IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Hanna Kiyko

09.03.2923

Outline

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

Executive Summary

- I collect information from various sources, process it and study what parameters and how the success of landing depends
- I train and test various machine learning models to obtain a prediction of the success of landing, and therefore the cost of the project.

Introduction

- The commercial space age is here, companies are making space travel affordable for everyone. One reason SpaceX can do this is the rocket launches are relatively inexpensive. 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.

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

- we will be working with SpaceX launch data that is gathered from an API, specifically the SpaceX REST API.
- you will be using the Python BeautifulSoup package to web scrape some HTML tables that contain valuable Falcon 9 launch records.

Data Collection – SpaceX API

- The SpaceX REST API endpoints, or URL, starts with api.spacexdata.com/v4/. We will be working with the endpoint api.spacexdata.com/v4/launches/past.

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

response = requests.get(spacex_url)
```

https://github.com/KiykoHanna/CourseraProgect/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

Data Collection - Scraping

- I will be using the Python BeautifulSoup package to web scrape some HTML tables that contain valuable Falcon 9 launch records.

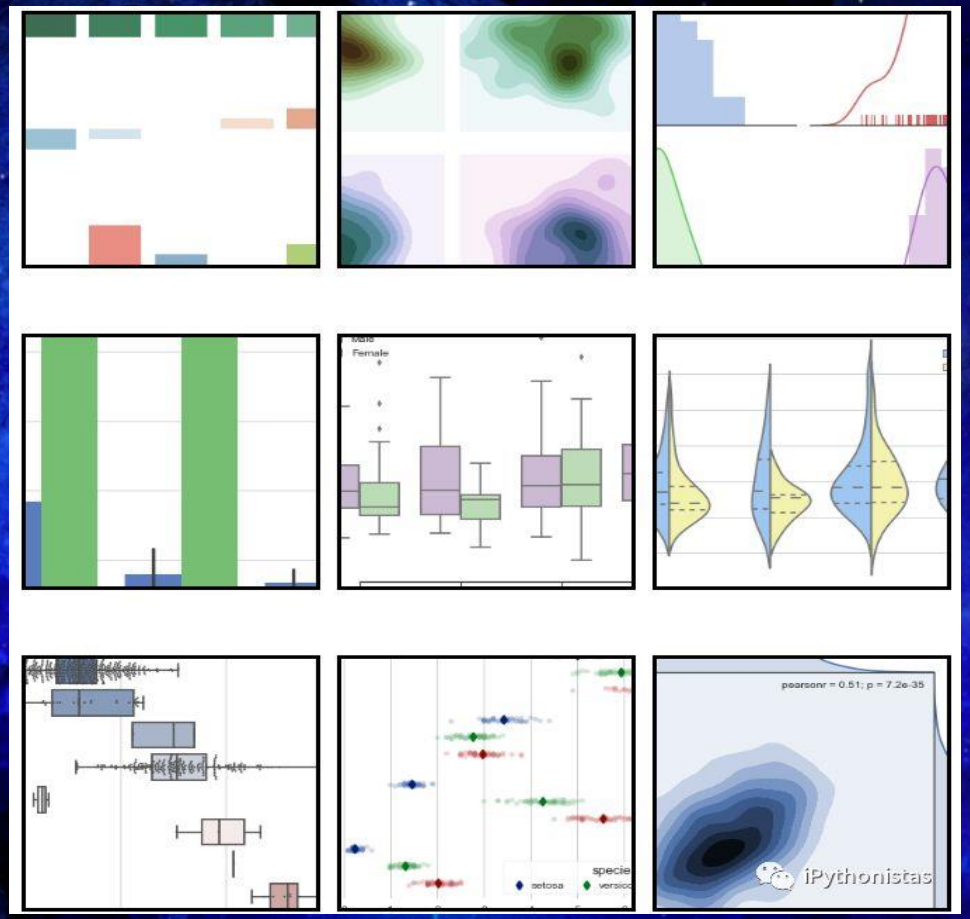
- static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"
- resp = requests.get(static_url).text
- soup = BeautifulSoup(resp,"html5lib")
- html_tables = soup.find_all('table')

https://github.com/KiykoHanna/CourseraProgect/blob/main/jupyter-labs-webscraping.ipynb

Data Wrangling

- I perform some Exploratory Data Analysis (EDA) to find some patterns in the data and determine what would be the label for training supervised models.

https://github.com/KiykoHanna/CourseraProgect/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb



EDA with Data Visualization

We use the Python modul: Seaborn.

- scatterplot
- barplot
- catplot
- lineplot

https://github.com/KiykoHanna/CourseraProgect/blob/main/jupyter-labs-eda-dataviz.ipynb

```
conn = sqlite3.connect("D:\study\python\py-
n\coursera\DS_LAST_PROJECT\my_data1.db")#
open a database connection
cur = conn.cursor()
cur.execute('SELECT name FROM sqlite_master
WHERE type="table"')
print(cur.fetchall())
Q= """
SELECT * FROM SPACEXTBL
"""
cur.execute(Q)
pd.read_sql_query(Q,conn)
```

EDA with SQL

- I perform some Exploratory Data Analysis using a database.
- I can see that the data can be used to automatically determine if the Falcon 9's second stage will land.

ub.com/KiykoHanna/CourseraProgect/blob/main/jupyter-labs-eda-sql-coursera.ipynb

Build an Interactive Map with Folium

https://github.com/KiykoHanna/CourseraProgect/blob/main/lab_jupyter_launch_site_location.ipynb

After I plot distance lines to the proximities, I can answer the following questions easily:

* Are launch sites in close proximity to railways?
* Are launch sites in close proximity to highways?
* Are launch sites in close proximity to coastline?
* Do launch sites keep certain distance away from cities?

Build a Dashboard with Plotly Dash

- This dashboard application contains input components such as a dropdown list and a range slider to interact with a pie chart and a scatter point chart.
- I can use it to find more insights from the SpaceX dataset more easily than with static graphs.

https://github.com/KiykoHanna/CourseraProgect/blob/main/spacex_dash_app.py

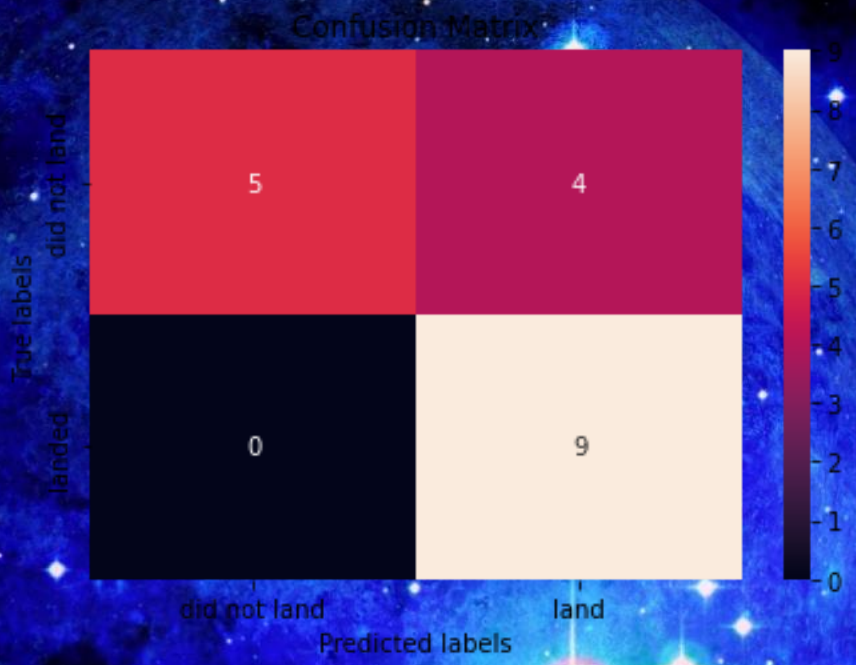
Predictive Analysis (Classification)

I build a machine learning pipeline to predict if the first stage of the Falcon 9 lands successfully.

My project include:

- Preprocessing, to standardize our data,
- Train_test_split, to split our data into training and testing data,
- Train the model and perform Grid Search, to find the hyperparameters that allow a given algorithm to perform best.
- Using the best hyperparameter values, I determine the model with the best accuracy using the training data.
- I test Logistic Regression, Support Vector machines, Decision Tree Classifier, and K-nearest neighbors.
- Finally, I output the confusion matrix.

Results



I Find best Hyperparameter for LogisticRegression, SVM, Classification Trees and KNN:

- LogisticRegression:'C': 1, 'penalty': 'l2', 'solver': 'lbfgs'

accuracy : 0.9444444444444444

- SVM: kernel='sigmoid', 'C': 0.001, 'gamma': 0.001

accuracy : 0.7222222222222222

- Classification Trees: 'criterion': 'gini', 'max_depth': 16, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 10, 'splitter': 'random'

accuracy : 0.8333333333333334

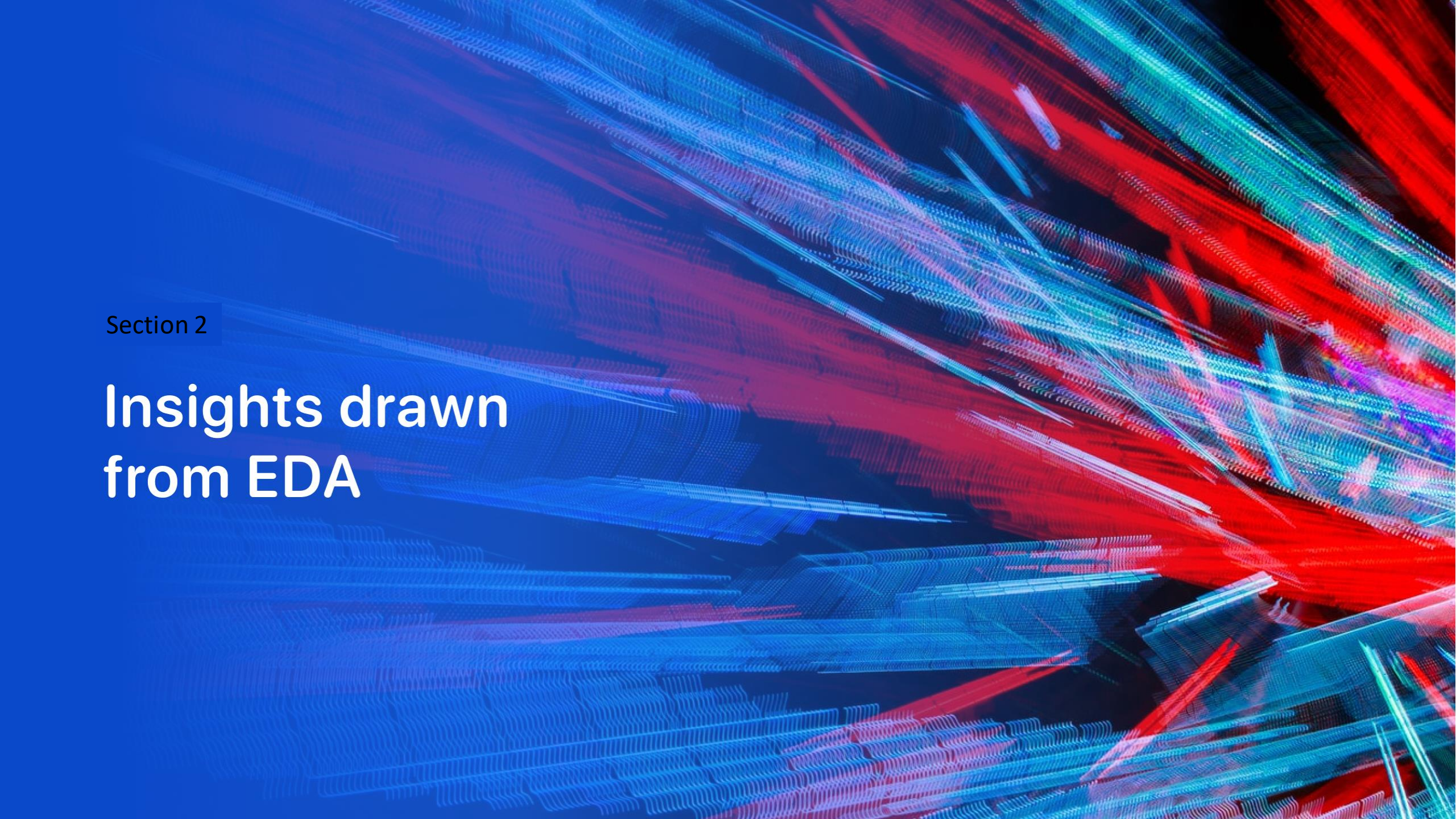
- KNN: 'algorithm': 'auto', 'n_neighbors': 7, 'p': 1

accuracy : 0.611111111111112

https://colab.research.google.com/gist/KiykoHanna/d16b5f4c8ef630c84a8a3c2e24a2e30e/ibm_ds0321en_skillsnetwork_labs_module_4_spacex_machine_learning_prediction_part_5_jupyterlite.ipynb

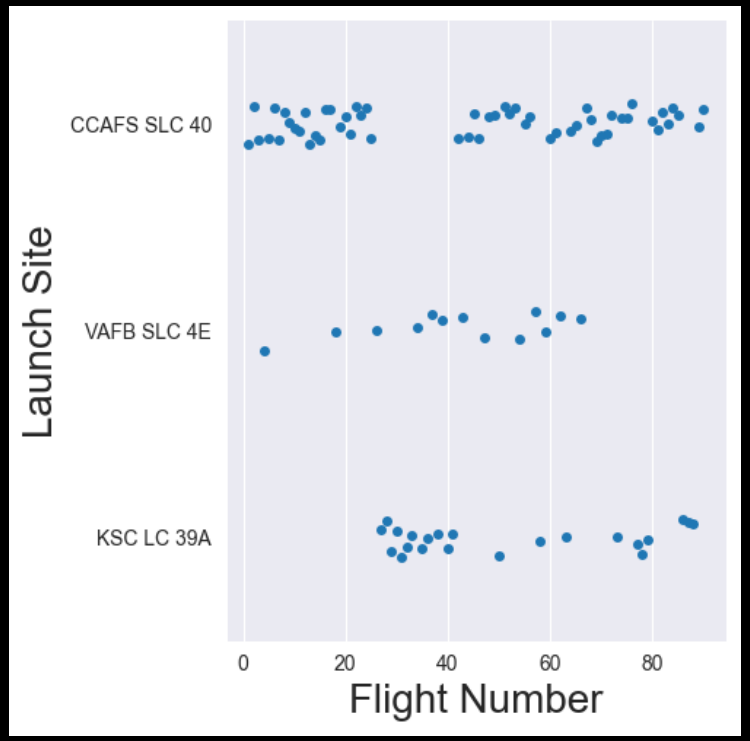
Section 2

Insights drawn from EDA



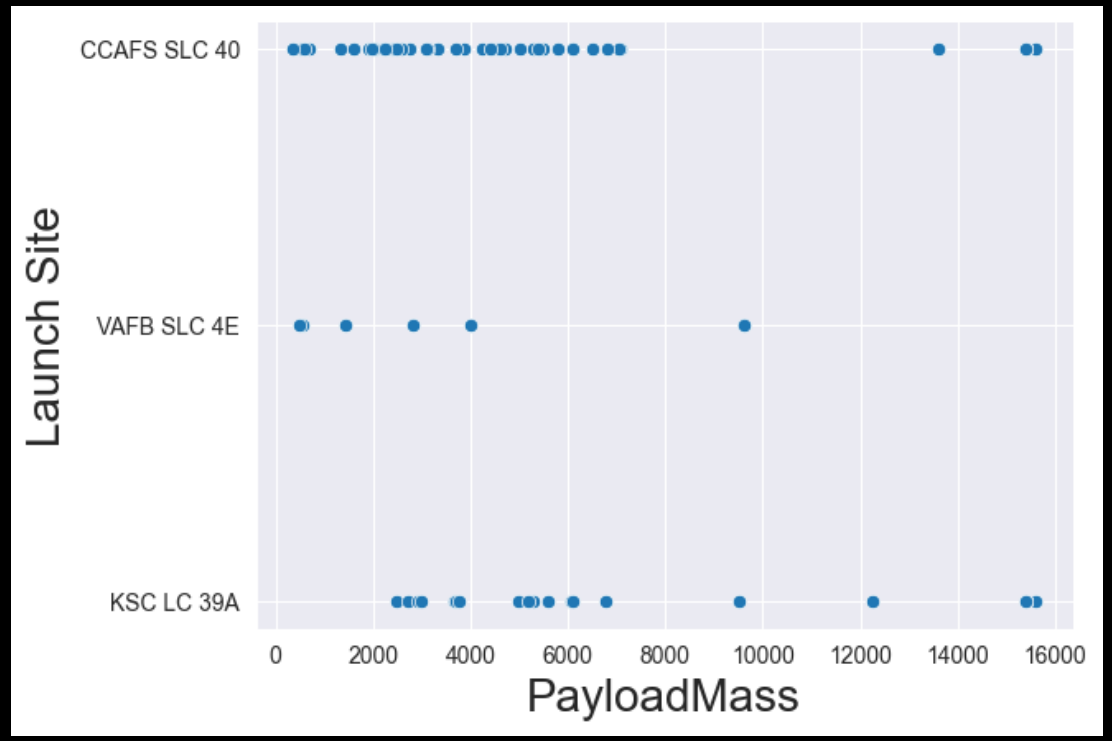
Flight Number vs. Launch Site

- sns.catplot(y="LaunchSite", x="FlightNumber", data=df,)
- plt.xlabel("Flight Number",fontsize=20)
- plt.ylabel("Launch Site",fontsize=20)
- plt.show()



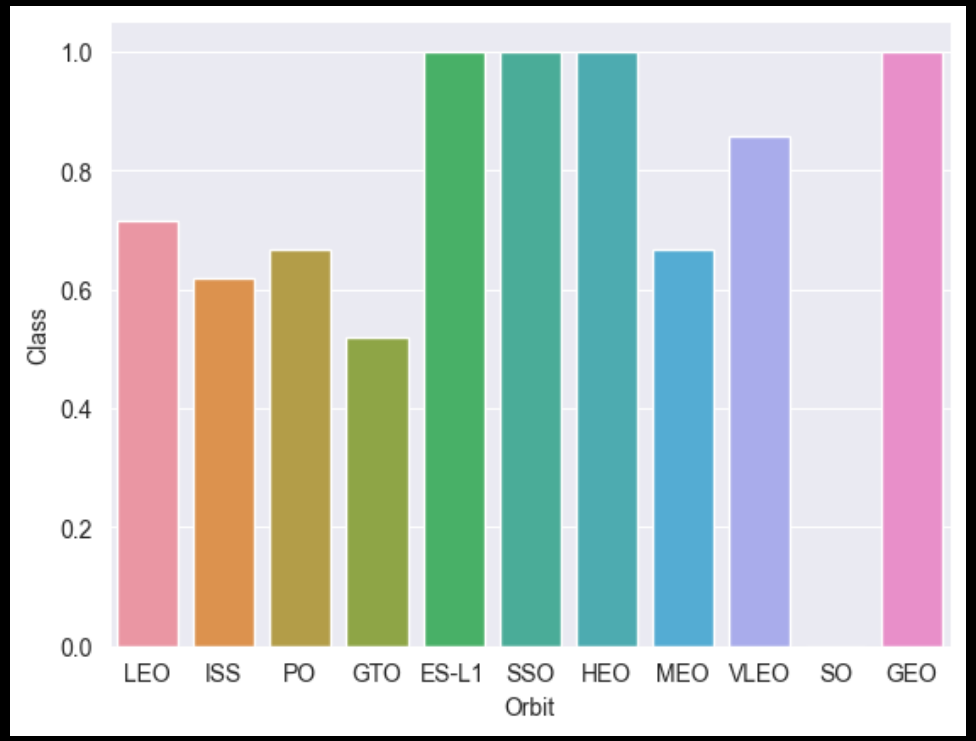
Payload vs. Launch Site

- sns.scatterplot(df, x="PayloadMass", y="LaunchSite")
- plt.xlabel("PayloadMass",fontsize=20)
- plt.ylabel("Launch Site",fontsize=20)
- plt.show()



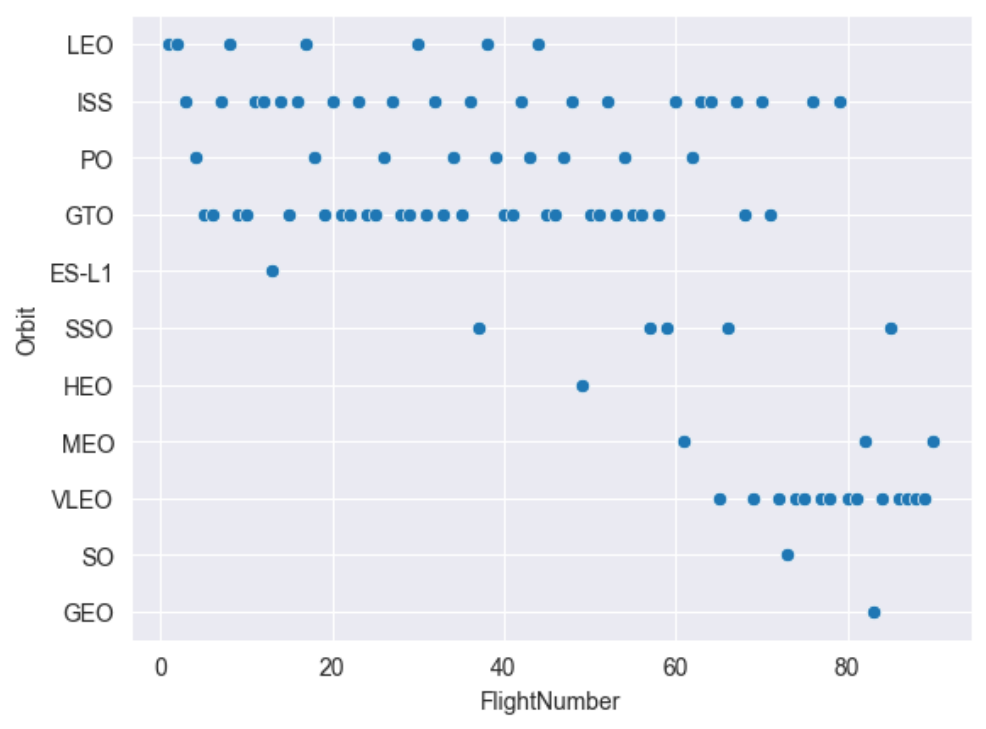
Success Rate vs. Orbit Type

- df1 = df.groupby(pd.Grouper(key="Orbit")).mean()
- sns.barplot(df1, x = df1.index, y = 'Class')



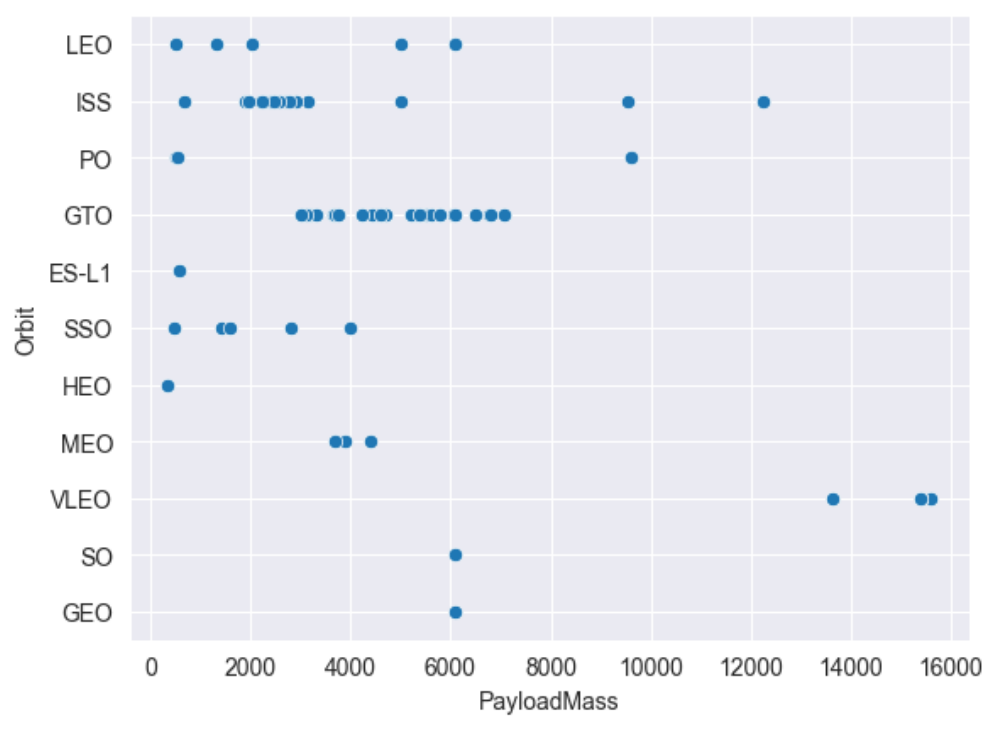
Flight Number vs. Orbit Type

- sns.scatterplot(df, x= 'FlightNumber', y= 'Orbit')



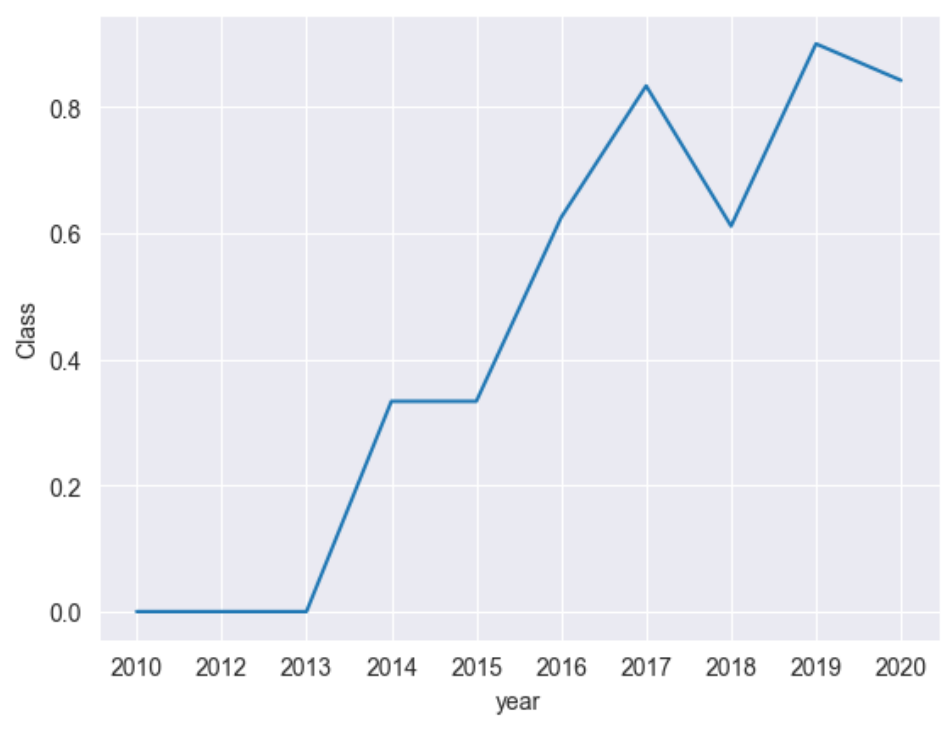
Payload vs. Orbit Type

- sns.scatterplot(df, x= 'PayloadMass', y= 'Orbit')



Launch Success Yearly Trend

- df2 = df.groupby(pd.Grouper(key="year")).mean()
- sns.lineplot(df2, x = df2.index, y = 'Class')



All Launch Site Names

- Q= """
- SELECT DISTINCT(Launch_Site) FROM SPACEXTBL
- """
- cur.execute(Q)
- pd.read_sql_query(Q,conn)

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

Launch Site Names Begin with 'CCA'

- Q= """
- SELECT * FROM SPACEXTBL
- WHERE Launch_Site LIKE 'CCA%'
- LIMIT 5
- """
- cur.execute(Q)
- pd.read_sql_query(Q,conn)

	Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome	
0	04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)	
1	08-12-2010	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of...	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)	
2	22-05-2012	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt	
3	08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt	
4	01-03-2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt	

Total Payload Mass

```
Q= """
SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL
WHERE Customer = 'NASA (COTS)'
"""
cur.execute(Q)
pd.read_sql_query(Q,conn)
```

	SUM(PAYLOAD_MASS__KG_)
0	525

Average Payload Mass by F9 v1.1

```
Q= """
SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTBL
WHERE Booster_Version = 'F9 v1.1'
"""
cur.execute(Q)
pd.read_sql_query(Q,conn)
```

	AVG(PAYLOAD_MASS__KG_)
0	2928.4

First Successful Ground Landing Date

```
Q= """
SELECT Date FROM SPACEXTBL
WHERE Mission_Outcome = 'Success'
LIMIT 1
"""
cur.execute(Q)
pd.read_sql_query(Q,conn)
```

Date

0 04-06-2010

Successful Drone Ship Landing with Payload between 4000 and 6000

```
Q= """
SELECT Booster_Version FROM SPACEXTBL
WHERE PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000

"""
cur.execute(Q)
pd.read_sql_query(Q,conn)
```

	Booster_Version
0	F9 v1.1
1	F9 v1.1 B1011
2	F9 v1.1 B1014
3	F9 v1.1 B1016
4	F9 FT B1020
5	F9 FT B1022
6	F9 FT B1026
7	F9 FT B1030
8	F9 FT B1021.2
9	F9 FT B1032.1
10	F9 B4 B1040.1
11	F9 FT B1031.2
12	F9 B4 B1043.1
13	F9 FT B1032.2
14	F9 B4 B1040.2
15	F9 B5 B1046.2
16	F9 B5 B1047.2
17	F9 B5 B1046.3
18	F9 B5B1054
19	F9 B5 B1048.3
20	F9 B5 B1051.2
21	F9 B5B1060.1
22	F9 B5 B1058.2
23	F9 B5B1062.1

Total Number of Successful and Failure Mission Outcomes

```
Q= """
SELECT Mission_Outcome, COUNT(*)  FROM SPACEXTBL
GROUP BY Mission_Outcome
"""
cur.execute(Q)
pd.read_sql_query(Q,conn)
```

MISSION_OUTCOME		COUNT(*)
0	Failure (in flight)	1
1	Success	98
2	Success	1
3	Success (payload status unclear)	1

Boosters Carried Maximum Payload

- Q= """
- SELECT Booster_Version FROM SPACEXTBL
- WHERE PAYLOAD_MASS__KG_ IN (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL)
- """
- cur.execute(Q)
- pd.read_sql_query(Q,conn)

	Booster_Version
0	F9 B5 B1048.4
1	F9 B5 B1049.4
2	F9 B5 B1051.3
3	F9 B5 B1056.4
4	F9 B5 B1048.5
5	F9 B5 B1051.4
6	F9 B5 B1049.5
7	F9 B5 B1060.2
8	F9 B5 B1058.3
9	F9 B5 B1051.6
10	F9 B5 B1060.3
11	F9 B5 B1049.7

2015 Launch Records

- Q= """
- SELECT substr(Date,4,2), `Landing _Outcome`, Booster_Version, Launch_Site FROM SPACEXTBL
- WHERE substr(Date,7,4) = '2015' AND `Landing _Outcome` = 'Failure (drone ship)'
- """
- cur.execute(Q)
- pd.read_sql_query(Q,conn)

	substr(Date,4,2)	Landing _Outcome	Booster_Version	Launch_Site
0	01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
1	04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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

```
Q= """
SELECT * FROM SPACEXTBL
WHERE `Landing _Outcome` LIKE 'Success%'
"""
cur.execute(Q)
df1 = pd.read_sql_query(Q,conn)
df1['Date'] = df1['Date'].astype("datetime64[ns]")
df4 = df1[df1['Date'] < '20-03-2017']
df4.sort_values('Date')
```

- explanation here

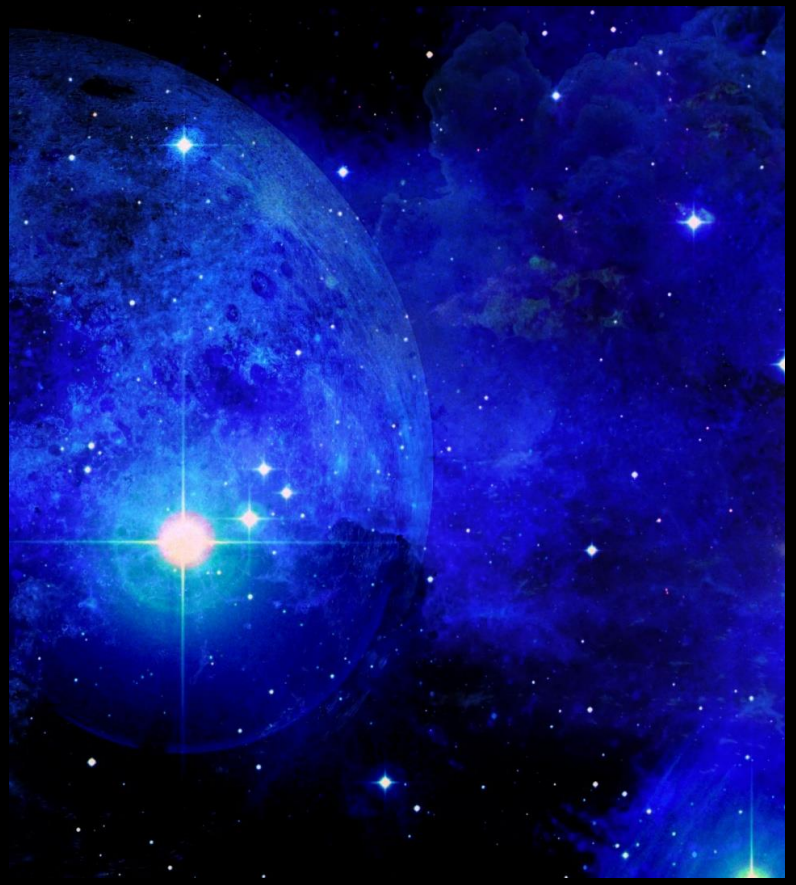
	Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing _Outcome	
0	2015-12-22	01:29:00	F9 FT B1019	CCAFS LC-40	OG2 Mission 2 11 Orbcomm-OG2 satellites	2034	LEO	Orbcomm	Success	Success (ground pad)
3	2016-05-27	21:39:00	F9 FT B1023.1	CCAFS LC-40	Thaicom 8	3100	GTO	Thaicom	Success	Success (drone ship)
2	2016-06-05	05:21:00	F9 FT B1022	CCAFS LC-40	JCSAT-14	4696	GTO	SKY Perfect JSAT Group	Success	Success (drone ship)
4	2016-07-18	04:45:00	F9 FT B1025.1	CCAFS LC-40	SpaceX CRS-9	2257	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
1	2016-08-04	20:43:00	F9 FT B1021.1	CCAFS LC-40	SpaceX CRS-8	3136	LEO (ISS)	NASA (CRS)	Success	Success (drone ship)
5	2016-08-14	05:26:00	F9 FT B1026	CCAFS LC-40	JCSAT-16	4600	GTO	SKY Perfect JSAT Group	Success	Success (drone ship)
9	2017-01-05	11:15:00	F9 FT B1032.1	KSC LC-39A	NROL-76	5300	LEO	NRO	Success	Success (ground pad)
6	2017-01-14	17:54:00	F9 FT B1029.1	VAFB SLC-4E	Iridium NEXT 1	9600	Polar LEO	Iridium Communications	Success	Success (drone ship)
7	2017-02-19	14:39:00	F9 FT B1031.1	KSC LC-39A	SpaceX CRS-10	2490	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
10	2017-03-06	21:07:00	F9 FT B1035.1	KSC LC-39A	SpaceX CRS-11	2708	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)

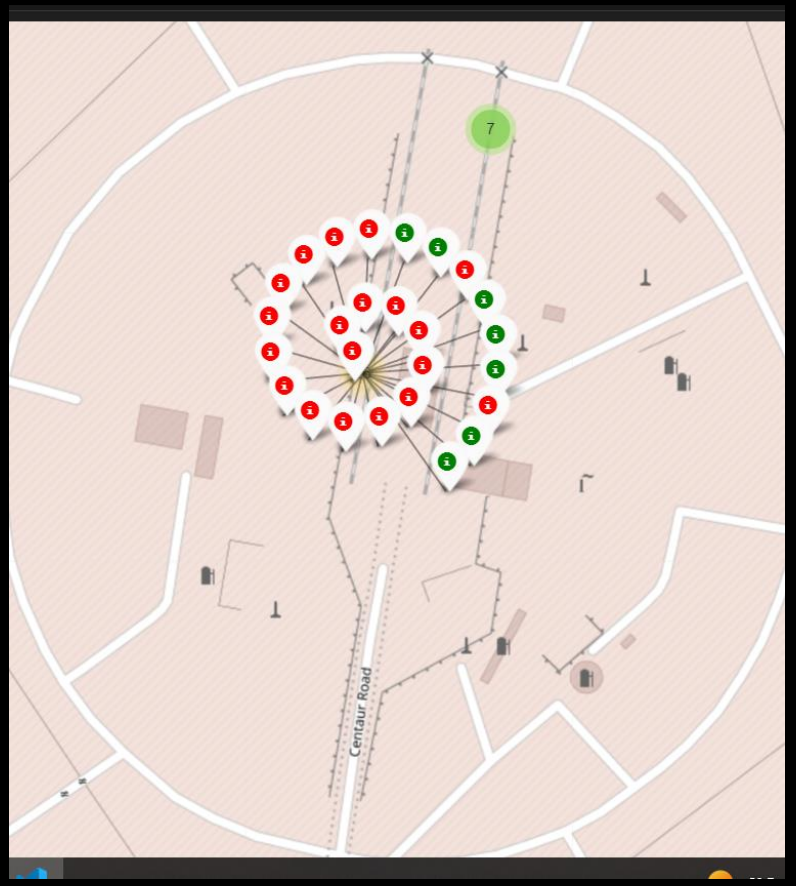
Section 3

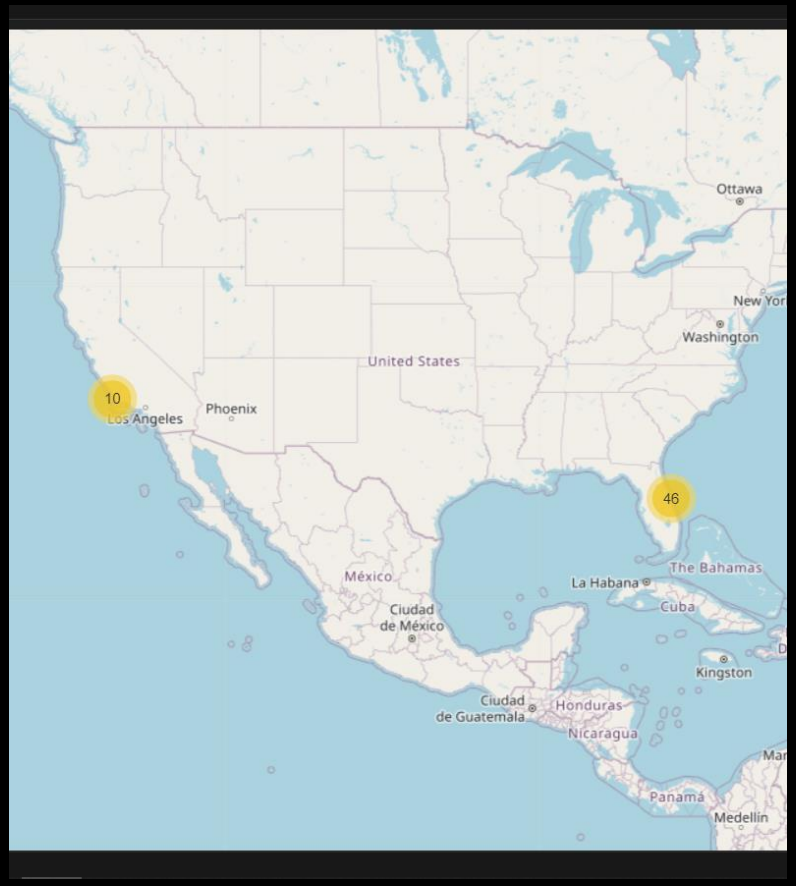
Launch Sites Proximities Analysis


Code
Markdown
Run All
Clear All Outputs
Restart
Variables
Outline
Python 3.10.7
All launch sites' location markers on a global map
Boise
Salt Lake City
Denver
Kansas City
Chicago
Milwaukee
Toronto
Hamilton
Detroit
Boston
New York
Philadelphia
Washington
Richmond
Raleigh
Charlotte
Memphis
Tulsa
Sacramento
San Jose
Fresno
Las Vegas
Los Angeles
VAFB SLC 4E
Phoenix
Tucson
Tijuana
Mexicali
Ensenada
Dallas
Atlanta
Ciudad Juárez
Austin
Houston
Baton Rouge
Jacksonville
San Antonio
NASA JSC
CCAFS LC 40
Nuevo Laredo
Reynosa
Monterrey
Torreón
Saltillo
Culiacán
La Paz
Durango
México
Miami
Nassau
The Bahamas
La Habana
Santa Clara
Cuba
Camaguey
Turks and Caicos Islands
Mérida
Cancun
Cayman Islands
Santiago de Cuba
Republica Dominicana
Jamaica
Belize
Aguascalientes
San Luis Potosí
León
Ciudad de México
Morelia
Veracruz
Puebla
Acapulco
Leaflet | Data by © OpenStreetMap, under ODbL
Git Graph
Cell 32 of 85

Mark the success/failed launches for each site on the map

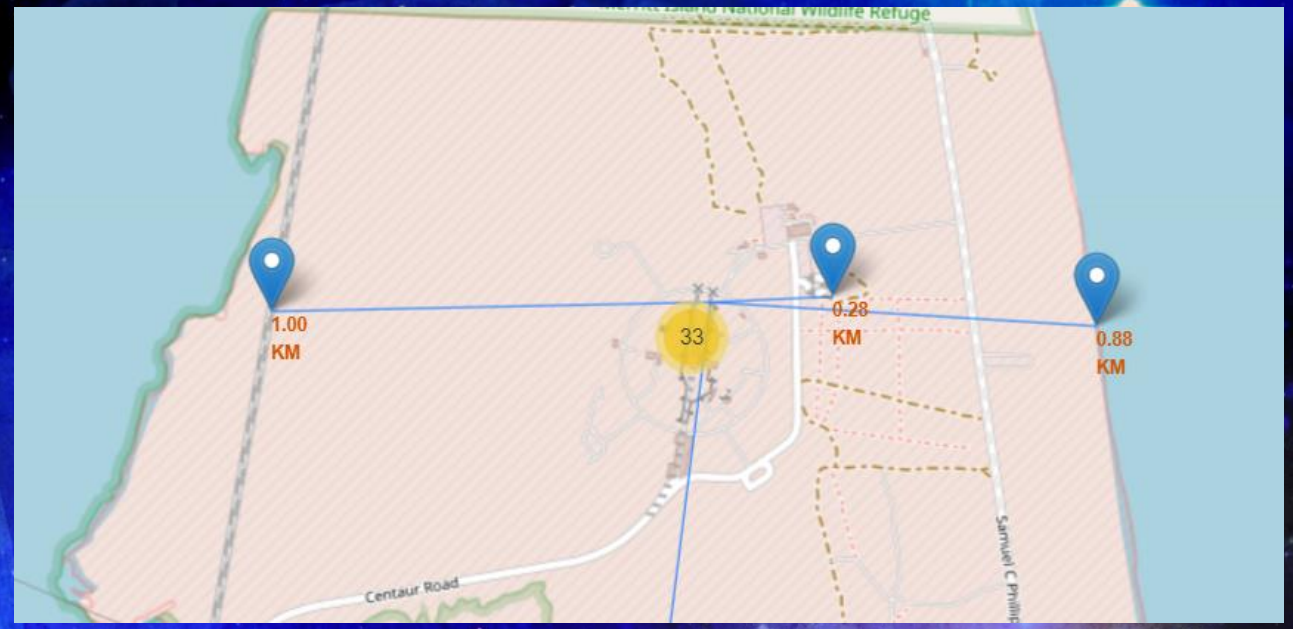


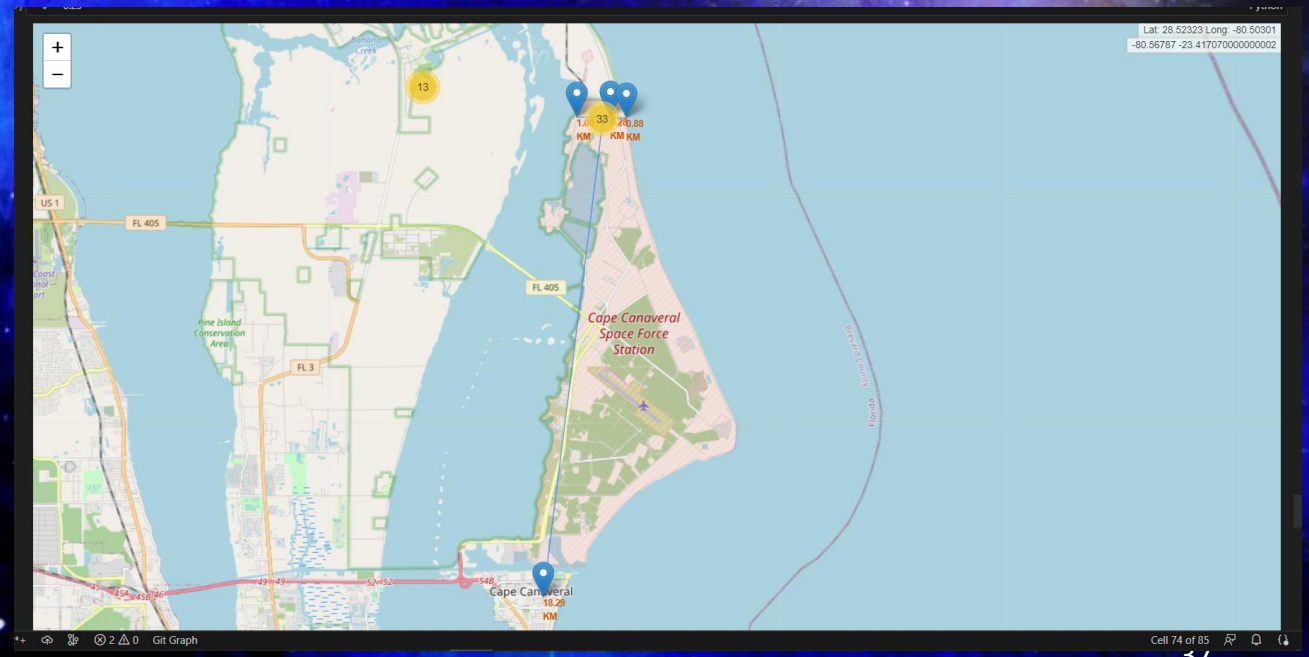


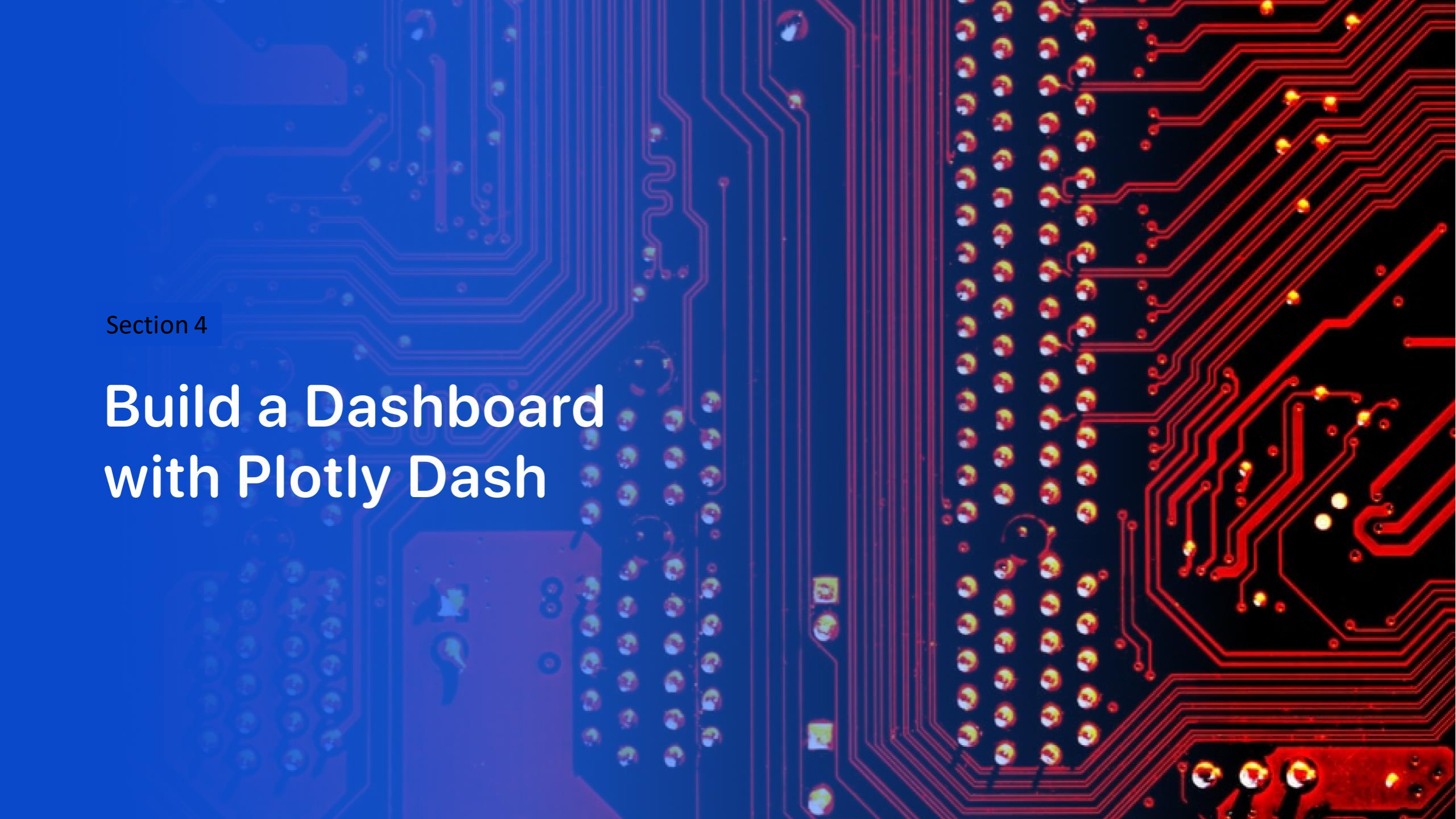


Folium Map

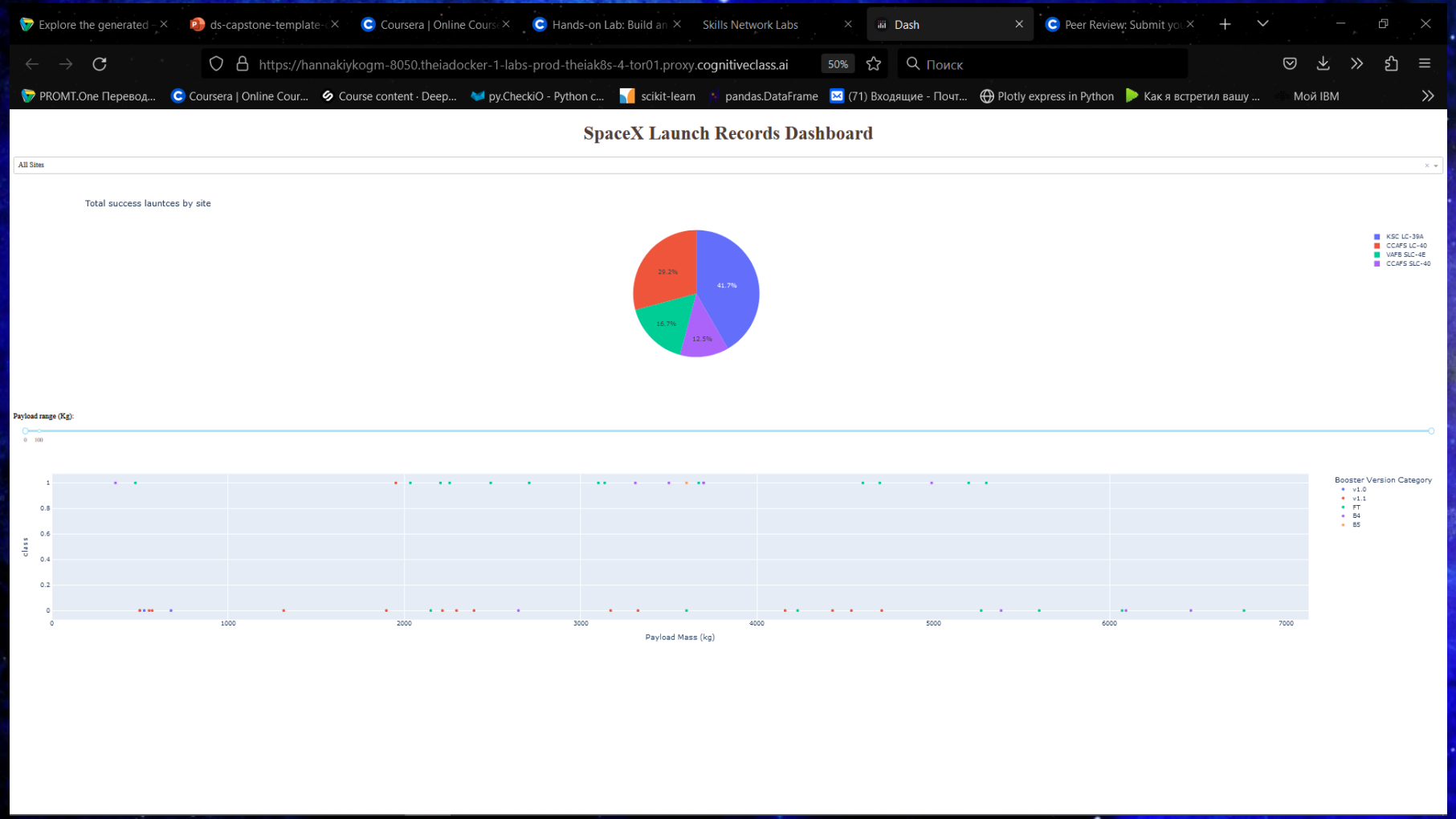
- I draw a line between a launch site to its closest city, railway, highway





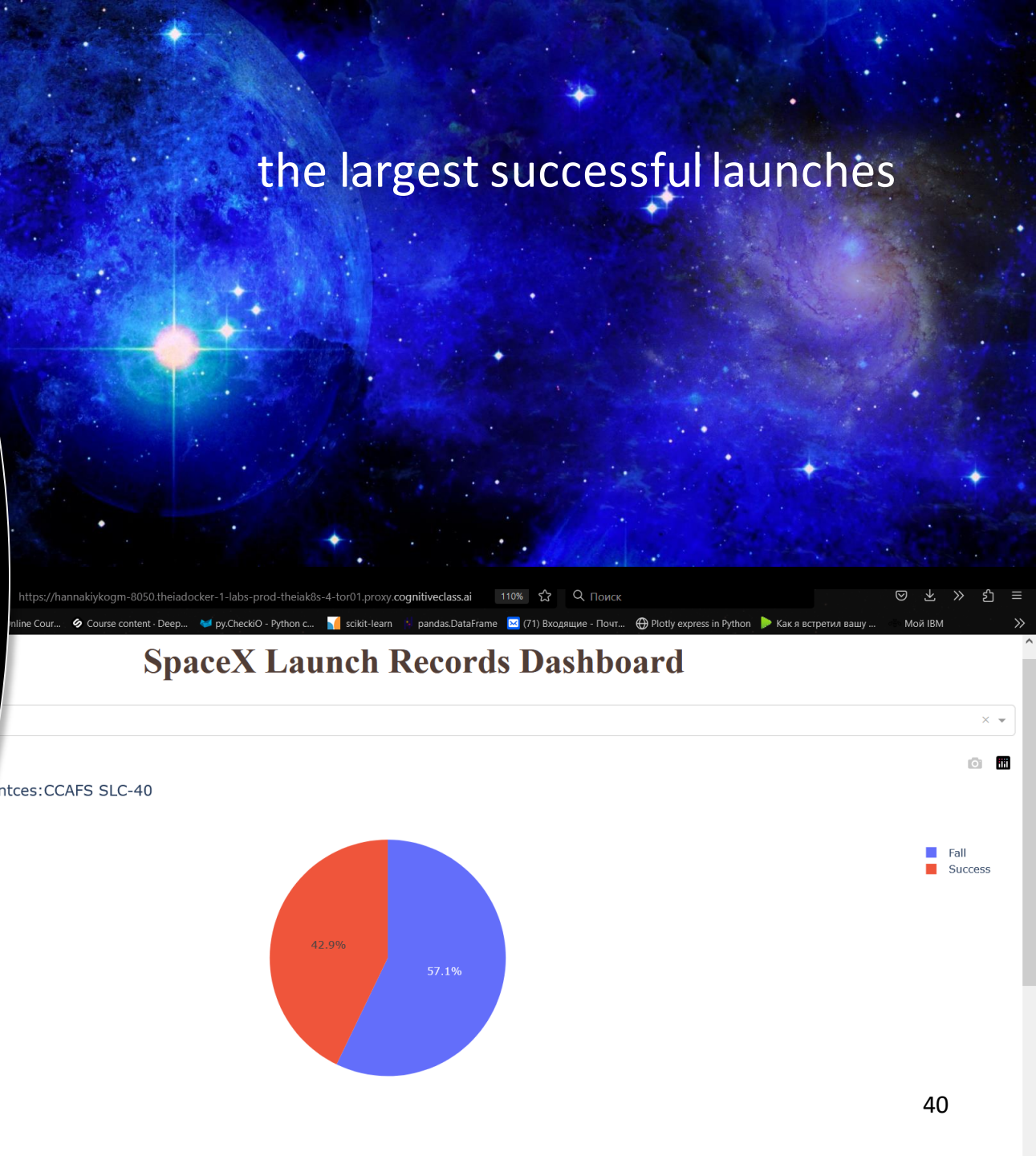

Section 4
Build a Dashboard with Plotly Dash

Dashboard


Explore the generated
ds-capstone-template
Coursera | Online Cours
Hands-on Lab: Build an
Skills Network Labs
Dash
Peer Review: Submit you
https://hannakiykogm-8050.theiadocker-1-labs-prod-theiak8s-4-tor01.proxy.cognitiveclass.ai
50%
Поиск
PROMT.One Перевод...
Coursera | Online Cour...
Course content · Deep...
py.CheckiO - Python c...
scikit-learn
pandas.DataFrame
(71) Входящие - Почт...
Plotly express in Python
Как я встретил вашу ...
Мой IBM
SpaceX Launch Records Dashboard
All Sites
Total success launtces by site
41.7%
29.2%
16.7%
12.5%
KSC LC-39A
CCAFS LC-40
VAFB SLC-4E
CCAFS SLC-40
Payload range (Kg):
0
100
Booster Version Category
v1.0
v1.1
FT
B4
B5
class
Payload Mass (kg)

CCAFS SLC-40

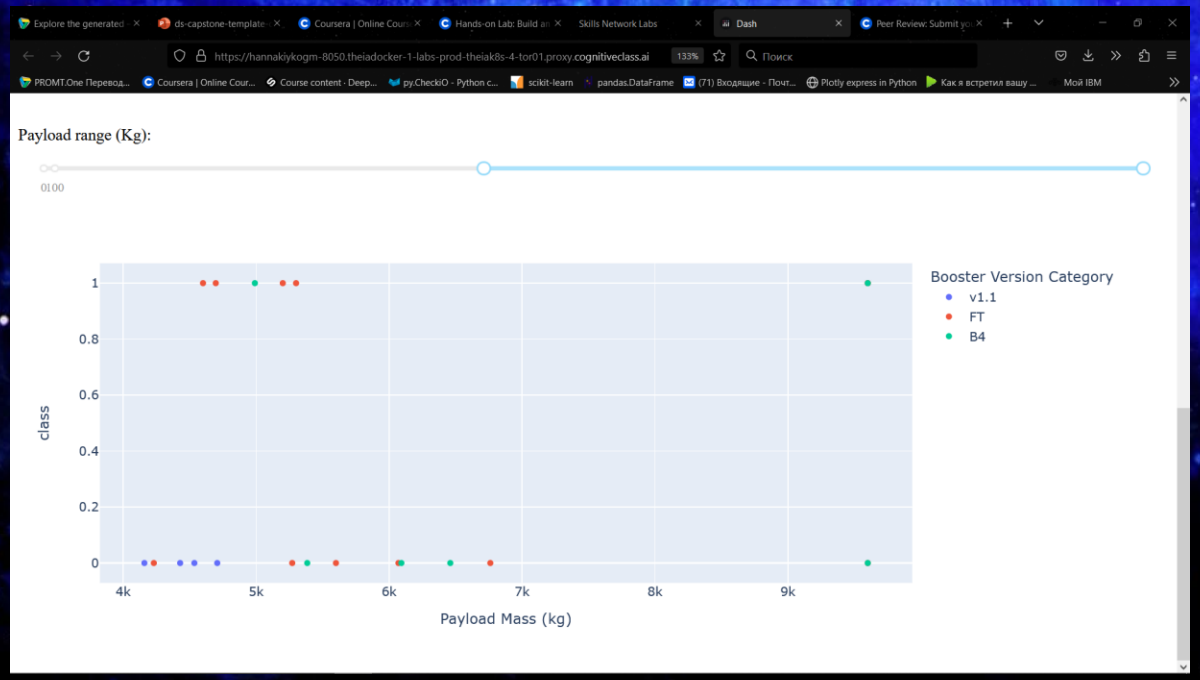
the largest successful launches



The answer for the following questions:

1.Which payload range(s) has the highest launch success rate?

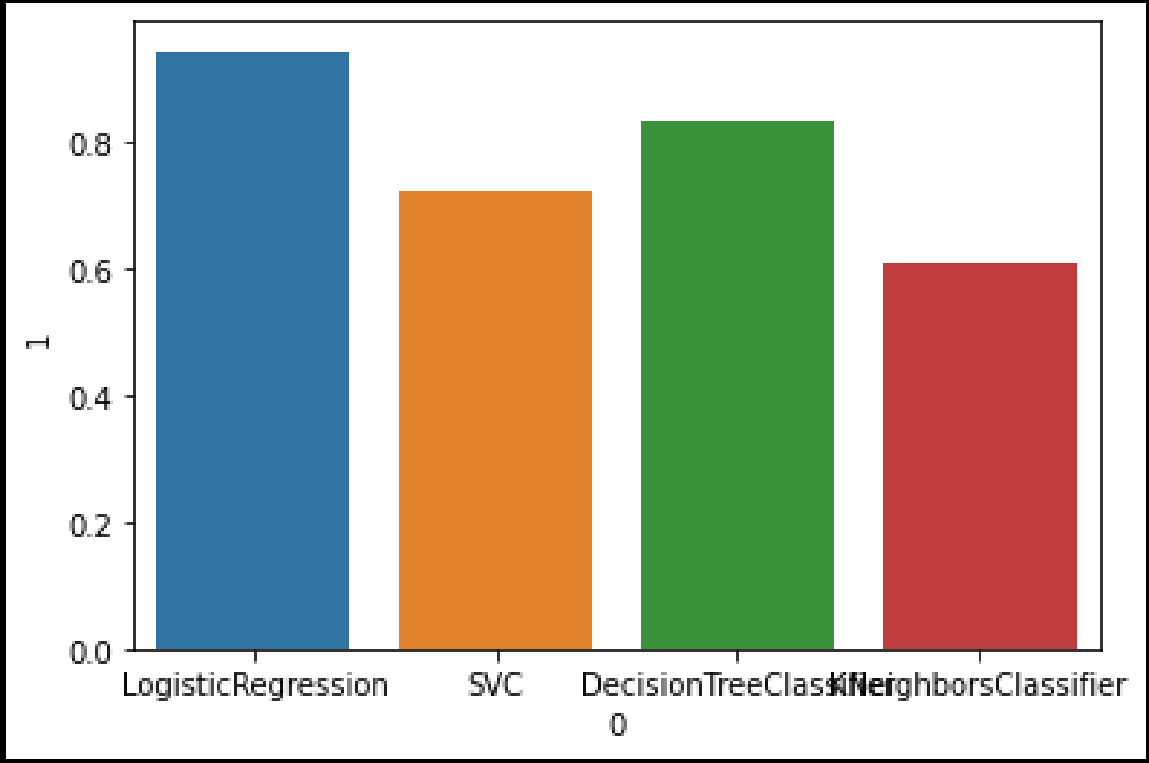
2.Which payload range(s) has the lowest launch success rate?



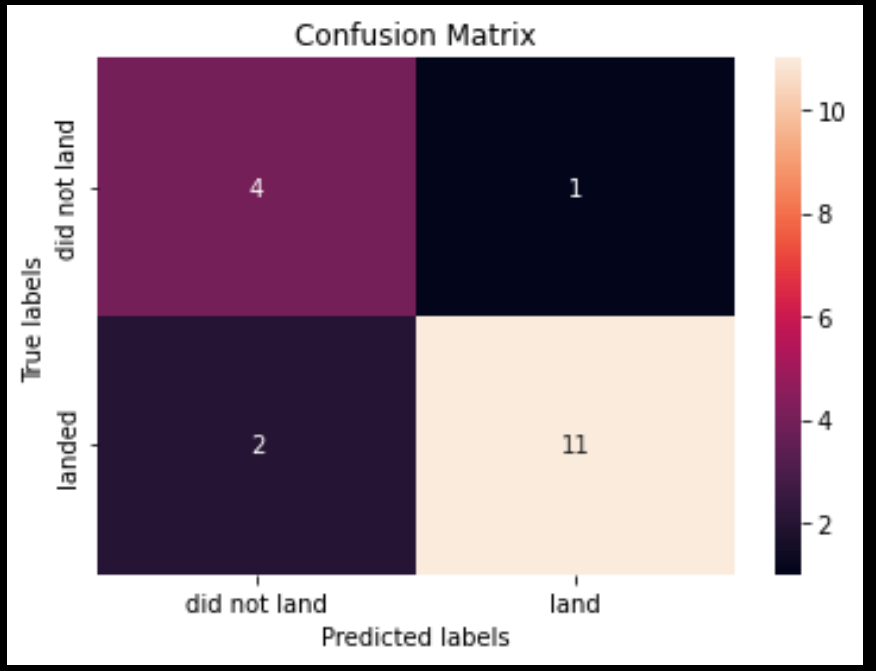
Section 5

Predictive Analysis (Classification)

Classification Accuracy



Confusion Matrix



Conclusions

- Now, we can predict the result with high accuracy!

Appendix


FlightNumber
PayloadMass
Flights
GridFins
Reused
Legs
Block
ReusedCount
Longitude
Latitude
Class
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75

Thank you!