IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Joseoh Okelly
July 22, 2024

Outline

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

Executive Summary

- Summary of methodologies
- Jupyter labs spacex data collection api
- Jupyter labs webscraping Falcon 9
- Labs Jupyter spacex data wrangling
- Jupyter labs eda sql coursera sqlite
- Eda dataviz using pandas and matplotlib
- Spacex launch site locations Analysis
- Build Interactive Dashboard – spacex dashboard
- Spacex Mchine Learning Preductions

Summary of results

- EDA
- Visual Analsis
- Predictive Analysis

Introduction

PERFECTING
PROPULSIVE
LANDING

- Project background and context

  SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage. Therefore if we can determine if the first stage will land, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch.

- Problems you want to find answers

  In this capstone, we will predict if the Falcon 9 first stage will land successfully using data from Falcon 9 rocket launches advertised on its website.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Describe how data was collected
- Perform data wrangling
  - Describe how data was processed
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection

- Describe how data sets were collected.
- Data was collected using Spacex Restful webservices, using Get request Spacecx api.

  The request was later stored for processing using Pandas Dataframe

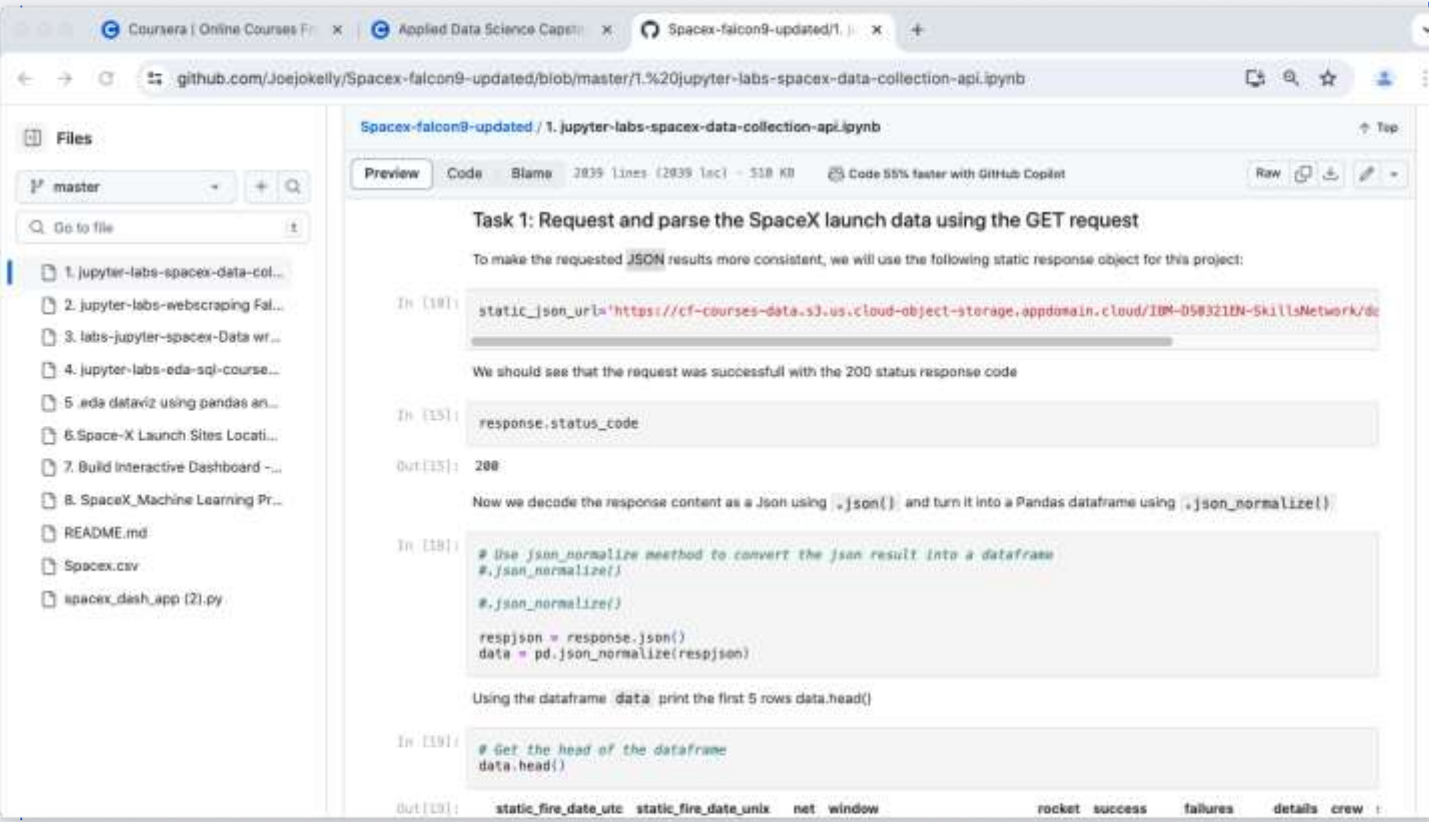
Data was collected using Spacex Restful webservices, using Get request Spacecx api.

- Web scraping performed to Falcon9 historical records.

- You need to present your data collection process use key phrases and flowcharts

Data Collection – SpaceX API

- Data collected using Spacex Api (Restful call using get parameter)
- Get request is decoded by pandas data frame and it is stored for processing.
- https://github.com/Joejokelly/Spacex-falcon9-updated/blob/master/1.%20jupyter-labs-spacex-data-collection-api.ipynb


Coursera | Online Courses Fr
Applied Data Science Capst
Spacex-falcon9-updated/1.
github.com/Joejokelly/Spacex-falcon9-updated/blob/master/1.%20jupyter-labs-spacex-data-collection-api.ipynb
Files
master
Go to file
1. jupyter-labs-spacex-data-col...
2. jupyter-labs-webscraping Fal...
3. labs-jupyter-spacex-Data wr...
4. jupyter-labs-eda-sql-course...
5 .eda dataviz using pandas an...
6.Space-X Launch Sites Locati...
7. Build Interactive Dashboard -...
8. SpaceX_Machine Learning Pr...
README.md
Spacex.csv
spacex_dash_app (2).py
Spacex-falcon9-updated / 1. jupyter-labs-spacex-data-collection-api.ipynb
Top
Preview
Code
Blame
2839 lines (2839 loc) · 518 KB
Code 55% faster with GitHub Copilot
Raw

Task 1: Request and parse the SpaceX launch data using the GET request

To make the requested JSON results more consistent, we will use the following static response object for this project:

```
In [10]: static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/d
```

We should see that the request was successfull with the 200 status response code

```
In [15]: response.status_code

Out[15]: 200
```

Now we decode the response content as a Json using `.json()` and turn it into a Pandas dataframe using `.json_normalize()`

```
In [18]: # Use json_normalize meethod to convert the json result into a dataframe
         #.json_normalize()

         #.json_normalize()

         respjson = response.json()
         data = pd.json_normalize(respjson)
```

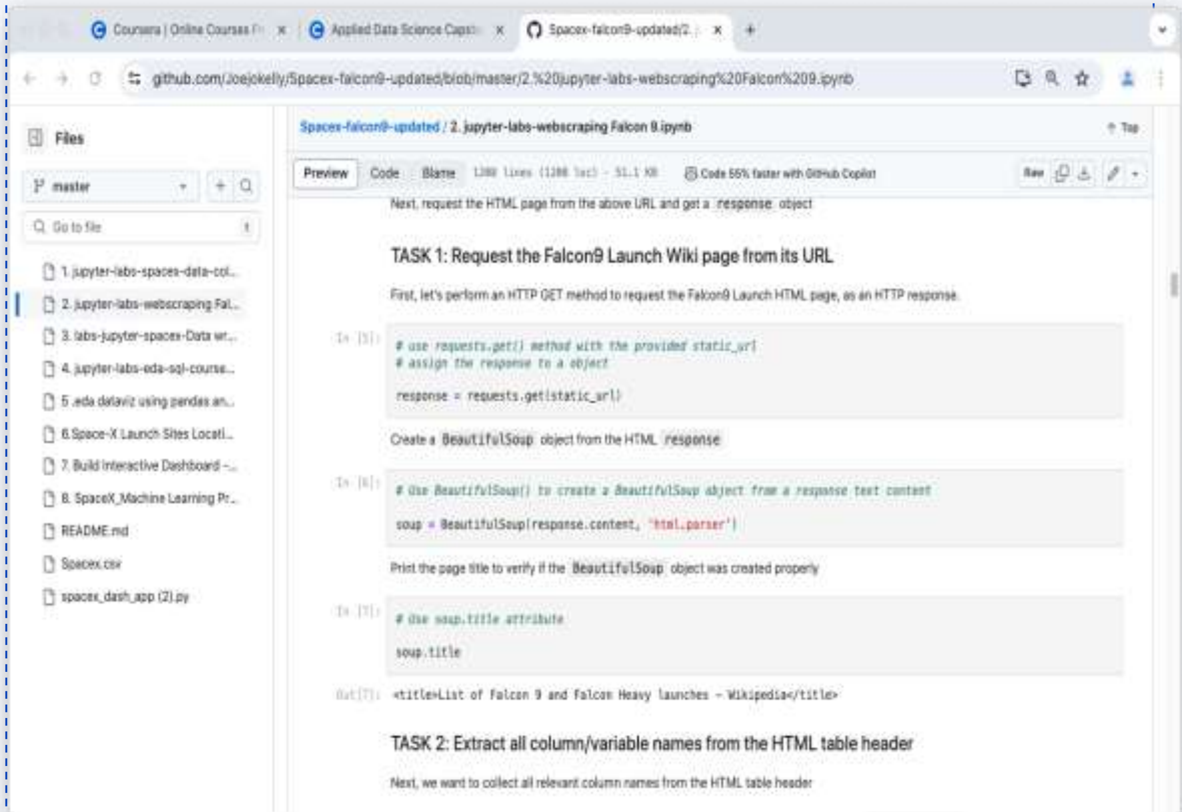
Using the dataframe `data` print the first 5 rows data.head()

```
In [19]: # Get the head of the dataframe
         data.head()

Out[19]:    static_fire_date_utc  static_fire_date_unix   net  window   rocket  success   failures   details  crew
```

Data Collection - Scraping

- Webscraping is performed using historical launch records using Beautiful utilility, next the Data frame is created by passing the Launch HTML file.
- Github url :
- https://github.com/Joejokelly/Spacex-falcon9-updated/blob/master/2.%20jupyter-labs-webscraping%20Falcon%209.ipynb purpose


Coursera | Online Courses Fr
Applied Data Science Capst
Spacex-falcon9-updated/2.
github.com/Joejokelly/Spacex-falcon9-updated/blob/master/2.%20jupyter-labs-webscraping%20Falcon%209.ipynb
Files
master
Go to file
1. jupyter-labs-spacex-data-col...
2. jupyter-labs-webscraping Fal...
3. labs-jupyter-spacex-Data wr...
4. jupyter-labs-eda-sql-course...
5 .eda dataviz using pandas an...
6.Space-X Launch Sites Locati...
7. Build Interactive Dashboard -...
8. SpaceX_Machine Learning Pr...
README.md
Spacex.csv
spacex_dash_app (2).py
Spacex-falcon9-updated / 2. jupyter-labs-webscraping Falcon 9.ipynb
Top
Preview
Code
Blame
Code 55% faster with GitHub Copilot
Raw

Next, request the HTML page from the above URL and get a `response` object

TASK 1: Request the Falcon9 Launch Wiki page from its URL

First, let's perform an HTTP GET method to request the Falcon9 Launch HTML page, as an HTTP response.

```
In [5]: # use requests.get() method with the provided static_url
        # assign the response to a object

        response = requests.get(static_url)
```

Create a `BeautifulSoup` object from the HTML `response`

```
In [6]: # Use BeautifulSoup() to create a BeautifulSoup object from a response text content

        soup = BeautifulSoup(response.content, 'html.parser')
```

Print the page title to verify if the `BeautifulSoup` object was created properly

```
In [7]: # Use soup.title attribute

        soup.title

Out[7]: <title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>
```

TASK 2: Extract all column/variable names from the HTML table header

Next, we want to collect all relevant column names from the HTML table header

Data Wrangling

- Describe how data were processed
- After creation of Pandas Data frame

Hhh

- You need to present your data wrangling process using key phrases and flowcharts

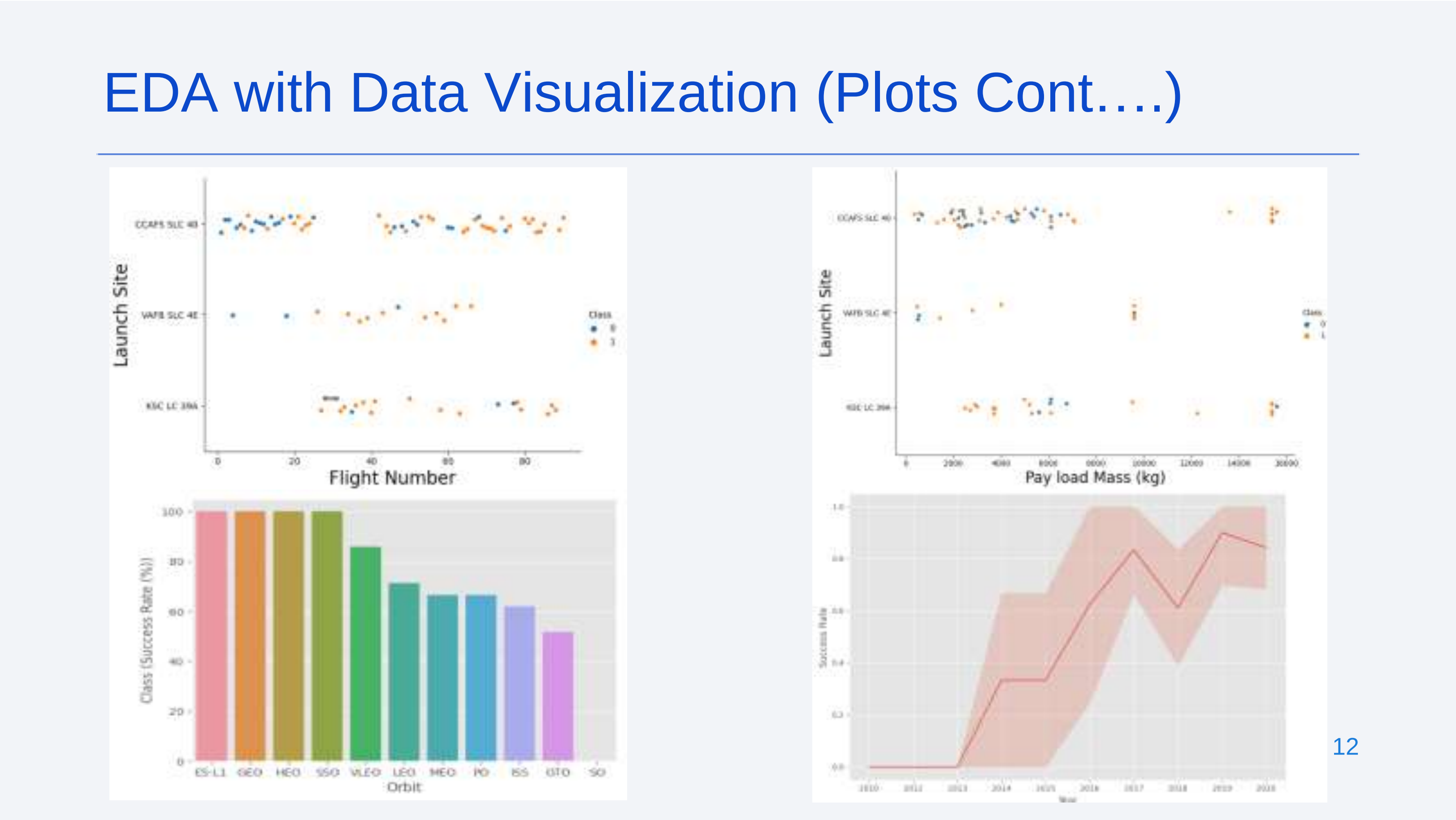
- https://github.com/Joejokelly/Spacex-falcon9-updated/blob/master/3.%20labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

- Data Analysis and Feature Engineering using Pandas and Matplotlib
    - Exploratory Data Analysis
    - Preparing Data Feature Engineering

- https://github.com/Joejokelly/Spacex-falcon9-updated/blob/master/5%20.eda%20dataviz%20using%20pandas%20and%20matplotlib%20.ipynb

EDA with sql (contd)..



EDA with Data Visualization (Plots Cont….)

EDA with SQL

- **Display the names of the unique launch sites in the space mission**
- **%sql** select distinct Launch_site as Launch_site from SPACEXTABLE;

- **Display 5 records where launch sites begin with the string 'CCA'**
- **%sql** select * from 'spacextbl' where Launch_site like 'CCA%' limit 5

- **Display the total payload mass carried by boosters launched by NASA (CRS)**
- *#%sql select sum(payload_mass_kg) as payload_mass from 'SPACEXTBL'* **%sql** SELECT SUM(PAYLOAD_MASS__KG_) as "Total Payload KG)", Customer FROM 'SPACEXTBL' WHERE Customer = 'NASA (CRS)’;

Sql contd..

- **Display average payload mass carried by booster version F9 v1.1**
- **%sql** select avg(PAYLOAD_MASS__KG_) as "Average mass kg", customer Booster_Version FROM 'SPACEXTBL' WHERE Booster_Version LIKE 'F9 v1.1%’;

- **List the date when the first succesful landing outcome in ground pad was acheived.**
- **%sql** SELECT min(date) FROM 'SPACEXTBL' WHERE "Landing_outcome" = "Success (ground pad)"
- https://github.com/Joejokelly/Spacex-falcon9-updated/blob/master/4.%20jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- Created folium map to mark launch sites, and markers, circles, lines to mark failure or success of launches for each site
- Created launch outcomes (failure = 0, success = 1)

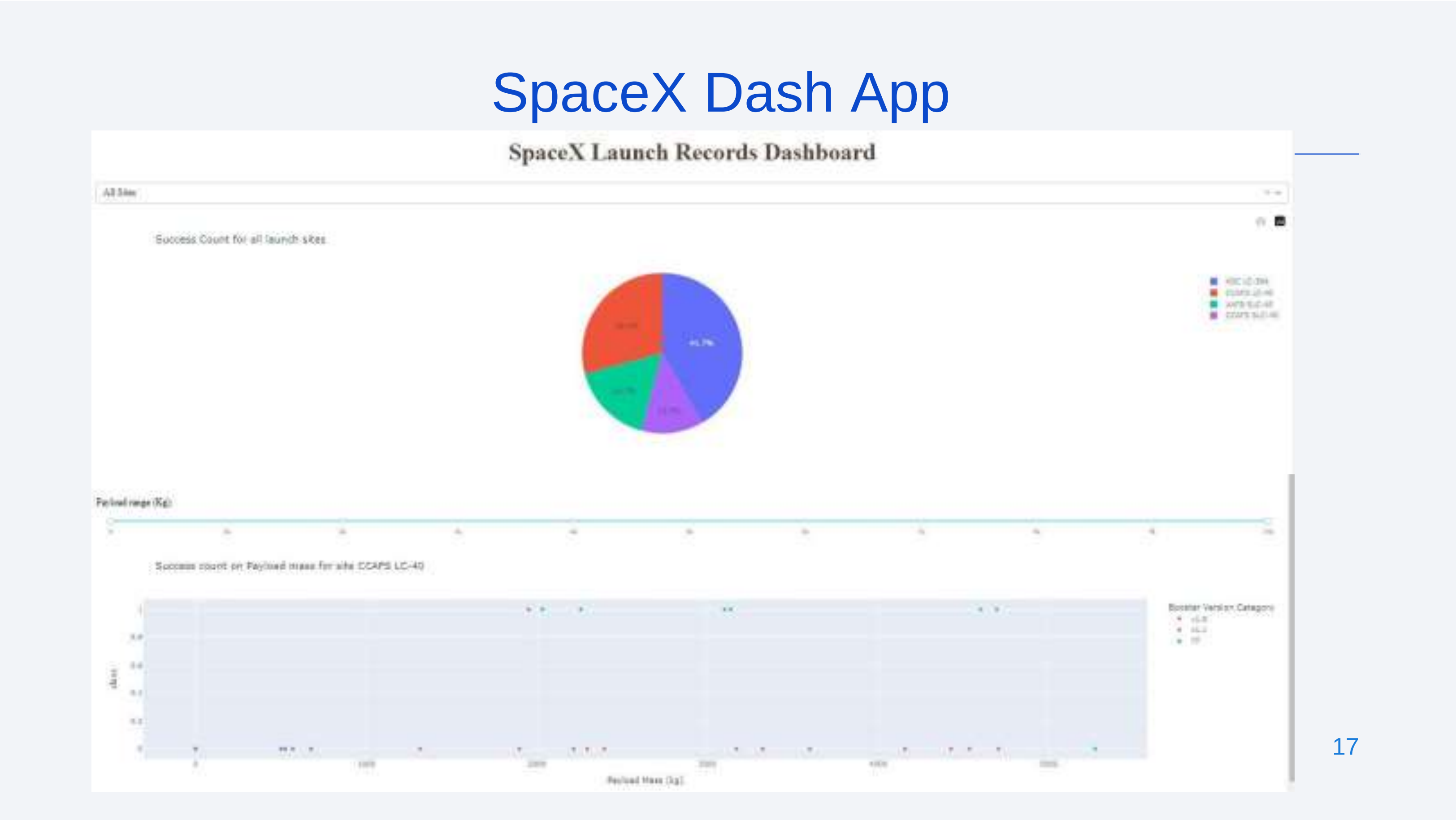
- https://github.com/Joejokelly/Spacex-falcon9-updated/blob/master/6.Space-X%20Launch%20Sites%20Locations%20Analysis%20with%20Folium-Interactive%20Visual%20Analytics.ipynb

Build a Dashboard with Plotly Dash

- Build an interactive dashboard application with Plotly dash ;
- Adding Launch Site Drop-down Input component
- Adding call-back function success-pie-chart
- Adding range slider to selected payload
- Adding call back function success-payload-scaler –chart

- Github link :
- https://github.com/Joejokelly/Spacex-falcon9-updated/blob/master/7.%20%20Build%20Interactive%20Dashboard%20-%20spacex_dash_app.py

Lauch Record Dashboard


SpaceX Dash App
SpaceX Launch Records Dashboard
All Sites
Success Count for all launch sites
Payload range (Kg)
Success count on Payload mass for site CCAFS LC-40
Booster Version Category
Payload Mass (kg)

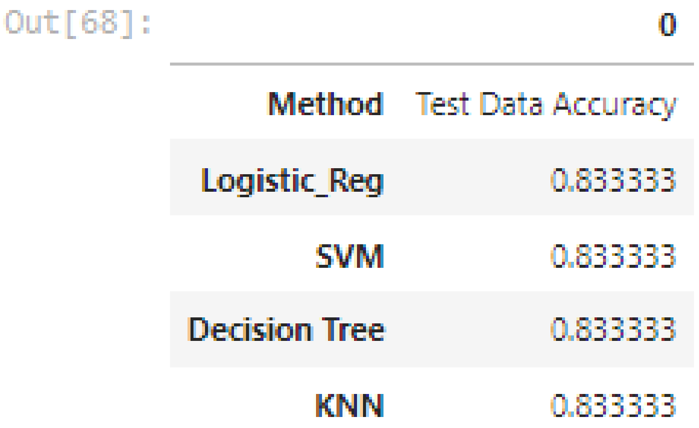
Predictive Analysis (Classification)

- Summary of how model was built, evaluated and improved :
  - After initial processing, data was loaded to the model using Pandas Data frame, created Numpy Array, then used StandardScaler function, to standardize the data.
  - Split the resulting data into training data and test data, train_test_Split from sklearn.
- In order to find the best Machine Language
  - Created GridsearchCV object, after fitting the training data, into GridsearchCV, displayed the best parameter using data attributes.
  - Used method score to calculate accuracy of each model and plotted confusion matrix for teach test and predicted outcomes.

- Add the GitHub URL of your completed predictive analysis lab, as an external reference and peer-review purpose

Predictive Analysis (Classification, contd)

- The table below shows the test accuracy score :



Out[68]:

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

- Github link :

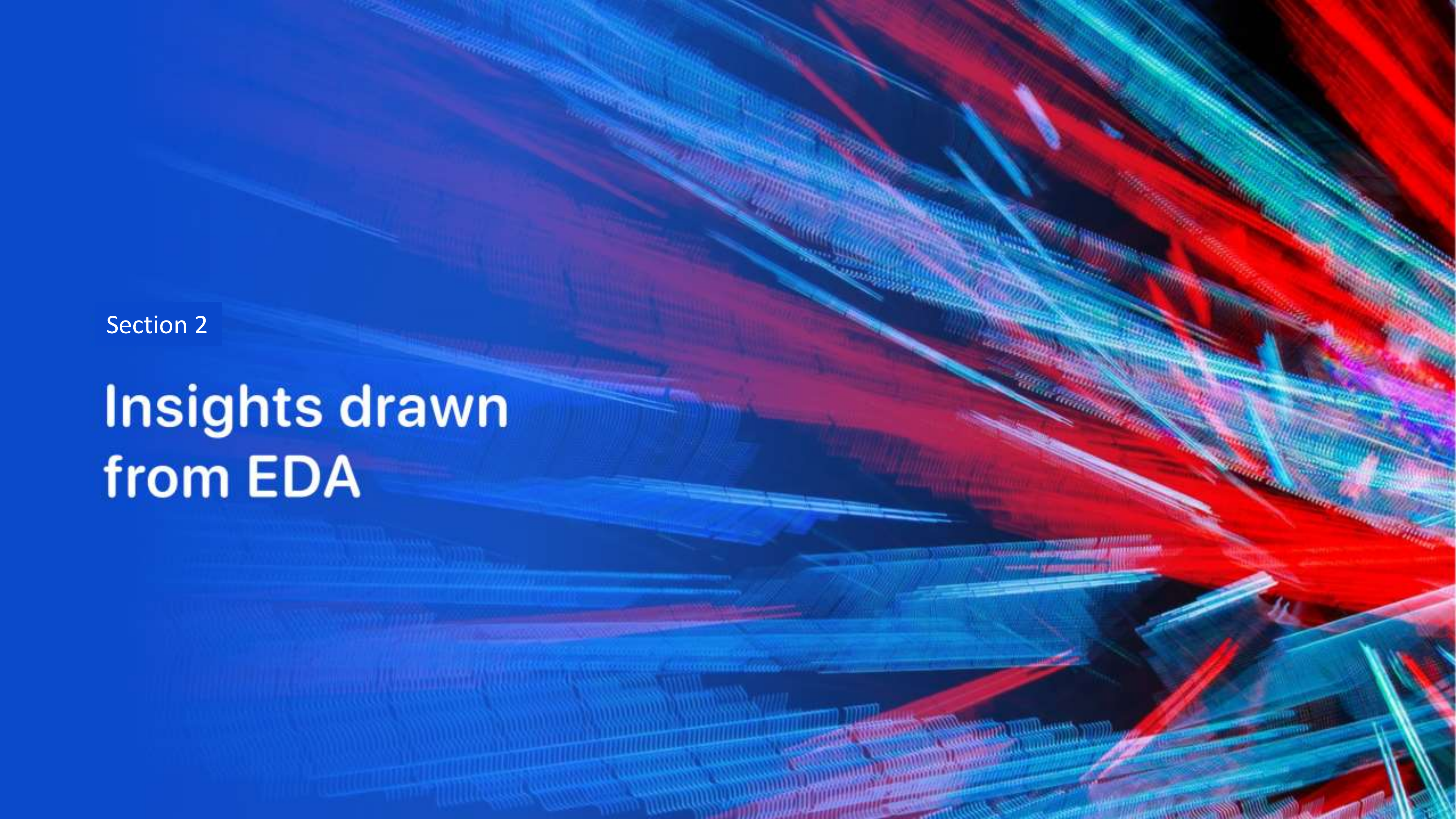
https://github.com/cgatama/SpaceX-Falcon-9-1st-stage-Success-Landing-Prediction/blob/main/8.%20SpaceX%20Machine%20Learning%20Prediction.ipynbhttps://github.com/cgatama/SpaceX-Falcon-9-1st-stage-Success-Landing-Prediction/blob/main/8.%20SpaceX%20Machine%20Learning%20Prediction.ipynb

Results

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

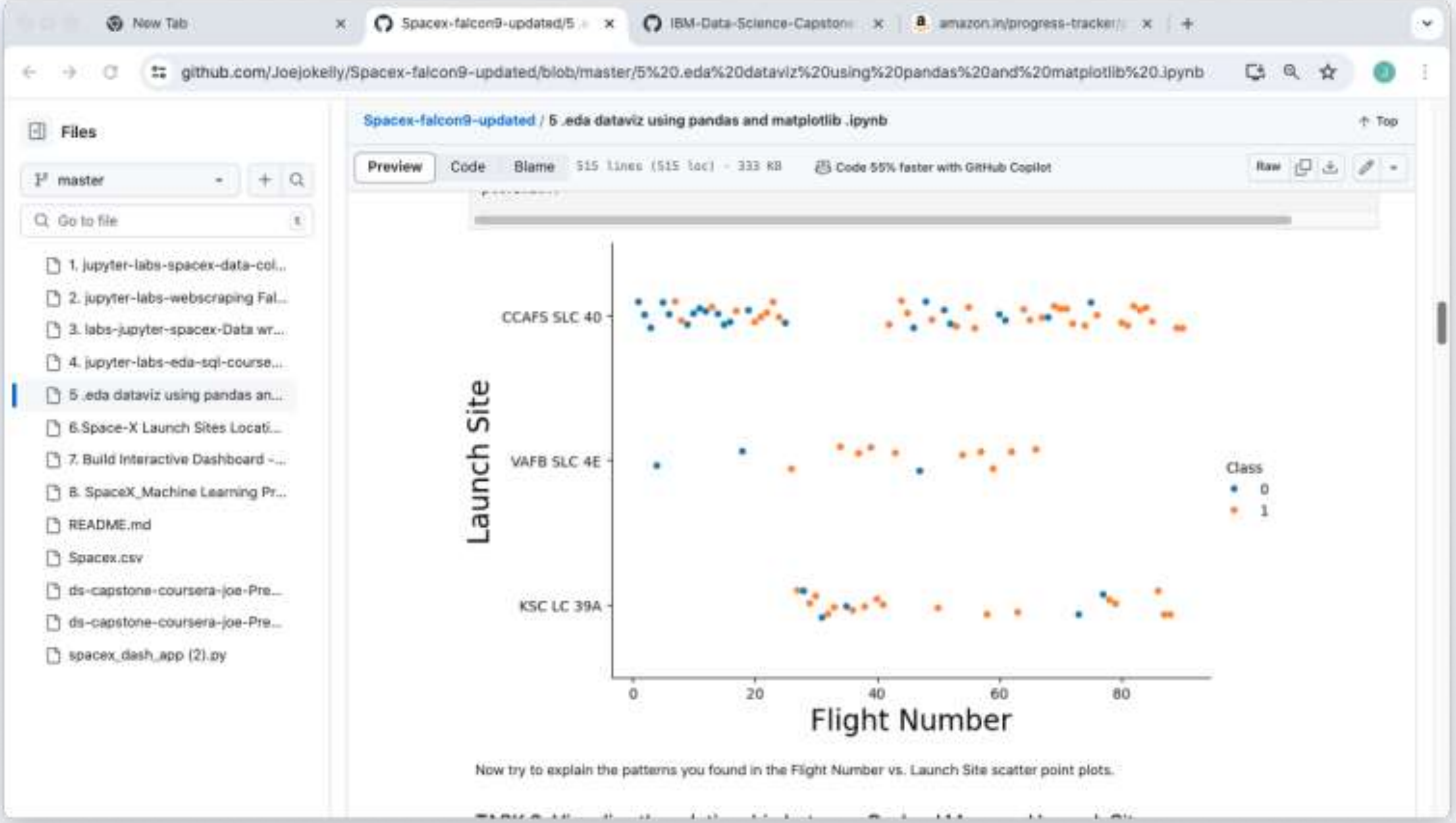
Section 2

Insights drawn from EDA



Flight Number vs. Launch Site

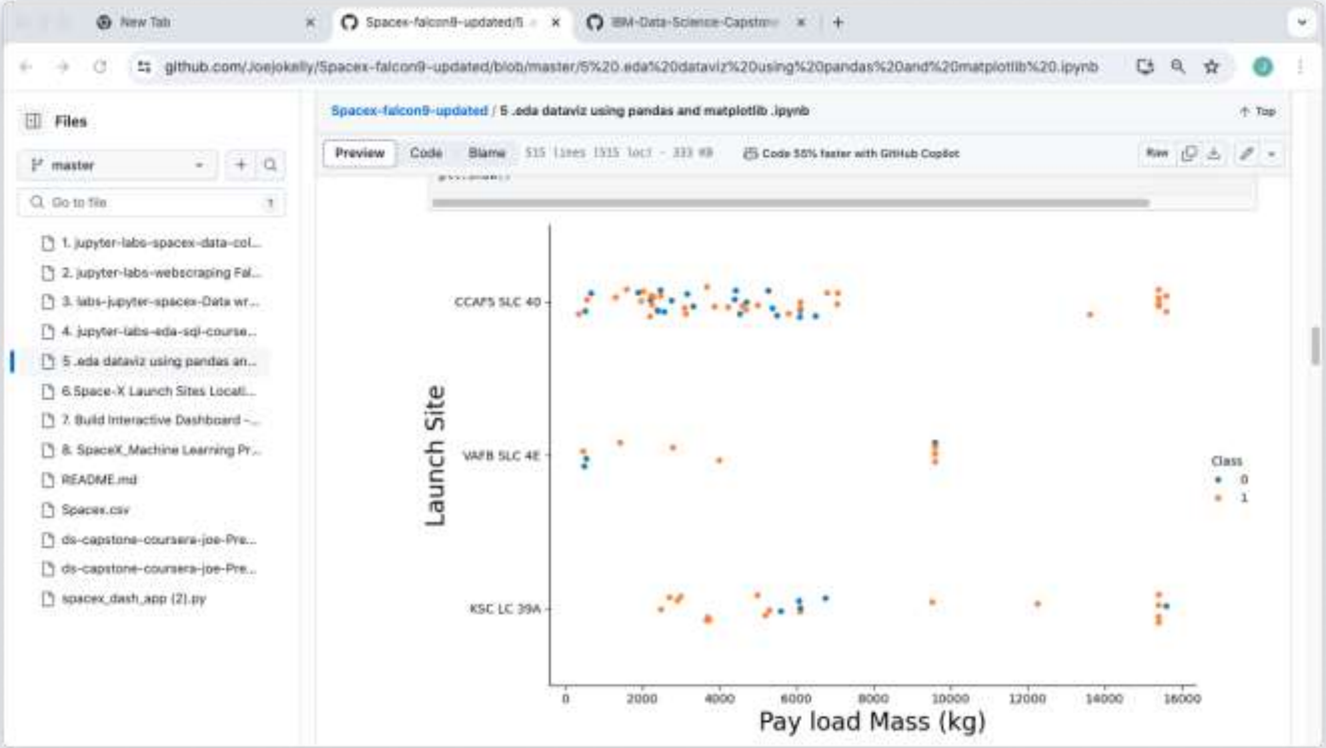
- scatter plot of Flight Number vs. Launch Site


New Tab
Spacex-falcon9-updated/5
IBM-Data-Science-Capstone
amazon.in/progress-tracker/
github.com/Joejokelly/Spacex-falcon9-updated/blob/master/5%20.eda%20dataviz%20using%20pandas%20and%20matplotlib%20.ipynb
Files
master
Go to file
1. jupyter-labs-spacex-data-col...
2. jupyter-labs-webscraping Fal...
3. labs-jupyter-spacex-Data wr...
4. jupyter-labs-eda-sql-course...
5 .eda dataviz using pandas an...
6.Space-X Launch Sites Locati...
7. Build Interactive Dashboard -...
8. SpaceX_Machine Learning Pr...
README.md
Spacex.csv
ds-capstone-coursera-joe-Pre...
ds-capstone-coursera-joe-Pre...
spacex_dash_app (2).py
Spacex-falcon9-updated / 5 .eda dataviz using pandas and matplotlib .ipynb
Top
Preview
Code
Blame
515 lines (515 loc) · 333 KB
Code 55% faster with GitHub Copilot
Raw
Launch Site
CCAFS SLC 40
VAFB SLC 4E
KSC LC 39A
Flight Number
0
20
40
60
80
Class
0
1

Now try to explain the patterns you found in the Flight Number vs. Launch Site scatter point plots.

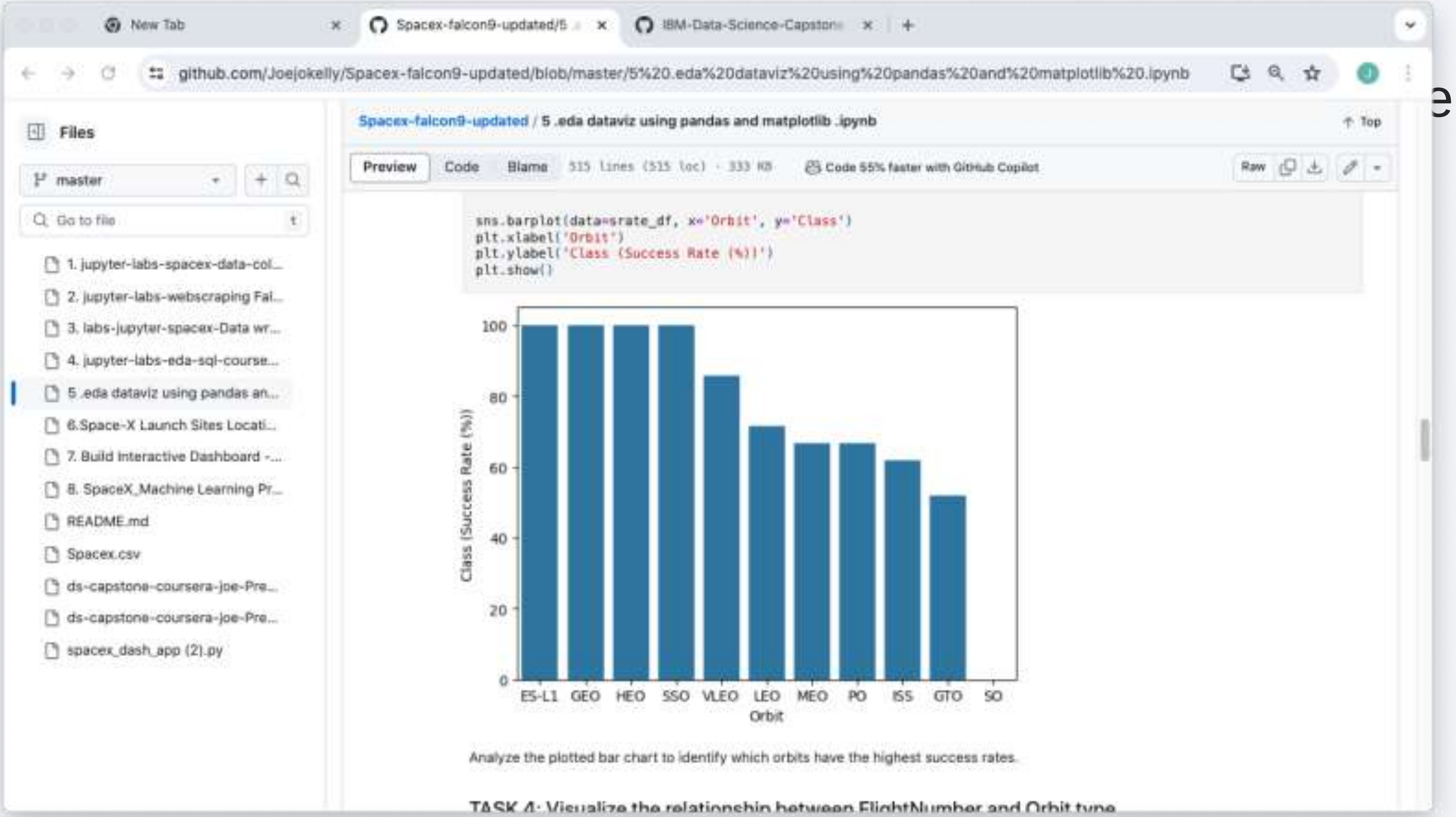
Payload vs. Launch Site

- Show a scatter plot of Payload vs. Launch Site


New Tab
Spacex-falcon9-updated/5
IBM-Data-Science-Capstone
github.com/Joejokelly/Spacex-falcon9-updated/blob/master/5%20.eda%20dataviz%20using%20pandas%20and%20matplotlib%20.ipynb
Files
master
Go to file
1. jupyter-labs-spacex-data-col...
2. jupyter-labs-webscraping Fal...
3. labs-jupyter-spacex-Data wr...
4. jupyter-labs-eda-sql-course...
5 .eda dataviz using pandas an...
6.Space-X Launch Sites Locati...
7. Build Interactive Dashboard -...
8. SpaceX_Machine Learning Pr...
README.md
Spacex.csv
ds-capstone-coursera-joe-Pre...
ds-capstone-coursera-joe-Pre...
spacex_dash_app (2).py
Spacex-falcon9-updated / 5 .eda dataviz using pandas and matplotlib .ipynb
Top
Preview
Code
Blame
515 lines (515 loc) · 333 KB
Code 55% faster with GitHub Copilot
Raw
Launch Site
CCAFS SLC 40
VAFB SLC 4E
KSC LC 39A
Pay load Mass (kg)
0
2000
4000
6000
8000
10000
12000
14000
16000
Class
0
1

Payload vs Launch site scatter plot, there are no rocket lauches for heavy payload > 10000

Success Rate vs. Orbit Type


New Tab
Spacex-falcon9-updated/5
IBM-Data-Science-Capstone
github.com/Joejokelly/Spacex-falcon9-updated/blob/master/5%20.eda%20dataviz%20using%20pandas%20and%20matplotlib%20.ipynb
Files
master
Go to file
1. jupyter-labs-spacex-data-col...
2. jupyter-labs-webscraping Fal...
3. labs-jupyter-spacex-Data wr...
4. jupyter-labs-eda-sql-course...
5 .eda dataviz using pandas an...
6.Space-X Launch Sites Locati...
7. Build Interactive Dashboard -...
8. SpaceX_Machine Learning Pr...
README.md
Spacex.csv
ds-capstone-coursera-joe-Pre...
ds-capstone-coursera-joe-Pre...
spacex_dash_app (2).py
Spacex-falcon9-updated / 5 .eda dataviz using pandas and matplotlib .ipynb
Top
Preview
Code
Blame
515 lines (515 loc) · 333 KB
Code 55% faster with GitHub Copilot
Raw
100
80
60
40
20
0
Class (Success Rate (%))
ES-L1
GEO
HEO
SSO
VLEO
LEO
MEO
PO
ISS
GTO
SO
Orbit

```
sns.barplot(data=srate_df, x='Orbit', y='Class')
plt.xlabel('Orbit')
plt.ylabel('Class (Success Rate (%))')
plt.show()
```

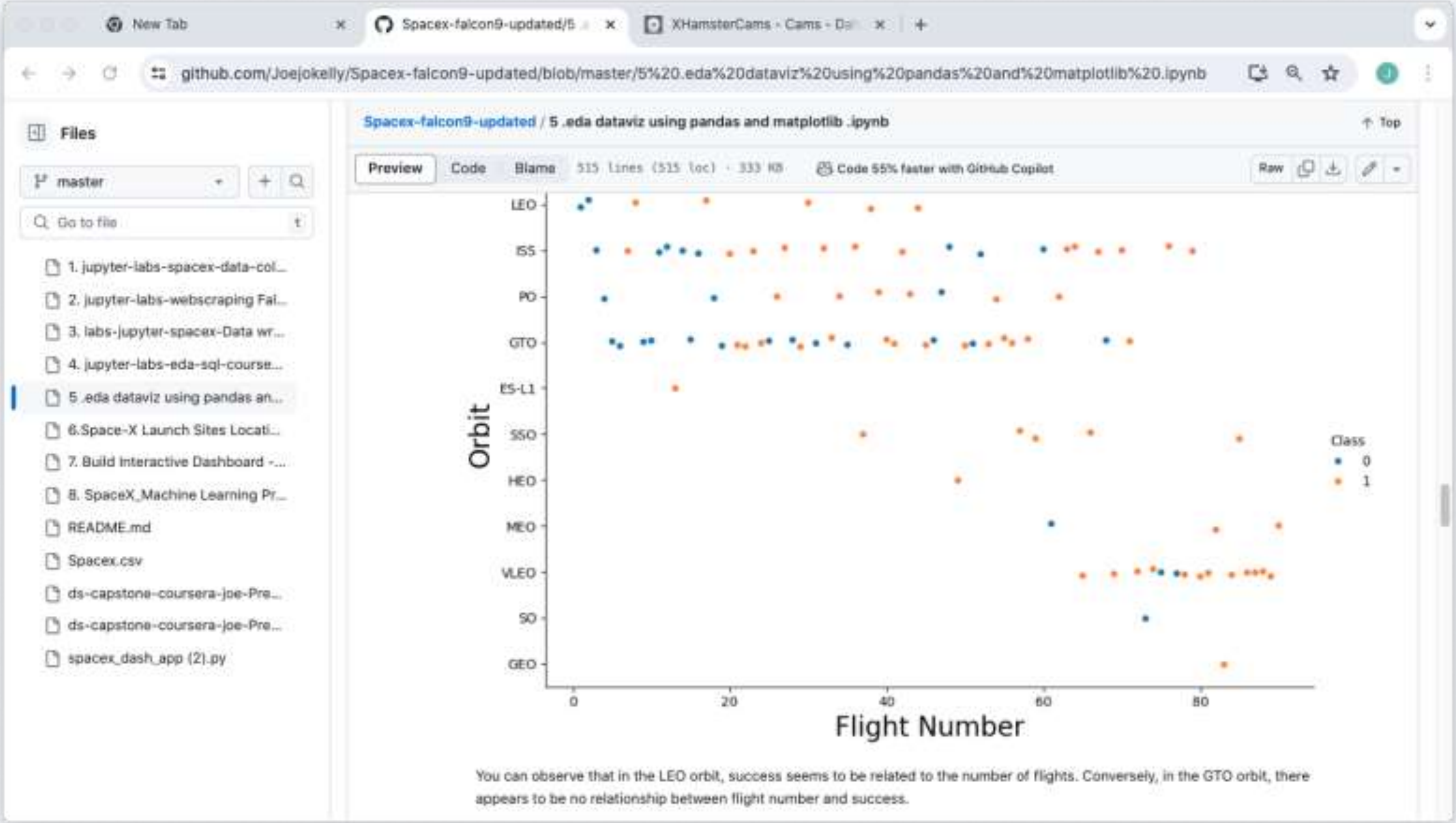
Analyze the plotted bar chart to identify which orbits have the highest success rates.

TASK 4: Visualize the relationship between FlightNumber and Orbit type

Orbits ES-L1, Geo, Heo, seo, have a higher rate of success, 100%m while GTO has about 50%

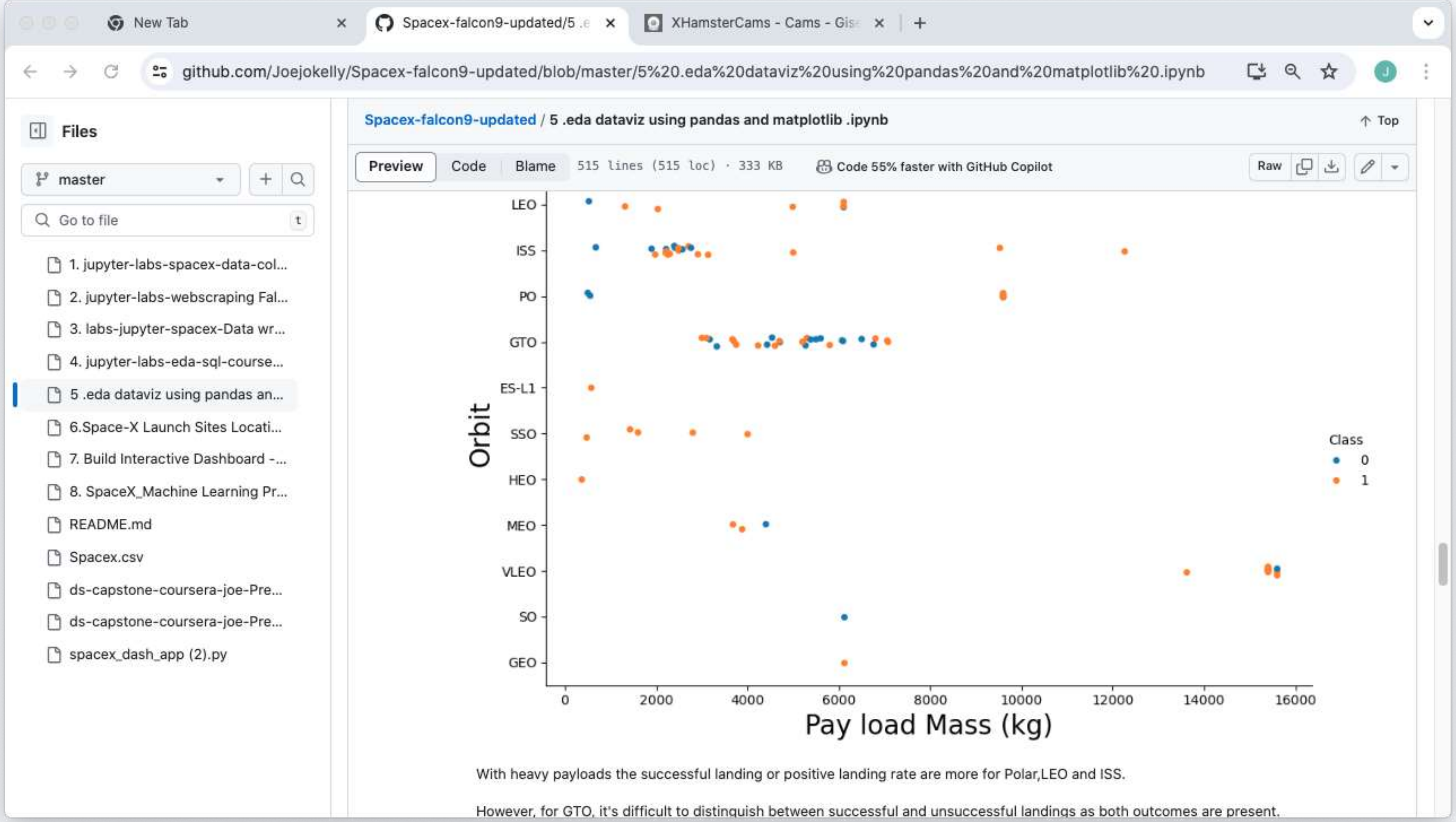
Flight Number vs. Orbit Type

- Show a scatter point o Flight number vs. Orbit type


New Tab
Spacex-falcon9-updated/5
XHamsterCams - Cams - Da
github.com/Joejokelly/Spacex-falcon9-updated/blob/master/5%20.eda%20dataviz%20using%20pandas%20and%20matplotlib%20.ipynb
Files
master
Go to file
1. jupyter-labs-spacex-data-col...
2. jupyter-labs-webscraping Fal...
3. labs-jupyter-spacex-Data wr...
4. jupyter-labs-eda-sql-course...
5 .eda dataviz using pandas an...
6.Space-X Launch Sites Locati...
7. Build Interactive Dashboard -...
8. SpaceX_Machine Learning Pr...
README.md
Spacex.csv
ds-capstone-coursera-joe-Pre...
ds-capstone-coursera-joe-Pre...
spacex_dash_app (2).py
Spacex-falcon9-updated / 5 .eda dataviz using pandas and matplotlib .ipynb
Top
Preview
Code
Blame
515 lines (515 loc) · 333 KB
Code 55% faster with GitHub Copilot
Raw
LEO
ISS
PO
GTO
ES-L1
SSO
HEO
MEO
VLEO
SO
GEO
Orbit
0
20
40
60
80
Flight Number
Class
0
1

You can observe that in the LEO orbit, success seems to be related to the number of flights. Conversely, in the GTO orbit, there appears to be no relationship between flight number and success.

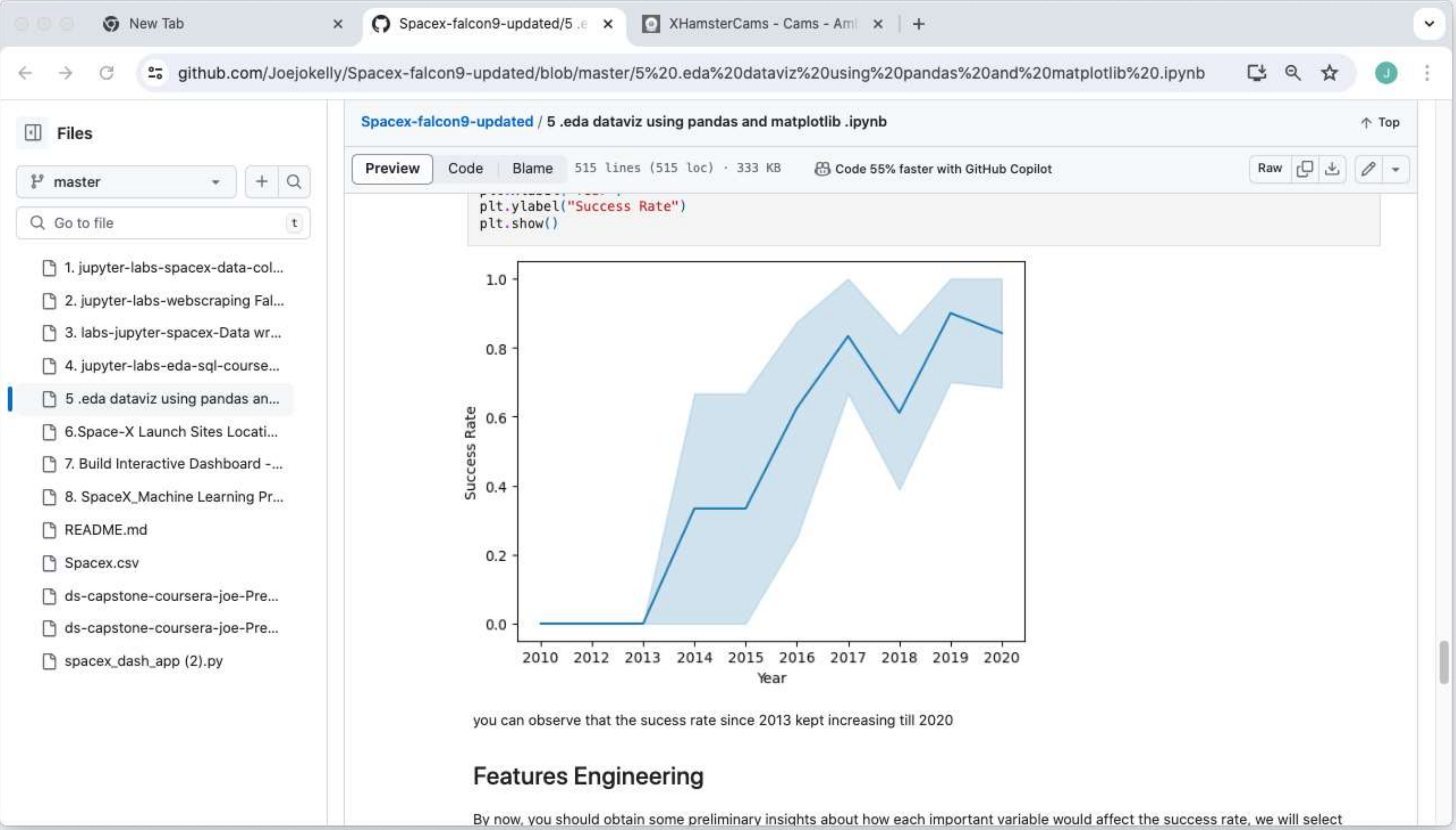
Payload vs. Orbit Type


New Tab
Spacex-falcon9-updated/5 .e
XHamsterCams - Cams - Gis
github.com/Joejokelly/Spacex-falcon9-updated/blob/master/5%20.eda%20dataviz%20using%20pandas%20and%20matplotlib%20.ipynb
Files
master
Go to file
t
1. jupyter-labs-spacex-data-col...
2. jupyter-labs-webscraping Fal...
3. labs-jupyter-spacex-Data wr...
4. jupyter-labs-eda-sql-course...
5 .eda dataviz using pandas an...
6.Space-X Launch Sites Locati...
7. Build Interactive Dashboard -...
8. SpaceX_Machine Learning Pr...
README.md
Spacex.csv
ds-capstone-coursera-joe-Pre...
ds-capstone-coursera-joe-Pre...
spacex_dash_app (2).py
Spacex-falcon9-updated / 5 .eda dataviz using pandas and matplotlib .ipynb
Top
Preview
Code
Blame
515 lines (515 loc) · 333 KB
Code 55% faster with GitHub Copilot
Raw
Orbit
LEO
ISS
PO
GTO
ES-L1
SSO
HEO
MEO
VLEO
SO
GEO
0
2000
4000
6000
8000
10000
12000
14000
16000
Pay load Mass (kg)
Class
0
1

With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS.

However, for GTO, it's difficult to distinguish between successful and unsuccessful landings as both outcomes are present.

Launch Success Yearly Trend


New Tab
Spacex-falcon9-updated/5
XHamsterCams - Cams - Am
github.com/Joejokelly/Spacex-falcon9-updated/blob/master/5%20.eda%20dataviz%20using%20pandas%20and%20matplotlib%20.ipynb
Files
master
Go to file
1. jupyter-labs-spacex-data-col...
2. jupyter-labs-webscraping Fal...
3. labs-jupyter-spacex-Data wr...
4. jupyter-labs-eda-sql-course...
5 .eda dataviz using pandas an...
6.Space-X Launch Sites Locati...
7. Build Interactive Dashboard -...
8. SpaceX_Machine Learning Pr...
README.md
Spacex.csv
ds-capstone-coursera-joe-Pre...
ds-capstone-coursera-joe-Pre...
spacex_dash_app (2).py
Spacex-falcon9-updated / 5 .eda dataviz using pandas and matplotlib .ipynb
Top
Preview
Code
Blame
515 lines (515 loc) · 333 KB
Code 55% faster with GitHub Copilot
Raw
Success Rate
1.0
0.8
0.6
0.4
0.2
0.0
2010
2012
2013
2014
2015
2016
2017
2018
2019
2020
Year

```
plt.ylabel("Success Rate")
plt.show()
```

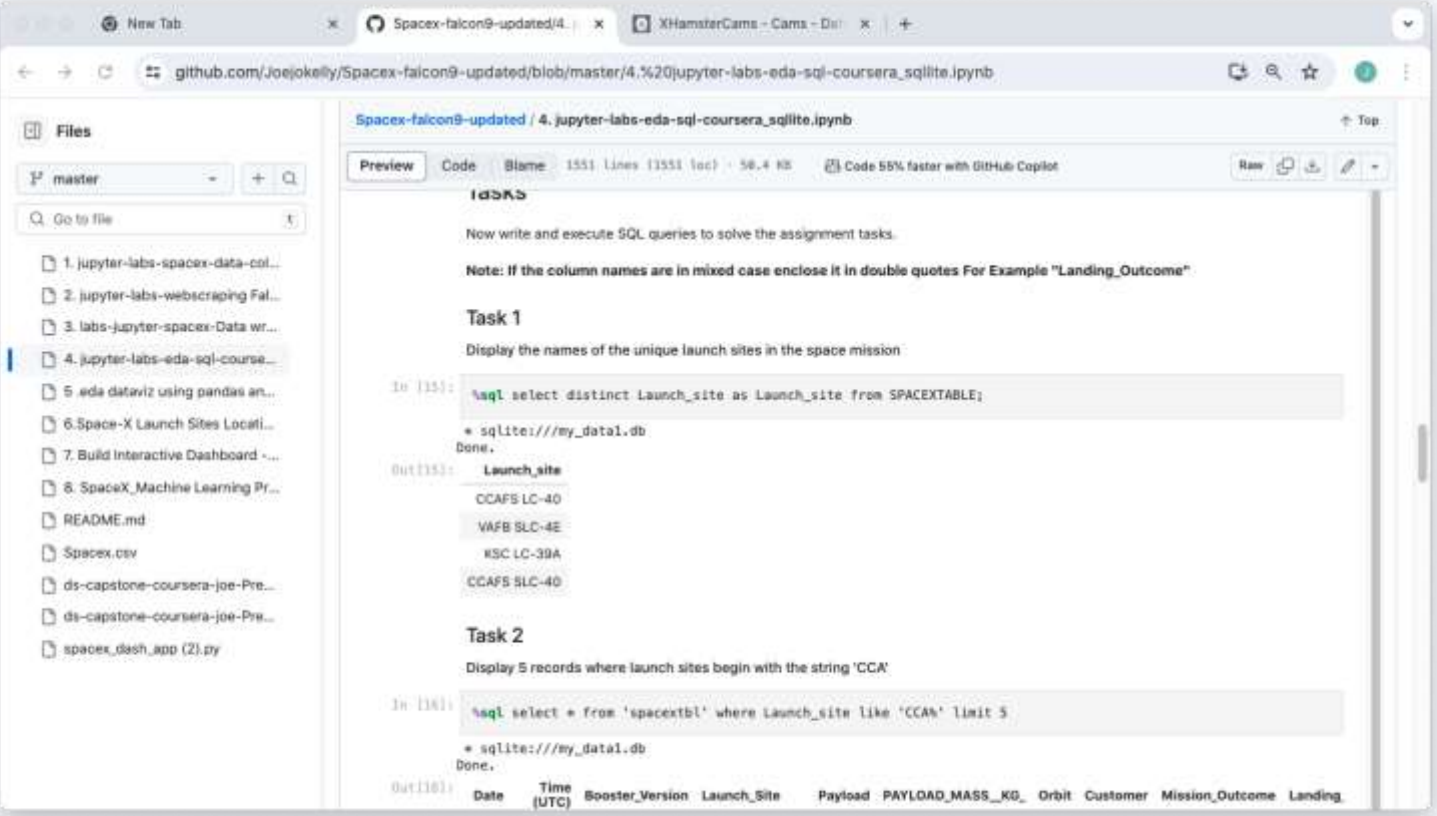
you can observe that the sucess rate since 2013 kept increasing till 2020

Features Engineering

By now, you should obtain some preliminary insights about how each important variable would affect the success rate, we will select

All Launch Site Names

- Find the names of the unique launch sites


New Tab
Spacex-falcon9-updated/4.
XHamsterCams - Cams - Dat
github.com/Joejokelly/Spacex-falcon9-updated/blob/master/4.%20jupyter-labs-eda-sql-coursera_sqllite.ipynb
Files
master
Go to file
1. jupyter-labs-spacex-data-col...
2. jupyter-labs-webscraping Fal...
3. labs-jupyter-spacex-Data wr...
4. jupyter-labs-eda-sql-course...
5 .eda dataviz using pandas an...
6.Space-X Launch Sites Locati...
7. Build Interactive Dashboard -...
8. SpaceX_Machine Learning Pr...
README.md
Spacex.csv
ds-capstone-coursera-joe-Pre...
ds-capstone-coursera-joe-Pre...
spacex_dash_app (2).py
Spacex-falcon9-updated / 4. jupyter-labs-eda-sql-coursera_sqllite.ipynb
Top
Preview
Code
Blame
1551 lines (1551 loc) · 50.4 KB
Code 55% faster with GitHub Copilot
Raw

Tasks

Now write and execute SQL queries to solve the assignment tasks.

Note: If the column names are in mixed case enclose it in double quotes For Example "Landing_Outcome"

Task 1

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

```
In [15]: %sql select distinct Launch_site as Launch_site from SPACEXTABLE;
```

```
 * sqlite:///my_data1.db
Done.
Out[15]:  Launch_site
          CCAFS LC-40
          VAFB SLC-4E
          KSC LC-39A
          CCAFS SLC-40
```

Task 2

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

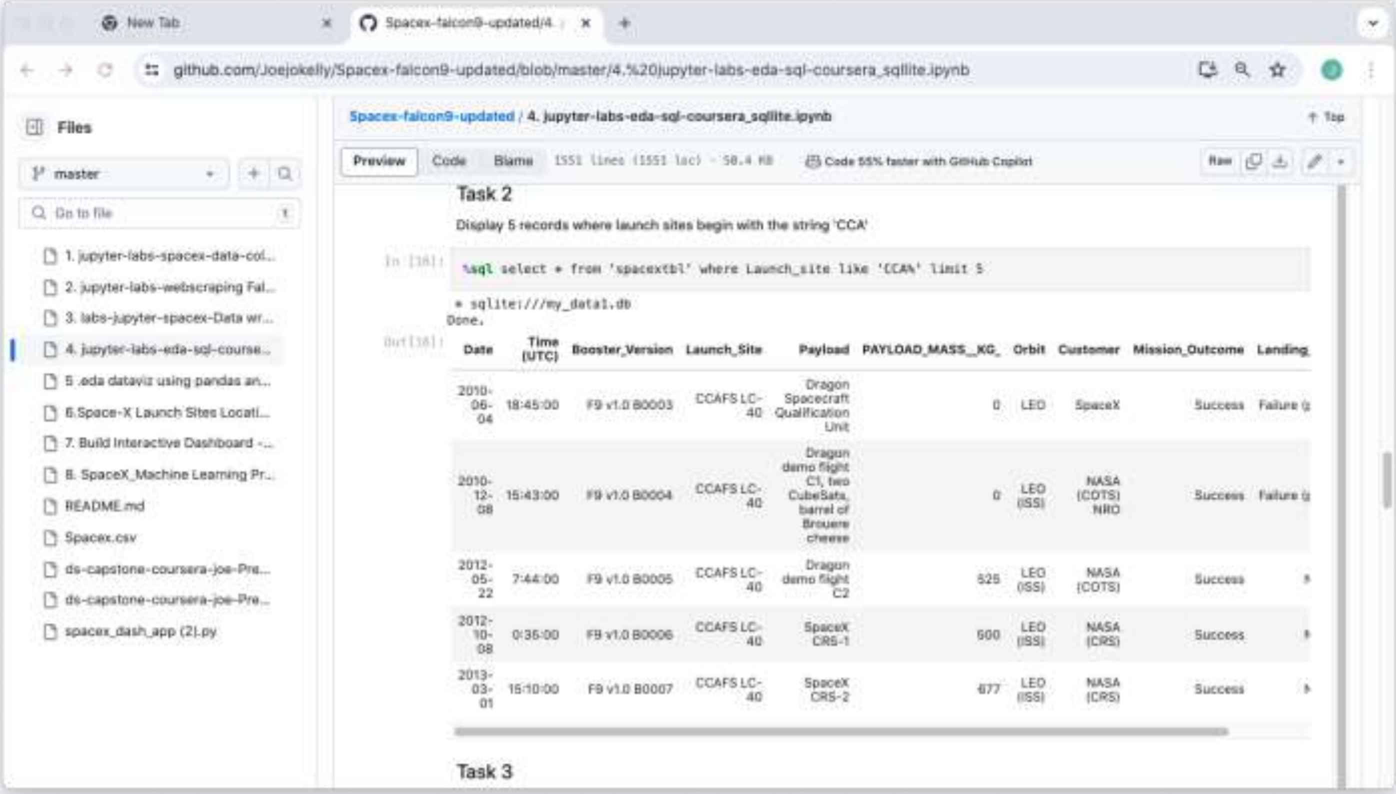
```
In [16]: %sql select * from 'spacextbl' where Launch_site like 'CCA%' limit 5
```

```
 * sqlite:///my_data1.db
Done.
Out[16]:  Date   Time (UTC)   Booster_Version   Launch_Site   Payload   PAYLOAD_MASS__KG_   Orbit   Customer   Mission_Outcome   Landing
```

%sql select distinct Launch_site as Launch_site from SPACEXTABLE;

Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with `CCA`


New Tab
Spacex-falcon9-updated/4.
github.com/Joejokelly/Spacex-falcon9-updated/blob/master/4.%20jupyter-labs-eda-sql-coursera_sqllite.ipynb
Files
master
Go to file
1. jupyter-labs-spacex-data-col...
2. jupyter-labs-webscraping Fal...
3. labs-jupyter-spacex-Data wr...
4. jupyter-labs-eda-sql-course...
5 .eda dataviz using pandas an...
6.Space-X Launch Sites Locati...
7. Build Interactive Dashboard -...
8. SpaceX_Machine Learning Pr...
README.md
Spacex.csv
ds-capstone-coursera-joe-Pre...
ds-capstone-coursera-joe-Pre...
spacex_dash_app (2).py
Spacex-falcon9-updated / 4. jupyter-labs-eda-sql-coursera_sqllite.ipynb
Top
Preview
Code
Blame
1551 lines (1551 loc) · 50.4 KB
Code 55% faster with GitHub Copilot
Raw

Task 2

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

```
In [18]: %sql select * from 'spacextbl' where Launch_site like 'CCA%' limit 5
```

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

Out[18]:

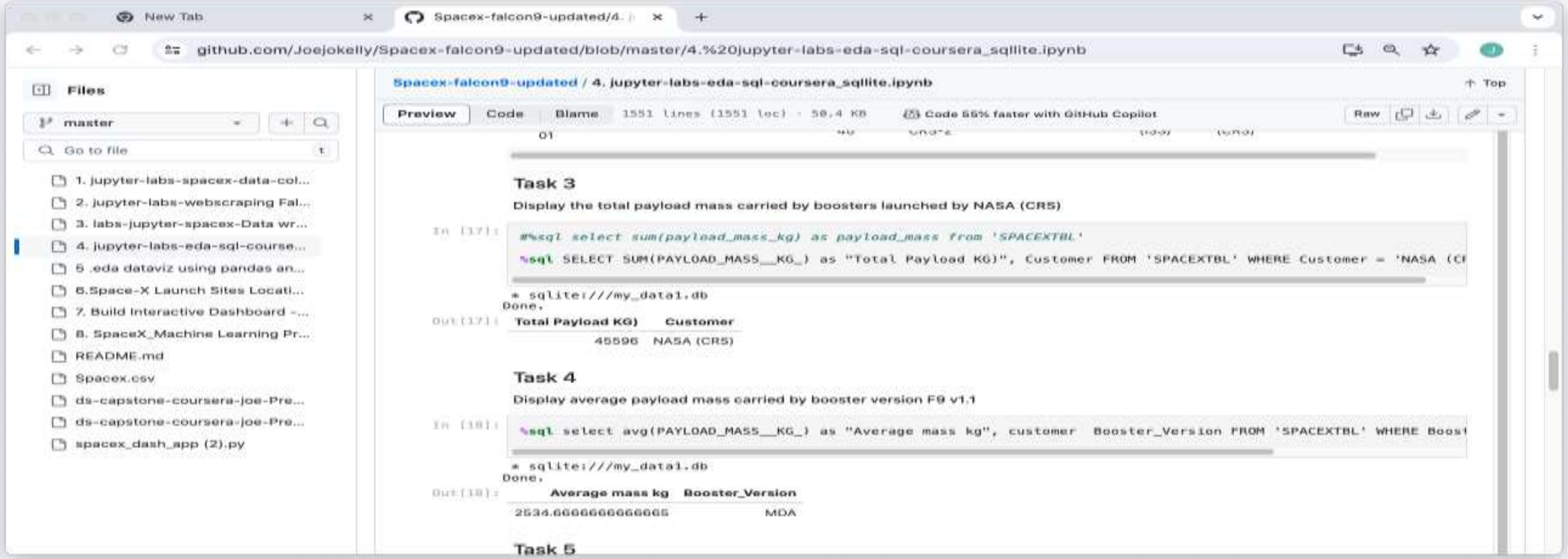
Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	

Task 3

Used wild cars in the above query to display records with site begin with ‘CCA’

Total Payload Mass

- Calculate the total payload carried by boosters from NASA


New Tab
Spacex-falcon9-updated/4.
github.com/Joejokelly/Spacex-falcon9-updated/blob/master/4.%20jupyter-labs-eda-sql-coursera_sqllite.ipynb
Files
master
Go to file
1. jupyter-labs-spacex-data-col...
2. jupyter-labs-webscraping Fal...
3. labs-jupyter-spacex-Data wr...
4. jupyter-labs-eda-sql-course...
5 .eda dataviz using pandas an...
6.Space-X Launch Sites Locati...
7. Build Interactive Dashboard -...
8. SpaceX_Machine Learning Pr...
README.md
Spacex.csv
ds-capstone-coursera-joe-Pre...
ds-capstone-coursera-joe-Pre...
spacex_dash_app (2).py
Spacex-falcon9-updated / 4. jupyter-labs-eda-sql-coursera_sqllite.ipynb
Top
Preview
Code
Blame
1551 lines (1551 loc) · 50.4 KB
Code 55% faster with GitHub Copilot
Raw

Task 3

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

```
In [17]: #%sql select sum(payload_mass_kg) as payload_mass from 'SPACEXTBL'
         %sql SELECT SUM(PAYLOAD_MASS__KG_) as "Total Payload KG)", Customer FROM 'SPACEXTBL' WHERE Customer = 'NASA (C
```

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

Out[17]:

Total Payload KG)	Customer
45596	NASA (CRS)

Task 4

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

```
In [18]: %sql select avg(PAYLOAD_MASS__KG_) as "Average mass kg", customer  Booster_Version FROM 'SPACEXTBL' WHERE Boos
```

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

Out[18]:

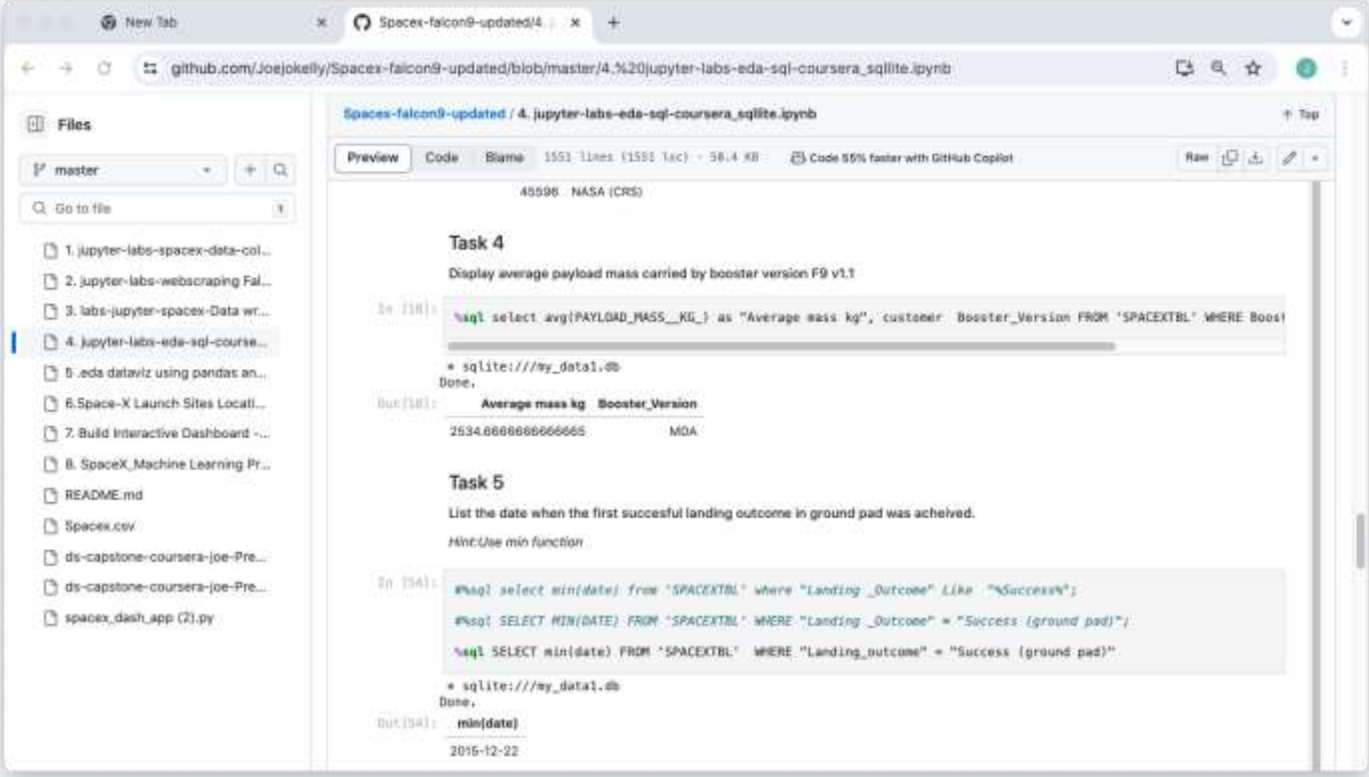
Average mass kg	Booster_Version
2534.6666666666665	MDA

Task 5

Used the sum function to display the total payload (kg)

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1


New Tab
Spacex-falcon9-updated/4
github.com/Joejokelly/Spacex-falcon9-updated/blob/master/4.%20jupyter-labs-eda-sql-coursera_sqllite.ipynb
Files
master
Go to file
1. jupyter-labs-spacex-data-col...
2. jupyter-labs-webscraping Fal...
3. labs-jupyter-spacex-Data wr...
4. jupyter-labs-eda-sql-course...
5 .eda dataviz using pandas an...
6.Space-X Launch Sites Locati...
7. Build Interactive Dashboard -...
8. SpaceX_Machine Learning Pr...
README.md
Spacex.csv
ds-capstone-coursera-joe-Pre...
ds-capstone-coursera-joe-Pre...
spacex_dash_app (2).py
Spaces-falcon9-updated / 4. jupyter-labs-eda-sql-coursera_sqllite.ipynb
Top
Preview
Code
Blame
1551 lines (1551 loc) · 58.4 KB
Code 55% faster with GitHub Copilot
Raw

45596 NASA (CRS)

Task 4

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

```
In [18]: %sql select avg(PAYLOAD_MASS__KG_) as "Average mass kg", customer  Booster_Version FROM 'SPACEXTBL' WHERE Boos
```

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

Out[18]:

Average mass kg	Booster_Version
2534.6666666666665	MDA

Task 5

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

Hint:Use min function

```
In [54]: #%sql select min(date) from 'SPACEXTBL' where "Landing _Outcome" Like  "%Success%";

#%sql SELECT MIN(DATE) FROM 'SPACEXTBL' WHERE "Landing _Outcome" = "Success (ground pad)";

%sql SELECT min(date) FROM 'SPACEXTBL'  WHERE "Landing_outcome" = "Success (ground pad)"
```

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

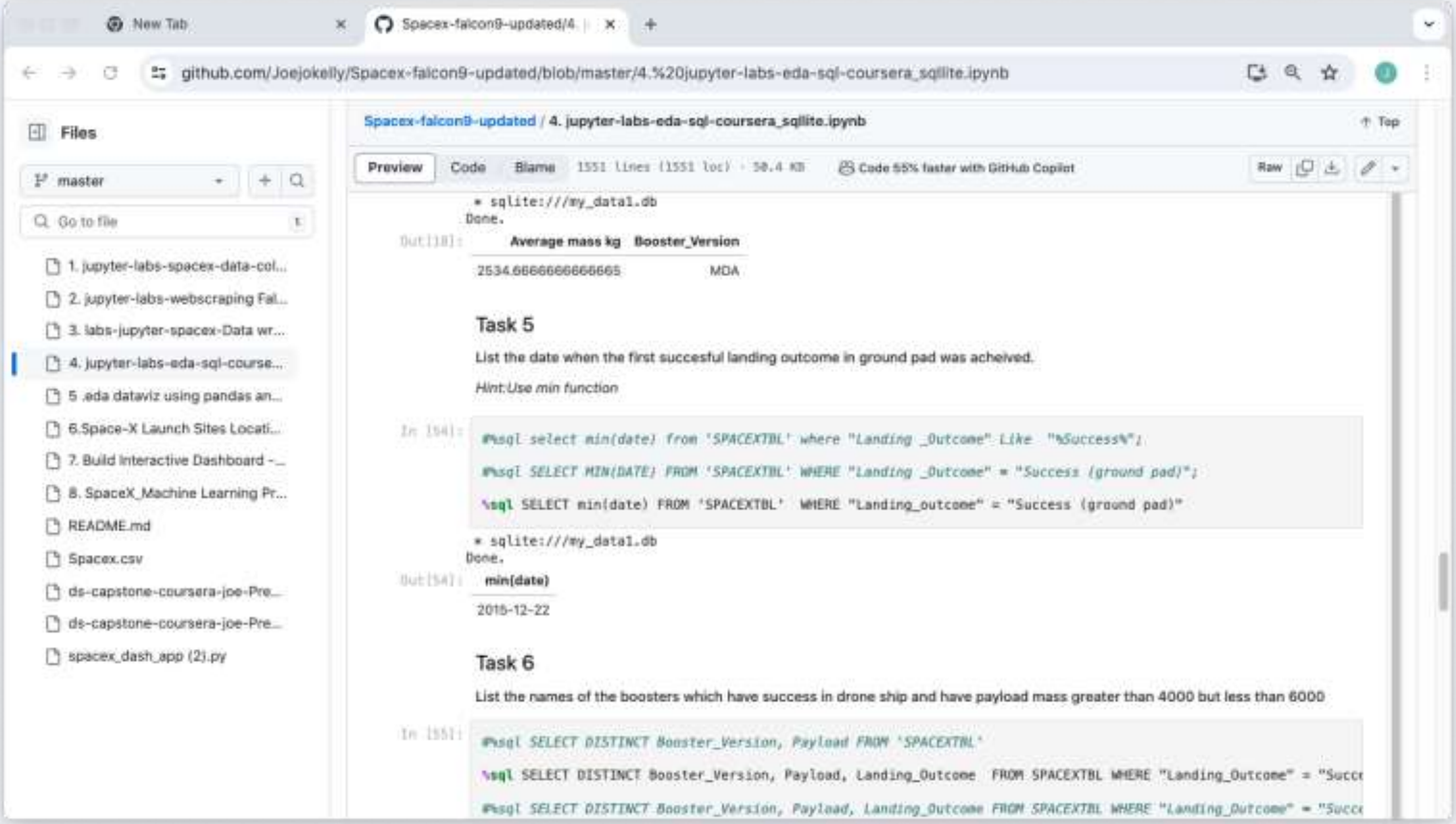
Out[54]:

min(date)
2015-12-22

Used the avg() function to display average payload booster version

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad


New Tab
Spacex-falcon9-updated/4.
github.com/Joejokelly/Spacex-falcon9-updated/blob/master/4.%20jupyter-labs-eda-sql-coursera_sqllite.ipynb
Files
master
Go to file
1. jupyter-labs-spacex-data-col...
2. jupyter-labs-webscraping Fal...
3. labs-jupyter-spacex-Data wr...
4. jupyter-labs-eda-sql-course...
5 .eda dataviz using pandas an...
6.Space-X Launch Sites Locati...
7. Build Interactive Dashboard -...
8. SpaceX_Machine Learning Pr...
README.md
Spacex.csv
ds-capstone-coursera-joe-Pre...
ds-capstone-coursera-joe-Pre...
spacex_dash_app (2).py
Spacex-falcon9-updated / 4. jupyter-labs-eda-sql-coursera_sqllite.ipynb
Top
Preview
Code
Blame
1551 lines (1551 loc) · 50.4 KB
Code 55% faster with GitHub Copilot
Raw

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

Out[18]:

Average mass kg	Booster_Version
2534.6666666666665	MDA

Task 5

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

Hint:Use min function

In [54]:

```
#%sql select min(date) from 'SPACEXTBL' where "Landing _Outcome" Like  "%Success%";

#%sql SELECT MIN(DATE) FROM 'SPACEXTBL' WHERE "Landing _Outcome" = "Success (ground pad)";

%sql SELECT min(date) FROM 'SPACEXTBL'  WHERE "Landing_outcome" = "Success (ground pad)"
```

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

Out[54]:

min(date)
2015-12-22

Task 6

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

In [55]:

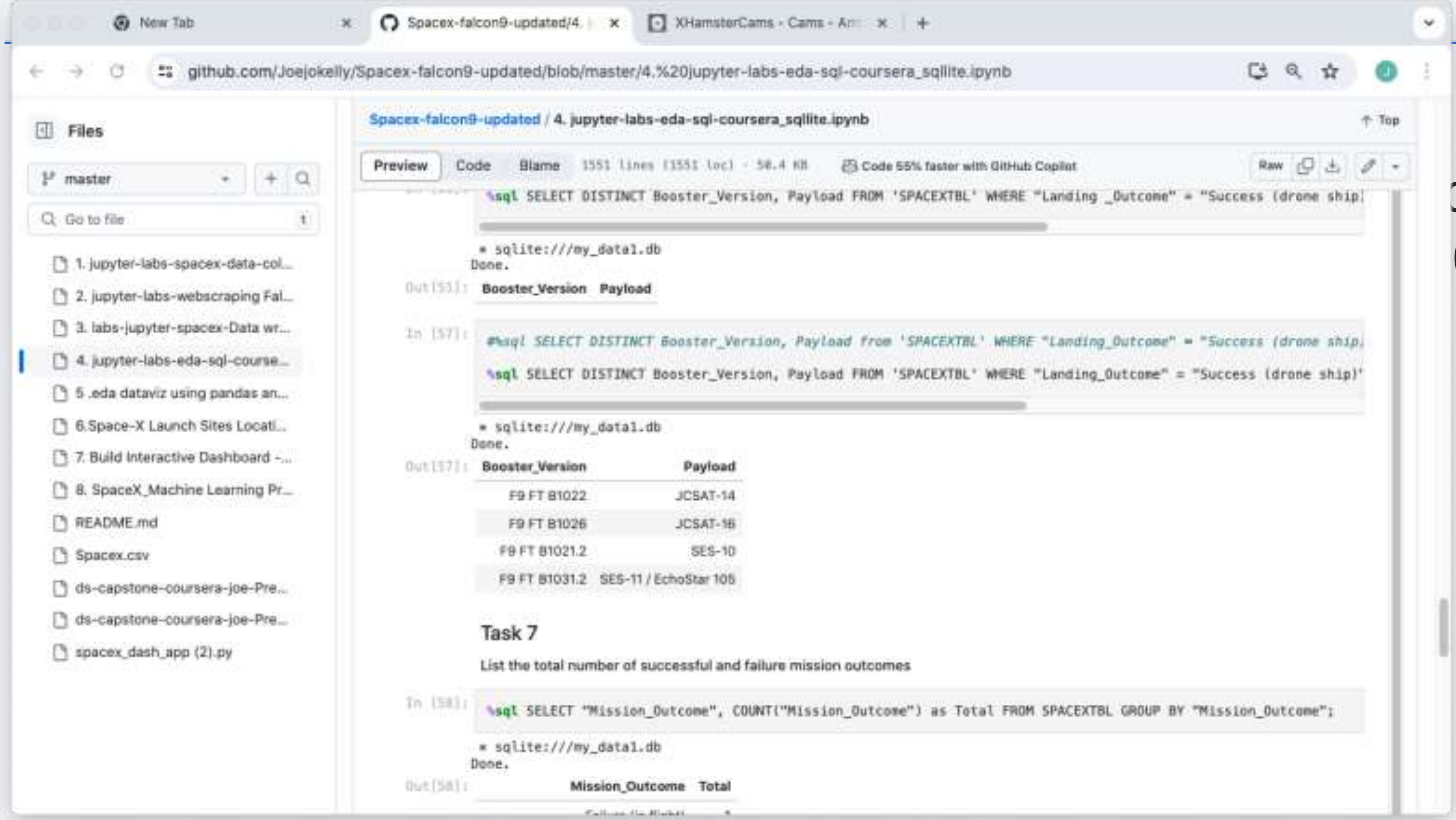
```
#%sql SELECT DISTINCT Booster_Version, Payload FROM 'SPACEXTBL'

%sql SELECT DISTINCT Booster_Version, Payload, Landing_Outcome  FROM SPACEXTBL WHERE "Landing_Outcome" = "Succ

#%sql SELECT DISTINCT Booster_Version, Payload, Landing_Outcome FROM SPACEXTBL WHERE "Landing_Outcome" = "Succ
```

Successful Drone Ship Landing with Payload between 4000 and 6000

d on drone ship
00


New Tab
Spacex-falcon9-updated/4.
XHamsterCams - Cams - An
github.com/Joejokelly/Spacex-falcon9-updated/blob/master/4.%20jupyter-labs-eda-sql-coursera_sqllite.ipynb
Files
master
Go to file
1. jupyter-labs-spacex-data-col...
2. jupyter-labs-webscraping Fal...
3. labs-jupyter-spacex-Data wr...
4. jupyter-labs-eda-sql-course...
5 .eda dataviz using pandas an...
6.Space-X Launch Sites Locati...
7. Build Interactive Dashboard -...
8. SpaceX_Machine Learning Pr...
README.md
Spacex.csv
ds-capstone-coursera-joe-Pre...
ds-capstone-coursera-joe-Pre...
spacex_dash_app (2).py
Spacex-falcon9-updated / 4. jupyter-labs-eda-sql-coursera_sqllite.ipynb
Top
Preview
Code
Blame
1551 lines (1551 loc) · 50.4 KB
Code 55% faster with GitHub Copilot
Raw

```
%sql SELECT DISTINCT Booster_Version, Payload FROM 'SPACEXTBL' WHERE "Landing _Outcome" = "Success (drone ship]
```

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

Out[51]:

Booster_Version	Payload

In [57]:

```
#%sql SELECT DISTINCT Booster_Version, Payload from 'SPACEXTBL' WHERE "Landing_Outcome" = "Success (drone ship.
%sql SELECT DISTINCT Booster_Version, Payload FROM 'SPACEXTBL' WHERE "Landing_Outcome" = "Success (drone ship)'
```

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

Out[57]:

Booster_Version	Payload
F9 FT B1022	JCSAT-14
F9 FT B1026	JCSAT-16
F9 FT B1021.2	SES-10
F9 FT B1031.2	SES-11 / EchoStar 105

Task 7

List the total number of successful and failure mission outcomes

In [58]:

```
%sql SELECT "Mission_Outcome", COUNT("Mission_Outcome") as Total FROM SPACEXTBL GROUP BY "Mission_Outcome";
```

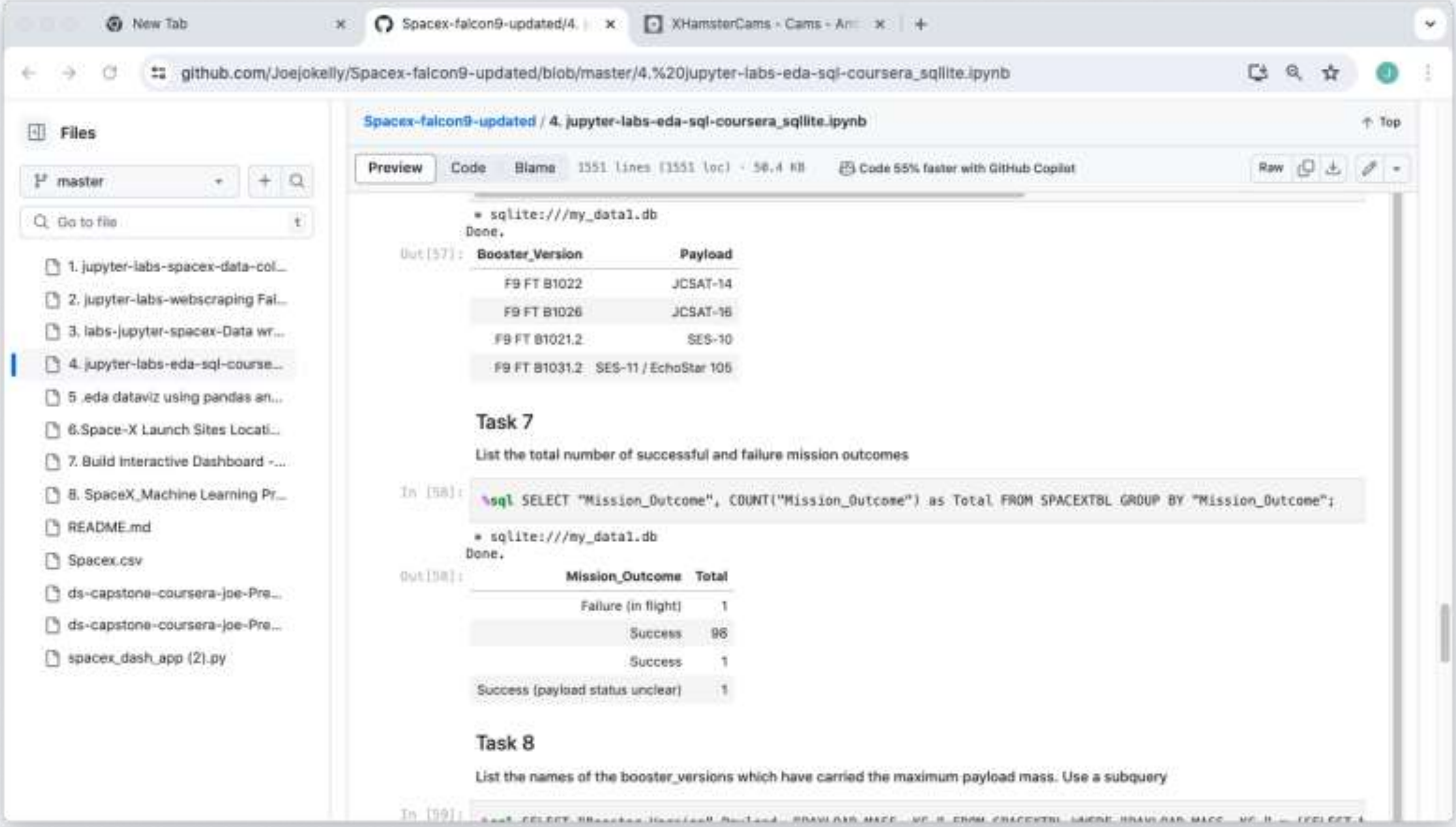
```
* sqlite:///my_data1.db
Done.
```

Out[58]:

Mission_Outcome	Total

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes


New Tab
Spacex-falcon9-updated/4.
XHamsterCams - Cams - Am
github.com/Joejokelly/Spacex-falcon9-updated/blob/master/4.%20jupyter-labs-eda-sql-coursera_sqllite.ipynb
Files
master
Go to file
1. jupyter-labs-spacex-data-col...
2. jupyter-labs-webscraping Fal...
3. labs-jupyter-spacex-Data wr...
4. jupyter-labs-eda-sql-course...
5 .eda dataviz using pandas an...
6.Space-X Launch Sites Locati...
7. Build Interactive Dashboard -...
8. SpaceX_Machine Learning Pr...
README.md
Spacex.csv
ds-capstone-coursera-joe-Pre...
ds-capstone-coursera-joe-Pre...
spacex_dash_app (2).py
Spacex-falcon9-updated / 4. jupyter-labs-eda-sql-coursera_sqllite.ipynb
Top
Preview
Code
Blame
1551 lines (1551 loc) · 50.4 KB
Code 55% faster with GitHub Copilot
Raw

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

Out[57]:

Booster_Version	Payload
F9 FT B1022	JCSAT-14
F9 FT B1026	JCSAT-16
F9 FT B1021.2	SES-10
F9 FT B1031.2	SES-11 / EchoStar 105

Task 7

List the total number of successful and failure mission outcomes

In [58]:

```
%sql SELECT "Mission_Outcome", COUNT("Mission_Outcome") as Total FROM SPACEXTBL GROUP BY "Mission_Outcome";
```

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

Out[58]:

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

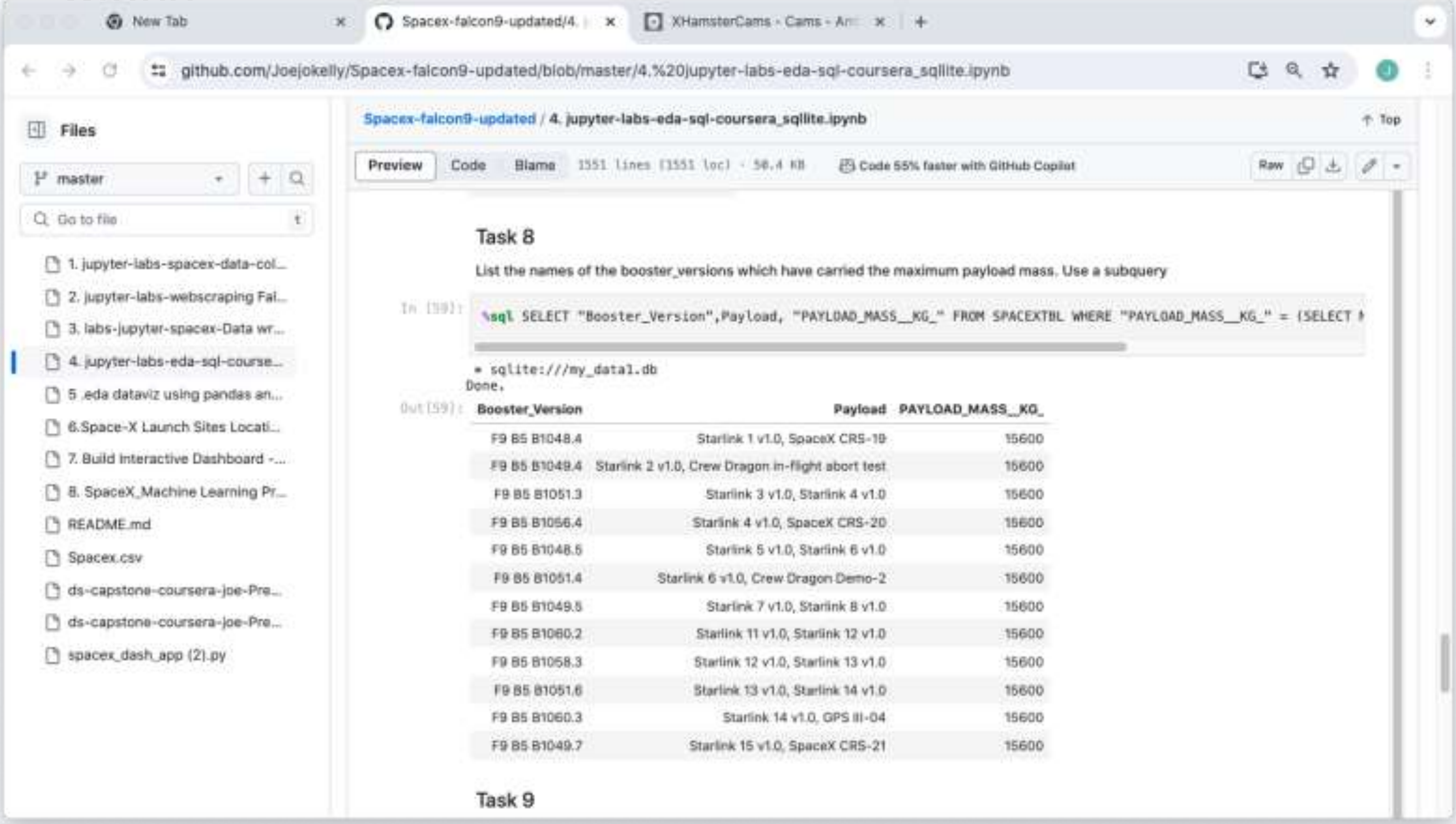
Task 8

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

In [59]:

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass


New Tab
Spacex-falcon9-updated/4.
XHamsterCams - Cams - An
github.com/Joejokelly/Spacex-falcon9-updated/blob/master/4.%20jupyter-labs-eda-sql-coursera_sqllite.ipynb
Files
master
Go to file
1. jupyter-labs-spacex-data-col...
2. jupyter-labs-webscraping Fal...
3. labs-jupyter-spacex-Data wr...
4. jupyter-labs-eda-sql-course...
5 .eda dataviz using pandas an...
6.Space-X Launch Sites Locati...
7. Build Interactive Dashboard -...
8. SpaceX_Machine Learning Pr...
README.md
Spacex.csv
ds-capstone-coursera-joe-Pre...
ds-capstone-coursera-joe-Pre...
spacex_dash_app (2).py
Spacex-falcon9-updated / 4. jupyter-labs-eda-sql-coursera_sqllite.ipynb
Top
Preview
Code
Blame
1551 lines (1551 loc) · 50.4 KB
Code 55% faster with GitHub Copilot
Raw
In [59]:
Out[59]:

Task 8

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

```
%sql SELECT "Booster_Version",Payload, "PAYLOAD_MASS__KG_" FROM SPACEXTBL WHERE "PAYLOAD_MASS__KG_" = (SELECT M
```

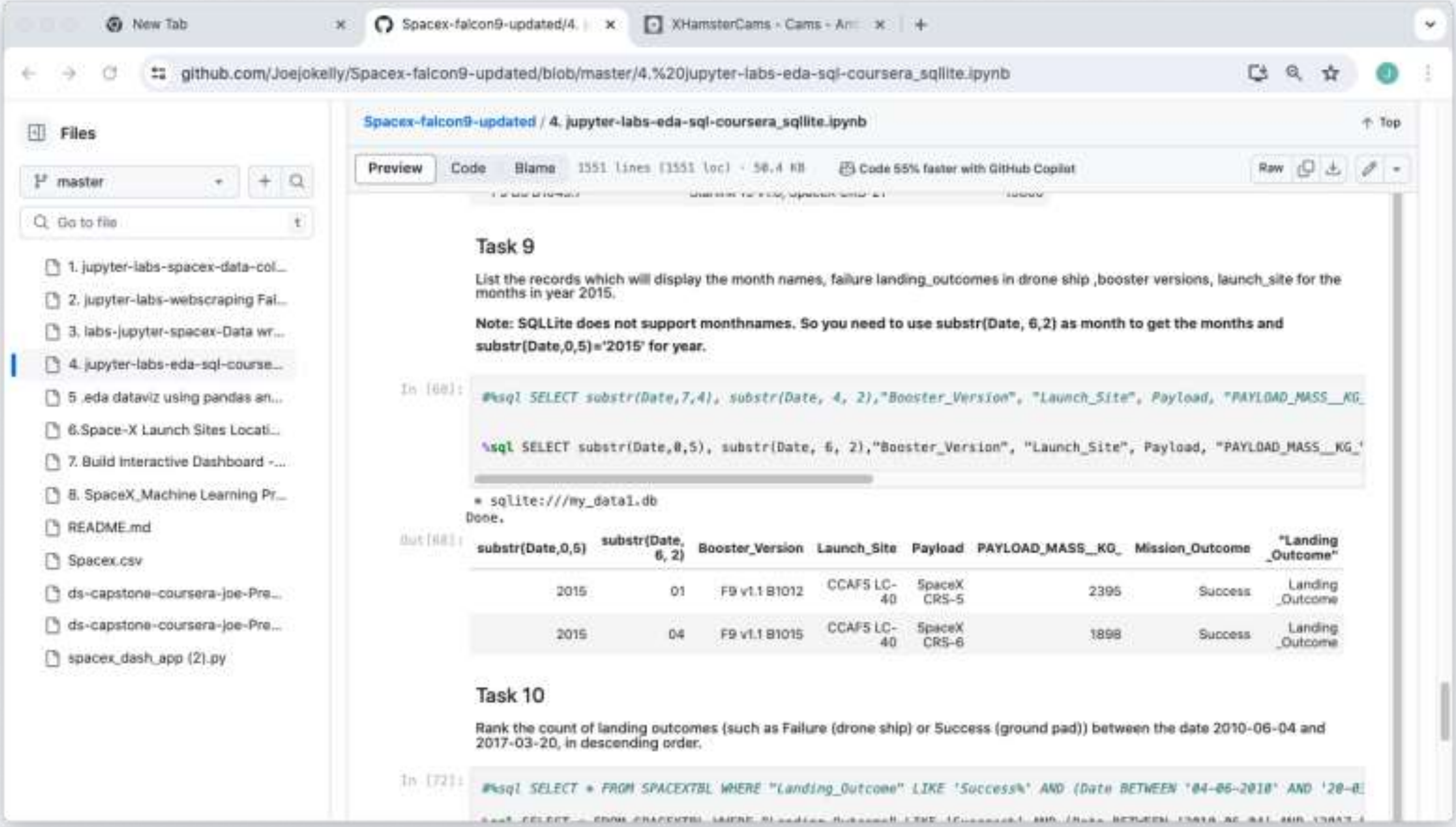
```
 * sqlite:///my_data1.db
Done.
```

Booster_Version	Payload	PAYLOAD_MASS__KG_
F9 B5 B1048.4	Starlink 1 v1.0, SpaceX CRS-19	15600
F9 B5 B1049.4	Starlink 2 v1.0, Crew Dragon in-flight abort test	15600
F9 B5 B1051.3	Starlink 3 v1.0, Starlink 4 v1.0	15600
F9 B5 B1056.4	Starlink 4 v1.0, SpaceX CRS-20	15600
F9 B5 B1048.5	Starlink 5 v1.0, Starlink 6 v1.0	15600
F9 B5 B1051.4	Starlink 6 v1.0, Crew Dragon Demo-2	15600
F9 B5 B1049.5	Starlink 7 v1.0, Starlink 8 v1.0	15600
F9 B5 B1060.2	Starlink 11 v1.0, Starlink 12 v1.0	15600
F9 B5 B1058.3	Starlink 12 v1.0, Starlink 13 v1.0	15600
F9 B5 B1051.6	Starlink 13 v1.0, Starlink 14 v1.0	15600
F9 B5 B1060.3	Starlink 14 v1.0, GPS III-04	15600
F9 B5 B1049.7	Starlink 15 v1.0, SpaceX CRS-21	15600

Task 9

2015 Launch Records

- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015


New Tab
Spacex-falcon9-updated/4.
XHamsterCams - Cams - An
github.com/Joejokelly/Spacex-falcon9-updated/blob/master/4.%20jupyter-labs-eda-sql-coursera_sqllite.ipynb
Files
master
Go to file
1. jupyter-labs-spacex-data-col...
2. jupyter-labs-webscraping Fal...
3. labs-jupyter-spacex-Data wr...
4. jupyter-labs-eda-sql-course...
5 .eda dataviz using pandas an...
6.Space-X Launch Sites Locati...
7. Build Interactive Dashboard -...
8. SpaceX_Machine Learning Pr...
README.md
Spacex.csv
ds-capstone-coursera-joe-Pre...
ds-capstone-coursera-joe-Pre...
spacex_dash_app (2).py
Spacex-falcon9-updated / 4. jupyter-labs-eda-sql-coursera_sqllite.ipynb
Top
Preview
Code
Blame
1551 lines (1551 loc) · 50.4 KB
Code 55% faster with GitHub Copilot
Raw

Task 9

List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.

Note: SQLLite does not support monthnames. So you need to use substr(Date, 6,2) as month to get the months and substr(Date,0,5)='2015' for year.

In [68]:

```
#%sql SELECT substr(Date,7,4), substr(Date, 4, 2),"Booster_Version", "Launch_Site", Payload, "PAYLOAD_MASS__KG_

%sql SELECT substr(Date,0,5), substr(Date, 6, 2),"Booster_Version", "Launch_Site", Payload, "PAYLOAD_MASS__KG_'
```

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

Out[68]:

substr(Date,0,5)	substr(Date, 6, 2)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Mission_Outcome	"Landing _Outcome"
2015	01	F9 v1.1 B1012	CCAFS LC-40	SpaceX CRS-5	2395	Success	Landing _Outcome
2015	04	F9 v1.1 B1015	CCAFS LC-40	SpaceX CRS-6	1898	Success	Landing _Outcome

Task 10

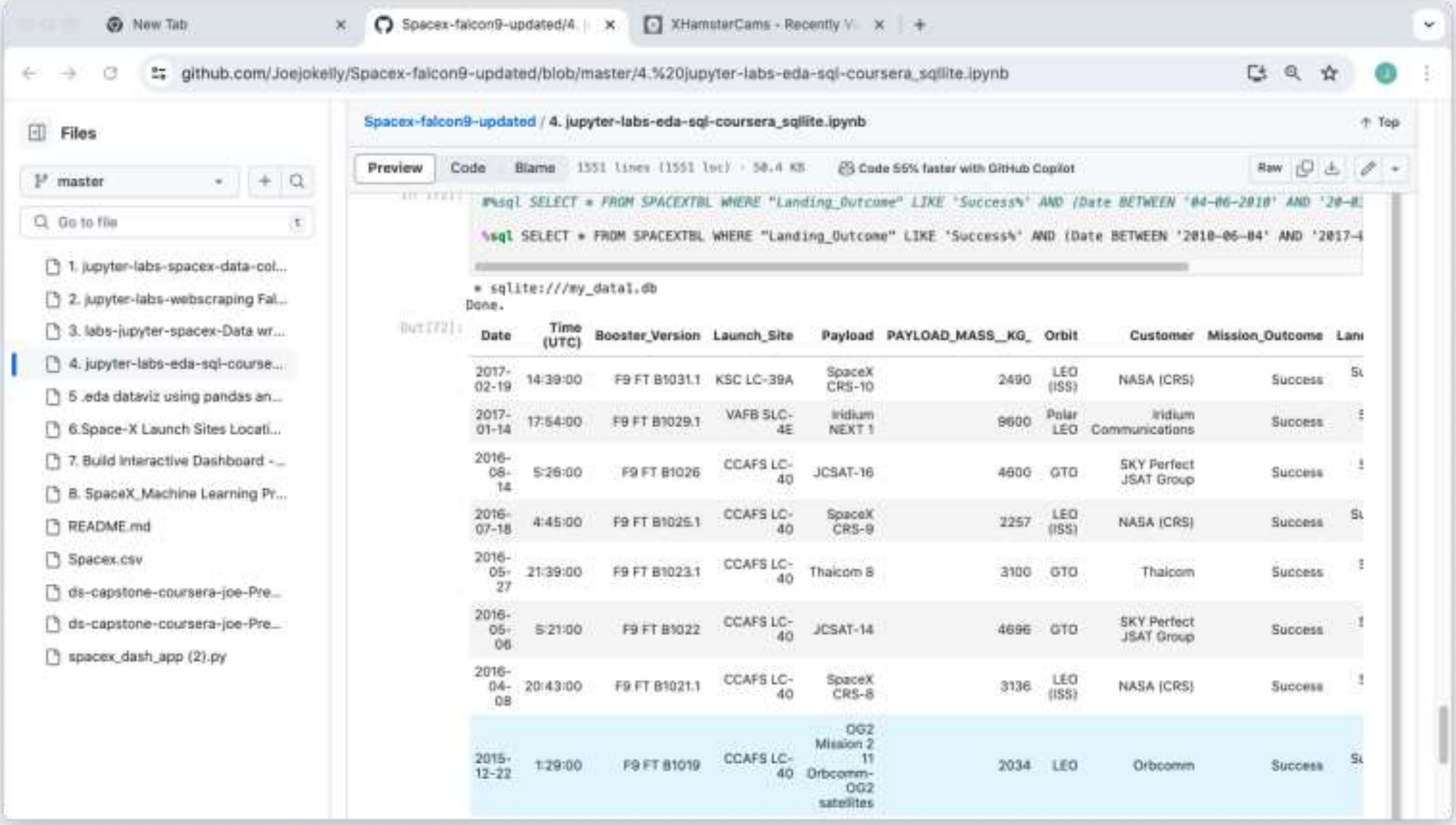
Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order.

In [72]:

```
#%sql SELECT * FROM SPACEXTBL WHERE "Landing_Outcome" LIKE 'Success%' AND (Date BETWEEN '04-06-2010' AND '20-0
```

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


New Tab
Spacex-falcon9-updated/4.
XHamsterCams - Recently V
github.com/Joejokelly/Spacex-falcon9-updated/blob/master/4.%20jupyter-labs-eda-sql-coursera_sqllite.ipynb
Files
master
Go to file
1. jupyter-labs-spacex-data-col...
2. jupyter-labs-webscraping Fal...
3. labs-jupyter-spacex-Data wr...
4. jupyter-labs-eda-sql-course...
5 .eda dataviz using pandas an...
6.Space-X Launch Sites Locati...
7. Build Interactive Dashboard -...
8. SpaceX_Machine Learning Pr...
README.md
Spacex.csv
ds-capstone-coursera-joe-Pre...
ds-capstone-coursera-joe-Pre...
spacex_dash_app (2).py
Spacex-falcon9-updated / 4. jupyter-labs-eda-sql-coursera_sqllite.ipynb
Top
Preview
Code
Blame
1551 lines (1551 loc) · 50.4 KB
Code 55% faster with GitHub Copilot
Raw
Out[72]:

```
#%sql SELECT * FROM SPACEXTBL WHERE "Landing_Outcome" LIKE 'Success%' AND (Date BETWEEN '04-06-2010' AND '20-0
%sql SELECT * FROM SPACEXTBL WHERE "Landing_Outcome" LIKE 'Success%' AND (Date BETWEEN '2010-06-04' AND '2017-
```

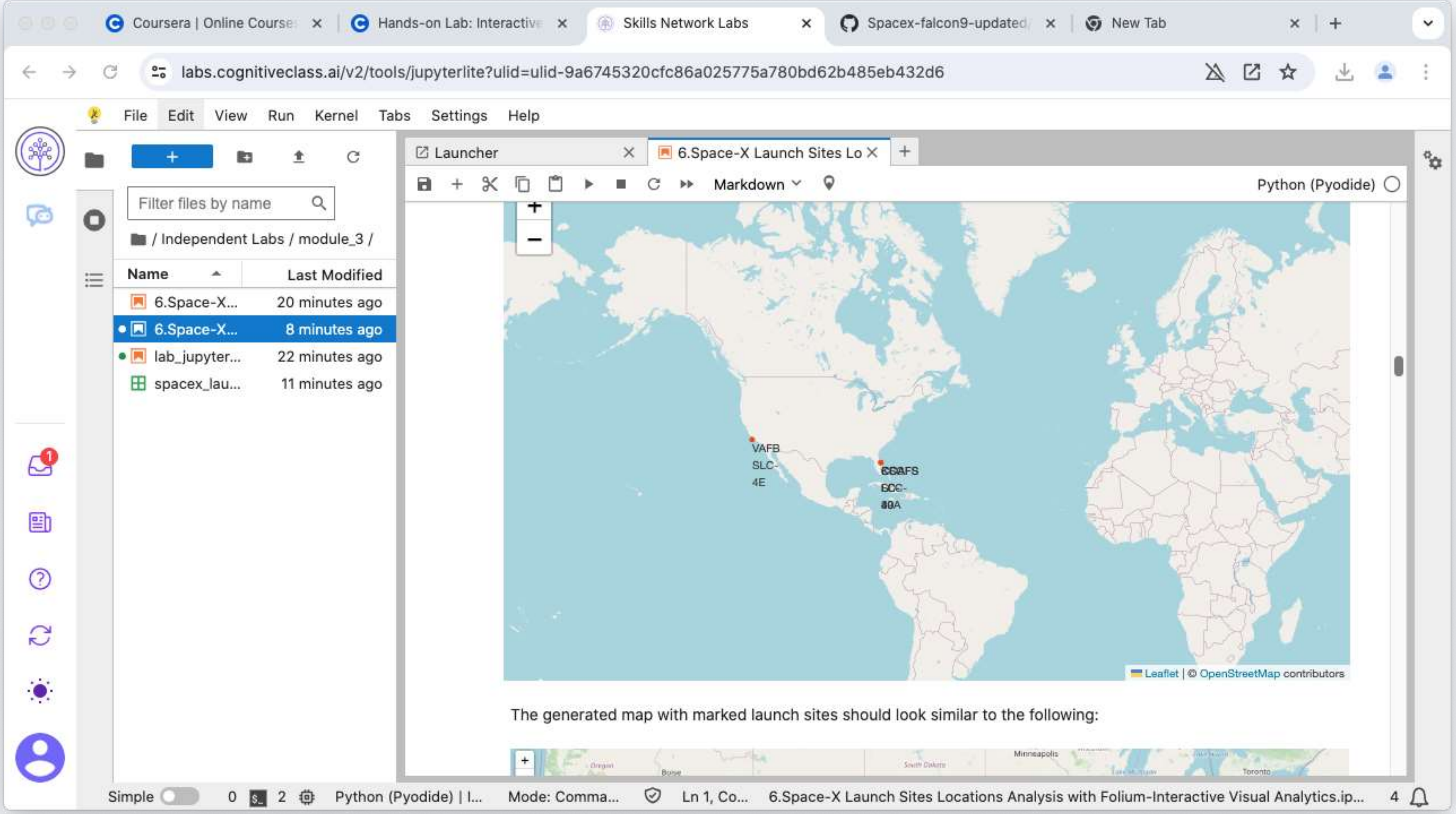
```
 * sqlite:///my_data1.db
Done.
```

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Lan
2017-02-19	14:39:00	F9 FT B1031.1	KSC LC-39A	SpaceX CRS-10	2490	LEO (ISS)	NASA (CRS)	Success	Su
2017-01-14	17:54:00	F9 FT B1029.1	VAFB SLC-4E	Iridium NEXT 1	9600	Polar LEO	Iridium Communications	Success	S
2016-08-14	5:26:00	F9 FT B1026	CCAFS LC-40	JCSAT-16	4600	GTO	SKY Perfect JSAT Group	Success	S
2016-07-18	4:45:00	F9 FT B1025.1	CCAFS LC-40	SpaceX CRS-9	2257	LEO (ISS)	NASA (CRS)	Success	Su
2016-05-27	21:39:00	F9 FT B1023.1	CCAFS LC-40	Thaicom 8	3100	GTO	Thaicom	Success	S
2016-05-06	5:21:00	F9 FT B1022	CCAFS LC-40	JCSAT-14	4696	GTO	SKY Perfect JSAT Group	Success	S
2016-04-08	20:43:00	F9 FT B1021.1	CCAFS LC-40	SpaceX CRS-8	3136	LEO (ISS)	NASA (CRS)	Success	S
2015-12-22	1:29:00	F9 FT B1019	CCAFS LC-40	OG2 Mission 2 11 Orbcomm-OG2 satellites	2034	LEO	Orbcomm	Success	Su

Section 3

Launch Sites Proximities Analysis

launch site on the global map

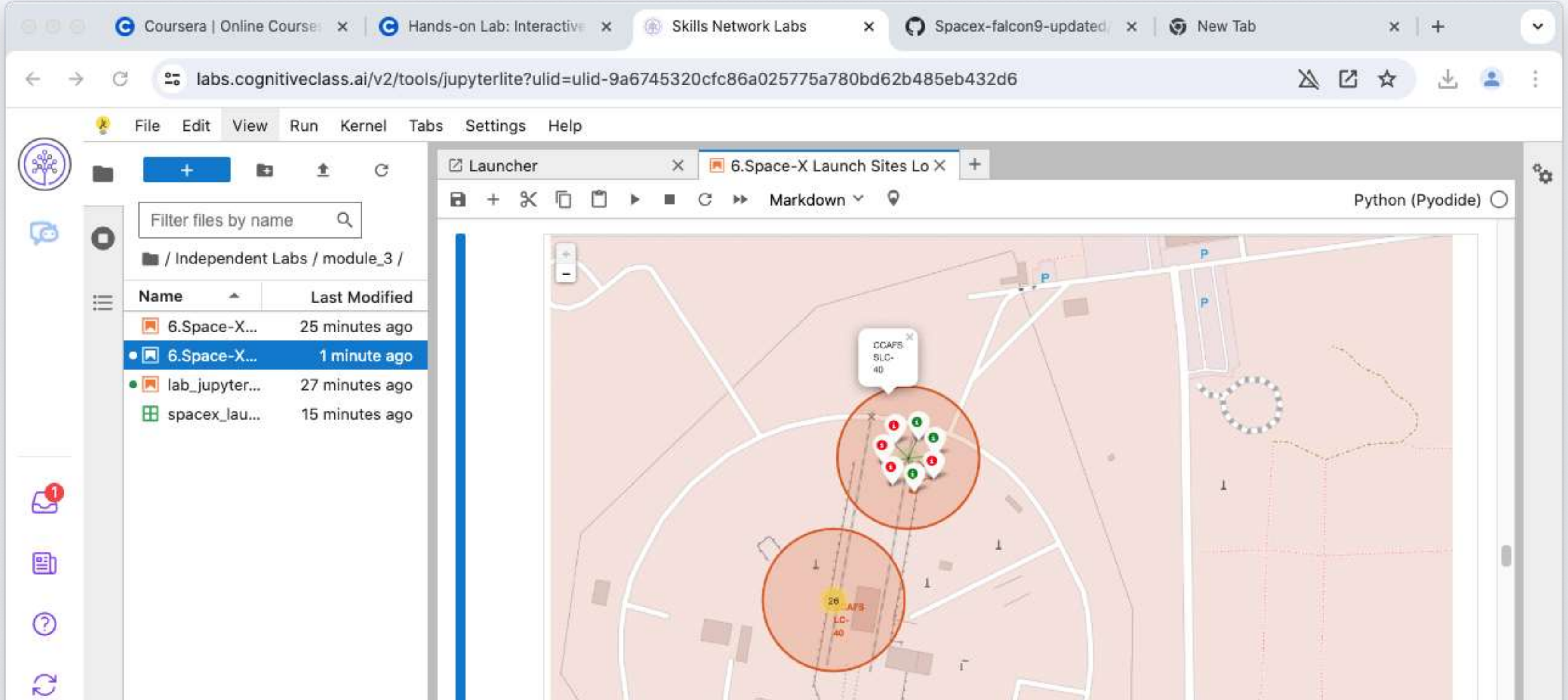

Coursera | Online Course
Hands-on Lab: Interactive
Skills Network Labs
Spacex-falcon9-updated
New Tab
labs.cognitiveclass.ai/v2/tools/jupyterlite?uid=ulid-9a6745320cfc86a025775a780bd62b485eb432d6
File
Edit
View
Run
Kernel
Tabs
Settings
Help
Filter files by name
/ Independent Labs / module_3 /
Launcher
6.Space-X Launch Sites Lo
Markdown
Python (Pyodide)
VAFB
SLC-
4E
CCAFS
SLC-
40
Leaflet | © OpenStreetMap contributors
Simple
Python (Pyodide) | I...
Mode: Comma...
Ln 1, Co...
6.Space-X Launch Sites Locations Analysis with Folium-Interactive Visual Analytics.ip...

Name	Last Modified
6.Space-X...	20 minutes ago
6.Space-X...	8 minutes ago
lab_jupyter...	22 minutes ago
spacex_lau...	11 minutes ago

The generated map with marked launch sites should look similar to the following:

color-labeled launch outcome

- Explore the folium map and make a proper screenshot to show the color-labeled launch outcomes on the map



labs.cognitiveclass.ai/v2/tools/jupyterlite?ulid=ulid-9a6745320cfc86a025775a780bd62b485eb432d6

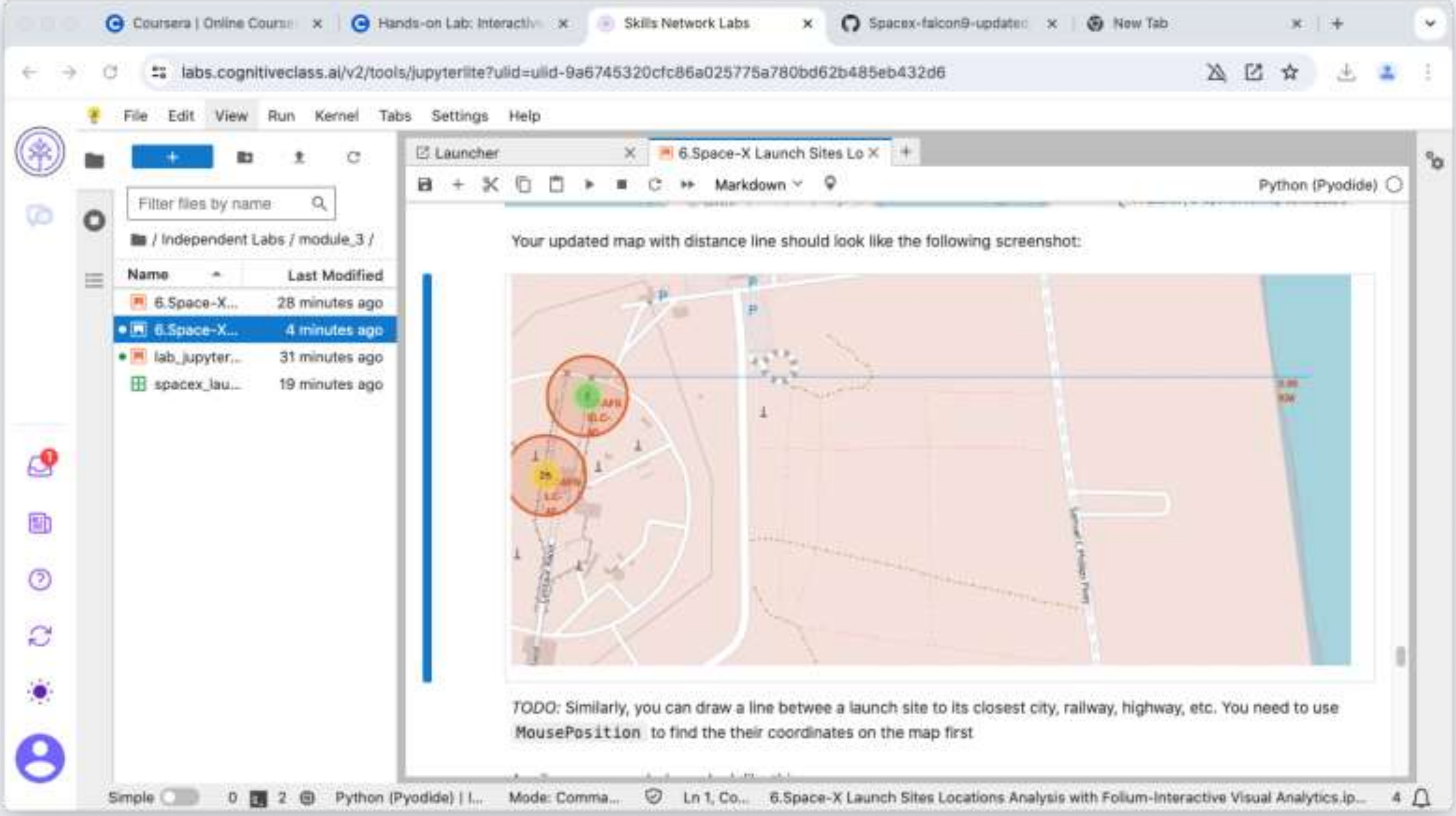
/ Independent Labs / module_3 /

Name	Last Modified
6.Space-X...	25 minutes ago
6.Space-X...	1 minute ago
lab_jupyter...	27 minutes ago
spacex_lau...	15 minutes ago

Coursera | Online Course
Hands-on Lab: Interactive
Skills Network Labs
Spacex-falcon9-updated
New Tab
File
Edit
View
Run
Kernel
Tabs
Settings
Help
Filter files by name
Launcher
6.Space-X Launch Sites Lo
Markdown
Python (Pyodide)
CCAFS SLC-40
26
CCAFS LC-40
P

Proximity of Launch site to railway, highway

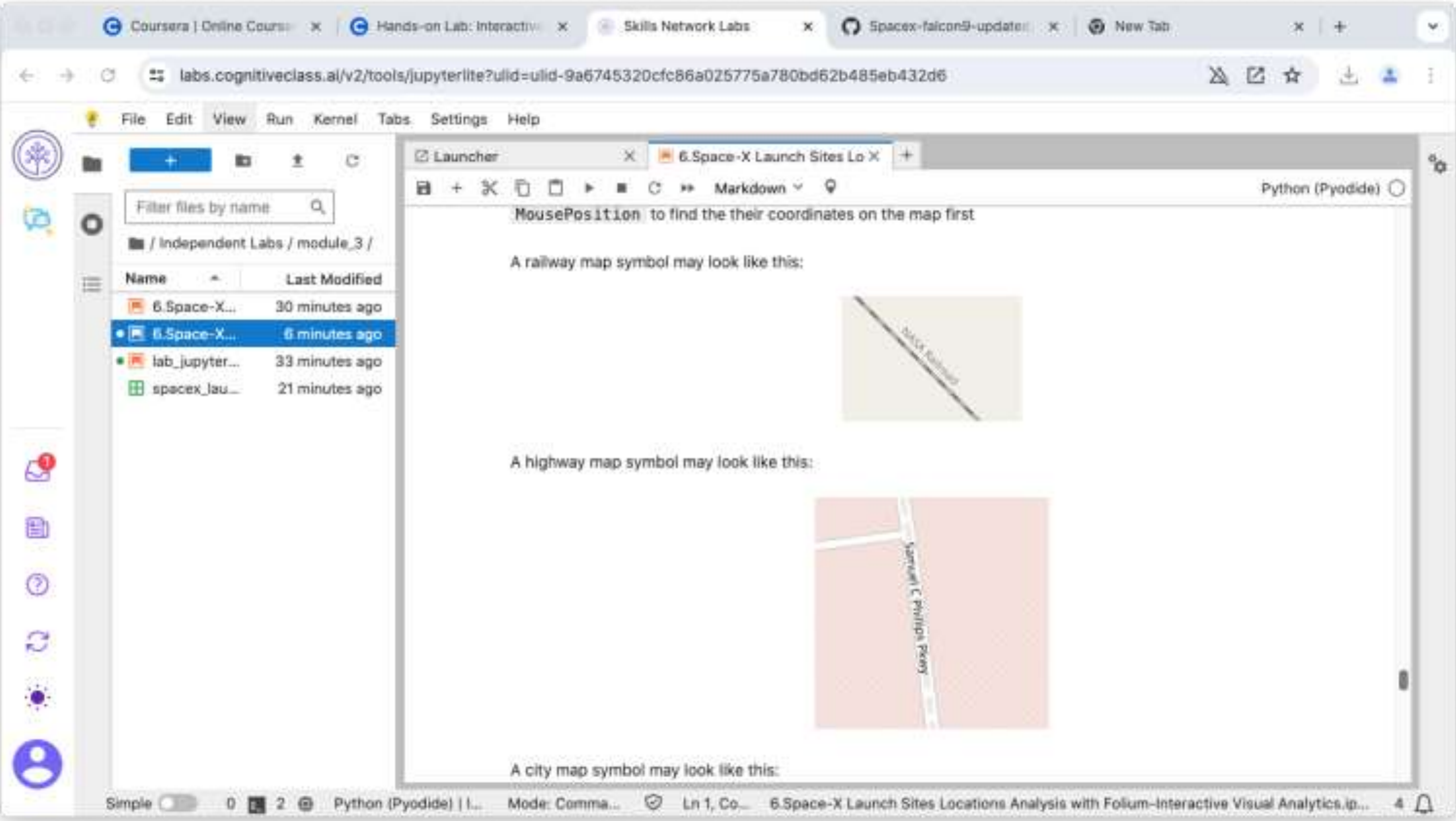
- Replace <Folium map screenshot 3> title with an appropriate title


Coursera | Online Course
Hands-on Lab: Interactive
Skills Network Labs
Spacex-falcon9-updated
New Tab
labs.cognitiveclass.ai/v2/tools/jupyterlite?ulid=ulid-9a6745320cfc86a025775a780bd62b485eb432d6
File Edit View Run Kernel Tabs Settings Help
Filter files by name
/ Independent Labs / module_3 /
Name
Last Modified
6.Space-X...
28 minutes ago
6.Space-X...
4 minutes ago
lab_jupyter...
31 minutes ago
spacex_lau...
19 minutes ago
Launcher
6.Space-X Launch Sites Lo
Markdown
Python (Pyodide)
Simple
Python (Pyodide) | I...
Mode: Comma...
Ln 1, Co...
6.Space-X Launch Sites Locations Analysis with Folium-Interactive Visual Analytics.ip...

Your updated map with distance line should look like the following screenshot:

TODO: Similarly, you can draw a line betwee a launch site to its closest city, railway, highway, etc. You need to use `MousePosition` to find the their coordinates on the map first

Proximity contd...

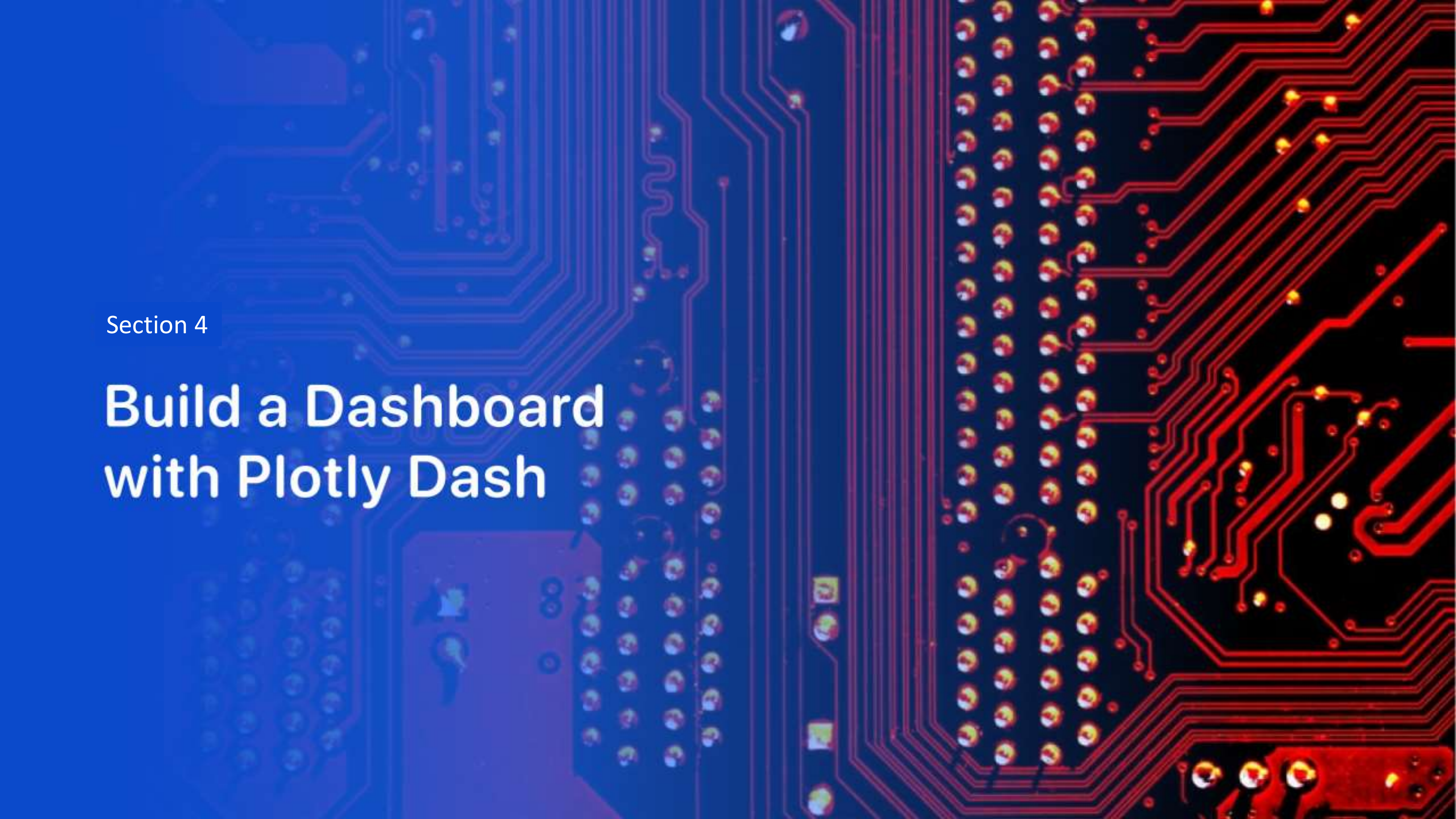

labs.cognitiveclass.ai/v2/tools/jupyterlite?ulid=ulid-9a6745320cfc86a025775a780bd62b485eb432d6
File Edit View Run Kernel Tabs Settings Help
Launcher
6.Space-X Launch Sites Lo
/ Independent Labs / module_3 /
Name
Last Modified
6.Space-X... 30 minutes ago
6.Space-X... 6 minutes ago
lab_jupyter... 33 minutes ago
spacex_lau... 21 minutes ago
Markdown
Python (Pyodide)
NASA Railroad
Samuel C Phillips Pkwy

`MousePosition` to find the their coordinates on the map first

A railway map symbol may look like this:

A highway map symbol may look like this:

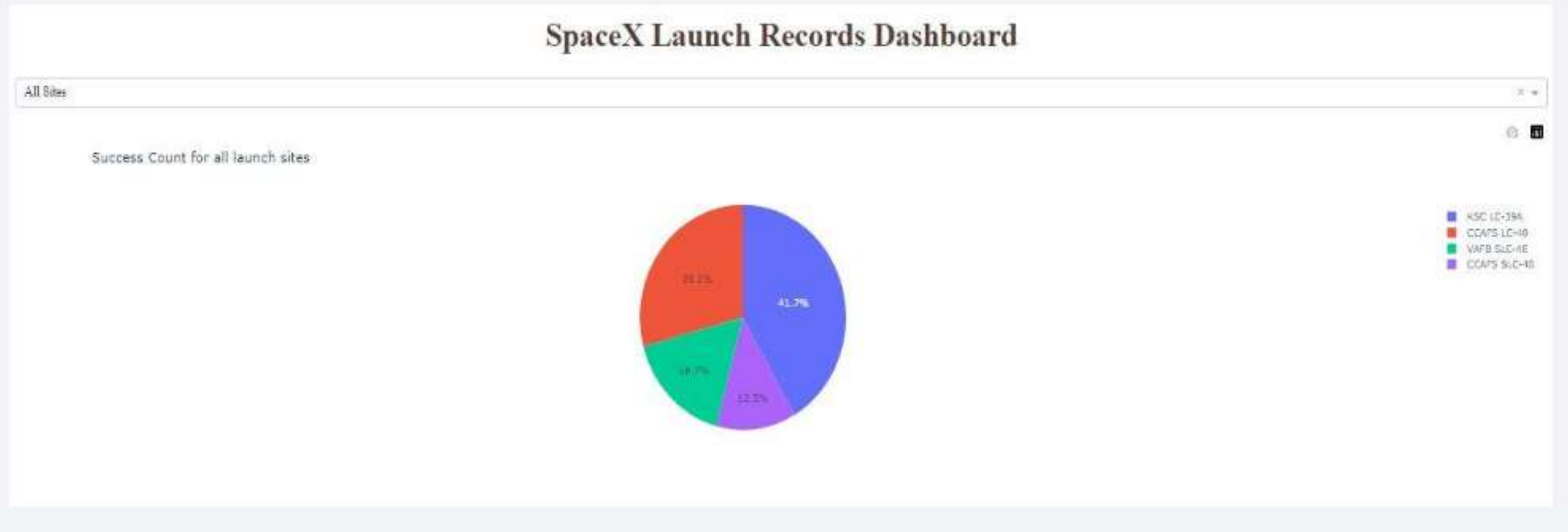
A city map symbol may look like this:



Section 4

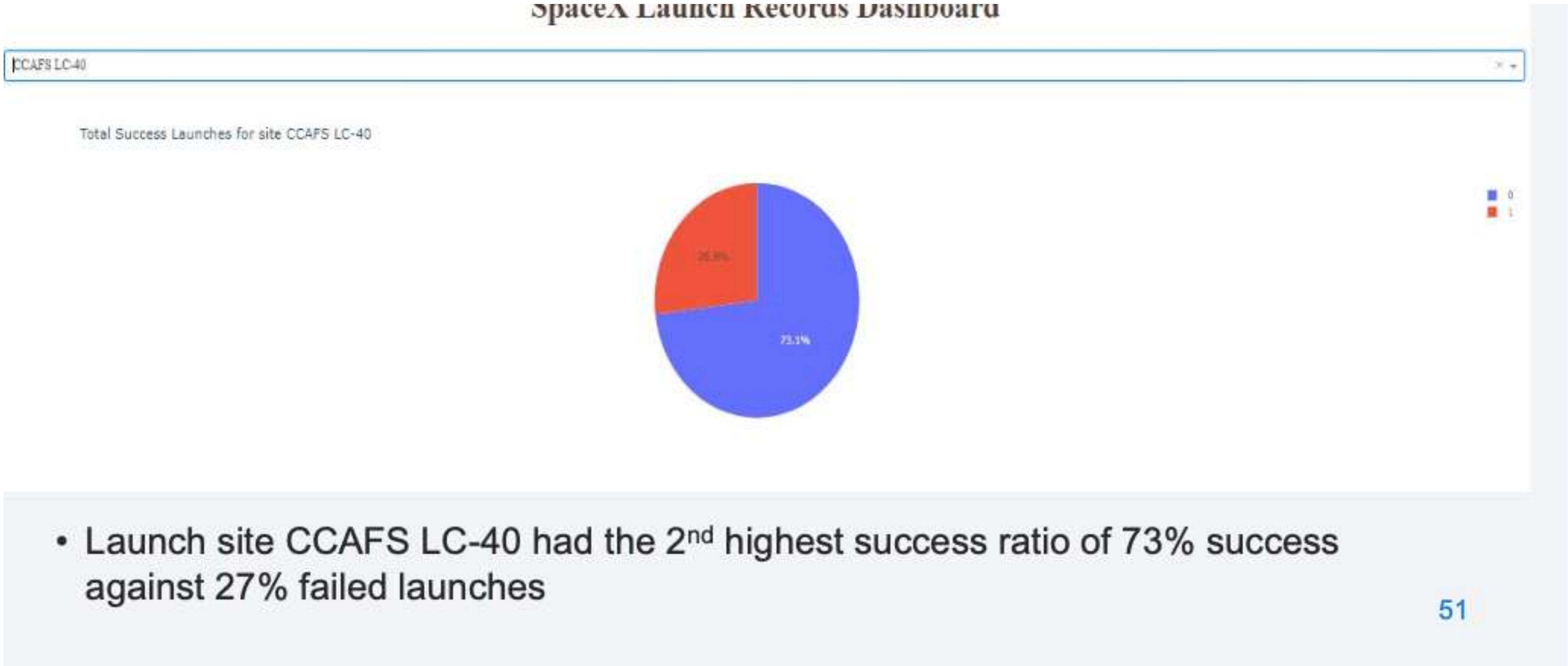
Build a Dashboard with Plotly Dash

Pie chart for launch Success


SpaceX Launch Records Dashboard
All Sites
Success Count for all launch sites
41.7%
29.2%
16.7%
12.5%
KSC LC-39A
CCAFS LC-40
VAFB SLC-4E
CCAFS SLC-40

Launch site KSC LC-39A highest success rate 42%
CCAFS LC-40 at 29%, CCAFS LC-40 at 29%

Lauch site success rate



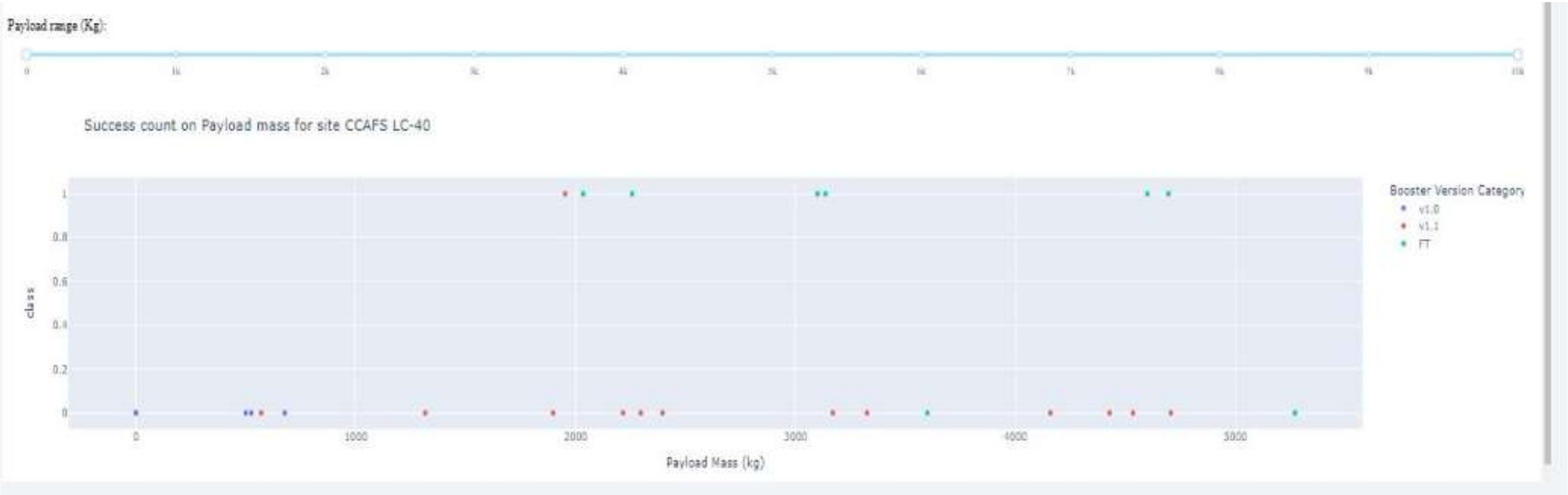
SpaceX Launch Records Dashboard

CCAFS LC-40

Total Success Launches for site CCAFS LC-40

- Launch site CCAFS LC-40 had the 2nd highest success ratio of 73% success against 27% failed launches

Payload vs launch site


Payload range (Kg):
0
1k
2k
3k
4k
5k
6k
7k
8k
9k
10k
Success count on Payload mass for site CCAFS LC-40
class
1
0.8
0.6
0.4
0.2
0
0
1000
2000
3000
4000
5000
Payload Mass (kg)
Booster Version Category
v1.0
v1.1
FT

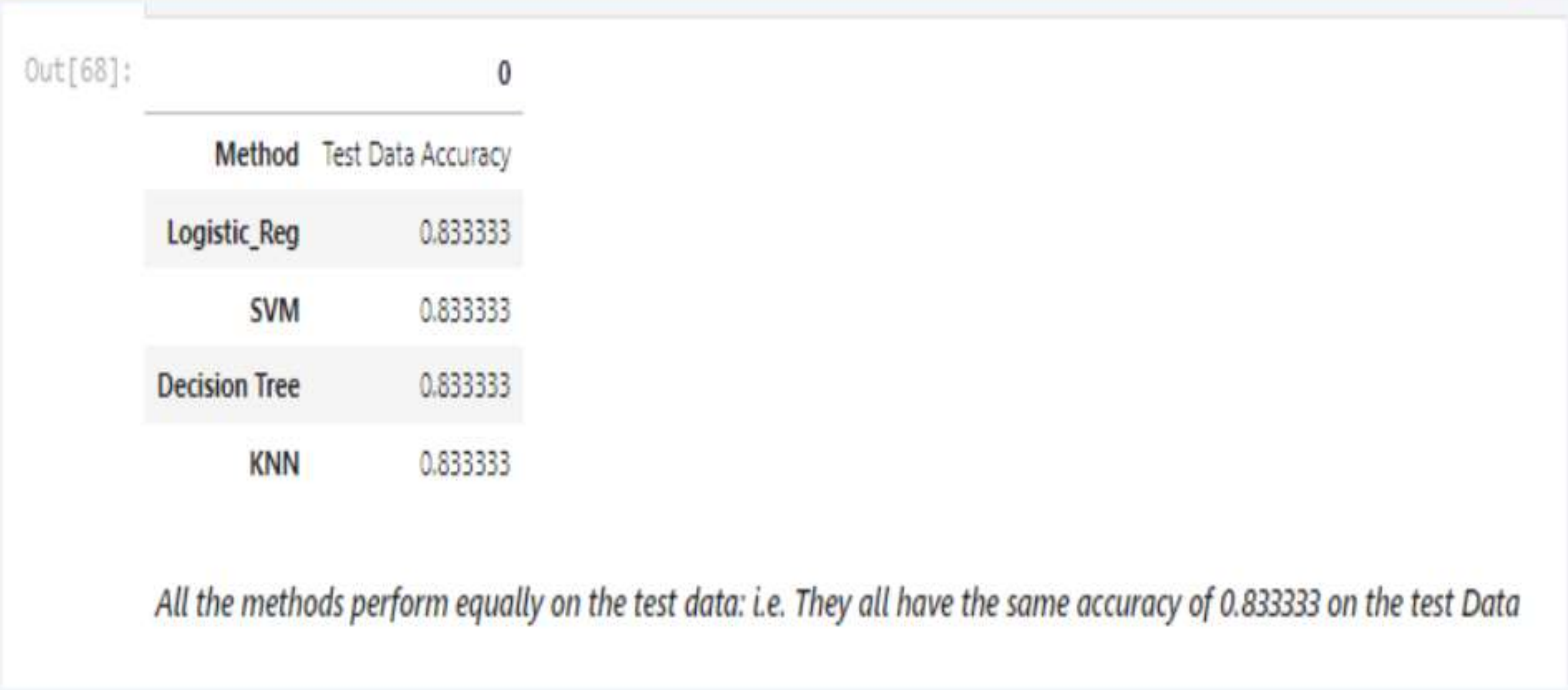
For launch ccfa-lc 40, the booster version has the largest success rage payload mass > 2000

Section 5

Predictive Analysis (Classification)

Classification Accuracy

- All models perform the same, accuracy of 0.83333 on test Data



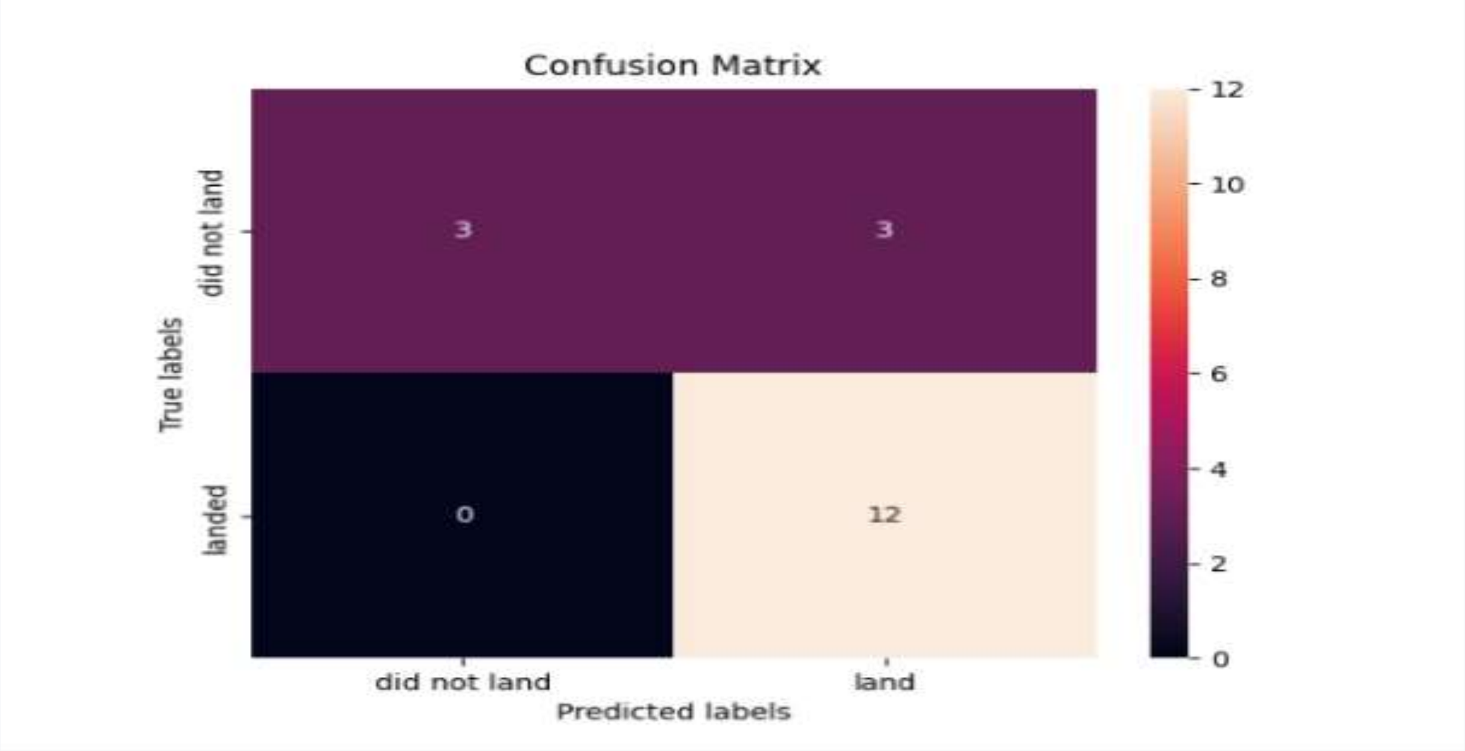
Out[68]:

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

All the methods perform equally on the test data: i.e. They all have the same accuracy of 0.833333 on the test Data

Confusion Matrix

- All models have the same confusion matrix



Conclusions

- Success rates are different for different launch sites.

  CCAFS LC-40 has success rate of 60%

  KSC LC-39A and VAFB SLC 4E has a success rate of 77%

- The flight number increases so does the success rate,
- Payload vs Lauch site Scatter plot, we find there are no rockets lauched s
- Orbits ES-L1, GEO, HEO & SSO have the highest success rates at 100%, with SO orbit having the lowest success rate at ~50%. Orbit SO has 0% success rate.
- LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit

Conclusion contd..

- Heavy payloads the successful landing is higher for Polar, LEO and ISS. GTO has both positive and negative landings
- Finally the success rate since 2013 kept increasing till 2020

Thank you!