [
             f1_score(Y_test, logreg_yhat, average='binary'),
             f1_score(Y_test, svm_yhat, average='binary'),
             f1_score(Y_test, tree_yhat, average='binary'),
             f1_score(Y_test, knn_yhat, average='binary'),
            ]

accuracy = [logreg_accuracy, svm_accuracy, tree_accuracy, knn_accuracy]

scores = pd.DataFrame(np.array([jaccard_scores, f1_scores, accuracy]), index=['Jaccard_Score', 'F1_Score', 'Accuracy'] , columns=['LogReg', 'SVM', 'Tree', 'KNN'])
scores
```

[39]:

	LogReg	SVM	Tree	KNN
Jaccard_Score	0.800000	0.800000	0.800000	0.800000
F1_Score	0.888889	0.888889	0.888889	0.888889
Accuracy	0.833333	0.833333	0.833333	0.833333

4. Results – 8. ML Prediction

TASK 12-2: Find the method performs best:

3. we can try the methods on Whole data.
4. Based on the scores and accuracies on the whole dataset, we can conclude that the SVM has the best score among the 4 methods.

```
[40]: # Examining the scores from the whole Dataset
jaccard_scores = [
                  jaccard_score(Y, logreg_cv.predict(X), average='binary'),
                  jaccard_score(Y, svm_cv.predict(X), average='binary'),
                  jaccard_score(Y, tree_cv.predict(X), average='binary'),
                  jaccard_score(Y, knn_cv.predict(X), average='binary'),
                 ]

f1_scores = [
             f1_score(Y, logreg_cv.predict(X), average='binary'),
             f1_score(Y, svm_cv.predict(X), average='binary'),
             f1_score(Y, tree_cv.predict(X), average='binary'),
             f1_score(Y, knn_cv.predict(X), average='binary'),
            ]

accuracy = [logreg_cv.score(X, Y), svm_cv.score(X, Y), tree_cv.score(X, Y), knn_cv.score(X, Y)]

scores = pd.DataFrame(np.array([jaccard_scores, f1_scores, accuracy]),
                      index=['Jaccard_Score', 'F1_Score', 'Accuracy'],
                      columns=['LogReg', 'SVM', 'Tree', 'KNN'])
scores
```

[40]:

	LogReg	SVM	Tree	KNN
Jaccard_Score	0.833333	0.845070	0.810811	0.819444
F1_Score	0.909091	0.916031	0.895522	0.900763
Accuracy	0.866667	0.877778	0.844444	0.855556

5. Conclusion

Conclusion

- Launches with a low payload mass show better results than launches with a larger payload mass.
- SpaceX's record for Falcon 9 first stage landing outcomes has improved.
- The trend is toward better performance and greater success as more launches are made.
- The machine learning models can be used to predict future SpaceX Falcon 9 first stage landing outcomes.
- 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.
- SVM is the best algorithm for this dataset.

Appendix

Special Thanks to:

Instructors

Coursera

IBM

Thank you for watching!