SpaceX Data Science Project

ANTONIO SILVA

AUGUST 2021

Outline

- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion
- Appendix

Executive Summary

- The methodologies used in project include data collection using web scraping, wrangling to clean the data, data analysis using SQL as well as different types data visualization using python, including plots charts and maps, and an interactive dashboard was created for further analysis and visualization. Machine learning methods were used to produce predictive models, evaluated and validated on testing data.
- The machine learning models were produced in order to predict whether a first stage will land. The decision tree model showed the highest testing accuracy at 88%.

Introduction

- The project uses SpaceX data to accomplish data science tasks using the knowledge acquired in all previous courses. The goal is to gather and clean the data, use visualizations and exploratory data analysis, as well as evaluate predictive models and machine learning algorithms.

- We want to be able to predict the success of SpaceX landing outcomes and understand the factors involved. Let’s go!

Methodology

Methodology

Data collection methodology:

- The data was collected using web scraping techniques on the Wikipedia web page on SpaceX Falcon rocket launches.

Perform data wrangling

- The data is processed using normalization and scaling, and missing numeric values are replaced with the mean or rows are removed if necessary. Relevant categorical attributes are made binary using one hot encoding.

Perform exploratory data analysis (EDA) using visualization and SQL

- This method involved importing the cleaned data to a database and performing subsetting queries or calculation to gain insights into the data

Perform interactive visual analytics using Folium and Plotly Dash

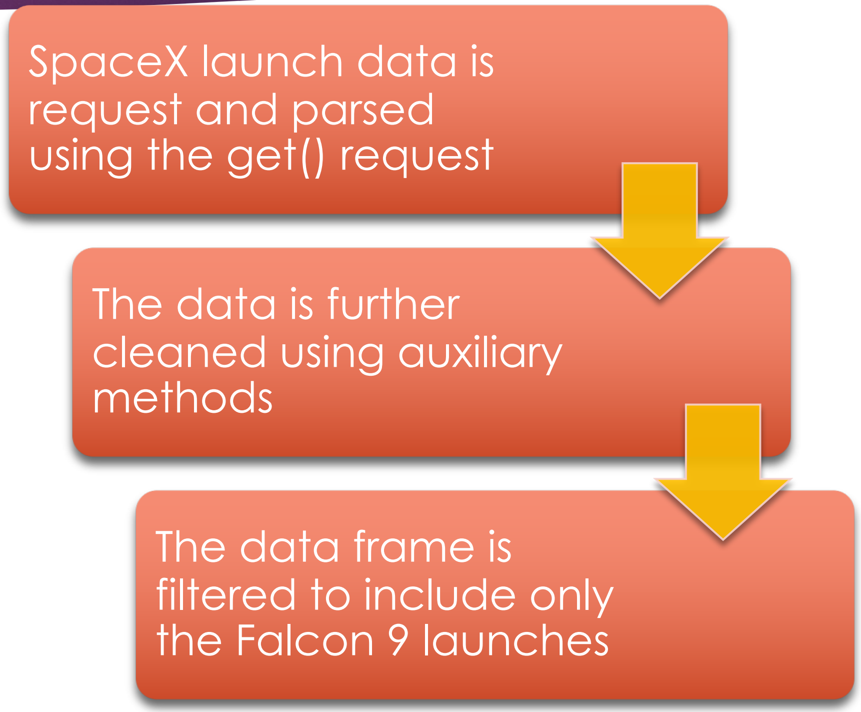
- Map visualization can be created in order to gain a more spatial understanding of location data. Markers can be added, and distances can be calculated using navigational coordinates.

Perform predictive analysis using classification models

- Predictive models are built and tested by splitting data into training and testing sets. The models are built and testing on the training set, allowing for the tuning of parameters. The models are then cross validated on the testing sets for accuracy and to assure there are not problems with over fitting before deployment.

Data collection – SpaceX API

- SpaceX launch data is request and parsed using the get() request. A JSON file is converted to a Pandas data frame and normalized.
- The data is then cleaned and organized in a way that makes it useful to analysis, and irrelevant columns are eliminated.
- The data frame is then filtered to include only the Falcon 9 launches and all missing values for payload mass are replaced using the mean of that column



https://github.com/TonyLikesPhysics/TonyLikesData/blob/main/Data%20Science%20Professional%20Certificate/SpaceX/jupyter-labs-spacex-data-collection-api.ipynb

Data Collection – Web Scraping

- The SpaceX Falcon 9 data is extracted from HTML tables of launch records on Wikipedia.
- The columns extracted from the tables are turned into the attributes of the Pandas data frame to be created.
- The data frame is then written to a a csv file and exported.

https://github.com/TonyLikesPhysics/TonyLikesData/blob/main/Data%20Science%20Professional%20Certificate/SpaceX/Data%20Collection%20with%20Web%20Scraping.ipynb

SpaceX Falcon 9 launch data is extracted from Wikipedia

The extracted data is turned into a Pandas data frame

The data frame saved and exported to a csv file

Data Wrangling

- The SpaceX data is first scaled and normalized.
- Missing values for payload mass were filled with the mean of the column.
- The class column is created from the launch outcomes attribute, where 1 is a successful outcome and 0 is a failure

Data is prepared first by scaling and normalizing

Missing values are added using the mean of the column, or rows removed

Categorical columns are transformed to binary for further analysis

https://github.com/TonyLikesPhysics/TonyLikesData/blob/main/Data%20Science%20Professional%20Certificate/SpaceX/labs-jupyter-spacex-Data%20wrangling.ipynb

Exploratory Data Analysis with Visualization

EDA with data visualization

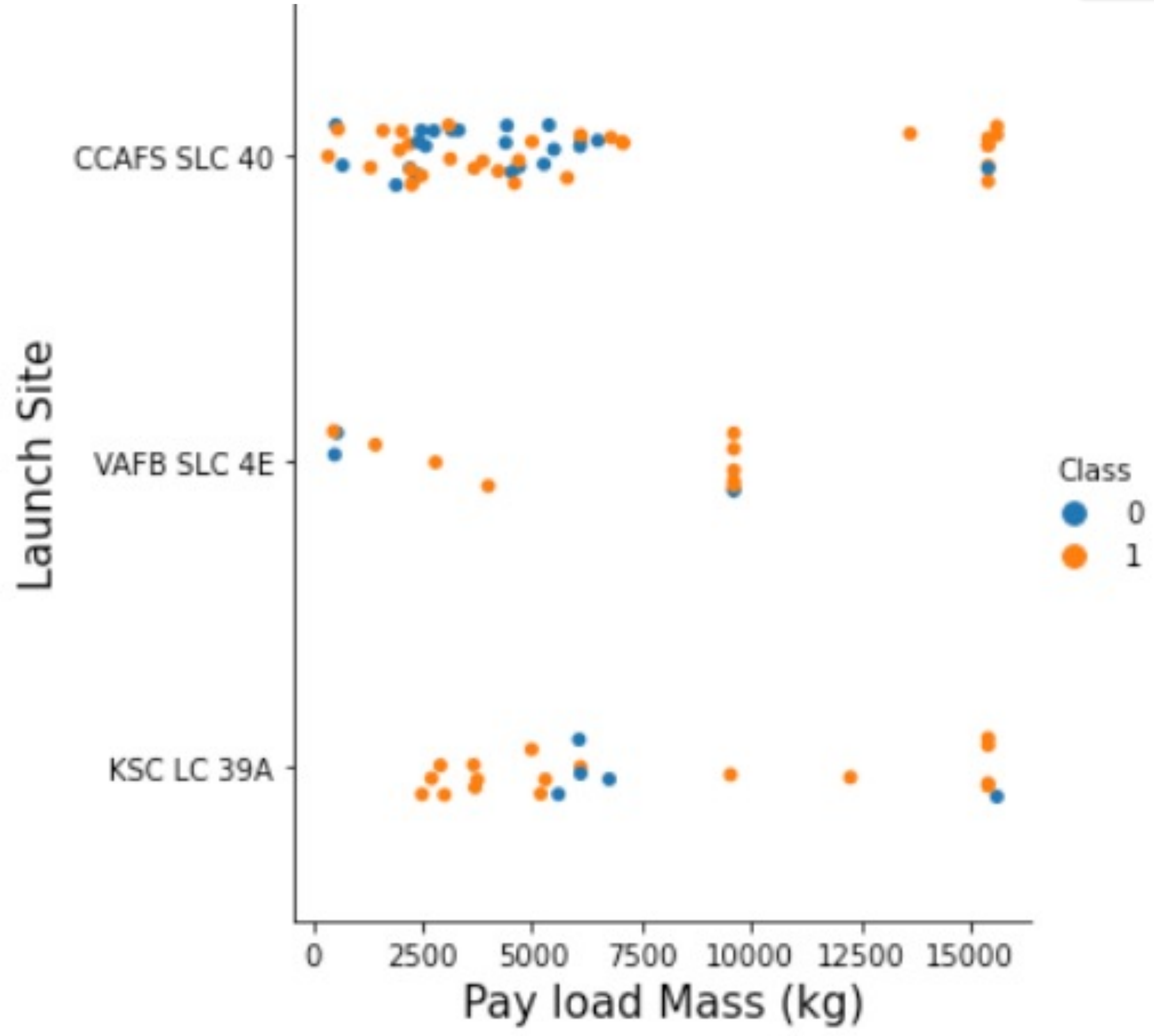
Several charts were created for exploratory data analysis.

- The plot of payload mass versus flight number shows that heavier payloads have higher failed lading rates and the trend that heavier payloads are always being tested.
- The plot of flight number versus launch site shows that certain launch sites have higher success rates.
- The plot of launch site versus payload mass reveals that most of the launches are lighter payloads from Cape Canaveral.
- A bar graph showing the launch success rate for different orbits revealed some orbits are associated with much higher success rate than others.
- The plot of orbit versus flight number shows an evolution of more difficult orbits being attempted. It also shows success to be related to the number of flights for certain orbits but not for others.
- A plot of payload mass for given orbit types show that heavier payload have a negative influence on some orbits and a positive influence on others.
- A final plot of success rate for a given year showed increasing success rates from 2013 to 2020.

https://dataplatform.cloud.ibm.com/analytics/notebooks/v2/9facebbd-1ddc-415b-9c05-96645806dcb2/view?access_token=e8008b14aa2a1ae552af81806c1ff6deb16a50da14a8f8dd384756070ee00a06

Payload vs. Launch Site

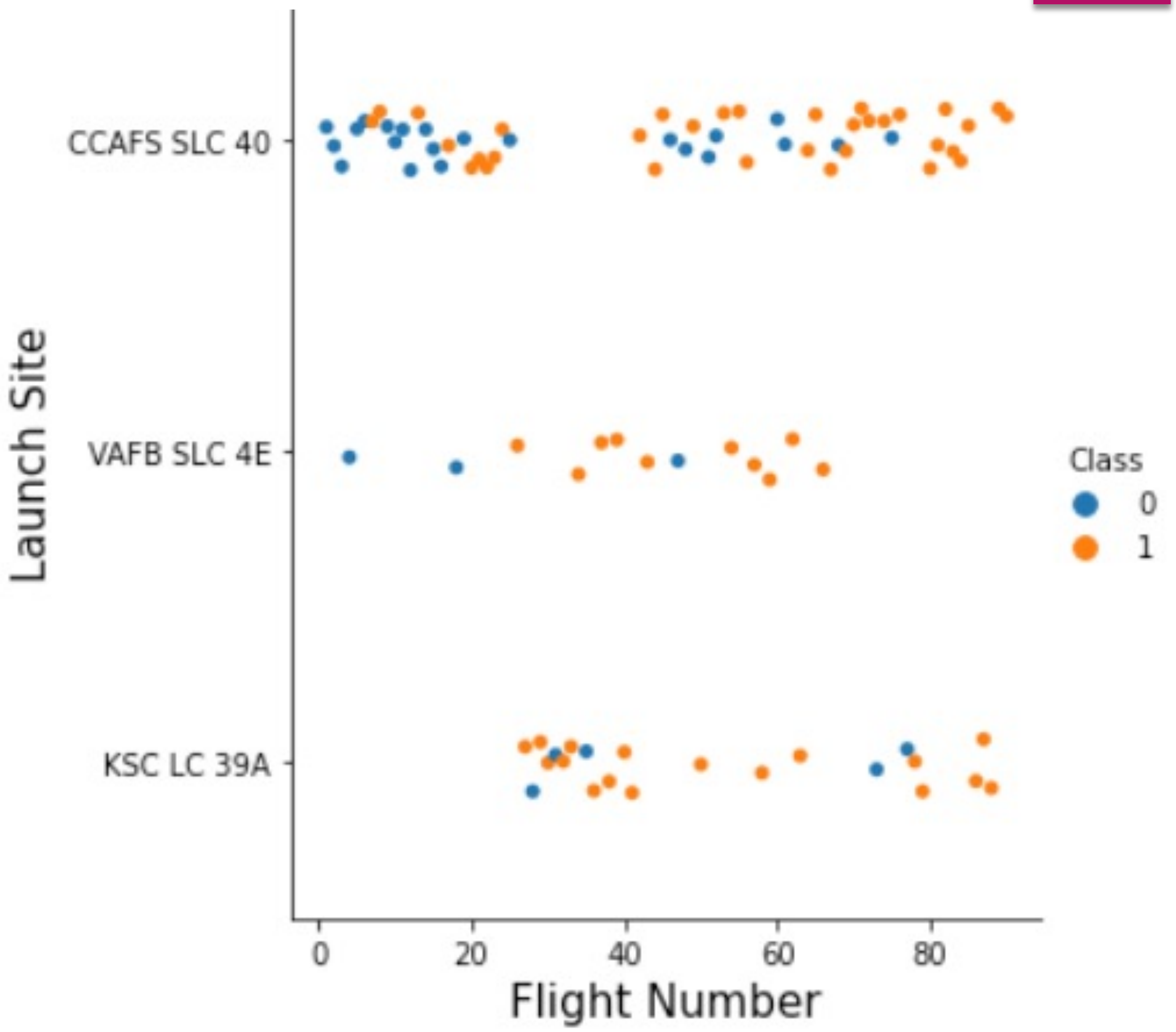
- Scatter plot of Payload vs. Launch Site. The plot shows heavier payloads are less successful from all launch sites.
- The plot also shows that most of the launches are from the Cape Canaveral launch site having lighter payloads.



Flight Number vs. Launch Site

- Scatter plot of Flight Number vs. Launch Site. The plot shows most of the flight coming from CCAFS SLC-40, but at a lower success rate, at 60%, than the other two sites having 77% success.

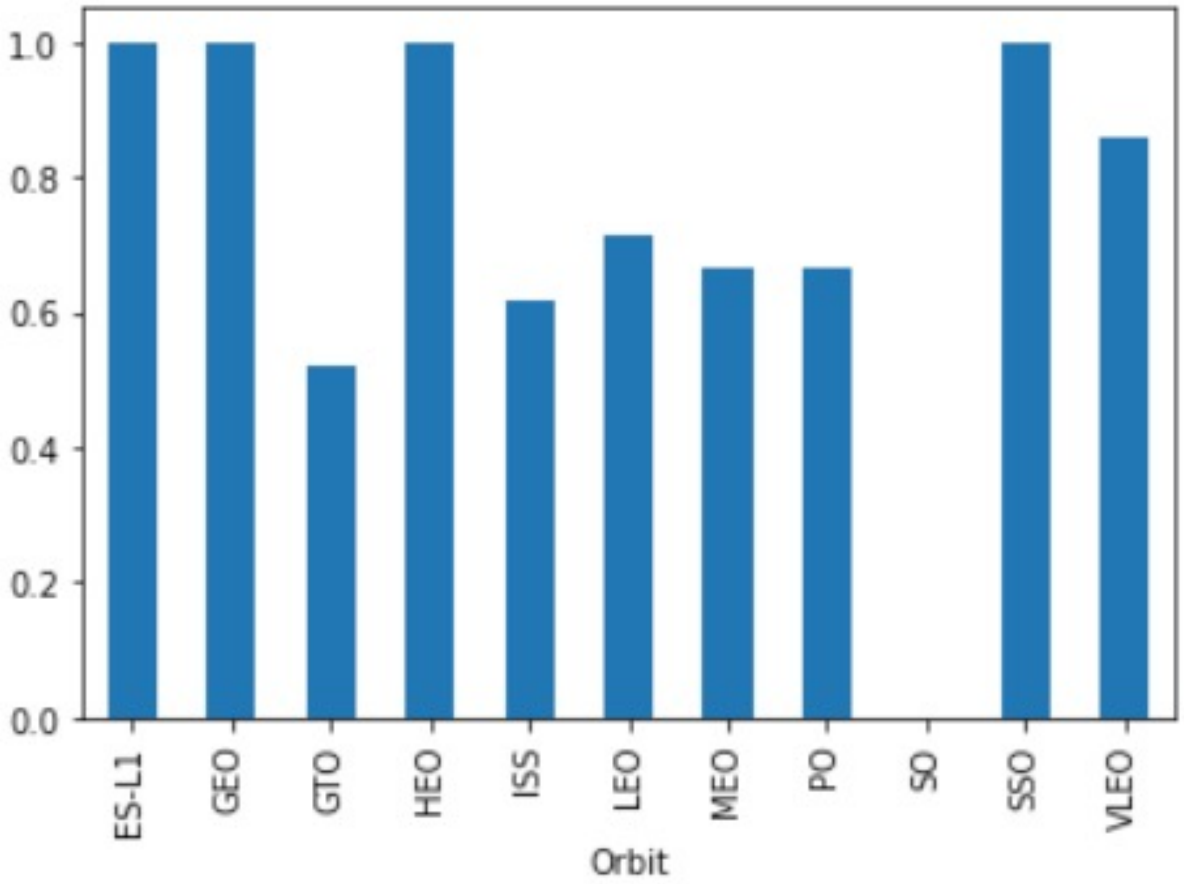
- The plot also shows that mission outcomes become less successful as the number of flights increases.



Success rate vs. Orbit type

- Bar chart showing the success rate of each orbit type. The chart reveals that geosynchronous transfer orbits has the lowest success rate among the lowest in terms of success rate, however, a majority of the launches fall into this category.

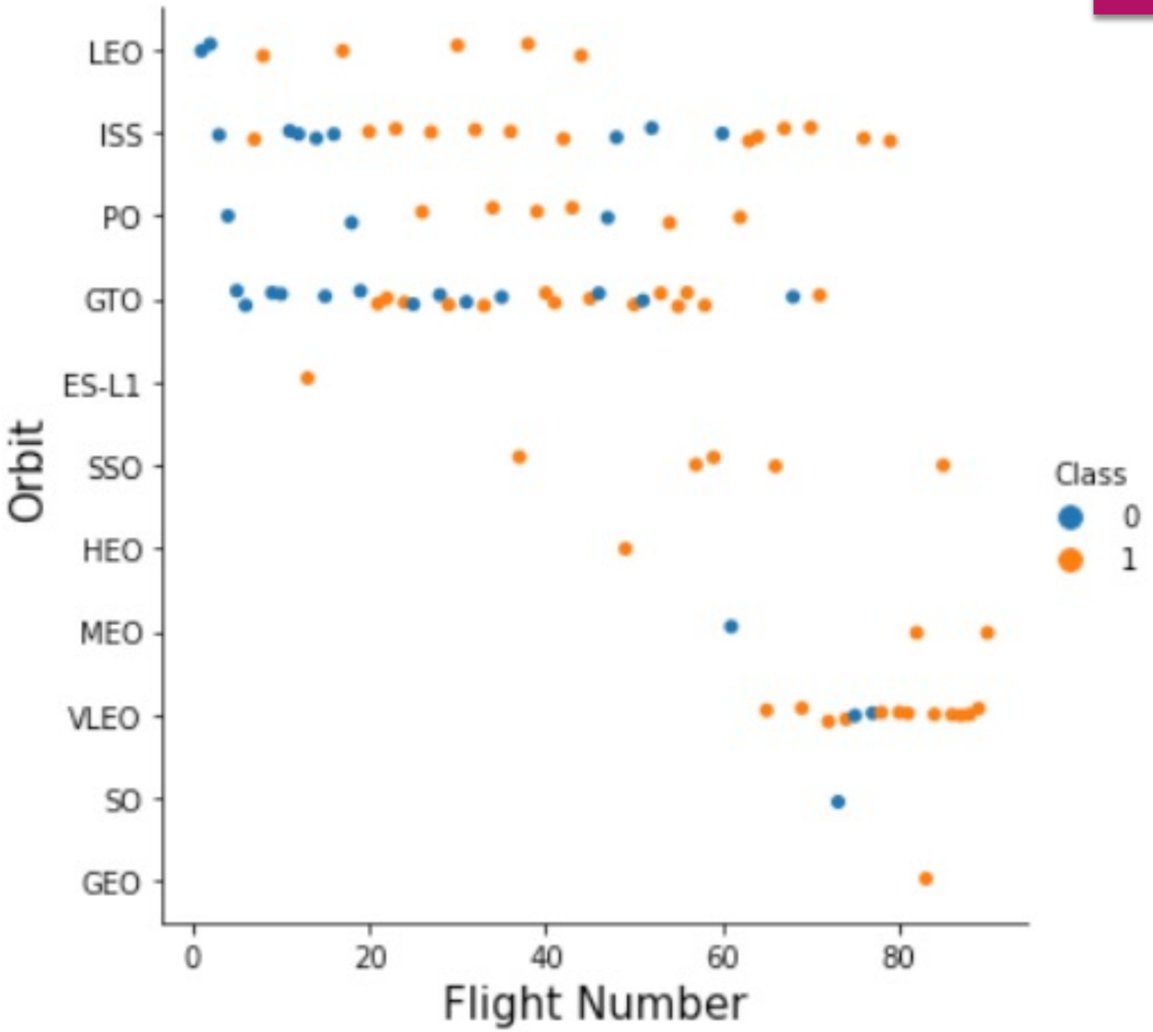
- Some of the orbits involve very few launches, so this data should be observed along with the plot of orbit vs. flight number to get a better sense of the data



Flight Number vs. Orbit type

- Scatter plot of flight number vs. orbit type. This plot shows that GTO and ISS launches are majority of launches.

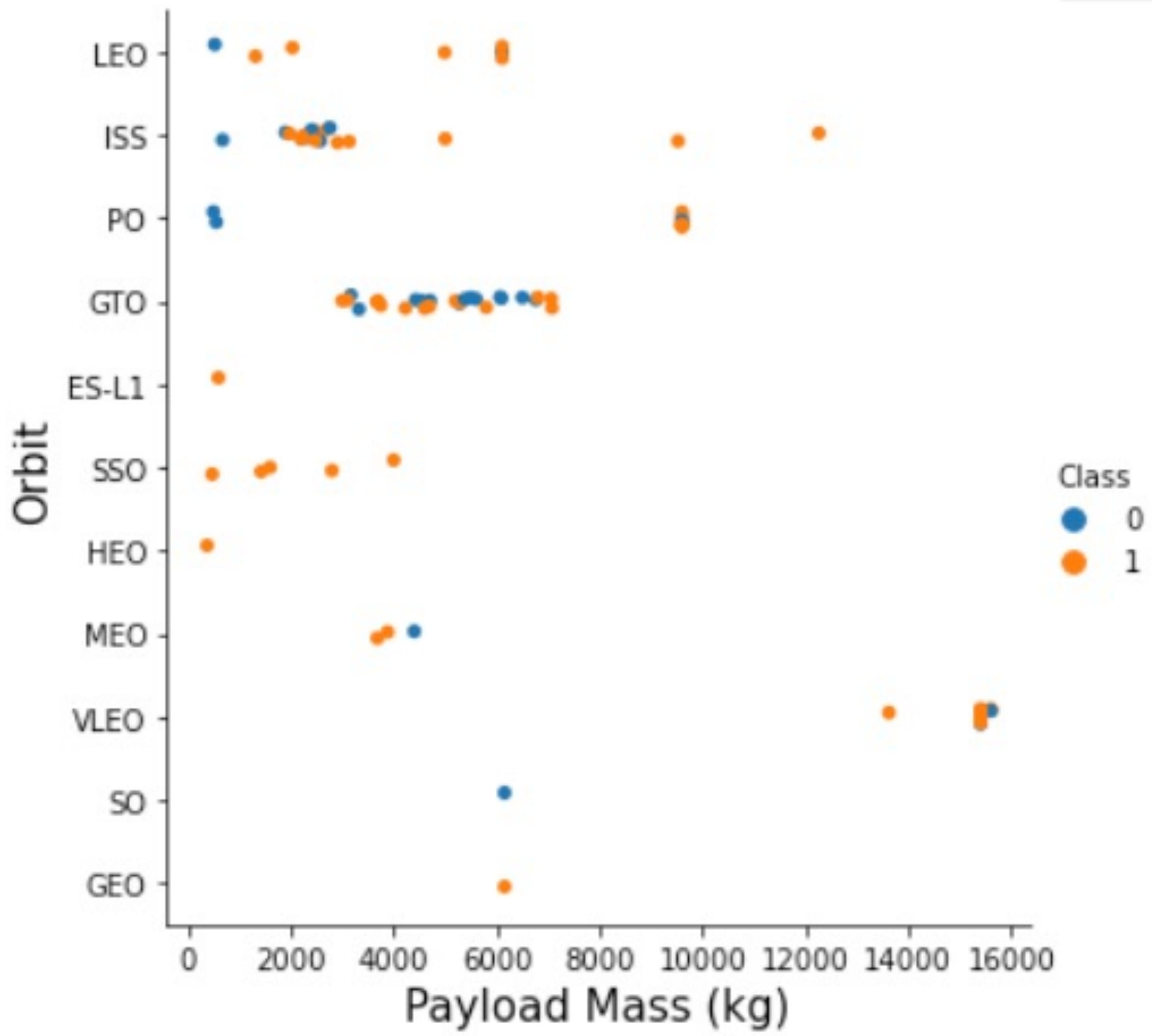
- The plot also shows that mission success seems to be linked to the number of flights for LEO where there is no such relationship for GTO.



Payload vs. Orbit type

- Scatter plot of payload mass vs. orbit type. The plot reveals that heavy payloads have a negative correlation with LEO, ISS, and polar orbits, while GTO and VLEO might show positive correlation.

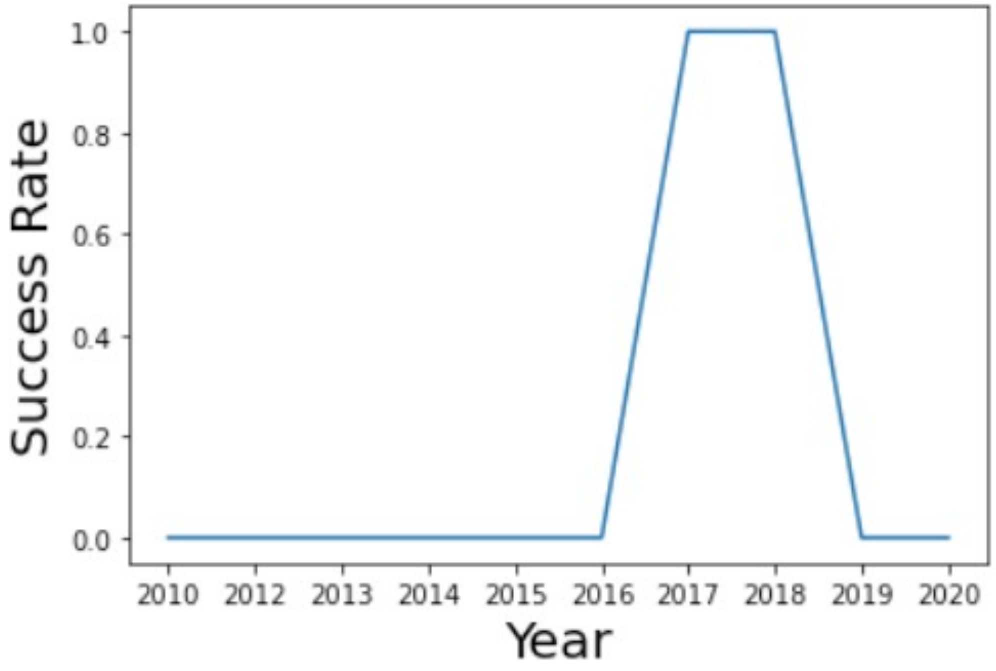
- This plot, like the last two orbit visualizations vs. flight number and success rate, should be observed among the others , including the one shown here, for a better understanding.

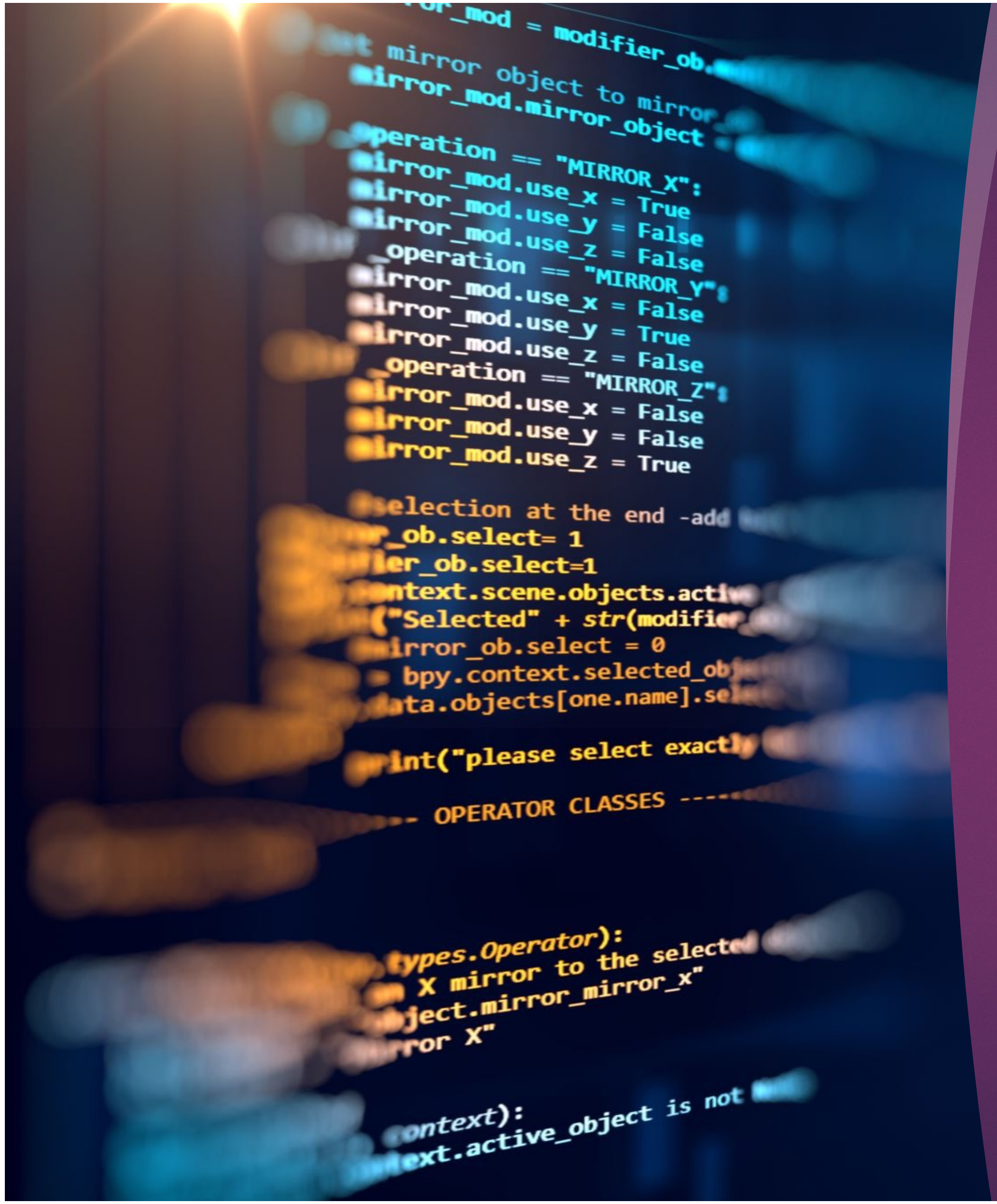


Launch success yearly trend

- Line chart of yearly average success rate. The line plot shows that the success rate rises between 2016 and 2017. The success rate then falls between 2018 and 2019.

- Line plots like these can be deceptive, and there can be missing information between data points that are not illustrated by the line that connects them. A scatter plot may be more objective.





Exploratory Data Analysis with SQL

EDA with SQL

After loading the data into a database, several queries were preformed on the dataset using SQL.

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

https://github.com/TonyLikesPhysics/TonyLikesData/blob/main/Data%20Science%20Professional%20Certificate/SpaceX/jupyter-labs-eda-sql-coursera.ipynb

All launch site names

- To find the names of the unique launch sites, the following query was used:

```
%sql select distinct LAUNCH_SITE from SPACEXTBL
```

- The result shown here illustrates how the Cape Canaveral launch site name has a few different versions

launch_site
CCAFS LC-40
CCAFS SLC-40
CCAFSSLC-40
KSC LC-39A
VAFB SLC-4E

Launch site names begin with `CCA`

- To find all launch sites that begin with `CCA` the following query was used:

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

- The result of the query is shown here in the table to the right

launch_site
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40

Total payload mass

- To calculate the total payload carried by boosters from NASA, the following query was used:

```
%sql select sum(PAYLOAD_MASS__KG_) as
Total_Payload_Mass_Kg from SPACEXTBL where
CUSTOMER = 'NASA (CRS)'
```

total_payload_mass_kg
45596

- The result of the query is shown here and is in units of kilograms

Average payload mass by F9 v1.1

- To calculate the average payload mass carried by booster version F9 v1.1, the following query was used:

```
%sql select avg(PAYLOAD_MASS__KG_) as average_payload_mass_kg from SPACEXTBL where BOOSTER_VERSION = 'F9 v1.1'
```

- The result of the query is shown here and is in units of kilograms

average_payload_mass_kg
2928

First successful ground landing date

- To find the date when the first successful landing outcome in ground pad, the following query was used:

```
%sql select min(DATE) as first_successful_landing from SPACEXTBL where LANDING__OUTCOME = 'Success (ground pad)'
```

first_successful_landing
2015-12-22

- The result of the query is shown here as December 22nd, 2015

Successful drone ship landing with payload between 4000 and 6000

- In order to list the names of boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000, the following query was employed:

```
%sql select distinct BOOSTER_VERSION from SPACEXTBL where PAYLOAD_MASS__KG_ between 4000 and 6000
```

- The result of the query is shown in the long table here

booster_version
F9 B4 B1040.2
F9 B4 B1040.1
F9 B4 B1043.1
F9 B5 B1046.2
F9 B5 B1046.3
F9 B5 B1047.2
F9 B5 B1048.3
F9 B5 B1051.2
F9 B5 B1058.2
F9 B5B1054
F9 B5B1060.1
F9 B5B1062.1
F9 FT B1021.2
F9 FT B1031.2
F9 FT B1032.2
F9 FT B1020
F9 FT B1022
F9 FT B1026
F9 FT B1030
F9 FT B1032.1
F9 v1.1
F9 v1.1 B1011
F9 v1.1 B1014
F9 v1.1 B1016

Total number of successful and failed mission outcomes

- To calculate the total number of successful and failed mission outcomes, the following query was used:

```
%sql select count(MISSION_OUTCOME) as
total_mission_outcomes from SPACEXTBL
```

total_mission_outcomes
101

- The result of the query is shown here as 101 total mission outcomes

Boosters carried maximum payload

- To list the names of the booster which have carried the maximum payload mass, the following query was employed:

```
%sql select distinct BOOSTER_VERSION,
PAYLOAD_MASS__KG_ from SPACEXTBL where
PAYLOAD_MASS__KG_ = (select
max(PAYLOAD_MASS__KG_) from SPACEXTBL)
```

- The result of the query is shown in the table to the right

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

2015 launch records

- To list the records which will display the month names, failure landing outcomes in drone ship, booster versions, and launch site for the months in the year 2015, the following query was used:

```
%sql select monthname(DATE) as "month", year(DATE) as "year", LANDING__OUTCOME, BOOSTER_VERSION, LAUNCH_SITE from SPACEXTBL where LANDING__OUTCOME = 'Failure (drone ship)' and year(DATE) = '2015'
```

- The resulting table is shown here

month	year	landing__outcome	booster_version	launch_site
January	2015	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
April	2015	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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

- In order to rank the count of successful landing outcomes between the date 2010-06-04 and 2017-03-20 in descending order, the following query was used:

```
%%sql select LANDING__OUTCOME, count(*) as
"count" from SPACEXTBL where
LANDING__OUTCOME like 'Success%' and DATE
between '2010-06-04' and '2017-03-20' group
by LANDING__OUTCOME order by count(*) desc
```

- The resulting table is shown here

landing__outcome	count
Success (drone ship)	5
Success (ground pad)	3

Interactive Map with Folium

Build an interactive map with Folium

- Several map objects such as markers, circles, lines were created and added to a folium map. These were used to mark and label launch sites that SpaceX uses. Using the Folium map to gain insight into the data was fun!

- Markers were used to label locations on the map and to make these locations easy to spot from any zoom level. Circles were used to encircle the specific areas being marked off making them easy to spot and providing for a reference radial distance. Lines were used to mark distances calculated between sets of location coordinates.

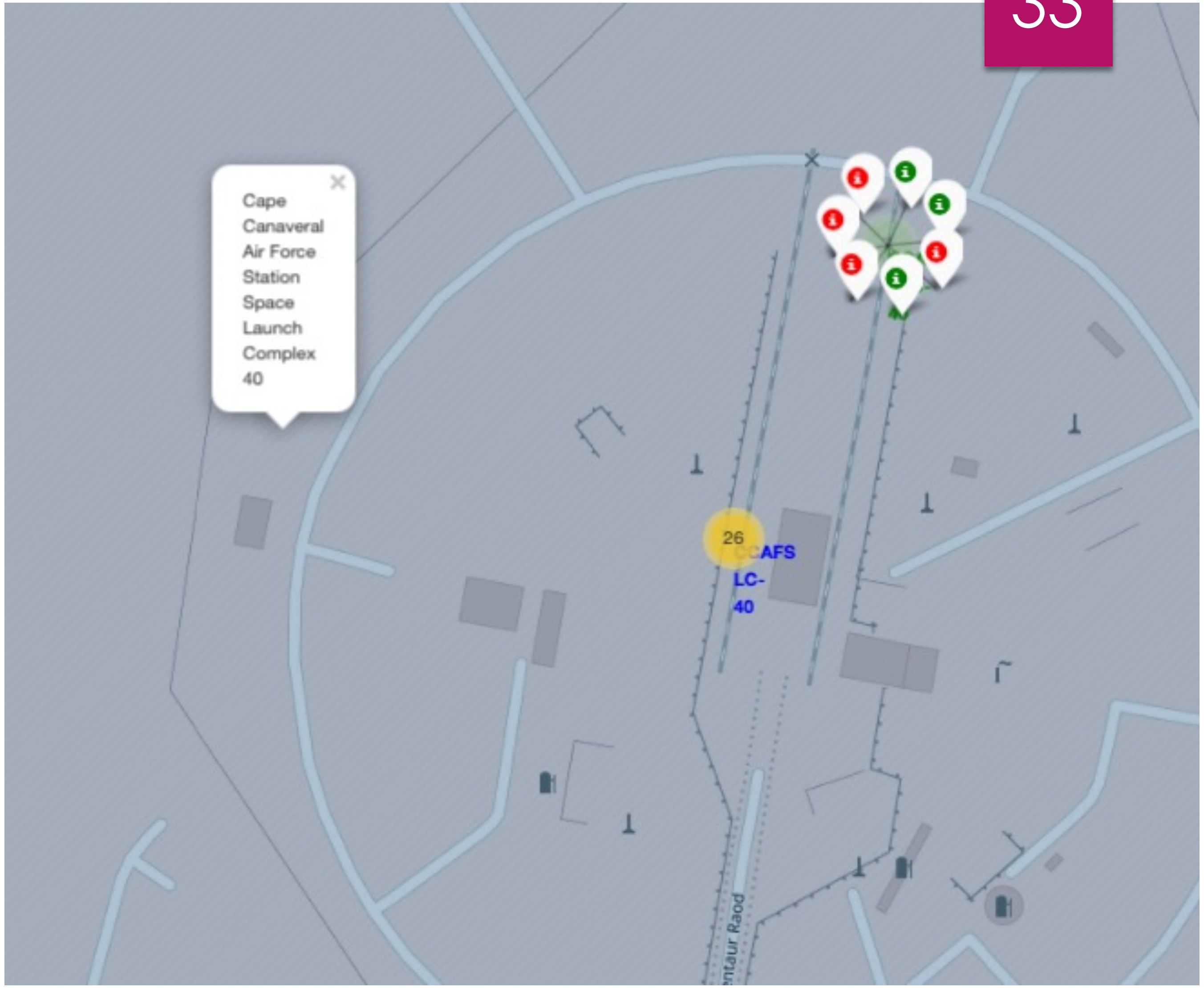
https://github.com/TonyLikesPhysics/TonyLikesData/blob/main/Data%20Science%20Professional%20Certificate/SpaceX/lab_jupyter_launch_site_location.ipynb

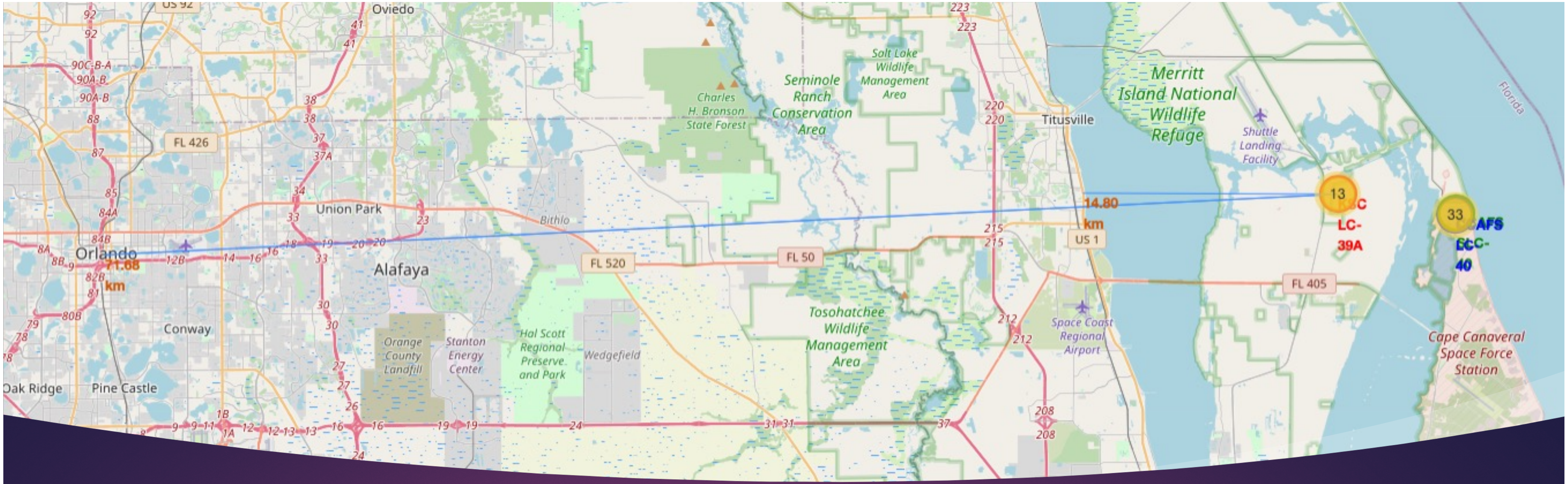
Global View of Launch Sites

- This view of the map shows how the marker objects make the sites on the map easy to see even from a global view
- The marker labels were given different colors to make them easy to differentiate

Launch Record Labels

- The picture shown here illustrates the use of marker clusters to represent a record of both successful and failed launches that can be seen on the map at each site
- Adding these elements to the map makes them highly interactive and informative





Distance to Railway and City

- The distances to what is believed to be the nearest railway and to the nearest large city, Orlando, are calculated using the location coordinates
- The distances are labeled in kilometers and shown above. Both distances are calculated from Kennedy Space Center launch site

Building a Dashboard with Plotly Dash

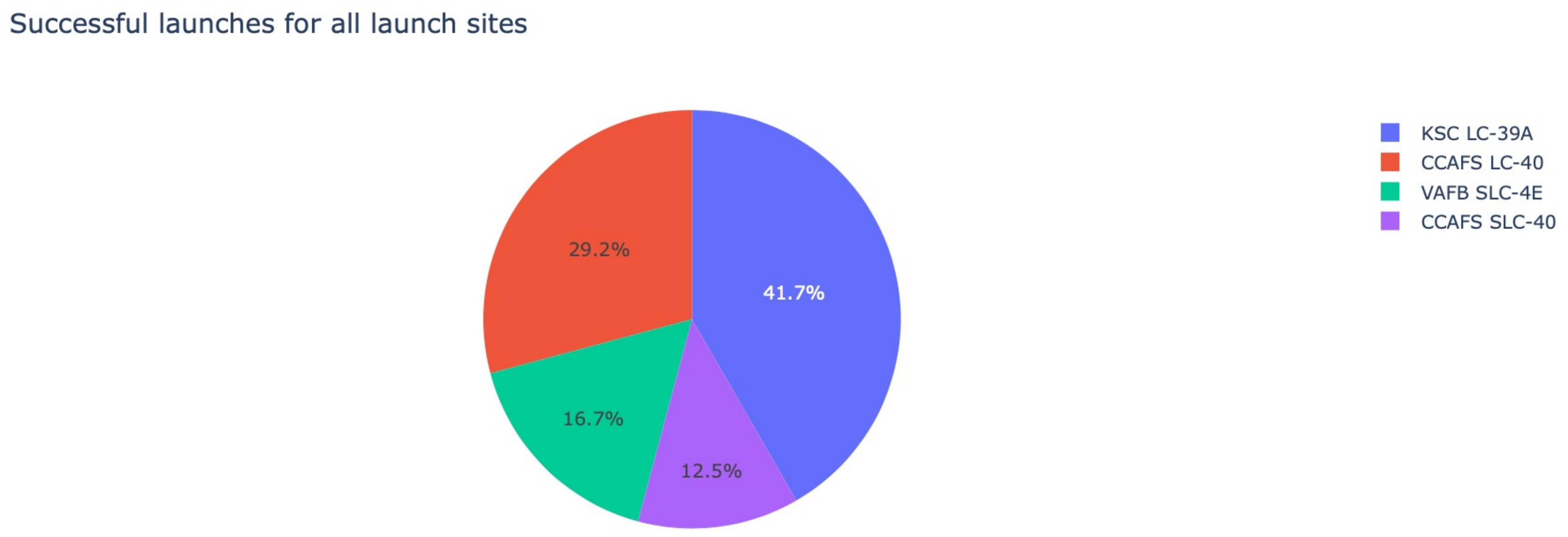
Build a Dashboard with Plotly Dash

- Several plots/graphs and interactions were added to the SpaceX launch records dashboard. A dropdown menu providing a choice of launch site was added.
- A pie chart is added showing successful launches versus failed ones for individual sites and the percentage of successful launches among all sites when selecting an ‘all site’ option from the dropdown menu.
- A scatter plot of payload mass versus the launch success was also added along with a slider object that interactively filters the range of the payload masses plotted in the graph

https://github.com/TonyLikesPhysics/TonyLikesData/blob/main/Data%20Science%20Professional%20Certificate/SpaceX/SpaceX_Dash_app.ipynb

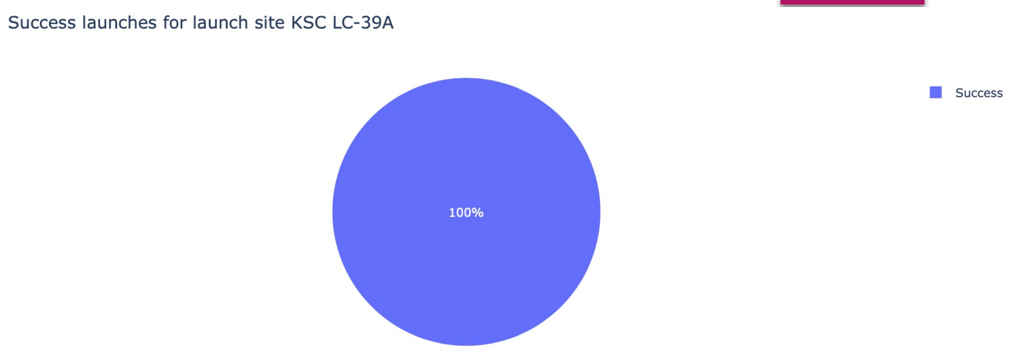
All Sites Dropdown Selection

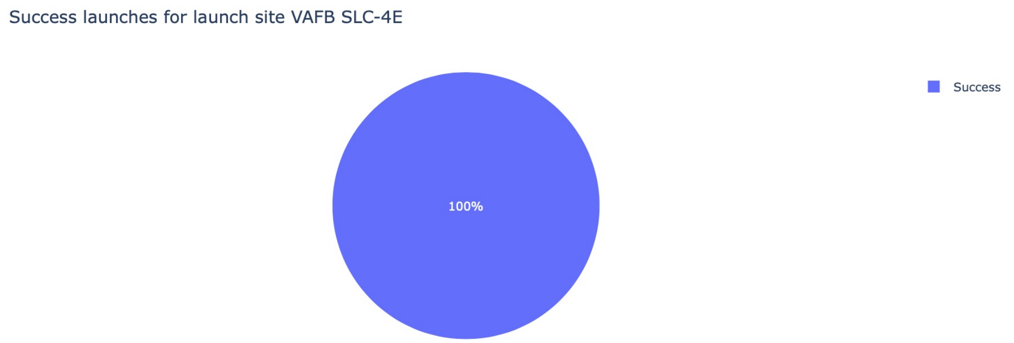
- The pie chart shown is the result of choosing the 'all sites' dropdown menu option
- The pie chart shows the percentage of successful outcomes among all the launch sites in the data set.



Highest Launch Success Ratio

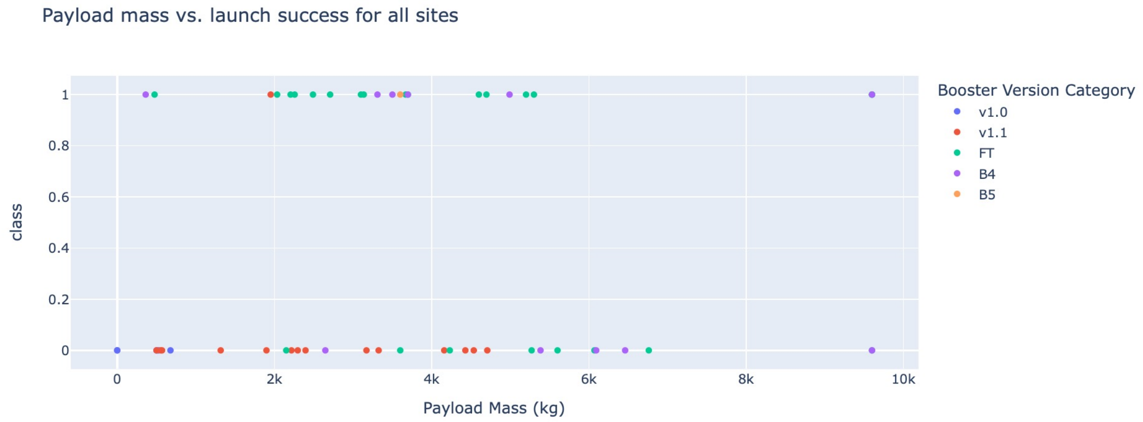
- The Kennedy Space Center and Vandenberg AFB launch sites show a 100% success rate. The charts were produced by choosing these launch sites from the dropdown menu.

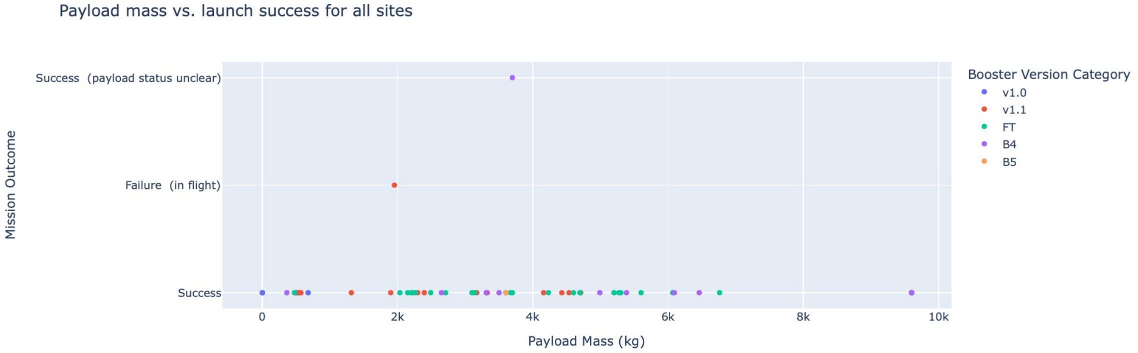




Payload Mass vs. Launch Outcome

- Shown here on top is the plot for the payload mass in kilograms versus the class. While it follows the instructions, I'm not sure this was the best way to plot. To try to better understand what is going on, I also added a plot of the payload mass versus the mission outcome, shown at the bottom.
- The bottom plot also shows how a categorical attribute like mission outcome can be plotted against something numeric like payload mass.





Predictive Analysis: Classification

Predictive Analysis: Classification

- The first step is to import the dataset, normalize it, and split it into training and testing sets of appropriate size.
- After the training/testing split is made, the models are built and tested. The models used include logistic regression, support vector machine, decision tree classifier, and a K nearest-neighbors model.
- Each predictive model is evaluated using the testing portion of the dataset, and the accuracy is recorded. The model with the highest accuracy is chosen

https://github.com/TonyLikesPhysics/TonyLikesData/blob/main/Data%20Science%20Professional%20Certificate/SpaceX/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Load the SpaceX launch data, normalize and split the data into test and train sets

Each predictive model is built individually, using algorithms to find the optimal parameters where it was applicable

Each model is evaluated on the testing set for accuracy and the best model is chosen

Results

- The code shown here is that of the decision tree classifier predictive model.
- The code for parameter tuning and evaluation for each model can be be accomplished with relatively few lines of code.
- The training accuracy results of the predictive analysis for the decision tree model can be seen in the output

```
parameters = {'criterion': ['gini', 'entropy'],
      'splitter': ['best', 'random'],
      'max_depth': [2*n for n in range(1,10)],
      'max_features': ['auto', 'sqrt'],
      'min_samples_leaf': [1, 2, 4],
      'min_samples_split': [2, 5, 10]}

tree = DecisionTreeClassifier()
```

```
tree_cv = GridSearchCV(tree, parameters, cv = 10)
tree_cv.fit(X_train,Y_train)
```

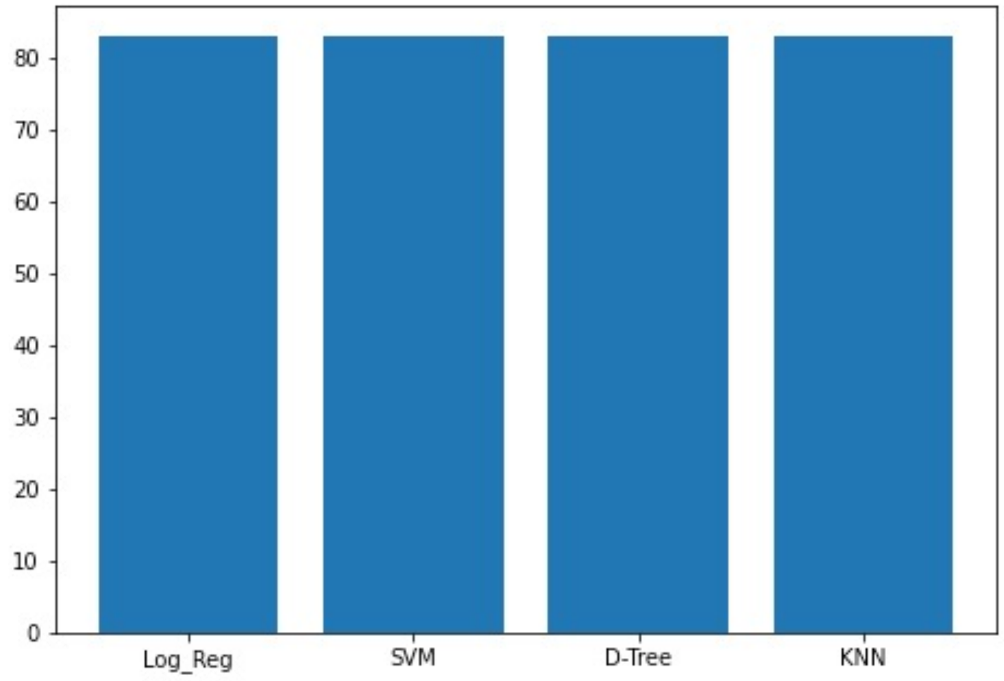
```
GridSearchCV(cv=10, estimator=DecisionTreeClassifier(),
             param_grid={'criterion': ['gini', 'entropy'],
                         'max_depth': [2, 4, 6, 8, 10, 12, 14, 16, 18],
                         'max_features': ['auto', 'sqrt'],
                         'min_samples_leaf': [1, 2, 4],
                         'min_samples_split': [2, 5, 10],
                         'splitter': ['best', 'random']})
```

```
print("tuned hpyerparameters :(best parameters) ",tree_cv.best_params_)
print("accuracy :",tree_cv.best_score_)
```

```
tuned hpyerparameters :(best parameters)  {'criterion': 'gini', 'max_depth':
6, 'max_features': 'sqrt', 'min_samples_leaf': 4, 'min_samples_split': 10, '
splitter': 'best'}
accuracy : 0.875
```

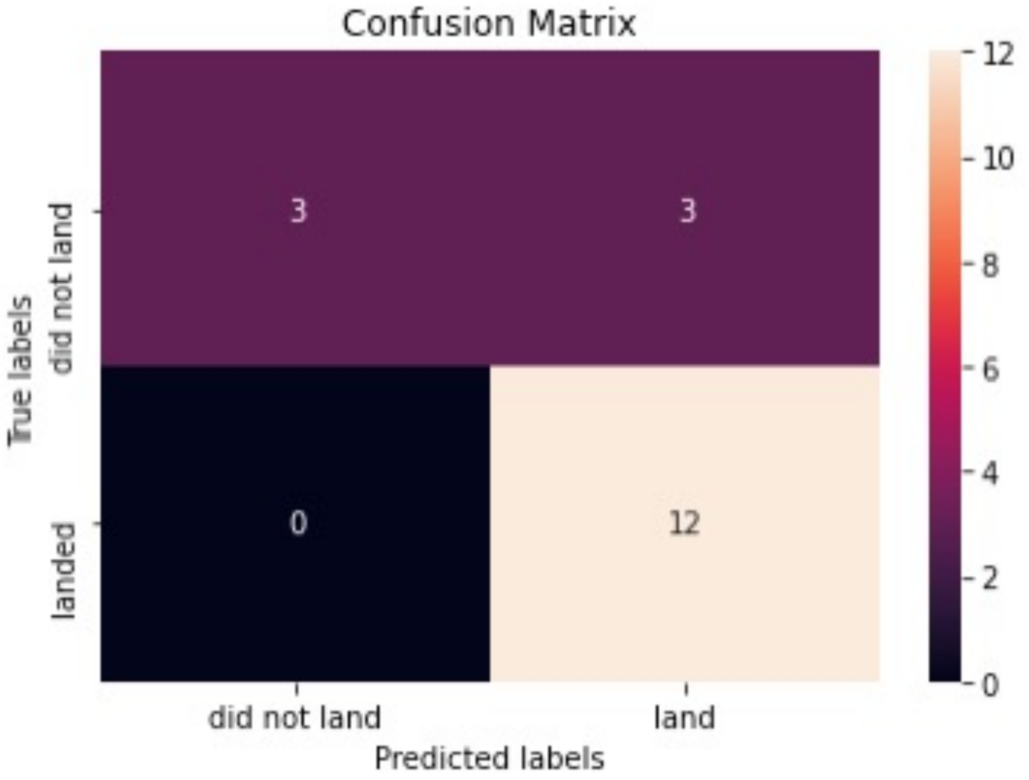
Classification Accuracy

- The model with the highest training accuracy was the decision tree model, but it turned out this this was the model that had the most overfitting based on the cross validation we performed
- The bar graph to the right shows that the testing accuracy for the four models used are all the same, at 83%



Confusion Matrix

- The confusion matrix for all four models is the same as that shown here.
- The matrix plot is a diagram of the success and failure outcomes of the model. These four squares are the false positives and negatives, as well as true positives and negatives.



CONCLUSION

- Data collection can be accomplished by acquiring preproduced datasets or using web scraping techniques to build data sets from websites
- The data must then be processed and cleaned in order to make it useful, using web wrangling techniques. This might require scaling and normalization, dealing with missing values, and transforming categorical attributes to binary ones.
- Exploratory data analysis should be started by using visualizations to gain insights into the data, and to find relationships between attributes that might otherwise be difficult to perceive.
- This type of exploratory data analysis can also be done using SQL as it was in this project, to gain insights using subsetting and mathematical calculations.
- EDA can also be accomplished using folium maps as it was done for the launch sites in this project example. As well, interactive dashboard can be created using Dash and Plotly, and can be used to make visualizations that can be altered with buttons, sliders, and dropdown menus.
- Predictive analysis can be done using machine learning models such as logistic regression, support vector machine learning, decision tree algorithms and K nearest-neighbors classification algorithms. Models can be evaluated for accuracy before deployment using cross validation techniques.
- This was a fun assignment!

APPENDIX

Notebook links in Watson Studio:

- https://dataplatform.cloud.ibm.com/analytics/notebooks/v2/b3f2559a-6c0a-4249-9d66-1b9d1d4f7920/view?access_token=353b5bd074075bbf037e35d2b4ef850d1444e867d3c289a35f55ac8c5c20cdc3
- https://dataplatform.cloud.ibm.com/analytics/notebooks/v2/6abd83fa-2750-4d9a-b55f-d1fa6478d4f6/view?access_token=331486c6b0fbc4c269178dadeff1f9501738a4a748bb5848ff32070e4aaf5927
- https://dataplatform.cloud.ibm.com/analytics/notebooks/v2/7a2527f0-1691-490a-8530-06f52624505f/view?access_token=d13b9a648d80c2eaf62287f0c270bd18f990d250be89fb8d1c4aa9cc862fa2f0
- https://dataplatform.cloud.ibm.com/analytics/notebooks/v2/9facebbd-1ddc-415b-9c05-96645806dcb2/view?access_token=e8008b14aa2a1ae552af81806c1ff6deb16a50da14a8f8dd384756070ee00a06
- https://dataplatform.cloud.ibm.com/analytics/notebooks/v2/d77fc6da-fc32-420b-b8b0-b5829302b0d7/view?access_token=fd605fab498f190b60a51f2ed0b7a7dcbdef4df62098d4bc6bbc375e8376ad02
- https://dataplatform.cloud.ibm.com/analytics/notebooks/v2/618f531e-beb0-430c-8f04-33d086a14aba/view?access_token=6e17b1ef18da5da934885b3e90cec033bd4d10bbd142cfab647bdeedc5e8096f
- https://dataplatform.cloud.ibm.com/analytics/notebooks/v2/ede6b930-8462-4eb3-b575-226b885f1f8f/view?access_token=13b1a471734717058a8a1ef53366b24c3d766d0f0b15912016bbb371c25551f6
- https://dataplatform.cloud.ibm.com/analytics/notebooks/v2/e0504cbd-284f-4366-ba9e-f3100ae1f3d6/view?access_token=d0a23c59eef46b7933edac5bf3ee258ed4d2bd845b306a01a0a385dcf7e62258

Github project link:

- https://github.com/TonyLikesPhysics/TonyLikesData/tree/main/Data%20Science%20Professional%20Certificate/SpaceX