Winning Space Race with Data Science

Shashank Dutt Sagiraju
03-03-2023

Outline

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

Concise Executive Summary

In this commercial space age, companies are making space travel affordable for everyone. SpaceX is an industry leader and successful organization making this possible. What makes this more intresting is the fact that it comprises a reusable first stage launch vehicle which significantly reduces launching costs for its customers. The amount of savings is close to 100 million dollars that can be saved if the first stage of Falcon9 rocket lands successfully in order for it to be reused.

Ability to predict the successful landing of SpaceX Falcon9 first stage is the subject of this Data Science project so that the company and its partners can pass on significant benefits to their end customers.

Data for this project is collected through two sources:

1. SpaceX's own data site https://api.spacexdata.com which makes available data through custom Public API.

2. Historical records of launches from the Wikipedia website for exploratory analysis through web scraping.

Data collected from these two sources is run through a pre-processing pipeline that cleans the data, removing any inconsistent data, replaces null values with appropriate average or other standard techniques, removes any irrelevant columns, one hot encodes categorical values etc., so that the data is prepared and ready for consumption by AI / ML and Deep learning algorithms.

Preprocessed data is run through a sequence of exploratory data analysis using several plotting and data correlation statistical analysis to identify correlated features that can be used by the predictive algorithms to predict the dependent variable, which is the successful landing of the first stage in this case.

Preprocessed data is split into a training and test set and run through a grid search algorithm to identify hyper-parameters that allow a given algorithm / prediction model to perform best. The ultimate goal of this exercise is to identify the best model with the best hyper-parameters. We have tested Logistic Regression, Support Vector Machine, Decision Tree Classifier and K-Nearest Neighbor models on the test data. A confusion matrix was built to identify the model with best performance on the test dataset.

Results: Amongst the tested Logistic Regression, Support Vector Machine, Decision Tree Classifier and K-Nearest Neighbor models: Decision Tree Classifier: with a 100% Precision to predict Negatives (Failure to Land) and 86% precision to predict Positives (Successful Landing) with an accuracy of 89% with F1-Score of 0.80 for predicting Failure to land (Negative) and 0.92 for predicting successful landing (positive) of first stage Rocket Motor of SpaceX Falcon9.

Decision Tree Classifier:

	precision	recall	f1-score	support
0	0.83	0.83	0.83	6
1	0.92	0.92	0.92	12
accuracy			0.89	18
macro avg	0.88	0.88	0.88	18
weighted avg	0.89	0.89	0.89	18

Introduction

- In the present day commercial space age many private organizations are competing to make space travel affordable for everyone.
- Virgin Galactic is one of those providers with suborbital spaceflights.
- Rocket Lab is also an small satellite provider.
- Blue Origin manufactures sub-orbital and orbital reusable rockets.
- Perhaps the most successful amongst these competitors is SpaceX.
- https://data.spacexdata.com
- SpaceX's accomplishments include:
  - Sending spacecraft to the International Space Station.
  - Starlink, a satellite internet constellation providing satellite Internet access.
  - Sending manned missions to Space.
- One reason SpaceX can do this is the rocket launches are relatively inexpensive due to their reusable rocket systems.
- SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars;
- Other providers cost upwards of 165 million dollars each, much of the savings is because
- SpaceX can reuse its first stage.
- Therefore, if we can determine if the first stage will land, we can determine the cost of a launch.
- This information can be useful if an alternate company wants to bid against SpaceX for rocket launch.
- Spaces X's Falcon 9 launch like regular rockets with re-usable first stage.
- Most unsuccessful landings are planned. SpaceX performs controlled landing in such situations in deep oceans.

- Problems we want to find answers in this Data Science Problem is
  - ***What is the likelihood of successful landing of first stage of Falcon9 rocket so that there could be substantial savings for the customers of SpaceX using its launch facilities.***

Section 1

Methodology

Methodology

- Data collection methodology:

  Data for this Data Science Problem of "Predicting the successful landing of First stage of Falcon-9 Rocket'' Subsystem was collected from two sources:

  1. SpaceX's own data site https://api.spacexdata.com which makes available data through their own Public SpaceX REST API.
  2. Historical records of launches from the Wikipedia website for exploratory analysis through web scraping.

  This API gave us data about launches, including information about the rocket used, payload delivered, launch specifications, landing specifications, and landing outcome. Our goal was to use this data to predict whether SpaceX will attempt to land a rocket or not.
  The SpaceX REST API endpoints, or URL, starts with api.spacexdata.com/v4/. We have the different end points, for example:
  "/capsules" and "/cores", etc., tailored for different sections of data. We were working with the endpoint "api.spacexdata.com/v4/launches/past".

- Perform data wrangling

  We have employed extensively data cleaning, normalization and flattening methodologies for curating the data. we performed some Exploratory Data Analysis (EDA) to find some patterns in the data and determine what would be the label for training supervised models with them.

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

  SQL was extensively employed to search and retrieve data from multiple tables, repositories, and API that employed usage of various criterion. Data visualization techniques of Python **plotly** and **seaborn** libraries was employed which immensely helped to visualize the data for identifying the relationship between various features in the data, the relationship between them, such as correlation etc. It also helped to visualize how the independent variables are related to the dependent variable.

- Perform interactive visual analytics using Folium and Plotly Dash

  Advanced GeoSpatial Analysis using Folium package for Python was employed to analyze the GeoSpatial positioning of the launch sites and their proximity to the Coastline, RoadRail, Highways, Cities, human inhabitation and other installations for safety and security Analysis.

Methodology

- Perform predictive analysis using classification models

Once a curated dataset was created after extensive Exploratory Data Analysis (EDA), Data Wrangling and Visualization. Based upon the observed data characteristics, the features of the data were identified. Both independent variables and dependent variables their correlations, relationships etc., were identified. Data insights helped shortlist four different Artificial Intelligence (AI) and Machine Learning (ML) models for fitting the data. They are:

1. Logistic Regression
2. Support Vector Machine
3. Decision Tree Classifier.
4. K-Nearest Neighbor.

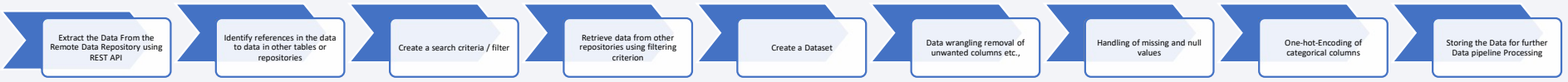
In order to find the best hyperparameters for these models to work at their best, GridSearchCV of SciKitLearn package of Python language was employed. This internal Cross Validation technique helps identify optimal hyper parameters for each of these models.

Data was split into Training and Testing datasets with 80%-20% ratio respectively. Each of the models were employed to fit the Training Dataset in consonance with GridSearchCV hyper-parameter optimization engine. The fitted models were run against test data and the model outcome was captured. The captured outcome was compared to the actual outcome (labeled data). A confusion matrix was created for each of the model outcomes comparing against the labeled outcome.

A bar chart was plotted comparing the performances of each of the models on test data. This required computing a Confusion Matrix of all the models and a best performing model was chosen amongst them that possessed best F1 score for both positive (successful landing) and negative (failure landing) output prediction.

Data Collection

- Data was retrieved as a HTML from https://api.spacexdata.com/v4/launches/past URI, as a HTML document and converted into a Pandas dataset using JASON format as intermediatory formatting. Relevant referenced data was again retrieved from other API endpoints of the SpaceXData API, such as https://api.spacexdata.com/v4/cores/, https://api.spacexdata.com/v4/payloads/, https://api.spacexdata.com/v4/launchpads/, https://api.spacexdata.com/v4/rockets/.
- We have filtered data to include only rockets with single payload, and limiting the data to only those launches before 13th November 2020.
- From other REST API endpoints we have retrieved, Rocket name, Payload mass, Oribit it is launched into, Name of the launch site and its latitude and longitude.
- We have limited the data to only Falcon-9 launches and filter-out any Falcon-1 lanunch data.
- We have reset the FlightNumber serial count since we have dropped some data of Falcon-1.
- Data wrangling is performed to handle null and missing values in the dataset.
- Now this curated data is stored to a disk for further processing down the pipeline.



Data Collection – SpaceX API

- Data collection with SpaceX REST API calls was done as shown in the flowcharts.
- The following URI's was used to collect the Data

https://api.spacexdata.com/v4/launches/past

https://api.spacexdata.com/v4/cores/,
https://api.spacexdata.com/v4/payloads/,
https://api.spacexdata.com/v4/launchpads/,
https://api.spacexdata.com/v4/rockets/.

- Link to the completed Jupyter Notebook on GitHub:
- https://github.com/Shashank-1234/CapstoneProject/blob/b3f682ee35d39a75d9cabd63ed9b374c21e82436/SpaceX-Data-Collection-API.ipynb

Data Collection - Scraping

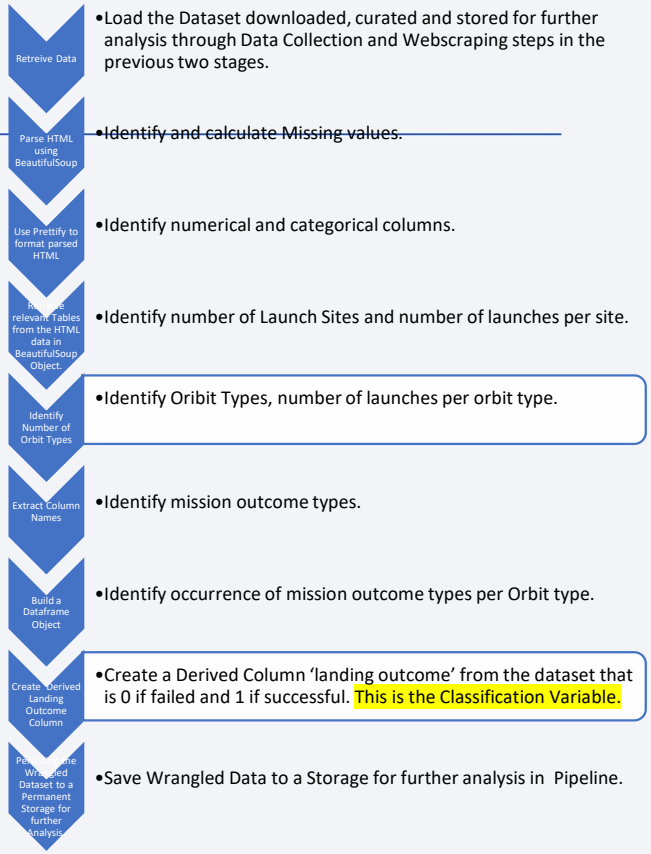
- Webscraping of Falcon-9 and Falcon Heavy Launches from the Wikipedia Website.
- `https://en.wikipedia.org/wiki/List_of_Falcon\_9\_and_Falcon_Heavy_launches`
- `*  Extract a Falcon 9 launch records HTML table from Wikipedia`
- `*  Parse the table and convert it into a Pandas data frame`
- Using 'BeautifulSoup', 'Prettify' for Python the webscraping was simplified.
- Link to the completed Jupyter Notebook on GitHub:
- https://github.com/Shashank-1234/CapstoneProject/blob/b3f682ee35d39a75d9cabd63ed9b374c21e82436/jupyter-labs-webscraping.ipynb

Data Wrangling

- The main objective is to :
- Perform exploratory Data Analysis and determine Training Labels
  - Exploratory Data Analysis
  - Determine Training Labels
- Identify the number and types of landing outcomes and which of them signify success and which failure
- Identify number of numerical and categorical columns.
- Create a new derived column that gives the final 'landing outcome' of success or failure. This is going to be the classification variable
- **This variable will represent the classification variable that represents the outcome of each launch. If the value is zero, the first stage did not land successfully; one means the first stage landed Successfully**
-

- Link to the completed Jupyter Notebook on GitHub:

https://github.com/Shashank-1234/CapstoneProject/blob/b3f682ee35d39a75d9cabd63ed9b374c21e82436/labs-jupyter-spacex-Data%20wrangling.ipynb



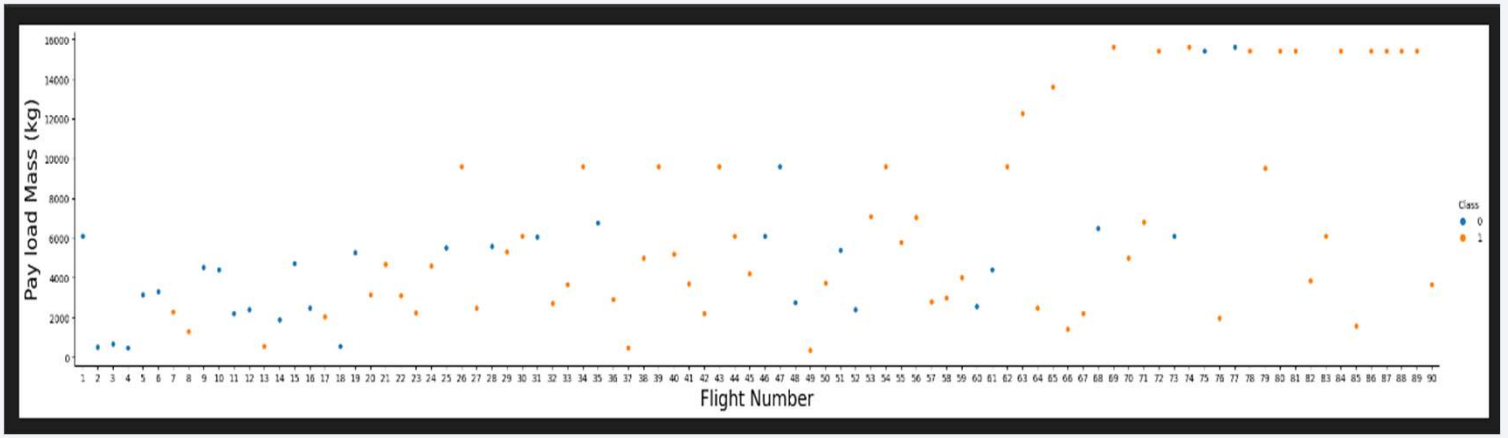
EDA with Data Visualization

- Data Visualization is a very crucial step in any Data Science Project in determining and identifying the related features / columns in a dataset. It is at the heart of Exploratory Data Analysis (EDA) and Feature Engineering.
- It helps identify how the independent variables / features (the influencers) are related to the dependent variable / feature (the influenced variable)
- It helps identify cross correlation between one or more independent features and the dependent variable that is being predicted.
- Link to the completed Jupyter Notebook on GitHub:

https://github.com/Shashank-1234/CapstoneProject/blob/b3f682ee35d39a75d9cabd63ed9b374c21e82436/jupyter-labs-eda-dataviz.ipynb

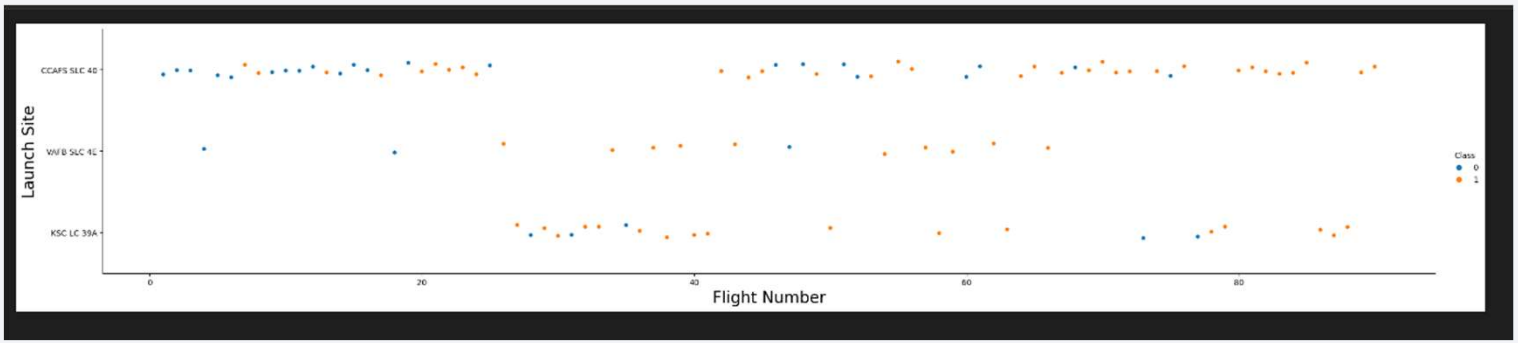
EDA with Data Visualization…

Identifying if the Flight Number has any correlation with the Pay Load Mass (kgs) and how it is related to mission outcome (success or failure)



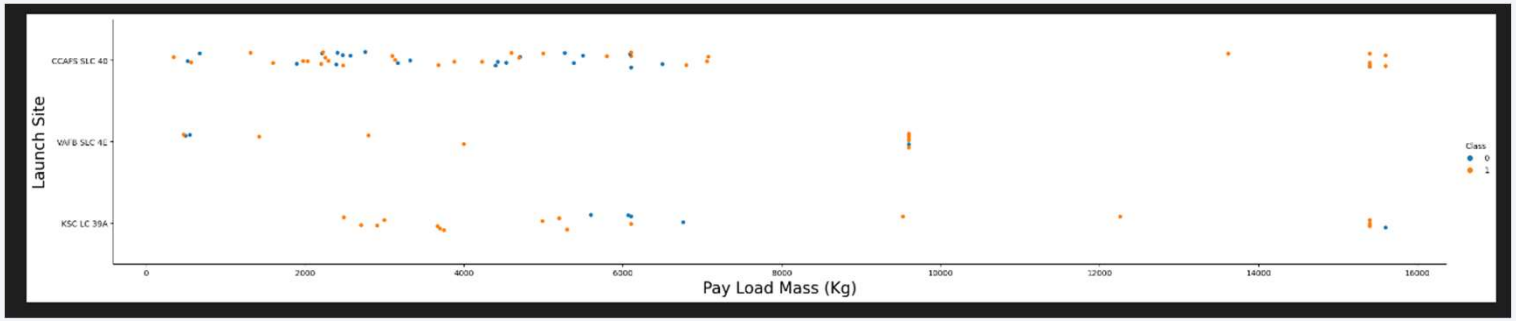
EDA with Data Visualization…

Identifying if the Flight Number has any correlation with the Launch Site and how it is related to mission outcome (success or failure)



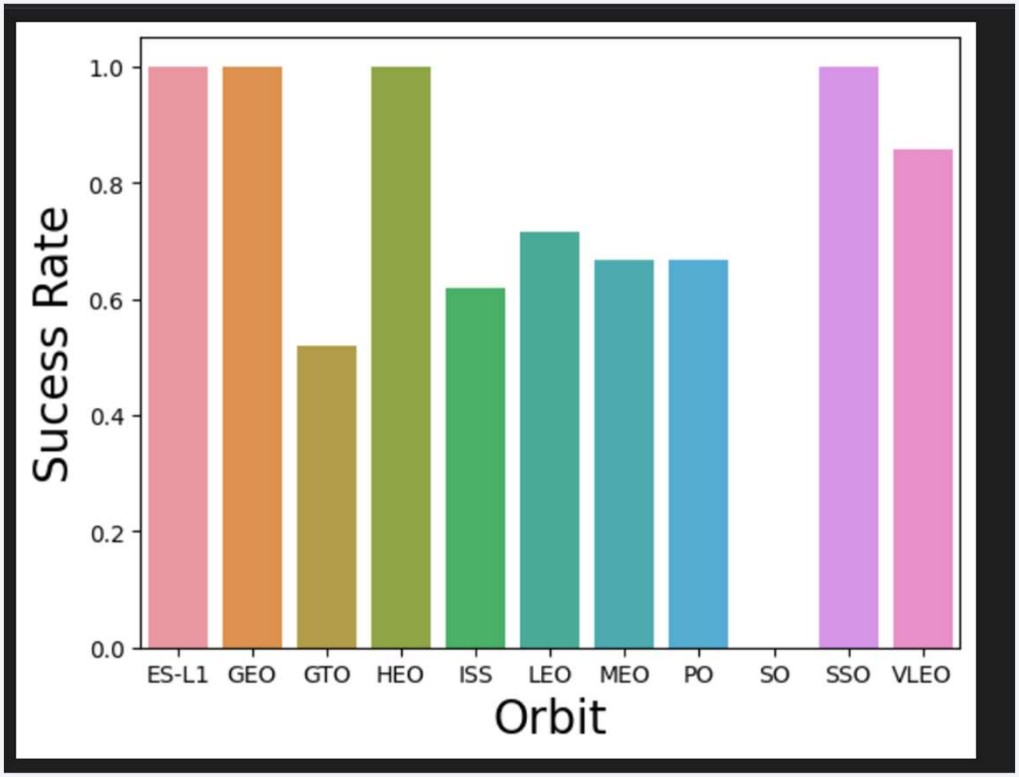
EDA with Data Visualization…

Identifying if the Payload Mass (Kg) has any correlation with the Launch Site and how it is related to mission outcome (success or failure)



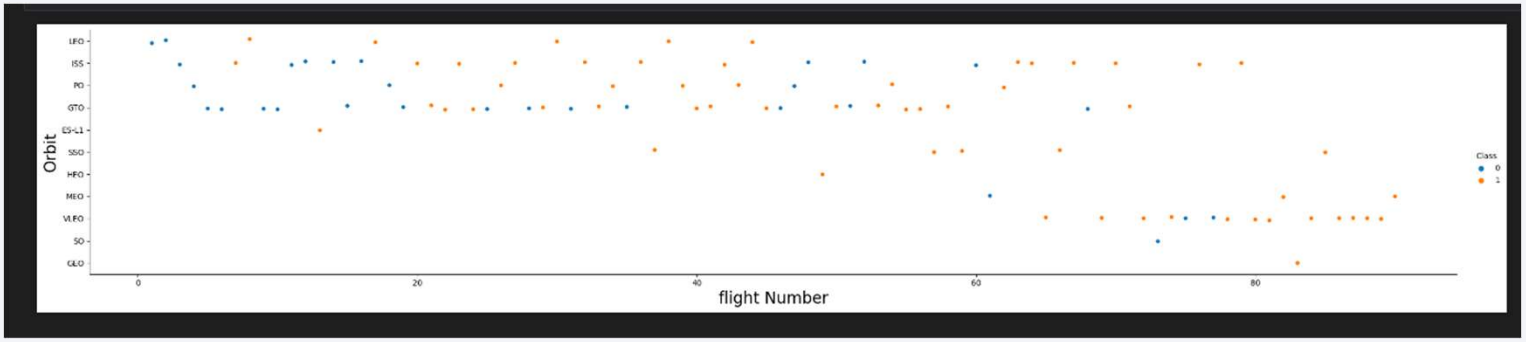
EDA with Data Visualization…

Identifying if the Orbit has any correlation to mission outcome (success or failure)



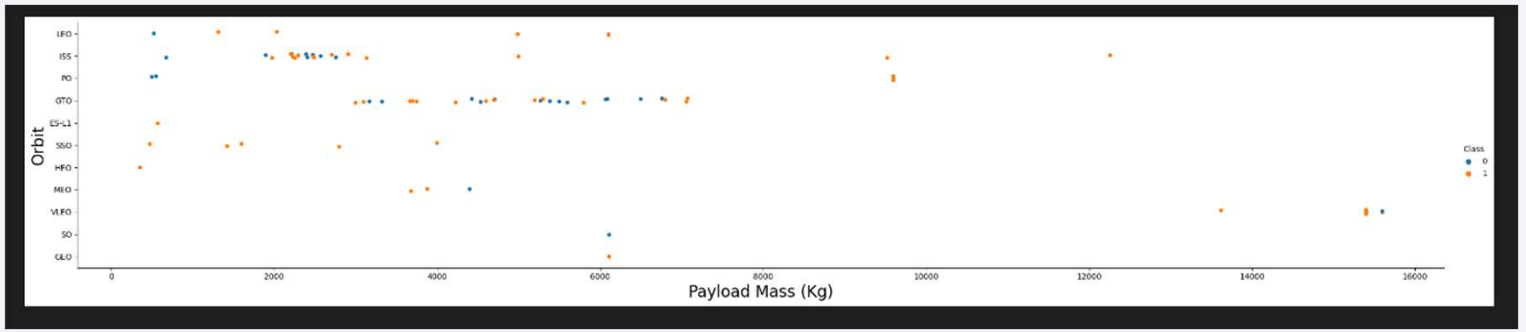
EDA with Data Visualization…

Identifying if the Flight Number has any correlation with the Orbit and how it is related to mission outcome (success or failure)



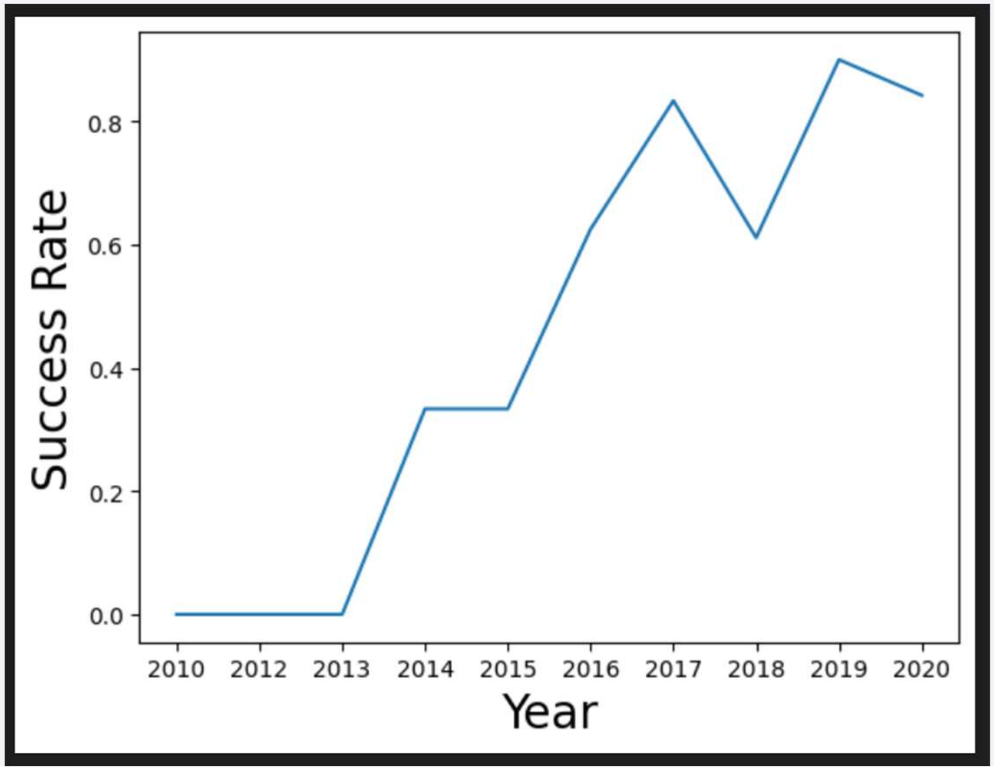
EDA with Data Visualization…

Identifying if the Payload Mass (Kg) has any correlation with the Orbit and how it is related to mission outcome (success or failure)



EDA with Data Visualization…

Identifying if the Success Rate Vs Year



EDA with Data Visualization…

Using One-Hot-Encoding to convert categorical variables to mutually exclusive Boolean columns for consumption by the AI / ML Algorithms

lights	GridFins	Reused	Legs	Block	ReusedCount	Orbit_ES-L1	Orbit_GEO	Orbit_GTO	Orbit_HEO	Orbit_ISS	Orbit_LEO	Orbit_
1	False	False	False	1.0	0	0	0	0	0	0	1	
1	False	False	False	1.0	0	0	0	0	0	0	1	
1	False	False	False	1.0	0	0	0	0	0	1	0	
1	False	False	False	1.0	0	0	0	0	0	0	0	
1	False	False	False	1.0	0	0	0	1	0	0	0	

EDA with Data Visualization…

Converting all the integer columns to float datatype for consumption by the AI / ML Algorithms

```
# HINT: use astype function
features_one_hot = features_one_hot.astype(np.float64)
features_one_hot.head(5)
```

Python

	FlightNumber	PayloadMass	Flights	GridFins	Reused	Legs	Block	ReusedCount	Orbit_ES-L1	Orbit_GEO	Orbit_GTO	Orbi
0	1.0	6104.959412	1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	
1	2.0	525.000000	1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	
2	3.0	677.000000	1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	
3	4.0	500.000000	1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	
4	5.0	3170.000000	1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	1.0	

EDA with SQL

- Summary of the SQL queries performed to explore the Data:

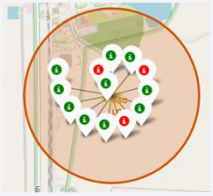
- **Display the names of the unique launch sites in the space mission**
- **Display 5 records where launch sites begin with the string 'CCA'**
- **Display the total payload mass carried by boosters launched by NASA (CRS)**
- **Display average payload mass carried by booster version F9 v1.1**
- **List the date when the first succesful landing outcome in ground pad was acheived.**
- **List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000**
- **List the total number of successful and failure mission outcomes**
- **List the names of the booster_versions which have carried the maximum payload mass. Use a subquery**
- **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.**
- **Rank the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.**

Link to the completed Jupyter Notebook on GitHub:

https://github.com/Shashank-1234/CapstoneProject/blob/b3f682ee35d39a75d9cabd63ed9b374c21e82436/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- GeoMapping and GeoLocating is one of the crucial steps in identifying visually the correlation between the dependent variable and the independent variable. It has been identified in the previous EDA analysis that the Launch Site Location has some correlation to the Mission Outcome. To further this analysis Folium Maps component of Python was employed to visually map the Launch Site, the nearby coastline, railroad, highways, cities and other inhabitations for further analysis.
- Map objects such as markers, circles, lines, etc. were created and added to a folium map of all the launch sites to analyze this:
- Marked all launch sites on a map using their latitude and Longitude values and a highlighted circle area with a text label.


- Marked the success/failed launches for each site on the map using a marker cluster with one marker for each launch, with red color for failure Launch outcome and green color for successful Launch outcome.
- Calculated the distances between a launch site to its proximities such as RailRoad, Highways, Coastlines, Cities and other installations.

Link to the completed Jupyter Notebook on GitHub:

https://github.com/Shashank-1234/CapstoneProject/blob/b3f682ee35d39a75d9cabd63ed9b374c21e82436/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

- A neat Dashboard with Plotly Dash for Python was created that had:
    1. Two main graph components:
        A. A Pie Chart.
        B. A Scatter Plot.
    2. A dropdown menu to select ‘ALL’ or a selected Launch Site of Virgin Atlantic SpaceX Launch Sites, which defaults to ‘ALL’ on startup.
    3. A RangeSlider control to choose ‘Payload Mass (Kgs)’, defaulting to a range of 0-10000. You are free to choose any subrange.
    4. Callback functions and Callback Decorators were added to the Dash Application to update the Graph Objects when the Launch Location and / or Payload Mass (Kg) range changes.
- This Dashboard application was created to explore Landing Success / Failure analysis based upon the choice of Launch location Site and the Payload Mass in (Kgs).

Link to the GitHub Repository to the completed Python Source running the SpaceX Dash Application:

https://github.com/Shashank-1234/CapstoneProject/blob/f1bbaf2e4e3b0b5e6d0705037e54c2402911075e/spacex_dash_app.py

Predictive Analysis (Classification)

- Summarize how you built, evaluated, improved, and found the best performing classification model

- The Model building involved loading the curated Data that was cleaned using Data Wrangling and other analysis in the previous Data Pipeline.
- A labeled Dependent variable, in this case the 'Class' which indicates the success / failure outcome of the landing of the first stage of the Falcon-9 Rocket was copied into an array.
- The Dataset was passed to a StandardScalar Normalizing function which normalized the data in the columns of the Dataset removing the mean and scaling with the Standard Deviation so that no feature will unduly influence the model.
- The Data was split into a 80:20 ratio of training and test subsets.
- A set of four model instances one each for LogisticRegression, Support Vector Machine, Decision Tree Classifier and K-Nearest Neighbor was created.
- Default Hyper-parameters were chosen for each of the models.
- A GridSearchCV object was created and the above model instances with their respective parameters was passed to the GridSearchCV object to find the optimal Hyper-Parameters for best performance.
- Each of the models were fit to the training set.
- Model performance score was calculated.
- Each of the models were run with the test data and the outcome was predicted.
- A confusion matrix was built for each of the model.
- SciKitLearn library function classification_report was passed the confusion matrix of each of the model to calculate the F1-Score of success and failure, the accuracy and the precision for both success and failure of each of the models.
- F1-Score of success and failure, the accuracy and the precision for both success and failure of each of the models were compared and the best performing model was identified based upon the best parameters,
- A bar-chart was plotted with the above parameters for easy visualization.
- Decision Tree Classifier was identified as the best performing model from the perspective of F1-Score for both success and failure and for the Accuracy compared to the other three models for the given dataset.
- LogisticRegression, Support Vector Machine and K-Nearest Neighbor showed next best performance with identical, precision, F1-Score and Accuracy for both success and Failure Outcomes.

Comparison of Performance of Various Models

Logistic Regression:

	precision	recall	f1-score	support
0	1.00	0.50	0.67	6
1	0.80	1.00	0.89	12
accuracy			0.83	18
macro avg	0.90	0.75	0.78	18
weighted avg	0.87	0.83	0.81	18

Decision Tree Classifier:

	precision	recall	f1-score	support
0	0.83	0.83	0.83	6
1	0.92	0.92	0.92	12
accuracy			0.89	18
macro avg	0.88	0.88	0.88	18
weighted avg	0.89	0.89	0.89	18

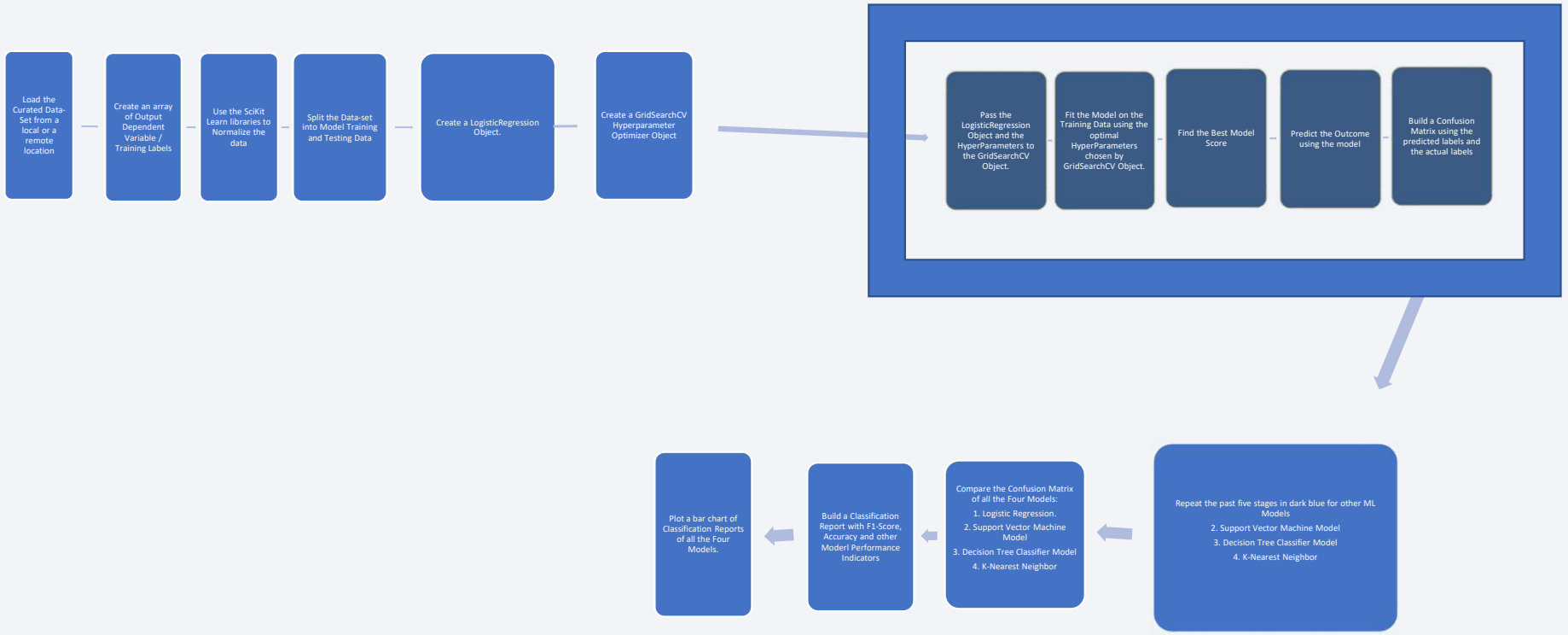
Support Vector Machine:

	precision	recall	f1-score	support
0	1.00	0.50	0.67	6
1	0.80	1.00	0.89	12
accuracy			0.83	18
macro avg	0.90	0.75	0.78	18
weighted avg	0.87	0.83	0.81	18

K-Nearest Neighbor:

	precision	recall	f1-score	support
0	1.00	0.50	0.67	6
1	0.80	1.00	0.89	12
accuracy			0.83	18
macro avg	0.90	0.75	0.78	18
weighted avg	0.87	0.83	0.81	18

Predictive Analysis (Classification)



Predictive Analysis (Classification) ….

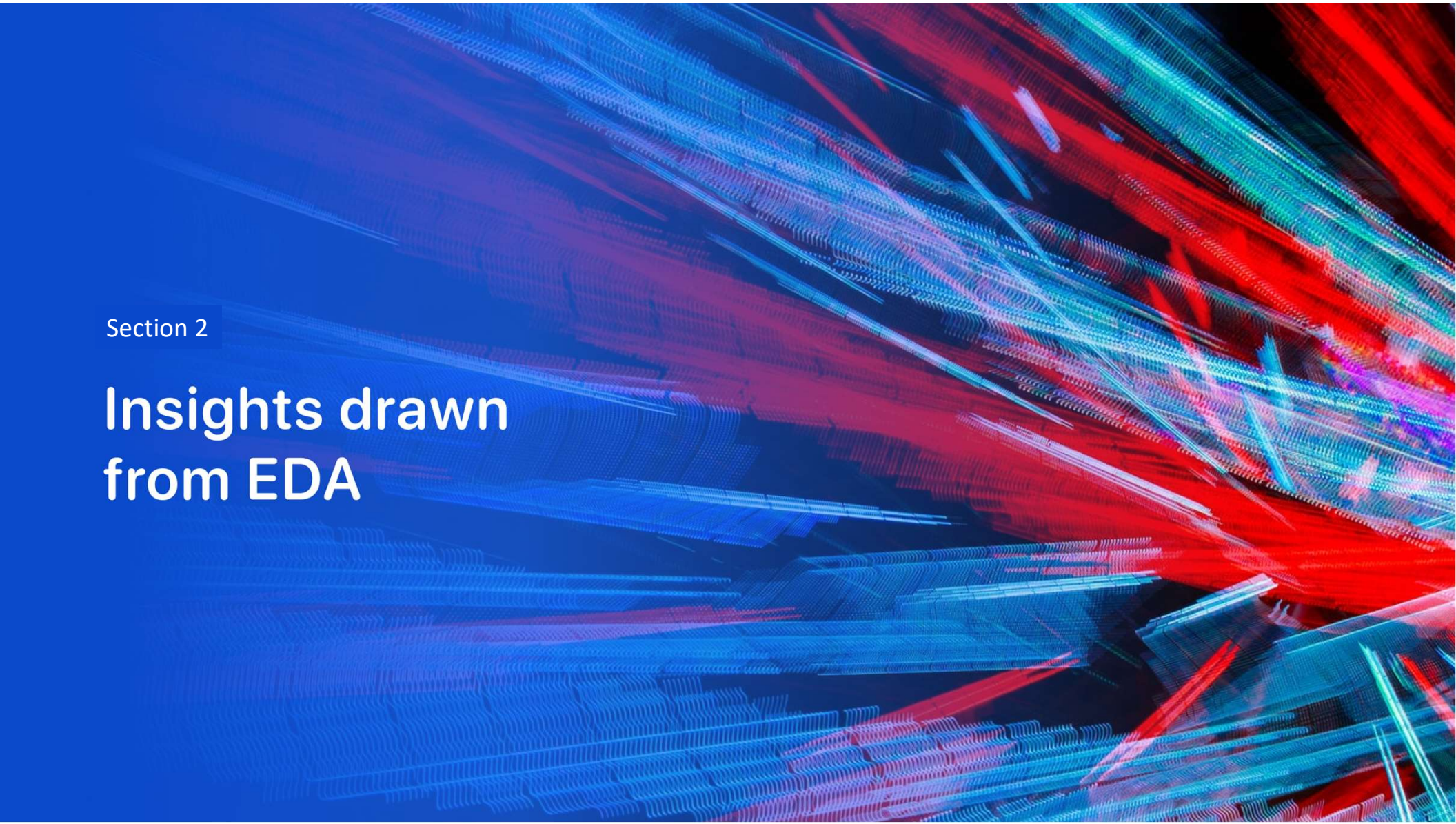
Link to the GitHub Repository to the completed Python Source Code that build various AI / ML models and compared the performance on the SpaceX Dataset :

https://github.com/Shashank-1234/CapstoneProject/blob/4ae6ef8f34d9678e11f645b24f1281e53963d4d6/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results

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

- The following slides summarize the analysis results and the screen capture of exploratory data analysis, Interactie analytics and Predictive analysis.

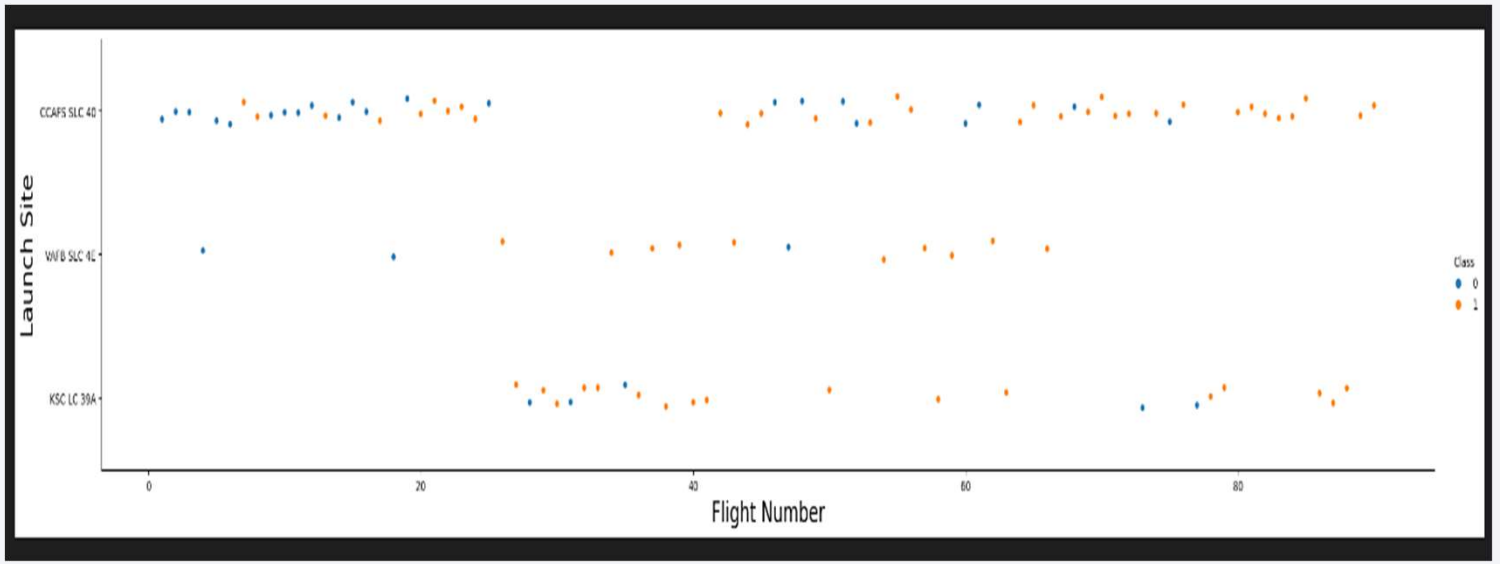


Section 2

Insights drawn from EDA

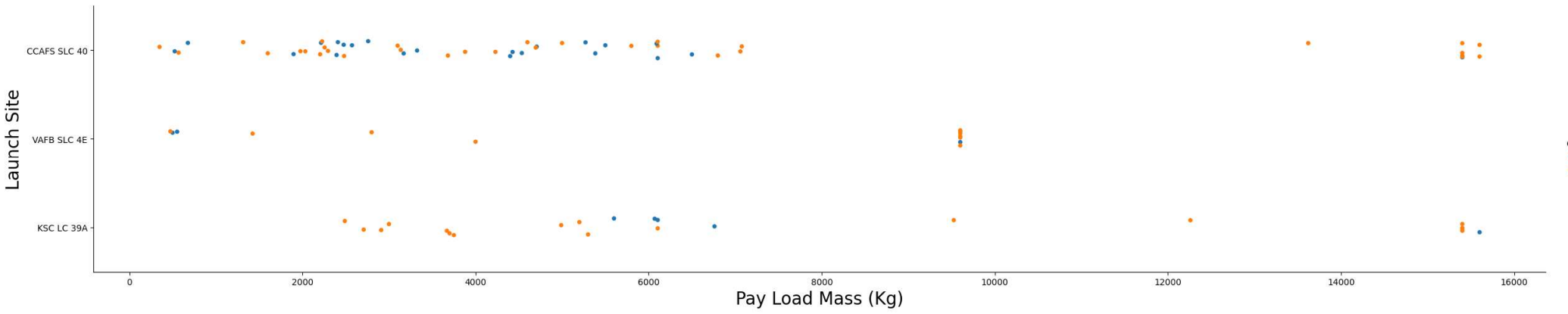
Flight Number vs. Launch Site

- CCSFS SLC-40, Cape Carnival Space Force Station had successful outcomes when the flight number increased.



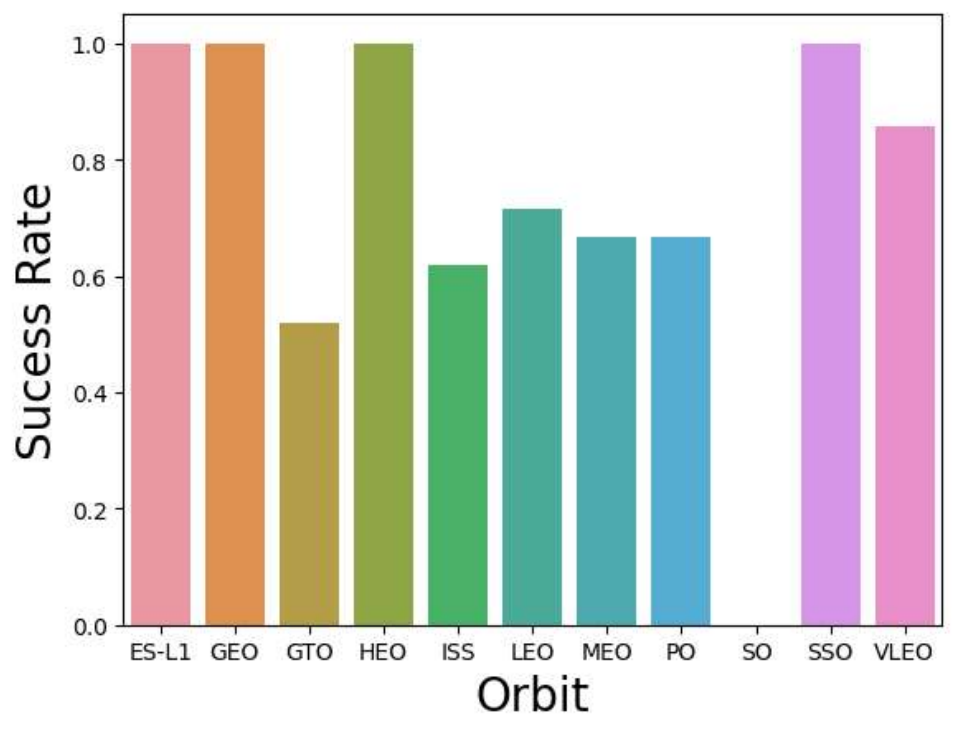
Payload vs. Launch Site

- CCSFS SLC-40, Cape Carnival Space Force Station had successful outcomes at higher payload masses above 5000kg
- VAFB-SLC did not have any launches for heavy payload greater than 10,000 KGS.



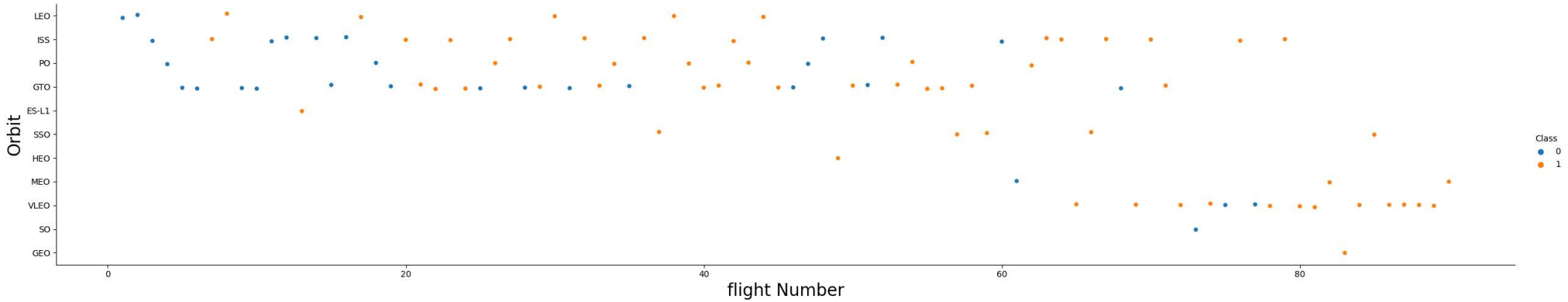
Success Rate vs. Orbit Type

- ES-L1,GEO,HEO, SSO orbits had better (100%) success rate compared to other orbits.



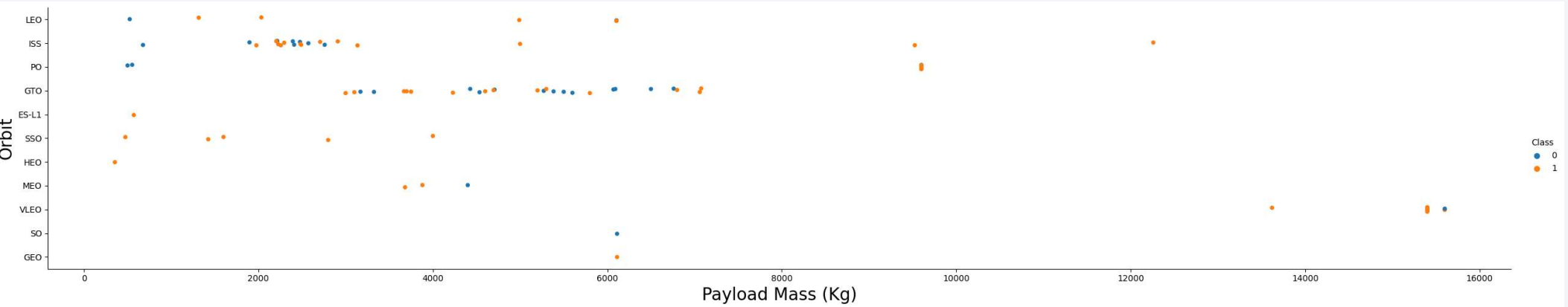
Flight Number vs. Orbit Type

- Higher flight numbers had successful launches in VLEO orbit and ISS while in LEO orbit for midrange flight numbers.



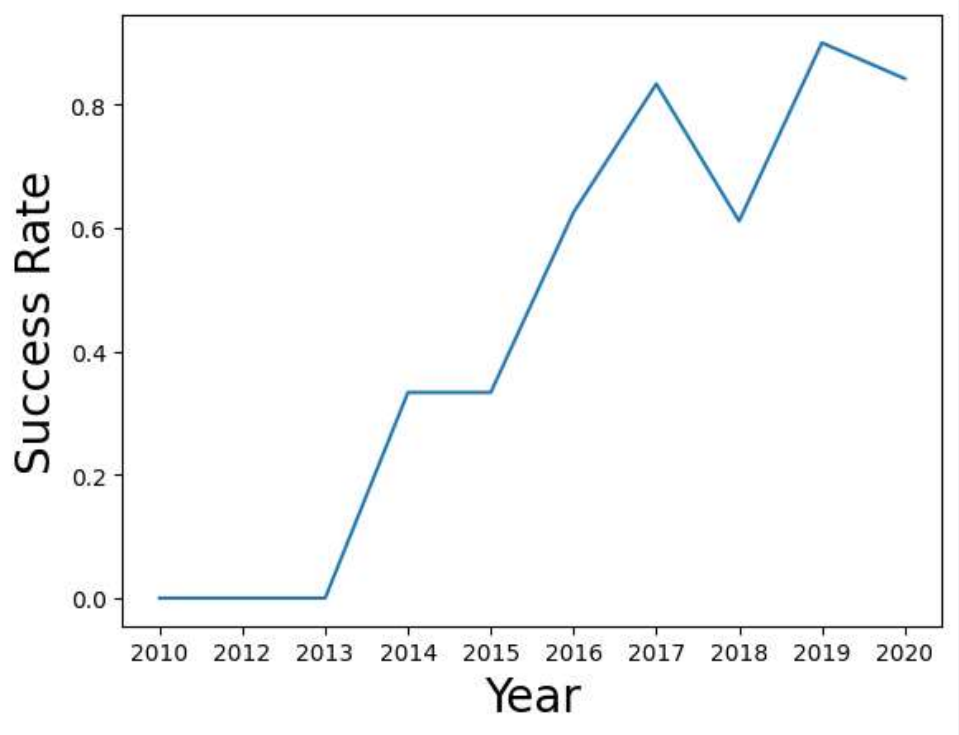
Payload vs. Orbit Type

- Higher payload mass had better success rate for Polar, LEO, ISS and VLEO orbits.



Launch Success Yearly Trend

- Success rate since 2013 kept increasing rapidly till 2020.



All Launch Site Names

```
%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'

```
%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

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

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE CUSTOMER = 'NASA (CRS)'
```

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

SUM(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

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

+ Code + Markdown

```
%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

```
%sql SELECT MIN(DATE) FROM SPACEXTBL WHERE  "Landing _Outcome" LIKE "Success (ground pad)"
```

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

MIN(DATE)
01-05-2017

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%sql SELECT BOOSTER_VERSION, "Landing _Outcome", PAYLOAD_MASS__KG_ FROM SPACEXTBL WHERE \
"Landing _Outcome" LIKE "Success (Drone %)" AND (PAYLOAD_MASS__KG_ > 4000) AND (PAYLOAD_MASS__KG_ < 6000)
```

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

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

```
%sql SELECT MISSION_OUTCOME,COUNT(*) as count FROM SPACEXTBL GROUP BY MISSION_OUTCOME \
ORDER BY count DESC;
```

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

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

Boosters Carried Maximum Payload

```
%sql SELECT BOOSTER_VERSION, COUNT(BOOSTER_VERSION) AS COUNT, "PAYLOAD_MASS__KG_" FROM SPACEXTBL \
GROUP BY "PAYLOAD_MASS__KG_"\
ORDER BY "PAYLOAD_MASS__KG_" DESC LIMIT 10
```

✓ 0.0s

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

Booster_Version	COUNT	PAYLOAD_MASS__KG_
F9 B5 B1048.4	12	15600
F9 B5 B1049.6	1	15440
F9 B5 B1059.3	1	15410
F9 B5 B1051.5	1	14932
F9 B5 B1049.3	1	13620
F9 B5B1058.1	1	12530
F9 B5B1061.1	1	12500

2015 Launch Records

```
%sql SELECT substr('JanFebMarAprMayJunJulAugSepOctNovDec', 1 + 3*SUBSTR(Date,4,2), -3) as Month, \
"Landing _Outcome", BOOSTER_VERSION, LAUNCH_SITE, DATE FROM SPACEXTBL WHERE "Landing _Outcome" LIKE \
"Failure (Drone %)" AND substr(Date,7,4)='2015'
```

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

Month	Landing _Outcome	Booster_Version	Launch_Site	Date
Jan	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40	10-01-2015
Apr	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40	14-04-2015

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

Rank the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.

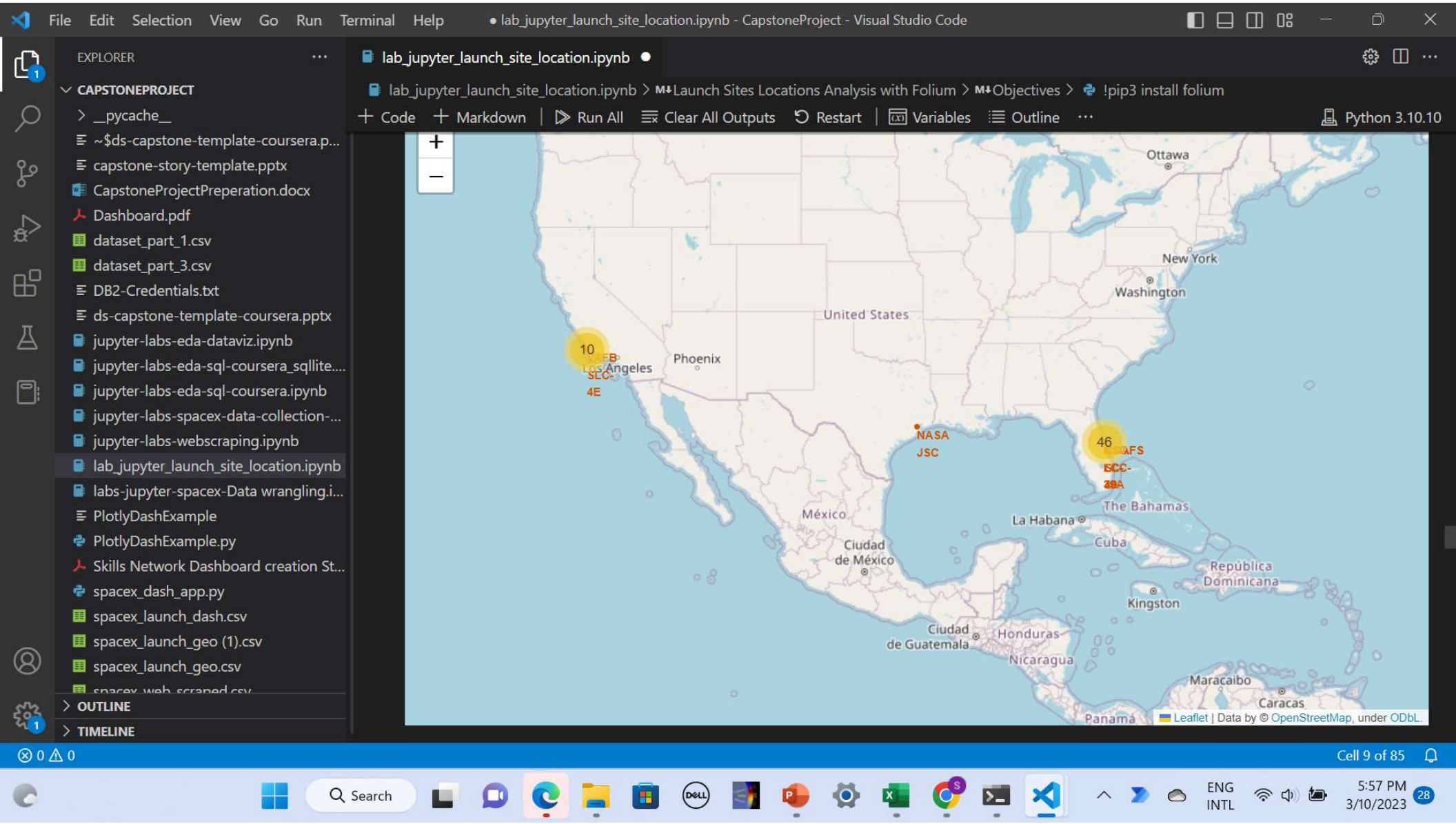
```
%sql SELECT DISTINCT("Landing _Outcome"), COUNT(*) as count FROM \
(SELECT * FROM SPACEXTBL WHERE DATE BETWEEN '04-06-2010' AND '20-03-2017') WHERE \
"Landing _Outcome" LIKE "success%" GROUP BY "Landing _Outcome"
```

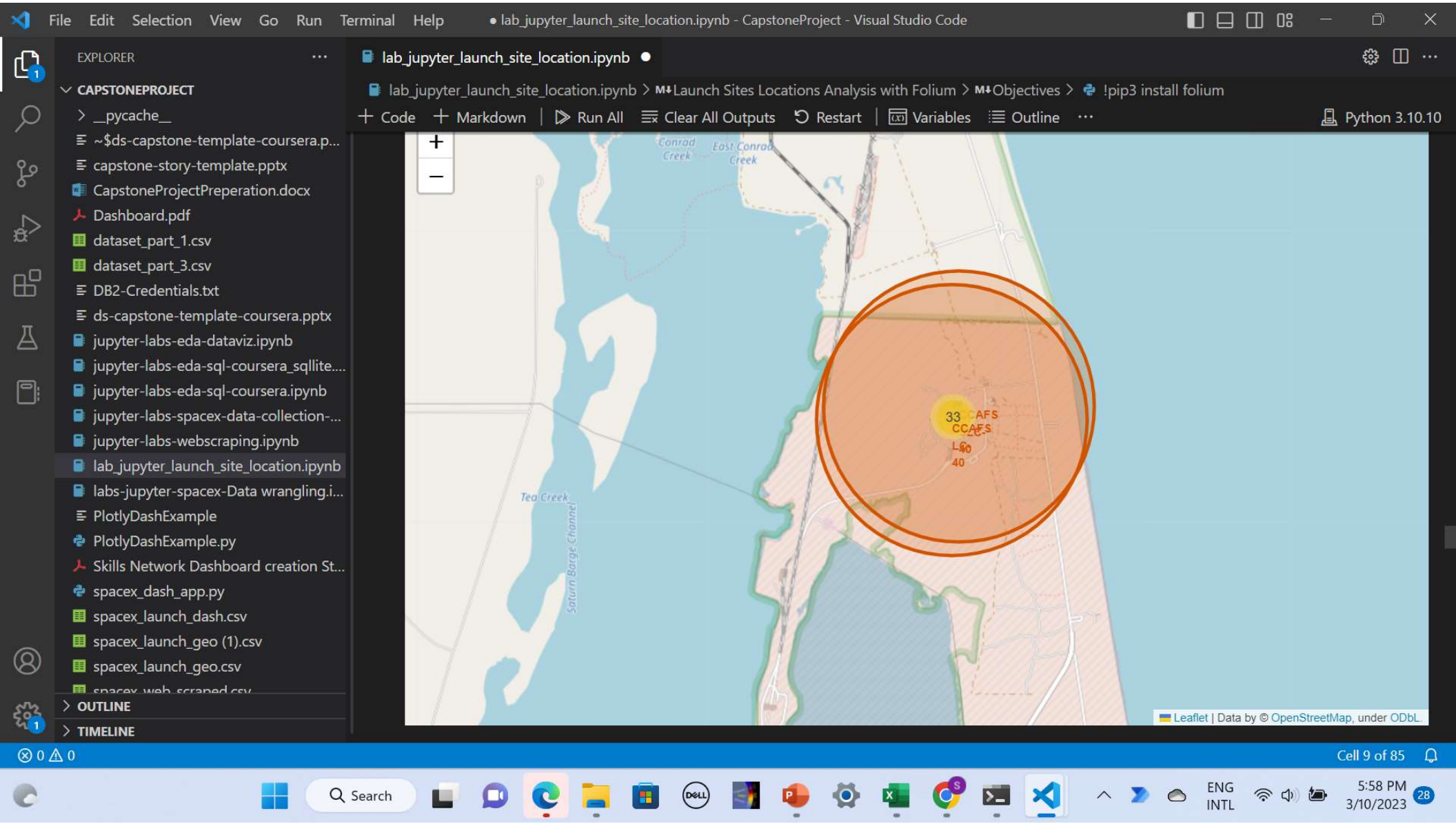
```
 * sqlite:///my_data1.db
Done.
```

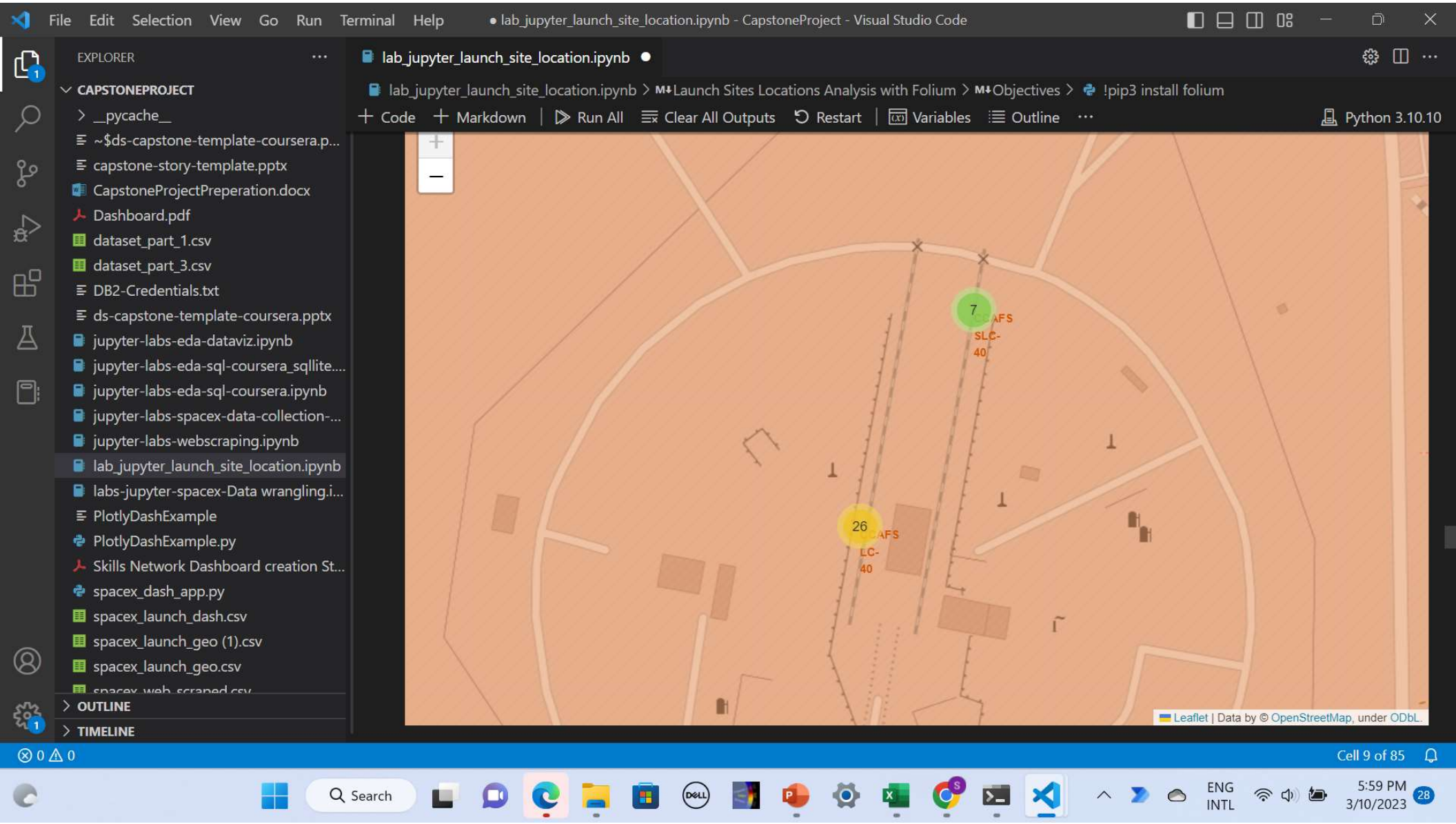
Landing _Outcome	count
Success	20
Success (drone ship)	8
Success (ground pad)	6

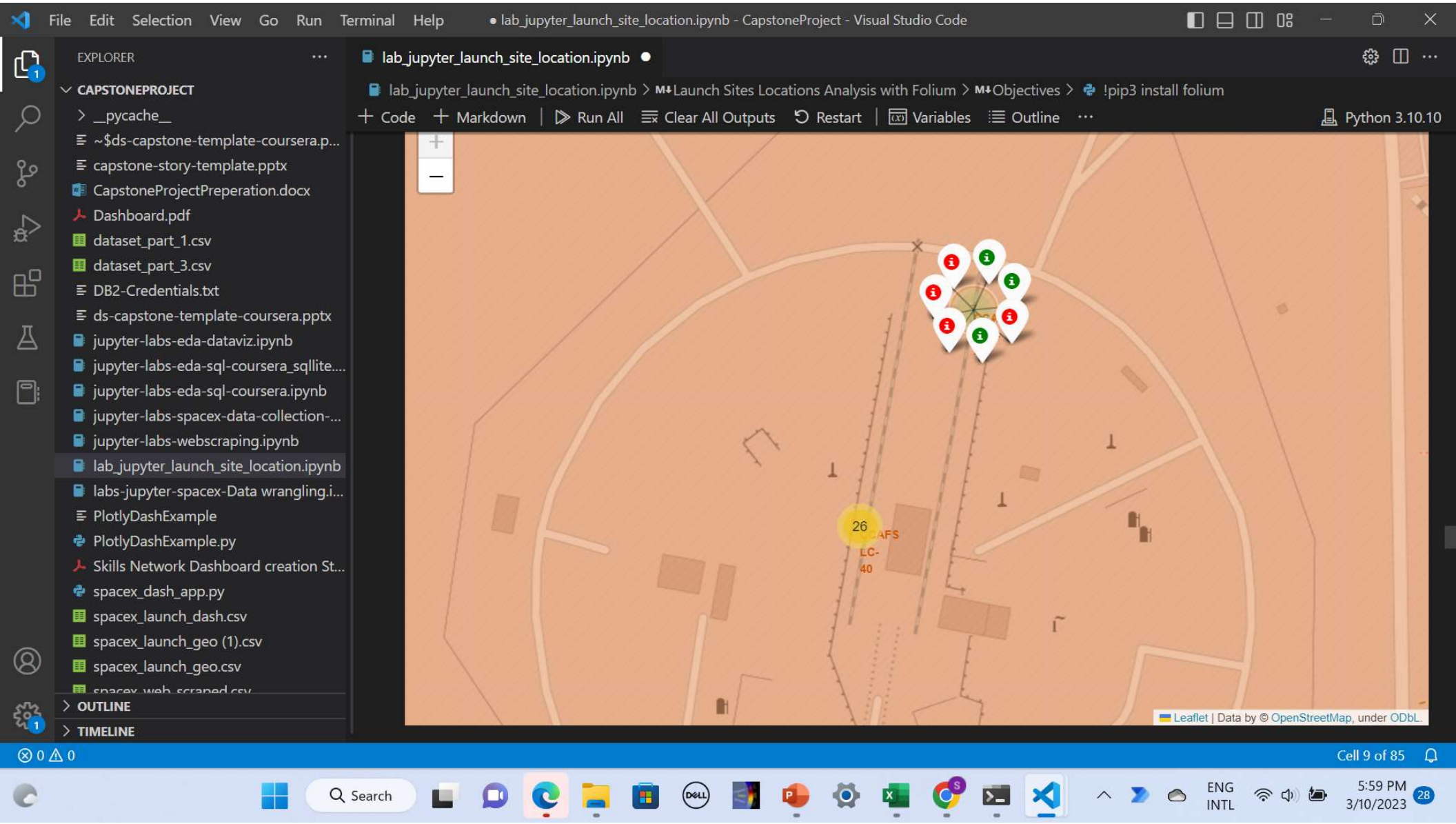
Section 3

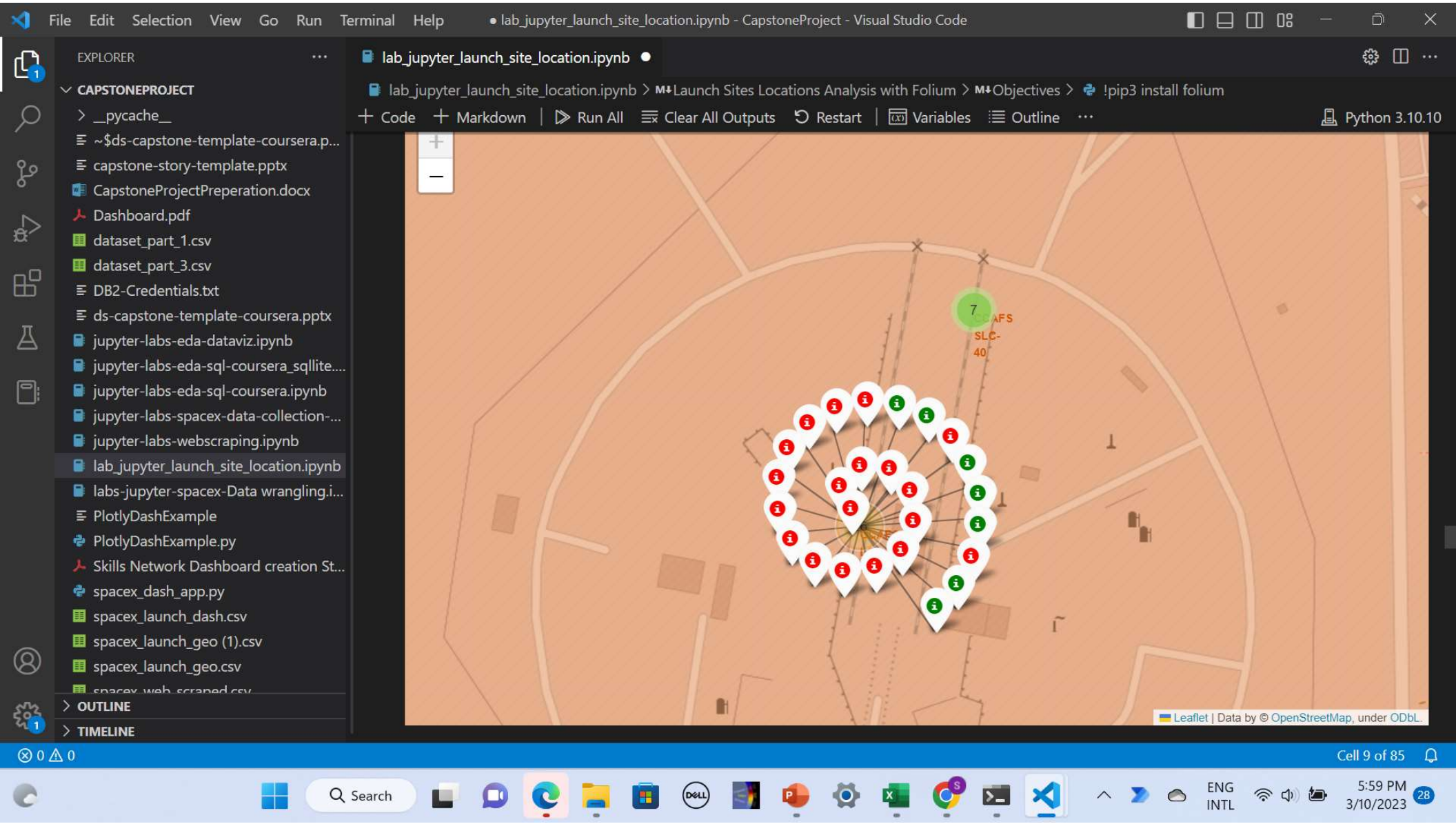
Launch Sites Proximities Analysis

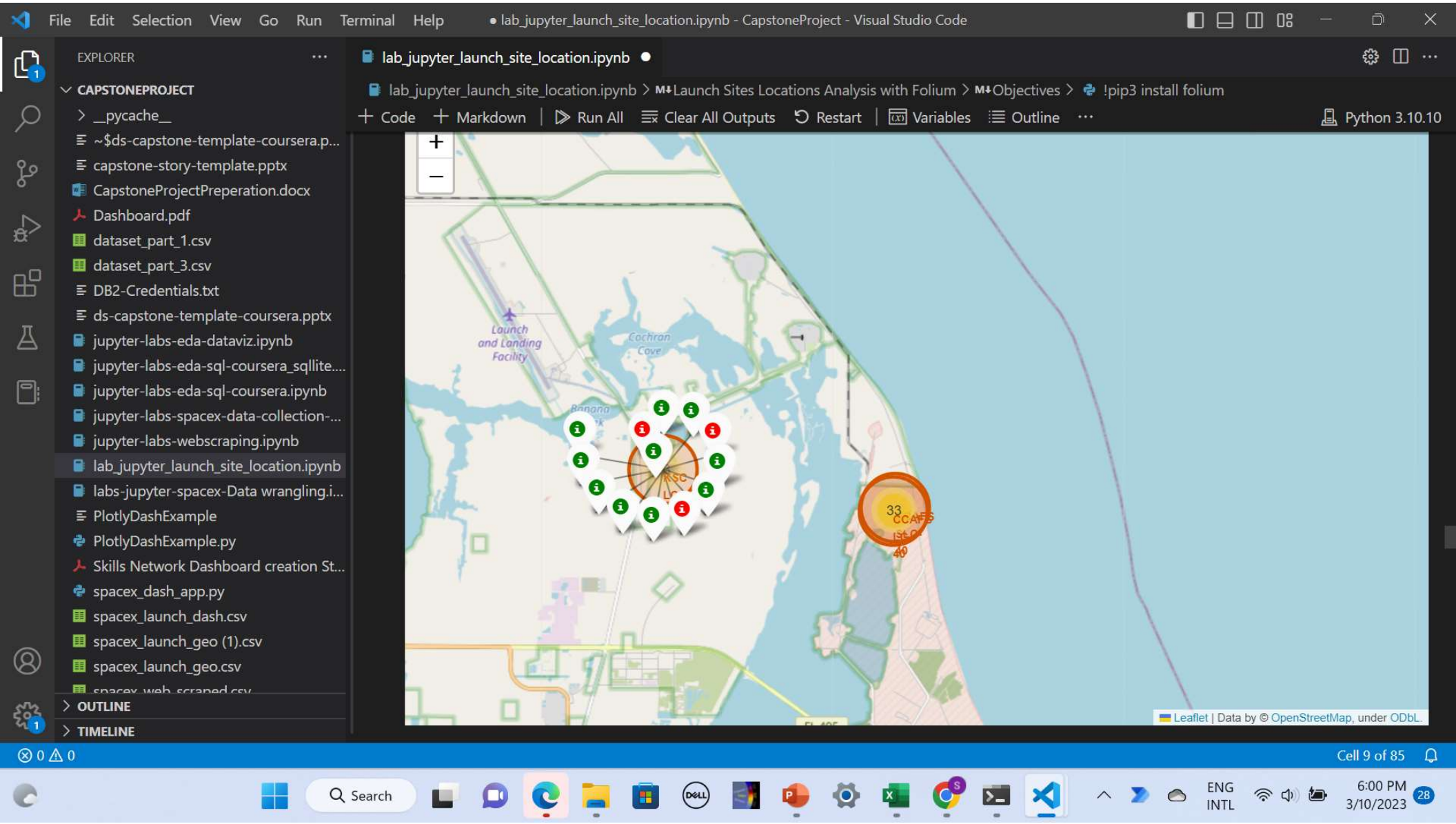

File Edit Selection View Go Run Terminal Help
• lab_jupyter_launch_site_location.ipynb - CapstoneProject - Visual Studio Code
EXPLORER
CAPSTONEPROJECT
pycache
~$ds-capstone-template-coursera.p...
capstone-story-template.pptx
CapstoneProjectPreperation.docx
Dashboard.pdf
dataset_part_1.csv
dataset_part_3.csv
DB2-Credentials.txt
ds-capstone-template-coursera.pptx
jupyter-labs-eda-dataviz.ipynb
jupyter-labs-eda-sql-coursera_sqllite....
jupyter-labs-eda-sql-coursera.ipynb
jupyter-labs-spacex-data-collection-...
jupyter-labs-webscraping.ipynb
lab_jupyter_launch_site_location.ipynb
labs-jupyter-spacex-Data wrangling.i...
PlotlyDashExample
PlotlyDashExample.py
Skills Network Dashboard creation St...
spacex_dash_app.py
spacex_launch_dash.csv
spacex_launch_geo (1).csv
spacex_launch_geo.csv
OUTLINE
TIMELINE
lab_jupyter_launch_site_location.ipynb
lab_jupyter_launch_site_location.ipynb > Launch Sites Locations Analysis with Folium > Objectives > !pip3 install folium
Code Markdown Run All Clear All Outputs Restart Variables Outline
Python 3.10.10
Ottawa
New York
Washington
United States
Phoenix
Los Angeles
10
FB
SLC-4E
NASA JSC
46
AFS
SCC-
The Bahamas
La Habana
Cuba
México
Ciudad de México
República Dominicana
Kingston
Ciudad de Guatemala
Honduras
Nicaragua
Maracaibo
Caracas
Panamá
Leaflet | Data by © OpenStreetMap, under ODbL.
0 0
Cell 9 of 85
Search
ENG INTL
5:57 PM 3/10/2023
28

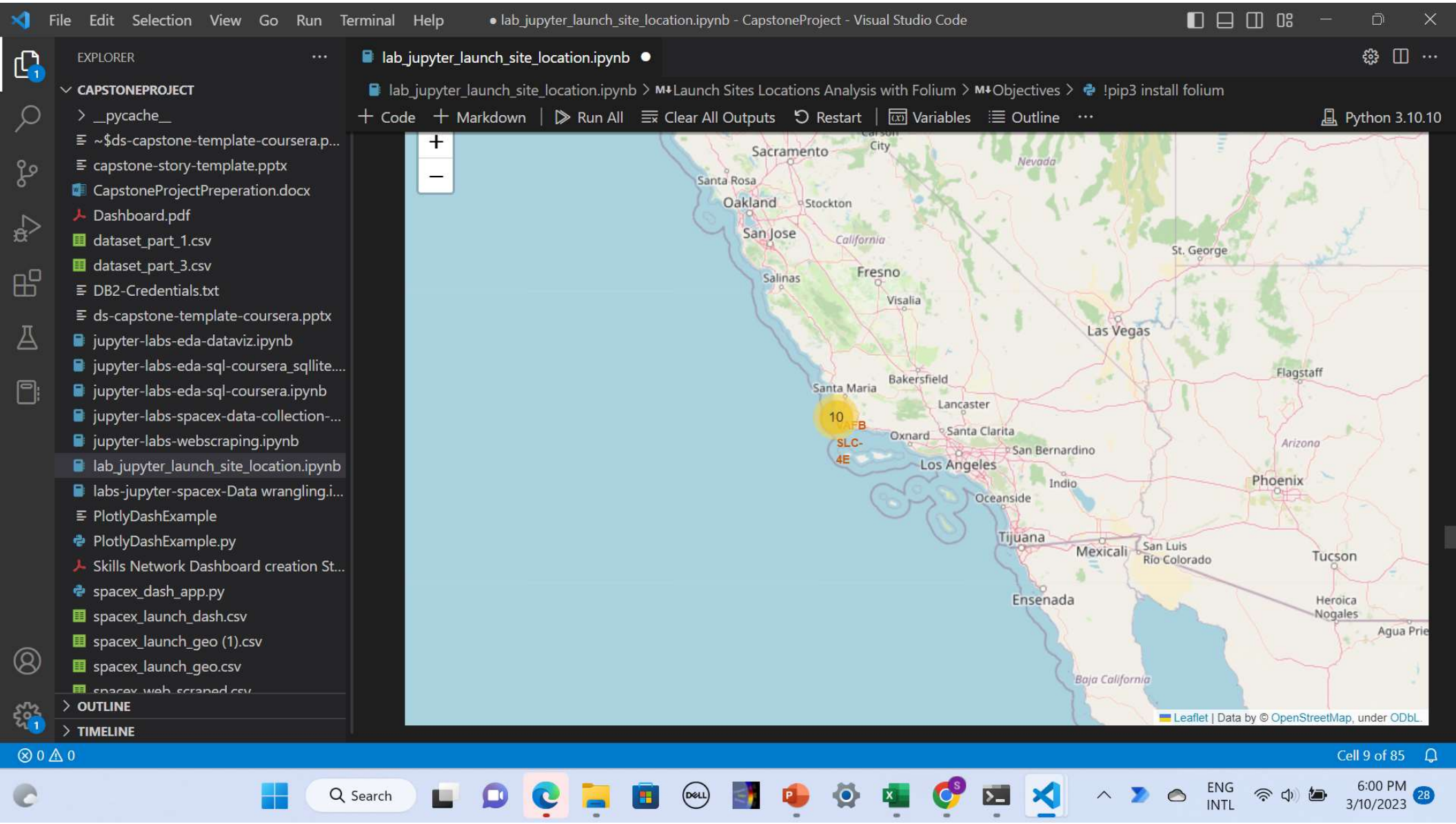

File Edit Selection View Go Run Terminal Help
• lab_jupyter_launch_site_location.ipynb - CapstoneProject - Visual Studio Code
EXPLORER
CAPSTONEPROJECT
pycache
~$ds-capstone-template-coursera.p...
capstone-story-template.pptx
CapstoneProjectPreperation.docx
Dashboard.pdf
dataset_part_1.csv
dataset_part_3.csv
DB2-Credentials.txt
ds-capstone-template-coursera.pptx
jupyter-labs-eda-dataviz.ipynb
jupyter-labs-eda-sql-coursera_sqllite....
jupyter-labs-eda-sql-coursera.ipynb
jupyter-labs-spacex-data-collection-...
jupyter-labs-webscraping.ipynb
lab_jupyter_launch_site_location.ipynb
labs-jupyter-spacex-Data wrangling.i...
PlotlyDashExample
PlotlyDashExample.py
Skills Network Dashboard creation St...
spacex_dash_app.py
spacex_launch_dash.csv
spacex_launch_geo (1).csv
spacex_launch_geo.csv
OUTLINE
TIMELINE
lab_jupyter_launch_site_location.ipynb
lab_jupyter_launch_site_location.ipynb > Launch Sites Locations Analysis with Folium > Objectives > !pip3 install folium
Code Markdown Run All Clear All Outputs Restart Variables Outline
Python 3.10.10
Conrad Creek
East Conrad Creek
Teo Creek
Saturn Barge Channel
33
CAFS
CCAFS
Leaflet | Data by © OpenStreetMap, under ODbL.
Cell 9 of 85
Search
ENG
INTL
5:58 PM
3/10/2023

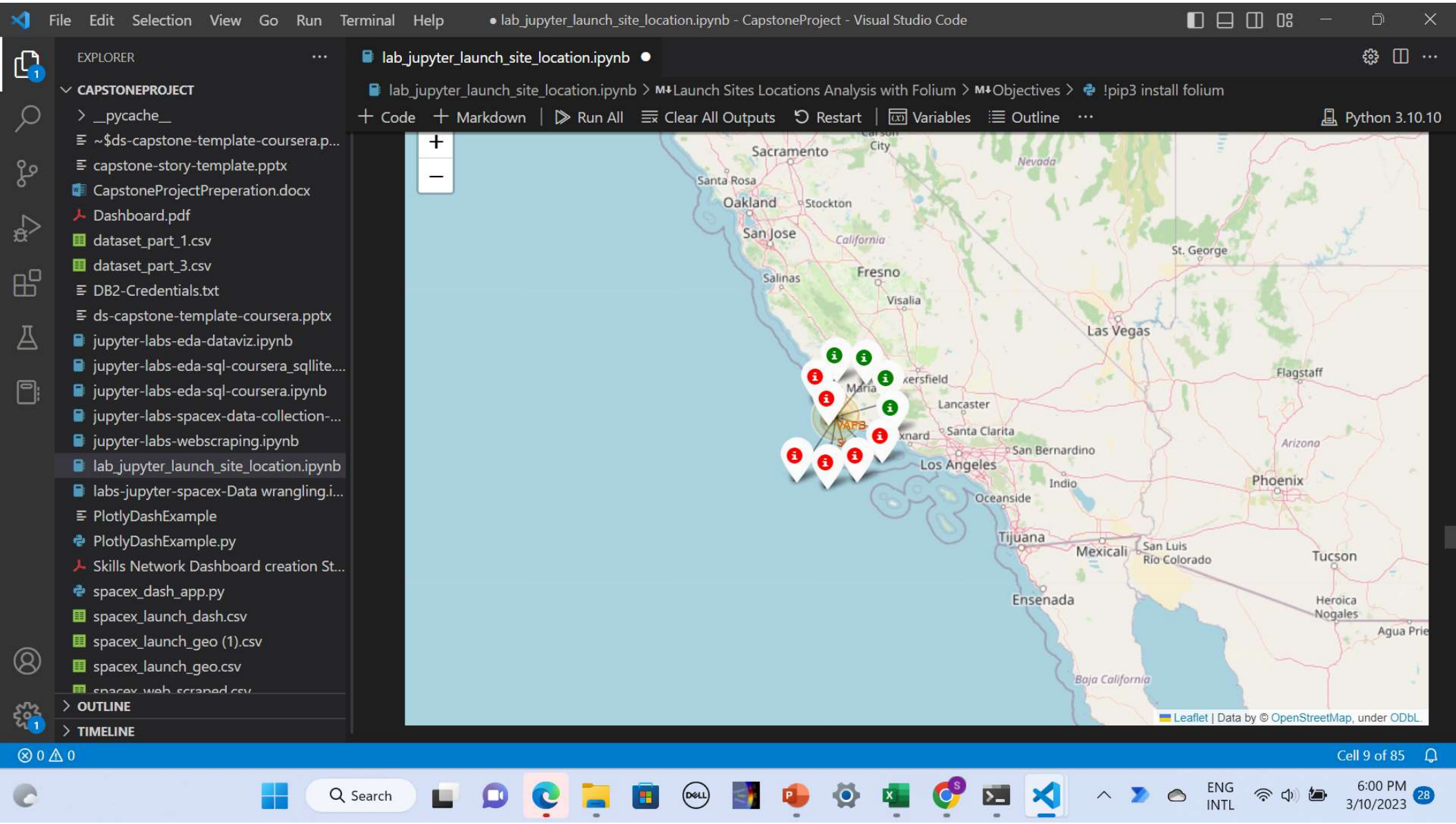

File Edit Selection View Go Run Terminal Help
• lab_jupyter_launch_site_location.ipynb - CapstoneProject - Visual Studio Code
EXPLORER
CAPSTONEPROJECT
pycache
~$ds-capstone-template-coursera.p...
capstone-story-template.pptx
CapstoneProjectPreperation.docx
Dashboard.pdf
dataset_part_1.csv
dataset_part_3.csv
DB2-Credentials.txt
ds-capstone-template-coursera.pptx
jupyter-labs-eda-dataviz.ipynb
jupyter-labs-eda-sql-coursera_sqllite....
jupyter-labs-eda-sql-coursera.ipynb
jupyter-labs-spacex-data-collection-...
jupyter-labs-webscraping.ipynb
lab_jupyter_launch_site_location.ipynb
labs-jupyter-spacex-Data wrangling.i...
PlotlyDashExample
PlotlyDashExample.py
Skills Network Dashboard creation St...
spacex_dash_app.py
spacex_launch_dash.csv
spacex_launch_geo (1).csv
spacex_launch_geo.csv
OUTLINE
TIMELINE
lab_jupyter_launch_site_location.ipynb
lab_jupyter_launch_site_location.ipynb > Launch Sites Locations Analysis with Folium > Objectives > !pip3 install folium
Code Markdown Run All Clear All Outputs Restart Variables Outline
Python 3.10.10
7
AFS
SLC-
40
26
AFS
LC-
40
Leaflet | Data by © OpenStreetMap, under ODbL.
0 0
Cell 9 of 85
Search
ENG
INTL
5:59 PM
3/10/2023
28

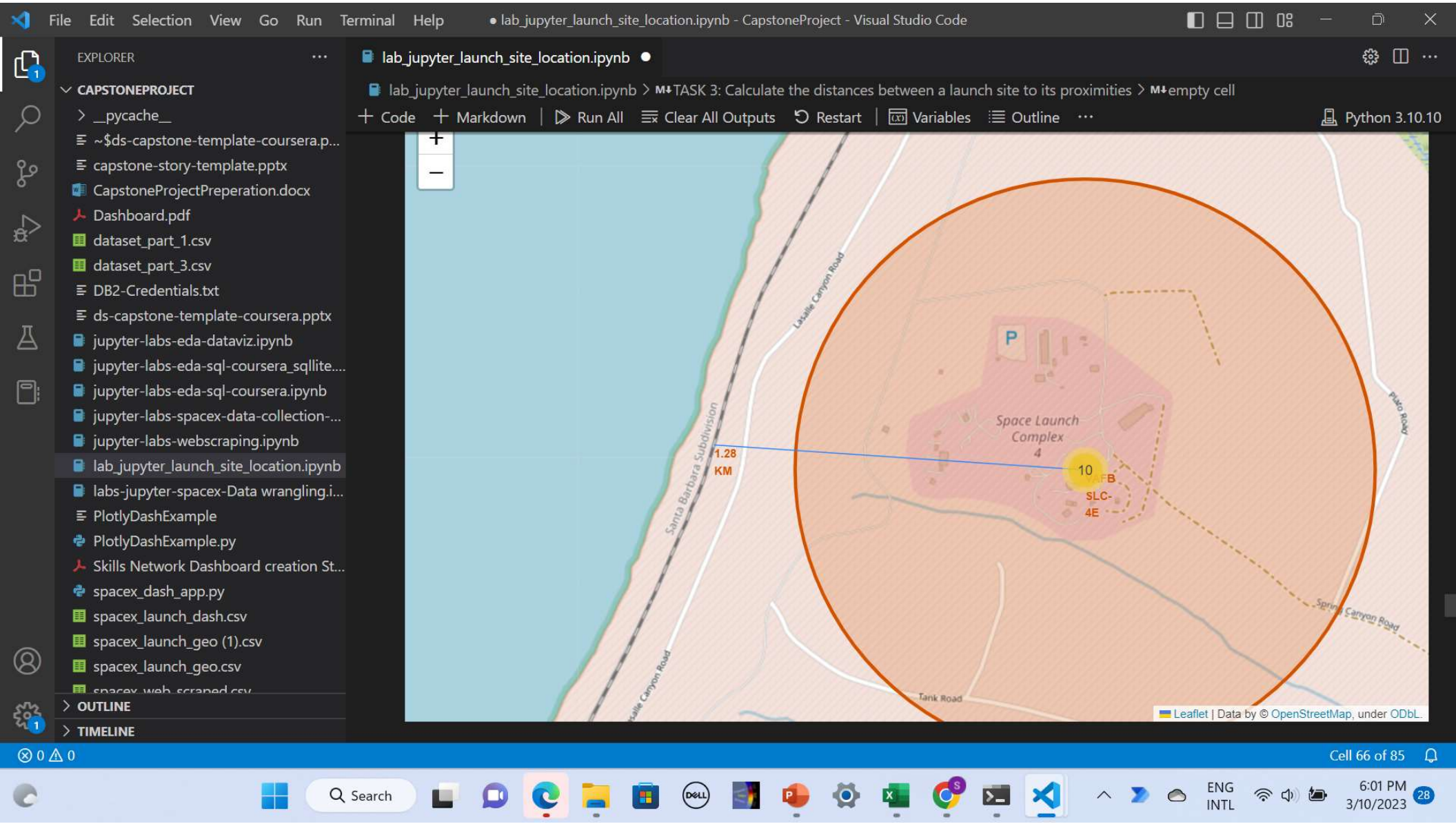

File Edit Selection View Go Run Terminal Help
• lab_jupyter_launch_site_location.ipynb - CapstoneProject - Visual Studio Code
EXPLORER
CAPSTONEPROJECT
pycache
~$ds-capstone-template-coursera.p...
capstone-story-template.pptx
CapstoneProjectPreperation.docx
Dashboard.pdf
dataset_part_1.csv
dataset_part_3.csv
DB2-Credentials.txt
ds-capstone-template-coursera.pptx
jupyter-labs-eda-dataviz.ipynb
jupyter-labs-eda-sql-coursera_sqllite....
jupyter-labs-eda-sql-coursera.ipynb
jupyter-labs-spacex-data-collection-...
jupyter-labs-webscraping.ipynb
lab_jupyter_launch_site_location.ipynb
labs-jupyter-spacex-Data wrangling.i...
PlotlyDashExample
PlotlyDashExample.py
Skills Network Dashboard creation St...
spacex_dash_app.py
spacex_launch_dash.csv
spacex_launch_geo (1).csv
spacex_launch_geo.csv
OUTLINE
TIMELINE
lab_jupyter_launch_site_location.ipynb
lab_jupyter_launch_site_location.ipynb > Launch Sites Locations Analysis with Folium > Objectives > !pip3 install folium
Code Markdown Run All Clear All Outputs Restart Variables Outline
Python 3.10.10
26
AFS
LC-
40
Leaflet | Data by © OpenStreetMap, under ODbL.
Cell 9 of 85
Search
ENG
INTL
5:59 PM
3/10/2023

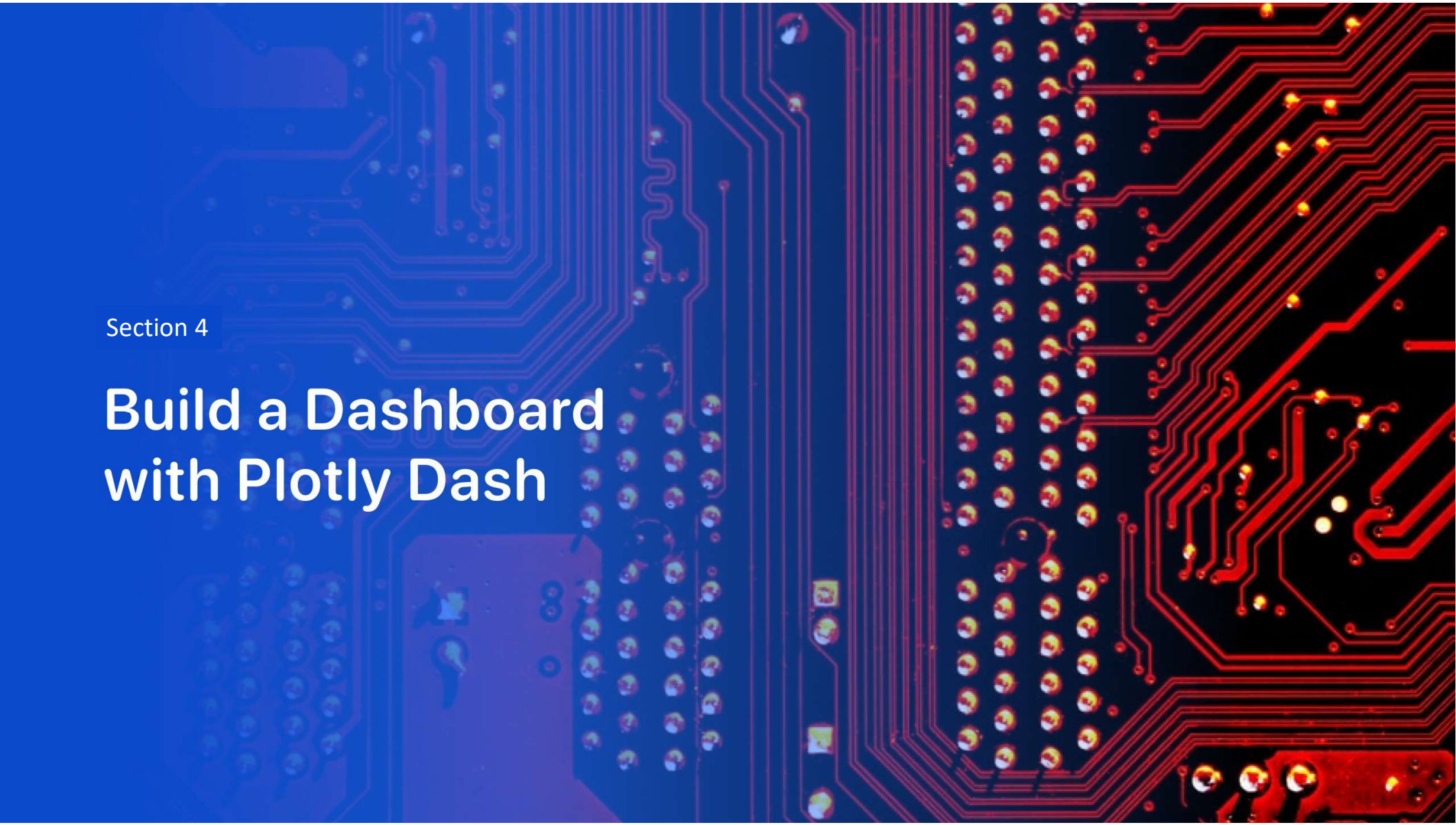

File Edit Selection View Go Run Terminal Help
• lab_jupyter_launch_site_location.ipynb - CapstoneProject - Visual Studio Code
EXPLORER
CAPSTONEPROJECT
pycache
~$ds-capstone-template-coursera.p...
capstone-story-template.pptx
CapstoneProjectPreperation.docx
Dashboard.pdf
dataset_part_1.csv
dataset_part_3.csv
DB2-Credentials.txt
ds-capstone-template-coursera.pptx
jupyter-labs-eda-dataviz.ipynb
jupyter-labs-eda-sql-coursera_sqllite....
jupyter-labs-eda-sql-coursera.ipynb
jupyter-labs-spacex-data-collection-...
jupyter-labs-webscraping.ipynb
lab_jupyter_launch_site_location.ipynb
labs-jupyter-spacex-Data wrangling.i...
PlotlyDashExample
PlotlyDashExample.py
Skills Network Dashboard creation St...
spacex_dash_app.py
spacex_launch_dash.csv
spacex_launch_geo (1).csv
spacex_launch_geo.csv
OUTLINE
TIMELINE
lab_jupyter_launch_site_location.ipynb
lab_jupyter_launch_site_location.ipynb > Launch Sites Locations Analysis with Folium > Objectives > !pip3 install folium
Code Markdown Run All Clear All Outputs Restart Variables Outline
Python 3.10.10
7
SLC-40
Leaflet | Data by © OpenStreetMap, under ODbL.
0 0
Cell 9 of 85
Search
ENG INTL
5:59 PM 3/10/2023


File Edit Selection View Go Run Terminal Help
• lab_jupyter_launch_site_location.ipynb - CapstoneProject - Visual Studio Code
EXPLORER
CAPSTONEPROJECT
pycache
~$ds-capstone-template-coursera.p...
capstone-story-template.pptx
CapstoneProjectPreperation.docx
Dashboard.pdf
dataset_part_1.csv
dataset_part_3.csv
DB2-Credentials.txt
ds-capstone-template-coursera.pptx
jupyter-labs-eda-dataviz.ipynb
jupyter-labs-eda-sql-coursera_sqllite....
jupyter-labs-eda-sql-coursera.ipynb
jupyter-labs-spacex-data-collection-...
jupyter-labs-webscraping.ipynb
lab_jupyter_launch_site_location.ipynb
labs-jupyter-spacex-Data wrangling.i...
PlotlyDashExample
PlotlyDashExample.py
Skills Network Dashboard creation St...
spacex_dash_app.py
spacex_launch_dash.csv
spacex_launch_geo (1).csv
spacex_launch_geo.csv
OUTLINE
TIMELINE
lab_jupyter_launch_site_location.ipynb
lab_jupyter_launch_site_location.ipynb > Launch Sites Locations Analysis with Folium > Objectives > !pip3 install folium
Code Markdown Run All Clear All Outputs Restart Variables Outline
Python 3.10.10
Launch and Landing Facility
Cochran Cove
Banana
KSC
33
CCAFS
SLC
40
Leaflet | Data by © OpenStreetMap, under ODbL.
0 0
Cell 9 of 85
Search
ENG INTL
6:00 PM 3/10/2023
28


File Edit Selection View Go Run Terminal Help
• lab_jupyter_launch_site_location.ipynb - CapstoneProject - Visual Studio Code
EXPLORER
CAPSTONEPROJECT
pycache
~$ds-capstone-template-coursera.p...
capstone-story-template.pptx
CapstoneProjectPreperation.docx
Dashboard.pdf
dataset_part_1.csv
dataset_part_3.csv
DB2-Credentials.txt
ds-capstone-template-coursera.pptx
jupyter-labs-eda-dataviz.ipynb
jupyter-labs-eda-sql-coursera_sqllite....
jupyter-labs-eda-sql-coursera.ipynb
jupyter-labs-spacex-data-collection-...
jupyter-labs-webscraping.ipynb
lab_jupyter_launch_site_location.ipynb
labs-jupyter-spacex-Data wrangling.i...
PlotlyDashExample
PlotlyDashExample.py
Skills Network Dashboard creation St...
spacex_dash_app.py
spacex_launch_dash.csv
spacex_launch_geo (1).csv
spacex_launch_geo.csv
OUTLINE
TIMELINE
lab_jupyter_launch_site_location.ipynb
lab_jupyter_launch_site_location.ipynb > Launch Sites Locations Analysis with Folium > Objectives > !pip3 install folium
Code Markdown Run All Clear All Outputs Restart Variables Outline
Python 3.10.10
Sacramento
Santa Rosa
Oakland
Stockton
San Jose
California
Salinas
Fresno
Visalia
Nevada
St. George
Las Vegas
Flagstaff
Bakersfield
Santa Maria
Lancaster
10
SLC-4E
Oxnard
Santa Clarita
San Bernardino
Los Angeles
Arizona
Phoenix
Indio
Oceanside
Tijuana
Mexicali
San Luis Río Colorado
Tucson
Ensenada
Heroica Nogales
Baja California
Leaflet | Data by © OpenStreetMap, under ODbL.
0 0
Cell 9 of 85
Search
ENG INTL
6:00 PM 3/10/2023


File Edit Selection View Go Run Terminal Help
• lab_jupyter_launch_site_location.ipynb - CapstoneProject - Visual Studio Code
EXPLORER
CAPSTONEPROJECT
pycache
~$ds-capstone-template-coursera.p...
capstone-story-template.pptx
CapstoneProjectPreperation.docx
Dashboard.pdf
dataset_part_1.csv
dataset_part_3.csv
DB2-Credentials.txt
ds-capstone-template-coursera.pptx
jupyter-labs-eda-dataviz.ipynb
jupyter-labs-eda-sql-coursera_sqllite....
jupyter-labs-eda-sql-coursera.ipynb
jupyter-labs-spacex-data-collection-...
jupyter-labs-webscraping.ipynb
lab_jupyter_launch_site_location.ipynb
labs-jupyter-spacex-Data wrangling.i...
PlotlyDashExample
PlotlyDashExample.py
Skills Network Dashboard creation St...
spacex_dash_app.py
spacex_launch_dash.csv
spacex_launch_geo (1).csv
spacex_launch_geo.csv
OUTLINE
TIMELINE
lab_jupyter_launch_site_location.ipynb
lab_jupyter_launch_site_location.ipynb > Launch Sites Locations Analysis with Folium > Objectives > !pip3 install folium
Code Markdown Run All Clear All Outputs Restart Variables Outline
Python 3.10.10
Sacramento
Santa Rosa
Oakland
Stockton
San Jose
California
Salinas
Fresno
Visalia
Nevada
St. George
Las Vegas
Flagstaff
Lancaster
Santa Clarita
San Bernardino
Los Angeles
Indio
Oceanside
Phoenix
Arizona
Tijuana
Mexicali
San Luis Río Colorado
Tucson
Ensenada
Heroica Nogales
Baja California
Leaflet | Data by © OpenStreetMap, under ODbL.
0 0
Cell 9 of 85
Search
ENG INTL
6:00 PM 3/10/2023
28

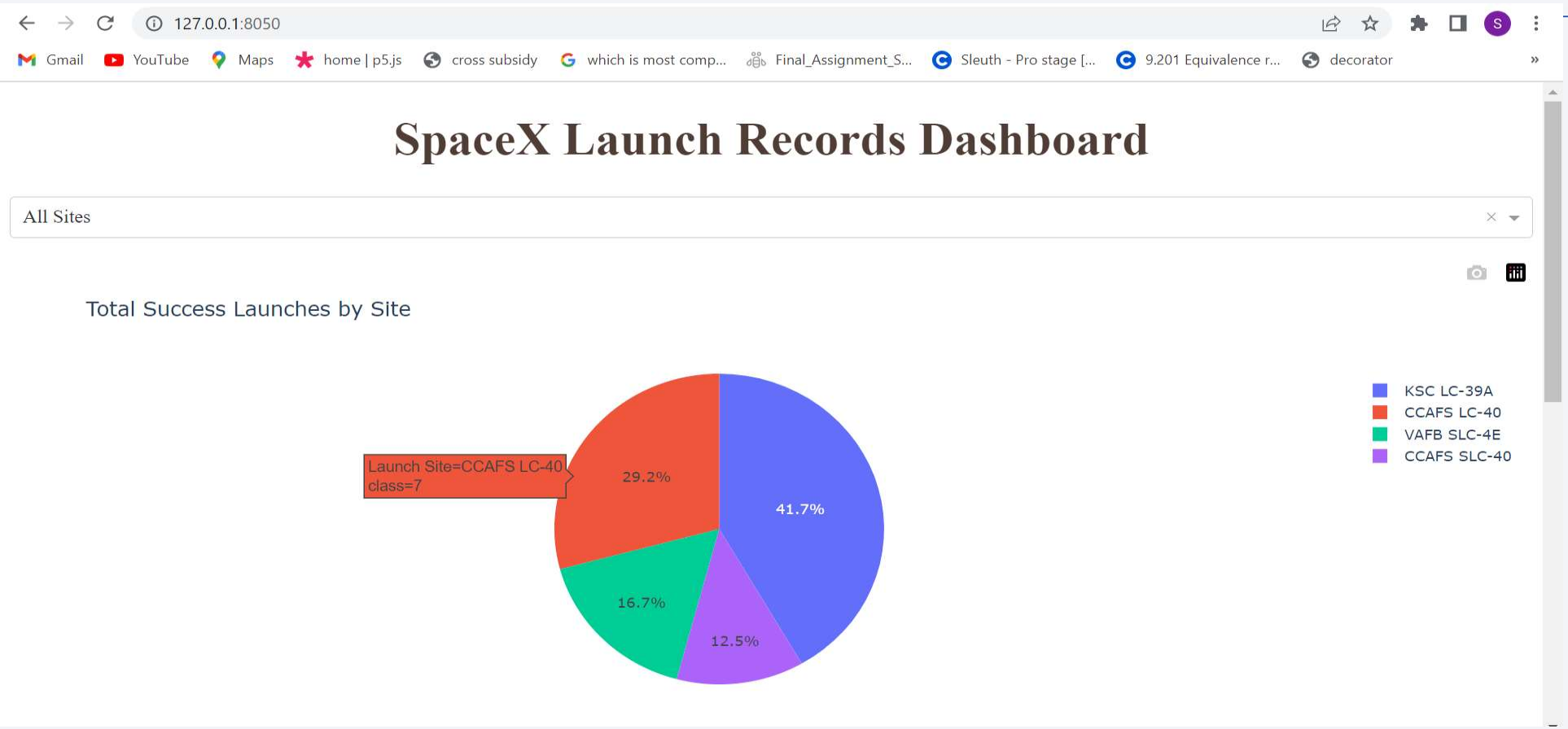

File Edit Selection View Go Run Terminal Help
• lab_jupyter_launch_site_location.ipynb - CapstoneProject - Visual Studio Code
EXPLORER
CAPSTONEPROJECT
pycache
~$ds-capstone-template-coursera.p...
capstone-story-template.pptx
CapstoneProjectPreperation.docx
Dashboard.pdf
dataset_part_1.csv
dataset_part_3.csv
DB2-Credentials.txt
ds-capstone-template-coursera.pptx
jupyter-labs-eda-dataviz.ipynb
jupyter-labs-eda-sql-coursera_sqllite....
jupyter-labs-eda-sql-coursera.ipynb
jupyter-labs-spacex-data-collection-...
jupyter-labs-webscraping.ipynb
lab_jupyter_launch_site_location.ipynb
labs-jupyter-spacex-Data wrangling.i...
PlotlyDashExample
PlotlyDashExample.py
Skills Network Dashboard creation St...
spacex_dash_app.py
spacex_launch_dash.csv
spacex_launch_geo (1).csv
spacex_launch_geo.csv
OUTLINE
TIMELINE
lab_jupyter_launch_site_location.ipynb
lab_jupyter_launch_site_location.ipynb > TASK 3: Calculate the distances between a launch site to its proximities > empty cell
Code Markdown Run All Clear All Outputs Restart Variables Outline
Python 3.10.10
Space Launch Complex 4
1.28 KM
10
VAFB SLC-4E
Lasalle Canyon Road
Santa Barbara Subdivision
Tank Road
Leaflet | Data by © OpenStreetMap, under ODbL.
Cell 66 of 85
Search
ENG INTL
6:01 PM 3/10/2023



Section 4

Build a Dashboard with Plotly Dash

SpaceX Launch Records Dashboard



SpaceX Launch Records Dashboard ….

- Kennedy Space Center Launch Complex 39A (LC-39A)’, ('KSC LC-39A’) has the highest launch activity(41.7%)followed by 'Cape Canaveral Space Launch Complex 40 (LC-40)’, ('CCAFS LC-40’)(29.2%).

Windows 10,8 | Telugu Keyboa | lakshmi sahas | Semaphores in | What is a DDO | Dash | Scatter plots in

127.0.0.1:8050

Gmail | YouTube | Maps | home | p5.js | cross subsidy | which is most comp... | Final_Assignment_S... | Sleuth - Pro stage [... | 9.201 Equivalence r... | decorator

SpaceX Launch Records Dashboard

Cape Canaveral Space Launch Complex 40 (SLC-40)

Total Success Launches at "CCAFS SLC-40"

Search

ENG
INTL

5:48 PM
3/10/2023

Windows 10,8 | Telugu Keyboa | lakshmi sahas | Semaphores in | What is a DDC | Dash | Scatter plots in

127.0.0.1:8050

Gmail | YouTube | Maps | home | p5.js | cross subsidy | which is most comp... | Final_Assignment_S... | Sleuth - Pro stage [... | 9.201 Equivalence r... | decorator

SpaceX Launch Records Dashboard

Kennedy Space Center Launch Complex 39A (LC-39A)

Total Success Launches at "KSC LC-39A"

Search

ENG
INTL

5:48 PM
3/10/2023

Windows 10,8,7 | Telugu Keyboa | lakshmi sahasr | Semaphores in | What is a DDC | Dash | Scatter plots in

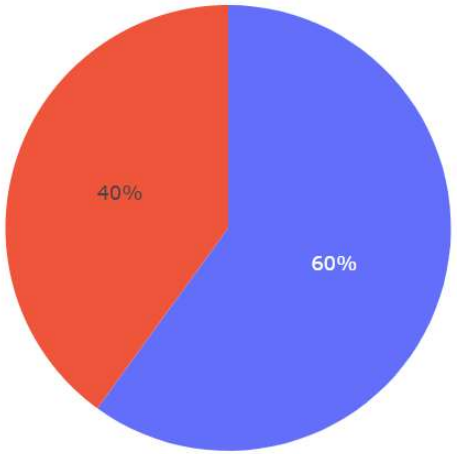
127.0.0.1:8050

Gmail YouTube Maps home | p5.js cross subsidy which is most comp... Final_Assignment_S... Sleuth - Pro stage [... 9.201 Equivalence r... decorator

SpaceX Launch Records Dashboard

Vandenberg Space Force Base Space Launch Complex 4E (SLC-4E)

Total Success Launches at "VAFB SLC-4E"



Search ENG INTL 5:48 PM 3/10/2023 29

Windows 10,8 | Telugu Keyboa | lakshmi sahas | Semaphores in | What is a DDO | Dash | Scatter plots in

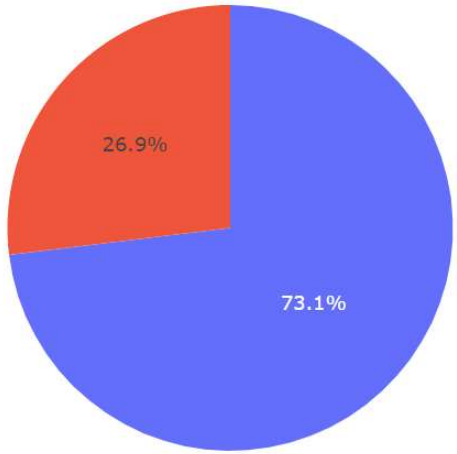
127.0.0.1:8050

Gmail YouTube Maps home | p5.js cross subsidy which is most comp... Final_Assignment_S... Sleuth - Pro stage [... 9.201 Equivalence r... decorator

SpaceX Launch Records Dashboard

Cape Canaveral Space Launch Complex 40 (LC-40)

Total Success Launches at "CCAFS LC-40"



Search ENG INTL 5:49 PM 3/10/2023 29

Windows 10,8 | Telugu Keyboa | lakshmi sahas | Semaphores in | What is a DDO | Dash | Scatter plots in

127.0.0.1:8050

Gmail YouTube Maps home | p5.js cross subsidy which is most comp... Final_Assignment_S... Sleuth - Pro stage [... 9.201 Equivalence r... decorator

Payload range (Kg):

0 2k 4k 6k 8k 10k

Search ENG INTL 5:46 PM 3/10/2023 28

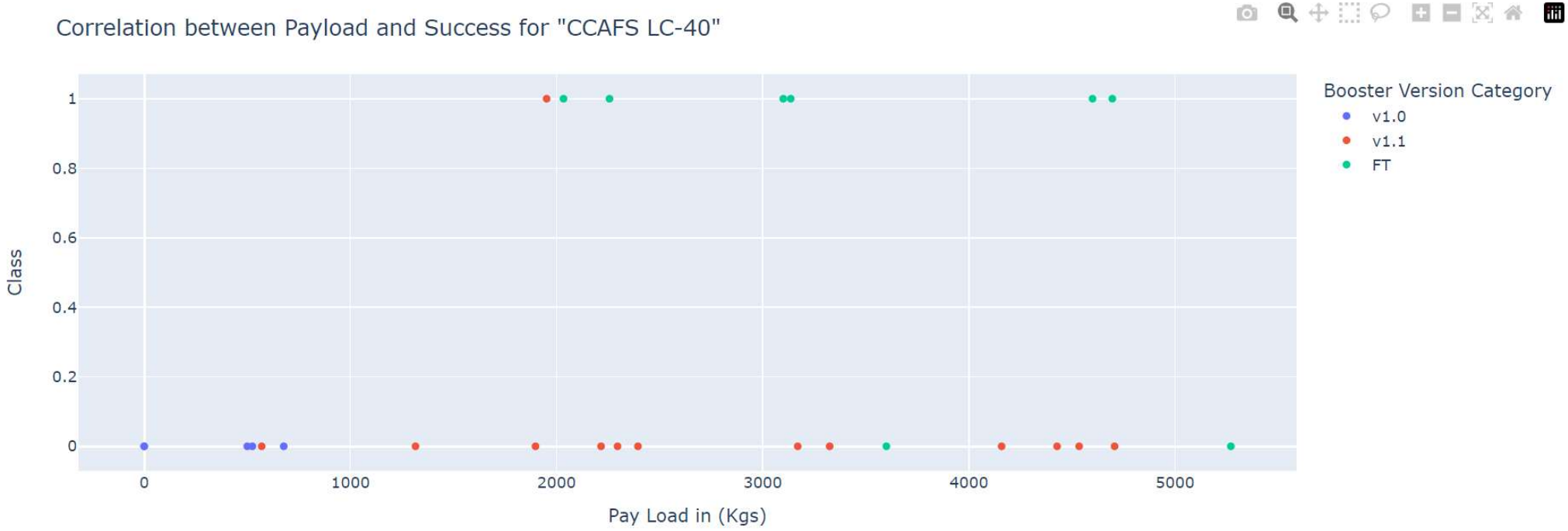
Windows 10,8 | Telugu Keyboa | lakshmi sahas | Semaphores in | What is a DDO | Dash | Scatter plots in

127.0.0.1:8050

Gmail YouTube Maps home | p5.js cross subsidy which is most comp... Final_Assignment_S... Sleuth - Pro stage [... 9.201 Equivalence r... decorator

Payload range (Kg):

0 2k 4k 6k 8k 10k



Search ENG INTL 5:49 PM 3/10/2023 29

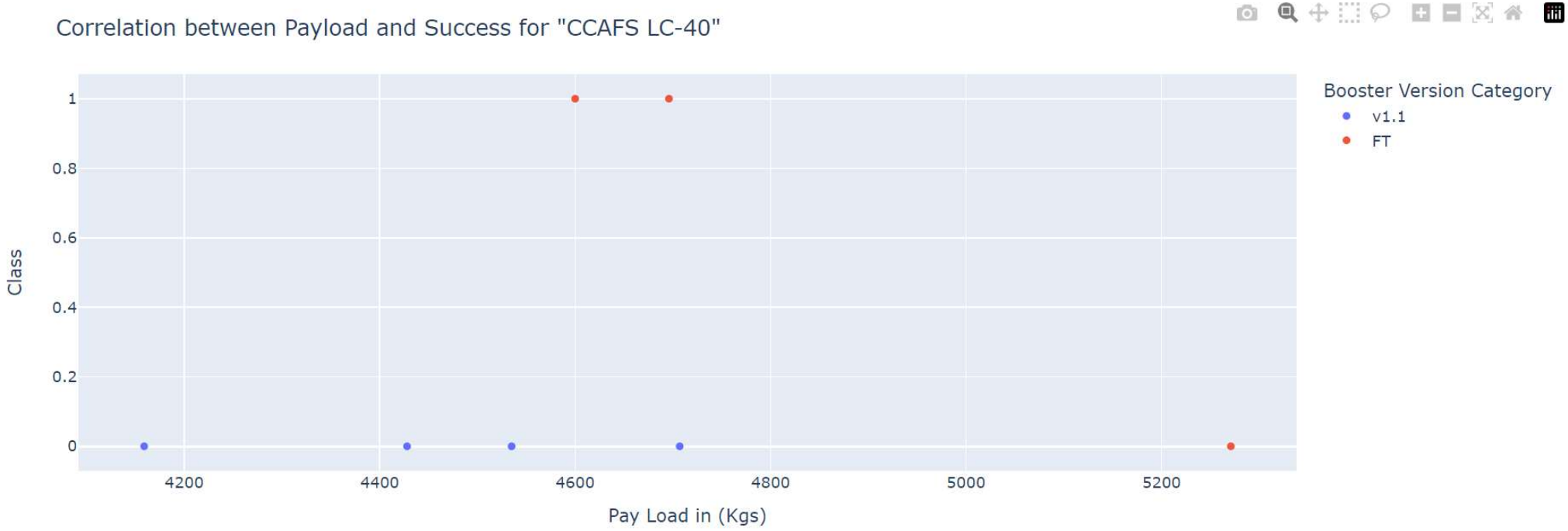
Windows 10,8 | Telugu Keyboa | lakshmi sahas | Semaphores in | What is a DDO | Dash | Scatter plots in

127.0.0.1:8050

Gmail YouTube Maps home | p5.js cross subsidy which is most comp... Final_Assignment_S... Sleuth - Pro stage [... 9.201 Equivalence r... decorator

Payload range (Kg):

0 2k 4k 6k 8k 10k



Search ENG INTL 5:50 PM 3/10/2023 29

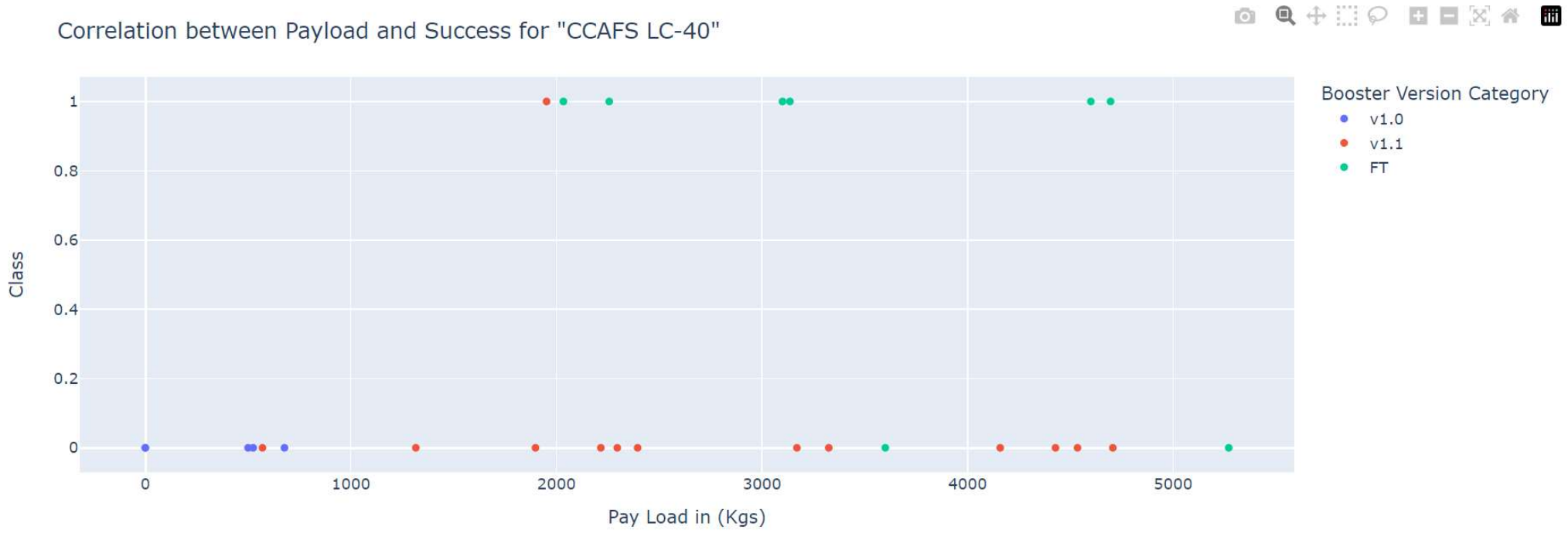
Windows 10,8 | Telugu Keyboa | lakshmi sahas | Semaphores in | What is a DDO | Dash | Scatter plots in

127.0.0.1:8050

Gmail | YouTube | Maps | home | p5.js | cross subsidy | which is most comp... | Final_Assignment_S... | Sleuth - Pro stage [... | 9.201 Equivalence r... | decorator

Payload range (Kg):

0 2k 4k 6k 8k 10k



Search

ENG INTL

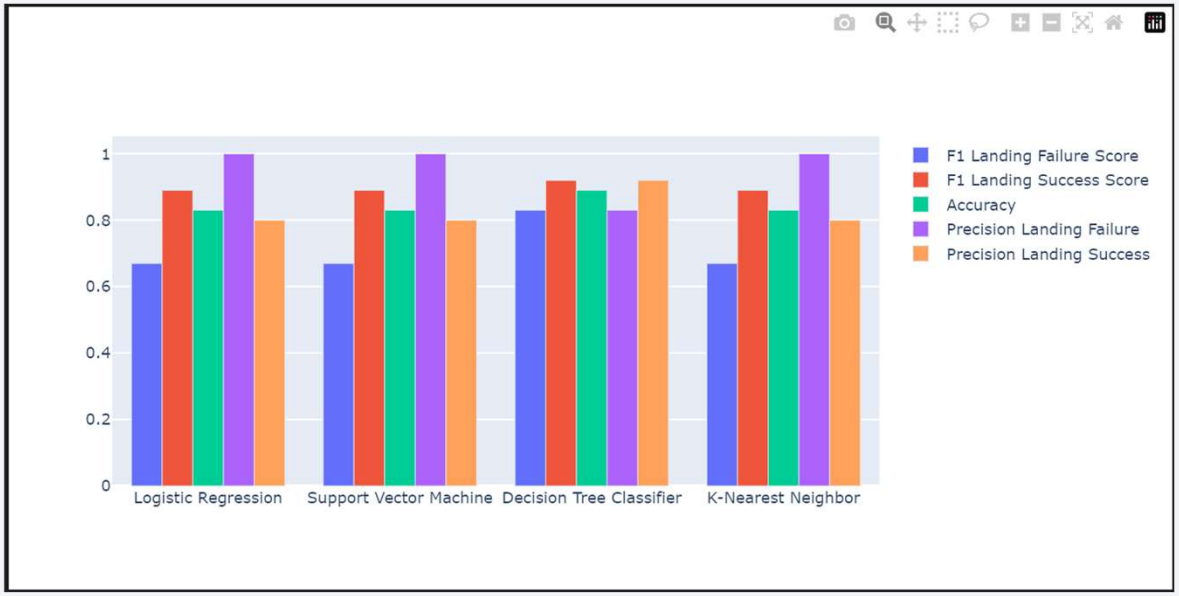
5:52 PM 3/10/2023

Section 5

Predictive Analysis (Classification)

Classification Accuracy

- You can observe that “Decision Tree Classifier” outperforms other three in terms of better F1-Score for both positives and negatives.
- Also notice that all the other three classifiers have almost close and similar performance.



Comparison of Performance of Various Models

Logistic Regression:

	precision	recall	f1-score	support
0	1.00	0.50	0.67	6
1	0.80	1.00	0.89	12
accuracy			0.83	18
macro avg	0.90	0.75	0.78	18
weighted avg	0.87	0.83	0.81	18

Decision Tree Classifier:

	precision	recall	f1-score	support
0	0.83	0.83	0.83	6
1	0.92	0.92	0.92	12
accuracy			0.89	18
macro avg	0.88	0.88	0.88	18
weighted avg	0.89	0.89	0.89	18

Support Vector Machine:

	precision	recall	f1-score	support
0	1.00	0.50	0.67	6
1	0.80	1.00	0.89	12
accuracy			0.83	18
macro avg	0.90	0.75	0.78	18
weighted avg	0.87	0.83	0.81	18

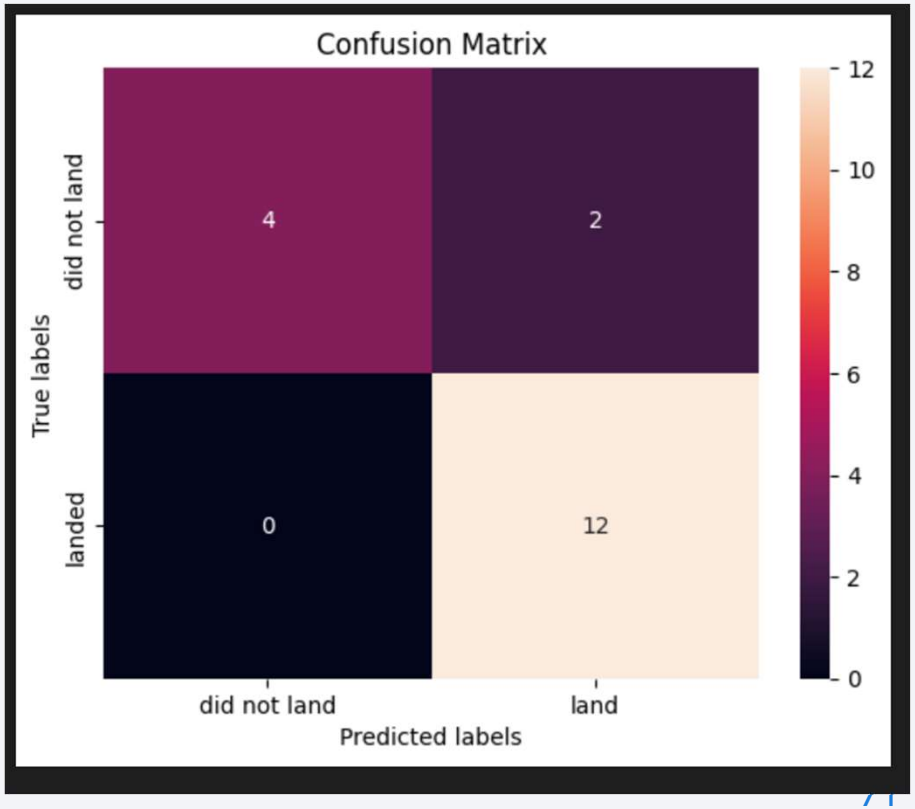
K-Nearest Neighbor:

	precision	recall	f1-score	support
0	1.00	0.50	0.67	6
1	0.80	1.00	0.89	12
accuracy			0.83	18
macro avg	0.90	0.75	0.78	18
weighted avg	0.87	0.83	0.81	18

Confusion Matrix of Best Model

Decision Tree Classifier:

	precision	recall	f1-score	support
0	0.83	0.83	0.83	6
1	0.92	0.92	0.92	12
accuracy			0.89	18
macro avg	0.88	0.88	0.88	18
weighted avg	0.89	0.89	0.89	18



Conclusions

- Decision Tree Classifier stands out as the winner in modeling and predicting the landing outcome of Falcon9 Rocket of SpaceX, which this Data Science Project set out to predict.
- Logistic Regression, Support Vector Machine, K-Nearest Neighbor perform with an accuracy, precision and F1-Score values very closely trailing behind Decision Tree Classifier and all of them have same indistinguishable performance on the available dataset.

Appendix

```
# Comparison of various model performanes using SCIKIT LEARN PYTHON LIBRARY
# ACC = (TP + TN) / (P + N)
# F1 = 2TP / (2TP + FP + FN)
from sklearn.metrics import classification_report
print('Logistic Regression: \n {}'.format(classification_report(Y_test,yhat_logreg)))
print('Support Vector Machine: \n {}'.format(classification_report(Y_test,yhat_svm)))
print('Decision Tree Classifier: \n {}'.format(classification_report(Y_test,yhat_tree)))
print('K-Nearest Neighbor: \n {}'.format(classification_report(Y_test,yhat_knn)))
```

Thank you!