IBM DATA SCIENCE CAPSTONE PROJECT

MOHD AFIQ LUKHMAN BIN MOHD FAUZI

19/5/2024

Outline

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

Executive Summary

Summary of Methodologies:

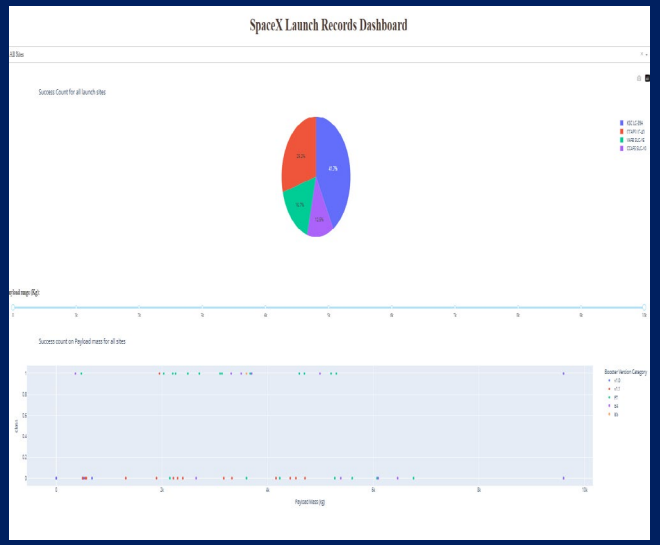
This project is completed by following these steps:

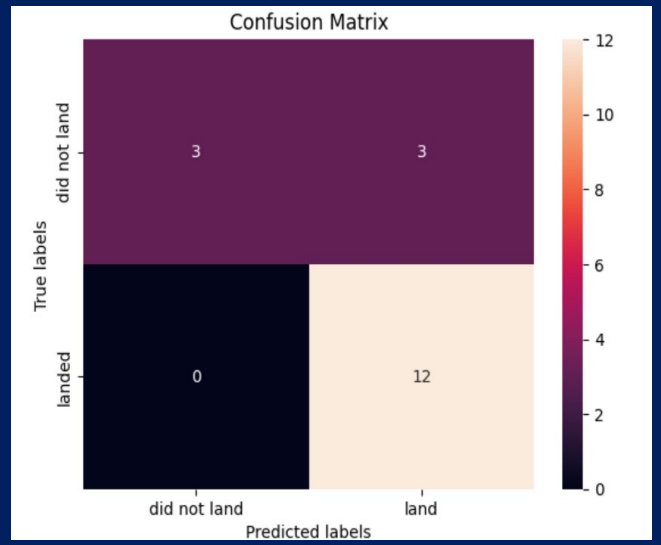
- Data Collection
- Data Wrangling
- Exploratory Data Analysis
- Interactive Visual Analytics
- Predictive Analysis (Classification)

Summary of Results:

This project has produced the following outputs and visualizations:

1. Exploratory Data Analysis results
2. Geospatial analytics
3. Interactive dashboard
4. Predictive analysis of classification models





INTRODUCTION

- SpaceX launches Falcon 9 rockets at a cost of around $62m. This is considerably cheaper than other providers (which usually cost upwards of $165m), and much of the savings are because SpaceX can land, and then re-use the first stage of the rocket.

- If we can make predictions on whether the first stage will land, we can determine the cost of a launch, and use this information to assess whether or not an alternate company should bid and SpaceX for a rocket launch.

- This project will ultimately predict if the Space X Falcon 9 first stage will land successfully.

METHODOLOGY SUMMARY

Data Collection

- Request to the SpaceX API using GET request API
- Use web scrapping

Data Wrangling

- Dealing with missing values by replacing np.nan values with mean value
- Using .value_counts() method to determine following:
  - Number of launches on each site
  - Number and occurrence of each orbit
  - Number and occurrence of mission outcome per orbit type
- Create a landing outcome label

Exploratory Data Analysis

- Exploratory Analysis using SQL queries to retrieve data
- Exploratory Analysis using Pandas and Matplotlib for data visualization

Interactive Visual Analytics

- Geospatial analytics using Folium
- Interactive Dashboard using Plotly

Data Modelling and Evaluation

- Predictive Analysis-Classification

Data Collection

Task 1

- Requesting and parsing the SpaceX launch data using the GET request
- Decode the response content as a Json using *.json()* and turn it into a Pandas DataFrame using *.json_normalize()* method
- Create dictionary and convert into Pandas DataFrame using *.from_dict()* method

Task 2

- Filter the dataframe to only include `Falcon 9` launches using BoosterVersion

Task 3

- Dealing with Missing Values
- Calculate mean value of PayloadMass column
- Replace the np.nan value with mean

https://github.com/AfiqLukhman/Applied-Data-Science-Capstone-Project-IBM/blob/cac5151200541c658d1f7a6fedf3c3ac2753c763/jupyter-labs-spacex-data-collection-api.ipynb

Data Wrangling

Task 1

Calculate the number of launches on each site

- Use the method *.value_counts()* on the column 'LaunchSite' to determine the number of launches on each site

Task 2

Calculate the number and occurrence of each orbit

- Use the method *.value_counts()* to determine the number and occurrence of each orbit in the column Orbit

Task 3

Calculate the number and occurence of mission outcome of the orbits

- Use the method .value_counts() on the column to determine the number of Landing_Outcome .Then assign it to a variable landing_outcomes.

Task 4

Create a landing outcome label from Outcome column

- Create a list where the element is zero if the corresponding row in 'Outcome' is in the set *bad_outcome*
- Assign it to the variable landing_class

https://github.com/AfiqLukhman/Applied-Data-Science-Capstone-Project-IBM/blob/0af2e4a599eddecc241f9e0aedff8656f928cc01/labs-jupyter-spacex-Data%20wrangling.ipynb

EXPLORATORY DATA ANALYSIS

SQL

Using SQL queries to retrieve data as following:

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 succesful landing outcome in ground pad was acheived.
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. Use a subquery
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.

GitHub Link

EXPLORATORY DATA ANALYSIS

Scatter Plot	Bar Plot	Line Plot
Use to visualize the relationship between: • Flight number and Launch Site • PayloadMass and Launch Site • Flight number and Orbit • PayloadMass and Orbit	Use to visualize the relationship between: • The success rate and orbit type	Use to visualize the relationship between: • Launch success yearly trend

GitHub Link

GEOSPATIAL ANALYSIS – FOLIUM

The following steps were taken to visualize the launch data on an interactive map:

1. Mark all launch sites on a map

- Initialise the map using a Folium Map object
- Add a folium.Circle and folium.Marker for each launch site on the launch map

2.Mark the success/failed launches for each site on a map

- As many launches have the same coordinates, it makes sense to cluster them together.
- Before clustering them, assign a marker colour of successful (class = 1) as green, and failed (class = 0) as red.
- To put the launches into clusters, for each launch, add a folium.Marker to the MarkerCluster() object.
- Create an icon as a text label, assigning the icon_color as the marker_colour determined previously.

3. Calculate the distances between a launch site to its proximities

- To explore the proximities of launch sites, calculations of distances between points can be made using the Lat and Long values.
- After marking a point using the Lat and Long values, create a folium.Marker object to show the distance.
- To display the distance line between two points, draw a folium.PolyLine and add this to the map.

INTERACTIVE DASHBOARD-PLOTLY DASH

The following plots were added to a Plotly Dash dashboard to have an interactive visualisation of the data:

1. Pie chart (px.pie()) showing the total successful launches per site
   - This makes it clear to see which sites are most successful
   - The chart could also be filtered (using a dcc.Dropdown() object) to see the success/failure ratio for an individual site

2. Scatter graph (px.scatter()) to show the correlation between outcome (success or not) and payload mass (kg)
   - This could be filtered (using a RangeSlider() object) by ranges of payload masses
   - It could also be filtered by booster version

GitHub Link

Data Modelling and Evaluation - Predictive Analysis(Classification)

The following steps for predictive analysis:

- 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']).
- Standardize the data in X then reassign it to the variable X using the transform provided below.
- Use the function train_test_split to split the data X and Y into training and test data. Set the parameter test_size to 0.2 and random_state to 2. The training data and test data should be assigned to the following labels.
- Create a logistic regression object then create a GridSearchCV object logreg_cv with cv = 10. Fit the object to find the best parameters from the dictionary parameters.
- Calculate the accuracy on the test data using the method score:
- Create a support vector machine object then create a GridSearchCV object svm_cv with cv = 10. Fit the object to find the best parameters from the dictionary parameters.
- Calculate the accuracy on the test data using the method score:
- Create a decision tree classifier object then create a GridSearchCV object tree_cv with cv = 10. Fit the object to find the best parameters from the dictionary parameters.
- Create a decision tree classifier object then create a GridSearchCV object tree_cv with cv = 10. Fit the object to find the best parameters from the dictionary parameters.
- Create a k nearest neighbors object then create a GridSearchCV object knn_cv with cv = 10. Fit the object to find the best parameters from the dictionary parameters.
- Calculate the accuracy of knn_cv on the test data using the method score
- Find the method performs best

GitHub Link

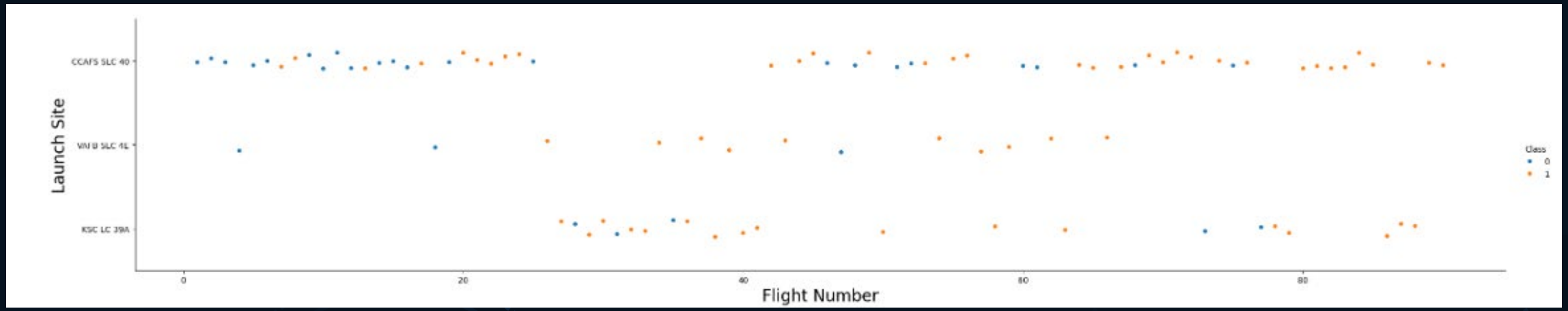
RESULTS

Exploratory Data Analysis

Interactive Analytics

Predictive Analytics

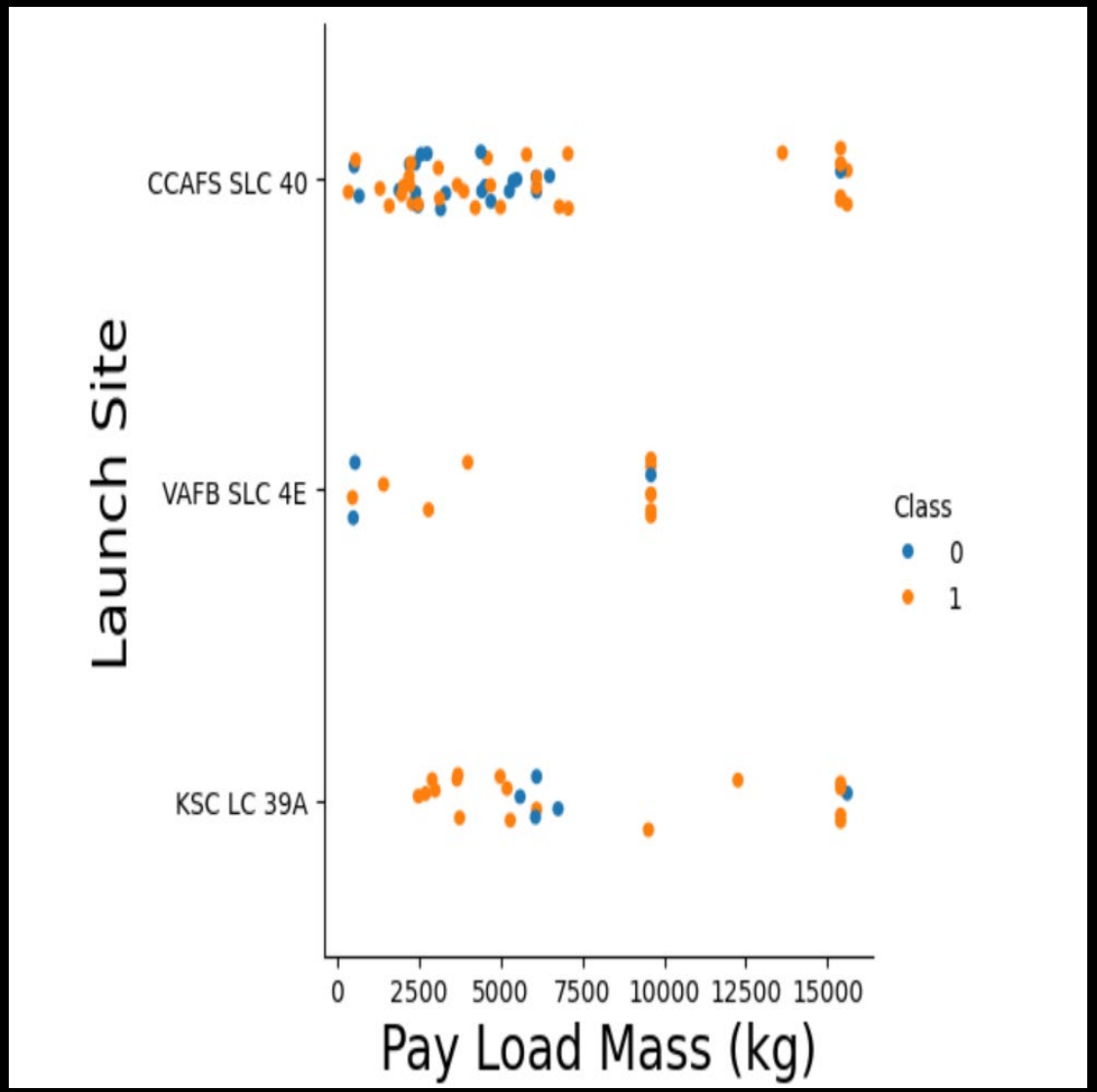
EDA-VISUALIZATION



Flight Number VS Launch Site

The scatter plot of Launch Site vs. Flight Number shows that:

- As the number of flights increases, the rate of success at a launch site increases.
- Most of the early flights (flight numbers < 30) were launched from CCAFS SLC 40, and were generally unsuccessful.
- The flights from VAFB SLC 4E also show this trend, that earlier flights were less successful.
- No early flights were launched from KSC LC 39A, so the launches from this site are more successful.
- Above a flight number of around 30, there are significantly more successful landings (Class = 1).



PayLoadMass vs Launch Site

The scatter plot of Launch Site vs. Payload Mass shows that:

- Above a payload mass of around 7000 kg, there are very few unsuccessful landings, but there is also far less data for these heavier launches.
- There is no clear correlation between payload mass and success rate for a given launch site.
- All sites launched a variety of payload masses, with most of the launches from CCAFS SLC 40 being comparatively lighter payloads (with some outliers

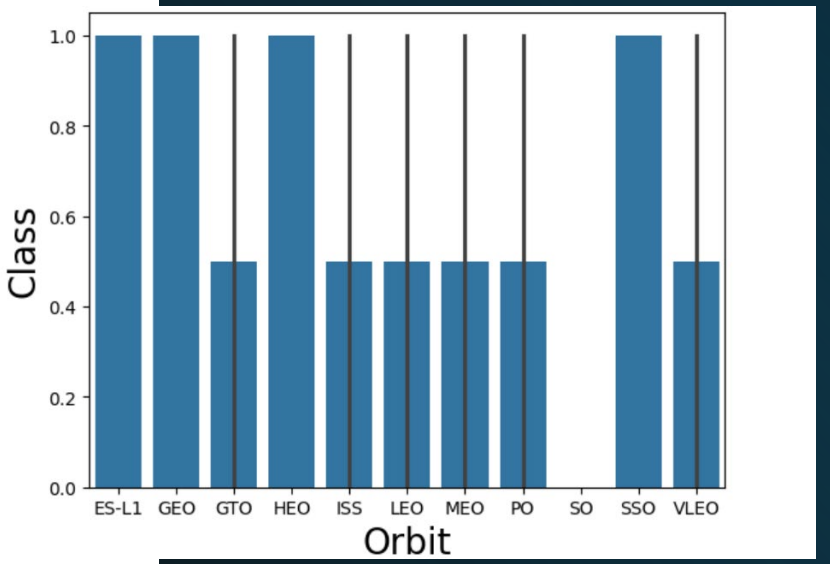
Success Rate vs Orbit Type

The bar chart of Success Rate vs. Orbit Type shows that the following orbits have the highest (100%) success rate:

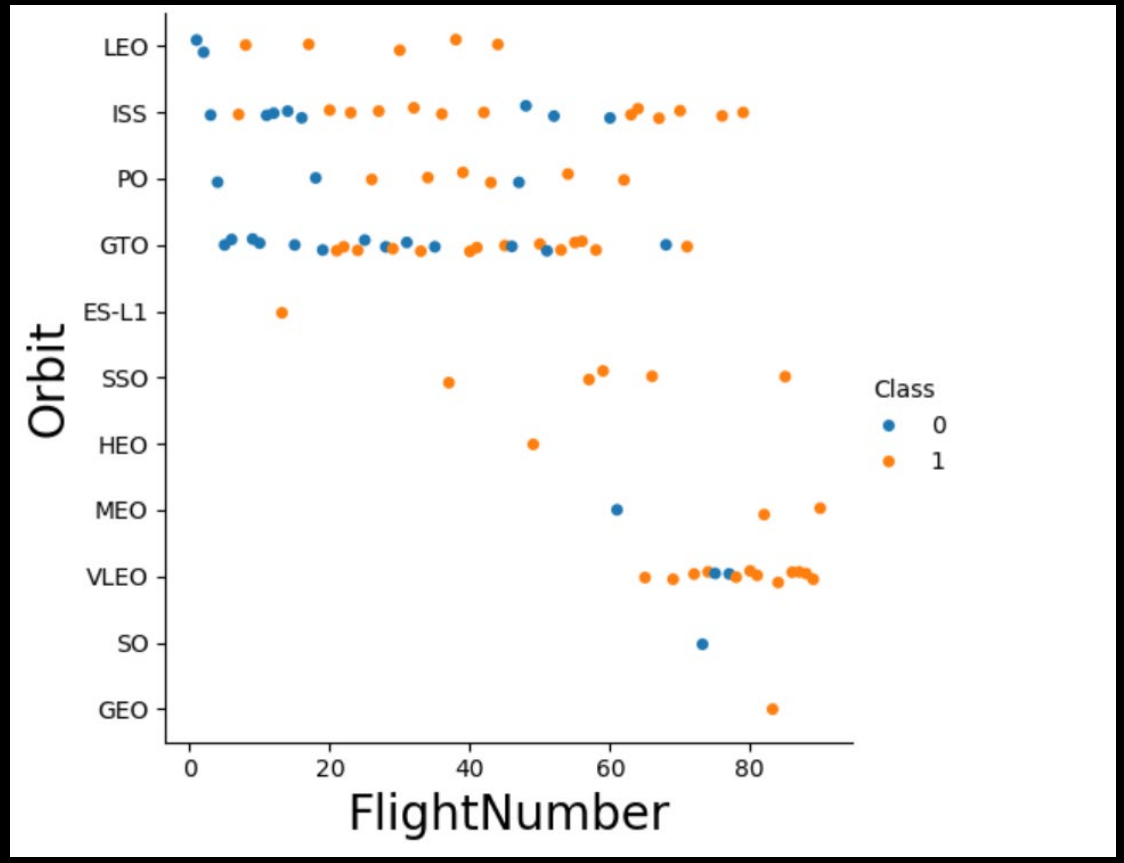
- ES-L1 (Earth-Sun First Lagrangian Point)
- GEO (Geostationary Orbit)
- HEO (High Earth Orbit)
- SSO (Sun-synchronous Orbit)

The orbit with the lowest (0%) success rate is:

- SO (Heliocentric Orbit)



Flight Number vs Orbit Type



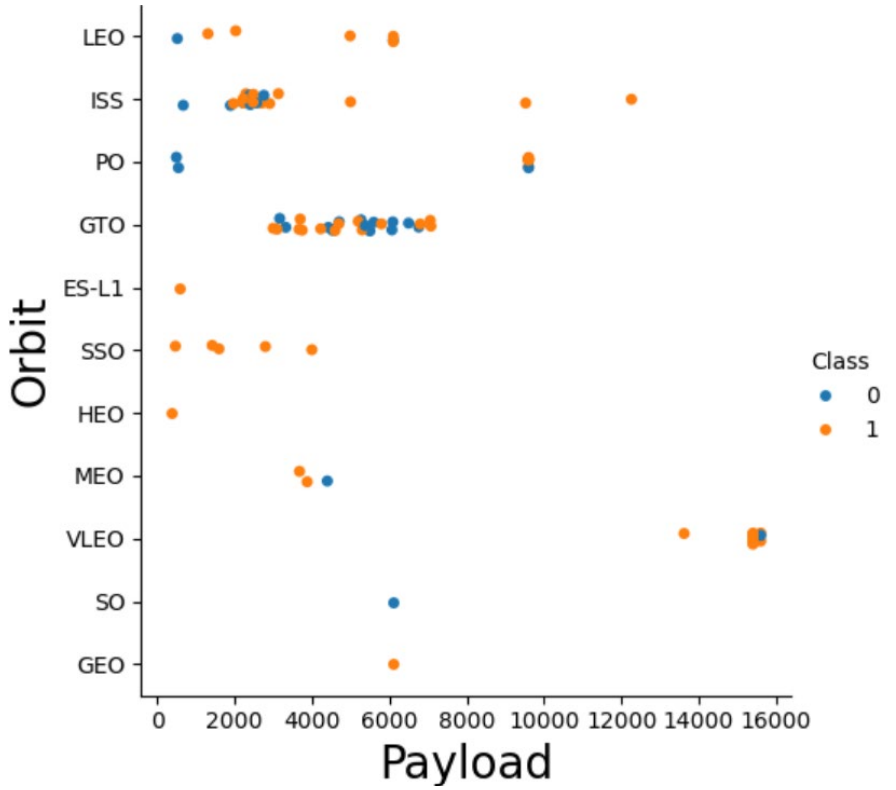
This scatter plot of Orbit Type vs. Flight number shows a few useful things that the previous plots did not, such as:

- The 100% success rate of GEO, HEO, and ES-L1 orbits can be explained by only having 1 flight into the respective orbits.
- The 100% success rate in SSO is more impressive, with 5 successful flights.
- There is little relationship between Flight Number and Success Rate for GTO.
- Generally, as Flight Number increases, the success rate increases. This is most extreme for LEO, where unsuccessful landings only occurred for the low flight numbers (early launches).

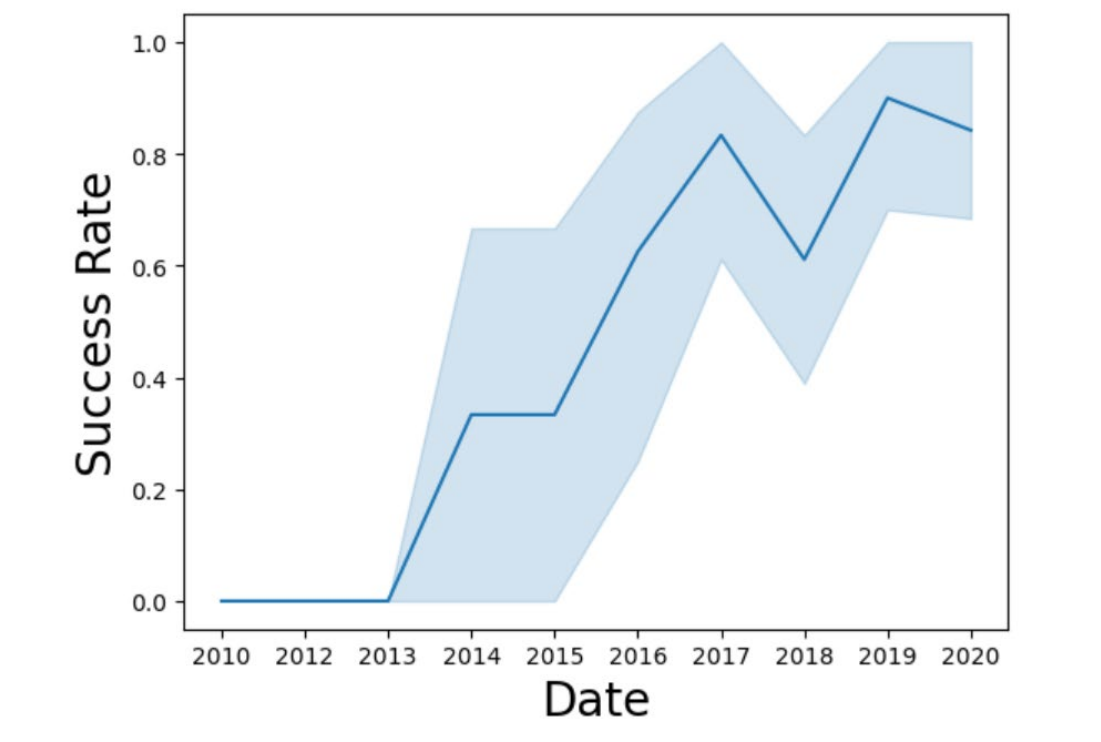
Payload vs Orbit Type

This scatter plot of Orbit Type vs. Payload Mass shows that:

- The following orbit types have more success with heavy payloads:
  - PO (although the number of data points is small)
  - ISS
  - LEO
- For GTO, the relationship between payload mass and success rate is unclear.
- VLEO (Very Low Earth Orbit) launches are associated with heavier payloads, which makes intuitive sense.



Launch Successful Yearly Trend



The line chart of yearly average success rate shows that:

- Between 2010 and 2013, all landings were unsuccessful (as the success rate is 0).
- After 2013, the success rate generally increased, despite small dips in 2018 and 2020.
- After 2016, there was always a greater than 50% chance of success.

EDA-SQL

ALL LAUNCH SITES NAME

- Find the names of the unique launch sites.
- The word UNIQUE returns only unique values from the LAUNCH_SITE column of the SPACEXTBL table.

```
%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

LAUNCH SITE NAMES BEGIN WITH 'CCA'

- Display 5 records where launch sites begin with the string 'CCA'.
- LIMIT 5 fetches only 5 records, and the LIKE keyword is used with the wild card 'CCA%' to retrieve string values beginning with 'CCA'.

```
%sql select Launch_Site from SPACEXTBL where Launch_Site like 'CCA%' limit 5;
```

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

Launch_Site
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40

TOTAL PAYLOAD MASS

- Calculate the total payload carried by boosters from NASA
- The SUM keyword is used to calculate the total of the LAUNCH column, and the SUM keyword (and the associated condition) filters the results to only boosters from NASA (CRS).

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE CUSTOMER = 'NASA(CRS)';
 * sqlite:///my_data1.db
Done.
```

SUM(PAYLOAD_MASS__KG_)
None

AVERAGE PAYLOAD MASS BY F9 V1.1

- Calculate the average payload mass carried by booster version F9 v1.1
- The AVG keyword is used to calculate the average of the PAYLOAD_MASS__KG_ column, and the WHERE keyword (and the associated condition) filters the results to only the F9 v1.1 booster version

```
%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

FIRST SUCCESSFUL GROUND LANDING DATE

- Find the dates of the first successful landing outcome on ground pad
- The MIN keyword is used to calculate the minimum of the DATE column, i.e. the first date, and the WHERE keyword (and the associated condition) filters the results to only the successful ground pad landings.

```
%sql select min(DATE) from SPACEXTBL where Landing_Outcome = 'Success (ground pad)';
```

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

min(DATE)
2015-12-22

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
- The WHERE keyword is used to filter the results to include only those that satisfy both conditions in the brackets (as the AND keyword is also used).
- The BETWEEN keyword allows for 4000 < x < 6000 values to be selected

```
%sql SELECT BOOSTER_VERSION from SPACEXTBL WHERE LANDING_OUTCOME = 'Success (drone ship)' and  PAYLOAD_MAS
```

```
 * 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

- Calculate the total number of successful and failure mission outcomes
- The COUNT keyword is used to calculate the total number of mission outcomes, and the GROUPBY keyword is also used to group these results by the type of mission outcome.

```
%sql select count(MISSION_OUTCOME) from SPACEXTBL where MISSION_OUTCOME = 'Success' or MISSION_OUTCOME = '
```

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

count(MISSION_OUTCOME)
99

BOOSTERS CARRIED MAXIMUM PAYLOAD

- List the names of the booster which have carried the maximum payload mass
- A subquery is used here. The SELECT statement within the brackets finds the maximum payload, and this value is used in the WHERE condition
- The DISTINCT keyword is then used to retrieve only distinct /unique booster versions.

```
%sql SELECT BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT max(PAYLOAD_MASS__KG_) FROM S
```

```
 * 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

2015 LAUNCH RECORDS

- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
- The WHERE keyword is used to filter the results for only failed landing outcomes, AND only for the year of 2015.

```
%sql SELECT BOOSTER_VERSION,LAUNCH_SITE,LANDING_OUTCOME FROM SPACEXTBL WHERE LANDING_OUTCOME = 'Failure (d
```

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

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

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, in descending order
- The WHERE keyword is used with the BETWEEN keyword to filter the results to dates only within those specified.
- The results are then grouped and ordered, using the keywords GROUP BY and ORDER BY, respectively, where DESC is used to specify the descending order.

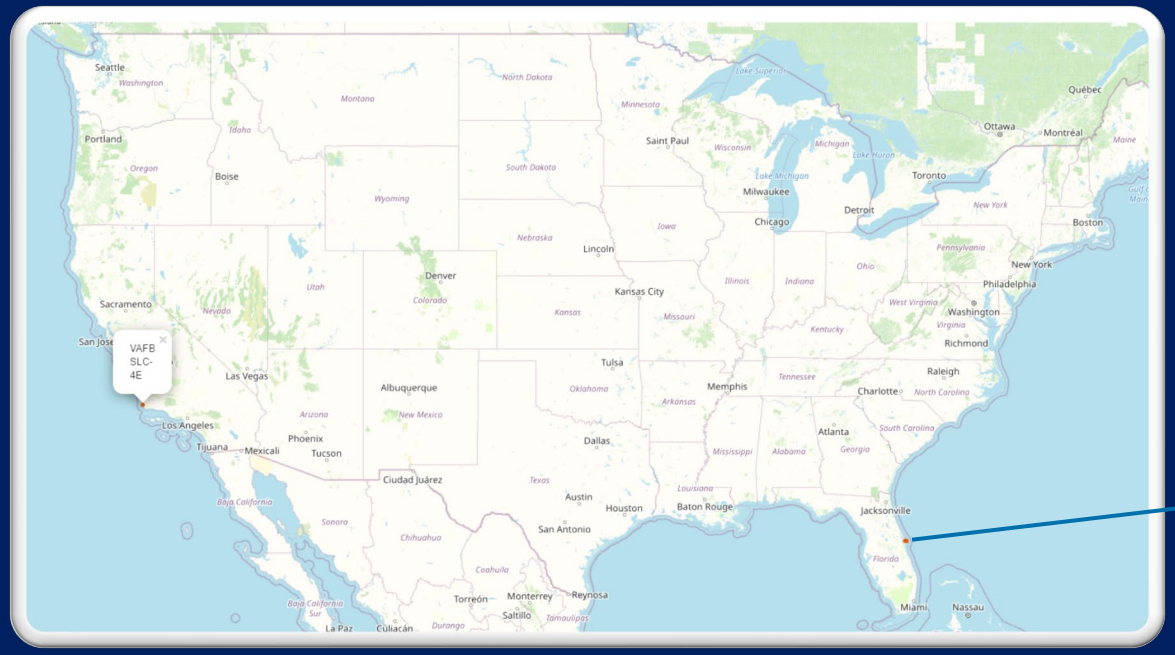
```
%sql select * from SPACEXTBL where Landing_Outcome = 'Success (ground pad)' and date between '2010-06-04'
```

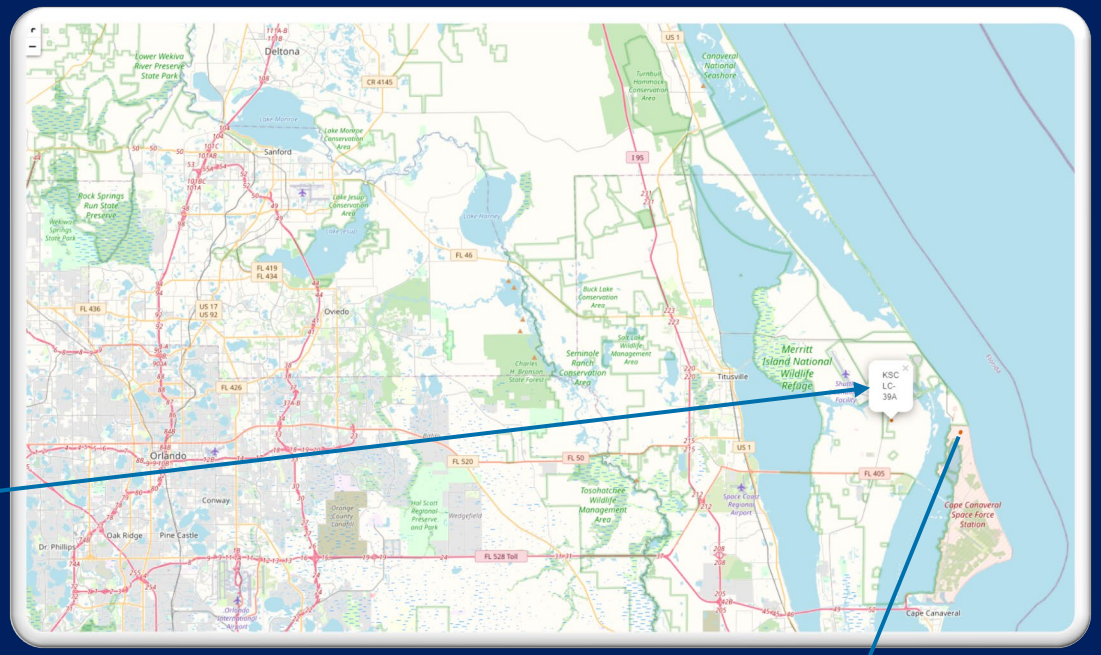
```
 * sqlite:///my_data1.db
Done.
```

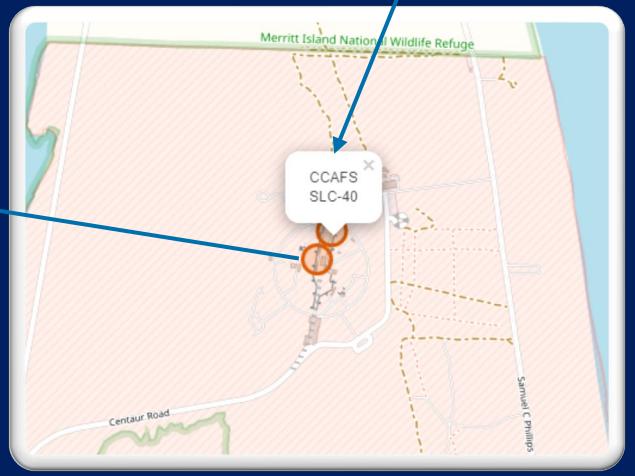
Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome
2017-02-19	14:39:00	F9 FT B1031.1	KSC LC-39A	SpaceX CRS-10	2490	LEO (ISS)	NASA (CRS)	Success
2016-07-18	4:45:00	F9 FT B1025.1	CCAFS LC-40	SpaceX CRS-9	2257	LEO (ISS)	NASA (CRS)	Success
2015-12-22	1:29:00	F9 FT B1019	CCAFS LC-40	OG2 Mission 2 11 Orbcomm-OG2 satellites	2034	LEO	Orbcomm	Success

LAUNCH SITES PROXIMITY ANALYSIS : FOLIUM INTERACTIVE MAP

ALL LAUNCH SITES ON A MAP

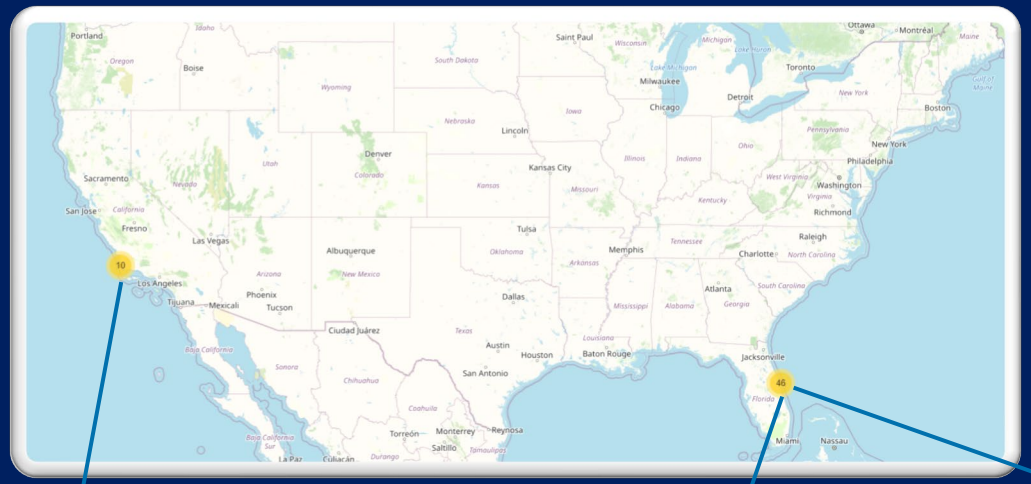






All SpaceX launch sites are on coasts of the United States of America, specifically Florida and California.

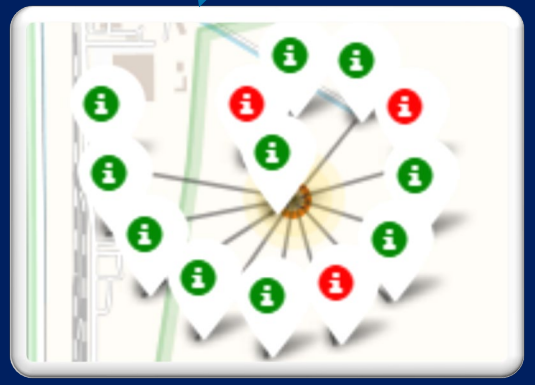
SUCCESS/FAILED LAUNCHES FOR EACH SITE



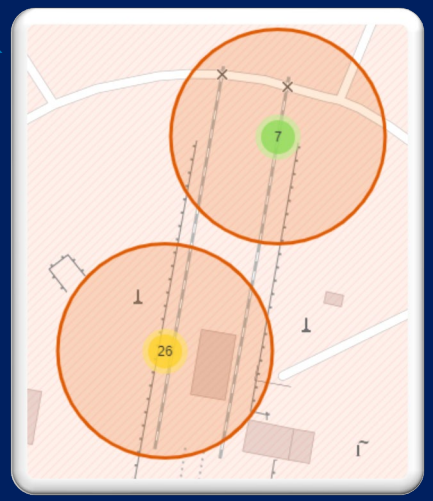
Launches have been grouped into clusters, and annotated with green icons for successful launches, and red icons for failed launches.

VAFB SLC-4E

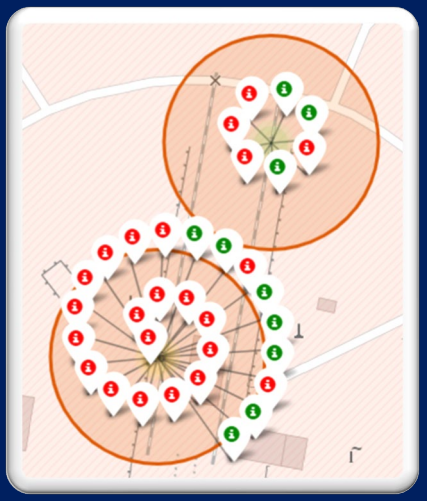
KSC LC-39A



CCAFS SLC-40 and CCAFS LC-40

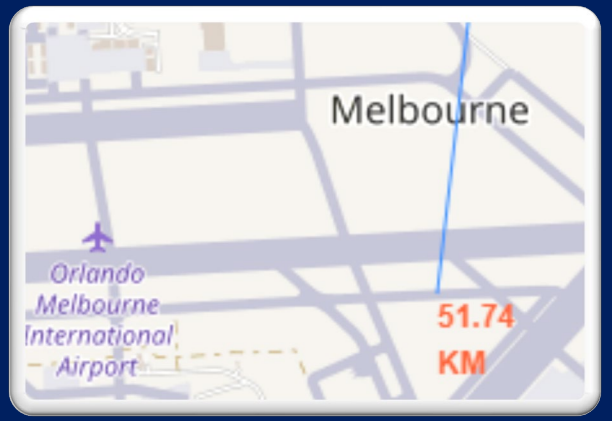


=



PROXIMITY OF LAUNCH SITES TO OTHER POINTS OF INTEREST

Using the CCAFS SLC-40 launch site as an example site, we can understand more about the placement of launch sites.

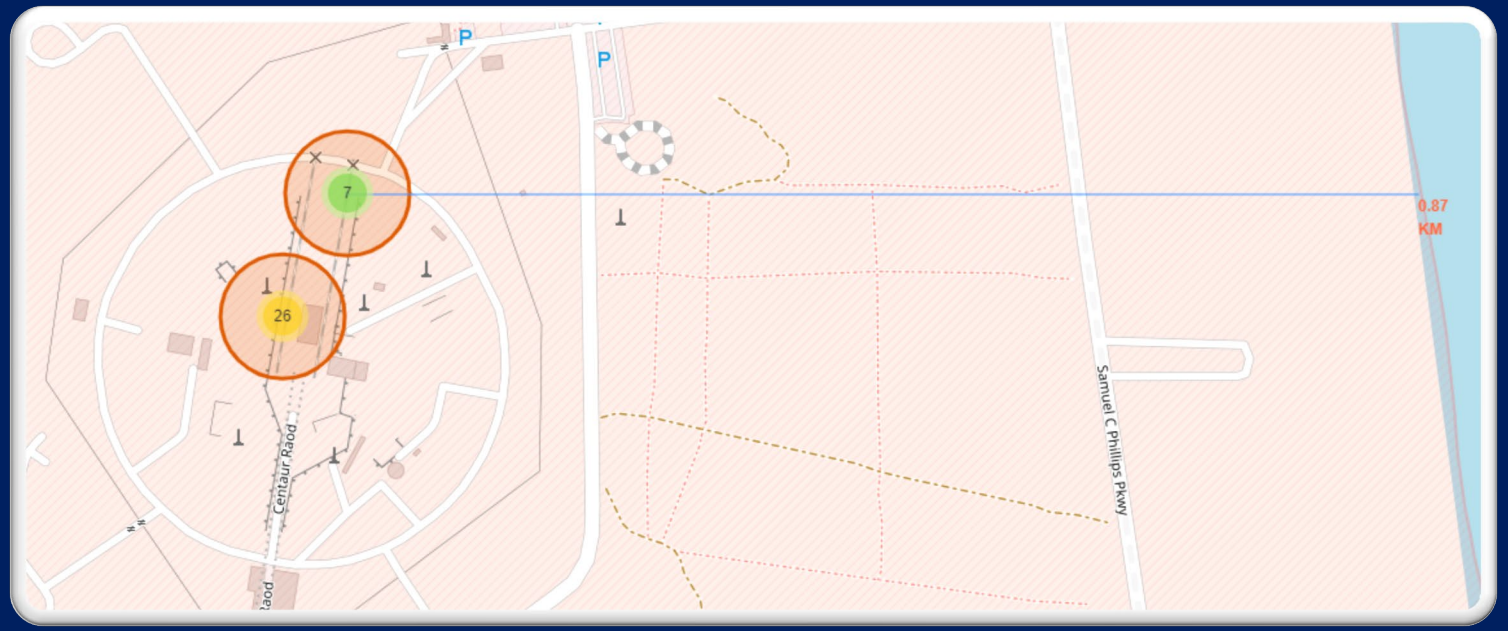


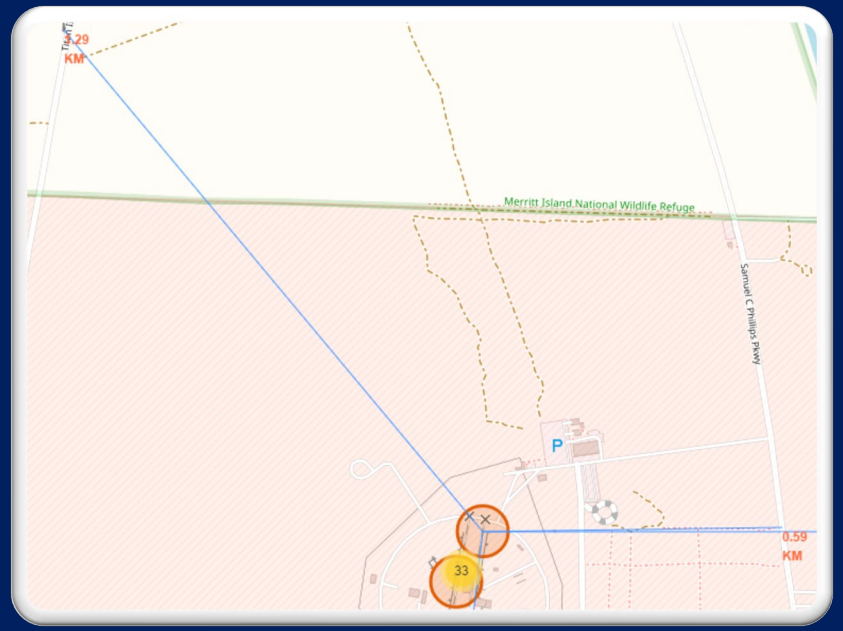
Are launch sites in close proximity to railways?
- YES. The coastline is only 0.87 km due East.

Are launch sites in close proximity to highways?
- YES. The nearest highway is only 0.59km away.

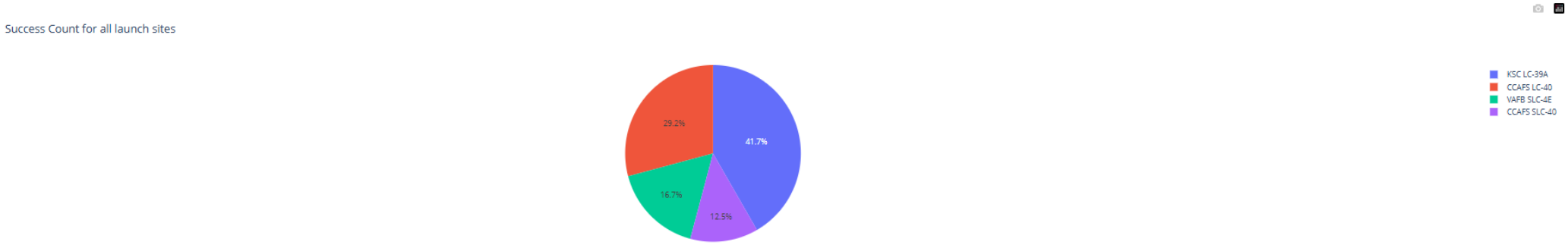
Are launch sites in close proximity to railways?
- YES. The nearest railway is only 1.29 km away.

Do launch sites keep certain distance away from cities?
- YES. The nearest city is 51.74 km away.





INTERACTIVE DASHBOARD: PLOTLY DASH

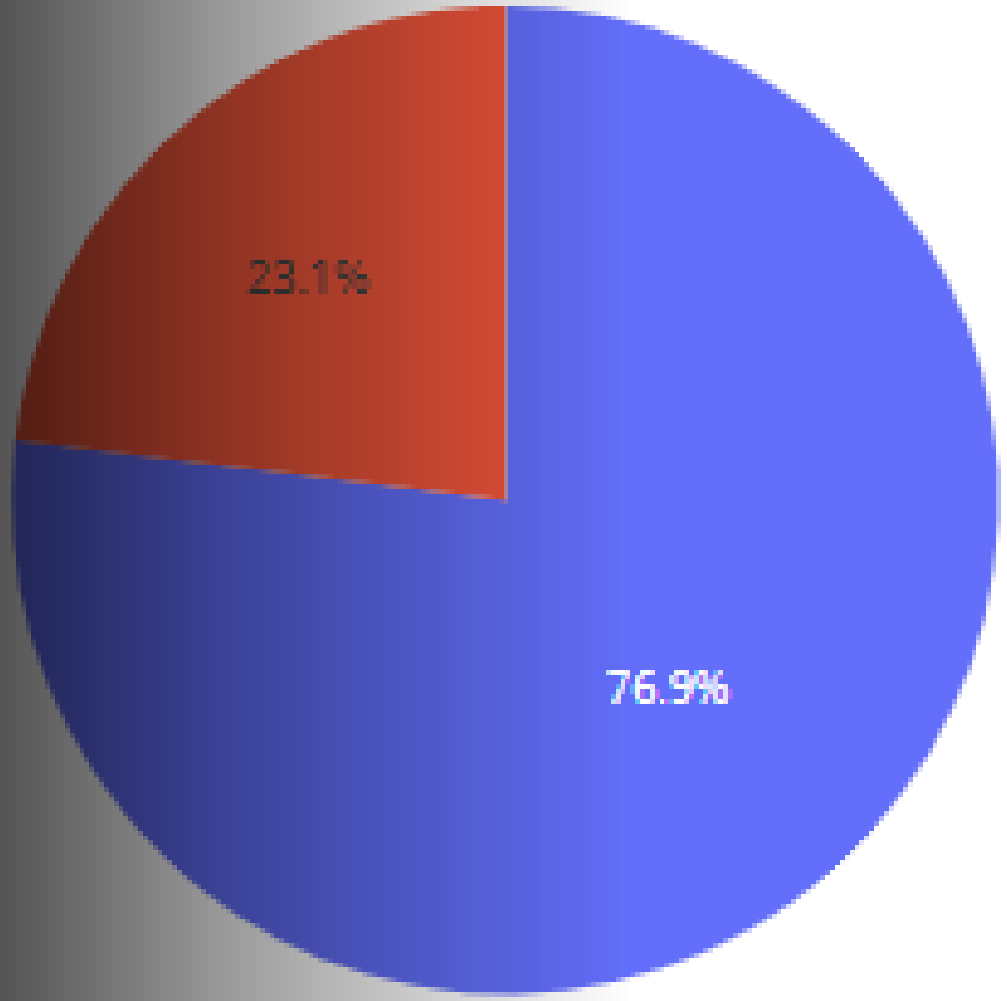


LAUNCH SUCCESS COUNT FOR ALL SITES

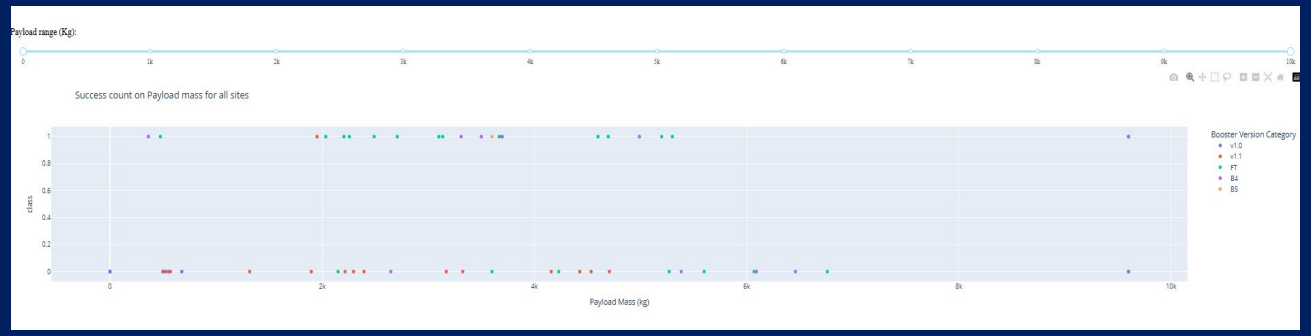
- The launch site KSC LC-39 A had the most successful launches, with 41.7% of the total successful launches.

PIE CHART FOR THE HIGHEST LAUNCH SUCCESS RATIO

- The launch site KSC LC-39 A also had the highest rate of successful launches, with a 76.9% success rate.

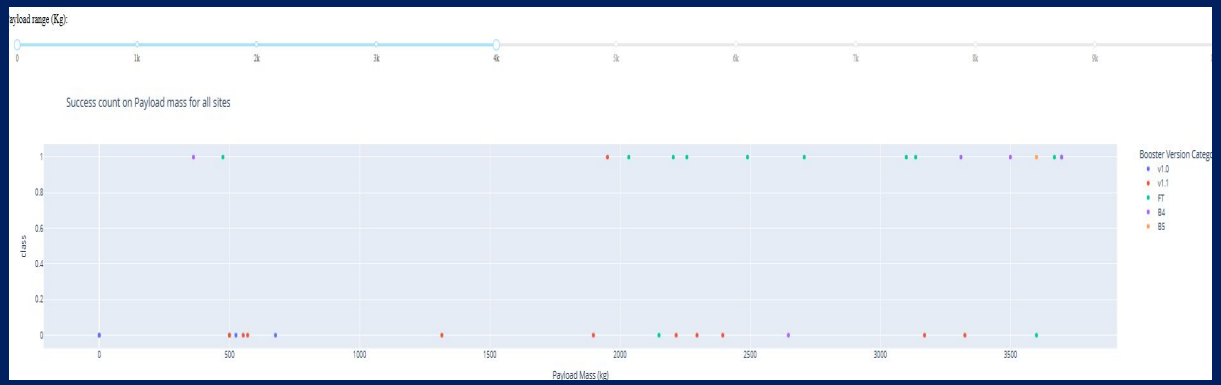


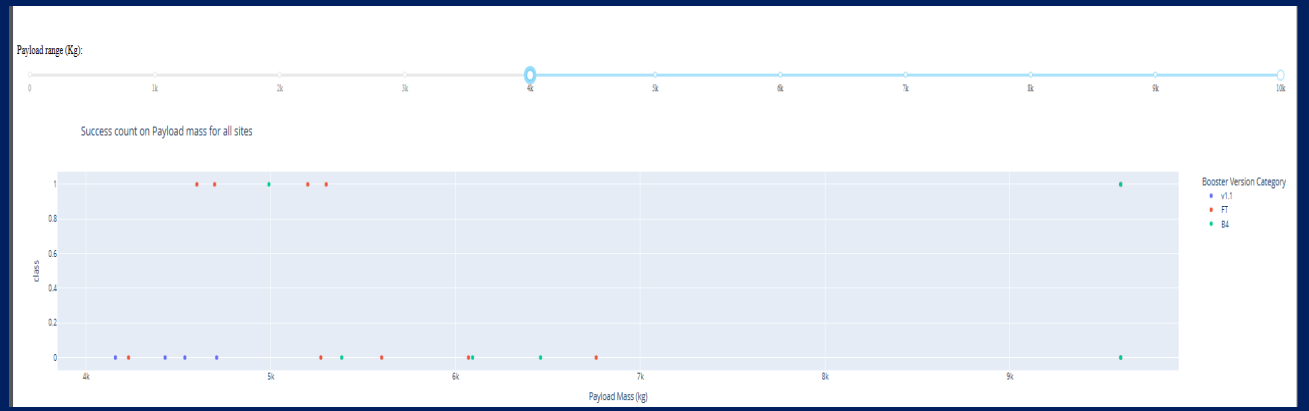
LAUNCH OUTCOME VS. PAYLOAD SCATTER PLOT FOR ALL SITES



- Plotting the launch outcome vs. payload for all sites shows a gap around 4000 kg, so it makes sense to split the data into 2 ranges:
  - 0 – 4000 kg (low payloads)
  - 4000 – 10000 kg (massive payloads)

From these 2 plots, it can be shown that the success for massive payloads is lower than that for low payloads





- It is also worth noting that some booster types (v1.0 and B5) have not been launched with massive payloads.

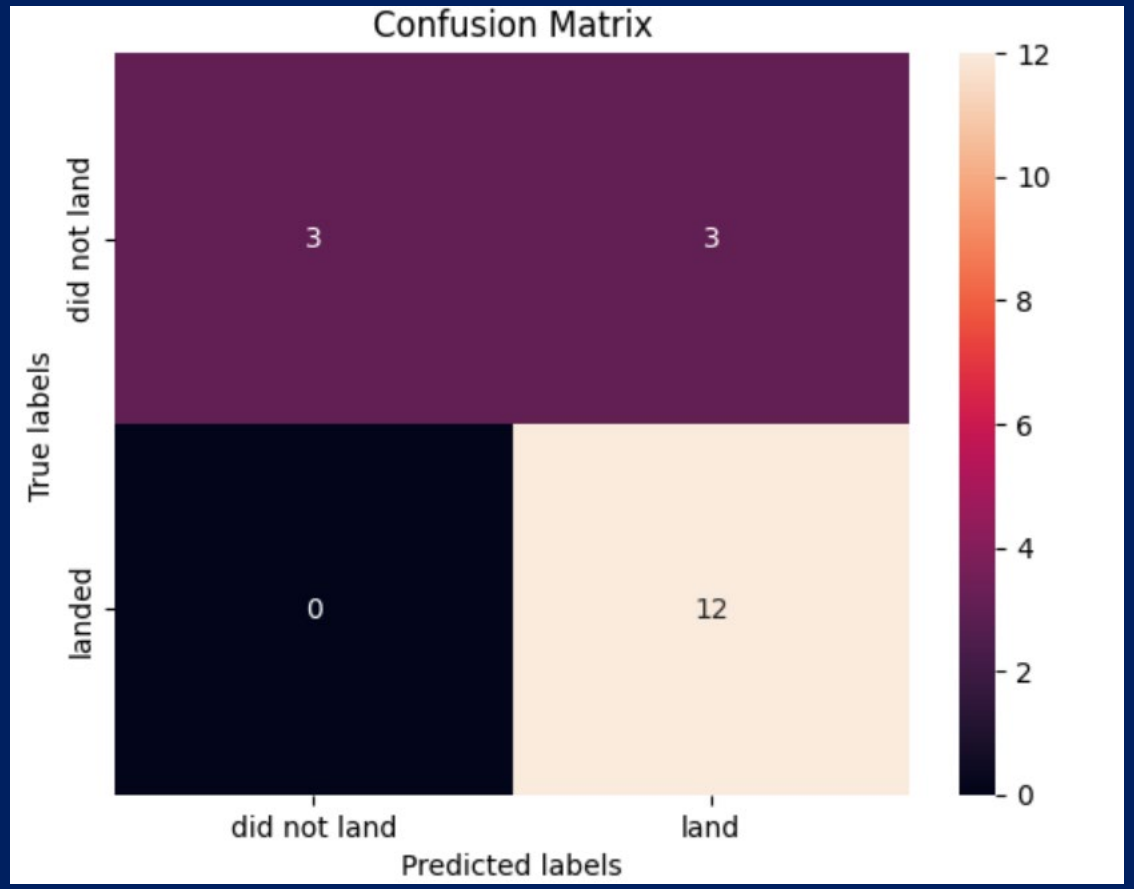
PREDICTIVE ANALYSIS (CLASSIFICATION)

Classification Accuracy

	0
Method	Test Data Accuracy
Logistic_Reg	0.833333
SVM	0.833333
Decision Tree	0.833333
KNN	0.833333

- All methods are performing equally on the test data
- The result is getting the same accuracy of 0.833333 test data accuracy

Confusion matrix



- All 4 classification models had the same confusion matrixes and were able equally distinguish between the different classes.
- The major problem is false positives for all the models

Conclusion

- Orbit types ES-L1, GEO, HEO, and SSO, have the highest (100%) success rate.
- The launch site KSC LC-39 A had the most successful launches, with 41.7% of the total successful launches, and also the highest rate of successful launches, with a 76.9% success rate.
- 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(unsuccessful mission) are both there here
- As the number of flights increases, the rate of success at a launch site increases, with most early flights being unsuccessful

```
[98]: spacex_url="https://api.spacexdata.com/v4/launches/past"

[99]: response = requests.get(spacex_url)
```

```
launch_dict = {'FlightNumber': list(data['flight_number']),
               'Date': list(data['date']),
               'BoosterVersion': BoosterVersion,
               'PayloadMass': PayloadMass,
               'Orbit': Orbit,
               'LaunchSite': LaunchSite,
               'Outcome': Outcome,
               'Flights': Flights,
               'GridFins': GridFins,
               'Reused': Reused,
               'Legs': Legs,
               'LandingPad': LandingPad,
               'Block': Block,
               'ReusedCount': ReusedCount,
               'Serial': Serial,
               'Longitude': Longitude,
               'Latitude': Latitude}
```

Select Report Type:

Which report would you like to display, yearly or recession?

Yearly Statistics

Year:

Which year would you like to display for the yearly report?

2005

```
# Create a data from launch_dict
data_launch = pd.DataFrame.from_dict([launch_dict])
```

```
[ ]: # Calculate the mean value of PayloadMass column
     PayloadMassAvg = data_falcon9['PayloadMass'].mean()
     PayloadMassAvg.mean()

[ ]: # Replace the np.nan values with its mean value
     PayloadMassAvg = PayloadMassAvg.mean().astype(int)
     data_falcon9['PayloadMass'].replace(np.nan, PayloadMassAvg)
     data_falcon9
```